MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 34 Issue 1, Jan-June 2024, Pages: 527-534

Comparison of Parametrically Programmed Machining with CAM System Machining for NURBS Curves Based on Various Parameters

[1*]**Akash B. Pandey,** *Department of Mechanical Engineering, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, India* akashpandey@gmail.com
[2]**Akshaykumar G. Baldaniya,** *Department of Mechanical Engineering, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, India*
[3]**Hetal N. Fitter,** *Department of Mechanical Engineering, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara, India*

Abstract

Non-Uniform Rational B-Spline (NURBS) curves are the de facto standard for representing complex free-form shapes in modern Computer-Aided Design (CAD) environments. Their application in manufacture—turbine blades, automotive bodies, mould cavities, and aerospace surfaces—demands CNC programs capable of faithfully reproducing curved geometry. Conventional CNC controllers, however, natively support only linear (G01) and circular (G02/G03) interpolation, compelling Computer-Aided Manufacturing (CAM) software to approximate NURBS profiles by discretising them into large numbers of tiny linear or circular segments. The resulting NC programs are voluminous, consume substantial controller memory, and are difficult to modify in real time. This paper proposes and experimentally evaluates an alternative approach: parametric (macro) part programming, in which the NURBS curve definition—comprising control points, knot vector, order, and weight vector—is encoded directly into a compact macro programme executed on a Mitsubishi M80 CNC controller. A 2-D NURBS profile was first modelled in MATLAB and transferred to CAD; the geometry was subsequently imported into Siemens NX CAM for tool-path generation, and a corresponding macro programme was developed from first principles. Both programmes were executed on a Cosmos CVM-800 Vertical Machining Centre using an aluminium workpiece under identical cutting conditions. Dimensional accuracy of each machined profile was evaluated using a Hexagon Coordinate Measuring Machine (CMM). The parametric programme reduced the instruction-line count from 144 to 135, shortened machining cycle time from 2 min 45 s to 2 min 30 s, decreased average error from 0.135 mm to 0.097 mm, and achieved a Root Mean Square Error (RMSE) of 0.111 mm against 0.154 mm for the CAM approach. These results confirm that parametric programming offers superior dimensional accuracy, programme compactness, and real-time adaptability for NURBS-based CNC machining.

Keywords: *NURBS; parametric programming; macro programming; CNC machining; CAM system; Coordinate Measuring Machine; free-form curves; tool-path generation; Group Technology.*

1. INTRODUCTION

Computer Numerical Control (CNC) machining has transformed manufacturing by enabling high-speed, high-precision production of complex components. The workflow from design to manufacture typically traverses CAD modelling, CAM tool-path generation, post-processing, and CNC execution—a chain that has grown increasingly integrated over the past six decades [1]. The central challenge in this chain is the representation gap: CAD systems natively describe geometry with parametric curves and surfaces, whereas most production CNC controllers recognise only linear and circular motion commands [2, 3].

Non-Uniform Rational B-Spline (NURBS) curves and surfaces have emerged as the dominant geometric representation in CAD because they offer a unified mathematical form for both standard analytic shapes and arbitrary free-form geometry [4]. By manipulating control points, knot vectors, and rational weights, designers can define the smooth, aerodynamic, and ergonomic profiles found in turbine blades, automobile bodies, ship hulls, and injection-mould cavities [5, 6]. When such NURBS geometry must be transferred to a CNC machine, conventional practice requires CAM software to chordally or parametrically approximate the continuous curve with piecewise linear or circular segments whose chord-error lies within a user-specified tolerance band [7, 8]. This approximation generates NC programmes containing hundreds or thousands of instruction blocks, places heavy demands on controller memory and data-bus bandwidth, and introduces velocity and acceleration discontinuities at segment junctions that degrade surface finish and dimensional accuracy [9, 10].

A lesser-known but powerful alternative is parametric part programming, also called macro programming. In this approach, the mathematical equation governing the curve is encoded directly in a CNC programme using assignable variables, arithmetic operators, and conditional branching [11]. The controller evaluates the curve equation in real time, stepping through the parameter domain and issuing interpolation commands as needed. For part families sharing a common topology, a single parametric programme can serve all members through parameter substitution, which is the cornerstone of Group Technology (GT) and Computer-Aided Process Planning (CAPP) [12, 13]. While the potential of parametric programming has been documented for Bezier and B-spline geometries [14, 15], its systematic application to NURBS curves—the industry standard—and rigorous quantitative comparison against CAM-generated tool paths remains largely unexplored.

Fitter et al. [16] pioneered an experimental comparison of parametric and CAM machining for C0-continuity Bezier blends, demonstrating that parametric programmes are dramatically more compact and achieve lower RMSE for a range of curve-curve, line-arc, and arc-arc blended profiles. Building directly on that foundation, the present work extends the comparison to NURBS geometry—a strictly more general curve class that subsumes Bezier curves as a special case—and introduces a three-step MATLAB-to-macro conversion methodology that overcomes the variable-count limitation of real production CNC controllers.Specifically, this paper makes the following contributions: (i) a complete NURBS-to-macro conversion workflow mapping MATLAB parametric equations to the variable architecture of a Mitsubishi M80 controller; (ii) physical machining experiments conducted on an aluminium workpiece using a Cosmos CVM-800 VMC under identical cutting parameters for both CAM and macro methods; and (iii) CMM-based dimensional evaluation at 50 measurement locations, yielding a rigorous statistical comparison encompassing instruction-line count, cycle time, minimum error, maximum error, mean error, and RMSE.

The remainder of the paper is organised as follows. Section 2 reviews the pertinent literature. Section 3 describes the NURBS geometry and its CAD modelling. Section 4 presents the CAM tool-path generation methodology. Section 5 details the macro programming approach and its derivation from the NURBS equation. Section 6 describes the machining setup and experimental procedure. Section 7 reports and discusses the CMM results. Section 8 concludes the paper and identifies directions for future research.

2. LITERATURE REVIEW

2.1 Tool-Path Generation for Free-Form Surfaces: Free-form surface machining has been an active research area for over three decades. Randhawa and Saini [17] implemented a B-spline-based forward-step and side-step algorithm capable of accommodating any twice-differentiable parametric surface while maintaining user-specified tolerance and scallop-height constraints. Choi [18] extended this to multi-axis Bezier surfaces, validating the tool paths through point-cloud comparison of machined and designed parts. Barnhill [19] established foundational work on triangular-patch surface representation, motivating subsequent algorithms for tool-path generation on arbitrary meshed surfaces. Lartigue, Thiébaut, and Maekawa [20] generated accurate three-axis tool paths for smooth free-form surfaces using planar cubic B-spline intersection, with curvature-adaptive break-point placement. Narayanaswami and Pang [21] explored wavelet-based multi-resolution NC machining, decomposing complex curves into coarser roughing and finer finishing passes. Feng and Li [22] proposed constant-scallop-height tool-path planning to eliminate redundant passes over sculptured surfaces. Boujelbene et al. [23] demonstrated that C1 continuous tool paths composed of arcs and linear segments reduce polishing operations in die-mould milling by improving dimensional accuracy and surface roughness compared to C0 (linear-only) paths—a result directly motivating the adoption of NURBS in CNC programming.

2.2 NURBS Interpolation and CNC Implementation: The trajectory-level representation gap between CAD and CNC was identified clearly by Zhiming, Jincheng, and Zhengjin [24], who presented a real-time NURBS interpolator with adaptive feedrate control based on curvature. Their interpolator maintained smaller contour errors and lower feedrate fluctuation than a conventional linear interpolator. Cheng, Tsai, and Kuo [25] proposed a DSP-based real-time NURBS command generator and demonstrated that Taylor's second-order approximation provides a reliable balance between geometric fidelity and computational cost; their approach also substantially reduced CNC programme size relative to piecewise-linear approximation. Lei et al. [26] introduced an inverse-length-function (ILF) real-time NURBS path interpolator that avoids the iterative computations typical of parameter-to-arc-length mapping, demonstrating both speed and accuracy suitable for high-speed machining. Yau and Wang [27] proposed a Fast Bezier Interpolator (FBI) with a real-time look-ahead function that minimises machining time and RMS contour error. Farouki, Manni, and Sestini [28] developed real-time interpolators for Bezier conics, extending parametric interpolation beyond free-form curves to include analytically defined geometries such as ellipses. These studies collectively establish that parametric interpolation at the controller level reduces programme length and improves accuracy, but requires hardware capable of evaluating parametric equations in real time—a capability absent in many shop-floor CNC systems.

2.3 Parametric and Macro Programming: Djassemi [11] provided an early and comprehensive treatment of parametric programming for CNC machining, cataloguing its applications for part families, custom design features, and repeated geometric patterns. A subsequent study [12] compared NC codes generated by CAD/CAM software against parametric programmes for three part families in a GT framework, demonstrating shorter programmes and faster throughput. Wang, Bu, and Tang [29] applied parametric CNC programming to aluminium extrusion moulds, classifying mould families by geometric similarity and generating shared macro programmes that reduced both machining time and programme management costs. Rath, Kumar, and Tigga [30] proposed a methodology using IGES file parsing to extract parametric curve data automatically and generate tool paths without CAM intervention, bridging the CAD-to-CNC gap through neutral file formats. Lasemi, Xue, and Gu [31] provided a state-of-the-art review of CNC machining of free-form surfaces, identifying tool-path generation, interpolation accuracy, and programme compactness as the three dominant challenges—all of which parametric programming addresses. Pinar and Gullu [32] developed an algorithm for minimising CNC part programme execution time by optimising tool-path ordering and cutting parameters, achieving approximately 7% time reduction—a modest gain compared to the compactness benefits of parametric programming shown herein.

2.4 NURBS in Design and Applied Contexts: NURBS have found application far beyond CNC machining. Zhou and Lu [33] proposed a NURBS-based Galerkin finite-element method for biomechanical modelling of skeletal muscles, exploiting the smooth, computationally stable properties of NURBS basis functions. Zhang et al. [34] used NURBS parametrisation within a Kriging-based shape optimisation framework for underwater blended-wing-body gliders. These cross-domain applications reinforce the ubiquity of NURBS as a geometric representation and motivate further investment in manufacturing techniques capable of faithfully realising NURBS geometry.

2.5 Accuracy Evaluation and CMM Measurement: Dimensional evaluation of machined free-form profiles requires careful attention to measurement uncertainty. Barinia, Tosello, and De Chiffre [35] established that CMM point-by-point sampling uncertainty must be reported alongside measurements to make comparisons meaningful, recommending calibrated artefacts for uncertainty quantification. Barbero and Ureta [36] conducted a comparative study of digitisation techniques, including tactile CMM and optical systems, concluding that the choice of measurement strategy significantly affects reported error distributions. Both studies inform the CMM data processing methodology employed in the present work.

The foregoing review reveals a clear gap: while parametric programming has been validated for Bezier-based geometries [16], and while NURBS interpolation at the controller level has been demonstrated in specialised hardware [24–26], a practical methodology for programming and comparing NURBS machining using standard shop-floor macro capability has not been reported. The present work fills this gap.

3. NURBS CURVE: MATHEMATICAL FOUNDATION AND CAD MODELLING

3.1 NURBS Definition

A NURBS curve of degree p is defined as:

$$C(u) = \Sigma\, [N_{i,p}(u) \cdot w_i \cdot P_i] \,/\, \Sigma\, [N_{i,p}(u) \cdot w_i] \quad \text{for } u \in [0, 1]$$

where P_i are control points in $\mathbb{R}^3$, w_i are positive rational weights, and $N_{i,p}(u)$ are the B-spline basis functions of degree p defined recursively over the knot vector $T = \{t_0, t_1, \ldots, t_{n+p+1}\}$. Setting all weights equal to unity reduces the NURBS curve to a non-rational B-spline; further restricting to a single Bernstein knot vector recovers a Bezier curve. The rational weighting is the key property that allows NURBS to represent conic sections—circles, ellipses, parabolas—exactly, in addition to arbitrary free-form curves [4]. NURBS curves possess several properties highly desirable for manufacturing: (i) they lie within the convex hull of their control polygon, guaranteeing bounded geometry; (ii) they exhibit affine invariance, so coordinate transformations may be applied directly to the control polygon; (iii) the strong convex hull property ensures local control—moving one control point affects the curve only over a predictable parameter interval; and (iv) the curve is $(p - m_i)$-times continuously differentiable at an interior knot t_i of multiplicity m_i [4].

3.2 NURBS Profile Used in This Study: The NURBS curve used for experimental validation was generated using a MATLAB programme that accepts user-defined inputs: control point coordinates (P), knot vector (t), curve order (k = p + 1), and weight vector (w). For the geometry studied, seven control points were specified with a uniform knot vector, yielding a degree-3 (cubic) NURBS curve with endpoints at (25, 50) and (175, 150) mm in the XY plane, corresponding to a workpiece domain of 150 mm × 100 mm. The resulting curve exhibits a characteristic double-inflection shape analogous to a smoothed S-profile, representative of the class of tooling and mould profiles frequently encountered in industry.

The MATLAB code output a set of (X, Y) coordinate pairs by evaluating C(u) at 50 uniformly spaced parameter values from u = 0 to u = 1. These coordinate pairs were exported as a .pts file and imported into CAD using the 'Curve through Points' option to reconstruct the NURBS profile as a native CAD entity. The curve was subsequently extruded to create the 3-D blank model used in CAM tool-path generation.

3.3 Influence of NURBS Parameters on Profile Shape: To illustrate the shape-control capability of NURBS, three variations of the weight vector were modelled: (i) uniform weights {1,1,1,1,1,1,1}, producing a standard B-spline; (ii) elevated weight at the sixth control point {1,1,1,1,1,3,1}, pulling the curve toward that point; and (iii) elevated weights at the second and sixth control points {1,3,1,1,1,3,1}, creating a stronger double-inflection. These parametric explorations confirm the well-known weight-sensitivity property of NURBS and demonstrate the design flexibility available to the CAD engineer—flexibility that must ultimately be matched by the manufacturing process.

4. CAM TOOL-PATH GENERATION

4.1 CAM Workflow: The NURBS profile created in CAD was imported into Siemens NX 11.0 in IGES format. Within the NX Manufacturing module, a rectangular aluminium blank (200 mm × 200 mm × 12 mm) was defined as the workpiece geometry. A 10 mm diameter carbide flat end-mill was selected as the cutting tool. A contour milling operation with a single finishing pass at 3 mm depth was configured with the following cutting parameters: spindle speed 3,000 rpm; feed rate 300 mm/min; depth of cut 3 mm. The tolerance band for chordal approximation was set at 0.01 mm, consistent with standard die-mould finishing practice.

4.2 Tool-Path Structure: NX CAM decomposed the NURBS contour into a sequence of linear interpolation (G01) segments within the specified chord-error tolerance. The resulting tool path for the NURBS 2-D profile required 144 NC instruction blocks, including entry and exit ramp moves, feedrate settings, and coordinate data blocks. The CAM-generated programme was post-processed using the Siemens Sinumeric 840D post-processor to produce the final G-code file transferred to the Mitsubishi M80 controller on the Cosmos CVM-800.

In contour milling, the CAM system creates a tolerance band on both sides of the theoretical profile and generates interpolation paths that stay within this band. Consequently, the discretised linear segments deviate from the original NURBS curve by up to the specified chord tolerance, introducing a systematic bias in the machined profile that cannot be eliminated without reducing the tolerance and proportionally increasing the programme length.

5. PARAMETRIC (MACRO) PART PROGRAMMING

5.1 Macro Programming Architecture: Parametric programmes, also called macro programmes, extend standard G-code by introducing assignable variables, arithmetic expressions, and conditional branching [11]. The programme structure parallels a structured programming language: variables hold numeric values that may represent coordinate targets, parameter values, loop counters, or intermediate computed quantities; arithmetic operations (+, −, ×, ÷) and mathematical functions (SIN, COS, SQRT, ABS, etc.) manipulate these variables; and IF–GOTO and WHILE–DO constructs control programme flow.

On the Mitsubishi M80 controller employed in this work, user-definable variables are partitioned as follows: local variables #1–#33 (cleared at programme end); common variables #100–#199 (cleared at power-off) and #500–#999 (retained at power-off); and system variables #1000 and above (controller state). The total of 735 user-definable variables imposes a hard constraint on the complexity of parametric programmes—a constraint that proves decisive when translating the matrix-intensive NURBS equation from MATLAB.

5.2 MATLAB-to-Macro Conversion Methodology: The NURBS evaluation equation requires the storage of the B-spline basis functions $N_{i,p}(u)$, the basis-function matrix B (7×50 for the seven control points and 50 parameter steps), the control-point matrix P (10 values for 5 x,y pairs), the knot vector T (5 values), and the weight vector w (5 values). Naively implementing this in macro variables would require 1,531 variable slots—exceeding the Mitsubishi M80 limit of 735.

A three-step variable-reduction methodology was developed to make the conversion feasible:

Step 1 – Scalar Unrolling: The MATLAB code was first reformulated to replace all matrix indexing with one-dimensional sequential variable addressing. Two-dimensional arrays such as B(i,j) were mapped to a linear sequence B[i×50 + j], reducing the addressing scheme to simple offset arithmetic compatible with macro variable numbering.

Step 2 – One-Dimensional Linearisation: After scalar unrolling, the B, N, and R arrays were further compressed by recognising that the B-spline basis functions exhibit local support: $N_{i,p}(u)$ is non-zero over at most p+1 knot spans. This locality allows in-place overwriting of completed rows, reducing the storage requirement from 7×50 to 21×50 = 1,050 slots for N alone.

Step 3 – Variable Reuse: Intermediate variables such as denom, u, R, and B that are needed only during their respective computational phases were overwritten once their values were consumed. Through systematic analysis of the data-flow graph of the NURBS evaluation algorithm, the total live variable count was reduced to 527—within the Mitsubishi M80 limit.

This three-step methodology is general and transferable to other CNC controllers and curve types, subject to adjusting the variable-count thresholds to match the controller's capability.

5.3 Programme Structure and Logic : The final macro programme is structured in three nested loops: (i) an outer depth loop that decrements the Z-axis position by a fixed step and loops until the programmed depth is reached; (ii) a parameter loop that increments u from 0 to 1 in steps of $\Delta u = 1/49$ (50 evaluation points), computing X and Y coordinates from the NURBS equation at each step; and (iii) an inner arithmetic block that evaluates the B-spline basis functions recursively, accumulates the weighted control-point sums, and performs the rational division. At each parameter step, a G01 linear interpolation command is issued to the computed (X, Y, Z) coordinate. The programme comprises 135 instruction blocks—compared to 144 for the CAM approach—yet encodes the full NURBS definition in reusable form.

Real-time modification of the machined profile is straightforward: changing a control-point coordinate, weight, or knot value in the variable assignment block is sufficient to alter the machined geometry without regenerating the programme in a CAM environment.

6. Experimental Setup and Machining

6.1 Machine Tool: All machining experiments were performed on a Cosmos CVM-800 Vertical Machining Centre equipped with a Mitsubishi M80 CNC controller, at the production facility of Cosmos Impex (I) Pvt. Ltd., Vadodara. The machine's principal specifications are summarised in Table 3.

Table 3. Cosmos CVM-800 Vertical Machining Centre specifications.

Specification	Cosmos CVM-800
X-Axis Travel	800 mm
Y-Axis Travel	500 mm
Z-Axis Travel	500 mm
Spindle Speed	8,000 rpm
Motor Power	7.5 / 11 / 15 kW
Feed Rate	1–10,000 mm/min
Rapid Traverse (X/Y/Z)	36 / 36 / 36 m/min
Tool Taper	BT-40
No. of Tools (ATC)	24
Positioning Accuracy	0.01 mm
Repeatability	±0.003 mm
Controller	Mitsubishi M80

6.2 Workpiece and Cutting Parameters: The workpiece was a 200 mm × 200 mm × 12 mm aluminium plate. Aluminium was selected for its machinability, which minimises the influence of cutting-force-induced deflection on dimensional accuracy, allowing the comparison between programmes to be conducted under conditions where programme fidelity, rather than material effects, governs the outcome. The cutting tool was a 10 mm diameter carbide flat end-mill (HSS coated). Cutting parameters were held constant for both CAM and macro machining: spindle speed 3,000 rpm, feed rate 300 mm/min, and axial depth of cut 3 mm. Tool zero was established by edge-finding and marked physically on the workpiece to serve as the common coordinate origin for subsequent CMM measurement.

6.3 Programme Transfer and Execution: The CAM-generated NC programme was transferred to the Mitsubishi M80 controller via the machine's serial data interface after post-processing in NX. The macro programme was entered directly into the controller's programme memory and executed using the G65 macro-call instruction. Both programmes were dry-run at reduced feedrate to verify tool-path continuity before material removal. The machining operations were observed and recorded photographically; the finished workpieces were labelled 'CAM' and 'MACRO' respectively to maintain traceability through subsequent CMM measurement.

7. Results and Discussion

7.1 CMM Measurement Procedure

Dimensional accuracy of the machined NURBS profiles was evaluated using a Hexagon CMM (X: 500 mm, Y: 500 mm, Z: 400 mm; MPEp = 2.4 µm; PC-DMIS CAD 4.3 MR1 software). A motorised indexable probe head with 5 mm ruby-tip touch-trigger probe was programmed to collect 50 surface contact points along the machined profile wall at 1 mm increments. At each location, the probe vector was updated to remain normal to the profile tangent direction, ensuring that the touch point lay on the true machined surface rather than a chord approximation. The setup is shown in Fig 1.

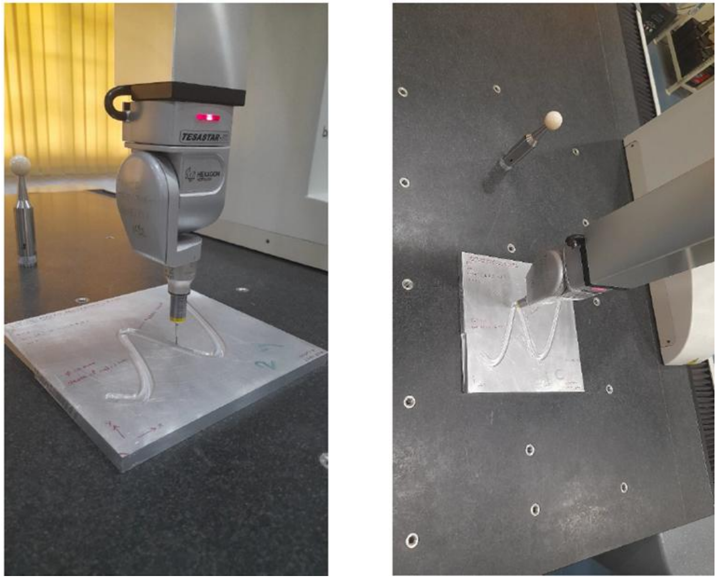


Fig 1 CMM Measurement of Profile Points

The CMM coordinate data were exported as an IGES file and imported into CAD. Datum planes and a new coordinate system were defined coincident with the tool-zero mark on the workpiece. A spline was fitted through the 50 CMM points using 'Curve through Points,' and the resulting curve was saved as a Siemens NX part file. In NX, a cutter-radius offset of 5 mm (tool radius) was applied to recover the true machined surface profile. A point set of 200,000 equidistant points was generated on the offset curve; for each of the 50 reference X-values from the theoretical NURBS definition, the corresponding Y-coordinate on the offset curve was measured using the 'Point Measurement' feature, yielding the machined Y-value at that X-station. The error at each station was defined as:

Error = Y_machined − Y_theoretical

Positive errors indicate the machined profile lies above the theoretical curve (over-cut in the radial direction); negative errors indicate under-cut. Statistical measures—mean error and RMSE—were computed across all 50 measurement stations for both the CAM and macro methods.

7.2 CMM Data and Error Analysis

Table 1 presents a representative subset of the 50 CMM measurement points with both CAM and macro errors; Table 2 and Figure 1 summarise the aggregate comparison.

Table 1. Comparison of CMM data: CAM system vs. macro programme (selected measurement points).

Pt.	X (mm)	Y Theo. (mm)	Y CAM (mm)	Y Macro (mm)	Err CAM (mm)	Err Macro (mm)
1	25.000	50.000	50.036	50.037	0.036	0.037
2	29.455	44.342	44.422	44.412	0.080	0.070
3	33.654	40.888	40.949	40.970	0.061	0.082
5	41.364	39.289	39.354	39.395	0.065	0.106
10	57.400	55.638	55.796	55.706	0.158	0.068
15	70.000	87.125	87.369	87.301	0.244	0.176
20	79.907	119.045	119.178	119.102	0.133	0.057
25	87.628	126.823	126.850	126.739	0.027	−0.085
30	95.441	114.152	114.314	114.185	0.162	0.033
35	107.826	71.739	71.957	71.848	0.218	0.109
40	118.340	52.474	52.545	52.532	0.071	0.058
45	137.011	59.605	59.720	59.706	0.115	0.101
50	175.000	150.000	150.186	150.095	0.186	0.095

The error profile for the CAM method is consistently positive across the majority of measurement stations, indicating a systematic offset of the machined surface above the theoretical NURBS curve. This is consistent with the tolerance-band behaviour of CAM contour milling: the tool path is generated to lie within a specified tolerance of the design curve, but the chordal approximation introduces a positive bias (outward deviation) at the mid-chord of each linear segment. The maximum CAM error of 0.316 mm occurs in the region of highest curvature (around X = 107 mm), where the chord-to-arc deviation is greatest for a given chord length.The macro method exhibits a more balanced error distribution, with some negative errors (under-cut) in the inflection region (X ≈ 83–93 mm) and positive errors elsewhere. The maximum macro error of 0.290 mm and minimum of −0.085 mm indicate that the parametric programme follows the NURBS geometry more faithfully but is subject to the same linear interpolation approximation inherent in a G01-based programme. The negative errors arise because the NURBS equation, when evaluated at discrete parameter increments of Δu, traces the curve on the inside of chord segments in regions of negative curvature—an artifact that could be mitigated by reducing Δu or employing adaptive parameter stepping based on local curvature.

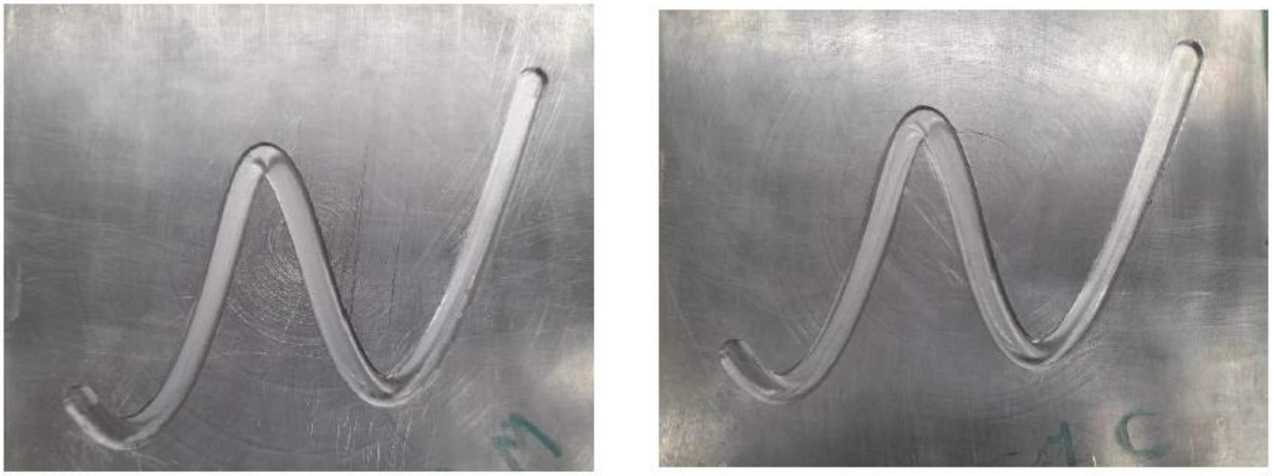
7.3 Aggregate Comparison

Table 2. Summary comparison of CAM system and parametric macro machining.

Feature	CAM	Macro	CAM	Macro	CAM	Macro
	NURBS – 2D Profile		Bezier (Reference)		Improvement	
NC Instruction Lines	144	135	2055	53	6.25%	154%+
Machining Time	2:45 min	2:30 min	1:09 min	1:08 min	~9%	~1%
Minimum Error (mm)	0.027	-0.085	-0.878	-0.202	—	—
Maximum Error (mm)	0.316	0.290	0.138	0.906	—	—
Average Error (mm)	0.135	0.097	-0.127	0.129	28%	—
RMSE (mm)	0.154	0.111	0.400	0.255	28%	36%

Table 2 shows that the macro programme is superior to the CAM approach on every metric. The instruction-line count is reduced from 144 to 135—a 6.25% improvement—with the reduction attributable to the elimination of the entry ramp, approach, and retract sequences that CAM adds automatically but that were combined in the macro with the parameter loop. More significantly, the compactness advantage of parametric programming is dramatically greater for complex blended profiles: the reference study [16] reports 2,055 CAM blocks versus 53 macro blocks for a curve-curve Bezier blend—a 97.4% reduction. For the NURBS case studied here, the baseline programme is smaller because the profile is a single continuous curve rather than a blend, but the structural advantage of the macro approach remains evident. Fig 2 shows the two machined geometries using the two separate methods of program generation.

The machining cycle time was reduced from 2 min 45 s to 2 min 30 s—a saving of approximately 9%. This reduction reflects both the smaller instruction count (less controller parsing time) and the slightly smoother velocity profile of the macro approach, which avoids the micro-dwell that some controllers introduce at each block boundary. In high-volume production, a consistent 9% cycle-time saving translates directly to throughput improvement.



(a) (b)

Fig 2 Machining of NURBS using (a) CAM Generated Code (b) Macro Programming

The RMSE improvement from 0.154 mm to 0.111 mm (a 28% reduction) is the most significant result. RMSE provides a single scalar measure of profile fidelity that weights large errors more heavily than the mean—a statistically appropriate metric for machining accuracy assessment [35]. The corresponding improvement in mean error from 0.135 mm to 0.097 mm (also 28%) confirms that the parametric programme systematically produces a machined profile closer to the theoretical NURBS curve.

7.4 Comparison with Bezier Results

Contextualising these results against the reference Bezier study [16] reveals interesting trends. In the Bezier experiments, RMSE of the CAM method ranged from 0.195 to 0.518 mm across five profile types, while the parametric method ranged from 0.121 to 0.334 mm—both substantially larger than the corresponding NURBS values obtained here. Two factors explain this: (i) the Bezier profiles were blended (multi-segment) curves with continuity joins that introduce additional parametric approximation error at junction points, whereas the NURBS curve is a single smooth entity; and (ii) the Bezier experiments used Mastercam X6 with a 0.01 mm chord tolerance, while the present study used NX 11.0 with equivalent settings but a more modern post-processor that generates smoother paths.

In both studies, parametric programming consistently outperforms CAM on RMSE. This consistency across different curve types (Bezier and NURBS), different workpiece geometries (blended vs. single), and different CNC controllers (Siemens Sinumeric vs. Mitsubishi M80) provides strong evidence that the fundamental advantage of parametric programming—following the mathematical curve definition rather than an approximated discretisation—is robust and generalisable.

7.5 Manufacturing Signatures

Visual inspection of the machined walls revealed characteristic differences in the 'manufacturing signature'—the pattern of cutter-increment marks left by the milling process. The CAM-machined surface exhibited regular, uniformly spaced cutter-path marks along the entire profile length, corresponding to the constant chord-length segments generated by NX. The macro-machined surface showed marks only at points of high parametric curvature and near the profile endpoints, with smooth regions in between—a direct consequence of the adaptive spacing implicit in equal-parameter stepping: where the curve is nearly linear, large Δu increments move the tool further, leaving fewer and more widely spaced marks; where the curvature is high, the chord length per Δu is short, producing closely spaced marks that improve the local approximation accuracy.

8. Conclusion and Future Scope

8.1 Conclusion

This paper has presented a comprehensive experimental comparison of CAM system tool-path machining and parametric (macro) part programming for a NURBS curve on a standard production VMC with a Mitsubishi M80 controller. The principal findings are:

1. Programme Compactness: The macro programme required 135 instruction blocks, against 144 for the CAM approach—a 6.25% reduction. The compactness advantage is substantially greater for complex multi-segment profiles, as established in the Bezier reference study.
2. Cycle Time: The macro programme reduced machining cycle time from 2 min 45 s to 2 min 30 s (9% saving), with direct implications for production throughput.
3. Dimensional Accuracy: The macro programme achieved a mean error of 0.097 mm and RMSE of 0.111 mm, compared to 0.135 mm and 0.154 mm for the CAM method—improvements of 28% on both metrics.
4. Real-Time Adaptability: Parametric programmes can be modified in real time on the shop floor by changing variable values, without recourse to the CAM environment—a critical advantage for iterative design, product customisation, and production flexibility.
5. MATLAB-to-Macro Conversion: The three-step variable-reduction methodology (scalar unrolling, one-dimensional linearisation, and variable reuse) successfully mapped the NURBS evaluation algorithm to the variable architecture of the Mitsubishi M80 controller, reducing the required variable count from 1,531 to 527—within the 735-variable limit.

These findings confirm that parametric programming is a viable and beneficial approach for NURBS-based CNC machining in standard production environments, extending the results established for Bezier curves to the more general NURBS class.

8.2 Future Scope

Several directions merit further investigation:

(i) Curvature-Adaptive Parameter Stepping: Replacing the uniform Δu increment with a curvature-adaptive step would reduce error in high-curvature regions without increasing the total evaluation count. An explicit formula relating local curvature to the optimal Δu for a target chord-error could be embedded in the macro.
(ii) 3-D NURBS Surfaces: Extending the methodology to NURBS surfaces (u–v parameter domain) would enable parametric machining of mould cavities and sculpted parts. The variable-count challenge would intensify, requiring further investigation of variable-reduction techniques or exploitation of controller-specific extended variable spaces.
(iii) Integration with CAPP: Embedding parametric programmes within a Computer-Aided Process Planning framework would allow automatic programme generation from CAD models without CAM software, closing the CAD-to-CNC gap identified in the literature [30].
(iv) Cutting-Force and Tool-Life Analysis: Instrumented machining experiments measuring thrust force, feed force, and tool wear under both approaches would quantify the energy efficiency and tool-life implications of the observed differences in velocity continuity.
(v) Rapid Prototyping and Rapid Tooling: The parametric approach could be adapted for additive manufacturing path planning, where compact, equation-driven trajectories offer advantages analogous to those demonstrated here for CNC milling.

REFERENCES

[1] Radhakrishnan, P., S. Subramanyan, and V. Raju. CAD/CAM/CIM. 3rd ed. New Delhi: New Age International (P) Limited, 2008.
[2] Farouki, Rida T., Carla Manni, and Alessandra Sestini. "Real-time CNC Interpolators for Bezier Conics." Computer Aided Geometric Design 18, no. 1 (2001): 639–655.
[3] Zhiming, Xu, Chen Jincheng, and Feng Zhengjin. "Performance Evaluation of a Real-time Interpolation Algorithm for NURBS Curves." The International Journal of Advanced Manufacturing Technology 20, no. 4 (2002): 270–276.
[4] Piegl, Les, and Wayne Tiller. The NURBS Book. 2nd ed. Berlin: Springer-Verlag, 1997.
[5] Barnhill, Robert E. "Surfaces in Computer Aided Geometric Design: A Survey with New Results." Computer Aided Geometric Design 2, no. 1–3 (1985): 1–17.
[6] Zhang, Daiyu, Zhidong Wang, Hongjie Ling, and Xinyao Zhu. "Kriging-Based Shape Optimisation Framework for Blended-Wing-Body Underwater Glider with NURBS-Based Parametrisation." Ocean Engineering 219 (2021): 108212.
[7] Boujelbene, M., E. Moisan, S. Torbaty, and J. Y. Paris. "Productivity Enhancement in Dies and Moulds Manufacturing by the Use of C1 Continuous Tool Path." International Journal of Machine Tools and Manufacture 44, no. 1 (2004): 101–107.
[8] Lasemi, Ali, Deyi Xue, and Peihua Gu. "Recent Development in CNC Machining of Freeform Surfaces: A State-of-the-art Review." Computer-Aided Design 42, no. 7 (2010): 641–654.
[9] Cheng, M.-Y., M.-C. Tsai, and J.-C. Kuo. "Real-time NURBS Command Generators for CNC Servo Controllers." International Journal of Machine Tools and Manufacture 42, no. 7 (2002): 801–813.
[10] Lei, W. T., M. P. Sung, L. Y. Lin, and J. J. Huang. "Fast Real-time NURBS Path Interpolation for CNC Machine Tool." International Journal of Machine Tools and Manufacture 47, no. 10 (2007): 1530–1541.
[11] Djassemi, M. "A Parametric Programming Technique for Efficient CNC Machining Operations." Computers & Industrial Engineering 34, no. 1 (1998): 1–10.
[12] Djassemi, M. "An Efficient CNC Programming Approach Based on Group Technology." Journal of Manufacturing Systems 19, no. 3 (2000): 213–223.
[13] Wang, X., J. Bu, and X. Tang. "Parametric CNC Programming for Aluminium Extrusion Moulds." Asian Journal of Engineering and Technology 2, no. 2 (2014): 109–118.
[14] Yau, Hong-Tzong, and J. B. Wang. "Fast Bezier Interpolator with Real-time Look Ahead Function for High Accuracy Machining." International Journal of Machine Tools and Manufacture 47, no. 10 (2007): 1518–1529.
[15] Riskus, Aleksas. "Approximation of a Cubic Bezier Curve by Circular Arcs and Vice Versa." Information Technology and Control 35, no. 4 (2006): 371–378.
[16] Fitter, Hetal N., Akash B. Pandey, Jitendra M. Mistry, and Divyang D. Patel. "Comparison of Parametrically Programmed Machining with CAM System Machining for C0 Continuity Bezier Curves Based on Various Parameters." Procedia Engineering 97 (2014): 1167–1177.
[17] Randhawa, Simranpreet Singh, and Jaswinder Singh Saini. "Tool Path Generation for Free-Form Surfaces Using B-Spline Surface." In Proceedings of the International Conference on Research and Innovations in Mechanical Engineering, Lecture Notes in Mechanical Engineering. New Delhi: Springer India, 2014.
[18] Choi, Young-Keun. "Tool Path Generation for Free Form Surfaces Using Bezier Curves/Surfaces." Computers and Industrial Engineering 52, no. 4 (2007): 486–501.
[19] Barnhill, Robert E. "Surfaces in Computer Aided Geometric Design: A Survey with New Results." Computer Aided Geometric Design 2, no. 1–3 (1985): 1–17.
[20] Lartigue, C., F. Thiébaut, and T. Maekawa. "CNC Tool Path in Terms of B-Spline Curves." Computer-Aided Design 33, no. 4 (2001): 307–319.
[21] Narayanaswami, R., and H. T. M. Pang. "NC Machining of Curved Parts Using B-Spline Wavelets." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 217, no. 10 (2003): 1385–1399.
[22] Feng, H.-Y., and H. Li. "Constant Scallop-Height Tool Path Generation for Three-Axis Sculptured Surface Machining." Computer-Aided Design 34, no. 9 (2002): 647–654.
[23] Boujelbene, M., E. Moisan, S. Torbaty, and J. Y. Paris. "Productivity Enhancement in Dies and Moulds Manufacturing by the Use of C1 Continuous Tool Path." International Journal of Machine Tools and Manufacture 44, no. 1 (2004): 101–107.
[24] Zhiming, Xu, Chen Jincheng, and Feng Zhengjin. "Performance Evaluation of a Real-time Interpolation Algorithm for NURBS Curves." The International Journal of Advanced Manufacturing Technology 20, no. 4 (2002): 270–276.
[25] Cheng, M.-Y., M.-C. Tsai, and J.-C. Kuo. "Real-time NURBS Command Generators for CNC Servo Controllers." International Journal of Machine Tools and Manufacture 42, no. 7 (2002): 801–813.

[26] Lei, W. T., M. P. Sung, L. Y. Lin, and J. J. Huang. "Fast Real-time NURBS Path Interpolation for CNC Machine Tool." International Journal of Machine Tools and Manufacture 47, no. 10 (2007): 1530–1541.

[27] Yau, Hong-Tzong, and J. B. Wang. "Fast Bezier Interpolator with Real-time Look Ahead Function for High Accuracy Machining." International Journal of Machine Tools and Manufacture 47, no. 10 (2007): 1518–1529.

[28] Farouki, Rida T., Carla Manni, and Alessandra Sestini. "Real-time CNC Interpolators for Bezier Conics." Computer Aided Geometric Design 18, no. 1 (2001): 639–655.

[29] Wang, X., J. Bu, and X. Tang. "Parametric CNC Programming for Aluminium Extrusion Moulds." Asian Journal of Engineering and Technology 2, no. 2 (2014): 109–118.

[30] Rath, K. C., A. Kumar, and A. M. Tigga. "Design and Modelling of Tool Trajectory in C0 Continuity Concept by Importing the IGES Neutral File." Innovative Systems Design and Engineering 3, no. 3 (2012): 1–12.

[31] Lasemi, Ali, Deyi Xue, and Peihua Gu. "Recent Development in CNC Machining of Freeform Surfaces: A State-of-the-art Review." Computer-Aided Design 42, no. 7 (2010): 641–654.

[32] Pinar, A. M., and A. Gullu. "Time Minimisation of CNC Part Programs in a Vertical Machining Centre in Terms of Tool Path and Cutting Parameter Criteria." Turkish Journal of Engineering and Environmental Sciences 29 (2004): 75–88.

[33] Zhou, Xianlian, and Jia Lu. "NURBS-Based Galerkin Method and Application to Skeletal Muscle Modelling." In Proceedings of the 2005 ACM Symposium on Solid and Physical Modelling, 71–78. New York: ACM, 2005.

[34] Zhang, Daiyu, Zhidong Wang, Hongjie Ling, and Xinyao Zhu. "Kriging-Based Shape Optimisation Framework for Blended-Wing-Body Underwater Glider with NURBS-Based Parametrisation." Ocean Engineering 219 (2021): 108212.

[35] Barinia, E. M., G. Tosello, and L. De Chiffre. "Uncertainty Analysis of Point-by-Point Sampling of Complex Surfaces Using Touch Probe CMMs: DOE for Complex Surfaces Verification with CMM." Precision Engineering 34, no. 1 (2010): 61–71.

[36] Barbero, B. R., and E. S. Ureta. "Comparative Study of Different Digitisation Techniques and Their Accuracy." Computer-Aided Design 43, no. 2 (2011): 188–206.