

國立中央大學機械工程學系
畢業專題成果報告書

國立中央大學

『FESTO 產線排程優化技術』

指導教授：林錦德

成員：郭洧銓

成員：黃曜駿

成員：賴柏綸

成員：連爾諾

中華民國 111 年 1 月

摘要

本專題目的為解決智慧工廠排程問題。傳統的排程預測技術僅適用固定標準加工時間的流程，然而有些機台會受環境因素影響，導致沒有一個固定標準工時（如：加熱製程會隨著環境溫度而導致加熱時間長度不同），因此我們引入深度學習技術幫助我們預測這些非固定的標準工時，並結合排程技術縮小傳統排程的誤差，使排程系統能有效最佳化產線加工時間。

我們以排程程式重新規劃加工順序。原先使用的套件僅支援固定加工工時。然而在我們實作之智慧工廠中有許多機台的工時並非定值，其中又以加熱站最為明顯。為了得到更準確的排程結果，我們決定將加熱站內部溫度作為參數放入排程程式中，並藉由深度學習預測加熱站工時。為了在有限時間內收集到足以訓練深度學習模型的資料量。我們開發能穩定自動化收集機台資訊的程式，藉由此程式協助我們完成深度學習模型，並且加入排程程式中。

完成排程後我們控制機台來實現排程結果。先以工廠製造執行系統下工單，我們再以程式覆蓋原先製造執行系統的控制命令，藉此實現排程結果。為了方便比較前後工時差異，另外開發將加工工時資料繪製成甘特圖的程式。

最後，在時間有限的情況下先收集 6 筆工件最佳化前後工時，可以節省將近 14% 的工時。未來我們會繼續收集多種工單比較其最佳化前後差異，並持續優化深度學習模型，期望讓排程預測更加準確。

目 錄

壹、	團隊任務分工.....	1
貳、	研究動機.....	1
參、	研究方法.....	2
一、	加熱機台資料自動化收集程式.....	2
1.	資料儲存格式（每秒紀錄一筆）.....	2
2.	連線失敗提醒.....	2
二、	排程式.....	3
1.	循環神經網路（LSTM）.....	3
2.	OR-Tool.....	5
3.	排程結果甘特圖.....	6
三、	深度學習結合排程系統.....	6
四、	產線控制程式.....	7
五、	自動化收集工單工時與加熱機台預熱程式.....	8
六、	MES4 資料庫（實際加工結果）甘特圖.....	8
肆、	結果討論.....	10
伍、	結論.....	11
附錄一、	程式碼.....	12

圖表目次

圖 1	FESTO 智慧工廠配置圖	1
圖 2	加熱機台資料摘錄.....	2
圖 3	OPC UA 連線逾時提示	2
圖 4	深度學習模型架構圖.....	3
圖 5	深度學習模型架構.....	4
圖 6	訓練完 Loss 與 Learning Rate	4
圖 7	測試集對比結果.....	5
圖 8	排程輸出結果（加工件數量：3）	5
圖 9	排程結果甘特圖範例.....	6
圖 10	深度學習結合排程系統架構流程圖.....	6
圖 11	深度學習模型預測加熱時間結果.....	7
圖 12	紅綠燈.....	7
圖 13	MES 資料庫欄位格式與資料摘錄	8
圖 14	自動化收集工單工時輸出欄位格式與資料摘錄.....	8
圖 15	資料庫讀取成功畫面.....	8
圖 16	指令提示.....	9
圖 17	資料庫歷史工單甘特圖.....	9
圖 18	工單編號查詢指令說明.....	9
圖 19	查詢結果範例.....	9
圖 20	查詢結果範例.....	9
圖 21	英文終端機模式.....	10
圖 22	未優化工時甘特圖，總工時：1622 秒.....	11
圖 23	優化後工時甘特圖，總工時：1391 秒.....	11
附錄圖 1	檔案連結 QR-Code	12
表 1	團隊分工表.....	1

壹、 團隊任務分工

表 1 團隊分工表

成員	工作項目
郭洧銓	深度學習模型建立、OPC UA 操作、收集工單資料。
連爾諾	OPC UA 機台控制程式編寫、收集工單資料。
黃曜駿	排程程式編寫、深度學習模型建立、整合預測模型與排程系統。
賴柏綸	OPC UA 自動擷取資料程式、甘特圖繪製程式編寫。

貳、 研究動機

傳統的排程預測技術僅適用固定標準加工時間的流程，然而有些機台會因為環境因素影響，導致沒有一個固定標準工時（如：加熱製程會隨著環境溫度而導致加熱時間長度不同），因此我們引入深度學習技術幫助我們預測這些非固定的標準工時，並結合排程技術縮小傳統排程的誤差，使排程系統能有效最佳化產線加工時間。

實作場域：

中央大學機械系 FESTO 智慧工廠。為具備 AGV（無人搬運車）之小型產線，可以透過 MES（製造執行系統）下工單進行生產。



FESTO 智慧工廠配置與製程

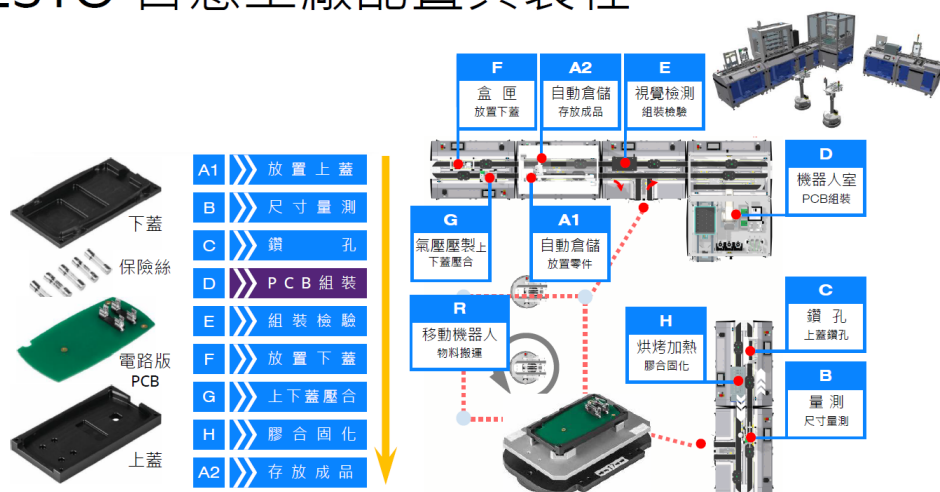


圖 1 FESTO 智慧工廠配置圖

參、 研究方法

此畢業專題使用以下幾種方法進行研究：

1. 深度學習（循環類神經網路 LSTM）。
2. 排程系統（OR-Tools）。
3. OPC UA 通訊協定。
4. Matplotlib 繪圖庫。

我們先撰寫程式利用 OPC UA 協定來自動收集加熱站的資料，再將加熱站經過深度學習輸出模型，而後把模型結合排程系統後得出優化解，並使用 OPC UA 協定控制機台來實現排程結果。結果數據使用自動收集資料程式，透過 OPC UA 將各站數據傳入電腦，最後將排程前及排程後之工時用 Matplotlib 繪成甘特圖。

一、 加熱機台資料自動化收集程式

此程式目標為自動化收集加熱機台資訊，用來作為深度學習模型的學習與驗證。我們運用 OPC UA 協定取得機台資料，再透過邏輯判斷將資訊寫入 Excel 工作表中。此外為了收集多樣化的資料，我們也加入機台的控制針對不同初始溫度下進行加熱，來讓機器學習的結果更準確。詳細資料收集格式如下。

1. 資料儲存格式（每秒紀錄一筆）

收集資料包含：機台當前溫度、當前時間、經過時間、是否在加熱、下一筆資料溫度（在 LSTM 模型中作為 Label）。

	A	B	C	D	E	F
1	NowTemp	Time	Duration	isHeat	NextTemp	
2	31.47424698	2021-11-26 16:22:30	0.011992	0	31.43807983	
3	31.43807983	2021-11-26 16:22:31	1.009669	0	31.39467621	
4	31.39467621	2021-11-26 16:22:32	2.004256	0	31.35850716	
5	31.35850716	2021-11-26 16:22:33	3.000831	0	31.31510353	
6	31.31510353	2021-11-26 16:22:34	3.994044	0	31.27893448	
7	31.27893448	2021-11-26 16:22:35	4.986398	0	31.23553085	
8	31.23553085	2021-11-26 16:22:36	5.977205	0	31.19212914	
9	31.19212914	2021-11-26 16:22:37	6.981925	0	31.14872551	

圖 2 加熱機台資料摘錄

2. 連線失敗提醒

若程式無法與機台正確連線，會顯示提醒字樣如下圖。

```
timed out
請確認是否已連上Taiwan-CP-Factory後再啟動程式
Press 'Enter' to exit...[]
```

圖 3 OPC UA 連線逾時提示

二、 排程程式

由於一般的排程架構（如：OR-Tool）的輸入僅允許固定的參數，因此在預測像是加熱機台對於每一次加工時間不固定的情況，無法進行排程。故我們結合深度學習與排程程式，詳細說明如下。

1. 循環神經網路（LSTM）

由於生產線中的加熱機台對於每一筆工單的加熱時間有所不同，每一次加溫的起始溫度以及環境溫度的不同，均會影響到加熱長度。因此我們在加熱的預測上，建立循環類神經網路模型（LSTM）用以預測加熱每一秒的溫度的長度。

由於工廠加熱機台的特性是加熱到某固定溫度就停止加熱，因此我們設計模型讓模型估計加熱機台內每一秒的溫度，只要知道工單進入機台的起始溫度以及目標加熱溫度，就可以計算出加熱工時。因此我們訓練的模型需要有能夠預測每秒加熱機台內的能力。

我們模型的輸入為：[當前加熱機台內溫度，加熱機台是否在加熱]

輸出為：[下一秒機台溫度]

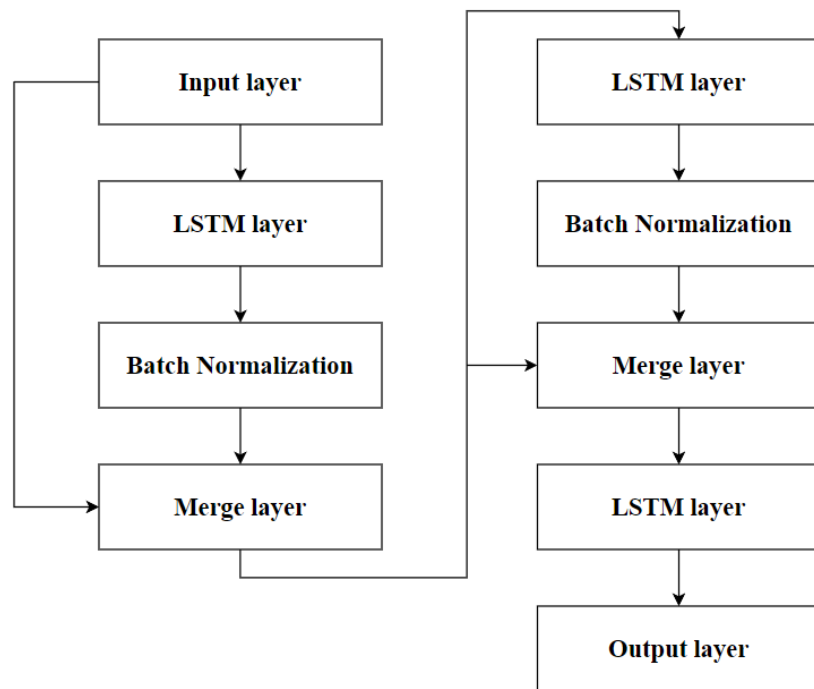


圖 4 深度學習模型架構圖

Model: "LSTM"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 1, 2)]	0	
lstm (LSTM)	(None, 1, 2)	40	input_1[0][0]
batch_normalization (BatchNorma	(None, 1, 2)	8	lstm[0][0]
add (Add)	(None, 1, 2)	0	input_1[0][0] batch_normalization[0][0]
lstm_1 (LSTM)	(None, 1, 2)	40	add[0][0]
batch_normalization_1 (BatchNor	(None, 1, 2)	8	lstm_1[0][0]
add_1 (Add)	(None, 1, 2)	0	batch_normalization_1[0][0] add[0][0]
lstm_2 (LSTM)	(None, 1, 1)	16	add_1[0][0]

Total params: 112
Trainable params: 104
Non-trainable params: 8

圖 5 深度學習模型架構

LSTM 層設定如下：

權重初始化用 Glorot normal distribution

Loss Function 用 Mean Absolute Error 並加上 Ridge regression 限制權重大小

Optimizer 用 Adam $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e - 9$

Learning rate 用 Warm up learning rate scheduling method.

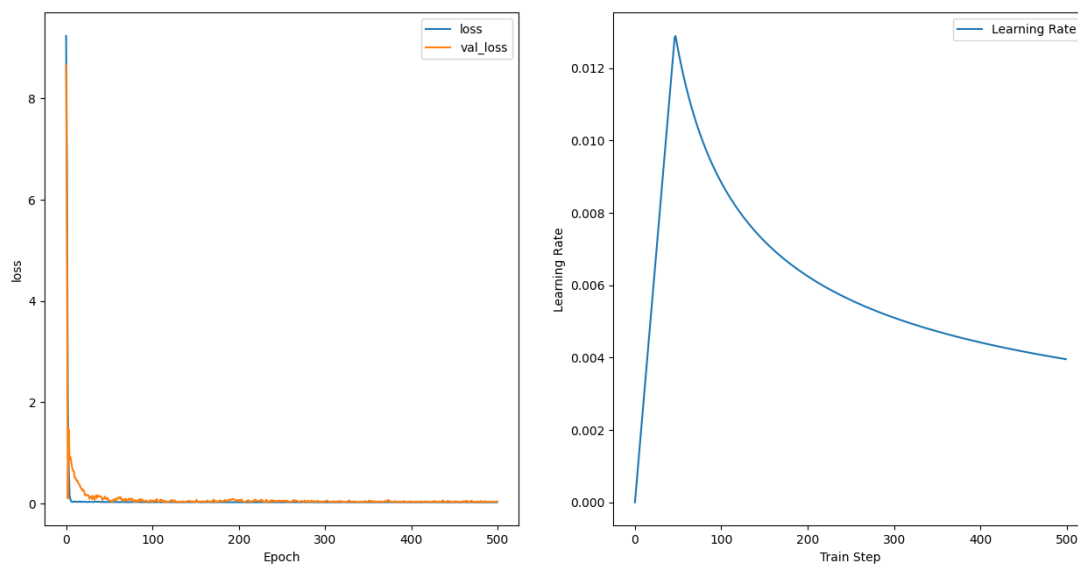


圖 6 訓練完 Loss 與 Learning Rate

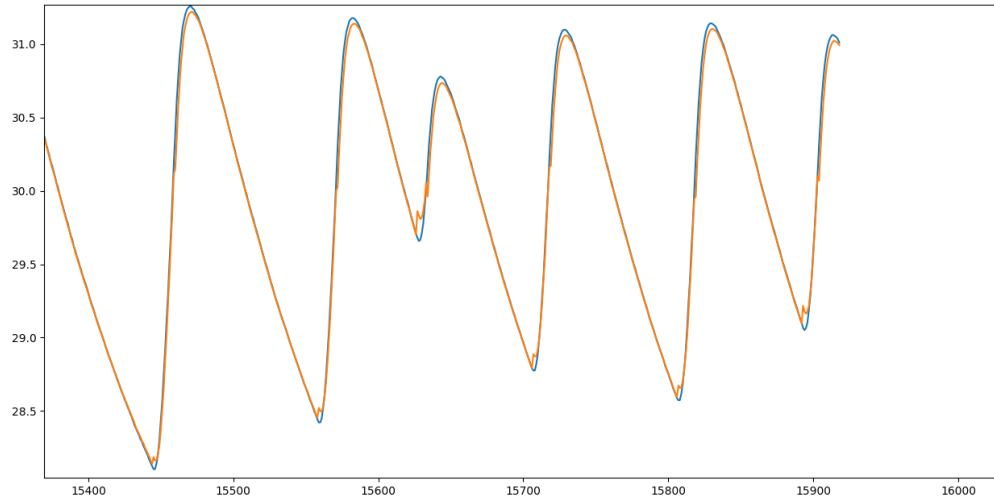


圖 7 測試集對比結果

X 軸：時間（秒）、Y 軸：溫度（℃）

藍色：實際溫度、橘色：模型預測溫度

2. OR-Tool

OR-Tool 為 Google 開發的開源函式庫，用於優化各種 Constrain Problem，支援線性規劃、整數規劃。我們開發的排程程式中，使用到 OR-Tool 的 Constraint Optimization 的優化器以解決 Job Shop 問題。

在使用上必須依照加工順序，依序工單對應的機台編號以及加工時間，做為整個 OR-Tool 排程的輸入，程式碼中預先加入許多不同的限制條件，以模擬整個產線的運作，最後待整個 OR-Tool 優化器運作完畢後，回傳最後的排程結果（如圖 8 所示¹）。

```
Optimal Schedule Length: 964 Seconds
Storage : job_0_0 job_1_0 job_2_0 job_0_12 job_1_12 job_2_12
          [0,15] [15,30] [30,45] [722,736] [814,828] [950,964]
AGV : job_0_1 job_1_1 job_0_4 job_2_1 job_1_4 job_2_4 job_0_9 job_1_9 job_0_11 job_2_9 job_1_11 job_2_11
      [57,106] [152,201] [201,247] [247,296] [296,342] [360,406] [436,482] [531,577] [624,670] [670,716] [716,762] [852,898]
Check part : job_0_2 job_1_2 job_2_2
              [125,128] [220,223] [315,318]
Milling : job_0_3 job_1_3 job_2_3
          [138,146] [233,241] [328,336]
PCB : job_0_5 job_1_5 job_2_5
      [269,341] [364,436] [436,508]
Fuse check : job_0_6 job_1_6 job_2_6
              [370,372] [465,467] [537,539]
Cover : job_0_7 job_1_7 job_2_7
        [398,400] [493,495] [565,567]
Pressure : job_0_8 job_1_8 job_2_8
           [410,426] [505,521] [577,593]
Heat : job_0_10 job_1_10 job_2_10
       [526,613] [621,655] [760,841]
```

圖 8 排程輸出結果（加工件數量：3）

¹ 註：job_X_Y, X 為加工件編號，Y 為當前加工序編號；[A, B], A 為加工起始時間，B 為加工結束時間（單位秒）。

3. 排程結果甘特圖

此程式撰寫動機起因於我們發現原先排程程式的結果僅以文字輸出（圖 9）可讀性較低，因此決定在排程程式中引入「matplotlib」函式庫，我們使用水平長條圖做出甘特圖效果。x 軸為加工時間（秒），y 軸為加工站名稱，各工件以不同顏色區分，總加工時間以秒數為單位顯示於標題上，成果如下圖。

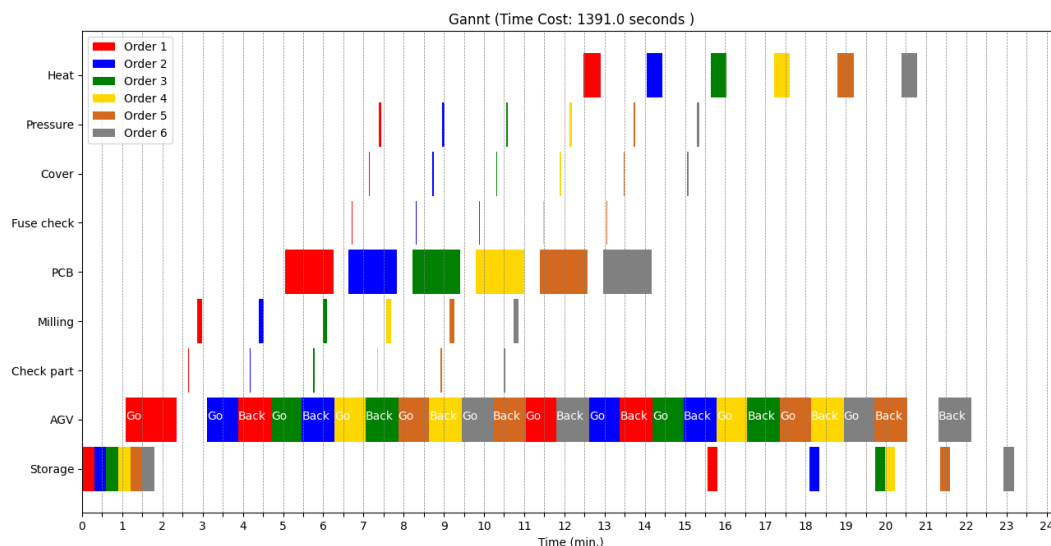


圖 9 排程結果甘特圖範例

三、 深度學習結合排程系統

前面的深度學習是用來預測固定的標準工時，我們訓練的 LSTM 模型可以精準的預測每秒加熱機台內的溫度。我們只需要透過 OPC UA 得知機台初始的溫度，提供給深度學習系統計算出加熱的時間後，排程系統會計算出當前工件的工作時間，並且將當前工件加熱完畢時間以及下一個工件的加熱起始時間交給深度學習系統計算出加熱時間，重複以上動作直到計算完所有機台的加熱時間，最後排程系統會將所有資料一併計算，得出最佳化結果。

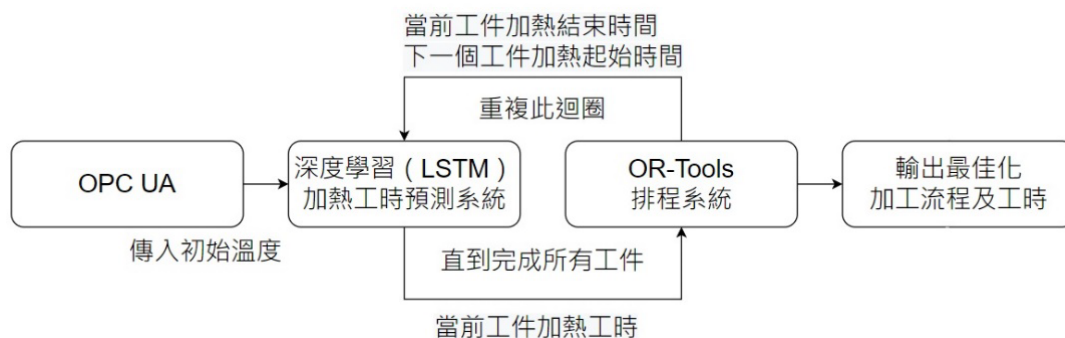


圖 10 深度學習結合排程系統架構流程圖

```

預測加熱機台加工時長
預測次數: 1
oPos= 1 加熱預測時間 = 87
Optimal Schedule Length: 2177 Seconds
預測次數: 2
oPos= 2 加熱預測時間 = 33
Optimal Schedule Length: 2177 Seconds
預測次數: 3
oPos= 3 加熱預測時間 = 69
Optimal Schedule Length: 2177 Seconds
預測次數: 4
oPos= 4 加熱預測時間 = 46
Optimal Schedule Length: 2177 Seconds
預測次數: 5
oPos= 5 加熱預測時間 = 63
Optimal Schedule Length: 2177 Seconds
預測次數: 6
oPos= 6 加熱預測時間 = 50
Optimal Schedule Length: 2177 Seconds
預測次數: 7
oPos= 7 加熱預測時間 = 60
Optimal Schedule Length: 2177 Seconds
預測次數: 8
oPos= 8 加熱預測時間 = 52
Optimal Schedule Length: 2177 Seconds
預測次數: 9
oPos= 9 加熱預測時間 = 58
Optimal Schedule Length: 2177 Seconds
預測次數: 10
oPos= 10 加熱預測時間 = 53
Optimal Schedule Length: 2177 Seconds

```

圖 11 深度學習模型預測加熱時間結果

四、 產線控制程式

以 Python 運用此協定撰寫控制系統、讀取各機台點位資料，並將加工區以程式邏輯設計出類似紅綠燈的概念，以決定載具順序，進而改變 AGV 的運行規律，控制機台以達成排程系統預測之加工流程。

由排程圖可知，只有前兩筆必須連續出發，從第三筆開始都會是一筆回來接著一筆出發。由於 FESTO 工廠的設計是：一旦工件進入等待區，下一車就會被跳過，序位會被排到最後一台車之後。故我們所設計的解決方案就是將其中兩個工作站的氣壓閥，作為紅綠燈，一車通過後就擋住下一車，進而避免下一車加工序列延後。



圖 12 紅綠燈

五、 自動化收集工單工時與加熱機台預熱程式

我們發現使用 FESTO 的資料庫並不會記錄工單中 AGV 花費的時間，因此為了取得完整資料並驗證排程可行性，必須寫出一套取代 MES 的工時記錄程式。

使用 OPC UA 讀取機台節點資訊再透過邏輯判斷，記錄各加工站工時，再寫入欄位與 MES 資料庫相同的 Excel 檔案中（此程式僅填入必要欄位，如圖 14），藉此可以將資料直接附加在原資料庫中，資料庫甘特圖繪製程式亦可沿用。另外在此程式中加入加熱機台預熱功能。邏輯如下：在工件第一次加熱前，若機台溫度小於 28 度時則開啟預熱，若大於 28.5 度則關閉預熱。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	WPNNo	StepNo	OPNo	OPPos	Description	OpNo	NextStepNo	FirstStep	ErrorStepNo	NewPNNo	PlannedStart	PlannedEnd	Start	End
85	9211	90	2050		1 store a part from stopper 1	210	0	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
86	9211	80	2050		1 heating Part	112	90	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
87	9211	70	2050		1 pressing with force regulation	111	80	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
88	9211	60	2050		1 feed back cover from magazine	201	70	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
89	9211	50	2050		1 check part with camera	400	60	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
90	9211	40	2050		1 assemble a PCB with both fuses	304	50	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
91	9211	30	2050		1 drilling both	122	40	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
92	9211	20	2050		1 measure a part (analog)	115	30	FALSE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21
93	9211	10	2050		1 release a defined part on stopper 1	212	20	TRUE	0	0	27-Sep-21	27-Sep-21	27-Sep-21	27-Sep-21

圖 13 MES 資料庫欄位格式與資料摘錄

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	WPNNo	StepNo	OPNo	OPPos	Description	OpNo	NextStepNo	FirstStep	ErrorStepNo	NewPNNo	PlannedStart	PlannedEnd	Start	End
94	9211		2095		1 release a defined part on stopper	212					2021/9/30 20:59		2021/9/30 21:00	
95	9211		2095		2 release a defined part on stopper	212					2021/9/30 21:00		2021/9/30 21:00	
96	9211		2095		3 release a defined part on stopper	212					2021/9/30 21:00		2021/9/30 21:00	
97	9211		2095		4 release a defined part on stopper	212					2021/9/30 21:00		2021/9/30 21:01	
98	9211		2095		5 release a defined part on stopper	212					2021/9/30 21:01		2021/9/30 21:01	
99	9211		2095		1 AGV go	115					2021/9/30 21:01		2021/9/30 21:02	
100	9211		2095		1 measure a part (analog)	115					2021/9/30 21:02		2021/9/30 21:02	
101	9211		2095		1 drilling both	122					2021/9/30 21:03		2021/9/30 21:03	
102	9211		2095		2 AGV go	115					2021/9/30 21:03		2021/9/30 21:04	
103	9211		2095		1 AGV back	304					2021/9/30 21:04		2021/9/30 21:04	
104	9211		2095		2 measure a part (analog)	115					2021/9/30 21:04		2021/9/30 21:04	
105	9211		2095		2 drilling both	122					2021/9/30 21:04		2021/9/30 21:04	
106	9211		2095		3 AGV go	115					2021/9/30 21:04		2021/9/30 21:05	
107	9211		2095		1 assemble a PCB with both fuses	304					2021/9/30 21:05		2021/9/30 21:06	
108	9211		2095		2 AGV back	304					2021/9/30 21:05		2021/9/30 21:06	
109	9211		2095		2 measure a part (analog)	115					2021/9/30 21:06		2021/9/30 21:06	

圖 14 自動化收集工單工時輸出欄位格式與資料摘錄

六、 MES4 資料庫（實際加工結果）甘特圖

在 FESTO 工廠中，所有以 MES 系統下單的工單都會將結果儲存至 Access 資料庫中，我們為了方便比較實際加工結果，另外撰寫了一個程式用來將資料庫中資料繪製成如同排程結果的甘特圖。此程式與原先僅能將排程結果繪製相較之下有了更多的功能，因此以終端機形式提供多種功能的執行指令，詳細介紹如下。

1. 資料庫讀取

啟動程式後會在根目錄尋找 tblFinStep.xlsx 檔案（可由工廠 Access 資料庫直接轉存）。讀取成功後會顯示如下圖，即可開始輸入指令操作

歡迎使用甘特圖繪製，輸入指令以繼續。或輸入 'help' 取得指令說明
資料庫讀取成功。

圖 15 資料庫讀取成功畫面

2. 指令提示

在終端機輸入「help」可以看到所有功能的簡單介紹。

```
>> help
指令資訊:
  輸入 'show'+工單編號(oNo)' 顯示該工單的甘特圖, 例: 'show 1816'。
  若甘特圖無法顯示中文, 可在指令最後加上 'en' 來顯示英文甘特圖, 例: 'show 1816 en'。
  輸入 'query *' 查詢資料庫內所有工單資訊。
  輸入 'refresh' 重新讀取資料庫檔案。
  輸入 'tw' 或 'en' 切換終端機顯示語言。
  輸入 'exit' 結束程式。
```

圖 16 指令提示

3. 甘特圖繪製

在終端機輸入「show 工單編號」顯示該工單的工時甘特圖，並於標題顯示工單編號。

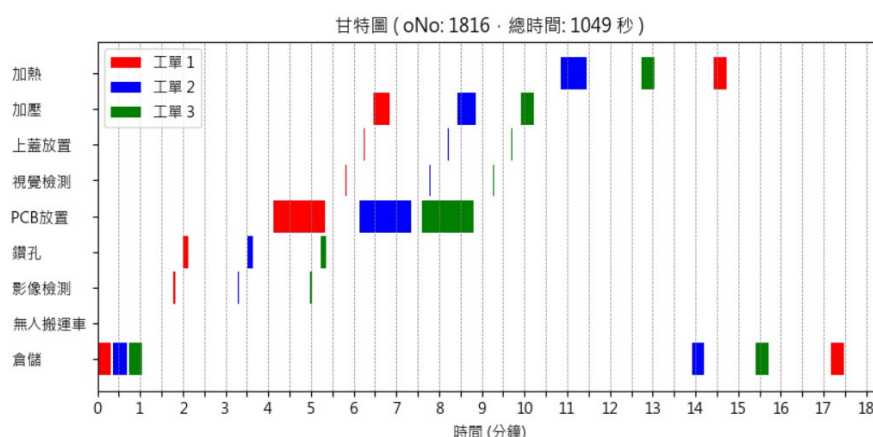


圖 17 資料庫歷史工單甘特圖

4. 工單編號查詢

若使用者不清楚工單編號，可以此功能快速查找，在此可以以類似 SQL 的指令搜尋，支援包括顯示項目、欄位條件、排序。亦可用「query help」指令查看查詢功能說明。

```
>> query help
搜尋詳細指令資訊:
  架構: query (1) where (2) order by (3)
  (1) 選擇顯示項目, 例: query WPNo and oNo
  (2) 欄位條件, 例: query * where WPNo = 1211
      若輸入時間, 格式須為: YYYY/mm/dd 或 YYYY/mm/dd-HH:MM
  (3) 排序條件, 例: query * order by WPNo asc
```

圖 18 工單編號查詢指令說明

>> query *				>> query ono and opos where opos > 1 order by opos desc			
WPNo	oNo	oPos	Start-Time	oNo	oPos		
1211	1816	3	2021/03/06 16:25	2023	10		
1211	1817	1	2021/03/12 14:52	2022	5		
1211	1818	1	2021/03/12 15:27	1816	3		
1211	1969	3	2021/08/13 23:47	1969	3		
1211	1975	2	2021/08/27 16:13	2035	3		
1211	1974	2	2021/08/27 15:43	2037	3		
1211	1973	2	2021/08/26 15:16	2038	3		
1211	1972	2	2021/08/26 14:57	2039	3		
1211	1971	1	2021/08/26 14:33	1975	2		
1211	1970	1	2021/08/26 14:07	1974	2		
9211	1994	1	2021/09/22 21:14	1973	2		
9211	1995	1	2021/09/23 09:36	1972	2		
9211	1997	1	2021/09/23 18:28				

圖 19 查詢結果範例

圖 20 查詢結果範例

5. 多語系

此程式終端機支援中英文雙語版本，可以輸入「en」切換成英文，「tw」切換成中文。

```
>> en
Welcome to gannt drawing, enter command to continue. Or enter 'help' for more infomation
>> help
Command Information:
Enter 'show'+ 'oNo' to show gannt of the work order number, e.g. 'show 1816'.
Enter 'query *' to query about all the orders information in the database.
Enter 'refresh' to reload the database file.
Enter 'tw' or 'en' to switch the language used in cosole.
Enter 'exit' to leave the program.
>> query help
Details of query commands:
Structure: query (1) where (2) order by (3)
(1) Rows to show, ex: query WPNo and oNo
(2) Filter for columns, ex: query * where WPNo = 1211
    For time format, it must be YYYY/mm/dd or YYYY/mm/dd-HH:MM
(3) Sort condition, ex: query * order by WPNo asc
```

圖 21 英文終端機模式

6. 資料庫刷新

輸入「refresh」可以更新資料庫內容，而不需重新啟動程式。

7. 結束程式

使用完成後，輸入「exit」即可結束程式。

肆、 結果討論

根據上述研究方法，我們對研究動機中的問題進行分析及改善，說明如下。

1. 產線輸送帶控制邏輯導致不必要時間浪費。

由於此工廠 MES 在規劃多筆工件加工時，為保持輸送帶順暢會將等待中的工件載具排入輸送帶，重複輸送行程。直到下一次輸送帶送至欲加工機台時，才會開始加工。

➤ 改善方法

使用我們撰寫的程式，利用 OPC UA 模組控制機台覆蓋原 MES 指令，改善輸送效率。

2. 工件進入加熱站時的站內溫度不同，使加熱時間難以預測。

➤ 改善方法

盡可能收集各種加熱起始溫度資料，然而因收集低溫資料較不易，導致低溫時預測時間較不準確。最終我們選擇在加熱站第一次加熱工件前，將機台預熱至 28 度，使得溫度落在深度學習模型較準確的預測範圍內。

3. 為收集不同溫度起始點之加熱數據，造成時間浪費。

➤ 改善方法

以自動化收集加熱機台資料程式解決此問題。

4. 數據雜亂無章，難以一眼分辨其代表之意義。

➤ 改善方法

以甘特圖繪製程式將數據圖像化，提高資訊可讀性。

伍、 結論

經由產線控制實現排程結果後，可以有效節省以下幾種時間浪費：

(1) AGV 因無法預測工單進入 AGV 等待區的所需時間而造成 AGV 多跑一趟

(2) 工單因 AGV 等待區無空間造成空跑圈

我們以六單的甘特圖為例，優化後的工時減少超過 14%。

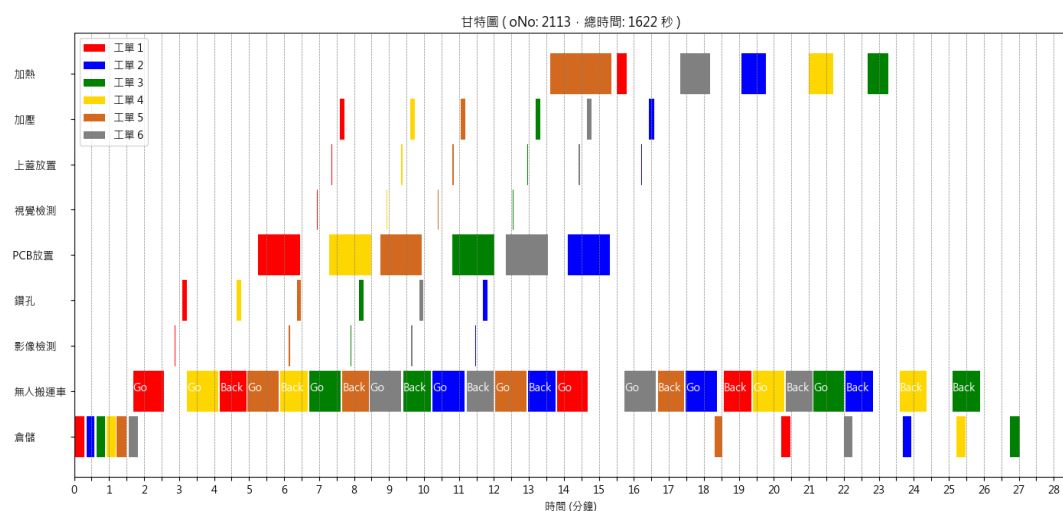


圖 22 未優化工時甘特圖，總工時：1622 秒

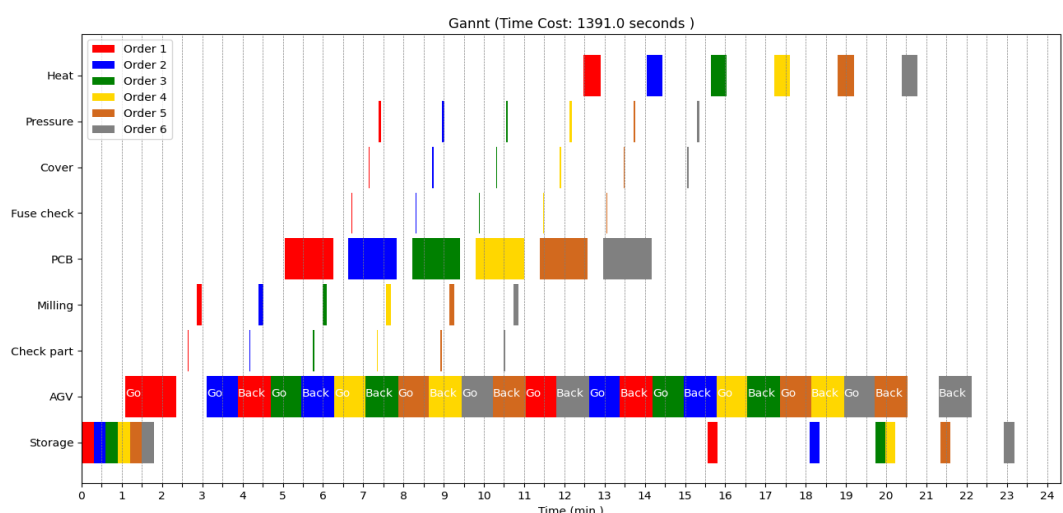


圖 23 優化後工時甘特圖，總工時：1391 秒

附錄一、程式碼

※以下附上本專題實作之程式碼連結及排程輸出甘特圖與文字檔。

內含檔案如下：

auto_heating_v3.0.py	加熱機台資料自動化收集程式
AutoCollectWO_v2.4.py	OPC UA 自動收集工時程式
DB_Gannt_V3.3.py	資料庫甘特圖繪製程式
DeeplearningModel.py	調用深度學習模組 Library
ordercontrol.py	OPC UA 流程控制程式
ORtool_DeepLearing.py	OR-Tool 排程結合深度學習程式
DL_LSTM.h5	LSTM 模型
tblFinStep.xlsx	自動收集工時成果 Excel
排程結果	內含 1 到 6 單甘特圖及文字紀錄排程結果

https://drive.google.com/drive/folders/1isceK_CDWWAuAf7W9D9FRLTEUS6iBRMR?usp=sharing



附錄圖 1 檔案連結 QR-Code