視覺化智慧人機控制介面發展與實現

Development and Implementation of Visual Intelligent Human Machine Control Interface

黃立武[1]、范智翔[1]、陳響亮[2]、黃兆平[3]、曾俊彥[3*]

[1] 國立成功大學 製造資訊與系統研究所 研究生

[2] 國立成功大學 製造資訊與系統研究所 教授

[3] 工研院機械所 控制核心技術組 機電控制整合部 副研究員

摘要：為了降低編輯運動控制程式之專業門檻，並提供易於入門之操作環境，本研究於 PC-Based 控制器中針對六軸機械手臂與生產線周邊設備，提出一個易學、易理解、易操作的視覺化智慧人機控制介面。此人機控制介面採用流程圖之概念設計視覺化程式語言 (Visual programming language, VPL)，使用者可透過拖曳、連結不同功能區塊之方式，實現生產線控制流程之建立與調整。此介面後端以混合式編譯器作為 VPL 之運行核心，並將工研院機械所之運動控制函式庫 (Motion Control Command Library, MCCL) 包裝為 Python 語法之程式應用介面 (Application Programming Interface, API)，使本研究提出之 VPL 可順利轉換為供直譯器執行之控制函式。

Abstract：To reduce the professional threshold of editing motion control programs and provide an easier operating environment, a visual intelligent human-machine control interface is proposed in this research. The interface is easy to learn, easy to understand, and easy to operate in a PC-based controller for the six-axis robot arm and the peripheral equipment of a production line. Through the flowchart-based Visual Programming Language (VPL), users can set up and adjust the control process by dragging and linking different functional blocks. The hybrid compiler is used in the back-end of the interface as the core of the VPL. The Motion Control Command Library (MCCL) of ITRI is packaged as an Application Programming Interface (API) in Python syntax. The VPL can be converted into a control function for literal translation.

關鍵詞：視覺化程式語言、運動控制、人機控制介面

Keywords：Visual programming language, Motion control, Human machine control interface

前言

於智慧製造之製程設計中，機械手臂之運動控制具備相當高之專業及技術要求，須由受過專業訓練之工程師負責此項工作。目前許多機器人大廠皆有發展各自的機器人控制語言，例如 ABB、Funac 與 Kuka 即分別開發出 KRL、RAPID、Karel 等文字形式之控制程式，導致市面上運動控制程式語言的種類繁多，對於不具程式編輯經驗與非專業領域的終端使用者而言，必須付出更多時間與心力方能熟悉不同的系統環境；Alexandrova 等人之研究 [1] 亦顯示，市售的機械手臂軟體環境仍過於封閉，針對不同的系統環境與日益複雜的應用情境，機械手臂的運動程式將更要求由專業的程式人員撰寫。為改善上述問題，本研究將基於 VPL[2] 設計一視覺化人機控制介面，並於其中整合工業機械手臂與生產線周邊設備之控制功能，以期降低智慧製造領域使用者編輯運動控制程式之門檻。

相較於文字形式之程式，VPL 能夠讓使用者不再受限於程式語法，只需排列、連結功能區塊

即可實現程式執行流程的設計，將有助於簡化程式編輯之難度並使程式更易於理解，於教育領域已被廣泛應用於初階程式的教學中 [3-4]，近年來較著名且成功的案例為 Scratch。Scratch 雖然作為一個程式教育軟體，但其獨特之編輯模式為後續積木式 (block-based)VPL 相關研究打下了良好的發展基礎。積木式 VPL 可根據應用情境設計成特定用途之語言，並提供相對簡易的程式編輯方式，但積木式 VPL 的語句排列及語法組成皆有嚴格限制，將可能造成使用者花費更多時間在適應及學習此種 VPL[5]。為了充分發揮 VPL 之優勢，增加可使用之 VPL 類型以表達更多元的程式內容，並且設計具備可擴展性之軟體架構將成為 VPL 系統的發展趨勢。

本研究提出一用於製造領域之視覺化智慧人機控制介面，以基於流程圖之形式 (flowchart-based) 設計此介面之 VPL，並著重於實現可被移植的運動程式。本研究之目的如下：

(1) 降低運動程式編輯困難度：將流程圖形式之 VPL 應用於生產線設備之控制介面，使程式編輯的過程能更加直覺且簡易，以降低生產線中工程師之程式能力需求。

(2) 透過軟體介面整合與控制不同外圍裝置：本研究之人機控制介面將能透過硬體的 I/O 卡輸出與接收外部訊號，以外部訊號作為 VPL 控制流程中的判斷條件，或利用特定之 VPL 圖示輸出控制信號以驅動外部設備，達成藉由此介面實現生產線自動化作業之目的。

(3) 開發支援不同運動控制函式庫之系統：透過後端之混合式編譯器，此人機控制介面可編譯出符合特定程式語法的運動程式。本研究以 Python 語法作為編譯器之目標程式語言，並藉由第三方開源函式庫的輔助，使此介面可引用基於 C#、C/C++ 語法封裝的運動控制函式庫，為系統帶來更高的功能擴充性。

視覺化人機控制介面總體設計

本視覺化智慧人機控制介面以「功能圖示」作為 VPL 之基本元件，各圖示皆具備其獨特的語意，例如執行點對點運動、邏輯選擇、程式起始

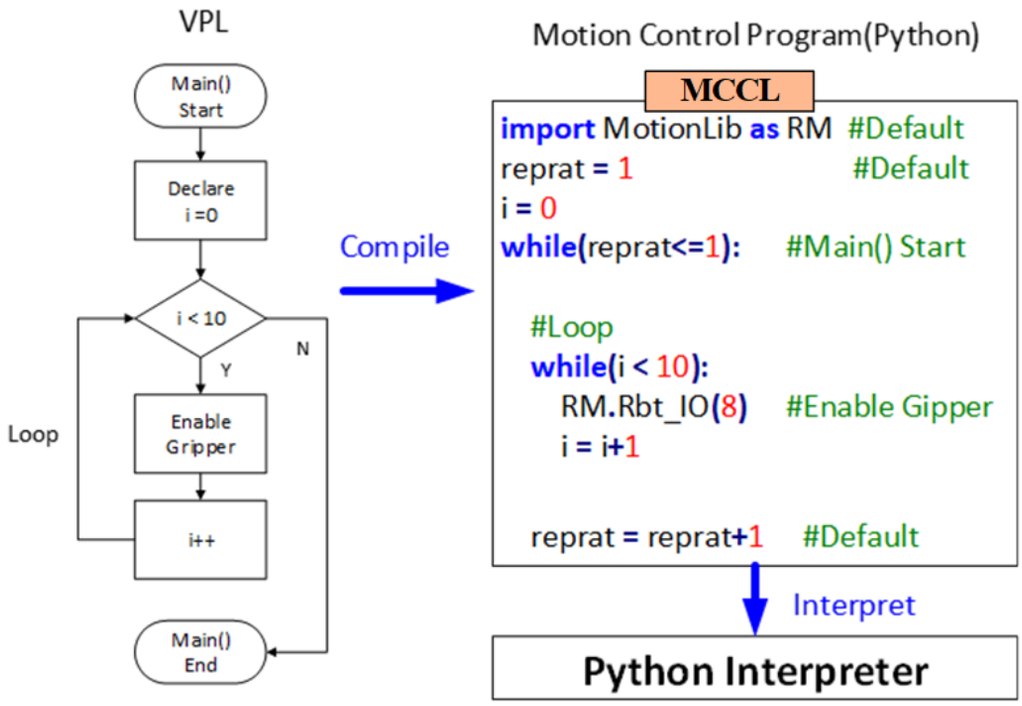

圖 1 視覺化人機控制介面運作方式示意圖

點，並以圖示間相連之線段表示程式執行的順序關係。藉由上述設計，本介面之終端使用者只需要專注於設計機台控制流程及相關設定，而無須知曉專業且繁瑣的程式語法。本介面之運作方式示意圖如**圖 1** 所示，使用者可藉由功能圖示調整特定機台或設備之控制流程，介面後端即可根據各個功能圖示之連結關係，將描述運動控制程序的 VPL 轉換為 Python 運動程式，並藉由 MCCL 第三方的 Python 程式直譯器執行。本介面之 UI 設計則如**圖 2** 所示，包含陳列可用功能圖示之「Toolbox」、編輯 VPL 之「Edit panel」、提供編輯 VPL 功能之「Context menu」、以樹狀結構顯示 VPL 內容之「Monitor window」，以及用於設定功能圖式各項設定值之「Properties window」。

以轉換 VPL 為 Python 語法之中間碼並驅動加工設備為目的，本視覺化智慧人機控制介面之系統模組被分為前端與後端二部分，系統前端包含 VPL 編輯介面與 VPL 中不同種類的功能圖示，讓使用者透過滑鼠拖曳、連結功能圖示的行為，設計並編輯預期執行之 VPL；系統後端則包含 VPL 預處理器、VPL 混合式編譯器，首先透過預處理器將 VPL 轉換為用以記錄程式內容之 XML(Extensible Markup Language) 文件，接著以編譯器中語彙分析、語法分析、語意分析等各個分析步驟確認 VPL 之正確性，最後由 Python 中間碼產生、直譯等步驟執行 VPL 代表之控制程式。

VPL 執行流程將分為使用者行為與系統行為，其中設置工業機械手臂、加工機台、輸送帶等生產線周邊設備以及編輯 VPL 等工作，屬於使用者執行運動控制程式前的準備行為；VPL 執行前的系統初始化動作、解析 VPL、編譯 XML 文件、運行 Python 中間碼與驅動加工設備，則屬於使用者觸發特定事件後系統自動執行之行為。

VPL 功能圖示設計及實現

本視覺化智慧人機控制介面之 VPL 以本研究自行設計的功能圖示所構成，其中功能圖示以流程圖之特性與定義作為基礎的設計概念，並將工業機械手臂與周邊設備所需控制功能融入其中。於實現的系統中，功能圖示分為「程序節點」與

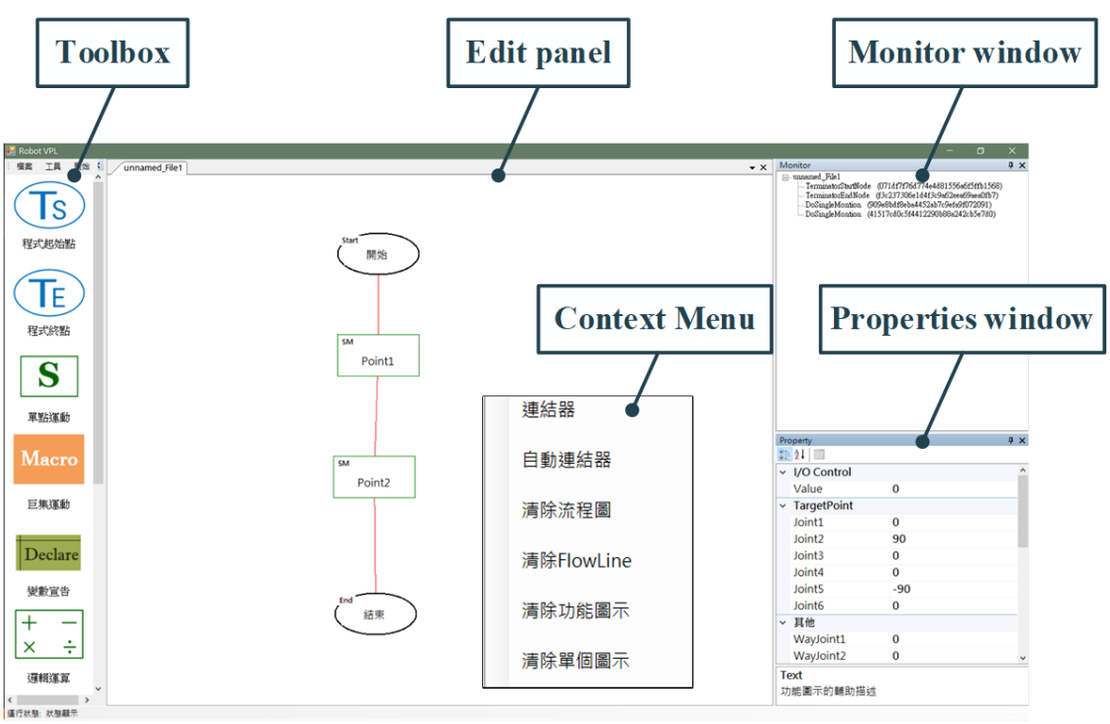

圖 2 視覺化人機控制介面 UI 設計

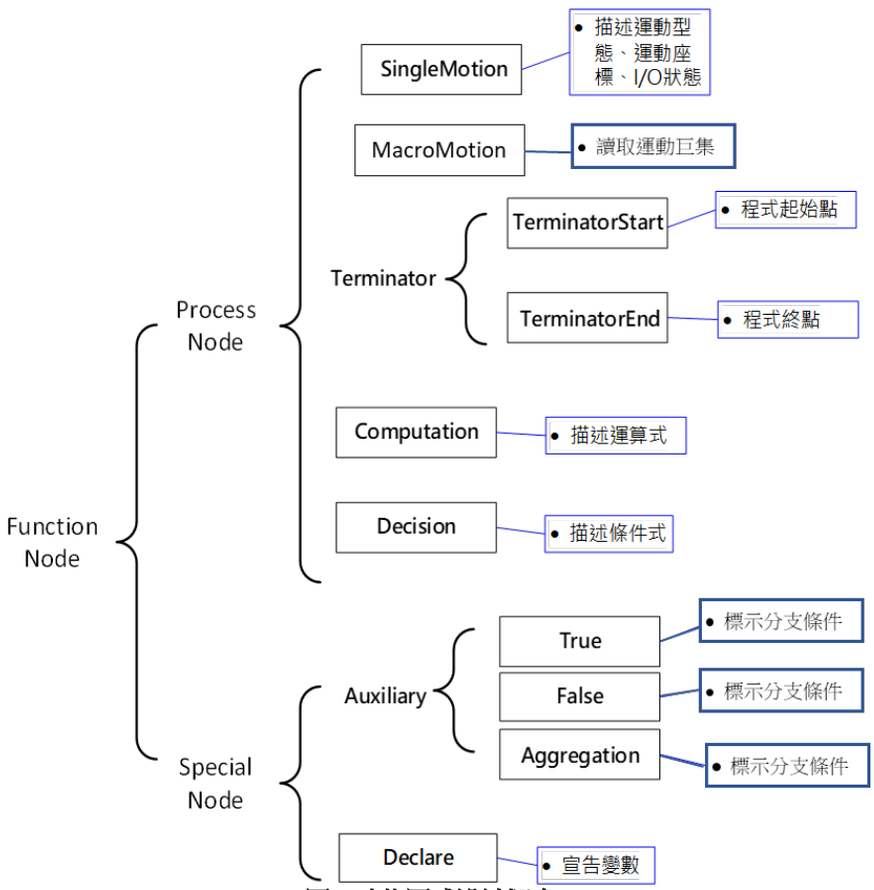

圖 3 功能圖式設計概念

「特殊節點」兩類進行設計，如**圖 3** 所示。其中「程序節點」類型代表構成 VPL 程式主結構之功能圖示，「特殊節點」圖示則用於輔助「程序節點」於表達 VPL 中邏輯判斷結構時使用。透過上述定義明確之 VPL 圖示表達程式內容，並由線條清楚地表達圖示之間的關係，可幫助隱藏程式中的語法規則、使編程方式更簡單。

然而另一方面，自行設計的 VPL 圖示可能造成使用者混淆及辨識不易，因此為增強本介面 VPL 的通用性，各功能圖式之外觀即依照國際規範設計，並以鮮明的顏色及英文縮寫呈現。本研究設計之十種基本功能圖示如**圖 3** 所示，透過這些圖示可確實實現生產線中多項加工設備之整合控制，後續亦可視應用情境實際需求，將更多製程控制或輔助功能包裝為新的圖示加進此 VPL 系統中。**圖 4** 中各個 VPL 功能圖式之程式內容如下：

(1) Terminator 圖示：可宣告 VPL 的執行起始點與終點，亦可用於設定該程式重複執行之次數，於工具箱介面中呈現的外觀如**圖 4(a)**、**圖 4(b)** 所示。

(2) SingleMotion 圖示：可設定單一工業機械手臂或周邊設備之運動控制行為，例如指定點對點運動目標座標、設定運動模式、動作延遲時間等等，介面中呈現的外觀如**圖 4(c)** 所示。

(3) MacroMotion 圖示：可用於載入特定格式之運動指令巨集文件，並根據文件內容目標設備進行完整控制流程，介面呈現外觀如**圖 4(d)** 所示。

(4) Declare 圖示：為程式內宣告變數時使用，變數可用於記錄外部 I/O 訊號或進行流程之條件判斷，介面中呈現的外觀如**圖 4(e)** 所示。

(5) Computation 圖示：用於描述程式中的運算式，

圖 4 功能圖式外觀設計

可針對使用者宣告之變數進行基本之四則運算，介面中呈現的外觀如**圖 4(f)** 所示。

(6) Decision 圖示：用於描述程式中的判斷式，並可構成 VPL 流程圖之分歧結構，使用者可以搭配 Auxilary 圖示建立「選擇結構」或「重複結構」之程式流程，介面中呈現的外觀如**圖 4(g)** 所示。

(7) Auxiliary 圖示：True、False 圖示分別代表條件判斷結果為真以及為否時執行的流程，而 Aggregation 圖示則為代表結束 VPL 分歧流程的標示。上述功能圖示於介面中呈現的外觀如**圖 4(h)、4(i)、4(j)** 所示，並分別以字母 T、F 與 A 標示於圖示中央以利區別。

使用者藉由連結器描述功能圖示之間的關係後，介面中之功能圖示將以直觀的線條與屬性記錄下彼此的關聯性，並組合成使用者所需之 VPL。

VPL 執行架構

本視覺化智慧人機控制介面之 VPL 在被後端模組編譯及執行前，會先由 VPL 預處理器將其轉換為樹狀結構的文字檔，以記錄由多種功能圖示組成的控制結構。本研究採用 XML 作為文字檔的標準格式，並定義功能圖示於 XML 文件中代表的節點，透過文件的表示將能準確的紀錄下圖形化程式之結構。系統後端的運行模組則以解析並執行運動控制 VPL 為目的，運行時將透過 VPL 編譯

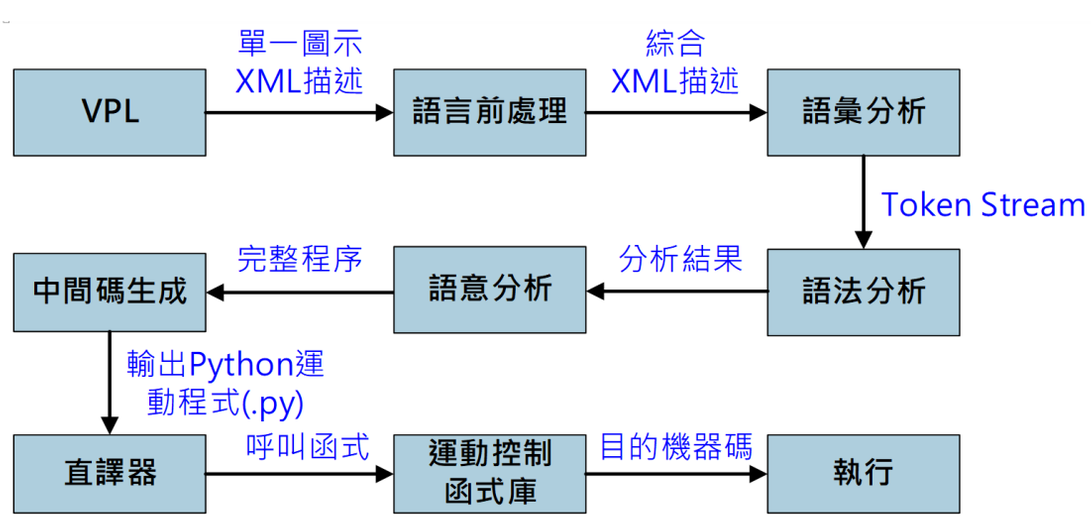

圖 5 系統後端運作流程圖

器解析與翻譯 XML 文件為 Python 程式，再由直譯器運行該程式。系統後端之完整運作流程如**圖 5** 所示。

系統後端軟體架構由預處理器、編譯器、直譯器三個類別構成，而各類別所具備的方法與代表意義如下列所示：

(1) 預處理器類別：由於本介面之 VPL 以 XML 文件表示，需以本類別擷取該程式中有效的資訊，將其轉化為自訂的單節程式，讓後續的分析步驟驗證其正確性。

(2) 編譯器類別：將預處理器類別解析出的單節的字串集合並作為編譯器的原始碼，再分別經由三個分析步驟確保語法的正確性。本類別由四個模組所組成，照分析步驟排列為語彙分析、語法分析、語意分析與程式碼生成。

(3) 直譯器類別：引用本研究基於 MCCL 所定義之 Python 運動程式 API，並使用第三方直譯器 IronPython 作為運動函式的直譯環境。

測試案例 ──CNC 雕刻機自動化上下料

於本研究之測試案例中，我們於提出的視覺化智慧人機控制介面中，建立基於 VPL 的 CNC 雕刻機自動化上下料作業。於該作業流程中，工件及成品皆由輸送帶傳遞，工業機械手臂則作為輸送帶與 CNC 雕刻機間的搬運媒介，各項設備之間並以 I/O 訊號作為通訊媒介。此案例運作流程如**圖 6** 所示，工件將由輸送帶上之 A 點傳遞至 B 點，並由機械手臂將其夾持至 CNC 雕刻機之加工平台上方 (C 點)，交由 CNC 對其進行模擬之加工動作；CNC 加工完成後，則透過 I/O 控制卡發送訊號通知機械手臂夾持成品，並經由原路徑將成品傳遞回輸送帶之 A 點。

本測試案例之 VPL 如**圖 7** 所示，此 VPL 透過「選擇結構」之方式構成，以表達機械手臂必須於特定條件下才得以作動的情境 (於此案例中，該判斷條件即為 CNC 是否已對工件完成加工)。而在 VPL 中顯示的四個「巨集 (Macro) 圖示」，代表此 VPL 透過引用外部的運動指令文件，實現機械手臂移動至各個工件夾取點所需之一連串動作，各個巨集圖示之說明如**表 1** 所示。為確保本人機控制介面於 VPL 編譯步驟與機械手臂運動控制過程之正確性，本研究於測試案例執行期間，將機械手臂各軸編碼器回傳值以及 Python 中間碼

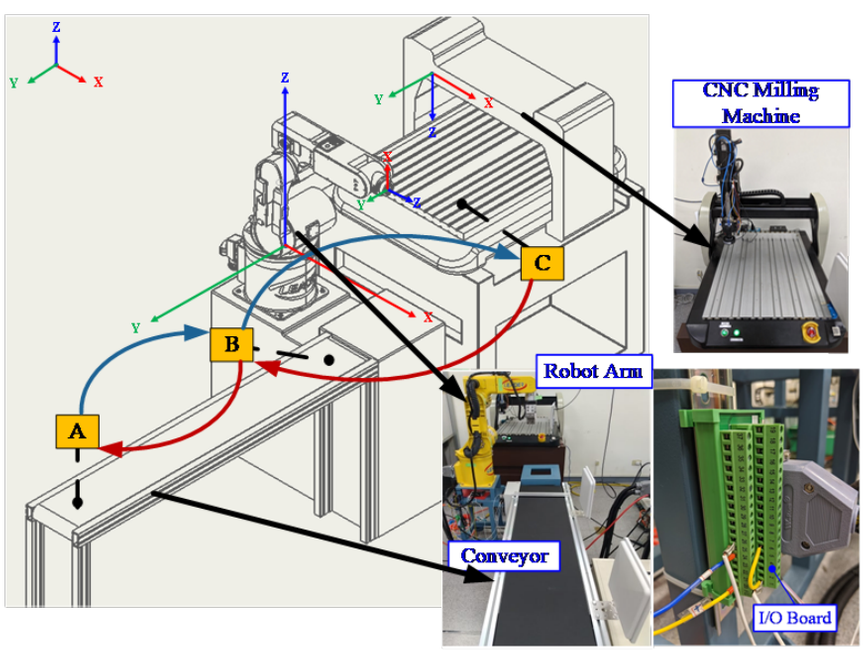

圖 6 CNC 雕刻機自動化上下料案例運作流程圖

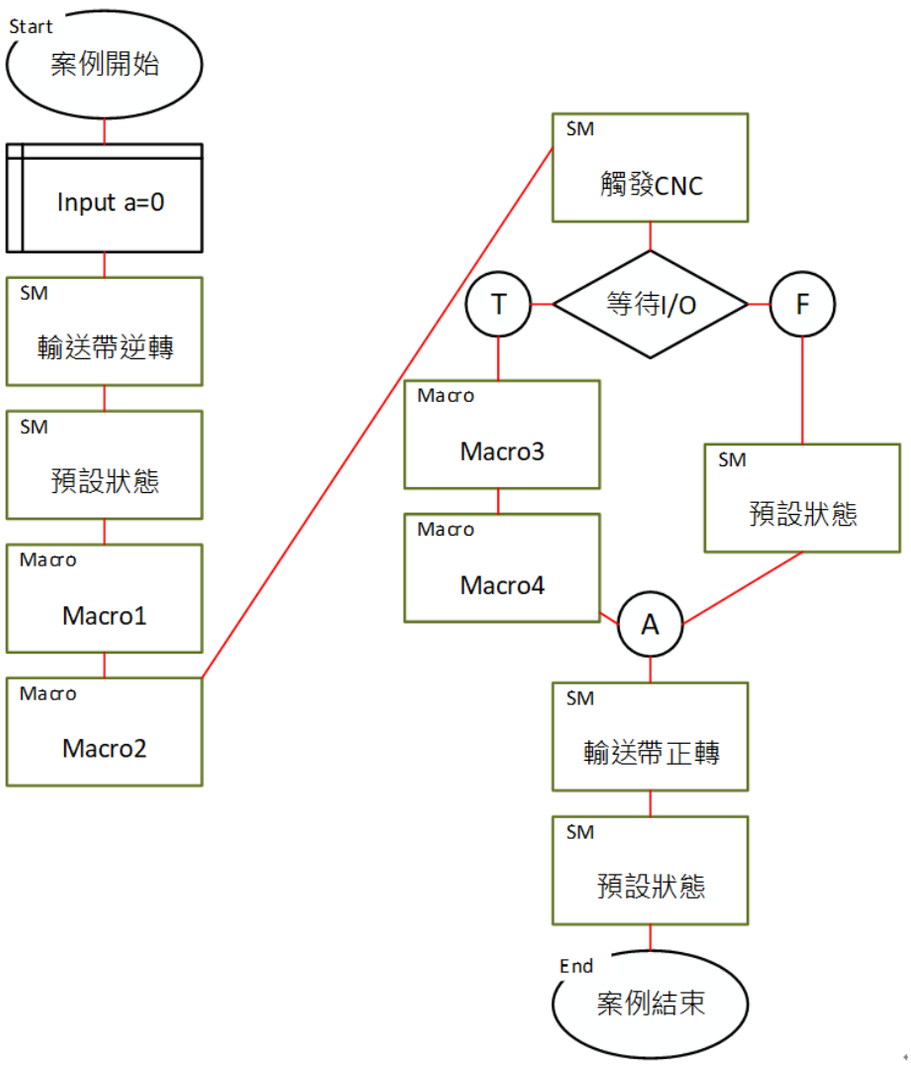

圖 7 CNC 雕刻機自動化上下料案例之 VPL

表 1 測試案例中各巨集圖示說明

巨集編號	程式目標動作	程式內容
Macro 1	機械手臂夾取輸送帶之工件	藉由讀取巨集文件中多筆控制點位及 I/O 輸出值，控制機械手臂由原點移動至輸送帶上工件取放點 (**圖 6** 點 B)，並觸發夾爪夾取工件，最後返回機械手臂原點。
Macro 2	機械手臂放置工件於 CNC 加工平台	藉由讀取巨集文件中多筆控制點位及 I/O 輸出值，控制機械手臂由原點移動至 CNC 加工平台上之加工位置 (**圖 6** 點 C)，並觸發夾爪放置工件，最後返回機械手臂原點。
Macro 3	機械手臂夾取於 CNC 加工平台之成品	藉由讀取巨集文件中多筆控制點位及 I/O 輸出值，控制機械手臂由原點移動至 CNC 加工平台上之加工位置 (**圖 6** 點 C)，並觸發夾爪夾取加工成品，最後返回機械手臂原點。
Macro 4	機械手臂放置成品於輸送帶	藉由讀取巨集文件中多筆控制點位及 I/O 輸出值，控制機械手臂保持夾取成品之狀態由原點移動至輸送帶上工件取放點 (**圖 6** 點 B)，並觸發夾爪放置加工成品，最後返回機械手臂原點。

之控制數值相比對，比對結果皆一致，可證實本人機控制介面具備完善之 VPL 編譯及執行能力。

結論

不同於傳統基於文字程式語言之控制系統，本研究所提出之視覺化人機控制介面將 VPL 導入運動控制領域，以降低非專業領域使用者編輯運動程式之技術要求、簡化實現運動控制邏輯之複雜度為目標，並已完成初步之功能發展與實現。後續開發者可利用本介面之模組化系統架構，將各式先進製程輔助或控制功能，例如智慧人機協作、自動化光學檢測、機台預測性維護等功能，以 API 之形式加入本介面之中，以提升本人機控制介面於智慧製造領域之產業及應用價值。

誌謝

本計畫承經濟部技術處高值金屬成型機械智慧機電整合技術開發計畫 (L353C70000)，並承行政院科技部提供研究經費，計畫編號 MOST109-2221-E006-098，特此致謝。

參考文獻

[1] S. Alexandrova, Z. Tatlock, and M. Cakmak, "RoboFlow: A flow-based visual programming language for mobile manipulation tasks," in 2015 *IEEE International Conference on Robotics and Automation* (ICRA), 2015: IEEE, pp. 5537-5544.

[2] M. Erwig, K. Smeltzer, and X. Wang, "What is a visual language?," *Journal of Visual Languages & Computing*, Vol. 38, pp. 9-17, 2017.

[3] I. Plauska, R. Lukas, and R. Damasevicius, "*Reflections on using robots and visual programming environments for project-based teaching,*" Elektronika ir Elektrotechnika, Vol. 20, no. 1, pp. 71-74, 2014.

[4] R. V. Roque, "*OpenBlocks: an extendable framework for graphical block programming systems*," Massachusetts Institute of Technology, 2007.

[5] D. Weintrop and U. Wilensky, "Comparing block-based and text-based programming in high school computer science classrooms," *ACM Transactions on Computing Education* (TOCE), Vol. 18, no. 1, pp. 1-25, 2017.

[6] 范智翔 , " 機械手臂之圖形化運動控制介面及編譯器發展與實現 ," 成功大學製造資訊與系統研究所學位論文 , pp. 1-81, 2020.