

Universidade Federal de Juiz de Fora  
Faculdade de Engenharia  
Programa de Pós Graduação em Engenharia Elétrica

**Marina Borges Arantes de Souza**

**Técnicas de Otimização para Controle e Operação de Máquinas Inteligentes**

Juiz de Fora

2017

**Marina Borges Arantes de Souza**

**Técnicas de Otimização para Controle e Operação de Máquinas Inteligentes**

Dissertação apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia Elétrica, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Edimar José de Oliveira

Coorientador: Leonardo Willer de Oliveira

Juiz de Fora

2017

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Souza, Marina Borges Arantes de.

Técnicas de Otimização para Controle e Operação de Máquinas Inteligentes / Marina Borges Arantes de Souza. – 2017.

87 f. : il.

Orientador: Edimar José de Oliveira

Coorientador: Leonardo Willer de Oliveira

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós Graduação em Engenharia Elétrica, 2017.

1. Programação Não Linear. 2. Sistemas de Múltiplos Robôs. 3. Coordenação de Robôs. 4. Pontos Fictícios. I. Oliveira, Edimar José de, orient. II. Willer de Oliveira, Leonardo, coorient. III. Título.

Marina Borges Arantes de Souza

Técnicas de Otimização para Controle e Operação de Máquinas Inteligentes

Dissertação apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia Elétrica, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em: 28 de agosto de 2017.

BANCA EXAMINADORA

  
\_\_\_\_\_  
Prof. Dr. Edimar José de Oliveira - Orientador  
Universidade Federal de Juiz de Fora

  
\_\_\_\_\_  
Prof. Dr. Leonardo Willer de Oliveira - Coorientador  
Universidade Federal de Juiz de Fora

  
\_\_\_\_\_  
Prof. Dr. Carlos Henrique Valério de Moraes  
Universidade Federal de Itajubá

  
\_\_\_\_\_  
Prof. Dr. Leonardo de Mello Honório  
Universidade Federal de Juiz de Fora

## AGRADECIMENTOS

Primordialmente, agradeço a Deus pelo dom da vida e por me iluminar, trazendo clareza de qual caminho seguir.

Agradeço imensamente aos meus pais, Walter e Sandryonara, que estiveram sempre ao meu lado, me incentivando e me ajudando a passar pelos momentos de alegria e de dificuldades. A toda minha família, meu muito obrigada.

Ao meu noivo Vitor por estar presente em todas as horas, me dando suporte, carinho e atenção.

O meu agradecimento especial, ao meu orientador Prof. Edimar Oliveira, que acreditou em mim e me deu a oportunidade de poder aprender um pouco com seus exemplos e experiências. Com certeza, foram fundamentais para a minha formação. Ao meu coorientador Prof. Leonardo Willer que sempre esteve disposto a ajudar.

Aos membros da banca examinadora por todas contribuições e considerações.

Ao corpo docente do Programa de Pós Graduação em Engenharia Elétrica da UFJF por todos ensinamentos recebidos.

A todos os colegas do PPEE que me ajudaram a concluir essa etapa com maior leveza, especialmente Luana, Saulo, Arthur, Marcelo e Igor.

Ao LABSPOT (Laboratório de Sistemas de Potência da Faculdade de Engenharia Elétrica), pelo suporte técnico e ao CNPq pelo auxílio financeiro que possibilitou a realização desse trabalho.

Por fim, agradeço a todos que contribuíram de alguma forma.

“Quanto mais aumenta nosso conhecimento,  
mais evidente fica nossa ignorância”  
(John F. Kennedy, 1962)

## RESUMO

Levando em conta a crescente utilização de sistemas robóticos em várias situações da atualidade, métodos que coordenam as atividades dos robôs são essenciais para se obter movimentos sincronizados e livres de possibilidades de colisão. Uma forma de coordená-los é através de métodos de otimização. O presente trabalho enfoca uma abordagem baseada em Programação Não Linear para encontrar perfis de velocidade ótimos para robôs com caminhos previamente especificados. A metodologia é aplicada em modelos de robôs móveis e manipuladores robóticos. Apesar das diferenças construtivas, de forma geral e para fins de coordenação, é permitido compartilhar, para os dois casos, a mesma formulação de otimização, fundamentada na maximização do quadrado da diferença de tempo em que os robôs atingem um mesmo ponto de colisão. Não obstante o grande número de trabalhos encontrados na literatura que envolvem o assunto, abordagens relacionadas a uma modelagem não linear do problema são escassas. A vantagem do método encontra-se na facilidade de representar não linearidades do sistema, como limitações de velocidade, aceleração e torque dos robôs. Além disso, a complexidade de formulação e resolução é reduzida em comparação com outros métodos que envolvem o tratamento de variáveis inteiras. O método também engloba situações em que os robôs podem se colidir na forma de segmentos. Para tanto, determina-se uma abordagem baseada na inclusão de pontos fictícios para representar tais trechos. Os testes foram realizados em diferentes sistemas de robôs móveis e manipuladores e os resultados comprovaram a eficiência da metodologia proposta, encontrando perfis de velocidade ótimos que determinam trajetórias sem acidentes. Comparações com Programação Linear Inteira Mista, amplamente utilizada para esse fim, comprovaram a superioridade da técnica apresentada, com relação à complexidade e modelagem de não linearidades, aproximando o modelo a sistemas reais.

Palavras-chave: Programação Não Linear. Sistemas de múltiplos robôs. Coordenação de robôs. Pontos fictícios.

## ABSTRACT

The use of robotic systems in various industrial and logistics situations is increasing. Methods that coordinate the activities of the robots are essential to obtain synchronized and free of collision possibilities movements. One manner to coordinate them is through optimization methods. This work focuses on an approach based on Nonlinear Programming to determine optimal velocity profiles of robots with previously specified paths. The methodology is applied to mobile robots and manipulators models. Despite the constructive differences, in general, the same optimization formulation can be shared for both cases. The arrangement is based on the maximization of the square of the time difference in which the robots reach the same point of collision. Notwithstanding a large number of papers found in the literature involving the subject, approaches related to a nonlinear modeling of the problem are scarce. The advantage of the method lies in the easiness of representing nonlinearities of the system, such as speed, acceleration and torque limitations of robots. In addition, the formulation and resolution complexity is reduced compared to other methods encompassing the processing of integer variables. The method also encompasses situations where robots may collide in the form of segments. In this sense, an approach based on the inclusion of fictitious points to represent such stretches is determined. The tests were realized in different mobile robot and manipulators systems. The results proved the efficiency of the proposed methodology, finding optimal speed profiles that determine trajectories without accidents. Comparisons with Mixed Integer Linear Programming, widely used for this purpose, proved the superiority of the proposed technique with respect to the complexity and modeling of nonlinearities, bringing the model closer to real systems.

Key-words: Nonlinear Programming. Multi-robot systems. Coordination of robots. Fictitious points.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Atribuição de tarefas e caminhos seguidos por três robôs em uma planta industrial . . . . .	18
Figura 2 – Sistema composto por doze robôs e suas respectivas configurações de caminhos: (a) Radiais com simetria, (b) Linha reta e circulares, (c) Aleatória . . . . .	20
Figura 3 – Sistema composto por três robôs (triângulos) que devem percorrer caminhos cíclicos com interseções (pontos pretos). . . . .	22
Figura 4 – Caminhos a serem percorridos por quatro robôs com vinte pontos de colisão. . . . .	25
Figura 5 – Layout industrial do caso de teste de coordenação de manipuladores. . . . .	25
Figura 6 – Casos de conflito de trens num sistema ferroviário. . . . .	28
Figura 7 – Esquema de carregamento no armazém. . . . .	29
Figura 8 – Sistema composto por dois robôs e quatro pontos de colisão. . . . .	33
Figura 9 – Sistema composto por dois robôs com alto risco de colisão. . . . .	34
Figura 10 – Sistema composto por dois robôs com caminhos coincidentes. . . . .	35
Figura 11 – Inclusão de pontos fictícios para representar o segmento de colisão. . . . .	36
Figura 12 – Fluxograma de solução do problema de Coordenação de Robôs Móveis. . . . .	43
Figura 13 – Sistema de manipuladores compartilhando um mesmo espaço de trabalho. . . . .	45
Figura 14 – Possíveis formatos para trajetória de uma junta. . . . .	46
Figura 15 – Fluxograma de solução do problema de Coordenação de Robôs Manipuladores. . . . .	52
Figura 16 – Esquemas de sistemas multi-robóticos: (a) 3 robôs, (b) 4 robôs, (c) 5 robôs. . . . .	55
Figura 17 – Esquema de teste com três robôs. . . . .	56
Figura 18 – Esquema de teste com quatro robôs. . . . .	60
Figura 19 – Esquema de teste com cinco robôs. . . . .	63
Figura 20 – Detalhe da inclusão dos pontos fictícios no Esquema C. . . . .	64
Figura 21 – Esquemas de sistemas de múltiplos manipuladores: (a) 3 robôs, (b) 6 robôs. . . . .	69
Figura 22 – Manipulador Epson PROSIX C3-A601C. . . . .	70
Figura 23 – Esquema de teste com três manipuladores. . . . .	71
Figura 24 – Velocidade angular (rad/s) das articulações ao longo tempo (s) para cada robô - Esquema D: (a) $M_1$ , (b) $M_2$ , (c) $M_3$ . . . . .	73
Figura 25 – Passagem de cada robô pelo ponto de colisão: (a) $M_2$ em $t=0,31s$ , (b) $M_3$ em $t=0,86s$ , (c) $M_1$ em $t=1,40s$ . . . . .	73
Figura 26 – Esquema de teste com seis manipuladores. . . . .	74
Figura 27 – Velocidade angular (rad/s) das articulações ao longo tempo (s) para cada robô - Esquema E: (a) $M_1$ , (b) $M_2$ , (c) $M_3$ , (c) $M_4$ , (c) $M_5$ , (c) $M_6$ . . . . .	75

## LISTA DE TABELAS

Tabela 1 – Limites de velocidade - Esquema A. . . . .	57
Tabela 2 – Comprimentos dos trechos - Esquema A. . . . .	57
Tabela 3 – Média de velocidade e tempo em cada trecho - Esquema A com restrições de rampa. . . . .	57
Tabela 4 – Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema A. . . . .	58
Tabela 5 – Média de velocidade e tempo em cada trecho - Esquema A sem restrições de rampa. . . . .	58
Tabela 6 – Limites de velocidade - Esquema B. . . . .	59
Tabela 7 – Comprimentos dos trechos - Esquema B. . . . .	59
Tabela 8 – Média de velocidade e tempo em cada trecho - Esquema B com restrições de rampa. . . . .	60
Tabela 9 – Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema B. . . . .	61
Tabela 10 – Média de velocidade e tempo em cada trecho - Esquema B sem restrições de rampa. . . . .	62
Tabela 11 – Limites de velocidade - Esquema C. . . . .	63
Tabela 12 – Comprimentos dos trechos - Esquema C. . . . .	63
Tabela 13 – Comprimentos dos trechos após inclusão dos pontos fictícios - Esquema C. . . . .	64
Tabela 14 – Média de velocidade e tempo em cada trecho - Esquema C com restrições de rampa. . . . .	65
Tabela 15 – Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema C. . . . .	65
Tabela 16 – Média de velocidade e tempo em cada trecho - Esquema C sem restrições de rampa. . . . .	66
Tabela 17 – Comparações entre resultados de PLIM e PNL - robôs móveis. . . . .	67
Tabela 18 – Comparações entre formulações PLIM e PNL - robôs móveis. . . . .	68
Tabela 19 – Limites de velocidade angular para cada junta do manipulador. . . . .	70
Tabela 20 – Configurações angulares de passagem - Esquema D. . . . .	71
Tabela 21 – Média de velocidade das articulações e tempo em cada trecho - Esquema D. . . . .	72
Tabela 22 – Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema D. . . . .	72
Tabela 23 – Configurações angulares de passagem - Esquema E . . . . .	75
Tabela 24 – Média de velocidade das articulações e tempo em cada trecho - Esquema E. . . . .	76

Tabela 25 – Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema E. . . . .	76
Tabela 26 – Comparação entre PNL e PLIM - manipuladores robóticos . . . . .	77

## LISTA DE ABREVIATURAS E SIGLAS

MILP	<i>Mixed Integer Linear Programming</i>
NLP	<i>Nonlinear Programming</i>
NP	<i>Nondeterministic Polynomial Time</i>
PLIM	Programação Linear Inteira Mista
PNL	Programação Não Linear
PNLIM	Programação Não Linear Inteira Mista
PPEE	Programa de Pós Graduação em Engenharia Elétrica
SMR	Sistema Multi-Robóticos
UAV	<i>Unmanned Aerial Vehicles</i>
UFJF	Universidade Federal de Juiz de Fora

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>13</b>
1.1	CONSIDERAÇÕES INICIAIS . . . . .	13
1.2	MOTIVAÇÃO . . . . .	14
1.3	DEFINIÇÃO DO PROBLEMA E OBJETIVOS . . . . .	14
1.4	PUBLICAÇÕES DECORRENTES DA PESQUISA . . . . .	15
1.5	ESTRUTURA DA DISSERTAÇÃO . . . . .	15
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>17</b>
2.1	INTRODUÇÃO . . . . .	17
2.2	TÉCNICAS DE COORDENAÇÃO DE SMR . . . . .	18
2.3	COORDENAÇÃO EM SISTEMAS COM TRECHOS COMUNS . . . . .	26
2.4	CONSIDERAÇÕES FINAIS . . . . .	30
<b>3</b>	<b>METODOLOGIA PROPOSTA . . . . .</b>	<b>32</b>
3.1	INTRODUÇÃO . . . . .	32
3.2	COORDENAÇÃO DE MÚLTIPLOS ROBÔS MÓVEIS . . . . .	32
3.2.1	<b>Descrição do Problema . . . . .</b>	<b>32</b>
3.2.2	<b>Representação dos Trechos Coincidentes . . . . .</b>	<b>34</b>
3.2.3	<b>Modelagem do Problema de Otimização . . . . .</b>	<b>36</b>
3.2.3.1	Função Objetivo . . . . .	37
3.2.3.2	Restrições . . . . .	38
3.2.3.3	Formulação Proposta . . . . .	40
3.2.4	<b>Processo de Resolução . . . . .</b>	<b>41</b>
3.3	COORDENAÇÃO DE MÚLTIPLOS ROBÔS MANIPULADORES . . . . .	43
3.3.1	<b>Descrição do Problema . . . . .</b>	<b>44</b>
3.3.2	<b>Geração de Trajetórias . . . . .</b>	<b>45</b>
3.3.3	<b>Modelagem do Problema de Otimização . . . . .</b>	<b>48</b>
3.3.3.1	Função Objetivo . . . . .	49
3.3.3.2	Restrições . . . . .	49
3.3.3.3	Formulação Proposta . . . . .	50
3.3.4	<b>Processo de Resolução . . . . .</b>	<b>51</b>
3.4	FORMA DE RESOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO . . . . .	53
3.5	CONSIDERAÇÕES FINAIS . . . . .	53
<b>4</b>	<b>ESTUDO DE CASOS . . . . .</b>	<b>55</b>
4.1	SISTEMAS DE MÚLTIPLOS ROBÔS MÓVEIS . . . . .	55

4.1.1	Sistema com três robôs . . . . .	56
4.1.2	Sistema com quatro robôs . . . . .	59
4.1.3	Sistema com cinco robôs . . . . .	61
4.1.4	Comparações com PLIM . . . . .	66
4.2	SISTEMAS DE MÚLTIPLOS MANIPULADORES ROBÓTICOS . . .	68
4.2.1	Sistema com três manipuladores . . . . .	70
4.2.2	Sistema com seis manipuladores . . . . .	74
4.2.3	Comparações com PLIM . . . . .	77
4.3	CONSIDERAÇÕES SOBRE CONVEXIDADE . . . . .	77
<b>5</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>79</b>
5.1	CONCLUSÃO . . . . .	79
5.2	TRABALHOS FUTUROS . . . . .	81
	<b>REFERÊNCIAS . . . . .</b>	<b>82</b>
	<b>APÊNDICE A – RESUMO DO MÉTODO DE PONTOS IN- TERIORES - BARREIRA LOGARÍTMICA</b>	<b>85</b>

# 1 INTRODUÇÃO

## 1.1 CONSIDERAÇÕES INICIAIS

Devido ao desenvolvimento tecnológico e científico, particularmente nas áreas de eletrônica, computação, controle e automação, o número de robôs utilizados na indústria e logística tem aumentado de forma a atender novas demandas. Tanto robôs móveis quanto manipuladores são amplamente utilizados para atender tarefas que exigem boa repetibilidade. Sistema Multi-Robôs (SMR), podem ser empregados em algumas situações específicas, como em monitoramento e varredura de plantas industriais e também em tarefas que exigem periodicidade e ações em conjunto.

Em robótica móvel, navegação é a habilidade do veículo contornar um caminho usando uma decisão autônoma ou híbrida (semi-autônoma). Um robô móvel autônomo deve ser capaz de perceber o território ao seu redor e decidir sobre a melhor ação a ser realizada, levando a um erro mínimo da tarefa objetivo. De forma simplificada, o problema da navegação autônoma consiste em assegurar que o robô navegue através de sua trajetória sem colidir com obstáculos, desde um ponto de partida até um ponto final [1]. Vários veículos autônomos podem ser necessários em aplicações específicas, como o monitoramento de grandes regiões, onde a trajetória cíclica de múltiplos robôs deve ser avaliada para evitar a colisão entre os robôs. A utilização de múltiplos veículos autônomos em vez de um, pode melhorar o desempenho da tarefa, reduzindo, por exemplo, o tempo de execução para cobrir certas regiões, melhorando a eficiência.

Já os robôs manipuladores são empregados em diversas áreas, tendo abrangente aplicabilidade na indústria. Atividades como pintura, montagem, soldagem e manuseio de materiais estão no rol de aplicações cujos manipuladores podem ser incluídos. Assim como os robôs móveis, eles também devem realizar uma ação que leve a seu objetivo com menor possibilidade de erro possível. A movimentação desse robô deve ser acompanhada pelo controle sincronizado de suas juntas para que ele atinja com êxito suas tarefas. Algoritmos de pesquisa inteligentes podem ser aplicados para encontrar um melhor caminho e posição do efetuador (órgão terminal ou ferramenta) do robô, criando trajetórias seguras [2]. Em um sistema com vários braços articulados, os movimentos de suas articulações devem ser coordenados para garantir que o conjunto de trajetórias cobertas pelos robôs esteja livre de colisões.

A utilização de Programação Linear Inteira Mista (PLIM), do inglês *Mixed Integer Linear Programming* (MILP), é recorrente em trabalhos de coordenação de movimentos de robôs em geral. No caso de robôs móveis, determina-se o perfil de velocidade linear que evite a colisão entre os veículos pertencentes ao sistema [3] e [4]. No sistema composto por braços robóticos, deve-se encontrar o perfil de velocidade de suas articulações que atenda às restrições físicas do manipulador e que evite colisões nos pontos comuns das trajetórias

[5]. Nessas situações, as restrições dinâmicas do robô, tais como os limites de velocidade e aceleração, são consideradas, além das restrições de tempo e ciclos realizadas, fazendo com que o desenvolvimento seja complexo.

O desenvolvimento de uma técnica de Programação Não Linear (PNL), do inglês, *Nonlinear Programming* (NLP), simplifica a metodologia e proporciona um ganho computacional ao sistema simulado. Adicionalmente à simplificação da formulação pode ser citado como vantagem a possibilidade de tratamento de não linearidades pertencentes ao processo.

## 1.2 MOTIVAÇÃO

O desenvolvimento deste trabalho é motivado pela utilização de Programação Não Linear que determina uma vantagem, por ser um método mais robusto frente às metodologias apresentadas pela literatura, englobando variáveis não lineares inerentes aos sistemas robóticos reais, que possuem características dinâmicas limitadas. Tais limitações são modeladas por restrições não lineares, que é o caso envolvendo velocidade, aceleração e torque ao longo do tempo.

Outro ponto a ser destacado é o caso de trechos em que os robôs podem se colidir. Não obstante o elevado número de trabalhos focados em encontrar a configuração ideal de múltiplos robôs, a literatura não oferece formas de resolver problemas que abordam trechos de colisão. Em algumas situações, podem-se encontrar casos em que os veículos tem o mesmo caminho a percorrer. Portanto, os métodos que evitam a colisão pontual não são ideais para resolver esta situação. Uma variação da formulação PNL para a adaptar ao caso dos segmentos de colisão, também é uma contribuição relevante do presente trabalho.

## 1.3 DEFINIÇÃO DO PROBLEMA E OBJETIVOS

Considerando um grupo de robôs que devem seguir caminhos especificados de forma a completar suas tarefas em um determinado tempo, o objetivo é coordenar seus movimentos a fim de evitar possíveis colisões em pontos ou segmentos comuns de suas trajetórias. A trajetória é compreendida pelo caminho em que o robô deverá percorrer e o perfil de velocidade ao longo do tempo. Os robôs devem iniciar seus movimentos ao mesmo tempo. Essa coordenação não deve contemplar possíveis modificações nos caminhos, ou seja, ele não deve alterar a rota planejada. Além disso, também não há a possibilidade de interrupção do movimento como forma de não ocorrerem acidentes ao longo de suas tarefas.

O objetivo principal é comprovar a eficiência da utilização de um método de otimização baseado em Programação Não Linear num problema dessa natureza. A metodologia proposta abrange sistemas de múltiplos robôs compostos por robôs móveis

e também manipuladores. A modelagem do movimento dos robôs móveis considera a velocidade de cada um. Já para os movimentos dos manipuladores, deve-se considerar o perfil de velocidade e aceleração das juntas. Com o objetivo de encontrar o perfil de velocidade ótimo para ambos os tipos de robôs, uma formulação baseada na maximização do tempo em que cada robô atinge um mesmo ponto é utilizada. A situação em que os robôs possuem trechos em que podem se colidir também é abordada e, nesse caso, é realizada uma adaptação da formulação original.

#### 1.4 PUBLICAÇÕES DECORRENTES DA PESQUISA

Durante a realização desta pesquisa, foram obtidos alguns resultados que podem ser destacados:

- a) “Optimal Speed of Multiple robots based on Nonlinear Programming to Avoid Collision in Periodic Paths”. COSTA, L., SOUZA, M., OLIVEIRA, E., OLIVEIRA, L., *Proceedings of the 7th International Conference on Engineering Optimization, EngOpt2016*, Foz do Iguaçu, Brazil, June 2016.
- b) Participação no EngOpt2016 e apresentação do trabalho descrito em [a].
- c) “Collision avoidance for Multi-robot Systems with Coincident Paths based on Fictitious Collision Points using Nonlinear Formulation”. SOUZA, M., OLIVEIRA, E., OLIVEIRA, L., aceito para o *ROBOT’2017 - Third Iberian Robotics Conference*.

#### 1.5 ESTRUTURA DA DISSERTAÇÃO

Além deste, este trabalho possui quatro capítulos que são descritos a seguir:

O Capítulo 2 é destinado à revisão das técnicas empregadas na literatura para a coordenação de sistemas compostos por múltiplos robôs com caminhos especificados compartilhando o mesmo espaço de trabalho. É verificada a forma mais usual de coordenação, suas vantagens e desvantagens.

O Capítulo 3 apresenta a descrição do problema e as formulações propostas para resolução da coordenação de movimentos de sistemas robóticos. São demonstradas a função objetivos e restrições modeladas. O estudo do método é feito tanto para robôs móveis quanto para manipuladores, considerando suas peculiaridades. Além disso é realizado o desenvolvimento passo a passo do processo de solução.

O Capítulo 4 é destinado aos testes de validação da metodologia. Será dividido em duas seções envolvendo robôs móveis e manipuladores. Adicionalmente, apresenta comparações dos resultados obtidos com os oriundos de técnica já utilizada pela literatura.

O Capítulo 5 apresenta as principais conclusões e contribuições da utilização da metodologia, bem como propostas de trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 INTRODUÇÃO

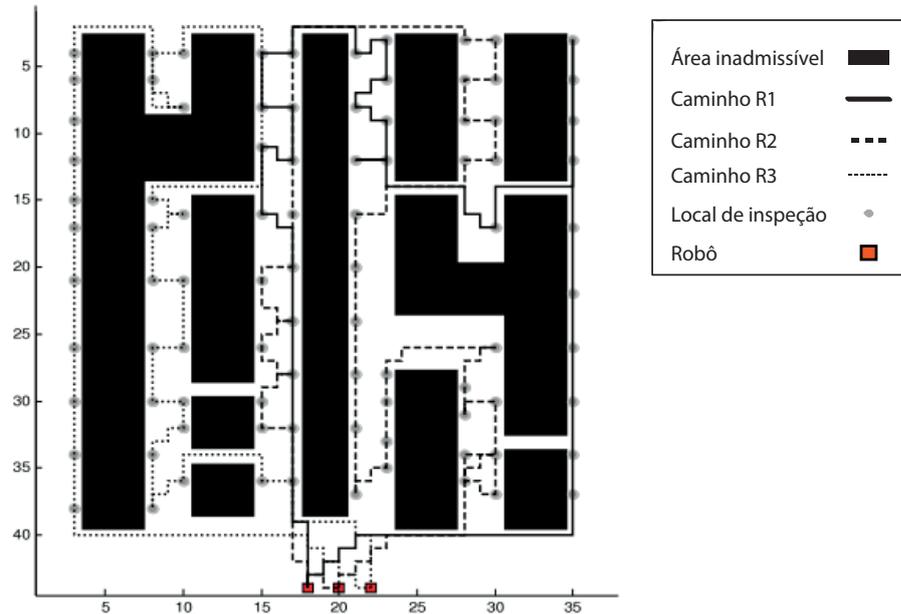
Os sistemas de múltiplos robôs, ou sistemas multi-robôs, (SMR), são uma importante área de pesquisa em robótica. Isso se deve às múltiplas aplicações em potencial para áreas como redes de sensores autônomos, vigilância de edifícios, transporte de objetos, monitoramento de poluição atmosférica e submarina, detecção de incêndios florestais, sistemas de transporte ou busca e resgate após grandes desastres [6]. Embora alguns problemas possam ser manipulados por um único robô, a utilização alternativa de uma equipe de robôs, em vez de apenas um, pode ser benéfica, uma vez que robustez e confiabilidade podem, muitas vezes, ser aumentadas pela combinação de vários robôs que, individualmente tem essas qualidades reduzidas. Focando no caso de robôs móveis, pode-se citar como exemplo a vigilância de uma grande área que, se adequadamente coordenada, a equipe de robôs é capaz de executar o trabalho mais rapidamente e com uma eficiência aumentada em comparação ao varrimento da área realizado por um só veículo. Além disso, coordenar o movimento dos robôs pode levar a uma redução da probabilidade de colisões entre os membros.

Normalmente, a solução mais adequada para este tipo de situação é desviar os robôs de suas trajetórias originais de forma a não causar acidentes. Com base em uma planta industrial, Jose e Pratihari [7] executam o planejamento de trajetórias sem colisões para três robôs (R1, R2 e R3) trabalhando em um espaço de trabalho comum, assegurando a segurança dos robôs. O objetivo dos robôs é inspecionar locais fixos predeterminados ao longo da planta. Apesar de terem locais específicos de monitoramento por parte dos robôs, eles tem a autonomia de alterar o planejamento a fim de executarem suas tarefas sem a interferência de outros. A trajetória ótima dos robôs é encontrada através dos métodos algoritmos genéticos e algoritmo A\*, considerando o planejamento offline. A Figura 1 ilustra os caminhos planejados para o sistema de três robôs. Outras obras também desenvolvem maneiras de planejar o caminho livre de colisões de robôs com a liberdade de desviá-los de seus caminhos previamente especificados [8], [9] e [10].

Focando no caso em que os caminhos são previamente especificados e possuem pontos de interseção entre eles, não há a possibilidade de desvios em seus caminhos a fim de obter êxito na redução de probabilidade de colisões. Dessa forma, há a necessidade de encontrar métodos que solucionem o problema sem alterar a trajetória original.

Esse problema com caminhos fechados e especificados pode ser observado em situações em que há a necessidade de realizar um caminho de forma frequente, como é o caso de veículos que realizam monitoramento de uma área [11]. Nesses casos, a solução se baseia em encontrar um perfil de velocidade ótimo que atenda às restrições específicas e que determine caminhos livres de colisão [12] e [13].

Figura 1: Atribuição de tarefas e caminhos seguidos por três robôs em uma planta industrial



Fonte: Adaptado de [7]

A maior parte da literatura especializada trata do problema de coordenação de robôs pelo ponto de vista de apenas um ponto em que os robôs podem se colidir, ou seja no ponto de interseção entre os caminhos. Porém devido às características físicas dos caminhos e dos robôs em algumas situações esse tipo de modelagem passa a ser inviável. Isso porque, em sistemas cujos caminhos especificados são muito próximos ou coincidentes, a passagem de mais de um veículo pode ficar impedida.

A partir desse cenário, ao longo desse capítulo, serão descritos alguns métodos desenvolvidos na literatura que são viáveis para resolução desses tipos de situações. A abordagem será focada tanto em robôs móveis quanto em manipuladores.

## 2.2 TÉCNICAS DE COORDENAÇÃO DE SMR

A coordenação dos movimentos de sistema de múltiplos robôs com caminhos predeterminados exige que não exista a possibilidade de desvios ao longo da trajetória. Assim, o intuito desse tipo de coordenação é criar um perfil de velocidade ótimo que atenda às restrições físicas dos robôs e evite possíveis acidentes. Nesse sentido, alguns trabalhos na literatura criaram técnicas capazes de sincronizar um grupo de robôs sujeito a essas condições. Carpin e Pagello [14] desenvolveram um algoritmo que seleciona alguns esquemas de prioridade aleatórios e, em seguida, agendam os movimentos dos robôs, inserindo atrasos ou até interrupções adequados quando colisões no espaço e no tempo são detectadas. Soltero et al. [15] recorreram a algoritmos com objetivo de evitar colisões

entre robôs de monitoramento persistente. Esses métodos se baseiam em políticas de paragem que quantificam o impacto dos tempos de interrupção na estabilidade global dos controladores de velocidade.

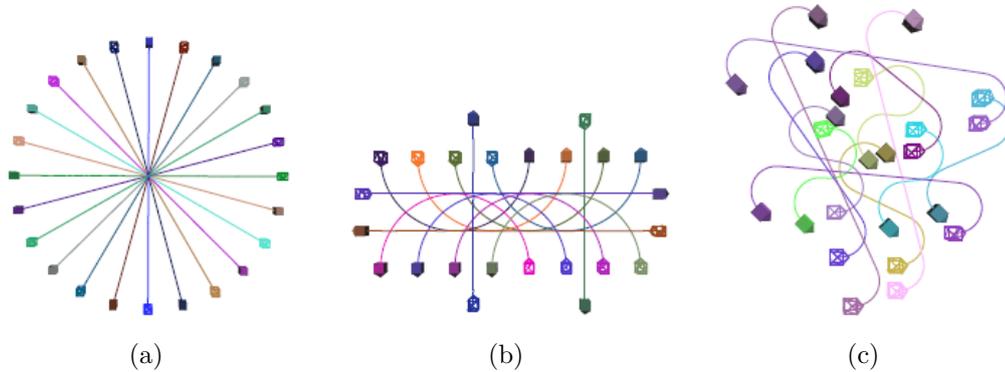
O atraso no início da movimentação dos robôs também pode ser uma abordagem que soluciona o problema de possíveis colisões ao longo da trajetória. Através da modelagem de dois braços robóticos como esferas, representando seus efetadores, trabalhando em um mesmo espaço de trabalho, Lee e Lee [16] determinaram uma trajetória através de uma programação de tempo, fazendo com que os robôs iniciem seus movimentos em diferentes momentos. Bien e Lee [17] também utilizam esse método, porém nesse último trabalho, consideram restrições de velocidade e acelerações lineares do manipulador em um espaço bidimensional.

No sentido de encontrar soluções ótimas que evitem possíveis colisões entre os robôs, formulações de otimização podem ser utilizadas. Considerando o cenário de células de trabalho de soldagem e pintura da indústria automobilística, Akella e Hutchinson [5] identificaram condições suficientes e necessárias para a coordenação livre de colisões dos robôs quando somente os tempos de início do movimento dos robôs podem ser variados e formularam o problema através de PLIM. O método é capaz de coordenar os movimentos de até vinte robôs e não há imposição no número de graus de liberdade dos manipuladores.

Outros trabalhos na literatura, abordam a coordenação de múltiplos robôs como um problema modelado por Programação Linear Inteira Mista [18]. Ao identificar pontos de colisão ao longo do caminho de um robô, Peng e Akella [19] formularam um problema de Programação Não Linear Inteira Mista (PNLIM). Uma vez que a formulação PNLIM não convexa resultante é de difícil resolução, os autores transformam em uma combinação de problemas modelados através de PLIM. Esse problema resultante é capaz de tratar restrições dinâmicas relacionadas à velocidade dos robôs. Para cada segmento de cada robô, os tempos de percurso mínimos e máximos possíveis que satisfazem as restrições são considerados. E assim, o objetivo é otimizar as velocidades dos robôs ao longo de segmentos do caminho e evitando colisões. As aplicações realizadas confirmam a viabilidade e efetividade do método, juntamente com resultados de implementação para 12 robôs móveis em três situações diferentes ilustradas pela Figura 2. No entanto, formulações de PLIM para a coordenação de múltiplos robôs possuem sua complexidade aumentada diretamente diante do número de zonas de colisão, podendo inviabilizar a aplicação do método diante de algumas situações [19].

A formulação PLIM desenvolvida nos trabalhos [5] e [19] é apresentada em forma resumida a seguir. Inicialmente, determina-se a trajetória de cada robô e assume-se que eles somente podem se colidir uns com os outros em uma região e apenas uma vez nesse determinado local. Para cada robô, há a identificação de sua zona de colisão e cálculo do intervalo de tempo durante o qual está nessa zona. Além disso, esses trabalhos revelam

Figura 2: Sistema composto por doze robôs e suas respectivas configurações de caminhos: (a) Radiais com simetria, (b) Linha reta e circulares, (c) Aleatória



Fonte: [19]

uma dificuldade de modelar restrições dinâmicas do robô, tendo que recorrer à formulações de PNLIM ou combinações de PLIM, dificultando o processo de resolução.

Para um caso geral considerado por [5], foram coordenados múltiplos robôs com múltiplas regiões de colisão. Assumindo que  $T_i$  é o tempo do movimento do robô  $i$  para percorrer toda a sua trajetória que se inicia no instante  $T_{start}$ , os robôs apresentam zonas de colisão que são modeladas através de pares de intervalos  $[T_{i0}^j, T_{if}^j]$ ,  $[T_{k0}^j, T_{kf}^j]$ . Esse parâmetro representa o par de intervalo de tempo para dois robôs  $i$  e  $k$ , sendo que os índices 0 e  $f$  indicam os tempos iniciais e finais, respectivamente que um par de robôs  $i$  e  $j$  pode colidir no ponto de colisão  $j$ . A parte inteira do problema é incluída através da variável binária  $\delta_{ikn}$  que é definida como sendo 0 se o robô  $i$  entra na zona de colisão  $j$  com o robô  $k$  antes do robô  $k$  e 1 se ele entra na sua zona de colisão com o robô  $k$  depois do robô  $k$ .

Nesse trabalho a forma utilizada para evitar as colisões é atrasar o início do movimento dos robôs, sem alterar a velocidade dos robôs ao longo do tempo. O objetivo é minimizar o tempo de conclusão  $t_c$ , ou seja, encontrar o menor valor de atraso, ao mesmo tempo em que é assegurado que os robôs não estejam em suas zonas de colisão ao mesmo tempo. O problema de PLIM resultante para a coordenação dos robôs demonstrado pela formulação composta pela função objetivo 2.1 e as restrições 2.2 a 2.5.

$$\text{Min } t_c \quad (2.1)$$

sujeito a:

$$t_c - t_i^{start} - T_i \geq 0, \quad 1 \leq i \leq n_r \quad (2.2)$$

$$t_i^{start} + T_{if}^k - t_j^{start} - T_{is}^n - M\delta_{ikj} \leq 0, \quad (2.3)$$

$$[T_{is}^n, T_{if}^n], [T_{ks}^n, T_{kf}^n] \in N_{ik}, \quad 1 \leq i < j \leq n_r$$

$$t_j^{start} + T_{jf}^k - t_i^{start} - T_{i0}^n - M(1 - \delta_{ikj}) \leq 0, \quad (2.4)$$

$$\delta_{ikj} \in \{0, 1\}, \quad 1 \leq i < k \leq n_r, \quad 1 \leq n \leq N_{ik}$$

$$t_i^{start} \geq 0 \quad (2.5)$$

onde:

$t_c$	Tempo de conclusão das trajetórias;
$t_i^{start}$	Tempo de início da trajetória relacionado ao robô $i$ ;
$T_i$	Tempo do movimento do robô $i$ ;
$n_r$	Número de robôs;
$T_{i0}^n$	Tempo relacionado à $n$ -ésima vez que o robô $i$ entrará na zona de colisão com outro robô;
$T_{if}^n$	Tempo relacionado à $n$ -ésima vez que o robô $i$ sairá na zona de colisão com outro robô;
$N_{ik}$	Número de pares de intervalos de tempo de colisão para os robôs $i$ e $k$ ;
$M$	Valor elevado positivo;
$\delta_{ikj}$	Variável binária relacionada à entrada dos robôs $i$ e $k$ na zona de colisão $j$ .

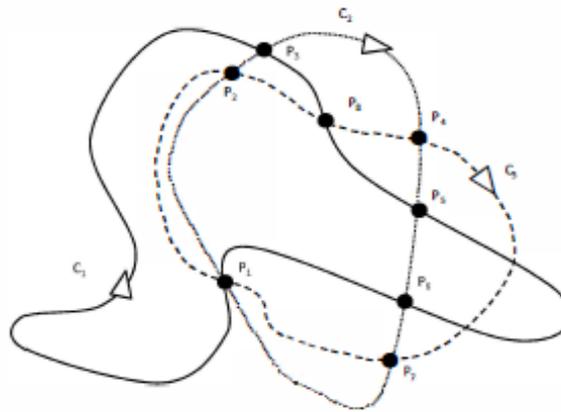
Quando o robô  $i$  entra na zona de colisão primeiro,  $\delta_{ikj} = 0$  e a restrição 2.3 está ativa e quando o robô  $k$  entra na zona de colisão primeiro, a restrição 2.4 está ativa. A solução resultante garante uma estratégia de coordenação de trajetória livre de colisão para todos os robôs. As restrições de tempo de conclusão e as restrições de intervalo de tempo de colisão são necessárias apenas para os robôs que podem colidir.

Nessa formulação a redução da probabilidade de colisão é incluída como uma restrição, no caso, as representadas pelas inequações 2.3 e 2.4. Elas asseguram que a

diferença de tempo em que os robôs atingirão o ponto de colisão seja elevada de tal forma que eles não se encontrem nesse local.

Um outro tipo de abordagem, mas também utilizando PLIM, é desenvolvida por Gonçalves et al. [4]. Diferentemente dos anteriores, nesse trabalho, os autores tratam a redução da possibilidade de colisão como função objetivo e minimizam indiretamente a probabilidade desse evento. O problema consiste em um conjunto de veículos aéreos não tripulados, do inglês *Unmanned Aerial Vehicles* (UAV), que devem percorrer caminhos cíclicos e especificados com pontos de interseção, conforme ilustrado pela Figura 3. Sendo que os pontos pretos identificam os pontos onde podem ocorrer colisões entre dois robôs.

Figura 3: Sistema composto por três robôs (triângulos) que devem percorrer caminhos cíclicos com interseções (pontos pretos).



Fonte: [4]

O objetivo é determinar um perfil de velocidade que garanta, com uma certa margem de segurança, que os robôs não estejam num determinado ponto de colisão no mesmo tempo. Caso a passagem dos robôs por um mesmo ponto seja em momentos diferentes e o mais distante possível, pode-se garantir que não haverá uma colisão pontual. Assim, determinar um intervalo de tempo mínimo  $\Delta t$ , em que dois robôs passem por um mesmo ponto é uma boa solução. Nesse caso,  $\Delta t$  é a solução do problema de otimização que tem como função objetivo a maximização desse parâmetro. Dessa forma, é maximizada a margem de segurança que assegura que a passagem dos robôs por um ponto de interseção seja realizada em momentos diferentes e o mais espaçados possíveis, de forma temporal. A formulação completa é descrita pela função objetivo 2.6 e restrições dadas pelas desigualdades 2.7 a 2.13 [4].

$$\text{Max } \Delta t \tag{2.6}$$

sujeito a:

Restrições relativas à margem de segurança:

$$t_j[i, q] - t_j[k, q] \geq \Delta t - \xi \delta_{ikj} \quad (2.7)$$

$$t_j[i, q] - t_j[k, q] \leq -\Delta t + \xi(1 - \delta_{ikj}), \quad (2.8)$$

$$0 \leq q \leq n_c, \forall j \in \mathcal{Q}, i \neq k, \forall i, k \in \mathcal{D}_j, \forall \delta_{ikj} \in \{0, 1\}$$

Restrição relativa à periodicidade e à comensurabilidade dos ciclos:

$$T_i = \gamma_i T, \forall i \in \mathcal{D}_j \quad (2.9)$$

Restrição de velocidade:

$$\tau_{min}[i, j] \leq t_j[i, q] \leq \tau_{max}[i, j], \forall i \in \mathcal{R}, \forall j \in \mathcal{J} \quad (2.10)$$

Condições iniciais:

$$-T + \Delta t \leq t_j[i, 0] - t_j[k, 0] \leq T - \Delta t, \quad (2.11)$$

$$\forall j \in \mathcal{Q}, i \neq k, \forall i, k \in \mathcal{D}_j$$

$$t_{l(1)}[1, 0] = 0 \quad (2.12)$$

Condição de não negatividade de  $\Delta t$ :

$$\Delta t \geq 0 \quad (2.13)$$

Sendo que os seguintes parâmetros representam:

$\Delta t$	Menor diferença de tempo entre a passagem de robôs em um mesmo ponto de colisão;
$\mathcal{R}$	Conjunto de robôs indexado por $i$ ;
$\mathcal{Q}$	Conjunto de pontos de colisão indexado por $j$ ;
$\mathcal{D}_j$	Conjunto de robôs que podem colidir no ponto $j$ ;

$\mathcal{J}_i$	Conjunto de pontos de colisão presentes no caminho do robô $i$ ;
$\xi$	Valor elevado relacionado a margem de segurança;
$n_c$	Número de ciclos;
$\delta_{ikj}$	Valor binário que indica a ordem de passagem dos robôs $i$ e $k$ no ponto $j$ ;
$t_j[i, q]$	Tempo em que o robô $i$ alcança o ponto de colisão $j$ no $q$ -ésimo ciclo;
$T_i$	Tempo que o robô $i$ completa sua trajetória;
$T$	Tempo base do ciclo;
$\gamma$	Valor relativo a comensurabilidade de ciclos para cada robô $i$ ;
$\tau_{min}[i, j]$	Tempo mínimo do robô $i$ no trecho $j$ ;
$\tau_{max}[i, j]$	Tempo máximo do robô $i$ no trecho $j$ ;
$t_{l(1)}[1, 0]$	Tempo de saída do robô 1.

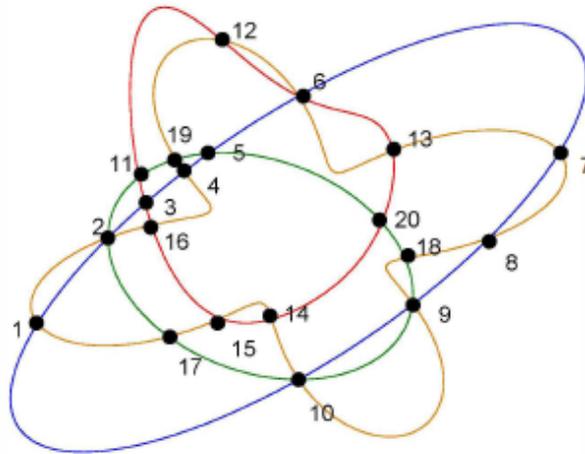
Essa formulação considera como variáveis inteiras às referentes ao número de ciclos,  $q$  e a variável binária  $\delta_{ikj}$  que representa a ordem de passagem dos robôs por um ponto de colisão. Ou seja, considerando os robôs R1 e R2, caso o robô R1 passe pelo ponto antes do robô R2 no ponto de colisão 1,  $\delta_{121} = 0$ , caso contrário  $\delta_{121} = 1$ . Dessa forma quando  $\delta_{ikj} = 0$ , a desigualdade 2.7 se torna  $t_j[i, 0] - t_j[k, 0] \geq \Delta t$  e a desigualdade 2.8 fica sem efeito, devido ao alto valor de  $\xi$ . E quando  $\delta_{ikj} = 1$  a desigualdade 2.7 perde o efeito e a desigualdade 2.8 se torna  $t_j[i, 0] - t_j[k, 0] \geq \Delta t$ .

Nessa aplicação os ciclos devem ser periódicos, ou seja todos eles devem ser relacionados com relação ao tempo. A restrição 2.9 garante essa comensurabilidade dos ciclos. A restrição 2.10 garante que os robôs não ultrapassem os limites máximo e mínimo determinados.

O exemplo utilizado para demonstrar a eficácia do método apresenta quatro robôs que devem seguir curvas fechadas com 20 interseções conforme ilustrado pela Figura 4. O sistema deve obedecer limitações máxima e mínima de velocidade e não podem desviar de seus caminhos predeterminados. O resultado garante a movimentação dos veículos livre de colisões com uma certa margem de segurança.

No caso de múltiplos manipuladores robóticos, o problema pode ser modelado dessa mesma forma. Spensieri et al. [20] trataram a situação de evitar colisões na coordenação braços robóticos da mesma maneira que o caso anterior para robôs móveis, porém adaptada às peculiaridades do sistema. Em um sistema composto por quatro robôs industriais que podem se colidir em pontos especificados, o objetivo é encontrar um ciclo mínimo que

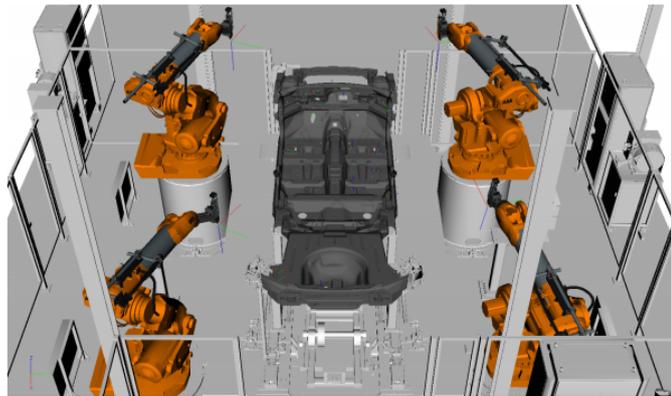
Figura 4: Caminhos a serem percorridos por quatro robôs com vinte pontos de colisão.



Fonte: [4]

evite possíveis colisões. A Figura 5 ilustra o espaço de trabalho dos robôs. O método se refere à possibilidade de colisão através de uma restrição, considerando a posição de entrada na zona de colisão. Dessa forma a formulação também de se baseia em PLIM.

Figura 5: Layout industrial do caso de teste de coordenação de manipuladores.



Fonte: [20]

O problema consiste em que, modelado da forma de PLIM, o problema passa a ser NP-completo. Um problema NP, é aquele que é solucionado em tempo polinomial não determinístico, do inglês *Nondeterministic Polynomial Time*. Todo problema NP-completo pode ser resolvido por busca exaustiva. Entretanto, quando o tamanho das instâncias cresce, o tempo de execução para busca exaustiva se torna proibidamente grande, mesmo para instâncias de tamanho razoavelmente pequeno [21]. Embora ainda não resolva em tempo polinomial, para alguns problemas é possível projetar algoritmos que são significativamente mais rápidos do que pesquisa exaustiva. Esse fato permite a modelagem do problema de

sistema de múltiplos robôs com caminho especificados através de PLIM. Porém quando a determinação do perfil de velocidade dos robôs acontece em um sistema real, limitações não lineares podem ser fatores que impedem a aplicação desse método. Caso sejam impostas restrições que limitam acelerações e torque, ou ainda restrições relacionadas a custo de operação que sejam não lineares, não é possível a resolução através de PLIM.

Problemas de coordenação de movimentos também podem ser modelados através de Programação Não-Linear. Abichandani et al. [22] tratou a situação dessa forma. Considerando caminhos acíclicos que possuem pontos de interseção, a otimização aplicada tem por objetivo minimizar o tempo que o último robô chegará ao seu objetivo. As colisões são evitadas através da inclusão de uma restrição que determina que a distância entre os robôs nunca ultrapassem um limite de segurança. Nesse caso não há especificação de tempo e os caminhos não são periódicos.

### 2.3 COORDENAÇÃO EM SISTEMAS COM TRECHOS COMUNS

Em todos os trabalhos anteriormente mencionados, cujo objetivo é a coordenação de robôs, a possibilidade de colisão é tratada como um ponto comum entre as trajetórias em que os robôs podem se encontrar. Em sistemas cujos caminhos são elevadamente extensos com relação ao tamanho físico dos robôs e os pontos de interseção são formados por caminhos aproximadamente perpendiculares, esse tipo de abordagem não gera grandes problemas. A literatura específica, atrelada a coordenação de robôs, não oferece, porém, formas de resolver problemas que abordam trechos de colisões. Ou seja, um segmento do caminho de um robô é coincidente com o segmento do caminho de outro. Em algumas situações, podem existir momentos em que os veículos tem o mesmo caminho a percorrer. Assim, os métodos de solução abordados não podem ser aplicados.

Devido ao tamanho físico dos robôs ou mesmo às peculiaridades dos caminhos, pode haver mais de um ponto onde haverá possibilidades de colisão. Assim, esta situação determina um novo conceito de configuração do sistema. Nesses sistemas, os caminhos se cruzam na forma de trechos chamados segmentos de colisão. A classe desses problemas não pode ser resolvida da mesma forma dos que abordam apenas pontos de colisão, pois os acidentes podem acontecer ao longo do trecho comum dos caminhos. Esta situação é observada em sistemas cujos caminhos especificados são muito próximos ou coincidentes, o que impede a passagem de mais de um veículo. É comum discutir esse tema em obras de planejamento ferroviário. Neste caso, os trens possuem riscos de colisão em determinados trechos da ferrovia. Assim como na hipótese do presente trabalho, é preciso planejar a trajetória dos trens a fim de evitar acidentes ao longo da ferrovia. Yan e Yang [23] propuseram um método baseado em Programação Linear Inteira Mista para resolver o Problema do Planejador de Movimento, considerando que, nesse caso, deve-se planejar as velocidades para evitar colisões em segmentos comuns da trajetórias de robôs que

apenas podem se colidir frontalmente. Dessa forma, os trens possuem a possibilidade de seguir por um caminho alternativo já especificado, no momento que outro trem passa pelo caminho original. De uma forma geral, a função objetivo, utilizada nesse trabalho, minimiza os custos totais incorridos pelo atraso do trem. Além de outras peculiaridades do sistema ferroviário, simplificada, com o objetivo de evitar colisões, eles incluíram restrições na formulação com o objetivo de proibir a permanência de mais de um trem na mesma trilha. Essas restrições, demonstradas pelas desigualdades 2.14 a 2.16, foram incluídas de forma parecida com o trabalho de Gonçalves et al. [4], por meio da margem de segurança, chamado no planejamento ferroviário de frequência dos trens, ou intervalo mínimo. Entretanto, no caso de trens devem ser considerados os comprimentos dos comboios.

$$x_{i',j}^{entrada} - x_{i,j}^{saida} \geq \Delta P - M(1 - \beta_{i,i',j}) \quad (2.14)$$

$$x_{i,j}^{entrada} - x_{i',j}^{saida} \geq \Delta P - M(1 - \lambda_{i,i',j}) \quad (2.15)$$

$$\lambda_{i,i',j} + \gamma_{i,i',j} \leq 1 \quad (2.16)$$

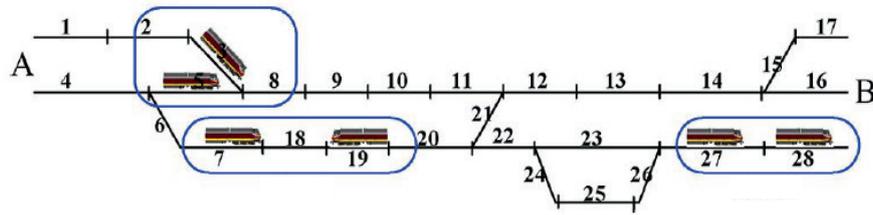
$$\forall i, i' \in \mathcal{W}_j, \forall j \in \mathcal{G}, i \neq i'$$

Sendo que os seguintes parâmetros representam:

$\mathcal{W}_j$	Conjunto de trens que ocupam o segmento $j$ indexados por $i$ ;
$\mathcal{G}$	Conjunto de segmentos indexados por $j$ ;
$x_{i,j}^{entrada}, x_{i,j}^{saida}$	Tempo de entrada e saída, respectivamente do trem $i$ no segmento $j$ ;
$\Delta P$	Intervalo mínimo dos trens;
$M$	Valor suficientemente alto;
$\beta_{i,i',j}$	Variável igual a 1 se o comboio $i$ está agendado antes do comboio $i'$ no segmento $j$ e igual a 0 caso contrário;
$\lambda_{i,i',j}$	Variável igual a 1 se o comboio $i$ está agendado depois do comboio $i'$ no segmento $j$ e igual a 0 caso contrário.

Zhao e Ioannou [24] também abordaram o problema de controle ferroviário. O objetivo é planejar o movimentos dos trens, partidas e chegadas de forma a evitar acidentes

Figura 6: Casos de conflito de trens num sistema ferroviário.



Fonte: Adaptado de [24]

ao longo da estrada de ferro. A Figura 6 ilustra casos que em que podem ocorrer colisões entre os trens compartilhando a mesma via, podendo ser em direções opostas ou não.

Eles trataram o problema que envolve diminuição da probabilidade de colisão no planejamento ferroviário de forma análoga ao trabalho anterior [23] e também utilizando PLIM. O objetivo, nesse caso, é minimizar o tempo de viagem total exigido para todos os trens expedidos para se deslocar da sua origem até as estações de destino. Isso é equivalente a minimizar a soma dos tempos de chegada ponderados de todos os comboios no seu segmento de estação de destino.

Apesar de mudar o foco do trabalho, ainda conforme as outras abordagens mencionadas relacionadas a pontos de colisão, o método aplicado a segmentos de colisão no planejamento ferroviário apresenta como solução um problema formulado através de PLIM, que é difícil de resolver por sua natureza combinatória e não comporta variáveis não lineares.

Há outros cenários onde podem existir caminhos coincidentes e movimentos que devem ser planejados a fim de evitar acidentes. É o caso dos armazéns logísticos. No local, os robôs devem estabelecer interações colaborativas para atingir seus objetivos individuais e coletivos, a fim de otimizar a função objetivo de todo o sistema. Em um ambiente onde os robôs devem transportar itens do carregamento para o descarregamento de baias, de modo a concluir os pacotes a serem entregues aos clientes, Farinelli et al. [25] propuseram uma solução baseada em técnicas de otimização. Os robôs devem cooperar para maximizar o número de pacotes concluídos em um determinado tempo. Para esse fim um fator crucial é evitar interferências ao se mover no ambiente.

A situação pode ser descrita como itens de vários tipos que são armazenados em determinadas seções, as baias de carregamento, e devem ser transportados para um conjunto de baias de descarregamento onde esses itens são embalados em conjunto por operadores humanos, conforme ilustrado pela Figura 7. Os robôs envolvidos no processo de transporte se movimentam no armazém e são susceptíveis à interferências quando se



objetivo, relacionada a redução da probabilidade de colisões, é apresentada como:

$$\text{Min} \sum_q I_q(a_q) \quad (2.17)$$

Essa formulação voltada somente para colisões é irrestrita, havendo restrições apenas para peculiaridades do sistema logístico. Os resultados aplicados em esquemas envolvendo 5 robôs, 20 baias de carregamento e 5 de descarregamento demonstraram a efetividade do método. Para esquemas com mais robôs e baias, este tipo de abordagem pode ser oneroso em termos da dificuldade de modelar essas restrições na formulação, aumentando a complexidade do sistema.

Assim como no planejamento ferroviário e logístico, na coordenação de múltiplos robôs com segmentos comuns, é desejável que os robôs completem suas tarefas, ou seja, alcancem seus destinos finais, sem probabilidade de colisões ao longo da trajetória. Considerando que não existe a possibilidade de se desviar do trecho especificado e que os robôs não devem estar no trecho comum ao mesmo tempo, o objetivo é planejar o perfil de velocidade do robô.

## 2.4 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais métodos utilizados pela literatura para resolver o problema de coordenação de múltiplos robôs. O objetivo desse problema é encontrar a configuração ótima de robôs de forma a evitar colisões entre eles em pontos comuns dos caminhos. Algumas metodologias permitem que o robô desvie de seu caminho, para evitar interferências, o que não é possível quando eles possuem caminhos previamente especificados. Dessa forma, apenas trabalhos que envolvendo coordenação de robôs com caminhos especificados foram destacados. Abordando dessa maneira, o intuito é determinar um perfil de velocidade que evite possíveis colisões nos pontos de interseção entre os caminhos. Nessa linha, a doutrina majoritária converge em soluções que modelam o problema através de Programação Linear Inteira Mista.

A representação de limitações dinâmicas do sistema fica dificultada em problemas que utilizam PLIM por se tratarem de restrições não lineares. Dessa forma, para evitar a manipulação de variáveis não lineares e inteiras, formando um problema de PNLIM, uma combinação de modelos envolvendo variáveis binárias deve ser realizada a fim de encontrar a configuração ótima dos robôs.

Como em alguns casos o ponto de colisão não é a melhor forma de representar o local onde pode acontecer a colisão, foram discutidas formas de evitar interferências entre veículos em uma situação em que os robôs compartilham o mesmo espaço de trabalho e têm seus caminhos especificados, sendo que esses possuem trechos comuns, determinando

um segmento em que os robôs podem colidir. Nesse caso, também foram encontradas abordagens utilizando PLIM.

Apesar de resolver diversas situações de forma satisfatória e encontrar um perfil de velocidade que evite colisões, encontrar soluções através de PLIM pode ser inviável. Isso é explicado pelas não linearidades da situação real. Esse tipo de formulação não permite a inclusão de variáveis não lineares, impossibilitando, por exemplo, a inclusão de restrições de aceleração ao longo da trajetória. Tais restrições limitam a variação de velocidade ao longo trecho e criam perfis de velocidade suaves. Caso não seja possível a inclusão dessa limitação, podem ser encontradas velocidades com discrepâncias, podendo inviabilizar a aplicação prática.

### 3 METODOLOGIA PROPOSTA

#### 3.1 INTRODUÇÃO

Sistemas multi-robôs podem ser descritos como um conjunto de múltiplos robôs, executando tarefas de forma coordenada, em busca de um objetivo comum. Considerando robôs que compartilham um mesmo espaço de trabalho, coordenar os movimentos deles significa buscar a configuração que reduza a possibilidade de colisão entre eles. Não existindo a presença de obstáculos fixos, deve-se planejar a velocidade dos robôs para que, num determinado ponto comum de seus caminhos, não haja a possibilidade de acidentes.

Quando os caminhos são previamente especificados, ou seja, eles devem seguir exatamente por pontos predeterminados, não existe a possibilidade de alterar a sua rota, tendo que encontrar outra forma de evitar a colisão com outros robôs. Assim, a maneira mais adequada é determinar um perfil de velocidade que respeite os limites físicos dos robôs e limites temporais da trajetória.

Através da revisão da literatura especializada no assunto, percebe-se que o método mais utilizado para encontrar o perfil de velocidade ótimo é através de Programação Linear Inteira Mista. De maneira geral, essas formulações consideram elementos como limites de velocidade, tempo de execução das tarefas e chegada ao ponto de colisão. Esse último elemento que inclui a variável inteira binária no problema. Alguns trabalhos também contabilizam números de ciclos, o que também dá natureza inteira à formulação.

Nesse capítulo será abordada e estudada uma forma de tratar esse problema sem a inclusão de variáveis inteiras na formulação, fato que dificulta a resolução. O método proposto utiliza Programação Não-Linear e será aplicado ao caso de múltiplos robôs móveis e adaptado ao caso de manipuladores robóticos.

#### 3.2 COORDENAÇÃO DE MÚLTIPLOS ROBÔS MÓVEIS

##### 3.2.1 Descrição do Problema

Dado um conjunto de robôs móveis  $\mathcal{R}$  que devem sair de um ponto inicial, contido no conjunto de pontos iniciais  $\mathcal{P}$  e passar por um conjunto de caminhos cíclicos e especificados  $\mathcal{C}$  localizados num mesmo espaço de trabalho a serem percorridos por cada robô  $i \in \mathcal{R}$ , o objetivo é selecionar as configurações de velocidades ideais que permitam a redução da possibilidade de colisões. Esse risco existe no caso em que os caminhos se cruzam de maneira pontual, determinando no local um ponto de interseção entre os caminhos, chamado de ponto de colisão. O conjunto de pontos de colisão  $j$  que o robô  $i$  terá em seu caminho é determinado por  $\mathcal{J}_i$ . E o conjunto de robôs que passarão por um mesmo ponto de colisão  $j$  é representado por  $D_j$ . Assim, a otimização da trajetória terá que considerar os pontos críticos onde os robôs podem se colidir e fazer com que eles os cruzem

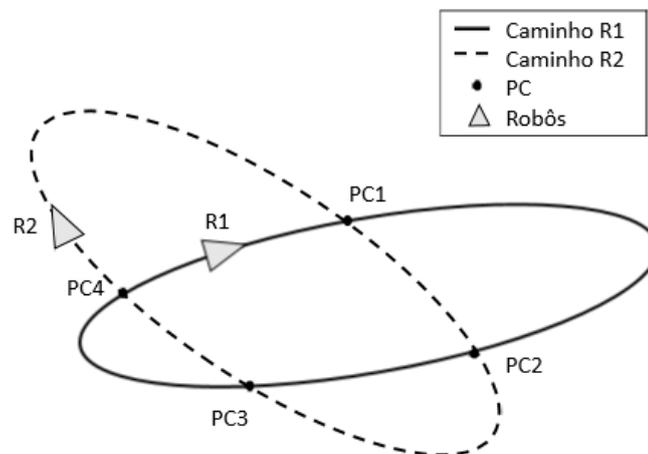
em momentos diferentes.

Nesse problema, além de terem caminhos especificados, os tempos de conclusão de cada trajetória percorrida pelos robôs para sair de um ponto inicial e retornar ao mesmo devem ser previamente determinados e periódicos entre si. Isso não significa que os robôs devem terminar suas tarefas no mesmo momento, mas sim que, para garantir a coordenação, os tempos de término dos ciclos devem ser comensuráveis. Ou seja, para todo  $i$  deve existir um número natural  $\gamma_i$  e um real  $T$ , que atenda a:

$$T_i = \gamma_i T \quad (3.1)$$

Para facilitar o entendimento do problema, é ilustrado na Figura 8 um sistema composto por dois robôs,  $R_1$  e  $R_2$ , que possuem caminhos que se interceptam em quatro pontos,  $CP_1$  a  $CP_4$ . Entre dois pontos de colisão, os robôs percorrem rotas distantes em caminhos diferentes, garantindo uma passagem livre por este trecho. É claro que os robôs somente podem se colidir em pontos de interseção, sem a possibilidade de se encontrarem em outros pontos ao longo do caminho.

Figura 8: Sistema composto por dois robôs e quatro pontos de colisão.



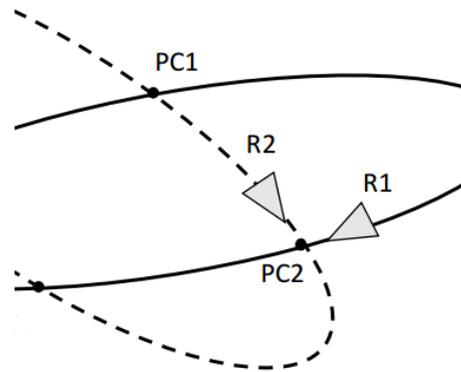
Fonte: Elaborada pelo autor

Alguns termos devem ser conceituados. Caminho pode ser entendido como a configuração geométrica de uma curva no espaço de trabalho que o robô deverá seguir, sendo que cada robô possui apenas um caminho previamente especificado. Trecho representa partes do caminho, que podem ser caracterizadas pela distância entre pontos de colisão e entre pontos de colisão e pontos de partida. O conjunto de trechos que o robô  $i$  possui ao longo de seu caminho é representado por  $\mathcal{S}_i$ . Entende-se como trajetória, o caminho com características dinâmicas especificadas ao longo do tempo, isto é, a trajetória de um robô é composta pelo caminho adicionado do perfil de velocidade do robô. As velocidades são

obtidas pelo tempo e pelo comprimento de cada trecho. Assim, o objetivo é, na realidade, a determinação da trajetória de cada robô.

Na Figura 8 os caminhos que cada robô deve seguir podem ser observados pelas duas elipses, que têm, neste caso, quatro pontos de interseção. A colisão pode ocorrer caso os robôs atravessem um mesmo ponto em momentos muito próximos. A Figura 9 apresenta uma aproximação do sistema mencionado e ilustra uma situação em que os robôs, não coordenados de forma correta, estão em uma situação de alto risco de colisão no ponto  $PC_2$ .

Figura 9: Sistema composto por dois robôs com alto risco de colisão.



Fonte: Elaborada pelo autor

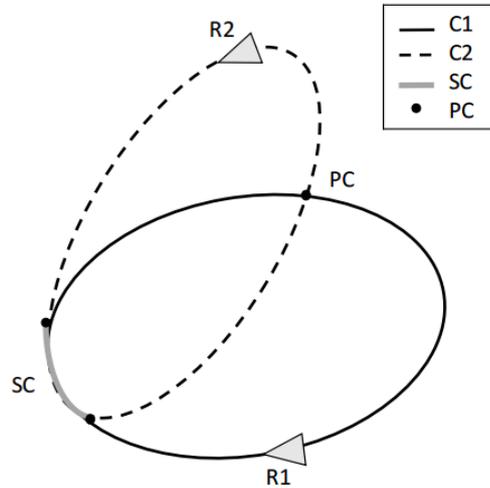
Uma forma de se evitar colisões entre robôs do sistema conforme ilustrado pela figura é parar um dos robôs antes que ele encontre com o outro no ponto de colisão. Porém essa pode não ser uma boa ideia, pois frenagens bruscas podem não ser tão eficientes, não evitando acidentes. Assim, de forma preventiva, a melhor solução é determinar um perfil de velocidade adequado para cada robô que resulte em movimentos sincronizados que garantam caminhos livres de colisões.

### 3.2.2 Representação dos Trechos Coincidentes

Na situação descrita, entre dois pontos de colisão, os robôs percorrem caminhos diferentes e distantes, garantindo uma passagem livre por este trecho. Dessa forma, os robôs só podem colidir em pontos de interseção, sem a possibilidade de se encontrarem em outros pontos ao longo do caminho. Devido ao tamanho físico dos robôs ou mesmo às peculiaridades dos caminhos, pode haver mais de um ponto onde haverá possibilidades de colisão. Assim, essa situação determina um novo conceito de configuração do sistema. Nesse sistema, os caminhos também podem se cruzar na forma de trechos que serão chamados segmentos de colisão. A Figura 10 ilustra a configuração de dois robôs e seus

respectivos caminhos que se cruzam em um ponto de colisão ( $PC$ ) e também se coincidem na forma de um segmento de colisão ( $SC$ ).

Figura 10: Sistema composto por dois robôs com caminhos coincidentes.



Fonte: Elaborada pelo autor

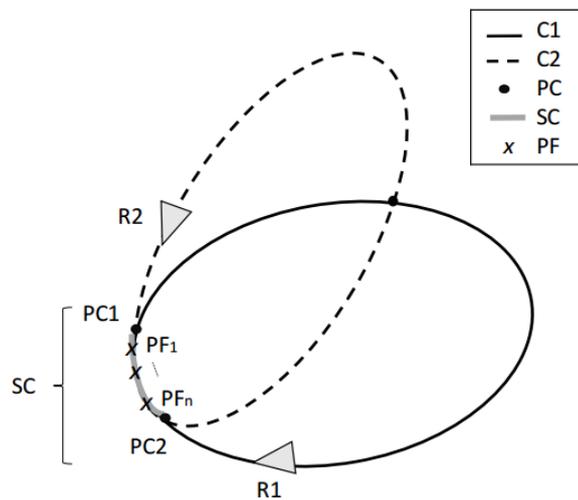
Não se pode admitir que mais de um robô esteja presente no mesmo segmento de colisão, pois há o risco de uma colisão frontal entre os veículos, supondo que eles estão em direções opostas, conforme na Figura 10 ou colisão devido a velocidades diferentes caso eles estiverem na mesma direção.

Com a finalidade de modelar esses trechos, para os segmentos de colisão, as extremidades podem ser consideradas como pontos de colisão. Assim, cada segmento de colisão é modelado como dois pontos de colisão através de seus pontos de extremidade. A introdução desses pontos pode fazer com que os robôs não se colidam nas extremidades do trecho, mas não os impede de colidir ao longo do segmento.

Para resolver esta situação, pode-se incluir um novo ponto de colisão fictício (PF) entre as extremidades. Assim, além de aumentar a margem de segurança na entrada do trecho, garante que os robôs não se colidam nesse local. A eficiência desse novo ponto de colisão depende da extensão do segmento. Se o comprimento do trecho comum for muito longo, pode ser que apenas um ponto entre as extremidades não seja suficiente para impedir que os robôs estejam presentes no mesmo segmento. Esse número de pontos é determinado pela situação de não-colisão ao longo do segmento, ou seja, após encontrar o perfil de velocidade final, caso identifique que haverá colisão, inclui-se um novo ponto fictício e repete-se esse processo até que seja encontrada uma configuração livre de colisões. Dessa forma, a formulação final deverá considerar para efeito de otimização os pontos de colisão já identificados pelo sistema e os pontos fictícios suficientes para garantir um caminho seguro.

A Figura 11 ilustra o sistema focalizando no trecho comum entre os caminhos de  $R_1$  e  $R_2$ . Os pontos  $PF_1, \dots, PF_n$  simbolizam os pontos fictícios inseridos para representar os segmentos de colisão, compreendidos entre os pontos  $PC_1$  e  $PC_2$  (extremidades do trecho), sendo que  $n$  é o número de pontos fictícios incluídos.

Figura 11: Inclusão de pontos fictícios para representar o segmento de colisão.



Fonte: Elaborada pelo autor

Um segmento de colisão, portanto, é modelado pelos pontos de colisão localizados nas extremidades adicionado de pontos fictícios ao longo do trecho suficientes para evitar colisões. E para fins de formulação, todos esses pontos são tratados da mesma forma. Tratando o segmento de colisão dessa maneira, não importa se os robôs estão em direções opostas ou não, pois os pontos fictícios criarão barreiras que impedirã a permanência de mais de um robô no trecho. Além disso, consegue-se retirar a necessidade de variáveis binárias relacionadas à entrada e saída do segmento.

### 3.2.3 Modelagem do Problema de Otimização

Indo de encontro às soluções já apresentadas na literatura e discutidas anteriormente nesse trabalho, utilizando PLIM, a modelagem proposta utiliza a metodologia de Programação Não-Linear, com variáveis de decisão não lineares.

Os conjuntos considerados para formulação são descritos a seguir:

$\mathcal{R} = \{R_1, R_2, \dots, R_i\}$  Representa o conjunto de robôs, indexado por  $i$ .

$\mathcal{C} = \{C_1, C_2, \dots, C_i\}$  Representa o conjunto de caminhos a serem percorridos pelos robôs, indexado por  $i$ .

$\mathcal{P}_o = \{P_{o_1}, P_{o_2}, \dots, P_{o_i}\}$  Representa o conjunto de pontos iniciais dos caminhos dos robôs, indexado por  $i$ .

$\mathcal{J}_i = \{PC_1, PC_2, \dots, PC_j\}$  Representa o conjunto de pontos de colisão presentes no caminho do robô  $i$ , indexado por  $j$ .

$\mathcal{S}_i = \{S_1, S_2, \dots, S_s\}$  Representa o conjunto de trechos presentes no caminho do robô  $i$ , indexado por  $s$ .

$\mathcal{D}_j \in \mathcal{R}$  Representa o conjunto de robôs que passarão por um mesmo ponto de colisão  $j$ .

A fim de chegar à formulação final do problema de coordenação, além da descrição do sistema, algumas considerações adicionais devem ser explicitadas:

- Os obstáculos são exclusivamente móveis, ou seja, os veículos que se deslocam no espaço de trabalho só podem colidir entre si. Portanto, o caminho especificado para cada robô não fornece colisões com obstáculos fixos, que não são abordados pelo presente trabalho;
- A velocidade inicial do robô é nula, ou seja, todos os robôs, antes de iniciar suas trajetórias, estão parados em relação ao sistema;
- As velocidades têm valores mínimos e máximos para cada robô  $i$ ,  $Vmin_i$  e  $Vmax_i$ , respectivamente, aos quais o cálculo das trajetórias deve respeitar. Dessa forma, considerando os trechos dos caminhos, existirão tempos mínimos e máximos  $tmin_{i,s}$  e  $tmax_{i,s}$ , respectivamente, associados para completá-los;
- Como há uma possível alteração de velocidade a cada troca de trechos, não é recomendável que os robôs alterem-na de forma brusca. Assim, os robôs devem respeitar um certo limite percentual na mudança de velocidade. Esse fato é representado pelas restrições de rampa relacionada às velocidades consecutivas.

### 3.2.3.1 Função Objetivo

Como o objetivo é evitar que os robôs estejam em um mesmo ponto de colisão no mesmo momento, independente de ser fictício ou não, fazer com que o intervalo entre a chegada dos robôs ao ponto de interseção seja elevado pode ser uma boa solução. Abordagens usando PLIM maximizam a diferença entre os tempos de chegada e as restrições consideram uma variável binária que indica qual robô chegou primeiro ao ponto. A maneira de remover essas restrições e deixar a formulação livre de variáveis inteiras é criar uma expressão quadrática. Através desta estratégia, além de resolver esta situação, devido ao fato de ser uma função de maximização, ainda faz com que pequenos valores de diferença de tempo sejam penalizados e que os valores elevados se beneficiem. Assim, a função objetivo irá maximizar o intervalo mínimo de tempo em que os robôs passarão por um ponto em comum,  $\Delta t$ . A função objetivo não inclui número de ciclos, parâmetros de

aceleração e velocidade dos robôs, sendo esses, abordados pelas restrições. O intervalo de tempo é representado conforme pode ser observado pela expressão 3.2.

$$\Delta t = \min(TC_{i,j} - TC_{k,j})^2, \quad (3.2)$$

$$i \neq k, i \in \mathcal{D}_j, k \in \mathcal{D}_j, j \in \mathcal{J}_i$$

onde:

$\Delta t$  Menor diferença de tempo entre a chegada de robôs a um ponto de colisão;

$TC_{i,j}$  Tempo que o robô  $i$ , leva para atingir o ponto de colisão  $j$ .

O tempo de chegada ao ponto de colisão  $j$  pode ser descrito como a soma dos tempos que o robô despense em cada trecho  $s$  até o trecho correspondente ao ponto de colisão. Assim como é descrito abaixo:

$$TC_{i,j} = \sum_{s=1}^j t_{i,s}, \quad (3.3)$$

$$i \in \mathcal{R}, j \in \mathcal{J}_i, s \in \mathcal{S}_i$$

onde:

$t_{i,s}$  Tempo que o robô  $i$  leva para se mover através do trecho  $s$ .

### 3.2.3.2 Restrições

Os robôs devem obedecer o tempo predeterminado para completarem seus ciclos. Dessa forma, a soma dos tempos que cada robô gasta para passar por cada trecho  $s$  deve ser igual ao tempo de conclusão  $T_i$ , conforme a seguinte restrição:

$$\sum_{s=1}^{n_t} t_{i,s} = T_i, i \in \mathcal{R}, s \in \mathcal{S}_i \quad (3.4)$$

sendo que os seguintes componentes representam:

$n_t$  Número de trechos;

$T_i$  Tempo que o robô  $i$  tem para terminar sua trajetória.

Como já dito, o tempo  $T_i$  de cada robô deve ser relacionado com o tempo dos outros robôs através da constante  $\gamma_i$ . Essas duas constantes são dados do problema.

Os limites de velocidade impostos ao robô, devem ser considerados para essa formulação. Para cada trecho, deve-se calcular o limite de tempo mínimo e máximo que cada robô pode despendido nele, sendo associados às velocidades máxima e mínima, respectivamente. Os limites de tempo, são dados pelas seguintes expressões:

$$tmin_{i,s} = XC_{i,s}/Vmax_i \quad (3.5)$$

$$tmax_{i,s} = XC_{i,s}/Vmin_i \quad (3.6)$$

onde:

$tmin_{i,s}$  Tempo mínimo a ser despendido pelo robô  $i$  no trecho  $s$ ;

$tmax_{i,s}$  Tempo máximo a ser despendido pelo robô  $i$  no trecho  $s$ ;

$XC_{i,s}$  Comprimento do trecho  $s$  do caminho do robô  $i$ ;

$Vmin_i$  Velocidade mínima permitida do robô  $i$ ;

$Vmax_i$  Velocidade máxima permitida do robô  $i$ .

Assim, a restrição associada aos limites de velocidade do robô impõe que o tempo despendido por cada robô em um determinado trecho não deve ultrapassar os limites de tempo associados e pode ser escrita da seguinte forma:

$$tmin_{i,s} \leq t_{i,s} \leq tmax_{i,s}, \quad i \in \mathcal{R}, \quad s \in \mathcal{S}_i \quad (3.7)$$

As restrições de velocidade fazem com que não sejam planejadas velocidades maiores ou menores do que é permitido. Porém, a possibilidade de existência de um perfil de velocidade com mudanças abruptas é muito grande. Pode ocorrer, por exemplo, de em um determinado trecho a velocidade ser a máxima e no próximo passar para a mínima, impossibilitando a desaceleração em alguns casos reais. Dessa forma, é necessário criar restrições que não permitam diferenças de velocidade muito grandes entre os trechos, as chamadas restrições de limite de rampa. As desigualdades 3.8 e 3.9 representam tais restrições.

$$|v_{i,s} - v_{i,s-1}| \leq Rp_i, \quad i \in \mathcal{R}, \quad \forall s \geq 2, \quad s \in \mathcal{S}_i \quad (3.8)$$

$$|v_{i,s} - v_{i,n_t}| \leq Rp_i, \quad i \in \mathcal{R}, \quad s = 1 \quad (3.9)$$

onde:

$v_{i,s}$  Velocidade do robô  $i$  no trecho  $s$ ;

$Rp_i$  Variação de velocidade máxima permitida entre trechos para o robô  $i$ .

Essa restrição com relação ao tempo em cada trecho, pode ser reescrita como:

$$\left| \frac{XC_{i,s}}{t_{i,s}} - \frac{XC_{i,s-1}}{t_{i,s-1}} \right| \leq Rp_i, \quad i \in \mathcal{R}, \quad \forall s \geq 2, \quad s \in \mathcal{S}_i \quad (3.10)$$

$$\left| \frac{XC_{i,s}}{t_{i,s}} - \frac{XC_{i,n_t}}{t_{i,n_t}} \right| \leq Rp_i, \quad i \in \mathcal{R}, \quad s = 1, \quad s \in \mathcal{S}_i \quad (3.11)$$

### 3.2.3.3 Formulação Proposta

Dado que o objetivo é evitar colisões entre os robôs em pontos de interseção na trajetória, a modelagem deve permitir a maximização do menor intervalo de tempo em os robôs chegarão ao ponto de colisão. Portanto, em resumo, a formulação do problema é a seguinte:

$$Max \Delta t \quad (3.12)$$

sujeito à:

Restrição temporal:

$$\sum_{s=1}^{n_t} t_{i,s} = T_i, \quad i \in \mathcal{R}, \quad s \in \mathcal{S}_i \quad (3.13)$$

Restrições relacionadas aos limites de velocidade:

$$tmin_{i,s} \leq t_{i,s} \leq tmax_{i,s}, \quad i \in \mathcal{R}, \quad s \in \mathcal{S}_i \quad (3.14)$$

Restrições de limite de rampa de velocidade:

$$\left| \frac{XC_{i,s}}{t_{i,s}} - \frac{XC_{i,s-1}}{t_{i,s-1}} \right| \leq Rp_i, \quad i \in \mathcal{R}, \quad \forall s \geq 2, \quad s \in \mathcal{S}_i \quad (3.15)$$

$$\left| \frac{XC_{i,s}}{t_{i,s}} - \frac{XC_{i,n_t}}{t_{i,n_t}} \right| \leq Rp_i, \quad i \in \mathcal{R}, \quad s = 1, \quad s \in \mathcal{S}_i \quad (3.16)$$

Essa formulação possui função objetivo não linear, restrições 3.13 e 3.14 lineares e restrições 3.15 e 3.16 não lineares, fato que permite a resolução através de técnicas baseadas em Programação Não-Linear.

### 3.2.4 Processo de Resolução

Usando esta formulação com base na maximização do tempo em que os robôs passam por um ponto comum de seus caminhos, consegue-se obter o tempo de chegada dos robôs em cada ponto de interseção dos caminhos. A descrição detalhada para resolução utilizando a formulação de PNL é descrita abaixo.

---

**Algoritmo 1:** Coordenação de um conjunto de robôs móveis  $\mathcal{R}$

---

**1: Entrada de dados**

$$\begin{aligned} \mathcal{R} &= \{R_1, R_2, \dots, R_i\} \\ \mathcal{P}o &= \{Po_1, Po_2, \dots, Po_i\} \\ \mathcal{C} &= \{C_1, C_2, \dots, C_i\} \\ \mathcal{J}_i &= \{CP_1, CP_2, \dots, CP_i\} \\ \mathcal{D}_j &= \{R_i, R_k\}, i \neq k \\ \mathcal{S}_i &= \{XC_{i,1}, XC_{i,2}, \dots, XC_{i,s}\} \\ i, k &\in \{1, 2, \dots, n_R\}, j, s \in \{1, 2, \dots, n_{cp}\} \\ Vmin_{i,j}, Vmax_{i,j}, T_i \end{aligned}$$

**2: Inicializa variáveis**

$$\begin{aligned} n_{tp} &= n_{cp} \\ n_{fp} &= 0 \\ niter &= 0 \end{aligned}$$

**3: Cálculo do número de pontos fictícios**

$$\begin{aligned} n_{tp} &= n_{cp} + n_{fp} \\ \text{Atualiza conjuntos: } &\mathcal{J}_i, \mathcal{D}_j, \mathcal{S}_i \\ \text{Saída: } &tmin_i, tmax_i \end{aligned}$$

**4: Executa pacote de PNL para resolver o problema**

$$\begin{aligned} \text{Saída: } &t_{i,s} \\ \text{Calcula: } &v_{i,s} \end{aligned}$$

**5: para cada SC faça**

**6: se Colisão for detectada então**

**Atualiza pontos fictícios**

$$\begin{aligned} niter &= niter + 1 \\ n_{fp} &= n_{fp} + 1 \\ \text{Retorna ao Passo 3} \end{aligned}$$

**7: senão**

Continua com o próximo Passo

**fim se**

**fim para**

**8: Saída:**  $V_{i,s}$

---

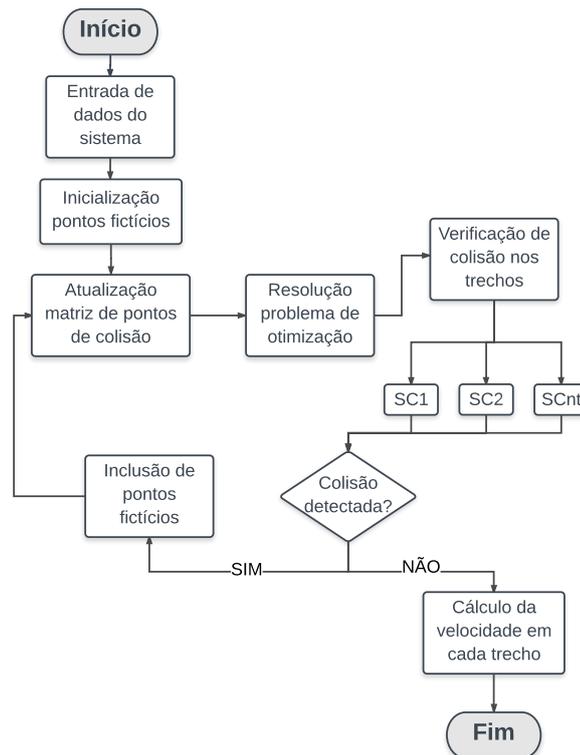
Os comentários a seguir ajudam a explicar o algoritmo proposto:

- Passo 1** : Conjuntos e dados de entrada, como limites de velocidade, o tempo total em que o robô deve completar suas trajetórias, o número de robôs,  $n_R$  e o número de pontos de colisão,  $n_{cp}$ . Inicialmente, os conjuntos considerados como pontos de colisão englobam os pontos de interseção e as extremidades dos segmentos de colisão.
- Passo 2** : Neste primeiro momento, o número de pontos de colisão fictícios,  $n_{fp}$ , é igual a zero. O número de iterações  $n_{iter}$  é inicializado como zero. E o número total de pontos de colisão,  $n_{tp}$ , é igual ao número de pontos de colisão detectados.
- Passo 3** : Cálculo do número total de pontos considerando os pontos de colisão e pontos fictícios. E assim, é realizada a atualização do conjunto de pontos de colisão  $\mathcal{J}_i$  e do conjunto de robôs que podem colidir no ponto de colisão  $j$ ,  $\mathcal{D}_j$ . Os limites de tempo,  $t_{min_{i,j}}$  e  $t_{max_{i,j}}$ , são obtidos nesta etapa.
- Passo 4** : Resolução do problema cuja formulação apresentada é apresentada pelas expressões 3.12 a 3.16. Como resultado é encontrado o valor de tempo em cada trecho para cada robô,  $t_{i,j}$ . A partir daí, consegue-se obter a velocidade em cada trecho  $v_{i,s}$ .
- Passo 5** : Para cada segmento de colisão, é verificada a possibilidade de colisão, que é realizada através da análise das diferenças de tempo em que os robôs atingem as extremidades do segmento de colisão. Se essas diferenças assumirem sinais opostos, a colisão é detectada. Já caso elas assumam mesmo sinal, isso significa que os robôs passarão pelo segmento sem acidentes, encontrando o melhor perfil de velocidade.
- Passo 6** : Se a colisão for confirmada, um novo ponto de colisão é incluído entre as extremidades de cada trecho. Após a inclusão dos novos pontos fictícios, retorna-se ao passo 3, momento em que o número total de pontos é atualizado. Este fato muda drasticamente as velocidades a fim de evitar falhas no trecho, bem como nos pontos de colisão.
- Passo 7** : Caso não sejam detectadas colisões ao longo do caminho dos robôs, a formulação utiliza o menor número de pontos fictícios possível para simplificar a formulação.
- Passo 8** : A solução final é composta pela velocidade ao longo tempo  $V_{i,j}$  para cada robô que garanta um caminho livre de colisões usando o mínimo de pontos fictícios possível.

Embora a resolução do problema envolvendo segmentos de colisão seja equivalente à formulação de pontos de colisão, a inclusão de segmentos aumenta o número de pontos fictícios, alterando a matriz de distância entre os trechos e conseqüentemente os limites do tempo máximo e mínimo que o robô pode gastar em cada trecho. O número de pontos fictícios  $n_{fp}$  é determinado pela verificação da possibilidade de colisão. De forma simplificada, a verificação de não colisão é realizada primeiro com o menor número possível

de pontos fictícios: dois pontos nas extremidades e nenhum ponto fictício no segmento. Se esse arranjo de pontos não garantir um caminho livre, deve-se incluir um novo ponto fictício no meio do segmento. Posteriormente, é verificada a possibilidade de colisão. No momento em que não for detectado nenhum acidente, o processo iterativo termina. O fluxograma apresentado pela Figura 12 auxilia o entendimento do processo de solução descrito anteriormente.

Figura 12: Fluxograma de solução do problema de Coordenação de Robôs Móveis.



Fonte: Elaborada pelo autor

### 3.3 COORDENAÇÃO DE MÚLTIPLOS ROBÔS MANIPULADORES

Considerando apenas a coordenação dos movimentos, os problemas envolvendo robôs móveis são análogos aos manipuladores porque o interesse está em encontrar uma trajetória livre de acidentes. No entanto, no caso de braços articulados, as trajetórias são definidas por conjuntos de coordenadas cartesianas resultantes da cinemática direta obtida pelas posições das juntas ao longo do tempo. A abordagem proposta determina a trajetória das articulações do braço que gera um caminho livre de acidentes.

Os manipuladores consistem em elos quase rígidos que são conectados por juntas que permitem o movimento relativo dos elos vizinhos. No caso das juntas rotacionais, tais deslocamentos são chamados de ângulos de junta. O número de graus de liberdade que um manipulador possui é relativo ao número de variáveis de posição independentes que teriam

de ser especificadas para se localizarem todas as peças do mecanismo. Normalmente em manipuladores industriais o número de juntas é igual ao de graus de liberdade [26].

Na ponta livre da cadeia de elos que forma o manipulador fica o efetuador. Dependendo da aplicação, o efetuador pode ser uma garra, um maçarico de solda ou outro dispositivo. Um problema relacionado ao estudo da manipulação chama-se cinemática direta. Dado um conjunto de ângulos de junta, o problema da cinemática direta é computar a posição e a orientação do sistema de referência da ferramenta com relação ao sistema base.

Situações que envolvem robôs que compartilham um ambiente comum são particularmente relevantes na indústria automotiva, onde vários robôs realizam operações de soldagem, por exemplo. É importante que os movimentos das juntas desses manipuladores sejam coordenados de forma a evitar colisões entre os robôs pertencentes ao sistema.

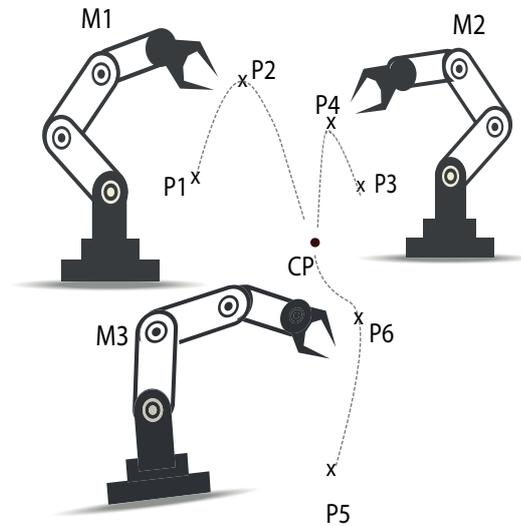
### 3.3.1 Descrição do Problema

Em uma célula de trabalho robótica, um conjunto de braços articulados devem executar certas tarefas em um mesmo espaço de trabalho. Essas atividades são cronometradas e devem ser periódicas e persistentes. Considerando que os robôs têm pontos predeterminados a serem percorridos, o objetivo é determinar a trajetória ótima do efetuador dos manipuladores, a fim de evitar colisões em pontos coincidentes do seu caminho. Essa trajetória pode ser entendida como o conjunto de configurações de velocidade das juntas de cada robô ao longo do tempo que leva o robô para o ponto desejado.

A fim de exemplificar, uma situação hipotética simples é descrita. Três braços robóticos, Robô  $M_1$ , Robô  $M_2$  e Robô  $M_3$  que trabalham juntos no mesmo espaço de trabalho têm tarefas a serem cumpridas. Eles devem obedecer à imposição de passar através de pontos predeterminados ( $P_n$ ) para completar sua atividade com sucesso. Os caminhos propostos para os manipuladores têm um ponto de coincidência, chamado ponto comum ( $PC$ ). E, para que eles completem seus caminhos, deve haver uma maneira de evitar colisões neste ponto. A Figura 13 ilustra essa situação. Os três caminhos representados na figura são apenas uma ilustração de como o efetuador do robô poderia se mover para percorrer os pontos especificados.

O caminho a ser coberto pelo manipulador é representado por um conjunto de poses dos robôs, compostas pelas coordenadas cartesianas e pela configuração angular das juntas, nas quais há a necessidade de passagem do efetuador. No caso acima mencionado, cada robô tem três pontos de passagem onde é obrigatório passar, formando seus caminhos. O caminho do robô  $R_1$ , por exemplo, é definido por  $C_1 = \{P_1, P_2, CP, P_2, P_3\}$ . Entre os pontos obrigatórios de passagem e pontos de colisão, os trechos são formados. O objetivo é determinar a trajetória de cada robô em cada seção. A trajetória envolve a descrição temporal da posição e velocidade para cada grau de liberdade do manipulador.

Figura 13: Sistema de manipuladores compartilhando um mesmo espaço de trabalho.



Fonte: Elaborada pelo autor

Assim como no caso de robôs móveis, o problema é encontrar a velocidade em cada trecho que evite possíveis colisões. A nova situação apenas exige que esta velocidade seja tratada em termos de articulações robóticas. Desta forma, pode-se usar o mesmo método proposto adaptado para resolver o problema de coordenação de manipuladores.

### 3.3.2 Geração de Trajetórias

O problema de geração de trajetória de um robô manipulador deve considerar qual o objetivo a ser alcançado. Dada as posições inicial e final das juntas, deve-se calcular uma trajetória entre a posição atual e o próximo objetivo. A trajetória a ser cumprida, determina os pontos intermediários que o efetuador do manipulador deve percorrer. Esses pontos, correspondem a configurações angulares das juntas que, ao longo do tempo, dependem da velocidade e aceleração de cada articulação. O manipulador deve mover-se suavemente da posição atual para a posição desejada de modo a não gerar acelerações muito altas que possam forçar os atuadores ou movimentos repentinos que poderiam danificar a estrutura mecânica do manipulador. Para esse propósito considera-se suave uma função contínua e que tem primeira derivada contínua. Às vezes, também é desejável que a segunda derivada também seja contínua.

A trajetória, no caso de manipuladores, é um caminho sujeito a restrições temporais. A geração de trajetória envolve a descrição temporal da posição, velocidade e aceleração para cada grau de liberdade do manipulador. Este processo pode ser executado antes da sua execução, geração offline, ou em tempo real, à medida que é executado.

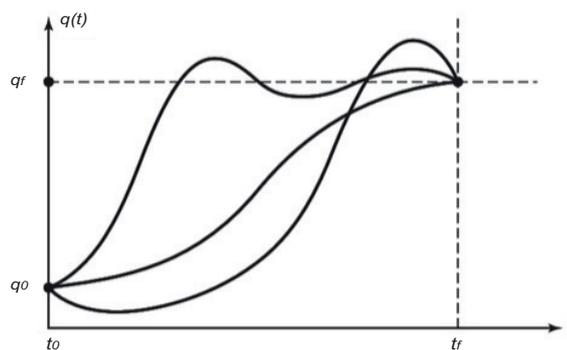
Com o objetivo de determinar a trajetória, devem ser conhecidas as poses inicial e final do robô, ou seja sua posição cartesiana e a orientação de suas juntas. No caso de vários

pontos a serem atingidos, deve-se conhecer a pose do robô em cada ponto intermediário do caminho. Além disso, deve ser especificada a duração do movimento do robô no trecho considerado.

Caso não se conheça a configuração das juntas inicial e final do robô, mas apenas as coordenadas cartesianas no efetuador, a geração depende do cômputo da cinemática inversa do manipulador. A cinemática inversa é o processo de determinação dos parâmetros de uma junta em um objeto flexível, de forma a alcançar o posicionamento adequado. Os pontos intermediários obtidos são convertidos no espaço das juntas. Já quando se possui essa dado, isto é, são conhecidas as poses do robô em cada ponto do caminho, a geração é obtida através de interpolação no espaço das juntas. E o ponto cartesiano em que o efetuador se encontra é calculado através da cinemática direta. Na cinemática direta deseja-se obter a posição e orientação do atuador no espaço cartesiano da base, a partir das posições das articulações. Como no caso desse trabalho são conhecidas as poses dos robôs, a geração da trajetória será realizadas no espaço das juntas. Os esquemas de espaço das juntas são, de modo geral, mais fáceis de computar por, essencialmente, não existirem problemas com singularidades do mecanismo.

O problema básico consiste na forma de selecionar uma trajetória entre as posição inicial  $p_0$  e final  $p_f$  do efetuador que se enquadre num tempo determinado  $t_d$ . Existem vários métodos que podem ser aplicados [27] e várias funções suaves  $q(t)$  que podem ser usadas para interpolar a trajetória das juntas de uma angulação  $q_0$  para  $q_f$ , conforme ilustrado pela Figura 14.

Figura 14: Possíveis formatos para trajetória de uma junta.



Fonte: Adaptado de [26]

Uma forma simples e confiável é utilizar uma função polinomial ao longo do tempo para interpolar as trajetórias das juntas [28] e [29]. Os polinômios são uma escolha conveniente para a representação da trajetória porque suas propriedades são bem conhecidas e suas derivadas podem ser facilmente calculados e também são polinomiais.

Supondo que as configurações das juntas inicial  $q_0$  e final  $q_f$ , associadas às posições

$p_0$  e  $p_f$ , respectivamente, sejam conhecidas. Se somente  $p_0$  e  $p_f$  sejam dadas,  $q_0$  e  $q_f$  devem ser determinados pela cinemática inversa. A variável arbitrária  $q(t)$  pode ser utilizada para interpolar a configuração angular de uma junta ao longo de um tempo  $t$ , sendo que seu valor no tempo  $t_0$  é a posição inicial e no tempo  $t_f$ , a posição final desejada. Dessa forma, pode-se determinar as seguintes restrições, relacionadas ao posicionamento:

$$q(0) = q_0 \quad (3.17)$$

$$q(t_f) = q_f \quad (3.18)$$

Além disso, também deseja-se que a função seja contínua com relação à velocidade. Nessa situação serão consideradas que a velocidade inicial e final sejam nulas. Assim, a derivada da função nos instantes inicial e final deve ser igual a zero, conforme as restrições 3.19 e 3.20.

$$\dot{q}(0) = 0 \quad (3.19)$$

$$\dot{q}(t_f) = 0 \quad (3.20)$$

Uma função polinomial cúbica pode ser utilizada caso somente essas restrições sejam relacionadas a fim de encontrar a trajetória. Porém, se houver a necessidade de especificar posição, velocidade e aceleração no início e fim do segmento, um polinômio de quinta ordem se faz necessário, como mostrado abaixo.

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (3.21)$$

onde  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  e  $a_5$  são parâmetros determinados pelas restrições impostas, que são as seguintes:

$$\begin{aligned} q_0 &= a_0; \\ q_f &= a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5; \\ \dot{q}_0 &= a_1; \\ \dot{q}_f &= a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4; \\ \ddot{q}_0 &= 2a_2; \\ \ddot{q}_f &= 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3 \end{aligned} \quad (3.22)$$

Tais restrições especificam um conjunto linear de seis equações com seis incógnitas, cuja solução é:

$$\begin{aligned}
a_0 &= q_0 \\
a_1 &= \dot{q}_0 \\
a_2 &= \frac{\ddot{q}_0}{2} \\
a_3 &= \frac{20(q_f - q_0) - (8\dot{q}_f + 12\dot{q}_0)t_f + (3\ddot{q}_0 - \ddot{q}_f)t_f^2}{2t_f^3} \\
a_4 &= \frac{30(q_0 - q_f) + (14\dot{q}_f + 16\dot{q}_0)t_f - (2\ddot{q}_f - 3\ddot{q}_0)t_f^2}{2t_f^4} \\
a_5 &= \frac{12(q_f - q_0) - (6\dot{q}_f + 6\dot{q}_0)t_f + (\ddot{q}_f - \ddot{q}_0)t_f^2}{2t_f^5}
\end{aligned} \tag{3.23}$$

### 3.3.3 Modelagem do Problema de Otimização

Considerando a finalidade de adaptar a formulação não linear ao caso de braços robóticos, a modelagem será apresentada com base em  $n_M$  robôs manipuladores com  $n_{dof}$  graus de liberdade. Seus caminhos terão  $n_{pc}$  pontos comuns e  $n_s$  trechos. Sendo que os trechos são compreendidos entre pontos de passagem.

Assim como na modelagem para robôs móveis, a velocidade no início da trajetória deve ser nula e o tempo de conclusão da trajetória deve ser igual a  $T_i$ . Além disso, os obstáculos são exclusivamente móveis, ou seja, no caminho determinado, os robôs somente tem a possibilidade de colisão com eles mesmos. Já especificadamente no caso dos manipuladores, o robô possui pontos de passagem, representados por determinadas configurações de juntas, que devem ser sequenciados de forma a completar sua trajetória. Nesses pontos a velocidade das juntas deve ser nula, pois neles, eles terão que desempenhar tarefas especificadas. Essas tarefas não são descritas no presente trabalho e considera que nesses pontos todos eles a desempenham no mesmo intervalo tempo.

O tempo que cada robô gasta em cada trecho entre pontos de passagem pode ser usado para calcular as velocidades angulares de suas juntas, sabendo as poses inicial e final. Desta forma, se o resultado do problema for o tempo em cada seção, a velocidade pode ser obtida manipulando essa solução e encontrando a trajetória do robô para atingir a determinada pose final. Assim, a otimização baseada na abordagem de diferença de tempo, como mencionado para os sistemas de múltiplos robôs móveis, também pode ser aplicada a este caso.

Os conjuntos considerados para modelagem do problema de coordenação de movimento de manipuladores são os seguintes:

$\mathcal{M} = \{M_1, M_2, \dots, M_i\}$  Representa o conjunto de braços robóticos, indexado por  $i$ .

$\mathcal{P}_i = \{P_1, P_2, \dots, P_l\}$  Representa o conjunto de pontos a serem percorridos pelo robô  $i$ , indexado por  $l$ .

$\mathcal{J}_i = \{PC_1, PC_2, \dots, PC_j\}$  Representa o conjunto de pontos de colisão presentes no caminho do robô  $i$ , indexado por  $j$ .

$\mathcal{S}_i = \{S_1, S_2, \dots, S_s\}$  Representa o conjunto de trechos presentes no caminho do robô  $i$ , indexado por  $s$ .

$\mathcal{D}_j \in \mathcal{M}$  Representa o conjunto de robôs que passarão por um mesmo ponto de colisão  $j$ .

### 3.3.3.1 Função Objetivo

Indo ao encontro da solução proposta para robôs móveis, já que nesse caso o objetivo também é evitar que o efetuador esteja presente no mesmo local em um mesmo momento, a função objetivo também pode considerar a maximização da diferença de tempo em que os robôs se encontrarão em um ponto comum entre seus caminhos. Portanto a função objetivo permanece a mesma nessa situação, ou seja a maximização da soma do quadrado da diferença de tempo em que os robôs passarão por um ponto coincidente, presente na Expressão 3.2.

### 3.3.3.2 Restrições

Uma das restrições da formulação é obtida pelos limites de tempo impostos. A soma do tempo gasto pelos robôs em cada seção  $t_{i,s}$  deve ser igual ao tempo total que o robô  $i$  gasta para completar um ciclo de movimento. E assim como a formulação já mencionada os ciclos devem ser comensuráveis.

A diferença da formulação para manipuladores consiste na representação do limite de velocidade. As juntas possuem limites de velocidade máximos que não devem ser ultrapassados. E, para que isso aconteça, o limite de tempo mínimo  $tmin_{i,s}$  em cada trecho é sujeito à essas limitações dinâmicas. Ele é dado pelo limite máximo de velocidade angular  $\omega_{max_g}$  para cada junta  $g$ . O valor de limite de tempo mínimo é obtido, verificando o menor valor de tempo que respeita a restrição de velocidade.

Considerando  $\dot{q}_g(t)$  como uma função de velocidade da junta  $g$  ao longo do tempo, em cada seção, o valor absoluto desta função em cada instante, não pode exceder o valor que está definido como limite superior de velocidade angular para aquela articulação. A escolha de  $tmin_{i,s}$  pode ser tratada como um problema de otimização secundário em que o objetivo é minimizar o tempo em cada trecho, sujeito a restrição de velocidade. Como

pode ser descrito pela Formulação 3.24.

$$\begin{aligned}
 & \text{Min } tmin_{i,s} \\
 & \text{sujeito à :} \\
 & \max[abs(\dot{q}_g(t))] < \omega max_g \\
 & i \in \mathcal{M}, s \in \mathcal{S}
 \end{aligned} \tag{3.24}$$

onde os seguintes parâmetros representam:

$tmin_{i,s}$  Tempo mínimo que o robô  $i$  pode gastar no trecho  $s$ ;

$\dot{q}_g(t)$  Velocidade angular da junta  $g$  ao longo do tempo  $t$ ;

$g \in \{1, 2, \dots, n_{dof}\}$ ;

$n_{dof}$  Número de juntas do robô;

$t$  Variável de tempo;

$\omega max_g$  Limite superior de velocidade angular para a junta  $g$ .

Encontrando o limite de tempo mínimo que o robô pode gastar em um determinado trecho, determina-se a restrição relacionada indiretamente com a velocidade:

$$t_{i,s} \geq tmin_{i,s}, i \in \mathcal{M}, s \in \mathcal{S} \tag{3.25}$$

onde:

$t_{i,s}$  Tempo que o robô  $i$  gasta no trecho  $s$ ;

### 3.3.3.3 Formulação Proposta

Apesar da função objetivo e restrição de tempo serem idênticas às da formulação para robôs móveis, a peculiaridade considerando manipuladores robóticos é observada na restrição de velocidade, já que nessa situação é utilizada a limitação de velocidade angular para cada junta. A formulação final pode ser observada abaixo.

$$\text{Max } \Delta t \tag{3.26}$$

sujeito à:

Restrição temporal:

$$\sum_{s=1}^{n_t} t_{i,s} = T_i, i \in \mathcal{M}, s \in \mathcal{S}_i \tag{3.27}$$

Restrições relacionadas aos limites de velocidade:

$$t_{i,s} \geq tmin_{i,s}, \quad i \in \mathcal{M}, \quad s \in \mathcal{S} \quad (3.28)$$

Sendo que  $tmin_{i,s}$  é obtido através do sub-problema de otimização que utiliza o limite de velocidade máximo de cada junta.

### 3.3.4 Processo de Resolução

Usando a formulação descrita baseada na maximização do tempo em que os robôs passam por um ponto comum de seus caminhos, consegue-se obter o tempo de chegada dos robôs em cada ponto obrigatório da passagem. De acordo com o resultado obtido, o tempo em cada seção é usado para determinar os parâmetros da função que determina a trajetória das juntas (Expressão 3.21) e assim obter  $\theta_g(t)$ ,  $\dot{\theta}_g(t)$  e  $\ddot{\theta}_g(t)$ . Para demonstrar os passos de resolução do problema de otimização, é apresentado o Algoritmo 2.

---

**Algoritmo 2:** Coordenação de um conjunto de manipuladores robóticos  $\mathcal{M}$ .

---

1: **Entrada de dados**

$\mathcal{M}, \mathcal{Q}_i, \mathcal{J}_i, \mathcal{D}_j, \mathcal{S}_i$

$T_i, i \in \mathcal{M}$

$\omega max_g, g \in \{1, 2, \dots, n_{dof}\}$

2: **Resolve problema contido na Formulação 3.24**

Saída:  $tmin_{i,s}$

3: **Executa pacote de PNL para resolver o problema principal da Formulação 3.26 a 3.28**

Saída:  $t_{i,s}, i \in \mathcal{M}, s \in \mathcal{S}_i$

4: **para Cada Robô  $i$  faça**

5:     **para Cada trecho  $s$  faça**

        Calcula  $a0, a1, a2, a3, a4, a5$  através das equações 3.23

**fim para**

**fim para**

6: **Saída:**  $\theta_g(t), \dot{\theta}_g(t), \ddot{\theta}_g(t)$

---

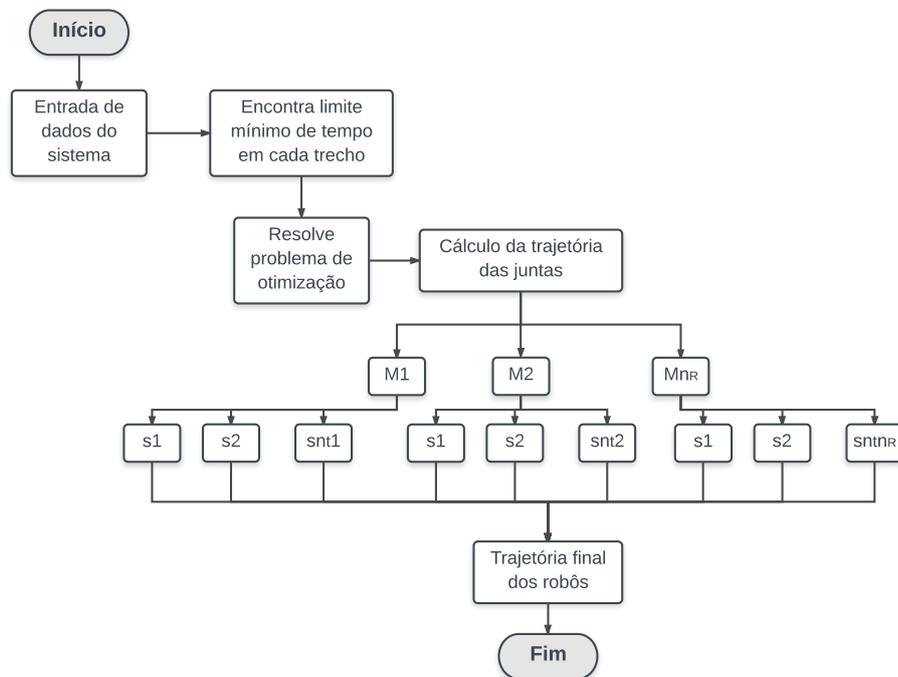
Os comentários a seguir ajudam a explicar o algoritmo proposto:

**Passo 1 :** Conjuntos e dados de entrada. Dentre eles, o tempo de conclusão de um ciclo para cada robô  $T_i$ , velocidade máxima de cada junta  $\omega max_g$  e número de graus de liberdade dos robôs  $n_{dof}$ . Como apenas a pose do robô é considerada, dados para o cálculo da cinemática dos manipuladores não são necessários.

- Passo 2** : Resolver o problema proposto pela Formulação 3.24. Nesse passo serão obtidos os limites mínimo de tempo em cada trecho,  $t_{min_{i,s}}$  dados pela velocidade angular máxima permitida para a junta.
- Passo 3** : Encontrar o tempo despendido pelo manipulador em cada trecho,  $t_{i,s}$  através da resolução do problema de otimização principal dado pela Formulação 3.26 a 3.28.
- Passo 4 e 5** : Para cada robô, de um total de  $n_R$  e cada trecho, de um total de  $n_t$ , os coeficientes da Equação 3.23 serão atribuídos de acordo com o tempo em que os robôs passarão por cada seção. Para determinar a função de trajetória das articulações ao longo do tempo,  $\theta_g(t)$ , o cálculo dos coeficientes será baseado no conjunto de Equações 3.23, sendo que  $t_f$  receberá o tempo de cada seção,  $t_{i,s}$ .
- Passo 6** : O resultado é composto das funções que descrevem a trajetória de cada articulação: a função dos movimentos das juntas,  $\theta_g(t)$ ; a função que descreve a velocidade angular das articulações,  $\dot{\theta}_g(t)$ ; e a aceleração das articulações ao longo do tempo,  $\ddot{\theta}_g(t)$ .

O fluxograma disposto pela Figura 15 auxilia o entendimento do processo de solução anteriormente descrito.

Figura 15: Fluxograma de solução do problema de Coordenação de Robôs Manipuladores.



Fonte: Elaborada pelo autor

Seguindo esses passos consegue-se determinar a trajetória de cada junta dos robôs envolvidos no sistema ao longo do tempo. E dessa forma, aplicar a função de velocidade

e aceleração ao movimento dos manipuladores. O resultado esperado é um movimento suave para cada manipulador e de forma conjunta, caminhos livres de colisões.

### 3.4 FORMA DE RESOLUÇÃO DO PROBLEMA DE OTIMIZAÇÃO

O método proposto por esse trabalho contempla um problema de otimização não linear. A implementação desse método de otimização não é o foco do presente trabalho, sendo apenas uma ferramenta de resolução do problema descrito. Portanto, pode-se haver flexibilidade na escolha do algoritmo a ser utilizado na resolução do problema.

A fim de realizar os testes que virão a seguir no presente documento, será usado o Método de Pontos Interiores Barreira Logarítmica. O método descrito no Apêndice A, em forma de resumo, é implementado pela *toolbox* denominada *fmincon* do programa MATLAB, sendo essa ferramenta utilizada para a realização da etapa de otimização vislumbrada a partir da formulação proposta para coordenação dos movimentos de múltiplos robôs. A *fmincon* utiliza o método de pontos interiores valendo-se do método Barreira Logarítmica [30], assim como é explicitado na própria página do programa [31].

Para fins de comparação, foi utilizada a metodologia de Programação Linear Inteira Mista, que é amplamente aplicada para coordenação de robôs. A *toolbox* do MATLAB, *intlinprog* [32] é uma ferramenta para solucionar problemas desse tipo. Dessa forma, ela será utilizada para obter soluções e compará-las com a modelagem baseada em PNL.

### 3.5 CONSIDERAÇÕES FINAIS

Nesse capítulo, foram apresentadas as metodologias propostas para resolução do problema de coordenação dos movimentos de robôs tanto manipuladores quanto móveis. O problema envolve uma forma de evitar colisões em robôs que praticam caminhos predeterminados e específicos. Com relação ao tempo, suas trajetórias devem ser periódicas e comensuráveis. O método compreende da aplicação de uma Formulação Não Linear baseada na maximização da diferença de tempo em que os robôs chegarão em um mesmo ponto espacial de seus caminhos. Apesar de compartilhar a mesma função objetivo, a aplicação em robôs móveis e braços robóticos tem suas peculiaridades.

A formulação para robôs móveis deve considerar velocidades mínimas e máximas específicas para cada robô. E, a partir daí, calcular o tempo máximo e mínimo que o robô poderá despender em cada trecho de seu caminho. Além dessa restrição, são incluídos limites temporais e de velocidade, para que não existam grandes diferenças de velocidade na passagem de trechos. A formulação final é composta por função objetivo não linear e restrições lineares e não linear.

Já para os manipuladores, a formulação deve conter os limites de velocidade angular máximos para cada junta. Esse fator depende das características físicas do braço robótico.

Para esse problema, há a inclusão de restrições relacionadas ao limite mínimo de tempo que o robô pode gastar no trecho, o qual depende da velocidade máxima permitida. Além desse limite ainda devem ser consideradas as restrições temporais relacionada com o tempo máximo para completar o movimento. O resultado desse problema de otimização não linear será a trajetória, a velocidade e a aceleração ao longo do tempo para cada junta de cada robô.

Nessas formulações não são incluídas variáveis inteiras e binárias relacionadas a número de ciclos ou ordem de chegada dos robôs ao ponto de colisão. Esse fato simplifica a formulação, que trata a colisão através da função objetivo, sem adicionar restrições com esse fim.

## 4 ESTUDO DE CASOS

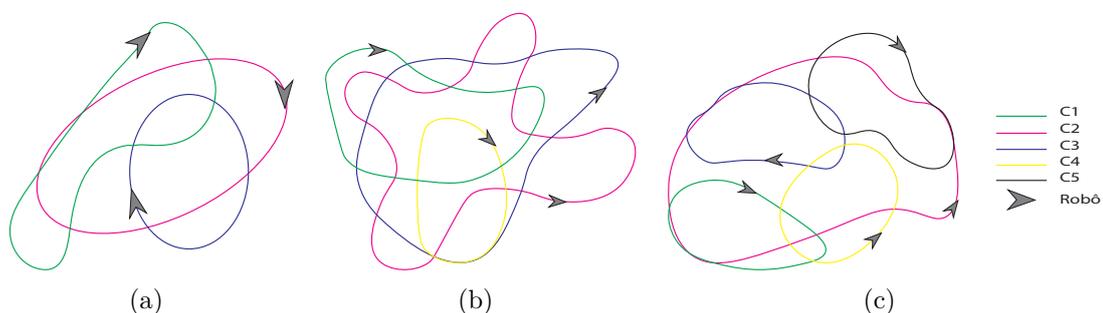
A fim de verificar a aplicabilidade da metodologia proposta baseada em Programação Não Linear, os testes serão realizados considerando sistemas de múltiplos robôs móveis e manipuladores. No caso de robôs móveis, serão determinadas velocidade mínimas e máximas permitidas para os robôs. Isso garante que eles não atinjam velocidades muito altas e nem tenham a possibilidade de parar para evitar as colisões. Já no caso dos braços robóticos devem ser determinadas velocidades angulares máximas para a movimentação das juntas. Esses valores são delimitados pelas limitações físicas do modelo robótico. Os esquemas utilizados para simulação serão apresentados a seguir para os dois casos.

### 4.1 SISTEMAS DE MÚLTIPLOS ROBÔS MÓVEIS

Os testes foram realizados com três tipos de esquemas diferentes. Os sistemas considerados serão os seguintes:

- A** : Sistema composto por três robôs, seis pontos de colisão e um segmento de colisão, com caminhos que podem ser observados pela Figura 16.a;
- B** : Sistema composto por quatro robôs, vinte pontos de colisão e um segmento de colisão, com caminhos que podem ser observados pela Figura 16.b;
- C** : Sistema composto por cinco robôs, quinze pontos de colisão e três segmentos de colisão, com caminhos que podem ser observados pela Figura 16.c.

Figura 16: Esquemas de sistemas multi-robóticos: (a) 3 robôs, (b) 4 robôs, (c) 5 robôs.



Fonte: Elaborada pelo autor

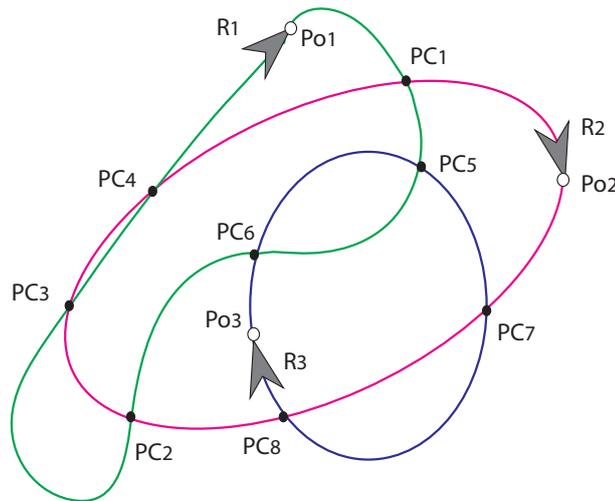
Para todos os casos, limitações de velocidade são consideradas nas simulações e são específicas em cada esquema. Para as restrições de rampa, foi determinada a variação máxima de velocidade de 30% em cada trecho apenas para avaliar a funcionalidade do método. O tempo de conclusão de um ciclo, em todos os sistemas foi especificado em 100s.

#### 4.1.1 Sistema com três robôs

Nesse caso, três robôs atravessam caminhos especificados com oito pontos de interseção entre eles. Esses pontos determinam os pontos de colisão. A Figura 17 ilustra a situação descrita com destaque para os pontos de colisão,  $PC_j$ , e os pontos de saída,  $Po_i$ , de cada robô,  $R_i$ . A sequência de pontos a serem seguidos pelos robôs juntamente com a indicação dos trechos delimitados entre os pontos são descritas abaixo:

- Robô  $R_1$  (curva verde):  $Po_1 \xrightarrow{1} PC_1 \xrightarrow{2} PC_5 \xrightarrow{3} PC_6 \xrightarrow{4} PC_2 \xrightarrow{5} PC_3 \xrightarrow{6} PC_4 \xrightarrow{7} Po_1$
- Robô  $R_2$  (curva rosa):  $Po_2 \xrightarrow{1} PC_7 \xrightarrow{2} PC_8 \xrightarrow{3} PC_2 \xrightarrow{4} PC_3 \xrightarrow{5} PC_4 \xrightarrow{6} PC_1 \xrightarrow{7} Po_2$
- Robô  $R_3$  (curva azul):  $Po_3 \xrightarrow{1} PC_6 \xrightarrow{2} PC_5 \xrightarrow{3} PC_7 \xrightarrow{4} PC_8 \xrightarrow{5} Po_3$

Figura 17: Esquema de teste com três robôs.



Fonte: Elaborada pelo autor

Pode ser observado pela Figura 17 que entre os pontos  $PC_3$  e  $PC_4$ , os robôs  $R_1$  e  $R_2$  possuem caminhos muito próximos, fato que, se os movimentos não forem corretamente coordenados, eles podem se colidir nesse trecho. Dessa forma, é determinado um segmento de colisão no sistema.

Para fins de simulação devem ser consideradas convenções de limites de velocidade e comprimento dos caminhos, que são apresentadas pela Tabela 1.

Além dos limites de velocidade, devem ser especificados os comprimentos dos trechos para cada robô,  $XC_{i,j}$ , os quais podem ser observados pela Tabela 2. Os comprimentos totais dos caminhos dos robôs  $R_1$ ,  $R_2$  e  $R_3$  são respectivamente  $518m$ ,  $506m$  e  $335m$ .

Tabela 1: Limites de velocidade - Esquema A.

	<b>R1</b>	<b>R2</b>	<b>R3</b>
$V_{min}$ (m/s)	3	3	2
$V_{max}$ (m/s)	8	8	6

Tabela 2: Comprimentos dos trechos - Esquema A.

<b>Robô</b>	<b>Comprimento (m)</b>						
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>R1</b>	59	20	79	106	131	43	80
<b>R2</b>	56	86	89	51	48	118	58
<b>R3</b>	44	71	66	131	23	0	0

A partir dos dados de entrada pode-se aplicar os passos apresentados pelo Algoritmo 1. Através da resolução do problema de otimização consegue-se determinar as velocidades médias dos robôs em cada trecho que determinam trajetórias livres de colisões. As velocidades médias obtidas para cada trecho podem ser observadas pela Tabela 3. A variação percentual máxima de velocidade entre trechos observada nesse caso foi de 29,97%.

Tabela 3: Média de velocidade e tempo em cada trecho - Esquema A com restrições de rampa.

<b>Trecho</b>	<b>Velocidade (m/s)</b>			<b>Tempo (s)</b>		
	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R1</b>	<b>R2</b>	<b>R3</b>
<b>1</b>	5,40	5,75	3,45	10,9	9,7	12,8
<b>2</b>	3,79	6,75	2,42	5,3	12,7	29,4
<b>3</b>	3,51	6,59	3,14	22,5	13,5	21,0
<b>4</b>	4,56	5,10	4,08	23,2	10,0	32,1
<b>5</b>	5,93	4,22	4,89	22,1	11,4	4,7
<b>6</b>	7,69	3,90	-	5,6	30,3	-
<b>7</b>	7,71	4,69	-	10,4	12,4	-

A Tabela 4 mostra as diferenças de tempo em cada ponto de colisão presente no sistema.

A diferença de tempo considerada é determinada pela subtração dos tempos de chegada dos robôs envolvidos no ponto de colisão em questão. No ponto  $PC_1$ , por exemplo, a diferença de tempo em que os robôs  $R_1$  e  $R_2$  o atingiram é de  $-76,71s$ . O valor negativo atribui ao robô  $R_2$  a chegada anterior a  $R_1$ . Já no ponto  $PC_2$  a diferença de tempo para os robôs  $R_1$  e  $R_2$  é de  $25,95s$ . Como o valor é positivo, o robô  $R_1$  chegou primeiro ao ponto com relação ao  $R_2$ . A variação de tempo mínima de chegada de dois robôs a um mesmo ponto de colisão resultante do processo de otimização,  $\Delta t$ , foi de  $25,95s$ . Esse fato que garante que não haja colisões no ponto considerado e nos demais.

Outra informação a ser desprendida através da Tabela 4 é a identificação de não

Tabela 4: Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema A.

Ponto de Colisão	Robôs		Diferença de tempo (s)
	R1	R2	
<b>PC1</b>	R1	R2	-76,71
<b>PC2</b>	R1	R2	25,95
<b>PC3</b>	R1	R2	38,05
<b>PC4</b>	R1	R3	32,27
<b>PC5</b>	R1	R3	-25,95
<b>PC6</b>	R1	R3	25,95
<b>PC7</b>	R2	R3	-53,42
<b>PC8</b>	R2	R3	-72,81

colisão no segmento de colisão, compreendido entre  $PC_3$  e  $PC_4$ . As diferenças de tempo nesse ponto determinam que, com o perfil de velocidade selecionado, a colisão não será detectada no segmento, pois abas possuem valores positivos, indicando que o robô  $R_1$  atingiu os dois pontos de forma antecipada, com relação à  $R_2$  não havendo o encontro dos mesmos no trecho.

Caso não sejam aplicadas as restrições de velocidade de rampa, os resultados apresentam velocidades com variações maiores a cada trecho, por não apresentarem limitações. A variação de velocidade máxima chega a 90,9% para o robô  $R_3$  na alteração de velocidade entre os trechos 2 e 3. A Tabela 5 apresenta as velocidades médias observadas para cada trecho. A variação de tempo mínima de chegada a um ponto de colisão, nesse caso, foi de 30,91s. Apesar de ser um valor maior em comparação com o caso com restrições de rampa, a elevada variação de velocidade pode impossibilitar a aplicação em um caso real.

Tabela 5: Média de velocidade e tempo em cada trecho - Esquema A sem restrições de rampa.

Trecho	Velocidade (m/s)			Tempo (s)		
	R1	R2	R3	R1	R2	R3
<b>1</b>	5,46	6,68	4,13	10,8	8,4	10,6
<b>2</b>	4,52	7,10	2,00	4,4	12,1	35,5
<b>3</b>	3,00	7,13	3,82	26,3	12,5	17,3
<b>4</b>	4,50	4,11	4,41	23,6	12,4	29,7
<b>5</b>	7,34	3,71	3,35	17,9	12,9	6,9
<b>6</b>	6,84	3,94	-	6,3	30,0	-
<b>7</b>	7,46	4,96	-	10,7	11,7	-

Para os dois casos, com restrições de rampa e sem, no segmento de colisão compreendido entre os pontos  $PC_3$  e  $PC_4$ , colisões não foram detectadas, sendo suficiente apenas uma iteração para encontrar uma trajetória que não envolva acidentes.

#### 4.1.2 Sistema com quatro robôs

O sistema é composto por quatro robôs e vinte pontos de colisão, como pode ser observado pela Figura 18. Nesse esquema, um segmento de colisão é determinado entre os pontos  $PC_{19}$  e  $PC_{20}$ . Esse segmento possibilita a colisão entre os robôs  $R_3$  e  $R_4$ . A sequência de pontos a serem seguidos pelos robôs juntamente com a indicação dos trechos delimitados entre os pontos são descritas abaixo:

- Robô  $R_1$  (curva verde):  $P_{o1} \xrightarrow{1} PC_5 \xrightarrow{2} PC_1 \xrightarrow{3} PC_2 \xrightarrow{4} PC_3 \xrightarrow{5} PC_7 \xrightarrow{6} PC_8 \xrightarrow{7} PC_4 \xrightarrow{8} PC_6 \xrightarrow{9} P_{o1}$
- Robô  $R_2$  (curva rosa):  $P_{o2} \xrightarrow{1} PC_9 \xrightarrow{2} PC_{17} \xrightarrow{3} PC_{10} \xrightarrow{4} PC_{18} \xrightarrow{5} PC_{11} \xrightarrow{6} PC_4 \xrightarrow{7} PC_{12} \xrightarrow{8} PC_{13} \xrightarrow{9} PC_1 \xrightarrow{10} PC_{14} \xrightarrow{11} PC_{15} \xrightarrow{12} PC_2 \xrightarrow{13} PC_3 \xrightarrow{14} PC_{16} \xrightarrow{15} P_{o2}$
- Robô  $R_3$  (curva azul):  $P_{o3} \xrightarrow{1} PC_{14} \xrightarrow{2} PC_{15} \xrightarrow{3} PC_{16} \xrightarrow{4} PC_9 \xrightarrow{5} PC_{19} \xrightarrow{6} PC_{10} \xrightarrow{7} PC_{20} \xrightarrow{8} PC_{11} \xrightarrow{9} PC_6 \xrightarrow{10} PC_{12} \xrightarrow{11} PC_{13} \xrightarrow{12} PC_5 \xrightarrow{13} P_{o3}$
- Robô  $R_4$  (curva amarela):  $P_{o4} \xrightarrow{1} PC_{17} \xrightarrow{2} PC_{19} \xrightarrow{3} PC_{18} \xrightarrow{4} PC_{20} \xrightarrow{5} PC_8 \xrightarrow{6} PC_7 \xrightarrow{7} P_{o4}$

Os limites de velocidade são apresentados pela Tabela 6. E os comprimentos dos trechos estão dispostos na Tabela 7.

Tabela 6: Limites de velocidade - Esquema B.

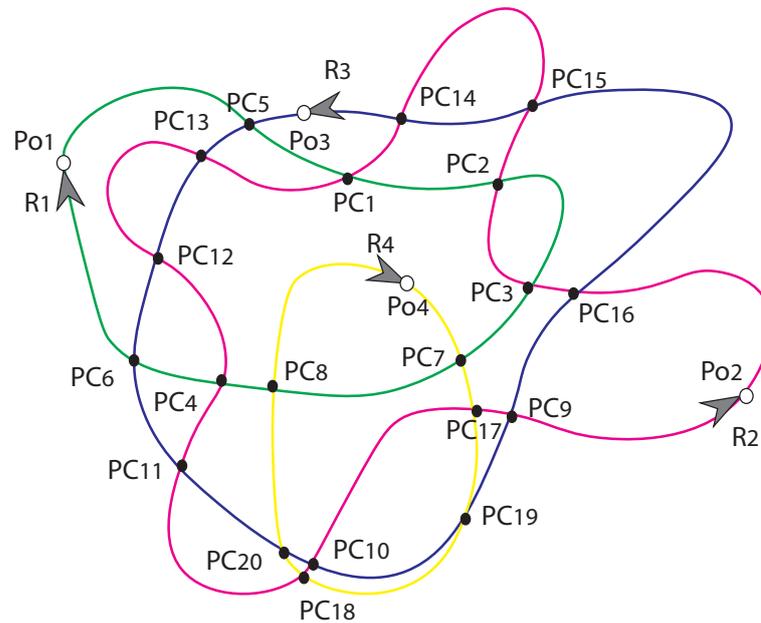
	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>
$V_{min}$ (m/s)	3	7	4	2
$V_{max}$ (m/s)	8	12	8	6

Tabela 7: Comprimentos dos trechos - Esquema B.

<b>Robô</b>	<b>Comprimento (m)</b>														
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>R1</b>	81	48	55	73	28	77	20	37	93	-	-	-	-	-	-
<b>R2</b>	139	10	88	4	108	38	52	74	73	26	88	34	59	14	83
<b>R3</b>	15	55	163	54	30	69	13	42	49	38	46	28	46	-	-
<b>R4</b>	11	31	77	11	72	134	20	-	-	-	-	-	-	-	-

Da mesma forma que foi feito para o caso A, serão avaliados os resultados oriundos do processo de otimização para o caso com e sem a inclusão de restrições de rampa.

Figura 18: Esquema de teste com quatro robôs.



Fonte: Elaborada pelo autor

Considerando o caso com as restrições de rampa, as velocidades médias obtidas para cada trecho, bem como o tempo, podem ser observados pela Tabela 8. A variação percentual máxima de velocidade entre trechos observada nesse caso foi de 29,56%.

Tabela 8: Média de velocidade e tempo em cada trecho - Esquema B com restrições de rampa.

Trecho	Velocidade (m/s)				Tempo (s)			
	R1	R2	R3	R4	R1	R2	R3	R4
1	5,96	8,13	5,55	5,00	13,6	17,1	2,7	2,2
2	5,07	9,84	6,51	3,52	9,5	1,0	8,5	8,8
3	4,74	11,98	7,32	2,51	11,6	7,3	22,3	30,7
4	3,98	11,55	5,49	3,25	18,3	0,3	9,8	3,4
5	3,54	10,05	6,08	4,08	7,9	10,7	4,9	17,6
6	4,33	9,32	6,34	4,07	17,8	4,1	10,9	32,9
7	5,20	8,80	6,68	4,62	3,8	5,9	1,9	4,3
8	6,34	8,42	7,98	-	5,8	8,8	5,3	-
9	7,99	8,75	7,98	-	11,6	8,3	6,1	-
10	-	8,72	6,98	-	-	3,0	5,4	-
11	-	8,11	6,65	-	-	10,9	6,9	-
12	-	8,63	5,23	-	-	3,9	5,4	-
13	-	8,80	4,67	-	-	6,7	9,8	-
14	-	8,77	-	-	-	1,6	-	-
15	-	8,10	-	-	-	10,3	-	-

A Tabela 9 distribui as diferenças de tempo em que os robôs envolvidos atingem

cada ponto de colisão.

Tabela 9: Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema B.

Ponto de Colisão	Robôs		Diferença de tempo (s)	Ponto de Colisão	Robôs		Diferença de tempo (s)
<b>PC1</b>	R1	R2	-40,62	<b>PC11</b>	R2	R3	-29,75
<b>PC2</b>	R1	R2	-46,79	<b>PC12</b>	R2	R3	-31,35
<b>PC3</b>	R1	R2	-35,15	<b>PC13</b>	R2	R3	-29,48
<b>PC4</b>	R1	R2	41,89	<b>PC14</b>	R2	R3	63,95
<b>PC5</b>	R1	R3	-76,57	<b>PC15</b>	R2	R3	66,35
<b>PC6</b>	R1	R3	15,92	<b>PC16</b>	R2	R3	56,32
<b>PC7</b>	R1	R4	-34,76	<b>PC17</b>	R2	R4	15,92
<b>PC8</b>	R1	R4	15,92	<b>PC18</b>	R2	R4	37,20
<b>PC9</b>	R2	R3	-26,16	<b>PC19</b>	R3	R4	37,20
<b>PC10</b>	R2	R3	33,63	<b>PC20</b>	R3	R4	15,92

A variação de tempo mínima em que os robôs chegam a um mesmo ponto de colisão foi de 15,92s, impedindo possíveis colisões nos pontos de interseção dos caminhos. No segmento de colisão compreendido entre os pontos  $PC_{19}$  e  $PC_{20}$  não foi detectada colisão entre os robôs  $R_3$  e  $R_4$ . Isso se deve ao fato de que as diferenças de tempos nos pontos mencionados são de mesmos sinais, significando que não houve o cruzamento dos robôs no segmento de caminho comum.

A Tabela 10 apresenta as velocidades médias observadas para cada trecho, caso as restrições de velocidade de rampa não sejam aplicadas.

Os resultados apresentam velocidades com variações maiores a cada trecho, sendo que a variação máxima chega a 143,4% para o robô  $R_4$  na alteração de velocidade do trecho 3 para o 5. A variação de tempo mínima de chegada a um ponto de colisão, nesse caso, foi de 16,40s.

Para os dois casos, com restrições de rampa e sem, no segmento de colisão compreendido entre os pontos  $PC_{19}$  e  $PC_{20}$ , colisões não foram detectadas, sendo suficiente apenas uma iteração para encontrar uma trajetória que não envolva acidentes.

#### 4.1.3 Sistema com cinco robôs

O sistema é composto por cinco robôs e dezesseis pontos de colisão, como pode ser observado pela Figura 19.

Nesse esquema, três segmentos de colisão são determinados: entre os pontos  $PC_2$  e  $PC_3$ , para os robôs  $R_1$  e  $R_2$ ; entre os pontos  $PC_6$  e  $PC_7$ , para os robôs  $R_2$  e  $R_3$ ; entre os pontos  $PC_{10}$  e  $PC_{11}$ , para os robôs  $R_2$  e  $R_5$ . A sequência de pontos a serem seguidos pelos

Tabela 10: Média de velocidade e tempo em cada trecho - Esquema B sem restrições de rampa.

Trecho	Velocidade (m/s)				Tempo (s)			
	R1	R2	R3	R4	R1	R2	R3	R4
<b>1</b>	5,63	8,12	6,07	5,96	14,4	17,1	2,5	1,8
<b>2</b>	4,85	8,91	6,19	3,43	9,9	1,1	8,9	9,0
<b>3</b>	5,05	11,98	7,06	2,45	10,9	7,3	23,1	31,4
<b>4</b>	4,92	11,60	6,77	5,96	14,8	0,3	8,0	1,8
<b>5</b>	5,20	9,07	5,93	4,75	5,4	11,9	5,1	15,2
<b>6</b>	3,79	8,78	6,39	3,91	20,3	4,3	10,8	34,3
<b>7</b>	4,08	8,81	5,65	3,13	4,9	5,9	2,3	6,4
<b>8</b>	4,79	8,58	7,98	-	7,7	8,6	5,3	-
<b>9</b>	7,99	8,83	7,98	-	11,6	8,3	6,1	-
<b>10</b>	-	8,81	5,54	-	-	3,0	6,9	-
<b>11</b>	-	8,58	5,46	-	-	10,3	8,4	-
<b>12</b>	-	8,78	5,25	-	-	3,9	5,3	-
<b>13</b>	-	8,82	6,21	-	-	6,7	7,4	-
<b>14</b>	-	8,83	-	-	-	1,6	-	-
<b>15</b>	-	8,58	-	-	-	9,7	-	-

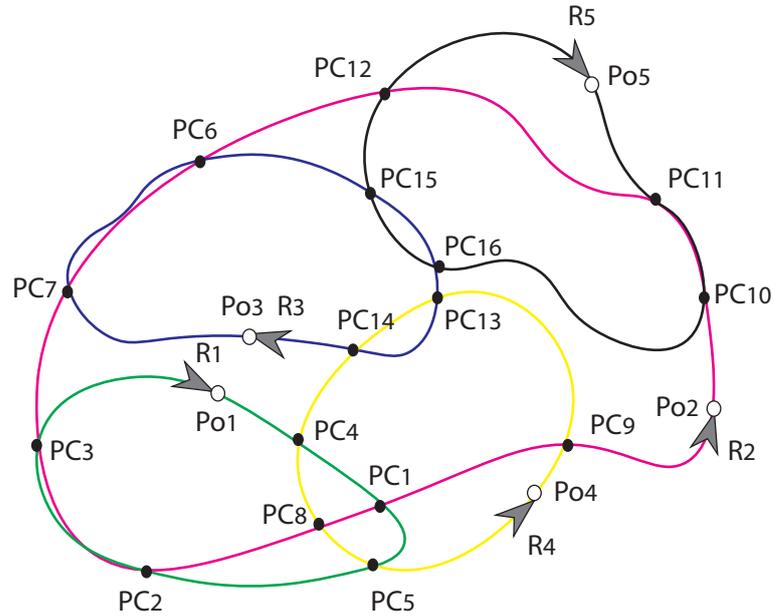
robôs juntamente com a indicação dos trechos delimitados entre os pontos são descritas abaixo:

- Robô  $R_1$  (curva verde):  $P_{o1} \xrightarrow{1} PC_4 \xrightarrow{2} PC_1 \xrightarrow{3} PC_5 \xrightarrow{4} PC_2 \xrightarrow{5} PC_3 \xrightarrow{6} P_{o1}$
- Robô  $R_2$  (curva rosa):  $P_{o2} \xrightarrow{1} PC_{10} \xrightarrow{2} PC_{11} \xrightarrow{3} PC_{12} \xrightarrow{4} PC_6 \xrightarrow{5} PC_7 \xrightarrow{6} PC_3 \xrightarrow{7} PC_2 \xrightarrow{8} PC_8 \xrightarrow{9} PC_1 \xrightarrow{10} PC_9 \xrightarrow{11} P_{o2}$
- Robô  $R_3$  (curva azul):  $P_{o3} \xrightarrow{1} PC_7 \xrightarrow{2} PC_6 \xrightarrow{3} PC_{15} \xrightarrow{4} PC_{16} \xrightarrow{5} PC_{13} \xrightarrow{6} PC_{14} \xrightarrow{7} P_{o3}$
- Robô  $R_4$  (curva amarela):  $P_{o4} \xrightarrow{1} PC_5 \xrightarrow{2} PC_8 \xrightarrow{3} PC_4 \xrightarrow{4} PC_{14} \xrightarrow{5} PC_{13} \xrightarrow{6} PC_9 \xrightarrow{7} P_{o4}$
- Robô  $R_5$  (curva preta):  $P_{o5} \xrightarrow{1} PC_{11} \xrightarrow{2} PC_{10} \xrightarrow{3} PC_{16} \xrightarrow{4} PC_{15} \xrightarrow{5} PC_{12} \xrightarrow{6} P_{o5}$

Essas sequências de pontos a serem seguidos determinam os trechos pelos quais cada robô deve completar. Os limites de velocidade são apresentados pela Tabela 11. E os comprimentos dos trechos estão dispostos na Tabela 12.

Através do processo de resolução proposto, foram detectadas possíveis colisões no segmento entre os pontos  $PC_2$  e  $PC_3$  e os pontos  $PC_6$  e  $PC_7$ , tanto para o método com

Figura 19: Esquema de teste com cinco robôs.



Fonte: Elaborada pelo autor

Tabela 11: Limites de velocidade - Esquema C.

	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>	<b>R5</b>
$V_{min}$ (m/s)	2	3	2	2	2
$V_{max}$ (m/s)	6	8	6	6	6

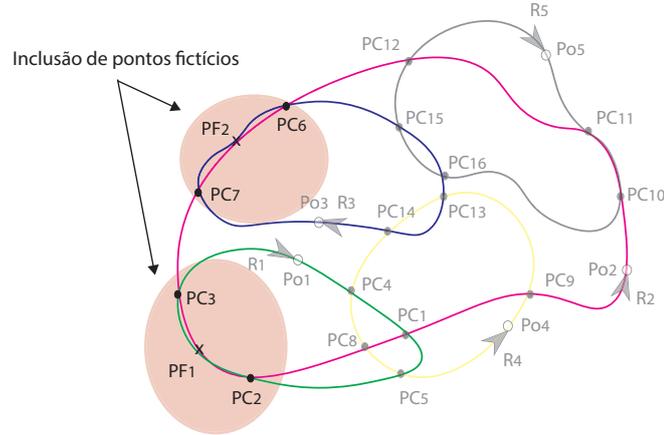
Tabela 12: Comprimentos dos trechos - Esquema C.

Robô	Comprimento (m)										
	1	2	3	4	5	6	7	8	9	10	11
<b>R1</b>	47	31	27	83	69	51	-	-	-	-	-
<b>R2</b>	53	30	105	60	55	63	66	65	23	66	58
<b>R3</b>	76	62	52	44	9	41	37	-	-	-	-
<b>R4</b>	62	19	34	39	31	80	19	-	-	-	-
<b>R5</b>	62	31	111	48	33	66	-	-	-	-	-

rampa, quanto para sem rampa, na primeira iteração do processo, sendo necessária a inclusão de pontos fictícios. No outro segmento (entre os pontos  $PC_{10}$  e  $PC_{11}$ ), a trajetória encontrada não determina colisões. Dessa forma, o algoritmo chegou à solução ótima em 2 iterações, incluindo mais dois pontos fictícios no segmento, como pode ser observado pela Figura 20, que apresenta o esquema C com destaque para inclusão dos novos pontos nos caminhos dos robôs  $R_1$  e  $R_2$ .

Através da inclusão dos pontos fictícios, dois novos trechos foram introduzidos nos caminhos dos robôs envolvidos nas colisões detectada, para fins de formulação. Esses trechos estão compreendidos nos segmentos de colisão em que foram detectadas colisões. A

Figura 20: Detalhe da inclusão dos pontos fictícios no Esquema C.



Fonte: Elaborada pelo autor

nova sequência de pontos a serem seguidos pelos robôs envolvidos no processo de inclusão de pontos fictícios juntamente com a indicação dos trechos delimitados entre os pontos são descritas abaixo:

- Robô  $R_1$  (curva verde):  $P_{o1} \xrightarrow{1} PC_4 \xrightarrow{2} PC_1 \xrightarrow{3} PC_5 \xrightarrow{4} PC_2 \xrightarrow{5} PF_1 \xrightarrow{6} PC_3 \xrightarrow{7} P_{o1}$
- Robô  $R_2$  (curva rosa):  $P_{o2} \xrightarrow{1} PC_{10} \xrightarrow{2} PC_{11} \xrightarrow{3} PC_{12} \xrightarrow{4} PC_6 \xrightarrow{5} PF_2 \xrightarrow{6} PC_7 \xrightarrow{7} PC_3 \xrightarrow{8} PF_1 \xrightarrow{9} PC_2 \xrightarrow{10} PC_8 \xrightarrow{11} PC_1 \xrightarrow{12} PC_9 \xrightarrow{13} P_{o2}$
- Robô  $R_3$  (curva azul):  $P_{o3} \xrightarrow{1} PC_7 \xrightarrow{2} PF_2 \xrightarrow{3} PC_6 \xrightarrow{4} PC_{15} \xrightarrow{5} PC_{16} \xrightarrow{6} PC_{13} \xrightarrow{7} PC_{14} \xrightarrow{8} P_{o3}$

Dessa forma, deve-se atualizar a matriz de trechos, representada pelos comprimentos em cada trechos. Os novos comprimentos podem ser observados na Tabela 13.

Tabela 13: Comprimentos dos trechos após inclusão dos pontos fictícios - Esquema C.

Robô	Comprimento (m)												
	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>R1</b>	47	31	27	83	35	34	51	-	-	-	-	-	-
<b>R2</b>	53	30	105	60	28	27	63	33	33	65	23	66	58
<b>R3</b>	76	31	31	52	44	9	41	37	-	-	-	-	-
<b>R4</b>	62	19	34	39	31	80	19	-	-	-	-	-	-
<b>R5</b>	62	31	111	48	33	66	-	-	-	-	-	-	-

Considerando os novos trechos no problema, o método de otimização foi capaz de encontrar um caminho sem colisões já na segunda iteração do processo. As velocidades

médias e os tempos obtidos para cada trecho, podem ser observados pela Tabela 14. A variação percentual máxima de velocidade entre trechos observada nesse caso foi de 29,90%.

Tabela 14: Média de velocidade e tempo em cada trecho - Esquema C com restrições de rampa.

Trecho	Velocidade (m/s)					Tempo (s)				
	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
1	3,19	5,05	4,72	3,78	2,52	14,72	10,49	16,11	16,42	24,58
2	2,24	6,35	5,48	3,05	3,21	13,82	4,73	5,65	6,23	9,66
3	2,12	6,94	3,85	2,46	4,05	12,76	15,14	8,05	13,82	27,43
4	2,74	6,81	2,70	2,27	4,45	30,30	8,82	19,25	17,20	10,79
5	3,56	7,97	2,00	2,56	4,18	9,84	3,51	21,97	12,12	7,90
6	4,61	7,97	2,17	2,77	3,36	7,37	3,39	4,14	28,88	19,64
7	4,55	7,99	2,80	3,57	-	11,20	7,88	14,64	5,32	-
8	-	7,98	3,63	-	-	-	4,14	10,18	-	-
9	-	7,98	-	-	-	-	4,14	-	-	-
10	-	7,90	-	-	-	-	8,23	-	-	-
11	-	7,59	-	-	-	-	3,03	-	-	-
12	-	5,59	-	-	-	-	11,81	-	-	-
13	-	3,94	-	-	-	-	14,71	-	-	-

A Tabela 15 contém as diferenças de tempo encontradas para todos os pontos de colisão presentes no sistema e os robôs envolvidos na possibilidade de acidente.

Tabela 15: Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema C.

Ponto de Colisão	Robôs		Diferença de tempo (s)	Ponto de Colisão	Robôs		Diferença de tempo (s)
	R1	R2			R2	R4	
PC1	R1	R2	-44,95	PC8	R2	R4	47,80
PC2	R1	R2	9,36	PC9	R2	R4	-9,39
PF1	R1	R2	23,34	PC10	R2	R5	-23,75
PC3	R1	R2	34,84	PC11	R2	R5	-9,36
PC4	R1	R4	-21,76	PC12	R2	R5	-50,00
PC5	R1	R4	24,88	PC13	R3	R4	9,37
PC6	R2	R3	9,36	PC14	R3	R4	36,13
PF2	R2	R3	20,92	PC15	R3	R5	-23,40
PC7	R2	R3	29,96	PC16	R3	R5	9,36

As diferenças de tempo são apresentadas para cada ponto de colisão original do sistema acrescidos dos pontos fictícios. Essa solução determina uma diferença de tempo mínima em que os robôs chegam a um mesmo ponto de colisão de 9,36s, impedindo possíveis colisões nos pontos de interseção dos caminhos. A verificação de não colisão no segmento é justificada pelas diferenças de tempo nos pontos  $PC_2$ ,  $PF_1$  e  $PC_3$ , que representam o segmento comum. Nesses pontos as diferenças de tempo de chegada apresentam valores

de mesmo sinal, positivo, garantindo que o robô  $R_1$  atingiu todos de forma antecipada, evitando o encontro dos robôs no caminho comum. A mesma situação é encontrada nos pontos  $PC_6$ ,  $PF_2$  e  $PC_7$  que também apresentam valores de diferença de tempo positivos, determinando um caminho livre de colisões no segmento.

Caso não sejam aplicadas as restrições de velocidade de rampa, os resultados apresentam velocidades com variações maiores a cada trecho, sendo que a variação máxima chega a 193,44% para o robô R4 na passagem do trecho 6 ao 7. A Tabela 16 apresenta as velocidades médias observadas para cada trecho. A variação de tempo mínima de chegada a um ponto de colisão, nesse caso, foi de 15,75s.

Tabela 16: Média de velocidade e tempo em cada trecho - Esquema C sem restrições de rampa.

Trecho	Velocidade (m/s)					Tempo (s)				
	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
1	2,92	7,20	6,00	5,01	2,25	16,07	7,36	12,67	12,36	27,56
2	3,30	6,75	5,99	3,80	5,07	9,40	4,44	5,17	5,00	6,11
3	2,23	7,56	5,99	2,25	5,69	12,09	13,88	5,17	15,11	19,49
4	2,07	4,58	2,04	2,22	3,97	40,00	13,09	25,46	17,59	12,09
5	5,43	7,99	2,04	4,10	5,07	6,45	3,51	21,54	7,55	6,51
6	5,41	7,98	2,24	2,04	2,34	6,28	3,38	4,02	39,20	28,24
7	5,26	7,99	4,53	5,99	-	9,70	7,88	9,04	3,17	-
8	-	7,99	2,19	-	-	-	4,13	16,91	-	-
9	-	7,99	-	-	-	-	4,13	-	-	-
10	-	7,99	-	-	-	-	8,13	-	-	-
11	-	7,98	-	-	-	-	2,88	-	-	-
12	-	7,99	-	-	-	-	8,26	-	-	-
13	-	3,06	-	-	-	-	18,93	-	-	-

Nesse caso, sem restrições de rampa, também ocorreu a necessidade de inclusão de dois pontos fictícios no mesmo segmento, bastando três iterações para livrar o segmento de colisões entre os robôs.

#### 4.1.4 Comparações com PLIM

Em busca de um método mais robusto, foi proposta uma formulação baseada em Programação Não Linear, fato que confere ao modelo uma aproximação mais fiel às não linearidades presentes no sistema real, com relação à modelagem através de Programação Linear Inteira Mista. Nessa seção, serão comparados os resultados obtidos até o momento com a solução proveniente da otimização baseada em PLIM.

A Tabela 17 apresenta as comparações entre os dois métodos com relação à diferença de tempo mínima em que dois robôs chegam a um mesmo ponto,  $\Delta t$ , e à variação de

velocidade máxima entre trechos,  $\Delta v$ , observada para cada esquema. Foram comparados os resultados obtidos através de PLIM e PNL com e sem inclusão de restrições de rampa.

Tabela 17: Comparações entre resultados de PLIM e PNL - robôs móveis.

Parâmetro	Esquema	PLIM	PNL	
			Sem Restrições de rampa	Com Restrições de rampa
$\Delta t$ (s)	A	30,0	30,9	25,9
	B	16,4	16,4	15,9
	C	15,8	15,8	9,4
$\Delta v$ (%)	A	200,0	90,9	30,0
	B	200,0	143,4	29,6
	C	200,0	193,4	29,9

Com relação à diferença de tempo mínima, a metodologia de PLIM consegue determinar um resultado satisfatório. Porém, no Esquema A, a diferença apresentou um valor menor ao obtido por PNL sem restrições de rampa. A solução baseada em PNL chega a um resultado similar ao do método comparado, porém com a vantagem de apresentar variações de velocidades entre os trechos menores, mesmo na formulação sem a inclusão de restrições de rampa. PLIM não consegue suavizar o perfil de velocidade podendo apresentar velocidades muito discrepantes ao longo do tempo, inclusive, variando entre o limite máximo e mínimo de um trecho para outro, criando momentos de aceleração ou desaceleração elevados.

Já para o método de PNL com restrições de rampa,  $\Delta t$  apresenta um valor menor, com uma redução de cerca de 16,2% no Esquema A, 3,0% no Esquema B e 40,5% no Esquema C. Essa diminuição no tempo traz uma maior proximidade entre os robôs na passagem nos pontos de colisão, fato que pode não ser desejado. Apesar do inconveniente de passagens com maior proximidade entre os veículos, a variação de velocidade máxima em cada trecho está percentualmente dentro dos limites estabelecidos em todos os esquemas, elevando a possibilidade de implementar e viabilizar o método em sistemas reais.

Pode-se chegar a conclusão que assim como PLIM, com relação à diferença de tempo mínima em que os robôs chegam a um ponto de colisão, a modelagem através de PNL também encontra um perfil de velocidade adequado que evita possíveis colisões nos caminhos dos robôs. Porém, ao contrário de PLIM, a formulação PNL permite a inclusão de restrições não lineares, sendo esse um fato importante para a utilização dessa metodologia.

Outras comparações relacionadas à formulação dos métodos são apresentadas pela Tabela 18. São dispostos os número de variáveis, variáveis inteiras e restrições envolvidas no processo de resolução.

Por formular o problema através de PLIM, variáveis inteiras e restrições são

Tabela 18: Comparações entre formulações PLIM e PNL - robôs móveis.

Parâmetro	Esquema	PLIM	PNL	
			Sem Restrições de rampa	Com Restrições de rampa
Número de variáveis	A	30	21	21
	B	81	60	60
	C	84	65	65
Número de variáveis inteiras	A	8	0	0
	B	20	0	0
	C	18	0	0
Número de restrições	A	19	3	41
	B	44	4	92
	C	41	5	87

utilizadas para representar possíveis colisões nos pontos críticos. Assim, o método apresenta um número de variáveis e restrições elevado comparado ao método PNL. A complexidade de modelagem do método PLIM conta como uma desvantagem, pois resulta num mesmo intervalo de tempo da formulação PNL, porém lançando mão de mais restrições e variáveis, inclusive binárias. Em termos percentuais verifica-se uma redução mínima do número de variáveis de PLIM de cerca de 22,6% no Esquema C e uma redução mínima de número de restrições de 84,2% no Esquema A em relação ao método PLN sem rampa. Apesar do método com limitações de rampa apresentar um número de restrições elevado, tais restrições conferem ao sistema um perfil de velocidade suave, sendo altamente necessário para a viabilizar a aplicabilidade prática.

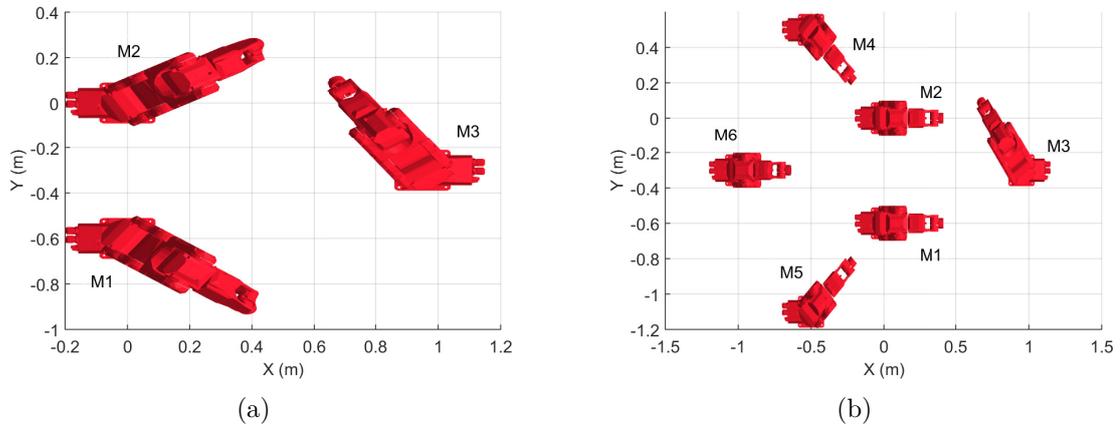
Dessa forma, a formulação proposta apresenta resultados similares, porém com maior robustez, podendo englobar outras situações que necessitam de uma modelagem não linear.

## 4.2 SISTEMAS DE MÚLTIPLOS MANIPULADORES ROBÓTICOS

Os teste foram realizados com dois tipos de esquemas diferentes. Os sistemas considerados serão os seguintes:

- D** : Sistema composto por três robôs com um ponto comum em seus caminhos, determinando três possibilidades de colisões. Sua configuração pode ser observada pela Figura 21.a;
- E** : Sistema composto por cinco robôs com cinco pontos comuns em seus caminhos, determinando sete possibilidades de colisões. Sua configuração pode ser observada pela Figura 21.b;

Figura 21: Esquemas de sistemas de múltiplos manipuladores: (a) 3 robôs, (b) 6 robôs.



Fonte: Elaborada pelo autor

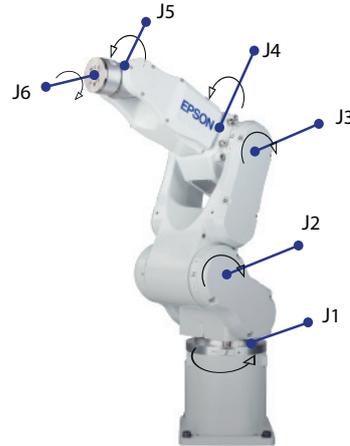
A Figura 21 é uma ilustração da visão superior dos manipuladores, sendo que eles estão dispostos na configuração inicial dos seus movimentos. Tais configurações representam as angulações das juntas de cada manipulador no ponto de partida de sua trajetória. E as configurações de passagem representam as angulações das juntas que devem ser obedecidas ao longo de suas trajetórias. Nessas configurações os robôs realizam tarefas que não são consideradas no presente trabalho. Os pontos de colisões são pontos em que eles tem a possibilidade de se colidir, pois são pontos comuns aos caminhos determinados.

Para validar o método aplicado à coordenação de robôs manipuladores, foi utilizado para as simulações um robô com seis graus de liberdade com somente juntas rotacionais, o Epson PROSIX C3-A601C [33], que pode ser observado pela Figura 22, com a especificação de cada junta  $J_g$  e suas direções. Para que seja viável para essa simulação, o robô deve ter a possibilidade de ser controlado diretamente pelas juntas, já que o resultado do problema de otimização é o perfil de velocidade angular de cada junta. O robô escolhido atende a essa exigência.

Os limites de velocidade angular para cada junta do robô devem ser considerados para fins de simulação, já que o objetivo é determinar a velocidade ao longo do tempo, sendo que essa não deve ultrapassar limites estabelecidos pelo fabricante. Na Tabela 19 encontram-se tais limitações [33]. Como os pontos a serem seguidos são preterminados para fins do processo de otimização não são necessárias as limitações de angulação, sendo necessário apenas, obedecê-las na escolha dos pontos a serem percorridos pelo efetuator. Outros dados do braço robótico podem ser obtidos em seu manual [33].

A representação do robô foi feita através da *toolbox* ARTE (*A Robotic Toolbox for Education*) desenvolvida por Gil A. et al. [34] para o MATLAB. Essa ferramenta consegue modelar uma diversidade de manipuladores comerciais, apresentando gráficos

Figura 22: Manipulador Epson PROSIX C3-A601C.



Fonte: Adaptado de [33]

Tabela 19: Limites de velocidade angular para cada junta do manipulador.

Junta	J1	J2	J3	J4	J5	J6
$ \omega_{max} $ (rad/s)	7,85	7,85	8,97	9,65	9,65	12,57

3D que possibilitam a observação dos movimentos do robô robô simulador. Ela permite, também, traçar e observar a posição, velocidade, aceleração e torque das coordenadas das articulações do robô quando um movimento é realizado.

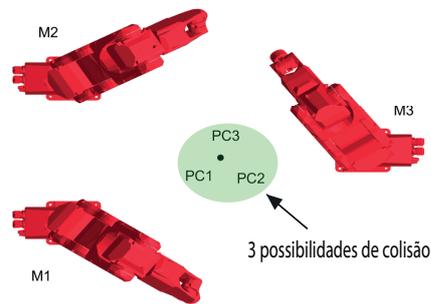
Todos os robôs possuem configurações angulares de passagem que devem cumprir, sendo consideradas o caminho a ser percorrido por eles. Deve-se convencionar que a colisão somente tem a possibilidade de ocorrer no ponto comum, com a configuração estabelecida.

#### 4.2.1 Sistema com três manipuladores

Nessa situação três robôs devem percorrer configurações predeterminadas, sendo que existe a possibilidade de se colidirem num ponto comum de seus caminhos. A colisão pode existir entre  $M_1$  e  $M_2$  ( $PC_1$ ),  $M_1$  e  $M_3$  ( $PC_2$ ) e  $M_2$  e  $M_3$  ( $PC_3$ ), configurando três possibilidades de colisões diferentes, conforme ilustrado pela Figura 23. A formulação, portanto, vai considerar essas possibilidades e maximizar a diferença de tempo em que um par de robôs chega ao ponto de colisão.

As configurações angulares das juntas relacionadas ao pontos em que os robôs devem seguir são dispostas pela Tabela 20. Os robôs devem completar seus movimentos, ou seja, cumprir as configurações das articulações determinadas em cada trecho, em 1.7s.

Figura 23: Esquema de teste com três manipuladores.



Fonte: Elaborada pelo autor

Tabela 20: Configurações angulares de passagem - Esquema D.

Robô	Ponto	Configuração angular (rad)					
		J1	J2	J3	J4	J5	J6
M1	1	0,52	0,61	0,10	0,52	-0,75	-0,06
	2	0,00	0,00	0,00	0,00	0,00	0,00
	3	-0,52	0,61	0,10	-0,52	-0,75	0,06
	4	0,00	0,00	0,00	0,00	0,00	0,00
	5	0,52	0,61	0,10	0,52	-0,75	-0,06
M2	1	-0,62	0,60	0,16	-0,62	-0,78	0,02
	2	0,00	0,00	0,00	0,00	0,00	0,00
	3	0,52	0,61	0,10	0,52	-0,75	-0,06
	4	0,00	0,00	0,00	0,00	0,00	0,00
	5	-0,62	0,60	0,16	-0,62	-0,78	0,02
M3	1	0,96	0,56	-0,06	0,00	-0,10	0,00
	2	0,42	0,00	-0,06	0,00	0,00	0,00
	3	0,00	0,67	0,21	0,00	0,00	0,00
	4	0,42	0,00	-0,06	0,00	0,00	0,00
	5	0,96	0,56	-0,06	0,00	-0,10	0,00

A Tabela 21 apresenta as médias de velocidade de cada junta dos robôs e o tempo que eles gastaram em cada trecho.

Tabela 21: Média de velocidade das articulações e tempo em cada trecho - Esquema D.

Robôs	Trecho	Tempo (s)	Velocidade angular (rad/s)					
			J1	J2	J3	J4	J5	J6
M1	1	0,15	-0,75	-0,88	-0,15	-0,76	1,08	0,09
	2	0,16	-0,74	0,87	0,15	-0,75	-1,07	0,09
	3	0,70	3,25	-3,81	-0,64	3,26	4,67	-0,40
	4	0,70	3,25	3,81	0,64	3,26	-4,67	-0,40
M2	1	0,70	3,89	-3,77	-1,00	3,89	4,87	-0,12
	2	0,70	3,25	3,81	0,64	3,26	-4,67	-0,40
	3	0,15	-0,75	-0,88	-0,15	-0,76	1,08	0,09
	4	0,15	-0,89	0,86	0,23	-0,89	-1,11	0,03
M3	1	0,43	-1,28	-1,35	0,00	0,00	0,24	0,00
	2	0,43	-0,95	1,52	0,61	0,00	0,00	0,00
	3	0,42	0,95	-1,52	-0,61	0,00	0,00	0,00
	4	0,42	1,28	1,35	0,00	0,00	-0,24	0,00

O gráfico de velocidade angular de cada junta para cada robô ao longo do tempo é ilustrado pela Figura 24, com as indicação do momento em que os robôs passarão pelos pontos de colisão. Nos instantes em que a velocidade angular é igual a zero, o robô está em uma configuração de passagem na qual ele deve parar seus movimentos. O resultado encontrado apresenta um perfil de velocidade adequado que previne colisões no ponto de colisão existente no sistema. As velocidades das articulações não ultrapassaram os limites impostos pela Tabela 19.

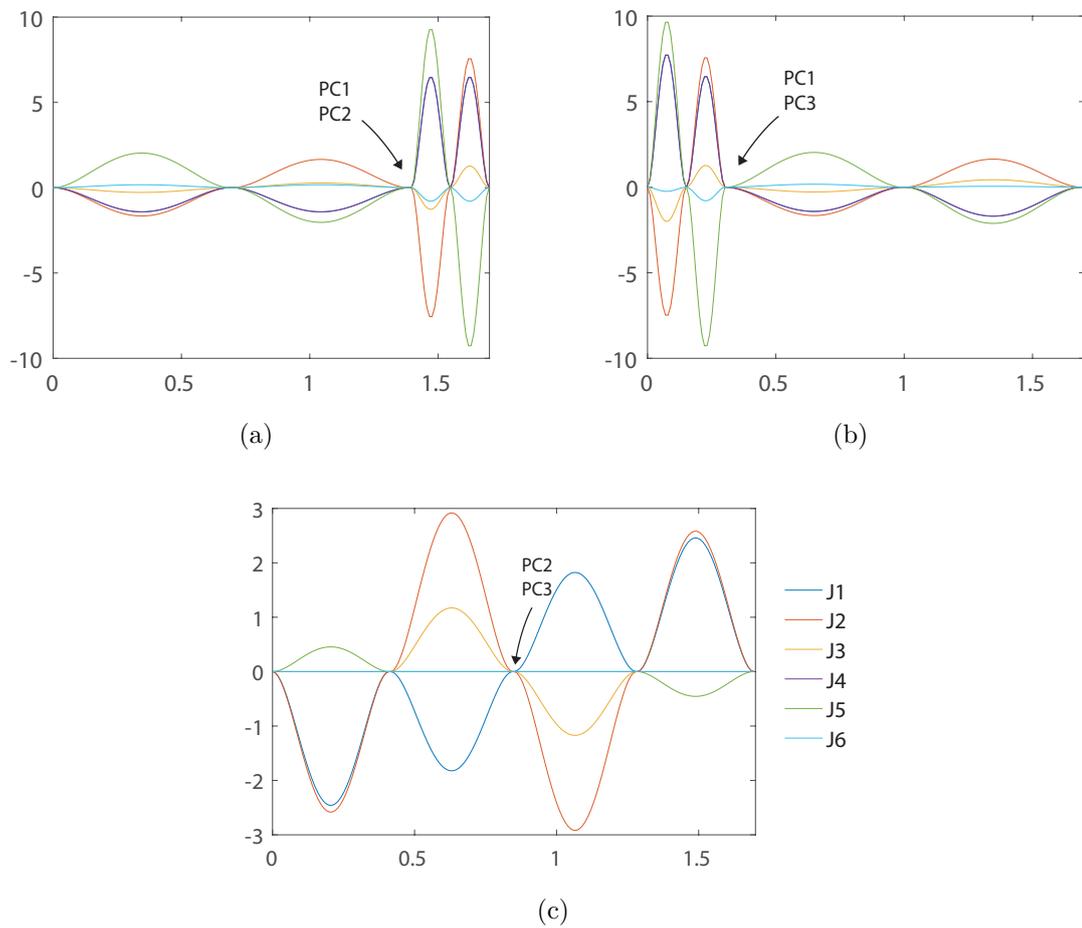
Como há apenas um ponto comum, o resultado apresenta tempos de chegada a ele, para cada robô, distintos, evitando a colisão. A diferença de tempo mínima,  $\Delta t$ , em que um par de robôs chega ao ponto de colisão é de  $0.55s$ . A Tabela 22 apresenta as diferenças de tempo de chegada dos robôs associados aos pontos de colisão.

Tabela 22: Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema D.

Ponto de Colisão	Robôs		Diferença de Tempo (s)
PC1	M1	M2	-0,55
PC2	M1	M3	0,55
PC3	M2	M3	1,09

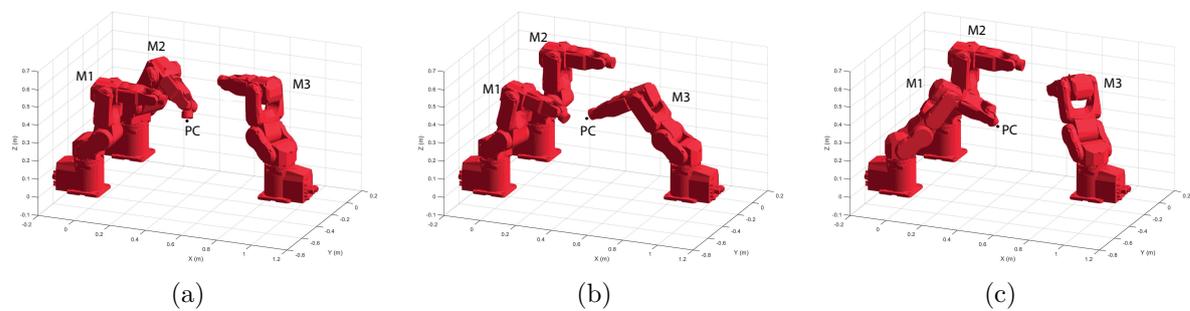
A Figura 25 apresenta a ilustração da chegada de cada robô ao ponto de colisão enquanto os outros se encontram em outros locais distintos de suas trajetórias.

Figura 24: Velocidade angular (rad/s) das articulações ao longo tempo (s) para cada robô - Esquema D: (a)  $M_1$ , (b)  $M_2$ , (c)  $M_3$ .



Fonte: Elaborada pelo autor

Figura 25: Passagem de cada robô pelo ponto de colisão: (a)  $M_2$  em  $t=0,31s$ , (b)  $M_3$  em  $t=0,86s$ , (c)  $M_1$  em  $t=1,40s$ .

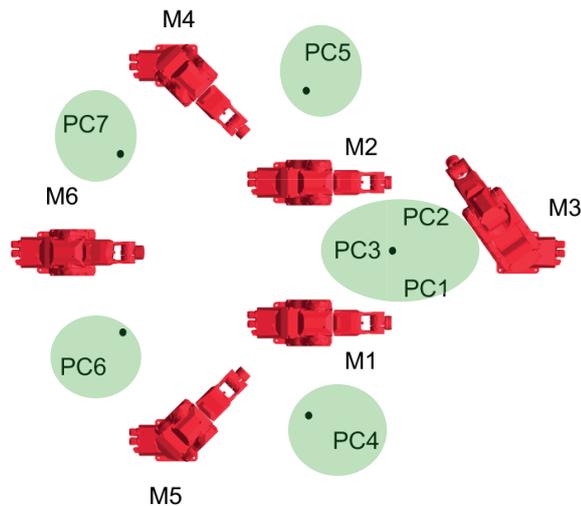


Fonte: Elaborada pelo autor

#### 4.2.2 Sistema com seis manipuladores

O sistema conta com seis robôs e cinco pontos comuns em seus caminhos. Porém a possibilidade de colisão existe em sete momentos diferentes, como pode ser observado pela Figura 26, que apresenta a visão superior do sistema com indicação da localização dos pontos de colisão.

Figura 26: Esquema de teste com seis manipuladores.



Fonte: Elaborada pelo autor

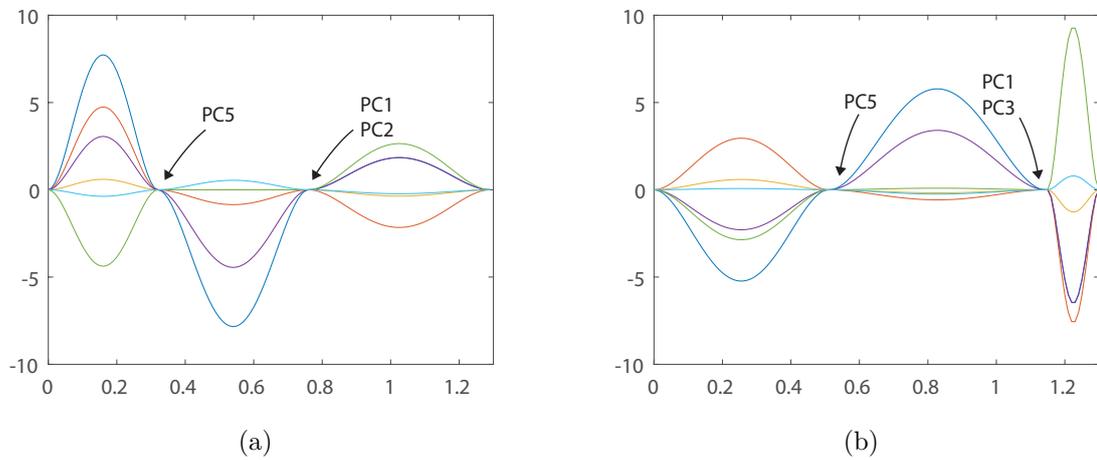
No caso, os robôs  $M_1$ ,  $M_2$  e  $M_3$  compartilham um mesmo ponto em seus caminhos. Dessa forma, a colisão pode ocorrer em três momentos diferentes nesse ponto. Além disso existem outros quatro pontos comuns nos caminhos dos manipuladores.

As configurações angulares das juntas relacionadas ao pontos em que os robôs devem seguir são dispostas pela Tabela 23. Os robôs devem completar seus movimentos, ou seja, cumprir as configurações das articulações determinadas em cada trecho, em 1.3s.

Após o processo de otimização, as velocidades angulares resultantes apresentaram valores factíveis e dentro dos limites estabelecidos. O perfil de velocidade ao longo do tempo é apresentado pela Figura 27 juntamente com as indicações dos momentos em que os robôs passam pelos pontos de colisão. As velocidades não ultrapassam os valores estabelecidos como limites para as articulações.

Tabela 23: Configurações angulares de passagem - Esquema E

Robô	Ponto	Configuração angular (rad)					
		J1	J2	J3	J4	J5	J6
M1	1	0,00	0,00	0,00	0,00	0,00	0,00
	2	1,32	0,81	0,10	0,52	-0,75	-0,06
	3	-0,52	0,61	0,10	-0,52	-0,75	0,06
	4	0,00	0,00	0,00	0,00	0,00	0,00
M2	1	0,00	0,00	0,00	0,00	0,00	0,00
	2	-1,42	0,80	0,16	-0,62	-0,78	0,02
	3	0,52	0,61	0,10	0,52	-0,75	-0,06
	4	0,00	0,00	0,00	0,00	0,00	0,00
M3	1	0,96	0,56	-0,06	0,00	-0,10	0,00
	2	0,42	0,00	-0,06	0,00	0,00	0,00
	3	0,00	0,67	0,21	0,00	0,00	0,00
	4	0,42	0,00	-0,06	0,00	0,00	0,00
	5	0,96	0,56	-0,06	0,00	-0,10	0,00
M4	1	0,79	0,00	0,00	0,00	0,00	0,00
	2	1,57	0,79	0,01	0,00	0,40	0,20
	3	0,00	0,70	0,00	0,00	0,00	0,00
	4	0,79	0,00	0,00	0,00	0,00	0,00
M5	1	-0,79	0,00	0,00	0,00	0,00	0,00
	2	-1,57	0,79	0,01	0,00	0,40	0,20
	3	0,00	0,90	0,20	0,60	0,00	0,00
	4	-0,79	0,00	0,00	0,00	0,00	0,00
M6	1	0,00	-0,20	0,00	0,00	0,00	0,00
	2	0,55	0,85	0,20	0,00	-0,40	0,30
	3	-0,55	0,85	0,20	-0,20	-0,40	0,60
	4	0,00	-0,20	0,00	0,00	0,00	0,00

Figura 27: Velocidade angular (rad/s) das articulações ao longo tempo (s) para cada robô - Esquema E: (a)  $M_1$ , (b)  $M_2$ , (c)  $M_3$ , (c)  $M_4$ , (c)  $M_5$ , (c)  $M_6$ .

A Tabela 24 apresenta as médias de velocidade das articulações em cada trecho e os tempos que cada robô gastou em cada trecho.

Tabela 24: Média de velocidade das articulações e tempo em cada trecho - Esquema E.

Robôs	Trecho	Tempo (s)	Velocidade angular (rad/s)					
			J1	J2	J3	J4	J5	J6
M1	1	0,32	4,00	2,45	0,31	1,58	-2,27	-0,19
	2	0,44	-4,09	-0,44	0,00	-2,32	0,00	0,28
	3	0,54	0,96	-1,13	-0,19	0,97	1,39	-0,12
M2	1	0,51	-2,74	1,55	0,31	-1,20	-1,50	0,04
	2	0,64	3,04	-0,30	-0,09	1,79	0,05	-0,13
	3	0,15	-3,25	-3,81	-0,64	-3,26	4,67	0,40
M3	1	0,18	-2,99	-3,14	0,00	0,00	0,55	0,00
	2	0,19	-2,09	3,35	1,34	0,00	0,00	0,00
	3	0,47	0,89	-1,42	-0,57	0,00	0,00	0,00
	4	0,46	1,14	1,20	0,00	0,00	-0,21	0,00
M4	1	0,36	2,12	2,12	0,03	0,00	1,08	0,54
	2	0,55	-2,86	-0,16	-0,02	0,00	-0,73	-0,36
	3	0,39	2,01	-1,79	0,00	0,00	0,00	0,00
M5	1	0,36	-2,12	2,12	0,03	0,00	1,08	0,54
	2	0,55	2,86	0,21	0,35	1,09	-0,73	-0,36
	3	0,39	-1,96	-2,25	-0,50	-1,50	0,00	0,00
M6	1	0,43	1,28	2,44	0,47	0,00	-0,93	0,70
	2	0,44	-2,44	0,00	0,00	-0,44	0,00	0,67
	3	0,43	1,28	-2,44	-0,47	0,47	0,93	-1,40

Os perfis de velocidade encontrados asseguram que não haverá colisões nos pontos comuns devido à diferença de tempo em que eles chegam até eles. Nesse caso, a variação mínima de tempo,  $\Delta t$ , é de  $0.39s$ . A Tabela 25 apresenta a indicação dos robôs envolvidos em cada ponto de colisão e as diferenças de tempo de chegada dos robôs associados aos pontos de colisão.

Tabela 25: Diferenças de tempo de chegada dos robôs em cada ponto de colisão - Esquema E.

Ponto de Colisão	Robôs		Diferença de tempo (s)
PC1	M1	M2	0,39
PC2	M1	M3	0,39
PC3	M2	M3	0,39
PC4	M1	M5	0,78
PC5	M2	M4	0,39
PC6	M5	M6	0,87
PC7	M4	M6	0,42

### 4.2.3 Comparações com PLIM

A literatura apresenta formas para coordenar manipuladores robóticos. No entanto, a doutrina majoritária trabalha com Programação Linear Inteira Mista para resolver esse tipo de situação, assim como em robôs móveis. De forma a encontrar um método que possa cobrir restrições e função objetivos não lineares, foi proposta uma formulação baseada em PNL. A Tabela 26 apresenta uma comparação entre os dois métodos. Os parâmetros avaliados são a diferença de tempo mínima de chegada em um ponto de colisão,  $\Delta t$ , o número de variáveis, restrições e variáveis inteiras.

Tabela 26: Comparação entre PNL e PLIM - manipuladores robóticos

Parâmetro	Esquema	PLIM	PNL
$\Delta t$ (s)	D	0,55	0,55
	E	0,39	0,39
Número de variáveis	D	16	12
	E	27	19
Número de variáveis inteiras	D	3	0
	E	7	0
Número de restrições	D	6	3
	E	20	6

Como pode ser percebido, pelos dados resultante de  $\Delta t$ , as soluções obtidas pelas duas formulações são equivalentes, permitindo chegar em uma mesma diferença de tempo mínima de chegada a um ponto de colisão. A vantagem do método de PNL consiste em uma menor complexidade na formulação, chegando à solução ótima com menor número de variáveis, cerca de 25,0% para o Esquema D e 29,6% para o Esquema E. As restrições também apresentam uma redução de 50,0% para o Esquema D e de 70,0% para o Esquema E. Além disso, a otimização não linear permite a inclusão de restrições relacionadas a aceleração e velocidade ao longo do tempo, sendo portanto, uma metodologia mais robusta.

### 4.3 CONSIDERAÇÕES SOBRE CONVEXIDADE

O conjunto de soluções viáveis para todas as situações apresentadas é não convexo, apresentando vários ótimos locais. Dessa forma, a otimização através de formulação Não Linear, pode encontrar diferentes soluções de acordo com a solução inicial apresentada ao solucionador, no caso, a toolbox *fmincon* do software MATLAB.

Como o algoritmo proposto pelo solucionador não é capaz de fazer uma distinção entre soluções ótimas locais e a solução ótima global, é necessário determinar um ponto inicial adequado que leve o algoritmo a uma solução ótima global. Destacando que a solução do problema de otimização são os tempos em cada trecho, a forma encontrada para determinar a variação de tempo ótima em que os robôs chegam a um determinado ponto de colisão, foi considerar, como soluções iniciais, os tempos relacionados à velocidade média,

que é calculada através dos limites previamente determinados e do comprimento total do caminho ou a angulação total das articulações que o manipulador deverá se mover. No caso dos robôs móveis, o resultado para o caso sem restrições de rampa, é uma solução com uma variação de velocidade entre trechos menor, em comparação com o método de PLIM. E para o caso com tais restrições, essa solução inicial fica mais aproximada da solução ótima. Portanto, determinando o ponto inicial dessa forma, consegue-se um conjunto de soluções próximas convexo, levando a uma maior confiabilidade do método.

## 5 CONSIDERAÇÕES FINAIS

### 5.1 CONCLUSÃO

O presente trabalho apresentou uma abordagem baseada em Otimização Não Linear com a finalidade de resolver situações que envolvem sistemas com múltiplos robôs que possuem espaços de trabalho compartilhados. No caso, o objetivo foi encontrar um perfil de velocidade ótimo que evitasse colisões em seus caminhos previamente especificados. Esse tipo de problema pode ser observado tanto no caso de robôs móveis, para atividades como inspeção e logística, quanto em manipuladores robóticos que exercem algum tipo de atividade que devem ter seus movimentos sincronizados, como em soldagem de pontos determinados ou pintura de peças.

A literatura especializada apresenta várias formas de solucionar problemas similares, sendo que a maior parte utiliza Programação Linear Inteira Mista. Indo de encontro a essa metodologia, foi aplicada uma técnica de Programação Não Linear com o objetivo de encontrar trajetórias que não envolvam acidentes entre os dispositivos presentes no sistema. A formulação considerada possui uma função objetivo que maximiza o quadrado do menor intervalo de tempo em que dois robôs chegarão a um mesmo ponto onde existe a possibilidade de colisão. Ao contrário de outros trabalhos que tratam as colisões como restrições, a forma de evitá-las no método proposto é através da função objetivo. Já as restrições são compostas por limitações temporais e de velocidade e são específicas de acordo com as peculiaridades dos sistemas.

Sistemas envolvendo robôs móveis devem considerar limitações de velocidade ao longo do tempo, para que o perfil de velocidade encontrado não apresente discrepâncias entre os trechos de passagem. Essa situação criaria momentos em que aceleração ou desaceleração seriam muito altas, podendo ser prejudicial ao dispositivo ou até mesmo inviável, considerando limitações físicas dos robôs. Tais restrições foram formuladas de forma não linear, não permitidas, portanto, em métodos lineares.

Podem existir sistemas que envolvam caminhos muito próximos ou coincidentes. Nesse caso, deve existir algum método que evite a passagem de mais um veículo através dele em um mesmo instante de tempo. A simples aplicação do método não evita que haja colisão nesse segmento comum dos caminhos. Assim sendo, uma forma de verificação de colisão é a criação de pontos fictícios ao longo dos segmentos. Esses pontos criam barreiras para que não haja a permanência mútua de robôs no trecho, evitando colisões.

Já no caso de manipuladores robóticos as peculiaridades envolvem o perfil de velocidade a ser determinado que deve considerar limitações de velocidade angular de cada articulação do robô. Assim, o limite de tempo mínimo de passagem do manipulador pelo trecho deve obedecer essas limitações dinâmicas. A formulação proposta abordou apenas

as limitações temporais e associadas a tempo de conclusão da trajetória.

Os testes envolvendo robôs móveis foram realizados em três sistemas com diferentes configurações. Em todos os casos o objetivo foi atingido e o método conseguiu coordenar os veículos de forma a evitar acidentes ao longo de seus caminhos. Adicionalmente, o objetivo de poder limitar a variação de velocidade também foi verificado. Apesar da redução do intervalo de tempo mínimo de chegada ao ponto de colisão de até 40,5%, com relação à formulação sem a inclusão de restrições de rampa, o resultado se aproximou de uma solução viável apresentado um perfil de velocidade suave com alterações de até 30,0%. A inclusão dessas restrições são fundamentais para o perfeito funcionamento do sistema, restringindo a velocidade à limitações de aceleração e torque de robôs reais.

A inclusão de pontos fictícios também se mostrou eficiente e foi suficiente para evitar acidentes nos segmentos de colisão. Em dois dos esquemas de teste apenas os pontos das extremidades permitiram que acidentes fossem evitados. Já no esquema com cinco robôs, foram necessárias três iterações do problema, incluindo dois pontos fictícios em um dos três segmentos considerados. Essa abordagem evita a utilização de PLIM com variáveis binárias, não sendo necessária a consideração da posição de chegada dos robôs no trecho comum.

Já a coordenação de sistemas de múltiplos manipuladores foram testados em esquemas com três e seis robôs, possuindo pontos comuns em seus caminhos. As configurações de velocidades encontradas não excedem os limites estabelecidos. A forma de satisfazer essa restrição foi contemplada na metodologia através de um problema secundário de otimização que determina os limites de tempo mínimo em que o robô deve cumprir os trechos. A solução conseguiu encontrar uma variação de tempo mínima satisfatória e um perfil de velocidade que não envolva colisões.

Para ambos os casos a solução é semelhante ao método usado pela literatura baseado em PLIM. Em um dos casos, inclusive, consegue chegar a uma diferença de tempo mínima maior. A vantagem da utilização de PNL reside no fato de ser um método capaz de englobar variáveis e restrições não lineares, trazendo consigo uma maior aproximação à realidade. Além disso o resultado é obtido com menos restrições e variáveis, excluindo variáveis inteiras binárias da formulação. Para problemas de coordenação, permite incluir restrições dinâmicas como aceleração e torque, evitando possíveis extrapolações ao que o equipamento resiste.

Face ao exposto, os principais objetivos foram cumpridos, garantindo trajetórias que não envolvam acidentes. A metodologia foi capaz de tratar as colisões através da função objetivo e dentro dos limites estabelecidos, encontrar perfis ótimos de velocidade linear, no caso de móveis e angular, no caso de manipuladores. E se comparando com modelos já discutidos, se mostrou mais promissora, por sua modelagem ser compatível com limitações dinâmicas dos robôs.

## 5.2 TRABALHOS FUTUROS

O desenvolvimento do trabalho versando sobre a aplicação de técnicas de otimização ao problema de coordenação de robôs pode ser aprimorado em diversos aspectos, dentre eles, como proposta de trabalhos futuros, é possível citar:

- Solucionar o problema envolvendo, na função objetivo, a otimização da energia consumida pelos robôs presentes no sistema, obtendo uma solução viável em termos de colisões e também eficiente com relação ao consumo;
- Implementar o algoritmo de solução do problema de otimização Não Linear, possibilitando a comparação dos métodos com relação a tempo computacional;
- Considerar sistemas com um número maior de robôs, aumentando a complexidade a fim de avaliar a eficiência do método em sistemas de grande porte;
- Considerar restrições dinâmicas relacionadas a torque e aceleração em sistemas de manipuladores;
- Considerar outros tipos de manipuladores robóticos com diferentes tipos juntas, além da rotacional;
- Implementar o método em sistemas reais de forma online;
- Envolver tarefas que impactem na formulação aos robôs, como por exemplo, carregamento e descarregamento de produtos em armazéns logísticos ou custos não lineares relacionados ao processo de coordenação.

## REFERÊNCIAS

- [1] BARRAQUAND, J., LATOMBE, J.-C., “Robot motion planning: A distributed representation approach”, *The International Journal of Robotics Research*, v. 10, n. 6, pp. 628–649, 1991.
- [2] KALTSOUKALAS, K., MAKRIS, S., CHRYSOLOURIS, G., “On generating the motion of industrial robot manipulators”, *Robotics and Computer-Integrated Manufacturing*, v. 32, pp. 65–71, 2015.
- [3] SNAPE, J., VAN DEN BERG, J., GUY, S. J., MANOCHA, D., “The hybrid reciprocal velocity obstacle”, *IEEE Transactions on Robotics*, v. 27, n. 4, pp. 696–706, 2011.
- [4] GONÇALVES, V. M., PIMENTA, L. C., MAIA, C. A., PEREIRA, G. A., “Coordination of multiple fixed-wing UAVs traversing intersecting periodic paths”. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 849–854, 2013.
- [5] AKELLA, S., HUTCHINSON, S., “Coordinating the motions of multiple robots with specified trajectories”. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, v. 1, pp. 624–631, 2002.
- [6] LIMA, P. U., CUSTODIO, L. M., “Multi-robot systems”, In: *Innovations in robot mobility and control*, pp. 1–64, Springer, 2005.
- [7] JOSE, K., PRATIHAR, D. K., “Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods”, *Robotics and Autonomous Systems*, v. 80, pp. 34–42, 2016.
- [8] ŠVESTKA, P., OVERMARS, M. H., “Coordinated path planning for multiple robots”, *Robotics and Autonomous Systems*, v. 23, n. 3, pp. 125–152, apr 1998.
- [9] MIYATA, N., OTA, J., ARAI, T., ASAMA, H., “Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment”, *IEEE transactions on robotics and automation*, v. 18, n. 5, pp. 769–780, 2002.
- [10] PEASGOOD, M., CLARK, C. M., MCPHEE, J., “A complete and scalable strategy for coordinating multiple robots within roadmaps”, *IEEE Transactions on Robotics*, v. 24, n. 2, pp. 283–292, 2008.
- [11] SMITH, S. L., SCHWAGER, M., RUS, D., “Persistent robotic tasks: Monitoring and sweeping in changing environments”, *IEEE Transactions on Robotics*, v. 28, n. 2, pp. 410–426, 2012.
- [12] LIU, S., SUN, D., ZHU, C., “Coordinated motion planning for multiple mobile robots along designed paths with formation requirement”, *IEEE/ASME transactions on mechatronics*, v. 16, n. 6, pp. 1021–1031, 2011.
- [13] ALONSO-MORA, J., BREITENMOSER, A., BEARDSLEY, P., SIEGWART, R., “Reciprocal collision avoidance for multiple car-like robots”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 360–366, 2012.

- [14] CARPIN, S., PAGELLO, E., “An experimental study of distributed robot coordination”, *Robotics and Autonomous Systems*, v. 57, n. 2, pp. 129–133, 2009.
- [15] SOLTERO, D. E., SMITH, S. L., RUS, D., “Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3645–3652, 2011.
- [16] LEE, B. H., LEE, C. G., “Collision-free motion planning of two robots”, *IEEE Transactions on Systems, Man, and Cybernetics*, v. 17, n. 1, pp. 21–32, 1987.
- [17] BIEN, Z., LEE, J., “A minimum-time trajectory planning method for two robots”, *IEEE Transactions on Robotics and Automation*, v. 8, n. 3, pp. 414–418, 1992.
- [18] ALTCHÉ, F., QIAN, X., DE LA FORTELLE, A., “Time-optimal coordination of mobile robots along specified paths”. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 5020–5026, 2016.
- [19] PENG, J., AKELLA, S., “Coordinating multiple robots with kinodynamic constraints along specified paths”, *The International Journal of Robotics Research*, v. 24, n. 4, pp. 295–310, 2005.
- [20] SPENSIERI, D., BOHLIN, R., CARLSON, J. S., “Coordination of robot paths for cycle time minimization”. In: *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pp. 522–527, 2013.
- [21] WOEGINGER, G. J., “Exact algorithms for NP-hard problems: A survey”, In: *Combinatorial Optimization—Eureka, You Shrink!*, pp. 185–207, Springer, 2003.
- [22] ABICHANDANI, P., BENSON, H. Y., KAM, M., “Multi-vehicle path coordination under communication constraints”. In: *American Control Conference, 2008*, pp. 650–656, 2008.
- [23] YAN, C., YANG, L., “Mixed-integer programming based approaches for the movement planner problem: Model, heuristics and decomposition”, *Proc. RAS Problem Solving Competition*, pp. 1–14, 2012.
- [24] ZHAO, Y., IOANNOU, P., “Positive Train Control With Dynamic Headway Based on an Active Communication System”, *IEEE Transactions on Intelligent Transportation Systems*, v. 16, n. 6, pp. 3095–3103, 2015.
- [25] FARINELLI, A., BOSCOLO, N., ZANOTTO, E., PAGELLO, E., “Advanced approaches for multi-robot coordination in logistic scenarios”, *Robotics and Autonomous Systems (2016)*, 2016.
- [26] CRAIG, J. J., *Robótica. 3ª edição*. São Paulo: Editora Pearson, 2012.
- [27] BRADY, M., *Robot motion: Planning and control*. MIT press, 1982.
- [28] PAUL, R., *Modelling, trajectory calculation and servoing of a computer controlled arm*, Tech. rep., DTIC Document, 1972.
- [29] YOSHIKAWA, T., *Foundations of robotics: analysis and control*. MIT press, 1990.

- [30] BYRD, R. H., HRIBAR, M. E., NOCEDAL, J., “An interior point algorithm for large-scale nonlinear programming”, *SIAM Journal on Optimization*, v. 9, n. 4, pp. 877–900, 1999.
- [31] THE MATHWORKS, I., “fmincon Interior-Point Algorithm with Analytic Hessian”, .
- [32] THE MATHWORKS, I., “intlinprog Mixed-integer linear programming (MILP)”, .
- [33] Seiko Epson Corporation, *Epson Prosix 6-axis Robots*, 5 2011.
- [34] GIL, A., REINOSO, O., MARIN, J. M., PAYA, L., RUIZ, J., “Development and deployment of a new robotics toolbox for education”, *Computer Applications in Engineering Education*, v. 23, n. 3, pp. 443–454, 2015.

## APÊNDICE A – RESUMO DO MÉTODO DE PONTOS INTERIORES - BARREIRA LOGARÍTMICA

O método a ser descrito, de forma sucinta, é implementado na toolbox denominada *fmincon* do programa MATLAB, sendo que essa ferramenta é utilizada para a realização da etapa de otimização considerada na formulação proposta para coordenação de múltiplos robôs. Essa *toolbox* utiliza o método de pontos interiores valendo-se do método Barreira Logarítmica [31]. A abrangência deste método e todas as suas etapas, assim como toda a sua teoria são, obviamente mais complexas do que este breve resumo sobre a mesma.

O método de otimização utilizado pelo programa considera o seguinte problema de otimização não linear:

$$\begin{aligned} & \text{Min } f(x) \\ & \text{sujeito a :} \\ & g(x) = 0 \\ & h(x) \leq 0 \end{aligned} \tag{A.1}$$

O objetivo é minimizar uma função  $f(x)$ , sujeita às restrições de desigualdades  $g(x)$  e de igualdades  $h(x)$ , sendo que  $x$  é um vetor, cujas componentes são as variáveis de otimização. Utiliza-se variáveis de folga  $s$  para transformar as inequações  $h(x)$  em restrições de igualdade [30], isto é,

$$h(x) + s = 0, \quad s \geq 0 \tag{A.2}$$

onde  $s$  é o vetor composto pelas variáveis de folga  $s_i$ . O próximo passo é relaxar a restrição  $s_0$ . Para tanto, é inserido na função objetivo o parâmetro de barreira  $\mu$ , com  $\mu$  tendendo a zero. E o problema de otimização é reescrito como:

$$\begin{aligned} & \text{Min } f(x) - \mu \sum_{i=1}^n \ln(s_i) \\ & \text{sujeito a :} \\ & g(x) = 0 \\ & h(x) + s = 0 \end{aligned} \tag{A.3}$$

Como o problema em questão é de minimização, o termo logarítmico age neste caso como uma barreira que impede a violação da restrição de desigualdade. Isto garante que, se a restrição de desigualdade é satisfeita na condição inicial, a trajetória da solução será localizada no interior da região das soluções viáveis com relação à referida restrição. Por esta razão, este termo é denominado barreira logarítmica.

A função Lagrangiana é dada por:

$$L = f(x) - \mu \sum_{i=1}^n \ln(s_i) + \pi^t(h(x) + s) + \lambda^t g(x) \quad (\text{A.4})$$

onde  $\pi$  e  $\lambda$  são os multiplicadores de Lagrange.

Aplicando as condições de otimalidade, obtém-se:

$$\nabla L = \begin{bmatrix} \nabla f(x) + \pi^t \nabla h(x) + \lambda^t \nabla g(x) \\ h(x) + s \\ g(x) \\ S\pi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu e \end{bmatrix} \quad (\text{A.5})$$

onde  $S$  é a matriz diagonal cujos elementos diagonais são os elementos de  $s$ . Utilizando o método de Newton, chega-se em:

$$\begin{pmatrix} \nabla^2 f(x) + \pi^t \nabla^2 h(x) + \lambda^t \nabla^2 g(x) & \nabla h(x) & \nabla g(x) & 0 \\ \nabla h(x) & 0 & 0 & I \\ \nabla g(x) & 0 & 0 & 0 \\ 0 & S & 0 & \pi \end{pmatrix} \cdot \begin{pmatrix} dx \\ d\pi \\ d\lambda \\ ds \end{pmatrix} = \begin{pmatrix} \nabla f(x) + \pi^t \nabla h(x) + \lambda^t \nabla g(x) \\ h(x) + s \\ g(x) \\ -\mu e + S\pi \end{pmatrix} \quad (\text{A.6})$$

sendo que  $I$  representa a matriz identidade. Dessa forma, consegue-se obter as direções  $dx$ ,  $d\pi$ ,  $d\lambda$  e  $ds$ . A partir daí, consegue-se calcular os passos  $\alpha_p$  e  $\alpha_d$  correspondentes as variáveis primais e duais respectivamente, para que  $s$  e  $\pi$  permaneçam estritamente positivas. Os passos são calculados a fim de preservarem a interioridade de todas as variáveis restritas do problema.

Sejam:

$$\rho_p = \min \left( \frac{-s_i}{ds_i} \right), \quad ds_i \leq 0 \quad (\text{A.7})$$

$$\rho_d = \min \left( \frac{-\pi_i}{d\pi_i} \right), \quad d\pi_i \leq 0 \quad (\text{A.8})$$

Sendo assim, o passo  $\alpha$  é dado por:

$$\alpha = \min(1, \rho_p, \rho_d) \quad (\text{A.9})$$

Conhecendo as direções e os passos, todas as variáveis podem ser atualizadas a cada iteração, até que uma certa tolerância seja atingida:

$$\begin{aligned}
 x^{k+1} &= x^k + \alpha dx \\
 s^{k+1} &= s^k + \alpha ds \\
 \pi^{k+1} &= \pi^k + \alpha d\pi \\
 \lambda^{k+1} &= \lambda^k + \alpha d\lambda
 \end{aligned}
 \tag{A.10}$$

E a atualização do parâmetro barreira é dada por:

$$\mu^{k+a} = \frac{\mu^k}{\beta}, \quad \beta \geq 1
 \tag{A.11}$$

O algoritmo para a resolução de um problema de otimização via método de Pontos Interiores versão Primal-Dual é sumarizado nos passos descritos a seguir:

1. Inicialização das variáveis satisfazendo as condição  $(s, \pi) > 0$  e  $(\lambda, \beta, \mu) > 0$ ;
2. Cálculo do vetor gradiente da função Lagrangeana (Eq. A.4);
3. Cálculo da Hessiana e resolução do sistema;
4. Cálculo do fator de passo  $\alpha$  na direção obtida (Eq. A.9);
5. Atualização das variáveis primais e duais (Eq. A.10)
6. Atualização do parâmetro de barreira  $\mu$  (Eq. A.11) até que  $|\nabla L|$  seja menor que uma tolerância determinada.