The International Journal of Advanced Manufacturing Technology (2024) 133:2219–2233
https://doi.org/10.1007/s00170-024-13970-5

ORIGINAL ARTICLE

Advanced CNC thread milling: a comprehensive canned cycle for efficient cutting of threads with fixed or variable pitch and radius

Sotiris Omirou[1] · **Marios Charalambides**[2] · **Charalambos Chasos**[1]

Received: 21 February 2024 / Accepted: 31 May 2024 / Published online: 7 June 2024

Abstract

This paper presents the design, implementation, and experimental validation of a novel canned cycle for CNC milling machines, enabling the precise and efficient cutting of threads with fixed or variable pitch and radius. Conventional canned cycles are limited to fixed pitch threads, restricting the versatility of CNC milling machines in thread machining applications. The development process involves integrating a sophisticated control algorithm into the CNC milling machine's software, giving the operator remarkable control over the thread cutting process. This algorithm allows the operator to choose between external or internal threads, set both initial and final radii, determine initial and final pitches, specify the number of turns, and select the left or right-hand thread type. Such flexibility enables the creation of threads with diverse geometries. Furthermore, the proposed canned cycle provides the capability to switch between roughing and finishing passes by adjusting the step motion along the prescribed helical curve.

Simulation tests conducted under various threading cases clearly demonstrate the efficiency of the proposed canned cycle. These results showcase its capability to address a wide range of machining scenarios, offering practical solutions applicable across a spectrum of applications.

Keywords CNC machining · Canned cycles · Thread cutting · Interpolation algorithms

Nomenclature

H	Height of the spiral, mm
m	Slope of the cone's lines with respect to the $x - y$ plane
n	Total number of turns
n_0	Angular velocity, rev/min
$n(t)$	Number of turns of a helix at time t
p_f	Final pitch of the helix, mm
p_i	Initial pitch of the helix, mm
$p(t)$	Pitch of the helix at time t, mm
R_f	Final radius of the helix, mm
R_i	Initial radius of the helix, mm
$r(t)$	Radius of the helix at time t, mm
T	Time required for traversing a complete revolution, min
t	Time variable, min
t_f	Time it takes the cutting tool to complete the process, min
$x(t)$	x – Axis position of the cutting tool at time t, mm
$y(t)$	y – Axis position of the cutting tool at time t, mm
z_0	Calibration constant for the initial position on the z axis
$z(t)$	z – Axis position of the cutting tool at time t, mm
τ	Normalized time

1 Introduction

The evolution of Computer Numerical Control (CNC) machining has brought about a revolution in manufacturing processes, not only enhancing precision and efficiency but also addressing the challenges posed by complex geometries and demanding machining applications. This paper deals with the domain of CNC thread milling, introducing an innovative approach through the development and implementation of a comprehensive canned cycle.

A canned cycle is a predefined machining operation that entails a sequence of machine movements to perform various machining tasks like drilling, pocketing, slotting, boring, and

✉ Sotiris Omirou
eng.os@frederick.ac.cy

1 Department of Mechanical Engineering, Frederick University, Nicosia, Cyprus

2 Department of Business Administration, Center of Sciences, Frederick University, Nicosia, Cyprus

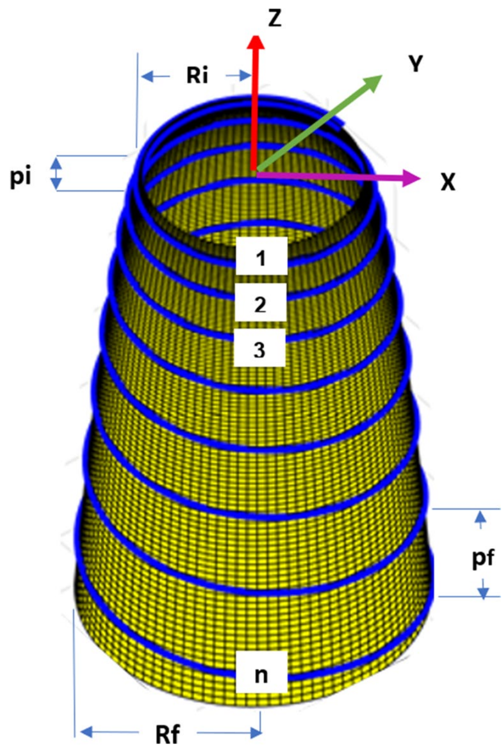


Fig. 1 Helix with variable radius and pitch

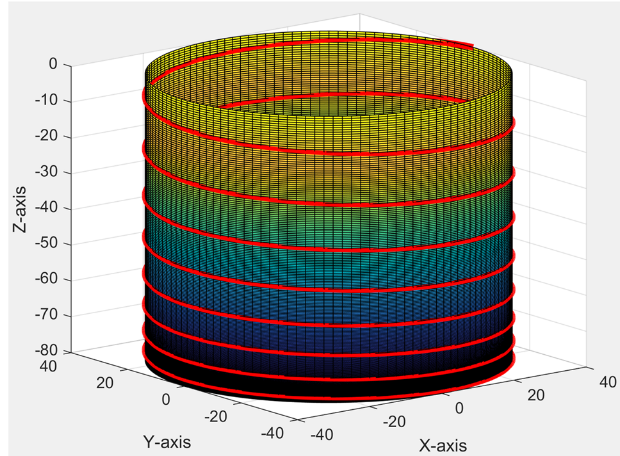


Fig. 3 Helix with constant radius and variable decreasing pitch ($R_i=R_f=40$, pi = 16, pf = 4, n = 8)

tapping. It aims to simplify programming by consolidating multiple commands into a single block, utilizing G-code to define machining operations that would typically necessitate multiple blocks [1, 2]. Continuous efforts are made by CNC manufacturers to incorporate new canned cycles into their controllers, specifically designed to handle intricate and complex geometries [3, 4]. This is easily concluded by comparing the canned cycles available nowadays with the older ones.

Centered on programming and facilitating the precise and efficient cutting of threads within a user-friendly environment, incorporating both fixed and variable pitch and radius, the presented canned cycle addresses limitations posed by conventional canned cycles, which primarily target fixed pitch threads. In the precise domains of engineering and manufacturing, the use of variable pitch threads is necessitated by diverse potential applications. These applications encompass manufacturing components for automotive vehicle engines, fasteners for aerospace systems, industrial

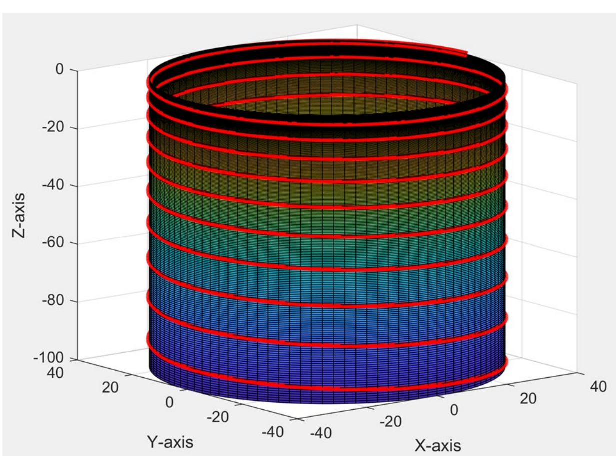


Fig. 2 Helix with constant radius and variable increasing pitch ($R_i=R_f=40$, pi = 4, pf = 16, n = 10)

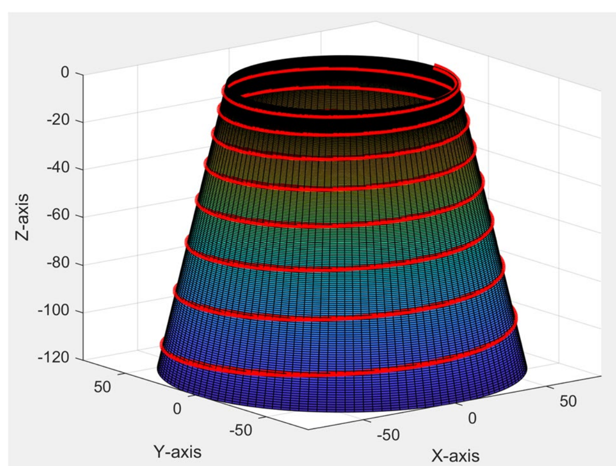


Fig. 4 Helix with variable radius and variable increasing pitch (Ri = 50, Rf = 80, pi = 5, pf = 25, n = 8)

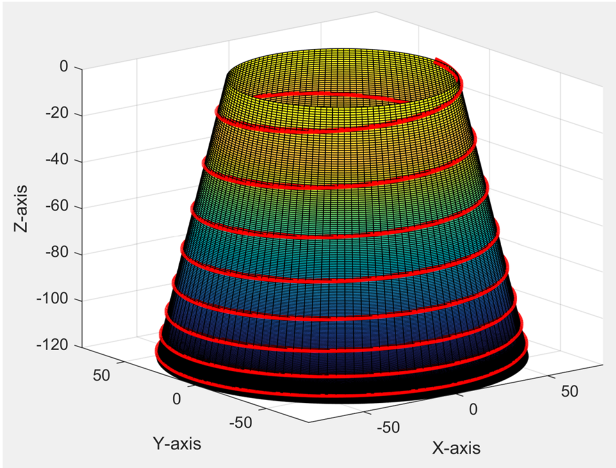


Fig. 5 Helix with variable radius and variable decreasing pitch (Ri = 50, Rf = 80, pi = 25, pf = 5, n = 8)

devices, robots manufacturing parts [5–8] and the creation of medical implants [9–12]. The mentioned references constitute a sample of recent related research work.

The foundation of work's innovation lies in the integration of a sophisticated control algorithm into the CNC milling machine's software. This integration empowers operators with an unprecedented level of control over the thread cutting process. Operators can seamlessly choose between external or internal threads, define initial and final radii, determine initial and final pitch, specify the number of turns, and even select the thread's direction. This flexibility allows for the creation of threads with diverse geometries, significantly broadening the applicability of CNC milling machines in thread machining applications. Furthermore, the proposed canned cycle provides the means to switch between roughing and finishing passes. This is achieved through precise regulation of the step motion along the prescribed helical curve, adding another layer of adaptability to the machining process.

To validate the effectiveness of the approach, extensive simulation tests have been conducted under various threading cases. The results clearly demonstrate the efficiency of the proposed canned cycle, showcasing its capability to address a wide spectrum of machining scenarios.

The present study represents an evolution of prior research [13], wherein plans were discussed to extend the scope by incorporating cases with variable radius and pitch, along with predefined initial and final radii and pitches, with the number of turns also pre-selected. The current work builds upon and extends the findings from the previous research, resulting in a significant expansion of the capabilities of the proposed new canned cycle.

In the following sections, the paper presents the mathematical formulation of helix equations (Section 2), CNC parametric programming (Section 3), and the design and formulation of the interpolation algorithm (Sections 4 and 5). Implementation issues (Section 6) and comprehensive test results (Section 7) are then discussed. In the conclusion (Section 8), the paper highlights the innovative canned cycle's contribution to advancing CNC thread milling.

2 Mathematical formulation of helix equations

To facilitate the cutting of threads with constant or variable radius and pitch on a 3-axis CNC milling machine, it is crucial to have the ability to generate 3D motion along a helix

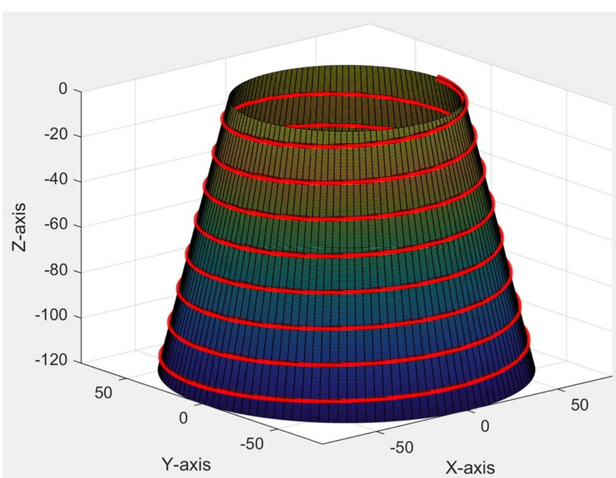


Fig. 6 Helix with variable radius and constant pitch (Ri = 50, Rf = 80, pi = 15, pf = 15, n = 8)

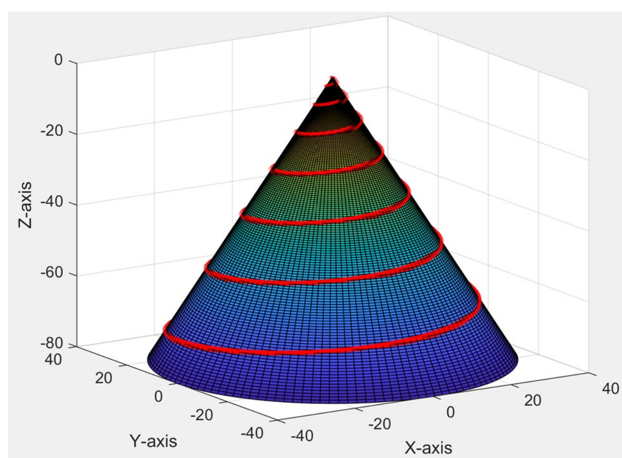


Fig. 7 Helix with initial radius set to zero and variable Increasing pitch (Ri = 0, Rf = 40, pi = 0, pf = 20, n = 7)

Table 1 Heidenhain Control System Parameter Guide

D CODE	FUNCTION	EXAMPLE	EXPLANATION
D00	ASSIGNMENT	D00 Q1 P01 20	Assignment of the value 20 to parameter Q1
D01	ADDITION	D01 Q2 P01 Q3 P02 Q4	The result of (Q3 + Q4) is assigned to Q2
D02	SUBTRACTION	D02 Q5 P01 Q6 P02 8	The result of (Q6 – 8) is assigned to Q5
D03	MULTIPLICATION	D03 Q8 P01 Q9 P02 -2	The result of (–2)(Q9) is assigned to Q8
D04	DIVISION	D04 Q10 P01 Q11 P02 4	The result of Q11/4 is assigned to Q10
D05	SQUARE ROOT	D05 Q12 P01 Q13	The square root of Q13 is assigned to Q12
D06	SINE	D06 Q14 P01 Q15	The result of sin(Q15) is assigned to Q14
D07	COSINE	D07 Q16 P01 Q17	The result of cos(Q17) is assigned to Q16
D09	IF EQUAL, JUMP	D09 P01 Q20 P02 5 P03 1	If Q20 is equal to 5, jump to label 1
D10	IF NOT EQUAL, JUMP	D10 P01 Q21 P02 Q22 P03 2	If Q21 is not equal to Q22, jump to label 2
D11	IF GREATER, JUMP	D11 P01 Q23 P02 0 P03 3	If Q23 is greater than 0, jump to label 3
D12	IF LESS, JUMP	D12 P01 Q24 P02 0 P03 4	If Q24 is less than 0, jump to label 4
D13	CALCULATE ANGLE	D13 Q30 P01 Q25 P02 Q26	The result of arctan(Q25/Q26) is assigned to Q30

with the specific characteristics. This involves solving the parametric equations x(t), y(t), z(t) of the helix within the framework of an interpolation algorithm.

When examining a conical helix, which is a helix situated on the curved surface of a cone, its parametric equations can be formulated as follows [14, 15]

$$x(t) = r(t)\cos(2\pi n(t)), \tag{1}$$

$$y(t) = r(t)\sin(2\pi n(t)), \tag{2}$$

$$z(t) = z_0 + mr(t) \tag{3}$$

where,

t_f the time it takes the cutting tool to complete the process.

t the time variable, with $0 \leq t \leq t_f$.

r(t)- the radius of the helix at time t.

n(t)- the number of turns of a helix at time t.

m- the slope of the cone's lines with respect to the $x - y$ plane.

z_0- a constant.

In the following stage of the modeling process, the definition of a function, labeled as r(t), is undertaken to embody ideal construction specifications. The preferred input parameters for the model, as illustrated in Fig. 1, include:

R_i the initial radius of the helix.

Table 2 Summary of thread features along with the corresponding assigned input parameters (Q)

Feature	Parameter 'Q'	Assigned value
external or internal thread	Q1	'0' for external, '1' for internal
left-hand or right-hand thread	Q2	'0' for left-hand, '1' for right-hand
tool radius	Q3	r (mm)
initial thread radius	Q4	Ri (mm)
final thread radius	Q5	Rf (mm)
initial thread pitch	Q6	pi (mm)
final thread pitch	Q7	p_f (mm)
number of turns	Q8	n (an integer number)
normalized time	Q9	$\boldsymbol{\tau}$ (a real number)
feedrate	Q10	F (mm/min)
spindle speed	Q11	S (rpm)

R_f the final radius of the helix.

p_i the initial pitch of the helix.

p_f the final pitch of the helix.

n the number of turns.

Then $r(t)$ should be calibrated accordingly accommodating these special parameter values. Assuming linear increase of the pitch from p_i to p_f, $p(t)$ can be expressed in the form:

$$p(t) = p_i + (p_f - p_i)\frac{t}{t_f}. \tag{4}$$

If T represents the time required for traversing a complete revolution, then $z(t)$ at time t is given by:

$$\begin{aligned} z(t) &= \frac{1}{T}\int_0^t p(s)ds = \frac{1}{T}\int_0^t \left(p_i + (p_f - p_i)\frac{t}{t_f}\right)ds \\ &= \frac{p_i t}{T} + \frac{(p_f - p_i)t^2}{2Tt_f} + C \end{aligned} \tag{5}$$

Supposing that $z(0) = 0$, then obtain that $C = 0$. As a result:

$$z(t) = \frac{p_i t}{T} + \frac{(p_f - p_i)t^2}{2Tt_f}. \tag{6}$$

Assuming that the helix is traversed with constant angular velocity, n_0 (rev/min), implies that $T = \frac{1}{n_0}$, and $t_f = \frac{n}{n_0}$, which gives that $\frac{T}{t_f} = \frac{1}{n}$. Given this, Eq. (6) simplifies to:

$$z(t) = np_i\frac{t}{t_f} + \frac{n(p_f - p_i)}{2}\left(\frac{t}{t_f}\right)^2. \tag{7}$$

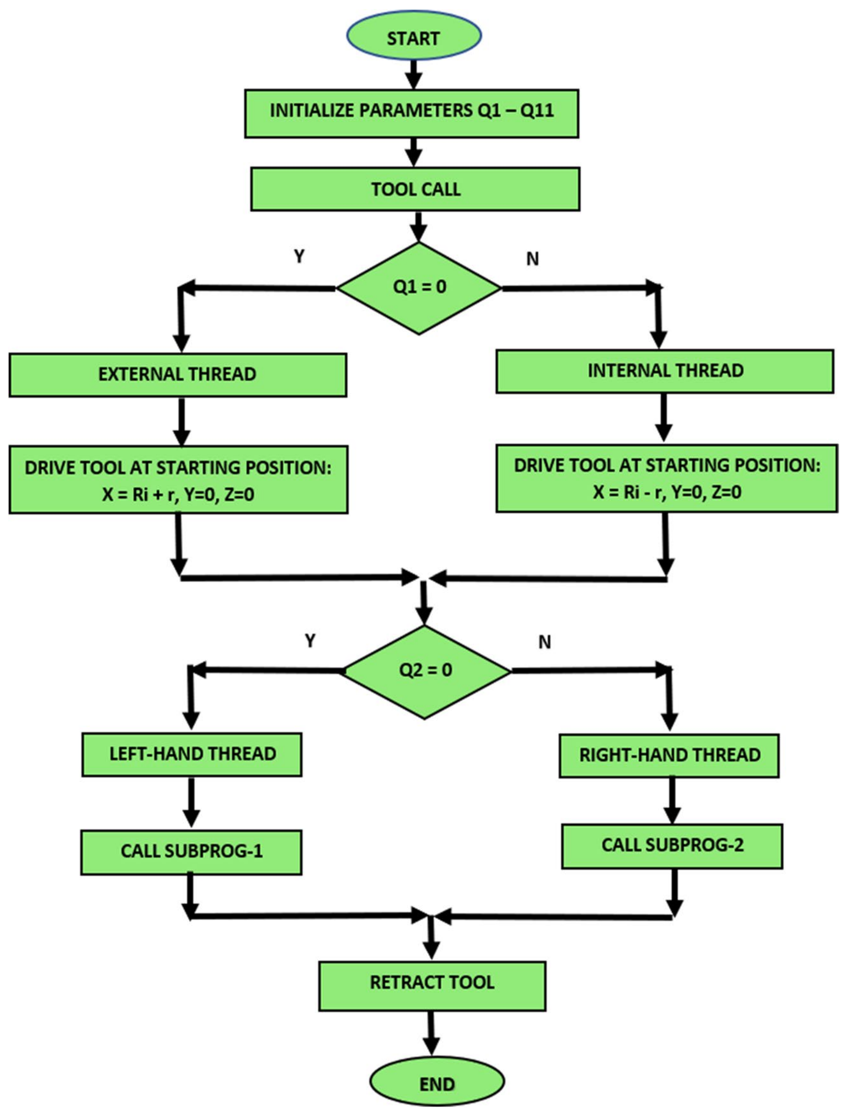


Fig. 8 Main program flowchart

Equation (3) suggests that the radius $r(t)$ can then be expressed as:

$$r(t) = \frac{1}{m}\left(z(t) - z_0\right) = \frac{1}{m}\left(np_i\frac{t}{t_f} + \frac{n(p_f - p_i)}{2}\left(\frac{t}{t_f}\right)^2 - z_0\right). \tag{8}$$

Given that $r(0) = R_i$ and that $r\left(t_f\right) = R_f$, obtain that:

$$z_0 = -mR_i, \text{ with } m = \frac{n\left(p_f + p_i\right)}{2\left(R_f - R_i\right)} \tag{9}$$

A simple calculation then gives that:

$$r(t) = R_i + \frac{2p_i\left(R_f - R_i\right)}{p_f + p_i}\frac{t}{t_f} + \frac{(p_f - p_i)\left(R_f - R_i\right)}{p_f + p_i}\left(\frac{t}{t_f}\right)^2 \tag{10}$$

Given the constant angular velocity n_0, one can also derive $n(t) = n_0 t$. Thus, the corresponding equations of motion are given by:

$$x(t) = \left(R_i + \frac{2p_i\left(R_f - R_i\right)}{p_f + p_i}\frac{t}{t_f} + \frac{(p_f - p_i)\left(R_f - R_i\right)}{p_f + p_i}\left(\frac{t}{t_f}\right)^2\right)\cos\left(2\pi n_0 t\right), \tag{11a}$$

$$y(t) = \left(R_i + \frac{2p_i\left(R_f - R_i\right)}{p_f + p_i}\frac{t}{t_f} + \frac{(p_f - p_i)\left(R_f - R_i\right)}{p_f + p_i}\left(\frac{t}{t_f}\right)^2\right)\sin\left(2\pi n_0 t\right), \tag{11b}$$

$$z(t) = np_i\frac{t}{t_f} + \frac{n(p_f - p_i)}{2}\left(\frac{t}{t_f}\right)^2 \tag{11c}$$

Since $0 \le t \le t_f$, dividing by t_f gives that $0 \le \frac{t}{t_f} \le 1$ and thus define $\tau = \frac{t}{t_f} = \frac{n_0 t}{n}$. Thus, the equations of motion in terms of the new variable τ with $0 \le \tau \le 1$, can be written as follows:

$$x(\tau) = \left(R_i + \frac{2p_i\left(R_f - R_i\right)}{p_f + p_i}\tau + \frac{(p_f - p_i)\left(R_f - R_i\right)}{p_f + p_i}\tau^2\right)\cos(2\pi n\tau), \tag{12a}$$

$$y(\tau) = \left(R_i + \frac{2p_i\left(R_f - R_i\right)}{p_f + p_i}\tau + \frac{(p_f - p_i)\left(R_f - R_i\right)}{p_f + p_i}\tau^2\right)\sin(2\pi n\tau), \tag{12b}$$

$$z(\tau) = np_i\tau + \frac{n(p_f - p_i)}{2}\tau^2. \tag{12c}$$

Notice that the final height of the workpiece H is given by $H = z(1) = n\frac{p_f + p_i}{2}$.

It can be easily shown that these equations are also valid for $R_i = R_f$ as the equations of motion reduce to:

$$x(\tau) = R_i\cos(2\pi n\tau), \tag{13a}$$

$$y(\tau) = R_i\sin(2\pi n\tau), \tag{13b}$$

$$z(\tau) = np_i\tau + \frac{n(p_f - p_i)}{2}\tau^2. \tag{13c}$$

Moreover, swapping the sine and cosine functions in the initial two Eqs. (12a) and (12b), allows for either a left-hand or right-hand thread type. The validation of Eqs. (12)

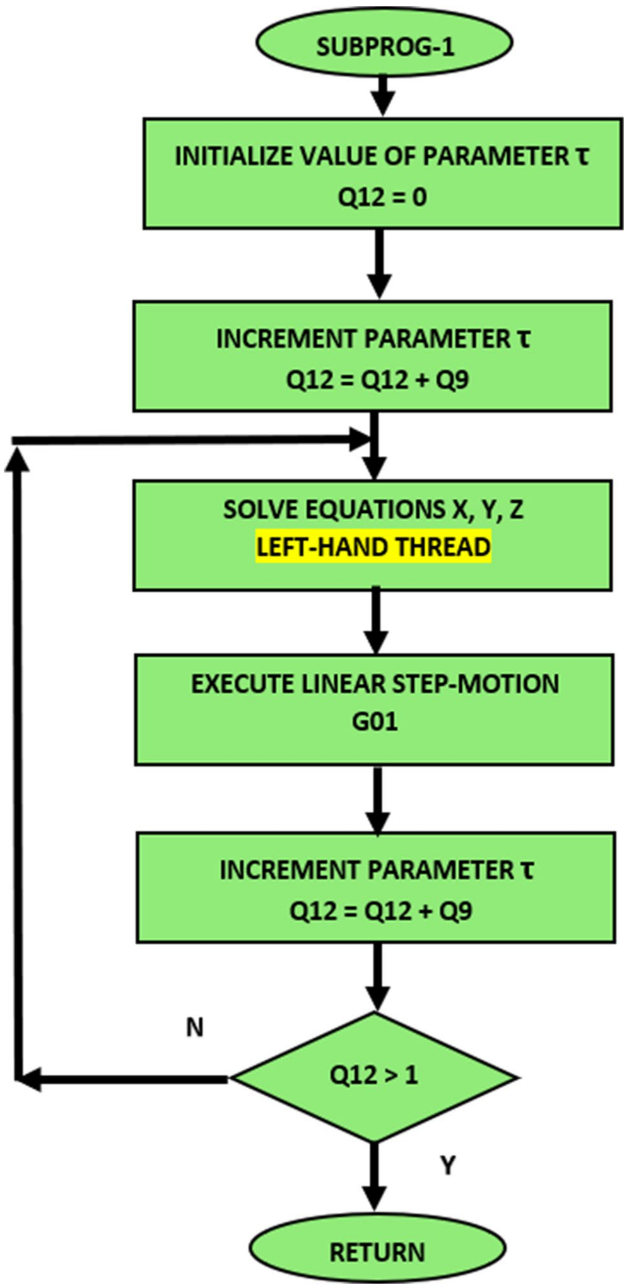


Fig. 9 Subprogram flowchart

is illustrated in the plots of Figs. 2, 3, 4, 5, 6 and 7 using representative values for pitch, radius and number of turns.

3 CNC parametric programming

CNC parametric programming serves as a flexible platform accessible in widely adopted controllers, known by various names depending on the provider controller. Fanuc custom macro B, Fadal macro, Okuma user task, Sinumeric and Heidenhain parametric technique are among the most widely used. Numerous research studies [16–20] have utilized this platform to resolve complex machining cases. This section aims to provide a detailed explanation of Heidenhain's parametric programming, the chosen programming language for this study. For a comprehensive understanding of the language, readers are directed to Lynch's in-depth book on the subject [21].

Within the Heidenhain control system, parameters are denoted by the letter Q followed by an integer. These parameters can be assigned numerical values directly or via arithmetic or trigonometric operations. Furthermore, the Q parameters can undergo dynamic updates during program execution and continuous checks to determine adherence to specific conditions, thereby influencing the program flow. In addition to the Q parameters, the letter D is employed to encode various functions, with specific functions represented by numbers 1 to 13 (refer to Table 1). The table presented herein has been adopted from the reference [13] mentioned earlier in the introduction. To illustrate the language's structure, the table provides examples and explanations for each individual D code. It is significant to emphasize that Q parameters and fixed numerical values can coexist within the same function, allowing enhanced flexibility in programming.

4 Design of the interpolation algorithm

The interpolation algorithm is designed to handle a broad spectrum of thread geometries under desired cutting conditions. In this regard, the operator is empowered with the

Table 3 List of G-code for the main program

Main program (Heidenhain G-code)	Comments
N0 ;THREAD CYCLE	Initialize thread data and cutting conditions
N10 D00 Q1 P01 +0*	(Parameters Q1-Q11)
N20 D00 Q2 P01 +0*	
N30 D00 Q3 P01 +5*	
N40 D00 Q4 P01 +40*	
N50 D00 Q5 P01 +80*	
N60 D00 Q6 P01 +5*	
N70 D00 Q7 P01 +25*	
N80 D00 Q8 P01 +8*	
N90 D00 Q9 P01 +0.001*	
N100 D00 Q10 P01 +300*	
N110 D00 Q11 P01 +600*	
N120 T1 G17 SQ11*	Tool call – Start rotation CW
N130 M3*	
N140 G90 G00 X+0 Y+0 Z+20*	Tool approach
N150 D09 P01 +Q1 P02 +1 P03 1*	Check the type of thread (internal or external)
N160 D01 Q4 P01 +Q3 P02 +Q4*	Compute coordinates of starting point
N170 D01 Q5 P01 +Q3 P02 +Q5*	
N180 L2*	
N190 G98 L1*	
N200 D02 Q4 P01 +Q4 P02 +Q3*	
N210 D02 Q5 P01 +Q5 P02 +Q3*	
N220 G98 L2*	
N230 D09 P01 +Q2 P02 +1 P03 3*	Check direction of thread (left or right) and direct the flow to one of the 2 subprograms
N240 G90 G00 X+Q4 Y+0 Z+20*	
N250 G01 Z+0 FQ10*	
N260 % SUBPROG_1.I	Subprogram-1: Right-hand
N270 L4*	
N280 G98 L3*	
N290 G90 G00 X+0 Y+Q4 Z+20*	
N300 G01 Z+0 FQ10*	
N310 % SUBPROG_2.I	Subprogram-2: Left-hand
N320 G98 L4*	
N330 G00 X+0 Y+0*	Retract tool
N340 G00 Z+20*	
N350 M30*	Program end

Table 4 List of G-code for the Subprogram-1

Subprogram-1 (Heidenhain G-code)	Comments
%SUBPROG_1 G71 *	
N10 D00 Q12 P01 +0*	Initialize τ.
N20 D01 Q12 P01 +Q12 P02 +Q9*	Update τ
N30 G98 L5*	Set a label (Head of the loop)
N40 D02 Q13 P01 +Q5 P02 +Q4*	Solve equations for **left-hand thread**.
N50 D03 Q14 P01 +2 P02 +Q6*	
N60 D03 Q15 P01 +Q13 P02 +Q14*	
N70 D03 Q16 P01 +Q15 P02 +Q12*	
N80 D01 Q17 P01 +Q7 P02 +Q6*	
N90 D04 Q18 P01 +Q16 P02 +Q17*	
N100 D01 Q19 P01 +Q18 P02 +Q4*	
N110 D02 Q20 P01 +Q7 P02 +Q6*	
N120 D03 Q21 P01 +Q20 P02 +Q13*	
N130 D03 Q22 P01 +Q12 P02 +Q12*	
N140 D03 Q23 P01 +Q21 P02 +Q22*	
N150 D04 Q24 P01 +Q23 P02 +Q17*	
N160 D01 Q25 P01 +Q24 P02 +Q19*	
N170 D03 Q26 P01 +360 P02 +Q8*	
N180 D03 Q27 P01 +Q26 P02 +Q12*	
N190 D07 Q28 P01 +Q27*	
N200 D03 Q50 P01 +Q25 P02 +Q28*	
N210 D06 Q29 P01 +Q27*	
N220 D03 Q51 P01 +Q25 P02 +Q29*	
N230 D03 Q30 P01 +Q8 P02 +Q6*	
N240 D03 Q31 P01 +Q30 P02 +Q12*	
N250 D03 Q32 P01 +Q8 P02 +Q20*	
N260 D04 Q33 P01 +Q32 P02 +2*	
N270 D03 Q34 P01 +Q33 P02 +Q22*	
N280 D01 Q52 P01 +Q31 P02 +Q34*	
N290 G01 X+Q50 Y+Q51 Z-Q52 FQ10*	Execute a linear 3D motion.
N300 D01 Q12 P01 +Q12 P02 +Q9*	Update τ
N310 D12 P01 +Q12 P02 +1 P03 5*	If τ <1 repeat the loop, otherwise return to Main

capability to define any of the features outlined in Table 2. Each feature is correspondingly assigned to a Q parameter, constituting the input values for the algorithm.

The development of the G-code algorithm is achieved through a main program and two subprograms. The main program initiates thread data by configuring parameters Q1-Q11 (refer to Table 2), positions the cutter at the starting point, and directs the execution flow to one of the two subprograms. These subprograms are formulated to address either a right-hand or a left-hand thread. The structure of the

Fig. 10 A single-form thread cutter

Fig. 11 External thread milling

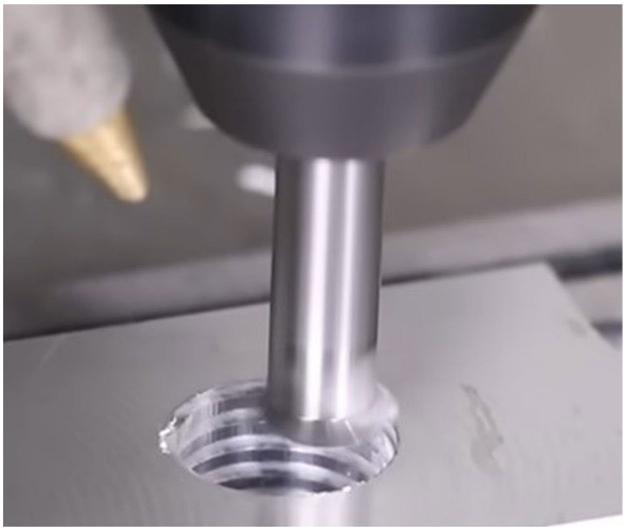


Fig. 12 Internal thread milling

main program and the two subprograms is depicted in the flowcharts of Figs. 8 and 9 respectively.

While the flowchart of the subprogram (Fig. 9) illustrates the scenario of a right-hand thread, a similar flowchart can be easily created for the left-hand thread case by interchanging the sine and cosine functions in the first two Eqs. (12a) and (12b).

5 Formulation of the G-code algorithm

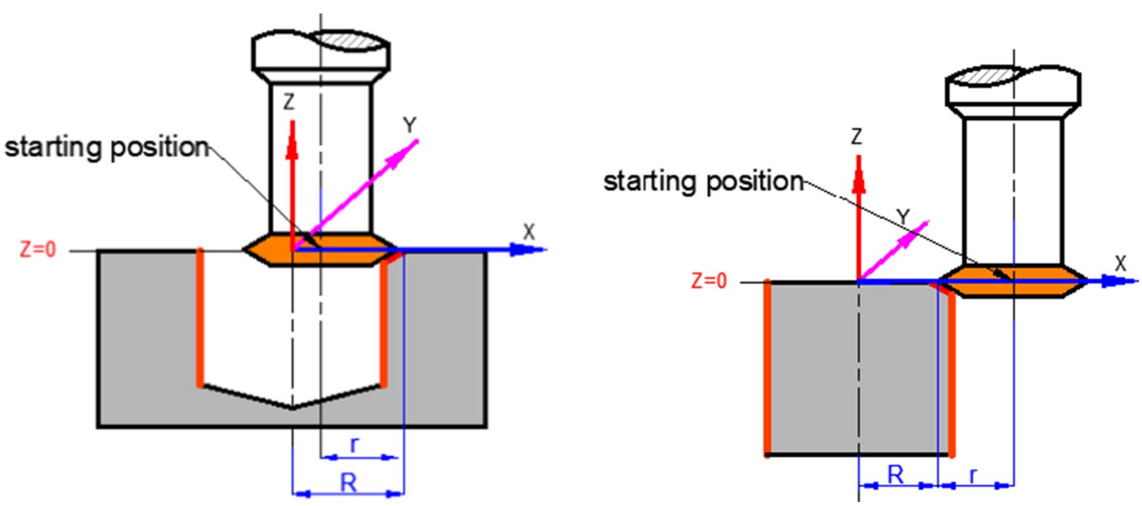
The G-code algorithm is developed within the Heidenhain control framework. Furthermore, it can be effortlessly adapted for integration with any other CNC control system supporting parametric programming. Entering the necessary thread data and cutting conditions into the G-code algorithm is achieved through the utilization of Q parameters, as detailed in Section 4.

The flowcharts presented in Figs. 8 and 9 serve as the foundation for developing the G-code parametric algorithm, illustrating the logical structure of both the main program and the two subprograms. The corresponding code, along with accompanying comments, can be found in Tables 3 and 4. Table 3 outlines the main program, while Table 4 details subprogram 1. Subprogram 2 maintains the same structure but exchanges the sine and cosine functions in the first two Eqs. (12a and 12b).

6 Implementation

6.1 Practical aspects

A suitable cutter for threading on a CNC milling machine is the single form cutter depicted in Fig. 10. Figures 11 and 12 illustrate the same cutter during the milling process for an external and internal thread respectively.



(a) Internal threading **(b)** External threading

Fig. 13 Thread milling – Case: constant radius

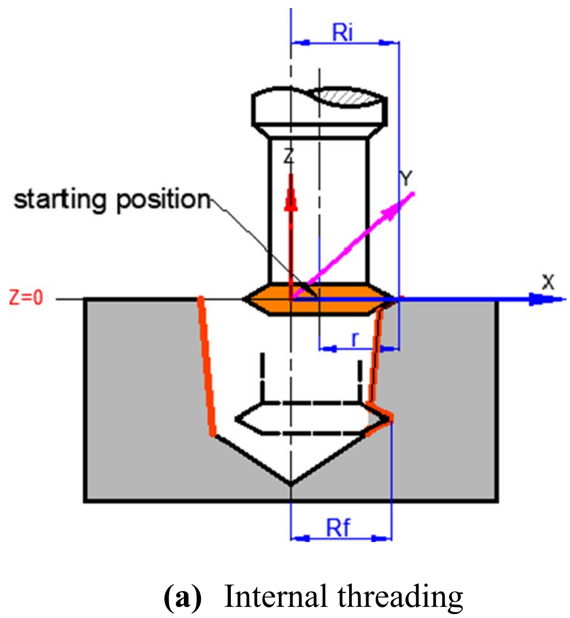


Fig. 14 Thread milling – Case: variable radius (**a**) Internal threading (**b**) External threading

The zero-reference point is located at the upper surface of the workpiece, aligning with the center of the helix. The algorithm directs the rotating cutter differently based on the thread type, whether internal or external. For internal threads, the cutter is moved linearly to the starting position with coordinates X = Ri – r, Y = 0, Z = 0. Alternatively, for external threads, the starting position is at coordinates X = Ri + r, Y = 0, Z = 0. The starting position is indicated in Figs. 13 and 14 for cases involving a constant and variable radius, respectively.

After positioning the cutter at the starting position, the algorithm directs the flow to one of the two subprograms based on the desired thread direction. Upon completion of the machining process, the tool first retracts from the workpiece before moving to a safe height above the top surface.

6.2 G-codification

Typically, canned cycles are standardized and encoded using a G-code accompanied by a set of parameters. It's essential to highlight that CNC controllers deliberately reserve certain G-codes for potential future applications, enabling customized implementations and enhanced flexibility. Specifically, within the Heidenhain control

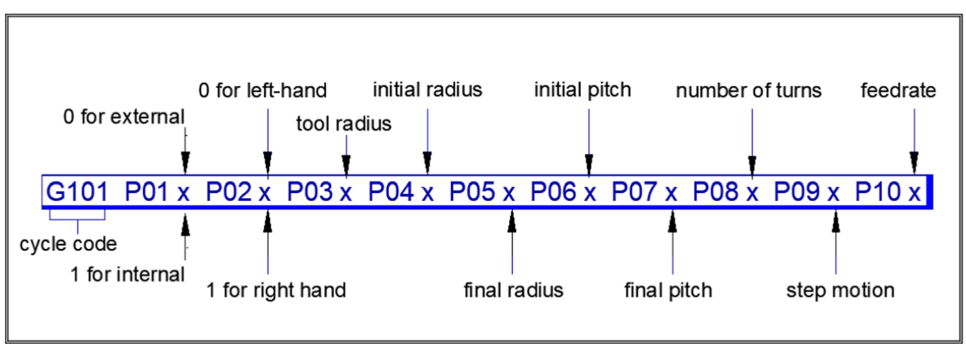


Fig. 15 Structure of the proposed thread canned cycle

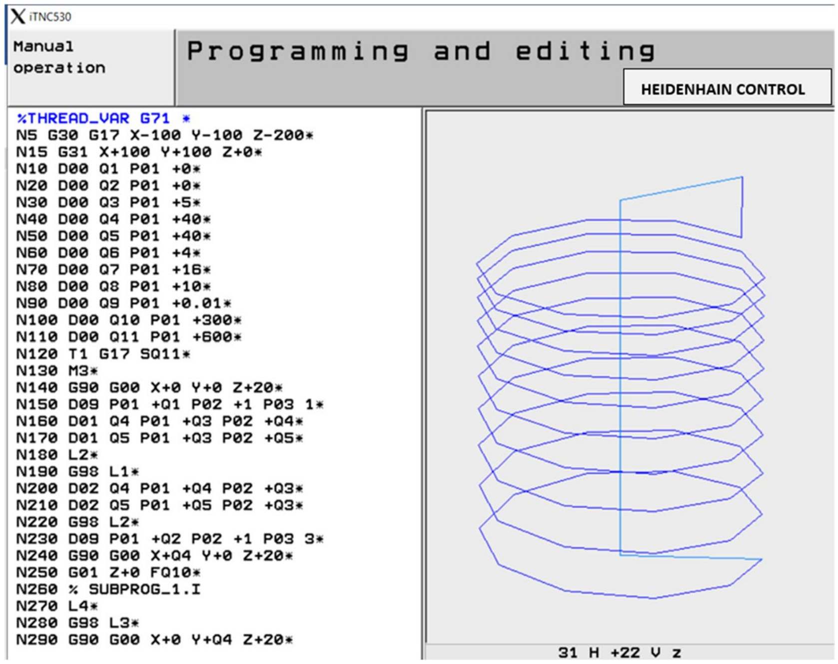


Fig. 16 TEST-1: Left-hand thread on constant radius with increasing pitch ($R_i = R_f = 40$, pi = 4, pf = 16, n = 10, precision $\tau = 0.01$)

framework, the unassigned G101 code has been chosen for the proposed implementation. The required supplemental data for the canned cycle can be conveyed through parameters P01 to P10 in a statement following the structure given in Fig. 15.

7 Test results

Simulation tests (depicted in Figs. 16, 17, 18, 19, 20, 21, 22 and 23) have been carefully chosen to showcase the efficiency of the proposed canned cycle, encompassing all

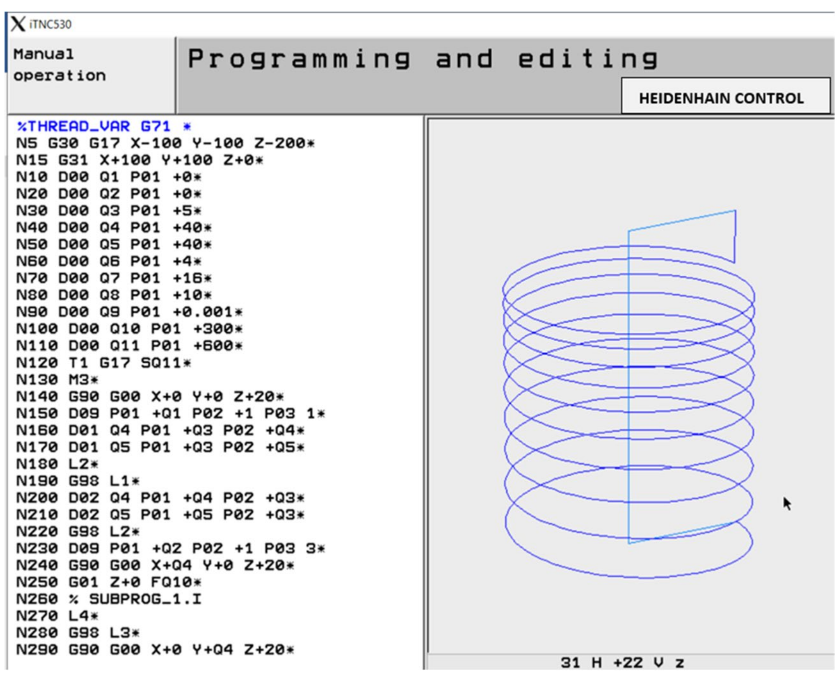


Fig. 17 TEST-2: Left-hand thread on constant radius with increasing pitch ($R_i = R_f = 40$, pi = 4, pf = 16, n = 10, precision $\tau = 0.001$)

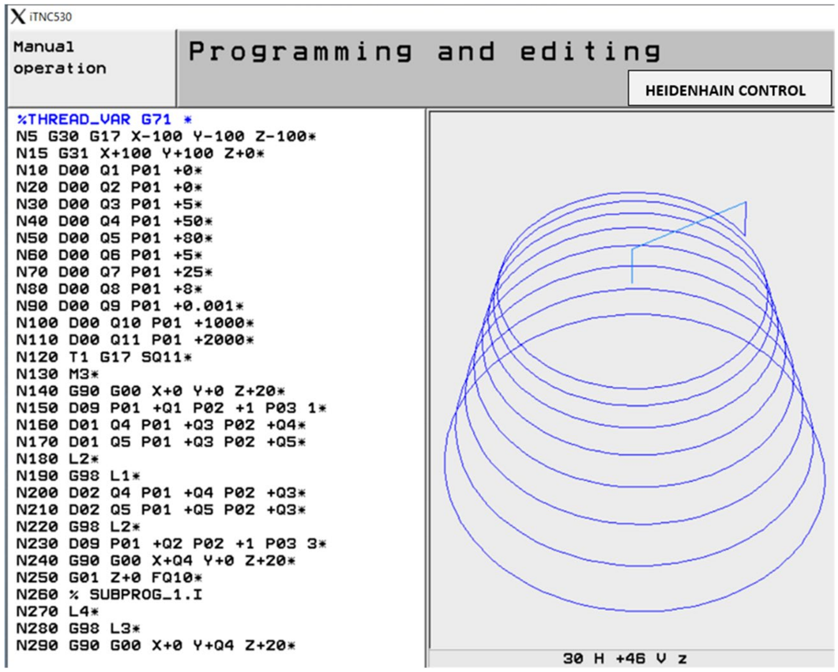


Fig. 18 TEST-3: Left-hand thread on variable radius with increasing pitch ($R_i = 50$, $R_f = 80$, pi = 5, pf = 25, n = 8, precision $\tau = 0.001$)

possible threading cases it is capable of handling. TEST-1 and TEST-2 simulate the same threading scenario but with different precision steps, offering a valuable capability for both roughing and finishing passes. TEST-3 represents a left-hand thread with a variable radius and increasing pitch, while TEST-4 demonstrates a right-hand thread with a variable radius and decreasing pitch. TEST-5 maintains the settings of TEST-4 but incorporates an increased number of turns. TEST-6 focuses on a left-hand thread with a variable radius and constant pitch. TEST-7 examines a case of a left-hand thread with an initial radius set to zero and a variable increasing pitch. While all previous tests relate to external threads, the final TEST-8 illustrates a case of an internal thread. In all instances, the traced path displayed in the graphical simulation corresponds to the center of the cutting tool.

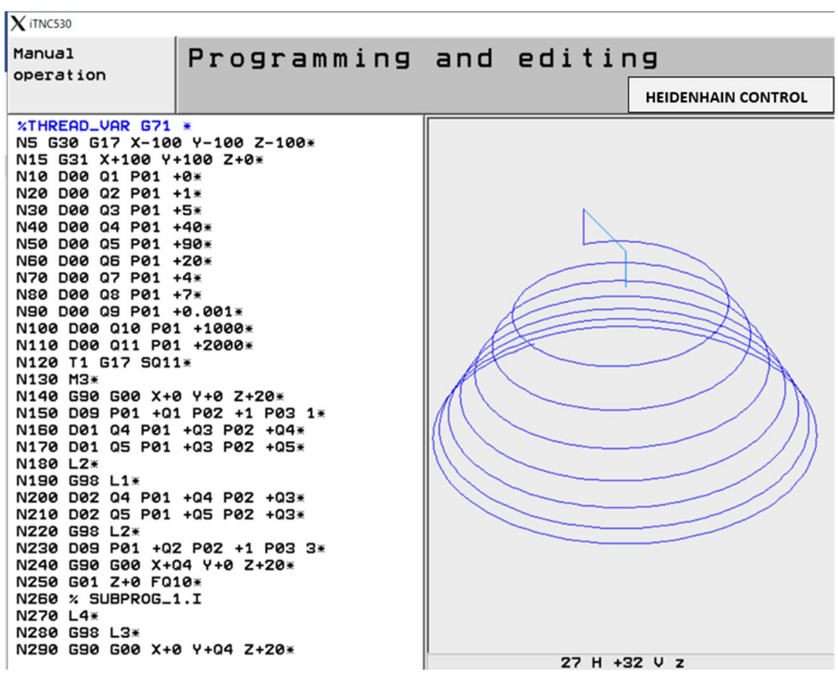


Fig. 19 TEST-4: Right-hand thread on variable radius with decreasing pitch ($R_i = 40$, $R_f = 90$, pi = 20, pf = 4, n = 7, precision $\tau = 0.001$)

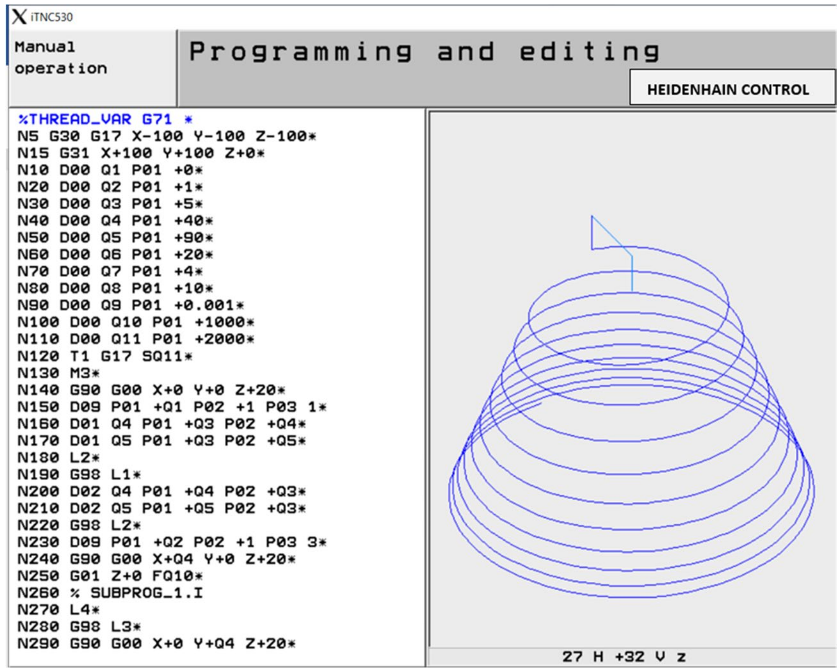


Fig. 20 TEST-5: Right-hand thread on variable radius with decreasing pitch ($R_i = 40$, $R_f = 90$, pi = 20, pf = 4, n = 10, precision $\tau = 0.001$)

8 Conclusions

The work presented in this paper built upon the foundations laid in previous mentioned research. As predicted in that earlier work, the present study advanced and extended the capabilities of the thread canned cycle to encompass threads with predefined start and ending radii, start and ending pitches, and the desired number of turns, applicable to both left- or right-hand, internal or external threads. Moreover, the proposed canned cycle seamlessly transitions between roughing and finishing passes, allowing precise regulation of step motion along the prescribed helical

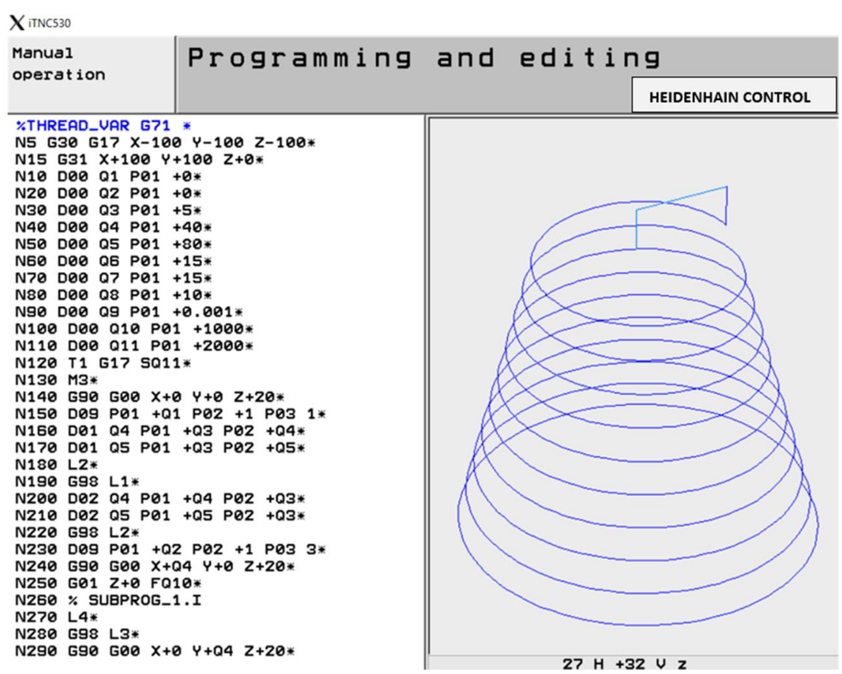


Fig. 21 TEST-6: Left-hand thread on variable radius with constant pitch ($R_i = 40$, $R_f = 80$, pi = 15, pf = 15, n = 10, precision $\tau = 0.001$)

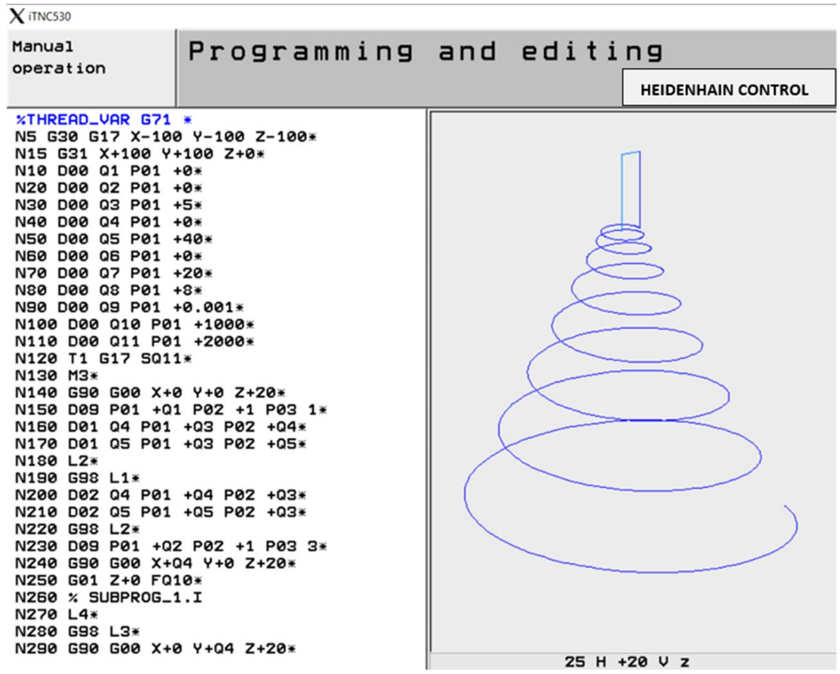


Fig. 22 TEST-7: Left-hand thread with initial radius set to zero and increasing pitch ($R_i = 0$, $R_f = 40$, pi = 0, pf = 20, n = 8, precision $\tau = 0.001$)

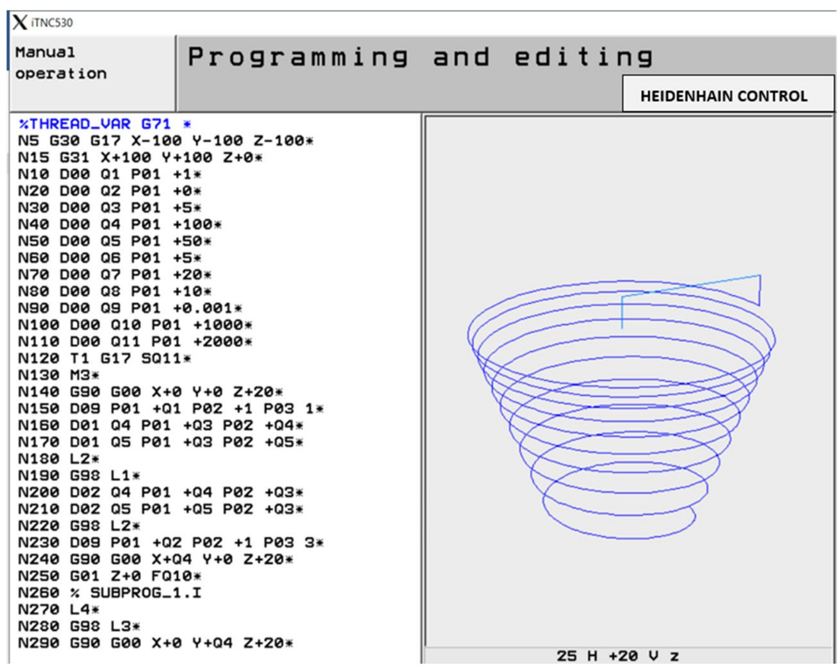


Fig. 23 TEST-8: Internal left-hand thread on variable radius with increasing pitch ($R_i = 100$, $R_f = 50$, pi = 5, pf = 20, n = 10, precision $\tau = 0.001$)

curve. The introduced canned cycle surpasses the limitations of conventional cycles, which are confined to fixed pitch threads, by providing a comprehensive solution for diverse threading requirements.

Simulation tests confirm the adaptability of the proposed canned cycle across diverse threading conditions. The graphical outputs, combined with the mathematical formulation of helix equations and CNC milling machine simulations across various threading cases, collectively verify the efficiency, and validation of the proposed canned cycle.

Funding Open access funding provided by the Cyprus Libraries Consortium (CLC). No funding was received to assist with the preparation of this manuscript.

Declarations

Competing of interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Kumar K, Ranjan C, Davim J (2020) Canned cycle. In: CNC programming for machining. Materials forming, machining and tribology. Springer, Cham. https://doi.org/10.1007/978-3-030-41279-1_9
2. Sinha, SK (2010) Custom Canned Cycles. Chap. 10 in CNC Programming Using Fanuc Custom Macro B. 1st ed. New York: https://www.accessengineeringlibrary.com/content/book/9780071713320/chapter/chapter10. Accessed 5 Jun 2024
3. Martinova L, Fokin N, (2023) Development of a Universal Software Application for Programming Canned Cycles on CNC Turning and Milling Machine Tools, 2023 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, pp. 198–203, https://doi.org/10.1109/RusAutoCon58002.2023.10272910
4. Mohamed A (2015) Enhancement of the Capabilities of CNC Machines via the Addition of a New Counter boring Cycle with a Milling Cutter, Mechanical Engineering Research, Vol. 5, No. 2. https://doi.org/10.5539/mer.v5n2p45
5. Keran C, Dongli M et al (2023) Design and Analysis of passive variable pitch propeller for VTOL UAVs in Aerospace Science and Technology, Elsevier, vol. 132. https://doi.org/10.1016/j.ast.2022.108063
6. Tao R, Qingyou L, Yonghua C et al (2016) Variable pitch helical drive in-pipe robot. Int J Rob Autom 31(3):263–271. https://doi.org/10.2316/Journal.206.2016.3.206-4774
7. Alizade Rasim (2019) Structural Synthesis of Robot Manipulators by Using Screw with Variable Pitch. Universal J Mechanical Eng 7(2):50–63. https://doi.org/10.13189/ujme.2019.070203
8. Alabi Kehinde, Busari Rasheed (2022) Development and Performance Evaluation of a Variable-Pitch Tapered-Shaft Screw Press for Palm Oil Extraction. J Sci, Part A: Eng Innov 9(2):49–61. https://doi.org/10.54287/gujsa.1069996
9. Pravin Salunke, Madhivanan Karthigeyan, et. al., (2023) A Novel Pedicle Screw Design with Variable Thread Geometry: Biomechanical Cadaveric Study with Finite Element Analysis World Neurosurgery, Volume 172, pages e144-e150, ISSN 1878–8750, https://doi.org/10.1016/j.wneu.2022.12.120
10. Gustafson PA, Veenstra JM, Bearden CR, Jastifer JR (2019) The Effect of Pitch Variation and Diameter Variation on Screw Pullout 2(3):258–263. https://doi.org/10.1177/1938640018789999
11. Austin J, Roebke BS, Logan J et al (2018) Fracture Gap Reduction With Variable-Pitch Headless Screws. The J Hand Surgery 43(4):385.e1-385.e8. https://doi.org/10.1016/j.jhsa.2017.10.018
12. Veenstra J, Gustafson P, Bearden C, Jastifer J (2018) The Effect of Pitch Variation and Diameter Variation on Screw Pullout in 3D Printed Screws Foot & Ankle Orthopaedics. 3(3). https://doi.org/10.1177/2473011418S00503
13. Omirou SL, Chasos CA (2024) Design and implementation of an innovative canned cycle for variable pitch thread cutting on CNC milling machines. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-024-12960-x
14. Günther S, von Braunmühl AE, Wieleitner H (1921) Geschichte der mathematik. Göschen GJ (ed) p 92. https://books.google.com.cy/books/about/Geschichte_der_mathematik_t_Von_Cartesiu.html?id=Gy5kzwEACAAJ&redir_esc=y. Accessed 5 Jun 2024
15. https://en.wikipedia.org/wiki/Conical_spiral Retrieved 21 February 2024
16. Wang, BW, Xu, BS, Sun, YH (2011) Machining Process Analysis and Parametric Programming of Parts with Complex Surfaces Advanced Materials Research (Vols. 418–420, pp. 1851–1855). Trans Tech Publications, Ltd. https://doi.org/10.4028/www.scientific.net/amr.418-420.1851
17. Joshi V, Desai K, Raval H (2016) Machining of archimedean spiral by parametric programming. Int J Modern Manufact Technol 8(2):25–30. Elsevier. https://modtech.ro/international-journal/vol8no22016/04_Vratraj_Joshi.pdf. Accessed 5 Jun 2024
18. Rodriguez-Alabanda O, Romero PE, Guerrero-Vaca G (2019) “Application of Custom Macro B high level CNC programming language in a five-axis milling machine for drilling holes distributed in axi-symmetric working planes”, Procedia Manufacturing, Volume 41. ISSN 976–983:2351–9789. https://doi.org/10.1016/j.promfg.2019.10.023
19. Omirou S, Fyrillas M (2015) A general G-code algorithm for deep hole drilling. J Manufact Sci Product 15(2):225–237. https://doi.org/10.1515/jmsp-2014-0028
20. Omirou S (2016) A CNC Parametric Programming Method for Manufacturing of Axisymmetric Mould Cavities. J Manufact Sci Product 16(3):173–181. https://doi.org/10.1515/jmsp-2016-0017
21. Mike Lynch (1997) Parametric programming for computer numerical control machine tools and touch probes. Soc Manufact Eng p 433