*Tạp chí Tin học và Điều khiển học, T.21, S.1 (2005), 84–90*

# BÀI TOÁN ĐIỀU KHIỂN TRÊN CÁC HỆ ĐIỀU KIỆN - BIẾN CỐ CÓ TÍNH THỜI GIAN

HOÀNG CHÍ THÀNH[1], PHẠM XUÂN ĐỒNG[2]

[1] *Trường Đại học Khoa học tự nhiên Hà Nội*
[2] *Trường Đại học Thủy lợi Hà Nội*

**Abstract.** In this paper, we represent and solve the Control problem on Timed Condition/Event systems by proposing a concurent composition. We also show that some important characteristics of the system such as the safeness, the cyclicness and the aliveness of Timed Condition/Event Systems are predetermined and preserved by the controller.

**Tóm tắt.** Bài báo đề xuất một phép hợp thành tương tranh để giải bài toán điều khiển trên các hệ điều kiện - biến cố có tính thời gian. Đồng thời cũng chỉ ra rằng các tính chất quan trọng của hệ thống như: an toàn, chu trình và sống là các tính chất định trước, được bảo toàn qua phép điều khiển này.

## 1. MỞ ĐẦU

Nghiên cứu, phân tích, thiết kế, điều khiển các hệ thống tương tranh và ứng dụng của chúng đang là vấn đề được nhiều người quan tâm. Cơ sở chính cho các bài toán phân tích, thiết kế và điều khiển là phép hợp thành tương tranh. Bài toán điều khiển trên các hệ tương tranh đã được L. Alfaro, T. A. Henzinger và F. Y. C. Mang hình thức hóa trong CONCUR' 2000 [2] như sau:

Với một hệ tương tranh $\mathcal{M}$ đã cho, hãy tìm hệ tương tranh $\mathcal{N}$ sao cho hệ hợp thành $\mathcal{M}\|\mathcal{N}$ thỏa mãn các tính chất định trước. Hệ tương tranh cần tìm $\mathcal{N}$ (nếu có) được gọi là hệ điều khiển (controller). Hiển nhiên, nghiệm của bài toán điều khiển phụ thuộc rất nhiều vào mô hình biểu diễn của hệ tương tranh và phép toán hợp thành.

Bài này trình bày việc giải quyết bài toán điều khiển trên các hệ điều kiện - biến cố có tính thời gian, một trong các mô hình phổ dụng thường được sử dụng để mô hình hóa các hệ phân bố thời gian thực, đồng thời cũng chỉ ra rằng các tính chất an toàn, chu trình và sống của các hệ điều kiện - biến cố có tính thời gian được bảo toàn qua phép điều khiển.

## 2. HỆ ĐIỀU KIỆN - BIẾN CỐ CÓ TÍNH THỜI GIAN

Hệ điều kiện - biến cố có tính thời gian được xây dựng dựa trên khái niệm mạng Petri, được định nghĩa trong [3] như sau:

**Định nghĩa 1.** Bộ ba $N = (B, E, F)$ được gọi là một *mạng Petri* nếu:
- $B, E$ là hai tập hợp không rỗng rời nhau,
- $F \subseteq (B \times E) \cup (E \times B)$ là một quan hệ nhị nguyên và được gọi là quan hệ *luồng* của $N$.

Người ta thường dùng đồ thị hai phía định hướng để biểu diễn hình học cho mạng $N$ với tập đỉnh là tập các phần tử của mạng $X_N = B \cup E$, còn các cung thể hiện quan hệ luồng $F$.

Giả sử $N = (B, E, F)$ là một mạng. Với mỗi phần tử của mạng $x \in X_N$, ta kí hiệu:
$^\bullet x = \{y \,|\, yFx\}$ và được gọi là tập trước (preset) của $x$,
$x^\bullet = \{y \,|\, xFy\}$ và gọi là tập sau (postset) của $x$.

Mạng $N$ được gọi là *đơn giản* nếu hai phần tử khác nhau không có chung tập trước và tập sau. Mỗi tập con $c \subseteq B$ được gọi là một *trường hợp*.

Giả sử $e \in E$ và $c \subseteq B$. Phần tử $e$ được trường hợp $c$ *kích hoạt* nếu $^\bullet e \subseteq c \wedge e^\bullet \cap c = \varnothing$. Khi đó, $c' = (c \setminus^\bullet e) \cup e^\bullet$ được gọi là trường hợp kế tiếp của $c$ do sự thực hiện của e và ta viết: $c[e > c'$.

Tập con $G \subseteq E$ được gọi là *tách được* nếu

$$\forall\, e_1, e_2 \in G : e_1 \neq e_2 \;\Rightarrow\; {}^\bullet e_1 \cap {}^\bullet e_2 = e_1^\bullet \cap e_2^\bullet = \varnothing.$$

Giả sử $c$, $c'$ là các trường hợp và $G$ là tập tách được.

**Định nghĩa 2.** Tập $G$ được gọi là *một bước* trên mạng $N$ từ $c$ tới $c'$, nếu mỗi phần tử $e \in G$ là c kích hoạt và $c' = (c \setminus {}^\bullet G) \cup G^\bullet$.

Khi đó ta ký hiệu: $c[G > c'$ và các biến cố trong bước $G$ có thể được thực hiện một cách tương tranh.

Các bước trên mạng $N$ tạo nên quan hệ đạt được một bước $r_N \subseteq \mathcal{D}(B) \times \mathcal{D}(B)$ được định nghĩa như sau:

$$\forall\, c, c' \in \mathcal{D}(B) : c\, r_N\, c' \Leftrightarrow \exists G \subseteq E,\; c[G > c'.$$

Bao đóng của quan hệ đạt được tiến và lùi, $R_N = (r_N \cup r_N^{-1})^*$ cho ta quan hệ đạt được trên mạng $N$. Đây là một quan hệ tương đương trên $\mathcal{D}(B)$.

**Định nghĩa 3.** *Hệ điều kiện - biến cố có tính thời gian* là một bộ năm $\mathcal{N} = (B, E, F, c_0, D)$, trong đó:

- $N = (B, E, F)$ là một *mạng đơn giản*, với các phần tử của tập $B$ biểu diễn các *điều kiện*, còn các phần tử của tập $E$ biểu diễn các *biến cố* của hệ,.
- $c_0 \subseteq B$ là *trường hợp đầu tiên* (hay còn gọi là trạng thái đầu tiên),
- $D : E \to [0, +\infty)$ là hàm khoảng biểu diễn *thời gian xảy ra* của các biến cố.

Lớp tương đương $\mathcal{C} = [c_0]_{RN}$ được gọi là không gian các trạng thái của hệ $\mathcal{N}$ (xem [3, 4]).

Một biến cố của hệ mạng có thể xảy ra nếu trong hệ có trạng thái làm thỏa mãn các điều kiện trước (pre-conditions) của biến cố đó và khi ấy các điều kiện sau (post-conditions) của biến cố này chưa thỏa mãn. Khi biến cố xảy ra, các điều kiện trước không thỏa mãn nữa và các điều kiện sau được thỏa mãn. Trạng thái kế tiếp nhận được sau khi biến cố trên xảy ra phải thuộc vào không gian các trạng thái để có thể kích hoạt các biến cố khác. Không gian các trạng thái của hệ là môi trường để dãy các bước có thể xảy ra trên hệ, tạo nên các quá trình trên hệ.

Hệ điều kiện - biến cố có tính thời gian thường dùng để mô hình các hệ thống tương tranh, chẳng hạn như các hệ thống truyền tin, các hệ điều hành, các dây chuyền sản xuất, các chương trình tương tranh....

Giả sử $\mathcal{N} = (B, E, F, c_0, D)$ là một hệ điều kiện - biến cố có tính thời gian và trên hệ này có một dãy các trạng thái $c_1, c_2, ..., c_n, c_{n+1}$ và một dãy các bước: $G_1, G_2, ..., G_n$ mà:

$c_i[G_i > c_{i+1},\ i = 1, 2, ..., n.$

Khi đó, dãy: $c_1[G_1 > c_2[G_2 > c_3...c_n[G_n > c_{n+1}$ thể hiện một quá trình xảy ra trên hệ.

Trong nhiều trường hợp, ta thường quan tâm đến các dãy các bước $G_1, G_2, ..., G_n$ mà từ đó sinh ra ngôn ngữ các dãy cháy được (firing sequences language) hay còn gọi là ngôn ngữ sinh bởi hệ.

Nếu các biến cố của hệ chỉ có thể thực hiện một cách tuần tự thì thời gian thực hiện xong quá trình trên sẽ là

$$\sum_{i=1}^{n} \sum_{e \in G_i} D(e).$$

Còn nếu các biến cố của hệ có thể thực hiện một cách đồng thời thì thời gian thực hiện xong quá trình trên sẽ giảm xuống còn

$$\sum_{i=1}^{n} \max_{e \in G_i} D(e).$$

Để giảm thời gian thực hiện một quá trình, ta có thể làm giảm số bước của quá trình này bằng cách sử dụng thuật toán "dồn trái" như sau.

**Thuật toán 1.** (Thuật toán "dồn trái")

1. Nếu $e \in G_{i+1}$, $e \notin G_i$, $c_i[e > c_{i+1}$ và tập $G_i \cup \{e\}$ vẫn là tách được thì chuyển biến cố $e$ từ bước $G_{i+1}$ sang bước $G_i$, nghĩa là:
$$G_i := G_i \cup \{e\} \ \text{ và } \ G_{i+1} := G_{i+1} \setminus \{e\}.$$
2. Lặp lại bước 1 với mọi biến cố thỏa mãn điều kiện trên.

Khi thuật toán kết thúc, ta sẽ nhận được các bước tương tranh cực đại. Do vậy, số các bước của quá trình sẽ là ít nhất và thời gian thực hiện quá trình sẽ giảm đáng kể.

Dưới đây, chúng ta nhắc lại một số tính chất quan trọng của các hệ điều kiện - biến cố có tính thời gian.

**Định nghĩa 4.** Hệ điều kiện - biến cố có tính thời gian $\mathcal{N} = (B, E, F, c_0, D)$ được gọi là *an toàn* nếu với mỗi biến cố $e \in E$ và với mỗi trường hợp $c \in [c_0]_R$:

$$^{\bullet}e \subseteq c \Rightarrow e^{\bullet} \subseteq B \setminus c \quad \text{và} \quad e^{\bullet} \subseteq c \Rightarrow {}^{\bullet}e \subseteq B \setminus c.$$

Tính an toàn của một hệ thống đảm bảo rằng khi các điều kiện trước hay các điều kiện sau của một biến cố thỏa mãn trong một trạng thái nào đó thì trạng thái ấy có thể kích hoạt biến cố này. Đây là một tính chất thiết yếu cho các hệ thống thực.

W. Reisig [3] đã chỉ ra rằng: từ một hệ điều kiện - biến cố tùy ý chúng ta luôn có thể làm đầy đủ để nhận được một hệ điều kiện - biến cố an toàn tương đương ngôn ngữ với nó. Do đó, đối với một số bài toán hay trên các hệ điều kiện - biến cố tùy ý ta chỉ cần xét chúng trên các hệ điều kiện - biến cố an toàn tương đương.

Dựa vào lý thuyết đồ thị, tính chu trình và tính sống của hệ điều kiện - biến cố có tính thời gian được định nghĩa như sau.

**Định nghĩa 5.** Hệ điều kiện - biến cố có tính thời gian $\mathcal{N} = (B, E, F, c_0, D)$ được gọi là *chu trình* nếu: $\forall c_1, c_2 \in \mathcal{C} : c_1 \, r_N^* \, c_2$.

Hệ điều kiện - biến cố có tính thời gian là chu trình khi và chỉ khi đồ thị các trường hợp của nó là liên thông mạnh (xem [3]). Tính chu trình tạo nên các quá trình vô hạn.

**Định nghĩa 6.** Hệ điều kiện - biến cố có tính thời gian $\mathcal{N} = (B, E, F, c_0, D)$ được gọi là *sống* nếu: $\forall c \in \mathcal{C}, \forall e \in E, \exists c' \in \mathcal{C}$ sao cho $c\, r_N^*\, c'$ và biến cố $e$ là $c'$ kích hoạt được.

Tính sống đảm bảo rằng mỗi biến cố của hệ luôn được kích hoạt bởi một trạng thái nào đó và hệ sẽ không có biến cố vô nghĩa.

Tính chu trình và tính sống của hệ điều kiện - biến cố có tính thời gian phụ thuộc vào quan hệ đạt tới tiến. Hiển nhiên tính chu trình kéo theo tính sống, nhưng điều ngược lại chưa chắc đúng.

## 3. PHÉP HỢP THÀNH TƯƠNG TRANH TRÊN CÁC HỆ ĐIỀU KIỆN - BIẾN CỐ CÓ TÍNH THỜI GIAN

Giả sử $\mathcal{M} = (B^M, E^M, F^M, c_0^M, D^M)$ và $\mathcal{N} = (B^N, E^N, F^N, c_0^N, D^N)$ là hai hệ điều kiện - biến cố có tính thời gian. Chúng tôi đề xuất một phép hợp thành tương tranh trên hai hệ này qua định nghĩa dưới đây.

**Định nghĩa 7.** Hệ điều kiện - biến cố có tính thời gian $\mathcal{S} = (B, E, F, c_0, D)$, trong đó:

1. $B = B^M \cup B^N$,
2. $E = E^M \cup E^N$,
3. $F = F^M \cup F^N$,
4. $c_0 = c_0^M \cap (B^M \setminus B^N) \cup c_0^M \cap c_0^N \cup c_0^N \cap (B^N \setminus B^M)$,
5. hàm khoảng $D$ được xác định như sau:

$$D(e) = \begin{cases} D^M(e) & \text{nếu} \quad e \in B^M \setminus B^N \\ \min(D^M(e), D^N(e)) & \text{nếu} \quad e \in B^M \cap B^N \\ D^N(e) & \text{nếu} \quad e \in B^N \setminus B^M \end{cases}$$

được gọi là *hợp thành tương tranh* của hai hệ $\mathcal{M}$ và $\mathcal{N}$, và ta ký hiệu:

$$\mathcal{S} = M \| \mathcal{N}.$$

Chú ý rằng, trạng thái đầu tiên $c_0$ của hệ hợp thành được xây dựng từ trạng thái đầu tiên của các hệ thành phần theo cách sau đây:

Trên các phần riêng $B^M \setminus B^N$ và $B^N \setminus B^M$, các điều kiện riêng được chọn hết, đó là: $c_0^M \cap (B^M \setminus B^N)$ và $c_0^N \cap (B^N \setminus B^M)$; còn trên phần chung $B^M \cap B^N$ thì chỉ chọn các điều kiện chung $c_0^M \cap c_0^N$. Đây chính là một thể hiện của nguyên lý hợp thành tổng quát mà các tác giả đã đề xuất trong [5].

Như vậy, bài toán điều khiển trên các hệ điều kiện - biến cố có tính thời gian là giải được. Hay nói một cách khác, với một hệ điều kiện - biến cố có tính thời gian cho trước, luôn luôn có thể chọn một hệ điều kiện - biến cố có tính thời gian khác mà hợp thành của chúng lại là một hệ điều kiện - biến cố có tính thời gian. Phép hợp thành trình bày ở trên không những là lời giải cho bài toán điều khiển mà còn là một giải pháp tốt cho việc thiết kế hệ thống theo phương pháp dưới-lên. Hơn nữa, chúng ta xét những tính chất nào của hệ hợp thành được bảo toàn sau khi điều khiển. Các định lý dưới đây sẽ chỉ ra điều đó.

**Định lý 1.** *Nếu $\mathcal{M}$ và $\mathcal{N}$ là các hệ điều kiện - biến cố an toàn có tính thời gian thì hệ điều kiện - biến cố có tính thời gian hợp thành $\mathcal{S}$ cũng an toàn và không gian trạng thái của nó của có cấu trúc như sau:*

$$[c_0]_R = \{c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N \cup c^N \cap (B^N \setminus B^M) \mid c^M \in [c_0^M]_{R_M} \wedge c^N \in [c_0^N]_{R_N}\}.$$

*Chứng minh.* Ta ký hiệu:

$$C = \{c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N \cup c^N \cap (B^N \setminus B^M) \mid c^M \in [c_0^M]_{RM} \wedge c^N \in [c_0^N]_{RN}\}.$$

Chúng ta phải chứng minh rằng họ các trường hợp $C$ là đóng đối với quan hệ đạt tới $R_S$ của hệ điều kiện - biến cố có tính thời gian hợp thành $\mathcal{S}$.

Thật vây, giả sử $c \in C$ và $d \subseteq B$.

1) Nếu $(c, d) \in r_S$ thì: $\exists e \in E$, ${}^\bullet e \subseteq c \wedge e^\bullet \cap c = \emptyset \wedge d = (c \setminus {}^\bullet e \cup e^\bullet$.

Vì $c \in C$ nên $\exists c^M \in [c_0^M]_{R_M}$ và $c^N \in [c_0^N]_{R_N}$ sao cho:

$$c = c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N \cup c^N \cap (B^N \setminus B^M).$$

Đặt: $c' = c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N$ và $c'' = c^M \cap c^N \cup c^N \cap (B^N \setminus B^M)$.

Thế thì $c = c' \cup c''$. Do ${}^\bullet e \subseteq c$ nên: ${}^\bullet e \cap B_M \subseteq c'$ và ${}^\bullet e \cap B_N \subseteq c''$.

Nhưng ${}^\bullet e \cap B_M$ và ${}^\bullet e \cap B_N$ chính là ${}^\bullet e$ trong mạng $\mathcal{M}$ và $\mathcal{N}$ tương ứng.

Hơn nữa, do các hệ $\mathcal{M}$ và $\mathcal{N}$ là an toàn nên ta có: $d' = (c' \setminus {}^\bullet e \cap B_M) \cup e^\bullet \cap B_M \in [c_0^M]_{R_M}$ và $d'' = (c'' \setminus {}^\bullet e \cap B_N) \cup e^\bullet \cap B_N \in [c_0^N]_{R_N}$.

Vậy: $d = d' \cap d$". Theo cách xây dựng họ $C$, ta có: $d \in C$.

Trong trường hợp ngược lại, nếu $(d, c) \in r_S$ thì cách chứng minh là hoàn toàn tương tự.

Bây giờ chúng ta chứng minh tính an toàn của hệ hợp thành $\mathcal{S}$.

Giả sử $e \in E$ và $c \in C$ mà ${}^\bullet e \subseteq c$. Thế thì: $\exists c^M \in [c_0^M]_{R_M}$, $c^N \in [c_0^N]_{R_N}$ sao cho:

$$c = c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N \cup c^N \cup (B^N \setminus B^M).$$

Tương tự như trên, ta đặt:

$$c' = c^M \cap (B^M \setminus B^N) \cup c^M \cap c^N \quad \text{và} \quad c'' = c^M \cap c^N \cup c^N \cap (B^N \setminus B^M).$$

Vì ${}^\bullet e \subseteq c$ nên ${}^\bullet e \cap B_M \subseteq c'$ và ${}^\bullet e \cap B_N \subseteq c''$. Mà ${}^\bullet e \cap B_M$ và ${}^\bullet e \cap B_N$ chính là ${}^\bullet e$ trong mạng $\mathcal{M}$ và $\mathcal{N}$ tương ứng và do tính an toàn của $\mathcal{M}$ và $\mathcal{N}$ nên ta có:

$$e^\bullet \cap BM \cap c' = \emptyset \ \text{ và } e^\bullet \cap BN \cap c" = \varnothing.$$

Nhưng $e^\bullet \cap B_M$ và $e^\bullet \cap B_N$ chính là $e^\bullet$ trong $\mathcal{M}$ và $\mathcal{N}$. Vậy $e^\bullet \cap (c' \cup c'') = \varnothing$, nghĩa là $e^\bullet \cap c = \emptyset$ hay $e^\bullet \subseteq B \setminus c$.

Trường hợp $e^\bullet \subseteq c$, chứng minh hoàn toàn tương tự.

Định lý được chứng minh đầy đủ. ■

Định lý 1 không những chỉ ra rằng tính an toàn được bảo toàn qua phép điều khiển mà còn cho chúng ta một phương pháp hữu hiệu để xây dựng đầy đủ không gian các trạng thái của hệ hợp thành.

Chú ý rằng, trong trường hợp tổng quát, nghĩa là khi các hệ $\mathcal{M}$ và $\mathcal{N}$ là không an toàn thì không gian các trạng thái $[c_0]_R$ của hệ hợp thành và họ $C$ các trường hợp như đã ký hiệu trong chứng minh của định lý trên có thể khác nhau. Thậm chí họ $C$ không đóng đối với quan hệ đạt tới của hệ hợp thành. Chúng tôi có một phản ví dụ tương đối phức tạp cho tình huống này.

**Định lý 2.** *Tính chu trình và tính sống được bảo toàn qua phép điều khiển trên các hệ điều kiện - biến cố an toàn có tính thời gian.*

*Chứng minh.* Hoàn toàn tương tự như chứng minh của Định lý 1.

Từ Định lý 1 và Định lý 2 chúng ta thấy rằng một hệ điều kiện - biến cố an toàn (chu trình, sống) có tính thời gian luôn có thể điều khiển một hệ điều kiện - biến cố an toàn (chu trình, sống) có tính thời gian khác để nhận được hệ điều kiện - biến cố an toàn (chu trình, sống) có tính thời gian hợp thành. Vậy các tính chất định trước trong bài toán điều khiển trên các hệ điều kiện - biến cố có tính thời gian chính là tính an toàn, chu trình và sống.

## 4. KẾT LUẬN

Các kết quả đã trình bày có thể áp dụng được cho bài toán điều khiển trên một số mô hình biểu diễn các hệ tương tranh khác như: mạng các vị trí-chuyển, hệ thống kiểm tra, các chương trình tương tranh... Hơn nữa, việc xây dựng các bước tương tranh của hệ hợp thành trực tiếp từ các bước tương tranh của các hệ thành phần là việc làm rất cần thiết. Nó góp phần làm cho quá trình điều khiển thực hiện trên các hệ thống hợp thành trở nên nhanh chóng và đơn giản hơn. Các kết quả nghiên cứu này sẽ phát triển lý thuyết đại số các quá trình và có thể áp dụng trong thiết kế hệ thống.

## TÀI LIỆU THAM KHẢO

[1] J. I. Albersberg and G. Rozenberg, Theory of Traces, *Theoretical Computer Science* **60** (1988) 1–82.

[2] L. Alfaro, T. A. Henzinger, and F. Y. C. Mang, *The Control of Synchronous Systems*, LNCS 1877, Springer, 2000, 458–473.

[3] W. Reisig, *Petri Nets: An Introduction*, Springer-Verlag, 1985.

[4] Hoang Chi Thanh, *Behavioural Synchronization of Net Systems*, IMSc Report 115, Madras, India, 1991, 136–145.

[5] Hoàng Chí Thành, Bài toán đồng bộ đầy đủ trong lý thuyết tương tranh, *Kỷ yếu Hội thảo Quốc gia lần thứ IV "Một số vấn đề chọn lọc của Công nghệ thông tin"*, NXB Khoa học Kỹ thuật, 2002, 242–244.

[6] J. Winkowski, *A representation of processes of Timed Petri Nets*, ICS PAS Report 728, Warsaw, Poland, 1993.

*Nhận bài ngày 10 - 4 - 2004*
*Nhận lại sau sửa ngày 27 - 12 - 2004*