Mô hình hóa và điều khiển hệ thống sản xuất tự động bằng mạng Petri

Modelling and controlling automatic manufacturing system using Petri net

Phạm Trường Tùng[1,*], Phạm Đăng Phước[1], Lưu Đức Bình[2]
[1]*Trường Đại học Phạm Văn Đồng*
[2]*Trường Đại học Bách khoa, Đại học Đà Nẵng*
**Email:pttung@pdu.edu.vn*
Tel: +84-255.3822901; Mobile: 0935418001

Từ khóa:
Điều khiển; Mạng Petri; Mô hình hóa; PetriNet2MCU;

Tóm tắt

Bài báo trình bày phương pháp sử dụng mạng Petri để mô hình hóa và điều khiển hệ thống sản xuất tự động. Trên cơ sở mô hình mạng Petri, các phương pháp mô hình hóa cấu trúc điều khiển logic bằng mạng Petri được đề xuất. Phần mềm điều khiển hệ thống trên cơ sở lý thuyết mạng Petri được phát triển để mô hình hóa và điều khiển hệ thống. Các I/O của các thiết bị trên hệ thống được gắn liền với các vị trí, chuyển tiếp trên mạng Petri. Tiến trình hoạt động của hệ thống được phần mềm điều khiển dựa trên cơ sở lý thuyết về sự phát triển trạng thái của mạng Petri. Các kết quả nghiên cứu được thực nghiệm dựa trên hệ thống phân loại và lắp ráp sản xuất tại Trường Đại học Phạm Văn Đồng.

Keywords:
Controlling; Modelling; Petri net; PetriNet2MCU;

Abstract

This paper presents a method using Petri net to model and control the automated manufacturing system. The methods for modeling logical control structures using the Petri net are proposed based on PN model. Software control system based Petri net was developed to model and control the system. The devices I/Os are attached to the places or transition on the Petri net. The operation of the system is controlled by the software based on the theory of the development of the state of the Petri net. The experimental results are based on the classification and assembly system of Pham Van Dong University.

Ngày nhận bài: 01/07/2018
Ngày nhận bài sửa: 08/9/2018
Ngày chấp nhận đăng: 15/9/2018

1. GIỚI THIỆU

Hệ thống sản xuất tự động (Automated Manufacturing Systems - AMSs) là một khái niệm để chỉ việc một hệ thống sản xuất được điều khiển từ máy tính mà có thể thực hiện đồng thời nhiều công việc. Một hệ thống sản xuất tự động được tổ chức bởi các nhóm máy CNC, robot, các hệ thống vận chuyển được bố trí theo dây chuyền công nghệ. Hoạt động của hệ thống thường

được điều khiển bởi các bộ điều khiển logic. Ưu điểm của hệ thống sản xuất tự động là có thể đạt được năng suất sản xuất cao cho các công việc có quy mô loạt nhỏ. Để đảm bảo điều đó, hệ thống cần được thực hiện các nhiệm vụ lập kế hoạch, lập lịch, kiểm soát hệ thống và điều khiển.

Mạng Petri là một công cụ toán học và đồ hình rất hữu hiệu cho việc mô hình hóa, phân tích và thiết kế các hệ thống rời rạc, đặc biệt là các hệ thống sản xuất tự động. Các nghiên cứu ứng dụng của mạng Petri tập trung vào các hướng như lập lịch cho hệ thống sản xuất bằng cách sử dụng các giải thuật tìm kiếm, tối ưu [1-9]; Kiểm soát hệ thống, tránh hệ thống rơi vào các trạng thái chết (deadlock) [8, 10-13].

Mạng Petri còn được ứng dụng để mô phỏng kiểm soát, thiết kế điều khiển các hệ thống. A. A. Pouyan cùng các cộng sự trong [14] đã sử dụng phương pháp KST (Knitting Synthesis Technique) để đồng bộ mô hình mạng Petri cho hệ thống FMS để điều khiển hệ thống. M. Taleb cùng các cộng sự trong [15] đã sử dụng mô hình điều khiển dự báo (MPC - Model Predictive Control) để tìm vector điều khiển theo một tiêu chuẩn cho trước qua đó xác định chuỗi thông để điều khiển mạng T-TDPT (T - Timed Discrete Petri Net) từ trạng thái ban đầu đến trạng thái mô tả. L. Wang cùng các cộng sự trong [16] đã trình bày sử dụng ba phương pháp heuristic trên cơ sở chiến lược ON/OFF để cực tiểu hóa thời gian điều khiển của mạng Petri thời gian liên tục (Time Continuous Petri Net - TCPN). Luật điều khiển đã được áp dụng mô phỏng trên mô hình một FMS.

Tuy nhiên, các nghiên cứu trên đây chưa đề cập đến việc sử dụng mạng Petri để điều khiển trực tiếp các thiết bị trên hệ thống sản xuất tự động. Trong khi đó, với tính tương đồng giữa mạng Petri, hệ thống sản xuất tự động và phương pháp điều khiển logic thì việc áp dụng mạng Petri để điều khiển trực tiếp là hoàn toàn khả thi. Do đó, bài báo trình bày phương pháp mô hình hóa hệ thống sản xuất tự động bằng mạng Petri. Trên cơ sở mô hình mạng Petri, bằng cách gắn các cấu trúc I/O của các bộ điều khiển của các thiết bị trong hệ thống với các vị trí và chuyển tiếp của mạng Petri, ta đã tạo được một chương trình điều khiển hệ thống.

2. NỘI DUNG NGHIÊN CỨU

2.1. Mạng Petri thời gian

Mạng Petri được ứng dụng nhiều trong các lĩnh vực liên quan đến việc mô tả và phát triển các sự kiện. Lý thuyết này được công bố lần đầu tiên năm 1962 trong luận án tiến sĩ "Giao tiếp với các thiết bị tự động" của Carl Adam Petri tại Leipzig. Mạng Petri được định nghĩa như sau:

Định nghĩa 1: Mạng Petri gồm 4 phần tử:

$$N = < P, T, Pre, Post > \quad (1)$$

Trong đó:

- P là tập hữu hạn $n = |P|$ vị trí; T tập hữu hạn $m = |T|$ chuyển tiếp; $P \cap T = \varnothing$.
- Pre : $(P \times T) \rightarrow N+$ là hàm vào của mạng được định nghĩa bằng trọng số các cung từ vị trí vào chuyển tiếp.
- Post: $(P \times T) \rightarrow N+$ là hàm ra của mạng được định nghĩa bằng trọng số các cung từ chuyển tiếp ra vị trí.

Mạng Petri được định nghĩa trong định nghĩa 1 là một cấu trúc tĩnh. Để mô tả hệ thống, cần mô tả các trạng thái của hệ thống và sự thay đổi trạng thái của nó. Cấu trúc động của hệ thống được định nghĩa bằng các đánh dấu và quy tắc phát triển của nó.

Định nghĩa 2: Một đánh dấu M của một mạng N là một ánh xạ từ P vào N, thể hiện bằng một giá trị không âm (gọi là số token) tại mỗi vị trí.

Định nghĩa 3: Một mạng Petri đánh dấu là một cặp <N, M_0>, với N là mạng Petri và M_0 là đánh dấu ban đầu của mạng. Bắt đầu từ đánh dấu M_0, các token sẽ di chuyển qua lại các chuyển tiếp tuân theo hai quy tắc sau:

Quy tắc 1: Một chuyển tiếp được gọi là tích cực nếu tất các các vị trí vào của nó có số token lớn hơn hoặc bằng trọng số của cung nối giữa vị trí đó với chuyển tiếp.

$$M \geq \mathrm{Pre}(t) \tag{2}$$

Quy tắc 2: Quá trình thông của một chuyển tiếp tích cực xảy ra sẽ lấy (thêm) số token của tại vị trí vào (ra) bằng với trọng số của cung nối giữa vị trí (chuyển tiếp) đó và chuyển tiếp (vị trí). Khi đó:

$$M_k = M_{k-1} + \mathrm{Post}(t) - \mathrm{Pre}(t) = M_k + C(t) \tag{3}$$

- C(t) được gọi là ma trận tiến.

Gọi M_k là trạng thái đạt được từ trạng thái M_0 sau chuỗi các chuyển tiếp $\sigma = t_i t_j ...$ Khi đó:

$$M_k = M_0 + C.\sigma \qquad \sigma \geq 0 \tag{4}$$

Định nghĩa 4: Một mạng Petri thời gian – chuyển tiếp (A transition Timed Petri net - t-TPN) là một cặp <N,Z> với N = <P, T, Pre, Post> và Z là hàm với giá trị là một số thực không âm, z_i tại mỗi chuyển tiếp Z: T →R+; zi = Z(ti) được gọi là thời gian thông của chuyển tiếp t_i.

Quy tắc phát triển của mạng t-TPN gần giống như của PN. Nó chỉ khác là có quá trình thông mất khoảng thời gian z_i. Theo đó, quá trình thông của một chuyển tiếp gồm 3 pha.

1. Khi một chuyển tiếp t tích cực, quá trình thông bắt đầu.

2. Quá trình thông đếm ngược z_i thời gian.

3. Khi kết thúc thời gian z_i, quá trình thông kết thúc. Lúc này, các token sẽ được luân chuyển theo quy tắc 2 của định nghĩa 4.

2.2. Mô hình các cấu trúc điều khiển logic bằng mạng Petri thời gian chuyển tiếp

- Các tác động được gắn với vị trí: Tác động sẽ được thực hiện khi vị trí có token hiện diện. Như vậy, mỗi tác vụ hoạt động của hệ thống, ta xây dựng thành một vị trí tương ứng trên sơ đồ mạng Petri. Tác vụ sẽ thực hiện/không thực hiện tương ứng với trạng thái vị trí đó có/không có token.

- Chuyển tiếp có điều kiện: Một chuyển tiếp sẽ thông khi nó thõa mãn điều kiện ba pha ở định nghĩa 5 và thõa mãn điều kiện ngoài gắn vào nó.

- Cấu trúc AND: Với yêu cầu $P_n = (P_1 \text{ AND } P_2 \text{ AND} \ldots P_{n-1})$ thì các vị trí P_1 đến P_{n-1} sẽ là vị trí vào của một chuyển tiếp, P_n sẽ là vị trí ra của chuyển tiếp đó. Khi đó, nếu tất cả các P_{n-1} thõa mãn quy tắc 1 thì P_n sẽ nhận được token theo quy tắc 2. Mô hình được mô tả ở Hình 1a.

- Cấu trúc OR: Với yêu cầu $P_n = (P_1 \text{ OR } P_2 \text{ OR} \ldots P_{n-1})$, thì mỗi P_i (i=1…n-1) sẽ là đầu vào của một chuyển tiếp t_i. Khi đó, vị trí P_n là vị trí ra của tất cả các chuyển tiếp t_i. Khi đó, P_n sẽ nhận token theo quy tắc 2 nếu một trong số các P_i thõa mãn quy tắc 1. Mô hình được mô tả ở Hình 1b.

- Cấu trúc If-Else: Cho cấu trúc If – Else được mô tả bằng đoạn mã sau

```
If (T2=true) then
        P2
Else
        P3
```

Với cấu trúc này, yêu cầu điều kiện gắn vào chuyển tiếp T_3 và T_4 là ngược nhau (T_3 = NOT(T_4)). Khi đó mô hình mạng Petri cho cấu trúc này như Hình 1c.

- Cấu trúc While – do: cho cấu trúc While - do được mô tả bằng đoạn mã sau:

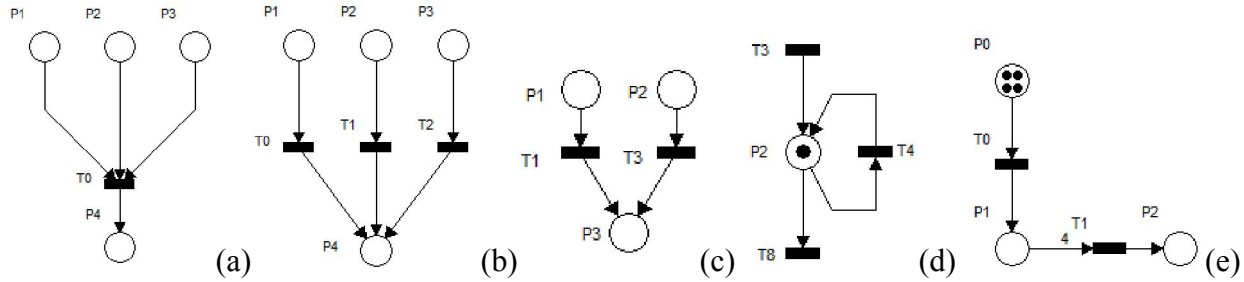
```
While (Condition T4=true)
{
 P1
}
```

Yêu cầu điều kiện gắn với chuyển tiếp T_4 và T_8 ngược nhau. Mô hình được mô tả ở Hình 1d.

- Cấu trúc FOR: Cho cấu trúc FOR được định nghĩa bởi đoạn mã sau:

```
For( i = 4; i≠0)
{
        P1
        i=i-1
}
```

Khi đó, mô hình được mô tả ở Hình 1e.



Hình 1. Mô tả các cấu trúc điều khiển logic bằng mạng Petri

2.3. Điều khiển hệ thống bằng mạng Petri

Như đã phân tích ở trên, hoạt động của mạng Petri tương đối giống với cơ chế điều khiển logic. Do đó, với các hệ thống hoạt động theo cơ chế điều khiển logic, ta có thể sử dụng mạng Petri để lập trình, điều khiển. Nguyên lý của điều khiển hệ thống bằng phần mềm là phần mềm sẽ giao tiếp với một bộ điều khiển, các tín hiệu vào/ra của hệ thống sẽ kết nối với bộ điều khiển. Do đó, thực chất là phần mềm sẽ điều khiển I/O của bộ điều khiển.

2.3.1. Phần mềm PetriNet2MCU

Phần mềm PetriNet2MCU được sử dụng để thiết kế, mô hình hóa việc điều khiển các thiết bị trong các hệ thống, phỏng trạng thái hoạt động của hệ thống cần lập trình; Kiểm soát hệ thống tránh cho hệ thống khi vận hành bị rơi vào các điểm chết (deadlock) hoặc rơi vào tình huống hoạt động mất kiểm soát (bùng nổ trạng thái). Để điều khiển các tác vụ của hệ thống, phần mềm cho phép gắn kết các vị trí, chuyển tiếp tương ứng với các I/O (input and output) của vi điều khiển. Bằng cách sử dụng các dạng cấu trúc AND, OR, NOT, IF…THEN,FOR…NEXT,… ta có thể thực hiện việc lập trình logic cho vi điều khiển, trên cơ sở đó có thể điều khiển các hệ thống mong muốn.

2.3.2. Bộ điều khiển

Bộ điều khiển được thiết kế sử dụng vi điều khiển PIC18F4431. PetriNet2MCU gởi các dữ liệu điều khiển đến phần mạch vi điều khiển. Mạch vi điều khiển là mạch trung gian nhận dữ liệu

điều khiển từ máy tính để điều khiển hệ thống trong thời gian thực. Mạch được thiết kế với đầy đủ các chân vào/ra của PIC. Ở các chân ra được thiết kế với mạch công suất để thực hiện khả năng điều khiển thiết bị cần dòng lớn.

2.3.3. Cấu trúc điều khiển

Hệ thống của ta sử dụng vi điều khiển PIC18F4431 (dạng DIP-40) để điều khiển. Để hỏi về trạng thái một chân nào đó trên vi điều khiển, ta phải định địa chỉ cho các chân đó. Mỗi gói tin dữ liệu gởi cho vi điều khiển có 8 bit, ta dành 6 bit cuối của gói tin cho việc định địa chỉ, 2 bit cuối ta sử dụng bit8 = 1 để báo cho vi điều khiển biết đây là gói tin hỏi về trạng thái input, bit7 ta không sử dụng (cho bằng 1).

Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
1	1	Address					

Đầu ra của hệ thống ở đây được hiểu là một chân của vi điều khiển được sử dụng là chân ra (output pin), chân này được nối với các thiết bị cần điều khiển (van điện từ, động cơ...). Trên mạng Petri, chân này được khai báo trong một vị trí nào đó của mạng mà mỗi khi vị trí này có token, chân này sẽ có mức tích cực cao. Như vậy, về mặt cấu trúc của vi điều khiển, ta cũng sẽ có 33 chân có khả năng làm chân ra. Và cũng để dễ dàng quản lý, ta định danh các chân này bằng một địa chỉ.

Cũng tương tự như trên, để biểu diễn hết các địa chỉ này, ta cần 6 bit. Do đó, ta sử dụng 6 bit thấp để định địa chỉ, bit8 = 0 để báo cho vi điều khiển biết đây là gói tin cập nhật trạng thái hệ thống, bit7 = {1,0} tùy theo trạng thái của vị trí (có token thì bit7 = 0, không có token thì bit7 = 1).

Như vậy cấu trúc gói tin như sau:

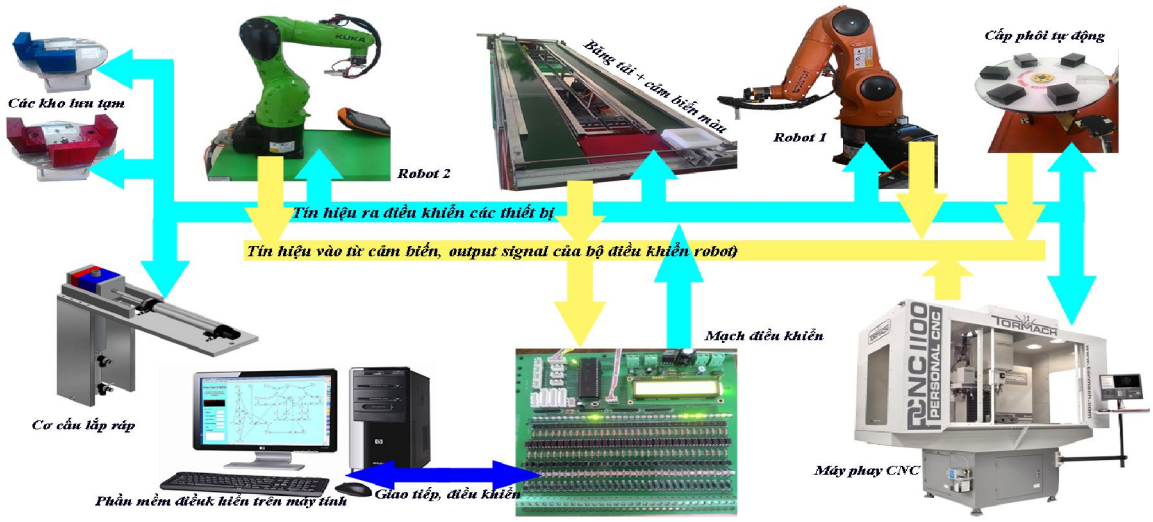
Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
0	0/1	address					

Gói dữ liệu này là câu trả lời của hệ thống cho gói dữ liệu hỏi về trạng thái đầu vào của hệ thống để làm điều kiện thông cho các chuyển tiếp. Gói tin từ hệ thống gởi đến phần mềm mạng Petri có cấu trúc như sau:

Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
1	0/1	address					

2.3.4. Kết quả thực nghiệm

Mô hình phân loại và lắp ráp sản phẩm được thiết kế, chế tạo và điều khiển trong [17]. Hệ thống làm nhiệm vụ gia công các chi tiết có màu sắc khác nhau trên máy phay CNC và tiến hành lắp ráp thành một sản phẩm. Hệ thống dùng vi điều khiển PIC18F4431 để làm bộ điều khiển trung tâm. Do đó khi muốn thay đổi chương trình điều khiển ta phải viết lại chương trình điều khiển cho vi điều khiển. Điều này dẫn đến mức độ linh hoạt của hệ thống thấp. Do vậy, để tăng mức độ linh hoạt của hệ thống, ta sử dụng phần mềm PetriNet2MCU để điều khiển hệ thống. PetriNet2MCU sẽ nhận các tín hiệu cảm biến từ các thiết bị ngoại vi truyền đến vi điều khiển và gởi tín hiệu điều khiển đến vi điều khiển. Do đó, ta kết nối các thiết bị của hệ thống như Hình 2.



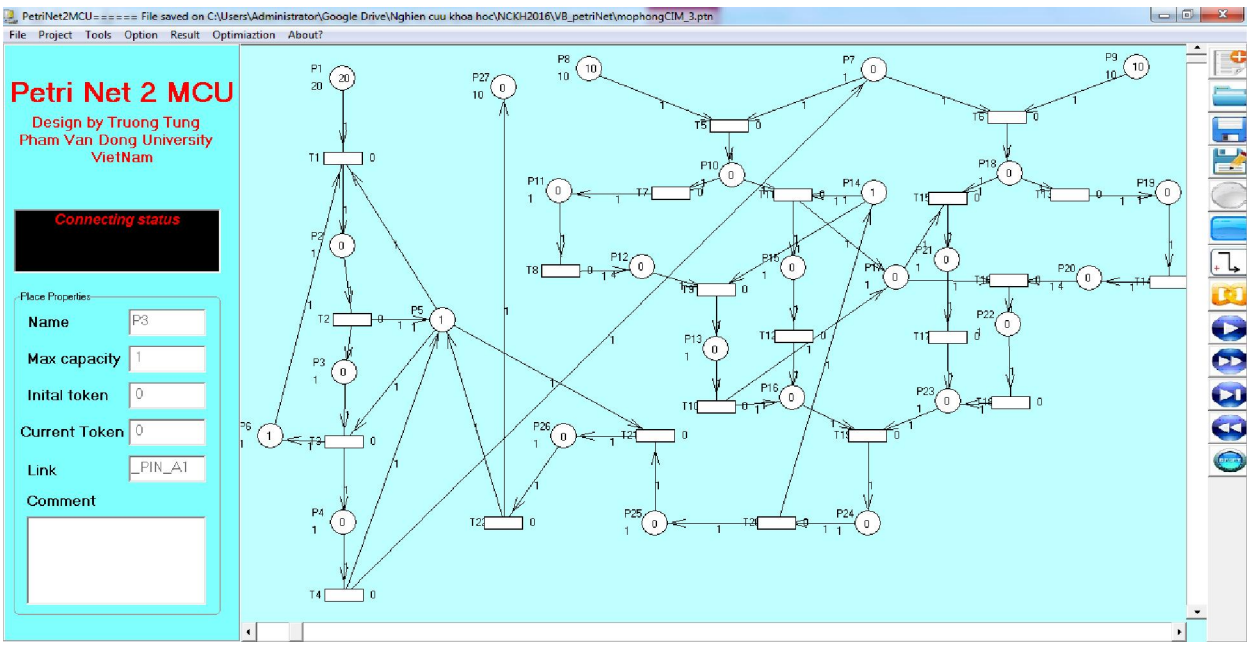
Hình 2. Sơ đồ kết nối hệ thống và PetriNet2MCU

Bảng 1. Bảng kết nối I/O

Cụm chức năng		Vi điều khiển PIC18F4431		Cụm chức năng		Vi điều khiển PIC18F4431	
Tên	**Thiết bị, cổng I/O của thiết bị**	**Chân tương ứng**	**Cấu hình I/O trên vi điều khiển**	**Tên**	**Thiết bị, cổng I/O của thiết bị**	**Chân tương ứng**	**Cấu hình I/O trên vi điều khiển**
Kho lưu tạm sản phẩm đỏ	Động cơ điều khiển	RB0	Output	Cấp phôi tự động	Động cơ	RD0	Output
Kho lưu tạm sản phẩm xanh	Động cơ điều khiển	RB1	Output		Cảm biến quang	RA3	Input
Băng tải 2 chiều	Động cơ chiều thuận	RB2	Output	Robot 1	$IN[10]	RA4	Output
	Động cơ chiều nghịch	RB3	Output		$IN[11]	RA5	Output
	Cảm biến màu TCS3200	RA0	Input		$OUT[5]	RE1	Input
Cơ cấu lắp ráp	Xy lanh 1	RB4	Output	Robot 2	$IN[10]	RD2	Output
	Xy lanh 2	RB5	Output		$IN[11]	RD3	Output
Máy phay CNC	OUT1	RB6	Input		$IN[12]	RD4	Output
	OUT2	RB7	Chưa dùng		$IN[13]	RD5	Output
	IN1	RA1	Output		$IN[14]	RE2	Output
	IN2	RA2	Chưa dùng		$IN[15]	RD1	Output
Mạch giao tiếp FT232	Tx	RC7	Giao tiếp RS232		$IN[16]	RE0	Output
	Rx	RC6			$OUT[6]	RD6	Input
					$OUT[7]	RD7	Input

2.3.5. Mô hình mạng Petri điều khiển hệ thống

Trên cơ sở yêu cầu hoạt động của hệ thống phải lắp ráp sản phẩm một cách nhanh nhất, ta xây dựng mô hình mạng Petri thời gian mô tả hệ thống như sau:



Hình 3. Mô hình mạng Petri hệ thống phân loại và lắp ráp sản phẩm

Các chức năng của các vị trí/chuyển tiếp gắn với các tác vụ được mô tả ở Bảng 2.

Bảng 2. Định nghĩa các I/O trên mạng Petri

Vi điều khiển PIC18F4431		Vị trí, chuyển tiếp tương tứng trên mạng Petri	Vi điều khiển PIC18F4431		Vị trí, chuyển tiếp tương tứng trên mạng Petri	Vi điều khiển PIC18F4431		Vị trí, chuyển tiếp tương tứng trên mạng Petri
Chân tương ứng	**Cấu hình I/O trên vi điều khiển**		**Chân tương ứng**	**Cấu hình I/O trên vi điều khiển**		**Chân tương ứng**	**Cấu hình I/O trên vi điều khiển**	
RB0	Output	P12	RA1	Output	P3	RD2	Output	P15
RB1	Output	P20	RA2	Chưa dùng		RD3	Output	P26
RB2	Output	P4	RC7	Giao tiếp RS232		RD4	Output	P13
RB3	Output	P27	RC6			RD5	Output	P11
RA0	Input	T4	RD0	Output	P1	RE2	Output	P21
RB4	Output	P24	RA3	Input	T1	RD1	Output	P22
RB5	Output	P25	RA4	Output	P2	RE0	Output	P19
RB6	Input	T3	RA5	Output		RD6	Input	T22
RB7	Chưa dùng		RE1	Input	T2	RD7	Input	

3. KẾT LUẬN

Bài báo đã đề xuất sử dụng mạng Petri để mô hình hóa và điều khiển hệ thống sản xuất tự động. Với đặc trưng của mạng Petri phù hợp với lập trình logic, bài báo đã đề nghị các mô hình mạng Petri cho các cấu trúc trong lập trình điều khiển logic. Phần mềm PetriNet2MCU đã được tác giả phát triển với các chức năng mô hình hóa, điều khiển hệ thống. Một mạch điều khiển sử dụng vi điều khiển PIC18F4431 làm trung gian được thiết kế để nhận/gởi các lệnh với PetriNet2MCU. Trên cơ sở lý thuyết về phát triển trạng thái của mạng Petri, các lệnh điều khiển

được gởi đến mạch điều khiển để điều khiển các thiết bị của hệ thống tương ứng với các trạng thái của mạng Petri.

Việc mô hình hóa và điều khiển hệ thống sản xuất là phương pháp điều khiển online, tức việc lập trình và điều khiển được thao tác trực tiếp từ máy tính, không cần trình biên dịch. Do đó, tính linh hoạt trong việc thay đổi chương trình điều khiển rất cao, đồng thời cũng có thể giám sát, theo dõi hoạt động của hệ thống thông qua việc phân tích các trạng thái của mạng Petri. Phương pháp đã được ứng dụng vào hệ thống sản xuất, phân loại và lắp ráp sản phẩm tại trường ĐH Phạm Văn Đồng.

Bằng việc sử dụng mô hình mạng Petri, quá trình lập kế hoạch, lập lịch của hệ thống bằng các giải thuật tối ưu có thể được sử dụng. Các phương pháp tối ưu hóa lập lịch hoạt động cho hệ thống bằng mạng Petri đã được tác giả nghiên cứu trong [18]. Giải thuật được trình bày trong [18] đang được tác giả nghiên cứu ứng dụng vào phần mềm PetriNet2MCU để có thể tối ưu hóa quá trình điều khiển hệ thống.

TÀI LIỆU THAM KHẢO

[1]. Lee Yin-Hsuan, Chang Chuei-Tin, et al.,2011. Petri-net based scheduling strategy for semiconductor manufacturing processes. *Chemical Engineering Research and Design*, 89(3): 291-300.

[2]. Wang Qiushuang and Wang Zhongdong,2012. Hybrid Heuristic Search Based on Petri Net for FMS Scheduling. *Energy Procedia*, 17, Part A(0): 506-512.

[3]. Huan Li, Wu Xiao, et al., 2012. Optimization of the scheduling and water integration in batch processes based on the Timed Petri net. *Computer Aided Chemical Engineering*, A.K. Iftekhar and S. Rajagopalan, Editors. Elsevier. 1447-1451.

[4]. Huang Bo, Jiang Rongxi, et al.,2014. Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions. *Computers & Industrial Engineering*, 71: 21-26.

[5]. Başak Özkan and Albayrak Y. Esra,2015. Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems. *Computers & Industrial Engineering*, 86: 116-126.

[6]. Petrović Milica, Vuković Najdan, et al.,2016. Integration of process planning and scheduling using chaotic particle swarm optimization algorithm. *Expert Systems with Applications*, 64(Supplement C): 569-588.

[7]. Al-Ahmari Abdulrahman,2016. Optimal robotic cell scheduling with controllers using mathematically based timed Petri nets. *Information Sciences*, 329: 638-648.

[8]. Mejía Gonzalo and Niño Karen,2017. A new Hybrid Filtered Beam Search algorithm for deadlock-free scheduling of flexible manufacturing systems using Petri Nets. *Computers & Industrial Engineering*, 108: 165-176.

[9]. Gang Xie and Quan Qiu,2016. Research on Planning Scheduling of Flexible Manufacturing System Based on Multi-level List Algorithm. *Procedia CIRP*, 56: 569-573.

[10]. Patel Ajay M. and Joshi Anand Y.,2013. Modeling and Analysis of a Manufacturing System with Deadlocks to Generate the Reachability Tree Using Petri Net System. *Procedia Engineering*, 64: 775-784.

[11]. Lefebvre Dimitri,2016. Deadlock-free scheduling for flexible manufacturing systems using untimed Petri nets and model predictive control. *IFAC-PapersOnLine*, 49(12): 384-389.

[12]. Chen YuFeng, Li ZhiWu, et al.,2017. Deadlock recovery for flexible manufacturing systems modeled with Petri nets. *Information Sciences*, 381: 290-303.

[13]. Lei Hang, Xing Keyi, et al.,2017. Hybrid heuristic search approach for deadlock-free scheduling of flexible manufacturing systems using Petri nets. *Applied Soft Computing*, 55: 413-423.

[14]. Pouyan Ali A., Shandiz Heydar Toossian, et al.,2011. Synthesis a Petri net based control model for a FMS cell. *Computers in Industry*, 62(5): 501-508.

[15]. Taleb M., Leclercq E., et al.,2014. Control Design of Timed Petri Nets via Model Predictive Control with Continuous Petri Nets. *IFAC Proceedings Volumes*, 47(2): 149-154.

[16]. Wang Liewei, Mahulea Cristian, et al.,2014. ON/OFF strategy based minimum-time control of continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 12(0): 50-65.

[17]. Nguyễn Ngọc Thiện, Phạm Trường Tùng, et al., 2017. Thiết kế chế tạo hệ thống phân loại và lắp ráp sản phẩm sử dụng robot Kuka. *Hội nghị - Triển lãm quốc tế lần thứ 4 về Điều khiển và Tự động hoá 2017*. TP Hồ Chí Minh, 321-326

[18]. Phạm Trường Tùng, Phạm Đăng Phước, et al., 2017. Sử dụng mạng Petri và thuật toán tìm kiếm heuristic để lập tiến trình sản xuất cho hệ thống sản xuất linh hoạt. *Hội nghị tự động hóa toàn quốc - VCCA - 2017*. TP Hồ Chí Minh, 294-299