

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



DISSERTAÇÃO DE MESTRADO

*Controle Preditivo Não Linear Baseado em Modelo Aplicado a
Formações de Robôs Móveis Terrestres com Seleção Dinâmica
de Líder*

Augusto de Holanda Barreto Martins Tavares

João Pessoa - PB

Julho de 2018

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Controle Preditivo Não Linear Baseado em Modelo Aplicado a
Formações de Robôs Móveis Terrestres com Seleção Dinâmica
de Líder

Augusto de Holanda Barreto Martins Tavares

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Informática da Universidade Federal da Paraíba como parte dos requisi-
tos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sinais, Sistemas Digitais e Gráficos

Tiago Pereira Nascimento
(Orientador)

João Pessoa, Paraíba, Brasil

©Augusto de Holanda Barreto Martins Tavares, 26 de Julho de 2018

Catálogo na publicação
Seção de Catalogação e Classificação

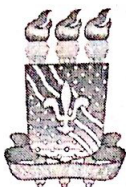
T 231c Tavares, Augusto de Holanda Barreto Martins.

Controle preditivo não linear baseado em modelo
aplicado a formações de robôs móveis terrestres com
seleção dinâmica de líder / Augusto de Holanda Barreto
Martins Tavares. - João Pessoa, 2018.
77 f. : il.

Orientação: Tiago Pereira Nascimento.
Coorientação: Alisson Vasconcelos Brito.
Dissertação (Mestrado) - UFPB/CI.

1. NMPC. 2. Controle de Formação. 3. Robôs Móveis Não
Holonômicos. 4. Seleção de Líder. 5. Controle Não
Linear. I. Pereira Nascimento, Tiago. II. Vasconcelos
Brito, Alisson. III. Título.

UFPB/BC

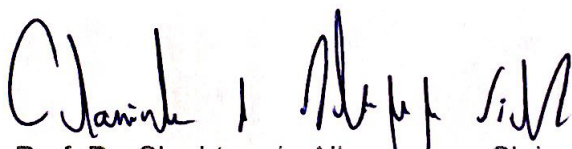


UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Augusto de Holanda Barreto Martins Tavares, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 26 de julho de 2018.

1 Aos vinte e seis dias do mês de julho, do ano de dois mil e dezoito, às dez horas, no Centro
2 de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
3 membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Augusto de
4 Holanda Barreto Martins Tavares, vinculado a esta Universidade sob a matrícula nº
5 20161020274, candidato ao grau de Mestre em Informática, na área de "Sistemas de
6 Computação", na linha de pesquisa "Sinais, Sistemas Digitais e Gráficos", do Programa de
7 Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão
8 examinadora foi composta pelos professores: Tiago Pereira do Nascimento (PPGI-UFPB),
9 Orientador e Presidente da Banca, Camila Mara Vital Barros (UFPB), Examinadora Externa
10 ao programa, Antonio Marcus Nogueira Lima (UFCG), Examinador Externo à Instituição.
11 Dando início aos trabalhos, o Presidente da Banca, cumprimentou os presentes, comunicou
12 aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo
13 fizesse a exposição oral do trabalho de dissertação intitulado "Controle Preditivo Não-Linear
14 Baseado em Modelo Aplicado a Formações de Robôs Móveis com Seleção Dinâmica de
15 Líder". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o
16 seguinte parecer: "**aprovado**". Do ocorrido, eu, Claurton de Albuquerque Siebra,
17 Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai
18 assinada por mim e pelos membros da banca examinadora. João Pessoa, 26 de julho de
19 2018.

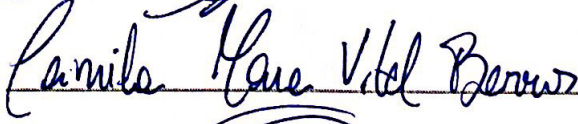


Prof. Dr. Claurton de Albuquerque Siebra

Prof. Dr. Tiago Pereira do Nascimento
Orientador (PPGI-UFPB)

Prof. Dr^a. Camila Mara Vital Barros
Examinadora externa ao programa (UFPB)

Prof. Dr^a. Antonio Marcus Nogueira Lima
Examinadora externa à instituição (UFCG)


Agradecimentos

Agradeço ao professor Tiago Pereira Nascimento, pela orientação que foi de essencial importância para a realização desta pesquisa, desde a sugestão de tema em seus primórdios até a ajuda com as dificuldades em seus momentos finais. Ao professor Alisson Vasconcelos de Brito, assim como aos demais integrantes do LASER, por providenciar um ambiente de pesquisa que estimula novas ideias. Aos professores e funcionários do PPGI, que muitas vezes trabalham anônimos.

Aos meus familiares e amigos, que foram porto seguro, empurrão nas costas, balcão de bar e abraço de mãe, tudo de uma vez só.

A minha esposa, a quem eu devo tanto que os agradecimentos não cabem nesta página, no capítulo ou na dissertação.

“Link, you may not yet be at a point where you have fully recovered your power or all of your memories, but courage need not be remembered, for it is never forgotten.”

Zelda, Princess of Hyrule

Resumo

Esta pesquisa apresenta uma arquitetura de controle distribuído para controle de formações de robôs, aplicando uma versão modificada do algoritmo de controle do *Non-linear Model Predictive Control* com uma função custo genérica, que permite a incorporação da camada comportamental ao controlador, mantendo ou melhorando a precisão do algoritmo. Também propõe uma estrutura que permite um processo de seleção de líder dinâmico para a formação de robôs. São apresentados de maneira breve os conceitos teóricos que fundamentam o texto e o estado da arte, contendo as referências que foram utilizadas como base para o trabalho. As modificações do *Non-linear Model Predictive Control*, que emprega um líder virtual para melhorar o comportamento do sistema em malha fechada, e a estrutura baseada em árvore proposta, que opera como uma máquina de estados, são elaboradas. São apresentados os resultados obtidos com esse algoritmo na resolução do problema de rastreamento de trajetória para uma formação de três robôs terrestres utilizando uma estrutura líder-seguidor, tanto em ambiente simulado como experimental.

Palavras-chave: NMPC, Controle de Formação, Robôs Móveis Não Holonômicos, Seleção de Líder, Controle Não Linear

Abstract

This research presents a distributed control architecture for robot formation control, using a modified version of the Non-linear Model Predictive Control algorithm with a generic cost function, which allows for the combination of the behavior layer into the controller while keeping or improving upon the precision algorithm. A dynamic leader-selection structure is also proposed. The theoretical concepts which served as a basis for the text and the reference works which constitute the state of the art are briefly presented. The proposed Non-linear Model Predictive Control modifications, which employs a virtual leader to improve closed loop performance, and the tree based selection structure, used as a state machine, are explained. The results obtained for a three robot formation dealing with the trajectory tracking problem are shown, for both simulated and experimental environments.

Keywords: NMPC, Formation Control, Non-Holonomic Mobile Robots, Leader Selection, Non-linear Control

Sumário

Lista de Figuras	vii
Lista de Tabelas	x
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Metodologia	3
1.4 Contribuições	4
1.5 Estrutura da Dissertação	5
2 Fundamentação Teórica	6
2.1 Modelo de um Robô Móvel Não holonômico	6
2.2 Controle Preditivo Não Linear Baseado em Modelo (NMPC)	10
2.3 Controle de Formações de Robôs Móveis	14
2.4 Considerações Finais	17
3 Trabalhos Relacionados	18
3.1 Controle Preditivo Não Linear Baseado em Modelo	18
3.2 Controle de Formações de Robôs	21
3.3 Considerações Finais	23
4 <i>Non linear Model Predictive Control</i> Modificado para Formações de Robôs e Se- leção Dinâmica de Líder	24
4.1 NMPC Modificado para Controle de Formação	24
4.1.1 Otimizador e Preditor	25
4.1.2 Função Custo	26

4.1.3	<i>Resilient Propagation</i>	30
4.1.4	Análise das Restrições	30
4.2	Estrutura para Seleção de Líder Dinâmico	33
4.3	Arquitetura de Controle	37
4.4	Considerações Finais	38
5	Resultados	39
5.1	Simulação	39
5.1.1	Parâmetros de Simulação	39
5.1.2	Resultados Simulados	42
5.2	Experimentos	49
5.2.1	Arranjo Experimental	49
5.2.2	Resultados Experimentais	50
5.3	Considerações Finais	55
6	Conclusão	56
	Referências Bibliográficas	63

Lista de Figuras

1.1	Exemplos de diversas aplicações de formações de robôs móveis em pesquisa, na indústria e em competições.	2
2.1	Modelo de um robô móvel não holonômico.	7
2.2	Diagrama com as etapas do processo de um controlador NMPC para um caso geral.	13
2.3	Imagem ilustrando todas as combinações possíveis de tipos de camadas de Controle e de Comportamento para Formações de Robôs Móveis. As linhas vermelhas representam o comportamento da formação, enquanto os elementos em azul representam o controle. Em sentido horário, a partir do canto superior esquerdo: controle e comportamento centralizados, controle centralizado e comportamento distribuído, controle e comportamento distribuídos e controle distribuído e comportamento centralizado.	17
4.1	Diagrama mostrando o processo de construção da função custo a partir do bloco preditor e do modelo do robô.	26
4.2	Mapa da restrição de velocidade linear da plataforma Turtlebot 2. Áreas em azul indicam regiões fora das capacidades do sistema.	32
4.3	Mapa dos valores de v_x e v_y que satisfazem à Equação 2.1 para algum ω , representados pelos pontos em amarelo.	32
4.4	Estrutura em árvore proposta para uma formação com n robôs.	33
4.5	Estrutura em árvore proposta para uma formação com n robôs.	37
4.6	Sistema implementado no Simulink para as simulações.	38
5.1	Plataforma Pioneer 3.	40

5.2	Trajatória de α durante o seu tempo como líder de formação em ambiente simulado.	42
5.3	EQM entre α e a trajetória de referência durante o seu tempo como líder de formação no ambiente simulado.	43
5.4	Composição da trajetória dos robôs durante o seu tempo como líder de formação em um ambiente simulado. Os desvios correspondem ao robô desviando de um ex-membro da formação.	44
5.5	MSE dos três robôs durante a simulação. Os gráficos de α e β estão apresentados apenas até o ponto em que ambos apresentaram falhas, o que corresponde ao momento em que ocorre a troca do líder da formação.	44
5.6	EQM entre β e a trajetória de referência em simulação.	45
5.7	EQM entre γ e a trajetória de referência em simulação.	45
5.8	Comparação do termo J_1 da função custo para os três robôs durante a o tempo de α como líder.	47
5.9	Análise dos termos da função custo de α até o ponto em que foi observada uma falha.	47
5.10	Análise dos termos da função custo de β até o ponto em que foi observada uma falha.	48
5.11	Análise dos termos da função custo de γ	48
5.12	Arranjo experimental utilizado. α é o líder inicial.	49
5.13	Trajatória de α durante o seu tempo como líder de formação em ambiente experimental.	51
5.14	EQM entre α e a trajetória de referência durante o seu tempo como líder de formação em ambiente experimental.	51
5.15	Composição da trajetória dos robôs durante o seu tempo como líder de formação em um ambiente experimental. Os desvios correspondem ao robô desviando de um ex-membro da formação.	52
5.16	EQM dos três robôs durante o experimento. Os gráficos de α e β estão apresentados apenas até o ponto em que ambos apresentaram falhas, o que corresponde ao momento em que ocorre a troca do líder da formação.	52
5.17	EQM entre β e a trajetória de referência no experimento.	53

5.18 EQM entre γ e a trajetória de referência no experimento.	53
--	----

Lista de Tabelas

5.1	Parâmetros de simulação para a plataforma Pioneer 3.	41
5.2	Conjunto de ganhos para a função custo na plataforma Pioneer 3 em ambiente simulado.	41
5.3	EQM médio para os três robôs em relação às suas diferentes referências - Simulação	46
5.4	Variância do EQM para os três robôs em relação às suas diferentes referências - Simulação	46
5.5	Parâmetros experimentais para a plataforma Turtlebot 2.	49
5.6	Conjunto de ganhos para a função custo na plataforma Turtlebot 2 em ambiente experimental.	50
5.7	EQM médio para os três robôs em relação às suas diferentes referências - Experimento	54
5.8	Variância do EQM para os três robôs em relação às suas diferentes referências - Experimento	54
5.9	Comparação dos resultados obtidos nos ambientes simulado e experimental com os valores encontrados na literatura.	55

Capítulo 1

Introdução

Este capítulo apresenta uma introdução para o texto, fornecendo o contexto de pesquisa que motiva o trabalho, as metas e contribuições obtidas, assim como a metodologia empregada durante a pesquisa para a obtenção de resultados. Por fim, é delineada a organização do documento.

1.1 Motivação

Sistemas automatizados que envolvem robôs móveis são uma área de pesquisa amplamente estudada, visando auxiliar ou substituir humanos na realização de tarefas tediosas ou que apresentem risco [Simba, Uchiyama e Sano 2016]. Dentro desse contexto, o estudo de formações de robôs é de particular importância [Sakurama 2017], uma vez que para a maioria das tarefas desejadas é necessária a participação de mais de um agente. Entretanto, a maioria das pesquisas nessa área tende a ficar restrita a ambientes simulados [Karpov e Karpova 2015], fruto da natureza fragmentada do estudo da robótica, que exige a solução de problemas em múltiplas áreas para a implementação de um sistema funcional que possibilite a realização de experimentos que visam validar apenas um dos componentes do sistema. Na Figura 1.1 é possível ver diversos exemplos de aplicações que utilizam formações de robôs móveis.

Ao mesmo tempo, há um crescente interesse em técnicas de controles capazes de lidar com não linearidades, tanto as inerentes ao sistema como as provenientes de perturbações, uma vez que a maioria dos sistemas tem de ser capaz de lidar com essas limitações na prática.



Figura 1.1: Exemplos de diversas aplicações de formações de robôs móveis em pesquisa, na indústria e em competições.

Entre os diversos algoritmos de controle capazes de lidar com não linearidades, o Controle Preditivo Não-Linear Baseado em Modelo, do inglês *Non-Linear Model Predictive Control* (NMPC), tem tido destaque para aplicações na área de robótica, como pode ser visto em [Ostafew, Schoellig e Barfoot 2014] e [Du, Htet e Tan 2016], graças à sua capacidade de lidar com as não linearidades mencionadas acima, assim como com as restrições e incertezas inerentes ao sistema, além de possuir a possibilidade de ser adaptado a cada caso por meio do ajuste dos seus elementos.

Assim, fica claro o espaço existente para uma pesquisa que alie o uso do NMPC como método de controle capaz de lidar com não linearidades em conjunto com o controle de uma formação de robôs móveis terrestres lidando com o problema de rastreamento de trajetória, sendo o algoritmo e sistema desenvolvidos testados em um ambiente experimental, fator este que testa o seu desempenho quando submetidos a condições reais. Ainda, aproveitando-se da adaptabilidade do NMPC, foi proposto, e nesta pesquisa mostrado, que é possível incorporar a camada comportamental de um sistema multirrobô, responsável por definir os parâmetros de comportamento para cada membro de uma formação, ao controlador em si, diminuindo a complexidade do sistema. Além disso, o uso de uma versão modificada do NMPC, visando a aplicação específica de rastreamento de trajetórias em robôs móveis, permitiria um melhor desempenho, com trajetórias mais estáveis e erros menores.

1.2 Objetivos

O objetivo central da pesquisa que é o objeto desta dissertação é a implementação de um algoritmo de NMPC para o controle formações de robôs móveis não holonômicos, utilizando uma estratégia líder-seguidor. De maneira mais detalhada, a meta é implementar a versão modificada do NMPC apresentada em [Nascimento, Dórea e Gonçalves 2018], que usa apenas um robô e utiliza um líder virtual para melhorar o desempenho em malha fechada, levando menos tempo para definir os sinais de controle ótimos, quando comparada a versões tradicionais do algoritmo, em um sistema multirrobô lidando com o problema de rastreamento de trajetória, juntamente com uma estratégia de controle de formação baseada na manipulação dos ganhos da função custo do NMPC, que será devidamente explicada no Capítulo 4, que permite que seja omitida do sistema de controle de formação uma camada dedicada apenas ao comportamento dos membros da função, sendo esta integrada à camada de controle de trajetória. Isso implica deixar de utilizar funções e métodos que exijam mais das capacidades computacionais do sistema para a definição do comportamento de cada robô, passando a ser necessário apenas um conjunto de ganhos referente ao comportamento desejados.

Ainda, tem-se como objetivo a incorporação de uma heurística que permite a seleção do líder de formação de maneira dinâmica, o que é uma tendência para os trabalhos que lidam com múltiplos robôs, como pode ser visto em [Saeidi, Mikulski e Wang 2017] e [Li et al. 2017]. Essa necessidade surge da natureza dinâmica do ambiente em que a maioria das aplicações de robótica é realizada, fato que leva à necessidade de elaborar estratégias que garantam o funcionamento da formação em caso de eventos inesperados, especialmente quando se trata de uma falha do robô-líder em uma estrutura do tipo líder-seguidor. Para o caso aqui estudado, a meta é que o sistema seja capaz de definir um novo líder caso o atual sofra uma falha por perda de comunicação com os demais componentes do sistema.

1.3 Metodologia

Foi adotada uma metodologia de pesquisa iterativa baseada em incrementos graduais de complexidade, alternando entre os ambientes simulado e experimental. O primeiro passo foi

o estudo do NMPC e do modelo de um robô móvel não holonômico. Concluída essa etapa, foi elaborado um modelo do sistema de controle na plataforma Simulink do MATLAB, que foi testado. O mesmo controlador da simulação foi então aplicado para o robô real, utilizando a integração do *Robot Operating System* (ROS), por meio da *Robotics Systems Toolbox* com o Simulink.

Após o sucesso da etapa anterior, foi realizado um estudo acerca do controle de formações de robôs, que por sua vez permitiu a implementação de um modelo multirrobô baseado na estrutura que já havia sido desenvolvida. Esse modelo foi testado tanto em simulação como em experimentos. Na etapa final, foi introduzida uma heurística que permite que o líder da formação seja selecionado de maneira dinâmica a nível de controle. Todos os resultados simulados foram obtidos mediante utilização do modelo descrito em [Martins 2010] para um robô Pioneer 3, enquanto os resultados experimentais foram realizados na plataforma Turtlebot 2.

1.4 Contribuições

O trabalho apresenta como contribuições principais:

- O desenvolvimento de uma plataforma de testes sólida para experimentos envolvendo o NMPC modificado aplicado a formações de robôs móveis;
- Expansão do NMPC modificado proposto em [Nascimento, Dórea e Gonçalves 2018] para um sistema multirrobô;
- Seleção de líder dinâmica a nível de controle, que simplifica a estrutura da formação quando comparada esquemas de controle tradicionais, onde o comportamento dos membros da formação não está atrelado ao controlador.

A pesquisa também produziu dois artigos, um já publicado e outro em processo de publicação:

- TAVARES, A. d. H. B. M. et al. *Non-linear model predictive formation control applied to non-holonomic autonomous mobile robots*. In: 2017 Latin American Robotics

Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR). 2017. p.1–@
(Publicado)

- TAVARES, A. d. H. B. M. et al. *Dynamic Leader Allocation in Multi-Robot Systems Based on Nonlinear Model Predictive Control*. In: IEEE Robotics and Automation Letters. 2018.(Submetido)

1.5 Estrutura da Dissertação

Esse texto segue a seguinte estrutura: no Capítulo 2 são apresentados os conceitos teóricos que servem de base para o trabalho de pesquisa realizado. No Capítulo 3 é apresentado o estado da arte na área, situando os métodos utilizados e os resultados obtidos dentro da produção atual na área. O Capítulo 4 é dedicado a explicar as contribuições do trabalho, como a versão modificada do algoritmo NMPC utilizada e a estrutura de controle e seleção de líder implementada para a obtenção de resultados tanto simulados quanto experimentais. O Capítulo 5 é dedicado à apresentação e análise dos resultados obtidos em simulação e no ambiente experimental. Finalmente, o Capítulo 6 apresenta as conclusões obtidas durante a pesquisa, assim como indica direções para trabalhos futuros. Na Bibliografia são apresentados os textos que foram utilizados como referência.

Capítulo 2

Fundamentação Teórica

O propósito deste capítulo é elucidar os conceitos teóricos que norteiam este trabalho, demonstrando de forma sucinta e clara os princípios básicos que foram utilizados ao longo da dissertação. A base teórica por trás do trabalho pode ser dividida essencialmente em três áreas: o estudo do modelo de um robô móvel não holônomico, a análise do algoritmo de controle preditivo não-linear baseado em modelo e, por fim, as técnicas referentes ao controle de formações de robôs. Cada uma dessas áreas será devidamente explorada em sua respectiva seção.

2.1 Modelo de um Robô Móvel Não holonômico

Esta seção é dedicada ao modelo matemático que descreve o comportamento físico de um robô móvel não holonômico. O modelo é baseado na imagem da Figura 2.1, que mostra um robô de duas rodas unidas por um eixo sobre um plano, onde são consideradas apenas as suas características cinemáticas, uma vez que o deslocamento considerado aqui se dá apenas em duas dimensões. Isso permite a simulação do comportamento das bases Pioneer 3 e Turtlebot 2, sobre as quais foi desenvolvido o trabalho, mas suas características são gerais o suficiente para permitir sua aplicação a outros modelos que apresentem características similares, principalmente no que tange às suas restrições.

O primeiro passo é definir o significado da natureza não holônica do sistema. Segundo [Martins 2010], ao longo de um movimento, um sistema é submetido a um conjunto de restrições que devem ser obedecidas, e que tomam a forma de equações relacionando a sua

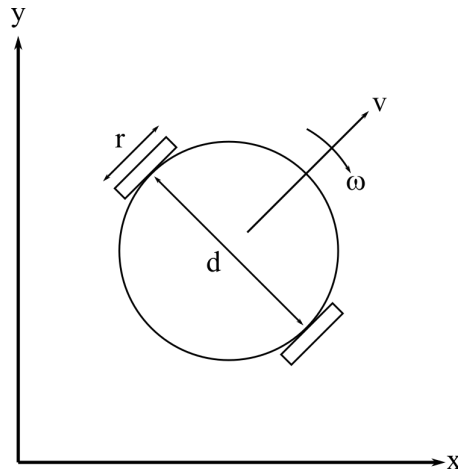


Figura 2.1: Modelo de um robô móvel não holonômico.

posição e a sua velocidade. Caso uma das restrições do sistema não seja integrável, diz-se que essa restrição, e, por conseguinte, o sistema como um todo, é não holonômico.

Em termos práticos, isso implica uma redução dos graus de liberdade do sistema. No caso aqui abordado, isso é uma consequência de o fato do mecanismo de movimento do robô mostrado na Figura 2.1 não ser capaz de se mover ao longo de uma direção arbitrária sobre plano, variando apenas a sua velocidade linear, sendo necessária a mudança de sua velocidade angular em conjunto para qualquer movimento cuja velocidade linear resultante não seja ortogonal ao eixo das rodas do robô. Essa restrição, considerando que não há deslizamento lateral, é representada pela Equação 2.1, onde $\dot{x}$ e $\dot{y}$ são as velocidades lineares ao longo dos eixos x e y , respectivamente, e θ é a orientação do robô em relação ao eixo x .

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (2.1)$$

Ou seja, caso um par de velocidades angular e linear não respeite a restrição da Equação 2.1, o movimento descrito não é possível para a estrutura física do robô. Tendo em mente que esta restrição deve ser respeitada quando se trata do controle do robô, podemos então estabelecer o modelo cinemático para um robô móvel não holonômico. O primeiro passo é a definição de um vetor com as variáveis de estado relevantes, que pode ser visto na Equação 2.2, onde x e y são a posição do robô no plano de referência e θ é a sua orientação em relação

ao eixo x desse plano.

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2.2)$$

Uma das consequências de o sistema ser não holonômico pode ser vista na Equação 2.3, onde estão representadas as entradas do sistema: a velocidade linear v e a velocidade angular ω . Em um sistema holonômico esse vetor seria formado pelas componentes de velocidade linear ao longo de cada um dos eixos do plano de referência, o que resultaria em um sistema mais simples.

$$\mathbf{U} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.3)$$

De posse das Equações 2.2 e 2.3, é possível definir o sistema apresentado na Equação 2.4.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} \delta v_x \\ \delta v_y \\ \delta \omega \end{bmatrix} \quad (2.4)$$

As expressões para $\dot{x}$, $\dot{y}$ e $\dot{\theta}$ em função das entradas são obtidas a partir da aplicação de conceitos básicos de cinemática e trigonometria. δv_x , δv_y e $\delta \omega$ representam possíveis perturbações no sistema, como o atrito com o solo, inclinação do plano de movimento e perdas no motor das rodas. Utilizando as expressões encontradas em [Nascimento, Dórea e Gonçalves 2018], e que podem ser vistas nas Equações 2.5 e 2.6, é possível relacionar v e ω com as características físicas do robô, sendo r o raio das rodas e d a distância entre elas, e com as velocidades angulares das rodas, onde ω_1 e ω_2 são as velocidades angulares produzidas pelas rodas direita e esquerda, respectivamente.

$$v = \frac{r}{2}(\omega_1 + \omega_2) \quad (2.5)$$

$$\omega = \frac{r}{d}(\omega_1 - \omega_2) \quad (2.6)$$

As equações acima abrem duas possibilidades. A primeira é a elaboração de um modelo correspondente ao da Equação 2.4 em que os elementos do vetor de entradas U apresentado na Equação 2.3 são substituídos pelas velocidades angulares das rodas. A segunda é a formulação das expressões que realizam o caminho inverso, ou seja, permitem que a partir das velocidades linear e angular desejadas para o robô sejam obtidas as velocidades angulares das rodas correspondentes, levando em consideração as suas dimensões. O modelo modificado, assim como as expressões para as velocidades angulares ω_1 e ω_2 das rodas em função de v e ω estão nas Equações 2.7, 2.8, 2.9, respectivamente.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} + \begin{bmatrix} \delta v_x \\ \delta v_y \\ \delta \omega \end{bmatrix} \quad (2.7)$$

$$\omega_1 = \frac{1}{r} \left(v + \frac{d}{2} \omega \right) \quad (2.8)$$

$$\omega_2 = \frac{1}{r} \left(v - \frac{d}{2} \omega \right) \quad (2.9)$$

Embora o sistema acima descrito possa descrever os movimentos do robô, ele pressupõe o entendimento dos mecanismos responsáveis por gerar as velocidades angulares ω_1 e ω_2 . Sendo assim, é necessário o estudo do modelo dos motores responsáveis pelo movimento das rodas. A plataforma Turtlebot trabalha com motores de corrente contínua para a movimentação das rodas. Em [Ferreira 2010] o modelo para um conjunto motor-roda é apresentado, e pode ser visto nas Equações 2.10 e 2.11.

$$u(t) = L_a \frac{di(t)}{dt} + R_a i_a(t) + K_V \omega_m(t) \quad (2.10)$$

$$T_m = K_T i_a(t) \quad (2.11)$$

As variáveis apresentadas nas Equações 2.10 e 2.11 são:

- Tensão nos terminais do motor u ;
- Corrente na armadura do motor i_a ;

- Indutância de armadura do motor L_a ;
- Resistência de armadura do motor R_a ;
- Constante de força eletromotriz K_V ;
- Constante de torque K_T ;
- Velocidade angular do motor ω_m ;
- Torque do motor T_m .

É importante notar que, apesar de ser importante compreender o mecanismo pelo qual o robô é capaz de realizar seu movimento, análises de cunho mais específico, como, por exemplo, a determinação da correspondência entre os níveis de tensão fornecidos na entrada do motor e a rotação na saída fogem ao escopo do trabalho, uma vez que a utilização do ROS em conjunto com a plataforma Turtlebot 2 permite o controle do robô apenas com o fornecimento dos sinais de v e ω apropriados para a trajetória desejada. Entretanto, é importante ter em mente o modelo do motor, uma vez que o seu funcionamento é o responsável por inserir perturbações não lineares no modelo da Equação 2.4, relacionadas a perdas elétricas e mecânicas.

Outro ponto a ser explorado é a necessidade de discretizar o modelo apresentado. Isso é feito por meio de uma aproximação trapezoidal, e as expressões correspondentes podem ser vistas nas Equações 2.12, onde T é o passo de discretização.

$$x(k+1) = x(k) + Tv \cos(\theta) \quad (2.12a)$$

$$y(k+1) = y(k) + Tv \sin(\theta) \quad (2.12b)$$

$$\theta(k+1) = \theta(k) + T\omega \quad (2.12c)$$

2.2 Controle Preditivo Não Linear Baseado em Modelo (NMPC)

De acordo com [Grüne e Pannek 2011], o Controle Preditivo Não-Linear Baseado em Modelo, é um método de controle baseado em otimização para sistemas que sejam realimen-

tados. Suas principais aplicações são em problemas de estabilização e rastreamento, e a sua capacidade de lidar com perturbações não lineares sem a necessidade de recorrer a métodos de linearização o torna um método particularmente atrativo.

Para o contexto deste trabalho, a aplicação utilizada é o rastreamento de trajetória para robôs móveis. A Equação 2.13 representa o estado do sistema, correspondente a pose do robô, em k instantes discretos, a partir do vetor apresentado na Equação 2.2.

$$\mathbf{X}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}, k \in \mathbb{N} \quad (2.13)$$

O objetivo do problema de rastreamento é que o vetor de estados do sistema em um instante k qualquer esteja o mais próximo possível de um vetor de referência $\mathbf{X}_{ref}$ desejado. Isso é equivalente a minimizar o valor dos termos de e na Equação 2.14.

$$e = \mathbf{X}(k) - \mathbf{X}_{ref}(k) = \begin{bmatrix} x(k) - x_{ref}(k) \\ y(k) - y_{ref}(k) \\ \theta(k) - \theta_{ref}(k) \end{bmatrix}, k \in \mathbb{N} \quad (2.14)$$

Considerando, então, um modelo como o apresentado na Equação 2.4, que é do tipo $x(k+1) = f(x(k), u(k))$, como pode ser visto nas Equações 2.12, onde o estado do sistema x e a entrada u no instante k determinam o estado do sistema no instante $k+1$, é possível obter uma predição do estado do sistema em um instante de tempo futuro a partir de um conjunto de entradas de controle, desde que o estado atual do sistema seja conhecido. Assim, para um conjunto de n entradas de controle tem-se uma predição do estado do sistema x_p até um instante $k+n$. Esse processo é apresentado na Equação 2.15.

$$\begin{aligned} x_p(k+1) &= f(x(k), u(k)) \\ x_p(k+2) &= f(x(k+1), u(k+1)) \\ &\vdots \\ x_p(k+n) &= f(x(k+n-1), u(k+n-1)) \end{aligned} \quad (2.15)$$

Nesse momento, um processo de otimização é aplicado para determinar os valores de u tais que x_p minimize o valor dos termos de e . Para isso é utilizada uma função l que pode

levar em consideração não só a distância entre o estado previsto do sistema em relação à referência, normalmente calculada a partir da norma euclidiana entre os dois vetores, como da entrada de controle em relação a um valor desejado. Essa função é calculada para todos os valores de x_p obtidos, e o processo de minimização é ao resultado desse processo, o que resulta em entradas de controle que levam em consideração não só qual o melhor comportamento do sistema no próximo instante do laço de controle, mas também o melhor comportamento n instantes no futuro, de acordo com a previsão fornecida pelo modelo. Sendo assim, o problema pode ser reduzido à minimização do termo J , apresentado na Equação 2.16.

$$J(x(k), u(k)) = \sum_{i=1}^n l(x_p(k+i), u(k+i-1)) \quad (2.16)$$

Assumindo que o problema de otimização de J possui solução e que essa solução é uma sequência de controle ótima u_o com n elementos, o valor da entrada de controle para que o estado do sistema $\mathbf{X}$ seja o mais próximo possível de $\mathbf{X}_{ref}$ no instante k é $u_o(k)$. Esse valor é fornecido ao sistema, e as entradas $u_o(k+1), u_o(k+2), \dots, u_o(k+n-1)$ são descartadas. A partir desse ponto o processo recomeça, tendo como estado inicial do sistema $x(k+1)$. Esse é o fator de que define o NMPC como um processo de controle iterativo. Esse processo como um todo está apresentado de maneira gráfica na Figura 2.2.

Algumas observações devem ser feitas sobre o algoritmo explicado acima. A primeira é deixar explícita a importância do modelo $x(k+1) = f(x(k), u(k))$ para o processo. Como o comportamento do sistema até n , denominado horizonte de predição, é previsto iterando sobre o modelo, caso o modelo não represente o sistema com fidelidade a sequência de controle obtida a partir do processo de otimização não conseguirá controlar o sistema. Esse é o mesmo motivo pelo qual as entradas de instantes futuros são descartadas e o processo recomeça para cada instante de tempo, uma vez que por mais preciso que um modelo seja, a medida que se avança no horizonte de predição há um acúmulo de erros e a precisão dos valores encontrados diminui. Outro fator importante é notar que não há nenhuma exigência acerca da natureza do modelo envolvido, podendo este ser não linear. No Capítulo 5, será mostrado que o modelo apresentado na Seção 2.1 resulta em um bom desempenho do controlador, o que evidencia outra característica na seleção do modelo de predição que é satisfazer o meio termo entre fidelidade e simplicidade, uma vez que iterações para predição de sistemas mais complexos tendem a ser mais custosas.

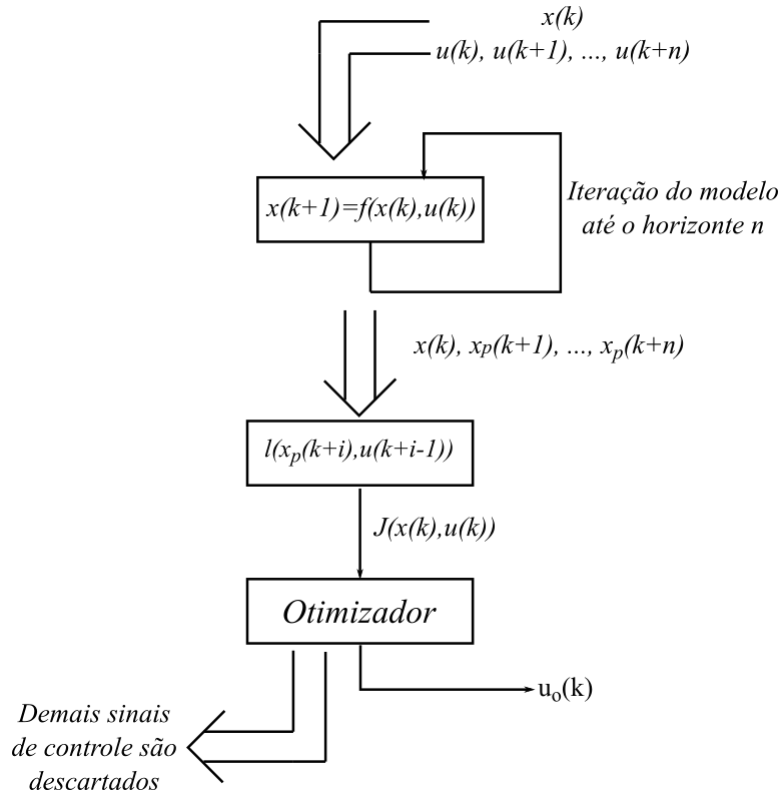


Figura 2.2: Diagrama com as etapas do processo de um controlador NMPC para um caso geral.

Outra análise importante é a definição de um valor apropriado para o n . Valores elevados de n levam a mais iterações em cada etapa do processo de controle, o que pode elevar o tempo necessário para encontrar a sequência de controle ótima u_o a partir da minimização de J , e, como foi dito acima, a partir de um certo ponto as previsões do modelo passam a perder confiabilidade. Entretanto, valores muito pequenos de n levam à perda de um dos maiores atrativos do NMPC, que é a definição de entradas controle ótimas que levam em consideração não só o estado atual do sistema, mas também os seus estados futuros. Na literatura, o valor de n é comumente definido de forma empírica, como pode ser visto em [Wilson, Charest e Dubay 2016], [Feng, Lu e Chen 2015] e [Xiao et al. 2017]. Essa é normalmente a abordagem adotada, pois, como é demonstrado em [Ferreira 2010], o valor ideal de n para um sistema depende de suas características específicas, sendo difícil a determinação de um método geral para a definição do horizonte de controle de uma aplicação qualquer. Entretanto, é possível afirmar que para a vasta maioria das aplicações encontradas na literatura o valor tende a ficar na casa das unidades, outra razão para que ele seja definido de modo empírico, pois uma

busca exaustiva pelo valor ideal pode ser realizada rapidamente.

Por fim, é necessário fazer algumas considerações acerca da natureza *online* do NMPC e do processo de otimização envolvido. A predição dos estados futuros do sistema e a minimização de J são realizadas durante o seu funcionamento, o que significa que o algoritmo possui um tempo de execução limitado pelo tempo de espera disponível até que uma nova entrada de controle seja necessária, sob o risco de o funcionamento do sistema ser interrompido, o que é particularmente desastroso para aplicações em formação, uma vez que o erro em um robô tende a se propagar para os demais. A consequência direta disso é que a heurística utilizada na minimização deve ser capaz de encontrar uma solução em tempo hábil. Como será discutido no Capítulo 4, isso implica a limitação do número de iterações do processo, assim como o estabelecimento de critérios de parada apropriados, de maneira a encontrar-se o meio-termo entre obter um valor ótimo para u_o e um tempo de execução que seja compatível com as necessidades práticas da aplicação de controle. Uma vantagem nesse ponto é o fato de que a formulação do NMPC exige apenas que a heurística adotada nessa etapa seja capaz de solucionar um problema não convexo, tornando possível a escolha do método mais apropriado para o caso em questão.

2.3 Controle de Formações de Robôs Móveis

Antes de mais nada, é importante frisar que, ao contrário dos conceitos apresentados nas Seções 2.1 e 2.2, o tópico aqui discutido carece de um corpo teórico bem estabelecido, sendo este formado a partir dos pontos em comum entre as técnicas utilizadas em diversas publicações, inclusive artigos, teses de doutorado e dissertações de mestrado. Sendo assim, esta Seção explica em linhas gerais quais são as grandes áreas adotadas pelo meio científico no que tange às formações de robôs e o seu controle.

Quando se trata do controle de uma formação de robôs, há duas camadas envolvidas: a camada de controle e a camada de comportamento. Para que se possa entender o propósito dessas duas camadas, é necessário definir as diferenças entre as hierarquias distribuídas e hierarquias centralizadas dentro do seu contexto. Na camada de controle, a diferença entre centralizado e distribuído se refere a se há apenas um controlador responsável por todos os membros da formação ou se cada um deles possui um controlador próprio. Uma abor-

dagem centralizada pode ser vista em [Almonfrey, Vassallo e Dalfior 2010], enquanto uma abordagem distribuída pode ser vista em [Zou et al. 2018].

O principal fator que leva à escolha de um método ou o outro é a dificuldade de elaborar um único controlador mais complexo que lide com o sistema inteiro versus a dificuldade de implementar uma interface entre diversos controladores mais simples. Como será visto no Capítulo 3, atualmente há uma forte tendência de que os sistemas tenham um grau maior de distribuição, uma vez que as técnicas de hardware e software disponíveis favorecem a operação em conjunto de múltiplos dispositivos de capacidade moderada, em vez de um único dispositivo de maior capacidade. Sendo assim, arquiteturas de controle centralizadas têm caído em desuso.

Já na camada de comportamento, a diferença entre centralizado ou distribuído pode ser determinada pela presença, ou não, de um líder de formação. Em formações centralizadas há a presença de um líder, que possui metas e parâmetros a serem seguidos, diferentes daqueles dos demais membros da formação. Tipicamente, o líder de formação possui mais liberdade para realizar o objetivo desejado, como rastrear uma trajetória. Em contrapartida, os demais membros, denominados seguidores, tem o seu comportamento pautado pelas ações do líder. Como exemplo, no caso de rastreamento de trajetória, geralmente o líder é o responsável por seguir o caminho desejado, enquanto os seguidores devem manter uma distância desejada entre o líder e entre eles mesmos. Um exemplo é o trabalho desenvolvido em [Gao e Guo 2018].

Em camadas de comportamento distribuídas há duas opções: o uso de estruturas virtuais ou de comportamentos desejados. No caso de comportamentos desejados é definido um comportamento alvo, que cada robô membro da formação deve tentar atingir desde que não sejam desrespeitados uma série de parâmetros preestabelecidos. Em [Hoshino, Tamura e Hayakawa 2017] é demonstrada uma aplicação deste tipo, e é colocada em evidência uma de suas principais vantagens, que é a flexibilidade da formação na realização da tarefa e a sua capacidade de adaptação, uma vez que os membros possuem mais liberdade. Entretanto, esse nível de liberdade por parte dos membros pode interferir no desempenho para aplicações que necessitem de uma estrutura mais rígida.

Quando se trata de estruturas virtuais, é gerada uma imagem do comportamento desejado para a formação como um todo. Essa imagem é analisada, de maneira a se determinar

o comportamento desejado para cada um dos membros, que recebem um conjunto de instruções específicas destinadas a fazer com que a tarefa desejada para cada um deles dentro da formação seja realizada. A palavra imagem foi utilizada aqui com certa liberdade, uma vez que dentro de uma formação não necessariamente a tarefa de um robô está restrita ao seu posicionamento espacial. A natureza distribuída é uma consequência do fato de cada robô possuir parâmetros operacionais individuais, sendo a formação o resultado da composição dos seus comportamentos. Este tipo de formação é tipicamente muito rígida, uma vez que o conjunto de instruções de cada membro é elaborado pensando no desempenho da formação como um todo, como pode ser visto em [Peterson e Barton 2015].

Diferentemente da camada de controle, onde a abordagem distribuída tem ganho maior destaque, a escolha entre centralizado e distribuído na camada de comportamento é dependente da aplicação desejada, sendo necessário analisar as características da tarefa a ser realizada e o ambiente onde os robôs irão operar, para definir qual das alternativas é a mais adequada. Na Figura 2.3 temos uma imagem mostrando todas as combinações possíveis para uma formação. É apresentada uma formação de três robôs, alternando entre as configurações de controle centralizado, presentes na linha de cima, e distribuído, na linha de baixo. As setas em vermelho representam o comportamento, sendo as formações a esquerda do tipo líder-seguidor e as a direita distribuídas. Como dito anteriormente, sistemas dos tipos apresentados na primeira linha são raros, sendo a maioria dos sistemas utilizados atualmente dos tipos presentes na segunda linha. No Capítulo 4 será mostrado que o sistema utilizado nesta pesquisa possui o controle distribuído e o comportamento centralizado, que se encontra no canto inferior esquerdo da imagem.

Por fim, não se pode deixar de mencionar que independentemente de seus esquemas de controle e comportamento, as formações também estão sujeitas a uma classificação acerca de sua homogeneidade. Especificamente, formações constituídas por apenas um tipo de robô são ditas homogêneas, enquanto formações que possuam dois ou mais tipos de robôs inclusos são classificadas como heterogêneas. Formações heterogêneas apresentam desafios específicos, uma vez que os robôs envolvidos possuem, por definição, características diferentes, em maior ou menor grau, o que significa que para a realização de tarefas em formação as camadas aqui descritas devem adequar-se às necessidades específicas de hardware de cada membro, além de acomodar as suas tarefas específicas enquanto membro da formação em si.

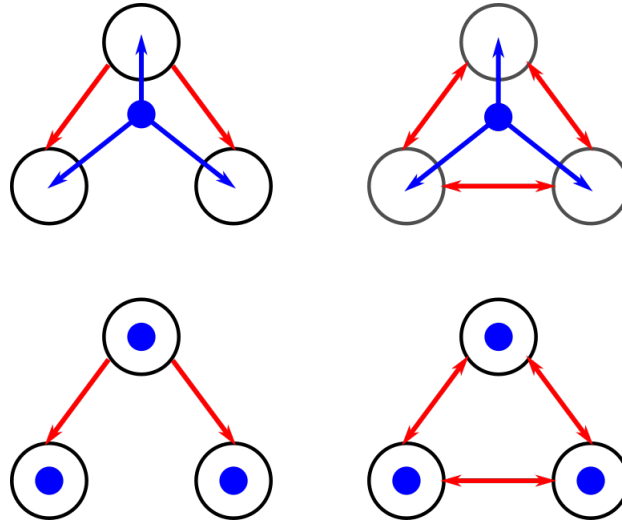


Figura 2.3: Imagem ilustrando todas as combinações possíveis de tipos de camadas de Controle e de Comportamento para Formações de Robôs Móveis. As linhas vermelhas representam o comportamento da formação, enquanto os elementos em azul representam o controle. Em sentido horário, a partir do canto superior esquerdo: controle e comportamento centralizados, controle centralizado e comportamento distribuído, controle e comportamento distribuídos e controle distribuído e comportamento centralizado.

Embora mais versáteis, formações heterogêneas são tipicamente mais difíceis de implementar, por necessitarem de ajustes individuais para cada um de seus membros.

2.4 Considerações Finais

Foram apresentados os conceitos teóricos básicos que nortearam esta pesquisa. Embora o texto contenha uma grande parte da fundamentação teórica que foi utilizada para nortear os tópicos desenvolvidos no Capítulo 4 e os resultados apresentados no Capítulo 5, é necessário frisar que eles estão apresentados aqui apenas de maneira introdutória, e que para uma compreensão mais profunda é necessário recorrer aos textos referenciados na bibliografia.

Capítulo 3

Trabalhos Relacionados

Este capítulo tem como meta definir o estado da arte nas áreas de interesse para o objeto desta dissertação. Essencialmente, trata-se de mostrar as aplicações mais recentes em termos de pesquisa para os conceitos teóricos definidos no Capítulo 2. Em tempo, o tema do modelo físico do robô não será abordado aqui, uma vez que este serve como pré-requisito para o desenvolvimento dos trabalhos em robótica, e possui uma base teórica e prática já consolidada, não sendo objeto de pesquisa por si só a não ser em trabalhos que lidem diretamente com técnicas de modelagem de sistemas, tema que foge ao escopo deste texto.

3.1 Controle Preditivo Não Linear Baseado em Modelo

O NMPC é um método de controle amplamente utilizado em diversas aplicações, principalmente por sua capacidade de lidar com sistemas não lineares e de incorporar as restrições impostas a estes sistemas no processo de definição das entradas de controle ótimas. Uma de suas principais características é a fácil adaptação do algoritmo a diversos problemas. Em [Schultze e Horn 2016] o método é utilizado para controlar a exaustão de gás em células de combustível, em [Ławryńczuk 2016] o método é utilizado em um reator de neutralização e em [Rosero et al. 2016] o algoritmo é aplicado a uma planta de secagem de sementes de cacau.

Na área de robótica móvel o NMPC tem sido adotado como uma possível solução para o problema de rastreamento de trajetória, extremamente comum para esse tipo de aplicação. Em [Chen, Wang e Wu 2017] o NMPC é utilizado para otimizar os parâmetros de controle

de um veículo aéreo não tripulado. Em [Lei, Chaofang e Na 2017] o NMPC é o responsável pelo controle da trajetória de um robô capaz de planejar uma rota para o objetivo desejado desviando de obstáculos no caminho. Em [Duijkeren et al. 2016] o objeto de estudo é um manipulador robótico, numa interpretação diferente do problema de controle de trajetória onde o caminho a ser seguido corresponde à posição desejada da ponta do manipulador. [Kayastha et al. 2017] propõem o NMPC como o algoritmo de controle a ser utilizado para robôs operando no espaço, como uma alternativa a atividades extraveiculares realizadas por astronautas em missões espaciais.

Um outro aspecto do NMPC é a possibilidade de escolher o otimizador do processo de minimização envolvido no algoritmo, descrito no Capítulo 2, de acordo com as necessidades do sistema. Um algoritmo baseado em uma dinâmica neural é utilizado em [Li et al. 2018] para o controle de uma formação de robôs não holonômicos, [Xiao e Chen 2017] emprega uma heurística de otimização baseado em um enxame de partículas e [Pei et al. 2017] utiliza um método de resíduo mínimo. Esse é um assunto de particular interesse quando se trata deste método, uma vez que a sua natureza de operação *online* faz com que haja uma grande importância sobre o tempo de execução necessário para o processo de otimização e a precisão da solução obtida. Isso leva à necessidade de implementação de heurísticas capazes de buscar valores ótimos em superfícies não convexas em tempo hábil, motivo pelo qual muitos trabalhos são focados apenas em analisar o comportamento do algoritmo quando implementado com diferentes métodos de otimização.

Em [El-Ferik, Siddiqui e Lewis 2016] é estudado um problema pouco abordado, que é a necessidade de uma rede confiável para um sistema multirrobô utilizando o NMPC, e é apresentada uma solução incorporada ao controlador para lidar com limitações de banda e eventuais atrasos de comunicações. Essa é uma área particularmente interessante, pois a construção do algoritmo não apresenta uma resistência inata a esse tipo de falha, e dada a natureza do seu funcionamento, em que o comportamento do sistema precisa ser o mais próximo possível do previsto pelo modelo, esse tipo de falha por comunicação pode ser particularmente danosa. [Mehrez, Mann e Gosine 2015] ataca o problema de estabilidade em malha fechada de um robô controlado pelo NMPC, aplicando dois métodos de estabilização do estado final desejado. Em [Jayasiri, Gros e Mann 2016] há uma extensão do conceito de predição aplicado no NMPC, utilizando também uma estimação de horizonte móvel para

prever os estados futuros do sistema enquanto o controlador fica responsável pelo rastreamento da trajetória. Em [Wang et al. 2016] é demonstrada a capacidade de utilizar-se o NMPC para um robô seguindo um humano, enquanto [Zarghami et al. 2014] lida com limitações dinâmicas do sistema. Estes dois últimos exemplos utilizam robôs omnidirecionais, em contraste com a maioria dos demais trabalhos citados, pois o objeto da pesquisa aqui apresentada lida necessariamente com robôs não omnidirecionais graças a sua natureza não holonômica, então foi dada a preferência no processo de coleta de referências a trabalhos que lidem com as mesmas condições. Essa diferenciação fica restrita, entretanto, ao modelo de predição e às limitações do sistema, uma vez que o algoritmo em si não é afetado pela natureza do robô.

Para todos os casos citados acima, é importante notar que uma limitação típica do NMPC é geralmente ignorada, que é a sua dificuldade de operar em tempo hábil em um sistema de malha fechada, advinda dos termos não lineares complexos e incertezas envolvidas no rastreamento de trajetória. Assim sendo, é comum que os parâmetros do controlador sejam otimizados com relação a uma trajetória específica, uma vez que esse processo é muito mais simples, sendo normalmente realizado de modo empírico, do que tentar reduzir a complexidade computacional inerente ao problema. Consequentemente, o controlador passa a ter suas aplicações restritas, pois é necessário re-sintonizar os seus ganhos para cada trajetória possível. Esse é o motivo pelo qual a abordagem escolhida neste trabalho foi a apresentada em [Nascimento, Dórea e Gonçalves 2018]. Neste artigo, é apresentada uma modificação do NMPC clássico, em que, em vez de o controlador tentar minimizar o erro entre o estado do robô e uma trajetória alvo, é utilizado o estado de um líder virtual como alvo. Essa abordagem apresenta dois benefícios significativos.

O primeiro é que é eliminada a necessidade de um bloco gerador de mini trajetórias, utilizado como uma maneira de sanar o comportamento em malha fechada descrito no parágrafo anterior. Além de simplificar o sistema, isso também resulta em uma redução do tempo exigido para a operação do controlador, uma vez que essa etapa de geração de trajetórias é eliminada. O segundo é que as modificações realizadas permitem que o mesmo conjunto de ganhos do controlador seja utilizado para diferentes trajetórias, o que é um ganho significativo em termos da flexibilidade de operação do sistema. Outro fator para o uso desta versão do NMPC é o fato de que a sua estrutura é particularmente favorável a uma implementação

com múltiplos robôs, uma vez o algoritmo simula uma situação do tipo líder-seguidor como parte do processo de controle. Sendo assim, é possível incorporar essa estratégia a uma formação de robôs meramente substituindo o líder virtual por um líder real.

3.2 Controle de Formações de Robôs

Esta seção aborda pesquisas na área de controle de formações de robôs. Como este trabalho adota uma estrutura do tipo líder-seguidor, foi dada ênfase a esse tipo de estrutura. Primeiramente, é importante notar que o tipo de estrutura organizacional para controle de formação descrita na Seção 2.3 possui um caráter genérico, podendo ser aplicada em diversas áreas, como pode ser visto em [Cook, Ray e Lever 2018], que utiliza a estrutura para o controle de comboios de tratores. Entretanto, essa seção irá se concentrar em aplicações na área de robôs móveis autônomos.

Em [Ma, Liu e Chen 2016] é proposta uma variação da estrutura líder-seguidor, onde o líder possui um comportamento inteligente baseado no consenso das informações fornecidas pelos seguidores. [Maghenem, Loria e Panteley 2018] lida com o problema de uma formação rastreando uma trajetória com incertezas, enquanto [Xing e Deng 2018] utiliza uma estratégia de controle baseada em um observador para um sistema líder-seguidor com múltiplos agentes. [Lin 2018] analisa o desempenho desta estrutura em sistemas que operam em redes direcionais.

Para formações do tipo líder-seguidor, uma questão que surge constantemente é o problema de seleção do líder, seja para otimizar o desempenho da formação, seja porque uma mudança de parâmetros, do ambiente ou não, faz com que seja necessário que um novo líder seja escolhido. Para isso, diversas heurísticas são propostas para escolher um líder ótimo. [Saeidi, Mikulski e Wang 2017] propõe um algoritmo baseado em confiança para a seleção de líder em formações em que haja iterações humano-robô. Em [Feng, Lu e Chen 2015] e [Li et al. 2017], uma abordagem similar é adotada, agregando uma medida de insatisfação aos membros da formação, que supervisionam o desempenho do líder. Um algoritmo baseado no mecanismo de regulação neuroendócrino do ser humano é proposto como uma solução para esse problema em [Li, Ding e Hao 2017]. Um operador de seleção por menor redução absoluta é utilizado em [Sakurama 2017] para definir múltiplos líderes e uma formação su-

jeita a atrasos de transmissão heterogêneos. [Karpov e Karpova 2015] utiliza uma seleção dinâmica baseada em um sistema de votos combinada com uma estratégia distribuída para atingir um melhor desempenho.

Embora as estratégias apresentadas acima representem alguns dos mais recentes avanços no que tange à seleção de líder de maneira dinâmica em formações de robôs, não se pode negligenciar mencionar alguns fatores que põem em cheque as contribuições apresentadas por estes trabalhos. A primeira, e mais importante, é a prevalência quase que total de resultados meramente simulados. Esse é um fator preocupante por vários motivos, mas o principal é o simples fato de que qualquer sistema simulado é, por natureza, dependente de um ou mais modelos que não representam o sistema real com fidelidade absoluta. Assim sendo, o nível de confiança desses trabalhos acerca da capacidade dos métodos desenvolvidos de apresentar em um ambiente prático o mesmo desempenho obtido em um ambiente simulado pode se mostrar injustificado quando considerados fatores que se apresentam apenas quando da implementação do sistema em condições reais, em que normalmente são verificadas limitações e características previamente ignoradas

Um segundo ponto a ser considerado acerca da natureza simulada dos trabalhos mencionados é o fato de que os algoritmos propostos são constantemente processados ao longo da operação do sistema. Isso significa que uma carga computacional a mais é adicionada ao sistema, sobre a carga já presente, que corresponde ao controlador de formação, controlador de trajetória para os casos relevantes, processamento de dados coletados por sensores, etc. Esse fato levanta a questão sobre a capacidade de um sistema embarcado, como são os robôs móveis, ser capaz de rodar esses algoritmos com o desempenho desejado, além de realizar todas as demais funções necessárias. A estrutura de seleção proposta no Capítulo 4 possui uma abordagem bastante simplificada quando comparada às alternativas aqui apresentadas, mas tem como vantagem o fato de que a maior parte do processamento necessário para o seu funcionamento é realizado antes da operação do sistema, transferindo a complexidade do algoritmo do processador para a memória, um recurso tipicamente mais disponível nessa espécie de sistema. Ainda, o seu desempenho foi comprovado experimentalmente, como será mostrado no Capítulo 5, e é explicado que o impacto computacional sobre o desempenho do sistema é mínimo.

3.3 Considerações Finais

O capítulo apresentou uma pesquisa acerca dos trabalhos mais recentes que foram considerados relevantes dentro das áreas de controle de formações de robôs e utilização do NMPC. Assim como toda pesquisa de estado da arte, as observações feitas aqui discorrem acerca de um campo em constante evolução, e que possui muito mais trabalhos sendo publicados do que é possível mencionar em um único capítulo, ou mesmo dissertação. Assim sendo, as publicações referenciadas aqui devem ser consideradas como um ponto de partida para uma pesquisa mais aprofundada dentro das publicações da área.

Capítulo 4

Non linear Model Predictive Control

Modificado para Formações de Robôs e Seleção Dinâmica de Líder

Neste capítulo são apresentados os conceitos e algoritmos desenvolvidos ao longo da pesquisa. Como mencionado previamente, é apresentada a variante do NMPC proposta em [Nascimento, Dórea e Gonçalves 2018] adaptada a formações de robôs, incluindo a capacidade de unir as camadas de controle e de comportamento no controle da formação. Também são introduzidas a estrutura baseada em árvore, utilizada para definir o líder da formação de forma dinâmica, e a arquitetura do sistema de controle.

4.1 NMPC Modificado para Controle de Formação

Esta seção visa elaborar os diversos aspectos do algoritmo NMPC utilizado no controle dos robôs utilizados na formação. Serão explorados o preditor e os termos da função custo utilizados, o método de otimização empregado para encontrar o sinal de controle ótimo e o espaço dentro do qual essa solução deverá ser encontrada. Considerações mais práticas acerca do funcionamento do algoritmo, como número máximo de iterações do algoritmo de otimização e critério de parada serão feitas no Capítulo 5.

4.1.1 Otimizador e Preditor

O processo do NMPC descrito na Seção 2.2 descreve uma abordagem genérica do algoritmo de controle, entretanto, para a sua implementação, é necessário analisar o papel do modelo de predição e da função custo dentro do algoritmo. Enquanto o modelo de predição do sistema é relativamente independente dos demais elementos do algoritmo, a função custo é dependente de dois outros elementos: o otimizador e o preditor. O otimizador é de entendimento mais imediato, sendo apresentado na Figura 2.2, e tem como função definir as entradas de controle ótimo para o sistema a partir da função custo fazendo uso de um processo de minimização.

Por sua vez, o preditor é responsável por realizar a ponte entre o modelo de predição e a função custo. De acordo com o que se deseja do sistema, é necessário realizar-se predições de grandezas que vão além daquelas fornecidas pela predição feita para apenas um robô, sendo necessário calcular o comportamento futuro de colegas de formação para, por exemplo, estimar a distância entre eles. Assim, o bloco preditor é o responsável por construir essa predição do estado do sistema no futuro para que possam ser obtidos todos os fatores necessários para a construção da função custo. Para isso, é utilizado o modelo apresentado na Seção 2.1, partindo do pressuposto de que a velocidade do alvo e dos colegas de formação é constante e igual ao último valor obtido durante o laço de controle.

O bloco preditor aqui utilizado, então, prevê as distâncias entre todos os robôs da formação até o horizonte de controle. É importante notar que, por causa da abordagem adotada, em que robôs que param de funcionar são tratados como obstáculos, o cálculo da distância com relação a esses robôs permanece relevante, não se tornando um apenas um ônus computacional sobre o algoritmo a medida em que os robôs mudam o seu comportamento. A Figura 4.1 mostra esse processo de predição, e expande sobre a Figura 2.2, mostrando que a noção de modelo genérico utilizado para a construção da função custo, introduzida anteriormente, é expandida para a implementação na formação de robôs, onde é necessário realizar a predição não só do estado futuro de cada robô mas da formação como um todo.

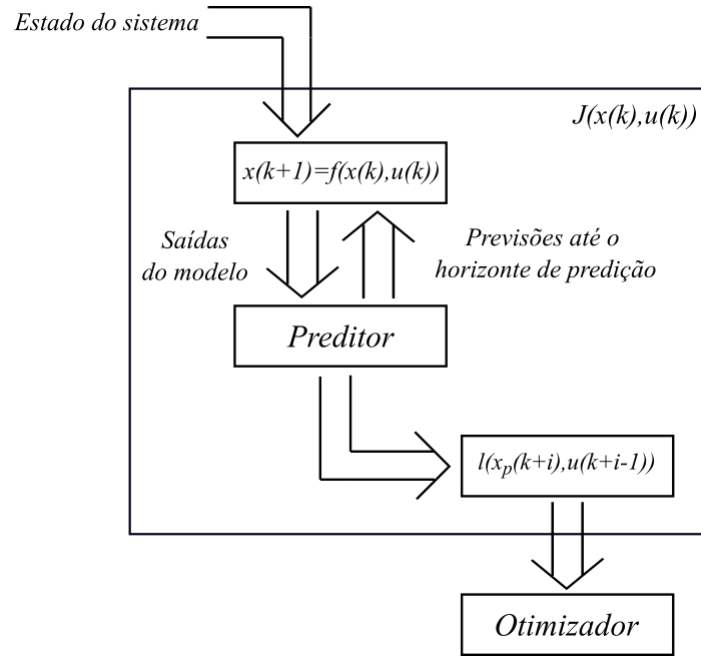


Figura 4.1: Diagrama mostrando o processo de construção da função custo a partir do bloco preditor e do modelo do robô.

4.1.2 Função Custo

Como foi mencionado no Capítulo 2, uma das partes mais importantes do NMPC é a elaboração de uma função custo apropriada para o problema em questão. Uma vez que os sinais de controle são determinados a partir da minimização desta função, a sua forma molda a saída de controle e consequentemente determina o comportamento do sistema. Isso significa que, se bem escolhidos, os termos dessa função podem influenciar o sistema de maneira a agregar ao controlador capacidades além do rastreamento de uma trajetória ou manutenção de uma distância desejada de um companheiro de formação. Essa propriedade é explorada aqui para modificar o comportamento dos membros da formação dentro do controlador, agregando os termos relevantes à função elaborada e manipulando apenas os ganhos de cada um dos termos da função. Isso é possível graças às modificações propostas em [Nascimento, Dórea e Gonçalves 2018], que foram implementadas aqui e permitem que os valores dos ganhos sejam conservados para diferentes comportamentos.

É importante frisar que a estrutura da formação aqui utilizada é do tipo líder-seguidor, e que essa escolha pauta as decisões tomadas acerca dos termos analisados a seguir. Também é necessário mencionar que a escolha da forma matemática dos termos reflete a necessidade

de um comportamento particular desejado para a função custo, sendo a forma utilizada geralmente definida de maneira empírica, embora haja conceitos matemáticos que são utilizados de maneira recorrente na literatura, como a norma euclidiana. Assim, o primeiro termo da função pode ser visto na Equação 4.1, em que é penalizado o erro entre a posição do robô e a posição de um alvo virtual fornecida pelo gerador de trajetória para o caso em que o robô é o líder, ou a posição do líder caso o robô seja um seguidor.

$$J_1 = \sum_{j=N_1}^{N_p} \lambda_1 |(D_t - ||P_t^{R_n}(k+j)||)| \quad (4.1)$$

onde

$$||P_t^{R_n}(k+j)|| = \sqrt{(x_t(k+j) - x(k+j))^2 + (y_t(k+j) - y(k+j))^2} \quad (4.2)$$

Primeiramente, é necessário explicar que na notação adotada $|\cdot|$ representa o módulo de argumentos vetoriais e o valor absoluto para argumentos escalares, enquanto $||\cdot||$ representa a norma euclidiana. N_1 e N_p são os limites superior e inferior do horizonte de predição, respectivamente, sendo o valor do horizonte definido no Capítulo 2 dado por $n = N_p - N_1$. D_t é a distância desejada entre o líder e o seguidor, enquanto $P_t^{R_n}$ denota a diferença de pose entre o robô e o alvo, apresentada na Equação 4.2, onde (x_t, y_t) é a posição do líder, seja ele virtual ou real, e (x, y) é a posição do robô que está sendo controlado. Uma variável de particular importância é λ_1 , que representa o ganho deste termo dentro da função custo. Cada um dos termos possui um destes ganhos, e é justamente a definição desses valores que permite que o controlador seja sintonizado para que o comportamento desejado seja atingido.

O segundo termo da função custo é destinado a penalizar a diferença de orientação entre a referência e o robô, e está descrito na Equação 4.3.

$$J_2 = \sum_{j=N_1}^{N_p} \lambda_2 (\theta_t(k+j) - \theta(k+j))^2 \quad (4.3)$$

θ_t representa a orientação do líder e θ representa a orientação do robô controlado. Entre os termos J_1 e J_2 são cobertos os aspectos necessários para o rastreamento de uma trajetória desejada, que são a penalização do erro de trajetória e de orientação. Em conjunto com o termo que penaliza o esforço de controle, que será mencionado posteriormente, já é possível elaborar uma função custo apropriada para o NMPC. Entretanto, essa função lida com um

sistema envolvendo apenas um robô, mesmo que este utilize um líder virtual. Assim sendo, para que o método possa ser aplicado a todos os membros de uma formação de robôs é necessária a inclusão de termos adicionais que penalizem fatores específicos a essa situação. O primeiro destes termos pode ser visto na Equação 4.4, utilizado para manter os membros da formação a uma distância desejada uns dos outros.

$$J_3 = \sum_{j=N_1}^{N_p} \sum_{i=1}^{N_{mates}} \lambda_{3i} \max \left(1 - \frac{\|P_{R_i}^{R_n}(k+j)\|}{D_i}, 0 \right) \quad (4.4)$$

onde

$$\|P_{R_i}^{R_n}(k+j)\| = \sqrt{(x_i(k+j) - x(k+j))^2 + (y_i(k+j) - y(k+j))^2} \quad (4.5)$$

Primeiramente, $\max(\cdot, \cdot)$ é uma função que assume o valor de seu maior argumento. D_i representa a distância desejada entre os companheiros de formação e (x_i, y_i) representa a posição de um robô companheiro. Esse termo é calculado para cada outro robô na formação fora o líder. Essencialmente, o termo penaliza o robô toda vez que o seu erro de posição relativo ao companheiro em questão desvia do valor desejado, normalizando esse erro com relação ao valor desejado e calculando o desvio com relação a meta $\|P_{R_i}^{R_n}(k+j)\| = D_i$.

O próximo termo destina-se a corrigir uma perturbação gerada como uma consequência da capacidade de troca dinâmica de líder. Como também será mencionado no Capítulo 5, a situação suposta para a troca de líder é a detecção de uma falha de operação no líder atual. Como o tópico de detecção de falhas em sistemas autônomos é uma área ampla e objeto de múltiplas pesquisas, foi adotada a suposição de que a falha é representada pelo líder não sendo capaz de receber, ou não sendo capaz de seguir, os sinais de controle fornecidos pelo controlador. Supondo que até então o líder estivesse rastreando a trajetória de maneira apropriada, o robô defeituoso ficará sobre a trajetória que o novo líder e os demais robôs deverão seguir. Isso implica que os robôs devem ser capazes, então, de desviar deste obstáculo, e, como a proposta desta dissertação é que o comportamento dos robôs seja definido no nível do controlador, é necessário que esta capacidade seja incorporada ao NMPC, o que resulta no termo apresentado na Equação 4.6.

$$J_4 = \sum_{j=N_1}^{N_p} \sum_{i=1}^{N_{mates}} \lambda_{4i} \left| \log \left(\frac{1}{\|P_{R_i}^{R_n}(k+j)\|} \right) \right| \quad (4.6)$$

Assim como J_3 , esse termo é calculado com relação a múltiplos robôs, com a diferença de que em vez de serem os companheiros de formação são líderes anteriores que pararam de funcionar. Ainda, é importante notar que ao contrário do termo J_3 , J_4 é calculado tanto para o líder como para os seguidores da formação, uma vez que ambos precisam desviar de líderes anteriores.

Por fim, o termo J_5 , descrito na Equação 4.7, penaliza o esforço de controle do sistema, sendo N_u o horizonte de controle.

$$J_5 = \sum_{j=N_1}^{N_U} \lambda_5 |\Delta U(k+j-1)| \quad (4.7)$$

com

$$\Delta U(k) = \begin{bmatrix} \Delta v_c(k) \\ \Delta \omega_c(k) \end{bmatrix} \quad (4.8)$$

O vetor de controle acima foi definido como uma consequência das características do controle da plataforma Turtlebot 2 e da sua integração com o ROS, que realiza o controle do sistema apenas com valores absolutos de v e ω . Por fim, a função custo do sistema é dada pela soma de todos os termos apresentados acima, e pode ser vista na Equação 4.9. $\hat{R}_p$ indica a pose do robô, dada pelas suas variáveis de posição e orientação.

$$J(\hat{R}_p, \Delta U) = \sum_{j=N_1}^{N_p} \lambda_1 |(D_t - \|P_t^{R_n}(k+j)\|)| + \quad (4.9a)$$

$$\sum_{j=N_1}^{N_p} \lambda_2 (\theta_t(k+j) - \theta(k+j))^2 + \quad (4.9b)$$

$$\sum_{j=N_1}^{N_p} \sum_{i=1}^{N_{mates}} \lambda_{3i} \max \left(\left| 1 - \frac{\|P_{R_i}^{R_n}(k+j)\|}{D_i} \right|, 0 \right) + \quad (4.9c)$$

$$\sum_{j=N_1}^{N_p} \sum_{i=1}^{N_{mates}} \lambda_{4i} \left| \log \left(\frac{1}{\|P_{R_i}^{R_n}(k+j)\|} \right) \right| + \quad (4.9d)$$

$$\sum_{j=1}^{N_U} \lambda_5 |\Delta U(k+j-1)| \quad (4.9e)$$

4.1.3 *Resilient Propagation*

Uma parte integral do NMPC é o algoritmo de otimização empregado. Para este trabalho, foi escolhido o *Resilient Propagation*(RPROP), inicialmente proposto em [Riedmiller e Braun 1993] e tipicamente utilizado em aplicações de redes neurais. Para sanar os problemas de um método de gradiente descendente, é realizada uma atualização dos pesos utilizados ao longo da busca de acordo com a função de erro do sistema. Como foi posto em [Nascimento, Dórea e Gonçalves 2018], o processo de adaptação dos pesos do RPROP não é afetado pela influência imprevisível do valor da derivada, dependendo apenas do comportamento do seu sinal ao longo do tempo, o que a torna uma técnica apropriada para uso em conjunto com o NMPC.

O Algoritmo 4.1.3 mostra o pseudocódigo do RPROP. Um peso variante com o tempo Δ_i é introduzido, com o objetivo de determinar qual será o valor da atualização do peso. Esse valor é alterado ao longo do processo de aprendizado de acordo com a função de erro $f(\cdot)$. Caso seja detectada uma mudança no sinal da derivada parcial do peso correspondente $w_i(t)$, o que implica que a última atualização foi grande demais e o algoritmo ignorou um mínimo local, o valor de Δ_i é diminuído por um fator η^- . Caso não haja mudança de sinal, o valor aumenta ligeiramente, de modo a acelerar a convergência em regiões rasas. Com o valor de atualização adaptado para cada peso, a atualização do valor de Δw_i segue a seguinte regra: se a derivada é positiva, o que indica um erro crescente, o peso é diminuído de acordo com o valor de atualização; caso o erro seja negativo, o valor de atualização cresce de acordo com o valor de atualização; e caso haja uma mudança no sinal da derivada parcial a última atualização de Δw_i é desfeita.

Essa mudança de sinal implica outra mudança de sinal no próximo passo, o que gera uma penalização não planejada sobre o valor de atualização. Assim, o valor de $\frac{\delta f}{\delta w_i}(t)$ é definido como zero para evitar que haja uma adaptação no próximo passo.

4.1.4 *Análise das Restrições*

Uma das características que torna o NMPC tão apropriado para lidar com o problema de rastreo de trajetória para robôs móveis é a sua capacidade de lidar com restrições inerentes ao sistema, como a restrição não holonômica apresentada na Equação 2.1. A maneira como

Algorithm 1 Algoritmo descrevendo o processo de ajuste dos pesos do RPROP

```

1: for  $i = 0$  até  $N_i$  do
2:   if  $\frac{\delta f}{\delta w_i}(t - I) \cdot \frac{\delta f}{\delta w_i}(t) > 0$  then
3:      $\Delta_i(t) = \min(\Delta_i(t - I) * \eta^+, \Delta_{max})$ 
4:      $\Delta w_i(t) = -\text{sign}(\frac{\delta f}{\delta w_i}(t)) * \Delta_i(t)$ 
5:      $w_i(t + I) = w_i(t) + \Delta w_i(t)$ 
6:   else if  $\frac{\delta f}{\delta w_i}(t - I) \cdot \frac{\delta f}{\delta w_i}(t) < 0$  then
7:      $\Delta_i(t) = \min(\Delta_i(t - I) * \eta^-, \Delta_{min})$ 
8:      $w_i(t + I) = w_i(t) - \Delta w_i(t - I)$ 
9:      $\frac{\delta f}{\delta w_i}(t) = 0$ 
10:  else if  $\frac{\delta f}{\delta w_i}(t - I) \cdot \frac{\delta f}{\delta w_i}(t) = 0$  then
11:     $\Delta w_i(t) = -\text{sign}(\frac{\delta f}{\delta w_i}(t)) * \Delta_i(t)$ 
12:     $w_i(t + I) = w_i(t) + \Delta w_i(t)$ 
13:  end if
14: end for

```

o algoritmo lida com essas restrições é limitando o espaço sobre o qual o algoritmo de otimização pode buscar soluções ótimas para o problema de controle, definindo limites para a área de busca.

Assim, é possível realizar uma breve análise acerca do espaço sobre o qual será feita a busca das soluções a partir das restrições do sistema. Além da restrição não holonômica, o sistema também possui restrições de velocidades linear e angular máximas. Como será mencionado novamente no Capítulo 5, a plataforma Turtlebot 2 possui uma velocidade linear máxima de $v = 0.7m/s$ e uma velocidade angular máxima $\omega = \pi rad/s$.

Com a restrição de velocidade linear é possível definir os pares de velocidades ao longo dos eixos x e y que resultam em valores válidos, como pode ser visto na Figura 4.2.

De posse dos dados apresentados na Figura 4.2, é possível definir quais valores de v e ω satisfazem a restrição da Equação 2.1. Como o valor resultante é uma função de três variáveis, é possível visualizar apenas uma projeção de parte da superfície resultante desta análise. Assim, a Figura 4.3 mostra a composição de múltiplas projeções obtidas para diferentes valores de v e ω . Dada a natureza da projeção, é importante notar que os pontos amarelos representam que o v em questão é válido para pelo menos um valor de ω , e não que este

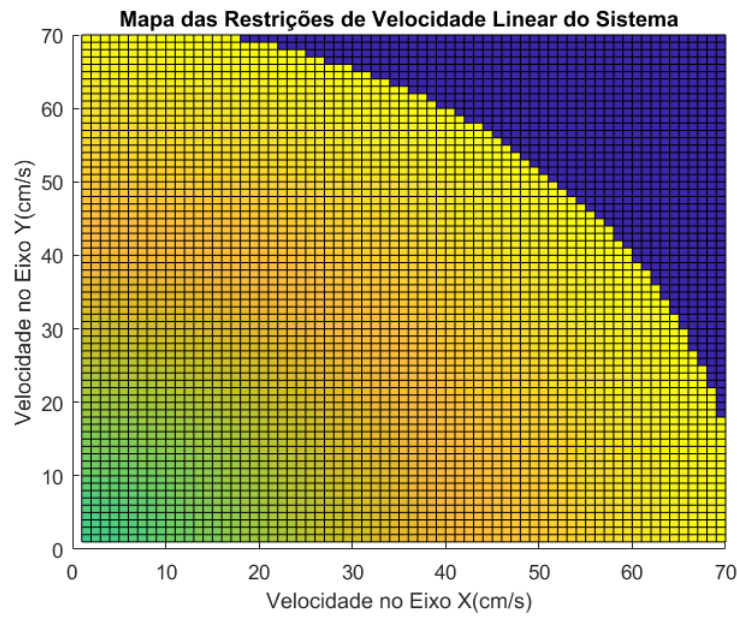


Figura 4.2: Mapa da restrição de velocidade linear da plataforma Turtlebot 2. Áreas em azul indicam regiões fora das capacidades do sistema.

valor é válido por si só. Esse mapa de restrições restringe a área de busca do RPROP quando este minimiza a função custo em busca dos valores ótimos para as entradas de controle.

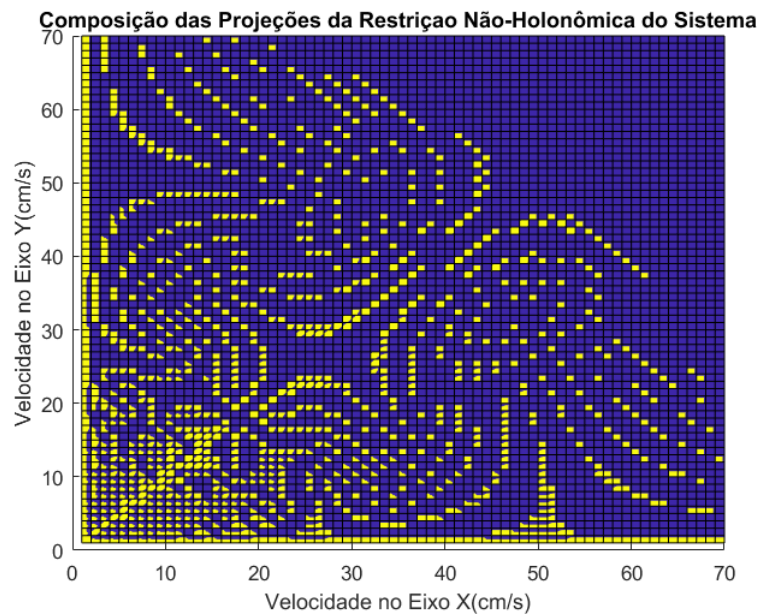


Figura 4.3: Mapa dos valores de v_x e v_y que satisfazem à Equação 2.1 para algum ω , representados pelos pontos em amarelo.

4.2 Estrutura para Seleção de Líder Dinâmico

Como foi mostrado no Capítulo 3, a área de algoritmos para seleção de líder em formações de robô tem sido o objeto de muitos trabalhos na área de robótica. Também foi mencionado no mesmo capítulo que vários desses trabalhos se baseiam em algoritmos de complexidade elevada e se restringem a provar a sua efetividade em ambientes simulados. O uso de métodos de seleção complexos se prova um desafio, uma vez que a implementação de operações complexas em um robô necessariamente lida com as restrições de um sistema embarcado, que necessitam dividir uma capacidade de processamento reduzida entre as múltiplas tarefas necessárias para a operação do robô, como receber e tratar dados de sensores e enviar sinais de controle para os motores.

Como uma maneira de contornar estes problemas, é proposta a estrutura apresentada na Figura 4.4, baseada em uma forma de árvore tipicamente utilizada em estruturas de dados, para definir o líder da formação.

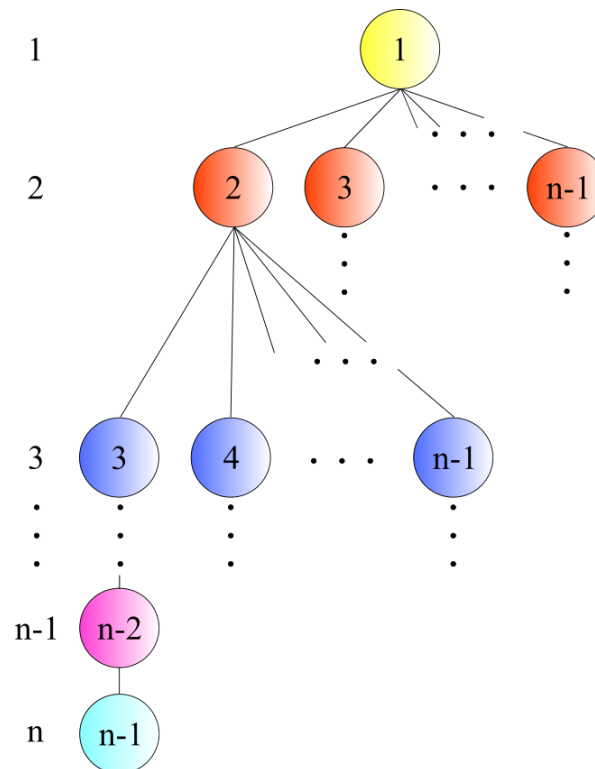


Figura 4.4: Estrutura em árvore proposta para uma formação com n robôs.

O funcionamento da estrutura é simples: supondo uma formação de n robôs que comece com um deles atuando como líder, caso o líder pare de funcionar por qualquer motivo, a

formação deve se rearranjar de maneira a poder continuar a realizar a tarefa estipulada, que é o rastreamento de trajetória para o nosso caso. Então, há $n - 1$ líderes possíveis caso ocorra uma primeira falha, $n - 2$ para a segunda, e assim por diante. A partir desse pressuposto, uma máquina de estados finitos é construída baseada na árvore do sistema, escolhendo-se uma condição apropriada para que seja realizada a transição entre estados, que no nosso caso é a detecção de uma falha pelo líder. O próximo estado, ou líder, é definido a partir de um critério arbitrário, que pode ser ajustado de acordo com o comportamento desejado, como a proximidade entre o robô e o líder anterior ou a proximidade entre o robô e o líder virtual que o líder seguia.

Embora essa abordagem possa parecer exageradamente simples, ela possui algumas vantagens inerentes que a tornam bastante apropriada para esse tipo de aplicação. A primeira, e mais importante, é a transferência da carga computacional exigida do sistema do momento de operação para um momento *a priori*. Como a máquina de estados na forma da Figura 4.4 é construída antes da operação do sistema, e não é dependente de fatores operacionais, uma vez que o número de membros da formação e o líder inicial são conhecidos, o único esforço de processamento exigido do sistema é que ele navegue entre os estados da máquina, navegação esta que só adota um sentido, sempre indo de nós pais para nós filhos. Não só isso, os critérios que decidem qual será o próximo líder, ou estado da máquina, podem variar de acordo com a necessidade, mas essencialmente se resumem à verificação de expressões condicionais, um processo computacionalmente barato.

Uma segunda vantagem é o fato de que, por ser simples, o processo de troca de líder ocorre da maneira rápida, o que significa que o restabelecimento da formação também ocorre de maneira rápida. Por fim, algumas considerações acerca do crescimento da estrutura a medida que o número de robôs aumenta devem ser feitas. A Equação 4.10 mostra a expressão para o número de nós por nível da árvore para uma formação de n robôs, enquanto a Equação

4.11 mostra a expressão que calcula do número total de nós na árvore.

$$\begin{aligned}
 N_{nodes}(1) &= 1 \\
 N_{nodes}(2) &= (n - 1) \\
 N_{nodes}(3) &= (n - 1)(n - 2) \\
 &\vdots \\
 \Rightarrow N_{nodes}(k) &= \frac{(n - 1)!}{(n - k)!}, k \in \mathbb{N}^*
 \end{aligned} \tag{4.10}$$

$$N_{total} = \sum_{k=1}^n N_{nodes}(k) \tag{4.11}$$

A partir dessas equações, é visível que o tamanho da estrutura aumenta por um fator de $n!$, o que é tipicamente preocupante. Entretanto, para o caso aqui estudado, é possível contornar esse fator ao explorar as especificidades da aplicação aqui estudada. Formações do tipo líder-seguidor, principalmente quando lidando com o problema de rastreamento de trajetória, tipicamente possuem poucos membros, sendo arquiteturas distribuídas mais utilizadas para aplicações que envolvam enxames de robôs. Caso o número de robôs seja restrito, fica claro que o fator de crescimento da árvore não se torna um problema, o que faz com que esse método de escolha dinâmica de líder seja apropriado para a aplicação em questão.

Outra questão se refere ao que a mudança de estados em si implica para o sistema. Como foi dito na Seção 4.1.2, o comportamento de um robô na formação é determinado a partir do conjunto dos ganhos de sua função custo. Assim, os diferentes estados da estrutura de seleção de líder nada mais são do que conjuntos de ganhos que representam os comportamentos apropriados para cada robô da formação. A questão é que, dada a natureza da estrutura, há uma propriedade complementar entre os termos. Mais uma vez, supondo uma formação de n robôs, de acordo com a Figura 4.4, temos uma árvore com n níveis. Tomando um nível k qualquer, sabemos que naquele nível de n robôs totais nós temos $n - k - 1$ robôs que eram líderes antes e pararam de funcionar, e $k + 1$ robôs restantes na formação. Considerando que os líderes anteriores não precisam mais ser controlados, pois já não fazem mais parte da formação, isso significa que cada nó desse nível corresponde a $k + 1$ conjuntos de ganhos. Seja o conjunto de ganhos para o líder dado por um vetor $\mathbf{L}$ e o conjunto para cada um dos seus seguidores dado por $\mathbf{S}_i$, temos na Equação 4.12 os vetores possíveis para os nós do nível

k , supondo que até então a árvore sempre tomou o caminho mais à esquerda.

$$\begin{aligned}
 \mathbf{G}_{set}^k &= [\mathbf{L}, \mathbf{S}_{k+1}, \mathbf{S}_{k+2}, \dots, \mathbf{S}_{n-1}] \\
 \mathbf{G}_{set}^{k+1} &= [\mathbf{S}_k, \mathbf{L}, \mathbf{S}_{k+2}, \dots, \mathbf{S}_{n-1}] \\
 \mathbf{G}_{set}^{k+2} &= [\mathbf{S}_k, \mathbf{S}_{k+1}, \mathbf{L}, \dots, \mathbf{S}_{n-1}] \\
 &\vdots \\
 \mathbf{G}_{set}^{n-1} &= [\mathbf{S}_k, \mathbf{S}_{k+1}, \mathbf{S}_{k+2}, \dots, \mathbf{L}]
 \end{aligned} \tag{4.12}$$

É evidente que os vetores apresentados acima são meramente o complemento um dos outros com respeito a qual robô assume a posição de líder. Assim, ao determinar-se o conjunto de ganhos apropriados para um estado, automaticamente são determinados os conjuntos de ganhos para todos os irmãos daquele estado, e por extensão para os demais nós daquele nível da árvore, uma vez que o processo é o mesmo, alterando-se apenas os robôs ainda disponíveis para a posição de líder. Essa propriedade elimina outra complexidade inerente à estrutura de árvore, que é a suposição de que a informação contida em cada nó é independente da informação contida nos demais.

A última parte desse processo a ser analisada é o critério de decisão que será utilizado para decidir para qual estado o sistema irá se deslocar caso o líder atual sofra uma falha. O Algoritmo 2 representa a abordagem utilizada, onde R é o conjunto dos robôs restantes e A é o líder atual. Esse algoritmo é baseado na proximidade, sendo o robô mais próximo do líder no momento em que ocorre o defeito escolhido para assumir a sua posição.

Algorithm 2 Critério de decisão por proximidade de posição.

```

1:  $i = 1$ 
2: if  $R_i \rightarrow \min \left( \sqrt{(x_A - x_i)^2 + (y_A - y_i)^2} \right) \forall R_i \in R$  then
3:    $NextState = R_i$ 
4: else
5:    $i = i + 1$ 
6: go to 2
7: end if

```

4.3 Arquitetura de Controle

A junção dos conceitos apresentados anteriormente nos permite analisar a arquitetura do sistema de controle utilizada por uma perspectiva mais abrangente. O esquema da arquitetura está representado na Figura 4.5.

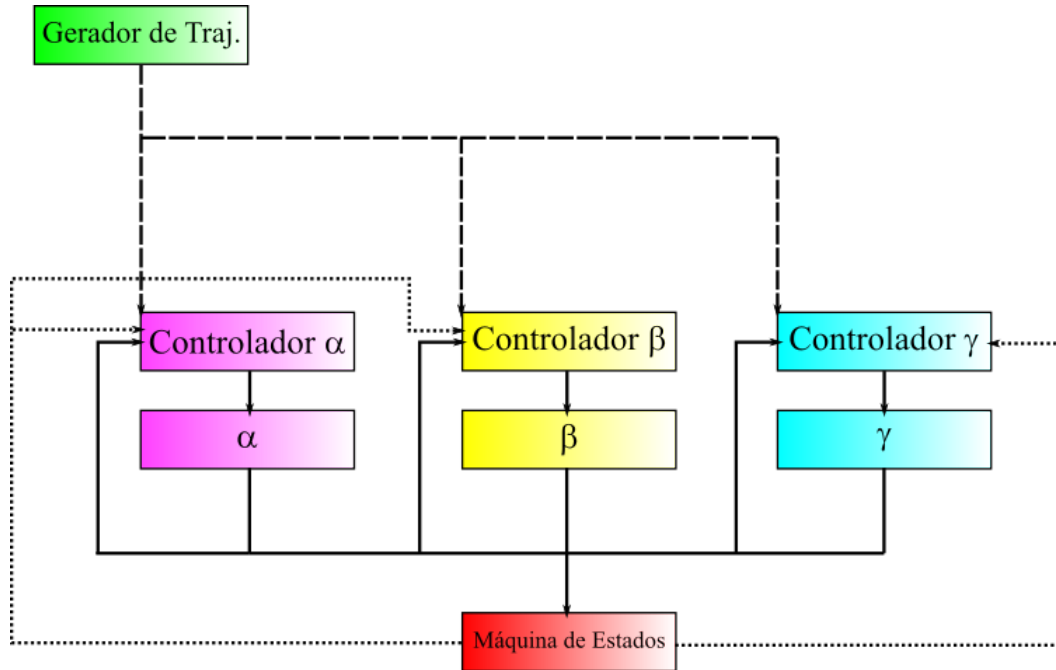


Figura 4.5: Estrutura em árvore proposta para uma formação com n robôs.

Foi utilizada uma formação com três robôs, nomeados α , β e γ . α sempre é considerado como o líder inicial. Como o controle é distribuído, há um controlador correspondente a cada um dos membros da formação. Todos eles têm acesso às informações de saída dos demais, justamente para poder alimentar os termos J_3 e J_4 da função de custo apresentada na Equação 4.9. O gerador de trajetória fornece a rota a ser rastreada, que é interpretada pelo controlador como a posição de um líder virtual a ser seguido, no caso do líder. Os demais membros da formação ignoram a entrada do gerador e seguem a posição de saída do líder de formação. Por fim, a máquina de estados descrita na Seção 4.2 determina quais os conjuntos de ganhos dos controladores de cada robô a partir dos dados de saída.

Esse diagrama de blocos é análogo ao sistema construído no ambiente Simulink. A Figura 4.6 apresenta esses blocos. As áreas em azul, laranja e amarelo representam os robôs, enquanto a área roxa contém os blocos correspondentes ao gerador de trajetória e a máquina

de estados. Esses blocos trocam entre si informações sobre a pose dos robôs, ou seja, posição e orientação, assim como os dados de posição e velocidade da trajetória desejada e os sinais responsáveis por sinalizar a mudança de comportamento para os controladores e a correspondente troca de ganhos.

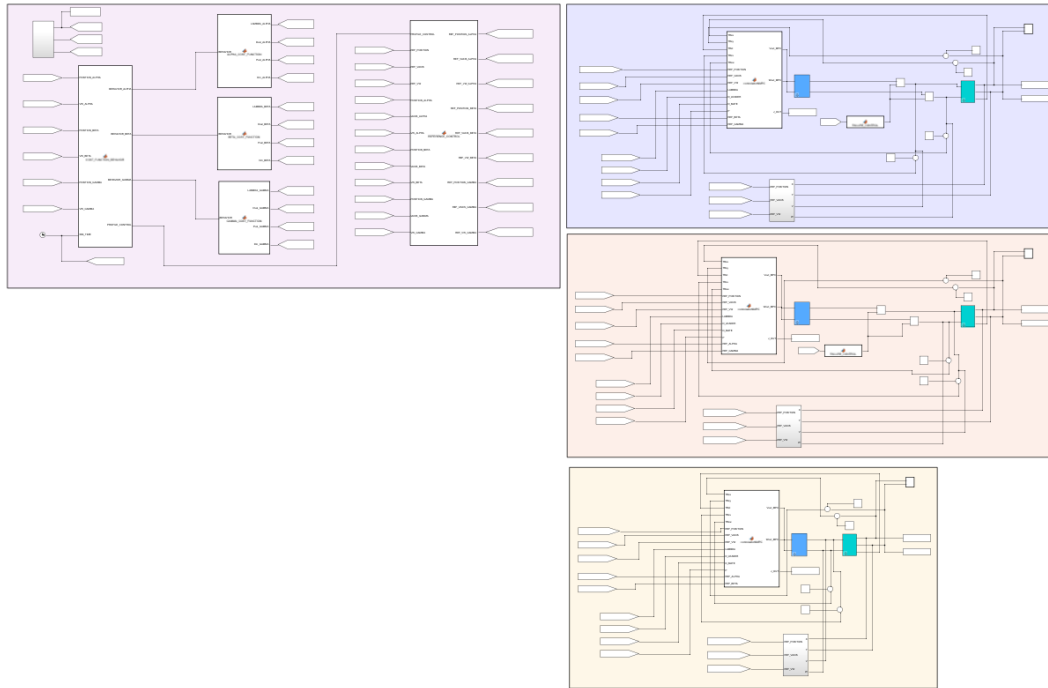


Figura 4.6: Sistema implementado no Simulink para as simulações.

4.4 Considerações Finais

Foram explicados os conceitos propostos, e apresentada a fundamentação por trás das escolhas realizadas. Não obstante, há áreas onde é possível realizar melhoras ou utilizar abordagens diferentes, assunto que será tratado no Capítulo 6, ao listar possíveis trabalhos futuros.

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos a partir de simulações e experimentos realizados com os conceitos apresentados no Capítulo 4. São explicados tanto o ambiente de simulação quanto o arranjo experimental utilizado e discutidos os resultados obtidos.

Uma nota a ser feita é que as simulações foram realizadas utilizando um modelo da plataforma Pioneer 3, enquanto os experimentos utilizaram a plataforma Turtlebot 2. Esse fato, em conjunto com a introdução de perturbações presentes nos experimentos, como deslizamento, atrasos de comunicação via rede, entre outros, justifica as diferenças entre os resultados nos dois ambientes. Entretanto, como será mostrado, para ambos os casos todos os robôs convergiram para as trajetórias desejadas com erros pequenos.

5.1 Simulação

5.1.1 Parâmetros de Simulação

Como foi mencionado na Seção 4.3, tanto a simulação quanto os experimentos utilizam uma formação de três robôs, denominados α , β e γ , seguindo uma estrutura líder-seguidor com uma arquitetura de controle distribuída. Também comum aos dois ambientes é a existência de um conjunto de parâmetros relativos ao NMPC, aos quais foi feita uma alusão na Seção 2.2.

O primeiro é o critério de parada ϵ para o laço de otimização, que deve ser escolhido de modo a evitar iterações que não tragam ganhos significativos em termos desempenho. Em

paralelo ao critério de parada, é necessário implementar um número de iterações máximo I_{max} , pois dada a natureza *online* do NMPC, ao término do tempo de execução é necessário que haja uma entrada de controle disponível, independentemente de o algoritmo ter atingido o critério de parada ou não. Esse tempo de execução é determinado por um passo de cálculo T . Ainda, como foi analisado na Seção 4.1.3, é necessário especificar as restrições de velocidade linear e angular máximas do sistema.

As simulações foram realizadas utilizando o ambiente *Simulink* do MATLAB, com o método numérico escolhido sendo o de Runge-Kutta. O modelo utilizado no lugar do robô real para as simulações é o modelo da plataforma Pioneer 3 proposto em [Martins 2010], e que pode ser vista na Figura 5.1.



Figura 5.1: Plataforma Pioneer 3.

Os parâmetros para o NMPC podem ser vistos na Tabela 5.1, e os conjuntos dos ganhos para o controlador estão na Tabela 5.2. O objetivo da formação é o seguinte: o líder deve rastrear uma trajetória em forma de oito com um raio de 1.5m, enquanto os seguidores devem ficar a 1.0m e 1.5m do líder. Não podem ocorrer colisões entre os robôs em qualquer momento, e o líder é considerado como tendo parado de funcionar caso ambas as suas velocidades linear e angular sejam zero ao mesmo tempo. O Algoritmo 2 foi utilizado para a seleção do próximo líder, de acordo com a estrutura apresentada na Seção 4.2. O defeito considerado para o líder foi uma falha de comunicação, o que significa a sua parada imediata por não conseguir receber comandos de controle.

Algumas observações devem ser feitas com relação aos ganhos apresentados na Tabela 5.2. A primeira é que para todos comportamentos desejados fora o estado de falha, os termos J_1 , J_2 e J_5 possuem um valor não nulo, uma vez que os robôs em formação estão sempre rastreando ou a trajetória alvo ou o líder. O termo J_3 está presente apenas nos seguidores,

ϵ	I_{max}	T	v_{max}	ω_{max}
0.05	20	0.1	1.2m/s	$\frac{5}{3}\pi$ rad/s

Tabela 5.1: Parâmetros de simulação para a plataforma Pioneer 3.

	λ_1	λ_2	λ_{31}	λ_{32}	λ_{41}	λ_{42}	λ_5
Líder	1400	1.75	0	0	0	0	0.5
Líder desviando do companheiro 1	1400	1.75	0	0	200	0	0.5
Líder desviando do companheiro 2	1400	1.75	0	0	0	200	0.5
Líder desviando de ambos os companheiros	1400	1.75	0	0	100	100	0.5
Seguidor com companheiro	500	200	100	100	0	0	0.5
Seguidor sem o companheiro 1	500	200	100	0	0	200	0.5
Seguidor sem o companheiro 2	500	200	0	100	200	0	0.5
Falha	0	0	0	0	0	0	0

Tabela 5.2: Conjunto de ganhos para a função custo na plataforma Pioneer 3 em ambiente simulado.

uma vez que a abordagem líder-seguidor é adotada. O termo J_4 está presente sempre que há um ou mais robôs sobre a trajetória dos quais é necessário desviar-se. Por fim, como foi observado na Seção 4.2, é fácil observar a natureza complementar dos conjuntos de ganhos referentes ao mesmo comportamento.

5.1.2 Resultados Simulados

Antes de simular a formação em si, é necessário verificar a capacidade do controlador proposto de resolver o problema de rastreamento de trajetória para o líder, de modo a garantir que um comportamento indesejado por parte dele não altere a análise do comportamento dos seguidores. As Figuras 5.2 e 5.3 mostram trajetória percorrida por α e o seu Erro Quadrático Médio (EQM) ao longo dessa trajetória, respectivamente.

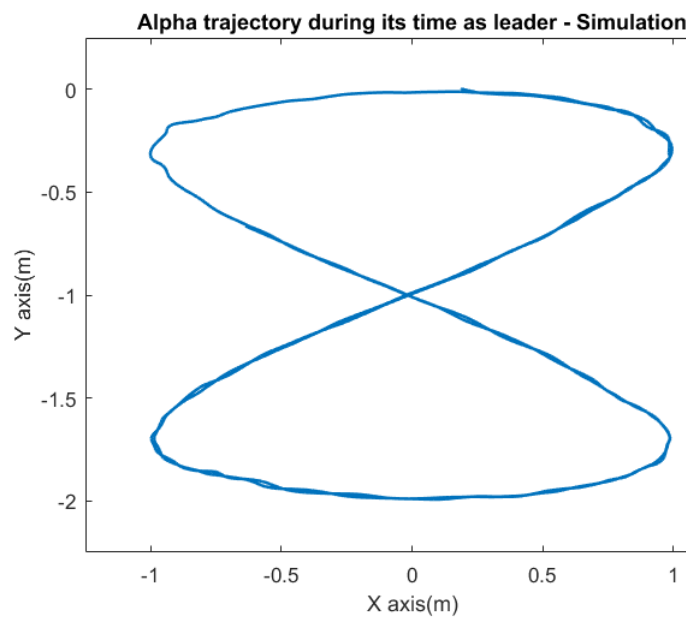


Figura 5.2: Trajetória de α durante o seu tempo como líder de formação em ambiente simulado.

As figuras mostram que α foi capaz de seguir a trajetória, entrando em um regime estacionário rapidamente e apresentando um erro pequeno. As pequenas oscilações que podem ser observadas são fruto da natureza do funcionamento do algoritmo RPROP utilizado pelo NMPC, mas, como será mostrado posteriormente, não possuem uma amplitude significativa. Sabendo que o líder é capaz de rastrear a trajetória desejada com sucesso, foram realizadas

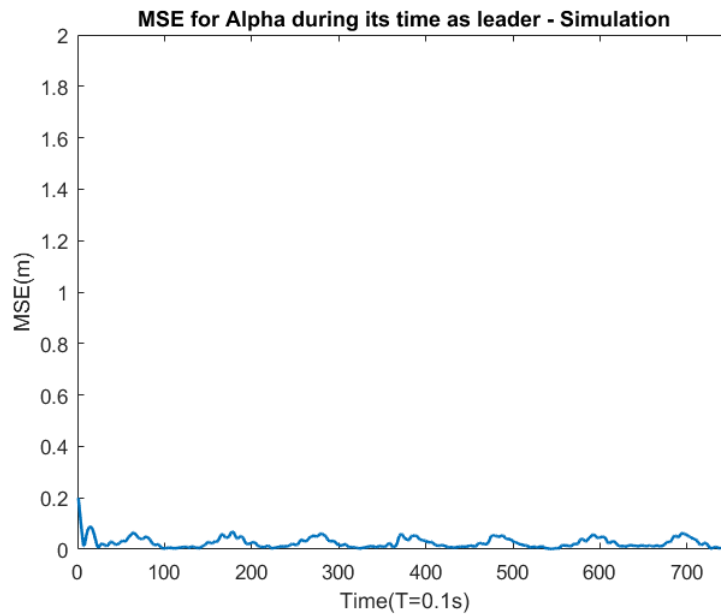


Figura 5.3: EQM entre α e a trajetória de referência durante o seu tempo como líder de formação no ambiente simulado.

as simulações para a formação com três robôs, cujos resultados podem ser vistos nas Figuras 5.4 e 5.5. A Figura 5.5 mostra o EQM com referência à trajetória para cada robô, onde β a princípio tem uma distância alvo de α de 1.0m, enquanto γ a princípio tem uma distância alvo de 1.5m para α , mudando para 1.0m de β quando este passa a ser o líder da formação. Os gráficos para α e β estão representados apenas até o ponto em que ambos apresentam defeito. Para uma melhor visualização, os EQMs de β e γ estão apresentados individualmente nas Figuras 5.6 e 5.7. O EQM de α pode ser visto individualmente na Figura 5.3.

A estratégia de seleção de líder atua a partir do momento em que um dos líderes anteriores para de funcionar. Há um curto período transitório, intensificado pela necessidade de desviar do líder anterior que está sobre a trajetória, movimento este que pode ser visualizado na Figura 5.4. O pequeno pico que pode ser visto no EQM de γ após o momento em que este se torna o líder da formação corresponde justamente a um desses desvios.

As Tabelas 5.3 e 5.4 mostram uma análise numérica dos dados apresentados nos gráficos acima. A Tabela 5.3 mostra a média do EQM para cada robô com relação às suas várias referências ao longo da simulação. Esses valores incluem o regime transitório e os momentos em que há desvios, motivo pelo qual alguns valores são elevados. Entretanto, quando são observados os valores para momentos em que o comportamento predominante é o regime

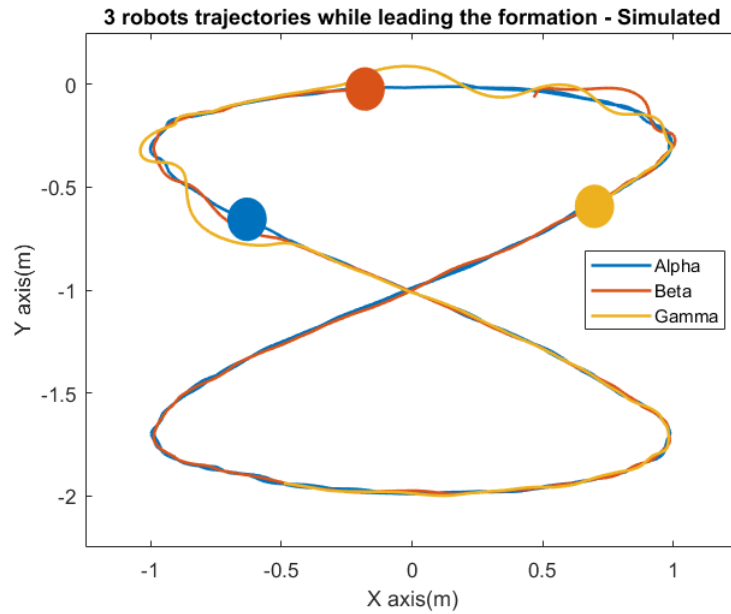


Figura 5.4: Composição da trajetória dos robôs durante o seu tempo como líder de formação em um ambiente simulado. Os desvios correspondem ao robô desviando de um ex-membro da formação.

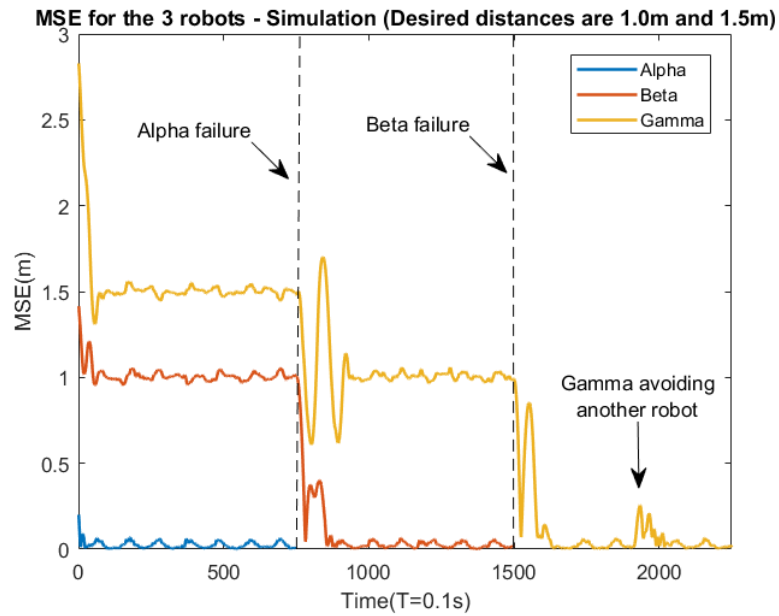


Figura 5.5: MSE dos três robôs durante a simulação. Os gráficos de α e β estão apresentados apenas até o ponto em que ambos apresentaram falhas, o que corresponde ao momento em que ocorre a troca do líder da formação.

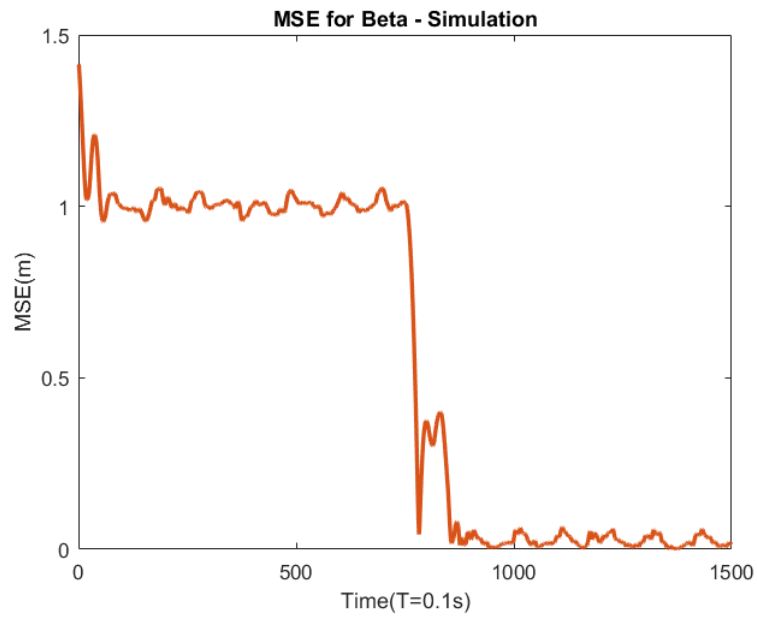


Figura 5.6: EQM entre β e a trajetória de referência em simulação.

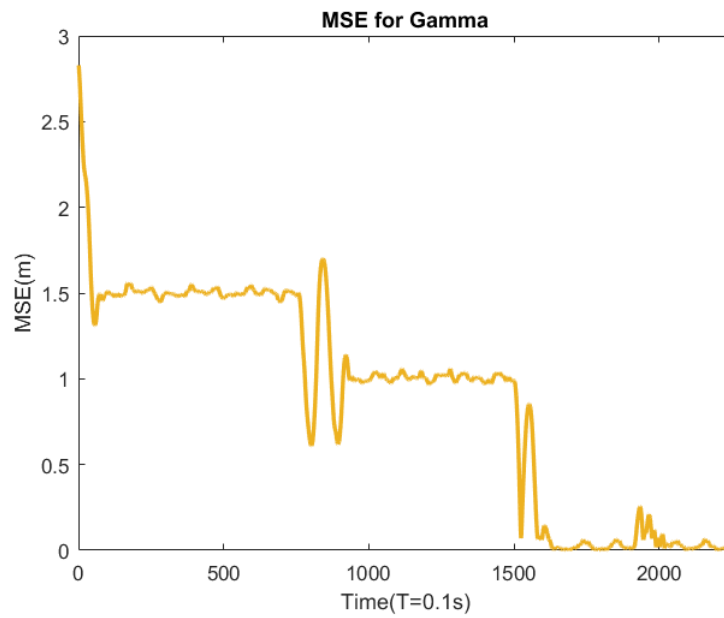


Figura 5.7: EQM entre γ e a trajetória de referência em simulação.

permanente, é possível ver que o erro obtido é pequeno em relação à dimensão da trajetória alvo. A Tabela 5.4 apresenta uma análise com as mesmas características, mostrando que quando o regime permanente é predominante, não só o erro é baixo como as oscilações em torno do alvo são pequenas.

Referência	Trajectoria	α	β	γ
α	0.0253m	-	-	-
β	0.0654m	0.088m	-	-
γ	0.628m	0.0389m	0.0371m	-

Tabela 5.3: EQM médio para os três robôs em relação às suas diferentes referências - Simulação

Referência	Trajectoria	α	β	γ
α	0.0003	-	-	-
β	0.0193	0.0017	-	-
γ	0.151	0.0221	0.0114	-

Tabela 5.4: Variância do EQM para os três robôs em relação às suas diferentes referências - Simulação

Para o ambiente simulado, também foi realizada uma análise do comportamento dos termos da função custo para cada um dos robôs. O termo J_5 está ausente dos gráficos justamente por ser o responsável pelas entradas de controle que serão obtidas pelo controlador, sendo a sua análise análoga ao estudo dos gráficos de trajetória e EQM. A Figura 5.8 mostra a justaposição do termo J_1 para os três robôs durante o período de tempo em que α é o líder da formação. O esforço dos seguidores é significativamente maior que o do líder, fruto do fato de que a tarefa de rastrear outro robô é mais extenuante do que rastrear a trajetória em si.

Na Figura 5.9 é visível que o termo J_1 é muito mais influente sobre o comportamento de α do que o termo J_2 . Isso é uma consequência da natureza não holonômica do robô, que exige um movimento angular para o ajuste de posição, o que termina indiretamente ajustando a posição do robô, ou deixando-a muito próximo do desejado. O comportamento

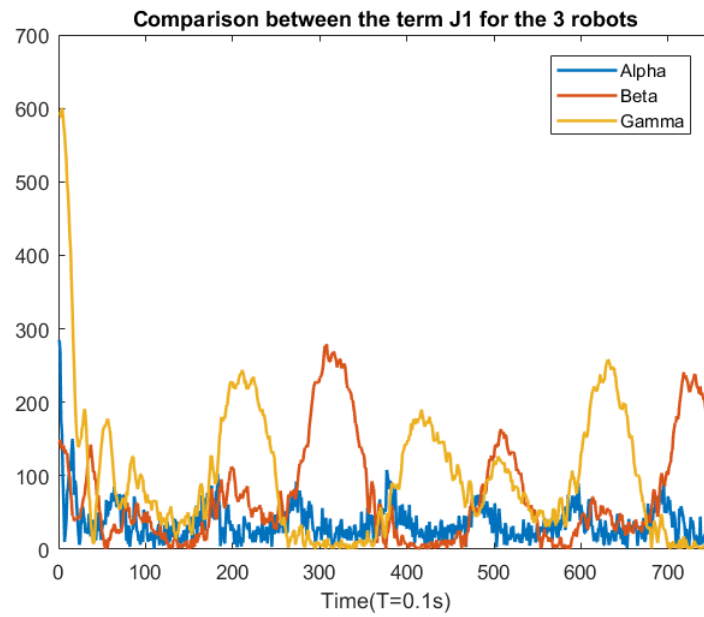


Figura 5.8: Comparação do termo J_1 da função custo para os três robôs durante a o tempo de α como líder.

mais interessante a ser observado na Figura 5.10 é o pico de J_1 no momento em que β assume como líder de formação, e o pico de J_4 que corresponde ao momento em que o robô realiza um desvio. Um comportamento semelhante pode ser observado na Figura 5.11.

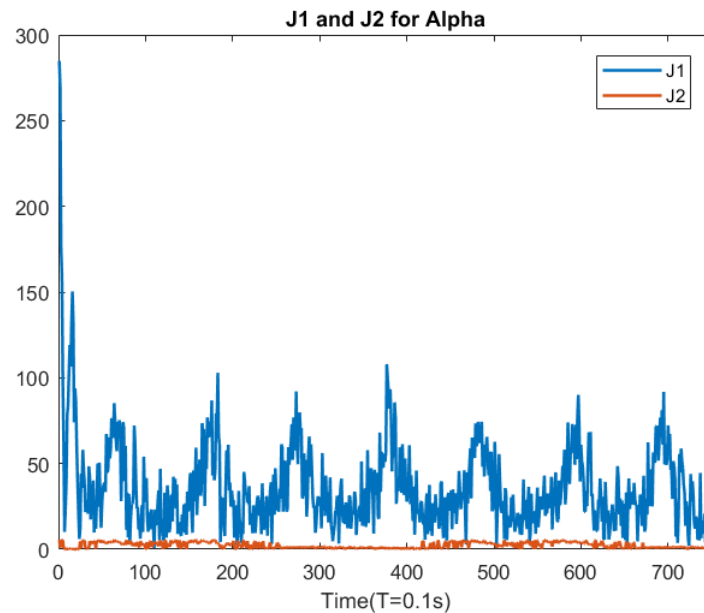


Figura 5.9: Análise dos termos da função custo de α até o ponto em que foi observada uma falha.

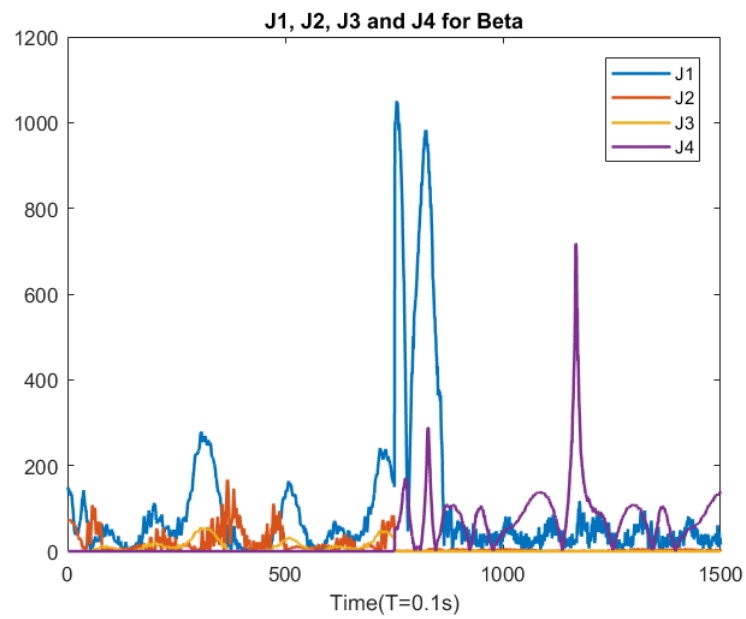


Figura 5.10: Análise dos termos da função custo de β até o ponto em que foi observada uma falha.

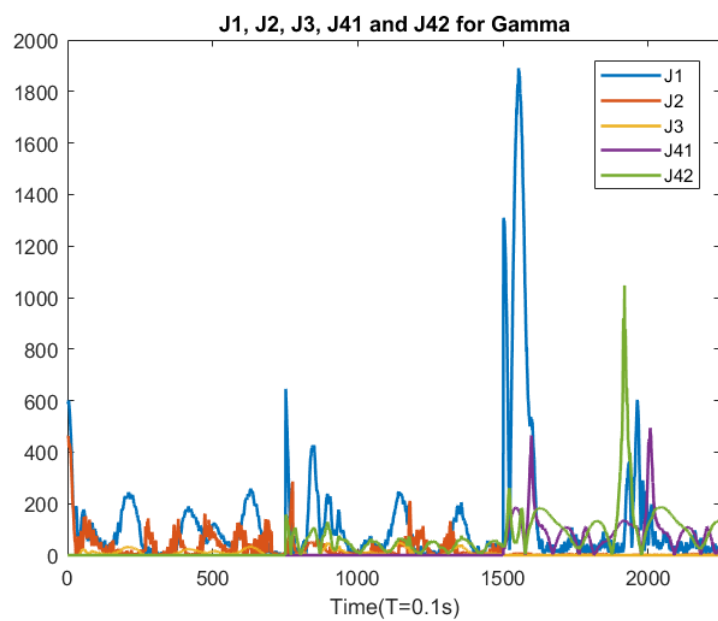


Figura 5.11: Análise dos termos da função custo de γ .

5.2 Experimentos

5.2.1 Arranjo Experimental

A primeira observação a ser feita é que no que concerne aos parâmetros do NMPC e objetivos da formação, incluindo o robô que é o líder inicial e o critério para a seleção de líder, são mantidas as mesmas observações que foram feitas na Seção 5.1.1. Os valores dos parâmetros do NMPC para o ambiente experimental podem ser vistos na Tabela 5.5, e o arranjo experimental pode ser visto na Figura 5.12. Os dados de posição dos robôs foram obtidos a partir da odometria já inclusa na plataforma Turtlebot 2, e a comunicação entre eles ocorre por meio de uma rede *wireless*.

ϵ	I_{max}	T	v_{max}	ω_{max}
0.05	15	0.04	0.7m/s	π rad/s

Tabela 5.5: Parâmetros experimentais para a plataforma Turtlebot 2.

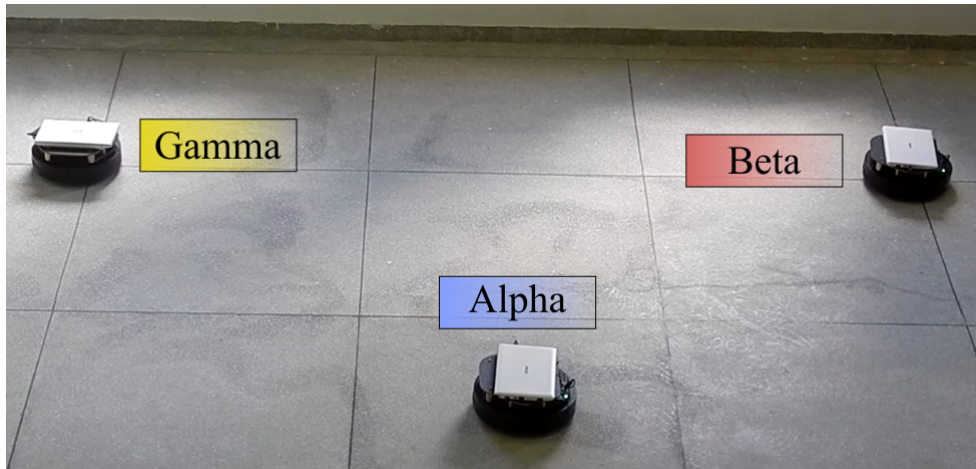


Figura 5.12: Arranjo experimental utilizado. α é o líder inicial.

Os experimentos foram realizados na plataforma Turtlebot 2, que também pode ser vista na Figura 5.12. Foi utilizada a ferramenta *Robotics System System Toolbox* para realizar a integração entre o controlador rodando no Simulink e o ROS. Os ganhos do controlador para o experimento podem ser vistos na Tabela 5.6, e são válidas as mesmas observações que foram feitas acerca dos ganhos em ambiente simulado.

	λ_1	λ_2	λ_{31}	λ_{32}	λ_{41}	λ_{42}	λ_5
Líder	800	500	0	0	0	0	0.05
Líder desviando do companheiro 1	800	500	0	0	600	0	0.05
Líder desviando do companheiro 2	800	500	0	0	0	600	0.05
Líder desviando de ambos os companheiros	800	500	0	0	600	600	0.05
Seguidor com companheiro	800	600	500	500	0	0	0.03
Seguidor sem o companheiro 1	800	600	500	0	0	600	0.03
Seguidor sem o companheiro 2	800	600	0	500	600	0	0.03
Falha	0	0	0	0	0	0	0

Tabela 5.6: Conjunto de ganhos para a função custo na plataforma Turtlebot 2 em ambiente experimental.

5.2.2 Resultados Experimentais

De maneira análoga a simulação, a primeira análise a ser realizada é acerca da capacidade de α de rastrear a trajetória. A Figura 5.13 mostra α rastreando a trajetória, enquanto a Figura 5.14 mostra o EQM correspondente. Quando são comparados esses resultados aos obtidos em simulação, fica visível que os resultados experimentais são melhores. Isso é uma consequência do fato de que a capacidade do NMPC de corrigir um sistema é derivada da correspondência entre o modelo de predição e o sistema em si, e, no nosso caso, o modelo de predição corresponde melhor ao sistema do robô real do que ao modelo simulado.

Em seguida, é realizada a análise do desempenho da formação, que pode ser visto nas Figuras 5.15 e 5.16. A natureza do desvio de γ é diferente quando comparada à simulação, adotando um comportamento em que é traçada uma intercessão entre a trajetória e um caminho que não resulte em uma colisão. Isso ocorre porque o modelo simulado não leva em conta as dimensões dos robôs, o que permite desvios menores. As Tabelas 5.2.2 e 5.2.2 apresentam a análise numérica dos dados contidos nesses gráficos, e assim como na simulação os erros são mínimos nos trechos em que predomina o regime permanente.

A última análise a ser realizada é a comparação dos resultados obtidos ao longo desta pesquisa com os resultados obtidos na bibliografia. Uma dificuldade na realização dessa comparação é a predominância de trabalhos que apresentam resultados apenas de maneira

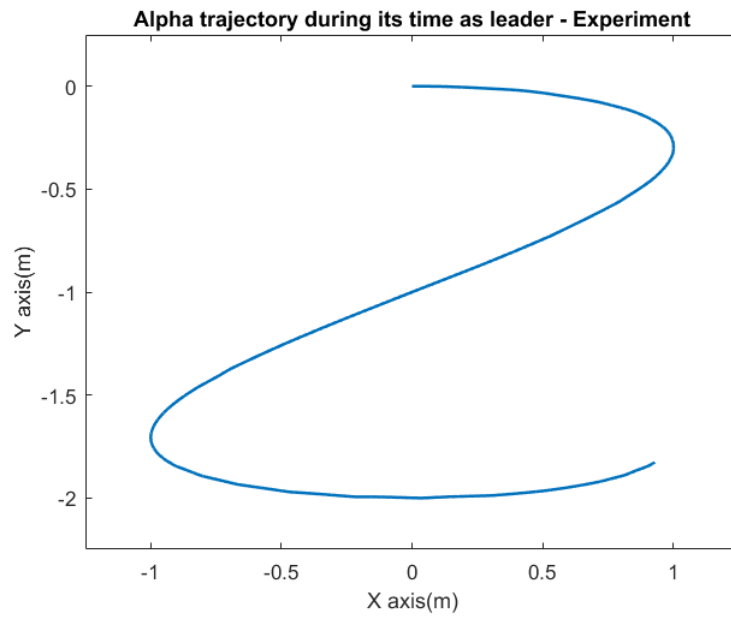


Figura 5.13: Trajetória de α durante o seu tempo como líder de formação em ambiente experimental.

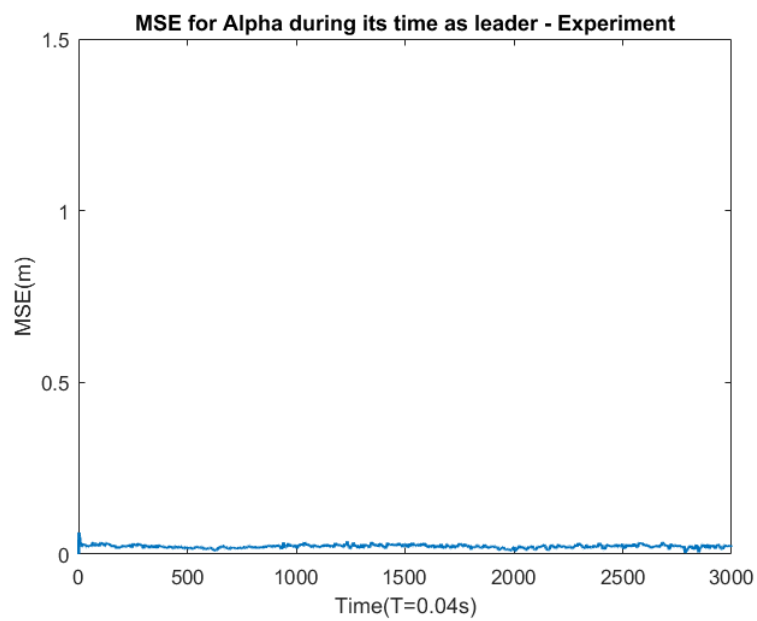


Figura 5.14: EQM entre α e a trajetória de referência durante o seu tempo como líder de formação em ambiente experimental.

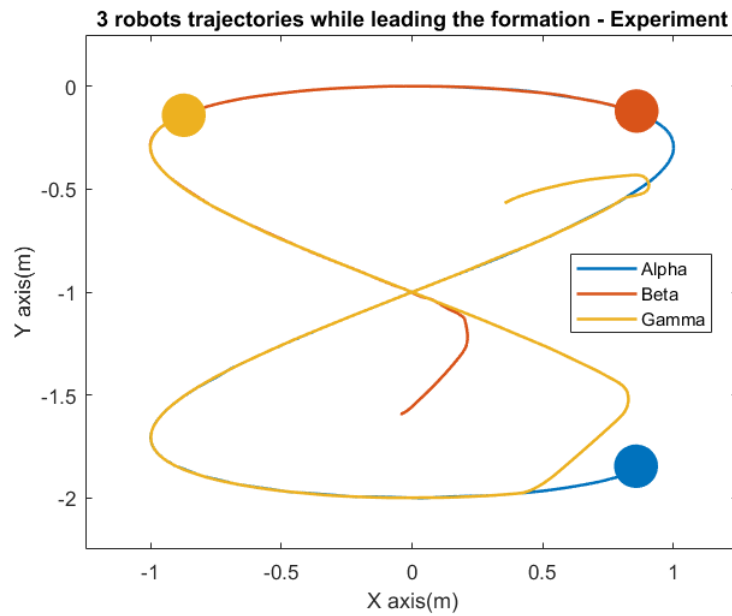


Figura 5.15: Composição da trajetória dos robôs durante o seu tempo como líder de formação em um ambiente experimental. Os desvios correspondem ao robô desviando de um ex-membro da formação.

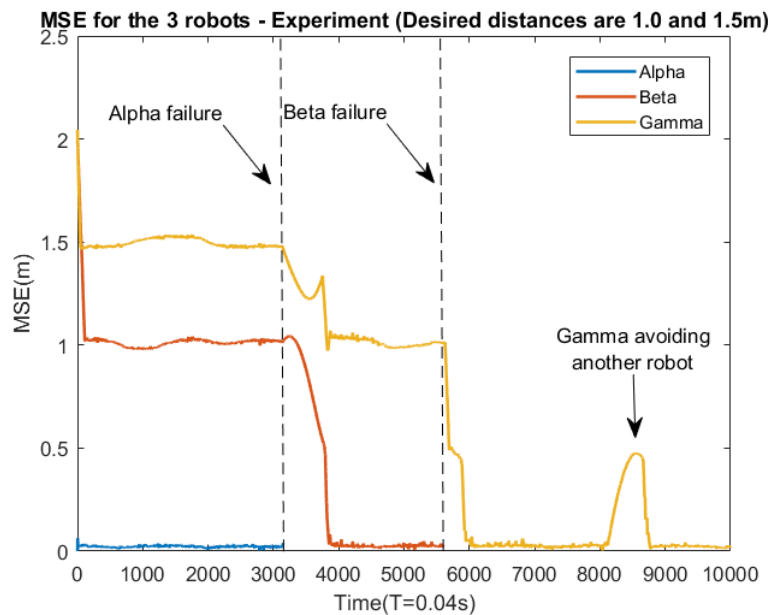


Figura 5.16: EQM dos três robôs durante o experimento. Os gráficos de α e β estão apresentados apenas até o ponto em que ambos apresentaram falhas, o que corresponde ao momento em que ocorre a troca do líder da formação.

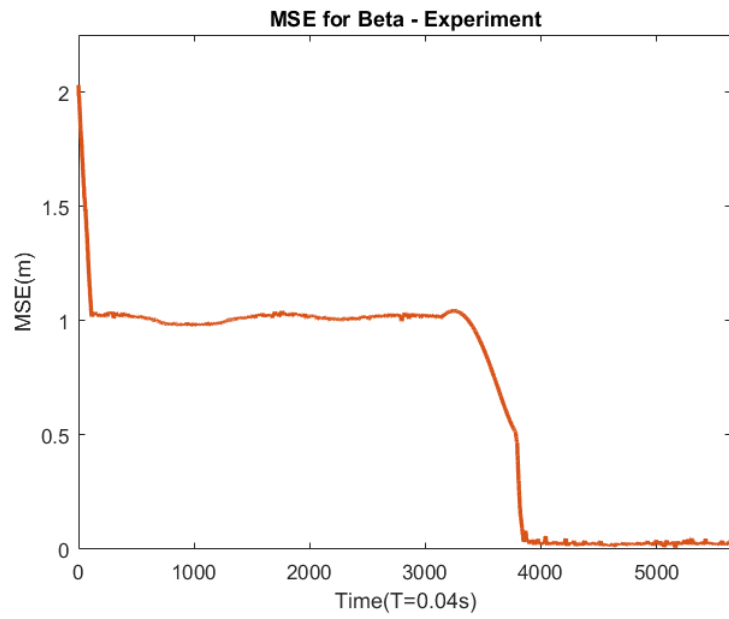


Figura 5.17: EQM entre β e a trajetória de referência no experimento.

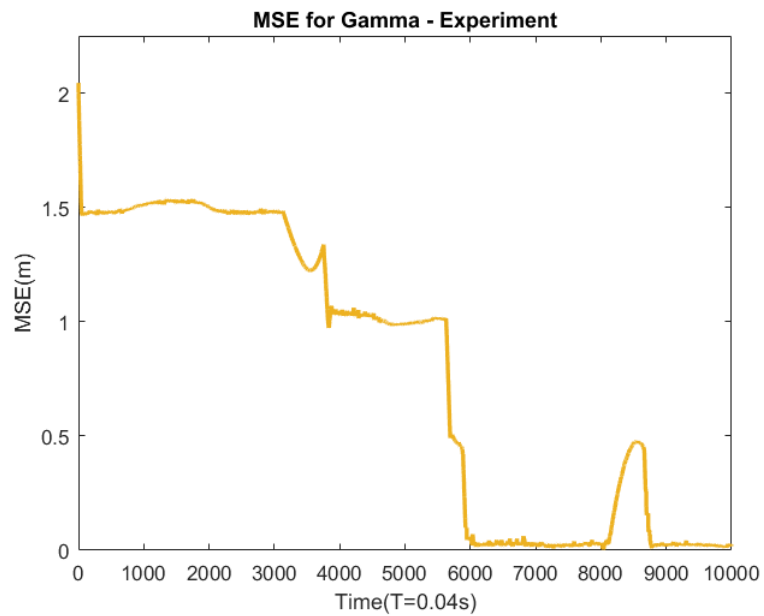


Figura 5.18: EQM entre γ e a trajetória de referência no experimento.

Reference	Target Trajectory	α	β	γ
α	0.0226m	-	-	-
β	0.0872m	0.0303m	-	-
γ	0.1033m	0.0006m	0.0376m	-

Tabela 5.7: EQM médio para os três robôs em relação às suas diferentes referências - Experimento

Reference	Target Trajectory	α	β	γ
α	0.0001	-	-	-
β	0.0312	0.0122	-	-
γ	0.029	0.0017	0.006	-

Tabela 5.8: Variância do EQM para os três robôs em relação às suas diferentes referências - Experimento

gráfica, o que dificulta realizar uma comparação quantitativa. Em [Mehrez M.W. e Gosine 2017] foi implementado um algoritmo de horizonte variável em conjunto com o NMPC para uma formação de robôs, enquanto em [Ostafew, Schoellig e Barfoot 2016] também foi implementado um NMPC para um robô móvel rastreando uma trajetória. No segundo caso, foi realizada uma análise dos gráficos, considerando os experimentos realizados em ambiente *indoor*. Como foi citado previamente, os resultados apresentados nas Tabelas 5.3 e 5.2.2 incorporam o período transitório, enquanto a maioria dos resultados lida apenas com o regime permanente. Assim, para o fim de comparação foi escolhido o valor de erro de β durante o seu tempo como líder, em que o efeito do regime transitório é menor e consequentemente há uma baixa variância, o que aumenta a confiabilidade da média do EQM. A comparação pode ser vista na Tabela 5.2.2. É importante notar que os artigos utilizados como base de comparação utilizam robôs com características similares aos escolhidos neste trabalho, ou seja, robôs terrestres móveis não-holonômicos.

Simulação(β)	Experimento(β)	[Mehrez M.W. e Gosine 2017]	[Ostafew et al. 2016]
0.088m	0.0303m	$> 0.12m$	$> 0.1m$

Tabela 5.9: Comparação dos resultados obtidos nos ambientes simulado e experimental com os valores encontrados na literatura.

5.3 Considerações Finais

Foram apresentados os resultados decorrentes da implementação dos conceitos apresentados no Capítulo 4. Os resultados obtidos, tanto em simulações quanto em experimentos, atestam a capacidade das técnicas de controle e de seleção de líder de cumprirem aquilo a que elas se propõem, não só por fornecer uma validação como por mostrar que o desempenho é superior aos valores encontrados na pesquisa da literatura realizada para esta pesquisa.

Capítulo 6

Conclusão

Foi apresentada uma abordagem multirrobô para o NMPC modificado, assim como uma estrutura para seleção de líder dinâmica. Os resultados, tanto no ambiente simulado como no experimental, mostram que essa é uma solução efetiva para o problema de rastreamento de trajetória para formações de robôs móveis, inclusive quando comparada aos resultados encontrados na literatura. Assim, os objetivos enunciados no Capítulo 1 foram cumpridos, assim como foram apresentadas as contribuições propostas no mesmo capítulo, desde a elaboração de uma plataforma de testes confiável até os resultados mostrando que os robôs rastreiam a trajetória desejada com erros médios na ordem de $10^{-2}m$. Os robôs foram capazes de realizar a troca de líder quando foi detectada uma falha de comunicação com o líder atual, redefinindo a formação dentro de um espaço de tempo apropriado, e fazendo isso apenas a partir da manipulação dos ganhos da função custo do controlador, o que indica que a meta de mudar o comportamento dos robôs sem a necessidade de uma camada do sistema dedicada exclusivamente para esse propósito também foi atingida.

É claro que o trabalho aqui apresentado não é final, e há diversas áreas de interesse que os resultados obtidos levantam. O primeiro é uma análise aprofundada dos métodos de otimização para o NMPC. Embora o algoritmo RPROP utilizado apresente um bom desempenho, dado que a versão do NMPC utilizada aqui é modificada quando comparada ao algoritmo tradicional, as análises conduzidas acerca da efetividade de heurísticas clássicas para esse problema precisam ser reconsideradas. Essa linha é similar à análise do impacto do método de resolução numérica utilizado sobre o desempenho do sistema, que é outra direção possível para pesquisas futuras. Também foi observado ao longo da pesquisa que há um impacto

considerável da confiabilidade da rede utilizada para a comunicação entre os robôs sobre o desempenho da formação, sendo esta uma outra área a ser explorada.

Uma extensão natural para o trabalho é a transição para formações de robôs heterogêneas, incorporando veículos aéreos não tripulados, por exemplo, o que adiciona uma camada de complexidade a mais ao sistema, em troca de permitir uma variedade de aplicações muito maior. A formulação do algoritmo de controle proposto não impõe restrições acerca do sistemas, o que significa que *a priori* não há restrições para a sua implementação para esse tipo de aplicação, necessitando-se apenas do modelo de predição e função custo apropriados.

Outra área de possível expansão é a incorporação de métodos de posicionamento e odometria mais avançados do que os aqui utilizados, principalmente com o uso de algoritmos de processamento de imagem aplicados a imagens coletadas por câmeras acopladas ao robô, podendo-se incorporar o planejamento dinâmico de trajetórias, melhorar a precisão da odometria nativa do robô através de uma fusão de sensores ou incorporar o mapeamento de características do ambiente como um fator a ser considerado pelos robôs da formação.

Referências Bibliográficas

- [Almonfrey, Vassallo e Dalfior 2010]ALMONFREY, D.; VASSALLO, R. F.; DALFIOR, J. S. Visual-servo controller applied to the leader of a robot team with a centralized formation control. In: *2010 IEEE International Conference on Industrial Technology*. [S.l.: s.n.], 2010. p. 1481–1486.
- [Chen, Wang e Wu 2017]CHEN, Y.; WANG, C.; WU, C. A stable path guidance law for a small uav with parameters optimized by nmpc. In: *2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 787–792.
- [Cook, Ray e Lever 2018]COOK, J. T.; RAY, L. E.; LEVER, J. H. Dynamics modeling and robotic-assist, leader-follower control of tractor convoys. *Journal of Terramechanics*, v. 75, p. 57 – 72, 2018. ISSN 0022-4898. Agriculture and Forestry in Terrain-Machine Systems. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022489816301136>>.
- [Du, Htet e Tan 2016]DU, X.; HTET, K. K. K.; TAN, K. K. Development of a genetic-algorithm-based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles. *IEEE Transactions on Industrial Electronics*, v. 63, n. 11, p. 6970–6977, 2016.
- [Duijkeren et al. 2016]DUIJKEREN, N. van et al. Path-following nmpc for serial-link robot manipulators using a path-parametric system reformulation. In: *2016 European Control Conference (ECC)*. [S.l.: s.n.], 2016. p. 477–482.
- [El-Ferik, Siddiqui e Lewis 2016]EL-FERIK, S.; SIDDIQUI, B. A.; LEWIS, F. L. Distributed nonlinear mpc of multi-agent systems with data compression and random delays. *IEEE Transactions on Automatic Control*, v. 61, n. 3, p. 817–822, March 2016. ISSN 0018-9286.

- [Feng, Lu e Chen 2015]FENG, K.; LU, J.; CHEN, J. Nonlinear model predictive control based on support vector machine and genetic algorithm. *Chinese Journal of Chemical Engineering*, v. 23, n. 12, p. 2048 – 2052, 2015.
- [Ferreira 2010]FERREIRA, J. R. A. *Controlo Coordenado de Equipas de Robots Móveis*. Tese (Doutorado) — Universidade do Porto, 2010.
- [Gao e Guo 2018]GAO, Z.; GUO, G. Fixed-time leader-follower formation control of autonomous underwater vehicles with event-triggered intermittent communications. *IEEE Access*, v. 6, p. 27902–27911, 2018.
- [Grüne e Pannek 2011]GRÜNE, L.; PANNEK, J. *Nonlinear Model Predictive Control - Theory and Algorithms*. [S.l.]: Springer, 2011.
- [Hoshino, Tamura e Hayakawa 2017]HOSHINO, S.; TAMURA, T.; HAYAKAWA, Y. Interactive formation by multiple mobile robots for guiding humans. In: *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. [S.l.: s.n.], 2017. p. 327–333.
- [Jayasiri, Gros e Mann 2016]JAYASIRI, A.; GROS, S.; MANN, G. K. I. Tracking control and state estimation of a mobile robot based on nmpe and mhe. In: *2016 American Control Conference (ACC)*. [S.l.: s.n.], 2016. p. 1999–2004.
- [Karpov e Karpova 2015]KARPOV, V.; KARPOVA, I. Leader selection algorithms for static swarms. *Biologically Inspired Cognitive Architectures*, v. 12, p. 54 – 64, 2015. ISSN 2212-683X.
- [Kayastha et al. 2017]KAYASTHA, S. et al. Nonlinear model predictive control of a planar three-link space manipulator. In: *2017 11th Asian Control Conference (ASCC)*. [S.l.: s.n.], 2017. p. 635–640.
- [Lei, Chaofang e Na 2017]LEI, C.; CHAOFANG, H.; NA, W. Obstacle avoidance control of unmanned ground vehicle based on nmpe. In: *2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 6402–6406.

- [Li, Ding e Hao 2017]LI, F.; DING, Y.; HAO, K. A neuroendocrine inspired dynamic leader selection model in formation control for multi-robot system. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2017. p. 5454–5459.
- [Li et al. 2017]LI, F. et al. An affection-based dynamic leader selection model for formation control in multirobot systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 47, n. 7, p. 1217–1228, July 2017. ISSN 2168-2216.
- [Li et al. 2018]LI, Z. et al. Neural-dynamic optimization-based model predictive control for tracking and formation of nonholonomic multirobot systems. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–10, 2018. ISSN 2162-237X.
- [Lin 2018]LIN, F. Performance of leader–follower multi-agent systems in directed networks. *Systems and Control Letters*, v. 113, p. 52 – 58, 2018. ISSN 0167-6911. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167691118300100>>.
- [Ma, Liu e Chen 2016]MA, Z.; LIU, Z.; CHEN, Z. Leader-following consensus of multi-agent system with a smart leader. *Neurocomputing*, v. 214, p. 401 – 408, 2016. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231216306804>>.
- [Maghenem, Loria e Panteley 2018]MAGHENEM, M.; LORIA, A.; PANTELEY, E. A robust delta-persistently exciting controller for leader-follower tracking-agreement of multiple vehicles. *European Journal of Control*, v. 40, p. 1 – 12, 2018. ISSN 0947-3580. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0947358017300080>>.
- [Martins 2010]MARTINS, F. N. *Modelagem e Compensação da Dinâmica de Robôs Móveis e sua Aplicação em Controle de Formação*. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2010.
- [Mehrez, Mann e Gosine 2015]MEHREZ, M. W.; MANN, G. K. I.; GOSINE, R. G. Comparison of stabilizing nmpe designs for wheeled mobile robots: An experimental study. In: *2015 Moratuwa Engineering Research Conference (MERCon)*. [S.l.: s.n.], 2015. p. 130–135.

- [Mehrez M.W. e Gosine 2017]MEHREZ M.W., I. M. G.; GOSINE, R. J. An optimization based approach for relative localization and relative tracking control in multi-robot systems. *Intell Robot Syst*, 2017. ISSN 0921-0296.
- [Nascimento, Dórea e Gonçalves 2018]NASCIMENTO, T. P.; DÓREA, C. E. T.; GONÇALVES, L. M. G. Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach. *International Journal of Advanced Robotic Systems*, v. 15, n. 1, p. 1729881418760461, 2018.
- [Ostafew, Schoellig e Barfoot 2014]OSTAFEW, C. J.; SCHOELLIG, A. P.; BARFOOT, T. D. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2014. p. 4029–4036.
- [Ostafew, Schoellig e Barfoot 2016]OSTAFEW, C. J.; SCHOELLIG, A. P.; BARFOOT, T. D. Robust constrained learning-based nmpc enabling reliable mobile robot path tracking. *The International Journal of Robotics Research*, v. 35, n. 13, p. 1547–1563, 2016.
- [Pei et al. 2017]PEI, Y. et al. Nonlinear model predictive tracking control of nonholonomic wheeled mobile robot using modified c/gmres algorithm. In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. [S.l.: s.n.], 2017. p. 6298–6303.
- [Peterson e Barton 2015]PETERSON, C. K.; BARTON, J. Virtual structure formations of cooperating uavs using wind-compensation command generation and generalized velocity obstacles. In: *2015 IEEE Aerospace Conference*. [S.l.: s.n.], 2015. p. 1–7. ISSN 1095-323X.
- [Riedmiller e Braun 1993]RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: *IEEE International Conference on Neural Networks*. [S.l.: s.n.], 1993. p. 586–591 vol.1.
- [Rosero et al. 2016]ROSERO, P. P. et al. Temperature nonlinear model predictive controller (nmpc) for a dryer plant of cocoa beans. In: *2016 IEEE International Conference on Automatica (ICA-ACCA)*. [S.l.: s.n.], 2016. p. 1–6.

- [Saeidi, Mikulski e Wang 2017] SAEIDI, H.; MIKULSKI, D. G.; WANG, Y. Trust-based leader selection for bilateral haptic teleoperation of multi-robot systems. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2017. p. 6575–6581.
- [Sakurama 2017] SAKURAMA, K. Leader selection via lasso for formation control of time-delayed multi-agent systems. *Neurocomputing*, v. 270, p. 18 – 26, 2017.
- [Schultze e Horn 2016] SCHULTZE, M.; HORN, J. Modeling, state estimation and nonlinear model predictive control of cathode exhaust gas mass flow for {PEM} fuel cells. *Control Engineering Practice*, v. 49, p. 76 – 86, 2016.
- [Simba, Uchiyama e Sano 2016] SIMBA, K. R.; UCHIYAMA, N.; SANO, S. Real-time smooth trajectory generation for nonholonomic mobile robots using bézier curves. *Robotics and Computer-Integrated Manufacturing*, v. 41, p. 31 – 42, 2016. ISSN 0736-5845. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0736584516300552>>.
- [Wang et al. 2016] WANG, G. et al. Distributed cooperative control of multiple nonholonomic mobile robots. *Journal of Intelligent & Robotic Systems*, v. 83, n. 3, p. 525–541, 2016.
- [Wilson, Charest e Dubay 2016] WILSON, J.; CHAREST, M.; DUBAY, R. Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator. *Robotics and Computer-Integrated Manufacturing*, v. 41, p. 23 – 30, 2016.
- [Xiao e Chen 2017] XIAO, H.; CHEN, C. L. P. Leader-follower multi-robot formation system using model predictive control method based on particle swarm optimization. In: *2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. [S.l.: s.n.], 2017. p. 480–484.
- [Xiao et al. 2017] XIAO, H. et al. General projection neural network based nonlinear model predictive control for multi-robot formation and tracking. *IFAC-PapersOnLine*, v. 50, n. 1, p. 838 – 843, 2017. ISSN 2405-8963. 20th IFAC World Congress. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S240589631730191X>>.
- [Xing e Deng 2018] XING, M.; DENG, F. Distributed event-triggered observer-based tracking control of leader–follower multi-agent systems. *Neurocom-*

puting, v. 273, p. 650 – 658, 2018. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231217314637>>.

[Zarghami et al. 2014]ZARGHAMI, M. et al. Model-based predictive control of wheeled omni-directional robots considering nonlinear dynamical constraints and input delay. In: *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*. [S.l.: s.n.], 2014. p. 1379–1385.

[Zou et al. 2018]ZOU, Y. et al. Distributed formation control for multiple vertical takeoff and landing uavs with switching topologies. *IEEE/ASME Transactions on Mechatronics*, p. 1–1, 2018. ISSN 1083-4435.

[Ławryńczuk 2016]ŁAWRYŃCZUK, M. Modelling and predictive control of a neutralisation reactor using sparse support vector machine wiener models. *Neurocomputing*, v. 205, p. 311 – 328, 2016.