Chapter 1
Continuous Petri nets: controllability and control

Jorge Júlvez, C. Renato Vázquez, Cristian Mahulea and Manuel Silva

Abstract Controllability is a property related to the capability of driving a system to any desirable (steady) state or state trajectory. As in discrete nets, continuous nets can be controlled by means of control actions applied on the transitions. For the sake of generality, the set of transitions is partitioned into *controllable* and *uncontrollable*; control actions can be applied only on controllable transitions. These actions may only reduce the flow of the uncontrolled model, because transitions (machines for example) cannot work faster than their nominal speed. Under this framework, the chapter overviews conditions for *controllability*, different control approaches for systems working under infinite server semantics and introduces a basic control method based on *consensus* for distributed systems.

1.1 Introduction and Motivation

The previous chapter has dealt with the concepts of observability and observers in the framework of continuous Petri nets. In contrast to these concepts that aim at estimating state variables that are not directly measurable, *controllability* and *control* can be seen as dual concepts pursuing to drive the system state in a desired way. In order to manipulate, i.e., control, the system behavior, control actions can be applied on transitions in order to modify their flow. Similarly to observability, a system requirement for a successful design of a control method is the system controllability, i.e., the possibility to drive the system to "any" desired state.

Example 1. As a simple introductory example, consider the net system in **??**(a) of Chapter **??** and assume that the system works under *infinite* server semantics with $\lambda_1 = \lambda_2 = 1$. Thus, the flow of transitions is $f_1 = m_1/2$ and $f_2 = m_2$. If the system is left to evolve freely from the initial marking $m_0 = [2 \quad 0]^T$, it will tend to the

The authors are with the Instituto de Investigación en Ingeniería en Aragón (I3A), University of Zaragoza, e-mail: `{julvez,cvazquez,cmahulea,silva}@unizar.es`

steady state marking $m = [4/3 \ \ 2/3]^T$ at which the flow of both transitions is the same, $f_1 = f_2 = 2/3$. Assume now that control actions can be applied on transitions in order to modify their flow. Let us just apply a constant control action $u_1 = 0.5$ on transition t_1; this means that the original flow of t_1 will be decreased by 0.5, i.e., $f_1 = (m_1/2) - 0.5$ and $f_2 = m_2$ (notice that the flow of transitions cannot be negative, i.e., $f_1 \geq 0$ and $f_2 \geq 0$ must hold). In this particular case, if such a control action is kept indefinitely, the system marking will evolve to the steady state $m = [5/3 \ \ 1/3]^T$ at which the flows of transitions are $f_1 = f_2 = 1/3$. It is important to remark that the control actions that will be introduced can only slow down the original flow of transitions, and they are dynamically upper bounded by the enabling degree of the transition, e.g., in this simple example the initial control action for transition t_1 cannot be higher than 1 given that the initial enabling degree of t_1 is 1. Notice that the constraint that the flow of transitions can only be decreased when control actions are applied, does not necessarily imply a decrease in the overall system throughput. In fact, due to the non-monotonic behavior that continuous nets can exhibit, its overall throughput can increase. This effect has already been shown in Chapter **??** (Fig. **??**), where a decrease in the firing speed of a transition can involve a higher global system throughput.

Notice that continuous Petri nets are relaxations of discrete Petri nets, but at the same time, they are continuous-state systems (in fact, they are technically hybrid systems in which the discrete state is implicit in the continuous one). That is why it is reasonable to consider at least two different approaches for the controllability and control concepts:

1) the extension of control techniques used in discrete Petri nets, such as the *supervisory-control* theory (for instance, [11–13]);

2) the application of control techniques developed for continuous-state systems.

Usually, the control objective in the first approach is to meet some safety specifications, like avoiding *forbidden* states, by means of disabling transitions at particular states. The objective of the second approach consists in driving the system, by means of a usually continuous control action, towards a desired steady state, or state trajectory (see, for instance, [8]). Regarding continuous Petri nets, most of the specific works that can be found in the literature deals with the second control approach applied to the *infinite* server semantics model.

Several works in the literature have addressed the study of controllability in the context of continuous Petri nets. For instance, the work in [1] studies controllability for linear nets, i.e., Join-Free nets, pointing out that the classical rank condition is not sufficient (detailed in subsection 1.3.2). In [14] controllability was studied for Join-Free continuous nets from a different perspective, by characterizing the set of markings that can be reached and maintained. Unfortunately, those results are difficult to extend to general subclasses of nets, where the existence of several regions makes the general reachability problem intractable.

It is important to remark that enforcing a desired target marking in a continuous Petri net is analogous to reaching an average marking in the original discrete model (assuming that the continuous model approximates correctly the discrete one), which may be interesting in several kinds of systems. This idea has been illustrated by different authors. For example, the work in [2] proposes a methodology for the control of open and closed manufacturing lines. The control actions consist in modifying the maximal firing speeds of the controlled transitions. It was also illustrated how the control law can be applied to the original discrete Petri net model (a T-timed model with constant firing delays). This approach has been used in [18] and [17] as well, in the same context of manufacturing lines. A related approach was presented in [25] for a stock-level control problem of an automotive assembly line system originally modeled as a stochastically timed discrete Petri net [10]. The resulting scheme allows to control the average value of the marking at the places that represent the stock-level, by means of applying additional delays to the controllable transitions.

The rest of the chapter is organized as follows: Section 1.2 introduces control actions and the way they modify the flow of transitions in systems working under infinite server semantics. In section 1.3, the controllability property is discussed and some results are extracted for the case in which all transitions are controllable, and the case in which there are some non-controllable transitions. Section 1.4 describes some control methods for systems in which all transitions are controllable and sketches a couple of methods that can be applied when some transitions are not controllable. Finally, section 1.5 discusses how a distributed control approach can be designed on a net system composed of several subsystems connected by buffers.

1.2 Control actions under infinite server semantics

Like in discrete Petri nets, control actions are applied on the transitions. These actions can only consist in the reduction of the flow, because transitions (machines for example) should not work faster than their nominal speed. The set of transitions T is partitioned into two sets T_c and T_{nc}, where T_c is the set of controllable transitions and T_{nc} is the set of uncontrollable transitions. The control vector $u \in \mathbb{R}^{|T|}$ is defined s.t. u_i represents the control action on t_i. In the following *infinite* server semantics will be assumed. Since u_i represents a reduction of the flow, then the following inequality must hold $0 \leq u_i \leq \lambda_i \cdot enab(t_i, m)$. The behavior of a *forced* (or controlled) continuous Petri net can be described by the state equation:

$$\begin{array}{c} \dot{m} = C\Lambda\Pi(m)m - Cu \\ \text{s.t. } 0 \leq u \leq \Lambda\Pi(m)m \text{ and } \forall t_i \in T_{nc}, u_i = 0. \end{array} \tag{1.1}$$

where matrix Π is the configuration matrix defined in Chapter **??**:

$$\Pi_k[t,p] = \begin{cases} \frac{1}{Pre[p,t]}, & \text{if } (p,t) \in \mathscr{C}_k \\ 0, & \text{otherwise} \end{cases} \tag{1.2}$$

where $\mathscr{C}_k$ denotes a configuration as defined in Chapter **??**. Notice that the slack variables introduced in section 5 of Chapter **??** play a similar role to the one of the control actions. There are, however, important differences, in that case slack variables are associated to places and only the steady state was optimized.

1.3 Controllability

Among the many possible control objectives, we will focus on driving the system, by applying a control law, towards a desired steady state, i.e., a *set-point* control problem, frequently addressed in continuous-state systems. This control objective is related to the classical controllability concept, according to which a system is controllable if for any two states x_1, x_2 of the state space it is possible to transfer the system from x_1 to x_2 in finite time (see, for instance, [8]).

Marking conservation laws frequently exist in most Petri nets with practical significance. Such conservation laws imply that timed continuous Petri net (TCPN) systems are frequently not controllable according to the classical controllability concept [21, 22]. More precisely, if y is a *P-flow* then any reachable marking m must fulfill $y^T m = y^T m_0$, defining thus a *state invariant*. Nevertheless, the study of controllability "over" this invariant is particularly interesting. This set is formally defined as $Class(m_0) = \{m \in \mathbb{R}^{|P|}_{\geq 0} | B_y^T m = B_y^T m_0\}$, where B_y is a basis of P-flows, i.e., $B_y^T C = 0$. For a general TCPN system, every reachable marking belongs to $Class(m_0)$.

Another important issue that must be taken into account in TCPN systems is the nonnegativeness and boundedness of the input, i.e., $0 \leq u \leq \Lambda \Pi(m)m$. An appropriate local controllability concept, once these issues are considered, is [24]:

Definition 1 *The TCPN system* $\langle N, \lambda, m_0 \rangle$ is controllable with bounded input *(BIC)* over $S \subseteq Class(m_0)$ *if for any two markings* $m_1, m_2 \in S$ *there exists an input* u *transfering the system from* m_1 *to* m_2 *in finite or infinite time, and it is suitably bounded, i.e.,* $0 \leq u \leq \Lambda \Pi(m)m$*, and* $\forall t_i \in T_{nc}$ $u_i = 0$ *along the marking trajectory.*

1.3.1 Controllability when all *the transitions are controllable*

An interesting fact is that when all the transitions are controllable, the controllability of TCPNs, depends exclusively on the structure of the net. Let us give some intuition about this by rewriting the state equation as:

$$\dot{m} = C \cdot w \tag{1.3}$$

where the innovation vector $w = \Lambda \Pi(m)m - u$ can be seen as an auxiliary input. The constraints for u are transformed into $0 \leq w \leq \Lambda \Pi(m)m$. In this way, given a marking $m_1 \in Class(m_0)$, if $\exists \sigma \geq 0$ such that $C\sigma = (m_1 - m_0)$ then m_1 is reachable from m_0. This can be achieved by setting $w = \alpha\sigma$ (with a small enough $\alpha > 0$), so the field vector results $\dot{m} = C\alpha\sigma = \alpha(m_1 - m_0)$ which implies that the system will evolve towards m_1 describing a straight trajectory (assuming that the required transitions can be fired from this marking, what always happens if m is a relative interior point of $Class(m_0)$).

Example 2. Consider, for instance, the TCPN of Fig. 1(a) and the markings $m_0 = [2\ 3\ 1\ 1]^T$, $m_1 = [1\ 3\ 2\ 1]^T$ and $m_2 = [2\ 1\ 1\ 3]^T$. Given that this system has 2 P-semiflows (involving $\{p_1, p_3\}$ and $\{p_2, p_4\}$ respectively), the marking of two places is sufficient to represent the whole state. For this system $\exists\sigma \geq 0$ such that $C\sigma = (m_1 - m_0)$, but $\nexists\sigma \geq 0$ such that $C\sigma = (m_2 - m_0)$, so, m_1 is reachable but m_2 is not. The shadowed area in Fig. 1(a) corresponds to the set of reachable markings, note that it is the convex cone defined by vectors c'_1 and c'_2, which represent the columns of C (here restricted to p_1 and p_2).

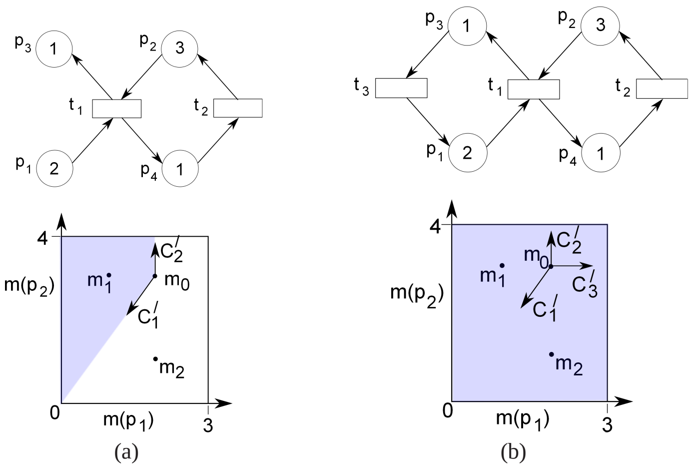

Fig. 1.1 Two TCPN systems with identical P-flows. The shadowed areas correspond to the sets of reachable markings. Only the system (b) is consistent and controllable over $Class(m_0)$.

A full characterization of controllability [24] can be obtained from this structural reachability reasoning:

Proposition 2 *Let $\langle N, \lambda, m_0\rangle$ be a TCPN system in which all the transitions are controllable. The system $\langle N, \lambda, m_0\rangle$ is BIC over the interior of $Class(m_0)$ iff the net is* consistent. *Furthermore, the controllability is extended to the whole $Class(m_0)$ iff (additionally to consistency) there exist no* empty siphon *at any marking in $Class(m_0)$.*

It is important to remark that controllability does not depend on the timing λ. In fact, the key condition here is consistency, i.e., $\exists x > 0$ such that $C \cdot x = 0$. Remember that a reachable marking $m \geq 0$ fulfills $m = m_0 + C \cdot \sigma$ with $\sigma \geq 0$, which implies $B_y^T m = B_y^T m_0$ (equivalently, $m \in Class(m_0)$). In the opposite sense, if the net is consistent then $\forall m \geq 0$ s.t. $B_y^T m = B_y^T m_0$ (i.e., $m \in Class(m_0)$) it exists $\sigma \geq 0$ s.t. $m = m_0 + C \cdot \sigma$, thus m is reachable (assuming σ is fireable). A very informal and intuitive explanation is that consistency permits movements of marking in any direction inside the reachability space (see Fig. 1.1(b)), i.e., if there exists σ such that $m_1 = m_0 + C \cdot \sigma$, under consistency any $\sigma' = \sigma + k \cdot x \geq 0$, permits the reachability of m_1.

Let us consider again the TCPN system of Fig. 1.1(a). Given that the net is not consistent, it can be deduced by Proposition 2 that it is not controllable over $Class(m_0)$. Let us now consider the system of Fig. 1.1(b). In this case, due to the consistency of the net, it holds that the vector $(m - m_0)$ is in the convex cone defined by the vectors c'_1, c'_2 and c'_3 for any marking $m \in Class(m_0)$. Therefore m is reachable from m_0. Furthermore, since at the border markings of $Class(m_0)$ there are not unmarked siphons then, according to Proposition 2, the system is *BIC* over $Class(m_0)$.

1.3.2 *Controllability when* some *transitions are uncontrollable*

If a TCPN contains uncontrollable transitions it becomes not controllable over $Class(m_0)$, even if the net is consistent. Thus, the concept of controllability must be constrained to a smaller set of markings. The work in [14] studies this idea by defining a set named Controllability Space (*CS*) for Join-Free nets, over which the system is controllable. Unfortunately, this set depends on the marking, and therefore, its characterization for general subclasses of nets is difficult. The existence of several regions makes the general reachability problem intractable. For practical reasons, the controllability was studied in [24] over sets of *equilibrium markings* : $m^q \in Class(m_0)$ is an equilibrium marking if $\exists u^q$ suitable such that $C(\Lambda \Pi(m^q) m^q - u^q) = 0$, i.e., there exists a control action that keeps the system marking constant at m^q. They represent the *possible stationary operating points* of the original discrete system. These markings are particularly interesting, since controllers are frequently designed in order to drive the system towards a desired *stationary operating point*.

Given that inside each region $\mathscr{R}_i$ the state equation is linear ($\Pi(m)$ is constant), it becomes convenient to study, in a first step, the controllability over equilibrium markings in each region and later over the union of them. This approach is supported by the following proposition:

Proposition 3 *Let $\langle N, \lambda, m_0 \rangle$ be a TCPN system. Consider some equilibrium sets S_1, S_2,..., S_j related to different regions $\mathscr{R}_1$, $\mathscr{R}_2$,..., $\mathscr{R}_j$. If the system is BIC (in finite time) over each one and their union $\bigcup_{i=1}^{j} S_i$ is connected, the system is BIC over the union.*

The connectivity of the set of all the equilibrium markings in $Class(m_0)$ has not been demonstrated for the general case. Nevertheless, in every studied system such property holds.

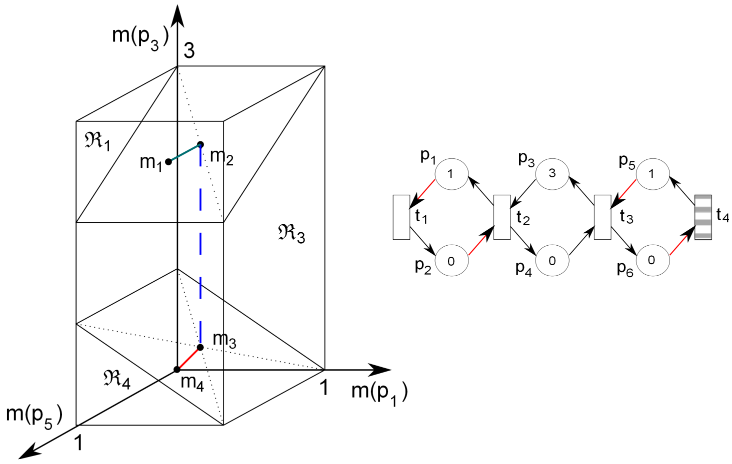

Fig. 1.2 TCPN system with its $\mathbb{E}$. Transition t_4 is the only controllable one. There are four possible configurations: $\mathscr{C}_1 = \{(p_2,t_2), (p_4,t_3)\}$, $\mathscr{C}_2 = \{(p_3,t_2), (p_4,t_3)\}$, $\mathscr{C}_3 = \{(p_2,t_2), (p_5,t_3)\}$ and $\mathscr{C}_4 = \{(p_3,t_2), (p_5,t_3)\}$, however, $\mathscr{C}_2$ cannot occur from the given m_0 because p_3 and p_4 cannot concurrently constrain t_2 and t_3, respectively. Equilibrium sets depend on the timing, but regions do not.

Example 3. Let us consider as an example the timed continuous marked graph of Fig. 1.2 with $T_c = \{t_4\}$ and $\lambda = [1\,1\,1\,2]^T$. According to the net structure, there are four possible configurations, but given the initial marking, one of them cannot occur. The polytope in Fig. 1.2 represents the $Class\{m_0\}$. Since the system has 3 P-semiflows, the marking at $\{p_1, p_3, p_5\}$ is enough to represent the whole state. This is divided into the regions $\mathscr{R}_1$, $\mathscr{R}_3$ and $\mathscr{R}_4$, related to the feasible configurations. The segments $E_1 = [m_1, m_2]$, $E_3 = [m_2, m_3]$ and $E_4 = [m_3, m_4]$ are the sets of equilibrium markings in regions $\mathscr{R}_1$, $\mathscr{R}_3$ and $\mathscr{R}_4$, respectively. Since the union of E_1, E_3 and E_4 is connected, if the system was *BIC* over each E_i (this will be explored in a forthcoming example) then, according to Proposition 3, the system would be *BIC* over $E_1 \cup E_3 \cup E_4$. For instance, the system could be driven from m_3 to m_1 and in the opposite sense.

Notice that the behavior of the TCPN system is linear and time-invariant in a given region $\mathscr{R}_i$, then some of the classical results in control theory can be used for its analysis. Null-controllability (controllability around the origin) of this kind of systems with input constraints was studied in [7]. Recalling from there, if a linear system $\dot{x} = Ax + Bu$, with input constraint $u \in \Omega$ (called the set of admissible inputs), is controllable then the controllability matrix $Contr(A,B) = [BAB \ldots A^{n-1}B]$ has full rank (equivalently, $\forall x_1, x_2$: $\exists z$ s.t. $(x_2 - x_1) = Contr(A,B) \cdot z$). Moreover, if 0 is in the interior of Ω then the previous rank condition is also sufficient for null-controllability. Otherwise, if there are inputs that can be only settled as positive (or negative) then the controllability depends also on the eigenstructure of the state matrix. These results can be adapted to TCPNs. For this, the state equation of a TCPN is firstly transformed in order to represent the behavior around an equilibrium marking m^q, i.e., the evolution of $\Delta m = m - m^q$. As a consequence, some transformed inputs $\Delta u = (u - u^q)$ can be settled only as nonnegative while others can be settled as either positive or negative. The set of transitions related to this last kind of inputs is denoted as $T^i_{cf} \subseteq T_c$. Let us denote as E^*_i the set of all equilibrium markings in a region $\mathscr{R}_i$ s.t. $\Delta u[T^i_{cf}]$ can be settled as either positive or negative (equivalently, $[\Lambda \Pi_i m^q]_j > u^q_j > 0$ for all $t_j \in T^i_{cf}$). In this way, it can be proved that if a TCPN is controllable over a set E^*_i then $\forall m_2, m_1 \in E^*_i$: $\exists z$ s.t. $(m_2 - m_1) = Contr((C\Lambda\Pi_i), C[T_c]) \cdot z$. This condition is only necessary, as already pointed out in [1], because the existence of input constraints. Furthermore, a system is controllable (in finite time) over E^*_i if $\forall m_2, m_1 \in E^*_i$: $\exists z$ s.t. $(m_2 - m_1) = Contr((C\Lambda\Pi_i), C[T^i_{cf}]) \cdot z$. This sufficient condition is also necessary if $T^i_{cf} = T_c$ (but not if $T^i_{cf} \subset T_c$). Note that now the controllability depends not only on the structure of the net, but also on the timing [24].

As an example, let us consider the region $\mathscr{R}_3$ in the system of Fig. 1.2, where $T^3_{cf} = \{t_4\}$. Given that $T^3_{cf} = T_c$ then the span condition introduced above is sufficient and necessary for controllability. In this case, it can be verified that the system is *BIC* over $E^*_3 = E_3$. Consider now the same system but with $\lambda_4 = 1$ instead of $\lambda_4 = 2$. In this case, $T^3_{cf} = \emptyset$ (this set depends on the timing), then we cannot use the same sufficient condition. Nevertheless, it is still fulfilled that $\forall m_2, m_1 \in E^*_3 : \exists z$ s.t. $(m_2 - m_1) = Contr((C\Lambda\Pi_3), C[T_c]) \cdot z$. Therefore, the controllability matrices do not provide enough information for deciding whether the system is *BIC* or not over E^*_3. By using other results from [24], it can be proved that the system is not *BIC* with $\lambda_4 = 1$. This implies that controllability is a timing-dependent property.

1.4 Control techniques under infinite server semantics

This section describes some few techniques proposed in the literature for the control of TCPNs when all transitions are controllable. Similarly to the *set-point* control problem in state-continuous systems, the control objective here consists in driving the system towards a desired target marking here denoted as m_d. This desired marking can be selected, in a preliminarily planning stage, according to some optimality criterion [23], e.g., maximizing the flow. Most of the work done on this issue is devoted to centralized dynamic control assuming that *all* the transitions are controllable. We will first present those control techniques that require all the transitions to be controllable, then a basic comparison of such techniques will be performed, and finally a couple of approaches where uncontrollable transitions are allowed will be presented.

1.4.1 Control for a piecewise-straight marking trajectory

This subsection introduces a control approach that aims at reaching a given target marking by following a piecewise-straight trajectory. A similar approach was studied in [15] for Join-Free nets where the tracking control problem of a mixed ramp-step reference signal was explored, and later extended to general Petri nets in [16]. In such a work, a high & low gain proportional controller is synthesized, while a ramp-step reference trajectory, as a sort of *path-planning* problem at a higher level, is computed. We will discuss the more simple synthesis procedure introduced in [3].

Let us consider the line l connecting m_0 and m_d, and the markings in the intersection of l with the region's borders, denoted as m_c^1, m_c^2, ..., m_c^n. Define $m_c^0 = m_0$ and $m_c^{n+1} = m_d$. Then, $\forall k \in \{0, n\}$ compute τ_k by solving the linear programming problem (LPP):

$$\begin{array}{ll} min\, \tau_k & \\ s.t.: & m_c^{i+1} = m_c^i + C \cdot x \\ & 0 \leq x_j \leq \lambda_j \Pi^z_{ji} min\{m^i_{c,i}, m^{i+1}_{c,i}\}\tau_k \\ & \forall j \in \{1, ..., |T|\} \text{ where i satisfies } \Pi^z_{ji} \neq 0 \end{array} \tag{1.4}$$

where the first constraint is the fundamental state equation and the second constraint ensures the applicability of the input actions. This way, the control law to be applied is $w = x/\tau_k$ (the model is represented as in (1.3)), when the system is between the markings m_c^k and m_c^{k+1}. The time required for reaching the desired marking is given by $\tau_f = \sum_{k=0}^n \tau_k$. *Feasibility* and *convergence* to m_d were proved in [3].

If one aims at obtaining faster trajectories, intermediate states, not necessarily on the line connecting the initial and the target marking, can be introduced [16]. According to [3], they can be computed by means of a *bilinear* programming problem (BPP). The idea is to currently compute the intermediate markings m_c^k, on the borders of the regions that minimizes the total time $\tau_f = \sum_{k=0}^n \tau_k$ with some additional monotonicity constraints. Finally, the same algorithm can be adapted in order to recursively compute intermediate markings in the interior of the regions, obtaining thus faster trajectories.

1.4.2 Model Predictive Control

Within the Model Predictive Control (MPC) framework, two main solutions can be considered based on the *implicit* and *explicit* methods (see, for instance, [6]). The evolution of the timed continuous Petri net model (1.3), in *discrete-time*, can be represented by the difference equation: $m(k+1) = m(k) + \Theta \cdot C \cdot w(k)$, subject to the constraints $0 \leq w(k) \leq f(k)$ with $f(k)$ being the flow without control, which is equivalent to $G \cdot [w^T(k), m^T(k)]^T \leq 0$, for a particular matrix G. The sampling Θ must be chosen small enough in order to avoid spurious markings, in particular, for ensuring the positiveness of the markings. For that, the following condition is required to be fulfilled $\forall\, p \in P \,:\, \sum_{t_j \in p^\bullet} \lambda_j \Theta < 1$.

A MPC control scheme can be derived [20] by using this representation of the continuous Petri net. The considered goal is to drive the system towards a desired marking m_d, while minimizing the quadratic performance index

$$\begin{aligned} J(m(k),N) = {} & (m(k+N)-m_d)'Z(m(k+N)-m_d) \\ & +\textstyle\sum_{j=0}^{N-1}[(m(k+j)-m_d)'Q(m(k+j)-m_d) \\ & \quad +(w(k+j)-w_d)'R(w(k+j)-w_d)] \end{aligned}$$

where Z, Q and R are positive definite matrices and N is a given time horizon. This leads to the following optimization problem that needs to be solved in each time step:

$$\begin{array}{ll} \min\ J(m(k),N) & \\ \text{s.t.} : \forall j \in \{0,...,N-1\}, & m(k+j+1) = m(k+j) + \Theta \cdot C \cdot w(k+j) \\ & G \cdot \begin{bmatrix} w(k+j) \\ m(k+j) \end{bmatrix} \leq 0 \\ & w(k+j) \geq 0 \end{array} \tag{1.5}$$

Let us show that, in general, the standard MPC approach does not guarantee convergence [20].

Example 4. Consider the net system in Fig. 1.3 with $\lambda = [1\ 5]^T$. Let $\Theta = 0.1$, $m_d = [0\ 1]^T$ and $w_d = [0\ 0]^T$. Moreover, let $Q = Z = R = I$ and $N = 1$.

Fig. 1.4 shows the marking evolution of the system controlled with the MPC policy. It can be seen that the desired marking is not reached. Observe that to obtain m_d, only t_1 should fire. Given that the timing horizon is too short and $\lambda_2 = 5 >> \lambda_1 = 1$, the optimality of (1.5) implies that it is better to fire at the beginning "a little" t_2 so that m_1 approaches the desired final value $m_{f,1} = 0$. However, once t_2 has fired, m_d cannot be reached because there is not enough marking in p_1 to be transferred to p_2. ■

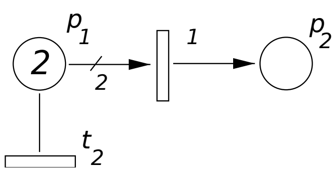

Fig. 1.3 Example of an unstable TCPN system with basic MPC scheme.

The work in [20] shows that the standard techniques used for ensuring converge in linear/hybrid systems (i.e., terminal constraints or terminal cost) cannot be applied in continuous nets if the desired marking has zero components. However, several approaches can be considered to improve convergence. Let us discuss one of

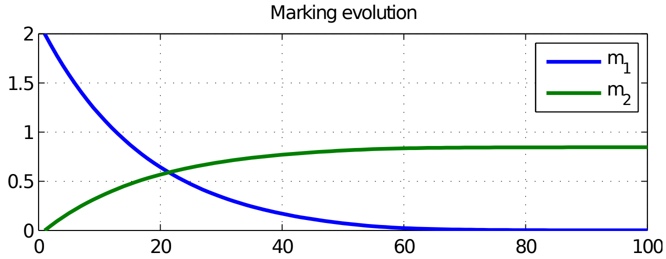

Fig. 1.4 Marking evolution of the TCPN system in Fig. 1.3.

them that consists in constraining the system state at time $k+N$ to belong to the straight line $m(k)$ — m_d. Roughly, this is equivalent to add a terminal constraint of the form:

$$\begin{cases} m(k+N) = m_k + \alpha \cdot (m_d - m(k)) \\ 0 \leq \alpha \leq 1 \end{cases} \tag{1.6}$$

to the optimization problem (1.5), where α is a new decision variable. As stated in the following proposition, the inclusion of this constraint guarantees asymptotically stability.

Proposition 4 *Consider a TCPN system with m_0 and m_d the initial and target markings, respectively, being $m_0 > 0$ and m_d reachable from m_0. Assume that the system is controlled using MPC with a terminal constraint of the form* (1.6) *and prediction horizon $N = 1$. Then, the closed-loop system is asymptotically stable.*

Example 5. Let us exemplify this result through the TCPN in Fig. 1.5. Assume $m_0 = [1\ 0.1]^T > 0$, $m_d = [0\ 0]^T$, $w_d = [0\ 0\ 0]^T$, $\lambda = [1\ 1\ 1]^T$, $Z = R = I$, $Q = [1\ 0; 0\ 100]$ and $\Theta = 0.1$.

Fig. 1.6 shows the marking evolution after applying MPC with the terminal constraint $m(k+N) = \alpha \cdot m_d + (1-\alpha) \cdot m(k)$, for $N = 1$ and $N = 2$ respectively. It can be observed that if $N = 1$, m_d is reached, but if $N = 2$, m_d is not reached. Notice that $m_d = [0\ 0]^T$ is on the boundary of the feasible states since in a Petri net, $m(k) \geq 0$ for all reachability markings.

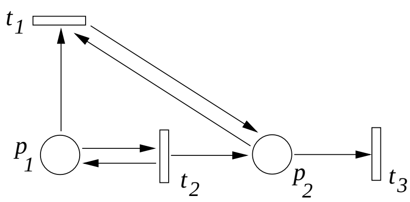

Fig. 1.5 A TCPN showing that the terminal equality constraint may not ensure stability if $N > 1$.

An alternative MPC approach for this problem is the so-called *explicit* solution [6], where the set of all states that are controllable is split into polytopes. In each polytope the control command is defined as a piecewise affine function of the state. The closed-loop *stability* is guaranteed with this approach. On the contrary, when either the order of the system or the length of the prediction horizon are not small, the *complexity* of the explicit controller becomes computationally prohibitive. Furthermore, the computation of the polytopes sometimes is unfeasible.

1.4.3 ON-OFF Control

If the control problem is constrained to particular net subclasses, stronger results may be obtained. For instance, for *structurally persistent continuous Petri nets*, i.e., net systems where the enabling of any transition t_j cannot decrease by the firing of any other transition $t_i \neq t_j$ (in continuous nets this corresponds to choice-free nets), the minimum-time control problem has been solved [28].

The solution to this problem can be obtained as follows. First, a minimal firing count vector σ s.t. $m_d = m_0 + C\sigma$ is computed (σ is minimal if for any T-semiflow x, $||x|| \not\subseteq ||\sigma||$, where $||\cdot||$ stands for the support of a vector). Later, the control law is defined, for each transition t_j, as:

$$u[t_j] = \begin{cases} 0 & \text{if } \int_0^{\tau^-} w[t_j] d\tau < \sigma[t_j] \\ f[t_j] & \text{if } \int_0^{\tau^-} w[t_j] d\tau = \sigma[t_j] \end{cases}$$

This means that if t_j has not been fired an amount of $\sigma[t_j]$, then t_j is completely ON. Otherwise, t_j is completely OFF (it is blocked).

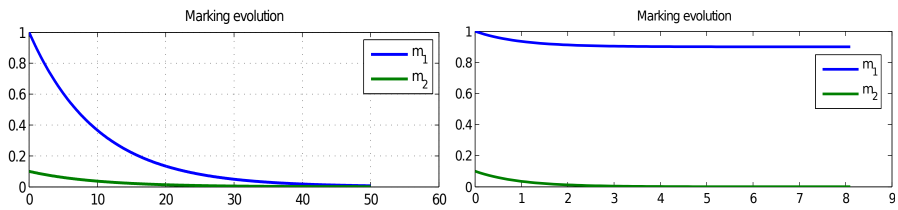

Fig. 1.6 Marking evolution of the net in Fig. 1.5 with $N = 1$ (left) and $N = 2$ (right).

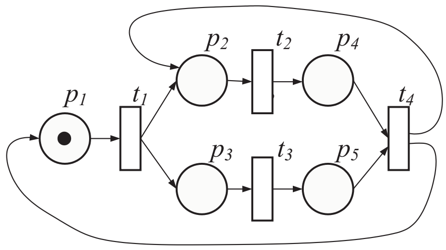

Fig. 1.7 Structurally persistent Petri net system

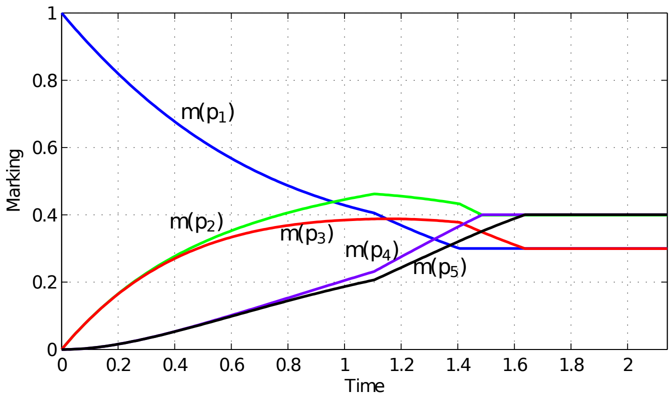

Fig. 1.8 Marking trajectories of the net system in Fig. 1.7 under ON-OFF control

Example 6. Let us consider the structurally persistent net in Fig. 1.7 to show the ON-OFF control. Let us assume that the initial marking is $m_0 = [1\ 0\ 0\ 0\ 0]^T$, the target marking is $m_d = [0.3\ 0.4\ 0.3\ 0.4\ 0.4]^T$ and that $\lambda = [1\ 1\ 1\ 1]^T$. Fig. 1.8 shows the marking trajectory after the application of the control. It can be appreciated that the trajectory exhibits *sudden* changes (the first derivative is not continuous) due to the change from ON to OFF in the transitions. The marking m_d is reached in 1.65 time units.

According to the ON-OFF approach, once the final marking is reached, all transitions are stopped. This trivially produces a steady state with no flow in which the final marking is kept. As it has been seen in previous sections, steady states with positive flows can be easily maintained as long as the flow vector is a T-semiflow. The system in Fig. 1.7 has no T-semiflows, and therefore, the final marking cannot be kept with a non-null flow vector.

In [28], it is proved that this ON-OFF control policy drives structurally persistent continuous Petri net systems towards m_d in *minimum time*. An intuitive reason for this is that, for persistent nets, the firing order is irrelevant for reaching a marking. Hence, what only matters is the amount of firings required, which is provided by σ.

1.4.4 Comparison of control methods

The availability of several control methods for TCPNs raises concerns about the selection of the most appropriate technique for a given particular system and purpose. In order to make an appropriate choice, several properties may be taken into account, e.g., feasibility, closed-loop stability, robustness, computational complexity (for the synthesis and during the application), etc.

Table 1.1 Qualitative characteristics of control laws (assuming $m_d > 0$). The following abbreviations are used: config. (configuration), min. (minimize), func. (function), compl. (complexity) and poly. (polynomial).

Technique	Computational issues	Optimality criterion	Subclass	Stability
PW-straight trajectory	a LPP for each config.	heuristic for min. time	all	yes
MPC	poly. compl. on $\|T\|, N$	min. quadratic or linear func. of m, u	all	under suf. conditions
ON-OFF	linear compl. on $\|T\|$	minimum time	structurally persistent	yes

Table 1.1 summarizes a few qualitative properties of some of the control methods described above. According to the presented properties, if the TCPN under consideration is structurally persistent, then the natural choice will be an ON-OFF control law, since it does not exhibit computational problems, ensures convergence and provides the minimum-time transient behavior. For non-persistent nets, MPC ensures convergence and minimizes a quadratic criterion. Nevertheless, when the number of transitions grows, the complexity may become intractable. In such a case, control synthesis based on other approaches as piecewise-straight trajectories would be more appropriate.

Given a TCPN system with just few configurations and transitions most of the described control laws could be synthesized and applied to it, ensuring convergence. In such a case, the criterion for selecting one of them may be a quantitative one, like minimizing either a quadratic optimization criterion or the time spent for reaching the desired marking.

1.4.5 Control with uncontrollable transitions

This subsection briefly discusses two control methods that can be used when the system contains uncontrollable transitions.

Gradient-based control with uncontrollable transitions [19]. This method produces control actions that reduce the rates of the controllable transitions from their nominal maximum values. This is equivalent to reducing the transitions flow, as considered along this chapter. However, the goal of the control problem is slightly different, since it is no longer required to drive the whole marking of the system to a desired value, but only the marking of a subset of places (the *output* of the system). The analysis is achieved in discrete time. Let us provide the basic idea for the case of a single-output system. Firstly, a cost function is defined as $v(k) = 1/2\varepsilon(k)^2$, where $\varepsilon(k)$ denotes the output error. The control proposed has a structure like: $u(k) = u(k-1) - (s(k)s(k)^T + \alpha I)^{-1}s(k)\varepsilon(k)$, where the input $u(k)$ is the rate of the controllable transitions and $s(k)$ is the output sensitivity function vector with respect to the input (the gradient vector $\nabla_u y$). The factor $\alpha > 0$ is a small term added to avoid ill conditioned matrix computations. The gradient is computed by using a first order approximation method. One of the advantages of this approach is that the change of regions (or configurations) is not explicitly taken into account during the computation of the gradient. Furthermore, a sufficient *condition for stability* is provided.

Pole assignment control with uncontrollable transitions [26]. This technique assumes initially that the initial and desired markings are equilibrium ones and belong to the same region. The control approach considered has the following structure: $u = u'_d + K(m - m'_d)$, where $(m'_d,\ u'_d)$ is a suitable intermediate equilibrium marking. The gain matrix K is computed, by using any pole-assignment technique, in such a way that the controllable poles are settled as distinct, real and negative. Intermediate markings m'_d, with their corresponding input u'_d, are computed during the application of the control law (either at each sampling period or just at an arbitrary number of them) by using a given LPP with linear complexity that guarantees that the required input constraints are fulfilled. Later, those results are extended in order to consider several regions. For this, it is required that the initial and desired markings belong to a connected union of equilibrium sets (as defined in subsection 1.3.2), i.e., $m_0 \in E_1^*$, $m_d \in E_n^*$ and $\cup_{i=1}^n E_i^*$ is connected. Thus, there exist equilibrium markings $m_1^q, ..., m_{n-1}^q$ on the borders of consecutive regions, i.e., $m_j^q \in E_j \cap E_{j+1}, \forall j \in \{1, .., j-1\}$. A gain matrix K_j, satisfying the previously mentioned conditions, is computed for each region. Then, inside each jth region, the control action $u = u'_d + K_j(m - m'_d)$ is applied, where m'_d is computed, belonging to the segment $[m_j^q, m_{j+1}^q]$, by using a similar LPP. It was proved that this control law can always be computed and applied (*feasibility*). Furthermore, *convergence* to the desired m_d was also demonstrated, whenever the conditions for controllability are fulfilled and $\cup_{i=1}^n E_i^*$ is connected (see section 1.3.2). The main drawback of this technique is that a gain matrix and a LPP have to be derived for each region in the marking path.

Similarly to the previous subsection, Table 1.2 summarizes the main features of the two presented methods.

Given that a pole assignment is required for each configuration, if the TCPN has many configurations, the implementation of the pole assignment method becomes tedious although automatizable. This problem does not appear for the gradient based controller. On the contrary, the gradient based controller does not guarantee convergence for the general case, while the pole assignment does it.

Table 1.2 Qualitative characteristics of control laws (assuming $m_d > 0$) with uncontrollable transitions. The following abbreviations are used: min. (minimize), compl. (complexity) and poly. (polynomial).

Technique	Computational issues	Optimality criterion	Subclass	Stability
Gradient-based	poly. compl. on # outputs	min. quadratic error	all	under a suf. condition
Pole-assignment	a pole-assignment for each config.	none	all	yes

1.5 Towards distributed control

A natural approach to deal with systems having large net structures is to consider decentralized and distributed control strategies. In a completely distributed approach, the model can be considered as composed of several subsystems that share information through *communication channels*, modeled by places. This problem has been addressed in few works. For instance, [27] proposes the existence of an upper-level controller, named *coordinator*. This coordinator may receive and send information to the local controllers, but it cannot apply control actions directly to the TCPN system. The existence of such coordinator increases the capability of the local controllers, allowing to consider wider classes for the net subsystems (they are assumed to be separately live and consistent, but they are not restricted to particular net subclasses). Affine control laws are proposed for local controllers. Feasibility and concurrent convergence to the required markings are proved.

We will describe in more detail an alternative approach [5] that considers a system composed of *mono-T-semiflow* (MTS) subsystems working under infinite server semantics connected through places (recall that a net is said to be MTS if it is conservative and has a unique minimal T-semiflow whose support contains all the transitions). For each subsystem, a local controller will be designed, being its goal to drive its subsystem from its initial marking to a required one. In order to achieve this goal, it must take into account the interaction with the other subsystems. For this, it is required that neighboring local controllers share information in order to meet a *consensus* that determines the amounts in which transitions must fire in order to reach the target marking. We propose to reach such a consensus by means of an iterative algorithm executed locally at each subsystem.

In order to illustrate the kind of systems that will be handled, consider a simple net modeling a car manufacturing factory composed by two plants A and D in two different cities. The Petri net model is given in Fig. 1.9. The plant A produces the car body (place p_1) and then sends it to the plant D (place p_2). The plant A can produce concurrently a limited number of car bodies (the initial marking of p_3). In plant D, the engine is constructed (p_4) and then it is put in an intermediate buffer p_5. The same plant paints the body received from plant A in p_6 and puts it in p_7 to be assembled together with the engine. The firing of t_8 means the production of a new car. We assume that D can produce concurrently a limited number of engines (initial marking of p_8) and can paint a limited number of car bodies in parallel (initial marking of p_9). Place p_b is the buffer containing the car bodies produced by plant A while p_a is the buffer containing the finished products. Since we do not want to produce more than we sell, the plant A starts to produce a new body (firing of t_1) only when a car is sold.

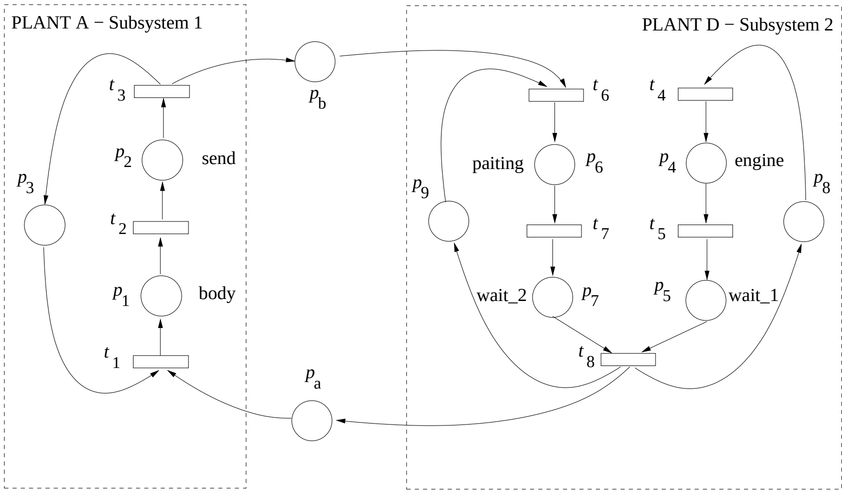

Fig. 1.9 A *distributed* marked graph modeling a car manufacturing plant where p_a and p_b are buffer places.

1.5.1 Distributed Continuous Petri nets

We will focus on distributed continuous Petri nets (DcontPN) which consists of a set of MTS net systems (called subsystems) interconnected through buffers modeled as places. Let K denote the set of subsystems of a given DcontPN. The set of places, transitions and token flow matrix of subsystem $k \in K$ is denoted by P^k, T^k and $C^k \in \mathbb{R}^{|P^k|\times|T^k|}$, respectively. We assume, $P^k \cap P^l = \emptyset$ and $T^k \cap T^l = \emptyset$, $\forall k,l \in K,\ k \neq l$. The directional connection between subsystems is provided by a set of places called *channel* or *buffer places*. In particular, the directional connection from subsystem k to l is provided by a set of places denoted $B^{(k,l)}$, whose input transitions are contained only in subsystem k and output transitions are contained only in subsystem l, i.e., $B^{(k,l)} = \{p \in P|^\bullet p \in T^k,\ p^\bullet \in T^l,\ p \notin P^q\ \ \forall q \in K\}$ for every $k,l \in K$, $k \neq l$, and $B^{(l,l)} = \emptyset$ for every $l \in K$.

Note that a place $p \in B^{(k,l)}$ is an *input buffer* of subsystem l and an *output buffer* of subsystem k. The set of all output buffers of subsystem k is denoted by $B^{(k,*)}$, i.e., $B^{(k,*)} = \bigcup_{l \in K} B^{(k,l)}$, and the set of all input channels of subsystem k is denoted by $B^{(*,k)}$, i.e., $B^{(*,k)} = \bigcup_{l \in K} B^{(l,k)}$.

The marking vector of a subsystem k is denoted by $m(P^k) \in \mathbb{R}_{\geq 0}^{|P^k|}$. When designing a controller, it must be taken into account that the controller of a given subsystem only knows its marking and the marking of its input buffers, i.e., the marking of the other subsystems and their input buffers are not observable.

Among the different existing control problems, we will deal with a control problem of DcontPN which aims at reaching a particular target marking m_d at each subsystem. That is, after a finite period of time each subsystem is at its target marking. In contrast to a centralized control, each subsystem is equipped with its own controller that computes the control actions that drive the subsystem to the target marking. Given that the subsystems are interconnected, they may require resources to be available in the communication buffers to reach the target marking. The following example shows this situation.

Example 7. Consider the DcontPN in Fig. 1.9 with $m_0(P^1) = [0\ 0\ 3]^T$, $m_0(P^2) = [0\ 0\ 0\ 0\ 2\ 2]^T$, $m_0(p_a) = 1$, $m_0(p_b) = 0$ and let $m_d(P^1) = [0\ 0\ 3]^T$, $m_d(P^2) = [0\ 0\ 1\ 0\ 2\ 1]^T$ be the target markings of each subsystem. Let the flow integrals of subsystem 1 and 2 be denoted as s^1 and s^2 respectively.

Let us assume that the controller of the second subsystem computes $s^2(t_6) = 1$, $s^2(t_4) = s^2(t_5) = s^2(t_7) = s^2(t_8) = 0$ so that the subsystem reaches the target marking. Given that the initial marking and target marking of subsystem 1 are the same, a controller for that subsystem could yield: $s^1(t_1) = s^1(t_2) = s^1(t_3) = 0$. Since $m_0(p_b) = 0$, transition t_6 cannot fire unless t_3 fires. Unfortunately, according to the computed controls, t_3 will not fire ($s^1(t_3) = 0$). Hence, these controls are not valid to reach the desired target marking of subsystem 2. In order to solve this situation, subsystem 2 may ask subsystem 1 to put enough tokens in p_b. This can be achieved easily by firing t_3. However, this will imply that subsystem 1 moves away from its desired target marking.

Apart from the problem of tokens (resources) required in the buffer places at the initial time, it could happen that the target markings cannot be reached due to the system structure and initial marking (the following example deals with this case). When stating the problem we are implicitly assuming that all target markings of subsystems are reachable, meaning that the final marking of the overall net system, i.e., the net containing all subsystems and buffers, is reachable.

Example 8. Consider again the DcontPN in Fig. 1.9. For subsystem 1, let the target marking be $m_d(P^1) = [0\ 0\ 3]^T$ which is reachable from $m_0(P^1) = [0\ 0\ 3]^T$ locally. For subsystem 2, let the target marking be $m_d(P^2) = [0\ 0\ 1\ 0\ 2\ 1]^T$ which is reachable from $m_0(P^2) = [0\ 0\ 0\ 0\ 2\ 2]^T$ locally by firing t_6, i.e., if it is considered isolated from the rest of the system. But when both subsystems are connected through the buffers p_a and p_b with $m_0(p_a) = m_0(p_b) = 0$, the target markings are not reachable.

1.5.2 A control strategy for DcontPNs

In this subsection, a distributed controller for DcontPN with *tree structure* is proposed (this extends the results in [4] where the problem has been studied for DcontPN with two subsystems). In a system with tree structure cycles are not allowed. The following assumptions will be taken on the considered DcontPNs: (A1) The target marking m_d is strictly positive and reachable at the overall system; (A2) The DcontPN is composed of MTS subsystems. The minimal T-semiflows of the subsystem i is denoted by x^i; (A3) The overall system is a MTS net system.

The first assumption is simply a necessary condition for reachability of the target markings. The second assumption reduces the class of DcontPN to those systems composed by MTS subsystems while the third one states that the overall system is MTS. In order to drive the subsystems from their initial states to the target states, Algorithm 5 is developed. It represents logic of the rules to be executed in each subsystem to meet a consensus.

In step 1, each subsystem computes the flow integral $\bar{s}$ required to reach its target marking without taking into account the marking of the buffers. Step 2 computes the amounts of tokens q_p^{req} to be produced in each input buffer p in order to be able to fire $\bar{s}$. The connected subsystems are informed about the amounts of required tokens q_p^{req} in step 4. In step 5, each subsystem receives the amount of tokens it has to produce (if any) in its output buffers. In step 6, it is computed how many tokens would remain in each output buffer if the present control was applied. If this value is negative, more tokens must be produced in the output buffers, and therefore the control law must be recomputed. This re-computation is achieved in step 7 using LPP (1.10). Observe that comparing with LPP (1.7) of step 2 only one extra constraint is added in order to ensure that enough tokens are produced in the output buffers. Steps 4-7 are repeated $|K|-1$ times in order to allow the communication along the longest path connecting a pair of subsystems.

The following Theorem shows that Algorithm 5 computes a control law for all subsystems that ensures the reachability of their target markings (see [5] for the proof).

Proposition 6 *Let N be a DcontPN with tree structure satisfying assumptions (A*1*), (A*2*) and (A*3*), and let s^k be the flow integral vectors computed by Algorithm 1 for each subsystem for a given initial and target marking. The application of s^k drives the subsystems to their target markings.*

Example 9. Consider the net system in Fig. 1.9 used also in Ex. 8. Assume for the first subsystem the same initial and desired markings: $m_0(P^1) = m_d(P^1) = [0\ 0\ 3]^T$ while for the second one: $m_0(P^2) = [0\ 0\ 0\ 0\ 2\ 2]^T$ and $m_d(P^2) = [0\ 0\ 1\ 0\ 2\ 1]^T$. For the buffers, let us assume $m_0(p_a) = 1$, $m_0(p_b) = 0$. Let us compute local control laws in each subsystem. For the first one, since $m_0(P^1) = m_f(P^1)$, the minimum firing vector is unique equal to $s^1 = [0\ 0\ 0]^T$, i.e., not firing any transition. For the second subsystem, it is easy to observe that the minimum firing vector is $s^2 = [0\ 0\ 1\ 0\ 0]^T$, i.e., firing t_6 in an amount equal to 1. Notice that t_6 cannot fire from the initial marking because $m_0(p_b) = 0$. In order to avoid this, it is possible to fire once the

Algorithm 5 *[Distributed controller of subsystem k]*
Input: $C^k, m_0(P^k), m_d(P^k), B^{(k,*)}, B^{(*,k)}, m_0(B^{(k,*)})$
Output: *flow integral vector s*
1) Solve

$$\begin{array}{ll} \min & 1^T \cdot \bar{s} \\ s.t. & m_d(P^k) - m_0(P^k) = C^k \cdot \bar{s}, \\ & \bar{s} \geq 0 \end{array} \tag{1.7}$$

2) ***Repeat*** $|K|-1$ *times*
3) For every $p \in B^{(*,k)}$ *calculate*

$$q_p^{req} = \Big(\sum_{t \in p^\bullet} Pre(p,t) \cdot \bar{s}(t)\Big) - m_0(p) \tag{1.8}$$

4) For all $p \in B^{(*,k)}$ *send* q_p^{req} *to the connected subsystem*
5) For all $p \in B^{(k,*)}$ *receive* r_p^{req} *from the connected subsystem*
6) For all $p \in B^{(k,*)}$ *calculate*

$$h_p = \Big(\sum_{t \in {}^\bullet p} Post(p,t) \cdot \bar{s}(t)\Big) - r_p^{req} \tag{1.9}$$

7) ***If*** $\min_{p \in B^{(k,*)}} \{h_p\} < 0$ ***then*** *solve*

$$\begin{array}{ll} \min & 1^T \cdot s \\ s.t. & m_d(P^k) - m_0(P^k) = C^k \cdot s, \\ & \sum_{t \in {}^\bullet p} Post(p,t) \cdot s(t) \geq r_p^{req}, \forall p \in B^{(k,*)} \\ & s \geq 0 \end{array} \tag{1.10}$$

Else
$s = \bar{s}$
End If
8) $\bar{s} = s$
9) ***End Repeat***
10) ***return*** *s*

T-semiflow of subsystem 1 (equal to the vector of ones). This is the control action that Algorithm 5 computes for subsystem 1 after the first iteration. The algorithm performs just one iteration because, in this example, $|K| = 2$.

Once the flow integral vectors s of the evolution from the initial marking to the target marking have been computed by Algorithm 5, the value of the control actions u can be derived in several ways (for example applying the procedure in [3]) as long as $s = \int_{\tau_a}^{\tau_b} (f-u) d\tau$ is satisfied where τ_a and τ_b are the initial and final time instants respectively. Remark that s can be seen as a firing count vector in the untimed system and the problem of finding a control law u is equivalent to a reachability problem: if the desired marking is reachable in the untimed net system it is reachable in the timed one with an appropriate control law if all transitions are controllable. This result is proved in [21] (Prop. 14. 3) where a procedure that executes a firing sequence of the untimed system in the timed one is also presented.

1.5.3 Further reading

For further reading on the presented topics, the reader is referred to the survey paper [22]. An introduction to fluidization of net models can be found in [23], and a comprehensive definition and application examples of discrete, continuous and hybrid Petri nets can be found in the book by R. David and H. Alla [9].

References

1. A. Amer-Yahiaj, N. Zerhouni, A. El-Moudni, and M. Ferney. State variable description and controllability of a class of continuous Petri nets. In *Proc. of the IEEE Int. Symp. on Circuits and Systems*, 1996.
2. A. Amrah, N. Zerhouni, and A. El-Moudni. On the control of manufacturing lines modelled by controlled continuous Petri nets. *Int. Journal of Systems Science*, 29(2):127–137, 1998.
3. H. Apaydin-Ozkan, J. Julvez, C. Mahulea, and M. Silva. An efficient heuristics for minimum time control of continuous Petri nets. In *3rd IFAC Conf. on Analysis and Design of Hybrid Systems, ADHS09*, 2009.
4. H. Apaydin-Ozkan, J. Julvez, C. Mahulea, and M. Silva. A control method for timed distributed continuous Petri nets. In *Proc. of the American Control Conf. 2010*, Baltimore, Maryland, USA, 2010. IEEE Press.
5. H. Apaydin-Özkan, C. Mahulea, J. Júlvez, and M. Silva. An iterative control method for distributed continuous petri nets with tree structures. In *Conf. on Decision and Control CDC 2010*, 2010.
6. A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
7. R. Brammer. Controllability in linear autonomous systems with positive controllers. *SIAM Journal Control*, 10(2):329–353, 1972.
8. C. Chen. *Linear system theory and design*. Oxford University Press, New York, 1984.
9. R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2004. (Revised 2^{nd} edition, 2010).
10. M. Dub, G. Pipan, and Z. Hanzálek. Stock optimization of a kanban-based assembly line. In *Proc. of the 12th Int. Conf. on Flexible Automation and Intelligent Manufacturing*, 2002.
11. A. Giua, F. DiCesare, and M. Silva. Petri Net Supervisors for Generalized Mutual Exclusion Constraints. In 12^{th} *IFAC World Congress*, volume 1, pages 267–270, Sidney, Australia, July 1993.
12. L.E. Holloway, B.H. Krogh, and A. Giua. A survey of Petri nets methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.
13. M.V. Iordache and P.J. Antsaklis. *Supervisory control of concurrent systems*. Birkhäuser, Boston, 2006.
14. E. Jiménez, J. Júlvez, L. Recalde, and M. Silva. On controllability of timed continuous Petri net systems: the join free case. In *Proc. of the 44th IEEE Conf. on Decision and Control (Joint CDC-ECC), Seville, Spain*, 2005.
15. X. Jing, L. Recalde, and M. Silva. Tracking control of join-free timed continuous Petri net systems under infinite servers semantics. *Journal of Discrete Event Dynamic Systems*, 18(2):263–283, 2008.
16. X. Jing, L. Recalde, and M. Silva. Tracking control of timed continuous Petri net systems under infinite servers semantics. In *IFAC World Congress*, 2008.
17. R. Kara, S. Djennoune, and J.J. Loiseau. State feedback control for the manufacturing systems modeled by continuous Petri nets. In *Proc. of Information Control Problems in Manufacturing (INCOM)*, 2006.

18. D. Lefebvre. Feedback control designs for manufacturing systems modelled by continuous Petri nets. *Int. Journal of Systems Science*, 30(6):591–600, 1999.
19. D. Lefebvre, D. Catherine, E. Leclerq, and F. Druaux. Some contributions with Petri nets for the modelling, analysis and control of HDS. *In Proc. of the Int. Conf. on Hybrid Systems and Applications*, 1(4):451–465, 2007.
20. C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous Petri nets. *Automatic Control, IEEE Trans. on*, 53(7):1731–1735, Aug. 2008.
21. C. Mahulea, A. Ramirez, L. Recalde, and M.Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Trans. on Automation Science and Engineering*, 5(2):307–320, April 2008.
22. M. Silva, J. Júlvez, C. Mahulea, and C..R. Vázquez. On fluidization of discrete event models: observation and control of continuous petri nets. *Discrete Event Dynamic Systems*, 21:427–497, 2011. 10.1007/s10626-011-0116-9.
23. M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28:253–266, 2004.
24. C.R. Vázquez, A. Ramírez, L. Recalde, and M. Silva. On controllability of timed continuous Petri nets. In *Egerstedt M., Mishra B. eds.: Proc. of the 11th Int. Workshop Hybrid Systems: Computational and Control (HSCC08), LNCS*, volume 4981, pages 528–541, 2008.
25. C.R. Vázquez and M. Silva. Performance control of Markovian Petri nets via fluid models: A stock-level control example. In *5th IEEE Conf. on Automation Science and Engineering (IEEE CASE)*, 2009.
26. C.R. Vázquez and M. Silva. Piecewise-linear constrained control for timed continuous Petri nets. In *Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, 2009.
27. C.R. Vázquez, J.H. van Schuppen, and M. Silva. A modular-coordinated control for continuous Petri nets. In 18^{th} *IFAC World Congress*, Milano, Italy, July 2011.
28. L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time control for structurally persistent continuous Petri nets. In *Conf. on Decision and Control CDC 2010*, 2010.