A Control Method for Timed Distributed Continuous Petri nets

Hanife Apaydın Özkan, Jorge Júlvez, Cristian Mahulea, Manuel Silva

Universidad de Zaragoza
Instituto de Investigación en Ingeniería de Aragon (I3A), Spain

December 9, 2009
A Control Method for Timed Distributed Continuous Petri nets

Outline

Introduction

Distributed Continuous Petri nets
- Continuous Petri nets (contPN)
- Distributed Continuous Petri nets (DcontPN)

Control Strategy
- Control Actions
- Problem Statement

Controller for DcontPN
- Design of a Distributed Controller
- Reachability of Target Marking

Case Study

Conclusion
Outline

Introduction

Distributed Continuous Petri nets
- Continuous Petri nets (contPN)
- Distributed Continuous Petri nets (DcontPN)

Control Strategy
- Control Actions
- Problem Statement

Controller for DcontPN
- Design of a Distributed Controller
- Reachability of Target Marking

Case Study

Conclusion
A control problem for distributed contPNs (which is a set of different contPNs communicating between them) is considered.

The application of the obtained control inputs drives the subsystems from the initial states to the target states in a finite amount of time.

An algorithm is developed to calculate the control inputs for each subsystem.
A Control Method for Timed Distributed Continuous Petri nets

Objective

Outline

1. Introduction
2. Distributed Continuous Petri nets
   - Continuous Petri nets (contPN)
   - Distributed Continuous Petri nets (DcontPN)
3. Control Strategy
   - Control Actions
   - Problem Statement
4. Controller for DcontPN
   - Design of a Distributed Controller
   - Reachability of Target Marking
5. Case Study
6. Conclusion
A Control Method for Timed Distributed Continuous Petri nets

Continuous Petri nets (contPN)

Definition (contPN)

A continuous Petri net (contPN) \( \mathcal{N} \) is a tuple \( \mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle \) where:

- \( P \) and \( T \) are the sets of places and transitions respectively.
- \( \text{Pre}, \text{Post} \in \mathbb{R}_{\geq 0}^{\mid P \mid \times \mid T \mid} \) are the pre and post incidence matrices.
- \( \lambda \in \mathbb{R}_{>0}^{\mid T \mid} \) is the firing rate of transition.

Mostly used two server semantics for contPN

- Finite Server Semantics
- Infinite Server Semantics
Continuous Petri nets (contPN)

Definition (contPN)

A continuous Petri net (contPN) \( N \) is a tuple \( N = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle \) where:

- \( P \) and \( T \) are the sets of places and transitions respectively.
- \( \text{Pre}, \text{Post} \in \mathbb{R}_{\geq 0}^{|P| \times |T|} \) are the pre and post incidence matrices.
- \( \lambda \in \mathbb{R}_{>0}^{|T|} \) is the firing rate of transition.

The flow of a transition \( t_j \) at time \( \tau \):

\[
f_j(\tau) = \lambda_j \cdot \text{enab}(t_j, m(\tau)) = \lambda_j \cdot \min_{p_i \in \bullet t_j} \left\{ \frac{m_i(\tau)}{\text{Pre}_{ij}} \right\}
\]
Definition (contPN)

A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle$ where:

- $P$ and $T$ are the sets of places and transitions respectively.
- $\text{Pre}, \text{Post} \in \mathbb{R}^{|P| \times |T|}$ are the pre and post incidence matrices.
- $\lambda \in \mathbb{R}_{>0}$ is the firing rate of transition.

$$m = [m_1 \ m_2 \ m_3]^T$$
$$\lambda = [\lambda_1 \ \lambda_2]^T$$
$$f_1 = \lambda_1 \cdot \min\{m_1, m_2\}$$
$$f_2 = \lambda_2 \cdot m_3$$
Definition (contPN)

A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle$ where:

- $P$ and $T$ are the sets of places and transitions respectively.
- $\text{Pre}, \text{Post} \in \mathbb{R}^{P \times |T|}_{\geq 0}$ are the pre and post incidence matrices.
- $\lambda \in \mathbb{R}^{|T|}_{> 0}$ is the firing rate of transition.

\[
\begin{align*}
  p_1 & \quad p_2 \\
  0.5 & \quad 1.5 \\
  t_1 & \quad \lambda_1 \\
  t_2 & \quad \lambda_2 \\
  p_3 & \\
\end{align*}
\]

\[
\mathbf{m} = [0.5 \ 1.5 \ 0]^T
\]

\[
\begin{align*}
  f_1 &= \lambda_1 \cdot m_1 \\
  f_2 &= \lambda_2 \cdot m_3
\end{align*}
\]
A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle$ where:

- $P$ and $T$ are the sets of places and transitions respectively.
- $\text{Pre}, \text{Post} \in \mathbb{R}^{\left| P \right| \times \left| T \right| \geq 0}$ are the pre and post incidence matrices.
- $\lambda \in \mathbb{R}^{\left| T \right| > 0}$ is the firing rate of transition.

$$m = [1.5 \ 0.5 \ 0]^T$$

$$f_1 = \lambda_1 \cdot m_2$$

$$f_2 = \lambda_2 \cdot m_3$$
Definition (contPN)

A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle$ where:

- $P$ and $T$ are the sets of places and transitions respectively.
- $\text{Pre}, \text{Post} \in \mathbb{R}^{\left|P\right| \times \left|T\right|}_{\geq 0}$ are the pre and post incidence matrices.
- $\lambda \in \mathbb{R}_{>0}^{\left|T\right|}$ is the firing rate of transition.

The state equation of uncontrolled contPN

$$\dot{m} = C \cdot f \quad (2)$$
A control method for timed distributed continuous Petri nets

Distributed Continuous Petri nets

Continuous Petri nets (contPN)

Definition (contPN)

A continuous Petri net (contPN) \( \mathcal{N} \) is a tuple \( \mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle \) where:

- \( P \) and \( T \) are the sets of places and transitions respectively.
- \( \text{Pre}, \text{Post} \in \mathbb{R}_{\geq 0}^{|P| \times |T|} \) are the pre and post incidence matrices.
- \( \lambda \in \mathbb{R}_{>0}^{|T|} \) is the firing rate of transition.

Definition (contPN system)

A contPN system is a pair \( \langle \mathcal{N}, m_0 \rangle \) where \( \mathcal{N} \) is a contPN and \( m_0 \in \mathbb{R}_{\geq 0}^{|P|} \) is the initial marking.
A continuous Petri net (contPN) $\mathcal{N}$ is a tuple $\mathcal{N} = \langle P, T, \text{Pre}, \text{Post}, \lambda \rangle$ where:

- $P$ and $T$ are the sets of places and transitions respectively.
- $\text{Pre}, \text{Post} \in \mathbb{R}^{\mid P \mid \times \mid T \mid}$ are the pre and post incidence matrices.
- $\lambda \in \mathbb{R}^{\mid T \mid}$ is the firing rate of transition.

Left and right natural annullers of the token flow matrix $\mathbf{C}$ are called $P$-semiflows (denoted by $r$) and $T$-semiflows (denoted by $s$), respectively.

If $\exists r > 0$, $r \cdot \mathbf{C} = 0$, then the net is said to be conservative.

If $\exists s > 0$, $\mathbf{C} \cdot s = 0$ it is said to be consistent.
Definition (MTS)

A contPN is mono T-semiflow (MTS) if it is conservative, consistent and has only one minimal T-semiflow.

Figure: A mono-T-semiflow manufacturing system
Definition (DcontPN)

A Distributed timed contPN (DcontPN) system is a set of contPN systems connected through channels modeled as places.
Subsystems

- $K$: denote the set of subsystems of a given DcontPN
- $P^k$: The set of places of subsystem $k \in K$
- $T^k$: The set of transitions of subsystem $k \in K$
- $P^k \cap P^l = \emptyset$ and $T^k \cap T^l = \emptyset$, $\forall k, l \in K$, $k \neq l$

![Diagram of a simple DcontPN](image)

**Figure:** A simple DcontPN
Channels

- The communication from subsystem $k$ to $l$:
  \[ P^{k,l} = \{ p \in P | \bullet_p \in T^k, \ p^* \in T^l, \ p \notin P^q \ \forall q \in K \} \]

- $P^{k,*}$: The set of all output channels of subsystem $k$: $P^{k,*} = \bigcup_{l \in K, l \neq k} P^{k,l}$

- $P^{*,k}$: The set of all input channels of subsystem $k$: $P^{*,k} = \bigcup_{l \in K, l \neq k} P^{l,k}$

Figure: A simple DcontPN
Subsystem 1

- $k = 1$
- $P^1 = \{p_1, p_2, p_3, p_4, p_5\}$
- $T^1 = \{t_1, t_2, t_3, t_4\}$

Figure: A simple DcontPN
Subsystem 2

- $k = 2$
- $P^2 = \{p_6, p_7, p_8\}$
- $T^2 = \{t_5, t_6, t_7\}$

Figure: A simple DcontPN
Channels

- $P^{1,2} = \{p_b\}$
- $P^{2,1} = \{p_a\}$
- $P^{*,1} = P^{2,*} = \{p_a\}$
- $P^{*,2} = P^{1,*} = \{p_b\}$

Figure: A simple DcontPN
Outline

- Introduction
- Distributed Continuous Petri nets
  - Continuous Petri nets (contPN)
  - Distributed Continuous Petri nets (DcontPN)
- Control Strategy
  - Control Actions
  - Problem Statement
- Controller for DcontPN
  - Design of a Distributed Controller
  - Reachability of Target Marking
- Case Study
- Conclusion
Control Actions

Definition

The controlled flow, $w$, of a timed DcontPN is defined as $w(\tau) = f(\tau) - u(\tau)$, with $0 \leq u(\tau) \leq f(\tau)$, where $f$ is the flow of the uncontrolled system, and $u$ is the control action.

The overall behaviour of the controlled system:

$$\dot{m} = C \cdot [f - u] = C \cdot w$$

$$0 \leq u \leq f$$

(3)

The integral of the controlled flow of a transition $t_j$ over $(\tau_a, \tau_b)$:

$$x(t_j) = \int_{\tau_a}^{\tau_b} w(t_j) d\tau$$
Problem Statement

- Design of local controller for each subsystem in order to reach their target marking from their initial marking.

- Each subsystem is equipped with its own controller that computes the control actions that drive the subsystem to the target marking.

- Controllers communicate between them during computation.
A Control Method for Timed Distributed Continuous Petri nets

Controller Strategy

Problem Statement

Example

\[ m_0(P^1) = [1 \ 2 \ 1 \ 1 \ 2]^T \quad m_0(P^2) = [1 \ 3 \ 2]^T \quad m_0(p_a) = 0 \quad m_0(p_b) = 1 \]

\[ m_f(P^1) = [2 \ 2 \ 2 \ 1 \ 1]^T \quad m_f(P^2) = [1 \ 3 \ 2]^T \]
Example

\[ \mathbf{m}_0(P^1) = [1 \ 2 \ 1 \ 1 \ 2]^T \quad \mathbf{m}_0(P^2) = [1 \ 3 \ 2]^T \quad m_0(p_a) = 0 \quad m_0(p_b) = 1 \]
\[ \mathbf{m}_f(P^1) = [2 \ 2 \ 2 \ 1 \ 1]^T \quad \mathbf{m}_f(P^2) = [1 \ 3 \ 2]^T \]
Problem Statement

Example

\[ x(t_1) = 1 \]
\[ x(t_2) = 0 \]
\[ x(t_3) = 0 \]
\[ x(t_4) = 0 \]
A Control Method for Timed Distributed Continuous Petri nets

Controller Strategy

Problem Statement

Example

\[ x(t_1) = 1 \]
\[ x(t_2) = 0 \]
\[ x(t_3) = 0 \]
\[ x(t_4) = 0 \]
**Example**

\begin{align*}
  x(t_5) &= 0 \\
  x(t_6) &= 0 \\
  x(t_7) &= 0
\end{align*}
A Control Method for Timed Distributed Continuous Petri nets

Controller Strategy

Problem Statement

Example

\[ x(t_5) = 0 \]
\[ x(t_6) = 0 \]
\[ x(t_7) = 0 \]
Example

\[ S1 : x(t_1) = 1 \quad x(t_2) = 0 \quad x(t_3) = 0 \quad x(t_4) = 0 \]

\[ S2 : x(t_5) = 0 \quad x(t_6) = 0 \quad x(t_7) = 0 \]
Example

\[ \mathbf{m}_f(P^1) = [2 2 2 1 1]^T \text{ is reachable from } \mathbf{m}_0(P^1) = [1 2 1 1 2]^T \]

\[ \mathbf{m}_f(P^2) = [1 3 2]^T \text{ is reachable from } \mathbf{m}_0(P^2) = [1 3 2]^T \]
If subsystems are connected through the communication places $p_a$ and $p_b$ with $m_0(p_a) = 0$, $m_0(p_b) = 0$, the target markings are not reachable simultaneously.
Outline

1. Introduction
2. Distributed Continuous Petri nets
   - Continuous Petri nets (contPN)
   - Distributed Continuous Petri nets (DcontPN)
3. Control Strategy
   - Control Actions
   - Problem Statement
4. Controller for DcontPN
   - Design of a Distributed Controller
   - Reachability of Target Marking
5. Case Study
6. Conclusion
Design of a distributed controller

Assumptions:

(A1) The DcontPN is composed of two subsystems that are MTS.

(A2) The target marking $m_f$ is strictly positive and reachable at the overall system.

(A3) The following equalities are satisfied $\forall p_a \in P^{2,1}$ $\forall p_b \in P^{1,2}$

$$\sum_{t \in p_b^\bullet} Pre(p_b, t) \cdot s^2(t) = \sum_{t \in p_a^\bullet} Post(p_a, t) \cdot s^2(t)$$

$$\sum_{t \in p_b^\bullet} Post(p_b, t) \cdot s^1(t) = \sum_{t \in p_a^\bullet} Pre(p_a, t) \cdot s^1(t)$$

(4)
(A3) The following equalities are satisfied $\forall p_a \in P^{2,1} \ \forall p_b \in P^{1,2}$

$$
\sum_{t \in p^p_b} Pre(p_b, t) \cdot s^2(t) = \sum_{t \in \bullet p_a} Post(p_a, t) \cdot s^2(t)
$$

$$
\sum_{t \in \bullet p_b} Post(p_b, t) \cdot s^1(t) = \sum_{t \in p^p_a} Pre(p_a, t) \cdot s^1(t)
$$

Figure: A DcontPN not satisfying (A3)
(A3) The following equalities are satisfied: \( \forall p_a \in P^{2,1} \ \forall p_b \in P^{1,2} \)

\[
\sum_{t \in p_b} Pre(p_b, t) \cdot s^2(t) = \sum_{t \in p_a} Post(p_a, t) \cdot s^2(t)
\]

\[
\sum_{t \in p_b} Post(p_b, t) \cdot s^1(t) = \sum_{t \in p_a} Pre(p_a, t) \cdot s^1(t)
\]
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

A Control Method for Timed Distributed Continuous Petri nets

Design of a distributed controller

**Algorithm**

**Input:** $C^1, m_0(P^1), m_f(P^1), \text{Pre}(P^1, T^1), \text{Post}(P^1, T^1), \gamma$

- Solve

\[
\begin{align*}
\min & \quad 1^T \cdot \bar{x} \\
\text{s.t.} & \quad m_f(P^1) - m_0(P^1) = C^1 \cdot \bar{x}, \\
& \quad \bar{x} \geq 0
\end{align*}
\]  

(7)

- For every $p \in P^{2,1}$ calculate

\[
q^\text{REQ}_p = \left( \sum_{t \in p \cdot} \text{Pre}(p, t) \cdot \bar{x}(t) \right) - m_0(p), \quad \forall p \in P^{2,1}
\]

- Send $q^\text{REQ}_p$, $\forall p \in P^{2,1}$ to the other subsystem

- Receive $r^\text{REQ}_p$, $\forall p \in P^{1,2}$ from the other subsystem

- Calculate

\[
h^1_p = \left( \sum_{t \in \bullet p} \text{Post}(p, t) \cdot \bar{x}(t) \right) - r^\text{REQ}_p, \quad \forall p \in P^{1,2}
\]

- If $\min_{p \in P^{1,2}} \{h^1_p\} < \gamma$ then solve

\[
\begin{align*}
\min & \quad 1^T \cdot x \\
\text{s.t.} & \quad m_f(P^1) - m_0(P^1) = C^1 \cdot x, \\
& \quad \left( \sum_{t \in \bullet p} \text{Post}(p, t) \cdot x(t) \right) \geq r^\text{REQ}_p + \gamma, \quad \forall p \in P^{1,2} \\
& \quad x \geq 0
\end{align*}
\]  

(8)

Else $x = \bar{x}$ End

Hanife Apaydın Özkan, Jorge Júlvez, Cristian Mahulea, Manuel Silva
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Example

\[
\mathbf{m}_0(P^1) = [1 \ 2 \ 1 \ 1 \ 2]^T, \quad \mathbf{m}_0(P^2) = [1 \ 3 \ 2]^T, \quad \mathbf{m}_0(p_a) = 0, \quad \mathbf{m}_0(p_b) = 1
\]

\[
\mathbf{m}_f(P^1) = [2 \ 2 \ 2 \ 1 \ 1]^T, \quad \mathbf{m}_f(P^2) = [1 \ 3 \ 2]^T
\]
Example

\[
\begin{align*}
\mathbf{m}_0(P^1) &= [1 \ 2 \ 1 \ 1 \ 2]^T, \\
\mathbf{m}_0(P^2) &= [1 \ 3 \ 2]^T, \\
\mathbf{m}_0(p_a) &= 0, \\
\mathbf{m}_0(p_b) &= 1 \\
\mathbf{m}_f(P^1) &= [2 \ 2 \ 2 \ 1 \ 1]^T, \\
\mathbf{m}_f(P^2) &= [1 \ 3 \ 2]^T
\end{align*}
\]
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Algorithm

Input: $C^1$, $m_0(P^1)$, $m_f(P^1)$, $\text{Pre}(P^1, T^1)$, $\text{Post}(P^1, T^1)$, $\gamma$

1. Solve

   $\min 1^T \cdot \bar{x}$

   $s.t. \quad m_f(P^1) - m_0(P^1) = C^1 \cdot \bar{x}$

   $\bar{x} \geq 0$ (9)

2. For every $p \in P^{2,1}$ calculate $q_p^{\text{REQ}} = \left( \sum_{t \in p^\bullet} \text{Pre}(p, t) \cdot \bar{x}(t) \right) - m_0(p)$, $\forall p \in P^{2,1}$

3. Send $q_p^{\text{REQ}}, \forall p \in P^{2,1}$ to the other subsystem

4. Receive $r_p^{\text{REQ}}, \forall p \in P^{1,2}$ from the other subsystem

5. Calculate $h_p^1 = \left( \sum_{t \in p^\bullet} \text{Post}(p, t) \cdot \bar{x}(t) \right) - r_p^{\text{REQ}}, \forall p \in P^{1,2}$

6. If $\min_{p \in P^{1,2}} \{h_p^1\} < \gamma$ then solve

   $\min 1^T \cdot x$

   $s.t. \quad m_f(P^1) - m_0(P^1) = C^1 \cdot x,$

   $\left( \sum_{t \in p^\bullet} \text{Post}(p, t) \cdot x(t) \right) \geq r_p^{\text{REQ}} + \gamma, \forall p \in P^{1,2}$

   $x \geq 0$ (10)

Else $x = \bar{x}$ End
A Control Method for Timed Distributed Continuous Petri nets

Example

\[
\bar{x}(t_1) = 1 \\
\bar{x}(t_2) = 0 \\
\bar{x}(t_3) = 0 \\
\bar{x}(t_4) = 0
\]
Example

\[ \bar{x}(t_5) = 0 \]
\[ \bar{x}(t_6) = 0 \]
\[ \bar{x}(t_7) = 0 \]
Algorithm  

Input: $C^1, m_0(P^1), m_f(P^1), \text{Pre}(P^1, T^1), \text{Post}(P^1, T^1), \gamma$

1. Solve

$$\min \quad 1^T \cdot \bar{x}$$
$$\text{s.t.} \quad m_f(P^1) - m_0(P^1) = C^1 \cdot \bar{x}, \quad \bar{x} \geq 0 \quad (11)$$

2. For every $p \in P^{2,1}$ calculate $q_p^{\text{REQ}} = \left( \sum_{t \in p} \text{Pre}(p, t) \cdot \bar{x}(t) \right) - m_0(p), \quad \forall p \in P^{2,1}$

3. Send $q_p^{\text{REQ}}, \forall p \in P^{2,1}$ to the other subsystem

4. Receive $r_p^{\text{REQ}}, \forall p \in P^{1,2}$ from the other subsystem

5. Calculate $h_p^1 = \left( \sum_{t \in \bullet p} \text{Post}(p, t) \cdot \bar{x}(t) \right) - r_p^{\text{REQ}}, \forall p \in P^{1,2}$

6. If $\min_{p \in P^{1,2}} \{h_p^1\} < \gamma$ then solve

$$\min \quad 1^T \cdot x$$
$$\text{s.t.} \quad m_f(P^1) - m_0(P^1) = C^1 \cdot x, \quad \left( \sum_{t \in \bullet p} \text{Post}(p, t) \cdot x(t) \right) \geq r_p^{\text{REQ}} + \gamma, \forall p \in P^{1,2} \quad (12)$$

Else $x = \bar{x}$ End
A Control Method for Timed Distributed Continuous Petri nets

Example

- **One token in input channel** $p_a$ is required for the computed control law at previous step.

- This token can be there initially or should be produced by the other system.

$$q_{p_a}^{REQ} = \bar{x}(t_1) - m_0(p_a) = 1$$
A Control Method for Timed Distributed Continuous Petri nets

- A controller for DcontPN
- Design of a distributed controller

**Example**

- No tokens are required for the computed control law
- \[ q_{p_b}^{REQ} = \bar{x}(t_7) - m_0(p_b) = -1 \]
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Algorithm

Input: $C^1, m_0(P^1), m_f(P^1), Pre(P^1, T^1), Post(P^1, T^1), \gamma$

- Solve

\[
\begin{align*}
\min \quad & 1^T \cdot \bar{x} \\
\text{s.t.} \quad & m_f(P^1) - m_0(P^1) = C^1 \cdot \bar{x}, \\
& \bar{x} \geq 0
\end{align*}
\]  

(13)

- For every $p \in P^{2,1}$ calculate $q_{p}^{REQ} = \left( \sum_{t \in p} Pre(p, t) \cdot \bar{x}(t) \right) - m_0(p), \quad \forall p \in P^{2,1}$

- Send $q_{p}^{REQ}, \quad \forall p \in P^{2,1}$ to the other subsystem

- Receive $r_{p}^{REQ}, \quad \forall p \in P^{1,2}$ from the other subsystem

- Calculate $h_{p}^1 = \left( \sum_{t \in \bullet p} Post(p, t) \cdot \bar{x}(t) \right) - r_{p}^{REQ}, \quad \forall p \in P^{1,2}$

- If $\min_{p \in P^{1,2}} \{h_{p}^1\} < \gamma$ then solve

\[
\begin{align*}
\min \quad & 1^T \cdot x \\
\text{s.t.} \quad & m_f(P^1) - m_0(P^1) = C^1 \cdot x, \\
& \left( \sum_{t \in \bullet p} Post(p, t) \cdot x(t) \right) \geq r_{p}^{REQ} + \gamma, \quad \forall p \in P^{1,2} \\
& x \geq 0
\end{align*}
\]  

(14)

Else $x = \bar{x}$ End

Hanife Apaydın Özkan, Jorge Júlvez, Cristian Mahulea, Manuel Silva

A Control Method for Timed Distributed Continuous Petri nets
Example

Communication between controllers

$q^{\text{REQ}}_{pa} = 1$

$q^{\text{REQ}}_{pb} = -1$
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Example
Algorithm

Input: $C^1$, $m_0(P^1)$, $m_f(P^1)$, $\text{Pre}(P^1, T^1)$, $\text{Post}(P^1, T^1)$, $\gamma$

- Solve

\[
\begin{align*}
\min & \quad 1^T \cdot \bar{x} \\
\text{s.t.} & \quad m_f(P^1) - m_0(P^1) = C^1 \cdot \bar{x}, \\
& \quad \bar{x} \geq 0
\end{align*}
\] (15)

- For every $p \in P^{2,1}$ calculate $q_{p}^{\text{REQ}} = \left( \sum_{t \in p \cdot} \text{Pre}(p, t) \cdot \bar{x}(t) \right) - m_0(p), \forall p \in P^{2,1}$

- Send $q_{p}^{\text{REQ}}, \forall p \in P^{2,1}$ to the other subsystem

- Receive $r_{p}^{\text{REQ}}, \forall p \in P^{1,2}$ from the other subsystem

- Calculate $h_{p}^1 = \left( \sum_{t \in \bullet_p} \text{Post}(p, t) \cdot \bar{x}(t) \right) - r_{p}^{\text{REQ}}, \forall p \in P^{1,2}$

- If $\min_{p \in P^{1,2}} \{h_{p}^1\} < \gamma$ then solve

\[
\begin{align*}
\min & \quad 1^T \cdot x \\
\text{s.t.} & \quad m_f(P^1) - m_0(P^1) = C^1 \cdot x, \\
& \quad \left( \sum_{t \in \bullet_p} \text{Post}(p, t) \cdot x(t) \right) \geq r_{p}^{\text{REQ}} + \gamma, \forall p \in P^{1,2} \\
& \quad x \geq 0
\end{align*}
\] (16)

Else $x = \bar{x}$ End
Example

\[ x(t_1) = 1 \]
\[ x(t_2) = 0 \]
\[ x(t_3) = 0 \]
\[ x(t_4) = 0 \]

\[ r_{p_b}^{REQ} = -1 \]

\[ h_{p_b}^1 = x(4) - r_{p_b}^{REQ} = 1 \]

- A surplus of one token is remained in \( p_b \).
- It is not necessary to produce more tokens in the output channel.
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Example

\[ \bar{x}(t_5) = 0 \]
\[ \bar{x}(t_6) = 0 \]
\[ \bar{x}(t_7) = 0 \]

\[ h_{pa}^2 = \bar{x}(5) - r_{pa}^{REQ} \]
\[ = -1 \]

- One token should be produced in the output channel
- The control law should be recomputed
Algorithm

Input: $C^1, m_0(P^1), m_f(P^1), \text{Pre}(P^1, T^1), \text{Post}(P^1, T^1), \gamma$

1. Solve

$$\begin{align*}
1^T \cdot \bar{x} \\
\text{s.t.} \quad m_f(P^1) - m_0(P^1) &= C^1 \cdot \bar{x}, \\
\bar{x} &\geq 0
\end{align*}$$

2. For every $p \in P^{2,1}$ calculate

$$q_p^{\text{REQ}} = \left( \sum_{t \in p} \text{Pre}(p, t) \cdot \bar{x}(t) \right) - m_0(p), \quad \forall p \in P^{2,1}$$

3. Send $q_p^{\text{REQ}}, \forall p \in P^{2,1}$ to the other subsystem

4. Receive $r_p^{\text{REQ}}, \forall p \in P^{1,2}$ from the other subsystem

5. Calculate

$$h^1_p = \left( \sum_{t \in p} \text{Post}(p, t) \cdot \bar{x}(t) \right) - r_p^{\text{REQ}}, \forall p \in P^{1,2}$$

6. If $\min_{p \in P^{1,2}} \{ h^1_p \} < \gamma$ then solve

$$\begin{align*}
1^T \cdot x \\
\text{s.t.} \quad m_f(P^1) - m_0(P^1) &= C^1 \cdot x, \\
\left( \sum_{t \in p} \text{Post}(p, t) \cdot x(t) \right) &\geq r_p^{\text{REQ}} + \gamma, \quad \forall p \in P^{1,2} \\
x &\geq 0
\end{align*}$$

Else $x = \bar{x}$ End
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Design of a distributed controller

Example

\[
\begin{align*}
\bar{x}(t_1) &= x(t_1) = 1 \\
\bar{x}(t_2) &= x(t_2) = 0 \\
\bar{x}(t_3) &= x(t_3) = 0 \\
\bar{x}(t_4) &= x(t_4) = 0
\end{align*}
\]

\[h_{p_b}^1 \geq 0 \Rightarrow\]
No new computation is required
A Control Method for Timed Distributed Continuous Petri nets

Example

$h_{pa}^2 < 0 \Rightarrow$

New control law is obtained

\[ x(t_5) = 1 \]
\[ x(t_6) = 1 \]
\[ x(t_7) = 1 \]
Example

\[
\begin{align*}
\mathbf{m}_0(P^1) &= [1 \ 2 \ 1 \ 1 \ 2]^T, \quad \mathbf{m}_0(P^2) = [1 \ 3 \ 2]^T, \quad m_0(p_a) = 0, \quad m_0(p_b) = 1 \\
\mathbf{m}_f(P^1) &= [2 \ 2 \ 2 \ 1 \ 1]^T, \quad \mathbf{m}_f(P^2) = [1 \ 3 \ 2]^T \\
\mathbf{m}_f(p_a) &= 0, \quad \mathbf{m}_f(p_b) = 0
\end{align*}
\]
Reachability of the target marking

Theorem

Let $\mathcal{N}$ be a DcontPN satisfying assumptions (A1), (A2) and (A3). Algorithm 1 computes a control law that:

- drives the subsystems from $m_0(P^1)$ and $m_0(P^2)$ to target markings $m_f(P^1)$ and $m_f(P^2)$, simultaneously.
- the final markings of channels satisfy $m_f(p) \geq \gamma \ \forall p \in P^{1,2} \cup P^{2,1}$
Theorem

Let $\mathcal{N}$ be a $\text{DcontPN}$ satisfying assumptions (A1) and (A3). Algorithm 1 computes a control law that:

- drives the subsystems from $m_0(P^1)$ and $m_0(P^2)$ to target markings $m_f(P^1)$ and $m_f(P^2)$, simultaneously.
- the final markings of channels satisfy $m_f(p) \geq \gamma \ \forall p \in P^{1,2} \cup P^{2,1}$

iff the target marking is reachable.
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Reachability of the target marking

Example

\[ S_1 : x(t_1) = 1 \quad x(t_2) = 0 \quad x(t_3) = 0 \quad x(t_4) = 0 \]
\[ S_2 : x(t_5) = 1 \quad x(t_6) = 1 \quad x(t_7) = 1 \]
Since one of the controller can not implement the computed control law, it is concluded that target are not reachable simultaneously while

\[ m_0(p_a) = m_0(p_b) = 0. \]
A Control Method for Timed Distributed Continuous Petri nets

A controller for DcontPN

Reachability of the target marking

Outline

Introduction

Distributed Continuous Petri nets
  - Continuous Petri nets (contPN)
  - Distributed Continuous Petri nets (DcontPN)

Control Strategy
  - Control Actions
  - Problem Statement

Controller for DcontPN
  - Design of a Distributed Controller
  - Reachability of Target Marking

Case Study

Conclusion
Case study
A Control Method for Timed Distributed Continuous Petri nets

Case study

Subsystem 1

\[ P^1 = \{ p_1, p_2, p_3, p_4, p_5 \ldots p_{12}, p_{13} \} \]
Subsystem 2

\[ P^2 = \{ p_{14}, p_{15}, p_{16}, p_{17}, p_{18} \ldots p_{25}, p_{26} \} \]
Channels

\[ P_{2,1} = \{ p_{c1}, p_{c3}, p_{c5} \} \quad P_{1,2} = \{ p_{c2}, p_{c4}, p_{c6} \} \]
Initial State

A Control Method for Timed Distributed Continuous Petri nets

Case study

Hanife Apaydm Özkan, Jorge Jülvez, Cristian Mahulea, Manuel Silva
A Control Method for Timed Distributed Continuous Petri nets

Case study

Final State

Subsystem1

Subsystem2

Hanife Apaydün Özkan, Jorge Jülvez, Cristian Mahulea, Manuel Silva
Subsystem 1:

Step 1

\[ \bar{x}(t_1) = \bar{x}(t_7) = \bar{x}(t_8) = 1 \]
\[ \bar{x}(t_2) = \bar{x}(t_3) = \bar{x}(t_4) = \bar{x}(t_5) = \bar{x}(t_6) = x(t_9) = 0 \]
Subsystem 2:

Step 1

\[ \bar{x}(t_{10}) = \bar{x}(t_{11}) = \bar{x}(t_{12}) = \bar{x}(t_{14}) = \bar{x}(t_{15}) = 1 \]
\[ \bar{x}(t_{16}) = \bar{x}(t_{17}) = \bar{x}(t_{18}) = 0 \]
\[ \bar{x}(t_{13}) = 2 \]
Subsystem 1:

Step 2

\[ q_{p_1}^{REQ} = 0 \quad q_{p_3}^{REQ} = 0 \quad q_{p_5}^{REQ} = -1 \]
Subsystem 2

Step 2

\[ q_{p_{c2}}^{REQ} = 1 \quad q_{p_{c4}}^{REQ} = 1 \quad q_{p_{c6}}^{REQ} = 0 \]
**Subsystem 1**

Step 3-4

\[ r_{p_{c2}}^{REQ} = 1 \quad r_{p_{c4}}^{REQ} = 1 \quad r_{p_{c6}}^{REQ} = 0 \]
A Control Method for Timed Distributed Continuous Petri nets

Case study

Subsystem 2

Step 3-4

\[ r_{p_{c1}}^{REQ} = 0 \quad r_{p_{c3}}^{REQ} = 0 \quad r_{p_{c5}}^{REQ} = -1 \]
Subsystem 1

Step 5

\[ h^{1}_{p_{c2}} = -1 \quad h^{1}_{p_{c4}} = -1 \quad h^{1}_{p_{c6}} = 0 \]
Subsystem 2

Step 5

\[ h_{pc1}^2 = 1 \quad h_{pc3}^2 = 1 \quad h_{pc5}^2 = 1 \]
Subsystem 1

Step 6

\[ h^1_{p_{c2}} = -1 \quad h^1_{p_{c4}} = -1 \quad h^1_{p_{c6}} = 0 \]

\[ x(t_1) = x(t_7) = x(t_8) = 2 \]
\[ x(t_2) = x(t_3) = x(t_4) = x(t_5) = x(t_6) = x(t_9) = 1 \]
A Control Method for Timed Distributed Continuous Petri nets

Case study

Subsystem 2

Step 6

\[ h_{p_{c1}}^2 = 1 \quad h_{p_{c3}}^2 = 1 \quad h_{p_{c5}}^2 = 1 \]

\[ x(t_{10}) = x(t_{11}) = x(t_{12}) = x(t_{14}) = x(t_{15}) = 1 \]

\[ x(t_{16}) = x(t_{17}) = x(t_{18}) = 0 \quad x(t_{13}) = 2 \]
Results:

Channels

\[ m_f(p_{c1}) = m_f(p_{c2}) = m_f(p_{c3}) = m_f(p_{c4}) = m_f(p_{c5}) = 0 \]
\[ m_f(p_{c6}) = 1 \]
Outline

1. Introduction
2. Distributed Continuous Petri nets
   - Continuous Petri nets (contPN)
   - Distributed Continuous Petri nets (DcontPN)
3. Control Strategy
   - Control Actions
   - Problem Statement
4. Controller for DcontPN
   - Design of a Distributed Controller
   - Reachability of Target Marking
5. Case Study
6. Conclusion
Conclusion

- This paper has focused on distributed systems modeled by continuous Petri nets.
- Each subsystem is modeled as a subnet, and the communication among subsystems is achieved by means of places.
- In the framework of distributed continuous Petri nets a control problem has been considered.
- The approach developed here is based on the design of a local controller for each subsystem.
- It is proved that, under certain assumptions on the system, the proposed algorithm for the controllers always yields an appropriate control law.
- Moreover, we establish a necessary and sufficient condition for the reachability of the target marking in every subsystem simultaneously.
A paper based on these results has been submitted to 2010 American Control Conference.

The extended work is under preparation for the journal version.