UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales

Non-planar path planning for robot-based additive manufacturing processes

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Álvaro Guzmán Bautista

Bachelor's Degree in Industrial Engineering Technologies
Master's Degree in Industrial Engineering
Master of Science in Mechanical Engineering

Madrid, 2025

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales

Doctoral Degree in Mechanical Engineering

Non-planar path planning for robot-based additive manufacturing processes

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Álvaro Guzmán Bautista

Bachelor's Degree in Industrial Engineering Technologies

Master's Degree in Industrial Engineering
Master of Science in Mechanical Engineering

Under the supervision of:
Dr. Antonio Vizán Idoipe

Madrid, 2025

Title: Non-planar path planning for robot-based additive manufacturing processes
Author: Álvaro Guzmán Bautista
Doctoral Programme: Mechanical Engineering

Thesis Supervision:

Dr. Antonio Vizán Idoipe, Professor Emeritus, Universidad Politécnica de Madrid(Supervisor)

External Reviewers:

Thesis Defense Committee:

Thesis Defense Date:

This Thesis has been partially supported by the Spanish Ministry of Science, Innovation and Universities, through projects with grant numbers PID2021-124838OB-I00 and CPP2023-010651.

This Thesis has been partially supported by the Escuela Técnica Superior de Ingenieros Industriales, through projects with grant number ETSII-UPM24-PM02.

A todas y cada una de las personas que figuran a continuación,
que saben por dónde pasé y cuándo pude salir.

Institutional Acknowledgements

This Thesis was made possible thanks to the support of several institutions.

Thanks to the **Mechanical Technology Laboratory (IPF)** for providing the facilities and financially supporting this Thesis.

Thanks to the **Machinery Engineering Division (DIM)** for their support, guidance, and financially supporting this Thesis.

Thanks to the **Department of Mechanical Engineering (DIMEC)** of **Escuela Técnica Superior de Ingenieros Industriales**, **Universidad Politécnica de Madrid**, for financially supporting this Thesis.

Thanks to the **Centre for Automation and Robotics (CAR-CSIC-UPM)** for their support and guidance.

Personal Acknowledgements

Escribo unas pocas barras sobrevolando el Océano Ártico, desde el fin del mundo, a diez kilómetros sobre el suelo y a novecientos kilómetros por hora para expresar lo que llevo guardado durante más de cuatro años (en condiciones ambientales y cinemáticas parecidas a las que me encuentro ahora). Este texto que escribo a continuación es mi más sincera expresión, al desnudo, a tumba abierta, yendo de frente, como lo hice siempre (aPoLLo13, 2023). Como no puede ser de otra manera en los textos que escribo, hay un leitmotiv que espero que os guste.

Antonio Vizán, nos han pasado muchas (tal vez demasiadas) cosas en el transcurso de esta tesis. Tú me has enseñado una visión única sobre todo este mundo, las tres patas de Fabricación, los tres motivos por los que moverse. He de decir que en esta tesis he descubierto muchas facetas sobre mí mismo, sobre todo, mi necesidad de disfrutar del proceso. Ahora que soy todo un hombre (Mi Chain de Roque, 2018) finalmente puedo ponerme a actuar de doctor. Gracias por todo.

Ramón, Javi Tena, Juan de Juanes, Félix, Juan Carlos, Mónica, Carlos, Jon. En este tortuoso camino que llamo tesis doctoral, siempre me habéis acompañado, y sobre todo, enseñado millones de cosas. Con vosotros aprendí a volar, ya no me caigo (Chico Estrella, 2021).

Daniel Galán, Javier Badesa, David Orbea, Javier Laserna, gracias por la confianza depositada en mí, sobre todo en la etapa final. Tengo muchas ganas de seguir colaborando con vosotros y sacando cositas adelante, la verdad es que es un gustazo trabajar con vosotros. Tengo una sensación muy grande de que podemos sacar grandes cosas juntos, porque es obvio que nuestras reuniones son la junta de los jefes (H.I.E.L.O., 2020).

Pablo, José Antonio y Arley de Nano4Energy, gracias por haberme apoyado y siempre estar interesados en cómo me va, fuera de nuestras colaboraciones (que espero que siga habiendo). Aunque tengamos inconvenientes en el camino y aunque haya presión, es imposible que torzamos (Si Quieren Frontear, 2022).

A mis alumnos de TFX Adrián García, María Talavera, Victoria Gonçalves, María Méndez, Cristina Ramiro, Antón Levytskyy, Toni de Miguel y Alejandro Franco. Especiales agradecimientos a quienes habéis formado parte indispensable de este proyecto: Beatriz Pérez-Hickman, Pedro Sánchez, Lu Chen, Iñaki Echepare, Miguel Lerín, Víctor Bueno, Adrián Martínez, Jorge Lecocq, Charo Martínez, Elio Sánchez-Oro y Adela Jiménez. Ya sabéis todos, nos llevamos el 2026 (Tumbando el Club, 2019). No obstante, hay varias personas que también forman parte de la lista anterior que merecen un párrafo aparte.

Daniel Sanmartín, tú abriste el melón. Qué curioso ha sido que aparezcas en este texto y de esta manera. Probablemente tú, más que ningún otro, has visto mis mayores profundidades, el abismo. Contigo me he dado cuenta de este traje que llevaba puesto, negro como Venom (Givenchy, 2022). Poco a poco me lo estoy sacando de encima. Gracias tío.

Amparo Sancho, tú eres el origen. Me acuerdo como si fuera ayer de la poca confianza que deposité en Rami cuando nos dijo que querías hacer un TFM exprés en 3 meses. Ahora ya no dudo ni un segundo cuando me recomienda a alguien. Tus skills y tu resolución nos hicieron crear un imperio en tierra de nadie (Cascada, 2021). Tu trabajo fue la primera piedra, la que nos hizo confirmar que algo sabíamos, que algo teníamos que dar, que algo valíamos.

Laura Gil, tú eres el brillo. En la aventura que podríamos resumir como Los TFTs han pasado muchas cosas, y en todas ellas siempre me has apoyado. Me has hecho descubrir versiones de mí mismo que no sabía que tenía dentro, me has identificado frecuencias características cercanas a los cincuenta y dos hercios. Hemos conseguido aguantar los disparos y pelear con tipos malos (Si Me Sobrara el Tiempo, 2023). Gracias a ti, puedo decir que soy el primer doctor de mi familia.

Virginia, Alicia, todas las caras comprendidas entre Carol T y Yasmin la más viral e Irene, vosotras sois la ilusión. Tengo unas ganas tremendas de descubrir qué tan lejos va a llegar el Bootcamp. Sois mi proyecto más preciado ahora mismo. Yo lo soñé. Ojalá me veáis como vuestro copiloto para irnos de viaje a las estrellas (Hijo de la Noche, 2018).

A mis compañeros y amigos de DIM, sin ningún orden en particular: Soqui, Andrés, Javier Echávarri, Paco, William, Javi Segura, Gonzalo, Pedro, Rodri, Adrián Martínez, y por supuesto, a Mabel, Pedro, David, Edu, Óscar, Chema y todos los demás real Gs. Gracias por acompañarme y apoyarme en todos los momentos. Si estáis a mi lado se me hace más fácil afrontar esta vida (Rockstar, 2018). También mencionar a los de la vieja escuela, a Adrián De Blas, a Guli y, obviamente, a Alex Abou-Assali, que ha terminado siendo una parte clave tanto de este proyecto como de los que vendrán.

Hay gente que tengo que mencionar en particular en este punto, empezando por ti, Muru. Te agradezco muy mucho la amistad que hemos conseguido mantener a lo largo de estos años. Siempre me has apoyado, te has preocupado por mí y has velado por mí, especialmente cuando no era feliz. A partir de ahora estamos preparados para ir como zombies, volando bajito por las avenidas (Volando Bajito, 2021). Bueno, en tu caso igual tienes que volar más alto, tema representación corporativa, ¿no?

Juanma, desde que nos conocimos cuando era feliz, siempre he tenido una afinidad especial para trabajar contigo. Eres una persona que me ha inspirado mucho, con la que además comparto ciertas peculiaridades (la dinámica, la Suica, el berliner techno...), y sobre todo, me ha inculcado el gusto por hacer las cosas bien, y el go hard or go home. Tú también confiaste en mí antes de mis fallos, antes de mis logros, antes de todo (don't liE, 2023), y por ello, te doy las gracias.

José Luis, gracias, gracias, gracias, por todo el apoyo y confianza que nos has brindado a lo largo de estos años. Gracias por haber estado ahí, por haberme escuchado, por haberme calmado cuando me salía veneno por los dientes. Gracias por haberme mostrado la templanza y el talante que hay que tener en las relaciones interpersonales. Aunque se esté por perder todo, nunca se pierde el honor (Ameri, 2024).

Sonia, gracias por siempre, siempre, siempre preocuparte por mí y escucharme. Tenemos que seguir metiéndole gas hasta que salga el sol, vamos a intentarlo con ganas, como si no nos importara (Sol, 2021).

Alex, somos los últimos guerreros Saiyajin, la generación perdida, somos los pibes del fondo (Hello Cotto, 2017). Pero igual que nos metimos en esto, sabremos salir, gritando, levantando las manos al cielo y compartiendo nuestras energías. Métele con todo y yo estaré aquí para apoyarte.

Andrea, en ti he encontrado una compañera que nunca me imaginé. Tengo que serte muy sincero, soy tu mayor fan, tengo una necesidad por dentro muy grande de que llegues tan lejos como te quieras proponer. Además de eso, tengo unas ganas tremendas de compartir asignaturas contigo. Creo que somos la alianza perfecta, si estamos juntos solo hay beneficios (RCP, 2024), como se ha demostrado en las (de momento) pocas veces que hemos arrimado el hombro para los proyectos. Eres una auténtica crack y voy a luchar todo lo posible para que sigas aquí.

Miguel Clavijo, GOAT. Creo que en el breve tiempo que estuvimos trabajando juntos conseguimos crear verdaderamente algo nuevo, junto con Enrique, hicimos nuestra propia Antezana 247 en Fabricación. Fuimos artistas, empresarios, productores, la evolución de todas las décadas anteriores (TRAP.N.EXPORT, 2019). El hueco que dejaste fue gigante, pero ten por seguro que lo estamos rellenando siendo fieles a nosotros mismos.

Ramiro García. Tu forma de pensar es el motor del cambio. Ya sea regando ríos, haciendo LEGOS o construyendo ciudades (de LEGO también), tu mentalidad es tan fuerte que puede tumbar montañas (creo que sabes cuál). Tienes mi amistad, mi lealtad (algo que muchos no entienden, Antes de Perderte, 2022), mi respeto y, sobre todo, mi tiempo. Eres un auténtico maestro en lo que haces.

Adrián López, nunca he tenido suficientes palabras para agradecerte la responsabilidad emocional que has tenido conmigo, especialmente durante los peores años. Nada de lo que hemos hecho juntos hubiera salido adelante sin tu perspectiva. Eres la mano amiga que decidió mirar abajo y ayudarme, estábamos abajo y ahora estamos en la cima (hARAkiRI, 2023). Eres fundamental en todo este documento. Lo hemos conseguido, King.

Enrique Chacón. El primero en confiar en mí. Sabes que este éxito es más tuyo que mío. Sin tu apoyo absoluto y en todo momento, contra todo y contra todos, nada de esto hubiera pasado. Ahora que ya he salido de este trance, es la hora de abrir la boca y decir: "No drama kid, no es para mí, quizá para vos y para tu team (Vuelta a la Luna, 2019)". Seguiremos haciéndolo Modo Diablo, dentro y fuera de los escenarios.

En otro orden de cosas.

Josecar, Pau, Bronzer, Playmo, Bishe, Darkay, Damn, Sebas, Odi, Lorenzo, David, Manu, Víctor, Isma, Rels, Rafita, Dima, Juanjo, Muru, Antonio (mi primo político), Blanca (mi prima política), Rubén (mi otro primo político), Patricia, Erik. Esta también es para vosotros. Os tengo pegados en la mente, como los pies en el dance floor (Si Te Sentis Sola, 2018).

A mis familiares, es decir; mis tíos Gabino y Alicia, Esther y Francis; mis primos Fran, Marina y Rubén. Esta os la dedico, por ser quienes me seguís desde chico (Bottas, 2022). Muy especialmente a mis padres y a mi abuelo, quien no podrá venir a verme sacar los cuernos de diablo. Sé que no soy bueno para esto, que a veces estar conmigo deber ser molesto (No Me Llores, 2018), pero gracias a vosotros no me he quedado sin cartas en la baraja.

Treste, quisiste ser el rey y no un cuatro de copas.

Ali, estamos donde dijimos que íbamos a estar, ¿los demás donde están? No los vemos.

Vuestro amigo y vecino,

Álvaro

Abstract

Additive manufacturing (AM) still relies on planar, three-axis deposition, a constraint that imposes stair-step surfaces, orientation-dependent strength, and extensive support material. This PhD Thesis aims to demonstrate that non-planar material-extrusion paths can be generated and executed repeatably on an industrial six-axis robot, turning robot-based, polymer extrusion non-planar AM (RBNPAM-MEX-P) from a laboratory curiosity into a dependable process with sub-millimeter accuracy.

The research pursues four objectives. First, it establishes a mathematically rigorous slicing strategy valid for any C3-continuous axisymmetric surface. Second, it defines an strategy for infill planning and realization, based on the previously established mathematical framework. Third, it designs a compact manufacturing cell whose thermal, kinematic, and control subsystems can realize those trajectories. Lastly, it integrates planning and hardware into a single workflow that an operator can run from "file to part" without expert intervention.

Development begins with cubic and sixth-degree Bézier embedded maps that parameterize curved layers in closed form. Exact metric tensors and Christoffel symbols convert geodesic computation into a stable boundary-value problem, while a tessellation-cleansing routine produces ready-to-slice meshes. A geodesic-repulsion algorithm scatters interior points quasi-uniformly; a traveling-salesman heuristic then links them into collision-free Hamiltonian paths whose spacing reproduces any commanded infill density within five percent. These algorithms generate both solid walls and homogeneous non-solid interiors without projecting back to a plane.

On the hardware side, a UR10 collaborative robot was characterized kinematically; joint-space repeatability propagated through the Jacobian yields workspace error ellipsoids that guide fixture placement. Finite-element thermal studies validated a cartridge heater, nylon insulator, and conical coupling that warm a cylindrical build platform to 70 degrees Celsius in three minutes while protecting the flange. Twin ESP32 controllers implement PID regulation, timed-relay modulation, and bidirectional serial messaging with extruder and build platform; all power and control electronics reside in a single DIN-rail panel, producing a RBNPAM-MEX-P cell.

Calibration is split into two fast, low-cost routines. An extrinsic procedure, executed after any fixture change, probes a cubic totem with a flange-mounted laser to recover the six-degree-of-freedom worktable and sensor frames. An intrinsic routine, triggered before every build, re-levels the heated platform and re-zeros the sensor in under one minute. Coordinate-measuring-machine tests confirm absolute positioning errors below 0.4 millimeters, competing with far costlier vision or tracker systems.

The workflow orchestrates offline slicing, online warm-up, calibration, continuous joint-space streaming, and synchronized extrusion. Three case studies validate performance: a shark-scale infill on a convex barrel, a concave hourglass shell, and successive iterations of a cylindrical spring, with satisfactory results. Switching from Cartesian interpolation to continuous servo-movement raises effective deposition speed an order of magnitude and eliminates polymer

build-up. The complete cell therefore converts NPAM planning into tangible parts with cycle times limited only by extrusion and robot physics.

In conclusion, the Thesis delivers a unified geometric, computational, and experimental framework that closes the gap between non-planar path planning and shop-floor execution. Embedded-map slicing, quasi-uniform infill generation, calibration, and thermal hardware together prove that RBNPAM-MEX-P can achieve predictable accuracy, smooth surfaces, and reliable adhesion on standard collaborative-robot equipment, paving the way for higher-performance, support-free additive manufacturing.

Resumen

La fabricación aditiva sigue dependiendo de la deposición plana en tres ejes, lo que genera superficies en escalón, anisotropía en las piezas y gran consumo de material de soporte. Esta Tesis pretende demostrar que la extrusión de material no plana sobre un robot industrial de seis ejes puede ejecutarse de forma repetible y con precisión submilimétrica, convirtiendo la fabricación aditiva no plana robotizada basada en extrusión de polímero (RBNPAM-MEX-P) en un proceso fiable y repetible.

El trabajo persigue cuatro metas. Primero, se elabora una estrategia matemática de laminado válida para cualquier superficie C3 continua de tipo axisimétrico. Segundo, se concibe una estrategia para plantear y realizar estructuras de relleno interior basada asimismo en la estrategia matemática definida previamente. Tercero, se diseña una célula de fabricación compacta cuyas prestaciones térmicas, cinemáticas y de control permiten materializar trayectorias curvadas. Por último, se integra planificación y hardware en un flujo de trabajo que lleva el modelo a la pieza sin intervención experta.

La base geométrica se asienta en mapas encajados de Bézier cúbicas o de sexto grado. Sus tensores métricos y símbolos de Christoffel exactos convierten el cálculo de geodésicas en un problema estable; un limpiador de mallados garantiza variedades válidas. Un algoritmo de repulsión geodésica dispersa puntos cuasi-uniformemente y una heurística tipo viajante enlaza caminos Hamiltonianos cuya separación reproduce la densidad de consigna con fidelidad. Así, es posible generar paredes sólidas y rellenos homogéneos sin recurrir a proyecciones.

En hardware, se caracteriza cinemáticamente un UR10; la propagación de su incertidumbre articular define elipsoides de error que guían la posición de los utillajes. Simulaciones termomecánicas validan una plataforma cilíndrica con un cartucho calefactor, aislante de nylon y acople cónico capaz de alcanzar 70 grados centígrados en tres minutos sin sobrecalentar la brida. Dos ESP32 ejecutan el PID, la modulación temporizada y la mensajería serie compatible con ROS2; toda la electrónica se organiza en un cuadro DIN seguro y limpio.

La calibración se divide en dos rutinas diferenciadas. La extrínseca, tras cada cambio de configuración de la estación, sondea un tótem cúbico con láser para fijar los sistemas de referencia de la mesa y el extrusor. La intrínseca, previa a cada pieza, nivela la plataforma y reajusta el cero en menos de un minuto. Ensayos en máquina de medición por coordenadas sitúan el error absoluto por debajo de 0,4 milímetros, comparable a sistemas ópticos mucho más costosos.

El flujo de trabajo orquesta laminado offline, calentamiento, calibración, ejecución de trayectoria y extrusión sincronizada. Tres casos lo validan: un infill de “escama de tiburón” sobre un barril convexo, una plataforma cóncava tipo reloj de arena y sucesivas versiones de un muelle cilíndrico, con resultados satisfactorios. Además, al pasar de interpolación cartesiana a servomovimiento continuo, la velocidad efectiva aumenta un orden de magnitud y desaparece la acumulación de polímero. El ciclo queda limitado solo por la física de extrusión y el robot.

Por consiguiente, la Tesis ofrece un marco geométrico, computacional y experimental que enlaza la planificación no plana con la realidad industrial. El laminado mediante mapas

encajados, el relleno cuasi uniforme, la doble calibración y el hardware térmico prueban que la RBNPAM-MEX-P puede lograr exactitud, acabado y adherencia predecibles usando robots colaborativos estándar, abriendo camino a una fabricación aditiva sin soportes y de mayores prestaciones.

Table of Contents

List of Figures

List of Tables

Abbreviations and acronyms

AM	Additive Manufacturing
NPAM	Non-Planar Additive Manufacturing
RBNPAM	Robot-based Non-Planar Additive Manufacturing
MEX	Material Extrusion
MEX-P	Polymer Extrusion
DfAM	Design for Additive Manufacturing
BJT	Binder Jetting
MJT	Material Jetting
PBF	Powder Bed Fusion
SHL	Sheet Lamination
VPP	Vat Photopolymerization
DED	Directed Energy Deposition
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
TRL	Technology Readiness Level
CPPS	Cyber-Physical Production Systems
UV	Ultraviolet
SLA	Stereolithography
FDM	Fused Deposition Modeling
ABS	Acrylonitrile-butadiene-styrene
PLA	Poly-(lactic acid)
SLS	Selective Laser Sintering
SLM	Selective Laser Melting
EBM	Electron Beam Melting

LOM	Laminated Object Manufacturing
UAM	Ultrasonic Additive Manufacturing
CNC	Computer Numerical Control
PETG	Poly-(ethylene terephthalate glicol)
CLFDM	Curved-layer Fused Deposition Modeling
CLFFF	Curved-layer Fused Filament Fabrication
C/PEEK	Carbon continuous long fiber-reinforced polyetheretherketone
NURBS	Non-uniform Rational B-Spline
WAAM	Wire + Arc Additive Manufacturing
TPMS	Triply Periodic Minimal Surface
RBAM	Robot-based Additive Manufacturing
DOF	Degree(s) of Freedom
PLC	Programmable Logic Controller
CT	Computed tomography
TCP	Tool center point
STL	Standard Tesselation Language
TSP	Traveling Salesman Problem
NN	Nearest Neighbors
EBNS	Euclidean Blue Noise Sampling
ERE	Euclidean Repulsion Energy
GBNS	Geodesic Blue Noise Sampling
GRE	Geodesic Repulsion Energy
PHC	Pythagorean-hodograph curve
ROS	Robot Operating System
URDF	Unified Robot Description Format
FG	Functional group
FS	Functional solution

RTD	Resistance temperature detector
PVA	Poly-(vinyl alcohol)
FEM	Finite Element Method
IDE	Integrated Development Environment
LiDAR	Light Detection and Ranging
CMM	Coordinate Measuring Machine

Chapter 1

Introduction and Objectives

The present Thesis is focused on the development of a robot-based manufacturing cell for Non-Planar Additive Manufacturing processes, especially Material Extrusion Additive Manufacturing. This Chapter will provide a brief overview of the background in which this Thesis is embedded, with some insights on the context and the vision. Additionally, this Chapter will provide the motivation behind this Thesis, as well as its objectives and structure.

1.1 Background

This Section will cover the framework in which this Thesis is embedded, beginning with some basic notions on Additive Manufacturing (AM), as well as its transformation towards Non-Planar Additive Manufacturing (NPAM) and Robot-based Non-Planar Additive Manufacturing (RBNPAM). In a more particular scope, this Section will also cover the Manufacturing Technology Laboratory vision on manufacturing, which will serve as the main vehicle for the structure of this Thesis. Combining both of these scopes of background will serve as the main motivation for the present Thesis.

1.1.1 From conventional Additive Manufacturing to Non-Planar Additive Manufacturing: the context

Additive Manufacturing has gained relevance in the industrial production and prototyping landscape over the past three decades. By building objects layer by layer [1], AM has enabled designers and manufacturers to create geometries that were once thought impossible or prohibitively expensive through traditional subtractive or formative processes. The technology has proven indispensable for rapid prototyping, allowing engineers to iterate designs quickly while reducing material waste and manufacturing lead times.

Despite these advantages, many conventional AM processes follow a strictly planar approach: each layer is deposited horizontally, and the build platform (i.e., the part of the machine which provides the surface upon which the building of parts is done) only moves in the vertical (z) direction as each layer is completed. This planar method simplifies the hardware design and path planning, but it also imposes significant limitations [2]. For instance, planar layering can lead to anisotropic material properties, often resulting in weaker inter-layer bonds compared to in-plane properties [3]. Additionally, certain geometries—especially those with steep overhangs or internal cavities—require extensive support structures, complicating post-processing and increasing waste [4].

In recent years, researchers have begun to question whether new technologies could circumvent the inherent constraints of planar layering. This research drive has given rise to a new paradigm: Non-Planar Additive Manufacturing [5]–[7]. These advanced systems enable the deposition of material along freeform paths in three-dimensional space, creating components with enhanced mechanical performance and more intricate geometries [8].

NPAM is to be combined with current Design for Additive Manufacturing (DfAM) strategies [9]. These strategies include the generation of freeform internal geometries [10], topological optimization [11], custom texturing [12], multi-material AM [13] and multi-material embedding [14], or directly-assembled products. Enabling NPAM in an industrial context would facilitate many novel industrial applications in several sectors, namely, the aerospace and automotive industries [15], biomedical applications [16], heat exchanging components, and more [17].

However, NPAM necessitated some technological advancements to perform the transition between a concept and an industrial reality. In this sense, one of the main technological enablers of NPAM is the introduction of robotic manipulators in manufacturing processes, giving birth to Robot-Based Non Planar Additive Manufacturing (RBNPAM) [18]. According to standard ISO/ASTM 52900:2021 [1], there are seven main categories of AM processes:

- **Binder Jetting (BJT)**, which consists of the joining of powder materials by means of a liquid bonding agent selectively deposited.
- **Material Jetting (MJT)**, where droplets of material are selectively deposited to create parts.
- **Powder Bed Fusion (PBF)**, where thermal energy selectively fuses regions of a powder bed.
- **Sheet Lamination (SHL)**, where sheets of material cut to specific patterns are bonded to form a part.
- **Vat photopolymerization (VPP)**, where liquid photopolymer is selectively polymerized by means of selective photon application.
- **Material Extrusion (MEX)**, where material is selectively deposited through a nozzle.
- **Directed Energy Deposition (DED)**, where focused thermal energy selectively fuses materials as they are being deposited.

Of these seven categories, only the last two (MEX and DED) may be integrated in RBNPAM technologies. This is due to the fact that other processes are not geometrically compatible, since they require the use of powdered or liquid feedstock materials, which are heavily constrained by gravity.

The introduction of robotics in NPAM comes along with several advantages. In addition to the previously mentioned design freedom, enhancement of mechanical properties, enhancement of surface quality, ease of post-processing, embedding of functionalities, and energy consumption and waste reduction [19], RBNPAM can also allow for larger build volumes, or even on-site manufacturing. Furthermore, in terms of operational costs (i.e., power consumption and maintenance) and upfront investment, robotized systems are more approachable for industries seeking to keep their batch sizes low, while focusing on high customization [20]. This production prospect is the ideal scenario for NPAM technologies. As robotic arms and associated technologies become more widespread, these costs will likely decrease, making non-planar AM a more accessible option for a range of industries. These advantages make RBNPAM processes very attractive from an industrial and research perspective.

Nonetheless, the integration of robotics within the NPAM context comes with some limitations. Despite the potential of non-planar deposition, planning paths in full $\mathbb{R}^6$ space is not trivial [21]. Current commercial slicers are specialized in planar processes, where machine kinematics are relatively simple. As such, these slicers do not yet accommodate multi-axis movements. Currently, there is a need for specialized path generation methods that accounts for machine kinematics, collision detection, nozzle orientation, and real-time feedback in a systematic manner [22], [23].

Advanced simulation tools can help predict issues such as collision between the robot arm and the part, or collisions with support fixtures. Additionally, understanding the robotized station and how to properly calibrate it can result in significant accuracy improvements [24]. However, within this context, bridging the gap between simulation and real-world performance remains a challenge. To tackle this problem, it is necessary an extensive mathematical knowledge (in addition to general robotics and NPAM know-how) to create systematic solutions that can be integrated into industrial workflows, and result in the main research questions posed in this Thesis.

1.2 Motivation

Several advantages may be obtained by developing a systematic way of defining trajectories: shorter manufacturing set-up stages, asynchronization of set-up and operation, better understanding and application of DfAM strategies, and the extrapolation onto other robot-based manufacturing processes, such as machining. This is the main Motivation of this Thesis. Additionally, most of the examined solutions for RBNPAM model slicing and path planning are currently disconnected from the commonplace industry-level robot operation. By implementing a more robust online set-up procedure, it may be possible to elevate RBNPAM processes from a low technology readiness level (TRL) laboratory curiosity, to an industrial, high TRL reality.

A secondary motivation of this Thesis is to develop a RBNPAM cell that integrates in-house offline and online set-up procedures, as well as in-house process control to be able to obtain further RBNPAM-related research advancements. This motivation combines the broader context of Non-Planar Additive Manufacturing as an emergent AM strategy with the vision of manufacturing of the Manufacturing Technology Laboratory. As such, the present Thesis is the fundamental pillar of the Further Project, fostered by the Manufacturing Technology Laboratory for the exploration of this set of processes, and catalyzes its historic vision onto a novel manufacturing technology. This Thesis is set to provide the foundation for many of the tasks depicted in Figure 1.1. Specifically, the knowledge areas and tasks tackled in this Thesis will be introduced in the Objectives Section, and extensively developed throughout the document.

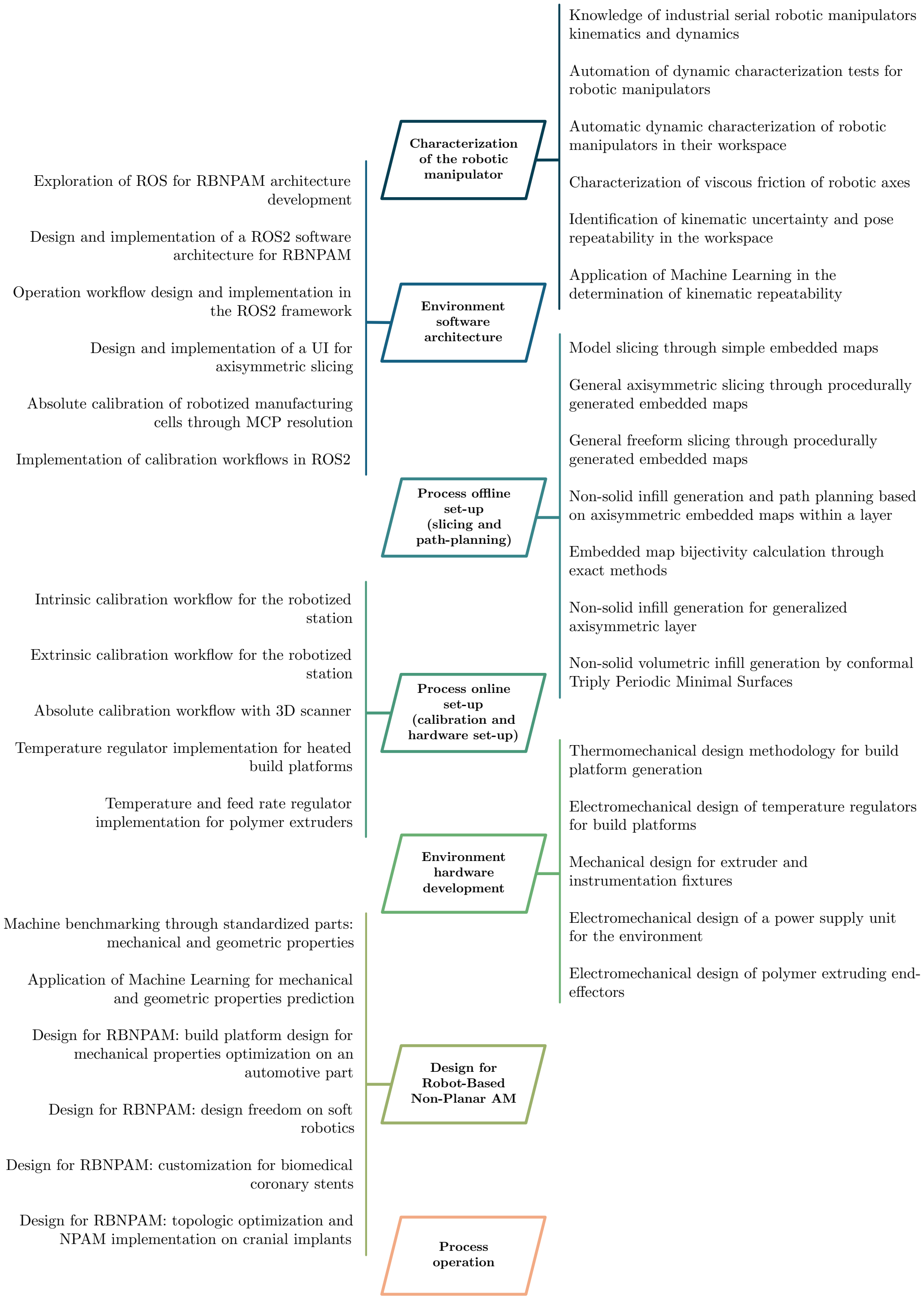


Figure 1.1: The Further Project areas of work.

From top to bottom, the Further Project is arranged in seven packages:

- **Characterization of the robotic manipulator.** This work package comprehends the ongoing understanding of the manufacturing machine, the robotic manipulator. As such, this area aims to establish procedures to identify, and subsequently, predict the kinematic and dynamic behavior of the robotic manipulator. The results obtained from this knowledge influence future equipment design decisions, as well as defining procedures for offline set-up.
- **Environment software architecture.** This work package comprehends the development of a software architecture to govern the operation of all equipment elements. This activity falls outside the scope of the present Thesis, and, for the purposes of this Thesis, a simplified version has been utilized. Nonetheless, it is actively being developed and is a fundamental part of the project.
- **Process offline set-up.** Process offline set-up, as previously mentioned, is fundamental in asynchronizing operation planning from the actual operation. As such, this work package is one of the core activities in the Further Project. The process offline set-up includes the creation of build platforms, their parametrization, the slicing of solids, and the robotic path planning for non-solid infills. Each of these subactivities will be deeply explained throughout the present Thesis.
- **Process online set-up.** In order to make the process repeatable, it is necessary to perform a proper process online set-up procedure. In robotized manufacturing cells, this often includes the determination of all reference frames involved in the process. This procedure is often known as calibration. This process will distinguish between two types of calibration: intrinsic (if the reference frame includes the robot kinematic architecture and is therefore dependent on configuration) and extrinsic (the robot kinematic architecture is not involved in the determination of stationary reference frames). Both these calibration procedures are instrumental to the success of the RBNPAM cell. Additionally, process online set-up includes having all hardware elements ready to engage in the process, namely, the build platform and the extruders.
- **Environment hardware development.** The manufacturing equipment would not be complete without the development of the necessary hardware. This work package focuses on the design of all non-commercial components, fixtures, or tooling in general. Some brief insights on hardware development will be provided throughout this Thesis.
- **Design for Robot-Based Non-Planar Additive Manufacturing.** In order to make use of all the benefits of RBNPAM, it is necessary to introduce some Design for Manufacturing concepts. This work package explores the application of current Design for AM and NPAM onto the RBNPAM cell developed, as well as establishing novel design ideas specifically aiming at RBNPAM processes. In addition to this, some equipment benchmarking tasks are being done in order to estimate the repeatability of the overall workflow. This is done by means of evaluating the mechanical properties and tolerances achieved by other AM equipment, and transferring these concepts onto the RBNPAM cell.

- **Process operation.** Lastly, the combination of all above packages leads to the process operation, or, in other words, the RBNPAM cell making a part. This Thesis presents the first steps in this regard. At the time of writing this Thesis, efforts are being made in order to fully integrate all work packages, and the completion date is estimated to be the first quarter of 2026.

In addition to the Further Project, this Thesis is embedded within several competitive project grants from a range of institutions. The main supporters of this Project have been:

- **The CYBERCELL project**. The CYBERCELL project (Cyber-Physical Production System for intelligent manufacturing cell planning), awarded by the Spanish Ministry of Science, Innovation and Universities to the team the present author belongs to, consists of the application of Cyber-Physical Production Systems (CPPS) onto robotized manufacturing cells. The present Thesis has been supported by it in the conception of the robot-based manufacturing cell, and many of the results of this Thesis are applicable in both scenarios.
- **The CPP2023 project**. Through grant call “Public-Private Collaboration Projects 2023” (CPP2023) by the Spanish Ministry of Science, Innovation and Universities, the present Project was supported by means of project grant “Production cost reduction of current and future parts based on new composite materials”, awarded to the team the present author belongs to. This project is centered around the design and implementation of RBNPAM processes in the rapid manufacture of tooling for composite parts. As such, the topics of this Thesis concerning path planning and infill planning have been supported by this project, and the results will be made extensive use of throughout the development of the CPP2023 project.
- **First Projects 2024**. Grant call “First Projects 2024”, organized by ETSI Industriales, Universidad Politécnica de Madrid, focuses on fostering the access of novel researchers from the institution into the environment of competitive grant calls. Within this context, the present author and their team have been awarded with project grant “Robotic absolute calibration through geometric pattern recognition by means of Maximum Clique Problem resolution”. This project has supported the calibration tasks developed in this Thesis.
- **The ETSI Industriales Mechanical Engineering Department**. In addition to the previously cited competitive grants, the Mechanical Engineering Department has also financially supported the development of the present Thesis throughout all its stages.
- **Center for Automation and Robotics CAR-CSIC-UPM**. The Center for Automation and Robotics has supported this research by kindly sharing their knowledge on robotic systems and architecture, as well as by actively participating in projects CPP2023 and First Projects 2024.

1.3 Hypotheses and Objectives

Combining all previous context and motivation, the main Hypotheses of this Thesis are:

- It may be possible to generalize the robot path planning generation and part slicing in RBNPAM contexts through a rigorous mathematical framework.
- This mathematical framework may also solve the generation and manufacture of non-solid NPAM infills in parts.

Accounting for these hypotheses, and in order to validate them, the following Objective is posed:

The conception, implementation and demonstration of a systematic path planning strategy for the automation and simplification of Robot-based Non-Planar Additive Manufacturing processes.

This broad objective can be organically subdivided into four sub-goals:

- **Establishing a mathematically rigorous NPAM slicing and path-planning strategy.** This sub-goal is intimately related to the offline set-up flow. It is considered that advancements in offline set-up are valuable in an actual industrial manufacturing context, as they allow for asychronicity between operation planning and operation execution. This asychronicity greatly reduces process downtime, as the process engineering is able to plan parts while the machine is executing the process.
- **Defining a non-solid infilling generation methodology.** Based on the previously established mathematical framework, an NPAM non-solid infill generation and path planning methodology may allow for significant material, energy, build cycle time, and cost savings. This topic has not been properly addressed yet from a scientific viewpoint, and may be solvable through the application of the previously established strategy.
- **Devising an experimental setup for RBNPAM, including the understanding of the machine, as well as the design and implementation of the tooling.** In order to feasibilize RBNPAM processes and to demonstrate the power of the presented mathematical framework, it is necessary to understand all existing physical components, or to design and implement tooling to bring the process to a reality.
- **Postulating and validating the overall operation workflow.** This sub-goal includes the offline set-up flow previously mentioned, and extends towards defining the online set-up flow (i.e., calibration and hardware set-up) and process operation proper. These flows are to be validated through the manufacture of actual parts.

Lastly, these objectives can be expressed in terms of tasks, which are the fundamental units of work. In this Thesis, the tasks which will be done to reach the outlined goals are:

- Postulating model slicing through embedded map methods.
- Establishing a axisymmetric embedded map generation methodology for uniform model slicing.
- Developing a toolbox for axisymmetric embedded map handling.

- Making the axisymmetric embedded map generation methodology compatible with current commercial slicers.
- Devising a path planning algorithm for hollow infill axisymmetric sliced models.
- Postulating a method for non-solid uniform conformal infill mesh generation.
- Developing a workflow for non-solid uniform conformal infill path planning.
- Understanding serial robotic manipulator kinematics.
- Devising the testing needed for estimating the kinematic uncertainty of a robotic manipulator.
- Predicting the kinematic uncertainty of a robotic manipulator within a confined workspace.
- Designing and manufacturing a cylindrical build platform for methodology demonstration, as well as its regulator.
- Designing and manufacturing a regulator for a polymer extruder.
- Designing and manufacturing an experimental manufacturing setup.
- Conceptualizing an RBNPAM manufacturing set-up and operation workflow.
- Postulating and applying an extrinsic calibration methodology for workspace reference frame estimation, independent from the robotic manipulator kinematic chain.
- Postulating and implementing an intrinsic calibration methodology for build platform reference frames estimation, dependent of the robotic manipulator kinematic chain.
- Demonstrating the feasibility of the RBNPAM manufacturing workflow through several case studies.

1.4 Structure of this document

The present Thesis is subdivided into eight Chapters.

The first and present Chapter, **Introduction**, provides a brief overview of NPAM as a whole, and the NPAM project active at the time of writing this Thesis. Additionally, Chapter 1 sets the motivations behind the studies, as well as the objectives of this work.

The second Chapter, **State of the art**, aims to provide the reader with an in-depth review of current NPAM literature, including benefits of NPAM, current slicing methods, as well as robot-based systems in the context of NPAM.

The third Chapter, **Theoretical basis**, explains the fundamental concepts that support the present Thesis.

The fourth Chapter, **Offline Set-up**, depicts the foundational principle that drives the present Thesis, the embedded map method for NPAM path planning. This definition is supported by several examples and a complete documentation of the developed toolset.

The fifth Chapter, **Non-solid infills**, expands the previous idea by generating NPAM infills based on the geodesic repulsion of points. With both Chapters 4 and 5, it is possible to define the complete trajectory of a part made by NPAM in a systematic way.

The sixth Chapter, **Experimental setup**, explains the design decisions and the implementation of a robot-based NPAM-MEX-P demonstrator cell, which will be used in the next Chapter to manufacture parts through this technology.

The seventh Chapter, **Manufacturing workflow**, provides the technology workflow required to manufacture parts, from the station configuration to the material extrusion on the axisymmetric build platform. Additionally, to put all the previous concepts into a manufacturing reality, several case study parts have been manufactured and discussed.

The eighth and last Chapter, **Conclusions**, serves as a means to close up the present Thesis, explaining the main achievements, limitations, future research prospects, and scientific contributions of this Thesis.

Chapter 2

State of the art

This Chapter aims to provide a comprehensive review of all scientific literature relevant to Additive Manufacturing, Non-Planar Additive Manufacturing offline planning, and Robot-based Additive Manufacturing systems. This compilation is structured in these three Sections, and provides a scientific foundation and justification of interest on the themes covered in this Thesis.

2.1 Additive Manufacturing: concepts and technologies

As presented in the Introduction, Additive Manufacturing (AM) is the set of manufacturing technologies that build parts by means of adding material, usually layer by layer. Standard ISO 52900 [1] include seven categories of processes, which are detailed below:

- Vat Photopolymerization (VPP): Vat photopolymerization involves a vat of liquid photopolymer resin that is cured solid layer-by-layer using a light source such as a UV laser or a digital light projector [25]. In a typical process like stereolithography (SLA), a focused light traces each cross-sectional pattern on the resin surface, causing the exposed liquid to polymerize and solidify; the build platform then moves to allow new resin to flow for the next layer. This technology offers very high resolution and fine feature detail due to the precision of the light-induced polymerization, often yielding parts with smooth surfaces. However, vat photopolymerization is generally limited to photosensitive polymer materials and usually requires support structures for overhanging geometries. Printed parts also often need post-curing (exposure to additional UV light or heat) to reach their full strength and stability after printing.
- Material Jetting (MJT): Material jetting creates objects by depositing droplets of a build material in a fashion similar to a 2D inkjet printer, followed by curing or hardening of these droplets to form solid layers [26]. Typically, the printer head jets a photopolymer or wax material which is immediately cured by an ultraviolet light source, building the part layer-by-layer with very high dimensional accuracy. This process can use multiple print heads to jet different materials, enabling multi-material parts or full-color prints in a single build. Unique aspects of material jetting include its ability to produce parts

with smooth surfaces and fine details without visible layer lines, and to combine soft and rigid materials or color inks within one object. On the other hand, material jetting is usually restricted to certain thermoset photopolymers or waxes, and the printed parts may have relatively limited mechanical strength or heat resistance compared to those made by powder-bed or extrusion-based methods.

- Binder Jetting (BJT): Binder jetting uses a two-material system: a powdered base material and a liquid binder that acts as a glue. In this process, a thin layer of powder is spread across the build area, and an inkjet print head selectively deposits droplets of binder onto the powder, adhering the particles in the desired pattern for that layer [27]. The build platform then lowers and another layer of powder is spread, repeating the process layer-by-layer. The result is a “green” part embedded in loose powder; after printing, the unbound powder is removed and the part is typically cured and then sintered or infiltrated in a furnace to achieve its final density and strength. A unique advantage of binder jetting is that it involves no high temperatures during printing (the powder is not melted in situ) so there are no thermal stresses, and it can accommodate a wide range of materials (from metals and ceramics to sand or polymers) at relatively high speed. It also allows for full-color printing by mixing colored binders. However, binder-jetted parts are porous and fragile prior to post-processing, and even after sintering they may exhibit lower density or strength than parts made by powder bed fusion unless additional infiltration or densification steps are applied.
- Material Extrusion (MEX): Material extrusion is exemplified by the Fused Deposition Modeling (FDM) technique, in which a continuous filament of thermoplastic material is fed into a heated nozzle and extruded as a thin strand that solidifies to form a layer of the object [28]. The nozzle moves according to the cross-sectional geometry, laying down rasters of molten polymer, and the build platform typically lowers one layer thickness for each new layer to be deposited. This family of processes is one of the most accessible forms of AM, with relatively low-cost machines widely used for prototypes and functional parts. Material extrusion can work with a variety of thermoplastics (ABS, PLA, nylon, composites, etc.), and it builds structures with internal infill patterns that can be adjusted for weight or strength. Unique aspects include the need for support structures for overhangs (often printed in a second, dissolvable material) and the anisotropic properties of the final parts. Due to the fact that each new layer fuses to the one below, the strength in the vertical (layer) direction is typically lower than in the horizontal plane. The process may produce visible layer lines on the surface, and achieving very fine detail is harder compared to some other AM methods, but it excels in simplicity and is well-suited for rapid prototyping and custom plastic components.
- Powder Bed Fusion (PBF): Powder bed fusion refers to a group of processes where a thermal energy source selectively fuses regions of a powder layer to build up a part, with the powder bed supporting the part as it forms [29]. In these techniques, a thin layer of powder (metal, polymer, or other material) is spread over a build platform, then a high-powered laser or electron beam scans the layer based on the digital slice geometry, melting or sintering the powder particles in those areas. The platform is then lowered and a fresh layer of powder is applied, repeating until the entire object is formed

within the powder cake. Examples of PBF include Selective Laser Sintering (SLS) for plastics and certain composites, and Selective Laser Melting (SLM) or Electron Beam Melting (EBM) for metals. The latter fully melt the powder to achieve near-wrought densities in metal parts. The key advantage of PBF is the ability to create complex, high-resolution parts often without dedicated support structures for overhangs (since unused powder surrounds the part, although metal PBF still requires supports to anchor parts and manage thermal stresses). PBF can produce functional parts with excellent mechanical properties (e.g., metal PBF parts can be up to 9̃9% dense, comparable to forged metal). Challenges of this family include the need for careful powder handling (as materials must be fine powders), relatively high equipment and operational costs (inert gas chambers, lasers, etc.), and issues like residual stresses, powder recyclability, and surface roughness that may necessitate post-process treatments.

- Sheet Lamination (SHL): Sheet lamination builds objects by bonding and stacking sheets of material, cutting each sheet to the desired shape of each cross-section before adding the next layer [30]. In one approach, Laminated Object Manufacturing (LOM), sheets of paper or plastic film coated with adhesive are successively glued together and cut with a laser or blade to outline each layer of the part. Another approach uses metal foils and ultrasonic welding to bind layers (Ultrasonic Additive Manufacturing), with periodic CNC machining to trim the edges to shape. The unique aspect of sheet lamination is that it can combine different materials in layer form and tends to be a relatively low-temperature, low-cost process – for example, LOM paper parts are inexpensive and quick to produce, and UAM can embed sensitive components (like sensors or fibers) between metal layers since it's a solid-state bonding process. However, the resolution and accuracy of sheet lamination are limited by the sheet thickness and the cutting method, and the parts may require additional finishing (for instance, excess material around the part must be removed, and edges might be rough from the layer cutting). This technology is often used for visual models, concept prototypes, or creating composite metal laminates, rather than for high-performance end-use parts, due to the materials and bonding involved.

- Directed Energy Deposition (DED): Directed energy deposition is an AM family wherein focused thermal energy is used to melt material as it is being deposited, typically in the form of a metal powder or wire feedstock, to create a part in open space or onto an existing surface [31]. A DED machine usually consists of a multi-axis robotic arm or gantry that moves an deposition head, which emits either a laser beam, electron beam, or plasma arc to create a melt pool. Material is simultaneously fed into this melt pool (for example, metallic powder blown through a nozzle or metal wire fed from a spool), and it solidifies to form a track that becomes a layer of the object. DED is commonly used for metals and is essentially a form of automated build-up welding; it can add material to repair damaged components or fabricate near-net-shape parts that are later machined to final dimensions. The process allows for high deposition rates and the fabrication of very large parts beyond the size of typical powder bed machines. It also enables functionally graded materials or alloy mixing by controlling feedstock composition in real time. On the downside, directed energy deposition generally produces relatively coarse resolution (the deposited bead is often on the order of 1-2 [mm] or more in width), so surface finishes are rough and fine features are hard to achieve without post-machining. Nonetheless, the DED family fills important niches in industry for repairing aerospace or military parts, adding features to pre-made components, and efficiently manufacturing bulky metal shapes with less material waste than subtractive methods.

Pictures of parts manufactured by each of these process categories is shown in Figure 2.1.

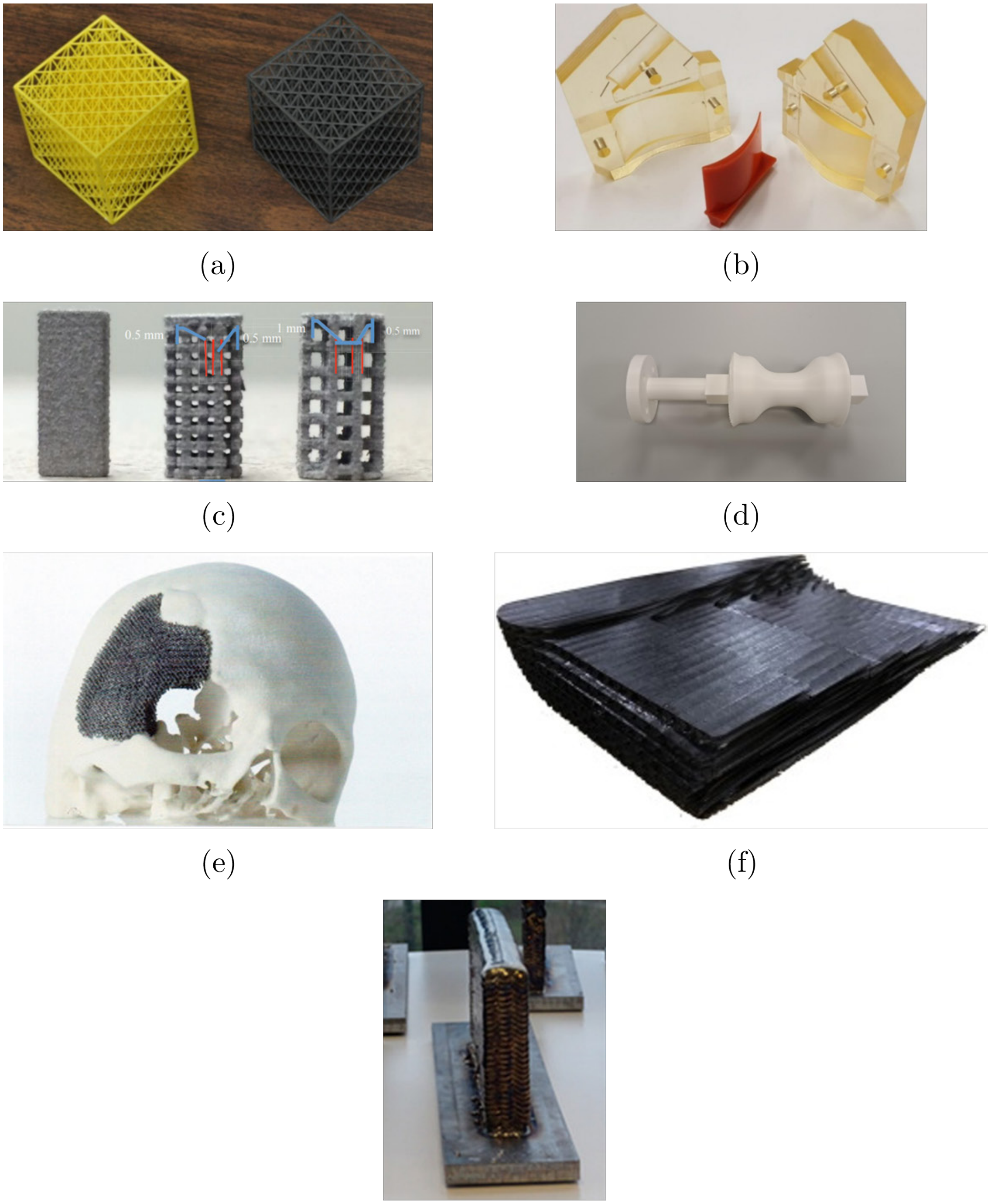
One common denominator of all these technologies is the introduction of layer-by-layer manufacturing. This poses a few challenges, especially regarding topics such as mechanical properties, and geometric and surface finish. Concerning mechanical properties, some of the issues include anisotropy between build and planar directions, porosity-driven loss of strength, surface-initiated fatigue, and the presence of residual thermal stresses. These issues trace back to the layer-wise thermal history that distinguishes all seven ASTM families of additive manufacturing, and their combined effect has kept many applications in the prototyping or repair domain rather than true load-bearing service, as emphasized in an across-family review of process–property linkages [38]. In this Chapter, a special focus will be made on MEX and DED technologies, as these are the most appropriate for robotization.

For MEX processes such as fused filament fabrication, the dominant mechanical limitation is weak inter-raster and inter-layer bonding. Polymer strands solidify in milliseconds, freezing in entanglement densities below bulk equilibrium; the consequence is a tensile strength reduction of 30–60% in the build (z) direction compared with injection-molded analogues. Thermal-kinetic modeling of acrylonitrile–butadiene–styrene shows that the time–temperature integral governing reptation can now be predicted and linked directly to weld-line fracture energy, enabling numerical estimation of ultimate strength and toughness for a given toolpath [39].

Once the physics of inter-layer bonding are understood, process parameters provide the next advantage. Experiments that combined artificial-neural-network tuning with genetic algorithms were able to raise the tensile strength of polycarbonate MEX specimens by roughly 20% through optimized raster angle, air-gap and extrusion temperature [40]. Complementary fatigue tests on carbon-fiber-reinforced PLA reveal that identical parameter changes produce diverging fatigue responses—high infill and 0° rasters favor static strength, whereas 45° rasters at lower density deliver longer fatigue lives, underscoring the trade-offs designers face [41].

Another study, provided by Bakhtiari and their colleagues [42], asserts that that parameters beneficial for static strength do not always translate to fatigue resistance, especially regarding build orientation. The most robust prints consistently use cross-over infill patterns, high infill densities and relatively thick layers, while the optimal raster orientation flips with loading mode (parallel to tensile stresses but perpendicular to compressive, flexural and bending stresses). A y-build orientation (specimen laid flat with its width parallel to the build plate) maximizes tensile and flexural life, whereas print velocity has only a secondary effect, enabling faster production without major fatigue penalties.

Beyond parameter tuning, reinforcement and post-processing strategies are being translated from classical composites into MEX. A 2024 synthesis paper cataloged feed-stock modification (nanofillers, discontinuous fibers), in-process hybridization (co-extruded continuous fibers, in-nozzle impregnation) and post-print treatments such as autoclave consolidation or microwave welding, reporting up to five-fold increases in specific stiffness when methodologies are



(a) (b)

(c) (d)

(e) (f)

(f)

Figure 2.1: Examples of parts manufactured by different AM technologies. (a) VPP [32]. (b) MJT [33]. (c) BJT [34]. (d) MEX. (e) PBF [35]. (f) SHL [36]. (g) DED [37].

combined [43].

DED parts, in contrast, start dense and metallurgically bonded but inherit the coarse, columnar grains and steep thermal gradients of welding. This microstructure gives high tensile strength parallel to the deposit tracks yet limited ductility and pronounced anisotropy. Svetlizky and their colleagues [38] quantified typical room-temperature tensile strengths (e.g., 950 [MPa] for laser-DED Inconel 718) and highlighted residual-stress magnitudes exceeding 300 [MPa] before stress relief.

Recent work couples DED with mechanical inter-layer deformation (e.g., rolling, hammer peening or ultrasonic impact) to break up epitaxial grains, accelerate recrystallization and reduce porosity. A 2024 state-of-the-art survey reports grain-size refinement from >100 [μm] to <10 [μm] and a 30–50% rise in yield strength, together with high residual-stress relaxation, without sacrificing deposition rate [44].

In summary, MEX parts are mainly limited by polymer chain interdiffusion and can match or exceed bulk properties only through thermal management, reinforcement and annealing, whereas DED parts begin with near-bulk density but require microstructural refinement and stress mitigation to attain wrought-like toughness.

On the other hand, geometric and surface-quality problems in additive manufacturing also follow from the layerwise nature of the processes, but the specific defect mechanisms differ markedly between MEX and DED. In both cases, dimensional fidelity and surface finish are governed by a competition between the physics of material delivery, local heat transfer and solidification, and the control system's ability to keep tool-center-point motion synchronized with those phenomena.

For MEX parts, the most pervasive geometric defect is warping: differential cooling between newly deposited beads and the colder substrate builds up shrinkage stresses that curl edges upward, displacing subsequent layers and compromising overall dimensional accuracy. Real-time force sensing on the print head has measured these out-of-plane loads and linked them to warp magnitudes, establishing a quantitative pathway for feedback correction; the experimental study that formalized this link reported average deviations of only 0.18 [mm] after implementing force-based control [45].

Surface defects in MEX arise from the staircase effect at shallow build angles, from bead surface ripple as molten polymer is sheared against previous roads, and from artifacts of incomplete flow control such as stringing and oozing. A recent parametric investigation of PLA prints demonstrated that stringing strands decrease almost linearly with retraction distance until a threshold of about 5 [mm], with the optimized process eliminating visible strings entirely; the work provides design charts derived from 125 factorial trials [46]. Complementary research on post-print mechanical burnishing has shown that ironing the outermost layers at elevated nozzle temperatures can push roughness values below 1 [μm], comparable to fine machining, while adding less than 2% to build time [47].

Beyond parameter tuning, thermal–mechanical modeling indicates that maintaining the deposited polymer above its glass-transition temperature for an additional 15–20 [s] significantly reduces inter-raster gaps, smoothing both internal and external surfaces; predictive equations for the critical dwell time were validated across ABS, PETG and nylon [48].

DED parts exhibit an entirely different defect landscape because material is delivered as wire or powder into a melt pool that solidifies in milliseconds. The high energy density and steep thermal gradients leave behind large residual stresses that manifest macroscopically as plate-like distortions or gradual "lean" of tall walls. Systematic X-ray diffraction and contour-method measurements on Ti-6Al-4V walls built by laser-DED reported peak tensile stresses exceeding 300 [MPa] and out-of-plane deflections above 2 [mm] on 100 [mm]-tall coupons; applying in-process inter-layer dwell or post-deposition thermal cycling cut both metrics by half [49].

Even when global shape is retained, the surface of a DED bead can be markedly rough because of melt-pool oscillations, partial solidification of ejected droplets and successive overbuilding where neighboring rasters overlap. A 2024 comparative study of powder-blown versus wire-fed DED showed arithmetic mean roughness ranging from 1 [μm] for stable wire-arc regimes to nearly 40 [μm] for blown-powder tracks deposited at low linear mass density; regression analysis identified volumetric energy density as the strongest single predictor of roughness [50]. Additionally, high-speed imaging and diagnostics further reveal that spatter generated at the leading edge of the pool can redeposit as partially sintered particles, forming "satellite" asperities that survive subsequent passes [51].

In summary, the geometry and surface state of MEX parts are dominated by polymer thermal contraction and flow-instability artifacts, whereas DED parts are shaped primarily by melt-pool hydrodynamics and solid-state stress evolution. For MEX, improved temperature control, retraction strategies and post-deposition surface treatments have already shown order-of-magnitude reductions in roughness and warping. For DED, lowering residual stress through path planning or in-process deformation, coupled with real-time optical monitoring to suppress spatter and humping, is proving effective at narrowing the gap between as-built and machined quality. Continued convergence of empirical studies with physics-based monitoring promises further reductions in both geometric deviation and surface roughness, accelerating the qualification of extrusion- and deposition-based additive parts for structural service. However, another promising solution that aims to mitigate this issues lies in Non-Planar Additive Manufacturing.

2.2 Non-Planar Additive Manufacturing

Non-Planar Additive Manufacturing (NPAM) is the deposition of material following curved layers, instead of the regular planar ones. The idea of letting each printed layer follow a curved surface rather than lie in a single plane first appeared as a theoretical off-shoot of early slicing research; Dolenc and Mäkelä sketched the concept in 1994 while trying to minimize cusp height during adaptive slicing [52]. The first experimental implementation came much later, when Diegel and co-workers demonstrated "curved-layer fused deposition modelling" (CLFDM) in 2011, using a modified FDM nozzle to deposit continuous, non-planar rasters that could embed conductive circuitry inside thermoplastic shells [53]. Since then, non-planar strategies have branched out into both polymer and metal additive manufacturing, driven by the twin goals of smoother surfaces and better stress-aligned filament or bead placement.

Algorithmic groundwork matured rapidly after 2015. A multi-axis process-planning study showed how geodesic distance fields can be used to generate collision-free, curved layers for arbitrary free-form solids, eliminating many support structures [54]. Parallel reviews formalized the emerging vocabulary—terms such as "curved-layer fused filament fabrication" (CLFFF) and "active-Z printing", and surveyed machine kinematics from tilting beds to robotic arms [55].

For MEX systems, the main practical advance has been variable-height, non-planar slicing on ordinary three-axis printers. Pelzer and Hopmann showed that letting layer thickness float inside a single curved layer reduces stair-step roughness by up to 76% and improves tensile strength by keeping raster paths tangent to load directions [7]. Complementary optimization experiments found that extrusion-multiplier and temperature settings dominate surface-quality gains once a curved path is adopted, confirming that non-planar printing can achieve sub-micron roughness values with only marginal increases in build time [56]. Beyond finish, curved deposition distributes filaments more uniformly across inclined faces, closing inter-raster gaps and raising inter-layer fracture energy. This behavior was captured in the broader NPAM literature survey [55].

When extra degrees of freedom are available, non-planar MEX can be taken further. Dai and colleagues' "support-free volume printing" used five-axis motion to orient each curved layer normal to gravity, printing bowl-shaped parts entirely without sacrificial material [57]. Robotic systems meanwhile rely on non-planar slicing coupled with real-time path smoothing to maintain constant bead width along doubly curved shells, as documented in an early slicing study for robot-assisted FDM [58].

In metal DED, non-planar thinking arrived via path-generation research aimed at thin-walled or repairing geometry on complex substrates. Liu and Xing recently published a helical, non-planar toolpath that lets laser metal deposition build overhanging cylindrical walls without pausing or reorienting the workpiece, maintaining uniform melt-pool energy and bead aspect ratio [59]. Finite-element studies confirm that aligning beads to the part curvature also mitigates bulging and residual-stress-driven distortion—as quantified on an arbitrarily curved 316 L steel wall where simulation and in-situ digital image correlation gave consistent deflection trends [60]. Beyond shape fidelity, non-planar DED opens design space for graded or multi-material architectures. To this effect, Squires and their colleagues [61]. used wire-arc DED to grow a radial bimetallic cylinder whose concentric deposition path produced compressive residual stresses and a 33–42% rise in compressive strength over monolithic controls.

Material research confirms that non-planar trajectories deliver tangible performance gains. A large-scale review of void-reduction methods highlights curved-layer deposition as the only strategy that simultaneously addresses inter-filament porosity and inter-layer adhesion without chemical post-processing [62]. Stress-field-aware infill generation for continuous-fiber composites aligns both fiber and filament with principal loads, raising the flexural modulus of C/PEEK ribs by 38% relative to planar lay-ups [63]. Isogeometric topology optimization further couples geometric design to NPAM reality; Wei et al. used smooth basis functions to produce porous infills whose iso-stress surfaces translate directly into printable curved toolpaths, halving von Mises peaks compared with pixel-based optimized counterparts [64].

At the mesoscale, Wang et al. derived a load-path-oriented gyroid design that re-orients unit cells along principal stress trajectories, achieving a 1.7× increase in energy absorption at equal mass [65].

Taken together, the shift from planar to curved layers offers several recurring advantages. It suppresses the staircase effect at shallow angles, improves surface integrity without post-machining, and lets filaments or weld beads follow principal-stress trajectories, boosting inter-layer bonding in polymers and reducing thermal stress concentrations in metals. Non-planar paths also cut support consumption, shorten print times for domed or sculpted parts, and enable conformal deposition onto existing components for repair, electronics, or biomimetic multimaterial builds. As tool-path planners now integrate real-time sensing and machine-learning optimization, non-planar additive manufacturing is evolving from an academic curiosity into a practical route for higher-performance, lower-waste production across both MEX and DED platforms.

It is clear that NPAM provide significant advantages over regular planar AM. However, one of the main challenges found in NPAM is the trajectory generation, also known as path planning. In recent years, there have been several procedures developed by other authors that aim to simplify this task. Ma and co-workers' NURBS-based adaptive slicer showed in 2004 that, if the tessellated model is interrogated as a smooth surface rather than as a stack of polygons, stair-stepping can be softened without compromising build time [66]. That insight laid the mathematical foundation for everything that is today called "non-planar", because it cast slicing as the extraction of parametric manifolds instead of z-indexed contours.

The first generation of explicitly curved-layer strategies appeared a decade later. Jin et al. [67] formulated a physical model for curved-layer fused deposition and derived a tool-path planner able to follow bowl-shaped surfaces with continuous bead orientation, reporting markedly smoother finish and more homogeneous strength along the build height (Figure 2.2). In parallel, other authors began to treat the print head as a five- or six-axis end-effector rather than a Cartesian Sharpie. Zhao et al.'s non-planar slicing and path generation module for a six-axis robot arm demonstrated that decomposition-based curved surface slicing can be executed repeatedly on industrial hardware, so long as collision-free orientation planning is integrated into the slicer itself [58].

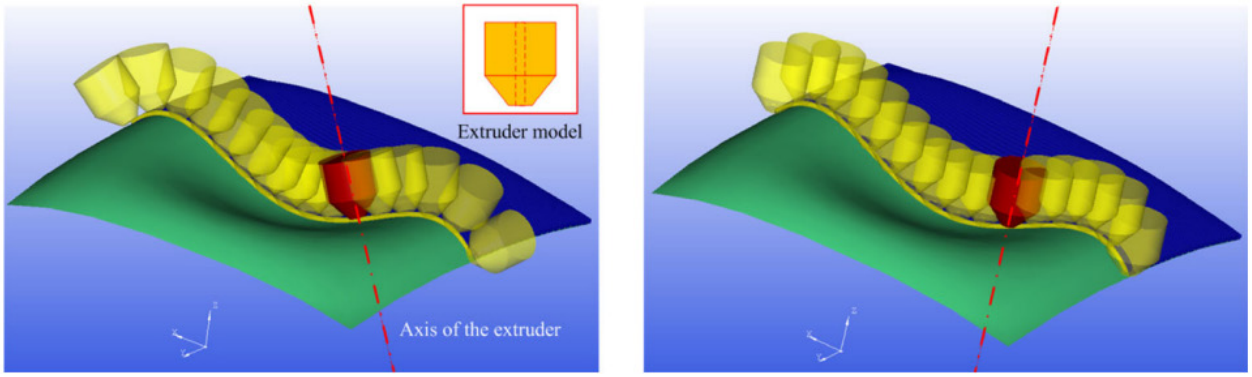


Figure 2.2: Extruder orientation-aware slicing method [67].

Once multi-axis motion was accepted, researchers explored which reference surfaces deliver the best compromise between kinematic accessibility and support elimination. Helical slicing removes layer seams by turning the entire model into a single continuous spiral; Yigit and Lazoglu [68] validated the method experimentally and quantified its seam-free benefit for cylindrical shells (Figure 2.3). A sister idea from the same authors (spherical slicing) expands that concept to onion-like shells. Their 2020 spherical slicer [69] needed only minor changes to the helical mathematics to let a robot lay beads normal to concentric spheres, yielding isotropic tensile response on domes and turbine shrouds (Figure 2.4). Liu et al. [59] recently generalized the approach: by mapping truss primitives onto a virtual sphere they could fabricate complex space-frame lattices completely support-free, printing node junctions in the same pass as primary struts (Figure 2.5).

Figure 2.3: Helical shell slicing method [68]. Both the final result and the toolpath are shown.

Alternative coordinate systems were proposed for special geometries. Munasinghe and Paul's radial slicing targets helical-shaped mineral-separation spirals; by revolving a ray around the part centre they reduced support by 70% compared with planar layers [70] (Figure 2.6). Shan et al. [71] went orthogonal to temperature instead of geometry: their isothermal-surface slicer

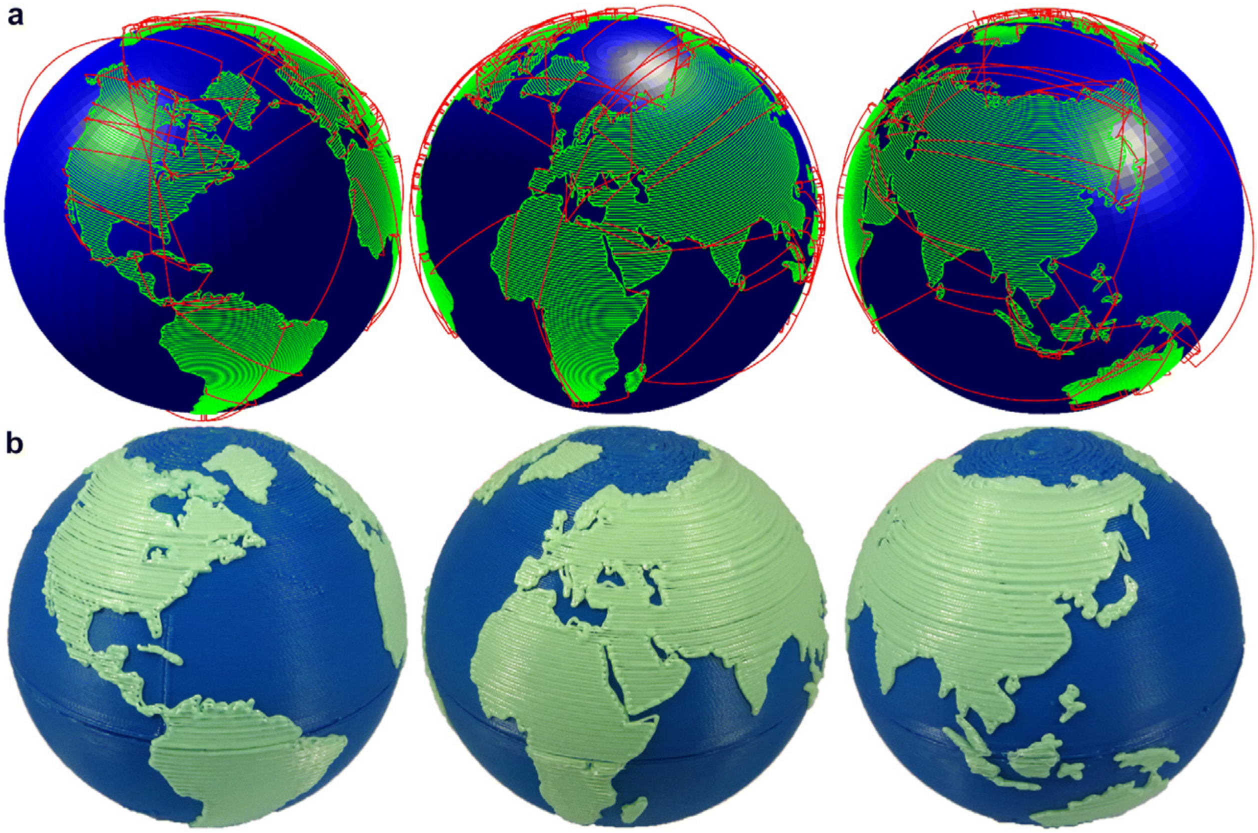


Figure 2.4: Helical shell slicing method [69]. (a) Toolpaths. (b) Final result.

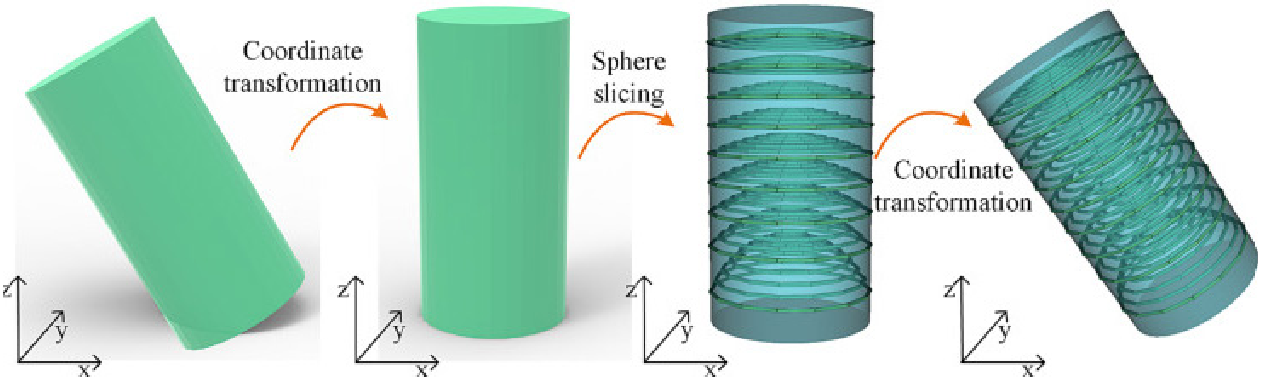


Figure 2.5: Improved spherical slicing, able to build free shapes [59].

tracks constant-temperature fields inside the part so that each curved layer is deposited under near-uniform cooling, mitigating residual stress in polymer parts with thin walls. An example of this is provided in Figure 2.7.

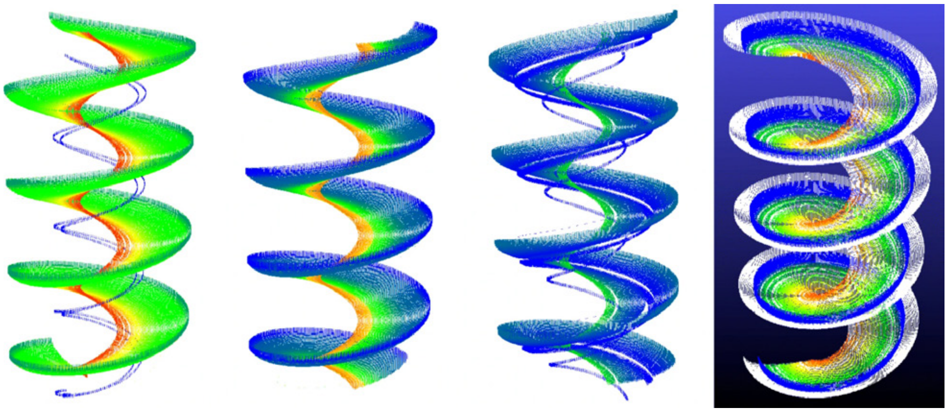


Figure 2.6: Different versions of improved helical slicing [70].

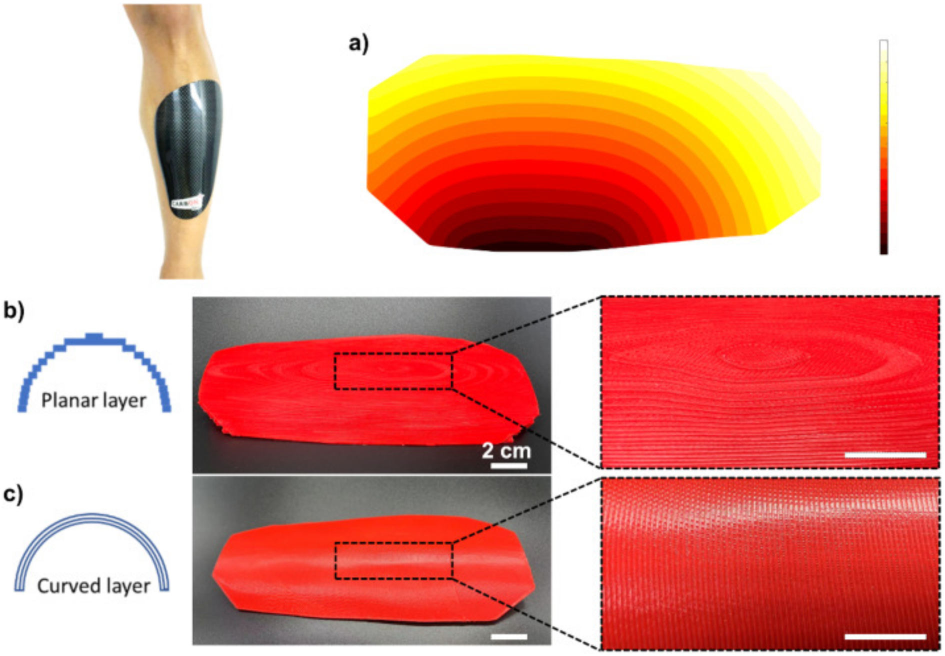


Figure 2.7: Example of isothermal slicing [71]. (a) Shin guard, part utilized for the study, along with the proposed heat source. (b) Planar result. (c) Isothermal NPAM result with improved surface quality.

Because no single surface suits every feature, hybrid schemes now dominate the literature. Zhao and Guo's mixed-layer adaptive slicer [72] decomposes the model, assigning planar, curved, or inclined layers to each sub-volume according to local curvature and overhang metrics; simulation shows 30–50% support reduction while keeping path length in check. Fortunato et al. automated that decision process end-to-end: their fully automatic non-planar algorithm analyzes tessellation roughness, picks candidate scalar fields, scores manufacturability, and outputs ISO-code without user intervention, opening NPAM to non-experts [73] (Figure 2.8).

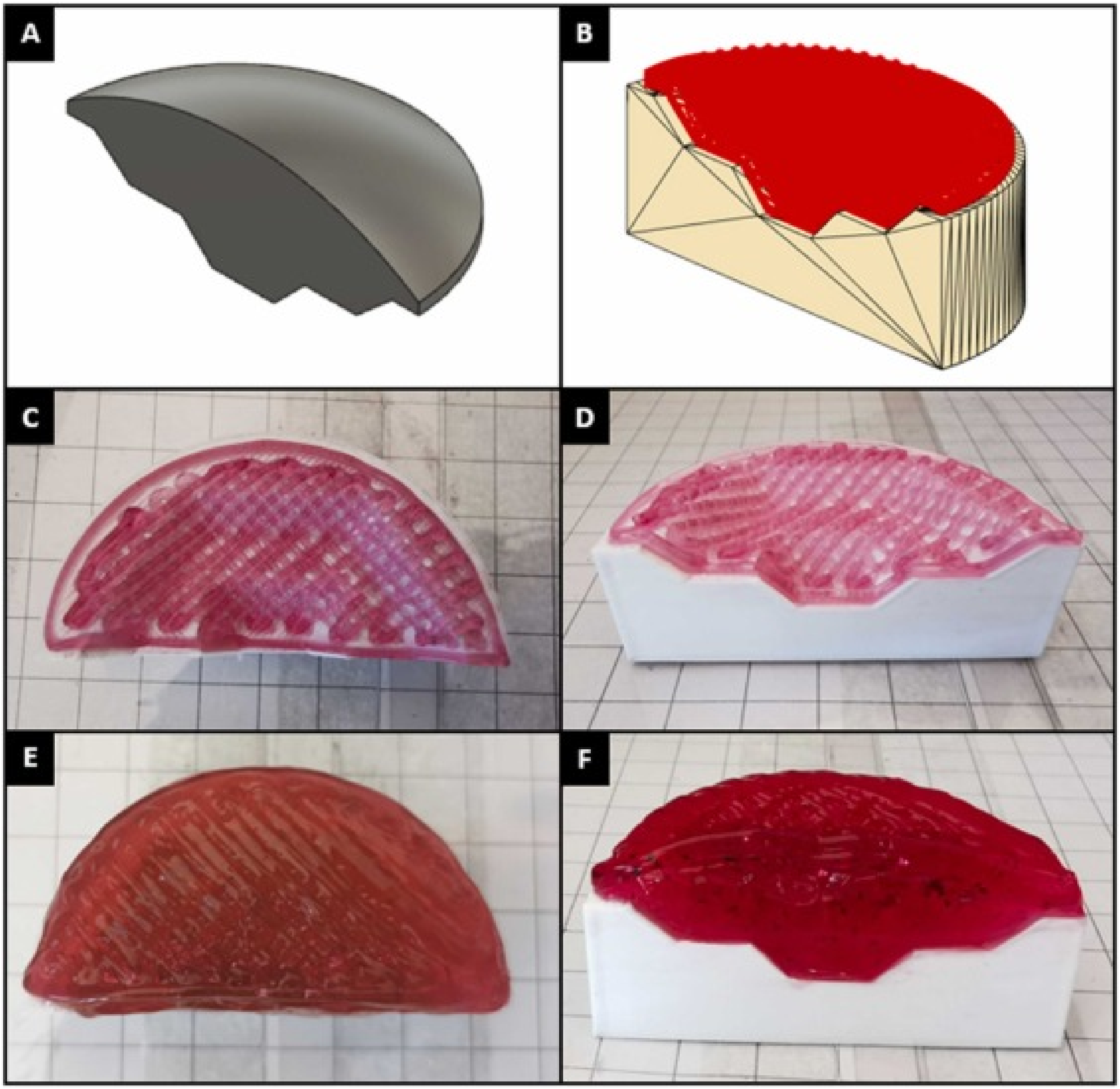


Figure 2.8: Example of end-to-end slicing [73]. (a) Part utilized for the study. (b) Generated trajectory. (c, d) First NPAM layer. (e, f) Final result.

Path planning alone does not guarantee performance; bead ordering, speed, and smoothing matter just as much. Allum et al.'s "ZigZagZ" trajectory blends slight out-of-plane oscillations into a nominally planar raster, boosting z-direction toughness by more than 240% in single-filament tensile coupons because inter-layer interfaces are no longer orthogonal to principal loads [74]. Rodríguez-Padilla et al. [5] exploited conformal tessellation to create lattice infill directly onto doubly curved skins, enabling pattern continuity across the entire part surface without tool re-orientation (Figure 2.9).

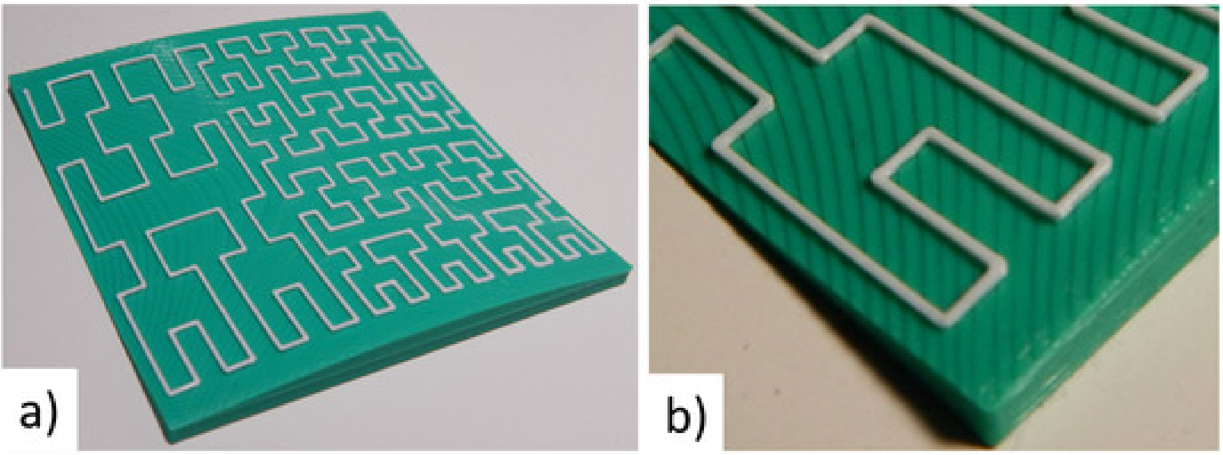


Figure 2.9: Example of conformal slicing [5]. (a) Hilbert path on a doubly curved surface. (b) Detailed view.

Li et al. integrated rotary and tilt axes into a purpose-built machine so that thin-walled shells are laid in one support-free pass, their metrology revealing sub-0.2 [mm] geometric deviation over 400 [mm] diameters [4]. Compact systems achieve similar freedom; Han et al. combined a translational stage with a gimbal in a five-degree-of-freedom table-top printer and validated fully self-supporting prints in polymer foams, opening NPAM to laboratory settings where industrial robots are impractical [75].

As it can be seen, curvature-aligned deposition demands new algorithmic foundations. Early adaptive medial-axis planners for wire-arc processes already hinted at the potential of geometry-driven paths [76], but modern extrusion research formalizes tool-orientation and collision constraints as integrated optimization problems. Li et al. introduced a support-free lattice-aware slicer that treats skin and infill as a coupled field; by solving for scalar values whose iso-surfaces satisfy both overhang and stress criteria they eliminated auxiliary scaffolds in free-form brackets [77].

Continuous graph-based planners extend the idea to sparse interiors: Gupta et al. mapped filament trajectories onto a weighted graph and solved a traveling-salesman variant to obtain retract-free, bead-consistent paths that shortened build time of non-solid demonstrators by 27% [78]. Maze-inspired rasterization schemes generate evenly spaced strands with controllable tortuosity for damping applications [79], while Eulerian rib-web strategies combine machine-learning void filling with non-planar path generation to maintain stable melt pools in arc-energy deposition [80]. Modular software now packages these techniques for industry; Michel et al.

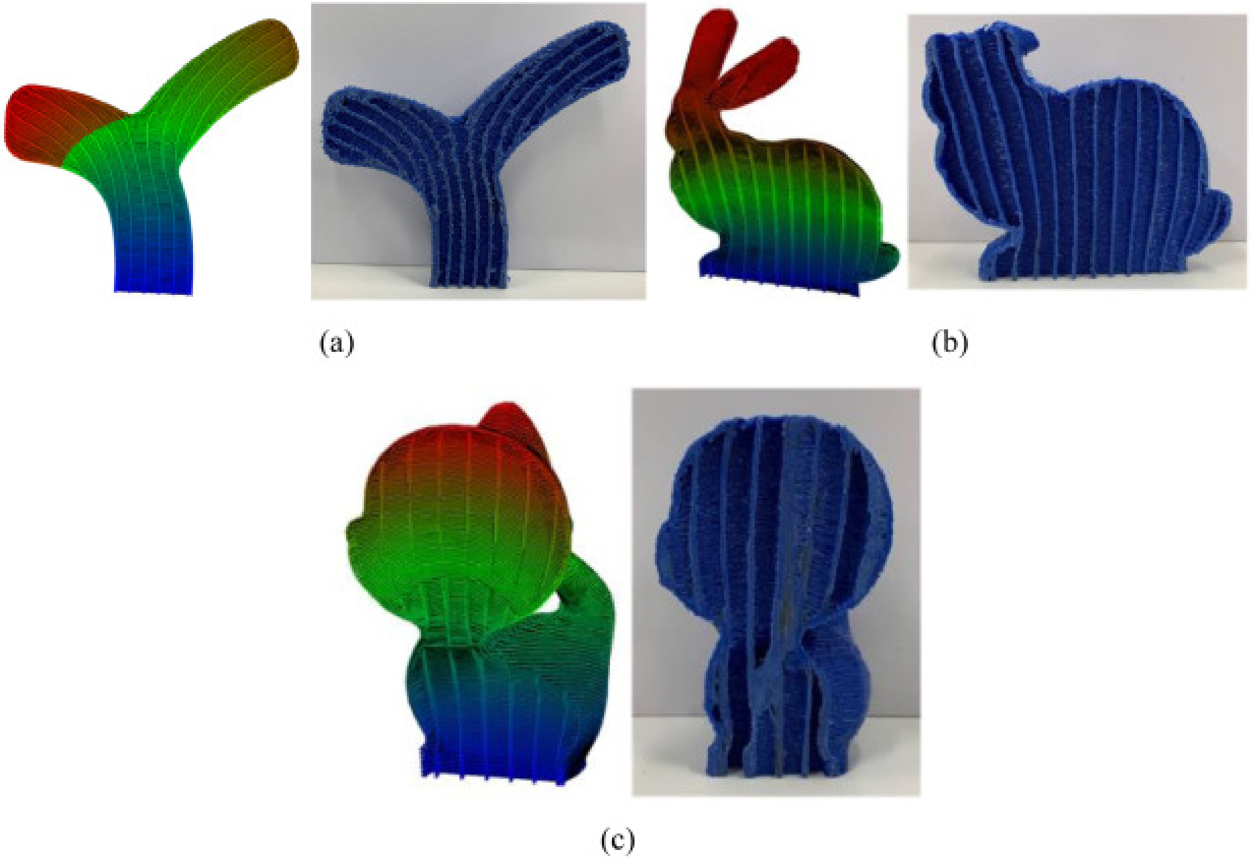


Figure 2.10: Examples of lattice-aware slicing and their respective parts [77]. (a) Tree branch. (b) Stanford bunny. (c) Kitten.

showed that plug-in path-planning modules can switch seamlessly between straight-bead and curved-bead logic within a single Wire + Arc platform [81] (Figure 2.11). However, none of these strategies have found their way into NPAM non-solid infill generation. Therefore, this is an open problem.

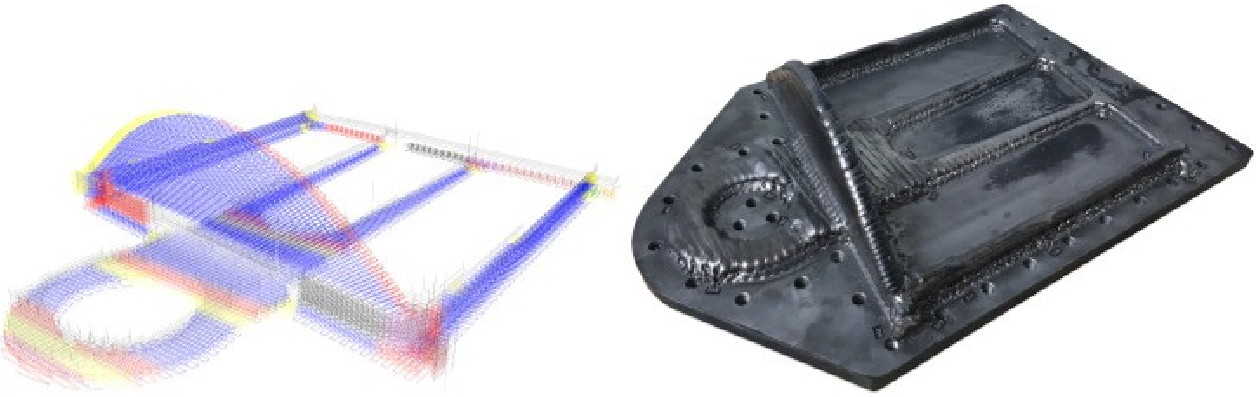


Figure 2.11: Examples of the strategy postulated in [81]. Left-hand side: tool path. Right-hand side: final result.

Applications that exploit the synergy between complex geometry and multi-axis motion are proliferating. Triply periodic minimal-surface (TPMS) structures are now printed as monolithic, support-free heat exchangers whose morphology gradually morphs along the flow path to keep the Nusselt number above 80 [82]. TPMS lattices tuned to trabecular modulus guide bone in-growth in additively manufactured implants and have prompted a surge of biomedical investigations [83]. Functionally graded gyroids push the concept further: Shaikh et al. modulated cell thickness and orientation within a single, curvature-aligned toolpath to create puncture-proof non-pneumatic tyres that pass standard rebound tests [84]. Beyond lattices, Franco-Martínez et al. took advantage non-planar motion to manufacture micro-textured coronary stents whose surface grooves are oriented circumferentially, demonstrating sub-100 [μm] feature fidelity on Ti-6Al-4V tubing [85]. These applications may also benefit from the application of NPAM strategies to align principal stresses or critical geometric features appropriately.

Surveys observe that robotics and NPAM are converging rapidly. Urhal et al. catalogued early robot-assisted extrusion systems and noted that tool-path generation was the bottleneck; recent reviews show that this gap is closing thanks to unified kinematic and thermal models [86]–[89]. Yet design rules remain under-defined: a systematic exploration of wave-guide-type geometries found critical curvature thresholds below which bead sagging re-appears despite optimal orientation planning, underscoring the need for quantitative build-envelop criteria [90]. The ISO/ASTM 52900 vocabulary has not yet incorporated curvature-specific definitions, although eco-efficiency studies suggest that non-planar strategies cut material waste and build energy by double-digit percentages [19].

Collectively, these advances prompted two comprehensive surveys. Nayyeri et al. [91] critiqued planar versus non-planar slicing for fused deposition, emphasizing that curved layers are most beneficial when overhangs exceed 45° or when surface aesthetics trump absolute dimensional accuracy. Lettori et al. [23] extended the discussion to geometry representation, arguing that future robot-based additive systems will need native field or implicit formats so that slicing and stress-aligned deposition can be fused into a single optimization loop.

Looking forward, three research frontiers stand out. First, deposition-aware field generation. The automatic creation of scalar fields whose iso-surfaces satisfy both mechanical and kinematic constraints remains largely heuristic; differentiable slicers such as CurviSlicer's slight-curvature framework hint at a more principled route but still rely on user-tuned parameters [6]. Second, real-time process monitoring must catch up with five-axis kinematics; integrating bead-height sensing with spherical or helical paths is non-trivial because the nozzle seldom points orthogonally to the substrate. Third, hybrid multiprocess platforms, exemplified by Chauvette et al.'s non-planar multiprocess composite printer [17], promise multifunctional parts but demand synchronized tool-change planning and thermomechanical scheduling that current slicers cannot yet deliver.

In short, non-planar additive manufacturing has matured from proof-of-concept algorithms into a toolbox of surface-aligned, support-free, and mechanically aware strategies. What began as an academic curiosity is rapidly becoming an industrial reality, driven by increasingly automated slicers, affordable multi-axis hardware, and a growing body of evidence that curved layers can simultaneously enhance surface quality, structural integrity, and material efficiency.

2.3 Robot-based Additive Manufacturing: equipment for NPAM

To implement the mentioned NPAM strategies, researchers are increasingly integrating multi-axis robotic systems into AM processes. Robot-based additive manufacturing (RbAM) enables non-planar deposition strategies, where a robot arm (or other robotic manipulator) can orient and move manufacturing elements in six or more degrees of freedom (6+ DOF). Over the past few years, significant progress has been made in developing RbAM systems and techniques. This Section aims to synthesize recent developments in robotic manipulators and multi-axis AM machines for NPAM, including integration of sensing/feedback, materials and process innovations, and applications across industries from aerospace to construction to biomedicine.

The dominant platform for NPAM is the industrial robotic arm, typically offering 6-axes of articulation. Recent developments in robotic arms for AM focus on enhancing reach, accuracy, and multi-process integration. For instance, researchers have mounted extruders or welding torches on standard 6-DOF arms (e.g. by KUKA, ABB) to create flexible AM machines capable of building in any direction. Multi-axis arms allow continuously varying tool orientation, which enables conformal deposition, where the nozzle remains normal to a surface, depositing material in curved layers that conform to part geometry. An important benefit of this approach is the ability to build complex overhangs or horizontal features without support material, by orienting each new deposit so that it rests on existing solidified material. Multi-directional AM also reduces the total number of layers for tall sloped surfaces, improving surface smoothness and cutting build time. An illustration of this is the Digital Construction Platform at MIT, a compound robotic arm on a mobile base, which was used to manufacture a dome structure with continuously curved toolpaths, demonstrating autonomous large-scale construction building [92]. An example of this platform is presented in Figure 2.12.

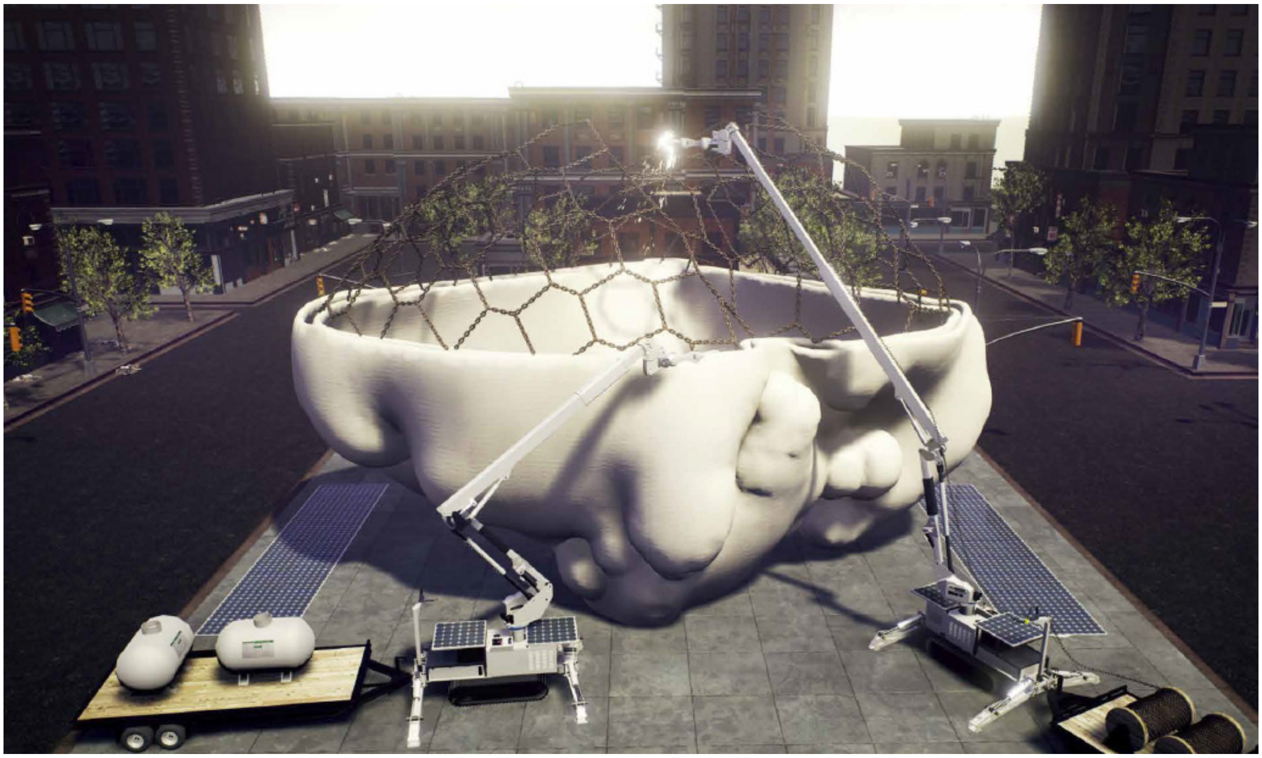
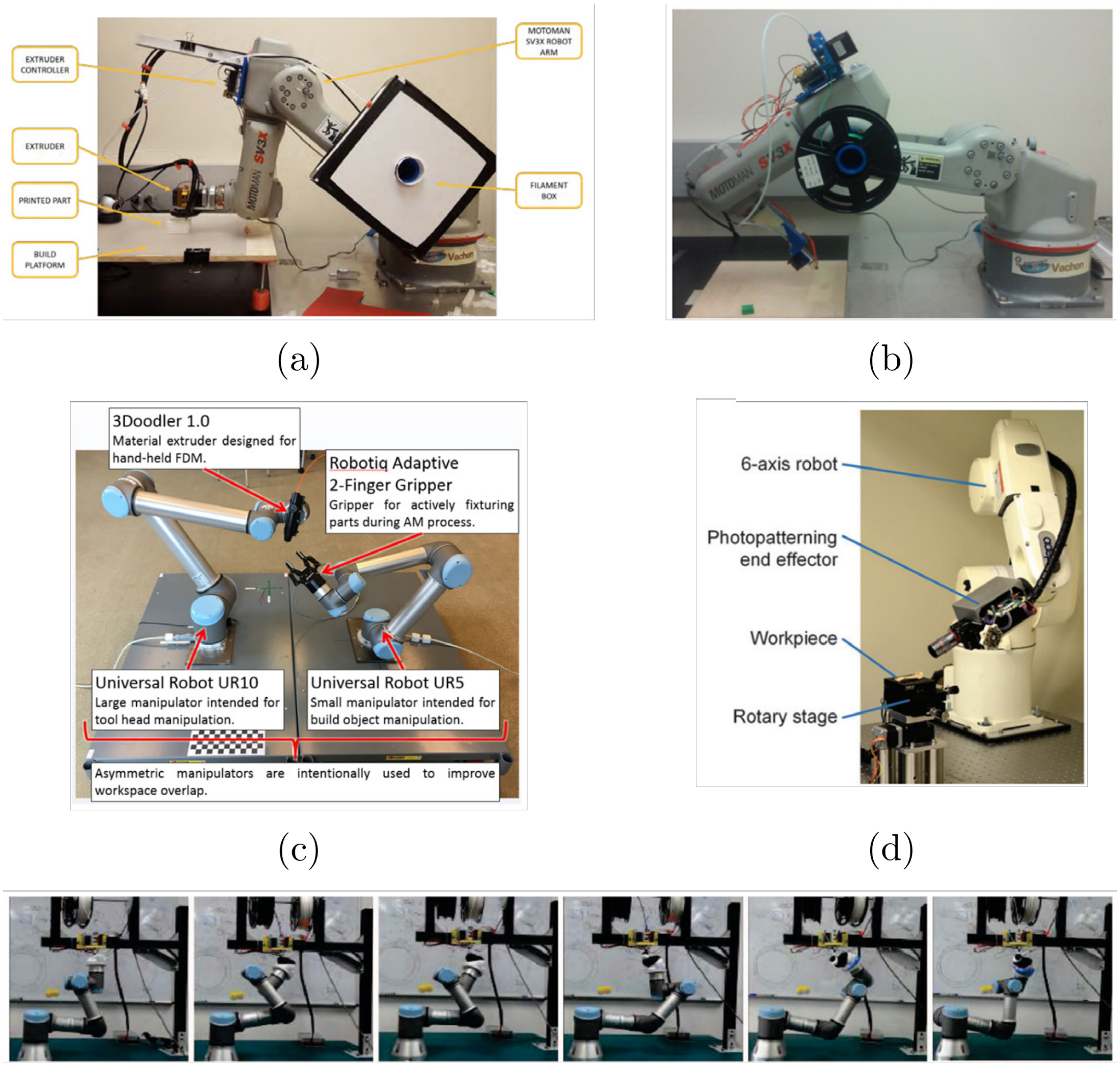


Figure 2.12: Rendering of the Digital Construction Platform [92].

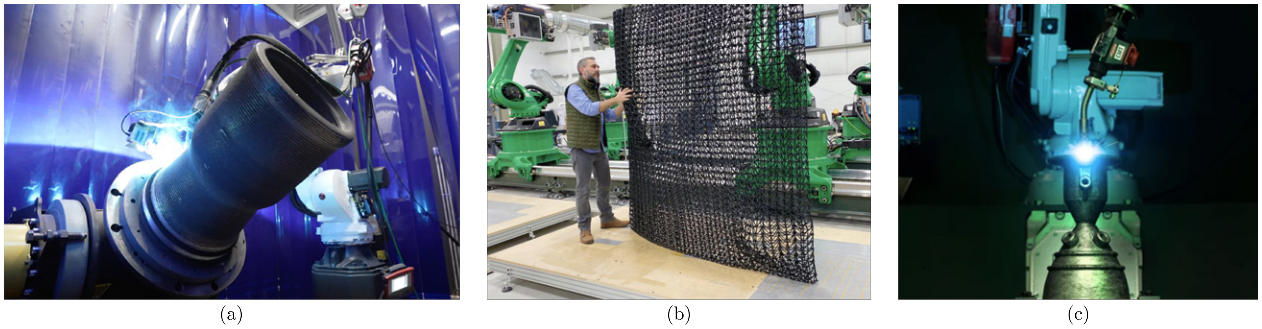
There are many other examples found in the literature utilizing serial industrial robotic arms for AM [93]–[97]. Figure 2.13 presents examples of these robotic systems. Of special interest is the last work [97], covering conformal robotic stereolithography, a hybrid manufacturing platform that mounts a custom, mask-less projection lithography unit on a six-axis industrial robot so it can photo-polymerize resin directly onto curved substrates instead of being limited to flat, layer-by-layer builds. After first creating a mesh of the target free-form surface, the authors generate a triangulated "curved layer" toolpath, rigidly fixture the substrate in the robot work-cell, and then coordinate robot motion with dynamic image projection to print features as small as 10 [μm] on objects ranging from 19 [mm] ball bearings to 210 [mm] plastic spheres. They further show the ability to fabricate small structures (e.g., arch-like features) on those spheres without the supports traditional stereolithography would require. Although accuracy is constrained by robot kinematics, the study argues that improved calibration and path-planning could ultimately enable large, support-free polymer components whose size and geometry are unattainable with conventional planar SLA systems.

On the commercial side, companies like Relativity Space [98] employ multi-axis robotic metal AM machines to fabricate entire rocket stages, utilizing the ability of robots to deposit metal along optimized paths and build large structures with minimal tooling. Likewise, Branch Technology [99] in the U.S. uses a 6-axis arm to additively manufacture architectural wall panels in freeform polymer lattice patterns, and MX3D [100] in the Netherlands commercialize robot-based metal DED systems. Figure 2.14 showcases examples of these systems.



(a) (b) (c) (d) (e)

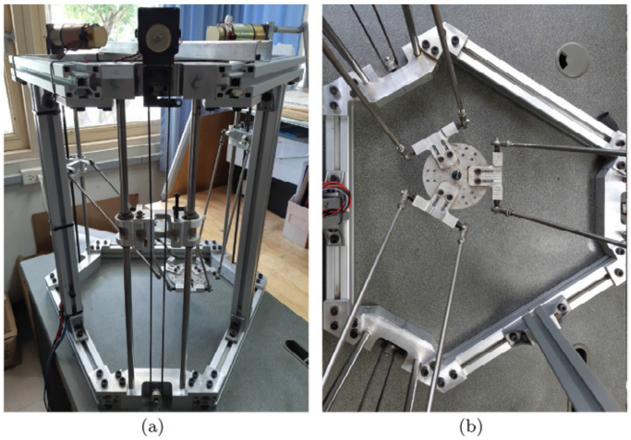
Figure 2.13: Examples of robotic systems for AM. (a) [93]. (b) [94]. (c) [95]. (d) [97]. (e) [96].



(a) (b) (c)

Figure 2.14: Example of commercial robotic systems for NPAM. (a) Relativity Space WAAM manufacturing system [98]. (b) Branch Technology robotic manufacturing system and part [99]. (c) MX3D robotized WAAM manufacturing system [100].

In some scenarios, hybrid systems combine robotic manipulators with additional axes to extend capabilities. An example is a robotic arm mounted on a linear track or gantry, providing effectively a seventh axis for greater workspace coverage (as used in some construction-scale AM machines). Another approach is integrating turntables or rotating build platforms synchronized with the robot, a method that can enable full 6-DOF control of the part orientation relative to the printhead, as demonstrated by multi-axis bioprinters that manufacture on rotating mandrels for tubular tissues. Parallel kinematic machines (e.g. Stewart platforms or delta robots) have also been explored for AM; these can offer high speeds and stiffness. For instance, Carabin and their colleagues [101] developed a linear delta robot with spring-loaded joints to 3D print non-planar layers more energy-efficiently. Figure 2.15 portrays this robotic system.



(a) (b)

Figure 2.15: Linear delta robot developed in [101].

However, using robots for AM introduces complex kinematics and control considerations. Unlike cartesian AM machines, a serial robot's printhead follows a non-linear motion through joint space, raising issues of accuracy and synchronization. Manufacturing quality can suffer if the robot's tool center point deviates from the intended path due to slight calibration errors or dynamic deflections. Researchers have addressed this through both open-loop and closed-loop control strategies.

On the open-loop side, some have implemented preemptive measures: motion primitives are added to the toolpath to improve outcomes, such as slowing down at sharp corners or inserting pauses to allow layer cooling. More recently, there is a strong push toward closed-loop control in robotic AM, utilizing sensors to adapt the process in real time. Vision and laser scanning systems are among the most prevalent feedback modalities. For example, Shojaei Barjuei et al. [102] demonstrated a vision-based compensation loop for robotic bioprinting: a camera captures each deposited bio-ink filament and compares it to the intended path, and the controller then adjusts subsequent toolpaths to correct any error accumulation. This approach reduced layer width variance by 75% and significantly improved trajectory accuracy in printing complex hydrogel constructs. In metal AM with robots (such as WAAM), optical sensors (cameras, infrared thermography) are used to monitor the melt pool and track bead height. Other sensor types include laser line scanners to map the 3D profile of the build, and force/torque sensors to detect contact or clogging in paste extrusion.

For robotic AM, sensing and control are especially critical due to material uncertainties. In one project [103], researchers implemented an extruder-mounted laser scanner to perform adaptive remanufacturing, essentially, if a printed layer was slightly off, the next layer's path is dynamically recalculated to compensate, achieving a smooth final surface. The use of digital twins and advanced sensing is paving the way for fully autonomous robotic AM systems that can "self-correct." However, challenges remain: vision systems can struggle with shiny metal deposits or poor lighting, and real-time data processing for large structures is computationally intensive. Additionally, closed-loop control needs to be tightly synchronized with the robot motion; modern robot controllers are being augmented with external PCs or PLCs to handle the high-frequency feedback loop. Nonetheless, research clearly shows that integrating sensors greatly enhances process reliability. In aerospace-focused RAM, ultrasonic or X-ray inspection modules can be attached to robots to inspect printed wings or engine parts layer-by-layer, ensuring defect-free builds without post-process CT scans. As RbAM technology moves towards industrialization, it will be expected to see more sophisticated sensor fusion (combining visual, thermal, and force feedback) and AI-driven control strategies that continuously learn and optimize the building process in real-time.

Chapter 3

Theoretical basis

The goal of this Chapter is to provide the reader with the necessary concepts to fully understand the present work. The work developed in this PhD Thesis is fundamentally interdisciplinary, and as such, various knowledge fields have been explored. These include mathematics and topology, as well as robotics. The following Subsections will cover each one of these topics.

3.1 Mathematical concepts

3.1.1 Differential and computational geometry

Differential geometry may be defined as the branch of mathematics that studies curves and surfaces, using fundamental concepts from differential calculus (hence the name). This Subsection aims to introduce the reader to mathematical concepts, which will gradually increase in conceptual difficulty along the chapter. Thus, this Subsection will cover basic topics on differential geometry, with a particular focus on the computational aspects of curves and surfaces.

The first topic is parametric curves and their geometric properties. A parametric curve $\mathcal{S}(u)$ is represented by defining a parameter u that varies over an interval of real numbers, called the parameter domain. Each coordinate of a point on the curve is the expressed as a function of u, by the following expression:

$$\mathcal{S}(u) = \{x(u), y(u), z(u)\} \tag{3.1}$$

These curves are characterized by several fundamental properties:

- **Continuity**. A curve is said to be continuous if its generating functions $x(u)$, $y(u)$, $z(u)$ are continuous over the parameter domain.
- **Differentiability**. A curve is differentiable over the parameter domain if its generating functions are differentiable. This implies the derivatives of the generating functions ($\dot{x}(u)$, $\dot{y}(u)$, $\dot{z}(u)$) are continuous over the parameter domain.

With these concepts, it is possible to define the continuity class n, otherwise expressed as C^n for any given curve. Some examples include:

- **$\mathbf{C^0}$ continuity**. The curve is continuous across the parameter domain.
- **$\mathbf{C^1}$ continuity**. The curve's first derivative is continuous across the parameter domain.
- **$\mathbf{C^\infty}$ continuity**. The curve is completely differentiable, no matter how many times the derivative is applied.

These classes may seem uninteresting in the context of single segment curves, as they are usually C^∞ continuous. However, if a curve is built by means of the concatenation of more than one curve segments, these properties become crucial, especially at the knots (i.e., the junction points). The present Thesis utilizes curve concatenations, also called composite curves or splines, which will be discussed later.

Other useful differential geometry concept is the tangent vector. At any point on the parametric curve $\mathcal{S}(u)$, the tangent vector $\mathbf{t}(u)$ is defined by:

$$\mathbf{t}(u) = \nabla\mathcal{S}(u) = \{\dot{x}(u), \dot{y}(u), \dot{z}(u)\} \tag{3.2}$$

This vector indicates the local direction of the curve at point u in the parameter domain. Oftentimes, this vector is normalized to obtain a unit vector whose magnitude is one.

The next useful concept within differential geometry is curvature, which indicates the rate of change in direction of the curve. This curvature κ is defined as follows:

$$\kappa(u) = \frac{\|\nabla\mathcal{S}(u) \times \nabla^2\mathcal{S}(u)\|}{\|\nabla\mathcal{S}(u)\|^3} \tag{3.3}$$

Where:

$$\nabla^2\mathcal{S}(u) = \{\ddot{x}(u), \ddot{y}(u), \ddot{z}(u)\} \tag{3.4}$$

In addition to the curvature κ, it is possible to define the radius of curvature ρ_κ as the curvature's inverse.

The last differential geometry to leverage in this work is the arc length formula. Given a parameter subdomain $u \in [A, B]$, where $A, B \in \mathbb{R}$, the length of the curve between $\mathcal{S}(A)$ and $\mathcal{S}(B)$ is calculated as:

$$L = \int_A^B \|\nabla\mathcal{S}(u)\| \, du \tag{3.5}$$

All these concepts are useful in the context of Euclidean geometry, which comprehends most engineering applications. However, Euclidean geometry is constrained by the fact that the Euclidean space has null curvature, and therefore, unsuitable for NPAM contexts. Therefore, the following Subsections, regarding topology and Riemannian geometry, will expand these differential geometry concepts onto non-Euclidean spaces.

Computational geometry is the branch of computer science which relates algorithms with geometry. The main areas of application of this field include the development of computer graphics, CAD/CAM software, computer vision, among others. These applications are directly related to the concepts developed in this Thesis. Following the ideas regarding parametric curves, it is necessary to actually implement them from a computational perspective. In this context, Bézier curves are a good fit for the methods presented in this work. Bézier splines are highly suitable for modeling parametric curves due to their flexibility. They can be concatenated while ensuring the desired level of continuity–be it C^0, C^1, and so on–at the junction points. They are also locally modifiable: adjusting control points in one region of the curve does not affect other regions. Lastly, additional control points can be easily inserted if greater detail is required in a specific segment of the curve.

Mathematically, a Bézier curve $\mathcal{S}$ is a type of parametric curve, usually characterized by its degree n. Additionally, a Bézier curve is defined by a control polygon $\mathbf{P}$, comprised of $n+1$ row vectors $P_0, P_1, \cdots, P_n$, which are points defined in the plane. This results in the following expression:

$$\mathcal{S}(t) = \sum_{i=0}^{n} b_{i,n}(t) P_i \tag{3.6}$$

Where $t \in [0,1]$ is the curve parameter, and $b_{i,n}(t)$ are the Bernstein polynomials:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{3.7}$$

Where:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{3.8}$$

In the previous formulation, the Bernstein polynomials are the equivalent to shape functions in Finite Element contexts, where these functions serve as means of interpolation between nodes. However, to enhance the curve's computational applicability, it is possible to rewrite Equation 3.6 into a matrix form by rearranging the polynomial coefficients into $\mathbf{B}$, the Bézier basis matrix:

$$\mathcal{S}(t) = \mathbf{tBP} \tag{3.9}$$

$$\mathbf{t} = \begin{pmatrix} t^n & t^{n-1} & \cdots & t & 1 \end{pmatrix} \tag{3.10}$$

$$\mathbf{B} = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix} \tag{3.11}$$

$$\mathbf{P} = \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{pmatrix} \tag{3.12}$$

Two examples of the Bézier basis matrix include the cubic Bézier basis matrix ($\mathbf{B_3}$) and the degree-six Bézier basis matrix ($\mathbf{B_6}$), as these two will be the objects of study in the present Thesis. These basis matrices take the following forms:

$$\mathbf{B_3} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B_6} = \begin{pmatrix} 1 & -6 & 15 & -20 & 15 & -6 & 1 \\ -6 & 30 & -60 & 60 & -30 & 6 & 0 \\ 15 & -60 & 90 & -60 & 15 & 0 & 0 \\ -20 & 60 & -60 & 20 & 0 & 0 & 0 \\ 15 & -30 & 15 & 0 & 0 & 0 & 0 \\ -6 & 6 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.13}$$

These Bézier basis matrices enable key properties, which are useful in the context of computational geometry:

- **Fast calculation of Bernstein functions.** The product **tB** is in fact, the curve basis functions, otherwise known as shape functions. This can be seen in Figure 3.1.

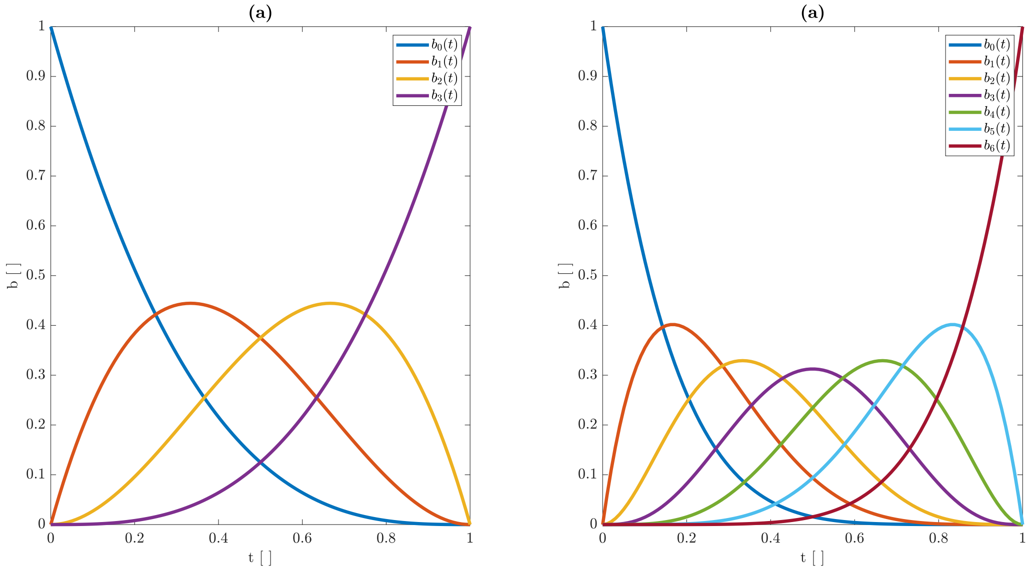


Figure 3.1: (a) Basis functions for $\mathbf{B_3}$. (b) Basis functions for $\mathbf{B_6}$.

- **Fast calculation of the derivatives.** As both **B** and **P** remain constant in the matrix form, the curve derivatives are calculated as follows:

$$\dot{\mathcal{S}}(t) = \frac{d}{dt}\mathcal{S} = (\frac{d}{dt}\mathbf{t})\mathbf{BP} \tag{3.14}$$

$$\ddot{\mathcal{S}}(t) = \frac{d^2}{dt^2}\mathcal{S} = (\frac{d^2}{dt^2}\mathbf{t})\mathbf{BP} \tag{3.15}$$

$$\vdots$$

The matrix form allows for easy analytical and numerical calculation of the derivatives. These derivatives are of critical importance to all methods presented in this Thesis, but especially for the concatenation of curves, which will be explained shortly.

A noteworthy characteristic of a Bézier curve is its complexity. A curve is complex if it intersects with itself, forming nodes (if the curve has two t solutions for a given point in space, i.e., non-bijective) or cusps (where the directional derivative changes sign and both parametric derivatives are zero). Examples of a self-intersection and a cusp are provided in Figure 3.2.

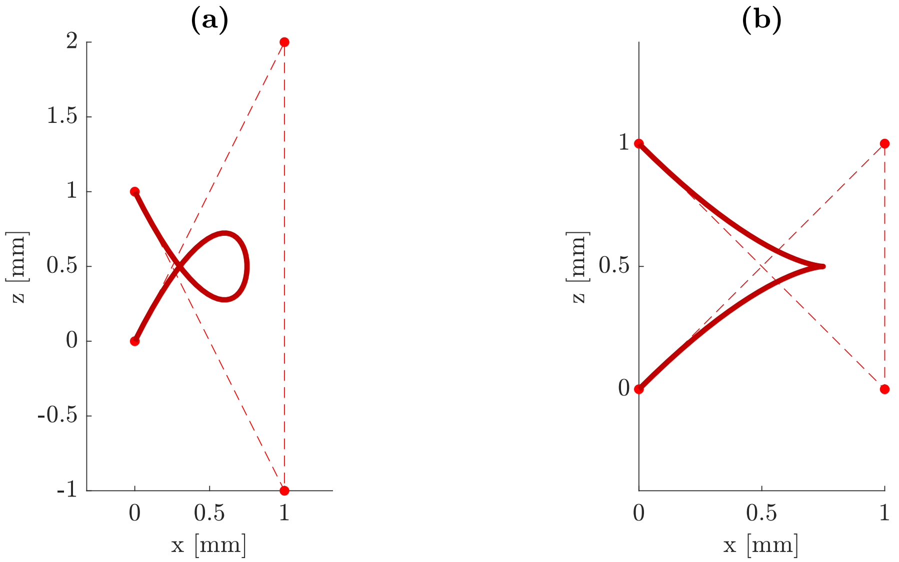


Figure 3.2: (a) Example of a self-intersection. (b) Example of a cusp.

A straightforward method of determining if a curve is complex is by examining the control polygon. If the control polygon is complex (it self-intersects), then the Bézier curve will be complex as well. This is noticeable in the example previously represented in Figure 3.2. Most of the methods developed in this Thesis require for the curves to be bijective, therefore rendering complex curves ineffective in the general case.

Bézier curves can be concatenated, forming a Bézier composite curve, or otherwise called, a Bézier spline. A Bézier spline expands the concept of the curve parameter t, previously constrained within the interval $[0, 1]$, onto $u \in [0, m]$, where m is the total number of concatenated curves (i.e., curve segments). For each of the curve segments j, a different control polygon $\mathbf{P_j}$ is defined. The main point of study of Bézier splines is the curve behavior in its junctions, also known as knots. Whereas a single segment Bézier curve is parametrically C^{∞} continuous, a Bézier spline may not be even C^0, depending on the splines's behavior in the knots. It is possible, however, to establish different continuity constraints to enforce a continuity class throughout the parameter domain. For instance, if a spline (composed by segments $\mathcal{S}_1(u), u \in [a, b]$ and $\mathcal{S}_2(u), u \in [b, c]$) is desired to be C^0 continuous, the following condition shall be applied:

$$\mathcal{S}_1(u = b) = \mathcal{S}_2(u = b) \tag{3.16}$$

This relationship implies that the end point of segment $\mathcal{S}_1$ is the beginning point of $\mathcal{S}_2$. More generally, it is possible to enforce any kind of continuity class C^I across a spline through the following forumulae, and visually expressed in Figure 3.3.

$$\frac{d^i}{du^i}\mathcal{S}_j(u = u_{knot}) = \frac{d^i}{du^i}\mathcal{S}_k(u = u_{knot}) \ \forall k = j + 1, i \in \mathbb{Z}, i \in [0, I] \tag{3.17}$$

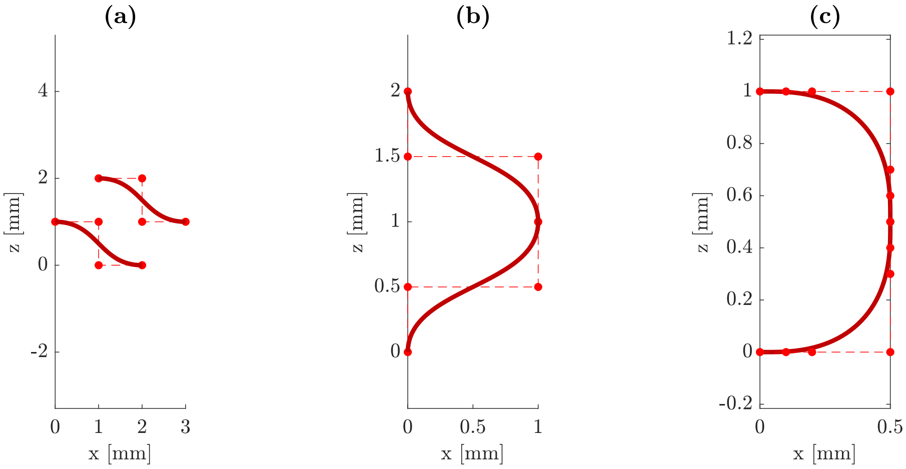


Figure 3.3: (a) Example of a non-continuous spline. (b) Example of a cubic C^1-continuous spline. (c) Example of a degree-six C^3-continuous spline.

The introduction of constraints limits the design versatility of Bézier splines, where a degree-n spline can only accommodate for $n-1$ constraints. If n constraints are put in place, then the spline becomes a degenerate form of the expansion of a regular curve segment, from $t \in [0,1]$ onto $t \in [0,m]$, allowing for no design space when $t \geq 1$. This is the reason degree-six splines are introduced in this work, and will be explained in the following Chapter.

3.1.2 Topology and embedded maps

Topology is the branch of mathematics dedicated to the study of geometrical objects under continuous deformations. In this sense, a continuous deformation is translating an object, rotating, reflecting, stretching, twisting, or bending it. Examples of non-continuous deformations include cutting, punching holes, or gluing objects together. With this ruleset clear, a topological space is the set of an object and the formal definition of allowed continuous transformations, which may be called homeomorphisms.

A topological space lives inside an ambient space of equal or higher dimension. For convenience, in this work, the ambient space is the Cartesian and Euclidean space $\{a\} \equiv \mathbb{R}^3$, also known as the common reality in which we usually live. In this ambient space, it is possible to define and combine several types of topological spaces, such as:

- **A line**. A line L is a map of an $\mathbb{R}$ linear space onto a part of the ambient space $\Omega \in \{a\} \equiv \mathbb{R}^3$ ($L : \mathbb{R} \to \Omega \in \{a\}$).
- **A plane**. A plane π is a combination of two lines, creating an object which is traversable in two different directions. This is expressed as $\pi : \mathbb{R}^2 \to \Omega^2 \in \{a\}$.
- **A three-dimensional space**. Adding a third line to the previous plane creates another object $\{v\}$, with $\{v\} : \mathbb{R}^3 \to \Omega^3 \in \{a\}$.
- **A circumference**. If instead of taking a linear space $\mathbb{R}$ to map onto the ambient space we select an angular space S^1, then the result is a circumference $C : [0, 2\pi] \to \Omega \in \{a\}$.

- **A spherical surface**. This is the combination of two S^1 spaces, resulting in an S^2 space. This space is traversable in two angular coordinates, for instance, latitude and longitude.

There is a common denominator for all topological spaces above listed: there is a mapping operation happening between them and a region in the ambient space, which is a kind of homeomorphism (i.e., an allowed topological operation, in this case, the homeomorphism of a domain into its image through an injective function). This idea is the underlying principle of embedded maps. Among all previous examples, the most common use of an embedded map is the Earth's coordinates, latitude and longitude. This is one of the infinitely many maps that can be embedded on the Earth globe. A mapping is valid as long as the homeomorphism is injective (i.e., each point on the cartographic map, the domain, has a corresponding point on the Earth globe proper, the image). This is the reason for the many Earth projections available in cartographic maps.

As previously outlined, a homeomorphism is a transformation (f) between two topological spaces ($f : X \to Y$), that fulfills certain conditions:

- f is bijective (all points of X are pairwise countable with all points of Y, and there is a unique transformation $Y = f(X)$ for all points of X).
- f is continuous.
- f^{-1} (the homeomorphism inverse function) is continuous.

All these conditions imply f is an open mapping, akin to the cylindrical homeomorphism $\{x, y, z\} \rightleftharpoons \{r, \theta, z\}$ (also known as the cylindrical coordinate system). An open mapping means the homeomorphism is bijectively defined between open sets, such as $\mathbb{R}^3 \rightleftharpoons \mathbb{R}^3$ if θ is allowed to extend beyond 2π, wrapping around.

This general definition of a homeomorphism is too strict for the procedures outlined in this work. For the purposes of NPAM, a closed mapping is sufficient. A closed mapping fundamentally breaks the third axiom, in the sense that f^{-1} may not be generally continuous, or even defined across an open image. Such is the case of the embedded map methodology outlined in Section 4.1, where coordinates are constrained to be non-negative, or constrained to a closed domain $\omega \in \mathbb{R}$. Despite these compact subsets $\omega \in \mathbb{R}$, a local homeomorphism is considered sufficient for the methods described. This forgoes the need of a globally bijective homeomorphism, as long as the parts are contained in regions of local homeomorphism.

Topological spaces are many, and they may include objects with holes (such as a toric surface, representing a $S^1 \times S^1$ topological space), or non-orientable surfaces (such as Klein bottles). In this work, the scope is limited to closed, orientable, and compact topological spaces. Such spaces are called manifolds.

Formally, a manifold $\mathcal{M}$ is an n-dimensional topological space which is locally homeomorphic to the n-dimensional Euclidean space $\mathbb{E}^n$. This means a manifold is an n-dimensional object which, in a sufficiently small region, all five Euclid's postulates are true. Of these, the most interesting one is the fifth, the parallel postulate. This postulate establishes the uniqueness of a parallel line to another, given another non-collinear point in space. This postulate enables essential mathematical theorems (e.g., the Pythagorean theorem), but is generally false outside of an ε-close vicinity of a point in a manifold.

The falseness of Euclid's fifth postulate in non-local regions of a manifold gave birth to a whole new field of mathematics, initially discovered but discarded by Carl Friedrich Gauss, and formalized by his student, Bernhard Riemann. This field is known as Riemannian geometry, and discusses particular cases of non-Euclidean geometry. This topic will be extensively covered in the next Subsection. Taking these concepts into account, 2-manifolds can be split in three main categories:

- **The Euclidean manifold:** This is the 2-dimensional space everyone is used to, also known as the 2D Cartesian space. This manifold features null ambient curvature across any embedded map, and is homeomorphic to $\mathbb{R}^2$. This manifold fulfills Euclid's fifth postulate locally and globally.
- **Gaussian manifolds:** Given K, the Gaussian curvature, which is the product of the principal curvatures of a manifold $K = \kappa_1\kappa_2$, Gaussian manifolds are defined as manifolds whose Gaussian curvature is constant. There are three main types of Gaussian manifolds, spheres (with constant and positive K), hyperboles (with constant and negative K) and the Euclidean space (with null K, as discussed previously). The first two types are generally non-developable, which is the reason no 100% accurate Earth cartographic map can be made, and give birth to elliptic and hyperbolic geometries, respectively.
- **Riemannian manifolds:** The most general case of a manifold, which includes the previous two categories, and all 2-manifolds with non-constant differentiable Gaussian curvature. These objects are the main topic studied in this Thesis. Per definition, a Riemannian manifold comes equipped with both a set of locally homeomorphic embedded maps, as well as a metric. These two concepts will be fundamental in the development of the methods explained in Chapters 4 and 5, and will be explained in the following Subsection.

3.1.3 Riemannian geometry

Riemannian geometry is the field of mathematics arising from the need of generalizing the classic Euclidean concepts of length, distance, angle, surface, or volume onto non-Euclidean spaces. As such, this Subsection will cover the fundamental notions of general non-Euclidean geometry, and will provide powerful knowledge tools to work with Riemannian manifolds.

First, it is necessary to establish the scope of the tools that will be developed throughout the Subsection. Riemannian geometry is the study of differentiable manifolds (appropriately called Riemannian manifolds) through differential geometry. Riemannian geometry provides an intrinsic way of studying manifolds (i.e., without the need for an ambient space), and becomes useful when creating Euclidean-like concepts on curved spaces like manifold embedded maps.

Riemannian geometry is fundamentally based upon the definition of the Riemannian metric, otherwise known as the metric tensor. The metric tensor provides all Euclidean-like information there is to know in the context of a manifold embedded map, irrespective of its rank. Given a 2-manifold in an ambient Cartesian space $\mathbb{R}^3$, with an embedded map $\{\varphi\} = \{\mathbf{e}^1, \mathbf{e}^2\}$ (where

e^i indicates the map basis vectors expressed in the ambient space coordinates), the metric tensor **g** is defined as:

$$\mathbf{g} = \mathbf{e}^i \otimes \mathbf{e}^j \tag{3.18}$$

Expressed in the Einstein notation, where i and j are indices spanning the embedded map coordinates. In matrix form, for the above $\{\varphi\}$ embedded map, this results in:

$$\mathbf{g} = \begin{pmatrix} \mathbf{e}^1 \cdot \mathbf{e}^1 & \mathbf{e}^1 \cdot \mathbf{e}^2 \\ \mathbf{e}^2 \cdot \mathbf{e}^1 & \mathbf{e}^2 \cdot \mathbf{e}^2 \end{pmatrix} \tag{3.19}$$

This definition is significantly powerful in the context of smooth manifolds, since the embedded map provides a locally homeomorphism between the manifold proper and a region of the ambient space. Therefore, performing the inner product between the basis vectors actually captures all geometric aspects of the manifold in an intrinsic way. The metric tensor has several properties:

- **g** is symmetric. This is a natural property arising from the inner product inside the tensor components.
- **g** is semi-definite positive. This implies the possibility for the metric tensor to be non-invertible.
- If the embedded map basis vectors are orthogonal (or more generally, the vectors conform a Hilbert space), **g** is diagonal. This property is to be accounted for when constructing embedded maps, as it simplifies matrix inversion and clarifies the matrix invertibility.

Let us provide a simple example: calculate the metric tensor **g** of an Euclidean manifold $\mathcal{E}$ with an embedded map $\{\varphi\} = \{x, y\}$ (i.e., the Cartesian map), in a Cartesian ambient space $\mathbb{R}^3$. This embedded map is a proper homeomorphism of the Cartesian space $\mathbb{R}^2$, and it is represented in Figure 3.4, which is none other than the regular xy coordinate system.

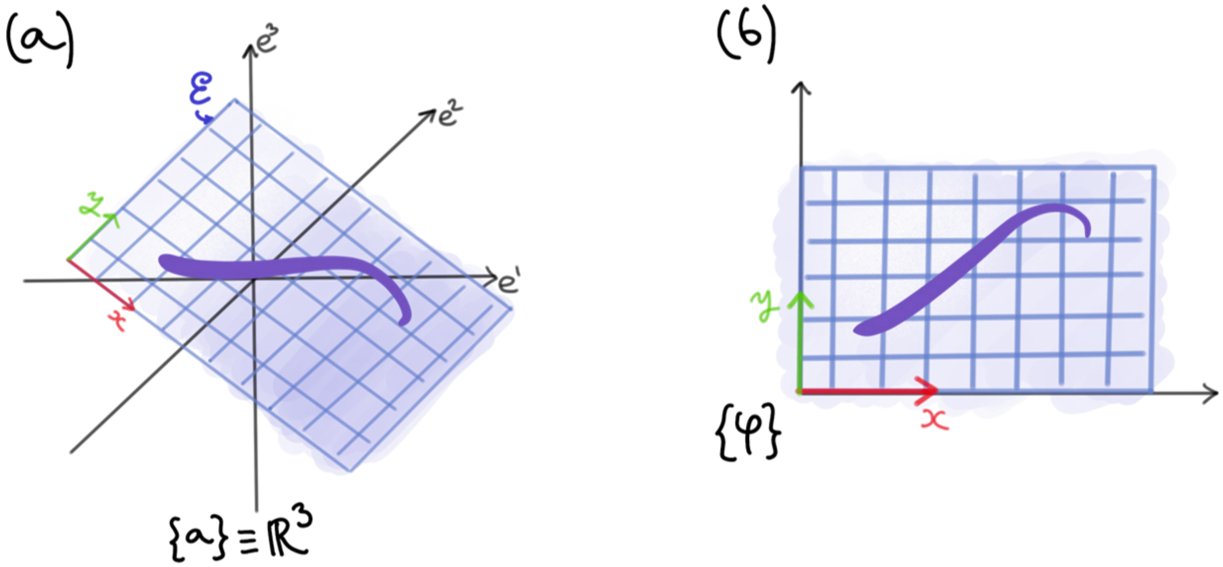


Figure 3.4: (a) An Euclidean manifold in an ambient space $\mathbb{R}^3$. (b) The proposed embedded map coordinates.

This embedded map features orthogonal basis vectors, therefore, it is possible to assume the off-diagonal components are equal to zero. Regarding the diagonal, non-zero components, their calculation goes as follows:

$$g_{xx} = \mathbf{e^x} \cdot \mathbf{e^x} = \mathbf{i} \cdot \mathbf{i} = 1 \tag{3.20}$$

$$g_{yy} = \mathbf{e^y} \cdot \mathbf{e^y} = \mathbf{j} \cdot \mathbf{j} = 1 \tag{3.21}$$

$$\mathbf{g} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{1} \tag{3.22}$$

The metric tensor $\mathbf{g}$ for all Euclidean manifolds is equal to identity. This result, apparently inoffensive at first glance, is the cornerstone of Euclidean geometry, as it enables all five Euclid's postulates. Let us compare it with a different, more complex example: calculate the metric tensor $\mathbf{g}$ of a cylindrical Gaussian manifold $\mathcal{C}$ with an embedded map $\{\xi\} = \{\mathbf{r}, \theta, \mathbf{z}\}$ (i.e., the cylindrical coordinate system), in a Cartesian ambient space $\mathbb{R}^3$. This embedded map is represented in Figure 3.5.

In this case, the metric tensor $\mathbf{g}$ is calculated as follows (owing to the orthogonal property of the embedded map):

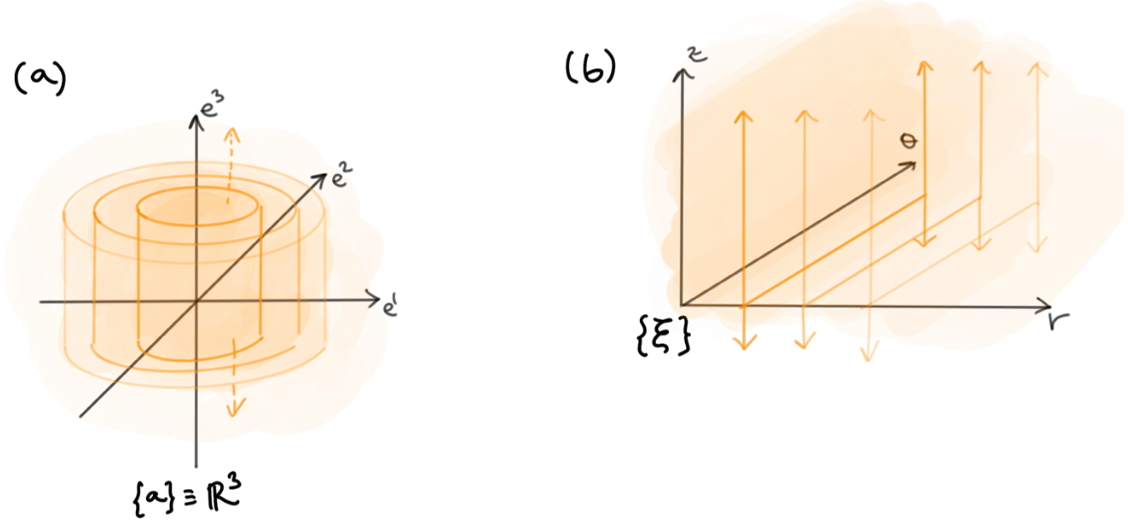


Figure 3.5: (a) A cylindrical Gaussian manifold in an ambient space $\mathbb{R}^3$. (b) The proposed embedded map coordinates.

$$g_{rr} = \mathbf{e^r} \cdot \mathbf{e^r} = 1 \tag{3.23}$$

$$g_{\theta\theta} = \mathbf{e}^{\theta} \cdot \mathbf{e}^{\theta} = r^2 \tag{3.24}$$

$$g_{zz} = \mathbf{e^z} \cdot \mathbf{e^z} = 1 \tag{3.25}$$

$$\mathbf{g} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.26}$$

Contrary to the previous case, the metric tensor is no longer the identity matrix. An interesting result is that the metric tensor determinant is, in fact, the transformation Jacobian squared. This provides an idea that the metric tensor also implicitly accounts for manifold curvature. This effect will become more apparent when the Levi-Civita connection is discussed later in the document.

Let us discuss distances within a manifold. Distance is the qualitative physical span between two points. Clasically, in analytical Euclidean geometry, distance has been intimately related with the length of the segment joining two points. This length is usually calculated through the Pythagorean theorem. However, in Riemannian geometry, this information is stored in the metric tensor through the following formula:

$$ds^2 = g_{ij} dx^i dx^j \tag{3.27}$$

$$s^2 = \int_{\mathcal{G}} g_{ij} dx^i dx^j \tag{3.28}$$

Where ds is the differential line element, s is the length along a curve $\mathcal{G}$, and the indices i, j span the embedded map coordinates. In an Euclidean manifold (akin to the one presented in

Figure 3.4), the distance between points A ($\{x_A, y_A\}$) and B ($\{x_B, y_B\}$) can be analytically solved as:

$$ds^2 = g_{xx}dx^2 + 2g_{xy}dxdy + g_{yy}dy^2 \tag{3.29}$$

$$ds^2 = dx^2 + dy^2 \tag{3.30}$$

$$s = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \tag{3.31}$$

This is, unsurprisingly, the Pythagorean theorem. Let us move onto a more complex example, the cylindrical manifold presented in Figure 3.5. In this case, the distance between points A ($\{r_A, \theta_A, z_A\}$) and B ($\{r_B, \theta_B, z_B\}$) is expressed as follows:

$$\begin{aligned} ds^2 = g_{rr}dr^2 + g_{\theta\theta}d\theta^2 + g_{zz}dz^2 + \\ +2g_{\theta z}d\theta dz + 2g_{rz}drdz + 2g_{r\theta}drd\theta \end{aligned} \tag{3.32}$$

$$ds^2 = dr^2 + r^2d\theta^2 + dz^2 \tag{3.33}$$

This is harder to solve analytically, but it makes sense: distances in a given θ plane are solved by the Pythagorean theorem ($s = \sqrt{(r_B - r_A)^2 + (z_B - z_A)^2}$), and distances between points within the same r and z coordinates are solved through the arc length formula ($s = r(\theta_B - \theta_A)$, note the factor of r arising from $g_{\theta\theta}$).

Let us now discuss area and volume. These concepts are the qualitatively measures of a region's size, be it area for 2D regions or volume for 3D regions. In the context of manifolds, the n-dimensional size (dV) is related to the metric tensor determinant, through the following formula:

$$dV = \sqrt{\det \mathbf{g}}\ dx^1 \cdots dx^n \tag{3.34}$$

Particularly for the Euclidean case, this results in:

$$dA = \sqrt{g_{xx}g_{yy} - 2g_{xy}}\ dxdy \tag{3.35}$$

$$dA = dxdy \tag{3.36}$$

$$A = (x_B - x_A)(y_B - y_A) \tag{3.37}$$

Which is none other than the rectangle's base times its height. Regarding the cylindrical manifold, volume is defined as:

$$dV = \sqrt{g_{rr}g_{\theta\theta}g_{zz}}\ drd\theta dz \tag{3.38}$$

$$dV = r\ drd\theta dz \tag{3.39}$$

Once again, this is a classic result from differential geometry, where a volume is the integral over a field of the differential components of the coordinate system, times the transformation Jacobian matrix ($dV = J\ dx^1 dx^2 dx^3$).

In summary, all these concepts may appear too convoluted in the context of Euclidean or Gaussian manifolds, however, this is the only way to define geometry in Riemannian manifolds. With this background, the following Subsection will cover the topic of segment length calculation on Riemannian manifolds, which is central for some of the methods proposed in this Thesis.

3.1.4 Geodesics on a Riemannian manifold

The main focus of this Subsection is the calculation of the distance between two points along any given manifold. Let us start framing the problem in mathematical terms. Let an n-manifold $\mathcal{M}$, with an embedded map $\{\varphi\} = \{\mathbf{e}^1, \cdots, \mathbf{e}^n\}$. On top of this manifold lie two points, $A = \{a^1, \cdots, a^n\}$ and $B = \{b^1, \cdots, b^n\}$. Calculate the distance along the manifold between points A and B (Figure 3.6). This is known as the geodesic problem.

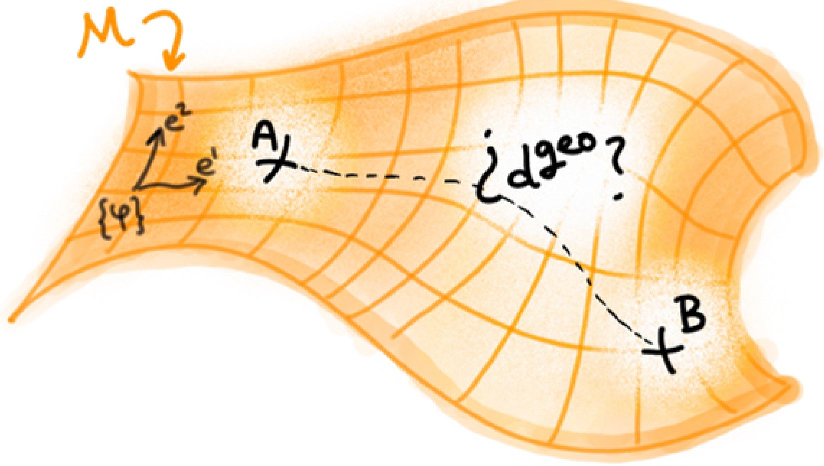


Figure 3.6: The geodesic problem of calculating the distance between two points.

Before tackling the geodesic problem, it is important to discuss how to "derivate" a manifold. This derivative concept greatly differs from the basic notions of multivariable calculus, as all those concepts are clasically based on Euclidean concepts. In order to establish the concept of derivative, it is necessary to define what an affine connection is.

An affine connection describes the transformation between two tangent spaces of a manifold in an smooth way. A tangent space consists of an embedded map basis, located around a point P (i.e., the Euclidean vicinity of P). Therefore, an affine connection indicates how to transform a tangent space into another nearby tangent space. This is remarkably similar to the concept of a derivative in calculus, where the operation defines how to provide a connection between two nearby points by means of the function's tangents. In a sense, the function's derivative serves as a means of parallel transport along the function, and so does an affine connection, providing parallel transport along a manifold. Figure 3.7 graphically exemplifies these concepts.

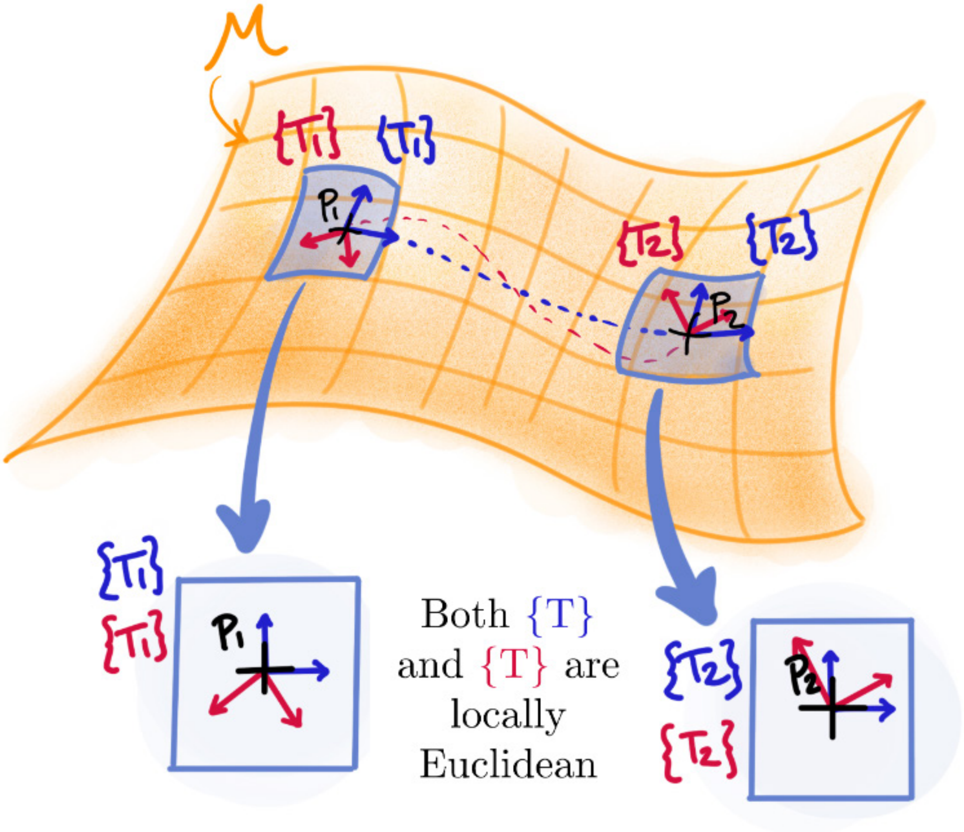


Figure 3.7: A visual representation and comparison of the affine connection and the linear derivative.

Affine connections are many in the context of a manifold, as it is possible to define infinitely many embedded maps, and transformations among them. However, the Levi-Civita connection stands out among the rest, as it is the only affine connection that preserves the metric. All other affine connection implies a non-identity deformation of the embedded map along it, which results in map torsion. The Levi-Civita connection has profound implications in the understanding of the "derivative" concept, and applies covariate derivation in the context of manifolds.

An affine connection ∇_X is a Levi-Civita connection if:

- $\mathbf{g}$ remains equal for any given embedded map $\{\varphi\} = \{\mathbf{e}^1, \cdots, \mathbf{e}^n\}$, yielding $\nabla \mathbf{g} = \mathbf{0}$.
- ∇ is torsion free. This means that, given vector fields $\mathbf{e}^a$, $\mathbf{e}^b$, the following operation $\nabla_{\mathbf{e}^a}\mathbf{e}^b - \nabla_{\mathbf{e}^b}\mathbf{e}^a = \mathbf{e}^a(\mathbf{e}^b) - \mathbf{e}^b(\mathbf{e}^a)$ is true for any two vectors $\mathbf{e}^a$ and $\mathbf{e}^b$ of $\{\varphi\}$, making the connection symmetric.

The Levi-Civita connection on a manifold is expressed as:

$$\nabla_i \mathbf{e}_j = \Gamma^k_{ij} \mathbf{e}_k \tag{3.40}$$

Where Γ^k_{ij} are the Christoffel symbols of the second kind of the given manifold. The Christoffel symbols contain the information of the rate of change of the embedded map vector basis (and therefore, of the metric tensor) when moving along the manifold. The Christoffel symbols are calculated as:

$$\Gamma^i_{jk} = \frac{1}{2} (\mathbf{g}^{-1})_{il} (\frac{\partial}{\partial x^j} g_{lk} + \frac{\partial}{\partial x^k} g_{lj} - \frac{\partial}{\partial x^l} g_{jk}) \tag{3.41}$$

With these concepts in place, it is now possible to define parallel transport along a manifold, and therefore, geodesics. A geodesic curve ($\mathcal{G}$) is the shortest path between two points of a manifold, and may be implicitly expressed as the following Levi-Civita connection:

$$\nabla_{\dot{\mathcal{G}}}\ \dot{\mathcal{G}} = 0 \tag{3.42}$$

This equality implies that the parallel transport of a curve's tangent vector along the curve lets the vector remain constant. In terms of Christoffel symbols, on a 2-manifold equipped with an embedded map $\{\varphi\} = \{\mathbf{e}^x, \mathbf{e}^y\}$, this differential equation may be expressed as follows:

$$\begin{cases} \ddot{x} = -\Gamma^x_{xx} \dot{x}\dot{x} - \Gamma^x_{yy} \dot{y}\dot{y} - \Gamma^x_{xy} \dot{x}\dot{y} \\ \ddot{y} = -\Gamma^y_{xx} \dot{x}\dot{x} - \Gamma^y_{yy} \dot{y}\dot{y} - 2\Gamma^y_{xy} \dot{x}\dot{y} \end{cases} \tag{3.43}$$

This multivariable second order differential equation's solution (i.e., the geodesic) is $\mathcal{G} = \{x(\tau), y(\tau)\}$, where τ is a curve parameter, also called proper time in relativity physics. The differential equation is a boundary value problem, aiming to calculate the geodesic path between points A and B. Thus, it has the following boundary conditions:

$$\begin{cases} \mathcal{G}_{AB}(\tau = 0) = \{x_A, y_A\} \\ \mathcal{G}_{AB}(\tau = 1) = \{x_B, y_B\} \end{cases} \tag{3.44}$$

Now that the geodesic curve has been determined, it is possible to estimate the distance between points A and B as the length of the geodesic segment joining them. This is expressed as:

$$d^{geo}_{AB} = \int_{\tau_0 = 0}^{\tau_1 = 1} \sqrt{g_{xx} \dot{\mathcal{G}}^x_{AB}(\tau) + g_{yy} \dot{\mathcal{G}}^y_{AB}(\tau)}\ d\tau \tag{3.45}$$

This solves the problem of arc length along manifolds. As a final remark, all the discussed concepts equip the reader with the necessary mathematical notions to understand the methods developed in this Thesis.

3.2 Robotics concepts

A robot is an engineered system designed to interact with, manipulate, or perceive its environment under some level of autonomous or semi-autonomous control. This broad definition includes many types of robotic systems, however, the scope of this Thesis is mainly focused on industrial robotic manipulators. According to standard ISO 8373 [104], a manipulating industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes which may be either fixed in place or mobile for use in industrial automation applications. This Thesis will focus on fixed in site robots, with an emphasis on serial robots.

Within this context, all previously explained mathematical concepts find their way into the definition of robotics and their use. This Subsection will provide the reader with a theoretical foundation on robotics and a usual industrial serial robot structure, topological robot spaces, as well as intrinsic and extrinsic chains, which are the cornerstones of Chapters 6 and 7.

3.2.1 Structure of a robotized station

For the purposes of this Thesis, a robotized station is comprised of several key components:

- **The robotic manipulator.** The main component, it includes several subsystems. These subsystems include the mechanical frame or chassis (comprised by rigid links), a set of actuated joints or axes, a base mounting surface to enable installation, and a mechanical interface with the end effector. All these elements can be connected in different ways, which are called kinematic architectures.
- **The end effector.** A device specifically designed for attachment to the mechanical interface to enable the robot to perform its task. Several types of end effectors were designed and utilized In this Thesis, including distance sensor fixtures and build platforms.
- **The extrinsic elements.** These are all other station elements, with whom the robotic manipulator may interact. These elements include the robot's physical workspace, part fixtures, external sensors on fixed reference frames, and so on.

Particularly, this theoretical basis Chapter will focus on the robotic manipulator, learning how to define reference frames for links and axes, their transformations and several useful operations. All these concepts will be based on some of the topology concepts previously defined.

3.2.2 Topology of robot spaces

Classically, robotic manipulators have been defined by their number of Degrees of Freedom (DOF), which are the variables required to define the motion of a body in space, and are usually equal to the number of axes in serial robotic manipulators. When each of the axes is set to a joint displacement value, a configuration $\boldsymbol{\theta} = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{pmatrix}^T$ is defined. This configuration completely determines the shape of the robotic manipulator at any time.

However, it is possible to apply topology concepts onto this notion by defining the configuration space $\{\theta\}$. This configuration space is an implicit embedded map on an n-manifold. Usually, for 6 DOF serial robotic manipulators, this n-manifold takes the shape of the concatenation of six angular spaces $S^1 \times \cdots \times S^1$ (i.e., a 6-dimensional torus T^6). The reason for this is the usual kinematic architecture displayed by industrial serial robotic manipulators, which consists of 6 revolute joints, connected with rigid links.

This configuration space, oftentimes called workspace, lives in conjunction with the task space, space in which the task is naturally expressed. This is often a ambient Cartesian space, with $\{xyz\}$ translations, and both extrinsic $\{xy\}$ and intrinsic $\{z\}$ rotations enabled, resulting in a $\mathbb{R}^3 \times S^2 \times S^1$ space. If a point in this space is reachable by the robot, that point may be called a pose. In robotics, two spaces are usually defined for convenience: the space frame $\{s\}$, which is often located in the robot's base and serves as both the world coordinate system $\{wcs\}$ and the base coordinate system; and the body frame $\{b\}$, associated with the end effector (i.e., the tool center point, or TCP). Figure 3.8 visually shows all three frames.

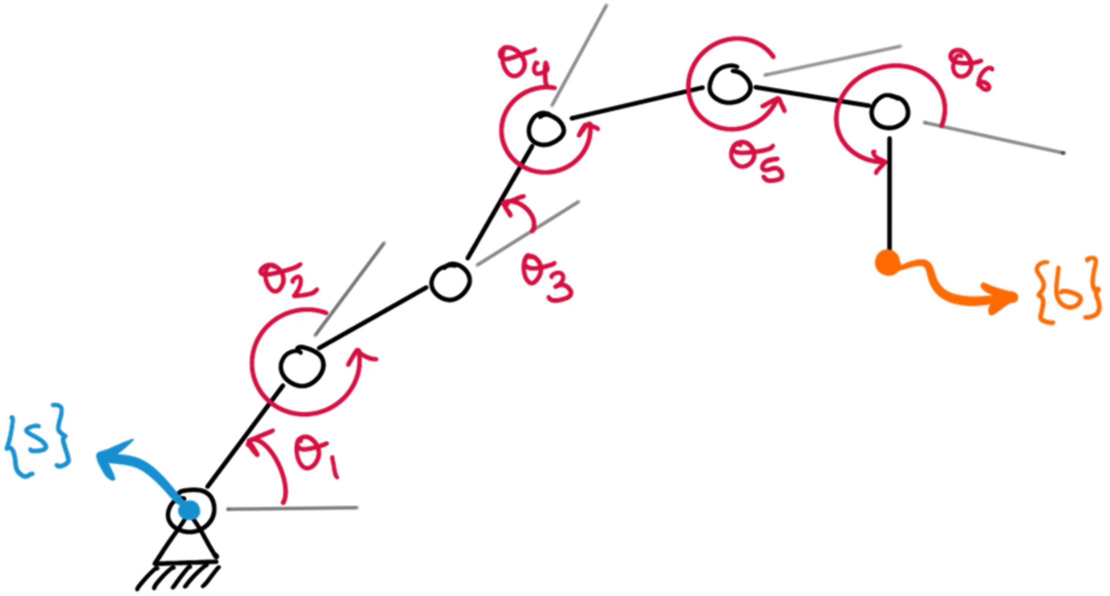


Figure 3.8: A visual representation of the configuration space $\{\theta\}$, the space frame $\{s\}$, and the body frame $\{b\}$.

These spaces are not infinitely large. The configuration space is constrained to $\theta = [0, N] \in S^1$ subdomains, which are often called joint limits. These joint limits restrict the amount of total configurations a robot may display, and are critical in the development of robot applications.

As per the concepts outlined in the previous Subsection, the existence of both a configuration space and an ambient task space begs the introduction of the homeomorphism of the robotic manifold between them. This homeomorphism is fundamentally locally bijective due to the aforementioned joint limits, which restrict the amount of reachable poses. Both directions of the homeomorphism are none other than the forward and inverse kinematics, and will be explained in a later Subsection. Before that, to fully understand these concepts, it is necessary to introduce rigid body motions in robotic manifolds, which may be performed by means of

Lie algebras, otherwise known as Screw Theory.

3.2.3 Lie algebras: Screw Theory

The management of rigid body motions has been classically founded on rotation matrices. A rotation matrix $\mathbf{R}$ is a 3×3 object belonging to the special orthogonal group $SO(3)$ with the following properties:

- **Orthogonality.** All column vectors of a rotation matrix are orthogonal to each other. This can be expressed as:
$$\mathbf{R}^T\mathbf{R} = 1 \tag{3.46}$$
- **Right-handedness and unity.** For the purposes of this work, a rotation is right-handed by definition, and all vectors inside the rotation matrix are of unit length. This is expressed as:
$$\det \mathbf{R} = 1 \tag{3.47}$$

However, rotation matrices are one of the many rotation conventions available to robotics users. A different rotation convention is the axis-angle notation, where any given frame rotation is expressed through an angular velocity $\omega_{\{s\}} = \hat{\omega}_{\{s\}}\dot{\theta}$, where $\hat{\omega}_{\{s\}}$ is a unit vector indicating the axis of rotation (the hat notation indicates unit length), and $\dot{\theta}$ is the amount of rotational velocity. The subscript curly braced notation indicates the frame in which any object is expressed. Once the angular velocity is defined, a frame's ($\{b\} = \{\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b\}$) rotational velocity about the axis $\hat{\omega}_{\{s\}}$ is calculated through the following expressions:

$$\dot{\hat{\mathbf{x}}}_b = \omega_{\{s\}} \times \hat{\mathbf{x}}_b \tag{3.48}$$
$$\dot{\hat{\mathbf{y}}}_b = \omega_{\{s\}} \times \hat{\mathbf{y}}_b \tag{3.49}$$
$$\dot{\hat{\mathbf{z}}}_b = \omega_{\{s\}} \times \hat{\mathbf{z}}_b \tag{3.50}$$

A visual example is provided in Figure 3.9.

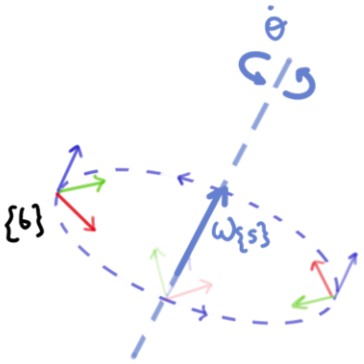


Figure 3.9: A visual representation of an axis-angle rotation.

To facilitate the cross product, it is possible to define the skew-symmetric form of a vector through the following procedure:

$$x \times y = [x]y \tag{3.51}$$

$$[x] = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{3.52}$$

Simplifying the previous formulae to:

$$\dot{\hat{\mathbf{x}}}_b = [\omega_{\{s\}}]\hat{\mathbf{x}}_b \tag{3.53}$$

$$\dot{\hat{\mathbf{y}}}_b = [\omega_{\{s\}}]\hat{\mathbf{y}}_b \tag{3.54}$$

$$\dot{\hat{\mathbf{z}}}_b = [\omega_{\{s\}}]\hat{\mathbf{z}}_b \tag{3.55}$$

To transform this angular velocity into angular displacement, it is necessary to introduce exponential coordinates for rotation. Given an axis of rotation $\hat{\omega}$ and an angular displacement θ, the rotation of p θ about $\hat{\omega}$ may be expressed as the following differential equation:

$$\dot{p}(t) = [\hat{\omega}]p(t) \tag{3.56}$$

$$p(t) = e^{[\hat{\omega}]\theta}p(0) \tag{3.57}$$

Where the exponential of a matrix is expressed as its Taylor expansion:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 \cdots \tag{3.58}$$

If expressed in matrix form, the exponential $e^{[\hat{\omega}]\theta}$ results as:

$$e^{[\hat{\omega}]\theta} = \mathbf{I} + \sin\theta[\hat{\omega}] + (1 - \cos\theta)[\hat{\omega}]^2 \in SO(3) \tag{3.59}$$

Which is in fact, the Rodrigues's formula. These concepts are useful when generalizing $SO(3)$ groups onto $SE(3)$ groups (the special Euclidean group), which also include translations. The most common $SE(3)$ object is the homogeneous transformation matrix $\mathbf{T}$, which is expressed as the following 4×4 matrix:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.60}$$

Where R is any given rotation matrix $\mathbf{R} \in SO(3)$ and $\mathbf{t}$ is a translation vector $\mathbf{t} \in \mathbb{R}^3$.

Homogeneous transformation matrices may represent configurations, change frames of reference, or displace vectors or frames. A particular property of homogeneous transformation matrix arises from the pre- or post-multiplication of $\mathbf{T}$ when applied onto a frame relationship $\mathbf{T}_{\{s\}\rightarrow\{b\}}$. If the pre-multiplication $\mathbf{T}\mathbf{T}_{\{s\}\rightarrow\{b\}}$ is executed, then the result expresses the extrinsic movement or space-frame transformation (i.e., with respect to origin, in this case, $\{s\}$) and the location of $\{b\}$ after the movement. This implies that $\{b\}$ is first rotated about $\{s\}$, then translated in a direction parallel to $\{s\}$. Otherwise, if the post-multiplication $\mathbf{T}_{\{s\}\rightarrow\{b\}}\mathbf{T}$ is applied, then the result expresses the intrinsic movement or body-frame transformation (i.e., with respect to itself, in this case, $\{b\}$). In this case, $\{b\}$ is first translated along a direction parallel to itself, then rotated about itself. Both cases are represented in Figure 3.10 for visual clarity. This property is useful in the context of robot calibration and will be made extensive use of.

$$\text{Let } \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \mathrm{L} \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation of 90º about z

Translation of L along x

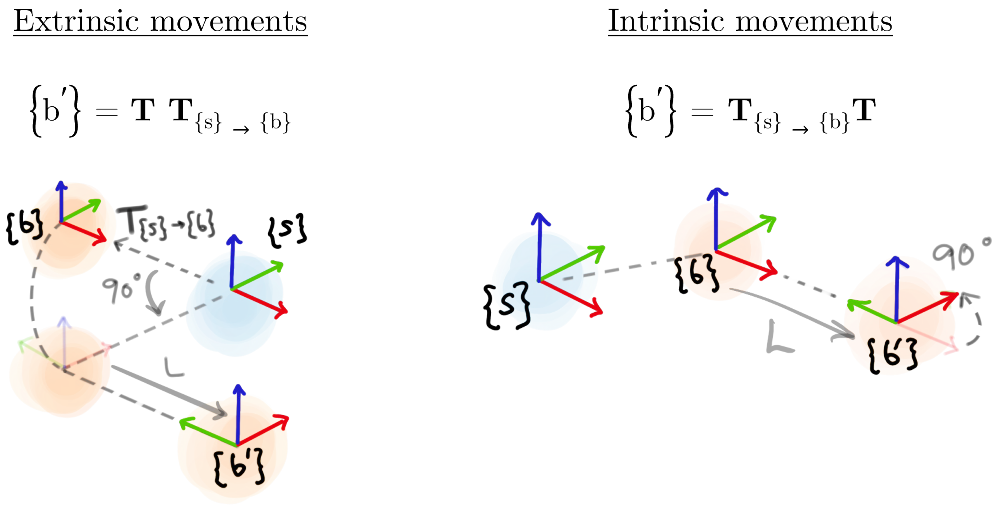


Figure 3.10: Difference between extrinsic and intrinsic movements.

Within this paradigm, it is possible to define the rigid body velocity (with linear and angular components) through the use of twists and screws. Any body's velocity may be expressed as the instantaneous velocity of the body moving about a screw axis $\hat{s}$, rotating at a $\dot{\theta}$ velocity with pitch h. Figure 3.11 visually represents the situation.

In mathematical terms, a screw $\mathcal{S}$ is defined as follows:

$$\mathcal{S} = \begin{pmatrix} \mathcal{S}_\omega \\ \mathcal{S}_v \end{pmatrix} = \begin{pmatrix} \text{angular velocity when } \dot{\theta} = 1 \\ \text{linear velocity of the origin when } \dot{\theta} = 1 \end{pmatrix} \tag{3.61}$$

Screw axes show several interesting properties:

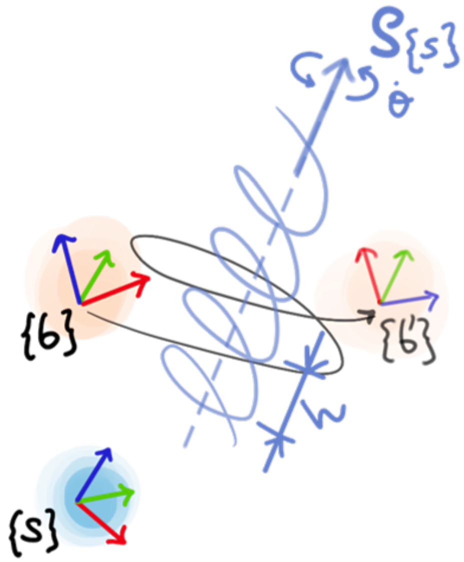


Figure 3.11: A visual representation of a screw-axis movement.

- **The pitch h may be infinite.** This results in a pure translation, where $\mathcal{S}_\omega = 0$, $\|\mathcal{S}_v\| = 1$, and $\dot{\theta}$ becomes a linear speed. This is useful in the context of actuated prismatic joints.
- **The screw axis is dependent of the origin configuration.** This allows for the definition of a home configuration, $\boldsymbol{\theta_H}$, in which screw axes are easily defined and invariant from then on.
- **Screw axes are expressed with respect to a frame.** This will have implications when twists (the generalization of angular velocity) are discussed.

In a similar fashion to the axis-angle formulation, a twist $\mathcal{V}$ is the application of a $\dot{\theta}$ velocity about screw axis $\mathcal{S}$, such that:

$$\mathcal{V} = \begin{pmatrix} \omega \\ v \end{pmatrix} = \mathcal{S}\dot{\theta} \tag{3.62}$$

Depending on the frame the screw axis has been defined, the twist may take several forms, be them $\mathcal{V}_{\{b\}}$ for the body twist (if the screw axis was defined in $\{b\}$) or $\mathcal{V}_{\{s\}}$ (if the screw axis was defined in $\{s\}$). Additionally, it is necessary to be able to change twist frames, as well as expressing twists in matrix representation. If a twist $\mathcal{V}_{\{a\}}$ is to be represented in $\{b\}$, the most logical way of doing such task would imply utilizing the homogeneous transformation matrix $\mathbf{T}_{\{a\}\to\{b\}}$. However, twists are 6-dimensional objects, which are incompatible with 4×4 homogeneous transformation matrix. Therefore, it is necessary to introduce the adjoint representation of a transformation matrix $[\mathrm{Ad}_T]$, which is calculated as:

$$\text{Given } \mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \; [\mathrm{Ad}_{\mathbf{T}}] = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ [\mathbf{t}]\mathbf{R} & \mathbf{R} \end{pmatrix} \in \mathbb{R}^{6\times 6} \tag{3.63}$$

Now, it is possible to change frames $\{a\}$ and $\{b\}$ by means of:

$$\mathcal{V}_{\{a\}} = [\mathrm{Ad}_{\mathbf{T}_{\{a\}\to\{b\}}}]\mathcal{V}_{\{b\}} \tag{3.64}$$

Lastly, the matrix representation of a twist $[\mathcal{V}]$ is performed similarly to the matrix representation of angular velocities $[\omega]$ through the following equation, obtaining an $se(3)$ object akin to the $so(3)$ object obtained from skew-symmetric representations of rotations:

$$[\mathcal{V}] = \begin{pmatrix} [\omega] & v \\ 0 & 0 \end{pmatrix} \in se(3) \tag{3.65}$$

To convert these $se(3)$ objects into $SE(3)$, which include homogeneous transformation matrices, it is possible to apply an exponential matrix map, analogous to the exponential applied in the $so(3)$ case. Generally, the displacements in $SE(3)$ environments can be expressed through the Rodrigues's formula as:

$$e^{[\mathcal{S}]\theta} = \begin{pmatrix} e^{[\mathcal{S}_\omega]\theta} & (\mathbf{I}\theta + (1 - \cos\,\theta[\mathcal{S}_\omega] + (\theta - \sin\,\theta)[\mathcal{S}_\omega]^2))\mathcal{S}_v \\ 0 & 1 \end{pmatrix} \tag{3.66}$$

Let us provide a practical example, with the help of Figure 3.12.

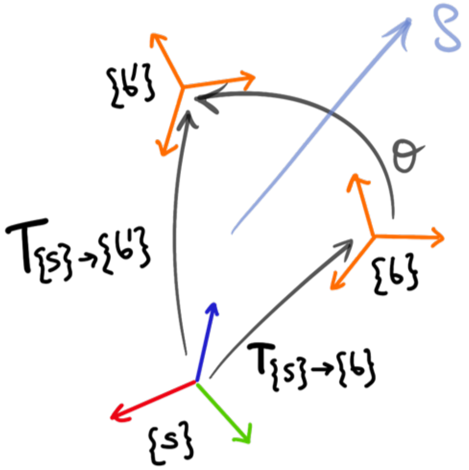


Figure 3.12: Example for screw axis movement calculation.

Let two frames $\{s\}$ and $\{b\}$. If a θ movement is to be applied onto $\{b\}$ about screw axis $\mathcal{S}_{\{b\}}$, the result $\{b'\}$ is calculated through:

$$\mathbf{T}_{\{s\}\to\{b'\}} = \mathbf{T}_{\{s\}\to\{b\}} e^{[\mathcal{S}_{\{b\}}]\theta} \tag{3.67}$$

Conversely, if screw axis is defined in the $\{s\}$ frame, then the resulting $\{b'\}$ is obtained by:

$$\mathbf{T}_{\{s\}\rightarrow\{b'\}} = e^{[\mathcal{S}_{\{s\}}]\theta}\mathbf{T}_{\{s\}\rightarrow\{b\}} \tag{3.68}$$

Notice the difference between pre- and post-multiplying by $\mathbf{T}$, which alludes to the aforementioned concept of intrinsic and extrinsic rotations in $SO(3)$. In $SE(3)$, this behavior is identical, and allows for seamless frame transitions between the world coordinate system $\{s\}$ and the tool center system $\{b\}$.

The last screw theory notion revolves around the definition of forces, which in this context are appropriately called wrenches. Let us have frames $\{s\}$ and $\{b\}$, rigidly joint by $\mathbf{T}_{\{s\}\rightarrow\{b\}}$. A force $f_{\{b\}}$ is applied at point $r_{\{b\}}$, both expressed in $\{b\}$. Hence, the force $f_{\{b\}}$ applies a moment $m_{\{b\}} = r_{\{b\}} \times f_{\{b\}}$. This situation is represented in Figure 3.13. In this situation, it is possible to define a wrench $\mathcal{F}_{\{b\}} = \begin{pmatrix} m_{\{b\}} \\ f_{\{b\}} \end{pmatrix}$, packing both components akin to angular and linear velocities in twists.

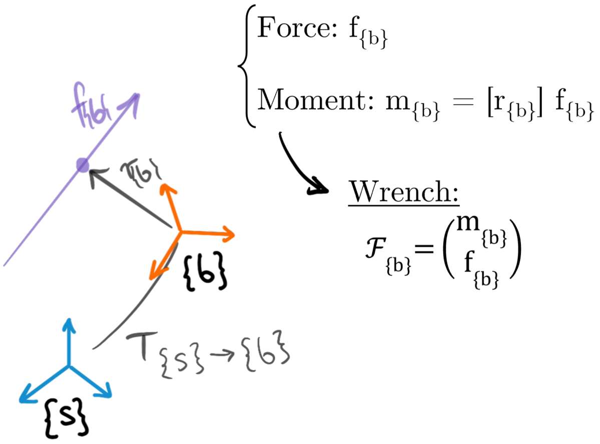


Figure 3.13: Example for wrench definition, including force and moment.

To estimate $\mathcal{F}_{\{s\}}$ (i.e., the wrench about the $\{s\}$ frame, it is first necessary to introduce power. Power is the dot product between a twist and a wrench $P = \mathcal{V}^T_{\{b\}}\mathcal{F}_{\{b\}}$, and is irrespective of frame of reference. Therefore, the following expression is true:

$$\text{power} = P = \mathcal{V}^T_{\{b\}}\mathcal{F}_{\{b\}} = \mathcal{V}^T_{\{s\}}\mathcal{F}_{\{s\}} \tag{3.69}$$

Introducing the frame shift formula for twists yields:

$$\mathcal{V}^T_{\{s\}}\mathcal{F}_{\{s\}} = ([\mathrm{Ad}_{\mathbf{T}_{\{b\}\rightarrow\{s\}}}]\mathcal{V}_{\{s\}})^T\mathcal{F}_{\{b\}} \tag{3.70}$$

$$\mathcal{V}^T_{\{s\}}\mathcal{F}_{\{s\}} = \mathcal{V}^T_{\{s\}}[\mathrm{Ad}_{\mathbf{T}_{\{b\}\rightarrow\{s\}}}]^T\mathcal{F}_{\{b\}} \tag{3.71}$$

$$\mathcal{F}_{\{s\}} = [\mathrm{Ad}_{\mathbf{T}_{\{b\}\rightarrow\{s\}}}]^T\mathcal{F}_{\{b\}} \tag{3.72}$$

With these screw theory concepts, it is possible to define the kinematics and dynamics of intrinsic chains (i.e., dependent on robot configuration) and extrinsic chains (i.e., irrespective of robot configuration).

3.2.4 Intrinsic chains: kinematics

The kinematics problem on serial industrial robotic manipulators is establishing the homeomorphism between the configuration space and the task space. In classical terms, this is referred as both the forward kinematics (from the configuration space towards the task space), and the inverse kinematics (from the task space towards the configuration space), which are the two directions of the homeomorphism.

The forward kinematics is the transformation between a given configuration $\boldsymbol{\theta}$ and its corresponding pose $\mathbf{T}_{\{s\}\to\{b\}}(\boldsymbol{\theta})$ within the task space. This may be done through the Product of Exponentials formula. For a 6R serial robotic manipulator, whose screw axes are defined in $\{s\}$, this is expressed as:

$$\mathbf{T}_{\{s\}\to\{b\}}(\boldsymbol{\theta}) = e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_6]\theta_6}\mathbf{M}(\boldsymbol{\theta} = \mathbf{0}) \tag{3.73}$$

The previous Product of Exponentials formula corresponds to the situation of Figure 3.14. In this situation, $\mathbf{M}(\boldsymbol{\theta} = \mathbf{0})$, or just $\mathbf{M}$, corresponds to the home pose. $\mathbf{M}$ is the location of the $\{b\}$ frame with respect to the $\{s\}$ frame when all axes values are equal to zero. It is in this home pose that the screw axes $\mathcal{S}_1$ through $\mathcal{S}_6$ are defined. Owing to the previously defined property of screw axes, $\mathcal{S}_1$ through $\mathcal{S}_6$ remain invariant from this point onward.

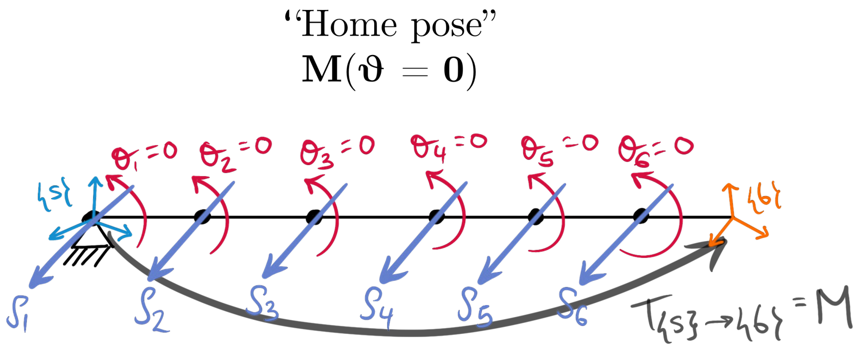


Figure 3.14: Situation for Product of Exponentials formula derivation.

To express velocities, it is necessary to introduce the Jacobian. Each column of the Jacobian $\mathbf{J}(\boldsymbol{\theta})$ represents the velocity of the end-effector when a velocity of that joint is equal to one, and the velocity at all other joints is zero. It is possible to express the Jacobian with respect to the space frame $\{s\}$, or the body frame $\{b\}$. The space Jacobian $\mathbf{J}_{\{s\}}$ is defined as follows for a 6 DOF manipulator:

$$\mathcal{V}_{\{s\}} = \mathbf{J}_{\{s\}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \begin{pmatrix} \mathbf{J}_{\{s\}1}(\boldsymbol{\theta}) & \cdots & \mathbf{J}_{\{s\}6}(\boldsymbol{\theta}) \end{pmatrix} \dot{\boldsymbol{\theta}} \tag{3.74}$$
$$\mathbf{J}_{\{s\}}(\boldsymbol{\theta}) \in \mathbb{R}^{6\times 6}$$
$$\mathbf{J}_{\{s\}1} = \mathcal{S}_1 \tag{3.75}$$
$$\mathbf{J}_{\{s\}i}(\boldsymbol{\theta}) = [\mathrm{Ad}_{(e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_{i-1}]\theta_{i-1}})}]\mathcal{S}_i \text{ for } i = 2, \dots, n. \tag{3.76}$$

Conversely, the body Jacobian is defined as follows:

$$\mathcal{V}_{\{b\}} = \mathbf{J}_{\{b\}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \begin{pmatrix} \mathbf{J}_{\{b\}1}(\boldsymbol{\theta}) & \cdots & \mathbf{J}_{\{b\}6}(\boldsymbol{\theta}) \end{pmatrix} \dot{\boldsymbol{\theta}} \tag{3.77}$$
$$\mathbf{J}_{\{b\}}(\boldsymbol{\theta}) \in \mathbb{R}^{6\times 6}$$
$$\mathbf{J}_{\{b\}6} = \mathcal{B}_n \tag{3.78}$$
$$\mathbf{J}_{\{b\}i}(\boldsymbol{\theta}) = [\mathrm{Ad}_{(e^{-[\mathcal{B}_n]\theta_n} \cdots e^{-[\mathcal{B}_{i+1}]\theta_{i+1}})}]\mathcal{B}_i \text{ for } i = 1, \dots, n-1. \tag{3.79}$$

To easily transform between the space and body Jacobians, the following formulae shall be applied:

$$\mathbf{J}_{\{b\}}(\boldsymbol{\theta}) = [\mathrm{Ad}_{\mathbf{T}_{\{s\}\to\{b\}}}]\mathbf{J}_{\{s\}}(\boldsymbol{\theta}) \tag{3.80}$$
$$\mathbf{J}_{\{s\}}(\boldsymbol{\theta}) = [\mathrm{Ad}_{\mathbf{T}_{\{b\}\to\{s\}}}]\mathbf{J}_{\{b\}}(\boldsymbol{\theta}) \tag{3.81}$$

Where both transformation matrices $\mathbf{T}$ may be obtained through the Product of Exponentials formula.

The conditioning of the Jacobian matrix is paramount to the viability of a robotic manufacturing process. If $\mathbf{J}$ is singular (i.e., its rank is lower than 6) at a certain configuration $\boldsymbol{\theta}^*$, then the robotic manipulator is at a singularity, losing degrees of freedom. Therefore, the Jacobian also informs of the manipulability of the robotic system when the joint limits are accounted for, establishing how close it is to a singularity. In a general case where the joint velocities form a 6-sphere such that $\dot{\boldsymbol{\theta}}^T\dot{\boldsymbol{\theta}} = 1$, and:

$$v_{\text{end-effector}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{3.82}$$
$$v_{\text{end-effector}} \in \mathbb{R}^6, \dot{\boldsymbol{\theta}} \in \mathbb{R}^6, \mathbf{J}(\boldsymbol{\theta}) \in \mathbb{R}^{6\times 6}$$

Then, accounting for the joint limits yields:

$$\dot{\boldsymbol{\theta}}^T\dot{\boldsymbol{\theta}} = 1 \tag{3.83}$$

$$(\mathbf{J}^{-1}v_{\text{end-effector}})^T(\mathbf{J}^{-1}v_{\text{end-effector}}) = 1 \tag{3.84}$$

$$v_{\text{end-effector}}^T(\mathbf{J}\mathbf{J}^T)^{-1}v_{\text{end-effector}} = 1 \tag{3.85}$$

$$v_{\text{end-effector}}^T\mathbf{A}^{-1}v_{\text{end-effector}} = 1 \tag{3.86}$$

$$\mathbf{A} = \mathbf{J}\mathbf{J}^T \in \mathbb{R}^{6\times 6} \tag{3.87}$$

A is a positive definite and symmetric matrix, and its inverse creates a quadratic form with the end-effector velocity. This allows for the definition of the manipulability 6-ellipsoid, with semi-axes lengths $\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_6}$, in directions parallel to $v_1, \ldots, v_6$, which are the matrix eigenvalues and eigenvectors. Additionally, three independent measurements can express the closeness to singularity of a given configuration, defined by the following equalities:

$$\mu_1(\mathbf{A}) = \frac{\sqrt{\lambda_{\text{max}}(\mathbf{A})}}{\sqrt{\lambda_{\text{min}}(\mathbf{A})}} \geq 1 \tag{3.88}$$

$$\mu_2(\mathbf{A}) = \frac{\lambda_{\text{max}}(\mathbf{A})}{\lambda_{\text{min}}(\mathbf{A})} \geq 1 \tag{3.89}$$

$$\mu_3(\mathbf{A}) = \sqrt{\lambda_1 \cdots \lambda_6} = \sqrt{\det(\mathbf{A})} \geq 0 \tag{3.90}$$

The inverse kinematics of a manipulator is obtaining the transformation from the task space towards the configuration space. For open chains, such as a 6R serial robotic manipulator, this solution is generally not unique, and therefore, the topologic space transformation is not bijective. In this sense, there may be poses with one, more than one, or no configurations associated with it. This poses a challenge for path planning. The inverse kinematics problem may be formulated as follows:

$$\begin{gathered}\text{find } \boldsymbol{\theta_s} \text{ such that:}\\ \mathbf{x_t} - f(\boldsymbol{\theta_s}) = \mathbf{0}\end{gathered} \tag{3.91}$$

Where $\mathbf{x_t}$ is the target pose, and $\boldsymbol{\theta_s}$ is the configuration solution. This is numerically solvable through a plethora of methods, including the Newton-Raphson algorithm.

Chapter 4

Trajectory definition through an embedded map

Being able to asynchronize the Process Set-up from the Process Operation is one of the keys for efficiency in a manufacturing cell-based job shop. This enables the parallelization of order management, Process planning, and manufacturing proper. Additionally, this drastically increases the time the Process is performing value-adding tasks within the work day, and therefore, productivity. Additionally, in the context of RBNPAM, this Offline Set-up is considered one of the current challenges, owing to the difficulty of the mathematical concepts involved and the need for knowledge of both NPAM path planning procedures and robotics.

In this regard, this Chapter proposes a novel methodology of performing Offline Set-ups in the context of RBNPAM. This will be done through a methodology of embedded maps generation, that accounts for build platform geometry (Section 4.1). An extensive set of tools was developed to realize this methodology, and documentation is provided in Section 4.2. This methodology is able to slice models uniformly (i.e., with a constant layer height), and conformally (i.e., with layers parallel to the build platform), according to axisymmetric layers. In addition to the slicing, it is necessary to determine the paths the manufacturing equipment needs to follow. To do so, a slicing methodology has been developed (Section 4.3). This slicing methodology presents some drawbacks regarding infilling, and some insights relating to the following Chapter are provided to solve this issue. The concepts explained in this Chapter are the foundation of a publication in which the present author has actively participated [105].

4.1 Systematically generating an axisymmetric embedded map

One idea of the RBNPAM cell developed, as previously outlined, is the introduction of build platforms (carried by the robotic manipulator) moving around a fixed position extruder that additively manufactures parts on them. This build platform sets all the layer geometries through the definition of an embedded map.

The embedded map method consists of the homeomorphism of any Riemannian manifold (i.e., generally curved object) into an Euclidean manifold. To properly define this homeomorphism, it would be necessary to define both its forward transformation (from the $n-1$-manifold contained in the ambient space $\mathbb{R}^n$ to the embedded map $\{\varphi\}$) and its inverse transformation (from $\{\varphi\}$ to the $n-1$-manifold). Let us start with a simple example: the cylindrical embedded map. Figure 4.1 represents the 2-manifold $\mathcal{M}$ in question, as well as an embedded map $\{\varphi\}$ of it.

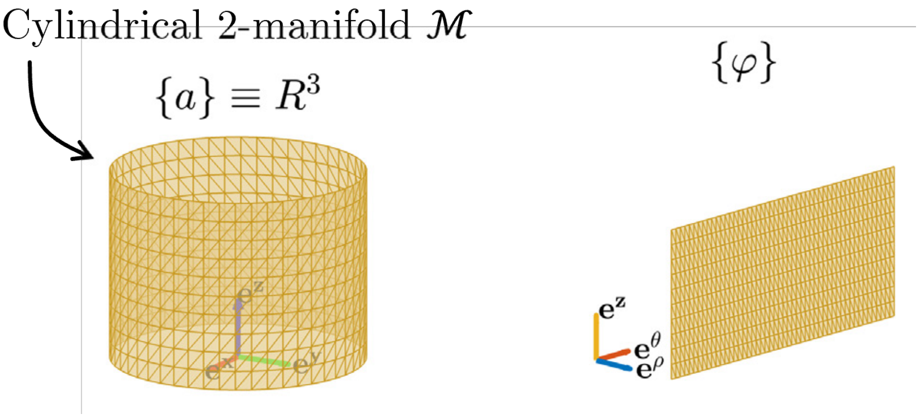


Figure 4.1: The cylindrical embedded map example.

The homeomorphism depicted above corresponds to the usual equations of the cylindrical coordinate system:

$$\mathcal{M} \rightarrow \{\varphi\}$$

$$\{x, y, z\} \rightarrow \begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \tan^{-1}(\frac{y}{x}) \\ z = z \end{cases} \tag{4.1}$$

$$\mathcal{M} \leftarrow \{\varphi\}$$

$$\left.\begin{matrix} x = r\text{cos}\theta \\ y = r\text{sin}\theta \\ z = z \end{matrix}\right\} \leftarrow \{r, \theta, z\} \tag{4.2}$$

This homeomorphism implies the transformation between a curved object ($\mathcal{M}$) and an Euclidean object ($\{\varphi\}$). It is possible to represent several other objects in both representations, namely some planes. These elements are represented in both spaces in Figure 4.2.

Of note, the purple plane in $\{\varphi\}$ from Figure 4.2 corresponds with a cylindrical surface in the ambient space $\mathbb{R}^3$. In fact, offsetting the purple plane a given distance H along the r axis in $\{\varphi\}$ gives as a result a lamination of the ambient space following cylindrical layers. This means that, given a build platform in the shape of a cylinder with radius r_0, the Cartesian ambient space can be sliced with cylindrical layers from r_0 and above. This is a form of uniform conformal slicing. The main question appears when trying to generalize this behavior.

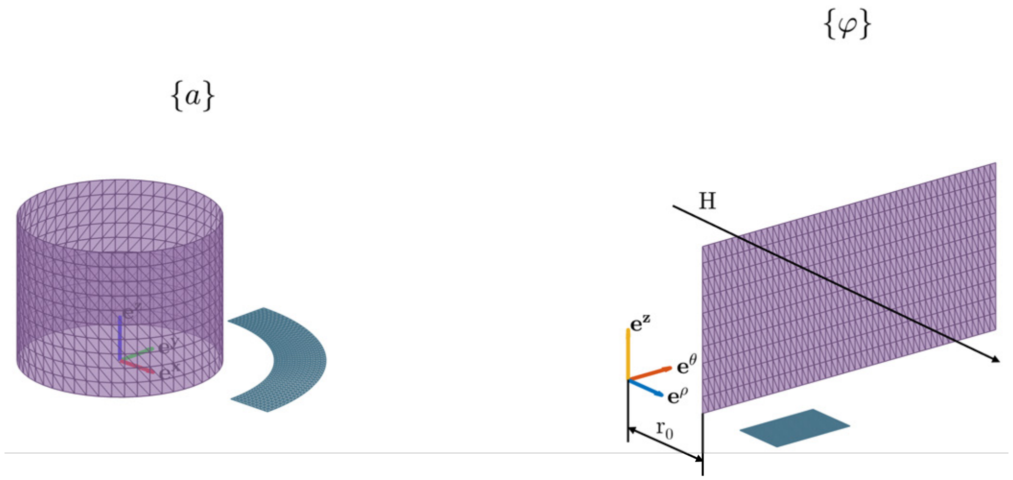


Figure 4.2: Several objects represented in both spaces.

This methodology is easy to understand and apply when faced with the usual coordinate systems (e.g., cylindrical or spherical). However, it is not intuitive how to generalize this procedure towards axisymmetric slicing. In this sense, it is necessary to introduce a method to systematically create embedded maps akin to the cylindrical coordinate system for cylindrical manifold slicing.

Given a parametric curve $\mathcal{S}(u) = \mathbf{uBP}$, expressed in the ambient space $\{b\} \equiv \mathbb{R}^3$ and contained within a plane π, and given a unit axis of revolution $\mathcal{E}$ within π that traverses the origin of $\{b\}$; there exists an embedded map $\{\varphi\} = \{u, \theta, h\}$ such that:

$$\mathbf{e_u} = e^{[\mathcal{E}]\theta} \frac{\nabla \mathcal{S}}{\|\nabla \mathcal{S}\|} \tag{4.3}$$

$$\mathbf{e}_{\boldsymbol{\theta}} = e^{[\mathcal{E}]\theta} [\mathcal{E}] \frac{[\mathcal{S}]\mathcal{E}}{\|[\mathcal{S}]\mathcal{E}\|} \tag{4.4}$$

$$\mathbf{e_h} = [\mathbf{e_u}]\mathbf{e}_{\boldsymbol{\theta}} \tag{4.5}$$

A visual representation of this map embedding is shown in Figure 4.3. In a simplified sense, the base curve $\mathcal{S}$ (which serves as a generatrix), provides a support for the trihedron responsible for the homeomorphism. The first vector, $\mathbf{e_u}$, is $\mathcal{S}$'s tangent unit vector and is calculated as such. The second vector, $\mathbf{e}_{\boldsymbol{\theta}}$, is the unit vector resulting from the revolution of the curve around the axis of revolution $\mathcal{E}$. The definition for $\mathbf{e}_{\boldsymbol{\theta}}$ results from the Rodrigues's formula in exponential form applied onto the rotation of any given point $\mathcal{S}$ around $\mathcal{E}$, where the bracket notation indicates the skew-symmetric form of the vector inside. Lastly, $\mathbf{e_h}$ is $\mathcal{S}$'s unit normal vector, and it is calculated from the cross product of $\mathbf{e_u}$ and $\mathbf{e}_{\boldsymbol{\theta}}$. This trihedron definition is irrespective of any $\mathcal{E}$, π, or more importantly, $\mathcal{S}$, which means it is a general and systematic manner of producing an axisymmetric slicing procedure.

Once the trihedron is defined, it is time to define the homeomorphism between the Cartesian $\mathbb{R}^3$ space and the embedded map $\{\varphi\}$, which may be called the slicing space. To do this, it

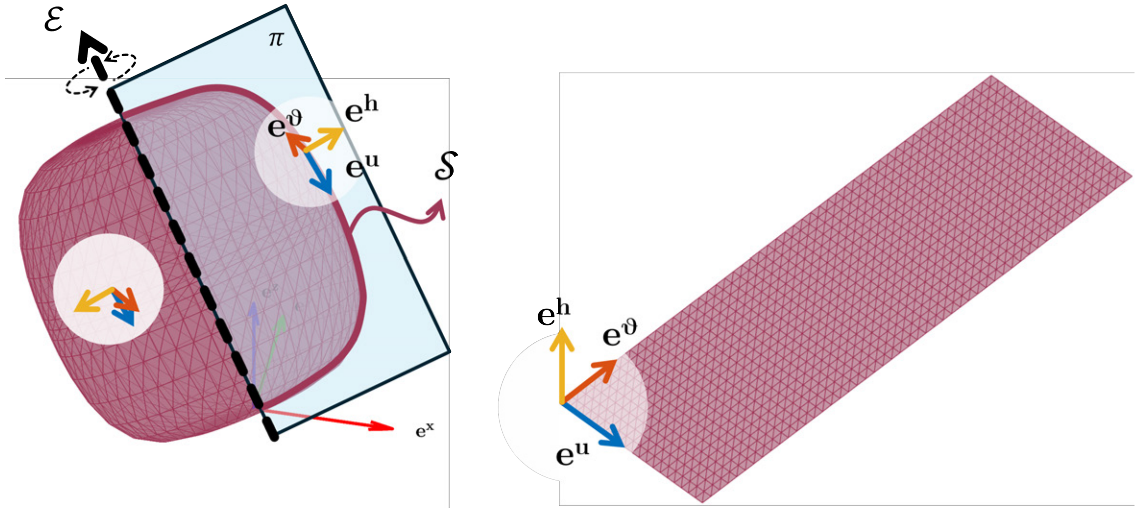


Figure 4.3: Systematic map embedding methodology, representing an example including all base elements ($\mathcal{S}$, π, and $\mathcal{E}$), and indicating the three basis vectors ($\mathbf{e_u}$, $\mathbf{e_{\boldsymbol{\theta}}}$, $\mathbf{e_h}$) for a given point in the ambient space.

is necessary to define the forward and inverse transformations, in a similar fashion to the cylindrical coordinate system previously outlined. First, the inverse transformation is assessed. The inverse transformation (from the slicing space $\{\varphi\}$ towards the Cartesian space $\{b\}$) is defined as:

$$\{\varphi\} \rightarrow \{b\}$$

$$\{u, \theta, h\} \rightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix} = e^{[\mathcal{E}]\theta}(\mathcal{S}(u) + h\mathcal{N}(u)) \tag{4.6}$$

Where $\mathcal{N}(u)$ is $\mathcal{S}$'s unit outward normal vector:

$$\mathcal{N}(u) = \left[\frac{\nabla\mathcal{S}(u)}{\|\nabla\mathcal{S}(u)\|}\right][\mathcal{E}]\frac{[\mathcal{S}]\mathcal{E}}{\|[\mathcal{S}]\mathcal{E}\|} \tag{4.7}$$

This inverse transformation is equivalent to finding the point of $\mathcal{S}$ when the parameter is equal to u, then displace it a distance h along its unit outward normal vector, then rotate it around the axis of revolution $\mathcal{E}$. A graphical representation is provided in Figure 4.4.

The forward transformation problem consists of finding the point $p_{\{\varphi\}}$ which inverse transforms into a given point $p_{\{b\}}$. This forward problem is impossible to solve analytically in a general sense, however, it is possible to apply numerical methods. To apply them, some simplifications may be introduced to the problem to alleviate some of the calculations. First, defining $\mathcal{E}$ on an arbitrary π plane introduces complexity. Without loss of generality, it is possible to always

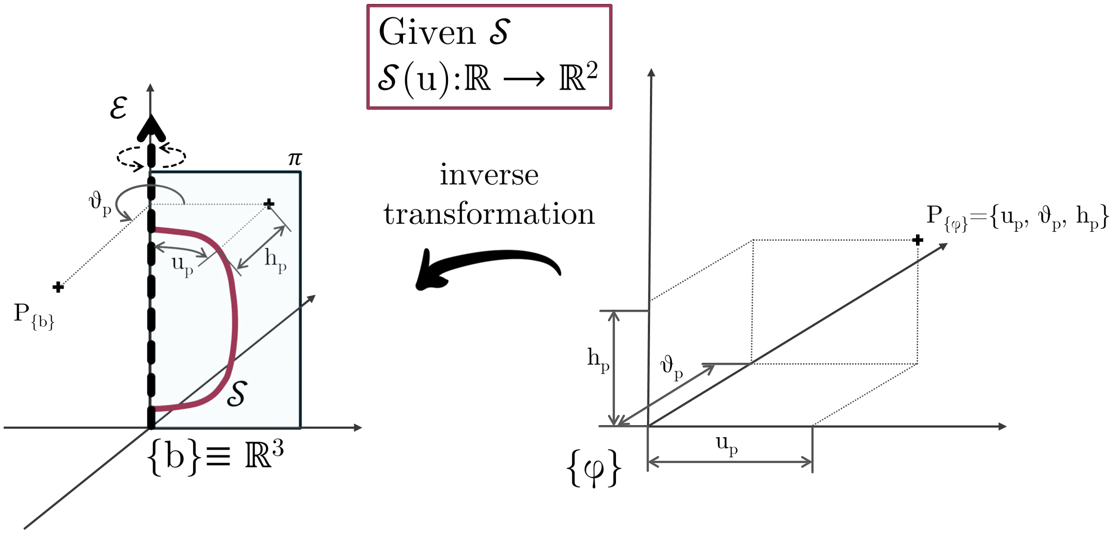


Figure 4.4: Representation of the inverse transformation for a given point p.

define π as the zx-plane, then rotate and translate the whole ambient space. This simplifies the homeomorphism to:

$$\mathcal{N}(u) = \left[\frac{\nabla\mathcal{S}(u)}{\|\nabla\mathcal{S}(u)\|}\right]\mathbf{j} \tag{4.8}$$

$$\{u,\theta,h\} \rightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix} = e^{[\mathbf{e^z}]\theta}(\mathcal{S}(u) + h\left[\frac{\nabla\mathcal{S}(u)}{\|\nabla\mathcal{S}(u)\|}\right]\mathbf{e^y}) \tag{4.9}$$

Where $\mathbf{e^y}$ and $\mathbf{e^z}$ are the unit vectors of the Cartesian basis axes y and z, respectively. Still, for a general case curve $\mathcal{S}$, the forward transformation is not analytically solvable due to the denominator $\|\nabla\mathcal{S}(u)\|$. However, not only has the problem been mathematically simplified, but also the design procedure for creating the embedded map has. Within this simplification, it is possible to define curves on the zx-plane, then revolve them around the z axis, then transform the whole system onto the preferred reference frame $\{b\}$. However, both the general form and the simplified form have their uses, especially if one wishes to extend the axisymmetric embedded map concept onto a helical-symmetry embedded map methodology. This, however, falls outside the scope of the present Thesis.

Let us provide two examples of the procedure, anticipating some of the tools that will be discussed in the next Section. The first example is defined with the following inputs:

$$\mathcal{S}(u) = \mathbf{uBP}, \mathbf{u} = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0.75 \\ 1 & 0 & 0.25 \\ 1 & 0 & 0 \end{pmatrix} \tag{4.10}$$

$$\pi \equiv -y = 0, \mathcal{E} = \mathbf{e^z} \tag{4.11}$$

This situation is represented in Figure 4.5, and corresponds to a cylindrical build platform. Therefore, the results should be similar to the cylindrical coordinate system previously introduced.

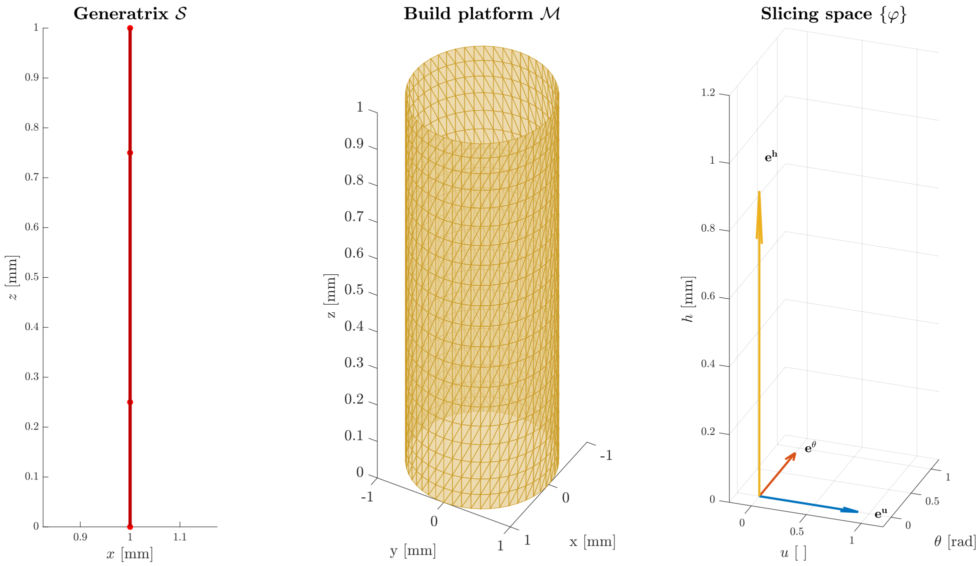


Figure 4.5: Example 1 for homeomorphism demonstration.

Within this embedded map definition, let us transform several elements: four points (A through D), four segments (a through d), and therefore, the rectangle α, delimited by the previous elements. As seen in Figure 4.6, this rectangle α is contained within the plane $h = 1$ in $\{\varphi\}$. Rectangles on successive h planes also exhibit the predicted behavior (Figure 4.7).

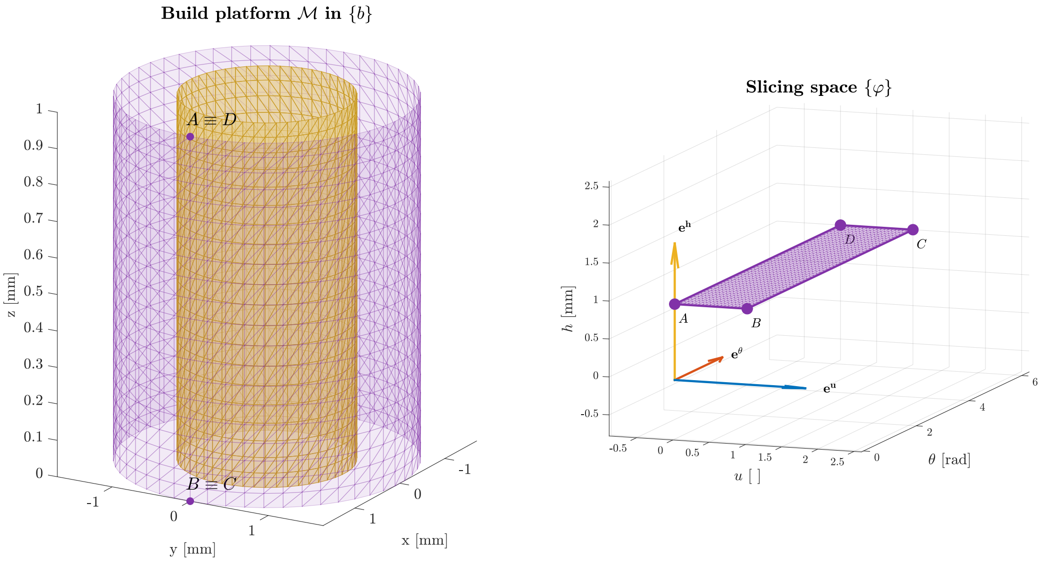


Figure 4.6: The rectangle α in both spaces.

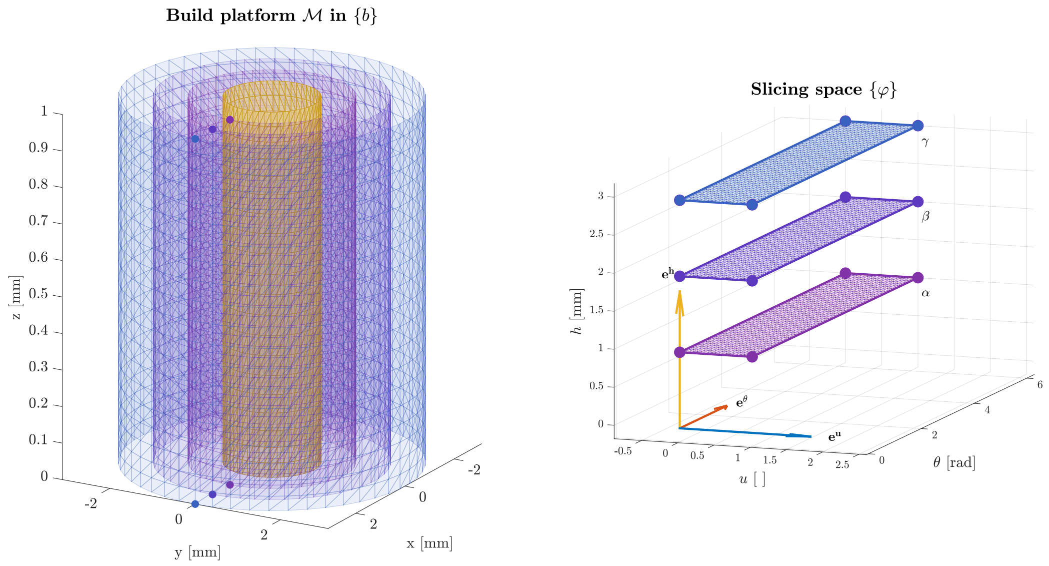


Figure 4.7: Several h planes in both spaces.

There exist some differences between the cylindrical coordinate system and this embedded map method. Firstly, in the embedded map there is a region of bijectivity, unlike in the cylindrical coordinate system. One of the implications of this is that there are points which have no correspondence in the embedded map (the forward transformation is not injective). This phenomenon is represented in Figure 4.8, where the red points in $\{b\}$ do not have an image on $\{\varphi\}$.

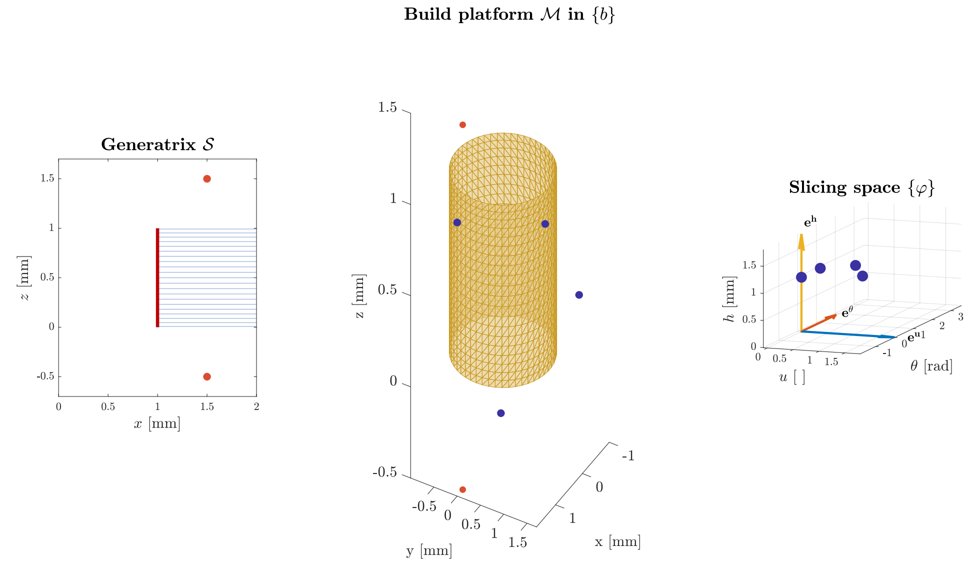


Figure 4.8: Lack of bijectivity of the presented example. The bijectivity region is represented as a blue striped area.

Figure 4.9 represents the second example of an homeomorphism. The plane and the revolution axis are the same as in the previous example ($\pi \equiv -y = 0$, $\mathcal{E} = \mathbf{e^z}$), whereas the curve $\mathcal{S}$ is defined as:

$$\mathcal{S}(u) = \mathbf{uBP}, \mathbf{u} = \begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.12}$$

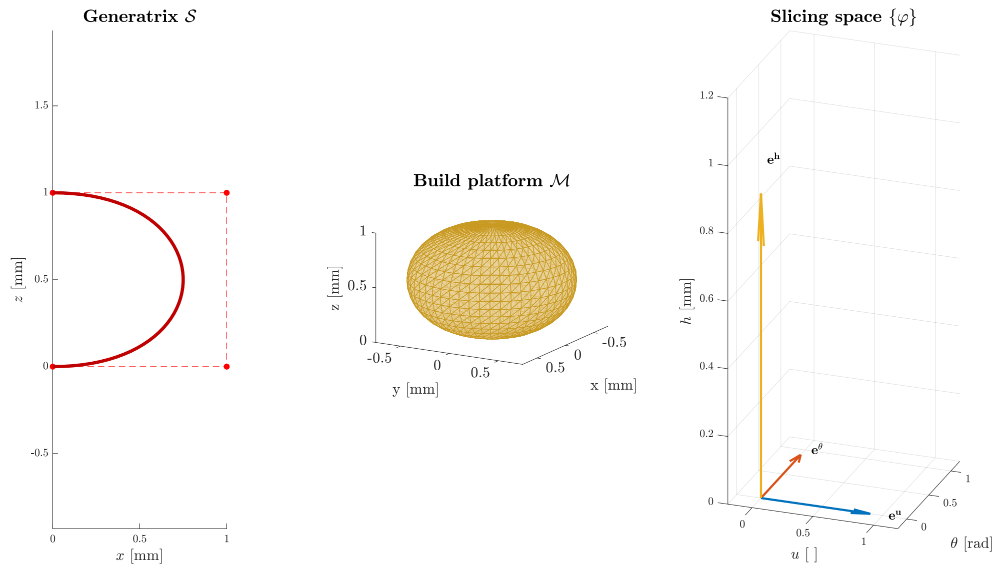


Figure 4.9: Example 2 for homeomorphism demonstration.

This curve produces sphere-like surfaces when defining horizontal planes along the h axis. This homeomorphism is actually bijective for $h > 0$, which is the usual range of application of the methodology. Visualizations of these concepts are presented in Figure 4.10.

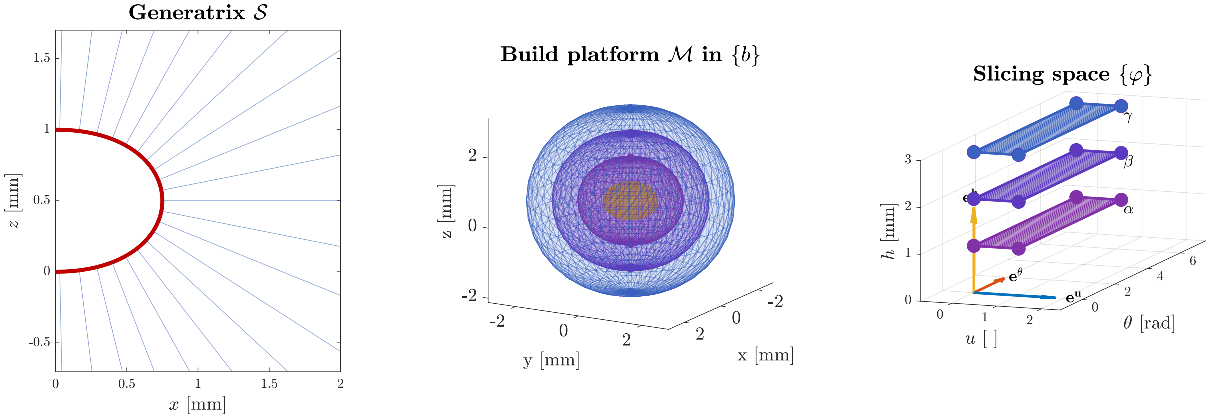


Figure 4.10: $h > 0$ planes (in colors) and bijectivity region as a blue striped area.

The previous examples provide but a glimpse of the possibilities the present methodology may achieve. In order to extend the previous concepts onto general axisymmetric slicing, a toolset has been developed in MATLAB.

4.2 Tools for handling axisymmetric embedded maps

The embedded map concepts previously outlined provided the core idea for embedded map-based axisymmetric slicing. However, to realize this, two MATLAB classes have been developed to manage two particular instances of parametrized curves: degree-3 and degree-6 Bézier splines, which are composites of Bézier curve segments of each degree. This Section will explain all the tools contained in both classes (*Bezier2DCat.m* and *SixDegBezier2DCat.m*), serving as documentation for the code. This Section will cover the degree-3 Bézier curve toolbox extensively, and it is important to note that all the attributes, methods, and overall concepts, have been applied in the same manner for the degree-6 Bézier curve class.

First, the attributes (called properties in MATLAB jargon), are the intrinsic parameters the class stores. The properties of these classes are the following:

- `controlPoints`
  - The original set of control points provided by the user (possibly more than four points if there are multiple segments).
- `segmentLengths`
  - A vector holding the "length" in parameter space for each Bézier segment. Typically these add up to a total parameter range. Note that these are not geometric arc-lengths; rather, they are how the parameter u domain is partitioned.
- `totalLength`
  - Sum of all the elements in `segmentLengths`. Effectively, the end of the parameter domain (u from 0 to `totalLength`).
- `weights`
  - A property for storing weights (if needed by the user).
- `controlMatrix`
  - The main control-points matrix used for the piecewise Bezier segments. If `controlPoints` has more than 4 points, the constructor manipulates `controlMatrix` by duplicating some rows so each piecewise segment lines up properly. Essentially ensures each set of 4 rows in `controlMatrix` corresponds to one cubic Bézier segment.
- `bezierUnitBasis`

  - A 4×4 matrix for a cubic Bezier basis. In the code, it is:

$$\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

    This is effectively the matrix that, when multiplied by $\begin{pmatrix} u^3 & u^2 & u & 1 \end{pmatrix}$ gives the typical cubic Bézier basis functions.

- `bezierBasisMatrix`
  - A larger block-diagonal matrix that incorporates `bezierUnitBasis` multiple times (once for each segment). It is sized to match the `controlMatrix` if you have multiple segments.
- `continuityC` and `continuityG`
  - Strings or flags indicating the continuity class of the curve in terms of “C0”, “C1”, “G1”, “G2”, etc. These are updated or verified by methods like `checkG1Continuous`, `checkG2Continuous`, `makeC1Continuous`, etc.
- `segmentLimits`
  - Stores the start and end parameter values (in u) for each segment, based on `segmentLengths`. For example, if the first segment has length 2, `segmentLimits(:,1)` might be [0; 2]. If the second is length 3, `segmentLimits(:,2)` might be [2; 5], etc.
- `totalSegments`
  - Number of piecewise segments in the spline/curve.
- `target_segment`
  - An index property indicating which segment is being used in certain evaluations (it’s updated inside `evaluate`, `evaluateX`, `evaluateZ`, etc.).
- `color`
  - A storage for color or color mapping in advanced routines (like `splineify`). Not essential for the curve geometry, but used for visualization or data marking.

The constructor function defines how the class is initialized, how control matrices are arranged, and how the piecewise structure is defined.

`function obj = Bezier2DCat(controlPoints, segmentLengths, weights)`

Its key steps are:

1. Initializes the main properties `controlPoints`, `segmentLengths`, `weights`.
2. Defines `bezierUnitBasis` for a cubic Bézier (the 4x4 matrix).
3. Builds or adjusts the `controlMatrix` to ensure each group of four rows corresponds to

one segment, enforcing at least C^0 continuity. If the original `controlPoints` array has more than 4 points, duplicates are inserted at boundaries.

4. Initializes `continuityC` as `"C0"`.
5. Builds the block-diagonal `bezierBasisMatrix`, placing `bezierUnitBasis` in the appropriate rows/columns for each segment.
6. Sums `segmentLengths` to get `totalLength`.
7. Sets up `segmentLimits` so each column has the start-end subinterval in the global parameter domain u.

Following are curve evaluation and derivatives, which are methods that evaluate the Bézier curve and its derivatives at a given u parameter.

- `valor = evaluate(obj,u)`
  - Finds which segment the input u belongs to by checking `segmentLimits`.
  - Computes a local parameter $t = (u - a)/(b - a)$, where $[a, b]$ is that segment's subrange.
  - Forms the vector $\begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix}$ in the correct location for that segment.
  - Multiplies by `bezierBasisMatrix` × `controlMatrix` to get the evaluated point $\{x, z\}$.
- `evaluateCartesian3D(obj,u,theta)`
  - Calls `evaluate(u)` to get a 2D point.
  - Interprets that point as $(x, 0, z)$ in 3D.
  - Builds a 2D rotation matrix about the z-axis by angle θ and applies it, returning a 3D coordinate.
- `evaluateDerivative(obj,u)`
  - Similar to `evaluate`, but uses the cubic derivative polynomials $\begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix}$.
  - Returns the first derivative $\nabla\mathcal{S}(u)$ in 2D.
- `evaluateDerivative2(obj,u)`
  - Computes the second derivative $\nabla^2\mathcal{S}(u)$ using the polynomial $\begin{pmatrix} 6t & 2 & 0 & 0 \end{pmatrix}$.
- Other related methods
  - `evaluateDerivativeCartesian3D(obj,u,theta)`: Embeds the first derivative in 3D after rotation.
  - `evaluateDerivative2Cartesian3D(obj,u,theta)`: Same for the second derivative.
  - `evaluateVH(obj,u)`: A method returning a vector orthogonal to the first derivative

$\|\nabla\mathcal{S}(u)\|\mathcal{N}$.

- `evaluateX(obj,u), evaluateZ(obj,u)`: Specialized evaluations returning only x- or z-coordinate.

Additionally, there are geometric/curve-related methods, which calculate tangent, normal, curvature, and their respective geometric interpretations.

- `evaluateTangent(obj,u)`
  - Computes the unit tangent vector $\mathcal{T}(u)$ by normalizing $\nabla\mathcal{S}$.
- `evaluateNormal(obj,u)`
  - Computes a 2D normal by rotating the tangent 90° degrees using
$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$
    .
- `evaluateCurvature(obj,u)`
  - Returns the scalar curvature
$$\kappa(u) = \frac{\det\left[\nabla\mathcal{S}(u)^T,\ \nabla^2\mathcal{S}(u)^T\right]}{\|\nabla\mathcal{S}(u)\|^3}$$
    .
  - Uses the determinant of $[\nabla\mathcal{S}(u)^T,\ \nabla^2\mathcal{S}(u)^T]$ and the norm of the first derivative cubed.
- `evaluateCurvatureVector(obj,u)`
  - Computes $\kappa(u)\,\mathcal{N}(u)$, i.e., curvature times the unit normal.
- `evaluateRadiusCurvature(obj,u)`
  - Returns the radius of curvature $\rho(u) = 1/\kappa(u)$.
- Other related methods
  - `evaluateRadiusCurvatureVector, Negative, PositiveH`: Return vectors in the normal direction scaled by the radius of curvature, with different sign/orientation conventions.
  - `evaluateNormalPositiveH(obj,u)`: A specialized method ensuring a consistently outward orientation for $\mathcal{N}(u)$.

Following are continuity checks and adjustments, which are methods that check or enforce G1, G2, and C1 continuity.

- `checkG1Continuous(obj,tol)`
  - Verifies G^1 continuity at each segment boundary by comparing unit tangents on both sides.

  - If the dot product is close to 1 within `tol`, it is G^1 continuous; otherwise it is not.
- `checkG2Continuous(obj,tol)`
  - Checks G^2 continuity by comparing curvatures at the boundary.
  - If the difference in curvature is within `tol`, it is G^2; otherwise it calls `checkG1Continuous`.
- `makeG1Continuous(obj,b1)`
  - Adjusts certain control points in `controlMatrix` so that the tangents match (ensuring G^1 continuity).
  - Uses a parameter `b1` to scale how control points are shifted.
- `makeG2Continuous(obj,b1,b2)`
  - Extends the G^1 adjustment to also match curvature (making it G^2 continuous).
  - Involves modifying even more control points near the boundary using `b1` and `b2`.
- `makeC1Continuous(obj)`
  - Internally calls `makeG1Continuous(1)` and labels the curve as `"C1"`.

Furthermore, there are visualization methods, used to plot the curve, its control polygon, derivatives, curvature, etc.

- `drawBezier(obj)`
  - Plots the entire piecewise Bézier curve by sampling each segment in small steps.
  - Uses `plot` with thicker lines and sets up axis labels, titles, etc.
- `drawControl(obj)`
  - Plots the control polygon and the control points (scatter plot).
- `drawDerivative(obj)` / `drawDerivative2(obj)`
  - Plots the first or second derivative maps by sampling each segment in many points.
- `curvatureComb(obj)`
  - For many sample points, draws a short line from the curve in the direction of the curvature vector (a “curvature comb”).
- `curvatureGraph(obj)`
  - Plots $\kappa(u)$ versus the parameter u.

There also exist length and parametrization methods for handling arc-length computation and solving for parameter u given a length.

- `arcLength(obj,ueval)`
  - Integrates $\|\nabla \mathcal{S}(u)\|$ from 0 to $u =$ `ueval`, returning the arc-length in that interval.
- `solveArcLength(obj,Leval)`

  - Solves for u_* such that the arc-length from 0 to u_* is `Leval`.
  - Uses `fsolve` on $-\texttt{Leval} + \int_0^{u_*} \|\nabla\mathcal{S}(\tau)\|\, d\tau = 0$.

Following are 3D rotation and homeomorphism methods which allow for the realization of the methodology in parametrized curves.

- `makeOrthogonalToRevAxis(obj,dx0,dxend)`
  - Modifies the first and last control points (or tangents) so that the curve is orthogonal to the revolution axis at start/end.
- `splineify(obj,L,H,T)`
  - Takes arrays `L, H, T`.
  - For each (L_i, H_j, θ_k):
    1. Finds u_* by matching arc-length $= L_i$.
    2. Evaluates the curve at u_*.
    3. Offsets the point by H_j in a direction orthogonal to $\nabla\mathcal{S}(u_*)$.
    4. Rotates by θ_k using $\texttt{rotz}(\theta_k)$.
  - Collects these points in a 3D set `P` and does alpha-shape plotting.
  - This is the inverse transformation.
- `desplineify` and related methods
  - Inverse approach: given (X, Y, Z) in 3D, solve for the corresponding (L, H, T) so that the curve plus normal offset plus rotation reproduces the given points.
  - Uses `fsolve`, `fminsearch`, or other numerical approaches to find best-fit parameters.
  - This is the forward transformation.

Lastly, there exist other specialized methods for geodesics which are listed here as well. These methods will be of use in Chapter 5.

- `metricTensorPlanar(obj,u,theta)`
  - `P = evaluate(u)` returns the point $\{x, z\}$. The code sets $\texttt{P2} = 0$ ($\mathcal{S}^y = 0$) to keep a 2D interpretation in a pseudo-3D structure.
  - `dP = evaluateDerivative(u)` gives $\{dx/du, dz/du\}$. The code sets $\texttt{dP2} = 0$ ($\dot{\mathcal{S}}^y$).
  - This routine forms a 2×2 metric tensor,

    $$g_{uu} \;=\; \dot{\mathcal{S}}^{x2} + \dot{\mathcal{S}}^{z2}, \qquad g_{\theta\theta} \;=\; \mathcal{S}^{x2}, \qquad g_{u\theta} = g_{\theta u} = 0,$$

    which treats u as one parameter and θ as another.
- `christoffelSymbolsPlanar(obj,u,theta)`

- The function first checks if $u < 0$. If so, it flips u and shifts θ by π. This is utilized to be able to generate continuous Christoffel symbols around the seams of $\{\varphi\}$.
- Calls `metricTensorPlanar(u,theta)` to get the metric $\mathbf{g}$, and then uses `pinv(g)` for its inverse (or pseudo-inverse), $\mathbf{g}^{-1}$.
- It evaluates the curve $\mathcal{S}(u)$ and its first/second derivatives $(\nabla\mathcal{S}, \nabla^2\mathcal{S})$, storing the components in `dP1, dP3` and `ddP1, ddP3`.
- The code then forms expressions for the Christoffel symbols Γ^i_{jk}. It places them in a 2×4 matrix in the order:

$$\begin{pmatrix} \Gamma^u_{uu} & \Gamma^u_{u\theta} & \Gamma^u_{\theta u} & \Gamma^u_{\theta\theta} \\ \Gamma^\theta_{uu} & \Gamma^\theta_{ut} & \Gamma^\theta_{tu} & \Gamma^\theta_{tt} \end{pmatrix}$$

With this toolset in place, it is now possible to slice solids with systematically generated embedded maps. These concepts are explained in the following Section.

4.3 Slicing parts

The previously outlined homeomorphism and toolset may be used in order to conformally (i.e., parallel to the build platform) and uniformly (i.e., with a constant slice height) slice a part. This Section covers all necessary steps to do so.

4.3.1 Methodology overview

Given a part (usually in a tessellated format such as STL) and a build platform (i.e., the base surface on top of which the AM process occurs), defined as a parametrized curve $\mathcal{S}$ in the $\pi \equiv -y = 0$ plane, it is possible to apply a slicing Workflow. This is part of the Offline Set-up procedure necessary for a RBNPAM manufacturing cell. This Offline Set-up consists of the following steps:

1. Transforming the part, usually designed in the Cartesian space $\{b\} \equiv \mathbb{R}^3$, onto the slicing space $\{\varphi\}$.
2. Planarly slice the $\{\varphi\}$-part with a commercial planar software, or with in-house slicing methods.
3. If the part was sliced using commercial software, parse and cleanse the resulting ISO code.
4. Transform the result back to $\{b\}$, and prepare the definition of robot poses. The final robot pose definition utilizes concepts from Chapters 6 and 7, and will be explained in Chapter 7.

A schematic summary of this methodology is represented in Figure 4.11.

The following Subsections aim to explain the procedure through the slicing of a simple part: a helical spring. This part is represented in Figure 4.12.

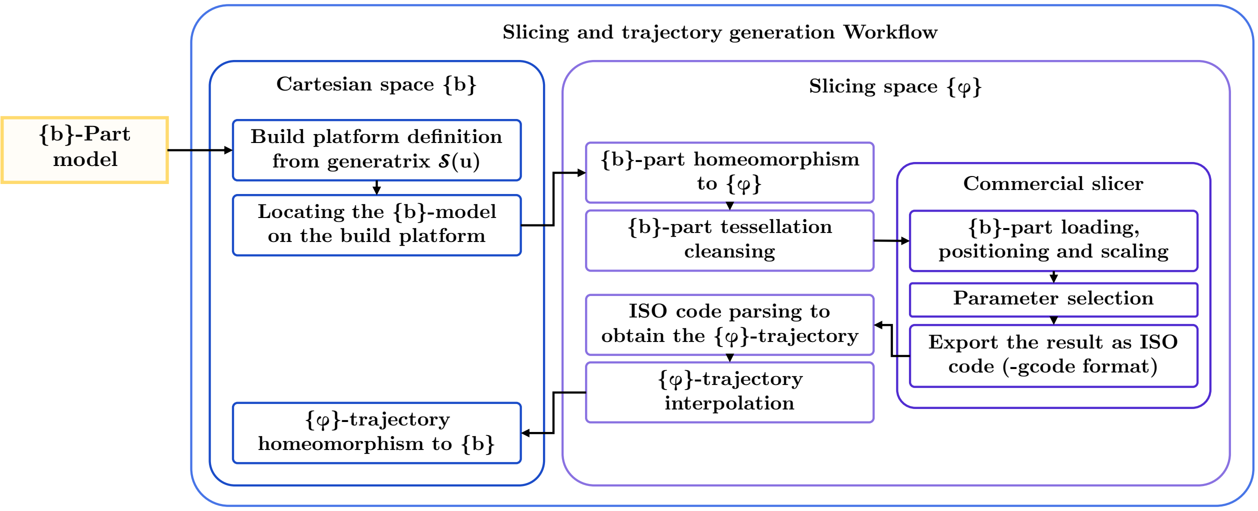


Figure 4.11: Schematic diagram of the part slicing Workflow, which is part of the Offline Set-up procedure.

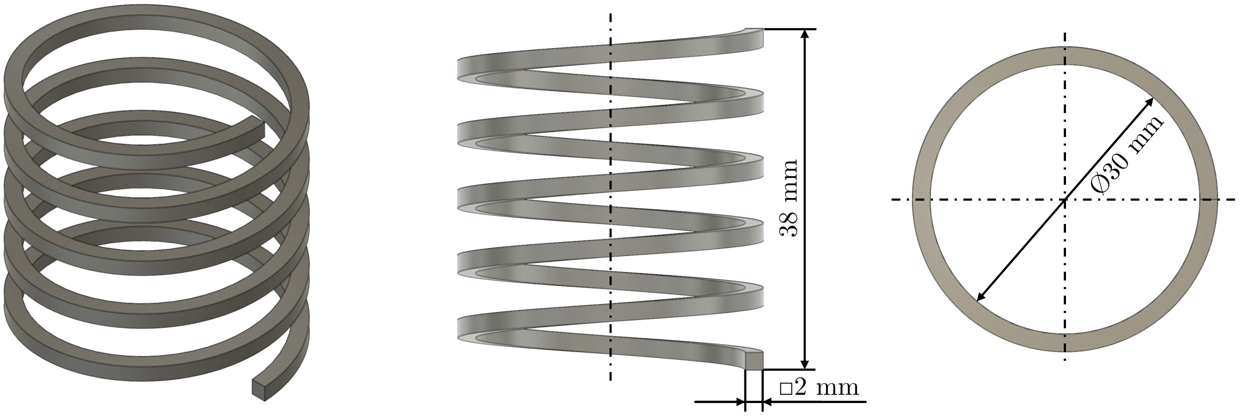


Figure 4.12: Helical spring with dimensions, part utilized for showcasing the present procedure.

4.3.2 Transforming the part onto the embedded map

The first step to homeomorph the part onto the slicing space is to define a suitable build platform. For the purposes of showcasing the present procedure, two different build platforms have been tested: a cylindrical build platform and a more complex "hourglass-shaped" build platform. These two build platform have been chosen to showcase the full procedure performance on a more well known build platform that provides easy-to-follow diagrams, and a general case build platform that includes several curve segments, as well as convex and concave regions.

The cylindrical build platform corresponds to the following cubic Bézier spline definition, with the control polygon outlined below. This control polygon produces a build platform in the shape of a cylinder, with radius equal to 15 [mm] and height equal to 50 [mm]. The generatrix $\mathcal{S}$, its control polygon $\mathbf{P}$, and the build platform ($\mathcal{M}$ represented in $\{b\}$) are represented in Figure 4.13. Of note, the spline is done through the composition of two curve segments, hence the matrix formulation for u.

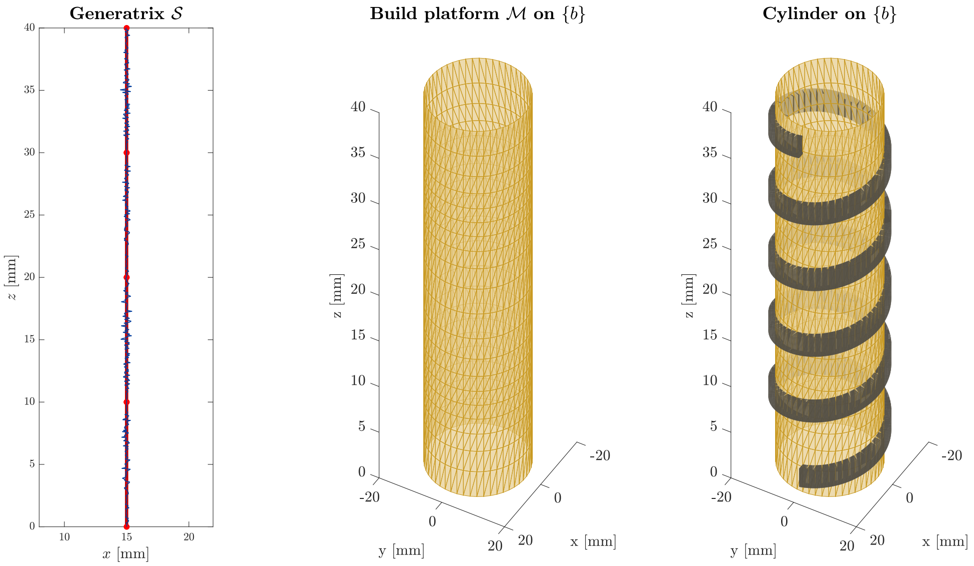


Figure 4.13: Generatrix $\mathcal{S}$, control polygon $\mathbf{P}$ (on top of the generatrix), and resulting build platform for the cylinder case.

$$\mathcal{S}(u) = \mathbf{uBP}$$

$$\mathbf{u} = \begin{pmatrix} u^3 & u^2 & u & 1 & & \cdots & & 0 \\ 0 & & \cdots & & (u-1)^3 & (u-1)^2 & (u-1) & 1 \end{pmatrix} \text{ when } \begin{cases} u \in [0,1) \\ u \in [1,2) \end{cases} \tag{4.13}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B_3}^{4\times4} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_3}^{4\times4} \end{pmatrix}^{16\times16} \tag{4.14}$$

$$\mathbf{P} = \begin{pmatrix} 15 & 40 \\ 15 & 30 \\ 15 & 30 \\ 15 & 20 \\ 15 & 20 \\ 15 & 10 \\ 15 & 10 \\ 15 & 0 \end{pmatrix} \tag{4.15}$$

Once the build platform has been defined and the part has been located on top of it (i.e., its exact position in the Cartesian space $\{b\}$, it is possible to perform the slicing procedure. It is important to note that the part is saved in a tessellated format (i.e., it is defined through a vertices matrix and a connectivity matrix indicating how these vertices form triangles). Therefore, to do the slicing procedure, each vertex of the part model mesh is forward transformed onto $\{\varphi\}$. The connectivity matrix, however, remains unchanged at this point. The result of the forward transformation is shown in Figure 4.14.

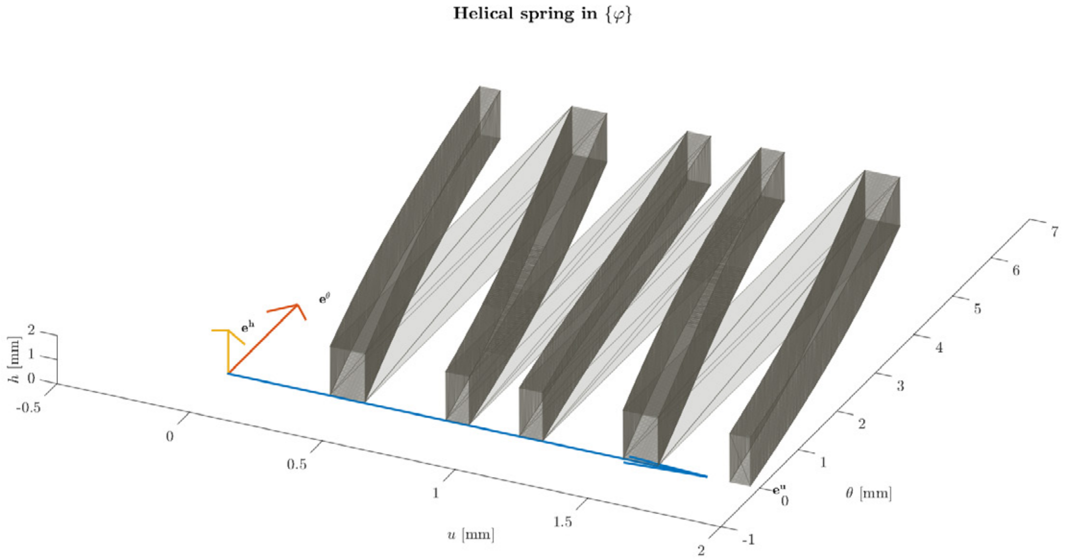


Figure 4.14: Forward-transformed homeomorphic model of the spring in the $\{\varphi\}$ space.

Each one of the thick grey-shaded regions represents a coil of the spring. Another interesting property of the homeomorphed model is the difference in shape between subsequent coils. This is due to the generatrix $\mathcal{S}$ definition. If the curve velocity $\nabla\mathcal{S}$ is not constant throughout the parameter domain, the homeomorphed model will display this behavior, where some regions appear slightly deformed (e.g., around $u = 0.5$ and $u = 1.5$).

A possible solution to this implies redefining the homeomorphism, and replacing the u coordinate for the L coordinate, which is the arc-length along $\mathcal{S}$. This is properly defined as:

$$L = \int_0^u \|\nabla \mathcal{S}(u)\| \; du \tag{4.16}$$

$$\mathbf{e_L} = \mathbf{e_u} \tag{4.17}$$

When this change is performed, the homeomorphism results in the shape represented in Figure 4.15. In this homeomorphism, the helical spring retains constant coil width and gap.

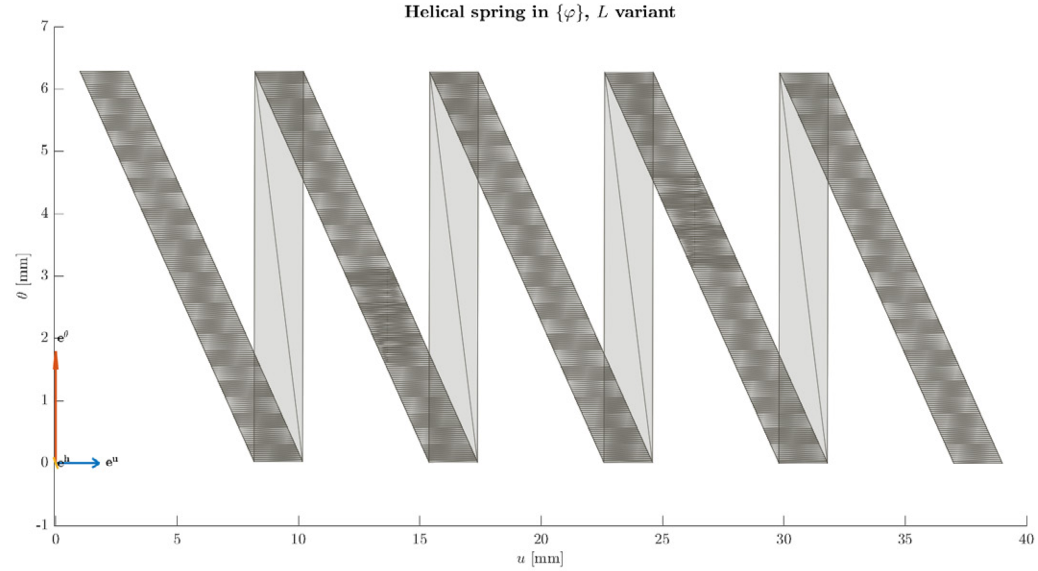


Figure 4.15: Forward-transformed homeomorphic model of the spring in the $\{\varphi\}$ space using the L homeomorphism variant.

However, this L homeomorphism variant has one critical drawback: if the curve's hodograph does not behave in a Pythagorean manner (i.e., $\|\{\dot{\mathcal{S}}^x, \dot{\mathcal{S}}^z\}\| = p(u)$, where p is a polynomial), then the homeomorphism is not analytically solvable. This severely hinders code run time, and puts into question the applicability of the present methods in an actual manufacturing Workflow. An extended discussion on this matter is provided in the Conclusions Chapter.

It is important to notice in Figures 4.14 or 4.15 the appearance of some grey shaded areas in the homeomorphed model, between the beginning and end of each coil. These areas are due to the θ-wrapping of the model. The embedded map $\{\varphi\}$ defined is not continuous in the θ dimension, as this coordinate is naturally constrained to span the range $\theta \in [0, 2\pi)$. When the forward transformation is performed, the algorithm looks for solutions in this θ range, as there is no other criteria for expanding the solution domain. This issue can be solved in two separate ways.

Firstly, and with an increase of generality, the present methodology may account for helical symmetry slicing. If a smaller build platform is defined (with a Cartesian length equal to the helical spring pitch), the $\mathbf{e}_{\boldsymbol{\theta}}$ definition may be extended to include a screw axis in the exponent such that:

$$\mathbf{T}_{\{\mathbf{b}\}} = e^{[\mathcal{E}]\theta}\mathbf{T}_{\{\varphi\}} \tag{4.18}$$

$$\text{where } \mathcal{E} = \begin{pmatrix} \mathcal{E}_{\omega}^{3\times 1} \\ \mathcal{E}_{v}^{3\times 1} \end{pmatrix}, \mathcal{E}_{\omega} = \mathbf{s}, \mathcal{E}_{v} = [-\mathbf{s}]\mathbf{q} + p\mathbf{s} \tag{4.19}$$

Where $\mathbf{s}$ is the axis of screw rotation, $\mathbf{q}$ is the distance between the axis of screw rotation and the origin, and p is the pitch of the helix. Without loss of generality, $\mathbf{s}$ may be equal to the $\mathbf{e^z}$ axis and $\mathbf{q} = \mathbf{0}$. This solution is experimental at the time of writing the present Thesis, and will be discussed briefly in the Conclusions Chapter.

A different approach consists of disregarding these connections by removing these triangles. This is the approach selected moving forward. This triangulation cleansing comes at the cost of generating a seam along the θ-axis in $\{b\}$, since the faces in the vicinities of $\theta = 0$ and $\theta = 2\pi$ will be disconnected from each other. This θ-seam aligns with the actual standard behavior of robotic manipulators. Advancing some ideas from the Environment proposed in Chapter 6, the build platform is to be attached to the end effector of the robotic manipulator. Therefore, the last robot axis θ_5 (sometimes called $wrist_3$), is aligned with $\mathbf{e_{\boldsymbol{\theta}}}$. θ_5 is usually constrained to a range, delimited by the joint limits. In the case of the UR10, θ_5 is restricted to the range $[0, 4\pi]$. Thus, even if the slicing method allowed for a limitless θ range, the robotic manipulator as is would be constrained to a θ range. Both issues could be combined by developing helical slicing, as well as actuated build platforms with an additional Degree of Freedom which allows for complete, limitless θ_6 movement.

Removing the wrapping-around triangles in Figure 4.14 results in the shape shown in Figure 4.16.

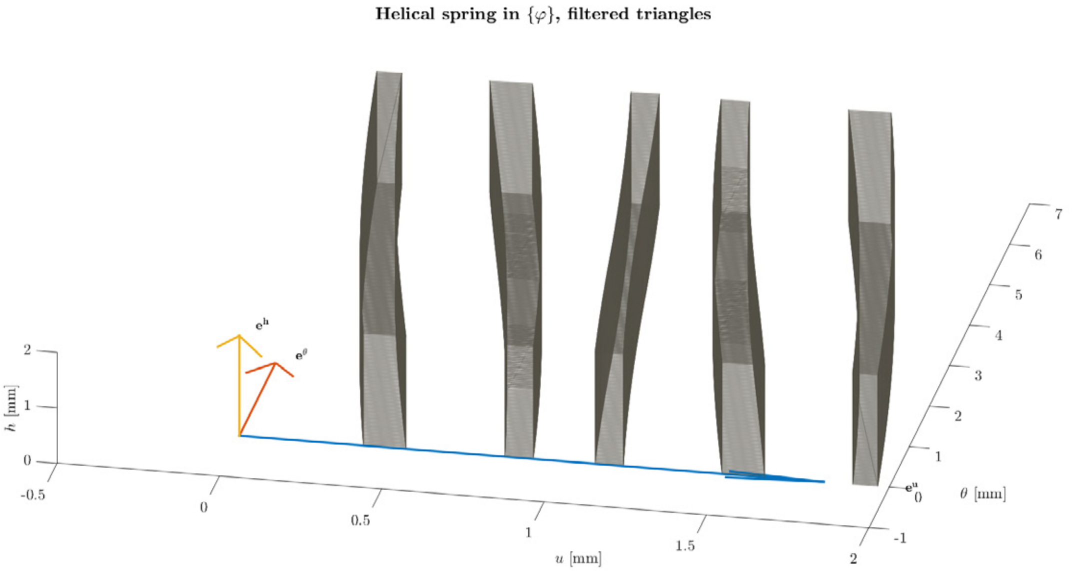


Figure 4.16: Forward-transformed homeomorphic model of the spring in the $\{\varphi\}$ space, with the θ-seam resulting from removing the wrapping-around triangles.

This new model is now ready to be planarly sliced using commercial software or in-house software. In the next Section, commercial slicing will be explored.

4.3.3 Planar slicing with commercial software

Once the model is homeomorphed and the connectivity matrix is cleansed of wrapping-around triangles, it is time to generate a trajectory through $\{\varphi\}$-planar commercial slicing.

The software utilized for this task is Ultimaker Cura version 4.11, although any commercial slicer would work for the purposes of this Workflow. The steps followed for slicing are the following:

1. Set up the software, selecting a kinematic architecture. In this case, the kinematic architecture of an Ultimaker S5 machine was selected, owing to the fact that the Manufacturing Technology Laboratory owns one, and its kinematic architecture is therefore well known. This AM machine has a build volume of x=330 mm × y=240 mm × z=300 mm.
2. Load the $\{\varphi\}$-model into the software.
3. Optionally, if the $\{\varphi\}$-part is small, it is possible to scale it. This is a common occurrence, since the x coordinate only spans the range $u = [0, u_{max}]$, usually small, and the y coordinate spans the range $\theta = [0, 2\pi]$ as previously mentioned. These measurements are either dimensionless or in radians, however, Cura interprets these as either millimeters or inches, depending on user configuration. Either case, they result in small parts which are not very suitable for slicing and need to be scaled.
4. Locate the part in the middle of the build platform. This corresponds to the point $\{x, y, z\} = \{165, 120, 0\}$ [mm].
5. Select process parameters for MEX-P RBNPAM. These include:
    - Slice Height: This is the fundamental parameter for model slicing, and sets the distance between subsequent slices. This parameter is often related to the part geometric quality, and is intimately related to the staircase effect.
    - Line Width: This parameter, when combined with the slice height and the relative speed between the extruder and the build platform, sets the volumetric flow of feedstock material exiting from the nozzle. This line width will have profound implications in Chapter 5.
    - Shell Thickness: Controls how thick the outer walls (perimeters) are. Typically set in multiples of the nozzle diameter (e.g., 0.8 [mm] for a 0.4 [mm] nozzle with two perimeters). Thicker walls generally increase strength and rigidity.
    - Top/Bottom Thickness: Determines how many layers of solid plastic are laid down on the top and bottom surfaces. More top layers help avoid gaps on the surface, and enough bottom layers ensure a solid foundation.
    - Infill Density: Specifies the percentage of the internal structure that is filled (e.g., 20%, 50%). Higher densities increase strength (and weight), while lower densities

reduce print time and material usage. This parameter is intimately related to Infill Pattern, which is outlined below. Planar infilling performed by Cura has several limitations (described in Subsection 4.4), and therefore, this method can only accommodate for 100% infill density.

- Infill Pattern: Defines the internal geometry (e.g., grid, gyroid, triangle, lines). Each pattern offers different trade-offs in strength, speed, and material consumption. This parameter is irrelevant for the purposes of the present Thesis, as these methods are specifically made for planar infilling. Chapter 5 will treat this problem extensively.
- Extruder Speed: The speed at which the extruder head moves while extruding, or in this RBNPAM context, the relative linear speed between the build platform and the extruding end. Faster speeds shorten build cycle time but can harm quality; slower speeds improve detail but take longer.
- Travel Speed: The speed of the extruder head when it is not extruding. Higher travel speeds can reduce overall build cycle time but may cause vibration and undesired robot behavior if set too high.
- Printing Temperature: The temperature at which the filament is melted and extruded. Different materials require different ranges (e.g., 190–210°C for PLA, 230–250°C for ABS). Proper temperature ensures good layer bonding. This parameter is not relevant for path planning, but will come into play in extruder control.
- Build Platform Temperature: The heated bed temperature, which improves adhesion and prevents warping. Common settings are about 60°C for PLA and 80–110°C for ABS. This parameter is not relevant for path planning, although it will is utilized for build platform control within the software architecture.
- Flow (Extrusion Multiplier): Adjusts the volumetric flow rate of filament extruded. A slightly higher flow can fix under-extrusion; a lower flow may resolve over-extrusion. Must be calibrated for dimensional accuracy. This parameter is not relevant for path planning, although it will is utilized for extruder control.
- Cooling (Fan Speed): Controls the part cooling fan. Critical for materials like PLA to reduce stringing and improve detail. Too much cooling for materials like ABS can lead to warping or poor layer adhesion. This parameter is not relevant for path planning, although it will is utilized for extrusion control within the software architecture.
- Support Settings: Define how support material is generated (e.g., everywhere or only on the build plate), the overhang angle at which supports appear, and the density and pattern of the supports. These factors influence how easily supports can be removed and how well they support overhangs. Supports may be needed if the part does not touch completely the build platform, to generate a scaffold for the part proper.

- Retraction Distance: The amount the filament is pulled back to prevent oozing or stringing during travel moves. Too little retraction leads to strings; too much can cause clogs or under-extrusion. This parameter is not relevant for path planning, although it will influence the extruder control.
- Retraction Speed: The speed at which the filament is retracted. Higher speeds can reduce stringing but may stress the extruder or filament. For path planning, this is not relevant yet, although it will influence the extruder control.
- Combing Mode: Manages travel paths to avoid crossing open areas (reducing stringing). Options include Off, All, or Within Infill to keep travel moves inside infill regions. This parameter is optional for a preliminary manufacturing, and it is left at the default value (All).
- Build Platform Adhesion Type: Methods (e.g., Skirt, Brim, Raft) to improve part adhesion. In this type of planning, this parameter is set to None, as there may be risk of overcoming the homeomorphism limits with the additional geometries generated.
- Acceleration and Jerk Settings (Advanced): Control how quickly the print head accelerates or changes direction. Lower values can improve quality by reducing ringing or ghosting, at the expense of print time. These parameters are set to the default values for the time being.

6. Slice the model and check for any errors.
7. Export the result in .gcode format. This .gcode format is based on ISO coding, according to standard ISO 6983 [106].

Following is an example of this slicing procedure on the helical spring. Firstly, the $\{\varphi\}$-model is loaded in Cura with the Ultimaker S5 machine kinematic architecture preset, located in the middle of the build platform, and then scaled to ten times its original size (Figure 4.17).

The slicing parameters are then selected. In this particular case, the parameters changed from the default profile and their values are shown in Table 4.1.

Slicing parameter	**Value**
Layer Height	0.5 [mm]
Line Width	0.35 [mm]
Shell Thickness	1 [mm]
Top/Bottom Thickness	1 [mm]
Infill Density	100 [%] Infill Pattern
Solid	
Adhesion Method	None

Table 4.1: Slicing parameters modified for the present example.

This parameter selection results in the slicing and infilling represented in Figure 4.18. Of note, the red trajectories indicate the outer shell rasters, the green lines represent the inner

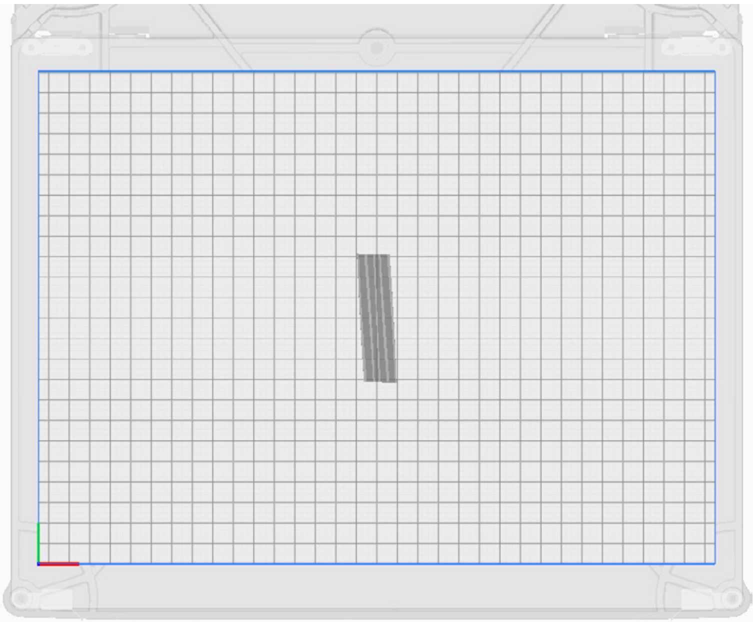


Figure 4.17: $\{\varphi\}$-model loaded in Cura.

shell rasters, and the yellow lines represent the uppermost shell rasters. Lastly, the lilac lines represent travel moves, which are machine movements between unconnected points of the trajectory. During these travel moves, no extrusion is done.

This slicing is exported as an ISO code (.gcode format) for its transformation back in the MATLAB engine developed.

4.3.4 ISO code parsing and cleansing

The ISO code obtained from the slicing software needs to be analyzed for its homeomorphism back to the $\{b\}$ space. Below is an explanation of the steps performed in order to parse the code, obtain the relevant information (machine positions, velocities, extruder and build platform temperatures, and so on), and represent the $\{\varphi\}$-model after slicing.

The parsing script begins with initialization. This includes:

- Clearing all variables from the workspace, closing any open figures, and clearing the command window.
- `D = readlines('filename.gcode')` reads each line of the ISO code file into a string array called `D`.
- `eof = numel(D)` determines the total number of lines in the G-code file.

Once the code is properly loaded, it is time to separate the different blocks of the ISO code. To do so, some markers set by Cura are identified within the code.

- `ixMESHNonMesh` is set by using `find(contains(D,';MESH:NONMESH'))`, identifying where the G-code contains “`;MESH:NONMESH`”.

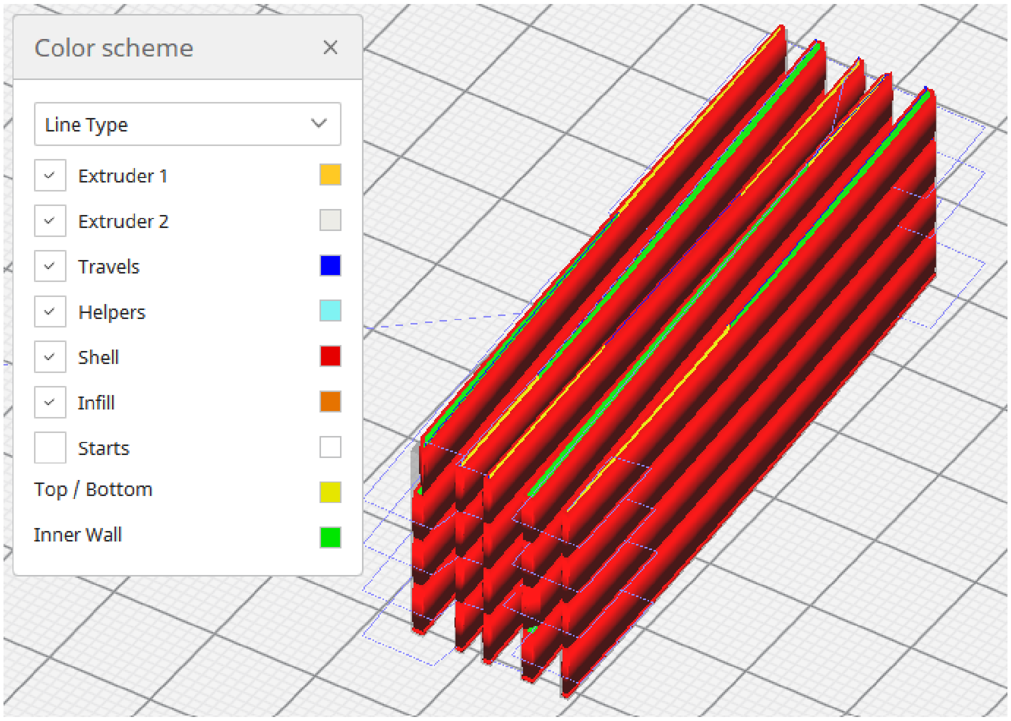


Figure 4.18: $\{\varphi\}$-model after slicing and infilling.

- `ixTYPE` is set by looking for “`;TYPE:WALL-OUTER`”.
- `ixMESH` is set by looking for “`;MESH:filename.stl`”.

These indices help locate key sections in the file where certain content or commands are expected, namely movement and temperature commands. The next step is extracting the values of the z-coordinate (in fact, the h-coordinate) for all slices.

1. A variable `Zstart` is set to the layer height value, which consequently is the initial z-height.
2. The code then identifies the lines immediately after each “`;MESH:NONMESH`” marker. Those lines contain new z values, indicating the next slice.
3. It compiles those candidate line indices into `ixZ_select`.
4. A regular expression (the result of `sprintf(’s+%c([+-]?d+.?d*)’,’Z’)`) is used to parse out the numeric value following the letter `Z`. These values are stored in an array `Z`.
5. `Zstart` is added at the beginning of the array, making `Z(1)=0.2` (the first layer’s height).

After extracting the z/h-coordinate values, the parser then extracts the x/u and y/θ values.

1. The script loops through each `;TYPE:WALL-OUTER` index in `ixTYPE`.
2. For each section, lines are selected until the next `;MESH:NONMESH` marker or end-of-file.
3. It ignores lines containing `M205`, which indicate jerk values and not actual xy coordinates

through a regular expression (the result of `sprintf(’s+(?!.*M205)%c([+-]?d+.?d*)’, ’XY’)`).

4. Another regular expression extracts `X` and `Y` values, ignoring any lines with `M205`.
5. It finds the appropriate Z index (from `ixZ_select`) that corresponds to this portion of the G-code, so each `(X, Y)` can be paired with the correct `Z` layer value.
6. The `(X, Y, Z)` data is appended into a matrix `XYZ`.

Now that all $\{u_{scaled}, \theta_{scaled}, h_{scaled}\}$ are extracted, it is time to undo the scaling, as well as the location of the part in the middle of the build platform, to constrain the results back to the axes original ranges. This is done in the following way:

- For the x/u-coordinate, `XYZ(:,1) = XYZ(:,1) - xc + Gx_STL` shifts all X-coordinates by `-xc + Gx_STL`, where `Gx_STL` is the STL $\{\varphi\}$-model barycenter.
- Similar shifts are performed on `Y`, including subtracting `yc` and adding π.
- Both X and Y are then scaled by the applied factor in Cura. In the case of the helical spring, this factor is 10 as previously established.

Lastly, the script provides a visualization of the results to confirm their validity. The trajectory is visualized through a gradient, which transitions from blue (the beginning of the trajectory) to red (the end of the trajectory). The visualization for the helical spring case is represented in Figure 4.19.

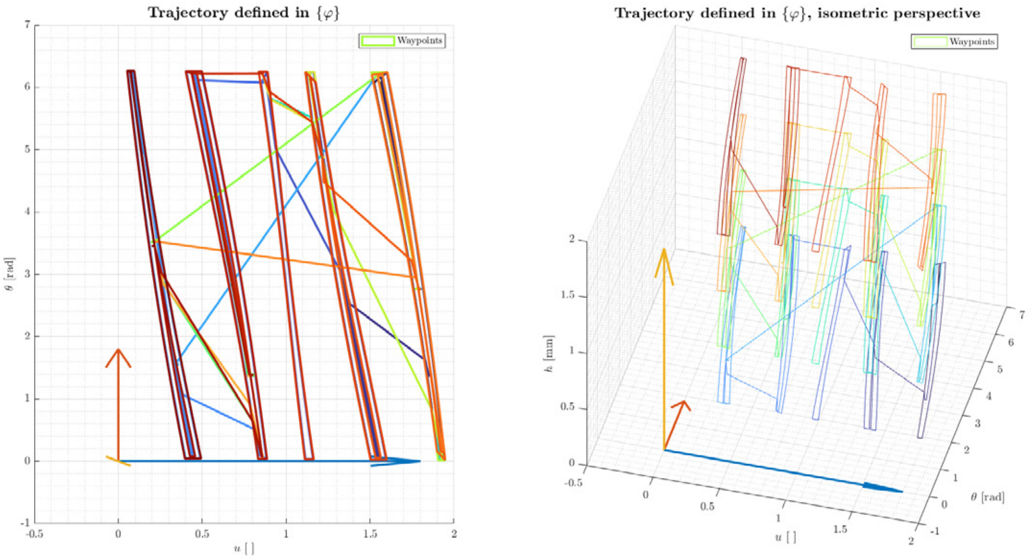


Figure 4.19: $\{\varphi\}$-trajectory after slicing and parsing. The axes showing at the bottom of each grid represent the trihedron $\{\mathbf{e_u}, \mathbf{e_\theta}, \mathbf{e_h}\}$.

The $\{\varphi\}$-trajectory as is present a challenge. Since the commercial slicer is made for planar slicing, in an Euclidean context, the shortest path between two points is a straight line. This is no longer the case in the Riemannian build platforms defined in this Chapter. Therefore, when the commercial slicer plans a raster between two points at a certain distance in the θ-axis, it plans it as a straight line. When the build platform and the $\{\varphi\}$-trajectories are homeomorphed back to the Cartesian space $\{b\}$, this introduces a chord error. This is exemplified in Figure 4.20, where a θ-wise raster is homeomorphed back into a cylindrical build platform. This problem can be solved in two separate ways.

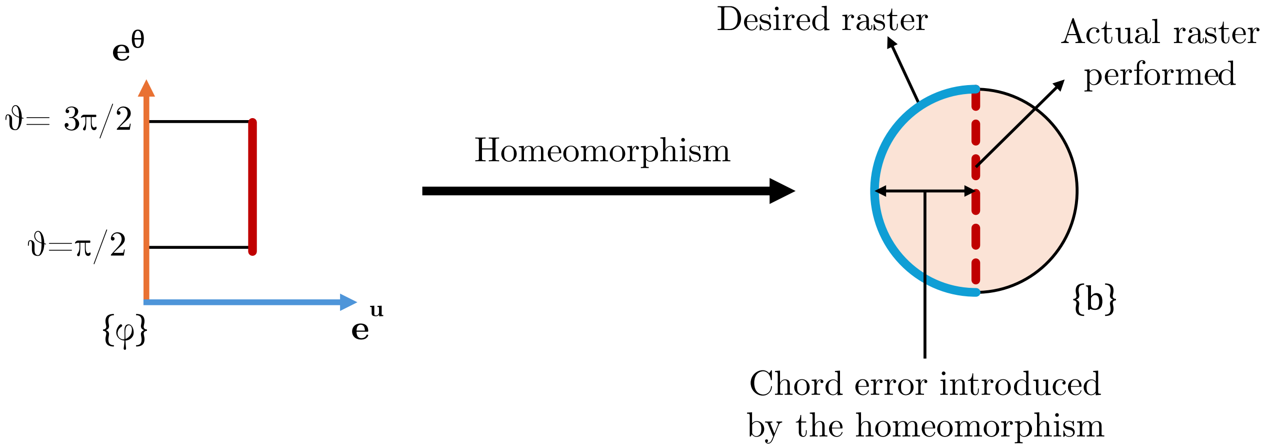


Figure 4.20: Chord error arising from lack of geodesic pathing in $\{b\}$.

The first solution leaves the $\{\varphi\}$-trajectory as is, but at the time of homeomorphing it back to $\{b\}$, the distance between points is no longer a straight line, but a geodesic path (i.e., the shortest distance between points located on a manifold). This idea is explored further in Chapter 5. A second, simpler solution, involves introducing interpolating points between long θ-wise rasters. This concept has been implemented, with the amount of interpolating points being a parameter selected by the user. In the case of the helical spring, a value of 5 points per raster has been used. However, it is possible to introduce a variable number of points per raster, depending on the raster length. The results of the interpolation are shown in Figure 4.21. In $\{\varphi\}$, this new trajectory appears to be no different from the previous, uninterpolated one. However, the difference will be clear when homeomorphed back to $\{b\}$ in the next Subsection.

4.3.5 Transforming the slices back to the Cartesian space towards the definition of robot poses

When the $\{\varphi\}$-trajectory is obtained, the final step is performing the homeomorphism to obtain the $\{b\}$-trajectory. To do so, the inverse transformation is used. This procedure is consistent, as per definition, the inverse transformation is injective throughout the $\{\varphi\}$ domain. Thus, it is always possible to map the $\{\varphi\}$-trajectory back to the Cartesian space $\{b\}$. The results of the homeomorphism for the uninterpolated and interpolated $\{\varphi\}$-trajectories are shown in Figures 4.22 and 4.23, respectively.

The introduction and mitigation of chord error is demonstrated through Figures 4.22 and 4.23. However, introducing interpolation points comes at compromise with algorithm runtime. For reference, the uninterpolated trajectory has a runtime of $t = 1.23$ [s] for $N = 1634$ points, whereas the interpolated trajectory has a runtime of $t = 2.68$ [s] for $N = 9798$ points.

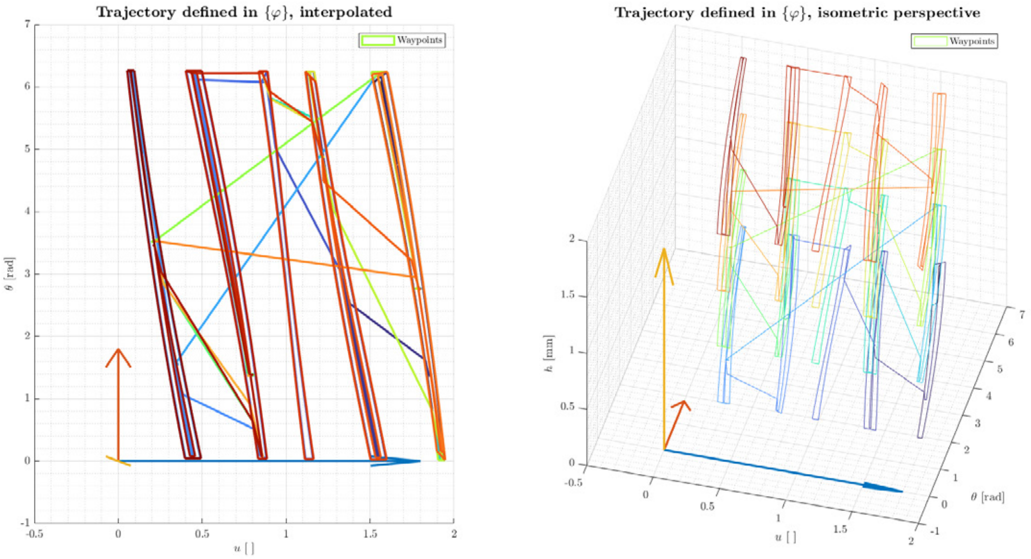


Figure 4.21: $\{\varphi\}$-trajectory after slicing, parsing, and interpolating.

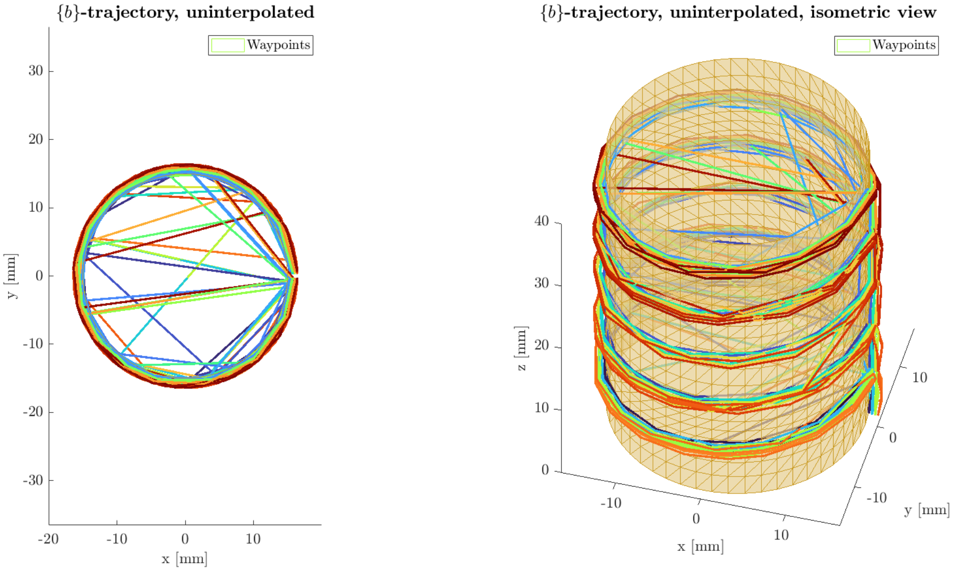


Figure 4.22: Uninterpolated $\{b\}$-trajectory after the homeomorphism.

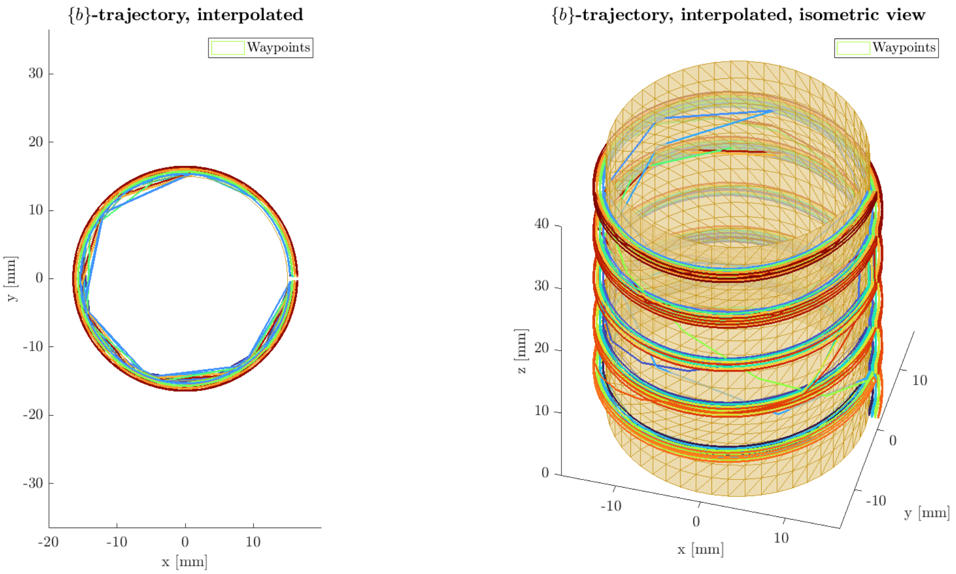


Figure 4.23: Interpolated $\{b\}$-trajectory after the homeomorphism.

It is possible to confirm that the postulated trajectory successfully replicates the original helical spring. Figure 4.24 represents the trajectory, overlaid with the original helical spring $\{b\}$-model object of study in a grey shade. The extruding trajectory follows the geometry of the spring faithfully, and the rasters falling outside of the $\{b\}$-model correspond to non-extruding travel moves. Additionally, due to the limitations of axisymmetric slicing, the θ-seam is present, and can be seen in the Figure.

Lastly, it is important to note that the $\{b\}$-trajectory obtained through the present method only indicates the extruding locations with respect to the part reference frame (i.e., the end-effector frame), which may not be identical to the forward kinematics of the robot through the Power of Exponentials formula. To relate the end-effector to the robot kinematic architecture it is possible to apply a rigid homogeneous transformation between these two frames. However, this rigid transformation is not invariant, as it heavily depends on end-effector assembly and manufacturing precision. Therefore, this $\{b\}$-trajectory cannot be translated into robot poses in a straightforward manner. This is one of the objectives posed for Chapter 7, and requires an intrinsic calibration.

4.4 Discussion

Using a commercial slicer provides no drawbacks if the part is desired to be solid, as the infilling algorithms only have to account for total slice geometry and completely fill it in. This action is performed the same way in Euclidean and Non-Euclidean contexts, and has been applied extensively in recent NPAM works.

However, using a commercial slicing solution has an inherent flaw: the in-layer non-solid infilling solutions are tailor-made for regular Cartesian contexts, i.e., Planar AM. This is due

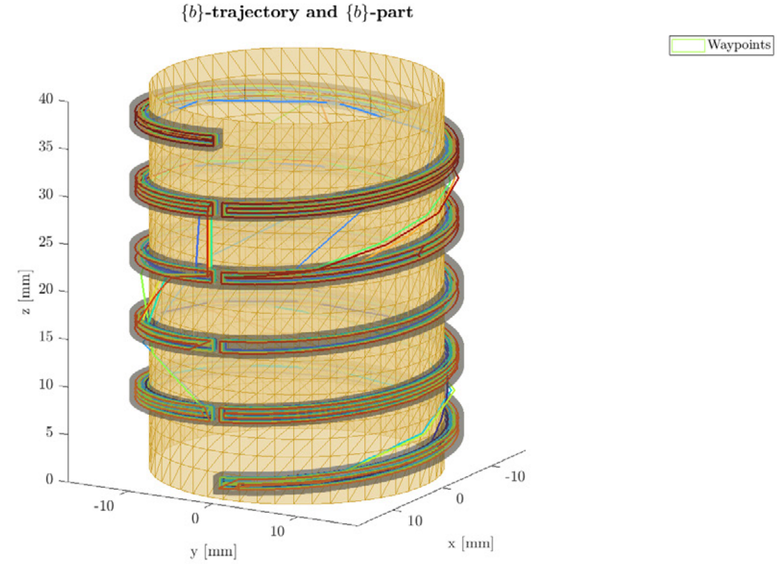


Figure 4.24: Interpolated $\{b\}$-trajectory after the homeomorphism, with the helical spring $\{b\}$-model overlaid on it.

to the fact that the algorithm proposes points for infilling in an Euclidean fashion, and does not take into account the intrinsic curvature of the slice manifolds generated. This can be seen clearly through the understanding of the raster width, which is a critical parameter in MEX-P AM processes. Generally, the raster width (e) is a constant parameter, and is the result of the relationship between the volumetric flow through the nozzle ($\dot{V}$), the relative speed between the nozzle and the build platform (ω_{bp}), and the slice height (Δh). This relationship is as follows:

$$e = \frac{\dot{V}}{\omega_{bp}\Delta h} \tag{4.20}$$

In Euclidean contexts, $\{\varphi\} = \{\mathbf{e^x}, \mathbf{e^y}, \mathbf{e^z}\}$. These basis vectors remain invariant through the homeomorphism, as they also properly define an ambient Cartesian space $\mathbb{R}^3$. In turn, the metric tensor throughout the domain $\mathbb{R}^3$ is $\mathbb{1}^{3\times3}$ in matrix form, the identity; and the Christoffel symbols of this sort of embedding are all equal to zero for the whole span of $\mathbb{R}^3$. This may look like a very convoluted way of saying the Cartesian space has no intrinsic curvature, but in a sense, the existence of intrinsic Gaussian curvature greatly affects the raster width. Depending on the type of curvature, given a raster deposited in $\{b\}$ with constant width, when homeomorphed onto $\{\varphi\}$, this raster width may apparently shrink (if the intrinsic Gaussian curvature is monotonically increasing), expand (if the intrinsic Gaussian curvature is monotonically decreasing), or remain as is (if the intrinsic Gaussian curvature is constant).

This poses two challenges. First, if an Euclidean non-solid infill method is applied onto a single layer, the local infill density varies as per the concepts previously outlined. This hinders the mechanical performance of the part, as there may be infill-abundant regions (not very optimal in terms of build cycle time, cost, or energy efficiency but functional nonetheless, denoted by concave, less than zero curvature), and infill-deficient regions in zones with high (i.e., convex) Gaussian curvature. This does not compromise the feasibility of the process, but provides poor performance parts, which is not ideal.

The second challenge, however, does compromise the feasibility of the process. The axisymmetric slicing method postulated hinges on the fact that the revolution maintains $\mathbf{e}^{\boldsymbol{\theta}}$ over a displacement in u or h. The intrinsic curvature along the revolution is therefore always positive. This implies that two points sitting on the same $\{u, h\}$ coordinate with different θ values tend to distance apart from each other when their h coordinate is increased. A visualization is represented in Figure 4.25, where the two arc lengths resulting from the forward homeomorphism show the effect of the convex curvature along θ.

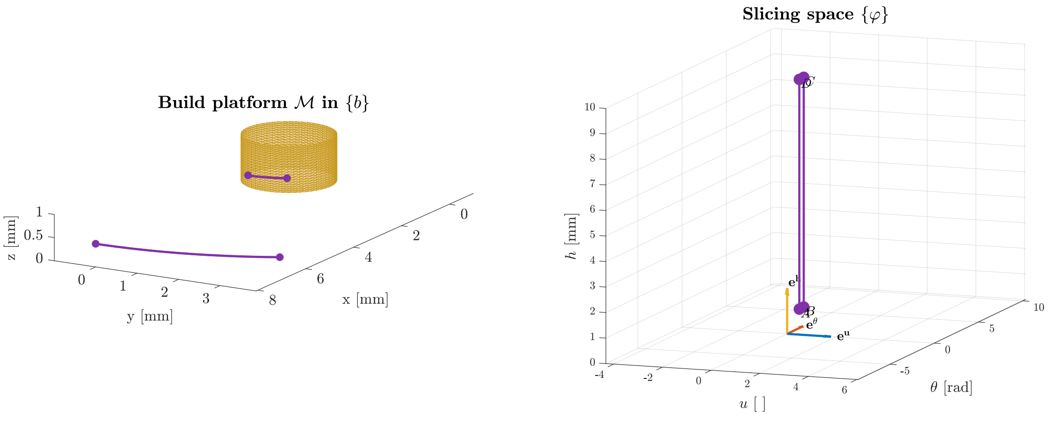


Figure 4.25: Points with the same $\{u, h\}$ coordinates when their h coordinate is increased. $\{b\}$ in the left-hand side and $\{\varphi\}$ on the right-hand side.

When utilizing an Euclidean infilling method, this phenomenon leads to increasingly deficient infilling towards the outer region of the part, culminating with the smallest infill density in the h-most zone. This poses a threat to the process feasibility. MEX-P parts are usually made of an outer shell and a hollow infill that supports this outer shell. In this situation, the outer shell may fail. This is due to the infill becoming deficient, to the point that it may not be able to properly support the outer shell. This situation occurs if the infill density is sparse enough that the mean distance between infill rasters is greater than the maximum bridging distance. This maximum bridging distance parameter arises from a combination of the thermal conditions regarding extrusion: material type, extrusion temperature, build volume temperature, and cooling performance.

A methodology to solve these problems has been developed using Riemannian geometry on the homeomorphism, and is presented in the following Chapter.

Chapter 5

Generating non-solid conformal uniform homogeneous-density infills

This Chapter covers the creation of non-solid infill paths for NPAM with homogeneous density, using the embedded map method for axisymmetric slicing. The first step involves creating sufficiently smooth axisymmetric platforms, which is critically important for the convergence of the geodesic equation on manifolds (i.e., the slice geometry, Section 5.1). Once this is done, for each slice, a uniformly-distributed mesh of points is required. This is accomplished by defining a geodesic repulsion energy minimization algorithm that uniformly distributes a number of points based on the command density and line width (Section 5.2). This algorithm is then further refined through several numerical optimizations (Section 5.3), which help ensure its applicability and feasibility for NPAM workflows. For illustrative purposes, a path planning algorithm is next presented to demonstrate the workflow's potential (Section 5.4). Afterwards, to validate the homogeneous density hypothesis, various metrics are implemented and examined (Section 5.5). Finally, all of these concepts are applied in the resolution of three case studies on two different build platforms (Section 5.6). An illustrative diagram of the methodology is shown in Figure 5.1. These concepts have been adapted from a journal publication written by the present author [107].

5.1 Defining a sufficiently smooth build platform

The first step for quasi-homogeneous density hollow infill path generation is the build platform definition. In the previous Chapter, the revolution of generatrices through the definition of an embedded map was proposed (5.1), where the Cartesian coordinates $\{x, y, z\}$ of the build platform are expressed by the parameters $\{u, \theta\}$:

$$\mathbf{r}_{\mathcal{M}}(u,\theta) = x(u,\theta)\mathbf{e}^{\mathbf{x}} + y(u,\theta)\mathbf{e}^{\mathbf{y}} + z(u,\theta)\mathbf{e}^{\mathbf{z}}. \tag{5.1}$$

In that earlier approach, the generatrices were defined by cubic Bézier spline compositions, where u is the spline parameter and θ is the angle of revolution. Nonetheless, the idea was that any parametrized curve fulfilling specific conditions on continuity, tangential constraints,

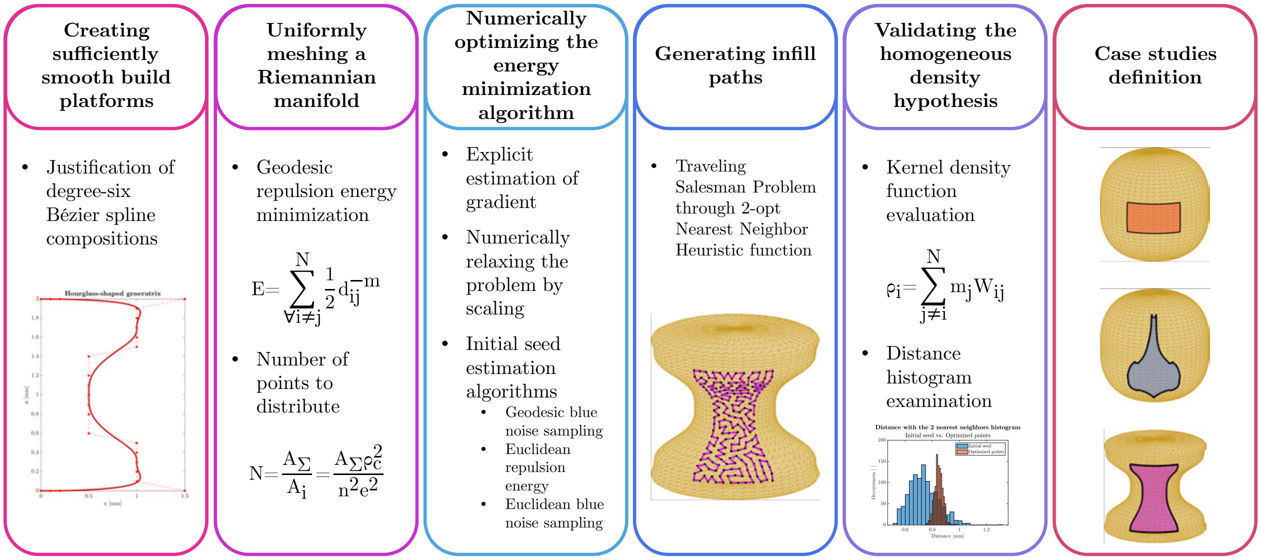


Figure 5.1: Schematic diagram of the methodology.

and complexity would be suitable for the embedded map methods. Following this principle, the methodology becomes more robust if the build platform is a Riemannian manifold–a manifold whose curvature is differentiable throughout its surface. Generating such a manifold by revolving parametric curves requires additional conditions on curve complexity (i.e., no cusps or self-intersections), C^3 continuity at spline joints, and perpendicular alignment with null curvature at the revolution axis.

First, regarding complexity, the parametric curve must not exhibit self-intersections; if it is a spline composition, the control polygons should not overlap. Second, in terms of continuity, the parametric curve should be C^3-continuous at any knot, guaranteeing a differentiable curvature on the surface. For spline compositions, this condition constrains four degrees of freedom (DoF) of each control polygon in the composition except for the first. Lastly, concerning the revolution axis, the curve should meet the axis perpendicularly and with zero curvature so that the manifold poles are also smooth. In spline compositions, this marginally restricts three DoF of the first control polygon and three DoF of the last, ensuring they connect perpendicularly to the axis. Figure 5.2 shows several examples of generatrices and their corresponding build platforms, highlighting whether they meet or fail these criteria. As noted, a locally smooth manifold may suffice, provided the algorithm remains confined to the region where the curvature is differentiable.

In the previous Chapter, cubic Bézier splines were employed with a C^0 continuity constraint, thanks to their simplicity. Here, degree-six Bézier splines are used instead, featuring seven DoF per segment and thus meeting continuity and axis-connection requirements of a globally adequate generatrix. Consequently, the surfaces remain Riemannian upon revolution, and maintain compatibility with the general embedded map strategies from the previous Chapter.

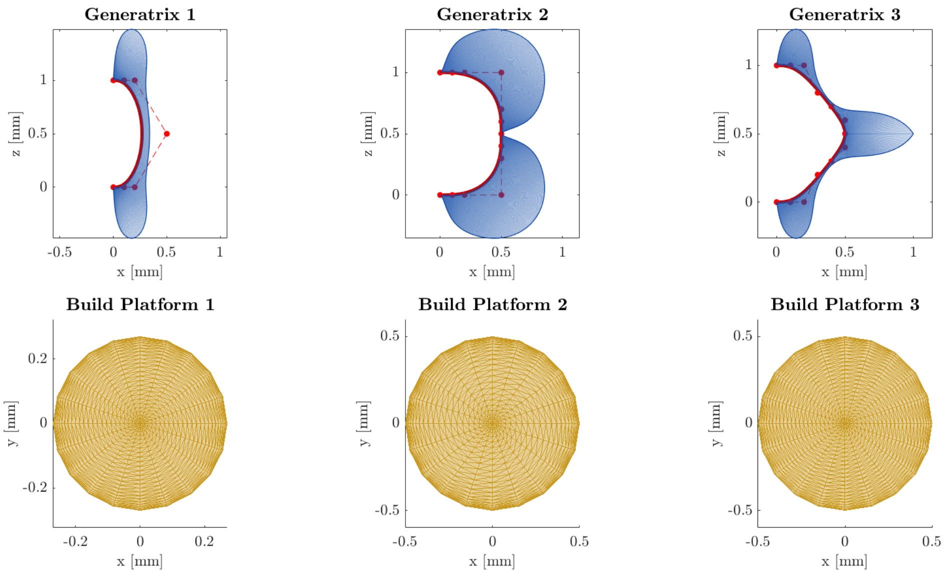


Figure 5.2: Examples of two globally adequate platforms (Build Platforms 1 and 2) and a globally inadequate platform (Build Platform 3), since its curvature is not differentiable at $z = 0.5$ [mm]. Platform 3 is, however, locally adequate above and below that z coordinate.

5.2 Uniformly meshing a Riemannian manifold

To establish a uniformly dense infill pattern, one must create a quasi-uniform grid of points over the target manifold (domain Ω). Unlike the planar (Euclidean) or Gaussian (constant-curvature) domains, the curved surface here is a Riemannian manifold, requiring adaptation of standard uniform mesh approaches. In this work, a repulsion-force-based algorithm is used due to its relative simplicity and direct geodesic control. Other algorithms [108] can be used if higher computational efficiency is required.

Each point i generates a repulsion potential E_{ij} on any point j in the manifold's parametric domain. This potential is:

$$E_{ij} = d_{ij}^{-m}, \tag{5.2}$$

where d_{ij} is the distance between i and j, and $m > 1$ (chosen as 5 here). The total repulsion energy is

$$E = \sum_{\forall i \neq j}^{N} \frac{1}{2} E_{ij}, \tag{5.3}$$

which, upon minimization, yields a quasi-uniform distribution of points. Notably, for a curved Riemannian manifold, d_{ij} must be replaced by the geodesic distance d_{ij}^{geo}. In addition, the method requires the number of points N, defined in terms of command density and line width. These details are expanded in subsequent Subsections.

Geodesic distance between points d_{ij}^{geo}

Computing the geodesic distance demands solving the geodesic equation via the manifold's metric tensor $\mathbf{g}$. A manifold $\mathcal{M}$, described by $\mathbf{r}_{\mathcal{M}}(u,\theta) = \{x(u,\theta), y(u,\theta), z(u,\theta)\}$, has a metric tensor $g_{p_{ij}}$ at point p:

$$g_{p_{ij}} = \mathbf{e}_p^i \cdot \mathbf{e}_p^j, \tag{5.4}$$

where $\mathbf{e}_p^i$ and $\mathbf{e}_p^j$ are the embedded map's basis vectors. For a degree-six Bézier generatrix 5.5–5.7 revolved around an axis, the metric tensor $\mathbf{g_p}$ follows (5.8).

$$\mathcal{S} = \mathbf{uBP} = \begin{pmatrix} u^6 & u^5 & \cdots & u & 1 \end{pmatrix} (\mathbf{B_6}) \begin{pmatrix} P_1^x & P_1^z \\ \vdots & \vdots \\ P_7^x & P_7^z \end{pmatrix} \tag{5.5}$$

$$\mathbf{e^u} = \frac{\nabla \mathcal{S}}{\|\nabla \mathcal{S}\|} \tag{5.6}$$

$$\mathbf{e}^{\boldsymbol{\theta}} = e^{[\mathbf{e^y}]\theta} \tag{5.7}$$

$$\mathbf{g_p} = \begin{pmatrix} (\mathbf{u'BP})^x + (\mathbf{u'BP})^z & 0 \\ 0 & (\mathbf{uBP})^x \end{pmatrix} \tag{5.8}$$

The Christoffel symbols Γ^i_{jk} then emerge from (5.9), describing the geodesic equation (5.10) with boundary conditions (5.11).

$$\Gamma^i_{jk} = \frac{1}{2}(\mathbf{g}^{-1})_{il}(\frac{\partial}{\partial x^j} g_{lk} + \frac{\partial}{\partial x^k} g_{lj} - \frac{\partial}{\partial x^l} g_{jk}) \tag{5.9}$$

$$\begin{cases} \ddot{u} = -\Gamma^u_{uu}(\dot{u})^2 - \Gamma^u_{\theta\theta}(\dot{\theta})^2 \\ \ddot{\theta} = -2\Gamma^\theta_{u\theta}\dot{u}\dot{\theta} \end{cases} \tag{5.10}$$

$$\begin{cases} \mathcal{G}_{ij}(\tau = 0) = \{u_0, \theta_0\} = \{u_i, \theta_i\} \\ \mathcal{G}_{ij}(\tau = 1) = \{u_1, \theta_1\} = \{u_j, \theta_j\} \end{cases} \tag{5.11}$$

Integrating the resulting path defines the geodesic distance:

$$d_{ij}^{\text{geo}} = \int_0^1 \sqrt{g_{uu}\, \dot{\mathcal{G}}^u_{ij}(\tau) + g_{\theta\theta}\, \dot{\mathcal{G}}^\theta_{ij}(\tau)\, d\tau}. \tag{5.12}$$

Number of points N to distribute in the mesh

Because the repulsion-energy approach requires one to know how many points N will be distributed, assumptions about local density are introduced. Borrowing from Laguerre-Voronoi concepts [109], if each mesh point p_i is inscribed in a cell c_i of area A_i, the local density ρ_i depends on how many lines pass through c_i (denoted n) and the nominal line width e. A few commonly used infill patterns in the planar case are shown in Figure 5.3, each having different n.

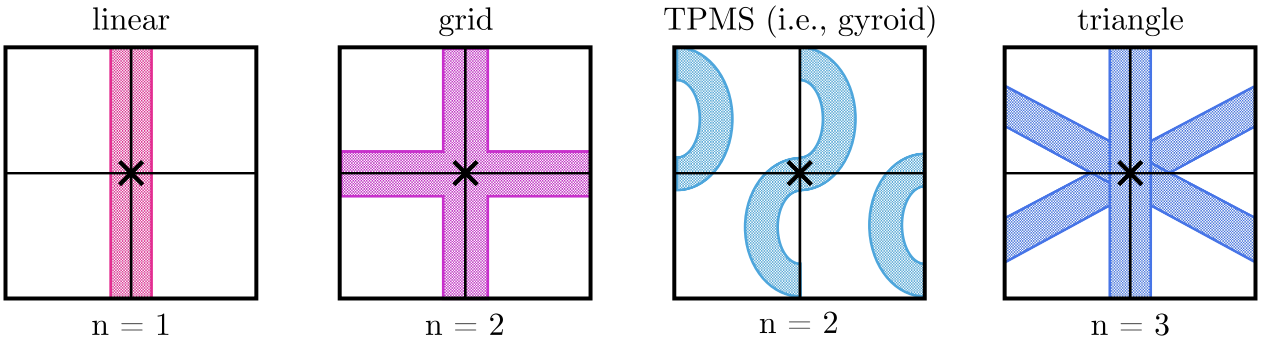


Figure 5.3: Common infill patterns with their respective n parameter.

If the local area A_i is approximated as a square of side l_i, then $A_i = l_i^2$, and

$$\rho_i \approx \frac{n\,e\,l_i}{A_i} = \frac{n\,e}{\sqrt{A_i}}. \tag{5.13}$$

For $n = 1$ or 2, the maximal density error ε_{max} in simpler patterns is:

$$\varepsilon_{max} = \frac{\rho_{max}}{\rho_{min}} = \sqrt{2}. \tag{5.14}$$

This means the actual infill density might exceed the target ρ_c, thus providing extra support for overhangs. A visual depiction of this error factor is given in Figure 5.4.

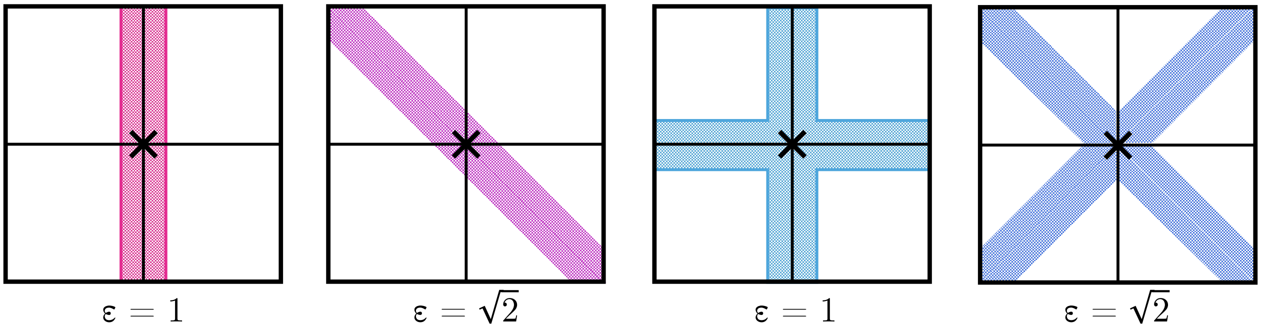


Figure 5.4: Illustration of the infill density error for $n = 1$ and $n = 2$.

Hence, to achieve a specific command density ρ_c across the domain Ω, the total number of points needed is:

$$N = \frac{A_\Sigma}{A_i} = \frac{A_\Sigma\,\rho_c^2}{n^2\,e^2}, \tag{5.15}$$

$$A_\Sigma = \int_\Omega \sqrt{\det \mathbf{g}}\; d\Omega \tag{5.16}$$

where A_Σ is the area (5.16) of Ω, computed via the determinant of $\mathbf{g}$. Once N is determined, those points are scattered initially and later repositioned to minimize (5.3).

5.3 Numerically optimizing the algorithm

While effective, the repulsion-energy-based approach can be computationally demanding. For instance, distributing even a modest number of points in a straightforward domain may be time-consuming. Therefore, several optimizations are introduced:

Explicit estimation of repulsion forces

Gradient-based optimizers benefit greatly from analytical or explicit gradients. When the gradient must be determined by finite differences, the cost can become prohibitive—especially here, where each evaluation involves solving geodesics. To reduce this cost, the repulsion-energy gradient is approximated explicitly through chord lengths (5.17–5.18), avoiding redundant integrations near every point.

$$d_{ij} \approx \sum_{k=1}^{n-1} \left\| \mathcal{G}(u(\tau_{k+1}), \theta(\tau_{k+1})) - \mathcal{G}(u(\tau_k), \theta(\tau_k)) \right\| \tag{5.17}$$

$$\begin{pmatrix} \frac{\partial d_{ij}}{\partial u} & \frac{\partial d_{ij}}{\partial \theta} \end{pmatrix} \approx \sum_{k=1}^{n-1} \frac{\mathcal{G}(u(\tau_{k+1}), \theta(\tau_{k+1})) - \mathcal{G}(u(\tau_k), \theta(\tau_k))}{\left\| \mathcal{G}(u(\tau_{k+1}), \theta(\tau_{k+1})) - \mathcal{G}(u(\tau_k), \theta(\tau_k)) \right\|} \nabla_{\{\mathbf{c}\}} \mathcal{S}(\tau_k) \tag{5.18}$$

Scaling of repulsion energy and forces

Numerical stiffness arises if a small change in positions yields a disproportionately large change in energy. Consequently, the energy is normalized by E_0, the initial mesh energy (5.19), and the gradient is further scaled by F_0 (5.20–5.21). These scaling factors ease the optimizer's convergence by reducing abrupt variations in function values.

$$E_{ij} = \frac{1}{E_0} d_{ij}^{-m} \tag{5.19}$$

$$F_0 = \max \mathbf{F}_0 \begin{cases} \frac{1}{0.05(u_{max} - u_{min})}, & \max F_0^u > \max F_0^\theta \\ \frac{1}{0.05(\theta_{max} - \theta_{min})}, & \max F_0^u < \max F_0^\theta \end{cases} \tag{5.20}$$

$$\mathbf{F}_i = \frac{1}{E_0 F_0} \sum_{j \neq i}^{N} -\nabla E_{ij} \tag{5.21}$$

Improving the initial seed

Providing a better initial seed distribution can drastically reduce overall runtime. Three main seeding methods are implemented: a geodesic blue noise sampling approach, a Euclidean repulsion energy distribution, and a Euclidean blue noise sampling strategy. Each option balances runtime, accuracy, and ease of implementation differently. If N is small, the geodesic blue noise approach often yields an initial distribution very close to the final solution, though at a high computational cost. For large N, Euclidean-based strategies are favored for faster seeding.

5.4 Generating infill paths

Once the geodesic mesh is obtained, an infill path must be generated. A simple illustration of how to do this is presented here. The objective is to traverse all mesh points exactly n times, forming n Hamiltonian paths. This is analogous to a Traveling Salesman Problem (TSP), but without requiring a return loop. We use the well-known Nearest Neighbors (NN) heuristic and then apply a 2-opt approach [110] to reduce self-intersections and path cost. These algorithms are defined in Algorithms 1 and 2.

Algorithm 1 Nearest Neighbor Heuristic (No Return TSP)

```
Input: A set of cities $C = \{c_1, c_2, \ldots, c_n\}$ and distances $d(c_i, c_j)$.
Output: A route (sequence of cities) visiting each city exactly once, not returning to the start.
Choose an arbitrary starting city $c_s$.
Visited ← $\{c_s\}$
CurrentCity ← $c_s$
Route ← $[c_s]$
while Visited ≠ $C$ do
    NextCity ← $\arg\min_{c \in (C \setminus \text{Visited})} d(\text{CurrentCity}, c)$
    Visited ← Visited ∪ {NextCity}
    Route ← Route appended with NextCity
    CurrentCity ← NextCity
end while
return Route                                  ▷ No return to start in this algorithm.
```

The path between consecutive points is constructed by solving the geodesic equation (5.10) for each pair. Although this is a relatively basic path planning scheme, it suffices to demonstrate the uniform geodesic point distribution's effectiveness. Future work might incorporate advanced path planning that addresses machine dynamics or specialized AM process constraints.

5.5 Validating the uniform density hypothesis

Two methods are used to evaluate whether the uniform density assumption holds: a kernel density evaluation and a distance histogram. First, each point i in the mesh can be assigned a local density ρ_i via a kernel function W_{ij} (5.22):

$$\rho_i = \sum_{j \neq i}^{N} m_j W_{ij}, \tag{5.22}$$

where m_j is set to 1 (each point contributes equally) and W_{ij} depends on a chosen vicinity radius h (5.23):

$$W_{ij} = \frac{15}{\pi\, h^6} \begin{cases} (h - d_{ij}^{\text{geo}})^3, & 0 \leq d_{ij}^{\text{geo}} \leq h, \\ 0, & d_{ij}^{\text{geo}} > h. \end{cases} \tag{5.23}$$

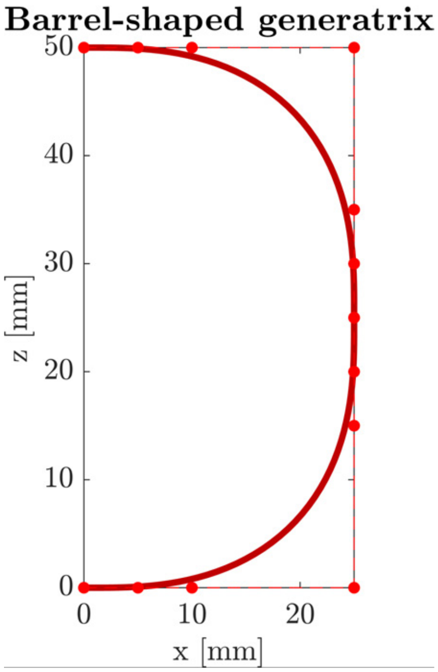
Algorithm 2 2-Opt Algorithm (Improvement Procedure)

```
Input: An initial route $R = [r_1, r_2, \ldots, r_n]$.
Output: An improved route (shorter if possible).
improvement ← true
while improvement do
    improvement ← false
    for $i = 1$ to $n - 2$ do
        for $j = i + 2$ to $n - 1$ do
            currentDistance ← $d(r_i, r_{i+1}) + d(r_j, r_{j+1})$
            newDistance ← $d(r_i, r_j) + d(r_{i+1}, r_{j+1})$
            if newDistance < currentDistance then
                Reverse the order of the cities from $r_{i+1}$ to $r_j$ in the route
                improvement ← true
            end if
        end for
    end for
end while
return $R$                                        ▷ Final improved route
```

When points are uniformly distributed, three main density values typically emerge: a small one (near vertices), an intermediate one (along the boundary), and a large one (in the interior). A distance histogram further quantifies uniformity by recording each point's distance to its $2n$ nearest neighbors. Narrower histograms (i.e., lower standard deviation) indicate more uniform spacing.

5.6 Case studies and results

Two build platforms are examined to illustrate the versatility and performance of the algorithm. First, a barrel-shaped platform is used as a simpler case, featuring monotonic, non-concave curvature in both $\{u, \theta\}$ directions. Multiple results are shown, including the generatrix definition, metric tensor and Christoffel symbols, geodesic repulsion energy minimization outcomes, density validation, and a path planning demonstration. Figure 5.5 summarizes the geometry, while Figure 5.6 shows two domains within this platform: Ω_1 (a rectangular subset with $\rho_c = 50\%$, $n = 1$) and Ω_2 (a non-rectangular slice from a complex part, $\rho_c = 25\%$, $n = 1$). The selected part [85] combines wide areas, sharp corners, and concavities, illustrating the design possibilities of this method.



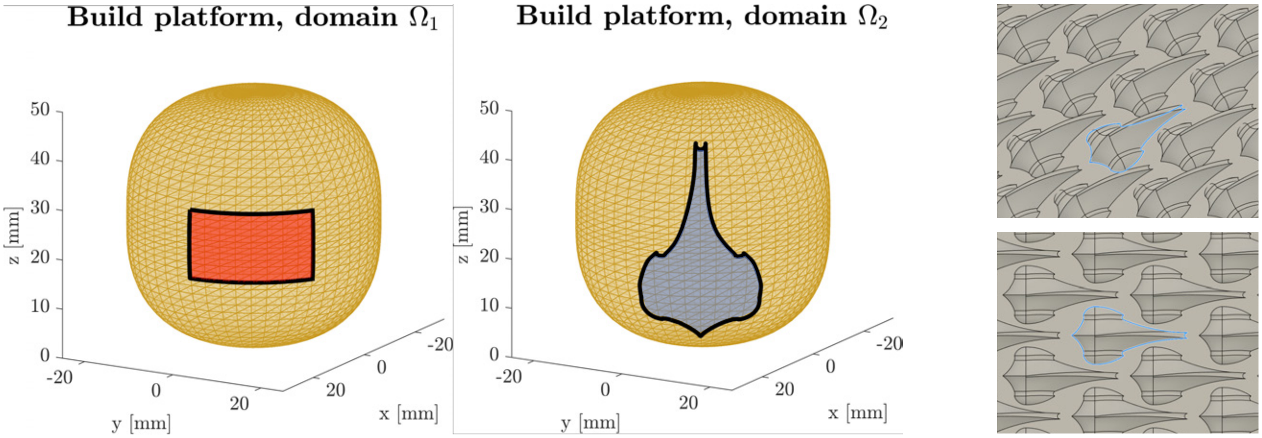
Revolved parametric curve

Degree-six Bézier spline composition

Control polygon defined in zx-plane

Segment	x [mm]	z [mm]
1	0	50
	5	50
	10	50
	25	50
	25	35
	25	30
	25	25
2	25	25
	25	20
	25	15
	25	0
	10	0
	5	0
	0	0

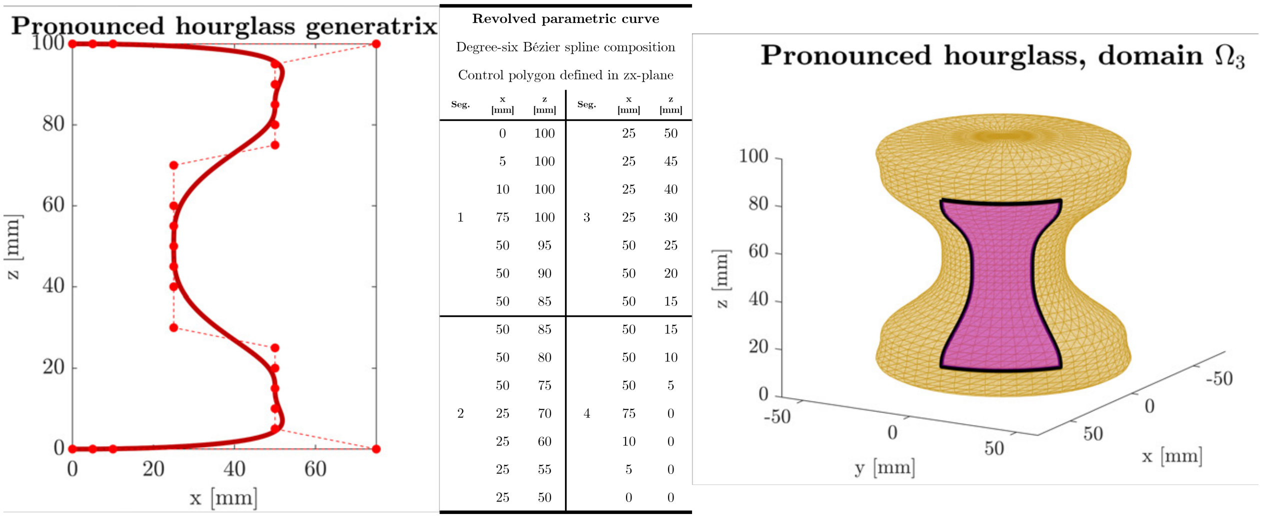
Figure 5.5: Barrel-shaped generatrix and its control polygon in the zx-plane.



(a) (b) (c)

Figure 5.6: (a) Domain Ω_1 (red) over the barrel platform; $u \in [0.8, 1.2], \theta \in [0, 1]$. (b) Domain Ω_2 (blue), a slice of a real part. (c) Isometric/top views of the part from [85]; the Ω_2 slice is highlighted in blue.

Next, two hourglass-shaped build platforms demonstrate how the method behaves with concave curvature zones. The difference between them is the steepness of their concavities. The same sequence of results is shown—continuity checks, metric/Christoffel analyses, mesh optimization, density validation, and path planning. Figures 5.7 and 5.8 show the geometries and a target domain Ω_3 ($u \in [1, 3]$, $\theta \in [0, 1]$, $\rho_c = 10\%$, $n = 1$).



Revolved parametric curve

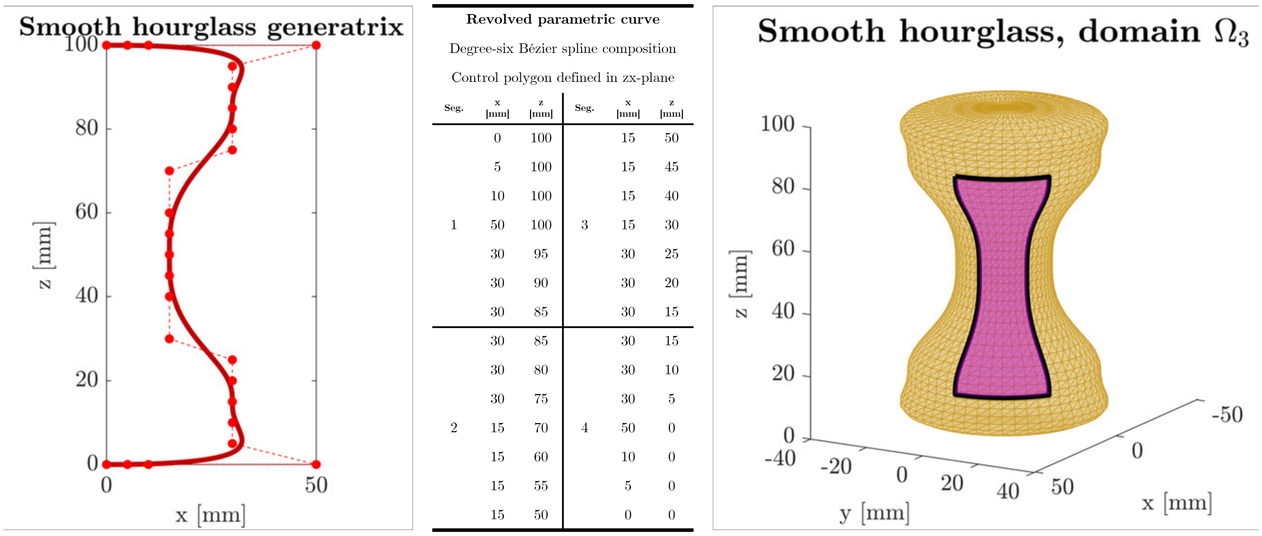
Degree-six Bézier spline composition

Control polygon defined in zx-plane

Seg.	x [mm]	z [mm]	Seg.	x [mm]	z [mm]
1	0	100	3	25	50
	5	100		25	45
	10	100		25	40
	75	100		25	30
	50	95		50	25
	50	90		50	20
	50	85		50	15
2	50	85	4	50	15
	50	80		50	10
	50	75		50	5
	25	70		75	0
	25	60		10	0
	25	55		5	0
	25	50		0	0

(a) (b) (c)

Figure 5.7: (a) Pronounced hourglass generatrix; (b) control polygon in the zx-plane; (c) domain Ω_3 (pink) on the pronounced hourglass platform.



Revolved parametric curve

Degree-six Bézier spline composition

Control polygon defined in zx-plane

Seg.	x [mm]	z [mm]	Seg.	x [mm]	z [mm]
1	0	100	3	15	50
	5	100		15	45
	10	100		15	40
	50	100		15	30
	30	95		30	25
	30	90		30	20
	30	85		30	15
2	30	85	4	30	15
	30	80		30	10
	30	75		30	5
	15	70		50	0
	15	60		10	0
	15	55		5	0
	15	50		0	0

(a) (b) (c)

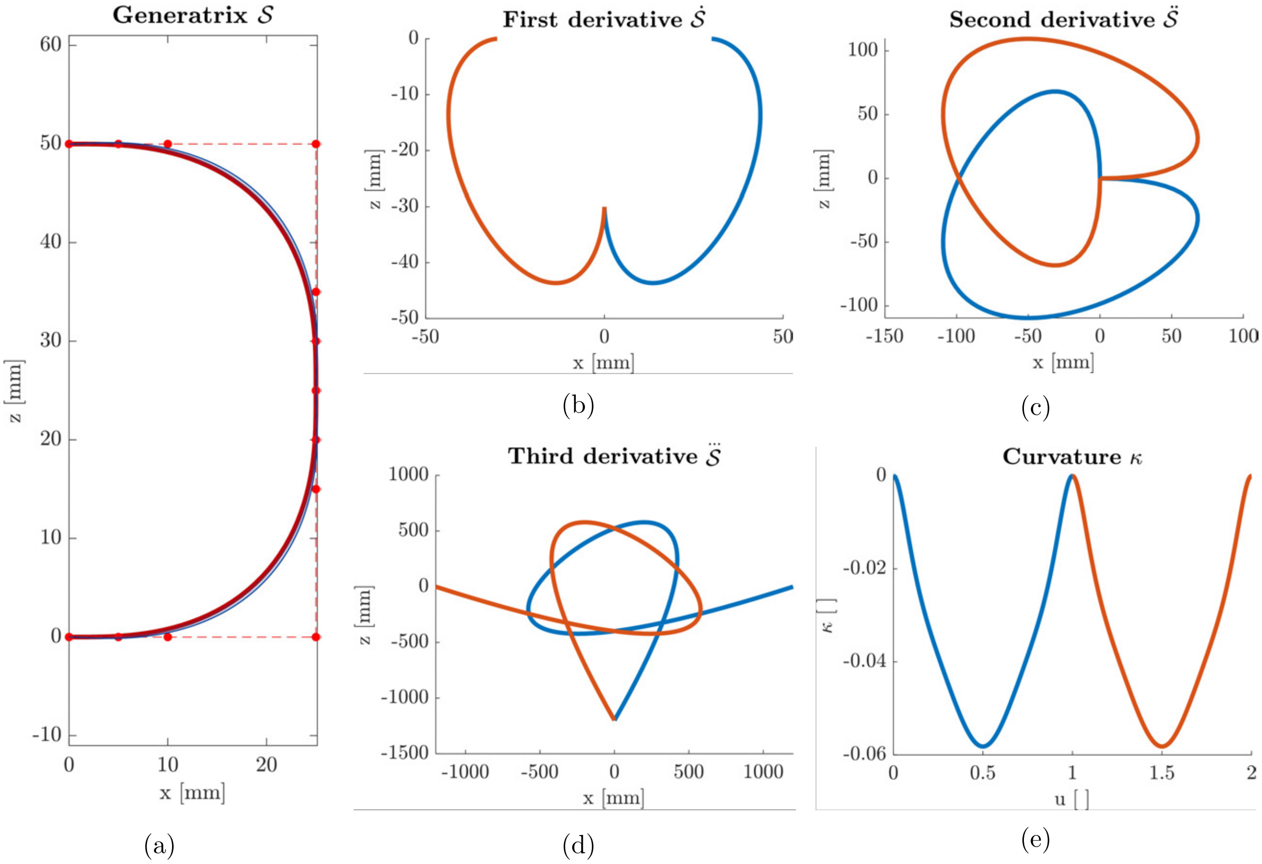
Figure 5.8: (a) Smooth hourglass generatrix; (b) control polygon in the zx-plane; (c) domain Ω_3 (pink) on the smooth hourglass platform.

5.6.1 Barrel-shaped case study

This Subsection covers the generatrix definition and continuity, the metric tensor and Christoffel symbols, the repulsion energy minimization algorithm's outcomes, uniformity checks of the resulting density, and a path-planning example. Results are shown for both Ω_1 and Ω_2.

Generatrix definition and continuity

First, the continuity requirements discussed in Section 5.1 are verified. Figure 5.9 presents the barrel generatrix $\mathcal{S}$, along with its first, second, and third derivatives ($\dot{\mathcal{S}}$, $\ddot{\mathcal{S}}$, $\dddot{\mathcal{S}}$), and the curvature κ. The C^3 continuity at the spline knot confirms that the surface curvature is fully differentiable throughout, thus forming a valid Riemannian manifold.

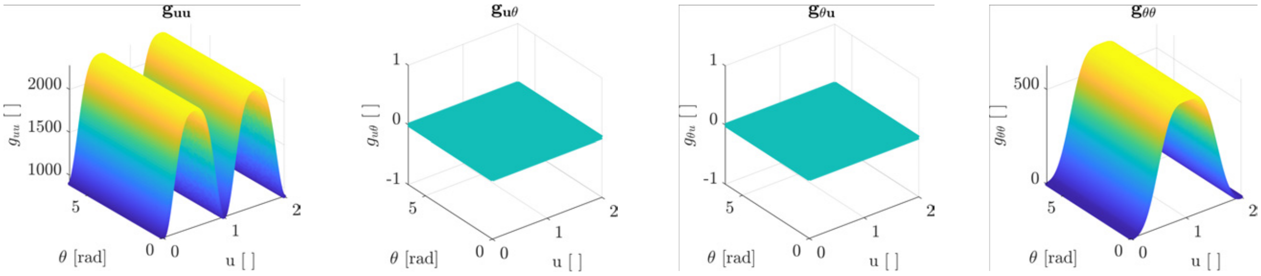


(a) (b) (c) (d) (e)

Figure 5.9: Barrel-shaped build platform. (a) Generatrix $\mathcal{S}$ in red, with its control polygon as a dashed line and a curvature comb in blue. (b, c, d) First, second, and third derivatives. Blue corresponds to $u \in [0, 1]$ and orange to $u \in [1, 2]$. (e) Generatrix curvature κ, similarly color-coded.

Metric tensor and christoffel symbols

Figures 5.10 and 5.11 show the metric tensor g_{ij} and the Christoffel symbols Γ^i_{jk} across the barrel platform. The off-diagonal terms remain zero (verifying orthogonality in the $\{u, \theta\}$ parametric basis), and any non-zero symbols vary smoothly, confirming the surface's differentiability. Near the poles ($u = 0$ or $u = 2$), certain Christoffel symbols tend to infinity because $g_{\theta\theta} \to 0$, so one must avoid those poles in practical implementations.



(a) (b) (c) (d)

Figure 5.10: Metric tensor components g_{ij} for the barrel platform. Colors represent the magnitude's z-coordinate.

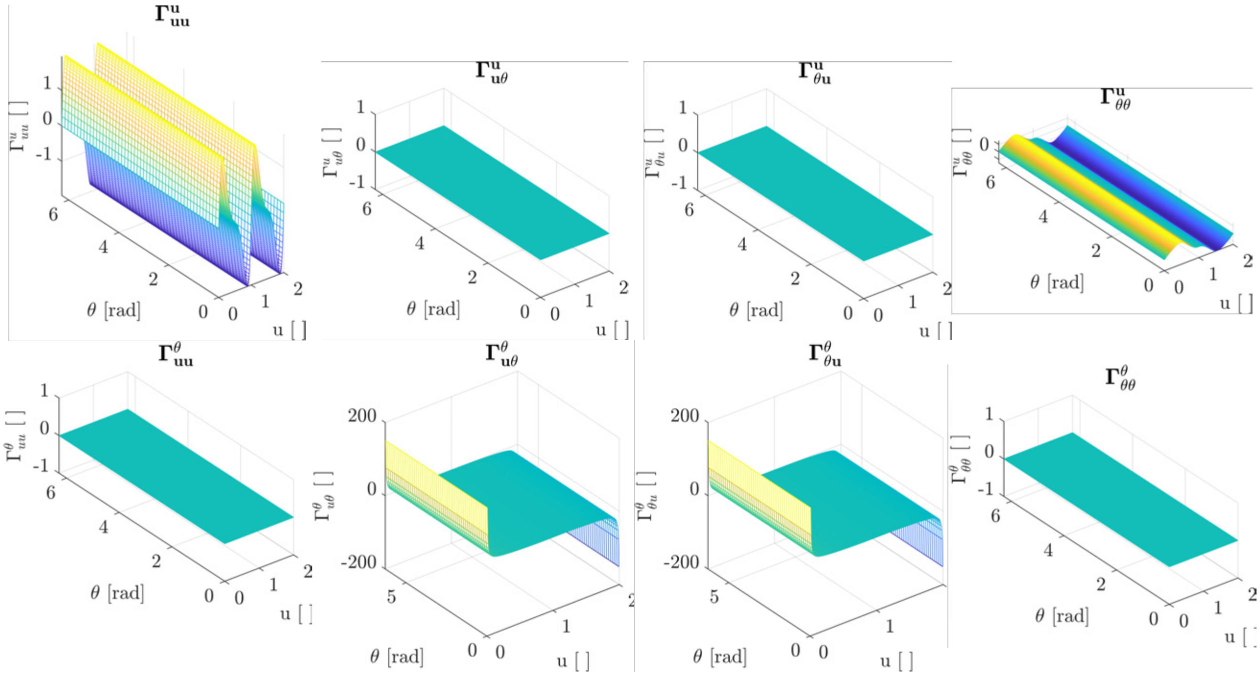
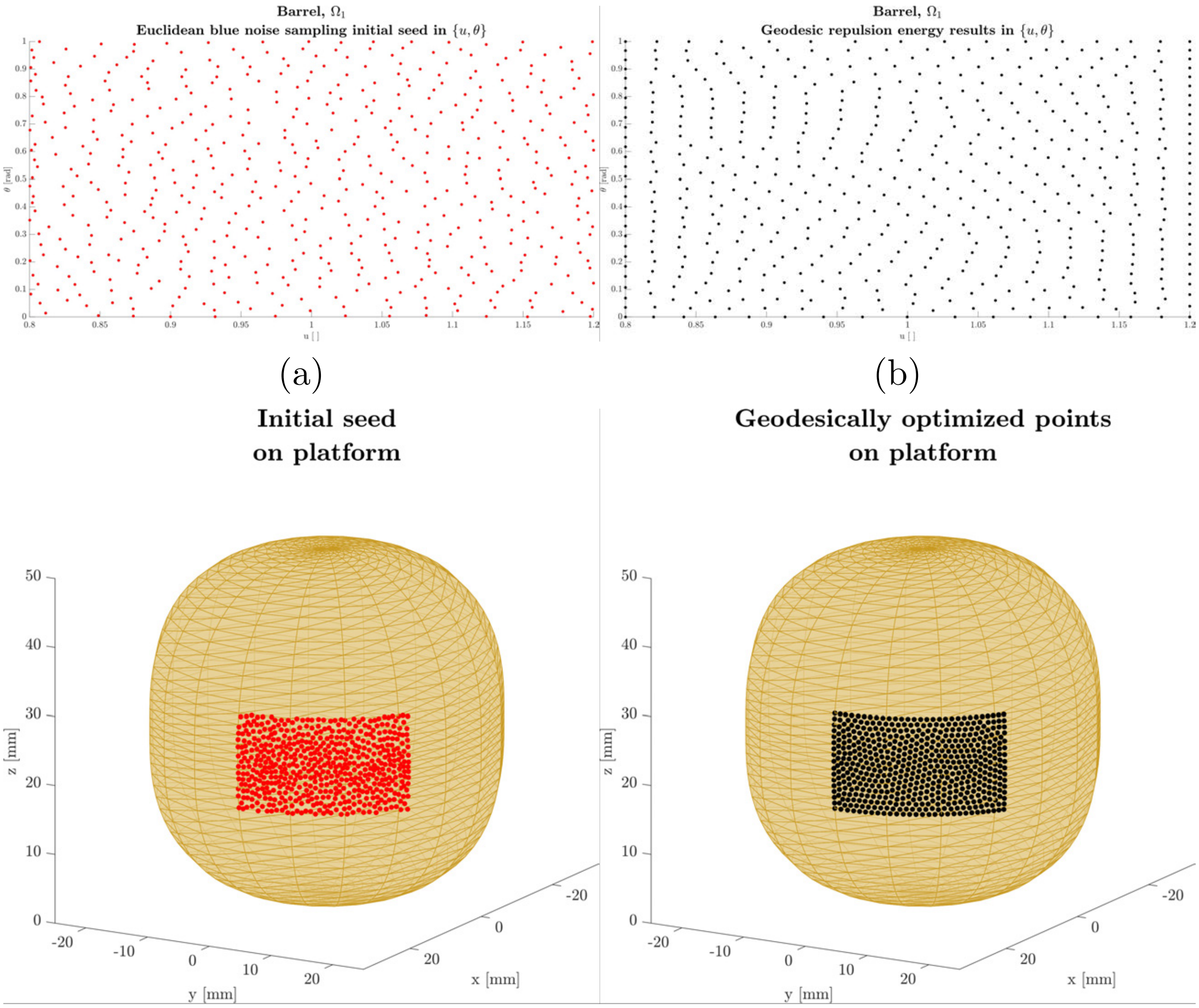


Figure 5.11: Christoffel symbols Γ^i_{jk} for the barrel platform. The zero-valued components confirm constant curvature along θ.

Results of the repulsion energy minimization and density validation

For Ω_1, $u \in [0.8, 1.2] \times \theta \in [0, 1]$ with $\rho_c = 50\%$ and $n = 1$. By 5.15, $N = 538$. An initial Euclidean blue noise sampling seed was computed in 210.0 s, and the overall optimization took 44190.4 s. Figure 5.12 compares the initial seed and final distribution. The repulsion-based algorithm yields a more uniform arrangement of points.

To confirm uniformity, the kernel density function is shown in Figure 5.13, illustrating how the repulsion algorithm greatly narrows the distribution range. Three main density levels



(a) (b)

(c) (d)

Figure 5.12: Barrel platform, domain Ω_1 ($u \in [0.8, 1.2], \theta \in [0, 1], \rho_c = 50\%$, $n = 1$). (a) Initial seed (Euclidean blue noise) in $\{u, \theta\}$. (b) Optimized distribution in $\{u, \theta\}$. (c) Initial seed on the platform (Cartesian). (d) Optimized distribution on the platform.

appear: the vertices, the boundary, and the interior—exactly as expected for a sufficiently dense infill pattern.

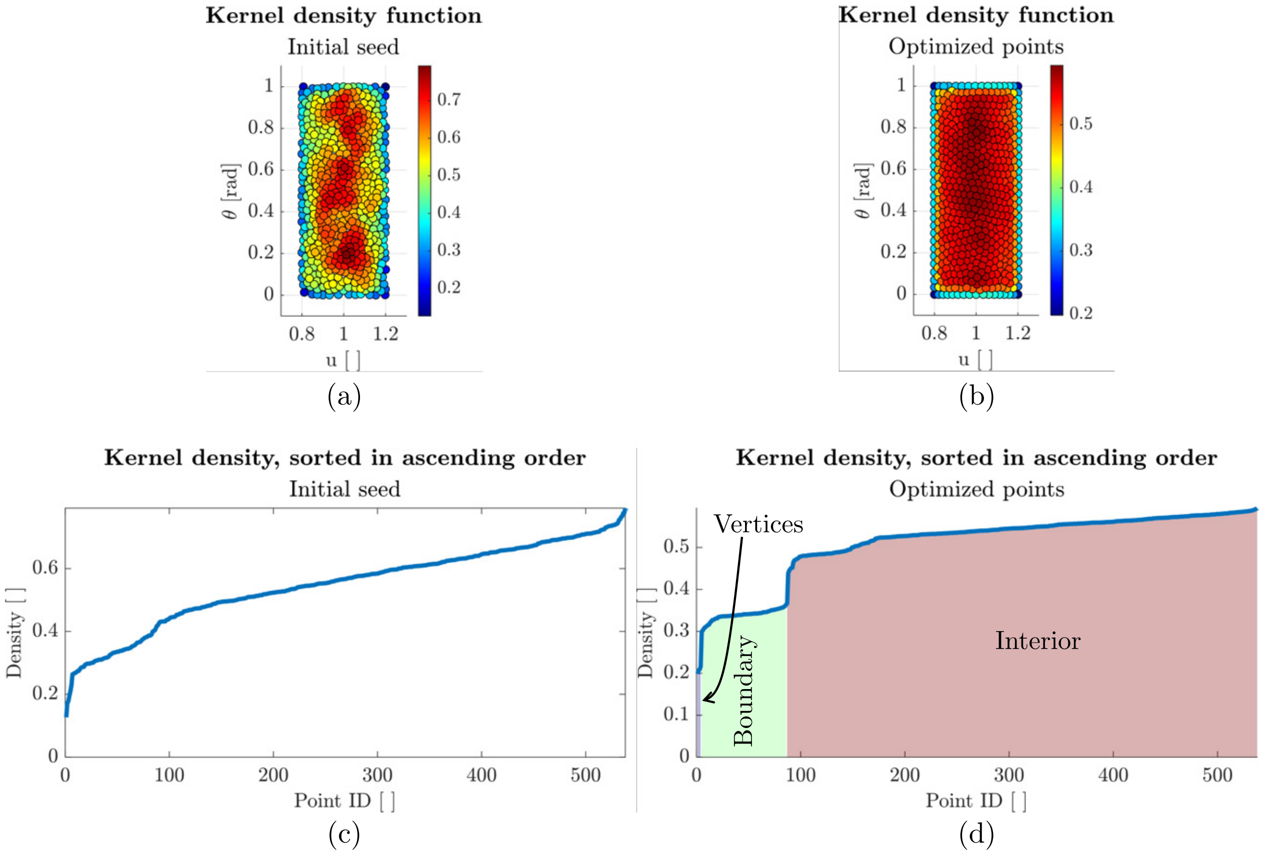


Figure 5.13: Kernel density evaluation for Ω_1. (a, b) Initial vs. optimized mesh in $\{u, \theta\}$, colored by local density. (c, d) Sorted density plots, illustrating a clear separation among vertices, boundary, and interior after optimization.

Figure 5.14 provides a histogram of each point's distance to its two nearest neighbors. For the initial seed, $\mu_{initial} = 0.740\,\text{mm}$ and $\sigma_{initial} = 0.11\,\text{mm}$. After optimization, $\mu_{optimized} = 0.848\,\text{mm}$ and $\sigma_{optimized} = 0.030\,\text{mm}$. This substantial decrease in standard deviation corroborates the distribution's improved uniformity, supporting stable overhangs and bridges in NPAM.

Turning to Ω_2, $N = 174$ points and a Euclidean repulsion energy seed was used, requiring 891.3 s for the initial step and 4403.0 s in total. Figure 5.15 shows the difference between the initial seed and the final, geodesically optimized mesh. The domain is non-rectangular, with boundaries enforced by a non-linear constraint in $\{u, \theta\}$.

Although the seed here was not especially close to the final arrangement, the algorithm successfully distributes points more uniformly (Figure 5.16). Only two distinct density levels are easily distinguishable in the sorted plot, reflecting a low overall density ($\rho_c = 25\%$) and more complex boundary geometry.

Figure 5.17 compares the distance histograms. The standard deviation falls from $\sigma_{initial} =$

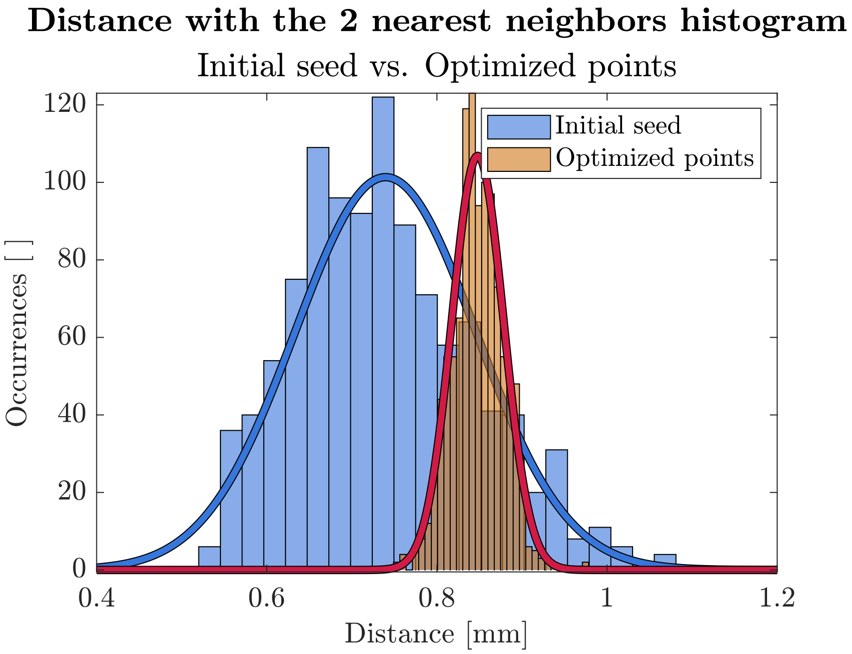
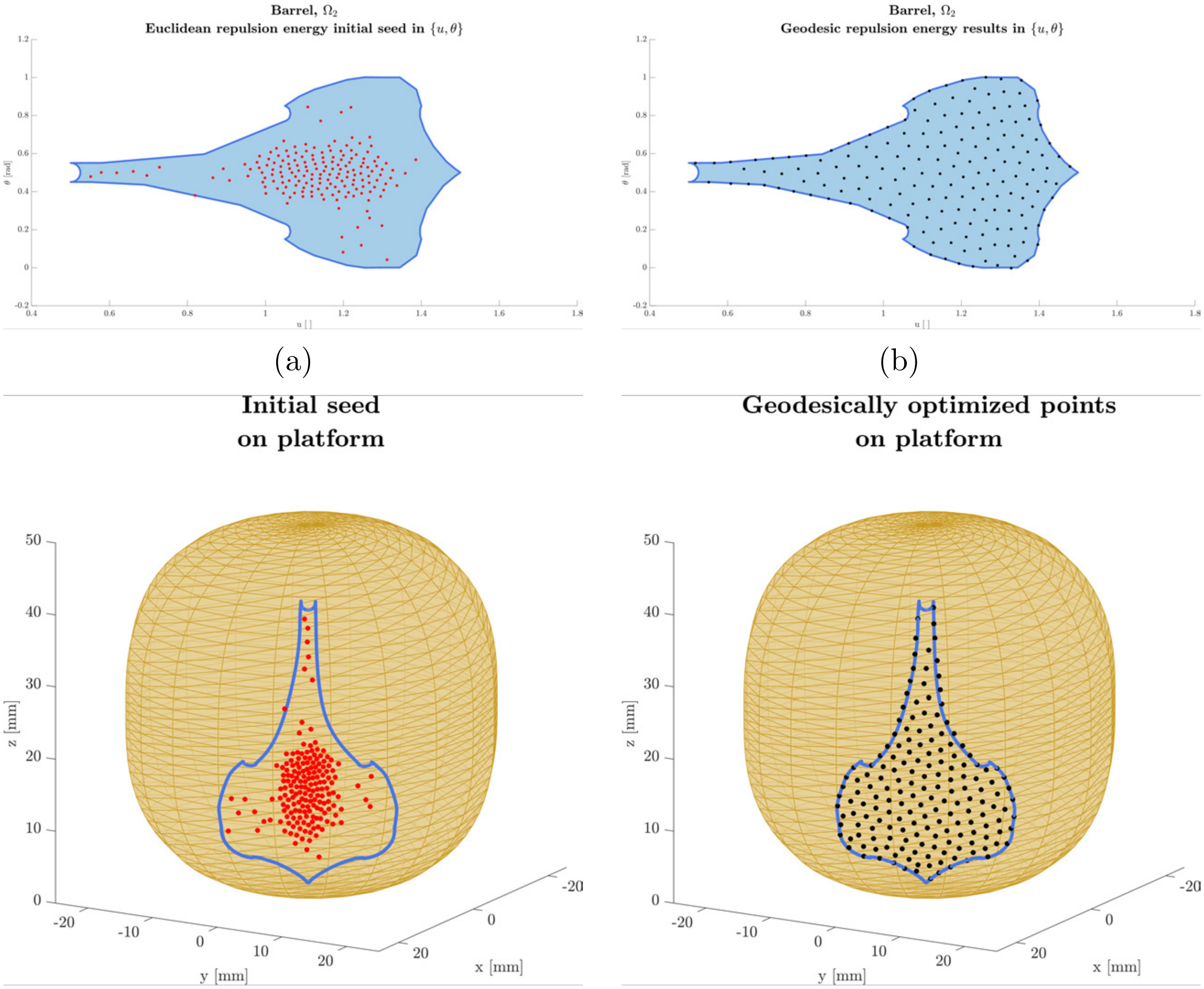


Figure 5.14: Histograms of the distance to the two nearest neighbors for Ω_1. Blue: initial seed; orange: optimized distribution. Overlaid lines are normal fits for visual reference.

0.560 mm to $\sigma_{optimized} = 0.159$ mm, a 71.6% reduction. However, around sharper corners, the method can still be further improved to enhance boundary coverage.

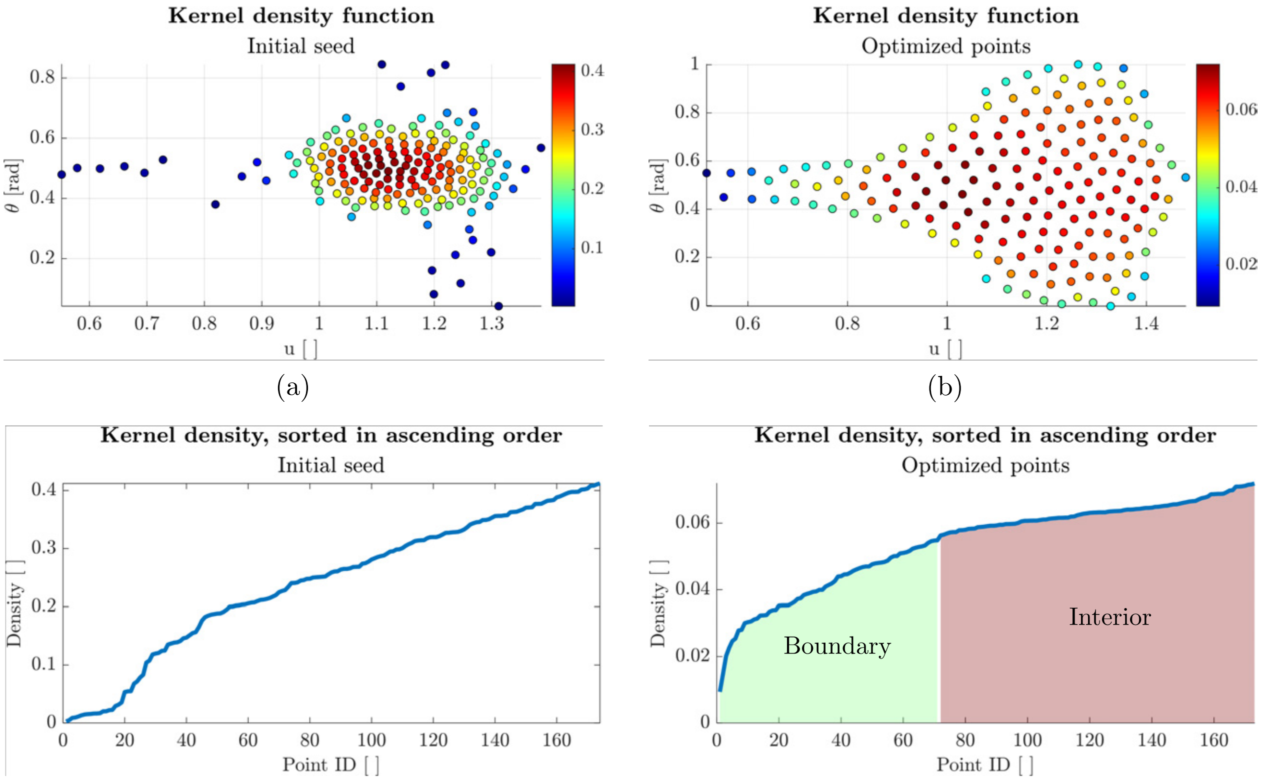
Path planning example with 2-opt Nearest Neighbor

As a demonstration, Figures 5.18 and 5.19 show geodesic path planning (2-opt NN) for Ω_1 and Ω_2. Although the outcome is a series of Hamiltonian paths with no returns, certain refinements could be introduced, such as specialized boundary constraints, smoothing strategies [111], or dynamic performance optimization [112].



(a) (b) (c) (d)

Figure 5.15: Barrel platform, domain Ω_2 (complex slice, $\rho_c = 25\%$, $n = 1$). (a) Initial seed (Euclidean repulsion) in $\{u, \theta\}$. (b) Optimized distribution in $\{u, \theta\}$. (c) Initial seed on the platform. (d) Optimized distribution on the platform. The slice boundary is shown in thick blue.



(a) (b)

(c) (d)

Figure 5.16: Kernel density for Ω_2. (a) Initial seed in $\{u, \theta\}$. (b) Optimized distribution. (c, d) Sorted density plots. Two main regions appear: boundary (green) and interior (red).

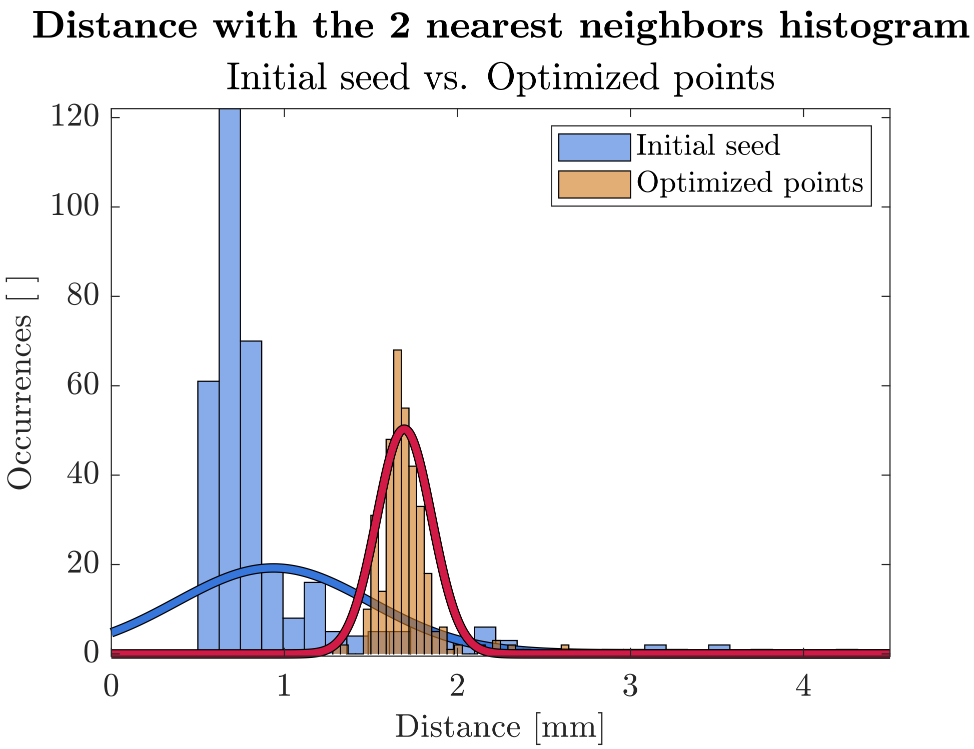


Figure 5.17: Histograms of distances to two nearest neighbors for Ω_2. Blue: initial seed; orange: optimized result. Normal fits overlaid for clarity.

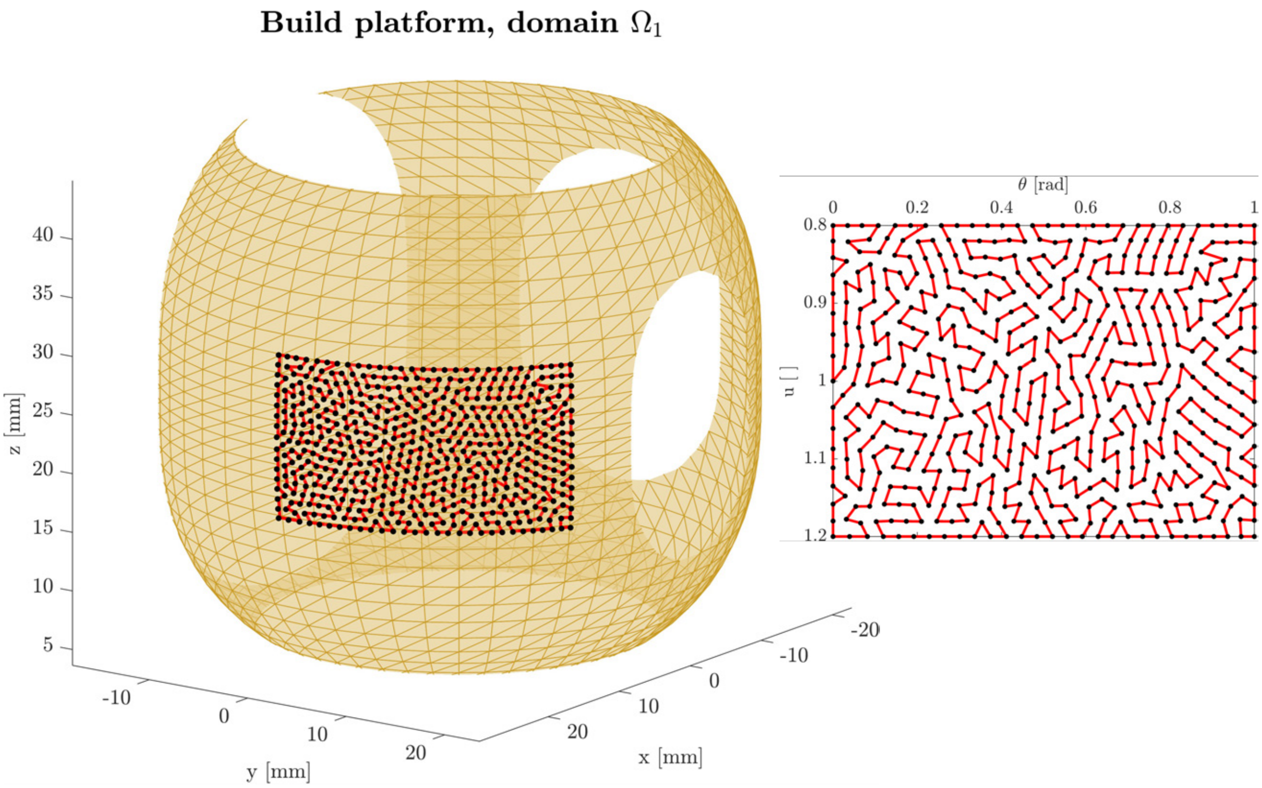


Figure 5.18: Geodesic path planning on the barrel platform, domain Ω_1, displayed in $\{u, \theta\}$ (right) and on the surface (left).

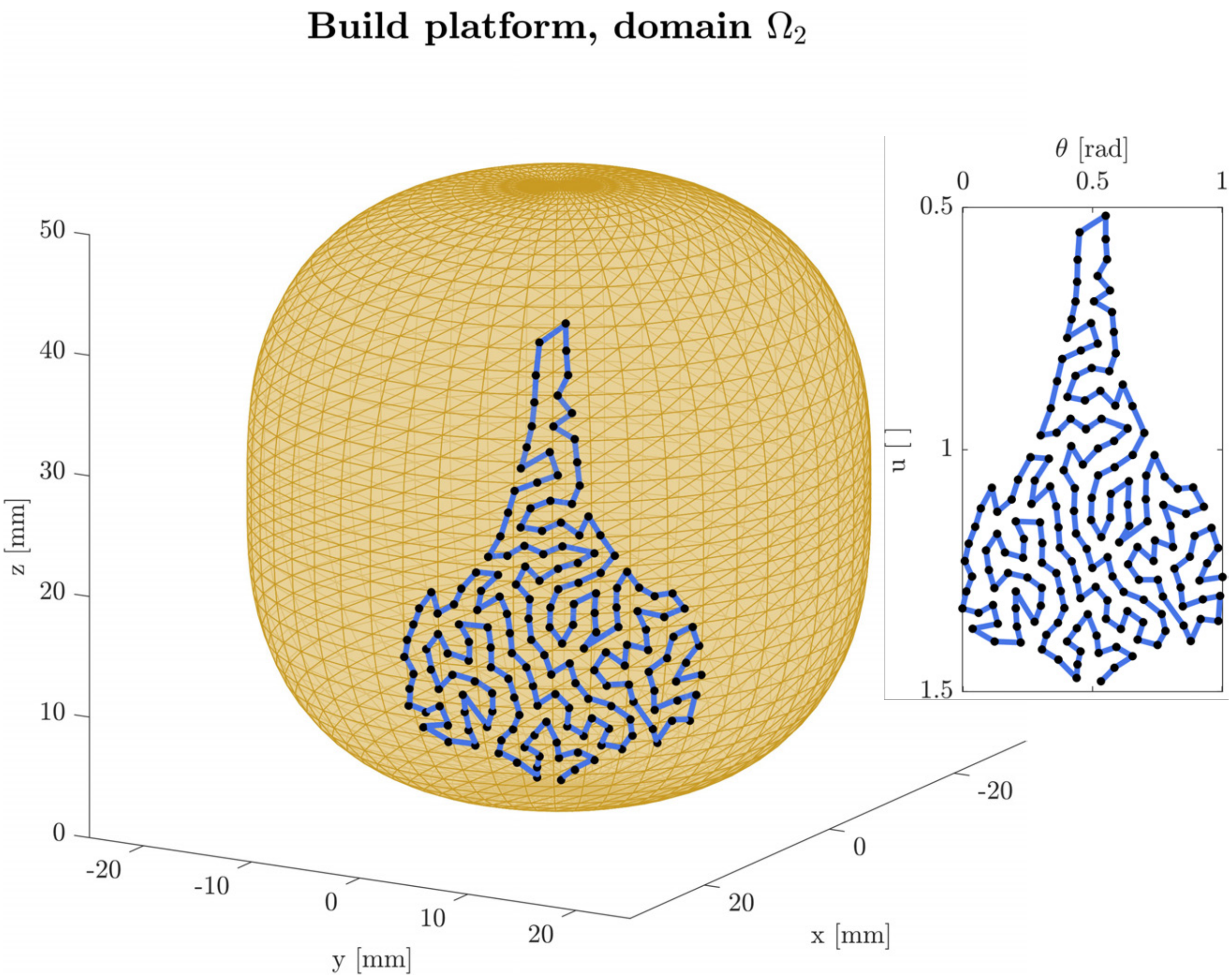


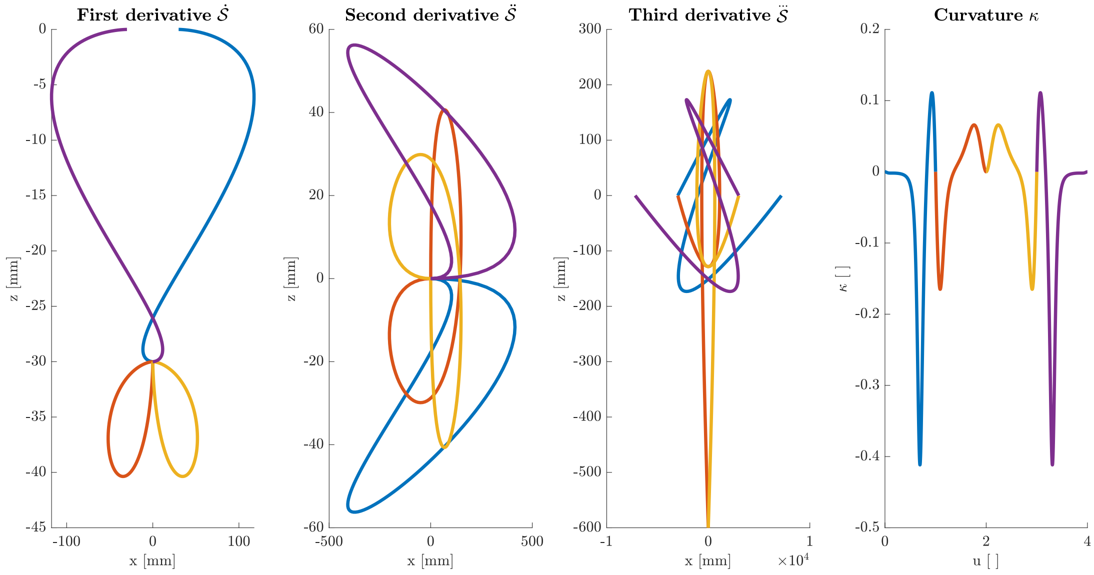
Figure 5.19: Geodesic path planning on the barrel platform, domain Ω_2, in $\{u, \theta\}$ (right) and on the surface (left). The boundary constraint could be further refined for complex slice polygons.

5.6.2 Hourglass-shaped case study

Here, the hourglass-shaped build platforms are used to assess the algorithm's performance in a more challenging type of geometry, including concave curvature. As in the previous examples, the spline continuity is verified, and the resulting metric tensor and Christoffel symbols are analyzed. Then, the repulsion algorithm is run, examining density and presenting an example path for both cases.

Generatrix definition and continuity, metric tensor, and Christoffel symbols

Figures 5.20–5.22 show the pronounced hourglass derivatives, curvature, and the manifold's metric tensor components and Christoffel symbols. The extended spline composition (four segments) ensures the generatrix remains smooth, preserving differentiable curvature over $u \in [0, 4]$. Figure 5.23 shows a summary of these variables for the smooth hourglass.



(a) (b) (c) (d)

Figure 5.20: Pronounced hourglass generatrix. (a, b, c) First, second, and third derivatives across segments $u \in [0, 1]$, $[1, 2]$, $[2, 3]$, $[3, 4]$. (d) Curvature κ also color-coded by segment.

Results of the repulsion algorithm and density validation

The selected domain Ω_3 is $u \in [1, 3], \theta \in [0, 1]$ with $\rho_c = 10\%$, $n = 1$, so $N = 205$ for the pronounced hourglass, and $N = 105$ for the smooth hourglass. Three initial seed methods were tested, each yielding different runtime and final standard deviations. Figure 5.24 illustrates the Euclidean blue noise sampling seed (8.9 s) vs. the geodesically optimized distribution (2777.1 s total) on the pronounced hourglass. As expected, fewer points cluster around the area of negative curvature near $u = 2$, consistent with the embedded map's geometry.

The kernel density plots are presented in Figure 5.25 for the pronounced hourglass using the Euclidean blue noise sampling initial seed method. Conversely, Figure 5.26 represents

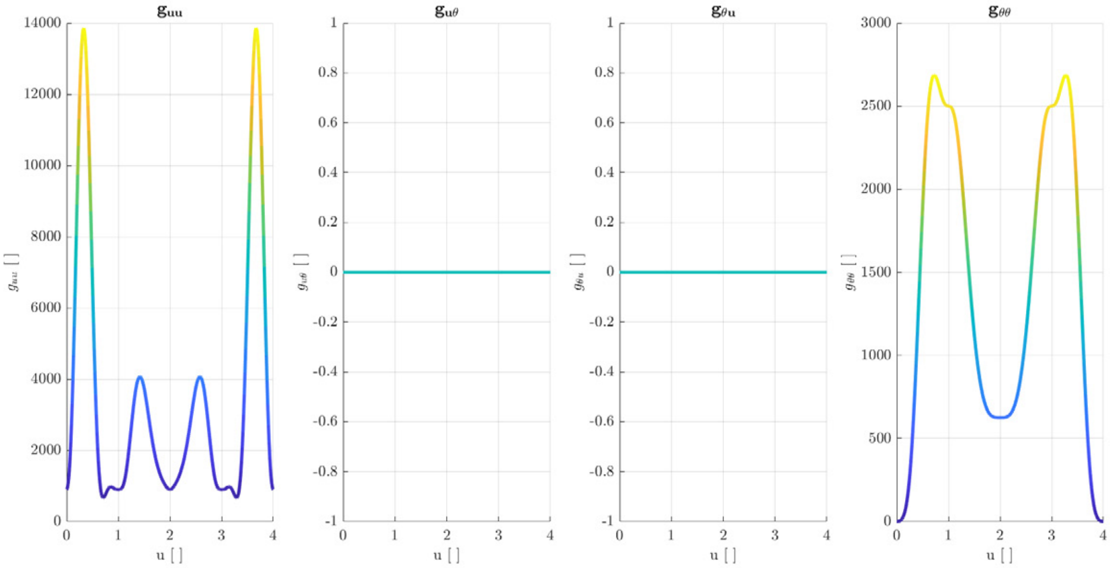


Figure 5.21: Metric tensor components g_{ij} for the pronounced hourglass platform, plotted against u. The θ-direction is constant.

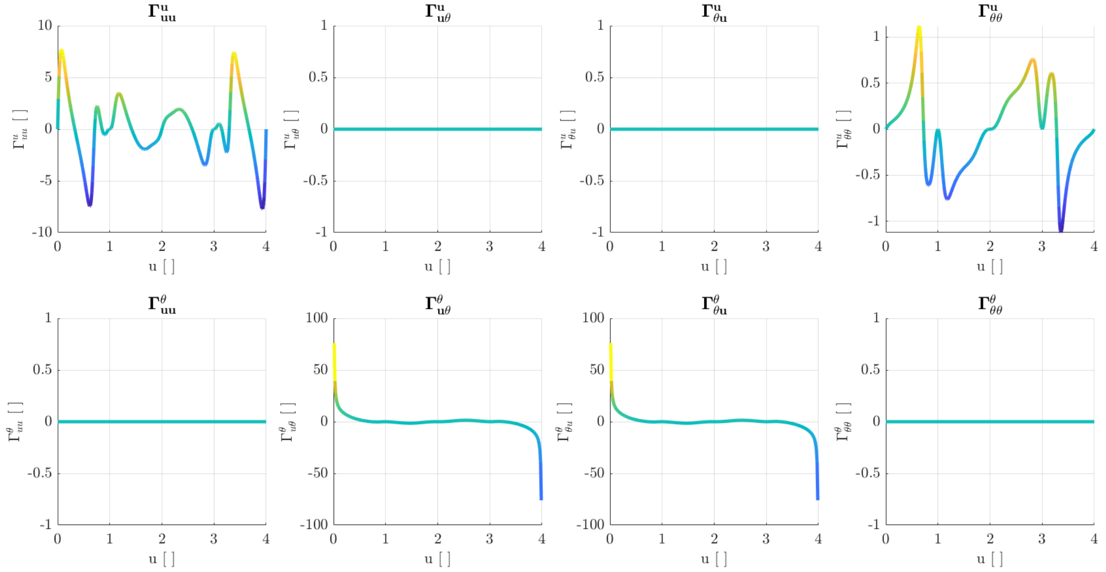


Figure 5.22: Christoffel symbols Γ^i_{jk} for the pronounced hourglass platform, also plotted against u.

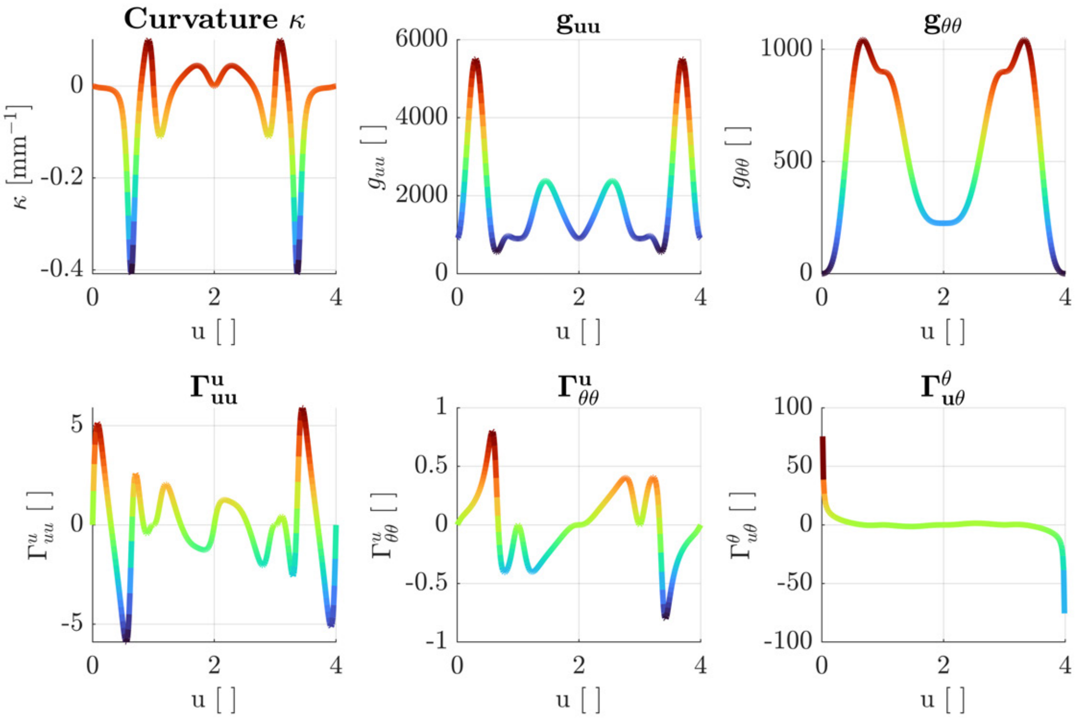
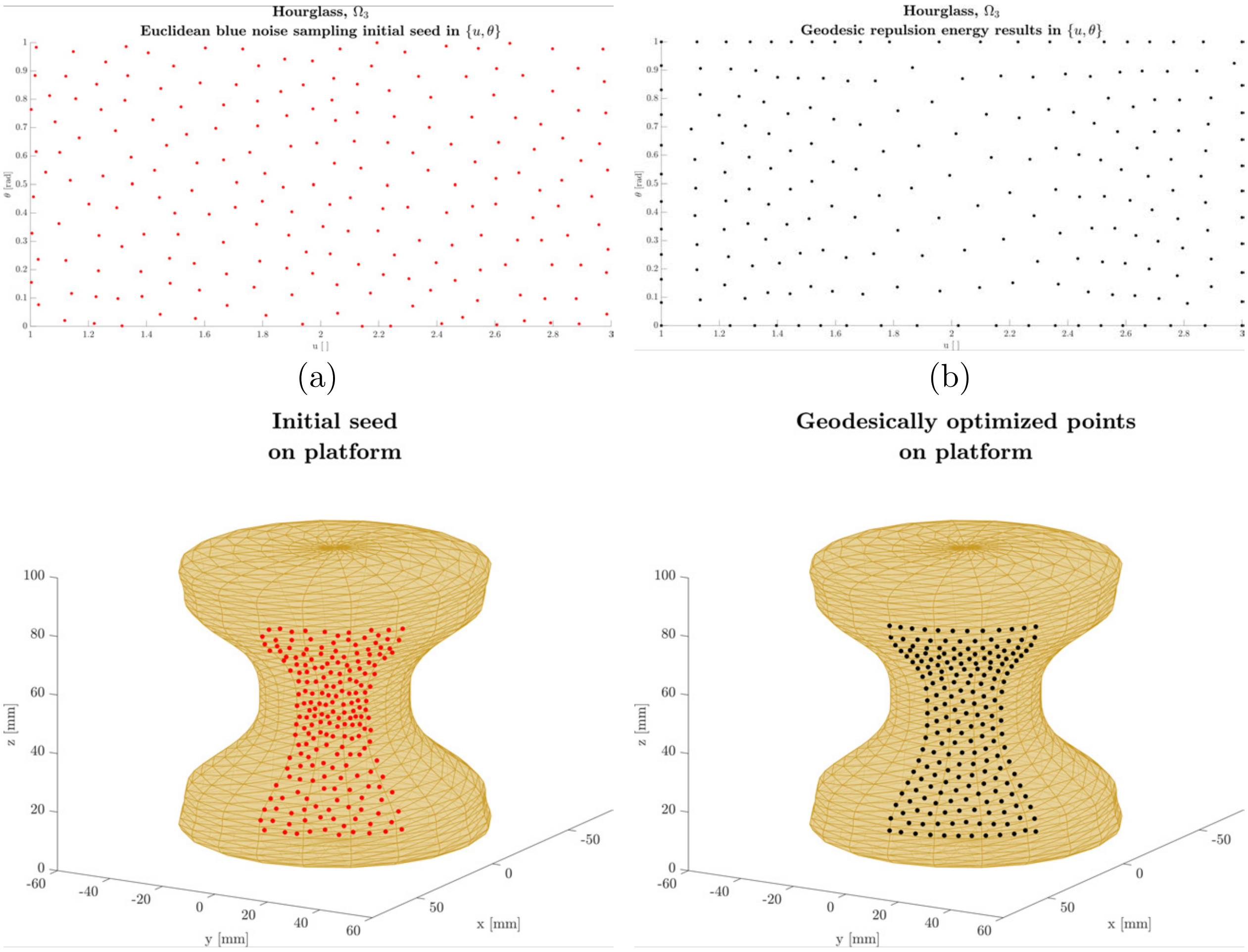


Figure 5.23: Curvature, non-zero metric tensor components and Christoffel symbols of the smooth hourglass platform, plotted against u.

the point distribution and the kernel density plots in a summarized manner for the smooth hourglass platform, using the geodesic blue noise sampling initial seed method. Both Figures confirm three main density levels (vertices, boundary, interior) but with some discretization effects at low ρ_c.

Figure 5.27 shows distance histograms for the Euclidean blue noise seed vs. final distribution: standard deviation drops from $\sigma_{initial} = 0.941\,\text{mm}$ to $\sigma_{optimized} = 0.316\,\text{mm}$ (66.4% reduction). Though not as dramatic as in the barrel examples, this still represents a solid improvement.

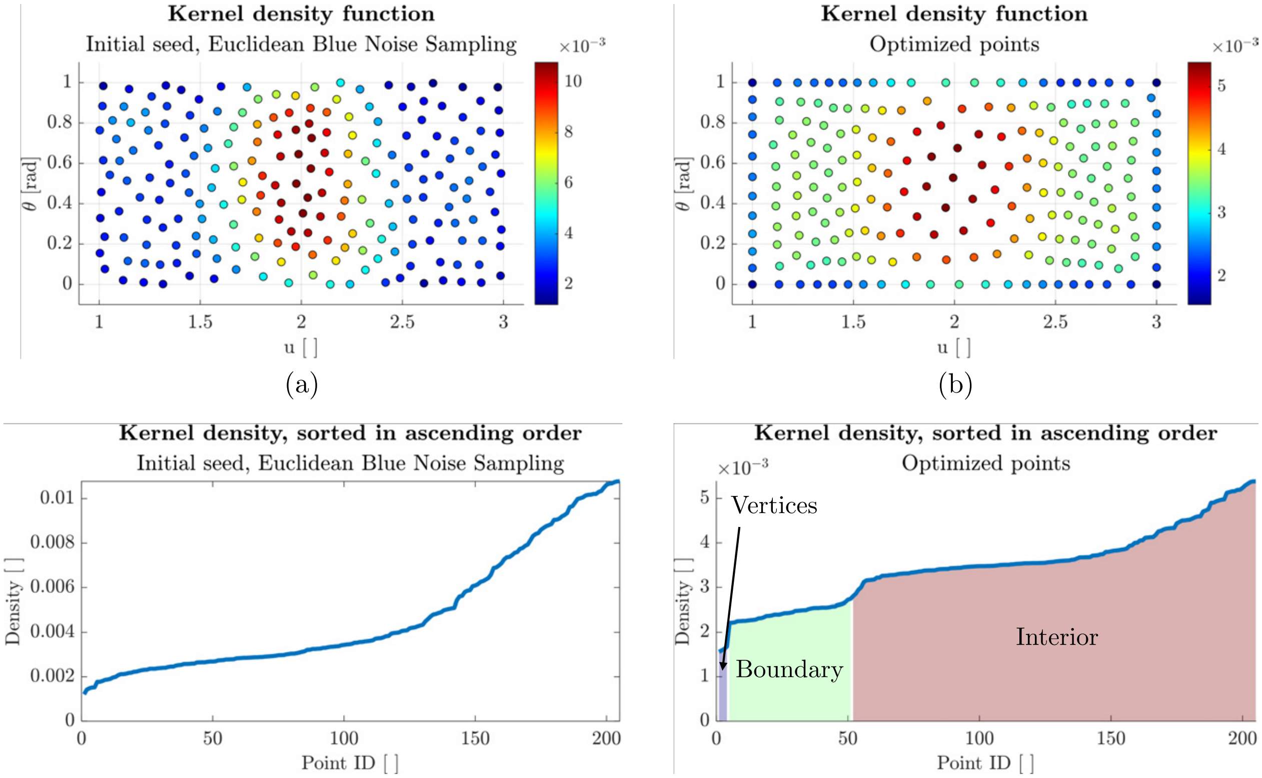
Lastly, a comparison of the three seed methods (Figure 5.28 and 5.29) shows that geodesic blue noise seeding yields the smallest final standard deviation but has the longest runtime. Euclidean repulsion and Euclidean blue noise are faster to compute, with slightly higher final standard deviations. Table 5.1 summarizes all the above results, including runtimes, average distances and standard deviations. Overall, all three methods confirm that the final distributions maintain quasi-uniform density.



(a) (b)

(c) (d)

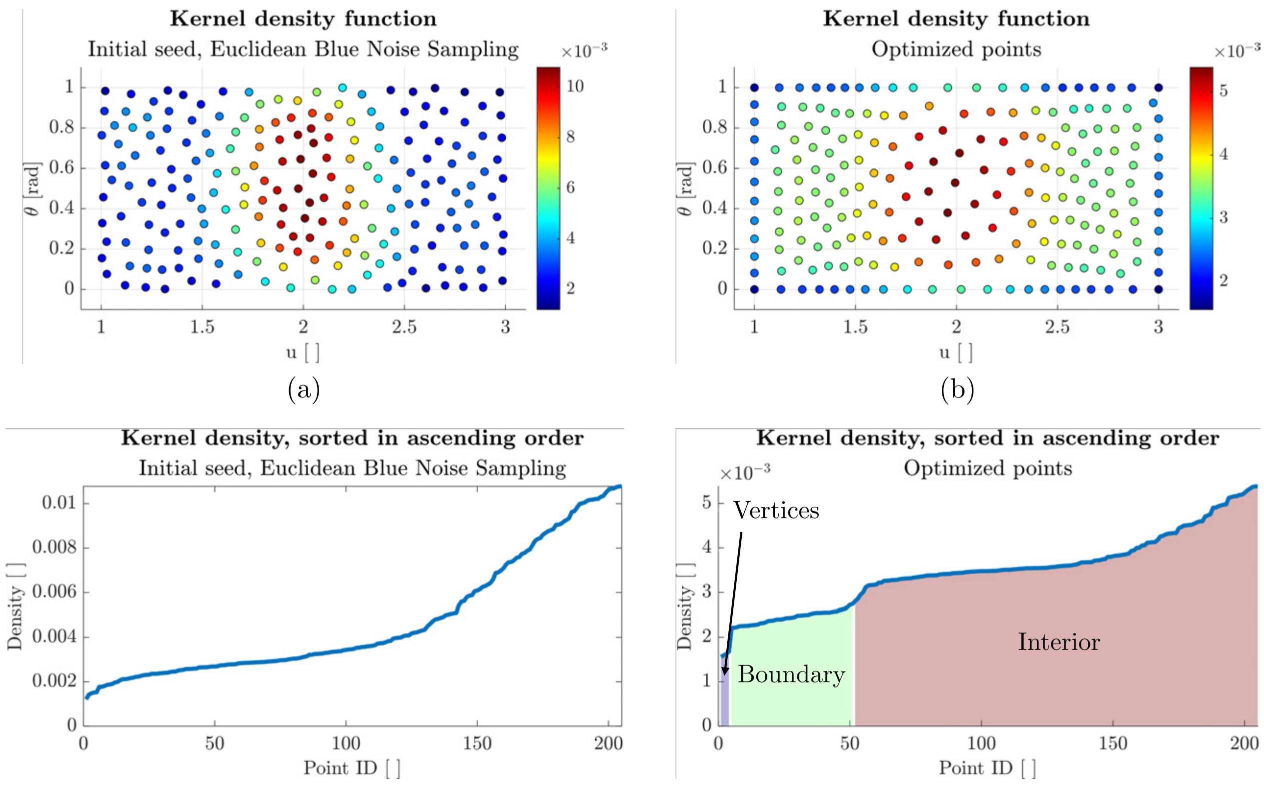
Figure 5.24: Pronounced hourglass platform, domain Ω_3 ($u \in [1, 3], \theta \in [0, 1], \rho_c = 10\%$, $n = 1$). (a) Initial Euclidean blue noise seed in $\{u, \theta\}$. (b) Optimized distribution in $\{u, \theta\}$. (c) Seed on the platform. (d) Optimized distribution on the platform.



(a) (b)

(c) (d)

Figure 5.25: Kernel density evaluation for Ω_3, comparing initial seed and final optimization. (a, b) Color-mapped distributions; (c, d) ascending density plots.



(a) (b)

(c) (d)

Figure 5.26: Smooth hourglass distribution using the geodesic blue noise initial seed method. Kernel density evaluation for Ω_3, comparing initial seed and final optimization.

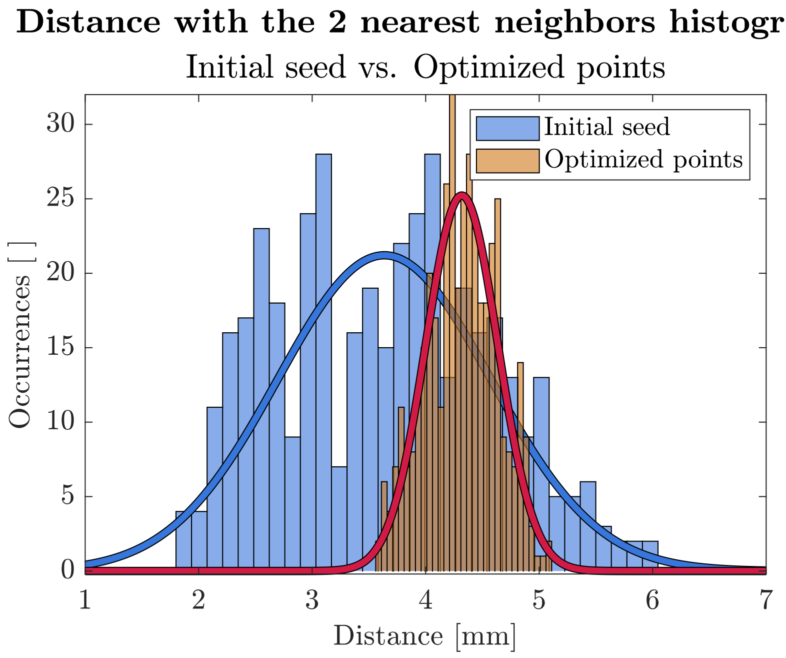
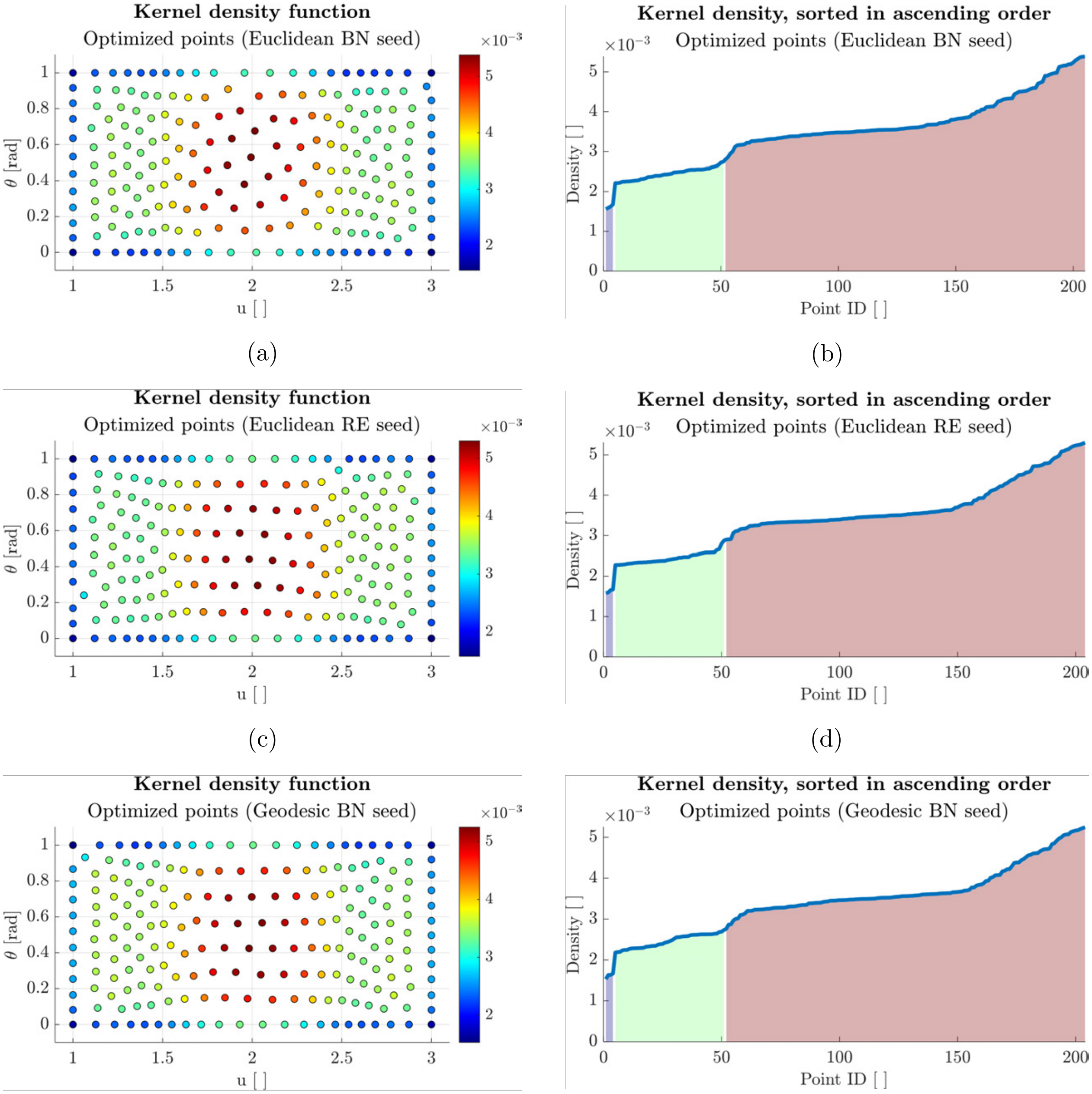


Figure 5.27: Distance histograms for Ω_3, contrasting Euclidean blue noise seeding (blue) and final solution (orange).

Case		**Mean [mm]**	**Standard deviation [mm]**	Runtime [s]
Pronounced hourglass	EBNS initial seed	3.648	0.930	5.8
	GRE optimization	4.334	0.340	2419.4
	ERE initial seed	3.836	1.038	742.9
	GRE optimization	4.359	0.284	2665.0
	GBNS initial seed	3.840	0.500	2312.0
	GRE optimization	4.384	0.278	3810.8
Smooth hourglass	EBNS initial seed	3.685	0.973	0.75
	GRE optimization	4.432	0.302	439.8
	ERE initial seed	3.839	1.155	110.1
	GRE optimization	4.522	0.310	626.6
	GBNS initial seed	3.933	0.521	290.5
	GRE optimization	4.560	0.232	521.1

Table 5.1: Mean, standard deviation, and runtime values for all three initial seed methods on both hourglass platforms.



(a) (b)

(c) (d)

(e) (f)

Figure 5.28: Comparison of optimized distributions when using different seed methods: (a, c, e) final point placements, and (b, d, f) corresponding sorted density plots.

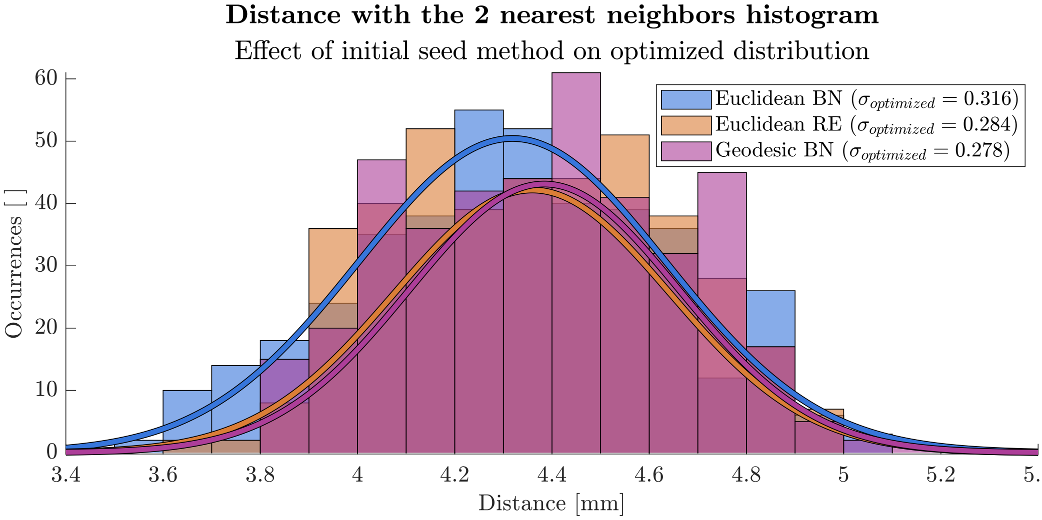
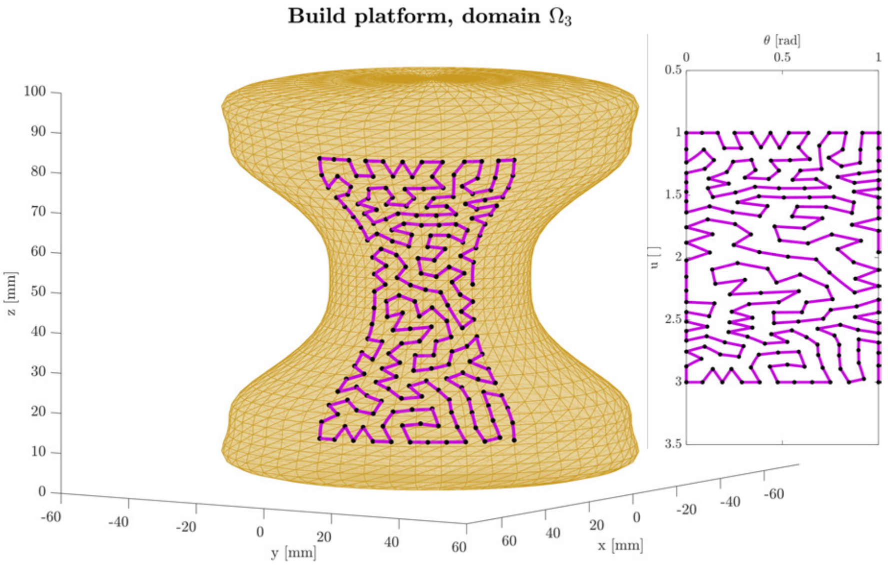


Figure 5.29: Distance histograms for all final distributions: Euclidean blue noise (BN), Euclidean repulsion (RE), and geodesic blue noise (BN).

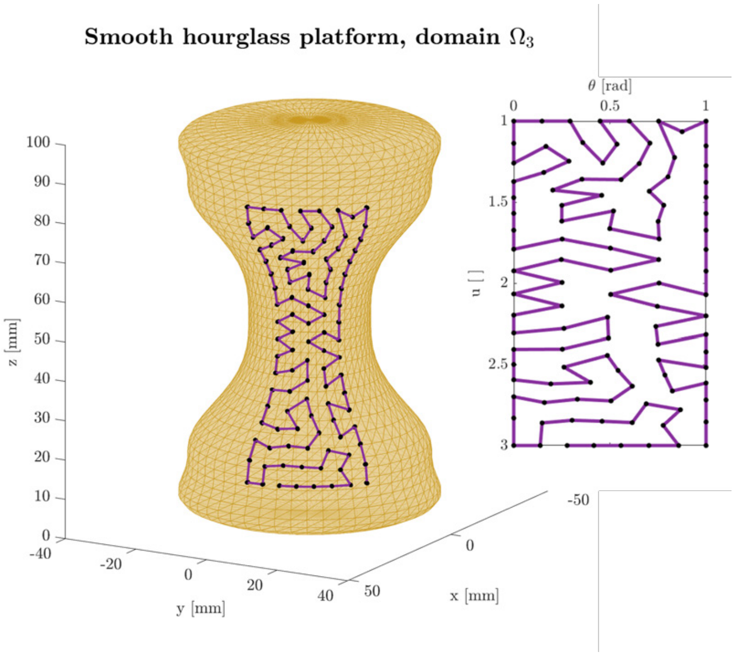
Path planning with 2-opt nearest neighbor

Finally, two path planning examples are shown in Figures 5.30 and 5.31, where the Nearest Neighbor heuristic (with 2-opt) links the optimized points in a Hamiltonian path. The geodesic equation (5.10) is solved between consecutive points. Future improvements could target boundary constraints or part-specific optimizations, but these examples demonstrate how the final uniform distribution translates into a feasible infill trajectory.



Build platform, domain Ω_3

Figure 5.30: Geodesic path planning results for the pronounced hourglass platform, domain Ω_3, shown in $\{u, \theta\}$ (right) and in Cartesian coordinates on the platform (left).



Smooth hourglass platform, domain Ω_3

Figure 5.31: Geodesic path planning results for the smooth hourglass platform, domain Ω_3.

5.7 Discussion

The present algorithm does a good job at providing a non-solid infilling solution in NPAM contexts. However, the methodology still leaves open some research gaps that will be addressed in the near future from the publication of this Thesis.

The first and foremost is the difficulty of scaling this algorithm onto a part comprised of many layers. The main boundary condition for this issue is achieving infill coherence. This means new layers must be supported by the previous one. This condition can be achieved up to several degrees, by either having a slight support of the previous layers (relying on the maximum bridging distance) or by fully supporting the previous layers (as found in planar infills). Several options are thought to solve this problem.

The first option revolves around parametrically determining the slice manifold for $h > 0$. This could be done by approximating the offset curve by a set of degree-six Bézier curves. This set can contain as many segments as desired, up to a preferred tolerance or total square error between the offset and the approximation. From this point onward, the application of the method is straightforward.

This method comes with some drawbacks. First and foremost, the way points are scattered in subsequent layers have no knowledge of the positions of points in previous layers. Additionally, the path planning becomes essentially random, so the generated rasters may or may not be fully supported by the previous layer. This methodology falls into the first category of support, where the infill relies on the maximum bridging distance inherent of the process. Another drawback relates to runtime. As previously seen in this Chapter, even though some numerical optimizations have been implemented, the runtimes for one single layer are borderline practical. If this manifold fitting algorithm is implemented, it is possible that runtimes may exceed the limit of application.

A second option revolves around the usage of Pythagorean hodograph curve (PHC) approximations [113]. PHCs have an interesting property that their derivative (hodograph, as it may be interpreted as a velocity) is a perfect square polynomial, resulting from the sum of squares of x and z components (hence Pythagorean). PHCs have been historically used in CNC trajectory planning owing to their algebraic properties. If a degree-six Bézier composite spline is approximated by a set of C^3-continuous at the knots PHCs, then the calculation of the offset curves is algebraic (i.e., without square roots in between that lead to numerical optimization to solve equations) [114]. This type of solution has been slightly introduced in other robotic environments [115]–[118].

This type of approximation largely solves the numerical limitation posed by the previous method, and opens the door for semi-analytical bijectivity checking methods. Being able to know whether a point is located in the null space of the homeomorphism would be invaluable for the algorithm and overall workflow efficiency. Owing to the algebraic property of PHCs, it would be possible to describe the homeomorphism null space, which is the non-bijective region. This null space is defined as the 4-tuples of points $\{u_1, u_2, h_1, h_2\}$ that fulfil the following relationship:

$$\mathcal{O}(u_1, h_1) - \mathcal{O}(u_2, h_2) = 0 \tag{5.24}$$

Where $\mathcal{O}(u, h)$ is an offset of the generatrix, defined by the following equation:

$$\mathcal{O}(u, h) = \mathcal{S}(u) + h\mathcal{N}(u) = \mathcal{S}(u) + h\frac{\nabla \mathcal{S}(u)}{\|\nabla \mathcal{S}(u)\|} \tag{5.25}$$

If $\mathcal{S}$ is a PHC, then the above equation would be rational. Given the homeomorphism presented in Chapter 4 where the curve is contained in the $-y = 0$ plane, this null space can also be delimited by the following $K - 1$ relationships for a composite curve comprised of K segments:

$$\left(\frac{\mathcal{S}_{k+1}^T[\nabla \mathcal{S}_k]}{\|\nabla \mathcal{S}_k\|} + \frac{\mathcal{S}_k^T[\nabla \mathcal{S}_{k+1}]}{\|\nabla \mathcal{S}_{k+1}\|}\right) \cdot (-\mathbf{e}^{\mathbf{y}}) = 0 \qquad \forall k \in [1, K-1] \tag{5.26}$$

With all previous conditions in place, it is thought to be possible to ascertain the homeomorphism null space, hence definining plausible $h > 0$ slices and knowing whether a part can be homeomorphed without checking point by point. Still, these optimizations, while valuable, do not provide maximum infill cohesion, where subsequent layers are fully supported by the previous ones.

An upgrade is proposed to solve this: extending the definition of the metric to the complete embedded map $\{\varphi\}$. This means that the metric tensor can be represented as a $\mathbb{R}^{3\times3}$ matrix, depending on all map coordinates $\{u, \theta, h\}$. Thus, the geodesic repulsion algorithm may be applied in volumetric domains. All concepts presented in this Chapter are extrapolable to volumes, such as the definition of the geodesic distance, the repulsion force, and the number of points to scatter. This last generalization can be done since the integral of the metric's determinant over a volumetric domain results in the volume of said domain.

At first glance, this framework looks incompatible with the notions of uniform slicing provided in this Thesis. Additionally, the geodesic repulsion actually encourages points to stay far away from each other. This may appear as counterintuitive, regarding the problem of infill cohesion. However, the infill generation and the infill slicing are two decoupled problems that can be solved in any order. Thus, volumetric infilling is introduced.

Volumetric infilling is a technique that aims to repeat a volumetric infill pattern (based on a volumetric base cell), as opposed to a planar infill pattern. This has been realized in a plethora of fashions, among which the use of Triply Periodical Minimal Surfaces (TPMS) has gained significant attraction [119]. These TPMS are often based on crystallographic units, which have specific symmetries.

Crystallographic units, and therefore TPMS base cells, are often inscribed in a cube. This is not ideal for the present methodology, as there is no way to ascertain a whole number of cubes can be fit into the resulting point scattering. An ideal shape for these base cells is the tetrahedron. Tetrahedrons can fill a volumetric space, albeit not with regular tetrahedrons. The objective is to find surfaces akin to TPMS that can fit in a base tetrahedron, accounting for the changes in symmetry group. This includes a requirement for non-chirality.

In regions tiled with cubes, any vertex is connected to an even number of cubes (with the exception of region corners). This can be expressed as a connected graph, where every node is linked to either one node, or an even number of nodes. This means that this graph is always 2-colorable, as it is a bipartite graph [120]. Hence, a 2-colorable graph allows for chirality. If one were to replace the tiling cubes with tetrahedron, the resulting graph may no longer be a bipartite graph, and its chromatic number may become 4.

With this result in hand, two possible solutions arise. The first one revolves around defining four differently "colored" (in the sense of symmetry) tetrahedron cells to fill the space. Another, more feasible solution, is the creation of an achiral tetrahedron cell with triple symmetry. In this regard, recent advances [121] have been made that point in this direction, with close application in the present Thesis.

Chapter 6

Experimental setup: machine modeling and hardware implementation

To support the systematic procedure of defining NPAM trajectories on a robotized cell, a experimental setup in the form of manufacturing equipment has been studied, developed, and implemented. This Chapter covers a study of the robotic manipulator's kinematic precision and accuracy to better understand the capabilities of the robot utilized. This study will serve to ascertain the expected upper bound on robot accuracy, as well as a design criterion for the location of the rest of the experimental setup elements. Additionally, this Chapter provides a brief description of design notions employed in the design of the manufacturing equipment developed in this Thesis. Lastly, this Chapter concludes with pictures of the final NPAM robotized cell, which will be utilized for the development of the workflow and the realization of NPAM-MEX-P parts presented in Chapter 7.

6.1 The UR10 robot

The serial robotic manipulator utilized in this PhD Thesis is the Universal Robots UR10, a collaborative robot present in the Manufacturing Laboratory at ETSI Industriales, Universidad Politécnica de Madrid. A summary of its most relevant specifications is shown below in Table 6.1.

Specification	**Value [units]**
Weight	28.9 [kg]
Payload	10 [kg]
Reach	1300 [mm]
Repeatability	±0.1 [mm]

Table 6.1: Technical specifications for Universal Robots UR10 collaborative robot [122].

The UR10 serial robotic manipulator is an ideal option in highly experimental scenarios, owing to its collaborative safety features. Therefore, low velocity collisions with operators or the cell elements may suppose no harm overall in case an error has been done. However, its comparatively low repeatability may become an issue, especially if high pose accuracy is required. This may be the case in the context of RBNPAM-MEX-P processes, especially when working with nozzle diameters under 1 [mm]. Additionally, this repeatability often refers to the stationary pose repeatability, which is the robot's ability to achieve a certain pose consistently. This may differ from on-route repeatability (often related to the robot's control and overall dynamic performance), and overall pose accuracy (related to the robot's ability to reach a known point in its environment with respect to a global reference frame). These measurements are often not provided by the manufacturer, and will be studied in the next Section.

The UR10 robot is programmed in several main ways:

- Utilizing the teach pendant, **online teaching**. In this method, an operator manually guides the robot to each waypoint in the trajectory. This method is often the preferred one when the need for automation is low, as the method's flexibility is equal to the time needed by the operator to record each new point individually. The trajectories presented in this Thesis, and in NPAM processes, can include millions of points, rendering this method unsuitable.
- **Offline programming**. In this method, points are programatically coded, along with other instructions such as moving, activating PLC outputs, or reading PLC inputs. This method is suitable for trajectories with large amounts of waypoints, as the codes can be generated through specialized software (RoboDK, RobotStudio, etc.), or by the user's algorithm (`Splineifyer`, and other algorithms developed in this Thesis). However, this method often struggles with safety and accuracy, as the trajectories are agnostic of the robot's environment and whether it changed (be it drastically or minimally).
- **Hybrid programming**. This is the preferred method when trajectories include a large amount of points and accuracy is a requirement. Hybrid programming involves generating a parametrized set of waypoints, which is adjusted at the moment of execution. To achieve this, the trajectories must be located with respect to a fixed Tool Center Point or Work Coordinate System, which is calibrated at the beginning of the operation. This is the method employed in this Thesis, as it can ensure the best accuracy results. The calibration methods are explained in Chapter 7.

In addition to these concepts, there are several practical methods to incorporate these movement commands into a proper robot program. These methods are:

- **Block-based programming** through the teach pendant. This is the most basic way of programming the robot, and its functionality is oriented towards simpler robot programs based on online teaching. This method is discouraged by the large amount of points of NPAM trajectories.
- **URScript**. URScript is the proprietary code developed by Universal Robots. URScript is a barebones version of Python, with no possibility of including libraries and limited mathematical functionality. This is the type of output specialized robotics software often give, and may accommodate for both offline and hybrid programming. This is the

preferred method for the purposes of this Thesis.

- **Open communication from a TCP/IP socket**. This last method is the lowest-level type of program inputting. This is the preferred method when utilizing higher-level interfaces in a PC, be it for simultaneous data acquisition, or when utilizing middleware such as ROS or ROS2. This is the goal towards this Thesis is oriented to. To achieve this type of command, there is an ongoing collaboration with CAR-CSIC-UPM members to implement the presented workflow in a ROS2 architecture.

Lastly, there are several commands that achieve robot movement, each with its peculiarities. In URScript, these commands are `movej`, `movel`, and `servoj`. These commands are compatible with all programming modes.

- `movej` is the configuration-to-configuration interpolation mode movement. This type of movement linearly interpolates from one configuration to other, in the configuration space. This type of movement is often unsafe due to its unpredictability. It is often difficult to picture how the robot will move in the configuration space, and therefore, it is difficult to ascertain whether the robot will clash with itself or with other cell elements.

- `movel` is the pose-to-pose interpolation mode movement. This is often referred as the Cartesian path, which results from the linear interpolation between two Cartesian poses in a rectilinear fashion. This movement mode is predictable, and often, the desired behavior. This movement mode is used in most of the workflow and results depicted in Chapter 7. Additionally, this type of movement is configured through tangential linear/angular velocities and accelerations. The robot will try to achieve command velocity following a linear acceleration ramp, then decelerate when it is about to reach the destination waypoint.

- `servoj` is a more advanced version of `movej`. `servoj` allows the user to directly tinker with the control parameters, namely, time to destination and joint stiffness. This movement is linearly interpolated in the configuration space, and unpredictable in turn. However, it allows for continuous interpolation among waypoints through a lookahead time parameter.

  This type of movement, albeit complex to adjust, is the preferred method when dealing with large amounts of points Cartesian-wise close to each other. In this type of context, `movel` is unable to reach the command velocity in a short amount of time, inducing vibrations in the robot and overall a poor dynamic behavior. `servoj` is able to supersede this limitation, creating a smooth trajectory through a set of points. This is the type of movement utilized for the final result in Chapter 7.

In order to properly dimension the robotized cell, and to know the minimum accuracy results in the future calibration and NPAM tests, a model of kinematic precision and accuracy has been developed in the next Section.

6.2 Kinematic precision and accuracy modeling

This Section outlines a comprehensive methodology developed to characterize and analyze the kinematic uncertainty of the Universal Robots UR10 collaborative robot. The study seeks to quantify the global positioning error of the robot by examining local joint errors and how they propagate through the robotic arm's kinematic chain. The results obtained through this analysis serve a double purpose: establishing the higher accuracy bound provided by the UR10 robot in the RBNPAM process devised in this Thesis, and providing a quantitative rationale to positioning the different hardware elements of the station.

6.2.1 Study hypothesis

The uncertainty of each robot joint is propagated through the kinematic architecture onto the end effector [123]. Hence, this kinematic uncertainty varies with each configuration, and may be expressed as a 6DOF uncertainty for each pose. Once the individual robot joint uncertainty is determined, it is possible to express the kinematic uncertainty (and therefore, the robot pose repeatability and accuracy with respect to a global reference frame) through an uncertainty ellipsoid. This may be expressed through the matrix $\mathbf{A_c}$ for each test configuration $\mathbf{c}$ of which data have been obtained. For each test point in the workspace, a semi-definite positive uncertainty matrix $\mathbf{A_c}$ can be calculated. This matrix, when diagonalized, provides the semi-axes length and direction of the uncertainty ellipsoids. These ellipsoids encapsulate both the magnitude of the uncertainty as well as the accurate-most and uncertain-most directions for movement. To extend this definition to the complete workspace, it is possible to define the following equation:

$$(\mathbf{x}-\mathbf{c})^T\mathbf{A_c}(\mathbf{x}-\mathbf{c}) = 1 \tag{6.1}$$

Where $\mathbf{x}$ is the query configuration, and $\mathbf{c}$ is the closest test configuration, accompanied with its uncertainty ellipsoid $\mathbf{A_c}$. The following Subsections provide an explanation on how the experiments were designed to obtain the $\mathbf{A_c}$ matrices (based on [123]), as well as the results to incorporate these $\mathbf{A_c}$ onto the hypothesis above described.

6.2.2 Test methodology

An oscillatory motion profile was developed for each joint to isolate and study its individual behavior. This trajectory consists of a sine wave superimposed on a linear ramp, designed to span a range of -2π to 2π radians over a duration of 2000 seconds. The only exception to this is the movement proposed for joint θ_2, which, due to the presence of the worktable, was limited to a range of $-\frac{5}{6}\pi$ to $\frac{5}{6}\pi$ The motion ensures frequent directional changes, enhancing the detection of backlash and mechanical inconsistencies. An example of this trajectory is portrayed in Figure 6.1. The trajectories were executed in each joint by means of `movej`.

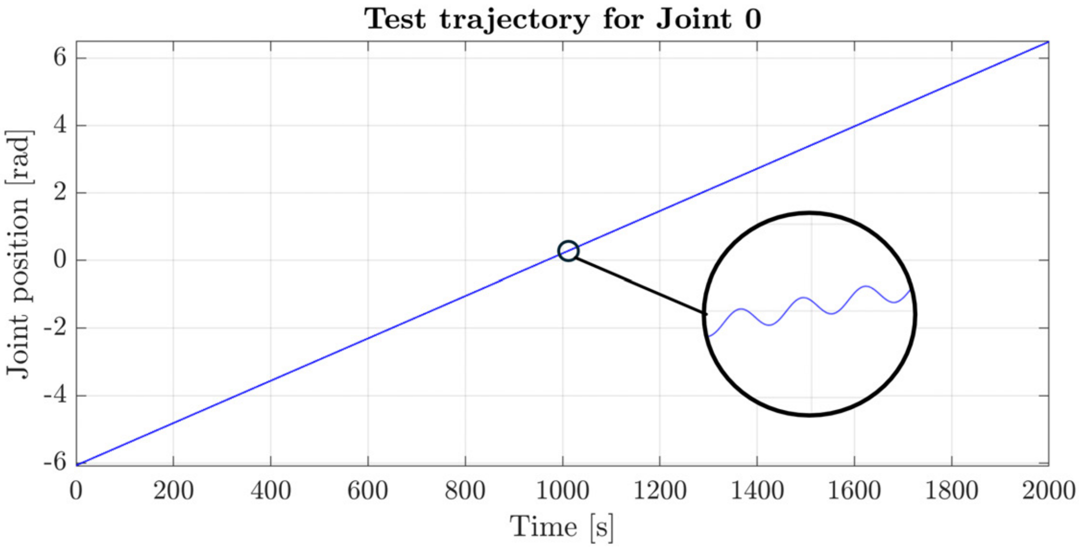


Figure 6.1: Example of oscillatory trajectory for kinematic repeatability.

Two methods were employed to measure the actual movement of each joint and detect deviations from the ideal trajectory, MPU6050 accelerometer-gyroscope sensors and the software UR LogViewer. MPU6050 sensors provide six degrees of motion tracking (three linear, three angular) and were mounted using custom-designed brackets produced via additive manufacturing. On the other hand, UR LogViewer was utilized as a backup measurement method, as it is limited by the robot's communication interface. This interface is capped at a transmission rate of 125 [Hz], which may be insufficient for this type of measurement. The accelerometers were interfaced with an Arduino Uno, using custom code to calibrate sensors and stream data into Microsoft Excel via PLX-DAQ. This allowed high-frequency recording of motion data, offering finer resolution than the robot's internal sensors.

These empirical data provide the axes uncertainties with respect to configuration. To obtain these relationships, the difference between the command axis joint angle $\theta_{command}$ and the actual axis joint angle θ_{actual} is calculated, then plotted against $\theta_{command}$. This provides a map of axis joint error over axis joint angle. By applying a window of 100 samples ($\frac{1}{250}\pi$ [rad]), it is possible to obtain the standard deviation of the samples within the window. This provides the relationship between axis joint angle and kinematic uncertainty.

With joint data acquired and analyzed, the next step involved modeling global positioning error using MATLAB. The data was used to create a model that estimates how localized errors affect the tool center point (TCP) in Cartesian space. Each joint's angular error was used to derive geometric uncertainty in the form of ellipses (2D) or ellipsoids (3D). The robot's workspace was divided into 325 test points across 13 radial planes in a spherical coordinate system (Figure 6.2).

At each point, an uncertainty cloud was generated based on simulated errors, and a minimum-volume ellipsoid was computed to encapsulate the deviation zone. For each Cartesian point (with a configuration **c**), then for each axis, an estimation of kinematic error based on configuration is introduced, then the forward kinematics algorithm is applied. This provides a curve with a beginning and end point. The superposition of 6 curves (one for each of the 6 axis of the UR10) provides 12 curve extrema. An minimum volume enclosing ellipsoid may approximate these 12 points, and its decomposition in eigenvalues and eigenvectors provides the matrix $\mathbf{A_c}$. A flow diagram is represented in Figure 6.3.

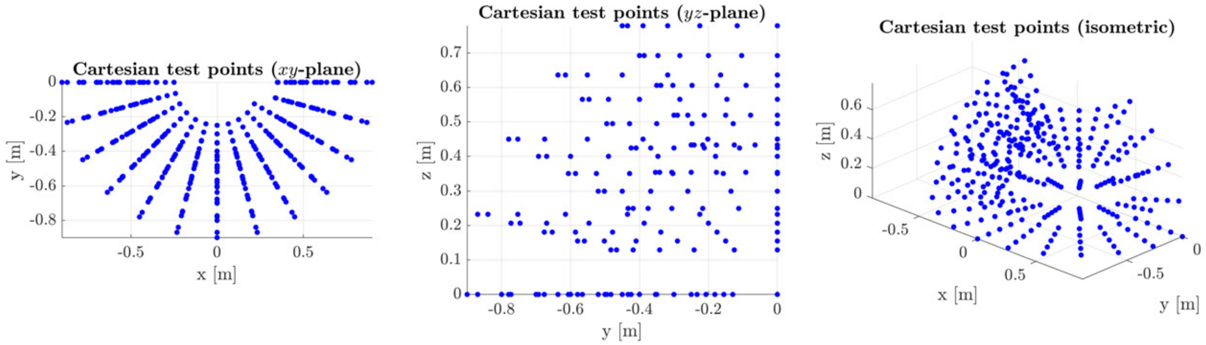


Figure 6.2: Cartesian points utilized for testing.

MATLAB code was developed to obtain the final kinematic uncertainty/accuracy results. The code solves inverse kinematics for all Cartesian targets above described. Once a configuration is found for each target, joint repeatability data are propagated to the tool center point by means of Jacobian-based error ellipsoids, after which a minimum–volume enclosing ellipsoid is fitted to the resulting cloud. By repeating the procedure in consecutive azimuthal slices that span from 0° to 180° in steps of 15°, the script builds a picture of how changing flange pose influences accuracy. Then, it is possible to request full three-dimensional ellipsoid clouds for qualitative inspection or, alternatively, allow the routine to condense the information into a planar heat map that summarizes the best attainable accuracy throughout the reachable surface.

Geometric and inertial data are imported from two Excel workbooks. The sheets labelled Kinematics and Dynamics hold the screw axes in space and body frames, the home pose $\mathbf{M}$, the spatial inertias $\mathbf{G}_i$ and the per-link frame transformations $\mathbf{M}_i$. This information is obtained from the robot's URDF (short for Unified Robot Description Format). This URDF perfectly defines the robotic manipulator as a whole, and is particularized for each individual robot through the default calibration procedure. This URDF is more reliable than a CAD model, since it is the result of factory calibration, and therefore includes slight imperfections and manufacturing deviations the robot might have.

These entities are reshaped with `cat(3,...)` so that they comply with the conventions used by the Modern Robotics toolbox [124]. Empirical repeatability measurements, previously described and stored in arrays called `X_JOINT0`, `Y_JOINT0` and so forth, provide one–sigma uncertainty values that will be interpolated for every configuration encountered. All subsequent calculations obey SI units. Gravity is set to $\mathbf{g} = [0, 0, -9.81][\text{m}\cdot s^{-2}]$ and all joint angles are expressed in radians.

The inverse-kinematic solver terminates when the angular component of the twist error falls below $\varepsilon_\omega = 0.001$ [rad] and the linear component drops beneath $\varepsilon_v = 0.0003$ [m]. Each azimuthal slice is initiated with a user-supplied seed posture, denoted `thetalist0`, which is

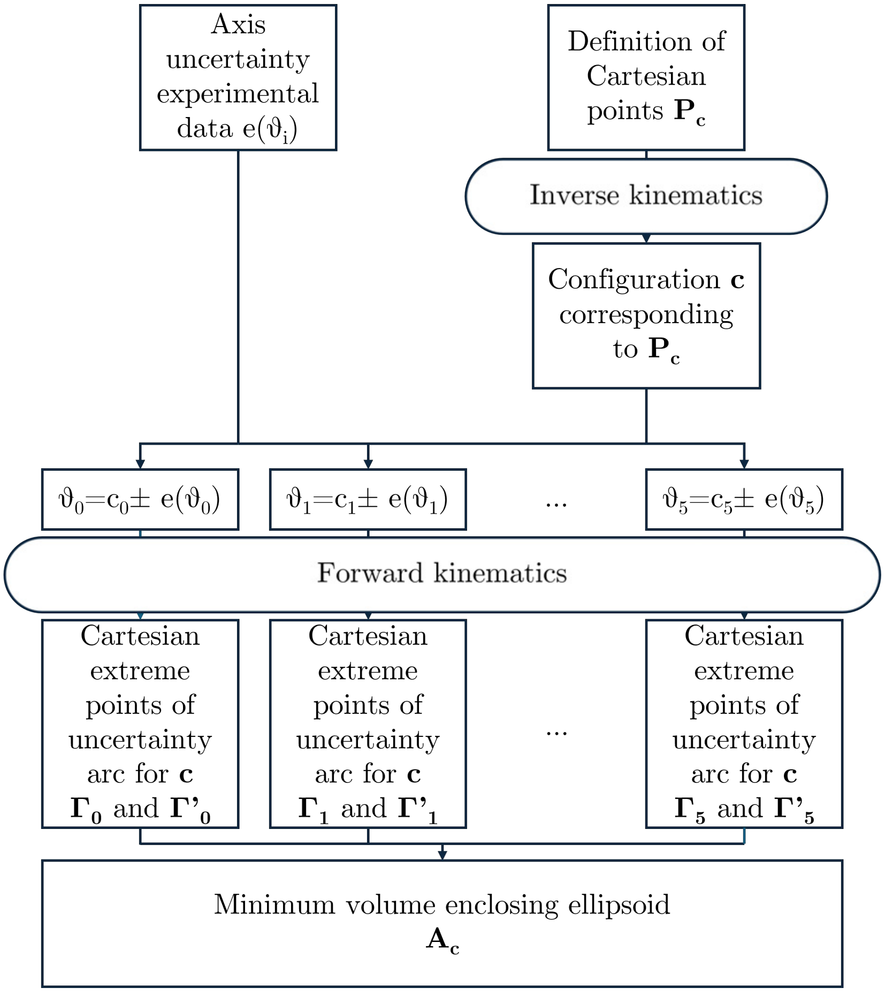


Figure 6.3: Flow diagram for minimum volume enclosing ellipsoid calculation.

replaced by the newly converged solution before the solver is called for the next neighboring point. This warm-starting strategy accelerates convergence markedly along a trajectory.

For every converged posture the program perturbs each joint by approximately 10° and recomputes the forward kinematics. The displacement vector from the unperturbed tool position to its perturbed counterpart is normalized so that it represents a unit sensitivity direction. A diagonal matrix, $M_\theta = \mathrm{diag}(|u_\theta|) \times 100$, scales those directions by the interpolated joint uncertainties, thereby producing twelve extremal points that outline a local error cloud around the nominal tool position.

The subroutine `MinVolEllipse` is used with a relaxation factor of $1\,\%$ to compute the smallest ellipsoid that encloses the twelve sensitivity points. The result is characterized by a center $\mathbf{c}$ and a symmetric, positive-definite shape matrix $\mathbf{A}$ that satisfies $(\mathbf{x} - \mathbf{c})^\mathsf{T}\, \mathbf{A}\, (\mathbf{x} - \mathbf{c}) \leq 1$. An eigen-decomposition of $\mathbf{A}$ yields semi-axis lengths $a_i = 1/\sqrt{\lambda_i}$ and their spatial directions, both of which are subsequently employed for visualization. The code also draws a red vertical segment whose length equals the largest positional spread along the z-axis, providing an intuitive indication of worst-case height error.

The workspace is explored slice by slice. The first slice covers indices 1 to 25 and corresponds to a flange yaw of 0° with an initial joint guess of $(-14.27, -55.5, 126.74, \ldots)$ degrees. The second slice spans indices 26 to 50 at 15° yaw and uses $(-3.12, -55.68, 127.2, \ldots)$ degrees as its starting posture. This pattern continues until the twelfth slice, which starts at index 299 and ends with the last target while holding the yaw at 180°; the seed posture for that final sector is $(150.29, -55.94, 163.4, \ldots)$ degrees. To avoid numerical issues the script clamps any joint that threatens to exceed its allowable range or approach a singular configuration.

A Boolean flag called `draw` allows the user to turn graphics on or off. When enabled, the program renders every computed ellipsoid together with its axis indicators and supporting whiskers. Color is mapped linearly from green to red, with green denoting high accuracy and red representing low accuracy. Once all slices have been processed, the query coordinates (X_S, Y_S, Z_S) are interpolated onto a regular grid of 500×500 nodes by bicubic interpolation. Regions that are unreachable by the robot, namely the rear half-disc behind the base, are blanked out by overlaying a white patch. The color-coded surface is then complemented by seven contour bands that delineate equi-accuracy regions, offering an immediate visual cue as to where the manipulator can work most precisely.

6.2.3 Results and Discussion

Firstly, the uncertainty results are presented for axes θ_0, θ_2, and θ_4, which corresponds to the base, elbow, and one of the wrists, respectively. The kinematic architecture of the UR10 features the same joint design for axes 0 and 1, as well as the same design for axes 3, 4, and 5. Therefore, it was hypothesized that these axes groups had approximately the same kinematic uncertainty behavior. The results for each of these joints are represented in Figures 6.4, 6.5, and 6.6.

The encoder models utilized in the UR10 axes belong to the MRA series from RLS. According to the axes encoder manufacturer [125], the encoder's resolution can range from $8 \cdot 10^{-4}$ [rad]

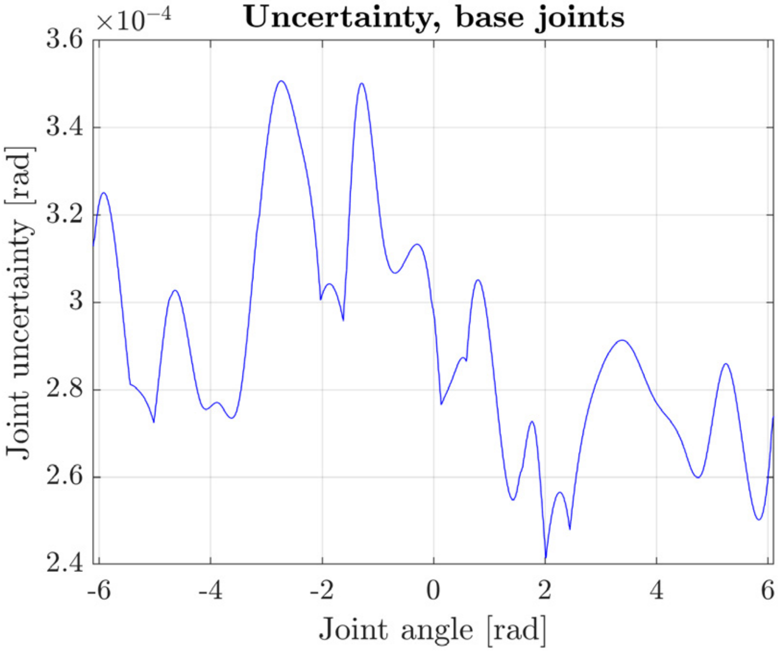


Figure 6.4: One-sigma uncertainty for base-like joints.

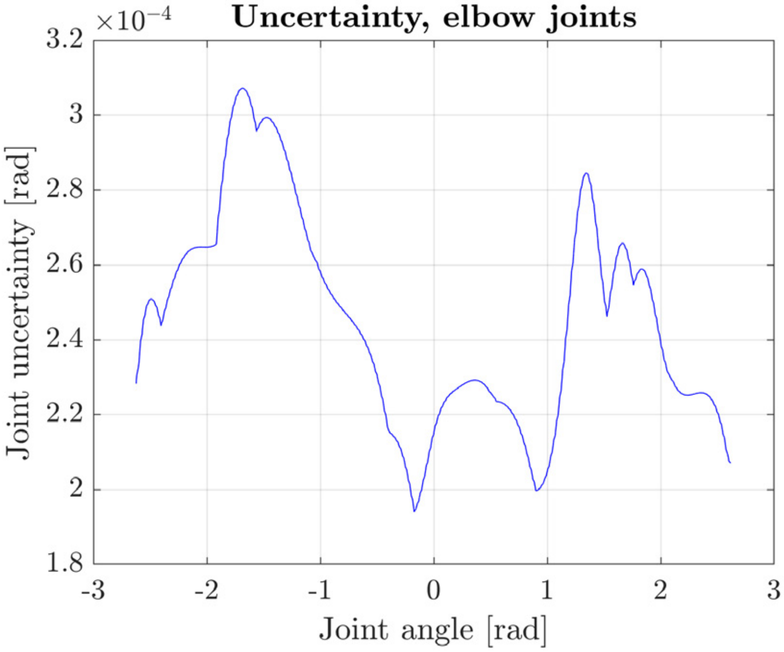


Figure 6.5: One-sigma uncertainty for elbow-like joints.

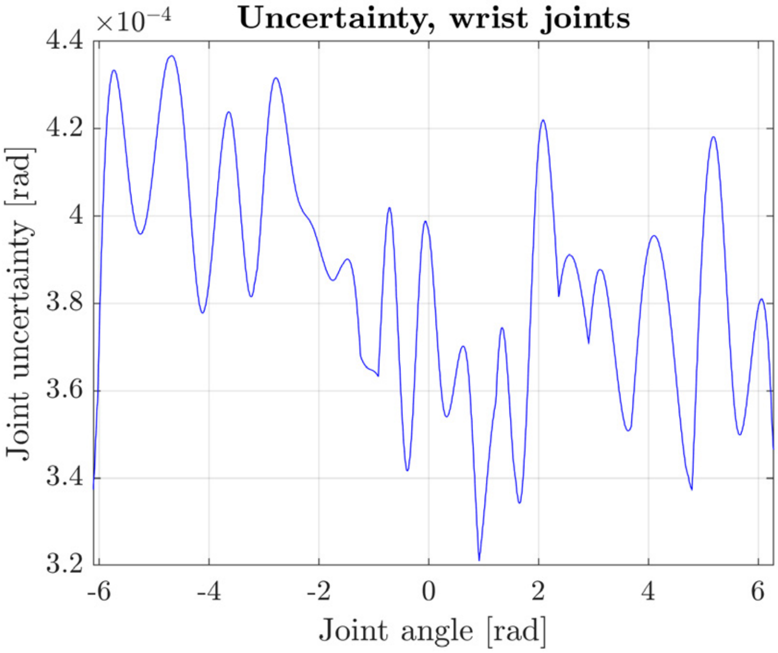


Figure 6.6: One-sigma uncertainty for wrist-like joints.

to $4 \cdot 10^{-4}$ [rad] or $8 \cdot 10^{-5}$ [rad], depending on the quality of calibration performed after assembly. The presented data oscillates in this range, validating the result. These uncertainty results are introduced in the global uncertainty model.

Three types of results will be shown to represent the global uncertainty results according to the model previously presented: maximum and minimum uncertainty directions, uncertainty heatmaps and uncertainty ellipsoids. Figures 6.7 to 6.18 show the maximum and minimum uncertainty directions, as well as the minimum uncertainty maps for each slice. Only the slices from 90 [deg] to 165 [deg] are shown, as these are the allowable locations for the calibration and extruder fixture.

The results show that the one-sigma uncertainty heatmaps have values above 0.018 [mm]. This value is greater than the repeatability established by the manufacturer, who indicates a stationary repeatability of ±0.1 [mm] [122]. This uncertainty discrepancy could be owed to the difference between stationary repeatability (the ability to reach a certain point in Cartesian space after enough time has been given) and on-route repeatability (the ability to follow a certain path with enough accuracy) [126]. The latter is not frequently analyzed, and has profound implications in the context of robotic manufacturing as a whole. Additionally, this discrepancy suggests that axes velocities play a role in characterizing the kinematic accuracy. A sensible solution for this may lie in state-space black-box characterization, utilizing methods similar to the ones proposed in [127].

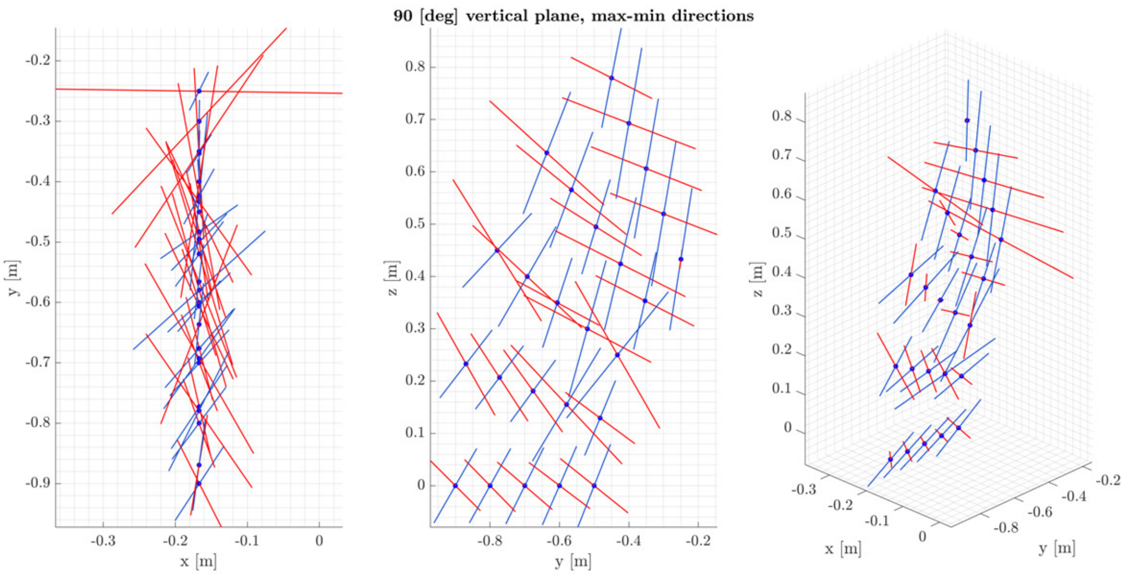


Figure 6.7: 90 [deg] vertical plane (the middle part of the work table) maximum (red) and minimum (blue) uncertainty directions.

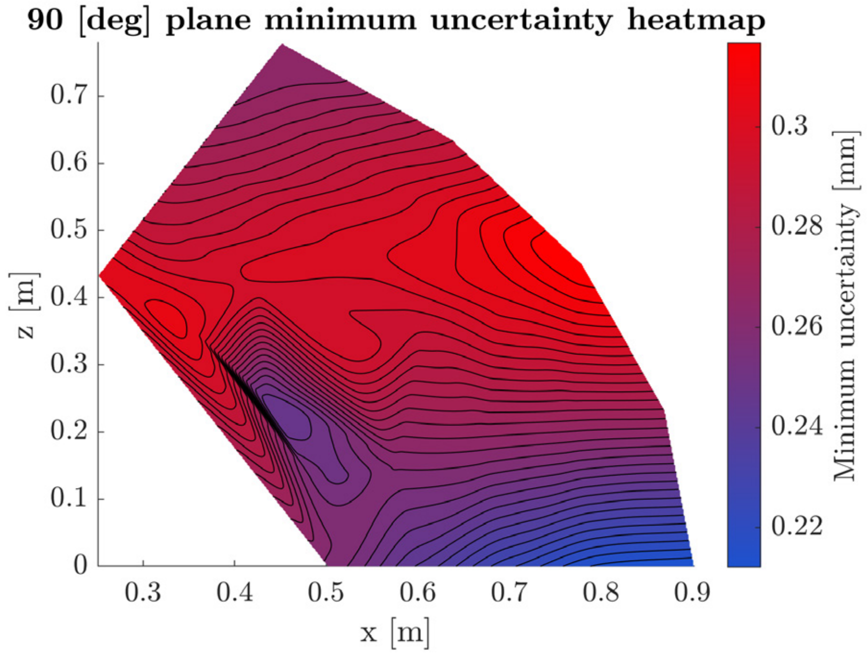


Figure 6.8: 90 [deg] vertical plane minimum uncertainty heatmap.

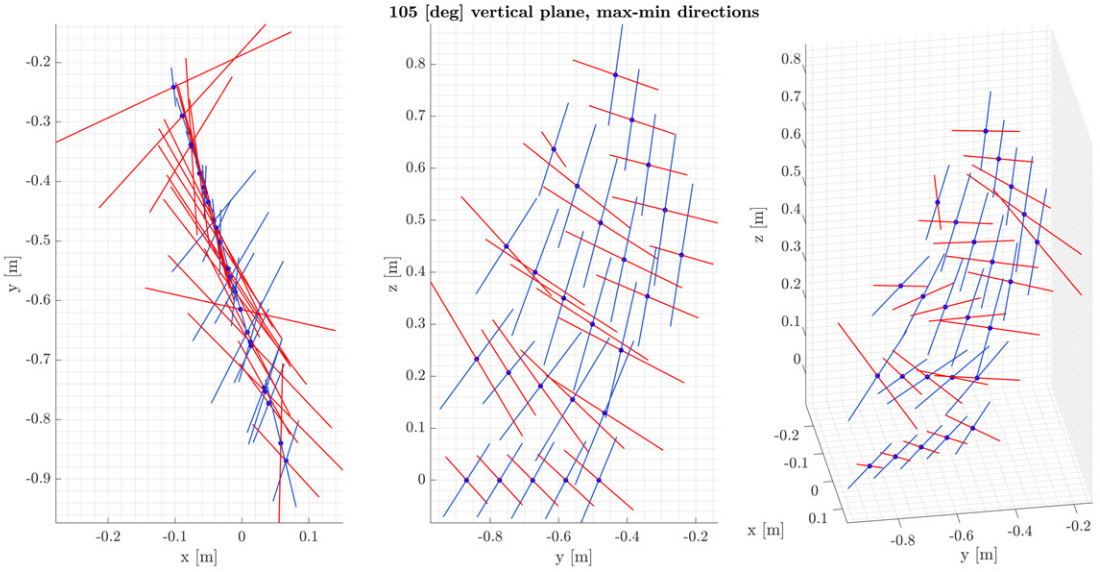


Figure 6.9: 105 [deg] vertical plane maximum (red) and minimum (blue) uncertainty directions.

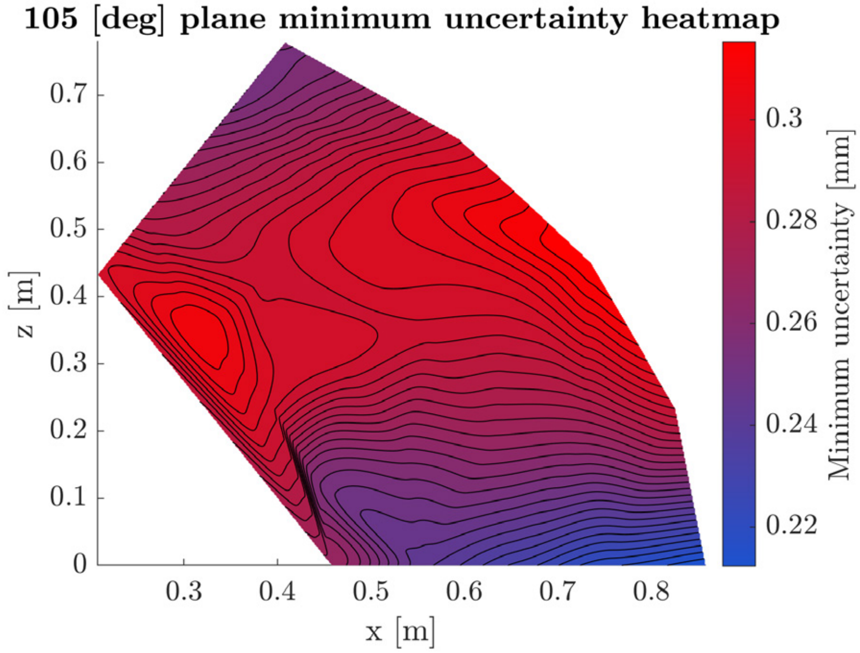


Figure 6.10: 105 [deg] vertical plane minimum uncertainty heatmap.

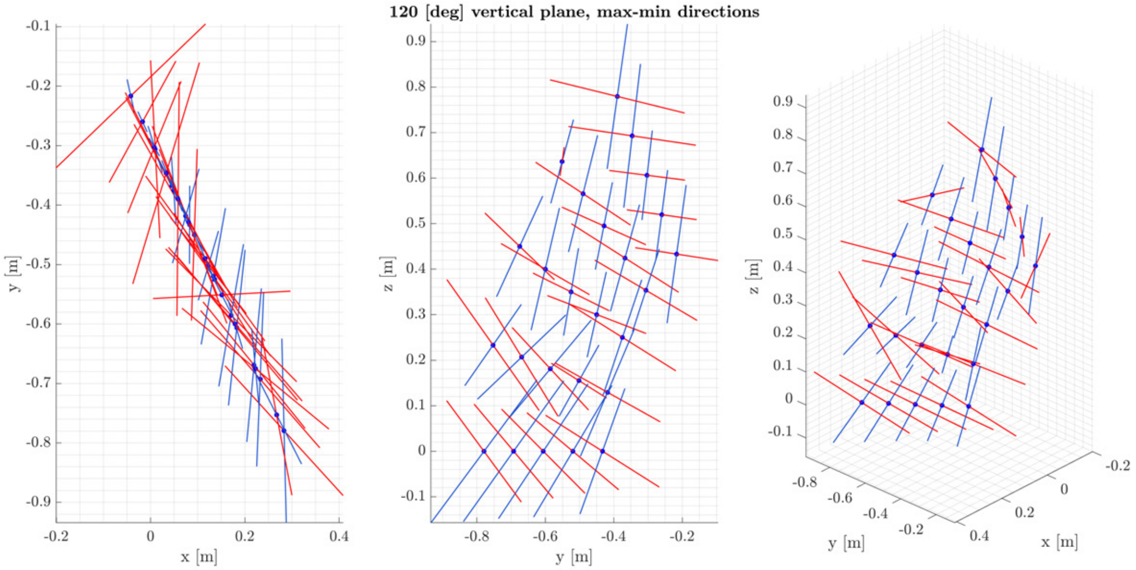


Figure 6.11: 120 [deg] vertical plane maximum (red) and minimum (blue) uncertainty directions.

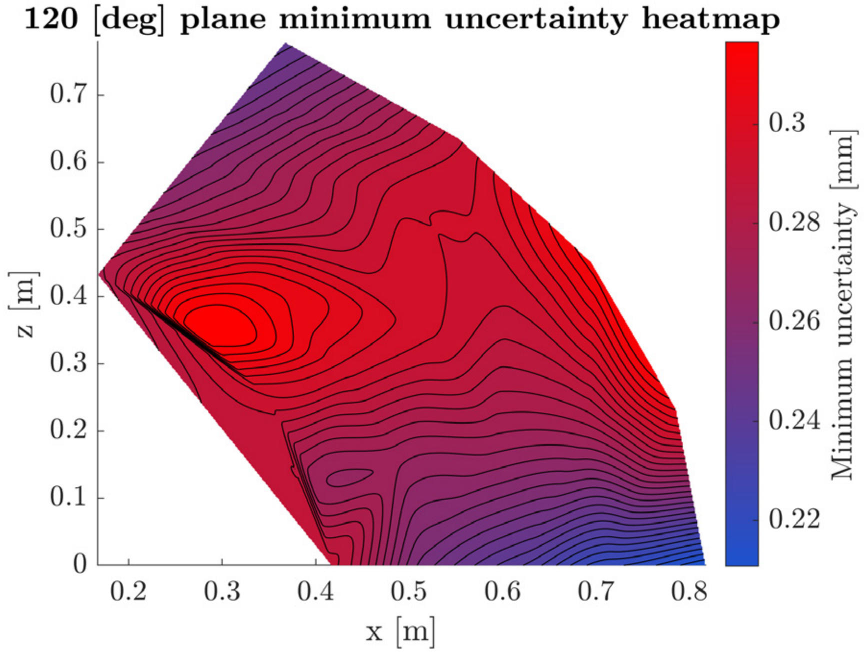


Figure 6.12: 120 [deg] vertical plane minimum uncertainty heatmap.

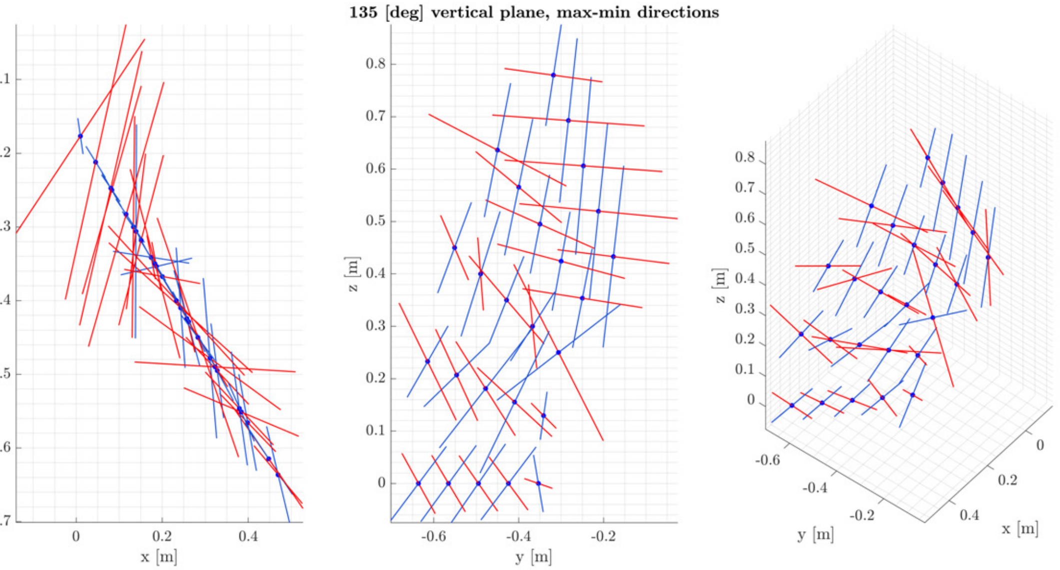


Figure 6.13: 135 [deg] vertical plane maximum (red) and minimum (blue) uncertainty directions.

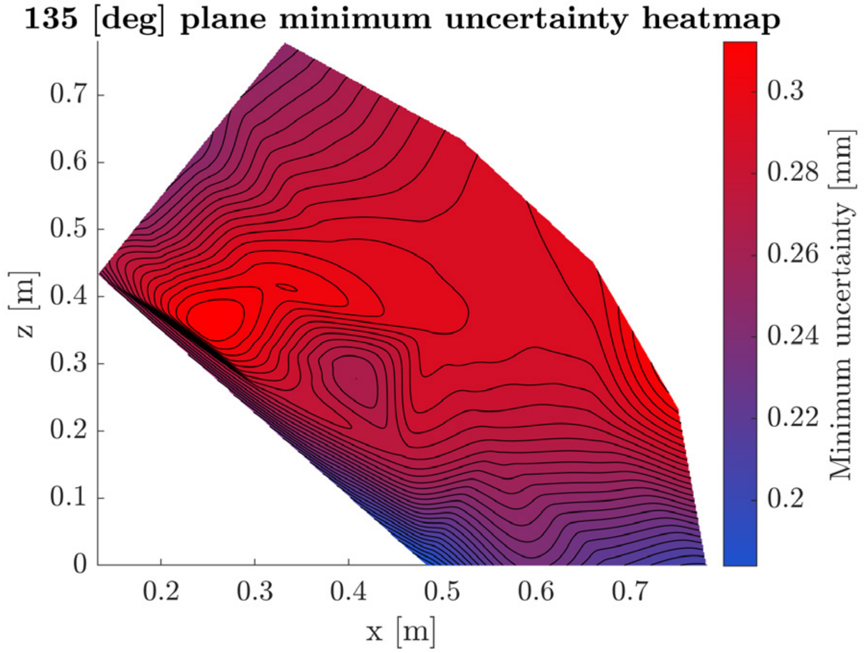


Figure 6.14: 135 [deg] vertical plane minimum uncertainty heatmap.

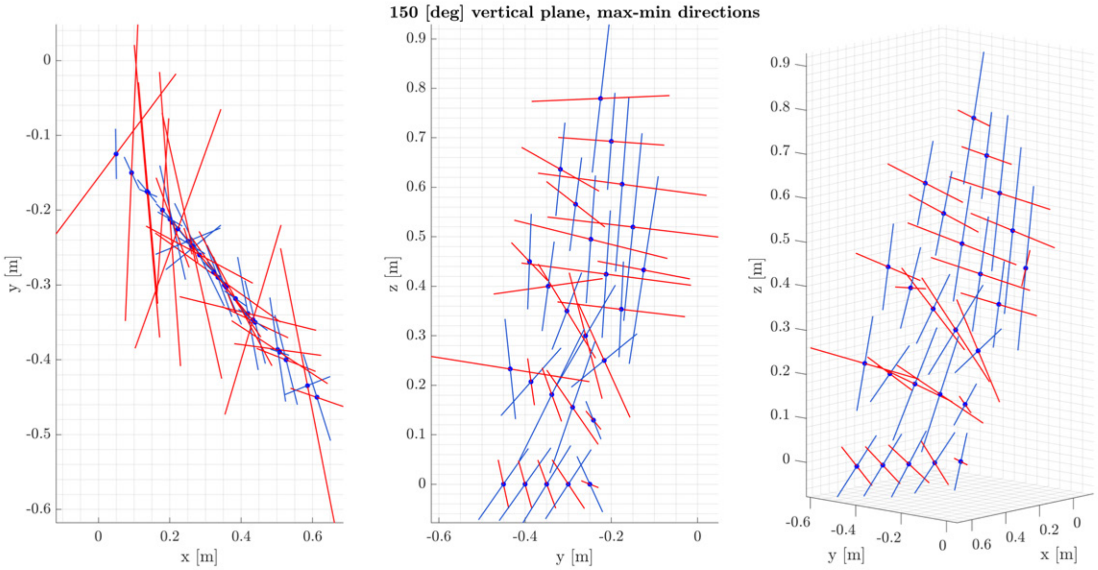


Figure 6.15: 150 [deg] vertical plane maximum (red) and minimum (blue) uncertainty directions.

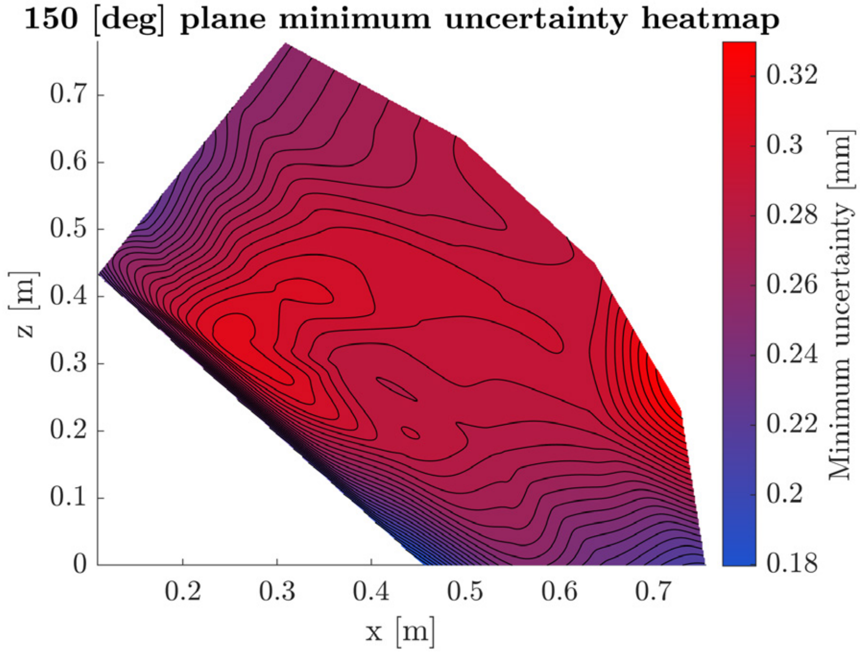


Figure 6.16: 150 [deg] vertical plane minimum uncertainty heatmap.

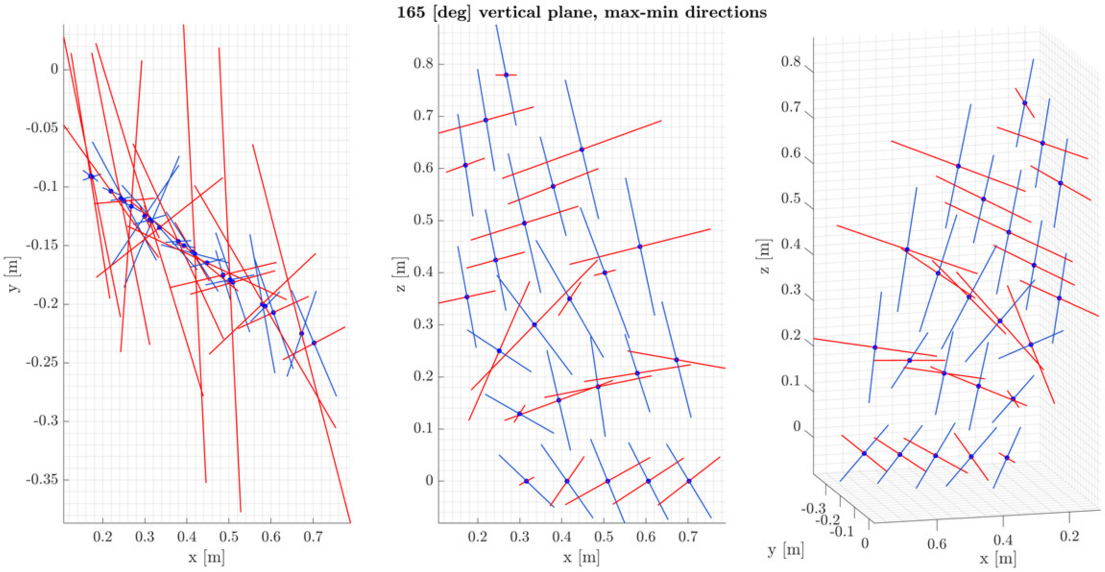


Figure 6.17: 165 [deg] vertical plane maximum (red) and minimum (blue) uncertainty directions.

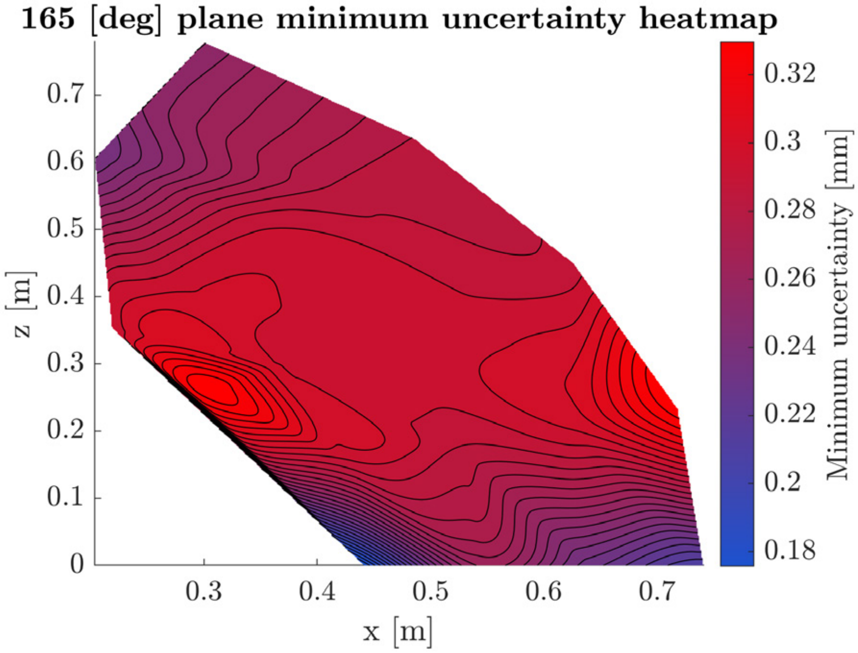


Figure 6.18: 165 [deg] vertical plane minimum uncertainty heatmap.

Additionally, these results serve as a design criterion. Figure 6.19 shows the uncertainty heatmap on the most favorable direction, as well as the uncertainty ellipsoids for each of the query points. The extruding and calibration fixture is to be put in the East side of this map, in the slices comprehended between 90 [deg] and 165 [deg]. Several candidates are shown in Figure 6.20, denoting both location and orientation. These locations take into account the robot's manipulability and try to provide a suitable location for the calibration and extrusion fixture. The three candidates are location A ($\{0.4, -0.8, 0.32\}$, oriented diagonally in the xy-plane), location B ($\{0.2, -0.85, 0.32\}$, oriented along the y-direction), and location C ($\{0.6, -0.2, 0.32\}$, oriented diagonally). These points are located in minimum uncertainty regions of the workspace, and oriented along minimum uncertainty directions as much as mechanically possible.

Of these locations, the preferred ones are A and B, owing to the bore pattern already present in the worktable. In this regard, the bore pattern would make the design unwieldy, and take up a significant space on the work table. Between A and B, location A shows the least uncertainty while still providing a simple design, whereas location B trades uncertainty for the most ease of design.

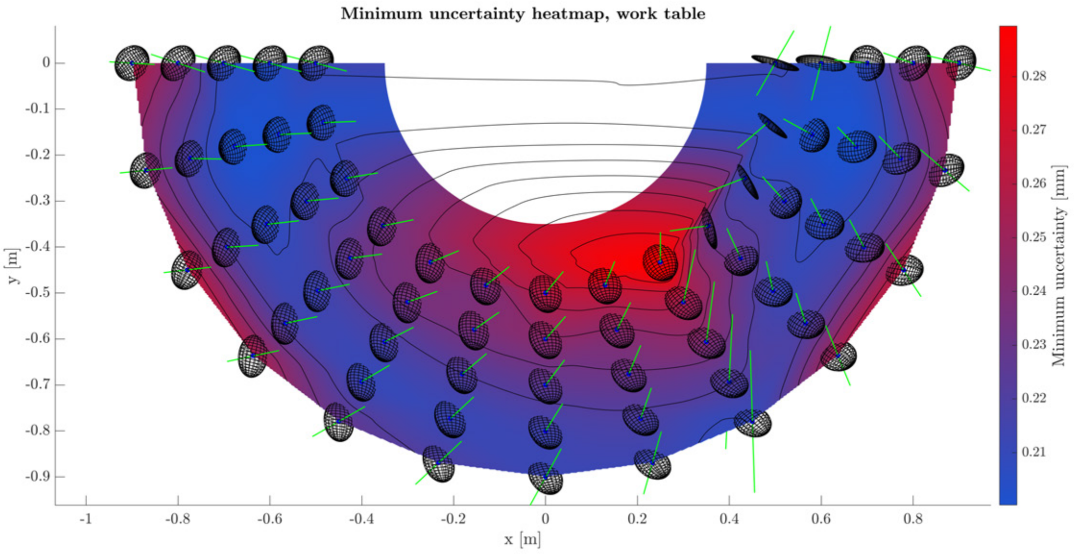


Figure 6.19: Minimum uncertainty map of the work table. The uncertainty ellipsoids are overlaid on top of it, with the least uncertain directions depicted in green.

In summary, the output of this methodology has direct implications for workspace planning, calibration routines, and trajectory optimization. By identifying zones of higher or lower uncertainty and understanding their directional characteristics, the robot's usage can be refined for precision-sensitive tasks. The work provides a foundation for real-time error compensation strategies and adaptive motion planning in collaborative robotics.

The study proceeded in two stages. The first comprised experimental trials to quantify joint-level kinematic error; the second used those results, together with the robot's kinematic model, to predict global positioning error. Stage 1 met all objectives. Joint errors were

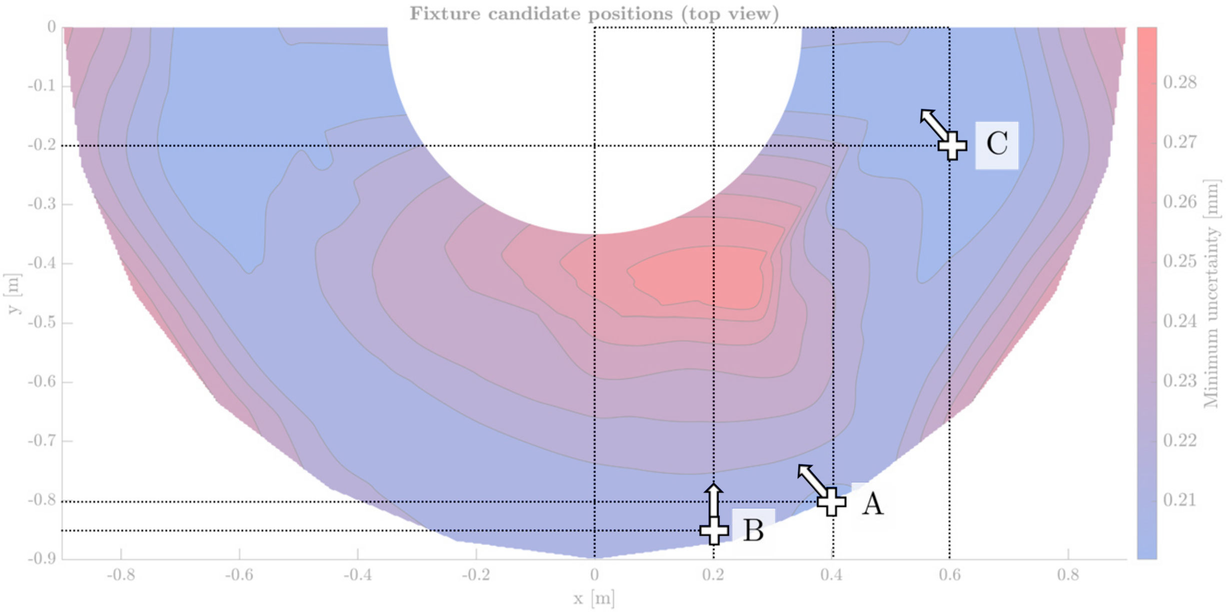


Figure 6.20: Candidate locations for the extruding and calibration fixture.

measured with in-house and commercial tools, and the processed data showed deviations well within the manufacturer's tolerances, with even tighter uncertainty bounds. Stage 2 implemented an error-estimation algorithm that produced the expected uncertainty ellipsoids. Predicted position errors at multiple test points agreed with both the experimental values and the manufacturer's specifications. From these ellipsoids, the principal directions and magnitudes of uncertainty were extracted, confirming that the robot's workspace contains regions whose accuracy varies with task trajectories.

The resulting model lets users either place a task at its optimal pose or, if relocation is impossible, align the operation along the axis of minimum error. Because the procedure depends only on forward and inverse kinematics, it is transferable to any manipulator for which these relationships are known. The framework therefore lays the groundwork for future optimization of robotic path planning and may form part of a broader research program in robotics for manufacturing.

As explained, all presented results are utilized in the following Section, Hardware implementation, to define the position of NPAM manufacturing elements, tooling and fixtures. This following Section will cover the design and implementation of these elements, providing insights on the how's and why's of the presented manufacturing equipment in the context of NPAM.

6.3 Hardware implementation

The NPAM station utilized for the development of this Thesis has been designed with a central working principle in mind: a serial robotic manipulator orients and locates the build platform, attached to it, with respect to a fixed extruder and calibration apparatus. This concept is presented in Figure 6.21. The following Subsections will cover the design and implementation of all the necessary elements for the station, including the build platform and its control, extruder control, power supply elements, the calibration and extruder fixture, and the overall result.

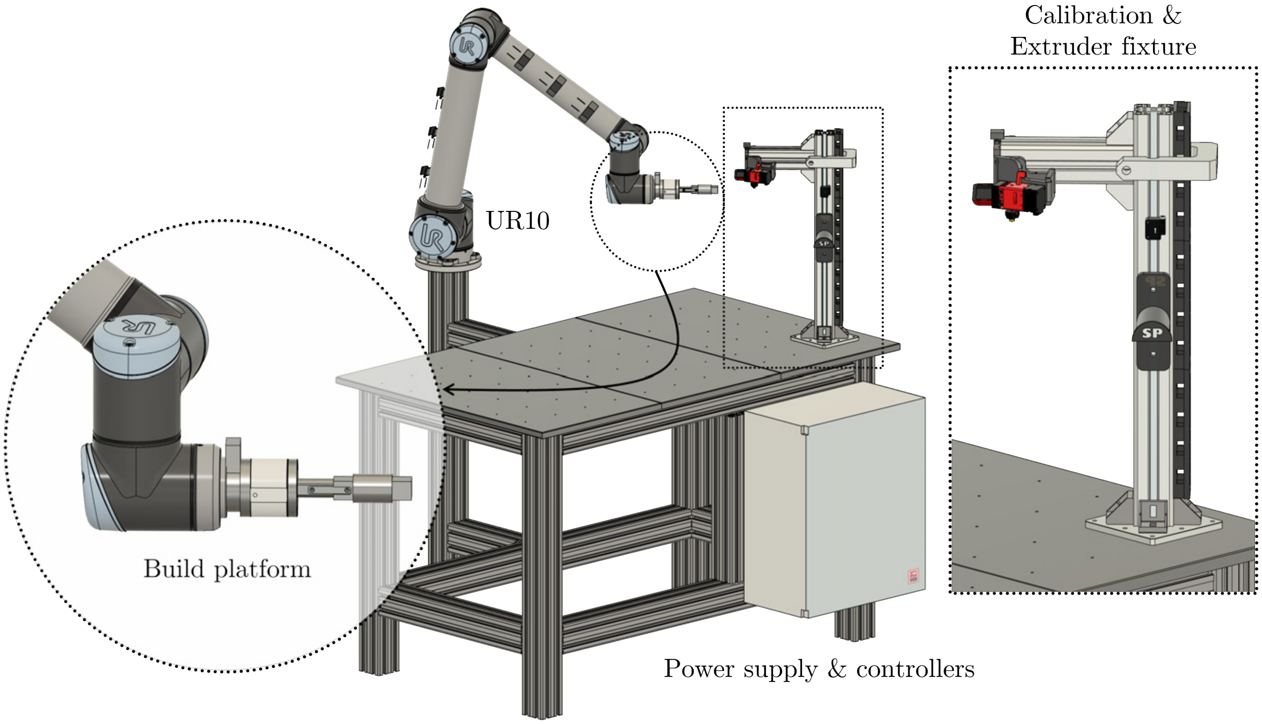


Figure 6.21: CAD representation of the NPAM environment concept developed in this Thesis.

6.3.1 Build platform design and control

The design process begins with defining five primary functional groups (FGs) that encapsulate the system's key roles:

- **FG1:** Heating the platform – Ensures the platform surface reaches and maintains the build platform temperature, critical for first layer adhesion during MEX-P processes.
- **FG2:** Temperature monitoring – Continuously measures the platform temperature to ensure safety and process reliability.
- **FG3:** Thermal protection for the robot – Prevents heat from damaging the robotic actuator by insulating the platform's mounting interface.
- **FG4:** Waterproofing – Prevents liquid infiltration into the platform internal components, which may be required during post-processing or cleaning.
- **FG5:** Alignment between parts – Guarantees precise positioning and calibration through mechanically robust connections with minimal clearance.

Each FG is then mapped to several potential functional solutions (FSs). For instance, for FG1, solutions range from conventional heated plates and cartridge heaters to more novel approaches like infrared heaters or heat exchangers. FG2 includes sensor options such as thermocouples, RTDs, thermistors, and thermal cameras. Thermal insulation for FG3 involves a broad array of materials (nylon, teflon, foam, rock wool, etc.) and cooling methods. FG4 features mechanical seals and design strategies to ensure water-tightness, while FG5 examines mechanical mating options like conical or cylindrical contacts and clearance-free screw patterns.

To select among competing FSs, each FG is evaluated using a weighted criteria system. Criteria include factors such as heating efficiency, ease of integration, assembly simplicity, thermal resistance, mechanical rigidity, manufacturing feasibility, and cost-effectiveness. This leads to quantitative comparisons using decision matrices.

For FG1, cartridge heaters were selected owing to their heat transfer capabilities, integration ease, and compatibility with the cylindrical bed. Similarly, FG2 showed that thermocouples best balanced performance, precision, and integration simplicity. In FG3, both nylon and teflon were deemed appropriate FSs, though a final choice is deferred to later stages after FEM modeling. For FG4, o-rings were deemed the most reliable and cost-effective sealing method. Finally, conical fitting for FG5 proved the best approach for alignment due to its repeatability and ease of implementation and manufacturing.

Following the selection, several conceptual sketches were created to explore assembly strategies. These focus on avoiding thermal conduction through the structure by isolating metallic components and securing them using mechanical fasteners that do not compromise the integrity of the insulating materials. The design integrates three main parts: the aluminum build platform, the nylon/teflon insulator, and a metallic anchor piece. The connection between parts employs conical mating surfaces and uses through-holes in the insulator to avoid threading in soft materials. Multiple alternatives were explored for the cartridge and thermocouple placements to optimize thermal performance and measurement accuracy while respecting geometric and functional constraints.

After the conceptual sketches are done, the engineering process is conducted. In this phase, the sketches are translated into detailed component designs supported by fundamental calculations. The primary goal is to develop digital 3D models of all the parts comprising the heated platform system. This involves selecting the thermal insulation material and determining the optimal position of the heating cartridge.

The process begins with a thermal study aimed at solving three issues: the required power of the heating cartridge, the minimum necessary thickness and material of the thermal insulator, and the ideal cartridge placement within the aluminum bed. A parametric study was conducted using COMSOL Multiphysics to simulate 12 scenarios that combined three heater powers (20 W, 30 W, and 50 W), two insulation materials (nylon and teflon), and two cartridge positions (centered or offset). The model included all relevant bodies (bed, cartridge, insulation) and considered thermal conduction, initial and boundary conditions, and radiative losses.

Simulation results indicated that only the 50 W cartridge met the heating time constraint of under three minutes, necessary to complete the robot's calibration process (detailed in Chapter 7). Nylon, although slightly inferior to teflon in thermal resistance, was ultimately selected due to availability. A conservative insulation thickness of 12 mm was chosen, later increased to 13.37 mm for safety in prolonged operation. Regarding cartridge placement, the centered position produced more uniform heat distribution across the bed, reducing temperature gradients and improving functional performance. The COMSOL results were validated through convergence analysis, confirming simulation accuracy.

With the simulation conclusions guiding the physical design, the next step involved 3D modeling. Standard Allen screws, existing lab stock dimensions, and ease of machining were prioritized in all decisions. The system consists of several parts, of which only the three most relevant are presented:

- **Anchor:** A cylindrical aluminum part that connects the nylon insulator to the robot's mechanical interface. It features a conical recess to support the insulator and threaded holes for attachment. A hollow vertical passage routes the heater and thermocouple cables internally, avoiding exposure to the process zone. Design iterations improved manufacturability by extending the cone bore through the piece for cleaner CNC tool paths.
- **Nylon Insulator:** This component provides thermal isolation. It features conical inner and outer surfaces to interface with the bed and anchor, respectively. The distance between these surfaces was optimized for insulation, according to the thermal simulation conducted previously. The piece was designed with features like poka-yoke flat spots for assembly and internal radii for machinability.
- **Build Platform:** As the most critical element, the platform integrates all core functionalities—heating, alignment, and mechanical support. Two initial designs were explored: one with a threaded shaft and nut and another with an internally threaded hole. The final design combined both concepts to achieve thermal cable routing and mechanical alignment. The central shaft with a through-hole allows heater and thermocouple cables to be routed cleanly, minimizing thermal noise and avoiding interference with the print area. A metal mesh sleeve was added to the thermocouple wire to reduce electromagnetic interference from the heater.

The fully assembled system is designed to interface with a robotic coupler, maintaining thermal isolation while supporting mechanical precision. It also allows internal cable routing, which reduces hazards and simplifies handling. All design iterations focused on minimizing the number of parts, simplifying manufacturing, and ensuring a clean, robust assembly. The final build platform is showcased in Figure 6.22.

Figure 6.22: Cylindrical build platform to be used as basis for NPAM processes.

Of note, the methodology presented in this Thesis not only works for cylindrical cases, but any differentiable manifold embedded in the Cartesian ambient space. Another limitation that can be thought of is the part extraction. If a part spans more than 180° in the θ-direction of the platform, then extraction may become impossible. To solve both issues, a procedure is proposed. First, the geometry of the final build platform is extruded utilizing a soluble material (i.e., poly(vinyl alcohol), or PVA) following cylindrical slices on top of the presented build platform. Then, on top of this soluble geometry, the actual part is manufactured following the designed slice geometry. After this process is done, the whole assembly (part, soluble support lattice, and the cylindrical build platform) is submerged in solvent. This is the reason for FG4: Waterproofing previously outlined. After the support has completely dissolved, the actual part can be extracted without problems.

After the thermomechanical design of the build platform, the heating cartridge controller was designed and implemented. The approach follows the same philosophy as the previous device. Firstly, eight FGs derived from engineering requirements and user needs were identified, each corresponding to a subsystem or role within the overall control device.

- **FG1:** Process and transmit information – A microcontroller must receive, process, and forward signals from a thermocouple to both a power control module and the central CPU.

- **FG2:** Measure temperature – Critical for both safety and feedback control, this function ensures real-time monitoring of platform temperature.
- **FG3:** Power control for the heater – This involves regulating a 230V AC signal to deliver the required 50W to the resistive heating element.
- **FG4:** External power input for electronics – Supplies the proper low-voltage DC power to the internal circuit board.
- **FG5:** Connection to high-voltage source – Interfaces the unit with the 230V AC grid for heater operation.
- **FG6:** External interface with the controller board – Provides physical ports for data and power without exposing internal components.
- **FG7:** Thermocouple connection – Interfaces the thermocouple with the control electronics securely.
- **FG8:** Enclosure – Encases and protects all electronic components while offering ventilation and modular connectors.

Each FG is supported by several FS, evaluated against specific criteria such as ease of integration, safety, cost, reliability, and form factor. Solutions include commercial off-the-shelf microcontrollers, analog-to-digital converter modules, voltage regulation methods, and connector types. Each FS was evaluated using a multi-criteria decision matrix where weights were assigned to relevant design considerations like electrical compatibility, integration effort, mechanical constraints, and future scalability. Each solution was scored numerically, and total scores guided final selections. Notably, the design avoided redundant complexity and prioritized components commonly used in embedded systems and maker platforms. The FSs selected were the following:

- For **FG1**, the NodeMCU ESP32 was chosen due to its high performance, built-in Wi-Fi, and expansion capabilities. It stood out as the most balanced option despite a slightly higher cost and size.
- **FG2** was addressed with the MAX31855 thermocouple-to-digital converter, selected for its compact size, precision, and ease of integration via Serial Protocol Interface.
- In **FG3**, a relay was selected over MOSFETs and operational amplifiers due to its simplicity, robustness, and suitability for closed-loop control at high voltages.
- Both **FG4** and **FG5** favored the IEC connector for its standardization, safety, and ease of enclosure integration.
- For **FG6**, a USB screw-mount socket was preferred to simplify connections with external devices while ensuring stability.
- **FG7** employed a type K female thermocouple connector to ensure compatibility and secure mechanical connection.
- **FG8** selected a rectangular prismatic enclosure due to its manufacturing simplicity and superior internal layout compatibility.

This thermal study, part of the Basic Engineering phase, aims to define how to control the heated bed temperature using a relay, determine the temperature profile of the heating cartridge over time, assess whether it exceeds the operational limits of the thermal paste (250 [°C]), and measure the time it takes to reach the target temperature. Simulations were carried out using COMSOL Multiphysics 5.5 and the Finite Element Method (FEM), with a time-dependent thermal analysis in a 3D domain. The physical model is the build platform previously analyzed and designed, with its three main components.

The thermal boundaries assumed initial and ambient temperatures of 20 [°C], heat transfer via conduction, and thermal insulation at specific surfaces. The thermal paste interface between the cartridge and the platform was modeled as a resistive thin layer with a conductivity of 9 [W/m^2K], with a critical temperature limit of 250 [°C]. Radiation was considered for heat loss, while convection was considered to be zero, as the platform stands still around air during the heating process.

To evaluate performance, two points were monitored: the thermocouple location and the center of the heating cartridge. The system uses a basic ON/OFF relay-controlled loop: the heater delivers 50 [W] if the measured temperature is below the target, and shuts off otherwise. In the first simulation iteration, the bed successfully reached setpoints of 50, 60, and 70[°C], which are common build platform temperatures for proper material adhesion during the first layer deposition. However, a notable overshoot was observed at the thermocouple location, and the cartridge temperature significantly exceeded 250[°C], breaching the paste's safe operating range.

To solve this, three alternatives were evaluated: using energy-free intervals to exploit thermal inertia (rejected due to inconsistency with environmental and initial conditions), modeling the cartridge temperature via interpolated equations (rejected for complexity), and implementing a timed relay activation strategy (selected). This method alternates fixed durations of power ON/OFF, allowing for controlled temperature rise while limiting the cartridge temperature.

A second iteration of simulations tested several on/off intervals. The most effective was a 10-second on / 10-second off cycle. This configuration reduced the cartridge temperature to a safe maximum of 188 [°C] and achieved the target surface temperature of 70 [°C] in 235 seconds—above the 3-minute calibration requirement, but acceptable. Compared to the original uncontrolled system, this approach increased the heat-up time by 80% but preserved system integrity and safety.

Consequently, the study concludes that the temperature control loop should be modified to include fixed time-based activation of the relay. The final governing equation integrates both temperature feedback and time modulation, ensuring safe and effective regulation of the heated bed:

$$P(T,t) = \begin{cases} 50 \text{ [W]} & \text{if } T(t) < T_{command} \text{ and } t \bmod 20 < 10, \\ 0 \text{ [W]} & \text{otherwise.} \end{cases} \tag{6.2}$$

The implementation of this control loop onto the ESP microcontroller was carried out using the Arduino IDE (version 1.8.19), programmed in C/C++. Although the development board used is a NodeMCU (not an original Arduino), it is compatible with the Arduino IDE.

The structure of the code follows the typical Arduino architecture: library inclusion, global variable definitions, the `setup()` function (executed once at startup), and the `loop()` function, which runs continuously. The program was validated using a heating element similar to the one used in the actual system, verifying compliance with key requirements, including temperature range, communication protocol, power control, and message feedback.

To control power delivery, a potentiometer was used, and the relationship between its rotation angle and output power was modeled using an empirical quadratic equation:

$$P(\theta) = \frac{12.34 + 16.39 \cdot \theta + 5.2 \cdot \theta^2}{1000}\,\mathrm{W} \tag{6.3}$$

This relationship was derived using measurements obtained with a Yokogawa WT330 oscilloscope, with support from PhD student Nikola Mirković at CEI.

The core of the program is encapsulated in a custom library named `thermoregulator`. The purpose of the library is to simplify the main code by isolating key tasks: controlling the power supply to the heater via a relay, and generating status messages. The library is composed of a header file (declaring public/private methods and attributes) and a source file (defining the logic). It relies on additional libraries such as `Adafruit_MAX31855` (for thermocouple communication), `Ticker` (to implement timing functionality), and the native `SPI` and `Arduino.h`.

The class constructor initializes hardware pins for the MAX31855 thermocouple module and the relay pin, as well as the sampling frequency. A public attribute allows the user to specify the target temperature at any point during execution. The key method `controlTemperature(boolean activated)` governs the behavior of the relay based on the process state and current temperature. It relies on two private helper functions: one to update and average the latest ten temperature readings, and another to handle state transitions.

To manage power cycling, the `Ticker` library is used to implement the 10-second on/off cycle, with the variable `periodicValue` toggling each cycle. The control logic follows a modified ON/OFF strategy defined previously. When activated, the heater is powered if the latest temperature reading is below the target and within the "ON" interval. If the user deactivates the process, the relay is turned off and the timing logic is halted.

The library also implements a messaging system to communicate the state of the regulator back to the CPU. These include real-time data such as the last measured temperature, the average of the last ten readings, the target temperature, and a numerical status code. This status message is categorized as follows:

- 0: Heating – regulator is active, temperature is below the lower threshold.
- 1: Heated – regulator is active, temperature is within acceptable limits.
- 2: Overheated – temperature exceeds upper threshold.
- 3: Cooling – regulator is off, but bed is still hot.

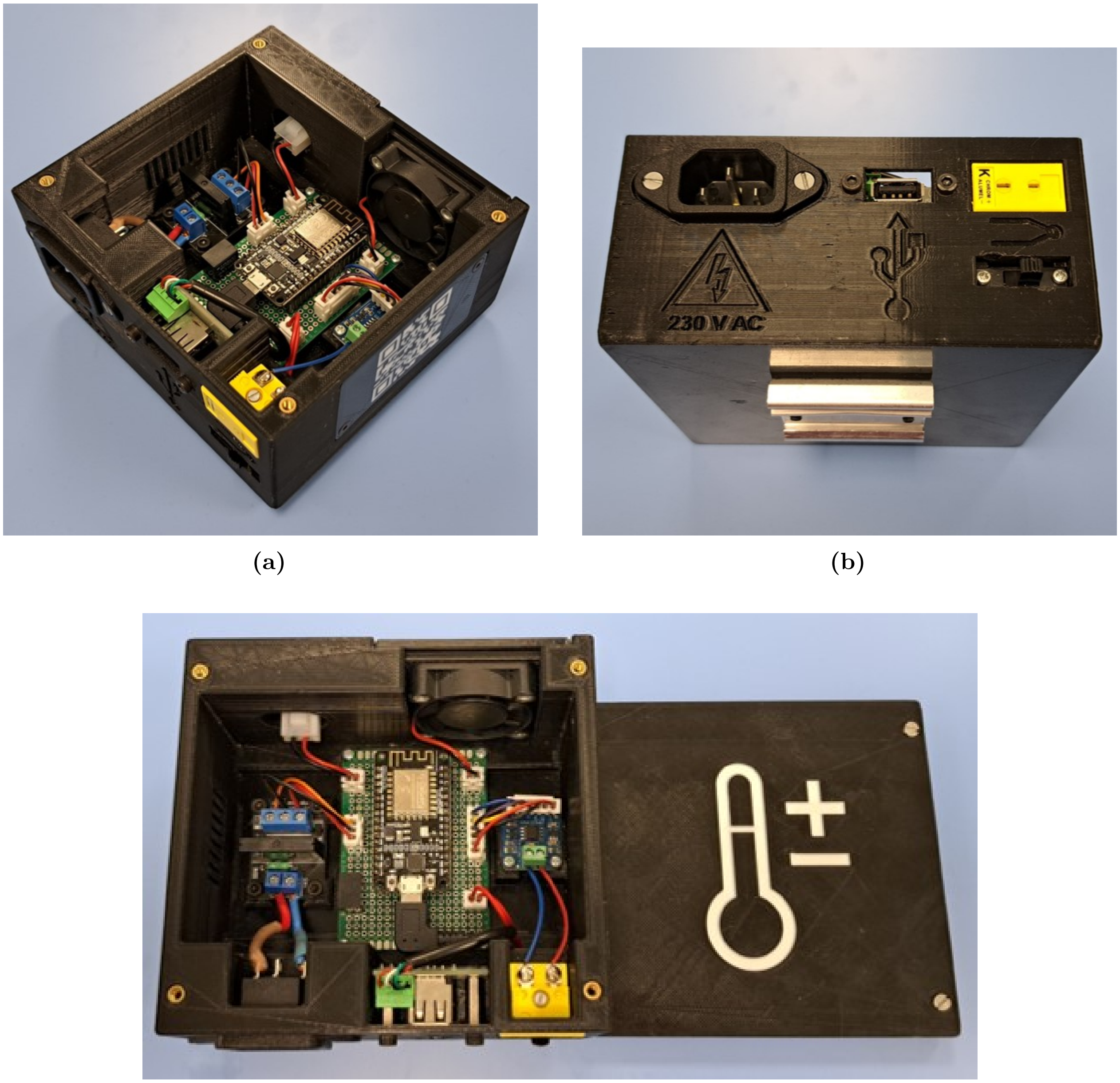
- 4: Cooled – regulator is off and temperature has returned to ambient.

At startup, the function `initializeSystem()` is used to pre-fill the temperature array with current values to ensure system stability. The main program file (`.ino`) defines pin mappings and includes the `thermoregulator` and `pinout` headers. In `setup()`, pins are initialized, serial communication is started at 2400 baud, and initial temperature readings are taken.

The system is designed to interact with commands sent over the serial port by the CPU. This communication is handled through the Arduino `serialEvent()` function, which triggers automatically when data is received. The input is checked: if it is `OFF`, the regulator stops. If the message is a valid number within the operating temperature range (50–70 [°C]), it is set as the new target, and the regulator activates.

On the CPU side, the serial communication is managed by a ROS2-based architecture developed by Miguel Lerín, Daniel Galán, and Javier Badesa, from CAR-CSIC-UPM, using the publisher-subscriber model. A dedicated node (`serial_reader`) within the `build_platform` package sends commands and receives feedback from the controller, allowing real-time monitoring and logging. Although this setup uses basic serial communication, it provides a robust framework for future upgrades to more advanced protocols such as TCP/IP. This programming architecture successfully implements the control logic required to regulate the bed temperature safely and effectively, ensuring compatibility with both hardware and higher-level system integration.

The fully assembled controller is provided in Figure 6.23, and is to be included in the Power Supply unit described further in the present Chapter.



(a)

(b)

(c)

Figure 6.23: Build platform temperature controller. (a) Isometric view of the interior. (b) Power and signal terminals. (c) Complete product.

6.3.2 Extruder control

This Subsection documents the systematic design of the extruder-controller enclosure. Again, a systematic product development methodology was utilized, providing a conceptual design, basic and detailed engineering, and lastly, building a prototype. Most of the concepts utilized for this element are drawn from the experience of the build platform driver prototype, and therefore, this Subsection is presented in an abridged fashion.

Regarding conceptual design, a list of mandatory requirements and optional aspirations was created. Mandatory constraints include limiting the enclosure to $120 \times 100\,\mathrm{mm}$ in plan and 200 mm in height, unobstructed external connectivity, orderly internal wiring, external power access, rapid circuit extraction, modular interchangeability, and maximal spatial efficiency. Optional aspirations comprise visual identification via a dedicated symbol and easy access to documentation through an external QR-code plate.

The enclosure is decomposed into fourteen Functional Groups (FGs), each addressing selected requirements and aspirations. Key elements include dual solid-state relays to implement the thermistor and operate the heater, a TB6600 driver for stepper actuation and control, segregated 3.3 [V] and 24 [V] perforated boards, guiding rails, auxiliary adapters, end-stops, and connector interfaces to enhance mechanical robustness and usability. Additionally, DIN-rail integration ensures easy cabinet installation. Spatial constraints are mitigated through 35 [deg] inclined Molex and USB adapters.

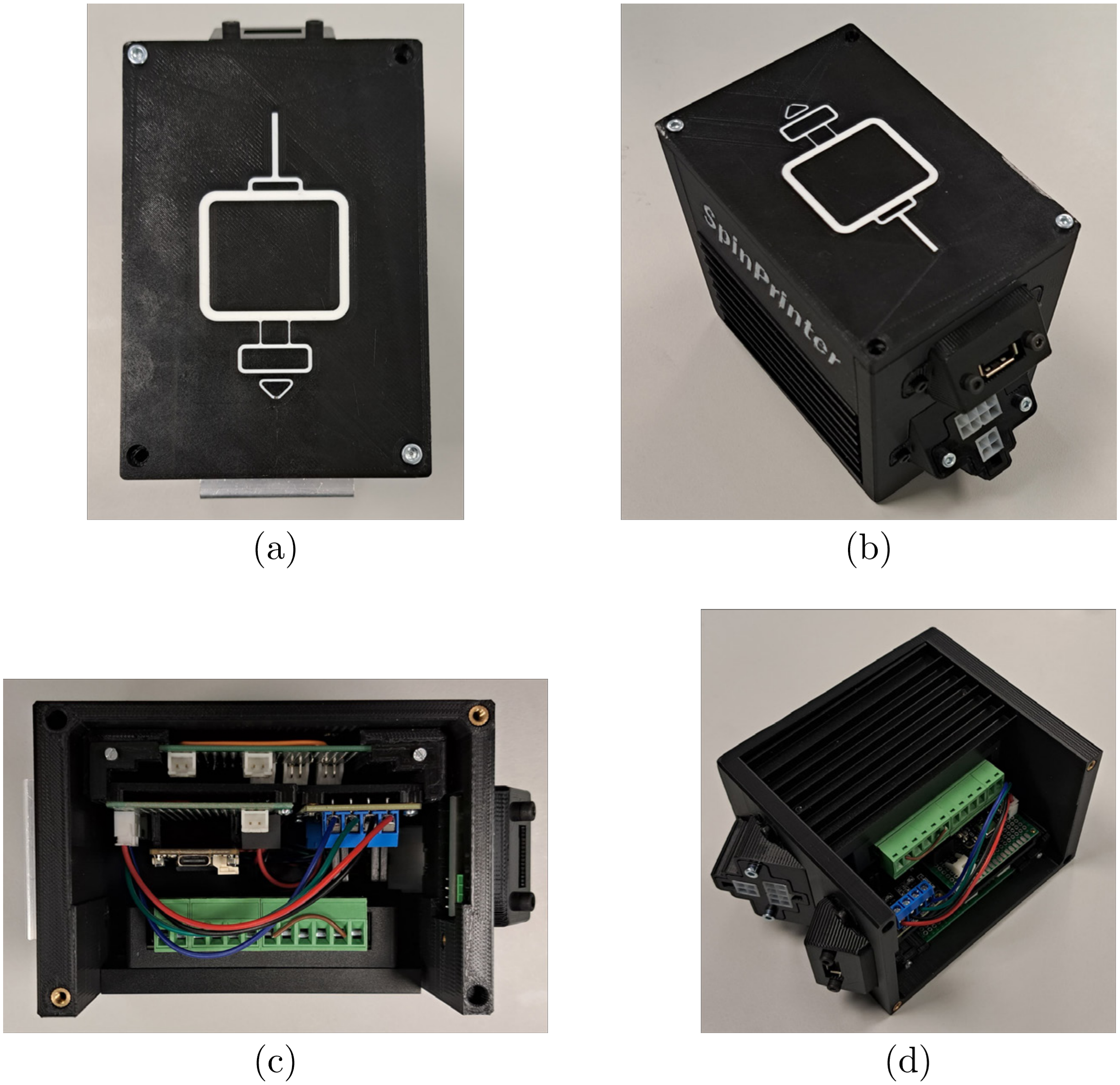
Three–dimensional models specify manufacturing tolerances: 0.5 [mm] clearances to the nearest wall, 0.3 [mm] insert-hole tolerances, and 0.2 [mm] recesses for connector faces. The enclosure base integrates all FGs: a sliding component plate traverses rails spaced 5 [mm] apart, while stops prevent accidental disengagement. Final CAD models also include a QR-code recess, and are validated via clash analysis. Additive manufacturing of all parts requires about thirty-eight hours of print time, exclusive of reprints required by design or print defects.

Regarding the controller implementation on the ESP32, the firmware integrates temperature regulation, stepper control, and serial I/O. Local headers hold pin maps and constants; external libraries `PID_v1` and `AccelStepper` provide closed-loop heating and non-blocking motion. `setup()` configures GPIO, launches the serial port at 115 [$\mathrm{kbit{\cdot}s^{-1}}$, enables a PID controller ($K_p = 40$, $K_i = 3$, $K_d = 500$), and sets safe limits for speed and acceleration. An additional function, `readT` obtains the temperature value from the thermistor. To enhance the thermistor's resolution on both low temperature ranges (below 100 [°C]) and high temperature ranges (above 85 [°C]), a commutable resistance circuit was implemented by means of the previously mentioned relay. `readT` also manages this relay to obtain a temperature resolution no worse than 2 [°C].

Every `cycleTime` [ms] the loop converts ADC data to temperature, updates the PID, and applies two heuristics: full power below 130 [°C] and cutoff when 2 [°C] above setpoint. Within a 5 [°C] band beneath target it transmits "TEMP_OK" through the serial port to the main PC and, if allowed, drives the stepper at the commanded speed through `runSpeed()`.

`serialEvent()` parses three commands: `OFF` shuts down heater and motor, `EXT_OK` enables extrusion once hot, and T, ω sets a new temperature and speed while arming the heater. Invalid inputs return “WRONG,” ensuring protocol discipline. By pairing a PID loop with non-blocking stepper control and a simple command grammar, the firmware delivers stable melt-zone temperature and filament feed on simple hardware.

Figure 6.24 showcases the final result.



(a) (b)

(c) (d)

Figure 6.24: Extruder driver integration. (a) Top view with lid. (b) Isometric view with lid. (c) Interior, top view. (d) Interior, isometric view.

6.3.3 Power supply

The power supply element consists of a power panel containing all circuit breakers, fuses, power distribution, voltage transformers, power supplies, and controllers, such as the build platform temperature controller and the extruder controller. The power supply unit also serves the purpose of neatly organizing the wiring. A simplified electric diagram is portrayed in Figure 6.25 to visually explain the connections, as well as the power demands and other types of wiring needed for the station.

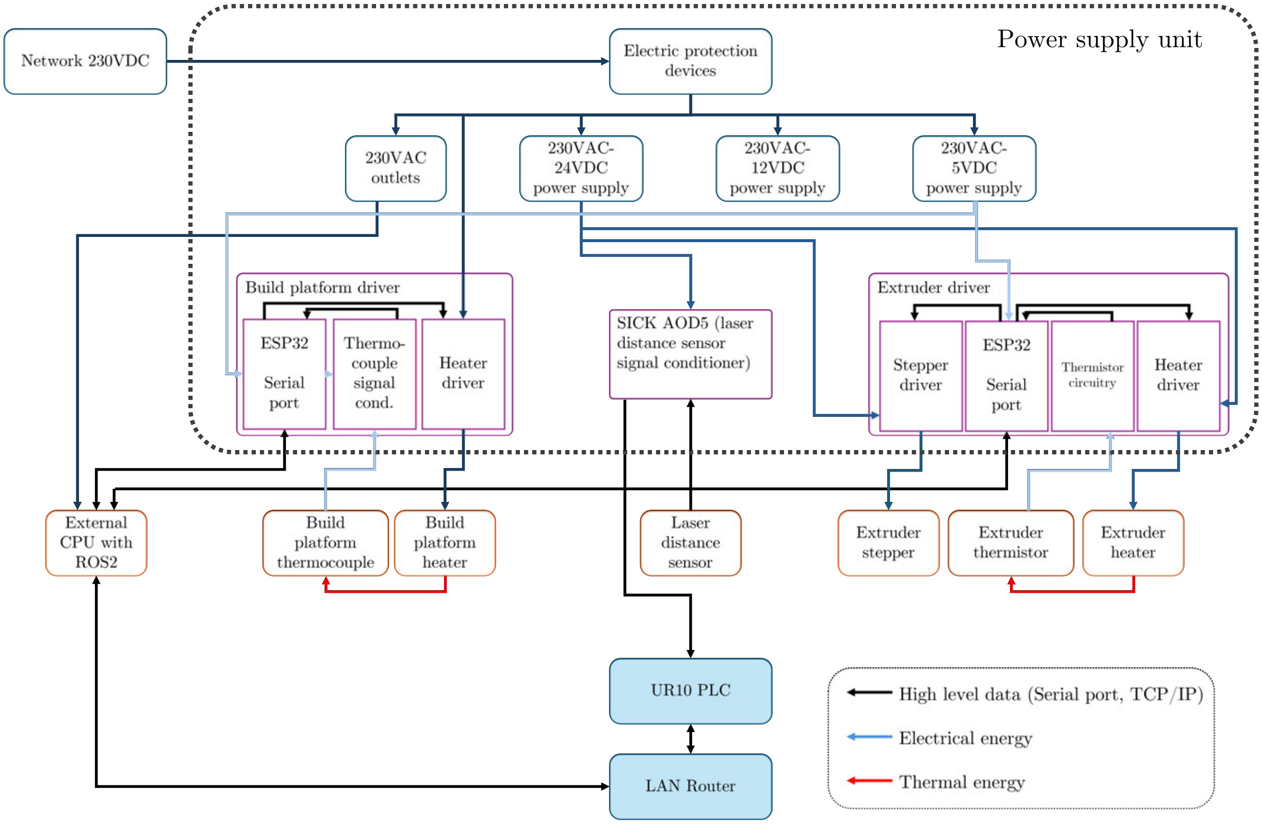


Figure 6.25: Simplified electric diagram summarizing the power supply elements and wiring. Electrical elements are represented in blue boxes, drivers in purple, actuators in light orange, sensors in dark orange.

The components selected for the power supply unit are slightly overdimensioned to allow for expansions of functionality, such as multiple extruders, build platforms, or other devices to be implemented in the future. As such, the elements of this power supply unit consist of:

- Circuit breakers and fuses to ensure operation safety.
- A panel ventilation fan on the side of the box.
- Voltage transformer (220 VAC – 24 VDC) for the extruders, the laser signal conditioner, and the panel ventilation unit.
- Voltage transformer (220 VAC – 12 VDC) for controller ventilation units.
- Voltage transformer (220 VAC – 5 VDC) for controller power supply.

- Two Schuko plugs to connect CPUs.
- Free slots on the DIM rail to allow for controller integration.

The overall result of the power supply unit is shown in Figure 6.26.

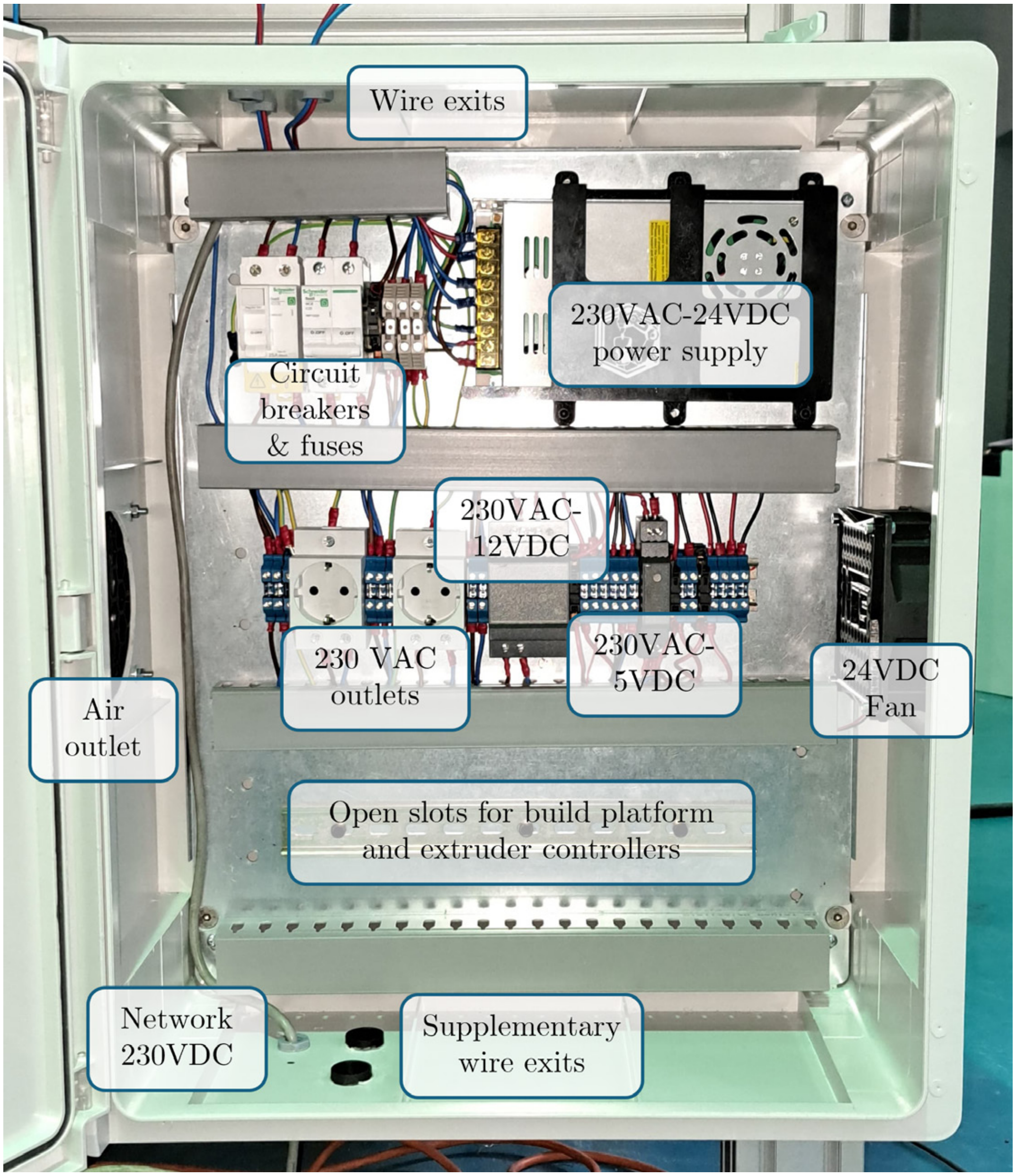


Figure 6.26: Power supply unit.

6.3.4 Calibration and extruder fixture

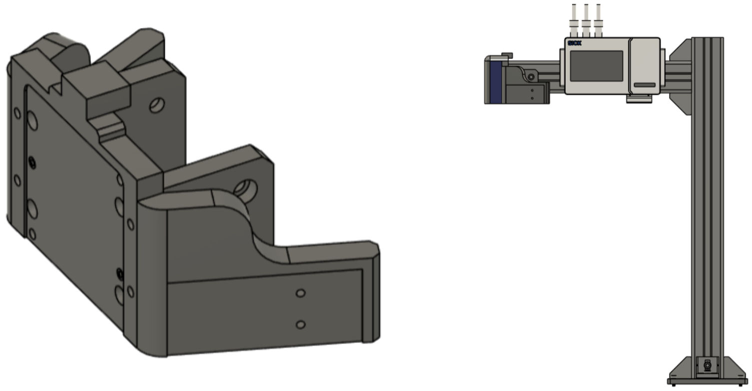
This Subsection summarizes the design process for the calibration laser and extruder fixture. As in the previous Subsections, the design is organized into functional groups, with technical solutions proposed, evaluated, and selected according to a structured set of acceptance criteria. The goal is to develop a modular, adaptable, and safe system for this fixture. The FGs proposed are the following:

- **FG1:** External Support – Provides a rigid, elevated structure for the extruder, enabling freedom of movement for the robotic arm.
- **FG2:** Extrusion Assembly – Manages filament deposition, ensuring compatibility with control software and operational robustness.
- **FG3:** Calibration System – Allows the calibration procedure outlined in the next Chapter, positioning with accuracy the extruder and the build platform.
- **FG4:** Extruder Head Mount – Connects extruder and sensors to the support frame, allowing secure mounting and modularity.
- **FG5:** Orientation Between Components – Ensures correct relative positioning between system components for extrusion accuracy.
- **FG6:** Protections – Protects sensitive elements from heat, collisions, and external errors.

An FS was proposed for each of these FGs:

- **FS1:** External Support – The inverted-L aluminum profile was selected due to its compact design, structural simplicity, and compatibility with multi-extrusion systems. It allows greater mobility for the robotic arm compared to arch or I-shaped structures.
- **FS2:** Extrusion Assembly – The independent extruder per material was chosen for the ability to handle different materials simultaneously without the need for material-switching mechanisms, reducing downtime and complexity.
- **FS3:** Calibration System – The laser calibration system emerged as the best option owing to its measurement accuracy, adaptability to various configurations, and superior performance compared to LiDAR or manual methods.
- **FS4:** Extruder Head Mounting – The multi-faceted “arrow” head was selected for its modular design, allowing several components such as extruders, sensors, and lasers to be mounted independently.
- **FS5:** Orientation Between Components – The angular configuration provides flexible spatial arrangement of components, facilitating robot arm paths and improving overall station efficiency.
- **FS6:** Protections – Multiple solutions were selected to cover different protection needs: a silicone sleeve for hot-end insulation, additively manufactured cable covers for routing and shielding wires, and a dedicated enclosure for the calibration system.

With these FSs, a CAD model of the fixture was created. The design is shown in Figure 6.27.



(a) (b)

Figure 6.27: CAD drawings of the extruder fixture and the overall structure.

The location of this fixture is selected according to the uncertainty maps developed in the previous Section. Due to the existence of previous parts for other projects, the selected location was location B. The final result is presented in Figure 6.28. There is a key difference between the design and the final result, and that is the location of the extruder. As a temporary solution, the extruder was located in a lower fixture to provide more movement freedom, especially for concave build platforms. In a future iteration, the additively manufactured parts, as well as the temporary extruder fixture are to be integrated according to designed.

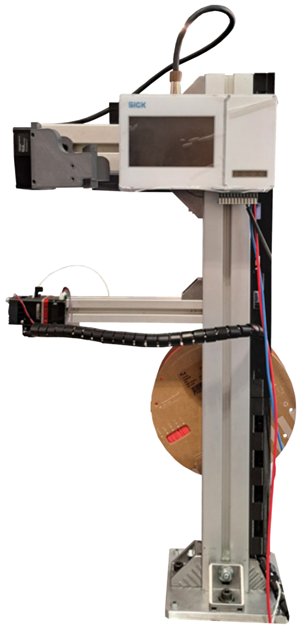


Figure 6.28: Detail of the extrusion and calibration fixture.

6.3.5 Overall result

In Figures 6.29 to 6.31 the final robot-based NPAM cell is presented. As a remark, the build platform and the extruder controllers were taken out from the power supply unit to better monitor the tests performed in the next Chapter. Some wire management parts were additively manufactured to improve the organization and tidiness of the station. The manufacturing cell is fully operable and its performance will be showcased in Chapter 7 through the realization of several case studies.

Figure 6.29: Robot-based NPAM-MEX-P manufacturing cell developed in this Thesis (front view).

Figure 6.30: Robot-based NPAM-MEX-P manufacturing cell developed in this Thesis (isometric view).

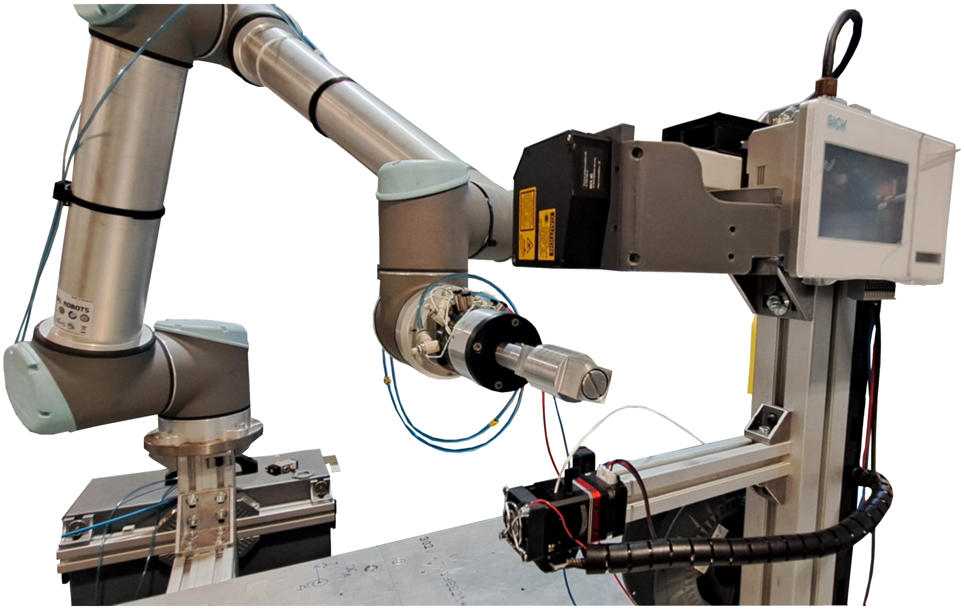


Figure 6.31: Detail of the build platform approaching the calibration and extrusion fixture.

Chapter 7

Manufacturing workflow: how to realize all previous concepts on actual manufacturing equipment

This Chapter aims to have all previous concepts converge onto the manufacturing of robot-based NPAM-MEX-P parts. To do so, it is necessary to establish a workflow that encompasses manufacturing equipment set-up, process set-up, as well as operation proper.

This Chapter will be divided in four Sections. Firstly, the overall workflow will be postulated, to properly define all necessary procedures for the process to work. Within this workflow, calibration procedures are the most critical, as the quality of these calibrations largely determines the success of the process. As such, two Sections define both the calibration of extrinsic reference frames (i.e., independent of the kinematic architecture) and intrinsic reference frames (i.e., dependent on the kinematic architecture). These processes are required to determine the locations of the extrusion apparatus and the build platform, outlined in the previous Chapter. Lastly, several examples of application of the workflow are shown in the manufacture of robot-based NPAM MEX-P parts, combining all the concepts postulated in this Thesis.

7.1 Overall robot-based non-planar additive manufacturing MEX-P workflow

The first step in realizing the methods presented in this Thesis is providing a general workflow. This workflow is arranged in the classical manufacturing steps, which include:

- Manufacturing equipment design, modeling, and implementation.
- Part design.
- Tooling design and implementation.
- Offline set-up, which usually consists of trajectory definition, as well as tool and materials

selection (if needed).

- Online set-up, which often comprises machine calibration and equipment warm-up.
- Operation, where the manufacturing process proper occurs.

Some of these steps can be done non-linearly. This asynchronization of tasks was the objective of Chapter 4. Additionally, many of the design decisions taken in 6 were taken with this workflow in mind, to facilitate and parallelize calibration and extruder heating. The overall workflow is presented in Figure 7.1. The arrows of the diagram symbolize the dependency of some tasks from others: as an example, it is not possible to intrinsically calibrate the build platform if the laser itself is not extrinsically calibrated.

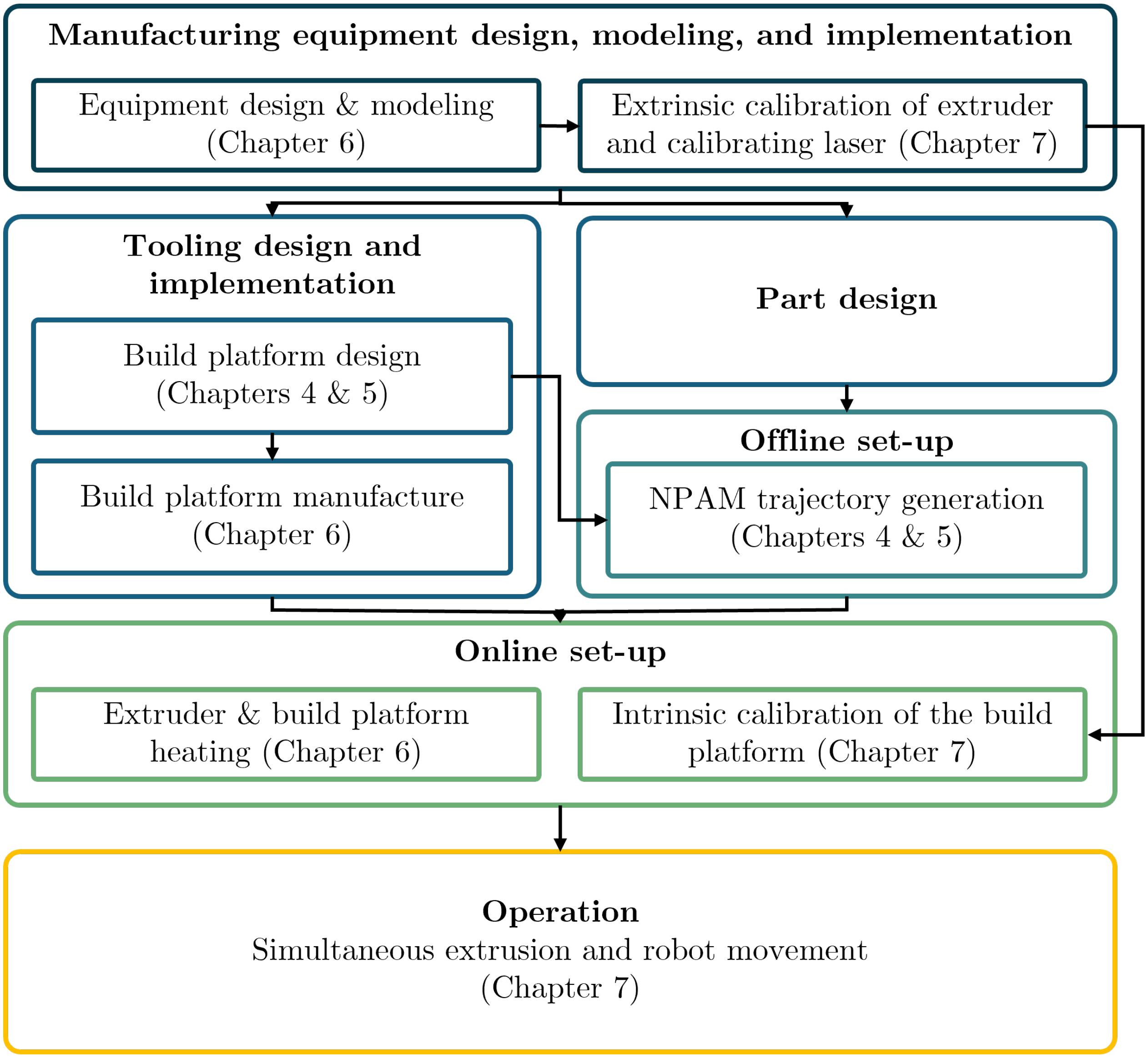


Figure 7.1: Manufacturing workflow postulated in this Thesis for the robot-based Non Planar Additive Manufacturing of MEX-P parts.

The online set-up procedure is divided in two parts, the extruder and build platform heating and the intrinsic calibration of the build platform, which are performed simultaneously. To achieve this, a ROS2-based architecture has been implemented by members of CAR-CSIC-UPM. As described in Chapter 6, a CPU serves as the system's "mastermind", which is responsible of coordinating all tasks during the online set-up and the operation.

Owing to the controllers' capability of bidirectional communication via serial ports, the mastermind sends messages to the controllers, indicating temperatures for both elements and extrusion rate. Once the messages are received by the controllers, the elements start heating up to the command temperatures. When the command temperatures are achieved, the controllers communicate a `READY` message to the mastermind. This is the end this phase, and when the calibration procedure is done, the operation proper begins, having the extruder apply the extrusion rate as commanded by the mastermind.

Regarding the calibration procedure, there are many reference frames to account for in the developed robot-based NPAM MEX-P station. These reference frames are presented in full in Figure 7.2. For visual clarity, intrinsic frames are depicted in red, whereas extrinsic frames are presented in blue. All these reference frames are coordinated through a trinomial of constant frames, which are the $\{base\ link\}$ (for the intrinsic chain), the $\{base\}$ (for the extrinsic chain), and the global reference frame ($\{grf\}$ in short), which are one and the same.

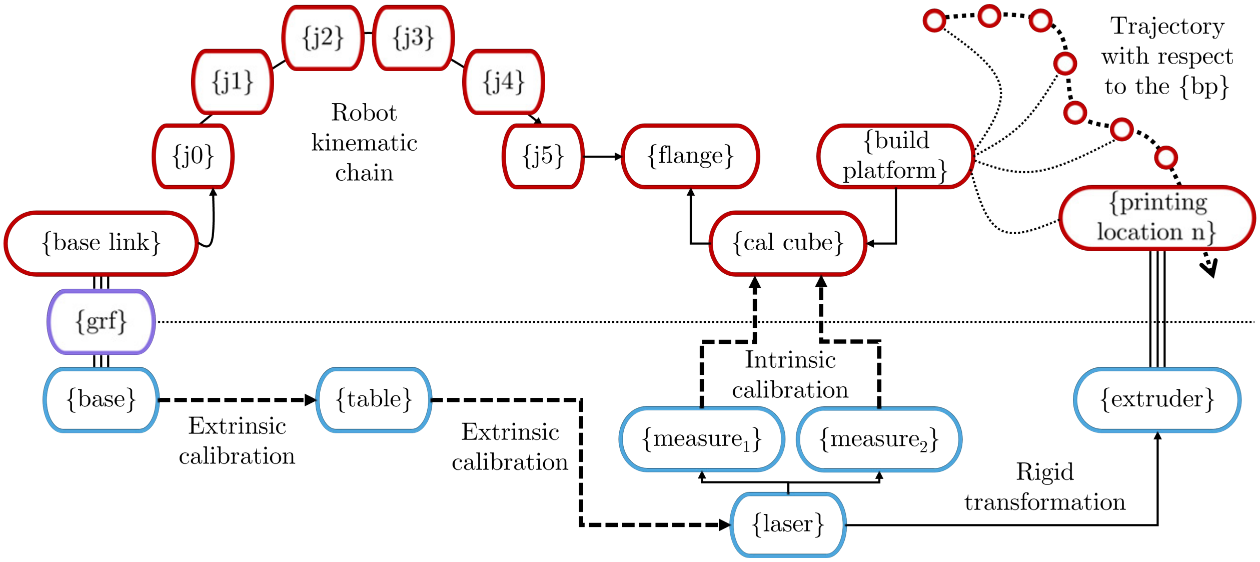


Figure 7.2: Manufacturing workflow postulated in this Thesis for the robot-based Non Planar Additive Manufacturing of MEX-P parts.

The previous Figure also serves as a way to establish the trajectory in the station workflow. The trajectories, obtained from the methods presented in Chapters 4 and 5, are obtained in the $\{build\ platform\}$ ($\{bp\}$ in short). To obtain the proper robot trajectory, it is necessary to locate these reference frames with respect to the flange. This is done in two steps. Firstly, the trajectory is located with respect to the calibration cube reference frame $\{calcube\}$ through a rigid transformation. This rigid transformation is provided through the build platform CAD model. Subsequently, this $\{calcube\}$reference frame is intrinsically calibrated, that is, its location and orientation with respect to the $\{flange\}$ reference frame are calculated during the online set-up.

Additionally, to ensure the extrusion success, both the $\{table\}$ and $\{laser\}$ reference frames are extrinsically calibrated. These procedures can be done as soon as the calibration fixture is implemented in the manufacturing equipment, and is done by attaching a laser distance sensor to the robot flange. Both these procedures are outlined in the following Subsections.

7.2 Calibration of extrinsic reference frames

In this Thesis, the main reference frame is the space frame $\{s\}$. In conventional manufacturing literature, especially Computer Numerical Control (CNC) texts, this frame is often called Machine Coordinate System ($\{MCS\}$), or otherwise, global reference frame ($\{grf\}$). Starting from this space frame $\{s\}$, several other working frames are defined. As shown previously in Figure 7.2, $\{table\}$ is a frame located in the worktable corner, $\{laser\}$ is a frame located in the laser fixture, and $\{measure\}$ represents the frame where laser distance measurements take place. This is dependent on the laser model utilized, having $\{measure_1\}$ and $\{measure_2\}$ accounting for the different sensor ranges. Lastly, $\{extruder\}$ is the frame where the MEX-P process occurs, and coincides with the nozzle's end. The next step is determining these frames with accuracy with respect to $\{grf\}$, which requires calibration.

Calibration can be understood as the correction of a certain CAD-based reference frame which accommodates for errors and tolerances in the assembly. This philosophy is ubiquitous in classic CNC workflows, and is best represented in the procedure of obtaining the Workpiece Coordinate System $\{WCS\}$. Working with the Machine Coordinate System $\{MCS\}$ is often unwieldy, as these coordinates do not reflect easy to use distances, especially in more than 3 axes contexts. Therefore, the $\{WCS\}$ is defined locally to the stock material to facilitate machine movements.

This procedure is often done manually by the operator, assisted by a measuring tool. This $\{WCS\}$ definition can be separated in two components, a transformation between the $\{MCS\}$ and the theoretical coordinates of the stock material, plus a $\Delta\mathbf{T}$ which accounts for the actual assembly of the fixture and the mounting of the stock on the fixture. Usually, the operator directly obtains the result of the sum in a procedure known as "obtaining the workpiece zero" (or workpiece calibration), but for the purposes of this Thesis, it is better to keep both terms separated. After the $\{WCS\}$ has been obtained, the rest of the CNC program is performed according to that reference frame. This is the main underlying philosophy for the methods presented below, and the station workflow as a whole.

Kinematics of extrinsic chains are more approachable than those of intrinsic chains, which are tied to the robot's configuration. For the purposes and methods of this Thesis, the kinematics of extrinsic chains will consist of the definition of several working frames outside of the kinematic architecture of the manipulator, whose mission is to facilitate the user's work.

An extrinsic calibration procedure is necessary to determine these frames. This Section aims to describe the extrinsic reference frame calibration procedure performed for both the $\{table\}$ and the $\{laser\}$ reference frames. Firstly, a description of the method is provided.

Subsequently, an experimental validation is described with results and discussion to further substantiate the suitability of the method. This Section has been adapted from a previous work of the author [128]. This workflow inherently defines the transformations between extrinsic reference frames. The initial reference frame is the robot $\{base\}$, as outlined in the previous Section, a Cartesian system associated with the robot's mounting plane, located at the center of its footprint. This reference frame is independent of the robot's position relative to other station components and serves as the base for defining all subsequent systems. This workflow is defined in a generalizable manner, accommodating varying configurations or the absence of certain station components. For example, if the station lacks a defined worktable, the transformation diagram simplifies accordingly.

With these concepts in hand, the kinematics between these systems are expressed by transformation matrices, whose elements are determined by a combination of CAD models and extrinsic calibration operations. For instance, the $\{laser\}$ frame is defined as:

$$\{laser\} = \{table\}\mathbf{T}_{\{table\}\to\{laser\}} \tag{7.1}$$

$$\begin{aligned}\mathbf{T}_{\{table\}\to\{laser\}} &= (\mathbf{T}_{\{table\}\to\{calibration\ totem\}_{CAD}} + \\ &+ \Delta\mathbf{T}_{\{table\}\to\{calibration\ totem\}})\mathbf{T}_{\{calibration\ totem\}\to\{laser\}_{CAD}}\end{aligned} \tag{7.2}$$

In this equation, $\mathbf{T}_{\{table\}\to\{laser\}}$ is the overall transformation, which in turn may be decomposed as the CAD model transformations $\mathbf{T}_{\{table\}\to\{calibration\ totem\}_{CAD}}$ and $\mathbf{T}_{\{calibration\ totem\}\to\{laser\}_{CAD}}$, and the extrinsic calibration $\Delta\mathbf{T}_{\{table\}\to\{calibration\ totem\}}$. This $\{calibration\ totem\}$ frame was briefly shown in Chapter 6, in the shape of a small cube on top of the fixture, ridigly joint to the laser+extruder part. These cubic shapes, called calibration totems, also present in the build platform, are a geometric pattern utilized for calibration (both extrinsic and intrinsic). Of note, to minimize the error of the $\mathbf{T}_{\{calibration\ totem\}\to\{laser\}_{CAD}}$ transformation, the calibration totems have been either monolithically designed (e.g., the calibration totems on the build platforms), or joint with strict tolerance requirements in assemblies (e.g., the calibration totem on the extruder fixture).

The problem at hand becomes the determination of $\Delta\mathbf{T}_{\{i\}\to\{j\}}$ throughout the extrinsic elements of the station, since the others are readily provided by the CAD model. This is the crux of this Section.

The method postulated is as follows. The calibration is calculated in three steps: given two frames (for instance, $\{base\}$ and $\{table\}$), the calibration of $\{table\}$ with respect to $\{base\}$ is calculated through a rotation of the xy-plane, followed by a rotation about the new z-axis, followed by the translation of the resulting frame. This concept is visually represented in Figure 7.3.

Given that the calibration totem has a square surface with edges, it is possible to obtain seven characteristic points: three on the surface, four on the edges. Given the three points in the surface (p_1, p_2, p_3), three distance measurements of said points are taken with respect to a distance laser mounted on the end-effector (d_1, d_2, d_3), as depicted in Figure 7.4. The surface plane can be defined through its equation:

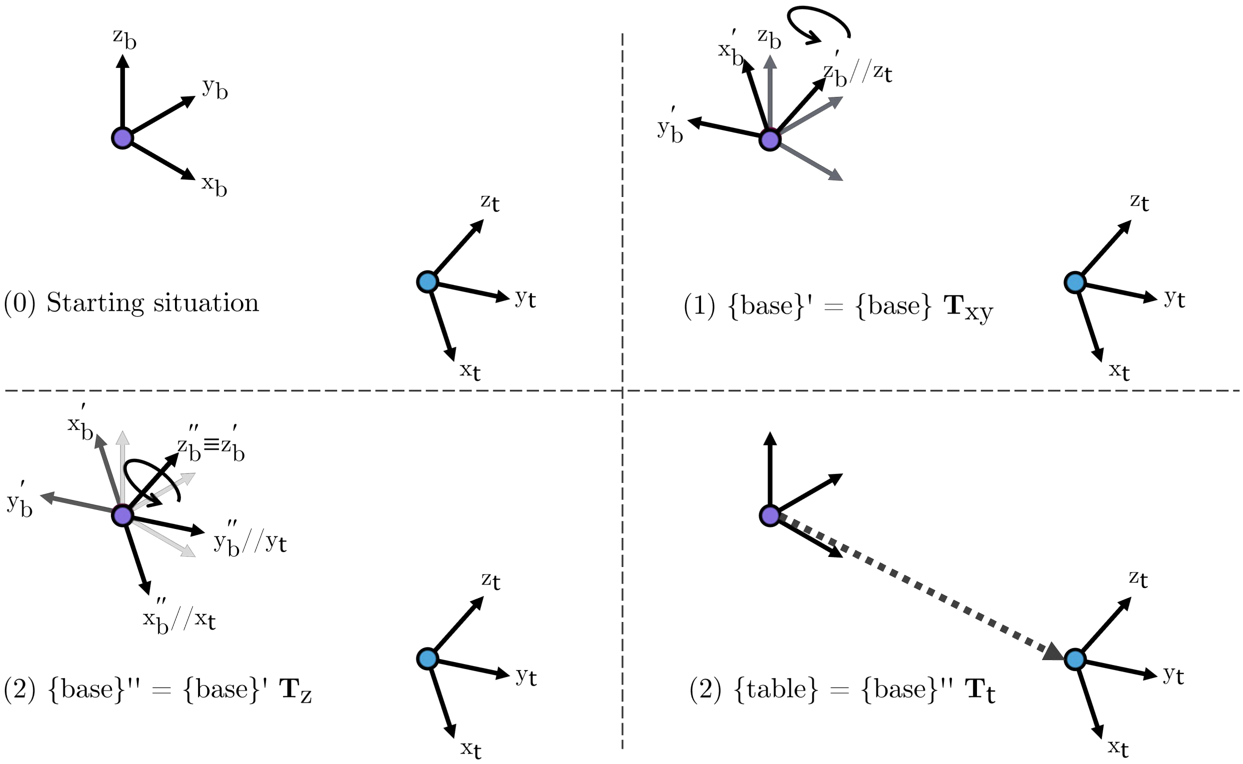


Figure 7.3: Visual representation of the three steps performed in extrinsic calibration.

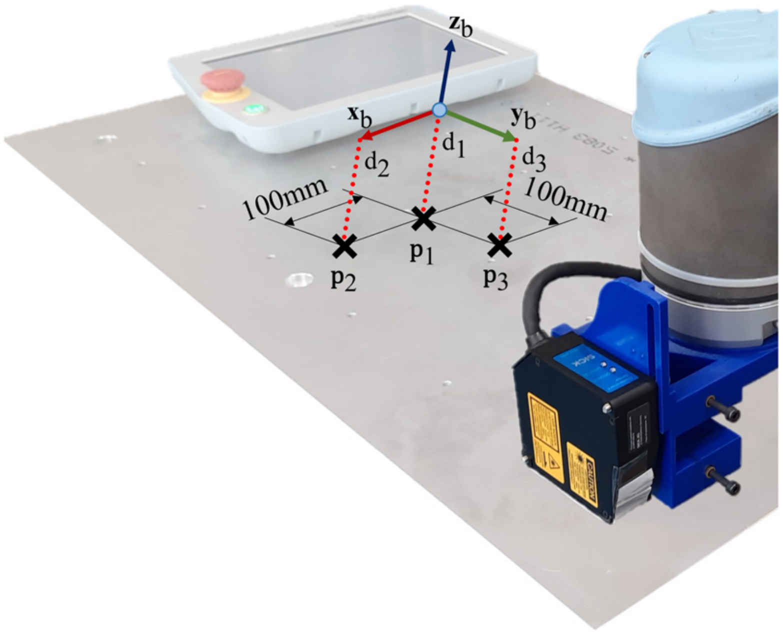


Figure 7.4: Definition of points for the first step of the extrinsic calibration.

$$z = \frac{p_2^y(d_3 - d_1) - p_3^y(d_2 - d_1)}{p_2^x p_3^y - p_3^x p_2^y} x + \frac{p_3^x(d_2 - d_1) - p_2^x(d_3 - d_1)}{p_2^x p_3^y - p_3^x p_2^y} y \tag{7.3}$$

To obtain the distance values d_i, a conversion was implemented between distance and the sensor analog signal. The signal conditioner AOD5 (compatible with the SICK sensors employed in this Thesis OD5-30W05 and OD5-350W100), allows for voltage or current analog outputs. For convenience, the current analog output was preferred, as it ranges between 4 [mA] and 24 [mA]. This current range is fully compatible with the UR10 PLC's current analog input range. The conversion obeys the graphs presented in Figure 7.5.

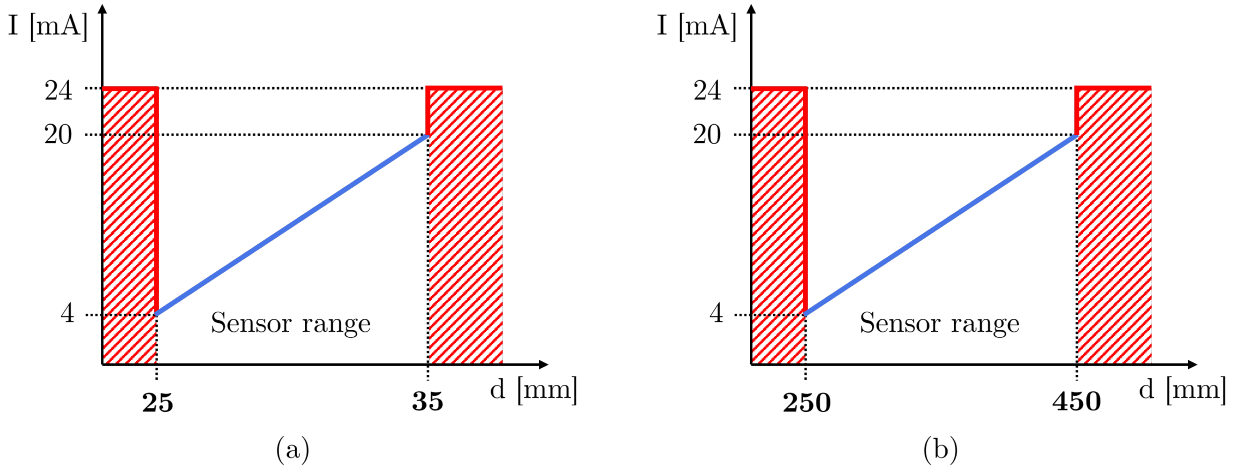


Figure 7.5: Current conversion according to SICK distance sensor datasheets. (a) Conversion for sensor OD5-30W05. (b) Conversion for sensor OD5-350W100.

The plane z defined in (7.2) has a natural orientation $\mathbf{R}_{xy} \in SO(3)$, related to a homogeneous transformation $\mathbf{T}_{xy} \in SE(3)$. These $\mathbf{R}_{xy}$ and $\mathbf{T}_{xy}$ can be calculated through the following equations:

$$\nabla z(x, y) = \left\{ \frac{\partial z}{\partial x}, \ \frac{\partial z}{\partial y} \right\} \tag{7.4}$$

$$\mathbf{u}_g = \frac{\{\nabla z^x, \ \nabla z^y, \ 0\}}{\|\nabla z\|} \tag{7.5}$$

$$\mathbf{m} = \{\nabla z^x,\ \nabla z^y,\ z(\nabla z^x, \nabla z^y)\}; \mathbf{u}_m = \frac{\mathbf{m}}{\|\mathbf{m}\|} \tag{7.6}$$

$$\mathbf{n} = \mathbf{p}_2 \times \mathbf{p}_3; \mathbf{u}_n = \frac{\mathbf{n}}{\|\mathbf{n}\|} \tag{7.7}$$

$$\mathbf{u}_{\mathbf{R}_{xy}} = \frac{\mathbf{u}_m \times \mathbf{u}_n}{\|\mathbf{u}_m \times \mathbf{u}_n\|} \tag{7.8}$$

$$\alpha_1 = \arccos(\mathbf{u}_g \cdot \mathbf{u}_m) \tag{7.9}$$

$$\mathcal{R}_{xy} = \begin{pmatrix} \mathbf{u}_{\mathbf{R}_{xy}}^{3\times 1} \\ \mathbf{0}^{3\times 1} \end{pmatrix} \tag{7.10}$$

$$\mathbf{T}_{xy} = e^{[\mathcal{R}_{xy}]\alpha_1} = \begin{pmatrix} \mathbf{R}_{xy} & \mathbf{0}^{3\times 1} \\ \mathbf{0}^{1\times 3} & 1 \end{pmatrix} \tag{7.11}$$

In the example previously described, the reference frame $\{table\}$ origin is arbitrarily defined at one corner of the table, with x- and y-axes aligned with table edges in opposite directions and the z-axis orthogonal to both. This definition, chosen for operational simplicity, is illustrated in Figure 7.6. These axes are similar to the $\{base\}$ axes, and the rotational transformation between both is expected to be small. The three points are located 100 [mm] separated from each other, following the x and y directions.

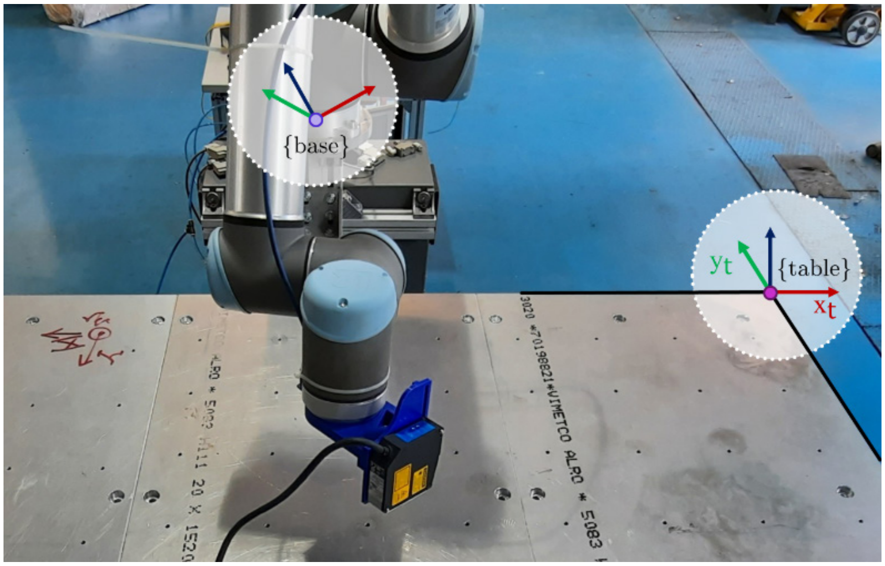


Figure 7.6: Definition of axes for $\{table\}$.

Once the $\mathbf{T}_{xy}$ transformation has been established, the transformations $\mathbf{T}_z$ and $\mathbf{T}_t$ are calculated. The remaining four points located in the edges of the square shape are represented in Figure 7.7.

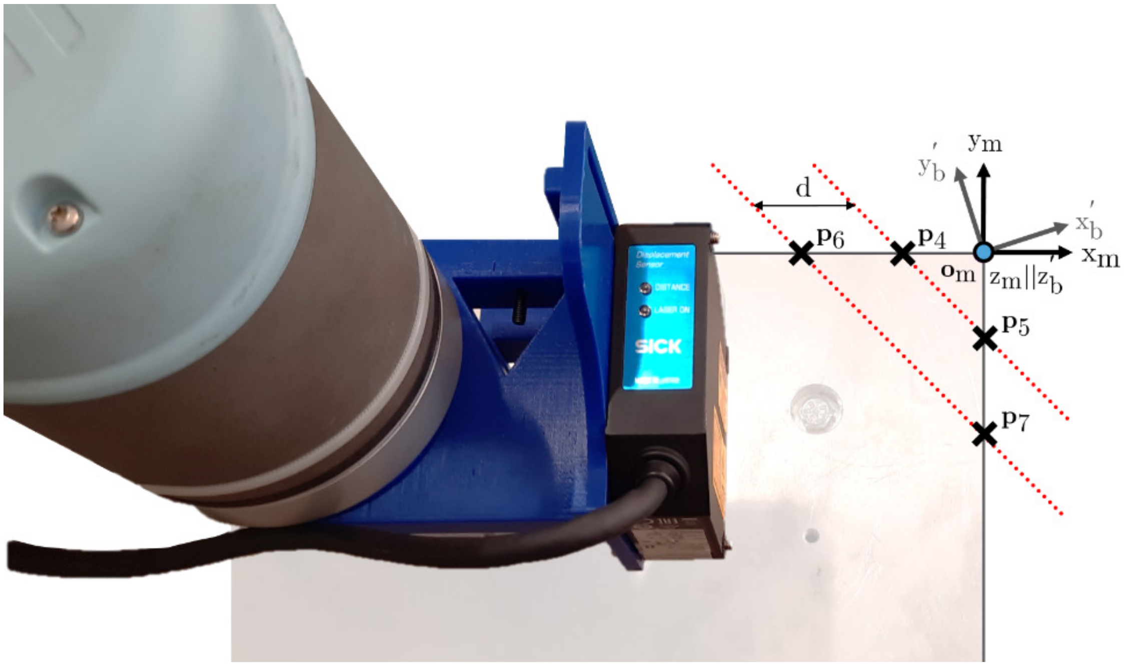


Figure 7.7: Points located in the edges of the calibration totems, used for the calculation of the 2 remaining transformations.

Alignment of x and y axes is then performed using two reference points on each axis utilizing two straight trajectories, with lines separated a distance of 100 [mm], as shown in Figure 7.7. The rotation angle and origin translation are computed using the following equations:

$$\begin{cases} (\mathbf{p}_4 - \mathbf{p}_6)t_1 + \mathbf{p}_4 = \mathbf{o}_t \\ (\mathbf{p}_7 - \mathbf{p}_5)t_2 + \mathbf{p}_5 = \mathbf{o}_t \end{cases} \xrightarrow{yields} \mathbf{o}_t \tag{7.12}$$

$$\mathbf{x}_t = \frac{\mathbf{p}_4 - \mathbf{p}_6}{\|\mathbf{p}_4 - \mathbf{p}_6\|} \tag{7.13}$$

$$\mathbf{u}_{\mathbf{R}_z} = \frac{\mathbf{x}_t \times (\mathbf{p}_7 - \mathbf{p}_5)}{\|\mathbf{x}_t \times (\mathbf{p}_7 - \mathbf{p}_5)\|} \tag{7.14}$$

$$\alpha_2 = \arccos(\mathbf{x}_b^T \cdot \mathbf{x}_t) \tag{7.15}$$

$$\mathcal{R}_z = \begin{pmatrix} \mathbf{u}_{\mathbf{R}_z}^{3\times1} \\ \mathbf{0}^{3\times1} \end{pmatrix} \tag{7.16}$$

$$\mathbf{T}_z = e^{[\mathcal{R}_z]\alpha_2} = \begin{pmatrix} \mathbf{R}_z & \mathbf{0}^{3\times1} \\ \mathbf{0}^{1\times3} & 1 \end{pmatrix} \tag{7.17}$$

$$\mathbf{T}_t = \begin{pmatrix} \mathbb{1}^{3\times3} & \mathbf{o}_m^{3\times1} \\ \mathbf{0}^{1\times3} & 1 \end{pmatrix} \tag{7.18}$$

The complete transformation between the two frames $\{base\}$ and $\{table\}$ (or any other two extrinsically calibrated frames) is then:

$$\{table\} = \{base\}\mathbf{T}_{xy}\mathbf{T}_z\mathbf{T}_t \tag{7.19}$$

To validate the proposed calibration workflow, a demonstrator was implemented on the NPAM station. Of note, the Universal Robots UR10 collaborative robotic arm offers a positioning repeatability of 0.2 [mm]. The work is conducted the station's table. Two verification artifacts, with a cubic calibration totem and a planar surface, are affixed to the table, aligned with its three coordinate axes, as depicted in Figure 7.8.

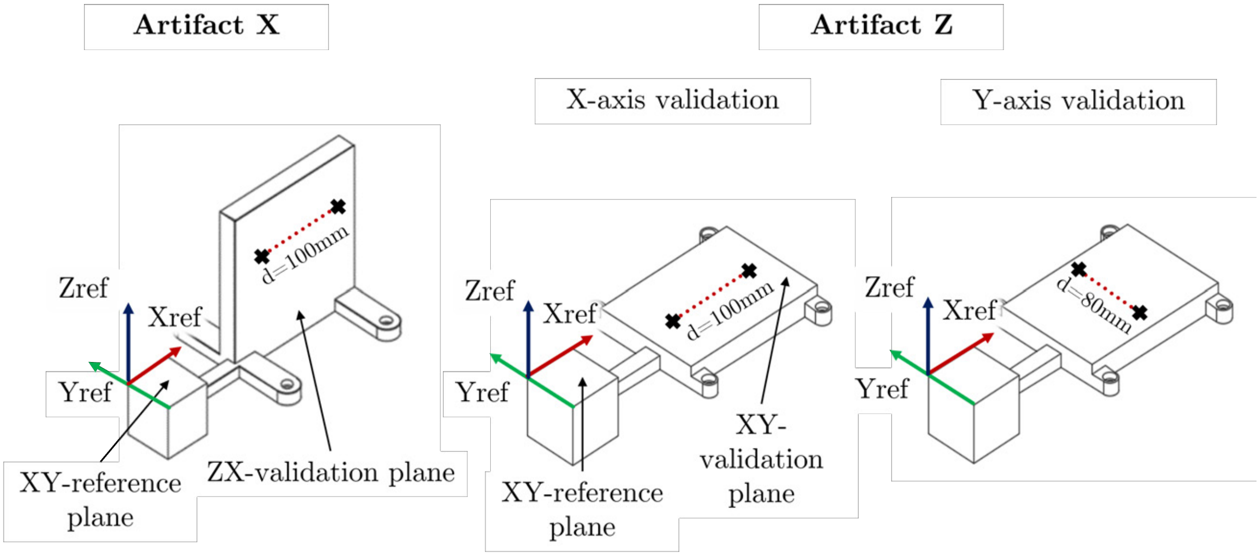


Figure 7.8: Validation artifacts employed, illustrating reference and validation regions, axis orientations, and proposed measurements.

These artifacts simulate the fixture reference via the cubic element, enabling quantification of the reference system correction for the workpiece when applying the proposed method versus the default reference system. The angles between the reference planes and verification zones of the artifacts were measured using a Mitutoyo Crysta-Plus M544 coordinate measuring machine (CMM). Tables 7.1 through 7.3 present the results obtained from CMM measurements of the reference artifacts. Theoretical trajectory start points were used to determine expected positioning errors, while expected angular errors were computed considering deviations between reference and validation planes and overall surface flatness.

Table 7.1: Artifact X: Validation along x-direction.

Expected positioning error [mm]	+0.49
Angle between zx planes (reference-validation) [°]	-0.073
zx validation flatness [mm]	0.313
Expected angular error [°]	-0.073 ± 0.18

Figures 7.9, 7.10, and 7.11 illustrate the trajectory results. Left-hand plots show pre-calibration data, right-hand plots show post-calibration. Colored lines indicate different trial repetitions,

Table 7.2: Artifact Z: Validation along x-direction.

Expected positioning error [mm]	-0.33
Angle between xy plane and x-direction validation [°]	-0.053
xy validation flatness [mm]	0.053
Expected angular error [°]	-0.053 ± 0.03

Table 7.3: Artifact Z: Validation along y-direction.

Expected positioning error [mm]	-0.29
Angle between xy plane and y-direction validation [°]	-0.085
xy validation flatness [mm]	0.053
Expected angular error [°]	-0.085 ± 0.03

with raw data in translucent lines and regression fits in bold. Insets show individual initial positioning and angular errors. Table 7.4 provides a statistical summary of all trajectories.

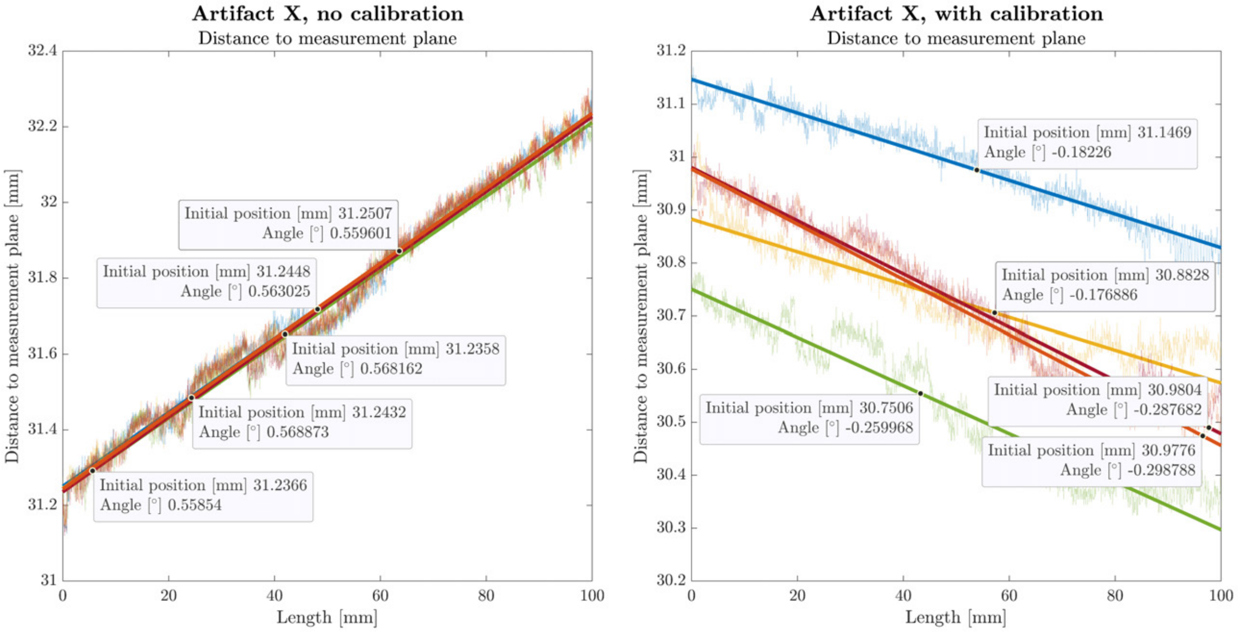


Figure 7.9: Verification distance results for Artifact X, pre-calibration (left) and post-calibration (right).

These results lead to several insights. Pre-calibration measurements showed lower standard deviations than post-calibration, due to the inclusion of relative positioning via the proposed method. This approach adds uncertainty from sensor and reference imperfections and requires solving inverse kinematics at multiple stages, introducing numerical and kinematic parameter errors. Nonetheless, if fixture calibration is infrequent within a production run, the cumulative effect of this uncertainty remains limited.

Angular correction about the x-axis is clearly improved (as shown by results from Artifact

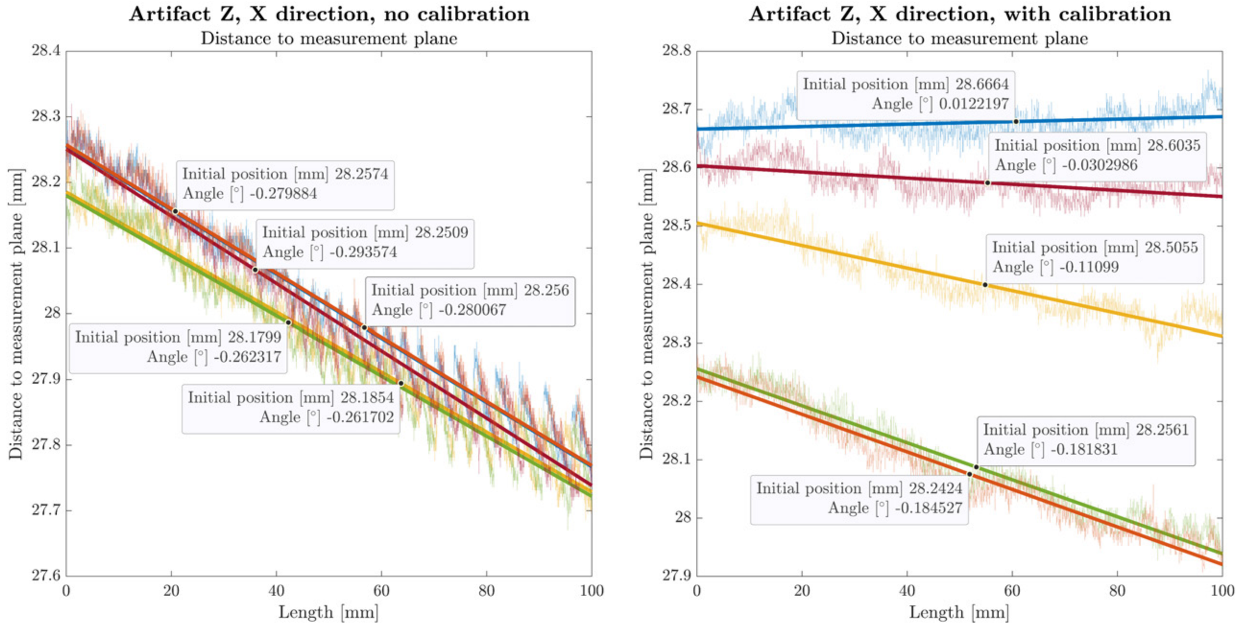


Figure 7.10: Verification distance results for Artifact Z along x, pre-calibration (left) and post-calibration (right).

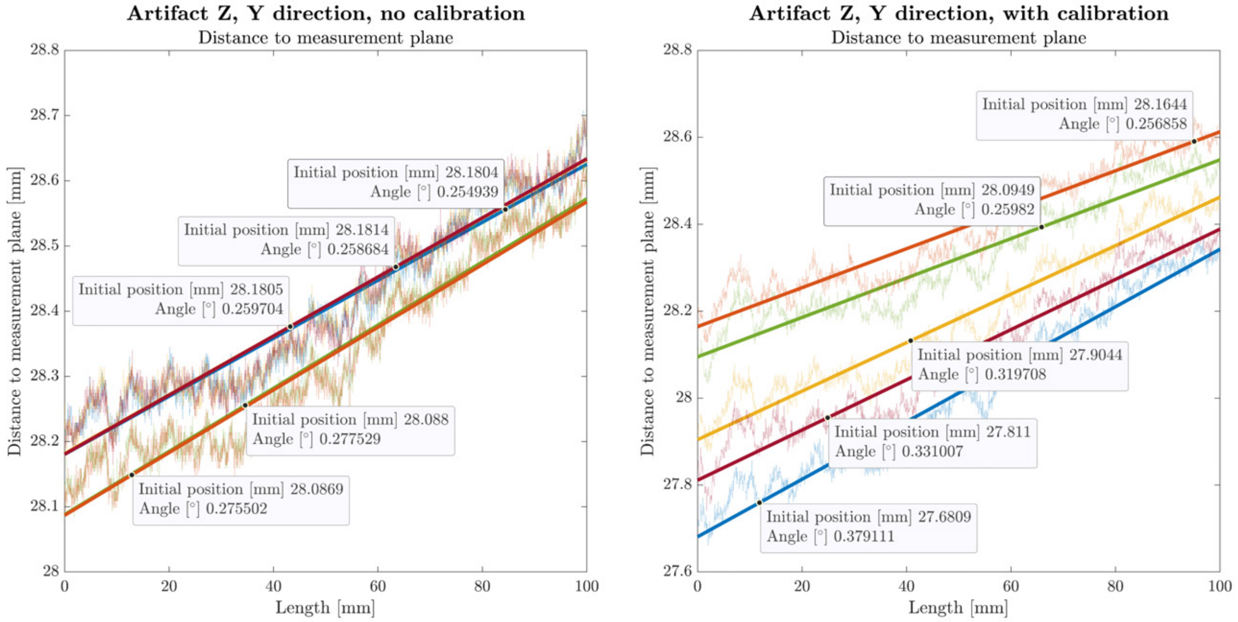


Figure 7.11: Verification distance results for Artifact Z along y, pre-calibration (left) and post-calibration (right).

Table 7.4: Summary of trajectory results.

Trajectory	Exp. Pos. [mm]	Avg. [mm]	Std. Dev. [mm]	Exp. Angle [°]	Avg. [°]	Std. Dev. [°]
Art. X (pre)	0.49	1.242	0.0062	-0.073±0.18	0.564	0.00476
Art. X (post)	0.49	0.948	0.146	-0.073±0.18	-0.241	0.058
Art. Z x (pre)	-0.33	-0.474	0.040	-0.053±0.03	-0.276	0.014
Art. Z x (post)	-0.33	-0.245	0.196	-0.053±0.03	-0.102	0.085
Art. Z y (pre)	-0.29	-0.557	0.051	-0.085±0.03	0.332	0.013
Art. Z y (post)	-0.29	-0.769	0.199	-0.085±0.03	0.386	0.065

X and Artifact Z, x-direction), while rotation about the y-axis shows no such improvement. This may be due to several factors. First, the laser sensor's fixture may not ensure perfect perpendicularity to the robot flange. Manufacturing defects can misalign the laser beam from the TCP's theoretical z-axis, introducing both position and angular errors. Improved fixturing and precision manufacturing would help mitigate this. Additionally, this type of laser sensor is not optimized for edge detection in the xy plane, introducing systematic positioning errors, particularly seen in Artifact X results. Finally, mechanical fixation of artifacts using bolts may induce deformations not present during CMM measurements.

Compared with existing work, the results align with prior findings. In computer vision-based methods, Diao et al. [129] reported 0.2 [mm] absolute error, while D'Avella et al. [130] achieved 0.05–0.5 [mm]. In laser-tracker-based approaches, Daiki et al. [131] reported 0.3 [mm] error, and Nubiola and Bonev [24] achieved 0.36 [mm]. Lei and Zhang [132] reduced absolute positioning error to below 0.1 [mm] using closed-loop correction.

Overall, the presented results are comparable to more complex and costlier methods, confirming the method validity for the extrinsic calibration required in this Thesis.

7.3 Calibration of intrinsic reference frames

Regarding the branch of the workflow diagram in Figure 7.2 related to the definition of intrinsic reference systems, it involves establishing the relationships between the reference systems of the equipment components that depend on kinematic configuration. The main reference frame to locate is the $\{calibration\ cube\}$, which is rigidly joint to the $\{build\ platform\}$. Knowing the exact position of the build platform (through the calibration cube in this case) in the work environment is essential to bring the workpiece's target point into alignment with the extruder, which remains stationary while the part moves.

The method utilized for intrinsic calibration is similar to the one used for extrinsic calibration, with a few key differences. The method is executed in-situ during the online set-up procedure (prior to operation), as any change to the build platform affects the $\{build\ platform\}$. This was not the case in extrinsic calibration procedures, as the environment elements are not that likely to change over time.

The intrinsic calibration procedure is divided in three steps, which can be summarized as:

1. Leveling the calibration cube surface through 3 characteristic points (same as step 1 of the extrinsic calibration procedure).
2. Adjusting the z-distance with respect to the fixed distance sensor. This step is done to ensure the calibration cube edges are properly detected.
3. Aligning the $\{calibration\ cube\}$ virtual origin with the actual calibration cube origin (steps 2 and 3 of the extrinsic calibration procedure).

Regarding stage one, three probe points $P_i = (x_i,\, y_i,\, d_i),\ i \in \{1, 2, 3\}$, where d_i is the laser range (converted from the analog voltage as explained in Section 7.2), define the local surface plane

$$z(x,y) = \beta_x\, x + \beta_y\, y + \beta_0.$$

The gradients are obtained in closed form:

$$\beta_x = \frac{(y_2 - y_1)(d_3 - d_1) - (y_3 - y_1)(d_2 - d_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}, \tag{7.20}$$

$$\beta_y = \frac{(x_3 - x_1)(d_2 - d_1) - (x_2 - x_1)(d_3 - d_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}. \tag{7.21}$$

With $\mathbf{n} = (-\beta_x, -\beta_y, 1)^\top \big/ \|\cdot\|$, a minimal-angle rotation $\mathbf{R}_{xy} = e^{\alpha_1\,[\mathbf{u}_1]}$ aligns TCP-z to $\mathbf{n}$. A fourth range sample at the reference point yields $\Delta z = d_4 - D$, realizing the nominal offset D through the transformation $\mathbf{T}_d$ for stage 2. This nominal offset depends on the distance sensor utilized. As previously stated, in this Thesis, two distance sensors have been interchangeably used: SICK OD5-30W05, and SICK OD5-350W100. These two sensors have different ranges and accuracies, and for each one of them, D is selected as the best-linearity point in the range (in the middle for both). Therefore, $D = 30$ [mm] for the 30W05 sensor, and $D = 350$ [mm] for the 350W100 sensor.

As in the extrinsic calibration, for origin location (stage 3), two perpendicular scans are executed along the calibration cube diagonal:

1) **Edge detection.** While a URScript thread repeatedly moves between extremal points $(a_{\text{init}}, a_{\text{fin}})$ at 3, the main program freezes the thread whenever the analogue input crosses the edge threshold $V_{\text{edge}} = 13$ and records the flange pose. Four such events deliver the $\{\textit{calibration cube}\}$ coordinates $P_4, \ldots, P_7$.
2) **Corner computation.** The intersection of the two secants (P_4, P_6) and (P_5, P_7) yields the physical origin of the calibration cube $\mathbf{o_{cc}} = P_4 + t_1(P_6 - P_4)$ with $t_1 = \frac{(P_5 - P_4)\times(P_5 - P_7)}{(P_4 - P_6)\times(P_5 - P_7)}$.
3) **Frame alignment.** The normalized vector $\mathbf{e}_x = \frac{P_5 - P_7}{\|P_5 - P_7\|}$ defines the cube x-axis. A yaw rotation $\mathbf{R}_z = e^{\alpha_2[\mathbf{e}^{\mathbf{z}}]}$, $\alpha_2 = \arccos(e_{x,1})$, aligns TCP-$x$ with $\mathbf{e}_x$.

The final homogeneous transform is therefore:

$$\{\textit{calibration cube}\} \;=\; \{\textit{build platform}\}\mathbf{T}_{xy}\, \mathbf{T}_{z1}\, \mathbf{T}_{z2}\, \mathbf{T}_t, \tag{7.22}$$

This intrinsic calibration was tested and validated through two experimental case studies [105]. Both case studies revolve around the spring trajectories developed previously on Chapter 4, utilizing the instrumented robotized cell designed and implemented in Chapter 6. A SICK OD5-30W05 laser sensor, conditioned by an AOD5-P1 module, continuously measured the distance between the $\{\textit{printing location}\}$ ($\{\textit{pl}\}$) and the $\{\textit{measure}_1\}$ frame while the robot traced the computed paths. Preserving this distance within tolerance confirms that an extruder mounted at $\{\textit{extruder}\}$ would deposit material accurately along the reference trajectory. Additionally, it simultaneously evaluates the six-DOF extrinsic calibration $\{\textit{base}\} \rightarrow \{\textit{measure}_1\}$ and intrinsic calibration $\{\textit{j5}\} \rightarrow \{\textit{calibration cube}\}$.

Two case studies were examined. Each build platform, planarly manufactured in PLA on an Ultimaker S5, is described by composite Bézier control polygons (Tables 7.5 and 7.6): (i) a cylinder defined by two curve segments and (ii) an hourglass shape defined by four curve segments. Around each platform a 2 [mm]-wide spring was conformally wrapped and radially offset by 2 [mm]. Both springs were planarly manufactured along with the build platform, as if the objects had already been manufactured. These build platforms are introduced in Figure 7.12.

Cylinder build platform Cubic Bézier control polygon			
Segment 1		Segment 2	
x [mm]	z [mm]	x [mm]	z [mm]
15	40	15	25
15	35	15	20
15	30	15	10
15	25	15	0

Table 7.5: Cylinder build platform: control polygon for cubic Bézier composite curve.

Hourglass build platform Cubic Bézier control polygon							
Segment 1		Segment 2		Segment 3		Segment 4	
x [mm]	z [mm]	x [mm]	z [mm]	x [mm]	z [mm]	x [mm]	z [mm]
0	40	15	35	11	20	15	5
10	40	15	28	11	13	15	-2
15	42	11	27	15	12	10	0
15	35	11	20	15	5	0	0

Table 7.6: Hourglass build platform: control polygon for cubic Bézier composite curve.

The spring trajectories are represented in Figures 7.13 and 7.14. For each geometry one-and-a-half coils of the first layer were executed in the robotized station, a representative subset capturing system accuracy. This shortens the test time for each case, and provides clarity in the understanding of the trajectories. These subsets consists of 500 points out of 14739 for the cylindrical case, and 3000 points out of 73389 for the hourglass case.

The trajectories were executed at a linear speed of 20 [mm/s]; total motion times were approximately 7200 [s] for the cylindrical and 7500 [s] for the hourglass artifact. If the online set-up times were accounted for in a real scenario, the build cycle time would be incremented by $\sim$ 180 [s] of platform heating and calibration and $\sim$ 210 [s] of platform cool-down.

Figure 7.12: Cylindrical spring on top of its build platform (left), hourglass spring on top of its build platform (right).

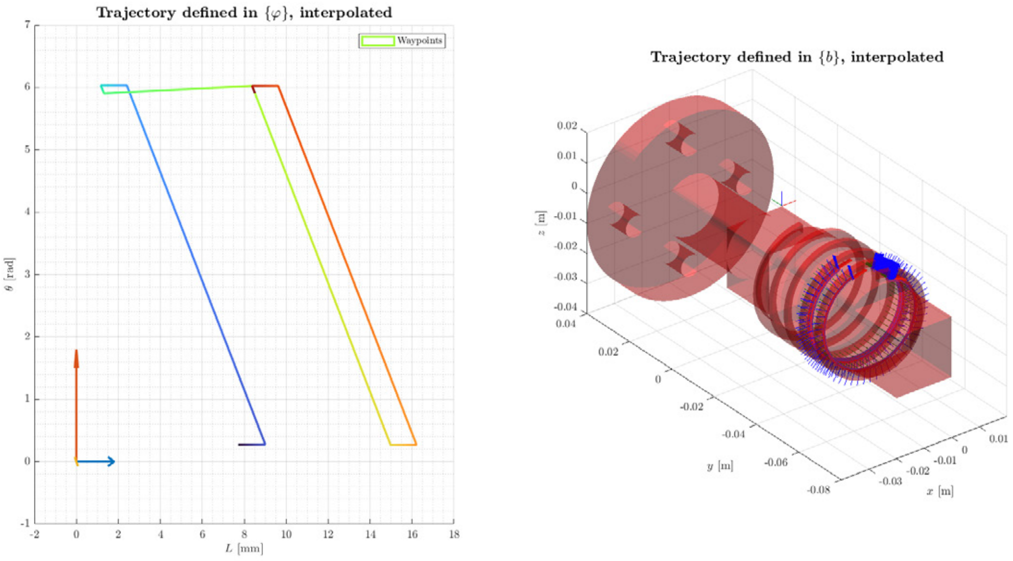


Figure 7.13: Cylindrical spring trajectory represented on the embedded map $\{\varphi\}$ (left) and on the build platform $\{b\} \equiv \{cal\ cube\}$ (right).

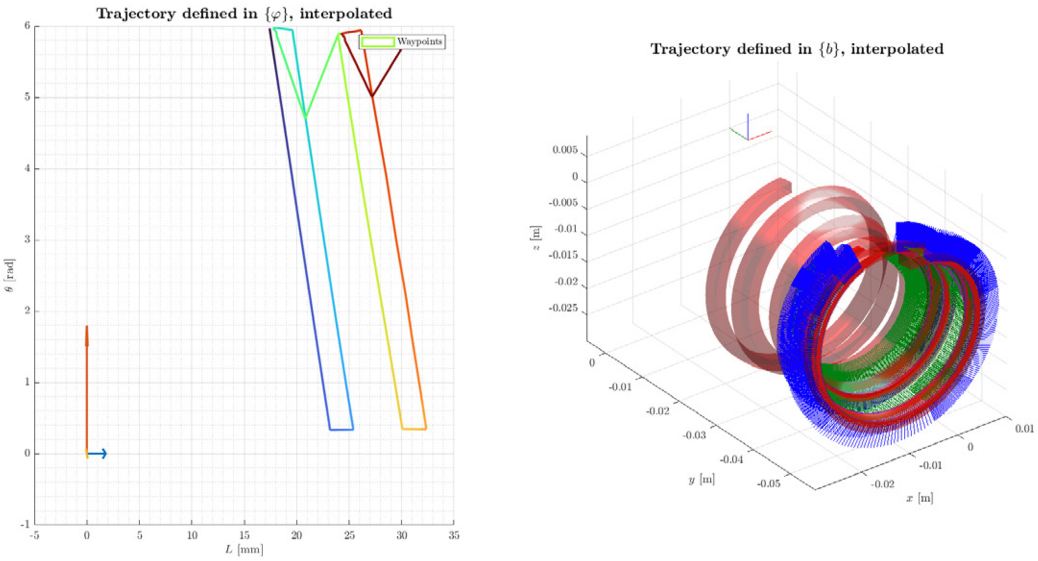


Figure 7.14: Hourglass spring trajectory represented on the embedded map $\{\varphi\}$ (left) and on the build platform $\{b\} \equiv \{cal\ cube\}$ (right).

The distance between the printing location $\{pl\}$ and the distance sensor $\{measure_1\}$ frames are represented in Figures 7.15 and 7.16. Five repetitions were done for each trajectory, as showcased in Figures 7.17 and 7.18.

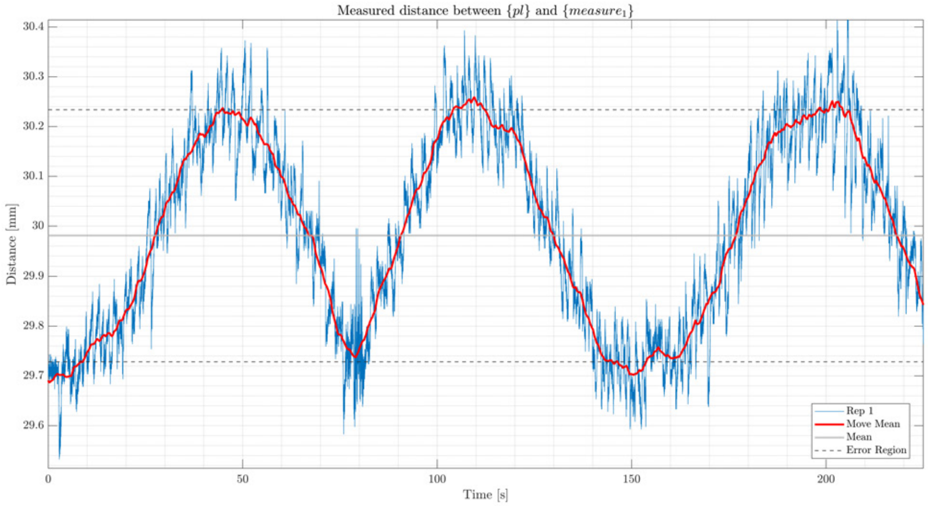


Figure 7.15: Distance results for the cylinder case.

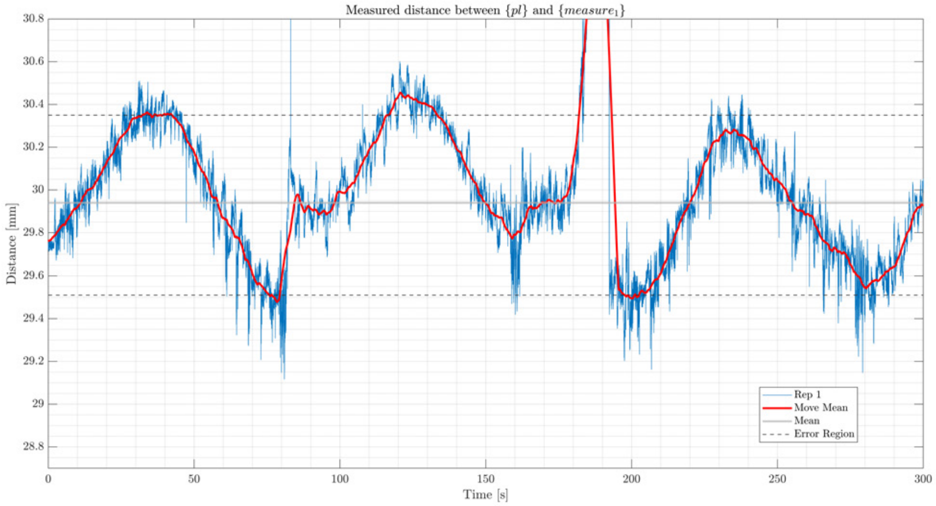


Figure 7.16: Distance results for the hourglass case.

Distance traces show peak deviations of ±250 [μm] for the cylinder and ±350 [μm] for the hourglass. The former aligns with the UR10 manufacturer's on-route repeatability (±200 [μm], doubly validated in Chapter 6) plus systematic calibration and inverse-kinematic error; the latter worsens because misalignment along the build-platform axis of revolution becomes relevant on a doubly curved surface. Five repetitions per case demonstrate repeatability and reveal that controller jitter is random whereas calibration and modeling errors are systematic

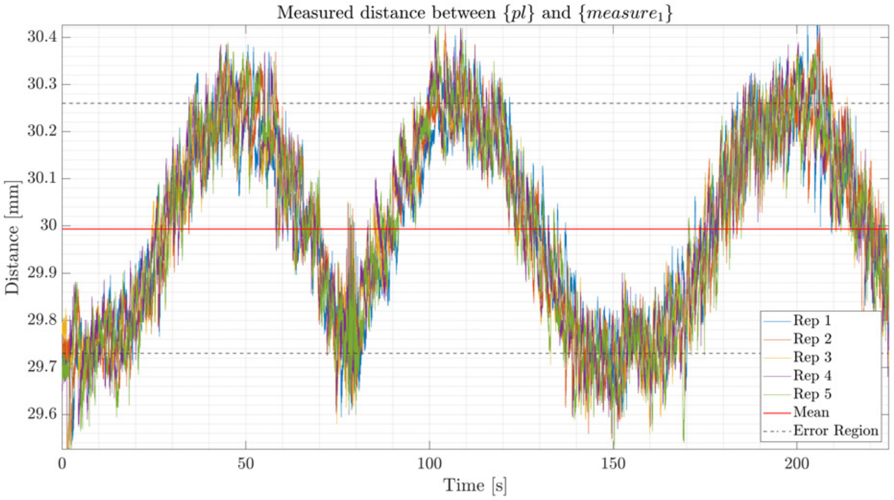


Figure 7.17: Distance results for the cylinder case, five repetitions.

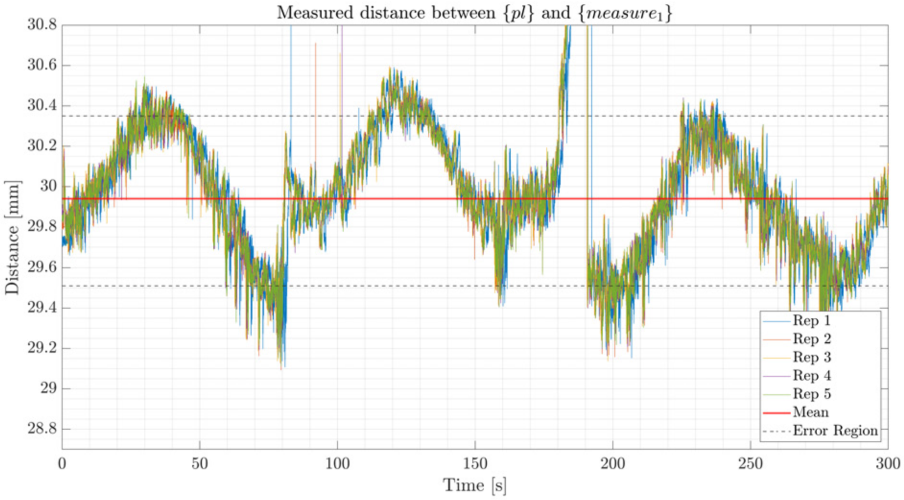


Figure 7.18: Distance results for the hourglass case, five repetitions.

and therefore correctable. The observed performance is comparable or even superior to other robot-based NPAM results reported in the literature [24], [131], [133].

There are several main insights to understand from this experiment. Firstly, Figure 7.15 shows a very distinct distance behavior, in the shape of a discontinuous (especially at around the 75 [s] timestamp) sinusoidal wave. This phenomenon is directly related to the quality of the intrinsic calibration. An error band of 0.5 [mm] implies the actual rotation axis and the calculated rotation axis are that distance apart. Additionally, when moving across printing locations, it is possible to see in longer tests that this error band may increase or decrease. This indicates that the two rotation axes are displaced and rotated with respect to each other, and this opens up opportunities for improving the intrinsic calibration method.

Regarding the hourglass trajectory performed in Figure 7.16, there is an abrupt increase in distance at around the 180 [s] timestamp. This is due to the travel trajectory performed by the slicer when extruding material on different coils. Travel moves have not been examined exhaustively in the context of NPAM and have been left as are. A key improvement that can improve process performance is the implementation of non-planar "z-hops", or movements away from the platform when performing travel moves. These movements have been taken directly from the planar slice, but do not currently take into account part progress nor part or build platform geometry. When accounting for these concepts, it would be possible to implement straight paths between end and start points of rasters that minimize build cycle time.

In terms of repeatability, Figures 7.17 and 7.18 show a good performance of the robotized cell. However, it would be expected of the noise to be more random, since the results portray an almost systematic noise for all five repetitions of both trajectories. This provides an idea that the robot on-route repeatability is not entirely random, and has a systematic error component. Previous analyses (Chapter 6) have shown the kinematic uncertainty behavior in 6-DOF robotic arms is dependent on configuration. Furthermore, dynamic analyses performed on the present robotized cell [127] indicate the dynamic behavior (usually unknown and difficult to characterize due to its dependence on the robot's multi-input multi-output control and pose) is dependent on the whole configuration state-space. This suggests the kinematic behavior of the manipulator may also be dependent on the configuration state-space, and therefore, predictable. This is another key insight obtained from these tests, and, at the moment of writing the present Thesis, validation tests for this hypothesis are under way.

Overall, these tests show the robotized cell is not only able to perform NPAM trajectories, but to execute them to a certain degree of accuracy. The tests validate the suitability of the embedded map methods presented in Chapter 4, part of the manufacturing equipment developed in Chapter 6, and the calibration methods developed in the present Chapter. To fully validate the methods presented, next Section illustrates the whole workflow, including the heating build platform, the polymer extruder and their controllers.

7.4 Performance of the machine

To conclude the Chapter, this Section presents a collection of several case studies that showcase the complete methodology conducted on actual equipment. These case studies include the execution with material extrusion of the non-solid non-planar infill trajectories of the shark-scale geometry on the barrel platform and the rectangle on the smooth hourglass platform (Chapter 5), as well as several iterations of the spring part (Chapter 4) on the heating cylinder platform (Chapter 6).

7.4.1 Shark scale-shaped non-solid, non-planar infill on the barrel

This case study revolves around realizing NPAM trajectories on a convex build platform. The objective of this case study is to demonstrate the combination of NPAM trajectories performed by a robot, synchronized with material extrusion. Additionally, the extruder's flow rate will be calibrated according to the results obtained.

The trajectories selected for this case study are the shark-scale non-solid infills developed in Chapter 5. As a reminder, the barrel-shaped build platform is presented through Table 7.7 and Figure 7.19 visually represents the trajectories to be done.

Barrel control polygon			
Segment 1		Segment 2	
x [mm]	z [mm]	x [mm]	z [mm]
0	50	25	25
5	50	25	20
10	50	25	15
25	50	25	0
25	35	10	0
25	30	5	0
25	25	0	0

Table 7.7: Barrel-shaped build platform control polygon.

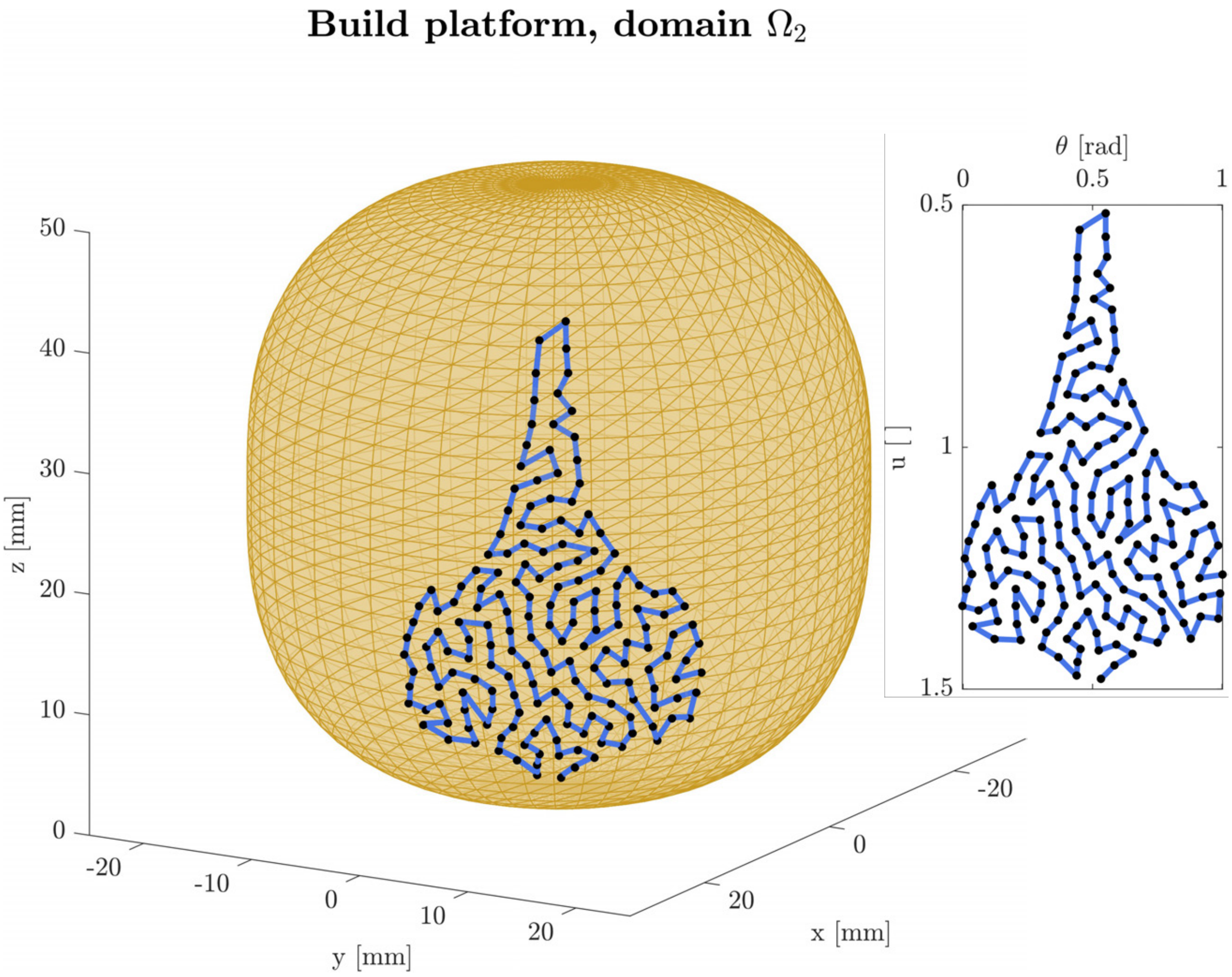


Figure 7.19: Non-solid infill performed in the shark scale shape.

The build platform was planarly manufactured in white Tough PLA in an Ultimaker S5 MEX desktop machine, with a slice height of 0.2 [mm]. Figure 7.20 visually presents both the CAD model, as well as the actual build platforms. A total of three platforms were manufactured for this test.

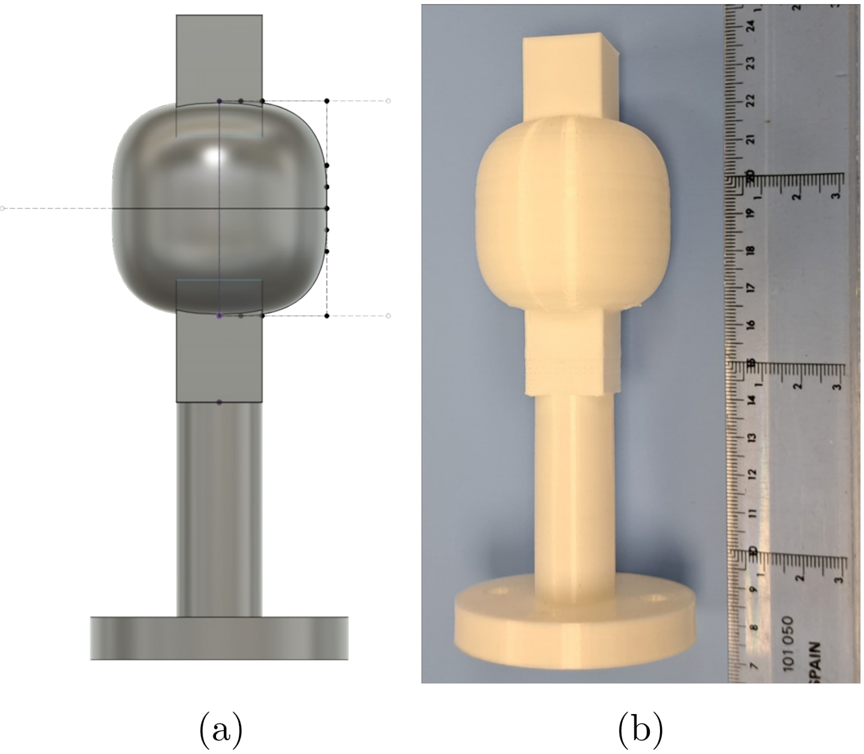
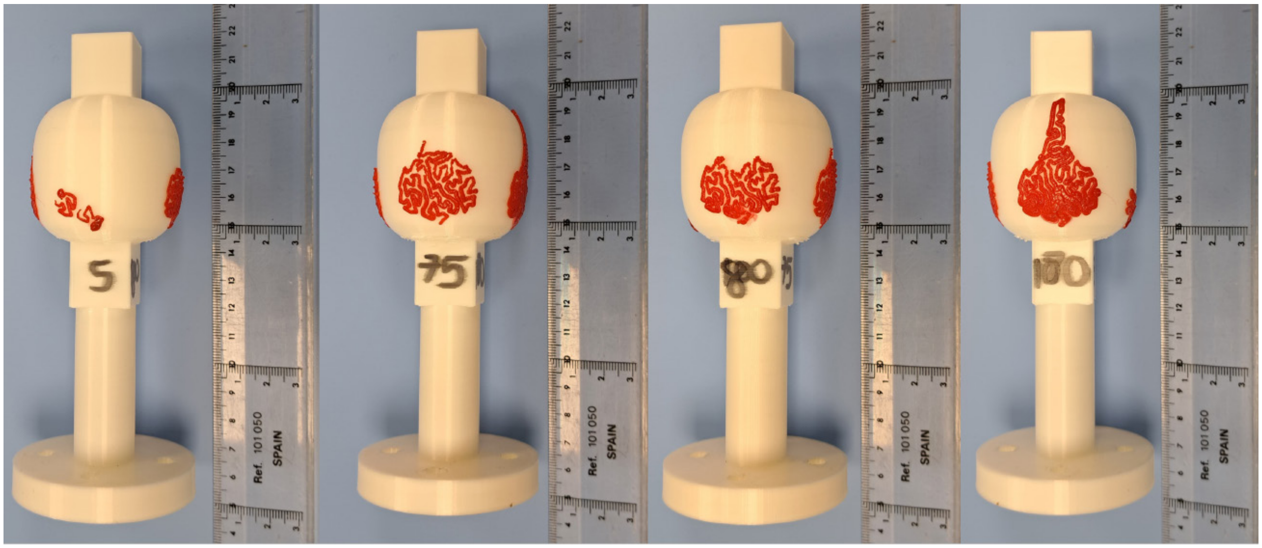


Figure 7.20: Barrel-shaped build platform. (a) CAD model. (b) Actual part.

Several tests were tried out, at extruder feeder speeds of 5 [steps/s], 75 [steps/s], 80 [steps/s], and 100 [steps/s]. The robot speed was set at 5 [mm/s] tangential to the geodesic trajectory, and the NPAM slice height was set to 0.5 [mm]. The first four tests are depicted in Figure 7.21, and serve to establish the proper extruder feeder speed at 80 [steps/s].

These preliminary tests offer meaningful information about the process. The main concern in this stage is material adhesion to the build platform. This is closely related to the actual distance between the build platform and the extruder nozzle. These preliminary tests were not accompanied by a previous calibration, therefore, great variations of distance between reference frames are to be expected. This phenomenon is especially seen on the 75 [steps/s] test, and, to some degree, in the 80 [steps/s] tests. These large distance variations lead to poor adhesion with low flow rates, and can lead to fragile parts. This is the reason the 75 [steps/s] was considered not good enough, as the top region of the part broke during manipulation.

Other notion to take away from this test is the difference between the two extreme cases, 5 and 100 [steps/s]. The 5 [steps/s] showed a very poor material extrusion due to the low flow rate. This resulted in discontinuous extrusion, in addition to poor adhesion, which are the main reasons the part manufacture did not succeed. On the other hand, the 100 [steps/s] had



(a) (b) (c) (d)

Figure 7.21: Barrel-shaped build platform preliminary tests. (a) Extruder speed at 5 [steps/s]. (b) Extruder speed at 75 [steps/s]. (c) Extruder speed at 80 [steps/s]. (d) Extruder speed at 100 [steps/s].

proper adhesion, and the part was successful. This is reinforced by the idea that during planar MEX-P processes, the material is slightly squashed by the nozzle to realize the command line width. However, the command line width of 0.4 [mm] is generally not realized, especially at the bottom region. This could be owed to a multitude of factors, of which, the quality of the intrinsic calibration is thought to be the most significant, as in the previous cases.

After these preliminary tests, two more iterations were done at 80 [steps/s] and 82 [steps/s] to demonstrate the repeatability of the process. These tests are depicted in Figure 7.22.

Once again, the process is overall successful, although the command infill density of 25% is not realized due to the limitations of realizing the NPAM-MEX-P process. This is seen in the bottom-right regions of both iterations, where the command line width is not respected. However, this phenomenon opens an opportunity to better estimate the intrinsic calibration of the build platform from a test line. This type of flow rate calibration is performed in some planar AM-MEX-P equipment through the addition of a LiDAR (short for Light Detection and Ranging)-based system. The LiDAR segments out a test raster performed in an unused region of the build platform, determines how wide is it, and adjusts flow rate and calibration accordingly.

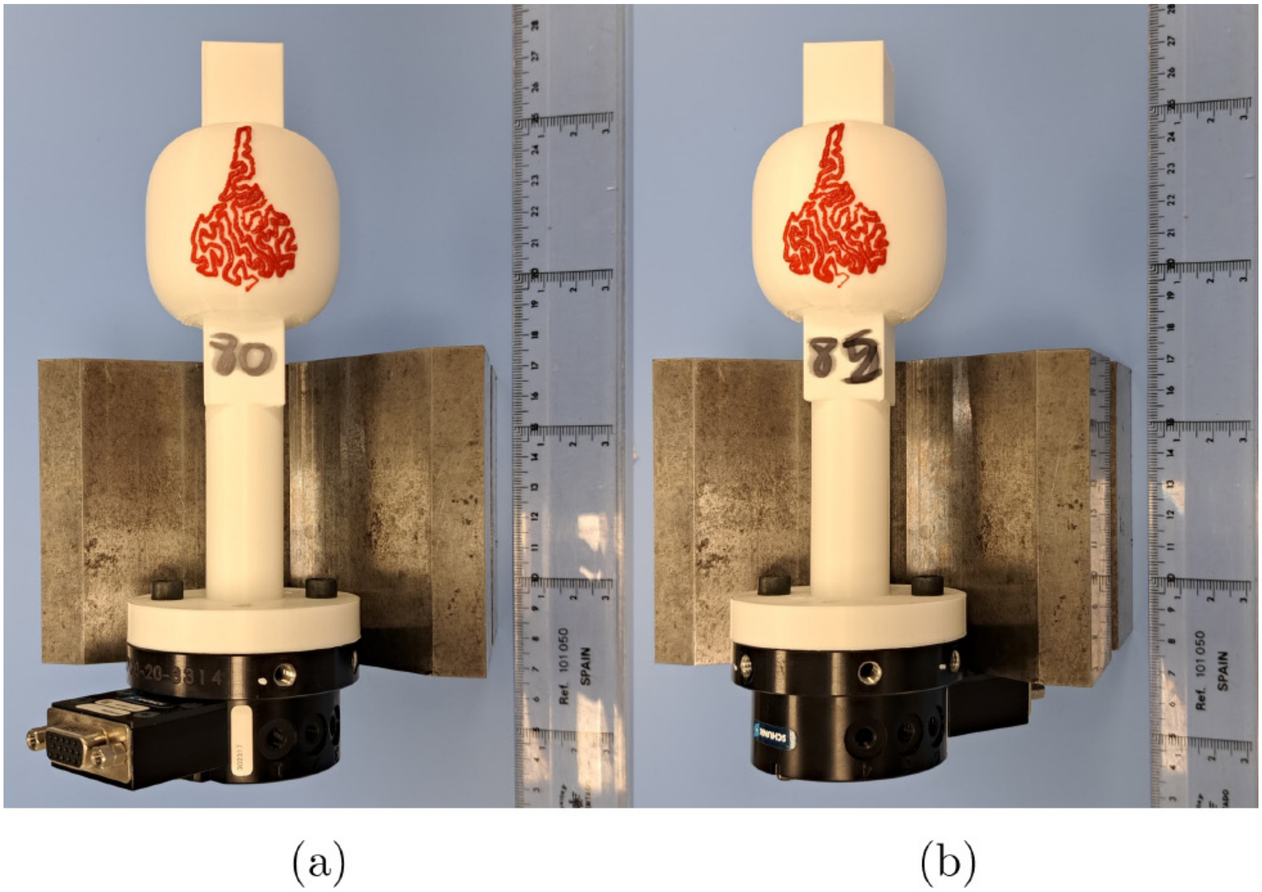


Figure 7.22: Repeatability tests. (a) Iteration 1 at 80 [steps/s]. (b) Iteration 2 at 82 [steps/s].

7.4.2 Rectangle non-solid, non-planar infill on the smooth hourglass

Equipped with the knowledge of actual manufacturing tests performed in the convex platform regarding flow rate, a new test is performed on a concave build platform. The case study selected for this task is the smooth hourglass case presented in Chapter 5. The objective of this test is to verify the workflow robustness on non-convex platforms, which are notoriously difficult to implement due to collision constraints. As a reminder, the hourglass-shaped build platform is presented through Table 7.8 and Figure 7.23 visually represents the trajectories to be done.

Hourglass control polygon							
Segment 1		Segment 2		Segment 3		Segment 4	
x [mm]	z [mm]	x [mm]	z [mm]	x [mm]	z [mm]	x [mm]	z [mm]
0	100	30	85	15	50	30	15
5	100	30	80	15	45	30	10
10	100	30	75	15	40	30	5
50	100	15	70	15	30	50	0
30	95	15	60	30	25	10	0
30	90	15	55	30	20	5	0
30	85	15	50	30	15	0	0

Table 7.8: Hourglass-shaped build platform control polygon.

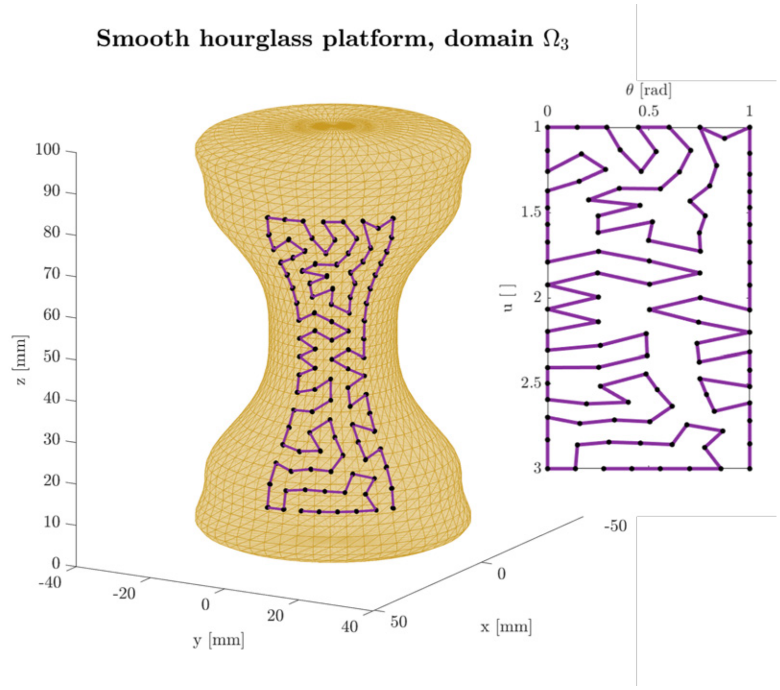
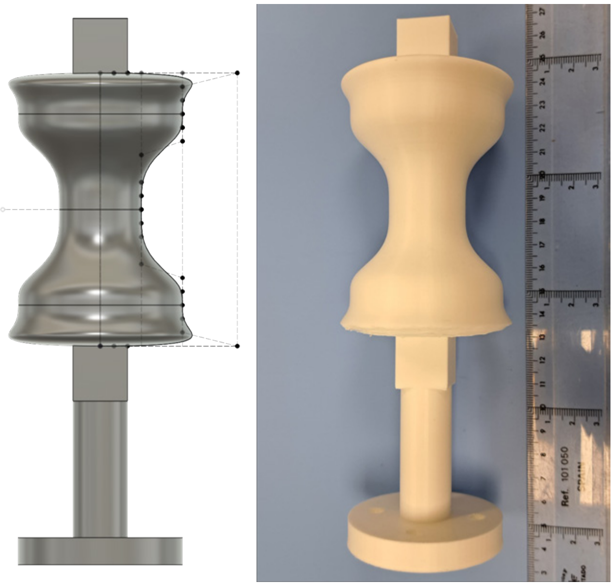


Figure 7.23: Non-solid infill performed in the hourglass platform.

As in the previous test, the build platform was planarly manufactured in white Tough PLA in an Ultimaker S5 MEX desktop machine, with a slice height of 0.2 [mm]. Figure 7.24 visually presents both the CAD model, as well as the actual build platforms. Only one platform was manufactured for this test.



(a) (b)

Figure 7.24: Hourglass-shaped build platform. (a) CAD model. (b) Actual part.

The extrusion test results are depicted in Figure 7.25. As in the previous case, the trajectory was executed at a robot tangential velocity of 5 [mm/s], and the extrusion flow rate was set at 80 [steps/s].

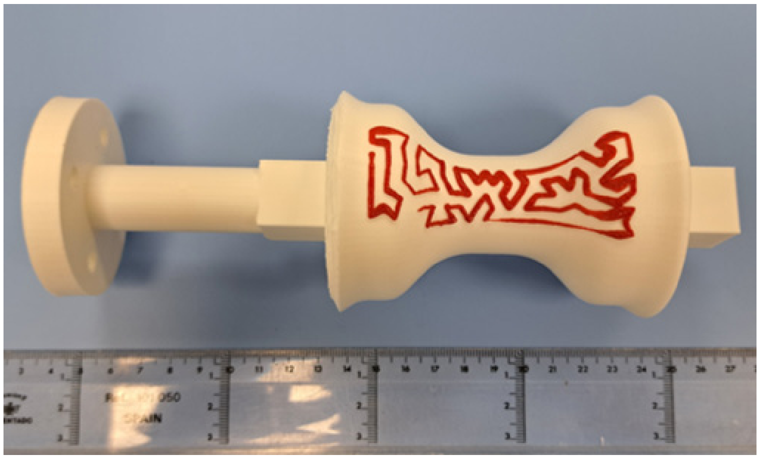


Figure 7.25: Hourglass-shaped build platform test results.

There are several notions to extract from this case study. Firstly, the workflow is successful at following any type of axisymmetric build platform. The only two limitations, which were not accounted for, revolve around the platform bijectivity regions, and checking for collisions between the extrusion equipment and the intrinsic elements. Additionally, as in the previous case, the calibration is an important aspect for process viability. This can be seen in the difference in line width between the left and right hips of the hourglass. However, the intrinsic calibration does not explain the difference between line widths between the central waist and the adjacent regions, where the becomes steeper.

The hypothesis for this discrepancy relies on how Fusion360 (the CAD software utilized for build platform design and planar manufacturing) defines Bézier curves. Fusion360 is thought to use degree 5 Bézier curve compositions, whereas the presented methods require degree 6 Bézier curve compositions. To verify this hypothesis, a comparison between the CAD build platform tessellation model and the `SixDegBezier2DCat` tessellation. This comparison is depicted in Figure 7.26. As predicted, in the steep regions next to the central waist, the actual build platform is almost 1 [mm] closer to the extruder than theoretized. This, in addition to the intrinsic calibration issue previously discussed, are the main reasons the extrusion does not provide a uniform line width.

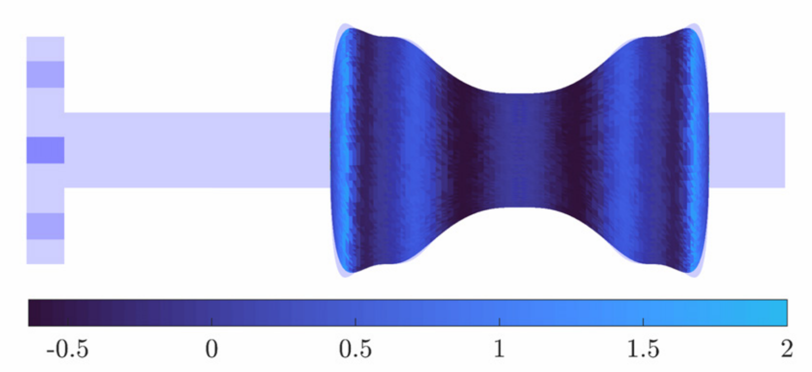


Figure 7.26: Distance (in [mm]) between the CAD (light blue shade in the background) and the theoretical tessellation (in colors, with accompanying legend).

7.4.3 Iterations of the cylindrical spring

To complete this Chapter, several iterations of the cylindrical spring were manufactured. In each of the iterations, a certain improvement was implemented in the workflow, and each of the results is analyzed in depth. The target part is the cylindrical spring presented in Chapter 4. As a reminder, the build platform control polygon, the part, and the initial trajectory are presented in Table 7.9, and Figures 7.27 and 7.28. These case studies aim to serve as a global validation of the process, including extrusion, build platform adhesion, and calibration. The trajectory has been performed at a robot tangential speed of 5 [mm/s], with a flow rate of 80 [steps/s], with a melting temperature of 195 [°C] and a build platform temperature of 60 [°C]. These extrusion parameters are the usual for poly(-lactic acid) polymers, although extrusion temperature is lower than the manufacturer's recommendation [134]. A build platform adhesive was sprayed on it before manufacturing to facilitate adhesion. For brevity, only the first layer was done.

Several pictures have been taken during the process: one during the first coil (Figure 7.29) and another one right at the end of the process, before unloading the part from the build platform (Figure 7.30).

Cylinder control polygon			
Segment 1		Segment 2	
x [mm]	z [mm]	x [mm]	z [mm]
15	40	15	20
15	30	15	10
15	30	15	10
15	20	15	0

Table 7.9: Cylinder-shaped build platform control polygon.

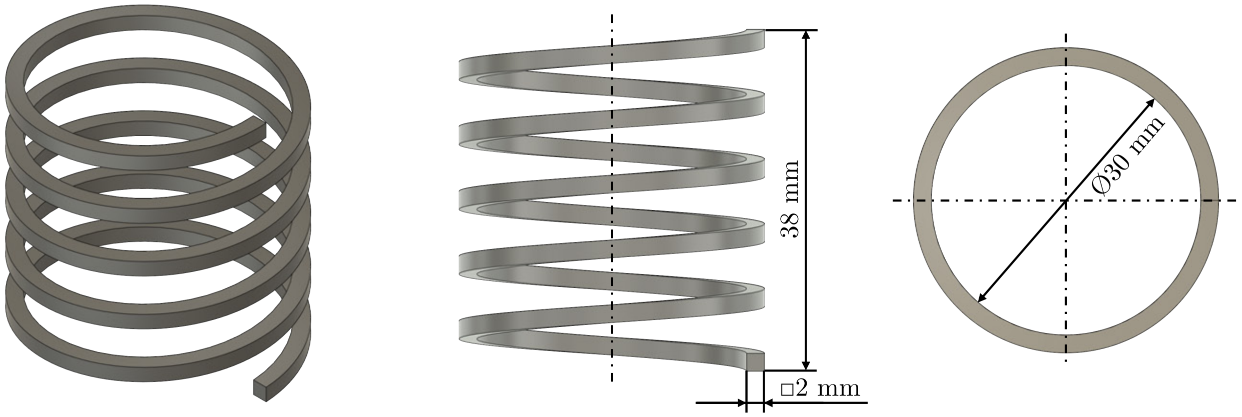
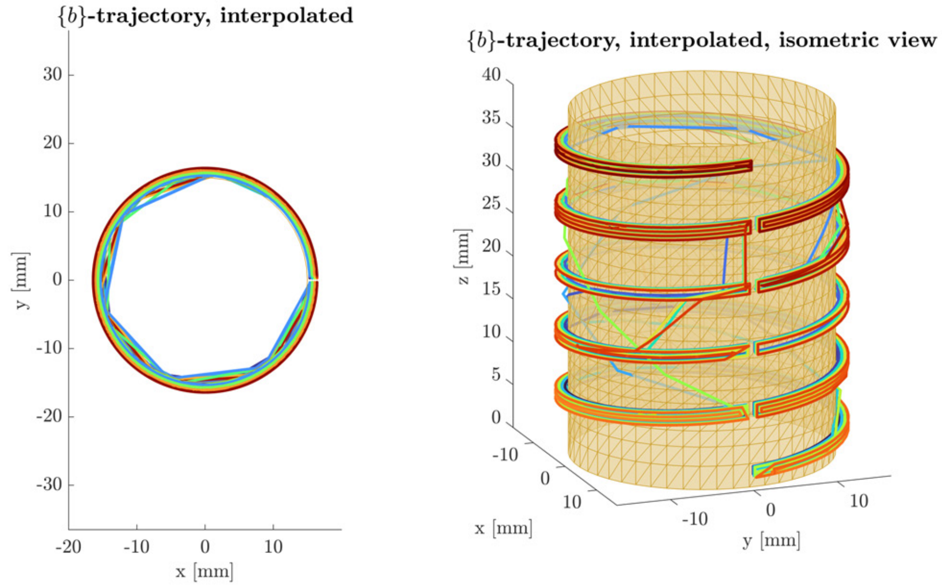


Figure 7.27: Target part for embedded map NPAM-MEX-P manufacturing.


$\{b\}$-trajectory, interpolated

$\{b\}$-trajectory, interpolated, isometric view

Figure 7.28: Target part for embedded map NPAM-MEX-P manufacturing.

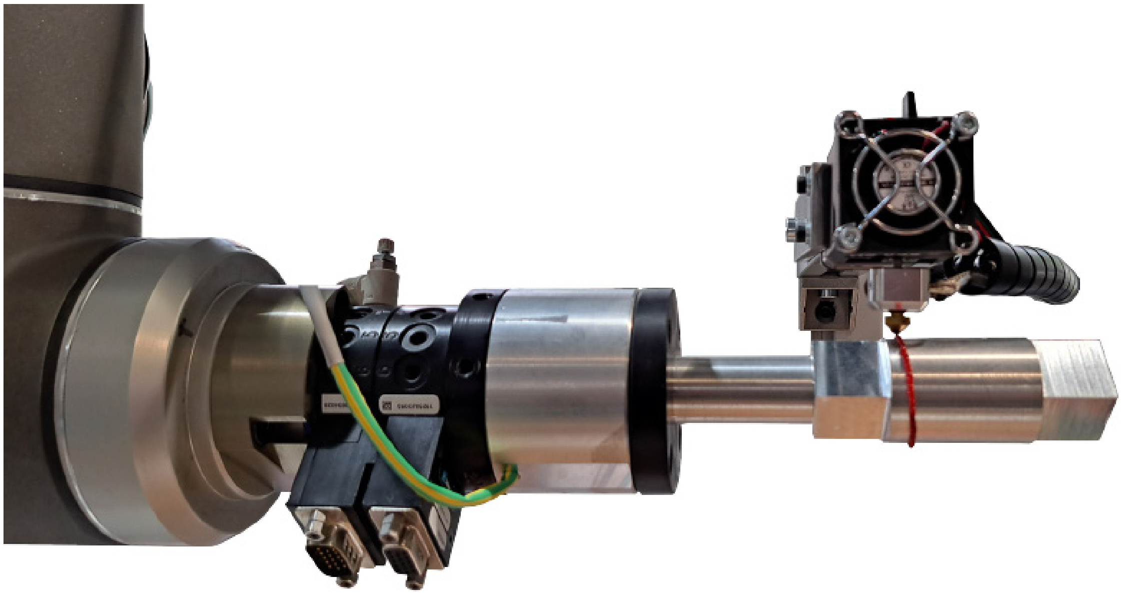


Figure 7.29: Process picture taken at the start of the trajectory.

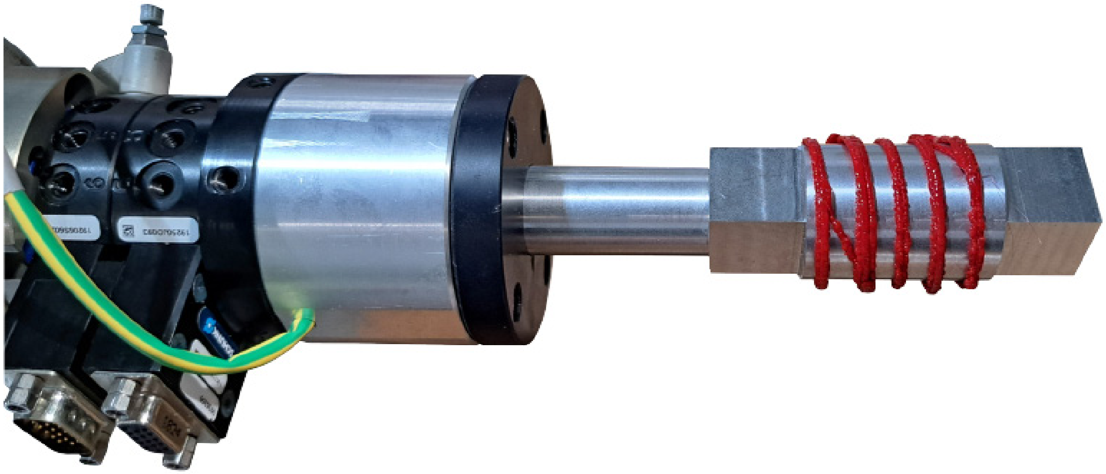
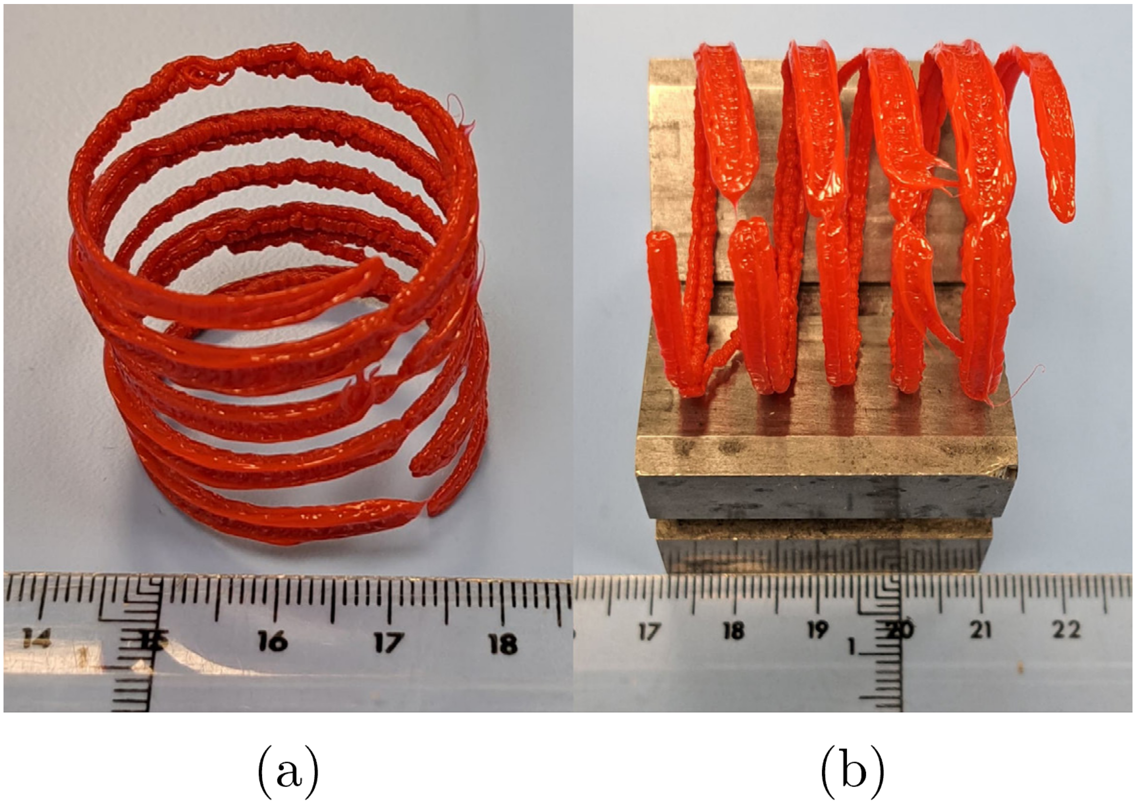


Figure 7.30: Process picture taken at the end of the trajectory.

The first iteration of the spring (after extraction from the build platform) is depicted in Figure 7.31. This is the very first complete part manufactured through the present workflow.



(a) (b)

Figure 7.31: Cylindrical spring, iteration 1. (a) Isometric view. (b) Lateral view.

There are many learnings to take from this case study. The first main insight is regarding build platform adhesion. The temperature used in this test (60 [°C]) is deemed to be too low for proper adhesion. This can be seen in Figure 7.32. This adhesion failure occurred prior to the second coil deposition. This made the second coil unable to overlap with the first coil, rendering the part unsuccessful. To solve this issue, the build platform temperature was set to 70 [°C] in the next iterations. This parameter value is more in line with other references found in the scientific literature [135].

A topic related to build plate adhesion concerns the joints or overlaps between coils. As previously seen in the trajectories depicted in Figure 7.28, it can be noticed that the coil trajectories do not actually overlap. This is due to the tessellation cleansing performed in Section 4.3.2, where the tessellations had some polyhedra removed to provide a clean slicing. This issue induces a lack of overlap in this particular case, which is detrimental to the part's functionality. The lack of coil overlap can also be seen in Figure 7.32. This issue will be addressed in the following iteration.

There were some regions where adhesion was successful, as depicted in Figure 7.33. However, on the opposite side, adhesion is non-existent (as seen in Figure 7.34). This is also caused to poor intrinsic calibration, where the real $\{build\ platform\}$ ($\{bp\}$) reference frame is displace

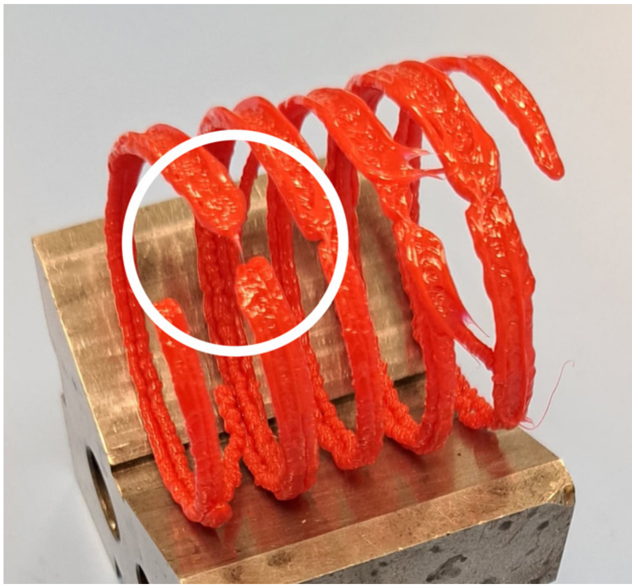


Figure 7.32: Build plate adhesion failure at the end of coil 1 (highlighted in white).

with respect to the assumed $\{bp\}$ frame. This is reinforced by the fact that the polymer looks squashed on the closer side, the one with proper build platform adhesion.

Lastly, there are two main insights regarding travel moves, and the relationship between flow rate and robot tangential velocity. Travel moves have not been completely integrated as of the writing of this Thesis. The usual behavior in planar MEX-P AM machines implies a filament retraction to avoid extruding during travel moves. This was not done for the examples presented, as can be seen in Figure 7.35. Additionally, from a visual inspection, it appears the robot tangential velocity is lower than the command tangential velocity, set at 5 [mm/s]. This behavior will be checked in the next iteration.

Several improvements were made for the second iteration. The main focus of the second iteration was improving build platform adhesion and coil-to-coil junctions. To achieve this, the build platform command temperature was raised to 70 [°C], and the $\{\varphi\}$-part model was scaled in Cura so that the θ scale factor was 10% larger. This is a quick workaround for coil-to-coil junction improvement, and results in the trajectory shown in Figure 7.36. The proper solution would involve helical slicing, as described in Chapter 4. As in the previous case, only the first layer was done for brevity.

The second iteration of the spring was manufactured in white PLA, keeping all parameters equal to the previous iteration except for the aforementioned ones. Two pictures were taken during manufacture, one at the start (Figure 7.37), and another at the end of the process (Figure 7.38). The result is shown in Figure 7.39.

Figure 7.33: Build plate adhesion success at coils 3-5 (highlighted in white).

(a) (b)

Figure 7.34: Build plate adhesion failure at coils 3-5. (a) Spring interior. (b) Spring exterior.

Figure 7.35: Travel moves with extrusion, highlighted in white.

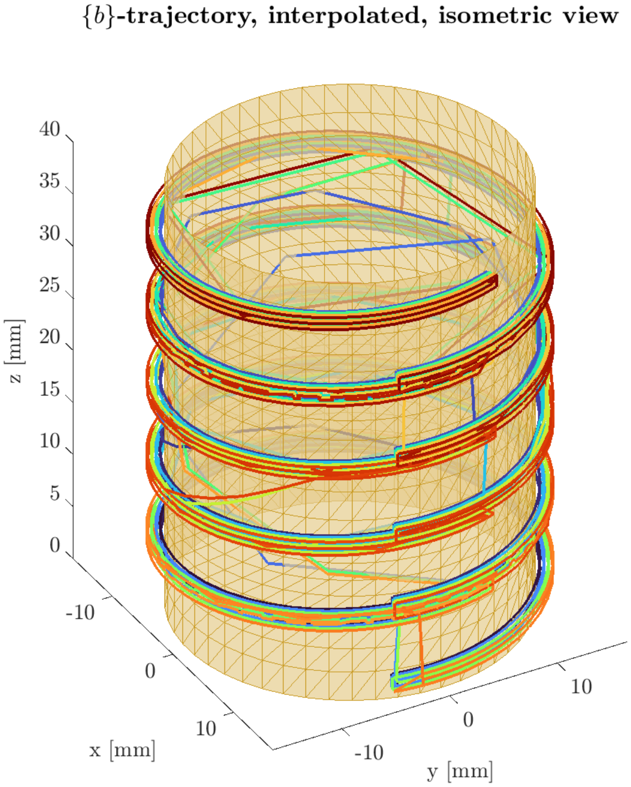


Figure 7.36: Spring iteration 2 trajectory. Only the first layer is done for brevity.

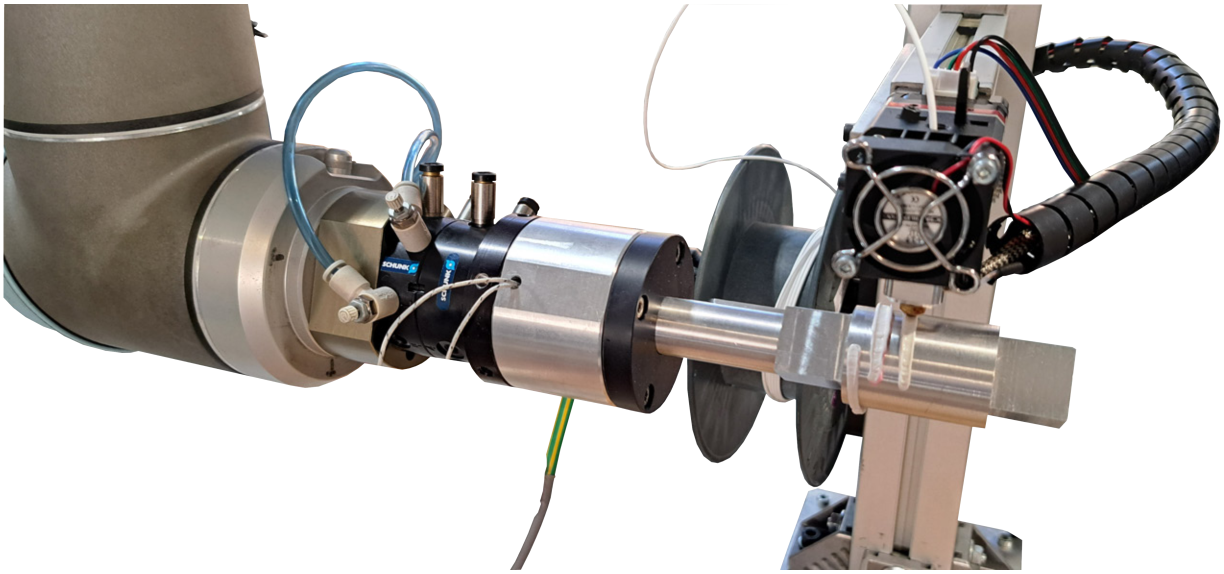


Figure 7.37: Manufacture of spring iteration 2, picture taken during extrusion of coil 2.

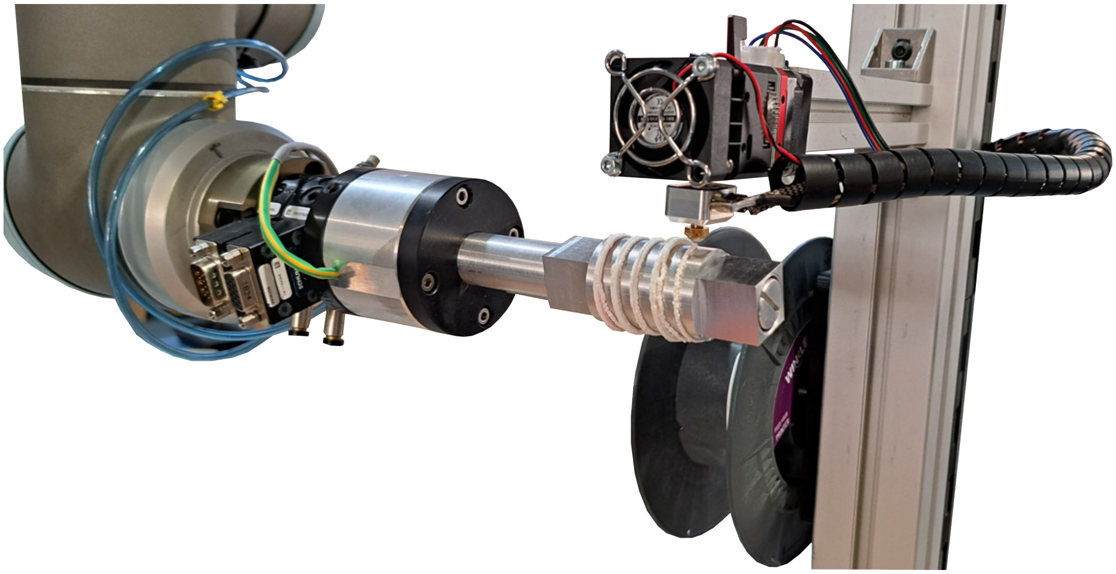
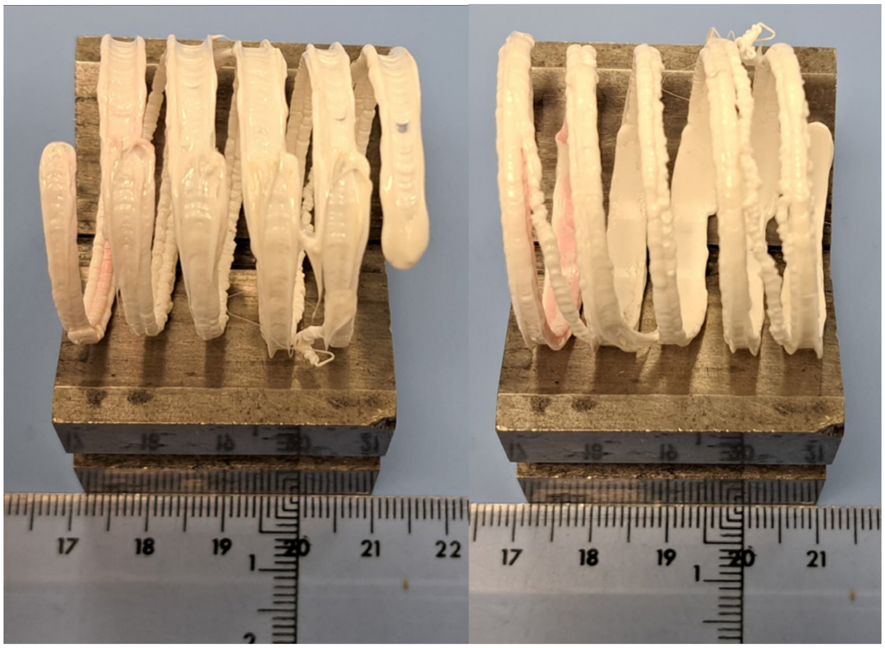


Figure 7.38: Manufacture of spring iteration 2, picture taken at the end of the process.



(a) (b)

Figure 7.39: Spring iteration 2.

As in the previous iteration, there are several realizations to learn from this test. The first main takeaway is that the temperature command of 70 [°C] provides a better adhesion to the build platform, as evidenced by Figure 7.40. However, a defect can be observed in Figure 7.41, where the part appears with a red tint. This is due to an improper purge of the previous filament, which was the red PLA utilized for spring iteration 1. This is not a performance issue, but an aesthetic one.

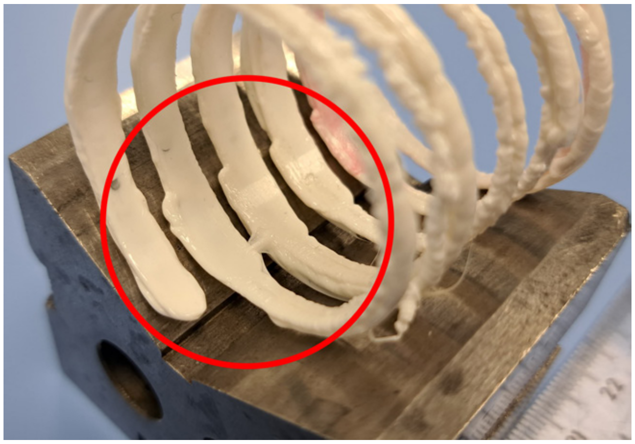


Figure 7.40: Detail of the build platform adhesion, highlighted in red.

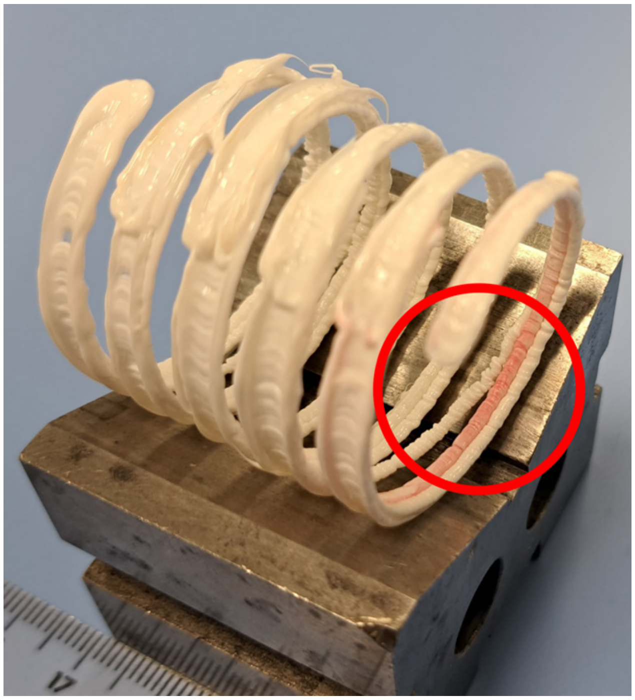


Figure 7.41: Improper purge of the previous filament, highlighted in red.

More serious issues are portrayed in Figure 7.42. Firstly, and as in the previous iteration, the build platform intrinsic calibration is critical for manufacturing success. This can be seen in the difference in build plate adhesion between Figure 7.42a and 7.42b. This once again suggest a large misalignment between theoretical and actual $\{build\ platform\}$ reference frames. Another issue, not seen before, is a slight clash between the build platform and the extruder. This is seen in Figure 7.42b, highlighted in red. This is also due to the intrinsic calibration (or lack thereof).

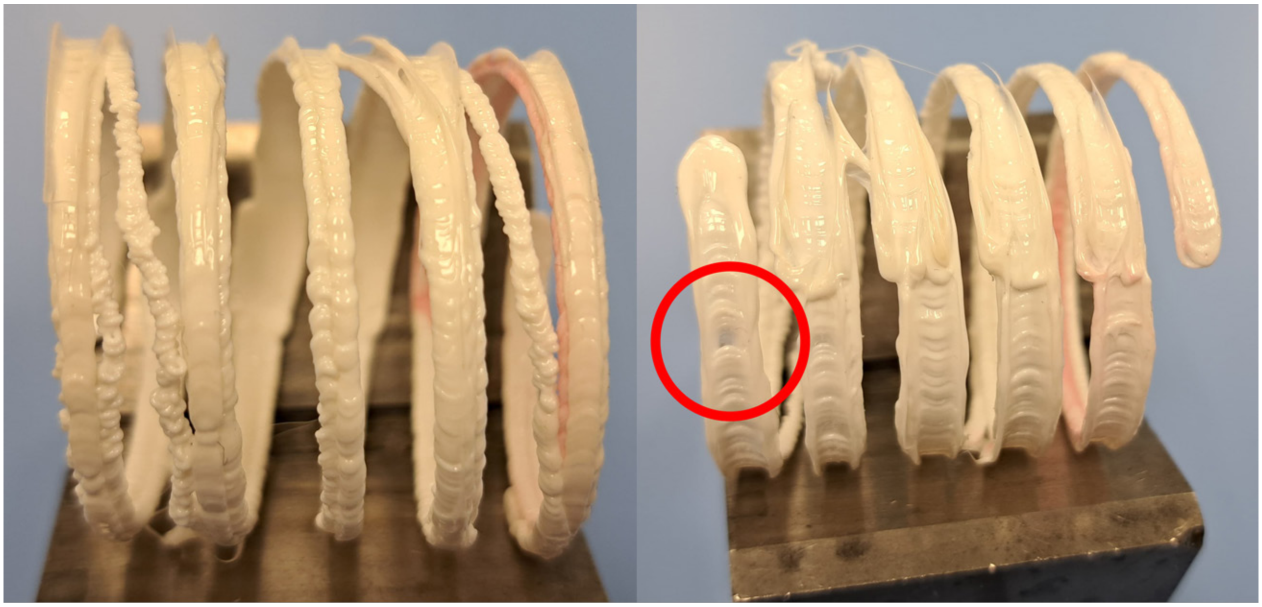

(a) (b)

Figure 7.42: Performance issues of iteration 2. (a) $\theta = \pi$ [rad] viewpoint. (b) $\theta = 0$ [rad] viewpoint. A collision between extruder and build platform is highlighted in red.

Concerning travel moves, the trajectory generated in this case (see Figure 7.36) implemented larger travel moves between coils, which can also be seen in Figure 7.42. These travel moves were done while extruding, which is something that will be corrected in future iterations. A more pressing issue revolves around the shape of the coils in the good adhesion side ($\theta = 0$ [rad]). This good adhesion was fostered by the nozzle's proximity to the build platform. However, there is a large amount of excess material, which tends to grow sideways, then upward. This is the reason for the material curls shown in Figure 7.42b.

This phenomenon points again to the suspicion that the robot's tangential velocity is not 5 [mm/s], but rather much lower. This could be due to the robot's inability to smoothly interpolate the trajectory between waypoints when the waypoints are close to each other. This forces the robot to accelerate and decelerate in short timespans, unable to reach the command tangential velocity. To check this, during the start of the trayectory, a robot data acquisition was performed using UR Log Viewer. The data included joint positions, velocities, accelerations, as well as TCP positions, velocities, and accelerations, both in 6 DoF. The acquisition was undertaken for 5 minutes, at a sampling rate of 125 [Hz], the maximum provided by the user interface of the UR10, 2017 model. The result is shown in Figure 7.43.

Figure 7.43 confirms the tangential velocity issue suspicion is true. Whereas the command velocity was set to 5 [mm/s] (in red), the mean actual tangential velocity was 0.34 [mm/s]. Additionally, the velocity largely varies, depending on the distance between points in the

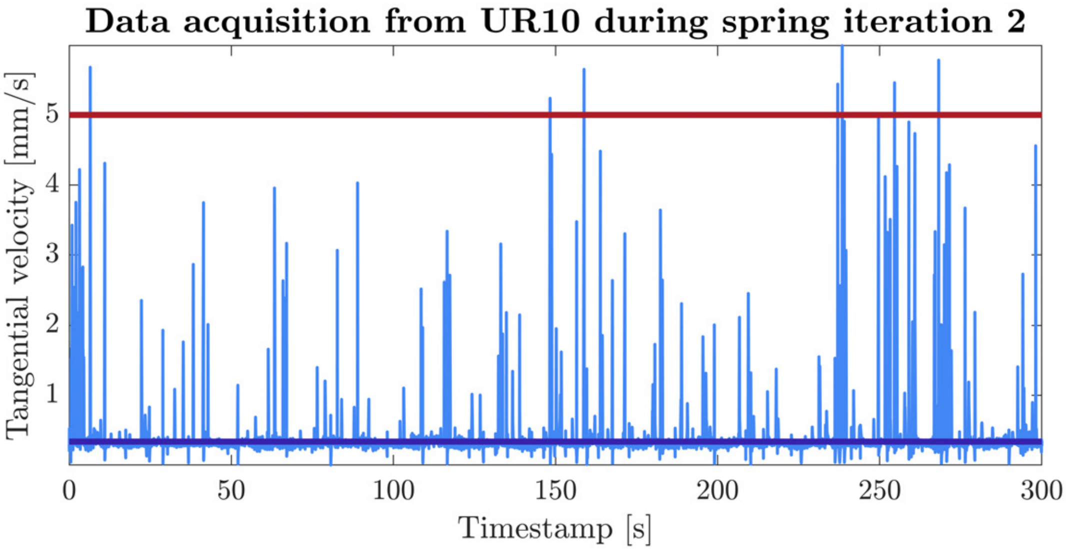


Figure 7.43: Tangential velocity acquisition in iteration 2. The recorded tangential velocity is shown in light blue, the command tangential velocity in red, and the mean actual tangential velocity in dark blue.

trajectory. This induces several detrimental effects on manufacturing. Firstly, the much lower than anticipated tangential velocity encourages material build-up, resulting in the curls previously shown on Figure 7.42b. This is not ideal for process quality, as this material build-up severely affects part precision. Secondly, the large velocity variations induces build platform vibration, which also negatively impacts part quality. This is critically important owing to the fact that serial robotic manipulators lack from the structural stiffness other kinematic architectures do have, such as Cartesian MEX-P AM machines, or traditional CNC machining centers.

Chapter 8

Conclusions, outputs and future research prospects

8.1 Main achievements of the present Thesis

The main objective of this Thesis was the conception, implementation and demonstration of a systematic path planning strategy for the automation of Robot-based Non-Planar Additive Manufacturing processes. This Section aims to analyze the degree of success of this main objective by compiling the main contributions developed in this Thesis:

- **Creation of procedurally generated axisymmetric embedded maps based on Riemannian geometry.** The first major contribution and the conceptual foundation of this Thesis is the embedded map generation procedure. This embedded map procedure is able to topologically transform any axisymmetric manifold into a sphere, and subsequently, into locally Euclidean tangent spaces. Any locally Euclidean tangent space is able to be planarly sliced, which leads to the next contribution.
- **Development of an axisymmetric slicing methodology for thin shells.** Following the embedded map generation procedure, the locally Euclidean tangent spaces are planarly sliceable. However, to encapsulate this planar slicing workflow, it is necessary to generate a set of mathematical and code tools that allow the transformation between spaces, and incorporate the planar slice. This is the second main contribution of this Thesis. This methodology is first constrained to the generation of thin shells, as the infilling (especially if it is hollow) is heavily conditioned by the geodesics of the build platforms.
- **Development of a non-planar, non-solid infilling strategy** to complement the previous achievement. Non-solid infilling is a tough challenge, and to solve this, a geodesic repulsion algorithm was conceived. This work provides one of the first generalized methods observed in the literature to implement non-solid, non-planar infills, and as such, it is the third main contribution of this Thesis.
- **Implementation of an experimental setup that serves as a demonstrator of**

the technology. To realize the trajectory generation on an actual RBNPAM station, an experimental setup was devised and deployed. This RBNPAM experimental platform allows for present and future research in the field of RBNPAM and other robot-based manufacturing process.

- **Development of calibration workflows for robot accuracy.** The fifth main contribution of this Thesis is the development of calibration procedures, both extrinsic and intrinsic. These calibration procedures allow for a more accurate material deposition in the case of an RBNPAM process, and will make this concept a high-Technology Readiness Level (TRL) technology in the future.
- **Realization of the first embedded map-based NPAM-MEX-P parts.** To conclude the present work, both the path planning strategy and the experimental setup were validated through the creation of NPAM-MEX-P parts, among the first in the literature. In this regard, there is still work to do to elevate the technology to higher TRL brackets, but this achievement demonstrates the potential of the technology.

8.2 Conclusions on the realization of Non-Planar Additive Manufacturing processes

This Thesis set out to prove that robot-based, non-planar material extrusion (RBNPAM-MEX-P) can be transformed from an appealing curiosity into a systematic and automated manufacturing process. Chapter 4 establishes a complete mathematical and computational foundation for robot-based non-planar slicing. By adopting cubic or sixth-degree Bézier spline systematic embedded maps, the Thesis shows that any axisymmetric or piecewise axisymmetric surface that is C^3-continuous can be represented in a single embedded map with an analytic metric tensor. That representation allows the exact Christoffel symbols and geodesic equations to be written in closed form, turning the search for shortest surface paths into a numerically stable two-point boundary-value problem.

The Chapter further derives explicit expressions for the area element, enabling direct control of layer height and raster width without resorting to planar approximations or empirical corrections. A tessellation-cleansing strategy eliminates non-manifold artifacts and generates ready-to-slice meshes, while trajectory interpolation aims towards the goal that robot motions remain smooth and dynamically feasible. Finally, the Chapter validates the framework on a cylindrical helix, demonstrating that the generated trajectories preserve curvature constraints, maintain constant offset from the true surface, and map unambiguously to Cartesian robot commands. Collectively, these results prove that embedded-map slicing provides both the theoretical rigor and the numerical efficiency required for practical non-planar material-extrusion processes.

Chapter 5 extends the embedded-map framework from surface description to volumetric deposition by introducing a method for generating quasi-uniform, conformal infills on curved layers. It formulates point distribution as the minimization of a geodesic-repulsion energy in the manifold's embedded map; substituting Euclidean distances with exact geodesic lengths preserves uniformity even on strongly curved or concave regions. The Chapter links the required number of mesh points to commanded density, raster width, and local curvature through an analytical scaling law, allowing infill density to be specified independently of layer geometry.

Several numerical refinements (explicit chord-based force derivatives, energy and gradient normalization, and adaptive seeding strategies) reduce computational cost by more than an order of magnitude without sacrificing mesh quality. A simple nearest-neighbor plus 2-opt heuristic then converts the optimized point sets into collision-free Hamiltonian paths whose spacing reproduces the target areal density to within five percent.

Regarding result validation, kernel-density and nearest-neighbor histograms confirm that interior, boundary, and vertex regions converge to the predicted uniformity limits, while case studies on convex barrel and concave hourglass platforms verify robustness across monotonic and non-monotonic curvature. In combination, these contributions demonstrate that homogeneous-density, non-solid infills can be planned directly on complex manifolds with predictable material usage and without relying on planar projections.

Chapter 6 aims at translating the theoretical path-planning advances into practice through the design and implementation of manufacturing hardware. A detailed kinematic error model of the UR10 shows how joint-level uncertainties propagate into task-space ellipsoids, enabling quantitative placement of fixtures at regions of minimal positional spread. Guided by that analysis, a robot-based NPAM-MEX-P manufacturing cell was devised. This cell includes a modular cylindrical build platform, a calibration and extrusion fixture, and all necessary power supply and control equipment.

Regarding the build platform, finite-element thermal studies validated a 50 [W] cartridge-heater, nylon insulator and conical coupling that reach target temperature within three minutes while thermally isolating the robot flange. Custom ESP32 controllers, programmed in C/C++, govern both the platform and a extruder through closed-loop PID, timed-relay modulation and ROS2-compatible serial messaging. All high- and low-voltage elements were consolidated into a DIN-rail panel, which provides safe power distribution and organized wiring. A rigid inverted-L fixture combines an extruder, laser metrology and a calibration totem, providing repeatable mechanical registration without hindering robot reach. The fully integrated cell, comprising thermal hardware, electronics, fixturing and power infrastructure, serve to establishing a reliable physical platform for robot-based non-planar material-extrusion experiments.

Chapter 7 translates the theoretical and hardware developments of the preceding chapters into an integrated manufacturing workflow that executes robot-based, non-planar material-extrusion builds repeatably and with minimal operator intervention. It first decomposes the entire process (equipment warm-up, calibration, trajectory execution, and synchronized extrusion) into a sequence of offline and online tasks coordinated by a ROS2 supervisor. The extrinsic calibration routine aligns the robot base with the stationary laser and table once per fixture installation, whereas the intrinsic calibration, triggered automatically before every build, brings the heated platform into micrometer-level coplanarity with the nozzle in a short time. Both procedures are embedded in the same control architecture that also drives the ESP32-based temperature and feed-rate regulators, enabling fully scripted start-to-finish operation.

Additionally, the workflow is demonstrated on increasingly demanding case studies. A shark-scale hollow infill deposited on a convex barrel validates simultaneous extrusion and trajectory execution of non-planar, non-solid infills. A concave hourglass part confirms that the same procedure accommodates negative curvature without loss of accuracy. Finally, two iterations of a cylindrical spring expose and eliminate practical failure modes: inadequate bed temperature, non-overlapping coil junctions, and velocity-induced material build-up. Upgrading the motion primitive from the robot's default Cartesian interpolator to continuous joint-space streaming raises effective deposition speed by an order of magnitude while maintaining smooth platform motion. Throughout all trials the cell maintains positional errors below the envelope established in calibration tests, proving that the proposed workflow can reliably convert embedded-map trajectories into tangible parts on standard collaborative-robot hardware.

Together, these results confirm that the proposed workflow bridges advanced geometric planning and shop-floor reality, turning robot-based non-planar extrusion from concept into a dependable, repeatable manufacturing operation.

8.3 Limitations found and future research prospects

Many improvement points have been found during the development of the present Thesis, and have been pinpointed throughout. In this Section, a compilation of these improvement points is presented, as well as some preliminary solutions that will be implemented shortly from the time of writing this Document.

Embedded map bijectivity is a paramount issue that needs to be addressed. If the part does not lie within the same bijectivity region of the embedded map, chances are it is not manufacturable. As of the moment of writing this Thesis, the bijectivity region has been delimited by the platform's evolute, a curve that collects information about the platform curvature radius. This is a good approach when the platform is a curve comprised of only one segment. However, when the curve is composite, this approach is insufficient.

An option to solve this issue has been found in Pythagorean hodograph curves, as introduced in Chapter 4. With these curves, two problems could be solved simultaneously. Firstly, the null space of simultaneous offsets could be analytically solved. This null space is the inverse of the bijectivity region, effectively delimiting it. Additionally, these curves may provide connections between bijectivity regions, in the same way these curves are used to provide rounded, smooth paths in CNC machining.

Regarding non-solid infills, there is still work to do. The presented algorithm supposes the embedding of each slice is provided. This is a sufficiently good approach in conventional NPAM, where slice height uniformity is disregarded in favor of conformality. However, providing an embedding for each slice is cumbersome, and contrary to the automation philosophy of this Thesis.

To solve this, two different approaches are thought to be of interest. The first approach involves the previously mentioned Pythagorean hodograph curves. These curves provide analytical solutions for offset curves given a generatrix, which is ideal in the presented workflow. If the previous point is realized, the application of this method would be straightforward. The second approach is more general, and involves establishing the geodesic repulsion algorithm on the full 3D embedding $\{u, \theta, h\}$. This approach, combined with tetrahedron Triply Periodic Minimal Surfaces, may fill the part volume with conformal, uniformly sliced, non-solid infills.

Another point of improvement revolves around the current functionalities available in the in-house slicer presented in this Thesis. The slicer currently offers the minimum functionality requirements for realizing an NPAM MEX-P process, such as slicing parts, providing wall and infill rasters, and setting extrusion parameters for the build platform and extrusion unit. However, some more key functionalities are to be implemented in the workflow.

These functionalities include:

- Filament retractions, at polygon changes or slice shifts.
- z-hops at layer changes.
- Support generation.
- Multi-material, multi-nozzle functioning modes.
- Station configuration modification to allow for moving-extruder architectures.
- Online data acquisition and part inspection.

Regarding calibration, the calibration method proposed is an effective method, but it could be improved to account for local geometry imperfections. To do so, a collaboration with CAR-CSIC-UPM is currently under way, through projects PP24 and CPP2023, mentioned in the Introduction. It is thought that with extensive point cloud information, a complete extrinsic-intrinsic calibration of station tooling could be performed. This calibration would be based on the resolution of the Maximum Clique Problem, thanks to the expertise of the CAR-CSIC-UPM members participating in the projects.

Lastly, all presented methods are thought to be applicable in other type of embeddings. These embeddings include the introduction of Bézier surfaces, or more generally, parametric surfaces. This implementation would require a deeper understanding of surgery theory in the context of topology, to be able to create coherent rasters traveling from one manifold to another. This research line is currently under development at the time of writing this Thesis.

8.4 Summary of scientific-technological contributions made in the present Thesis

Works in scientific journals

- Guzman-Bautista, A.; López-Arrabal, A.; Sanchez-Oro-Aguado, E.; Fernández Gorgojo, A.; García-Galán, R.; Badesa, F.J.; Vizan-Idoipe, A. Quasi-Uniform Density Non-Solid Infill Strategy for Axisymmetric Non-Planar Additive Manufacturing. Applied Sciences. 2025, 15, Art. no. 5899. https://doi.org/10.3390/app15115899

- López-Arrabal, A.; Guzmán-Bautista, A.; Solórzano-Requejo, W.; Franco-Martínez, F.; Villaverde, M.; Axisymmetric non-planar slicing and path planning strategy for robot-based additive manufacturing. Materials & Design, 2024, 241, Art. no. 112915. https://doi.org/10.1016/j.matdes.2024.112915

Works in conference proceedings

- Á. Guzmán-Bautista, A. López Arrabal, E. Chacón-Tanarro, B. Pérez-Hickman las Matas, “Metodología de calibración de estación de mecanizado robótico basada en flujo de trabajo CNC,” 23er Congreso Nacional de Ingeniería Mecánica, Anales de la Ingeniería Mecánica, 2023, 1, p. 149, ISSN: 0212-5072
- A. Guzman-Bautista, L. Gil-Villacastin, A. Lopez-Arrabal, E. Chacón-Tanarro, J. M. Muñoz-Guijosa and A. Vizán-Idoipe, “Automatic State Space-Based Dynamic Characterization of Industrial Machining Robots,” 2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR), Barcelona, Spain, 2025, pp. 45-49, doi: 10.1109/ICMERR64601.2025.10949941.
- L. Gil-Villacastin, A. Guzman-Bautista, A. Lopez-Arrabal, E. Chacon-Tanarro, A. Sancho-Arellano and M. Lerin-Alonso, “Design for Robot-Based Non-Planar Additive Manufacturing Case Study: Soft Robotics Gripper,” 2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR), Barcelona, Spain, 2025, pp. 5-9, doi: 10.1109/ICMERR64601.2025.10949917.
- A. López-Arrabal, Á. Guzmán-Bautista, W. Solórzano-Requejo, A. Sancho-Arellano, F. Franco-Martínez and A. D. Lantada, “Path Planning Design for Robot Based Non-Planar Additive Manufacturing Case Study: Coronary Stent,” 2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR), Barcelona, Spain, 2025, pp. 17-21, doi: 10.1109/ICMERR64601.2025.10949923.
- Á. Guzmán-Bautista, A. López-Arrabal, E. Sánchez-Oro-Aguado, A. Fernández-Gorgojo, R. García-Galán, E. Chacón-Tanarro, F. J. Badesa-Clemente, “Rellenos de densidad pseudo-constante para Fabricación Aditiva No Plana,” 25o Congreso Nacional de Ingeniería Mecánica, 2025, 1, proceedings pending.

Main Bachelor’s and Master’s Thesis related to the Thesis

- Direction of the Bachelor’s Thesis of Ms. Beatriz Pérez-Hickman las Matas, titled: “Desarrollo y validación de un flujo de trabajo en una estación de mecanizado robótico.”
- Co-direction with Prof. Adrián López Arrabal and Prof. Miguel Clavijo Jiménez of the Bachelor’s Thesis of Mr. Mario García Martínez, titled: “Caracterización de la incertidumbre cinemática de robot colaborativo por propagación de error en las articulaciones.”
- Co-direction with Prof. Juan de Juanes Márquez Sevillano of the Bachelor’s Thesis of Mr. Antonio de Miguel Pérez, titled: “Desarrollo de una aplicación de realidad mixta para el ámbito de la fabricación industrial.”

- Co-direction with Prof. Adrián López Arrabal of the Bachelor's Thesis of Mr. Alejandro Franco Gutiérrez, titled: "Diseño de una metodología de laminado de sólidos 3D para generación de trayectorias no planas en Fabricación Aditiva Robótica."
- Co-direction with Prof. Miguel Clavijo Jiménez of the Master's Thesis of Ms. Lu Chen, titled: "Exploración de ROS como plataforma para la integración en Fabricación Robotizada."
- Co-direction with Prof. Adrián López Arrabal of the Master's Thesis of Ms. Amparo Sancho Arellano, titled: "Generación y validación de trayectorias no planares para Fabricación Aditiva Robotizada."
- Co-direction with Prof. Daniel Galán Vicente of the Master's Thesis of Mr. Miguel Lerín Alonso, titled: "Diseño, implementación y validación de arquitectura en ROS2 para estación robotizada de Fabricación Aditiva."
- Co-direction with Prof. Enrique Chacón Tanarro of the Bachelor's Thesis of Ms. Irene Rodríguez Gramún, titled: "Diseño, programación y prototipo de regulador de temperatura para estación robotizada de Fabricación Aditiva."
- Co-direction with Prof. Enrique Chacón Tanarro and Prof. Adrián López Arrabal of the Bachelor's Thesis of Mr. Adrián Martínez García, titled: "Diseño y fabricación de cabezal de extrusión para estación robotizada de Fabricación Aditiva No Plana."
- Co-direction with Prof. Juan Manuel Muñoz Guijosa and Prof. Adrián López Arrabal of the Master's Thesis of Ms. Laura Gil Villacastín, titled: "Caracterización dinámica automatizada de manipuladores robóticos basada en espacio de estados para mecanizado."
- Co-direction with Prof. Enrique Chacón Tanarro and Prof. Adrián López Arrabal of the Master's Thesis of Ms. María del Rosario Martínez Verdoy, titled: "Diseño para la Fabricación Aditiva No Plana: mangueta de vehículo de competición para FSAE."
- Co-direction with Prof. Daniel Galán Vicente of the Master's Thesis of Ms. Adela Jiménez Hervás, titled: "Integración de la arquitectura en ROS2 de una célula robotizada para fabricación aditiva."
- Co-direction with Prof. Andrea Fernández Gorgojo of the Master's Thesis of Mr. Elio Sánchez-Oro Aguado, titled: "Estrategia de relleno cuasi uniforme para fabricación aditiva no planar."
- Co-direction with Prof. Andrea Fernández Gorgojo of the Bachelor's Thesis of Ms. Virginia Padrós Rodero, titled: "Laminado NPAM de geometrías libres mediante mapas encajados generados sistemáticamente." (work in progress at the time of writing)
- Co-direction with Prof. Andrea Fernández Gorgojo of the Bachelor's Thesis of Ms. Alicia Zamorano Gándara, titled: "Cálculo de la biyectividad de mapas encajados para NPAM mediante métodos exactos." (work in progress at the time of writing)
- Co-direction with Prof. Andrea Fernández Gorgojo, Prof. Francisco Javier Badesa Clemente, and Prof. Santiago Martínez de la Casa of the Master's Thesis of Ms. Carolina Yasmin Tank Vela, titled: "Flujo de calibración robótica absoluta mediante escáner 3D."

(work in progress at the time of writing)

- Future co-direction with Prof. Andrea Fernández Gorgojo and Prof. Francisco Javier Badesa Clemente of the Doctoral Thesis of Ms. Carolina Yasmin Tank Vela.

Other Bachelor's and Master's Thesis related to the Thesis

- Co-direction with Prof. Antonio Barrientos Cruz and Prof. Adrián López Arrabal of the Bachelor's Thesis of Ms. Victoria Gonçalves da Corte, titled: "Robots Origami. Diseño, construcción y aplicaciones."
- Direction of the Bachelor's Thesis of Ms. María Méndez Peraza, titled: "Desarrollo del postprocesador de una máquina herramienta con cinemática híbrida paralela y su interconexión con un programa CAM."
- Co-direction with Prof. Enrique Chacón Tanarro of the Bachelor's Thesis of Mr. Pedro Sánchez Granda, titled: "Metodología de ensayos para benchmarking de procesos de Fabricación Aditiva por FDM."
- Co-direction with Prof. Francisco Franco Martínez and Prof. Adrián López Arrabal of the Bachelor's Thesis of Mr. Víctor Bueno Ruano, titled: "Desarrollo de interfaz gráfica y algoritmos de relleno para slicer de Fabricación Aditiva No Plana."
- Co-direction with Prof. Enrique Chacón Tanarro of the Bachelor's Thesis of Mr. Jorge Lecocq Rael, titled: "Aplicación de Machine Learning en la predicción de dimensiones y tolerancias de piezas MEX-P."

Research projects related to the Thesis

- "Calibración de posicionamiento absoluta de un robot por reconocimiento de patrones geométricos mediante algoritmos de resolución del problema de máximo clique" (ETSII-UPM24-PM02). Competitive grant call "Primeros Proyectos ETSII-UPM 2024". Participation as Principal Investigator.
- "Predicción automatizada del comportamiento dinámico de un brazo robot para corrección de trayectorias en la realización de tareas de alta precisión". Competitive grant call "Primeros Proyectos ETSII-UPM 2024". Participation as research team member.
- "CYBERCELL: Sistema de producción ciberfísico para la planificación inteligente de células de fabricación" (PID2021-124838OB-I00). Competitive grant call "Plan Estatal de Investigación Científica y Técnica y de Innovación 2021-2023". Participation as research team member.
- "Reducción del coste de producción de piezas actuales y futuras basado en nuevos materiales a partir de composites" (CPP2023-010651). Competitive grant call "Proyectos en colaboración público-privada 2023". Participation as research team member.

8.5 Educational contribution of this Thesis

This Thesis has also served to create a Master's course in the Industrial Engineering Master's Degree, offered at the Escuela Técnica Superior de Ingenieros Industriales (UPM). The course, named "Advanced Manufacturing Systems" (AMS), is available as one of the options for the Ingenia macro-course (12 ECTS). In this course, students are instructed to develop and operate robotized manufacturing cells, be it automatic inspection stations, robotized machining cells, or RBNPAM cells.

Ingenia is an initiative based on the fundamentals of CDIO, shorthand for Conceive, Design, Implement, and Operate. CDIO is an international framework created by faculty at the Massachusetts Institute of Technology and three Swedish universities to reform engineering education. It keeps solid disciplinary fundamentals but embeds them in the full product-system life-cycle, giving students repeated opportunities to conceive, design, implement and operate real systems in authentic team settings. CDIO turns an engineering program into a miniature version of professional practice. Students graduate as systems thinkers who can design something that works and justify why it works; faculty teach in more engaging, research-informed ways; accreditation evidence all but writes itself; and employers see graduates who can contribute on day one. These benefits explain why the framework has spread worldwide and why many educators use CDIO as the "backbone" onto which they graft emerging approaches such as challenge-based or sustainability-oriented learning.

AMS is a course currently taught by 7 instructors, including the present author, with a broad expertise on robotized systems, manufacturing processes, industrial automation, electric and electronic design, and mechanical characterization of materials, as well as soft skills such as leadership, teamwork in professional settings, and agile workflows. This Thesis serves as the cornerstone for the Further project developed in AMS and as a basis of documentation for current and future students.

Bibliography

[1] *Additive manufacturing - General principles - Fundamentals and vocabulary - ISO/ASTM 52900:2021*, 2021.

[2] M. N. M. Azlin, R. A. Ilyas, M. Y. M. Zuhri, S. M. Sapuan, and M. M. Harussani, “3D Printing and Shaping Polymers, Composites, and Nanocomposites: A Review”, *Polymers*, vol. 14, no. 180, 2022. DOI: `https://doi.org/3390/polym14010180`.

[3] D. Bi, M. Duan, T. Y. Lau, F. Xie, and K. Tang, “Strength-enhanced volume decomposition for multi-directional additive manufacturing”, *Additive Manufacturing*, vol. 69, no. March, p. 103 529, 2023. DOI: `10.1016/j.addma.2023.103529`.

[4] X. Li, W. Liu, Z. Hu, *et al.*, “Supportless 3D-printing of non-planar thin-walled structures with the multi-axis screw-extrusion additive manufacturing system”, *Materials and Design*, vol. 240, no. March, p. 112 860, 2024. DOI: `10.1016/j.matdes.2024.112860`.

[5] C. Rodriguez-Padilla, E. Cuan-Urquizo, A. Roman-Flores, J. L. Gordillo, and C. Vázquez-Hurtado, “Algorithm for the conformal 3d printing on non-planar tessellated surfaces: Applicability in patterns and lattices”, *Applied Sciences (Switzerland)*, vol. 11, no. 16, 2021. DOI: `10.3390/app11167509`.

[6] J. Etienne, N. Ray, D. Panozzo, *et al.*, “Curvislicer: Slightly curved slicing for 3-axis printers”, *ACM Transactions on Graphics*, vol. 38, no. 4, 2019. DOI: `10.1145/3306346.3323022`.

[7] L. Pelzer and C. Hopmann, “Additive manufacturing of non-planar layers with variable layer height”, *Additive Manufacturing*, vol. 37, p. 101 697, 2021. DOI: `https://doi.org/10.1016/j.addma.2020.101697`.

[8] P. M. Bhatt, R. K. Malhan, A. V. Shembekar, Y. J. Yoon, and S. K. Gupta, “Expanding capabilities of additive manufacturing through use of robotics technologies: A survey”, *Additive Manufacturing*, vol. 31, p. 100 933, 2020. DOI: `https://doi.org/10.1016/j.addma.2019.100933`.

[9] M. K. Thompson, G. Moroni, T. Vaneker, *et al.*, “Design for additive manufacturing: Trends, opportunities, considerations, and constraints”, *CIRP Annals*, vol. 65, no. 2, pp. 737–760, 2016. DOI: `https://doi.org/10.1016/j.cirp.2016.05.004`.

[10] K. Altaf, A. Majdi Abdul Rani, and V. R. Raghavan, “Prototype production and experimental analysis for circular and profiled conformal cooling channels in aluminium filled epoxy injection mould tools”, *Rapid Prototyping Journal*, vol. 19, no. 4, pp. 220–229, 2013. DOI: `10.1108/13552541311323236`.

[11] J. Plocher and A. Panesar, "Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures", *Materials & Design*, vol. 183, p. 108 164, 2019. DOI: `https://doi.org/10.1016/j.matdes.2019.108164`.

[12] L. M. Ballesteros, D. Cano, C. A. García, *et al.*, "Comparison of textured nylon surfaces manufactured by cnc micromachining and 3d printing", *Surface Topography: Metrology and Properties*, vol. 12, no. 1, p. 015 016, 2024. DOI: `10.1088/2051-672X/ad2af3`.

[13] A. Nazir, O. Gokcekaya, K. Md Masum Billah, *et al.*, "Multi-material additive manufacturing: A systematic review of design, properties, applications, challenges, and 3d printing of materials and cellular metamaterials", *Materials & Design*, vol. 226, p. 111 661, 2023. DOI: `https://doi.org/10.1016/j.matdes.2023.111661`.

[14] G. Stano and G. Percoco, "Additive manufacturing aimed to soft robots fabrication: A review", *Extreme Mechanics Letters*, vol. 42, p. 101 079, 2021. DOI: `10.1016/j.eml.2020.101079`.

[15] C. Li, B. Wu, Z. Zhang, and Y. Zhang, "A novel process planning method of 3 + 2-axis additive manufacturing for aero-engine blade based on machine learning", *Journal of Intelligent Manufacturing*, vol. 34, no. 4, pp. 2027–2042, 2023. DOI: `10.1007/s10845-021-01898-6`.

[16] C.-Y. Liao, Y.-W. Tseng, S.-D. Wu, L.-Y. Kang, N.-T. Dai, and S. hui Hsu, "Non-planar bioprinting with molding assistance for irregular wound shape", *Journal of Materials Research and Technology*, vol. 26, pp. 1596–1608, 2023. DOI: `https://doi.org/10.1016/j.jmrt.2023.07.279`.

[17] J.-F. Chauvette, I. L. Hia, J. Pierre, *et al.*, "Non-planar multiprocess additive manufacturing of multifunctional composites", *Advanced Materials Technologies*, vol. 8, no. 17, p. 2 300 399, 2023. DOI: `https://doi.org/10.1002/admt.202300399`. eprint: `https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/admt.202300399`.

[18] J. Jiang, S. T. Newman, and R. Y. Z. and, "A review of multiple degrees of freedom for additive manufacturing machines", *International Journal of Computer Integrated Manufacturing*, vol. 34, no. 2, pp. 195–211, 2021. DOI: `10.1080/0951192X.2020.1858510`. eprint: `https://doi.org/10.1080/0951192X.2020.1858510`.

[19] A. M. Cendrero, G. M. Fortunato, J. M. Munoz-Guijosa, C. De Maria, and A. D. Lantada, "Benefits of non-planar printing strategies towards eco-efficient 3d printing", *Sustainability (Switzerland)*, vol. 13, no. 4, pp. 1–17, 2021. DOI: `10.3390/su13041599`.

[20] B. Shen, J. Zhang, M. Cheng, S. Guo, and R. He, "Supply chain integration in mass customization", *Annals of Operations Research*, vol. 344, no. 2, pp. 595–616, 2025. DOI: `10.1007/s10479-023-05202-y`.

[21] F. Xie, L. Chen, Z. Li, and K. Tang, "Path smoothing and feed rate planning for robotic curved layer additive manufacturing", *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101 967, 2020. DOI: `https://doi.org/10.1016/j.rcim.2020.101967`.

[22] T. Lehmann, A. Jain, Y. Jain, *et al.*, "Concurrent geometry- and material-based process identification and optimization for robotic cmt-based wire arc additive manufacturing", *Materials & Design*, vol. 194, p. 108 841, 2020. DOI: `https://doi.org/10.1016/j.matdes.2020.108841`.

[23] J. Lettori, R. Raffaeli, P. Bilancia, M. Peruzzini, and M. Pellicciari, "A review of geometry representation and processing methods for cartesian and multiaxial robot-

based additive manufacturing", *The International Journal of Advanced Manufacturing Technology*, vol. 123, no. 11, pp. 3767–3794, 2022. DOI: 10.1007/s00170-022-10432-8.

[24] A. Nubiola and I. A. Bonev, "Absolute calibration of an abb irb 1600 robot using a laser tracker", *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 236–245, 2013. DOI: https://doi.org/10.1016/j.rcim.2012.06.004.

[25] M. Pagac, J. Hajnys, Q.-P. Ma, *et al.*, "A Review of Vat Photopolymerization Technology: Materials, Applications, Challenges, and Future Trends of 3D Printing", *Polymers*, vol. 13, no. 4, 2021. DOI: 10.3390/polym13040598.

[26] O. Gülcan, K. Günaydın, and A. Tamer, "The State of the Art of Material Jetting—A Critical Review", *Polymers*, vol. 13, no. 16, 2021. DOI: 10.3390/polym13162829.

[27] M. Ziaee and N. B. Crane, "Binder jetting: A review of process, materials, and methods", *Additive Manufacturing*, vol. 28, pp. 781–801, 2019. DOI: https://doi.org/10.1016/j.addma.2019.05.031.

[28] R. B. Kristiawan, F. Imaduddin, D. Ariawan, Ubaidillah, and Z. Arifin, "A review on the fused deposition modeling (FDM) 3D printing: Filament processing, materials, and printing parameters", *Open Engineering*, vol. 11, no. 1, pp. 639–649, 2021. DOI: doi:10.1515/eng-2021-0063.

[29] R. J. N. Joshua, S. A. Raj, M. T. Hameed Sultan, *et al.*, "Powder Bed Fusion 3D Printing in Precision Manufacturing for Biomedical Applications: A Comprehensive Review", *Materials*, vol. 17, no. 3, 2024. DOI: 10.3390/ma17030769.

[30] A. Vafadar, F. Guzzomi, A. Rassau, and K. Hayward, "Advances in Metal Additive Manufacturing: A Review of Common Processes, Industrial Applications, and Current Challenges", *Applied Sciences*, vol. 11, no. 3, 2021. DOI: 10.3390/app11031213.

[31] A. Dass and A. Moridi, "State of the Art in Directed Energy Deposition: From Additive Manufacturing to Materials Design", *Coatings*, vol. 9, no. 7, 2019. DOI: 10.3390/coatings9070418.

[32] D. Lin, S. Jin, F. Zhang, *et al.*, "3D stereolithography printing of graphene oxide reinforced complex architectures", *Nanotechnology*, vol. 26, 2015. DOI: 10.1088/0957-4484/26/43/434003.

[33] P. Turek, G. Budzik, J. Sęp, *et al.*, "An analysis of the casting polymer mold wear manufactured using polyjet method based on the measurement of the surface topography", *Polymers*, vol. 12, no. 12, 2020. DOI: 10.3390/polym12123029.

[34] Y. Zhou, Y. Tang, T. Hoff, M. Garon, and F. Y. Zhao, "The Verification of the Mechanical Properties of Binder Jetting Manufactured Parts by Instrumented Indentation Testing", *Procedia Manufacturing*, vol. 1, pp. 327–342, 2015, 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States. DOI: https://doi.org/10.1016/j.promfg.2015.09.038.

[35] L. Murr, "Open-cellular metal implant design and fabrication for biomechanical compatibility with bone using electron beam melting", *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 76, pp. 164–177, 2017, Structure-Property Relationships in Biological and Bioinspired Materials. DOI: https://doi.org/10.1016/j.jmbbm.2017.02.019.

[36] P. M. Bhatt, A. M. Kabir, M. Peralta, H. A. Bruck, and S. K. Gupta, "A robotic cell for performing sheet lamination-based additive manufacturing", *Additive Manufacturing*, vol. 27, pp. 278–289, 2019. DOI: https://doi.org/10.1016/j.addma.2019.02.002.

[37] F. Veiga, A. Gil Del Val, A. Suárez, and U. Alonso, "Analysis of the Machining Process of Titanium Ti6Al-4V Parts Manufactured by Wire Arc Additive Manufacturing (WAAM)", *Materials*, vol. 13, no. 3, 2020. DOI: 10.3390/ma13030766.

[38] D. Svetlizky, M. Das, B. Zheng, *et al.*, "Directed energy deposition (DED) additive manufacturing: Physical characteristics, defects, challenges and applications", *Materials Today*, vol. 49, pp. 271–295, 2021. DOI: https://doi.org/10.1016/j.mattod.2021.03.020.

[39] A. M. Peterson and D. O. Kazmer, "Predicting mechanical properties of material extrusion additive manufacturing-fabricated structures with limited information", *Scientific Reports*, vol. 12, no. 1, p. 14 736, 2022. DOI: 10.1038/s41598-022-19053-3.

[40] V. C. Boppana and F. Ali, "Improvement of tensile strength of fused deposition modelling (FDM) part using artificial neural network and genetic algorithm techniques", *International Journal of Industrial Engineering and Operations Management*, vol. 6, no. 2, pp. 117–142, 2024. DOI: 10.1108/IJIEOM-01-2023-0006.

[41] E. Kargar and A. Ghasemi-Ghalebahman, "Experimental investigation on fatigue life and tensile strength of carbon fiber-reinforced PLA composites based on fused deposition modeling", *Scientific Reports*, vol. 13, no. 1, p. 18 194, 2023. DOI: 10.1038/s41598-023-45046-x.

[42] H. Bakhtiari, M. Aamir, and M. Tolouei-Rad, "Effect of 3D Printing Parameters on the Fatigue Properties of Parts Manufactured by Fused Filament Fabrication: A Review", *Applied Sciences (Switzerland)*, vol. 13, no. 2, 2023. DOI: 10.3390/app13020904.

[43] H. Sadeghian, M. R. Ayatollahi, M. R. Khosravani, and N. Razavi, "From prototyping to functional parts: A review of mechanical reinforcing procedures for polymeric parts fabricated via material extrusion", *Journal of Manufacturing Processes*, vol. 130, pp. 1–34, 2024. DOI: https://doi.org/10.1016/j.jmapro.2024.08.058.

[44] F. W. C. Farias, T. J. G. dos Santos, and J. P. Oliveira, "Directed energy deposition + mechanical interlayer deformation additive manufacturing: a state-of-the-art literature review", *The International Journal of Advanced Manufacturing Technology*, vol. 131, no. 3, pp. 999–1038, 2024. DOI: 10.1007/s00170-024-13126-5.

[45] D. Gunasegaram, A. Barnard, M. Matthews, *et al.*, "Machine learning-assisted in-situ adaptive strategies for the control of defects and anomalies in metal additive manufacturing", *Additive Manufacturing*, vol. 81, p. 104 013, 2024. DOI: https://doi.org/10.1016/j.addma.2024.104013.

[46] K. M. V, R. R, P. K. J, and A. P. R, "Experimental analysis of the stringing problem in fdm printers", EAI, Jun. 2024. DOI: 10.4108/eai.23-2-2024.2346991.

[47] A. Montalti, G. Galiè, E. Pignatelli, and A. L. and, "Enhancing surface roughness of material extrusion additive manufacturing components via an innovative ironing process", *Virtual and Physical Prototyping*, vol. 19, no. 1, e2401929, 2024. DOI: 10.1080/17452759.2024.2401929. eprint: https://doi.org/10.1080/17452759.2024.2401929.

[48] M. Sgrulletti, M. Bragaglia, S. Giarnetti, L. Paleari, and F. Nanni, "Understanding the impact of Fused Filament Fabrication conditions on the microstructure and tensile properties of polyamide 6 by thermal and optical live monitoring", *Materials Today Communications*, vol. 28, p. 102 679, 2021. DOI: https://doi.org/10.1016/j.mtcomm.2021.102679.

[49] A. L. Vyatskikh, X. Wang, J. Haley, *et al.*, "Residual stress mitigation in directed energy deposition", *Materials Science and Engineering: A*, vol. 871, p. 144 845, 2023. DOI: `https://doi.org/10.1016/j.msea.2023.144845`.

[50] L. Nuñez, C. M. Downey, I. J. van Rooyen, I. Charit, and M. R. Maughan, "Analysis of surface roughness in metal directed energy deposition", *The International Journal of Advanced Manufacturing Technology*, 2024. DOI: `10.1007/s00170-024-13587-8`.

[51] B. Mo, T. Li, L. Deng, F. Shi, W. Liu, and H. Z. and, "Mechanisms and influencing factors of defect formations during laser-based directed energy deposition with coaxial powder feeding: a review", *Virtual and Physical Prototyping*, vol. 19, no. 1, e2404155, 2024. DOI: `10.1080/17452759.2024.2404155`. eprint: `https://doi.org/10.1080/17452759.2024.2404155`.

[52] A. Dolenc and I. Mäkelä, "Slicing procedures for layered manufacturing techniques", *Computer-Aided Design*, vol. 26, no. 2, pp. 119–126, 1994. DOI: `https://doi.org/10.1016/0010-4485(94)90032-9`.

[53] O. Diegel, S. Singamneni, B. Huang, and I. Gibson, "Curved Layer Fused Deposition Modeling in Conductive Polymer Additive Manufacturing", in *Advances in Mechanical Design*, ser. Advanced Materials Research, vol. 199, Trans Tech Publications Ltd, Apr. 2011, pp. 1984–1987. DOI: `10.4028/www.scientific.net/AMR.199-200.1984`.

[54] K. Xu, Y. Li, L. Chen, and K. Tang, "Curved layer based process planning for multi-axis volume printing of freeform parts", *Computer-Aided Design*, vol. 114, pp. 51–63, 2019. DOI: `https://doi.org/10.1016/j.cad.2019.05.007`.

[55] J. L. Pérez-Castillo, E. Cuan-Urquizo, A. Roman-Flores, *et al.*, "Curved layered fused filament fabrication: An overview", *Additive Manufacturing*, vol. 47, p. 102 354, 2021. DOI: `https://doi.org/10.1016/j.addma.2021.102354`.

[56] S. Mukka, A. T. Mathew, and R. K. Bhagchandani, "Optimization of curved layer fused deposition modeling process parameters to enhance surface finish", *Progress in Additive Manufacturing*, 2025. DOI: `10.1007/s40964-025-01193-3`.

[57] C. Dai, C. C. L. Wang, C. Wu, S. Lefebvre, G. Fang, and Y.-J. Liu, "Support-free volume printing by multi-axis motion", *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018. DOI: `10.1145/3197517.3201342`.

[58] G. Zhao, G. Ma, J. Feng, and W. Xiao, "Nonplanar slicing and path generation methods for robotic additive manufacturing", *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 9, pp. 3149–3159, 2018. DOI: `10.1007/s00170-018-1772-9`.

[59] H. Liu and F. Xing, "Non-planar helical path generation method for laser metal deposition of overhanging thin-walled structures", *Applied Sciences*, vol. 14, no. 15, 2024. DOI: `10.3390/app14156445`.

[60] M. Biegler, A. Marko, B. Graf, and M. Rethmeier, "Finite element analysis of in-situ distortion and bulging for an arbitrarily curved additive manufacturing directed energy deposition geometry", *Additive Manufacturing*, vol. 24, pp. 264–272, 2018. DOI: `https://doi.org/10.1016/j.addma.2018.10.006`.

[61] L. Squires, E. Roberts, and A. Bandyopadhyay, "Radial bimetallic structures via wire arc directed energy deposition-based additive manufacturing", *Nature Communications*, vol. 14, no. 1, p. 3544, 2023. DOI: `10.1038/s41467-023-39230-w`.

[62] X. Sun, M. Mazur, and C. T. Cheng, "A review of void reduction strategies in material extrusion-based additive manufacturing", *Addit. Manuf.*, vol. 67, 2023, Art. no. 103463. DOI: `10.1016/J.ADDMA.2023.103463`.

[63] G. Liu, W. Huang, Y. Wang, *et al.*, "Stress field-aware infill toolpath generation for additive manufacturing of continuous fiber reinforced polymer composites", *Mater. Des.*, vol. 239, 2024, Art. no. 112756. DOI: `10.1016/J.MATDES.2024.112756`.

[64] D. Wei, G. Zhu, Z. Shi, L. Gao, B. Sun, and J. Gao, "Isogeometric topology optimization for infill designs of porous structures with stress minimization in additive manufacturing", *Int. J. Numer. Methods Eng.*, vol. 125, no. 3, 2024, Art. no. e7391. DOI: `10.1002/NME.7391`.

[65] Z. Wang, C. Wang, F. Wu, Z. Liu, L. Liu, and D. Wang, "Design method of gyroid lattice structure based on the load paths direction and capacity", *Additive Manufacturing*, vol. 97, no. December 2024, p. 104 586, 2025. DOI: `10.1016/j.addma.2024.104586`.

[66] W. Ma, W. But, and P. He, "Nurbs-based adaptive slicing for efficient rapid prototyping", *Computer-Aided Design*, vol. 36, no. 13, pp. 1309–1325, 2004. DOI: `10.1016/j.cad.2004.02.001`.

[67] Y. Jin, Y. He, G. Fu, A. Zhang, and J. Du, "A non-retraction path planning approach for extrusion-based additive manufacturing", *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 132–144, 2017. DOI: `10.1016/J.RCIM.2017.03.008`.

[68] I. E. Yigit and I. Lazoglu, "Helical slicing method for material extrusion-based robotic additive manufacturing", *Progress in Additive Manufacturing*, vol. 4, pp. 225–232, 2019. DOI: `10.1007/s40964-019-00090-w`.

[69] I. E. Yigit and I. Lazoglu, "Spherical slicing method and its application on robotic additive manufacturing", *Progress in Additive Manufacturing*, vol. 5, pp. 387–394, 2020. DOI: `10.1007/s40964-020-00135-5`.

[70] N. Munasinghe and G. Paul, "Radial slicing for helical-shaped advanced manufacturing applications", *Int. J. Adv. Manuf. Technol.*, vol. 112, no. 3-4, pp. 1089–1100, 2021. DOI: `10.1007/s00170-020-05999-z`.

[71] Y. Shan, Y. Shui, J. Hua, and H. Mao, "Additive manufacturing of non-planar layers using isothermal surface slicing", *J. Manuf. Process.*, vol. 86, pp. 326–335, 2023. DOI: `10.1016/J.JMAPRO.2022.12.054`.

[72] D. Zhao and W. Guo, "Shape and performance controlled advanced design for additive manufacturing: A review of slicing and path planning", *J. Manuf. Sci. Eng.*, vol. 142, no. 1, 2020, Art. no. 010801. DOI: `10.1115/1.4045055`.

[73] G. M. Fortunato, M. Nicoletta, E. Batoni, G. Vozzi, and C. De Maria, "A fully automatic non-planar slicing algorithm for the additive manufacturing of complex geometries", *Additive Manufacturing*, vol. 69, no. December 2022, p. 103 541, 2023. DOI: `10.1016/j.addma.2023.103541`.

[74] J. Allum, J. Kitzinger, Y. Li, V. V. Silberschmidt, and A. Gleadall, "ZigZagZ: Improving mechanical performance in extrusion additive manufacturing by nonplanar toolpaths", *Additive Manufacturing*, vol. 38, no. November 2020, p. 101 715, 2021. DOI: `10.1016/j.addma.2020.101715`.

[75] X. Han, G. Wu, X. Liu, X. Song, and L. Cui, "Research on a support-free five-degree-of-freedom additive manufacturing method", *Micromachines*, vol. 15, no. 7, 2024, Art. no. 855. DOI: `10.3390/MI15070855`.

[76] D. Ding, Z. Pan, D. Cuiuri, H. Li, and N. Larkin, “Adaptive path planning for wire-feed additive manufacturing using medial axis transformation”, *J. Clean. Prod.*, vol. 133, pp. 942–952, 2016. DOI: 10.1016/J.JCLEPRO.2016.06.036.

[77] Y. Li, K. Tang, D. He, and X. Wang, “Multi-axis support-free printing of freeform parts with lattice infill structures”, *Comput. Aided Des.*, vol. 133, p. 102 986, 2021. DOI: 10.1016/J.CAD.2020.102986.

[78] P. Gupta, B. Krishnamoorthy, and G. Dreifus, “Continuous toolpath planning in a graphical framework for sparse infill additive manufacturing”, *Comput. Aided Des.*, vol. 127, 2020, Art. no. 102880. DOI: 10.1016/J.CAD.2020.102880.

[79] S. Lin, L. Xia, G. Ma, S. Zhou, and Y. M. Xie, “A maze-like path generation scheme for fused deposition modeling”, *Int. J. Adv. Manuf. Technol.*, vol. 104, no. 1-4, pp. 1509–1519, 2019. DOI: 10.1007/S00170-019-03986-7.

[80] L. Nguyen, J. Buhl, and M. Bambach, “Continuous eulerian tool path strategies for wire-arc additive manufacturing of rib-web structures with machine-learning-based adaptive void filling”, *Addit. Manuf.*, vol. 35, 2020, Art. no. 101265. DOI: 10.1016/J.ADDMA.2020.101265.

[81] F. Michel, H. Lockett, J. Ding, F. Martina, G. Marinelli, and S. Williams, “A modular path planning solution for wire + arc additive manufacturing”, *Robot. Comput.-Integr. Manuf.*, vol. 60, pp. 1–11, 2019. DOI: 10.1016/J.RCIM.2019.05.009.

[82] S. H. Oh, C. H. An, B. Seo, J. Kim, C. Y. Park, and K. Park, “Functional morphology change of tpms structures for design and additive manufacturing of compact heat exchangers”, *Addit. Manuf.*, vol. 76, 2023, Art. no. 103778. DOI: 10.1016/J.ADDMA.2023.103778.

[83] Y. Wakjira, A. Cioni, and H. G. Lemu, “Current status of the application of additive-manufactured TPMS structure in bone tissue engineering”, *Prog. Addit. Manuf.*, vol. 10, no. 2, pp. 1085–1102, 2024. DOI: 10.1007/S40964-024-00714-W.

[84] A. Shaikh, A. Saxena, J. Griffis, K. Shahed, and G. Manogharan, “Functionally graded TPMS gyroid structures for additive manufacturing of non-pneumatic tires”, *Mater. Sci. Addit. Manuf.*, vol. 3, no. 4, 2024, Art. no. 5022. DOI: 10.36922/MSAM.5022.

[85] F. Franco-Martínez, W. Solórzano-Requejo, A. de Blas-de Miguel, *et al.*, “Design and manufacturing of microtextured patient-specific coronary stent”, in *Proc. 16th Int. Jt. Conf. Biomed. Eng. Syst. Technol. (BIOSTEC 2023) - BIODEVICES*, 2023, pp. 142–149. DOI: 10.5220/0011691600003414.

[86] P. Urhal, A. Weightman, C. Diver, and P. Bartolo, “Robot assisted additive manufacturing : A review”, *Robotics and Computer Integrated Manufacturing*, vol. 59, no. July 2018, pp. 335–345, 2019. DOI: 10.1016/j.rcim.2019.05.005.

[87] G. C. Zutin, E. C. Pulquerio, A. V. Pasotti, G. F. Barbosa, and S. B. Shiki, “Application of robotic manipulator technology and its relation to additive manufacturing process — a review”, *Int. J. Adv. Manuf. Technol.*, vol. 133, no. 1-2, pp. 257–271, 2024. DOI: 10.1007/S00170-024-13710-9/FIGURES/10.

[88] P. Tang, X. Zhao, H. Shi, *et al.*, “A review of multi-axis additive manufacturing : Potential , opportunity and challenge”, *Additive Manufacturing*, vol. 83, no. March, 2024. DOI: 10.1016/j.addma.2024.104075.

[89] Y. Yao, L. Cheng, and Z. Li, “A comparative review of multi-axis 3D printing”, *J. Manuf. Process.*, vol. 120, pp. 1002–1022, 2024. DOI: 10.1016/J.JMAPRO.2024.04.084.

[90] P. Štefčák, I. Gajdoš, J. Slota, J. Varga, Z. Kimáková, and M. Vrabeľ, "Determination of design limitations of curved profiles manufactured by robotics non-planar additive manufacturing", *Adv. Sci. Technol. Res. J.*, vol. 18, no. 3, pp. 92–98, 2024. DOI: 10.12913/22998624/185960.

[91] P. Nayyeri, K. Zareinia, and H. Bougherara, "Planar and nonplanar slicing algorithms for fused deposition modeling technology: A critical review", *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 5-6, pp. 2785–2810, 2022. DOI: 10.1007/s00170-021-08347-x.

[92] S. J. Keating, J. C. Leland, L. Cai, and N. Oxman, "Toward site-specific and self-sufficient robotic fabrication on architectural scales", *Science Robotics*, vol. 2, no. 5, eaam8986, 2017. DOI: 10.1126/scirobotics.aam8986. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.aam8986.

[93] I. Ishak and P. Larochelle, "Robot arm platform for additive manufacturing: 3d lattice structures", in *30th Florida conference on recent advances in robotics*, 2017, pp. 11–12.

[94] I. Bin Ishak, J. Fisher, and P. Larochelle, "Robot arm platform for additive manufacturing using multi-plane toolpaths", in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 50152, 2016, V05AT07A063.

[95] C. Wu, C. Dai, G. Fang, Y.-J. Liu, and C. C. Wang, "RoboFDM: A robotic system for support-free fabrication using FDM", in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1175–1180. DOI: 10.1109/ICRA.2017.7989140.

[96] M. D. M. Kutzer and L. D. DeVries, "Testbed for Multilayer Conformal Additive Manufacturing", *Technologies*, vol. 5, no. 2, 2017. DOI: 10.3390/technologies5020025.

[97] A. G. Stevens, C. R. Oliver, M. Kirchmeyer, *et al.*, "Conformal Robotic Stereolithography", *3D Printing and Additive Manufacturing*, vol. 3, no. 4, pp. 226–235, 2016, PMID: 29577062. DOI: 10.1089/3dp.2016.0042. eprint: https://doi.org/10.1089/3dp.2016.0042.

[98] "Relativity space". Launching the future of humanity in space, Relativity Space, Inc. (2025), [Online]. Available: https://www.relativityspace.com/ (visited on 06/21/2025).

[99] "Branch technology — freeform 3d printed building facades". Patented construction-scale 3D-printing technology, Branch Technology. (2025), [Online]. Available: https://branchtechnology.com/ (visited on 06/21/2025).

[100] "Mx3d | robotic 3d metal printing solutions". Wire Arc Additive Manufacturing (WAAM) systems and services, MX3D. (2025), [Online]. Available: https://mx3d.com/ (visited on 06/21/2025).

[101] G. Carabin, L. Scalera, T. Wongratanaphisan, and R. Vidoni, "An energy-efficient approach for 3D printing with a Linear Delta Robot equipped with optimal springs", *Robotics and Computer-Integrated Manufacturing*, vol. 67, no. July 2020, p. 102 045, 2021. DOI: 10.1016/j.rcim.2020.102045.

[102] E. S. Barjuei, J. Shin, K. Kim, and J. Lee, "Precision improvement of robotic bioprinting via vision-based tool path compensation", *Scientific Reports*, vol. 14, no. 1, p. 17 764, 2024. DOI: 10.1038/s41598-024-68597-z.

[103] W. Ma, T. Hu, C. Zhang, and Q. Chen, "Adaptive remanufacturing for freeform surface parts based on linear laser scanner and robotic laser cladding", *Robotics and*

Computer-Integrated Manufacturing, vol. 91, p. 102 855, 2025. DOI: https://doi.org/10.1016/j.rcim.2024.102855.

[104] *Robotics — Vocabulary ISO 8373:2021*, 2021.

[105] A. López-Arrabal, Á. Guzmán-Bautista, W. Solórzano-Requejo, F. Franco-Martínez, and M. Villaverde, "Axisymmetric non-planar slicing and path planning strategy for robot-based additive manufacturing", *Materials and Design*, vol. 241, no. March, 2024. DOI: 10.1016/j.matdes.2024.112915.

[106] *Automation systems and integration - Numerical control of machines - Program format and definitions of address words, ISO 6983-1:2009*, 2009.

[107] A. Guzman-Bautista, A. López-Arrabal, E. Sanchez-Oro-Aguado, *et al.*, "Quasi-uniform density non-solid infill strategy for axisymmetric non-planar additive manufacturing", *Applied Sciences*, vol. 15, no. 11, 2025. DOI: 10.3390/app15115899.

[108] M. Jiang, Y. Zhou, R. Wang, R. Southern, and J. J. Zhang, "Blue noise sampling using an SPH-based method", *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–11, 2015. DOI: 10.1145/2816795.2818102.

[109] H. Imai, M. Iri, and K. Murota, "Voronoi diagram in the laguerre geometry and its applications", *SIAM J. Comput.*, vol. 14, no. 1, pp. 93–105, 2006. DOI: 10.1137/0214006.

[110] G. A. Croes, "A method for solving Traveling-Salesman problems", *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958. DOI: 10.1287/OPRE.6.6.791.

[111] J. Yang, Q. Qi, A. Adili, and H. Ding, "An analytical tool path smoothing algorithm for robotic machining with the consideration of redundant kinematics", *Robotics and Computer-Integrated Manufacturing*, vol. 89, no. April, 2024. DOI: 10.1016/j.rcim.2024.102768.

[112] F. Insero, V. Furlan, and H. Giberti, "Non-planar slicing for filled free-form geometries in robot-based FDM", *Journal of Intelligent Manufacturing*, vol. 36, no. 2, pp. 833–851, 2023. DOI: 10.1007/s10845-023-02250-w.

[113] R. T. Farouki and T. Sakkalis, "Pythagorean hodographs", *IBM Journal of Research and Development*, vol. 34, no. 5, pp. 736–752, 1990. DOI: 10.1147/rd.345.0736.

[114] R. T. Farouki, M. Knez, V. Vitrih, and E. Žagar, "On L2 approximation by planar Pythagorean-hodograph curves", *Mathematics and Computers in Simulation*, vol. 233, pp. 296–310, 2025. DOI: https://doi.org/10.1016/j.matcom.2025.02.001.

[115] C.-Y. Chen, S.-S. Shieh, M.-Y. Cheng, and K.-H. Su, "Vision-based Pythagorean hodograph spline command generation and adaptive disturbance compensation for planar contour tracking", *The International Journal of Advanced Manufacturing Technology*, vol. 65, no. 5, pp. 1185–1199, 2013. DOI: https://doi.org/10.1007/s00170-012-4250-9.

[116] X. Liang and T. Su, "Quintic pythagorean-hodograph curves based trajectory planning for delta robot with a prescribed geometrical constraint", *Applied Sciences*, vol. 9, no. 21, 2019. DOI: 10.3390/app9214491.

[117] A. Alves Neto, D. G. Macharet, and M. F. M. Campos, "On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles", *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, pp. 123–141, 2010. DOI: 10.1007/s10846-009-9365-3.

[118] T. Su, X. Liang, X. Zeng, and S. Liu, "Pythagorean-Hodograph curves-based trajectory planning for pick-and-place operation of Delta robot with prescribed pick and place heights", *Robotica*, vol. 41, no. 6, 1651–1672, 2023. DOI: 10.1017/S0263574722001898.

[119] O. Ibhadode, Z. Zhang, J. Sixt, *et al.*, "Topology optimization for metal additive manufacturing: Current trends, challenges, and future outlook", *Virtual Phys. Prototyp.*, vol. 18, no. 1, 2023, Art. no. e2181192. DOI: 10.1080/17452759.2023.2181192.

[120] S. Ovchinnikov, *Graphs and Cubes*. Springer New York, NY, 2011.

[121] S. Daynes, "TPMS-based metamaterials with tuneable elastic anisotropy and mechanical coupling", *Materials and Design*, vol. 253, p. 113 866, 2025. DOI: https://doi.org/10.1016/j.matdes.2025.113866.

[122] Universal Robots, *Technical specifications UR10*, Universal Robots, Oct. 2014.

[123] E. Madsen, O. S. Rosenlund, D. Brandt, and X. Zhang, "Comprehensive modeling and identification of nonlinear joint dynamics for collaborative industrial robot manipulators", *Control Engineering Practice*, vol. 101, p. 104 462, 2020. DOI: https://doi.org/10.1016/j.conengprac.2020.104462.

[124] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st. USA: Cambridge University Press, 2017.

[125] RLS, *AksIM-2 off-axis rotary absolute encoder*, RLS, Jun. 2019.

[126] *Manipulating industrial robots — Performance criteria and related test methods ISO 9283:1998*, 1998.

[127] A. Guzman-Bautista, L. Gil-Villacastin, A. Lopez-Arrabal, E. Chacón-Tanarro, J. M. Muñoz-Guijosa, and A. Vizán-Idoipe, "Automatic State Space-Based Dynamic Characterization of Industrial Machining Robots", in *2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, 2025, pp. 45–49. DOI: 10.1109/ICMERR64601.2025.10949941.

[128] A. Guzmán-Bautista, A. López-Arrabal, E. Chacón-Tanarro, and B. Pérez-Hickman, "Metodología de calibración de estación de mecanizado robótico basada en flujo de trabajo CNC (in Spanish)", in *XXIV Congreso Nacional de Ingeniería Mecánica*, 2023.

[129] S. Diao, X. Chen, and J. Luo, "Development and Experimental Evaluation of a 3D Vision System for Grinding Robot", *Sensors*, vol. 18, no. 9, 2018. DOI: 10.3390/s18093078.

[130] S. D'Avella, C. A. Avizzano, and P. Tripicchio, "ROS-Industrial based robotic cell for Industry 4.0: Eye-in-hand stereo camera and visual servoing for flexible, fast, and accurate picking and hooking in the production line", *Robotics and Computer-Integrated Manufacturing*, vol. 80, p. 102 453, 2023. DOI: https://doi.org/10.1016/j.rcim.2022.102453.

[131] D. Kato, K. Yoshitsugu, N. Maeda, T. Hirogaki, E. Aoyama, and K. Takahashi, "Positioning Error Calibration of Industrial Robots Based on Random Forest", *International Journal of Automation Technology*, vol. 15, no. 5, pp. 581–589, 2021. DOI: 10.20965/ijat.2021.p0581.

[132] P. Lei and L. Zheng, "An automated in-situ alignment approach for finish machining assembly interfaces of large-scale components", *Robotics and Computer-Integrated Manufacturing*, vol. 46, pp. 130–143, 2017. DOI: https://doi.org/10.1016/j.rcim.2017.01.004.

[133] R. Kluz and T. Trzepieciński, “The repeatability positioning analysis of the industrial robot arm”, *Assembly Automation*, vol. 34, no. 3, pp. 285–295, 2014. DOI: 10.1108/AA-07-2013-070.

[134] Smart Materials 3D. “PLA – Filamento (bobina L 1000 g, Ø 1.75 mm)”. es. Filamento PLA biodegradable; sello “High-Speed”, Smart Materials 3D. (2025), [Online]. Available: https://www.smartmaterials3d.com/pla-filamento#/3-tamano-l_1000g/26-diametro-175_mm/ (visited on 04/29/2025).

[135] O. Bouzaglou, O. Golan, and N. Lachman, “Process design and parameters interaction in material extrusion 3d printing: A review”, *Polymers*, vol. 15, no. 10, 2023. DOI: 10.3390/polym15102280.