An Efficient Heuristics for Minimum Time Control of Continuous Petri nets

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

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

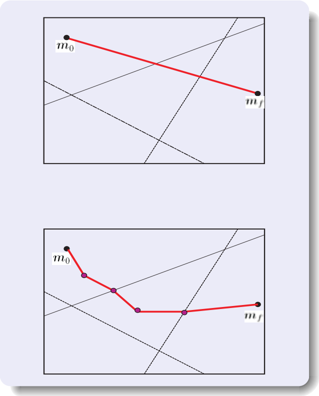
December 9, 2009

Outline

Outline

Introduction

- Controlling the system from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ through the linear trajectory by minimizing the time.
- In order to reduce the time, controlling the system from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ through a PWL trajectory.
- In order to minimize noise effect, controlling the system online by closed loop control approach.



Outline

Timed Continuous Petri nets (contPN)

Definition

A (deterministic) contPN system is a tuple $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P| \times |T|}$. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ 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 $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P|\times|T|}$. $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ 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 $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P|\times|T|}$. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is incidence matrix.

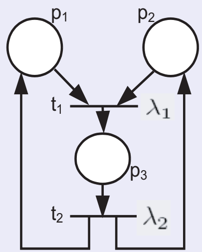
For a contPN under infinite server semantics:

$$f(\tau, t_j) = f_j = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m(\tau, p_i)}{Pre[p_i, t_j]} \right\} \quad (1)$$

Timed Continuous Petri nets (contPN)

Definition

A (deterministic) contPN system is a tuple $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P| \times |T|}$. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is incidence matrix.

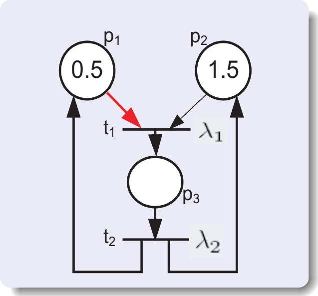


$$\boldsymbol{m} = [m_1 \ m_2 \ m_3]^T$$
$$\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2]^T$$
$$f_1 = \lambda_1 \cdot \min\{m_1, m_2\}$$
$$f_2 = \lambda_2 \cdot m_3$$

Timed Continuous Petri nets (contPN)

Definition

A (deterministic) contPN system is a tuple $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P| \times |T|}$. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is incidence matrix.



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

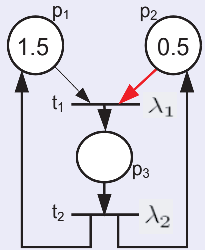
$$f_1 = \lambda_1 \cdot m_1$$

$$f_2 = \lambda_2 \cdot m_3$$

Timed Continuous Petri nets (contPN)

Definition

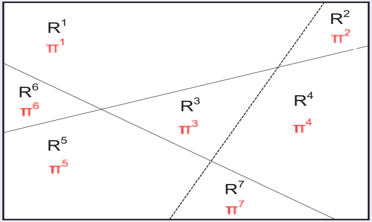
A (deterministic) contPN system is a tuple $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ with the set of places P, the set of transitions T, pre and post matrices $\boldsymbol{Pre}, \boldsymbol{Post} \in \mathbb{R}^{|P| \times |T|}$. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ is the firing rate vector and $\boldsymbol{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$ is the initial state. $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is incidence matrix.



$$\boldsymbol{m} = [1.5 \; 0.5 \; 0]^T$$
$$f_1 = \lambda_1 \cdot m_2$$
$$f_2 = \lambda_2 \cdot m_3$$

A contPN with infinite server semantics is a PWL sysem with polyhedral regions. For a region $\mathcal{R}^z$, the constraint matrix is $\mathbf{\Pi}^z : T \times P \rightarrow \mathbb{R}^+$:

$$\mathbf{\Pi}^z[t_j, p_i] = \begin{cases} \frac{1}{Pre[p_i,t_j]}, & if(\forall \boldsymbol{m} \in \mathcal{R}^z) \quad \frac{m(p_i)}{Pre[p_i,t_j]} = \min\limits_{p_h \in {}^\bullet t_j} \left\{ \frac{m_h}{Pre[p_h, t_j]} \right\} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$



Firing rate of transitions:

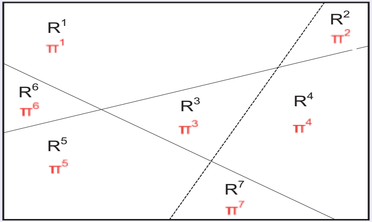
$$\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, ...\lambda_{|T|}\} \tag{3}$$

The nonlinear flow of the transitions at a given state $\boldsymbol{m}$:

$$\boldsymbol{f} = \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} \tag{4}$$

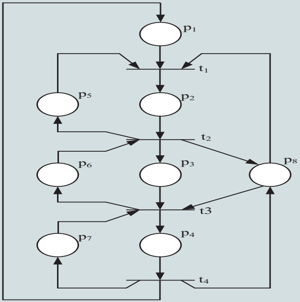
The state equation of uncontrolled contPN:

$$\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{C} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} \tag{5}$$



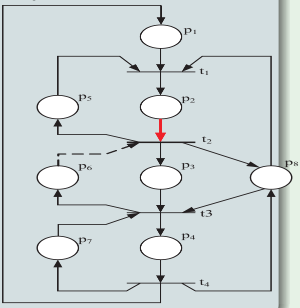
Example

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



Example

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

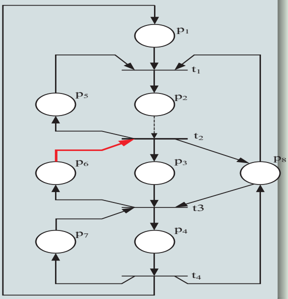


$$
\begin{cases}
\dot{m}_1 = m_4 - 4 \cdot m_5 \\
\dot{m}_2 = 4 \cdot m_5 - m_2 \\
\dot{m}_3 = m_2 - 3 \cdot m_8 \\
\dot{m}_4 = 3 \cdot m_8 - m_4 \\
\dot{m}_5 = m_2 - 4 \cdot m_5 \\
\dot{m}_6 = 3 \cdot m_8 - m_2 \\
\dot{m}_7 = m_4 - m_8 \\
\dot{m}_8 = m_2 + m_4 - 4 \cdot m_5 + 3 \cdot m_8
\end{cases} \tag{6}
$$

Example

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

$\boldsymbol{m}_f = [7\ 5\ 5\ 1\ 1\ 4\ 5\ 3]^T$



$$\left\{\begin{array}{lcl}
\dot{m}_1 & = & m_4 - 4 \cdot m_5 \\
\dot{m}_2 & = & 4 \cdot m_5 - m_6 \\
\dot{m}_3 & = & m_6 - 3 \cdot m_8 \\
\dot{m}_4 & = & 3 \cdot m_8 - m_4 \\
\dot{m}_5 & = & m_6 - 4 \cdot m_5 \\
\dot{m}_6 & = & 3 \cdot m_8 - m_6 \\
\dot{m}_7 & = & m_4 - m_8 \\
\dot{m}_8 & = & m_6 + m_4 - 4 \cdot m_5 + 3 \cdot m_8
\end{array}\right. \tag{7}$$

Controlled contPN

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 $\boldsymbol{w}(\tau) = \boldsymbol{f}(\tau) - \boldsymbol{u}(\tau)$, with $0 \leq \boldsymbol{u}(\tau) \leq \boldsymbol{f}(\tau)$, where $\boldsymbol{f}$ is the flow of the uncontrolled system [i.e. defined as in equation (1)] and $\boldsymbol{u}$ is the control action.

Under these conditions, the overall behaviour of the controlled system:

$$\begin{aligned} \dot{\boldsymbol{m}} &= \boldsymbol{C} \cdot [\boldsymbol{f} - \boldsymbol{u}] \\ &= \boldsymbol{C} \cdot [\boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} - \boldsymbol{u}] \\ & 0 \leq \boldsymbol{u} \leq \boldsymbol{f} \end{aligned} \tag{8}$$

Under these conditions, the overall behaviour of the controlled system: $(\boldsymbol{w} = \boldsymbol{f} - \boldsymbol{u})$

$$\begin{aligned} \dot{\boldsymbol{m}} &= \boldsymbol{C} \cdot \boldsymbol{w} \\ & 0 \leq \boldsymbol{w} \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}) \cdot \boldsymbol{m} \end{aligned} \tag{9}$$

Outline

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)
- ...

Control Strategy

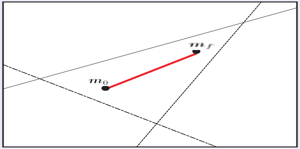
Compute a control action $\boldsymbol{u}$ that drives the system from the initial marking $\boldsymbol{m}_0$ to a desired target marking $\boldsymbol{m}_f$ by minimizing time

- Linear Trajectory (LPP)
- Piecewise Linear Trajectory (BLP)
  - * Total time is reduced
  - * Computational complexity is reasonable

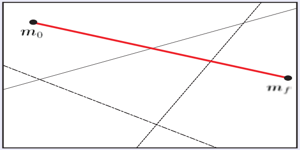
Computation of Linear Trajectories

We distinguish two cases:

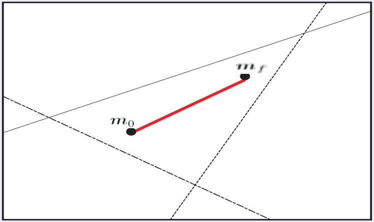
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same region



(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



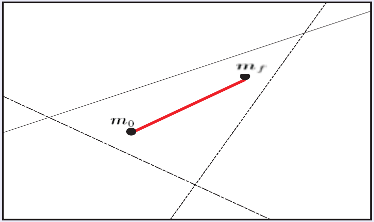
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same region $\mathcal{R}^z$.



Problem to be solved for linear trajectory

$$\begin{aligned} \min \quad & \tau_f \\ & \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{w} \cdot \tau_f && (a) \\ & 0 \le w_j \le \lambda_j \cdot \Pi^z_{ji} . \min\left\{m_{0i}, m_{fi}\right\}, \\ & \quad \forall j \in \{1, .., |T|\} \text{ where } i \text{ satisfies } \Pi^z_{ji} \neq 0 && (b) \end{aligned} \quad (10)$$

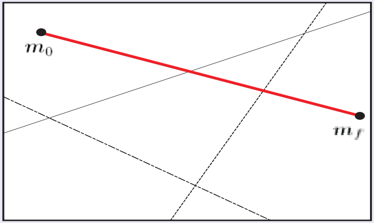
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same region $\mathcal{R}^z$.

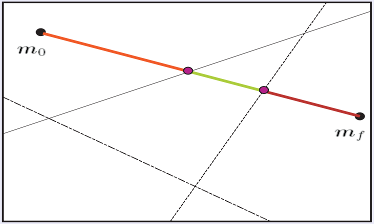


LPP for linear trajectory ($\boldsymbol{x} = \boldsymbol{w} \cdot \tau_f$)

$$\begin{aligned}
\min \quad & \tau_f \\
& \boldsymbol{m}_f = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{x} && (a) \\
& 0 \leq x_j \leq \lambda_j \cdot \Pi^z_{ji}. \min \left\{ m_{0i}, m_{f_i} \right\} \cdot \tau_f, \\
& \quad \forall\, j \in \{1, .., |T|\} \text{ where } i \text{ satisfies } \Pi^z_{ji} \neq 0 && (b)
\end{aligned} \tag{11}$$

(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions.





Linear Trajectory

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f$

Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

Compute the intersection of ℓ and the crossed borders: $\boldsymbol{m}_c^1,\ \boldsymbol{m}_c^2,\ \ldots.\boldsymbol{m}_c^n$

$\boldsymbol{m}_c^0 = \boldsymbol{m}_0,\ \boldsymbol{m}_c^{n+1} = \boldsymbol{m}_f,\ \tau_f = 0$

for $i = 0$ to n **do**

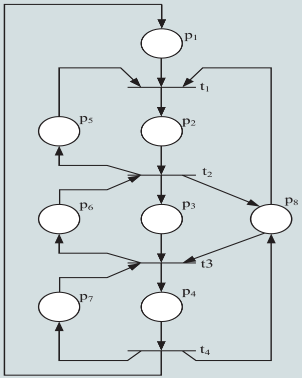
  Determine τ_i by solving LPP in (11) with $\boldsymbol{m}_0 = \boldsymbol{m}_c^i$ and $\boldsymbol{m}_f = \boldsymbol{m}_c^{i+1}$

end for

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

Example

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



Linear Trajectory

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f$

Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

Compute the intersection of ℓ and the crossed borders: $\boldsymbol{m}_c^1,\ \boldsymbol{m}_c^2,\ \ldots.\boldsymbol{m}_c^n$

$\boldsymbol{m}_c^0 = \boldsymbol{m}_0,\ \ \boldsymbol{m}_c^{n+1} = \boldsymbol{m}_f,\ \ \tau_f = 0$

for $i = 0$ to n **do**

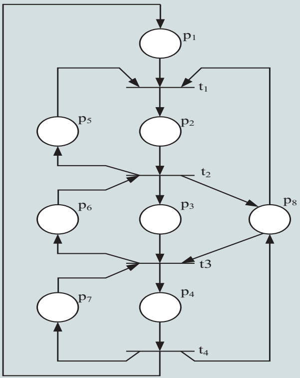
  Determine τ_i by solving LPP in (11) with $\boldsymbol{m}_0 = \boldsymbol{m}_c^i$ and $\boldsymbol{m}_f = \boldsymbol{m}_c^{i+1}$

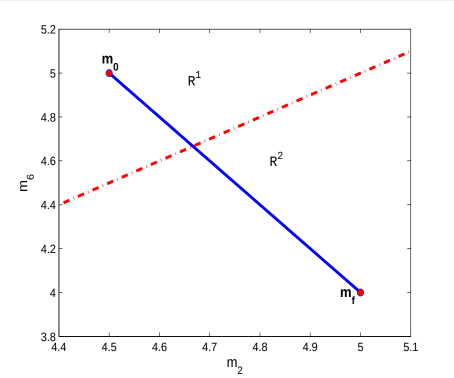
end for

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

Example

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





Linear Trajectory

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f$

Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

Compute the intersection of ℓ and the crossed borders: $\boldsymbol{m}_c^1, \ \boldsymbol{m}_c^2, \ .....\boldsymbol{m}_c^n$

$\boldsymbol{m}_c^0 = \boldsymbol{m}_0, \ \ \boldsymbol{m}_c^{n+1} = \boldsymbol{m}_f, \ \ \tau_f = 0$

for $i = 0$ to n **do**

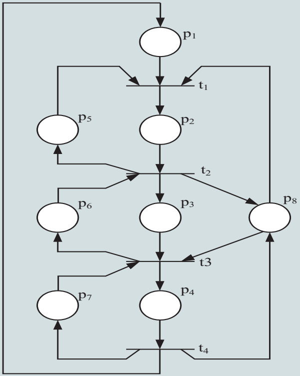
Determine τ_i by solving LPP in (11) with $\boldsymbol{m}_0 = \boldsymbol{m}_c^i$ and $\boldsymbol{m}_f = \boldsymbol{m}_c^{i+1}$

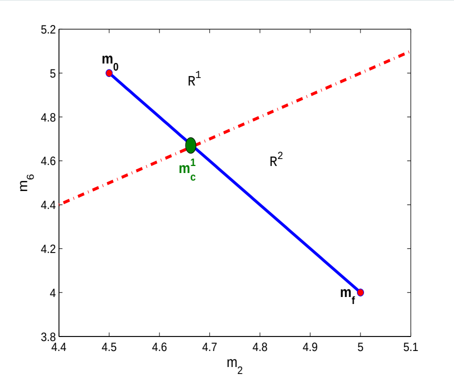
end for

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

Example

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





Linear Trajectory

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f$

Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

Compute the intersection of ℓ and the crossed borders: $\boldsymbol{m}_c^1, \ \boldsymbol{m}_c^2, \ \ldots .\boldsymbol{m}_c^n$

$\boldsymbol{m}_c^0 = \boldsymbol{m}_0, \ \ \boldsymbol{m}_c^{n+1} = \boldsymbol{m}_f, \ \ \tau_f = 0$

for $i = 0$ to n **do**

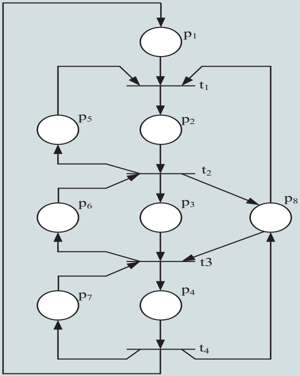
  Determine τ_i by solving LPP in (11) with $\boldsymbol{m}_0 = \boldsymbol{m}_c^i$ and $\boldsymbol{m}_f = \boldsymbol{m}_c^{i+1}$

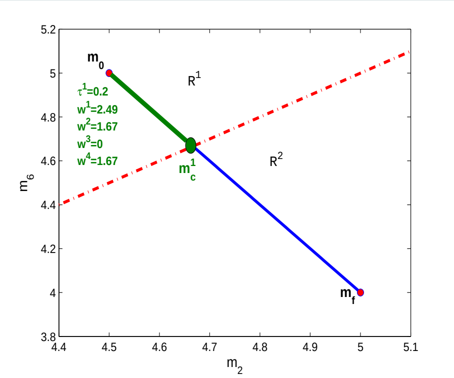
end for

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

Example

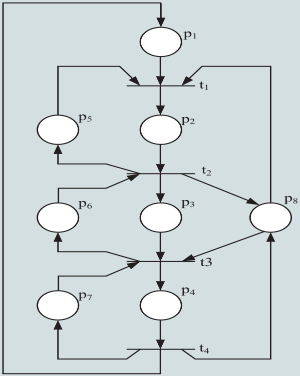
$\boldsymbol{\lambda} = [4\ 1\ 3\ 1]^T$, $\boldsymbol{m}_0 = [7.5\ 4.5\ 4\ 2\ 1.5\ 5\ 4\ 2.5]^T$, $\boldsymbol{m}_f = [7\ 5\ 5\ 1\ 1\ 4\ 5\ 3]^T$.

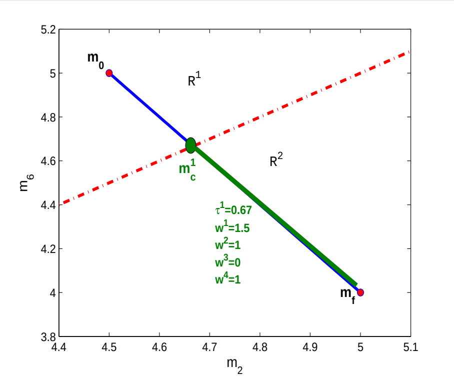




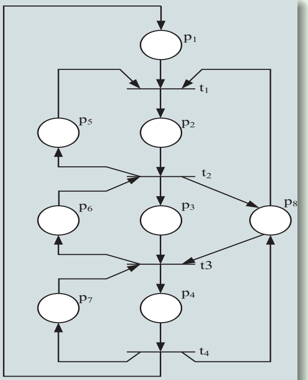
Example

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





Example



$\tau_{total} = 0.2 + 0.67 = 0.87$ t.u.

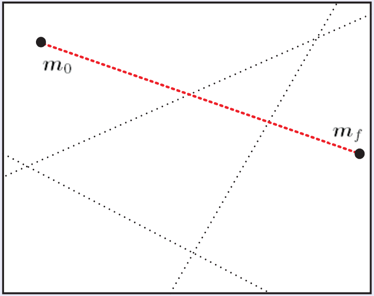
The control law:

$$\boldsymbol{u}(\tau) = \begin{cases} \begin{bmatrix} 4 \cdot m_5(\tau) - 2.49 \\ m_2(\tau) - 1.67 \\ 3 \cdot m_8(\tau) \\ m_4(\tau) - 1.67 \end{bmatrix}, \\ \quad \text{if } 0 \le \tau \le 0.2 \\ \\ \begin{bmatrix} 4 \cdot m_5(\tau) - 1.5 \\ m_2(\tau) - 1 \\ 3 \cdot m_8(\tau) \\ m_4(\tau) - 1 \end{bmatrix}, \\ \quad \text{if } 0.2 < \tau \le 0.87 \end{cases} \tag{12}$$

A Heuristics for minimum time control

In order to improve the time spent to move from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, intermediate states are introduced to the trajectory.

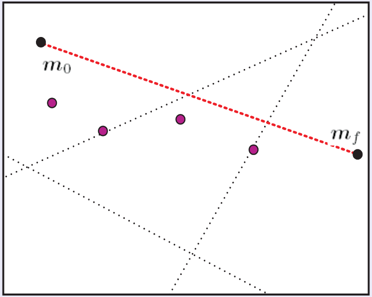
New trajectory: $\boldsymbol{m}_0 \rightarrow \boldsymbol{m}_I^1 \rightarrow \boldsymbol{m}_I^2 \rightarrow ... \rightarrow \boldsymbol{m}_I^s \rightarrow \boldsymbol{m}_f$



A Heuristics for minimum time control

In order to improve the time spent to move from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, intermediate states are introduced to the trajectory.

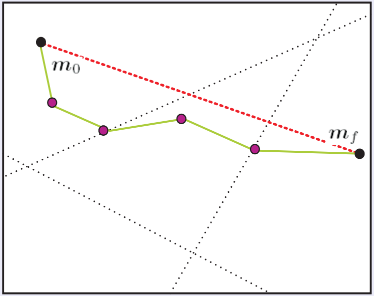
New trajectory: $\boldsymbol{m}_0 \rightarrow \boldsymbol{m}_I^1 \rightarrow \boldsymbol{m}_I^2 \rightarrow ... \rightarrow \boldsymbol{m}_I^s \rightarrow \boldsymbol{m}_f$



A Heuristics for minimum time control

In order to improve the time spent to move from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, intermediate states are introduced to the trajectory.

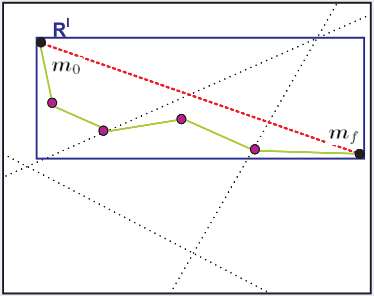
New trajectory: $\boldsymbol{m}_0 \to \boldsymbol{m}_I^1 \to \boldsymbol{m}_I^2 \to \ldots \to \boldsymbol{m}_I^s \to \boldsymbol{m}_f$



A Heuristics for minimum time control

In order to improve the time spent to move from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$, intermediate states are introduced to the trajectory.

New trajectory: $\boldsymbol{m}_0 \rightarrow \boldsymbol{m}_I^1 \rightarrow \boldsymbol{m}_I^2 \rightarrow ... \rightarrow \boldsymbol{m}_I^s \rightarrow \boldsymbol{m}_f$

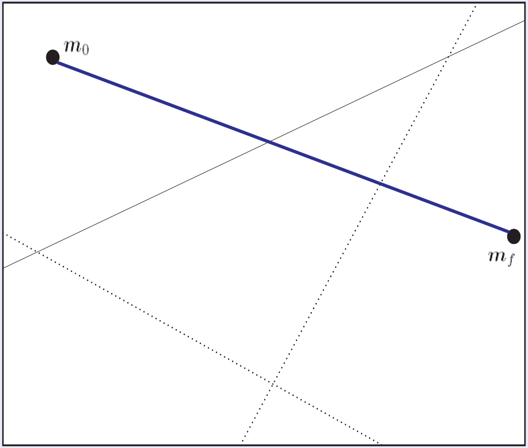


We may realize two steps to obtain PWL trajectory:

(1) Intermediate states on the borders (BLP1)
All intermediate states on the borders that the line from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ crosses are calculated by solving BLP1 once.

(2) Interior intermediate states (BLP2)
Each intermediate state is calculated by solving BLP2.

(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



Problem to be solved BLP1

$$\min \quad \sum_{k=1}^{s+1} \tau_k$$

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

$$\left(\boldsymbol{\Pi}^k - \boldsymbol{\Pi}^{k+1}\right) \cdot \boldsymbol{m}^k = 0, \quad k \in \{1, 2, .., s\} \qquad (b)$$

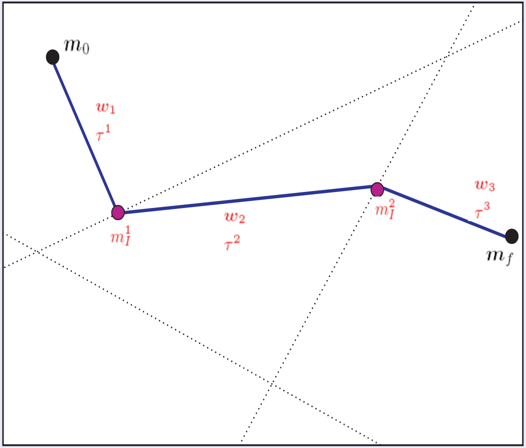
$$m_i^k \leq m_i^{k+1} \text{ if } m_{0_i} \leq m_{f_i} \quad i \in \{\ 1, 2..|P|\}, \quad k \in \{0, 1, 2, .., s\} \qquad (c)$$

$$m_i^k \geq m_i^{k+1} \text{ if } m_{0_i} \geq m_{f_i} \quad i \in \{\ 1, 2..|P|\}, \quad k \in \{0, 1, 2, .., s\} \qquad (d)$$

$$0 \leq x_j^k \leq \lambda_j \cdot \Pi_{ji}^k \cdot \min\{m_i^{k-1},\ m_i^k\} \cdot \tau_k, \quad \text{with } p_i \text{ st. } \Pi_{ji}^k \neq 0,\ j \in \{1, 2...|T|\}, \quad k \in \{1, 2..s\} \qquad (e)$$

(13)

(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



We may realize two steps to obtain PWL trajectory:

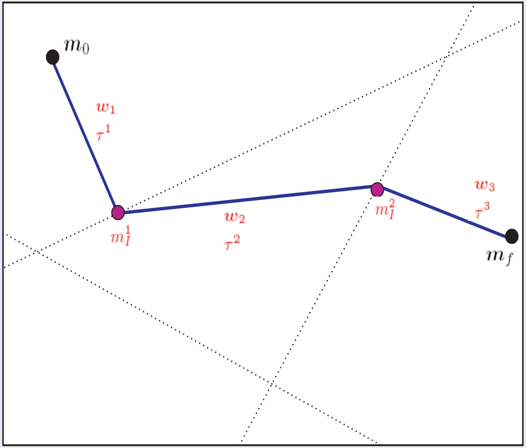
(1) Intermediate states on the borders (BLP1)
All intermediate states on the borders that the line from $\boldsymbol{m}_0$ to $\boldsymbol{m}_f$ crosses are calculated by solving BLP1 once.

(2) Interior intermediate states (BLP2)
Each intermediate state is calculated by solving BLP2.

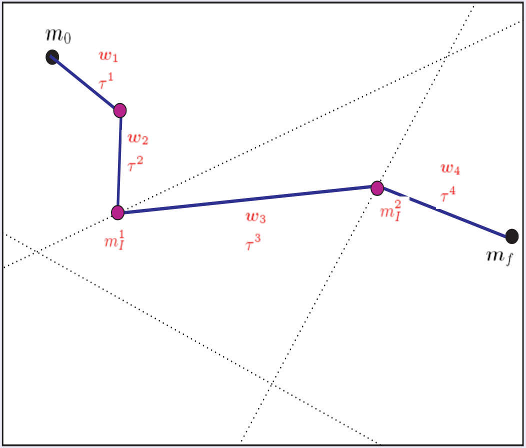
Problem to be solved BLP2

$$
\begin{aligned}
\min \quad & \tau_1 + \tau_2 \\
& \boldsymbol{m}^d = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{x}^1 \\
& \boldsymbol{m}_f = \boldsymbol{m}^d + \boldsymbol{C} \cdot \boldsymbol{x}^2, && (a) \\
& \min\{m_{f_i}, m_{0_i}\} \leq m_i^d \leq \max\{m_{f_i}, m_{0_i}\}, \\
& \qquad\qquad \forall i \in \{1, 2 ... |P|\} && (b) \\
& 0 \leq x_j^1 \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min\{m_{0_i}, m_i^d\} \cdot \tau_1, \\
& \qquad \text{with } i \text{ st. } \Pi_{ji}^k \neq 0, \;\; j \in \{1, 2 ... |T|\} \\
& 0 \leq x_j^2 \leq \lambda_j \cdot \Pi_{ji}^z \cdot \min\{m_i^d, m_{f_i}\} \cdot \tau_2, \\
& \qquad \text{with } i \text{ st. } \Pi_{ji}^k \neq 0, \;\; j \in \{1, 2 ... |T|\} && (c)
\end{aligned}
\tag{14}
$$

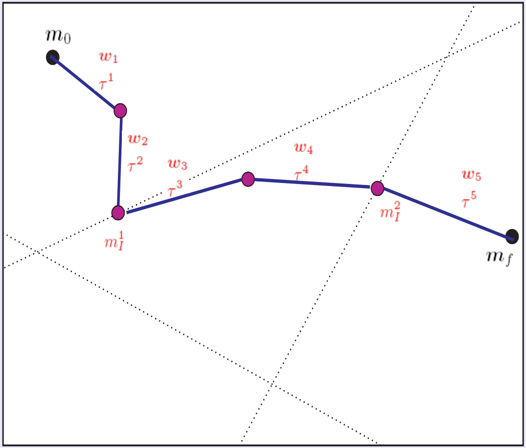
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



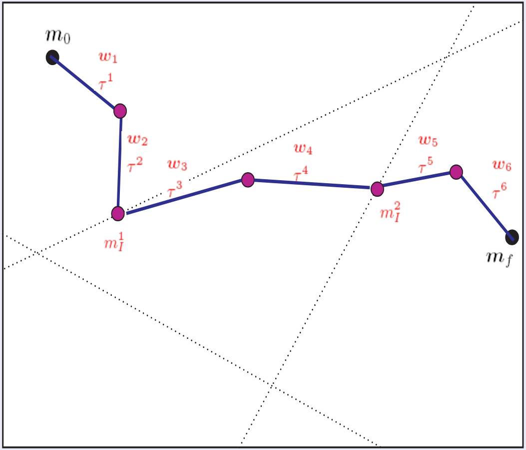
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



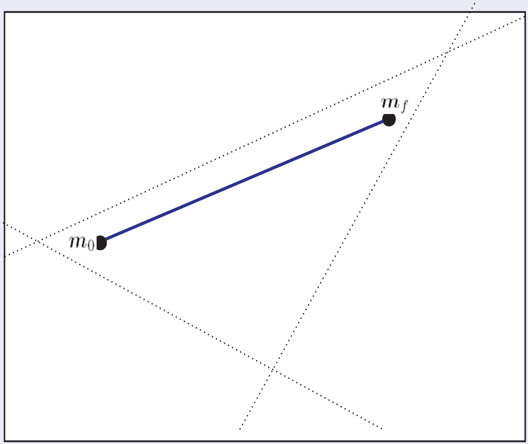
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



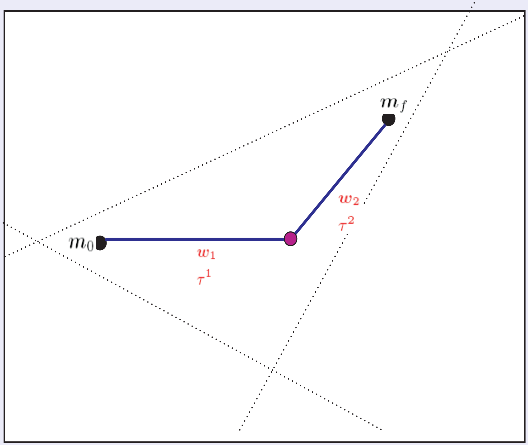
(A) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in different regions



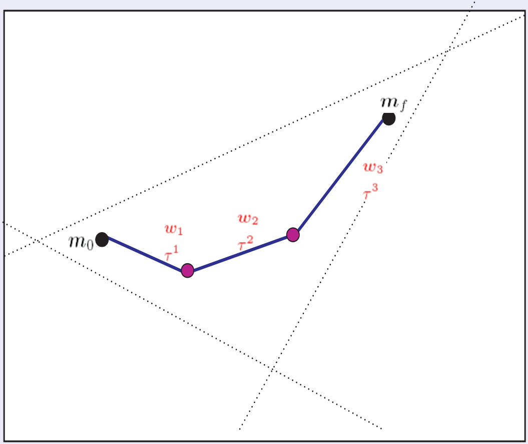
(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same regions



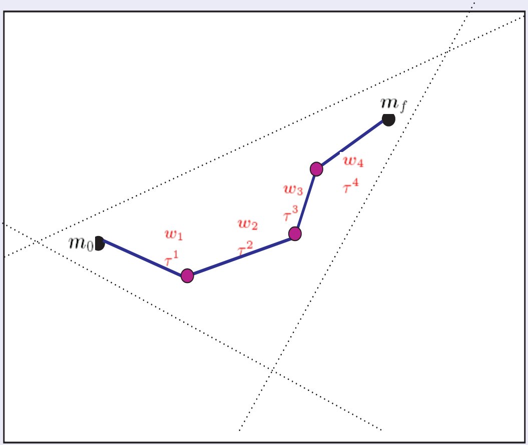
(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same regions



(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same regions



(B) $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$ are in the same regions



Piecewise Trajectory

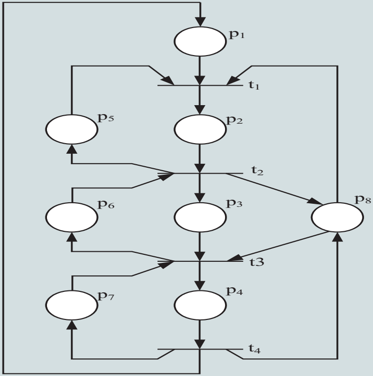
Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

Compute intermediate states on the borders that the line ℓ crosses.

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

Example

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



Piecewise Trajectory

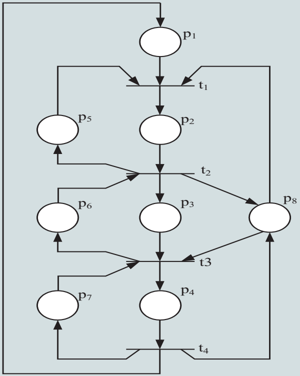
Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

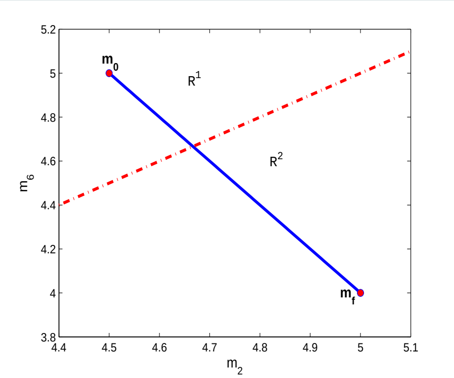
Compute intermediate states on the borders that the line ℓ crosses.

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

Example

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





Piecewise Linear Trajectory

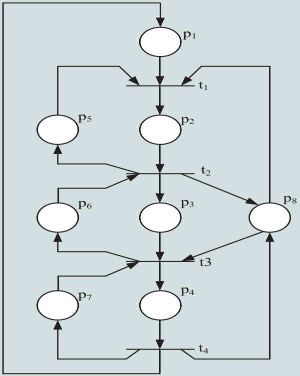
Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

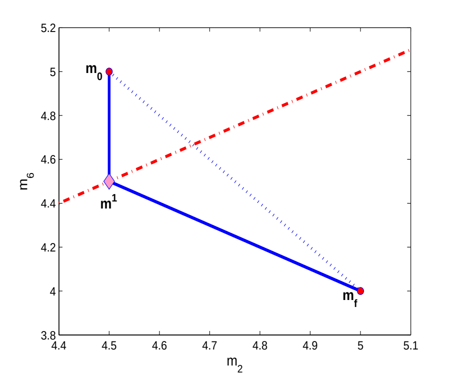
Compute intermediate states on the borders that the line ℓ crosses.

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

Example

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





Piecewise Trajectory

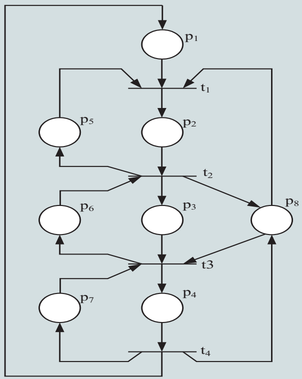
Compute the line ℓ connecting $\boldsymbol{m}_0$ and $\boldsymbol{m}_f$

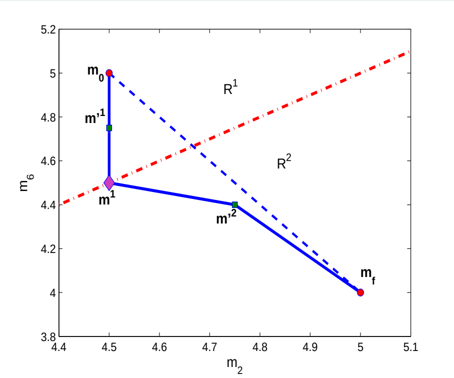
Compute intermediate states on the borders that the line ℓ crosses.

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

Example

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





Example

Table: Intermediate States

Number of int. states	0	1	3	9	13
Total duration (t.u.)	0.87	0.83	0.74	0.73	0.72
CPU time (sec.)	0.03	0.11	0.32	1.65	2.32

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

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:

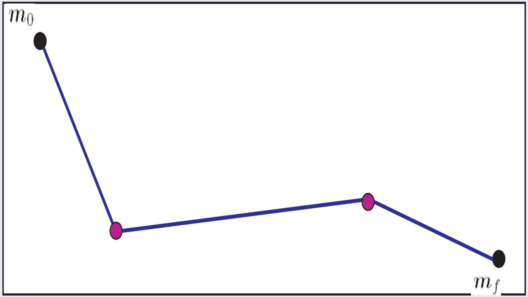
$$\begin{aligned} \boldsymbol{m}(k+1) &= \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w}(k) \\ 0 &\leq \boldsymbol{w}(k) \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot \boldsymbol{m}(k) \end{aligned} \tag{15}$$

$(\tau = k \cdot \Theta)$

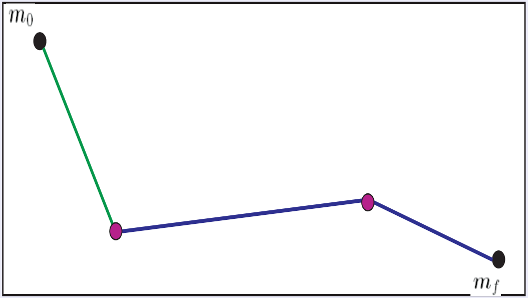
Θ should be small enough to avoid spurious states:

$$\sum_{t_j \in p^\bullet} \lambda_j \cdot \Theta < 1 \tag{16}$$

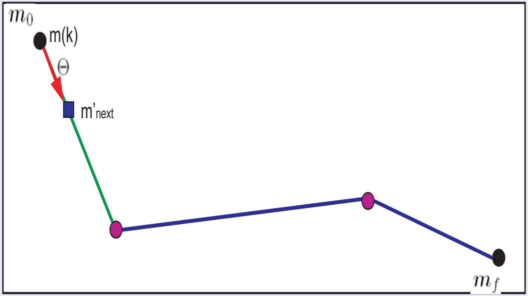
Closed Loop Control



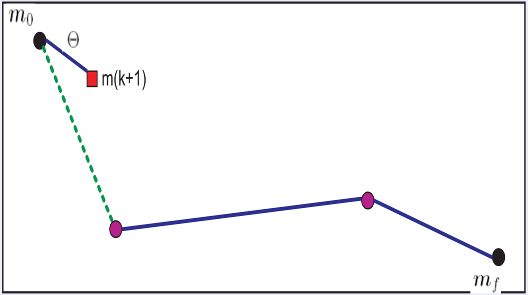
Closed Loop Control



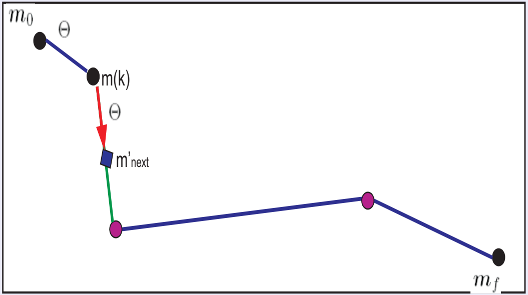
Closed Loop Control



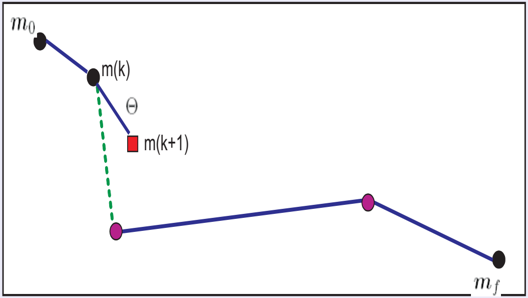
Closed Loop Control



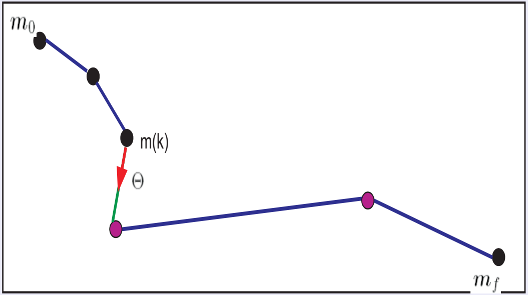
Closed Loop Control



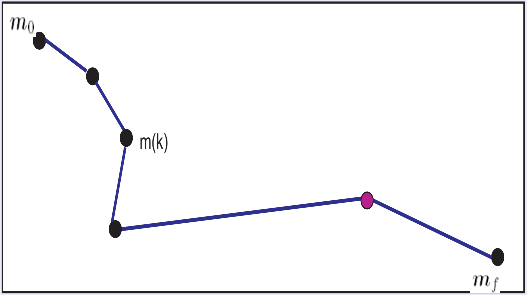
Closed Loop Control



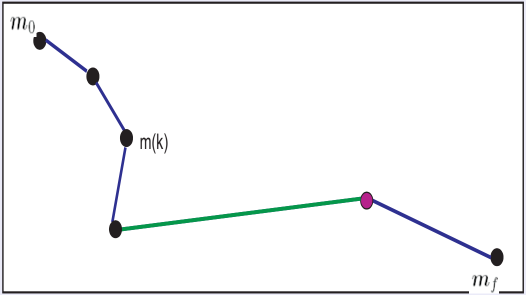
Closed Loop Control



Closed Loop Control



Closed Loop Control



Closed Loop Control

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f, \quad path, \quad s$

$\boldsymbol{m'}^0 = \boldsymbol{m}_0 \quad \boldsymbol{m'}^{s+1} = \boldsymbol{m}_f \quad \boldsymbol{m}(0) = \boldsymbol{m'}^0 \quad k = 0$

for $j = 0$ to s **do**

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

Solve the following LPP

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \boldsymbol{m'}_{next} = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w} \\ & \boldsymbol{m'}_{next} = (1-\alpha) \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{m'}^{j+1} \\ & 0 \leq \alpha \leq 1 \\ & 0 \leq \boldsymbol{w} \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot min\{\boldsymbol{m}(k), \boldsymbol{m'}_{next}\} \end{aligned} \tag{17}$$

Advance one step and obtain the new marking

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot (\boldsymbol{w} + \boldsymbol{z}) \tag{18}$$

$k = k + 1$

end while

end for

Closed Loop Control

Input: $\langle \mathcal{N}, \boldsymbol{m_0} \rangle, \quad \boldsymbol{m_f}, \quad path, \quad s$

$\boldsymbol{m'}^0 = \boldsymbol{m}_0 \;\; \boldsymbol{m'}^{s+1} = \boldsymbol{m}_f \;\; \boldsymbol{m}(0) = \boldsymbol{m'}^0 \quad k = 0$

for $j = 0$ to s **do**

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

Solve the following LPP

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \boldsymbol{m'}_{next} = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w} \\ & \boldsymbol{m'}_{next} = (1-\alpha) \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{m'}^{j+1} \\ & 0 \le \alpha \le 1 \\ & 0 \le \boldsymbol{w} \le \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot min\{\boldsymbol{m}(k), \boldsymbol{m'}_{next}\} \end{aligned} \tag{19}$$

Advance one step and obtain the new marking

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot (\boldsymbol{w} + \boldsymbol{z}) \tag{20}$$

$k = k + 1$

end while

end for

Closed Loop Control

Input: $\langle \mathcal{N}, \boldsymbol{m_0} \rangle, \quad \boldsymbol{m_f}, \quad path, \quad s$

$\boldsymbol{m'}^0 = \boldsymbol{m}_0 \quad \boldsymbol{m'}^{s+1} = \boldsymbol{m_f} \quad \boldsymbol{m}(0) = \boldsymbol{m'}^0 \quad k = 0$

for $j = 0$ to s **do**

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

Solve the following LPP

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \boldsymbol{m'}_{next} = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w} \\ & \boldsymbol{m'}_{next} = (1-\alpha) \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{m'}^{j+1} \\ & 0 \leq \alpha \leq 1 \\ & 0 \leq \boldsymbol{w} \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot min\{\boldsymbol{m}(k), \boldsymbol{m'}_{next}\} \end{aligned} \tag{21}$$

Advance one step and obtain the new marking

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot (\boldsymbol{w} + \boldsymbol{z}) \tag{22}$$

$k = k + 1$

end while

end for

Closed Loop Control

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f, \quad path, \quad s$

$\boldsymbol{m'}^0 = \boldsymbol{m}_0 \quad \boldsymbol{m'}^{s+1} = \boldsymbol{m}_f \quad \boldsymbol{m}(0) = \boldsymbol{m'}^0 \quad k = 0$

for $j = 0$ to s **do**

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

Solve the following LPP

$$\begin{aligned} \text{max} \quad & \alpha \\ \text{s.t.} \quad & \boldsymbol{m'}_{next} = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w} \\ & \boldsymbol{m'}_{next} = (1-\alpha) \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{m'}^{j+1} \\ & 0 \leq \alpha \leq 1 \\ & 0 \leq \boldsymbol{w} \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot min\{\boldsymbol{m}(k), \boldsymbol{m'}_{next}\} \end{aligned} \tag{23}$$

Advance one step and obtain the new marking

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot (\boldsymbol{w} + \boldsymbol{z}) \tag{24}$$

$k = k + 1$

end while

end for

Closed Loop Control

Input: $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle, \quad \boldsymbol{m}_f, \quad path, \quad s$

$\boldsymbol{m}'^0 = \boldsymbol{m}_0 \quad \boldsymbol{m}'^{s+1} = \boldsymbol{m}_f \quad \boldsymbol{m}(0) = \boldsymbol{m}'^0 \quad k = 0$

for $j = 0$ to s **do**

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

Solve the following LPP

$$\begin{aligned} \text{max} \quad & \alpha \\ \text{s.t.} \quad & \boldsymbol{m'}_{next} = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot \boldsymbol{w} \\ & \boldsymbol{m'}_{next} = (1-\alpha) \cdot \boldsymbol{m}(k) + \alpha \cdot \boldsymbol{m'}^{j+1} \\ & 0 \leq \alpha \leq 1 \\ & 0 \leq \boldsymbol{w} \leq \boldsymbol{\Lambda} \cdot \boldsymbol{\Pi}(\boldsymbol{m}(k)) \cdot min\{\boldsymbol{m}(k), \boldsymbol{m'}_{next}\} \end{aligned} \tag{25}$$

Advance one step and obtain the new marking

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) + \Theta \cdot \boldsymbol{C} \cdot (\boldsymbol{w} + \boldsymbol{z}) \tag{26}$$

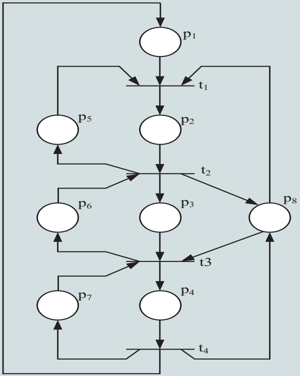
$k = k + 1$

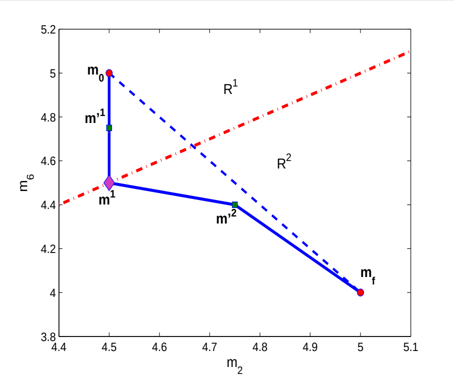
end while

end for

Example

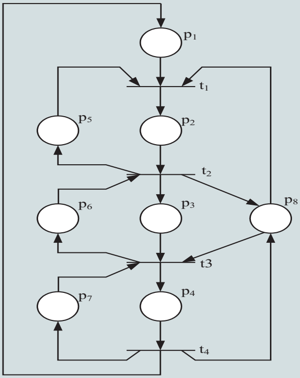
$\boldsymbol{\lambda} = [4\ 1\ 3\ 1]^T$, $\boldsymbol{m}_0 = [7.5\ 4.5\ 4\ 2\ 1.5\ 5\ 4\ 2.5]^T$, $\boldsymbol{m}_f = [7\ 5\ 5\ 1\ 1\ 4\ 5\ 3]^T$.

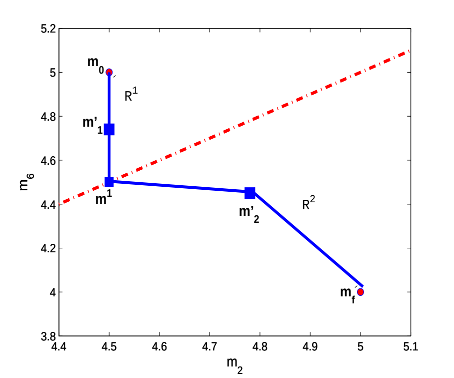




Example

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





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

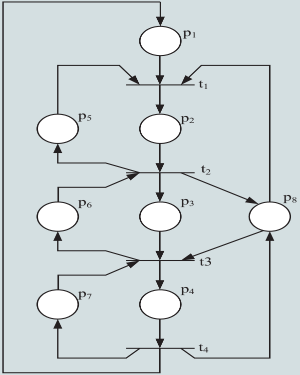
$\boldsymbol{m'}^2 = [7.5\ 4.5\ 4.5\ 1.5\ 1.5\ 4.5\ 4.5\ 3]^T$

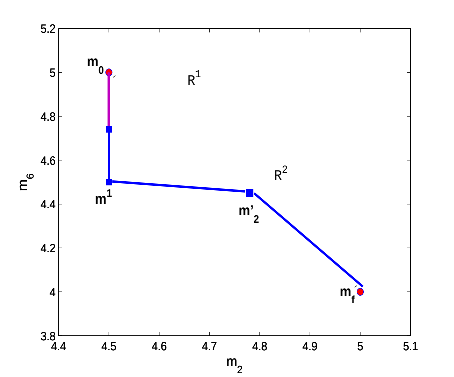
$\boldsymbol{m'}^3 = [7.45\ 4.78\ 4.55\ 1.22\ 1.22\ 4.45\ 4.78\ 3]^T$

Example

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

$\Theta = 0.01$ no noise effect



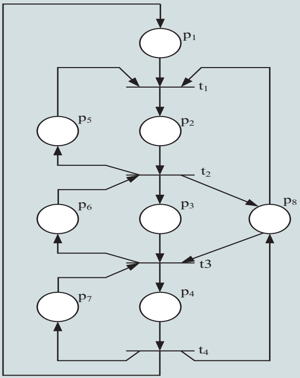


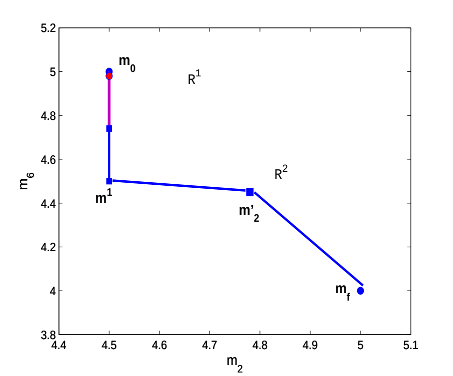
$\boldsymbol{m'}^1 = [7.5\ 4.5\ 4.26\ 1.74\ 1.5\ 4.74\ 4.26\ 2.76]^T$

$\boldsymbol{m'}^2 = [7.5\ 4.5\ 4.5\ 1.5\ 1.5\ 4.5\ 4.5\ 3]^T$

Example

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



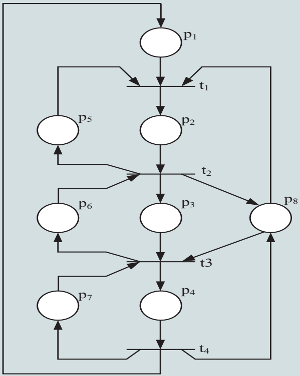


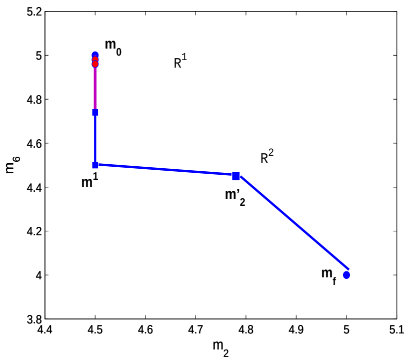
$\boldsymbol{w}(1) = [2.02\ 2.02\ 0\ 2.02]^T$

$\boldsymbol{m}(1) = [7.5\ 4.5\ 4.02\ 1.98\ 1.5\ 4.98\ 4.02\ 2.52]^T$

Example

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





$\boldsymbol{w}(2) = [1.99\ 1.99\ 0\ 1.99]^T$

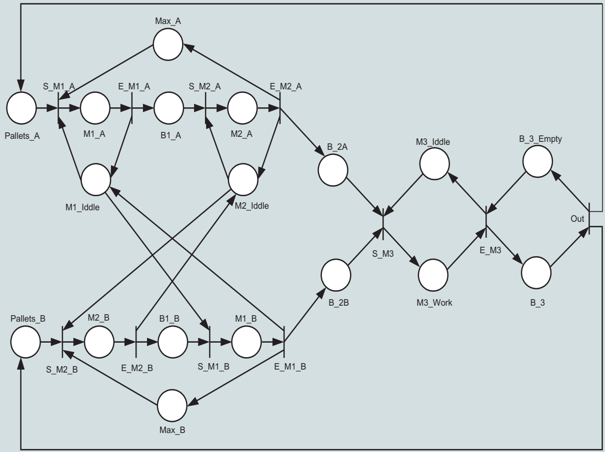
$\boldsymbol{m}(2) = [7.5\ 4.5\ 4.04\ 1.96\ 1.5\ 4.96\ 4.04\ 2.54]^T$

Total time=0.66 t.u.

Outline

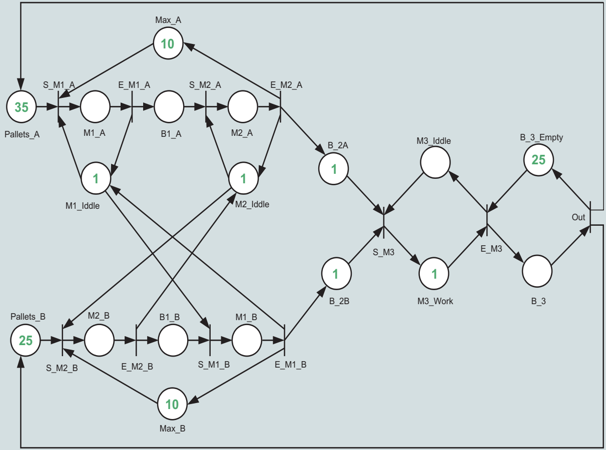
Case Study

Example



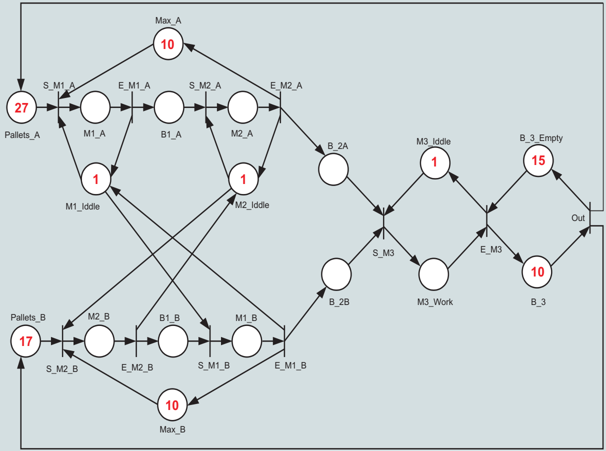
Case Study

Example



Case Study

Example



Example

place	m_0	m_f
Pallet_A	35.0100	27.0000
Pallet_B	25.0100	17.0000
B2_A	1.0100	0.0100
B2_B	1.0100	0.0100
B3_Empty	25.0100	15.0000
B_3	0.0100	10.0100

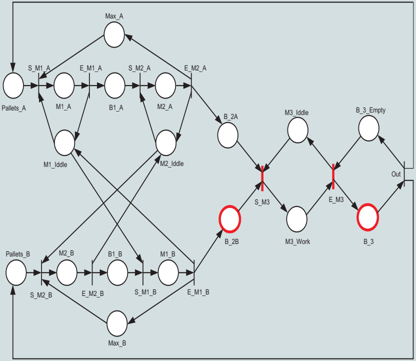
Without intermediate states: 1000 t.u.

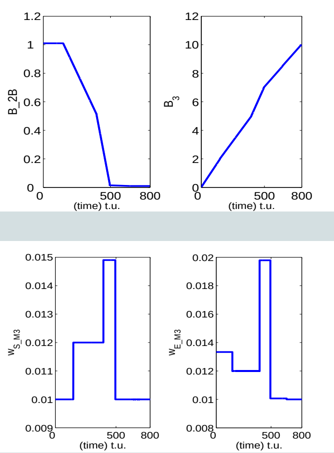
place	m_0	m'^1	m'^2	m'^3	m'^4	m_f
Pallet_A	35.0100	33.5000	31.0000	30.0000	28.5008	27.0000
Pallet_B	25.0100	23.5000	21.0000	20.0000	18.5008	17.0000
B2_A	1.0100	1.0100	0.5100	0.0149	0.0100	0.0100
B2_B	1.0100	1.0100	0.5100	0.0149	0.0100	0.0100
B3_Empty	25.0100	23.0000	20.0000	18.0148	16.5009	15.0000
B_3	0.0100	2.0100	5.0100	6.9952	8.5091	10.0100

With 4 intermediate states: 802 t.u.

Example

Closed loop control without noise effect:

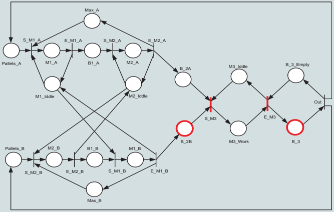


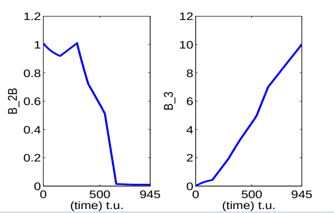


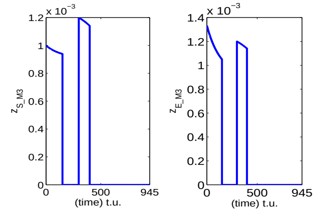
Closed Loop control without noise: 800 t.u.

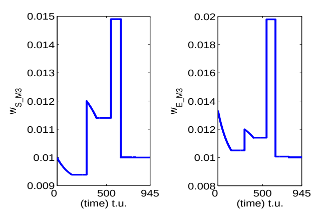
Example

Closed loop control under noise effect:









Closed Loop control with noise: 945 t.u.

Outline

Conclusion

- A control law that drives the system from initial state to target state through a linear trajectory is developed.
- Calculation of linear tarjectories 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 tarjectories 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:
  Apaydin-Ozkan, H., Julvez, J., Mahulea, C., and Silva, M. (2009). An Efficient Heuristics for Minimum Time Control of Continuous Petri nets. In 3rd IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 2009), 44–49. Zaragoza, Spain.
- The extended work is under preperation for the journal version.