



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

**JOSIAS GUIMARÃES BATISTA**

**UM ESTUDO DE CASO SOBRE A INTERAÇÃO MÁQUINA-MÁQUINA NA  
PERSPECTIVA DOS ALGORITMOS CLÁSSICOS DE GERAÇÃO DE CAMINHOS  
LIVRES DE COLISÃO**

**FORTALEZA**

**2017**

JOSIAS GUIMARÃES BATISTA

UM ESTUDO DE CASO SOBRE A INTERAÇÃO MÁQUINA-MÁQUINA NA  
PERSPECTIVA DOS ALGORITMOS CLÁSSICOS DE GERAÇÃO DE CAMINHOS LIVRES  
DE COLISÃO

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

Orientador: Prof. Dr. George André Pereira Thé

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B337e Batista, Josias Guimarães.

Um estudo de caso sobre a interação máquina-máquina na perspectiva dos algoritmos clássicos de geração de caminhos livres de colisão / Josias Guimarães Batista. – 2017.

106 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2017.

Orientação: Prof. Dr. George André Pereira Thé.

1. Manipulador SCARA. 2. Campos Potenciais Artificiais. 3. Planejamento de Caminho Livre de Colisão. 4. Intereração Máquina-máquina. 5. Produtividade e Eficiência. I. Título.

CDD 621.3

---

JOSIAS GUIMARÃES BATISTA

UM ESTUDO DE CASO SOBRE A INTERAÇÃO MÁQUINA-MÁQUINA NA  
PERSPECTIVA DOS ALGORITMOS CLÁSSICOS DE GERAÇÃO DE CAMINHOS LIVRES  
DE COLISÃO

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

Aprovada em: 17 de Julho de 2017

BANCA EXAMINADORA

---

Prof. Dr. George André Pereira Thé (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Guilherme de Alencar Barreto  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Fabrício Gonzalez Nogueira  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Ao meu orientador, Prof. Dr. George André Pereira Thé, pela orientação efetiva, compreensão, incentivo e valiosa colaboração.

A minha esposa, Amanda, pelo amor, cuidado e atenção.

Aos meus pais pelo exemplo de vida, superação e conquista.

Ao meu amigo Leonardo e aos colegas Nícolas, Tarcísio Bruno e a todos que de alguma forma contribuíram para a conclusão deste trabalho.

A todos do CENTAURO, pelo aprendizado, pela infraestrutura e por todo o suporte oferecido.

A Deus por tudo.

## RESUMO

Neste trabalho é apresentada uma proposta de implementação do algoritmo de Campos Potenciais Artificiais (CPA) em um manipulador *Selective Compliant Assembly Robot Arm* (SCARA). Explora-se um estudo de caso sobre a interação máquina-máquina, representados pelo manipulador e um robô móvel. Esse estudo foi realizado através da implementação do algoritmo de Campos Potenciais Artificiais (CPA), que é um algoritmo clássico, o mesmo é utilizado na geração de caminhos livres de colisão em um ambiente dinâmico, monitorado por um sensor de imagens, em tempo real.

São exploradas várias abordagens entre o manipulador e o robô móvel. Abordagens essas, que levam em consideração as combinações entre o robô móvel e o manipulador como corpo extenso e ponto. O algoritmo foi implementado no manipulador para gerar os caminhos livres de colisão, com obstáculo dinâmico, com posições aleatórias, em tempo real. Os resultados apresentam a eficiência do algoritmo com relação à produtividade e se mostraram satisfatórios nesse estudo de caso. Também são apresentados os resultados da eficiência do algoritmo através do OEE (*Overall Equipment Effectiveness*), o consumo de energia e os resultados da análise do erro entre as posições geradas pelo algoritmo de campos potenciais e as posições reais do manipulador.

**Palavras-chave:** Manipulador SCARA. Campos Potenciais Artificiais. Planejamento de Caminho Livre de Colisão. Geração de Caminho e Trajetória. Interação Máquina-máquina. Produtividade e Eficiência.

## ABSTRACT

This paper presents a proposal for the implementation of the Artificial Potential Fields algorithm (APF) in a SCARA manipulator (Selective Compliant Assembly Robot Arm). We explore a case study on the machine-machine interaction, represented by the manipulator and a mobile robot. This study was carried out through the implementation of the APF algorithm, which is a classic algorithm, the same is used in the generation of collision free paths in a dynamic environment, monitored by an image sensor, in real time..

Several approaches are explored between the manipulator and the mobile robot. These approaches, which take into account the combinations between the mobile robot and the manipulator as the extended body and point. The algorithm was implemented in the manipulator to generate the collision-free, dynamic obstacle paths with random positions, in real time. The results show the efficiency of the algorithm with respect to productivity and were satisfactory in this case study. We also present the results of the efficiency of the algorithm through OEE (Overall Equipment Effectiveness), the energy consumption and the results of the error analysis between the positions generated by the potential field algorithm and the actual positions of the manipulator.

**Keywords:** SCARA Manipulator. Artificial Potential Field. Collision-Free Path Planning. Path Generation. Machine-machine Interaction. Productivity and Efficiency.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Informações sobre o mercado mundial de robôs . . . . .	16
Figura 2 – Sensor de segurança a laser . . . . .	17
Figura 3 – Funcionamento do controle cinemático . . . . .	27
Figura 4 – Transformações da conversão D-H . . . . .	29
Figura 5 – Sistemas de coordenadas das juntas do robô SCARA . . . . .	30
Figura 6 – Tangente trigonométrica de uma diferença angular para determinar $\theta_1$ . . . . .	33
Figura 7 – Deslocamento das juntas no tempo para execução de trajetória . . . . .	36
Figura 8 – Posição, velocidade e aceleração em uma interpolação linear . . . . .	42
Figura 9 – Sequência de execução para implementar o algoritmo de CPA . . . . .	51
Figura 10 – Robô SCARA . . . . .	52
Figura 11 – Sensor industrial de imagens PMD O3D200 . . . . .	53
Figura 12 – Robô Zumo - partes e conexões . . . . .	54
Figura 13 – Robô Zumo com arduino Uno . . . . .	55
Figura 14 – Caminhos arbitrários para o robô móvel . . . . .	55
Figura 15 – Imagens obtidas a partir do sensor de imagem: a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	56
Figura 16 – Diagrama de blocos com as etapas do processamento das imagens do sensor . . . . .	57
Figura 17 – Imagens limiarizadas utilizando método de Otsu: a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	57
Figura 18 – Imagens após a extração de componentes conexos: a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	58
Figura 19 – Circunscrição de obstáculo: a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	59
Figura 20 – Espaços de configuração de obstáculos ( $QO$ ). a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	61
Figura 21 – Caminho realizado pelo CPA a partir de $QO$ . a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	62
Figura 22 – Linhas do campo potencial e caminhos realizado pelo algoritmo de CPA. a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	63
Figura 23 – Superfícies resultantes do algoritmo de CPA. a) Cena 01 a esquerda; b) Cena 02 a direita . . . . .	63
Figura 24 – Funcionamento do processo de implementação do CPA . . . . .	65

Figura 25 – Aquisição da imagem pelo sensor de imagens . . . . .	66
Figura 26 – Identificação do obstáculo na cena . . . . .	67
Figura 27 – Círculo em volta do obstáculo . . . . .	67
Figura 28 – Caminho livre de colisão do manipulador . . . . .	68
Figura 29 – Representação do manipulador e robô móvel. a) à esquerda: manipulador e robô móvel considerados como corpos extensos; b) à direita: manipulador e robô móvel considerados como pontos . . . . .	70
Figura 30 – Torque do motor 1 para abordagem corpo extenso - ponto, missão: zona . .	80
Figura 31 – Torque do motor 1 para abordagem ponto - corpo extenso, missão: zona . .	81
Figura 32 – Caminho gerado pelo algoritmo de CPA x caminho real no espaço cartesiano para a abordagem corpo extenso - ponto, missão zona . . . . .	83
Figura 33 – Caminho gerado pelo algoritmo de CPA x caminho real no espaço cartesiano para a abordagem ponto - corpo extenso, missão ponto . . . . .	83
Figura 34 – Elementos de um sistema de processamento de imagens . . . . .	91
Figura 35 – Passos fundamentais em processamento de imagens digitais . . . . .	93
Figura 36 – Operação de erosão de A (esquerda) por B (central) resultando na imagem a direita . . . . .	94
Figura 37 – Operação de dilatação de A (esquerda) por B (central) resultando na imagem a direita . . . . .	95
Figura 38 – Bancada de Acionamento e Controle do Manipulador SCARA . . . . .	97
Figura 39 – Imagem da Bancada de Acionamento e CLP . . . . .	98
Figura 40 – Manipulador e robô móvel . . . . .	98
Figura 41 – Montagem do sensor de imagens . . . . .	99
Figura 42 – Sistema com o sensor de imagens, manipulador e robô móvel . . . . .	99
Figura 43 – Variáveis do OEE . . . . .	100

## LISTA DE TABELAS

Tabela 1 – Parâmetros D-H do Manipulador SCARA . . . . .	30
Tabela 2 – Dados de placa do motor da junta 1 . . . . .	73
Tabela 3 – Resultados dos experimentos na implementação do algoritmo de CP para cada abordagem e cada missão . . . . .	76
Tabela 4 – Valores da eficiência do algoritmo de CPA para as abordagens ensaiadas . .	76
Tabela 5 – Valores da variabilidade do tempo da missão . . . . .	77
Tabela 6 – Valores de disponibilidade, performance, qualidade e OEE para as abordagens	78
Tabela 7 – Consumo de energia mecânica, em kWh, do motor 1 para as abordagens implementadas . . . . .	79
Tabela 8 – Erro médio e desvio padrão entre os ângulos gerado pelo algoritmo de CP ( <i>set-points</i> ) e os ângulos da posição real das juntas do manipulador . . . . .	82

## LISTA DE ABREVIATURAS E SIGLAS

CLP	Controlador Lógico Programável
CPA	Campos Potenciais Artificiais
CPP	Controle Ponto a Ponto
CTC	Controle por Trajetória Contínua
D-H	Denavit-Hartenberg
GDL	Graus de liberdade
OEE	<i>Overall Equipment Effectiveness</i>
OLE	<i>Object Linking and Embedding</i>
OPC	Controle de Processos OLE
PDI	Processamento Digital de Imagens
PNP	Tempo para Paradas não Planejadas
PP	Tempo para Paradas Planejadas
SCADA	<i>Supervisory Control and Data Acquisition</i>
SCARA	<i>Selective Compliant Assembly Robot Arm</i>
TEO	Tempo Efetivo de Operação
TO	Tempo de Operação
TTD	Tempo Total Disponível
VANT	Veículos Aéreos Não Tripulados

## LISTA DE SÍMBOLOS

$A$	Matriz de transformação homogênea
$A(q)$	Subconjunto do espaço de trabalho
$d$	Distância de referência
$E$	Energia do motor 1
$f$	Frequência do motor 1
$F(x)$	Campo vetorial de forças
$F(\theta)$	Forças de atrito dinâmico e estático nas juntas do manipulador
$G(\theta)$	Vetor de gravidade
$K_a$	Constante de atração
$K_r$	Contante de repulsão,
$\eta_{CP}$	Eficiência do algoritmo de campos potenciais
$n$	Quantidade das posições medidas
$O_i$	Obstáculos no espaço de trabalho
$p$	Distância de referência
$P$	Potência nominal do motor 1
$P_{m1}$	Potência do motor 1
$N_{m1}$	Velocidade do motor 1
$q$	Vetor de variáveis das articulações
$Q$	Espaço de configuração
$QO$	Espaço de configurações de obstáculos
$\sigma_{temp}$	Variância do tempo das missões
$U(\theta)$	Potencial resultante do campo potencial artificial
$U_{atr}(\theta)$	Potencial atrativo
$U_{rep}(\theta)$	Potencial repulsivo
$U(\theta)$	Campo escalar potencial
$W$	Espaço de trabalho do robô

$t$	Tempo
$T_{m1}$	Torque do motor 1
$\theta$	Ângulo de junta do manipulador
$\theta_f$	Configuração final desejada
$\theta_{CP}$	Posição gerada pelo algoritmo do CP
$\theta_R$	Posição real da junta ()
$\nabla(U(\theta))$	Gradiente do campo potencial

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>16</b>
<b>1.1</b>	<b>Importância do trabalho . . . . .</b>	<b>16</b>
<b>1.2</b>	<b>Utilização de campos potenciais artificiais em trajetórias livres de colisão</b>	<b>19</b>
<b>1.3</b>	<b>Motivação e problemática . . . . .</b>	<b>22</b>
<b>1.4</b>	<b>Objetivos . . . . .</b>	<b>23</b>
<b>1.4.1</b>	<b><i>Objetivo geral . . . . .</i></b>	<b>23</b>
<b>1.4.2</b>	<b><i>Objetivos específicos . . . . .</i></b>	<b>23</b>
<b>1.5</b>	<b>Produção científica . . . . .</b>	<b>24</b>
<b>1.6</b>	<b>Organização geral da dissertação . . . . .</b>	<b>24</b>
<b>2</b>	<b>MODELAGEM MATEMÁTICA DO MANIPULADOR . . . . .</b>	<b>26</b>
<b>2.1</b>	<b>Introdução . . . . .</b>	<b>26</b>
<b>2.2</b>	<b>Cinemática de manipuladores . . . . .</b>	<b>28</b>
<b>2.2.1</b>	<b><i>Cinemática direta . . . . .</i></b>	<b>28</b>
<b>2.2.2</b>	<b><i>Cinemática inversa . . . . .</i></b>	<b>32</b>
<b>3</b>	<b>GERAÇÃO DE CAMINHO E TRAJETÓRIA . . . . .</b>	<b>35</b>
<b>3.1</b>	<b>Introdução . . . . .</b>	<b>35</b>
<b>3.2</b>	<b>Geração de trajetória . . . . .</b>	<b>35</b>
<b>3.2.1</b>	<b><i>Geração de trajetórias no espaço Cartesiano . . . . .</i></b>	<b>37</b>
<b>3.2.2</b>	<b><i>Geração de trajetórias no espaço das juntas . . . . .</i></b>	<b>38</b>
<b>3.3</b>	<b>Tipos de trajetórias . . . . .</b>	<b>39</b>
<b>3.3.1</b>	<b><i>Trajetória degrau . . . . .</i></b>	<b>39</b>
<b>3.3.2</b>	<b><i>Trajetórias com velocidade trapezoidal . . . . .</i></b>	<b>40</b>
<b>3.3.3</b>	<b><i>Trajetórias de tempo mínimo . . . . .</i></b>	<b>40</b>
<b>3.3.4</b>	<b><i>Trajetórias polinomiais . . . . .</i></b>	<b>41</b>
<b>3.3.4.1</b>	<b><i>Trajetórias cúbicas . . . . .</i></b>	<b>41</b>
<b>3.3.5</b>	<b><i>Trajetórias lineares . . . . .</i></b>	<b>41</b>
<b>3.4</b>	<b>Planejamento de trajetória utilizando campos potenciais artificiais . . . . .</b>	<b>42</b>
<b>3.5</b>	<b>Geração de caminho utilizando campos potenciais artificiais . . . . .</b>	<b>44</b>
<b>3.5.1</b>	<b><i>Campo potencial de atração . . . . .</i></b>	<b>45</b>
<b>3.5.2</b>	<b><i>Campo potencial de repulsão . . . . .</i></b>	<b>46</b>

<b>3.6</b>	<b>Controle de trajetória . . . . .</b>	47
<b>3.6.1</b>	<b><i>Controle de trajetória dos robôs industriais . . . . .</i></b>	48
<b>3.6.1.1</b>	<b><i>Trajetórias ponto a ponto . . . . .</i></b>	48
<b>3.6.1.2</b>	<b><i>Movimento eixo a eixo . . . . .</i></b>	48
<b>3.6.1.3</b>	<b><i>Movimento simultâneo dos eixos . . . . .</i></b>	49
<b>3.6.1.4</b>	<b><i>Trajetórias coordenadas . . . . .</i></b>	49
<b>3.6.1.5</b>	<b><i>Trajetórias contínuas . . . . .</i></b>	49
<b>4</b>	<b>DESCRIÇÃO DO EXPERIMENTO . . . . .</b>	51
<b>4.1</b>	<b>Introdução . . . . .</b>	51
<b>4.2</b>	<b>Materiais . . . . .</b>	52
<b>4.2.1</b>	<b><i>Manipulador SCARA . . . . .</i></b>	52
<b>4.2.2</b>	<b><i>Sensor de imagens . . . . .</i></b>	53
<b>4.2.3</b>	<b><i>Robô móvel (obstáculo móvel) . . . . .</i></b>	54
<b>4.3</b>	<b>Métodos . . . . .</b>	56
<b>4.3.1</b>	<b><i>Processamento digital de imagem . . . . .</i></b>	56
<b>4.3.1.1</b>	<b><i>Segmentação da imagem proveniente do sensor . . . . .</i></b>	57
<b>4.3.1.2</b>	<b><i>Extração de componentes conexos . . . . .</i></b>	58
<b>4.3.1.3</b>	<b><i>Cálculo do centro do obstáculo . . . . .</i></b>	58
<b>4.3.1.4</b>	<b><i>Cálculo do raio da circunferência que circunscreve o obstáculo . . . . .</i></b>	58
<b>4.3.2</b>	<b><i>Transformação do espaço cartesiano . . . . .</i></b>	59
<b>4.3.3</b>	<b><i>Simulação do algoritmo de CPA na geração de caminho livre de colisão . . . . .</i></b>	60
<b>4.3.3.1</b>	<b><i>Espaço de configurações de obstáculos . . . . .</i></b>	60
<b>4.3.3.2</b>	<b><i>Simulação de caminho com o algoritmo de campo potencial . . . . .</i></b>	61
<b>4.3.3.3</b>	<b><i>Área e superfície do campo potencial artificial . . . . .</i></b>	62
<b>4.4</b>	<b>Ensaios preliminares . . . . .</b>	64
<b>4.4.1</b>	<b><i>Sequência de operações . . . . .</i></b>	64
<b>4.4.2</b>	<b><i>Simulação do algoritmo de CPA na geração de caminho do manipulador . . . . .</i></b>	66
<b>4.5</b>	<b>Cenários de ensaios . . . . .</b>	68
<b>4.5.1</b>	<b><i>Implementação do algoritmo de campos potenciais . . . . .</i></b>	68
<b>4.5.2</b>	<b><i>Eficiência do algoritmo de campos potenciais . . . . .</i></b>	70
<b>4.5.3</b>	<b><i>Eficiência do algoritmo através do OEE . . . . .</i></b>	71
<b>4.5.4</b>	<b><i>Cálculo do torque e do consumo de energia mecânica do motor da junta 1 . . . . .</i></b>	72

<b>4.5.5</b>	<i>Cálculo do erro médio entre as posições reais e as posições geradas pelo algoritmo do CP</i>	74
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	75
<b>5.1</b>	<b>Resultados</b>	75
<b>5.1.1</b>	<i>Resultados da implementação do algoritmo de CPA na geração de caminho do manipulador</i>	75
<b>5.1.2</b>	<i>Avaliação do algoritmo de campos potenciais</i>	76
<b>5.1.3</b>	<i>Avaliação do algoritmo de campos potenciais através do OEE</i>	77
<b>5.1.4</b>	<i>Consumo de energia mecânica e análise do torque do motor da junta 1</i>	79
<b>5.1.5</b>	<i>Erro médio entre as posições geradas pelo algoritmo de CP e a posição real das juntas</i>	81
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	85
<b>6.1</b>	<b>Contribuições</b>	85
<b>6.2</b>	<b>Conclusões do trabalho</b>	85
<b>6.3</b>	<b>Sugestões de trabalhos futuros</b>	86
	<b>REFERÊNCIAS</b>	87
	<b>APÊNDICES</b>	91
	<b>APÊNDICE A – Fundamentação teórica de PDI</b>	91
<b>A.1</b>	<b>Processamento digital de imagens</b>	91
<b>A.1.1</b>	<i>Processamento morfológico de imagens</i>	93
<b>A.1.1.1</b>	<i>Erosão</i>	94
<b>A.1.1.2</b>	<i>Dilatação</i>	94
<b>A.1.2</b>	<i>Segmentação de imagens</i>	95
<b>A.1.2.1</b>	<i>Limiarização e o método de Otsu</i>	96
	<b>APÊNDICE B – Sistema e equipamentos</b>	97
	<b>APÊNDICE C – OEE - Overall Equipment Effectiveness</b>	100
<b>C.1</b>	<b>Eficiência Global do Equipamento</b>	100
<b>C.1.1</b>	<i>Disponibilidade (Índice de tempo operacional)</i>	100
<b>C.1.2</b>	<i>Performance (Índice de Desempenho Operacional)</i>	101
<b>C.1.3</b>	<i>Qualidade (Índice de Produtos Aprovados)</i>	101
<b>C.1.4</b>	<i>Cálculo do OEE</i>	102
	<b>APÊNDICE D – Código principal utilizado no arduino</b>	103

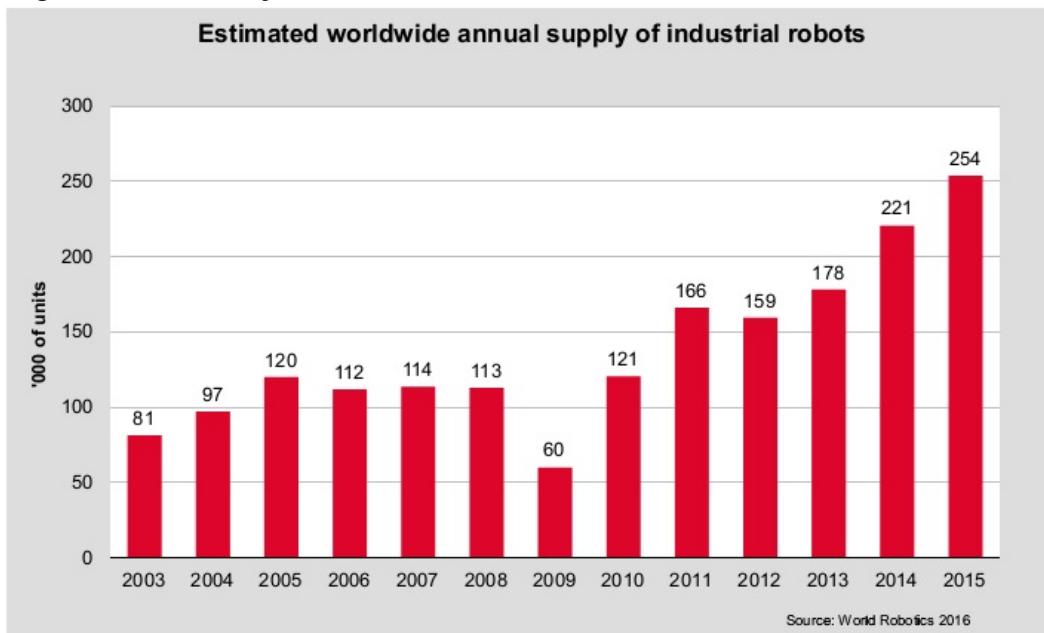
## 1 INTRODUÇÃO

Neste capítulo, é feita uma revisão bibliográfica sobre os trabalhos que utilizam planejamento de trajetórias livres de colisão, utilizando campos potenciais artificiais. Também serão apresentados os objetivos desse trabalho, bem como as produções científicas e a organização da dissertação.

### 1.1 Importância do trabalho

No intuito de suprir a sempre crescente atividade produtiva mundial, as indústrias dos mais diversos setores têm encontrado na modernização e na automação de seu chão de fábrica importantes aliados e, em especial, têm cada vez mais contado com a utilização de robôs na realização de tarefas. De fato, a utilização de robôs em tarefas de manipulação teve um crescimento significativo no contexto de produção industrial nos últimos anos, como mostram os dados da IFR (*International Federation of Robotics*), na figura 1.

Figura 1 – Informações sobre o mercado mundial de robôs



Fonte: IFR (2016).

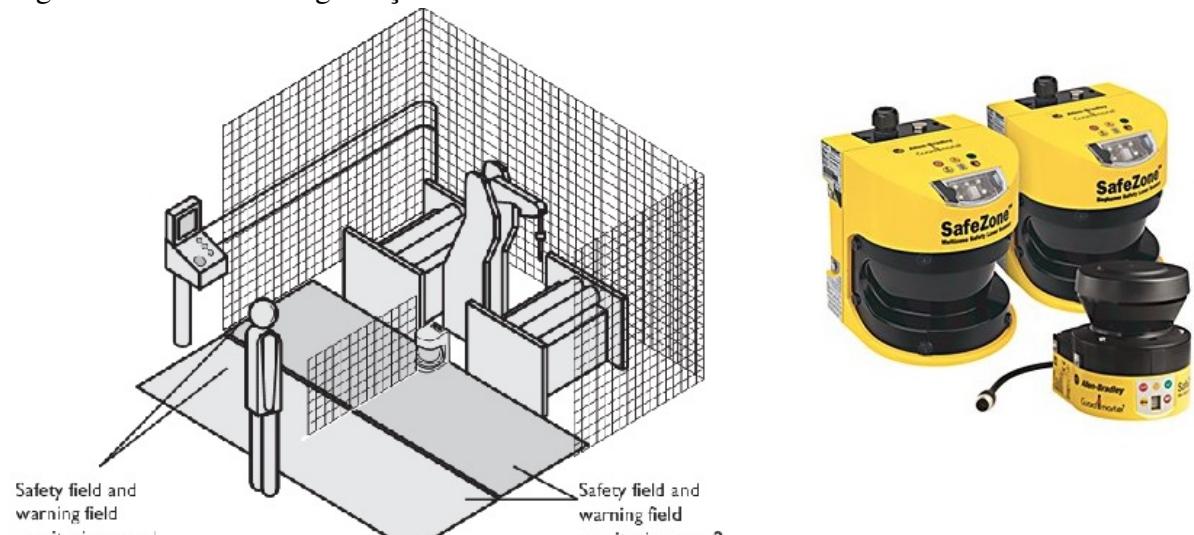
A referida difusão dos robôs em ambiente industrial propiciou, ao longo dos anos, que vários métodos fossem desenvolvidos com o intuito de monitorar e controlar robôs móveis ou robôs manipuladores, dando a eles a capacidade de operar em ambientes perigosos para os seres humanos como, por exemplo, além da atmosfera terrestre, em explorações aquáticas, no

transporte de materiais, entre outros (PINTO *et al.*, 2014).

Tipicamente, a execução de tarefas por robôs industriais ocorre em zonas classificadas, o que requer uma adequação do espaço circunstante; dependendo do segmento industrial e das normas vigentes, isso geralmente implica a construção de células ou ilhas para promover a devida separação entre a máquina e os operadores. A norma ISO 10218:1992 - *Manipulating Industrial Robots - Safety*, fornece orientações sobre as considerações de segurança para a concepção, construção, programação, operação, utilização, reparação e manutenção de robôs (ISO10218, 2011).

Mais ainda, também se recomenda o emprego de mecanismos de sensoreamento de presença de homens nos arredores da célula e o consequente desligamento da máquina em caso de invasão a célula. A figura 2 a seguir, ilustra uma solução típica para esta questão que se utiliza de sensores laser de varredura, a figura também apresenta uma aplicação da proteção de área perigosa com monitoração múltipla.

Figura 2 – Sensor de segurança a laser



Fonte: Fonte: Rockwell Automation (2017).

Uma reflexão sobre a questão levantada revela que há, em ambiente industrial, uma preocupação com a segurança e a saúde do operador. Entretanto, há que se considerar também a questão econômica decorrente; de fato, a interrupção do processo de fabricação em consequência da presença humana na célula robotizada gera atrasos e custos para a produção. Isso tem motivado o aparecimento de um campo de pesquisa recente, o qual trata justamente da coexistência segura de homens e máquinas móveis em zonas classificadas (ORIOLO, 2015). Em uma primeira linha de investigação, almeja-se, por exemplo, conhecer quais seriam as condições limites de operação

(isto é velocidade, posição, configuração do robô) em que a colisão com um humano na cena começaria a se apresentar com risco não-nulo. Mais ainda, numa segunda vertente de pesquisa, pode-se questionar se realmente é necessário o desligamento da máquina, isto é, não bastaria reduzir a velocidade de operação ou alterar sua trajetória?

Pensando em perspectiva, a questão acima pode ser estendida para a coexistência entre robôs. Neste contexto, assumiria um caráter desafiador, pois a forma do obstáculo móvel não é conhecida a priori (como seria o contorno de um corpo humano ou o contorno de outro robô), e assim, a identificação e localização dos obstáculos demandaria o uso de técnicas de processamento mais sofisticadas, com o uso de técnicas que após o reconhecimento do obstáculo, tomasse a decisão de parar o robô ou desviar para evitar colisões (PEREIRA *et al.*, 2014).

O problema da prevenção de colisão que se põe é o de não apenas detectar uma “travessia de fronteira” e não apenas tomar a decisão de desligar o robô, mas considerando o paradigma da interação máquina-máquina, permitir que mais de um agente de automação possa estar na cena, coexistindo em modo harmônico. Em outras palavras, enquanto a colisão não for iminente, que a tarefa de automação continue ou, seja, em último caso, readequada.

Deve-se ressaltar que, na literatura, o paradigma da interação máquina-máquina costuma vir apresentado como robótica cooperativa, em que os agentes de automação tem uma tarefa em comum (HABIB, 2014), ou, ainda, no contexto da robótica assistiva, em que robôs interagem para se antecipar e realizar tarefas de auxílio direto ao homem, tais como abrir portas, pegar objetos distantes, etc (KOPPULA; SAXENA, 2016) (a bem da verdade, esse último teria de certa forma uma interação humano-máquina).

Merecem destaque, também, as situações de interações entre robôs em que as máquinas figuram como adversárias umas das outras (PINTO *et al.*, 2016) ou, simplesmente, seus interesses sejam conflitantes, tais quais aquelas em que os espaços de trabalho se superponham (GAYLE *et al.*, 2007). É exatamente nesse cenário que o presente trabalho encontra lugar: traz-se para discussão um processo de produção automatizado em que robôs móveis e robôs manipuladores sejam agentes protagonistas de tarefas conflitantes, avaliando como a interação máquina-máquina pode apresentar perspectivas rumo a um processo mais produtivo e/ou mais eficiente (MOELLMANN *et al.*, 2006).

Do ponto de vista técnico-científico, conjugar as necessidades de uma produção industrial sem paradas com as potencialidades de um sistema robótico inteligente requer a implementação de um algoritmo que detecte um obstáculo em movimento e, antevendo colisão,

evite-a. Para este fim, existem duas abordagens tradicionais para a geração de caminhos livres de colisão, são elas: os algoritmos que utilizam Campos Potenciais Artificiais - CPA e aqueles baseados em Mapas de Rotas Probabilísticos (PRM - *Probabilistic Roadmap Methods*).

Embora sejam algoritmos clássicos, são de implementação desafiadora quando se trata de manipuladores industriais em tarefas de automação flexível, visto que, nesses casos, a pluralidade de movimentos de cada unidade articulada pode, especialmente se os robôs trabalham próximos uns dos outros, levar a colisões ou, no mínimo, a interrupções do processo. A ideia a ser explorada aqui é que o robô possa adaptar sua trajetória dinamicamente, desviando dos obstáculos da cena e reduzindo os eventos de parada, o que implica otimizar tempo e recursos, bem como aumentar a produtividade do sistema.

## 1.2 Utilização de campos potenciais artificiais em trajetórias livres de colisão

Campos Potenciais Artificiais tem sido bastante utilizados por diversos pesquisadores em várias aplicações para geração de trajetória livre de colisão. Dentre as aplicações podemos citar: Veículos Aéreos Não Tripulados (VANT), robôs móveis, robôs manipuladores industriais, robótica com aplicação na medicina, carros autônomos, dentre outras.

O método de CPA foi introduzido por Khatib em 1986, em sua pesquisa sobre campos potenciais, foram realizadas inicialmente em robôs manipuladores com um ponto alvo e obstáculos estáticos ou em movimento. Neste método, considera-se uma posição de um ponto que se move num campo de forças. O alvo fornece uma força atrativa e os obstáculos, forças repulsivas. Khatib (1986) descreve o método de campos potenciais artificiais como: “o manipulador se move em um campo de forças. A posição final é um polo atraente para o *end effector* e os obstáculos são superfícies repulsivas.”<sup>1</sup>

Apesar deste método ter sido inicialmente utilizado em robôs manipuladores, atualmente, a maioria das aplicações são na robótica móvel. O método de CPA deve o seu nome ao fato de o campo (vetorial) de forças  $F(x)$  ser derivado de um campo (escalar) potencial  $U(x)$  (SILVA *et al.*, 2014).

Os trabalhos apresentados por Budiyanto *et al.* (2015) e Mac *et al.* (2016) destacam a utilização de CPA para geração de trajetórias e desvio de obstáculos em VANTS. Estes trabalhos utilizam o método de campo potencial, pois é particularmente atraente, uma vez que

---

<sup>1</sup> “The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end effector and obstacles are repulsive surfaces.”

tem uma estrutura simples, baixa complexidade computacional e fácil de implementar, além de ser utilizado na navegação autônoma em ambientes conhecidos ou não. Budiyanto *et al.* (2015) citam alguns métodos propostos para resolver problemas de desvio de obstáculos, tais como sistema de visão computacional, radar, Sistema de Posicionamento Global (GPS), digitalização de laser, entre outros, porém utilizou CPA, pois apresenta simplicidade para implementação e análise matemática. Ele utilizou CPA em um quadrirrotor para evitar colisões, visto que o VANT opera próximo da superfície da terra ou voa em baixa altitude.

Li *et al.* (2015) e Jorge *et al.* (2015) propõem estratégias onde o método de CPA é utilizado com outros métodos para resolver o problema de mínimos locais em navegação de robôs móveis em ambientes de exploração. Um método usa o CPA melhorado com base em um campo simultâneo, onde as funções potenciais são utilizadas para calcular o caminho otimizado e encurtar a distância do percurso planejado. Uma outra abordagem é usar o campo vetorial calculado a partir da solução numérica definida como Problema do Valor Limite (PVL) que gera um caminho suave e livre de mínimos locais.

Cheng *et al.* (2015) propõem o planejamento de trajetória livre de colisão de um veículo não tripulado no ambiente subaquático. Neste trabalho, um novo algoritmo de planejamento de trajetória é proposto pela combinação do algoritmo de síntese de velocidade e o método de CPA. Primeiramente, o método de campo potencial é utilizado para evitar colisão e em seguida, de acordo com as características de navegação do veículo não tripulado, o algoritmo de síntese de velocidade é utilizado para definir o melhor caminho.

Wang *et al.* (2015) apresentam um trabalho onde combina o método de campo potencial com o método de mapa de grade (*Grid Map Method*) no planejamento de trajetória livre de colisão de um robô móvel. O método reduz a quantidade de cálculos e também evita a um grave problema de mínimo local que é decorrente do campo potencial artificial. Essa combinação traz como resultado uma otimização do método de CPA, confiabilidade, segurança e eficiência na geração de trajetória.

Chatraei e Javidian (2015) apresentam uma pesquisa onde um robô móvel desvia de obstáculos estáticos e dinâmicos utilizando CPA e um sistema *Fuzzy Mandani*. O método de CPA é utilizado para realizar o desvio de obstáculos e o sistema *Fuzzy* para controlar a posição e a orientação do robô até o ponto desejado.

Galceran *et al.* (2015) implementaram CPA em um carro autônomo. O veículo utilizado foi um automóvel Ford Fusion equipado com sensores capazes de detectar obstáculos e

guiar o veículo em um caminho específico. O CPA foi utilizado para guiar o veículo em direção a seu caminho e desviar de obstáculos estáticos e outros veículos participantes no tráfego.

Hargas *et al.* (2015) aplicaram CPA para gerar um caminho livre de colisões para um manipulador móvel desviar de obstáculos estáticos em um ambiente conhecido. Um manipulador móvel é um sistema mecânico articulado montado sobre uma base móvel não holonômica. Essa aplicação apresenta novos desafios pois a dificuldade de geração de trajetória aumenta com a quantidade de graus de liberdade (GDL) do manipulador. Além disso, a utilização de manipuladores com muitos GDL envolve outros problemas, tais como modelar seu ambiente de trabalho (geralmente desestruturado, desconhecido ou parcialmente conhecido).

Outro trabalho, de simulação, apresentado por Kazemi *et al.* (2015) é a aplicação de CPA em um robô planar paralelo para evitar colisão entre suas partes mecânicas (links e a estrutura do robô) e obstáculos em seu espaço de trabalho. Neste trabalho é utilizado a combinação de CPA com Lógica *Fuzzy* para o desenvolvimento do algoritmo de desvio de colisões.

Yu *et al.* (2015a) e Yu *et al.* (2015b) apresentam trabalhos onde utiliza o método de CPA em um robô assistente de cirurgia de implante dentário. No primeiro trabalho o CPA é aplicado em um manipulador de cinco GDL onde o robô manipulador é adaptado para encontrar a posição e ajuste do ângulo. No segundo trabalho, foi utilizado um sistema de navegação baseado em visão estéreo implementado em um robô de três GDL. Os trabalhos se propõem a três requisitos, que são: 1) o usuário pode liberar ou mover o manipulador quando estiver distante do destino; 2) o manipulador irá aplicar a força de ajuste para convergir para o destino; 3) o manipulador vai aplicar a força necessária para evitar colisões ao passar do ponto de destino.

Muitos trabalhos com CPA também foram desenvolvidos em robôs manipuladores. Guan *et al.* (2015) apresentam um trabalho onde utiliza CPA em um manipulador de seis GDL para desviar de obstáculos dinâmicos em tempo real. Nesse trabalho foi utilizado esferas como obstáculos para escapar de pontos de mínimo local.

Ataka *et al.* (2016) e Badawy (2014) utilizaram CPA em manipuladores para geração de trajetória e desvio de colisão em obstáculos estáticos e dinâmicos em tempo real. O primeiro autor testou o algoritmo de planejamento e validou através de simulação em tempo real em um manipulador de seis GDL. Pinto *et al.* (2014) também utilizaram CPA em trabalhos de simulação para geração de trajetória. Neste trabalho, o campo potencial foi gerado com carga variável e os campos atraentes foram adaptados, para permitir que o robô atingisse o ponto de destino com

maior facilidade, em um ambiente dinâmico e complexo onde uma abordagem original não seria capaz de resolver. O algoritmo foi validado usando o simulador MobileSim® juntamente com Matlab®. Os resultados ilustram a eficiência do algoritmo em aplicações com mínimos locais múltiplos, como obstáculos com passagens estreitas. Este método de controle pode ser usada em aplicações industriais, tais como a operação de empilhadeira automática, manipuladores robóticos e robôs móveis.

Silva *et al.* (2014) e Pereira *et al.* (2014) utilizaram CPA em um manipulador robótico SCARA (*Selective Compliant Assembly Robot Arm*) com dois GDL. Nestes trabalhos, o CPA foi utilizado para geração de trajetória e desvio de obstáculos estáticos. O segundo utilizou um sensor de imagens para identificação dos obstáculos na cena e a partir das imagens realizar o planejamento de caminho para o manipulador, sem que haja colisão da sua estrutura com os obstáculos.

### **1.3 Motivação e problemática**

Conforme destacado nos trabalhos supracitados, as pesquisas envolvendo CPA parecem estar limitadas a simulações apenas (vide Guan *et al.* (2015); Ataka *et al.* (2016); Badawy (2014); Pinto *et al.* (2014)) e em muitos casos, a ambientes contendo somente obstáculos estáticos. Há, portanto, uma lacuna relevante, qual seja, aquela de avaliar o uso do algoritmo CPA em manipuladores industriais, especialmente no caso em que o espaço de trabalho do robô contenha obstáculos arbitrariamente posicionados e em movimento. Isso permitiria, por exemplo, expandir os cenários de aplicações para processos de manufatura de uma indústria flexível e inteligente. A pergunta de partida seria: qual a dificuldade de realizar o planejamento de caminho por CPA em um manipulador robótico em ambientes dinâmicos?

As pesquisas que envolvem CPA, são na maioria aplicados em robôs móveis, ou são aplicações nas quais o objeto tratado pode ser considerado um ponto, isso facilita a implementação do CPA. Já em manipuladores não podemos considerar apenas o *end-effector* e sim toda a cadeia de elos do manipulador, o mesmo deve ser tratado como um corpo extenso, isso torna a implementação mais complexa pois o manipulador precisa ser discretizado em vários pontos. Neste caso, a implementação se torna mais onerosa, além de ser necessário um ambiente monitorado por sensores que detectem os obstáculos e execute o processamento do algoritmo em tempo real para evitar a colisão entre robôs.

Alinhado com o contexto da produção industrial, um segundo questionamento

poderia estar relacionado à eficiência ou à produtividade desse manipulador, entendido como máquina, numa tarefa industrial típica: o algoritmo de planejamento por CPA é limitante da produtividade ou da eficiência em tarefas *pick-and-place*? Em que medida?

Sabendo que essa técnica de planejamento de caminho sofre do aparecimento de mínimos locais, busca-se estudar o desempenho da tarefa quando o processo é ensaiado em diferentes abordagens, de modo a se conhecer as condições de implementação do algoritmo que melhor combinam custo e benefício.

Pretende-se, portanto, com o estudo aqui realizado, mostrar dentro do contexto da produção industrial o efeito das restrições impostas na esfera da automação, quando há interação não-colaborativa entre máquinas.

## **1.4 Objetivos**

A seguir são apresentados os objetivos geral e específicos do trabalho.

### ***1.4.1 Objetivo geral***

O objetivo geral deste trabalho é a aplicação e implementação do algoritmo de campos potenciais artificiais na geração de caminho livre de colisão, em tempo real, de obstáculo dinâmico em um manipulador robótico SCARA, onde ocorre a interação não-colaborativa máquina-máquina.

### ***1.4.2 Objetivos específicos***

Os objetivos específicos desta dissertação são enumerados a seguir:

1. Desenvolver um sistema de automação para tarefas de manipulação livres de colisão;
2. Desenvolver um algoritmo de campo potencial artificial para um manipulador SCARA e avaliá-lo em cena com obstáculo móvel;
3. Estudar a interação máquina-máquina do ponto de vista da eficiência de uma tarefa *pick-and-place*;
4. Propor uma figura de mérito para a produtividade do manipulador e comparar com as utilizadas na literatura de produção industrial;
5. Analisar a eficiência da tarefa de *pick-and-place* à luz dos esforços dos motores

do manipulador e o consumo de energia mecânica.

### **1.5 Produção científica**

Os resultados parciais deste trabalho foram reunidos nos artigos a seguir.

*Artigo publicado na Revista Proceeding Series of the Brazilian Society of Computational and Applied, 2016*

BATISTA, J. G. et al. Modelagem Matemática e simulação Computacional da Dinâmica de um Robô SCARA. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 4, n. 1, 2016.

*Artigo publicado no SBAI/DINCON, 2015*

BATISTA, J. G. et al. Modelagem Matemática e Simulação Computacional da Dinâmica de um Robô SCARA. DINCON – Conferência Brasileira de Dinâmica, Controle e Aplicações, 2015.

*Artigo publicado no XX CBA, 2014*

SILVA, J. L. N.; BATISTA, J. G. et al. Proposta de Planejamento e Controle de Trajetória para Manipuladores Industriais. Anais do XX Congresso Brasileiro de Automática, v. 20,n. 20, p.39543961, 2014.

### **1.6 Organização geral da dissertação**

Esta dissertação está organizada em seis capítulos, os quais apresentam uma sequência que mostra passo a passo o estudo e desenvolvimento do trabalho.

No capítulo 2 é apresentada a modelagem matemática do manipulador SCARA, objeto principal desse trabalho, é descrito as equações que representam a cinemática do manipulador.

No capítulo 3 é feita a fundamentação teórica descrevendo os temas principais relacionados a esse trabalho: geração de caminho, tipos de trajetórias, planejamento de trajetória utilizando campos potenciais artificiais, geração de caminhos utilizando campos potenciais

artificiais e controle de trajetória.

No capítulo 4 é realizada a descrição do experimento, mostrando os materiais utilizados, os métodos e ensaios preliminares para a implementação do algoritmo de campos potenciais no planejamento de caminhos livres de colisão no manipulador.

No capítulo 5 são apresentados os resultados e as discussões técnicas sobre a utilização de campos potenciais artificiais no planejamento de caminho aplicado no manipulador.

Por fim, no capítulo 6, são feitas as considerações finais sobre o trabalho, citando as contribuições, as conclusões do trabalho e as perspectivas para trabalhos futuros.

## 2 MODELAGEM MATEMÁTICA DO MANIPULADOR

Este capítulo apresenta a modelagem matemática do manipulador SCARA utilizado neste trabalho e do espaço de trabalho. Aqui será realizada somente a modelagem da cinemática do manipulador.

### 2.1 Introdução

O modelo cinemático têm como objetivo fundamental estabelecer as estratégias adequadas de controle do robô, que resultem em uma melhor qualidade de seus movimentos. Tendo em conta a variedade e complexidade das estruturas que compõem os robôs manipuladores, a cinemática descreve o movimento relativo dos vários sistemas de referência à medida que a estrutura se movimenta, associando sistemas de referência às várias partes da estrutura.

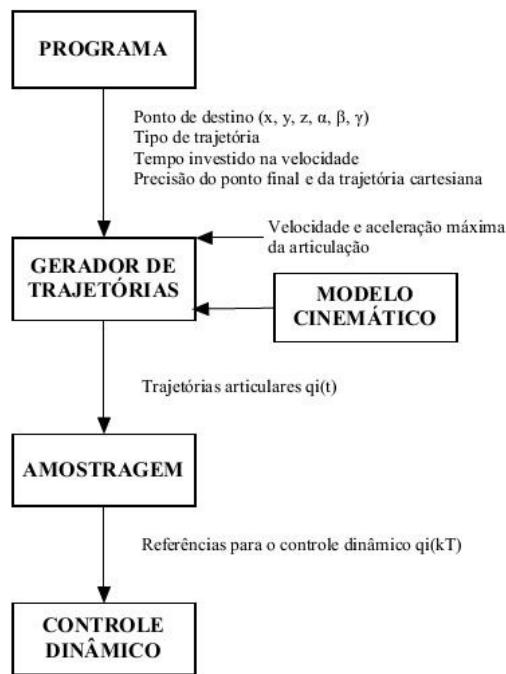
O controle cinemático estabelece quais são as trajetórias que devem ser seguidas por cada articulação do robô, num intervalo de tempo, para atingir os objetivos fixados pelo usuário (ponto de destino, trajetória cartesiana da ferramenta do robô, tempo gasto pelo usuário, etc.). Estas trajetórias são selecionadas atendendo às restrições físicas próprias dos acionamentos e a certos critérios de qualidade da trajetória, como suavidade ou precisão da mesma.

O movimento relativo a uma tarefa a ser executado pelo robô deve ser representado como um sinal de referência, o qual é enviado ao sistema de controle. O sistema de controle age no sentido de fazer com que o robô acompanhe o sinal de referência ativando o atuador adequado. Se o sinal de referência muda muito rapidamente, o acompanhamento do sinal de referência poderá ser pobre, em vista do projeto do sistema de controle e pelas limitações dinâmicas do sistema. Algoritmos computacionais são projetados para calcular o sinal de referência apropriado baseado na trajetória da tarefa a ser executada e nos limites de tempo relacionados (tais como velocidade e aceleração). Este sinal de referência corresponde à trajetória e pode ser definido como o local de pontos no espaço operacional (espaço da tarefa, relativo ao elemento terminal), ou no espaço das articulações, na qual uma lei temporal é especificada (SCIAVICCO; SICILIANO, 1996).

A figura 3 a seguir, apresenta um fluxograma de como é o funcionamento do controle cinemático. O sistema recebe como entrada os dados do programa do robô escrito pelo usuário (ponto de destino, precisão, tipo de trajetória desejada, velocidade, etc) e com base no modelo cinemático do robô estabelece as trajetórias para cada articulação em função do tempo.

Essas trajetórias devem ser apresentadas com um período de tempo  $T$  escolhido, gerando em cada instante  $K_T$ , um vetor de referências articulares para os algoritmos de controle dinâmico (ANTONIO *et al.*, 1997).

Figura 3 – Funcionamento do controle cinemático



Fonte: Barrientos - adaptado (1997).

De maneira geral, o controle cinemático deverá realizar as seguintes funções (ANTONIO *et al.*, 1997):

1. Converter as características do movimento de um programa em uma trajetória analítica no espaço cartesiano.
2. Mostrar a trajetória cartesiana para obter um número finito de pontos dessa trajetória. Cada um desses pontos será dado por uma 6-upla  $(x, y, z, \alpha, \beta, \gamma)$ .
3. Com o uso da cinemática inversa, converter cada um dos pontos em suas correspondentes coordenadas articulares  $(q_1, q_2, q_3, q_4, q_5, q_6)$ . Deve-se saber a quantidade de soluções da cinemática inversa, assim como a possibilidade de ausência de soluções e pontos singulares, de modo que a continuidade da trajetória seja garantida.
4. Interpolação dos pontos articulares obtidos, gerando para cada variável articular uma expressão  $q_i(t)$  que passe ou se aproxime das juntas de modo que, uma trajetória realizada pelas juntas, se transforme em uma trajetória cartesiana que se aproxime da trajetória especificada pelo programa do usuário (quanto a precisão, velocidade, etc.).

5. Apresentar a trajetória articular para gerar as referências do controle dinâmico.

## 2.2 Cinemática de manipuladores

A cinemática de um robô estuda o movimento do mesmo em relação a um sistema de referência. Portanto, a cinemática trata da descrição analítica do movimento espacial do robô como uma função do tempo, e em particular das relações entre a posição e a orientação da ferramenta do robô com os valores que tornam suas coordenadas articulares. O problema da cinemática direta consiste em determinar qual a posição e a orientação do atuador do manipulador, em relação a um sistema de coordenadas de referência fixo, conhecido os valores das articulações; o problema da cinemática inversa resolve a configuração que deve adotar o robô para uma posição e orientação do extremo conhecido (ROMANO, 2002).

### 2.2.1 Cinemática direta

Os conceitos fundamentais da cinemática de manipuladores com elos em série e como relacionar os sistemas de coordenadas das juntas foram introduzidos em 1955 por Jacques Denavit e Richard Hartenberg em Denavit (1955) e depois em seu artigo clássico “Kinematic Synthesis of Linkages”. Hartenberg e Denavit (1964) apresentaram as bases para seu desenvolvimento em computadores digitais e até hoje denominada de convenção de Denavit-Hartenberg (SILVA, 2016).

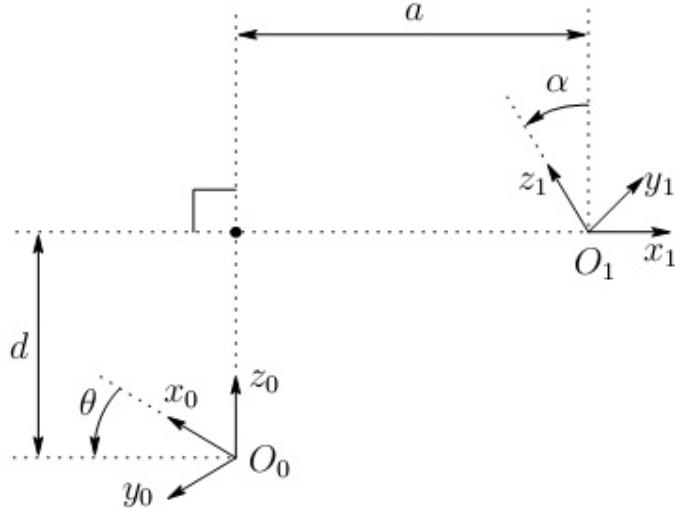
Para realizar a convenção Denavit-Hartenberg (D-H) cada matriz de transformação homogênea dos sistemas de coordenadas é composta por quatro transformações básicas (duas rotações e duas translações) mostrados em 2.1. Desta forma, mediante a escolha adequada dos sistemas de coordenadas, de forma que certos cancelamentos ocorram, são necessários somente quatro parâmetros ( $\alpha$ ,  $a$ ,  $d$ ,  $\theta$ ) para descrever a cinemática de cada elo do manipulador (PAUL, 1981) e (CRAIG, 2012).

$$A = \text{Rot}_{X,\alpha} \text{Trans}_{X,a} \text{Trans}_{Z,d} \text{Rot}_{Z,\theta} \quad (2.1)$$

O resultado da expressão tem como resultado uma matriz de transformação homogênea que relaciona dois sistemas de coordenadas. As transformações básicas são rotação em torno do eixo X com valor de  $\alpha$ , translação ao longo do eixo X de módulo  $a$ , translação ao longo do

eixo Z de módulo  $d$  e rotação em torno do eixo Z com valor de  $\theta$ . Essas transformações podem ser visualizadas na figura 4.

Figura 4 – Transformações da conversão D-H



Fonte: Spong (2006).

Para desenvolver a cinemática direta do manipulador SCARA foi adotada a convenção de Denavit-Hartenberg na forma modificada como apresentada em (SILVA, 2016), mostrado a seguir em 2.2 e 2.3.

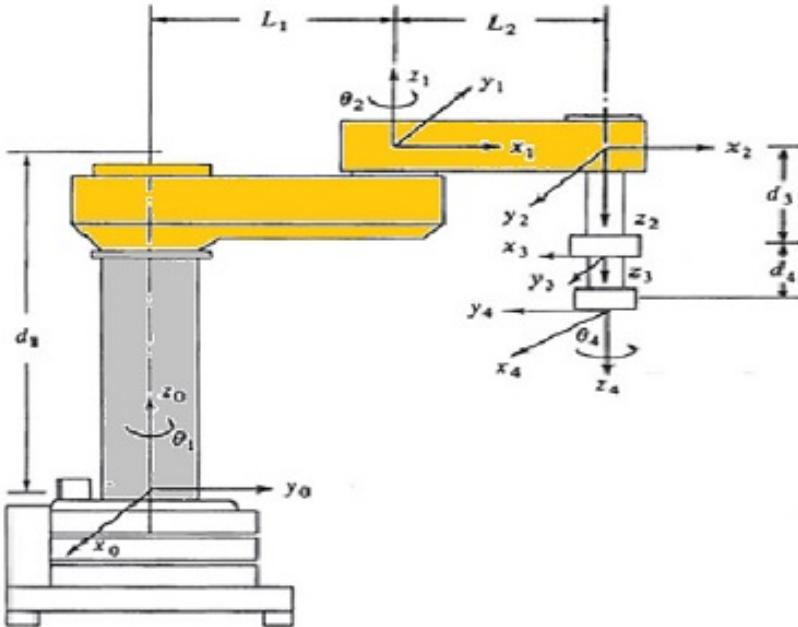
$$A = \text{Rot}_{Z,\theta} \text{Trans}_{Z,d} \text{Trans}_{X,a} \text{Rot}_{X,\alpha} \quad (2.2)$$

$${}_{n-1}^n A = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

De acordo com a representação de Denavit-Hartenberg (D-H), são estabelecidos os sistemas de coordenadas nas juntas do robô SCARA, conforme a figura 5. As equações da cinemática direta levam em consideração que as juntas 1,2 e 4 são rotacionais e a junta 3 prismática.

Os parâmetros do manipulador foram selecionados de acordo com os critérios definidos por D-H, assim como, os sistemas de coordenadas que foram atrelados às juntas como mostrado na figura 5. Na tabela 1 são mostrados os parâmetros de cada junta.

Figura 5 – Sistemas de coordenadas das juntas do robô SCARA



Fonte: Silva (2016).

Tabela 1 – Parâmetros D-H do Manipulador SCARA

Número do eixo	Ângulo de junta $\theta_i$	Offset do elo $d_i$	Comprimento do elo $a_i$	Ângulo de rotação $\alpha_i$
1	$\theta_1$	$d_1 = 0,32$	$L_1 = 0,35$	0
2	$\theta_2$	0	$L_2 = 0,30$	$\pi$
3	0	$d_3$	0	0
4	$\theta_4$	$d_4$	0	0

Fonte: Silva (2016).

A equação matricial 2.3, deve ser interpretada como uma representação geométrica da orientação e posição do manipulador, as três primeiras colunas representam a orientação e a ultima a posição, como mostrada na equação 2.4.

$$A = \begin{bmatrix} n_x & O_x & a_x & P_x \\ n_y & O_y & a_y & P_y \\ n_z & O_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

A partir dos parâmetros D-H podemos obter as matrizes de transformação homogê-

neas que são:

$${}^1_0A = \begin{bmatrix} C_1 & -S_1 & 0 & l_1C_1 \\ S_1 & C_1 & 0 & l_1S_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$${}^2_1A = \begin{bmatrix} C_2 & S_2 & 0 & l_2C_2 \\ S_2 & -C_2 & 0 & l_2S_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$${}^3_2A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$${}^4_3A = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Onde:  $S_1 = \sin\theta_1$ ;  $C_1 = \cos\theta_1$ ;  $S_2 = \sin\theta_2$ ;  $C_2 = \cos\theta_2$ ;  $S_4 = \sin\theta_4$  e  $C_4 = \cos\theta_4$ .

Após calculada a matriz de transformação de cada junta será calculada a matriz de transformação homogênea que indica a localização do elemento final do robô, em relação ao sistema de referência da base.

$$T = {}^1_0A \cdot {}^2_1A \cdot {}^3_2A \cdot {}^4_3A = {}^4_0A \quad (2.9)$$

$${}^4_0A = T = \begin{bmatrix} S_4S_{12} + C_4C_{12} & S_4C_{12} + C_4S_{12} & 0 & l_1C_1 + l_2C_{12} \\ S_4C_{12} + C_4S_{12} & -S_4S_{12} + C_4C_{12} & 0 & l_1S_1 + l_2S_{12} \\ 0 & 0 & -1 & d_1 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Onde:  $S_{12} = \sin(\theta_1 + \theta_2)$  e  $C_{12} = \cos(\theta_1 + \theta_2)$ .

A partir da matriz de transformação homogênea 2.10 e comparando com sua forma geométrica 2.4, podemos determinar a posição do elemento final do robô que pode ser encontrada no espaço de tarefa a partir das coordenadas no espaço das juntas como observado nas equações 2.11, 2.12 e 2.13.

$$P_x = 0,35C_1 + 0,30C_{12} \quad (2.11)$$

$$P_y = 0,35S_1 + 0,30S_{12} \quad (2.12)$$

$$P_z = d_1 + 0,42 \quad (2.13)$$

As equações 2.11, 2.12 e 2.13 são a solução do problema da cinemática direta de posição para o manipulador SCARA em estudo. O modelo cinemático direto é fundamental nesse trabalho, pois a partir dele será construído o modelo da dinâmica e principalmente todas as trajetórias, tanto no espaço cartesiano quanto no espaço das juntas, que serão geradas a partir destas equações.

### 2.2.2 Cinemática inversa

A cinemática inversa resolve a configuração que deve adotar o robô para uma posição e orientação de um ponto conhecido. Uma maneira simples de resolver o problema da cinemática inversa do robô é usando o método geométrico. A partir da cinemática inversa, será possível determinar o valor do ângulo de cada junta, a fim de colocar o manipulador na posição e a partir de então gerar trajetórias no espaço das juntas (SILVA, 2016).

Partindo das equações da cinemática direta 2.11 e 2.12, elevando ambos os membros das duas expressões ao quadrado e somando-os tem-se:

$$P_x^2 + P_y^2 = l_1^2 C_1^2 + l_1^2 S_1^2 + l_2^2 S_{12}^2 + l_2^2 C_{12}^2 + 2l_1 l_2 C_1 C_{12} + 2l_1 l_2 S_1 S_{12} \quad (2.14)$$

Aplicando-se as identidades trigonométricas, obtém-se:

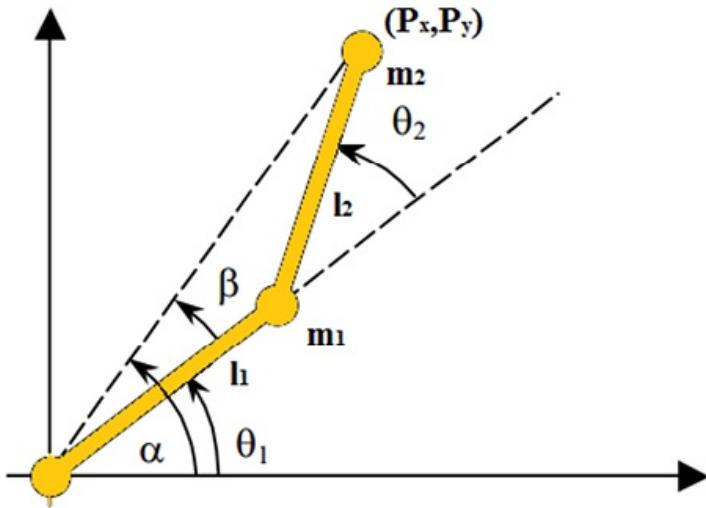
$$P_x^2 + P_y^2 = l_1^2 + l_2^2 + 2l_1 l_2 C_2 \quad (2.15)$$

Isolando  $\theta_2$  tem-se:

$$\theta_2 = \pm \cos^{-1} \left( \frac{P_x^2 + P_y^2 - l_1^2 - l_2^2}{2l_1l_2} \right) \quad (2.16)$$

Para determinar  $\theta_1$  utilizou-se a tangente trigonométrica da diferença angular (ver 2.17), como pode ser observado na figura 6.

Figura 6 – Tangente trigonométrica de uma diferença angular para determinar  $\theta_1$



Fonte: Silva (2016).

$$\tan(A - B) = \left( \frac{\tan A - \tan B}{1 + \tan A \tan B} \right) \quad (2.17)$$

A partir da figura 6, pode-se concluir que:  $\theta_1 = \alpha - \beta$ , assim:

$$\tan \beta = \left( \frac{l_2 S_2}{l_1 + l_2 C_2} \right) \quad (2.18)$$

$$\tan \alpha = \frac{P_y}{P_x} \quad (2.19)$$

Aplicando a identidade mostrada na equação 2.17 para  $\theta_1$ , tem-se:

$$\tan \theta_1 = \tan(\alpha - \beta) \quad (2.20)$$

Realizando as substituições adequadas,  $\theta_1$  será dado por:

$$\theta_1 = \tan^{-1} \left( \frac{P_y(l_1 + l_2 C_2) - P_x l_2 S_2}{P_x(l_1 + l_2 C_2) + P_y l_2 S_2} \right) \quad (2.21)$$

Para encontrar o parâmetro da junta três, será necessário isolar a variável  $d_3$ :

$$d_3 = 0,42 - P_z \quad (2.22)$$

As equações 2.16, 2.21 e 2.22 são as soluções do problema da cinemática inversa do manipulador SCARA e serão usadas para realizar o controle de posição e geração de caminho do manipulador. Vale ressaltar que para o manipulador SCARA, existem duas soluções da cinemática inversa para uma mesma posição no espaço cartesiano. Ou seja, uma posição no espaço cartesiano pode ser alcançada a partir de duas combinações das posições das juntas no espaço das juntas.

Este capítulo apresentou a modelagem matemática do manipulador SCARA, objeto deste trabalho. Foram apresentadas as soluções para a cinemática direta e inversa do manipulador. No capítulo 3 será apresentado os tipos de trajetórias, o controle de trajetória e a geração de caminho através de campos potenciais artificiais.

### 3 GERAÇÃO DE CAMINHO E TRAJETÓRIA

#### 3.1 Introdução

O problema de geração de trajetória consiste em gerar o histórico temporal das variáveis de junta para que o manipulador robótico realize um movimento desejado, esse movimento pode ser realizado tanto no espaço das juntas quanto no espaço cartesiano. Usualmente este movimento é especificado através dos pontos inicial e final, da forma geométrica do caminho que deve ser percorrido entre estes pontos e o tempo final do movimento (CRAIG, 2012).

Existem diferenças entre trajetória e caminho. Em uma trajetória existem as restrições de tempo, velocidade e aceleração, enquanto que caminho é uma descrição puramente espacial, sem relação com o tempo. Uma trajetória pode ser encarada como um caminho ao qual se impõe um perfil de velocidade.

Os métodos de geração de trajetória mais utilizados baseiam-se em calcular os parâmetros da curva tomada como caminho, de forma que sejam especificados os pontos iniciais e finais e o tempo final e tem-se como resultado uma função parametrizada no tempo. O controle de trajetória consiste em resolver o problema de rastreamento da trajetória de referência.

De acordo com Spong (2006), planejamento de trajetória é uma especificação completa da localização de todos os pontos em que o robô é identificado como uma configuração, e o conjunto de todas as configurações possíveis é chamado de espaço de configuração.

#### 3.2 Geração de trajetória

A geração de trajetória especifica um ponto no espaço de trabalho do manipulador que pode ser traduzido em condições adequadas para um ponto no espaço da articulação. O problema consiste em levar o manipulador para a posição especificada, independentemente da posição inicial e das variáveis do ambiente. Este problema está enquadrado no quadro mais geral da chamada navegação de robôs. O problema de navegação robô consiste em realizar, em uma única etapa, as seguintes tarefas (SCIAVICCO; SICILIANO, 1996):

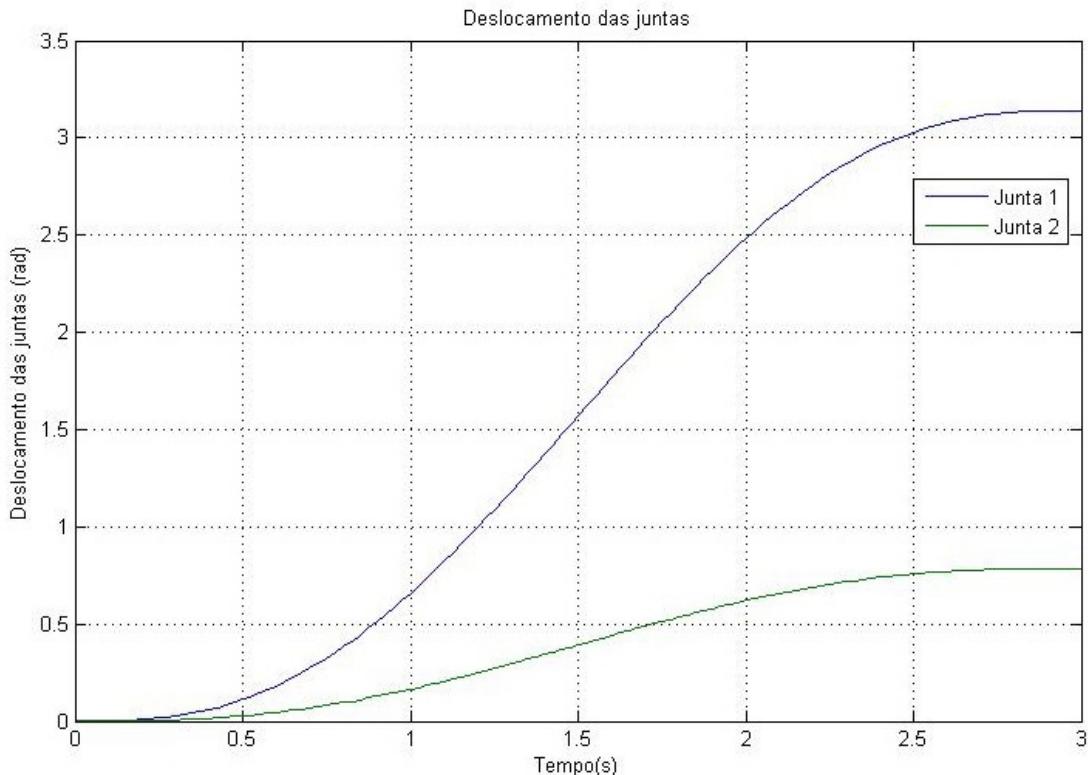
1. Planejamento de trajetória;
2. Geração de trajetória;
3. Controle de trajetória.

Planejamento de trajetória consiste em determinar uma curva no espaço de trabalho,

ligando a posição, desejada, inicial e final do atuador, evitando qualquer obstáculo. A união das posições no espaço cartesiano define dois tipos de perfis para deslocamentos lineares e circulares (KHATIB, 1986). Neste trabalho, será usada a técnica de campo potencial artificial para realizar o planejamento de caminho e trajetória. A trajetória parametrizada em tempo resultante, que é comumente chamado de trajetória de referência, é obtida principalmente em termos das coordenadas e perfis definidos no espaço de trabalho. Então, usando o método da cinemática inversa para obter uma trajetória parametrizada no tempo para as coordenadas no espaço das juntas (ROMANO, 2002).

Para geração de trajetórias é necessário à solução da cinemática inversa do robô e expressões polinomiais que representam a evolução no tempo da posição, da velocidade e aceleração das juntas do robô, isto é, a geração de incrementos angulares de juntas necessárias para que o robô realize uma determinada tarefa. A figura 7 representa o deslocamento das juntas de um robô de dois links durante uma trajetória.

Figura 7 – Deslocamento das juntas no tempo para execução de trajetória



Fonte: elaborada pelo autor.

As trajetórias podem ser especificadas nas coordenadas das juntas ou cartesianas. Em muitas aplicações a programação de tarefas de robôs é realizada no espaço das juntas (*joint space*), não necessitando de um modelo da cinemática do robô, apenas das funções polinomiais,

e a trajetória angular de mesma natureza dos sinais provenientes do transdutor de posição servirá como sinal de referência para o controlador de cada junta. Entretanto, na maioria das aplicações, a realização de tarefas está relacionada com o tipo de ferramenta utilizada a partir de um sistema de coordenadas cartesianas fixo à base do robô, designado espaço cartesiano ou espaço operacional (*operacional space*). Consequentemente os movimentos desejados e as leis de controle estão em espaços diferentes (BROOKS, 1983).

### **3.2.1 Geração de trajetórias no espaço Cartesiano**

A geração de trajetórias no espaço cartesiano possui muitas vantagens em relação à programação no espaço das juntas, cuja trajetória gerada entre dois pontos pode ser mais bem definida, como, por exemplo, quando a ferramenta do robô deve seguir um caminho específico, tal como uma reta ou um arco de círculo. Para a implementação de um algoritmo de geração de trajetórias no espaço cartesiano, é necessário o conhecimento da modelagem da cinemática do robô e também de métodos para a inversão do mesmo, para gerar os ângulos das juntas correspondentes a uma determinada configuração espacial (ANTONIO *et al.*, 1997).

Normalmente, o operador do robô indica o movimento que este deve realizar especificando as localizações espaciais pelas quais deve passar a extremidade, junto com outros dados, como instante da passagem, velocidades ou tipos de trajetória. Estando estes pontos excessivamente separados é necessário selecionar pontos intermediários próximos para que o controle do robô consiga atingir somente o ponto final especificado e cumprindo as restrições do programa como tipo de trajetória e velocidade.

Para isso é preciso estabelecer um interpolador entre as localizações expressadas no espaço de tarefa que resultará numa expressão analítica da evolução de cada coordenada. Entre as funções adotadas para realizar esta interpolação de pontos estão: função linear, função cúbica, função de ordem superior etc.

Para um robô de três graus de liberdade executar uma reta no espaço cartesiano, as variáveis cartesianas devem ser tais que:

$$\begin{aligned} y_t &= a_1x(t) + a_0 \\ z_t &= b_1x(t) + b_0 \end{aligned} \tag{3.1}$$

Supondo que  $x(t)$  tenha a forma canônica de um polinômio  $p_x(t)$ , tem-se que:

$$\begin{aligned} y_t &= a_1 p_x(t) + a_0 = p_y(t) \\ z_t &= b_1 p_x(t) + b_0 = p_z(t) \end{aligned} \quad (3.2)$$

Sendo  $p_y(t)$  e  $p_z(t)$  polinômios de mesma ordem que  $p_x(t)$ , apenas com coeficientes diferentes.

Cada ponto  $(y(t); z(t))$  deve ainda ser mapeado através do modelo cinemático inverso para o espaço das juntas, de forma a se obter os valores das variáveis de junta  $(q_1(t); q_2(t); q_3(t))$ .

É interessante notar que a forma de calcular os coeficientes para uma determinada trajetória canônica é exatamente a mesma, independentemente se é desejada uma trajetória no espaço das juntas ou no espaço cartesiano. As diferenças estão apenas na parametrização da trajetória em função do tempo, que determina a forma da trajetória e no mapeamento do espaço cartesiano para o espaço das juntas, se for o caso. Um caminho ou trajetória canônica é uma função da tarefa a ser desempenhada pelo robô, que pode ser uma reta, por exemplo. Neste caso, a trajetória terá a sua parametrização em função do tempo e merece especial atenção, pois se a trajetória especificada for diferente de uma reta no espaço cartesiano, as formas expressões das trajetórias para cada variável cartesiana serão diferentes (LAGES, 2005).

### **3.2.2 Geração de trajetórias no espaço das juntas**

Quando uma trajetória é gerada no espaço das juntas, a própria trajetória já é o histórico temporal das variáveis de junta, que pode ser utilizado como referência para os controladores das juntas.

Neste caso, escolhe-se como caminho canônico uma reta no espaço das juntas, e utiliza-se polinômios para parametrizar a trajetória no tempo. Sendo gerada no espaço das juntas, a trajetória resultante não apresenta problemas de singularidades. Por outro lado, a trajetória cartesiana resultante não é facilmente inferida a partir da trajetória das juntas, devido ao mapeamento não linear entre o espaço das juntas e o espaço cartesiano, realizado pelo modelo cinemático (CRAIG, 2012).

Como exemplo podemos citar um robô com três juntas, será definido a trajetória

canônica da seguinte forma:

$$\begin{aligned} q_1(t) &= p_1(t) \\ q_2(t) &= p_2(t) \\ q_3(t) &= p_3(t) \end{aligned} \tag{3.3}$$

Onde  $q_i(t)$  são os históricos temporais das variáveis de junta (trajetória no espaço das juntas) e  $p_i(t)$  são polinômios de mesma ordem com coeficientes a determinar em função dos pontos iniciais e finais e do tempo final da trajetória.

### 3.3 Tipos de trajetórias

A escolha da melhor forma de trajetória é uma decisão importante no planejamento para implementar o gerador de trajetórias. Esta trajetória escolhida, além de ser factível, deve ser computacionalmente executável (LAGES, 2005).

#### 3.3.1 *Trajetória degrau*

Trajetória do tipo degrau de posição tem características de um movimento ponto-a-ponto. Esta abordagem simplifica o problema de rastreamento da trajetória, que passa a ser um problema de set-point. No entanto, esta estratégia mostra-se desaconselhável, pois produz movimentos difíceis de serem previstos, visto que todas as juntas começam a se mover simultaneamente e param conforme vão atingindo suas posições de referência (LAGES, 2005).

Além disso, várias juntas movem-se com maior velocidade do que seria necessário caso fosse realizado um movimento coordenado (com todas as juntas partindo e parando simultaneamente), sofrendo um maior stress e produzindo perturbações maiores do que o necessário nas demais juntas. Como as trajetórias percorridas fisicamente pelo manipulador devem ser contínuas em posição e velocidade, tendo-se um degrau como trajetória de referência, haverá um grande erro de rastreamento nos instantes iniciais da trajetória, o que impossibilita a utilização de algumas estratégias avançadas de controle, que assumem um erro de rastreamento pequeno o suficiente para que se possa utilizar um modelo linearizado em torno de um ponto de operação (PAUL, 1981).

### ***3.3.2 Trajetórias com velocidade trapezoidal***

Neste tipo de trajetória o manipulador é acionado com aceleração máxima até atingir a velocidade máxima, move-se nesta velocidade e a seguir é desacelerado ao máximo até atingir o ponto final da trajetória, gerando um perfil trapezoidal para a velocidade. Para que este tipo de trajetória possa ser utilizado, a aceleração deve ser suavizada empregando-se algum tipo de interpolação nos pontos de descontinuidade da aceleração, o que complica a implementação e reduz o desempenho deste tipo de trajetória.

De modo a permitir o rastreamento da trajetória, o limite de aceleração máxima deve ser escolhido de forma conservativa, para que exista uma certa margem de torque para a atuação do controlador, distanciando ainda mais o desempenho do manipulador dos seus limites (ANDERSSON, 1989).

### ***3.3.3 Trajetórias de tempo mínimo***

Trajetórias de tempo mínimo usualmente utilizam ao máximo as potencialidades do manipulador. No entanto, estas trajetórias são bastante descontínuas em aceleração. Como estas trajetórias operam o manipulador sobre a sua curva de torque máximo x velocidade, não deixam margem de torque para a atuação do controlador (SHIN; MCKAY, 1985).

Normalmente, para a obtenção destas trajetórias é necessário a utilização de algoritmos de busca em grafos e algoritmos de otimização numérica, sendo portanto computacionalmente dispendiosa, impossibilitando a sua aplicação em sistemas que devem operar on-line, como pode ser observado pelos tempos de processamento obtidos. Estas trajetórias podem ser interessantes para sistemas que operam com tarefas repetitivas, pois a geração de trajetória pode ser feita off-line, e o tempo necessário para que seja calculada não é crítico (SHILLER; DUBOWSKY, 1991).

Trajetórias deste tipo não podem ser divididas em uma fase de planejamento e uma fase expansão da trajetória em tempo real. O planejamento deste tipo de trajetória requer a expansão da trajetória completa, o que pode levar a um desperdício computacional caso a trajetória deva ser modificada durante o movimento.

### 3.3.4 Trajetórias polinomiais

Polinômios são uma escolha conveniente para representação de trajetórias porque suas propriedades são bem conhecidas e suas derivadas podem ser calculadas facilmente e são também polinômios.

#### 3.3.4.1 Trajetórias cúbicas

Trajetórias lineares são problemáticas porque implicam em velocidade constante e consequentemente acelerações descontínuas nos instantes iniciais e finais. Adicionalmente, é comum a situação em que se deseja implementar uma trajetória completa a partir de trajetórias mais simples. Para estes casos, deseja-se ter uma trajetória canônica onde se possa impor tanto as posições iniciais e finais quanto as velocidades iniciais e finais. Como neste caso têm-se quatro restrições, torna-se necessário a utilização de um polinômio cúbico, para que se possa acomodar todas as restrições (CRAIG, 2012).

### 3.3.5 Trajetórias lineares

Caso se pretenda que uma das articulações  $q$  do robô passe sucessivamente pelos valores  $q^i$  nos instantes  $t^i$ . Uma primeira solução para este problema consiste em manter constante a velocidade do movimento entre dois valores sucessivos  $(q^{i-1}, q^i)$  da articulação (ANTONIO *et al.*, 1997). A trajetória entre os pontos  $q^{i-1}, q^i$ , será dado pela seguinte expressão:

$$q(t) = (q^i - q^{i-1}) \frac{t - t^{i-1}}{T} + q^{i-1}$$

$$t^{i-1} < t < t^i$$

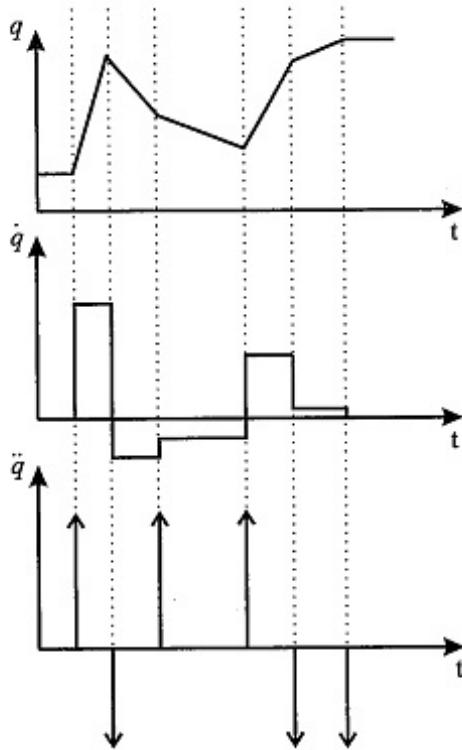
$$T = t^i - t^{i-1} \quad (3.4)$$

Esta expressão gera uma trajetória que assegura a continuidade da posição, no entanto, origina saltos bruscos na velocidade  $\dot{q}$ , da junta e consequentemente precisa de acelerações  $\ddot{q}$ , de valor infinito, o que na prática não é possível. A figura 8 a seguir, apresenta a trajetória, a velocidade e aceleração de uma interpolação linear.

Três critérios são importantes para seleção da interpolação linear:

1 - Cada articulação  $q$ , deve alcançar o ponto de destino no menor tempo possível, sem considerar as outras articulações, o que resultará em velocidades constantes e iguais a

Figura 8 – Posição, velocidade e aceleração em uma interpolação linear



Fonte: Barrientos (1997).

máxima.

2 - Ajustar os instantes de passagem pela articulação mais lenta, isto resultará em movimentos coordenados.

3 - Selecionar os tempos a partir das especificações dadas no espaço da tarefa de modo que a extremidade do robô descreva uma trajetória predeterminada.

### 3.4 Planejamento de trajetória utilizando campos potenciais artificiais

A utilização de campos potenciais artificiais consiste de um ponto inicial  $q_{inicial}$ , que gera uma força repulsiva, o alvo  $q_{final}$ , fornece uma força atrativa e os obstáculos, forças repulsivas.

O método de campos de potencial artificiais deve o seu nome ao fato de o campo (vetorial) de forças  $F(\theta)$  ser derivado do campo (escalar) potencial  $U(\theta)$ .

$$F(x) = \nabla[U(\theta)] \quad (3.5)$$

A função potencial  $U(\theta)$  é tipicamente definida no espaço livre como a combinação linear de um campo atrativo,  $U_{atr}(\theta)$ , que puxa o robô para a posição alvo, e de um potencial

repulsivo,  $U_{rep}(\theta)$ , que o empurra para longe dos obstáculos (LATOMBE, 1990). Assim, considerando a presença de um único obstáculo, define-se as coordenadas da posição do obstáculo como sendo  $\theta_{obs} = (\theta_{1obs}, \theta_{2obs})$ , a posição do alvo por  $\theta_{alvo} = (\theta_{1alvo}, \theta_{2alvo})$ , e a posição do centro do robô por  $\theta = (\theta_1, \theta_2)$  (no caso de um robô móvel) ou origem do sistema de coordenadas da ferramenta (no caso de um manipulador), a trajetória gerada é função do seguinte campo de potencial artificial:

$$U(\theta) = U_{atr}(\theta) + U_{rep}(\theta) \quad (3.6)$$

Onde  $U(\theta)$  é o potencial resultante do campo artificial,  $U_{atr}(\theta)$  é o potencial atrativo produzido pela posição do alvo,  $\theta_{alvo}$ , no centro do robô,  $\theta$ , e  $U_{rep}(\theta)$  é o potencial repulsivo induzido pelo obstáculo em  $\theta$ . O vetor que representa a força que deve ser aplicada no centro do robô móvel ou ferramenta de um manipulador é dado por:

$$F(\theta) = F_{atr}(\theta) + F_{rep}(\theta) \quad (3.7)$$

com

$$F_{atr}(\theta) = -\nabla[U_{atr}(\theta)] \quad (3.8)$$

e

$$F_{rep}(\theta) = -\nabla[U_{rep}(\theta)] \quad (3.9)$$

Onde  $F(\theta)$  é a força resultante,  $F_{atr}(\theta)$  é uma força atrativa que guia o centro do robô ao alvo, e  $F_{rep}(\theta)$  é uma força que induz uma repulsão artificial da superfície do obstáculo produzida por  $U_{rep}(\theta)$  (SCIAVICCO; SICILIANO, 1996). A geração de trajetória usando campos de potencial é um método que pode ser utilizado para planejamento global off-line, quando o ambiente do robô é conhecido à priori, ou em planejamento local on-line quando o ambiente é desconhecido e a presença dos obstáculos vai sendo detectada pelos sensores montados no robô (CHEN *et al.*, 2009).

Sob a influência do campo potencial artificial, o robô move-se na direção do simétrico do gradiente, de zonas de potencial mais elevado para a zona de menor potencial (na posição do alvo - mínimo global), onde o gradiente é nulo. No entanto, o campo potencial pode possuir outros mínimos onde o gradiente também é nulo. Assim, o robô pode ficar preso num mínimo local. Este é um problema comum para todas as técnicas de otimização que usam o gradiente de uma função (VOLPE; KHOSLA, 1990). Nos manipuladores, se o campo de repulsão for fixado

somente na origem do sistema de coordenada da junta, não garantirá que o robô evitará a colisão com o obstáculo. Portanto é necessária a definição dos chamados pontos de controle, que são definidos como pontos nos elos do robô que são sensíveis aos obstáculos. Um exemplo de ponto de controle é o centro de massa do elo (SPONG, 2006).

Há algumas aplicações deste método para ambientes móveis de geometria complexa, funcionando para controle de trajetórias de manipuladores robotizados (HARDEN, 1997). A técnica de campos potenciais artificiais, permite uma implementação em tempo real (on line), gerando um caminho contínuo e geralmente suave. Sua principal desvantagem está na aparição de mínimos locais que chegam a produzir uma parada no algoritmo empregado (LATOMBE, 1990). Adicionalmente, existe outro contra tempo: o número de pontos de controle. Se não houver números suficientes, existe o risco de colidir com obstáculo (HARDEN, 1997), (KROGH, 1984).

Devido o campo potencial, o robô move-se na direção do simétrico do gradiente, de zonas de potencial mais elevado para a zona de menor potencial onde o gradiente é nulo. No entanto, o campo potencial pode possuir outros mínimos locais. Este é um problema comum para todas as técnicas de otimização que usam o gradiente de uma função (SILVA *et al.*, 2014).

Uma ferramenta para tratar mínimos locais são as funções harmônicas que obedecem à equação de Laplace. Connolly *et al.* (1990) utilizaram a equação de Laplace onde as fronteiras dos obstáculos são condições de fronteira, daqui resulta uma função harmônica que não tem mínimos locais e que pode ser utilizada como função potencial para o planeamento da trajetória do robô.

### **3.5 Geração de caminho utilizando campos potenciais artificiais**

Como citado anteriormente, a utilização de campos potenciais consiste de um ponto inicial que gera uma força repulsiva, fornece uma força atrativa em relação ao ponto final e forças repulsivas aos obstáculos. O planejamento do caminho pode ser tratado como um problema de otimização, isto é, encontrar o mínimo global em U, a partir da configuração inicial  $\theta_{inicial}$ . Um dos algoritmos mais fáceis para resolver este problema é através do gradiente. Neste caso, o gradiente negativo de U pode ser considerado como uma força que atua sobre o robô (no espaço de configuração) (SPONG, 2006).

De maneira geral, o campo U é um campo resultante de uma adição onde uma parcela constitui a atração e a outra parcela constitui a repulsão, como mostrado na equação 3.6:

A força resultante dos campos de potencial artificiais que deve ser aplicada no robô é dada conforme a equação 3.7.

### 3.5.1 Campo potencial de atração

Existem vários critérios que o campo potencial,  $U_{atr}$  deve satisfazer. Primeiro,  $U_{atr}$  deve ser uniformemente crescente com a distância de  $q_{final}$ . A escolha mais simples é um campo que cresce linearmente com a distância de  $q_{final}$ , assim chamado potencial cônico. No entanto, o gradiente desse campo pode assumir um determinado valor em qualquer posição, mas no ponto final ( $q_{final}$ ) deve ser zero.

Isto pode levar a problemas de estabilidade, uma vez que existe uma descontinuidade na força de atração na origem. Isto acontece devido a atuação da força de repulsão na origem, a força de repulsão dos obstáculos e a força de atração do ponto final, dificultando a geração do caminho suave e causando o aparecimento de mínimos locais. O mais adequado é empregar um campo que é continuamente diferenciável, de modo que a força atrativa diminui à medida que o robô se aproxima de  $q_{final}$ . O campo mais simples é um campo que cresce quadraticamente com a distância até  $q_{final}$  (SPONG, 2006).

O problema de estabilidade que o campo potencial pode apresentar está relacionado a não geração de um caminho suave e contínuo, livre de colisão e podendo causar mínimos locais. Não é um problema de estabilidade numérica.

O campo potencial de atração pode ser utilizado de duas formas. A primeira é chamada de cônica, sendo apresentada representada pela seguinte equação:

$$U_{atr}(\theta) = K_a \parallel \theta - \theta_f \parallel \quad (3.10)$$

Já a segunda forma, conhecida como paraboloide, é escrita como

$$U_{atr}(\theta) = \frac{1}{2}K_a \parallel \theta - \theta_f \parallel^2 \quad (3.11)$$

onde,  $K_a$  é constante de atração,  $\theta$  é a configuração atual do robô e  $\theta_f$  é a configuração final desejada.  $\parallel \theta - \theta_f \parallel$  é a norma (comprimento do vetor a partir da origem), ou seja, a distância Euclidiana entre  $\theta$  e  $\theta_f$ .

Para resolver o problema de estabilidade é utilizada a combinação das vantagens das duas possibilidades de forma de campo de potencial, definidas pelas equações 3.10 e 3.11. Isto é

feito definindo uma distância de referência, na qual para distâncias superiores a esta referência o potencial é da forma cônica e para distâncias inferiores a distância de referência o potencial é da forma paraboloide, conforme mostra as seguintes equações (SPONG, 2006):

$$U_{atr1}(\theta) = dK_a \|\theta - \theta_f\| - \frac{1}{2}d^2K_a : \|\theta - \theta_f\| > d \quad (3.12)$$

$$U_{atr2}(\theta) = \frac{1}{2}K_a \|\theta - \theta_f\|^2 : \|\theta - \theta_f\| \leq d \quad (3.13)$$

Ou seja, para distâncias maiores que a distância de referência, 'd', a função do campo de atração toma a forma de 3.12 e para distâncias menores, ou iguais, que a distância de referência, a função do campo de atração toma a forma de 3.13. Consequentemente, a força de atração resultante é apresentada por:

$$F_{atr1}(\theta) = -\nabla U_{atr1}(\theta) = -dK_a \left( \frac{\theta - \theta_f}{\|\theta - \theta_f\|} \right) : \|\theta - \theta_f\| > d \quad (3.14)$$

$$F_{atr2}(\theta) = -\nabla U_{atr2}(\theta) = -K_a(\theta - \theta_f) : \|\theta - \theta_f\| \leq d \quad (3.15)$$

Onde a força de atração assume a forma de 3.14 para distâncias maiores que a de referência e assume a forma de 3.15 para distâncias menores, ou iguais, que a de referência.

### 3.5.2 Campo potencial de repulsão

Para implementação do campo repulsivo devemos observar alguns critérios. O campo potencial de repulsão deve repelir o robô de obstáculos, nunca permitindo que o robô venha a colidir com um, e quando o robô estiver longe de um obstáculo, deve exercer pouca ou nenhuma influência sobre o movimento do robô. Uma maneira de conseguir isso é definir um potencial que vai para o infinito em limites de obstáculos, e cai para zero a uma certa distância do obstáculo. Se definimos  $p(\theta)$  como a distância de influência de um obstáculo (isto é, um obstáculo não repelirá o robô se a distância do robô ao obstáculo for maior que  $p(\theta)$ ), um potencial que atende a esses critérios é dado por (SPONG, 2006):

$$U_{rep1}(\theta) = \frac{1}{2}K_r \left[ \frac{1}{p(\theta)} - \frac{1}{p} \right]^2 : p(\theta) \leq p \quad (3.16)$$

$$U_{rep2}(\theta) = 0 : p(\theta) > p \quad (3.17)$$

onde  $K_r$  é a constante de repulsão,  $p(\theta)$  é a menor distância entre  $\theta$  e todo o espaço de configurações de obstáculos ( $QO$ ) de um obstáculo específico e  $p$  é a distância de referência.

Caso a configuração do robô se encontre numa distância menor ou igual que a referência, o campo de repulsão assume a forma de 3.16. Caso contrário, os obstáculos não influenciem no caminho descrito pelo robô, o campo de repulsão assume a forma de 3.17. Dessa forma os obstáculos só passam a influir no planejamento do caminho do robô a partir do momento que o robô entra numa região próxima do obstáculo, definida pela distância de referência. Com isso, a força de repulsão será dada por

$$F_{rep1}(\theta) = \nabla U_{rep1}(\theta) = k_r \left( \frac{1}{p(\theta)} - \frac{1}{p} \right) \frac{1}{p^2(\theta)} \nabla p(\theta) : p(\theta) \leq p \quad (3.18)$$

$$F_{rep2}(\theta) = \nabla U_{rep2}(\theta) = 0 : p(\theta) > p \quad (3.19)$$

onde a força de repulsão assume a forma de 3.18 para distâncias menores, ou iguais, que a de referência e assume a forma de 3.19 para distâncias maiores que a referência.

### 3.6 Controle de trajetória

O controle de trajetória é fundamental em aplicações industriais para garantir a produção, qualidade dos produtos e segurança dos processos. Controle de servo-válvulas, servo-motores, máquinas-ferramenta, robôs industriais (manipuladores) e células de manufatura integrada são exemplos dessa importância. Os manipuladores industriais são as máquinas em sistemas de automação flexível que mais dependem do controle de posição, vários modelos de manipuladores calculam suas variáveis internas como velocidades e acelerações baseadas no controle de posição. Os métodos de controle de trajetórias utilizados em robôs industriais podem ser divididos segundo duas classificações principais: Controle Ponto a Ponto (CPP) e Controle por Trajetória Contínua (CTC) (ISO, 1998).

Quando a geração da trajetória é concluída as coordenadas estão no espaço de trabalho (espaço cartesiano). Existem posições na trajetória gerada que não são facilmente

convertidos para os pontos no espaço das juntas  $q$ , isto devido ao mapeamento não linear entre o espaço das juntas e o espaço cartesiano, realizado pelo modelo cinemático direto. Devido a isso, o mapeamento é feito ponto a ponto utilizando-se o modelo cinemático inverso (SCIAVICCO; SICILIANO, 1996):

$$q = T(q)^{-1}X \quad (3.20)$$

Ou seja, para cada ponto da trajetória cartesiana são computados, através do modelo cinemático inverso, os valores das variáveis de junta correspondentes, que podem então ser utilizados como referência para os controladores de junta. No entanto, devido ao mapeamento não linear realizado pelo modelo cinemático inverso, podem ocorrer problemas de singularidades, isto é, pode ocorrer que para alguns pontos da trajetória espacial não seja possível obter o ponto equivalente no espaço das juntas, significando que a trajetória espacial desejada não seja executável pelo manipulador (CRAIG, 2012).

### **3.6.1 Controle de trajetória dos robôs industriais**

Os métodos de controle de trajetórias utilizados na maioria dos robôs industriais comercializados no Brasil podem ser classificadas como: trajetórias ponto a ponto, movimento eixo a eixo, movimento simultâneo dos eixos, trajetórias coordenadas e trajetórias contínuas.

#### *3.6.1.1 Trajetórias ponto a ponto*

Neste tipo de trajetória cada articulação desenvolve desde sua posição inicial a até a final sem realizar consideração alguma sobre o estado ou desenvolvimento das demais articulações. Normalmente, cada atuador trata de levar sua articulação ao ponto de destino no menor tempo possível, podendo distinguir dois casos: movimento eixo a eixo e movimento simultâneo dos eixos.

#### *3.6.1.2 Movimento eixo a eixo*

Apenas se move um eixo de cada vez. Começará a mover-se a primeira articulação, e uma vez que esta tenha alcançado sua posição final à segunda inicia seu movimento, e assim sucessivamente. Este tipo de movimentação resulta um maior tempo de ciclo, tendo como única vantagem um menor consumo de potência instantânea para os atuadores.

### *3.6.1.3 Movimento simultâneo dos eixos*

Neste caso todos os atuadores começam simultaneamente a mover as articulações do robô a uma velocidade específica para cada uma delas. Dado que a distância a percorrer e as velocidades são em diferentes, cada articulação acabará sua movimentação em um instante diferente.

O movimento do robô não acabará até que se alcance definitivamente o ponto final, o que se produzirá quando o eixo que mais tarde conclua sua movimentação. Desta maneira, o tempo total gasto na movimentação coincidirá com o do eixo que mais tempo emprega em sua movimentação particular, podendo-se acontecer à situação em que o restante dos atuadores sejam forçados a movimentar a uma velocidade e aceleração elevada, tornando-se obrigados finalmente a esperar a articulação mais lenta. Por estes motivos expostos, as trajetórias ponto a ponto não estão empregadas salvo em robôs muito simples ou com unidades de controle muito limitadas. Exemplo robôs cartesianos e aplicações de paletização.

### *3.6.1.4 Trajetórias coordenadas*

Para evitar que alguns atuadores trabalhem forçando suas velocidades e acelerações, tendo que esperar depois da conclusão do movimento da articulação mais lenta, pode fazer-se um cálculo prévio, averiguando qual é esta articulação e que tempo gastará.

O tempo total gasto na movimentação é o menor possível e não são necessárias acelerações e velocidades elevadas dos atuadores de maneira desnecessária. Um conhecimento do modelo e controle cinemático do robô permitirá seu cálculo (CRAIG, 2012).

### *3.6.1.5 Trajetórias contínuas*

Quando se deseja que a trajetória que será seguida pela extremidade do robô seja conhecida (determinada) pelo usuário é preciso calcular de maneira contínua as trajetórias articulares. Geralmente, a trajetória que o usuário deseja que o robô realize apresenta o perfil em linha reta ou em arco de círculo no espaço de trabalho. Sendo necessário um mapeamento deste perfil para o espaço das articulações. O resultado será que cada articulação segue um movimento aparentemente caótico com possíveis mudanças de direção e velocidade e sem coordenação com as outras articulações no espaço das articulações.

O tipo de trajetória utilizado na implementação deste trabalho é a degrau, sendo que

os pontos são dados a partir do caminho gerado pelo algoritmo de campos potenciais. Para a execução dos movimentos, nos experimentos, os atuadores são acionados simultaneamente.

## 4 DESCRIÇÃO DO EXPERIMENTO

### 4.1 Introdução

Neste capítulo são apresentados os materiais utilizados no experimento; os métodos; os ensaios preliminares para a implementação do algoritmo de campos potenciais para geração de caminho livre de colisão; bem como os cenários dos ensaios para obtenção dos resultados.

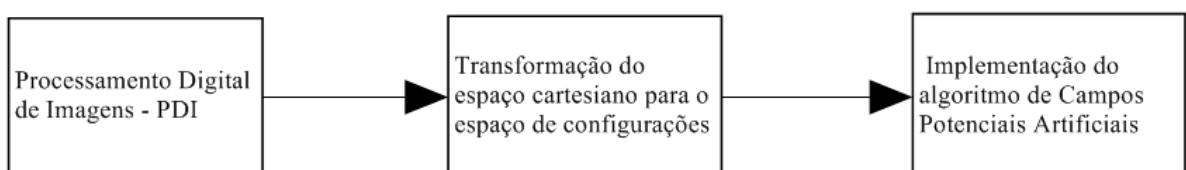
Aqui, pretende-se mostrar os passos desde o processamento da imagem até a implementação do algoritmo de CPA. A ideia principal é realizar o experimento para implementar campos potenciais artificiais na prevenção de colisão de um manipulador SCARA, em um ambiente com um obstáculo dinâmico, representado por um robô móvel. Todo o ambiente é monitorado por um sensor de imagens que indica a posição do obstáculo.

Um ponto importante com relação à metodologia adotada está no fato de que a partir das técnicas utilizadas, o obstáculo é substituído por uma circunferência que o circunscreve, reduzindo a geração de mínimos locais no momento do planejamento de caminhos para o algoritmo dos campos de potenciais artificiais, conforme indica Volpe e Khosla (1990), além do fato de tornar o sistema capaz de analisar obstáculos de vários formatos como triangular, retangular e trapezoidal que também reduzem a geração de mínimos locais no planejamento de caminhos.

Pretendemos verificar também, como será o comportamento do campo potencial na interação máquina-máquina, representada pelo manipulador e robô móvel. A partir dessa verificação serão coletados resultados, os quais nos permitirão analisar a eficiência da implementação do campo potencial na prevenção de colisão do manipulador em ambientes dinâmicos.

A sequência de execução para implementação do algoritmo de campos potenciais artificiais será executada para cada posição do obstáculo, ocorrerá uma adaptação do processo tornando o tratamento da informação mais adequada de acordo com a localização no espaço cartesiano. A figura 9 mostra o diagrama de blocos com a sequência de execução do processo.

Figura 9 – Sequência de execução para implementar o algoritmo de CPA



Fonte: elaborada pelo autor.

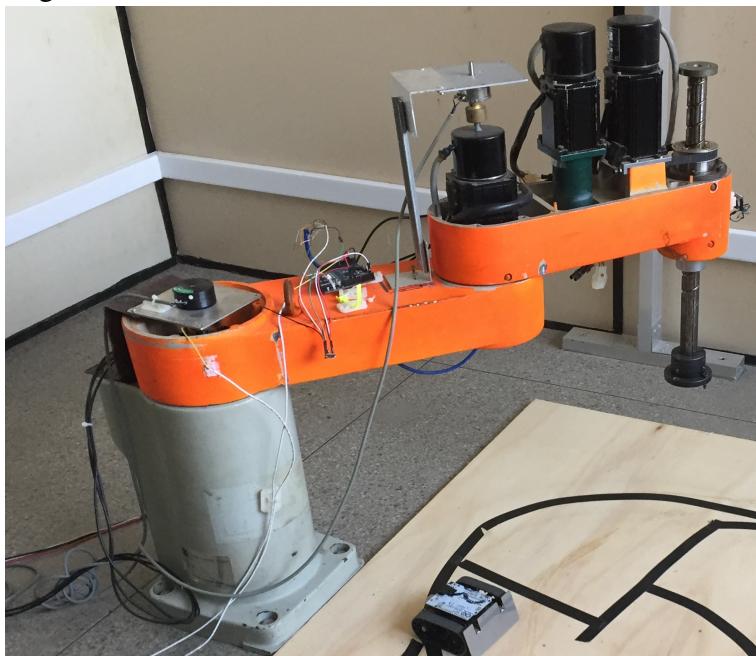
## 4.2 Materiais

Aqui são apresentados os materiais e equipamentos utilizados no experimento, para implementação do algoritmo de campos potenciais. Vamos apresentar as características do manipulador SCARA, sensor de imagens e robô móvel (obstáculo móvel).

### 4.2.1 Manipulador SCARA

O manipulador SCARA utilizado nesse trabalho é um robô de quatro Graus de liberdade (GDL), sendo que aqui será trabalhado com dois GDL, a figura 10 mostra o robô.

Figura 10 – Robô SCARA



Fonte: autor.

Por ser um robô tipo SCARA de 4 GDL, as duas primeiras juntas, a partir da base, são de rotação em torno de eixos verticais e trabalhando num plano horizontal (plano XY), como se fosse um robô de 2 GDL planar. A terceira junta é de translação, também chamada prismática permitindo deslocamentos no sentido vertical, ao longo do eixo z. E a quarta junta, de rotação, permite definir a orientação (ângulo  $\theta$ ) da ferramenta do manipulador em relação ao eixo Z.

#### **4.2.2 Sensor de imagens**

O sensor de imagens utilizado nesse trabalho para a aquisição de imagens é o sensor industrial de imagens PMD O3D200.

O sensor PMD 3D200 é uma câmera óptica de medição de distâncias utilizada em ambientes industriais. Os pixels medem a distância entre a câmera e a superfície mais próxima. A unidade ilumina a cena usando sua própria fonte de luz interna e processa a luz refletida da superfície. Em seguida, as imagens são processadas e podem ser utilizadas nas aplicações (IFM, 2009). A figura 11 mostra o sensor de imagens PMD O3D200. Neste trabalho são utilizadas as imagens de distância por não apresentar a mesma dependência do material das superfícies monitoradas.

Figura 11 – Sensor industrial de imagens  
PMD O3D200



Fonte: IFM (2009).

A medição de distância do sensor é baseada no princípio de tempo de voo onde é calculado o tempo que um feixe de luz gasta para atingir a superfície monitorada e retornar ao dispositivo (IFM, 2009).

A comunicação com o sensor é baseada em duas portas TCP/IP, onde uma porta é responsável por configurar a câmera a partir do protocolo servidor XML-RPC. A outra porta é responsável por transferir os dados de imagem a partir de um socket TCP/IP (IFM, 2009).

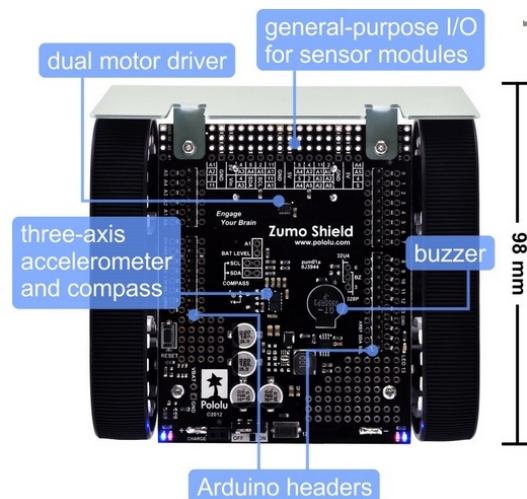
#### 4.2.3 Robô móvel (obstáculo móvel)

Nesse trabalho foi utilizado como obstáculo dinâmico um robô móvel, o robô Zumo da Pololu<sup>©</sup> Robotics e Electronics.

O robô Zumo é uma plataforma robótica sobre um par de trilhos de silicone, controlável por Arduino e menor do que 10 x 10 cm. O robô é acionado por dois micromotores metálicos acoplados a um par de trilhos de silicone, um conjunto de seis sensores de refletância para rastreamento de linhas ou detecção de bordas, um acelerômetro de 3 eixos com magnetômetro, uma buzina para sons simples e música. A alimentação elétrica é realizada através de quatro pilhas AA.

A figura 12 mostra as partes e conexões do robô e a figura 13 o arduino Uno montado no robô móvel.

Figura 12 – Robô Zumo - partes e conexões



Fonte: Pololu (2016).

Para o deslocamento do robô móvel foi elaborado vários caminhos arbitrários, dessa forma, não se sabe o caminho que o robô irá percorrer. O robô segue as linhas pretas no fundo claro, essa identificação é realizada através dos sensores de refletância. A figura 14 apresenta os caminhos arbitrários para o robô móvel.

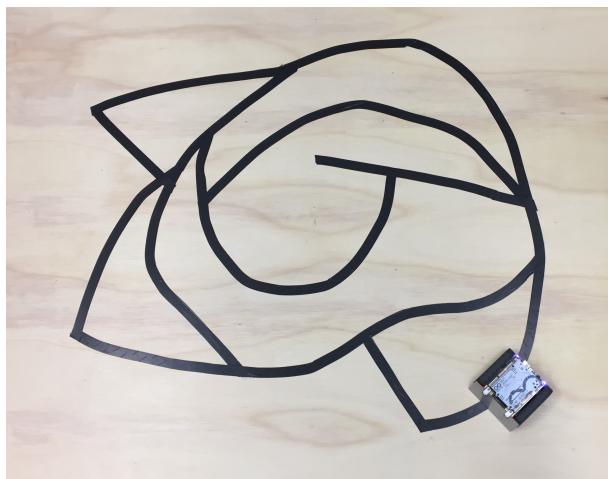
A ideia do caminho arbitrário é que o obstáculo estará na cena, em uma determinada posição, não conhecida pelo algoritmo de processamentos de imagens e pelo CPA, o qual este, será responsável pela geração de caminho livre de colisão do manipulador SCARA.

Figura 13 – Robô Zumo com arduino Uno



Fonte: autor.

Figura 14 – Caminhos arbitrários para o robô móvel



Fonte: autor.

Não foi utilizado um caminho mais simples pois pretende-se mostrar o comportamento da interação não-colaborativa máquina-máquina. A ideia aqui é, investigar o comportamento do algoritmo de campos potenciais na geração de caminho livre de colisão de forma mais complexa, ou seja, o robô móvel se comporta como um agente autônomo. Um caminho mais simples para o robô móvel não traduz a interação não-colaborativa máquina-máquina e o manipulador deve seguir o caminho livre de colisão independente da posição do robô móvel. Isso torna o sistema inteligente do ponto de vista da prevenção de colisão.

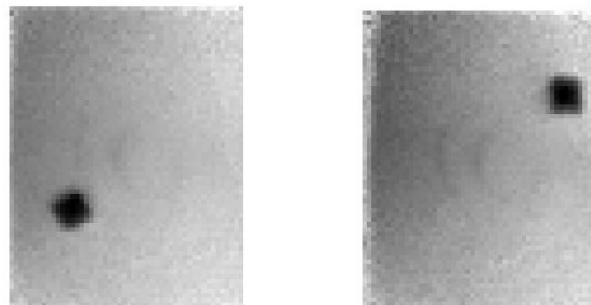
### 4.3 Métodos

Aqui serão apresentados os passos utilizados no Processamento Digital de Imagens (PDI), como o processamento morfológico e segmentação de imagens, juntamente com técnicas de mapeamento de obstáculos e a implementação do algoritmo de campos potenciais artificiais para geração de caminho livre de colisão. O obstáculo utilizado é um robô móvel, apresentado anteriormente, que estará realizando caminhos arbitrários no espaço de trabalho do manipulador. O algoritmo de CPA fará com que o manipulador desvie do robô móvel para evitar colisão. Também será mostrado como foi realizada transformação do espaço cartesiano para espaço de configurações.

#### 4.3.1 Processamento digital de imagem

Para exemplificar melhor os processos, serão utilizadas duas cenas como referência, apresentadas na figura 15. As cenas são as imagens obtidas pelo sensor de imagens de duas posições diferentes do robô móvel.

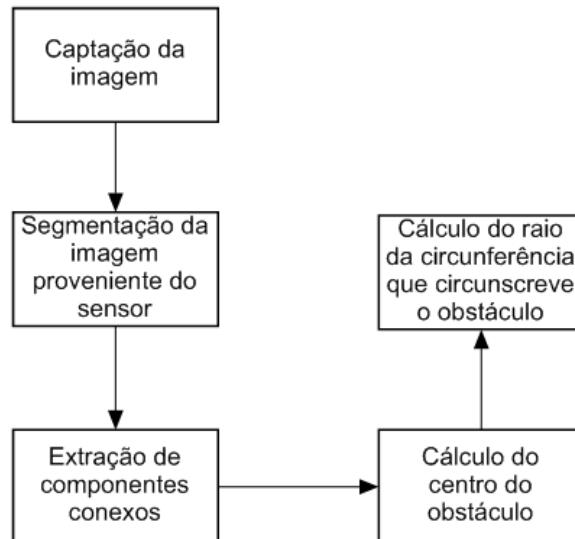
Figura 15 – Imagens obtidas a partir do sensor de imagem:  
a) Cena 01 a esquerda;  
b) Cena 02 a direita



Fonte: autor.

Para a execução do processamento digital das imagens obtidas pelo sensor de imagens, são executadas as etapas mostradas no diagrama de blocos, na figura 16, a seguir.

Figura 16 – Diagrama de blocos com as etapas do processamento das imagens do sensor

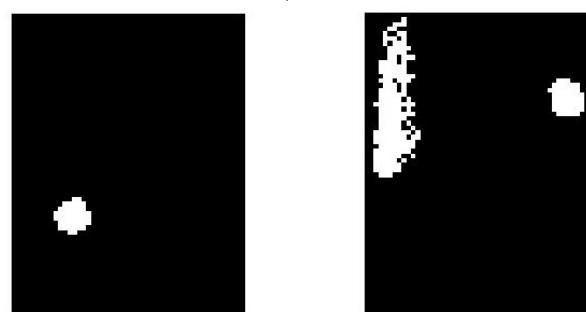


Fonte: autor.

#### 4.3.1.1 Segmentação da imagem proveniente do sensor

A atividade de segmentação é responsável pela diferenciação entre objeto e fundo. Esse resultado é obtido através da segmentação por limiarização, onde o nível de limiarização é escolhido a partir do método de Otsu. A figura 17 mostra o resultado da atividade para as cenas 01 e 02.

Figura 17 – Imagens limiarizadas utilizando método de Otsu: a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

Na figura 17 vê-se o fundo representado pela cor preta e o obstáculo pela cor branca.

#### 4.3.1.2 Extração de componentes conexos

Na atividade de extração de componentes conexos, cada conjunto de pontos referente ao obstáculo são separados dos elementos que não são obstáculos, tornando possível a diferenciação entre o obstáculo e quaisquer elemento que não esteja em contato, como pode ser observado na figura 18.

Figura 18 – Imagens após a extração de componentes conexos: a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

#### 4.3.1.3 Cálculo do centro do obstáculo

Após a atividade de extração de componentes conexos, onde o obstáculo é diferenciado de outros elementos, é possível desenvolver o cálculo do centro do obstáculo. A partir da posição de cada pixel de fronteira do obstáculo, é calculada a média aritmética para as coordenadas x e y obtendo assim um ponto para o centro do obstáculo.

#### 4.3.1.4 Cálculo do raio da circunferência que circunscreve o obstáculo

A partir da obtenção do centro do obstáculo é realizado um cálculo de distância euclidiana entre cada pixel de fronteira do obstáculo e o centro do mesmo. A maior distância encontrada nesta comparação iterativa é utilizada como raio da circunferência que deve circunscrever o obstáculo, tal qual é mostrado na figura 19.

Figura 19 – Circunscrição de obstáculo: a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

#### 4.3.2 Transformação do espaço cartesiano

Na literatura, planejamento de caminho é uma especificação completa da localização de cada ponto no robô e é referida como uma configuração, e o conjunto de todas as configurações possíveis é referido como o espaço de configuração. Para os nossos propósitos, o vetor de variáveis de articulação,  $q$ , fornece uma representação conveniente de uma configuração e  $Q$  é o espaço de configuração.

Para um robô de um link de revolução, o espaço de configuração é o conjunto de orientações do link, assim  $Q = S^1$ , onde  $S^1$  representa um círculo unitário. De qualquer forma, podemos parametrizar  $Q$  por um único parâmetro, o ângulo da junta  $\theta_1$ . Para um robô planar de dois links, teremos  $Q = S^1 \times S^1$ , e podemos representar essa configuração por  $q = (\theta_1, \theta_2)$ . Para um robô cartesiano, teremos  $Q = \mathbb{R}^3$ , podemos representar a configuração por  $q = (d_1, d_2, d_3) = (x, y, z)$  (SPONG, 2006).

Uma colisão ocorre quando o robô entra em contato com um obstáculo no espaço de trabalho. Denotaremos o robô por  $A$ , e por  $A(q)$  o subconjunto do espaço de trabalho que é ocupado pelo robô na configuração  $q$ .  $O_i$  é definido como os obstáculos no espaço de trabalho e  $W$  o espaço de trabalho do robô (espaço cartesiano no qual o robô se move). Para planejar um caminho livre de colisão, devemos garantir que o robô nunca chegue a uma configuração  $q$ , que o faça entrar em contato com um obstáculo no espaço de trabalho. O conjunto de configurações para o qual o robô colide é referido como o obstáculo no espaço de configuração e é definido por (SPONG, 2006):

$$QO = \{q \in Q | A(q) \cap O \neq \emptyset\} \quad (4.1)$$

Definiremos  $O = \cup O_i$ . O conjunto de configurações livre de colisões será denotado por:

$$Q_{free} = Q \setminus QO \quad (4.2)$$

O problema de planejamento de caminho é encontrar um caminho a partir de uma configuração inicial  $q_{inic}$  para uma configuração final  $q_{final}$ , de modo que o robô não colide com nenhum obstáculo à medida que percorre o caminho.

Uma questão neste contexto é a necessidade de se transformar a informação de localização do obstáculo na cena, levando-a do domínio de pontos do espaço cartesiano em pontos do espaço de configurações do robô. Isto se consegue mediante cálculo de cinemática inversa do robô em questão, para cada ponto da superfície do obstáculo, gerando o espaço de configurações de obstáculos. A prevenção de uma colisão é, portanto, uma estratégia que impeça o robô de atingir alguma configuração constante em  $QO$ . A complexidade e o esforço computacional para gerar o  $QO$  de uma dada cena crescem se não apenas o *end-effector*, mas toda a cadeia articulada do robô tiver que ser protegida de colisões (PEREIRA *et al.*, 2014).

#### 4.3.3 Simulação do algoritmo de CPA na geração de caminho livre de colisão

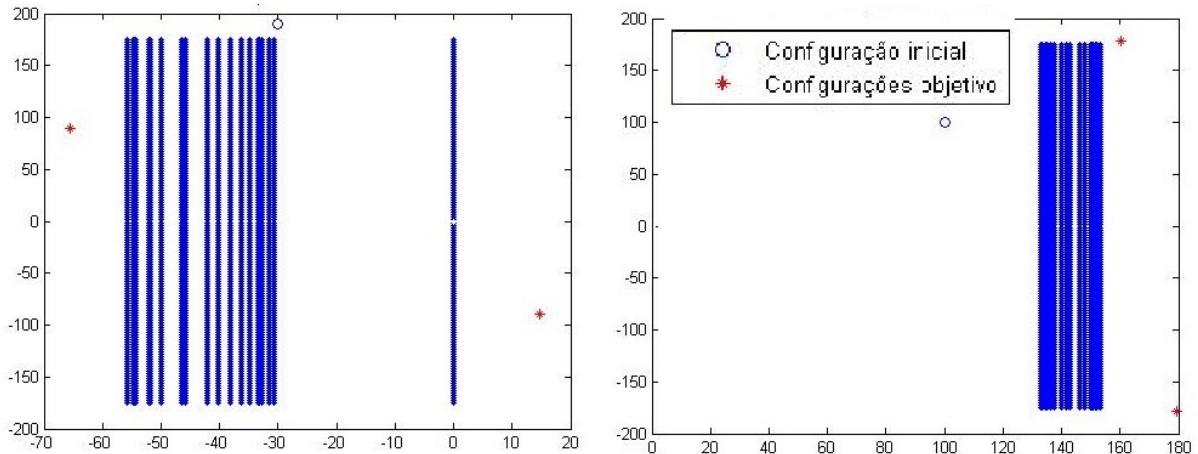
Aqui será apresentado como foi realizada a simulação da implementação do algoritmo de campos potenciais para a geração de caminho livre de colisão para o manipulador SCARA em estudo. Essa implementação foi realizada através de simulações no Matlab.

##### 4.3.3.1 Espaço de configurações de obstáculos

Após o processamento das imagens do obstáculo é realizado o mapeamento do espaço de configurações de obstáculos ( $QO$ ). Para executar esse mapeamento, são calculadas as configurações do robô, a partir da cinemática inversa. Foram utilizadas como entrada as coordenadas cartesianas de cada ponto da circunferência que representa o obstáculo. Aqui vamos analisar várias abordagens que leva em consideração tanto o mapeamento do  $QO$  referente ao atuador do manipulador, como o  $QO$  referente a toda a estrutura do manipulador. Para que se tenha um  $QO$  que represente por completo as configurações de colisão, a estrutura cinemática do manipulador é discretizada em pontos de controle, agregando uma nova cinemática inversa para cada ponto de controle atribuído.

A figura 20 mostra o espaço de configurações de obstáculos para duas posições onde o obstáculo se encontra na cena, para o atuador do manipulador.

Figura 20 – Espaços de configuração de obstáculos ( $QO$ ). a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

A faixa de valores de  $QO$  similar a um retângulo, corresponde a colisão dos obstáculos com o primeiro elo do manipulador. Esta faixa de valores de  $QO$  se apresenta desta forma devido ao fato de que caso ocorra colisão com este primeiro elo, o valor do ângulo  $\theta_2$  é irrelevante, sendo então considerados todos os valores de  $\theta_2$  dentro do espaço de trabalho como sendo pertencentes a  $QO$ , para este  $\theta_1$  específico.

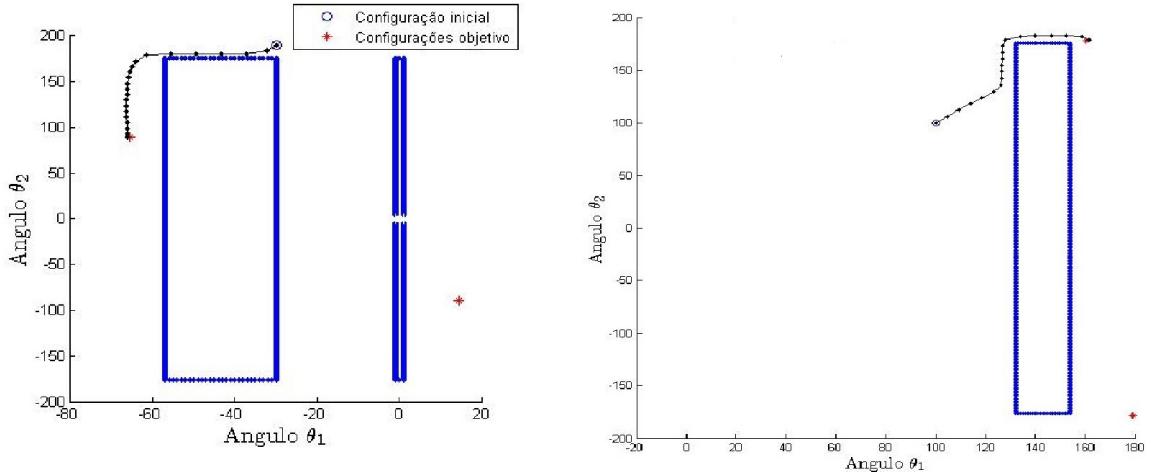
#### 4.3.3.2 Simulação de caminho com o algoritmo de campo potencial

Para a geração de caminho usando CPA foram selecionadas intensidades para os parâmetros utilizados no algoritmo de campo potencial. Os valores das constantes de atração do ponto final ( $K_a$ ) e repulsão do obstáculo ( $K_r$ ) utilizados foram: 1 e 5000, respectivamente. Estes valores apresentaram os melhores resultados na simulação, adotando como critérios atingir a posição sem colisão com obstáculo e a manutenção da distância mínima do obstáculo para o caminho não ser demasiadamente longo. Considerar o caminho gerado como sendo o caminho executado pelo atuador do manipulador planar no espaço de trabalho, dados os pontos inicial e final. O campo potencial gerado com estes parâmetros esta representado na figura 21. O caminho gerado pelo algoritmo de campos potenciais é formado por vários pontos, a quantidade de pontos aqui utilizada foi igual a 50..

A figura 21 mostra o caminho realizado pelo CPA a partir do espaço de configurações

de obstáculos, mostrado na figura 20.

Figura 21 – Caminho realizado pelo CPA a partir de  $QO$ . a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

Nas simulações observa-se que a principal vantagem do método de campo potencial artificial é o baixo custo computacional do algoritmo, permitindo aplicações em ambientes dinâmicos. Este custo pode ser ainda menor se o espaço de trabalho for pré-processado, como em ambientes estáticos. Nesse trabalho o obstáculo é dinâmico como apresentado anteriormente.

#### 4.3.3.3 Área e superfície do campo potencial artificial

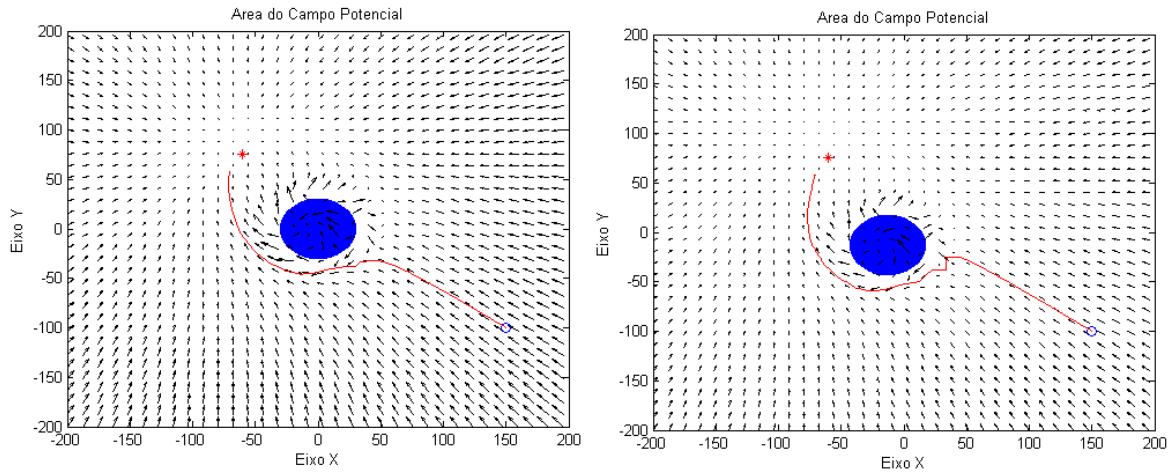
Realizamos uma simulação de geração de caminho para apresentar as linhas do campo potencial artificial e a superfície potencial resultante, para mostrar o comportamento do obstáculo móvel posicionado em dois locais diferentes.

O campo vetorial de forças gerado pelo gradiente, ou as linhas do campo artificial, são representadas com espaçamento (distância entre as linhas de campo) de 0,4 milímetros entre linhas. As simulações foram realizadas com os parâmetros  $K_a$  e  $K_r$  constantes.

A figura 22 apresenta as linhas do campo potencial artificial e o caminho livre de colisão, no espaço cartesiano para a simulação executada, obtidos a partir das localizações do obstáculo. Na cena 01 o obstáculo se encontra na posição [0, 0] e na cena 02 em uma posição mais adiante, após decorrer um determinado tempo.

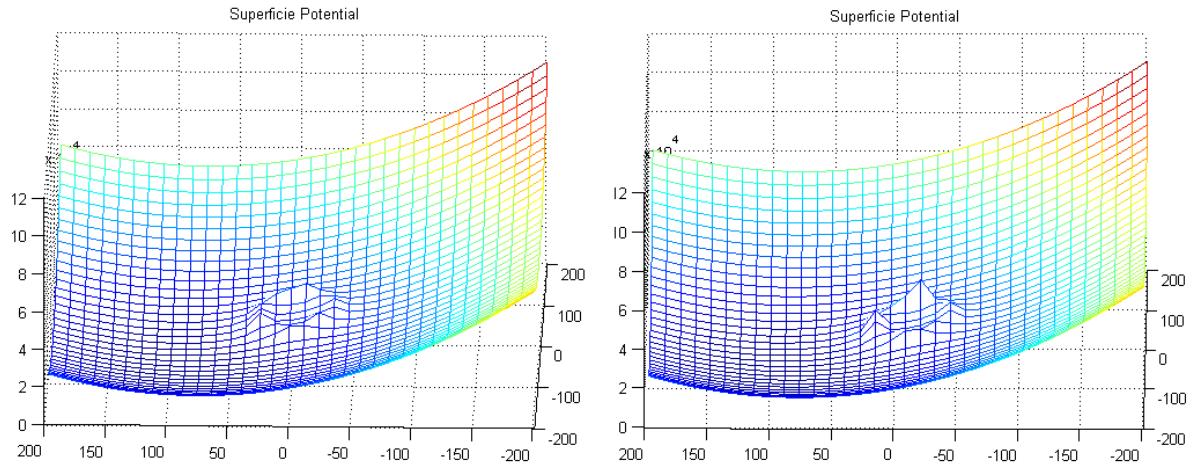
A figura 23 apresenta a superfície resultante do campo potencial para a cena 01 e 02, de acordo com a simulação executada na figura 22.

Figura 22 – Linhas do campo potencial e caminhos realizado pelo algoritmo de CPA. a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

Figura 23 – Superfícies resultantes do algoritmo de CPA. a) Cena 01 a esquerda; b) Cena 02 a direita



Fonte: autor.

Na figura 22 observa-se que na posição inicial do manipulador (ponto inicial do caminho) tem-se uma maior concentração das linhas do campo potencial e no ponto final uma concentração menor. As linhas do campo potencial estão orientadas na mesma direção do caminho. Percebe-se também que a orientação das linhas de campo próximo ao obstáculo tem sentido saindo do obstáculo (forças de repulsão) para evitar a colisão.

Na figura 23 a região em vermelho está relacionada à posição inicial do caminho, onde apresenta uma maior quantidade de linhas do campo potencial, que é um ponto de máximo da curva e a região em azul está localizado o ponto final onde a quantidade de linhas é menor, esta região é um ponto de mínimo da curva do campo potencial. Os picos representam a região onde está localizado o obstáculo, ou seja, onde apresenta a força de repulsão para evitar a colisão.

## 4.4 Ensaios preliminares

Nesta seção mostraremos alguns ensaios preliminares que foram necessários para realizar a implementação do algoritmo de CPA no manipulador SCARA.

### 4.4.1 Sequência de operações

Para a implementação do algoritmo de campos potenciais foi desenvolvida uma sequência de operações, que vai desde a aquisição da imagem, através do sensor de imagens, até o envio dos pontos para o controlador do manipulador.

O primeiro passo executado foi a aquisição da imagem realizada através do sensor de imagens (ver figura 11). A partir da obtenção da imagem do obstáculo dinâmico (robô móvel, ver figura 13) foi realizado o processamento da imagem no Matlab<sup>®</sup> que gera, a partir do obstáculo identificado pelo sensor e da cinemática inversa do manipulador o espaço de configurações do obstáculo. Em seguida, o algoritmo de campo potencial gera o caminho livre de colisão a partir dos pontos inicial e final escolhidos.

Nesse primeiro momento para a geração do caminho do manipulador, pegou-se o segundo ponto do caminho para enviar para o controlador, esse ponto encontra-se no espaço de configurações. Porém, antes do envio desse ponto foi necessário colocá-lo no espaço cartesiano da base do manipulador, através da resolução da cinemática direta, e em seguida foi transformado, novamente, para o espaço das juntas resolvendo a cinemática inversa. Essa transformação foi necessária pois, o caminho gerado pelo algoritmo de CPA foi gerado pelo sistema cartesiano do sensor de imagens.

Esse ponto, agora no espaço das juntas, é transformado, através de alguns cálculos, no set-point que é então enviado para o controlador *Fuzzy* através do Matlab<sup>®</sup> utilizando o protocolo OPC.

Para enviar os sinais de referência das posições para o controlador, embarcado no CLP, foi utilizado o protocolo Controle de Processos OLE (OPC), que significa “*Object Linking and Embedding (OLE)*” juntamente com um sistema de aquisição de dados, sistema *SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA)*, que envia os dados do Matlab<sup>®</sup> para o CONTROLADOR LÓGICO PROGRAMÁVEL (CLP).

Para a realização do controle de trajetória foi utilizado o controlador *Fuzzy* do tipo Sugeno de Ordem Zero para cada junta do manipulador, implementado em um CLP Twido da

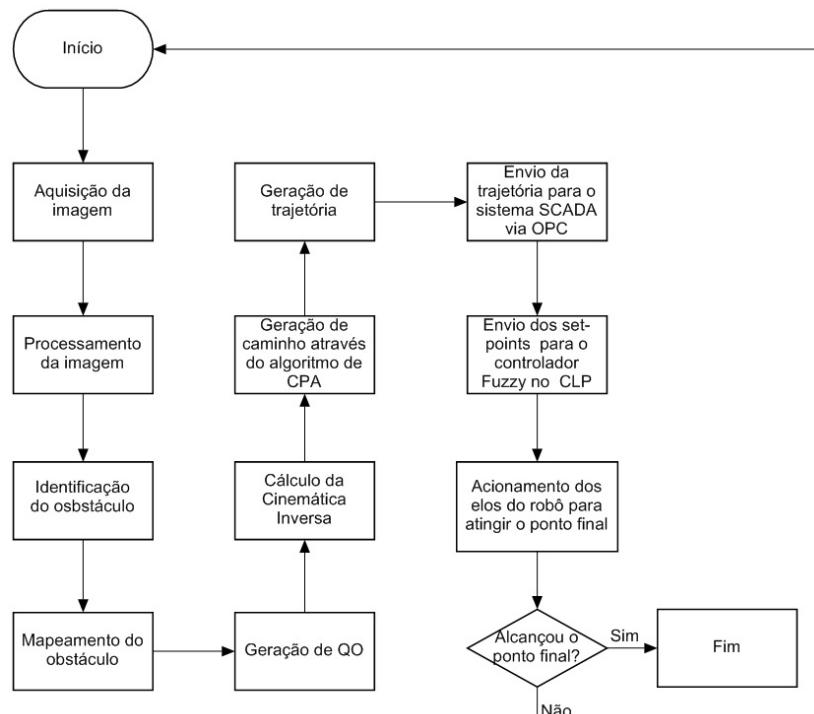
Schneider. O controlador *Fuzzy* foi implementado através de blocos de funções especiais da linguagem Ladder embarcados no CLP (SILVA, 2016).

O sistema SCADA consiste de uma aplicação desenvolvida para controlar e registrar as leituras dos transdutores que realizam a medição dos ângulos das juntas e a supervisão do controlador *Fuzzy* embarcado no CLP. Além de funcionar como servidor OPC para comunicação com o Matlab®. A comunicação entre o CLP e o computador, onde roda o supervisório, é estabelecida via serial através de cabo dedicado para a comunicação Modbus. Foi utilizada nesse trabalho uma versão gratuita do software Elipse SCADA® (SILVA, 2016).

Importante ressaltar aqui, que a velocidade do robô móvel foi ajustada de acordo com com a velocidade do manipulador e também, em função do tempo de processamento da imagem, tempo de processamento do algoritmo de campos potenciais e tempo de envio dos set-points para o controlador. A velocidade do manipulador é considerada satisfatória para uma tarefa *pick-and-place* dentro de um processo de manufatura.

Para uma melhor visualização de todo o sistema, a figura 24 mostra o fluxograma do funcionamento do processo. Todo o processo acontece em tempo real, caso o manipulador não consiga alcançar o ponto final o mesmo deve parar para evitar a colisão como o obstáculo dinâmico (robô móvel).

Figura 24 – Funcionamento do processo de implementação do CPA

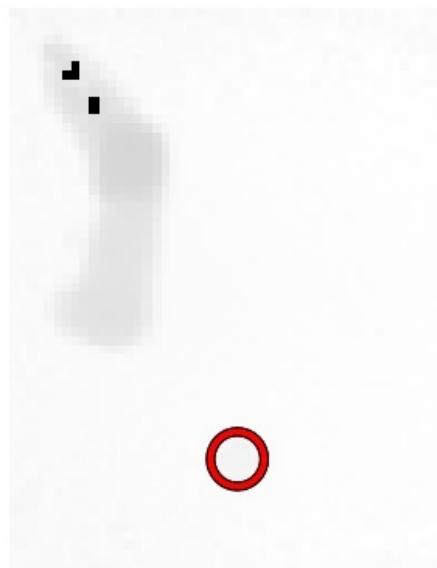


Fonte: autor.

#### **4.4.2 Simulação do algoritmo de CPA na geração de caminho do manipulador**

A seguir será apresentado um ensaio de caminho livre de colisão do manipulador. Primeiramente foi realizada a aquisição da imagem através do sensor de imagens como mostra a figura 25. Nessa figura observa-se claramente a imagem do manipulador, o círculo em vermelho é para destacar o obstáculo (robô móvel). Essa é a imagem real do sensor sem nenhum processamento.

Figura 25 – Aquisição da imagem pelo sensor de imagens

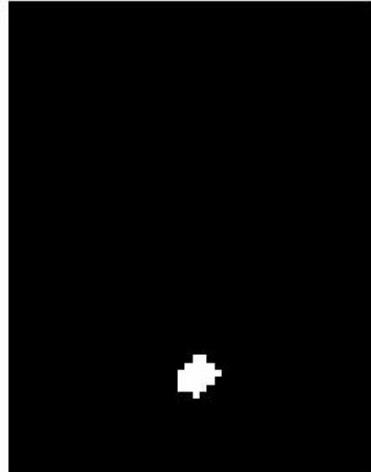


Fonte: autor.

Após o processamento da imagem, teremos na cena somente o obstáculo, pois a partir dele será gerado o mapeamento do obstáculo, a geração de QO e o caminho livre de colisão. Na figura 26 é mostrado o obstáculo na cena.

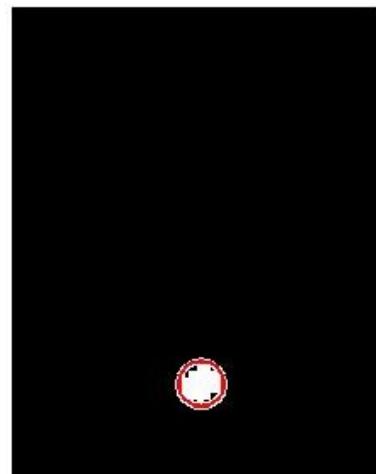
No processamento da imagem é desenhado um círculo (figura 27 ) em volta do obstáculo para melhorar o processamento do algoritmo de CPA e evitar a ocorrência de mínimos locais.

Figura 26 – Identificação do obstáculo na cena



Fonte: autor.

Figura 27 – Círculo em volta do obstáculo

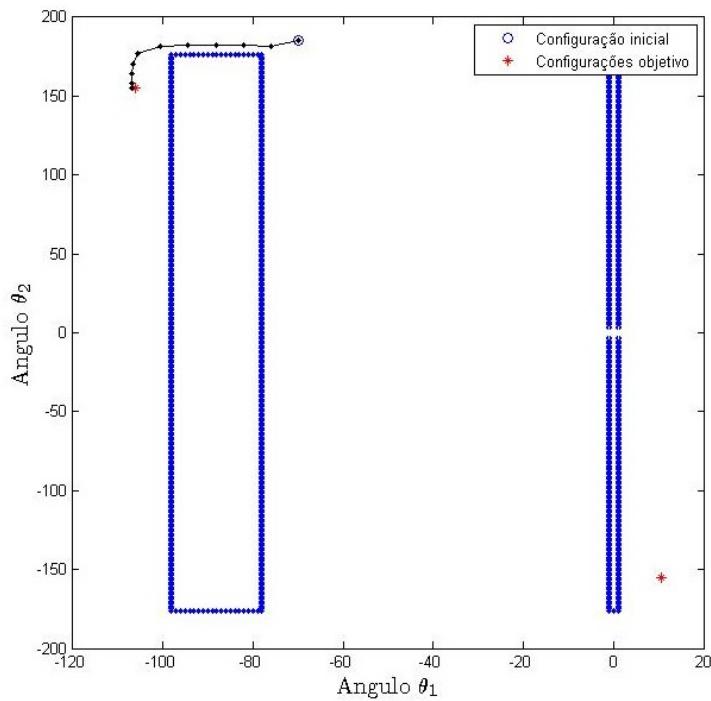


Fonte: autor.

Após o processamento da imagem é gerado então o espaço de configurações de obstáculo, QO e o caminho livre de colisão. Na figura 28, a seguir, é mostrado um caminho livre de colisão. Os retângulos em azul são os espaços de colisão gerado pelas juntas do manipulado que irá colidir com o obstáculo.

Para realizar o ensaio foram utilizados o ponto inicial,  $q_0 = [-70, 185]$  e o ponto final  $q_F = [10, -11]$ , pontos esses no espaço de configurações (juntas).

Figura 28 – Caminho livre de colisão do manipulador



Fonte: autor.

Esse ensaio foi necessário para fazer alguns ajustes em todo os sistema e verificar o comportamento do manipulador.

## 4.5 Cenários de ensaios

Aqui serão apresentados os cenários relacionados a implementação do algoritmo de campos potenciais e como foi realizado a avaliação da eficiência do algoritmo.

### 4.5.1 Implementação do algoritmo de campos potenciais

Para realizar a implementação do algoritmo de CPA, foram feitas várias abordagens com relação ao manipulador e o robô móvel. As abordagens consideram o manipulador e o robô móvel como um ponto ou um corpo extenso. Realizando as combinações dessas considerações temos as seguintes abordagens: dois corpos extensos; corpo extenso - ponto; ponto - corpo extenso; ponto - ponto.

Na abordagem dois corpos extensos o manipulador e o robô móvel são considerados corpos extensos. Neste caso, para o manipulador, é levado em consideração toda a cadeia de elos do mesmo e o robô móvel é considerado um círculo onde os pontos de repulsão estão localizados em todo o perímetro do círculo ao redor do robô móvel, ou seja, o círculo circunscreve o robô

móvel. Nessa abordagem, o espaço de configurações de obstáculo é em função dos elos do manipulador e o obstáculo (robô móvel). Para todas as abordagens onde o manipulador foi considerado corpo extenso cada elo foi discretizado em 5 pontos.

Na abordagem ponto - ponto, o manipulador e o robô móvel são considerados, cada um, como um ponto. No manipulador a localização desse ponto está no *end-effector*, no final do segundo elo. No robô móvel esse ponto está localizado no centro do mesmo.

As outras abordagens são combinações entre o robô móvel e o manipulador sendo considerado ponto ou corpo extenso. Na abordagem corpo extenso - ponto, o manipulador é um corpo extenso e o robô móvel um ponto e na ponto - corpo extenso, o manipulador é considerado um ponto e o robô móvel um corpo extenso (círculo).

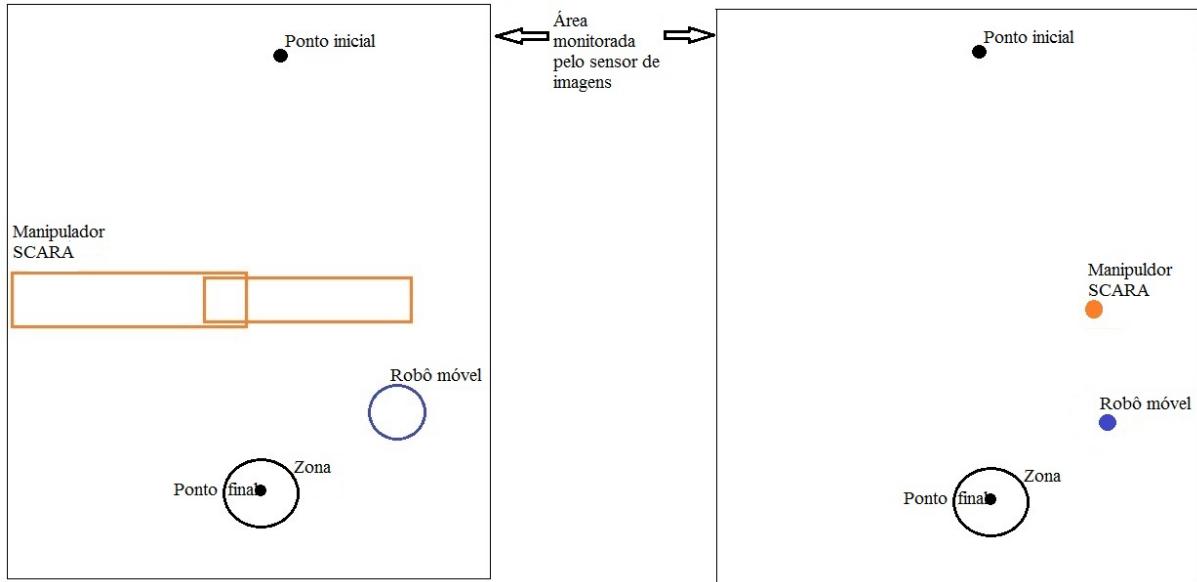
Para a realização do experimento foram escolhidos o ponto inicial,  $q_0 = [58, 6]$  e o ponto final,  $q_F = [-42, -43]$ , pontos no espaço das juntas e o caminho entre esses pontos foi gerado pelo algoritmo de campos potenciais. A decisão da escolha desses ponto, foi por se tratar de pontos que o caminho entre eles percorresse o máximo do espaço de trabalho do manipulador.

Para cada abordagem foi realizado um ensaio onde o manipulador tem como destino alcançar um ponto ou uma zona. Se ele conseguisse chegar a esse ponto ou a essa zona, considerou-se que a missão foi concluída. Na missão ponto para ponto o manipulador alcançou um ponto de destino. Tarefas *pick and place*, montagens, etc, são tarefas que exigem que o manipulador alcance um determinado ponto com precisão. Na missão ponto para zona, considerou-se uma determinada região, próxima ao ponto de destino, que o manipulador conseguisse alcançar, neste caso, considerou-se um círculo onde o centro é o ponto final e a zona é a área interna desse círculo. Tarefas como paletização, colocar uma peça em uma esteira, esses locais onde a peça será colocada podem ser considerados uma zona onde o manipulador pode deixar a peça e não exige-se grande precisão do mesmo.

A figura 29, a seguir, apresenta uma ilustração que representa o manipulador e robô móvel. Na imagem a esquerda, tem-se a abordagem em que o manipulador e robô móvel são considerados corpos extensos e na figura a direita, a representação de ambos como ponto. Também são mostrados as missões zona e ponto.

Durante a realização dos experimentos foram coletados: a quantidade de ensaios; o tempo de cada missão; o número de colisões; a quantidade de vezes que o manipulador parou para não ocorrer colisão; e a produtividade.

Figura 29 – Representação do manipulador e robô móvel. a) à esquerda: manipulador e robô móvel considerados como corpos extensos; b) à direita: manipulador e robô móvel considerados como pontos



Fonte: autor.

A não conclusão da missão ocorreu quando o manipulador parou para evitar a colisão ou quando ocorreu a colisão, nessas situações, considerou-se que o manipulador não conseguiu concluir a missão. Já a produtividade, é a quantidade de vezes que o manipulador concluiu a missão sem a ocorrência de colisão.

#### 4.5.2 Eficiência do algoritmo de campos potenciais

Para a avaliação do algoritmo de campos potenciais, identificamos uma função matemática que leva em consideração a produtividade, a variância (variabilidade) do tempo da missão e o número de colisões. Essa função é a eficiência do algoritmo de campos potenciais artificiais, que é uma figura de mérito. A eficiência é diretamente proporcional a produtividade e inversamente proporcional ao produto da variância do tempo e o número de colisões, como mostrado na seguinte equação:

$$\eta_{CP} = \frac{\text{Prod}}{\sigma_{temp} N_{col.}} \quad (4.3)$$

onde  $\eta_{CP}$  é a eficiência do algoritmo de campos potenciais; Prod (Produtividade) é a quantidade de vezes que a missão foi cumprida;  $\sigma_{temp}$  é a variância do tempo das missões e  $N_{col.}$  é o número de colisões.

A variância do tempo é dada por 4.4, abaixo:

$$\sigma_{temp} = \frac{\sum_{i=1}^n (t_i - \bar{t}_i)^2}{n - 1} \quad (4.4)$$

A variância é uma medida de dispersão que apresenta a distância que cada valor do conjunto está do valor central (médio). Quanto menor a variância, mais próximos da média estão os valores do conjunto; mas quanto maior ela for, os valores estão mais distantes da média.

#### **4.5.3 Eficiência do algoritmo através do OEE**

O índice de eficiência global de equipamentos (OEE - *Overall Equipment Effectiveness*), é um sistema de medição de manufatura que avalia o equipamento através dos índices de performance, disponibilidade e qualidade. Estes índices permitem identificar as perdas do equipamento (NAKAJIMA, 1989), (PEREIRA, 2011).

Neste trabalho o OEE foi utilizado para medir a eficiência do manipulador SCARA, na execução da tarefa de deslocamento de um ponto inicial a um ponto final utilizando o algoritmo de campos potenciais para evitar a colisão.

#### *Disponibilidade*

É o percentual do tempo que o equipamento estava trabalhando, comparado ao total do tempo disponível para ser utilizado. A disponibilidade foi calculada a partir da seguinte equação:

$$Disp.(%) = \frac{TTD - PP - PNP}{TTD - PP} * 100 \quad (4.5)$$

onde, Tempo Total Disponível (TTD), Tempo para Paradas Planejadas (PP) e Tempo para Paradas não Planejadas (PNP).

Aqui, considerou-se que não aconteceram paradas planejadas, as paradas não planejadas aconteceram quando houve colisão, quando o manipulador parou pra evitar colisão ou a parada para algum ajuste. Para realizar o cálculo da disponibilidade, computou-se o tempo total do ensaio como tempo total disponível e o tempo de parada não planejada. A diferença entre o tempo total disponível e o tempo de trabalho é o tempo de paradas não planejadas.

### *Performance*

Apresenta uma relação entre o total de peças produzidas real e teórico, levando em consideração o tempo de ciclo. Ou seja, avalia o ritmo de produção do equipamento. Para o cálculo da performance utilizou-se a equação a seguir:

$$\text{Perf.}(\%) = \frac{\text{TEO}}{\text{TO}} * 100 \quad (4.6)$$

onde, Tempo Efetivo de Operação (TEO) e Tempo de Operação (TO).

Para o cálculo da performance, consideramos o tempo de efetivo de operação o tempo que as missões foram executadas com sucesso e o tempo de operação, a soma de todos os tempos (tempo total).

### *Qualidade*

Refere-se à geração de produtos defeituosos, que resultam em refugo ou retrabalhos.

Aqui consideramos que a qualidade foi igual a 100%, pois não é produzido nenhuma peça, ou seja, o manipulador apenas desloca-se de um ponto a outro. Neste caso é uma tarefa de manipulação de um produto (*pick-and-place*).

### *Cálculo do OEE*

O indicador *Overall Equipment Effectiveness* (OEE) é o produto dos três índices relacionados acima, disponibilidade, performance e qualidade. Este índice pode ser obtido através da equação a seguir:

$$\text{OEE}(\%) = \text{Disp.} * \text{Perf.} * \text{Qual.} * 100 \quad (4.7)$$

#### **4.5.4 Cálculo do torque e do consumo de energia mecânica do motor da junta 1**

Para cada ensaio realizado, calculou-se o consumo de energia mecânica e o torque do motor da junta 1. Escolheu-se somente o motor da junta 1, pois o mesmo apresenta uma maior potência em relação ao motor da junta 2.

Para auxiliar na execução do cálculo do torque do motor, foi necessário as informações de placa do motor, como mostrado na tabela 2 a seguir:

Tabela 2 – Dados de placa do motor da junta 1

Velocidade máxima ( $min^{-1}$ )	Nº de polos	Potência (W)
3000	2	540

Fonte: autor.

Para realizar o cálculo do torque do motor 1, utilizou-se a equação a seguir (FITZGERALD *et al.*, 2006):

$$T_{m1} = 9,55 \frac{P_{m1}}{N_{m1}} \quad (4.8)$$

onde,  $T_{m1}$  é o torque do motor 1 em Nm,  $P_{m1}$  é potência nominal do motor 1 em W e  $N_{m1}$  o valor da velocidade do motor em rpm ( $min^{-1}$ ).

Os valores de velocidade foram obtidos a partir das referências de velocidades enviadas do CLP para o drive de acionamento (inversor de frequência) do motor. Por se tratar de um motor síncrono de ímãs permanentes, o mesmo não tem escorregamento de velocidade e, conhecendo-se a frequência de saída do inversor, determina-se a velocidade a partir de (FITZGERALD *et al.*, 2006):

$$N_{m1} = \frac{120f}{P} \quad (4.9)$$

onde, f é a frequência da saída do inversor em Hz e P o número de polos do motor.

A energia mecânica foi calculada a partir de:

$$E = \sum_{i=1}^n (P_{m1i} t_i) \quad (4.10)$$

onde, E é energia mecânica do motor em KWh, P é a potência calculada durante os ensaios em W e t é o tempo dos ensaios em h (horas).

A potência durante os ensaios foi calculada da seguinte forma (FITZGERALD *et al.*, 2006):

$$P_{m1} = \frac{T_{m1} V_{m1}}{9549,2965} \quad (4.11)$$

onde,  $P_{m1}$  é a potência durante os ensaios em KW,  $T_{m1}$  é o torque obtido a partir de 4.8 e  $V_{m1}$  é a velocidade obtida a partir da posição da junta 1 do manipulador.

#### 4.5.5 Cálculo do erro médio entre as posições reais e as posições geradas pelo algoritmo do CP

Quando realiza-se uma medição, não importa o quanto confiável ela seja, deve-se estabelecer uma relação entre o valor verdadeiro e o valor medido. A diferença entre estes valores é o erro. Os desvios ou erros de medidas são falhas na medição que pode ocorrer por diversos motivos, erros do operador, pela forma incorreta de medir, ou algum problema que o equipamento pode apresentar (ARNOT, 2002).

Calculamos aqui o erro médio entre as posições geradas pelo algoritmo do CP e as posições reais das juntas. Para isto, utilizou-se a equação abaixo:

$$Err = \frac{\sum_{i=1}^n (\theta_{CPi} - \theta_{Ri})}{n} \quad (4.12)$$

Onde,  $Err$  é o erro médio entre entre as posições geradas pelo algoritmo do CP ( $\theta_{CP}$ ) e as posições reais ( $\theta_R$ ); e  $n$  é a quantidade das posições medidas.

## 5 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados e discussões técnicas sobre a aplicação do método de campos potenciais artificiais, bem como as considerações finais deste trabalho.

### 5.1 Resultados

Nesta seção são apresentados os resultados após a implementação do algoritmo de campos potenciais, na geração do caminho do manipulador; a avaliação do algoritmo; a análise do torque e consumo de energia mecânica do motor da junta 1; e por último, a análise do erro entre os ângulos gerado pelo algoritmo de CP e a posição real das juntas.

#### 5.1.1 *Resultados da implementação do algoritmo de CPA na geração de caminho do manipulador*

Para cada abordagem implementada, conforme apresentado em 4.5.1, foram realizados os experimentos e coletadas as informações referentes a quantidade de ensaios, tempo de cada missão, número de colisões, a quantidade de vezes que o manipulador parou para não ocorrer colisão e a produtividade. O tempo total do ensaio é a soma dos tempos de cada missão, seja ela cumprida, ou não.

É importante ressaltar que, quando ocorreu uma colisão ou o manipulador parou para evitar uma colisão, foi medido o tempo em que o manipulador saiu do ponto inicial até a ocorrência da colisão e também o tempo que o manipulador parou.

A tabela 3, a seguir, apresenta os resultados dos experimentos para cada abordagem e cada missão, com o número de ensaios, tempo total em segundos, número de colisões, produtividade e a quantidade de vezes que o manipulador parou.

Observa-se que, a abordagem ponto - corpo extenso apresentou a melhor produtividade, com o menor número de colisões, tanto para a missão ponto quanto para zona. Nesta mesma abordagem o manipulador não parou nenhuma vez. A abordagem corpo extenso - corpo extenso apresentou uma baixa produtividade, porém para a missão zona ocorreu apenas uma colisão.

Tabela 3 – Resultados dos experimentos na implementação do algoritmo de CP para cada abordagem e cada missão

<b>Abordagem</b>	<b>Missão: ponto para ponto</b>				
	Nº de ensaios	Tempo total (s)	Nº de colisões	Produtividade	Parou (evitou colisão)
Corpo extenso - Corpo extenso	27	182,42	3	20	4
Corpo extenso - Ponto	29	171,04	6	18	5
Ponto - Corpo extenso	36	230,36	5	31	0
Ponto - Ponto	35	198,17	6	29	0
<b>Abordagem</b>					
<b>Missão: ponto para zona</b>					
Corpo extenso - Corpo extenso	32	207,96	1	27	4
Corpo extenso - Ponto	29	146,70	3	23	3
Ponto - Corpo extenso	35	213,22	3	32	0
Ponto - Ponto	35	210,92	7	28	0

Fonte: elaborada pelo autor.

A quantidade de vezes que o manipulador parou para evitar colisão, significa que a missão não foi concluída, mas não houve colisão. Isto está relacionado a execução da trajetória do manipulador, ou seja, existe um tempo para a trajetória ser realizada, mas devido a parada do manipulador esse tempo é estourado e o manipulador retorna a posição inicial para iniciar uma nova missão. A parada do manipulador está ligada ao fato do mesmo ter um menor espaço livre de colisão. Isto acontece durante a implementação das abordagens em que o manipulador é considerado corpo extenso, pois o espaço de configurações de obstáculo (QO) é maior que nas abordagens em que o mesmo é considerado como um ponto.

### 5.1.2 Avaliação do algoritmo de campos potenciais

Aqui, apresentamos a avaliação para cada abordagem e cada missão. Para isto foram calculados os valores da eficiência e os valores da variância do tempo. A tabela 4 mostra os valores da eficiência, calculados a partir da equação 4.3, para cada abordagem e missão dos ensaios e a tabela 5 mostra os valores da variabilidade do tempo obtidos através da equação 4.4.

Tabela 4 – Valores da eficiência do algoritmo de CPA para as abordagens ensaiadas

<b>Abordagem</b>	<b>Missão</b>	
	Ponto para ponto	Ponto para zona
Corpo extenso - Corpo extenso	1,333	6,921
Corpo extenso (manipulador) - Ponto	0,704	5,614
Ponto (manipulador) - Corpo extenso	8,324	12,465
Ponto - Ponto	3,491	3,303

Fonte: elaborada pelo autor.

Tabela 5 – Valores da variabilidade do tempo da missão

<b>Abordagem</b>	<b>Missão</b>	
	Ponto para ponto	Ponto para zona
Corpo extenso - Corpo extenso	4,989	3,901
Corpo extenso (manipulador) - Ponto e Ponto (manipulador) - Corpo extenso	4,258 0,744	1,365 0,855
Ponto - Ponto	1,384	1,210

Fonte: elaborada pelo autor.

Observa-se que nas abordagens em que o manipulador é considerado um corpo extenso a eficiência é menor (ver tabela 4), pois aumenta o espaço de configurações de colisão do manipulador com o robô móvel. Nessa abordagem, é levado em consideração a cadeia de elos do manipulador e isso diminui o espaço de trabalho livre de colisões, causando dificuldades na geração do caminho, como pode ser observado na tabela 3. Aconteceram situações em que o manipulador parou para evitar a colisão, não concluindo a missão.

Quando o manipulador é considerado um ponto tem-se uma melhoria na eficiência (ver tabela 4), comparado quando o mesmo é considerado um corpo extenso, como pode ser vistos na tabela 3. Neste caso tem-se um aumento do espaço de trabalho do manipulador, pois não é levado em consideração os elos do mesmo, consequentemente os caminhos gerados são melhores, ou seja, são otimizados.

A eficiência do algoritmo apresentou melhor resultado na abordagem ponto - corpo extenso (ver tabela 4). Nesta situação, como citado anteriormente, o manipulador tem um maior espaço de trabalho e o obstáculo tem uma maior área de repulsão, levando o algoritmo a otimizar os caminhos sem colisão.

Na abordagem ponto - ponto, o manipulador também é considerado um ponto, mas como o obstáculo também é um ponto, diminui a repulsão entre o manipulador e o robô móvel, aumentando o número de colisões, com isso houve uma queda na eficiência.

Os valores da variabilidade do tempo influem diretamente na eficiência. Observa-se que menores valores de variância tem-se um aumento da eficiência (ver tabela 5), isso significa que o manipulador executou as missões em tempos constantes, quer dizer que, em cada missão cumprida os valores de tempo foram aproximados.

### 5.1.3 Avaliação do algoritmo de campos potenciais através do OEE

Aqui são apresentados os valores da disponibilidade, performance, qualidade e OEE para as abordagens aplicadas no algoritmo de campos potenciais implementado no manipulador.

A tabela 6, a seguir, apresenta os valores de disponibilidade, performance, qualidade e OEE para as abordagens entre o manipulador e robô móvel, aplicando o algoritmo de campos potenciais na prevenção de colisão.

**Tabela 6 – Valores de disponibilidade, performance, qualidade e OEE para as abordagens**

<b>Abordagem</b>	<b>Missão: Ponto para ponto (valores em %)</b>			
	Disponibilidade	Performance	Qualidade	OEE
Corpo extenso - Corpo extenso	59,08	74,92	100,00	44,27
Corpo extenso - Ponto	60,00	70,84	100,00	42,71
Ponto - Corpo extenso	76,99	87,64	100,00	67,47
Ponto - Ponto	42,17	88,37	100,00	37,27
<b>Abordagem</b>	<b>Missão: Ponto para zona (valores em %)</b>			
	Disponibilidade	Performance	Qualidade	OEE
Corpo extenso - Corpo extenso	40,62	88,95	100,00	36,13
Corpo extenso - Ponto	56,62	84,16	100,00	47,65
Ponto - Corpo extenso	73,36	92,44	100,00	67,82
Ponto - Ponto	73,04	85,07	100,00	62,14

Fonte: elaborada pelo autor.

Outra verificação relacionada a eficiência, está no fato da implementação do algoritmo ser avaliado através da Eficiência Global do Equipamento (OEE - Overall Equipment Effectiveness). Através desse indicador é possível ter uma análise mais criteriosa da eficiência, visto que o mesmo é utilizado mundialmente para verificar a eficiência dos equipamentos. Importante ressaltar que, para calcular os valores do OEE aqui apresentados, foi realizada uma adaptação para que fosse possível a realização dos cálculos da disponibilidade, performance e qualidade. Outra observação é o fato do OEE representar aqui, a eficiência de cada abordagem implementada, não somente a eficiência do manipulador e com isto, ser possível, comparar o OEE com os resultados da equação 4.3, mostrados na tabela 4.

A abordagem que apresentou um melhor resultado foi a que levou em consideração o manipulador como um ponto e o obstáculo como corpo extenso, ou seja a abordagem ponto-corpo extenso. Para a missão ponto e zona os valores do OEE foram iguais a 67,47% e 67,82%, respectivamente, como mostrado na tabela 6. Estes valores são considerados satisfatórios, pois de acordo com Hansen (2006), o valor do OEE entre 65% e 75% é considerado aceitável, apesar do valor de Classe Mundial ser igual a 85,00%.

Nesta situação a missão em que o manipulador vai de um ponto para uma zona, ainda apresentou melhor resultado, tendo valores de disponibilidade, performance e qualidade iguais a 73,36%, 92,44% e 100,00%, respectivamente, como mostrado na tabela 6.

Dentro deste contexto, podemos afirmar que, a avaliação da eficiência do algoritmo, realizada aqui através da equação 4.3, é satisfatória para a análise aqui apresentada. Pois a melhor eficiência obtida pela equação 4.3, foi a melhor também na análise do OEE.

#### **5.1.4 Consumo de energia mecânica e análise do torque do motor da junta 1**

Para cada experimento/abordagem, foram calculados os valores do consumo de energia mecânica do motor 1. Escolheu-se somente o motor 1, pois o mesmo apresenta um maior consumo de energia, por ter uma potência mais elevada, maior momento de inércia e maior torque. A influência do motor 2 não interfere diretamente no consumo de energia, neste caso, optamos por não apresentar aqui.

A tabela 7, apresenta os valores do consumo de energia mecânica, em kWh, do motor 1 calculados a partir da equação 4.10, para as abordagens implementadas.

**Tabela 7 – Consumo de energia mecânica, em kWh, do motor 1 para as abordagens implementadas**

<b>Abordagem</b>	<b>Missão: ponto para ponto</b>
	Energia mecânica do Motor 1 (kWh)
Corpo extenso - Corpo extenso	0,0245
Corpo extenso - Ponto	0,0194
Ponto - Corpo extenso	0,0251
Ponto - Ponto	0,0249
<b>Abordagem</b>	<b>Missão: ponto para zona</b>
Corpo extenso - Corpo extenso	0,0223
Corpo extenso - Ponto	0,0124
Ponto - Corpo extenso	0,0277
Ponto - Ponto	0,0237

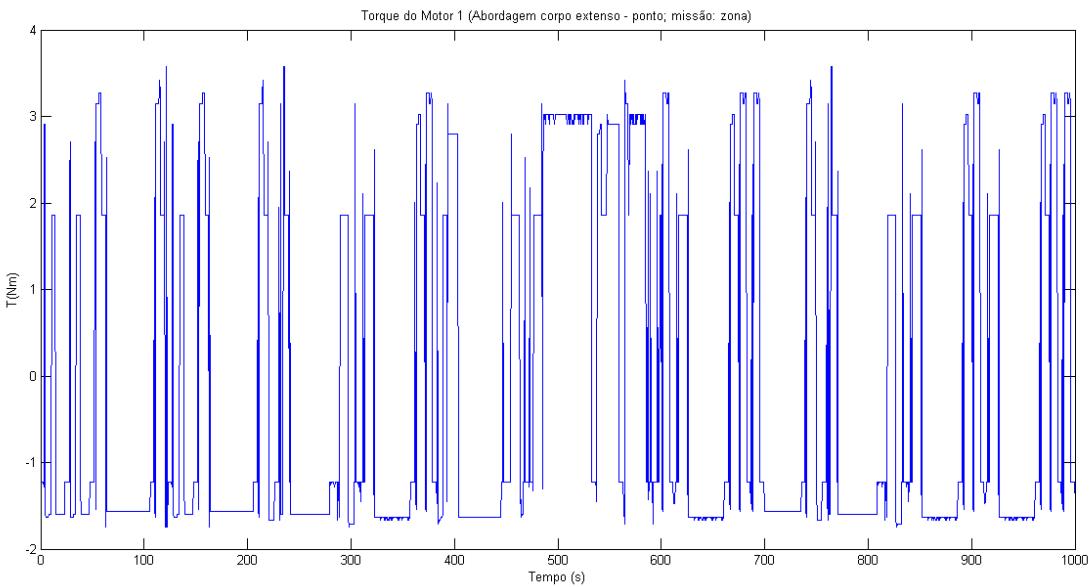
Fonte: elaborada pelo autor.

Podemos observar que o maior consumo de energia apresentou-se na abordagem ponto - corpo extenso, tanto para a missão ponto como para zona, isto deve-se ao fato de que o motor realizou um maior trabalho, pois essas abordagens foram as que a implementação do algoritmo apresentou melhor eficiência. Neste caso o motor foi mais solicitado para garantir uma maior produtividade. Podemos afirmar então que, uma maior produtividade e melhor eficiência resultou em um maior consumo de energia. Já para a abordagem corpo extenso - ponto, para a missão ponto e zona, o motor 1 teve um consumo menor de energia, isto implica em uma menor produtividade como mostra a tabela 4. Também os valores da eficiência do algoritmo, nesta abordagem, apresentaram-se baixos.

Apresentamos, a seguir, nas figuras 30 e 31, os gráficos com os valores de torque do motor 1, para as abordagens: corpo extenso - ponto, missão zona; e ponto corpo extenso, missão zona. Escolheu-se aqui, apresentar os valores dessas duas abordagens, pois foram as que apresentaram menor e maior consumo de energia, respectivamente, como pode ser visto na tabela 7.

Os valores dos torques nos permitem fazer uma análise da solicitação desses motores no desenvolvimento de cada abordagem. A solicitação está ligada diretamente ao esforço dos motores e ao consumo de energia.

Figura 30 – Torque do motor 1 para abordagem corpo extenso - ponto, missão: zona



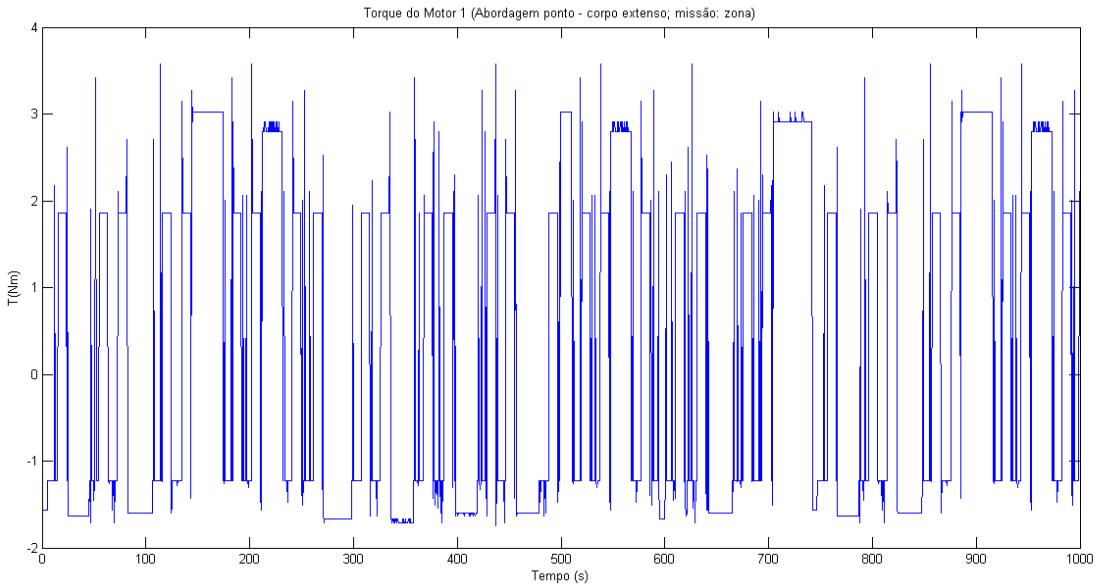
Fonte: autor.

Os valores dos torques foram obtidos através dos valores de referência de velocidade enviados do CLP para o Elipse SCADA e calculados a partir da equação 4.8, valores esses em Nm (Newton-metro).

Na figura 30 (abordagem corpo-extenso, missão zona), observa-se que os valores de torque têm uma menor variação de valores entre positivo e negativo, no mesmo intervalo de tempo, comparados com os valores apresentados na figura 31 (abordagem ponto-corpo extenso, missão zona). Isto significa que, na abordagem ponto-corpo extenso, missão zona o motor foi mais solicitado durante a realização das tarefas. Valores negativos de torque estão associado a mudança de sentido de rotação do motor. Por exemplo, se motor gira no sentido horário temos valores positivos e no sentido anti-horário valores negativos de torque.

Observamos que o motor 1 apresenta uma maior solicitação na abordagem em que o

Figura 31 – Torque do motor 1 para abordagem ponto - corpo extenso, missão: zona



Fonte: autor.

manipulador teve uma melhor produtividade, também uma melhor eficiência do algoritmo e um maior consumo de energia, que foi a abordagem ponto - corpo extenso, missão zona, como pode ser observado na figura 31.

Uma maior solicitação de torque significa um maior consumo de energia mecânica, como já apresentado e mostrado na tabela 7, maiores acelerações e frenagens durante a execução da tarefa. Essa análise é importante para associar as abordagens aqui apresentadas com: o consumo de energia mecânica, produtividade e eficiência do algoritmo.

### 5.1.5 Erro médio entre as posições geradas pelo algoritmo de CP e a posição real das juntas

O objetivo do cálculo do erro aqui é, verificar o quanto confiável está a relação entre o valor real das posições das juntas e o valor das posições geradas pelo algoritmo de campos potenciais. Os valores gerados pelo algoritmo são os *set-points*, que foram enviados para o controlador.

Os valores dos erros foram calculados a partir das posições das juntas do manipulador, isto é, posições no espaço de configurações (espaço das juntas), ou seja, as posições são os ângulos de cada junta. A tabela 8, abaixo, apresenta os valores do erro médio entre os ângulos do caminho gerado pelo algoritmo de CP (*set-points*) e os ângulos da posição real da junta, calculados a partir da equação 4.12, também são apresentados os valores do desvio padrão para cada medida de erro. São apresentados os valores dos erros e desvio padrão para a junta 1 e 2,

em cada abordagem e missão.

**Tabela 8 – Erro médio e desvio padrão entre os ângulos gerado pelo algoritmo de CP (*set-points*) e os ângulos da posição real das juntas do manipulador**

<b>Abordagem</b>	<b>Missão: ponto para ponto</b>	
	Erro médio das posições da junta 1	Erro médio das posições da junta 2
Corpo extenso - Corpo extenso	3,1004 ( $\pm$ 1,0175)	2,9796 ( $\pm$ 0,9957)
Corpo extenso - Ponto	0,9758 ( $\pm$ 1,6771)	3,2802 ( $\pm$ 1,1592)
Ponto - Corpo extenso	1,0868 ( $\pm$ 1,3603)	4,5680 ( $\pm$ 1,8991)
Ponto - Ponto	1,4581 ( $\pm$ 0,9856)	4,0907 ( $\pm$ 2,5936)
<b>Abordagem</b>	<b>Missão: ponto para zona</b>	
	3,1771 ( $\pm$ 1,0277)	3,5030 ( $\pm$ 1,0524)
Corpo extenso - Ponto	0,6926 ( $\pm$ 1,2004)	3,0091 ( $\pm$ 1,1583)
Ponto - Corpo extenso	1,1674 ( $\pm$ 1,2096)	4,0372 ( $\pm$ 2,4083)
Ponto - Ponto	1,5921 ( $\pm$ 1,3968)	4,3822 ( $\pm$ 2,4701)

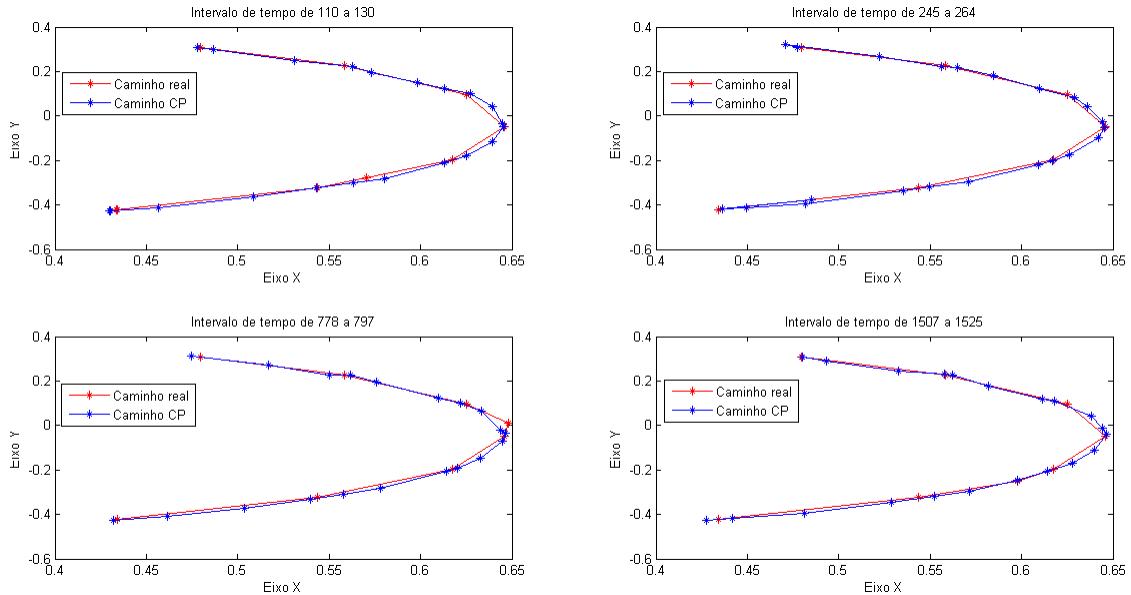
Fonte: elaborada pelo autor.

Para cada ensaio, foram calculados os valores de erro médio, a partir da equação 4.12, durante a realização do experimento de cada abordagem. Ou seja, para cada missão (ponto inicial ao ponto final) realizou-se o cálculo do erro médio e a partir do conjunto dos valores de erro, obtido durante a realização do experimento de cada abordagem, calculou-se a média e o desvio padrão, como apresentado na tabela 8. Isso permite uma análise mais criteriosa e mais precisa dos valores de erro entre as posições reais do manipulador e os valores das posições geradas pelo algoritmo de CP.

Para uma melhor visualização do erro, mostramos nas figuras 32 e 33, a seguir, o caminho gerado pelo algoritmo de campos potenciais e o caminho real do manipulador para a abordagem corpo extenso - ponto, missão zona e a abordagem ponto - corpo extenso, missão ponto, respectivamente.

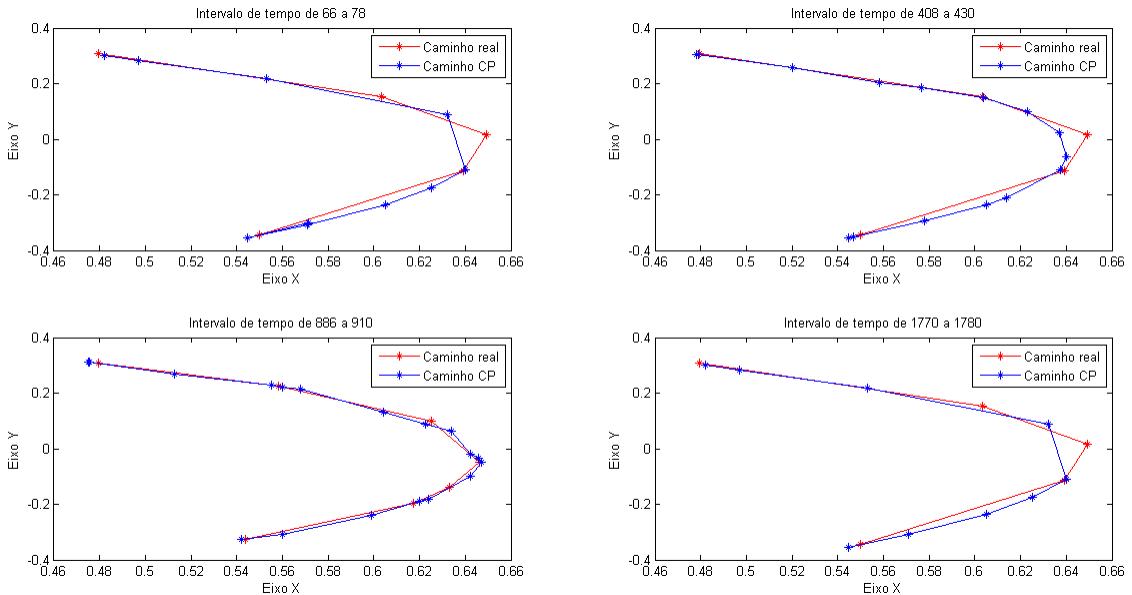
Os caminhos apresentados, estão no espaço cartesiano, pois permitem uma melhor visualização que no espaço das juntas. Para apresentar os caminhos no espaço cartesiano foi necessário calcular a cinemática direta através das equações 2.11 e 2.12.

Figura 32 – Caminho gerado pelo algoritmo de CPA x caminho real no espaço cartesiano para a abordagem corpo extenso - ponto, missão zona



Fonte: autor.

Figura 33 – Caminho gerado pelo algoritmo de CPA x caminho real no espaço cartesiano para a abordagem ponto - corpo extenso, missão ponto



Fonte: autor.

A figura 32 apresenta os caminhos com menor erro em relação aos caminhos da figura 33. Ou seja, a abordagem corpo extenso-ponto, missão zona, apresentou melhor resultado que a ponto-corpo extenso, missão ponto, isso em relação aos valores de erro médio, como pode ser verificado na tabela 8.

Calcular o erro é muito importante para verificar o comportamento do gerador de caminho (CPA) e as posições reais do manipulador, além de apresentar uma análise estatística do

processo. Isto é, verificar a geração de caminho através do algoritmo de campos potenciais, com obstáculo dinâmico, em posições aleatórias, para prevenir colisões em tempo real.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentada uma proposta de algoritmo campos potenciais artificiais, aplicado em um estudo de caso da interação máquina-máquina, para a geração de caminho livre de colisão, com obstáculo dinâmico, em tempo real. Aqui são apresentadas as conclusões do trabalho.

### 6.1 Contribuições

Neste trabalho, buscou-se mostrar a implementação do algoritmo de campos potenciais, em um estudo de caso, na geração de caminho livre de colisão entre um manipulador robótico e um robô móvel (obstáculo com posições aleatórias), com isso fazendo uma interação máquina-máquina e realizou-se uma análise os resultados.

Esse trabalho tem como contribuição principal apresentar os resultados da implementação do algoritmo de campos potenciais, na geração de caminho livre de colisão, em um manipulador industrial, em um ambiente dinâmico, monitorado por um sensor de imagens, em tempo real, onde ocorre a interação máquina-máquina.

Foram realizadas os experimentos, mostrando qual a melhor situação (abordagem) da utilização do algoritmo de campos potenciais, através de duas formas: a eficiência que propomos e através do indicador OEE. Também foi apresentado o consumo de energia, além da análise estatística do erro entre as posições geradas pelo algoritmo de campos potenciais e as posições reais do manipulado.

Com a realização deste trabalho foi possível desenvolver uma célula de manufatura inteligente, onde é possível a análise de caminhos e trajetórias livres de colisão.

### 6.2 Conclusões do trabalho

Podemos destacar que este trabalho apresentou uma proposta de implementação do algoritmo de campos potenciais levando em consideração várias abordagens, que podemos considerar satisfatórias neste estudo de caso, do ponto de vista da interação máquina-máquina.

Uma outra observação importante da implementação do algoritmo, é que foi feita a abordagem que leva em consideração toda a cadeia de elos do manipulador, isto tem uma grande vantagem, pois em muitos trabalhos considera-se apenas como um ponto.

Como o estudo de caso fez-se uso de um manipulador robótico acionado por um

CLP, um robô móvel; a fim de validar os resultados, foi implementado o campo potencial para gerar o caminho livre de colisão, tornando o sistema um célula de manufatura inteligente, no ponto de vista da prevenção de colisão e da interação máquina-máquina.

Outra análise realizada foi a eficiência do algoritmo de campos potenciais através do OEE, que se mostrou satisfatória, pois os valores obtidos estão dentro da faixa considerada aceitável. Comparamos também, com a análise realizada aqui, através da equação 4.3, confirma-se que esta equação satisfaz a avaliação do algoritmo, pois a abordagem que apresentou melhor eficiência, foi também a melhor, através da análise do OEE.

Para complementação deste trabalho foi realizado o cálculo do consumo de energia durante os ensaios. Este cálculo, nos permite comparar a implementação do algoritmo com o consumo de energia, a eficiência e a produtividade. Também foi realizada a análise estatística do erro entre as posições geradas pelo algoritmo de campos potenciais e as posições reais do manipulador.

Por último, podemos concluir que todos os objetivos propostos foram alcançados de maneira satisfatória. Na seção a seguir são apresentadas as sugestões de trabalhos futuros.

### **6.3 Sugestões de trabalhos futuros**

Como trabalhos futuros, propõe-se a expansão deste trabalho para sistemas de maior complexidade permitindo melhorias na geração de caminhos e trajetórias livres de colisão em manipuladores industriais ou em ambientes com vários robôs. Dentre os quais podemos citar:

1. Implementar o algoritmo de campos potenciais no robô móvel;
2. Estudar como melhorar a produtividade se os parâmetros do algoritmo de CPA variarem;
3. Implementar algoritmos com técnicas baseada em PRM (*Probabilistic Roadmap Methods*) na prevenção de colisão;
4. Comparar a eficiência entre CPA e PRM na prevenção de colisão;
5. Utilizar técnicas de predição para prever o deslocamento do robô móvel e otimizar o caminho gerado pelo algoritmo de campos potenciais;
6. Utilizar técnicas que otimizem as constantes de atração e repulsão do campo potencial para otimizar o caminho;

## REFERÊNCIAS

- ANDERSSON, R. L. Aggressive trajectory generator for a robot ping-pong player. **IEEE Control Systems Magazine**, IEEE, v. 9, n. 2, p. 15–21, 1989.
- ANTONIO, B.; FELIPE, P. L.; CARLOS, B.; RAFAEL, A. Fundamentos de robótica. **España: Editorial McGraw-Hill**, 1997.
- ARNOT, A. Estatística fácil. **São Paulo: Saraiva**, 2002.
- ATAKA, A.; QI, P.; LIU, H.; ALTHOEFER, K. Real-time planner for multi-segment continuum manipulator in dynamic environments. **2016 IEEE International Conference on Robotics and Automation (ICRA)**, p. 4080–4085, 2016.
- BADAWY, A. Manipulator trajectory planning using artificial potential field. In: **IEEE. Engineering and Technology (ICET), 2014 International Conference on**. [S.I.], 2014. p. 1–6.
- BROOKS, R. A. Solving the find-path problem by good representation of free space. **IEEE Transactions on Systems, Man, and Cybernetics**, IEEE, n. 2, p. 190–197, 1983.
- BUDIYANTO, A.; CAHYADI, A.; ADJI, T. B.; WAHYUNGGORO, O. Uav obstacle avoidance using potential field under dynamic environment. In: **IEEE. Control, Electronics, Renewable Energy and Communications (ICCEREC), 2015 International Conference on**. [S.I.], 2015. p. 187–192.
- CHATRAEI, A.; JAVIDIAN, H. Formation control of mobile robots with obstacle avoidance using fuzzy artificial potential field. In: **IEEE. Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), 2015 IEEE International Workshop of**. [S.I.], 2015. p. 1–6.
- CHEN, F.; DI, P.; HUANG, J.; SASAKI, H.; FUKUDA, T. Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly. In: **IEEE. Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on**. [S.I.], 2009. p. 92–97.
- CHENG, C.; ZHU, D.; SUN, B.; CHU, Z.; NIE, J.; ZHANG, S. Path planning for autonomous underwater vehicle based on artificial potential field and velocity synthesis. In: **IEEE. Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on**. [S.I.], 2015. p. 717–721.
- CONNOLLY, C. I.; BURNS, J. B.; WEISS, R. Path planning using laplace's equation. In: **IEEE. Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. [S.I.], 1990. p. 2102–2106.
- CRAIG, J. J. **Robótica. 3<sup>a</sup> edição.** [S.I.]: São Paulo: Editora Pearson, 2012.
- DENAVIT, J. A kinematic notation for lower-pair mechanisms based on matrices. **Trans. of the ASME. Journal of Applied Mechanics**, v. 22, p. 215–221, 1955.
- FILHO, O. M.; NETO, H. V. **Processamento digital de imagens.** [S.I.]: Brasport, 1999.

- FITZGERALD, A. E.; JR, C. K.; UMANS, S. D. **Máquinas Elétricas: Com introdução à eletrônica de potência. Tradução de Anatólio Laschuk.** [S.l.]: Porto Alegre: Bookman, 2006.
- GALCERAN, E.; EUSTICE, R. M.; OLSON, E. Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving. In: IEEE. **2015 IEEE Intelligent Vehicles Symposium (IV).** [S.l.], 2015. p. 304–309.
- GAYLE, R.; SUD, A.; LIN, M. C.; MANOCHA, D. Reactive deformation roadmaps: motion planning of multiple robots in dynamic environments. In: IEEE. **Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.** [S.l.], 2007. p. 3777–3783.
- GONZALEZ, R. C. **Processamento digital de imagens. tradução: Cristina yamagami e leonardo piamonte.** [S.l.]: Pearson Prentice Hall, São Paulo, 2009.
- GUAN, W.; WENG, Z.; ZHANG, J. Obstacle avoidance path planning for manipulator based on variable-step artificial potential method. In: IEEE. **The 27th Chinese Control and Decision Conference (2015 CCDC).** [S.l.], 2015. p. 4325–4329.
- HABIB, M. K. **Handbook of Research on Advancements in Robotics and Mechatronics.** [S.l.]: IGI Global, 2014.
- HANSEN, R. C. Eficiência global dos equipamentos: uma poderosa ferramenta de produção/manutenção para o aumento dos lucros. **Tradução de Altair Flamarion klippeL Porto alegre: bookman,** 2006.
- HARDEN, T. A. **The implementation of artificial potential field based obstacle avoidance for a redundant manipulator.** [S.l.: s.n.], 1997.
- HARGAS, Y.; MOKRANE, A.; HENTOUT, A.; HACHOUR, O.; BOUZOUIA, B. Mobile manipulator path planning based on artificial potential field: Application on robuter/ulm. In: IEEE. **2015 4th International Conference on Electrical Engineering (ICEE).** [S.l.], 2015. p. 1–6.
- HARTENBERG, R. S.; DENAVIT, J. **Kinematic synthesis of linkages.** [S.l.]: McGraw-Hill, 1964.
- IFM, E. **Operating instructions PMD 3D sensor O3D200.** Tese (Doutorado), 2009.
- ISO, E. 9283: 1998. **Manipulating industrial robots-Performance criteria and related test methods,** 1998.
- ISO10218. **Manipulating industrial robots - Safety.** Tese (Doutorado), 2011.
- JORGE, V. A.; MAFFEI, R.; FRANCO, G. S.; DALTRIZZO, J.; GIAMBASTIANI, M.; KOLBERG, M.; PRESTES, E. Ouroboros: Using potential field in unexplored regions to close loops. In: IEEE. **2015 IEEE International Conference on Robotics and Automation (ICRA).** [S.l.], 2015. p. 2125–2131.
- KAZEMI, H.; PORSHOKOUIHI, P. N.; MASOULEH, M. T.; NOVIN, R. S. Path planning of 3-rrr parallel robot by avoiding mechanical interferences via artificial potential field. In: IEEE. **Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on.** [S.l.], 2015. p. 240–245.

- KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **The international journal of robotics research**, Sage Publications, v. 5, n. 1, p. 90–98, 1986.
- KOPPULA, H. S.; SAXENA, A. Anticipating human activities using object affordances for reactive robotic response. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 38, n. 1, p. 14–29, 2016.
- KROGH, B. H. **A generalized potential field approach to obstacle avoidance control**. [S.l.]: RI/SME, 1984.
- LAGES, W. F. Geração de trajetórias. **Departamento de Engenharia Elétrica, UFRGS**, 2005.
- LATOMBE, J.-C. Robot motion planning (the kluwer international series in engineering and computer science). Kluwer Academic, 1990.
- LI, G.; TONG, S.; LV, G.; XIAO, R.; CONG, F.; TONG, Z.; YAMASHITA, A.; ASAMA, H. An improved artificial potential field-based simultaneous forward search (improved apf-based sifors) method for robot path planning. In: **IEEE. Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on**. [S.l.], 2015. p. 330–335.
- MAC, T. T.; COPOT, C.; HERNANDEZ, A.; KEYSER, R. D. Improved potential field method for unknown obstacle avoidance using uav in indoor environment. In: **IEEE. 2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)**. [S.l.], 2016. p. 345–350.
- MARCOTTI, A. G. **Aumento de produtividade em processo de uma indústria têxtil**. Tese (Doutorado) — EPUSP, 2006.
- MOELLMANN, A. H.; ALBUQUERQUE, A. S.; CONTADOR, J. L.; MARINS, F. A. S. Aplicação da teoria das restrições e do indicador de eficiência global do equipamento para melhoria de produtividade em uma linha de fabricação. **Revista gestão industrial**, v. 2, n. 1, 2006.
- NAKAJIMA, S. **Introdução ao TPM – Total Productive Maintenance**. [S.l.]: IMC Internacional Sistemas Educativos Ltda, 1989.
- ORIOLO, G. **A case study of safe human/robot coexistence, Research Proposal**. [S.l.]: DIAG Robotics Lab Sapienza University of Rome, 2015.
- PAUL, R. P. **Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators**. [S.l.]: Richard Paul, 1981.
- PEREIRA, M. J. **Engenharia de manutenção: teoria e prática**. [S.l.]: Editora Ciência Moderna, 2011.
- PEREIRA, N.; NUNES, J. *et al.* Planejamento de caminho livre de colisão para robô scara em ambiente monitorado por sensor de imagem. **Anais do XX Congresso Brasileiro de Automática**, v. 20, n. 20, p. 3946–3953, 2014.
- PINTO, L.; DAVIDSON, J.; GUPTA, A. Supervision via competition: Robot adversaries for learning tasks. **arXiv preprint arXiv:1610.01685**, 2016.

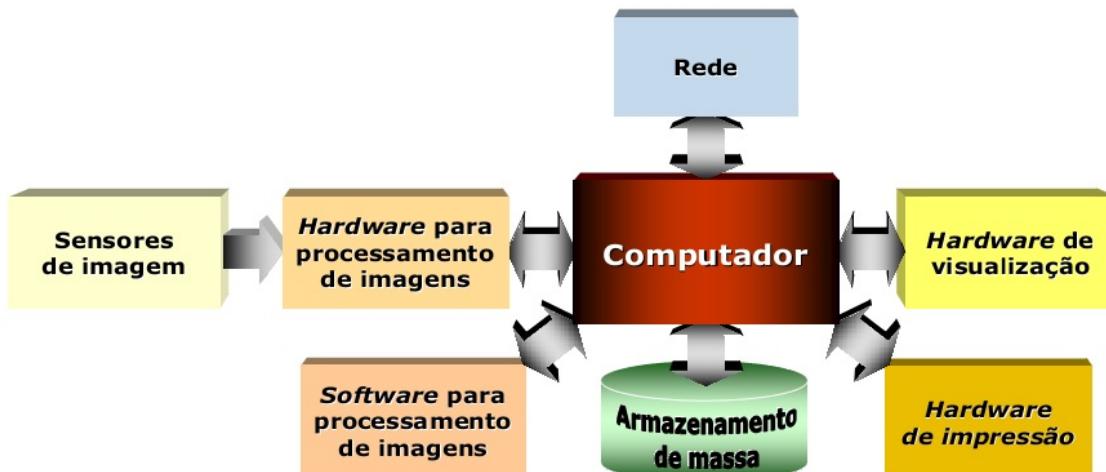
- PINTO, M. F.; MENDONÇA, T. R.; OLIVI, L. R.; COSTA, E. B.; MARCATO, A. L. Modified approach using variable charges to solve inherent limitations of potential fields method. In: IEEE. **Industry Applications (INDUSCON), 2014 11th IEEE/IAS International Conference on.** [S.I.], 2014. p. 1–6.
- QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. **RITA**, v. 13, n. 2, p. 11–42, 2006.
- ROMANO, V. F. **Robótica industrial: aplicação na indústria de manufatura e de processos.** [S.I.]: Edgard Blucher, 2002.
- SCIAVICCO, L.; SICILIANO, B. **Modeling and control of robot manipulators.** [S.I.]: McGraw-Hill New York, 1996. v. 8.
- SHILLER, Z.; DUBOWSKY, S. On computing the global time-optimal motions of robotic manipulators in the presence of obstacles. **IEEE Transactions on Robotics and Automation**, IEEE, v. 7, n. 6, p. 785–797, 1991.
- SHIN, K.; MCKAY, N. Minimum-time control of robotic manipulators with geometric path constraints. **IEEE Transactions on Automatic Control**, IEEE, v. 30, n. 6, p. 531–541, 1985.
- SILVA, J. L. N.; BATISTA, J. G. *et al.* Proposta de planejamento e controle de trajetória para manipuladores industriais. **Anais do XX Congresso Brasileiro de Automática**, v. 20, n. 20, p. 3954–3961, 2014.
- SILVA, L. J. N. **Projeto e Realização de Controle Fuzzy para Manipulador Industrial Tipo SCARA e sua Avaliação de Desempenho à Luz da Norma ISO 9283.** [S.I.]: Dissertação submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de Mestre em Engenharia de Teleinformática., 2016.
- SPONG, M. W. **Robot modeling and control.** [S.I.]: Wiley New York, 2006. v. 3.
- STAMATIS, D. H. **The OEE primer: understanding overall equipment effectiveness, reliability, and maintainability.** [S.I.]: CRC Press, 2010.
- VOLPE, R.; KHOSLA, P. Manipulator control with superquadric artificial potential functions: Theory and experiments. **IEEE Transactions on Systems, Man, and Cybernetics**, IEEE, v. 20, n. 6, p. 1423–1436, 1990.
- WANG, X.; JIN, Y.; DING, Z. A path planning algorithm of raster maps based on artificial potential field. In: IEEE. **Chinese Automation Congress (CAC), 2015.** [S.I.], 2015. p. 627–632.
- YU, K.; OHNISHI, K.; KAWANA, H.; USUDA, S. Modulated potential field using 5 dof implant assist robot for position and angle adjustment. In: IEEE. **Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE.** [S.I.], 2015. p. 002166–002171.
- YU, K.; UOZUMI, S.; OHNISHI, K.; USUDA, S.; KAWANA, H.; NAKAGAWA, T. Stereo vision based robot navigation system using modulated potential field for implant surgery. In: IEEE. **Industrial Technology (ICIT), 2015 IEEE International Conference on.** [S.I.], 2015. p. 493–498.

## APÊNDICE A – FUNDAMENTAÇÃO TEÓRICA DE PDI

### A.1 Processamento digital de imagens

Os sistemas para processamento de imagens têm sido propostos e comercializados, em vários modelos, no mundo inteiro nas duas últimas décadas. Um sistema de processamento de imagens de uso genérico, apresenta os seguintes componentes representados pelo diagrama de blocos na figura 34. Este diagrama representa desde sistemas de baixo custo até sofisticadas estações de trabalho utilizadas em aplicações que envolvem intenso uso de imagens.

Figura 34 – Elementos de um sistema de processamento de imagens



Fonte: Queiroz (2006).

O diagrama da figura 34 abrange as principais operações que se pode efetuar sobre uma imagem, a saber: aquisição, processamento, armazenamento e visualização. Além disso, uma imagem pode ser transmitida à distância utilizando meios de comunicação disponíveis (FILHO; NETO, 1999), (QUEIROZ; GOMES, 2006). Todas estas operações são descritas a seguir:

#### *Aquisição*

A etapa de aquisição, também conhecida como sensoriamento, é composta por dois elementos: o dispositivo sensível à faixa de energia irradiada pelo alvo e o dispositivo conversor da saída de sensoriamento em formato digital. Essa etapa tem como função converter uma imagem em uma representação numérica adequada para o processamento digital subsequente. O diferencial do hardware nessa etapa está na velocidade de processamento em operações que

requerem transferências rápidas de dados da entrada para a saída.

### *Processamento*

Nessa etapa é necessário um computador, cuja função é processar um algoritmo que executa procedimentos para tratamento da imagem de acordo com uma aplicação específica. O poder de processamento do computador depende da capacidade de processamento exigido pela tarefa. Em função disto, com exceção das etapas de aquisição e exibição, a maioria das funções de processamento de imagens pode ser implementada via software. O uso de hardware especializado para processamento de imagens somente será necessário em situações nas quais certas limitações do computador principal (por exemplo, velocidade de transferência dos dados através do barramento) forem intoleráveis.

### *Armazenamento*

O armazenamento de imagens tem se tornado um dos grandes desafios para a área de PDI, uma vez que os sistemas de aquisição vêm sendo cada vez mais aprimorados para capturar uma grande quantidade de dados, o que requer dispositivos com capacidades de armazenamento cada vez maiores, além de taxas de transferência de dados mais elevadas e maiores índices robustez e confiabilidade do processo de armazenamento. Este armazenamento pode ser dividido em três categorias: (1) armazenamento de curta duração de uma imagem, enquanto ela é utilizada nas várias etapas do processamento, (2) armazenamento de massa para operações de recuperação de imagens relativamente rápidas, e (3) arquivamento de imagens, para recuperação futura quando isto se fizer necessário.

### *Visualização*

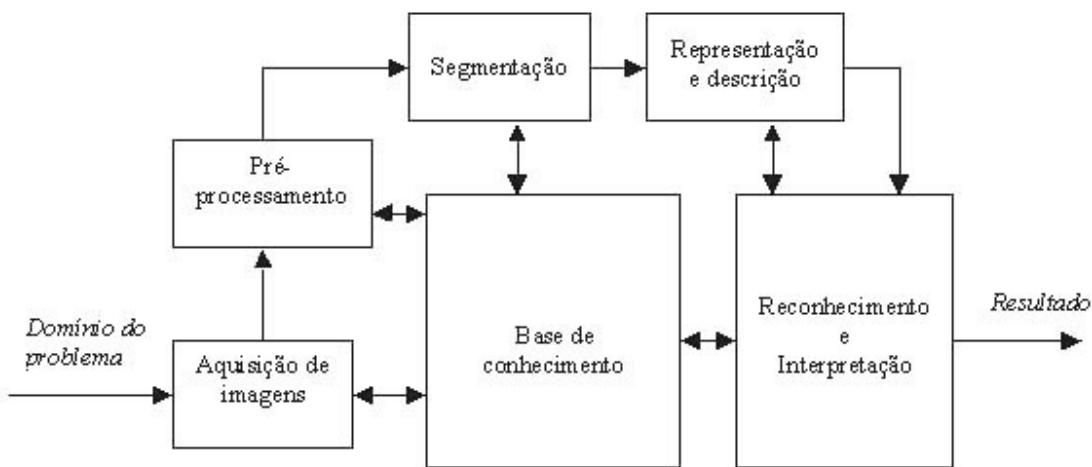
A visualização é uma etapa de saída do processamento de imagens que consiste em duas opções: a visualização dos dados em monitores de vídeo coloridos preferencialmente de tela plana de LED ou LCD, que recebem dados de placas gráficas comerciais ou dedicadas; e através da impressão que utiliza-se de dispositivos de impressão de pequeno, médio e grande porte - impressoras e/ou traçadores gráficos (plotters) de jato de tinta, sublimação de cera ou laser.

Imagens digitalizadas podem ser transmitidas à distância utilizando redes de compu-

tadores como a internet e protocolos de comunicação já existentes.

O PDI abrange uma ampla escala de hardware, software e fundamentos teóricos. A figura 35 mostra que o objetivo global é produzir um resultado a partir do domínio do problema por meio do processamento de imagens (GONZALEZ, 2009). Para o processamento de imagens existem softwares para que consiste, em geral, de módulos destinados à realização de tarefas específicas de acordo com a aplicação.

Figura 35 – Passos fundamentais em processamento de imagens digitais



Fonte: Gonzalez (2009).

#### A.1.1 Processamento morfológico de imagens

Processamento morfológico de imagens é uma modelagem destinada à descrição ou análise da forma de um objeto digital. O modelo morfológico para a análise de imagens fundamenta-se na extração de informações a partir de transformações morfológicas, nos conceitos da álgebra booleana e na teoria dos conjuntos e reticulados. A morfologia digital de imagens baseia-se no de que uma imagem é um conjunto de pontos elementares (pixels), que formam subconjuntos elementares bi ou tridimensionais. Os subconjuntos e a inter-relação entre eles formam estruturalmente a morfologia da imagem (QUEIROZ; GOMES, 2006). A morfologia é utilizada como ferramenta para extração de componentes de imagens que são indispensáveis na representação e descrição da forma de uma região ou fronteira.

Duas operações consideradas fundamentais para o processamento morfológico de imagens são a erosão e a dilatação. A maioria dos algoritmos morfológicos se baseiam nestas duas operações. Elas podem ser utilizadas para remoção de componentes, extração de fronteiras

de imagens, filtragem morfológica, extração de componentes conexos e outros algoritmos (GONZALEZ, 2009).

#### A.1.1.1 Erosão

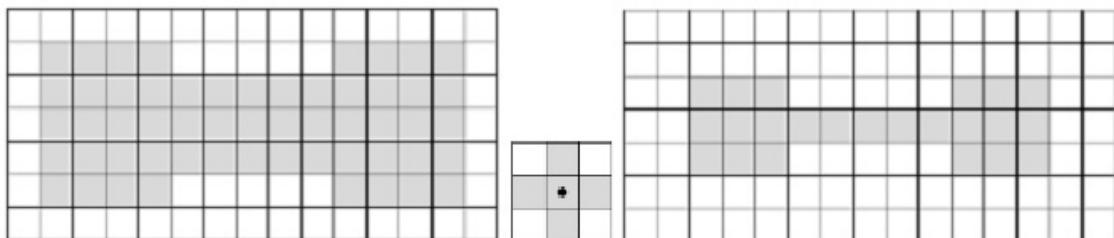
Com A e B como conjunto de  $Z^2$ , a erosão de A por B é definida como mostra a equação A.1.

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (\text{A.1})$$

Onde B é um elemento estruturante. A equação A.1 indica que a erosão de A por B é o conjunto de todos os pontos z de forma que B, transladado por z, está contido em A. Ou podemos dizer que, o elemento B terá seu centro transladado por todos os pixels do elemento A e a cada iteração, caso o elemento B esteja totalmente contido no elemento A, o pixel central do elemento B referente a iteração será considerado um dos pixels presentes na imagem resultante da erosão (GONZALEZ, 2009).

A figura 36 mostra a operação de erosão entre dois elementos A e B e o resultado obtido.

Figura 36 – Operação de erosão de A (esquerda) por B (central) resultando na imagem a direita



Fonte: Pereira (2014).

#### A.1.1.2 Dilatação

Com A e B como conjuntos de  $Z^2$ , a dilatação de A por B é definida como mostra a equação A.2.

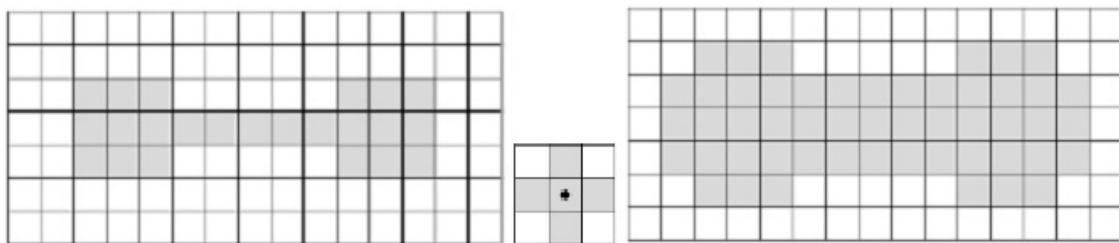
$$A \oplus B = \{z | (B_R)_z \cap A \neq \emptyset\} \quad (\text{A.2})$$

Onde  $B_R$  é a reflexão do elemento estruturante em torno de sua origem. A equação A.2 indica que a dilatação de A por B é o conjunto de todos os deslocamentos em z de forma que

$B_R$  e A sobreponham pelo menos por um elemento (GONZALEZ, 2009). Assim, o elemento  $B_R$  terá seu centro transladado por todos os pixels do elemento A e a cada iteração, caso o elemento  $B_R$  possua pelo menos um pixel em intersecção com o elemento A, o pixels central do elemento  $B_R$  referente a iteração será considerado um dos pixels presentes na imagem resultante da dilatação.

A figura 37 mostra a operação de dilatação entre dois elementos A e B e o resultado obtido.

Figura 37 – Operação de dilatação de A (esquerda) por B (central) resultando na imagem a direita



Fonte: Pereira (2014).

### A.1.2 Segmentação de imagens

Na execução de processos de análise de imagens, faz-se necessária a extração de medidas, características ou informação de uma dada imagem por métodos automáticos ou semi-automáticos.

A segmentação consiste na subdivisão da imagem em partes ou em regiões ou objetos que a compõem e ela deve ser aplicada até quando os objetos ou as regiões de interesse de uma aplicação forem detectados. A precisão da segmentação determina a qualidade final dos procedimentos de análise computadorizada. Em geral, os algoritmos de segmentação se baseiam em propriedades básicas dos valores de intensidade: descontinuidade ou similaridade. Para o primeiro, o método é dividir a imagem em regiões de acordo com mudanças bruscas de intensidade, como as bordas (GONZALEZ, 2009).

Na similaridade, os algoritmos se baseiam na divisão da imagem a partir do reconhecimento de regiões semelhantes de acordo com critérios predefinidos. A limiarização, o crescimento de regiões, a divisão e a fusão de regiões são exemplos de métodos desta abordagem (GONZALEZ, 2009).

#### A.1.2.1 Limiarização e o método de Otsu

Limiarização é uma abordagem para a segmentação fundamentada na análise da similaridade de níveis de cinza, de modo a extrair objetos de interesse mediante a definição de um limiar que separe os agrupamentos de níveis de cinza da imagem. Uma das dificuldades do processo reside na determinação do valor mais adequado de limiarização, que é a separação dos pixels da imagem considerada. Devido sua simplicidade de implementação e velocidade computacional, a limiarização de imagens possui grande utilização em aplicações de segmentação de imagens (QUEIROZ; GOMES, 2006).

Quando as distribuições de intensidade dos pixels de fundo e dos objetos de uma imagem são suficientemente diferentes, é possível utilizar um único limiar aplicável a toda imagem. Nestes casos, a limiarização se dá a partir da obtenção de um valor limiar no histograma da imagem para o qual valores acima deste limiar possam se associar a uma classe (objeto ou fundo) e para valores abaixo do limiar possa se associar a classe restante (GONZALEZ, 2009).

Uma possibilidade para a seleção do valor de limiar é através de inspeção, já que muitas vezes a partir da análise do histograma da imagem a escolha do limiar que consiga segmentar objeto de fundo é intuitiva, porém este método impede a utilização de tratamentos automáticos de imagens.

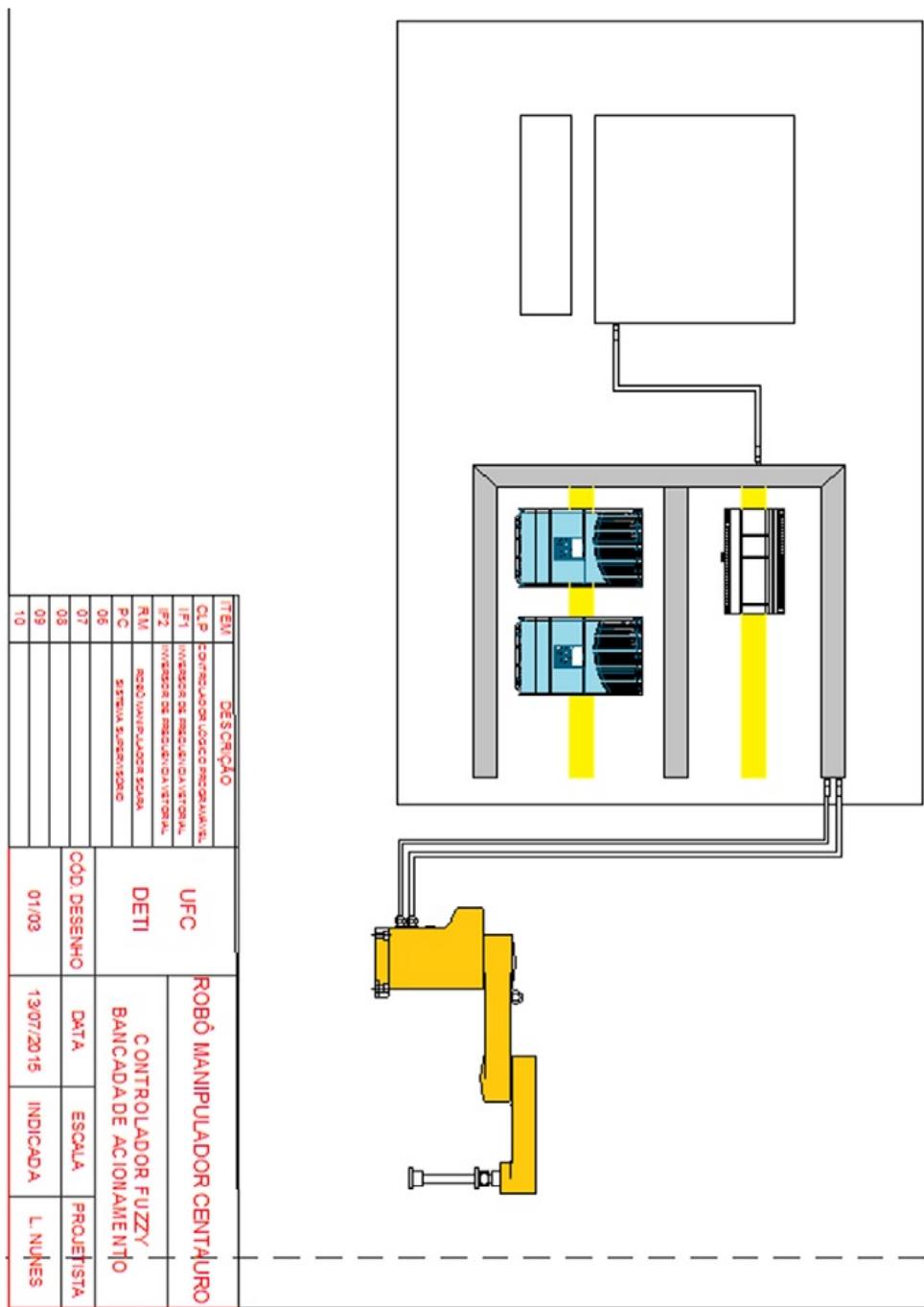
O método de Otsu se apresenta como uma solução para escolha de limiar em uma imagem de tons de cinza, no sentido de maximizar a variância entre as diferentes classes. Este método se baseia em obter um limiar que ofereça a melhor separação entre classes levando em conta os valores de intensidade dos pixels, uma outra qualidade do método é que se baseia somente no histograma da imagem (GONZALEZ, 2009).

Na prática, a utilização do método de Otsu é feita iterando o valor do nível de limiar dentre todos os valores possíveis na imagem e para cada iteração calcular a variância entre as classes. O valor de limiar que apresentar a maior variância entre classes será escolhido como o valor ótimo para limiar global da imagem (GONZALEZ, 2009).

## APÊNDICE B – SISTEMA E EQUIPAMENTOS

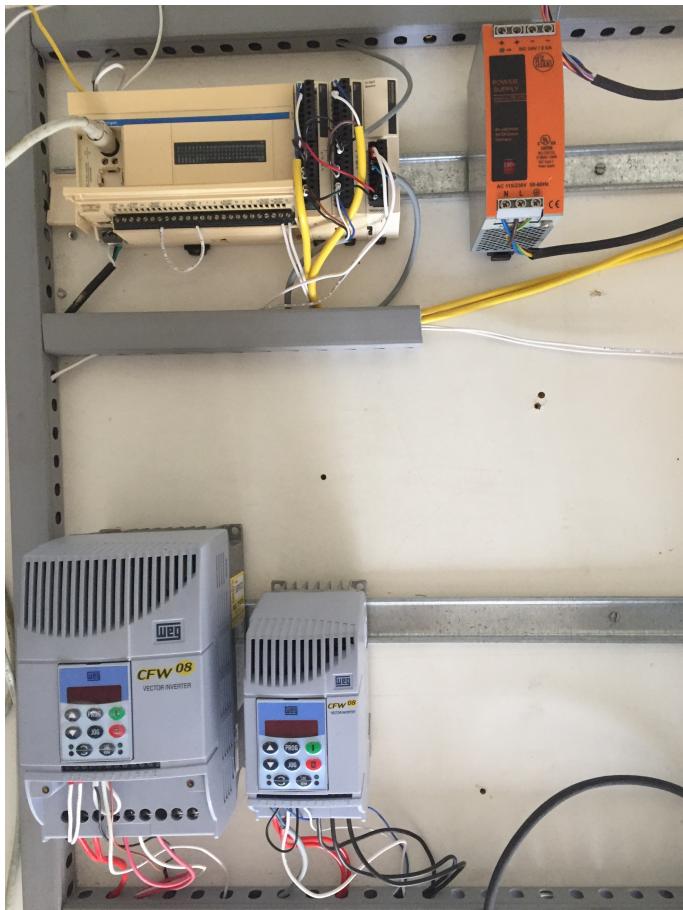
A seguir são apresentadas algumas imagens dos equipamentos e do sistema onde foram realizados os ensaios da implementação do algoritmo de campos potenciais.

Figura 38 – Bancada de Acionamento e Controle do Manipulador SCARA



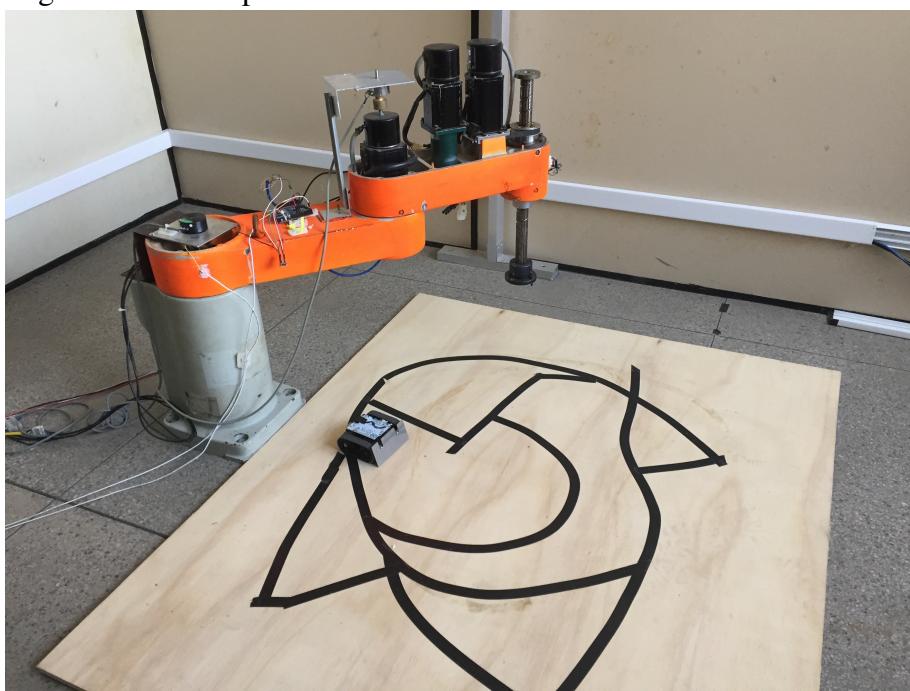
Fonte: Silva (2015).

Figura 39 – Imagem da Bancada de Acionamento e CLP



Fonte: autor.

Figura 40 – Manipulador e robô móvel



Fonte: autor.

Figura 41 – Montagem do sensor de imagens



Fonte: autor.

Figura 42 – Sistema com o sensor de imagens, manipulador e robô móvel



Fonte: autor.

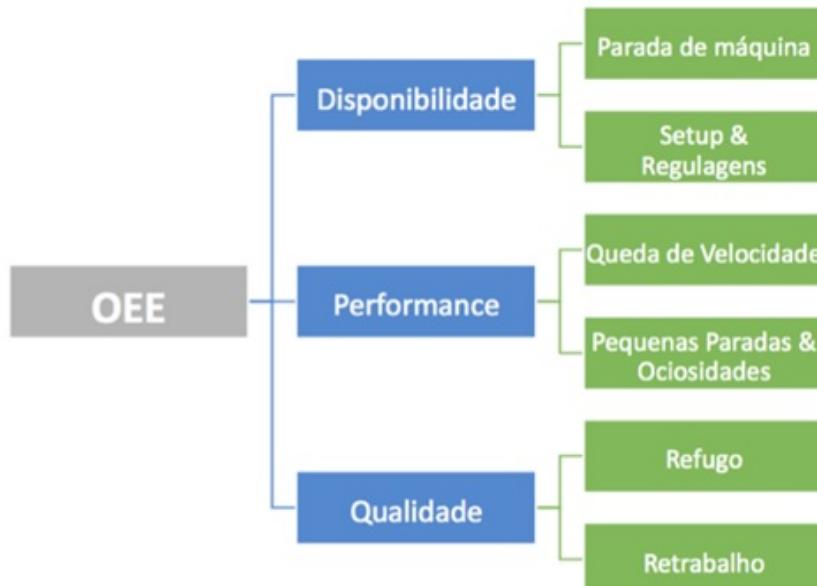
## APÊNDICE C – OEE - OVERALL EQUIPMENT EFFECTIVENESS

### C.1 Eficiência Global do Equipamento

O índice de eficiência global de equipamentos (OEE - Overall Equipment Effectiveness), tem origem no TPM - Manutenção Produtiva Total (Total Productive Maintenance) é um sistema de medição de manufatura que busca revelar os custos escondidos na corporação, é utilizado para identificar as áreas que necessitam de melhorias, bem como serve como referência para quantificar as melhorias obtidas nos equipamentos (NAKAJIMA, 1989).

O OEE envolve os índices de performance, disponibilidade e qualidade, utilizados para eliminar as seis grandes perdas, conforme mostrado na figura 43, a seguir

Figura 43 – Variáveis do OEE



Fonte: Prodwin Tecnologias (2017).

#### C.1.1 Disponibilidade (*Índice de tempo operacional*)

O índice de disponibilidade está relacionado às duas primeiras perdas, que são: parada da máquina por quebra ou falha do equipamento, e setup, ajustes e regulagens para troca do modelo a ser fabricado.

É o percentual do tempo que o equipamento estava trabalhando, comparado ao total

do tempo disponível para ser utilizado (MARCOTTI, 2006).

$$Disp.(%) = \frac{TTD. - PP - PNP}{TTD - PP} * 100 \quad (C.1)$$

Onde, TTD é o tempo total disponível, PP é tempo para paradas planejas e PNP é o tempo para as paradas não planejadas.

A disponibilidade mede a influência das perdas por manutenção e ajustes ou setups (PEREIRA, 2011).

### **C.1.2 Performance (Índice de Desempenho Operacional)**

O índice de performance é influenciado somente pela velocidade reduzida do equipamento, ou seja, quando o equipamento está trabalhando abaixo da velocidade em que foi especificado, e por ociosidade e pequenas paradas onde o próprio operador faz a correção do equipamento (STAMATIS, 2010).

Apresenta uma relação entre o total de peças produzidas real e teórico, levando em consideração o tempo de ciclo. Ou seja, avalia o ritmo de produção do equipamento.

$$Perf.(%) = \frac{TEO}{TO} * 100 \quad (C.2)$$

Onde, TEO é o tempo efetivo de operação e TO é o tempo de operação.

A performance (desempenho), mede a influência das perdas por ociosidade ou pequenas interrupções e velocidade de trabalho reduzido, isto é, a máquina operando com capacidade abaixo do normal (PEREIRA, 2011).

### **C.1.3 Qualidade (Índice de Produtos Aprovados)**

O índice de qualidade contempla as perdas por problemas de qualidade, ou seja, quando o equipamento produz peças defeituosas que são os refugos e os retrabalhos e por queda de rendimento ou startup, que é o tempo que o equipamento leva para retomar ao seu regime normal depois de um tempo de paradas (STAMATIS, 2010).

Refere-se à geração de produtos defeituosos, que resultam em refugo ou retrabalhos.

$$Qual.(%) = \frac{TPP - TRR}{TPP} * 100 \quad (C.3)$$

Onde, TPP é o total de peças produzidas,e TRR é o total de refugos e retrabalhos.

A qualidade mede a influência das perdas por peças fora das especificações e queda de rendimento.

#### **C.1.4 Cálculo do OEE**

O indicador OEE é o produto dos três índices relacionados acima, disponibilidade, performance e qualidade. Este índice pode ser obtido através da equação C.4a seguir.

$$OEE(\%) = Disp. * Perf. * Qual. * 100 \quad (\text{C.4})$$

De acordo com Nakajima (1989), um OEE de 85% deve ser buscado como meta ideal para os equipamentos. Empresas que obtiveram OEE superior a 85% ganharam o prêmio TPM Award. Para se obter esse valor de OEE é necessário que seus índices sejam de: 90% para disponibilidade, 95% performance e 99% qualidade.

Hansen (2006) define os resultados da OEE da seguinte maneira:

- < 65%. Inaceitável. Dinheiro escondido e jogado fora. Peça ajuda agora;
- 65% – 75%. Aceitável somente se as tendências trimestrais estiverem melhorando;
- 75%~85%. Muito bom. No entanto, não fique parado. Valores de classe mundial: > 85% para processos em lotes; e > 90% para processos discretos e contínuos. Indústrias de fluxo continuo devem ter valores da OEE de 95% ou superior).

Contudo, a média mundial do OEE está em torno de 60% para as industrias em geral, ainda é um valor baixo, pois o aceitável é acima de 65%.

## APÊNDICE D – CÓDIGO PRINCIPAL UTILIZADO NO ARDUINO

A seguir é mostrado o código principal do programa utilizado no arduino para controlar o robô Zumo para seguir linhas pretas em um fundo claro.

```
/*
 * Demo line-following code for the Pololu Zumo Robot
 *
 * This code will follow a black line on a white background, using a
 * PID-based algorithm. It works decently on courses with smooth, 6"
 * radius curves and has been tested with Zumos using 30:1 HP and
 * 75:1 HP motors. Modifications might be required for it to work
 * well on different courses or with different motors.
 */

#include <QTRsensors.h>
#include <ZumoReflectanceSensorArray.h>
#include <ZumoMotors.h>
#include <ZumoBuzzer.h>
#include <Pushbutton.h>

ZumoBuzzer buzzer;
ZumoReflectanceSensorArray reflectanceSensors;
ZumoMotors motors;
Pushbutton button(ZUMO_BUTTON);
int lastError = 0;

// This is the maximum speed the motors will be allowed to turn.
// (400 lets the motors go at top speed; decrease to impose a speed limit)
const int MAX_SPEED = 150;
```

```
void setup()
{
    // Play a little welcome song
    buzzer.play(">g32>>c32");

    // Initialize the reflectance sensors module
    reflectanceSensors.init();

    // Wait for the user button to be pressed and released
    button.waitForButton();

    // Turn on LED to indicate we are in calibration mode
    pinMode(13, OUTPUT);
    digitalWrite(13, HIGH);

    // Wait 1 second and then begin automatic sensor calibration
    // by rotating in place to sweep the sensors over the line
    delay(1000);
    int i;
    for(i = 0; i < 80; i++)
    {
        if ((i > 10 && i <= 30) || (i > 50 && i <= 70))
            motors.setSpeeds(-150, 150);
        else
            motors.setSpeeds(150, -150);
        reflectanceSensors.calibrate();
    }
}
```

```
// Since our counter runs to 80, the total delay will be
// 80*20 = 1600 ms.
delay(20);
}

motors.setSpeeds(0,0);

// Turn off LED to indicate we are through with calibration
digitalWrite(13, LOW);
buzzer.play(">g32>>c32");

// Wait for the user button to be pressed and released
button.waitForButton();

// Play music and wait for it to finish before we start driving.
buzzer.play("L16 cdegreg4");
while(buzzer.isPlaying());
}

void loop()
{
    unsigned int sensors[6];

    // Get the position of the line. Note that we *must* provide the "sensors"
    // argument to readLine() here, even though we are not interested in the
    // individual sensor readings
    int position = reflectanceSensors.readLine(sensors);
```

```
int error = position - 2500;

int speedDifference = error / 4 + 6 * (error - lastError);

lastError = error;

// Get individual motor speeds.  The sign of speedDifference
// determines if the robot turns left or right.
int m1Speed = MAX_SPEED + speedDifference;
int m2Speed = MAX_SPEED - speedDifference;

// Here we constrain our motor speeds to be between 0 and MAX_SPEED.
// Generally speaking, one motor will always be turning at MAX_SPEED
// and the other will be at MAX_SPEED-|speedDifference| if that is positive,
// else it will be stationary.  For some applications, you might want to
// allow the motor speed to go negative so that it can spin in reverse.
if (m1Speed < 0)
    m1Speed = 0;
if (m2Speed < 0)
    m2Speed = 0;
if (m1Speed > MAX_SPEED)
    m1Speed = MAX_SPEED;
if (m2Speed > MAX_SPEED)
    m2Speed = MAX_SPEED;

motors.setSpeeds(m1Speed, m2Speed);
}
```