

# 工具機數控程式編譯器與應用

## Study of Machine Tool CNC Program Compiler

陳響亮<sup>1\*</sup>、蔣承儒<sup>2</sup>、陳文泉<sup>3</sup>、李峰吉<sup>4</sup>、李桂銘<sup>5</sup>

<sup>1</sup> 國立成功大學 製造資訊與系統研究所 資訊與機電整合實驗室 教授

<sup>2</sup> 國立成功大學 製造資訊與系統研究所 資訊與機電整合實驗室

<sup>3</sup> 工研院機械所 控制核心技術組 副組長

<sup>4</sup> 工研院機械所 控制核心技術組 機電控制整合部 產品經理

<sup>5</sup> 工研院機械所 控制核心技術組 機電控制整合部 研究員

**摘要：** EtherCAT 通訊協定具高速且即時傳輸之特色，已經成為新一代運動控制中重要通訊協定；本文提出工具機數控程式編譯原理，研究數控程式編譯並於 EMP(EtherCAT Motion Platform) 上發展應用，協助 EMP 於數控工具機之發展。依照實機測試結果，輸入的數控程式與運動控制的路徑相同，可驗證程式編譯之正確性。在本研究的編譯系統中，編譯系統以混合式編譯的方法，使用語彙分析、語法分析和語意分析等編譯分析的步驟，將數控程式編譯為中間語言，並直譯為 EMP 運動控制系統使用的運動控制函式。

**Abstract :** EtherCAT is a high performance fieldbus protocol and has become one of the major protocols in motion control. This paper presents a study of machine tool CNC program compiler theory to facilitate the development of CNC machine tools. To validate the accuracy of the method, input digital controls are set to be the same as these made by CNC program. In this study, a compiler system is developed according to the method of hybrid compiler. The compiler system analyzes CNC programs according the order of lexical analysis, syntax analysis and semantic analysis and to generates intermediate code. In the last phase of the compiler system, intermediate code is interpreted to motion control code.

**關鍵詞：** GM 碼指令、數值控制、編譯器

**Keywords :** GM Code Instruction, Numerical control, Compiler

### 前言

目前在製造產業中，精密的機械加工多使用 CNC 數控工具機。加工生產之零件可以大量使用於航太、汽車、醫療設備、光學設備、鐘錶等等不同應用範疇。隨著製造產業蓬勃發展，世界各國的數控工具機之市場也因此不斷成長，吸引眾

多投資。數控工具機之價值，主要取決於開發成本最高的工具機控制器，世界大廠如：日本發那科、德國西門子、海德漢 ... 等，在控制器領域居於領導地位。

各家控制器都提供使用者以程式輸入的方式操作工具機，以程式控制工具機的切削路徑。這些程式稱為數控程式語言或數控指令 (NC

Instruction)。其中 GM Code 是最常見的一種數控指令，除了主要的幾個指令外，各家開發廠會額外自訂指令，因此衍生出多種 GM Code 版本。GM Code 目前被大多數的開發廠採用，且多數的製造工廠也廣泛使用 GM Code。本系統將 GM Code 轉換為 MCCL 運動控制函式 (Motion Control Command Library, MCCL)，並整合工研院機械所研發之 EtherCAT 運動控制平台 EMP，拓展 EMP 於數控工具機使用之彈性。

### 目的

數控指令編譯系統基於 EMP 開發，提供使用者輸入 G Code 的方式進行數控加工。為達成此一目的，本系統發展包含以下關鍵技術：

#### 1. 行動化數控智慧型人機介面

開發智慧行動裝置端使用者介面，作為此編譯系統之輸入界面，包含 NC 指令輸入、編譯資訊顯示等功能，由智慧行動裝置運行此介面進而執行位於 EMP 上之編譯器 [1]。

#### 2. 編譯數控指令

依照語彙分析、語法分析、語意分析的分析順序，對數控指令進行編譯 [2][3]。編譯模組會檢測使用者輸入的數控指令，避免不符合數控指令的規則，而造成程式非預期中斷。所有數控指令經過編譯處理後，會輸出中間碼並交予直譯模組。

#### 3. 直譯數控指令

接續編譯模組輸出之中間碼，將中間語言直譯為 MCCL 運動控制程式 [4][5]。此直譯模組執行時會進行中間碼分析，但不再作語彙分析、語法分析、語意分析等語法錯誤分析之篩選。

#### 4. 紀錄模組

在使用者執行直譯時，其執行之運動控制命令將紀錄於運動控制命令紀錄檔，方便使用者了解運動控制命令的執行結果。

### 系統架構

本系統基於 EMP 開發，系統之架構如下圖 1 所示，使用者可以輸入數控指令，經過訊息傳輸與編譯的程序，運行對應的 MCCL 運動程式控制馬達。系統包含 3 個模組：

#### 1. 行動化數控智慧型人機介面

- (1) 機構參數設定：使用者可以設定、檢視當前之機構參數，並且在完成參數設定之後將馬達激磁。
- (2) 刀具設定：顯示刀具資訊，以及控制夾刀、放刀等等刀具相關功能。
- (3) 機台座標設定：提供連續移動和吋動兩種方式，用以控制刀具位置；加工前可以控制刀具移動到工件邊緣，以對準程式原點。
- (4) 程式編輯：指定數控指令程式檔案的路徑，並且在此確認或編輯程式內容。
- (5) 程式執行模式：將數控指令的路徑當作參數，傳給編譯與直譯模組進行編譯，最後執行加

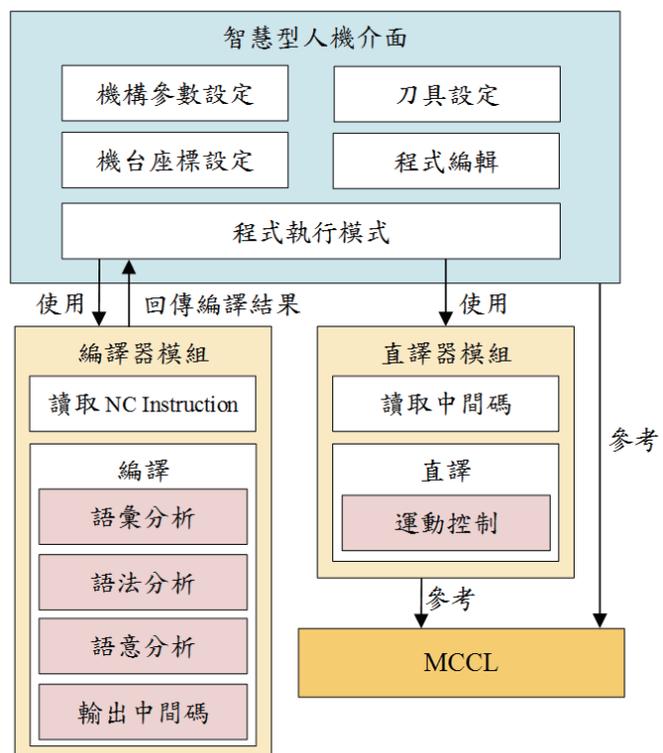


圖 1 數控指令編譯系統架構圖

工程式。此功能可以透過另外 2 個模組，將輸入的數控指令轉為 MCCL 運動程式並運行，並且在加工的同時以圖形化的方式展現加工路徑。

### 2. 數控指令編譯器模組

- (1) 讀取數控指令：依照使用者在人機介面所設定的路徑，將數控指令檔的內容讀取至程式記憶體。
- (2) 編譯：對於輸入的內容進行分析並轉換為中間語言，最後輸出並儲存於中間碼檔案。

### 3. 數控指令直譯器模組

- (1) 讀取中間碼：依照編譯器輸出時的路徑讀取中間碼檔案，並將中間碼檔案的內容讀取至程式記憶體。
- (2) 直譯：經由直譯器將中間碼直譯為運動控制程式。

## 行動化數控智慧型人機介面

本系統使用 Android Studio 與 Visual Studio C# 開發，並應用 SKETCH 繪圖軟體進行 UI 設計。設計架構分為以下五大模式：機構參數設定模式、刀具設定模式、機台座標設定模式、程式編輯模式、程式執行模式。人機介面功能樹狀圖如圖 2，各模式簡述如下。

### 1. 機構參數設定模式

機構設定模式的主要功能為機台參數設定與動作啟用。本模式可以設定或載入機構參數，當設定完成後可點選 Sevro On 讓馬達激磁。

### 2. 刀具設定模式

刀具設定模式的主要功能為：選擇換刀模式與選擇加工刀具。換刀時呼叫 MCCL I/O 控制函式鬆開、夾緊筒夾，使用者可手動選取刀具進行換刀。選擇加工刀具功能，可將刀具規格、刀具

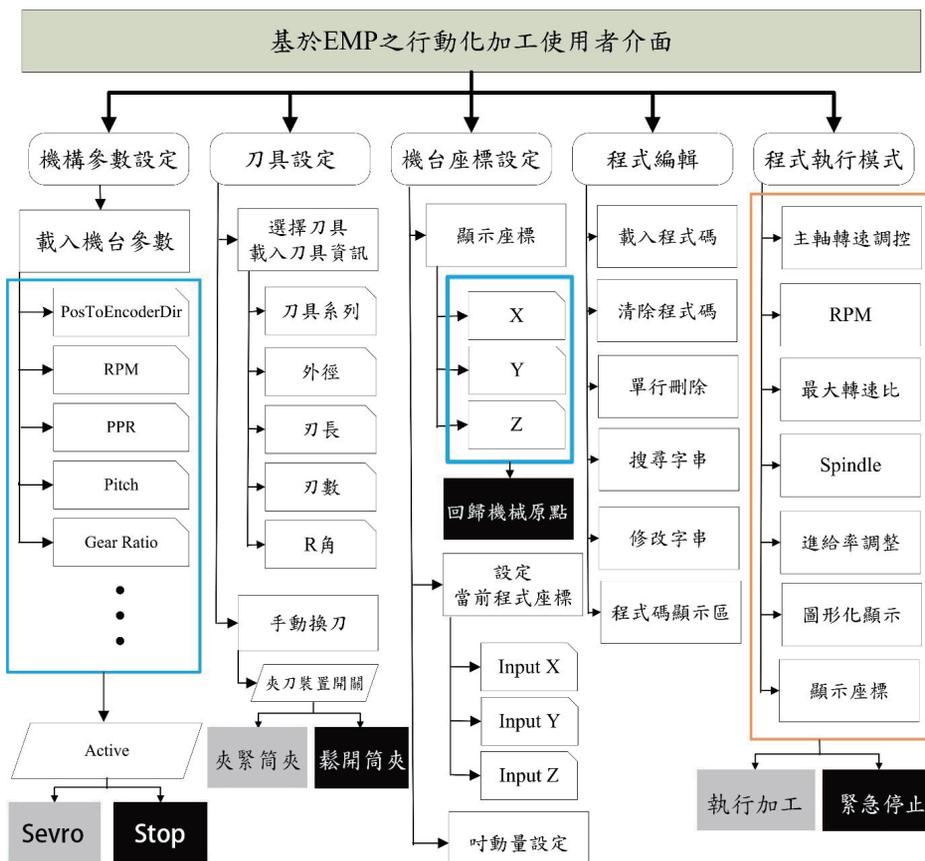


圖 2 人機介面功能樹狀圖

系列、外徑、刃長、刃數、R角、螺紋規格、螺距、柄徑、全長、球半徑、有效長等資訊，顯示於介面提供使用者確認當前欲進行加工的刀具。

### 3. 機台座標設定模式

機台座標設定模式主要功能為提供使用者進行機台微調，並可透過此模式設定工件加工起始位置。功能含機台進行 JOG-XYZ 三直交方向的運動調整、回歸機械原點、定義當前座標為程式座標原點、設定當前程式座標、吋動量設定等功能。

### 4. 程式編輯模式

程式編輯模式主要功能為載入由 CAM(Computer Aided Manufacturing) 軟體產生的數控指令並顯示於介面，提供使用者檢視與修改指令。

### 5. 程式執行模式

程式執行模式主要有調控主軸轉速、進給率調整、顯示當前主軸轉速、當前最大轉速比、顯示當前機械座標、顯示當前程式座標、執行加工、運動軌跡顯示圖等功能。系統運行時呼叫編譯模組與直譯模組中的函式，將數控指令轉換為 MCCL 運動函式進行加工。

## 數控指令編譯模組

為了分析使用者輸入的數控指令，以及檢測

控制器不支援 CAD/CAM 軟體輸出的數控指令，本系統使用編譯模組可提早發現輸入的數控指令是否正確；編譯模組利用前置處理、語彙分析、語法分析、語意分析之順序檢測輸入，最終產生中間語言。智慧型人機介面、編譯模組、直譯模組三者之編譯流程如圖 3。

以下對數控指令編譯模組做詳細介紹

#### 1. 語彙分析：

對於數控指令的輸入，語彙分析將數控指令去除空白、拆解並分類成數個 Token；Token 後續將會提供給語法分析使用。Token 的分類包含 PER、NCODE、GCODE、GCODE'、MCODE、TCODE、FCODE、AXIS、SCODE、EOT 和 UNDEF 一共 11 種，對應內容如下：

- (1) PER 代表檔案的開頭與結束符號。只有 '%' 字元屬於此類，GM Code 檔案的開頭與結尾各有一個 '%' 字元。
- (2) NCODE 代表行號，N0010、N0020 等等行號指令屬於此分類。
- (3) G 指令可以被分類至 GCODE 和 GCODE' 兩個 Token。其中 GCODE 代表需要搭配座標值使用的 G 指令，G00、G01... 等屬於此分類，是主要的運動行為。
- (4) GCODE' 代表不需要搭配座標值使用的 G 指令，如 G09、G17、G18 和 G19... 等。

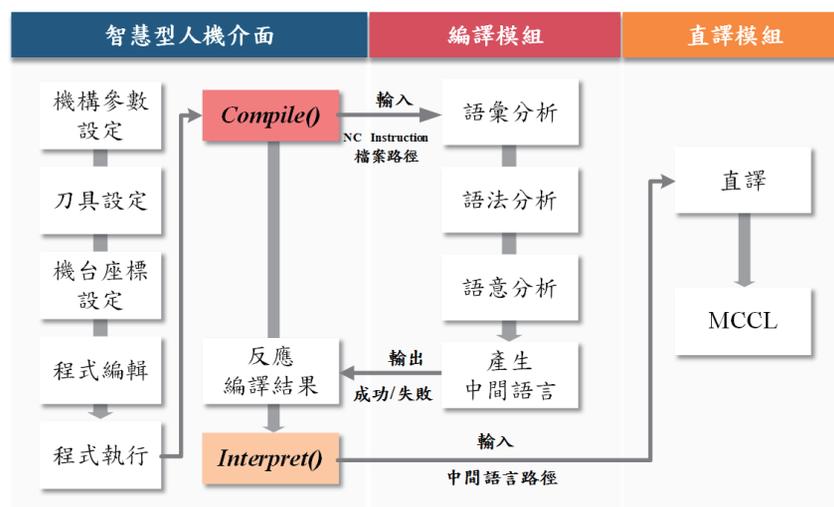


圖 3 編譯流程

- (5) MCODE 是輔助機能，如程式停止，換刀、刀具夾刀、刀具鬆刀等功能。
- (6) TCODE 是刀具索引編號。
- (7) FCODE 是進給速率，在 GM Code 中作為進給速率設定的 F 指令屬於此分類。
- (8) AXIS 是座標軸值，X、Y、Z、I、J、K 和 R 等等的座標軸值指令屬於此分類。
- (9) SCODE 是主軸轉速設定。
- (10) EOT 是 end of token 的縮寫，換行符號 '\n' 屬於此分類。
- (11) UNDEF，當出現不屬於以上 Token 分類之文字時，歸於此分類。

2. 語法分析：

依據 GM Code 的語法，建立一個文法如下圖 4。文法的建立是使用 LL(1) [6] 的規則，以由左至右的方向，一次使用一個 token 做最左推導。圖 4 左半部是 Goal、E、t1、t2、feed、list 和 list' 等 Nonterminal(非終結符號)及其語法的規則。右半部將語彙分析使用的 Token，如：NCODE、

GCODE、GCODE'、FCODE、AXIS、EOT 等作為 Terminal(終結符號)。

Nonterminal 可以藉由推導得到 Terminal，當推導得到的 Terminal 可以與輸入的一連串 Token 有相同結果時，這一連串 Token 就是符合語法分析的。每次使用文法進行推導時，會記錄使用的文法，並在完成語法分析後，將每個 GM Code 單節使用過的文法交給語意分析。

文法是依據 GM Code 的語法而來，建立文法前需要先確認 GM Code 語法。依照 GM Code 的說明，可以整理出每一單節 GM Code 的語法且必須符合以下敘述：

- (1) 每個單節由 N 指令開頭。
- (2) 當一個單節的指令有包含 G 指令時，需至少符合以下 a 或 b 其中一條。
  - i.2a. 部分 G 指令需要配合座標值，例如 G00、G01、G02 和 G03 等；此時座標值指令必須位於 G 指令的右側，且每一單節只會有一個此分類的 G 指令。

<p><b>Goal :</b> → E S</p> <p><b>E :</b> r1. → NCODE t1 t2 feed</p> <p><b>t1 :</b> r2. → GCODE' t1 r3. → □</p> <p><b>t2 :</b> r4. → GCODE t1 list r5. → list r6. → □</p> <p><b>feed :</b> r7. → FCODE r8. → □</p> <p><b>list :</b> r9. → AXIS list'</p> <p><b>list' :</b> r10. → AXIS list' r11. → □</p>	<p><b>NCODE :</b> 行號</p> <p><b>FCODE :</b> 切削速率</p> <p><b>GCODE :</b> 使用時需要配合座標值(AXS)的G Code</p> <p><b>GCODE' :</b> 不需要配合座標值的G Code</p> <p><b>AXIS:</b> 座標值</p> <p><b>□:</b> 空集合</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

圖 4 完整 G Code 文法

ii.2b. 其餘 G 指令不需要配合座標值，一行指令允許存在多個此分類的 G 指令。

(3) 一行 GM Code 可以不含有 G 指令，只存在座標值指令。

(4) F 指令只出現於行末。

建立 GM Code 文法是將以上的敘述，應用於文法的規則中，本系統建立文法時，將一個單節的 GM Code 視作一段陳述 (Expression，記作 E) 和一個終結符號：

**Goal : → E EOT**

E 的語法結構是以 N 指令開頭，加上 t1、t2 與 feed。t1 與 t2 的推導規則對應上方的敘述 2~3，feed 的推導規則對應敘述 4。

**E :→ NCODE t1 t2 feed**

t1 可以推導出零或至少一個符合敘述 2b 的 G 位址 (記作 GCODE')。t1 包含兩條規則，有兩種可行的推導，分別是「GCODE' t1」和「ε」。其中「ε」代表空字串。

**t1 : → GCODE' t1**

**→ ε**

t2 可以推導的語法結構符合敘述 2a 或 3 其中之一，其中第一條規則符合敘述 2a，可以產生需要搭配 G 指令 (記作 GCODE) 的座標值 (記作 list)，第一條規則的中間含有一個 t1，表示 GCODE' 可以出現於 GCODE 的右側、座標值的左側；第二條規則符合敘述 3，推導出一組座標值。

**t2 : → GCODE t1 list**

**→ list**

**→ ε**

feed 可以推導出 F 位址，或推導出空字串。

**feed :→ FCODE**

**→ ε**

在此文法中，list 代表座標值，可以對應於敘述 2a 或 3。座標值不可以是空字串，list 必須推導出一個以上的 AXIS；因此在 list 以外，也加入一

個 list'，使 list 在推導時必定會推導出一個以上的 AXIS。

**list : → AXIS list'**

**list' : → AXIS list'**

**→ ε**

3. 語意分析：

語意分析是使用語彙分析與語法分析的結果，補足 GM Code 省略的語意，並以語法分析建立的語法紀錄作為參數，進行以下兩種工作：

A. 判別 modal G code 的有效範圍，補足欠缺的語意；G code 分為 modal G code 和 one-shot G code，如下表 1。

one-shot G code 表示此 G code 只在一個單節內有效；而 modal G code 表示此 G code 在執行後可以繼續維持，直到另一個相同 Group 的 G code 下達為止。

B. 檢查 one-shot GM Code 的使用條件，例如：檢查 G27(參考點復歸)與 G28(從參考點回歸)，G28 使用到的中點座標需要由 G27 輸入，可以利用語意分析避免 G28 被單獨使用。

### 數控指令直譯模組

經過分析的步驟之後，編譯器根據數控指令陣列和 modal 狀態輸出中間語言；中間語言採用與 MCCL 函式相同的名稱，函式名稱後方的參數是由 GM Code 的座標值轉換而來。

得到中間語言後，由直譯器運行對應於中間語言的 MCCL 運動函式，直譯器依照中間語言的內容，一行一行執行對應的運動函式，控制工具機進行加工。MCCL 運動控制函式庫依照輸入之順序，將命令放置於運動命令佇列，在前一筆命令完成之前，其餘的運動命令會在庫存裡等待，和 GM Code 的執行順序相同，所以直譯器可以一行一行將中間語言直譯為對應的 MCCL 運動程式並且執行；大部分運動類型的中間語言輸入，可以直接運行，例如 MCC\_Line、MCC\_PtP 等等。

但是並非所有的 MCCL 函式，都會加入這一個庫存。例如 MCC\_EcatHome 函式，不會與

表 1 G Code 分群

Group	分類	主要功能	G Code
0	one-shot	暫時停止、原點復歸、刀具補償	{G04, G09, G27, G28, G29, G30, G31, G45, G46, G47, G48, G52, G53, G65}
1		運動	{G00, G01, G02, G03, G33}
2		平面選擇	{G17, G18, G19}
3		座標模式	{G90, G91}
4		軟體行程檢查	{G22, G23}
5		進給速率模式	{G93, G94}
6		單位 (mm 或 inch)	{G20, G21}
7		切削補償	{G40, G41, G42}
8		刀具長度補償	{G43, G44, G49}
9	modal	切削循環	{G73, G74, G76, G80, G81, G82, G83, G84, G85, G86, G87, G88, G89}
10		切削循環回歸模式	{G98, G99}
11		比例功能	{G50, G51}
12		模式巨集呼叫	{G66, G67}
13		表面速度控制	{G96, G97}
14		座標系統選擇	{G54, G55, G56, G57, G58, G59}
15		切削模式	{G61, G62, G63, G64}
16		座標旋轉	{G68, G69}
17		極座標	{G15, G16}

MCC\_Line 函式進入同一個庫存，導致 MCC\_EcatHome 執行的時間點與預期不同；直譯器運行這些 MCCL 運動程式時，需要等待庫存的運動命令清空才可以執行。

## 研究結果

本系統以 Visual Studio 2013 作為開發環境，建立 engraving(C# 語言)、CompilerFuncsDLL(C++ 語言)、InterpreterFuncsDLL(C++ 語言) 三個專案程式。三個專案各別對應於「數控智慧型人機介面」、「數控指令編譯模組」與「數控指令直譯模組」。

本系統以三軸銑床工具機為例，建置一基於 EMP 之數控工具機人機介面，其包含「機構參數設定模式」、「刀具設定模式」、「機台座標設定模式」、「程式編輯模式」、「程式執行模式」，各模式操作流程如下所述。

### 機構設定模式：

1. 點擊按鈕「載入參數」，將機構參數顯示於介面，並可於欄位中進行機構參數的修改與儲存，

確認機構參數無誤後，則點擊按鈕「Sevro On」使機構馬達激磁。機構設定模式之介面如圖 5。

### 刀具設定模式：

本模式含四個區塊，新增 / 修改刀具資料、選擇刀具、手動換刀。刀具設定模式介面如圖 6。

1. 新增 / 修改刀具資料：可以於左方區塊填選刀具各項規格於對應欄位中，進行刀具資料的新增與修改。
2. 選擇刀具：於中央區塊提供使用者確認加工刀具各項資訊。
3. 手動換刀：於右上角區塊，使用者可以利用「夾緊筒夾」與「鬆開筒夾」功能進行手動更換刀具。

### 機台座標設定模式：

本模式含有三個區塊：機械座標、程式座標和吋動量設定。機台座標設定模式之介面如圖 7。

1. 機械座標：進行 XYZ 三直交方向的運動調整與主軸回歸機械原點，此外可透過輸入座標將主軸移至所要特定位置。
2. 程式座標區：顯示當前程式座標，並可點擊按鈕「定義當前座標為程式原點」作為工件加工的



圖 5 機構參數設定模式



圖 6 刀具設定模式



圖 7 機台座標設定



圖 8 程式編輯模式



圖 9 程式執行模式

刀軸起始原點設定。

3. 吋動量設定：輸入吋動量值進行 XYZ 三直交方向的運動微調。

**程式編輯模式：**

程式碼選取後點按載入程式碼，可將 CAM 軟體產生的數控指令顯示於程式碼顯示區，並可透過點擊按鈕「清除程式碼」、「單行刪除」、「收尋字串」、「修改字串」、「儲存」，提供使用者檢視與修改加工指令，程式編輯模式之介面參考如圖 8。

**程式執行模式：**

程式執行模式可進行調控加工主軸轉速與進

給率調整，並顯示當前主軸轉速、當前最大轉速比、當前機械座標、當前程式座標，提供使用者檢視與調控，並於使用者確認前置作業完畢後執行加工。此外於加工期間，可透過運動軌跡顯示圖顯示當前加工路徑，程式執行模式之介面如圖 9 所示。

在實際操作時，先利用 Unigraphics NX (UG) 設計要加工的圖形，接著設定加工參數、刀具參數並產生 GM 碼，CAM 模擬路徑圖如圖 10 所示，運動軌跡顯示圖與 CAM 模擬路徑圖的軌跡一致。

最終透過本系統將數值命令讀入，並實際執行加工，成品如圖 11 所示。

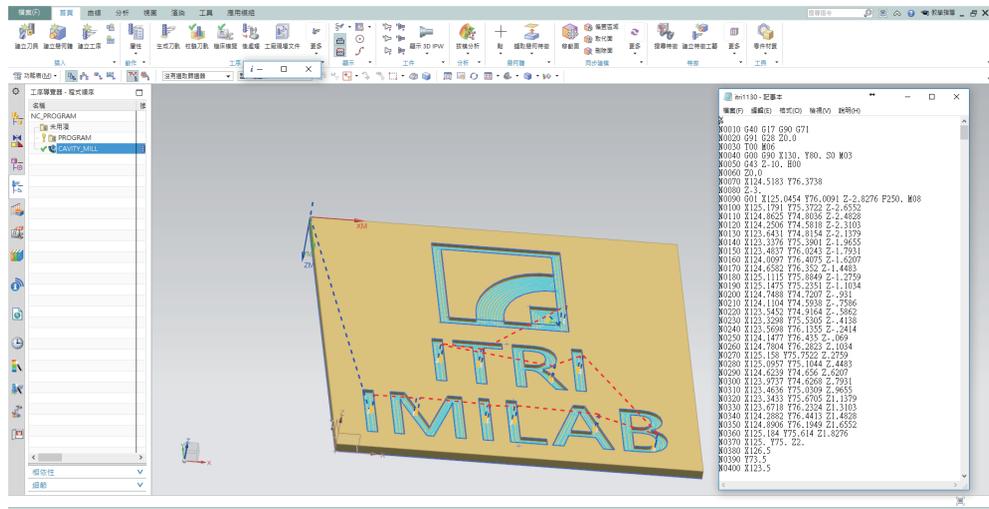


圖 10 利用 UG 設計與產生 GM 碼



圖 11 加工成品

### 參考文獻

- [1] 陳響亮, 陸品丞, “基於 EMP 之行動化系統發展與應用”, 工業技術研究院補助專題研究計畫結案報告, 2016.
- [2] Aho, A., R. Sethi, and J. Ullman, “Compilers: principles, techniques, and tools”, Addison-Wesley Publishing Co., 1986.
- [3] 范劭寧, “CNC 控制器之編譯器原理研究與發展以 PCB 鑽孔機 Excellon 碼為例”, 國立成功大學製造工程研究所碩士論文, 2008.
- [4] H. Hong, D. Yu, X. Zhang, and L. Chen, “Research On a New Model of Numerical Control Program Interpreter”, *IEEE*, 2010, Vol. 2, pp. 467-472.
- [5] X. Lai, “A Design of General Compiler for NC Code in Embedded NC System”, *Industrial Electronics and Applications (ICIEA)*, 2014 *IEEE 9th Conference on. IEEE*, pp.1515-1519.
- [6] Torben Ægidius Mogensen, “LL(1) parsing”, in *Basics of Compiler Design*, Anniversary edition, Copenhagen, Denmark, 2010, ch. 3, sec. 12, pp. 79-87.

### 結論

本系統將數控程式 GM 碼轉換為 MCCL 運動控制函式庫指令, 並實際在 EMP 上進行測試與驗證; 此外也開發簡易版 CNC 控制器, 結合編譯器的設計與實現, 可在使用者介面繪製工具機的加工路徑, 實際加工的成品與 CAM/CAM 軟體設計一致, 證明本系統正確與可行性, 藉此拓展 EMP 於數控工具機使用之彈性。

### 誌謝

感謝工業技術研究院控制核心技术組機電控制整合部(計畫編號 D453RFF431)的支持, 使本計畫得以順利進行, 特此致上感謝之意。