



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA

DAVI ALEXANDRE PAIVA

FORTALEZA

2024

DAVI ALEXANDRE PAIVA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Wilkley Bezerra Correia (Orientador)
Universidade Federal do Ceará (UFC)

DAVI ALEXANDRE PAIVA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Engenharia Elétrica.

Orientador: Prof. Dr. Wilkley Bezerra Correia.

FORTALEZA

2024

RESUMO

Atualmente o controle e modelagem de sistemas industriais, envolvem um controle supervisorio com processos a eventos discretos e também o controle entre os agentes envolvidos na planta para realizar determinado processo. O presente trabalho visa desenvolver uma técnica de controle e modelagem para sistema multiagentes, em que a modelagem é feita através da rede de Petri e o controle através do controle por consenso. A modelagem dos eventos ao longo do sistema são feitos através da rede de petri, com transições e lugares. O controle é feito através do algoritmo de consenso de multiagentes em que é feita a sincronia dos agentes da planta para diferentes pontos de operação alternados pela rede de petri. A técnica de controle e modelagem é apresentada através da aplicação em um cenário de sincronia e formação de autômatos em uma trajetória definida, em que a ordem e a forma de organização dos autômatos é alterada ao longo de eventos modelados pela rede de petri. É possível perceber que a técnica de utilização conjunta da modelagem em redes de petri com o controle por consenso possui uma boa performance e abstração em termos de modelagem de um sistema envolvendo muitos agentes com diferentes eventos, assim como o uso de controle por consenso auxilia de forma eficiente a sincronia entre os diversos agentes industriais.

Palavras-chave: Controle por Consenso; Redes de Petri; Controle multiagente; Plantas industriais, Sistemas Híbridos.

LISTA DE FIGURAS

Figura 1 – Exemplo de nós e arestas em um grafo	9
Figura 2 – Exemplo de um grafo	11
Figura 3 – Árvore de escopo de um sistema a eventos discretos	16
Figura 4 – Exemplo paralelismo Síncrono	19
Figura 5 – Exemplo paralelismo Assíncrono	19
Figura 6 – Exemplo de Caminhos alternativos e Repetição	20

LISTA DE TABELAS

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Organização do trabalho	8
2	SISTEMA MULTIAGENTES	9
2.1	Teoria dos Grafos	9
2.2	Teoria algébrica dos grafos e consenso do controle cooperativo	9
2.2.1	<i>Representação matricial dos grafos</i>	<i>10</i>
2.2.2	<i>Matriz de Grafo Laplaciana</i>	<i>10</i>
2.3	Consenso com Integrador único	12
2.3.1	<i>Protocolo de controle distribuído para o consenso</i>	<i>12</i>
2.3.2	<i>Consenso com líder</i>	<i>13</i>
2.3.3	<i>Consenso para nós com estados como vetores</i>	<i>13</i>
2.3.4	<i>Movimento invariante para consenso de primeira ordem</i>	<i>14</i>
3	REDES DE PETRI	15
3.1	Modelagem de Sistemas	15
3.1.1	<i>Modelagem de sistemas a eventos discretos</i>	<i>16</i>
3.2	Representação em máquina de estados	16
3.2.1	<i>Modelagem de processos sequenciais</i>	<i>17</i>
3.2.1.1	<i>Representação com refinamentos sucessivos</i>	<i>17</i>
3.2.1.2	<i>Explosão combinatória</i>	<i>17</i>
3.2.1.3	<i>Não-independência de submáquinas e bloqueio</i>	<i>18</i>
3.3	Modelagem utilizando Rede de Petri	18
3.3.1	<i>Evolução síncrona e assíncrona</i>	<i>19</i>
3.3.2	<i>Caminhos alternativos e Repetição</i>	<i>19</i>
3.4	Redes de Petri colorida	20
4	SIMULAÇÃO	21
5	CONCLUSÃO	22
	REFERÊNCIAS	23

1 INTRODUÇÃO

Um sistema industrial é composto por sensores, atuadores, sinalizadores, controladores entre outros componentes voltados para a realização de determinada cadeia de processos dentro de uma linha de produção. Tal que para realizar determinado processo é necessário uma sincronia entre diversos equipamentos, sensores e atuadores ao longo da planta industrial. Além do desafio de sincronizar uma gama de processos, os sistemas modernos possuem a necessidade de adaptar-se a novas variações e configurações, abrindo espaço para máquinas e sistemas com programação mais robusta e reconfigurável.

Dado este desafio, as redes de Petri coloridas se oferecem como uma ótima ferramenta de modelagem para os sistemas modernos de manufatura em linha de produção em que há um aumento da versatilidade e flexibilidade da estrutura e também a necessidade de uma programação com alto nível de abstração. (WENZELBURGER; ALLGÖWER, 2019)

De acordo com (GIUA; SILVA, 2017), as redes de Petri têm sido consideradas com um modelo adequado para um controle supervisão com o objetivo de abranger uma grande classe de problemas e explorar a análise algébrica necessária para otimização. Tratando-se também da análise para a planta não alcançar determinadas marcações indesejadas;

As redes de Petri também são uma ferramenta de modelagem inicial para o algoritmo de programação com ferramentas intrínsecas que analisam o algoritmo para evitar que o sistema entre em exceções, (LEE *et al.*, 2006)

Para modelagem e controle desse sistema composto por vários agentes, escolheu-se a abordagem por redes de Petri coloridas. As redes de Petri coloridas são uma ferramenta gráfica e matemática que se adaptam bem a um grande numero de aplicações, tais como protocolos de comunicação, controle de oficinas de fabricação.

A complexidade dos sistemas, em particular o de fabricação automatizada, leva a uma decomposição de vários níveis de controle, tais como planejamento, escalonamento, coordenação global, coordenação de sub-sistemas e controle direto (autômatos programáveis conectados aos sensores e aos atuadores). (CARDOSO; VALLETE, 1997)

A utilização de redes de petri como camada de abstração para tomada de decisões, escolha de estratégias diante dos problemas e organização dos agentes para seguir um determinado plano foi trabalhada como framework por (EBADI *et al.*, 2010), em que todo o processo de mais alto nível de decisão foi modelado para a rede de petri, nesse presente trabalho além da abstração pela rede de petri como tomada de decisão também é apresentando o algoritmo

de consenso como solução do problema de controle e cooperação entre os agentes dada uma tomada de decisão de organização dos grupos.

1.1 Organização do trabalho

Para a estruturação do presente trabalho, adota-se a seguinte metodologia de estudo

1. **Introdução:** Este capítulo contém as premissas básicas de estudo e evolução dos temas recorrentes na área de Controle Multiagentes. Ainda incluem-se os princípios básicos de apresentação do projeto, tais como os objetivos, a justificativa e a motivação do estudo.
2. **Sistema Multiagentes:** Partindo-se do princípio mais básico relacionado a modelagem de sistemas multiagentes. Assim, definem-se as representações matemáticas e gráficas de um sistema multiagente assim como técnicas de controle Cooperativo. Por fim são repassadas as principais técnicas de modelagem, representação em grafos e controle de sistemas multiagentes.
3. **Redes de Petri:**
4. **Simulação:**
5. **Conclusão:** Por fim, este último capítulo trata das considerações gerais sobre os conceitos apresentados e uma discussão crítica acerca dos resultados de simulação.

2 SISTEMA MULTIAGENTES

2.1 Teoria dos Grafos

No estudo da interação e comportamento entre sistemas dinâmicos, as interconexões entre os agentes e o fluxo de informações formam uma rede de comunicação. Essa rede é modelada através da teoria dos grafos em que cada sistema é representado por um nó, também chamado de agente.

Um grafo é um par $G = (V, E)$, tal que $V = \{v_1, v_2, \dots, v_N\}$ é um conjunto de N nós ou vértices e E um conjunto de vetores ou arcos. Um elemento pertencente a E é um par (v_i, v_j) tal que é um vetor que liga v_i à v_j , e é representado como uma flecha em que a cauda está em v_i e a ponta em v_j como demonstrado na figura 1.

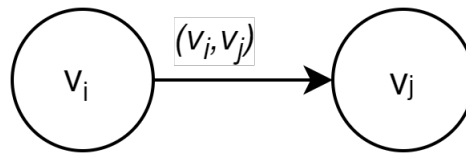


Figura 1 – Exemplo de nós e arestas em um grafo

Os graus de liberdade de entrada de um dado nó v_i é definido como o número de vetores que a ponta da flecha se encontra em v_i . Do mesmo modo, os graus de liberdade de saída de um nó v_i é dado pelo número de vetores que em v_i se encontra a cauda da flecha.

Associado à cada aresta de um elemento em $E = (v_i, v_j)$ tem-se um peso $a_{ij} > 0$. O peso a_{ij} representa a força de interação entre os nós v_i e v_j . De modo que quanto maior o peso maior a influência tem o comportamento do agente j sobre o agente i . Um grafo é dito bidirecional se e somente se $a_{ij} \neq 0$ e $a_{ji} \neq 0$, então tem-se que a comunicação entre agentes flui bidirecionalmente. Um grafo é dito unidirecional se $a_{ij} = a_{ji}$, para qualquer i e j .

2.2 Teoria algébrica dos grafos e consenso do controle cooperativo

O controle cooperativo estuda a dinâmica de sistemas dinâmicos com múltiplos agentes com iterações um com o outro através de uma comunicação em grafo. O grafo representa as iterações e comunicações entre agentes. O objetivo do controle cooperativo é garantir a sincronia entre o comportamento e estados dos agentes, de modo que para cada agente só é disponível que as informações sejam entre o agente com os agentes vizinhos.

2.2.1 Representação matricial dos grafos

A estrutura e propriedade dos grafos podem ser estudadas examinando as propriedades de certas matrizes associadas. Dados os pesos a_{ij} associados, o grafo pode ser representado pela **Matriz de Adjacência** ou conectividade $A = [a_{ij}]$, com $a_{ij} > 0$ se $(v_j, v_i) \in E$ e a_{ij} caso contrário. Define-se duas propriedades locais dos grafos, o graus de entrada e os graus de saída. Os graus de entrada de um nó v_i é definido pela equação 2.1, tal que d_i é o somatório dos pesos a_{ij} da linha i -th.

$$d_i = \sum_{j=1}^N a_{ij} \quad (2.1)$$

Os graus de saída de um nó v_i é definido pela equação 2.2, tal que d_i é o somatório dos pesos a_{ij} da coluna i -th.

$$d_i^0 = \sum_{j=1}^N a_{ji} \quad (2.2)$$

Define-se duas propriedades globais dos grafos, o diâmetro do grafo, dada pela maior distância entre dois nós e o volume de entrada (*in*) – *volume*, dado pela soma dos nós de entrada

$$VolG = \sum_i d_i \quad (2.3)$$

2.2.2 Matriz de Grafo Laplaciana

Uma definição importante aplicada a sistemas multiagentes é a **Matriz Laplaciana**, que auxilia o estudo das propriedades da dinâmica do grafo de multiagentes. A mesma é obtida através da operação entre duas matrizes, a Matriz Diagonal e a Matriz de Adjacência. Define-se a matriz diagonal de graus de entrada, pela equação 2.4, em que para um agente i , tem-se o elemento d_i como o somatório das flechas que apontam para o dado agente.

$$D = diag\{d_i\} \quad (2.4)$$

Por fim a matriz Laplaciana L é definida como $L = D - A$, tal que D é a matriz Diagonal e A é a matriz de Adjacência.

Um exemplo de matriz laplaciana associada ao grafo pode ser ilustrada através da figura 2, em que a matriz Diagonal D é dada pela matriz 2.5.

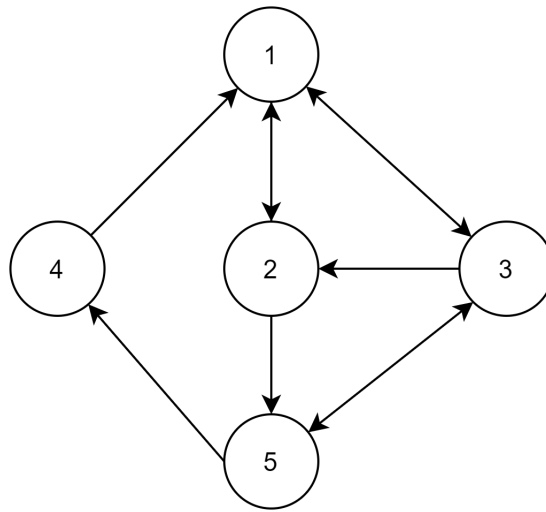


Figura 2 – Exemplo de um grafo

$$D = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (2.5)$$

A matriz A, dada pela Matriz de Adjacência 2.6.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (2.6)$$

Por fim, a matriz Laplaciana é dada por $L = D - A$.

$$L = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & -1 & 2 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (2.7)$$

2.3 Consenso com Integrador único

Para o estudo inicial de controle cooperativo, tem-se a análise de um sistema multi-agente formado por agentes i com dinâmica dada por um integrador escalar único, modelada pela equação 2.8

$$\dot{x}_i = u_i \quad (2.8)$$

com $x_i, u_i \in \mathbb{R}$. Isso corresponde que cada nó do grafo G , possui um agente com memória.

2.3.1 Protocolo de controle distribuído para o consenso

Para cada agente i , considere o protocolo de controle local dado pela equação 2.9

$$u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \quad (2.9)$$

com a_{ij} sendo o peso de interação entre os estados dos agentes. Essa equação é conhecida como protocolo de votação local, em que o estado de cada agente depende tão somente do estado do agente vizinho, e a entrada de controle depende da diferença dos estados em relação aos agentes vizinhos. De modo que percebe-se que se todos os estados forem os mesmo a entrada de controle tende a zero $\dot{x}_i = u_i = 0$.

Para a dinâmica de integrador único, é desejável que a equação 2.8, resolva o problema de consenso, da dinâmica de malha fechada dada pela equação 2.10

$$\dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \quad (2.10)$$

Reorganizando a equação 2.10, tem-se que a equação. 2.11

$$\begin{aligned} \dot{x}_i &= -x_i \sum_{j \in N_i} a_{ij} + \sum_{j \in N_i} a_{ij}x_j \\ &= -d_i x_i + [a_{i1} \quad \cdots \quad a_{iN}] \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \end{aligned} \quad (2.11)$$

tal que d_i são os graus de liberdade, $x = [x_1 \cdots x_N] \in \mathbb{R}^N$ o vetor de estados. Define-se a matriz D como matriz diagonal formada por $D = \text{diag}\{d_i\}$, organiza-se a dinâmica global, através da matriz dada pela equação Laplaciana.

$$\begin{aligned} \dot{x} &= -Dx + Ax = -(D - A)x \\ \dot{x} &= u = -Lx \end{aligned} \quad (2.12)$$

Através da equação 2.12, e da matriz laplaciana de grafo tem-se que a dinâmica de malha fechada pode ser analisada através da matriz laplaciana dada. Para a dinâmica de integrador único tem-se que se e somente se o grafo possui a topologia de spanning tree, então todos os estados dos nós vão a um valor de consenso dado por $x_i = x_j = c$. O valor de consenso é dado pela equação 2.13.

$$c = \sum_{i=1}^N p_i x_i(0) \quad (2.13)$$

tal que $w_1 = [p_1 \cdots p_N]^T$, é o vetor normalizado pela esquerda da matriz laplaciana L , para $\lambda_1 = 0$. De modo que a constante de tempo é dada por 2.14 e λ_2 sendo o segundo autovalor da matriz L .

$$\tau = 1/\lambda_2 \quad (2.14)$$

2.3.2 Consenso com líder

Um "(directed) tree" é um grafo onde todo nó exceto um é chamado de líder, e possui grau de entrada unitário. De modo que todos os outros nós possuem um consenso liderado pelas condições iniciais do líder. O valor de consenso é dado pela equação 2.13, de modo que p_i é o i -th componente para o autovetor pela esquerda de w_i para $\lambda_1 = 0$, tal consenso é na verdade a média ponderada das condições iniciais das raízes dos nós ou do líder em um grafo.

2.3.3 Consenso para nós com estados como vetores

Nas condições de integrador único e integrado duplo apresentadas anteriormente os estados são tidos como escalares, para os exemplos em que os estados são vetores tais como $x_i, u_i \in R^N$ tem-se que os vetores globais de estados e controle são respectivamente $x = [x_1^T \cdots x_N^T]^T \in R^{nN}$, $u = [u_1^T \cdots u_N^T]^T \in R^{nN}$ e os elementos dados pelos pesos de consenso a_{ij} e a diagonal d_i são multiplicados pela matriz identidade I_n , de modo que a dinâmica global do sistema é dada pela equação 2.15

$$\begin{aligned} u &= -(L \otimes I_N)x \\ \dot{x} &= -(L \otimes I_N)x \end{aligned} \quad (2.15)$$

dado que \otimes é definido como produto de kronecker.

2.3.4 Movimento invariante para consenso de primeira ordem

Para o protocolo de primeira ordem local dado por 2.12 foi garantido que para a topologia de *spannigtree* o consenso é alcançado.

3 REDES DE PETRI

3.1 Modelagem de Sistemas

Para a modelagem de sistemas é importante entender as principais características do sistema que está sendo estudado, no propósito de uma revisão na classificação de sistemas, foi elencado em (CASSANDRAS; LAFORTUNE, 2021) tais classificações, observe que elas não são necessariamente auto exclusivas:

Sistema Dinâmico e Estático: Em sistema estático a saída é sempre independente dos valores de entrada passados, já em um sistema dinâmico os valores de saída dependem dos valores anteriores da entrada. Equações diferenciais ou a diferença são geralmente requeridas para representar o comportamento de sistemas dinâmicos.

Sistema Variante e Invariante no Tempo: Em um sistema invariante no tempo o comportamento do mesmo não se altera com o passar do tempo. Tal sistema é dado com estacionário pois se aplicada a mesma entrada ao longo do tempo o sistema responderá com o mesmo comportamento.

Sistemas Lineares e Não Lineares: Um sistema linear obedece a condição $g(a_1u_1 + a_2u_2) = a_1g(u_1) + a_2g(u_2)$ tais que u_1 e u_2 são dois vetores de entrada, a_1 e a_2 dois números reais e $g(.)$ é a saída resultante.

Sistema em Estado Contínuo e Discreto: Um sistema contínuo pode ter a variável de estado representa por um número real ou complexo, já o sistema discreto é representado por uma série de elementos discretos.

Sistema Orientado a Eventos e Tempo: Em um sistema orientado ao tempo o estado do sistema muda com a mudança de tempo, já a eventos a mudança de estado ocorre de forma instantânea a partir da ocorrência de um evento.

Sistema Determinístico e Estocástico: Um sistema se torna estocástico quando possui uma ou mais de uma variável estocástica. Nesse caso o estado do sistema é descrito por um processo estocástico.

Sistema em Tempo Contínuo e Discreto: Um sistema em tempo contínuo possui todas as entradas definidas para todos os valores de tempo possíveis.

No escopo do trabalho proposto o a modelagem em redes de petri é utilizada para um sistema a eventos discretos seguindo a seguinte árvore de escopos dada pela figura 3

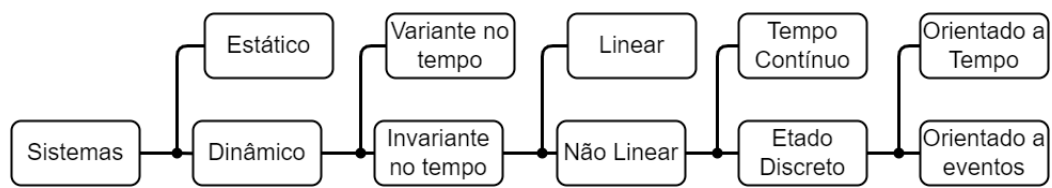


Figura 3 – Árvore de escopo de um sistema a eventos discretos

3.1.1 Modelagem de sistemas a eventos discretos

Em si tratando da modelagem de sistemas a eventos discretos tem-se três elementos principais utilizados na modelagem eventos (instantes e mudanças de estados), atividades (evolução do sistema físico entre dois eventos) e processos (sequência de eventos e atividades no sistema).

Na evolução dos processos no sistema os processos podem ocorrer de forma totalmente independente entre si, enquanto outras atividades necessitam de uma determinada sincronização ou até uma sequência de eventos prévios. Uma forma de diferenciação de interações entre processos é apresentada por (CARDOSO; VALLETE, 1997), como:

Cooperação: Os processos convergem para um objetivo comum, de modo que anteriormente há uma relativa independência antes do ponto de sincronização.

Competição: Os processos necessitam de um dado recurso, caso esse recurso seja abundante para todos os processos, dados processos poderiam ser descritos de forma independente, caso contrário faz-se necessário o compartilhamento de recursos envolvendo uma exclusão mútua a partir de um ponto de sincronização.

Pseudo-Paralelismo: O paralelismo é apenas aparente e os eventos por mais que sejam independentes nunca serão simultâneos pois são acionados por um relógio comum, a exemplo de um sistema operacional que por mais que processe várias tarefas, porém o processador só processa um ciclo de instrução por vez.

Paralelismo Verdadeiro: Os eventos podem ocorrer de forma simultânea, não existindo uma escala de tempo em comum, a exemplo de vários processadores operando tarefas distintas.

3.2 Representação em máquina de estados

Uma das representações mais clássicas para modelagem de sistemas à eventos discretos é a máquina de estados, para o caso de uma número de estados finitos enumera-se

os possíveis estados e descreve-se os eventos referente as mudanças de estado, descrevendo-se assim cada estado a a partir do estado anterior.

O modelo matemático para a máquina de estado finita é dada a partir da equação 3.1, em que E é um conjunto finito de estados, dado pelo estado inicial E_0 , um alfabeto de entrada A , e uma função de transição de estados θ , dado por $\theta : E \times A \rightarrow E$, associando cada par de estado-entrada ao próximo estado.

$$M = (E; A; \theta; E_0) \quad (3.1)$$

De acordo com (CARDOSO; VALLETE, 1997) este modelo explicita a noção de eventos e parcialmente a de atividade, não explicitando, entretanto, a noção do processo com as evoluções simultâneas de diversos processos paralelos, de modo que uma máquina de estado finita descreve apenas um único processo sequencial.

3.2.1 Modelagem de processos sequenciais

Para a descrição de vários processos sequências, uma das soluções é representar o sistema por um conjunto de máquinas de estados finitos. Quando as máquinas de estados são independentes, esse modelo se aplica sem dificuldade, porém quando existe competição ou cooperação entre os processos faz-se necessário o uso de processos sequenciais comunicantes. A sincronização é descrita através da intervenção na função de transição de estados θ de uma máquina.

3.2.1.1 Representação com refinamentos sucessivos

Um dos contrapontos dessa abordagem é que independente do método utilizado, a representação das comunicações entre as máquinas é diferente da representação interna da sequência de uma máquina. Portanto, tal abordagem não é compatível com a abordagem top-down de refinamento sucessivos. É necessário desde do início da modelagem a escolha de uma decomposição que não será colocada em causa a posteriori.

3.2.1.2 Explosão combinatória

Outro ponto de análise dessa forma de modelagem é que para cada informação partilhada entre as máquina ou troca de sinais entre as máquinas é necessário analisar o compor-

tamento global do sistema através do recálculo de uma nova máquina de estado que descreva o sistema de forma global. Neste caso, ocorre a problemática da explosão combinatória do número de estados definida pela relação entre k máquinas e n estados, produzindo uma máquina de n^k estados, ocorrendo uma explosão combinatória a medida que k e n aumentam.

3.2.1.3 *Não-independência de submáquinas e bloqueio*

Em se tratando de sistemas com paralelismo um dos problemas comuns que podem acontecer é o de bloqueio (dead-lock) em que a máquina de estado não consegue evoluir pois depende da transição de um estado de outra máquina que por sua vez encontra-se igualmente no mesmo estado de espera. Existem técnicas na teoria de máquinas de estados finitos que evitam o bloqueio, porém a estrutura do sistema acaba sendo comprometida, perdendo assim a representativa clara do sistema e suas transições.

3.3 **Modelagem utilizando Rede de Petri**

A rede de Petri é uma ferramenta gráfica e matemática para modelagem e controle de sistemas à eventos discretos, dado que o sistema a ser escolhido é um sistema que pode ser modelado através de tal ferramenta, com o intuito de obter uma visualização gráfica do processo, implementar lógica de controle e sincronismo, analisar propriedades da rede entre outras.

A descrição dos eventos e transições do sistema é dada através da rede de petri pelos lugares, fichas e transições, para representar os "estados" do sistema são utilizados os lugares, já as transições movimentam os recursos, ou seja, as fichas de um lugar para outro, dada a condição que a transição só possa ser disparada caso os lugares a ela ligada estejam completas com os recursos requisitados, fazendo assim interdependências em que um determinado estado só pode ser alcançado caso determinadas condições satisfeitas.

O comportamento dinâmico do sistema se dá através do disparo das transições, evento que faz com que o sistema passe do estado atual para o próximo estado. Tal disparo consiste em duas etapas, a primeira de retirar as fichas dos lugares de entrada e por fim depositar as fichas em cada lugar de saída.

3.3.1 Evolução síncrona e assíncrona

A rede de petri pode representar sistemas com eventos síncronos e assíncronos, em que são necessários momento de espera para acontecer determinados eventos assim como a independência de eventos que podem ou não ocorrer de forma simultânea.

Observa-se na figura 4 um evento caracterizado como divisão em que no disparo da transição t_2 uma ficha é retirada de p_1 e simultaneamente é colocada uma ficha em p_3 e p_4 daí em diante a evolução do sistema ocorre de forma assíncrona podendo ou não haver disparos concorrentes.

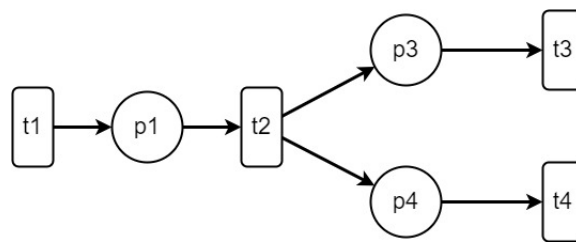


Figura 4 – Exemplo paralelismo Síncrono

Observa-se na figura 5 um evento caracterizado como junção em que para haver o disparo de t_3 é necessário que haja uma ficha tanto em p_3 quanto em p_4 , o consumo dessas fichas ocorre de forma síncrona tal evento implica necessariamente de uma espera em que ou p_3 espera a chegada do recurso em p_4 ou o contrário, garantindo assim que p_5 só receba uma ficha quando essas duas condições forem satisfeitas. Antes do disparo de t_3 o sistema pode evoluir de forma assíncrona com disparos independentes de t_1 e t_2 .

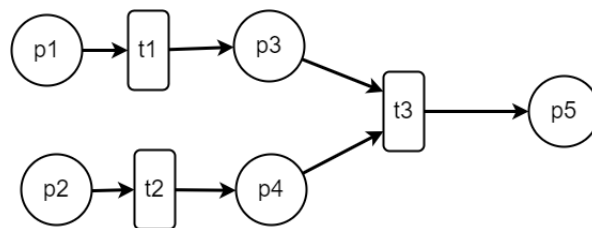


Figura 5 – Exemplo paralelismo Assíncrono

3.3.2 Caminhos alternativos e Repetição

Um evento que também pode ser modelado são o de caminhos alternativos e variações de sequência de disparos em uma rede de petri, em que em determinado momento na rede há um lugar ligado a entrada de uma ou mais transições, ocorrendo assim a sensibilização de tais

transições, podendo ocorrer portanto o disparo de qualquer uma das transições, tomando-se assim um caminho alternativo caso outra transição fosse disparada.

Observa-se esse fenômeno na modelagem descrita pela figura 6, em que no momento em que p_1 recebe uma ficha é sensibilizada as transições t_1 e t_2 , de modo que a rede de petri não restringe a escolha de um ou outra transição, porém caso uma transição seja disparada a outra não poderá ser, ocorrendo assim uma competição pelo recurso em que a transição que disparar primeiro recebe o recurso. Dado a ocorrência da repetição da rede através de t_5 , novamente a tomada de decisão entre t_1 e t_2 ocorrerá, podendo assim ocorrer um caminho alternativo ao caminho prévio dada a sequência de disparo ocorrida.

A figura 6, também modela um evento de repetição através da transição t_5 , em que dada a chegada da ficha no lugar p_5 tal ficha pode retornar ao lugar de origem p_1 ocorrendo assim o restabelecimento da rede e infinitas repetições.

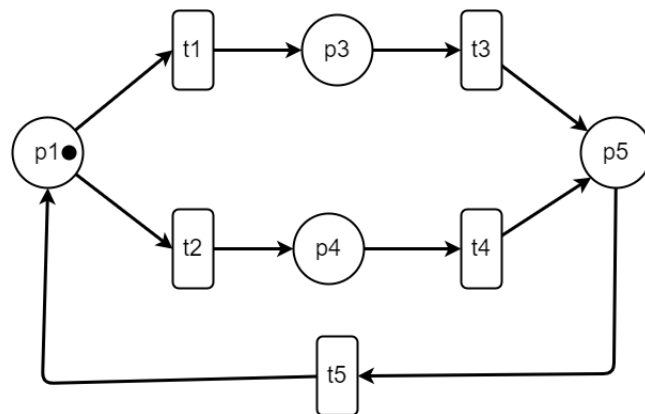


Figura 6 – Exemplo de Caminhos alternativos e Repetição

3.4 Redes de Petri colorida

Para modelagem e controle desse sistema composto por vários agentes, escolheu-se a abordagem por redes de Petri coloridas.

As redes de Petri coloridas são uma ferramenta gráfica e matemática que se adaptam bem a um grande numero de aplicações, tais como protocolos de comunicação, controle de oficinas de fabricação.

4 SIMULAÇÃO

5 CONCLUSÃO

O método de modelagem e controle proposto foi realizado no problema de uma planta industrial envolvendo a sincronia e formação de um grupo de autômatos ao longo de trajetórias pré definidas, em que a partir de alguns eventos modelados pela rede de Petri foi alterado a formação do grupo de autômatos assim como os pontos de sincronia. O método se apresentou como uma técnica viável e eficiente, pois para aplicações de sistemas com muitos agentes o controle por consenso se apresenta como uma implementação simples sem grande uso de recurso matemático que fornece a sincronia necessária para aplicação de formação ordenada do grupo de autômatos. No ponto de vista de robustez e adaptabilidade do sistema, observou-se que cada agente respeita as limitações dos agentes vizinhos, seja ela de posição de velocidade, evitando assim colisões e independente da mudança da dinâmica de um agente todo o sistema tem sua dinâmica adaptada, trazendo assim uma sincronia entre os diferentes agentes com diferentes dinâmicas ao longo do sistema. A principal contribuição desse trabalho é a técnica conjunta de modelagem e controle que diminui o processamento local em cada agente deixando assim as lógicas de processamento centralizada em um sistema supervisor modelado via rede de petri, além de uma lógica de controle de baixo custo computacional, todavia é necessário uma ótima comunicação entre os agentes, pois a base do controle é dada pela sincronia entre os estados do agente vizinho.

REFERÊNCIAS

CARDOSO, J.; VALLETE, R. **Redes de Petri**. Florianópolis, SC: [S. n.], 1997.

CASSANDRAS, C. G.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. 3. ed. Springer Cham, 2021. XXVI, 804 p. ISBN 978-3-030-72272-2 Hardcover. Disponível em: <https://doi.org/10.1007/978-3-030-72274-6>.

EBADI, T.; PURVIS, M.; PURVIS, M. A framework for facilitating cooperation in multi-agent systems. **The Journal of Supercomputing**, v. 51, n. 3, p. 393–417, 03 2010. ISSN 1573-0484. Disponível em: <https://doi.org/10.1007/s11227-009-0372-8>.

GIUA, A.; SILVA, M. Modeling, analysis and control of discrete event systems: a petri net perspective. **IFAC-PapersOnLine**, v. 50, n. 1, p. 1772–1783, 2017. ISSN 2405-8963. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896317302793>.

LEE, J.; CHOI, M.; SUNG, C. A modeling technique applying an object-oriented petri net for embedded system. In: **2006 International Conference on Hybrid Information Technology**. [S. l.: s. n.], 2006. v. 2, p. 642–645.

WENZELBURGER, P.; ALLGÖWER, F. A petri net modeling framework for the control of flexible manufacturing systems. **IFAC-PapersOnLine**, v. 52, n. 13, p. 492–498, 2019. ISSN 2405-8963. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896319310596>.