

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

Cristiane Pescador Tonetto

**UMA SISTEMATIZAÇÃO DA MODELAGEM E
PROGRAMAÇÃO CINEMÁTICA DE SISTEMAS
ROBÓTICOS COOPERATIVOS PARA A REALIZAÇÃO
DE TAREFAS**

Florianópolis

2011

Cristiane Pescador Tonetto

**UMA SISTEMATIZAÇÃO DA MODELAGEM E
PROGRAMAÇÃO CINEMÁTICA DE SISTEMAS
ROBÓTICOS COOPERATIVOS PARA A REALIZAÇÃO
DE TAREFAS**

Tese submetida ao Programa de Pós-
Graduação em Engenharia Mecânica
para a obtenção do Grau de Doutor
em Engenharia Mecânica.
Orientador: Prof. Altamir Dias, D.Sc.
Coorientador: Prof. Henrique Simas,
Dr. Eng.

Florianópolis

2011

Catálogo na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

T664s Tonetto, Cristiane Pescador

Uma sistematização da modelagem e programação cinemática de sistemas robóticos cooperativos para a realização de tarefas [tese] / Cristiane Pescador Tonetto ; orientador, Altamir Dias. - Florianópolis, SC, 2011.

220 p.: il., grafs., tabs.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica.

Inclui referências

1. Engenharia mecânica. 2. Robótica. 3. Cinemática. 4. Linguagem de programação (Computadores). 5. Robôs industriais. I. Dias, Altamir. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

CDU 621


Cristiane Pescador Tonetto

**UMA SISTEMATIZAÇÃO DA MODELAGEM E
PROGRAMAÇÃO CINEMÁTICA DE SISTEMAS
ROBÓTICOS COOPERATIVOS PARA A REALIZAÇÃO
DE TAREFAS**

Esta Tese foi julgada adequada para a obtenção do Título de “Doutor em Engenharia Mecânica”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Mecânica.

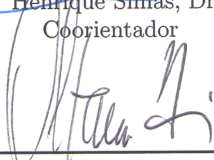
Florianópolis, 16 de setembro 2011.

Prof. Júlio César Passos, Dr.
Coordenador do Curso



Prof. Henrique Simas, Dr.Eng.
Coorientador


Banca Examinadora:




Prof. Altamir Dias, D.Sc.
Presidente



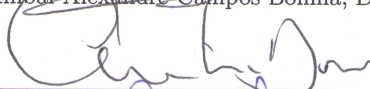
Prof. Marcelo Becker, Dr.



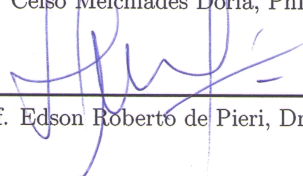
Prof. Sebastião Cícero Pereira Gomes, Dr.



Prof. Anibal Alexandre Campos Bonilla, Dr.Eng.



Prof. Celso Melchiades Doria, PhD.



Prof. Edson Roberto de Pieri, Dr.

Aos meus pais, Humberto e Olívia.
Ao meu marido, Mathias.
À minha irmã, Cláudia.

AGRADECIMENTOS

Aos meus pais Humberto e Olívia, e à minha irmã Cláudia, cujos os esforços e o apoio não foram poupados na minha educação, por todo amor, carinho e incentivo. Amo muito vocês.

Ao meu marido Mathias, que aqui as palavras não são suficientes para dizer tudo o que representa para mim, tenho um imenso respeito e amor por você. Agradeço muito por fazer parte da minha vida, por compartilhar todos os momentos e torná-los especiais.

Ao professor Altamir Dias, pela orientação, incentivo, dedicação e conhecimento compartilhado na realização deste trabalho.

Aos professores Henrique Simas, Marcelo Becker, Sebastião Cícero Pinheiro Gomes, Alexandre Campos, Celso Melchiadés Dória e Edson Roberto de Pieri pela dedicação e disponibilidade ao lerem o trabalho. Ao professor Daniel Martins, que não pode estar presente na banca, mas que muito ajudou na realização deste trabalho. E ao professor Raul Guenther (*in memoriam*), pela referência em lecionar. Muito obrigada à todos pelas contribuições.

Aos colegas de laboratório, em especial ao Carlos, que apesar do “lá é bem melhor”, foi aqui que a amizade começou, e que as longas conversas e discussões aconteceram.

À minha amiga Ana Maria Basei pela companhia no chá das 5, pelas conversas e pela amizade.

À CAPES e ao CNPq por terem financiado este projeto pelo período de 3 anos e 1 ano, respectivamente.

“E aqui descobri o que de fato não procurava, mas, ao encontrar, quis possuir para sempre.”

J.R.R. Tolkien

RESUMO

A utilização de robôs na indústria vem aumentando devido as exigências como qualidade do produto, custo, agilidade de produção, entre outros, provocado pela competitividade entre as empresa. As indústrias têm investido em robôs buscando satisfazer tais exigências e seu chão de fábrica é composto por inúmeros robôs, e dependendo da aplicação é interessante fazer com que esses robôs realizem tarefas de forma cooperativa. Um Sistema Robótico Cooperativo (SRC) é um sistema composto por vários robôs que auxiliam ou colaboram na realização de uma ou mais tarefas simultaneamente. A programação desses robôs para atuarem de forma cooperativa na realização de tarefas é complexa, pois existe uma relação de dependência entre eles. Neste trabalho é criada uma sistemática para a modelagem e programação cinemática de SRC. A sistemática é generalizada para trabalhar com um número arbitrário de robôs, sendo independente do número e tipo de juntas e focada na especificação das várias tarefas. Para implementar a sistemática foram definidos três ambientes para a programação robótica de SRC: a estrutura robótica, o ambiente da tarefa e a cinemática diferencial. Esses ambientes estabelecem os modelos dos objetos envolvidos: robôs e tarefas. Uma formalização matemática é apresentada demonstrando que o crescimento do SRC é estruturado, conforme são adicionados novos robôs ao sistema. Esta formalização matemática permite introduzir no SRC as características adicionais como o controle do erro de integração, evitamento de colisão e deslocamentos relativos das bases dos robôs com o sistema de coordenadas fixo. Por fim, são apresentados os casos de estudos analisando a sistemática desenvolvida para SRCs. Em um dos casos é destacada a programação de um SRC composto pelos robôs XR-4 e Scara da fabricante Rhino, nesse sistema são realizadas três abordagens diferentes para a execução da tarefa e os resultados das simulações são implementados nos robôs. Em um outro caso é apresentado o sistema composto por quatro robôs e especificadas as tarefas. A partir desse cenário são feitas alterações nos parâmetros e analisado o impacto que elas provocam no desempenho do sistema. Para mostrar a flexibilidade da sistemática da modelagem, é também realizada uma nova simulação em que um dos robôs tem a base móvel.

Palavras-chave: Sistema robótico cooperativo, cinemática diferencial, programação de robôs, tarefas industriais.

ABSTRACT

The application of robots in the industry is growing due to the needs of product quality, cost and production speed, that are reinforced by the companies competition. The industries are investing in robots that fulfil those needs and their shop-floors are composed by several robots, and, depending on the application, it may be useful that those robots could accomplish tasks cooperatively. A Cooperative Robotics System (CRS) is a system composed by several robots that help each other or collaborate in the accomplishment of one or more tasks simultaneously. The robots programming process, in order to operate cooperatively in the task accomplishment, is complex, since there is a dependency relationship between the robots. In this work, a systematization for the kinematics modeling of CRS is created. The systematization is general and can be applied to an arbitrary number of robots, each one with different number and kind of joints, focusing the multiple tasks specification. For the systematics implementation, three environments for robotics programming were defined: the robotics structure, the task environment and the differential kinematics. These environments establish the models of the objects: robots and tasks. A mathematical formalization is presented, showing that the growing of CRS is structured while adding new robots to the system. After the mathematical formalization, it is possible to introduce in the CRS the additional characteristics, such as the integration error control, collision avoidance and relative displacements of the robots bases according to the fixed coordinate system. Finally, study cases are presented, analysing the developed systematization for CRS. In the first case, the programming of a CRS composed by the XR-4 and Scara robots (built by Rhino) is described. Three different approaches for the task execution and the simulation results are implemented in the robots. In the second case a system composed by four robots and its tasks specifications are presented. Starting from a reference scenario, changes in the systems parameters are made, and their impact on the system performance is evaluated. The third case shows the flexibility of the modeling systematics, by adding a mobile base to one of the robots.

Keywords: Cooperative Robotics System, differential kinematics, robot programming, industrial tasks.

LISTA DE FIGURAS

Figura 1	Classificação de sistemas multirrobõs.....	30
Figura 2	Suprimento anual de rob̃s industriais no mundo pelas principais indústrias nos anos de 2008 e 2009(IFR, 2010).....	40
Figura 3	Número previsto de vendas de rob̃s de serviçõs para o período de 2010-2013 (IFR, 2010).....	40
Figura 4	Métodos de geração de caminhos	43
Figura 5	Descrição do processo de planejamento de trajetórias robóticas.	44
Figura 6	Fluxograma para o cálculo da cinemática de rob̃s.	52
Figura 7	Manipuladores – Scara e Antropomórfico (SICILIANO et al., 2009).	53
Figura 8	Parâmetros de Denavit-Hartenberg (SICILIANO et al., 2009).....	54
Figura 9	Deslocamento helicoidal de um ponto P (TSAI, 1999)..	55
Figura 10	Fluxograma para o cálculo da cinemática de um SRC. .	57
Figura 11	Sistema robótico composto por um rob̃o.....	58
Figura 12	Representação por grafos do sistema composto por um rob̃o e duas cadeias virtuais.....	60
Figura 13	Grafos que representam SRCs: com $1, 2, 3, \dots, m$ rob̃os...	63
Figura 14	Representação por grafos de um SRC com a base dos manipuladores fixa.	66
Figura 15	Sistema composto por um rob̃o.	75
Figura 16	Sistema composto por um rob̃o com um grau de liberdade e seu grafo.....	75
Figura 17	Sistema composto por um rob̃o de n graus de liberdade e seu grafo.....	76
Figura 18	Sistema composto por um rob̃o de $n + 1$ graus de liberdade.....	77
Figura 19	Grafo de um sistema composto por um rob̃o com n graus de liberdade.	79
Figura 20	Grafo do sistema robótico composto por m rob̃os.	79
Figura 21	Grafo do sistema robótico composto por $m + 1$ rob̃os. .	82
Figura 22	Adição de grafos em série.	88
Figura 23	Grafo F gerado ao se adicionar o grafo H	89

Figura 24	Adição de grafos em paralelo.....	90
Figura 25	Grafo F gerado ao se adicionar um grafo H	91
Figura 26	Sistema robótico cooperativo com m cadeias virtuais de erro adicionadas em série.....	96
Figura 27	Representação de um SRC com cadeias virtuais para controle de juntas.....	98
Figura 28	Representação por grafos de um SRC no qual se evidencia o relacionamento entre dois elos quaisquer de robôs distintos.....	100
Figura 29	SRC com o agente externo.....	101
Figura 30	Diagrama de classes dos programas implementados.....	107
Figura 31	Robô Rhino XR-4 (esquerda); Robô Rhino Scara (direita).....	109
Figura 32	Robôs participantes do sistema com as devidas identificações.....	109
Figura 33	Trajetória a ser seguida: (a) pelo efetuator do robô XR-4 com relação à referência da peça; (b) pela peça com relação ao sistema de coordenadas fixo.....	111
Figura 34	Representação do sistema com a adição das cadeias virtuais de Assur e da tarefa.....	112
Figura 35	Grafo do SRC e sua representação simplificada.....	112
Figura 36	Posição e velocidade das juntas do robô XR-4 escrevendo o acrônimo UFSC.....	114
Figura 37	Posição e velocidade das juntas dos robôs XR-4 e Scara.....	116
Figura 38	Grafos dos três casos destacando as juntas primárias e secundárias.....	117
Figura 39	Posição e velocidade das juntas dos robôs XR-4 e Scara trabalhando em cooperação.....	118
Figura 40	Comparação da posição das juntas do robô XR-4 entre os três casos.....	119
Figura 41	Comparação da velocidade das juntas robô XR-4 entre os três casos.....	120
Figura 42	Sistema robótico cooperativo e sua representação por grafos.....	123
Figura 43	Robôs executando a tarefa cooperativamente.....	125
Figura 44	Gráfico mostra a posição das juntas para cada robô.....	125
Figura 45	Indicação das modificações no sistema: características da tarefa (número 1), distância entre os robôs (número 2) e confi-	

gurações iniciais (número 3).....	127
Figura 46 Posição das juntas dos robôs para o cenário G.....	132
Figura 47 Peça sobre a qual o robôs executarão suas tarefas.....	134
Figura 48 SRC e sua representação por grafos.....	135
Figura 49 Sequência de execução da tarefa com base móvel do robô IRB 140.....	137
Figura 50 Gráfico das posições de cada uma das juntas dos robôs e o gráfico das posições da base do robô IRB 140.....	138
Figura 51 Robô XR-4 executando a tarefa.....	144
Figura 52 Resultado da tarefa executada pelo robô XR-4.....	145
Figura 53 Os robôs XR-4 e Scara executando as tarefas em coope- ração.....	145
Figura 54 Resultado da tarefa executada pelos robôs em coope- ração.....	146
Figura 55 Caminho de referência e caminho resultante da simula- ção.....	146
Figura 56 Comparação do erro entre o Caso 1 e o Caso 2.....	147
Figura 57 Cinemática direta e inversa.....	167
Figura 58 Deslocamento helicoidal de um ponto P (TSAI, 1999).....	169
Figura 59 Manipulador planar de dois elos com duas juntas rotati- vas.....	172
Figura 60 Manipulador planar com duas juntas prismáticas e uma rotativa.....	173
Figura 61 Transformação de helicoides.....	180
Figura 62 Mecanismo planar com 5 barras (CAMPOS, 2004).....	183
Figura 63 Classificação das cadeias virtuais.....	187
Figura 64 Cadeia virtual planar PPR (CAMPOS, 2004).....	187
Figura 65 Cadeia virtual planar RPR (CAMPOS, 2004).....	188
Figura 66 Cadeia virtual espacial $PPPS$ (CAMPOS, 2004).....	190
Figura 67 Cadeia virtual espacial $RPPS$ (CAMPOS, 2004).....	191
Figura 68 Cadeia virtual espacial $RRPS$ (CAMPOS, 2004).....	192
Figura 69 (a) Manipulador planar de dois elos com três juntas ro- tativas (R_1 , R_2 e R_Z) e duas juntas prismáticas (P_X e P_Y) (b) Grafo representativo do manipulador.....	194
Figura 70 (a) Grafo não-direcionado; (b) Grafo direcionado.....	198
Figura 71 Grafo direcionado e seus circuitos (MAYEDA, 1972).....	200
Figura 72 Um dos robôs deixa de segurar a peça.....	211

Figura 73 Um SRC segura uma peça.	212
Figura 74 Cadeia cinemática fechada como $M = 5$. Para os elos 1 e 4: $K_{14} = 3$, $C_{14} = 3$ e $R_{14} = 0$ e para os elos 1 e 5: $K_{15} = 4$, $C_{15} = 3$ e $R_{15} = 1$ (CARBONI, 2008).	213

LISTA DE TABELAS

Tabela 1	Características dos principais trabalhos relacionados a SRC.....	49
Tabela 2	Valores de s e s_o para as juntas do manipulador com relação a uma configuração de referência.....	111
Tabela 3	Especificação do Robô: ABB IRB 6620.....	121
Tabela 4	Especificação do Robô: ABB IRB 1600.....	122
Tabela 5	Especificação do Robô: ABB IRB 140.....	122
Tabela 6	Especificação do Robô: ABB IRB 120.....	122
Tabela 7	Posição da base dos robôs e configuração inicial das juntas.....	123
Tabela 8	Variações dos parâmetros nos diferentes cenários.....	128
Tabela 9	Variações nos parâmetros nos cenários de referência e nos de A até F.....	130
Tabela 10	Variações nos parâmetros nos cenários de G até M.....	131
Tabela 11	Parâmetros das tarefas impostas ao SRC.....	135
Tabela 12	Indicadores de avaliação do SRC com a base de um dos robôs móvel.....	137
Tabela 13	Estrutura do arquivo enviado para os manipuladores (este exemplo se refere ao manipulador XR-4, com 5 juntas).....	142

LISTA DE ABREVIATURAS E SIGLAS

SRC	Sistema Robótico Cooperativo	29
CRS	<i>Cooperative Robotics System</i>	29
SMR	Sistemas Multirrobo	30
IFR	Federação Internacional de Robótica	41
APT	<i>Automatically Programmed Tool</i>	42
D-H	Denavit-Hartenberg	48
PPR	Cadeia virtual planar com duas juntas prismáticas e uma junta rotativa	68
RPR	Cadeia virtual planar com uma junta prismática e duas juntas rotativas	68
3P3R	Cadeia virtual espacial com três juntas prismáticas e três juntas rotativas	68
RPPS	Cadeia virtual espacial com duas juntas prismáticas, uma junta rotativa e uma junta esférica	68
RRPS	Cadeia virtual espacial com uma junta prismática, duas juntas rotativas e uma junta esférica	68
<i>HI</i>	Hipótese de Indução	77
GPL	Licença Pública Geral	106
LGPL	Licença Pública Geral Menor V3	106
CAD	<i>Computer Aided Design</i>	108
STEP	<i>STandard for the Exchange of Product model data</i>	108
UFSC	Universidade Federal de Santa Catarina	108
MS-DOS	<i>Microsoft Disk Operating System</i>	141
LAI	Laboratório de Automação Industrial	144
USB	<i>Universal Serial Bus</i>	144

LISTA DE SÍMBOLOS

a_i	distância entre os sistemas de coordenadas fixo aos elos do robô	54
d_i	direção do eixo de coordenada de O'_i ao longo de z_{i-1}	54
α_i	ângulo entre os eixos z_{i-1} e z_i com relação ao eixo x_i	54
ϑ_i	ângulo entre os eixos x_{i-1} e x_i com relação ao eixo z_{i-1}	54
s	é o vetor unitário que indica a direção do eixo de helicóide da junta i	54
s_o	é o vetor posição de um ponto de s com relação a um sistema de coordenadas fixo da junta i	54
m	número de robôs	56
R_i	robô i	58
CV_0	cadeia virtual que relaciona o movimento/posição relativo da peça com o sistema de coordenadas fixo (base)	58
CV_i	cadeia virtual que relaciona o movimento entre o efetuador do robô i e a peça	59
n	número de juntas	60
B	matriz de circuitos	60
λ	ordem do sistema de helicóides, onde $\lambda = 6$ para casos espaciais e $\lambda = 3$ para casos planares	61
D	matriz de helicóides normalizados	61
$\$$	helicóide	61
$\hat{\$}$	helicóide normalizado	61
P_{X0}	junta prismática da CV_0 com deslocamento no eixo X	61
P_{Y0}	junta prismática da CV_0 com deslocamento no eixo Y	61
P_{Z0}	junta prismática da CV_0 com deslocamento no eixo Z	61
R_{X0}	junta rotativa da CV_0 com rotação em torno do eixo X	61
R_{Y0}	junta rotativa da CV_0 com rotação em torno do eixo Y	61
R_{Z0}	junta rotativa da CV_0 com rotação em torno do eixo Z	61
P_{Xi}	junta prismática da CV_i com deslocamento no eixo X	61
P_{Yi}	junta prismática da CV_i com deslocamento no eixo Y	61
P_{Zi}	junta prismática da CV_i com deslocamento no eixo Z	61
R_{Xi}	junta rotativa da CV_i com rotação em torno do eixo X	61
R_{Yi}	junta rotativa da CV_i com rotação em torno do eixo Y	61

R_{Zi}	junta rotativa da CV_i com rotação em torno do eixo Z.....	61
N	matriz de rede.....	61
N_p	submatriz contendo os helicoides referentes às juntas primárias.....	62
N_s	submatriz contendo os helicoides referentes às juntas secundárias.....	62
\dot{q}	vetor contendo a magnitude da velocidade das juntas.....	62
\dot{q}_p	vetor contendo a magnitude da velocidade das juntas primárias.....	62
\dot{q}_s	vetor contendo a magnitude da velocidade das juntas secundárias.....	62
L	número de circuitos independentes de um grafo.....	74
e	número de arestas do grafo.....	75
v	número de vértices do grafo.....	75
l	número de circuitos do grafo.....	84
F_b	somatório dos graus de liberdade de todas as juntas da cadeia cinemática.....	85
N_e	matriz que contém os helicoides referentes a cadeia virtual de erro.....	94
\dot{q}_e	vetor de magnitudes correspondente as velocidades das juntas de controle de erro.....	94
$[A_p]$	matriz de transformação homogênea correspondentes às juntas primárias.....	95
$[A_s]$	matriz de transformação homogênea correspondentes às juntas secundárias.....	95
E_i	matriz homogênea do erro de fechamento.....	95
$\hat{\$}_{CV_{ei}}$	helicóide normalizado referente a cadeia virtual de erro do robô i	96
CV_{ci}	cadeia virtual de controle de junta do robô i	99
$\hat{\$}_{A_i}$	helicóide normalizado referente ao agente externo do robô i	102
D_i	distância da base do robô i ao sistema de coordenadas fixo.....	110
W	matriz de pesos.....	117
J	Jacobiano.....	168
θ	ângulo de rotação.....	170
t	translação.....	170
ω	velocidade angular.....	170

v_p	velocidade linear	170
h	passo de helicoides	171
$A_{\vec{x},\theta}$	matriz de Rodrigues	181
T	árvore	198
T'	árvore de espalhamento	199

SUMÁRIO

1 INTRODUÇÃO	29
1.1 DEFINIÇÃO DE SISTEMAS ROBÓTICOS COOPERATIVOS	30
1.2 CONTEXTUALIZAÇÃO	32
1.3 MOTIVAÇÃO	34
1.4 OBJETIVOS GERAL E ESPECÍFICOS	35
1.5 ORGANIZAÇÃO DA TESE	36
2 REVISÃO BIBLIOGRÁFICA	39
2.1 SITUAÇÃO ATUAL DA ROBÓTICA NA INDÚSTRIA ...	39
2.2 DESCRIÇÃO DE TAREFAS PARA UM SRC	41
2.3 PLANEJAMENTO DE TRAJETÓRIAS	42
2.4 TRABALHOS RELACIONADOS A SRC	45
2.5 CONCLUSÕES	50
3 PROGRAMAÇÃO CINEMÁTICA DIFERENCIAL E TAREFAS ROBÓTICAS PARA SRC	51
3.1 AMBIENTES PARA A PROGRAMAÇÃO ROBÓTICA ...	51
3.2 AMBIENTES PARA A PROGRAMAÇÃO DE SRC	55
3.3 PROGRAMAÇÃO ROBÓTICA	57
3.3.1 Programação robótica simples	57
3.3.2 Programação robótica de SRC	63
3.4 ETAPAS DO PROCESSO DE PROGRAMAÇÃO DE SRC .	69
3.5 CONCLUSÕES	71
4 FORMALIZAÇÃO MATEMÁTICA PARA O CÁLCULO DA CINEMÁTICA DE SRC	73
4.1 FORMALIZAÇÃO MATEMÁTICA DO GRAFO DE UM SRC	73
4.2 FORMALIZAÇÃO MATEMÁTICA DA MATRIZ DE REDE DE UM SRC	84
4.3 ADIÇÃO DE GRAFOS NUM SRC	88
4.4 CONCLUSÕES	91
5 CARACTERÍSTICAS ADICIONAIS A SRC	93
5.1 CADEIA VIRTUAL DO ERRO	93
5.1.1 Controle do erro de integração em SRC	95
5.2 CONTROLE DE JUNTAS: EVITAMENTO DE COLISÃO .	97
5.2.1 Controle de juntas em SRC	98
5.3 DESLOCAMENTOS RELATIVOS DAS BASES DOS ROBÔS COM O SISTEMA DE COORDENADAS FIXO	101

5.4	CONCLUSÕES	102
6	SIMULAÇÕES DE SRC	105
6.1	DESENVOLVIMENTO DE UM PROGRAMA PARA CINEMÁTICA DE SRC	105
6.2	SIMULAÇÃO DOS ROBÔS RHINOS: XR4 E SCARA	108
6.3	SIMULAÇÃO DE QUATRO ROBÔS EM COOPERAÇÃO	121
6.3.1	Modificações em alguns parâmetros do SRC	126
6.4	SIMULAÇÃO DE UM SRC COM BASE MÓVEL	134
6.5	CONCLUSÕES	138
7	APLICAÇÕES EM SISTEMAS REAIS	141
7.1	CARACTERÍSTICAS DO SISTEMA	141
7.2	DESAFIOS E PROBLEMAS ENCONTRADOS DURANTE A APLICAÇÃO DA METODOLOGIA EM UM CASO REAL	142
7.2.1	Desafios técnicos	142
7.2.2	Desafios estruturais	143
7.3	RESULTADOS EXPERIMENTAIS	144
7.4	VANTAGENS DA METODOLOGIA APLICADA	147
7.5	CONCLUSÕES	148
8	CONCLUSÕES E TRABALHOS FUTUROS	149
8.1	CONCLUSÕES	149
8.2	TRABALHOS FUTUROS	152
	Referências Bibliográficas	155
	APÊNDICE A – Cinemática diferencial de robôs	167
	APÊNDICE B – Aspectos Gerais: Transformações de Helicoides, Método de Davies e Cadeias Virtuais de Assur	179
	APÊNDICE C – Teoria de Grafos	197
	APÊNDICE D – Pseudo-inversa adaptada para SRC	207
	APÊNDICE E – Redundância em SRC	211

1 INTRODUÇÃO

Os robôs têm ganhado uma certa atenção na indústria nos últimos anos, visto que eles proporcionam algumas vantagens na realização de tarefas repetitivas e insalubres. Além disso, a utilização de robôs pode aumentar a produção, reduzir custos e melhorar a qualidade final do produto ou serviço. As indústrias de grande porte têm em sua linha de fabricação inúmeros robôs trabalhando de forma independente. Acredita-se que para algumas aplicações seja conveniente que esses robôs trabalhem conjuntamente, em cooperação.

Um Sistema Robótico Cooperativo (SRC) possui vários robôs e tarefas, o que o torna um sistema complexo e de difícil programação. Por outro lado, a flexibilidade que o sistema robótico cooperativo proporciona definir os parâmetros e distribuir as tarefas de forma a melhorar o desempenho do sistema. Devido a essa flexibilidade e a complexidade é essencial a elaboração de técnicas que permitam a modelagem, simulação e análise de desempenho do sistema programado. Assim, esta tese tem como contribuição principal a sistemática de programação cinemática de Sistemas Robóticos Cooperativos.

Um SRC é importante porque com ele é possível realizar tarefas que um único robô não conseguiria. Um único robô pode não ser capaz de realizar o deslocamento de uma carga, quando ela for grande ou pesada demais, mas o deslocamento poderia ser realizado com a carga compartilhada com dois ou mais robôs, o que caracteriza um SRC. Um outro exemplo é uma soldagem que exige a reorientação da peça durante o processo, onde, um robô posiciona a peça enquanto o outro efetua a soldagem. A sistemática proposta nesta tese permite que problemas como estes sejam modelados e simulados.

Neste capítulo é apresentada uma descrição geral sobre sistemas multirrobôs com a finalidade de definir sistemas robóticos cooperativos. Em seguida é realizada a contextualização, comentando-se sobre alguns aspectos gerais da programação e as funcionalidades de SRC. Posteriormente aborda-se a motivação em se trabalhar com SRC, levantando-se algumas questões em aberto sobre SRC. Por fim, são apresentados os objetivos geral e específicos, bem como a estrutura da tese.

1.1 DEFINIÇÃO DE SISTEMAS ROBÓTICOS COOPERATIVOS

Quando o espaço em um ambiente industrial é ocupado por vários robôs, é dito que nesse ambiente existem sistemas multirrobôs (SMR), ou seja, um conjunto de elementos que dividem o mesmo ambiente e que possuem algum grau de relacionamento entre eles (ZINANOVIC; VUKOBRATOVIC, 2006). Um sistema multirrobô é classificado em três categorias relevantes de comportamento: coordenação, cooperação e a competição (RIBEIRO, 2010). Essa classificação é sintetizada em um diagrama ilustrado na Figura 1.

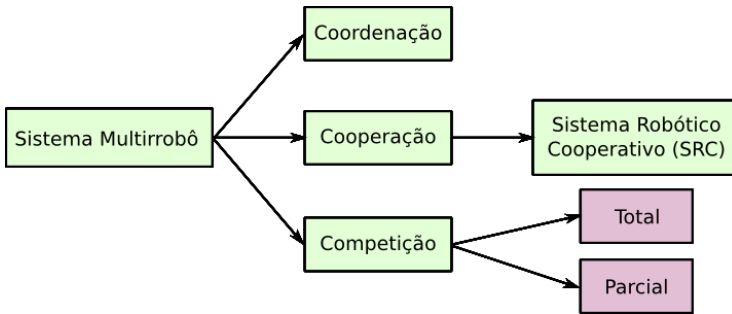


Figura 1: Classificação de sistemas multirrobôs.

O sistema multirrobô é classificado como de **coordenação** quando cada robô realiza suas tarefas sem se preocupar com as tarefas dos outros robôs, sendo que não existe conflito algum entre cada tarefa com as demais. Pode-se dizer que na coordenação não existe qualquer relação de dependência entre as tarefas e nem entre os robôs, ou seja, as tarefas realizadas por cada um dos robôs são desacopladas e os robôs também não possuem acoplamento entre eles. Contudo, o espaço operacional é compartilhado entre os robôs. Um exemplo de uma aplicação em que os robôs trabalham coordenadamente seriam dois robôs próximos, onde cada um realiza uma operação sobre a superfície da sua peça. Em linhas de montagem industrial, os robôs geralmente atuam coordenadamente.

O sistema multirrobô de **cooperação** é definido pela capacidade de vários robôs atuarem juntos na realização de um objetivo maior. Existe a realização de uma ou mais tarefas, mas diferentemente da coordenação, as tarefas ou os robôs guardam alguma relação de depen-

dência entre si. Os robôs têm a função de auxiliar na realização da tarefa, um robô depende do trabalho do outro. Como os movimentos de um robô afetam e são afetados pelos demais, caso ele não execute a sua tarefa corretamente, comprometerá a execução de todas as tarefas. Uma aplicação comum de robôs trabalhando cooperativamente é quando eles compartilham uma carga. Um outro exemplo seria um robô posicionando e um segundo robô executando alguma operação sobre a peça.

A **competição** ocorre quando deixa de existir cooperação, seja pela limitação de recursos ou por interesses conflitantes, onde apenas um ou alguns robôs possam realizar suas tarefas. Na competição sempre existe intersecção do espaço de trabalho entre os robôs e com isso, há possibilidade de conflito o tempo todo. Um exemplo de competição entre robôs é o caso onde existem ferramentas que são compartilhadas (competição de recursos), ou tarefas distintas sobre uma mesma região do objeto que exija acesso restrito a apenas um robô (competição de acesso), ou a exigência por parte da tarefa sobre a orientação da peça, em que duas tarefas distintas exijam orientações conflitantes da peça (competição de posicionamento). Entre SMR, a competição pode ser classificada como total, em que todos os robôs envolvidos competem entre si, e parcial, quando uma parcela dos robôs compete com uma outra parcela. Uma modificação do diagrama originalmente apresentado em Ribeiro (2010) foi realizada, na qual foram adicionadas duas subclasses na categoria competição: total e parcial (Figura 1).

Dentro da perspectiva de planejamento de tarefas robóticas com sistemas compostos com mais de um robô, o foco desta pesquisa será o Sistema Robótico Cooperativo - SRC.

Neste sentido, o dicionário Aurélio (FERREIRA, 2004) define o termo **cooperar** como:

1. Operar, ou obrar, simultaneamente; colaborar;
2. Prestar colaboração, serviços; trabalhar em comum;
3. Ajudar; auxiliar;

A primeira definição formal encontrada é a de Ribeiro (2010), que define um **sistema robótico cooperativo** da seguinte forma: “*é todo sistema composto por múltiplos manipuladores robóticos industriais, onde estes estejam atuando de modo cooperativo em prol da realização de múltiplas tarefas que guardem algum tipo de relação de dependência entre si.*”

No sentido de tornar a definição mais geral, assume-se nesta tese a definição que: **Um Sistema Robótico Cooperativo (SRC) é um sistema composto por vários robôs, que auxiliam ou colaboram na realização de uma ou várias tarefas simultaneamente.** Essa definição confere com o senso comum ilustrado no dicionário e, apesar de mais geral, é equivalente a definição proposta por Ribeiro (2010). Um sistema robótico cooperativo pode ser composto por qualquer tipo de robô (manipuladores, mãos robóticas, robôs móveis, robôs subaquáticos), não necessariamente com a mesma estrutura física.

A partir deste momento, o termo sistema robótico cooperativo será abreviado pela sigla SRC no decorrer deste texto. Os sistemas cooperativos abordados neste trabalho são compostos por robôs manipuladores.

1.2 CONTEXTUALIZAÇÃO

Os sistemas robóticos cooperativos têm sido um tópico de interesse especial na programação de tarefas robóticas nos últimos 20 anos. Segundo Navarro-Alarcon et al. (2009), vários manipuladores trabalhando em uma dada tarefa podem melhorar a destreza de execução, aumentar a capacidade de carga, prover uma maior flexibilidade de manipulação de objetos, entre outros. Como consequência dessas vantagens da utilização de SRC, tem-se uma agilidade no processo de produção, sendo que uma determinada tarefa pode ser realizada mais rapidamente por mais robôs quando comparada a um único robô em uma linha de produção. Além disso, com a utilização de SRC surgem possibilidades de criação de tarefas que não podem ser realizadas por apenas um robô (CACCAVALE; UCHIYAMA, 2008).

Navarro-Alarcon et al. (2009) destacam ainda a importância de braços robóticos cooperativos redundantes na aplicação de tarefas cooperativas, enfatizando que os benefícios e as possibilidades aumentam ainda mais nestes casos, pois resultam em múltiplas configurações cinemáticas disponíveis para uma dada posição e orientação do efetuador. Além de poder utilizar a redundância do sistema para evitar colisões com obstáculos enquanto manipula um objeto, pode-se também reconfigurar o próprio sistema a fim de otimizá-lo.

Um sistema robótico composto por dois ou mais manipuladores cooperativos pode executar tarefas mais complicadas e manipular peças com formas complexas. Contudo, a cooperação entre manipuladores não é amplamente utilizada devido ao planejamento do movimento

e sua programação serem mais difíceis (CACCAVALE; UCHIYAMA, 2008). Assim, os SRCs apresentam um desafio de estudo devido ao alto grau de acoplamento entre os robôs para a execução de tarefas, levando a problemas de programação de tarefas como evitamento de colisão, otimização de trajetórias e evitamento de singularidades.

A programação *on-line*¹ de um SRC não é trivial, devido a existência de interação entre os robôs. Os robôs devem estar bem sincronizados e realizar as suas tarefas simultaneamente, o que seria impraticável caso operadores distintos comandassem separadamente os robôs. Adicionalmente, essa operação exigiria muitos homens/horas para a programação, além de ocupar as máquinas impedindo sua utilização na produção.

Existem algumas interfaces que auxiliam a programação *off-line*² de um robô no planejamento, simulação e programação de tarefas robóticas. São exemplos de ambientes com essas características os programas: Workspace (SOLUTIONS, 2011), V-REP (FREESE, 2011), RobotStudio (ABB, 2011), KUKA.SIM PRO (KUKA, 2001), MotoSim[®] EG(MOTOMAN, 2011) e Plantra (TONETTO; DIAS, 2008). Porém esses programas são limitados com relação a SRC, no sentido de não suportarem vários robôs e caso aceitem adicionar mais robôs no ambiente, não possuem uma abordagem direcionada para cinemática de sistemas em cooperação, que permita a interação entre os robôs. Com isso, falta um ambiente para a programação de SRCs, ou seja, um ambiente em que seja possível simular e programar vários robôs executando tarefas de forma cooperativa.

Uma característica própria de robôs em cooperação é a influência que um robô poderá exercer sobre os outros durante a execução de tarefas. Essa característica surge devido à necessidade de auxílio mútuo durante a execução da tarefa. Atualmente, quando um ou mais robôs industriais trabalham juntos dividindo o mesmo ambiente, a fim de realizar uma tarefa comum, a programação deles é feita independente, ou seja, o programador configura cada um dos robôs definindo a trajetória que ele irá executar e cada robô realiza sua tarefa sem que haja trocas de informações entre si. O problema em lidar com essa forma de programação é que os robôs podem não estar perfeitamente sincronizados, podendo haver colisão entre eles. Além disso, se um dos robôs parar de executar sua tarefa devido alguma falha técnica, os outros robôs provavelmente continuarão realizando as suas tarefas, e assim, pode também ocorrer algum dano ao sistema. A programação independente de robôs,

¹Termo em português: Em linha.

²Termo em português: Fora de linha

que compartilham o ambiente de trabalho, exclui imediatamente a possibilidade de realizar um grande conjunto de tarefas, que são aquelas que somente podem ser executadas com a cooperação de mais de um robôs.

1.3 MOTIVAÇÃO

A utilização de SRCs propicia uma série de vantagens como, por exemplo: a redução do tempo necessário na execução de tarefas, possibilidade de realizar novas tarefas, maior flexibilidade e destreza na manipulação de objetos, quando comparados a de apenas um robô. Porém, a vantagem que mais se destaca é a possibilidade de realizar as tarefas que um único robô nunca conseguiria. Desta forma, se existem tarefas que possuem limitações quanto a sua execução com apenas um robô, elas podem ser realizadas por vários robôs simultaneamente. Essa é uma das características mais importantes quando se trata de SRC.

No sentido de amenizar a dificuldade de programação de SRC, é importante o desenvolvimento de metodologias e técnicas para a determinação de parâmetros e para o auxílio no planejamento da tarefa.

Na execução de tarefas por robôs cooperativos é possível formular as seguintes perguntas de pesquisas:

1. **Existe alguma forma de programar a cinemática de um SRC para a execução de qualquer tarefa e com um número arbitrário de robôs?**
2. **Qual metodologia pode ser aplicada para calcular a cinemática de um SRC?**
3. **É possível simplificar o processo de programação de SRC para que a partir das especificações das tarefas e dos robôs se obtenham opções de solução para as trajetórias dos robôs?**
4. **Como ocorre o aumento no número de parâmetros e variáveis na programação de SRC, conforme são adicionados robôs e tarefas ao sistema?**
5. **Caso se desenvolva uma sistemática para a programação de SRC, é possível generalizá-la e aplicá-la a qualquer SRC?**

6. Dada uma tarefa, é possível determinar o número e o modelo ideal de robôs que formarão o SRC?

Na literatura observa-se que estas questões são tratadas de forma a solucionar um problema específico, cuja solução não pode ser utilizada e nem generalizada em outros sistemas. Assim, buscando responder a essas perguntas, ao longo deste trabalho serão focados alguns aspectos importantes com relação às informações mínimas necessárias à programação de robôs em cooperação. Será analisado se esses dados têm alguma relação com a estrutura dos robôs, levando-se em consideração os parâmetros e limites de velocidades e acelerações das juntas e consequentemente do efetuador, necessários à execução da tarefa. Além disso, devem ser consideradas as informações relativas ao planejamento dos caminhos e trajetórias a serem percorridos pelo efetuador de cada robô no processo de execução das tarefas.

1.4 OBJETIVOS GERAL E ESPECÍFICOS

O objetivo geral desse trabalho é sistematizar a modelagem cinemática de SRCs para a automatização do processo de programação de SRCs no planejamento e execução de tarefas robóticas.

São objetivos específicos:

- estruturar a programação de SRC e definir os ambientes que a compõe;
- modelar as cadeias cinemáticas para SRCs diferentes, através da adição estratégica de cadeias virtuais de Assur;
- determinar a cinemática diferencial (inversa e direta) para SRC, em função da modelagem da cadeia cinemática para qualquer SRC;
- formalizar matematicamente a construção do modelo cinemático de SRC;
- estender a determinação da cinemática para SRCs com características adicionais (controle de junta, deslocamento relativo, controle do erro);
- planejar as trajetórias para o desenvolvimento de tarefas a serem executadas por SRC;

- definir um método para a geração automática de trajetórias dos robôs componentes de um SRC a partir do modelo cinemático deste e da especificação da tarefa;
- implementar o método de geração automática de trajetórias, realizar simulações e execução do planejamento em um sistema real.

Os detalhes sobre os estudos da cinemática são comentados no Capítulo 3 e comentários mais específicos são descritos no Apêndice A.

1.5 ORGANIZAÇÃO DA TESE

O trabalho foi elaborado em sete partes, expostas ao longo dos capítulos desta pesquisa.

No Capítulo 2 será apresentada a situação atual da robótica no mundo, mostrando a perspectiva de vendas de robôs até 2013. Além disso, é comentado o conceito de trajetórias e as principais técnicas de planejamento encontradas na literatura. Por fim, são descritos os principais trabalhos relacionados a SRC e como eles abordam o desafio de programá-los para a execução de tarefas.

No Capítulo 3 são definidos os ambientes principais envolvidos na programação de SRC, sendo considerados os três principais: estrutura robótica, tarefa e cinemática diferencial. A partir deles é detalhado o cálculo da cinemática para um robô, em seguida, para um SRC composto por m robôs e apontado as principais diferenças. É descrita também uma análise sobre o crescimento da cadeia cinemática de SRC conforme são adicionados novos robôs. Por fim, é comentada uma rotina para o cálculo da cinemática de SRC baseada na teoria de helicoides, no método de Davies e nas cadeias virtuais de Assur.

A formalização matemática da metodologia é apresentada no Capítulo 4. Ela tem como base a teoria de grafos e a equação de Euler. Baseada nelas são provados dois teoremas: um deles associa o número de robôs ao número de circuitos independentes do grafo e o outro teorema relaciona o número de robôs com o crescimento da matriz de rede N . Com esses teoremas se obtém uma estrutura de modelagem genérica de SRC. Além disso, duas definições denominadas de adição em série e adição em paralelo e suas propriedades subsequentes mostram o significado de se adicionar subgrafos em um SRC.

No Capítulo 5 é introduzido o conceito de características adicionais em SRC, baseado nas Propriedades 1 e 2 que foram apresentadas no Capítulo 4. São exemplos de características adicionais os elementos

que visam complementar a programação de um SRC e auxiliar na realização da tarefa, como o tratamento da colisão, o controle do erro e os movimentos relativos entre as bases dos robôs, entre outros. Assim, com esses cinco capítulos tem-se todo o embasamento para tratar da programação de SRC de forma mais genérica possível incluindo todos os casos, além de ter subsídio matemático para definir novas características.

Nos Capítulos 6 e 7 são apresentados os exemplos simulados e a implementação dos resultados em um SRC real. Também são propostos critérios para a avaliação e análise de desempenho de um SRC: velocidade máxima, picos de aceleração e deslocamento médio das juntas. No Capítulo 6 é descrito o procedimento para a simulação de três SRCs distintos, um deles é composto por dois robôs (Scara e um XR-4) e definida uma tarefa. Para esse sistema são comentados três casos possíveis de execução da tarefa. Os resultados obtidos nessa simulação são implementados em um sistema real e comentados no Capítulo 7. Os outros dois SRCs são compostos por quatro robôs. Para um deles é estabelecido um cenário de referência para a realização da tarefa e, com relação a esse cenário, são alterados alguns parâmetros (configuração inicial, distância entre robôs e dados da tarefa) criando novos cenários. Além disso, alguns critérios de avaliação de desempenho do sistema são propostos. Por fim, o terceiro SRC é simulado, em que um dos robôs terá a base móvel, simulando estar sobre uma plataforma móvel. Esse exemplo faz referência a uma característica adicional introduzida no Capítulo 5.

No Capítulo 8 são feitas as considerações finais sobre a metodologia desenvolvida e os resultados obtidos. São comentadas algumas sugestões de trabalhos futuros relacionados a programação de SRC.

Este trabalho possui ainda 5 apêndices:

No Apêndice A é comentado sobre a cinemática diferencial para um robô utilizando os métodos de Denavit-Hartenberg e a teoria de helicoides.

No Apêndice B são apresentadas as ferramentas básicas para o auxílio no cálculo da cinemática como: transformações de helicoides, método de Davies e cadeias virtuais de Assur.

A teoria de grafos é apresentada no Apêndice C, que auxiliará na leitura do Capítulo 4 sobre a formalização matemática.

Por fim, nos Apêndices D e E é descrita a pseudo-inversa adaptada para SRC e é abordada a redundância em SRC, respectivamente.

2 REVISÃO BIBLIOGRÁFICA

O objetivo principal da instalação de um robô na indústria é a execução de alguma tarefa, nesse sentido, é muito importante que se dê enfoque na tarefa a ser executada por ele. Assim, estratégias de planejamento de trajetórias são abordadas durante todo o processo de programação, já que as tarefas exigem o movimento das peças e dos efetadores dos robôs.

A programação robótica convencional, composta por apenas um robô, possui um nível de abordagem bem estabelecido na literatura. Porém, pouco enfoque é dado no processo de programação de sistemas robóticos que trabalham em cooperação, no sentido de organizar e estabelecer sistemáticas para determinar o cálculo da cinemática de SRC.

A programação robótica envolve várias etapas desde a descrição das tarefas e robôs até o planejamento da trajetória robótica. Estas etapas possibilitam uma visão geral dos estágios necessários para a descrição das tarefas, detalhando os movimentos que os robôs precisam realizar.

Neste capítulo são apresentados os principais trabalhos encontrados na literatura que focam em sistemas compostos por dois ou mais robôs e na sua programação para execução de tarefas, além de comentar como os autores resolvem a programação cinemática desses sistemas.

2.1 SITUAÇÃO ATUAL DA ROBÓTICA NA INDÚSTRIA

O ano de 2008 foi um dos melhores anos com relação a instalação de novos robôs. Porém, devido à crise econômica mundial, em 2009 a venda de robôs industriais teve uma queda se comparada aos anos anteriores. A queda nas vendas chegou ao nível mais baixo desde 1994. A instalação de novos robôs nunca teve um decréscimo tão grande, chegando a ser de 47% (60.000 unidades a menos que em 2008) (IFR, 2010). A Figura 2 mostra claramente esse decréscimo em praticamente todas as áreas industriais em suprimentos anual de robôs no ano 2009. Percebe-se também que a área automotiva estava fazendo um investimento considerável em robótica e no ano de 2009 chegou a reduzir pela metade a compra de novas unidades (IFR, 2010).

A perspectiva de vendas de robôs para o período de 2010-2013 é ilustrada na Figura 3, na qual, existe uma tendência de aumento con-

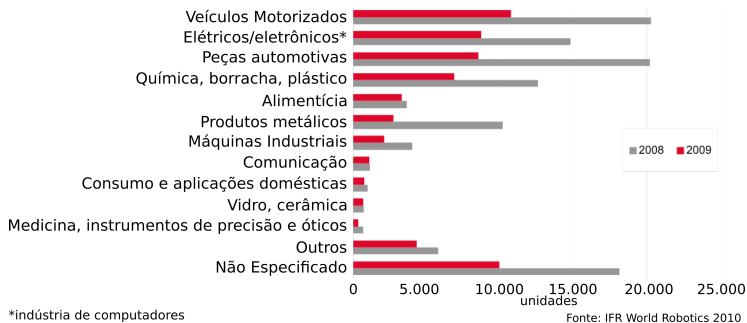


Figura 2: Suprimento anual de robôs industriais no mundo pelas principais indústrias nos anos de 2008 e 2009(IFR, 2010).

siderável de quase 23.000 unidades no ramo da agricultura, principalmente com robôs ordenhadores. Em seguida percebe-se uma tendência de aumento de robôs aplicados em defesa, podendo-se chegar a venda de 19.000 unidades de robôs novos até 2013. É destaque também a previsão de vendas de 10.000 unidades de plataformas móveis e 10.000 unidades para outros serviços (IFR, 2010).

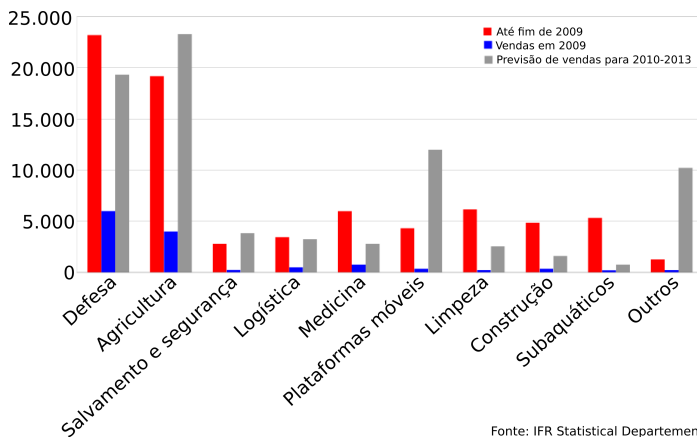


Figura 3: Número previsto de vendas de robôs de serviços para o período de 2010-2013 (IFR, 2010).

Não foi encontrada uma classificação mais específica nas estatís-

ticas realizadas pela Federação Internacional de Robótica – IFR (IFR, 2010) de quais aplicações robóticas utilizam vários robôs de forma coordenada ou cooperativa. Mas visualizando a Figura 2 nota-se que um grande número de robôs é empregado na fabricação de veículos motorizados e em peças automotivas. Visitando os endereços eletrônicos, como os das empresas KUKA (2011) e ABB (2011), encontra-se exemplos de aplicações em que é possível operar conjuntamente mais de um robô para a aplicação na fabricação de veículos. Porém, nenhuma das empresas oferece no seu catálogo soluções próprias para cooperação e coordenação, mas descrevem ser possível implantar tais soluções.

2.2 DESCRIÇÃO DE TAREFAS PARA UM SRC

A descrição da tarefa consiste em estabelecer quais os estágios necessários para a programação e planejamento de tarefas por um SRC. Zinanovic e Vukobratovic (2006) comentam uma classificação que aborda as características essenciais do processo de realização de uma tarefa por um sistema cooperativo. Tem-se como característica básica, *a priori*, que a descrição de tarefas considera que a peça irá sofrer a influência dos robôs em um ambiente (estruturado ou não-estruturado). Nessa classificação, o estágio completo do processo de realização de uma tarefa por um sistema robótico é dado por:

- planejamento de aproximação;
- aproximação da peça;
- execução da tarefa;
- retirada do efetuador para a posição de descanso.

Na etapa de planejamento de aproximação, são escolhidas as trajetórias de movimento livre antes do efetuador entrar em contato com a peça. A aproximação é a etapa de realização do movimento da trajetória definida anteriormente, em que os manipuladores se deslocam em direção à peça e termina ao tocá-la. Nesta etapa nenhuma força é estabelecida entre a garra e a peça. A execução da tarefa é a etapa que compreende no robô pegar, segurar (estabelecendo a força de contato), levantar, transferir/mover, baixar e soltar a peça, nos casos onde a tarefa consiste em manipular objetos. Nos casos em que as tarefas são soldagem, pintura, usinagem, compartilhamento de carga, etc, a execução de tarefas consiste em seguir uma determinada trajetória com ou

sem contato. A etapa retirada do efetuador corresponde a um movimento livre em que o robô se afastará da peça manipulada, mantendo uma certa distância.

Na descrição das etapas necessárias à realização de tarefas é exigido um movimento do efetuador por um caminho contínuo. Esse movimento necessita de um planejamento mais detalhado e serão tratados como planejamento de trajetórias, conforme descrito a seguir.

2.3 PLANEJAMENTO DE TRAJETÓRIAS

O objetivo do planejamento de trajetórias é gerar dados de referência de entrada para que o manipulador seja capaz de executar o movimento desejado. Geralmente as entradas do algoritmo de planejamento de trajetórias são o caminho geométrico, as restrições de velocidades e acelerações relacionadas à qualidade necessária para a execução das tarefas, respeitando os limites das juntas dos robôs. A saída é a trajetória no espaço de juntas expressa como uma sequência no tempo dos valores de posição, velocidade e aceleração (GASPARETTO; ZANOTTO, 2008; LAVALLE, 2006).

O caminho geométrico é definido no espaço operacional, e tanto a tarefa a ser executada quanto os obstáculos a serem evitados são descritos nesse espaço. O caminho definido no espaço operacional significa que as trajetórias a serem definidas são as referentes a um sistema de coordenadas no efetuador do robô.

Para tarefas que exigem o recobrimento de superfícies, existem estudos de técnicas de geração do caminho geométrico. Para Chen, Ni e Wu (1993) existem basicamente três técnicas de geração de caminhos: baseado em *Automatically Programmed Tool* (APT), geração do caminho no espaço de trabalho e geração do caminho no espaço paramétrico da superfície. Uma classificação mais específica desses métodos pode ser observada em Suresh e Yang (1994), em que são considerados os métodos APT e iso-paramétrico propostos por Chen, Ni e Wu (1993). Suresh e Yang (1994) descrevem ainda três métodos de geração de caminho: iso-planar, *iso-offsets* e *iso-scallop*. A Figura 4 ilustra os três métodos frequentemente utilizados na geração do caminho da ferramenta: (a) iso-paramétrico, (b) iso-planar e o (c) *iso-scallop*. Outros autores trabalham com a geração de caminho para tarefas de recobrimento de superfícies como é o caso de Sarma e Dutta (1997), Feng e Li (2001), Tonetto e Dias (2007), Simas (2008).

O requisito mínimo desejado do efetuador de um robô é que

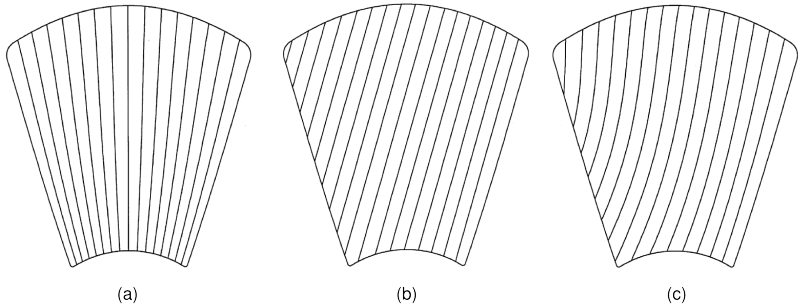


Figura 4: Métodos de geração de caminhos: (a) iso-paramétrico, (b) iso-planar e o (c) *iso-scallop* (FENG; LI, 2001).

ele tenha a capacidade de movimentar-se de uma posição inicial até a posição final desejada. Essa transição deve ser feita seguindo algumas restrições determinadas pelo sistema do robô e pelos processos de fabricação como as citadas a seguir (SICILIANO et al., 2009; ZHA, 2002):

- as trajetórias do suporte da ferramenta ou do efetuador do robô precisam ser suaves, devendo ser levadas em consideração as continuidades de percurso, de velocidade e de aceleração. Além disso, as trajetórias devem ser geradas no menor intervalo de tempo possível;
- em algumas operações que exijam qualidade superficial (alguns processos de usinagem, por exemplo), é necessária uma precisão na trajetória levando em consideração a distância entre os caminhos, a fim de controlar a rugosidade da superfície trabalhada;
- devem ser evitadas as regiões de singularidades no movimento das juntas do robô e o movimento do robô não pode ultrapassar os limites do intervalo (de operação) das juntas, suas velocidades, acelerações e torques/forças máximos;

Existe uma relação entre o planejamento da trajetória e o planejamento do caminho (SICILIANO et al., 2009), de forma que usualmente ambas as atividades precisam ser executadas:

Planejamento do caminho – denota o caminho geométrico dos pontos no espaço das juntas, ou no espaço operacional, que o manipulador deve seguir na execução do movimento. Nesse caso, o caminho

é uma descrição puramente geométrica do movimento sem a informação tempo.

Planejamento da trajetória – é o caminho associado a uma lei temporal que define as velocidades e/ou acelerações em cada ponto do caminho geométrico especificado.

O processo de planejamento de trajetórias foi estudado por Tonetto (2007), para operações de recobrimento de superfícies. A sistematização consistiu em obter as informações da superfície (em forma de um conjunto de pontos) em que será executada a tarefa. A partir destes dados a superfície é reconstruída e gera-se o caminho a ser percorrido pelo manipulador na execução da tarefa. A Figura 5 mostra as etapas do processo completo de recobrimento de superfícies.

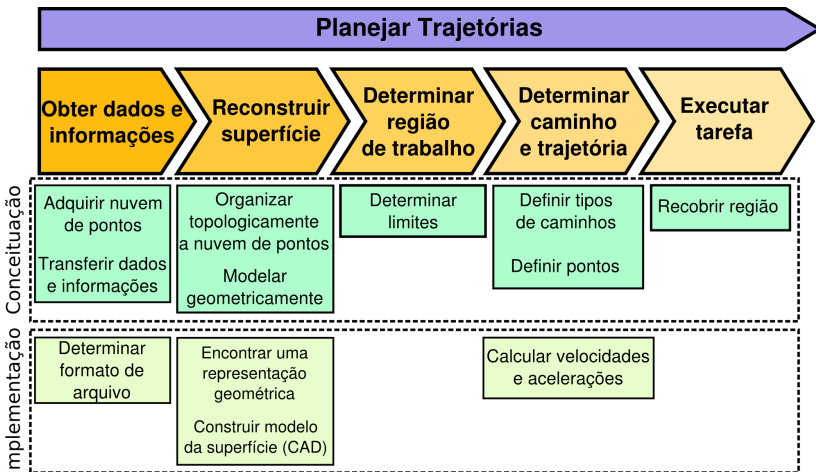


Figura 5: Descrição do processo de planejamento de trajetórias robóticas.

O trabalho de Tonetto (2007) utiliza a técnica de reconstrução de superfícies a partir de pontos. Assim, dada a descrição da superfície do objeto pode-se trabalhar no planejamento de tarefas robóticas. A abordagem utilizada no planejamento de trajetórias consiste na definição da trajetória no espaço operacional, indicando o caminho a ser seguido pela ferramenta sobre a superfície. Desta forma, o processo completo de planejamento de trajetórias é definido em cinco etapas: obter dados e informações sobre a superfície, reconstruir a superfície,

determinar a região de trabalho, determinar o caminho e a trajetória e por fim, executar a tarefa.

Simas, Dias e Guenther (2008) trabalharam com uma metodologia para planejar caminhos paralelos para recobrimento de superfícies baseado no critério *iso-scallop*. A metodologia inclui a seleção de superfícies matemáticas (para representar uma nuvem de pontos), o cálculo da região erodida e um algoritmo para gerar caminhos para o robô. A metodologia foi testada experimentalmente com o manipulador Roboturb (ROBOTURB, 2006).

2.4 TRABALHOS RELACIONADOS A SRC

Um dos primeiros trabalhos a comentar sobre dois manipuladores dividindo o mesmo espaço de trabalho e voltado ao planejamento de trajetórias é de Marushima, Yamamoto e Mohri (1991). Eles usam as equações dinâmicas dos manipuladores (em função de variáveis de juntas) e adicionam restrições que evitam o problema de colisão entre os manipuladores.

O problema trabalhado refere-se à busca de uma trajetória ótima das juntas que minimize o tempo de percurso do efetuador considerando os limites de torque/força e as condições de fronteira do caminho a ser percorrido (posições e velocidades inicial e final das juntas). Tem-se como restrição que a intersecção do espaço de trabalho ocupado pelos manipuladores deve ser vazia em um instante de tempo t . O evitamento de colisão é realizado fornecendo o que eles chamam de movimento prioritário aos dois manipuladores, onde, um dos manipuladores é considerado mestre e o outro escravo.

O manipulador mestre se move ao longo de uma trajetória já calculada de tempo mínimo e a trajetória do manipulador escravo é modificada satisfazendo as restrições de torque/força, fronteira e evitamento de colisão. A proposta deles, para evitar a colisão, é basicamente que o manipulador escravo só inicie o movimento depois que o robô mestre iniciar. Trabalhos subsequentes de Marushima, Yamamoto e Mohri e mais detalhes sobre a metodologia podem ser encontrados em (MOHRI; YAMAMOTO; MARUSHIMA, 1993, 1996).

Tzafestas, Prokopiou e Tzafestas (1998) trabalham com um sistema composto por três manipuladores idênticos em cooperação. O objetivo da tarefa é mover um objeto de grandes dimensões (objeto planar – como uma chapa metálica), de uma posição/orientação inicial para a posição/orientação desejada, segurando-o em três pontos dife-

rentes, de forma que esses três pontos definam um triângulo isósceles. Nesse trabalho foi adotado o método de operação dos robôs como um sendo mestre e os outros dois como escravos.

O movimento do objeto é definido no espaço 3D através de uma matriz de transformação homogênea variante no tempo que determina os deslocamentos lineares e angulares de uma posição/orientação inicial para uma posição/orientação final. Em seguida são determinados os deslocamentos diferenciais dos efetuadores dos três manipuladores que compõem o sistema. Por fim, o controlador emprega o deslocamento diferencial para calcular os deslocamentos diferenciais das juntas dos robôs.

Alguns autores trabalham com dois robôs e consideram principalmente a conexão com o problema da dinâmica, o controle de força, como por exemplo, Kwon e Lee (1998) que pesquisam sobre a distribuição de força ótima entre robôs cooperativos. Yim, Selvarajan e Wells (1999) também pesquisam o controle cooperativo de braços duais na qual manipulam um objeto com a presença de incertezas nos parâmetros dinâmicos entre os robôs e os objetos envolvidos no sistema.

Um outro exemplo de aplicação de vários robôs na execução de uma tarefa é o trabalho de Jutard-Malinge e Bessonnet (2000), o objetivo é otimizar a transferência repetida de objetos de um manipulador para outro.

Um trabalho de pesquisa envolvendo o evitamento de colisão entre os elos de dois manipuladores participantes da tarefa é o de Ju, Liu e Hwang (2002), onde é apresentada uma estratégia que consiste em alterar a velocidade do efetuador em tempo real quando for verificada a proximidade dos elos dos robôs (que são representados por elipsóides). Dessa forma, através da proximidade entre os elipsóides é estimada a distância entre os elos e alterada a velocidade do efetuador quando existir tendência à colisão. Os robôs são tratados como mestre e escravo e as trajetórias dos efetuadores são planejadas independentemente.

Owen, Croft e Benhabib (2003, 2008) trabalham com dois manipuladores planares (um robô posicionador e outro robô operador) executando uma tarefa de usinagem. Eles determinam trajetórias ótimas para os dois manipuladores coordenados fornecendo uma trajetória que descreve o movimento desejado da ferramenta relativo ao robô posicionador. Para relacionar o movimento do caminho da ferramenta desejado com relação ao efetuador é utilizado o Jacobiano relativo. É aplicado o método da pseudo-inversa para calcular as velocidades das juntas que minimizem uma função custo apropriada no espaço nulo do Jacobiano relativo. A configuração factível é considerada a principal

componente da função custo, além disso, eles pretendem que com a contribuição das juntas, as forças durante a usinagem sejam mínimas, melhorando a precisão do processo de usinagem.

Yamano et al. (2004) focam no controle cooperativo de braços duais flexíveis para manipulação de objetos no espaço 3D, o controlador proposto integra o controle híbrido de posição/força e o controle de supervisão de vibração para os elos dos robôs.

Em Chuang, Lin e Chou (2006), a arquitetura mestre-escravo é usada com o objetivo de coordenar o movimento entre dois manipuladores. O método proposto utiliza a teoria de campo potencial para avaliar a repulsão entre os manipuladores e um obstáculo do ambiente, garantindo o evitamento de colisão para os caminhos a serem percorridos pelos robôs. Um caminho para o manipulador mestre é planejado, considerando o robô escravo como estático e então, um caminho para o manipulador escravo é planejado considerando o manipulador mestre como estático. Assim, os robôs são tratados como obstáculos em tempos diferentes e deve-se recalcular os parâmetros do ambiente para o tempo total.

Navarro-Alarcon et al. (2009) propõem um modelo de controle de força para tarefas de manipulação cooperativa com braços duais redundantes. Já, Tavasoli, Eghtesad e Jafarian (2009) trabalham com o planejamento de trajetória de objetos flexíveis, no qual utilizam dois manipuladores planares para realizar o movimento.

Os trabalhos descritos acima são resolvidos para casos particulares envolvendo no máximo três robôs para tarefas específicas, como mover um determinado objeto de uma posição inicial para uma posição final ou robôs planares realizando uma tarefa de usinagem. O ideal é que o planejamento e a programação fossem feitos não somente para casos específicos, mas que fosse possível programar vários robôs diferentes executando qualquer tarefa. Alguns trabalhos como os citados a seguir são focados no desenvolvimento e aplicação de robôs executando tarefas diversas, como robôs posicionando as peças e outros efetuando operações sobre elas ou tarefas de manipulação e deslocamentos de objetos.

Dourado (2005) trabalha com a cinemática inversa diferencial de robôs em cooperação executando o planejamento de trajetórias e simulações com dois e três robôs planares. As tarefas realizadas num mesmo sistema robótico podem ser soldagem, inspeção, compartilhamento de carga. A metodologia de Dourado (2005) pode ser estendida para o desenvolvimento de trabalhos que incluam o planejamento de trajetórias no espaço 3D com vários robôs trabalhando em cooperação

(DOURADO; MARTINS, 2005).

Com relação a SRC, Ribeiro (2010) propôs o Jacobiano de cooperação utilizando o conceito de Jacobiano relativo para dois manipuladores introduzido por Lewis (1996). O Jacobiano relativo é obtido pela diferenciação da equação da posição de dois robôs operando em cooperação cinemática, seguindo a convenção de Denavit-Hartenberg (LEWIS; MACIEJEWSKI, 1990; LEWIS, 1996). Desta forma, Ribeiro (2010) definiu o Jacobiano de cooperação baseado em helicoides, que relaciona as velocidades da ferramenta (fixa em um manipulador), relativo a peça (fixa em outro manipulador), como uma função das velocidades das juntas dos manipuladores, usando a representação por helicoides para a cinemática diferencial. O método apresentado por Ribeiro (2010) é apropriado para qualquer geometria de robôs, generalizado para qualquer número de robôs participantes do sistema e com robôs possuindo qualquer número de juntas (RIBEIRO; GUENTHER; MARTINS, 2008), (RIBEIRO; MARTINS, 2010) e (RIBEIRO, 2010).

A Tabela 1 destaca as principais características dos trabalhos comentados anteriormente sobre SRC.

A proposta de Simas (2008) não contempla um relação direta com a cooperação de robôs. No entanto, surge uma série de questões e características adicionais que também são válidas em SRC. Simas (2008) desenvolveu uma sistemática para o planejamento de tarefas fora de linha e o estudo da cinemática inversa de posição, para mecanismos de cadeia fechada, usando o modelo da cinemática diferencial. O modelo da cinemática diferencial é obtido utilizando uma proposta de modelagem de mecanismo de cadeia fechada baseada no método de Davies e nas cadeias virtuais de Assur. O trabalho de Simas (2008) trata de um caso específico para um único robô redundante executando tarefas de recobrimento. Neste caso, o robô é serial e são adicionadas cadeias virtuais para o fechamento da cadeia cinemática e consequentemente a resolução da cinemática inversa.

A abordagem da redundância é também importante em SRC. A adição de robôs em SRC acrescenta mobilidade no sistema, tornando-o redundante e melhorando a flexibilidade na execução de tarefas (as definições de redundância e mobilidade são comentadas no Apêndice E). Assim, por se tratar de sistemas redundantes com um único robô, a metodologia de Simas (2008) pode ser estendida para o cálculo da cinemática de SRC.

Além de serem utilizadas para o fechamento da cadeia, as cadeias virtuais também podem ser aplicadas para incluir características como por exemplo, evitamento de colisão (SIMAS et al., 2008; SIMAS, 2008;

Tabela 1: Características dos principais trabalhos relacionados a SRC.

Autor	N. de robôs	Observações
Marushima, Yamamoto e Mohri (1991)	2 planares	planejamento trajetórias
Lewis (1996)	2 espaciais	planejamento trajetórias; Jacobiano relativo
Tzafestas, Prokopiou e Tzafestas (1998)	3 espaciais	mover objeto grande; cinemática de posição
Kwon e Lee (1998)	2 espaciais	distribuição de força; usa o Jacobiano para o cálculo da cinemática
Yim, Selvarajan e Wells (1999)	braços duais	controle de força/posição; usa o Jacobiano para o cálculo da cinemática
Ju, Liu e Hwang (2002)	2 espaciais	evitamento de colisão (elipsóides); não comenta sobre o cálculo da cinemática
Owen, Croft e Benhabib (2003)	2 planares	planejamento de trajetórias; Jacobiano relativo
Yamano et al. (2004)	braços duais	controle de força/posição; usa o Jacobiano para o cálculo da cinemática
Dourado (2005)	2 e 3 planares	planejamento de trajetórias; cinemática inversa por helicoides
Chuang, Lin e Chou (2006)	2 espaciais	evitamento de colisão (campo potencial); não comenta sobre de cálculo da cinemática
Ribeiro (2010)	n espaciais	conceito Jacobiano de cooperação; Jacobiano de cooperação por Denavit-Hartenberg e por helicoides

FONTAN, 2007) e cadeia virtual de erro (SIMAS, 2008; GUENTHER et al., 2008).

Guenther et al. (2008) introduz o conceito de cadeias virtuais de erro, minimizando os problemas numéricos resultantes dos métodos de integração. Como poderá ser visto no Capítulo 5, estes conceitos são

estendidos como características adicionais em SRC. Já que, assim, como existem situações que envolvem evitar colisão e a abertura da cadeia cinemática devido ao método de integração em sistemas com um único robô, existem também questões semelhantes em SRC e que precisam ser consideradas em algum momento da programação.

Nesta tese, cada uma dessas características é integrada em um SRC de forma generalizada. Sua tarefa e o conceito das características adicionais é apresentado de forma a possibilitar extensões futuras.

Esta tese está inserida no contexto de pesquisa do Laboratório de Robótica Raul Guenther - UFSC, que vem realizando estudos na área de robôs cooperativos (DOURADO, 2005; DOURADO; MARTINS, 2005; RIBEIRO, 2010; RIBEIRO; MARTINS, 2010), sobre a teoria de helicoides, o método de Davies e suas aplicações (MARTINS, 2002; CAMPOS, 2004; CAMPOS; GUENTHER; MARTINS, 2005; SANTOS, 2006; SANTOS et al., 2006; FONTAN, 2007; SIMAS et al., 2008; SIMAS, 2008; SIMAS et al., 2009; SIMONI, 2008; CARBONI, 2008; CAZANGI, 2008).

2.5 CONCLUSÕES

Entre os trabalhos relacionados a SRC, nota-se que somente o trabalho de Ribeiro (2010) trata o problema para um número arbitrário de robôs, enquanto os demais limitam em dois ou três robôs. Quanto aos métodos para o cálculo da cinemática de manipuladores, o uso do Jacobiano é bastante comum. Outros tópicos de estudo são ainda encontrados, como o evitamento de colisão e controle de força.

Baseado nos trabalhos comentados anteriormente, observa-se que falta sistematizar a modelagem cinemática e a programação de SRC. O trabalho do Ribeiro (2010) trata esse tópico de maneira suave, pois seu foco principal é o Jacobiano de cooperação. Ele também mostra que o resultado é o mesmo deduzindo o Jacobiano de cooperação através de Denavit-Hartenberg, helicoides e pelo método de Davies. Mas cabe ressaltar que o trabalho de Ribeiro (2010) não aborda de forma completa o problema de programação de SRC, com foco nos robôs e nas tarefas. Para programar **tarefas** e **robôs** é necessário ir além do Jacobiano de cooperação, precisa-se considerar todos os passos para se determinar a cinemática, com o objetivo de integrar os dados dos robôs com os parâmetros das tarefas.

3 PROGRAMAÇÃO CINEMÁTICA DIFERENCIAL E TAREFAS ROBÓTICAS PARA SRC

Neste capítulo são apresentadas as etapas necessárias para a programação robótica, que devem ser cumpridas para que os robôs realizem a tarefa objetivo. São analisados três ambientes principais com o objetivo de organizar a programação de SRC: a estrutura robótica, o ambiente da tarefa e a cinemática diferencial. Com esses ambientes definidos, é detalhada a programação de um sistema robótico convencional com apenas um robô, enfatizando o fato de serem adicionadas duas cadeias virtuais. Em seguida, a metodologia é estendida para a programação de um SRC, analisando-se o crescimento do sistema, na medida em que são adicionados novos robôs e novas tarefas. Por fim, é descrita uma rotina para a programação de SRC, sintetizando a metodologia.

3.1 AMBIENTES PARA A PROGRAMAÇÃO ROBÓTICA

O objetivo da programação robótica é planejar os movimentos dos robôs de forma a executar as tarefas. Devido às dificuldades de utilização dos sistemas reais para a etapa de planejamento, a programação *off-line* torna-se indispensável. Como os sistemas físicos reais não são utilizados no planejamento, é necessário que exista uma forma de representação, estabelecendo os modelos dos objetos envolvidos: robôs e tarefas.

O que precisa ser feito na programação é encontrar quais são as ações a serem tomadas nos modelos dos robôs para que se cumpram os modelos desejados das tarefas. Existe assim um ambiente distinto que realiza esta função de aliar os robôs às tarefas. A presença de três ambientes principais, a estrutura robótica, as tarefas e a cinemática é encontrada em vários trabalhos na literatura aplicando-se à atividades com um único robô.

A divisão da programação nestes ambientes nem sempre é explícita. No entanto, devido as complexidades ao lidar com SRC, faz-se necessário a definição, o detalhamento e a separação destas estruturas. A partir destas estruturas é facilitada a criação dos métodos que viabilizam a programação *off-line* de SRC.

Os três ambientes principais são definidos para a programação de robôs no planejamento e execução de tarefas robóticas, conforme é ilustrado na Figura 6. O ambiente da **estrutura robótica** faz refe-

rência ao estado dos manipuladores, com as informações e as condições iniciais de funcionamento no espaço de juntas do robô. O segundo ambiente é o da **tarefa**, onde é feita uma descrição com base nos dados de referência da tarefa a ser executada. É definido qual a tarefa a ser desempenhada pelos robôs, assim como parâmetros de velocidades, acelerações, além do caminho e da trajetória a serem percorridas no espaço operacional. O terceiro ambiente, a **cinemática diferencial**, associa as trajetórias definidas no ambiente da tarefa com o modelo cinemático obtido a partir do ambiente estrutura robótica (TONETTO; ROCHA; DIAS, 2010).

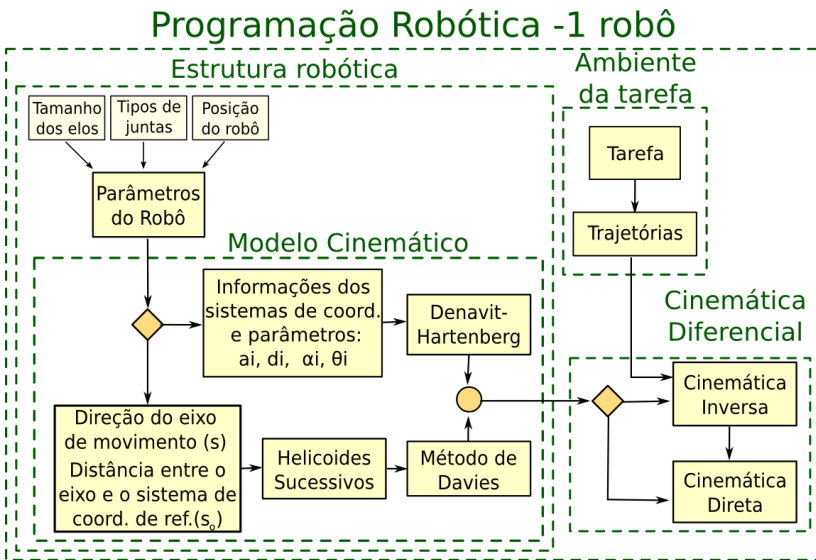


Figura 6: Fluxograma para o cálculo da cinemática de robôs.

No ambiente da estrutura robótica define-se qual o modelo cinemático é utilizado e esses modelos propriamente ditos são construídos a partir dos parâmetros do robô. Os parâmetros do robô compreendem um conjunto de informações como o comprimento e o número de elos, limites de velocidades, alcance do robô, tipos de juntas (como podem ser citadas: rotativa, prismática, cilíndrica, esférica, plana) e seus limites de deslocamento.

Devido ao processo de padronização, muitos robôs apresentam estruturas físicas semelhantes e conhecidas, sendo o caso de somente

existirem alterações nos tamanhos dos elos. Para estes casos, pode-se utilizar modelos já definidos e arquivados em uma base de dados, sendo estes modelos parametrizados pelos tamanhos dos elos. Alguns exemplos de estruturas padronizadas em manipuladores comerciais são ilustrados na Figura 7, como o Scara e o Antropomórfico. Esses robôs são manipuladores que possuem propriedades bem estabelecidas. Por exemplo, os robôs Scara tem em sua estrutura física duas juntas rotativas com eixos paralelos, e em seguida uma junta prismática com deslocamento na direção dos eixos das juntas rotativas.

Um outro dado relevante para o modelo cinemático é a posição da base do robô com relação a um sistema de coordenadas, que também é considerada como um dos parâmetros do robô. Assim, os parâmetros do robô compõem o conjunto de dados físicos fundamentais para se definir o modelo cinemático do robô.

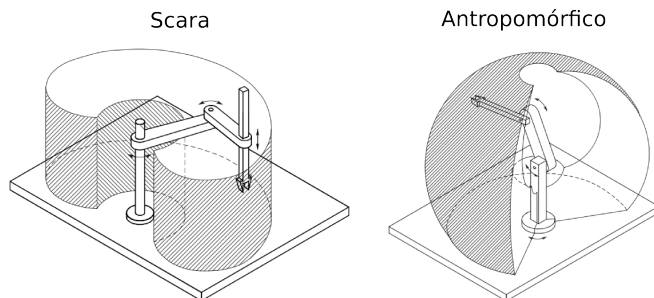


Figura 7: Manipuladores – Scara e Antropomórfico (SICILIANO et al., 2009).

O modelo cinemático relaciona as variáveis do robô com o movimento físico do efetuator e dos elos do robô. Essas variáveis são relativas às juntas e parametrizadas pela estrutura dos elos.

No modelo cinemático é definido qual é o relacionamento entre os elos e as juntas do robô. O conjunto de parâmetros do robô precisa ser organizado, utilizando alguma técnica de modelagem, de forma a se obter um relacionamento físico entre os elos e juntas do robô. Este relacionamento físico é então representado de maneira unificada e permite conhecer o comportamento dos movimentos do robô a partir da variação de seus parâmetros.

O modelo cinemático pode ser construído através de duas formas mais comuns – convenção de Denavit-Hartenberg e o método de helicoides:

- convenção de Denavit-Hartenberg, que é definida por quatro parâmetros (ver Figura 8):
 - a_i distância entre os sistemas de coordenadas fixo aos elos do robô (O_i e O'_i);
 - d_i direção do eixo de coordenada de O'_i ao longo de z_{i-1} ;
 - α_i ângulo entre os eixos z_{i-1} e z_i com relação ao eixo x_i ;
 - ϑ_i ângulo entre os eixos x_{i-1} e x_i com relação ao eixo z_{i-1} ;

Os parâmetros a_i e α_i são sempre constantes e dependem somente da geometria de conexão entre as juntas consecutivas estabelecida pelo elo i . Em particular, se a junta i é rotativa a variável é ϑ_i , se a junta i é prismática, o termo que varia é d_i (SICILIANO et al., 2009; SPONG; HUTCHINSON; VIDYASAGAR, 2006);

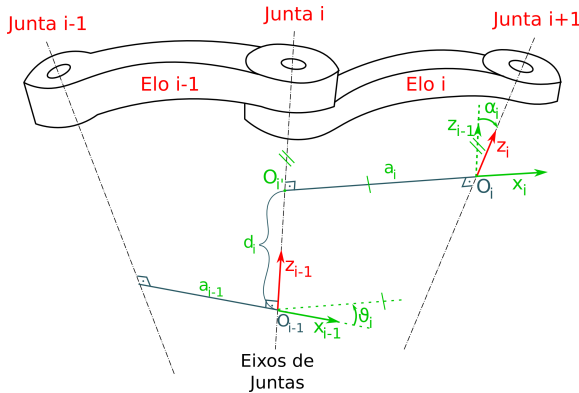


Figura 8: Parâmetros de Denavit-Hartenberg (SICILIANO et al., 2009).

- método de helicoides, em que é necessário estabelecer os valores de s e s_o (conforme ilustrado na Figura 9):
 - s é o vetor unitário que indica a direção do eixo de helicóide da junta i ;
 - s_o é o vetor posição de um ponto de s com relação a um sistema de coordenadas fixo da junta i .

Os valores de s e s_o são obtidos a partir dos parâmetros do robô (HUNT, 2000; CECCARELLI, 2000). Mais detalhes sobre a teoria de helicoides são apresentados no Apêndice A.

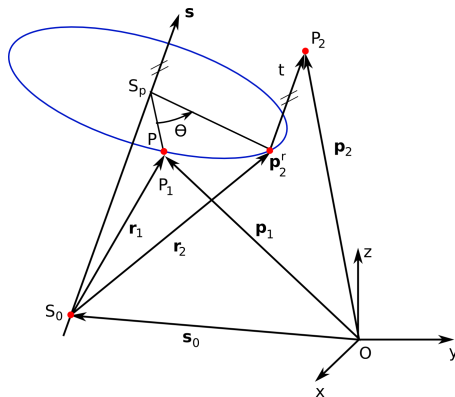


Figura 9: Deslocamento helicoidal de um ponto P (TSAI, 1999).

O segundo ambiente inclui a tarefa a ser realizada pelo robô e conseqüentemente as trajetórias que o robô deve percorrer. As trajetórias são descritas no espaço operacional com referência ao efetuador do robô. Alguns métodos de geração de caminhos foram discutidos na seção 2.3 do Capítulo 2.

Com a estrutura robótica contendo o modelo cinemático e o ambiente da tarefa definidos, passa-se ao terceiro ambiente, a cinemática diferencial, em que são determinadas as velocidades das juntas do robô para cada instante. O objetivo da programação é definir as posições e velocidades das juntas do robô (dados de saída da cinemática inversa) de forma a realizar a tarefa desejada. A execução correta da tarefa pode ser conferida pelo uso da cinemática direta nas posições das juntas definidas pela programação. Assim, a programação é afetada pela estrutura do robô, de onde são carregadas as informações necessárias para o cálculo da cinemática inversa, e pela descrição da tarefa.

Esses ambientes quando explorados possibilitam que o robô disponível execute a tarefa. Apesar desse ser um procedimento comum na literatura para um robô, a organização da estruturação proposta pode ser ampliada para múltiplos robôs.

3.2 AMBIENTES PARA A PROGRAMAÇÃO DE SRC

A abordagem apresentada para um robô pode ser estendida para SRC, pois, sem diferenciação alguma, existem tarefas a serem realizadas

e existem robôs para executá-las. No entanto, é necessário que algumas adaptações sejam feitas para aplicá-las a múltiplos robôs.

O SRC possui um conjunto de robôs e tarefas se relacionando e a dificuldade de programação aumenta se comparado a um robô com uma tarefa. De determinado ponto de vista, um sistema com múltiplos robôs apresenta uma flexibilidade maior com relação às escolhas de qual robô realizará determinado movimento. No entanto, por outro ponto de vista, a relação de interdependência entre os robôs cria novas restrições e assim, como se tem novas flexibilidades e restrições, se configura um sistema complexo quando comparado a um sistema com um robô e uma tarefa. Então, uma nova proposta, adaptada a SRC, é ilustrada na Figura 10.

A Figura 6 é estendida para SRC, sendo que neste caso o sistema contém um número arbitrário de robôs (m) e tarefas. Todos os dados das estruturas dos robôs precisam ser reunidos dentro de um modelo cinemático e então agrupados utilizando uma abordagem generalizada. Uma proposta de integrar essas informações é apresentada por Ribeiro, Guenther e Martins (2008), através da definição do Jacobiano de cooperação. As generalizações são realizadas de tal forma que a compatibilidade com a cinemática diferencial é mantida, o que significa que a teoria para o cálculo da cinemática direta e inversa para um robô pode ser aplicada para um SRC. Para efeitos práticos, pode-se considerar a estrutura robótica de um SRC como equivalente a um robô complexo, com todos os elos e juntas inseridos no sistema.

Desta forma, o processo de programação robótica tem como informações de entrada: os parâmetros de todos os robôs componentes do SRC e a descrição das tarefas. Como informação de saída, têm-se as trajetórias das juntas dos robôs (Figura 10). Essas trajetórias são suficientes para que o SRC execute a tarefa, de forma que a programação conforme proposta torna possível o planejamento da tarefa utilizando como informações somente as especificações dos robôs e da tarefa. Os métodos descritos ao longo desta tese tornam possível este processo automático de programação de SRC.

Independente dos robôs que irão executar as tarefas, são especificadas todas as tarefas que se deseja realizar na forma de trajetória relativa do sistema de coordenadas da peça para o efetuador e também do sistema de coordenadas fixo para o sistema de coordenadas da peça. Todo o processo de programação de SRC será descrito a seguir em detalhes.

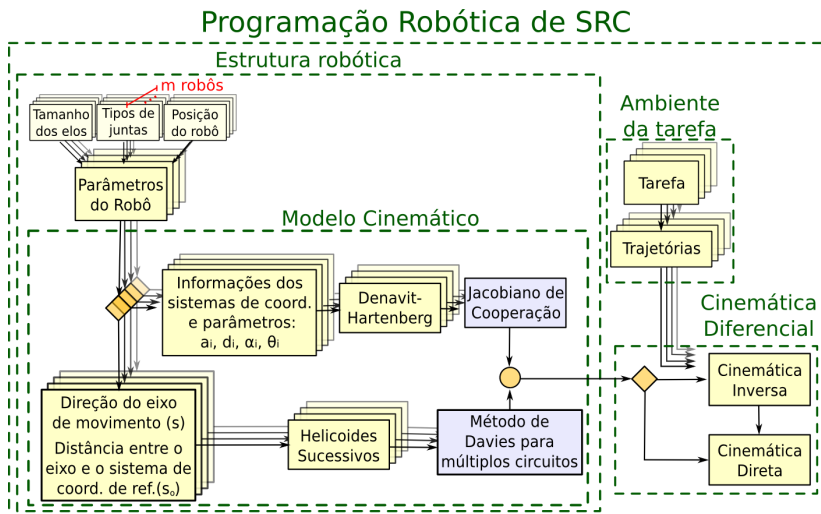


Figura 10: Fluxograma para o cálculo da cinemática de um SRC.

3.3 PROGRAMAÇÃO ROBÓTICA

Os ambientes de programação robótica foram apresentados nas seções 3.1 e 3.2. Eles servem como base para ilustrar a programação de um robô e de SRC. Tanto para um robô quanto para SRC, os estágios de programação robótica são praticamente os mesmos, ou seja, em ambos precisa-se das informações do(s) robô(s), do modelo cinemático, das tarefas e do cálculo da cinemática diferencial. A principal diferença está em como fazer o agrupamento de todos esses ambientes para manter a relação existente entre os robôs em sistemas cooperativos. Pelo que foi apresentado anteriormente, esses agrupamentos são realizados, através de, por exemplo, Jacobiano de cooperação ou método de Davies. Nesta seção será descrito o processo de programação de um robô e em seguida de SRC, com o objetivo de evidenciar a passagem da programação de um sistema com um robô para sistemas com mais robôs.

3.3.1 Programação robótica simples

Para entender a metodologia que será desenvolvida para SRC, é detalhada nesta subseção uma abordagem para o cálculo da cinemática

para um robô. Apesar de muitos autores abordarem a programação de um robô por métodos baseados em Denavit-Hartenberg ou teoria de helicoides, a representação que será introduzida visa organizar os diferentes elementos do sistema para possibilitar a generalização do processo para múltiplos robôs. A metodologia segue os ambientes comentados na seção 3.1

Configuração da cadeia cinemática

A descrição do processo de programação robótica será iniciada pela especificação da estrutura robótica (ver a Figura 6), cujo o resultado servirá de insumo para a cinemática diferencial. Dentro desta caracterização o resultado é a matriz de rede, obtida para aplicação do método de Davies, que indica as relações entre os elementos do sistema.

Faz parte da estrutura robótica um robô (R_1), cujo os parâmetros são obtidos, e duas cadeias virtuais¹ adicionais, denominadas CV_0 e CV_1 , para auxiliar na determinação do modelo cinemático, conforme ilustrado na Figura 11. A cadeia virtual CV_0 relaciona o movimento/posição relativo da peça com o sistema de coordenadas fixo (base) e a cadeia virtual CV_1 relaciona o movimento do efetuador sobre a peça. Assim, as cadeias virtuais CV_0 e CV_1 representam os movimentos relativos entre a peça e a base, e entre o efetuador do robô e a peça. De fato, o movimento pode ser causado por imposição de movimento na peça (CV_0) ou pela ação do próprio robô (CV_1). Esses movimentos serão especificados nas trajetórias definidas através das tarefas.

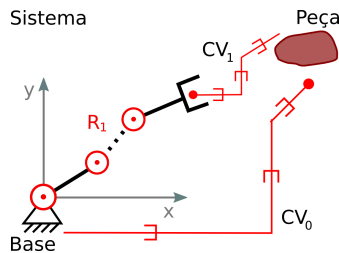


Figura 11: Sistema robótico composto por um robô.

Desta forma, considerando o sistema com um robô, ambas as situações podem ocorrer, então, CV_0 e CV_1 representam esses movimen-

¹Informações sobre as cadeias virtuais de Assur podem ser obtidas no Apêndice B.3.

tos. Porém, em alguns casos, existirá somente o movimento exercido pela ação do próprio robô sobre a peça (robô faz um movimento sobre a peça), que caberá apenas a CV_1 representar o movimento. Caso a tarefa, seja, por exemplo, movimentar a peça, então a CV_0 representará o movimento. Conclui-se, que independente do tipo de movimento/tarefa, as cadeias virtuais CV_0 e CV_1 podem representá-lo. Este é um detalhe muito importante e explica o fato de serem adicionadas duas cadeias virtuais em um sistema com um único robô.

A percepção desse detalhe foi fundamental para o desenvolvimento da metodologia de programação de SRC abordada nesta tese. Os trabalhos anteriores não discriminavam o uso de cadeias virtuais para tarefas como sendo uma para movimentação da peça com relação a base e outra para relacionar o efetuador do robô com a peça. A adição das cadeias virtuais era realizada conforme o objetivo a ser atingido pelo robô. Se fosse necessário mover o objeto de uma posição a outra, então uma cadeia virtual de movimento da peça com relação a base era adicionada, aqui definida como CV_0 . Se o objetivo era realizar uma operação sobre a peça, então uma cadeia virtual que relacionava o efetuador com a peça era adicionada, definida aqui como CV_1 . Nos trabalhos anteriores, esta atribuição de movimentos relativos não era comentada, nem a possibilidade de serem adicionadas duas cadeias virtuais em um sistema com um único robô.

Assim, ao se adicionar duas cadeias virtuais, fica estabelecido que existem dois componentes que podem ter movimentos relativos no sistema e as cadeias virtuais entre eles. Nos trabalhos anteriores, a estrutura da cadeia cinemática dependia do objetivo a ser atingido pelo robô. Na nova formulação, a estrutura não depende do objetivo, mas é flexível e permite representar os diferentes tipos de tarefas a serem atribuídas ao sistema.

A utilização de cadeias virtuais que relacionam os movimentos relativos entre pontos convenientes em uma cadeia cinemática é uma importante contribuição introduzida neste trabalho, pois permite representar um conjunto maior de tarefas e de movimentos da peça e do efetuador, além de possibilitar a aplicação de novas tarefas aos sistemas. As vantagens da nova abordagem são ainda mais enfatizadas quando da aplicação em SRC, como será visto na subseção 3.3.2.

Com a adição destas duas cadeias virtuais, o sistema é tratado como fechado, o que torna viável a aplicação do método de Davies. A seguir, tem-se o detalhamento do modelo cinemático, tomando a configuração da cadeia cinemática estabelecida anteriormente. A forma como é especificada a tarefa e o cálculo da cinemática diferencial tam-

bém são descritos.

Cálculo da cinemática

O modelo cinemático (Figura 6) para um robô com n juntas é estabelecido utilizando a teoria de helicoides e o método de Davies². A abordagem por helicoides inclui a definição dos eixos de helicoides (s) e a sua distância com relação a um sistema de referência (s_0). A partir dos eixos e das distâncias são gerados os helicoides de cada uma das juntas, que, por fim, são agrupadas aplicando o método de Davies.

Uma ferramenta para programação e análise de sistemas robóticos que vem sendo usada é a teoria de grafos³. O grafo que representa o sistema para um robô é ilustrado na Figura 12. No lado esquerdo tem-se o grafo que representa a cadeia cinemática do robô R_1 , a cadeia virtual CV_1 ligando o efetuator a peça e a cadeia virtual CV_0 que une o sistema de coordenadas fixo a peça. No lado direito é representado o mesmo grafo, porém de forma simplificada, sem a representação dos vértices componentes do robô e das cadeias virtuais no grafo.

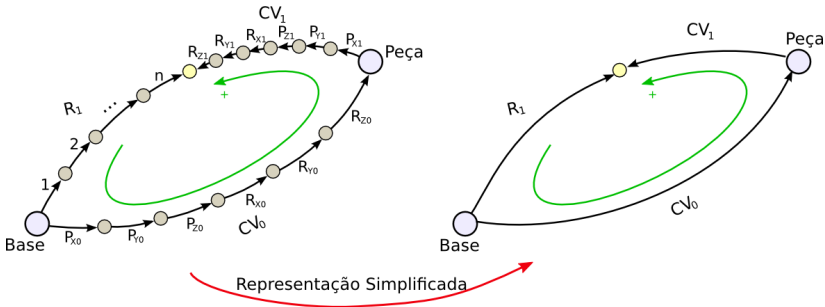


Figura 12: Representação por grafos do sistema composto por um robô e duas cadeias virtuais.

A matriz de circuitos B indica a presença de arestas em cada malha do grafo, e neste caso ela é dada por:

²Detalhes da teoria de helicoides e do método de Davies podem ser visualizados nos Apêndices A e B.2.

³A teoria de grafos e a matriz de circuitos é apresentada resumidamente no Apêndice C.

$$B = \begin{bmatrix} CV_0 & CV_1 & R_1 \\ [1]_{1 \times 6} & [1]_{1 \times 6} & [-1]_{1 \times n} \end{bmatrix} \quad (3.1)$$

A matriz B representa um grafo que possui uma malha e, consequentemente, ela possui uma linha. Para este sistema foram adicionadas cadeias virtuais espaciais, ou seja, com seis graus de liberdade e assim, a ordem do sistema de helicoides para esse caso é $\lambda = 6$. Ao longo deste trabalho foram abordados somente casos espaciais, sendo esses mais gerais. Na matriz descrita na Equação 3.1 as colunas CV_0 , CV_1 e R_1 foram escritas de forma compacta. Por exemplo, a coluna CV_0 expressa todas as juntas componentes da cadeia virtual CV_0 , na seguinte ordem: P_{X0} , P_{Y0} , P_{Z0} , R_{X0} , R_{Y0} e R_{Z0} .

A matriz de helicoides D que contém os helicoides normalizados (CAMPOS, 2004) é dada por:

$$D = \begin{bmatrix} \hat{\$}_{P_{X0}} & \hat{\$}_{P_{Y0}} & \hat{\$}_{P_{Z0}} & \hat{\$}_{R_{X0}} & \hat{\$}_{R_{Y0}} & \hat{\$}_{R_{Z0}} & \hat{\$}_{R_{Z1}} & \hat{\$}_{R_{Y1}} \\ \hat{\$}_{R_{X1}} & \hat{\$}_{P_{Z1}} & \hat{\$}_{P_{Y1}} & \hat{\$}_{P_{X1}} & \hat{\$}_n & \cdots & \hat{\$}_2 & \hat{\$}_1 \end{bmatrix}_{6 \times (n+12)} \quad (3.2)$$

Reescrevendo a matriz D também de forma compacta, tem-se:

$$D = [[\hat{\$}_{CV_0}]_{6 \times 6} \quad [\hat{\$}_{CV_1}]_{6 \times 6} \quad [\hat{\$}_{R_1}]_{6 \times n}] \quad (3.3)$$

em que $[\hat{\$}_{CV_0}]_{6 \times 6}$ representa a matriz que contém todos os helicoides da cadeia virtual CV_0 , $[\hat{\$}_{CV_1}]_{6 \times 6}$ é a matriz contendo os helicoides da cadeia virtual CV_1 e $[\hat{\$}_{R_1}]_{6 \times n}$ é a matriz que contém os helicoides do robô R_1 com n juntas.

Cada helicoide normalizado é associado a uma junta (real ou virtual), e são calculados em função do eixo de helicoide s e da sua distância s_o com relação a um sistema de coordenadas fixo. O movimento de uma junta é dependente do movimento da junta anterior e dessa forma, para considerar a influência que uma junta exerce sobre a outra, faz-se necessário aplicar os métodos dos helicoides sucessivos comentados no Apêndice B.1

A matriz de rede N da equação de restrição é obtida multiplicando-se a matriz de helicoides D pela matriz diagonal B :

$$N = D \text{diag}\{B\} \quad (3.4)$$

Logo, a matriz de rede N é:

$$N = [[\hat{\$}_{CV_0}]_{6 \times 6} \quad [\hat{\$}_{CV_1}]_{6 \times 6} \quad [-\hat{\$}_{R_1}]_{6 \times n}] \quad (3.5)$$

Esta matriz é o resultado da estrutura robótica e é utilizada para calcular a cinemática diferencial, relacionando os elementos do sistema (robô e tarefa).

Pelo método de Davies, a matriz de rede N pode ser decomposta em submatrizes contendo os helicoides referentes às juntas primárias N_p e nas submatrizes contendo os helicoides referentes às juntas secundárias N_s (no Apêndice B.2 tem-se as definições de juntas primárias e juntas secundárias):

$$\begin{aligned} N_p &= [[\hat{\$}_{CV_0}]_{6 \times 6} \quad [\hat{\$}_{CV_1}]_{6 \times 6}] \\ N_s &= [[-\hat{\$}_{R_1}]_{6 \times n}] \end{aligned} \quad (3.6)$$

A definição das matrizes N_p e N_s , contendo os helicoides normalizados e as magnitudes das velocidades correspondentes são os dados utilizados pela cinemática diferencial nos procedimentos de cinemática direta e inversa.

Na cinemática inversa, a velocidade das juntas secundárias \dot{q}_s do robô é dada pela equação:

$$\dot{q}_s = -[N_s]^{-1}[N_p][\dot{q}_p] \quad (3.7)$$

onde \dot{q}_p é o vetor de velocidade referente as juntas primárias, especificadas pelas juntas pertencentes as cadeias virtuais. E então, neste caso, o vetor \dot{q}_p é:

$$\dot{q}_p = \begin{bmatrix} \dot{q}_{CV_0} \\ \dot{q}_{CV_1} \end{bmatrix} \quad (3.8)$$

em que \dot{q}_{CV_0} é a velocidade da peça com relação ao sistema de coordenadas fixo (caso exista movimento relativo entre a peça e o sistema de coordenadas fixo) e \dot{q}_{CV_1} é a velocidade desejada do efetuador com relação à peça. Os vetores de velocidade são os resultados das trajetórias especificadas no ambiente da tarefa.

Desta forma, foi apresentado o desenvolvimento para o cálculo das velocidades das juntas do robô conhecidas as velocidades do efetuador. A técnica é estendida para SRC, composto por m robôs e $m + 1$ cadeias virtuais.

3.3.2 Programação robótica de SRC

Configuração da cadeia cinemática

A estrutura robótica para SRC (Figura 10) é generalizada a partir da estrutura caracterizada para um sistema simples. Da mesma forma que em um sistema simples, o resultado da estrutura robótica é insumo para a cinemática diferencial e dentro desta proposta também é constituído por uma matriz de rede, gerada pelo método de Davies, que inclui as estruturas de múltiplos robôs.

Ao se adicionar um robô em um sistema, pode-se desenvolver uma sistemática para a modelagem genérica das cadeias virtuais que representam um SRC. A Figura 13 ilustra os grafos que representam diversos SRCs, com 1, 2, 3, ..., m robôs. Nela é possível observar como é a estrutura de um SRC conforme o número de robôs existentes no sistema. Assim, baseado no fluxograma apresentado no início deste capítulo, nas seções 3.1 e 3.2, em que foram destacados três ambientes principais para a programação de um SRC, é agora sintetizada a estrutura da cadeia cinemática de SRC.

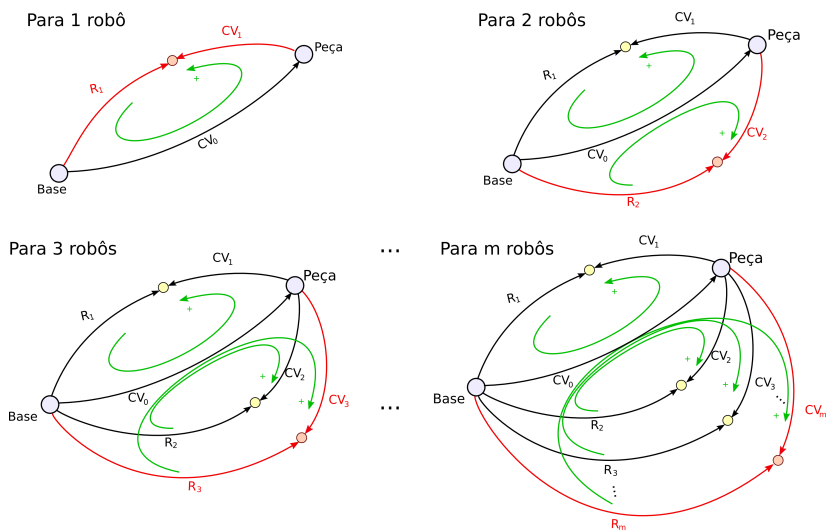


Figura 13: Grafos que representam SRCs: com 1, 2, 3, ..., m robôs.

Em um sistema pode existir o deslocamento da peça e uma tarefa sendo executada sobre ela, a peça é unida à base por uma cadeia virtual CV_0 . Esta cadeia virtual representa o movimento imposto à peça. A tarefa executada sobre a peça é representada pela cadeia virtual CV_1 . Assim, o grafo que representa esse sistema:

- para um robô (executando a tarefa especificada) é composto das seguintes cadeias cinemáticas:
 - $CV_0 \rightarrow$ cadeia virtual que une a base à peça;
 - $R_1 \rightarrow$ cadeia cinemática que representa o robô 1;
 - $CV_1 \rightarrow$ cadeia virtual que une o efetuador do robô à peça.

O sistema formado por um robô é representado por um grafo que contém uma malha.

- supondo que é adicionado mais um robô ao sistema, ou seja, o sistema é composto por dois robôs e então a sua representação por grafos passa a apresentar:
 - $CV_0 \rightarrow$ cadeia virtual que une a base à peça;
 - $R_1 \rightarrow$ cadeia cinemática que representa o robô 1;
 - $CV_1 \rightarrow$ cadeia virtual que une o efetuador do robô 1 à peça;
 - $R_2 \rightarrow$ cadeia cinemática que representa o robô 2;
 - $CV_2 \rightarrow$ cadeia virtual que une o efetuador do robô 2 à peça.

Nesse sistema, o grafo que o representa possui duas malhas, sendo que a CV_0 é comum às duas. Pode-se dizer que as duas malhas são semelhantes, pois cada uma contém a cadeia virtual CV_0 , a cadeia cinemática do robô R_i e a cadeia virtual CV_i que une o efetuador do robô à peça, com $i = \{1, 2\}$.

- Adicionando agora o terceiro robô ao sistema, tem-se:
 - $CV_0 \rightarrow$ cadeia virtual que une a base à peça;
 - $R_1 \rightarrow$ cadeia cinemática que representa o robô 1;
 - $CV_1 \rightarrow$ cadeia virtual que une o efetuador do robô 1 à peça;
 - $R_2 \rightarrow$ cadeia cinemática que representa o robô 2;
 - $CV_2 \rightarrow$ cadeia virtual que une o efetuador do robô 2 à peça;
 - $R_3 \rightarrow$ cadeia cinemática que representa o robô 3;

- $CV_3 \rightarrow$ cadeia virtual que une o efetuador do robô 3 à peça.

Para este caso, o grafo possui três malhas, em que, a terceira malha é formada pela adição ao sistema de mais um robô R_3 e uma cadeia virtual CV_3 . Observe que este grafo é uma evolução do grafo do caso anterior (com dois robôs). Cada malha continua apresentando a mesma estrutura CV_0 , R_i e CV_i , com $i = \{1, 2, 3\}$.

– Considerando agora, que o sistema é composto por m robôs, tem-se:

- $CV_0 \rightarrow$ cadeia virtual que une a base à peça;
- $R_1 \rightarrow$ cadeia cinemática que representa o robô 1;
- $CV_1 \rightarrow$ cadeia virtual que une o efetuador do robô 1 à peça;
- $R_2 \rightarrow$ cadeia cinemática que representa o robô 2;
- $CV_2 \rightarrow$ cadeia virtual que une o efetuador do robô 2 à peça;
- $R_3 \rightarrow$ cadeia cinemática que representa o robô 3;
- $CV_3 \rightarrow$ cadeia virtual que une o efetuador do robô 3 à peça;
- $\quad \quad \quad \vdots \quad \quad \quad \vdots$
- $R_m \rightarrow$ cadeia cinemática que representa o robô m ;
- $CV_m \rightarrow$ cadeia virtual que une o efetuador do robô m à peça.

Assim, duas características importantes são conferidas nessa sistemática de modelagem genérica. Uma diz respeito a cada malha do sistema, ou seja, se o sistema tiver m robôs, este passa a apresentar m malhas e cada uma delas é composta pela mesma estrutura: CV_0 , R_i e CV_i , em que $i = 1, 2, \dots, m$.

A segunda característica é que, conforme são adicionados os robôs ao sistema, o sistema anterior está incluído no sistema atual, mas com uma malha a mais referente ao robô adicionado. Assim, a representação por grafos do sistema anterior passa a estar contida no sistema atual. Desta forma, quando um sistema for composto por m robôs, é possível utilizar a estrutura por grafos com $m - 1$ robôs, apenas adicionando mais um robô e uma cadeia virtual. Portanto, independente do número de robôs no sistema, do número de juntas de cada robô e da estrutura de cada robô componente, a representação por grafos de um SRC é sempre a mesma, seguindo um padrão conforme ilustrada na Figura 13.

Desta forma, foi apresentada uma estrutura do sistema para a programação cinemática de um SRC. A partir dela, é desenvolvida uma

metodologia de cálculo da cinemática para SRC. A metodologia de cálculo se insere no ambiente de cinemática diferencial, mas como mostrado na estrutura, inclui as informações fornecidas pelos ambientes de estrutura robótica e da tarefa (Figura 10).

Cálculo da cinemática

Esta seção apresenta a metodologia para o cálculo da cinemática diferencial para um sistema composto por vários robôs. Assim, é assumido que o sistema possui m robôs e em seguida são determinadas as matrizes para aplicação do método de Davies.

Em um SRC composto por m robôs são adicionadas $m + 1$ cadeias virtuais (uma para cada robô e a cadeia de movimentação da peça). O grafo que representa um SRC possui arestas que ligam a peça ao sistema de coordenadas fixo por uma cadeia virtual (aqui denominada CV_0), enquanto cada robô (representado no grafo por R_i) é ligado ao sistema de coordenadas fixo e a uma cadeia virtual. Para cada robô existe uma cadeia virtual relacionando a posição de seu efetuator à peça (CV_i , com $i = 1, 2, \dots, m$). O sistema é representado na forma simplificada pelo grafo na Figura 14.

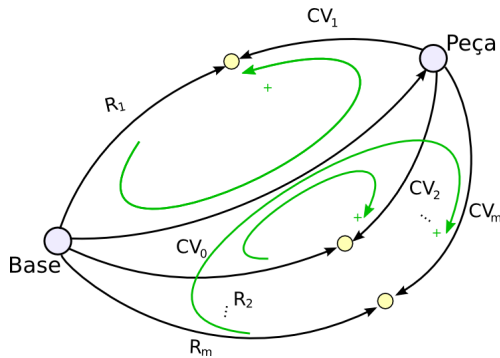


Figura 14: Representação por grafos de um SRC com a base dos manipuladores fixa.

Como pode ser observado no grafo, quando o sistema possui vários robôs se torna mais clara a utilização de uma cadeia virtual CV_0 específica para relacionar a peça ao sistema de coordenadas fixo, pois desta forma é possível trabalhar com todas as malhas de interesse do grafo utilizando a mesma estrutura. Além disso, os movimentos

relativos são definidos de forma independente. É importante notar que a malha referente ao robô 1, por exemplo, tem a seguinte composição: R_1 , CV_1 e CV_0 . E para a malha referente ao robô i , tem-se: R_i , CV_i e CV_0 com $1 \leq i \leq m$.

Assim, a sistematização de modelagem da cadeia cinemática de um SRC é abrangente no sentido de que não é necessário definir um robô principal. Todos os circuitos que contêm os robôs têm a mesma estrutura: sua base é ligada ao sistema de coordenadas fixo e seu efetuador, à cadeia virtual com relação à peça. Esta estrutura não necessita da definição prévia de quais robôs são fixos à peça e quais executam tarefa sobre a peça. Obtém-se, então, uma flexibilização com relação às abordagens que definem um robô como mestre e/ou quais são fixos à peça.

No grafo da Figura 14 a notação R_i representa o i -ésimo robô e CV_i a i -ésima cadeia virtual, em que $i = 1, 2, \dots, m$, com m robôs participantes do SRC. A terminologia CV_0 representa a cadeia virtual de posicionamento da peça com relação ao sistema de coordenadas fixo.

A matriz de circuitos B é organizada considerando a primeira coluna indicando a presença das arestas da cadeia virtual CV_0 , que é comum a todas as malhas. Em seguida, são consideradas as colunas referentes às cadeias virtuais CV_i alternadamente com as colunas que representam os robôs R_i , partindo da *malha*₁ até a *malha* _{m} . A organização da matriz D deverá estar de acordo com a proposta para a matriz de circuitos B . Assim, a matriz de circuitos B é:

$$B = \begin{matrix} & CV_0 & CV_1 & R_1 & CV_2 & R_2 & \dots & CV_m & R_m \\ \begin{matrix} malha_1 \\ malha_2 \\ \vdots \\ malha_m \end{matrix} & \left[\begin{array}{cccccccc} 1 & 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{array} \right] & \end{matrix} \quad \begin{matrix} m \times 6(m+1) + \sum_{i=1}^m n_i \\ (3.9) \end{matrix}$$

em que n_i é o número de elos do robô i , com $1 < i < m$ e m é o número de robôs componentes do sistema. Cada coluna da matriz B representa um componente principal da cadeia cinemática de forma compacta. Assim, a coluna CV_0 , por exemplo, descreve todos os elementos componentes da cadeia virtual CV_0 na ordem: P_{X0} , P_{Y0} , P_{Z0} , R_{X0} , R_{Y0} e R_{Z0} . Para este caso a dimensão do espaço de helicoides é $\lambda = 6$.

As cadeias virtuais podem ser $3P3R$, $RPPS$ e $RRPS$ dependendo da tarefa a ser designada a cada robô (mais detalhes sobre as cadeias

virtuais podem ser encontradas no Apêndice B.3).

A matriz de helicoides normalizados D é representada por:

$$D = [\hat{\$}_{CV_0} \quad \hat{\$}_{CV_1} \quad \hat{\$}_{R_1} \quad \hat{\$}_{CV_2} \quad \hat{\$}_{R_2} \quad \cdots \quad \hat{\$}_{CV_m} \quad \hat{\$}_{R_m}]_{6 \times 6(m+1) + \sum_{i=1}^m n_i} \quad (3.10)$$

A matriz de rede N da equação de restrição é obtida por:

$$N = \begin{bmatrix} D \text{ diag}\{B_1\} \\ D \text{ diag}\{B_2\} \\ \vdots \\ D \text{ diag}\{B_m\} \end{bmatrix} \quad (3.11)$$

em que m é o número de robôs que contém o sistema. A matriz de helicoides D é a matriz que contém os helicoides normalizados e $\text{diag}\{B_b\}$ é a matriz diagonal da linha b da matriz de circuitos B , com $b = 1, 2, \dots, m$.

Logo, a matriz de rede N pode ser expressa:

$$N = \begin{bmatrix} \hat{\$}_{CV_0} & \hat{\$}_{CV_1} & -\hat{\$}_{R_1} & 0 & 0 & \cdots & 0 & 0 \\ \hat{\$}_{CV_0} & 0 & 0 & \hat{\$}_{CV_2} & -\hat{\$}_{R_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{\$}_{CV_0} & 0 & 0 & 0 & 0 & \cdots & \hat{\$}_{CV_m} & -\hat{\$}_{R_m} \end{bmatrix}_{6m \times 6(m+1) + \sum_{i=1}^m n_i} \quad (3.12)$$

em que $\hat{\$}_{CV_i}$ representa a matriz 6×6 que contém os helicoides normalizados das juntas referentes a cadeia virtual i , com $i = 0, 1, \dots, m$ e $\hat{\$}_{R_j}$ é a matriz ($6 \times n_j$) que contém os helicoides normalizados das juntas do robô, com n_j sendo o número de juntas do robô j , e $j = 1, 2, \dots, m$.

Baseado no método de Davies, a matriz N pode ser decomposta em submatrizes N_p e N_s , visando o cálculo pela cinemática inversa, e são dadas por:

$$N_p = \begin{bmatrix} [\hat{\$}_{CV_0}]_{6 \times 6} & [\hat{\$}_{CV_1}]_{6 \times 6} & 0 & \cdots & 0 \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & [\hat{\$}_{CV_2}]_{6 \times 6} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & 0 & \cdots & [\hat{\$}_{CV_m}]_{6 \times 6} \end{bmatrix}$$

$$N_s = \begin{bmatrix} [-\hat{\$}_{R_1}]_{6 \times n_1} & 0 & \cdots & 0 \\ 0 & [-\hat{\$}_{R_2}]_{6 \times n_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & [-\hat{\$}_{R_m}]_{6 \times n_m} \end{bmatrix} \quad (3.13)$$

As velocidades das juntas \dot{q}_s dos robôs ficam determinadas pela Equação 3.7, reescrita aqui:

$$\dot{q}_s = -[N_s]^{-1}[N_p][\dot{q}_p] \quad (3.14)$$

Neste caso, o vetor de magnitude de velocidades referentes às juntas primárias (\dot{q}_p) é dada por:

$$\dot{q}_p = [[\dot{q}_{CV_0}]^T \quad [\dot{q}_{CV_1}]^T \quad [\dot{q}_{CV_2}]^T \quad \cdots \quad [\dot{q}_{CV_m}]^T]^T_{6(m+1) \times 1} \quad (3.15)$$

em que \dot{q}_{CV_0} é o vetor com as magnitudes da velocidade da peça com relação ao sistema de coordenadas fixo e \dot{q}_{CV_i} é o vetor com as magnitudes da velocidade desejada do efetuador i com relação à peça, com $i = 1, 2, \dots, m$ (m é o número de robôs que compõe o sistema). As velocidades do vetor \dot{q}_p são especificadas pelas trajetórias das tarefas sendo assim o resultado do ambiente da tarefa (Figura 10).

A Equação 3.14 é válida somente para o caso em que a matriz N_s é quadrada e tem posto completo. Assim, para matrizes que não admitem inversa, a Equação 3.14 não pode ser utilizada para determinar as magnitudes das velocidades das juntas \dot{q}_s . No Apêndice D é descrito a dedução da pseudo-inversa à direita adaptada para SRC como uma alternativa para encontrar as velocidades das juntas quando a matriz N_s não for invertível.

3.4 ETAPAS DO PROCESSO DE PROGRAMAÇÃO DE SRC

A partir das abordagens anteriores é estabelecida a programação de um SRC seguindo uma rotina para o cálculo da cinemática do sistema como um todo. Esta rotina segue a metodologia apresentada nas seções anteriores e é baseada na teoria de helicoides, no método de Davies e nas cadeias virtuais de Assur. Como dados iniciais são estabelecidas as informações dos robôs e as tarefas e tem-se como resposta final os parâmetros de velocidades e aceleração de cada uma das juntas dos robôs. A sistemática para determinar a cinemática de SRC é constituída das seguintes etapas (TONETTO; SIMAS; DIAS, 2010):

1. especificar as tarefas e sub-tarefas a serem executadas pelo sistema:
 - (a) definir as trajetórias necessárias para a realização das tarefas.
2. especificar os parâmetros dos robôs e organizar os dados para obter o modelo cinemático:
 - (a) estabelecer os sistemas de coordenadas fixo, da peça e dos efetuadores;
 - (b) descrever o tamanho dos elos e o tipo de juntas (rotativa ou prismática) de cada robô;
 - (c) estruturar os valores de s e s_o , e a transformação entre o sistema de coordenadas fixo com relação ao sistema de coordenadas da base de cada robô;
 - (d) aplicar a técnica dos helicoides sucessivos para os valores de s e s_o ;
 - (e) montar os helicoides normalizados ($\hat{\$}$) de cada uma das juntas que compõe o sistema, a partir dos valores de s e s_o (obtidos no item anterior).
3. estabelecer o modelo cinemático (pelo método de Davies):
 - (a) adicionar as cadeias virtuais de Assur ao sistema;
 - (b) fazer a representação abstrata do sistema através de um grafo;
 - (c) estabelecer, a partir do grafo do item anterior, as matrizes de circuitos B e as matrizes de helicoides normalizados D ;
 - (d) determinar a matriz de rede N ;
 - (e) definir as matrizes: N_p , N_s e os vetores: \dot{q}_p e \dot{q}_s .
4. aplicar a cinemática diferencial:
 - (a) calcular a cinemática inversa aplicando o método de Davies.

Essa nova sistemática adiciona estrategicamente as cadeias virtuais de Assur, sendo a cadeia virtual CV_0 adicionada para representar o movimento da peça com relação ao sistema de coordenadas fixo e as cadeias virtuais CV_i para representar o movimento do efetuador com relação à peça. Assim, as tarefas podem ser definidas de forma a representar os movimentos relativos desejados no sistema. A modelagem

cinemática proposta é aplicável em qualquer SRC e tem como foco a abordagem através da especificação da tarefa. O sistema pode conter qualquer número de robôs e é independente do número de juntas.

3.5 CONCLUSÕES

Este capítulo apresentou as principais etapas que envolvem a programação de SRC. Essas etapas foram definidas em três ambientes: (1) estrutura do robô, na qual são especificados os parâmetros dos robôs, a partir do qual podem ser gerados os seus modelos cinemáticos, (2) o ambiente da tarefa, em que são especificados os dados da tarefa a ser realizada pelo SRC e (3) a cinemática diferencial, na qual são determinados as velocidades das juntas dos robôs, para cada instante, através do modelo cinemático e do ambiente da tarefa de SRC.

Foi apresentada uma proposta de sistemática de modelagem genérica das cadeias cinemáticas que representam um SRC. Essa sistemática servirá de base para representar qualquer SRC. Uma característica importante é que cada malha do grafo que representa um SRC possui a mesma estrutura. Assim, a malha contém as arestas e vértices referentes: ao robô (R_i), a cadeia virtual (CV_0) que une a peça à base e a cadeia virtual (CV_i) que une cada efetuador a peça. Os movimentos relativos são tratados de forma independente em um SRC, o que permite maior flexibilidade e uma visão geral do SRC no tratamento da cinemática e programação.

Essa proposta inédita de sistematização da modelagem permite que se desenvolvam métodos para resolver de forma concisa e bem definida uma grande parte dos problemas da programação de SRC. Nessa sistematização da modelagem é possível determinar a cinemática de SRC, impor ou restringir movimentos em determinados elos para, por exemplo, evitar a colisão de elos do robô com o ambiente e tratar o erro de abertura da cadeia devido aos métodos de integração.

A construção utilizando as cadeias virtuais CV_0 e CV_i determinam o modo de tratamento do sistema robótico como orientado à tarefa. A CV_0 indica a tarefa de deslocamento da peça como um corpo rígido, enquanto as CV_i indicam as tarefas a serem executadas sobre a peça, independente do movimento da peça. Assim, as cadeias virtuais deixam de ser apenas um artifício que permitem a execução dos cálculos e passam a ser uma representação das tarefas.

A definição dos ambientes para a programação e a sistematização da modelagem da cadeia cinemática para SRCs possibilitou a simulação

de robôs trabalhando em cooperação para a execução de tarefas e será apresentada nos Capítulos 6 e 7.

Este capítulo permitiu analisar os componentes de um SRC e perceber que a montagem de um SRC é genérica, ou seja, cada um dos componentes é definido e são montados por blocos: robôs e cadeias virtuais, com relação a um sistema de coordenadas fixo. No próximo capítulo essa característica é enfatizada e provada matematicamente.

4 FORMALIZAÇÃO MATEMÁTICA PARA O CÁLCULO DA CINEMÁTICA DE SRC

Como apresentado no Capítulo 3, a estrutura de um grafo que representa um SRC estabelece uma forma sistemática de crescimento conforme são adicionados robôs ao sistema. Baseado nessa verificação, é proposta, neste capítulo, uma formalização matemática do grafo de um SRC e da matriz de rede que o representa. Essa formalização toma como base a equação de Euler e algumas definições sobre a teoria de grafos. É provado matematicamente que a estrutura do grafo de um SRC é dependente do número de robôs componentes no sistema, ou seja, se um SRC possui m robôs e $m + 1$ cadeias virtuais, seu grafo conterá m circuitos independentes. Como consequência desse fato, a matriz de rede N que representa um SRC também tem um crescimento estruturado e uma representação base.

Por fim, são introduzidos duas definições, denominadas de adição em série e adição em paralelo. Essas definições tratam da adição de grafos auxiliares no grafo que representa um SRC. Além disso, a partir das propriedades associadas as definições, é possível saber se o grafo adicionado alterou o número de circuitos independentes do grafo que representa o SRC.

A formalização matemática associando sistemas robóticos cooperativos com o grafo e a matriz de rede, que será apresentada nesse capítulo, é inédita.

4.1 FORMALIZAÇÃO MATEMÁTICA DO GRAFO DE UM SRC

A teoria de grafos auxilia na representação de SRC. Com a representação de grafo se determina a matriz de circuitos que é utilizada para a construção da matriz de rede N . A teoria básica sobre grafos e matrizes de circuitos é apresentada no Apêndice C. Nele são apresentados os termos e definições necessários para introduzir a formalização matemática sobre SRC.

Segundo Davies (1981), é desnecessário determinar um conjunto completo de circuitos de um grafo. Uma matriz que representa apenas os circuitos independentes de um grafo conexo é suficiente para a análise cinemática. Além disso, estabelecer uma matriz de circuitos que represente todos os circuitos de um grafo pode não ser trivial.

O foco deste trabalho são grafos conexos direcionados formados

apenas por circuitos. Esses grafos representam as cadeias cinemáticas de um SRC.

Para o caso de estudo, o objetivo é estabelecer uma forma única e padronizada de representar as cadeias cinemáticas de SRC através dos grafos e da matriz de circuitos, buscando facilitar a programação de SRC.

Assim, considera-se que o grafo que representa um SRC é composto por vários subgrafos. Cada um desses subgrafos representará os seguintes componentes do SRC:

- a base;
- os robôs;
- a peça (objeto de trabalho);
- as cadeias cinemáticas virtuais espaciais.

Desta forma, suponha que um SRC é composto por apenas **um robô**. O seu grafo é composto pelos seguintes subgrafos:

- da base;
- do robô;
- da peça (objeto de trabalho);
- da cadeia virtual que representa o movimento da peça com relação base;
- da cadeia virtual que representa o movimento da peça com relação ao efetuador do robô.

A Figura 15 mostra o grafo que representa um sistema composto por um robô. Neste grafo não estão presentes os vértices e arestas referentes aos elos e juntas da cadeia cinemática, facilitando o entendimento dos componentes principais desse grafo.

O Lema 1, descrito a seguir, estabelece que independente do número de juntas do robô, o número de circuitos independentes¹ de um grafo que representa um sistema composto por um robô é $L = 1$. Ainda, prova-se que o número de arestas e o número de vértices do subgrafo do robô é igual ao número de juntas do robôs.

¹A teoria básica sobre grafos e circuitos independentes é apresentada no Apêndice C.

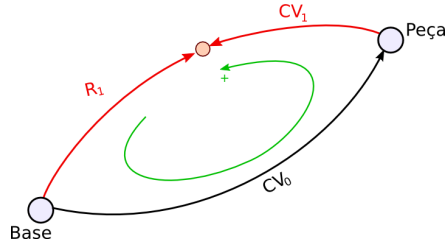


Figura 15: Sistema composto por um robô.

Lema 1. A relação $e-v+1=1$ é válida para um robô R_1 com qualquer número de juntas, em que, e é o número de arestas, v é o número de vértices. E também, ${}^n e_{R_1} = n$ e ${}^n v_{R_1} = n$, no qual, n é o número de juntas do robô, e_{R_1} é o número de arestas do grafo do robô e v_{R_1} é o número de vértices do grafo do robô.

Demonstração:

A Figura 16 auxiliará na visualização do sistema e seus componentes. Ela ilustra o sistema composto por: um robô R_1 com uma junta e um elo, uma cadeia virtual CV_1 que relaciona o movimento da peça com o efetuator, a peça, a cadeia virtual CV_0 que relaciona o movimento da peça com relação a base e por fim, a base. A Figura 16 também mostra o grafo desse sistema.

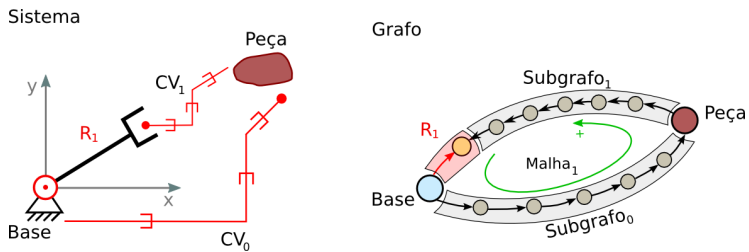


Figura 16: Sistema composto por um robô com um grau de liberdade e seu grafo.

Considerando o sistema do robô R_1 , tem-se que o número de vértices e arestas do subgrafo que representa:

- o robô é: ${}^1 e_{R_1} = 1$; ${}^1 v_{R_1} = 1$;

- a peça é: $e_p = 0; v_p = 1$;
- a base é: $e_b = 0; v_b = 1$;
- o subgrafo 0 (referente a cadeia virtual 0) é: $e_{s_0} = 6; v_{s_0} = 5$;
- o subgrafo 1 (referente a cadeia virtual 1) é: $e_{s_1} = 6; v_{s_1} = 5$.

Assim, a soma de todas as arestas desse sistema é:

$${}^1e_T = {}^1e_{R_1} + e_p + e_b + e_{s_0} + e_{s_1} = 1 + 0 + 0 + 6 + 6 = 13$$

E a soma de todos os vértices desse sistema é:

$${}^1v_T = {}^1v_{R_1} + v_p + v_b + v_{s_0} + v_{s_1} = 1 + 1 + 1 + 5 + 5 = 13$$

Então, substituindo o número total de arestas e vértices deste sistema na equação de Euler, tem-se que o número de circuitos independentes é:

$${}^1e_T - {}^1v_T + 1 = 13 - 13 + 1 = 1$$

Suponha agora que o robô possua n juntas de um grau de liberdade cada (a Figura 17 ilustra o sistema).

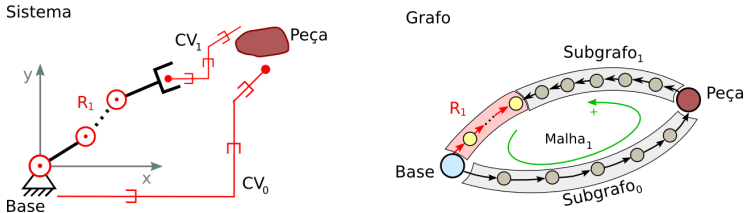


Figura 17: Sistema composto por um robô de n graus de liberdade e seu grafo.

Suponha ainda que, para esse sistema, cada subgrafo tenha os seguintes números de vértices e arestas associados:

- ao robô é: ${}^n e_{R_1} = n; {}^n v_{R_1} = n$
- a peça é: $e_p = 0; v_p = 1$
- a base é: $e_b = 0; v_b = 1$
- ao subgrafo 0 (referente a cadeia virtual 0) é: $e_{s_0} = 6; v_{s_0} = 5$

- ao subgrafo 1 (referente a cadeia virtual 1) é: $e_{s_1} = 6$; $v_{s_1} = 5$

Somando todas as arestas desse sistema, tem-se:

$${}^n e_T = {}^n e_{R_1} + e_p + e_b + e_{s_0} + e_{s_1} = n + 0 + 0 + 6 + 6 = n + 12$$

Já, o número total de vértices desse sistema é:

$${}^n v_T = {}^n v_{R_1} + v_p + v_b + v_{s_0} + v_{s_1} = n + 1 + 1 + 5 + 5 = n + 12$$

Desta forma, se o robô possui n juntas, então, o número de circuitos independentes calculado a partir da equação de Euler, será:

$${}^n e_T - {}^n v_T + 1 = n + 12 - n - 12 + 1 = 1$$

Provando que a relação é válida para um sistema em que o robô possua $n + 1$ juntas (esse sistema e seu grafo está representado na Figura 18):

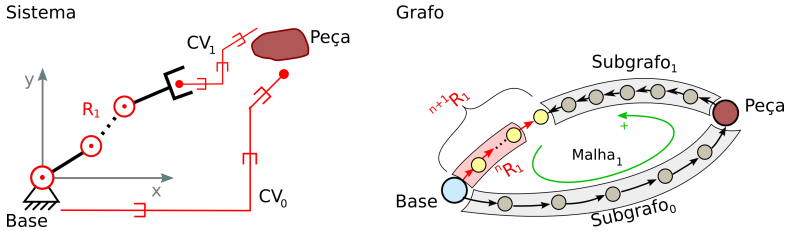


Figura 18: Sistema composto por um robô de $n + 1$ graus de liberdade.

Pela hipótese de indução (*HI*), o número de arestas e vértices do subgrafo do robô com n juntas é ${}^n e_{R_1} = n$; ${}^n v_{R_1} = n$ (OBS: Os subgrafos do robô R_1 que estão destacados nas Figuras 17 e 18 são iguais).

Adicionalmente ao subgrafo do robô R_1 demarcado na Figura 18 tem-se uma aresta e um vértice. O número de vértices e arestas para o subgrafo que representa:

- o robô é:

$${}^{n+1} e_{R_1} = n + 1 \Rightarrow {}^{n+1} e_{R_1} = {}^n e_{R_1} + 1$$

$${}^{n+1} v_{R_1} = n + 1 \Rightarrow {}^{n+1} v_{R_1} = {}^n v_{R_1} + 1$$

- a peça é: $e_p = 0$; $v_p = 1$
- a base é: $e_b = 0$; $v_b = 1$

- ao subgrafo 0 (referente a cadeia virtual 0) é: $e_{s_0} = 6$; $v_{s_0} = 5$
- ao subgrafo 1 (referente a cadeia virtual 1) é: $e_{s_1} = 6$; $v_{s_1} = 5$

As equações ${}^{n+1}e_{R_1} = n + 1$ e ${}^{n+1}v_{R_1} = n + 1$ confirmam a hipótese de indução. A hipótese de indução é verdadeira.

Desta forma, somando todas as arestas do sistema, tem-se:

$$\begin{aligned}
 {}^{n+1}e_T &= {}^{n+1}e_{R_1} + e_p + e_b + e_{s_0} + e_{s_1} \\
 &= n + 1 + 0 + 0 + 6 + 6 \\
 &= n + 13 \Rightarrow {}^{n+1}e_T = (n + 1) + 12
 \end{aligned}$$

E a soma de todas os vértices é:

$$\begin{aligned}
 {}^{n+1}v_T &= {}^{n+1}v_{R_1} + v_p + v_b + v_{s_0} + v_{s_1} \\
 &= n + 1 + 0 + 0 + 6 + 6 \\
 &= n + 13 \Rightarrow {}^{n+1}v_T = (n + 1) + 12
 \end{aligned}$$

Assim, pela equação de Euler, tem-se:

$${}^{n+1}e_T - {}^{n+1}v_T + 1 = (n + 13) - (n + 13) + 1 = 1$$

E o número de arestas e vértices total é dado por:

$${}^{n+1}e_T = (n + 1) + 12 \text{ e } {}^{n+1}v_T = (n + 1) + 12$$

Que confirmam a hipótese de indução:

$${}^n e_T = n + 12; \quad {}^n v_T = n + 12$$

$${}^n e_T - {}^n v_T + 1 = 1, \quad \forall n \in \mathbb{N}$$

■

Teorema 1. *O número de circuitos independentes de um grafo é igual ao número de subgrafos que representam os robôs adicionados ao sistema robótico cooperativo, ou seja, $\mathbf{e-v+1=m}$, em que, \mathbf{e} é o número de arestas, \mathbf{v} é o número de vértices, \mathbf{m} é o número de robôs e o número de circuitos independentes do grafo.*

Demonstração:

Essa demonstração está dividida em duas partes, na primeira é demonstrado que a relação $e - v + 1$ é válida para um robô para n juntas e na segunda, que a equação $e - v + 1 = m$ é válida para sistemas compostos por qualquer número de robôs.

Parte 1: Pelo Lema 1, a relação é válida para um robô com n juntas.

Parte 2:

Deseja-se provar que: $e - v + 1 = m$, onde m é o número de robôs.

Para o caso em que o sistema possua apenas um robô, ou seja, $m = 1$, o grafo do sistema está ilustrado na Figura 19.

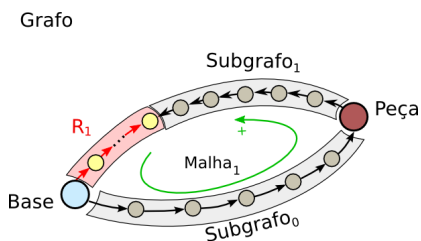


Figura 19: Grafo de um sistema composto por um robô com n graus de liberdade.

Pelo Lema 1, mostrado anteriormente ${}^{n_1}e_{T_1} - {}^{n_1}v_{T_1} + 1 = 1$, para $m = 1$, é válido.

Suponha agora, que o sistema tenha m robôs, ou seja, caso $m = m$ robôs. A Figura 20 apresenta o grafo do sistema com m robôs.

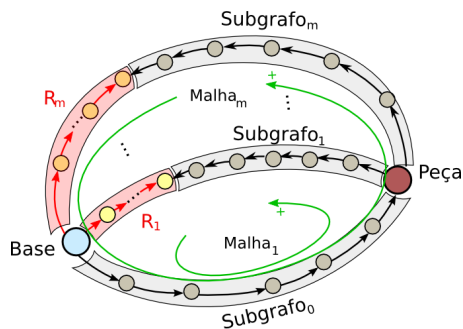


Figura 20: Grafo do sistema robótico composto por m robôs.

Sejam:

$${}^{nm}e_{TR_m} = \sum_{i=1}^m n_i e_{R_i} = \sum_{i=1}^m n_i$$

$${}^{nm}v_{TR_m} = \sum_{i=1}^m n_i v_{R_i} = \sum_{i=1}^m n_i \text{ (pelo Lema 1).}$$

Assume-se as seguintes hipóteses de indução (HI):

- ${}^{nm}e_{T_m} - {}^{nm}v_{T_m} + 1 = m$
- ${}^{nm}e_{T_m} = \sum_{i=1}^m n_i + (m+1)6$
- ${}^{nm}v_{T_m} = \sum_{i=1}^m n_i + 2 + (m+1)5$

De fato, a hipótese de indução parte do desenvolvimento apresentado a seguir.

O número de vértices e arestas dos subgrafos associados:

- aos robôs é:

$${}^{nm}e_{TR_m} = n_1 e_{R_1} + n_2 e_{R_2} + \cdots + n_m e_{R_m} = n_1 + n_2 + \cdots + n_m = \sum_{i=1}^m n_i \Rightarrow$$

$${}^{nm}e_{TR_m} = \sum_{i=1}^m n_i$$

$${}^{nm}v_{TR_m} = n_1 v_{R_1} + n_2 v_{R_2} + \cdots + n_m v_{R_m} = n_1 + n_2 + \cdots + n_m = \sum_{i=1}^m n_i \Rightarrow$$

$${}^{nm}v_{TR_m} = \sum_{i=1}^m n_i$$

- à peça é: $e_{p_m} = 0$; $v_{p_m} = 1$
- à base é: $e_{b_m} = 0$; $v_{b_m} = 1$
- ao subgrafo 0 (referente a cadeia virtual 0) é: $e_{s_0} = 6$; $v_{s_0} = 5$
- ao subgrafo 1 (referente a cadeia virtual 1) é: $e_{s_1} = 6$; $v_{s_1} = 5$
- \vdots \vdots \vdots \vdots
- ao subgrafo m (referente a cadeia virtual m) é: $e_{s_m} = 6$; $v_{s_m} = 5$

O número total de arestas dos subgrafos (referente às cadeias virtuais) é:

$$e_{s_T} = e_{s_0} + e_{s_1} + \cdots + e_{s_m} = 6 + 6 + \cdots + 6 = (m+1)6$$

O número total de vértices dos subgrafos (referente às cadeias virtuais) é:

$$v_{s_T} = v_{s_0} + v_{s_1} + \cdots + v_{s_m} = 5 + 5 + \cdots + 5 = (m+1)5$$

Assim, o número total de arestas e vértices do sistema é:

$$\begin{aligned}
 {}^{n_m}e_{T_m} &= {}^{n_m}e_{TR_m} + e_{p_m} + e_{b_m} + e_{s_T} \\
 &= \sum_{i=1}^m n_i + 0 + 0 + (m+1)6 \\
 &= \sum_{i=1}^m n_i + (m+1)6
 \end{aligned}$$

$$\begin{aligned}
 {}^{n_m}v_{T_m} &= {}^{n_m}v_{TR_m} + v_{p_m} + v_{b_m} + v_{s_T} \\
 &= \sum_{i=1}^m n_i + 1 + 1 + (m+1)5 \\
 &= \sum_{i=1}^m n_i + 2 + (m+1)5
 \end{aligned}$$

Desta forma, pela equação de Euler:

$$\begin{aligned}
 {}^{n_m}e_{T_m} - {}^{n_m}v_{T_m} + 1 &= \left[\sum_{i=1}^m n_i + (m+1)6 \right] - \left[\sum_{i=1}^m n_i + 2 + (m+1)5 \right] + 1 \\
 &= \sum_{i=1}^m n_i + (m+1)6 - \sum_{i=1}^m n_i - 2 - (m+1)5 + 1 \\
 &= (m+1)(6-5) - 1 \\
 &= m + 1 - 1 \\
 &= m
 \end{aligned}$$

Tem-se que a hipótese de indução é:

$${}^{n_m}e_{T_m} - {}^{n_m}v_{T_m} + 1 = m, \quad \forall \quad n_1, n_2, \dots, n_m \in \mathbb{N}$$

Provando que vale para o caso em que o sistema possua $m+1$ robôs. A Figura 21 ilustra o grafo desse sistema.

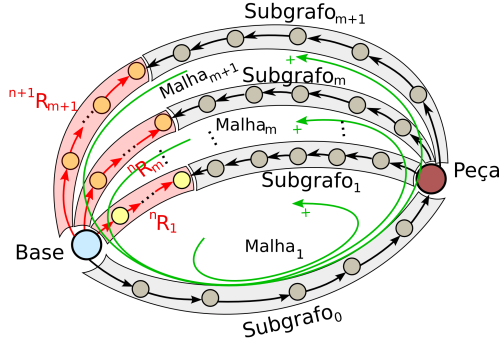
Para $m+1$ robôs, tem-se pelo Lema 1 que:

$${}^{n_{m+1}}e_{R_{m+1}} = n_{m+1} \quad \text{e} \quad {}^{n_{m+1}}v_{R_{m+1}} = n_{m+1}$$

Assim, o número de arestas e vértices dos subgrafos associados:

- aos robôs é:

Grafo

Figura 21: Grafo do sistema robótico composto por $m + 1$ robôs.

$${}^{n_{m+1}}e_{TR_{m+1}} = \sum_{i=1}^{m+1} n_i e_{R_i} = \sum_{i=1}^{m+1} n_i$$

$${}^{n_{m+1}}v_{TR_{m+1}} = \sum_{i=1}^{m+1} n_i v_{R_i} = \sum_{i=1}^{m+1} n_i$$

- à peça é: $e_{p_m} = 0$; $v_{p_m} = 1$
- à base é: $e_{b_m} = 0$; $v_{b_m} = 1$
- ao subgrafo 0 (referente a cadeia virtual 0) é: $e_{s_0} = 6$; $v_{s_0} = 5$
- ao subgrafo 1 (referente a cadeia virtual 1) é: $e_{s_1} = 6$; $v_{s_1} = 5$
- \vdots \vdots \vdots \vdots
- ao subgrafo m (referente a cadeia virtual m) é: $e_{s_m} = 6$; $v_{s_m} = 5$
- ao subgrafo $m + 1$ (referente a cadeia virtual $m + 1$) é: $e_{s_m} = 6$; $v_{s_m} = 5$

O número total de arestas dos subgrafos (referente às cadeias virtuais) é:

$$\begin{aligned} e_{s_T} &= e_{s_0} + e_{s_1} + \dots + e_{s_m} + e_{s_{m+1}} \\ &= 6 + 6 + \dots + 6 \\ &= 6(m+2) \end{aligned}$$

O número total de vértices dos subgrafos (referente às cadeias virtuais) é:

$$\begin{aligned}
 v_{s_T} &= v_{s_0} + v_{s_1} + \cdots + v_{s_m} + v_{s_{m+1}} \\
 &= 5 + 5 + \cdots + 5 \\
 &= 5(m+2)
 \end{aligned}$$

Desta forma, o número total de arestas do sistema é²:

$$\begin{aligned}
 {}^{n_{m+1}}e_{T_{m+1}} &= {}^{n_{m+1}}e_{TR_{m+1}} + e_{p_m} + e_{b_m} + e_{s_T} \\
 &= \sum_{i=1}^{m+1} n_i + 0 + 0 + (m+2)6 \\
 &= \sum_{i=1}^m n_i + n_{i+1} + (m+1)6 + 6 \\
 &\stackrel{HI}{=} {}^{n_m}e_{T_m} + n_{i+1} + 6
 \end{aligned}$$

E o número total de vértices é:

$$\begin{aligned}
 {}^{n_{m+1}}v_{T_{m+1}} &= {}^{n_{m+1}}v_{TR_{m+1}} + v_{p_m} + v_{b_m} + v_{s_T} \\
 &= \sum_{i=1}^{m+1} n_i + 1 + 1 + (m+2)5 \\
 &= \sum_{i=1}^m n_i + n_{i+1} + 2 + (m+1)5 + 5 \\
 &\stackrel{HI}{=} {}^{n_m}v_{T_m} + n_{i+1} + 5
 \end{aligned}$$

Assim, substituindo na equação de Euler, tem-se:

$$\begin{aligned}
 {}^{n_{m+1}}e_{T_{m+1}} - {}^{n_{m+1}}v_{T_{m+1}} + 1 &= ({}^{n_m}e_{T_m} + n_{i+1} + 6) - ({}^{n_m}v_{T_m} + n_{i+1} + 5) + 1 \\
 &= {}^{n_m}e_{T_m} + n_{i+1} + 6 - {}^{n_m}v_{T_m} - n_{i+1} - 5 + 1 \\
 &= {}^{n_m}e_{T_m} - {}^{n_m}v_{T_m} + 1 + 6 - 5 \\
 &\stackrel{HI}{=} m + 1
 \end{aligned}$$

²A notação $\stackrel{HI}{=}$ indica que neste momento foi aplicada a hipótese de indução.

Foi provado que o número de circuitos independentes de um grafo planar conexo é igual ao número de robôs considerando o sistema composto pelos subgrafos que representam: os robôs (m), a base, a peça e as cadeias virtuais ($m + 1$).

■

4.2 FORMALIZAÇÃO MATEMÁTICA DA MATRIZ DE REDE DE UM SRC

No Teorema 1 demonstrado anteriormente foi provado que o número de robôs do sistema é igual ao número de circuitos independentes³ de um grafo conexo, dadas as devidas características do sistema (robô, peça, base, subgrafos das cadeias virtuais). Esse teorema auxiliará na demonstração de um segundo teorema que aborda a adição de elementos na matriz de rede N de forma sistemática. O número de linhas da matriz de rede N está associado ao número de circuitos do grafo. Assim, se o número de circuitos do grafo tem alguma relação com o número de robôs do sistema (Teorema 1), é possível estabelecer algum critério de composição dos elementos da matriz de rede N , já que a estrutura de cada circuito é semelhante.

Neste trabalho são abordadas apenas as matrizes de circuitos B que representam os circuitos independentes do sistema, ou seja, a matriz de circuitos B será formada apenas com referência aos circuitos independentes do grafo.

A matriz de rede N é obtida multiplicando-se a matriz de helicoides normalizados D pela matriz diagonal de cada uma das linhas da matriz B (DAVIES, 1981; CAMPOS, 2004).

$$N_{(\lambda l \times F_b)} = \begin{bmatrix} D \cdot B_1 \\ D \cdot B_2 \\ \vdots \\ D \cdot B_l \end{bmatrix} \quad (4.1)$$

em que, B_i , com $i = 1, 2, \dots, l$, é obtida a partir dos circuitos independentes do grafo que representa a cadeia cinemática do SRC, l é o número de circuitos que o grafo possui, neste caso, l será igual a L número de circuitos independentes.

É possível provar que a matriz de rede N cresce de forma siste-

³A teoria básica sobre grafos e matrizes de circuitos é apresentada no Apêndice C.

mática à medida que são adicionados robôs ao sistema. O sistema é composto por m robôs, a base, a peça e $(m+1)$ cadeias virtuais.

A matriz de rede N possui $\lambda \cdot l$ linhas e F_b colunas, em que, λ é a ordem mínima do sistema de helicoides ($\lambda = 3$ para casos planares, $\lambda = 6$ para casos espaciais), l é o número de circuitos independentes do grafo e F_b é o somatório dos graus de liberdade de todas as juntas da cadeia cinemática (CAMPOS, 2004).

Teorema 2. *O grafo de um sistema composto por m robôs é representado através de uma matriz de rede N , com $\lambda \cdot l$ linhas e F_b colunas, em que $F_b = f_{CV_0} + \sum_{i=1}^m (f_{R_i} + f_{CV_i}) = \lambda(m+1) + \sum_{i=1}^m n_i$, com $\lambda = 6$ para casos espaciais e n_i é o número de juntas do i -ésimo robô.*

Demonstração:

Para o caso em que o SRC é composto por **um robô**, ou seja, $m = 1$. Então:

$$\lambda = 6$$

$$l = m = 1 \text{ (pelo Teorema 1)}$$

$F_b = f_{CV_0} + f_{R_1} + f_{CV_1}$, em que f_{CV_0} , f_{CV_1} e f_{R_1} representam a soma dos graus de liberdade das juntas da cadeia virtual 0, da cadeia virtual 1 e da cadeia cinemática do robô, respectivamente.

Assim, considerando que as cadeias virtuais adicionadas ao sistema são espaciais, cada cadeia virtual possui 6 graus de liberdade, logo, $f_{CV_0} = f_{CV_1} = 6$. Assumindo que o robô possui n_1 juntas simples (com um grau de liberdade: juntas prismáticas ou rotativas), logo, $f_{R_1} = n_1$

Desta forma,

$$F_b = f_{CV_0} + f_{R_1} + f_{CV_1} = 6 + n_1 + 6 = n_1 + 12$$

Então, a matriz N , para um SRC composto por um robô, possui $\lambda \cdot l$ linhas $\Rightarrow 6 \cdot 1 = 6$ e $F_b = n_1 + 12$ colunas.

OBS: As matrizes contendo o helicoides normalizados são:

$$[CV_0] = \begin{bmatrix} \$P_{x0} & \$P_{y0} & \$P_{z0} & \$R_{x0} & \$R_{y0} & \$R_{z0} \end{bmatrix}_{6 \times 6}$$

$$[CV_1] = \begin{bmatrix} \$P_{x1} & \$P_{y1} & \$P_{z1} & \$R_{x1} & \$R_{y1} & \$R_{z1} \end{bmatrix}_{6 \times 6}$$

$$[-R_1] = -[\$1 \quad \$2 \quad \$3 \quad \dots \quad \$n_1]_{6 \times n_1}$$

$$\text{Em que, } N = [CV_0 \quad CV_1 \quad -R_1]_{6 \times (12+n_1)}$$

O objetivo é provar que ao adicionar m robôs ao sistema tem-se que a matriz N terá $\lambda \cdot l$ linhas e $6(m+1) + \sum_{i=1}^m n_i$ colunas.

Para o caso em que o sistema possui $m = m$ robôs, tem-se que:

$$\lambda = 6;$$

$$l = m;$$

$$F_{b_m} = f_{CV_0} + f_{R_1} + f_{CV_1} + f_{R_2} + f_{CV_2} + \dots + f_{R_m} + f_{CV_m} = 6(m+1) +$$

$\sum_{i=1}^m n_i$ (hipótese de indução).

E como hipótese de indução, assume-se:

$$N_m = \begin{bmatrix} CV_0 & CV_1 & -R_1 & 0 & 0 & \cdots & 0 & 0 \\ CV_0 & 0 & 0 & CV_2 & -R_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ CV_0 & 0 & 0 & 0 & 0 & \cdots & CV_m & -R_m \end{bmatrix} \quad (4.2)$$

Provando que vale para o caso com $m+1$ robôs:

$$\lambda = 6;$$

$$l = m + 1;$$

$$\begin{aligned} F_{b_{m+1}} &= f_{CV_0} + f_{R_1} + f_{CV_1} + \cdots + f_{R_m} + f_{CV_m} + f_{R_{m+1}} + f_{CV_{m+1}} \\ &= (f_{CV_0} + f_{R_1} + f_{CV_1} + \cdots + f_{R_m} + f_{CV_m}) + f_{R_{m+1}} + f_{CV_{m+1}} \\ &\stackrel{HI}{=} \left[6(m+1) + \sum_{i=1}^m n_i \right] + n_{m+1} + 6 \\ &= 6[(m+1) + 1] + \sum_{i=1}^m n_i + n_{m+1} \\ &= 6(m+2) + \sum_{i=1}^{m+1} n_i \end{aligned}$$

A matriz N tem a seguinte estrutura:

$$N_{m+1} = \begin{bmatrix} N_m & 0 & 0 \\ CV_0 & [0] & CV_{m+1} & -R_{m+1} \end{bmatrix}$$

$$\stackrel{HI}{=} \begin{bmatrix} CV_0 & CV_1 & -R_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ CV_0 & 0 & 0 & CV_2 & -R_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ CV_0 & 0 & 0 & 0 & 0 & \cdots & CV_m & -R_m & 0 & 0 \\ CV_0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & CV_{m+1} & -R_{m+1} \end{bmatrix}$$

A matriz de rede N_{m+1} tem $l \cdot \lambda$ linhas e F_b colunas.

■

O número de linhas de N_m é igual ao número de circuitos independentes (Teorema 1) multiplicado pela dimensão do sistema de helicoides. Além disso, todos os dados dos robôs são inclusos na ma-

triz de rede N . Segundo Davies (DAVIES, 1981), estas condições são suficientes para que a matriz N_m apresente um conjunto de circuitos independentes e assim, descrever completamente a cinemática de um SRC. O SRC é composto por m robôs, $m + 1$ cadeias virtuais, peça e base.

Corolário 1. *A adição de um novo robô no sistema, implica no aumento de λ linhas na matriz de rede N e $n + p$ colunas, em que n é o número de juntas do robô adicionado e p é o número de juntas da cadeia virtual (em que, $p = 6$ se a cadeia virtual é espacial e $p = 3$ se for planar, $\lambda = 6$ para casos espaciais e $\lambda = 3$ para casos planares).*

Demonstração:

Para o caso $m = m$ robôs no sistema, tem-se que a matriz N possui $m \cdot \lambda$ linhas e o número de colunas é $F_{b_m} = f_{CV_0} + \sum_{i=1}^m (f_{R_i} + f_{CV_i})$

Para o caso em que o sistema contém $m + 1$ robôs, tem-se que o número de linhas da matriz N é $(m + 1)\lambda$ e o número de colunas é $F_{b_{m+1}} = f_{CV_0} + \sum_{i=1}^{m+1} (f_{R_i} + f_{CV_i}) = f_{CV_0} + \sum_{i=1}^m (f_{R_i} + f_{CV_i}) + f_{R_{m+1}} + f_{CV_{m+1}}$

Assim, o número de linhas adicionais na matriz N é dado pela diferença entre o número de linhas do sistema com m robôs e o sistema com $m + 1$ robôs:

$$(m + 1)\lambda - md = md + \lambda - md = \lambda$$

E o número de colunas adicionais na matriz N é calculado também pela diferença entre o número de colunas do sistema com m robôs e do sistema com $m + 1$ robôs:

$$\begin{aligned} F_{b_{m+1}} - F_{b_m} &= f_{CV_0} + \left(\sum_{i=1}^m f_{R_i} + f_{CV_i} \right) + f_{R_{m+1}} + f_{CV_{m+1}} - \left[f_{CV_0} + \sum_{i=1}^m (f_{R_i} + f_{CV_i}) \right] \\ &= f_{CV_0} + \left(\sum_{i=1}^m f_{R_i} + f_{CV_i} \right) + f_{R_{m+1}} + f_{CV_{m+1}} - f_{CV_0} - \sum_{i=1}^m (f_{R_i} + f_{CV_i}) \\ &= f_{R_{m+1}} + f_{CV_{m+1}} \\ &= n + p \end{aligned}$$

onde: n é o número de juntas do robô adicional e p é o número de juntas da cadeia virtual adicional.

■

Desta forma, tem-se que a adição de um novo robô na malha implica em um aumento de λ linhas e $n + p$ colunas na matriz de rede, onde n é o número de juntas do robô adicional e p é o número de

juntas da cadeia virtual adicional (ver a Equação 4.3, as linhas e colunas destacadas em vermelho fazem referência a adição do novo robô no sistema).

$$N_{m+1} = \begin{bmatrix} CV_0 & CV_1 & -R_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ CV_0 & 0 & 0 & CV_2 & -R_2 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ CV_0 & 0 & 0 & 0 & 0 & \dots & CV_m & -R_m & 0 & 0 \\ CV_0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & CV_{m+1} & -R_{m+1} \end{bmatrix} \quad (4.3)$$

4.3 ADIÇÃO DE GRAFOS NUM SRC

Suponha agora que se deseja adicionar um grafo em outro grafo, ou seja, unir dois grafos com determinadas características, gerando um terceiro grafo. Para isso são apresentadas a seguir duas definições, denominadas adição em série e adição em paralelo de grafos, que auxiliarão na adição de grafos extras que representam características adicionais no SRC. Dessa forma, um dos grafos pode representar, por exemplo, uma cadeia cinemática virtual de erro ou uma cadeia cinemática virtual para evitamento de colisão. As Figuras 22 e 24 ilustram essas definições.

Definição 1. Adição em série: *Sejam dois grafos \mathbf{G} e \mathbf{H} . Sejam, ainda, dois vértices adjacentes de \mathbf{G} , \mathbf{v} e \mathbf{w} , e a aresta \mathbf{e} que os conecta, e dois vértices quaisquer de \mathbf{H} , \mathbf{t} e \mathbf{u} . A adição em série de \mathbf{H} em \mathbf{G} consiste em tornar adjacente (\mathbf{v}, \mathbf{t}) por meio de uma nova aresta \mathbf{e}' e (\mathbf{w}, \mathbf{u}) pela aresta \mathbf{e} , de forma que \mathbf{v} e \mathbf{w} não sejam mais adjacentes.*

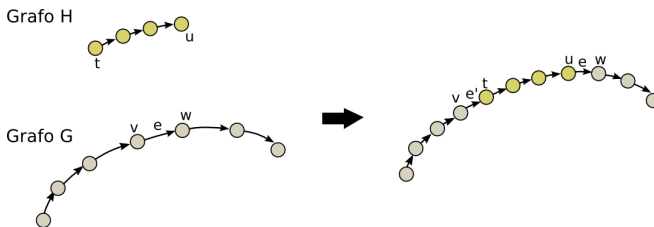


Figura 22: Adição de grafos em série.

Propriedade 1. *A adição em série de um grafo \mathbf{H} sem circuitos em um grafo \mathbf{G} não altera o número de circuitos independentes desse grafo.*

Suponha que o número de circuitos independentes do grafo \mathbf{G} é \mathbf{A} . Pela fórmula de Euler (ver anexo C), tem-se:

$e_G - v_G + 1 = A$, em que e_G é o número de arestas do grafo \mathbf{G} e v_G é o número de vértices do grafo \mathbf{G} .

Da mesma forma, para o grafo \mathbf{H} sem circuitos, tem-se:

$e_H - v_H + 1 = 0$, em que e_H é o número de arestas do grafo \mathbf{H} e v_H é o número de vértices do grafo \mathbf{H} .

Demonstra-se, assim, que na adição em série é criada uma nova aresta e' . Pela Definição 1, o número de circuitos independentes do novo grafo \mathbf{F} , é dado por:

$e_F - v_F + 1 = (e_G + e_H + 1) - (v_G + v_H) + 1 = e_G + e_H + 1 - v_G - v_H + 1 = (e_G - v_G + 1) + (e_H - v_H + 1) = A + 0 = A$,

onde: e_F é o número de arestas e v_F é o número de vértices do novo grafo, gerado pela adição do grafo \mathbf{H} no grafo \mathbf{G} .

■

Para facilitar o entendimento e aplicação da Propriedade 1 é ilustrado na Figura 23 um novo grafo \mathbf{F} que representa um SRC com m robôs. Nesse grafo \mathbf{F} foi adicionado um grafo \mathbf{H} em série.

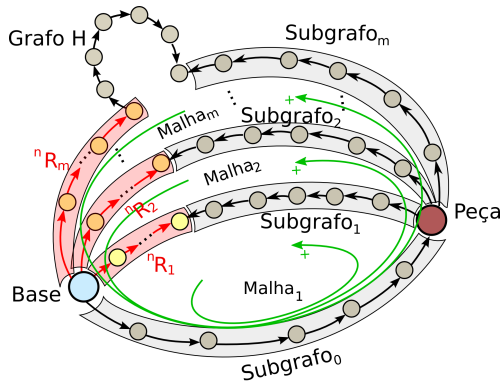


Figura 23: Grafo \mathbf{F} gerado ao se adicionar o grafo \mathbf{H} .

Definição 2. Adição em paralelo: *Sejam dois grafos \mathbf{G} e \mathbf{H} . Sejam, ainda, dois vértices quaisquer de \mathbf{G} : \mathbf{v} e \mathbf{w} e dois vértices quaisquer de \mathbf{H} : \mathbf{t} e \mathbf{u} . A adição em paralelo de \mathbf{H} em \mathbf{G} consiste em tornar adjacente*

(\mathbf{v}, \mathbf{t}) por meio de uma nova aresta \mathbf{e} e (\mathbf{w}, \mathbf{u}) por meio de uma nova aresta \mathbf{e}' .

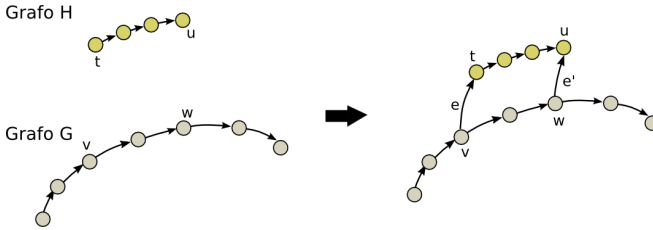


Figura 24: Adição de grafos em paralelo.

Propriedade 2. A adição em paralelo de um grafo \mathbf{H} sem circuitos em um grafo \mathbf{G} , com \mathbf{A} circuitos independentes, resulta no novo grafo \mathbf{F} com $\mathbf{A} + 1$ circuitos independentes.

Pela Equação de Euler, o número de circuitos independentes de um grafo \mathbf{G} é \mathbf{A} (conforme o Anexo C), então:

$$e_G - v_G + 1 = A$$

E para o grafo \mathbf{H} sem circuitos, tem-se:

$$e_H - v_H + 1 = 0$$

Demonstra-se, que na adição em paralelo são criadas duas novas arestas: e e e' . Pela Definição 2, o número de circuitos independentes do novo grafo \mathbf{F} , é dado por:

$$\begin{aligned} e_F - v_F + 1 &= (e_G + e_H + 2) - (v_G + v_H) + 1 = e_G + e_H + 1 - v_G - v_H + 1 + 1 \\ &= (e_G - v_G + 1) + (e_H - v_H + 1) + 1 = A + 0 + 1 = A + 1 \end{aligned}$$

■

A Figura 25 mostra a adição em paralelo de um grafo \mathbf{H} no grafo \mathbf{F} que representa um SRC.

Pelas propriedades é possível notar que quando se adiciona um grafo em série, ele não alterará o número de circuitos independentes do grafo. Quando o grafo é adicionado em paralelo, o número de circuitos independentes aumentará em uma unidade.

A adição de circuitos significa que a matriz de rede N é alterada em seu número de linhas e colunas quando adicionado subgrafos em paralelo. Se são adicionados subgrafos em série, apenas o número de colunas é alterado na matriz N .

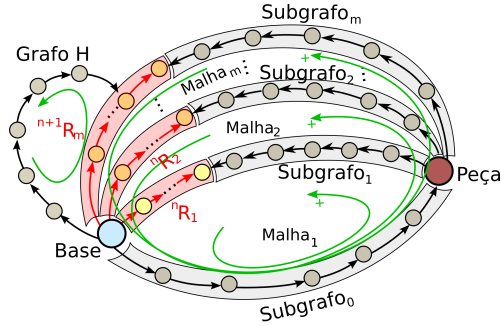


Figura 25: Grafo F gerado ao se adicionar um grafo H .

4.4 CONCLUSÕES

Este capítulo formalizou a metodologia apresentada no Capítulo 3. Foram apresentados dois teoremas e um corolário que permitem trabalhar com SRC. O SRC é composto por m robôs e $m + 1$ cadeias virtuais.

Além disso, com as Propriedades 1 e 2, a metodologia torna-se mais completa e genérica, pois é possível tratar de casos que não estão incluídos em um SRC com m robôs, como por exemplo, tratar da colisão ou mesmo do erro de integração. Esses casos serão especificados no próximo capítulo e serão tratados como características adicionais de um SRC. As operações de adição em paralelo e série definidas neste capítulo permitem um grande número de modificações, sendo algumas possibilidades apresentadas no Capítulo 5.

5 CARACTERÍSTICAS ADICIONAIS A SRC

O SRC foi tratado nos capítulos anteriores como sendo composto por m robôs e $m + 1$ cadeias virtuais e com a possibilidade de adição de cadeias cinemáticas baseadas nas Propriedades 1 e 2 (Capítulo 4). Neste capítulo são abordados alguns casos extras de programação de SRC que podem ser resolvidos fazendo uso da adição de cadeias virtuais. Esses casos extras são tratados como características adicionais em SRC. São discutidas três características adicionais: controle do erro de integração, controle de juntas e deslocamentos relativos dos robôs com relação a um sistema de coordenadas fixo.

5.1 CADEIA VIRTUAL DO ERRO

Um SRC é considerado um sistema fechado, já que são adicionadas cadeias virtuais de fechamento do sistema em todas as malhas conforme metodologia proposta no Capítulo 3. O procedimento de conectar ou fechar as cadeias cinemáticas que representam o SRC é denominado de método das restrições cinemáticas (SANTOS, 2006). Segundo esse método, com as cadeias cinemáticas fechadas é possível impor ou monitorar os movimentos da cadeia cinemática real.

O método de Davies, quando aplicado em cadeias cinemáticas fechadas, tem como resultado as velocidades das juntas secundárias em função das velocidades das juntas primárias. Para se obter a posição de cada uma das juntas secundárias do sistema é necessário utilizar algum método numérico de integração. Porém, esses métodos fornecem resultados aproximados e, com isso, provocam um fenômeno chamado de *drift* (escorregamento) na solução. Como consequência da integração da cinemática diferencial, os erros numéricos gerados no processo de integração provocam uma localização não desejada no posicionamento do efetuador. Assim, os resultados obtidos não estão de acordo com a posição, na qual o efetuador deveria estar (SICILIANO et al., 2009; SIMAS, 2008). Esse fenômeno é mais crítico em sistemas fechados, uma vez que esses erros podem implicar na abertura física da estrutura ou podem gerar torques excessivos nas juntas (SIMAS, 2008).

Guenther et al. (2008) desenvolveram um novo método de integração que permite minimizar o erro de fechamento das cadeias cinemáticas. O método consiste em adicionar cadeias virtuais de controle do erro de fechamento. O erro pode ser representado convenientemente

utilizando as cadeias virtuais do tipo PPR (para casos planares) e 3P3R (para casos espaciais), pois através delas é possível desacoplar o erro linear do erro de orientação. Entretanto, se utilizadas as cadeias RPR, PPRS e PRRS, estas podem provocar erros numéricos semelhantes a singularidades, se as dimensões dos erros forem relativamente pequenas (GUENTHER et al., 2008; FONTAN, 2007).

Assim, utilizando o método de Davies para determinar a cinemática diferencial, tem-se que, a equação considerando a cadeia virtual de erro passa a ser:

$$N_s \dot{q}_s + N_p \dot{q}_p + N_e \dot{q}_e = 0 \quad (5.1)$$

Nesta equação, a matriz de rede N foi decomposta em N_s , N_p e N_e , em que, N_s é a matriz referente aos helicoides correspondentes às juntas secundárias obtidas pela integração numérica. A matriz N_p é associada aos helicoides correspondentes às juntas primárias e, por fim, N_e é a matriz que contém os helicoides referentes a cadeia virtual de erro. O vetor \dot{q}_e é o vetor de magnitudes correspondente as velocidades das juntas controle do erro (SIMAS, 2008). Assim, a magnitude das velocidades das juntas secundárias, que pode ser obtida isolando-se \dot{q}_s na Equação 5.1, é:

$$\dot{q}_s = -N_s^{-1} N_p \dot{q}_p - N_s^{-1} N_e K_e q_e \quad (5.2)$$

em que, $\dot{q}_e = -K_e q_e$, sendo K_e a matriz de ganhos escolhida positiva definida e q_e é o vetor de erro de fechamento obtido a partir da posição das juntas da cadeia virtual de erro (SIMAS, 2008).

Ao substituir a Equação 5.2 na Equação 5.1 obtém-se:

$$N_e \dot{q}_e + N_e K_e q_e = 0 \quad (5.3)$$

$$\dot{q}_e + K_e q_e = 0 \quad (5.4)$$

Como a matriz de ganhos K_e é positiva definida, o vetor q_e tende a zero à medida que o tempo t tende para o infinito.

As juntas da cadeia de erro são juntas primárias e, como tal, precisam ter seus valores definidos. Ao respeitar a relação definida na Equação 5.4, garante-se que os erros tenderão a zero. No entanto, para calcular os valores de \dot{q}_e segundo esta relação, é necessário calcular o erro atual q_e , o que pode ser feito por meio da cinemática direta por helicoides sucessivos.

Em uma cadeia cinemática fechada, o produto das transforma-

ções homogêneas que representam a posição e a orientação de um determinado elo em relação a si mesmo é igual à matriz identidade. Assim, aplicando a técnica dos helicoides sucessivos tem-se que:

$$\prod_{i=1}^{n_p} [A_p]_i \prod_{i=1}^{n_s} [A_s]_i = I \quad (5.5)$$

As matrizes $[A_p]_i$, com $i = 1, 2, \dots, n_p$, são as matrizes homogêneas correspondentes às juntas primárias e $[A_s]_i$, com $i = 1, 2, \dots, n_s$, são as matrizes homogêneas correspondentes às juntas secundárias (SIMAS, 2008). O erro de fechamento pode ser representado através de uma matriz homogênea E_i da seguinte forma:

$$\left\{ \prod_{i=1}^{n_p} [A_p]_i \prod_{i=1}^{n_s} [A_s]_i \right\} E_i = I \Rightarrow E_i = \left\{ \prod_{i=1}^{n_p} [A_p]_i \prod_{i=1}^{n_s} [A_s]_i \right\}^{-1} = \begin{bmatrix} R_e & p_e \\ 0 & 1 \end{bmatrix} \quad (5.6)$$

onde: $p_e = [p_{ex} \ p_{ey} \ p_{ez}]^T$ é o vetor erro de posição e R_e é a matriz que representa o erro de orientação (SIMAS, 2008).

Da matriz R_e são extraídos os erros de orientação $r_e = [r_{ex} \ r_{ey} \ r_{ez}]$, a partir das rotações sucessivas nos eixos x , y e z atuais.

A composição dos erros de posição e orientação forma o vetor q_e , originário da cadeia de erro 3P3R:

$$q_e = [p_e \ r_e]^T = [p_{ex} \ p_{ey} \ p_{ez} \ r_{ex} \ r_{ey} \ r_{ez}]^T \quad (5.7)$$

Com o valor de q_e definido, pode-se calcular os valores das juntas secundárias pela Equação 5.2.

O método de integração através da adição de cadeias virtuais de Assur como apresentado, resume-se a aplicar esta metodologia para anular os erros de abertura de cadeia. Cabe ressaltar que em qualquer ponto crítico da cadeia é possível adicionar a cadeia virtual para o controle do erro. A descrição do método, para um sistema composto por um robô, pode ser encontrada no trabalhos de Simas (2008), Guenther et al. (2008), Fontan (2007).

5.1.1 Controle do erro de integração em SRC

A metodologia generalizada no Capítulo 3 sobre a programação de SRC trata o sistema como fechado, no qual são adicionadas cadeias virtuais que unem os efetuidores à peça (denominada de CV_i), e uma

cadeia virtual que permite definir o deslocamento da peça com relação a um sistema de coordenadas fixo. Ao se calcular a cinemática de um SRC são utilizadas técnicas de integração numérica, como por exemplo o método de integração de Euler, para determinar a posição das juntas do sistema, e estes provocam a abertura da cadeia.

Assim, visando controlar o erro de abertura da cadeia cinemática em SRC serão adicionadas cadeias virtuais de erro em cada um dos circuitos do SRC. As cadeias virtuais de erro são adicionadas em série entre o efetuador do robô e a cadeia virtual que une a peça ao efetuador, como pode ser visualizado na Figura 26, na qual são adicionadas m cadeias virtuais de erro.

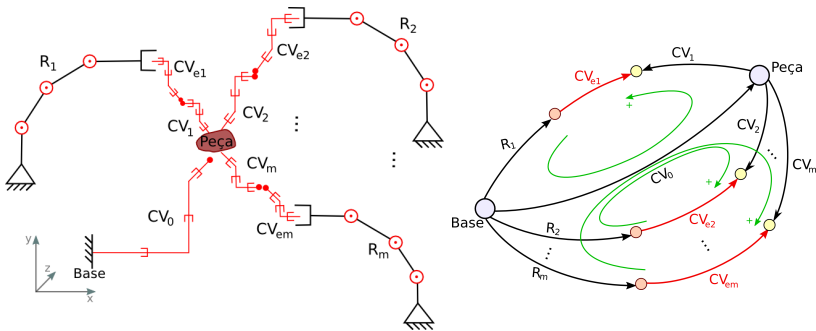


Figura 26: Sistema robótico cooperativo com m cadeias virtuais de erro adicionadas em série.

Para este caso, a matriz de rede N , tem a seguinte estrutura (os elementos destacados em vermelho na matriz fazem referência as cadeias virtuais de erro adicionadas no sistema):

$$N = \begin{bmatrix}
 \begin{bmatrix}
 \hat{\$}_{CV_0} & \hat{\$}_{CV_1} & [-\hat{\$}_{CV_{e1}}] & [-\hat{\$}_{R_1}] \\
 \hat{\$}_{CV_0} & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots \\
 \hat{\$}_{CV_0} & 0 & 0 & 0
 \end{bmatrix} &
 \begin{bmatrix}
 0 & 0 & 0 & \dots \\
 \hat{\$}_{CV_2} & [-\hat{\$}_{CV_{e2}}] & [-\hat{\$}_{R_2}] & \dots \\
 \vdots & \vdots & \vdots & \ddots \\
 0 & 0 & 0 & \dots
 \end{bmatrix} \\
 \dots & 0 & 0 & 0 \\
 \dots & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots \\
 \dots & \hat{\$}_{CV_m} & [-\hat{\$}_{CV_{em}}] & [-\hat{\$}_{R_m}]
 \end{bmatrix} \tag{5.8}$$

Pela Propriedade 1, demonstrado no Capítulo 4, a adição em série de uma cadeia cinemática em uma outra cadeia cinemática fechada, não altera o número total de circuitos desse sistema.

Observa-se que ao adicionar as cadeias virtuais de erro, apenas o número de colunas da matriz N é alterada. Então a generalização descrita nos Capítulos 3 e 4 continua sendo válida após a adição da cadeia cinemática de erro. A matriz da Equação 5.8 é então composta de N_s , N_p e N_e e pode ser substituída na Equação 5.1.

5.2 CONTROLE DE JUNTAS: EVITAMENTO DE COLISÃO

Uma outra característica adicional que pode ser incluída em um SRC é o evitamento de colisão dos elos dos robôs com obstáculos fixos. Na literatura são encontrados alguns autores que propõem técnicas de evitamento de colisão, dos quais destaca-se o método proposto por Simas et al. (2009). Ele desenvolveu um método de evitamento de colisão baseado na adição de cadeias virtuais de Assur aplicado em um sistema com um robô e uma tarefa.

O método proposto por Simas et al. (2009), consiste em adicionar uma cadeia virtual de Assur entre o ponto da cadeia cinemática iminente de colisão e o obstáculo. O objetivo é monitorar a possibilidade de colisão entre um determinado elo do manipulador e o obstáculo, e caso exista possibilidade de colisão, impor uma restrição ao movimento, de tal forma que o obstáculo seja desviado.

O estágio de monitoramento é realizado identificando-se o momento em que as juntas da cadeia virtual adicionada está próxima do limite definido pela posição do obstáculo. As juntas da cadeia virtual que fazem o monitoramento do limite de colisão são tratadas como secundárias, quando aplicado o método de integração cinemática. Assim, é possível identificar o momento em que elas atingem os limites pré-definidos. Quando os valores começam a chegar muito próximos dos limites, é feita uma alteração da equação de restrição cinemática, tornando as juntas virtuais de monitoramento de colisão secundárias em primárias. Em seguida, é imposto o movimento nas juntas de monitoramento da colisão com o objetivo de afastar o elo (do robô) do obstáculo.

Uma consulta mais detalhada sobre o método de evitamento de colisão, em sistemas com um único robô, pode ser realizada nos trabalhos de Campos, Guenther e Martins (2005), Simas et al. (2008), Simas (2008) e Fontan (2007).

5.2.1 Controle de juntas em SRC

Uma representação mais abrangente é proposta para SRC, adicionando cadeias virtuais para controle de juntas (Figura 27). Elas ligam um determinado elo de um robô com o obstáculo. Assim, é possível controlar esse elo através da cadeia virtual adicionada ou determinar o movimento relativo desse elo com um outro elo qualquer de outro robô componente do sistema.

A cadeia virtual de controle CV_{ci} de junta é adicionada em paralelo, com início no sistema de coordenadas fixo e final na junta a ser controlada. Nesta construção, a cadeia de controle precisa ter sua origem em um ponto fixo com relação ao sistema de coordenadas. Alternativamente, pode-se ligar a cadeia virtual de controle de junta à origem do sistema de coordenadas fixo por uma cadeia de posicionamento 3P3R, sendo então possível variar a posição da cadeia de controle.

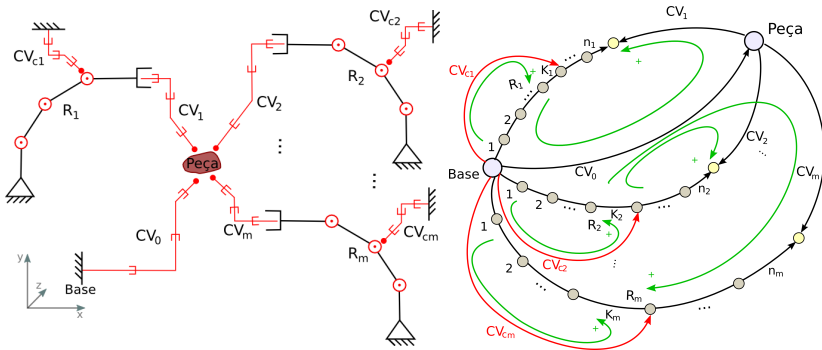


Figura 27: Representação de um SRC com cadeias virtuais para controle de juntas.

Para controlar uma determinada junta do robô, são adicionadas as cadeias virtuais (indicadas por CV_{ci}) relacionando um determinado elo ao sistema de coordenadas fixo. Este controle pode servir para monitorar o movimento da junta instantaneamente e verificar se ela está próxima a algum de seus limites ou próxima a um determinado ponto no espaço, no qual deseja-se verificar a distância entre este ponto e a junta. A adição de cadeias virtuais de controle só podem ser aplicadas em sistemas redundantes.

A matriz de circuitos B é dada por (destacado em vermelho tem-se os elementos referentes a cadeia virtuais de controle de juntas):

$$B = \begin{matrix} & & CV_0 & CV_1 & R_1 & CV_2 & R_2 & \dots \\ M1 & \left[\begin{array}{cccccc} [1]_{1 \times 6} & [1]_{1 \times 6} & [-1]_{1 \times n_1} & 0 & 0 & \dots \\ [1]_{1 \times 6} & 0 & 0 & [1]_{1 \times 6} & [-1]_{1 \times n_2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ Mm & [1]_{1 \times 6} & 0 & 0 & 0 & \dots \\ Mc1 & 0 & 0 & [\beta_1]_{1 \times n_1} & 0 & \dots \\ Mc2 & 0 & 0 & 0 & 0 & [\beta_2]_{1 \times n_2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ Mcm & 0 & 0 & 0 & 0 & 0 & \dots \end{array} \right. \\ \dots & CV_m & R_m & CV_{c1} & CV_{c2} & \dots & CV_{cm} \\ \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & [1]_{1 \times 6} & [-1]_{1 \times n_m} & 0 & 0 & \dots & 0 \\ \dots & 0 & 0 & [1]_{1 \times 6} & 0 & \dots & 0 \\ \dots & 0 & 0 & 0 & [1]_{1 \times 6} & \dots & 0 \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & 0 & [\beta_m]_{1 \times n_m} & 0 & 0 & \dots & [1]_{1 \times 6} \end{matrix} \quad (5.9)$$

em que M_{ci} indica a malha formada pelas cadeias virtuais que unem um elo do manipulador, que se deseja controlar, ao sistema de coordenadas fixo. As cadeias virtuais CV_{ci} unem o elo do robô ao sistema de coordenadas fixo.

A adição de cadeias virtuais para controle e monitoramento de juntas é realizado seguindo a Propriedade 2 de adição em paralelo de subgrafos. Conforme visto, a adição de cadeias virtuais em paralelo alteram o número de circuitos independentes do grafo e cada adição acrescenta um circuito novo ao sistema. Devido a este fato, a matriz N tem o seu número de linhas e colunas acrescidas conforme a adição de novas cadeias.

As submatrizes β_i são referentes aos elos presentes na malha que contém as cadeias virtuais de controle das juntas para tarefas secun-

dárias. A sua composição com relação a presença de arestas na malha é dependente da escolha da junta a ser controlada, por exemplo, caso seja desejado controlar a junta K_i do robô i com $i = 1, \dots, m$, em que m é o número de robôs que compõe o sistema, então, a submatriz β_i é:

$$\beta_i = \begin{bmatrix} 1 & 2 & \dots & k_i & k_{i+1} & \dots & n \\ -1 & -1 & \dots & -1 & 0 & \dots & 0 \end{bmatrix} \quad (5.10)$$

É possível ainda estabelecer restrições entre quaisquer dois elos de dois robôs distintos utilizando a mesma estrutura de cadeia virtual proposta na Figura 27, ligando a junta K_i do i -ésimo robô com a junta K_j do j -ésimo robô através das cadeias virtuais de restrições CV_{ci} e CV_{cj} . A Figura 28 ilustra um exemplo de controle de movimento relativo entre dois elos, utilizando a mesma estrutura para controle de um determinado elo com relação ao obstáculo.

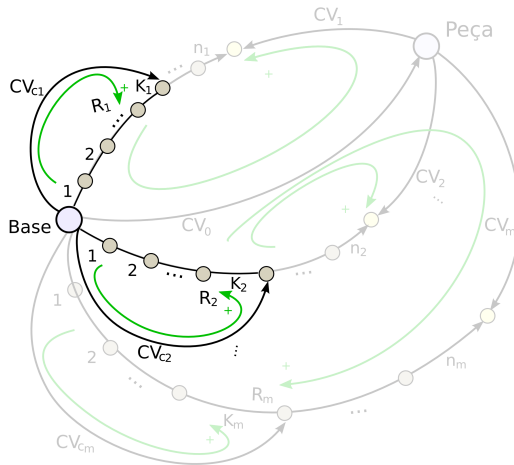


Figura 28: Representação por grafos de um SRC no qual se evidencia o relacionamento entre dois elos quaisquer de robôs distintos.

A metodologia apresentada de evitamento de colisão só pode ser aplicada quando o sistema apresentar algum grau de redundância (Ver Apêndice E).

Pela Propriedade 2, a adição de cadeias virtuais em paralelo, geram novas malhas, e assim, provocam um aumento no número de circuitos independentes do sistema. Devido a esse fato, a matriz de

rede N apresenta um acréscimo no número de linhas e colunas. Este caso, pode ser tratado como uma versão mais ampla da generalização proposta nos Capítulos 3 e 4.

Ao se adicionar uma cadeia virtual de evitamento de colisão deve ser considerado também o problema da abertura da cadeia quando utilizado algum método de integração. Assim, a essa nova malha gerada no sistema pode ser adicionada uma cadeia virtual de erro.

5.3 DESLOCAMENTOS RELATIVOS DAS BASES DOS ROBÔS COM O SISTEMA DE COORDENADAS FIXO

Uma outra característica adicional é a possibilidade de considerar que os robôs não têm suas bases fixas com relação a um sistema de coordenadas fixo, ou seja, a base do robô está em movimento. Algum agente externo, que não pertence ao robô, realiza esse movimento relativo, um exemplo seria o robô sobre alguma plataforma móvel, sobre esteiras. O grafo do SRC com essa característica adicional é ilustrado na Figura 29.

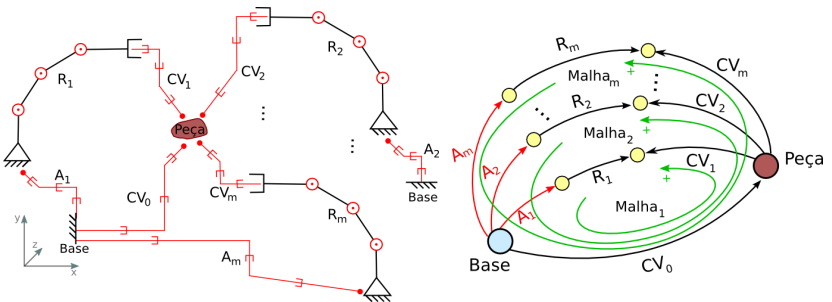


Figura 29: SRC com o agente externo.

A matriz N que representa o sistema completo seria como a Equação 5.11 (é destacado em vermelho os elementos da matriz que fazem parte da cadeia virtual adicionada para impor movimento na base dos robôs).

$$N = \begin{bmatrix}
[\hat{\$}_{CV_0}] & [\hat{\$}_{CV_1}] & [-\hat{\$}_{R_1}] & [-\hat{\$}_{A_1}] & 0 & 0 & 0 & \cdots \\
[\hat{\$}_{CV_0}] & 0 & 0 & 0 & [\hat{\$}_{CV_2}] & [-\hat{\$}_{R_2}] & [-\hat{\$}_{A_2}] & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
[\hat{\$}_{CV_0}] & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & & & & \\
\cdots & 0 & 0 & 0 & & & & \\
\ddots & \vdots & \vdots & \vdots & & & & \\
\cdots & [\hat{\$}_{CV_m}] & [-\hat{\$}_{R_m}] & [-\hat{\$}_{A_m}] & & & &
\end{bmatrix} \quad (5.11)$$

em que $\hat{\$}_{A_i}$ representa a matriz contendo os helicoides normalizados referente ao agente externo A_i associado.

Esse caso é apoiado pela Propriedade 1, demonstrado no Capítulo 4, no qual, são adicionadas em série as cadeias cinemáticas que representam o agente externo na cadeia cinemática do sistema. Assim, essa adição não altera o número total de circuitos independentes do grafo.

Na matriz N , esses agentes são destacados através da letra A_i e podem conter um número qualquer de juntas, i representa o robô no qual ele está associado, com $0 < i < m$, e m é o número de robôs componentes do sistema.

A matriz N está descrita da forma mais geral possível, supondo que todos os robôs estão sobre algum agente. Em um SRC, pode acontecer de apenas um dos robôs estar sobre a ação de um agente, neste caso, apenas o agente referente a esse caso é incluído na matriz N .

5.4 CONCLUSÕES

Neste capítulo foram apresentados três casos especiais que podem ser tratados na generalização da programação de SRC: a adição das cadeias virtuais de erro, a adição das cadeias virtuais de controle de juntas e o deslocamento relativo entre a base dos robôs e o sistema de coordenadas fixo. De certa forma, isto demonstra a flexibilidade da generalização proposta, já que além dessas três características adicionais, outros aspectos podem ser implementados na programação de SRC.

Pode-se incluir nesta generalização robôs com diferentes estruturas, incluindo-se, por exemplo, robôs móveis, robôs paralelos, entre outros. Neste caso, os movimentos dos robôs podem ser representados

por uma cadeia cinemática com deslocamento, em x , y , z , mais a rotação em torno dos eixos, da mesma forma como é feito com a característica adicional de deslocamento relativo da base.

As características adicionais são obtidas a partir da adição de cadeias. A adição de cadeias é sempre realizada em paralelo ou em série, seguindo os resultados obtidos nas Propriedades 1 e 2 do Capítulo 4.

Com a exposição de todos esses casos, o sistema resultante completo se torna complexo, com muitas malhas, vértices e arestas. O que pretende-se nessa tese é mostrar como abordar cada um desses casos e como ele é representado através de um grafo e de uma matriz (matriz de rede N). Quando essa metodologia for aplicada em um sistema real, *a priori*, já são conhecidos os principais componentes do sistema, como os robôs e as tarefas. Assim, nem sempre é necessário incluir todos os casos apresentados neste capítulo em todos os robôs do SRC. A aplicação de uma característica vai depender da necessidade de cada sistema.

6 SIMULAÇÕES DE SRC

A partir da metodologia abordada nos Capítulos 3 e 4, foi desenvolvido um programa que permite simular várias configurações de SRC. O programa é flexível e auxilia no cálculo da cinemática de sistemas cooperativos com um número arbitrário de tarefas e robôs.

Neste capítulo são descritas três simulações com SRC, a primeira tem como finalidade simular uma tarefa sendo realizada pelos robôs Rhinos XR-4 e Scara e em seguida aplicar os resultados em um sistema real (apresentado no Capítulo 7). Serão tratados três casos distintos nessa simulação: no Caso 1, o manipulador irá executar a tarefa sozinho, não existindo a cooperação; no Caso 2 os manipuladores realizarão suas tarefas, uma tarefa é designado para cada um deles; e no Caso 3, um robô terá a função de auxiliar o outro em sua tarefa, posicionando a peça enquanto o outro executa alguma operação sobre ela.

A segunda simulação é composta por quatro robôs em cooperação e um conjunto de tarefas a serem executadas. É apresentado um sistema de referência e neste sistema são realizadas alterações em alguns parâmetros e analisada a influência que elas provocam no resultado final da cinemática.

Na terceira simulação é descrito um sistema cooperativo realizando algumas tarefas. Neste caso, foca-se a exemplificação da aplicação de uma característica adicional ao sistema (comentado na seção 5.3 do Capítulo 5), na qual é simulada a base de um dos robôs como sendo móvel.

6.1 DESENVOLVIMENTO DE UM PROGRAMA PARA CINEMÁTICA DE SRC

Utilizando-se a metodologia proposta, foi desenvolvido um programa para o cálculo da cinemática, simulação e visualização do processo de execução de tarefas por SRC.

Os algoritmos foram implementados utilizando-se a linguagem de programação Python (PAVIOT, 2011) e GNU Octave (EATON, 2011).

O GNU Octave é uma linguagem interpretada de alto nível, com foco inicial em cálculos numéricos. Ela fornece soluções numéricas para problemas lineares e não lineares, e para a execução de outros experimentos numéricos. O Octave é normalmente utilizado através de uma interface de linha de comando interativa, mas também pode ser usado

para escrever programas não interativos. A linguagem Octave é muito similar ao MatLab[®] e seu código fonte é distribuído livremente sob os termos da Licença Pública Geral - GPL (EATON, 2011).

O programa desenvolvido no Octave carrega os dados dos robôs, tarefas e ambientes a partir de arquivos padronizados de texto formatados. Com estes dados, o programa executa os cálculos numéricos de forma que os robôs especificados executem a tarefa dentro da configuração do ambiente. O resultado final é um arquivo que segue a mesma formatação dos arquivos de entrada, incluindo as trajetórias de juntas de cada robô, além dos dados de ambiente e tarefa já fornecidos.

O Python OCC é um *framework* desenvolvido para CAD/CAE 3D baseada na linguagem de programação Python. Ele fornece características como operações geométricas e topológicas, troca de dados (importando e exportando os arquivos nos formatos STEP, IGES, STL), malhas 2D e 3D, simulação de corpos rígidos e modelamento paramétrico. A biblioteca Python OCC e os exemplos são licenciados sob os termos da Licença Pública Geral Menor V3 - LGPL (PAVIOT, 2011).

Neste trabalho, o Python OCC tem como funcionalidade principal permitir a importação de arquivos STEP com as características dos robôs, das peças e do ambiente. Além disso, através dele é possível visualizar os movimentos relativos entre estes diferentes componentes (robôs - elos e juntas, peças e componentes do ambiente). O programa desenvolvido é utilizado como um simulador do comportamento dos robôs. Ele recebe as trajetórias de juntas e as tarefas fornecidas pelo arquivo de saída do Octave, calcula e apresenta os movimentos referentes a essas trajetórias.

O programa está dividido em dois módulos denominados de módulo de resolução cinemática de SRC e o módulo de simulação e visualização. A Figura 30 ilustra o diagrama de classes compreendendo esses dois módulos.

No módulo de resolução cinemática foram implementadas as seguintes funcionalidades:

- cálculo do helicóide resultante através da técnica dos helicóides sucessivos (implementação analítica e numérica);
- composição dos helicóides sucessivos resultantes de cada subcadeia;
- construção das cadeias virtuais de Assur;
- método de Davies;

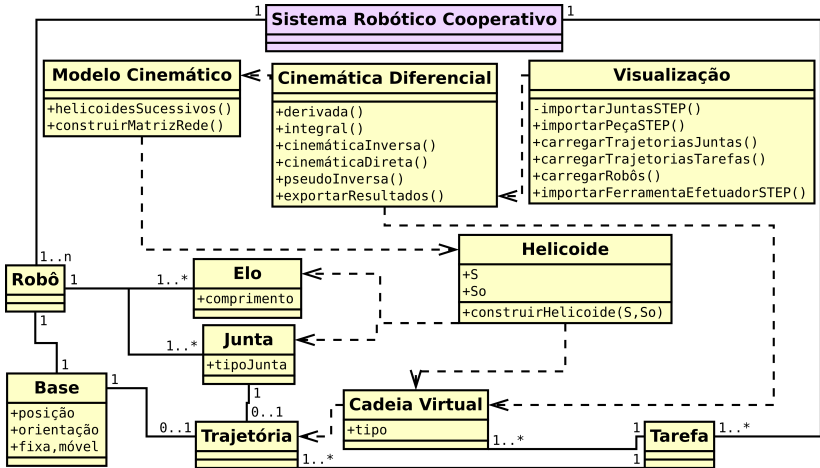


Figura 30: Diagrama de classes dos programas implementados.

- método de Davies para SRC redundantes;
- interpolação da trajetória referente a tarefa;
- integração das velocidades das juntas (resultando nas posições de cada uma das juntas);
- cinemática para a característica adicional → deslocamentos relativos dos robôs com o sistema de coordenadas fixo.

Para o módulo de simulação e visualização foram implementadas as seguintes funcionalidades:

- cinemática direta;
- importar o modelo CAD dos robôs, ferramentas, peças e ambientes em formato de arquivo STEP;
- visualização das trajetórias;
- simulação dos movimentos em tempo real.

Para a construção e simulação dos SRCs no programa é fornecido um conjunto de recursos que permitem a configuração do sistema e

obtenção de resultados com uma simulação próxima a realidade. O recursos são os seguintes:

- banco de dados com os arquivos CAD dos robôs em formato STEP. Os arquivos originais foram obtidos no endereço eletrônico da fabricante ABB (2011);
- banco de dados da cinemática dos robôs, contendo os helicoides normalizados pré-calculados analiticamente;
- banco de dados de peças e efetadores em formato STEP, permitindo a inclusão de novas peças e ferramentas;
- pacote de exemplos incluindo os casos estudados na tese, além de sistemas com um, dois, três, quatro e oito robôs, com características adicionais do movimento relativo, executando tarefas sobre a peça e seu deslocamento, resolução da redundância por pseudo-inversa.

Como foi visto no Capítulo 4, a matriz de rede N tem dimensões proporcionais ao número de robôs. A etapa de maior complexidade do algoritmo desenvolvido é a inversão da matriz N_s , que na pior das hipóteses tem complexidade $O(m^3)$. O algoritmo é repetido para cada ponto da tarefa, de forma que a complexidade final do programa é $O(p \cdot m^3)$, onde p é o número de pontos da tarefa e m é o número de robôs.

O programa desenvolvido e seu código fonte está disponibilizado no endereço eletrônico: www.labcadcam.ufsc.br/~cris.tonetto sob a licença GPL (GPL, 2007). A seguir são explorados alguns resultados obtidos dos SRCs que foram simulados no programa desenvolvido. Alguns vídeos das simulações estão disponíveis no canal do laboratório no endereço: <http://www.youtube.com/user/UFSCLabCADCAM>.

6.2 SIMULAÇÃO DOS ROBÔS RHINOS: XR4 E SCARA

Com a finalidade de mostrar a flexibilidade da generalização proposta, foi realizada a simulação com dois robôs da Rhino Robotics Ltd (RHINO, 2010). Estes robôs estão localizados para pesquisa no Laboratório de Automação Industrial (LAI, 2010) do Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina - UFSC.

Os robôs estão dispostos conforme as Figuras 31 (imagem fotográfica do sistema instalado) e Figura 32 (imagem do simulador). As



Figura 31: Robô Rhino XR-4 (esquerda); Robô Rhino Scara (direita).

dimensões do robô XR-4 são: $a_1 = 16,5\text{cm}$, $b_1 = 12\text{cm}$, $c_1 = 22,86\text{cm}$, $d_1 = 22,86\text{cm}$ e $e_1 = 15,87\text{cm}$, e todas as suas juntas são rotativas. O robô Scara tem as seguintes dimensões: $a_2 = 36,5\text{cm}$, $b_2 = 22,86\text{cm}$, $c_2 = 22,86\text{cm}$ e $d_2 = 10\text{cm}$, os seus elos são conectados por juntas rotativas e por uma junta p_3 prismática. O robô XR-4 está montado sobre uma mesa de coordenadas cartesianas XY de aproximadamente 10cm de altura, porém, os movimentos que essa mesa pode executar, não serão considerados como graus de liberdade do robô neste experimento.

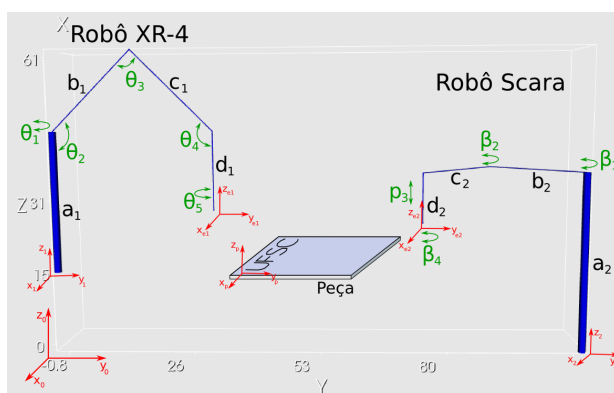


Figura 32: Robôs participantes do sistema com as devidas identificações.

O sistema de coordenadas fixo $(x_0; y_0; z_0)$ é definido na origem do sistema de coordenadas do SRC, conforme Figura 32. A localização da base do robô XR-4 com relação ao sistema de coordenadas fixo $(x_0; y_0; z_0)$ é $D_1 = (Dx_1; Dy_1; Dz_1) = (0; 0; 16,45)$ e para o robô Scara é $D_2 = (Dx_2; Dy_2; Dz_2) = (0; 106,7; 0)$. Os robôs tem suas bases fixas e estão a uma distância de 106,7cm entre si. A Figura 32 ilustra os robôs com a identificação dos ângulos, elos e sistemas de coordenadas fixos.

Para o SRC em questão, são definidos três casos diferentes de especificação das tarefas:

- CASO 1: O efetuador do robô XR-4 executará uma trajetória com o objetivo de escrever a sigla UFSC sobre a peça (Figura 33 (a)). O robô Scara não participará da tarefa.
- CASO 2: O efetuador do robô XR-4 executará a tarefa descrita no Caso 1 e o efetuador do robô Scara seguirá uma trajetória retilínea posicionando a peça (Figura 33 (b)).
- CASO 3: Novamente, o efetuador do robô XR-4 terá a função de realizar a tarefa proposta no Caso 1. O robô Scara terá a função de posicionar a peça, a ele não é imposta nenhuma trajetória pré-definida a ser seguida. Desta forma, o movimento do robô Scara é calculado a fim de auxiliar na execução da tarefa do robô XR-4.

As tarefas e trajetórias descritas nesses casos são ilustrativas. A Figura 33 (a) mostra o caminho que um dos robôs deverá percorrer com relação à origem da peça, que consiste em um caminho que forma a sigla UFSC. A Figura 33 (b) mostra o caminho que a peça deve percorrer, e que precisa ser respeitado pelo movimento do robô posicionador. Este caminho é uma reta $r(t) = (t; 50,8; 25,4)$, com $0 < t < 20s$ e unidade em polegadas. Nota-se que esta movimentação da peça terá o sentido contrário com relação ao deslocamento do outro robô, que escreverá a sigla UFSC.

A Tabela 2 descreve os valores de s e s_o para cada uma das juntas dos robôs, partindo de uma configuração inicial do sistema (TONETTO; DIAS, 2009, 2010). Lembrando que a localização da base do robô XR-4 com relação ao sistema de coordenadas fixo é indicada por $(Dx_1; Dy_1; Dz_1)$ e a do robô Scara é $(Dx_2; Dy_2; Dz_2)$. Para os valores de s e s_o , é então aplicada a técnica dos helicoides sucessivos para determinar o helicóide resultante (Apêndice B.1).

Seguindo a metodologia proposta para o cálculo da cinemática de SRC, são adicionadas as cadeias virtuais ao sistema (como represen-

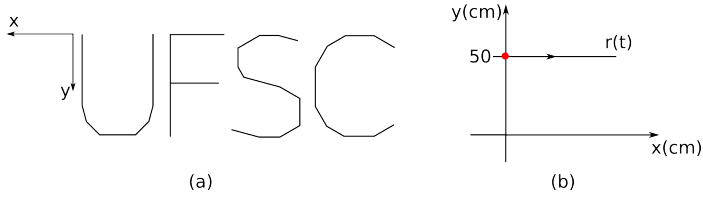


Figura 33: Trajetória a ser seguida: (a) pelo efetuador do robô XR-4 com relação à referência da peça; (b) pela peça com relação ao sistema de coordenadas fixo.

Tabela 2: Valores de s e s_o para as juntas do manipulador com relação a uma configuração de referência.

<i>Juntas</i>	s	s_o
θ_1	(0; 0; 1)	$(Dx_1; Dy_1; Dz_1 + a_1)$
θ_2	(1; 0; 0)	$(Dx_1, Dy_1; Dz_1 + a_1 + b_1)$
θ_3	(1; 0; 0)	$(Dx_1; Dy_1; Dz_1 + a_1 + b_1 + c_1)$
θ_4	(1; 0; 0)	$(Dx_1; Dy_1; Dz_1 + a_1 + b_1 + c_1 + d_1)$
θ_5	(0; 0; 1)	$(Dx_1; Dy_1; Dz_1 + a_1 + b_1 + c_1 + d_1 + e_1)$
β_1	(0; 0; 1)	$(Dx_2; Dy_2; Dz_2 + a_2)$
β_2	(0; 0; 1)	$(Dx_2; Dy_2 + b_2; Dz_2 + a_2)$
P_3	(0; 0; 1)	$(Dx_2; Dy_2 + b_2 + c_2; Dz_2 + a_2)$
β_4	(0; 0; 1)	$(Dx_2; Dy_2 + b_2 + c_2; Dz_2 + a_2)$

tadas na Figura 34). Para este sistema foram adicionadas três cadeias virtuais (indicadas por CV_0 , CV_1 e CV_2) do tipo 3P3R, ou seja, com três juntas prismáticas e três juntas rotativas. As cadeias virtuais CV_1 e CV_2 monitoram o movimento que cada um dos efetuadores dos robôs irá realizar sobre a peça. O movimento que a peça irá realizar com relação ao sistema de coordenadas fixo é mostrado pela cadeia CV_0 . Além disso, a cadeia cinemática que representa o sistema é agora tratada como fechada e podem ser aplicados neste sistema os métodos disponíveis para o cálculo da cinemática em sistemas fechados.

A cadeia cinemática fechada do SRC é representada por um grafo. Assim, utilizando a teoria de grafos é feita uma representação abstrata do sistema. O novo modelo robótico, com robôs reais e cadeias virtuais, é ilustrado na Figura 35, onde cada aresta do grafo é

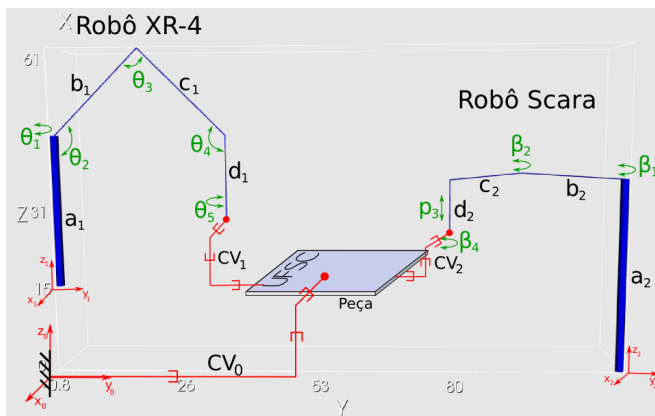


Figura 34: Representação do sistema com a adição das cadeias virtuais de Assur e da tarefa.

direcionada e representa uma junta do SRC monitorado. No lado direito da Figura 35 é mostrada uma simplificação desse grafo, omitindo os vértices referentes aos elos dos robôs.

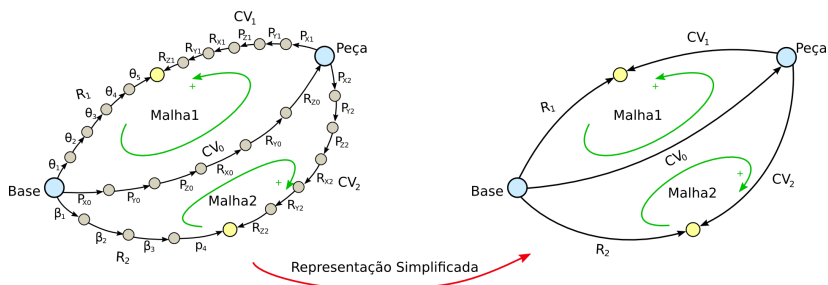


Figura 35: Grafo do SRC e sua representação simplificada.

Os grafos permitem gerar uma matriz de circuitos (B), que indica a presença de cada aresta na malha do grafo. Agrupando as arestas conforme a simplificação proposta do grafo (Figura 35), pode-se obter

a seguinte matriz de incidência:

$$B = \begin{matrix} & CV_0 & CV_1 & R_1 & CV_2 & R_2 \\ \begin{matrix} malha1 \\ malha2 \end{matrix} & \begin{bmatrix} [1]_{1 \times 6} \\ [1]_{1 \times 6} \end{bmatrix} & \begin{bmatrix} [1]_{1 \times 6} \\ 0 \end{bmatrix} & \begin{bmatrix} [-1]_{1 \times 6} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ [1]_{1 \times 6} \end{bmatrix} & \begin{bmatrix} 0 \\ [-1]_{1 \times 6} \end{bmatrix} \end{matrix} \quad (6.1)$$

É, então, gerada a matriz de helicoides normalizados D que está indicada na Equação 6.2. Note que a ordem de adição dos helicoides normalizados na matriz D deve ser a mesma adotada na matriz B :

$$D = [[\hat{\$}_{CV_0}]_{6 \times 6} \quad [\hat{\$}_{CV_1}]_{6 \times 6} \quad [\hat{\$}_{R_1}]_{6 \times 6} \quad [\hat{\$}_{CV_2}]_{6 \times 6} \quad [\hat{\$}_{R_2}]_{6 \times 6}] \quad (6.2)$$

A matriz de rede N é obtida multiplicando-se a matriz de helicoides D pela matriz diagonal de B :

$$N = D \text{diag}\{B_i\} \quad (6.3)$$

onde: B_i é a matriz formada pela linha i da matriz B .

Então, a matriz de rede N é dada pela equação:

$$N = \begin{bmatrix} [\hat{\$}_{CV_0}]_{6 \times 6} & [\hat{\$}_{CV_1}]_{6 \times 6} & [-\hat{\$}_{R_1}]_{6 \times 6} & 0 & 0 \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & 0 & [\hat{\$}_{CV_2}]_{6 \times 6} & [-\hat{\$}_{R_2}]_{6 \times 6} \end{bmatrix} \quad (6.4)$$

Para o Caso 1, em que o manipulador XR-4 irá executar a tarefa sozinho, as matrizes N_s , \dot{q}_s , N_p e \dot{q}_p são:

$$N_s = \begin{bmatrix} -\hat{\$}_{R_1} & \hat{\$}_{CV_{1Ry}} \end{bmatrix}_{6 \times 6} \quad (6.5)$$

$$\dot{q}_s = [\dot{q}_{R_1} \quad \dot{q}_{CV_{1Ry}}]_{6 \times 1}^T \quad (6.6)$$

$$N_p = \begin{bmatrix} \hat{\$}_{CV_0} & \hat{\$}_{CV_{1Rz}} & \hat{\$}_{CV_{1Rx}} & \hat{\$}_{CV_{1Pz}} & \hat{\$}_{CV_{1Py}} & \hat{\$}_{CV_{1Px}} \end{bmatrix}_{6 \times 11} \quad (6.7)$$

$$\dot{q}_p = [\dot{q}_{CV_0} \quad \dot{q}_{CV_{1Rz}} \quad \dot{q}_{CV_{1Rx}} \quad \dot{q}_{CV_{1Pz}} \quad \dot{q}_{CV_{1Py}} \quad \dot{q}_{CV_{1Px}}]_{11 \times 1}^T \quad (6.8)$$

Substituindo as matrizes N_s , N_q e \dot{q}_p na Equação 6.9, têm-se as velocidades das juntas secundárias em função das juntas primárias. Assim, a velocidade das juntas dos robôs é obtida a partir da especificação da tarefa, através da adição das cadeias virtuais CV_0 e CV_1 .

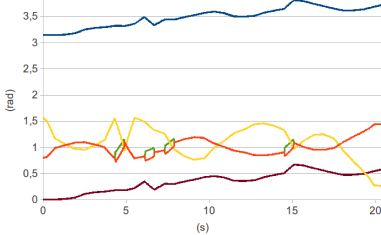
$$\dot{q}_s = -N_s^{-1} N_p \dot{q}_p \quad (6.9)$$

Os vetores \hat{q}_s e \hat{q}_p são vetores que representam a magnitude das velocidades das juntas, no qual, \hat{q}_s corresponde às velocidades das juntas desconhecidas, a serem determinadas e \hat{q}_p é o vetor que contém as velocidades conhecidas das juntas. O vetor \hat{q}_p é dado pela derivada das trajetórias de cada uma das tarefas.

Foi realizada a simulação com o robô XR-4 executando somente sua tarefa (Caso 1). A Figura 36 ilustra as posições (em radianos) e velocidades (em radianos por segundo) das juntas obtidas como resultado da simulação com o decorrer do tempo. O gráfico de posição das juntas mostra algumas inflexões que são mais perceptíveis no gráfico de velocidade. Esses pontos de inflexões são próprios da tarefa a ser executada pelo robô, elas acontecem no momento em que o robô está passando de uma letra para outra durante a escrita do acrônimo UFSC.

Caso 1:

XR-4
Posição



Velocidade

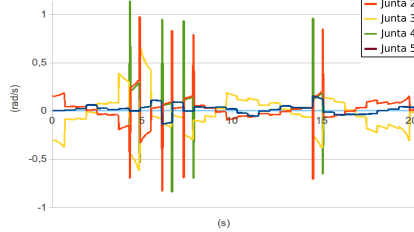


Figura 36: Posição e velocidade das juntas do robô XR-4 escrevendo o acrônimo UFSC.

Para o Caso 2, o efetuador do robô XR-4 tem como tarefa escrever o acrônimo UFSC e o efetuador do robô Scara deve posicionar a peça seguindo uma trajetória retilínea. Neste caso, os robôs estão cooperando no sentido de executar suas tarefas em conjunto, deslocar a peça e fazer a marcação das letras sobre a peça. Então, as matrizes N_s , \hat{q}_s , N_p e \hat{q}_p para o Caso 2 são:

$$N_s = \begin{bmatrix} -\hat{\$R}_1 & 0 & \hat{\$CV}_{1Ry} & \hat{\$CV}_{0Ry} & \hat{\$CV}_{0Rx} \\ 0 & -\hat{\$R}_2 & 0 & \hat{\$CV}_{0Ry} & \hat{\$CV}_{0Rx} \end{bmatrix}_{12 \times 12} \quad (6.10)$$

$$\hat{q}_s = \begin{bmatrix} \hat{q}_{R1} & \hat{q}_{R2} & \hat{q}_{CV_{1Ry}} & \hat{q}_{CV_{0Ry}} & \hat{q}_{CV_{0Rx}} \end{bmatrix}^T_{12 \times 1} \quad (6.11)$$

$$N_p = \begin{bmatrix} [\hat{\$}_{CV_0p}] & [\hat{\$}_{CV_1p}] & [0] \\ [\hat{\$}_{CV_0p}] & [0] & [\hat{\$}_{CV_2}] \end{bmatrix}_{12 \times 15} \quad (6.12)$$

$$\dot{q}_p = \begin{bmatrix} [\dot{q}_{CV_0p}] & [\dot{q}_{CV_1p}] & [\dot{q}_{CV_2}] \end{bmatrix}_{15 \times 1}^T \quad (6.13)$$

onde, as submatrizes $\hat{\$}_{CV_0p}$, $\hat{\$}_{CV_1p}$, \dot{q}_{CV_0p} e \dot{q}_{CV_1p} são dadas por:

$$\hat{\$}_{CV_0p} = \begin{bmatrix} \hat{\$}_{CV_0Rz} & \hat{\$}_{CV_0Px} & \hat{\$}_{CV_0Py} & \hat{\$}_{CV_0Pz} \end{bmatrix}_{6 \times 4} \quad (6.14)$$

$$\hat{\$}_{CV_1p} = \begin{bmatrix} \hat{\$}_{CV_1Rz} & \hat{\$}_{CV_1Rx} & \hat{\$}_{CV_1Pz} & \hat{\$}_{CV_1Py} & \hat{\$}_{CV_1Px} \end{bmatrix}_{6 \times 5} \quad (6.15)$$

$$\dot{q}_{CV_0p} = \begin{bmatrix} \dot{q}_{CV_0Rz} & \dot{q}_{CV_0Px} & \dot{q}_{CV_0Py} & \dot{q}_{CV_0Pz} \end{bmatrix}_{4 \times 1}^T \quad (6.16)$$

$$\dot{q}_{CV_1p} = \begin{bmatrix} \dot{q}_{CV_1Rz} & \dot{q}_{CV_1Rx} & \dot{q}_{CV_1Pz} & \dot{q}_{CV_1Py} & \dot{q}_{CV_1Px} \end{bmatrix}_{5 \times 1}^T \quad (6.17)$$

Os resultados foram obtidos substituindo as matrizes N_s , \dot{q}_s , N_p e \dot{q}_p na Equação 6.9. A Figura 37, ilustra as posições e velocidades de cada uma das juntas dos robôs em função do tempo da tarefa. As posições das juntas rotativas são medidas em radianos e a posição da junta prismática é medida em milímetro. Os robôs executaram suas tarefas satisfazendo os limites de junta impostos pelo fabricante. A junta prismática do robô Scara (junta 3 do gráfico apresentado na Figura 37) não foi utilizada durante a execução da tarefa, ou seja, não foi imposto nenhum movimento que precisasse utilizá-la, embora ela estivesse disponível, se fosse necessário.

As juntas do robô XR-4 tiveram comportamento muito semelhante nos Casos 1 e 2. Isso se deve ao fato do robô Scara, presente no Caso 2, precisar executar uma tarefa de movimentação da peça, que não necessariamente auxilia o robô XR-4.

Para que o robô Scara pudesse ser utilizado para auxílio do XR-4, seria necessário que seus movimentos não fossem restritos por uma tarefa, estando assim disponíveis para otimizar a utilização do sistema. No Caso 3, ao efetuador do robô Scara compete a tarefa de posicionar a peça estando disponível para que seu movimento seja realizado com a função de ajudar o robô XR-4 na execução de sua tarefa.

O que difere este caso do anterior é que, agora, tem-se dois robôs e apenas uma tarefa. Assim, é necessário uma definição diferente para as escolhas das juntas primárias e secundárias. Para este caso as submatrizes referentes a cadeia virtual CV_0 não possuem velocidades conhe-

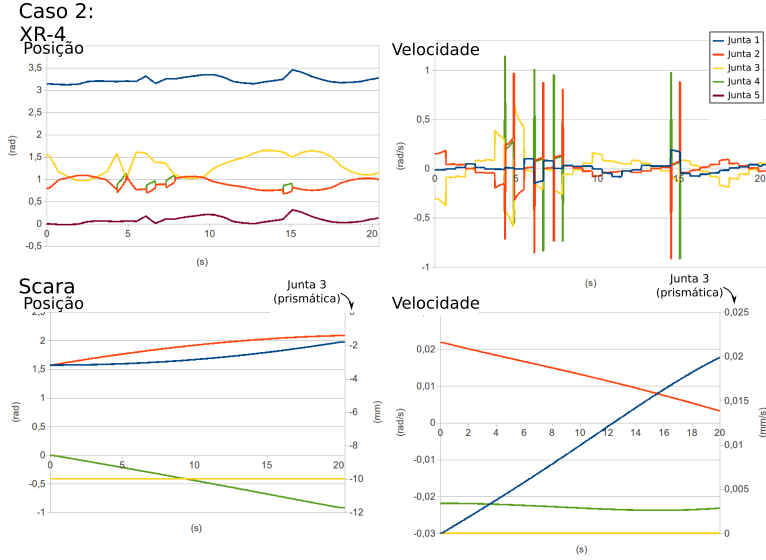


Figura 37: Posição e velocidade das juntas dos robôs XR-4 e Scara.

cidas, fazendo parte da matriz N_s . Isso significa que a posição da peça com relação ao sistema de coordenadas fixo não é pré-determinada, mas obtida como um resultado da cooperação. As matrizes para o Caso 3 estão especificadas nas seguintes equações:

$$N_s = \begin{bmatrix} -\hat{\$R}_1 & 0 & \hat{\$CV}_{1Ry} & \hat{\$CV}_0 \\ 0 & -\hat{\$R}_2 & 0 & \hat{\$CV}_0 \end{bmatrix}_{12 \times 16} \quad (6.18)$$

$$\dot{q}_s = \begin{bmatrix} \dot{q}_{R_1} & \dot{q}_{R_2} & \dot{q}_{CV_{1Ry}} & \dot{q}_{CV_0} \end{bmatrix}_{16 \times 1}^T \quad (6.19)$$

$$N_p = \begin{bmatrix} \hat{\$CV}_{1Rz} & \hat{\$CV}_{1Rx} & \hat{\$CV}_{1Pz} & \hat{\$CV}_{1Py} & \hat{\$CV}_{1Px} & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{\$CV}_2 \end{bmatrix}_{12 \times 11} \quad (6.20)$$

$$\dot{q}_p = \begin{bmatrix} \dot{q}_{CV_{1Rz}} & \dot{q}_{CV_{1Rx}} & \dot{q}_{CV_{1Pz}} & \dot{q}_{CV_{1Py}} & \dot{q}_{CV_{1Px}} & \dot{q}_{CV_2} \end{bmatrix}_{11 \times 1}^T \quad (6.21)$$

Como a matriz N_s resultante é não-quadrada (possui mais colunas que linhas), a Equação 6.9 possui infinitas soluções e então é

necessário utilizar algum critério de otimização ou eliminar as juntas passivas referentes as cadeias virtuais, através do conceito de matriz aniquiladora apresentado por Campos (CAMPOS, 2004). Neste caso, foi utilizado a pseudo-inversa com pesos (SICILIANO et al., 2009), adaptada neste trabalho para receber múltiplos robôs.

$$\dot{q}_s = -WN_s^T (N_sWN_s^T)^{-1}N_p\dot{q}_p \quad (6.22)$$

em que, W é a matriz de pesos dada por:

$$W = \begin{bmatrix} [I]_{12 \times 12} & [0]_{12 \times 6} \\ [0]_{12 \times 6} & [diag\{0.001\}]_{6 \times 6} \end{bmatrix} \quad (6.23)$$

Para o cálculo da cinemática, são separadas as juntas primárias e secundárias conforme a especificação da tarefa, tendo como resultado as matrizes N_s e N_p . Essa diferença está ilustrada na Figura 38, no qual é evidenciada quais são as juntas primárias e quais são as juntas secundárias no grafo para cada caso.

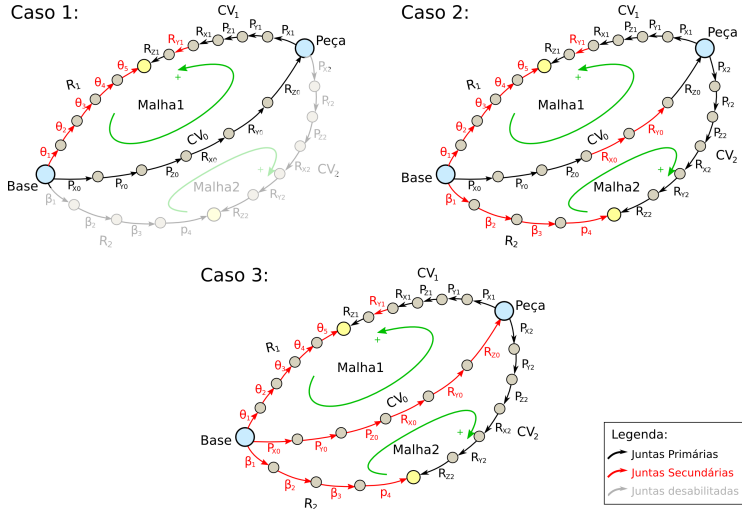


Figura 38: Grafos dos três casos destacando as juntas primárias e secundárias.

Na Figura 38, as juntas destacadas em vermelho devem permanecer na matriz N_s , que fazem referência às juntas com velocidades

desconhecidas e as juntas destacadas em preto são as juntas com velocidades conhecidas e seus helicoides devem formar a matriz N_p . As matrizes \hat{q}_p e \hat{q}_s devem ter seus elementos ordenados conforme as matrizes N_p e N_s , respectivamente.

Na Figura 39 têm-se as posições e velocidades das juntas dos dois robôs executando a tarefa, na qual o efetuador do robô XR-4 realiza uma operação sobre a peça enquanto o efetuador do robô Scara posiciona a peça de forma a reduzir as velocidades das juntas do robô XR-4, obtendo, dessa forma, um melhor proveito da cooperação. Nota-se que os gráficos de posição e velocidade não apresentam curvas tão suaves quando comparado com os gráficos ilustrados no Caso 2 (Figura 37). Por outro lado, a amplitude das curvas de velocidade do robô XR-4 diminuíram consideravelmente, e o robô Scara teve uma participação mais atuante durante no posicionamento da peça para o robô XR-4.

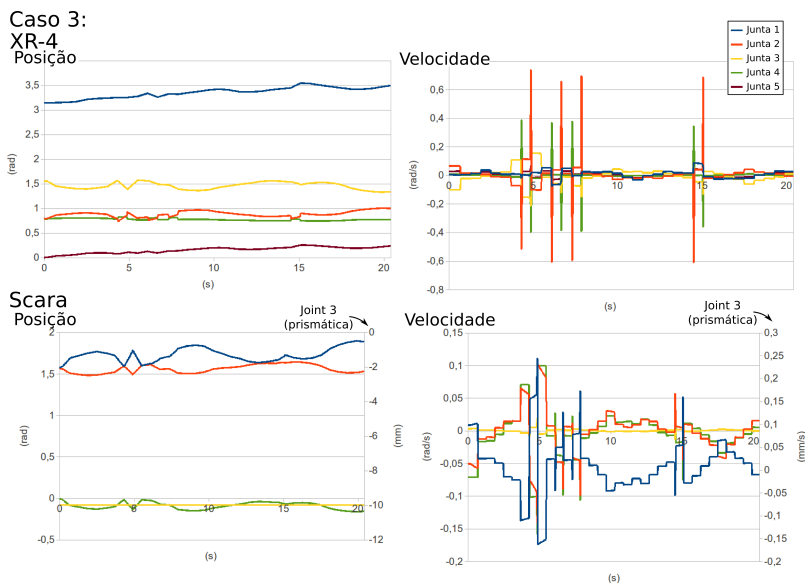


Figura 39: Posição e velocidade das juntas dos robôs XR-4 e Scara trabalhando em cooperação.

Nas Figuras 40 e 41 podem ser verificadas as posições e velocidades das juntas do robô XR-4 para os três casos. A amplitude das velocidades das juntas do robô XR-4 diminuiu consideravelmente quando é

utilizado o robô Scara para auxiliar na execução da tarefa. Por exemplo, analisando as posições e velocidades das juntas nos três casos, é notável a diferença observada na junta 3, considerada o cotovelo do robô XR-4. No sistema do Caso 1, composto apenas pelo robô XR-4, esta junta teve sua velocidade máxima de 0,65 rad/s e com a utilização do robô Scara em cooperação no Caso 3 passou para 0,18 rad/s, aproximadamente.

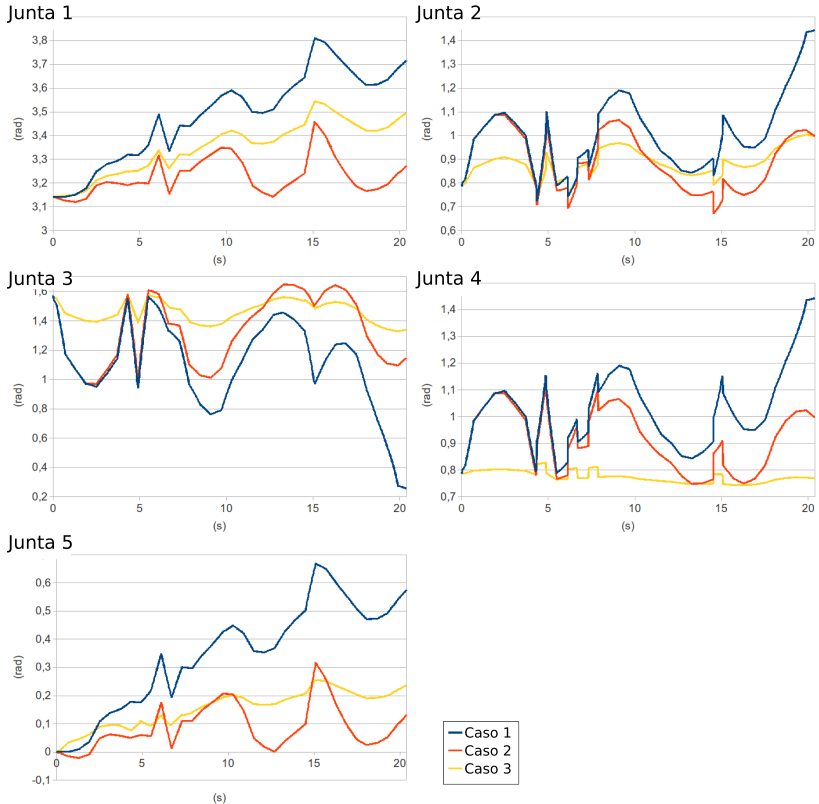


Figura 40: Comparação da posição das juntas do robô XR-4 entre os três casos.

De modo geral, todas as juntas do robô tiveram uma suave redução nas velocidades de juntas do Caso 1 para o Caso 2 e uma redução considerável do Caso 2 para o Caso 3.

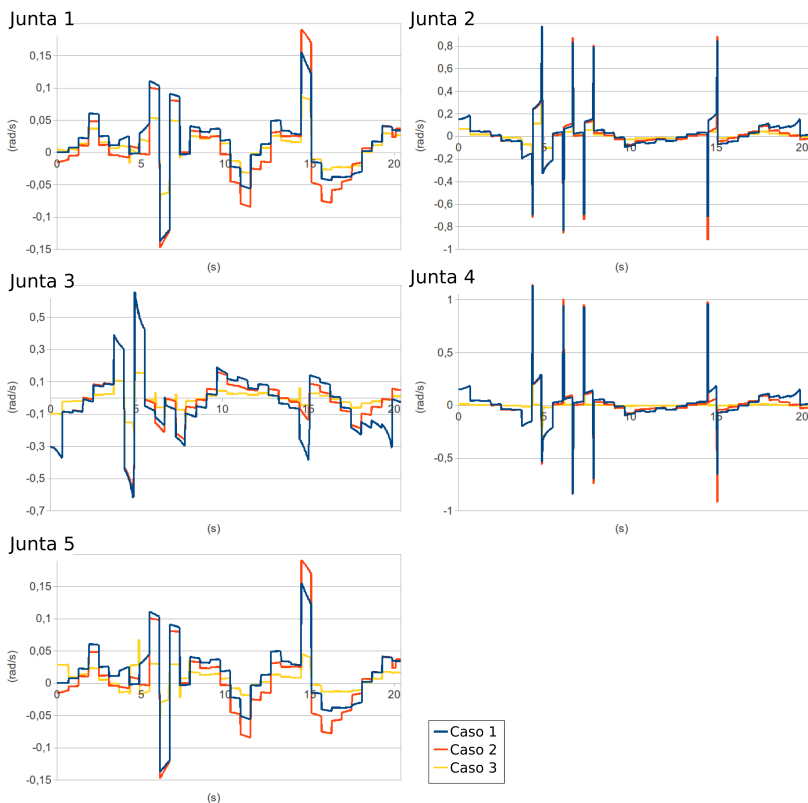


Figura 41: Comparação da velocidade das juntas robô XR-4 entre os três casos.

Para os Casos 2 e 3 os deslocamentos de juntas são realizadas próximos da posição inicial se comparados com o Caso 1, no qual o manipulador realizava a tarefa sozinho e chega mais próximo dos limites do espaço de trabalho. Se fosse definida uma tarefa que ultrapassasse os limites do espaço de trabalho do robô XR-4, seria impossível a sua realização sem utilização da cooperação. Neste sentido, a cooperação mantém os robôs operando próximo da configuração inicial das juntas, tornando viável a realização de tarefas que estejam fora dos limites do espaço de trabalho de um único robô. É possível notar também suavizações nas velocidades das juntas do robô XR-4 no Caso 3 que

apresentou menos transições bruscas.

6.3 SIMULAÇÃO DE QUATRO ROBÔS EM COOPERAÇÃO

A tomada de decisão de usar um SRC depende de vários fatores. Alguns desses fatores são relacionados à tarefa, outros aos robôs disponíveis no local para utilização. Outros fatores que também podem ser considerados são o ambiente, no qual se encontram os robôs, e o arranjo físico dos mesmos no ambiente.

O sistema utilizado neste exemplo é composto por quatro robôs e uma tarefa ilustrativa. Os robôs escolhidos são da fabricante ABB. Neste sistema será realizada uma série de simulações fazendo algumas alterações. As alterações propostas para o SRC fazem referência: a posição física dos robôs, a configuração inicial das juntas dos robôs, a especificação da tarefa aos robôs, entre outros. O experimento proposto servirá para analisar a influência que essas alterações provocam no desempenho da realização da tarefa pelo sistema. E, desta forma, propor um conjunto de critérios para avaliar qual a melhor configuração de SRC para a realização de determinada tarefa. Além disso, esse exemplo visa mostrar a flexibilidade nas simulações que a metodologia proposta exerce sobre um SRC.

Os robôs que compõem o SRC são: ABB IRB 6620, ABB IRB 1600, ABB IRB 140 e ABB IRB 120. As especificações técnicas de cada um são apresentadas nas seguintes Tabelas 3, 4, 5 e 6. Nessas tabelas constam os limites, as velocidades de desempenho e os parâmetros de s e s_o de cada uma das juntas. Essas informações servirão de base no cálculo da programação cinemática dos robôs. Na Tabela 3 do robô ABB IRB 6620, é importante notar que a primeira junta é prismática.

Tabela 3: Especificação do Robô: ABB IRB 6620.

Junta	Limite da Junta	Veloc.	s	s_o
1	1,8 a 33m	3,3m/s	(0; 1; 0)	(0; 0; 0)
2	125° a -125°	90°/s	(0; 1; 0)	(0; 0; 416)
3	70° a -180°	90°/s	(0; 1; 0)	(0; 0; 1371)
4	300° a -300°	150°/s	(1; 0; 0)	(245; 0; 1565)
5	130° a -130°	120°/s	(0; 1; 0)	(880; 0; 1565)
6	300° a -300°	190°/s	(1; 0; 0)	(1076; 0; 1565)

A Figura 42 mostra uma imagem da configuração do sistema e um grafo exibindo a adição das cadeias virtuais de Assur. Para este

Tabela 4: Especificação do Robô: ABB IRB 1600.

Junta	Limite da Junta	Veloc.	s	s_o
1	180° a -180°	150°/s	(0; 0; 1)	(0; 0; 0)
2	136° a -63°	160°/s	(0; 1; 0)	(150; 0; 486,5)
3	55° a -350°	170°/s	(0; 1; 0)	(150; 0; 961,5)
4	200° a -200°	320°/s	(1; 0; 0)	(450; 0; 961,5)
5	115° a -115°	400°/s	(0; 1; 0)	(750; 0; 961,5)
6	400° a -400°	460°/s	(1; 0; 0)	(750; 0; 961,5)

Tabela 5: Especificação do Robô: ABB IRB 140.

Junta	Limite da Junta	Veloc.	s	s_o
1	360°	200°/s	(0; 0; 1)	(0; 0; 0)
2	200°	200°/s	(0; 1; 0)	(70; 0; 352)
3	280°	260°/s	(0; 1; 0)	(70; 0; 712)
4	400°	360°/s	(1; 0; 0)	(260; 0; 712)
5	240°	360°/s	(0; 1; 0)	(450; 0; 712)
6	800°	450°/s	(1; 0; 0)	(450; 0; 712)

Tabela 6: Especificação do Robô: ABB IRB 120.

Junta	Limite da Junta	Veloc.	s	s_o
1	165° a -165°	250°/s	(0; 0; 1)	(0; 0; 0)
2	110° a -110°	250°/s	(0; 1; 0)	(0; 0; 292)
3	70° a -90°	250°/s	(0; 1; 0)	(0; 0; 562)
4	160° a -160°	320°/s	(1; 0; 0)	(156; 0; 632)
5	120° a -120°	320°/s	(0; 1; 0)	(306; 0; 632)
6	400° a -400°	420°/s	(1; 0; 0)	(374; 0; 632)

sistema, são adicionadas cinco cadeias virtuais do tipo 3P3R. Essas cadeias são espaciais e o seu nome significa que elas são compostas por três juntas prismáticas e três juntas rotativas. A cadeia virtual CV_0 define o movimento da peça com relação a base, e as cadeias CV_i , com $i = 1, 2, 3, 4$, descrevem o movimento do efetuador final com relação a peça.

O cenário para a posição inicial e outros parâmetros será descrito a seguir e é definido como Caso de Referência. Na próxima subseção serão apresentadas as mudanças nos parâmetros para o caso de referência.

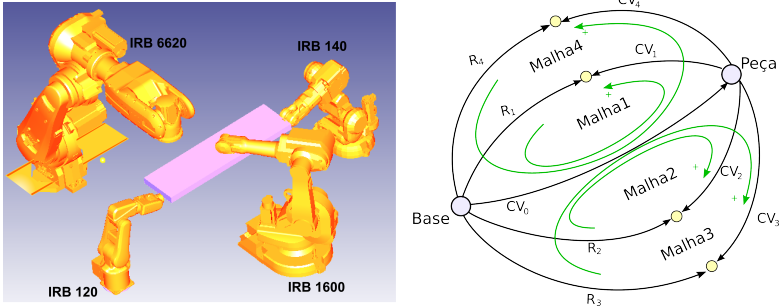


Figura 42: Sistema robótico cooperativo e sua representação por grafos.

A tarefa consiste de dois robôs posicionadores, um robô inspecionando a tarefa e o outro executando uma operação sobre a peça.

A cadeia virtual CV_0 relaciona o movimento da peça com relação ao sistema de coordenadas fixo, nesse sistema, a CV_0 tem a seguinte trajetória $CV_0 = (1200; -530 - 3,75t; 770; 0; 0; 0)$ com $0 < t < 40,8s$.

A cadeia virtual CV_1 descreve o movimento do efetuador do robô IRB 6620 com relação a peça. Nesse movimento é atribuído um conjunto de pontos que, quando unidos durante a execução da tarefa, formam a sigla UFSC sobre a peça.

Através da cadeia virtual CV_2 é atribuído o procedimento de inspeção e o robô IRB 1600 é designado para realizar a tarefa. A inspeção é feita, seguindo a trajetória $CV_2 = (100; -15t; 200; 0; 0; 0)$ com $0 < t < 40,8s$, relativa a origem da peça.

As cadeias virtuais CV_3 e CV_4 definem o movimento relativo do efetuador dos robôs IRB 140 e IRB 120 com relação a peça, respectivamente. Uma vez que esses robôs são designados a posicionar a peça, então suas trajetórias serão nulas com relação a peça, $CV_3 = CV_4 = 0$.

Com a tarefa e os robôs especificados, o próximo passo é definir a posição da base com relação ao sistema de coordenadas e a configuração inicial de cada robô. Estes parâmetros são descritos na Tabela 7.

Tabela 7: Posição da base dos robôs e configuração inicial das juntas.

Robôs	Posição da Base	Conf. inicial das juntas
IRB 6620	(0; -380; 0)	(0; 0,52; -0,52; 0; 1,57; 0)
IRB 1600	(2200; 0; 400)	(0,6; 0; 3,14; 0; -0,52; 0)
IRB 140	(1400; 1150; 200)	(-1,52; 0,26; 0; 0; -0,26; 0)
IRB 120	(1400; -1000; 250)	(1,52; 0,26; 0; 0; -0,2; 0)

A matriz de rede N do SRC é dada por:

$$N = \begin{bmatrix} \hat{\$}_{CV_0} & \hat{\$}_{CV_1} & -\hat{\$}_{R_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{\$}_{CV_0} & 0 & 0 & \hat{\$}_{CV_2} & -\hat{\$}_{R_2} & 0 & 0 & 0 & 0 \\ \hat{\$}_{CV_0} & 0 & 0 & 0 & 0 & \hat{\$}_{CV_3} & -\hat{\$}_{R_3} & 0 & 0 \\ \hat{\$}_{CV_0} & 0 & 0 & 0 & 0 & 0 & 0 & \hat{\$}_{CV_4} & -\hat{\$}_{R_4} \end{bmatrix}_{24 \times 54} \quad (6.24)$$

onde: $\hat{\$}_{CV_i}$ é uma matriz 6×6 que contém os helicoides normalizados referente às juntas das cadeias virtuais, com $i = 0, 1, 2, 3, 4$ e $\hat{\$}_{R_i}$ é uma matriz 6×6 que contém os helicoides normalizados referente às juntas dos robôs.

A matriz de rede é separada nas matrizes primária N_p e secundária N_s :

$$N_p = \begin{bmatrix} [\hat{\$}_{CV_0}]_{6 \times 6} & [\hat{\$}_{CV_1}]_{6 \times 6} & 0 & 0 & 0 \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & [\hat{\$}_{CV_2}]_{6 \times 6} & 0 & 0 \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & 0 & [\hat{\$}_{CV_3}]_{6 \times 6} & 0 \\ [\hat{\$}_{CV_0}]_{6 \times 6} & 0 & 0 & 0 & [\hat{\$}_{CV_4}]_{6 \times 6} \end{bmatrix} \quad (6.25)$$

$$N_s = \begin{bmatrix} [-\hat{\$}_{R_1}]_{6 \times 6} & 0 & 0 & 0 \\ 0 & [-\hat{\$}_{R_2}]_{6 \times 6} & 0 & 0 \\ 0 & 0 & [-\hat{\$}_{R_2}]_{6 \times 6} & 0 \\ 0 & 0 & 0 & [-\hat{\$}_{R_3}]_{6 \times 6} \end{bmatrix} \quad (6.26)$$

$$\dot{q}_p = [\dot{q}_{CV_0} \quad \dot{q}_{CV_1} \quad \dot{q}_{CV_2} \quad \dot{q}_{CV_3} \quad \dot{q}_{CV_4}]_{30 \times 1}^T \quad (6.27)$$

Os valores de N_p , N_s e \dot{q}_p definidos acima são substituídos na Equação 6.28 e assim, as velocidades das juntas dos robôs são calculadas, como uma função das velocidades definidas para cada efetuador.

$$\dot{q}_s = -N_s^{-1} N_p \dot{q}_p \quad (6.28)$$

A Figura 43 ilustra os robôs executando a tarefa designada em quatro momentos diferentes e uma vista de topo do sistema.

As posições das juntas para cada um dos robôs é apresentada na Figura 44, com a escala do tempo representada no eixo horizontal ($0 < t < 40,8s$).

A tarefa do robô IRB 6620 é mais complexa e por isso seu gráfico de posição apresenta um maior número de inflexões se comparado ao gráfico de posição dos outros robôs que apresentam curvas de posição suaves e algumas sendo constantes. Somente quatro juntas são usadas

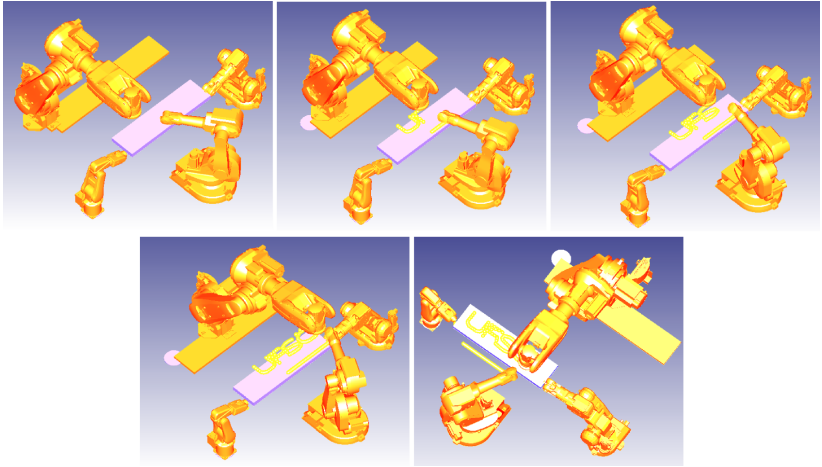


Figura 43: Robôs executando a tarefa cooperativamente.

no robô IRB 6620, assim, a quarta e a sexta juntas tem seus valores nulos.

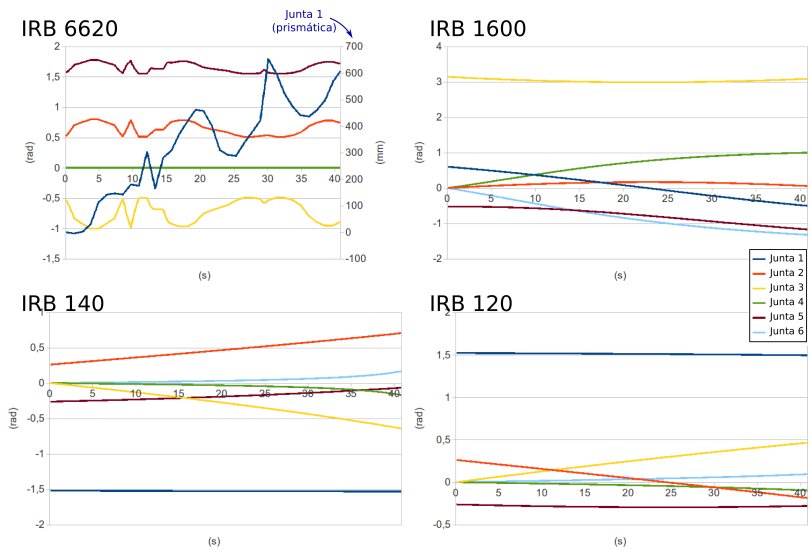


Figura 44: Gráfico mostra a posição das juntas para cada robô.

6.3.1 Modificações em alguns parâmetros do SRC

Agora são introduzidas algumas das modificações realizadas nos parâmetros do cenário de referência do SRC, as quais são propostas com a finalidade de avaliar o impacto que elas têm sobre o sistema como um todo.

Como ponto de partida foi investigado quais as variáveis poderiam impactar no resultado final do cálculo da cinemática do SRC. Sabe-se que se os manipuladores trabalham dentro de uma zona de conforto, com poucos movimentos bruscos nas juntas, a qualidade da tarefa pode ser melhor do que quando os robôs trabalham próximo de algum dos limites de suas juntas. Partindo desse fato, é considerado que a posição da base dos robôs e as suas configurações das juntas são variáveis que, quando combinadas corretamente, determinam a zona de conforto dos manipuladores. A zona de conforto varia de robô para robô e é constituída pela região da área de trabalho na qual o robô tem menores erros e melhores características dinâmicas. Além disso, a especificação da tarefa também está relacionada aos limites dos robôs, ou seja, determinadas tarefas podem ser executadas por alguns tipos de robôs, mas por outros não. No mais, quando especificadas as tarefas, estas devem ser definidas dentro do espaço operacional dos robôs.

Assim, alguns parâmetros como: configuração das juntas, posicionamento da base e especificação da tarefa possuem um conjunto de parâmetros que especificados coerentemente podem influenciar positivamente no desempenho de execução da tarefa pelo SRC. Foram, então, definidos quais os parâmetros a serem mudados de um cenário para outro, classificando as modificações em três diferentes tópicos: características da tarefa, distância entre as bases dos robôs e configuração inicial das juntas (como indicado na Figura 45, pelos números 1, 2 e 3, respectivamente).

As mudanças foram realizadas uma por vez em cada um desses tópicos e de forma heurística. A heurística utilizada tem como base o caso de referência, observando qual parâmetro pudesse impactar fortemente o resultado. A partir do caso de referência, notou-se que os dois robôs IRB 6620 e IRB 1600 possuem dimensões maiores que os robôs IRB 140 e IRB 120. Então, como primeira alteração a ser realizada, decidiu-se alterar a trajetória da tarefa imposta através da cadeia virtual CV_0 tornando a trajetória da peça mais longa no eixo y . Consequentemente, os robôs IRB 6620 e IRB 1600 teriam um deslocamento em determinadas juntas mais curto, se comparado ao caso de referência e os robôs IRB 120 e IRB 140 por serem de menor dimensão, teriam

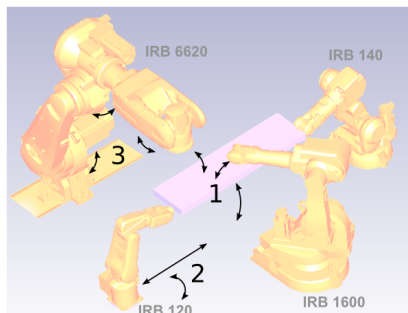


Figura 45: Indicação das modificações no sistema: características da tarefa (número 1), distância entre os robôs (número 2) e configurações iniciais (número 3).

deslocamento de juntas maiores. De fato, como será observado adiante (nas Tabelas 9 e 10), essa alteração teve um grande impacto se comparado ao caso de referência. Desta forma, deu-se início a criação de novos cenários, sempre analisando os cenários anteriores e buscando fazer alterações heurísticas, mas que influenciasse positivamente nos resultados.

Nos casos em que mais de um parâmetro era modificado, ele era uma composição das mudanças anteriores. Isto significa que, durante o processo de criação dos cenários, novas modificações ou recombinações eram adicionadas às modificações anteriores, gerando os novos cenários. O objetivo desses cenários não é apenas melhorar o resultado final, mas explorar as possibilidades de alteração dos parâmetros. Tanto o sistema quanto as modificações realizadas tentam simular casos aplicáveis na realidade.

A Tabela 8 indica quais modificações que foram realizadas usando como valores padrões os apresentados no caso de referência (característica da tarefa, configurações iniciais e distância entre os robôs), como definidas no início desta seção. Na segunda coluna são indicados quais parâmetros foram alterados e na terceira coluna uma breve explicação sobre a modificação. A Tabela 8 mostra doze cenários, designados por letras maiúsculas, em cada um desses cenários é realizado alguma modificação. Por exemplo, no cenário A, foi realizada uma alteração nos parâmetros da cadeia virtual CV_0 , neste caso, ao invés de ser como indicada na seção 6.3, ela passa a ser com os seguintes dados: $CV_0 = (1200; -530 - 7t; 770; 0; 0; 0)$ com $0 < t < 40,8s$ (a alteração

realizada na CV_0 está destacada em vermelho).

Tabela 8: Variações dos parâmetros nos diferentes cenários.

Cenário	Variação em relação ao caso de referência	Observações
A	CV_0	Aumentado o curso da trajetória imposta através da CV_0
B	Posição da base IRB120 e IRB140	Base dos robôs rotacionada em 90°
C	Posição da base IRB120 e IRB140; CV_0	Composição dos cenários A e B
D	Configuração inicial IRB6620, IRB1600, IRB120 e IRB140; CV_0	Configuração alterada nas juntas 2, executando as tarefas mais próximo do solo. Curso da trajetória CV_0 aumentado.
E	Configuração inicial, IRB6620, IRB1600, IRB120 e IRB140	Configuração alterada nas juntas 2, executando as tarefas mais próximo do solo.
F	CV_1, CV_2	IRB6620 executando tarefa de inspeção e IRB1600 escrevendo a sigla UFSC
G	CV_1, CV_2 ; Configuração inicial IRB6620, IRB1600	IRB6620 executando tarefa de inspeção e IRB1600 escrevendo a sigla UFSC. Configuração alterada buscando melhor desempenho
H	CV_1, CV_2 ; Configuração inicial IRB6620, IRB1600	IRB6620 executando tarefa de inspeção e IRB1600 escrevendo a sigla UFSC. Foi realizada outra alteração na configuração
I	CV_0	CV_0 anulada, mantém a peça fixa, sem necessidade dos robôs IRB120 e IRB140
J	CV_0	CV_0 é considerada junta secundária
L	$CV_0; CV_1, CV_2$	CV_0 é considerada junta secundária; IRB6620 executa a tarefa de inspeção e IRB1600 escreve a sigla UFSC
M	$CV_0; CV_1, CV_2$; Configuração inicial IRB6620, IRB1600	CV_0 é considerada junta secundária; IRB6620 executa a tarefa de inspeção e IRB1600 escreve a sigla UFSC; Configuração alterada buscando melhor desempenho

Para avaliar o impacto que cada uma das alterações exerceu so-

bre o sistema, foram criados três indicadores: velocidade máxima requerida, número total de picos de acelerações e deslocamento médio das juntas. O principal propósito desses indicadores é medir o desempenho do sistema.

No início da seção 6.3, nas Tabelas 3, 4, 5 e 6, foram mostradas as velocidades máximas definidas pelo fabricante dos robôs para cada uma das juntas. Baseado nesses valores, é calculado a fração da velocidade requerida de cada uma das juntas durante a execução da tarefa.

Para cada cenário, o percentual que a velocidade máxima requerida durante a execução da tarefa é calculada pela divisão da velocidade máxima requerida pela velocidade máxima nominal, como listadas pelos fabricantes. Por exemplo, se o valor é 50%, significa que, em algum momento durante a simulação, a velocidade de 50% da velocidade máxima listada é necessária para alguma junta.

A velocidade é calculada para cada uma das juntas de cada robô. Em seguida, a velocidade de maior valor é comparada com as especificações dos robôs (escolhendo a que possui maior percentual para ilustrar nas Tabelas 9 e 10).

Os picos de aceleração (segundo indicador) ocorrem quando uma determinada junta muda de trajetória, ou seja, a junta está seguindo a sua trajetória e precisa alterar seu movimento para o outro sentido. Isso provoca um pico no gráfico das acelerações. Assim, foram contados quantos picos teve cada uma das juntas dos robôs durante a realização da tarefa. Possivelmente, a tarefa já possui picos de aceleração em sua especificação, tornando a presença de picos inerente à execução da tarefa, no entanto, com a adequação do sistema, pode-se buscar minimizá-los através de alterações nos parâmetros. A aceleração de cada uma das juntas é calculada, e soma-se os valores absolutos de todas as juntas para cada um dos robôs. Se o valor é maior que 1 rad/s^2 , é então considerado um pico de aceleração. Finalmente, todos os picos de aceleração dos sistema são contados.

O outro indicador é o deslocamento médio das juntas dos robôs, em relação a configuração inicial. Esse indicador é útil para avaliar o quanto os robôs se afastam da configuração inicial. O indicador é calculado pela média do deslocamento de cada uma das juntas, em seguida o valor médio para o robô. E finalmente, o deslocamentos de todos os robôs são somados e então é encontrado o indicador de deslocamento médio.

Como no SRC existem 24 juntas e foram gerados 12 cenários diferentes, fica inviável apresentar os indicadores de todas as juntas, assim, nas Tabelas 9 e 10 são consolidados os indicadores para cada um

dos robôs componentes do SRC.

Tabela 9: Variações nos parâmetros nos cenários de referência e nos de A até F.

Cenário	Robôs	Velocidade Máxima	Número de Picos	Distância média da conf. inicial
REF.	IRB 6620	36,54%	43	0,08
	IRB 1600	1,16%	0	0,4
	IRB 140	0,46%	0	0,09
	IRB 120	0,31%	0	0,1
A	IRB 6620	36,54%	43	0,08
	IRB 1600	0,82%	0	0,3
	IRB 140	74,33%	61	0,2
	IRB 120	2,05%	0	0,6
B	IRB 6620	36,54%	43	0,08
	IRB 1600	1,16%	0	0,4
	IRB 140	0,43%	0	0,2
	IRB 120	0,6%	0	0,2
C	IRB 6620	36,54%	43	0,08
	IRB 1600	0,82%	0	0,3
	IRB 140	260,63%	53	0,2
	IRB 120	2,05%	1	0,6
D	IRB 6620	46,28%	44	0,1
	IRB 1600	0,85%	0	0,25
	IRB 140	25,88%	22	0,1
	IRB 120	0,35%	0	0,33
E	IRB 6620	36,54%	43	0,09
	IRB 1600	1,16%	0	0,53
	IRB 140	0,46%	0	0,2
	IRB 120	0,31%	0	0,1
F	IRB 6620	0,38%	1	0
	IRB 1600	24,87%	66	0,67
	IRB 140	0,46%	0	0,09
	IRB 120	0,31%	0	0,1

Houveram algumas alterações significativas durante a mudança de alguns dados definidos inicialmente. É possível notar no cenário C, que ao tornar o curso da trajetória dos robôs IRB 140 e IRB 120 mais longo e alterar a posição da base com relação ao sistema de coordenadas

Tabela 10: Variações nos parâmetros nos cenários de G até M.

Cenário	Robôs	Velocidade Máxima	Número de Picos	Distância média da conf. inicial
G	IRB 6620	0,38%	1	0
	IRB 1600	17,59%	60	0,58
	IRB 140	0,46%	0	0,09
	IRB 120	0,31%	0	0,1
H	IRB 6620	0,38%	1	0
	IRB 1600	21,55%	68	0,6
	IRB 140	0,46%	0	0,09
	IRB 120	0,31%	0	0,1
I	IRB 6620	36,54%	43	0,08
	IRB 1600	1,54%	0	0,5
	IRB 140	–	–	–
	IRB 120	–	–	–
J	IRB 6620	32,21%	113	0,17
	IRB 1600	5,3%	18	0,34
	IRB 140	2,51%	9	0,05
	IRB 120	1,24%	3	0,28
L	IRB 6620	11,17%	249	0,12
	IRB 1600	62,21%	51	0,5
	IRB 140	18,96%	31	0,13
	IRB 120	5,93%	24	0,5
M	IRB 6620	7,56%	226	0,07
	IRB 1600	17,54%	41	0,5
	IRB 140	4,9%	25	0,06
	IRB 120	2,14%	17	0,32

fixo, acabou prejudicando o desempenho da tarefa. Neste caso, avaliando as velocidades nas juntas de cada robô, verifica-se que a junta do robô IRB 140 não conseguiu executar sua parte na tarefa, prejudicando os outros robôs, configurando assim, uma singularidade do SRC.

Outro cenário que chama a atenção é o cenário A, que apresenta um grande número de picos e chega a requerer 74% da velocidade de uns dos robôs. Analisando as velocidades de juntas do robô IRB 140, verifica-se que ele também entrou em singularidade.

O cenário G chama a atenção devido ao fato de ter o menor valor de velocidade máxima, quando comparado aos outros cenários (Figura

46). Chegando a exigir menos da metade do cenário de referência. Vale lembrar que nesse cenário foram alteradas as tarefas entre os robôs IRB 6620 e IRB 1600, e neste caso, o robô menor (IRB 1600) passa a executar a tarefa de escrever a sigla UFSC, enquanto o robô maior inspeciona. Lembrando que o robô IRB 6620 possui uma junta prismática e para esse cenário a trajetória do efetuador tem o mesmo sentido de deslocamento da junta prismática.

Pelo gráfico de posição das juntas, ilustrado Figura 46, percebe-se que somente a junta prismática do robô IRB 6620 se deslocou para a realização da tarefa de inspeção, isso significa que o robô IRB 6620 pode estar sendo subutilizado, podendo ser substituído por um outro tipo de estrutura física, como uma mesa XY para a realização dessa tarefa. Lembrando que, quando o robô IBR 1600 realizava a tarefa de inspeção, todas as suas juntas se movimentavam (conforme Figura 44).

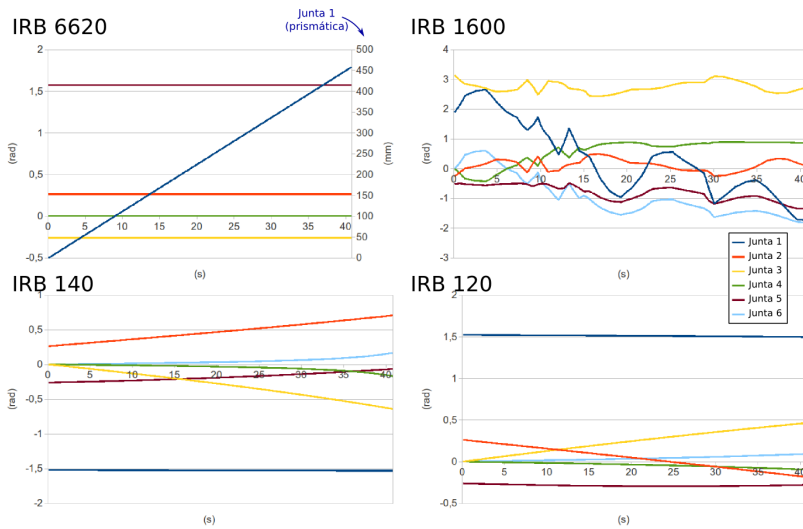


Figura 46: Posição das juntas dos robôs para o cenário G.

Inicialmente, se esperava que quando as juntas da cadeia virtual CV_0 fossem tratada como secundárias (com velocidades a serem determinadas), o sistema como um todo teria bons resultados nos indicadores, com baixas velocidades e poucos deslocamentos. Porém, o comportamento foi diferente do esperado, com a flexibilidade de movimento da peça, o robô IRB 6620 se movimentou mais, se comparado aos

outros cenários no qual era imposta uma trajetória a peça através da CV_0 . Como resultado, o robô IRB 6620 teve vários picos de acelerações, chegando a ter 249, no cenário L, de um total de 355 picos de aceleração para os quatro robôs, mesmo quando a ele lhe cabia a tarefa de inspeção. Esses picos foram ocasionados devido à movimentação calculada da peça. Mesmo assim, no cenário M ainda foi possível obter o menor valor entre todos os cenários para o percentual da velocidade máxima, indicando que, caso os picos e deslocamentos não sejam impeditivos, podem ser alcançados bons resultados utilizando a cadeia virtual CV_0 como secundária. Acredita-se que fazendo outras alterações no posicionamento da base, nas configurações iniciais ou nas especificações da tarefa é possível melhorar os resultados.

Como se observa, alguns dos cenários apresentados não exigem 100% da velocidade estabelecida pelo fabricante. Assim, é possível executar as tarefas em modo acelerado, no qual, se comanda os robôs para que sigam a trajetória calculada na simulação com uma velocidade multiplicada por um valor constante. Permitindo assim uma margem para que se execute a tarefa mais rapidamente.

O valor máximo que pode ser utilizado para multiplicar a velocidade é exatamente $1/(\% \text{ de velocidade máxima})$, de forma que é possível, por exemplo, executar a tarefa em 14,9s utilizando o cenário de referência e 7,18s utilizando o cenário G (a opção mais rápida possível dentre os cenários apresentados).

A avaliação dos indicadores depende da necessidade do programador e quais os resultados esperados para a qualidade da tarefa. Por exemplo, um maior número de picos de aceleração e um maior deslocamento médio das juntas significa que as juntas se moveram muito e tiveram muitas mudanças de orientações. Este fato pode resultar em um desgaste maior nas juntas dos robôs. Por outro lado, se velocidades maiores são requeridas, elas podem influenciar na qualidade final da tarefa. Devido a essas possíveis interpretações, é difícil avaliar o cenário mais interessante sem um objetivo real. Considerando essa observação, pode-se escolher o cenário G, como o melhor entre os disponíveis, uma vez que apresenta a menor velocidade e o deslocamento médio das juntas é o terceiro menor. Nesse caso, é um cenário que apresentou bons resultados nos três critérios propostos.

Os critérios utilizados para avaliar, assim como, a simulação do sistema, são importantes para detectar problemas durante a execução da tarefa: como colisão e singularidades.

6.4 SIMULAÇÃO DE UM SRC COM BASE MÓVEL

Com a finalidade de abordar um conjunto maior de simulações e buscando atender as várias opções que a metodologia proposta oferece é apresentado agora um exemplo de um sistema composto por quatro robôs em cooperação com uma característica adicional como detalhado na seção 5.3 do Capítulo 5.

Os robôs componentes do sistema são dois ABB IRB 1600, um ABB IRB 140 e um ABB IRB 120. Para um dos robôs IRB 1600 caberá a função de posicionar a peça e ao outro executar uma trajetória sobre a peça. O manipulador IRB 120 tem sua base inclinada com relação ao sistema de coordenadas fixo em um ângulo de 45° com relação ao plano XY. A base do robô IRB 140 está em movimento relativo com relação ao sistema de coordenadas fixo, ou seja, à ela é imposta uma trajetória simulando o deslocamento da base do manipulador sobre uma plataforma móvel. Além disso, os robôs IRB 120 e IRB 140 irão executar uma tarefa sobre a peça. A Figura 47 ilustra a peça que o robô IRB 1600 está posicionando, nela os outros robôs irão efetuar as trajetórias designadas. As informações das tarefas são descritas na Tabela 11.

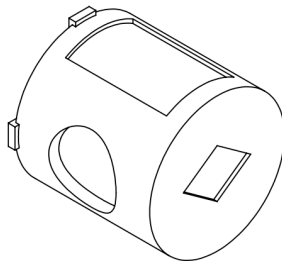


Figura 47: Peça sobre a qual o robôs executarão suas tarefas.

A Figura 48 ilustra o posicionamento dos robôs, da peça e do grafo do sistema. No grafo estão representadas as cadeias virtuais que foram adicionadas ao SRC, em que as cadeias virtuais A_i , são do tipo 3P3R e através delas é imposto o deslocamento relativo da base de cada robô com relação ao sistema de coordenadas fixo. Para este caso, a cadeia virtual A_3 terá a função de impor o movimento na base do robô IRB 140 (representado no grafo por R_3). As cadeias A_1 , A_2 e A_4 são constantes representando uma posição fixa da base de cada robô. No

Tabela 11: Parâmetros das tarefas impostas ao SRC.

Robô	Tarefa
IRB 1600	O efetuador do robô posicionará a peça, $CV_1=(0; 0; 0; 0; 0; 0)$
IRB 1600	O efetuador do robô irá executar uma trajetória equivalente a circunferência sobre a peça.
IRB 140	O efetuador do robô irá executar uma trajetória equivalente a um retângulo sobre a peça. A base do robô será móvel, como se ela estivesse sobre uma plataforma, deslocando-se no sentido vertical e horizontal (plano XZ).
IRB 120	O efetuador do robô irá executar uma trajetória equivalente a um losango sobre a peça. A base do robô é inclinada a 45° com relação ao plano XY.

entanto, a cadeia virtual A_4 posicionará a base do robô IRB 120 com a inclinação especificada na tarefa. As cadeias virtuais CV_i representam o deslocamento da peça com relação ao efetuador, com $i = 1, 2, 3, 4$ e CV_0 representa o deslocamento da peça com relação ao sistema de coordenadas fixo. As cadeias virtuais são todas adicionadas em série conforme a Propriedade 1 discutido no Capítulo 4.

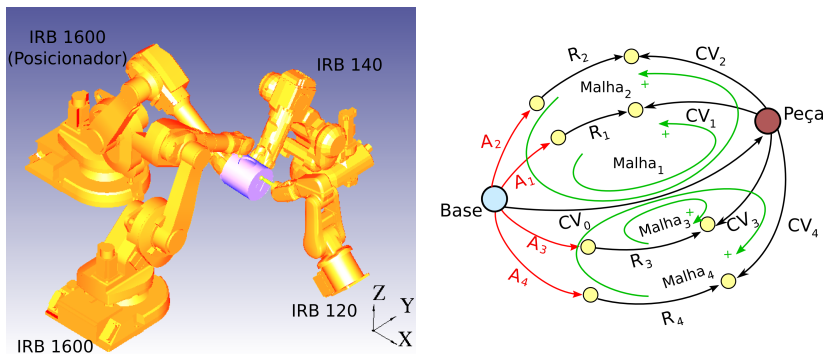


Figura 48: SRC e sua representação por grafos.

As matrizes N_p e N_s contendo os helicoides normalizados do sis-

tema são (os elementos em vermelho nas matrizes N_p e \dot{q}_p destacam a cadeia virtual que representa a base móvel):

$$N_p = \begin{bmatrix} [\hat{\$}_{CV_0}] & [\hat{\$}_{CV_1}] & -[\hat{\$}_{A_1}] & 0 & 0 & 0 & 0 & 0 & 0 \\ [\hat{\$}_{CV_0}] & 0 & 0 & [\hat{\$}_{CV_2}] & -[\hat{\$}_{A_2}] & 0 & 0 & 0 & 0 \\ [\hat{\$}_{CV_0}] & 0 & 0 & 0 & 0 & [\hat{\$}_{CV_3}] & -[\hat{\$}_{A_3}] & 0 & 0 \\ [\hat{\$}_{CV_0}] & 0 & 0 & 0 & 0 & 0 & 0 & [\hat{\$}_{CV_4}] & -[\hat{\$}_{A_4}] \end{bmatrix}_{24 \times 54} \quad (6.29)$$

$$N_s = \begin{bmatrix} [-\hat{\$}_{R_1}]_{6 \times 6} & 0 & 0 & 0 \\ 0 & [-\hat{\$}_{R_2}]_{6 \times 6} & 0 & 0 \\ 0 & 0 & [-\hat{\$}_{R_2}]_{6 \times 6} & 0 \\ 0 & 0 & 0 & [-\hat{\$}_{R_3}]_{6 \times 6} \end{bmatrix}_{24 \times 24} \quad (6.30)$$

$$\dot{q}_p = [\dot{q}_{CV_0} \quad \dot{q}_{CV_1} \quad \dot{q}_{A_1} \quad \dot{q}_{CV_2} \quad \dot{q}_{A_2} \quad \dot{q}_{CV_3} \quad \dot{q}_{A_3} \quad \dot{q}_{CV_4} \quad \dot{q}_{A_4}]_{45 \times 1}^T \quad (6.31)$$

Com as matrizes N_p , N_s e \dot{q}_p especificadas, elas são então substituídas na Equação 6.28 e assim, são obtidos os valores das velocidades de cada uma das juntas dos robôs. Na Figura 49 é mostrada uma sequência de quatro momentos da execução da tarefa.

A Figura 50 ilustra o gráfico contendo as posições de cada uma das juntas dos robôs componentes do sistema e as posições que a base do robô IRB 140 percorre durante a execução da tarefa. Percebe-se pelo gráfico – Base Móvel (IRB 140) – que a base do robô se deslocou 200mm no eixo X (de 1000mm para 1200mm), imposta pela junta prismática virtual P_x (junta 1 no gráfico). No eixo Z a base do robô se deslocou de 240mm para 338mm, depois de 338mm para 142mm e de 142mm voltou para os 240mm, esses valores foram definidos ao sistema através da junta virtual P_z (junta 3 no gráfico).

Aplicando os indicadores propostos na seção 6.3 para o caso de um sistema com a base móvel é possível perceber que o sistema teve um comportamento aceitável. O maior percentual da velocidade máxima foi atingido pelo Robô IRB 1600 posicionador, chegando a ser 55,91% da velocidade indicada pelo fabricante do robô. O robô IRB 140, que possui a base móvel, também apresentou bons indicadores.

Pelos indicadores foi possível perceber durante a simulação do SRC que, em outros cenários, o robô IRB 140 passava por singularidades. De fato, os resultados mostrados aqui foram selecionados após ajustes nas configurações iniciais de juntas, obtendo um melhor desempenho do SRC, sem a presença de singularidades. Os indicadores para cada um dos robôs componentes do sistema podem ser conferidos na Tabela 12.

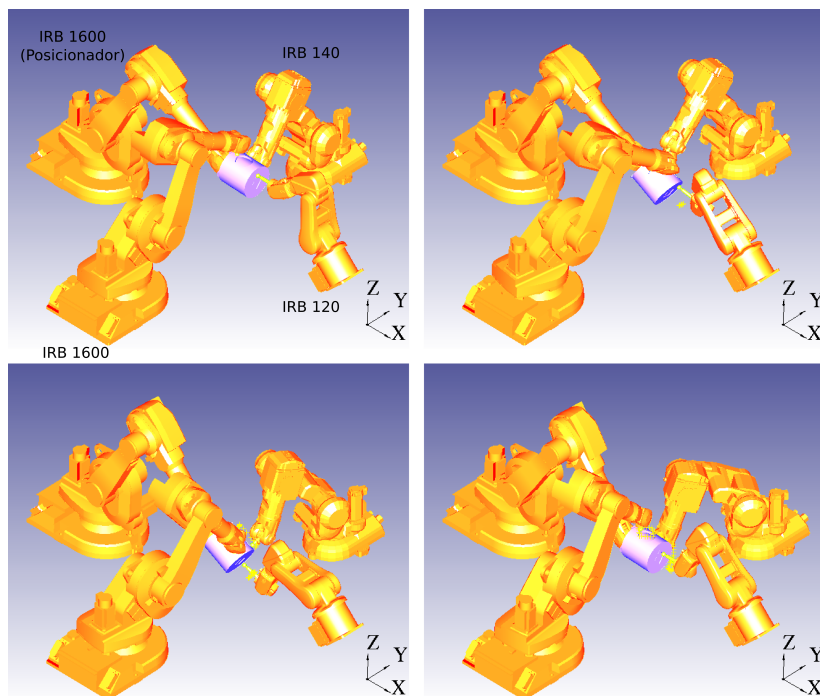


Figura 49: Sequência de execução da tarefa com base móvel do robô IRB 140.

Tabela 12: Indicadores de avaliação do SRC com a base de um dos robôs móvel.

Robôs	Veloc. Máx.	Picos	Desloc. médio
IRB 1600 (Pos)	55,91%	47	0,33
IRB 1600	36,88%	40	0,33
IRB 140	17,50%	41	0,44
IRB 120	26,29%	38	0,23

Esse exemplo completa o conjunto de simulações propostas que abordaram diversas aplicações factíveis de serem encontradas nas indústrias. É relevante comentar que os bons resultados obtidos nas simulações se devem principalmente à abordagem pela teoria de helicoides,

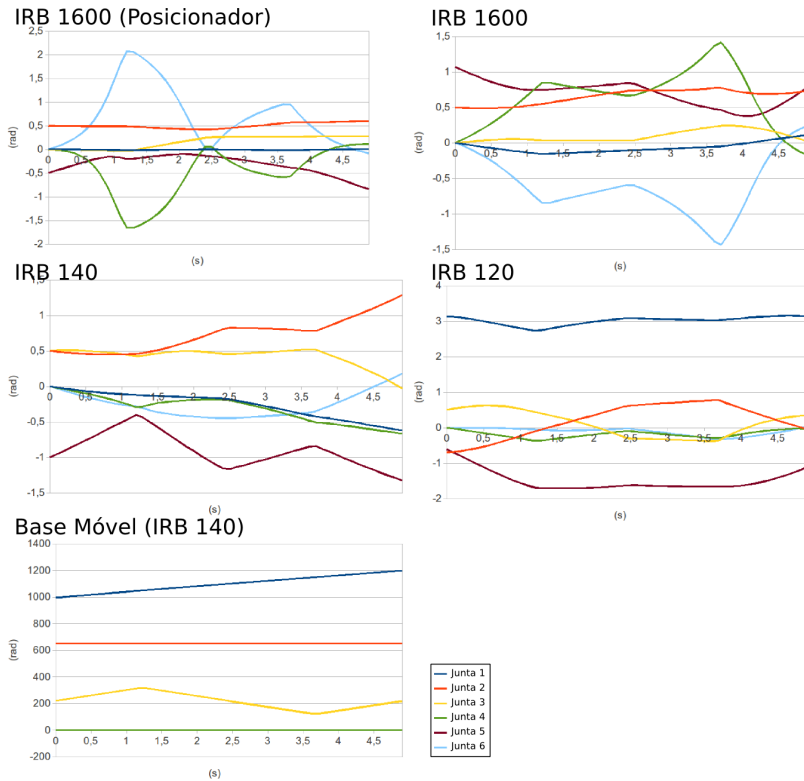


Figura 50: Gráfico das posições de cada uma das juntas dos robôs e o gráfico das posições da base do robô IRB 140.

método de Davies e pelas cadeias virtuais. Associada a essas teorias, a metodologia desenvolvida nesta tese facilita a aplicação dessas teorias em SRC de forma sistemática.

6.5 CONCLUSÕES

As simulações são de grande importância no processo de programação e planejamento de tarefas robóticas. Com elas é possível perceber a atuação de cada uma das juntas do sistema e o comportamento de cada robô na execução de sua tarefa.

Os resultados dessas simulações mostram a flexibilidade na escolha dos robôs componentes de um SRC e na especificação das tarefas. A simulação pode ser utilizada para avaliar a influência e a viabilidade do uso de cada robô no sistema para executar uma determinada tarefa industrial, além de também possibilitar uma visão geral do processo.

Verifica-se que a metodologia para a programação robótica permite diferentes combinações de configurações. Através da análise dos resultados obtidos é possível perceber se existe ou não a viabilidade de otimização do desempenho do sistema e a possibilidade de reduzir o tempo de execução da tarefa.

O fluxo de trabalho apresentado, que consiste em realizar alterações nas condições iniciais e avaliar os resultados via indicadores, permite realizar várias combinações de parâmetros e efetuar os cálculos necessários, baseado na metodologia que, vale ressaltar, é orientada à tarefa. Assim, se permite a exploração, pela simulação, de muitas opções diferentes de modificações dos parâmetros. Estas opções podem ser melhor analisadas e testadas, levando-se em conta os indicadores relevantes, de acordo com a necessidade dos programadores do SRC.

Os resultados das programações *off-line* são importantes na programação de SRC, especialmente quando vários robôs são incluídos, considerando o grande número de parâmetros a serem analisados. Devido a essas características, é difícil implementar uma tarefa real sem simulações prévias. A avaliação analítica poderia levar a resultados não práticos, devido ao grande número de variáveis e interdependências. Usando a simulação, é possível ter uma visão completa do comportamento do sistema, detectando singularidades, colisões e destacando o desempenho da tarefa na sua execução.

Com relação aos sistemas apresentados, as simulações apresentaram bons resultados, permitindo que um deles fosse aplicado em um sistema real, como será comentado no Capítulo 7 a partir dos dados obtidos nas simulações. A flexibilidade da metodologia é comprovada no segundo sistema, no qual é apresentado um caso de referência e baseado nele são alterados alguns parâmetros com facilidade.

É importante notar que as poucas modificações apresentadas na subseção 6.3.1 implicaram em significativas mudanças nos resultados finais do cálculo da cinemática do SRC. O uso de heurísticas melhoradas provavelmente levaria a resultados melhores, mas contudo o seu desenvolvimento precisa ser realizado a fim de implementar uma sistemática na escolha das mudanças ou heurísticas mais complexas.

A metodologia é completa no sentido de abordar qualquer SRC e qualquer tipo de tarefa. Além disso, ela é baseada na teoria de he-

licoides, método de Davies e cadeias virtuais de Assur o que facilitou a programação de uma diversidade de SRCs, incluindo casos com as características adicionais, como pôde ser comprovado no terceiro caso simulado (seção 6.4).

7 APLICAÇÕES EM SISTEMAS REAIS

A implementação real de uma tarefa em um SRC possui algumas dificuldades, pois além de todas as etapas necessárias para programação de sistemas com um único robô, ainda é preciso ajustar as simulações para as configurações de posições relativas entre os robôs, para o relacionamento da peça com as várias tarefas e para as características do ambiente. Com a ajuda dos resultados simulados (comentados no Capítulo 6) foram programados os Robôs XR-4 e Scara da Rhino para realizarem as tarefas. Neste capítulo, serão comentados alguns detalhes dessa implementação, além das dificuldades técnicas e estruturais encontradas durante o processo de programação. Por fim, são comentados os resultados obtidos da execução da tarefa pelos robôs.

7.1 CARACTERÍSTICAS DO SISTEMA

Os resultados obtidos nas simulações foram implementados nos robôs XR-4 e Scara. Para a comunicação com os controladores dos robôs foram utilizados dois computadores, um para cada robô, devido a limitação de uma porta paralela por computador. A comunicação é feita através da porta paralela e o programa utilizado para enviar os dados aos controladores é o desenvolvido originalmente pela Rhino Robotics Ltd e se chama *RoboTalk* para o sistema operacional MS-DOS (RHINO, 2010). A licença disponível para Sistema Operacional Windows é uma versão de demonstração e permite enviar apenas cinco linhas de comandos para os robôs. Devido a este fato, foi escolhido a versão do programa *RoboTalk* para MS-DOS.

Cada robô possui o seu controlador *Mark IV* independente. Para eles iniciarem o movimento ao mesmo tempo foi adaptado um conector ligando os controladores para enviar o sinal de início do movimento.

As trajetórias das juntas dos robôs foram geradas *off-line* utilizando os algoritmos implementados neste trabalho. Foram testados outros exemplos de tarefas e verificado se não havia descontinuidades nos caminhos, se a trajetória não ultrapassava os limites de juntas e do espaço de trabalho e, ainda, se não havia auto-colisão entre os robôs. Essas verificações foram feitas visualmente e também através dos gráficos de saída do programa apresentados no Capítulo 6 da seção 6.2. Por fim, as trajetórias das juntas foram transferidas aos controladores dos robôs através do programa *RoboTalk*.

A Tabela 13 ilustra a formatação do arquivo que contém os conjuntos de pontos da trajetória das juntas. O arquivo é transmitido para os manipuladores Rhino. Cada ponto da trajetória contém as posições de cada uma das juntas, medidas em pulsos, que é a unidade aceita pelos controladores. Assim sendo, para o robô XR-4, um ponto de trajetória possui cinco valores sequenciais, representando as juntas 1, 2, 3, 4 e 5, respectivamente. A extensão aceita do arquivo aceita pelo programa *RoboTalk* é *rt*.

Tabela 13: Estrutura do arquivo enviado para os manipuladores (este exemplo se refere ao manipulador XR-4, com 5 juntas).

```
CLEARPATH
POINT P1=0,0,-1587,1587,0
SETPATH P1
POINT P2=10,25,-1553,1654,2
SETPATH P2
POINT P3=21,51,-1517,1720,4
SETPATH P3
POINT P4=27,65,-1498,1755,5
SETPATH P4
POINT P5=30,71,-1490,1771,8
SETPATH P5
POINT P6=34,76,-1483,1786,11
SETPATH P6
POINT P7=39,80,-1477,1801,18
SETPATH P7
```

7.2 DESAFIOS E PROBLEMAS ENCONTRADOS DURANTE A APLICAÇÃO DA METODOLOGIA EM UM CASO REAL

Cabe ressaltar alguns detalhes e os desafios encontrados no procedimento realizado para a programação do SRC real.

7.2.1 Desafios técnicos

O programa *RoboTalk* disponível só aceita enviar aos robôs um arquivo contendo 100 pontos do caminho. Então para cada uma das juntas dos robôs eram enviadas no máximo 100 pontos da sua trajetória.

O programa *RoboTalk* permitia enviar as trajetórias ao robô Scara através do comando *to point*, enquanto que, para o robô XR-4 era possível enviá-las utilizando os comandos *to point* e *to path*. Esse aspecto dificultou a programação do SRC, pois, neste trabalho, é interessante que os robôs executem a trajetória percorrendo um caminho contínuo (utilizando, neste caso, o comando *to path*), ao invés de um caminho ponto-a-ponto. Para resolver esse problema e poder continuar a programação, o robô Scara foi tratado no programa *RoboTalk* como sendo um robô XR-4, assim, ao enviar as trajetórias para o controlador do robô Scara, era dito ao programa que se tratava de um robô XR-4. Porém, não foi possível avaliar os impactos que essa mudança pode causar no resultado final, já que os robôs são antigos e não há possibilidade de se medir os torques e rotações das juntas, sem implementar alterações estruturais nos robôs.

Uma das juntas do robô Scara não estava funcionando corretamente (a junta 4 indicada pelo seu deslocamento β_4 na Figura 32 do Capítulo 6). Ao iniciarem os movimentos, essa junta do robô Scara realizava o seu movimento todo de uma vez, antes das outras juntas iniciarem seus movimentos. Imagina-se que este problema se deve ao fato de o robô Scara ser tratado como XR-4 pelo programa e o controlador ter conhecimento de que se trata de um robô Scara, porém, não é possível afirmar porque essa junta tinha esse comportamento. Se fossem enviados os caminhos utilizando o comando *to point*, essa junta executa o movimento corretamente de acordo com o movimento das outras juntas.

7.2.2 Desafios estruturais

Alguns sensores de fim de curso estão comprometidos, e, sem eles, era necessário tomar cuidado para que a trajetória não ultrapassasse o limite de determinadas juntas, pois, caso ultrapassassem e sem esse sensor funcionando corretamente, a estrutura do robô poderia ser danificada. Além disso, os robôs têm muitas folgas e algumas peças precisam ser trocadas.

Os controladores possuem algumas chaves que quando conectadas, permitem a comunicação entre eles. Por algum motivo não apurado, essas chaves, quando conectadas, não sinalizavam a comunicação desejada. Desta forma, foi adaptado um botão que possibilitava essa comunicação para que os controladores pudessem trocar sinais e assim iniciarem o movimento dos robôs ao mesmo tempo.

Foi necessário também fazer uma conversão dos ângulos de juntas fornecido pelo programa de simulação de radianos para pulsos, que é a unidade aceita pelo controlador do robô.

Buscando amenizar esses problemas, o Laboratório LAI (LAI, 2010) está realizando estudos para modernizar esses robôs, como, por exemplo, adaptar a comunicação do controlador com o computador através de uma porta USB e a construção de um novo controlador totalmente aberto.

7.3 RESULTADOS EXPERIMENTAIS

Foram realizadas algumas tarefas experimentais com os robôs, a princípio alguns pontos eram enviados, depois foram testadas tarefas mais simples, no qual o efetuador do robô percorresse trajetórias retilíneas e circulares. As trajetórias eram testadas em um robô por vez. Após verificar o funcionamento dos robôs foi então aplicado os resultados obtidos nas simulações do Caso 1 e Caso 2 detalhados na seção 6.2 do Capítulo 6.

Para o Caso 1, em que o robô XR-4 deve escrever as letras UFSC, os resultados obtidos na simulação, ilustrados na Figura 36 do Capítulo 6, foram exportados gerando um arquivo contendo 80 pontos. Estes pontos descrevem as posições e velocidades de cada uma de suas juntas, para que o robô execute uma tarefa suave. A Figura 51 ilustra a tarefa sendo executada pelo robô em três momentos e a Figura 52 mostra uma imagem do resultado final da tarefa. Vale ressaltar que não existia nenhum objeto segurando a peça, ou seja, ela estava solta sobre a mesa.

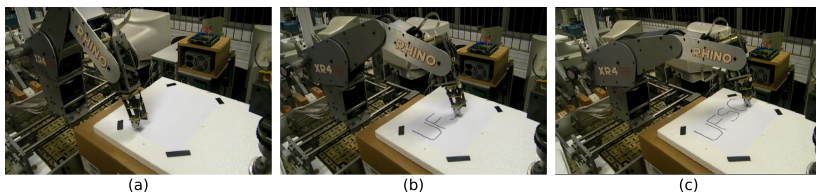


Figura 51: Robô XR-4 executando a tarefa.

Para o Caso 2 (no qual o robô XR-4 tem como tarefa escrever UFSC e o robô Scara seguir uma trajetória retilínea), as informações de juntas foram exportadas gerando dois arquivos, um para cada um

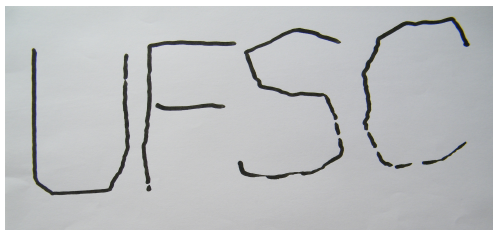


Figura 52: Resultado da tarefa executada pelo robô XR-4.

dos robôs, contendo as posições/velocidades das juntas. A Figura 53 ilustra três momentos dos robôs executando as tarefas especificadas. O resultado final da tarefa executada é mostrado na Figura 54.

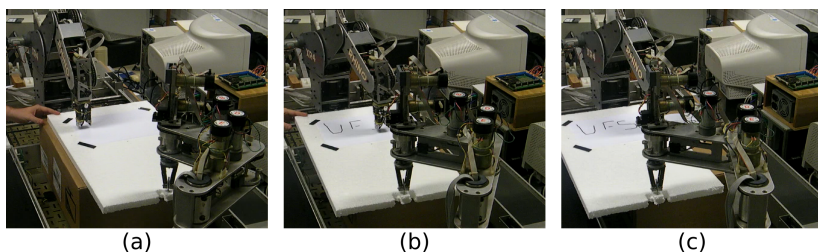


Figura 53: Os robôs XR-4 e Scara executando as tarefas em cooperação.

Com relação ao desempenho dos robôs, eles conseguiram realizar os movimentos de forma satisfatória, conforme os arquivos enviados. Porém, devido às limitações dos robôs citadas anteriormente e do material utilizado, o resultado da tarefa não teve a mesma qualidade apresentada quando o XR-4 executou ela sozinho. É necessário considerar também que a distância entre eles não foi alterada para realizar o experimento, ficando os robôs distantes um do outro (sendo de 106,7cm a distância entre as bases dos robôs). Com isso, existe um potencial ponto de complicação, pois as incertezas dos robôs que operam em cooperação são somadas e os erros de orientação se tornam significativos quando a distância entre os robôs é grande.

Devido a essas razões, o resultado final da tarefa utilizando dois robôs não foi muito bom se comparado ao robô XR-4 executando ela sozinho. Mas desconsiderando isso, é possível perceber que a utilização

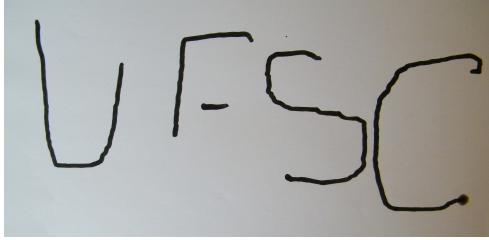


Figura 54: Resultado da tarefa executada pelos robôs em cooperação.

de mais de um robô é conveniente para alguns casos.

A Figura 55 ilustra o caminho que o efetuador do robô XR-4 deve percorrer para realizar sua tarefa. Este caminho é tratado na imagem como caminho de referência e está indicado em vermelho. O caminho resultante obtido na simulação está indicado na Figura 55 em azul. Tanto para o Caso 1 quanto para o Caso 2 percebe-se que os pontos do caminho se sobrepõem ou estão muito próximos em quase todo o período de execução da tarefa.

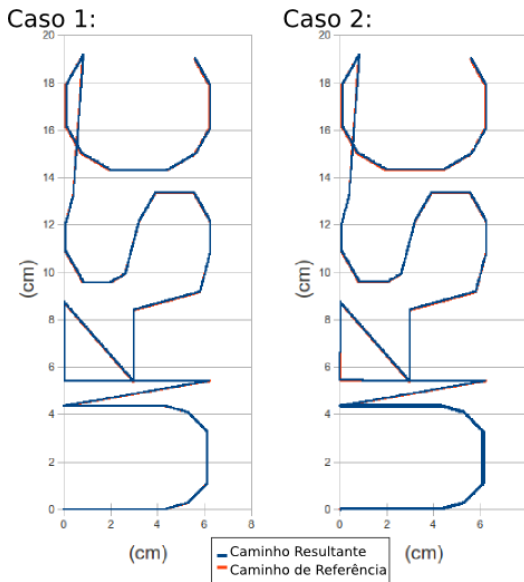


Figura 55: Caminho de referência e caminho resultante da simulação.

Uma comparação do erro de seguimento do caminho pelo efetuador do robô para os dois casos é mostrado na Figura 56. Nota-se que o erro é crescente devido a aplicação do método de integração de Euler e por não ter sido adicionada a cadeia de virtual para controle do Erro. Além disso, o erro tanto para o Caso 1 quanto para o Caso 2 é da mesma ordem de grandeza. Como pode ser observado na Figura 56 o erro obtido no cálculo da cinemática inversa é de apenas 0,35%, o que mostra que a diferença encontrada nos caminhos durante a aplicação em um sistema real é decorrente das imprecisões da estrutura do sistema cooperativo real.

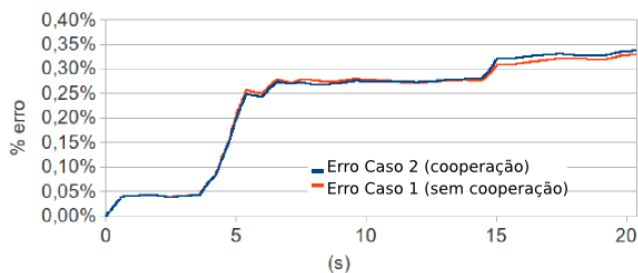


Figura 56: Comparação do erro entre o Caso 1 e o Caso 2.

7.4 VANTAGENS DA METODOLOGIA APLICADA

Acredita-se, ao final deste experimento e avaliando a metodologia proposta, que a programação de SRC tendo como base a teoria de helicoides é mais clara. A sistemática para determinar os parâmetros s e s_o da teoria de helicoides é simples se comparada à convenção de Denavit-Hartenberg (ROCHA; TONETTO; DIAS, 2011).

Além disso, na convenção de Denavit-Hartenberg é necessário definir sistematicamente um sistema de coordenadas para cada elo do manipulador, além de ter que definir quatro parâmetros. Muito embora a convenção de Denavit-Hartenberg defina uma sistemática, ainda restam pontos em que se pode realizar escolhas arbitrárias, que acabam impossibilitando a definição única dos sistemas de coordenadas referentes a cada elos. A teoria de helicoides é bem definida com relação aos parâmetros de elos s e s_o e, ao mesmo tempo, é mais generalizada (ROCHA; TONETTO; DIAS, 2011).

Com o auxílio do método de Davies e das cadeias virtuais de Assur, é possível realizar o cálculo da cinemática baseado em helicoides para sistemas com cadeias cinemáticas abertas e fechadas. Essas cadeias podem incluir cadeias cinemáticas que representam manipuladores, robôs paralelos, robôs móveis, robôs subaquáticos, robôs cooperativos, entre outros. A variedade de aplicações proporcionadas pelo uso destas metodologias é mais extensa quando comparada ao cálculo de cinemática através de Denavit-Hartenberg. Apesar da convenção de Denavit-Hartenberg ser amplamente apresentada na literatura, poucos resultados aplicados são encontrados, a não ser para sistemas compostos por um robô ou dois robôs.

A teoria de helicoides permitiu generalizar o cálculo da cinemática para SRC, incluindo diversas abordagens diferentes e tratando de casos especiais como base móveis, evitamento de colisão, sistemas redundantes e diversos tipos de tarefas. Além disso, a análise da cinemática em sistemas baseados na teoria de helicoides pode ser estendida para detecção de problemas como colisão e singularidades.

A forma de programar o SRC orientado à tarefa (baseado na adição de cadeias virtuais) possibilita separar e classificar diferentes tarefas. Para classificar as tarefas adequadamente é necessário que a estrutura de cadeias virtuais, com o uso das cadeias virtuais CV_0 e CV_i , seja aplicada.

7.5 CONCLUSÕES

Este capítulo finaliza o processo de programação de SRC, passando por todas as etapas, desde a aquisição de informações sobre os robôs e as tarefas até a sua execução pelo SRC.

A aplicação dos resultados em um SRC real possibilitou uma experiência única ao(s) autor(es). O contato com os robôs e os resultados da realização da tarefa pelo sistema foram construtivos.

A realização do experimento ocorreu no período de 17 à 21 de maio de 2010, sendo que os únicos contatos com os robôs reais foram durante esse período. A meta era em apenas uma semana realizar todo o experimento, desde aprender a ligar os robôs até fazer com que eles realizassem as tarefas. Três pessoas estavam envolvidas na programação dos robôs. Acredita-se que a meta foi atingida e os resultados obtidos provam que a programação de SRC utilizando a metodologia apresentada neste trabalho é viável para aplicação.

8 CONCLUSÕES E TRABALHOS FUTUROS

8.1 CONCLUSÕES

Os Sistemas Robóticos Cooperativos (SRC) determinam níveis de complexidade, tanto no seu entendimento como um todo quanto na sua programação. A programação manual, sem o auxílio de simuladores demanda muito tempo e o resultado final pode não ser compatível com o desejado. Assim, é necessário utilizar alguma metodologia ou sistemática e um programa auxiliar.

Neste trabalho, buscou-se encontrar uma forma de tornar o processo de programação de SRC menos trabalhoso e que facilite no entendimento do sistema como um todo. Pode-se perceber que o sistema não é composto apenas por robôs e tarefas, mas que também existe a influência de fatores externos, como por exemplo, do ambiente.

Este ambiente tem fundamental importância na programação e tem que ser incluído nela. Este trabalho tem como contribuições mais importantes:

- a sistematização do processo de programação de SRC para a realização de tarefas, ampliando a metodologia para a programação de robôs;
- a sistematização da modelagem cinemática de SRC;
- a ampliação do uso do método de helicoides e sua fundamentação para o entendimento, controle e programação de SRCs;
- o cálculo da cinemática direta e inversa de SRC seguindo a modelagem apresentada;
- a formalização matemática para a sistemática de adição de robôs e cadeias virtuais ao SRC;
- a utilização da sistemática de adição de cadeias virtuais para a representação de características adicionais;
- o desenvolvimento de uma plataforma de simulação cinemática de SRC;
- a experimentação das trajetórias geradas pela programação em sistemas reais.

A contribuição principal é a sistematização da modelagem cinemática de SRC, a qual foi discutida no Capítulo 3 e formalizada no Capítulo 4. Essa sistematização permite uma maior flexibilidade e facilidade na programação de SRC, sendo sua estrutura composta por m robôs e $m + 1$ cadeias virtuais. As características adicionais são tratadas através da adição em paralelo ou em série de subgrafos no sistema (abordadas nas Definições 1 e 2 e nas Propriedades 1 e 2), tornando a modelagem genérica e permitindo a inclusão de novas características.

Assim, foram apresentados e formalizados a modelagem genérica de SRC, com adição estratégica de cadeias virtuais bem como, considerado que o sistema tem um conjunto de m robôs e $m + 1$ cadeias virtuais. Conforme as Propriedades 1 e 2 podem ser incluídas características adicionais ao SRC, como evitamento de colisão, tratamento do erro e deslocamento relativo entre os robôs. Desta forma, o SRC é considerado generalizado com relação a quantidade de robôs e sua estrutura, com relação as tarefas e também com relação as características adicionais. Cabe ressaltar que a programação é realizada fora de linha e independente das características estruturais dos robôs e consequentemente do seu controlador. O requisito mínimo é que os robôs comecem a realizar as tarefas no mesmo instante de tempo.

A estrutura dos circuitos das cadeias cinemáticas que representam um SRC é sempre a mesma, ou seja, cada circuito possui a cadeia cinemática do robô, cadeia virtual CV_0 , que representa o movimento relativo entre a peça e o sistema de coordenadas fixo, a cadeia virtual CV_i , com $i > 1$, que relaciona o movimento do efetuador de cada robô sobre a peça. Essa forma mínima de representar um SRC permitiu criar uma rotina de programação de SRC. Além disso, os movimentos relativos são tratados de forma independente e orientados às tarefas.

Como contribuição secundária tem-se a especificação dos ambientes que envolvem a programação de SRC, como: estrutura robótica, tarefa e cinemática diferencial. Estes ambientes são, na verdade, o ponto de partida para a sistematização realizada.

Uma outra contribuição é a formalização matemática que prova a sistemática de adição de robôs e cadeias virtuais ao SRC. Essa formalização permitiu trabalhar com um SRC orientado à tarefa visualizando o sistema como um todo de forma mais simples. Assim, a programação de um SRC é organizada pela criação dos ambientes, pelo acréscimo de robôs ao sistema, pelas etapas do processo de programação (conforme visto no Capítulo 3) e completada pela formalização matemática (Capítulo 4).

A abordagem de um SRC composto por robôs e cadeias virtuais

de especificação das tarefas é tratada como sistema base. A formalização matemática realizada para ele permitiu criar um conceito chamado de características adicionais e assim identificar quais aspectos podem ser implementados em um SRC. Essas características adicionais (evitamento de colisão, controle de erro, e movimento relativo entre robôs) são tratadas como um complemento, por não ser necessário incluí-las em todos os SRCs. A aplicação delas depende das necessidades específicas das tarefas secundárias do SRC (como desvio de obstáculos). Além disso, novas características podem ser adicionadas baseadas nas Propriedades 1 e 2 de adição em série em adição em paralelo (Capítulo 4).

O desenvolvimento de uma plataforma de simulação de SRC é importante, visto que a programação e o planejamento de tarefas *on-line* de um SRC não fornece resultados prévios sobre colisão, velocidades de juntas e singularidades. Com a flexibilidade obtida pela metodologia foi possível utilizar a plataforma desenvolvida para avaliar o desempenho de um SRC através do estabelecimento de indicadores para avaliação e análise. Assim, são modificados alguns parâmetros de entrada do sistema, como configuração dos robôs, distância entre as bases, dados das tarefas, entre outros, sem dificuldades e avaliado a influência que essas alterações causam no resultado final da tarefa. Percebe-se que existem vários parâmetros que podem ser alterados e que por isso foram criados indicadores que permitam a sua avaliação. Nessa tese foram propostos três indicadores para análise do sistema, denominadas de velocidade máxima, picos de aceleração e deslocamento médio das juntas. Esses indicadores servem de exemplo e possibilitaram fazer verificar qual cenário é melhor com relação a outro.

A implementação em um sistema real proporcionou o contato e a experiência de trabalhar com os resultados simulados. Apesar de não se ter obtido um resultado de qualidade na execução da tarefa, foi possível observar que a programação de um SRC pode ser realizada rapidamente quando amparada pelo programa desenvolvido baseado na metodologia proposta. Os erros de execução, geralmente não comentados na literatura, se propagam mais facilmente em SRC quando comparado com um robô executando sua tarefa (um exemplo é o efeito alavanca, quando a peça possui grandes dimensões). Porém, a grande vantagem de se utilizar um SRC é que com ele é possível realizar tarefas que apenas um robô não consegue executá-la.

Por fim, as questões postas no Capítulo 1 são agora respondidas de forma geral. Conclui-se que existe, então, uma forma sistemática e generalizada, como apresentada e desenvolvida ao longo deste texto,

que permite programar um SRC envolvendo um número qualquer de robôs e tarefas. Além disso, a forma como a sistematização é realizada, possibilita calcular a cinemática de qualquer SRC com diferentes tipos de características adicionais de maneira simples e fornecendo a aberturas para a expansão da metodologia para novas abordagens. Na formalização matemática foi mostrada como é o crescimento da matriz de rede N ao se adicionar robôs no sistema, com essa formalização percebe-se quais são as novas variáveis e parâmetros a serem consideradas na programação de um SRC. Na seção 6.3 do Capítulo 6, a simulação envolvendo vários cenários mostrou que, dado um conjunto de robôs e tarefas, é possível trabalhar futuramente com a aplicação de teorias de otimização em SRC, com o objetivo de escolher qual o melhor robô para executar determinada tarefa, além de analisar quantos robôs são necessários realizar as tarefas especificadas.

8.2 TRABALHOS FUTUROS

O campo de pesquisa em SRC é amplo e com isso muitas perspectivas para trabalhos futuros podem ser citadas.

Com relação à programação é interessante implementar outras simulações de SRC com características adicionais, como controle do erro de integração, evitamento de colisão, entre outros. A composição de robôs móveis e robôs manipuladores pode ser viabilizada e é um caso de estudo que demanda certa atenção. Além disso, essa composição pode gerar novas tarefas cooperativas.

Um estudo para controle de determinadas juntas dos robôs pode ser realizado, verificando quando as juntas de robôs distintos estão próximas de colidirem. Através da metodologia discutida neste trabalho é possível conhecer as informações das juntas dos robôs, mas uma abordagem mais específica é necessária para tornar viável a comparação entre os dados de uma determinada junta com outra, através da adição em paralelo de cadeias virtuais.

A metodologia pode ser estendida para outros tipos de sistemas como robôs móveis, robôs subaquáticos, mãos robóticas, robôs paralelos, utilizando basicamente os conceitos apresentados como base para a modelagem.

Um aspecto importante que não foi considerado é a dinâmica. Assim, sugere-se fazer um estudo com relação à dinâmica em SRC e desenvolver uma metodologia para considerá-la, da mesma forma como foi feito para a cinemática.

Ao programa desenvolvido é necessário adicionar os limites de juntas, velocidades, a dinâmica, incluir mais robôs ao banco de dados e integrar um sistema gerador de trajetórias automático para as tarefas definidas no espaço de trabalho. Assim, a construção do ambiente de simulação é ampliada e novas tarefas podem ser incluídas e modificadas facilmente. O fluxo de trabalho para a programação das trajetórias dos robôs poderia então ser enriquecido com a possibilidade de testar facilmente uma maior combinação para a composição de SRC, o que ainda hoje é dificultado pela falta de ferramentas que lidam com as características dos sistemas cooperativos.

Sugere-se uma repetição dos testes práticos em robôs novos, com estudo e controle do erro de posição provocado pelo método de integração. Além disso, é interessante analisar o comportamento de SRC em uma aplicação real na indústria, visando a definição de uma tarefa com parâmetros que condizem com um processo industrial, fechando assim o ciclo de especificação da tarefa, planejamento, programação, parametrização, simulação e execução.

Verificou-se a importância de criar métodos para avaliar o desempenho de SRC, pois visto que, da mesma forma que o sistema é flexível no sentido de permitir inúmeras combinações de configurações, é necessário criar um método para avaliar os resultados das combinações, de forma a auxiliar no processo decisório. Assim, é de fundamental importância dar continuidade ao estudo de análise de desempenho de SRC, através do uso dos indicadores propostos nessa tese, além da definição de outros que convierem a depender do objetivo desejado. A quantidade de parâmetros em um SRC é grande e a sua variação pode resultar em melhor desempenho do sistema, sob algum critério específico. Essa estratégia é pouco comentada na literatura e neste trabalho foi mostrado que bons resultados podem ser obtidos com a variação mínima dos parâmetros e avaliação de indicadores, que concentram os resultados globais em um conjunto compacto de valores numéricos.

Uma outra pesquisa interessante é a propagação do erro em SRC, avaliando quais as influências desses erros na qualidade da tarefa e na estrutura do SRC. Como o sistema possui muitos robôs interagindo entre si, o erro de posição, por menor que seja, pode se propagar pelas juntas e pela peça a ser trabalhada, chegando ao efetuador de outro robô maior. Este fator, muitas vezes ignorado, pode provocar colisões, baixar a qualidade na tarefa, ou mesmo inviabilizar a execução da tarefa. Assim, um estudo aprofundado considerando esse aspecto ainda é pouco explorado na literatura, apesar de ser de fundamental importância para a aplicação prática dos SRC.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABB. 2011. Acesso em 09/04/2011. Disponível em:
<<http://www.abb.com/industries/>>.
- BELFIORE, N. P.; BENEDETTO, A. D. Connectivity and redundancy in spatial robots. *The International Journal of Robotics Research*, Sage Publications, Inc, v. 19, n. 12, p. 1245–1261, December 2000.
- BONDY, J.; MURTY, U. *Graduate Texts in Mathematics: Graph Theory*. Heidelberg: Springer, 2008. (Microprocessor-Based and Intelligent Systems Engineering, v. 244).
- CACCAVALE, F.; UCHIYAMA, M. Cooperative manipulators. In: _____. *Handbook of Robotics*. 1. ed. Berlin: Spring-Verlag, 2008. cap. 29, p. 701–718. ISBN 978-3-540-23957-4.
- CAMPOS, A. *Cinemat́ica diferencial de manipuladores empregando cadeias virtuais*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis-SC, março 2004.
- CAMPOS, A.; GUENTHER, R.; MARTINS, D. Differential kinematics of serial manipulators using virtual chains. *Brazilian Society of Mechanical Sciences and Engineering*, XXVII, n. 4, p. 345 – 356, October – December 2005. ABCM.
- CARBONI, A. P. *Análise Conceitual de estruturas cinemáticas planas e espaciais*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, Fevereiro 2008.
- CAZANGI, H. R. *Aplicação do método de Davies para análise cinemática e estática de mecanismo com múltiplos graus de liberdade*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2008.
- CECCARELLI, M. Screw axis defined by giulio mozzi in 1763 and early studies on helicoidal motion. *Mechanism and Machine Theory*, v. 35, n. 6, p. 761 – 770, 2000. ISSN 0094-114X. Disponível em:
<<http://www.sciencedirect.com/science/article/pii/S0094114X99000464>>.
- CHEN, W. *Applied graph theory*. Amsterdam: North-Holland Pub. Co., 1971. (Applied mathematics and mechanics, v. 13).

CHEN, Y. D.; NI, J.; WU, S. Real-time cnc tool path generation for machining iges surfaces. *Journal of Engineering for Industry*, v. 115, p. 480–486, november 1993.

CHUANG, J.-H.; LIN, C.-C.; CHOU, T.-H. An alternate priority planning algorithm for dual-arm systems. In: *Proc. IEEE International Symposium on Industrial Electronics*. Montreal - Quebec: IEEE, 2006. v. 4, p. 3084–3089.

CORMEN, T. H. et al. *Introduction to Algorithms*. 2nd. ed. Cambridge - Massachusetts: The MIT Press, 2001. ISBN 0-262-03293-7.

DAVIES, T. H. Kirchhoff's circulation law applied to multi-loop kinematic chains. *Mechanism and Machine Theory*, v. 16, n. 3, p. 171 – 183, 1981. ISSN 0094-114X.

DAVIES, T. H. The 1887 committee meets again.subject: freedom and constraint. In: BALL 2000 CONFERENCE. *A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on the Theory of Screws*. University of Cambridge, Trinity College, 2000. p. 1–56.

DIESTEL, R. *Graph Theory*. Heidelberg: Electronic Edition, 2005. (Microprocessor-Based and Intelligent Systems Engineering, v. 30).

DOURADO, A. O. *Cinemat́ica de Robˆos Cooperativos*. Dissertaçˆo (Mestrado) — Universidade Federal de Santa Catarina, Florianˆopolis-SC, Junho 2005.

DOURADO, A. O.; MARTINS, D. *Método de sincronizaçˆo de múltiplos mecanismos programáveis e célula de manufatura cooperativa*. Florianópolis: [s.n.], 2005. Online: www.patentesonline.com.br. PI0502438-2. Disponível em: <http://www.patentesonline.com.br/metodo-de-sincronizacao-de-multiplos-mecanismos-programaveis-e-celula-de-manufatura-12124.html>.

EATON, J. W. *GNU Octave*. 2011. Acesso em 14/04/2011. Disponível em: <http://www.gnu.org/software/octave/>.

FENG, H.-Y.; LI, H. Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer-Aided Design*, Elsevier Ltd, n. 34, p. 647 – 654, 2001.

FERREIRA, A. B. de H. *Dicionário Aurélio*. 2004. Positivo Informática Ltda. Acesso em 05/08/2009. Disponível em: <<http://aurelio.ig.com.br/>>.

FONTAN, D. *Implementação da cinemática inversa de robôs redundantes operando em ambientes confinados no projeto Roboturb*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis-SC, Setembro 2007.

FREESE, M. *V-Rep. Virtual Robot Experimentation Platform*. 2011. Acesso em 23/05/2011. Disponível em: <<http://www.v-rep.eu/>>.

GASPARETTO, A.; ZANOTTO, V. A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput.-Integr. Manuf.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 24, n. 3, p. 415–426, 2008. ISSN 0736-5845.

GPL. *GNU GENERAL PUBLIC LICENSE*. 29 June 2007. Free Software Foundation, Inc. Version 3. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>.

GUENTHER, R. et al. A new integration method for differential inverse kinematics of closed-chain robots. In: ABCM (Ed.). *ABCM Symposium Series in Mechatronics*. Rio de Janeiro - RJ: Brazilian Society of Mechanical Sciences and Engineering, 2008. v. 3, p. 225–235. ISBN 978-85-85769-38-3.

HUNT, K. K. Don't cross-thread the screw. In: BALL 2000 CONFERENCE. *A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on the Theory of Screws*. University of Cambridge, Trinity College, 2000. p. 1–37.

IFR. *World Robotics 2010 - Executive Summary 2010*. 2010. Acesso em 14/03/2010. Disponível em: <<http://www.worldrobotics.org/downloads/>>.

JU, M.-Y.; LIU, J.-S.; HWANG, K.-S. Real-time velocity alteration strategy for collision-free trajectory planning of two articulated robot manipulators. *J. Intell. Robotics Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 33, n. 2, p. 167–186, 2002. ISSN 0921-0296.

JUTARD-MALINGE, A.-D.; BESSONNET, G. Optimal motion planning of robotic manipulators removing mobile objects grasped in motion. *J. Intell. Robotics Syst.*, Kluwer Academic Publishers, Hingham,

MA, USA, v. 29, p. 233–255, November 2000. ISSN 0921-0296. Disponível em: <<http://portal.acm.org/citation.cfm?id=595593.595695>>.

KUKA. *KUKA.SIM PRO*. 2001. Acesso em 24/05/2011. Disponível em: <<http://www.kuka-robotics.com/en/products/software/>>.

KUKA. *Robots bend large heavy metal plates*. 2011. Acesso em 09/04/2011. Disponível em: <<http://www.kuka.com/en/solutions/>>.

KWON, W.; LEE, B. H. A new optimal force distribution scheme of multiple cooperating robots using dual method. *J. Intell. Robotics Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 21, n. 4, p. 301–326, 1998. ISSN 0921-0296.

LAI. *Laboratório de Automação Industrial – LAI*. 2010. Acesso em 10/12/2010. Disponível em: <<http://www.das.ufsc.br>>.

LAVALLE, S. M. *Planning Algorithms*. Cambridge University Press, 2006. Disponível em: <<http://planning.cs.uiuc.edu/>>.

LEWIS, C. Trajectory generation for two robots cooperating to perform a task. In: *Proc. IEEE International Conference on Robotics and Automation*. Minneapolis, MN, USA: IEEE, 1996. v. 2, p. 1626–1631.

LEWIS, C.; MACIEJEWSKI, A. Trajectory generation for cooperating robots. In: *Proc. IEEE International Conference on Systems Engineering*. Pittsburgh, PA: IEEE, 1990. p. 300–303.

MARTINS, D. *Análise Cinemática Hierárquica de Robôs Manipuladores*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, Fevereiro 2002.

MARTINS, D.; CARBONI, A. P. Variety and connectivity in kinematic chains. *Mechanism and Machine Theory*, v. 43, n. 10, p. 1236 – 1252, 2008. ISSN 0094-114X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0094114X07001735>>.

MARUSHIMA, S.; YAMAMOTO, M.; MOHRI, A. A planning method of collision-free trajectory for two manipulators. In: *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems IROS '91*. Osaka, Japan: IEEE, 1991. v. 2, p. 547 – 552.

MAYEDA, W. *Graph theory*. New York: Wiley-Interscience, 1972.

MOHRI, A.; YAMAMOTO, M.; HIRANO, G. Cooperative path planning for two manipulators. In: *Proc. IEEE International Conference on Robotics and Automation*. Minneapolis, Minnesota, USA: IEEE, 1996. v. 3, p. 2853–2858. ISBN 0-7803-2988-0.

MOHRI, A.; YAMAMOTO, M.; MARUSHIMA, S. Collision-free trajectory planning for two manipulators using virtual coordination space. In: *Proc. IEEE International Conference on Robotics and Automation*. Atlanta, GA: IEEE, 1993. v. 2, p. 674–679. ISBN 0-8186-3450-2.

MOTOMAN. *MotoSim® EG: Motoman Simulator Enhanced Graphics*. 2011. Acesso em 24/05/2011. Disponível em: <<http://www.motoman.com/products/software/>>.

NAVARRO-ALARCON, D. et al. Dexterous cooperative manipulation with redundant robot arms. In: *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Berlin, Heidelberg: Springer-Verlag, 2009. (CIARP '09), p. 910–917. ISBN 978-3-642-10267-7. Disponível em: <http://dx.doi.org/10.1007/978-3-642-10268-4_107>.

OWEN, W.; CROFT, E.; BENHABIB, B. A multi-arm robotic system for optimal sculpting. *Robotics and Computer-Integrated Manufacturing*, Elsevier Ltd, v. 24, n. 1, p. 92 – 104, 2008. ISSN 0736-5845. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0736584506001050>>.

OWEN, W. S.; CROFT, E. A.; BENHABIB, B. Minimally compliant trajectory resolution for robotic machining. In: *Proceedings of ICAR 2003: The 11th International Conference on Advanced Robotics*. Coimbra, Portugal: IEEE, 2003. p. 702–707.

PAVIOT, T. *Python OCC*. 2011. Acesso em 14/04/2011. Disponível em: <<http://www.pythonocc.org/>>.

RHINO. *Rhino Robotics Ltd*. 2010. Acesso em 10/12/2010. Disponível em: <<http://www.rhinorobotics.com/>>.

RIBEIRO, L.; GUENTHER, R.; MARTINS, D. *Screw-based relative jacobian for manipulators cooperating in a task*. Brasil: ABCM – Associação Brasileira de Engenharia e Ciências Mecânicas, 2008. 276-285 p. (ABCM Symposium Series Mechatronics, v. 3). ISBN 978-85-85769-38-3.

RIBEIRO, L.; MARTINS, D. *Screw-Based relative Jacobian for manipulators cooperating in a task using Assur virtual chains*. Brasil: ABCM – Associação Brasileira de Engenharia e Ciências Mecânicas, 2010. 729-738 p. (ABCM Symposium Series Mechatronics, v. 4).

RIBEIRO, L. P. G. *Modelagem Cinemática de Sistemas Robóticos Cooperativos: Proposta de um Jacobiano de Cooperação*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis-SC, Agosto 2010. Não Publicado.

ROBOTURB. 2006. Acesso em 13/04/2009. Disponível em: <<http://www.robotica.ufsc.br/tiki-index.php?page=ROBOTURB>>.

ROCHA, C.; TONETTO, C.; DIAS, A. A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, v. 27, n. 4, p. 723 – 728, 2011. ISSN 0736-5845. Conference papers of Flexible Automation and Intelligent Manufacturing - Intelligent manufacturing and services. Disponível em: <<http://www.sciencedirect.com/science/article/B6V4P-5253M0P-1/2/59315aa37dfe13dacb54187701f956b3>>.

SANTOS, C. H. F. d. et al. Virtual kinematic chains to solve the underwater vehicle-manipulator systems redundancy. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, scielo, v. 28, p. 354 – 361, 09 2006. ISSN 1678-5878.

SANTOS, C. H. F. dos. *Movimento coordenado de sistemas veículo-manipulador submarinos utilizando técnicas de inteligência artificial e sistemas híbridos*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis-SC, Junho 2006.

SARMA, R.; DUTTA, D. The geometry and generation of nc tool paths. *Journal of Mechanical Design*, v. 119, p. 253 – 258, June 1997.

SESHU, S.; REED, M. *Linear graphs and electrical networks*. London, England: Addison-Wesley Reading, Mass., 1961.

SICILIANO, B.; KHATIB, O. (Ed.). *Springer Handbook of Robotics*. Berlin: Springer-Verlag, 2008.

SICILIANO, B. et al. *Robotics: Modelling, Planning and Control*. London: Springer, 2009. 632 p. (Advanced Textbooks in Control and Signal Processing, XXIV). ISBN 978-1-84628-641-4.

SIMAS, H. *Planejamento de trajetórias e evitamento de colisão em tarefas de manipuladores redundantes operando em ambientes confinados*. Tese (Doutorado) — Universidade Federal de Santa Catarina - UFSC, Fevereiro 2008.

SIMAS, H.; DIAS, A.; GUENTHER, R. *A scallop-height based algorithm to compute parallel paths on parametric surfaces*. Rio de Janeiro - RJ: Brazilian Society of Mechanical Sciences and Engineering, 2008. 326-335 p. ISBN 978-85-85769-38-3.

SIMAS, H. et al. *A Collision Avoidance Method Using Assur Virtual Chains*. Rio de Janeiro - RJ: Brazilian Society of Mechanical Sciences and Engineering, 2008. 316-325 p. ISBN 978-85-85769-38-3.

SIMAS, H. et al. A new method to solve robot inverse kinematics using assur virtual chains. *Robotica*, v. 27, p. 1017–1026, march 2009.

SIMONI, R. *Síntese estrutural de cadeias cinemáticas e mecanismos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis-Brasil, Fevereiro 2008.

SOLUTIONS, W. *Workspace: Robot Offline Programming and Simulation Software*. 2011. Acesso em 23/05/2011. Disponível em: <<http://www.workspacelt.com/>>.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot modeling and control*. New Jersey: John Wiley & Sons, 2006. ISBN 978-0-471-64990-8.

SURESH, K.; YANG, D. C. H. Constant scallop-height machining of free-form surfaces. *Journal of Engineering for Industry*, v. 116, n. 2, p. 253 – 259, May 1994.

TAVASOLI, A.; EGHTEHAD, M.; JAFARIAN, H. Two-time scale control and observer design for trajectory tracking of two cooperating robot manipulators moving a flexible beam. In: . Amsterdam, Netherlands: North-Holland Publishing Co., 2009. v. 57, p. 212–221. ISSN 0921-8890. Disponível em: <<http://portal.acm.org/citation.cfm?id=1501036.1501304>>.

TONETTO, C. P. *Uma proposta de sistematização do processo de planejamento de trajetórias para o desenvolvimento de tarefas de robôs manipuladores*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis-SC, Março 2007.

TONETTO, C. P.; DIAS, A. Geração de caminhos da ferramenta sobre superfícies de forma livre. In: CIBIM8. *Anais do 8 Congresso Iberoamericano de Ingeniería Mecánica*. Cusco, Peru, 2007. ISBN 978-9972-2885-3-1. Pontificia Universidad Católica del Perú and Federación Iberoamericana de Ingeniería Mecánica.

TONETTO, C. P.; DIAS, A. An application prototype for offline continuous tool path planning on rebuilt surfaces. In: *Proceedings of the 18th International Conference on Flexible Automation and Intelligent Manufacturing*. Skovde, Sweden: FAIM, 2008.

TONETTO, C. P.; DIAS, A. Cooperative robot kinematics to perform load sharing tasks. In: TEESSIDE UNIVERSITY. *Proceedings of the 19th International Conference on Flexible Automation and Intelligent Manufacturing*. Middlesbrough: Gemini, 2009. p. 130 – 138. ISSN 9780956.

TONETTO, C. P.; DIAS, A. *Trajectory Planning with Redundant Cooperative Robotics Systems*. Brasil: ABCM – Associação Brasileira de Engenharia e Ciências Mecânicas, 2010. 710-718 p. (ABCM Symposium Series Mechatronics, v. 4). ISBN 978-85-85769-47-5.

TONETTO, C. P.; ROCHA, C. R.; DIAS, A. Simulation of multi-robot cooperative systems programming based on a three environment definition. In: THE 12TH MECHATRONICS FORUM BIENNIAL INTERNATIONAL CONFERENCE. *Proceedings of the 12th Mechatronics Forum Biennial International Conference*. Zurich - Switzerland: Swiss Federal Institute of Technology - ETH, 2010. ISBN 078-3-033-02507-3.

TONETTO, C. P.; SIMAS, H.; DIAS, A. Sistemática procedural para o processo de cálculo da cinemática de sistemas multirrobo cooperativos. In: ABCM. *Anais do VI Congresso Nacional de Engenharia Mecânica*. Campina Grande - Paraíba, 2010. ISSN 1278-180X.

TSAI, L.-W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. New York: Jonh Wiley & Sons, INC, 1999. ISBN 0-471-32593-7.

TSAI, L.-W. *Mechanism Design: Enumeration of Kinematic Structures According to Function*. USA: CRC Press LLC, 2001. (Mechanical Engineering).

TZAFESTAS, C. S.; PROKOPIOU, P. A.; TZAFESTAS, S. G. Path planning and control of a cooperative three-robot system manipulating large objects. *Journal of Intelligent & Robotic Systems*, Springer Netherlands, v. 22, p. 99–116, 1998. ISSN 0921-0296. 10.1023/A:1007943632269. Disponível em: <<http://dx.doi.org/10.1023/A:1007943632269>>.

YAMANO, M. et al. Cooperative control of a 3D dual-flexible-arm robot. *J. Intell. Robotics Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 39, n. 1, p. 1–15, 2004. ISSN 0921-0296.

YIM, W.; SELVARAJAN, M.; WELLS, W. R. Sliding mode cooperative motion control of dual arm manipulators. *Artificial Life and Robotics*, v. 3, n. 3, p. 166–169, September 1999.

ZHA, X. F. Optimal pose trajectory planning for robot manipulators. *Mechanism and Machine Theory*, v. 37, p. 1063–1086, May 2002.

ZINANOVIC, M. D.; VUKOBRATOVIC, M. K. *Multi-Arm Cooperating Robots: Dynamics and Control*. Netherlands: Springer, 2006. (Microprocessor-Based and Intelligent Systems Engineering, v. 30).

APÊNDICE A – Cinemática diferencial de robôs

Nesse apêndice é comentado brevemente sobre o cálculo da cinemática diferencial de sistemas compostos por apenas um robô. A finalidade é ajudar na leitura do Capítulo 3 sobre programação cinemática de SRC.

A programação de robôs manipuladores envolve o controle do movimento de juntas, que pode ser feito de dois modos: pelo uso da cinemática direta e pela cinemática inversa (Figura 57). Segundo Siciliano et al. (2009), a cinemática direta permite que a orientação e a posição do efetuador sejam armazenadas como funções das variáveis de juntas da estrutura mecânica com respeito a um sistema de coordenadas, para o movimento do efetuador. Já a cinemática inversa consiste na determinação das variáveis de juntas correspondentes para cada posição e orientação do efetuador do robô.

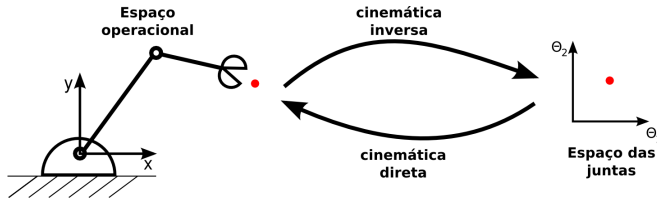


Figura 57: Cinemática direta e inversa.

Na cinemática direta o cálculo da posição do efetuador é obtido de maneira única, uma vez que são conhecidas as variáveis de juntas. Por outro lado, o cálculo da cinemática inversa é mais complexo, pois as equações que a descrevem são em geral não-lineares, podendo ou não existir solução, e se existir solução, existir múltiplas ou infinitas.

As informações da tarefa a ser executada pelo robô são dadas em função dos parâmetros, caminho e trajetória, no espaço operacional. Deve-se, então, utilizar a cinemática inversa na programação e execução de tarefas robóticas, pois, assim, bastaria fornecer a posição/orientação do efetuador do robô.

A relação entre o espaço das juntas e o espaço operacional pode ser dada pela equação:

$$x = k(q) \quad (\text{A.1})$$

em que x representa a posição e orientação do efetuador. O vetor $x = [p, \varphi]^T$ pode ser representado em função de p que descreve a posição e φ a orientação do efetuador. O vetor q descreve a configuração do espaço de juntas ($q = [q_1, q_2, \dots, q_n]$ e n é o número de juntas). O vetor

função $k(\cdot)$ permite o cálculo das variáveis do espaço operacional quando conhecido o vetor que descreve a configuração do espaço das juntas (SICILIANO et al., 2009; SICILIANO; KHATIB, 2008).

A abordagem descrita acima é conhecida como cinemática de posição e estabelece a relação entre as variáveis das juntas e a posição/orientação do efetuador. Uma outra alternativa é definir as trajetórias em função das velocidades ao invés das posições e assim, fazer um mapeamento entre as velocidades das juntas e a velocidade linear e angular do efetuador. Deste modo, no chamado modelo cinemático diferencial, tem-se um controle mais direto da execução da trajetória do manipulador no tempo e ainda pode-se obter as posições pela integração das velocidades.

A relação que a cinemática diferencial estabelece entre as velocidades das juntas e as velocidades do efetuador pode ser representada por uma matriz denominada Jacobiano (J) que depende da configuração instantânea do manipulador. O Jacobiano para manipuladores seriais é obtido geralmente através de dois métodos: um baseado na convenção de Denavit-Hartenberg e outro baseado na teoria de Helicoides (CAMPOS, 2004).

O método de Denavit-Hartenberg é uma convenção sistemática para definir a posição e orientação relativa de dois elos consecutivos. A técnica consiste em fixar dois sistemas de coordenadas em dois elos consecutivos e calcular a transformação de coordenadas entre eles. Porém, este método restringe a representação da velocidade do efetuador no sistema de coordenadas da base. A convenção de Denavit-Hartenberg e as etapas de fixação dos sistemas de coordenadas dos elos são descritas com mais detalhes em Siciliano et al. (2009), e têm sido descritos extensamente na literatura de robótica.

A cinemática diferencial de manipuladores pode ser estudada usando a teoria de helicoides, e esta é uma importante ferramenta na análise das características cinemáticas de mecanismos. Além disso, ela descreve a natureza do movimento e da estática de um corpo rígido, sendo, conseqüentemente de grande aplicação na robótica e cinemática. Assim, a seguir será comentado de maneira breve a teoria de helicoides (uma revisão mais completa pode ser encontrada nos trabalhos de Tsai (1999), Campos (2004) e Simas (2008)). Uma visão geral da teoria é descrita a fim de obter os resultados a serem aplicados no trabalho.

O teorema de Chasles estabelece que o deslocamento de um corpo rígido pode ser representado por uma rotação em torno de um eixo fixo no espaço 3D e uma translação ao longo do mesmo eixo. A combinação do movimento de translação e rotação simultânea de um corpo é

chamada de **deslocamento helicoidal** (TSAI, 1999). Com o deslocamento helicoidal pode-se representar tanto deslocamentos rotativos quanto prismáticos comuns no movimento de juntas de manipuladores. É possível, assim, representar os dois diferentes tipos de juntas (rotativas e prismáticas), que definem o deslocamento relativo entre os elos dos manipuladores, de maneira única, o que torna possível tratar as questões de posição e orientação do efetuador de uma maneira generalizada.

A Figura 58 ilustra o deslocamento de um ponto P no espaço e os vetores associados a ele relativos a um sistema de coordenadas de referência O . O deslocamento helicoidal de um ponto P no espaço da posição P_1 para a posição P_2 é dado pelo registro de uma rotação de um ângulo θ sobre o eixo de helicoides s seguido por uma translação t ao longo do mesmo eixo. A rotação transforma o ponto P de P_1 para p_2^r e a translação transforma o ponto P de p_2^r para P_2 .

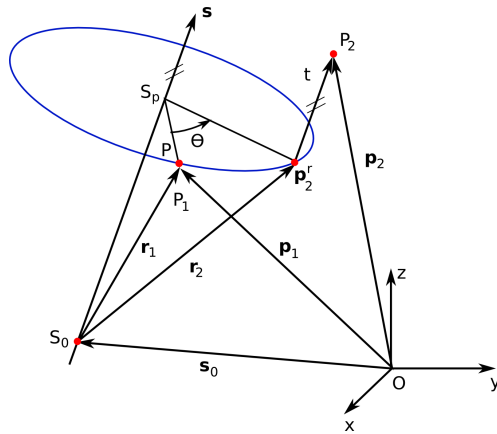


Figura 58: Deslocamento helicoidal de um ponto P (TSAI, 1999).

Considerando ainda a Figura 58, $s = [s_x, s_y, s_z]$ denota o vetor unitário ao longo da direção do eixo de helicoides e $s_o = [s_{ox}, s_{oy}, s_{oz}]$ denota o vetor posição de um ponto que está no eixo de helicoides com relação ao sistema de coordenadas de referência O . O ponto S_p é o ponto de intersecção entre o plano que contém os pontos P_1 e p_2^r e o eixo s . Uma vez que a transformação do ponto P_1 para p_2^r é uma rotação em torno do eixo s , o plano é normal ao eixo.

O vetor p_1 indica a posição inicial do ponto P em relação ao sistema de origem O , os vetores r_1 e r_2 são vetores de mesma norma,

com origem em S_0 , que determinam a posição inicial e final do ponto P referente a rotação θ . O vetor p_2 indica a localização do ponto P após sofrer uma translação t , referente ao sistema de coordenadas O . Desta forma, foi apresentado o deslocamento helicoidal de um ponto que pode ser pertencente a um corpo rígido e assim descrever o deslocamento desse corpo no espaço. Em um sistema composto por vários corpos rígidos que sofrerão deslocamentos, o vetor s_i denotará o vetor unitário que determina a direção de translação e rotação do eixo helicoidal, em que i é referente ao elo ou corpo fixo. O vetor s_{oi} indicará o vetor posição de s_i em relação a um sistema de coordenadas fixo a um elo de referência. O ângulo θ é o ângulo de rotação sobre o ponto P e t é o comprimento do deslocamento de translação do ponto P (TSAI, 1999; SIMAS, 2008). Por conveniência, os vetores s_i e s_{oi} devem ser escolhidos perpendiculares, ou seja:

$$s_{oi}^T s_i = 0 \quad (\text{A.2})$$

O teorema de Mozzi diz que as velocidades dos pontos de um corpo rígido em relação a um sistema de referência $O(x, y, z)$ podem ser representadas por uma rotação diferencial em torno de um eixo fixo e uma translação diferencial simultânea em torno do mesmo eixo (CAMPOS, 2004). Assim, qualquer movimento no espaço de um corpo rígido pode ser tratado como um movimento helicoidal, segundo um sistema de coordenadas de referência.

Dado que o movimento instantâneo completo de um corpo rígido em relação a um sistema inercial é composto por um par de vetores $(\omega, v_p)^T$, no qual $\omega = (\omega_x, \omega_y, \omega_z) = (\mathcal{L}, \mathcal{M}, \mathcal{N})$ representa a velocidade angular do corpo em relação ao sistema de coordenadas escolhido. O vetor $v_p = (v_{px}, v_{py}, v_{pz}) = (\mathcal{P}^*, \mathcal{Q}^*, \mathcal{R}^*)$ que representa a velocidade linear de um ponto P pertencente ao corpo passando instantaneamente pela origem do sistema de referência (CAMPOS, 2004).

O movimento instantâneo pode ser decomposto em amplitude e eixo normalizado, quando existir somente movimento rotacional ou somente movimento translacional. Caso o movimento seja de rotação, a amplitude é a magnitude da velocidade angular do corpo $\|\omega\|$. Caso o movimento seja de translação, a amplitude é a magnitude da velocidade linear $\|v_p\|$.

O movimento instantâneo dado por $(\mathcal{L}, \mathcal{M}, \mathcal{N}, \mathcal{P}^*, \mathcal{Q}^*, \mathcal{R}^*)^T$ pode ser representado pelo eixo normalizado $(L, M, N, P^*, Q^*, R^*)^T$ e sua

magnitude \dot{q} , dada pela equação (CAMPOS, 2004):

$$\begin{bmatrix} \mathcal{L} \\ \mathcal{M} \\ \mathcal{N} \\ \mathcal{P}^* \\ \mathcal{Q}^* \\ \mathcal{R}^* \end{bmatrix} = \begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} \dot{q} \quad (\text{A.3})$$

Por outro lado, um helicóide \mathcal{S} é um elemento geométrico definido por uma reta direcionada no espaço e por um parâmetro escalar com unidade de comprimento h que define o passo do helicóide. O eixo helicoidal normalizado $\hat{\mathcal{S}}$ de uma junta i é dado pela equação:

$$\hat{\mathcal{S}} = \begin{bmatrix} s_i \\ s_{oi} \times s_i + h s_i \end{bmatrix} \quad (\text{A.4})$$

em que, s_i é o vetor unitário que determina a direção do eixo de translação e rotação referente ao movimento helicoidal. O vetor s_{oi} indica a posição do vetor s_i em relação a um sistema de coordenadas fixo, h é o passo do helicóide e $s_{oi} \times s_i$ é o produto vetorial entre os vetores s_{oi} e s_i .

Seja o movimento de rotação dado em torno do eixo s_i e de translação no sentido do eixo s_i , o helicóide que representa o movimento instantâneo, pode ser expresso pela equação:

$$\hat{\mathcal{S}} = \begin{bmatrix} L \\ M \\ N \\ P^* \\ Q^* \\ R^* \end{bmatrix} \quad (\text{A.5})$$

Desta forma, o movimento helicoidal pode ser representado pelo helicóide \mathcal{S} , e escrito na equação:

$$\mathcal{S} = \hat{\mathcal{S}} \dot{q} \quad (\text{A.6})$$

O uso de helicóides permite associar o deslocamento diferencial entre dois corpos em relação a um sistema de coordenadas de referência. O deslocamento diferencial entre dois elos de uma cadeia cinemática geralmente está relacionado a uma junta rotativa ou uma junta prismática. Quando se trata de outros tipos de juntas, estas podem ser decompostas em juntas simples com um grau de liberdade, como,

por exemplo, juntas esféricas podem ser decompostas em três juntas rotativas.

Quando o movimento é somente relacionado a uma junta rotativa, não existe translação, ou seja, o passo do helicóide é nulo ($h = 0$), então, o deslocamento helicoidal é dado pela igualdade da equação (CAMPOS, 2004):

$$\hat{\$} = \begin{bmatrix} s_i \\ s_{oi} \times s_i \end{bmatrix} \quad (\text{A.7})$$

Para ilustrar o uso destas definições na aplicação para manipuladores, será feito um exemplo onde serão determinados os deslocamentos helicoidais para um robô planar com somente juntas rotativas.

A Figura 59 ilustra um manipulador planar com dois elos. Considere a primeira junta na origem do sistema de coordenadas cartesianas, assim, o valor de $s_1 = (0, 0, 1)^T$, o que significa que a direção do eixo de rotação é dado pelo vetor perpendicular ao plano XY do robô planar e $s_{o1} = (0, 0, 0)^T$, significa que o vetor posição do primeiro elo de s_1 coincide com o sistema de coordenadas de referência. Neste caso, o sistema de referência está na base do manipulador, na origem do sistema de coordenadas x_0 e y_0 . Assim, substituindo os valores de s_1 e s_{o1} na igualdade A.7, tem-se que o eixo helicoidal normalizado da junta 1 é dado pela equação A.8.

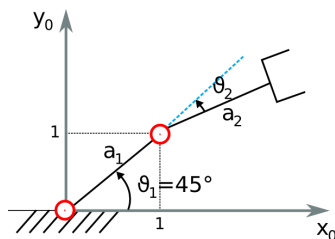


Figura 59: Manipulador planar de dois elos com duas juntas rotativas.

$$\hat{\$}_1 = \begin{bmatrix} (0, 0, 1)^T \\ (0, 0, 0)^T \times (0, 0, 1)^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.8})$$

Para a segunda junta do manipulador tem-se $s_2 = (0, 0, 1)^T$ e

$s_{o_2} = (a_1 \cos(\vartheta_1), a_1 \sin(\vartheta_1), 0)^T$. Neste caso, o vetor posição de s_2 do segundo elo está sobre a junta 2 do manipulador. Substituindo-se os valores novamente em A.7, tem-se o eixo helicoidal normalizado da junta 2 dado por:

$$\hat{s}_2 = \left[\begin{array}{c} (0, 0, 1)^T \\ (a_1 \cos(\vartheta_1), a_1 \sin(\vartheta_1), 0)^T \times (0, 0, 1)^T \end{array} \right] = \begin{bmatrix} 0 \\ 0 \\ 1 \\ a_1 \sin(\vartheta_1) \\ -a_1 \cos(\vartheta_1) \\ 0 \end{bmatrix} \quad (\text{A.9})$$

Quando se usa uma junta prismática, não existe movimento de rotação ($\theta = 0$). Tendo-se somente translação e assim, o passo de helicoides é infinito ($h = \infty$). O deslocamento helicoidal é dado pela relação:

$$\hat{s} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} \quad (\text{A.10})$$

Um exemplo de um manipulador planar com duas juntas prismáticas e uma rotativa é ilustrado na Figura 60. Para calcular o movimento helicoidal desse manipulador é necessário apenas encontrar o valor de s_i para as juntas prismáticas. Desta forma, para a junta P_X , $s_{P_X} = (1, 0, 0)^T$, o que significa que a direção do movimento de translação ocorre na direção do eixo x e para a junta P_Y tem-se $s_{P_Y} = (0, 1, 0)^T$, ocorre na direção do eixo y . Já, para a junta rotativa R_Z é necessário os valores de s_{oi} e s_i e substituí-los na equação A.7, sendo $s_{o_{R_z}} = (P_X, P_Y, 0)$ e $s_{R_z} = (0, 0, 1)^T$. Então substituindo os valores na relação A.10 e A.7 são obtidas as igualdades das equações descritas em A.11.

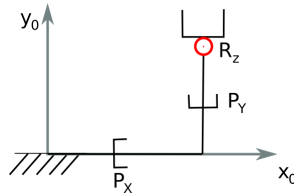


Figura 60: Manipulador planar com duas juntas prismáticas e uma rotativa.

$$\hat{\$}_{P_X} = \begin{bmatrix} (0, 0, 0)^T \\ (1, 0, 0)^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \quad \hat{\$}_{P_Y} = \begin{bmatrix} (0, 0, 0)^T \\ (0, 1, 0)^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix};$$

$$\hat{\$}_{R_Z} = \begin{bmatrix} (0, 0, 1)^T \\ (P_X, P_Y, 0)^T \times (0, 0, 1)^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ P_Y \\ -P_X \\ 0 \end{bmatrix} \quad (\text{A.11})$$

O helicóide resultante do deslocamento do efetuador pode ser obtido somando-se linearmente os movimentos helicoidais de cada uma das juntas, como pode ser visto na relação:

$$\$E = \sum_{i=1}^n \hat{\$}_i \dot{q}_i \quad (\text{A.12})$$

onde $\hat{\$}_i$ é o i -ésimo helicóide normalizado descrito em relação ao sistema de coordenadas de referência, \dot{q}_i é a magnitude correspondente ao corpo rígido i e $\$E$ é o helicóide resultante que representa o movimento do efetuador com relação ao sistema de coordenadas de referência.

Para o exemplo com as juntas rotativas (ilustrado na Figura 59) o deslocamento do efetuador é dado pela relação A.13, substituindo os valores de $\hat{\$}_1$ e $\hat{\$}_2$ obtidos anteriormente nas relações A.8 e A.9, respectivamente.

$$\$E = \hat{\$}_1 \dot{q}_1 + \hat{\$}_2 \dot{q}_2$$

$$\$E = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dot{q}_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ a_1 \sin(\vartheta_1) \\ -a_1 \cos(\vartheta_1) \\ 0 \end{bmatrix} \dot{q}_2 \quad (\text{A.13})$$

em que $\$E$ é o helicóide resultante do efetuador com relação ao sistema de coordenadas de referência, \dot{q}_1 e \dot{q}_2 são as magnitudes da velocidade

angular das juntas 1 e 2 respectivamente.

Para o exemplo com as juntas prismáticas (Figura 60) o deslocamento do efetuador é dado pela igualdade A.14, substituindo os valores de $\hat{\$}_{P_X}$, $\hat{\$}_{P_Y}$ e $\hat{\$}_{R_Z}$ obtidos em A.11.

$$\begin{aligned} \$_E &= \hat{\$}_{P_X} \dot{q}_{P_X} + \hat{\$}_{P_Y} \dot{q}_{P_Y} + \hat{\$}_{R_Z} \dot{q}_{R_Z} \\ \$_E &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \dot{q}_{P_X} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \dot{q}_{P_Y} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ P_Y \\ -P_X \\ 0 \end{bmatrix} \dot{q}_{R_Z} \end{aligned} \quad (\text{A.14})$$

em que $\$_E$ é o helicóide resultante do efetuador com relação ao sistema de coordenadas de referência, \dot{q}_{P_X} e \dot{q}_{P_Y} são as magnitudes da velocidade linear das juntas P_X e P_Y , e \dot{q}_{R_Z} é a magnitude da velocidade angular da junta R_Z .

Dessa forma, tem-se os deslocamentos das juntas representados através de helicóides normalizados e suas magnitudes.

**APÊNDICE B – Aspectos Gerais: Transformações de
Helicoides, Método de Davies e Cadeias Virtuais de Assur**

B.1 TRANSFORMAÇÕES DE HELICOIDES

A configuração de referência de um manipulador é definida pelos valores de s e s_o de toda a cadeia para uma dada posição e orientação inicial. Essa configuração, por conveniência, é escolhida de forma que seja de fácil visualização e entendimento, e tipicamente usa-se o manipulador esticado (TSAI, 1999). É necessário transformar a configuração de referência em uma configuração arbitrária em função das rotações e translações de todos os elos, considerando-se que algum movimento será realizado pelo efetuador.

Quando um elo é movimentado pela ação de uma junta, os elos subsequentes da cadeia também sofrerão os deslocamentos. Então, quando todos os elos tiverem os deslocamentos iniciando do primeiro¹ elo até o último², o deslocamento do primeiro elo influencia o deslocamento do segundo elo ao último, enquanto, o segundo elo afeta do terceiro elo ao último e assim sucessivamente.

Tsai (1999) descreve que a ordem com a qual os movimentos são realizados não altera a configuração final dos robôs. Assim, dados os movimentos de todos os elos é possível calcular a configuração do robô assumindo que os elos foram movimentados do primeiro ao último conforme descrito anteriormente. Tsai (1999) propõe outra ordem para aplicação dos movimentos dos elos, partindo do último elo até o primeiro, a técnica é denominada de **deslocamentos dos helicoides sucessivos**, como pode ser observado na Figura 61 (b).

A Figura 61 ilustra a ordem dos deslocamentos. Na Figura 61 (a) os deslocamentos são aplicados do primeiro elo ao último, enquanto na Figura 61 (b) é o método dos helicoides sucessivos. As duas metodologias são semelhantes e apresentam o mesmo resultado.

Para fazer a transformação dos movimentos entre os elos, trabalha-se com a matriz conhecida como matriz de transformação de Rodrigues, denominada pela letra A e definida na equação B.1. Ela representa o deslocamento espacial de um corpo rígido. O deslocamento é obtido a partir de uma rotação θ em torno do eixo de helicóide $s = [s_x \ s_y \ s_z]^T$ e uma translação t na direção do eixo s com relação ao sistema de referência O dado uma distância $s_o = [s_{ox} \ s_{oy} \ s_{oz}]^T$ de s . Assim, a matriz de Rodrigues pode ser escrita:

¹O primeiro elo de um robô é considerado o elo mais próximo a base do robô.

²O último elo é considerado o elo com o efetuador.

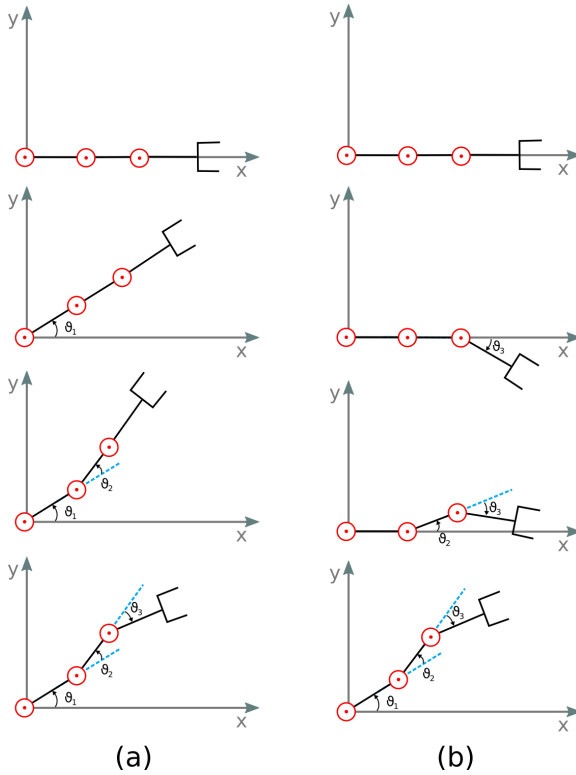


Figura 61: Transformação de helicoides.

$$A_{\vec{s}, \theta} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (\text{B.1})$$

$$\begin{aligned} a_{11} &= (s_x^2 - 1)(1 - \cos \theta) + 1 \\ a_{12} &= s_x s_y (1 - \cos \theta) - s_z \sin \theta \\ a_{13} &= s_x s_z (1 - \cos \theta) + s_y \sin \theta \\ a_{14} &= t s_x - s_{o_x} (a_{11} - 1) - s_{o_y} a_{12} - s_{o_z} a_{13} \\ a_{21} &= s_x s_y (1 - \cos \theta) + s_z \sin \theta \\ a_{22} &= (s_y^2 - 1)(1 - \cos \theta) + 1 \\ a_{23} &= s_y s_z (1 - \cos \theta) - s_x \sin \theta \\ a_{24} &= t s_y - s_{o_x} a_{21} - s_{o_y} (a_{22} - 1) - s_{o_z} a_{23} \end{aligned}$$

$$\begin{aligned}
a_{31} &= s_x s_z (1 - \cos \theta) - s_y \sin \theta \\
a_{32} &= s_y s_z (1 - \cos \theta) - s_x \sin \theta \\
a_{33} &= (s_x^2 - 1)(1 - \cos \theta) + 1 \\
a_{34} &= t s_z - s_{o_x} a_{31} - s_{o_y} a_{32} - s_{o_z} (a_{33} - 1) \\
a_{41} &= 0 \\
a_{42} &= 0 \\
a_{43} &= 0 \\
a_{44} &= 1
\end{aligned}$$

em que a sub-matriz superior esquerda 3×3 da matriz A representa a rotação R de um corpo rígido e a sub-matriz superior direita 3×1 da matriz A representa a translação de um corpo rígido. Quando se usa juntas prismáticas, assume-se que $\theta_i = 0$ e para as juntas rotativas assume-se $t_i = 0$.

Tsai (1999) mostra que a transformação de um ponto p_1 (pertencente a um elo/corpo rígido) para a nova posição p_2 rotacionado por um ângulo θ e/ou uma translação t é obtida utilizando a equação:

$$p_2 = A p_1 \quad (\text{B.2})$$

em que A é a matriz de Rodrigues.

Desta forma, se associa para cada junta a matriz de transformação A_i que corresponde a rotação/translação causada por ela nos elos subsequentes.

Considerando que s_i representa o vetor unitário que determina a direção de translação e rotação do eixo helicoidal referente ao elo i e s_{oi} é o vetor que indica a posição do vetor s_i com relação a um sistema de coordenadas fixo. Supondo que a junta i apresenta uma transformação dada pela matriz A_i , então, o eixo modificado s_{i+1} e o vetor modificado s_{oi+1} são dados pelas equações:

$$\begin{aligned}
s_{i+1} &= R_i s'_{i+1} \\
s_{oi+1} &= A_i s'_{oi+1}
\end{aligned} \quad (\text{B.3})$$

em que, R_i é a sub-matriz superior esquerda 3×3 da matriz A_i referente ao elo i (Equação B.1).

Assim, supondo que todos os elos podem ter deslocamentos:

$$s_{i+1} = \left(\prod_{j=1}^i R_j \right) s'_{i+1}$$

$$s_{oi+1} = \left(\prod_{j=1}^i A_j \right) s'_{oi+1} \quad (\text{B.4})$$

Como o cálculo para o elo $i + 1$ utiliza as matrizes de transformação de 1 até i , é necessário que estas sejam calculadas utilizando os valores modificados de s_j e s_{oj} , onde $j = 1, 2, \dots, i$. Desta forma, os valores das juntas devem ser modificados da junta 1 até a junta n .

Estas composições de transformações compreendem a transformação dos elos do robô com relação a base na origem. Caso a base do robô não coincida com a origem, existe uma transformação A_b que leva a origem ao sistema de coordenadas da base. Assim os vetores resultantes são:

$$\begin{aligned} s_{i+1} &= R_b \left(\prod_{j=1}^i R_j \right) s'_{i+1} \\ s_{oi+1} &= A_b \left(\prod_{j=1}^i A_j \right) s'_{oi+1} \end{aligned} \quad (\text{B.5})$$

Caso o sistema de coordenadas do efetuador não coincida com o último elo do robô, sua posição e orientação com relação a origem do sistema é dada pelas seguintes equações:

$$\begin{aligned} s_e &= R_b \left(\prod_{j=1}^n R_j \right) R_e s'_e \\ s_{oe} &= A_b \left(\prod_{j=1}^n A_j \right) A_e s'_{oe} \end{aligned} \quad (\text{B.6})$$

Como detalhado no anexo A, para as juntas rotativas o deslocamento helicoidal é dado por:

$$\hat{\$}_i = \begin{bmatrix} s_i \\ s_{oi} \times s_i \end{bmatrix}_{(6 \times 1)} \quad (\text{B.7})$$

Para as juntas prismáticas o deslocamento helicoidal é dada pela matriz:

$$\hat{\$}_i = \begin{bmatrix} 0 \\ s_i \end{bmatrix}_{(6 \times 1)} \quad (\text{B.8})$$

B.2 MÉTODO DE DAVIES

A lei das malhas de Kirchhoff, aplicada em circuitos elétricos na área da engenharia elétrica, estabelece que a soma algébrica das diferenças de potencial em um percurso fechado é nula. DAVIES (1981, 2000) adaptou essa lei das malhas para o cálculo da cinemática de mecanismos em cadeias fechadas, que diz que a soma das velocidades relativas de dois elos adjacentes de um manipulador ao longo de qualquer cadeia cinemática fechada é zero. Assim, com esse método, conhecido como método de Davies, é possível estabelecer a relação instantânea entre as velocidades de todas as juntas de uma cadeia cinemática fechada usando a representação das velocidades através de helicoides. Essa relação é denominada equação de restrição da cadeia cinemática do manipulador.

A aplicação do método de Davies é melhor entendida também a partir de um exemplo com um mecanismo com 4 barras e 4 juntas (Figura 62) (CAMPOS, 2004).

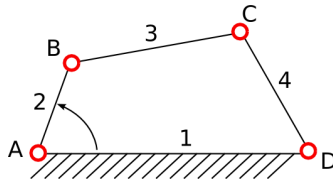


Figura 62: Mecanismo planar com 5 barras (CAMPOS, 2004).

O helicóide $\$A$ representa o movimento do elo 2 com relação ao elo 1, o helicóide $\$B$ representa o movimento do elo 3 com relação ao elo 2, o helicóide $\$C$ representa o movimento do elo 4 com relação ao elo 3 e o helicóide $\$D$ é associado ao movimento do elo 4 com o elo 1. Os helicóides $\$A$, $\$B$, $\$C$ e $\$D$ representam os pares cinemáticos A , B , C e D , respectivamente.

O mecanismo (Figura 62) é planar e devido a isso cada helicóide possui somente três componentes, uma vez que, v_p não tem componente na direção z (ou seja \mathcal{R}^*) e ω não possui as componentes \mathcal{L} e \mathcal{M} , referentes as velocidade angular em torno dos eixos x e y . Desta forma, os helicóides de um mecanismo planar possuem apenas as componentes \mathcal{N} , \mathcal{P}^* e \mathcal{Q}^* , isto é, a velocidade angular em torno do eixo z e a velocidade linear na direção do eixo x e y .

O movimento do elo 2 com relação ao elo 1 é representado pelo helicóide $\$A$, o movimento do elo 3 com relação ao elo 1 é representado

pela soma dos helicoides $\$A + \B . O movimento do elo 1 em relação a ele mesmo é nulo e pode ser expresso pela equação B.9, aplicando o método de Davies para cadeias fechadas.

$$\$A + \$B + \$C + \$D + \$E = 0 \quad (\text{B.9})$$

sendo que o vetor 0 tem dimensão (3×1) . Substituindo os helicoides pelo seu eixo normalizado $\hat{\$}$ e pela magnitude da velocidade \dot{q} , tem-se a equação:

$$\hat{\$}_A \dot{q}_A + \hat{\$}_B \dot{q}_B + \hat{\$}_C \dot{q}_C + \hat{\$}_D \dot{q}_D = 0 \quad (\text{B.10})$$

A equação B.10 é representada na forma matricial, resultando na relação:

$$\begin{bmatrix} \hat{\$}_A & \hat{\$}_B & \hat{\$}_C & \hat{\$}_D \end{bmatrix}_{(3 \times 4)} \begin{bmatrix} \dot{q}_A \\ \dot{q}_B \\ \dot{q}_C \\ \dot{q}_D \end{bmatrix}_{(4 \times 1)} = 0_{(3 \times 1)} \quad (\text{B.11})$$

De forma geral a equação de restrição é dada por:

$$N\dot{q} = \vec{0} \quad (\text{B.12})$$

em que N é a matriz de rede que contém os helicoides normalizados e \dot{q} é o vetor de magnitudes da velocidade.

A equação B.12 precisa ser rearranjada a fim de resolver a cinemática diferencial no espaço de juntas da cadeia cinemática. Em uma cadeia cinemática existem as juntas que são controladas por atuadores³, que são consideradas juntas ativas (primárias) e as juntas sem atuadores, chamadas juntas passivas (secundárias). Assim, para encontrar as magnitudes das velocidades das juntas passivas em função das juntas atuadas, as matrizes da equação B.12 podem ser divididas em duas partes: nas juntas com magnitudes de velocidades conhecidas (juntas primárias) e nas juntas com magnitudes de velocidades desconhecidas (juntas secundárias), ou seja, $\dot{q} = [\dot{q}_s \dot{q}_p]$. Assim, rearranjando-se a matriz N com relação a separação das magnitudes, tem-se $N = [N_s \dot{N}_p]$, resultando na equação:

³Atuador: elemento que produz movimento através de uma ação mecânica, induzida por cilindros pneumáticos, hidráulicos ou por motores.

$$\begin{bmatrix} N_s \vdots N_p \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \cdots \\ \dot{q}_p \end{bmatrix} = 0 \quad (\text{B.13})$$

Assim, separando as matrizes e isolando as matrizes referentes as juntas secundárias, tem-se a relação:

$$N_s \dot{q}_s = -N_p \dot{q}_p \quad (\text{B.14})$$

Reorganizando agora a equação B.11, que representa o exemplo do mecanismo planar (Figura 62) é possível resolver a cinemática diferencial no espaço das juntas. As magnitudes das juntas secundárias \dot{q}_B , \dot{q}_C e \dot{q}_D são em função das magnitudes das junta primária \dot{q}_A (CAMPOS, 2004).

$$\begin{bmatrix} \hat{\$}_B & \hat{\$}_C & \hat{\$}_D \end{bmatrix}_{3 \times 3} \begin{bmatrix} \dot{q}_B \\ \dot{q}_C \\ \dot{q}_D \end{bmatrix}_{3 \times 1} = - \begin{bmatrix} \hat{\$}_A \end{bmatrix}_{3 \times 1} \begin{bmatrix} \dot{q}_A \end{bmatrix}_{1 \times 1} \quad (\text{B.15})$$

Se a matriz N_s admitir inversa, as magnitudes das velocidades das juntas secundárias \dot{q}_s podem ser calculadas através da seguinte relação:

$$\dot{q}_s = -N_s^{-1} N_p \dot{q}_p \quad (\text{B.16})$$

Caso a matriz N_s não for invertível ainda é possível tornar a equação B.14 válida, pois pode existir uma combinação de valores para N_s , \dot{q}_s , N_p e \dot{q}_p que mantenha a igualdade válida. Porém, outras técnicas computacionais precisam ser aplicadas para encontrar os valores de \dot{q}_s , dadas as outras matrizes da equação.

A equação B.16 relaciona as velocidades das juntas atuadas e passivas. Quando as cadeias cinemáticas possuem múltiplas malhas a determinação da equação de restrição é difícil de ser obtida, mas pode ser facilitada utilizando a teoria de grafos.

B.3 CADEIAS CINEMÁTICAS VIRTUAIS

As cadeias cinemáticas virtuais foram proposta por Campos (2004) a fim de tornar possível a aplicação do método de Davies para manipuladores seriais, já que o método é somente aplicável em robôs com cadeias cinemáticas fechadas. Assim, a ideia de Campos (2004)

consiste no fechamento de cadeias cinemáticas, que representam manipuladores seriais, através da adição de cadeias cinemáticas virtuais. Desta forma, obtém-se informações do movimento relativo entre os elos do robô ou o movimento absoluto de um elo em particular, como do efetuador de um manipulador. As cadeias cinemáticas virtuais também são úteis para impor restrições cinemáticas particulares entre dois elos de uma cadeia cinemática, tornando-se, assim, um elemento de controle nas cadeias cinemáticas. O conjunto de cadeia cinemática do robô e cadeia cinemática virtual é denominado cadeia cinemática modificada.

A aplicação do método de Davies torna possível obter a equação de restrição da cadeia modificada e assim determinar a relação das velocidades das juntas em função de um outro conjunto de velocidades de juntas dadas. E, finalmente a partir do método estabelecer a cinemática direta e inversa de robôs paralelos e seriais, usando um mesmo procedimento.

Uma cadeia cinemática composta por elos e juntas virtuais é chamada de cadeia virtual de Assur e possui algumas propriedades (CAMPOS, 2004):

- a cadeia virtual é uma cadeia cinemática serial composta por elos e juntas virtuais;
- os movimentos das juntas virtuais são representados por helicoides linearmente independentes;
- a cadeia virtual não altera o grau de liberdade da cadeia cinemática real.

Assim sendo, ao utilizar uma cadeia virtual, é necessário definir uma cadeia serial, com grau de liberdade igual à ordem do sistema de helicoides ao qual pertence a cadeia cinemática real e cujo intuito é representar a movimentação desejada na cadeia real.

Como as cadeias cinemáticas são modificadas adicionando cadeias cinemáticas virtuais sem alterar a mobilidade da cadeia original, é possível obter informações e introduzir características a seu movimento, permitindo a aplicação dos algoritmos de cadeias fechadas em cadeias abertas.

Seja λ a ordem do sistema de helicoides que classifica uma cadeia cinemática virtual. Assim, se a cadeia cinemática real é planar, então o sistema de helicoides considerado é $\lambda = 3$ e se a cadeia cinemática real é espacial então o sistema de helicoides a ser considerado é $\lambda = 6$. A Figura 63 ilustra uma árvore hierárquica de classificação das cadeias virtuais planares e espacial.

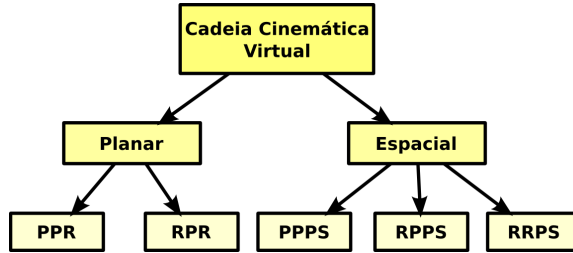


Figura 63: Classificação das cadeias virtuais.

As cadeias virtuais planas podem ser classificadas como *PPR* com duas juntas prismáticas e uma rotativa e *RPR* constituída de uma junta prismática e duas rotativas. As cadeias virtuais planas têm a mesma ordem que o sistema de helicoides $\lambda = 3$.

A **cadeia cinemática virtual PPR**, ilustrada na Figura 64, é composta de duas juntas prismáticas px e py nas direções ortogonais x e y e por uma junta rotativa rz de movimento em torno do eixo z . Os seus movimentos helicoidais são representados por $\$_{px}$, $\$_{py}$ e $\$_{rz}$. Um sistema de coordenadas C é escolhido na origem da junta rotativa virtual rz e os valores de s e s_o estão descritos pelas matrizes na equação B.17.

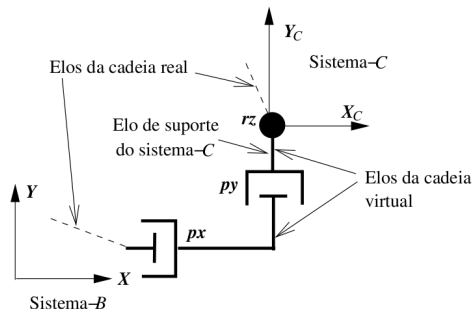


Figura 64: Cadeia virtual planar *PPR* (CAMPOS, 2004).

$$C_{S_{px}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; C_{S_{py}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; C_{S_{rz}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; C_{S_{o_{rz}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.17})$$

Usando as Equações A.7 e A.10 do Apêndice A, os movimentos helicoidais das juntas da cadeia virtual representadas no sistema de coordenadas C são dadas pelas relações representadas em:

$${}^C\hat{\$}_{px} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; {}^C\hat{\$}_{py} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}; {}^C\hat{\$}_{rz} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.18})$$

A cadeia virtual PPR representa um sistema cartesiano no plano, pode-se eliminar os termos \mathcal{L} , \mathcal{M} e \mathcal{R} (referentes ao deslocamento espacial, um com relação ao movimento prismático e dois rotativos) e assim os movimentos helicoidais das juntas são reescritos em:

$${}^C\hat{\$}_{px} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; {}^C\hat{\$}_{py} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; {}^C\hat{\$}_{rz} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.19})$$

Já, a **cadeia cinemática virtual RPR** (Figura 65) é formada por duas juntas rotativas $rz1$ e $rz2$ em torno do eixo z e por uma junta prismática pr com deslocamento na direção dada pelos elos virtuais da cadeia virtual (representado pelo ângulo α na Figura). Os seus movimentos helicoidais são representados por $\$_{rz1}$, $\$_{rz2}$ e $\$_{pr}$. Um sistema de coordenadas P é fixo a junta virtual $rz2$, os valores de s e s_o estão descritos nas matrizes em B.20.

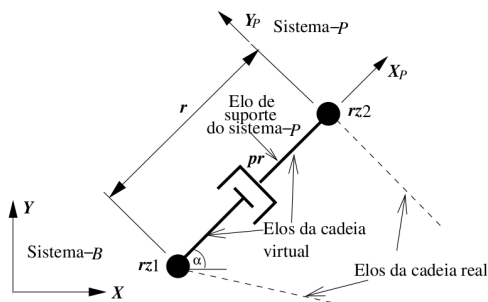


Figura 65: Cadeia virtual planar RPR (CAMPOS, 2004).

$${}^P S_{rz1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; {}^P S_{pr} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; {}^P S_{rz2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; {}^P S_{orz1} = \begin{bmatrix} -r \\ 0 \\ 0 \end{bmatrix}; {}^P S_{orz1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.20})$$

De acordo com as Equações A.7 e A.10 (Apêndice A) os movimentos helicoidais das juntas da cadeia virtual representadas no sistema de coordenadas P são dadas pelas matrizes:

$${}^P \hat{\$}_{rz1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ r \\ 0 \end{bmatrix}; {}^P \hat{\$}_{pr} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; {}^P \hat{\$}_{rz2} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.21})$$

A cadeia virtual RPR representa um sistema polar no plano, desta forma, pode-se eliminar os termos \mathcal{L} , \mathcal{M} e \mathcal{R} e assim os movimentos helicoidais das juntas são reescritos nas relações:

$${}^P \hat{\$}_{rz1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ r \end{bmatrix}; {}^P \hat{\$}_{pr} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; {}^P \hat{\$}_{rz2} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.22})$$

As cadeias virtuais espaciais podem ser classificadas como $PPPS$ com três juntas prismáticas e uma junta esférica, $RPPS$ com uma junta rotativa, duas juntas prismáticas e uma junta esférica e $RRPS$ com duas juntas rotativas, uma junta prismática e uma junta esférica. A escolha de qual conjunto de juntas virtuais é dependente das características desejadas de movimentação, determinadas pelo espaço no qual a trajetória será definida. Em cadeias virtuais cinemáticas espaciais a ordem do sistema de helicoides $\lambda = 6$, e, assim, a cadeia virtual deve ter seis graus de liberdade. As juntas esféricas podem ser substituídas instantaneamente por três juntas rotativas ortogonais.

A **cadeia cinemática virtual $PPPS$** (Figura 66), também conhecida como $3P3R$, consiste de três juntas prismáticas px , py e pz com deslocamento nas direções x , y e z e seus movimentos são representados por $\$_{px}$, $\$_{py}$ e $\$_{pz}$, respectivamente e por três juntas rotativas ortogonais rx , rz e ry com movimento em torno dos eixos x , y e z , e seus movimentos helicoidais são representados por $\$_{rx}$, $\$_{ry}$ e $\$_{rz}$, respectivamente. A cadeia virtual $3P3R$ está associada a um sistema de coordenadas cartesianas no espaço 3D e seus movimentos helicoidais com relação a um

sistema de coordenadas C fixo a junta esférica são dados na relação B.23.

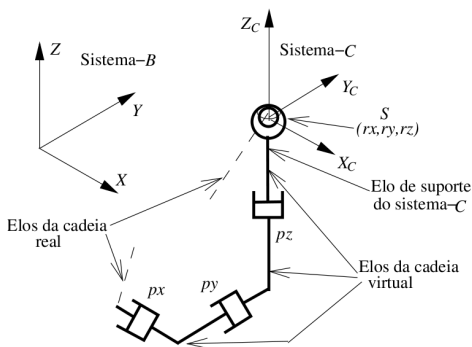


Figura 66: Cadeia virtual espacial PPS (CAMPOS, 2004).

$$C_{\hat{\$}_{rx}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; C_{\hat{\$}_{ry}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; C_{\hat{\$}_{rz}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; C_{\hat{\$}_{px}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; C_{\hat{\$}_{py}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}; C_{\hat{\$}_{pz}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad (\text{B.23})$$

A **cadeia cinemática virtual RPPS** (Figura 67) é composta por uma junta rotativa rz na direção do eixo z , duas juntas prismáticas pr e pz uma na direção do eixo z e a outra na direção ortogonal ao eixo z (denominada radial) e por uma junta esférica (S). Quando a junta esférica está sobre o eixo da junta rz , a cadeia $RPPS$ perde um grau de liberdade, devido ao alinhamentos de dois eixos, tornando o sistema linearmente dependente e neste caso a $RPPS$ não cumpre as condições de cadeia virtual (CAMPOS, 2004).

O movimento das três primeiras juntas rz , pz e pr são representadas pelos helicoides $\$_{rz}$, $\$_{pz}$ e $\$_{pr}$, respectivamente e o movimento é realizado dentro de um cilindro. Os movimentos realizados pelas juntas rotativas ortogonais referente a junta esférica tem seu movimento nas direções normal ao cilindro (rn), tangencial ao cilindro (rt) e binormal ao cilindro (rb), e seus helicoides são $\$_{rn}$, $\$_{rt}$ e $\$_{rb}$, respectivamente (CAMPOS, 2004).

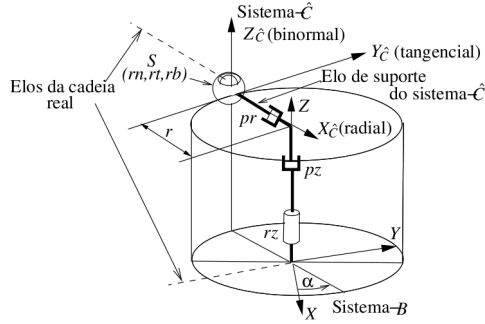


Figura 67: Cadeia virtual espacial *RPPS* (CAMPOS, 2004).

Considerar um sistema de coordenadas \hat{C} com origem na junta esférica (Figura 67), com eixo $X_{\hat{C}}$ na direção radial do cilindro, o eixo $Y_{\hat{C}}$ na direção tangencial ao cilindro e o eixo $Z_{\hat{C}}$ na direção binormal ao cilindro. Uma distância radial instantânea r é considerada entre sistema de coordenadas \hat{C} e o eixo da junta rz , tem-se que os valores de s e s_o das juntas virtuais são dados na Equação B.24.

$$\hat{C}_{s_{rn}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{s_{rt}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \hat{C}_{s_{rb}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \hat{C}_{s_{pr}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{s_{pz}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \hat{C}_{s_{rz}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix};$$

$$\hat{C}_{s_{o_{rn}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{s_{o_{rt}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{s_{o_{rb}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{s_{o_{rz}}} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}; \quad (\text{B.24})$$

A cadeia virtual *RPPS* representa um sistema de coordenadas cilíndricas e o movimento helicoidal das juntas virtuais, representados no sistema de coordenadas \hat{C} fixo a junta esférica, são dadas por pelas igualdades:

$$\hat{C}_{\$rn} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{\$rt} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{\$rb} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{\$pr} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{\$pz} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \hat{C}_{\$rz} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -r \\ 0 \end{bmatrix}; \quad (\text{B.25})$$

A **cadeia cinemática virtual RRPS** (Figura 68) é composta por duas juntas rotativas rz e ro uma na direção do eixo z e a outra na direção ortogonal ao eixo z e definida pelo ângulo α , uma junta prismática pr na direção radial estabelecida através do ângulo β e por uma junta esférica (S). Quando a junta esférica está sobre o eixo da junta rz , a cadeia $RPPS$ perde um grau de liberdade, devido ao alinhamentos de dois eixos, tornando o sistema linearmente dependente e neste caso a $RRPS$ não cumpre as condições de cadeia virtual (CAMPOS, 2004).

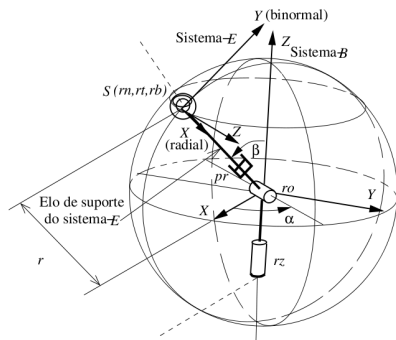


Figura 68: Cadeia virtual espacial $RRPS$ (CAMPOS, 2004).

O movimento das três primeiras juntas rz , ro e pr são representadas pelos helicoides $\$rz$, $\$ro$ e $\$pr$, respectivamente e o movimento é realizado dentro de uma esfera. Os movimentos realizados pelas juntas rotativas ortogonais referente a junta esférica tem seu movimento nas direções normal à esfera (rn), tangencial à esfera (rt) e binormal à esfera (rb), e seus helicoides são $\$rn$, $\$rt$ e $\$rb$, respectivamente (CAMPOS, 2004).

Considerar um sistema de coordenadas E com origem na junta esférica (Figura 68), com eixo X_E na direção radial do cilindro, o eixo

Y_E na direção tangencial ao cilindro e o eixo Z_E na direção binormal ao cilindro. Uma distância radial instantânea r é considerada entre sistema de coordenadas E e o sistema de coordenadas B , β é o ângulo instantâneo entre o eixo da junta ro e a junta prismática pr e os eixos dos helicoides das juntas na direção positiva dos eixos do sistema de coordenadas E , tem-se que os valores de s e s_o das juntas virtuais são dados na Equação B.26.

$$\begin{aligned}
 E_{S_{rn}} &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; E_{S_{rt}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; E_{S_{rb}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; E_{S_{pr}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; E_{S_{ro}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; E_{S_{rz}} = \begin{bmatrix} -\cos\beta \\ \sin\beta \\ 0 \end{bmatrix}; \\
 E_{S_{o_{rn}}} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; E_{S_{o_{rt}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; E_{S_{o_{rb}}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; E_{S_{o_{ro}}} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}; E_{S_{o_{rz}}} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}; \quad (B.26)
 \end{aligned}$$

A cadeia virtual $RRPS$ representa um sistema de coordenadas esférico e o movimento helicoidal das juntas virtuais, representados no sistema de coordenadas E fixo a junta esférica, são dadas pelas igualdades:

$$\begin{aligned}
 E_{\hat{S}_{rn}} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; E_{\hat{S}_{rt}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; E_{\hat{S}_{rb}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; E_{\hat{S}_{pr}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; E_{\hat{S}_{ro}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ r \\ 0 \\ 0 \end{bmatrix}; E_{\hat{S}_{rz}} = \begin{bmatrix} -\cos\beta \\ \sin\beta \\ 0 \\ 0 \\ 0 \\ r\sin\beta \end{bmatrix}; \\
 & \quad (B.27)
 \end{aligned}$$

A seguir é apresentado um exemplo de um manipulador e seu grafo (ilustrado na Figura 69). A matriz de incidência desse sistema está descrita na igualdade B.28.

$$\begin{aligned}
 & \quad R_1 \quad R_2 \quad R_Z \quad P_Y \quad P_X \\
 B = \text{malha} & \quad \begin{bmatrix} 1 & 1 & -1 & -1 & -1 \end{bmatrix} \quad (B.28)
 \end{aligned}$$

No exemplo da Figura 69 (b), o grafo é formado apenas por uma malha, assim a matriz tem apenas uma linha e as colunas contém as arestas (no caso, cinco arestas, três referentes as juntas rotativas as outras referentes as juntas prismáticas). A matriz de rede N da equação de restrição é obtida multiplicando a matriz de helicoides normalizados

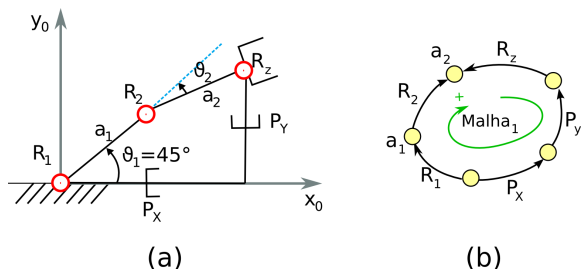


Figura 69: (a) Manipulador planar de dois elos com três juntas rotativas (R_1 , R_2 e R_z) e duas juntas prismáticas (P_x e P_y) (b) Grafo representativo do manipulador.

das juntas com a matriz de incidência B (DAVIES, 1981):

$$N = B \cdot \hat{\$} \quad (\text{B.29})$$

Para ilustrar uma aplicação de cadeias virtuais em um manipulador, na Figura 69 as três juntas P_x , P_y e R_z podem ser consideradas uma cadeia virtual e as duas juntas rotativas fazem parte da cadeia real que representa um robô. A cadeia virtual PPR foi adicionada para fechar a cadeia real e poder aplicar o método de Davies para o cálculo das velocidades das juntas do manipulador.

Campos (2004) comenta que o procedimento de escolha da cadeia virtual independe da sua estrutura e portanto, pode ser escolhida de acordo com a conveniência. Ele destaca ainda que é possível representar os movimentos das juntas virtuais em um sistema de coordenadas conveniente para obter os deslocamentos helicoidais mais simples. Assim, a escolha adequada de uma cadeia virtual facilita no planejamento de trajetória dos robôs.

As cadeias virtuais planares PPR e RPR estão relacionadas ao sistema de coordenadas cartesianas e polares, respectivamente. As cadeias virtuais espaciais 3P3R, RPPS e RRPS estão relacionadas aos sistemas de coordenadas cartesianas, cilíndricas e esféricas, respectivamente.

Assim, se for desejável fornecer as velocidades no sistema de coordenadas cartesianas, se escolhe a cadeia virtual 3P3R. Se for mais adequado fornecer no sistema coordenadas cilíndricas, se escolhe a cadeia virtual RRPS. Por fim, se for desejável fornecer as velocidades no sistema de coordenadas esféricas, usa-se a cadeia virtual RRPS.

APÊNDICE C – Teoria de Grafos

A teoria de grafos auxilia na representação de forma abstrata as cadeias cinemáticas. Um grafo que representa uma cadeia cinemática permite analisar o acoplamento existente entre os manipuladores e assim, determinar de modo sistemático as equações de restrições de cadeias cinemáticas complexas, conhecendo, por exemplo, a ordem de percorrer as arestas (elos) e a incidência dos elos nas malhas do grafo. Para desenvolver a formalização proposta serão utilizados alguns conceitos básicos da teoria de grafos e os principais termos que são fundamentais para o embasamento matemático que é proposto neste trabalho. O conteúdo a seguir foi baseado nos trabalhos de Diestel (2005), Bondy e Murty (2008), Tsai (2001), Seshu e Reed (1961), Mayeda (1972), Chen (1971).

Definição 3. *Um grafo simples G consiste de um conjunto finito e não vazio $V(G)$ de elementos chamados de vértices e de um conjunto finito $A(G)$ de pares não ordenados de elementos distintos de $V(G)$, chamados arestas.*

Em um grafo, um **caminho** é uma sequência de vértices $v_1, v_2, v_3, \dots, v_n$ do conjunto $V(G)$ conectados por arestas $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$. As arestas fazem parte do caminho.

Um **circuito** ou **malha** é um caminho fechado, ou seja, em que o vértice final é igual ao vértice inicial ($v_1 = v_n$, tal que v_1 e $v_n \in V(G)$).

Dado um grafo, diz-se que o vértice v é adjacente ao vértice w se existe uma aresta $e = (v, w)$ no grafo, com v e $w \in V(G)$.

Dois vértices são considerados conexos, se existe um caminho de um vértice a outro. Note que, dois vértices conexos não são necessariamente adjacentes. Um grafo é dito **conexo** se cada vértice em G é unido a outro vértice por pelo menos um caminho.

Um **subgrafo** (SG) de G é um grafo cujos vértices e arestas estão contidos no grafo G . Em outras palavras, um subgrafo de G é um grafo obtido removendo um número de arestas e/ou vértices de G . A remoção de um vértice a partir de G implica na remoção de todas as arestas incidentes naquele vértice, já a remoção de uma aresta não necessariamente implica na remoção de um vértice, embora isso possa resultar em um ou dois vértices isolados.

Um grafo pode ser **direcionado** ou **não-direcionado**. Este fato está associado ao conjunto de arestas deste grafo serem direcionadas e não-direcionadas. Uma aresta de um grafo direcionado é um par ordenado de vértices (u, v) , com u e $v \in V(G)$, tal que u é o vértice de partida e v é o vértice de chegada. A aresta direcionada (u, v) é diferente da aresta (v, u) . Em um grafo não-direcionado (Figura 70

(a)), uma aresta sempre conecta dois vértices em ambas as direções, não interessa a ordem do vértice em uma aresta, ou seja, (u, v) e (v, u) são a mesma aresta. As arestas são representadas por linhas simples. Quando duas ou mais arestas conectam os mesmos dois vértices, elas são chamadas de arestas paralelas. No grafo direcionado (Figura 70 (b)) a ordem entre os vértices de uma aresta é importante e as arestas são representadas por setas.

A Figura 70 (b) mostra um grafo direcionado que possui três circuitos, um dos circuitos seria composto pelas arestas $C_1 = \{A, B, C, D\}$, o outro pelas arestas $C_2 = \{D, E, F, G\}$, e o terceiro pelas arestas $C_3 = \{A, B, C, E, F, G\}$. Além disso, cada circuito possui um sentido de percurso como é ilustrado na Figura 70 (b).

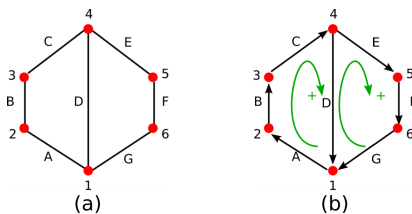


Figura 70: (a) Grafo não-direcionado; (b) Grafo direcionado.

Uma **árvore** T é um grafo conexo que não contém circuitos. As propriedades de uma árvore T com v vértices são (TSAI, 2001):

1. quaisquer dois vértices da árvore T são conectados por um e somente um caminho;
2. a árvore T contém $v-1$ arestas;
3. quando conectado quaisquer dois vértices não adjacentes de uma árvore T com uma aresta, tem-se um grafo com um e somente um circuito.

Para uma árvore T do grafo G , uma aresta que não está em T é chamada de **corda**. Um conjunto de todas as arestas do grafo G que não estão na árvore T é chamado de conjunto de cordas com relação a árvore T .

Uma vez que existe um único caminho conectado entre qualquer dois vértices em uma árvore T , a adição de uma corda em uma árvore T gera um único circuito no grafo resultante. Cada corda define um

circuito (com relação a árvore escolhida) de forma única em um grafo direcionado.

Uma **árvore de espalhamento** T' é uma árvore que contém todos os vértices de um grafo conexo G , logo, T' é um subgrafo de G , com v vértices e $v - 1$ arestas.

Esses são os conceitos básicos da teoria de grafos que auxiliará na estruturação da representação abstrata de SRC. O foco com relação a teoria de grafos deste trabalho são grafos direcionados conexos e será composto por caminhos fechados (circuitos).

C.1 EQUAÇÃO DE EULER E SUA ASSOCIAÇÃO COM A MATRIZ DE CIRCUITOS

C.1.1 Equação de Euler e circuitos independentes

Um grafo simples planar conexo divide o plano em regiões fechadas (os limites dessas regiões fechadas são os circuitos/malhas) e a região externa. A equação que associa o número de vértices (v), o número de arestas (e) e o número de regiões (r) de um grafo dada pela equação (TSAI, 2001):

$$v - e + r = 2 \quad (\text{C.1})$$

O número total de regiões de um grafo G é dado pelo somatório das regiões internas independentes, tratadas como o número de circuitos independentes L de um grafo planar, mais a região externa. Assim, tem-se:

$$r = L + 1 \quad (\text{C.2})$$

Substituindo a Equação C.2 na Equação C.1, tem-se que o número máximo de circuitos independentes de um grafo é dado por:

$$\begin{aligned} v - e + L + 1 &= 2 \\ L &= e - v + 1 \quad (\text{Equação de Euler}) \end{aligned} \quad (\text{C.3})$$

Um conjunto de cordas de um grafo está associado ao número de circuitos independentes do grafo, assim, em um grafo G , o número de cordas consiste de um conjunto de $(e - v + 1)$ arestas.

Definição 4. Considere $(e_1, e_2, \dots, e_{(e-v+1)})$ sendo um conjunto de cordas com relação a uma árvore de espalhamento T' de um grafo co-

nexo. Também considere C_i , como sendo um circuito em $T' \cup (e_i)$, para $i = 1, 2, \dots, (e - v + 1)$. Então, a coleção de circuitos $C_1, C_2, \dots, C_{(e-v+1)}$ é chamado de conjunto de circuitos independentes da árvore T' .

Os circuitos independentes de um grafo direcionado conexo com respeito a uma árvore de espalhamento T' são os $(e - v + 1)$ circuitos formados por cada uma das cordas e o único caminho na árvore entre os dois vértices da corda. Baseado nessa definição, é possível perceber que cada corda, de um conjunto de cordas, completa exatamente um circuito de um conjunto de circuitos independentes com relação a árvore T' (MAYEDA, 1972).

Um conjunto de circuitos é independente quando nenhum de seus circuitos pode ser escrito como combinação linear dos outros. Assim, os circuitos independentes são considerados linearmente independentes. Qualquer circuito arbitrário do grafo pode ser expresso como uma combinação linear dos circuitos independentes.

Um conjunto de circuitos independentes C_i constitui uma base para o espaço de circuitos e pode ser identificado, primeiramente, encontrando qualquer árvore de espalhamento de um grafo. Existem várias abordagens para construir uma árvores de espalhamentos, por exemplo, utilizando um algoritmo de busca em profundidade ¹ (CORMEN et al., 2001).

Para exemplificar, considere o grafo G da Figura 71, o conjunto de arestas (a, c, e, g) é uma árvore de espalhamento T' . As arestas (b, d, f) é um conjunto de cordas com respeito a T' . Note que o número de arestas e o número de vértices no grafo é $e = 7$ e $v = 5$, respectivamente, e o conjunto de cordas consiste de $e - v + 1 = 7 - 5 + 1 = 3$ elementos. Se a corda d for adicionada a T' , o grafo resultante $(d) \cup T'$ conterá exatamente um circuito.

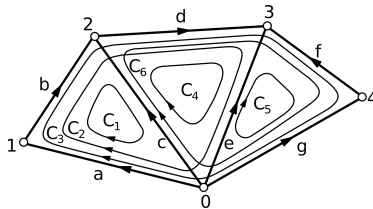


Figura 71: Grafo direcionado e seus circuitos (MAYEDA, 1972).

¹termo em inglês: *depth-first-search*.

C.1.2 Matriz de circuitos

Uma cadeia cinemática pode ser representada de forma unívoca por um grafo, cujos vértices correspondem aos elos da cadeia cinemática e as arestas correspondem as juntas da cadeia cinemática. O grafo auxilia a determinar a equação de restrição de um SRC, pois ele permite analisar a ligação dos vértices através das arestas direcionadas.

O grafo de uma cadeia cinemática pode ser representado por meio de uma **matriz de circuitos** B as quais indicam a presença de arestas em cada malha do grafo. As matrizes de circuitos também auxiliam na aplicação do método de Davies em cadeias cinemáticas com múltiplas malhas. O conteúdo mais completo para a representação da cadeia cinemática por grafos é encontrado em Campos (2004), Simoni (2008), Carboni (2008), incluindo os grafos de acoplamento e grafos de movimento.

A matriz de circuitos B pode ser construída a partir de um grafo usando um critério de atribuição de valores $(0, 1, -1)$ para preenchimento da matriz.

Definição 5. *As linhas de uma matriz de circuitos B de um grafo G representam todos os possíveis circuitos e as colunas representam as arestas.*

O elemento b_{ij} de uma matriz de circuitos $B = [b_{ij}]$ de um grafo G é definido como:

- 1, se o circuito i inclui a aresta j e a orientação da aresta coincide com a orientação do circuito;
- 1, se o circuito i inclui a aresta j e a orientação da aresta é de sentido contrário a orientação do circuito;
- 0 (zero), se o circuito i não inclui a aresta j .

A matriz de circuitos B da Figura 71 está representada na matriz:

$$B = \begin{matrix} & a & b & c & d & e & f & g \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{matrix} & \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & -1 & -1 \end{bmatrix} \end{matrix} \quad (C.4)$$

Seja uma matriz de circuitos B de um grafo direcionado conexo. Ela é rearranjada de forma que as $(e - v + 1)$ primeiras colunas sejam correspondem as cordas. As $(e - v + 1)$ linhas representam os circuitos do conjunto de circuitos independentes com relação a árvore de espalhamento T' . Então, tem-se a Equação C.5 rearrajanda, ela representa uma matriz de circuitos generalizada.

$$B = \begin{matrix} & e_1 & e_2 & \dots & e_{e-v+1} & e'_1 & \dots & e'_{v-1} \\ C_1 & \left[\begin{array}{ccccccc} 1 & 0 & \dots & 0 & & \dots & \\ 0 & 1 & \dots & 0 & & \dots & \\ \vdots & \vdots & \dots & \vdots & & \dots & \\ 0 & 0 & \dots & 1 & & \dots & \\ \vdots & \vdots & \dots & \vdots & & \dots & \\ \vdots & \vdots & \dots & \vdots & & \dots & \end{array} \right] & \end{matrix} \quad (C.5)$$

Assim, na matriz B , a submatriz superior quadrada esquerda é a matriz identidade, isto se deve ao fato de que cada corda está exatamente em um dos circuitos do conjunto de circuitos independentes com respeito a T' . A submatriz obtida considera as $e - v + 1$ linhas da matriz B e é chamada de matriz de circuitos independentes B_f do grafo G .

Uma matriz de circuitos independentes B_f do grafo G , quando arranjada na ordem de cordas e arestas da árvore, é uma matriz de circuitos da seguinte forma:

$$B_f = [U \quad B_t] \quad (C.6)$$

A matriz B_f possui $(e - v + 1)$ linhas, U é a matriz identidade correspondente as cordas do grafo e as colunas da matriz B_t correspondem as arestas da árvore T do grafo G .

Note que de um grafo G qualquer podem ser obtidas diferentes árvores e para cada árvore existe uma matriz de circuitos diferente, podendo resultar em uma matriz B distinta. Uma vez que o posto de uma matriz de circuitos B é $e - v + 1$, uma submatriz de B cujo seu posto seja $(e - v + 1)$ é suficiente para especificar todos os circuitos em grafo.

Para uma matriz de circuitos independentes, existe um conjunto de geradores E , ou seja, uma base em B_f , que produz todos os membros de um conjunto de circuitos C , em que, C é uma coleção de todos os possíveis circuitos.

As linhas da matriz de circuitos B representam todos os mem-

bros de C , os circuitos equivalentes às linhas na matriz de circuitos B_f formam um conjunto de geradores E que também produzem os membros de C .

Considerando o grafo da Figura 71, a matriz de circuitos desse grafo C.4 foi rearranjada sendo (MAYEDA, 1972):

$$B = \begin{matrix} & a & e & d & c & b & f & g \\ \begin{matrix} C_1 \\ C_5 \\ C_6 \\ C_2 \\ C_3 \\ C_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & -1 & -1 \\ 1 & -1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & -1 & -1 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (C.7)$$

Podendo, neste caso, identificar a matriz B_f com as cordas a, e, d e os circuitos independentes C_1, C_5 e C_6 :

$$B_f = \begin{matrix} & a & e & d & c & b & f & g \\ \begin{matrix} C_1 \\ C_5 \\ C_6 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & -1 & -1 \end{bmatrix} \end{matrix} \quad (C.8)$$

Todos os circuitos possíveis do grafo podem ser construídos a partir dos circuitos independentes C_1, C_5 e C_6 . Verifica-se, por exemplo, que o circuito C_3 é a combinação linear $C_1 + C_6$.

APÊNDICE D – Pseudo-inversa adaptada para SRC

Neste apêndice é apresentada a pseudo-inversa adaptada para ser aplicada em um SRC, considerando a utilização do método de Davies para determinar a cinemática de SRC.

A matriz N_s tem o número de linhas l da matriz dado pelo produto entre a ordem do sistema de helicoides λ e o número de robôs m que compõe o sistema, ou seja:

$$l = \lambda \cdot m \quad (\text{D.1})$$

em que $\lambda = 6$ para o espaço 3D e $\lambda = 3$ para os casos planares.

O número de colunas c da matriz N_s é dado pela soma do número total de juntas de todos os robôs componentes do sistema, aqui representada pela equação:

$$c = \sum_{i=1}^m n_i \quad (\text{D.2})$$

com n_i sendo o número de juntas do robô i .

Para os casos em que $c \neq l$, a matriz N_s não é quadrada, e caso $c > l$ o sistema dado na equação B.14 (descrita no apêndice B.2) possui infinitas soluções. Desta forma, considerando a equação B.14, aqui repetida por conveniência:

$$\begin{aligned} N_s \dot{q}_s &= -N_p \dot{q}_p \\ [N_s]_{l \times c} [\dot{q}_s]_{c \times 1} &= -[N_p]_{l \times a} [\dot{q}_p]_{a \times 1} \end{aligned} \quad (\text{D.3})$$

em que $a = l + \lambda$.

Retrabalhando a equação D.3:

$$\begin{aligned} N_s \dot{q}_s &= -N_p \dot{q}_p \quad \times (N_s N_s^T)^{-1} \\ (N_s N_s^T)^{-1} N_s \dot{q}_s &= -(N_s N_s^T)^{-1} N_p \dot{q}_p \quad \times (N_s^T) \\ N_s^T (N_s N_s^T)^{-1} N_s \dot{q}_s &= -N_s^T (N_s N_s^T)^{-1} N_p \dot{q}_p \end{aligned} \quad (\text{D.4})$$

Sabe-se que:

$$N_s = I N_s = (N_s N_s^T) (N_s N_s^T)^{-1} N_s = N_s (N_s^T (N_s N_s^T)^{-1} N_s) \quad (\text{D.5})$$

em que I é a matriz identidade.

Logo:

$$(N_s^T (N_s N_s^T)^{-1} N_s) = I \quad (\text{D.6})$$

Então:

$$\begin{aligned}
 (N_s^T(N_s N_s^T)^{-1}N_s)\dot{q}_s &= -N_s^T(N_s N_s^T)^{-1}N_p\dot{q}_p \\
 I\dot{q}_s &= -N_s^T(N_s N_s^T)^{-1}N_p\dot{q}_p \\
 \dot{q}_s &= -N_s^T(N_s N_s^T)^{-1}N_p\dot{q}_p
 \end{aligned} \tag{D.7}$$

Analisando os índices das matrizes que compõem a equação D.7:

$$\begin{aligned}
 [\dot{q}_s]_{c \times 1} &= -([N_s]_{l \times c})^T ([N_s]_{l \times c}([N_s]_{l \times c})^T)^{-1} [N_p]_{l \times a} [\dot{q}_p]_{a \times 1} \\
 [\dot{q}_s]_{c \times 1} &= -[N_s^T]_{c \times l} ([N_s]_{l \times c} [N_s^T]_{c \times l})^{-1} [N_p]_{l \times a} [\dot{q}_p]_{a \times 1} \\
 [\dot{q}_s]_{c \times 1} &= -[N_s^T]_{c \times l} ([N_s]_{l \times c} [N_s^T]_{c \times l})^{-1} ([N_p]_{l \times a} [\dot{q}_p]_{a \times 1})_{l \times 1} \\
 [\dot{q}_s]_{c \times 1} &= -([N_s^T]([N_s][N_s^T])^{-1})_{c \times l} ([N_p][\dot{q}_p])_{l \times 1} \\
 [\dot{q}_s]_{c \times 1} &= -([N_s^T]([N_s][N_s^T])^{-1} [N_p][\dot{q}_p])_{c \times 1}
 \end{aligned} \tag{D.8}$$

Conclui-se que é possível obter \dot{q}_s a partir de uma matriz N_s , em que N_s é uma matriz não-quadrada. Assim, tem-se uma forma consistente de se obter uma solução dentro do espaço de soluções possíveis.

A matriz $N_s^T(N_s N_s^T)^{-1}$ é conhecida como pseudo-inversa à direita do Jacobiano (SICILIANO et al., 2009), neste caso adaptada para receber a matriz N_s .

Cabe ressaltar que a pseudo-inversa à direita do Jacobiano é uma possibilidade relativamente rápida de se obter um resultado do sistema quando a matriz N_s não for quadrada. Porém, ela vai otimizar as velocidades das juntas a sua maneira, não se tem um controle sobre uma determinada junta ou sobre o sistema como um todo.

Essa adaptação permite aplicar o método de Davies em SRC redundantes.

APÊNDICE E – Redundância em SRC

Uma cadeia cinemática é formada por um conjunto de elos ou corpos rígidos que são conectados por juntas. Uma cadeia cinemática pode ser classificada como aberta ou fechada. Ela é considerada aberta quando dois elos quaisquer da cadeia se conectam por somente um caminho e é considerada fechada quando dois elos quaisquer da cadeia são conectados por no mínimo dois caminhos. Assim, um manipulador é dito serial quando a cadeia cinemática for aberta e um manipulador é paralelo quando a cadeia cinemática possui pelo menos duas cadeias cinemáticas fechadas. Por fim, um manipulador é tratado como híbrido quando é composto por cadeias cinemáticas abertas e fechadas (TSAI, 2001).

Em algumas ocasiões os SRCs podem ser tratados como robôs paralelos. Isso acontece quando todos os robôs estão agregados segurando um corpo rígido, como pode ser visto na Figura 72. Esse corpo rígido pode ser, por exemplo, uma peça. Neste caso, um SRC seria tratado como um robô paralelo.

No entanto, caso algum dos robôs não tenha seu efetuador preso ao corpo rígido e deseja-se operar sobre a superfície do corpo rígido, o SRC deixa de ser um robô paralelo como ilustrado na Figura 73. Assim, caso se queira realizar um movimento relativo entre a peça e o efetuador do robô que realiza a tarefa, existem duas alternativas: movimentar o efetuador ou movimentar a peça. Essa possibilidade dupla de movimentos que causam o mesmo efeito desejado é uma característica de redundância cinemática.

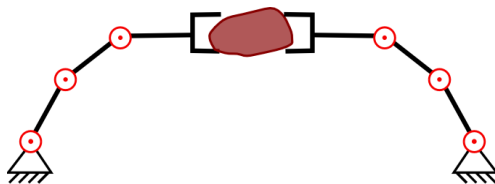


Figura 72: Um dos robôs deixa de segurar a peça.

Buscando definir o conceito de redundância em SRC, a princípio são definidos os conceitos de mobilidade ou grau de liberdade, grau de controle e conectividade de uma cadeia cinemática. As definições são elaboradas por Belfiore e Benedetto (2000), Carboni (2008), Simoni (2008), Martins e Carboni (2008).

Definição 6. *Mobilidade ou grau de liberdade de uma cadeia cinemática é definido pelo número de parâmetros independentes necessários*

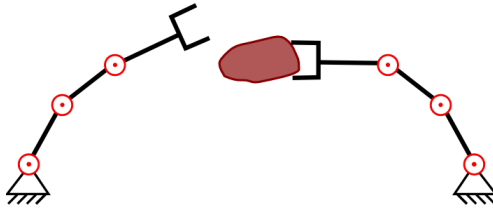


Figura 73: Um SRC segura uma peça.

para especificar completamente a configuração de uma cadeia cinemática no espaço, com respeito a um elo escolhido como referência.

Intuitivamente, pode-se dizer que a mobilidade de uma cadeia cinemática é igual ao somatório dos graus de liberdade de todos os elos menos o somatório dos graus de restrição impostos pelas juntas.

Excluindo alguns casos particulares, é possível derivar uma expressão geral para o número de graus de liberdade de uma cadeia cinemática em termos de número de elos, número de juntas e tipos de juntas. Pelo critério geral da mobilidade da cadeia cinemática pode ser escrito que:

$$M = \lambda(n - j - 1) + \sum_{i=1}^j f_i \quad (\text{E.1})$$

em que, n é o número de elos, j é o número de juntas, λ é a ordem do sistema de helicoides e f_i representa os graus de movimento relativos permitidos pela junta i . Quando uma cadeia cinemática é composta por somente juntas simples, ou seja, cada junta tem somente um grau de liberdade, a mobilidade da cadeia cinemática pode ser calculada pelo critério geral da mobilidade dado por:

$$M = \lambda(n - j - 1) + j \quad (\text{E.2})$$

Definição 7. *O grau de controle K_{ij} entre dois elos i e j de uma cadeia cinemática é o número mínimo de pares atuados independentes necessários para determinar a posição relativa entre dois elos i e j .*

Em outras palavras, as posições relativas entre dois elos não podem ser determinadas por um número de parâmetros independentes menor que o seu grau de controle.

Definição 8. *A conectividade C_{ij} entre dois elos i e j de uma cadeia cinemática é a mobilidade relativa entre os elos i e j .*

Em outras palavras, a conectividade pode ser definida como o número de graus de liberdade entre dois elos específicos em uma cadeia cinemática.

As definições de grau de controle e conectividade foram redefinidas por Carboni (2008) com o objetivo de desenvolver um método construtivo para obter os principais parâmetros de uma cadeia cinemática. As novas definições não entram em conflito com as citadas anteriormente e viabilizam o cálculo algorítmico da redundância em uma cadeia cinemática (as novas definições podem ser encontradas em (CARBONI, 2008)).

Agora, é possível definir redundância entre dois elos de uma cadeia cinemática.

Definição 9. A redundância R_{ij} entre dois elos i e j de uma cadeia cinemática é a diferença entre o número de graus de controle K_{ij} e a conectividade C_{ij} entre esses dois elos.

Para exemplificar os termos definidos acima, considerar os elos 1 e 5 da cadeia cinemática da Figura 74. Os graus de controle entre esses dois elos é $K_{15} = 4$, a conectividade é $C_{15} = 3$, pois é limitado superiormente pelo sistema de helicoides $\lambda = 3$, desta forma, a redundância é $R_{15} = 1$ (CARBONI, 2008).

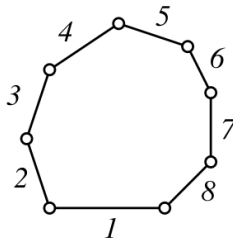


Figura 74: Cadeia cinemática fechada como $M = 5$. Para os elos 1 e 4: $K_{14} = 3$, $C_{14} = 3$ e $R_{14} = 0$ e para os elos 1 e 5: $K_{15} = 4$, $C_{15} = 3$ e $R_{15} = 1$ (CARBONI, 2008).

Desta forma, foi definido a redundância entre dois elos de uma cadeia cinemática, agora esse conceito é estendido para definir **SRC redundantes**.

Definição 10. Um SRC é redundante quando o número de graus de controle K_{ij} entre dois elos da cadeia cinemática formada pelo SRC é

maior que a conectividade C_{ij} entre esses dois elos, para algum par i, j .

De forma mais simples, um SRC é dito cinematicamente redundante quando o número de juntas disponíveis para atuar é maior do que aquelas necessárias para executar uma dada tarefa definida no espaço operacional. Assim, a redundância conforme definida quantifica quantas juntas estão disponíveis para, por exemplo, a otimização do sistema.