Leonardo Marquez Pedro

***Uma proposta de sistema robótico para manipulação
e interação física segura em ambientes não
estruturados***

Tese de Doutorado apresentada ao Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do Título de Doutor em Engenharia Mecânica

Orientador:

Prof. Dr. Glauco Augusto de Paula Caurin

SEM - DEPARTAMENTO DE ENGENHARIA MECÂNICA
EESC - ESCOLA DE ENGENHARIA DE SÃO CARLOS
USP - UNIVERSIDADE DE SÃO PAULO

São Carlos - SP

2013

ESTE EXEMPLAR TRATA-SE
DA VERSÃO CORRIGIDA.
A VERSÃO ORIGINAL
ENCONTRA-SE DISPONÍVEL
JUNTO AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA
EESC-USP

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

P372u Pedro, Leonardo Marquez
 Uma proposta de sistema robótico para manipulação e
 interação física segura em ambientes não estruturados /
 Leonardo Marquez Pedro; orientador Glauco Augusto de
 Paula Caurin. São Carlos, 2013.

 Tese (Doutorado) - Programa de Pós-Graduação em
 Engenharia Mecânica e Área de Concentração em Dinâmica
 de Máquinas e Sistemas -- Escola de Engenharia de São
 Carlos da Universidade de São Paulo, 2013.

 1. manipulação robótica. 2. planejador de
 manipulação. 3. planejador de trajetórias. 4. prevenção
 de colisão. 5. tratamento de colisão. 6. controle de
 impedância. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **LEONARDO MARQUEZ PEDRO**.

Título da tese: "Uma proposta de sistema robótico para manipulação e interação física segura em ambientes não estruturados".

Data da defesa: 28/06/2013

Comissão Julgadora:

Resultado:

Prof. Associado **Glauco Augusto de Paula Caurin (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

Aprovado

Prof. Associado **Marcelo Becker**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Associado **Evandro Luis Linhari Rodrigues**
(Escola de Engenharia de São Carlos/EESC)

APROVADO.

Profª. Associada **Roseli Aparecida Francelin Romero**
(Instituto de Ciências Matemáticas e de Computação/USP)

APROVADO

Prof. Associado **Arturo Forner Cordero**
(Escola Politécnica/USP)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Mecânica:
Prof. Associado **Marcelo Areias Trindade**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Denis Vinicius Coury**

*Dedico este trabalho às minhas irmãs
Fernanda e Manuela
pelo apoio incondicional.*

Agradecimentos

Primeiramente a Deus pela vida, iluminação, inspiração e saúde durante a realização deste trabalho.

Aos meu pais Manuel Augusto e Maria do Carmo, pelo amor, incentivo e apoio, e à toda a minha família por terem compreendido a minha ausência durante a dedicação aos trabalhos. Sem a ajuda e o apoio de vocês nada seria possível.

Ao Professor Glauco Caurin pela oportunidade de desenvolvimento do trabalho, paciência, ensinamentos e excelente orientação.

Aos colegas de trabalho e amigos Belini, Guilherme e Marius pelos trabalhos em parceria e pelas valiosas contribuições.

Aos muitos colegas, professores e amigos do Laboratório de Mecatrônica pelas idéias, sugestões e críticas, e principalmente pela amizade e ótimo convívio dentro e fora do ambiente de trabalho.

Ao CNPq pelo apoio financeiro com o oferecimento da bolsa de doutorado.

Por fim, gostaria de agradecer a todos os demais amigos e colegas, cujos nomes não caberiam nesta folha, que de alguma forma contribuíram para esta realização, e que ao lerem estas palavras saberão que foram lembrados. Muito obrigado!

"Um grama de ação vale mais que uma tonelada de teoria."
Friedrich Engels

Resumo

PEDRO, L. M. **Uma proposta de sistema robótico para manipulação e interação física segura em ambientes não estruturados**. Tese (Doutorado). Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.

Este trabalho propõe um sistema de manipulação robótica para interação física segura com objetos ou humanos em ambientes não estruturados. A proposta considera a execução de tarefas de manipulação e a prevenção e tratamento de colisões utilizando apenas uma lei de controle, o controle de impedância. A inovação científica consiste em um sistema multifuncional implementado com uma única lei de controle em contraste com os sistemas já existentes que utilizam chaveamento entre controladores para cada diferente funcionalidade do sistema, e que apresentam diversas desvantagens como instabilidade e oscilações, aumento da complexidade de programação, entre outras. Inicialmente é proposto um planejador de manipulação e *regrasping* baseado na combinação de trajetórias suaves e na adaptação dos parâmetros de um controle de impedância em tempo de execução. A mudança da impedância para cada etapa é obtida pela modificação dos parâmetros de inércia, rigidez e amortecimento do controlador. A estabilidade desta mudança dinâmica é possível pela utilização de trajetórias suaves obtidas com planejador *Squeezed Screw* modificado, cujas trajetórias geradas são livres de descontinuidades na posição e na velocidade. Adicionalmente, a prevenção de colisões é realizada com o auxílio de campos potenciais de forças de repulsão formados pela análise de dados de um sistema de visão também proposto. Estes mesmos dados são utilizados para a construção de um mapa de impedâncias ao redor do objeto cuja finalidade é suavizar efeitos de colisões indesejadas. Experimentos com um robô de arquitetura aberta e com um sistema de visão de baixo custo foram realizados na execução tarefa de manipulação de referência para se avaliar o desempenho da metodologia proposta em diferentes condições de operação encontradas em ambientes não estruturados, como por exemplo: erros de medida de posição, de calibração, ocorrência de colisões, etc. A tarefa de manipulação eleita foi a reorientação em 60° de um objeto circular no plano. Os resultados obtidos nos experimentos mostram a capacidade do controle de impedância associado a trajetórias suaves de realizar a tarefa eleita segundo avaliação utilizando como métricas de desempenho a porcentagem de reorientação, que apresentou uma média de 80% mesmo na presença de erros de medida do sensor de visão e erros de determinação da posição do objeto.

Palavras-chave: manipulação robótica; planejador de manipulação; planejador de trajetórias; prevenção de colisão; tratamento de colisão; controle de impedância.

Abstract

PEDRO, L. M. **A proposal of a robotic manipulation system for safe physical interaction in non-structured environments.** Thesis (PhD). Engineering School of São Carlos, University of São Paulo, São Carlos, 2013.

Recent applications in various robotics areas consider interaction between robots and objects or humans in non-structured environments. Under these conditions, in addition to the desire of robots to be able to perform their main tasks, handling, navigation, rehabilitation, etc, it is also desired to prevent and properly handle possible unwanted collisions, whether with objects, with other robots, animals or humans. There are several proposed methods for avoidance, handling and reaction for collisions, however, a widely used strategy is the controller switching between different robot states. There are several drawbacks within this strategy: instability and oscillation, increased programming complexity and consequent increased failure risk, need for different sensors and consequent increase in cost, among others. This work proposes a system applied to the robotic manipulation which is based on only one control law, the impedance control, whose expected capacity is, further performing manipulation tasks, avoidance and handling of potential undesired collisions. It is initially proposed a manipulation planner based the combination of smooth trajectories and the adjustment of parameters an impedance control at runtime. The change of impedance for each phase is achieved by modifying the parameters: mass, spring and damping controller. The stability of this dynamic change is possible by using smooth trajectories obtained with a modified Squeezed Screw trajectory planner, whose paths are discontinuities free in the position and speed. Additionally, collision avoidance is achieved through potential fields the repulsive forces of formed by analysis of data vision. The same data is used to construct an impedance map surrounding the object which objective is collision handling. Experiments with an open architecture robot and a low cost vision system are carried out in the execution of a benchmark manipulation task to evaluate the proposal performance under different operating conditions found in unstructured environments, for example, position measurement errors, calibration problems, occurrence of collisions, among others.

Keywords: robotic manipulation, collision avoidance, collision handling, impedance control

Sumário

Lista de Símbolos	xix
Lista de Abreviaturas e Siglas	xxiii
Lista de Figuras	xxv
Lista de Tabelas	xxxi
1 Introdução	1
1.1 Motivação e revisão bibliográfica	1
1.1.1 Desafios da manipulação em ambientes não estruturados	4
1.2 Objetivos do trabalho	8
1.3 Metodologia adotada	10
1.3.1 Definição da tarefa teste e métrica de desempenho	11
1.3.2 Ambiente experimental	11
1.4 Organização do texto	16
2 Sistema de visão	17
2.1 Introdução	17
2.2 Calibração Kinect e robô SCARA	20
2.2.1 Experimento de qualidade de calibração entre SCARA e Kinect	22
2.3 Detecção de objetos móveis	24
2.3.1 Métodos de subtração de fundo de imagens	25
2.3.2 O método de subtração de fundo <i>Temporal median filter</i>	26

2.3.3	Filtros morfológicos	27
2.4	Experimentos e resultados	28
2.4.1	Detecção de objetos móveis	28
2.4.2	Segmentação dos pontos e identificação do robô	29
2.5	Avaliação do sistema de visão	32
3	Planejador de trajetórias	33
3.1	Planejador de caminhos <i>Squeezed Screw</i>	33
3.1.1	Parâmetros de definição	34
3.2	Perfil de velocidades para o <i>Squeezed Screw</i>	35
3.3	Comparação <i>Squeezed Screw</i> com o RRT	37
3.3.1	Comparação quantitativa	37
3.4	Exemplo de geração de trajetória	41
4	Planejador de manipulação	45
4.1	Introdução	45
4.2	Controle de Impedância	47
4.2.1	Contextualização	47
4.2.2	Equacionamento	48
4.3	Arquitetura do planejador de manipulação	50
4.3.1	Planejador de trajetórias suaves para a manipulação	52
4.3.2	Desvio e tratamento da colisão	54
5	Experimentos de manipulação	61
5.1	Introdução	61
5.2	Experimento de estabilidade de contato	61
5.3	Experimento de manipulação: estabilidade e desempenho em diferentes fases	70
5.4	Experimento de integração de manipulação e visão	80

6 Conclusões, discussões e considerações finais	95
6.1 Proposta para trabalho futuro	100
Referências Bibliográficas	103
Apêndice A – Avaliação do sensor Kinect	109
A.1 Avaliação da repetibilidade	109
A.1.1 Experimento piloto de repetibilidade	110
A.1.2 Experimento de repetibilidade	111
A.2 Experimento de precisão de posicionamento e de registro	116
A.2.1 Detecção de pontos de identificação	116
A.2.2 Descrição do experimento	118
A.3 Avaliação do Kinect para aplicações em robótica	119
Apêndice B – Reconhecimento de objetos com redes neurais	121
B.1 Introdução	121
B.2 Normalização	122
B.3 Testes das RNAs e resultados	123
B.3.1 Rede <i>Perceptron</i>	124
B.3.2 Rede Adaline	126
B.3.3 Redes <i>Perceptron</i> Multicamadas (PMC)	127
B.3.4 Redes RBF	127
B.3.5 Redes LVQ	128
B.3.6 Redes Kohonen	128
B.4 Avaliação da classificação de objetos com RNAs	128
Apêndice C – Principais publicações no período de doutorado	131
C.1 Periódicos	131

C.2	Capítulos de livro	131
C.3	Congressos	131

Lista de Símbolos

α	constante de adaptação da média μ
β	constante de adaptação da média σ
act	subíndice para atuadores
dc	referente a parâmetros corrigidos
d	superíndice referente a desejado
est	subíndice para valor estimado
ext	subíndice para externo
i	subíndice de contagem
$manip$	subíndice referente à manipulação
m	superíndice referente ao mapa de impedância
obj	subíndice referente ao objeto
$real$	subíndice para valor real
rep	subíndice referente à repulsão
r	superíndice referente ao robô
t	subíndice e índice referentes a um instante de tempo
μ	média
∂	diferencial infinitesimal
σ	desvio padrão
τ	torque nas juntas
orb	comprimento da elipse de construção do <i>Squeezed Screw</i>
B	amortecimento da impedância (matricial ou escalar)
b	semi-eixo menor da elipse de construção do <i>Squeezed Screw</i>
C	matriz de termos de Coriolis e de forças centrípetas do manipulador
$L(\theta)$	cinemática direta generalizada do manipulador
x, y, z	coordenadas espaciais de um ponto ou vetor
d	distância
$\ddot{x}$	segunda derivada da posição no tempo

$\ddot{\theta}$	aceleração angular de junta
$\dot{x}$	primeira derivada da posição no tempo
$\dot{x}_0$	primeira derivada no tempo da posição de referência
$\dot{\theta}$	velocidade angular de junta
P^V	ponto espacial representado no sistema de referência V
F	força
I	matriz de inércia do manipulador
J	matriz jacobiana do manipulador
k	constante de ajuste para subtração de fundo
K	rigidez da impedância (matricial ou escalar)
M	inércia da impedância (matricial ou escalar)
m	massa
M	matriz binária para adaptação da imagem de referência para a subtração de fundo
M	tensor de mobilidade do manipulador
H_V^S	matriz de transformação dos sistema V para o sistema S
ϕ'	orientação relativa da elipse de construção do <i>Squeezed Screw</i>
s	deslocamento do <i>Squeezed Screw</i>
S	sistema de referência de base do robô SCARA
T	matriz de intensidade de <i>pixels</i>
T	período
T_c	tempo em velocidade de cruzeiro
T_f	tempo de aceleração final
θ	posição angular de junta
T_i	tempo de aceleração inicial
V	sistema de referência de base do Kinect
v	velocidade escalar
$v_a(t)$	perfil de velocidade de aceleração
v_c	velocidade de cruzeiro
$v_d(t)$	perfil de velocidade de desaceleração
v_f	velocidade final
v_i	velocidade inicial

W	trabalho mecânico
x	posição
x_0	posição de referência

Lista de Abreviaturas e Siglas

2D	Bidimensional
3D	Tridimensional
ICP	<i>Iterative Closest Point</i>
LED	<i>Light Emitting Diode</i>
LVQ	<i>Learning Vector Quantization</i>
PMC	<i>Perceptron Multicamadas</i>
RBF	<i>Radial Basis Function</i>
RGB	<i>Red, Green and Blue</i>
RNA	Rede Neural Artificial
RRT	<i>Rapidly-Exploring Random Trees</i>
SDK	<i>Software Development Kit</i>
SNC	Sistema Nervoso Central
TCP	<i>Tool Center Point</i>

Lista de Figuras

1.1	Exemplo da tarefa de manipulação eleita para os experimentos. Figura modificada de Xu et al. (2007).	11
1.2	Robô SCARA retrofitado e sua representação cinemática.	12
1.3	Sensor de visão: Kinect.	13
1.4	Ambiente experimental.	14
2.1	Sistemas de referência do sensor Kinect e do robô SCARA.	20
2.2	Resultado da calibração. Círculos representam a posição do LED e os pontos representam a posição medida com Kinect após transformação com a matriz H_K^S obtida.	22
2.3	Erro em x obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.	23
2.4	Erro em y obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.	23
2.5	Erro em z obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.	23
2.6	Exemplo de registro entre imagem RGB e <i>depth image</i> do Kinect.	24
2.7	Exemplo de identificação de objetos móveis na cena do robô utilizando o método <i>Running Gaussian Average</i> para subtração de fundo. A figura 2.7(a) apresenta a imagem de fundo, a figura 2.7(b) a imagem atual da cena e a figura 2.7(c) a imagem resultante da subtração de fundo.	28
2.8	Nuvem de pontos de objetos móveis.	29
2.9	Modelo poligonal simplificado do robô SCARA.	30
2.10	Exemplo de subtração de fundo. À esquerda a imagem original e à direita o resultado da subtração de fundo.	30

2.11	Nuvem de pontos da cena da figura 2.10(a).	31
2.12	Exemplo de segmentação 3D. Em verde os pontos pertencentes à mesa, em azul os pontos pertencentes ao objeto, e em vermelho os pontos pertencentes ao robô.	31
3.1	Exemplo de uma composição de rota com <i>Squeezed Screw</i> e sua elipse base (linhas contínuas) e sua translação (linhas tracejada).	33
3.2	Ilustração das opções para a variável <i>orb</i> . (a) ' <i>shrt</i> ', (b) ' <i>long</i> ' e (c) ' <i>xlng</i> '. Figuras de Stucheli et al. (2013).	34
3.3	Caminhos das trajetórias comparadas. Em linha tracejada a rota do RRT (XU et al., 2007) e em linha contínua a do <i>Squeezed Screw</i>	38
3.4	Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo <i>Squeezed Screw</i> e pelo RRT (XU et al., 2007). Velocidade angular.	39
3.5	Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo <i>Squeezed Screw</i> e pelo RRT (XU et al., 2007). Aceleração angular.	40
3.6	Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo <i>Squeezed Screw</i> e pelo RRT (XU et al., 2007). <i>Jerk</i> angular.	40
3.7	Trajetória de manipulação usando os dados apresentados na tab. 3.1.	42
3.8	Perfil de velocidade tangencial na trajetória.	43
4.1	Diagrama em blocos do sistema de manipulação proposto.	51
4.2	Trajetória de manipulação com deslocamento da trajetória inicial que tangencia o círculo que representa a tampa do pote.	53
4.3	Exemplo de discretização de um espaço em elementos volumétricos discretos, os <i>voxels</i>	57
4.4	Exemplo de elemento estruturante 3D.	58
4.5	Nuvem de pontos de um objeto.	58
4.6	Vista superior da nuvem de pontos da figura 4.5.	59
4.7	Projeção 2D do mapa de impedância.	59
5.1	Resultado do experimento 1 com $M = 5kgm^2$, $K = 6N/m$ e $B = 100Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.	63

- 5.2 Resultado do experimento 2 com $M = 5kgm^2$, $K = 6N/m$ e $B = 500Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 64
- 5.3 Resultado do experimento 3 com $M = 5kgm^2$, $K = 6N/m$ e $B = 750Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 64
- 5.4 Resultado do experimento 4 com $M = 5kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 65
- 5.5 Resultado do experimento 5 com $M = 5kgm^2$, $K = 6N/m$ e $B = 2000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 65
- 5.6 Resultado do experimento 6 com $M = 5kgm^2$, $K = 6N/m$ e $B = 3000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 65
- 5.7 Resultado do experimento 7 com $M = 5kgm^2$, $K = 100N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 66
- 5.8 Resultado do experimento 8 com $M = 5kgm^2$, $K = 200N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 66
- 5.9 Resultado do experimento 9 com $M = 5kgm^2$, $K = 400N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 67
- 5.10 Resultado do experimento 10 com $M = 5kgm^2$, $K = 1000N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 67
- 5.11 Resultado do experimento 11 com $M = 50kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 68
- 5.12 Resultado do experimento 12 com $M = 100kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 68
- 5.13 Resultado do experimento 13 com $M = 200kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto. 69

5.14	Resultado do experimento 14 com $M = 400kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.	69
5.15	Experimento de manipulação com mudança repentina de impedância entre as fases.	72
5.16	Experimento de manipulação com mudança suave de impedância entre as fases.	74
5.17	Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x	76
5.18	Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e na direção y	77
5.19	Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e de $+10mm$ na direção y	78
5.20	Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e de $+10mm$ na direção y	79
5.21	Resultados do processamento de imagens. Primeira repetição do experimento de integração de manipulação com sistema de visão.	82
5.22	Resultados da manipulação. Primeira repetição do experimento de integração de manipulação com sistema de visão.	83
5.23	Resultados do processamento de imagens. Segunda repetição do experimento de integração de manipulação com sistema de visão.	84
5.24	Resultados da manipulação. Segunda repetição do experimento de integração de manipulação com sistema de visão.	85
5.25	Resultados do processamento de imagens. Terceira repetição do experimento de integração de manipulação com sistema de visão.	86
5.26	Resultados da manipulação. Terceira repetição do experimento de integração de manipulação com sistema de visão.	87
5.27	Resultados do processamento de imagens. Quarta repetição do experimento de integração de manipulação com sistema de visão.	88
5.28	Resultados da manipulação. Quarta repetição do experimento de integração de manipulação com sistema de visão.	89

5.29	Resultados do processamento de imagens. Quinta repetição do experimento de integração de manipulação com sistema de visão.	90
5.30	Resultados da manipulação. Quinta repetição do experimento de integração de manipulação com sistema de visão.	91
5.31	Região do espaço de trabalho do robô preferencial para a manipulação.	93
A.1	Cena experimento piloto para a repetibilidade.	110
A.2	a) Imagem de representação do desvio padrão σ_x , b) Imagem de representação do desvio padrão σ_y , c) Imagem de representação do desvio padrão σ_z . <i>Pixels</i> com escala de cinza máximo (<i>pixels</i> pretos) são aqueles com o menor desvio padrão, <i>pixels</i> com escala de cinza mínimo (<i>pixels</i> branco) são aqueles com desvio padrão igual ao limiar empírico. Para σ_x o limiar foi $4mm$, para σ_y foi $3mm$ e $7mm$ para σ_z	111
A.3	Experimento de acuidade do Kinect.	112
A.4	Desvio padrão σ_x em função de $\bar{x}$ para $d = 1,9m$	112
A.5	Média σ_x para intervalos de $\bar{x}$. Dados foram divididos em 1000 intervalos ($d = 1,9m$).	113
A.6	Média σ_x em função de $\bar{x}$ e d . A distância d foi variada de $0,6m$ a $1,9m$ em passos de $0,1m$	113
A.7	Variação de σ_y médio em função de $\bar{y}$ e d	114
A.8	Variação de σ_z médio em função de $\bar{z}$	114
A.9	Variação de σ_y médio em função de $\bar{x}$ e d	115
A.10	Variação de σ_z médio em função de $\bar{y}$ e d	115
A.11	Exemplo de registro entre imagen RGB e <i>depth image</i> do Kinect.	117
A.12	Imagem RGB de um LED montado no robô SCARA. À esquerda a imagem sem o filtro de luz, e à direita a imagem com o filtro de luz.	117
A.13	Exemplo de binarização por Otsu. Limiar de binarização de 0,1.	118
A.14	Experimento de erro de posicionamento. À direita a escala de cores correspondente aos erros de posicionamento em mm	119
B.1	Exemplo de nuvem de pontos de um cilindro simplificada para 100 pontos. . .	123

Lista de Tabelas

3.1	Parâmetros cinemáticos para as subtrajetórias.	42
3.2	Parâmetros geométricos do <i>Squeezed Screw</i> para as subtrajetórias.	43
B.1	Saídas desejadas para cada uma das classes.	124
B.2	Tabela de grupos de treinamento para os testes. Número de dados de treinamento e de testes para cada um das classes.	124
B.3	Resultados de classificação de objetos com diferentes redes <i>perceptrons</i> (classificação geral).	125
B.4	Resultados de classificação de objetos com diferentes redes <i>perceptrons</i> (classificação por pares).	125
B.5	Resultados de classificação de objetos com diferentes redes <i>Adaline</i> (classificação geral).	126
B.6	Resultados de classificação de objetos com diferentes redes <i>perceptrons</i> (classificação por pares).	127
B.7	Resultados de classificação de objetos com diferentes redes RBF.	127
B.8	Resultados de classificação de objetos com diferentes redes LVQ.	128
B.9	Resultados de classificação de objetos com diferentes redes LVQ (sem a classe de esferas).	128

1 Introdução

1.1 Motivação e revisão bibliográfica

Os seres humanos utilizam suas habilidades de manipulação de objetos na maioria das atividades cotidianas. Organizar uma mesa de trabalho, por exemplo, envolve tarefas que requerem estas habilidades: mudar um livro de lugar, alinhar e organizar folhas, tampar uma caneta, etc. Na manipulação os seres humanos interagem com o ambiente modificando-o. No exemplo de organizar a mesa, a interação ocorre com objetos, porém, a manipulação também pode envolver interação com seres humanos, como por exemplo os terapeutas trabalhando exercícios em um paciente.

Mas o que é a manipulação robótica? A maioria dos autores da área fazem a sua própria definição e não há uma capaz de contemplar todas as aplicações e possibilidades. Apesar de não haver uma definição mais abrangente, o ponto em comum das várias existentes é a execução de tarefas específicas pela interação entre sistemas robóticos e o ambiente por meio do contato, fenômeno que diferencia a área das demais da robótica (BICCHI, 1999). O contato é o fenômeno que permite a interação do robô com seu ambiente aplicando forças e movimentos com a finalidade de executar tarefas.

Muitos autores consideram em suas definições somente a interação entre robôs e objetos inanimados. Contudo, assim como para a manipulação humana, a manipulação robótica também pode envolver interação com seres humanos (mesmo que não intencionalmente ou acidentalmente). Por exemplo, o contato estabelecido entre paciente e um sistema robótico de reabilitação, como por exemplo o sistema apresentado por Caurin et al. (2011), também pode ser considerado um caso de manipulação robótica.

Assim, considera-se neste trabalho uma definição mais abrangente para a manipulação robótica. O objetivo principal da manipulação robótica é o conhecimento e o controle do fenômeno do contato para que o robô seja capaz de aplicar conjugados de forças e movimentos sobre elementos presentes no seu ambiente, sejam estes objetos ou seres humanos, e então realizar tarefas específicas com eficiência e segurança.

A ideia de um sistema autônomo de manipulação, independente de sua definição, traz a expectativa de substituir o trabalho humano em atividades perigosas, insalubres e em lugares de difícil acesso como profundezas oceânicas e exploração espacial. Essa substituição também é possível em casos de escassez de mão de obra ou em casos em que se deseja aumentar a qualidade e a produtividade industrial.

Os primeiros casos de aplicações de sistemas de manipulação ocorreram em 1951 justamente em aplicações industriais, como por exemplo a instalação de robôs Unimate, da empresa Unimation, em aplicações de movimentação de blocos de motores pesados e em elevadas temperaturas (MURRAY et al., 1994). Nestas primeiras aplicações os manipuladores robóticos utilizados eram operados por controle de posição e com movimentação segundo trajetórias lineares e circulares. Normalmente a ferramenta utilizada para pegar peças era uma garra industrial de dois ou três dedos com controle de abertura ou fechamento. Nestes sistemas as peças eram posicionadas de forma precisa, e caso não fossem, a tarefa não era executada pois o sistema não era capaz de realizar correções e ajustes automáticos.

O advento de novas tecnologias e de novas técnicas de controle e planejamento de trajetórias ampliou a capacidade dos robôs industriais em aplicações de manipulação e consequentemente novas aplicações foram conquistadas¹. Porém, As características de operação e funcionalidade dos primeiros robôs industriais atualmente são encontrados como padrão na indústria, salvo exceções como o mais recente controle híbrido de força e posição.

Além das conquistas em aplicações industriais, avanços foram obtidos na ciência, como por exemplo o desenvolvimento de garras mais avançadas, tendo como precursor aquela projetada por Salisbury (SALISBURY, 1982a). Além do desenvolvimento das garras robóticas, outras tecnologias também foram propostas para aumentar a capacidade de manipulação, tal como sistemas de visão, como, por exemplo, aqueles apresentados por Boissonnat (1982), Xu et al. (1990), Kragic et al. (2002) e Saxena et al. (2006), que podem ser bidimensionais (2D) e tridimensionais (3D), e que informam ao robô a posição e a geometria do objeto a ser manipulado, assim, tarefas de manipulação robótica podem ser executadas sem a exigência de estruturação do ambiente.²

É importante ressaltar que a robótica industrial e a ciência robótica não evoluíram de forma sincronizada. Alguns avanços da ciência foram transferidos para a indústria, tal como atuadores, sensores, mas muitos não foram, como exemplo as garras mais sofisticadas. Mesmo

¹Não é objetivo desta introdução trazer um histórico da evolução da manipulação no contexto industrial, desta forma, recomenda-se a leitura de Pedro (2008) trabalho no qual pode-se encontrar referências mais detalhadas.

²Recomenda-se a leitura de Pedro (2008) que apresenta uma revisão de sistemas de manipulação baseados em visão.

com o avanço de tecnologias de manipulação, nas aplicações industriais o ambiente continua estático ou pouco dinâmico pois o mesmo já precisava ser organizado e estruturado para favorecer à máxima produção. Normalmente o mapa do espaço de trabalho do robô e a geometria dos objetos são bem conhecidas. As aplicações industriais também são caracterizadas pela minimização de problemas de incertezas de posicionamento e de erros de medidas pelo uso de sensores, incluindo os de visão, de elevada precisão e repetibilidade. Atualmente a estruturação do ambiente e o uso destes sensores são mais viáveis que as garras robóticas mais desenvolvidas, e conseqüentemente abre-se mão do seu uso. a principal razão é o fato de que as garras ainda são de alto custo e estão voltadas para outras aplicações.

Uma outra característica marcante das aplicações industriais é a velocidade de movimentação. Um sistema de sensoriamento de precisão juntamente com o isolamento do espaço de trabalho do robô permite que este se movimente em velocidades elevadas. Estudos recentes, como o realizado por Haddadin et al. (2008), avaliam o efeito das velocidades de movimentação e a massa do robô em impactos com seres humanos. Resultados mostram que a velocidade tem um papel mais significativo que a massa nos casos em que se pode colocar em risco a vida de seres humanos.

Porém, ressalta-se que além das aplicações industriais, o auxílio ou substituição do trabalho humano pela manipulação robótica pode ocorrer em diversas outras aplicações, todas elas envolvendo ambientes não estruturados. Cita-se como exemplos:

- serviços em geral (domésticos, escritórios, hospitais, condomínios, construção civil);
- exploração de regiões inóspitas ou de difícil acesso;
- reabilitação;
- auxílio a idosos e deficientes físicos.

Diferente das aplicações industriais, nestas aplicações, além destas características do ambiente (pouco estruturados, altamente dinâmicos e não controlados), os objetos têm formas e propriedades (rigidez, densidade e coeficiente de atrito) que podem variar em um mesmo objeto ou de objeto para objeto. Assim, se faz necessário o uso das garras robóticas mais avançadas, normalmente antropomórficas e equipadas com diversos sensores de posição e de força, além de diversas outras funcionalidades como desvio e tratamento de colisões e sistemas de inteligência que dependem do da autonomia desejada.

1.1.1 Desafios da manipulação em ambientes não estruturados

A complexidade do sistema de manipulação necessária para realizar tarefas em ambientes não estruturados envolve vários desafios da área, tal que a grande maioria dos sistemas apresentados por pesquisadores funcionam em condições específicas e laboratoriais, afirmam, entre outros, Kemp et al. (2007), Okamura et al. (2000) e Bicchi (1999). Normalmente os sistemas requerem calibrações e ajustes antes de experimentos e testes. Tais procedimentos e ajustes não são viáveis para as aplicações citadas anteriormente, pois um engenheiro ou desenvolvedor pode seguir tais procedimentos antes de testes e experimentos, mas um usuário comum antes de uma utilização ocasional não.

Pode-se dizer que os principais desafios da manipulação para tornar estes sistemas complexos mais funcionais em ambientes não estruturados são:

- modelagem e simulação do contato;
- tratamento de colisões (tanto as desejadas quanto as imprevistas);
- desenvolvimento de sensores de força;
- desenvolvimento de sensores de visão;
- desenvolvimento de controladores e planejadores.

Alguns destes desafios serão discutidos a seguir com base na divisão proposta por Okamura et al. (2000), que sugere que os planejadores da manipulação são divididos em três níveis: primário, secundário e terciário.

No nível primário de planejamento estão as operações de controle, ou seja, controle das trajetórias (perfil de forças e velocidades no tempo) durante as diferentes fases da manipulação: movimentação livre e interação com ambiente. O nível secundário de planejamento é responsável por detectar ou prever as mudanças das fases da manipulação: o estabelecimento e perda de contato; rolagem; e escorregamento. Este nível também é responsável por gerenciar a mudança das leis de controle para cada uma das diferentes fases. O nível terciário de planejamento inclui, por exemplo, a escolha de uma postura de mão cuja configuração permita a execução de uma manipulação planejada, e sincronismo entre os dedos (*finger gaiting*). Ainda segundo Okamura et al. (2000), os principais desafios da manipulação atualmente estão concentrados principalmente nos níveis primário e secundário, fato que dificulta o desenvolvimento de planejadores do nível terciário.

Partindo do nível primário, o desafio inicial a ser discutido é o modelo de contato, e para tanto faz-se aqui uma divisão da manipulação em: 1) *grasping* ; e 2) manipulação destra. A tarefa de *grasping*, a primeira a ser desenvolvida na área, consiste em pegar um objeto utilizando uma garra e mantê-lo em equilíbrio estático em relação à mesma. Seu desenvolvimento tornou-se possível após a proposta dos primeiros modelos de contato: contato pontual sem atrito; contato pontual com atrito; e contato macio³ (NGUYEN, 1988), e das teorias: "*form-closure*"(LAKSHMINARAYANA, 1978); "*force-closure*"(NGUYEN, 1988) e qualidade do *grasping* Q_{∞} (FERRARI; CANNY, 1992)⁴.

Os primeiros modelos e suas consequentes teorias deram suporte ao desenvolvimento de diversos sistemas de *grasping*. Segundo Okamura et al. (2000), até meados de 2000 a tarefa de *grasping* tinha sido o tema de maior desenvolvimento na área. A maioria dos sistemas implementados, mesmo que muitos deles apenas em ambiente laboratorial, apresentam capacidade de *grasping*, porém suas aplicações são restritas somente a este caso.

Para uma manipulação destra, ou seja, para movimentar um objeto através da aplicação de forças e movimentos do robô, existem dificuldades de modelagem e controle pois aqueles modelos de contato simplificados não são adequados. Tomando como exemplo a manipulação humana, o contato envolve escorregamento, rolagem e deformações, fenômenos que não são considerados naqueles modelos. Assim, muitos autores afirmam que para a manipulação destra é necessário um modelo de contato mais adequado que permita a elaboração de novas teorias e sistemas.

Ao longo do desenvolvimento da manipulação destra, autores como Hyde et al. (1997), Bicchi (1999), Okamura et al. (2000), Tegin e Wikander (2005) e Kemp et al. (2007) têm destacado duas dificuldades na modelagem do contato: a carência de ferramentas matemáticas e a carência de sensores de força, escorregamento e deformação. Assim, mesmo os modelos de contato mais sofisticados (MURRAY et al., 1994; YIN et al., 2003) são eficientes somente em simulações quando não há ocorrência dos fenômenos ainda não modelados adequadamente, ou em casos experimentais simplificados, como em Bicchi (1999) no qual o escorregamento, difícil de ser medido e controlado, é evitado.

O nível secundário é responsável por detectar mudanças de fases da manipulação e a partir delas realizar mudanças ou ajustes no controle do primário. Um dos maiores desafios no desenvolvimento de planejadores para deste nível é sincronizar o ajuste da lei de controle com a mudança do estado do robô, como estabelecimento e perda de contato, início de escorregamento

³Tradução livre adotada para o termo *soft contact*.

⁴Recomenda-se a leitura de Bicchi (1999), Bicchi e Kumar (2002), Okamura et al. (2000) e Pedro e Caurin (2009) para um aprofundamento nas tarefas de *grasping*.

e rolagem, e detecção e tratamento de colisões indesejadas ou não.

Estabilidade e bom desempenho no estabelecimento e a perda de contato são os maiores desafios para o nível secundário. Diferentes estratégias para solucionar este desafio foram propostas. A mais simplificada é a utilização de materiais flexíveis recobrimo a garra, porém esta estratégia dificulta o controle da trajetória enquanto a garra está em contato com o objeto. Outras estratégias mais elaboradas utilizam troca da lei de controle em função da fase (HYDE et al., 1997; OKAMURA et al., 2000), nas quais por exemplo, utiliza-se controle de posição na movimentação livre e controle de força durante a manipulação.

Esta troca de controladores deve considerar algumas características dinâmicas. Antes do contato, o robô movimenta-se livre, sem restrições. No instante do contato há uma colisão com consequente transferência de energia entre robô e objeto juntamente com a imposição de novas restrições cinemáticas e dinâmicas. Após o contato, a dinâmica do sistema é diferente daquela anterior, várias em função do objeto, e ainda pode ser variável para um objeto heterogêneo, como objeto com densidade variável ou flexível. Assim, a troca de controle deve ser sincronizada com o estabelecimento e perda do contato, que pode ser realizada pela detecção da colisão utilizando sensores de força, ou pela sua estimativa utilizando sistemas de visão.

Alguns autores, como Hyde et al. (1997) e Okamura et al. (2000), justificam a troca de controladores a partir da detecção da colisão com sensores de força por indícios de que a manipulação humana é dirigida por eventos. Contudo, os autores não apresentam indícios de que há troca da lei de controle ou de quais são os tipos utilizados em cada fase. Além disso, em contraste com a justificativa, estudos indicam que a latência de reflexos do ser humano a um estímulo tátil pode variar de 20 até 150ms (HOGAN, 1985)(TEGIN; WIKANDER, 2005).

Independente de justificativas, há várias desvantagens para a troca de controladores baseada na detecção da colisão. Hyde et al. (1997) destaca e avalia a instabilidade e as oscilações na troca entre diferentes tipos de controladores. Uma das possíveis explicações é a latência entre ocorrência e detecção da colisão, pois para a detecção é necessário que a colisão ocorra. Neste pequeno intervalo de tempo energia é transferida do robô para o ambiente, e o novo controle deverá estabilizar a resposta desta transferência. Também existem problemas de ruídos que podem ser detectados erroneamente como colisões (CUTKOSKY; HYDE, 1993).

Uma alternativa amplamente investigada é a previsão da colisão a partir da análise de dados de sensores de visão (KEMP et al., 2007). Utilizando visão é possível prever o estabelecimento de um contato e realizar a troca de controladores instantes antes da colisão, e assim, as instabilidades durante e após a colisão podem ser melhor controladas. Porém, também há desvantagens, sendo as principais: oclusão; custo computacional; precisão de medidas; ruídos; e

repetibilidade.

Existem propostas de solução para a oclusão utilizando modelos virtuais 3D, porém, para tanto é necessário conhecer previamente os objetos. Esta estratégia demanda ainda mais processamento computacional que já é elevado devido à grande quantidade de dados obtidos por um sistemas de visão. Fato que dificulta que a previsão seja realizada em tempo de execução. Quando o sistema de visão é baseado em câmeras também ocorrem problemas de processamento de imagens como a falta de iluminação e a ocorrência de sombras. Problemas de precisão de medida também estão presentes, sendo caros os sistemas mais precisos.

Além das colisões desejadas, sendo que muitas delas podem favorecer a manipulação, como por exemplo utilizar o contato com uma mesa para pegar um lápis, na manipulação também ocorrem colisões indesejadas, que podem ocorrer entre os próprios dedos da garra, entre os dedos e a sua palma, entre a garra e o objeto, e mais, quando se considera que algumas aplicações de manipulação envolvem ambientes não estruturados, colisões com humanos, animais e outros objetos, incluindo outros robôs no caso de cooperação robótica, também podem ocorrer. Em ambientes não estruturados torna-se inviável monitorar e prever toda a movimentação no ambiente, principalmente aquelas de atividades não relacionadas com as tarefas do robô.

Para o caso de evitar colisões, na manipulação normalmente são utilizados algoritmos de *collision avoidance* como o algoritmo *Rapidly-Exploring Random Trees* (RRT) (JR.; LAVALLE, 2000) entre outros (FOX et al., 1997; KHATIB, 1986), que são mais adequados para aplicações em que as colisões devem sempre ser evitadas e não procedimentos para tratamento de colisões indesejadas ou não esperadas. Em sistemas de manipulação utiliza-se com frequência controladores com realimentação de força, e problemas de instabilidade destes controladores são conhecidos principalmente em colisões com objetos de elevada rigidez (AN; HOLLERBACH, 1987)(EPPINGER; SEERING, 1987)(LAWRENCE, 1988) e (EPPINGER; SEERING, 1992).

Para que um robô de manipulação realize tarefas em ambientes dinâmicos é preciso considerar desde a estabilidade intrínseca do controlador utilizado quando em contato com objetos de elevada rigidez, e ainda na presença de humanos é preciso um sistema de tratamento mais inteligente de colisões como os propostos por Haddadin et al. (2008) e Haddadin et al. (2010).

Para o nível terciário existe muito desenvolvimento para *grasping* e muitas soluções propostas, porém, estes casos simplificam os desafios dos níveis primário e secundário. Neles o modelo de contato simplificado é adequado e a troca de controle é suave, pois estes sistemas apresentam baixas velocidade de aproximação com o objeto. Apesar dos bons resultados para *grasping*, Cutkosky e Hyde (1993) comentam que o controle do estabelecimento do contato

pode levar uma tarefa de *grasping* a falha mesmo com uma condição *force-closure* otimizada.

Para o caso da manipulação destra ainda existem desafios no nível terciário, alguns deles estão relacionados ao próprio planejador e outros estão relacionados a tentativas de contornar neste os desafios dos dois níveis abaixo. Em casos de simplificação do contato, tal como os que normalmente são encontrados nas simulações, apresentam falhas em experimentos, ou funcionam apenas em condições controladas (BICCHI, 1999), (OKAMURA et al., 2000), (KEMP et al., 2007).

Duas estratégias que trazem expectativas para o desenvolvimento da manipulação destra são a *Learning-by-Demonstration* e a teleoperação. Nestes dois casos, o planejamento é realizado pelo ser humano, e neste caso, a sua capacidade de manipulação é transferida para a manipulação robótica e alguns dos problemas dos níveis abaixo são resolvidos ou controlados de forma intrínseca. Estes sistemas têm apresentado desempenho satisfatório em aplicações específicas (BICCHI, 1999), (OKAMURA et al., 2000), (KEMP et al., 2007).

Para uma teleoperação mais eficiente, o operador deve interagir com dispositivos capazes de retransmitir, mesmo que de forma proporcional ou com redução de dados, as forças e os movimentos experimentados pelo robô (HOGAN, 1989). Este sistema de interface com o operador também é um sistema robótico e, segundo a definição de manipulação deste trabalho, também pode ser considerado como um sistema de manipulação. Neste caso de manipulação, a estabilidade e o tratamento de colisões é de extrema importância para a segurança do ser humano. Uma situação semelhante é observada em robôs de reabilitação (CAURIN et al., 2011). Assim, nestes dois casos, o tratamento de colisões torna-se mais importante.

1.2 Objetivos do trabalho

Os desafios levantados na seção anterior não são os únicos enfrentados pelos pesquisadores da área de manipulação. As discussões apresentadas dão suporte a duas questões que motivaram a proposta e desenvolvimento deste trabalho:

É necessário modelar e controlar precisamente o contato para realizar tarefas de manipulação?

É possível desenvolver um sistema simplificado capaz de realizar tarefas de manipulação?

A intenção deste trabalho é, em primeiro lugar, tentar responder a estas duas questões, sendo sua proposta o estudo e desenvolvimento de um sistema de manipulação para trabalhos em ambientes não estruturados. Assim, pretende-se estudar e propor um sistema de manipulação

com as seguintes características e funcionalidades:

1. Capacidade de realizar tarefas de manipulação sem a necessidade de modelagem e controle do contato;
2. Capacidade de estabelecer e perder contato de forma estável;
3. Capacidade de prevenir, e no caso de ocorrência, tratar colisões indesejadas de forma adequada;
4. Robustez a erros de medida e ruídos dos sensores de visão.

Conforme descrito anteriormente, normalmente utilizam-se: modelos de contato complexos e de difícil simulação e controle; sensores intrínsecos e extrínsecos não comerciais, de difícil calibração e baixa confiabilidade, pois são adaptados para cada caso; sistemas complexos de detecção de contato baseados em fusão de sensores de visão e de força, sistemas complexos para planejamento de trajetórias livres de colisão indesejadas; e estratégias de troca de controladores.

A proposta é um sistema simplificado para manipulação no qual algumas das principais funcionalidades da manipulação são solucionadas utilizando uma única lei de controle do nível primário de planejamento: o controle de impedância (HOGAN, 1985). Todas as funcionalidades desejadas serão implementadas no nível secundário e repassadas ao anterior na forma de ajuste dos parâmetros de impedância. O controle de impedância tem sido destacado como solução para os efeitos da colisão e para o problema da troca de controlador no estabelecimento do contato (HYDE et al., 1997). A primeira vantagem é a eliminação da necessidade de troca de controladores, pois com o mesma lei de controle é possível tanto realizar movimentos livres no espaço, quanto em contato com objetos controlando a relação dinâmica entre força e velocidade. Assim, não há problemas de instabilidade pela troca de lei de controle, pois a mudança para características adequadas para cada fase da manipulação pode ser conseguida pela mudança dos parâmetros de impedância.

Neste trabalho também se avalia a tolerância e robustez da proposta a problemas de medidas de posição, tais como ruído, precisão e repetibilidade. Estes erros são característicos de ambientes não estruturados, como exploração aquática por exemplo.

O controle de impedância estabelece ao robô um comportamento dinâmico no contato com o ambiente, assim, a ocorrência de erros de medida provoca uma resposta dinâmica e não uma falha no sistema.

Na proposta, as colisões indesejadas são prevenidas e tratadas com o controle de impedância. A impedância do sistema (inércia, rigidez e amortecimento) são alteradas de acordo com a aproximação de objetos diferentes daquele a ser manipulado. A detecção de obstáculos móvel será realizada por meio de métodos de *motion detection* (STAUFFER; GRIMSON, 1999). Com o uso de campos potenciais, por exemplo, é possível desviar de objetos, e ainda, utilizando mapas de impedância é possível tratar colisões, a impedância diminui os efeitos do impacto, e utilizá-las como auxílio à manipulação. Campos potenciais e mapa de impedância para prevenção e tratamento de colisões são baseados nos dados obtidos por um sistema de visão.

Os objetivos do trabalho foram estabelecidos considerando aplicações em ambientes aquáticos, no qual há possibilidades de colisão com animais e outros equipamentos (inclusive outros robôs trabalhando em cooperação) em movimentação devido às correntes marítimas. Ambientes aquáticos também estão sujeitos a erros de medida de posicionamento por meio de sistema de visão devido à refração do meio aquoso e à existência de partículas em suspensão.

1.3 Metodologia adotada

A objetivo principal deste trabalho é propor um sistema robótico capaz de executar tarefas de manipulação sem a necessidade de se considerar qualquer modelo de contato. Este objetivo foi baseado na afirmação de diversos pesquisadores da área de que mesmo os modelos mais avançados não são capazes de representar os fenômeno do contato, conforme discutido nas seções anteriores.

Além da dificuldade de representação do fenômeno de contato, as aplicações em ambientes não estruturados, como exploração aquática, possuem características de erros de medida dos sistema de visão. Essas características também são de difícil modelagem e consequentemente de difícil replicação em ambientes de simulação.

Devido ao fato de que não há modelos de contato confiáveis e às dificuldades de simulação das características dos erros de medida dos sistemas de visão. Tornou-se necessária a adoção de uma abordagem puramente experimental para o desenvolvimento e testes das propostas deste trabalho.

1.3.1 Definição da tarefa teste e métrica de desempenho

Para os experimentos foi eleita a tarefa de reorientação de um círculo no plano. A mesma tem sido utilizada por diversos trabalhos na área para avaliação de modelos de contato, controladores, planejadores de *regrasping* e de sincronismo entre dedos ou entre robôs trabalhando em cooperação (vide Stucheli et al. (2013)). A figura 1.1 mostra uma representação da tarefa na qual três robôs em cooperação, ou três dedos de uma garra, reorientam um círculo.

A tarefa considerada pode representar várias tarefas específicas como desrosquear uma lâmpada, abrir a tampa de uma garrafa, girar um volante, abrir um registro ou válvula, entre outros.

Figura 1.1: Exemplo da tarefa de manipulação eleita para os experimentos. Figura modificada de Xu et al. (2007).

1.3.2 Ambiente experimental

Robô SCARA

O dedo da mão humana apresenta uma cadeia cinemática que pode ser considerada como a melhor para realizar tarefas de manipulação. Considerando que a tarefa de manipulação eleita restringe-se à movimentação e forças em um plano, um robô do tipo SCARA de movimentação no plano foi selecionado como atuador robótico, pois no caso de movimentação no plano, o espaço de trabalho deste tipo de robô é semelhante ao de um dedo da mão humana.

Para efeitos de simplificação do ambiente experimental, porém sem perda da capacidade de generalização e de representação, apenas um robô foi selecionado para executar a tarefa.

Figura 1.2: Robô SCARA retrofitado e sua representação cinemática.

O robô, original da IBM modelo 7545, passou por um processo de *retrofitting* no qual seus sensores (*encoders* e fins de cursos) e *drivers* de potência foram atualizados. Os *drivers* foram selecionados com o propósito de permitir comandos de alto nível como controle de posição, velocidade e corrente dos motores. Foram selecionados os *drivers* modelos EPOS2 70/10 do fabricante Suíço Maxon. Adicionalmente, um sensor de forças e torques nas três direções, do fabricante ATI modelo Gamma, foi montado no TCP do robô para medida das forças de interação. Para aquisição das medidas do sensor Gamma foi utilizada uma placa de aquisição analógica da PHONEX de 16 *bits* de precisão e com módulo CANopen de comunicação.

A comunicação do sistema computacional de controle com as EPOS e com a placa de aquisição é realizada via CANopen. Suas características de confiabilidade, determinismo e velocidade de comunicação permitem a implementação de uma arquitetura com capacidade de tempo-real. Os controladores do robô foram implementados na linguagem C/C++. Detalhes de implementação são apresentados por Fernandes (2013).

Os dados geométricos do robô que são importantes para a implementação são: comprimento do primeiro elo $L_1 = 0,4m$; comprimento do segundo elo $L_2 = 0,25m$.

Sensor de visão Kinect

Como sistema de visão foi selecionado o sensor Kinect que foi originalmente desenvolvido para aplicações em jogos do Xbox 360 sendo lançado no mercado em setembro de 2010

(KINECT, 2011). Após seu lançamento, pelo seu baixo custo ele se tornou uma alternativa interessante como sensor 3D em substituição dos sistemas industriais nas aplicações em que não se requer longas distâncias de aquisição e grande precisão de medida (KHOSHELHAM, 2010).

Figura 1.3: Sensor de visão: Kinect.

O Kinect, desenvolvido pela PrimeSense (PRIMESENSE, 2011), possui dentre vários recursos, uma *webcam* e um *depth sensor* ambos com aquisição de 30 quadros por segundo com 640x480 e 320x240 *pixels* de resolução, respectivamente. Visto que o Kinect foi desenvolvido para jogos, pouca informação está disponível para aplicações em engenharia⁵, o que desfavorece sua confiabilidade.

O funcionamento do sensor 3D, de forma resumida, é baseado em um processo de triangulação. Um emissor de *laser* infravermelho emite um único feixe que é difratado em múltiplos outros por uma lente de difração. O padrão destes feixes projetados na cena é então capturado por uma câmera sensível apenas a comprimentos de ondas no infravermelho. O padrão detectado em uma aquisição da câmera de infravermelho é então comparado com o padrão obtido na calibração, sendo então possível determinar a posição x , y e z de cada *pixel* da câmera de infravermelho, obtendo-se assim uma matriz 320x240 de pontos tridimensionais. Uma descrição mais completa é realizada em Khoshelham (2010), que também apresenta um modelo matemático para o sensor. Khoshelham (2010) também apresenta uma primeira análise da acuidade, sendo que os principais resultados encontrados pelo autor são podem ser resumidos nos seguintes:

- Análise da distorção da câmera de infravermelho e disponibilização dos parâmetros necessários para a corrigir;
- Identificação das fontes de erro de medida: propriedades de superfícies da cena; e condições do ambiente medido como iluminação e sombras;

⁵Existem informações fornecidas para usuário de *games*.

- Análise da precisão de medida com um sensor 3D de alta resolução e acuidade de 0,7mm;
- Análise de erro de medida com aquisições a várias distâncias de um plano.

Apesar dos erros de medida do Kinect apresentados por Khoshelham (2010) serem elevados em comparação com os erros dos sensores similares utilizados na indústria e em sistemas robóticos de precisão, o Kinect foi selecionado como dispositivo de aquisição do sistema de visão com o objetivo de avaliar o desempenho e a robustez do sistema robótico proposto na presença de erros.

O Matlab foi selecionado para programação do processamento das imagens da câmera RGB e da *depth câmera* na implementação do sistema de visão deste trabalho.

Configuração do ambiente experimental

A figura 1.4 apresenta o ambiente experimental com a posição relativa entre o robô SCARA e o sensor Kinect, destacando seus sistemas de referência, e a mesa de trabalho na qual o objeto a ser manipulado é posicionado.

O objeto é um pote de 0,5kg de massa, e cuja tampa com $D = 0,113m$ de diâmetro é o objeto a ser manipulado. Vale ressaltar que não há nenhum tipo de fixação do pote com a mesa de trabalho.

Figura 1.4: Ambiente experimental.

A comunicação entre o ambiente de programação do sistema de controle do robô e do ambiente de programação do Kinect foi desenvolvida utilizando JSON (YANG; HU, 2008). Detalhes de implementação são apresentados por Fernandes (2013).

Métrica de desempenho

No caso específico deste trabalho, a tarefa consiste em realizar uma reorientação angular da tampa do pote, para tanto utiliza-se como elemento de medida um transferidor com $0,5^\circ$ de precisão, tendo-se como referência de posição inicial (ângulo 0° de giro) marcações no pote e na tampa. Assim, como principal métrica para análise do desempenho na execução da tarefa de manipulação eleita considera-se os seguintes casos:

Desempenho insatisfatório: quando o pote tombar, ou quando o mesmo transladar, ou cuja tampa apresentar reorientação igual ou inferior a 10% daquela desejada;

Desempenho parcialmente satisfatório: quando houver reorientação da tampa a partir de 10% até 80% (inclusive) daquela desejada, desde que o pote não apresente qualquer translação em relação à mesa;

Desempenho satisfatório: quando houver reorientação da tampa a partir de 80% até 100% daquela desejada, desde que o pote não apresente qualquer translação em relação à mesa.

As demais métricas de desempenho para os subsistemas do sistema de manipulação proposto, como por exemplo aquelas para desempenho do sistema de visão, são estabelecidas ao longo do texto para cada caso específico.

1.4 Organização do texto

Os próximos capítulos apresentam os seguintes conteúdos:

Capítulo 2: é apresentada uma proposta de sistema de visão baseado no sensor Kinect para a identificação de objetos móveis a serem manipulados e com possibilidade de colisão com o robô. Os resultados são utilizados pelo planejador de manipulação para evitar e tratar colisões, e para identificar a posição do objeto a ser manipulado;

Capítulo 3: neste capítulo é apresentada uma proposta de planejador de trajetórias suaves baseado no método *Squeezed Screw* e em perfis contínuos de velocidades para atuarem em combinação com o controle de impedância para estabelecimento de contato e posterior manipulação estáveis;

Capítulo 4: neste capítulo é apresentada a proposta de planejador de manipulação. Apresenta-se o controle de impedância utilizado no nível primário, e para o nível secundário os campos potenciais de força de repulsão para evitar colisões e o mapa de impedância para tratamento de colisões, ambos baseados nos dados do sistema de visão;

Capítulo 5: neste capítulo são apresentados três experimentos do planejador de manipulação. O primeiro consiste em um experimento de estabilidade de contato, no qual avalia-se a estabilidade do controle de impedância na manipulação. O segundo consiste em um experimento de manipulação utilizando o ambiente experimental descrito e a tarefa teste eleita. O terceiro experimento consiste na execução da tarefa de manipulação a partir da posição do objeto estimada a partir dos dados do sistema de visão.

Capítulo 6: neste capítulo são apresentadas as conclusões finais do trabalho e propostas para trabalhos futuros.

Apêndice A: neste apêndice é apresentada uma avaliação de repetibilidade e de precisão de medida do sensor Kinect. Os resultados desta avaliação são utilizados para justificativa da escolha do sensor como dispositivo de aquisição do sistema de visão do capítulo 2;

Apêndice B: neste apêndice é apresentada uma proposta de uso de diversas redes neurais para a classificação de objetos a partir de nuvem de pontos. Esta proposta não é utilizada no trabalho, porém, seus detalhes de implementação e respectivos resultados são apresentados para que possam ser utilizados em outros trabalhos.

2 *Sistema de visão*

2.1 Introdução

Na manipulação robótica, o uso de um sistema de visão é fundamental para que o robô detecte e identifique seu ambiente e os objetos nele contidos. A importância de um sistema de visão na manipulação robótica pode ser explicada por analogia com o sistema de visão humano e suas capacidades e características.

No sistema de visão humano, o processo inicia-se pela captura de imagens através dos olhos e posteriormente diversos processamentos são realizados pelo Sistema Nervoso Central (SNC) que vão desde a identificação de retas e contornos (ZEKI, 1993) até a classificação de objetos e determinação da melhor forma de pegá-los (MACKENZIE; IBERALL, 1994). Pesquisas apontam que os seres humanos apresentam posturas de mão específicas preferenciais para pegar objetos de acordo com a forma predominante deste e com a tarefa a ser executada ((CUTKOSKY, 1989)).

Assim, além de determinar a posição de objetos e de os classificar, através do sistema de visão os seres humanos são capazes de identificar ou estimar outras propriedades importantes como textura superficial, densidade, massa, volume e momento de inércia. O sistema de visão também permite o mapeamento do ambiente, por exemplo, alguns segundos de observação de um ambiente são suficientes para que um mapa virtual seja construído pelo SNC, contendo informações como presença e classificação de objetos, pessoas, animais, plantas, entre outros.

O sistema de visão também permite uma constante atualização do ambiente e detecção de modificações repentinas ou aproximação de objetos. Estes estímulos visuais são primeiramente detectados pelo sistema de visão periférico e são processados mais rapidamente que, por exemplo, a classificação de objetos ou mapeamento do ambiente. Quando um objeto se aproxima rapidamente de um observador, as informações seguem um caminho diferenciado pelo SNC que permite uma resposta rápida, geralmente na tentativa de evitar ou de amenizar os efeitos de uma colisão. Esta resposta reativa normalmente leva em torno de 200ms a 300ms e ocorre antes da classificação ou identificação do objeto (ZEKI, 1993) e (MACKENZIE; IBERALL, 1994).

Estas características observáveis do nosso SNC são importantes tanto para a concepção quanto para a implementação de sistemas robóticos de manipulação. Os sistemas robóticos de manipulação mais avançados apresentam diversos sensores 2D (câmeras, por exemplo) e 3D (sensores *depth range*, por exemplo), e a utilização de cada um deles, ou até mesmo a fusão de dados dos dois, pode ser determinada pela necessidade da aplicação ou pela resposta desejada. A detecção de objetos móveis para evitar e tratar colisões, por exemplo, é uma tarefa que exige rápido tempo de resposta, e neste caso, a grande quantidade de dados em uma única aquisição de uma cena inviabiliza o uso de dados de sensores 3D devido ao custo computacional.

Nas implementação de sistemas de manipulação, os sistemas de visão 2D são os mais utilizados, pois, em comparação com os sistemas 3D, apresentam menor necessidade de poder de processamento, menor custo de *hardware* e encontram-se facilmente bibliotecas disponíveis com algoritmos de processamento de imagens implementados para diversas linguagens de programação, como o OpenCV, por exemplo. Assim, os sistemas de visão 2D são utilizados na implementação de funcionalidades que necessitam de maior robustez e de velocidade de processamento como detecção de movimentação, identificação, segmentação, entre outros. Por sua vez, os sensores 3D apresentam maior quantidade de dados para um mesmo intervalo de aquisição e por sua vez são utilizados no processamento de funcionalidades de mais alto nível como informações volumétricas, identificação de alças, ou detalhes do ambiente para navegação no caso de robôs móveis.

As aplicações foco deste trabalho são sistemas de manipulação em ambientes não estruturados, no qual colisões com objetos, humanos e animais podem ocorrer com frequência significativa. Assim, o sistema de visão aqui proposto é destinado a detectar objetos móveis em um ambiente com a finalidade de evitar e tratar colisões indesejadas. As informações a serem obtidas com este sistema de visão, a saber: estimativa de posição e volume de objetos móveis, serão integrados ao sistema de manipulação para a formação de campos potenciais de repulsão e de mapa de impedância. Em sistemas de manipulação, não há propostas significativas de sistemas similares, normalmente o sistema de visão tem foco na identificação de objetos e na forma de pegá-los, normalmente colisões são tratadas como proibidas e evitadas a todo custo, e normalmente métodos de planejamento de trajetória livre de colisões são utilizados. Porém, vale ressaltar que na manipulação algumas colisões podem ocorrer, como por exemplos colisão entre dedos e colisão com superfícies de apoio no momento de pegar ou manipular um objeto. Assim, apresenta-se esta proposta como uma contribuição científica para a área.

O sistema de visão proposto deve identificar objetos móveis, estimar sua posição e seu volume no espaço de trabalho entre 200 e 300 *ms*. As informações são então disponibilizadas ao sistema de mais alto nível para evitar e, principalmente, tratar colisões sem que a tarefa de manipulação falhe por completo. Assim, o sistema de visão deve então ser simples, tanto em questão de dispositivo de aquisição necessário quanto em processamento, para garantir que o tempo de resposta requerido seja atendido.

O dispositivo de aquisição selecionado é o sensor Kinect, descrito em detalhes na seção 1.3.2 e avaliado no apêndice A, que apresenta dois dispositivos de aquisição: uma câmera com padrão cromático *red, green e blue* (RGB) e um sensor de profundidade tipo *depth range*. Para o uso deste sensor, inicialmente foi realizado um estudo de repetibilidade e precisão de aquisição, e os resultados obtidos estão aquém daqueles observados em sensores industriais utilizados em sistemas críticos, porém, as limitações de aquisição contemplam a proposta inicial deste trabalho de se desenvolver um sistema robótico capaz de executar tarefas de manipulação sem a exigência da utilização de sensores de elevada precisão, pois em algumas aplicações, como exploração aquáticas, erros de medida são encontrados devido às partículas em suspensão e à difração da luz em meio aquoso.

Para o uso do sensor também é necessário encontrar a matriz de transformação homogênea entre os sistemas de referência do Kinect e do robô SCARA, para que os dados obtidos com o sensor possam ser representados no espaço de trabalho do robô. Assim, apresenta-se um método para obtenção da matriz de transformação.

Em seguida, um método de detecção dos objetos móveis é implementado sobre a imagem RGB utilizando métodos convencionais de processamento de imagens com características de processamento rápido. Como existe o registro entre a imagem RGB e a imagem de profundidade, a detecção realizada sobre a imagem RGB pode ser utilizada como máscara para selecionar dentre todos os dados 3D quais são aqueles pertencentes ao objeto móvel, sendo assim possível estimar posição, volume e até mesmo classificar o objeto ou até mesmo executar tarefas em que informações 3D dos objetos móveis forem necessárias.

Para que a detecção de objetos móveis seja eficiente, é necessário detectar dentre estes dados quais são aqueles pertencentes ao robô, uma vez que o robô também se move. Um método de segmentação simplificado utilizando um modelo virtual poligonal do robô é apresentado.

O capítulo está organizado da seguinte forma: a seção 2.2 apresenta um procedimento para calibração do dispositivo de visão (o Kinect) com o robô, e um experimento foi conduzido para avaliar a qualidade da calibração; a seção 2.3 apresenta um método para detecção de objetos móveis no espaço de trabalho do robô; por fim, considerações do sistema de visão são realizadas na seção 2.5.

2.2 Calibração Kinect e robô SCARA

O robô SCARA e o sensor Kinect apresentam sistemas de referência distintos. O robô apresenta um sistema S fixo à sua base, no qual a posição de um ponto P representando seu ponto de ferramenta (TCP do Inglês *Tool Center Point*) pode ser representado por coordenadas x, y e z na forma $P^S = [x^s \ y^s \ z^s]^T$, e o sensor apresenta um sistema de referência K conforme mostra a figura 2.1. A calibração entre o SCARA e Kinect consiste em encontrar a matriz H_K^S de transformação homogênea tal que:

$$P^S = H_K^S P^K \quad (2.1)$$

Figura 2.1: Sistemas de referência do sensor Kinect e do robô SCARA.

A matriz H_K^S é necessária para descrever a posição 3D de cada um dos pontos de aquisição do sensor no sistema de referência do robô. Como não se pretende fixar o sensor em relação ao robô, a obtenção desta matriz deverá ser realizada sempre que o sensor for reposicionado,

ou antes de aplicações específicas para garantir que os dados do sensor sejam devidamente identificados no espaço de trabalho do robô.

A metodologia para determinação da matriz consiste em detectar através do Kinect pontos de identificação fixos ao robô conforme método de extração de posição 3D descrito na seção A.2.1. Para cada diferente posição i das juntas do robô, são registradas tanto a posição de um *light emitting diode* (LED) em relação ao robô, representada por $P_i^S = [x_i^S \ y_i^S \ z_i^S]^T$, quanto a posição do identificador obtida pelo Kinect $P_i^K [x_i^K \ y_i^K \ z_i^K]^T$. Obtendo n pontos é possível construir o seguinte sistema:

$$[P_1^S \ P_2^S \ \dots \ P_n^S] = H_K^S [P^K \ K_1 \ P_2^K \ \dots \ P_n^K] \quad (2.2)$$

Este sistema não possui solução exata devido, principalmente, à existência de erros de medida do Kinect e de erro de quantização ao se considerar que o LED representa um ponto espacial. O robô apresenta erro máximo de posicionamento de $0,013mm$. Sendo assim, os erros observados são provenientes do sensor e do procedimento de identificação da posição do LED identificador. Para a solução do sistema utiliza-se o método dos mínimos quadrados que encontra a solução para um sistema linear na forma $Ax = B$ minimizando a norma $|Ax - B|$.

Por meio de operações matriciais é possível demonstrar que a equação:

$$(P^K)^T (H_K^S)^T = (P^S)^T \quad (2.3)$$

é equivalente a equação 2.1, porém escrita na forma $Ax = B$, em que $(P^S)^T = B$, $(P^K)^T = A$ e $(H_K^S)^T = x$. Este sistema linear pode ser resolvido pelo Matlab com o comando »> mldivide(A,B)".

Fazendo-se medidas de posição dos LEDs com o Kinect para diferentes posições do robô é possível obter relações suficientes para determinar pelo método numérico citado a matriz de transformação.

Para que a calibração seja eficiente, ou seja, com valor reduzido para $|Ax - B|$, é preciso que A contenha o menor erro possível, uma vez que A são os valores do Kinect que apresenta erro de aquisição cerca de 100 vezes maior que aqueles apresentados no posicionamento do SCARA, cujo erro de posicionamento do seu TCP é de $0,5mm$). Assim, de acordo com os resultados de avaliação de precisão e de repetibilidade da aquisição do Kinect, os pontos de identificação para a calibração devem ser adquiridos o mais próximo possível do sensor. E ainda, várias aquisições devem ser realizadas para uma mesma posição do robô com a finalidade de reduzir os efeitos de erros aleatórios.

2.2.1 Experimento de qualidade de calibração entre SCARA e Kinect

Experimentos foram realizados para verificar a qualidade da matriz H_K^S obtida pelo método acima em função da distribuição dos pontos de calibração. Para exemplificação, a figura 2.2 apresenta os pontos utilizados no procedimento de calibração.

Figura 2.2: Resultado da calibração. Círculos representam a posição do LED e os pontos representam a posição medida com Kinect após transformação com a matriz H_K^S obtida.

Após a calibração, o LED de identificação foi posicionado ao longo de todo espaço de trabalho do robô em espaçamentos de $5mm$. Cada diferente posição do LED foi medida com o Kinect. Para eliminar erros aleatórios, foram realizadas 30 medidas para cada ponto e a média entre elas considerada como valor medido P^K . Posteriormente, a posição do LED descrita no espaço de trabalho do robô foi estimada utilizando a matriz de transformação estimada por $P_{est}^S = H_K^S P^K$ foi comparada com a posição real do LED P_{real}^S .

O erro de transformação dado pela diferença entre P_{real}^S é representado nas figuras a seguir. Cada um dos seus *pixels* representa um ponto de verificação. O erro é representado por uma tonalidade de cinza. Pontos brancos apresentam erro zero enquanto pontos preto representam erros de $40mm$ ou superior. As figuras 2.3, 2.4 e 2.5 apresentam o erro de transformação H_K^S ao longo do espaço de trabalho do robô SCARA para as componentes x , y e z , respectivamente.

Figura 2.3: Erro em x obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.

Figura 2.4: Erro em y obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.

Figura 2.5: Erro em z obtido com a matriz de transformação H_K^S ao longo do espaço de trabalho do robô SCARA.

2.3 Detecção de objetos móveis

Conforme já descrito anteriormente, a detecção de elementos móveis, que podem ser objetos inanimados, animais e seres humanos, e até mesmo outros robôs, é realizada sobre a imagem RGB do Kinect. O resultado final desta detecção é uma imagem binária com valores iguais a um (1) para os *pixels* pertencentes ao objeto identificado, e zero (0) para os *pixels* pertencentes ao fundo da cena. A imagem binária de identificação de objetos móveis é utilizada como máscara para extração de pontos 3D dos objetos móveis a partir dos dados 3D de uma cena completa uma vez que observa-se a existência de registro entre a imagem RGB e a *depth image* conforme mostra a figura 2.6. As bordas identificadas na imagem RGB (à esquerda) são sobrepostas à imagem de profundidade (à direita) mostrando o registro entre as imagens.

Figura 2.6: Exemplo de registro entre imagem RGB e *depth image* do Kinect.

Então, o sistema de visão apresenta como saída uma imagem binária de identificação, e uma nuvem de pontos referentes aos elementos identificados. Ambos os dados podem ser utilizados posteriormente nas funcionalidades dos níveis superiores do sistema robótico de manipulação como em tarefas de evitar e tratar colisões (*collision avoidance* e *collision handling* respectivamente) e também para classificar objetos.

A obtenção dos dados 2D e 3D dos objetos móveis é realizada pelas seguintes etapas:

1. Formação da imagem de fundo inicial;
2. Subtração de fundo da imagem (*background subtraction*);
3. Aplicação de filtros morfológicos para eliminação de ruídos;
4. Extração de dados 3D pela aplicação a imagem binária como máscara sobre a *depth image*;

5. Identificação dos pontos referentes ao robô;
6. Aplicação de filtros morfológicos para eliminar ruídos provenientes de erros de registro.

2.3.1 Métodos de subtração de fundo de imagens

Na literatura existem diversas propostas de métodos e algoritmos de processamento de imagens para subtração de fundos de cena (do Inglês *Background Subtraction*). Devido à grande quantidade de métodos e por este tema não ser o foco do trabalho, não é o objetivo desta seção trazer uma revisão completa dos métodos e de detalhes por exemplo. Faz-se aqui uma revisão dos principais métodos de extração de fundo de cena e uma comparação qualitativa baseada nos resultados obtidos por Piccardi (2004) com o objetivo de dar suporte e justificar a escolha do método selecionado para a subtração de fundo da imagem.

Subtração de fundo de cenas é um procedimento utilizado para detectar elementos móveis em uma sequência de imagens obtida por uma câmera fixa¹. A ideia principal é detectar modificações entre uma imagem corrente e uma imagem de fundo de referência, que deve ser livre de objetos móveis e deve ser atualizada na forma de adaptações às variações de luminosidade e à mudanças intrínsecas da cena como sombras, árvores ou superfícies de lagos, rios ou bandeiras balançando ao vento.

A maioria dos trabalhos nesta linha concentram esforços no desenvolvimento de métodos de atualização da imagem de referência que exijam pouco recurso computacional como processamento e memória, e que sejam eficientes as modificações da cena como aquelas citadas anteriormente. Após a atualização da imagem de referência, a subtração do fundo da cena pode ser simplificada, como por exemplo uma comparação lógica *pixel a pixel*, ou até mesmo mais complexas considerando medidas estatísticas de variação de cor em cada *pixel*.

Piccardi (2004) apresenta uma breve descrição e uma comparação qualitativa dos seguintes métodos:

1. *Running Gaussian average*;
2. *Temporal median filter*;
3. *Mixture of Gaussians*;
4. *Kernel density estimation*;

¹Estas técnicas não podem ser utilizadas em uma sequência de imagens de uma câmera em movimento.

5. *Sequential KD approximation*;
6. *Cooccurrence of image variations*;
7. *Eigenbackgrounds*.

De forma resumida, considerando requisitos de processamento e memória, os dois primeiros métodos são aqueles com menor exigência computacional. Os demais exigem mais recursos computacionais para implementação, tal como o *Mixture of Gaussians*, o qual tem sido implementado com técnicas de processamento paralelo utilizando placa gráficas da Nvidia.

Para a acurácia e precisão da subtração de fundo, Piccardi (2004) também traz uma comparação qualitativa. Segundo o autor, os dois primeiros métodos apresentam boa adaptação a mudanças lentas de iluminação, mas não apresentam bom desempenho na correção de mudanças rápidas do fundo da cena. Porém, quando este tipo de mudança ocorre de forma distribuída em pequenas regiões, o problema pode ser resolvido com o uso de filtros adequados.

Escolhe-se, então para prova de conceito, o método *Temporal median filter* como subtrator de fundo de imagens para detecção de objetos móveis neste trabalho.

2.3.2 O método de subtração de fundo *Temporal median filter*

Este método, proposto por Wren et al. (1997) trata cada um dos *pixels* da imagem de forma isolada. O método é baseado no cálculo de uma função de densidade de probabilidade Gaussiana de forma acumulativa, que é calculada a cada instante de tempo "*t*" da seguinte maneira:

$$\mu_t = \alpha T_t + (1 - \alpha) \mu_{t-1} \quad (2.4)$$

em que T_t é a intensidade de cada *pixel*; μ é a média calculada; e α é um escalar elegível que relaciona a velocidade de adaptação da média.

De forma similar, o desvio padrão σ_t para cada *pixel* pode ser calculado da seguinte forma:

$$\sigma_t = \sqrt{\beta (T_t - \mu_t)^2 + (1 - \beta) (\sigma_{t-1})^2} \quad (2.5)$$

em que β é um escalar elegível que relaciona a velocidade de adaptação de σ .

O método apresenta a vantagem de não ser necessário armazenar em memória os n últimos valores de intensidade de cada *pixel* e dos demais valores intermediários para o cálculo do desvio padrão.

Para cada nova imagem, a média μ (imagem de referência) e o desvio padrão são atualizados, sendo que a subtração é realizada segundo a inequação:

$$|T_t - \mu_t| > k\sigma_t \quad (2.6)$$

em que k é uma constante de ajuste. Se o valor da subtração for menor ou igual à comparação, o *pixel* é classificado como fundo, de onde origina-se o nome "subtração de fundo".

Ainda para este método, Koller et al. (1994) propõem melhorias no cálculo da média μ_t a partir de resultados da subtração anterior em $t - 1$:

$$\mu_t = M\mu_{t-1} + (1 - M)(\alpha T_t + (1 - \alpha)\mu_{t-1}) \quad (2.7)$$

em que M é um valor binário que assume 1 se o *pixel* pertencesse a um objeto detectado em $t - 1$ ou 0 caso contrário. Esta estratégia evita que um objeto seja acrescentado à imagem de referência.

Por fim, este método pode ser aplicado em imagens em tons de cinza ou em imagens RGB, desde que a subtração em cada camada seja devidamente combinada para a subtração final.

2.3.3 Filtros morfológicos

O método de subtração de fundo selecionado pode apresentar erros na ocorrência de modificação rápida de luminosidade, por exemplo. Este erro de subtração é observado na forma de ruídos na imagem binária final. Uma das formas de corrigir este problema é a aplicação de filtros sobre a imagem binária para eliminar os ruídos. Neste trabalho os filtros selecionados são formados por operadores morfológicos.

A morfologia matemática baseia-se na teoria de conjuntos e em operações lógicas para definir uma vasta gama de ferramentas úteis ao processamento de imagens, principalmente, imagens binárias, mas sendo aplicável também (com algumas adaptações) às imagens com tons de cinza ou colorida.

De um modo geral, todos os operadores morfológicos baseiam-se no conceito na convolução de um elemento estruturante sobre uma imagem. As operações básicas da morfologia matemática que foram aplicadas neste projeto são: a dilatação e a erosão.

A dilatação é a aplicação de um elemento estruturante de forma concêntrica sobre um conjunto definido de pontos (brancos ou pretos) em uma imagem, de maneira que o elemento estruturante adicione informação sobre a vizinhança destes pontos. Esta operação é utilizada

principalmente para preencher intervalos e lacunas indesejáveis na imagem.

A erosão é uma operação onde a aplicação do elemento estruturante ocorre analogamente à operação anterior, porém, ao invés de dilatar a vizinhança do ponto percorrido inserindo informação, o elemento retira informação (gerando erosão nas áreas percorridas). Esta operação é utilizada principalmente para eliminar detalhes irrelevantes, como ruídos, e abrir intervalos ou lacunas em regiões de conexão indesejada.

A operação de fechamento é uma operação derivada das operações de dilatação e erosão. Trata-se da operação onde é aplicada primeiramente uma dilatação seguida de uma erosão com o mesmo elemento estruturante. Esta sequência de operações visa restaurar conexões fracas entre objetos da imagem e preencher regiões.

2.4 Experimentos e resultados

2.4.1 Detecção de objetos móveis

Para a identificação dos objetos móveis, principal objetivo do sistema de visão deste trabalho, selecionou-se o método *Running Gaussian Average* como subtrator de fundo, que exige uma imagem de fundo inicial. Como o robô também é um dos objetos móveis da cena, é necessário formar uma imagem de fundo sem os seus elementos móveis. A figura 2.7(a) apresenta a imagem de fundo sem o robô obtida fazendo-se uma composição de duas imagens distintas, cada uma obtida com o robô em um de seus limites do espaço de trabalho. As imagens da figura 2.7 mostram o resultado da subtração de fundo identificando o robô e um segundo objeto na cena. A figura 2.7(a) apresenta a imagem de fundo, a figura 2.7(b) a imagem atual da cena e a figura 2.7(c) a imagem resultante da subtração de fundo.

Figura 2.7: Exemplo de identificação de objetos móveis na cena do robô utilizando o método *Running Gaussian Average* para subtração de fundo. A figura 2.7(a) apresenta a imagem de fundo, a figura 2.7(b) a imagem atual da cena e a figura 2.7(c) a imagem resultante da subtração de fundo.

A imagem resultante da subtração de fundo, figura 2.7(c), é utilizada como máscara sobre todos os pontos adquiridos pelo *depth sensor* do Kinect, e o resultado consiste na nuvem de pontos 3D dos objetos móveis presentes na cena. A figura 2.8 apresenta a nuvem de pontos da cena adotada como exemplo. É possível observar os pontos pertencentes ao robô, à cadeira, e ruídos provenientes tanto de problemas relacionados ao método de subtração, tal como sombras e variações de iluminação, quanto de erros de registro do Kinect.

Figura 2.8: Nuvem de pontos de objetos móveis.

2.4.2 Segmentação dos pontos e identificação do robô

O resultado final da detecção de objetos móveis inclui o próprio robô. Faz-se então necessário identificar, dentre os pontos móveis detectados, quais são aqueles pertencentes a este, para que então seja possível considerar apenas objetos móveis e implementar algoritmos de prevenção e tratamento de colisões do capítulo. Para esta identificação, utiliza-se para o robô SCARA um modelo poligonal simplificado conforme mostra a figura 2.9. É importante enfatizar que a utilização de um modelo mais refinado do robô foi descartado uma vez que existem erros consideráveis tanto de precisão e repetibilidade do Kinect, erros de calibração, e ainda a existência de cabos e elementos flexíveis cuja movimentação em função da movimentação do robô é de difícil previsão.

Figura 2.9: Modelo poligonal simplificado do robô SCARA.

A figura 2.10(a) apresenta uma segunda cena do ambiente de trabalho do robô com um objeto posicionado sobre uma mesa. A figura 2.10(b) apresenta a subtração de fundo para a cena com a aplicação de filtros morfológicos.

Figura 2.10: Exemplo de subtração de fundo. À esquerda a imagem original e à direita o resultado da subtração de fundo.

A figura 2.11 mostra os pontos 3D dos objetos móveis, incluindo o robô. A figura 2.12 apresenta a segmentação dos pontos 3D: em vermelho os pontos pertencente ao robô identificados por comparações com a distância Euclidiana entre os pontos e o modelo virtual do robô; em verde os pontos detectados como objetos móveis porém pertencentes a mesa e provenientes de erros devido a sombras e a erros de registro; em azul os pontos pertencentes ao objeto móvel de interesse.

Figura 2.11: Nuvem de pontos da cena da figura 2.10(a).

Figura 2.12: Exemplo de segmentação 3D. Em verde os pontos pertencentes à mesa, em azul os pontos pertencentes ao objeto, e em vermelho os pontos pertencentes ao robô.

2.5 Avaliação do sistema de visão

Os resultados obtidos no experimento de qualidade da calibração entre Kinect e SCARA indicam que o erro de calibração é menor nas proximidades do sensor e aumentam com a distância a este, chegando a erros de $40mm$ nas extremidades mais distantes do sensor. No experimento, várias medidas de um mesmo mensurando foram realizadas, eliminando assim erros de repetibilidade. A resposta deste experimento apresenta erros acumulativos de diversos erros: de precisão do Kinect; de distorção de câmera; do método de detecção de pontos de identificação; dos erros do método de calibração.

Na prática, os erros de calibração devem ser considerados juntamente com os erros de repetibilidade e de precisão. De forma geral, em convergência com os resultados de avaliação do sensor Kinect, as aplicações que requerem maior precisão de aquisição devem ser realizadas nas proximidades do sensor, de $1m$ a $1,5m$ de raio de distância.

Para aplicações de manipulação e *grasping* recomenda-se utilizar os dados adquiridos nas proximidades do sensor. Para implementação de métodos de desvio e de tratamento de colisões considera-se que é possível utilizar dados adquiridos de uma distância maior, uma vez que nestes casos há uma exigência menor de precisão.

Após a proposta e avaliação da calibração, foi apresentada uma proposta de método de processamento de imagens para detecção de objetos móveis e segmentação 3D cujo objetivo é obter uma nuvem de pontos do objeto móvel. Para prova de conceito, o método de processamento de imagens selecionado foi aquele que possui maior simplicidade de implementação e de menor eficiência dentre vários outros métodos de subtração de fundo de imagens.

As imagens resultantes do subtrator de fundo é aplicada como máscara sobre a *depth image*. O resultado são nuvens de pontos dos objetos móveis que incluem o próprio robô. A segmentação 3D tem o objetivo de dividir a nuvem de pontos naqueles pertencentes ao robô e aos demais objetos. Utilizando um modelo poligonal simplificado para representar o robô foi possível determinar o conjunto de pontos pertencentes a este através de verificação da distância Euclidiana entre ponto e modelo (poligonal).

Por fim, o resultado do sistema proposto neste trabalho, uma nuvem de pontos de objetos móveis descritos no espaço de trabalho do robô, pode ser utilizado para várias funcionalidades em aplicações em manipulação robótica, aplicação deste trabalho, ou em outras aplicações que se mostrarem convenientes.

3 *Planejador de trajetórias*

Este capítulo está organizado da seguinte forma: a seção 3.1 apresenta o planejador de rotas *Squeezed Screw*; a seção 3.2 apresenta uma proposta de perfil de velocidades para o *Squeezed Screw* com o objetivo de formar um planejador de trajetórias sem discontinuidades; a seção 3.3 apresenta resultados de uma comparação de trajetórias de *regrasping* geradas pelo *Squeezed Screw* e pelo RRT, um planejador amplamente utilizado em aplicações robóticas; por fim, a seção 3.4 apresenta os parâmetros utilizados para a geração de trajetória da tarefa de manipulação considerada neste trabalho.

3.1 Planejador de caminhos *Squeezed Screw*

O *Squeezed Screw* é um planejador de rotas para movimentos suaves (STüCHELI, 2009). A partir de pontos espaciais de partida e de chegada, e respectivas velocidades em direções também definidas, é definido um caminho elíptico e uma translação simultânea, conforme mostra a figura 3.1.

Figura 3.1: Exemplo de uma composição de rota com *Squeezed Screw* e sua elipse base (linhas contínuas) e sua translação (linhas tracejada).

3.1.1 Parâmetros de definição

Uma rota *Squeezed Screw* generalizada é constituída da superposição ao longo de uma elipse no espaço segundo um movimento linear em uma direção também espacial com uma relação de progressão entre os dois previamente definida.

Para o cálculo da rota são necessários os seguintes parâmetros: as coordenadas espaciais dos pontos de entrada e de saída e os vetores de direção de partida e de chegada. Parâmetros adicionais são aqueles correspondentes à elipse: o comprimento do semi-eixo menor; a excentricidade; e a orientação relativa ϕ' . Este último parâmetro é definido como o ângulo entre a direção do eixo principal da elipse e a direção bissecante às direções de partida e chegada (vide Stucheli et al. (2013)). Por fim, a variável *orb* especifica qualitativamente o comprimento utilizado do perímetro total da elipse base: '*shrt*' para até 0,5 comprimento do perímetro; '*long*' de 0,5 a 1,0 do perímetro; '*xlng*' entre de 1 a 1,5 de comprimento de elipse. Comprimentos maiores a 1,5 podem ser conseguidos pela composição de dois ou mais caminhos *Squeezed Screw*.

Figura 3.2: Ilustração das opções para a variável *orb*. (a) '*shrt*', (b) '*long*' e (c) '*xlng*'. Figuras de Stucheli et al. (2013).

É importante ressaltar que o *Squeezed Screw* também é capaz de gerar trajetórias convencionais utilizadas na robótica clássica. Por exemplo, círculos podem ser obtidos fazendo-se a excentricidade igual a 1, e retas podem ser conseguidas fazendo-se iguais as direções de partida e de chegada e utilizando uma elipse com eixo infinito.

O planejador de caminhos *Squeezed Screw* foi implementado em Matlab e funções encriptadas estão disponíveis em www.mecatronica.eesc.usp.br/wiki/index.php/SqzdScrw assim como um exemplo de cálculo.

3.2 Perfil de velocidades para o *Squeezed Screw*

Esta seção apresenta uma proposta de perfil de velocidade tangencial durante a rota gerada pelo planejador *Squeezed Screw* completando assim as informações necessárias para a composição de um planejador de trajetórias. O objetivo é iniciar a trajetória com uma velocidade inicial v_i , percorrer todo o deslocamento s do *Squeezed Screw* em um tempo T_T , e finalizar a trajetória com uma velocidade final v_f .

Para uma mudança suave da velocidade de movimentação, o polinômio do terceiro grau é escolhido, assim, o perfil de velocidades é dado por:

$$v_G(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 \quad (3.1)$$

em que v_G é a velocidade tangencial em cada instante de tempo de movimentação segundo um determinado trecho da rota do *Squeezed Screw*.

Os parâmetros a_0 , a_1 , a_2 e a_3 do polinômio do terceiro grau podem ser obtidos quando as velocidade inicial e final e suas derivadas (v_1 , v_2 , $\dot{v}_1$ e $\dot{v}_2$ respectivamente) são conhecidas.

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^2 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix}^{-1} \begin{bmatrix} v_1 \\ \dot{v}_1 \\ v_2 \\ \dot{v}_2 \end{bmatrix} \quad (3.2)$$

Para se descrever as velocidades em cada instante segundo uma rota s do *Squeezed Screw*, divide-se a movimentação em três trechos. O primeiro consiste em uma aceleração inicial da velocidade inicial v_i até uma velocidade de cruzeiro v_c em um tempo T_i , o segundo consiste em um movimento com velocidade v_c constante por um tempo T_c , e o terceiro consiste em uma segunda aceleração da velocidade de cruzeiro v_c para a velocidade final v_f em um tempo T_f . As velocidades sempre estão posicionadas tangencialmente à rota. Por simplicidade, as acelerações no início e no final do primeiro e do terceiro trecho são nulas.

Assim, para o primeiro trecho, com mudança de velocidade de v_i até v_c , os parâmetros do polinômio de terceiro grau são obtidos resolvendo o seguinte sistema:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T_i & T_i^2 & T_i^2 \\ 0 & 1 & 2T_i & 3T_i^2 \end{bmatrix}^{-1} \begin{bmatrix} v_i \\ 0 \\ v_c \\ 0 \end{bmatrix} \quad (3.3)$$

em que T_i é o tempo de execução da rota s .

A equação 3.4 apresenta o equacionamento do perfil de velocidades para o primeiro trecho

$$v_a(t_i) = v_i + \frac{3}{T_i^2}(v_c - v_i)t_i^2 + \frac{2}{T_i^3}(v_i - v_c)t_i^3 \quad (3.4)$$

em que t_i é um instante de tempo dentro do período de aceleração inicial T_i .

No segundo trecho a velocidade v_c tangencial a trajetória é constante.

Para o terceiro trecho, com mudança de velocidade de v_c até v_f , os parâmetros do polinômio de terceiro grau são obtidos resolvendo o seguinte sistema:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T_f & T_f^2 & T_f^3 \\ 0 & 1 & 2T_f & 3T_f^2 \end{bmatrix}^{-1} \begin{bmatrix} v_c \\ 0 \\ v_f \\ 0 \end{bmatrix} \quad (3.5)$$

A equação 3.6 apresenta o equacionamento do perfil de velocidades para aceleração final

$$v_d(t_f) = v_c + \frac{3}{T_f^2}(v_f - v_c)t_f^2 + \frac{2}{T_f^3}(v_c - v_f)t_f^3 \quad (3.6)$$

em que t_f é um instante de tempo dentro do período de aceleração final T_f .

Para completar o perfil de velocidades, a velocidade de cruzeiro v_c é calculada de forma que o deslocamento s da rota especificada pelo *Squeezed Screw* seja executada no tempo total $T_T = T_i + T_c + T_f$, ou seja, com aceleração inicial durante T_i segundos no primeiro trecho, movimentação com velocidade de cruzeiro durante T_c segundos segundo trecho, e aceleração final durante T_f segundos no terceiro trecho. Desta forma, o deslocamento s é dado por:

$$s = \int_{T_i} v_a(t)dt + v_c T_c + \int_{T_f} v_d(t)dt. \quad (3.7)$$

Integrando a equação 3.7 e substituindo as variáveis das Eqs. 3.4 e 3.6 obtém-se:

$$s = v_i T_i + (v_c - v_i)T_i + \frac{1}{2}(v_c - v_i)T_i + v_c T_c + v_c T_f + (v_f - v_c)T_f + \frac{1}{2}(v_c - v_f)T_f \quad (3.8)$$

isolando v_c :

$$v_c = \frac{\left(s - \left(\frac{v_i T_i + v_f T_f}{2}\right)\right)}{\left(T_c + \frac{1}{2}(T_i + T_f)\right)} \quad (3.9)$$

É importante ressaltar que v_c pode ser negativa ou positiva assim como as acelerações no

primeiro e no terceiro trecho. As várias possíveis combinações dependem da escolha dos parâmetros v_i , v_f , T_i , T_f , T_T e do deslocamento s do *Squeezed Screw*.

3.3 Comparação *Squeezed Screw* com o RRT

O RRT (JR.; LAVALLE, 2000)¹ é um dos algoritmos do estado da arte no planejamento de caminhos livres de colisão. É importante ressaltar que neste trabalho faz-se uma distinção entre planejador de caminhos e de trajetórias, sendo que apenas o segundo compreende perfis de velocidades e de aceleração durante o movimento.

O RRT, aplicado na geração de trajetórias de manipulação (SAUT et al., 2007) e (XU et al., 2007), utiliza uma geração randômica de caminhos possíveis para explorar o ambiente e gerar caminhos livres de colisões. Por esta característica randômica, a rota gerada é complexa e apresenta inúmeras mudanças repentinas de direção, tal como pode ser observado naquela gerada pelo RRT para reposicionamento em uma tarefa de manipulação apresentado por Xu e Li (2008) e mostrado na figura 3.3. Como a rota gerada é complexa, pode-se atingir velocidades e acelerações elevadas se houver restrições de tempo de percurso, ou pode-se atingir tempos elevados de execução se houver limitantes cinemáticos. Considerando que na manipulação há restrições de tempo impostas, por exemplo, pelo sincronismo entre os dedos, e que não se deseja elevadas velocidades para evitar suavizar problemas de colisões indesejadas, estes métodos tornam-se desinteressantes.

Uma outra limitação do RRT é a necessidade de conhecimento da geometria tanto do ambiente quanto do robô para que a rota seja gerada. Assim é necessário manter atualizada uma representação virtual do ambiente. Esta representação deve ser completa e rica em detalhes pois erros de posicionamento e oclusões podem levar o método a falhas. Oclusões são inerentes aos sistemas de visão em manipulação robótica, e ainda, manter um ambiente atualizado um ambiente virtual de um ambiente real altamente dinâmico e não estruturado tem um custo computacional elevado.

3.3.1 Comparação quantitativa

O exemplo de comparação quantitativa considerado é a tarefa de reorientar um objeto circular em um plano de 180° com uma mão com três dedos simétricos. Este exemplo tem sido considerado por pesquisas na área para avaliar estratégias de reposicionamento de dedos du-

¹ Recomenda-se a leitura de (JR.; LAVALLE, 2000) para detalhes matemáticos do RRT.

rante a manipulação (*finger gaiting*). A trajetória escolhida para a comparação foi retirada de (XU et al., 2007) no qual o RRT é utilizado.

Por simplicidade, a comparação é realizada para a movimentação de um dedo apenas (figura 3.3). O dedo tem dois graus de liberdade, o primeiro elo tem 200mm e o segundo tem 10mm de comprimento. O dedo deixa a superfície do objeto no ponto $D = [10 \cos 162^\circ \ 10 \sin 162^\circ \ 0]^T$ e atinge o objeto novamente no ponto $E = [10 \cos 82,5^\circ \ 10 \sin 82,5^\circ \ 0]^T$ (Coordenadas em cm). A figura 3.3 mostra a superfície do objeto circular é representada por um círculo em preto de raio de 10 cm. A rota RRT é representada em linha tracejada e a rota *Squeezed Screw* em linha cheia.

Os parâmetros de entrada para um *Squeezed Screw* para esta tarefa são os pontos D e E e $\mathbf{d}$ e $\mathbf{e}$, já definidos. Os vetores são tangentes ao objeto para que o contato com o objeto seja suave. Para permitir ainda uma circunavegação do objeto, o ângulo de elipse ϕ' é escolhido para ser $\pi/2$ ou seja, o eixo principal da elipse é perpendicular ao eixo $\overrightarrow{DE}$. O eixo b da elipse é de cerca de metade da distância $\overline{DE}$ e orb vale 'xlng' (uma vez que a velocidade tem de mudar de direção por mais de 360°). A excentricidade da elipse é $\epsilon 0,85$.

Figura 3.3: Caminhos das trajetórias comparadas. Em linha tracejada a rota do RRT (XU et al., 2007) e em linha contínua a do *Squeezed Screw*.

Para fins de comparação, a rota RRT (retirada de Xu et al. (2007)), é concluída com um planejador de trajetória polinomial de terceira ordem para cada segmento de reta. A velocidade em cada ponto de inicial e final de cada segmento de reta é considerado zero, pois a mudança de direção pode exigir acelerações infinitas caso a velocidade seja diferente de zero. O tempo total de execução é de 0,4s para ambos os casos. Para o RRT o tempo em cada segmento é proporcional ao seu comprimento. Uma vez que o tempo total de *regrasping* é o mesmo para ambos os planejadores de trajetória, é possível comparar tanto a velocidade (figura 3.4), a aceleração (figura 3.5), e o *jerk* (figura 3.6) para a primeira junta, considerando que resultados similares são encontrados para a segunda junta e para as juntas dos resultados apresentados por Xu et al. (2007) uma vez que as características das rotas de *regrasping* são semelhantes. Os perfis mostram claramente que os perfis cinemáticos do planejador *Squeezed Screw* são mais suaves que os do RRT, para a velocidade, aceleração e *jerk*, não havendo picos ou descontinuidades.

(a)

Figura 3.4: Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo *Squeezed Screw* e pelo RRT (XU et al., 2007). Velocidade angular.

(a)

Figura 3.5: Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo *Squeezed Screw* e pelo RRT (XU et al., 2007). Aceleração angular.

(a)

Figura 3.6: Comparação de trajetórias no espaço de configuração para a junta 1 geradas pelo *Squeezed Screw* e pelo RRT (XU et al., 2007). Jerk angular.

O RRT é um bom planejador quando o espaço de trabalho é complexo e possui muitos obstáculos conhecidos *à priori*. Ele é interessante quando se tem um espaço de trabalho com várias possibilidades de caminhos entre pontos. Nestas condições o *Squeezed Screw* não apresentaria bom desempenho. Se, contudo, o espaço de trabalho apresentar apenas alguns obstáculos (como por exemplo em uma condição em que apenas um objeto é manipulado) o *Squeezed Screw* tem a vantagem de que o obstáculo não precisa de representação detalhada. Uma estimativa aproximada da superfície do obstáculo é o suficiente para se definir parâmetros capazes de gerar um caminho capaz de circunavegar o obstáculo.

Outra desvantagem do RRT é o número de computações necessárias para construir uma árvore, em comparação com o cálculo de um *Squeezed Screw*. O custo computacional torna-se especialmente elevado, se considerarmos um espaço 3D no qual objetos estão se movendo. Com todas as mudanças dos obstáculos no espaço de trabalho, por exemplo, através de uma reorientação do objeto manipulado, uma nova árvore tem que ser construída a cada movimentação.

Uma terceira questão é a mudança de direção na trajetória. No RRT o componente aleatório de geração da árvore no planejamento é susceptível de produzir uma trajetória com muitas mudanças de direção. A mudança máxima de direção de um *Squeezed Screw* é cinco, no máximo, para o espaço 3D. Assim, a trajetória *Squeezed Screw* é mais rápida do que a trajetória RRT se ambas têm os mesmos limites de velocidade e aceleração.

3.4 Exemplo de geração de trajetória

Para exemplo de geração de trajetória com o *Squeezed Screw*, foi utilizada a tarefa considerada para os testes de manipulação deste trabalho. Especificamente, a tarefa consiste em girar a tampa de um pote (vide figura 2.10(a)), em 180° por duas rotações consecutivas de 90°. O robô inicia a tarefa a partir do repouso em um ponto qualquer *I*, estabelece contato com o objeto no ponto *A* com velocidade com módulo de 0,05m/s, em seguida executa o primeiro giro de 90° com velocidade constante e então perde contato com o objeto no ponto *B*. O robô executa o movimento de reposicionamento (em inglês *ginger gaiting*) estabelece um novo contato no mesmo ponto *A*, com a mesma velocidade. Então o robô repete o mesmo movimento de giro de 90° e finaliza a manipulação retornando ao ponto *I*. A figura 3.7 apresenta os pontos *I*, *A* e *B* e a posição do objeto no espaço de trabalho do robô.

Figura 3.7: Trajetória de manipulação usando os dados apresentados na tab. 3.1.

Para gerar a trajetória de manipulação, o pote foi posicionado no ponto do espaço de trabalho do robô quando as duas primeiras juntas do robô apresentam posicionamento angular de $\theta_1 = 45^\circ$ e $\theta_2 = 90^\circ$, sendo θ_1 e θ_2 a posição angular dos elos 1 e 2 do robô respectivamente. As direções tangenciais de estabelecimento de contato no pontos *A* e *B* são -45° e $+45^\circ$ em relação ao eixo *x*.

A trajetória de manipulação foi gerada a partir de 5 subtrajetórias de *Squeezed Screw*. O acoplamento entre as trajetórias é possível impondo que as velocidades e direções do início e do final de trajetórias vizinhas sejam iguais em módulo e direção. As tabelas 3.1 e 3.2 apresentam respectivamente os parâmetros cinemáticos e geométricos para as 5 subtrajetórias.

Tabela 3.1: Parâmetros cinemáticos para as subtrajetórias.

	Início			Fim			
	Ponto	Dir.	v_i	Ponto	Dir.	v_f	T_T
SqdScw 1	<i>I</i>	180°	0	<i>A</i>	45°	v	2s
SqdScw 2	<i>A</i>	45°	v	<i>B</i>	135°	v	2s
SqdScw 3	<i>B</i>	135°	v	<i>A</i>	45°	v	4s
SqdScw 4	<i>A</i>	45°	v	<i>B</i>	135°	v	2s
SqdScw 5	<i>B</i>	135°	v	<i>I</i>	180°	0	4s

Sendo que: $I = [160 \ 360 \ 0]^T mm$; $A = [139 \ 427 \ 0]^T mm$; $B = [139 \ 492 \ 0]^T mm$; e $v = 50 mm/s$. A direção é representada por um ângulo diretor no plano xy do sistema de referência da base do robô. Os tempos de movimentação do primeiro e do terceiro trecho, T_i and T_f respectivamente, são 49% do tempo total T_T para minimizar as acelerações.

Tabela 3.2: Parâmetros geométricos do *Squeezed Screw* para as subtrajetórias.

	$b[mm]$	ϵ	$\phi'[rad]$	orb
SqdScw 1	47	0,1	0	'shrt'
SqdScw 2	47	0,1	0	'shrt'
SqdScw 3	47	0,1	0	'xlng'
SqdScw 4	47	0,1	0	'shrt'
SqdScw 5	47	0,1	-1	'long'

A figura 3.7 apresenta a trajetória gerada, o espaço de trabalho do robô, e os pontos A , B e I . O perfil de velocidades do TCP durante a trajetória de manipulação é apresentado na figura 3.8.

Figura 3.8: Perfil de velocidade tangencial na trajetória.

4 *Planejador de manipulação*

4.1 Introdução

Nas aplicações industriais o ambiente de manipulação é estático ou muito pouco dinâmico, normalmente o mapa do espaço de trabalho do robô e a geometria dos objetos manipulados são bem conhecidas. Elas também são caracterizadas pela minimização de problemas de incertezas de posicionamento e de erros de medida pelo uso de sensores de elevada precisão e repetibilidade. Por todas estas características estes sistemas permitem o uso de controladores e planejadores simplificados, como por exemplo controladores PID e trajetórias ponto-a-ponto no espaço de junta, ponto-a-ponto por segmento de retas, ou combinações de arcos de circunferência com segmentos de retas.

Nas demais aplicações onde não há controle total do ambiente, como na manipulação em ambientes não estruturados, os planejadores industriais não são adequados, e outros tipos de controladores tem sido utilizados como controle de força (SICILIANO; VILLANI, 2000), controle híbrido de força e posição (RAIBERT; CRAIG, 1980), *kinestatic control* (GRIFFIS., 1991), *compliant control* (KAZEROONI; HOUP., 1986), entre outros. Nestas propostas os esforços são concentrados no desenvolvimento de controladores para as etapas da manipulação em que o robô está em contato com o objeto. Poucas propostas consideram as demais etapas e a mudança de condição de contorno como o estabelecimento e a perda do contato. Alguns autores adotam estratégias que utilizam troca da lei de controle em função do estado do robô (HYDE et al., 1997; OKAMURA et al., 2000). Por exemplo, pode-se utilizar controle de posição na movimentação livre e controle de força durante a manipulação.

A troca entre controladores deve considerar algumas características dinâmicas do estabelecimento e perda de contato. Antes do contato o robô movimenta-se livre, sem restrições. No instante do contato há uma colisão com consequente transferência de energia entre garra e objeto juntamente com a imposição de restrições cinemáticas e novas condições dinâmicas. Após o contato, a dinâmica do sistema é diferente daquela anterior. Esta nova dinâmica pode ser diferente entre objetos, ou pode ser variável para um objeto heterogêneo, como objeto com densidade variável ou flexível. Assim, a troca de controle deve ser sincronizada com o estabele-

cimento e perda do contato, que pode ser realizada pela detecção da colisão utilizando sensores de força, ou previsão da colisão utilizando sistemas de visão.

As duas estratégias citadas apresentam várias desvantagens. Na detecção com sensores de força é necessário que a colisão tenha ocorrido para ser detectada, ou seja, o estabelecimento do contato ocorre com o robô operando em controle para movimentação livre, o que pode desestabilizar ou provocar falhas do sistema. Na previsão do contato por sensores de visão é necessário reduzir erros de medida existente nos sensores e também ter um modelo detalhado do ambiente de trabalho, o que demanda muito poder computacional. Neste cenário, o controle de impedância tem sido utilizado como alternativa para o problema de chaveamento entre estados Goldenberg (1988), Schneider e Cannon (1992), Bonitz e Hsia (1996), Moosavian e Papadopoulos (1998), Okamura et al. (2000), Xu e Li (2008), Lippiello et al. (2007), e Schlegl et al. (2001),

De forma resumida, o controle de impedância estabelece uma relação dinâmica entre a força de interação do robô e seu deslocamento de uma trajetória estabelecida (posição no tempo). Desta forma, o controle de impedância trabalha sobre um trajetória. Stucheli et al. (2013) aponta como desvantagem a utilização de trajetórias provenientes da robótica industrial, estratégia comumente em diversos trabalhos na área. Assim, utiliza-se as trajetórias suaves geradas pelo planejador *Squeezed Screw* apresentado no capítulo 3.

Neste capítulo é apresentado um planejador de manipulação fundamentado na combinação de controle de impedância com trajetórias suaves, que são obtidas por modificações no algoritmo de geração de rotas *Squeezed Screw* (STUCHELI et al., 2013). A suavidade nas mudanças de direção e velocidade de movimentação, principalmente no estabelecimento e na perda do contato, permite a mudança dos parâmetros do controlador (massa, mola e amortecimento) na transição de estados sem a necessidade de chaveamento entre controladores. Espera-se com tal estratégia obter maior estabilidade na mudança de estados da manipulação. O planejador proposto, além de realizar tarefas de manipulação em diferentes fases, também apresenta funcionalidades de desvio e tratamento de colisões baseados em dados obtidos por um sistema de visão.

Este capítulo está organizado na seguinte forma: a seção 4.2 apresenta a lei de controle utilizada e seu equacionamento; a seção apresenta a arquitetura do sistema de manipulação, ou seja, a combinação de trajetórias suaves (seção 4.3.1) com funcionalidades de manipulação, e de desvio e tratamento de colisões (seção 4.3.2).

4.2 Controle de Impedância

4.2.1 Contextualização

O Controle de Impedância foi proposto por Hogan (1985) com objetivo de estabelecer uma lei de controle capaz de controlar sistemas robóticos de interação física com o ambiente em diferentes condições de trabalho: movimentação livre no espaço; estável no estabelecimento e perda de contato; e interação com objetivo rígidos ou flexíveis.

Hogan (1985) na época afirmava que um dos maiores desafios na área de manipulação é a estabilidade e controle do contato entre robôs e objetos. Sendo o controle de impedância uma proposta para controle de interação sem a necessidade de domínio ou de um modelo sofisticado do fenômeno do contato. Apesar dos diversos avanços da área, ainda não há um modelo de contato capaz de representar fielmente o contato físico, um dos motivos pelo qual o controle de Impedância é amplamente utilizado na área.

Os controladores baseados em controle da movimentação (posição e suas derivadas), apresentam elevada rigidez quando em contato com objetos. Por isso podem se tornar instáveis ou de comportamento indesejável, aplicando forças elevadas quando em contato com objetos e superfícies com restrições de movimentação, por exemplo. O problema consiste no fato de que estes negligenciam a dinâmica do ambiente e do contato. Se o objeto a ser manipulado apresentar restrições de movimentação, por exemplo, certos movimentos podem não ser possíveis e a força aplicada nele pode atingir valores elevados.

Quando um objeto está livre para a movimentação, muitas vezes não se observa neste a mesma movimentação imposta pelo efetuador do robô, pois o objeto reage às forças de contato e não há movimentação do robô. Assim, Hogan (1985) define como admitância um sistema que admite forças como entradas e apresenta movimentação como saída, sendo, neste caso, o ambiente e seus objetos sistemas de admitância.

A partir da definição de admitância, torna-se razoável deduzir que o melhor controlador para um manipulador seria um controle de força, porém, este tipo de controlador negligencia a movimentação do objeto na tentativa de manter a força de referência desejada para a execução da manipulação. Além disso, os controladores de força também apresentam instabilidade quando em contato com elementos de elevada rigidez (LAWRENCE, 1988)(EPPINGER; SEERING, 1992).

Uma vez que os objetos apresentam uma movimentação a partir da aplicação de forças, e a partir da consideração que o robô e o objeto quando em contato formam um sistema dinâmico acoplado, Hogan (1985) sugere que o robô deve apresentar um comportamento complementar a admitância do objeto, ou seja, o robô deve admitir movimentação como entrada e forças como saída. Esse comportamento desejado para o manipulador é definido como Impedância.

O conceito geral desta abordagem não está concentrada no controle da posição ou no controle da força, mas na relação destas duas variáveis. O comportamento de impedância pode ser conseguido atribuindo ao controlador propriedades dinâmicas de um sistema: inércia, rigidez e amortecimento. Desta forma, o manipulador pode trabalhar desde movimentação livre na qual as forças com o meio ambiente são próximas de zero ($\partial W = F \partial x \approx 0$) até o caso em que o trabalho realizado pelo robô sobre o ambiente não é zero ($\partial W = F \partial x \neq 0$). Modificando os parâmetros de impedância é possível ajustar a dinâmica do acoplamento entre robô e objeto conforme a necessidade.

Com base na capacidade de controlar a relação entre força e deslocamento, o controlador é capaz de estabelecer e perder contato de forma estável pois efeitos dinâmicos como tempo de resposta, oscilações e tempo de acomodação deixam de ser indesejáveis e evitáveis e passam a fazer parte do comportamento do sistema. O controle de impedância também evita a troca entre controladores para diferentes condições de trabalho do robô.

4.2.2 Equacionamento

O controle de impedância estabelece uma relação entre forças externas e deslocamentos do robô:

$$F_{ext} = M\ddot{x} + B(\dot{x} - \dot{x}_0) + K(x - x_0) \quad (4.1)$$

em que: F_{ext} são as forças externas; x é o deslocamento; M é a inércia; B é o amortecimento; K é a rigidez.

De forma generalizada, quando um robô está em movimentação e em contato com o ambiente por meio de forças externas o equacionamento dinâmico de um robô é dado por:

$$I(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = \tau_{act} - J^T(\theta)F_{ext} \quad (4.2)$$

em que: $I(\theta)$ é a inércia do sistema; $C(\theta, \dot{\theta})$ são os termos de Coriolis e forças centrípetas; τ_{act} são os torques dos atuadores sobre os elos (após redução); J^T é o jacobiano transposto.

Fazendo $\tau_{act} = J^T F_{act}$ e substituindo em 4.2, tem-se:

$$I\ddot{\theta} + C\dot{\theta} = J^T (F_{act} - F_{ext}) \quad (4.3)$$

A dependência de I , C e J em função de θ e suas derivadas foi suprimida para simplificar a notação.

Isolando $\ddot{\theta}$:

$$\ddot{\theta} = I^{-1} (J^T (F_{act} - F_{ext}) - C\dot{\theta}) \quad (4.4)$$

Se a relação entre as velocidades das juntas $\dot{\theta}$ e as velocidades do robô no espaço cartesiano $\dot{x}$ é dada por:

$$\dot{x} = J\dot{\theta} \quad (4.5)$$

Então a relação entre as acelerações $\ddot{x}$ e $\ddot{\theta}$ é dada por:

$$\ddot{x} = J\ddot{\theta} + \dot{J}\dot{\theta} \quad (4.6)$$

Substituindo $\ddot{\theta}$ por 4.4:

$$\ddot{x} = JI^{-1}J^T (F_{act} - F_{ext}) - JI^{-1}C\dot{\theta} + \dot{J}\dot{\theta} \quad (4.7)$$

O termo $JI^{-1}J^T$ é denominado como tensor de mobilidade do robô e denotado por W , assim:

$$W = JI^{-1}J^T \quad (4.8)$$

Para resolver a lei de controle, isola-se a força necessária nos atuadores, assim:

$$F_{act} = W^{-1}(\ddot{x} + JI^{-1}C\dot{\theta} - \dot{J}\dot{\theta}) + F_{ext} \quad (4.9)$$

A aceleração $\ddot{x}$ requerida para o comportamento de impedância é obtida manipulando a equação 4.1:

$$\ddot{x} = M^{-1}(K(x_0 - x) - B(\dot{x} - \dot{x}_0) - F_{ext}) \quad (4.10)$$

Substituindo 4.10 em 4.9, tem-se:

$$\begin{aligned} F_{act} = & W^{-1}M^{-1}(K(x_0 - x) - B(\dot{x} - \dot{x}_0)) + \dots \\ & \dots + W^{-1}(JI^{-1}C\dot{\theta} - \dot{J}\dot{\theta}) + (1 - W^{-1}M^{-1})F_{ext} \end{aligned} \quad (4.11)$$

Transformando de F_{act} para τ_{act} com a relação $\tau_{act} = J^T F_{act}$ tem-se:

$$\tau_{act} = J^T W^{-1} M^{-1} (K(x_0 - x) - B(\dot{x} - \dot{x}_0)) + J^T W^{-1} (JI^{-1} C \dot{\theta} - J \dot{\theta}) + J^T (1 - W^{-1} M^{-1}) F_{ext} \quad (4.12)$$

Finalmente, o termo x pode ser descrito em termos das coordenadas θ s das juntas pela cinemática direta $x = L(\theta)$, e o termo $\dot{x}$ pode ser substituído por $\dot{x} = J \dot{\theta}$. Assim, tem-se:

$$\tau_{act} = J^T W^{-1} M^{-1} (K(x_0 - L(\theta)) - B(J \dot{\theta} - \dot{x}_0)) + J^T W^{-1} (JI^{-1} C \dot{\theta} - J \dot{\theta}) + J^T (1 - W^{-1} M^{-1}) F_{ext} \quad (4.13)$$

Dadas medidas de posição e velocidade das juntas do robô e da força de interação do robô com o ambiente é possível controlar os torques nos motores para que o robô apresente um comportamento dinâmico de interação com o sistema em função de M , K e B desejados.

O controle de impedância apresenta as seguintes características: 1) Parâmetros de impedância podem ser mudados dinamicamente; 2) Não há necessidade de cálculo de cinemática inversa; 3) Medida de forças externas não é mandatório.

4.3 Arquitetura do planejador de manipulação

Segundo Okamura et al. (2000), o sistema de manipulação é formado por três níveis: primário, secundário e terciário. Nesta proposta considera-se que o nível terciário é responsável pelo planejamento e correções da trajetória da manipulação (perfis de forças e velocidades no tempo). Seguindo a abordagem da manipulação centrada no objeto, através de equações dinâmicas é possível encontrar as forças necessárias para que este apresente a movimentação desejada.

Como o foco deste trabalho é o nível secundário, a trajetória considerada será simplificada. Problemas de sincronismo entre dedos, por exemplo, não são considerados. O nível secundário é responsável pelas tarefas de prevenção e tratamento da colisão, e pela fusão de comportamentos. A prevenção e o tratamento da colisão é conseguido a partir de análise de dados obtidos pelo sensor de visão Kinect. A figura 4.1 apresenta um diagrama em blocos da arquitetura do sistema, em que: x^d , $\dot{x}^d$ e F_{manip}^d são respectivamente a movimentação e as forças desejadas para a manipulação; M^d , K^d e B^d são os parâmetros desejados para massa mola e amortecimento do controle de impedância; M^m , K^m e B^m são os parâmetros de impedância da posição x^r do robô no mapa de impedância; F_{rep}^d é a força de repulsão da posição x^r do robô; x^{dc} e $\dot{x}^{dc}$ são posição e velocidade desejadas corrigidas; M^c , K^c e B^c são os parâmetros de impedância corrigidos.

A figura 4.1 apresenta a arquitetura do sistema destacando as informações e dados transmitidos entre os três níveis da manipulação.

Figura 4.1: Diagrama em blocos do sistema de manipulação proposto.

4.3.1 Planejador de trajetórias suaves para a manipulação

Trajetória cinemática

Hogan (1989) sugere que tarefas de manipulação robótica, em que ocorrem estabelecimento e perda de contato, podem ser realizadas de forma eficiente desde que os parâmetros do controle de impedância sejam adequadamente ajustados. Muitos são os trabalhos que propõem estratégias de mudança de impedância Goldenberg (1988), Schneider e Cannon (1992), Bonitz e Hsia (1996), Moosavian e Papadopoulos (1998), Okamura et al. (2000), Xu e Li (2008), Lippiello et al. (2007), Schlegl et al. (2001). Uma característica comum aos trabalhos é, novamente, a carência de planejadores de trajetória durante *finger gaiting* and *regrasping*.

Stücheli (2009) aponta como desvantagem uma característica comum das propostas baseadas nestes controladores que é a utilização de trajetórias retangulares advindas da robótica industrial na movimentação do robô livre no espaço, como em Ohka et al. (2009) e Xu e Li (2008), ou até mesmo negligenciada em muitos outros casos. Assim, evidencia-se a carência de planejadores de trajetórias cinemáticas adequados, e ainda, mesmo quando existem, as trajetórias são chaveadas junto com os controladores no momento de estabelecimento do contato.

Estas trajetórias retangulares podem induzir a elevadas velocidades e acelerações do sistema ou tempos elevados de execução. Estes problemas são agravados em casos de mudança entre segmentos de reta em direções diferentes. A existência de interpoladores torna as trajetórias mais suaves, mas não resolve efetivamente estes problemas.

A suavidade das trajetórias é importante para evitar velocidades e acelerações elevadas e também para permitir a modificação dos parâmetros do controle de impedância de forma estável. As forças de manipulação podem ser obtidas pelas equações dinâmicas:

$$F_{obj} = m_{obj} \cdot \ddot{x}_{obj} \quad (4.14)$$

e

$$\tau_{obj} = I_{obj} \cdot \ddot{\alpha}_{obj} + \ddot{\alpha}_{obj} \times I_{obj} \dot{\alpha}_{obj} \quad (4.15)$$

em que m_{obj} é a massa do objeto; I_{obj} o tensor de inércia; F_{obj} e τ_{obj} são os vetores de força e torque para o movimento; $\ddot{x}_{obj}$ e $\ddot{\theta}_{obj}$ as acelerações linear e angular, respectivamente; e $\dot{\theta}_{obj}$ a velocidade angular do corpo. Por meio de relações cinemáticas é possível determinar as acelerações lineares e angulares em cada contato, e por meio da teoria de carregamentos equivalentes pode-se calcular as forças necessárias em cada contato. No caso de vários contatos pode-se utilizar teorias de minimização. Neste trabalho por motivos de simplificação utiliza-se apenas um ponto de contato na manipulação, assim simplifica-se a resolução das equações 4.14

e 4.15.

Para que a manipulação seja realizada, forças devem ser aplicadas sobre o objeto. Como a manipulação é focada no objeto (OKAMURA et al., 2000), dada uma movimentação desejada e as características dinâmicas do objeto como massa e momentos de inércia, as forças podem ser calculadas. Contudo, o controle de impedância segue como referência uma trajetória desejada. Para que a força de manipulação seja aplicada utilizando este controlador, uma estratégia para se conseguir estas forças é provocar um deslocamento específico da trajetória desejada (PRATT et al., 2004). O deslocamento é calculado a partir das forças e dos parâmetros do controlador (PRATT et al., 2004). Desta forma, as forças são reguladas intrinsecamente pelo controle de impedância. A figura 4.2 mostra a trajetória modificada, sendo que na trajetória original o contato ocorre tangencialmente à tampa do pote.

Figura 4.2: Trajetória de manipulação com deslocamento da trajetória inicial que tangencia o círculo que representa a tampa do pote.

Perfil de impedância

O planejador de manipulação utilizada a modificação dos parâmetros do controle de impedância em função do estado do robô, que ocorre segundo a trajetória contínua e suave do TCP do robô gerada pelo método do capítulo 3. Inicialmente, os parâmetros da impedância são pré-estabelecidos para toda a movimentação. Assim, além da movimentação, valores de massa, rigidez e amortecimento também são adicionados à trajetória. A mudança suave dos parâmetros é realizada utilizando um polinômio do terceiro grau. Os critérios para escolha dos parâmetros em cada um das fases da manipulação são discutidos no capítulo 5.

4.3.2 Desvio e tratamento da colisão

Em uma tarefa de manipulação em um ambiente não controlado e dinâmico, colisões indesejadas podem ocorrer, sendo que elas podem estar ou não estar relacionadas à tarefa de manipulação. Assim, um método para evitar colisões é necessário. Existem diversos trabalhos que propõem sistema de *collision avoidance*, podendo destacar como um dos primeiros aquele proposto por Khatib (1986), e mais recentemente o RRT.

O RRT, tal como já descrito, possui desvantagens de custo computacional e de geração de trajetórias formadas por segmentos de retas. O RRT constrói caminhos livres de colisões a partir de um mapa do ambiente, se o ambiente for dinâmico, novos caminhos devem ser calculados constantemente, o que requer mais custo computacional. Outra desvantagem, as trajetórias formadas por segmentos de retas podem induzir velocidades e acelerações elevadas, o que aumenta o consumo de energia durante a movimentação e ainda pode colocar em risco a vida de seres humanos (HADDADIN et al., 2008).

Evitar colisões é importante para obstáculos móveis e imóveis. Porém, considerando que para robôs de interação com o homem não é desejado que as velocidades de movimentação ultrapassem um limite (HADDADIN et al., 2008), colisões irão ocorrer caso algum objeto se mova em direção ao robô com uma velocidade maior que aquela permitida para que o mesmo não coloque em risco a integridade e a vida de um ser humano.

Assim, o robô deve tentar evitar colisões, mas sem a exigência de evitá-las sempre, pois não se deseja atingir velocidades elevadas caso obstáculos se movam rapidamente. No caso em que não é possível evitar uma colisão, o robô mesmo deve ser capaz de reagir à colisão de forma apropriada para evitar danos materiais e corporais. Trabalhos recentes nestas linhas tem sido desenvolvido seguindo esta filosofia, tal como o proposto por Haddadin et al. (2010).

Para reagir a uma colisão é necessário ou prevê-la ou detectá-la. Como já discutido na introdução deste trabalho, detectar a colisão não é conveniente pois para isso é necessário que ela ocorra. Prever é mais eficiente, e para isso por vezes é necessário conhecer a dinâmica de todo o ambiente, o que pode ter elevado custo computacional. E ainda, para prever uma colisão é necessário utilizar sistemas de visão, que são sensíveis a erros de medida devido a dificuldade de ajustes, correções, e calibração. Sistemas de maior precisão são mais eficientes, porém tem custo elevado.

A seguir é apresentada uma proposta de sistema de nível primário capaz resolver ao mesmo tempo *collision avoidance* e *collision handling*. O método utiliza como base campos potenciais e a modificação dos parâmetros massa, mola e amortecimento do controle de impedância. A modificação dos parâmetros ocorre em função dos obstáculos fixos, que podem ser estabelecidos em função de uma mapa do ambiente do robô, e dos obstáculos móveis que podem ser estimados a partir do sistema de visão. Este sistema é caracterizado por um comportamento reativo do nível primário, assim, as tarefas mais críticas são resolvidas no mesmo nível do controlador.

A modificação do comportamento do controlador é realizada por um mapa de impedância e um campo potencial de forças de repulsão. O mapa e o campo são formados sobre o espaço de trabalho do robô a partir das detecções de objetos móveis descritos na seção 2.3. Dois mapas de impedância e dois campos potenciais são formados. Um mapa e um campo são estáticos e estão relacionados ao espaço de trabalho estático do robô : obstáculos fixos e limites do espaço de trabalho. O mapa e o campo restantes são dinâmicos e formados a partir da detecção de objetos móveis.

O dispositivo de visão utilizado é o Kinect, cuja avaliação de precisão e repetibilidade é apresentada no apêndice A. No sistema de visão são utilizadas ambas imagem RGB e de profundidade. A detecção de objetos é realizada aplicando métodos de detecção de movimentos sobre a imagem RGB conforme a seção 2.3. Com base no registro entre imagem RGB e de profundidade, é possível obter dados tridimensionais destes obstáculos.

Campo potencial de repulsão

O campo potencial de forças de repulsão é formado a partir da nuvem de pontos dos objetos móveis detectados pelo sistema de visão apresentado no capítulo 2, ou por qualquer outro sistema capaz de executar a mesma tarefa. As forças de repulsão são calculadas diretamente proporcionalmente à massa e inversamente à distância ao objeto, considerando que cada ponto de aquisição possui uma determinada massa.

$$F_{rep}^d = \frac{Gm_{obj}}{d^2} \quad (4.16)$$

em que G é uma constante de valor elegível, m_{obj} é a massa do objeto, e d é a distância do objeto ao TCP do robô (KHATIB, 1986).

A força de repulsão é convertida para um deslocamento do robô em relação a trajetória inicial, sendo assim aplicada virtualmente sobre o robô na direção definida entre TCP e o ponto mais próximo a ele pertencente a superfície do objeto, e com sentido de afastamento entre TCP do robô e objeto. Ainda segundo Khatib (1986), o campo de repulsão também deve considerar a geometria do objeto ao invés de se considerar somente um único ponto de representação, tal como o seu centróide. Khatib (1986) apresenta equacionamentos para cálculo de mínimas distância para diversas classes geométricas (cubos, cilindros, etc.). As forças de repulsão também consideram o corpo do robô, não somente seu TCP.

Para calcular a distância a estas classes geométricas, primeiramente é necessário classificar o objeto. Para o caso 2D pode-se utilizar métodos já conhecidos de classificação, tal como o utilizado em Pedro (2008). Para o caso 3D propõe-se a utilização de redes neurais artificiais (RNAs). Adicionalmente, um campo potencial de repulsão também pode ser associado aos limites do espaço de trabalho do robô para evitar fins-de-curso mecânicos e pontos de singularidade geométrica.

Os campos potenciais apresentam-se mais adequados para ambientes dinâmicos pela simplicidade matemática, fato que favorece a sua execução em tempo-real. Porém, eles apenas evitam colisões, ou seja, não apresentam uma solução caso as colisão ocorra, ou seja, não tratam ou reagem a uma colisão. Na ocorrência de erros de medida, erro de modelagem do sistema, de objetos e do ambiente, posições indesejadas podem ocorrer. Cita-se Haddadin et al. (2010) como propostas de *collision handling and reaction*.

Mapa de impedância

Em movimentações livres no espaço, o robô deve se mover com baixa impedância. Porém, se houver a possibilidade de colisão com algum objeto, os parâmetros do controlador podem ser modificados com o objetivo de diminuir os efeitos da colisão. Essa modificação pode ser conseguida através de um mapa espacial de impedâncias.

Para a implementação do mapa de impedância, o espaço de trabalho é discretizado em elementos volumétricos denominados *voxels*, quem possuem dimensões configuráveis em função do refinamento desejado. Cada *voxel* representa uma região espacial com volume definido, formando uma matriz tridimensional. A figura 4.3 apresenta um conjunto de *voxels* formando um espaço volumétrico. Cada *voxel* pode contar uma informação referente ao ambiente: cor, densidade, temperatura, etc. No caso do mapa de impedância, cada *voxel* conterá um parâmetro de inércia, amortecimento e rigidez.

Figura 4.3: Exemplo de discretização de um espaço em elementos volumétricos discretos, os *voxels*.

Para a construção do mapa de impedância, inicialmente define-se quais são os *voxels* que a nuvem de pontos de um objeto ocupa. Dada uma nuvem de pontos de um objeto, como por exemplo aquelas obtidas pelo sistema de visão do capítulo 2, e dada a divisão do espaço de trabalho do robô em *voxel*, sendo que cada um deles representa uma região espacial limitada, é possível verificar em qual *voxel* cada ponto está localizado e assim definir o conjunto de *voxels* que uma nuvem de pontos ocupa. Os *voxels* ocupados por pontos recebem o valor 1, e os demais o valor 0.

De forma similar ao processamento de imagens por operadores morfológicos (vide seção 2.3.3), é possível realizar operações morfológicas sobre a matriz de *voxels* utilizando elementos estruturantes de convolução (ROSSL et al., 2000). A figura 4.4 apresenta um exemplo de elemento estruturante volumétrico. Uma operação de dilatação pode ser conseguida adicionando informação à matriz de *voxels* quando durante a convolução o elemento estruturante encontrar um valor de interesse, neste caso, um ponto pertencente ao objeto cujo valor de *voxel* é igual a 1.

Com o uso de operadores morfológicos 3D de dilatação é possível adicionar parâmetros de impedância ao redor do objeto. Assim, quando o robô se aproxima do objeto, os parâmetros massa e mola do controle de impedância devem ser diminuídos e o parâmetro amortecimento deve ser aumentado. A diminuição da massa diminui efeitos de colisão. A diminuição da rigidez permite que o robô responda com deslocamentos consideráveis quando pequenas forças atuam em seu TCP, evitando assim grandes forças de interação na colisão. O aumento do

Figura 4.4: Exemplo de elemento estruturante 3D.

amortecimento é necessário para evitar que o robô atinja elevadas velocidade.

A figura 4.5 apresenta a nuvem de pontos 3D, a figura 4.6 apresenta a vista superior da nuvem de pontos, e a figura 4.7 apresenta em tons de cinza a modificação de parâmetros de amortecimento ao redor do objeto. Por problemas de representação gráfica o mapa de impedância é representado em uma projeção 2D.

Figura 4.5: Nuvem de pontos de um objeto.

Figura 4.6: Vista superior da nuvem de pontos da figura 4.5.

Figura 4.7: Projeção 2D do mapa de impedância.

5 *Experimentos de manipulação*

5.1 Introdução

Uma série de experimentos foi planejada e conduzida para avaliar as propostas deste trabalho, cujo foco estão concentrados nos seguintes desafios da manipulação: estabilidade no estabelecimento do contato; estabilidade e desempenho em diferentes fases da manipulação; integração e funcionalidade sem troca de controladores; tratamento e colisões desejadas e indesejadas; integração do sistema de controle da manipulação com sistema de visão e tratamento de erros de aquisição inerente aos sistema de visão.

Um total de três experimentos foram planejados:

Experimento 1: Estabilidade do contato;

Experimento 2: Estabilidade e desempenho nas diferentes fases da manipulação;

Experimento 3: Integração de manipulação e visão.

Os detalhes de cada experimento e seus respectivos resultados são detalhados nas próximas seções.

5.2 Experimento de estabilidade de contato

A estabilidade no estabelecimento e perda de contato entre robô e ambiente é um dos desafios da manipulação robótica. Ela depende de diversos fatores, desde a lei de controle e detalhes de sua implementação, até a precisão de medidas do meio ambiente.

O controle de impedância foi selecionado para a implementação do sistema de manipulação deste trabalho devido à sua característica de transformar distúrbios de força, proveniente da mudança de fase ou de erros de medidas do ambiente, em resposta dinâmica regulável. Contudo, detalhes construtivos do manipulador e as técnicas de sua implementação influenciam a sua estabilidade (LAWRENCE, 1988).

O robô disponibilizado para os experimentos, originalmente específicos para aplicações industriais, possui elevadas reduções em seus atuadores e conseqüentemente apresenta elevada impedância passiva, e por essa característica, torna-se obrigatório o uso de retroalimentação de força através de um sensor intrínseco posicionado no TCP do robô.

Quando se utiliza um manipulador de elevada impedância passiva e realimentação de força existem diversas técnicas de implementação: com controle interno de força (EPPINGER; SEERING, 1992); com controle interno de posição (PELLETIER; DOYON, 1994); com controle interno de velocidade (PRATT et al., 2004); e por torque calculado a partir de modelo matemático (HOGAN, 1985). Destas técnicas, somente a última não foi implementada devido às dificuldades de modelagem experimental do robô e de determinação dos atritos nas reduções dos atuadores.

A implementação com controle interno de força é teoricamente instável quando estabelece contato com objetos de elevada rigidez. Algumas estratégias como utilização de elementos flexíveis na interface de contato e uso de compensadores de atraso e avanço de fase são propostas para suavizar este problema, mas não o resolvem completamente.

As implementações com realimentação de velocidade e posição apresentam estabilidade teórica e ainda apresentam a vantagem de não exigirem o uso de modelos matemáticos. Foi utilizada a implementação com realimentação interna de posição (PELLETIER; DOYON, 1994).

O termo utilizado "estabilidade teórica" refere-se ao fato de que o sistema poderá apresentar resposta estável dentro de critério estabelecidos desde que os seus parâmetros de impedância (inércia, rigidez e amortecimento) sejam ajustados adequadamente para que a resposta ao contrato esteja dentro de critérios e requisitos estabelecidos. A ajuste dos parâmetros de impedância devem levar em consideração características do ambiente uma vez que a resposta no estabelecimento do contato é função do acoplamento entre robô e ambiente.

Experimentos de estabelecimento de contato foram conduzidas para diversos valores de impedância. A resposta do robô ao contato (perfil de posição e forças no tempo) foi utilizada para avaliar o desempenho do controlador. Como o fenômeno do contato pode ser modelado como um sistema de segunda ordem resultante do acoplamento entre robô e objeto, a análise foi realizada segundo parâmetros de análise de estabilidade de resposta de sistemas de segunda ordem: amortecimento (crítico, super ou subamortecido) tempo de resposta, tempo de estabilização e sobressinal.

No experimento, o robô foi movido em linha reta na direção x do sistema de referência fixo à sua base em direção ao centro do pote (objeto selecionado para o experimento) de forma a provocar estabelecimento do contato ortogonal a sua superfície, assim as reações de contato são restritas a apenas uma direção, fato que facilita a análise.

O robô foi programado para executar a trajetória de forma a estabelecer o contato com velocidade constante e aceleração nula. A resposta de cada experimento é representada em termos da posição e da força ao longo do tempo. Para a força é apresentada a resultante da aplicação de um filtro passa-baixa, que é o sinal utilizado pelo sistema de controle de impedância.

Nos experimentos, cada um dos termos de M , K e B foram variados separadamente com a finalidade de identificar e avaliar a influência de cada parâmetro na estabilidade do contato.

O primeiro experimento de contato foi realizado com baixa impedância. Considerando que a impedância do robô é consideravelmente menor que aquela do objeto, garante-se assim que o robô apresentará respostas mensuráveis e passíveis de análise segundo a uma abordagem de análise de resposta dinâmica transiente.

A figura 5.1 apresenta os resultados para $M = 5\text{kgm}^2$, $K = 6\text{N/m}$ e $B = 100\text{Ns/m}$. A partir dos resultados é possível verificar a instabilidade do estabelecimento do contato. No experimento o robô estabelece e perde contato de forma oscilatória. O detalhe 1 mostra o instante em que o robô perde contato com o objeto após o primeiro estabelecimento. Pode ser observado que o comportamento se repete de forma cíclica.

Figura 5.1: Resultado do experimento 1 com $M = 5\text{kgm}^2$, $K = 6\text{N/m}$ e $B = 100\text{Ns/m}$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Investigação do parâmetro de amortecimento B

Nos experimentos de 2 a 6 os parâmetros M e K foram mantidos constantes, enquanto os valores de B foram progressivamente aumentados para se verificar a influência do amortecimento na resposta do estabelecimento do contato. Os valores de amortecimento do experimento 2 ao 6 foram respectivamente: $500Ns/m$; $750Ns/m$; $1000Ns/m$; $2000Ns/m$; e $3000Ns/m$. É possível verificar que o contato torna-se estável com o aumento de B . Para $B = 500Ns/m$ (figura 5.2) não há perda de contato e diminuição da amplitude de oscilação. A partir da resposta de força de contato (gráficos à direita), é possível observar que o aumento do amortecimento provoca aumento do decaimento das oscilações da resposta transitória.

Figura 5.2: Resultado do experimento 2 com $M = 5kgm^2$, $K = 6N/m$ e $B = 500Ns/m$. À esquerda a posição do TCP e à direita a força de interação do robô com o objeto.

Figura 5.3: Resultado do experimento 3 com $M = 5kgm^2$, $K = 6N/m$ e $B = 750Ns/m$. À esquerda a posição do TCP e à direita a força de interação do robô com o objeto.

Figura 5.4: Resultado do experimento 4 com $M = 5kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.5: Resultado do experimento 5 com $M = 5kgm^2$, $K = 6N/m$ e $B = 2000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.6: Resultado do experimento 6 com $M = 5kgm^2$, $K = 6N/m$ e $B = 3000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Investigação do parâmetro de rigidez K

A influência do parâmetro de rigidez K também foi investigada. A análise foi realizada aumentando-se este parâmetro a partir daqueles do experimento 4, que, lembrando, são $M = 5\text{kgm}^2$, $K = 6\text{N/m}$ e $B = 1000\text{Ns/m}$.

Para os experimentos de 7 a 10 os novos valores de rigidez foram, respectivamente: 100N/m ; 200N/m ; 400N/m e 1000N/m . Modificações na rigidez alteram, principalmente, a frequência natural do sistema, aumentando-a. Pode-se verificar, principalmente na resposta de força de contato da figura 5.10 que o aumento da magnitude da força é diretamente proporcional ao aumento de K .

Figura 5.7: Resultado do experimento 7 com $M = 5\text{kgm}^2$, $K = 100\text{N/m}$ e $B = 1000\text{Ns/m}$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.8: Resultado do experimento 8 com $M = 5\text{kgm}^2$, $K = 200\text{N/m}$ e $B = 1000\text{Ns/m}$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.9: Resultado do experimento 9 com $M = 5kgm^2$, $K = 400N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.10: Resultado do experimento 10 com $M = 5kgm^2$, $K = 1000N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Investigação do parâmetro de inércia M

Por fim, a influência da inércia no estabelecimento do contato foi observado. Os valores de inércia também foram modificados a partir da impedância do experimento 4, que, lembrando, são $M = 5kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$.

Figura 5.11: Resultado do experimento 11 com $M = 50kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.12: Resultado do experimento 12 com $M = 100kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.13: Resultado do experimento 13 com $M = 200kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

Figura 5.14: Resultado do experimento 14 com $M = 400kgm^2$, $K = 6N/m$ e $B = 1000Ns/m$. À esquerda a posição do TCP e à esquerda a força de interação do robô com o objeto.

As figuras de 5.11 a 5.14 apresentam as respostas para $M = 50kgm^2$, $M = 100kgm^2$, $M = 200kgm^2$ e $M = 400kgm^2$. É possível verificar a diminuição da frequência das oscilações e a diminuição da taxa de decaimento decaimento com o aumento da rigidez da impedância do robô.

5.3 Experimento de manipulação: estabilidade e desempenho em diferentes fases

O segundo experimento consiste na execução, em ambiente experimental, da tarefa de desrosqueamento da tampa de um pote, segundo uma trajetória similar àquela apresentada na seção 3.4. O objetivo deste experimento é avaliar o desempenho da combinação de trajetórias cinemáticas suaves com o perfil de mudança dos parâmetros de impedância durante a manipulação.

A trajetória de manipulação foi dividida em duas etapas para a sua geração: manipulação e *regrasping*. A manipulação é a fase em que o robô está em contato com o objeto, e o *regrasping* é a fase entre estabelecimento e perda de contato quando o robô se move livre no espaço.

A trajetória cinemática (posição no tempo e suas derivadas) para as duas fases foi gerada utilizando o planejador *Squeezed Screw*. A trajetória de *regrasping* foi gerada de forma a circunavegar o objeto sem descontinuidades no espaço e na velocidade. Para a manipulação, foi gerada uma trajetória de forma que o TCP do robô pudesse tangenciar a superfície circular da tampa do pote. A transição entre as duas trajetórias é suave sem descontinuidades para favorecer a estabilidade da transição de fase.

O controle de impedância é utilizado em todas as fases da manipulação. Contudo, para que o controle tenha comportamento diferente em cada uma das fases, seus parâmetros devem ser devidamente ajustados. Na movimentação livre os parâmetros de impedância são mantidos baixos ($M = 1\text{kgm}^2$, $K = 1\text{N/m}$ e $B = 100\text{Ns/m}$) para que a trajetória seja devidamente seguida¹. No estabelecimento do contato e durante a interação com o objeto, os parâmetros de impedância são: $M = 10\text{kgm}^2$; $K = 20\text{N/m}$; e $B = 2000\text{Ns/m}$, valores próximos àqueles que apresentaram menores oscilações após o estabelecimento de contato nos experimentos da seção 5.2. Neste Experimento o perfil de impedância é estabelecido em função da trajetória cinemática.

Para que a manipulação seja realizada, forças devem ser aplicadas sobre o objeto. Como a manipulação é focada no objeto (OKAMURA et al., 2000), dada uma movimentação desejada e as características dinâmicas deste como massa e momentos de inércia, as forças podem ser calculadas através de equações dinâmicas. Uma vez que o controle de impedância requer somente uma trajetória desejada e não forças, uma estratégia para se conseguir as forças é provocar um deslocamento da trajetória original, que poderia ser calculado a partir das forças e da impedância (PRATT et al., 2004). Contudo, a impedância a ser considerada não é aquela da impedância do robô, mas sim aquela equivalente do acoplamento dinâmico do robô com objeto. Uma vez que conhecer as propriedades dinâmicas do objeto não é uma tarefa trivial na prática,

¹Na próxima seção é discutido o papel da impedância na ocorrência de colisão durante a movimentação livre.

adota-se um deslocamento arbitrário, e desta forma espera-se que as forças de manipulação sejam geradas e reguladas intrinsecamente.

É importante ressaltar para contextualizar este experimento com um dos focos do trabalho que não foi considerado nenhum modelo de contato. O controle de impedância recebe realimentação de posição e de força medida por um sensor de força em seu TCP. Os fenômenos de contato como rolagem, escorregamento e deformação são intencionalmente negligenciados para avaliar a dependência do desempenho da manipulação em função da consideração dos fenômenos do contato.

Experimento de manipulação com mudança repentina de impedância

Inicialmente um experimento foi realizado com mudança repentina da impedância dentre aquela de movimentação livre para aquela de manipulação. No experimento foram realizadas as seguintes medidas: posição do robô e forças extrínsecas. A figura 5.16 apresenta os resultados obtidos. A figura 5.16(a) apresenta no espaço cartesiano a trajetória desejada em linha contínua e a trajetória executada em linha tracejada, o círculo externo representa a tampa do objeto e o interno representa o descolamento da trajetória para geração intrínseca das forças de manipulação. A figura 5.16(b) apresenta as forças normais e tangenciais durante o contato, ambas projetadas no espaço cartesiano para cada posição do contato. As figuras 5.16(c) e 5.16(e) apresentam, respectivamente, o perfil de força normal e tangencial no tempo. As figuras 5.16(d) e 5.16(f) apresentam, respectivamente a posição X e Y desejada (linha tracejada) e executada (linha contínua). No experimento não houve erro de posicionamento relativo entre objeto e robô, e a reorientação final desejada para a tampa após a manipulação é de 90° .

(a)

(b)

(c)

(d)

(e)

(f)

Figura 5.15: Experimento de manipulação com mudança repentina de impedância entre as fases.

Neste experimento não houve instabilidade no estabelecimento do contato, pois a movimentação livre foi iniciada sem erros de posicionamento do TCP do robô em relação à trajetória desejada sendo que este erro manteve-se pequeno até o momento do contato. Uma vez que não houve erros de posicionamento, a mudança de impedância aconteceu em sincronismo com a mudança de fase da manipulação.

Durante a manipulação, o erro entre trajetória desejada e executada foi maior que aquele previsto para a geração das forças de manipulação, desta forma, a mudança de impedância aconteceu antes da mudança de fase, o que provocou falha na manipulação. Mesmo com erro de posicionamento elevado, a baixa impedância não foi capaz de gerar forças de manipulação, conforme mostram os detalhes 2 e 3. No intervalo de 18s a 24s (aproximadamente) o robô manteve-se em contato com o objeto, porém, sem aplicar forças de manipulação e sem se movimentar, conforme os detalhes 5 e 6 mostram a posição constante neste intervalo. Após este intervalo de tempo, o avanço da trajetória desejada permitiu que o robô executa-se a movimentação livre. Após a manipulação, o giro total da tampa foi de $73,5^\circ$, ou seja, 81,67% do giro desejado.

Experimento de manipulação com mudança suave de impedância

Neste experimento de manipulação a movimentação livre foi realizada com a mesma com $M = 1\text{kgm}^2$, $K = 1\text{N/m}$ e $B = 100\text{Ns/m}$ de impedância, e com $M = 10\text{kgm}^2$; $K = 20\text{N/m}$; e $B = 2000\text{Ns/m}$ de impedância na manipulação, ou seja, as mesmas do experimento anterior, porém, neste caso, a mudança entre as impedâncias acontece de forma suave segundo polinômios do terceira ordem em um tempo total de 100ms.

Os gráficos da figura 5.16 apresentam as respostas deste experimento e seguem a mesma organização dos gráficos da figura 5.15. Neles, é possível observar que tanto o estabelecimento quanto a perda de contato foram estáveis, sendo que não é observado os problemas de perda de contato do experimento anterior. O giro total da tampa foi de $80,5^\circ$ (89,44%).

Figura 5.16: Experimento de manipulação com mudança suave de impedância entre as fases.

Experimento de manipulação com erro de posicionamento

Nos experimentos anteriores o pote foi posicionado sem erros dentro do espaço de trabalho do robô. Porém, considerando as limitações dos sistemas de visão na determinação da posição de objetos (conforme discutido nos capítulos 1 e 2), erros de posicionamento sempre existiram. Com o objetivo de avaliar o desempenho do sistema de manipulação na existência de erros de posicionamento, experimentos de manipulação foram realizados considerando erros de posicionamento conhecidos.

Nestes experimentos foram mantidas as mesmas impedâncias de movimentação livre e de manipulação dos anteriores com transição suave. O pote foi posicionado com erros crescentes de $5mm$ nas direções x e y do espaço de trabalho do robô.

Os gráficos da figura 5.17 apresentam os resultados do experimento de manipulação com erro de posicionamento do pote de $+5mm$ em x . É possível verificar o deslocamento da trajetória executada e das forças, na representação cartesiana, em comparação ao primeiro experimento (figura 5.16). Neste experimento o giro final da tampa do pote foi de $69,5^\circ$ (72,22% do total).

Figura 5.17: Experimento de manipulação com erro de posicionamento entre objeto e robô de +5mm na direção x .

Repetição do primeiro experimento com erro de posicionamento do pote em $+5mm$ em x e $+5mm$ em y . Neste experimento o giro final da tampa do pote foi de 55° (61,11% do total).

Figura 5.18: Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e na direção y .

Repetição do primeiro experimento com erro de posicionamento do pote em $+5mm$ em x e $+10mm$ em y . Neste experimento o giro final da tampa do pote foi de 41° (46,11% do total).

Figura 5.19: Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e de $+10mm$ na direção y .

Repetição do primeiro experimento com erro de posicionamento do pote em $-5mm$ em x e $-5mm$ em y . Neste experimento, os detalhes 6 e 7 mostram a colisão do robô com o objeto antes do previsto, quando a impedância ainda era aquela para movimentação livre. Devido à esta colisão antes do momento esperado, a manipulação falha, sendo que o giro final da tampa foi nulo, e ainda, as forças de interação entre robô e objeto resultaram em uma translação do pote sobre a mesa.

Figura 5.20: Experimento de manipulação com erro de posicionamento entre objeto e robô de $+5mm$ na direção x e de $+10mm$ na direção y .

Os resultados obtidos no experimento anterior mostram que a manipulação pode ser executada com desempenho satisfatório (giro da tampa entre 10% e 100% do total desejado) mesmo na presença de erros de posicionamento do objeto em relação à trajetória de referência.

Pode-se concluir que um dos motivos para que a manipulação seja satisfatória, ou parcialmente satisfatória, é o estabelecimento e perda do contato com impedância adequada. Quando o erro de detecção de posição do objeto é elevado, a mudança de fase ocorre em instantes diferentes daqueles previstos e a manipulação torna-se insatisfatória.

5.4 Experimento de integração de manipulação e visão

Em um ambiente não estruturado, os objetos a serem manipulados devem ser detectados e identificados. Neste sentido, pode-se citar pelo menos dois mecanismos para a detecção: por visão e por tato. Dentre diversas características, os sistema de visão fornecem maior volume de dados, porém com precisão menor que por meio do tato. Neste trabalho, interessa-se somente pela detecção pelo sistema de visão. Neste caso, existem erros de medida que vão desde erros de precisão e repetibilidade do dispositivo de aquisição, até erros de calibração e de transformação homogênea entre os dois sistemas.

A proposta de integração do sistema de manipulação com o sistema de visão é, em primeiro lugar, modificar a impedância com base em um mapa de impedância construído segundo a descrição da seção 4.3.2. Desta forma, a modificação da impedância ocorre em função do nuvem de pontos do objeto e não da trajetória cinemática. Assim, erros de seguimento de trajetória, podem não interferir, ou provocar uma interferência menor no sucesso da manipulação. Com esta abordagem, os mapas de impedância, inicialmente propostos para tratar colisões indesejadas, podem também serem utilizados para a manipulação. Adicionalmente, neste experimento interessa-se em verificar como o planejador de manipulação integrados com o sistema de visão se comporta na presença de erros de medida já investigados no capítulo 2.

A segunda proposta de integração com sistema de visão consiste na detecção da posição do objeto no espaço de trabalho para posicionamento da trajetória cinemática. Para tanto, pretende-se utilizar o método *Iterative Closest Point* (ICP) (KJER; WILM, 2010), que é capaz de ajustar automaticamente duas nuvens de pontos similares. Utilizando uma nuvem modelo do objeto cuja posição é conhecida é possível encontrar a posição da nuvem de pontos adquiridos, uma vez que o ICP retorna a matriz de transformação homogênea entre as duas nuvens de pontos (modelo e adquirida). Após a detecção, a trajetória de manipulação é corrigida para a atual posição do objeto utilizando a matriz de transformação resultante do ICP. Vale lembrar que o

erro de posicionamento da trajetória é acumulativo: erro de repetibilidade e precisão do Kinect; erro de calibração entre Kinect e robô; e erro de ajuste do ICP.

Experimentos de integração da manipulação com o sistema de visão foram planejados e conduzidos. O experimento foi repetido com o pote em 5 diferentes posições dentro do espaço de trabalho. As demais condições de contorno, como iluminação por exemplo, foram mantidas constantes e iguais àsquelas do experimento da seção anterior. A partir da nuvem de pontos detectada pelo sistema de visão, o mapa de impedância para a manipulação foi criado e a trajetória posicionada utilizando a matriz de transformação obtida com o ajuste do ICP. As figuras de 5.21 a 5.30 apresentam os resultados dos experimentos.

Os resultados estão organizados na seguinte forma, tomando-se como exemplo as figuras 5.21 e 5.22. A figura 5.21 apresenta os resultados do sistema de visão: a identificação da nuvem de pontos da tampa (figura 5.21(a)), o resultado do ajuste da nuvem de pontos identificada com uma nuvem de pontos de referência (figura 5.21(b)) pelo método ICP, a curva de erro de ajuste para cada iteração do ICP (figura 5.21(c)), a posição da nuvem de pontos do objeto e da trajetória posicionada a partir da identificação da posição do objeto, ambos no espaço de trabalho do robô (figura 5.21(d)), a vista superior da nuvem de pontos do objeto (figura 5.21(e)), e o mapa de impedância construído (figura 5.21(f)).

Os resultados da figura 5.22 apresentam a mesma organização dos resultados dos experimentos da seção anterior.

Figura 5.21: Resultados do processamento de imagens. Primeira repetição do experimento de integração de manipulação com sistema de visão.

Resultados da manipulação. O giro total da tampa foi de $85,5^\circ$ (95% do giro total).

Figura 5.22: Resultados da manipulação. Primeira repetição do experimento de integração de manipulação com sistema de visão.

Figura 5.23: Resultados do processamento de imagens. Segunda repetição do experimento de integração de manipulação com sistema de visão.

Resultados da manipulação. O giro total da tampa foi de $65,5^\circ$ (72,78% do giro total).

Figura 5.24: Resultados da manipulação. Segunda repetição do experimento de integração de manipulação com sistema de visão.

Figura 5.25: Resultados do processamento de imagens. Terceira repetição do experimento de integração de manipulação com sistema de visão.

Resultados da manipulação. O giro total da tampa foi de $87,0^\circ$ (96,67% do giro total).

Figura 5.26: Resultados da manipulação. Terceira repetição do experimento de integração de manipulação com sistema de visão.

Figura 5.27: Resultados do processamento de imagens. Quarta repetição do experimento de integração de manipulação com sistema de visão.

Resultados da manipulação. O giro total da tampa foi de $83,0^\circ$ (92,22% do giro total).

Figura 5.28: Resultados da manipulação. Quarta repetição do experimento de integração de manipulação com sistema de visão.

Figura 5.29: Resultados do processamento de imagens. Quinta repetição do experimento de integração de manipulação com sistema de visão.

Resultados da manipulação. O giro total da tampa foi de $41,0^\circ$ (44,44% do giro total).

Figura 5.30: Resultados da manipulação. Quinta repetição do experimento de integração de manipulação com sistema de visão.

Nesta última seção de experimentos o sistema de manipulação proposto foi avaliado na presença de erros de determinação da posição do objeto no espaço de trabalho do robô, provenientes tanto dos erros do sistema de visão quanto dos erros do método de ajuste de malhas ICP utilizado para determinar a posição do objeto. Nos experimentos também foi avaliado o desempenho da manipulação quando a inércia do robô tem seus parâmetros atualizados em função da posição cartesiana do TCP do robô segundo um mapa de impedância.

A partir dos resultados dos experimentos pode-se verificar que o mapa de impedância é mais eficiente para a manipulação na ocorrência de erros de determinação de posição do objeto que a modificação dos parâmetros de impedância em função da trajetória cinemática desejada. A eficiência foi comprovada pois para maiores erros de posicionamento, como por exemplo os resultados mostrados na figura 5.30, em que o erro total de posicionamento chegou a 15mm.

Quanto à utilização do mapa de impedância para o tratamento de colisões, partindo-se do princípio de que o estabelecimento do contato na manipulação é uma colisão (porém desejada), considera-se que o mapa de impedância é eficiente no tratamento de colisões a partir de dados de um sistema de visão com erros de precisão e de repetibilidade de aquisição (apresentados no apêndice A) e com métodos de processamento de sinais simplificados (apresentados no capítulo 2), pois o mapa de impedância é capaz de indicar novos parâmetros de impedância para o robô a em função de sua posição relativa ao objeto, que então podem ser utilizados para se reduzir os efeitos indesejados da colisão.

Por fim, os resultados obtidos nestes experimentos mostram que a medida que o objeto de manipulação está mais distante do sensor de visão e da região de aquisição dos pontos de calibração (vide seção 2.2), o desempenho da manipulação (neste caso a porcentagem de giro da tampa do pote em relação ao total desejado) diminui, chegando a uma região limite que a partir da qual a manipulação não apresenta desempenho satisfatório.

Esta última conclusão se deve, principalmente, a dois motivos: erro de precisão e repetibilidade de medida do sensor Kinect; e erro de calibração entre o robô SCARA e o Kinect. Dependendo da região de aquisição dos pontos de calibração, a região no espaço de trabalho do robô de melhor desempenho pode variar. Para os experimentos, a aquisição dos pontos de calibração foi feita na região de menor erro de repetibilidade do Kinect, ou seja $P^K = [0 \ 0 \ 40]^T mm$, assim, a região preferencial de manipulação é aquela mostrada na figura 5.31.

Figura 5.31: Região do espaço de trabalho do robô preferencial para a manipulação.

6 *Conclusões, discussões e considerações finais*

A manipulação robótica em ambientes não estruturados envolve diversos desafios que atualmente são investigados por pesquisadores da área. Diferente dos ambientes industriais e laboratoriais, os ambientes não estruturados apresentam como características principais: as constantes e rápidas modificações em seu ambiente; diversidade de objetos; e o mais importante, a presença de humanos, que dependendo da aplicação, podem ser leigos em robótica ou em sistemas autônomos ou automatizados.

Os ambientes não estruturados são, na grande maioria das vezes, encontrados em aplicações que, diferente das industriais que permitem investimento em equipamentos caros, requerem baixo investimento. O custo de dispositivos robóticos está diretamente relacionado com os sensores utilizados, com o grau de autonomia e com a complexidade da implementação de funcionalidades.

Devido a essas características, os requisitos para o desenvolvimento destes sistemas de manipulação são diferentes daqueles para aplicações em ambientes estruturados e semi-estruturados e em outras aplicações da robótica. Neste trabalho o foco foi estudar e propor um sistema capaz de realizar tarefas de manipulação sem troca de controladores e com funcionalidades inerentes ao tipo de ambiente: desvio e tratamento de colisão com baixa complexidade de implementação.

Comparando a manipulação robótica em ambientes não-estruturados com as demais aplicações em robótica, pode-se citar duas diferenças. A primeira é a interação com o ambiente por meio do fenômeno de contato, e a segunda é a necessidade de desvio de colisões. Neste trabalho defende-se que as colisões não precisam ser obrigatoriamente evitadas, mas sim tratadas pois a sua ocorrência pode não comprometer o sucesso da tarefa. E ainda, algumas colisões podem ser utilizadas como apoio a uma tarefa de manipulação.

Neste sentido, duas questões foram levantadas e guiaram o desenvolvimento do trabalho:

É necessário modelar e controlar precisamente o contato para realizar tarefas de manipulação?

É possível desenvolver um sistema simplificado capaz de realizar tarefas de manipulação?

Na tentativa de responder estas questões, o objetivo do trabalho foi estudar propor um sistema com as seguintes características:

1. Capaz de realizar tarefas de manipulação sem a necessidade de modelagem e controle do contato;
2. Capaz de estabelecer e perder contato de forma estável;
3. Capaz de prevenir, e no caso de ocorrência, tratar colisões indesejadas de forma adequada;
4. Ser robusto a erros de medida e ruídos dos sensores de visão.

Para o desenvolvimento do sistema, foi adotada a divisão da manipulação em três níveis, segundo aquela proposta por Okamura et al. (2000): níveis primário, secundário e terciário. O primário apresenta somente um único tipo de controlador, o Controle de Impedância, que é apontado pelos pesquisadores da área como o mais adequado para tarefas de contato. Uma outra vantagem deste controlador é a possibilidade de modificar sua impedância para que o robô apresente um comportamento dinâmico desejado quando em contato com objetos. Assim, é possível executar diferentes tarefas sem a necessidade de troca de controladores, uma prática comum na área, que apresenta desvantagens: custo computacional de sistemas de visão ou de sistemas de tato para que a colisão seja prevista ou detectada; e instabilidades devido à própria troca de lei de controle (CUTKOSKY; HYDE, 1993).

A mudança da lei de controle pode provocar instabilidade pois é necessário assegurar que ela ocorra no mesmo instante em que o estado do robô é modificado. Isso requer o uso de sistemas de sensoramento de alto desempenho para previsão ou rápida detecção de eventos. As desvantagens de se usar sistemas de previsão por meio de visão e detecção por meio de tato, de forma resumida, são o elevado custo dos dispositivos, custo computacional e complexidade dos métodos e algoritmos utilizados, principalmente para os sistemas de visão.

O nível secundário é composto pela integração de funcionalidades, que neste trabalho são: o desvio da colisão; e um mapa de modificação de impedância para o tratamento da colisão e para a manipulação. Essas funcionalidades são realizadas por meio de realimentação de dados obtidos por um sistema de visão. Optou-se pelo uso de um sistema estático e de métodos de

processamento de imagens simplificado para prova de conceito, uma vez que o desenvolvimento de um sistema de visão para manipulação não é o foco do trabalho e está sendo realizado em trabalhos dentro do Grupo de Mecatrônica da EESC contemporâneos a este.

O nível terciário, o de maior inteligência, é responsável pela autonomia do sistema, é responsável pela geração das trajetórias de manipulação e pelo sincronismo entre os dedos e braços robóticos. Este nível também não foi foco de estudos deste trabalho, tal que a trajetória para execução da tarefa de manipulação eleita para os testes foi gerada empiricamente.

Uma trajetória previamente definida é exigida para utilização do controle de impedância, uma vez que ele requer uma como referência. Neste sentido, foi proposto um planejador de trajetória complementar ao planejador de rotas *Squeezed Screw* proposto por Stücheli (2009). O planejador de rotas foi proposto como alternativa aos planejadores para desvio de obstáculos, tal como o RRT. O novo planejador consiste em uma abordagem alternativa que considera que colisões não devem ser obrigatoriamente evitadas, mas sim tratadas adequadamente caso ocorram para que a tarefa não seja interrompida tratar colisões pode permitir que o robô continue a execução da tarefa após a ocorrência da colisão. Em ambientes não-estruturados evitar colisões pode ser uma funcionalidade altamente custosa. Evitar colisões exige monitoramento constante e em elevadas velocidades de todo o ambiente, e mesmo que o monitoramento seja escutado satisfatoriamente, novas trajetórias devem ser calculadas a todo instante.

O planejador de rotas foi complementado com uma proposta de perfil de velocidades, formando assim um planejador de trajetórias suaves sem descontinuidades tanto no espaço quanto na velocidade. A trajetória para a execução da tarefa de manipulação eleita para os testes foi gerada usando subtrajetórias *Squeezed Screw*. Os resultados apresentados na seção 3.4 mostram que não há descontinuidades. Comparações com o RRT, planejador de rotas livres de colisão, mostram que não há picos e nem valores elevados tanto para a velocidade e para a aceleração, assim, problemas de colisão quando o robô apresenta altas velocidades podem ser amenizados.

Além de velocidades e acelerações menores, para que colisões ocorram sem que o robô apresente instabilidades é necessário que seus parâmetros de inércia, rigidez e amortecimento sejam adequados. Um experimento de colisão do robô com um objeto foi elaborado e conduzido com o objetivo de investigar a influência de cada parâmetro no evento. Os resultados são apresentados na seção 5.2, e a seguir seguem interpretações e considerações dos resultados.

Na colisão, o controle de impedância proporciona ao robô um comportamento dinâmico que relaciona o desvio de posição da trajetória desejada (e suas derivadas) com a força externa aplicada no mesmo. Os parâmetros podem ser ajustados em tempo de execução uma vez que o comportamento é obtido através da implementação em *software*. Apesar de não existirem

componentes reais de inércia, rigidez e amortecimento, o robô reage fisicamente ao contato e ao acoplamento dinâmico com o objeto como se estes componentes existissem.

No estabelecimento do contato, a velocidade relativa entre robô e objeto e a inércia virtual exercem influência direta: quanto maior a energia cinética maior é o impulso e maior é a energia transferida do robô para o objeto. De forma geral, quanto maior a velocidade do impacto, menor deverá ser a inércia e maior deverá ser o amortecimento da impedância virtual. Uma inércia reduzida diminui a energia do impacto e o amortecimento evita oscilações posteriores à colisão.

Outro fator que deve ser levado em consideração para análise da resposta do robô e do objeto à colisão é o período de ciclo de realimentação. Para manipuladores com alta impedância passiva, um comportamento dinâmico de baixa impedância só pode ser obtido com o uso de realimentação de força (além do de posição ou velocidade), assim, existe uma latência entre o instante da colisão e a resposta do sistema a partir da realimentação. Durante este intervalo, a impedância virtual não tem efeito sobre a colisão, sendo que a efetiva é a impedância passiva do robô.

Conclui-se que um sistema com um período longo de realimentação demora a responder a um impacto e apresenta instabilidade para elevadas velocidades mesmo com inércia baixa e amortecimento elevado. Consequentemente, a diminuição do período de ciclo do controle pode permitir contatos mais estáveis com maiores velocidades na colisão.

Após a ocorrência da colisão, o sistema dinâmico formado pelo acoplamento entre robô e objeto apresenta resposta dinâmica de segunda ordem.

Analisando as respostas do contato nos experimentos da seção 5.2 é possível observar os efeitos da mudança dos parâmetros de impedância. Em primeiro lugar, a resposta de segunda ordem no contato não é função somente da impedância virtual do robô, mas também de parâmetros do objeto. O coeficiente de amortecimento sofre maior influência do amortecimento da impedância e do objeto, porém, para objetos de elevada inércia e rigidez elevados, o amortecimento da impedância deve ser elevado para diminuir oscilações.

A inércia da impedância I_{imp} pode ser elevada para diminuir a frequência natural no caso de objetos de elevada rigidez K_{ob} . Contudo, o aumento de I_{imp} aumenta os efeitos indesejados no momento da colisão, principalmente para os casos em que o objeto apresenta elevada rigidez e baixa inércia. Para estes casos faz-se necessário o uso de recobrimento do TCP com material de baixa rigidez para diminuir a aquela aparente do objeto, o que permite, consequentemente, diminuir I_{imp} .

Os parâmetros de impedância que apresentaram a melhor resposta no estabelecimento do contato foram selecionados para a execução da tarefa de manipulação eleita. Para a movimentação livre os parâmetros de inércia e rigidez foram mantidos baixos enquanto o amortecimento foi mantido em um valor intermediário. A transição entre os parâmetros foi feita de forma suave através de polinômios de terceiro grau. Um experimento foi planejado para avaliar o desempenho da combinação das trajetórias cinemáticas suaves com mudanças, também suaves, dos parâmetros de impedância entre as diferentes fases. Ressalta-se que nenhum modelo de contato foi considerado, e que para a geração de forças de manipulação, um *offset* foi aplicado sobre a trajetória original.

Os resultados deste segundo experimento, apresentados na seção 5.3, mostram que foi possível realizar a tarefa de manipulação sem um modelo ou controle do contato, e que as forças de manipulação foram geradas pelo *offset* adotado e reguladas intrinsecamente pelo controle de impedância. A modificação suave da trajetória e da impedância também permitiu estabelecimento de contato de forma estável. Contudo, instabilidades, que não levaram à falha da tarefa, foram verificadas quando a impedância foi diminuída antes da previsão de perda de contato. Essa instabilidade ocorreu porque a mudança de impedância foi gerada em função da trajetória, e na prática, o erro de seguimento da trajetória foi maior que aquele previsto com o *offset*. Instabilidades também foram verificadas quando há estabelecimento do contato antes do previsto. Conclui-se que erros de posicionamento da trajetória cinemática em relação ao objeto não levam a erros e falhas da tarefa, porém, as mudanças de impedância em posições não adequadas geram grandes instabilidades no estabelecimento e na perda do contato.

A modificação da impedância não deve ocorrer em sincronismo com trajetória, mas sim em função da posição relativa ao objeto. Na prática, os objetos podem estar posicionados em qualquer local dentro do espaço de trabalho do robô, e assim, a trajetória de manipulação deve ser devidamente posicionada sobre o objeto, sendo que a impedância pode ser modificada em função de um mapa de impedância construído a partir da nuvem de pontos do objeto em uma etapa anterior a identificação de sua posição, por exemplo com o uso do ICP.

Um segundo experimento de manipulação foi elabora, neste, a posição do objeto foi determinada utilizando o algoritmo ICP, que ajusta duas nuvem de pontos similares. A identificação da posição apresentou um erro variável (não foi observado um comportamento de variação do erro) com valor máximo de $40mm$, e para reduzir os efeitos deste erro, o mapa de impedância para a manipulação foi criada sobre a nuvem de pontos inicialmente detectada, cujo erro de posicionamento é em média de $10mm$, desconsiderando os erros de repetibilidade, uma vez que foi utilizada uma média de 10 aquisições.

Os resultados deste segundo experimento mostram que é possível melhorar o desempenho da manipulação utilizando informações diretas sistema de visão, ou seja, a nuvem de pontos resultantes do método apresentado no capítulo 2, sem necessidade de reconstrução e identificação do objeto. Os resultados também mostraram que o mapa de impedância, originalmente proposto para tratar colisões, também pode ser utilizado, sem modificação qualquer em sua metodologia, para a manipulação, que pode ser considerada como um caso de colisão desejada e controlada por uma trajetória cinemática adequada.

Os resultados obtidos no desenvolvimento do trabalho e nos experimentos respondem à questão do trabalho. Em primeiro lugar, é possível realizar tarefas de manipulação sem domínio completo do contato e sem controlar todos os efeitos de escorregamento e rolagem que ocorrem durante a manipulação. O controle de impedância mostrou-se eficiente no estabelecimento do contato e também na geração intrínseca das forças de manipulação, desde que seus parâmetros de impedância sejam adequados. Em segundo lugar, foi possível desenvolver um sistema de manipulação integrando funcionalidades de desvio e tratamento de colisão de forma simplificada. O sistema também apresentou robustez a erros de medida do sensor de posição. Este resultado sugere que é possível resolver problemas em robótica sem a necessidade de aquisição precisa de sensores de visão e nem a utilização de modelos detalhados do ambiente.

Estratégias de manipulação e controladores mais robustos e confiáveis capazes de tratar com colisões, contatos com objetos e robustos a erros de medida podem trazer avanços tanto para robôs de serviço quanto para robôs industriais. Estes avanços podem permitir trabalhos com objetos flexíveis e com sistemas de sensoriamientos com maiores grau de incertezas e erros de medida.

6.1 Proposta para trabalho futuro

Nos experimentos de manipulação realizados, a impedância adotada foi obtida empiricamente, porém, para generalizar o método a impedância deve ser previamente estabelecida, e ainda, é necessário modificar os parâmetros do controlador de forma adequada. Esta variação depende de vários fatores como velocidade de movimentação do objeto, precisão de movimentação, massa, entre outros. Muitos são os trabalhos que utilizam controle de impedância na manipulação ou que propõem algumas modificação no controlador mantendo suas características principais Goldenberg (1988), Schneider Schneider e Cannon (1992), Bonitz Bonitz e Hsia (1996), Moosavian Moosavian e Papadopoulos (1998), Okamura Okamura et al. (2000), Xu Xu e Li (2008), Lippiello Lippiello et al. (2007), Schlegl Schlegl et al. (2001). A hipótese funda-

mental para esta proposta é que uma mudança adequada de impedância seja capaz de melhorar a eficiência de sistemas de manipulação.

O objetivo da proposta seria investigar o comportamento da impedância humana em tarefas de manipulação, ou seja, identificar quais são os parâmetros de rigidez e amortecimento durante a movimentação livre e durante o contato com objetivo, e principalmente, identificar como ocorre a mudança entre as diferentes impedâncias.

Uma vez que os padrões de movimentação e força forem identificados e modelados, os parâmetros de impedância podem ser identificados através de relações dinâmicas, para tanto, é necessário medir as movimentações e as forças de contato de manipulação.

Referências Bibliográficas

- AGGARWAL, J. K.; CAI, Q. Human motion analysis: a review. *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, p. 90–102, 1997.
- AN, C. H.; HOLLERBACH, J. M. Dynamic stability issues in force control of manipulators. p. 821–827, 1987.
- BICCHI, A. *Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity*. [S.l.]: IEEE, 1999. 652–662 p.
- BICCHI, A.; KUMAR, V. Robotic grasping and contact: a review. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, v. 1, p. 348–353, 2002.
- BOISSONNAT, J.-D. Stable matching between a hand structure and an object silhouette. *IEEE Transactions on Pattern Analysis and machine intelligence*, v. 1, n. 6, p. 603–612, nov, 1982.
- BONITZ, R. G.; HSIA, T. C. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, v. 3, p. 944–949, 1996.
- CANNY, J. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 8, n. 6, p. 679–698, nov. 1986.
- CAURIN, G. A. P. et al. Adaptive strategy for multi-user robotic rehabilitation games. *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 11)*, 2011.
- CUTKOSKY, M. R. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 5, n. 3, p. 269–279, 1989.
- CUTKOSKY, M. R.; HYDE, J. M. Manipulation control with dynamic tactile sensing. *6th International Symposium on Robotics Research*, 1993.
- EPPINGER, S.; SEERING, W. Understanding bandwidth limitations in robot force control. v. 4, p. 904–909, 1987.
- EPPINGER, S. D.; SEERING, W. P. Three dynamic problems in robot force control. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 8, n. 6, p. 751–758, 1992.
- FERNANDES, G. *Exploração de ambientes não estruturados através de manipulador robótico implementando controlador de impedância com parâmetros variáveis*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.
- FERRARI, C.; CANNY, J. Planning optimal grasps. *Proceedings of the IEEE International Conference on Robotics and Automation*, v. 3, p. 2290–2295, mai 1992.

- FIGLIOLA, R. S.; BEASLEY, D. E. *Theory and Design for Mechanical Measurements*. [S.l.]: John Wiley & Sons Inc., 1991.
- FOX, D.; BURGARD, W.; THRUN, S. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, v. 4, n. 1, 1997.
- GOLDENBERG, A. A. Implementation of force and impedance control in robot manipulators. *Proc. IEEE International Conference on Robotics and Automation*, p. 1626–1632, 1988.
- GRIFFIS., M. W. *Kinesthetic Control: A Novel Theory for Simultaneously Regulating Force and Displacement*. Tese (Doutorado) — University of Florida, Gainesville, USA, 1991.
- HADDADIN, S. et al. The role of the robot mass and velocity in physical human-robot interaction - part ii: Constrained blunt impacts. In: *IEEE International Conference on Robotics and Automation - ICRA*. [S.l.: s.n.], 2008. p. 1339–1345.
- HADDADIN, S.; ALBU-SCHÄFFER, A.; HIRZINGER, G. Soft-tissue injury in robotics. *IEEE International Conference on Robotics and Automation - ICRA*, IEEE, p. 3426–3433, 2010.
- HOGAN, N. Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, v. 107, p. 1–24, mar. 1985.
- HOGAN, N. Controlling impedance at the man/machine interface. Scottsdale, AZ, p. 1626–1631, 1989.
- HYDE, J. M.; TREMBLAY, M. R.; CUTKOSKY, M. R. An object-oriented framework for event-driven dextrous manipulation. *The 4th International Symposium on Experimental Robotics IV*, Springer-Verlag, London, UK, UK, p. 53–61, 1997.
- JR., J. J. K.; LAVALLE, S. M. Rrt-connect: An efficient approach to single-query path planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, p. 995–1001, 2000.
- KAZEROONI, H. S. H.; HOUP, P. Robust compliant motion for manipulators, part i: The fundamental concepts of compliant motion. *J. Robot. Automat.*, v. 2, n. 2, p. 83–92, JUN 1986.
- KEMP, C. C.; EDSINGER, A.; TORRES-JARA, E. Challenges for robot manipulation in human environments [Grand Challenges of Robotics]. *Robotics & Automation Magazine, IEEE*, v. 14, n. 1, p. 20–29, 2007.
- KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Rob. Res.*, Sage Publications, Inc., Thousand Oaks, CA, USA, v. 5, n. 1, p. 90–98, abr. 1986.
- KHOSHELHAM, K. Accuracy analysis of kinect depth data. *GeoInformation Science*, v. 38, p. 1–6, 2010.
- KINECT. Xbox 360 + kinect. *Microsoft*, Dec 2011. Disponível em: <<http://www.xbox.com/en-us/kinect/>>. Acesso em: 20 December 2011.
- KJER, H. M.; WILM, J. *Evaluation of surface registration algorithms for PET motion correction*. Tese (Doutorado) — Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2010.

- KOLLER, D. et al. Towards robust automatic traffic scene analysis in real-time. p. 126–131, 1994.
- KRAGIC, D.; PETERSSON, L.; CHRISTENSEN, H. I. Visually guided manipulation tasks. *Robotics and Autonomous Systems*, v. 40, n. 2-3, p. 193–203, 2002.
- LAKSHMINARAYANA, K. Mechanics of form closure. *ASME Technical Report*, p. 1–8, 1978.
- LAWRENCE, D. A. Impedance control stability properties in common implementations. p. 1185–1190, 1988.
- LIPPIELLO, V.; SICILIANO, B.; VILLANI, L. A position-based visual impedance control for robot manipulators. *Proc. IEEE International Conference on Robotics and Automation*, p. 2068–2073, 2007.
- MACKENZIE, C. L.; IBERALL, T. *The grasping hand*. [S.l.]: North Holland, 1994.
- MOOSAVIAN, S. A. A.; PAPADOPOULOS, E. Multiple impedance control for object manipulation. *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, v. 1, p. 461–466, 1998.
- MURRAY, R. M.; SASTRY, S. S.; ZEXIANG, L. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, Inc., 1994.
- NGUYEN, V.-D. Constructing force-closure grasps. *International Journal of Robotics Research*, Sage Publications, Inc., Thousand Oaks, CA, USA, v. 7, n. 3, p. 3–16, 1988. ISSN 0278-3649.
- OHKA, M.; MORISAWA, N.; YUSSOF, H. B. Trajectory generation of robotic fingers based on tri-axial tactile data for cap screwing task. *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, p. 427–432, 2009.
- OKAMURA, A. M.; SMABY, N.; CUTKOSKY, M. R. An overview of dexterous manipulation. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, v. 1, p. 255–262 vol.1, 2000.
- OTSU, N. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, IEEE, v. 9, n. 1, jan. 1979. ISSN 0018-9472.
- PEDRO, L. M. *Fixação de objetos por garras de robôs: aplicações de redes neurais e proposta de auto-aprendizagem para os casos 2D e 3D*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.
- PEDRO, L. M.; CAURIN, G. A. P. Grasping unknown 3d objects using the hopfield-rbf-hopfield (hrh) approach. *Proceedings of the IASTED Robotics and Applications Conference*, p. , 2009.
- PELLETIER, M.; DOYON, M. On the implementation and performance of impedance control on position controlled robots. p. 1228–1233, 1994.
- PICCARDI, M. Background subtraction techniques: a review. *IEEE International Conference on Systems, Man and Cybernetics*, IEEE, v. 4, p. 3099–3104, out. 2004. ISSN 1062-922X.

- PRATT, G. A. et al. Late motor processing in low impedance robots: Impedance control of series-elastic actuators. 2004.
- PRIMESENSE. Dec 2011. Disponível em: <<http://www.primesense.com>>.
- RAIBERT, M. H.; CRAIG, J. J. Hybrid position/force control of manipulators. *Journal of Dynamic Systems Measurement and Control*, v. 102, n. 2, p. 126–133, 1980.
- ROSSL, C.; KOBELT, L.; SEIDEL, H.-P. Extraction of feature lines on triangulated surfaces using morphological operators. p. 71–75, 2000.
- SALISBURY, J. K. *Kinematic and force analysis of articulated hands*. Tese (Doutorado) — Stanford University, 1982a.
- SAUT, J.-P. et al. Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- SAXENA, A. et al. Learning to grasp novel objects using vision. *Proceedings of the International Symposium of Experimental Robotics*, 2006.
- SCHLEGL, T. et al. Fast dextrous re-grasping with optimal contact forces and contact sensor-based impedance control. *Proceedings 2001 ICRA IEEE International Conference on Robotics and Automation*, v. 1, n. 3-4, p. 103–108, 2001.
- SCHNEIDER, S. A.; CANNON, R. H. Object impedance control for cooperative manipulation: theory and experimental results. *Robotics and Automation, IEEE Transactions on*, v. 8, n. 3, p. 383–394, 1992.
- SICILIANO, B.; VILLANI, L. *Robot Force Control*. 1. ed. [S.l.]: Springer Int. Series Eng. Comput. Sci., 2000.
- STAUFFER, C.; GRIMSON, W. E. L. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, IEEE Computer Society, v. 2, p. 246–252, 1999.
- STÜCHELI, M. *Jaguaruna: A Trajectory Planner and Executor for the Kangaroo Robot Hand*. Dissertação — ETH Zurich, 2009. Disponível em: <<http://e-collection.library.ethz.ch/eserv/eth:783/eth-783-01.pdf>>.
- STÜCHELI, M.; PEDRO, L. M.; CAURIN, G. A. P. Squeezed screw trajectories for smooth regrasping movements of robot fingers. *Journal of Brazilian Society of Mechanical Engineering and Science*, 2013.
- TEGIN, J.; WIKANDER, J. Tactile sensing in intelligent robotic manipulation: a review. *Industrial Robot: An International Journal*, Emerald Group Publishing Limited, v. 32, n. 1, p. 64–70, 2005.
- WANG, L.; HU, W.; TAN, T. Recent developments in human motion analysis. *Pattern Recognition*, v. 36, p. 585–601, 2003.
- WREN, C. R. et al. Pfnder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, p. 780–785, jul. 1997. ISSN 01628828.

- XU, G.; SCHERRER, H.; SCHWEITZER, G. Application of neural networks on robot grippers. *Proceedings of the IEEE International Joint Conference on Neural Networks*, v. 3, n. 4, jun, 1990.
- XU, J.; KOO, T.-K. J.; LI, Z. Finger gaits planning for multifingered manipulation. *IROS*, p. 2932–2937, 2007.
- XU, J.; LI, Z. A kinematic model of finger gaits by multifingered hand as hybrid automaton. *Automation Science and Engineering, IEEE Transactions on*, v. 5, n. 3, p. 467–479, 2008.
- YANG, S.-l.; HU, J.-p. Jsn data exchange format and its application in data confirmation [j]. *Journal of Beijing Institute of Graphic Communication*, v. 4, p. 024, 2008.
- YIN, Y. et al. Hybrid control of multi-fingered robot hand for dexterous manipulation. *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, v. 4, p. 3639–3644 vol.4, 2003.
- ZEKI, S. *A Vision of the Brain*. [S.l.]: Oxford Univ Press, 1993.

APÊNDICE A – Avaliação do sensor Kinect

Esta seção apresenta resultados de experimentos para avaliação do sensor Kinect como sistema de aquisição de imagens RGB e de dados 3D. Avaliações são realizadas para verificar se o sensor apresenta características adequadas para aplicações de manipulação em ambientes não estruturados.

Baixo custo é sempre um dos requisitos que se pretende atender na seleção de equipamentos e técnicas para uma determinada aplicação. Neste contexto, desenvolvedores de diversas áreas têm utilizado o Kinect da Microsoft (KINECT, 2011) como *depth range sensor*.

Khoshelham (2010) realiza uma primeira avaliação de características de aquisição do Kinect. Os resultados obtidos dão um direcionamento aos desenvolvedores para aplicações em geral. Entretanto, os resultados são insuficientes para se determinar se o sensor atende os requisitos de aplicações em robótica. Assim, experimentos foram conduzidos para avaliar se o sensor apresenta características que atendem os requisitos para manipulação: precisão e repetibilidade de medida.

Três experimentos foram realizados para avaliar se o Kinect atende aos requisitos das aplicações em reabilitação assistida por robô e robótica, são eles: experimento de acuidade e repetibilidade de medida; experimento de registro; experimento de precisão de medida.

A.1 Avaliação da repetibilidade

O desvio padrão de uma serie de medidas de um mesmo mesurando é considerado como medida de repetibilidade de um instrumento, quando o desvio padrão é muito alto, diz-se que existe um grande erro de repetibilidade, ou seja, duas medidas de um mesmo mensurando podem apresentar valores muito diferentes.

O objetivo deste experimento foi: a) determinar qual é o erro de repetibilidade de medida de distância em x , y e z (definidos em um sistemas de referência do sensor - vide figura 1.4);

b) estabelecer como a repetibilidade varia em função de cada uma das direções; c) verificar em função de quais variáveis a repetibilidade se modifica. A partir dos dados obtidos neste experimentos foi possível determinar a faixa de operação mais adequada para as aplicações consideradas e quais os erros de repetibilidade esperados.

A.1.1 Experimento piloto de repetibilidade

Um primeiro experimento piloto foi realizado. Neste experimento foram feitas 30 medidas de uma mesma cena, mostrada na figura A.1. O objetivo é obter resultados que indiquem qual o procedimento mais adequado para o experimento final. A cena do experimento piloto consiste na vista superior de caixas e ferramentas sobre uma mesa. A altura do Kinect em relação ao solo é de aproximadamente 2m.

Figura A.1: Cena experimento piloto para a repetibilidade.

A média $\bar{x}$, $\bar{y}$ e $\bar{z}$ das medidas x , y e z foram calculadas para cada *pixel* quando o número de aquisições válida foi maior que 20. Existem problemas de aquisição de superfícies lisas com grande inclinação em relação ao plano de aquisição das imagens do sensor. Os desvios padrão σ_x , σ_y e σ_z associados às médias $\bar{x}$, $\bar{y}$ e $\bar{z}$ também foram calculados.

A figura A.2 contém imagens de representação do resultado para cada ponto de aquisição. Como cada *pixel* da imagem representa um ponto de aquisição, a representação das características das medidas é feita pela alteração da cor dos *pixels*. Os verdes são aqueles cujo número de aquisições válidas foi menor que 20. Os vermelhos são aqueles acima de um limiar empírico, utilizado para permitir uma melhor observação da variação em escala de cinza que representa

o desvio padrão. *Pixels* com escala de cinza máximo (*pixels* pretos) são aqueles com o menor desvio padrão, *pixels* com escala de cinza mínimo (*pixels* branco) são aqueles com desvio padrão igual ao limiar empírico. Para σ_x o limiar foi $4mm$, para σ_y foi $3mm$ e $7mm$ para σ_z .

Figura A.2: a) Imagem de representação do desvio padrão σ_x , b) Imagem de representação do desvio padrão σ_y , c) Imagem de representação do desvio padrão σ_z . *Pixels* com escala de cinza máximo (*pixels* pretos) são aqueles com o menor desvio padrão, *pixels* com escala de cinza mínimo (*pixels* branco) são aqueles com desvio padrão igual ao limiar empírico. Para σ_x o limiar foi $4mm$, para σ_y foi $3mm$ e $7mm$ para σ_z .

Analisando as imagens de representação do desvio piloto pôde-se verificar que há um aumento do centro para as extremidades tanto para σ_x quanto para σ_y , e também há um aumento com a distância de aquisição, observável pela diferença das aquisições da mesa e do chão. Desta forma, foi estabelecido um procedimento para análise da repetibilidade do sensor Kinect.

A.1.2 Experimento de repetibilidade

O procedimento experimental consiste na medida de um plano, paralelo ao plano xy do Kinect, em diversas alturas d conhecidas e controladas, e para isso, o Kinect foi posicionado no TCP de um robô industrial KUKA modelo KR16 com erro máximo de posicionamento de $0,05mm$. O sensor foi posicionado paralelo ao chão da sala por meio de um ajuste manual da orientação do TCP. A figura A.3 apresenta o ambiente experimental.

O mesmo procedimento do experimento piloto para cálculo das médias $\bar{x}$, $\bar{y}$ e $\bar{z}$ e de seus respectivos desvios padrão foi realizado para diferentes posições relativas d entre Kinect e plano de medida. As posições relativas de medida foram de $0,6m$ a $1,9m$ com passo de $0,1m$.

A figura A.4 mostra o desvio padrão σ_x em função da média $\bar{x}$ para $d = 1,8m$. A princípio é possível assumir, de forma intrínseca, que σ_x aumenta linearmente com o módulo de $\bar{x}$, contudo, a figura não permite avaliar qual a distribuição de pontos dentro de uma determinada faixa de valores de $\bar{x}$.

Figura A.3: Experimento de acuidade do Kinect.

Figura A.4: Desvio padrão σ_x em função de $\bar{x}$ para $d = 1,9m$.

Os dados apresentados na figura A.4 foram divididos em 1000 intervalos segundo $\bar{x}$. Para cada intervalo foi calculada a média do desvio padrão (FIGLIOLA; BEASLEY, 1991). O gráfico da figura A.5 apresenta a média do desvio padrão em função do valor médio de $\bar{x}$ de um intervalo. É possível verificar que a média do desvio padrão aumenta quadraticamente em função do módulo $\bar{x}$.

Figura A.5: Média σ_x para intervalos de $\bar{x}$. Dados foram divididos em 1000 intervalos ($d = 1,9m$).

O mesmo procedimento para se construir o gráfico da figura A.5 foi repetido para todas as distâncias d . A figura A.6 apresenta as curvas da média do desvio padrão σ_x em função de $\bar{x}$ e das diferentes distâncias de medida d .

Figura A.6: Média σ_x em função de $\bar{x}$ e d . A distância d foi variada de $0,6m$ a $1,9m$ em passos de $0,1m$.

A figura A.7 apresenta resultado similar ao da figura A.6 para σ_y médio em função de $\bar{y}$ e d . A variação de σ_z em função de $\bar{z}$ é mostrada na figura A.8. Neste caso $\bar{z}$ tem a mesma direção de d .

Figura A.7: Variação de σ_y médio em função de $\bar{y}$ e d .

Figura A.8: Variação de σ_z médio em função de $\bar{z}$.

Adicionalmente, foi verificado como a média do desvio padrão em uma direção varia em função das direções perpendiculares. Assim, as figuras A.9 e A.10 mostram como σ_z em função de distâncias nas direções $\bar{x}$ e $\bar{y}$, respectivamente. Pode ser observada a dependência quadrática de σ_z em relação a $\bar{x}$ e o seu aumento com d . Por outro lado, a mesma relação não pôde ser observada para σ_z em função de $\bar{y}$.

Figura A.9: Variação de σ_y médio em função de $\bar{x}$ e d .

Figura A.10: Variação de σ_z médio em função de $\bar{y}$ e d .

A mesma verificação foi realizada para σ_x e σ_y em relação a $\bar{y}$ e $\bar{x}$, respectivamente. Ambas as relações apresentaram o mesmo comportamento daquele observado na figura A.10, ou seja, não há aumento significativo de σ em função de sua direção perpendicular, ou seja, não há aumento de σ_x em função de $\bar{y}$, e não há aumento de σ_y em função de $\bar{x}$.

A.2 Experimento de precisão de posicionamento e de registro

Neste experimento o objetivo é avaliar a precisão da aquisição tridimensional do Kinect, e para tanto utiliza-se o método de detecção de pontos de identificação, apresentado na próxima seção, e o robô industrial KUKA de erro de posicionamento máximo de $0,05mm$ assegurado pelo fabricante para aplicações de precisão. Dada uma movimentação precisa do robô, a leitura dos pontos de identificação obtida com o Kinect foi comparada com o deslocamento realizado pelo TCP do robô.

A.2.1 Detecção de pontos de identificação

Muitos são os métodos de detecção de pontos de identificação em imagens, recomenda-se a leitura de Wang et al. (2003) e Aggarwal e Cai (1997) nos quais é feita uma revisão dos principais métodos utilizados para identificação de pontos de interesse no estudo de movimentos humanos. A complexidade destes métodos pode variar dependendo do tipo de identificador utilizado e do tipo de controle do ambiente.

O método proposto é baseado na existência do registro entre a imagem RGB e a *depth image* do Kinect. A figura A.11 mostra uma imagem RGB e o *depth image* de uma cena qualquer adquiridas pelo Kinect. As bordas da imagem RGB, detectadas pelo método de detecção de bordas proposto por Canny (1986), foram destacadas em verde em ambas as imagens. Neste exemplo é possível verificar que as bordas detectadas são equivalentes nas duas imagens.

O registro permite que pontos de identificação sejam reconhecidos na imagem RGB e suas coordenadas tridimensionais extraídas do pontos equivalente na *depth image*. Os pontos de identificação podem ser extraídos utilizando-se marcadores com cores e padrões conhecidos e posterior aplicação de processamento digitais de imagens.

Para simplificar a implementação e reduzir o custo computacional do processamento digital da imagem, a proposta é utilizar LEDs de alto brilho e foco aberto em associação com filtros de luz. O filtro de luz permite que apenas a luz emitida pelos LEDs seja capturada pela câmera RGB. A figura A.12 apresenta uma imagem RGB de uma cena contendo um LED sem e com o

Figura A.11: Exemplo de registro entre imagen RGB e *depth image* do Kinect.

filtro de luz (esquerda e direita respectivamente).

Figura A.12: Imagem RGB de um LED montado no robô SCARA. À esquerda a imagem sem o filtro de luz, e à direita a imagem com o filtro de luz.

Aplicando métodos de processamento de imagens de simplicidade relativa como binarização por Otsu (OTSU, 1979) é possível identificar quais *pixels* da imagem são aqueles correspondentes ao LED, ou aos LEDs no caso da existência de mais de um marcador. O método de Otsu determina automaticamente, a partir do histograma de uma imagem em tons de cinza, um limiar para binarização. *Pixels* com tom de cinza acima desse limiar recebem o binário 1 e abaixo o binário 0. A figura A.13¹ apresenta uma imagem binarizada pelo método de Otsu. O método de Otsu é proposto para se determinar automaticamente o valor do limiar de binarização, entretanto, este limiar pode ser estabelecido empiricamente em casos em que o método de

¹Para economia de impressão a representação usual da resposta do método de Otsu foi invertida. Os *pixels* em preto são aqueles que representam o LED. Adicionalmente foi inserida uma borda preta para evidenciar os limites da imagem.

Otsu apresentar falhas de binarização.

Figura A.13: Exemplo de binarização por Otsu. Limiar de binarização de 0,1.

Após a binarização, um LED é representado por uma região contendo vários *pixels* com o valor binário 1. Como existe o registro entre as imagens RGB e a *depth image*, a posição tridimensional do LED é obtida fazendo-se a média das posições x , y e z todos os *pixels* correspondentes ao LED.

A.2.2 Descrição do experimento

No experimento, o sensor Kinect foi fixado ao TCP do robô e um LED foi posicionado em um ponto qualquer em seu espaço de trabalho. O robô KUKA possui um procedimento de configuração de sistema de coordenadas da ferramenta, sistema T . Neste procedimento deve-se posicionar o robô em quatro pontos distintos, um sendo a origem do sistema de referência da ferramenta e os demais indicando suas direções x_T , y_T e z_T . Na execução deste procedimento os pontos de referenciação foram selecionados de tal forma que o sistema de referência da ferramenta estivesse alinhado com o sistema de referência do *depth sensor* do Kinect.

O sistema de comando do robô permite realizar movimentações lineares com passos constantes e conhecidos nas direções de um sistema de coordenada conhecido pelo robô. Uma vez que o sistema de coordenadas da ferramenta está alinhado com o do Kinect tem-se uma mesma movimentação nos dois sistemas.

No experimento, o TCP do robô foi movimentado em passos de $10mm$ nas direções x e y para uma mesma profundidade z . Por simplicidade, os movimentos foram restringidos a apenas um quadrante do espaço de aquisição do sensor (x positivo e y negativo). Para cada diferente posição do TCP foi realizada uma única aquisição. O procedimento foi realizado para 4 profundidades diferentes $z = 580mm$, $z = 680mm$, $z = 780mm$ e $z = 880mm$.

Figura A.14: Experimento de erro de posicionamento. À direita a escala de cores correspondente aos erros de posicionamento em *mm*.

A.3 Avaliação do Kinect para aplicações em robótica

Experimentos foram realizados para se determinar a repetibilidade do sensor. Um experimento piloto foi realizado para se avaliar qualitativamente o desvio padrão de uma medida. Neste teste pôde-se observar que o sensor apresenta boa repetibilidade em sua região central (tanto para x , y e z) e ainda, a repetibilidade piora com a distância z (profundidade) da aquisição.

A partir dos resultados do experimento piloto um método experimental foi elaborado. Neste método mediu-se a repetibilidade do sensor em x , y e z para diferentes distâncias de aquisição entre sensor e um plano. Análises estatísticas foram realizadas.

A análise dos dados obtidos indicam que o desvio padrão médio de uma medida aumenta quadraticamente com a distância de aquisição. O resultado confirma aquele obtido por Khoshelham (2010). Contudo, considerando um limite de desvio máximo aceitável, o sensor pode ser utilizado nas aplicações de interesse até 2 metros de aquisição. Além desta distância, os erros de repetibilidade são estatisticamente maiores que $2mm$.

Sobre o erro de precisão, avaliando a figura A.14 pode-se verificar que o erro de posicionamento do ponto de identificação não segue o mesmo padrão observado para o erro de repetibilidade. Nas extremidades do quadrante o erro de posicionamento torna-se elevado mesmo quando o erro de repetibilidade é baixo. Cerca de 80% dos valores de erro de posicionamento são menores que o erro de repetibilidade. O menor erro de posicionamento obtido foi de $0,036mm$, e o máximo de $12,25mm$. A média foi de $4,95mm$ e o seu desvio padrão de $2,09mm$.

Assim, pode-se concluir que o sensor Kinect apresenta boa repetibilidade e boa precisão em sua região central, e distâncias de aquisição maiores a 2 metros apresentam grandes erros de repetibilidade, mesmo para as regiões centrais.

APÊNDICE B – Reconhecimento de objetos com redes neurais

B.1 Introdução

Nesta apêndice estuda-se o desempenho de diversas RNAs na tarefa de classificação de objetos a partir de sua superfície 3D incompleta obtidas pelo sensor Kinect.

Atualmente na robótica é comum o uso de sensores 3D para aquisição de dados do ambiente. Por exemplo, na robótica móvel é importante a construção de um modelo virtual ou mapa do ambiente para que o robô possa se locomover sem colidir com obstáculos. Além da aplicação em robôs móveis autônomos, outras aplicações também requerem o conhecimento do ambiente, tal como a manipulação robótica, na qual é necessário classificar objetos para diversas finalidades, entre elas o desvio de obstáculo para evitar colisões (KHATIB, 1986).

Na aquisição de dados 3D de um objeto em uma cena, existe o problema da presença de faces ocultas. Os dados adquiridos por estes sensores 3D são conhecidos como depth image, imagem ou foto com profundidade, e da mesma forma que não se pode ver todo o entorno de um objeto em uma foto, com o sensores tipo depth image só é possível obter dados 3D da parte visível do objeto ao robô, ficando a sua face oculta ainda desconhecida. Devido a este problema principal e de outros secundários como ruídos, torna-se interessante a aplicação de RNAs para a classificação devido a sua robustez, tolerância e capacidade de generalização.

A classificação de objetos em classes geométricas, como cubos, cilindros e esferas, é tarefa fundamental para a manipulação de objetos. Porém, existe a dificuldade de classificá-los quando não se conhece toda a geometria superficial de um objeto, que é uma das características dos sensores 3D mais utilizados nestas aplicações.

Desta forma, pretende-se avaliar a capacidade de diferentes RNAs na tarefa de classificar objetos em classes geométricas a partir de informações incompletas de sua superfície obtidas por um sensor 3D do tipo depth image. Pretende-se utilizar dados reais do sensor Kinect.

Os dados serão utilizados nas etapas de treinamento e de avaliação das RNAs. As amostras de treinamento e testes consistem em nuvens de pontos de objeto de geometria simplificada: cubos, cilindros e esferas. Dados serão coletados destes objetos posicionados em diferentes posições em uma cena estática. Por simplificação, a cena na qual o objeto está não apresenta nenhum tipo de movimentação de pessoas e de objetos ou mudança na sua iluminação.

Após treinada, as redes serão avaliadas com amostras dos mesmo objetos não utilizadas em seu treinamento. Pretende-se avaliar qual é a rede mais adequada para a aplicação, assim como a sua melhor topologia.

Diversas RNAs são avaliadas: *Perceptron*; *Adaline*; Função de Base Radial (RBF do termo em inglês *Radial Basis Function*); e *Learning Vector Quantization* (LVQ). Nas avaliação, 104 amostras obtidas por um sensor 3D são utilizadas, sendo 26 amostras para cada uma das classes: esfera; cilindro em pé; cilindro deitado; e cubo. Resultados são apresentados para cada uma das redes em função de diferentes topologias e de diferentes quantidades de amostras de treinamento. Por fim, considerações finais sobre a eficiência das RNAs são apresentadas.

B.2 Normalização

A nuvem de pontos referente ao objeto é obtida aplicando o método descrito no capítulo 2, porém antes da aplicação das redes neurais, a nuvem de pontos precisa ser normalizada.

O primeiro passo da normalização é a simplificação que consiste na seleção de um número fixo de pontos que melhor representa o objeto. Esta etapa é necessária e fundamental pois para a aplicação na rede todas as amostras devem apresentar o mesmo número de pontos.

Aplica-se a simplificação para 100 pontos em todas as nuvens de pontos utilizadas. A figura B.1 apresenta um cilindro simplificado para 100 pontos utilizando o método proposto por Pedro e Caurin (2009).

Ainda como parte da normalização, as coordenadas foram redimensionadas de forma que a menor coordenada entre x , y e z seja 0 e a maior coordenada seja 1. O redimensionamento foi realizado com o mesmo fator nas três coordenadas para evitar distorções.

Figura B.1: Exemplo de nuvem de pontos de um cilindro simplificada para 100 pontos.

B.3 Testes das RNAs e resultados

Para o problema considerado neste trabalho, diversas RNAs foram aplicadas e avaliadas, são elas:

- Rede *Perceptron*;
- Rede Adaline;
- Rede *Perceptron* Multicamadas
- Rede RBF;
- Rede LVQ;
- Rede Kohonen.

Em todos os casos as redes foram configuradas como classificadoras de padrões. As topologias testadas para cada um dos tipos de redes serão apresentadas adiante.

Os dados utilizados no treinamento e nos testes de cada uma das redes são compostos por diversas nuvens de pontos de objetos adquiridas com o Kinect. A normalização destes dados foi realizada segundo o procedimento descrito na seção B.2. Um total de 104 nuvens de pontos de objetos foram adquiridas, sendo 26 para cada uma das diferentes classes consideradas: esfera; cilindro em pé; cilindro deitado; e cubo. Para cada uma delas foram atribuídas as seguintes saídas:

Estes dados foram divididos em um conjunto de treinamento e um de teste. Vários testes foram realizados variando-se a quantidade de dados em cada grupo com o objetivo de verificar a

Tabela B.1: Saídas desejadas para cada uma das classes.

	y ₁	y ₂	y ₃	y ₄
Classe esfera	1	0	0	0
Classe cilindro em pé	0	1	0	0
Classe cilindro deitado	0	0	1	0
Classe cubo	0	0	0	1

capacidade de classificação e de generalização de cada uma das redes. A tab. B.2 apresenta a divisão dos dados em 4 grupos.

Tabela B.2: Tabela de grupos de treinamento para os testes. Número de dados de treinamento e de testes para cada um das classes.

	Núm. dados de trein.	Núm. dados teste
Grupo 1	5	21
Grupo 2	10	16
Grupo 3	15	11
Grupo 4	20	6

B.3.1 Rede *Perceptron*

Classificação 1 - Classificação geral

A Rede *Perceptron* apresenta apenas um neurônio com saída binária (0 ou 1), ou bipolar (-1 ou 1), e desta forma ela é capaz de classificar apenas uma classe por vez.

Quatro diferentes redes foram treinadas, cada uma para classificar uma das diferentes classes geométricas. As capacidade de classificação das redes são as seguintes:

- Rede *Perceptron* 1 classifica a classe esfera;
- Rede *Perceptron* 2 classifica a classe cilindro em pé;
- Rede *Perceptron* 3 classifica a classe cilindro deitado;
- Rede *Perceptron* 4 classifica a classe cubo.

O treinamento e teste das redes foi realizado para cada um dos 4 grupos, com a resposta desejada apropriada, por exemplo, para a Rede *Perceptron* 1 a resposta desejada para as esferas é 1 e para as demais classes é -1. Foi possível treinar todas as Redes *Perceptron*, o que mostra que as classes são linearmente separáveis entre si. Os resultados em porcentagem de acertos para cada Rede *Perceptron* encontram-se na tab. B.3.

Tabela B.3: Resultados de classificação de objetos com diferentes redes *perceptrons* (classificação geral).

	G1	G2	G3	G4
Perc.1	67,9%	71,9%	72,7%	79,0%
Perc.2	80,9%	93,8%	97,7%	87,5%
Perc.3	69,0%	82,8%	88,6%	91,7%
Perc.4	61,9%	84,4%	79,5%	87,5%

Classificação 2 - Classificação por pares

Nesta classificação, a rede foi treinada para classificar as classes duas a duas, ou seja:

- Rede *Perceptron* 5 classificação entre esfera e cilindro em pé;
- Rede *Perceptron* 6 classificação entre esfera e cilindro em deitado;
- Rede *Perceptron* 7 classificação entre esfera e cubo;
- Rede *Perceptron* 8 classificação entre cilindro em pé e cilindro deitado;
- Rede *Perceptron* 9 classificação entre cilindro em pé e cubo;
- Rede *Perceptron* 10 classificação entre cilindro deitado e cubo.

Cada uma das Redes *Perceptron* foi treinada e avaliada para uma configuração diferenciada dos grupos. Cada um foi formado com os dados referentes ao seu domínio de classificação, ou seja, os grupos de treinamento da Rede *Perceptron* 7 é formado pelos dados das esferas e dos cubos, sendo que cada grupo tem diferentes quantidade de dados de treinamento e de testes conforme a tabela B.2. Os resultados obtidos são apresentados na tab. B.4.

Tabela B.4: Resultados de classificação de objetos com diferentes redes *perceptrons* (classificação por pares).

	G1	G2	G3	G4
Perc.5	80,9%	87,5%	90,9%	83,3%
Perc.6	59,5%	59,4%	77,3%	91,7%
Perc.7	38,1%	62,5%	63,6%	50,0%
Perc.8	80,9%	90,6%	86,4%	91,7%
Perc.9	97,6%	93,8%	100%	100%
Perc.10	40,4%	71,9%	81,8%	75,0%

B.3.2 Rede Adaline

Os mesmos testes de classificação realizados para a Rede *Perceptron* foram realizado para a Rede Adaline.

Classificação 1 - Classificação geral

Quatro redes *Adaline* foram implementadas para classificação geral de cada um das classes:

- Rede *Adaline* 1 classifica a classe esfera;
- Rede *Adaline* 2 classifica a classe cilindro em pé;
- Rede *Adaline* 3 classifica a classe cilindro deitado;
- Rede *Adaline* 4 classifica a classe cubo.

Os resultados obtidos em porcentagem de acertos encontram-se na tab. B.5 para cada uma das redes em função dos diferentes grupos de treinamento.

Tabela B.5: Resultados de classificação de objetos com diferentes reds *Adaline* (classificação geral).

	G1	G2	G3	G4
Adaline.1	56,6%	59,4%	65,9%	66,7%
Adaline.2	85,7%	89,1%	100%	91,7%
Adaline.3	75,0%	89,0%	95,5%	95,8%
Adaline.4	65,5%	85,9%	86,4%	70,8%

Classificação 2 - Classificação por pares

Redes *Adaline* também foram treinadas para classificação par a par, são elas:

- Rede *Adaline* 5 classificação entre esfera e cilindro em pé;
- Rede *Adaline* 6 classificação entre esfera e cilindro em deitado;
- Rede *Adaline* 7 classificação entre esfera e cubo;
- Rede *Adaline* 8 classificação entre cilindro em pé e cilindro deitado;
- Rede *Adaline* 9 classificação entre cilindro em pé e cubo;
- Rede *Adaline* 10 classificação entre cilindro deitado e cubo.

Os resultados obtidos em porcentagem de acertos encontram-se na tabela B.6 para cada uma das redes em função dos diferentes grupos de treinamento.

Tabela B.6: Resultados de classificação de objetos com diferentes redes *perceptrons* (classificação por pares).

	G1	G2	G3	G4
Adaline.5	78,6%	84,4%	86,4%	91,7%
Adaline.6	47,6%	62,5%	86,4%	66,7%
Adaline.7	47,6%	53,1%	59,1%	66,7%
Adaline.8	69,0%	96,9%	95,5%	91,7%
Adaline.9	88,1%	96,9%	100%	91,7%
Adaline.10	50,0%	71,9%	86,4%	83,3%

B.3.3 Redes *Perceptron* Multicamadas (PMC)

Várias topologias da Rede *Perceptron* Multicamadas foram implementadas, treinadas e avaliadas. Topologias de 2 e 3 camadas foram testadas com diferentes quantidades de neurônio em cada camada. Em todas as diversas topologias testadas, nenhum resultado satisfatório foi obtido.

B.3.4 Redes RBF

Redes RBF também foram implementadas para a classificação dos objetos. Diversas quantidades de neurônios na camada escondida foram testadas, são elas:

- RBF 1 com 4 neurônios na camada escondida;
- RBF 2 com 6 neurônios na camada escondida;
- RBF 3 com 8 neurônios na camada escondida;
- RBF 4 com 10 neurônios na camada escondida.

Para cada uma das configurações, treinamentos e testes foram realizados para cada um dos grupos da tabela B.2. Os resultados obtidos em porcentagem de acertos encontram-se na tabela B.7.

Tabela B.7: Resultados de classificação de objetos com diferentes redes RBF.

	G1	G2	G3	G4
RBF 1	61,9%	60,9%	70,5%	66,7%
RBF 2	25,0%	62,5%	61,4%	62,5%
RBF 3	46,4%	46,9%	54,5%	70,8%
RBF 4	25,0%	43,8%	45,5%	58,3%

B.3.5 Redes LVQ

Uma Rede Learning Vector Quantization (LVQ) também foi implementada para a classificação dos objetos. Como se conhece a quantidade de classes (4 classes), o número de neurônios foi fixado em 4, um para cada classe, e desta forma apenas um rede foi avaliada para os diferentes grupos de treinamento e testes definidos na tabela B.2. Os resultados obtidos encontram-se na tabela B.8.

Tabela B.8: Resultados de classificação de objetos com diferentes redes LVQ.

	G1	G2	G3	G4
Classe 1	47,6%	25,0%	0,0%	0,0%
Classe 2	71,4%	0,0%	100%	100%
Classe 3	42,86%	87,5%	100%	100%
Classe 4	76,19%	100%	100%	100%

O mesmo teste foi repetido, porém retirando a Classe esfera do treinamento e da avaliação. Os resultados encontram-se na tabela B.9.

Tabela B.9: Resultados de classificação de objetos com diferentes redes LVQ (sem a classe de esferas).

	G1	G2	G3	G4
Classe 2	95,2%	100%	100%	100%
Classe 3	71,4%	93,7%	100%	100%
Classe 4	100%	100%	100%	100%

O mesmo comportamento não pode ser observado quando se retira as demais classes do treinamento e avaliação. A taxa de acerto é sempre muito baixa para a classe 1 (esfera).

B.3.6 Redes Kohonen

Uma rede auto-organizável de Kohonen também foi implementada com mapa topológico bidimensional com 16 neurônios e com raio de vizinhança igual a 1.

Não foi possível observar separação entre classes no mapa topológico.

B.4 Avaliação da classificação de objetos com RNAs

A proposta da utilização de RNAs era classificar objetos a partir de sua nuvem de pontos obtidas por um sensor 3D e experimentos foram realizados para avaliar o desempenho de diversas RNAs nesta tarefa.

Além da Rede LVQ, outras RNAs classificadoras foram implementadas e avaliadas: Rede *Perceptron*, Rede Adaline, Rede *Perceptron* Multicamadas, Rede RBF e Rede Kohonen.

As Redes *Perceptron* Multicamadas (PMC) e Kohonen não apresentaram resultados satisfatórios na classificação.

Para a Rede PMC observou-se taxa de acerto de 25%, sendo que a mesma resposta foi observada para todos os testes (0 0 0 1). A Rede PMC também foi implementada no *toolbox* do Matlab e o mesmo resultado foi observado. Estima-se que o elevado número de entradas (301 entradas) foi o fator que levou aos erros de classificação.

Para a Rede de Kohonen, não foi possível observar separação clara entre os neurônios do mapa topológico ativados para cada uma das classes.

As Redes *Perceptron* e *Adaline* apresentaram resultados semelhantes nas duas implementações testadas. Na primeira implementação, a classificação geral, bons resultados foram obtidos para a Rede *Perceptron* que teve desempenho sempre acima de 70% para os Grupos de treinamento G2, G3 e G4. Para a classificação aos pares o desempenho de ambas foi muito semelhante, sendo que há muita variação, desde cerca de 40% até 100% de acertos.

Também foi possível obter resultados consideráveis com as redes RBF, porém, o desempenho não chega a superar 70%, e ainda, observou-se pouca variação no desempenho com o aumento do número de neurônios.

Os melhores resultados foram aqueles obtidos com as Redes LVQ, além da vantagem do tempo reduzido de treinamento. No primeiro testes, com todas as classes geométricas, observou um desempenho variável, indo de 0% a 100%, contudo, verificou-se que os erros eram mais frequentes para a Classe esfera. Ao se analisar os dados de treinamento e testes, observou-se a presença de muito ruído. As classes foram retiradas do treinamento e da avaliação e o desempenho chegou a 100% para todas as classes em dois dos grupos de treinamento.

A presença de ruídos na Classe Esfera pode ter induzidos aos erros de classificação de outras Redes, como a PMC, RBF e Kohonen. Nas Redes *Perceptron* e *Adaline* não se observou aumento de desempenho nos casos de classificação aos pares na qual a Classe Esfera não estava presente. Portanto, deve-se reavaliar as redes PMC e RBF com dados com menos ruídos.

APÊNDICE C – Principais publicações no período de doutorado

C.1 Periódicos

STUCHELI, M.; CAURIN, G.A.P.; PEDRO, L.M.; SIEGWART, R.Y. **Squeezed Screw Trajectories for Smooth Regrasping Movements of Robot Fingers**. Journal of the Brazilian Society of Mechanical Science and Engineering, v. 1, p. 1-10, 2013.

PEDRO, L.M.; BELINI, V.L.; CAURIN, G. A.P. **Learning how to grasp based on neural network retraining**. Journal of the Robotic Society of Japan, v. 27, p. 1-13, 2013.

CAURIN, G.A.P.; PEDRO, L.M. **Hybrid Motion Planning Approach for Robot Dexterous Hands**. Journal of the Brazilian Society of Mechanical Science and Engineering, 2009.

C.2 Capítulos de livro

PEDRO, L. M.; et. al. **Hand Gesture Recognition for Robot Hand Teleoperation**. ABCM Symposium Series in Mechatronics, 2012.

C.3 Congressos

PEDRO, L.M.; FERNANDES, G.; STUCHELI, M.; SIQUEIRA, A.A.G.; CAURIN, G.A.P. **A Robust Manipulation Strategy based on Impedance Control Parameters Changes and Smooth**. XVI International Conference on Advanced Robotics, ICAR, 2013.

FERNANDES, G.; PEDRO, L.M.; ZANONI, G.P.; CAURIN, G.A.P. **Detection Of Moving Objects For Robotic Manipulation Applications In Non-Structured Environments**. XXII International Congress of Mechanical Engineering - COBEM, 2013.

PEDRO, L.M.; FERNANDES, G.; SANTOS, W.M.; SIQUEIRA, A.A.G.; CAURIN, G.A.P. **Um Estudo De Desempenho Do Controle De Impedância Para Tarefas De Contato Utilizando Robôs Industriais**. XI Simpósio Brasileiro de Automação Inteligente - SBAI, 2013.

CAURIN, G.A.P.; PEDRO, L.M.; FERNANDES, G.; BELINI, V.L. **Framework For Automatic Manipulation Using Robot Hands And Grippers**. DINAME, 2013.

PEDRO, L.M.; CAURIN, G.A.P. **Kinect Evaluation for Human Body Movement Analysis**. IEEE BIOROB, 2012.

PEDRO, L.M.; CAURIN, G.A.P. **Learning based grasping using neural network retraining**. In: IROS 2010 Workshop On Grasp Planning and Task Learning by Imitation, 2010, Taipei.

PEDRO, L.M.; CAURIN, G.A.P. **Grasping Unknown 3D Objects using the Hopfield-RBF-Hopfield (HRH) Approach**. In: 14th IASTED International Conference on Robotics and Applications, 2009.

BELINI, V.L.; MATSUO, D.T.; PEDRO, L.M.; STUCHELI, M.; CAURIN, G. A.P.; PURQUERIO, B. M. **Integrated System for the development and test of robot hands and grippers**. In: XIII International Symposium on Dynamic Problems of Mechanics - DINAME, 2009.