UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Campos potenciais modificados aplicados ao controle de robôs em ambientes tridimensionais

Marcelo Oliveira da Silva

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marcelo Oliveira da Silva

Campos potenciais modificados aplicados ao controle de robôs em ambientes tridimensionais

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Roseli Aparecida Francelin Romero

USP – São Carlos
Fevereiro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

O48c

Oliveira da Silva, Marcelo
Campos potenciais modificados aplicados ao controle de robôs em ambientes tridimensionais / Marcelo Oliveira da Silva; orientadora Roseli Aparecida Francelin Romero. -- São Carlos, 2019.
88 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

1. Robótica Movel Inteligente. 2. Campos Potenciais. 3. Planejamento de Caminhos. 4. Planejamento de Trajetórias. I. Aparecida Francelin Romero, Roseli, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Marcelo Oliveira da Silva

Modified potential fields applied to robot control inside three-dimentional environments

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Roseli Aparecida Francelin Romero

USP – São Carlos
February 2019

RESUMO

SILVA, M. O. **Campos potenciais modificados aplicados ao controle de robôs em ambientes tridimensionais**. 2019. 88 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Nos últimos anos, a área de robôs aéreos vêm se tornando cada vez mais importantes no dia-a-dia, em diversos usos, em que se pode destacar: segurança pública e particular, agricultura de precisão, registro fotográfico de eventos, serviços de entregas e apoio a diversas outras áreas, como monitoramento ambiental.Para que tais robôs aéreos possam cumprir suas mais variadas tarefas, faz-se necessária uma etapa de planejamento de movimento, que consiste em encontrar um caminho factível entre a postura atual e uma postura final (também chamada de postura alvo ou meta) do robô aéreo. Neste trabalho, a tarefa de planejamento de movimento é abordada para o caso tridimensional em ambientes dinâmicos, nos quais não se assume que todos os obtáculos permanecerão fixos ao longo do trajeto. Derivado da Teoria de Campos Potenciais Harmônicos, os Campos Potenciais Modificados (CPM) permitem a distorção do campo potencial favorecendo uma direção específica de chegada a postura meta. Tais CPM resultam em um planejador de movimentos para ambientes dinâmicos e multidimensionais, em especial, o caso tridimensional.

Palavras-chave: Campos Potenciais, Robôs Móveis, Robótica, Planejamento de Caminhos, Planejamento de Trajetórias.

ABSTRACT

SILVA, M. O. **Modified potential fields applied to robot control inside three-dimentional environments**. 2019. 88 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

In recent years, aerial robots have become increasingly important in day-to-day situations, in several uses, in which we can highlight: public and private security, precision agriculture, photographic record of events, delivery and support to several other areas, such as environmental monitoring. In order for aerial robots perform their broad range of tasks, a motion planning step is necessary. Motion planning consists in finding a feasible path between the current posture and a final posture (also called target posture or goal) of a robot. In this work, the task of motion planning is approached in three-dimensional and dynamic environments, in which it is not assumed that all the obstacles will remain fixed along the trajectory. Derived from Harmonic Potential Field Theory, Modified Potential Fields (MPF) allows a controlled distortion of the potential field, as an example, towards a specific direction of arrival to the target posture. Such MPF results in a motion planner for dynamic and multidimensional environments, especially the three-dimensional case.

Keywords: Potential Fields, Mobile Robots, Robotics, Path Planning, Trajectory Planning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Robô com espaço de configuração com 3 dimensões 26
Figura 2 – Robô com espaço de configuração com 21 dimensões 27
Figura 3 – Sistemas de coordenadas universal e de referência 28
Figura 4 – Distância entre ângulos 29
Figura 5 – Fase de planejamento de um PRM 34
Figura 6 – O procedimento auxiliar $EXTEND_RRT$ - Algoritmo 3 35
Figura 7 – Força repulsiva na vizinhança de um obstáculo 40
Figura 8 – Força repulsiva do objeto O sobre um robô localizado em R 40
Figura 9 – Força atrativa constante 41
Figura 10 – Força atrativa entre o robô R e a meta M 41
Figura 11 – Potencial atrativo da meta 42
Figura 12 – Potencial repulsivo de dois obstáculos 43
Figura 13 – Soma dos campos vetoriais das Figuras 11 e 12 43
Figura 14 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 1) 44
Figura 15 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 2) 44
Figura 16 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 3 - Situaçao 1) 45
Figura 17 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 3 - Situação 2) 45
Figura 18 – Uma situação que pode ocorrer dentro do ambiente de futebol de robôs. As áreas cinzas indicam as regiões de influência de cada robô 49
Figura 19 – Exemplo de situação aplicável ao CPO3D 54
Figura 20 – Caminho planejado a partir da situação da Figura 19 55
Figura 21 – Diagrama da MOPLALIB 60
Figura 22 – Screenshot da **mpl_GUI** 62
Figura 23 – Exemplo de Snapshots possíveis com a interface **mpl_GUI** 63
Figura 24 – Ambiente discretizado utilizando o conceito de células vermelhas e pretas alternantes 63
Figura 25 – Esquema lógico das instâncias paralelas, frisando o mecanismo de comunicação entre elas 64
Figura 26 – Associação de Dígrafo a Matrizes 88

LISTA DE ALGORITMOS

Algoritmo 1 – Procedimento para criação de PRMs 34
Algoritmo 2 – Procedimento para criação de RRTs 35
Algoritmo 3 – Procedimento auxiliar para criação de RRTs: $EXTEND_RRT$ 36

LISTA DE TABELAS

Tabela 1 – Especificações do Computador 67
Tabela 2 – Experimentos com o CPH 3D usando SOR com $\omega = 1.899$ 68
Tabela 3 – Experimentos com o CPH 3D usando Gauss-Seidel 68
Tabela 4 – Execução do CPH3D usando Gauss-Seidel 68
Tabela 5 – Execução do CPO3D usando Gauss-Seidel 69

LISTA DE ABREVIATURAS E SIGLAS

CPH	Campo Potencial Harmônico
CPK	Campo Potencial de Khatib
CPLO	Campo Potencial Localmente Orientado
CPO	Campo Potencial Orientado
CPO3D	Campo Potencial Orientado Tridimensional
EDP	Equação Diferencial Parcial
GPGPU	do inglês, *General Purpose GPUs*, isto é, placas gráficas de processamento genérico
GS	Gauss-Seidel
MOPLALIB	do inglês *Motion Planing Library*, ou Biblioteca de Planejamento de Movimento
PRM	*Probabilistic Road Maps*
PVC	Problema de Valor de Contorno
SLAM	do inglês, *simultaneous localization and mapping*, localização e mapeamento simultâneos
SOR	Successive Over-Relaxation

LISTA DE SÍMBOLOS

P — Potencial

∇ — Laplaciano

p_* — Potencial discreto no ponto $(*)$

$\langle , \rangle$ — Produto escalar

$||v||$ — Norma do vetor v

ε — Taxa de influência do vetor do CPO

λ_x — $h\varepsilon|\mathbf{v}_x|$

λ_y — $h\varepsilon|\mathbf{v}_y|$

λ_z — $h\varepsilon|\mathbf{v}_z|$

Ω — Ambiente pelo qual o robô poderá de deslocar; normalmente um subconjunto de $\mathbb{R}^n$

$\mathbb{R}^n$ — Espaço real com n dimensões

ξ_x — $\dfrac{(1+\lambda_x)}{6+\lambda_x+\lambda_y+\lambda_z}$

ξ_y — $\dfrac{(1+\lambda_y)}{6+\lambda_x+\lambda_y+\lambda_z}$

ξ_z — $\dfrac{(1+\lambda_z)}{6+\lambda_x+\lambda_y+\lambda_z}$

ξ_1 — $\dfrac{1}{6+\lambda_x+\lambda_y+\lambda_z}$

SUMÁRIO

INTRODUÇÃO 21
Objetivos 23
Organização do Texto 23

1 PLANEJAMENTO DE MOVIMENTO 25
1.1 Espaço de Configuração 27
1.2 Caminho 30
1.2.1 Extensão do problema básico: Obstáculos móveis 30
1.3 Considerações Finais 32

2 TRABALHOS CORRELATOS 33
2.1 Probabilistic Road Maps 33
2.2 Rapidly-exploring Random Tree 35
2.3 Considerações Finais 36

3 MÉTODOS DE PLANEJAMENTO DE MOVIMENTO BASEADOS EM CAMPO POTENCIAL 39
3.1 Campo Potencial de Khatib 39
3.1.1 Força de Repulsão 40
3.1.2 Força de Atração 41
3.1.3 Força Resultante 42
3.2 Campo Potencial Harmônico 46
3.3 Campo Potencial Orientado 47
3.4 Campo Potencial Localmente Orientado 48
3.5 Considerações Finais 49

4 MODELO PROPOSTO 51
4.1 Campos Potencial Orientado Tridimensional 51
4.2 Prova de convergência e estabilidade 53
4.3 Considerações Finais 57

5 BIBLIOTECA DE PLANEJAMENTO DE MOVIMENTO 59
5.1 Módulos 61
5.2 Paralelismo 62

5.3 Considerações Finais 64

6 ANÁLISE DE DESEMPENHO 67

7 CONCLUSÃO E TRABALHO FUTUROS 71

REFERÊNCIAS 75

APÊNDICE A SOLUÇÃO NUMÉRICA DE SISTEMAS LINEARES . 81
A.1 Jacobi-Richardson 82
A.2 Gauss-Seidel 84
A.3 Weighted Jacobi 85
A.4 Successive Over-Relaxation 85

APÊNDICE B DEFINIÇÕES E TEOREMAS 87

INTRODUÇÃO

As pesquisas recentes em robótica possuem o objetivo de criar robôs móveis autônomos, capazes de interagir com o ambiente em que estão inseridos, executando suas tarefas com o mínimo de assistência humana, se possível nenhuma.

Com isto, em um primeiro momento, pode-se abordar as tarefas conhecidas como "*the three D's*", do inglês, "*Dirty, Dull and Dangerous*", isto é, tarefas maçantes, perigosas ou que sejam desagradáveis por algum motivo. E esta é, conforme o trabalho de Takayama, Ju e Nass (2008), justamente a visão popular do que um robô deve ser e se portar na sociedade. Intuitivamente, isso decorre até da origem da palavra robô, utilizada pela primeira vez na peça de teatro do checo Karel Čapek em (ČAPEK, 1920), descrevendo seres semelhantes a humanos que realizavam tarefas forçadas ("*robota*", do checo, tem um sentido de "trabalho forçado").

No trabalho de Sharkey (2008) este tipo de robô é chamado de *robô de serviço*, que pode executar as mais diversas tarefas, como limpeza de esgotos e janelas ou coleta de frutas. Ainda no trabalho de Sharkey (2008), diversos outros usos para robôs além dos "três D's" são apresentados e discutidas as repercussões morais do uso de robôs como armas ou para educação de crianças e cuidado com idosos.

Um aspecto importante desse tipo de robôs é a habilidade de se deslocarem pelo ambiente, planejando os movimentos que seus atuadores deverão fazer, para que cumpram suas tarefas, sem colidir com outros objetos ou agentes inseridos no mesmo ambiente.

Planejamento de movimento é uma área de pesquisa que visa obter uma sequência de posturas (ou estados) de um robô, a partir de uma postura inicial até uma postura meta, de forma a não colidir consigo mesmo (no caso de um robô com diversos graus de liberdade em seus atuadores), com obstáculos e outros agentes no mesmo ambiente. De um modo geral, espera-se que esta sequência não envolva movimentos desnecessários, como executar repetidos movimentos, sem que esses sejam necessários a execução de alguma tarefa.

A tarefa básica de planejamento de movimento costuma ser chamada de planejamento de caminhos, já que produz uma sequência de posturas, ou um caminho, entre a postura inicial e a postura alvo. Podem ser especificadas restrições sobre as possíveis sequências de movimentos, que resultam em problemas mais complexos. Algumas dessas restrições são:

- Obstáculos e outros robôs móveis; (YANG *et al.*, 2017; STEFANO; CLEMENTINI; STAGNINI, 2018)

- Múltiplos robôs; (ALONSO-MORA *et al.*, 2015; DECASTRO *et al.*, 2018)
- Restrições cinemático-dinâmicas; (ALONSO-MORA *et al.*, 2015)
- Incerteza em relação as posturas (inicial, meta e intermediárias), e demais aspectos do ambiente. (BERG; PATIL; ALTEROVITZ, 2017; JANSON; SCHMERLING; PAVONE, 2018)

Conforme o trabalho de Canny e Reif (1987) e Agarwal, Fox e Salzman (2018) a tarefa de planejamento de caminhos já é *NP-Hard* por si só, servindo como limitante inferior das tarefas mais restritivas ou complexas.

Quando este tipo de método é aplicado em ambientes dinâmicos, nos quais há obstáculos e/ou agentes móveis, se algumas condições forem atendidas, estes também devem funcionar. As condições costumam estar relacionadas a capacidade de replanejamento frente as mudanças no ambiente, bem como na capacidade dos atuadores do robô. Conforme Berg, Ferguson e Kuffner (2006), esta é uma maneira comum de lidar com ambientes dinâmicos, isto é, utilizando um planejador de caminhos rápido em um robô capaz de reagir rapidamente.

A execução em tempo real é uma característica importante de um sistema de planejamento de movimento. Caso não seja possível fornecer um bom caminho em tempo hábil, o sistema deve retornar um caminho factível e, se possível, continuar buscando por um bom (ou ótimo) caminho, que é conhecido na literatura por *anytime planning* (SCHLOTFELDT *et al.*, 2018).

Além das aplicações da robótica supracitadas, planejamento de movimento pode ser aplicado em áreas como a bioquímica, na simulação do dobramento de proteínas ("*protein folding*") (PARSONS; CANNY, 1994; BERGER; WALDISPÜHL, 2011; THOMAS; SONG; AMATO, 2005; SONG; AMATO, 2004; SONG; AMATO, 2001), que por sua vez auxiliam no entendimento de doenças como a *Doença de Alzheimer*, a *Doença de Parkinson*, *Diabetes Tipo II*, bem como outras doenças amilóides (KELLY, 1998).

A tarefa de planejamento de movimento é particularmente interessante para veículos aéreos não-tripulados (ou, do inglês, *Unmanned Aerial Vehicles - UAV*) , na qual uma colisão costuma ser muito mais danosa do que para um robô terrestre (GOERZEN; KONG; METTLER, 2010). Ainda, tais métodos podem ser utilizados em veículos aquáticos de superfície não-tripulados (ou, do inglês, *Unmanned Surface Vehicles - USV*), ou mesmo em veículos submarinos não-tripulados (ou, do inglês, *Unmanned Underwater Vehicles - UUV*).

Dentre as aplicações de robôs aéreos, pode-se ressaltar:

- **cidades inteligentes** "*Designing uav surveillance frameworks for smart city and extensive ocean with differential perspectives*" (KIM; MOKDAD; BEN-OTHMAN, 2018)
- **agricultura de precisão** "*Alocação de tarefas para a coordenação de robôs heterogêneos aplicados a agricultura de precisão*" (FRACCAROLI, 2018)

- **registro fotográfico aéreo** "*UAV photogrammetry for mapping and 3d modeling–current status and future perspectives*" (REMONDINO *et al.*, 2011)
- **serviços de entregas** "*Unmanned aerial vehicle delivery system*" (KIMCHI *et al.*, 2017)
- **monitoramento ambiental** "*Development of unmanned aerial vehicle (UAV) for wildlife surveillance*" (LEE, 2004)

Objetivos

Este trabalho abordou o estudo e aprimoramento de técnicas de planejamento de movimento baseadas em campos potenciais para robôs em ambientes tridimensionais, tendo como objetivo principal a extensão tridimensional do modelo de Campo Potencial Orientado, contemplando a formalização matemática e prova de estabilidade do mesmo, e como objetivo secundário a criação de biblioteca de planejamento de movimento, com os métodos estudados neste trabalho, implementada em C++, licenciada como GPL.

Organização do Texto

Este texto está organizado como a seguir:

- No Capítulo 1 é apresentado a formalização da tarefa de planejamento de movimento, em específico da tarefa mais simplificada, que é a de planejamento de caminhos
- No Capítulo 2 parte da revisão bibliográfica é feita, apresentando técnicas de planejamento de movimento probabilísticas.
- No Capítulo 3 outra parte da revisão bibliográfica é feita, sendo apresentadas as técnicas de planejamento de movimento baseadas na teoria de campo potencial.
- No Capítulo 4, é apresentado a extensão tridimensional do método de Campo Potencial Orientado, bem como garantias matemáticas de sua convergência e estabilidade
- No Capítulo 5 é apresentada a biblioteca de planejamento de movimento desenvolvida neste trabalho, bem como feitas considerações acerca de sua implementação paralela.
- No Capítulo 6 são apresentados os resultados computacionais da execuções dos algoritmos implementados.

- No Capítulo 7 são feitas as considerações finais do trabalho, bem como apresentados possíveis trabalhos futuros.
- Nos Apêndices A e B, são apresentadas parte da literatura que auxiliam na compreensão deste trabalho.

CAPÍTULO

1

PLANEJAMENTO DE MOVIMENTO

Neste capítulo, serão apresentados conceitos fundamentais para o entendimento e formalização matemática da tarefa de planejamento de movimento, bem como a descrição de um tarefa básica, **planejamento de caminhos para corpos rígidos**, em que tarefas mais complexas são superconjuntos desta, isto é, englobam a tarefa de planejamento de caminhos.

O conceito de **espaço de configuração**, a ser formalizado na Seção 1.1, é muito importante para a tarefa de planejamento de movimento, pois a partir dele é possível converter um problema geométrico complexo dentro de $\mathbb{R}^N$, com $N = 2,3$, em um problema mais simples, porém dentro de um espaço de maior dimensão. Também será apresentado o conceito de **caminho** e **caminho livre de obstáculos** na Seção 1.2, utilizando como base o conceito de espaço de configuração.

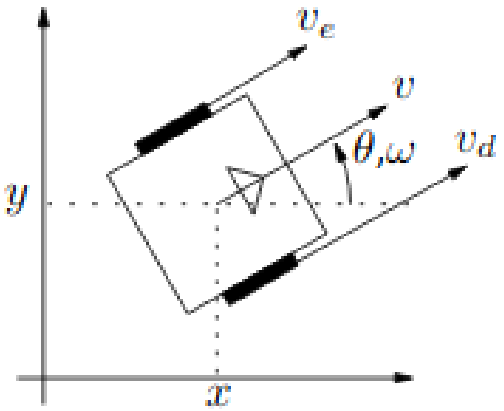
Segundo Spong, Hutchinson e Vidyasagar (2005), a tarefa de planejamento de movimento é descrita como uma tarefa aparentemente simples, porém se revela como uma das mais difíceis da computação, argumentando que entre os algoritmos de planejamento de movimento completos[1] mais conhecidos, a complexidade computacional cresce exponencialmente com o número de graus de liberdade de um robô e que, por tal motivo, para robôs com mais do que poucos graus de liberdade, algoritmos completos não são usados na prática, devido ao limite do poder computacional disponível para o agente.

Na Figura 1, um robô com 3 graus de liberdade, isto é, um espaço de configuração de dimensão 3, (x, y, θ), utilizado no futebol de robôs tem sua modelagem matemática discutida no trabalho de Cerqueira *et al.* (2006).

Em Cherubini *et al.* (2009), o robô NAO é modelado como tendo 21 graus de liberdade, como pode ser observado na Figura 2. São dois graus de liberdade na cabeça, quatro em cada braço, uma na pélvis e cinco em cada perna.

[1] Um algoritmos é dito ser completo se ele encontra uma solução caso ela exista, ou retorna algum tipo de erro em um tempo finito quando ela não existir

Figura 1 – Robô com espaço de configuração com 3 dimensões



Fonte: Cerqueira *et al.* (2006).

A tarefa de planejamento de movimento pode ser dividida em duas tarefas distintas:

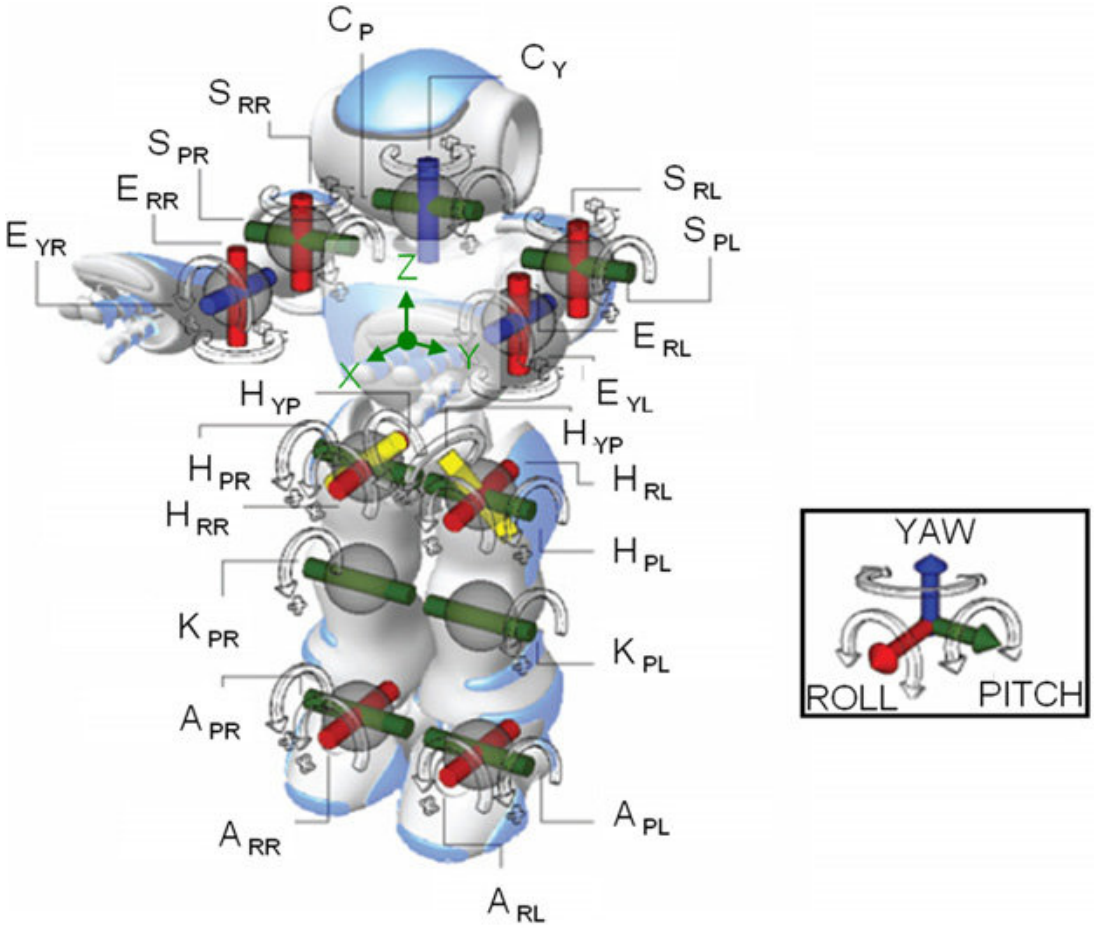
Planejamento de Caminhos Consiste em realizar o planejamento de movimento apenas considerando a geometria, tanto do agente a ser controlado quanto do ambiente e outros agentes inseridos dentro do mesmo.

Planejamento de Trajetórias Além de considerar a geometria, considera aspectos dinâmicos, como velocidade e aceleração no planejamento de movimento.

Conforme Latombe (1991), a tarefa de planejamento de movimento mais básica, isto é, planejamento de caminhos para corpos rígidos, resulta das seguintes especificações:

- Seja $\mathscr{A}$ um objeto rígido simples (o agente a ser controlado) se movimentando em um espaço euclidiano $\mathscr{W}$, chamado *Espaço de atuação* (ou *Workspace*), que representa $\mathbb{R}^N$, com $N = 2$ ou 3.
- Sejam $\mathscr{B}_1, \ldots, \mathscr{B}_z \subseteq \mathscr{W}$ objetos rígidos e estáticos, isto é obtáculos distribuídos por $\mathscr{W}$.
- Assuma que geometria de $\mathscr{A}, \mathscr{B}_i$ e localização de $\mathscr{B}_i$, com $i \in \{1, \ldots, z\}$, são conhecidas com grande precisão dentro de $\mathscr{W}$. Assuma ainda que não há restrições cinemáticas que limitem o movimento de $\mathscr{A}$.
- Então o problema consiste em, dada uma posição e orientação inicial e uma posição e orientação meta de $\mathscr{A}$ em $\mathscr{W}$, gerar um caminho γ especificando uma sequência continua de posições e orientações de $\mathscr{A}$, além disso deve ser evitado contato com os obstáculos

Figura 2 – Robô com espaço de configuração com 21 dimensões



Fonte: Cherubini *et al.* (2009).

$\mathscr{B}_i$, partindo da posição e orientação inicial e chegando a posição e orientação meta, ou reportar erro, se não existir caminho que satisfaça tais restrições.

1.1 Espaço de Configuração

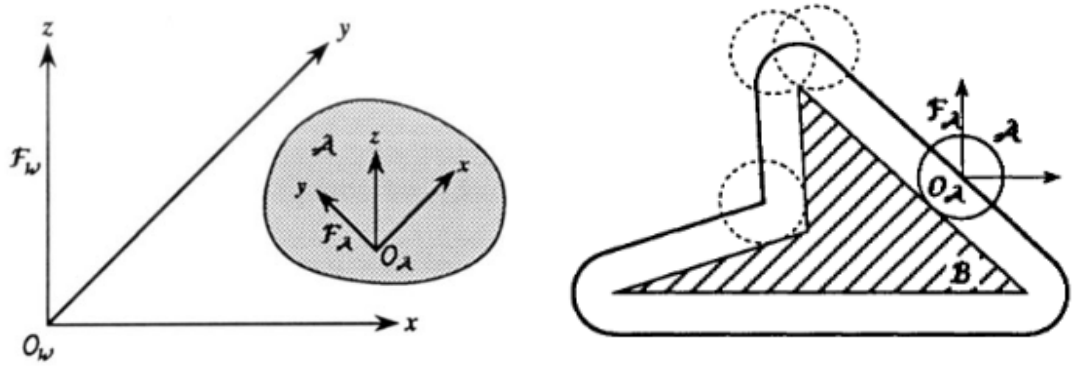
O conceito de espaço de configuração é muito importante para o planejamento de movimento. Através dele, a tarefa de encontrar um caminho para um agente bidimensional ou tridimensional inserido em um espaço bidimensional ou tridimensional é convertida na tarefa de achar um caminho para um agente pontual em um espaço de dimensão mais alta.

Nesta seção, será construído o espaço de configuração da tarefa de planejamento de caminhos para corpos rígidos, apresentado anteriormente. Em geral, quando mais complexa é a tarefa de planejamento de movimento, mais complexa se torna a descrição do espaço de configuração. Por exemplo, o planejamento de movimento para humanóides costuma ter um espaço de configuração com dezenas de dimensões (como na Figura 2).

Considerando ainda o problema básico, e adotando a notação apresentada anteriormente,

seja um robô $\mathscr{A}$, descrito como um subconjunto compacto de $\mathscr{W}$. Ainda, sejam $\mathscr{F}_{\mathscr{A}}$ e $\mathscr{F}_{\mathscr{W}}$ sistemas de coordenadas em $\mathscr{A}$ e $\mathscr{W}$ respectivamente. $\mathscr{F}_{\mathscr{A}}$ é um sistema de coordenadas móvel (pois o robô se movimenta), enquanto $\mathscr{F}_{\mathscr{W}}$ está fixo. Por definição, dado que $\mathscr{A}$ é rígido, todos os pontos $a \in \mathscr{A}$ tem uma posição fixa em relação a $\mathscr{F}_{\mathscr{A}}$. Porém, as posições dos pontos a em relação a $\mathscr{F}_{\mathscr{W}}$ dependem da posição e orientação de $\mathscr{F}_{\mathscr{A}}$ em relação a $\mathscr{F}_{\mathscr{W}}$. E, dado que todos os $\mathscr{B}_i$ são rígidos e fixos em $\mathscr{W}$, todos os pontos $b \in \mathscr{B}_i$ tem posição fixa em relação a $\mathscr{F}_{\mathscr{W}}$.

Figura 3 – Sistemas de coordenadas universal e de referência



Fonte: Latombe (1991).

Na Figura 3 a relação entre os sistemas de coordenadas universal e de referência podem ser visualizados, bem como a construção intuitiva de região de obstáculo $\mathscr{B}$.

A configuração de um objeto qualquer é a especificação da posição de todos os pontos deste objeto em relação a um sistema de coordenadas fixo. Desta forma, a **configuração** q de $\mathscr{A}$ é a especificação da posição $\mathscr{T}$ e orientação Θ, isto é, da postura de $\mathscr{F}_{\mathscr{A}}$ em relação a $\mathscr{F}_{\mathscr{W}}$. O **espaço de configuração** de $\mathscr{A}$ é o espaço $\mathscr{C}$ de todas as configurações de $\mathscr{A}$. O subconjunto de $\mathscr{W}$ ocupado por $\mathscr{A}$ em uma configuração q é denotado por $\mathscr{A}(q)$. Da mesma forma, um ponto a em $\mathscr{A}$ na configuração q é denotado por $a(q)$ em $\mathscr{W}$.

Descrever o espaço $\mathscr{C}$ através de uma lista de parâmetros propicia o uso de algoritmos para lidar com a tarefa de planejamento de movimento em que o robô se torna apenas um ponto q (pois com $\mathscr{A}(q)$ é possível encontrar a postura, isto é, posição e orientação, do robô no espaço de atuação $\mathscr{W}$).

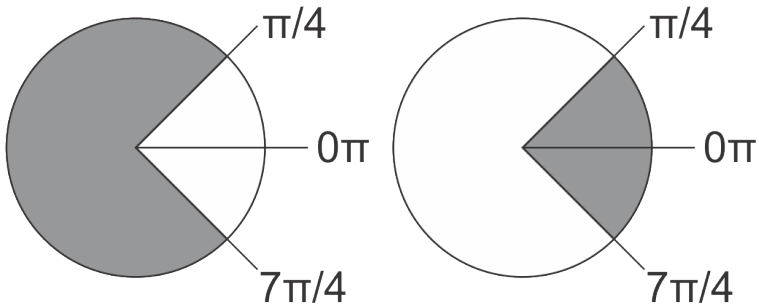
Uma das possíveis abordagens para descrever o espaço $\mathscr{C}$ é utilizar a posição $\mathscr{T}$, que pode ser descrita por um array de N coordenadas da origem de $\mathscr{F}_{\mathscr{A}}$ em $\mathscr{F}_{\mathscr{W}}$, e a orientação Θ, que pode ser descrita como uma matriz $N \times N$, na qual as colunas são componentes, em $\mathscr{F}_{\mathscr{W}}$, dos vetores unitários, ou base, de $\mathscr{F}_{\mathscr{A}}$.

Então, $q = (\mathscr{T}, \Theta)$ é representado de forma única por uma lista de $N(N+1)$ parâmetros. Entretanto, esta representação é redundante, pois a matriz que descreve Θ deve ter colunas ortonormais e determinante igual a 1 (LATOMBE, 1991). Com isto, é possível observar que

$\mathscr{C} \subseteq \mathbb{R}^{N(N+1)}$, isto é, $\mathscr{C}$ é um subconjunto próprio de $\mathbb{R}^{N(N+1)}$ e, portanto a dimensão de $\mathscr{C}$ é menor que a dimensão de $\mathbb{R}^{N(N+1)}$.

Outra possível abordagem é descrever o espaço $\mathscr{C}$ com uma lista de parâmetros mínima, isto é, um ângulo θ, se $N = 2$ (por exemplo, o ângulo entre os eixos x de $\mathscr{F}_{\mathscr{A}}$ e $\mathscr{F}_{\mathscr{W}}$), e três ângulos ϕ, θ e ψ, se $N = 3$ (por exemplo, os ângulos eulerianos (LATOMBE, 1991; LAVALLE, 2006)). Contudo, tal abordagem minimalista não é injetiva, isto é, a mesma configuração pode ser descrita por diferentes valores dos ângulos. Por exemplo, no caso bidimensional, dois ângulos θ_1 e θ_2 que difiram em 2π representam o mesmo Θ. Uma das formas de solucionar isso é restringir os valores do ângulo θ ao intervalo $[0, 2\pi)$ com aritmética módulo 2π, e levar em consideração que há continuidade e não uma ruptura em 2π, isto é, a distância entre os ângulos $\frac{\pi}{4}$ e $\frac{7\pi}{4}$ é $\frac{\pi}{2}$ e não $\frac{3\pi}{2}$, conforme a Figura 4. Conforme Latombe (1991), o caso tridimensional é substancialmente mais complexo de ser solucionado.

Figura 4 – Distância entre ângulos



(a) Distância considerando descontinuidade do intervalo

(b) Distância considerando a continuidade de S^1

Fonte: Elaborada pelo autor.

Descrever uma configuração q como uma lista de m parâmetros independentes, com $m = 3$ (se $N = 2$) e $m = 6$ (se $N = 3$), corresponde a representar $\mathscr{C}$ como $\mathbb{R}^m$. Entretanto, $\mathscr{C}$ é como $\mathbb{R}^m$ apenas localmente. O número m pode ser visto como a dimensão de $\mathscr{C}$, neste caso.

Conforme Latombe (1991), LaValle (2006), os obstáculos $\mathscr{B}_i$ podem ser mapeados em $\mathscr{C}$ como a região descrita na Equação 1.1, e desta forma podem ser chamados de **$\mathscr{C}$-obstáculos**.

$$\mathscr{C}\mathscr{B}_i = \{q \in \mathscr{C} | \mathscr{A}(q) \cap \mathscr{B}_i \neq \emptyset\} \tag{1.1}$$

A união de todos os $\mathscr{C}$-obstáculos, como na Equação 1.2, é chamada de **região de $\mathscr{C}$-obstáculos**.

$$\bigcup \mathscr{C}\mathscr{B}_i \tag{1.2}$$

Ainda, o espaço $\mathscr{C}$ subtraído da sua região de $\mathscr{C}$-obstáculos é chamado de **espaço livre**, ou $\mathscr{C}_{livre}$, e é definido como na Equação 1.3. Qualquer configuração q dentro de $\mathscr{C}_{livre}$ é chamada de **configuração livre de colisão**.

$$\mathscr{C}_{livre} = \mathscr{C} \setminus \bigcup \mathscr{C}\mathscr{B}_i = \left\{q \in \mathscr{C} | \mathscr{A}(q) \cap \left(\bigcup \mathscr{C}\mathscr{B}_i\right) = \emptyset\right\} \tag{1.3}$$

O conjunto $\overline{\mathscr{C}_{livre}}$ é o fecho do conjunto $\mathscr{C}_{livre}$, isto é $\overline{\mathscr{C}_{livre}} = \mathscr{C}_{livre} \cup \partial\mathscr{C}_{livre}$, com $\partial\mathscr{C}_{livre}$ sendo a fronteira de $\mathscr{C}_{livre}$, é também chamado de **espaço semi-livre de colisões**.

1.2 Caminho

Para formalizar o conceito de caminho, é fundamental ter estabelecido o conceito de continuidade do espaço $\mathscr{C}$, que por sua vez, depende de existir uma topologia em $\mathscr{C}$. Uma das maneiras de fazer isso (a que será usada neste trabalho) é especificar uma função distância $d : \mathscr{C} \times \mathscr{C} \to \mathbb{R}$, como na Equação 1.4, e induzir uma topologia através desta métrica. A distância d da Equação 1.4 foi adotada desta forma para coincidir com uma noção intuitiva do mundo real, na qual conforme duas configurações q e q' se aproximam e sua distância se aproxima de 0, as regiões $\mathscr{A}(q)$ e $\mathscr{A}(q')$ se aproximam até coincidirem.

$$d(q,q') = \max_{a \in \mathscr{A}} ||a(q) - a(q')|| \tag{1.4}$$

onde $||x - x'||$ denota a distância Euclidiana entre dois pontos x e x' em $\mathbb{R}^N$.

Desta forma, podemos estabelecer que um **caminho** de $\mathscr{A}$ a partir de uma configuração $q_{inicial}$ até uma configuração q_{final} é uma aplicação contínua $\gamma : [0,1] \to \mathscr{C}$, sujeita a $\gamma(0) = q_{inicial}$ e $\gamma(1) = q_{final}$. As configurações $q_{inicial}$ e q_{final} são chamadas de **configuração inicial** e **configuração meta** (ou **configuração alvo**), respectivamente.

A continuidade de γ implica que para todo $s, s_0 \in [0,1]$, com s_0 fixo,

$$\lim_{s \to s_0} \max_{a \in \mathscr{A}} ||a(\gamma(s)) - a(\gamma(s_0))|| = 0$$

Um **caminho livre de colisões** a partir de $q_{inicial}$ até q_{final} é uma aplicação contínua $\gamma : [0,1] \to \mathscr{C}_{livre}$, sujeita a $\gamma(0) = q_{inicial}$ e $\gamma(1) = q_{final}$

Em algumas técnicas de planejamento de movimento é admitido o contato (mas não a colisão) do agente a ser controlado com o obstáculo, gerando um **caminho semi-livre de colisões**. Um caminho semi-livre de colisões a partir de $q_{inicial}$ até q_{final} é uma aplicação contínua $\gamma : [0,1] \to \overline{\mathscr{C}_{livre}}$, sujeita a $\gamma(0) = q_{inicial}$ e $\gamma(1) = q_{final}$.

1.2.1 Extensão do problema básico: Obstáculos móveis

Nesta sub-seção serão apresentadas as principais modificações na modelagem da tarefa de planejamento de movimento ao se considerar que obstáculos (que podem ser outros robôs, mas que não são controláveis) podem se deslocar no ambiente enquanto o agente sendo controlado se direciona a sua meta.

Ainda considerando um agente $\mathscr{A}$, um objeto rígido se movimentando em um ambiente de atuação $\mathscr{W} = \mathbb{R}^N$, com $N = 2$ ou 3. $\mathscr{W}$ contém obstáculos $\mathscr{B}_i$, com $i = 1, \ldots, z$, os quais

podem estar em movimento, ou não. $\mathscr{B}_i(t)$ denota que região de $\mathscr{W}$ é ocupada pelo objeto $\mathscr{B}_i$ no tempo t (com $t \geq 0$) e, ainda, é assumido o conhecimento a priori das geometria (invariante no tempo) e da localização (variante no tempo) das regiões $\mathscr{B}_i(t)$. Com isto, é suposto que o movimento dos obstáculos $\mathscr{B}_i$ não é afetado pelo movimento do agente $\mathscr{A}$.

Seja $\mathscr{C}$ o espaço de configuração de $\mathscr{A}$ como construído na Seção 1.1. A cada instante t, o obstáculo $\mathscr{B}_i(t)$ é mapeado em $\mathscr{C}$ como o $\mathscr{C}$-obstáculo dado pela Equação 1.5.

$$\mathscr{C}\mathscr{B}_i(t) = \{q \in \mathscr{C} | \mathscr{A}(q) \cap \mathscr{B}_i(t) \neq \emptyset\} \tag{1.5}$$

Então, a tarefa de planejamento de movimento consiste em obter um caminho livre de colisões para $\mathscr{A}$ a partir de uma configuração inicial $q_{inicial}$ em um tempo inicial $t_{inicial}$, até uma configuração meta q_{final} em um tempo final $t_{final} \geq t_{inicial}$. Para simplificar a notação a seguir, pode ser assumido sem perda de generalidade que $t_{inicial} = 0$ e $t_{final} = T \geq 0$. Diferente do problema básico e as variações apresentadas anteriormente, é necessário especificar a configuração de $\mathscr{A}$ em cada instante de tempo no intervalo $[0,T]$. Formalmente, esta tarefa pode ser vista como encontrar uma aplicação contínua $q(t)$ conforme a Equação 1.6, restrita a $q(0) = q_{inicial}$ e $q(T) = q_{final}$.

$$q(t) : [0,T] \rightarrow \mathscr{C} \setminus \bigcup \mathscr{C}\mathscr{B}_i(t) \tag{1.6}$$

A aplicação $q(t)$ como na Equação 1.6 é chamada de **trajetória de** $\mathscr{A}$. Posto desta forma, a tarefa de planejamento de movimento também é conhecida por **planejamento de trajetória**.

Como a localização dos $\mathscr{C}$-obstáculos pode variar com t, eles não podem ser representados em $\mathscr{C}$ de forma que o planejamento de movimento de $\mathscr{A}$ em $\mathscr{W}$ possa ser transformado na tarefa de planejamento de movimento de uma partícula em $\mathscr{C}$. Entretanto, isto pode ser solucionado ao se adicionar uma dimensão temporal a $\mathscr{C}$. Desta forma, é obtido um novo espaço de configuração, $\mathscr{C}\mathscr{T} = \mathscr{C} \times [0,+\infty)$, que é chamado de **espaço de configuração temporal** de $\mathscr{A}$. O espaço $\mathscr{C}\mathscr{T}$ é uma *variedade suave* (MANFREDO, 1976) de dimensão $m+1$, onde m é a dimensão de $\mathscr{C}$ (HATCHER, 2002). Uma aplicação continua $\xi : [0,1] \rightarrow \mathscr{C}\mathscr{T}$ é chamado de **$\mathscr{C}\mathscr{T}$-caminho**.

Então, o agente $\mathscr{A}$ é mapeado no espaço $\mathscr{C}\mathscr{T}$ como uma **configuração temporal** (q,t). Isto significa que no instante t, o agente $\mathscr{A}$ está na configuração q em $\mathscr{C}$. A cada instante t, as seções transversais de $\mathscr{C}\mathscr{T}$, isto é, o estado do espaço $\mathscr{C}$ no tempo t, há os $\mathscr{C}$-obstáculos $\mathscr{C}\mathscr{B}_i(t)$. Então, cada obstáculo $\mathscr{B}_i$ é mapeado em $\mathscr{C}\mathscr{T}$ em uma região $\mathscr{C}\mathscr{T}\mathscr{B}_i$, chamada $\mathscr{C}\mathscr{T}$-obstáculo, definida como na Equação 1.7

$$\mathscr{C}\mathscr{T}\mathscr{B}_i = \{(q,t) | \mathscr{A}(q) \cap \mathscr{B}_i(t) \neq \emptyset\} \tag{1.7}$$

Então, o **espaço livre** em $\mathscr{CT}$ é definido como na Equação 1.8

$$\mathscr{CT}_{livre} = \mathscr{CT} \setminus \bigcup \mathscr{CT}\mathscr{B}_i \tag{1.8}$$

O fecho de $\mathscr{CT}_{livre}$, $\overline{\mathscr{CT}_{livre}}$, é chamado de **$\mathscr{CT}$-espaço semi-livre de colisões**.

Uma aplicação contínua $\xi : [0,1] \to \mathscr{CT}_{livre}$ (respectivamente $\overline{\mathscr{CT}_{livre}}$) é chamado de **$\mathscr{CT}$-caminho livre de colisões** (respectivamente **$\mathscr{CT}$-caminho semi-livre de colisões**). Desta forma, a tarefa de planejamento de movimento com obstáculos móveis pode ser vista como a construção de um $\mathscr{CT}$-caminho (livre ou semi-livre) entre duas configurações temporais $(q_{inicial}, 0)$ e (q_{final}, T).

Ainda, pode ser vista como encontrar um caminho entre objetos estáticos, isto é, os $\mathscr{CT}\mathscr{B}_i$. Entretanto, vale ressaltar que há diferenças importantes em relação ao problema básico de planejamento de movimento, pois nenhum objeto pode se mover no tempo para trás, um $\mathscr{CT}$-caminho ξ deve ser estritamente monotônico crescente em relação ao tempo, isto é, $\forall n, m \in [0,1]$, seja $\xi(s) = (q(t(s)), t(s))$, então se $n < m$ implica que $t(n) < t(m)$.

Entretanto, apenas restringir que o $\mathscr{CT}$-caminho seja estritamente monotônico crescente pode não ser suficiente, pois não há um limite para a velocidade e aceleração de $\mathscr{A}$.

Em muitas situações reais (ou mesmo simuladas), nem sempre é verdade que o movimento dos obstáculos $\mathscr{B}_i$ não possa ser afetado pelo movimento do agente $\mathscr{A}$. Uma das formas de mitigar tal efeito, apresentada em Latombe (1991), é lidar com a incerteza nos sensores a partir dos quais o agente $\mathscr{A}$ obtém informação de $\mathscr{W}$, já que em algumas tarefas, o efeito do movimento de $\mathscr{A}$ sobre o movimento de $\mathscr{B}_i$ é impercetível devido ao ruído proveniente dos sensores. Encontrar um $\mathscr{CT}$-caminho com restrições a velocidade e aceleração torna o problema de planejamento de trajetórias extremamente mais complexo. Estas classes de tarefas não serão tratadas neste trabalho.

1.3 Considerações Finais

Neste capítulo foram apresentadas as definições e formalizações matemáticas necessárias para a compreensão dos métodos apresentados nos Capítulos 2 e 3.

Tal formalização matemática é importante para que se possa garantir as condições da prova de estabilidade e convergência do modelo proposto neste trabalho, no Capítulo 4.

CAPÍTULO

2

TRABALHOS CORRELATOS

A área de robótica como um todo tem se apresentado como uma área de pesquisa capaz de revolucionar outras áreas do saber humano. Por exemplo, no trabalho de Fraccaroli (2018), que apresenta a análise e modelo de um sistema de alocação de tarefas para multiplos robôs com aplicação na agricultura. Tal sistema permite que diferentes robôs, com distintas capacidades, sensores, atuadores e etc, cooperem na realização da tarefa de agricultura de precisão, que resulta em menor impacto ao meio ambiente, menores custos ao produtor e maior produtividade.

2.1 Probabilistic Road Maps

Sugerido no trabalho de Kavraki *et al.* (1996), o método de planejamento de caminhos para espaços de configuração de alta dimensão estáticos chamado de *Probabilistic Roadmaps* (PRM), ou em tradução livre, *Mapas Rodovários Probabilísticos* continua sendo um método bastante utilizado (LEHNER; ALBU-SCHÄFFER, 2018; EKENNA *et al.*, 2015; BEKRIS *et al.*, 2015).

A versão mais básica do método, apresentada por Kavraki *et al.* (1996), consiste de duas fases:

Aprendizado Um grafo no qual os nós correspondem a configurações livres de colisão no espaço de configuração e as arestas correspondem a caminhos possíveis entre essas configurações é criado. Esses caminhos são gerados utilizando um planejador de caminhos local rápido. O grafo resultante é do tipo floresta, isto é, não tem ciclos e é uma coleção de árvores.

Planejamento Dadas as posturas inicial e meta, estas são conectadas ao grafo através de planejador de caminhos local, que pode ser o mesmo da fase anterior, e, em seguida, é verificado dentro do grafo um caminho conectando estes dois nós. Caso não seja possível conectar os nós correspondentes a postura inicial e meta, informar que não foi possível encontrar um caminho.

A fase de Aprendizado é melhor detalhada no Algoritmo 1.

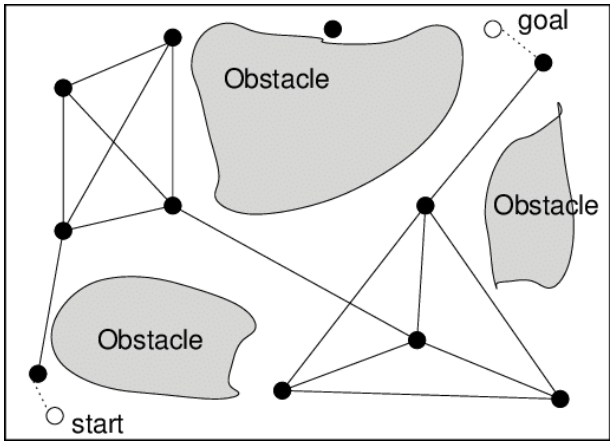
Algoritmo 1 – Procedimento para criação de PRMs

1: **procedimento** FASE DE APRENDIZADO
2: $\mathscr{N}_{prm} \leftarrow 0$ ▷ $\mathscr{N}_{prm}$ conjunto de nós
3: $\mathscr{A}_{prm} \leftarrow 0$ ▷ $\mathscr{A}_{prm}$ conjunto de arestas
4: **laço**
5: $q \leftarrow RANDOM_CONFIG(\mathscr{C}_{free})$ ▷ Uma configuração aleatória em $\mathscr{C}_{free}$
6: $\mathscr{N}^{q}_{prm} \leftarrow \mathscr{N}_{prm} \cup \{q\}$
7: **para todo** $n \in \mathscr{N}^{q}_{prm}$, em ordem crescente por $D_{prm}(q,n)$ **faça** ▷ $D_{prm}(q,n)$ é uma função de distância no espaço $\mathscr{C}_{free}$
8: **se** $\neg\, CONNECTED_COMPONENT(q,n)$ **então** ▷ $CONNECTED_COMPONENT(q,n)$ indica se as configurações q e n já estão conectadas no grafo
9: **se** $MOTION_PLAN(q,n)$ **então** ▷ $MOTION_PLAN(q,n)$ planejador local que indica se é possível conectar q e n, isto é, há um caminho entre as duas configurações
10: $\mathscr{A}_{prm} \leftarrow \mathscr{A}_{prm} \cup \{(q,n)\}$
11: **fim se**
12: **fim se**
13: **fim para**
14: **fim laço**
15: **fim procedimento**

Fonte: Adaptada de Kavraki *et al.* (1996).

Ainda no trabalho de Kavraki *et al.* (1996), a descrição do PRM não especifica alguns detalhes como, por exemplo, que planejador local utilizar, mas que deveria possuir as características de ser determinístico e rápido. Porém, no trabalho de Geraerts e Overmars (2002) diversas possíveis implementações do PRM são apresentadas e testadas em benchmarks.

Figura 5 – Fase de planejamento de um PRM



Fonte: Burchan Bayazit, e Amato (2002).

Conforme a Figura 5, é possível ver como a fase de planejamento, após a fase de

aprendizado, do PRM é executada. A postura inicial (*start*) e a postura final (*goal*) são conectadas ao grafo *roadmap* e o robô deverá procurar um caminho apenas entre esses pontos.

2.2 Rapidly-exploring Random Tree

Proposto no trabalho de LaValle (1998), *Rapidly-exploring Random Tree*, ou simplesmente RRT, é um planejador de movimento que contém uma estrutuda de dados de crescimento randomizado, que permite buscas rápidas em espaços de alta dimensionalidade.

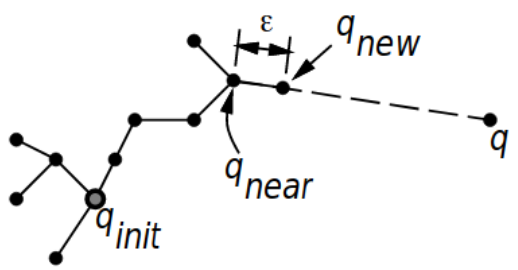
Na construção da árvore no espaço de configuração $\mathscr{C}$ de um determinado agente, é possível explicitar restrições algébricas (provenientes de obstáculos), evitando o crescimento dentro do subespaço $\mathscr{C}$-obstáculos, fazendo com que toda a árvore esteja contida em $\mathscr{C}_{livre}$.

Algoritmo 2 – Procedimento para criação de RRTs

1: **procedimento** BUILD_RRT(q_{init})
2: $T.init(q_{init})$
3: $k \leftarrow 1$
4: **enquanto** $k <= K$ **faça**
5: $q_{rand} \leftarrow RANDOM_CONFIG()$ ▷ $RANDOM_CONFIG()$ gera uma configuração aleatória no Espaço $\mathscr{C}_{livre}$
6: $EXTEND_RRT(T, q_{rand})$ ▷ O procedimento auxiliar $EXTEND_RRT$ é detalhado no Algoritmo 3 e visualizado na Figura 6
7: $k \leftarrow k+1$
8: **fim enquanto**
9: **retorna** T
10: **fim procedimento**

Fonte: Adaptada de Kuffner J.J. e LaValle (2000).

Figura 6 – O procedimento auxiliar $EXTEND_RRT$ - Algoritmo 3



Fonte: Kuffner J.J. e LaValle (2000).

O procedimento do Algoritmo 3 é ilustrado na Figura 6, que mostra a expansão da árvore na direção de q, através do nó mais próximo q_{near}, com uma distancia fixa ε. A postura q é gerada aleatoriamente no Algoritmo 2.

Algoritmo 3 – Procedimento auxiliar para criação de RRTs: $EXTEND_RRT$

1: **procedimento** $EXTEND_RRT(T,q)$
2: $q_{proximo} \leftarrow NEAREST_NEIGHBOR(T,q)$ ▷ $NEAREST_NEIGHBOR(T,q)$ É uma função construída para devolver o nó da árvore T mais próximo de q
3: **se** $NEW_CONFIG(q, q_{near}, q_{new})$ **então** ▷ o procedimento NEW_CONFIG faz um movimento em direção a q com uma determinada e fixada distância ε e verifica por colisões
4: $T.add_vertex(q_{new})$ ▷ Adiciona o vértice q_{new} na árvore T
5: $T.add_edge(q_{near}, q_{new})$ ▷ Adiciona a aresta $q_{near}, q_{new})$ na árvore T
6: **se** $q_{new} = q$ **então**
7: **retorna** *Reached* ▷ Retorna informando que o vértice q (gerado aleatoriamente no Algoritmo 2 na Linha 5) foi alcançado
8: **senão**
9: **retorna** *Advanced* ▷ Retorna informando que um vértice q_{new}, ε em direção a q, foi adicionado
10: **fim se**
11: **fim se**
12: **retorna** *Trapped* ▷ Retorna informando que não foi possível avançar em direção a q a partir de q_{near}
13: **fim procedimento**

Fonte: Adaptada de Kuffner J.J. e LaValle (2000).

Conforme Kuffner J.J. e LaValle (2000), a ideia central do método RRT (Algoritmo 2), é explorar o espaço $\mathscr{C}$ com viés para as regiões não visitadas do espaço. Com isto, RRTs fazem uma cobertura uniforme do espaço $\mathscr{C}$.

No trabalho de Kuwata *et al.* (2008), RRTs são utilizadas para planejamento de movimento de um carro dentro de ambiente urbano, considerando restrições cinemático-dinâmicas, bem como restrições de legislação de trânsito no planejamento. Tal resultado demonstra a flexibilidade do método em incorporar conceitos de alto nível, como legislação de trânsito, e conceitos de baixo nível, como restrições cinemático-dinâmicas do agente a ser controlado.

No trabalho de Moon e Chung (2015) é apresentado uma modificação do RRT para controle de robôs de acionamento diferencial (com duas rodas) que comporta replanejamento dinâmico, através de um modelo com duas árvores simultâneas, uma para planejamento de caminhos e outra para de trajetórias.

2.3 Considerações Finais

Neste Capítulo foram apresentadas duas das técnicas de planejamento de movimento mais utilizadas nas tarefas de planejamento de movimento em ambientes tridimensionais (GOERZEN; KONG; METTLER, 2010), a PRM e a RRT.

Ressalta-se que tais técnicas são de natureza probabilística, entretanto ambas possuem uma etapa que dependem de um planejador de movimento local rápido e determinístico (KA-

VRAKI *et al.*, 1996; LAVALLE, 1998), a saber, $MOTION_PLAN(q,n)$ no Algoritmo 1 e $NEW_CONFIG(q,q_{near},q_{new})$ no Algoritmo 3. Estes dois passos, descritos como pseudo-código de alto nível nos respectivos Algoritmos, podem ser resolvidos por, por exemplo, os métodos de campos potenciais, objeto de estudo deste presente trabalho.

CAPÍTULO

3

MÉTODOS DE PLANEJAMENTO DE MOVIMENTO BASEADOS EM CAMPO POTENCIAL

Neste Capítulo, serão apresentados métodos de planejamento de movimento que solucionam a tarefa básica apresentada no Capítulo 1.

Preliminarmente, serão apresentados os métodos de Campo Potencial de Khatib (CPK) (KHATIB, 1980), Campo Potencial Harmônico (CPH) (CONNOLLY; BURNS; WEISS, 1990; MOTONAKA *et al.*, 2016; TON; KAN; MEHTA, 2018), Campo Potencial Orientado (CPO) (PRESTES, 2003) e Campo Potencial Localmente Orientado (CPLO) (FARIA, 2006), inspirações diretas para o desenvolvimento deste trabalho.

Os métodos CPH e CPO dependem de soluções numéricas de equações diferenciais parciais elípticas (EDPs elípticas) e, um resumo da base matemática necessária (para compreender algumas das formalizações que serão apresentadas) pode ser vista em Silva (2011).

3.1 Campo Potencial de Khatib

O Campo Potencial de Khatib, ou simplesmente Campo Potencial (quando não houver confusão quanto ao nome da família de técnicas), é a técnica de planejamento de movimento baseada em campo potencial mais antiga, sendo primeiramente descrita em Khatib (1980).

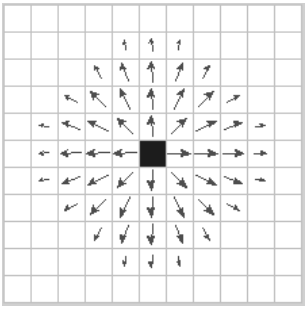
Neste método, obstáculos exercem forças repulsivas e, a meta aplica uma força atrativa sobre o robô, então, calcula-se a força resultante sobre o robô. No caso bidimensional, a força resultante pode ser decomposta e transformada em uma componente angular e componente linear do movimento do robô, e ser repassada para algum sistema de controle dentro do robô.

No trabalho de Faria (2006), essas forças são calculas da seguinte forma:

3.1.1 Força de Repulsão

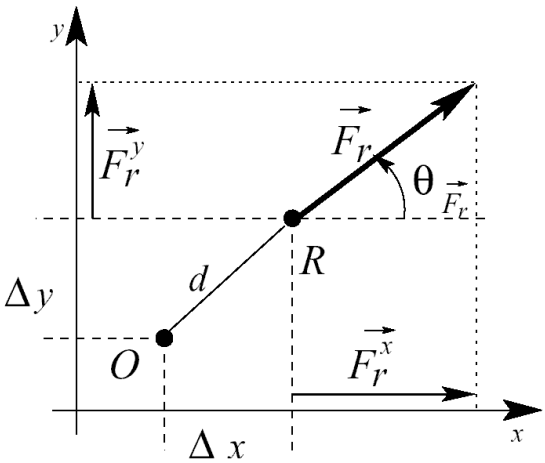
Na Figura 7 mostra-se que a força de repulsão decai ao se afastar de um obstáculo, pois a força de repulsão ($\overrightarrow{F_r}$) é definida como um vetor cujo módulo é inversamente proporcional ao quadrado da distância (d) entre o robô (R) e objeto observado (O) (Figura 8).

Figura 7 – Força repulsiva na vizinhança de um obstáculo



Fonte: Faria (2006).

Figura 8 – Força repulsiva do objeto O sobre um robô localizado em R



Fonte: Faria (2006).

O vetor $\overrightarrow{F_r}$ também pode ser representado por suas componentes: módulo e direção, apresentados nas Equações 3.1 e 3.2.

$$|\overrightarrow{F_r}| = \frac{Q}{d^2} \tag{3.1}$$

$$\theta_{\overrightarrow{F_r}} = arctan2\,(-\Delta y, -\Delta x) \tag{3.2}$$

sendo que: Q representa um escalar constante de repulsão; $arctan2$ é uma função na qual se considera os sinais de Δx e Δy para fornecer o ângulo correto cuja tangente seja $(-\Delta y/-\Delta x)$; e os sinais negativos de Δx e Δy fornecem uma direção oposta ao objeto detectado (O).

Para calcular a força de repulsão para vários obstáculos, deve-se fazer o somatório dos vetores das forças de repulsão geradas, como mostrado na Equação 3.3.

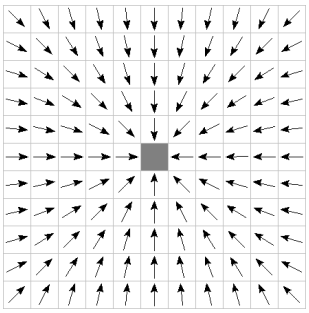
$$\overrightarrow{F_R} = \sum_i \left(\overrightarrow{F_r}\right)_i \tag{3.3}$$

onde $\left(\overrightarrow{F_r}\right)_i$ é a força de repulsão agindo sobre o robô devido ao objeto i.

3.1.2 Força de Atração

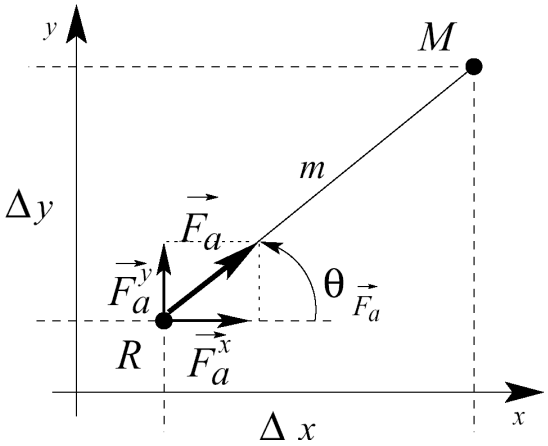
Nas Figuras 9 mostra-se o campo potencial de atração, no qual, conforme Figura 10 deve-se considerar $|\overrightarrow{F_a}|$ constante para que o agente seja atraído pela meta mesmo estando distante da mesma(Equações 3.4 e 3.5).

Figura 9 – Força atrativa constante



Fonte: Faria (2006).

Figura 10 – Força atrativa entre o robô R e a meta M



Fonte: Faria (2006).

$$|\overrightarrow{F_a}| = C \tag{3.4}$$

$$\theta_{\overrightarrow{F_a}} = arctan2\left(\Delta y, \Delta x\right) \tag{3.5}$$

no qual C representa um escalar constante de atração. É possível ser notado que Δx e Δy diferem da Equação 3.2 por não estarem acompanhados do sinal negativo, resultando, então, na direção do vetor força de atração ser na direção da meta.

Ressalta-se que em Faria (2006) não é citado a somatória de forças atrativas. Tal soma resultaria em multiplos objetivos simultâneos, podendo levar a comportamentos não desejados.

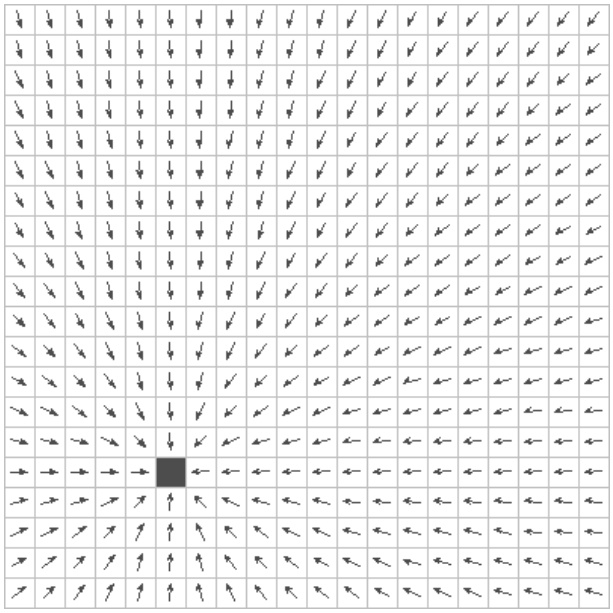
3.1.3 Força Resultante

Ao realizar a soma dos vetores força de atração e força de repulsão obtém-se a força resultante dada pela Equação 3.6.

$$\overrightarrow{F} = \overrightarrow{F_R} + \overrightarrow{F_a} \tag{3.6}$$

Para ilustrar a superposição dos campos de atração e repulsão, mostra-se na Figura 11 o campo de atração gerado pela meta, na Figura 12 o campo de repulsão gerado por dois obstáculos e na Figura 13 a soma dos campos de atração e de repulsão.

Figura 11 – Potencial atrativo da meta

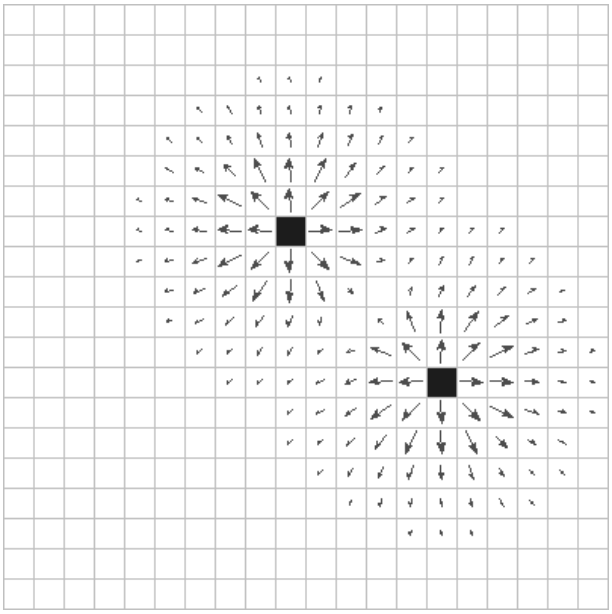


Fonte: Faria (2006).

Tendo em vista que este é um método de baixo custo computacional (pois realiza apenas arimética simples de vetores, com estes sendo calculados a partir de equações relativamente simples), esta técnica foi muito utilizada em planejamento de movimento de robôs móveis que necessitavam navegar por ambientes desconhecidos e/ou dinâmicos evitando colisões com obstáculos (PRESTES, 2003).

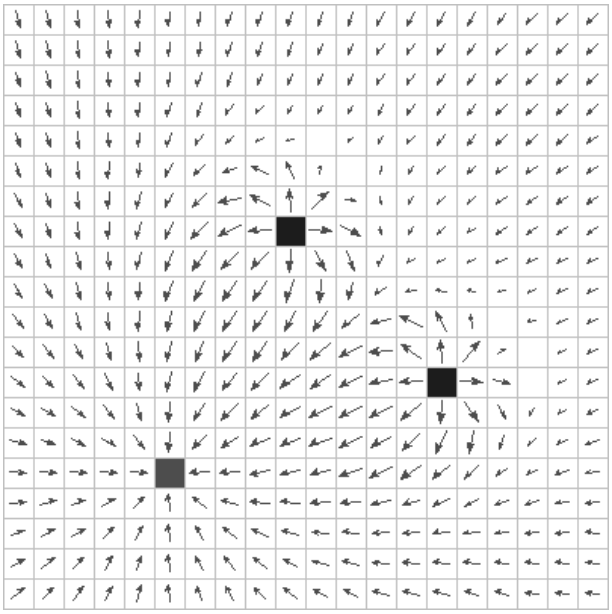
Entretanto, tal técnica costuma levar a regiões de mínimo local (PRESTES, 2003). Essas regiões são locais nos quais a resultante da soma das forças atuando sobre o robô é nula, entretanto o robô ainda não chegou a sua meta, conforme Figuras 14 a 17. Na Figura 16 e na

Figura 12 – Potencial repulsivo de dois obstáculos



Fonte: Faria (2006).

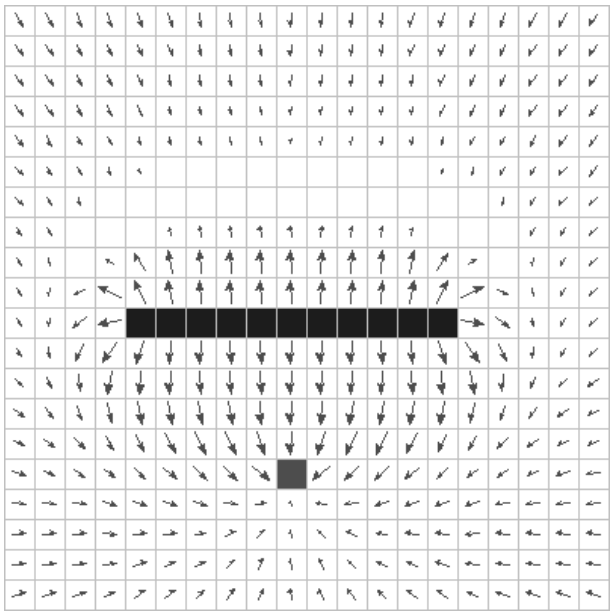
Figura 13 – Soma dos campos vetoriais das Figuras 11 e 12



Fonte: Faria (2006).

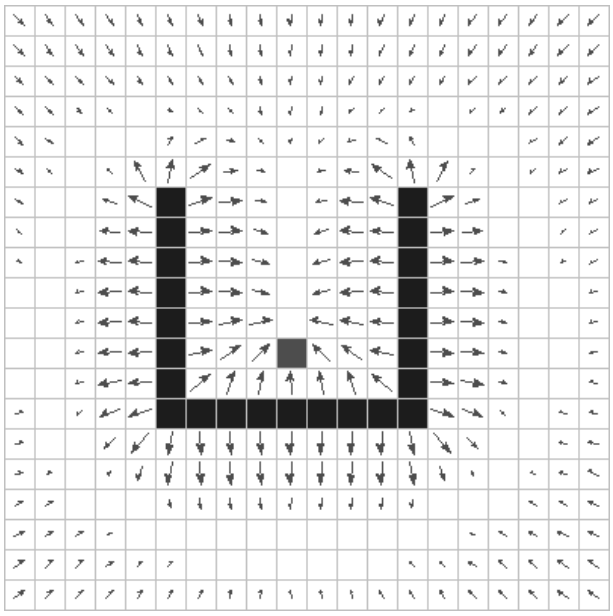
Figura 17 o ambiente a ser navegado é o mesmo, porém, a constante de repulsão dos obstáculos foi modificada e, embora não exista mais o mínimo local da Figura 16 na Figura 17, também não há mais rota entre a parte superior do ambiente e a parte inferior.

Figura 14 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 1)



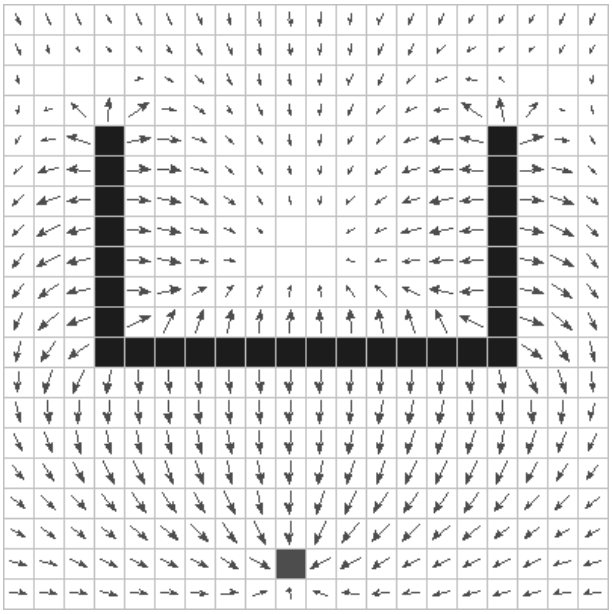
Fonte: Faria (2006).

Figura 15 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 2)



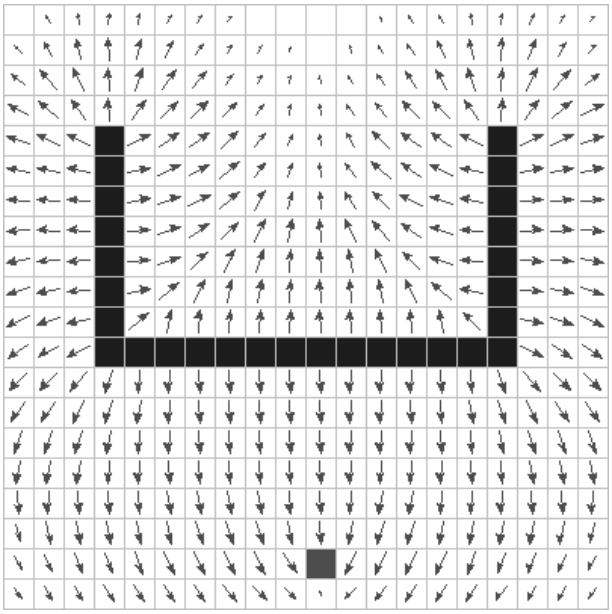
Fonte: Faria (2006).

Figura 16 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 3 - Situaçao 1)



Fonte: Faria (2006).

Figura 17 – Mínimos locais gerados pela anulação das forças de atração e repulsão. (Ambiente 3 - Situação 2)



Fonte: Faria (2006).

3.2 Campo Potencial Harmônico

Para superar o problema de mínimos locais, em Connolly, Burns e Weiss (1990) um novo método de campo potencial, baseado em funções harmônicas resultantes da solução da Equação de Laplace (Equação 3.7, (GILBARG; TRUDINGER, 1983)) foi utilizado para o caso bidimensional.

$$\nabla^2 P = \frac{\partial^2 P(x,y)}{\partial x^2} + \frac{\partial^2 P(x,y)}{\partial y^2} = 0 \qquad \text{para } P : \mathbb{R}^2 \mapsto \mathbb{R} \tag{3.7}$$

Tal método, chamado de Campo Potencial Harmônico (CPH), é um método de planejamento de movimento baseado na solução de uma equação diferencial parcial elíptica na qual se estabelece um Problema de Valor de Contorno (PVC) com condição de Dirichlet (GILBARG; TRUDINGER, 1983; SMITH, 1992; FIGUEIREDO, 2005; STRAUSS, 1992), com obstáculos possuindo alto potencial (normalmente 1), e a meta possuindo baixo potencial (normalmente 0). Conforme Connolly, Burns e Weiss (1990), devido a dificuldade em se encontrar uma solução analítica para tal PVC, o espaço $\mathscr{C}_{livre}$ é discretizado numa malha de células quadradas de igual tamanho e, aplicando o método das diferenças finitas (FRANCO, 2008). Uma aproximação via solução numérica de equações diferenciais parciais é obtida e, de posse desta solução, o gradiente descendente é seguido, pois tem a propriedade de levar até a meta, caso exista um caminho até ela (FIGUEIREDO, 2005).

Então, conforme apresentado em Silva (2011), obtém-se a molécula computacional da Equação 3.8. Para simplificar a notação, os índices dimensionais serão omitidos quando eles corresponderem a i ou j , isto é, $p_{i,j}$ e $p_{i,j+1}$ serão representados por p e p_{j+1}, respectivamente.

$$p_{i,j} = \frac{p_{j+1} + p_{j-1} + p_{i+1} + p_{i-1}}{4} \tag{3.8}$$

A partir da Equação 3.8, o método de relaxação de Gauss-Seidel e o Successive Over-Relaxation (SOR) (FRANCO, 2008) podem ser aplicados, com isso resultado nas Equações 3.9 e 3.11. Maiores detalhes desses métodos são apresentados no Apêndice A.

$$p_{GS}^{t+1} = \frac{p_{i+1}^{t} + p_{i-1}^{t+1} + p_{j+1}^{t} + p_{j-1}^{t+1}}{4} \tag{3.9}$$

$$\begin{aligned} p_{SOR}^{t+1} &= p_{GS}^{t+1} + \lambda_{sor}(p_{GS}^{t+1} - p^{t}) \\ &= (1+\lambda_{sor})p_{GS}^{t+1} - \lambda_{sor}p^{t} \end{aligned} \tag{3.10}$$

Tomando $\omega = (1+\lambda_{sor})$,

$$\begin{aligned} p_{SOR}^{t+1} &= \omega p_{GS}^{t+1} - (\omega - 1)p^t \\ &= \omega p_{GS}^{t+1} - \omega p^t + p^t \\ &= \frac{\omega}{4}(p_{i+1}^t + p_{i-1}^{t+1} + p_{j+1}^t + p_{j-1}^{t+1} - 4p^t) + p^t \end{aligned} \tag{3.11}$$

com $0 < \omega < 2$.

No trabalho de Connolly e Grupen (1993), os experimentos realizados adotaram $\omega = 1.8$ e em Press *et al.* (1992) uma equação para calcular o valor ótimo para ω foi proposta a partir do conhecimento a priori do número de linhas e de colunas da grade discreta do ambiente.

3.3 Campo Potencial Orientado

As soluções da Equação de Laplace (Equação 3.7) não apresentam mínimos locais (FIGUEIREDO, 2005), no entanto, este é apenas um caso particular de um conjunto maior de funções que também apresentam esta característica. Em Prestes (2003), é apresentada uma família de soluções de uma EDP elíptica (Equação 3.12), que também não apresenta mínimos locais.

$$\nabla^2 P + \varepsilon \langle \nabla P, v \rangle = \frac{\partial^2}{\partial x^2} P(x,y) + \frac{\partial^2}{\partial y^2} P(x,y) + \varepsilon \left(\frac{\partial P(x,y)}{\partial x} v_x + \frac{\partial P(x,y)}{\partial y} v_y \right) = 0 \tag{3.12}$$

com $\varepsilon \in \mathbb{R}$, $v \in \mathbb{R}^2$ e $||v|| = 1$.

Pode-se observar que a Equação de Laplace (Equação 3.7) é um caso especial da Equação 3.12, quando $\varepsilon = 0$.

Devido a influência do vetor v na orientação do campo gradiente, esta abordagem utilizando a Equação (3.12) foi denominada no trabalho de Prestes (2003) como Campos Potenciais Orientados (CPO).

Em Silva, Romero e Caurin (2008) foi mostrado que a discretização utilizada por Prestes (2003) (Equação 3.13), por consequência também no trabalho de Faria (2006), levava a um limitante artificial (isto é, inexistente no modelo matemático, porém existente na solução numérica) em ε, que induziam perturbações numéricas na relaxação, quando esta não deveria existir. Tais perturbações levavam a um comportamento indesejado do algoritmo de planejamento de movimento, que por vezes não era capaz de realizar a tarefa. Já em Silva (2011) foi demonstrado que haveria ao menos uma forma alternativa de discretizar o CPO, que não resultaria em tal limitante artificial, com isso chegando na molécula computacional da Equação 3.14.

$$p_{i,j} = \frac{1}{4}\left[(1+\lambda v_y)p_{i+1} + (1-\lambda v_y)p_{i-1} + (1+\lambda v_x)p_{j+1} + (1-\lambda v_x)p_{j-1}\right] \tag{3.13}$$

onde $\lambda = 0.5\varepsilon$, com $-2 < \varepsilon < 2$, $v = (v_x, v_y) \in \mathbb{R}^2$ e $||v|| = 1$.

$$p_{i,j} = \frac{(1+h\varepsilon|v_x|)\, p_{i,j+sgn(v_x)} + (1+h\varepsilon|v_y|)\, p_{i+sgn(v_y),j} + p_{i,j-sgn(v_x)} + p_{i-sgn(v_y),j}}{4+h\varepsilon\,(|v_y|+|v_x|)} \tag{3.14}$$

onde

$$sgn(x) = \begin{cases} +1, & x \geq 0 \\ \\ -1, & x < 0 \end{cases}$$

com $\varepsilon \in \mathbb{R}$, $v = (v_x, v_y) \in \mathbb{R}^2$ e $||v|| = 1$.

Tanto a partir da Equação 3.13 quanto da Equação 3.14, é possível aplicar os métodos de relaxação de malhas citados na Seção 3.2.

3.4 Campo Potencial Localmente Orientado

A técnica de Campo Potencial Localmente Orientado[1] (*CPLO*), foi proposta em Faria (2006) e, consiste de uma modificação do CPO (que, por sua vez, é uma modificação do próprio CPH).

A inspiração do CPLO é a capacidade de conduzir múltiplos agentes para a meta, através de diferentes direções (vetores), em um mesmo grid. Desta forma, vários robôs poderiam seguir diferentes trajetórias, usando diferentes vetores, que os conduzem até a meta. Tal feito é obtido aplicando a Equação 3.12 somente em uma região (denominada *área de influência*) em torno de cada robô. Normalmente, esta região é definida como um círculo de raio fixo com centro na posição do robô (como no exemplo da Figura 18). E, para as outras regiões, é aplicada a Equação 3.7.

O método CPLO pode ser matematicamete descrito como na Equação 3.15.

$$\nabla^2 P(x,y) + \varepsilon(x,y)\,\langle \nabla P(x,y), v(x,y)\rangle = 0 \tag{3.15}$$

onde

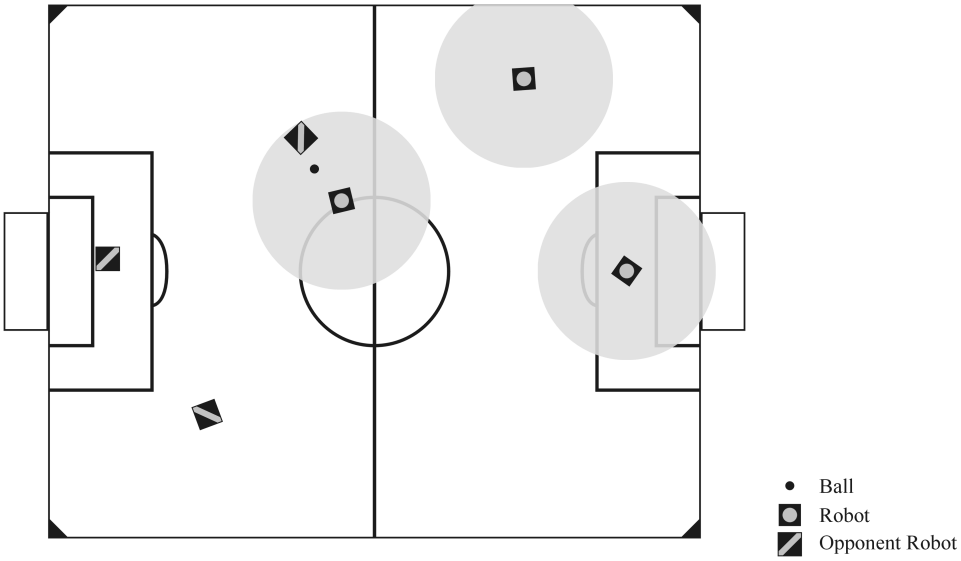
$$\varepsilon(x,y) = \begin{cases} \varepsilon_k & \text{com } \varepsilon_k \in \mathbb{R}\text{, caso } (x,y) \text{ esteja} \\ & \text{na área de influência do robô } k \\ \\ 0 & \text{caso contrário} \end{cases}$$

[1] Locally Oriented Potential Field (*LOPF*)

e,

$$v(x,y) = \begin{cases} v_k & \text{com } v_k \in \mathbb{R}^2, \text{ e } ||v_k|| = 1, \text{ caso } (x,y) \text{ esteja} \\ & \text{na área de influência do robô } k \\ \\ \overrightarrow{0} & \text{caso contrário} \end{cases}$$

Figura 18 – Uma situação que pode ocorrer dentro do ambiente de futebol de robôs. As áreas cinzas indicam as regiões de influência de cada robô



Fonte: Silva, Romero e Caurin (2008).

Pode ser visto pela Equação 3.15 que o CPLO mescla o CPO e o CPH, em uma mesma malha.

3.5 Considerações Finais

Neste capitulo foram apresentados os métodos de Campo Potencial investigados nesta tese. A discretização de cada método foi apresentada para o caso 2D, conforme consta na literatura, para que nos próximos capítulos sejam estendidas para o caso 3D.

Cabe ressaltar que o CPK tem a sua complexidade atrelada a quantidade de obstáculos, isto é, quanto menos obstáculos num ambiente, mais rápido o algoritmo irá funcionar. Já os métodos CPH, CPO e CPLO tem a sua complexidade atrelada ao tamanho do espaço que será representado no ambiente de atuação do robô, mas independe da quantidade de obstáculos presente no mesmo.

CAPÍTULO

4

MODELO PROPOSTO

Neste capítulo, na Seção 4.1, será apresentada a extensão tridimensional do Campo Potencial Orientado criada neste trabalho, e na Seção 4.2 a prova de estabilidade do método numérico Gauss-Seidel aplicado para solucioná-lo e como consequência a prova de sua convergência.

4.1 Campos Potencial Orientado Tridimensional

Apesar de o CPH, normalmente, ser utilizado em ambiente bidimensional, a Equação de Laplace é uma equação proposta para qualquer espaço euclidiano n-dimensional, como na Equação 4.1.

$$\nabla^2 P(x_1, x_2, \cdots, x_n) = \sum_{k=0}^{n} \frac{\partial^2}{\partial {x_k}^2} P(x_1, x_2, \cdots, x_n) = 0 \tag{4.1}$$

com $P : \mathbb{R}^n \mapsto \mathbb{R}$

A Equação 4.1 pode ser restrita a um espaço tridimensional, como na Equação 4.2.

$$\nabla^2 P(x,y,z) = \frac{\partial^2}{\partial x^2} P(x,y,z) + \frac{\partial^2}{\partial y^2} P(x,y,z) + \frac{\partial^2}{\partial z^2} P(x,y,z) = 0 \tag{4.2}$$

com $P : \mathbb{R}^3 \mapsto \mathbb{R}$.

A solução numérica da Equação 4.2 pode ser encontrada discretizando-a como na Equação 4.3. Para simplificar a notação, os índices serão omitidos quando eles apenas forem i, j ou k, isto é, $p_{i,j,k}$ e $p_{i,j+1,k}$ serão representados por p e p_{j+1}, respectivamente.

$$\frac{p_{j+1} - 2p + p_{j-1}}{\Delta x^2} + \frac{p_{i+1} - 2p + p_{i-1}}{\Delta y^2} + \frac{p_{k+1} - 2p + p_{k-1}}{\Delta z^2} = 0 \tag{4.3}$$

Assumindo $\Delta x = \Delta y = \Delta z$, isto é, espaçamento regular da malha de discretização, com células cúbicas, a Equação 4.3 pode ser simplificada como na Equação 4.4.

$$p = \frac{p_{j+1} + p_{j-1} + p_{i+1} + p_{i-1} + p_{k+1} + p_{k-1}}{6} \tag{4.4}$$

De forma similiar, o CPO em um espaço euclidiano n-dimensional pode ser visto como na Equação 4.5.

$$\nabla^2 P(\overrightarrow{\mathbf{w}}) + \varepsilon \left\langle \nabla P(\overrightarrow{\mathbf{w}}), \overrightarrow{\mathbf{v}} \right\rangle = 0 \tag{4.5}$$

com $P : \mathbb{R}^n \mapsto \mathbb{R}$, $\varepsilon \in \mathbb{R}$, $\overrightarrow{\mathbf{v}}, \overrightarrow{\mathbf{w}} \in \mathbb{R}^n$ e $||\overrightarrow{\mathbf{v}}|| = 1$.

A Equação 4.5 pode ser restrita a um espaço tridimensional como na Equação 4.6, e sua forma expandida pode ser vista na Equação 4.7.

$$\nabla^2 P(x,y,z) + \varepsilon \left\langle \nabla P(x,y,z), \overrightarrow{\mathbf{v}} \right\rangle = 0 \tag{4.6}$$

$$\begin{gathered} \frac{\partial^2 P(x,y,z)}{\partial x^2} + \frac{\partial^2 P(x,y,z)}{\partial y^2} + \frac{\partial^2 P(x,y,z)}{\partial z^2} + \\ \varepsilon \left[\frac{\partial P(x,y,z)}{\partial x} \mathbf{v}_x + \frac{\partial P(x,y,z)}{\partial y} \mathbf{v}_y + \frac{\partial P(x,y,z)}{\partial z} \mathbf{v}_z \right] = 0 \end{gathered} \tag{4.7}$$

com $P : \mathbb{R}^3 \mapsto \mathbb{R}$, $\varepsilon \in \mathbb{R}$, $\overrightarrow{\mathbf{v}} = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z) \in \mathbb{R}^3$ e $||\overrightarrow{\mathbf{v}}|| = 1$.

Com a utilização de diferenças upwind (SILVA, 2011), a Equação 4.8 pode ser obtida.

$$p = \frac{\begin{bmatrix} (1 + h\varepsilon|\mathbf{v}_x|)p_{j+sgn(\mathbf{v}_x)} + p_{j-sgn(\mathbf{v}_x)} + \\ (1 + h\varepsilon|\mathbf{v}_y|)p_{i+sgn(\mathbf{v}_y)} + p_{i-sgn(\mathbf{v}_y)} + \\ (1 + h\varepsilon|\mathbf{v}_z|)p_{k+sgn(\mathbf{v}_z)} + p_{k-sgn(\mathbf{v}_z)} \end{bmatrix}}{6 + h\varepsilon(|\mathbf{v}_x| + |\mathbf{v}_y| + |\mathbf{v}_z|)} \tag{4.8}$$

com

$$sgn(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{4.9}$$

Para efeito de simplificação, adota-se

- $\lambda_x = h\varepsilon|\mathbf{v}_x|$

- $\lambda_y = h\varepsilon|\mathbf{v}_y|$
- $\lambda_z = h\varepsilon|\mathbf{v}_z|$

Sem perda de generalização, adota-se $\varepsilon \geq 0$, pois caso fossem adotados um determinado $\varepsilon_{neg} < 0$ e um vetor v_{neg}, bastaria escolher um vetor $v = -v_{neg}$, em sentido contrário, e $\varepsilon = -\varepsilon_{neg}$, com $\varepsilon \geq 0$ nesse caso. Algebricamente, seriam formulações idênticas, porém, tal assunção facilita a compreensão da prova de estabilidade e convergência, a ser apresentada na Seção 4.2.

Então, a Equação 4.8 pode ser vista como na Equação 4.10.

$$p = \frac{\begin{bmatrix} (1+\lambda_x)p_{j+sgn(\mathbf{v}_x)} + p_{j-sgn(\mathbf{v}_x)} + \\ (1+\lambda_y)p_{i+sgn(\mathbf{v}_y)} + p_{i-sgn(\mathbf{v}_y)} + \\ (1+\lambda_z)p_{k+sgn(\mathbf{v}_z)} + p_{k-sgn(\mathbf{v}_z)} \end{bmatrix}}{6+\lambda_x+\lambda_y+\lambda_z} \tag{4.10}$$

4.2 Prova de convergência e estabilidade

Nesta seção serão avaliadas a estabilidade dos métodos de Jacobi-Richardson e Gauss-Seidel (BURDEN; FAIRES; REYNOLDS, 1978; FRANCO, 2008; ORTEGA, 1972) aplicados ao sistema linear resultante do método de campo potencial orientado tridimensional, discretizado através de diferenças upwind. Definições e teoremas utilizados nesta prova foram reproduzidos na íntegra no Apêndice B.

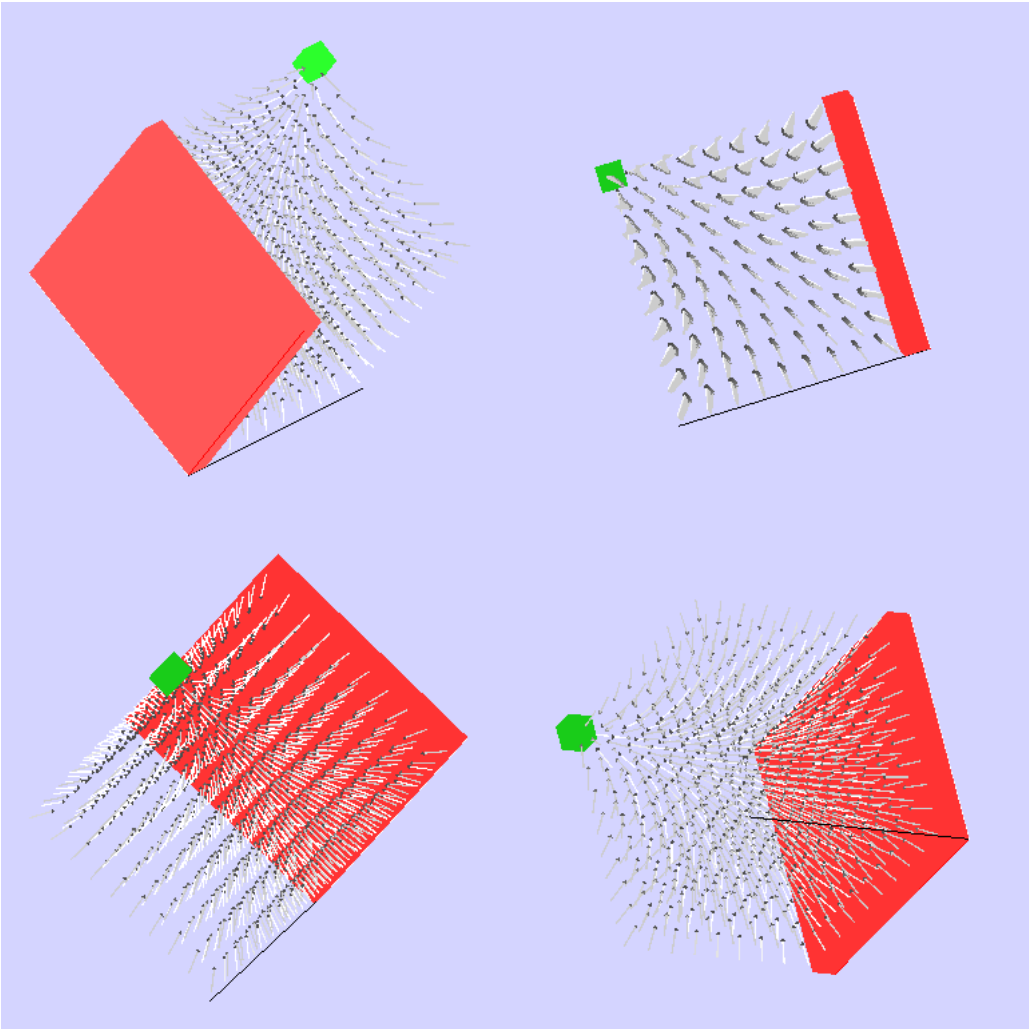
Suponha que $P : \Omega \subseteq \mathbb{R}^3 \to \mathbb{R}$ seja como na Equação 4.7 e que Ω foi discretizado com a propriedade de ser conectado por caminhos.

Para efeito demonstrativo, sem perda de generalização, será usado o exemplo da Figura 19 para ilustrar algumas passagens. Na Figura 19, Ω é um espaço tridimensional discretizado em células de espaçamento idêntico $h = 1$, com 10 células em cada dimensão, o vetor diretor é $v = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ e $\varepsilon = 1$.

Na Figura 20, é apresentado o resultado do CPO3D para o exemplo da Figura 19, com o caminho a partir da célula $(3,4,8)$.

Deve ser notado que apenas um dos lados está marcado como obstáculo, na cor vermelha. Tal escolha se deu por motivo de visualização pois, no exemplo, os seis lados serão obstáculos, para evitar que o robô saia da área delimitada. Com isto, na versão atual da interface gráfica desenvolvida, teria efeito de oclusão completa do espaço Ω. Em verde nota-se a meta e, nas demais células, células livres, vetores indicando que caminho adotar a partir daquela célula.

Figura 19 – Exemplo de situação aplicável ao CPO3D



Fonte: Elaborada pelo autor.

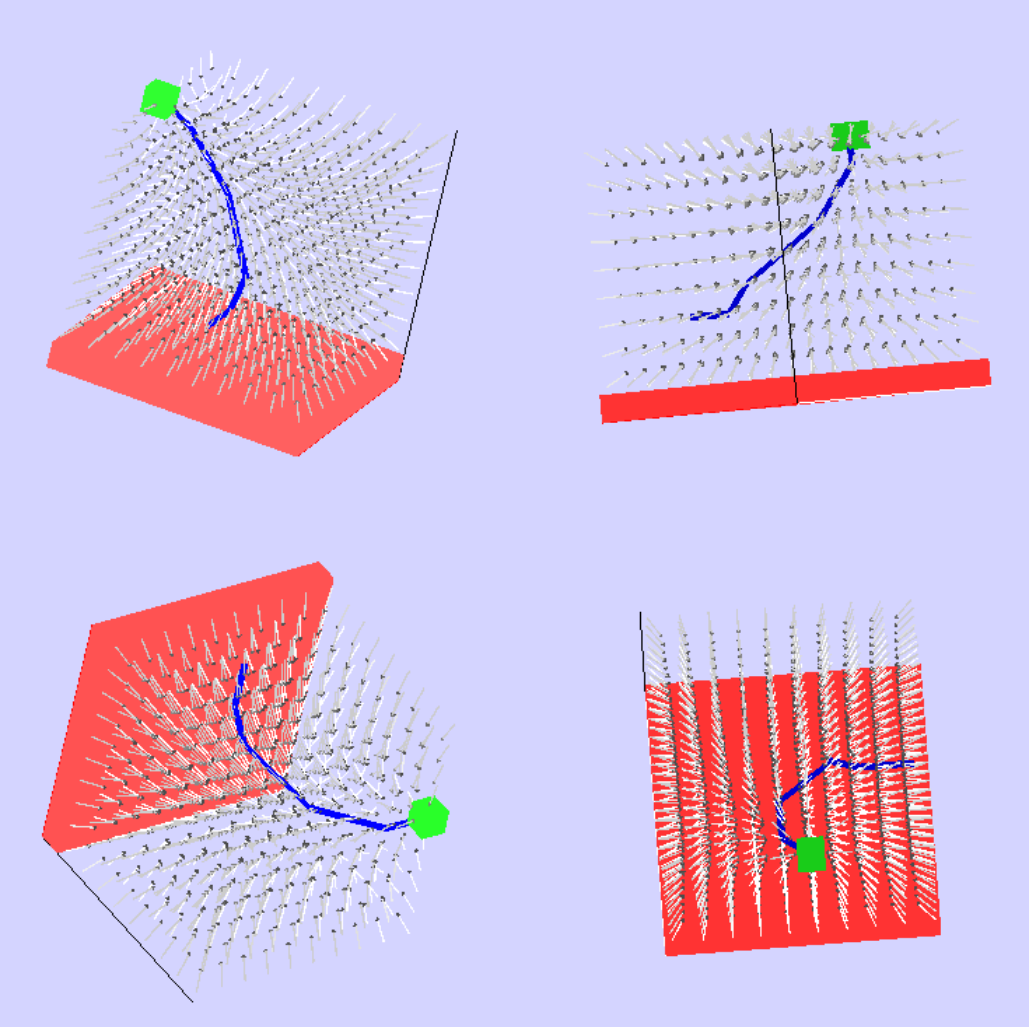
Para cada célula livre do exemplo da Figura 19 é aplicada a Equação 4.10 e, então, obtém-se um conjunto de equações, uma para cada célula livre, tendo suas primeiras equações apresentadas nas Equações 4.12 e 4.13, para as células $(1,1,1)$ e $(1,1,2)$.

$$p_{1,1,1} = \frac{\begin{bmatrix} (1+\lambda_x)p_{2,1,1} + p_{0,1,1} + \\ (1+\lambda_y)p_{1,2,1} + p_{1,0,1} + \\ (1+\lambda_z)p_{1,1,2} + p_{1,1,0} \end{bmatrix}}{6+\lambda_x+\lambda_y+\lambda_z} \tag{4.11}$$

A partir da Equação 4.11, é possível rearranjar as variáveis para o lado esquerdo da equação, a fim de obter a Equação 4.12.

$$p_{1,1,1} - \frac{[(1+\lambda_x)p_{2,1,1} + (1+\lambda_y)p_{1,2,1} + (1+\lambda_z)p_{1,1,2}]}{6+\lambda_x+\lambda_y+\lambda_z} = \frac{p_{0,1,1} + p_{1,0,1} + p_{1,1,0}}{6+\lambda_x+\lambda_y+\lambda_z} \tag{4.12}$$

Figura 20 – Caminho planejado a partir da situação da Figura 19



Fonte: Elaborada pelo autor.

Por consequência, o mesmo pode ser feito para a célula $(1,1,2)$, como na Equação 4.13.

$$p_{1,1,2} - \frac{[(1+\lambda_x)p_{2,1,2} + (1+\lambda_y)p_{1,2,2} + (1+\lambda_z)p_{1,1,3} + p_{1,1,1}]}{6+\lambda_x+\lambda_y+\lambda_z} = \frac{p_{0,1,2} + p_{1,0,2}}{6+\lambda_x+\lambda_y+\lambda_z} \tag{4.13}$$

Também deve ser observado que as células $p_{i,j,k}$ do lado direito da igualdade fazem parte da condição de contorno, ou seja, seus valores são conhecidos, enquanto que no lado esquerdo, as células $p_{i,j,k}$ são células livres, ou seja, variáveis. Assim, o valor das células pode ser atualizado através dos métodos de Jacobi-Richardson ou Gauss-Seidel, ao se adotar uma particular orientação das células. Para facilitar a notação, podemos considerar as variáveis como

na Equação 4.14

$$\begin{array}{cccc} x_1 = p_{1,1,1} & x_2 = p_{1,1,2} & \dots & x_8 = p_{1,1,8} \\ x_9 = p_{1,2,1} & x_{10} = p_{1,2,2} & \dots & x_{16} = p_{1,2,8} \\ x_{17} = p_{1,3,1} & x_{18} = p_{1,3,2} & \dots & x_{24} = p_{1,3,8} \\ & \vdots & & \\ & x_{512} = p_{8,8,8} & & \end{array} \tag{4.14}$$

Como $p_{0,1,1}$ não é uma variável, isto é, faz parte da condição de contorno, ao colocar na forma padrão $A \cdot x = b$ as equações resultantes da aplicação da Equação 4.10 em todas as células livres, utilizando a notação da Equação 4.14, obtém-se a Equação 4.15. Considerando que existem n_{free} células livres, a matriz A é uma matriz quadrada de $n_{free} \times n_{free}$ elementos, bem como os vetores-coluna x e b possuem n_{free} elementos.

Para facilitar a notação e visualização da matriz A, adota-se

- $\xi_x = \dfrac{(1+\lambda_x)}{6+\lambda_x+\lambda_y+\lambda_z}$
- $\xi_y = \dfrac{(1+\lambda_y)}{6+\lambda_x+\lambda_y+\lambda_z}$
- $\xi_z = \dfrac{(1+\lambda_z)}{6+\lambda_x+\lambda_y+\lambda_z}$
- $\xi_1 = \dfrac{1}{6+\lambda_x+\lambda_y+\lambda_z}$

$$\begin{bmatrix} 1 & -\xi_z & 0 & \dots & 0 & -\xi_y & 0 & \dots \\ -\xi_1 & 1 & -\xi_z & \dots & 0 & 0 & -\xi_y & \dots \\ 0 & -\xi_1 & 1 & \dots & 0 & 0 & 0 & \dots \\ & & & \vdots & & & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \xi_1 \left(p_{0,1,1} + p_{1,0,1} + p_{1,1,0}\right) \\ \xi_1 \left(p_{0,1,2} + p_{1,0,2}\right) \\ \xi_1 \left(p_{0,1,3} + p_{1,0,3}\right) \\ \vdots \end{bmatrix} \tag{4.15}$$

Uma importante propriedade da matriz A da Equação 4.15 é ser diagonalmente dominante (Definição 2) e isto pode ser analisado através de dois casos:

- Se uma determinada célula $p_{i,j,k} = x_q$ não tem vizinhança contida no contorno, isto quer dizer que na linha q, o vetor $b_q = 0$ e a soma dos elementos da linha q da matriz A é como na Equação 4.16

$$
\begin{aligned}
\sum_{\substack{r=0\\r\neq q}}^{n} |a_{qr}| &= (3|\xi_1| + |\xi_x| + |\xi_y| + |\xi_z|) \\
&= \frac{3 + |(1+\lambda_x)| + |(1+\lambda_y)| + |(1+\lambda_z)|}{|6+\lambda_x+\lambda_y+\lambda_z|} \\
&= \frac{3 + |(1+h\varepsilon|\mathbf{v}_x|)| + |(1+h\varepsilon|\mathbf{v}_y|)| + |(1+h\varepsilon|\mathbf{v}_z|)|}{|6+h\varepsilon|\mathbf{v}_x| + h\varepsilon|\mathbf{v}_y| + h\varepsilon|\mathbf{v}_z||} \\
&= \frac{3 + 1 + h\varepsilon|\mathbf{v}_x| + 1 + h\varepsilon|\mathbf{v}_y| + 1 + h\varepsilon|\mathbf{v}_z|}{6 + h\varepsilon|\mathbf{v}_x| + h\varepsilon|\mathbf{v}_y| + h\varepsilon|\mathbf{v}_z|} \\
&= \frac{6 + h\varepsilon|\mathbf{v}_x| + h\varepsilon|\mathbf{v}_y| + h\varepsilon|\mathbf{v}_z|}{6 + h\varepsilon|\mathbf{v}_x| + h\varepsilon|\mathbf{v}_y| + h\varepsilon|\mathbf{v}_z|} \\
&= 1
\end{aligned}
\tag{4.16}
$$

Isto é, $\sum_{\substack{r=0\\r\neq q}}^{n} |a_{qr}| = |a_{qq}| = 1$.

- Se uma determinada célula $p_{i,j,k} = x_q$ tem ao menos um elemento de sua vizinhança contida no contorno, isto quer dizer que na linha q, o vetor $b_q \neq 0$ e a soma dos elementos da linha q da matriz A é estritamente menor do que $|a_{qq}|$, como na Equação 4.17.

$$
\sum_{\substack{r=0\\r\neq q}}^{n} |a_{qr}| < (3|\xi_1| + |\xi_x| + |\xi_y| + |\xi_z|) = 1 = |a_{qq}| \tag{4.17}
$$

A matriz A também é irredutível, pois por definição a discretização de Ω preserva a propriedade do domínio ser conexo por caminhos e, portanto, o grafo associado a matriz A é fortemente conectado. Isto é, ser conexo por caminhos implica em ser fortemente conectado, conforme Definição 1.

Ainda, como pode ser visto na Equação 4.17, células livres que possuem vizinhança no contorno tornam a inequação da Definição 2 estrita, portanto, a matriz A é dita ser diagonalmente dominante e irredutível e, por fim, pelo Teorema 2, conclui-se que ao aplicar o método das diferenças finitas (FRANCO, 2008; SILVA; TOMAZELA; ROMERO, 2016) para discretizar a Equação 4.7, o método de Gauss-Seidel (e por consequência o SOR) e o método de Jacobi-Richardson convergirão se aplicados ao sistema linear da Equação 4.15.

4.3 Considerações Finais

Neste Capítulo foi apresentado uma discretização do método campo potencial orientado para o caso 3D. A prova de convergência e estabilidade do método também foi apresentada.

Considerando o caso especial de $\varepsilon = 0$, ou de $v = 0$, obtém-se o método de campos potenciais harmônicos em 3 dimensões e, por consequência, como na prova de convergência e estabilidade não há exigência alguma de que o valor de ε ou de v sejam não nulos, a mesma também é válida para o CPH em 3 dimensões.

Além disso, a prova de convergência continua válida para outro método de campos potenciais: o CPLO. Isso decorre do fato de o CPLO ser um método que mescla CPH em algumas regiões do campo potencial e CPO em outras.

Por fim, cabe ressaltar que tal prova de convergência pode ser estendida para o caso n-dimensional, considerando que as mesmas hipóteses foram satisfeitas:

- Ω ser conexo por caminhos, mesmo depois de discretizado
- A discretização do CPO ocorrer por diferenças finitas com esquema upwind

Então, mesmo para espaços de maior dimensionalidade o CPO continua sendo estável e convergindo, ainda que isso possa vir a ter um custo computacional elevado.

CAPÍTULO

5

BIBLIOTECA DE PLANEJAMENTO DE MOVIMENTO

Uma das propostas iniciais do presente trabalho era a criação de um arcabouço de métodos de planejamento de movimento baseados em campo potencial. Tal arcabouço se traduziria em uma biblioteca modular de técnicas de planejamento de movimento que pudesse ser expandida ao longo do tempo, com novas técnicas que viessem a surgir, bem como permitir fácil adaptação dos métodos já existentes a casos reais.

Tal biblioteca encontra-se implementada em sua versão sequencial e parcialmente em sua versão paralela e está disponível para download em <https://github.com/LAR-USP/MoPlaLib> (SILVA, 2014), licenciado sob a licença de código livre GPL (Free Software Foundation, Inc., 2007) e, provisoriamente, está sendo chamada de *MOPLALIB*, acrônimo de *Motion Planing Library*.

A MOPLALIB foi desenvolvida em C++, com alguns trechos críticos implementados diretamente em C, utilizando o framework Qt (The Qt Company Ltd., 2018). Tal framework foi escolhido devido a sua abrangência de recursos que incluem, mas não se restringem a: threads, opengl e interface gráfica (*widgets*), comunicação com bases de dados.

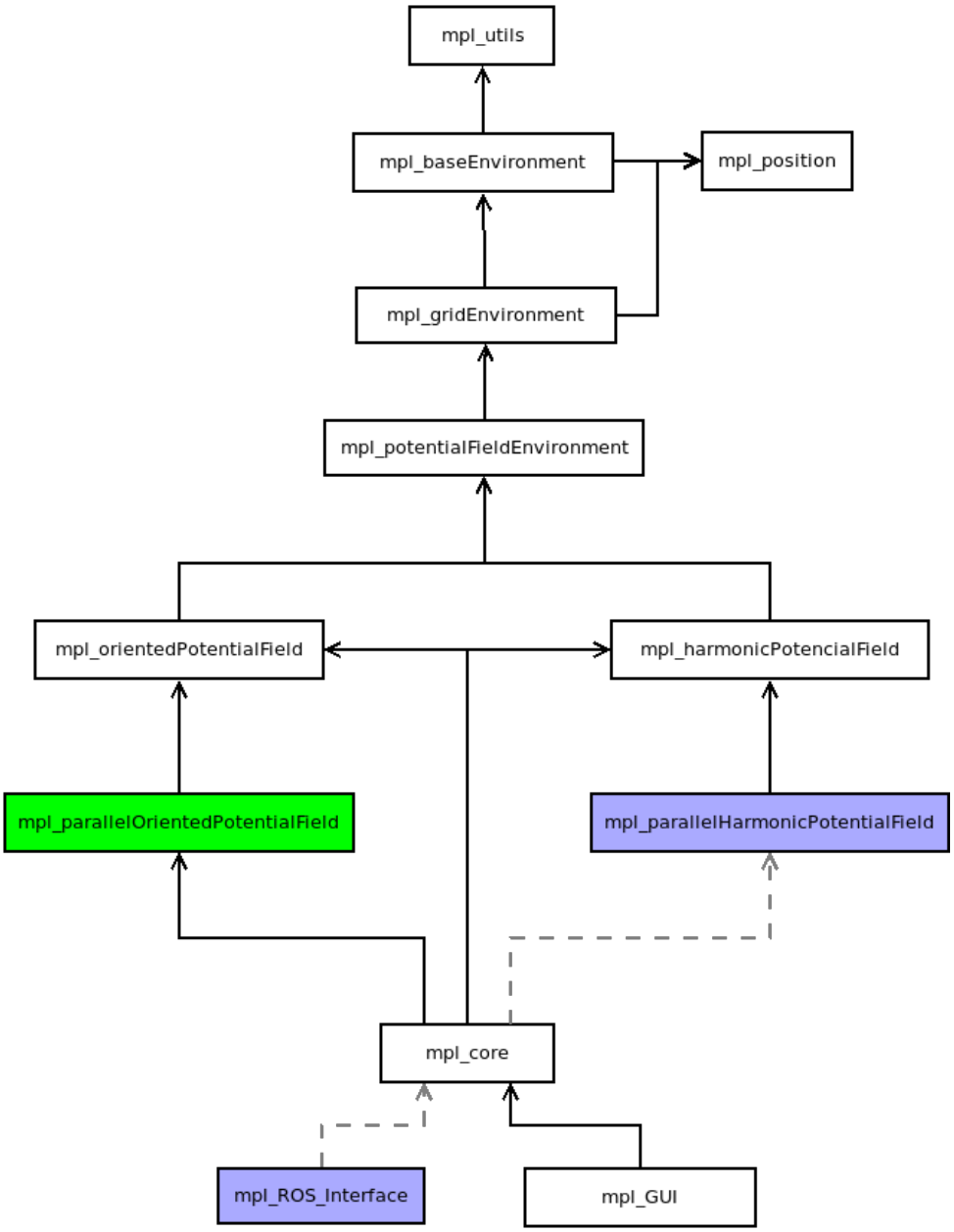
O diagrama da Figura 21 ilustra o estágio atual de desenvolvimento da mesma. Os módulos de fundo branco são os módulos já consolidados em versões estáveis.

O módulo **mpl_Core**, no presente momento, se comunica apenas com o módulo GUI **mpl_GUI**. Porém, uma módulo que faça interface com o ROS[1] (QUIGLEY *et al.*, 2009) não foi descartado. Durante a execução deste projeto, o desenvolvimento de uma interface em modo texto foi realizado apenas para fins de testes.

A utilidade do módulo **mpl_Core** se justifica ao servir de **middleware**, camada intermediária de software, entre a biblioteca de funções e alguma aplicação do cliente que venha a

[1] Meta sistema operacional para controle de robôs; do inglês *Robot Operational System*

Figura 21 – Diagrama da MOPLALIB



Fonte: Elaborada pelo autor.

utilizar tal, portanto a mesma encapsula todas as funções de mais baixo nível, como, por exemplo, relaxamento de malhas para solução de equações diferenciais parciais elípticas, fornecendo funções de mais alto nível

O módulo **mpl_parallelOrientedPotentialField** encontra-se em desenvolvimento. Diversas abordagens estão sendo experimentadas para se ter, dentro da MOPLALIB, uma versão paralela dos algoritmos de relaxamento de malhas dos métodos de planejamento de movimento. O módulo **mpl_parallelHarmonicPotentialField** decorre trivialmente do módulo **mpl_parallelOrientedPotentialField**, assim que este estiver em sua versão estável.

5.1 Módulos

Os demais módulos podem ser utilizadas diretamente, já que não estão acoplados ao **middleware** do módulo **mpl_Core**. Isto é, caso exista algum motivo para um eventual utilizador da MOPLALIB decidir acessar diretamente os outros módulos, o mesmo não encontrará dificuldades.

Cada um deles é brevemente descrito a seguir:

- **mpl_utils**
  É o módulo em que se pode encontrar funcionalidades genéricas utilizadas em outros módulos, mas que não fazem sentido serem reimplementadas em todas elas. Macros, definições gerais que podem vir a mudar o comportamento do código (por exemplo, como são geradas as mensagens de erro por alguma operação ilegal ou inválida) se encontram neste módulo.

- **mpl_position**
  Este módulo é responsável por abstrair o conceito de posição num ambiente tridimensional. Internamente possui três variáveis para as coordenadas, bem como diversos métodos que facilitam o desenvolvimento de outras classes que dependem dela, como cálculo de distâncias entre duas posições.

- **mpl_baseEnvironment**
  Este módulo é responsável por armazenar e gerenciar listas de posições de obstáculos e metas, bem como da posição atual do robô a ser controlado. Optou-se por um módulo específico para essa função pois a partir deste módulo, outros módulos que, por exemplo, lidem com técnicas de planejamento de movimento diferentes das já implementadas, podem ser desenvolvidos.

- **mpl_gridEnvironment**
  Este módulo encapsula o módulo **mpl_baseEnvironment**, sendo também responsável por, além de armazenar e gerenciar listas de posições como no módulo **mpl_baseEnvironment**, representar as posições de obstáculos e metas, bem como a posição do robô atual, numa grade de ocupação tridimensional de uma representação discretizada do ambiente. A exemplo do módulo **mpl_baseEnvironment**, este módulo também permite que eventual desenvolvedor o utilize para gerenciar grades de ocupação, bem como manter listas de obstáculos e metas.

- **mpl_potentialFieldEnvironment**
  Este módulo, por sua vez, encapsula o módulo **mpl_gridEnvironment**, sendo responsável pelo gerenciamento de listas de obstáculos e metas, suas representações em uma grade de ocupação, contudo também é responsável pelo armazenamento de uma grade de potenciais numa representação discretizada do ambiente. A partir deste módulo, técnicas de

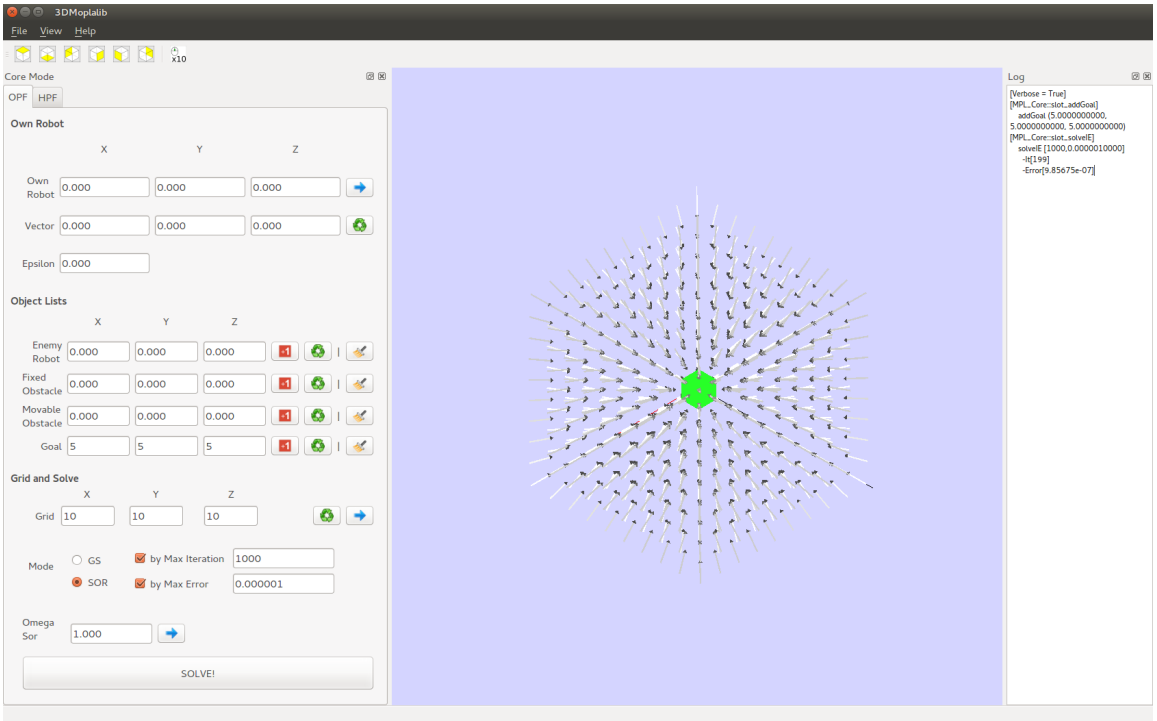
planejamento de movimento que dependam de gerenciamento de listas de obstáculos e metas, bem como de representação discreta do ambiente, podem vir a ser implementadas

- **mpl_harmonicPotentialField**
  Este módulo implementa a técnica de Campo Potencial Harmônico em três dimensões, utilizando o ferramental disponibilizado pelo módulo **mpl_potentialFieldEnvironment**. Ressalta-se que é possível utilizar tal módulo para realizar planejamento de movimento em duas dimensões, bastando apenas limitar uma das dimensões ao valor 1.

- **mpl_orientedPotentialField**
  Assim como o módulo **mpl_harmonicPotentialField**, este módulo implementa a técnica de Campo Potencial Orientado em três dimensões.

- **mpl_GUI** Módulo de interface gráfica desenvolvida em Qt. Principais componentes gráficos podem ser vistos na Figura 22 e 23, como menu para inserção manual de objetos, visualizador do campo potencial tridimensional (codificado em OpenGL) e log de execução do sistema.

Figura 22 – Screenshot da **mpl_GUI**

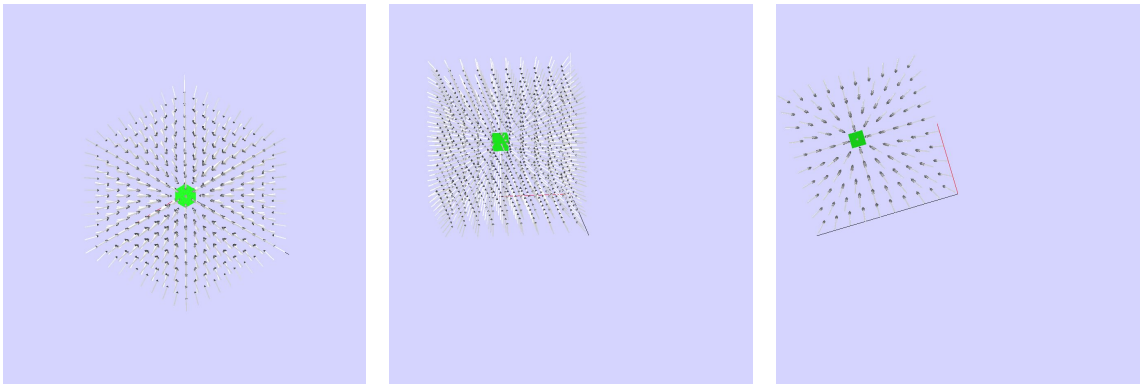


Fonte: Elaborada pelo autor.

5.2 Paralelismo

Com o objetivo de aumentar o desempenho do CPO3D, diminuindo o tempo de processamento, *threads* estão sendo usadas para solucionar o PVC da EDP do CPO3D.

Figura 23 – Exemplo de Snapshots possíveis com a interface **mpl_GUI**



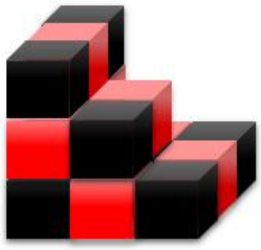
(a) Snapshot 1 (b) Snapshot 2 (c) Snapshot 3

Fonte: Elaborada pelo autor.

O método SOR tradicional não especifica uma sequência específica de atualização de células, apenas que elas devem serem atualizadas sincronicamente. Portanto, diversas estratégias de paralelismo (ou concorrência) podem ser idealizadas. No trabalho de Adams e Ortega (1982), são apresentadas diversas estratégias de coloração de células, de tal forma que cada conjunto de células de uma determina cor é executado por um núcleo de processamento (também conhecido por *core*).

O método *Red-Black SOR* é considerado uma das maneiras mais práticas de particionar as células (ADAMS; ORTEGA, 1982; ADAMS; JORDAN, 1986; MITTAL, 2014) a serem iteradas, particionando em células vermelhas (*red*) e pretas (*black*), de tal forma que nenhuma célula vermelha possua vizinha vermelha, apenas pretas. Respectivamente, nenhuma célula preta possui vizinha preta, apenas vermelhas. Isto pode ser observado na Figura 24

Figura 24 – Ambiente discretizado utilizando o conceito de células vermelhas e pretas alternantes



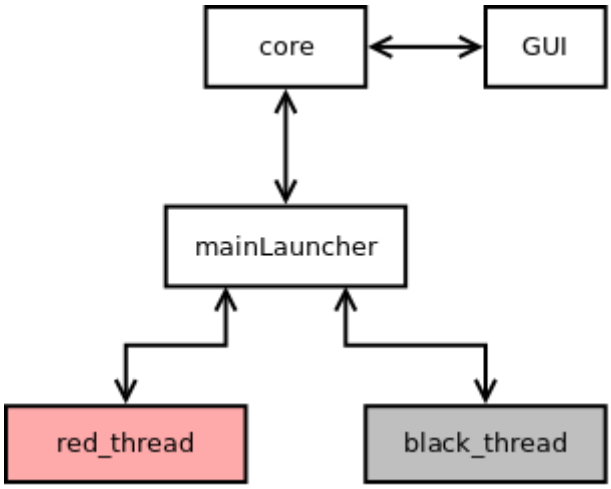
Fonte: Silva, Tomazela e Romero (2016).

A partir desta divisão, dois processos concorrentes, i.e. *threads*, são criados (*red_-thread* e *black_thread*), que serão responsáveis pela relaxação das células vermelhas e pretas, respectivamente. O sistema operacional (Ubuntu 18.04.1 LTS, kernel 4.15.0-39-generic) e o framework Qt (Qt 5.11.2) que encapsula as threads do sistema operacional de forma transparente ao desenvolvedor, não garante execução simultânea das threads.

Ainda, uma terceira *thread*, *mainLauncher*, foi criada para calcular o resíduo, isto é, a convergência, entre o tempo t e $t+1$, bem como coordenar a ação das *threads red_thread* e *black_thread* . A relaxação pode ser finalizada tanto por um número fixo de iterações, quanto por um limiar de resíduo, ou ambos simultaneamente.

Como pode ser visto anteriormente neste Capítulo, a biblioteca desenvolvida comporta interface gráfica e núcleo de execução independente. Com isto, mais duas *threads* são utilizadas no fluxo da solução paralela, sem afetar de forma significativa o desempenho numa máquina com múltiplos cores, porém, trazendo grande vantagem para a organização e usabilidade da biblioteca em si. Tal organização pode ser vista na Figura 25. A *thread mainLauncher* é responsável por coordenar as *threads red_thread* e *black_thread*, e se comunicar com a *thread core*, que por sua vez troca informações com a *thread GUI* (de *Graphics User Interface*).

Figura 25 – Esquema lógico das instâncias paralelas, frisando o mecanismo de comunicação entre elas



Fonte: Silva, Tomazela e Romero (2016).

5.3 Considerações Finais

Neste capítulo foi apresentada a biblioteca de planejamento de movimentos, MOPLALIB, desenvolvida neste trabalho, bem como considerações acerca de sua implementação paralela.

Para facilitar a compreensão dos principais módulos da MOPLALIB, algumas classes foram omitidas neste Capítulo, como por exemplo a classe *mainLauncher*, portanto não aparecendo no diagrama da Figura 21.

Três possibilidades de critério de convergência e parada estão implementados, tanto para o CPH quanto para o CPO:

- **Por quantidade de iteração** Uma quantidade exata de iterações do relaxador do SOR ou Gauss-Seidel é usada como critério de parada, independentemente da convergência do mesmo

- **Por erro máximo** A distância (norma $L1$) entre duas grades de potencial consecutivas é calculada e verificada se é menor que um valor pré-estabelecido. Se for, o método encerra a execução.

- **Por erro máximo e quantidade de iteração** Este critério considera os dois anteriores, simultaneamente. O primeiro que for acionado, é considerado como critério de parada.

CAPÍTULO

6

ANÁLISE DE DESEMPENHO

A classe **mpl_harmonicPotentialField** implementa a técnica de campo potencial harmônico, como dito no Capítulo 5. Para a solução numérica do problema de valor de contorno da equação diferencial parcial elíptica do método de campo potencial harmônico, são usados o SOR e Gauss-Seidel (FRANCO, 2008).

Experimentos com a versão sequencial foram executadas no computador descrito na Tabela 1, e podem ser vistos nas Tabelas 2 e 3. Em ambas sequências de experimentos, foram utilizadas os dois critérios, por distância ($< 10^{-8}$) e iteração (até 1.000 iterações).

Tabela 1 – Especificações do Computador

CPU	Xeon E3-1240 (Intel)
Instrução	x86-64
Quantidade de Núcleos	4 (8 em HyperTheading)
Frequência dos Núcleos	3.40GHz
Tipo de Memória RAM	DDR 3
Frequência da Memória RAM	1333MHz
Tamanho da Memória RAM	16GB
Máxima Largura de Banda	21GB/s
Tamanho do Cache L1	256KB
Tamanho do Cache L2	1MB
Tamanho do Cache L3	8MB
Sistema Operacional	Ubuntu 14.04 (Trusty Tahr)
Kernel	3.13.0-52

Na Tabela 2 são apresentadas execuções com o método SOR, utilizando $\omega = 1.899$. Para os três primeiros casos a convergência por distância de grids foi o critério de parada, enquanto que no quarto caso o critério utilizado foi o limite de iterações, atingindo uma distância de $2,98 * 10^{-5}$ entre grids consecutivos, sendo que levou mais de 7 minutos para tanto.

Na Tabela 3 são apresentadas execuções com o método de Gauss-Seidel. Para os três

Tabela 2 – Experimentos com o CPH 3D usando SOR com $\omega = 1.899$

Dimensões (XYZ)	Total de Células	Tempo de execução em milissegundos	Total de iterações	Distância final entre duas grades consecutivas
10 10 3	300	$31,77$	224	$9,84 * 10^{-9}$
10 10 10	1.000	$98,75$	234	$9,40 * 10^{-9}$
100 100 10	100.000	$12.592,25$	314	$9,39 * 10^{-9}$
100 100 100	1.000.000	$433.804,04$	1.000	$2,98 * 10^{-5}$

primeiros casos a convergência por distância de grades foi o critério de parada, enquanto que no quarto caso o critério utilizado foi o limite de iterações, atingindo uma distância de $2,20 * 10^{2}$ entre grades consecutivas, sendo que levou mais de 7 minutos para tanto.

Tabela 3 – Experimentos com o CPH 3D usando Gauss-Seidel

Dimensões (XYZ)	Total de Células	Tempo de execução em milissegundos	Total de iterações	Distância final entre duas grades consecutivas
10 10 3	300	$9,03$	77	$7,71 * 10^{-9}$
10 10 10	1.000	$101,44$	248	$9,61 * 10^{-9}$
100 100 10	100.000	$37.162,62$	933	$9,82 * 10^{-9}$
100 100 100	1.000.000	$432.583,55$	1.000	$2,20 * 10^{2}$

No caso de $10 \times 10 \times 3 = 300$ elementos, ambos os métodos foram capazes de convergirem, num tempo de execução e quantidade de iterações dentro do esperado. Já nos casos de $10 \times 10 \times 10 = 1.000$ elementos e $100 \times 100 \times 10 = 100.000$ elementos, apesar dos métodos terem convergido, o tempo de execução ou a quantidade de iterações passa a ser infactível num problema real.

E, por fim, no caso de $100 \times 100 \times 100 = 1.000.000$, nenhum dos métodos convergiu, atingindo o máximo de 1.000 iterações. Entretanto, vale ressaltar que, para uma mesma quantidade de iterações, o SOR atingiu uma distância de grades consecutivas com ordem de magnitude $7\times$ menor do que o método de Gauss Seidel, mostrando que, ao menos para os casos testados, em versão sequencial, ele se mostra o método mais promissor.

Tabela 4 – Execução do CPH3D usando Gauss-Seidel

	Sequencial		Paralelo	
Dimensões (XYZ)	Iterações	Tempo (ms)	Iterações	Tempo (ms)
10 10 2	30	2,76	27	3,09
11 11 11	256	122,46	236	88,33
50 50 50	1000	43455,7	1000	25958,9
100 100 100	1000	375611,2	1000	232071,7

Na Tabela 4, quatro cenários foram comparados com a versão atual do código em paralelo e a versão sequencial. Para grids com poucas células, os resultados são similares, já para grids com mais células, observa-se uma melhora de 40% no desempenho do algoritmo utilizando sua versão paralela.

Tabela 5 – Execução do CPO3D usando Gauss-Seidel

	Sequencial		Paralelo	
Dimensões (XYZ)	Iterações	Tempo (ms)	Iterações	Tempo (ms)
11 11 3	56	7,83	58	7,06
50 50 50	1000	45,00	1000	25,77
100 100 100	1000	457,33	1000	232,60

Já na Tabela 5, três cenários foram comparados. Pode se observar que a versão paralela do código demonstrou ser quase duas vezes mais rápida que o código sequencial, compatível com o que era esperado.

CAPÍTULO

7

CONCLUSÃO E TRABALHO FUTUROS

Apesar de planejamento de movimento não se restringir à área de robótica, tendo aplicações nas mais distintas áreas, é uma das etapas cruciais no uso de robôs inteligentes, estando o operador ou responsável pelo robô ciente desta etapa ou não, já que a mesma pode estar encapsulada em algum método de alto nível.

Neste trabalho foi proposto uma estensão da técnica de campos potenciais para espaço 3D. Foram realizadas as implementações sequencial e paralela e uma comparação foi realizada para diferentes tamanho de *grid*. A prova de convergência e estabilidade da técnica proposta foi também realizada, sendo também possível generalizar para outras dimensões.

Procurou-se fundamentar matematicamente todos os passos desta proposta, visando solidez e consistência dos resultados obtidos. Contudo, havia um objetivo de seguir a nomenclatura e simbolização mais aceita pela comunidade, para que o mesmo não destoasse e até conflitasse com a nomenclatura de trabalhos correlatos.

A técnica de campos potenciais 3D pode complementar as técnicas de planejamento de movimento probabilísticos: RRT e PRM, considerados planejadores globais, visto que um dos passos de ambos os métodos envolve o uso de um planejador de movimento local determinístico e rápido, características estas que podem ser atribuídas aos campos potenciais, uma vez que eles são executados em tempo real e garantem encontrar um caminho, se ele existir. Os método de campo potencial além de ser um planejador global, também pode atuar como um planejador local.

Como planejador global, sua versão sequencial não se mostrou promissora, devido a impossibilidade da malha convergir em menos de um segundo. Entretanto, isso demonstra ser promissor a paralelização do código com a utilização de GPGPUs , que permitem a paralelização massiva, sendo um dos possíveis trabalhos futuros.

Foi criada uma interface gráfica em OpenGL para visualização de determinada porção

do espaço 3D, aquela relacionada ao ambiente de operação do robô, exibindo campo potencial, obstáculos, meta e o caminho construído partindo-se de um ponto origem até a meta.

Ressalta-se que, também foi construída funções específicas para visualização dos vetores a partir dos 3 ângulos de eulerianos, na qual cada célula é considerada como sendo um cubo de lado unitário.

Foram gerados os seguintes artigos durante o período desta tese:

- SILVA, M. O.; TOMAZELA, L. ; Romero, R. A. F. . Extending numerical solutions of potential fields methods based on boundary value problems for 3D environments. In: 6th International Conference on Nonlinear Science and Complexity - NSC 2016, 2016, São José dos Campos. Conference Proceedings of 6th International Conference on Nonlinear Science and Complexity - NSC 2016, 2016. v. 1.
- TOMAZELA, L.; SILVA, M. O.; Romero, R. A. F. . Campos potenciais paralelos aplicados ao controle de múltiplos robôs. 24º Simpósio Internacional de Iniciação Científica da USP, 2016.
- TOMAZELA, L.; SILVA, M. O.; Romero, R. A. F. . Campos potenciais paralelos aplicados ao controle de múltiplos robôs. 23º Simpósio Internacional de Iniciação Científica da USP, 2015.
- SILVA, M. O.; Romero, Roseli A. F. ; OLIVEIRA, S. P. ; STEWART, D. . Improving the stability of algorithms for path planning based on boundary value problems. Far East Journal of Applied Mathematics, v. 69, p. 111-123, 2012.

Também foram gerados os seguintes artigos, não diretamente relacionados a esta tese:

- FRACCAROLI, E. S. ; SILVA, M. O. ; BATISTA, M. R. ; Romero, R. A. F. Controlador fuzzy para impressora 3D de baixo custo. In: III Congresso Brasileiro de Sistemas Fuzzy, 2014, João Pessoa. III Congresso Brasileiro de Sistemas Fuzzy, 2014.
- AROCA, R. V. ; AGUIAR, F. G. ; AIHARA, C. ; TONIDANDEL, F. ; Montanari, R. ; FRACCAROLI, E. S. ; SILVA, M. O.; Romero, R. A. F. . Olimpíada Brasileira de Robótica: relatos da primeira regional em São Carlos- SP. In: 5o. Workshop de Robótica Educacional - WRE 2014, 2014, São Carlos/SP. Anais do 5o Workshop de Robótica Educacional - WRE 2014, 2014. v. 1. p. 35-41.
- BATISTA, M. R. ; JORGE, L. L. R. ; Calvo, R. ; FRACCAROLI, E. S. ; DOREA, Y. ; SILVA, M. O. ; Montanari, R. ; Romero, R. A. F. . A Solution to Swarm Robot Escorting using the Probabilistic Lloyd Method. In: Simpósio Brasileiro de Automação Inteligente, 2013, Fortaleza. Anais do XI Simpósio Brasileiro de Automação Inteligente, 2013. v. 1. p. 1-6.

Foram recebidos os seguintes prêmios durante esta tese:

- Campeão na categoria IEEE Very Small Size na IEEE Latin American Robotics Competition
- Vice-Campeão na categoria RoboCup Simulation League 2D na IEEE Latin American Robotics Competition, Sociedade Brasileira de Computação.
- Vice-Campeão na categoria RoboCup Simulation League 3D na IEEE Latin American Robotics Competition, Sociedade Brasileira de Computação.

Também houve participação na construção e manutenção das impressoras 3D existentes no LAR-ICMC-USP.

No LAR encontra-se em desenvolvimento uma arquitetura de controle para drones em ambientes não estruturados. Os resultados obtidos nesta tese contribuem diretamente para a construção de um dos módulos desta arquitetura, que é o Planejador de Movimentos.

Como trabalhos futuros, destaca-se o seguinte:

- Utilização de métodos mais eficientes para discretização do CPO, visando uma redução da quantidade de células, embora mantendo as propriedades do método. Por exemplo, utilizando o Método dos Elementos Finitos
- Embarcar o código desenvolvido, utilizando arquiteturas de baixo custo, como Raspberry Pi e similares.
- Implementação de *middleware* de interface com o ROS
- Integração com drone em desenvolvimento no LAR. Para isso, será necessário ter um software de SLAM georeferenciado e combiná-lo com o método proposto nesta tese, que indicará qual caminho o drone deverá seguir para chegar ao seu destino. .

REFERÊNCIAS

ADAMS, L.; ORTEGA, J. A multi-color sor method for parallel computation. In: **ICPP**. [S.l.: s.n.], 1982. p. 53–56. Citado na página 63.

ADAMS, L. M.; JORDAN, H. F. Is sor color-blind? **SIAM Journal on Scientific and Statistical Computing**, SIAM, v. 7, n. 2, p. 490–506, 1986. Citado na página 63.

AGARWAL, P. K.; FOX, K.; SALZMAN, O. An efficient algorithm for computing high-quality paths amid polygonal obstacles. **ACM Transactions on Algorithms (TALG)**, ACM, v. 14, n. 4, p. 46, 2018. Citado na página 22.

ALONSO-MORA, J.; NAEGELI, T.; SIEGWART, R.; BEARDSLEY, P. Collision avoidance for aerial vehicles in multi-agent scenarios. **Autonomous Robots**, v. 39, n. 1, p. 101–121, Jun 2015. ISSN 1573-7527. Disponível em: <https://doi.org/10.1007/s10514-015-9429-0>. Citado na página 22.

BEKRIS, K.; SHOME, R.; KRONTIRIS, A.; DOBSON, A. Cloud automation: Precomputing roadmaps for flexible manipulation. **IEEE Robotics & Automation Magazine**, IEEE, v. 22, n. 2, p. 41–50, 2015. Citado na página 33.

BERG, J. V. D.; FERGUSON, D.; KUFFNER, J. Anytime path planning and replanning in dynamic environments. In: IEEE. **Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on**. [S.l.], 2006. p. 2366–2371. Citado na página 22.

BERG, J. van den; PATIL, S.; ALTEROVITZ, R. Motion planning under uncertainty using differential dynamic programming in belief space. In: ____. **Robotics Research : The 15th International Symposium ISRR**. Cham: Springer International Publishing, 2017. p. 473–490. ISBN 978-3-319-29363-9. Disponível em: <https://doi.org/10.1007/978-3-319-29363-9_27>. Citado na página 22.

BERGER, B.; WALDISPÜHL, J. Novel perspectives on protein structure prediction. In: HEATH, L. S.; RAMAKRISHNAN, N. (Ed.). **Problem Solving Handbook in Computational Biology and Bioinformatics**. [S.l.]: Springer US, 2011. p. 179–207. ISBN 978-0-387-09760-2. Citado na página 22.

Burchan Bayazit, O.; ; Amato, N. M. Roadmap-based flocking for complex environments. In: **10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.** [S.l.: s.n.], 2002. p. 104–113. Citado na página 34.

BURDEN, R. L.; FAIRES, J. D.; REYNOLDS, A. C. **Numerical Analysis**. [S.l.]: Prindle, Weber and Schimidt, 1978. Citado nas páginas 53 e 81.

CANNY, J.; REIF, J. New lower bound techniques for robot motion planning problems. In: **Foundations of Computer Science, 1987., 28th Annual Symposium on**. [S.l.: s.n.], 1987. p. 49 –60. ISSN 0272-5428. Citado na página 22.

CERQUEIRA, A. C.; LINS, F. C.; MEDEIROS, A. A.; ALSINA, P. J. A. A versão 2006 da equipe poti de futebol de robôs (the 2006 version of the poti team of robot soccer). In: **II JRI - Jornada de Robótica Inteligente**. [S.l.: s.n.], 2006. Citado nas páginas 25 e 26.

CHERUBINI, A.; GIANNONE, F.; IOCCHI, L.; LOMBARDO, M.; ORIOLO, G. Policy gradient learning for a humanoid soccer robot. **Robotics and Autonomous Systems**, v. 57, p. 808–818, 07 2009. Citado nas páginas 25 e 27.

CONNOLLY, C. I.; BURNS, J. B.; WEISS, R. Path planning using laplaces equation. In: **IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 1990. Citado nas páginas 39 e 46.

CONNOLLY, C. I.; GRUPEN, R. A. On the application of harmonic functions to robotics. **Journal of Robotic Systems**, v. 10, p. 931–946, 1993. Citado na página 47.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. [S.l.]: McGraw-Hill, 2001. ISBN 0262032937. Citado na página 87.

DECASTRO, J. A.; ALONSO-MORA, J.; RAMAN, V.; RUS, D.; KRESS-GAZIT, H. Collision-free reactive mission and motion planning for multi-robot systems. In: ____. **Robotics Research: Volume 1**. Cham: Springer International Publishing, 2018. p. 459–476. ISBN 978-3-319-51532-8. Disponível em: <https://doi.org/10.1007/978-3-319-51532-8_28>. Citado na página 22.

EKENNA, C.; UWACU, D.; THOMAS, S.; AMATO, N. M. Studying learning techniques in different phases of prm construction. In: **Machine Learning in Planning and Control of Robot Motion Workshop (IROSMLPC), Hamburg, Germany**. [S.l.: s.n.], 2015. Citado na página 33.

FARIA, G. **Uma Arquitetura de controle inteligente para múltiplos robôs**. Tese (Doutorado) — ICMC-USP São Carlos, 2006. Citado nas páginas 39, 40, 41, 42, 43, 44, 45, 47 e 48.

FIGUEIREDO, D. G. **Análise de Fourier e Equações Diferenciais Parciais**. [S.l.]: Projeto Euclides / IMPA, 2005. ISBN 8524401206. Citado nas páginas 46 e 47.

FRACCAROLI, E. S. **Alocação de tarefas para a coordenação de robôs heterogêneos aplicados a agricultura de precisão**. Tese (Doutorado) — ICMC-USP São Carlos, 2018. Citado nas páginas 22 e 33.

FRANCO, N. B. **Cálculo numérico**. [S.l.]: Pearson, 2008. ISBN 9788576050872. Citado nas páginas 46, 53, 57, 67, 81 e 84.

Free Software Foundation, Inc. **GNU General Public License**. 2007. Version 3, 29 June 2007. Disponível em: <https://www.gnu.org/licenses/gpl.txt>. Citado na página 59.

GERAERTS, R.; OVERMARS, M. H. A comparative study of probabilistic roadmap planners. In: **IN: WORKSHOP ON THE ALGORITHMIC FOUNDATIONS OF ROBOTICS**. [S.l.: s.n.], 2002. p. 43–57. Citado na página 34.

GILBARG, D.; TRUDINGER, N. S. **Elliptic Partial Differential Equations of Second Order**. 2nd. ed. [S.l.]: Springer-Verlag, 1983. Citado na página 46.

GOERZEN, C.; KONG, Z.; METTLER, B. A survey of motion planning algorithms from the perspective of autonomous uav guidance. **Journal of Intelligent Robotics Systems**, Kluwer Academic Publishers, Hingham, MA, USA, v. 57, n. 1-4, p. 65–100, jan. 2010. ISSN 0921-0296. Disponível em: <http://dx.doi.org/10.1007/s10846-009-9383-1>. Citado nas páginas 22 e 36.

GUO, H.; LI, M. Y.; SHUAI, Z. Global stability of the endemic equilibrium of multigroup sir epidemic models. **Canadian Applied Mathematics Quarterly**, v. 14, n. 3, p. 259–284, 2006. Citado na página 88.

HATCHER, A. **Algebric Topology**. Cambridge, U.K.: Cambridge University Press, 2002. Disponível em <http://www.math.cornell.edu/~hatcher/AT/ATpage.html>. ISBN 0-521-79540-0. Citado na página 31.

JANSON, L.; SCHMERLING, E.; PAVONE, M. Monte carlo motion planning for robot trajectory optimization under uncertainty. In: ____. **Robotics Research: Volume 2**. Cham: Springer International Publishing, 2018. p. 343–361. ISBN 978-3-319-60916-4. Disponível em: <https://doi.org/10.1007/978-3-319-60916-4_20>. Citado na página 22.

KAVRAKI, L.; SVESTKA, P.; LATOMBE, J.-C.; OVERMARS, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. **Robotics and Automation, IEEE Transactions on**, v. 12, n. 4, p. 566 –580, aug 1996. ISSN 1042-296X. Citado nas páginas 33, 34 e 37.

KELLY, J. W. The environmental dependency of protein folding best explains prion and amyloid diseases. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 95, n. 3, p. 930–932, 1998. Citado na página 22.

KHATIB, O. **Command Dynamic dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Osbtacles**. Tese (Doutorado) — Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, 1980. Em Francês. Citado na página 39.

KIM, H.; MOKDAD, L.; BEN-OTHMAN, J. Designing uav surveillance frameworks for smart city and extensive ocean with differential perspectives. **IEEE Communications Magazine**, p. 99, 2018. Citado na página 22.

KIMCHI, G.; BUCHMUELLER, D.; GREEN, S. A.; BECKMAN, B. C.; ISAACS, S.; NAVOT, A.; HENSEL, F.; BAR-ZEEV, A.; RAULT, S. S. J.-M. **Unmanned aerial vehicle delivery system**. [S.l.]: Google Patents, 2017. US Patent 9,573,684. Citado na página 23.

KUFFNER J.J., J.; LAVALLE, S. Rrt-connect: An efficient approach to single-query path planning. In: **Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on**. [S.l.: s.n.], 2000. v. 2, p. 995 –1001 vol.2. ISSN 1050-4729. Citado nas páginas 35 e 36.

KUWATA, Y.; FIORE, G.; TEO, J.; FRAZZOLI, E.; HOW, J. Motion planning for urban driving using rrt. In: **Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on**. [S.l.: s.n.], 2008. p. 1681 –1686. Citado na página 36.

LATOMBE, J.-C. **Robot Motion Planning**. [S.l.]: Kluwer, 1991. 651 p. ISBN 0792391292. Citado nas páginas 26, 28, 29 e 32.

LAVALLE, S. M. Rapidly-exploring random trees: A new tool for path planning. Citeseer, 1998. Citado nas páginas 35 e 37.

____. **Planning Algorithms**. Cambridge, U.K.: Cambridge University Press, 2006. Disponível em <http://planning.cs.uiuc.edu/>. ISBN 978-0-521-86205-9. Citado na página 29.

LEE, K. **Development of unmanned aerial vehicle (UAV) for wildlife surveillance**. Tese (Doutorado) — University of Florida, 2004. Citado na página 23.

LEHNER, P.; ALBU-SCHÄFFER, A. The repetition roadmap for repetitive constrained motion planning. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 4, p. 3884–3891, 2018. Citado na página 33.

MANFREDO, D. C. Geometria diferencial de curvas e superfícies. **Rio de Janeiro: IMPA**, 1976. Citado na página 31.

MITTAL, S. A study of successive over-relaxation method parallelization over modern hpc languages. **arXiv preprint arXiv:1401.0763**, 2014. Citado na página 63.

MOON, C.-b.; CHUNG, W. Kinodynamic planner dual-tree rrt (dt-rrt) for two-wheeled mobile robots using the rapidly exploring random tree. **IEEE Transactions on industrial electronics**, IEEE, v. 62, n. 2, p. 1080–1090, 2015. Citado na página 36.

MOTONAKA, K.; GOTO, T.; WATANABE, K.; MAEYAMA, S. Kinodynamic motion planning for a two-wheeled drive mobile robot using a harmonic potential field. In: IEEE. **IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society**. [S.l.], 2016. p. 6140–6145. Citado na página 39.

ORTEGA, J. M. **Numerical Analysis - A Second Course**. [S.l.]: Academic Press, 1972. 102-123 p. ISBN 0125285604. Citado nas páginas 53, 87 e 88.

PARSONS, D.; CANNY, J. Geometric problems in molecular biology and robotics. In: **International Conference on Intelligent Systems for Molecular Biology**. [S.l.: s.n.], 1994. p. 322–330. Citado na página 22.

PRESS, W.; TEUKOLSKY, S.; VETTERLING, W.; FLANNERY, B. **Numerical Recipes in C: The art of Scientific Computing**. 2nd. ed. [S.l.]: Cambridge University Press, 1992. Citado na página 47.

PRESTES, E. **Navegação Exploratória Baseada em Problemas de Valores de Contorno**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, 2003. Citado nas páginas 39, 42 e 47.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE, JAPAN. **ICRA workshop on open source software**. [S.l.], 2009. v. 3, n. 3.2, p. 5. Citado na página 59.

REMONDINO, F.; BARAZZETTI, L.; NEX, F.; SCAIONI, M.; SARAZZI, D. Uav photogrammetry for mapping and 3d modeling–current status and future perspectives. **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 38, n. 1, p. C22, 2011. Citado na página 23.

SCHLOTFELDT, B.; THAKUR, D.; ATANASOV, N.; KUMAR, V.; PAPPAS, G. J. Anytime planning for decentralized multirobot active information gathering. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 2, p. 1025–1032, 2018. Citado na página 22.

SHARKEY, N. The ethical frontiers of robotics. **Science**, American Association for the Advancement of Science, v. 322, n. 5909, p. 1800–1801, 2008. Citado na página 21.

SILVA, M. O. da. **Campos potenciais modificados aplicados ao controle de múltiplos robôs**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, São Carlos / SP, 2011. Disponível em <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-23112011-090055/pt-br.php>. Citado nas páginas 39, 46, 47 e 52.

SILVA, M. O. da. **Motion Planning Library**: Repositório github. 2014. Disponível em: <https://github.com/LAR-USP/MoPlaLib>. Acesso em: 01 Mai. 2018. Citado na página 59.

SILVA, M. O. da; ROMERO, R. A. F.; CAURIN, G. A. de P. Analysis of the convergence of locally oriented potential fields. In: REIS, L. P.; CORREIA, L.; LAU, N.; BIANCHI, R. (Ed.). **IROBOT 2008 - 3rd International Workshop on Intelligent Robotics**. Lisbon University Institute - ISCTE, Lisbon, Portugal: [s.n.], 2008. ISBN 972886208-3. Citado nas páginas 47 e 49.

SILVA, M. O. da; TOMAZELA, L.; ROMERO, R. A. F. Extending numerical solutions of potential fields method based on boundary value problems for 3d environments. In: MACAU, E. E. N. (Ed.). **6th Internacional Conference on Nonlinear Science and Complexity - NSC2016**. INPE, São José dos Campos, SP, Brazil: [s.n.], 2016. Citado nas páginas 57, 63 e 64.

SMITH, G. D. **Numerical Solution of partial differential equations: finite difference methods**. [S.l.]: Oxford University, 1992. ISBN 0198596502. Citado na página 46.

SONG, G.; AMATO, N. M. Using motion planning to study protein folding pathways. In: **Proceedings of the fifth annual international conference on Computational biology**. New York, NY, USA: ACM, 2001. (RECOMB '01), p. 287–296. ISBN 1-58113-353-7. Disponível em: <http://doi.acm.org/10.1145/369133.369239>. Citado na página 22.

____. A motion-planning approach to folding: from paper craft to protein folding. **Robotics and Automation, IEEE Transactions on**, v. 20, n. 1, p. 60 – 71, feb. 2004. ISSN 1042-296X. Citado na página 22.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. [S.l.]: Wiley, 2005. 496 p. ISBN 0471649902. Citado na página 25.

STEFANO, L. D.; CLEMENTINI, E.; STAGNINI, E. Reactive obstacle avoidance for multicopter uavs via evaluation of depth maps. In: FOGLIARONI, P.; BALLATORE, A.; CLEMENTINI, E. (Ed.). **Proceedings of Workshops and Posters at the 13th International Conference on Spatial Information Theory (COSIT 2017)**. Cham: Springer International Publishing, 2018. p. 41–43. ISBN 978-3-319-63946-8. Citado na página 21.

STRAUSS, W. A. **Partial Differential Equations: an introduction**. [S.l.]: John Wiley and Sons, 1992. ISBN 0471548685. Citado na página 46.

TAKAYAMA, L.; JU, W.; NASS, C. Beyond dirty, dangerous and dull: what everyday people think robots should do. In: ACM. **Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction**. [S.l.], 2008. p. 25–32. Citado na página 21.

The Qt Company Ltd. **Qt 5.11 - Documentation**. [S.l.], 2018. Disponível em: <https://doc.qt.io/qt-5.11/index.html>. Citado na página 59.

THOMAS, S.; SONG, G.; AMATO, N. M. Protein folding by motion planning. **Physical Biology**, Institute of Physics Publishing, v. 2, p. 148–155, Nov 2005. ISSN 1478-3967. Citado na página 22.

TON, C.; KAN, Z.; MEHTA, S. S. Obstacle avoidance control of a human-in-the-loop mobile robot system using harmonic potential fields. **Robotica**, Cambridge University Press, v. 36, n. 4, p. 463–483, 2018. Citado na página 39.

YANG, S.; KONAM, S.; MA, C.; ROSENTHAL, S.; VELOSO, M. M.; SCHERER, S. Obstacle avoidance through deep networks based intermediate perception. **CoRR**, abs/1704.08759, 2017. Disponível em: <http://arxiv.org/abs/1704.08759>. Citado na página 21.

ČAPEK, K. **Rossumovi Univerzální Roboti: Rossum's Universal Robots**. Praga, República Checa: [s.n.], 1920. Peça de Teatro. Citado na página 21.

APÊNDICE

A

SOLUÇÃO NUMÉRICA DE SISTEMAS LINEARES

Conforme Burden, Faires e Reynolds (1978), sistemas lineares estão associados com muitos problemas na engenharia e nas ciências, bem como com aplicações da matemática às ciências sociais e ao estudo quantitativo de problemas de negócios e da economia. Particularmente, a solução numérica de Equações Diferenciais Parciais depende de uma etapa de solução numérica de sistemas lineares.

Métodos numéricos para solução de sistemas de quações lineares são divididos principalmente em dois grupos (FRANCO, 2008):

Métodos Diretos São aqueles métodos que forneceriam a solução exata, não fossem os erros de arredondamento, com um número finito de passos.

Métodos Iterativos São aqueles que permitem obter a solução de um sistema linear com uma dada precisão, através de um processo infinito convergente.

Ainda no trabalho de Franco (2008), é ressaltado o fato de que os erros de arredondamento dos métodos diretos, em sistemas de grande porte, podem tornar a solução sem significado, enquanto que em métodos iterativos os erros não se acumulam. Já no trabalho de Burden, Faires e Reynolds (1978) é dito que matrizes esparsas[1] são frequentemente resolvidas utilizando-se técnicas iterativas.

Tais características fazem dos métodos iterativos os mais apropriados para a solução numérica de sistemas lineares decorrentes de EDP.

Um método iterativo consiste em uma técnica para resolver um determinado sistema linear $Ax = b$ (Equação A.1) que a partir de uma aproximação inicial $x^{(0)}$ para a solução x, gera

[1] com um grande volume de entradas iguais a zero

uma sequência de vetores $\left\{x^{(k)}\right\}_{k=1}^{\infty}$, que converge para x. Estas técnicas transformam o sistema $Ax = b$ em um sistema equivalente na forma da Equação A.2, para alguma matriz T e algum vetor c fixos. Após o vetor inicial $x^{(0)}$ ter sido obtido, a sequência de vetores para aproximar a solução é gerada calculando-se como na Equação A.3

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & + & \vdots & + & \ddots & + & \vdots & = & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{cases} \tag{A.1}$$

$$x = Tx + c \tag{A.2}$$

$$x^k = Tx^{(k-1)} + c \qquad para \quad k = 1, 2, 3, \ldots \tag{A.3}$$

A sequência apresentada na Equação A.3 é infinita e para tanto critérios de parada precisam ser estabelecidos. Dentre os mais comuns, destaca-se:

- **Quantidade de Iterações** que estabelece um número máximo, ou fixo, de iterações
- **Tempo de Execução** que estabelece um tempo máximo ou fixo de processamento
- **Convergência** que estabelece uma métrica $D : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ e um limiar l_{conv} de tal forma que $D(x^{(k)}, x^{(k-1)}) \leq l_{conv}$

Pode-se estabelecer critérios de parada mais complexos apenas por associar estes três tipos básicos.

A.1 Jacobi-Richardson

A matriz A do sistema linear da Equação A.1, pode ser decomposta como na Equação A.4.

$$A = L + D + R \tag{A.4}$$

onde $L = (l_{ij})$ é uma matriz triangular inferior formada pela parte inferior da matriz A, $D = (d_{ij})$ é uma matriz formada pela diagonal de A e $R = (r_{ij})$ é uma matriz triangular superior formada pela parte superior da matriz A, ou seja:

$$l_{ij} = \begin{cases} a_{ij}, & se \quad i > j \\ 0, & c.c. \end{cases}$$

$$d_{ij} = \begin{cases} a_{ij}, & se \quad i = j \\ 0, & c.c. \end{cases}$$

$$r_{ij} = \begin{cases} a_{ij}, & se \quad i < j \\ 0, & c.c. \end{cases}$$

Supondo que $det(D) \neq 0$, o sistema linear original (Equação A.1) pode ser transformado na Equação A.5. Então, pode ser definido o processo iterativo da Equação A.6, que é chamado de *Método de Jacobi-Richardson*. Pode ser visto que o processo iterativo da Equação A.6 está na forma padrão da Equação A.3, com $T = -D^{-1}(L+R)$ e $c = D^{-1}b$.

$$\begin{aligned} Ax &= b \Rightarrow \\ (L+D+R)x &= b \Rightarrow \\ Dx &= -(L+R)x + b \Rightarrow \\ x &= -D^{-1}(L+R)x + D^{-1}b \end{aligned} \tag{A.5}$$

$$x^{(k)} = -D^{-1}(L+R)x^{(k-1)} + D^{-1}b \tag{A.6}$$

Por hipótese, $a_{ii} \neq 0$, pois foi suposto que $det(D) \neq 0$. Então, antes de se decompor a matriz A em $L+D+R$, cada linha da matriz A no sistema da Equação A.1 pode ser dividido pelo correspondente termo da diagonal principal, resultando na decomposição da Equação A.7, onde A^* é a matriz obtida após divisão, I é a matriz identidade, L^* e R^* são as matrizes triangular inferior e superior, respectivamente, de A^*, que são dados como nas Equações A.8 - A.9.

$$A^* = L^* + I + R^* \tag{A.7}$$

$$l^*_{ij} = \begin{cases} a^*_{ij} = \dfrac{a_{ij}}{a_{ii}}, & se \quad i > j \\ 0, & c.c. \end{cases} \tag{A.8}$$

$$r^*_{ij} = \begin{cases} r^*_{ij} = \dfrac{a_{ij}}{a_{ii}}, & se \quad i < j \\ 0, & c.c. \end{cases} \tag{A.9}$$

Assim, o processo iterativo da Equação A.6, pode ser reescrito como na Equação A.10, onde b^* é como na Equação A.11, e é o vetor b obtido após a divisão de A anteriormente citada.

$$x^{(k)} = -(L^* + R^*)x^{(k-1)} + b^* \tag{A.10}$$

$$b^*_i = \frac{b_i}{a_{ii}}, \qquad para \quad i = 1, 2, 3, \ldots \tag{A.11}$$

Pelo processo iterativo da Equação A.10, não é necessário calcular a matriz D^{-1}. Este processo pode ser visto expandido na Equação A.12. Neste processo, devido a atualização

simultânea de todas as equações do sistema linear, este método também é conhecido por *Método dos Deslocamentos Simultâneos* ou, simplesmente, *Método de Jacobi* (FRANCO, 2008).

$$\begin{cases} x_1^{(k+1)} & = & - & a_{12}^* x_2^{(k)} & - & a_{13}^* x_3^{(k)} & - & \dots & - & a_{1n}^* x_n^{(k)} & + & b_1^* \\ x_2^{(k+1)} & = & - & a_{21}^* x_1^{(k)} & - & a_{23}^* x_3^{(k)} & - & \dots & - & a_{2n}^* x_n^{(k)} & + & b_2^* \\ x_3^{(k+1)} & = & - & a_{31}^* x_1^{(k)} & - & a_{32}^* x_2^{(k)} & - & \dots & - & a_{3n}^* x_n^{(k)} & + & b_3^* \\ \vdots & = & - & \vdots & - & \vdots & - & \ddots & - & \vdots & + & \vdots \\ x_n^{(k+1)} & = & - & a_{n1}^* x_1^{(k)} & - & a_{n2}^* x_2^{(k)} & - & \dots & - & a_{n,n-1}^* x_{n-1}^{(k)} & + & b_n^* \end{cases} \tag{A.12}$$

A.2 Gauss-Seidel

Seja o sistema linear $Ax = b$, tal como no método de Jacobi-Richardson, este sistema pode ser reescrito como $(L^* + I + R^*)x = b^*$ (ver Equação A.7). Tal sistema, agora pode ser transformado como na Equação A.13, que agora tem a forma da Equação A.2, com $T = -(L^* + I)^{-1}R^*$ e $c = (L^* + I)^{-1}b^*$.

$$\begin{aligned} Ax &= b \Rightarrow \\ (L^* + I + R^*)x &= b^* \Rightarrow \\ (L^* + I)x &= -R^*x + b^* \Rightarrow \\ x &= -(L^* + I)^{-1}R^*x + (L^* + I)^{-1}b^* \end{aligned} \tag{A.13}$$

$$x^{(k)} = -(L^* + I)^{-1}R^*x^{(k-1)} + (L^* + I)^{-1}b^* \tag{A.14}$$

O processo iterativo da Equação A.14 é conhecido como *Método de Gauss-Seidel*. Pela Equação A.14, pode ser visto que o método, nesta forma, necessita do cálculo de $(L^* + I)^{-1}$. Este processo pode ser simplificado como na Equação A.15

$$\begin{aligned} x^{(k)} &= -(L^* + I)^{-1}R^*x^{(k-1)} + (L^* + I)^{-1}b^* \Rightarrow \\ (L^* + I)x^{(k)} &= -R^*x^{(k-1)} + b^* \Rightarrow \\ x^{(k)} &= -L^*x^{(k)} - R^*x^{(k-1)} + b^* \end{aligned} \tag{A.15}$$

Pelo processo iterativo da Equação A.15, não é necessário calcular a matriz $(L^* + I)^{-1}$. Este processo pode ser visto expandido na Equação A.16. Este processo iterativo é também é conhecido por *Método dos Deslocamentos Sucessivos*, por utilizar componentes x_i^k no cálculo de x_j^k, para $j > i$. (FRANCO, 2008).

$$\begin{cases} x_1^{(k+1)} & = & - & a_{12}^* x_2^{(k)} & - & a_{13}^* x_3^{(k)} & - & \dots & - & a_{1n}^* x_n^{(k)} & + & b_1^* \\ x_2^{(k+1)} & = & - & a_{21}^* x_1^{(k+1)} & - & a_{23}^* x_3^{(k)} & - & \dots & - & a_{2n}^* x_n^{(k)} & + & b_2^* \\ x_3^{(k+1)} & = & - & a_{31}^* x_1^{(k+1)} & - & a_{32}^* x_2^{(k+1)} & - & \dots & - & a_{3n}^* x_n^{(k)} & + & b_3^* \\ \vdots & = & - & \vdots & - & \vdots & - & \ddots & - & \vdots & + & \vdots \\ x_n^{(k+1)} & = & - & a_{n1}^* x_1^{(k+1)} & - & a_{n2}^* x_2^{(k+1)} & - & \dots & - & a_{n,n-1}^* x_{n-1}^{(k+1)} & + & b_n^* \end{cases} \tag{A.16}$$

A.3 Weighted Jacobi

O Método Weighted Jacobi consiste em uma modificação no Método de Jacobi com o objetivo de acelerar sua convergência. Tal modificação traduz-se em uma ponderação entre $x^{(k)}$ e $x_J^{(k+1)}$ (aproximação obtida a partir do método de Jacobi) tal como na Equação A.17

$$x^{(k+1)} = x^k + \omega \left(x_J^{(k+1)} - x^k \right) \qquad \text{com } 0 < \omega < 2 \tag{A.17}$$

Realizando manipulações algébricas é possível mostrar que a forma padrão (Equação A.2) do método WJ resulta em $T = (1-\omega)I + \omega \left(D^{-1} (L+R) \right)$ e $c = -\omega D^{-1} b$.

Se $\omega = 1$, o método WJ recai no método de Jacobi. A escolha da constante ω ótima depende das propriedades da matriz A da Equação A.1.

A.4 Successive Over-Relaxation

Tal como na Sessão A.3, o método de Gauss-Seidel pode ser modificado, criando o método Successive Over-Relaxation (ou simplesmente SOR), como na Equação A.18.

$$x^{(k+1)} = (1-\omega)x^k + \omega x_{GS}^{(k+1)} \qquad \text{com } 0 < \omega < 2 \tag{A.18}$$

Realizando manipulações algébricas é possível mostrar que a forma padrão (Equação A.2) do método SOR resulta em $T = (D-\omega L)^{-1}[(1-\omega)D + \omega R]$ e $c = \omega(D-\omega L)^{-1} b$.

Analogamente a Seção A.3, se $\omega = 1$, o método SOR recai no método de Gauss-Seidel. A escolha da constante ω ótima depende das propriedades da matriz A da Equação A.1.

APÊNDICE

B

DEFINIÇÕES E TEOREMAS

Neste Apêndice, são apresentados alguns conceitos matemáticos necessários para a prova da convergência dos métodos baseados em PVC.

Definição 1. É dito que um grafo é *fortemente conectado se e somente se para qualquer par de nós P_i, P_j existir um caminho*

$$\overrightarrow{P_iP_{i_1}}, \overrightarrow{P_{i_1}P_{i_2}}, \ldots, \overrightarrow{P_{i_m}P_j}$$

conectando P_i e P_j.(CORMEN et al., 2001; ORTEGA, 1972)

Definição 2. Uma matriz $A \in \mathscr{L}(\Re^n)$ (onde $\mathscr{L}(\Re^n)$ é o espaço dos operadores lineares de $\Re^n$ a $\Re^n$) é diagonalmente dominante se

$$|a_{ii}| \geq \sum_{\substack{j=0 \\ j \neq i}}^{n} |a_{ij}|, \qquad i = 1, \ldots, n \tag{B.1}$$

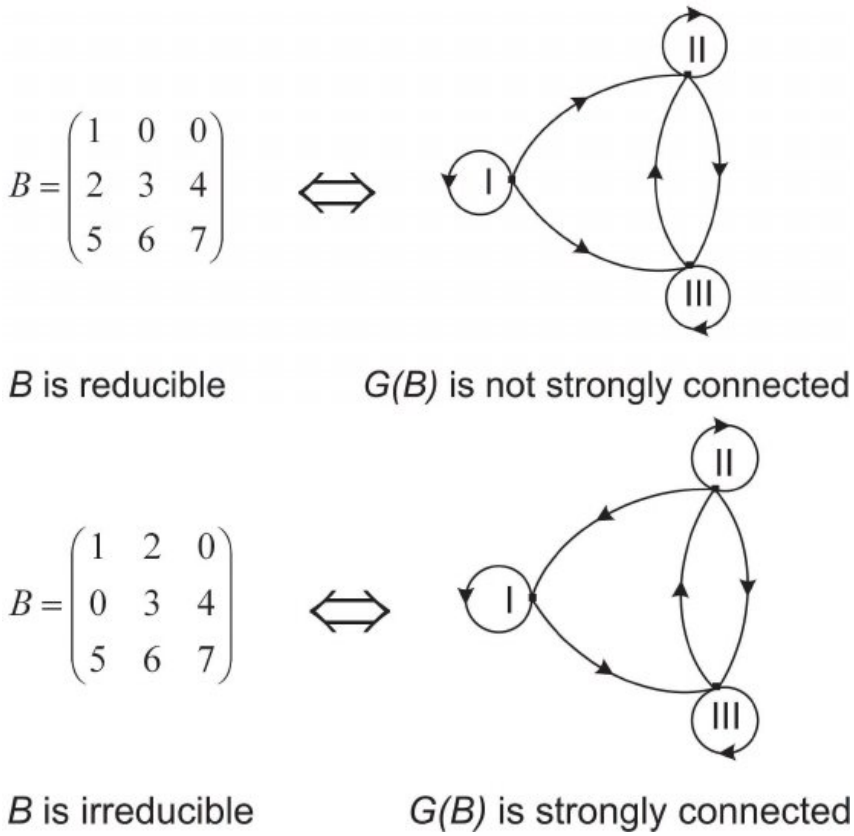
e *estritamente diagonalmente dominante* se as inequações são estritas na Equação B.1 para todo i. Uma matriz é *diagonalmente dominante e irredutível* se é irredutível, diagonalmente dominante e, para pelo menos i na Equação B.1 a inequação é estrita. (ORTEGA, 1972)

Definição 3. Com uma matriz $A_{n \times n}$, pode ser associado um grafo como a seguir. Sejam n pontos distintos $P_1, \ldots, P_n$, que serão chamados vértices e, para cada elemento não nulo a_{ij} de A constrói-se um aresta de P_i a P_j. Como resultado, há um dígrafo (grafo dirigido) associado com a matriz A.(ORTEGA, 1972)

Tal processo é ilustrado na Figura .

Teorema 1. Uma matriz $A \in \mathscr{L}(\Re^n)$ é irredutível se e somente se o grafo associado a ela é fortemente conectado.(ORTEGA, 1972)

Figura 26 – Associação de Dígrafo a Matrizes



Fonte: Guo, Li e Shuai (2006).

Teorema 2. Assumindo que $b \in \Re^n$ é arbitrário e que $A \in \mathscr{L}(\Re^n)$ ou é estritamente diagonalmente dominante ou é diagonalmente dominante e irredutível, então, o método de Jacobi-Richardson e o método de Gauss-Seidel convergem para $A^{-1}b$ para qualquer x_0 dado.(ORTEGA, 1972)

ICMC USP
SÃO CARLOS