

## **UM FORMALISMO DE DECISÃO PARA SUPERVISORES DE SISTEMAS A EVENTOS DISCRETOS**

A DECISION'S FORMALISM FOR SUPERVISORS OF DISCRETE EVENT SYSTEMS

**Gilberto Bispo da Silva Góes**

Departamento de Tecnologia Mecânica e Materiais  
Centro Federal de Educação Tecnológica da Bahia  
Salvador, BA, Brasil  
Email: [gilbertogoes@cefetba.br](mailto:gilbertogoes@cefetba.br)

**Eduard Montgomery Meira Costa**

Departamento de Engenharia Elétrica  
Universidade Federal da Bahia  
Salvador, BA, Brasil  
Email: [edmonty@ig.com.br](mailto:edmonty@ig.com.br)

### **ABSTRACT**

It's introduced the formalism to synthesize the supervisor for Timed Discrete Event Systems that ensure the continuity of processes. Here, the paradigm used to model the DES is the Synchronous Petri Net, in which, new concepts are introduced for control formalization both transitions enabling functions and transitions forced firing functions. The definition of forcible transitions is based on Brandin and Wonham for timed DES. The use of Synchronous Petri Nets ensures the system functionality through external events generated by active sensors in the system. It's proposed formalism ensures the timing process continuity, despite any out coming miss at sensor. An example of supervisor synthesis using the proposed formalism is presented to shown its efficacy.

**KEY WORDS:** Petri Nets, Supervisory Control, Discrete Event Systems, Manufacturing

### **RESUMO**

O formalismo para sintetizar supervisores para Sistemas a Eventos Discretos Temporizados que garantem a continuidade de processos é introduzido. Aqui, o paradigma utilizado para modelar o SED é a rede de petri síncrona, na qual são introduzidos novos conceitos para a formalização do controle, como funções de habilitação de transições e funções de disparos forçados de transições. A definição de transição forçável é baseada nos trabalhos de Brandin e Wonham para SEDs temporizados. O uso da rede de Petri síncrona assegura a funcionalidade do sistema através de eventos externos gerados por sensores ativos no sistema. O formalismo proposto garante a continuidade temporal do processo, mesmo havendo uma falha em algum sensor. Um exemplo de síntese do supervisor usando o formalismo proposto é apresentado para mostrar sua eficácia.

**PALAVRAS-CHAVE:** Redes de Petri, Controle Supervisório, Sistemas a Eventos Discretos, Manufatura.

## **1 INTRODUÇÃO**

Sistemas a Eventos Discretos (SEDs) [1] são sistemas que apresentam uma evolução dinâmica descrita pela ocorrência de eventos físicos que alteram seu estado. Os eventos em um SED são instantâneos, o que confere a estes sistemas o caráter de discreto no tempo. Os SEDs estão presentes em aplicações do cotidiano como redes de computadores, sistemas de manufatura e supervisão de tráfego aéreo e ferroviário. O estudo dos SEDs requer a utilização de uma representação adequada e que permita projetar um agente de controle automático, denominado supervisor. A partir de tarefas especificadas, o supervisor recebe informações da ocorrência dos eventos do SED e

determina a ação de controle a ser aplicada, enviando comandos para os atuadores que inibem ou habilitam determinados eventos.

A formalização do problema de controle de SEDs utilizando autômatos e linguagens formais [2] é denominada de Teoria de Controle Supervisório (TCS) [3], que é uma forma elegante de resolver o problema de controle de SEDs: a partir do modelo do SED e de uma especificação funcional, determina um supervisor. A TCS define condições para o controle considerando o comportamento lógico. Entretanto, em várias situações, há a necessidade explícita de definir instantes de tempos exatos para a aplicação de controle, bem como para impor sincronismo entre as ocorrências dos eventos. Para isto,

torna-se necessário a inclusão de uma representação temporal no SED.

Várias formalizações para o controle de SEDs temporizados são encontradas na literatura. Entre elas, encontra-se a abordagem da TCS para SEDs temporizados de Brandin e Wonham [4], utilizando autômatos que incluem a representação do tempo definido por meio de um evento sincronizado ao relógio global, denominado evento 'tick'. Quando se tratando da necessidade de sincronização entre eventos, a álgebra de dióides [5] tem sido largamente utilizada, seja para os autômatos temporizados [6,7], como visto em [8,9,10], seja para as redes de Petri temporizadas [11], como pode ser visto em [12,13]. Outros formalismos para a síntese de supervisores de SEDs temporizados que não utilizam a álgebra de dióides são encontrados, como em [14,15]. Entretanto, é em Silva [16] que é apresentado um importante formalismo para esta classe de SEDs, onde a abordagem de Barroso [17] é estendida para o caso temporizado, sendo aplicado a modelos de SEDs temporizados descritos por redes *seguras*. Nesta abordagem, de maneira similar à Barroso, Silva garante a construção de supervisores que mantêm a mesma estrutura do modelo do SED para qualquer especificação de comportamento definida, para tanto são definidas as redes de Petri Temporais Interpretadas com Função de Habilitação de Transição (RPTIFHT) para estruturar o supervisor, sendo formalizado um algoritmo para a construção da árvore de alcançabilidade de classes (AAAC), em que uma classe determina um conjunto de estados que contêm a mesma marcação com diferentes intervalos de tempo associados.

O formalismo de Silva se aplica a casos em que as associações de intervalos de tempo não são tão complexas como o caso descrito pela rede de Petri com Temporização Nebulosa [18]. Assim, no problema tratado por Silva, os intervalos de tempo podem ser obtidos por estimação ou por simples medição e sua aplicação permitiu construir um algoritmo para transformação das funções da rede supervisora em linguagem de Controladores Lógicos Programáveis, ou CLP (*Instruction List*).

Em outras abordagens de controle de SEDs utilizando redes de Petri, encontram-se formalismos para a transformação da estrutura da rede em programas de CLP, como nos trabalhos de Uzam *et al* [19], Frey e Litz [20,21] e Chirn e McFarlane [22], bem como para o formalismo temporizado, como em Uzam *et al* [23].

Entretanto, em nenhum destes casos, o supervisor sintetizado realiza tomada de decisão dentro da operação do sistema. Isto é, se houver falha em algum dos sensores do sistema físico que atualizam o CLP, o sistema pode parar sua execução esperando uma resposta de algum sensor, o que não é desejado. Também, observa-se que, com a inclusão de intervalos de tempo no modelo do sistema, como é o caso de Silva [16], há uma "aparente introdução de controle" no modelo, como descreve Costa [10], ou seja, na estruturação de um modelo livre de controle, um evento pode ou não ocorrer, o que implica que a duração da habilitação deve sempre ser considerada

com limite superior infinito. Por exemplo, a perfuração de uma chapa de aço, onde após o início de sua perfuração, um intervalo limite superior que implica no término da perfuração, pode não condizer com a realidade, pois nada se garante sobre acontecimentos imprevistos que impeçam seu término. Dessa forma, uma importante contribuição surge na utilização das redes de Petri Síncronas (RPS) [24] com uma introdução de temporizadores nas transições, para a aplicação de controle a SEDs temporizados, considerando algumas características para o sistema:

1. O modelo do sistema é considerado com todos os eventos descritos como eventos *remotos* [4], isto é, têm apenas a descrição do limite inferior para que o evento se torne habilitado, o limite superior é infinito, o que descreve o sistema livre de controle;
2. Todos os tempos de vida são considerados com limite inferior de tempo igual a zero, isto é, o evento está habilitado a partir do momento que o estado é atingido, mas que só podem disparar quando um evento externo ocorre, o que garante o uso das RPS na modelagem do SED;
3. Todos os eventos são descritos com igual prioridade de ocorrência, o que também garante que não há condições de controle prévio;
4. Alguns eventos controláveis podem ser definidos como *eventos forçáveis* [4], de forma que seja possível interferir na execução do sistema, como em [25].

Através destas considerações, pode-se utilizar o formalismo de Barroso [17] para construir a rede supervisora. Neste caso, observa-se que a árvore de alcançabilidade da RPS tem uma formalização similar às redes não temporizadas, e permitem incluir as funções de habilitação de transição, bem como um novo formalismo definido como temporizações que determinam limites superiores para o supervisor forçar o disparo das transições.

Este artigo é organizado como segue: na Seção 2 são apresentadas as definições das redes de Petri síncronas e sua árvore de alcançabilidade; na Seção 3 é apresentada a formalização para a síntese de supervisores com decisão; na Seção 4 é apresentado um exemplo ilustrativo da abordagem proposta e na Seção 5 são apresentadas as conclusões deste trabalho.

## 2 REDES DE PETRI SÍNCRONAS

Uma rede de Petri Síncrona (RPS) [24] pode ser definida como em Silva [16]:

**Definição 1** Uma Rede de Petri Síncrona é uma tripla

$$RPS = (RP, E, FE), \quad (1)$$

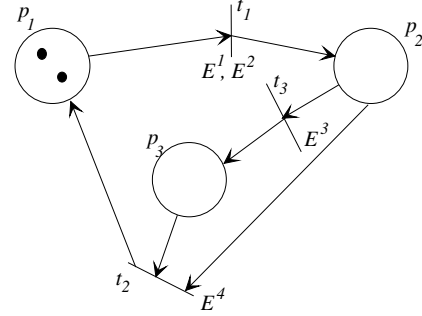
em que  $RP$  é uma rede de Petri marcada;  $E$  é um conjunto de eventos externos;  $FE$  é uma função do conjunto de transições  $T$  da  $RP$  para  $E \cup \{e\}$ , com  $e$  sendo o evento que possui ocorrência permanente, isto é, ele é o elemento neutro do monóide  $(E^1 + \dots + E^p)^*$ , onde  $E = \{E^1, \dots, E^p\}$  é o conjunto de eventos externos.

Esta definição é uma formalização mais simplificada que em [24], em que se consideram todos os eventos externos com as mesmas prioridades, sendo a que será utilizada para este trabalho. Nessa definição, o evento  $e$  é um evento que corresponde à sequência de eventos externos cujo período é zero. Uma transição sincronizada por este evento é disparada assim que ela se torna habilitada. A notação  $E^i$  refere-se ao nome de um evento externo e a notação  $E_j$  será utilizada para o evento associado à transição  $t_j$ .

Uma transição em uma RPS necessita que, além da habilitação formal definida para as redes de Petri, isto é, todos os lugares de entrada da transição devem ter o número de fichas que garantam seu disparo (e os lugares de saída não ultrapassem sua capacidade), é necessário que um evento externo  $E$ , que esteja associado à transição, ocorra para que ela possa disparar. Esta situação define que a transição está receptiva ao evento  $E$ , e com sua ocorrência, ela dispara, gerando uma nova marcação conforme as regras de disparo das redes de Petri clássicas.

As condições citadas sobre a receptividade de uma transição  $t$  se apresentam como uma importante formalização para a estruturação de controle de SEDs, quando é desejado avaliar a dinâmica do sistema físico sobre a visão do modelo, desde que os eventos externos podem ser as informações vindas do sistema através dos sensores. Assim, o estado do modelo em RPS do sistema, seja este formalizado em um computador, ou em um CLP, só é atualizado quando ocorre a chegada de uma informação de um determinado sensor do sistema.

**Exemplo 1** Considere a RPS apresentada na Figura 1. Nesta rede, tem-se o conjunto de eventos externos que estão associados às transições dado por  $E = \{E^1, E^2, E^3, E^4\}$ . Os eventos  $E^1$  e  $E^2$  estão associados à transição  $t_1$ , tal que quando esta transição está habilitada, ela é receptiva a estes dois eventos. Para a transição  $t_2$  há apenas um evento associado que é o evento externo  $E^4$ , e para a transição  $t_3$ , tem-se associado o evento  $E^3$ . Na marcação  $M_0 = [2 \ 0 \ 0]^T$ , a única transição habilitada é a transição  $t_1$ , a qual é receptiva aos eventos externos  $E^1$  e  $E^2$ . Assim, a ocorrência de qualquer um destes eventos faz a transição  $t_1$  disparar, levando à marcação  $M_1 = [1 \ 1 \ 0]^T$ , e uma nova ocorrência de qualquer um destes eventos leva a rede à marcação  $M_2 = [0 \ 2 \ 0]^T$ . Da mesma forma, na marcação  $M_1$ , se ocorre o evento  $E^3$ , a transição  $t_3$  que também está receptiva, dispara levando à marcação  $M_3 = [1 \ 0 \ 1]^T$ , e assim por diante. Entretanto, para uma marcação, como  $M_0$ , se ocorrer o evento  $E^4$ , que não está associado à transição  $t_1$ , não há mudança de marcação na rede, desde que ela é a única habilitada, porém não é receptiva a este evento.



**Figura 1:** Exemplo de uma RPS.

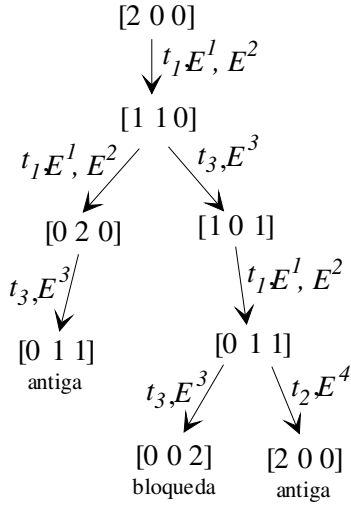
Deve-se observar no Exemplo 1 que a diferença básica da regra de disparos das transições, refere-se unicamente à sua receptividade em relação a um evento externo que esteja associado a esta transição. Dessa forma, a evolução dinâmica da RPS é similar à rede de Petri clássica.

## 2.1 ÁRVORE DE ALCANÇABILIDADE DAS RPS

Como visto no Exemplo 1, uma transição habilitada numa marcação  $M$ , dispara se houver a ocorrência de um evento associado a esta transição. Uma sequência de ocorrências de eventos externos que estejam associados a transições receptivas nesta sequência gera uma sequência de disparos de transições. Dessa forma, a evolução dinâmica da RPS tem uma forma similar às redes de Petri clássicas, podendo ter a árvore de alcançabilidade descrita da mesma maneira, apenas associando aos arcos os eventos que tornam as respectivas transições receptivas. Deve-se observar que estas condições são garantidas para o caso específico tratado aqui.

De acordo com as sequências de eventos externos ocorrendo no sistema, as transições irão disparar formando sequências de marcações, como visto no Exemplo 1. Assim, pode-se construir toda a árvore de alcançabilidade da RPS, desconsiderando a ocorrência de eventos externos.

**Exemplo 2** A RPS do Exemplo 1 tem sua árvore de alcançabilidade mostrada na Figura 2.



**Figura 2:** Árvore de alcançabilidade de uma RPS.

No Exemplo 2, como discutido anteriormente, observa-se que apenas são consideradas as ocorrências dos eventos externos associados a transições que estejam habilitadas, tornando a árvore de alcançabilidade similar à árvore de alcançabilidade de uma rede de Petri clássica com a mesma estrutura (não considerando a inclusão dos eventos externos). Com este formalismo, pode-se utilizar a abordagem de Barroso [17] como base para construir supervisores utilizando RPS, objeto deste trabalho.

### 3. FORMALIZAÇÃO PARA A SÍNTESE DE SUPERVISORES

Para definir a síntese do supervisor para os SEDs nesta abordagem, é necessário incluir temporizadores para as transições das RPS. Isto é, devem ser incluídos temporizadores locais para as transições da rede, os quais são incrementados sempre que uma marcação que habilita uma transição  $t$  é alcançada. Em seu disparo, o temporizador da transição  $t$  disparada é zerado, mantendo os temporizadores das outras transições habilitadas, que não são desabilitadas com o disparo de  $t$ , na mesma marcação. Assim, tem-se:

**Definição 2** Uma Rede de Petri Síncrona com Temporizadores é uma dupla

$$RPST = (RPS, \mathcal{T}),$$

em que RPS é uma Rede de Petri Síncrona marcada;  $\mathcal{T} = \{\tau_1, \dots, \tau_m\}$  é um conjunto de temporizadores sincronizados a um relógio global, os quais são associados às transições, tal que  $\forall M \mid M \in MA(RPS, M_0), \tau_i = 0$  se a transição  $t_i$  não está habilitada em  $M$ , e  $\tau_i = \tau_i + 1$  em sincronia com o relógio global, enquanto  $t_i$  estiver habilitada.

Com a formalização da RPST, pode-se obter os exatos tempos de ocorrência dos eventos externos associados a

uma determinada transição, bem como o tempo total de execução para caminhos (seqüências de disparos) da evolução dinâmica da rede, quando esta modela um SED. Dessa forma, como os temporizadores dependem da ocorrência dos eventos externos, não faz sentido construir uma árvore de alcançabilidade para a RPST que descreva os valores desses temporizadores nos momentos de tais ocorrências. Esta inclusão é apenas um formalismo para a estrutura de controle aqui apresentada, e sua árvore de alcançabilidade é a mesma da RPS.

A abordagem de Barroso [17] mostra que a síntese do supervisor baseada em redes de Petri necessita da construção da sua árvore de alcançabilidade com um passo a mais para detecção dos estados com crescimento ilimitado de fichas, e esta serve como base junto à especificação de comportamento para sintetizar um supervisor formalizado sobre uma Rede de Petri com Funções de Habilitação de Transições (RPFHT), que são redes que permitem controlar, através de funções lógicas em suas transições, a dinâmica da rede. Essas funções são dependentes da marcação de lugares quaisquer da rede (e não só os lugares a ela ligados por arcos).

Desde que o formalismo para a construção da árvore de alcançabilidade da RPS com as condições definidas para o modelo do sistema apenas difere na inclusão dos eventos externos associados às transições e a árvore de alcançabilidade da RPST é a mesma, o algoritmo AMArA (Algoritmo Modificado da Árvore de Alcançabilidade) desenvolvido por Barroso [17] pode ser utilizado sem problemas para a construção da árvore de alcançabilidade modificada das RPS. De igual forma, apenas os eventos externos serão incluídos junto às transições disparáveis nos arcos da árvore, como visto no Exemplo 2.

Tendo a árvore de alcançabilidade da RPST, uma especificação de comportamento deve ser definida. Esta especificação de comportamento, ou a tarefa que o sistema deve realizar é formalizada de maneira similar à Barroso. Esta similaridade se apresenta por ela ter de ser definida pela evolução lógica da rede como uma linguagem ou uma marcação a ser alcançada, o que irá definir as condições de não bloqueio e funções para que a rede não alcance estados não permitidos.

Considerando que a rede evolui sincronamente com os eventos externos associados às transições, e não é interessante que a rede se mantenha parada devido a não ocorrência de algum evento externo, ou seja, se algum evento externo não ocorre num determinado tempo tolerado, é necessário desviar a execução da rede, forçando o disparo de uma transição habilitada/receptiva naquela marcação. Os tempos limites tolerados para as ocorrências dos eventos externos da rede podem ser definidos sobre os tempos obtidos nas suas respectivas ocorrências através de medições simples no sistema (resposta dos sensores).

Como o supervisor deve ser construído utilizando uma RPST para modelar o sistema, torna-se necessário incluir na estrutura da RPST dois tipos de funções associadas às transições:

1. Funções de Habilitação de Transições - responsáveis pela habilitação ou inibição da transição em determinadas marcações;
2. Funções de Disparo Forçado de Transições - responsáveis pelo disparo forçado de uma transição habilitada em uma determinada marcação.

As funções de disparo forçado são funções específicas para comparação com os temporizadores locais das transições. Estas funções disparam forçadamente transições habilitadas quando o temporizador de uma transição receptiva ultrapassa o limite máximo de tempo permitido para a ocorrência de um de seus eventos externos associados.

Com esses conceitos, define-se uma Rede de Petri Síncrona com Temporizadores e Funções de Habilitação de Transições (RPSTFHT) como a seguir:

**Definição 3** Uma Rede de Petri Síncrona com Temporizadores e Funções de Habilitação de Transições é uma tripla

$$RPSTFHT = (RPST, \phi, \pi),$$

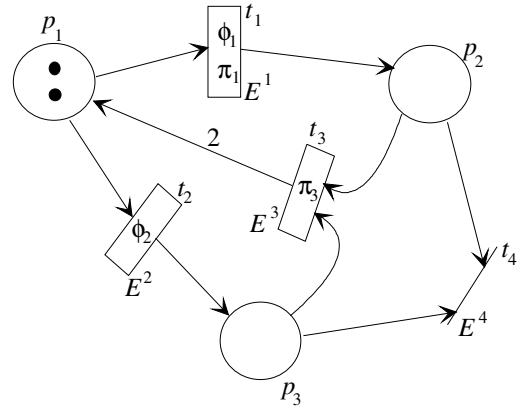
em que  $RPST$  é uma rede de Petri Síncrona com Temporizadores;  $\phi = \{\phi_1, \dots, \phi_m\} : MA(RP, M_0) \rightarrow \{0, 1\}$  é a função de habilitação das transições, que mapeia o conjunto de marcações alcançáveis em 0 ou 1;  $\Pi = \{\pi_1, \dots, \pi_m\} : \Phi \times T \times T \rightarrow \{0, 1\}$  é a função de disparos forçados das transições, tal que

$$\pi_i = \begin{cases} 1, & \text{se } \phi_i = 1 \text{ e } \tau_i > \tau_{ref}; \\ 0, & \text{caso contrário,} \end{cases} \quad (2)$$

com  $i$  representando as transições de saída do lugar  $p$  receptivas aos eventos externos e  $\tau_{ref}$  os valores de referência (limites máximos de tempo para ocorrência dos eventos associados) para forçar a transição  $t_i$ .

Desta definição, pode-se ver que as RPSTFHT unem as RPFHT com as RPS, bem como inclui um novo formalismo para forçar transições em casos da não ocorrência de determinados eventos externos associados a outras transições habilitadas/receptivas. Neste caso, para uma transição estar habilitada, ela necessita apresentar as condições da RPFHT (satisfazer as condições das redes de Petri clássicas e a função de habilitação ter valor 1). Por outro lado, se para uma dada transição habilitada/receptiva  $t_i$  uma função  $\pi_i$  for definida em relação ao limite máximo de tempo  $\tau_{ref}$  para a ocorrência de um evento externo associado a uma outra transição habilitada/receptiva  $t_j$  e seu temporizador ultrapassar este valor definido ( $\tau_i > \tau_{ref}$ )  $\pi_i=1$  e ela ( $t_i$ ) é forçada a disparar. Para esta rede, o valor padrão das funções  $\phi$  é 1, quando esta função não é definida em uma transição, significando que ela sempre está habilitada. Da mesma forma, o valor padrão das funções  $\pi$  é 0, quando esta função não é definida em uma transição, significando que ela nunca é forçada a disparar, mas podendo disparar em qualquer momento de acordo com as regras de disparo das RPFHTs.

**Exemplo 3** Considere a RPSTFHT apresentada na Figura 3. Nesta rede, as funções associadas às transições são:  $t_1$ :  $\phi_1=[M(p_1) \geq 1 \wedge M(p_2) > 0]$ ,  $\pi_1=[\tau_1 > 3]$ ;  $t_2$ :  $\phi_2=[M(p_1) \geq 1 \wedge M(p_3) > 0]$ ,  $\pi_2=0$  (valor padrão);  $t_3$ :  $\phi_3=1$  (valor padrão),  $\pi_3=[\tau_3 > 2]$ ;  $t_4$ :  $\phi_4=1$  (valor padrão),  $\pi_4=0$  (valor padrão). No início da execução da rede, o relógio global é inicializado em conjunto com os temporizadores locais das transições  $t_1$  e  $t_2$ , que são as únicas transições habilitadas pelas condições normais de habilitação das redes de Petri clássicas, além de atender as exigências das funções de habilitação pelas funções  $\phi_1$  e  $\phi_2$ , e receptivas aos eventos externos  $E^1$  e  $E^2$ , respectivamente. A transição  $t_1$  pode disparar a qualquer instante de tempo antes de 3 unidades de tempo passados, com a ocorrência do evento externo  $E^1$  associado a ela. Entretanto, após estas 3 unidades de tempo, esta transição é forçada a disparar, levando a rede para a marcação  $M=[1 \ 1 \ 0]^T$  (caso  $E^2$  não tenha ocorrido e disparado  $t_2$ ), ou para a marcação  $M=[0 \ 1 \ 1]^T$  (caso  $E^2$  tenha ocorrido, o que levou a rede à marcação  $M=[1 \ 0 \ 1]^T$  pelo disparo de  $t_2$ ). Quando a rede alcança a marcação  $M=[0 \ 1 \ 1]^T$ , os temporizadores das transições  $t_3$  e  $t_4$  são inicializados, desde que são as únicas transições habilitadas/receptivas aos eventos externos  $E^3$  e  $E^4$ , respectivamente. Nesta condição, a rede pára sua execução se houver a ocorrência do evento externo  $E^3$  antes do evento  $E^4$ , que a leva para a marcação  $M=[0 \ 0 \ 0]^T$ , ou retorna a marcação inicial  $M_0=[2 \ 0 \ 0]^T$ , caso ocorra o evento  $E^3$  antes do evento  $E^4$ , ou caso o temporizador da transição  $t_3$  ultrapasse o valor de 2 unidades de tempo, o que torna  $\pi_3=1$ , forçando seu disparo.



**Figura 3:** Rede de Petri Síncrona com Temporizadores e Funções de Habilitação de Transições.

Com a formalização da RPSTFHT, pode-se estruturar um supervisor para um SED modelado por uma RPST, utilizando sua árvore de alcançabilidade e uma especificação de comportamento. Para este caso, a partição do conjunto de transições é definida igualmente ao formalismo da TCS, isto é,

$$T = T_c \cup T_{uc} \quad (3)$$

(isto é, transições controláveis e transições não controláveis significando os eventos controláveis e os eventos não controláveis). Entretanto, define-se o conjunto de transições forçáveis

$$T_f \subseteq T_c, \quad (4)$$

que contém as transições nas quais as funções  $\pi$  podem ser definidas diferentes do valor padrão.

Deve-se observar, que tendo definidas as condições (2) e (3), na construção do modelo do sistema em RPST devem ser consideradas as seguintes situações de necessidade:

1. Necessariamente o modelo do sistema tem de ter transições forçáveis;
2. Para cada parte do sistema, deve existir pelo menos uma transição forçável, que garanta a continuidade da evolução dinâmica em casos de falhas nos sensores ou por outro problema inerente ao sistema, que gere ociosidade;
3. Os tipos de transições que podem ser forçáveis dentro do sistema e que devem existir nas partes isoladas do sistema são:
  - a. Transições do processo - ou seja, a própria transição que espera uma resposta do sensor (transição que está receptiva ao seu evento externo associado);
  - b. Transições que não levem o sistema para estados indesejados;
  - c. Transições de reinício de processo;
  - d. Transições de abandono do processo.

Partindo destas condições, o modelo terá sempre a possibilidade de descartar processos em andamento, ou reiniciá-los, ou continuá-los, ou mesmo, desviar situações indesejáveis. Essas condições são inerentes ao procedimento de modelagem de SEDs utilizando redes de Petri, como visto em Zhou e DiCesare [26]. Entretanto, aqui são utilizadas essas restrições específicas para a utilização das transições forçáveis, de forma a ser possível sintetizar o supervisor. Deve-se observar que este procedimento se apresenta como uma alternativa ao *watch-dog*, visto em [27], através das Redes de Petri temporais, desde que, neste caso aqui tratado, o tempo limite superior é pré-definido pelos próprios sensores que definem os tempos de referência, não sendo necessário temporizar as transições da rede [28].

Com o modelo do sistema, a especificação de comportamento, pode ser definida por meio de uma marcação a ser atingida, ou por meio de uma linguagem formal.

Por fim, a síntese da rede supervisora necessita da seguinte proposição:

**Proposição 1** *Dado um modelo de um sistema em RPST com transições forçáveis, através de sua árvore de*

*alcançabilidade modificada, pode-se construir um supervisor não bloqueável, cuja evolução é contínua mesmo havendo a não ocorrência de alguns eventos externos.*

**Demonstração:** *Pela abordagem de Barroso [17], garante-se o não bloqueio através da construção das funções de habilitação de transições. Por outro lado, para uma marcação  $M$ , se duas transições  $t \in T$  e  $t' \in T_f$  estão habilitadas e,  $t$  está receptiva a um evento externo  $E$ , ela disparará com a resposta do sensor que gera o evento  $E$  (ocorrência do evento externo  $E$ ). Entretanto, considerando que o sensor que gera esse evento externo tem um tempo limite máximo  $\tau_{ref}$  de sensibilidade para dar uma resposta à transição  $t$  da rede, a ultrapassagem deste limite ( $\tau > \tau_{ref}$ ) implica na possibilidade da ocorrência da transição  $t'$  que pode ser forçada. Caso  $t \in T_f$  e  $t'$  esteja receptiva ao evento externo  $E'$ , então tanto a transição  $t'$  pode disparar com a ocorrência do evento  $E'$  associado, como a transição  $t$  pode ser forçada a ocorrer. Para qualquer uma destas condições, sempre a condição de um evento externo não ocorrer em seu limite máximo, define uma falta no sensor, e dessa forma, o processo é continuado, impedindo o sistema de se tornar ocioso.*

Tendo as condições específicas para a construção do modelo utilizando as transições forçáveis, a estruturação do supervisor é fundamentada sobre o Algoritmo da Construção do Gerador da Suprema Sub-Linguagem Controlável (ACGS) de Barroso [17], e do Algoritmo para Construção do Processo de Decisão do Supervisor (ACPDS) que constrói as funções  $\pi$  das transições forçáveis do conjunto  $T_f$ . Este algoritmo é apresentado a seguir:

#### Algoritmo para Construção do Processo de Decisão do Supervisor (ACPDS)

- ii. Dados os valores de referência  $\tau_{ref}$  (limites máximos de resposta dos sensores), faça:
  - a. Para cada transição  $t_i$  habilitada em cada marcação  $M$ , tal que  $\phi_i = 1$ , faça:
    - i. Se  $t_i \in T_f$  e  $t_i$  é receptiva aos eventos externos  $E^{i1}, E^{i2}, \dots, E^{in}$ , faça
$$\pi_i = [\tau_i > \max \{ \tau_{ref,i1}, \tau_{ref,i2}, \dots, \tau_{ref,in} \}], \quad (5)$$
com  $\tau_{ref,in}$  sendo o tempo máximo para ocorrência do evento externo  $E^{ij}$ ,  $j=1,2,\dots,n$ ;
    - ii. Se  $t_i \notin T_f$  e  $\exists t_k | M [ t_k > \wedge \phi_k = 1 \wedge t_k \in T_f$ , faça:
      - I. Se  $t_k$  é receptiva a um evento externo  $E^k$ , então
$$\pi_k = [\tau_k > \tau_{ref,k}]; \quad (6)$$
      - II. Se existem  $E^{k1}, E^{k2}, \dots, E^{kn}$  associados à  $t_k$ , faça

$$\pi_k = [\tau_k > \max \{ \tau_{\text{ref},k1}, \tau_{\text{ref},k2}, \dots, \tau_{\text{ref},kn} \}]; \quad (7)$$

III. Se  $\nexists E^k$  associado à  $t_k$ , faça

$$\pi_k = [\tau_k > \tau_{\text{ref},i}]; \quad (8)$$

IV. Se existem  $t_{k1}, t_{k2}, \dots, t_{km}$  habilitadas em  $M$  e  $t_{kj} \in T_i$ ,  $j=1,2,\dots,m$ , então  $\pi_{kj}$  é definida pelas condições a.ii.I., a.ii.II. e a.ii.III.

Neste algoritmo, para cada transição habilitada em uma marcação  $M$ , a não ocorrência de um evento externo associado gera uma função de disparo forçado para a própria transição, se ela é forçável, forçando a continuidade do processo, ou para uma outra transição habilitada de acordo com as regras de disparos das RPFHTs, que desvia a execução do processo para uma outra atividade.

#### 4. EXEMPLO APLICATIVO DA ABORDAGEM PROPOSTA

Na Figura 4 é apresentado um simples sistema composto por um braço robótico - *br* (atuador montado sobre um parafuso infinito), três atuadores ( $A_1$ ,  $A_2$  e  $A_3$ ), um *buffer* de entrada (que recebe peças), um *buffer* de saída, um *buffer* de peças rejeitadas e três localizações específicas ( $L_1$ ,  $L_2$  e  $L_3$ ) para testar peças. Neste sistema, o braço robótico pode pegar peças no *buffer* de entrada e colocá-las nos lugares  $L_1$  e  $L_2$ , bem como movimentar as peças entre estes dois lugares de teste. O atuador  $A_1$  pode empurrar a peça do lugar  $L_1$  para o *buffer* de peças rejeitadas. O atuador  $A_2$  pode empurrar peças para uma posição intermediária entre  $L_1$  e o *buffer* de saída ( $L_3$ ) ou diretamente para o *buffer* de saída. O atuador  $A_3$  pode empurrar uma peça do lugar  $L_3$  para o *buffer* de peças rejeitadas. Por outro lado, os sensores que geram os eventos externos são:  $E^1$  no *buffer* de entrada para detectar se há uma peça,  $E^2$ ,  $E^7$  e  $E^8$  ao longo da base de movimentação do braço robótico para detectar seu posicionamento (*buffer* de entrada, lugar  $L_1$  ou lugar  $L_2$ ),  $E^3$  e  $E^4$  no lugar  $L_1$  para testar o tipo de peça que pode ser colorida ou metálica (aceitáveis) ou preta (rejeitável),  $E^5$  localizado no lugar  $L_2$  para testar se a peça é metálica e  $E^6$  localizada no lugar  $L_3$  para ver se a peça é colorida.

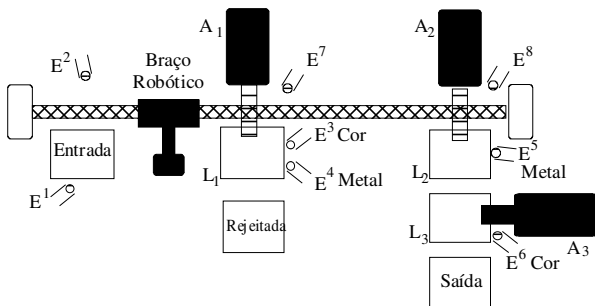


Figura 4: Sistema para modelar por uma RPST.

O modelo em RPST desse sistema é visto na Figura 5, cuja marcação inicial é

$$M_0 = [8 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T.$$

Neste modelo não há diferenciação entre as peças a serem trabalhadas no sistema (ao todo oito peças), cabendo aos sensores detectarem os tipos. Para este caso, considera-se que há três peças plásticas de cor branca, três peças de madeira de cor preta e duas peças metálicas.

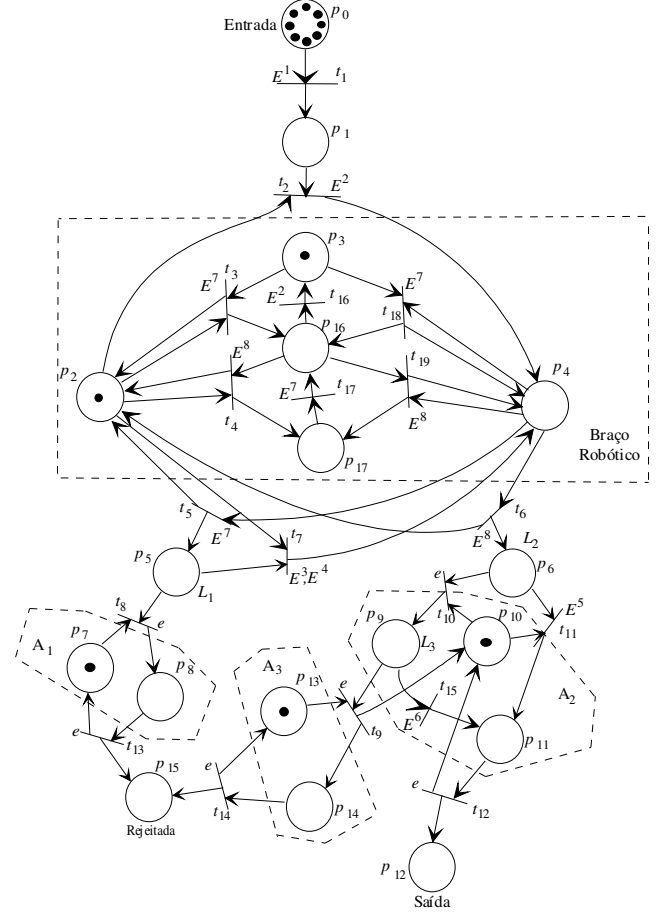


Figura 5: Modelo do sistema via RPST.

Neste sistema, são transições forçáveis as transições  $t_2$ ,  $t_5$ ,  $t_6$ , e  $t_7$ , (referentes ao braço robótico),  $t_8$  (referente ao atuador  $A_1$ ),  $t_{10}$ ,  $t_{11}$ ,  $t_{12}$  e  $t_{15}$  (referente ao atuador  $A_2$ ) e  $t_9$  (referente ao atuador  $A_3$ ). São transições não controláveis  $t_3$ ,  $t_4$ ,  $t_{12}$ ,  $t_{13}$ ,  $t_{14}$ ,  $t_{16}$ ,  $t_{17}$ ,  $t_{18}$  e  $t_{19}$ , que se referem às chegadas de peças aos *buffers* de saída/peças rejeitadas através dos atuadores. As demais (incluindo as forçáveis) são transições controláveis.

Para este sistema, é desejado que o braço robótico só pegue e libere peças nos devidos lugares, e que as peças rejeitadas sejam as peças plásticas de cor preta, em que o sistema deve trabalhar continuamente, mesmo com a falha de algum sensor. Ou seja, a marcação final a ser alcançada deve ser

$$M = [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 5 \ 1 \ 0 \ 3 \ 0 \ 0]^T,$$

em que se entende que após terminado o processamento das oito peças, o braço retorne a sua posição inicial e todos os atuadores estejam recuados com todas as peças plásticas de cor preta no lugar de peças rejeitadas e as demais na saída.

Neste sistema, os sensores geram os seguintes eventos externos:

- Sensor óptico: eventos externos  $E^3$  e  $E^6$ ;
- Sensor magnético: eventos externos  $E^2$ ,  $E^7$  e  $E^8$ ;
- Sensor capacitivo: evento externo  $E^1$ ;
- Sensor indutivo: eventos externos  $E^4$  e  $E^5$ .

Cada evento externo  $E^i$  tem um tempo de referência  $\tau_{ref\ i}$  significando o tempo máximo de sensibilização do respectivo evento, dados por:  $\tau_{ref\ 1}=10\text{ms}$ ;  $\tau_{ref\ 2}=\tau_{ref\ 7}=\tau_{ref\ 8}=2\text{ms}$ ;  $\tau_{ref\ 3}=\tau_{ref\ 6}=4\text{ms}$  e  $\tau_{ref\ 4}=\tau_{ref\ 5}=2,5\text{ms}$ .

Utilizando o AMARA, constrói-se a árvore de alcançabilidade, na qual nenhuma marcação é bloqueada. Entretanto, algumas marcações são não permitidas, como é o caso de marcações alcançadas que definem mais de uma peça em algum lugar de teste (*buffer* de entrada -  $p_1$ ,  $L_1$  -  $p_5$  e  $L_2$  -  $p_6$ :  $M(p_1)>1$ ,  $M(p_5)>1$  e  $M(p_6)>1$ , desde que cada lugar destes somente pode conter uma única peça, e o posicionamento do braço robótico sobre o lugar de entrada para pegar uma peça). Neste caso, através da execução do ACGS, encontram-se as funções de habilitação de transições definidas por:

$$\begin{aligned}\phi_1 &= [M(p_1) \leq 1], \\ \phi_2 &= [(M(p_3)=1) \wedge ((M(p_5)<1) \vee (M(p_6)<1))], \\ \phi_5 &= [(M(p_5) \Pi 1) \wedge ((M(p_{16})=1) \wedge (M(p_7)=1))], \\ \phi_6 &= [(M(p_6) \Pi 1) \wedge ((M(p_{17})=1) \wedge (M(p_{10})=1))], \\ \phi_8 &= 0.\end{aligned}$$

Por fim, com a utilização do ACPDS, encontram-se as funções de disparos forçados definidas por:

$$\begin{aligned}\pi_2 &= (\tau_2 > \tau_{ref2}), \\ \pi_5 &= (\tau_5 > \tau_{ref7}), \\ \pi_6 &= (\tau_6 > \tau_{ref8}), \\ \pi_7 &= (\tau_7 > \max\{\tau_{ref3}, \tau_{ref4}\}), \\ \pi_9 &= (\tau_9 > \tau_{ref6}), \\ \pi_{10} &= (\tau_{10} > \tau_{ref5}),\end{aligned}$$

onde  $\tau_{ref\ i}$  é o tempo máximo de sensibilização do sensor que gera o evento externo  $i$ .

Deve-se observar que na transição  $t_{11}$  não há uma função  $\pi_{11}$ , desde que a função  $\pi_{10}$  desvia o processo caso não haja a ocorrência do evento externo  $E^5$ . Da mesma forma, a transição  $t_{15}$  e a transição  $t_8$ . Por outro lado, a transição  $t_9$

tem sua função dada por  $\pi_9 = (\tau_9 > \tau_{ref6})$ , desde que quando o atuador estiver pronto, ele só pode empurrar a peça para o lugar de peças rejeitadas se o braço atuador  $A_2$  não empurra-la para a saída. De modo similar, a transição  $t_{10}$  pode ser avaliada. A rede supervisora tem a mesma estrutura do modelo e garante a execução contínua do sistema para a realização da tarefa especificada.

Agora, considerando que haja falhas em alguns sensores, é necessário avaliar o desempenho do sistema em relação à tarefa especificada.

Considerando que haja uma falha no sensor que gera o evento externo  $E^2$  (braço robótico chegar ao lugar de entrada), vê-se que a marcação do sistema garante a habilitação da transição  $t_2$  e da transição  $t_{16}$ . Se uma delas estiver habilitada e não houver resposta do sensor até o limite máximo de sensibilização, ela é forçada a disparar, determinando a continuidade do processo. Para este caso, o sistema alcança a marcação especificada.

Considerando que o sensor que falha no sistema seja o que gera o evento externo  $E^3$  (não ser detectada cor na peça) ou  $E^4$  (não detectar se a peça é metálica), o atuador  $A_1$  pode empurrar a peça para o lugar de peças rejeitadas. Entretanto, se o tempo máximo de resposta desses dois sensores for ultrapassado e o atuador  $A_1$  não empurrar a peça, o braço robótico retira-a do lugar  $L_1$  levando-a ao lugar  $L_2$ , onde novamente a peça é testada, se é metálica é empurrada para a saída, ou é levada à  $L_3$  que avalia se é de cor, só sendo empurrada para o lugar de peças rejeitadas se este último nada detectar.

Considerando ainda que  $E^5$  ou  $E^6$  falhem, as peças metálicas e de cor, respectivamente colocadas nestes lugares são levadas diretamente ao lugar de peças rejeitadas. Isto é, se  $E^5$  falha, uma peça metálica sempre é descartada e, se  $E^6$  falha, as peças de cor sempre são descartadas.

No geral, para este caso, o supervisor garante que o sistema não alcance estados não permitidos, bem como sua continuidade dinâmica com quase o mesmo tempo de execução tanto para o caso de todos os sensores estarem em perfeito estado, como para o caso de haver falhas em alguns sensores. Muito embora haja o descarte de peças exigidas na saída, o sistema não pára devido à falta de ocorrência de eventos externos. Este problema aqui encontrado pode ser solucionado se houver a mudança no sistema, com a inclusão de novos sensores.

## 5 CONCLUSÕES

Esse artigo introduz um formalismo para síntese de supervisores de sistemas a eventos discretos com processo de decisão temporal. O sistema a ser definido o controle é modelado por uma rede de Petri síncrona, a qual relaciona os disparos das transições com ocorrências de eventos externos gerados pelas sensibilizações de sensores do



sistema. Para a construção do supervisor são definidas as funções de habilitação das transições e as funções de disparos forçados das transições, as quais são construídas utilizando o procedimento de Barroso [17] (Algoritmo Modificado da Árvore de Alcançabilidade – AMArA – e Algoritmo para Construção do Gerador da Suprema Sublinguagem Controlável – ACGS) e o Algoritmo para Construção do Processo de Decisão do Supervisor (ACPDS) aqui apresentado. Este algoritmo gera funções de comparação temporal para temporizadores introduzidos nas transições da rede e forçam os disparos das transições quando os valores destes temporizadores ultrapassam os limites máximos de ocorrência dos eventos externos (sensibilização dos sensores), assegurando a continuidade do processo através de uma decisão tomada pelo supervisor. O procedimento apresentado garante a execução contínua do sistema sob controle, mesmo com a falha em alguns sensores [28]. O exemplo apresentado mostra a eficácia da metodologia proposta, o qual foi testado fisicamente, garantindo que o supervisor diferencia as peças testadas, o que se apresenta como uma vantagem na utilização deste procedimento, sem a necessidade do uso de redes de Alto Nível para a modelagem do sistema, ou da separação das informações via redes seguras. Deve-se observar que neste formalismo, a rede introduzida não apresenta temporização explícita, sendo um procedimento diferente da estrutura *watch-dog* [27], através das redes de Petri temporais.

## AGRADECIMENTOS

Os autores agradecem ao CNPq, à UFBA e ao CEFET-BA pelo suporte dado a esta pesquisa.

## REFERÊNCIAS

- [1] P.J.G. Ramadge and W.M. Wonham. Supervision of discrete event processes. *Proceedings of 21st Conference on Decision and Control*, pages 1228–1229, 1982.
- [2] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, USA, 1979.
- [3] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.
- [4] B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39(2):329–342, 1994.
- [5] F. Baccelli, G. Cohen, G.J. Olsder and J.P. Quadrat. Synchronization and Linearity. An Algebra for Discrete Event Systems. *John Wiley Sons*, 1992.
- [6] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, (126):183–235, 1994.
- [7] R. Alur. Timed automata. *Proc. NATO-ASI Summer School, Antalya, Turkey*, 1997.
- [8] S. Gaubert. Performance evaluation of (max,+) automata. *IEEE Transactions on Automatic Control*, 40(12):2014–2025, December 1995.
- [9] E.M.M.Costa. *Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados Utilizando Autômatos com Temporização Variável*. Anais do XIV Congresso Brasileiro de Automática (CBA2002), Natal, RN. Setembro de 2002.
- [10] E.M.M. Costa. *Síntese de Supervisores de Sistemas a Eventos Discretos Temporizados e Não Temporizados*. Tese de Doutorado. Universidade Federal da Paraíba – UFPB - Campus II, Campina Grande, Paraíba, Brasil. Novembro de 2001.
- [11] P. Chrétienne. *Les Réseaux de Petri Temporisés*. Thèse d'état. Université Paris VI, Paris. 1983.
- [12] L. Libeaut. *Sur l'utilisation des Dioïdes pour la Commande des Systèmes a Événements Discrets*. École Doctorale Sciences pour L'Ingenieur de Nantes. 1996.
- [13] G.V. Leandro. *Síntese de Controladores para Transitórios em Escalonamento de Job Shop Cíclico*. Tese de Doutorado. UNICAMP, Campinas, SP, Brasil. 2000.
- [14] K.X. He and M.D. Lemmon. *Petri Net Based Modeling and Analysis of Switched Systems*. Proc. of the 1998 IEEE, ISIC/CIRA/ISAS Joint Conference, Gaithersburg, MD. pp. 140-145. 1998.
- [15] P. Cofrancesco, A. Cristoforetti and R. Scattolini. *Petri Nets Based Approach to Software Development for Real-Time Control*. IEE Proceedings-D Control Theory and Applications, 138(5): 474-478. 1991.
- [16] V.P. Silva. *Uma Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos a partir do Modelo Temporizado*. Dissertação de Mestrado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil. 1999.
- [17] G.C. Barroso. *Uma Nova Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos*. Tese de Doutorado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil. 1996.
- [18] J.C.A. Figueiredo. *Redes de Petri com Temporização Nebulosa*. Tese de Doutorado. Universidade Federal da Paraíba, Campus II, Campina Grande, PB, Brasil. 1994.
- [19] M. Uzam, A.H. Jones and N. Ajlouni. *Conversion of Petri Nets Controllers for Manufacturing Systems Into Ladder Logic Diagrams*. Intelligent Machinery Division, Research Institute for Design, Manufacture and Marketing, University of Stanford, UK. 1996.
- [20] G. Frey. *Automatic Implementation of Petri Net Based Control Algorithms on PLC*. Proc. of the American Control Conference, Chicago, Illinois. June, pp. 2819-2823, 2000.
- [21] G. Frey and L. Litz. *Correctness Analysis of Petri Net Based Logic Controllers*. Proc. of the American Control Conference, Chicago, Illinois, June, pp. 3165-3166, 2000.
- [22] J-L. Chirn and D.C. McFarlane. *Petri Nets Based Design of Ladder Logic Diagrams*. Control 2000, Cambridge, UK. September, 2000.

- [23] A.H. Jones, M. Uzam and N. Ajlouni. *Design of Discrete Event Control Systems For Programmable Logic Controllers Using T-Timed Petri Nets*. Proc. of the 1996 IEEE Int. Symp. on Comp.-Aided Control System Design. pp. 212-217, September, 1996.
- [24] R. Hilal and P. Ladet. *Synchronous Petri Nets: Formalization and Interpretation*. Laboratoire d'Automatique de Grenoble, ENSIEG/INPG. 1994.
- [25] R. Hilal and P. Ladet. *A Reactive Real-Time Systems Modelling and Programming Aid Tool Based on the Synchronous Petri Nets Formalism*. 0-8186-4250-5/93-IEEE. pp.34-43. 1993.
- [26] M.C. Zhou and F. DiCesare. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publishers. 1993.
- [27] J. Cardoso e R. Valette. *Redes de Petri*. Editora da UFSC, Florianópolis, Santa Catarina. 1997.
- [28] G.B. da S. Góes, *Uma Abordagem para a Síntese de Supervisores de Sistemas a Eventos Discretos Utilizando Redes de Petri Síncronas*, Dissertação de Mestrado. Universidade Federal da Bahia, 2003.

## DADOS BIOGRÁFICOS

GILBERTO BISPO DA SILVA GÓES nasceu em Salvador, BA. Formou-se em Engenharia Mecânica pela UFBA em 1986. É professor do CEFET-BA desde 1987. Obteve o título de Mestre em Engenharia Elétrica pela UFBA na área de Automação Industrial com ênfase em Sistemas a Eventos Discretos em maio de 2003.

EDUARD MONTGOMERY MEIRA COSTA nasceu em Esperança, PB. Formou-se em Engenharia Elétrica pela UFPB, em 1995. Obteve os títulos de Mestre e Doutor em Ciências no domínio da Engenharia Elétrica também pela UFPB em 1997 e 2001, respectivamente. Atualmente é bolsista DCR/CNPq na UFBA, atuando na área de Sistemas a Eventos Discretos.