Integration of CAM Systems into Multi-Axes Computerized Numerical Control Machines

Chu Anh My
Department of Special Robot and Mechatronics
Faculty of Aerospace Engineering, Le Qui Don Technical University
Hanoi, Vietnam
e-mail: mychuanh@yahoo.com

Abstract—**A solution for integrating CAM (Computer - Aided Manufacturing) systems into multi-axis nutating table CNC (Computerized Numerical Control) machines is presented in this paper. The strategy of the method is to build a CL data processing algorithm. Thus, the CL data in ISO format produced by every CAMs can be transformed and translated into G-codes files (Numerical Control files) for controlling CNC machines. An implementation of the integration and real tests performing on industrial 5-axis DMU 50e CNC machine at Dong Anh Mechanical Comp. are carried out to verify the research results.**

Keywords-CAM/CNC integration; multi-axis CNC machine; postprocessor

I. INTRODUCTION

In design and manufacturing industry, numerical control (NC) concept refers to the automation of machine tools that are operated by abstractly programmed commands encoded on a storage medium, as opposed to manually controlled. The first NC machines were built in the 1940s and '50s, based on existing tools that were modified with motors that moved the controls to follow points fed into the system on paper tape. These early servomechanisms were rapidly augmented with analog and digital computers, creating the modern computed numerically controlled (CNC) machine tools that have revolutionized the design and manufacturing processes. Nowadays, CNC machines includes several types: 2-axis CNC machine, 3-axis CNC machine, 4-axis CNC machine, 5-axis machine, etc. The number of the axis of a CNC machine implies the number of degree of freedom that the controller of the machine can be simultaneously interpolated. If the axis number increases, the machining efficiency, effectiveness and accuracy will increase; however, it requires more complex techniques in control programming process.

Five-axis milling CNC machine (so-called 5-axis milling Robot) has been proven to be the most efficient tool for fabricating products of complex geometry which may include numerous freeform surfaces. The products are widely used in several high technology industries such as the aerospace industry, the automotive industry, the shipbuilding industry, etc. Using 5-axis CNC systems integrated with CAMs, end-to-end component design and manufacturing is highly automated.

The main function of any CAM program is to computes the cutter trajectory (tool path), basing the input on part surface modeling, the surface quality requirement (surface error), the cutter definition, the tool path pattern, etc. According to a specific CNC controller, the data of the path computed is then used to compile NC program for controlling the CNC machine.

For every CAM system, the tool path is defined as the piecewise curve passing through CL points (cutter location point) as shown in Fig. 1. The CL point is a given point defined on the cutter. At a given tool position, the CL point is computed according to the CC (cutter contact) point, the unit normal vector of the surface, the tool size, and the tool axis inclination [1].

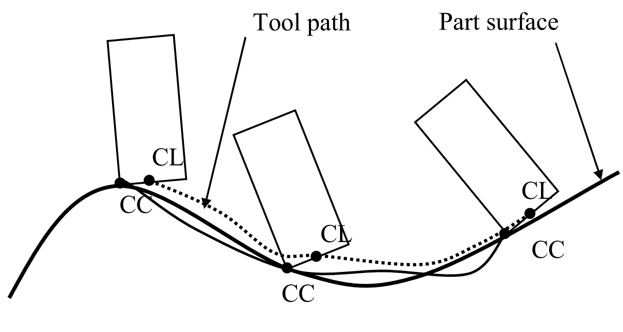

Figure 1. CL Tool path

The CL points data represents in ISO format as $\{x_i, y_i, z_i, I_i, K_i, J_i\}$ where (x_i, y_i, z_i) is coordinates of CL_i and (I_i, K_i, J_i) is direction cosines of the tool axis orientation correspondingly.

In traditional CNC machining techniques (2-axis, 3-axis), the orientation of the tool axis does not vary in workpiece coordinate system during machining. The CL data can be transferred into G-codes file very easily since only (x_i, y_i, z_i) is considered and linear

978-0-7695-4213-3/10 $25.00 © 2010 IEEE
DOI 10.1109/KSE.2010.30

transformation is needed. This transformation regards with the setting of the workpiece zero point and the machining zero point. If the two zero points coincide, the transformation will be negleged.

In the case of 5-axis machining, the tool axis orientation can be changed, so the direction cosines *I*, *J*, *K* of the tool axis vector for every tool posture (in each record of the CL data) must be considered. It means that each record of the data must consists of 6 components (fields): *x*, *y*, *z*, *I*, *J*, and *K*.

As introduced, the 5-axis CNC controller only interpolates 5 axis displacements simultaneously. Depending on the specific machine configuration, the axis displacements can be (X, Y, X, A, B), (X, Y, Z, A, C) or (X, Y, Z, B, C) where X,Y,Z are translation axis and (A,B), (B,C), (A,C) are rotation axis. In the other words, the maximum number of components in G-code statements controlling tool motion is only 5, not 6.

Hence, to integrate CAMs into 5-axis CNC machine, we need a CL data postprocessing program as in Fig. 2.

The main functions of the proposed program should be (i) understanding any CL data produced by CAMs, and (ii) transforming the data set $\{x_i, y_i, z_i, I_i, J_i, K_i\}$ into the set of NC commands (X, Y, X, A, B), (X, Y, Z, A, C) or (X, Y, Z, B, C).

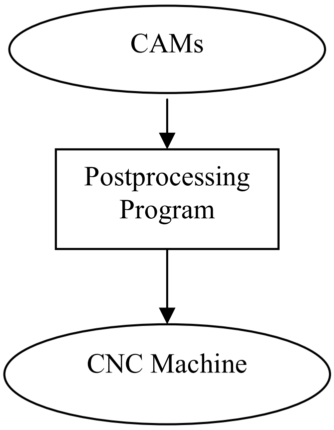

Figure 2. Data postprocessing for CNC machine

It notices that the CL data is standardized in ASCII format, so the first function of the postprocessing program is quite simple to implement. Only reading text and filtering fields of the data for all records is needed. However, the second function of the program requires a complex transformation (a mapping) $f : \Re^6 \rightarrow \Re^5$.

The transformation f depends mainly on configuration structure and system kinematics parameters of a specific 5-axis CNC machine [2,3,4]. The research presented in [5] investigates a controller capable of accepting the CL data to machine the workpiece in realtime without the need of a postprocessor; but it is relatively expensive and not commonly used in most industries. The methods develop multi-axis postprocessing presented in [6,7] just for orthogonal rotary axes, not for nonorthogonal axes. Relatively few studies have addressed nonorthogonal configuration [8,10]. Three types of configuration are considered to analyze three corresponding kinematics models [8]. The development in [10] emphasizes on the homogeneous coordinates transformation of the configuration with variable inclination of B axis. The main conclusions of [10] show a methodology and the derived NC data equations that are useful for such the specific machine tools. It has long been a common way that, for complex surface machining with high quality and efficiency, each specific 5-axis CNC machine needs a particular CAM-CNC integration for itself. At present, a generalized solution for the integration that can be used for all types of 5-axis CNC machine is still far from a reality. This is the reason why all of today's commercial CAMs just give ISO CL data as the conventional output, not NC file as expected.

This paper develops a postprocessing program based on constructing the invert kinematics modeling for a nutating table configuration - a specific nonorthogonal configuration. The program built is also integrated with an industrial CAM and CNC system; real cutting patterns is made on DMU 5-axis CNC machine to check and validate the program.

II. INVERT KINEMATICS TRANSFORMATION

Consider the configuration of the specific DMU 50e in Fig. 3. Five axis of the machine are denoted as X, Y, Z, B, C. Notice that the center line of B axis is inclined under an angle of 45^0 as compared with Z axis.

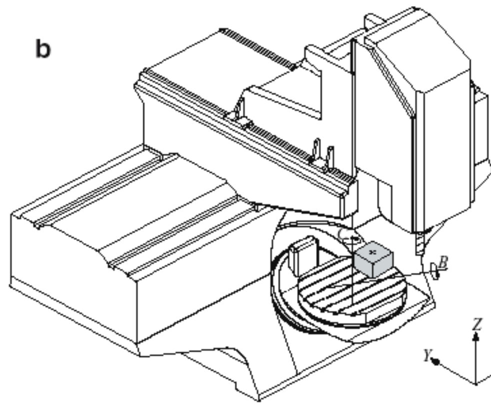

Figure 3. Configuration of DMU 50e [9]

As depicted in Fig. 4, we define two Castersian coordinate systems as follows.

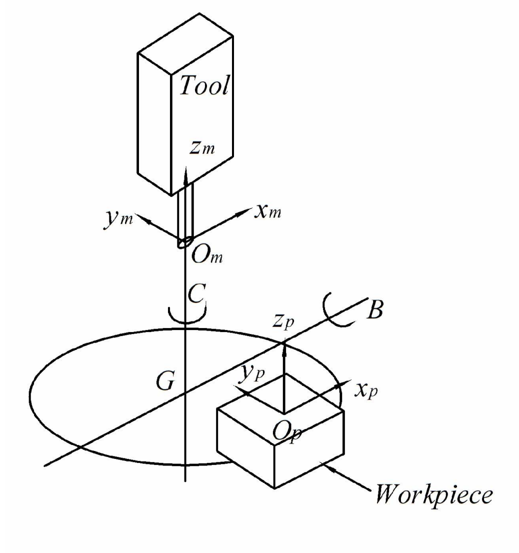

Figure 4. Definitions of coordinate systems

$O_p = (O_p, x_p, y_p, z_p)$ is called the workpiece coordinate system fixed on the workpiece. Usually, O_p is selected in CAM processing steps.

$O_m = (O_m, x_m, y_m, z_m)$ is called the machine coordinate system fixed on the machine. The origin O_p coincides with the tooltip, the axis z_m aligns with the tool axis, and x_m and y_m coincide with X and Y displacements of the machine motion, respectively.

The point G is the intersection of the centerlines of B and C axis. The transformation f can be yielded by translating the coordinate system O_p to the point G, then rotating it around the centerlines of C and B axis, and finally translating to O_m.

In essence, each rotation or translation can be characterized by a homogeneous transformation matrix. For the translation, the matrix can be denoted by **T**, and the rotation can be denoted by **R**.

Given a Castersian coordinate system

$O_s = (O_s, x_s, y_s, z_s)$ and a point $\mathbf{a} = \begin{bmatrix} x_a & y_a & z_a \end{bmatrix}^T$, the translation of O_s from current position to **a** is characterized by

$$\mathbf{T}(x_a, y_a, z_a) = \begin{bmatrix} 1 & 0 & 0 & x_a \\ 0 & 1 & 0 & y_a \\ 0 & 0 & 1 & z_a \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Consider the rotation of O_s under a given angle, and the new posture of O_s is denoted by $O_s' = (O_s', x_s', y_s', z_s')$, the rotation can be characterized by

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where r_{i1} are three direction cosines of axis x_s' in O_s, r_{i2} are three direction cosines of axis y_s' in O_s, and r_{i3} the direction cosines of axis z_s' in O_s.

In a specific case that O_s rotates around a vector $\mathbf{\Delta} = \begin{bmatrix} \Delta_x & \Delta_y & \Delta_z \end{bmatrix}^T$ an angle θ, the rotation matrix can be expressed as

$$\mathbf{R}(\mathbf{\Delta}, \theta) = \begin{bmatrix} \Delta_x^2(1-\cos\theta)+\cos\theta & \Delta_x\Delta_y(1-\cos\theta)-\Delta_z\sin\theta & \Delta_x\Delta_z(1-\cos\theta)-\Delta_y\sin\theta & 0 \\ \Delta_x\Delta_y(1-\cos\theta)+\Delta_z\sin\theta & \Delta_y^2(1-\cos\theta)+\cos\theta & \Delta_y\Delta_z(1-\cos\theta)-\Delta_x\sin\theta & 0 \\ \Delta_x\Delta_z(1-\cos\theta)-\Delta_y\sin\theta & \Delta_y\Delta_z(1-\cos\theta)+\Delta_x\sin\theta & \Delta_z^2(1-\cos\theta)+\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Take a look at Fig. 4. The translation of O_p to O_m through the point G and axis C and B, the following equation can be yielded

$$\mathbf{M} = \mathbf{T}_1.\mathbf{R}(\theta_C).\mathbf{R}(\theta_B).\mathbf{T}_2 = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $\mathbf{T}_1$ represents the translation of O_p to G,

$\mathbf{R}(\theta_C)$ represents the rotation around the centerline of C axis,

$\mathbf{R}(\theta_B)$ represents the rotation around the centerline of B axis, and

$\mathbf{T}_2$ represents the translation with the relative distances x_m, y_m, z_m.

It is clear that the components (m_{13}, m_{23}, m_{33}) are direction cosines of z_m (tool axis vector) represented in O_p; the components (m_{14}, m_{24}, m_{34}) in (1) are coordinates of a point $\begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T$ represented in O_p as well. Hence, we only need to consider the last two columns of the matrix **M**.

As mentioned, each record of the CL data represents a CL point on the toolpath. The point (record) consists of (x_p, y_p, z_p, I, J, K), and it can be expressed as

$$\mathbf{P} = \begin{bmatrix} I & x_p \\ J & x_p \\ K & x_p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Taking (1) equal to (2) obtains (3) and (4) as follows.

$$\begin{bmatrix} I \\ J \\ K \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_C\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)-\Delta_y\sin\theta_B\right]+\sin\theta_C\left[\Delta_y\Delta_z\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right] \\ -\sin\theta_C\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)-\Delta_y\sin\theta_B\right]+\cos\theta_C\left[\Delta_y\Delta_z\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right] \\ \Delta_z^2\left(1-\cos\theta_B\right)+\cos\theta_B \\ 0 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 0 \end{bmatrix} = \begin{bmatrix} x_m\left\{\cos\theta_C\left[\Delta_z^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+\sin\theta_B\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)-\Delta_z\sin\theta_B\right]\right\}+ \\ +y_m\left\{\cos\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)+\Delta_z\sin\theta_B\right]+\sin\theta_B\left[\Delta_y^2\left(1-\cos\theta_B\right)-\sin\theta_B\right]\right\}+ \\ +z_m\left\{\cos\theta_C\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)+\Delta_y\sin\theta_B\right]+\sin\theta_C\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]\right\}+G_x \\ x_m\left\{-\sin\theta_C\left[\Delta_x^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+\cos\theta_B\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)-\Delta_z\sin\theta_B\right]\right\}+ \\ +y_m\left\{-\sin\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)+\Delta_z\sin\theta_B\right]+\cos\theta_B\left[\Delta_y^2\left(1-\cos\theta_B\right)-\sin\theta_B\right]\right\}+ \\ +z_m\left\{-\sin\theta_C\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)+\Delta_y\sin\theta_B\right]+\cos\theta_C\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]\right\}+G_y \\ x_m\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)+\Delta_y\sin\theta_B\right]+y_m\left[\Delta_z\Delta_y\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]+ \\ +z_m\left[\Delta_z^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+G_z \\ 0 \end{bmatrix} \quad (4)$$

Notice that G_x, G_y, G_z are the coordinates of point G in O_p coordinate system. Equations (3) and (4) are the forward kinematics modeling of the systems which play and important role in behavior analysis of the kinematics chain. Also from the derived equations we can yield

$$B=\theta_B=\arccos\left(\frac{K-\Delta_z^2}{1-\Delta_z^2}\right) \quad (5)$$

$$C=\theta_C=\arctan\left(\frac{-J\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)+\Delta_y\sin\theta_B\right]+I\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]}{I\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)+\Delta_y\sin\theta_B\right]+J\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]}\right) \quad (6)$$

$$\begin{aligned} x_m=&\left(x_p-G_x\right)\left\{\cos\theta_C\left[\Delta_x^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+\sin\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)-\Delta_z\sin\theta_B\right]\right\}+ \\ &+\left(y_p-G_y\right)\left\{-\sin\theta_C\left[\Delta_x^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+\cos\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)-\Delta_z\sin\theta_B\right]\right\}+ \\ &+\left(z_p-G_z\right)\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)-\Delta_y\sin\theta_B\right]+G_x \end{aligned} \quad (7)$$

$$\begin{aligned} Y_m=&\left(x_p-G_x\right)\left\{\cos\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)+\Delta_z\cos\theta_B\right]+\sin\theta_C\left[\Delta_y^2\left(1-\cos\theta_B\right)-\sin\theta_B\right]\right\}+ \\ &+\left(y_p-G_y\right)\left\{-\sin\theta_C\left[\Delta_y\Delta_x\left(1-\cos\theta_B\right)+\Delta_z\sin\theta_B\right]+\cos\theta_C\left[\Delta_y^2\left(1-\cos\theta_B\right)-\sin\theta_B\right]\right\}+ \\ &+\left(z_p-G_z\right)\left[\Delta_z\Delta_x\left(1-\cos\theta_B\right)-\Delta_x\sin\theta_B\right]+G_y \end{aligned} \quad (8)$$

$$\begin{aligned} z_m=&\left(x_p-G_x\right)\left\{\cos\theta_C\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)+\Delta_z\sin\theta_B\right]+\sin\theta_C\left[\Delta_z\Delta_y\left(1-\cos\theta_B\right)+\Delta_x\sin\theta_B\right]\right\}+ \\ &+\left(y_p-G_y\right)\left\{-\sin\theta_C\left[\Delta_x\Delta_z\left(1-\cos\theta_B\right)-\Delta_y\cos\theta_B\right]+\cos\theta_C\left[\Delta_z\Delta_y\left(1-\cos\theta_B\right)+\Delta_x\cos\theta_B\right]\right\}+ \\ &+\left(z_p-G_z\right)\left[\Delta_z^2\left(1-\cos\theta_B\right)+\cos\theta_B\right]+G_z \end{aligned} \quad (9)$$

Finally, given values of all records of CL data, by using (5)-(9) we can calculate the axis displacements (x_m, y_m, z_m, B, C), respectively.

III. ALGORITHM AND SYSTEM INTEGRATION

The algorithm of the postprocessing program is as following.

- Step 1: reading CL data (the output of CAMs) record by record.
- Step 2: for each record, performing (5)-(9).
- Step 3: building G-code format command according to the results of Step 2, and repeat Step 2 until the end of CL file.

The program is tested and integrated with CAM software (ProE 3.0) to produce G-codes file controlling DMU CNC machine. Below is an example.

- Step 1: CL data generation in ProE (suppose that half of a sphere is machined).

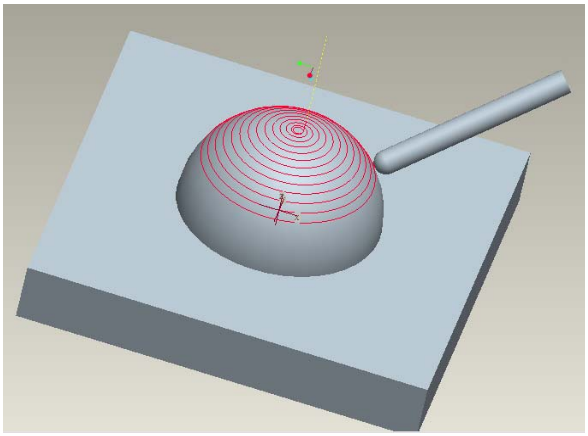

Figure 5. The tool path for machining haft of a sphere

```
$$*        Pro/CLfile  Version Wildfire 3.0 - M070
$$-> MFGNO / MFG0004-MATCONG
PARTNO / MFG0004-MATCONG
$$-> FEATNO / 29
MACHIN / UNCX01, 1
$$-> CUTCOM_GEOMETRY_TYPE / OUTPUT_ON_CENTER
UNITS / MM
LOADTL / 1
$$-> CUTTER / 12.000000
$$-> CSYS / 1.0000000000, 0.0000000000, 0.0000000000, 0.0000000000, $
        0.0000000000, 1.0000000000, 0.0000000000, 0.0000000000, $
        0.0000000000, 0.0000000000, 1.0000000000, 0.0000000000
MULTAX / ON
SPINDL / RPM, 3000.000000, CLW
RAPID
GOTO / -4.4924910505, 0.0000000000, 120.0000000000, $
-0.6417844358, 0.0000000000, 0.7668850879
RAPID
GOTO / -4.4924910505, 0.0000000000, 95.6046049468, $
-0.6417844358, 0.0000000000, 0.7668850879
FEDRAT / 5.000000, MMPM
GOTO / 0.0000000000, 0.0000000000, 90.2364093318, $
-0.6417844358, 0.0000000000, 0.7668850879
GOTO / 6.5456702944, 0.0000000000, 95.2743669780, $
-0.5774149281, 0.0000000000, 0.8164508563
GOTO / 13.6033685557, 0.0000000000, 99.8514046927, $
-0.5103283349, 0.0000000000, 0.8599796455
GOTO / 21.1435281073, 0.0000000000, 103.9311435617, $
-0.4410763363, 0.0000000000, 0.8974695903
GOTO / 29.0902279217, 0.0000000000, 107.4602220529, $
-0.3704212345, 0.0000000000, 0.9288638808
...
```

- Step 2: Running the postprocessing program yields the following G-codes for CNC machine.

```
....
N190 G01 X 5.098 Y .128 Z 110.199 B 4.054 C -1.433
N205 G01 X 7.751 Y .194 Z 57.194 B 4.054 C -1.433
N220 G01 X 7.751 Y .194 Z 50.194 B 4.054 C -1.433
N230 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 7.567
N240 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 16.567
N250 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 25.567
N260 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 34.567
N270 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 43.567
N280 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 52.567
N290 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 61.567
N300 G01 X 7.751 Y .194 Z 50.194 B 4.054 C 70.567

....
```

- Step 3: Machining a workpiece on DMU 50e CNC machine.

Figure 6. Real cut path on 5-axis CNC machine

IV. CONCLUSIONS

The proposed integration method is useful for manufacturing industry where CNC machines of the DMU 50e 5-axis family is used. In the method, the CL data in ISO format produced by every CAMs can be accurately translated into the right format of NC file for controlling the machines. The implementation of real cut parts performed on industrial 5-axis DMU 50e CNC machine at Dong Anh Mechanical Comp., Hanoi, Vietnam proves the success of the system integration. The algorithm proposed can be used for other CAM-CNC systems of which the structure, the control configuration and the kinematics parameters of the CNC machine is the same with the DMU 50e type.

REFERENCES

[1] Chu Anh My et al "On 5-axis freeform surface machining optimization: vector field clustering approach," International Journal of CAD/CAM, Vol. 5, 2005, pp. 1-11.

[2] Chu Anh My, Bohez E. L. J "3D vector field approach for optimization of 5-axis machining," Proceedings of the 8th International Conference on Mechatronics Technology, Hanoi, Vietnam, November 2004, pp. 51-62.

[3] Bohez E, Makhanov S S, Sonthipermpoon K "Adaptive nonlinear tool path optimization for five-axis machining," Int. J. Prod. Res., Vol. 38, No. 17, 2000, pp. 4329-4343.

[4] Makhanov S S, Ivanenko S "A Grid generation as applied to optimize cutting operations of a five-axis milling machine," Applied Numerical Mathematics, Vol. 46, 2003, pp. 353-377.

[5] Makhanov S S, Batanov D, Bohez E, Sonthipaumpoon K, Anotaipaiboon W, Tabucanon M "On the tool-path optimization of a milling robot," Computers & Industrial Engineering, Vol. 43, 2002, pp. 455-472.

[6] Affouard A, Duc E, Lartigue C, Langeron J-M, Bourdet P "Avoiding 5-axis singularities using tool path deformation," International Journal of Machine Tools and Manufacture Vol. 44,, 2004, pp. 415–425.

[7] Fauvel OR, Vaidyanathan J, Norrie DH "An analysis of linearization errors in multi-axis APT-based programming systems," Journal of Manufacturing Systems, Vol. 9(4), 1990, pp. 353–362.

[8] Lee RS, She CH "Developing a postprocessor for three types of five-axis machine tools," International Journal of Advanced Manufacturing Technology, Vol. 13(9), 1997, pp. 658–665.

[9] Deckel Maho "Technical Fundamentals for DMU 50 eVolution and TNC 530," 1997.

[10] She CH, Huang ZT "Postprocessor development of a five-axis CNC machine tool with nutating head and table configuration," Int J Adv Manuf Tech, Vol 38, 2008, pp. 728-740.