An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

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

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

December 9, 2009
Outline

1. Introduction
2. Timed Continuous Petri net (contPN)
   - Definition
   - Controlled contPN
3. Control Strategy
   - Computation of Linear Trajectories
   - A Heuristics for Minimum Time Control
4. Closed Loop Control
5. Case Study
6. Conclusion
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Outline

1. Introduction
2. Timed Continuous Petri net (contPN)
   - Definition
   - Controlled contPN
3. Control Strategy
   - Computation of Linear Trajectories
   - A Heuristics for Minimum Time Control
4. Closed Loop Control
5. Case Study
6. Conclusion
Introduction

- Controlling the system from $m_0$ to $m_f$ through the linear trajectory by minimizing the time.

- In order to reduce the time, controlling the system from $m_0$ to $m_f$ through a PWL trajectory.

- In order to minimize noise effect, controlling the system online by closed loop control approach.
Outline

- Introduction
- Timed Continuous Petri net (contPN)
  - Definition
  - Controlled contPN
- Control Strategy
  - Computation of Linear Trajectories
  - A Heuristics for Minimum Time Control
- Closed Loop Control
- Case Study
- Conclusion
Timed Continuous Petri nets (contPN)

**Definition**

A (deterministic) contPN system is a tuple $\langle P, T, Pre, Post, \lambda, m_0 \rangle$ with the set of places $P$, the set of transitions $T$, pre and post matrices $Pre, Post \in \mathbb{R}^{\mid P \mid \times \mid T \mid}$. $\lambda \in \mathbb{R}^{|T|}$ is the firing rate vector and $m_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $C = Post - Pre$ is incidence matrix.

Mostly used two server semantics for contPN

- Finite Server Semantics
- Infinite Server Semantics

Hanife Apaydün Özkan, Jorge Júlvez, Cristian Mahulea, Manuel Silva
Timed Continuous Petri nets (contPN)

**Definition**

A (deterministic) contPN system is a tuple $\langle P, T, Pre, Post, \lambda, m_0 \rangle$ with the set of places $P$, the set of transitions $T$, pre and post matrices $Pre, Post \in \mathbb{R}^{|P| \times |T|}$. $\lambda \in \mathbb{R}^{|T|}$ is the firing rate vector and $m_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $C = Post - Pre$ is incidence matrix.

Mostly used two server semantics for contPN

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

Definition

A (deterministic) contPN system is a tuple \( \langle P, T, Pre, Post, \lambda, m_0 \rangle \) with the set of places \( P \), the set of transitions \( T \), pre and post matrices \( Pre, Post \in \mathbb{R}^{|P| \times |T|} \). \( \lambda \in \mathbb{R}^{|T|} \) is the firing rate vector and \( m_0 \in \mathbb{R}^{|P|} \geq 0 \) is the initial state. \( C = Post - Pre \) is incidence matrix.

For a contPN under infinite server semantics:

\[
  f(\tau, t_j) = f_j = \lambda_j \cdot \min_{p_i \in \bullet t_j} \left\{ \frac{m(\tau, p_i)}{Pre[p_i, t_j]} \right\}
\]
Timed Continuous Petri nets (contPN)

Definition

A (deterministic) contPN system is a tuple \( \langle P, T, \text{Pre}, \text{Post}, \lambda, m_0 \rangle \) with the set of places \( P \), the set of transitions \( T \), pre and post matrices \( \text{Pre}, \text{Post} \in \mathbb{R}^{|P| \times |T|} \). \( \lambda \in \mathbb{R}^{|T|} \) is the firing rate vector and \( m_0 \in \mathbb{R}^{|P|} \geq 0 \) is the initial state. \( C = \text{Post} - \text{Pre} \) is incidence matrix.

\[ m = [m_1 \ m_2 \ m_3]^T \]
\[ \lambda = [\lambda_1 \ \lambda_2]^T \]
\[ f_1 = \lambda_1 \cdot \min\{m_1, m_2\} \]
\[ f_2 = \lambda_2 \cdot m_3 \]
A (deterministic) contPN system is a tuple \( \langle P, T, \text{Pre}, \text{Post}, \lambda, m_0 \rangle \) with the set of places \( P \), the set of transitions \( T \), pre and post matrices \( \text{Pre}, \text{Post} \in \mathbb{R}^{|P| \times |T|} \). \( \lambda \in \mathbb{R}^{|T|}_>0 \) is the firing rate vector and \( m_0 \in \mathbb{R}^{|P|}_\geq0 \) is the initial state. \( C = \text{Post} - \text{Pre} \) is incidence matrix.

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

\[ f_1 = \lambda_1 \cdot m_1 \]

\[ f_2 = \lambda_2 \cdot m_3 \]
Timed Continuous Petri Nets (contPN)

Definition

A (deterministic) contPN system is a tuple \( \langle P, T, \text{Pre}, \text{Post}, \lambda, m_0 \rangle \) with the set of places \( P \), the set of transitions \( T \), pre and post matrices \( \text{Pre}, \text{Post} \in \mathbb{R}^{\left| P \right| \times \left| T \right|} \). \( \lambda \in \mathbb{R}^{\left| T \right|} \) is the firing rate vector and \( m_0 \in \mathbb{R}^{\left| P \right|} \geq 0 \) is the initial state. \( C = \text{Post} - \text{Pre} \) is incidence matrix.

\[
m = \begin{bmatrix} 1.5 & 0.5 & 0 \end{bmatrix}^T
\]

\[
f_1 = \lambda_1 \cdot m_2
\]

\[
f_2 = \lambda_2 \cdot m_3
\]
A contPN with infinite server semantics is a PWL system with polyhedral regions. For a region $\mathcal{R}^z$, the constraint matrix is $\Pi^z : T \times P \to \mathbb{R}^+$:

$$\Pi^z[t_j, p_i] = \begin{cases} \frac{1}{Pre[p_i, t_j]}, & \text{if } (\forall m \in \mathcal{R}^z) \frac{m(p_i)}{Pre[p_i, t_j]} = \min_{p_h \in \bullet t_j} \left\{ \frac{m_h}{Pre[p_h, t_j]} \right\} \\ 0, & \text{otherwise} \end{cases}$$

(2)
Firing rate of transitions:

$$\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_{|T|}\}$$  \hspace{1cm} (3)

The nonlinear flow of the transitions at a given state $m$:

$$f = \Lambda \cdot \Pi(m) \cdot m$$  \hspace{1cm} (4)

The state equation of uncontrolled contPN:

$$\dot{m} = C \cdot f = C \cdot \Lambda \cdot \Pi(m) \cdot m$$  \hspace{1cm} (5)
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T \]
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Timed Continuous Petri nets (contPN)

Definition

Example

\( \lambda = [4 
1
3
1]^T, \ m_0 = [7.5 
4.5 
4 
2.1.5
5
4.2.5]^T, \ m_f = [7 
5 
5 
1 
1 
4 
5 
3]^T \)

\[
\begin{align*}
\dot{m}_1 &= m_4 - 4 \cdot m_5 \\
\dot{m}_2 &= 4 \cdot m_5 - m_2 \\
\dot{m}_3 &= m_2 - 3 \cdot m_8 \\
\dot{m}_4 &= 3 \cdot m_8 - m_4 \\
\dot{m}_5 &= m_2 - 4 \cdot m_5 \\
\dot{m}_6 &= 3 \cdot m_8 - m_2 \\
\dot{m}_7 &= m_4 - m_8 \\
\dot{m}_8 &= m_2 + m_4 - 4 \cdot m_5 + 3 \cdot m_8
\end{align*}
\]

(6)
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5.4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T \]

\[
\begin{align*}
\dot{m}_1 &= m_4 - 4 \cdot m_5 \\
\dot{m}_2 &= 4 \cdot m_5 - m_6 \\
\dot{m}_3 &= m_6 - 3 \cdot m_8 \\
\dot{m}_4 &= 3 \cdot m_8 - m_4 \\
\dot{m}_5 &= m_6 - 4 \cdot m_5 \\
\dot{m}_6 &= 3 \cdot m_8 - m_6 \\
\dot{m}_7 &= m_4 - m_8 \\
\dot{m}_8 &= m_6 + m_4 - 4 \cdot m_5 + 3 \cdot m_8
\end{align*}
\] (7)
Definition

If the flow of a transition can be reduced and even stopped, it is a controllable transition.

Definition

The flow of controlled timed contPN is denoted as $w(\tau) = f(\tau) - u(\tau)$, with $0 \leq u(\tau) \leq f(\tau)$, where $f$ is the flow of the uncontrolled system [i.e. defined as in equation (1)] and $u$ is the control action.
Under these conditions, the overall behaviour of the controlled system:

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

$$= C \cdot [\Lambda \cdot \Pi(m) \cdot m - u]$$

$$0 \leq u \leq f$$

(8)
Under these conditions, the overall behaviour of the controlled system:

\[
(w = f - u)
\]

\[
\dot{m} = C \cdot w \\
0 \leq w \leq \Lambda \cdot \Pi(m) \cdot m
\]  

(9)
Outline

1. Introduction
2. Timed Continuous Petri net (contPN)
   - Definition
   - Controlled contPN
3. Control Strategy
   - Computation of Linear Trajectories
   - A Heuristics for Minimum Time Control
4. Closed Loop Control
5. Case Study
6. Conclusion
Control Strategy

In the literature,

- Optimal steady state control problem by means of LPP has been solved (Mahulea et al. 2008)
- Transitory control problem has been solved MPC (Mahulea et al. 2008)
- A Lyapunov-function-based dynamic control algorithm is developed and it is proposed to introduce intermediate states in order to improve the time (Xu et al. 2008)
- A local controllability concept was proposed (C.R. Vázquez et al. 2008)
- ...

An Efficient Heuristics for Minimum Time Control of Continuous Petri nets
Control Strategy

Compute a control action $u$ that drives the system from the initial marking $m_0$ to a desired target marking $m_f$ by minimizing time

- Linear Trajectory (LPP)
- Piecewise Linear Trajectory (BLP)
  * Total time is reduced
  * Computational complexity is reasonable
We distinguish two cases:

(A) $m_0$ and $m_f$ are in the same region

(B) $m_0$ and $m_f$ are in different regions
(A) $m_0$ and $m_f$ are in the same region $\mathcal{R}^z$.

\[ (a) \quad m_f = m_0 + C \cdot w \cdot \tau_f \]

\[ (b) \quad 0 \leq w_j \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min \{m_{0i}, m_{fi}\}, \quad \forall j \in \{1, \ldots, |T|\} \text{ where } i \text{ satisfies } \Pi_{ji}^z \neq 0 \]
(A) $m_0$ and $m_f$ are in the same region $\mathcal{R}^z$.

LPP for linear trajectory ($x = w \cdot \tau_f$)

\[
\begin{align*}
\min \quad & \tau_f \\
\text{s.t.} \quad & m_f = m_0 + C \cdot x \\
0 \leq x_j & \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min \{m_{0i}, m_{fi}\} \cdot \tau_f, \\
\forall j & \in \{1, \ldots, |T|\} \text{ where } i \text{ satisfies } \Pi_{ji}^z \neq 0
\end{align*}
\]
(B) $m_0$ and $m_f$ are in different regions.
**Linear Trajectory**

**Input:** \( \langle N, m_0 \rangle, \ m_f \)

1. Compute the line \( \ell \) connecting \( m_0 \) and \( m_f \)
2. Compute the intersection of \( \ell \) and the crossed borders: \( m_c^1, m_c^2, \ldots, m_c^n \)
3. \( m_c^0 = m_0, \ m_c^{n+1} = m_f, \ \tau_f = 0 \)
4. **for** \( i = 0 \) **to** \( n \) **do**
   - Determine \( \tau_i \) by solving LPP in (11) with \( m_0 = m_c^i \) and \( m_f = m_c^{i+1} \)
5. **end for**

**Output:** \( \tau_1, w^1, \tau_2, w^2, \ldots, \tau_{n+1}, w^{n+1}, \ \tau_f = \sum_{i=1}^{n+1} \tau_i \)
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4.2 \ 5.5 \ 4.2 \ 4.5 \ 5.5 \ 4.2 \ 5.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Linear Trajectory

Input: \( \langle N, m_0 \rangle, m_f \)

Compute the line \( \ell \) connecting \( m_0 \) and \( m_f \)

Compute the intersection of \( \ell \) and the crossed borders: \( m_1^c, m_2^c, \ldots, m_n^c \)

\( m_0^c = m_0, \quad m_{n+1}^c = m_f, \quad \tau_f = 0 \)

for \( i = 0 \) to \( n \) do

Determine \( \tau_i \) by solving LPP in (11) with \( m_0 = m_i^c \) and \( m_f = m_{i+1}^c \)

end for

Output: \( \tau_1, w^1, \tau_2, w^2, \ldots, \tau_{n+1}, w^{n+1}, \tau_f = \sum_{i=1}^{n+1} \tau_i \)
Example

\( \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \)
### Linear Trajectory

**Input:** \( \langle N, m_0 \rangle, \ m_f \)

Compute the line \( \ell \) connecting \( m_0 \) and \( m_f \)

Compute the intersection of \( \ell \) and the crossed borders: \( m_1^c, m_2^c, \ldots, m_n^c \)

\[
m_0^c = m_0, \quad m_{n+1}^c = m_f, \quad \tau_f = 0
\]

**for** \( i = 0 \) to \( n \) **do**

Determine \( \tau_i \) by solving LPP in (11) with \( m_0 = m_i^c \) and \( m_f = m_{i+1}^c \)

**end for**

**Output:** \( \tau_1, w_1, \tau_2, w_2, \ldots, \tau_{n+1}, w_{n+1}, \tau_f = \sum_{i=1}^{n+1} \tau_i \)
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Control Strategy

Computation of Linear Trajectories

Linear Trajectory

Input: $\langle \mathcal{N}, m_0 \rangle, \ m_f$

Compute the line $\ell$ connecting $m_0$ and $m_f$

Compute the intersection of $\ell$ and the crossed borders: $m^1_c, m^2_c, \ldots, m^n_c$

$m^0_c = m_0, \ m^{n+1}_c = m_f, \ \tau_f = 0$

for $i = 0$ to $n$ do

Determine $\tau_i$ by solving LPP in (11) with $m_0 = m^i_c$ and $m_f = m^{i+1}_c$

end for

Output: $\tau_1, w^1, \tau_2, w^2, \ldots, \tau_{n+1}, w^{n+1}, \ \tau_f = \sum_{i=1}^{n+1} \tau_i$
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4.2 \ 4.5 \ 5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Example

\[ \tau_{total} = 0.2 + 0.67 = 0.87 \text{ t.u.} \]

The control law:

\[
\mathbf{u}(\tau) = \begin{cases} 
4 \cdot m_5(\tau) - 2.49 \\
m_2(\tau) - 1.67 \\
3 \cdot m_8(\tau) \\
m_4(\tau) - 1.67 
\end{cases}, \quad \text{if } 0 \leq \tau \leq 0.2
\]

\[
\begin{cases} 
4 \cdot m_5(\tau) - 1.5 \\
m_2(\tau) - 1 \\
3 \cdot m_8(\tau) \\
m_4(\tau) - 1 
\end{cases}, \quad \text{if } 0.2 < \tau \leq 0.87
\]

(12)
A Heuristics for minimum time control

In order to improve the time spent to move from $m_0$ to $m_f$, intermediate states are introduced to the trajectory.

New trajectory: $m_0 \rightarrow m^1_I \rightarrow m^2_I \rightarrow ... \rightarrow m^s_I \rightarrow m_f$
A Heuristics for minimum time control

In order to improve the time spent to move from $m_0$ to $m_f$, intermediate states are introduced to the trajectory.

New trajectory: $m_0 \rightarrow m^1_I \rightarrow m^2_I \rightarrow \ldots \rightarrow m^s_I \rightarrow m_f$
A Heuristics for minimum time control

In order to improve the time spent to move from $m_0$ to $m_f$, intermediate states are introduced to the trajectory.

New trajectory: $m_0 \rightarrow m^1_I \rightarrow m^2_I \rightarrow \ldots \rightarrow m^s_I \rightarrow m_f$
In order to improve the time spent to move from $m_0$ to $m_f$, intermediate states are introduced to the trajectory.

New trajectory: $m_0 \rightarrow m_1^I \rightarrow m_2^I \rightarrow \ldots \rightarrow m_s^I \rightarrow m_f$
We may realize two steps to obtain PWL trajectory:

1. Intermediate states on the borders (BLP1)
   All intermediate states on the borders that the line from $m_0$ to $m_f$ crosses are calculated by solving BLP1 once.

2. Interior intermediate states (BLP2)
   Each intermediate state is calculated by solving BLP2.
(A) $m_0$ and $m_f$ are in different regions
Problem to be solved BLP1

\[
\text{min} \quad \sum_{k=1}^{s+1} \tau_k
\]

\[
m^{k+1} = m^k + C \cdot x^{k+1}, \quad k \in \{0, 1, 2, .., s\} \quad (a)
\]

\[
(\Pi^k - \Pi^{k+1}) \cdot m^k = 0, \quad k \in \{1, 2, .., s\} \quad (b)
\]

\[
m_{i}^{k} \leq m_{i}^{k+1} \quad \text{if} \quad m_{0i} \leq m_{fi}
\]

\[
i \in \{1, 2..|P|\}, \quad k \in \{0, 1, 2, .., s\} \quad (c)
\]

\[
m_{i}^{k} \geq m_{i}^{k+1} \quad \text{if} \quad m_{0i} \geq m_{fi}
\]

\[
i \in \{1, 2..|P|\}, \quad k \in \{0, 1, 2, .., s\} \quad (d)
\]

\[
0 \leq x_{ji}^k \leq \lambda_j \cdot \Pi_{ji}^k \cdot \min\{m_{i}^{k-1}, m_{i}^{k}\} \cdot \tau_k,
\]

\[
\text{with } p_i \text{ st. } \Pi_{ji}^k \neq 0, \quad j \in \{1, 2..|T|\}, \quad k \in \{1, 2..s\} \quad (e)
\]
(A) $m_0$ and $m_f$ are in different regions
We may realize two steps to obtain PWL trajectory:

1. Intermediate states on the borders (BLP1)
   All intermediate states on the borders that the line from $m_0$ to $m_f$ crosses are calculated by solving BLP1 once.

2. Interior intermediate states (BLP2)
   Each intermediate state is calculated by solving BLP2.
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Control Strategy
- A Heuristics for minimum time control

Problem to be solved BLP2

\[ \min \tau_1 + \tau_2 \]

\[ m^d = m_0 + C \cdot x^1 \]
\[ m_f = m^d + C \cdot x^2, \quad (a) \]

\[ \min\{m_{f_i}, m_{0_i}\} \leq m_i^d \leq \max\{m_{f_i}, m_{0_i}\}, \quad \forall i \in \{1, 2...|P|\} \quad (b) \]

\[ 0 \leq x_j^1 \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min\{m_{0_i}, m_i^d\} \cdot \tau_1, \]
\[ \text{with } i \text{ st. } \Pi_{ji}^k \neq 0, \quad j \in \{1, 2...|T|\} \]

\[ 0 \leq x_j^2 \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min\{m_i^d, m_{f_i}\} \cdot \tau_2, \]
\[ \text{with } i \text{ st. } \Pi_{ji}^k \neq 0, \quad j \in \{1, 2...|T|\} \quad (c) \]
(A) $m_0$ and $m_f$ are in different regions
\( (A) \) \( m_0 \) and \( m_f \) are in different regions
(A) $m_0$ and $m_f$ are in different regions
(A) $m_0$ and $m_f$ are in different regions
(B) \( m_0 \) and \( m_f \) are in the same regions
(B) $m_0$ and $m_f$ are in the same regions
(B) $m_0$ and $m_f$ are in the same regions
(B) $m_0$ and $m_f$ are in the same regions
Piecewise Trajectory

Compute the line $\ell$ connecting $m_0$ and $m_f$.
Compute intermediate states on the borders that the line $\ell$ crosses.
Calculate the interior intermediate states until the time can not be improved significantly.
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
## Piecewise Trajectory

Compute the line $\ell$ connecting $m_0$ and $m_f$

Compute intermediate states on the borders that the line $\ell$ crosses.

Calculate the interior intermediate states until the time can not be improved significantly.
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Piecewise Linear Trajectory

Compute the line $\ell$ connecting $m_0$ and $m_f$

Compute intermediate states on the borders that the line $\ell$ crosses.

Calculate the interior intermediate states until the time cannot be improved significantly.
Example

\[ \lambda = [4\ 1\ 3\ 1]^T, \ m_0 = [7.5\ 4.5\ 4\ 2.1.5\ 5\ 4\ 2.5]^T, \ m_f = [7\ 5\ 5\ 1\ 1\ 4\ 5\ 3]^T. \]
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Control Strategy
A Heuristics for minimum time control

Piecewise Trajectory

Compute the line $\ell$ connecting $m_0$ and $m_f$

Compute intermediate states on the borders that the line $\ell$ crosses.

Calculate the interior intermediate states until the time can not be improved significantly.
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Example

Table: Intermediate States

<table>
<thead>
<tr>
<th>Number of int. states</th>
<th>0</th>
<th>1</th>
<th>3</th>
<th>9</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration (t.u.)</td>
<td>0.87</td>
<td>0.83</td>
<td>0.74</td>
<td>0.73</td>
<td>0.72</td>
</tr>
<tr>
<td>CPU time (sec.)</td>
<td>0.03</td>
<td>0.11</td>
<td>0.32</td>
<td>1.65</td>
<td>2.32</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Timed Continuous Petri net (contPN)
  - Definition
  - Controlled contPN
- Control Strategy
  - Computation of Linear Trajectories
  - A Heuristics for Minimum Time Control
- Closed Loop Control
- Case Study
- Conclusion
Closed Loop Control

In the real systems there may exist perturbation

\[ \Rightarrow \]

Online closed loop controller

The discrete-time representation of the continuous-time system is:

\[ m(k+1) = m(k) + \Theta \cdot C \cdot w(k) \]

\[ 0 \leq w(k) \leq \Lambda \cdot \Pi(m(k)) \cdot m(k) \]

\[ (\tau = k \cdot \Theta) \]

\( \Theta \) should be small enough to avoid spurious states:

\[ \sum_{t_j \in p^*} \lambda_j \cdot \Theta < 1 \]
Closed Loop Control
Closed Loop Control

$\mathcal{M}_0 \rightarrow \mathcal{M}_f$
Closed Loop Control

- $m'$
- $m(k)$
- $m_{\text{next}}$
- $m_f$
Closed Loop Control

$m_0$

$m(k+1)$

$m_f$
Closed Loop Control

\[ m(k) \]

\[ m'_{\text{next}} \]
Closed Loop Control

$m_0$

$m(k)$

$m(k+1)$

$m_f$
Closed Loop Control
Closed Loop Control

\[ m_i, m(k), \ldots, m_f \]
Closed Loop Control
Closed Loop Control

**Input:** \( \langle \mathcal{N}, m_0 \rangle, \ m_f, \ \text{path}, \ s \)

\[
m'_{0} = m_0 \quad m'_{s+1} = m_f \quad m(0) = m'_{0} \quad k = 0
\]

for \( j = 0 \) to \( s \) do

while \( ||m(k) - m'_{j+1}|| > \epsilon \) do

Solve the following LPP

\[
\begin{align*}
\text{max} & \quad \alpha \\
\text{s.t.} & \quad m'_{\text{next}} = m(k) + \Theta \cdot \mathbf{C} \cdot \mathbf{w} \\
& \quad m'_{\text{next}} = (1 - \alpha) \cdot m(k) + \alpha \cdot m'_{j+1} \\
& \quad 0 \leq \alpha \leq 1 \\
& \quad 0 \leq \mathbf{w} \leq \Lambda \cdot \Pi(m(k)) \cdot \min\{m(k), m'_{\text{next}}\}
\end{align*}
\]

Advance one step and obtain the new marking

\[
m(k + 1) = m(k) + \Theta \cdot \mathbf{C} \cdot (\mathbf{w} + \mathbf{z})
\]

\( k = k + 1 \)

end while

end for
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Closed Loop Control

**Input:** $\langle N, m_0 \rangle, \ m_f, \ \text{path}, \ s$

$m'^0 = m_0 \quad m'^{s+1} = m_f \quad m(0) = m'^0 \quad k = 0$

for $j = 0$ to $s$ do

while $||m(k) - m'^{j+1}|| > \epsilon$ do

Solve the following LPP

$$\max \alpha$$

s.t. $m'_{next} = m(k) + \Theta \cdot C \cdot w$

$$m'_{next} = (1 - \alpha) \cdot m(k) + \alpha \cdot m'^{j+1}$$

$0 \leq \alpha \leq 1$

$0 \leq w \leq \Lambda \cdot \Pi(m(k)) \cdot \min\{m(k), m'_{next}\}$

Advance one step and obtain the new marking

$$m(k+1) = m(k) + \Theta \cdot C \cdot (w + z)$$

$k = k + 1$

end while

end for
Closed Loop Control

**Input:** \( \langle \mathcal{N}, m_0 \rangle, m_f, \text{ path, } s \)

\[ m'_{0} = m_0 \quad m^{s+1} = m_f \quad m(0) = m'_{0} \quad k = 0 \]

for \( j = 0 \) to \( s \) do

while \( ||m(k) - m_{j+1}'|| > \epsilon \) do

Solve the following LPP

\[
\begin{align*}
\max & \quad \alpha \\
\text{s.t.} & \quad m'_{\text{next}} = m(k) + \Theta \cdot C \cdot w \\
& \quad m'_{\text{next}} = (1 - \alpha) \cdot m(k) + \alpha \cdot m_{j+1}' \\
& \quad 0 \leq \alpha \leq 1 \\
& \quad 0 \leq w \leq \Lambda \cdot \Pi(m(k)) \cdot \min\{m(k), m'_{\text{next}}\} \\
\end{align*}
\]

Advance one step and obtain the new marking

\[
m(k + 1) = m(k) + \Theta \cdot C \cdot (w + z)
\]

\( k = k + 1 \)

end while

end for
Closed Loop Control

**Input:** \( \langle N, m_0 \rangle, m_f, \text{ path, } s \)

\[
m'^0 = m_0 \quad m'^{s+1} = m_f \quad m(0) = m'^0 \quad k = 0
\]

**for** \( j = 0 \) to \( s \) **do**

**while** \( \| m(k) - m'^{j+1} \| > \epsilon \) **do**

Solve the following LPP

\[
\begin{align*}
\text{max} & \quad \alpha \\
\text{s.t.} & \quad m'_{\text{next}} = m(k) + \Theta \cdot C \cdot w \\
& \quad m'_{\text{next}} = (1 - \alpha) \cdot m(k) + \alpha \cdot m'^{j+1} \\
& \quad 0 \leq \alpha \leq 1 \\
& \quad 0 \leq w \leq \Lambda \cdot \Pi(m(k)) \cdot \min\{m(k), m'_{\text{next}}\}
\end{align*}
\]

**Advance one step and obtain the new marking**

\[
m(k + 1) = m(k) + \Theta \cdot C \cdot (w + z)
\]

\[
k = k + 1
\]

**end while**

**end for**
Closed Loop Control

**Input:** \( \langle N, m_0 \rangle, m_f, \text{path}, s \)

\[ m'^0 = m_0 \quad m'^{s+1} = m_f \quad m(0) = m'^0 \quad k = 0 \]

for \( j = 0 \) to \( s \) do

while \( \| m(k) - m'^{j+1} \| > \epsilon \) do

Solve the following LPP

\[
\begin{align*}
\max \ & \alpha \\
\text{s.t.} \ & m'_{\text{next}} = m(k) + \Theta \cdot C \cdot w \\
& m'_{\text{next}} = (1 - \alpha) \cdot m(k) + \alpha \cdot m'^{j+1} \\
& 0 \leq \alpha \leq 1 \\
& 0 \leq w \leq \Lambda \cdot \Pi(m(k)) \cdot \min\{m(k), m'_{\text{next}}\}
\end{align*}
\]

Advance one step and obtain the new marking

\[ m(k + 1) = m(k) + \Theta \cdot C \cdot (w + z) \]

\[ k = k + 1 \]

end while

end for
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 4.5 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]
Example

\[ \lambda = [4, 1, 3, 1]^T, \quad m_0 = [7.5, 4.5, 4, 2, 1.5, 5, 4, 2.5]^T, \quad m_f = [7, 5, 5, 1, 1, 4, 5, 3]^T. \]

\[ m'_1 = [7.5, 4.5, 4.26, 1.74, 1.5, 4.74, 4.26, 2.76]^T \]
\[ m'_2 = [7.5, 4.5, 4.5, 1.5, 1.5, 4.5, 4.5, 3]^T \]
\[ m'_3 = [7.45, 4.78, 4.55, 1.22, 1.22, 4.45, 4.78, 3]^T \]
Example

\( \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T \).

\( \Theta = 0.01 \)  

no noise effect

\[ m'^1 = [7.5 \ 4.5 \ 4.26 \ 1.74 \ 1.5 \ 4.74 \ 4.26 \ 2.76]^T \]

\[ m'^2 = [7.5 \ 4.5 \ 4.5 \ 1.5 \ 1.5 \ 4.5 \ 4.5 \ 3]^T \]
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ \bm{m}_0 = [7.5 \ 4.5 \ 4 \ 2 \ 1.5 \ 5 \ 4 \ 2.5]^T, \ \bm{m}_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]

\[ \bm{w}(1) = [2.02 \ 2.02 \ 0 \ 2.02]^T \]
\[ \bm{m}(1) = [7.5 \ 4.5 \ 4.02 \ 1.98 \ 1.5 \ 4.98 \ 4.02 \ 2.52]^T \]
Example

\[ \lambda = [4 \ 1 \ 3 \ 1]^T, \ m_0 = [7.5 \ 4.5 \ 4.2 \ 1.5 \ 5 \ 4.25 \ 2.5]^T, \ m_f = [7 \ 5 \ 5 \ 1 \ 1 \ 4 \ 5 \ 3]^T. \]

\[
\begin{align*}
w(2) &= [1.99 \ 1.99 \ 0 \ 1.99]^T \\
m(2) &= [7.5 \ 4.5 \ 4.04 \ 1.96 \ 1.5 \ 4.96 \ 4.04 \ 2.54]^T \\
\text{Total time} &= 0.66 \text{ t.u.}
\end{align*}
\]
Outline

- Introduction
- Timed Continuous Petri net (contPN)
  - Definition
  - Controlled contPN
- Control Strategy
  - Computation of Linear Trajectories
  - A Heuristics for Minimum Time Control
- Closed Loop Control
- Case Study
- Conclusion
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Case Study

Example
Case Study
An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

Case Study

Example
Example

<table>
<thead>
<tr>
<th>place</th>
<th>( m_0 )</th>
<th>( m_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pallet_A</td>
<td>35.0100</td>
<td>27.0000</td>
</tr>
<tr>
<td>Pallet_B</td>
<td>25.0100</td>
<td>17.0000</td>
</tr>
<tr>
<td>B2_A</td>
<td>1.0100</td>
<td>0.0100</td>
</tr>
<tr>
<td>B2_B</td>
<td>1.0100</td>
<td>0.0100</td>
</tr>
<tr>
<td>B3_Empty</td>
<td>25.0100</td>
<td>15.0000</td>
</tr>
<tr>
<td>B_3</td>
<td>0.0100</td>
<td>10.0100</td>
</tr>
</tbody>
</table>

Without intermediate states: 1000 t.u.

<table>
<thead>
<tr>
<th>place</th>
<th>( m_0 )</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( m_3 )</th>
<th>( m_4 )</th>
<th>( m_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pallet_A</td>
<td>35.0100</td>
<td>33.5000</td>
<td>31.0000</td>
<td>30.0000</td>
<td>28.5008</td>
<td>27.0000</td>
</tr>
<tr>
<td>Pallet_B</td>
<td>25.0100</td>
<td>23.5000</td>
<td>21.0000</td>
<td>20.0000</td>
<td>18.5008</td>
<td>17.0000</td>
</tr>
<tr>
<td>B2_A</td>
<td>1.0100</td>
<td>1.0100</td>
<td>0.5100</td>
<td>0.0149</td>
<td>0.0100</td>
<td>0.0100</td>
</tr>
<tr>
<td>B2_B</td>
<td>1.0100</td>
<td>1.0100</td>
<td>0.5100</td>
<td>0.0149</td>
<td>0.0100</td>
<td>0.0100</td>
</tr>
<tr>
<td>B3_Empty</td>
<td>25.0100</td>
<td>23.0000</td>
<td>20.0000</td>
<td>18.0148</td>
<td>16.5009</td>
<td>15.0000</td>
</tr>
<tr>
<td>B_3</td>
<td>0.0100</td>
<td>2.0100</td>
<td>5.0100</td>
<td>6.9952</td>
<td>8.5091</td>
<td>10.0100</td>
</tr>
</tbody>
</table>

With 4 intermediate states: 802 t.u.
Example

Closed loop control without noise effect:

Closed Loop control without noise: 800 t.u.
Example

Closed loop control under noise effect:

![Diagram of a Petri net model with transitions and places labeled as Max_A, S_M1_A, E_M1_A, S_M2_A, E_M2_A, B_2A, M3_Iddle, B_3_Empty Pallets_A, Pallets_B, M1_Iddle, M2_Iddle, B1_A, M2_AM1_A, S_M2_B, M2_B, B1_B, M1_B, E_M2_B, S_M1_B, E_M1_B, Max_B, B_2B, M3_Work, B_3, B_2B, B_3 graphs showing changes over time.

Closed Loop control with noise: 945 t.u.
Outline

- Introduction
- Timed Continuous Petri net (contPN)
  - Definition
  - Controlled contPN
- Control Strategy
  - Computation of Linear Trajectories
  - A Heuristics for Minimum Time Control
- Closed Loop Control
- Case Study
- Conclusion
Conclusion

- A control law that drives the system from initial state to target state through a linear trajectory is developed.
- Calculation of linear trajectories is based on Linear Programming.
- In order to improve the time introduce intermediate states are introduced. This can yield a piecewise linear trajectory.
- Calculation of piecewise linear trajectories is based on BiLinear Programming.
- The control law for both linear and piecewise linear trajectories assigns constant or piecewise constant flows to transitions.
- A closed loop control method is proposed to ensure that the final marking is reached.
This work was presented in ADHS’09: 3rd IFAC Conference on Analysis and Design of Hybrid Systems:


The extended work is under preparation for the journal version.