

Towards Accessible Non-Planar FFF Using Triple Z-Axis Kinematics

Giovanni Cocco

Université de Lorraine, CNRS, Inria,
LORIA
Nancy, France
giovanni.cocco@inria.fr

Eric Garner

Université de Lorraine, CNRS, Inria,
LORIA
Nancy, France
eric.garner@inria.fr

Vincent Belle

Université de Lorraine, CNRS, Inria,
LORIA
Nancy, France
vincent.belle@inria.fr

Cédric Zanni

Université de Lorraine, CNRS, Inria,
LORIA
Nancy, France
cedric.zanni@loria.fr

Xavier Chermann

Université de Lorraine, CNRS, Inria,
LORIA
Nancy, France
xavier.chermann@inria.fr

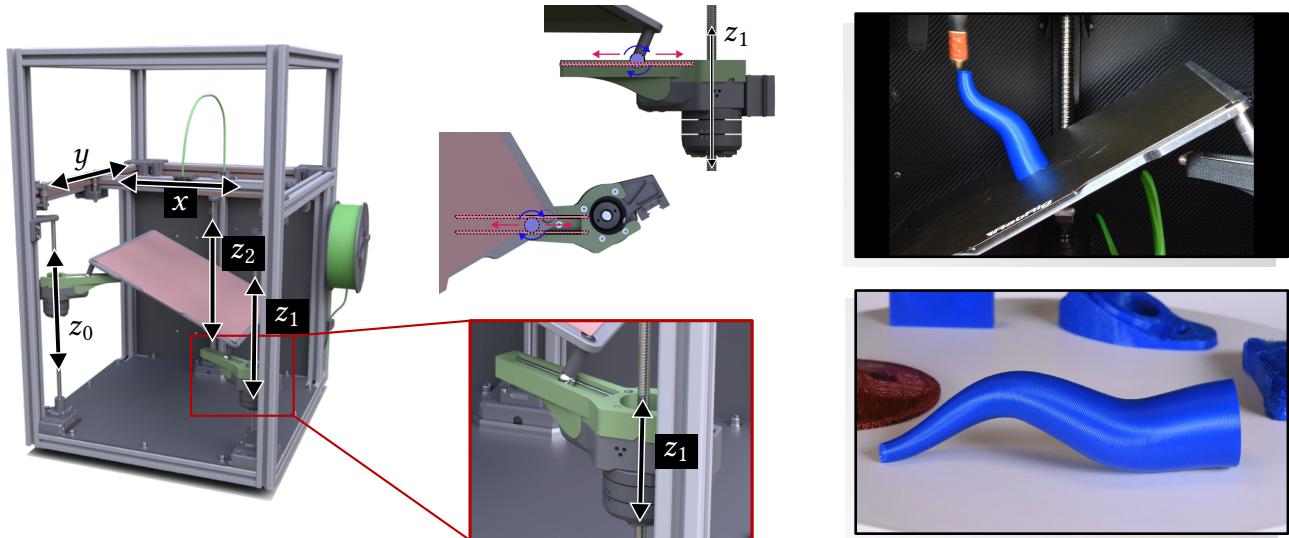


Figure 1: We propose to extend the capabilities of printers with three independent Z-axes (z_0, z_1, z_2), enabling a transition from planar to non-planar deposition. The required modification is minimal, involving only an extension of the bed screws and rails. We implemented this modification on the RatRig V-Core 3.1, illustrated with its sliding balls (blue) moving along planar rails (red). This ball-on-rail system enables precise non-planar printing (right).

Abstract

We present a novel approach to non-planar fused filament fabrication (FFF) aimed at making this advanced technique more accessible to the maker and research communities. By enabling the print bed to tilt using three independently actuated Z-axes, our method allows standard 3D printers—such as the commercially available RatRig—to perform non-planar printing with minimal hardware modifications. Specifically, only simple extensions to the rails and bed screws are

required. This enables complex, curved-layer fabrication without the need for expensive or sophisticated robotic arms.

We derive the kinematic model of the 3Z configuration and provide an open-source Python implementation that maps 3D toolpaths to machine commands. We provide a detailed study of this machine design’s strengths and limitations in the context of FFF, including a build volume study, tolerance analysis, and technical details on trajectory interpolation in machine space. Several models were 3D printed using non-planar deposition to demonstrate the feasibility of the proposed approach, with our experiments achieving bed tilts of up to 30 degrees.

By lowering the entry barrier to non-planar FFF, we aim to empower a broader community of makers, educators, and researchers to experiment, innovate, and contribute to the growing field of advanced additive manufacturing.

CCS Concepts

- Computing methodologies → Modeling methodologies.

Keywords

kinematic modeling, fused filament fabrication, non-planar printing, additive manufacturing, 3D printing

ACM Reference Format:

Giovanni Cocco, Eric Garner, Vincent Belle, Cédric Zanni, and Xavier Chermain. 2025. Towards Accessible Non-Planar FFF Using Triple Z-Axis Kinematics. In *ACM Symposium on Computational Fabrication (SCF '25), November 20–21, 2025, Cambridge, MA, USA*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3745778.3766652>

1 Introduction

In the rapidly advancing field of additive manufacturing, emerging techniques continually push the boundaries of part customizability and optimization. Non-planar fused filament fabrication (FFF), in particular, has garnered significant interest for its potential to enhance mechanical properties and manufacturing precision. By aligning material layers and trajectories with surface geometry or mechanical stresses, components can achieve increased stiffness and toughness, reduced weight, and improved accuracy [Etienne et al. 2019; Zhang et al. 2022]. The significance of non-planar printing is demonstrated by its wide range of applications and computational methodologies, including shell printing [Mitropoulou et al. 2025], support-free printing [Dai et al. 2018; Zhang et al. 2022], wireframe printing [Huang et al. 2016; Peng et al. 2016], carbon fiber printing [Zhang et al. 2025], neural slicing [Liu et al. 2024], and atomizing [Chermain et al. 2025]. While this substantial body of research focuses on novel algorithms for generating toolpaths, comparatively less attention has been given to the hardware requirements essential for fabrication.

Non-planar FFF relies on 5-axis extrusion systems, and the majority of the aforementioned methods use robotic arms [Dai et al. 2018; Huang et al. 2016; Liu et al. 2024; Mitropoulou et al. 2025; Zhang et al. 2022]. Robotic arms offer greater freedom of movement compared to 3-axis printers, but they are an order of magnitude more expensive and more technically complex, making them less accessible. To democratize non-planar FFF, Open5X [Hong et al. 2022] introduced an affordable and accessible method to upgrade standard 3-axis printers to 5-axis systems. This upgrade involves adding a 2-axis rotating gantry that introduces two rotational degrees of freedom—an approach that has also been applied to Delta printers [Peng et al. 2016]. While less complex than robotic arms, adding two rotational axes still requires significant hardware modifications.

Recently, the authors of Atomizer [Chermain et al. 2025] demonstrated non-planar FFF using a commercially available 3D printer—the RatRig V-Core 3.1 [RatRig 2025]—equipped with a three-point kinematic bed. The printer's two additional Z-axes, originally intended for automatic bed leveling, were repurposed to tilt the bed during printing. This effectively transformed the RatRig into a 5-axis printer without hardware modification. However, the approach had notable limitations, including a maximum tilt angle of only 5 degrees and a custom 3Z kinematic model that was not disclosed in the original article.

To overcome these limitations, we:

- present the 3Z kinematic model with closed-form solutions, mapping tool position and orientation to machine axis coordinates;
- release an open-source Python implementation of the 3Z kinematic model;
- propose a minimal hardware modification—extending the rails and bed screws (Fig. 1, left)—that allows the bed to tilt up to 30 degrees;
- validate the kinematic model and hardware modification by performing non-planar FFF (Fig. 1, right); and
- provide an analysis of build volume, tolerance, and machine-space interpolation in 3Z printers.

To facilitate bed tilting, each of the z-axes is equipped with a ball-on-rail system that enables the bed to both rotate and slide, as shown in Figure 1. This relatively inexpensive setup reduces the cost of entry for non-planar printing and offers a robust option for researchers and hobbyists alike to explore printing on sloped surfaces. The maximum tilt angle is limited to 30°, constrained by the maximum outward extension of the rails. Nevertheless, this proof of concept can inspire further innovation in 3D printing hardware design. Our kinematic derivation makes minimal assumptions about the machine's geometry, allowing experimentation with different machine dimensions, rail directions, and bed shapes.

The robotics literature [Nayak et al. 2018; Yang et al. 2011] typically assumes simpler geometric settings than those of the RatRig. To enable easier derivations, it often considers an equilateral triangle formed by the spherical joints (the balls), with the rails pointing toward its circumcenter. Since this is not the case for the RatRig, we derive a kinematic model that accounts for arbitrary rail directions and arbitrary *xy*-positions of the spherical joints. Given the nonlinear constraints and the large number of kinematic parameters, a closed-form solution is non-trivial. Our geometric approach relies on dimensionality reductions to obtain an efficient closed-form solution.

2 Related Work

Standard FFF printers typically operate with three degrees of freedom (DoFs), enabling precise control over the nozzle's position in space while keeping its orientation constant. However, incorporating as few as two additional DoFs allows for control of the nozzle orientation, significantly expanding the printer's capabilities. The ability to control both tool position and orientation facilitates support-free printing, improved mechanical properties, and reduced staircase effects [Zhang et al. 2022]. Various strategies have been explored to implement these additional DoFs [Yao et al. 2024].

Robotic Arms. Robotic arms equipped with six or more DoFs are commonly utilized for FFF in academic research [Bhatt et al. 2020]. While these additional DoFs enhance flexibility, they also introduce greater complexity, leading to higher costs, reduced precision, and the need for more sophisticated control strategies [Sciavicco and Siciliano 2012]. Specifically, in a chained-link system, such as a robotic arm, angular errors accumulate across each joint, which can only be mitigated with more precise motors and tighter manufacturing tolerances to achieve the same level of precision as simpler machines [Hayati and Mirmirani 1985].

5-Axis Machining Centers. Industrial machines with orientable toolheads, traditionally developed for subtractive manufacturing, have also been adapted for FFF [Gardner et al. 2018; Kaill et al. 2022]. Although they offer more precise tool positioning than robotic arms, the misalignment between the extruder orientation and the direction of gravity impacts material flow, resulting in reduced print quality [Chen et al. 2025].

Custom 3D Printers. Standard 3D printers have been equipped with two additional motors to control bed orientation, offering a low-cost approach to non-planar printing [Hong et al. 2022; Teibrich et al. 2015]. However, this design includes a singularity at the north pole, resulting in large movements to achieve small changes in orientation when the bed is nearly horizontal [Isa and Lazoglu 2019]. Another approach combines radial and rotational motion using a core R-Theta printer [Bird 2025], although it is limited to 4-axis motion.

3Z Configuration. In machines utilizing CoreXY kinematics, the nozzle operates within the xy -plane, while the bed moves along the z -axis. The bed height is typically controlled using one to three lead screws driven by a single motor. However, when three independently actuated lead screws are employed, it becomes possible to control not only the bed's height but also its orientation. Each z -axis is equipped with a carriage connected to the bed via a ball-on-rail system (Fig. 1). As the carriages move along their respective axes, the balls slide and rotate along the rails, resulting in bed tilt. This technology has been commercialized by manufacturers such as RatRig [RatRig 2025], and is primarily used for automatic bed leveling and hot-swapping of build plates [Vasquez et al. 2020]. Bed leveling is achieved by iteratively probing the build plate height and adjusting the z -axis coordinates accordingly prior to printing. However, this approach is unsuitable for non-planar printing, as it cannot be feasibly performed during a print whenever the target bed orientation changes. To enable non-planar FFF with a 3Z system, Chermann et al. [2025] employed a custom kinematic model. Nevertheless, the original RatRig rail and bed screw lengths only allow for a maximum bed tilt angle of 5°. Furthermore, their work focused on toolpath generation rather than the kinematic mapping between the toolpath and the corresponding machine coordinates.

Our method does not focus on toolpath generation, but instead introduces the 3Z kinematic model, which provides efficient closed-form forward and inverse mappings that do not require iterative methods. This formulation, along with a minimal hardware modification of the RatRig, constitutes a key contribution toward accessible non-planar printing.

Parallel Manipulator. The RatRig's bed can be reinterpreted as a 3-legged Prismatic-Prismatic-Spherical (3-PPS) parallel manipulator. In each leg, the active prismatic joint is located at the base and oriented vertically, while the passive prismatic joint is positioned horizontally. Although the robotics literature on 3-PPS kinematic configurations is extensive [Nayak et al. 2018; Yang et al. 2011], to the best of our knowledge, it does not address FFF systems. Moreover, existing methods typically assume that the spherical joints form an equilateral triangle and that the horizontal prismatic joints are directed toward the triangle's circumcenter. Since these assumptions do not match our hardware setup and do not incorporate the

extruder, we derive a custom kinematic model using a geometric approach.

3 Kinematics Derivation

FFF slicers convert input geometry into a toolpath represented as a sequence of points. These points are then translated into machine coordinates to position the machine's axes. In the case of non-planar printing with a 3Z machine, each position \hat{p} is associated with a specific tool orientation \hat{n} , and each pair (\hat{p}, \hat{n}) corresponds to a unique set of machine coordinates (x, y, z_0, z_1, z_2) . Section 3.1 addresses the inverse kinematics problem of converting toolpath points into machine coordinates, while Section 3.2 discusses the forward kinematics problem of translating machine coordinates back into toolpath points, enabling toolpath reconstruction for inspection.

Notation. We introduce the notation used throughout the paper. The components of a 3D vector v are denoted by $(v)_x$, $(v)_y$, and $(v)_z$. The projection of a 3D vector v onto the xy -plane is written as $(v)_{x,y}$, while the embedding of a 2D vector v into 3D space is denoted $(v)_{x,y,0}$. For compactness, rotations and reflections are represented by reordering and negating vector components. For example, $(v)_{-y,x}$ corresponds to a 90° rotation of the 2D vector v . Vectors defined in world-space, bed-space, and plane-space (defined later) are denoted by \square , $\hat{\square}$, and $\tilde{\square}$, respectively. We adopt the column convention for vectors.

Geometric Setup. The geometric setup and the printer's five axes are illustrated in Figures 1 and 2. We assume that the x and y axes are orthogonal, forming the xy -plane. Additionally, the three z -axes are parallel to each other and orthogonal to the xy -plane. The machine coordinates (x, y, z_0, z_1, z_2) control the nozzle position p , as well as the elevation and orientation of the bed. Specifically, the bed is connected to each z -axis via a ball-on-rail system. Each ball, firmly affixed to the bed, can rotate freely but is constrained to linear motion along a fixed direction d_i on its rail. An important assumption is that the directions d_i lie within the xy -plane. The positions of the balls, b_0 , b_1 , and b_2 , have vertical components $(b_0)_z$, $(b_1)_z$, and $(b_2)_z$, which are linearly correlated with the respective z -axis coordinates z_0 , z_1 , and z_2 . The bed orientation n' is derived from these heights under the assumption that the bed remains parallel to the triangle formed by the three balls. In other words, all balls are equidistant from the top surface of the bed. Another key assumption is that the bed is level when homed, which can be ensured by using the machine's bed-leveling feature prior to each print. In our geometric configuration, the z -component of each ball, $(b_i)_z$, is the opposite of the corresponding machine coordinate z_i :

$$(b_i)_z = -z_i, \quad i \in \{0, 1, 2\}. \quad (1)$$

Note that the proposed kinematics are valid only if the aforementioned geometric assumptions are satisfied.

World-Space and Bed-Space. The world-space origin is defined as the position of the first ball, b_0 , when the machine axes – x , y , z_0 , z_1 , and z_2 – are homed, that is, set to 0. The basis of this space consists of the directions (e_1, e_2, e_3) , where e_1 aligns with the x -axis, e_2 with the y -axis, and e_3 is parallel to the set of triple z -axes (Fig. 2).

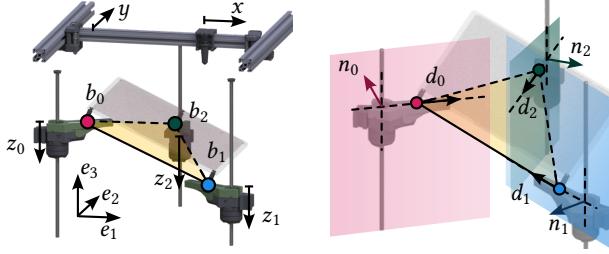


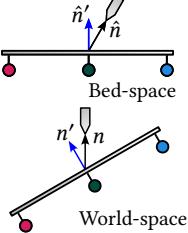
Figure 2: Machine coordinates: x and y move the nozzle, while z_0 , z_1 , and z_2 control the height of each rail. Each ball position b_i is constrained to its associated plane (red, blue, and green), defined by a normal vector n_i that is orthogonal to both the rail direction d_i and the up direction (Eq. 3). The balls form a triangle (yellow) that is parallel to the bed plane.

The trajectories for the printer – i.e., the nozzle position \hat{p} and its orientation \hat{n} – are always specified in *bed-space*, as the object to be fabricated is described within the 3D coordinate system of the print bed. Vectors in *bed-space* are denoted by $\hat{\cdot}$. In the context of planar printing, where the tool orientation \hat{n} aligns with the z -axis, we assume that the machine coordinates correspond directly to the position:

$$x = (\hat{p})_x, \quad y = (\hat{p})_y, \quad z_0 = z_1 = z_2 = (\hat{p})_z. \quad (2)$$

Consequently, the origin of *bed-space* corresponds to the position of the nozzle when all machine axes are homed. In this configuration, the nozzle touches the bed. Since all balls are equidistant from the top surface of the bed, the ball-to-bed distance is given by $|(\hat{b}_i)_z|$.

For non-planar printing, the tool orientation \hat{n} is indirectly controlled by tilting the print bed, which is achieved by independently adjusting the z -coordinates. This tilting alters the alignment between *bed-space* and *world-space*, making a thorough understanding of this relationship essential for accurately positioning the nozzle within *bed-space*. The target bed normal in *world-space*, n' , is derived from the tool orientation in *bed-space*, \hat{n} , by reflecting it about the z -axis; that is, $n' = (\hat{n})_{-x,-y,z}$ (see inset figure).



Kinematics Parameters. The parameters of the proposed kinematics are the ball positions \hat{b}_i and the rail directions d_i . Note that, as the balls are fixed on the bed, their position in *bed-space* \hat{b}_i is constant, while the rail directions d_i remain constant when expressed in *world-space*. Given these parameters, the target position and orientation (\hat{p}, \hat{n}) can be mapped to the 3Z machine coordinates (x, y, z_0, z_1, z_2) and vice versa.

Balls Constrained to Vertical Planes. From the geometric setup, we derive the following property, which we will use to derive the kinematics. The position of each ball is constrained to lie along a line defined by its rail, and the rail itself is constrained to translate vertically. In other words, the position of each ball b_i is restricted to a plane defined by the rail direction d_i and the vertical direction (Fig. 2). The normal of this plane is given by $n_i = e_3 \times d_i$, and the plane distance from the *world-space* origin is $k_i = -n_i \cdot \bar{b}_i$,

where \bar{b}_i denotes the *world-space* position of the balls in the homed configuration. In this configuration *world-space* and *bed-space* are aligned. We can calculate the ball positions by translating to account for the different origins of the spaces: $\bar{b}_i = \hat{b}_i - \hat{b}_0$. Formally, the ball positions b_i must satisfy (Fig. 2, right):

$$b_i \cdot n_i + k_i = 0, \quad i \in \{0, 1, 2\}. \quad (3)$$

3.1 Inverse kinematics

This section addresses the inverse kinematics problem, which involves finding the set of machine coordinates (x, y, z_0, z_1, z_2) required to achieve a specific nozzle position and orientation (\hat{p}, \hat{n}) . This mapping f^{-1} is essential for converting toolpath waypoints into machine instructions and may be executed millions of times during a single print. Consequently, an efficient solution is paramount.

The inverse kinematics problem is solved in two steps. First, the target orientation \hat{n} is obtained by adjusting the bed normal n' through the z -coordinates z_0, z_1, z_2 (Section 3.1.1). Then, the target nozzle position \hat{p} is achieved by finding the correct (x, y) coordinates and translating the bed along the up direction (Section 3.1.2).

3.1.1 Bed Orientation. In this section, the objective is to find the elevation differences between the rails, $z_1 - z_0$ and $z_2 - z_0$, that satisfy the constraints of the geometric setup while correctly positioning the bed normal n' for the tool orientation \hat{n} . Using Eq. 1, this is equivalent to solving for the elevation differences of the balls. Three constraints apply, namely: (1) the balls must lie in a plane orthogonal to n' ; (2) the ball positions are constrained to vertical planes (see Fig. 2, Eq. 3); and (3) the geometric arrangement of the balls in space is fixed, as the bed and the connections between the bed and the balls are rigid.

Plane-Space Definition. To simplify the problem, we reduce its dimensionality by expressing the last two constraints in a two-dimensional space, referred to as *plane-space*, which is defined such that the first constraint is satisfied by construction. Note that the *plane-space* is equivalent to the *bed-space* xy -plane up to an unknown translation and in-plane rotation.

Vectors in *plane-space* are denoted by $\tilde{\cdot}$. To map a vector v from *world-space* to *plane-space*, we first apply a rotation matrix M and then project the result onto the xy -plane ($z = 0$), yielding $\tilde{v} = (Mv)_{x,y}$, where

$$M = [t \quad b \quad n']^T, \quad b = \frac{n' \times e_1}{\|n' \times e_1\|}, \quad t = \frac{b \times n'}{\|b \times n'\|}, \quad (4)$$

and e_1 is the unit vector along the x -direction.

Balls Constrained to Lines. Let \tilde{b}_0, \tilde{b}_1 , and \tilde{b}_2 denote the ball positions in *plane-space*. The in-plane constraints from Eq. 3 are reduced to line constraints by expressing them in *plane-space* (Fig. 3, left):

$$\tilde{b}_i \cdot \tilde{n}_i + \tilde{k}_i = 0, \quad \tilde{n}_i = (Mn_i)_{x,y}, \quad \tilde{k}_i = k_i, \quad i \in \{0, 1, 2\}. \quad (5)$$

Fixed Geometric Arrangement of Balls. The balls are rigidly connected to the bed, forming a rigid triangle. In *plane-space*, the positions of the second and third balls can be expressed as translations relative to the first. If the balls were not constrained to lines, the rigid triangle they form could be positioned and oriented freely

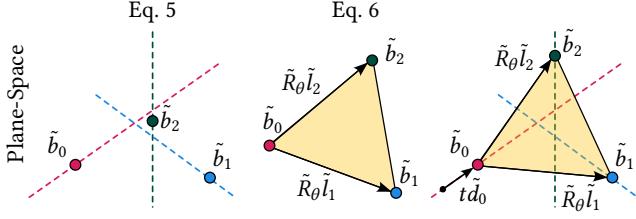


Figure 3: The constraints in plane-space. Left: The balls \tilde{b}_i are constrained to lie on lines. Middle: The three balls form a rigid triangle, whose position and orientation are parameterized by \tilde{b}_0 and an angle θ . Right: The rigid transformation of the triangle is parameterized by the distance t from the origin to the first ball \tilde{b}_0 and the angle θ . We solve for those two parameters to satisfy the constraints.

in the plane. We parameterize the rigid transformation of the triangle by the position of the first ball, \tilde{b}_0 , and an angle θ , which rotates the other ball positions about the first (Fig. 3, middle):

$$\tilde{b}_i = \tilde{b}_0 + \tilde{R}_\theta \tilde{l}_i, \quad \tilde{R}_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad i \in \{1, 2\} \quad (6)$$

The vectors between the balls in plane-space are given by $\tilde{R}_\theta \tilde{l}_i$, where $\tilde{l}_i = (\hat{b}_i - \hat{b}_0)_{x,y}$. Note that the projected vectors in bed-space correspond to the vectors in plane-space, up to an unknown in-plane rotation \tilde{R}_θ .

Solving for Both Constraints. The rigid triangle formed by the balls must be positioned and oriented such that the ball positions satisfy the line constraints (Eq. 3). The position of the first ball in plane-space \tilde{b}_0 depends on the translation component of the transformation (Eq. 6) and is independent of the rotation component. Additionally, it is constrained to lie on the line defined by the rail constraint in Eq. 5, and its position can be expressed as a ray $\tilde{o} + t \tilde{d}_0$, where t is a scalar parameter defining its position. Since the position of the first ball b_0 in the homed configuration was chosen as the origin of world-space, and the plane of the 2D problem passes through this point, the origin of plane-space lies on the rail constraint associated with the first ball. Therefore, the ray origin \tilde{o} can be omitted:

$$\tilde{b}_0 = t \tilde{d}_0. \quad (7)$$

By substituting Eq. 7 into Eq. 6, the position and orientation of the rigid triangle are parameterized by two scalars: t and θ . We can then solve for these two scalars to ensure that all balls lie on their respective constraint lines (Eq. 5, Fig. 3). The constraint on the first ball, \tilde{b}_0 , is satisfied by construction (Eq. 7), while the constraints on the other two balls yield a closed-form solution for t and θ .

First, the constraint on \tilde{b}_1 is solved for t :

$$t = -\frac{(\tilde{l}_1 \cdot \tilde{n}_1) \cos(\theta) + ((\tilde{l}_1)_{-y,x} \cdot \tilde{n}_1) \sin(\theta) + \tilde{k}_1}{\tilde{d}_0 \cdot \tilde{n}_1} \quad (8)$$

This expression is then substituted into the constraint on \tilde{b}_2 to obtain the equation:

$$\begin{aligned} \alpha \cos(\theta) + \beta \sin(\theta) + \gamma &= 0 \\ \alpha &= (\tilde{d}_0 \cdot \tilde{n}_1)(\tilde{l}_2 \cdot \tilde{n}_2) - (\tilde{d}_0 \cdot \tilde{n}_2)(\tilde{l}_1 \cdot \tilde{n}_1) \\ \beta &= (\tilde{d}_0 \cdot \tilde{n}_1)((\tilde{l}_2)_{-y,x} \cdot \tilde{n}_2) - (\tilde{d}_0 \cdot \tilde{n}_2)((\tilde{l}_1)_{-y,x} \cdot \tilde{n}_1) \\ \gamma &= \tilde{d}_0 \cdot (\tilde{k}_2 \tilde{n}_1 - \tilde{k}_1 \tilde{n}_2) \end{aligned} \quad (9)$$

This leads to two possible solutions, as shown in the inset figure. The correct solution is the one that introduces the least rotation, that is, the one with the smallest absolute value of θ . To that end, the tangent half-angle substitution $\eta = \tan(\theta/2)$ is applied, resulting in the second-degree equation:

$$(\gamma - \alpha)\eta^2 + 2\beta\eta + \alpha + \gamma = 0, \quad (10)$$

which can be solved in closed-form for θ with the smallest absolute value as:

$$\theta = \begin{cases} 2 \tan^{-1} \left(\frac{\alpha + \gamma}{-\beta - \text{sign}(\beta) \sqrt{\alpha^2 + \beta^2 - \gamma^2}} \right), & \alpha + \gamma \neq 0 \\ 0, & \alpha + \gamma = 0 \end{cases} \quad (11)$$

Once θ is known, it can be substituted into Eq. 8 to obtain t , which can then be used to compute the position of the first ball \tilde{b}_0 using Eq. 7. The positions of the other balls, \tilde{b}_1 and \tilde{b}_2 , can then be solved via Eq. 6.

The world-space positions of the balls b_i can now be computed by inverting the initial rotation M . However, since the position of the plane with n' as the normal was chosen arbitrarily, the ball positions are known up to a vertical translation C .

$$b_i = M^T (\tilde{b}_i)_{x,y,0} + C, \quad i \in \{0, 1, 2\} \quad (12)$$

Since the translation C is the same for each ball, the difference in z-axis coordinates, $z_1 - z_0$ and $z_2 - z_0$, is fully defined for a given target orientation \hat{n} .

3.1.2 Nozzle position. The second sub-problem aims to determine the machine coordinates to correctly place the nozzle in bed-space. Let \vec{p} be the world-space nozzle position when the machine is positioned for planar printing at the target position \hat{p} (Fig. 4, left). The bed is first tilted by moving z_1 and z_2 using the balls' relative positions $(b_1 - b_0)_z$ and $(b_2 - b_0)_z$. This operation ensures the correct bed normal n' , but the introduced roto-translation of the bed results in the wrong nozzle position in bed-space. This roto-translation moves the \vec{p} into \vec{p}' (Fig. 4, center). The vector $\vec{p}' - \vec{p}$ can be used to update the machine coordinates to correct the nozzle position in bed-space. Its xy -components are used to move the nozzle, whereas its z -component is used to move the three z-axes by the same distance (Fig. 4, right).

Bed Roto-translation. By fixing z_0 and adjusting z_1 and z_2 according to their offsets from z_0 , the bed rotates about the first ball, and the latter translates along its rail. The aforementioned rotation about the first ball \tilde{b}_0 is the combination of the in-plane rotation R_θ followed by the rotation to obtain the correct normal M^T . As the oriented bed now only moves horizontally, it is convenient to express this horizontal translation in world-space, after the rotations

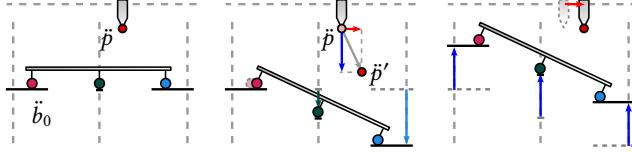


Figure 4: Left: The nozzle position derivation starts by positioning the tool at the target position \hat{p} with a planar configuration (Eq. 2). In this setting, let \hat{p} and \hat{b}_0 be the tool and first ball positions in world-space. Center: To achieve the target tool orientation \hat{n} , the second and third ball elevation are modified using Eq. 12. Consequently, the bed's roto-translation moves the point \hat{p} to \hat{p}' (Eq. 13). Right: The difference between \hat{p}' and \hat{p} is used to correctly position the nozzle in bed-space. The horizontal component of the difference vector is used to move the nozzle (red vector), while the vertical component is used to adjust the bed height (blue vectors).

have been applied, as $(b_0)_{x,y,0}$. By applying this roto-translation to \hat{p} , we obtain \hat{p}' :

$$\begin{aligned} \hat{p}' &= M^T R_\theta(\hat{p} - \hat{b}_0) + \hat{b}_0 + (b_0)_{x,y,0} \\ R_\theta &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (13)$$

Machine Axes Values. The values \hat{p} and \hat{b}_0 are not known, so we seek to reformulate the problem in terms of \hat{p} and \hat{b}_0 . To correct the machine axes, we are only interested in the difference between $\hat{p}' - \hat{p}$, so we can translate the space for our convenience. Let this translation be encoded by the vector g , we define g such that $\hat{p} = \hat{p}' + g$ and we define $p' = \hat{p}' - g$. Then, the following hold: $\hat{p}' - \hat{p} = p' - \hat{p}$. Additionally, given that in the planar configuration bed-space is aligned to world-space, we have that $\hat{b}_0 = \hat{b}_0 + g$. We can obtain p' from Eq. 13 by substituting the previous terms. Note that g cancels out and that the values on the right side of the equation are known:

$$p' = M^T R_\theta(\hat{p} - \hat{b}_0) + \hat{b}_0 + (b_0)_{x,y,0} \quad (14)$$

The machine axes must now be moved by their difference $p' - \hat{p}$, where the z -component of this difference must be applied to the three z -axes. For example, before the bed is tilted, the x -coordinate of the machine was $(\hat{p})_x$, as per Eq. 2. By adding the aforementioned difference, we obtain the correct value $(p')_x$. Repeating this process for all axes produces the machine coordinates:

$$\begin{aligned} x &= (p')_x, & y &= (p')_y, & z_0 &= (p')_z \\ z_1 &= z_0 + (b_0 - b_1)_z, & z_2 &= z_0 + (b_0 - b_2)_z \end{aligned} \quad (15)$$

3.2 Forward kinematics

The forward kinematics problem is concerned with the mapping f from machine coordinates (x, y, z_0, z_1, z_2) to the corresponding point in bed-space (\hat{p}, \hat{n}) . The sliding of the balls along their rails makes the bed normal non-trivial to compute. To derive the kinematics, we solve for the bed roto-translation that satisfies geometric

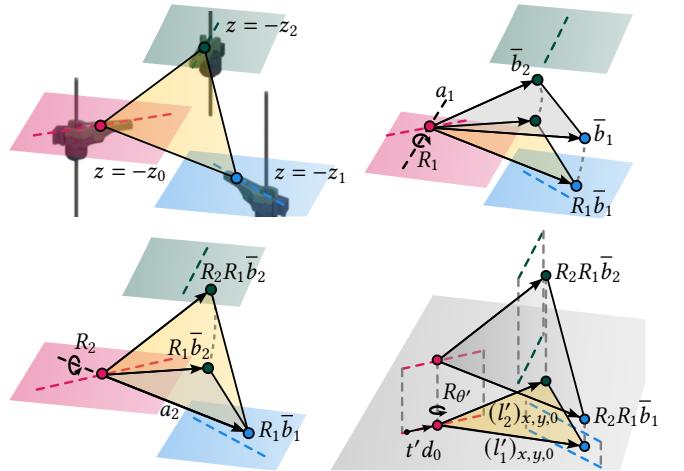


Figure 5: Illustration of the steps in the forward kinematics derivation. The given machine coordinates (z_0, z_1, z_2) constrain the balls to three horizontal planes (top left). The derivation begins with the planar configuration (top right), and the goal is to determine a bed roto-translation that satisfies the geometric constraints. First, we solve for a rotation R_1 that corrects the relative height between the red and blue balls (top right). Next, we compute a second rotation R_2 that adjusts the relative height between the red and green balls (bottom left). In the final step (bottom right), we solve for a horizontal rotation $R_{\theta'}$ and a translation $t'd_0$ in a 2D domain (gray plane) to align the balls with the rails (dashed lines).

constraints. The position of each ball b_i is constrained to a horizontal plane $z = -z_i$ (Eq. 1, Fig. 5, top left), and to a vertical plane (Eq. 3). The bed roto-translation is solved in two sub-steps. First, we solve for a rotation $R_2 R_1$ that satisfies the horizontal plane constraints (Section 3.2.1, Fig. 5, bottom left). Then, we solve for a rotation $R_{\theta'}$ about the world vertical axis e_3 , and for the bed translation that satisfies the vertical plane constraints (Section 3.2.2, Fig. 5, bottom right). By inverting the bed roto-translation in Eq. 14, the bed-space nozzle position \hat{p} and orientation \hat{n} can be recovered (Section 3.2.3).

3.2.1 Placement of Balls on Horizontal Planes. The rotation that satisfies the constraints in Eq. 1 up to a vertical translation depends on the relative positions of the second and third balls with respect to the first. As \bar{b}_i denotes the world-space positions of the balls in the homed configuration, and the origin of world-space is \bar{b}_0 , applying this rotation to \bar{b}_1 and \bar{b}_2 results in vectors that correspond to the relative positions of the balls up to a rotation about the vertical axis. This rotation can be expressed as two consecutive sub-rotations, R_1 and R_2 , which respectively satisfy the first and second horizontal plane constraints defined in Eq. 1. The rotation R_2 fulfills the second constraint without violating the first (Fig. 5):

$$(R_1 \bar{b}_1)_z = z_0 - z_1 \quad (16)$$

$$(R_2 R_1 \bar{b}_i)_z = z_0 - z_i \quad i \in \{1, 2\}. \quad (17)$$

To solve for R_1 and R_2 , we perform a dimensionality reduction by fixing the rotation axes and solving for the rotation angles. The

rotation R_1 is defined by an angle θ_1 and an axis a_1 . The axis a_1 is chosen such that it lies in the xy -plane, passes through the point \bar{b}_0 , and is orthogonal to the vector \bar{b}_1 (Fig. 5, top right). Positioning a_1 in this way ensures that such an angle θ_1 exists. Similarly, the rotation R_2 is defined by an angle θ_2 and an axis a_2 . The axis a_2 is given by the vector $R_1\bar{b}_1$. This ensures that the first ball constraint is satisfied for any value of θ_2 (Fig. 5, bottom left).

Solving for the Angles. We first solve for θ_1 using Eq. 16, then for θ_2 using Eq. 17 ($i = 2$). Both cases are equivalent to finding a rotation angle $\ddot{\theta}$ such that rotating a vector \ddot{v} about an axis \ddot{a} by $\ddot{\theta}$ results in a vector whose z -component matches a target value \ddot{z} . Using Rodrigues' rotation formula, this can be expressed as:

$$(\ddot{v} \cos(\ddot{\theta}) + (\ddot{a} \times \ddot{v}) \sin(\ddot{\theta}) + \ddot{a}(\ddot{a} \cdot \ddot{v})(1 - \cos(\ddot{\theta})))_z = \ddot{z}, \quad (18)$$

that can be rewritten as:

$$\begin{aligned} \ddot{\alpha} \cos(\ddot{\theta}) + \ddot{\beta} \sin(\ddot{\theta}) + \ddot{\gamma} &= 0 \\ \ddot{\alpha} = (\ddot{v})_z - (\ddot{a})_z(\ddot{v} \cdot \ddot{a}), \quad \ddot{\beta} = (\ddot{a} \times \ddot{v})_z, \quad \ddot{\gamma} = (\ddot{a})_z(\ddot{v} \cdot \ddot{a}) - \ddot{z} \end{aligned} \quad (19)$$

The correct solution corresponds to the angle $\ddot{\theta}$ with the smallest absolute value, which can be solved via Eq. 11.

3.2.2 Placement of Balls on Vertical Planes. Using the rotations R_1 and R_2 , we can compute the vectors between the balls up to a rotation about the vertical axis. To obtain the bed roto-translation, these vectors, as well as the machine constraints in Eq. 3, are projected onto the horizontal plane $z = 0$ (Fig. 5). This results in a 2D problem with the same structure as the one in the inverse kinematics. The system to solve for the scalars t' and θ' is:

$$\begin{aligned} (b_0)_{x,y} &= t'(d_0)_{x,y}, \quad (b_i)_{x,y} = (b_0)_{x,y} + \tilde{R}_{\theta'} l'_i, \quad i \in \{1, 2\}, \\ l'_i &= (R_2 R_1 \bar{b}_i)_{x,y} \end{aligned} \quad (20)$$

where $\tilde{R}_{\theta'}$ is defined by substituting θ with θ' in Eq. 6.

The system is solved by substitution; first solving for the translation t' as in Eq. 8:

$$t' = \frac{(l'_1 \cdot (n_1)_{x,y}) \cos(\theta') + ((l'_1)_{-y,x} \cdot (n_1)_{x,y}) \sin(\theta') + k_1}{-d_0 \cdot n_1}, \quad (21)$$

and then for θ' , using Eq. 11, as in Eq. 9:

$$\begin{aligned} \alpha' \cos(\theta') + \beta' \sin(\theta') + \gamma' &= 0 \\ \alpha' = (d_0 \cdot n_1)(l'_2 \cdot (n_2)_{x,y}) - (d_0 \cdot n_2)(l'_1 \cdot (n_1)_{x,y}) \quad (22) \\ \beta' = (d_0 \cdot n_1)((l'_2)_{-y,x} \cdot (n_2)_{x,y}) - (d_0 \cdot n_2)((l'_1)_{-y,x} \cdot (n_1)_{x,y}) \\ \gamma' = d_0 \cdot (k_2 n_1 - k_1 n_2) \end{aligned}$$

3.2.3 Solving for the Nozzle Position and Orientation. The overall bed rotation $R_{\theta'} R_2 R_1$, where $R_{\theta'}$ is defined by substituting θ with θ' in Eq. 13, is equivalent to the rotation $M^T R_{\theta}$ from Eq. 14. The nozzle position in bed-space \hat{p} is obtained by inverting Eq. 15, and the bed normal n' is obtained by applying the overall bed rotation to the unitary vector in the z -direction e_3 . Finally, the tool orientation \hat{n} is obtained by mirroring the bed normal n' about e_3 :

$$\begin{aligned} \hat{p} &= R_1^T R_2^T R_{\theta'}^T ((x - (b_0)_x, y - (b_0)_y, z_0)^T - \hat{b}_0) + \hat{b}_0 \\ n' &= R_{\theta'} R_2 R_1 e_3, \quad \hat{n} = (n')_{-x,-y,z} \end{aligned} \quad (23)$$

4 Results

We validate the kinematics introduced previously through both virtual and physical experiments. To that end, we implemented the forward (Section 3.2) and inverse kinematics (Section 3.1) in Python. For the physical validation, we performed non-planar printing in Section 4.6 using a RatRig V-Core 3.1 equipped with RepRap firmware version 3.6.0. The machine's design and build volume are described in Sections 4.1 and 4.2, respectively. We discuss the numerical stability of the kinematics by checking the property $(\hat{p}, \hat{n}) = (f \circ f^{-1})(\hat{p}, \hat{n})$ in Section 4.3. The tolerance analysis and firmware point interpolation are discussed in Sections 4.4 and 4.5, respectively. A comparison with 3-axis printing is presented in Section 4.7. The hardware limitations are analyzed in Section 4.8. The Python implementation of the kinematics is provided in the supplemental material.

4.1 Machine Design

The RatRig hardware was used [RatRig 2025], with modifications to the z -carriages and the bed standoffs, which were extended to increase the maximum achievable tilt angle. A commercially available longer nozzle—the Nonplanar Volcano brass 0.6 mm—was also installed to avoid collisions [Nonplanar.xyz 2025]. The relevant machine parameters (mm) are as follows: the ball positions in bed-space are: $\hat{b}_0 = (-4.07, -12.16, -45.7)^T$, $\hat{b}_1 = (304.93, -12.16, -45.7)^T$, and $\hat{b}_2 = (150.43, 296.84, -45.7)^T$; The slot directions d_0 , d_1 , and d_2 are oriented at 29.89° , 150.11° , and -90° , respectively, from the x -axis. The planar build volume is a box of $300 \times 293 \times 205$ mm.

The original z -carriages house a ball-on-rail system allowing for 10 mm of travel, with the balls centered in their respective rails when the bed is horizontal. Since the achievable tilt angle is limited by the length of the rails, the z -carriages were modified to accommodate longer rails. The relation between tilt angle and rail length was investigated by computing the signed distance from each ball to its position along its rail in the horizontal configuration and recording the maximum and minimum values as the bed precesses about the vertical axis. The resulting ball displacements, shown in Fig. 6, reveal that the balls tend to be displaced more prominently in the inward direction, towards the center of the bed. In order to achieve a maximum tilt angle of 30° , custom carriages were designed with the rails extended asymmetrically to allow for 69 mm of inward travel and 19 mm of outward travel from the planar position. A side effect of the elongated rails is that the bottom of the bed may collide with the rails at higher tilt angles. It was therefore necessary to increase the distance measured from the center of the balls to the top of the bed $|(\hat{b}_i)_z|$ from 19.3 to 45.7 mm by lengthening the standoffs. The required distance is shown as a function of the tilt angle in Fig. 6.

4.2 Build Volume

A major limitation of printer designs that achieve non-planar printing by tilting the bed is the restricted build volume relative to their overall footprint. Here, the 3Z design offers a key advantage: its build volume is inherently equivalent to that of a standard CoreXY printer and only reduces dynamically, depending on the maximum tilt angle used during a print. This means large planar prints can be accommodated without any retooling.

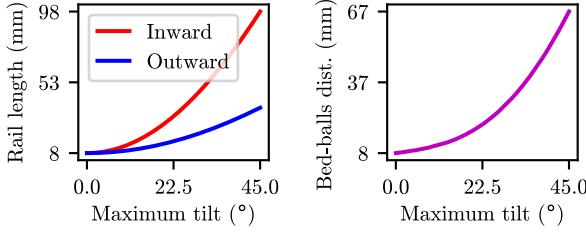


Figure 6: Plots of the required rail lengths (left) and minimum bed-ball distance $|(\hat{b}_i)_z|$ (right) as a function of the maximum tilt angle. Note that the minimum rail length of 8 mm corresponds to the radius of the balls.

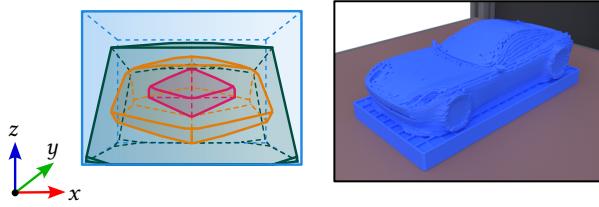


Figure 7: Left: Bed-space build volume for tilt angles of 0° (blue), 10°, 20°, and 30° (red). **Right:** Object with an additional raft to fit inside the build volume.

Fig. 7 illustrates the effective bounding box within bed-space for tilt angles ranging from 0° to 30°. As expected, the standard rectangular prism is preserved when the tilt angle is 0°. However, increasing the tilt angle transforms the volume into an asymmetrical polyhedron. The slanted faces arise either because the nozzle's reachable area, as well as the rails positions, are constrained by the limits of the machine axes, or due to potential collision between the bed and the overhanging XY gantry. Moreover, the shape of the base highlights a critical constraint: printing at high tilt angles far from the bed's center is not feasible when the nozzle is near the bed surface, as this would lead to a collision. This limitation can be addressed by either increasing the clearance between the bed and the gantry, reducing the size of the bed itself, or printing a raft to offset the first non-planar layer of the printed model from the bed (Fig. 7, right). The use of a longer nozzle further increases the clearance between the bed and the gantry.

4.3 Numerical Stability

To validate the numerical stability of our kinematics implementation, we check the property $(\hat{p}, \hat{n}) = (f \circ f^{-1})(\hat{p}, \hat{n})$. We define the possible tool positions as the points within the planar build volume (Sec. 4.1) and the possible orientations as all the orientations up to 30°. Considering the possible tool positions \hat{p} and orientations \hat{n} , we obtain a worst-case error of 3.2×10^{-13} mm for the position and $1.5^\circ \times 10^{-5}$ for the orientation, using double precision floating points.

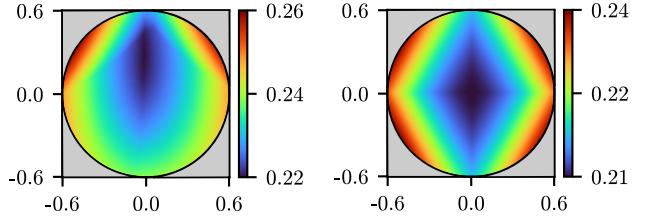


Figure 8: Worst-case error induced by an axis motor tolerance of ± 0.05 mm as a function of tool orientation (up to 30°) in slope-space. Left: position error (mm). Right: tool orientation error ($^\circ \times 10$). The circle represents the set of orientations at 30°.

Table 1: Error statistics, as defined in Figure 8, for various axis motor tolerance values.

Tolerance	Position (mm)			Orientation ($^\circ \times 10$)		
	Min	Max	Average	Min	Max	Average
± 0.01 mm	0.04	0.05	0.05	0.04	0.05	0.04
± 0.05 mm	0.22	0.26	0.23	0.21	0.24	0.22
± 0.1 mm	0.44	0.51	0.47	0.41	0.48	0.44

4.4 Tolerance Analysis

Printing accuracy is affected by the precision of the axis motors and the tolerances of the machine components. To evaluate the impact of motor position accuracy, each axis position is perturbed by ± 0.05 mm, and the resulting worst-case nozzle position and orientation errors are plotted as a function of the nozzle orientation in slope-space (Fig. 8). The slope-space position of a nozzle orientation \hat{n} is defined as: $-(\hat{n}_x / (\hat{n}_z), -(\hat{n}_y / (\hat{n}_z))^T$. The worst-case error is defined as the maximum error over the possible nozzle positions and all combinations of individually perturbed axes. Error statistics for various tolerance values are provided in Table 1.

In addition to motor precision, printing accuracy is also influenced by the precision of the machine components. The error induced by deviations from the nominal machine dimensions involved in the kinematics solver $(\hat{b}_0, \hat{b}_1, \hat{b}_2, d_0, d_1, d_2)$ is evaluated as follows. Given a target nozzle position and orientation, the machine coordinates are computed based on the nominal machine parameters. The true position and orientation are then computed via forward kinematics, with a deliberate tolerance error introduced in each of the machine parameters. In Figure 9, the worst-case error, defined as the maximum error over the possible nozzle positions and all combinations of individually perturbed machine dimensions, is plotted with respect to the tool orientation in slope-space. Statistics for various tolerance values are provided in Table 2. These results reveal that the error is positively correlated with the tilt angle.

4.5 Firmware Point Interpolation

In traditional FFF printing, toolpaths are transmitted as a list of cartesian coordinates to the printer controller. The machine's firmware is responsible for calculating the corresponding motor positions

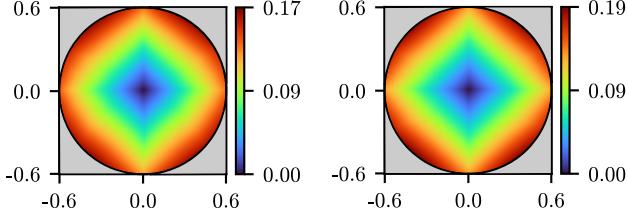


Figure 9: Worst-case error induced by a tolerance of ± 0.05 mm for the kinematics parameters \hat{b}_0 , \hat{b}_1 , and \hat{b}_2 , and a tolerance of $\pm 0.05^\circ$ for the angular parameters d_0 , d_1 , and d_2 , shown as a function of tool orientation (up to 30°) in slope-space. Left: position error (mm). Right: tool orientation error ($^\circ \times 10$).

Table 2: Statistics on the errors, as defined in Figure 9, for different tolerances.

Tolerance	Position (mm)			Orientation ($^\circ \times 10$)		
	Min	Max	Average	Min	Max	Average
± 0.01 mm/ $^\circ$	0.00	0.03	0.02	0.00	0.04	0.02
± 0.05 mm/ $^\circ$	0.00	0.17	0.10	0.00	0.19	0.11
± 0.1 mm/ $^\circ$	0.00	0.34	0.20	0.00	0.37	0.22

and interpolating the shortest path between each point, adjusting the machine’s axes accordingly. However, firmware development was beyond the scope of this study. Therefore, we explicitly provided machine coordinates (x, y, z_0, z_1, z_2) in the G-code instead of relying on onboard interpolation from cartesian coordinates.

Since the controller automatically applies component-wise linear interpolation of the machine coordinates as it moves from one position to the next, some deviation from the intended trajectory is introduced, as illustrated in Fig. 10. We mitigate this effect by providing a fine tessellation of the toolpath in the G-code.

To quantify the error induced by component-wise interpolation, we consider the deviation between this interpolation and the correct one, *i.e.*, linear interpolation of the positions and spherical interpolation of the tool orientations (Fig. 10, left, green trajectory). To compute this error, we finely tessellate the two trajectories and compute the maximum distance between corresponding points, *i.e.*, points generated by the same interpolant. The worst-case error, over the possible nozzle positions and orientations, is plotted as a function of the distance between endpoints in Figure 11. The results show that although both the cartesian distance and the polar angle affect the error, the latter plays a more significant role. To minimize this issue, a finely tessellated toolpath, with at most 1 mm and 1° between points, corresponding to a worst-case error of 0.05 mm and 0.01° , should be used. Finer tessellation may result in heavier G-code, which requires more memory and greater computational power for the firmware to process. Moreover, it can lead to less smooth machine motions, as toolpaths that appear smooth in terms of (\hat{p}, \hat{n}) may not be equally smooth in machine space (Fig. 10, right).

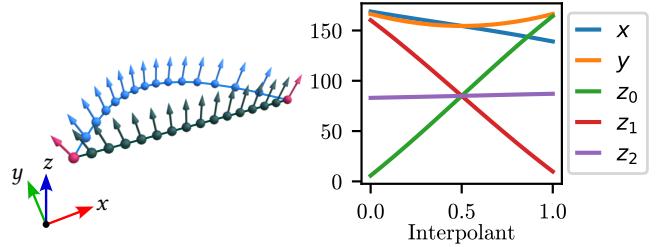


Figure 10: Left: Linear interpolation in machine space (blue trajectory) between two end points (red points), separated by 30 mm and 60° , compared to the desired trajectory obtained by separately interpolating the target position and tool orientation (green trajectory). Note how both the trajectory and point spacing, *i.e.* speed, are affected. Right: The required machine coordinates to obtain the desired trajectory. In this particular instance y requires a non-linear interpolation.

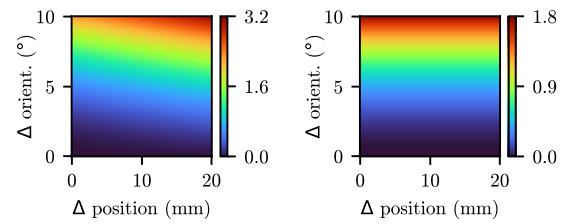


Figure 11: Worst-case error induced by component-wise linear interpolation as a function of cartesian and polar displacement. Left: position error (mm). Right: tool orientation error ($^\circ \times 10$). We recommend tessellating toolpath positions and orientations with a maximum spacing of 1 mm and 1° , respectively, to avoid machine interpolation errors.

4.6 Physical Validation

We validate the kinematics by printing various models (Fig. 12). First, we convert a toolpath, *i.e.*, a sequence of tool positions and orientations (\hat{p}, \hat{n}) , into a corresponding sequence of machine coordinates (x, y, z_0, z_1, z_2) . This sequence is then encoded into G-code. The machine firmware reads the G-code and actuates the motors to print the object. Notably, RepRap firmware can be configured to interpret 5-axis G-code without any modifications. The supplemental video shows the printer in action.

Procedural Models. Two toolpaths were generated procedurally. The *Horn* is composed of multiple layers tilted up to 30° . Each layer consists of a circle, and at each transition between layers, the radius is reduced and the machine orientation is adjusted, producing a twisted horn shape. This model shows no visible defects, highlighting the machine’s capability to print across a wide range of orientations. The *Bump* tests the machine’s ability to follow accurate trajectories while tilting up to 23° . Note that, despite its simple appearance, this model requires precise and complex nozzle movements to compensate for the tilting bed. The printed result

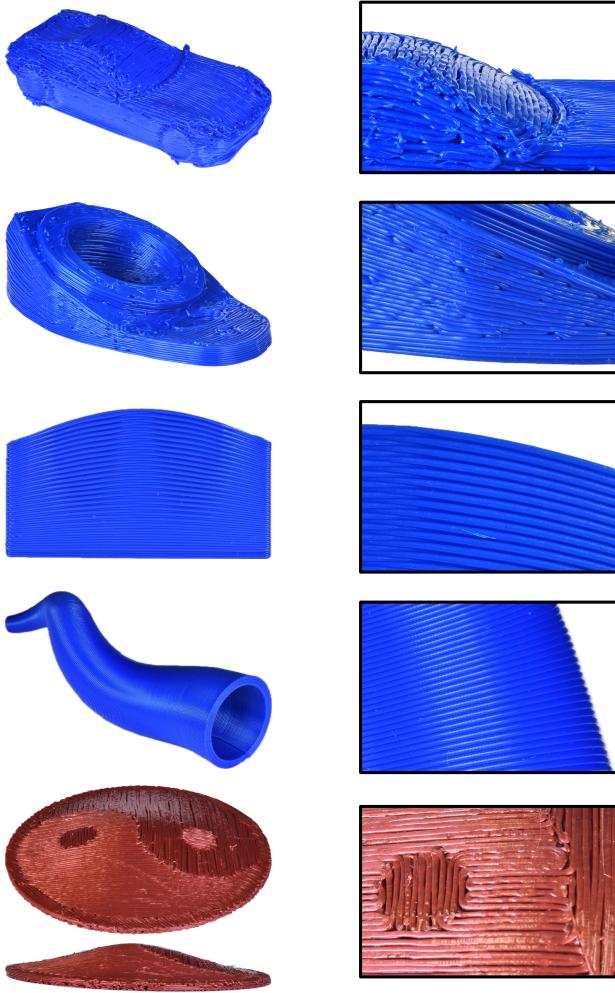


Figure 12: The printed objects, from top to bottom: *Car*, *Ankle*, *Bump*, *Horn*, and *Yin-yang*, with a close-up of the same objects shown on the right. The deposition width is 0.9 mm, and the layer height is approximately 0.45 mm. For procedural models, the layer height is dynamically adjusted to account for the compression and decompression effects caused by the curved layers.

demonstrates good quality, though it is slightly affected by minor artifacts due to mechanical backlash of the bed.

Mesh Models. Additional toolpaths were generated from STL meshes using Atomizer [Chermain et al. 2025]. To this end, we integrated the kinematic model into the Atomizer toolpath generator, targeting the printer described in Section 4.1. The *Ankle* tests the machine’s behavior with an object featuring layer compression and tilts up to 22°. The *Car* is a model with different curvature directions and tilts up to 26°. For this model, a raft was required to reach the build volume (Fig. 7). Both the *Ankle* and *car* models show no significant deviations from the toolpaths. Exploiting the filament deposition direction allows for the creation of controlled

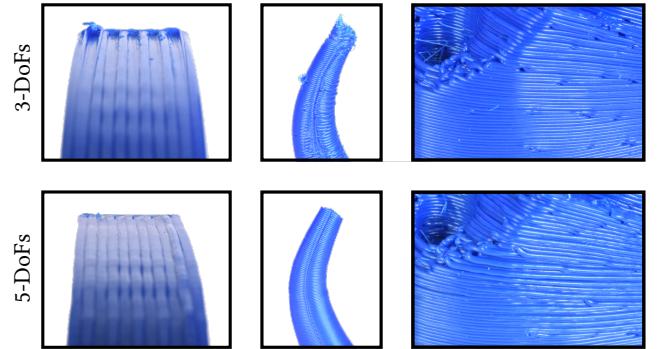


Figure 13: Comparison between 3-DoFs (top) and 5-DoFs (bottom). Printing with 3-DoFs results in noticeable gouging on the top of the *Bump*, which is avoided when using 5-DoFs (left). At higher tilting angles, the 3-DoFs configuration fails to print the *Horn*, whereas the 5-DoFs setup completes it successfully (center). Nevertheless, the machine demonstrates higher spatial accuracy when operating with only 3-DoFs, and the *Ankle* shows improved wall quality in this mode (right).

anisotropic appearances [Chermain et al. 2023]. However, with planar printers, this feature is limited to horizontal surfaces. The 3Z configuration lifts this limitation, enabling the fabrication of the *Yin-yang* model with tilts up to 14°.

4.7 3-DoFs Comparison

To evaluate the advantages of 5-axis printing, we compared prints produced by tilting the bed – *i.e.*, 5-DoFs – with those generated using a toolpath that follows the same sequence of positions \hat{p} but with fixed tool orientations $\hat{n} = e_3$ – *i.e.*, 3-DoFs (Fig. 13). Using 3-DoFs leads to severe gouging that not only significantly grooves the object’s surface, but also causes the nozzle to become stuck in the print. This often results in the object detaching from the build platform, requiring multiple attempts to successfully complete the print. Furthermore, highly tilted models, such as the *Horn*, could only be successfully printed using 5-DoFs. For *Mesh models*, the toolpath generator produces excessive orientation changes (see supplemental video). However, when using 5-DoFs, this behavior – combined with the increased error introduced by the added mechanical complexity – can lead to reduced wall quality compared to prints produced with 3-DoFs. This issue could be mitigated either by increasing the manufacturing tolerances of the machine or by using a toolpath generator that better accounts for the machine’s limitations.

4.8 Hardware Limitations

In this section, we discuss the limitations of the machine when used for non-planar printing. Understanding these limitations may be crucial for further refining the hardware for non-planar applications.

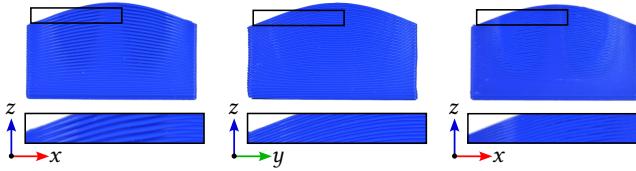


Figure 14: Mechanical backlash can introduce small geometric deviations from the prescribed toolpath. Left: The *Bump* is oriented toward the x -axis. In the inset, odd and even layers are offset inward and outward, respectively, creating misalignments. Center: The *Bump* is oriented toward the y -axis. The wall facing the viewer now has aligned layers, but the left and right walls are no longer straight. Right: The *Bump* is again oriented toward the x -axis, as in the left case, but the layer height is reduced to 0.2 mm instead of 0.4 mm. Reducing the layer height helps mitigate layer misalignment.

Z-Axes Speed. In the context of planar printing, the maximum speed of the z-axes is generally unimportant and, therefore, relatively low. For example, our Ratrig z-axes are limited to 1900 mm/min. However, in non-planar printing, this limits the maximum tilting speed of the bed. As a result, the deposition speed must be reduced in regions with high curvature or sharp corners to accommodate the slower bed rotation. Maintaining print quality under varying deposition speeds requires precise control to account for the nonlinear effects of pressure within the nozzle. In planar printing, this is typically handled using a technique called pressure advance. However, this technique is not available for non-planar printing and is challenging to implement with long nozzles, making it an interesting direction for future work.

Mechanical Backlash. The motion transfer between the z-axes and the bed is slightly affected by mechanical backlash. This can be problematic, as it may prevent the machine from accurately depositing material directly on top of the previous layer, causing misalignment (Fig. 14). Note that the error depends on the direction in which the bed is tilting, suggesting that static kinematic issues are unlikely to be the cause.

Rails-ball Connection. The balls are held on the rails by gravity alone. If the friction between the rails and the balls exceeds the force of gravity, the bed may lift instead of sliding smoothly along the rails. The bed lift may cause it to fall off the rails, leading to print failure. This issue can be avoided by greasing the rails. While lubrication proved to be an effective solution in our experiments, it may be beneficial to implement sensors to detect this type of failure and automatically halt the print. Since each rail consists of two metal rods and the ball makes contact with both (Fig.1), a simple sensor could be created by electrifying the rods: if the ball falls off, the circuit would open, signaling a failure.

5 Conclusion and Future Work

For its low-cost and simplicity compared to alternative technologies, the 3Z configuration presents a compelling option for non-planar FFF applications up to around 30°. By reducing the entry barrier to non-planar FFF, the 3Z configuration has the potential to enable a

wider community of makers, educators, and researchers to explore, innovate, and contribute to the expanding field of advanced additive manufacturing. The closed-form solutions presented in this work could be implemented in the controller firmware to ensure accurate interpolation. Nonetheless, several important aspects should be considered. Among others, objects to be printed should be automatically positioned to accommodate the limited build volume, and an automated method should be developed to mitigate inaccuracies caused by deviations in machine component dimensions. Additionally, pressure advance, input shaping, and other advanced techniques should also be adapted for the 3Z configuration to improve print quality. In particular, a more advanced algorithm is needed to coordinate extrusion flow, surface curvature, and bed-tilt to improve surface quality. Furthermore, we used a commercially available printer with only minor modifications. Optimizing the machine's dimensions and design specifically for non-planar applications represents an interesting area for further exploration. The generation of non-planar toolpaths remains an active and evolving area of research. Although numerous solutions have been proposed within the research community [Chermain et al. 2025; Liu et al. 2024; Zhang et al. 2022], a fully developed software package that matches the capabilities of existing planar slicers is still lacking and would require substantial engineering effort to realize.

Acknowledgments

The authors thank Pierre-Alexandre Hugron, who was crazy enough to believe that the RatRig could serve as a 5-axis 3D printer. We would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was supported by the French National Research Agency (ANR) under Grant No. ANR-24-CE10-6403 (<https://anr.fr/Projet-ANR-24-CE10-6403>), by the Grand Est Region (France) under Grant No. 25E00156671, and partially funded by the Inria Challenge DORNELL.

References

- Prahar M. Bhatt, Rishi K. Malhan, Aniruddha V. Shembekar, Yeo Jung Yoon, and Satyandra K. Gupta. 2020. Expanding capabilities of additive manufacturing through use of robotics technologies: A survey. *Additive Manufacturing* 31 (2020). doi:10.1016/j.addma.2019.100933
- Joshua Bird. 2025. Core R-Theta Printer. <https://github.com/jyblrd/Core-R-Theta-4-Axis-Printer>
- Yongxue Chen, Tianyu Zhang, Yuming Huang, Tao Liu, and Charlie C.L. Wang. 2025. Co-Optimization of Tool Orientations, Kinematic Redundancy, and Waypoint Timing for Robot-Assisted Manufacturing. *IEEE Transactions on Automation Science and Engineering* (2025). doi:10.1109/TASE.2025.3542218
- Xavier Chermain, Giovanni Cocco, Cédric Zanni, Eric Garner, Pierre-Alexandre Hugron, and Sylvain Lefebvre. 2025. Atomizer: Beyond Non-Planar Slicing for Fused Filament Fabrication. *Computer Graphics Forum (Proceedings of the Symposium on Geometry Processing)* (2025). doi:10.1111/cgf.70189
- Xavier Chermain, Cédric Zanni, Jonás Martínez, Pierre-Alexandre Hugron, and Sylvain Lefebvre. 2023. Orientable Dense Cyclic Infill for Anisotropic Appearance Fabrication. *ACM Trans. Graph. (Proc. SIGGRAPH)* 42, 4, Article 68 (2023), 13 pages. doi:10.1145/3592412
- Chengkai Dai, Charlie C. L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-Jin Liu. 2018. Support-free volume printing by multi-axis motion. *ACM Trans. Graph. (Proc. SIGGRAPH)* 37, 4, Article 134 (2018). doi:10.1145/3197517.3201342
- Jimmy Etienne, Nicolas Ray, Daniele Panozzo, Samuel Hornus, Charlie CL Wang, Jonás Martínez, Sara McMains, Marc Alexa, Brian Wyvill, and Sylvain Lefebvre. 2019. CurviSlicer: Slightly curved slicing for 3-axis printers. *ACM Trans. Graph.* 38, 4 (2019). doi:10.1145/3306346.3323022
- James Gardner, Tom Garabet, Nathaniel Kaill, R.I. Campbell, Guy Bingham, Daniel Engstrom, and Nicolae Balc. 2018. Aligning Material Extrusion Direction with Mechanical Stress via 5-Axis Tool Paths.

- Samad Hayati and M Mirmirani. 1985. Improving the absolute positioning accuracy of robot manipulators. *Journal of robotic systems* 2, 4 (1985), 397–413.
- Freddie Hong, Steve Hodges, Connor Myant, and David E Boyle. 2022. Open5x: Accessible 5-Axis 3D Printing and Conformal Slicing. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. doi:10.1145/3491101.3519782
- Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. FrameFab: robotic fabrication of frame shapes. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 35*, 6, Article 224 (2016). doi:10.1145/2980179.2982401
- Mohammed A. Isa and Ismail Lazoglu. 2019. Five-axis additive manufacturing of freeform models through buildup of transition layers. *Journal of Manufacturing Systems* 50 (2019), 69–80. doi:10.1016/j.jmsy.2018.12.002
- Nathaniel Kaill, R.I. Campbell, and Patrick Pradel. 2022. A Comparative Study Between 3-Axis and 5-Axis Additively Manufactured Samples and their Ability to Resist Compressive Loading. doi:10.26153/tsw/17440
- Tao Liu, Tianyu Zhang, Yongxue Chen, Yuming Huang, and Charlie C. L. Wang. 2024. Neural Slicer for Multi-Axis 3D Printing. *ACM Trans. Graph. (Proc. SIGGRAPH) 43*, 4, Article 85 (2024), 15 pages. doi:10.1145/3658212
- Ioanna Mitropoulou, Mathias Bernhard, and Benjamin Dillenburger. 2025. Curvature-informed paths for shell 3D printing. *Automation in Construction* 171 (2025), 105988. doi:10.1016/j.autcon.2025.105988
- Abhilash Nayak, Stéphane Caro, and Philippe Wenger. 2018. Comparison of 3-[PP]S Parallel Manipulators based on their Singularity Free Orientation Workspace, Parasitic Motions and Complexity. *Mechanism and Machine Theory* 129 (2018), 293–315. doi:10.1016/j.mechmachtheory.2018.08.001
- Nonplanar.xyz. 2025. Nozzles | Nonplanar.xyz. <https://www.nonplanar.xyz/nozzles>
- Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proc. CHI Conference on Human Factors in Computing Systems*. 887–896. doi:10.1145/2858036.2858106
- RatRig. 2025. RatRig V-Core 3.1. <https://docs.ratrig.com/product-details/v-core3-1>
- Lorenzo Sciaivice and Bruno Siciliano. 2012. *Modelling and control of robot manipulators*. Springer Science & Business Media.
- Alexander Teibrich, Stefanie Mueller, François Guimbretière, Robert Kovacs, Stefan Neubert, and Patrick Baudisch. 2015. Patching Physical Objects (*UIST '15*). 83–91. doi:10.1145/2807442.2807467
- Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2020. Jubilee: An Extensible Machine for Multi-tool Fabrication. In *Proc. CHI Conference on Human Factors in Computing Systems (CHI '20)*. doi:10.1145/3313831.3376425
- Guilin Yang, Tat Joo Teo, I-Ming Chen, and Wei Lin. 2011. Analysis and design of a 3-DOF flexure-based zero-torsion parallel manipulator for nano-alignment applications. In *2011 IEEE International Conference on Robotics and Automation*. 2751–2756. doi:10.1109/ICRA.2011.5979911
- Yuan Yao, Longyu Cheng, and Zhengyu Li. 2024. A comparative review of multi-axis 3D printing. *Journal of Manufacturing Processes* 120 (2024), 1002–1022. doi:10.1016/j.jmapro.2024.04.084
- Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, and Charlie C. L. Wang. 2022. S3-Slicer: A General Slicing Framework for Multi-Axis 3D Printing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 41*, 6, Article 277 (2022), 15 pages. doi:10.1145/3550454.3555516
- Tianyu Zhang, Tao Liu, Neelotpal Dutta, Yongxue Chen, Renbo Su, Zhizhou Zhang, Weiming Wang, and Charlie C.L. Wang. 2025. Toolpath generation for high density spatial fiber printing guided by principal stresses. *Composites Part B: Engineering* 295 (2025), 112154. doi:10.1016/j.compositesb.2025.112154