



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

MODELAGEM E SIMULAÇÃO DE UMA PLANTA INDUSTRIAL BASEADA EM
REDES DE PETRI COLORIDA E CONTROLE COOPERATIVO

FORTALEZA

2024

DAVI ALEXANDRE PAIVA

MODELAGEM E SIMULAÇÃO DE UMA PLANTA INDUSTRIAL BASEADA EM REDES
DE PETRI COLORIDA E CONTROLE COOPERATIVO

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

DAVI ALEXANDRE PAIVA

MODELAGEM E SIMULAÇÃO DE UMA PLANTA INDUSTRIAL BASEADA EM REDES
DE PETRI COLORIDA E CONTROLE COOPERATIVO

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)

À minha esposa Stephanie Paiva por todo o seu
amor, cuidado e dedicação.

AGRADECIMENTOS

A Deus por me guia e dar forças para chegar até aqui.

Ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará pela oportunidade de realização do curso de Mestrado.

Ao professor Wilkley Bezerra Correa pelo apoio e orientação no decorrer do mestrado.

Ao meu amigo Iury Amorim por todo apoio, ao longo do mestrado e da vida.

“Não te mandei eu? Esforça-te, e tem bom ânimo;
não te atemorizes, nem te espantes;
porque o Senhor teu Deus está contigo, por
onde quer que andares.” (Josué 1:9)

RESUMO

O controle e modelagem de sistemas industriais envolvem a implementação de um sistema supervisorio para processos de eventos discretos, bem como o controle entre os agentes na planta para executar processos específicos. Nesse cenário, os componentes industriais e os controladores locais não operam isoladamente no processo de manufatura, tornando essencial a otimização das referências e a sincronia por meio do controle cooperativo. Essa abordagem busca alcançar uma curva otimizada de referência para os controladores locais, simultaneamente evitando colisões entre os agentes. Na modelagem do sistema industrial, a necessidade de uma representação intuitiva e detalhada é atendida por meio das Redes de Petri Colorida. As Redes de Petri Colorida apresentam não apenas uma representação gráfica e hierárquica orientada a eventos, mas também proporcionam uma linguagem que facilita a comunicação com diferentes tipos de mensagens ao longo da rede. O objetivo deste trabalho é desenvolver uma abordagem integrada de modelagem e simulação, utilizando Redes de Petri Colorida e estratégias de Controle Cooperativo, para uma planta industrial. A modelagem dos eventos ao longo do sistema é realizada por meio da Rede de Petri, utilizando transições e lugares. O controle é implementado através do algoritmo de consenso de multiagentes, garantindo a sincronização dos agentes da planta para diferentes referências e etapas de operação na Rede de Petri. A técnica de controle e modelagem é apresentada por meio 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 são alteradas ao longo de eventos modelados pela Rede de Petri. Verificou-se que a 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.

ABSTRACT

The control and modeling of industrial systems involve the implementation of a supervisory system for discrete event processes, as well as control among the agents in the plant to execute specific processes. In this scenario, industrial components and local controllers do not operate independently in the manufacturing process, making it essential to optimize references and synchronize through cooperative control. This approach aims to achieve an optimized reference curve for local controllers while simultaneously avoiding collisions between agents. In the modeling of the industrial system, the need for an intuitive and detailed representation is addressed through Colored Petri Nets. Colored Petri Nets not only provide a graphical and hierarchical event-oriented representation but also offer a language that facilitates communication with different types of messages throughout the network. The objective of this work is to develop an integrated approach to modeling and simulation, using Colored Petri Nets and Cooperative Control strategies, for an industrial plant. The modeling of events throughout the system is carried out through the Petri Net, using transitions and places. Control is implemented through the multi-agent consensus algorithm, ensuring the synchronization of plant agents for different references and operation stages in the Petri Net. The control and modeling technique is presented through application in a scenario of synchronization and formation of automata along a defined trajectory, where the order and organization of automata are altered over events modeled by the Petri Net. It was found that the combined use of Petri Net modeling with consensus control performs well and abstracts the modeling of a system involving many agents with different events. Additionally, the use of consensus control efficiently aids in synchronizing various industrial agents.

Keywords: Consensus Control; Petri Nets; Multi-agent Control; Industrial Plants; Hybrid Systems.

LISTA DE FIGURAS

Figura 1 – Exemplo de nós e arestas em um grafo.	21
Figura 2 – Exemplo de um grafo.	23
Figura 3 – Árvore de escopo de um sistema a eventos discretos.	27
Figura 4 – Exemplo paralelismo Síncrono.	30
Figura 5 – Exemplo paralelismo Assíncrono.	31
Figura 6 – Exemplo de Caminhos alternativos e Repetição.	32
Figura 7 – Exemplo de Rede de Petri.	34
Figura 8 – Planta com pista, vagões e pontos de mudança de via	36
Figura 9 – Modelo dos Componentes Hierárquicos da RPC	38
Figura 10 – Rede de Nível Hierárquico superior	39
Figura 11 – Máscara dos Componentes da pista na RPC	40
Figura 12 – Máscara dos Componentes da pista com o controle das chaves na RPC . . .	41
Figura 13 – Máscara dos Componentes de reestabelecimento na RPC	42
Figura 14 – Máscara dos Componentes da pista com o controle das chaves na RPC . . .	43
Figura 15 – Máscara dos Componentes de controle das chaves e comunicadores na RPC	44
Figura 16 – Rede referente ao controle hierárquico X1 na RPC	45
Figura 17 – Rede referente ao controle hierárquico X2 na RPC	46
Figura 18 – Rede referente ao controle hierárquico X3 na RPC	47
Figura 19 – Rede referente ao controle hierárquico X4 na RPC	47
Figura 20 – Máscara de componentes relacionados a ordem na RPC	48
Figura 21 – Rede referente ao componente de Ordem na RPC	49
Figura 22 – Máscara referente a geração de comando do componente de Ordem na RPC	50
Figura 23 – Máscara referente ao restabelecimento do componente de Ordem na RPC . .	50
Figura 24 – Máscara referente ao restabelecimento do componente de Ordem na RPC . .	51
Figura 25 – Modelagem do grafo entre agentes	52
Figura 26 – Simulação Caso de Ultrapassagem	53
Figura 27 – Análise velocidade Caso de Ultrapassagem	53
Figura 28 – Análise Diminuição do peso da comunicação com o líder no Caso de Ultra- passagem	54
Figura 29 – Análise pesos muito diferentes na comunicação com o líder Caso de Ultra- passagem	55

Figura 30 – Simulação Caso de não ultrapassagem	55
Figura 31 – Simulação Caso de não ultrapassagem, diminuição peso comunicação com o líder	56
Figura 32 – Simulação Caso de não ultrapassagem, grande diferença peso na comunicação com o líder	57

LISTA DE QUADROS

Quadro 1 – Conjunto de Cores na RPC	37
Quadro 2 – Conjunto de Variáveis na RPC	38

LISTA DE ABREVIATURAS E SIGLAS

RP Rede de Petri

RPC Rede de Petri Colorida

LISTA DE SÍMBOLOS

G	Grafo
V	Conjunto de vértices no grafo
E	Conjunto de arcos no grafo
a_{ij}	Peso do arco
A	Matriz de Adjacência (Controle Cooperativo)
D	Matriz Diagonal
L	Matriz Laplaciana
d	Diâmetro do Grafo
c	Valor de consenso
\otimes	Produto de Kronecker
P	Conjunto finito de lugares
T	Conjunto finito de transições
A	Conjunto finito de arcos (Rede de Petri)
w	Função de ponderação dos arcos

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	16
1.2	Motivação	17
1.3	Estado da arte	18
1.4	Objetivos	19
<i>1.4.1</i>	<i>Objetivos gerais</i>	<i>19</i>
<i>1.4.2</i>	<i>Objetivos específicos</i>	<i>19</i>
1.5	Produção Científica	19
1.6	Organização do trabalho	20
2	SISTEMA MULTIAGENTES	21
2.1	Teoria dos Grafos	21
2.2	Teoria algébrica dos grafos e consenso do controle cooperativo	21
<i>2.2.1</i>	<i>Representação matricial dos grafos</i>	<i>22</i>
<i>2.2.2</i>	<i>Matriz de Grafo Laplaciana</i>	<i>22</i>
2.3	Consenso com Integrador único	24
<i>2.3.1</i>	<i>Protocolo de controle distribuído para o consenso</i>	<i>24</i>
<i>2.3.2</i>	<i>Consenso com líder</i>	<i>25</i>
<i>2.3.3</i>	<i>Consenso para nós com estados como vetores</i>	<i>25</i>
3	REDES DE PETRI	26
3.1	Modelagem de Sistemas	26
<i>3.1.1</i>	<i>Modelagem de sistemas a eventos discretos</i>	<i>27</i>
3.2	Representação em máquina de estados	28
<i>3.2.1</i>	<i>Modelagem de processos sequenciais</i>	<i>28</i>
<i>3.2.1.1</i>	<i>Representação com refinamentos sucessivos</i>	<i>28</i>
<i>3.2.1.2</i>	<i>Explosão combinatória</i>	<i>29</i>
<i>3.2.1.3</i>	<i>Não-independência de submáquinas e bloqueio</i>	<i>29</i>
3.3	Modelagem utilizando Rede de Petri	29
<i>3.3.1</i>	<i>Evolução síncrona e assíncrona</i>	<i>30</i>
<i>3.3.2</i>	<i>Caminhos alternativos e Repetição</i>	<i>31</i>
3.4	Redes de Petri	31

3.4.1	<i>Linguagem Formal</i>	32
3.4.2	<i>Redes de Petri Colorida</i>	33
4	SIMULAÇÃO	35
4.1	Planta Industrial	35
4.2	Modelagem em Redes de Petri Colorida	37
4.2.1	<i>Modelagem da Pista</i>	39
4.2.2	<i>Modelagem Controladores</i>	43
4.2.3	<i>Modelagem Ordenação dos vagões</i>	48
4.3	Controle Cooperativo aplicado a Multiagentes	51
4.3.1	<i>Casos de Ultrapassagem</i>	52
4.3.2	<i>Casos de Não Ultrapassagem</i>	54
5	CONCLUSÃO	58

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 coordenar uma gama de processos, os sistemas industriais inteligentes integrados possuem a necessidade de adaptar-se a novas variações e configurações, abrindo espaço para máquinas e sistemas com programação mais robusta e reconfigurável.

Visando abordar a problemática do aprimoramento da coordenação entre os diversos componentes e controladores em sistemas industriais, a eficiência dos processos e a otimização de recursos representam alguns dos principais desafios enfrentados pela indústria, sendo alvo de estratégias contínuas de melhoria.

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. (??)

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). (??)

No contexto do controle é aplicada a estratégia de controle cooperativo em sistemas multi-agentes para a coordenação de vários agentes. Essa abordagem busca alcançar uma curva otimizada de mudança de referência para os controladores locais, enquanto simultaneamente evita colisões entre os agentes. Adicionalmente, a estratégia promove robustez e adaptabilidade diante de possíveis travamentos, desgaste e mudanças de comportamento entre os componentes da planta.

Nesse sentido, o objetivo desta pesquisa é analisar e desenvolver uma abordagem integrada de modelagem e simulação, utilizando Redes de Petri Colorida e estratégias de Controle Cooperativo, aplicadas a uma planta industrial. De forma mais específica, buscou-se: realizar um estudo sobre o estado atual do controle cooperativo em sistemas multiagentes, com ênfase na modelagem por redes de Petri; propor um modelo detalhado de uma planta industrial utilizando redes de Petri colorida; aplicar estratégias de controle cooperativo para promover consenso

entre os diferentes componentes na planta industrial; e integrar o Controle Cooperativo com a simulação de um modelo de planta industrial por meio de Redes de Petri Colorida, buscando uma análise mais abrangente e eficaz das interações e comportamentos do sistema.

A metodologia empregada envolve a integração da modelagem dos eventos ao longo do sistema, realizada por meio da rede de Petri, que utiliza transições e lugares, com o controle implementado através do algoritmo de consenso de multiagentes. 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.

1.1 Justificativa

Na perspectiva da Engenharia de Controle e Automação, a integração do ciclo de modelagem, controle e simulação dos sistemas é essencial para orientar o planejamento e a programação dos controladores industriais, bem como para monitoramento no sistema supervisão. Através da modelagem e controle é possível obter impactos significativos na eficiência operacional do sistema, prevenção de falhas e bloqueios, otimização do uso de recursos, eficiência energética e sincronização eficaz entre os diversos componentes da planta.

De maneira geral, em sistemas de automação, o processo de manufatura não se realiza apenas por meio de componentes isolados, mas sim por meio de um conjunto em sincronia, executando etapas específicas de um processo. Assim, como controle de um sistema industrial não se limita apenas a controladores locais em cada componente de sensores e atuadores, mas também a otimização do acionamento e referência desses controladores locais visando a cooperação entre diversos agentes de uma planta para realizar uma sequência de etapas e processamento de peças e produtos.

Assim como o controle supervisão, híbrido, entre outros, apresentados por (??) o controle cooperativo é aplicado a um controle hierarquicamente de mais alto nível em relação aos controladores locais, emprega um algoritmo de consenso entre os diversos agentes envolvidos. Essa estratégia de controle alcança a curva otimizada de mudança de referência para os controladores locais, ao mesmo tempo em que evita colisões entre os agentes e promove mais robustez e adaptabilidade a travamentos, desgaste e mudanças de comportamento entre os componentes da planta.

A escolha da abordagem com Redes de Petri Colorida para a modelagem do sistema

industrial é respaldada pela capacidade única dessa metodologia em representar visualmente as interações complexas entre os diferentes componentes de um sistema. (??) atribui que as Redes de Petri oferecem não apenas uma representação orientada a eventos, mas também uma linguagem que possibilita a modelagem de diferentes tipos de mensagens ao longo da rede. Além disso, permitem a adoção de modelos hierárquicos, como os exemplos top-down e bottom-up, proporcionando uma análise, compreensão e nível de detalhamento mais preciso e abrangente. Essas características contribuem significativamente para uma representação visual completa e informativa do sistema industrial em questão.

1.2 Motivação

O Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará tem concentrado seus estudos na área de controle e automação industrial, explorando diversas abordagens de controladores e modelagens para esse propósito.

A proposta de elaborar um modelo detalhado de uma planta industrial utilizando Redes de Petri Colorida, além de aplicar estratégias de controle cooperativo para alcançar consenso entre os diferentes componentes e integrar o Controle Cooperativo com a simulação do modelo, representa uma abordagem inovadora e vantajosa para a automação industrial.

O controle cooperativo, ao promover a colaboração entre os diversos elementos da planta, busca otimizar a eficiência operacional, melhorar a resposta a eventos imprevistos e aumentar a flexibilidade do sistema. A busca por consenso entre os componentes é crucial para evitar conflitos e assegurar um funcionamento harmonioso.

A escolha das Redes de Petri Colorida como base para o modelo oferece uma representação visual clara e precisa da dinâmica do sistema, permitindo uma análise detalhada dos processos e comportamentos da planta. Essa abordagem facilita a identificação de pontos críticos, a avaliação de cenários complexos e a implementação efetiva de estratégias de controle.

Portanto, a combinação do Controle Cooperativo com Redes de Petri Colorida oferece uma solução robusta e eficiente para enfrentar os desafios da automação industrial, proporcionando um ambiente mais adaptável, resiliente e capaz de lidar com as demandas dinâmicas e imprevisíveis encontradas em ambientes industriais complexos.

1.3 Estado da arte

O artigo de (??) apresenta uma metodologia para criar uma Rede de Petri Hierárquica modelando as atividades de um sistema robótico multiagente. A metodologia segue a abordagem de separação de módulos para o design do software de controle de robô, resultando em cinco camadas que representam: o sistema composto por agentes, subsistemas de agentes, comportamentos dos subsistemas, padrão de comportamento e, finalmente, comunicação entre subsistemas e cálculo de funções de transição.

O desenvolvimento de um sistema robótico multiagente SMART, em que o conceito de agentes é aplicado tanto para as entidades de hardware quanto a de software. Os agentes de hardware cooperam fortemente com os agentes de software que são classificados como processamento de imagem, comunicação, gerenciamento de tarefas, tomadas de decisão, planejamento trajetória. Para a modelagem, controle e avaliação das tarefas cooperativas entre os agentes é utilizadas um tipo de Rede de Petri chamado Work-Flow Petri Nets. O principal aspecto de um sistema SMART é o problema de coordenação, que garante que a decisão tomada por cada agente coopere com o grupo inteiro, de modo que em um sistema multiagente o ponto crucial é a arquitetura de comunicação entre os agentes. (??)

No contexto da hierarquização e modularização na modelagem de redes de Petri, a análise sobre vivacidade e modularização é abordada por (??), que propõe um conjunto de regras de transformação de uma classe fundamental de redes multiagentes em redes de Petri modulares e comprova a isomorfia dos espaços de estados das redes multiagentes e das redes de Petri modulares geradas.

A aplicação de controle cooperativo para estratégia de controle de sistemas multiagentes através da abordagem de transmissão de token para coordenar a distribuição de tarefas para um grupo de robôs. Foi realizado por (??).

Na rede de protocolo de contrato em sistemas multiagente, (??) trabalha sobre a problemática da alocação excessiva de recursos para a formação de uma rede colaborativa, tal abordagem é tratada via Rede de Petri Colorida para coordenar o uso de recursos entre os diversos agentes do sistema.

1.4 Objetivos

1.4.1 *Objetivos gerais*

O presente trabalho dedica-se a desenvolver uma abordagem integrada de modelagem e simulação utilizando Redes de Petri Colorida e estratégias de Controle Cooperativo de uma planta industrial.

1.4.2 *Objetivos específicos*

- Realizar um estudo acerca do estado da arte no campo do controle cooperativo em sistemas multiagentes com modelagem em redes de Petri.
- Propor um modelo detalhado de uma planta industrial utilizando rede de Petri colorida.
- Aplicar estratégias de controle cooperativo para um consenso entre os diferentes componentes na planta industrial.
- Integrar o Controle Cooperativo com a simulação de um modelo de uma planta industrial utilizando Redes de Petri Colorida.

1.5 Produção Científica

Durante o desenvolvimento desta dissertação, foram publicados ou submetidos a congressos ou periódicos os seguintes artigos:

- **PAIVA, Davi Alexandre; VASCONCELOS, Felipe José de Sousa; FILGUEIRAS, Iury de Amorim Gaspar; CORREIA, Wilkley Bezerra.** A simple procedure for modeling and identification of a test bench 4-dof manipulator. In: SOCIEDADE BRASILEIRA DE AUTOMÁTICA (SBA) AND GALOÁ SCIENCE. Congresso Brasileiro de Automática. Online: Galoá, 2020. (CBA2020, 1), p. 1050. ISSN 2525-8311. Aparece nas coleções: DEEL - Trabalhos apresentados em eventos. Disponível em: <http://www.repositorio.ufc.br/handle/riufc/65012>. (??)
- **VASCONCELOS, F. J. S.; LEITE, G. C.; NETO, G. B. F.; CORREIA, W. B.; AGUIAR, V. P. B.; PAIVA, D. A.** ANFIS Identification Applied to a Reservoir Level Liquid System. 2021 9th International Conference on Control, Mechatronics and Automation (ICCMA), Belval, Luxembourg, 2021, pp. 135-140, doi: 10.1109/ICCMA54375.2021.9646226. (??)

1.6 Organização do trabalho

O texto está organizado em capítulos, permitindo uma abordagem sequencial lógica dos temas, conforme a seguinte estrutura:

- **Capítulo 1:** 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.
- **Capítulo 2:** 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. São abordadas as principais estratégias de modelagem, representação em grafos e controle de sistemas multiagentes, com um enfoque especial na interação entre agentes.
- **Capítulo 3:** Este segmento explora o uso de Redes de Petri como uma ferramenta fundamental na modelagem de sistemas a eventos discretos e, especificamente, em contextos de controle cooperativo em sistemas multiagentes. Serão apresentados conceitos essenciais, representações gráficas e aplicações práticas das Redes de Petri Colorida.
- **Capítulo 4:** Neste tópico, abordaremos a importância da simulação na validação e análise de sistemas multiagentes modelados por Redes de Petri.
- **Capítulo 5:** Por fim, este último capítulo trata das considerações gerais sobre os conceitos apresentados e uma discussão crítica acerca dos resultados de simulação.

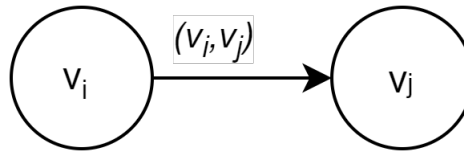
2 SISTEMA MULTIAGENTES

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.



Fonte: elaborada pelo autor.

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

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

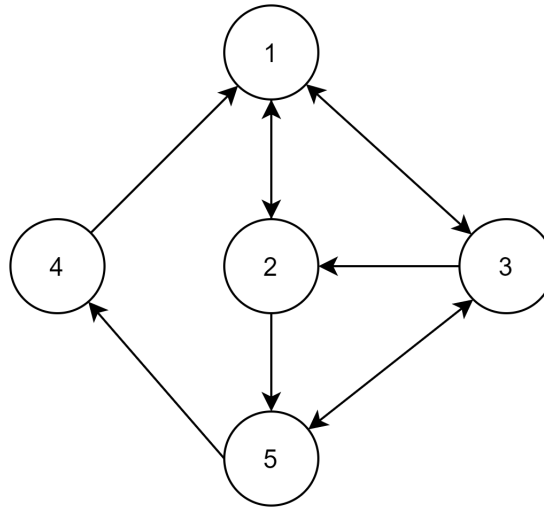
2.2.2 Matriz de Grafo Laplaciana

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.

Figura 2 – Exemplo de um grafo.



Fonte: elaborada pelo autor.

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.

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

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 mesmos 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 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.

Para o protocolo de primeira ordem local dado por

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 (??) 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.

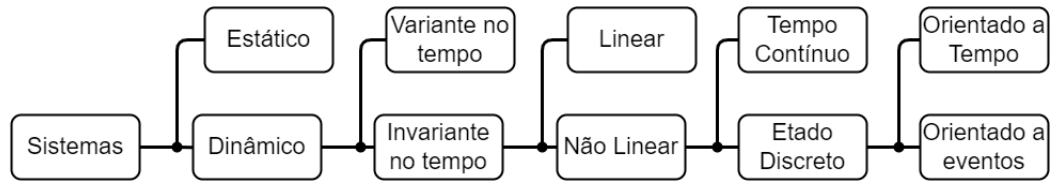
Sistema Orientado a Eventos e Tempo: Em um sistema orientado ao tempo o estado do sistema muda com a mudança de tempo, já a eventos a mudança de estado ocorre de forma instantânea a partir da ocorrência de um evento.

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.



Fonte: adaptado de (??).

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 (??), 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 (??) 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 compartilhada entre as máquina ou troca de sinais entre as máquinas é necessário analisar o comportamento 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

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). De acordo com (GIUA; SILVA, 2017), as redes de Petri têm sido consideradas com um modelo adequado para um controle supervisorio 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;

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.

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.

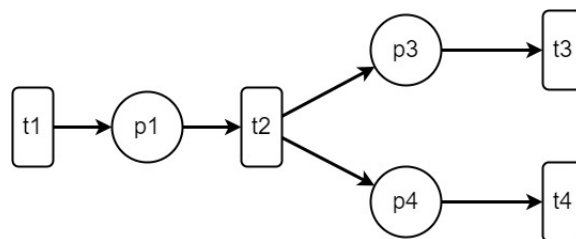
O comportamento dinâmico do sistema se dá através do disparo das transições, evento que faz com que o sistema passe do estado atual para o próximo estado. Tal disparo consiste em duas etapas, a primeira de retirar as fichas dos lugares de entrada e por fim depositar as fichas em cada lugar de saída.

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.

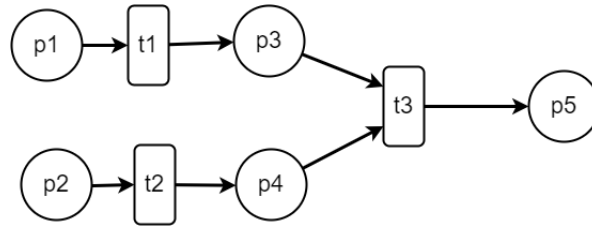
Figura 4 – Exemplo paralelismo Síncrono.



Fonte: Elaborado pelo autor.

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.



Fonte: Elaborado pelo autor.

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.

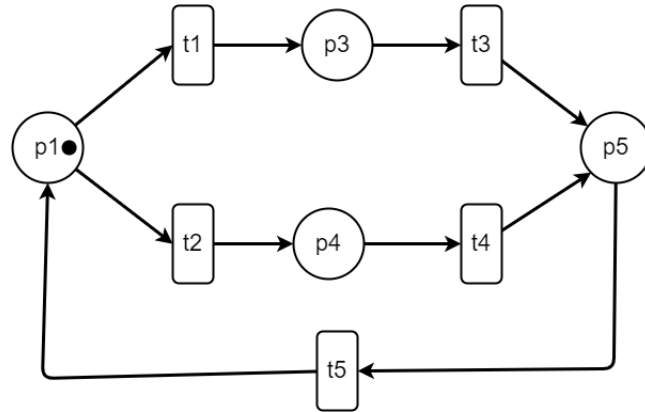
Observa-se esse fenômeno na modelagem descrita pela figura 6, em que no momento em que p_1 recebe uma ficha é sensibilizada as transições t_1 e t_2 , de modo que a rede de petri não restringe a escolha de um ou outra transição, porém caso uma transição seja disparada a outra não poderá ser, ocorrendo 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.

3.4 Redes 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. Para a modelagem proposta utilizou-se a ferramenta do CPN Tools, assim como as funcionalidades envolvendo hierarquia para maior legibilidade da rede.

Figura 6 – Exemplo de Caminhos alternativos e Repetição.



Fonte: Elaborado pelo autor.

De acordo com (??), os eventos na Rede de Petri Rede de Petri (RP) são relacionados as transições, de modo que para que aconteça o disparo de uma transição é necessário satisfazer determinadas condições inscritas na rede. As informações relacionadas a tais condições estão contidas em lugares, que são vistos como "entradas" das transições. Lugares, transições e as relações entre eles são os componentes básicos de uma RP. A RP possui dois tipos de nós, lugares e transições, e os arcos que conectam eles. Na definição de grafos a RP é classificada com um grafo bipartido de modo que os arcos não podem conectar diretamente dois nós do mesmo tipo, exemplo lugar a lugar ou transição a transição. A definição de grafo é dada a seguir.

3.4.1 Linguagem Formal

De acordo com (??), a Rede de Petri é definida como um grafo bipartido ponderado, dado pelo conjunto (P, T, A, w, x) , tal que:

- P é o conjunto finito de lugares (um tipo de nó na grafo)
- T é o conjunto finito de transições (o outro tipo de nó no grafo)
- $A \subseteq (P \times T) \cup (T \times P)$ é o conjunto de arcos dos lugares para as transições e das transições para os lugares no grafo.
- $w : A \rightarrow \{1, 2, 3, \dots\}$ são as funções de ponderação dos arcos.
- x : é a marcação da rede de Petri dada pela função $x : P \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$, tal marcação é definida como um vetor linha $x = x(p_1), x(p_2), \dots, x(p_n)$, tal que n , é o número de lugares na rede.

É considerado que em (P, T, A, w) , não tem lugares em transições isolados. Na representação em grafos os lugares $P = \{p_1, p_2, p_3, \dots\}$ são representados com um círculo, as transições

$T = \{t_1, t_2, t_3, \dots\}$ como retângulos. Os conjunto de arcos $A = \{w_1, w_2, w_3, \dots\}$, cada arco w é representado em função de um lugar e uma transição e o peso associado como $w_k = (p_i, t_j) = W$, tal que o arco liga o lugar p_i a transição t_j e possui o peso igual a W . A marcação no grafo é dada como fichas (ponto preto) no lugar marcado p_i tal que $x(p_i) \in \mathbb{N}$.

Um exemplo de rede com os elementos citados é representado pelo grafo na figura 7. Os lugares da rede de petri são representados pelos círculos $P = \{p_1, p_2, p_3, \dots, p_7\}$, as transições $T = \{t_1, t_2, \dots, t_5\}$, já no caso das transições o conjunto W é definido da seguinte forma: $W = \{w_1, w_2, \dots, w_8\}$ onde:

- $w_1 = (p_1, t_1)$ representa um arco que sai de p_1 e entra em t_1 .
- $w_2 = (t_1, p_3)$ representa um arco que sai de t_1 e entra em p_3 .
- $w_3 = (t_1, p_2)$ representa um arco que sai de t_1 e entra em p_2 .
- $w_4 = (p_2, t_3) = 2$ representa um arco que sai de p_2 e chega em t_3 com peso igual a 2.
- $w_5 = (p_3, t_4)$ representa um arco que sai de p_3 e entra em t_4 .
- $w_6 = (t_4, p_5)$ representa um arco que sai de t_4 e entra em p_5 .
- $w_7 = (p_5, t_3)$ representa um arco que sai de p_5 e entra em t_3 .
- $w_8 = (t_3, p_4)$ representa um arco que sai de t_3 e entra em p_4 . Por fim, a representação da marcação tem-se que no lugar p_1 a quantidade de 1 ficha e nos outros nenhum ficha, formando o vetor linha representado na equação 3.2:

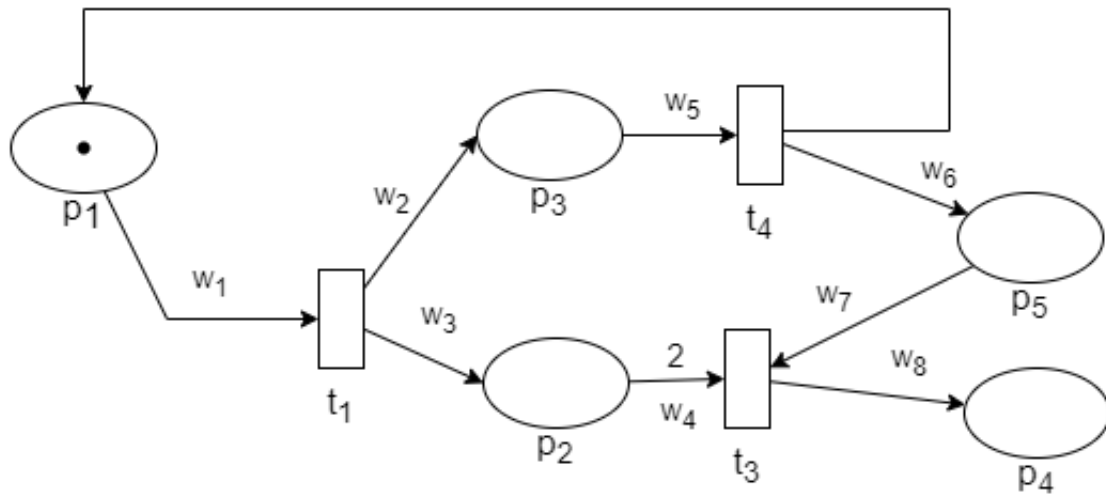
$$\begin{aligned} X &= [x(p_1), x(p_2), \dots, x(p_5)] \\ &= [1, 0, 0, 0, 0] \end{aligned} \tag{3.2}$$

3.4.2 Redes de Petri Colorida

A rede apresentada anteriormente com elementos simples de arcos, marcações, transições, de acordo com (??) são denominadas de redes de Petri de baixo nível, geralmente utilizadas para modelagem de hardware, já para a modelagem de sistemas com concorrência, comunicação, hierarquia. Exemplo sistemas de manufatura, sistemas baseados em agentes, pode-se utilizar as redes de Petri de alto nível, também denominadas como redes de Petri colorida.

As redes de Petri coloridas (Rede de Petri Colorida (RPC)) 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. É uma combinação das redes de Petri com linguagem de computação. Os elementos da RPC são baseados em hierarquias, que

Figura 7 – Exemplo de Rede de Petri.



Fonte: Elaborado pelo autor.

permite que módulos tenham submódulos, permitindo a reusabilidade dos módulos. Isso permite trabalhar tanto com a abordagem *top-down*, quanto *bottom-up*, na construção dos modelos em RPC, capturando assim diferentes camadas de abstração na modelagem do sistema.

4 SIMULAÇÃO

Nessa seção, busca-se validar a abordagem integrada de Modelagem por meio de Redes de Petri Coloridas (RPC) e Controle Cooperativo. Para realizar essa validação, optou-se por empregar a estratégia de simulação, utilizando um sistema multiagente composto por diversos componentes, os quais foram concebidos para representar um ambiente similar a um sistema industrial.

Neste processo de simulação, são utilizadas ferramentas de software dedicadas à modelagem em RPC, com destaque para o CPN-Tools.¹ Essa ferramenta oferece funcionalidades específicas para a edição, simulação e análise de Redes de Petri Coloridas. A escolha do CPN-Tools visa proporcionar uma representação fiel e detalhada do sistema em estudo.

Para a implementação do algoritmo matemático de consenso, optou-se por utilizar a linguagem de programação Python. Essa escolha deve-se à reputação da linguagem como sendo de alto nível, funcional e orientada a objetos, oferecendo assim uma abordagem versátil e eficiente para a execução do algoritmo em questão.

Ao unir a simulação do sistema multiagente com as capacidades do CPN-Tools e a flexibilidade do Python, almeja-se obter resultados robustos que validem a eficácia da abordagem proposta. Esta seção detalhará o processo de simulação, desde a definição dos parâmetros até a análise crítica dos resultados obtidos, proporcionando uma compreensão clara e abrangente da validade e desempenho da abordagem integrada.

4.1 Planta Industrial

A gestão de vagões em uma planta industrial, que permite a ultrapassagem entre eles por meio de mudanças nas pistas, representa uma abordagem inovadora no contexto ferroviário industrial. Esse sistema dinâmico busca otimizar o movimento e a alocação de vagões, proporcionando flexibilidade e eficiência operacional.

Ao incorporar a capacidade de ultrapassagem, a planta industrial adquire versatilidade, possibilitando a reorganização estratégica dos vagões para atender a demandas específicas. Essa funcionalidade torna-se particularmente valiosa em situações onde é necessário priorizar determinados vagões, reduzir tempos de espera ou melhorar o desempenho global do sistema ferroviário dentro do contexto da planta industrial.

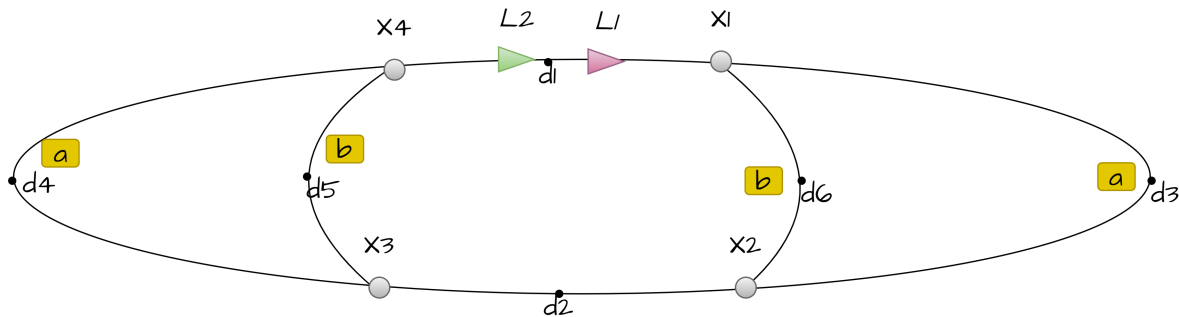
¹ pode ser acessada em <https://cpntools.org/>

A implementação de mudanças nas pistas como um meio de permitir a ultrapassagem requer uma coordenação precisa e um controle eficaz do sistema. Técnicas como Redes de Petri Coloridas (RPC) ou algoritmos de controle cooperativo podem ser empregados para modelar e simular o comportamento dinâmico da planta, considerando as interações entre os vagões e as mudanças nas pistas.

Um exemplo de representação desse sistema escolhido é demonstrada na figura 8, tal que:

- Os componentes L_1 e L_2 ilustrados em formato triangular são os vagões que representam os dois autômatos, ou agentes do sistema.
- As curvas d_1, d_2, \dots, d_6 representam as pistas pelas quais os vagões se locomovem, de modo os dois vagões estão posicionados inicialmente na pista denominada de d_1 .
- os elementos x_1, x_2, x_3, x_4 representam as quatro chaves responsáveis por fazer a mudança de via, elas podem comutar em duas posições diferentes a posição b que representa o caminho de menor comprimento ou a posição a que representa o caminho de maior comprimento.

Figura 8 – Planta com pista, vagões e pontos de mudança de via



Fonte: Elaborado pelo autor.

Para a simulação específica, os vagões movem-se entre pistas, e para efetuar uma ultrapassagem, é necessário que o vagão a ser ultrapassado siga pelo caminho mais longo, enquanto o vagão ultrapassante escolhe o caminho mais curto.

A chave de ultrapassagem foi designada como x_1 . Quando o comando de ultrapassagem é acionado, inicialmente a chave se move para a posição a , assegurando o desvio para a pista mais longa. Em seguida, ela se move para a posição b , garantindo o desvio para a pista mais curta.

A chave x_2 desempenha o papel de primeiro desviar para a posição b , para receber o vagão da pista d_6 . Posteriormente, comuta para a posição a , para receber o vagão da pista d_3 .

As chaves x_3 e x_4 permanecem fixas na posição b , assegurando o percurso mais curto. Esse arranjo permite uma ultrapassagem de vagões de forma segura e otimizada entre as pistas.

4.2 Modelagem em Redes de Petri Colorida

Para a modelagem do sistema proposto em Redes de Petri Colorida, foi definido o seguinte conjunto de cores para representar o tipo de token que vai ser alocado nos lugares da RPC de acordo com o seguinte quadro 1.

Quadro 1 – Conjunto de Cores na RPC

Nomenclatura	Tipo	Comando	Descrição
D	Inteiro	<code>colset D = int;</code>	Conjunto que indica a sequência dos vagões, onde o número 1 representa o vagão à frente, o número 2 representa o vagão atrás do 1, e assim por diante.
X	String	<code>colset X = string;</code>	Conjunto que representa a posição em que a chave se encontra, exemplo "a" para a posição A e "b" para a posição B.
L	String	<code>colset L = string;</code>	Conjunto de cores que representa os vagões ao longo da pista em que "L_1" é o vagão L1 e "L_2" o vagão L2.
Ord	Record	<code>colset Ord = record seq:D * train:L;</code>	Conjunto de cores que representa a associação do vagão L com a posição D, de modo que caso "L1" esteja na primeira posição, será representado por "seq:1,train:L1".

Fonte: Elaborado pelo autor.

Para cada conjunto de cores foi definido as variáveis correspondentes que serão alocados na inscrição dos arcos ao longo da RPC, de acordo com o quadro 2.

Para a abstração e melhor organização e legibilidade da representação na RPC, foram definidos dois níveis hierárquico o nível mais acima da Pista, e o nível mais abaixo o de controle e comando. Cada nível possui um conjunto de módulos que podem ser agrupados em 3 principais funcionalidades, Pista, Controle das Chaves "X" e controle de Ordenação dos vagões "L", como demonstrado na figura 9. Observa-se que para o controle das chaves X, foi definido 4 módulos de

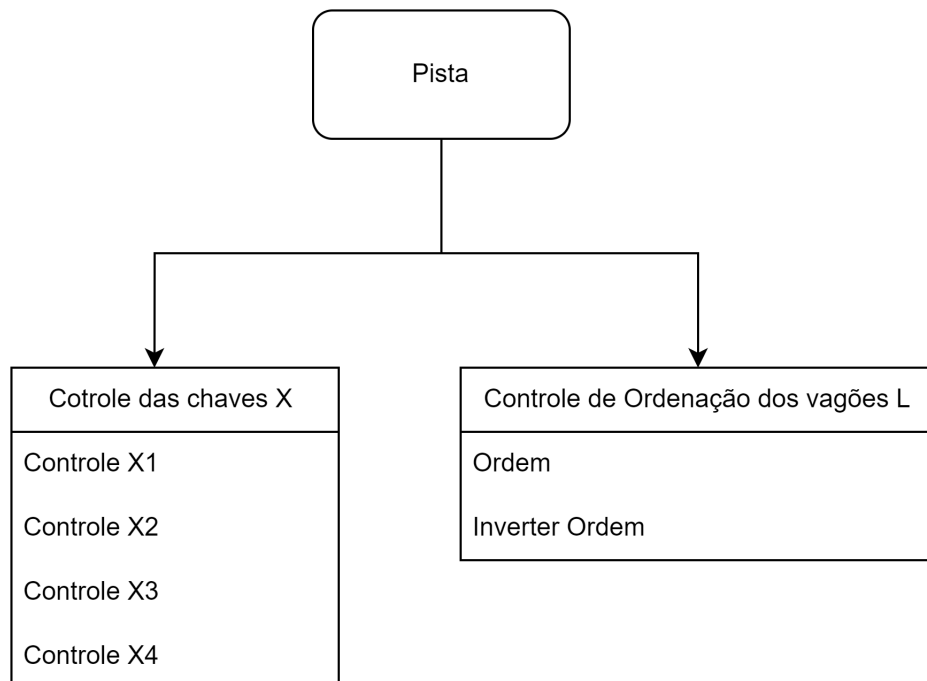
Quadro 2 – Conjunto de Variáveis na RPC

Variável	Tipo	Exemplo	Descrição
d	D	1'(1) ou 1'(2)	Variável de tipo Inteira, no exemplo tem-se 1 ficha com o valor 1, ou 2 fichas com valor 1, mais uma ficha com valor 2
x	X	1'("a") ou 1'("b")	Variável do Tipo <i>String</i> referente a posição de chave, pode ser do tipo "a" ou do tipo "b", como no exemplo.
l	L	1'("11") ou 1'("12")	Variável do Tipo <i>String</i> referente ao vagão/trem que nos casos podem ser do tipo "11" ou "12", como no exemplo para dois agentes
ord	Ord	1'{seq=2,train="11"}	Variável do Tipo <i>record seq:d * train:L</i> , no exemplo descrito indica que o trem "11" está na posição 2.

Fonte:Elaborado pelo Autor

controle 1 para cada chave, que são responsáveis por gerenciar em qual posição cada respectiva chave irá estar. No caso do componente de Controle de Ordenação foram definidos 2 módulos, o de "Ordem", responsável por gerar o comando de "Ultrapassagem" ou "Não Ultrapassagem", já o módulo de "Inverter Ordem" é responsável por atualizar as ordem dos vagões através da leitura da pista.

Figura 9 – Modelo dos Componentes Hierárquicos da RPC

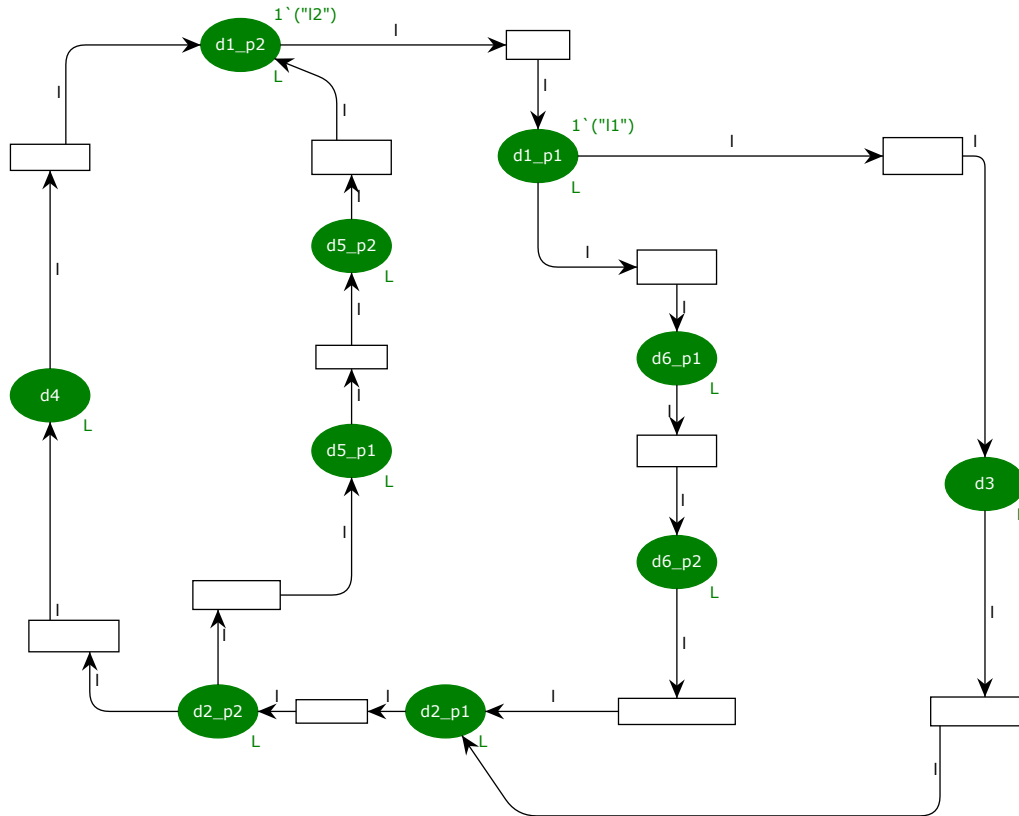


Fonte: Elaborado pelo autor.

A RPC de camada superior é dada pela figura 10, nela é possível verifica a existência

a área 1 da pista d_1 nela a pista sofre uma divisão, de modo que o vagão pode ir ou para a curva d_6 ou para a curva d_3 da pista. Os locais de junção ocorrem por exemplo na curva d_{2_p1} da pista, em que recebe tanto os vagões que veem da pista d_3 , quanto da pista d_6 .

Figura 11 – Máscara dos Componentes da pista na RPC



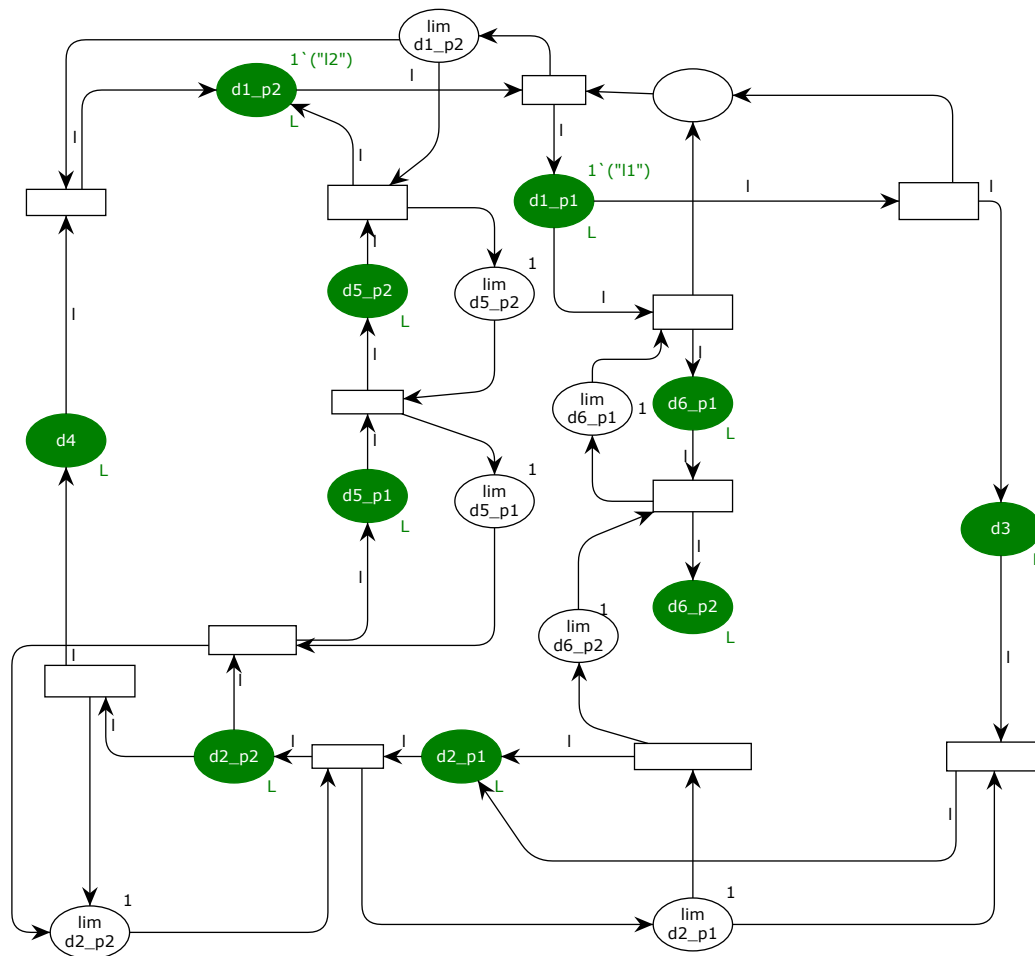
Fonte: Elaborado pelo autor.

Para a modelagem da pista também é importante a implementação do conceito limitadores de região, que são lugares que controlam a quantidade de vagões por região, que para níveis de modelagem na RPC são importantes para evitar que dois vagões estejam na mesma região ao mesmo tempo. Os limitadores de região são implementados em algumas regiões como demonstrado na figura 12, de modo que quando o vagão entra em determinada região é retirado uma ficha do lugar $lim\ d_{x_p_y}$, desabilitando a transição que permite a entrada de um vagão naquela região, tal ficha é devolvida quando o vagão que está sai para outra região.

Por exemplo, note que para que uma ficha entre no lugar d_{5_p2} é necessário que haja uma ficha no lugar $lim\ d_{5_p2}$, e que ao entrar nessa região uma ficha é consumida e ao sair a ficha é retornada ao limitador garantido que somente um vagão ocupara aquela região por vez.

Outro conceito fundamental na modelagem de uma RPC é o de reestabelecimento da rede, garantido que ao chegar no final do ciclo de uma volta a rede continue em funcionamento

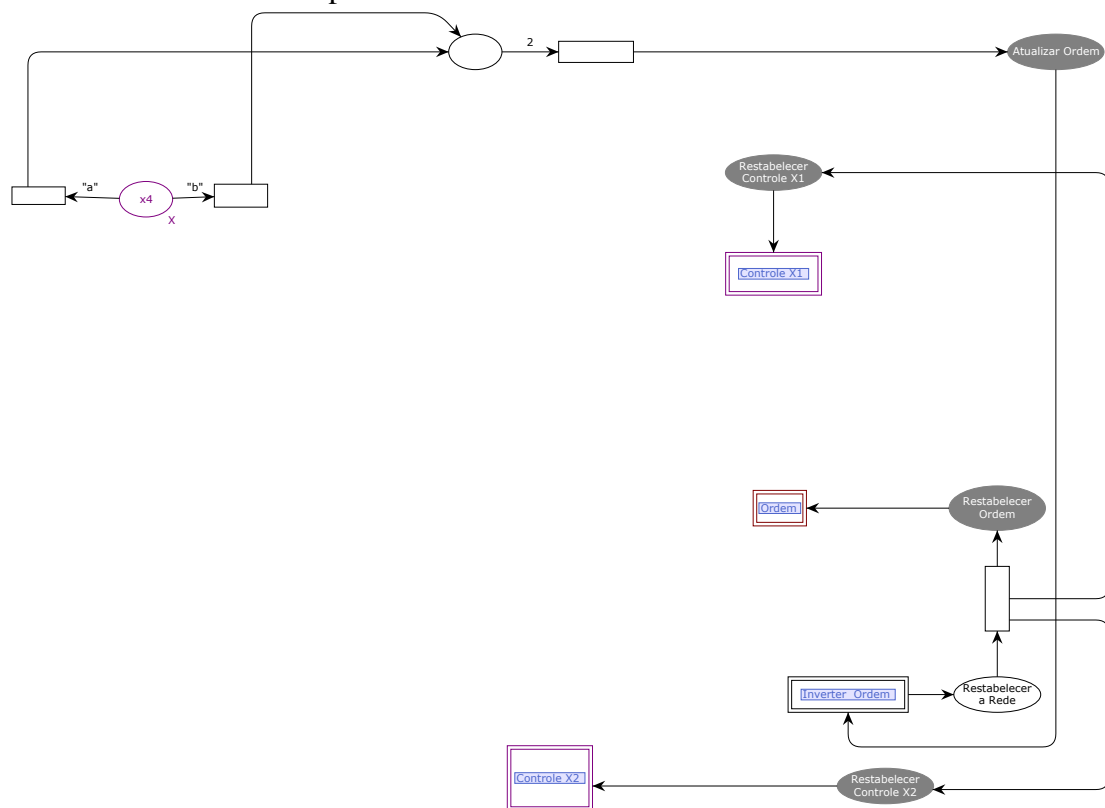
Figura 12 – Máscara dos Componentes da pista com o controle das chaves na RPC



Fonte: Elaborado pelo autor.

para um número ilimitado de iterações, tal reestabelecimento é demonstrado na figura 13. Note que após a segunda ativação da chave X_4 é dado o comando de atualizar a ordem vigente dos vagões e logo em seguida é dado o comando de *Reestabelecer a Rede*, enviando a ficha para reestabelecer os *Controles* X_1 e X_2 assim como o *Controle de Ordem*.

Figura 13 – Máscara dos Componentes de reestabelecimento na RPC



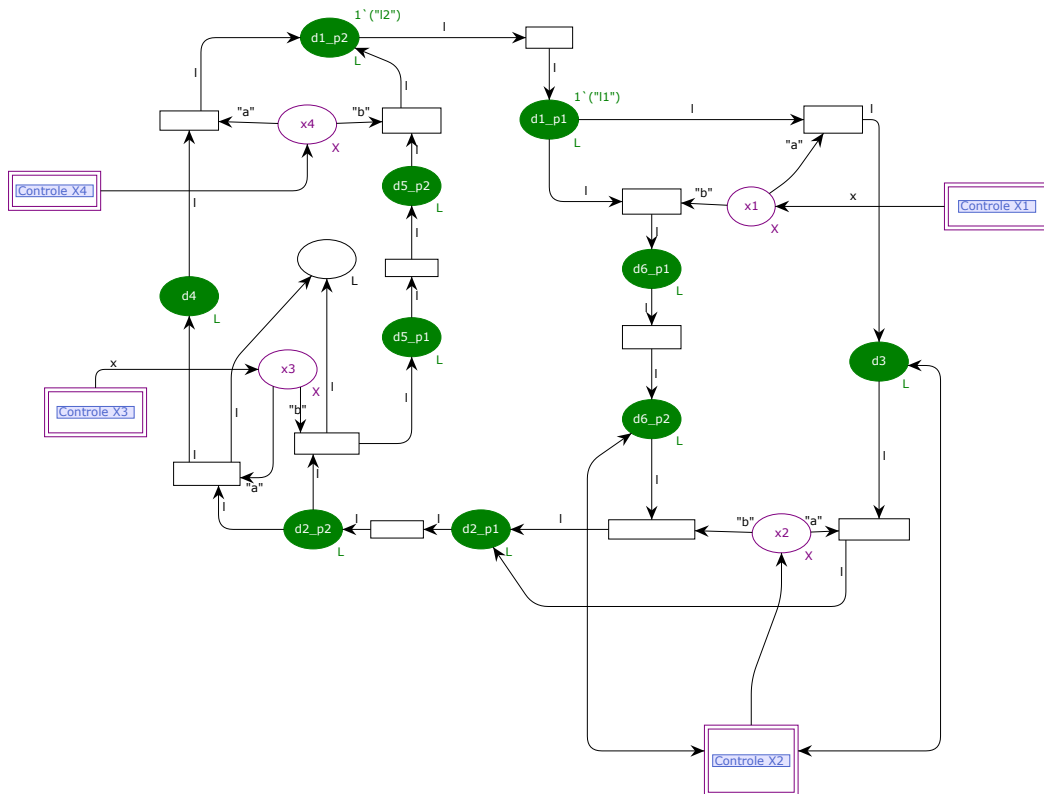
Fonte: Elaborado pelo autor.

4.2.2 Modelagem Controladores

Considerando o modelo anterior na subseção 4.2.1 de organização das pistas a figura 14 representa além dos componentes fundamentais das curvas das pistas d_1 a d_6 , os componentes de chaves X , com as transições hierárquicas de *Controle* X_1 a *Controle* X_4 . Observe que nos pontos de junção e divisão se encontram os locais X_1 a X_4 , que pertencem ao conjunto X que podem receber fichas do tipo "a" ou "b" como explicado no quadro 2, tais lugares controlam as condições de acionamento das transições ligadas os arcos de saída.

Por exemplo, dado um ficha "1'(l_1)" no lugar d_{1_p1} quem vai definir qual transição será habilita, se para ir para a pista d_3 ou para a pista d_6 , será a ficha no lugar x_1 , que é controlada pela transição hierárquica *Controle* X_1 . Tal que dependendo da ficha que o local X receber pelos respectivos Controladores será habilitada e desabilitada as transições que ligam as curvas ao longo da pista.

Figura 14 – Máscara dos Componentes da pista com o controle das chaves na RPC



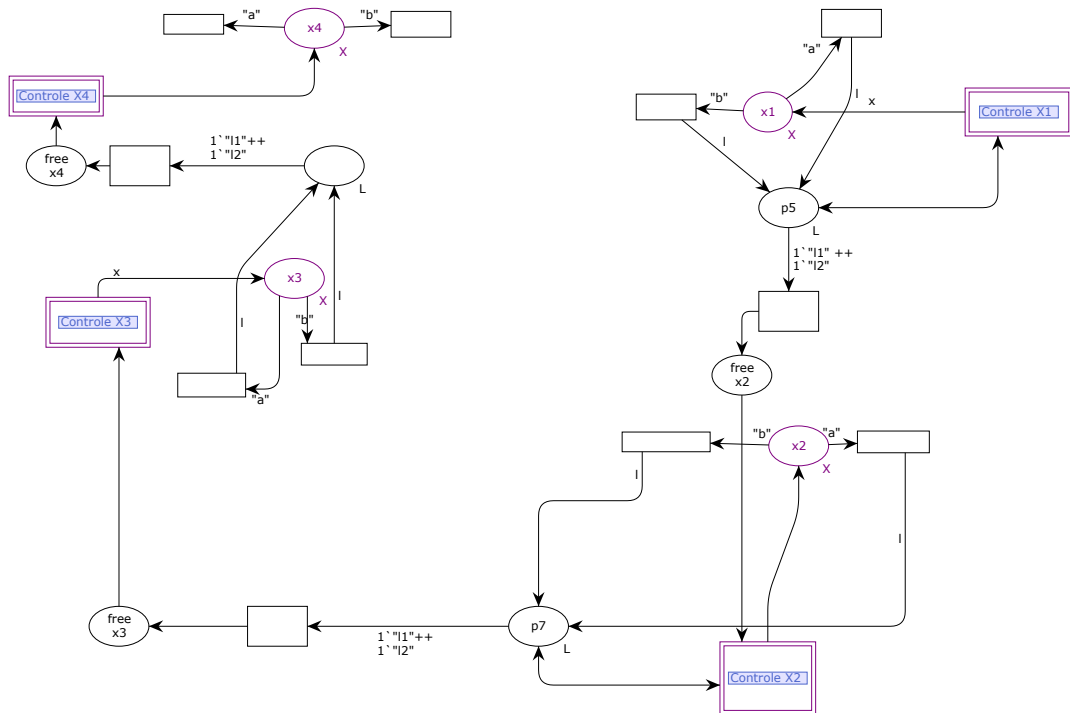
Fonte: Elaborado pelo autor.

Na modelagem dos controladores, é essencial assegurar a comunicação entre os elementos da pista em certos momentos, permitindo que os controladores tomem decisões com

base nas posições dos vagões. Um exemplo dessa comunicação é ilustrado na Figura 15. Em algumas transições conectadas às chaves, a informação sobre qual vagão passou pela região é transmitida à medida que as transições são acionadas. Posteriormente, essas informações são fornecidas, direta ou indiretamente, aos controladores.

Por exemplo, note que nas transições ligadas ao lugar X_2 possuem lugar de saída definida no conjunto L , de modo que ao passar algum vagão naquela região, a informação de qual vagão passou é processada pelo *Controle* X_2 . Posteriormente ao se passar os dois vagões diferentes a transição ligada ao lugar *free* x_3 é habilitada, transmitindo a informação de que já se passaram aqueles dois tipos de vagões naquela região, para a tomada de decisão do *Controle* X_3 .

Figura 15 – Máscara dos Componentes de controle das chaves e comunicadores na RPC



Fonte: Elaborado pelo autor.

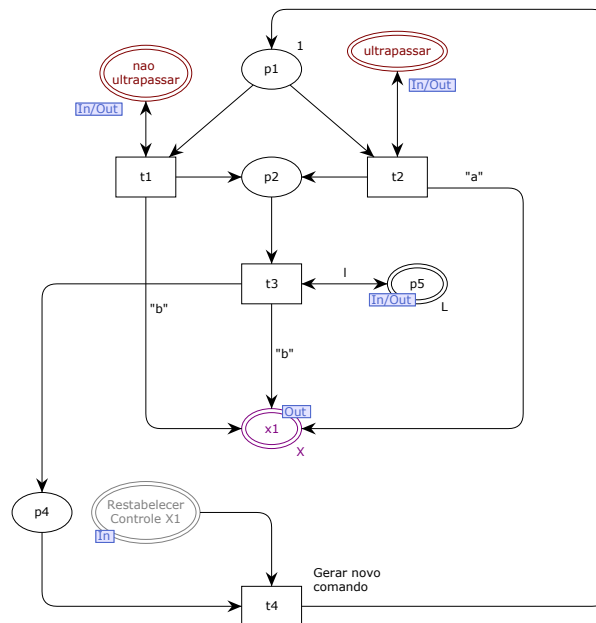
Em um nível hierárquico abaixo do nível da pista, como demonstrado na figura 9 estão os *Controladores* x_1 a x_4 , que serão detalhados a seguir.

O *Controlador* X_1 referente a transição *Controle* X_1 com borda vermelha na figura 15, é demonstrado na figura 16. Note que a rede de Controle de X_1 , possui alguns lugares com subscrição denominadas de *IN*, *OUT* ou *IN/OUT*, que são diretamente relacionadas aos lugares externos a rede, estabelecendo uma comunicação com lugares pertencentes ao nível hierárquico

superior. Alguns lugares como x_1 e p_5 já foram demonstrados na máscara de componentes na figura 15, outros serão demonstrados na subseção 4.2.3 referente a modelagem de ordenação dos vagões.

Para o controle do lugar X_1 note que a rede possui inicialmente uma ficha no lugar p_1 , de modo que dependendo do comando de ultrapassar ou não ultrapassar a transição t_2 ou t_1 será acionada, respectivamente. O acionamento primeiro coloca ficha de posição "a" ou "b" na chave X_1 e posteriormente a posição "b" pela transição t_3 pós a passagem do primeiro vagão sinalizada pelo lugar p_5 . Após os dois comandos de posição serem utilizados a rede aguarda o comando externo de restabelece o controle para assim, reiniciar a rede para o estado inicial e começar um novo ciclo.

Figura 16 – Rede referente ao controle hierárquico X_1 na RPC

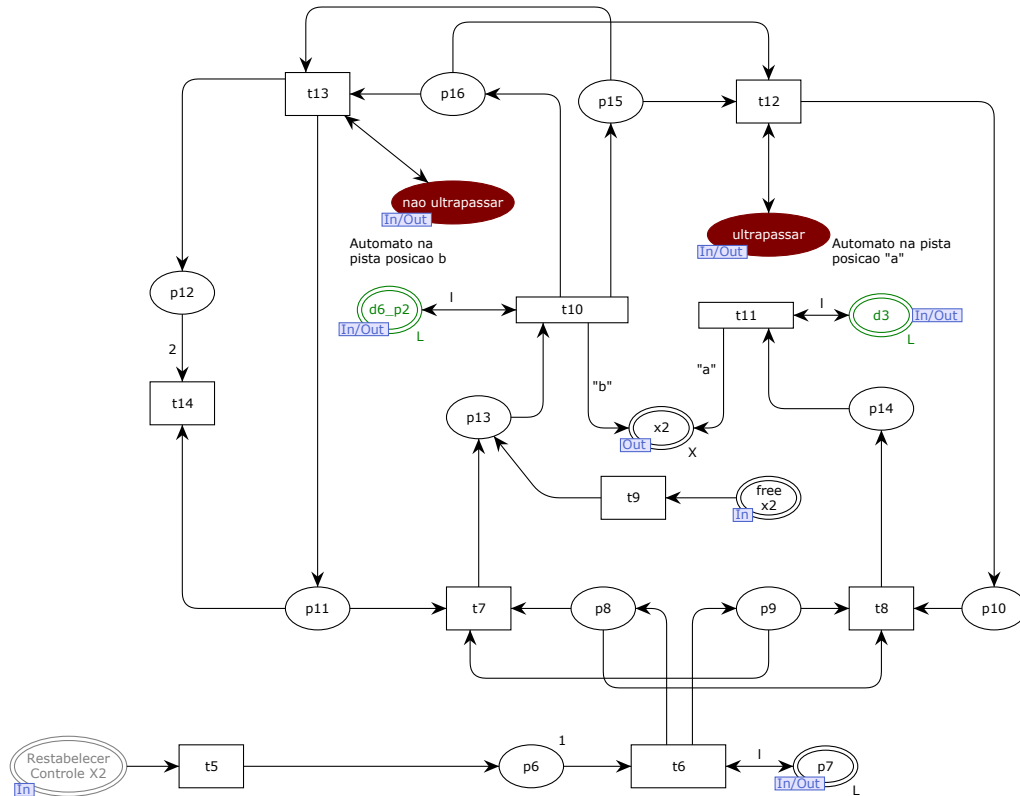


Fonte: Elaborado pelo autor.

A RPC referente a transição *Controle X_2* é demonstrado na figura 17. Tal controle tem como objetivo primeiramente receber o vagão da pista d_6 , ou seja do menor caminho e posteriormente, analisar caso tenha tido o comando de ultrapassagem o próximo vagão é esperado na pista d_3 , caso contrário virá novamente pela pista d_6 . Para alcançar tal objetivo primeiramente a estrutura da rede recebe uma ficha vinda da pista de *free* x_2 ativando a transição t_9 que espera o vagão chegar em d_6 , ao ser identificado na curva é liberada primeiramente a ficha para X_2 "b". Posteriormente, dependendo do comando relacionado à ultrapassagem é acionado t_{13} ou t_{12} , com as fichas respectivamente em p_{11} ou p_{10} , é esperado que chegar uma ficha no lugar

p_7 indicando que já pode dar o segundo comando para a chave, por fim é dado o comando de posição "a" ou "b" para a chave caso t_8 seja acionado ou t_7 , respectivamente. Ao fim do ciclo a rede espera uma ficha para restabelecer o *Controle* X_2 .

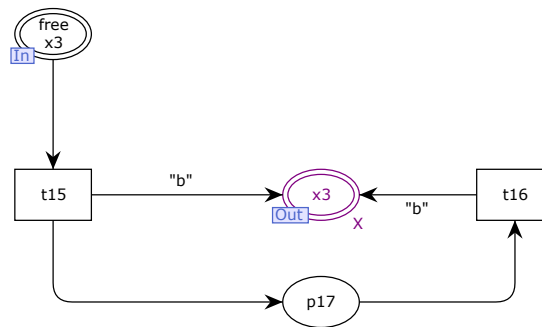
Figura 17 – Rede referente ao controle hierárquico X_2 na RPC



Fonte: Elaborado pelo autor.

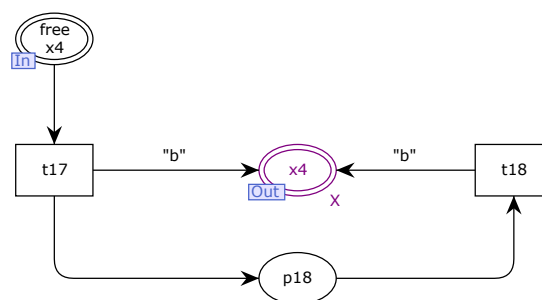
A RPC referente ao controlador X_3 e X_4 , possuem estruturas e comportamentos idênticos como demonstrados na figura 18 e 19. Como no escopo da simulação o local de ultrapassagem foi fixado no *Controle* X_1 , o objetivo do *Controle* X_3 e *Controle* X_4 é apenas garantir que uma vez liberado o controle para o controlador através do lugar *free* x_3 ou x_4 seja colocada a ficha de posição "b", ou seja o menor caminho para o vagão que vinher.

Figura 18 – Rede referente ao controle hierárquico X3 na RPC



Fonte: Elaborado pelo autor.

Figura 19 – Rede referente ao controle hierárquico X4 na RPC

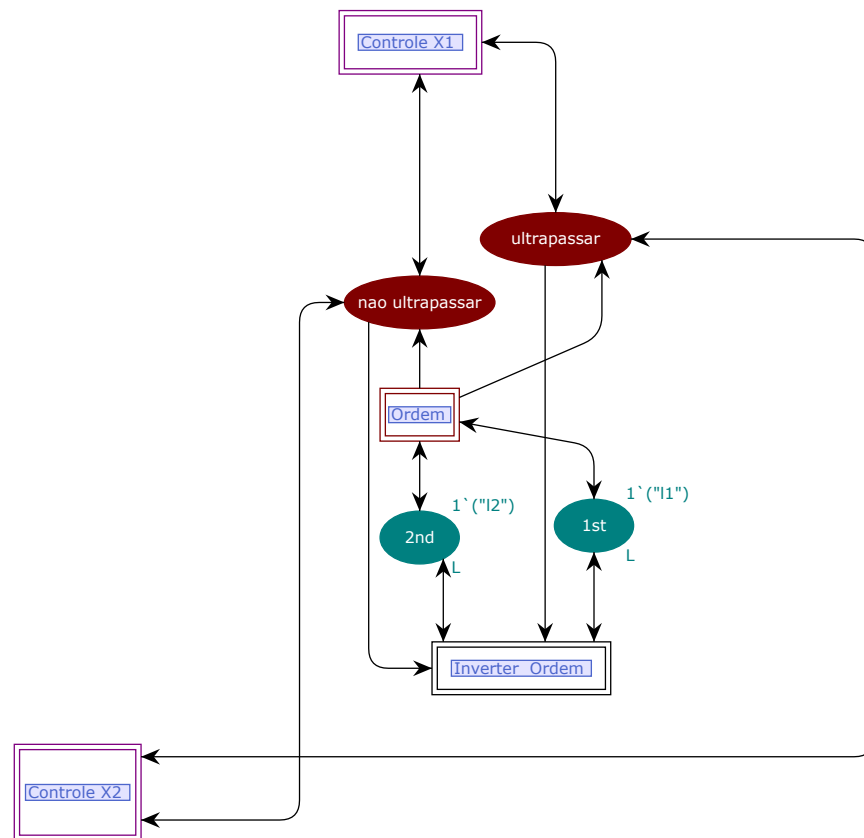


Fonte: Elaborado pelo autor.

4.2.3 Modelagem Ordenação dos vagões

Na hierarquia trabalhada referente a figura 9, que do modelo hierárquico referente a RPC, será demonstrado agora as redes relacionadas aos comando de coordenação dos vagões. Na camada de cima da hierarquia da pista as transições e lugares de ordem e inversão de ordem possuem comunicação principal com os controladores, como demonstrado na máscara do primeiro nível hierárquico na figura 20. Note que as informações de ultrapassagem e não ultrapassagem tem comunicação contro os *Controladores X1* e *X2* e quem gerencia o primeiro(1st) e segundo(2nd) lugar são os dois componentes de *Ordem* e *Inversão de Ordem*.

Figura 20 – Máscara de componentes relacionados a ordem na RPC

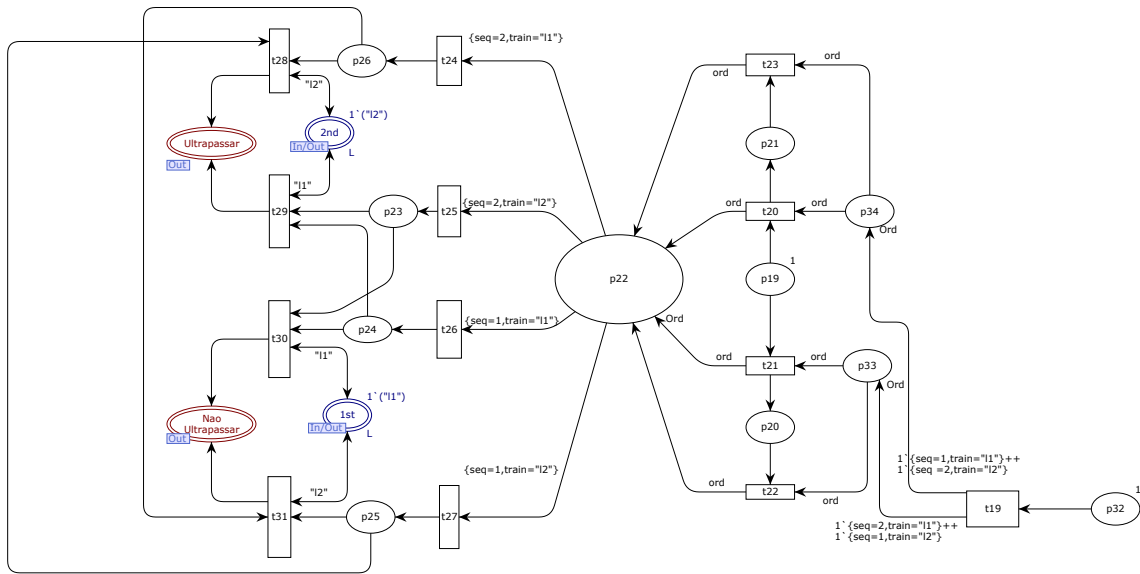


Fonte: Elaborado pelo autor.

A transição hierárquica do componente de *Ordem* é definida através da RPC demonstrada na figura 21, para melhor compreensão a mesma foi dividida em duas máscaras uma de geração de comando e outra de restabelecimento da rede para geração de um novo comando.

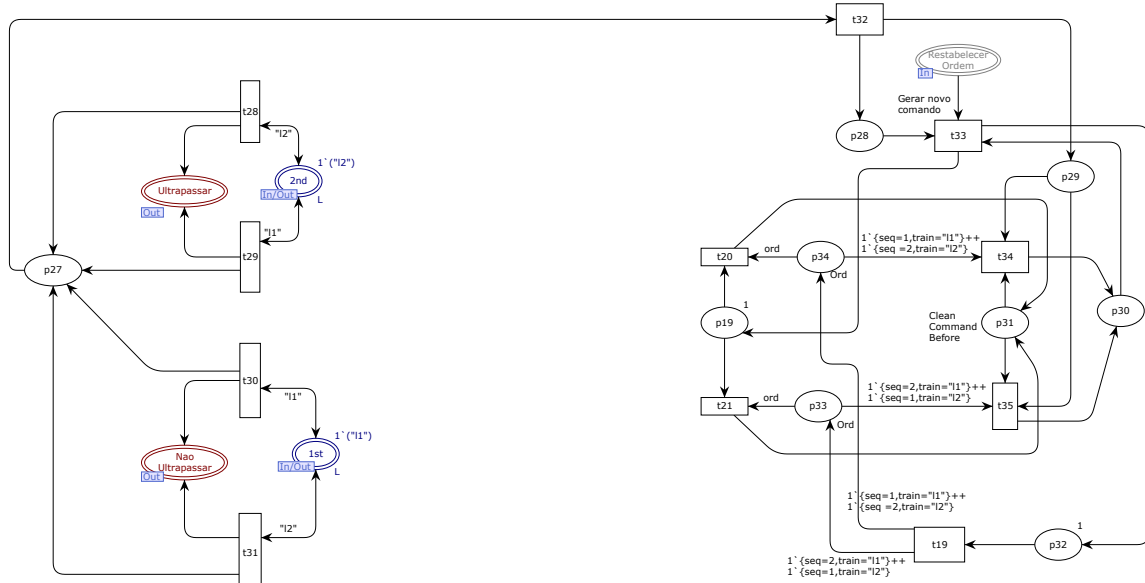
Na geração do comando referente a ultrapassagem ou não ultrapassagem a parte

Figura 22 – Máscara referente a geração de comando do componente de Ordem na RPC



Fonte: Elaborado pelo autor.

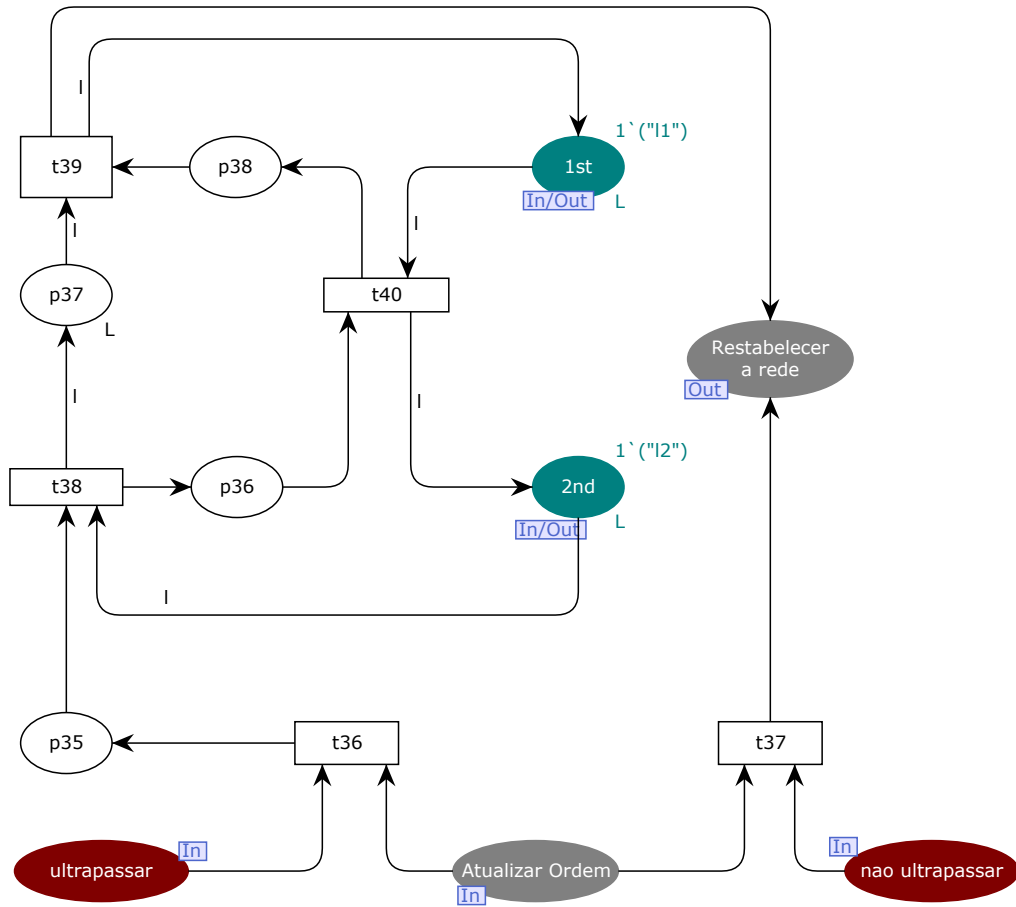
Figura 23 – Máscara referente ao restabelecimento do componente de Ordem na RPC



Fonte: Elaborado pelo autor.

comando de atualizar ordem caso haja uma ficha em ultrapassar, é ativado a transição t_{36} que tem como objetivo retirar a ficha que estava em 2^{nd} guardar temporariamente em p_{37} enquanto a ficha que estava em primeiro lugar é transferida para o segundo lugar e por fim é atualizado o primeiro lugar. Caso não tenha ocorrido ultrapassagem a transição t_{37} é acionada enviando o comando de *Restabelecer a Rede* para a hierarquia a cima.

Figura 24 – Máscara referente ao restabelecimento do componente de Ordem na RPC



Fonte: Elaborado pelo autor.

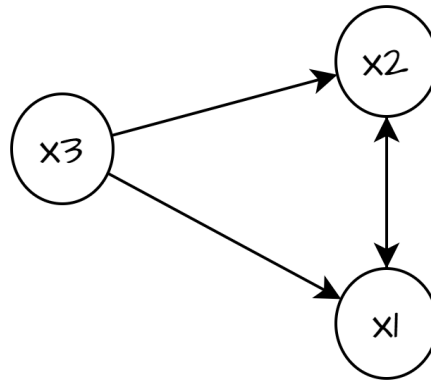
4.3 Controle Cooperativo aplicado a Multiagentes

Posteriormente à detalhada modelagem do sistema na seção 4.2 que foi definida os principais eventos do sistema, além da arquitetura da rede que gera algumas regras para o deslocamento dos vagões é buscada um controle de nível superior responsável por otimizar a referência de posição de cada vagão. Para o desenvolvimento do controle cooperativo entre os vagões com o objetivo de evitar colisões e promover a otimização das trajetórias é importante garantir que as transições na rede de petri possuem acionamentos praticamente instantâneos não influenciando assim de forma significativa na dinâmica do sistema.

Para a coordenação e intercomunicação entre os agentes e a referência (agente virtual ou líder), foi escolhido o modelado o seguinte grafo 25. Tal que x_1 e x_2 , são os dois agentes da planta e o líder e dado por x_3 . Note, que o grafo é classificado com um *direct tree*, pois existe somente um líder o qual possui grau de entrada nulo, como apresentado na subseção 2.3.2.

Foram formulados dois tipos de problemática para o controle, o primeiro relacionado

Figura 25 – Modelagem do grafo entre agentes



Fonte: Elaborado pelo autor.

aos momentos que acontecem uma situação de ultrapassagem no ciclo de uma volta partindo e chegando em d_1 na pista da figura 8, e o segundo o ciclo de uma volta sem ultrapassagem.

4.3.1 Casos de Ultrapassagem

Para a simulação e comportamento do controle por consenso nos casos de ultrapassagem foram escolhidas 4 referências diferentes para demonstrar o comportamento do sistema de acordo com a tabela 1. Por meio da simulação com todos os pesos nos arcos iguais a 0.5 é obtido

referência	trajetória
0.5	d_1 para d_6 e d_3
-0.5	d_3 e d_6 para d_2
-0.2	d_2 para d_5
0	d_5 para d_1

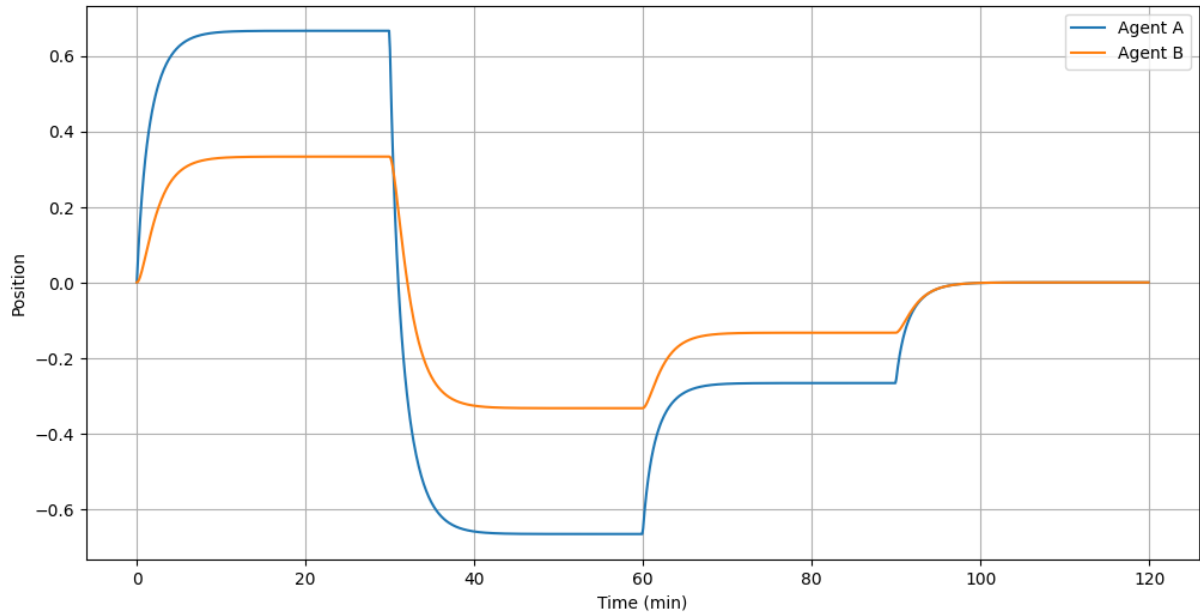
Tabela 1 – Escolha de referências ao longo do tempo com inversão de sinal

a seguinte o gráfico da figura 26. Note que inicialmente é praticamente considerado que os dois iniciam praticamente junto e vão se separando no tempo de 0 a 20 minutos, até que ao chegar aproximadamente em torno de 30 segundo a referência é alterada e é feita uma ultrapassagem nos momentos seguintes as colocações se mantêm mas a distância entre os agentes vai cada vez mais diminuindo coma escolha das referências.

A partir da figura 27 é visualizado o comportamento da velocidade entre os agentes assim como as mudanças de referência, note que mesmo o líder submetido a valores constantes de referência, o sistema gradualmente estabiliza naquele ponto de referência, gerando assim uma mudança de referência mais suave.

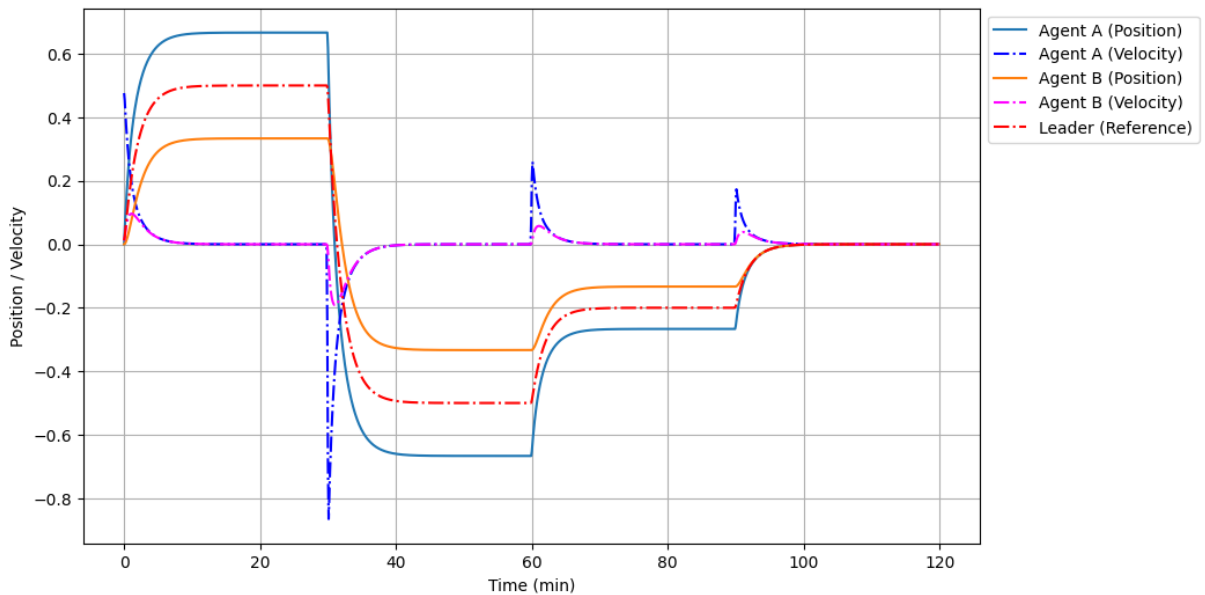
Um segundo resultado é obtido diminuindo a em torno de 4x a comunicação com o

Figura 26 – Simulação Caso de Ultrapassagem



Fonte: Elaborado pelo autor.

Figura 27 – Análise velocidade Caso de Ultrapassagem

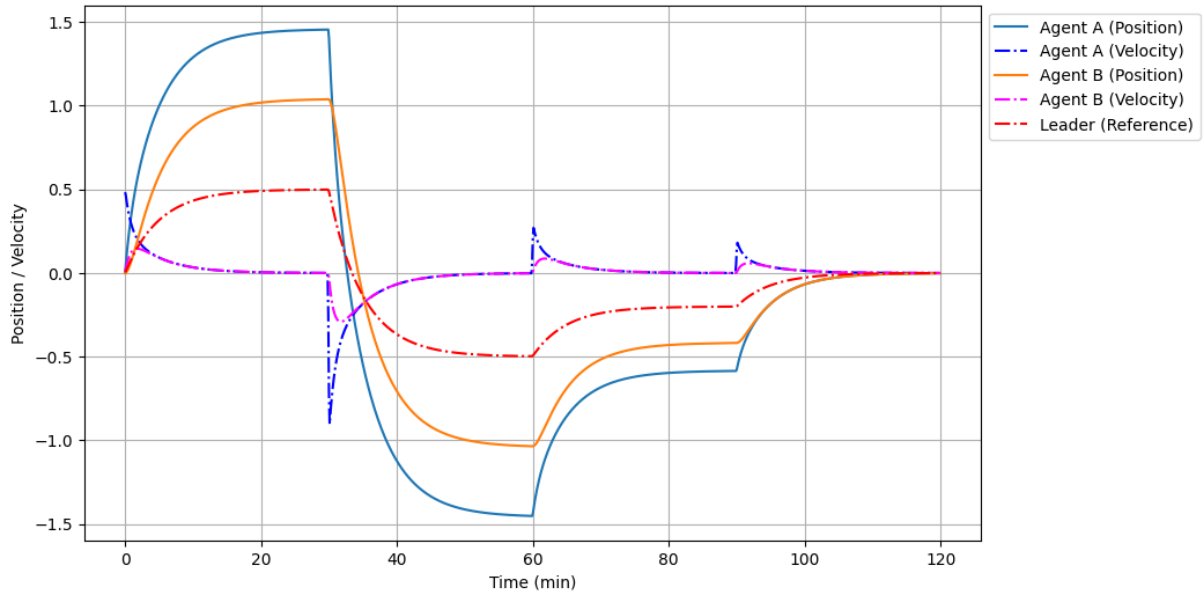


Fonte: Elaborado pelo autor.

líder, mas mantendo o mesmo valor de comunicação entre os agentes. Tal exemplo é demonstrada na figura 28, note que para esse caso a formação dos agentes se mantém ao longo de todas as mudanças de referência, o que muda é a velocidade em que os agentes alcançam a velocidade relativa de zero. Na aplicação proposta esse comportamento não é desencorajado uma vez que o foco tem sido o mantimento da formação nos casos de ultrapassagem.

Ainda na análise com os valores de referência descritos na tabela 1, um importante

Figura 28 – Análise Diminuição do peso da comunicação com o líder no Caso de Ultrapassagem



Fonte: Elaborado pelo autor.

ponto a ser realçado é que para manter a mesma configuração com a ultrapassagem somente no ponto onde a referência é dada como negativa é importante que o peso dos arcos entre os agentes seja semelhantes, pois a medida que a diferença aumenta a velocidade em que um agente procura a referência é maior que a do outro, podendo ocasionar uma segunda ultrapassagem, exemplo desse comportamento evitado para a aplicação dada é demonstrado na figura 29. Note que o perfil de velocidade se assemelha nos dois casos, mas na posição é possível perceber que em torno de 90 min o agente A vai de encontro a referência do líder mais rapidamente que o Agente B, ocasionando uma segunda ultrapassagem, e de fato para essa simulação o peso do arco do agente A em relação com o líder em relação ao agente B com o líder é em torno de 4x maior.

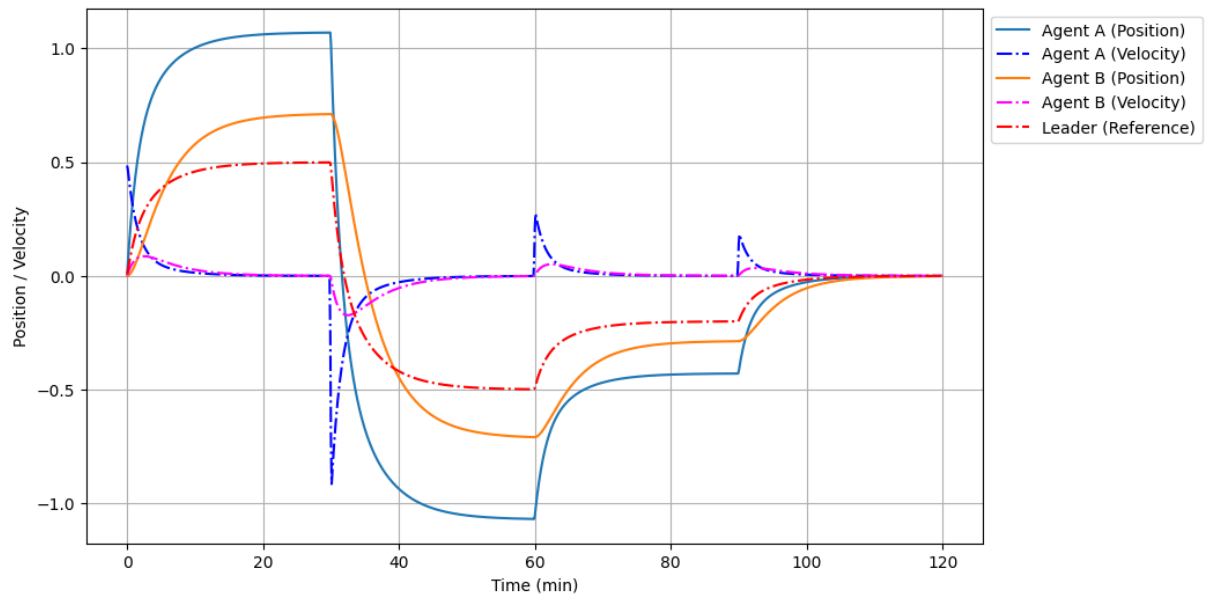
4.3.2 Casos de Não Ultrapassagem

O segundo grupo de simulações é relacionado aos casos onde não é necessário que ocorra ultrapassagem entre os agentes, ou seja a formação inicial seja mantida ao longo das mudanças de referência. Para isso é utilizado os valores de referência de acordo com a a tabela 2

referência	trajetória
0.2	d_1 para d_6 e d_3
0.5	d_3 e d_6 para d_2
0.2	d_2 para d_5
0	d_5 para d_1

Tabela 2 – Escolha de referências ao longo do tempo sem inversão de sinal

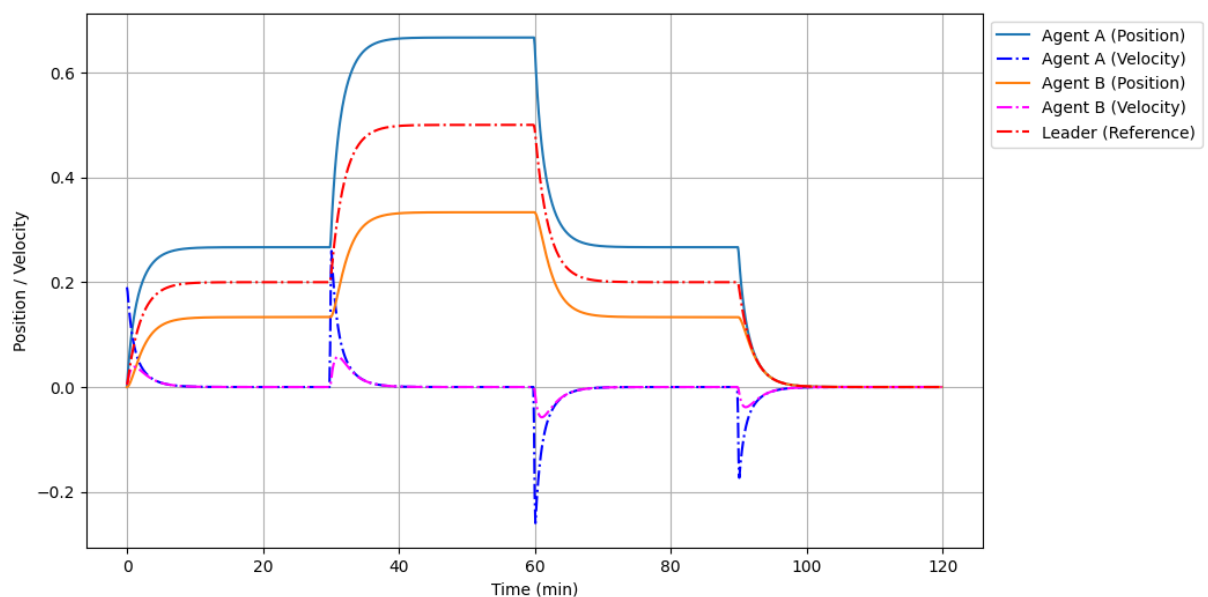
Figura 29 – Análise pesos muito diferentes na comunicação com o líder Caso de Ultrapassagem



Fonte: Elaborado pelo autor.

Por meio da simulação com todos os pesos nos arcos iguais a 0.5 é obtido a seguinte o gráfico da figura 30. Note que inicialmente é considerado que os dois iniciam praticamente junto e vão se separando no tempo de 0 a 20 minutos, e em quanto o valor de referência sobe, a distância entre eles aumenta até o momento que a referência começa a diminuir. Note que a formação do grupo de agentes se mantém a mesma ao longo de diferentes valores de referência positiva.

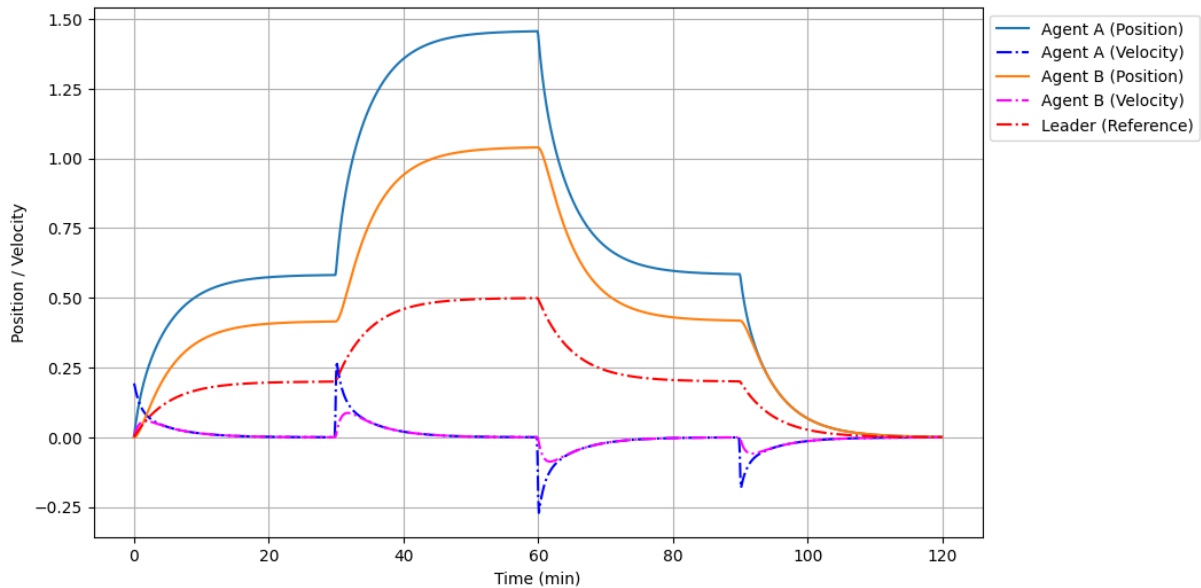
Figura 30 – Simulação Caso de não ultrapassagem



Fonte: Elaborado pelo autor.

De forma análoga um segundo resultado é obtido diminuindo em torno de 4x a comunicação com o líder, mas mantendo o mesmo valor de comunicação entre os agentes. Tal exemplo é demonstrada na figura 31, note que para esse caso a formação dos agentes se mantém ao longo de todas as mudanças de referência, o que muda é a velocidade em que os agentes alcançam a velocidade relativa de zero.

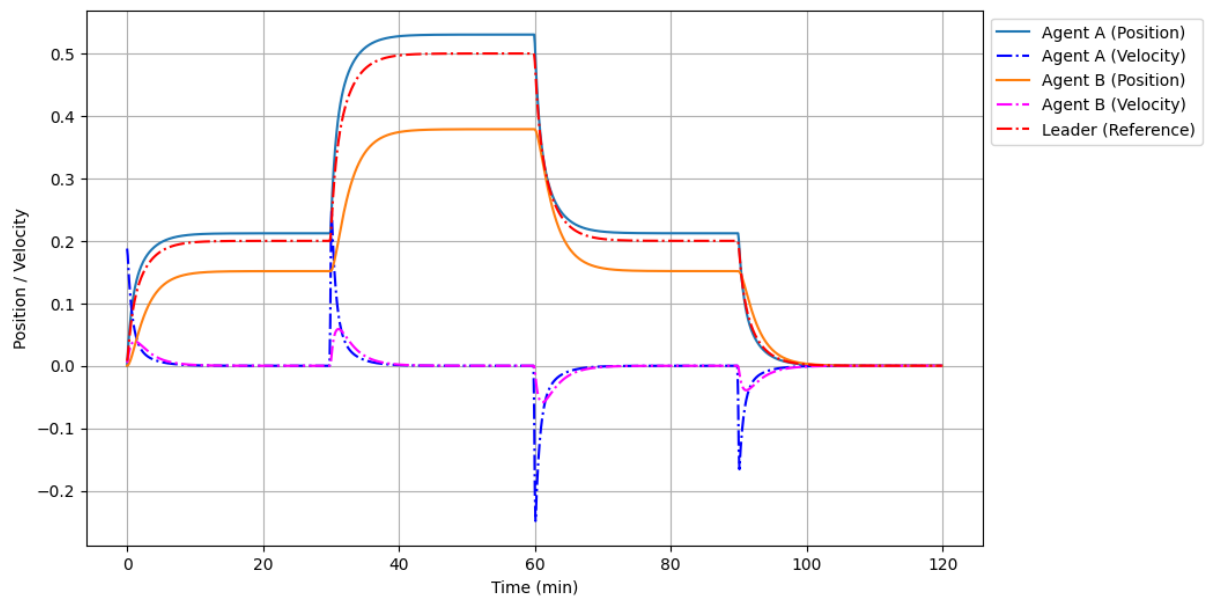
Figura 31 – Simulação Caso de não ultrapassagem, diminuição peso comunicação com o líder



Fonte: Elaborado pelo autor.

Por fim, para fins de análise, é feito o testes de pesos desbalanceados em relação ao líder, o resultado é demonstrado na figura 32, note que o mesmo tipo de comportamento acontece como demonstrado no caso de ultrapassagem, em que a formação é alterada perto do ponto de referência nula do líder. No caso de manter a formação tal comportamento não é desejado.

Figura 32 – Simulação Caso de não ultrapassagem, grande diferença peso na comunicação com o líder



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Este trabalho realizou uma abordagem integrada de modelagem e controle em sistemas industriais, utilizando Redes de Petri Colorida e estratégias de controle cooperativo. O método proposto de modelagem e controle foi aplicado a um problema prático¹ envolvendo uma planta industrial, abordando a sincronia e formação de autômatos ao longo de trajetórias definidas. Eventos modelados pela rede de Petri foram utilizados para alterar a formação e os pontos de sincronia do grupo de autômatos.

A implementação de um modelo detalhado da planta industrial por meio de Redes de Petri Colorida mostrou-se eficaz e intuitiva, oferecendo uma visualização hierárquica e orientada a eventos, além de facilitar a comunicação entre os componentes da planta. Assim, as Redes de Petri Colorida destacaram-se como uma ferramenta fundamental na modelagem precisa e abstrata de sistemas complexos, contribuindo para uma compreensão aprofundada e eficiente do funcionamento da planta industrial.

O método de abordagem integrada revelou-se viável e eficiente, especialmente em aplicações com muitos agentes, já que o controle por consenso proporciona uma implementação simples, sem grande demanda matemática, fornecendo a sincronia necessária para a aplicação de uma 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. 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 comportamentos ao longo do sistema.

A principal contribuição deste trabalho consistiu no uso da técnica integrada de modelagem e controle que diminui o processamento local em cada agente, deixando assim as lógicas de processamento centralizadas em um sistema supervisor modelado via rede de petri, além de uma lógica de controle de baixo custo computacional. Todavia é necessário ainda haver uma ótima comunicação entre os agentes, pois a base do controle é dada pela sincronia entre os estados do agente vizinho.

¹ Todas os códigos estão disponíveis em https://github.com/Davidap041/Master_Dissertation.git