

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL

TESE DEFENDIDA POR Jaime Humberto Carvajal Rojas

Metodologia de Modelagem, Simulação e Programação Off-Line de Robôs e Mecanismos Mecatrônicos Integrados e Direcionados à Células de Manufatura Flexíveis E AF

PELA COMISSÃO JULGADORA EM 23/08/04

João Maurício Rosário  
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

**Metodologia de Modelagem, Simulação e  
Programação *Off-Line* de Robôs e Mecanismos  
Mecatrônicos Integrados e Direcionados à  
Células de Manufatura Flexíveis**

Autor: Jaime Humberto Carvajal Rojas

Orientador: Dr. João Maurício Rosário

08/04

BIBLIOTECA CENTRAL  
DESENVOLVIMENTO  
COLEÇÃO  
UNICAMP

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO PROJETO MECÂNICO**

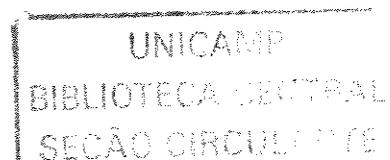
**Metodologia de Modelagem, Simulação e  
Programação *Off-Line* de Robôs e Mecanismos  
Mecatrônicos Integrados e Direcionados à  
Células Flexíveis de Manufatura**

**Autor: Jaime Humberto Carvajal Rojas  
Orientador: Dr. João Maurício Rosário**

Curso: Engenharia Mecânica  
Área de Concentração: Mecatrônica, Robótica e Sistemas Mecânicos Atuados.

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Doutor em Engenharia Mecânica.

Campinas, 2004  
S.P. – Brasil.



|            |                                     |
|------------|-------------------------------------|
| UNIDADE    | BC                                  |
| Nº CHAMADA | T/Unicamp                           |
|            | C253M                               |
| V          | EX                                  |
| TOMBO BC   | 06377                               |
| PROC.      | 16-P-00086-03                       |
| C          | <input type="checkbox"/>            |
| D          | <input checked="" type="checkbox"/> |
| PREÇO      | 11,00                               |
| DATA       | 24/11/05                            |
| Nº CPD     |                                     |

363d 373840

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C253m Carvajal Rojas, Jaime Humberto  
Metodologia de modelagem, simulação e programação *off-line* de robôs e mecanismos mecatrônicos integrados e direcionados à células de manufatura flexíveis / Jaime Humberto Carvajal Rojas.--Campinas, SP: [s.n.], 2004.

Orientador: João Maurício Rosário  
Tese (Doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robôs. 2. Modelamento matemático. 3. Simulação (Computadores). 4. Sistemas de fabricação por computador.  
I. Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: Methodology for modeling, simulation and *off-line* programming of robots and mechatronic mechanisms integrates toward flexible manufacturing cells FMC

Palavras-chave em Inglês: Robots, Mathematical modeling, Simulation computer e Computer integrated manufacturing

Área de concentração: Mecatrônica, Robótica e Sistemas Mecânicos Atuados

Titulação: Doutor em Engenharia Mecânica

Banca examinadora: José Manoel Balthazar, Vitor Ferreira Romano, José Reinaldo da Silva e Samuel Siqueira Bueno

Data da defesa: 27/08/2004

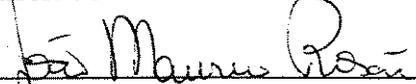
**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO PROJETO MECÂNICO**

**TESE DE DOUTORADO**

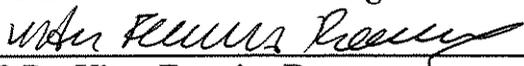
**Metodologia de Modelagem, Simulação e  
Programação *Off-Line* de Robôs e Mecanismos  
Mecatrônicos Integrados e Direcionados à  
Células de Manufatura Flexíveis**

**Autor: Jaime Humberto Carvajal Rojas**

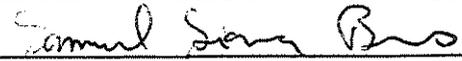
**Orientador: João Maurício Rosário**

  
\_\_\_\_\_  
**Prof. Dr. João Maurício Rosário, Presidente  
UNICAMP - Faculdade de Engenharia Mecânica**

**Prof. Dr. José Manoel Balthazar  
UNICAMP – Faculdade de Engenharia Mecânica**

  
\_\_\_\_\_  
**Prof. Dr. Vitor Ferreira Romano  
UFRJ – Universidade Federal do Rio de Janeiro**

  
\_\_\_\_\_  
**Prof. Dr. José Reinaldo da Silva  
EPUSP – Escola Politécnica da Universidade de São Paulo**

  
\_\_\_\_\_  
**Prof. Dr. Samuel Siqueira Bueno  
CenPRA – Centro Nacional de Pesquisas Renato Archer**

Campinas, 27 de agosto de 2004

200526656

## **Dedicatória:**

Dedico este trabalho de pesquisa a minha mãe Rita Elisa que desde quando era criança me mostrou o caminho do trabalho, da honestidade, da liberdade que orientou toda a minha vida; também dedico este trabalho de pesquisa a meu filho Pavel Ruffo e a minha filha Laura Clemência que sempre se mostraram pacientes e atenciosos comigo, mesmo nas minhas prolongadas ausências nas suas vidas.

## **Agradecimentos**

Tenho me maior agradecimento ao Programa de Pós Graduação da Faculdade de Engenharia Mecânica da UNICAMP por esta oportunidade de desenvolver meus estudos e minha pesquisa de doutorado trabalho em Brasil. Gostaria de expressar meu especial agradecimento ao Prof. Dr. João Maurício Rosário por sua orientação, apoio e amizade durante meus estudos e ao meu amigo e Prof. Dr. Hrishu Bera por seus conselhos profissionais e pessoais. Gostaria também de estender todos os meus sinceros agradecimentos a todos colegas, professores, técnicos e funcionários administrativos do Departamento de Projeto Mecânico da FEM-UNICAMP e Laboratório de Automação Integrada e Robótica, e em especial as secretárias Denise e Cleusa e o responsável técnico do laboratório Almiro.

*Nosso medo mais profundo  
não é o de sermos inadequados.  
Nosso medo mais profundo  
é que somos poderosos além de qualquer medida.  
É a nossa luz, não as nossas trevas,  
o que mais nos apavora.  
Nelson Mandela*

## Resumo

Carvajal Rojas, Jaime Humberto, *Metodologia de Modelagem, Simulação e Programação Off-Line de Robôs e Mecanismos Mecatrônicos Integrados e Direcionados à Células Flexíveis de Manufatura*, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2004. 168 p. Tese (Doutorado)

O principal objetivo de esta pesquisa consiste no desenvolvimento de uma metodologia geral para a integração, simulação e programação *off-line* de robôs e mecanismos mecatrônicos em células de manufatura flexíveis FMC. O desenvolvimento da metodologia implica a modelagem da FMC usando Redes de Petri PN, a modelagem geométrica com sistemas CAD do ambiente de atuação do robô incluindo equipamento e mecanismos associados, modelagem da cinemática direta e inversa do robô baseada na metodologia de Denavit - Hartenberg, modelagem do Jacobiano direto e inverso do robô, modelagem da cinemática com Quaterniones, simulação gráfica das trajetórias do robô, simulação gráfica da movimentação dos mecanismos, e coordenação e integração da movimentação dos robôs e dos mecanismos. A programação *off-line* do robô é o princípio de integração dos componentes dentro da FMC. Estudos de casos encaminhados à implementação, validação e provas da metodologia desenvolvida, são apresentados, incluindo programação *off-line* independente, programação *off-line* simultânea, programação *off-line* seqüencial e programação *off-line* integrada.

### *Palavras Chave*

Programação *Off-Line*, Simulação, Modelagem, Robôs, FMC

## **Abstract**

Carvajal Rojas, Jaime Humberto, *Methodology for Modeling, Simulation and Off-Line Programming of Robots and Mechatronic Mechanisms Integrates Toward Flexible Manufacturing Cells FMC*, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2004. 168 p. Tese (Doutorado)

The main objective of this research consists on the development of a general methodology for the integration, simulation and *off-line* programming of robots and mechatronic mechanisms into Flexible Manufacturing Cells FMC. The development of the methodology embraces the modeling of FMC using Petri Nets PN, the geometric modeling with system CAD of the environment of the robot's performance including equipment and associate mechanisms, mathematical modeling of the direct kinematics and inverse kinematics of the robot based on Denavit – Hartenberg methodology, mathematical modeling of kinematics with Quaternions, mathematical modeling of the direct Jacobian and inverse Jacobian of the robot, the graphic simulation of the robot's trajectories, graphic simulation of the movement of the mechanisms and the coordination and integration of the robot's movements and of mechanisms. The off line programming of the robots is the principle for the integration of the components. Studies of cases allows the implementation, validation and it proves of the developed methodology, are presented, including off line programming independent, *off-line* programming simultaneous, *off-line* programming sequential, and *off-line* programming integrated.

### *Key Words*

*Off-Line* Programming, Simulation, Modeling, Robots, FMC.

## Índice

|  |             |
|--|-------------|
| <b>ÍNDICE</b>  | <b>i</b>    |
| <b>LISTA DE FIGURAS</b>                                      | <b>v</b>    |
| <b>LISTA DE TABELAS</b>                                      | <b>viii</b> |
| <b>NOMENCLATURA</b>  | <b>ix</b>   |
| <b>CAPITULO 1</b>  |             |
| <b>INTRODUÇÃO</b>  | <b>1</b>    |
| <b>CAPITULO 2</b>  |             |
| <b>REVISÃO DA LITERATURA</b>                                 |             |
| <b>PROGRAMAÇÃO <i>OFF-LINE</i> DE ROBÔS INDUSTRIAIS</b>      | <b>10</b>   |
| 2. 1. Sistemas CAD   | 10          |
| 2. 2. Planejamento de trajetórias                            | 11          |
| 2. 3. Simulação  | 12          |
| 2. 4. Sistemas de Flexíveis de Manufatura (FMS)              | 18          |
| 2. 5. Metodologias para o Projeto e a Modelagem de FMS       | 22          |
| 2. 6. Colisões   | 25          |
| 2. 7. Ensino   | 27          |
| 2. 8. A Robótica na UNICAMP                                  | 28          |
| 2. 9. Contribuições desta Pesquisa                           | 30          |
| <b>CAPITULO 3</b>  |             |
| <b>MODELAGEM CAD DE ROBÔS E DE DISPOSITIVOS MECATRÔNICOS</b> | <b>34</b>   |
| 3. 1. Conceitos Básicos Direcionados à FMC                   | 37          |
| 3. 1. 1. Sistemas Automatizados de Produção (SAP)            | 37          |
| 3. 1. 2. Tecnologia de Grupo (GT) e Manufatura Celular (CM)  | 40          |
| 3. 1. 3. Flexibilidade dos Sistemas de Manufatura            | 41          |
| 3. 1. 4. FMS   | 42          |
| 3. 2. Modelagem com Sistemas CAD                             | 46          |
| 3. 2. 1. Linguagem   | 47          |
| 3. 2. 2. Álgebra Booleana                                    | 48          |
| 3. 3. Células Robotizadas                                    | 49          |

|   |    |
|---|----|
| 3. 3. 1. Operações dos Robôs Industriais.               | 49 |
| 3. 3. 2. Projeto Mecatrônico de Robôs                   | 50 |
| 3. 3. 2. 1. Modelagem                                   | 52 |
| 3. 3. 2. 2. Simulação                                   | 53 |
| 3. 3. 2. 3. Integração                                  | 53 |
| 3. 3. 3. Critérios de Seleção de Robôs                  | 54 |
| 3. 3. 4. Ferramenta Terminal                            | 56 |
| 3. 3. 4. 1. Punho do Robô                               | 56 |
| 3. 3. 4. 2. Garra do robô E Sistema de Coordenadas      | 58 |
| 3. 3. 4. 3. Ferramenta do Robô e Sistema de Coordenadas | 59 |
| 3. 3. 5. Espaço de Trabalho do Robô                     | 60 |
| 3. 4. Modelagem dos Mecanismos                          | 62 |
| 3. 5. Integração de Robôs e Mecanismos em FMC           | 65 |
| 3. 6. Modelagem com PN das FMCs                         | 66 |

## **CAPITULO 4**

|   |           |
|---|-----------|
| <b>INTEGRAÇÃO DA CINEMÁTICA DE ROBÔS EM FMC</b>             | <b>69</b> |
| 4. 1. Modelagem da Cinemática Direta com MTH                | 70        |
| 4. 2. Modelagem da Cinemática Inversa com MTH               | 76        |
| 4. 3. Quaternions: Números Hiper Complexos                  | 81        |
| 4. 4. Sistemas de Coordenadas com Quaternions               | 83        |
| 4. 4. 1. Quaternions de Posição                             | 88        |
| 4. 4. 2. Quaternions de Rotação                             | 89        |
| 4. 5. Modelagem da Cinemática com Quaternions               | 89        |
| 4. 6. De MTH a Quaternions e Vice-versa                     | 90        |
| 4. 7. Cinemática Diferencial Direta                         | 91        |
| 4. 7. 1. Diferencial de Translação                          | 91        |
| 4. 7. 2. Diferencial de Rotação                             | 92        |
| 4. 7. 3. Diferencial de Translação + Diferencial de Rotação | 93        |
| 4. 7. 4. A matriz Jacobiano                                 | 93        |
| 4. 7. 5. Modelagem da Matriz Jacobiano                      | 97        |
| 4. 8. Modelagem da Cinemática Diferencial Direta            | 103       |
| 4. 9. Modelagem da Cinemática Diferencial Inversa           | 104       |
| 4. 10. Modelagem da Cinemática das FMCs                     | 105       |

## **CAPITULO 5**

|  |            |
|--|------------|
| <b>SIMULAÇÃO E PROGRAMAÇÃO <i>OFF-LINE</i> DE FMC</b>  | <b>108</b> |
| 5. 1. Simulação  | 110        |
| 5. 2. PGs  | 110        |
| 5. 2. 1. Obtenção dos PGs a Partir das Coordenadas     | 110        |
| 5. 2. 2. Criação dos PGs Através da Aprendizagem       | 111        |
| 5. 2. 3. Obtenção dos PGs em Arestas dos Objetos CAD   | 112        |
| 5. 2. 4. Obtenção dos PGs em Interseção de Arestas     | 112        |
| 5. 2. 5. Obtenção dos PGs em Interseção de Superfícies | 113        |
| 5. 3. Interpolação das Trajetórias                     | 114        |

|   |     |
|---|-----|
| 5. 3. 1. Interpolação com Pontos Início e Fim                         | 115 |
| 5. 3. 1. 1. Interpolação Cúbica                                       | 116 |
| 5. 3. 1. 2. Interpolação Linear com Ajuste Parabólico                 | 116 |
| 5. 3. 2. Interpolação com Pontos Intermédios                          | 118 |
| 5. 3. 2. 1. Interpolação Cúbica                                       | 118 |
| 5. 3. 2. 2. Interpolação Linear com Ajuste Parabólico                 | 119 |
| 5. 4. Geração de Trajetórias  | 121 |
| 5. 5. Detecção e Tratamento de Colisões                               | 125 |
| 5. 6. Ajuste e Otimização das Trajetórias                             | 127 |
| 5. 7. Simulação e Animação  | 127 |
| 5. 8. Programação <i>Off-line</i>                                     | 128 |
| 5. 8. 1. Preparação do Robô, Linguagem e Célula                       | 129 |
| 5. 8. 2. Tipos de Programação <i>Off-line</i>                         | 130 |
| 5. 9. Metodologia da Simulação e Programação <i>Off-Line</i> das FMCs | 132 |

## CAPITULO 6

|   |            |
|---|------------|
| <b>VALIDAÇÃO DA SIMULAÇÃO E PROGRAMAÇÃO <i>OFF-LINE</i> DE FMCs</b>     | <b>134</b> |
| 6. 1. Exemplo 1, FMC com dois robôs de soldagem e mecanismo MB e MR     | 135        |
| 6. 1. 1. PN para FMC com Programação <i>Off-line</i> Independente       | 136        |
| 6. 1. 2. PN para FMC com Programação <i>Off-line</i> Simultânea         | 137        |
| 6. 1. 3. PN para FMC com Programação <i>Off-line</i> Seqüencial         | 138        |
| 6. 1. 4. PN para FMC com Programação <i>Off-line</i> Integrada          | 139        |
| 6. 2. Modelagem CAD de Dispositivos Mecatrônicos                        | 140        |
| 6. 3. Modelagem da Cinemática do Robô                                   | 143        |
| 6. 4. Simulação   | 147        |
| 6. 5. Simulação e Programação <i>Off-line</i> Independente              | 148        |
| 6. 6. Simulação e Programação <i>Off-Line</i> Simultânea                | 149        |
| 6. 7. Simulação e Programação <i>Off-Line</i> Seqüencial                | 152        |
| 6. 8. Simulação e Programação <i>Off-Line</i> Integrada                 | 153        |
| 6. 9. Célula Robotizada da UNICAMP e FMC Proposta                       | 155        |
| 6. 10. Exemplo 2, FMC de Soldagem com dois Robôs e Mecanismo de Rotação | 158        |
| 6. 11. Exemplo 3, FAC com dois Robôs e Mecanismo de Rotação             | 161        |
| 6. 12. Exemplo 4, FMC com um Robô e duas Máquinas CNC: Três variantes   | 163        |
| 6. 13. Exemplo 5, FMS com um Robô e cinco Máquinas NC                   | 169        |

## CAPITULO 7

|   |            |
|---|------------|
| <b>CONCLUSÕES E SUGESTÕES</b>                         | <b>175</b> |
| 7. 1. Conclusões                                      | 175        |
| 7. 2. Sugestões para Trabalhos Futuros                | 176        |
| 7. 2. 1. Desenvolvimento da CIM                       | 176        |
| 7. 2. 3. Ensino de Robótica e FMS no Ambiente Virtual | 177        |

|                                   |            |
|-----------------------------------|------------|
| <b>REFERÊNCIAS BIBLIOGRÁFICAS</b> | <b>179</b> |
|-----------------------------------|------------|

|                 |            |
|-----------------|------------|
| <b>ANEXO 1.</b> | <b>195</b> |
|-----------------|------------|

|   |            |
|---|------------|
| <b>MECANISMOS PARA GARRA DE ROBÔS</b>                   | <b>195</b> |
| Mecanismo de Quatro Barras Paralelas                    |            |
| Mecanismo com Levas em Paralelo                         |            |
| Mecanismo com Leva Cônica                               |            |
| Mecanismo com Deslocamento em Paralelo, Simples         |            |
| Mecanismo com Deslocamento em Paralelo, Duplo           |            |
| Braçadeira com Parafusos de Bolas                       |            |
| Braçadeira com Dobre Cremalheira e Pinhão               |            |
| Braçadeira com Cremalheira e Dobre Pinhão               |            |
| Braçadeira de Ação Simples                              |            |
| Mecanismo com Dois Pivôs                                |            |
| <br>  |            |
| <b>ANEXO 2.</b>   |            |
| <b>METODOLOGIA DE DENAVIT – HARTENBERG</b>              | <b>200</b> |
| <br>  |            |
| <b>ANEXO 3.</b>   |            |
| <b>MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEAS MTH</b>         | <b>204</b> |
| <br>  |            |
| <b>ANEXO 4.</b>   |            |
| <b>SIMULAÇÃO COM O APLICATIVO VISUAL BASIC VBA</b>      | <b>207</b> |
| Modelagem   |            |
| Trajetórias   |            |
| Simulação   |            |
| <br>  |            |
| <b>ANEXO 5.</b>   |            |
| <b>SIMULAÇÃO E PROGRAMAÇÃO COM APRENDIZAGEM VIRTUAL</b> | <b>214</b> |

## Lista de Figuras

|  |     |
|--|-----|
| Figura 1. 1. Arquitetura da metodologia de simulação e programação <i>off-line</i> de robôs integrados em FMC      | 3   |
| Figura 1. 2. Níveis de complexidade da programação <i>off-line</i> de robôs industriais em FMC                     | 4   |
| Figura 1. 3. FMC de 15 gdl a implementar-se na UNICAMP com o robô IRB 1400, o robô IRB 1500 e mecanismo MR + MB    | 6   |
| Figura 3. 1: Mecatrônica e CE em FMC.  | 35  |
| Figura 3. 2: Arquitetura aberta de integração mecatrônica para o projeto de FMCs direcionados aos CIM              | 36  |
| Figura 3. 3: Níveis de Automação Industrial.   | 38  |
| Figura 3. 4: Funções nos FMS.  | 39  |
| Figura 3. 5: Equipamento básico de uma FMC.  | 44  |
| Figura 3. 6: Modelagem CAD de peças cilíndricas e peças prismáticas.   | 49  |
| Figura 3. 7: Metodologia para o projeto mecatrônico de robôs industriais.  | 51  |
| Figura 3. 8: Modelagem CAD de robô de configuração angular RRR de 3 GL   | 53  |
| Figura 3. 9: Cadeia cinemática do punho esférico.  | 57  |
| Figura 3.10: Configuração ortogonal do punho esférico.   | 58  |
| Figura 3.11: Mecanismo de garra e sistema de coordenadas   | 59  |
| Figura 3. 12: Ferramentas para soldagem de arco elétrico e soldagem por pontos.                                    | 60  |
| Figura 3. 13. Espaço de trabalho do robô IRB1400 de ABB e da FMC.  | 62  |
| Figura 3. 14: Modelagem dos mecanismos   | 63  |
| Figura 3. 15. FMC com dois Robôs e Mecanismo MB e MR   | 66  |
| Figura 4.1: Robô industrial IRB 1400 de ABB.   | 70  |
| Figura 4. 2: Rotação e translação das coordenadas do garra o ferramenta em função das coordenadas da base do robô. | 71  |
| Figura 4. 3: Sistemas de coordenadas.  | 73  |
| Figura 4. 4: Condição de resolubilidade de Pieper (1968)   | 77  |
| Figura 4. 5: Sistema de coordenadas do robô IRB 1400 para a modelagem com Quaternions                              | 87  |
| Figura 4. 6: Cinemática diferencial direta.  | 94  |
| Figura 4. 7: Velocidade e aceleração nas juntas em Workspace™  | 106 |
| Figura 4. 8: Propriedades das juntas do IRB 1400 da ABB em Workspace™  | 106 |
| Figura 4. 9: Parâmetros de Denavit - Hartenberg do robô PUMA™ em Workspace™  | 107 |
| Figura 5. 1. Obtenção do PG por coordenadas.   | 111 |
| Figura 5. 2. Obtenção do PG através da aprendizagem.   | 111 |
| Figura 5. 3. PG em aristas.  | 112 |
| Figura 5. 4. Obtenção do PG em interseção de aristas.  | 113 |
| Figura 5. 5. Obtenção do PG em intersecção de superfícies.   | 114 |
| Figura 5. 6. Diferentes trajetórias para uma junta.  | 115 |

|   |     |
|---|-----|
| Figura 5. 7. Interpolação linear com ajuste parabólico.   | 117 |
| Figura 5. 8. Trajetória com pontos intermédios.   | 119 |
| Figura 5. 9. Interpolação linear com ajuste parabólico com vários pontos intermédios.                           | 120 |
| Figura 5. 10. Trajetória contínua linear TCL.   | 122 |
| Figura 5. 11. Trajetória contínua circular TCC.   | 122 |
| Figura 5. 12. Algoritmo de interpolação PG.   | 123 |
| Figura 5. 13. Trajetórias com três tipos de PGs.  | 125 |
| Figura 5. 14. Detecção de colisões.   | 126 |
| Figura 5. 15. Opções de simulação e animação.   | 128 |
| Figura 5. 16. Complexidade na programação <i>off line</i> .   | 131 |
| Figura 6. 1. Modelagem em Rede de Petri PN para uma FMC com programação <i>off-line</i><br>Independente         | 136 |
| Figura 6. 2. Modelagem em Rede de Petri PN para uma FMC com programação <i>off-line</i><br>simultânea.          | 137 |
| Figura 6. 3. Modelagem em Rede de Petri PN para uma FMC com programação <i>off-line</i><br>seqüencial.          | 138 |
| Figura 6. 4. Modelagem em Rede de Petri PN para uma FMC com programação <i>off-line</i><br>integrada.           | 139 |
| Figura 6. 5. Integração mecatrônica dos componentes da FMC  | 141 |
| Figura 6. 6. Vista superior do espaço de trabalho dos robôs da FMC  | 142 |
| Figura 6. 7. Modelagem CAD do espaço de trabalho da FMC   | 142 |
| Figura 6. 8. Dois robôs cooperativos e mecanismo dentro da FMC  | 143 |
| Figura 6. 9. Posição em coordenadas cartesianas e orientação em Quaternions dos robôs                           | 143 |
| Figura 6. 10. Posição e orientação das ferramentas de soldagem elétrico CAT241D                                 | 145 |
| Figura 6. 11. Propriedades do robô IRB 1400   | 146 |
| Figura 6. 12. Propriedades de posição e orientação do mecanismo.  | 146 |
| Figura 6. 13. PGs para o robô IRB 1400 e IRB 1500, respectivamente  | 149 |
| Figura 6. 14. PGs, trajetória e POL Independente para IRB 1400 de ABB   | 148 |
| Figura 6. 15. PGs, trajetória e POL Simultânea para IRB 1500 de ABB   | 150 |
| Figura 6. 16. Propriedades e linguagem do robô IRB 1500 de ABB.   | 151 |
| Figura 6. 17. Limites e cinemática dos robôs.   | 151 |
| Figura 6. 18. Trajetórias e Sinais Digitais para POL seqüencial dos robôs<br>IRB 1400 de ABB e IRB 1500 de ABB. | 153 |
| Figura 6. 19. Trajetórias integradas dos robôs IRB 1400 de ABB e IRB 1500 de ABB                                | 154 |
| Figura 6. 20. Sinais digitais de simulação e programação <i>off-line</i> integrada.                             | 154 |
| Figura 6. 21. FMC implementa na UNICAMP de 15 GL com dois robôs de ABB<br>e mecanismo MR + MB                   | 156 |
| Figura 6. 22. Célula robotizada da UNICAMP.   | 157 |
| Figura 6. 23. Robô IRB 1400 e IRB 140 da UNICAMP  | 157 |
| Figura 6. 24. FMC de soldagem com dois robôs  | 160 |
| Figura 6. 25. FMC de montagem com dois robôs  | 162 |
| Figura 6. 26. Variante 1: FMC com robô centralizado   | 165 |
| Figura 6. 27. Variante 2: FMC com robô centralizado e deslocamento  | 166 |
| Figura 6. 28. Variante 3: FMC com dois robôs dedicados  | 168 |
| Figura 6. 29. FMC com um robô e cinco máquinas NC   | 172 |
| Figura 7. 1. Integração – CIM   | 176 |
| Figura 7. 2. Ensino de robótica no ambiente virtual   | 168 |

## Lista de Tabelas

|   |     |
|---|-----|
| Tabela 3.1. Configurações fundamentais de robôs comerciais  | 55  |
| Tabela 4. 1. Parâmetros e variáveis de Denavit – Hartenberg   | 73  |
| Tabela 4. 2. Matrizes de transformação de coordenadas homogêneas  | 74  |
| Tabela 6.1: Programa em RAPID para a trajetória do robô IRB 1400  | 149 |
| Tabela 6.2: Programa em RAPID para a trajetória do robô IRB 1500  | 152 |
| Tabela 6. 3. Resumo de desenvolvimento de FMC baseadas na metodologia para simulação e programação <i>off-line</i> de robôs | 171 |

## Nomenclatura

### *Letras Latinas*

|   |  |
|---|--|
| <b>ai</b>                                   | Parâmetro distancia de Denavit – Hartenberg                              |
| <b>di</b>                                   | Parâmetro distancia de Denavit – Hartenberg                              |
| <b><math>\mathbf{p}_{3 \times 1}</math></b> | Vetor de translação de 3x1   |
| <b>q</b>                                    | Variável da junta  |
| <b><math>\dot{q}</math></b>                 | Velocidade da junta  |
| <b><math>\ddot{q}</math></b>                | Aceleração da junta  |
| <b><math>\mathbf{A}_i^{i-1}</math></b>      | Matriz de transformação homogênea do elemento 1 respeito ao elemento i-1 |
| <b><math>\mathbf{E}_{1 \times 1}</math></b> | Escalamento  |
| <b>J</b>                                    | Matriz Jacobiano   |
| <b>K</b>                                    | Constante de Quaternion  |
| <b><math>\mathbf{O}_{1 \times 3}</math></b> | Transformação em perspectiva de 1x3                                      |
| <b>Q</b>                                    | Quaternion   |
| <b><math>\mathbf{R}_{3 \times 3}</math></b> | Matriz de rotação de 3x3   |
| <b><math>\mathbf{T}_n^0</math></b>          | Matriz de transformação homogênea resultante                             |
| <b><math>V_A</math></b>                     | Velocidade angular   |
| <b><math>V_L</math></b>                     | Velocidade linear  |

### *Letras Gregas*

|                              |                               |
|------------------------------|-------------------------------|
| <b><math>\alpha_i</math></b> | Angulo entre eixos de junta   |
| <b><math>\theta_i</math></b> | Angulo da junta               |
| <b><math>\beta</math></b>    | Ângulo de Quaternion          |
| <b><math>\sum_i^m</math></b> | Somatória dos elementos i a m |

### *Abreviações*

|             |  |
|-------------|--|
| <b>AGV</b>  | Automatically Guided Vehicle           |
| <b>AMC</b>  | Autonomous Manufacturing Cell          |
| <b>AMS</b>  | Autonomous Manufacturing Systems       |
| <b>AMHS</b> | Automatically Materials Handled System |

|                  |  |
|------------------|--|
| <b>CAD</b>       | Computer Aided Design                                  |
| <b>CAM</b>       | Computer Aided Manufacturing                           |
| <b>CAD / CAM</b> | Computer Aided Design and Computer Aided Manufacturing |
| <b>CCD</b>       | Charge-couple device                                   |
| <b>CE</b>        | Concurrent Engineering                                 |
| <b>CIM</b>       | Computer Integrated Manufacturing                      |
| <b>CIMOSA</b>    | Open System Architecture for CIM                       |
| <b>CM</b>        | Cellular Manufacturing                                 |
| <b>CMM</b>       | Computer Machine Measurement                           |
| <b>CNC</b>       | Computer Numerical Control                             |
| <b>CP</b>        | Continuous Path  |
| <b>CPS</b>       | Cellular Production System                             |
| <b>FAC</b>       | Flexible Assembly Cell                                 |
| <b>FAS</b>       | Flexible Assembly System                               |
| <b>FMC</b>       | Flexible Manufacturing Cell                            |
| <b>FMRC</b>      | Force Model Referenced Control                         |
| <b>FMS</b>       | Flexible Manufacturing System                          |
| <b>FMSS</b>      | Flexible Manufacturing Station Single                  |
| <b>GP</b>        | Geometric Point  |
| <b>GT</b>        | Group Technology                                       |
| <b>IDE</b>       | Integrated Development Environment                     |
| <b>LAN</b>       | Local Area Network                                     |
| <b>LAP</b>       | Linear Approximation Problem                           |
| <b>MAS</b>       | Manufacturing Automatically System                     |
| <b>MB</b>        | Mecanismo Basculante                                   |
| <b>MCFMS</b>     | Multi Cell Flexible Manufacturing System               |
| <b>MCFMS</b>     | Multi Celd Flexible Manufacturing System               |
| <b>MHS</b>       | Materials Handled Systems                              |
| <b>MMFMS</b>     | Mul Machine Flexible Manufacturing System              |
| <b>MMFMS</b>     | Mul Machine Flexible Manufacturing System              |
| <b>MPG</b>       | Maximum Plant Graphs                                   |
| <b>MP5E</b>      | Mecanismo de Posicionar de 5 eixos                     |
| <b>MR</b>        | Mecanismo Rotatório                                    |
| <b>MTH</b>       | Matrizes de Transformação Homogêneas                   |
| <b>MWPG</b>      | Maximum Weigh Plant Graphs                             |
| <b>OLP</b>       | <i>Off-Line</i> Programming                            |
| <b>PAP</b>       | Ponto a ponto  |
| <b>PC</b>        | Personal Computer                                      |
| <b>PG</b>        | Plant Graphs   |
| <b>PG</b>        | Ponto Geométrico                                       |
| <b>PMRC</b>      | Position Model Referenced Control                      |
| <b>PN</b>        | Petri Net  |
| <b>POL</b>       | Programação <i>Off-Line</i>                            |
| <b>PTP</b>       | Point to point   |
| <b>QAP</b>       | Quadratic Assigation Problem                           |
| <b>RRS</b>       | Realistic Robot Simulation                             |
| <b>SAP</b>       | Sistema Automático de Produção                         |

|               |                              |
|---------------|------------------------------|
| <b>SFM</b>    | Single Manufacturing Machine |
| <b>SLP</b>    | Systematic Layout Plant      |
| <b>TCC</b>    | Trajatória contínua circular |
| <b>TCL</b>    | Trajatória contínua linear   |
| <b>TCP</b>    | Tool Center Point            |
| <b>U</b>      | Utilization                  |
| <b>VBA</b>    | Visual Basic Applications    |
| <b>VR</b>     | Virtual Reality              |
| <b>VR-PAD</b> | VR Point-and-Direct          |
| <b>VRS</b>    | Virtual Robot Simulation     |
| <b>WCS</b>    | Working Coordinate System    |
| <b>WIP</b>    | Work In Process              |
| <b>WOM</b>    | Workspace Object Model       |
| <b>WRL</b>    | Workspace Robot Language     |
| <b>XP</b>     | Mecanismo de posicionar      |

***Siglas***

|            |   |
|------------|---|
| <b>DPM</b> | Departamento de Projeto Mecânico              |
| <b>FEM</b> | Faculdade de Engenharia Mecânica              |
| <b>LAR</b> | Laboratório de Automação Integrada e Robótica |

## Capítulo 1

### Introdução

Um importante avanço na tecnologia da robótica é a programação *off-line* com sistemas CAD (CAD Computer Aided Design) que permite um projeto próximo à realidade de células flexíveis de manufatura e / ou robotizadas, com planejamento e distribuição de equipamentos e componentes em ambiente virtual, permitindo também o planejamento de tarefas de manipulação de braços robóticos.

A simulação é o primeiro passo necessário para a programação *off-line*, onde a partir da especificação funcional de uma determinada célula de trabalho, tem início a partir do modelo geométrico dos componentes da célula de trabalho para definição da configuração da planta (ambiente de trabalho) e posterior definição das trajetórias a serem realizadas pelo robô, a partir da definição exata dos pontos geométricos, a seqüência das trajetórias, velocidades, acelerações e forças.

As técnicas de simulação e programação *off-line* permitem a depuração dos programas fora da linha de produção, sem interrupção do fluxo normal de trabalho. Isto proporciona notáveis vantagens, como a redução do tempo não produtivo do robô, permitindo otimizar e minimizar os tempos de ciclo de tarefas.

A simulação permite observarmos o movimento gráfico do robô a partir da realização das trajetórias implementadas e sua interação com todos os mecanismos que intervierem de

forma paralela na célula. Ela permite a definição das melhores trajetórias a serem realizadas pelo robô, e a comprovação do correto funcionamento do sistema, para posterior elaboração de programas estruturados que deverão ser transferidos diretamente ao controlador.

A programação *off-line* conta com diversas vantagens sobre a programação manual e textual, entre elas podemos destacar a redução do tempo de parada do robô, a possibilidade de analisar situações complexas tais como tratamento de colisões e melhor acessibilidade, e a possibilidade de realizar a programação previamente à fase de desenvolvimento da instalação, diminuindo custos iniciais e tempo de implementação do projeto.

Por outro lado, o problema de programação e controle de sistemas robóticos cooperativos envolve um diversificado conhecimento de áreas e disciplinas tecnológicas e atualmente o campo de aplicações vêm crescendo em aplicações cada vez mais importantes tais como: a robótica espacial, onde dois robôs desenvolvem tarefas cooperativas de intervenção e manutenção, na área médica, em cirurgias onde dois robôs são necessários para manipular cooperativamente um objeto, em soldagem contínua de peças complexas, onde a peça é manipulada por um robô enquanto o outro realiza a operação de soldagem, e finalmente, em sistemas de manufatura e montagem, onde o alto nível de automatização esta crescendo em demanda.

Neste trabalho de pesquisa, dá continuidade a outros trabalhos desenvolvidos no Laboratório de Automação Integrada e Robótica - LAR da Faculdade de Engenharia Mecânica (FEM) da UNICAMP na área de Automação e Robótica, sendo proposto uma metodologia para a modelagem, simulação e programação *off-line* de robôs e outros dispositivos mecatrônicos integrados para configurar células flexíveis de manufatura FMC (FMC Flexible Manufacturing Cell ) e sua arquitetura é apresentada na Figura 1.1

O LAR - UNICAMP dispõe de infra-estrutura necessária para o desenvolvimento desse trabalho de pesquisa, contando com dois robôs industriais o IRB 1400 e IRB 140 da ABB<sup>TM</sup>, além dos aplicativos Mathematic 4.0<sup>TM</sup> e MATLAB<sup>TM</sup> utilizados neste trabalho para a realização das operações matemáticas e algébricas referentes aos modelos

implementados e os softwares RobotStudio™ e Workspace 5™ utilizados para o desenvolvimento da modelagem, simulação e programação *off-line*.

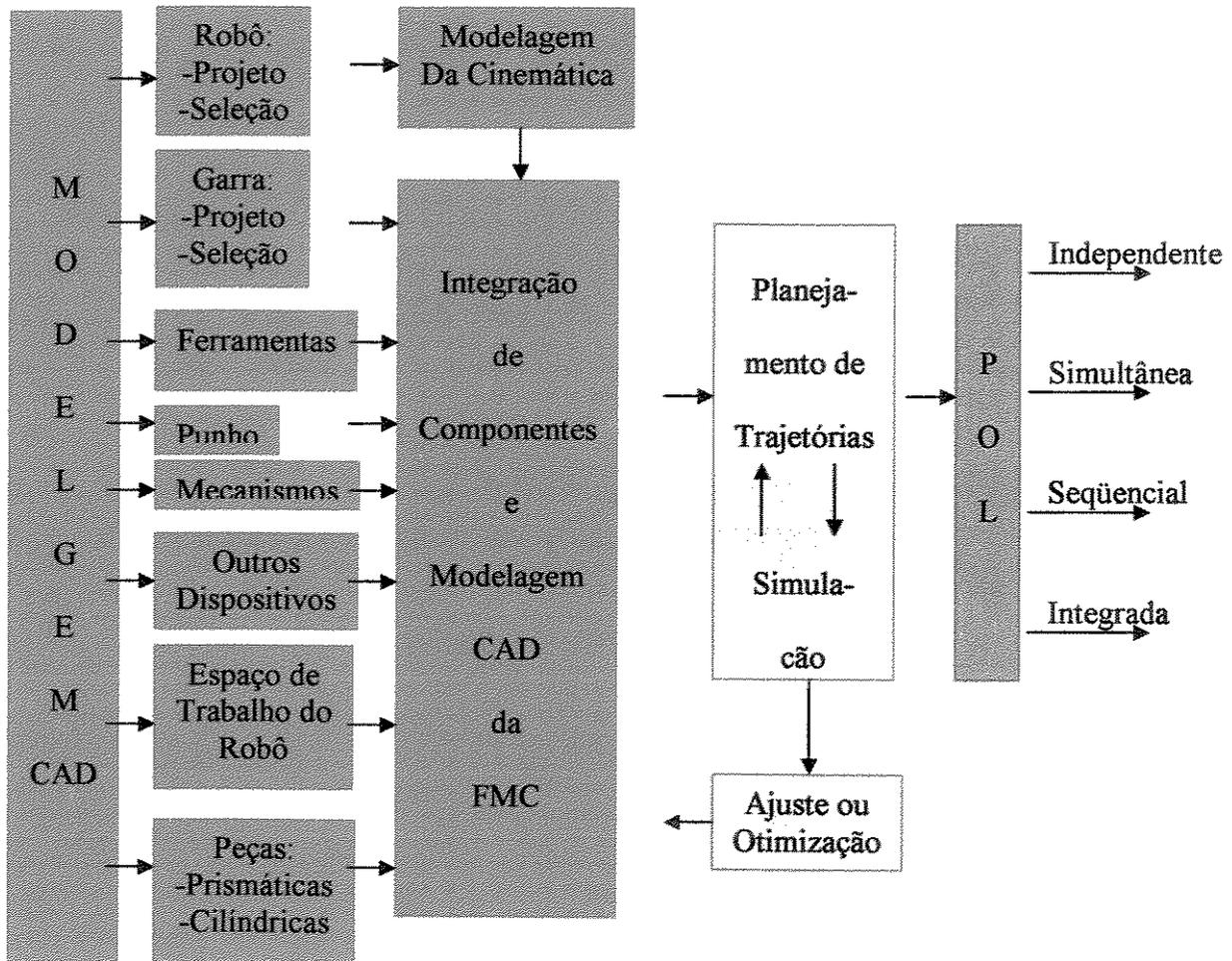


Figura 1. 1. Arquitetura da metodologia de simulação e programação *off-line* De robôs integrados em FMC

## Contribuições do Trabalho

Este trabalho de pesquisa contribui através do seguinte desenvolvimento:

(1) Apresentação de metodologia geral de simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados em FMC, baseada em três tarefas:

- a) Modelagem cinemática do robô e do seu ambiente de trabalho;
- b) Planejamento de trajetórias e simulação da movimentação dos robôs e mecanismos;
- c) Programação *off-line* dos robôs e mecanismos, conforme arquitetura apresentada anteriormente na Figura 1.1

(2) Definição de quatro níveis de complexidade na programação *off-line*:

Nível 1. Programação *off-line* independente: para um robô

Nível 2. Programação *off-line* simultânea: para dois robôs com movimentação simultânea.

Nível 3. Programação *off-line* seqüencial: para dois robôs com movimentação em serie, enviando e recebendo sinais digitais sincronizadas.

Nível 4. Programação *off-line* integrada: para um ou dois robôs com movimentação integrada com a movimentação dos mecanismos, enviando e recebendo sinais digitais sincronizados. Esses níveis de programação *off-line* são apresentados na Figura 1.2

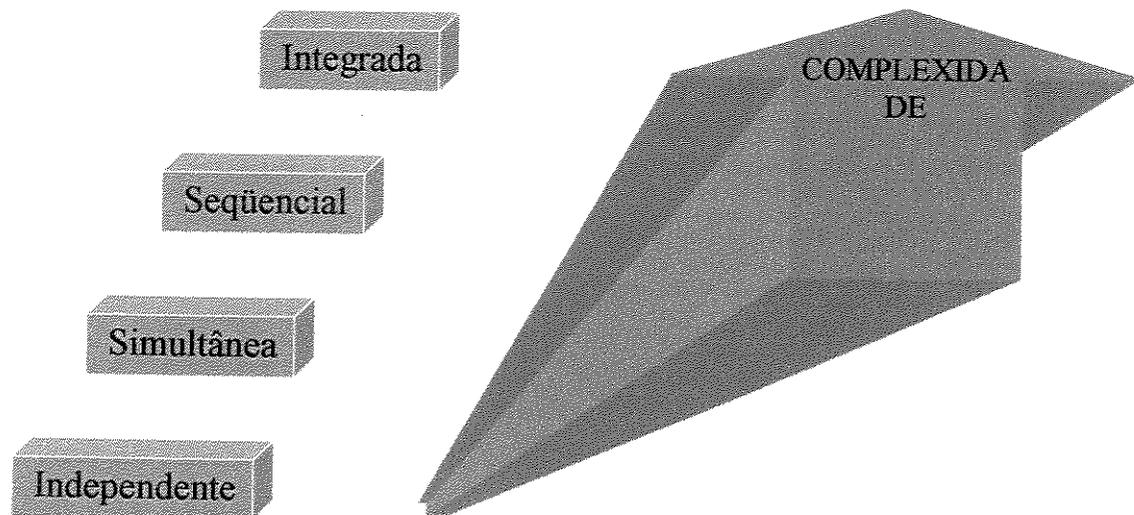


Figura 1.2. Níveis de complexidade da programação *off-line* de robôs industriais em FMC

(3) Esta metodologia é aplicada e os níveis de programação *off-line* para validação do trabalho através de cinco exemplos, desenvolvidos a partir da utilização do software Workspace™ no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP:

Exemplo 1: FMC com dois robôs de soldagem e mecanismo MB - MR.

Exemplo 2: FMC com um robô de soldagem, um robô de posicionamento e mecanismo de rotação.

Exemplo 3: Célula de montagem flexível FAC (FAC Flexible Assembly Cell) com dois robôs.

Exemplo 4: FMC com um robô de posicionamento e duas máquinas CNC: Três variantes.

Exemplo 5: FMC com um robô de posicionamento e cinco máquinas CNC.

(4) Configura uma FMC com quinze graus de liberdade direcionada ao ensino e formação de robótica na UNICAMP, baseada na metodologia proposta da simulação e programação *off-line* de robôs e mecanismos. Também, apresentam-se as modelagens da FMC em Redes de Petri PN (PN Petri Nets) para validação da:

- a) Programação *off-line* independente.
- b) Programação *off-line* simultânea.
- c) Programação *off-line* seqüencial.
- d) Programação *off-line* integrada.

A modelagem geométrica desta FMC é apresentada na Figura 1.3

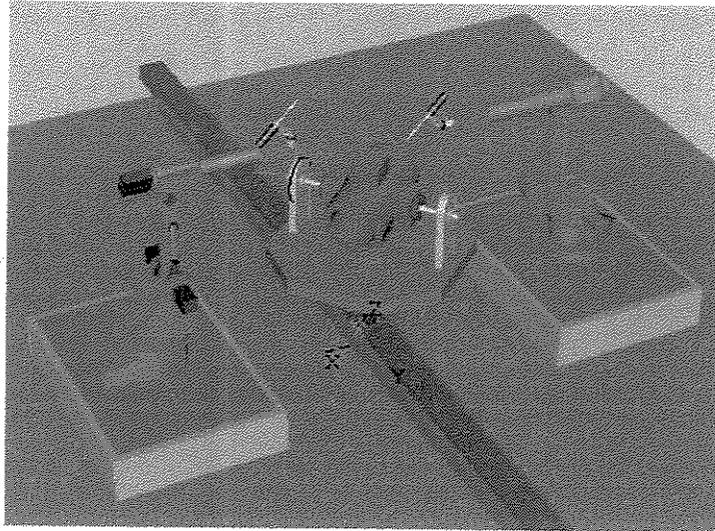


Figura 1.3. FMC com 15 gdl a implementar-se na UNICAMP com o robô IRB 1400, o robô IRB 140 e mecanismo MR + MB

(5) Apresentação de metodologia para a obtenção do sistema de coordenadas com Quaternions das juntas de um robô serial de seis graus de liberdade.

(6) Apresentação de metodologia e equação para obtenção da modelagem do posicionamento em Quaternions para a garra ou ferramenta de um robô serial com seis graus de liberdade.

(7) Apresentação de metodologia e equação para obtenção da modelagem da rotação em Quaternions da garra ou ferramenta de um robô serial com seis graus de liberdade.

(8) Apresentação de metodologia para obter a modelagem da cinemática com Quaternions para o robô serial IRB 1400 de ABB de seis graus de liberdade.

(9) Desenvolvimento e implementação da modelagem cinemática direta e inversa dum robô industrial serial 6R baseada na sistemática de Denavit – Hartenberg e matrizes de transformação homogêneas.

(10) Apresentação de metodologia para obter a modelagem da cinemática diferencial direta e inversa baseado nas MTH para o robô serial IRB 1400 de ABB de seis graus de liberdade.

(11) Apresentação de metodologia do projeto mecatrônico de robôs industriais baseada na integração sinérgica e concorrente de três tarefas fundamentais:

a) Modelagem da configuração em ambiente CAD, modelagem da cinemática, modelagem da dinâmica, e modelagem do sistema de controle.

b) Simulação da movimentação do robô e simulação da estratégia de controle.

c) Integração sinérgica e concorrente de componentes mecânicos, eletrônicos e computadorizados para configurar o novo robô.

(12) Finalmente, este projeto de pesquisa apresenta uma ferramenta direcionada aos sistemas de manufatura de pequeno e médio porte, e pode transformar-se em FMC ou para que se possam configurar novas FMC a partir de equipamentos disponíveis no mercado, permitindo que as indústrias possam competir a nível internacional, com flexibilidade, com alto nível de produtividade, e com qualidade na produção.

Para realização dos objetivos delineados anteriormente, este documento de tese é subdividido em sete capítulos, onde na introdução é realizado um posicionamento do trabalho conforme representação esquemática apresentada na Figura 1.1.

No segundo capítulo é apresentada uma revisão bibliográfica apresentando aspectos relacionados a programação *off-line* de robôs industriais com os sistemas CAD através do planejamento das trajetórias com o tratamento de colisões, simulação gráfica, integração com os sistemas de manufatura flexíveis FMS (FMS Flexible Manufacturing Systems), e o ensino e formação na área de Robótica.

No terceiro capítulo são apresentados conceitos de modelagem através de CAD dos componentes da FMC. O robô e seu elemento terminal ou ferramenta de trabalho podem apresentar-se através de duas opções, a primeira a partir da obtenção de informações

através de uma base de dados carregada no software de simulação e outra através do desenvolvimento de um projeto mecatrônico para obtenção de sua modelagem. Os demais componentes, como mecanismos de movimentação, estações de trabalho, estações de controle de qualidade e peças de processamento, podem ser implementados a partir de um software de CAD de modelagem. Neste capítulo também é apresentada uma metodologia do projeto mecatrônico de robôs.

No quarto capítulo são apresentadas aspectos referentes a modelagem referentes à movimentação dos robôs integrados na FMC, onde pode se constatar que a simulação e programação *off-line* de robôs em geral e integrados com outros componentes da FMC, necessitam do conhecimento exato dos modelos matemáticos para sua programação e controle centralizado em uma estação computadorizada. Assim, neste capítulo os seguintes aspectos são considerados:

a) Metodologias para a modelagem da cinemática direta e inversa baseada nas matrizes de transformação homogêneas (MTH), utilizando os robôs industriais IRB 1400 de ABB<sup>TM</sup>;

b) Apresentação de metodologia para a modelagem da cinemática utilizando teoria de Quaternions, conhecidos também como números hiper-complexos. Esta metodologia apresenta modelos matemáticos para obtenção dos Quaternions de posição e orientação baseados nos parâmetros e variáveis de Denavit–Hartenberg para estes robôs;

c) Finalmente, é proposta uma metodologia para a modelagem da cinemática diferencial direta e inversa através da obtenção da matriz Jacobiano, deste robô.

No quinto capítulo, apresenta-se uma metodologia para simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados em FMC. Conseqüentemente, neste capítulo os seguintes aspectos são considerados:

a) Proposta de metodologia para a simulação através de criação de pontos geométricos (PGs), planejamento, interpolação e geração de trajetórias, análise de colisões, ajuste e otimização, e simulação gráfica e animação.

b) Apresentação de uma metodologia integrada para a simulação e programação *off-line* de robôs e mecanismos.

No sexto capítulo são apresentados exemplos práticos para validação da metodologia para simulação e programação *off-line* de robôs integrados em FMC proposta neste trabalho. Através destes exemplos, são desenvolvidos os diferentes níveis de programação *off-line* e a proposta para a configuração de uma FMC de soldagem cooperativa, que deverá ser implementada futuramente no Laboratório de Automação Integrada e Robótica da FEM - UNICAMP. Esta proposta está baseada na utilização dos robôs IRB 1400<sup>TM</sup> e IRB 140<sup>TM</sup> da ABB<sup>TM</sup>, disponíveis na UNICAMP, e um mecanismo mecatrônico com três graus de liberdade, que poderão se integrar numa FMC de quinze graus de liberdade com programação *off-line* e controle computadorizado central.

E finalmente, no sétimo capítulo, são apresentadas conclusões finais do trabalho, e futuras orientações e sugestões para outros trabalhos de pesquisa relacionados à Modelagem, Simulação e Programação *Off-Line* de Robôs e Dispositivos Mecatrônicos Integrados em Células flexíveis de manufatura.

## Capítulo 2

### Revisão da Literatura – Programação *Off-line* de Robôs

#### 2.1 - Sistemas CAD

Nos dias atuais os sistemas CAD são fundamentais para o projeto de FMS's através da simulação e programação *off-line* de robôs industriais. O desenvolvimento de sistemas CAD, em especial a modelagem através de sólidos 3D, planos e com malhas constituem ferramentas de apoio aos pesquisadores para gerar bases de informação utilizando robôs industriais disponíveis no mercado.

Este avanço na modelagem dos robôs industriais permite o desenvolvimento e implementação de programação *off-line* de robôs em FMS e sistemas de montagem flexíveis FAS (FAS Flexible Assembly System). Atualmente, os softwares disponibilizados no mercado pelos fabricantes de robôs se apóiam nos sistemas CAD para a modelagem e simulação e programação *off-line* dos robôs em seus ambientes de trabalho. Edkins e Smith (1985), Sata, Kimura, Hiraoka, Suzuki e Fujita (1987), Pollman e Dzembritzki (1987), Booney, Marshall e Green (1987), utilizaram os sistemas CAD para desenvolver estas metodologias de projetos com programação *off-line*.

## 2. 2 - Planejamento de Trajetórias

Segundo Fu, González, e Lee (1988) o planejamento de trajetórias de um robô pode ser realizado no espaço de juntas ou no espaço cartesiano. No primeiro caso, para descrição do movimento desejado do robô é planejada a evolução temporal das variáveis da junta e de suas derivadas em relação ao tempo. No segundo caso, define-se a evolução temporal da posição da garra ou ferramenta terminal, sua velocidade, e sua aceleração; obtendo-se a partir dessas informações as correspondentes posições, velocidades e acelerações das juntas. Paul (1979), Taylor (1979) e Luh e Lin (1984) propuseram trajetórias de segmentos de linhas retas conectadas por transições suaves com aceleração controlada. Edwall et al (1982) e Lin et al (1983) utilizaram funções polinomiais de curvas *spline* cúbicas para criar trajetórias. Angeles et al (1988) propõem aproximação ao planejamento da trajetória baseada no conceito de funções polinomiais *spline*, mas não resolvem as equações. Xu et al (1993) estudaram o planejamento das trajetórias dos robôs em tarefas cooperativas no ambiente bidimensional, através da operação de carregamento de peças retangulares de uma posição e orientação inicial à outra posição e orientação final.

A utilização de robôs de grande porte representa um novo domínio na robótica avançada, que mesmo possuindo os mesmos sistemas de programação dos robôs convencionais, pela sua complexidade recomenda-se a utilização da programação *off-line*. Shan e Koren (1995) propõem uma metodologia para resolver o problema da cinemática inversa em robôs altamente redundantes e apresentam o conceito de *kinematic roadmap*.

Westkämper *et al* (1998) pesquisaram sobre programação estruturada de tarefas considerando robôs redundantes (mais de seis graus de liberdade), através da descrição da orientação da ferramenta terminal através de ângulos de Euler. Neste trabalho foram estabelecidas as condições iniciais e finais da trajetória, com transições suaves e velocidade e aceleração nulas, verificadas através da matriz Jacobiana, apresentando uma metodologia para o planejamento de trajetórias próximas da ótima. São considerados

também aspectos relacionados à dinâmica do robô, configurações singulares e restrições para tratamento de colisões, onde para efeitos de simulação é considerada a presença de uma esfera no espaço de trabalho do robô representando um obstáculo.

### 2.3 - Simulação

Billingsley (1985) identifica a complexidade na implementação de programação *off-line* em células robotizadas. Edkins e Smith (1985) utilizaram programação *off-line* através de um sistema CAD para geração do programa do robô, realizando estudos concernentes à análise de precisão, repetibilidade e calibração dos robôs. Storr e Schumacher (1987), Fu, Gonzalez e Lee (1988), e Fuller (1999), revisaram através de estudos, as técnicas para a programação de robôs industriais.

Através do modelo CAD do robô e da modelagem da cinemática direta; pesquisadores como Pfeiffer e Rainer (1987) elaboraram algoritmos para o planejamento de trajetórias desejadas. Pritschow, Storr, Grulher, e Schumacher (1987) apresentam um método de programação explícito para robôs industriais, utilizando a linguagem orientada a objetos BAPS, onde o robô é guiado manualmente ao longo da trajetória descrita para a operação de montagem.

Sata, Kimura, Hiraoka, Suzuki e Fujita (1987) propõem uma metodologia de modelagem e simulação com sistemas CAD / CAM (CAD / CAM Computer Aided Design and Computer Aided Manufacturing). A partir da implementação da mesma, pode-se desenvolver movimentos refinados para montagem de componentes mecânicos através da programação *off-line* de robôs. O movimento suave é gerado automaticamente considerando a geometria do modelo virtual do sistema de montagem e através da programação *off-line* do robô, permitindo assim uma alta flexibilidade e produtividade.

A modelagem detalhada do ambiente de trabalho do robô é condição favorável para a programação *off-line*; neste sentido Pollman e Dzembitzki (1987) também se apoiaram em sistemas CAD para programação *off-line* de robôs em tarefas de montagem e

soldagem por arco e através de pontos, utilizando as seguintes ferramentas: sistema CAD, cinemática do robô ao nível de junta, análise e tratamento de colisões, planejamento de trajetórias PTP (PTP Point-to-point) e CP (CP Continuous Path), programação por aprendizagem e sistema de aquisição de informações.

Booney, Marshall e Green (1987) da Universidade de Nottingham desenvolveram o software GRAPS<sup>TM</sup>, um sistema de simulação gráfica de robôs com modelador geométrico de sólidos em 3D e modelador da cinemática do robô. Os usuários podem projetar o modelo do ambiente de atuação do robô através de um computador pessoal PC (PC Personal Computer). O modelo do robô contém as seguintes informações: tipo de juntas, limites de velocidades e acelerações, e configurações. Este software apresenta dois métodos para desenvolver programas interativos, e constituem uma mistura de programação textual e gráfica.

O desenvolvimento de software para simulação e programação *off-line*, Spur, Duelen e Krause (1987) projetaram quatro módulos de implementação: sistema de informação, geração do programa, simulação e para modelagem.

Stobar e Dailly (1987), descreveram o processo de implementação de um sistema de simulação com apoio do modelador de sólidos BUILD<sup>TM</sup>. Eles apresentam uma estrutura hierárquica para simulação integrada através das especificações: do modelo do robô, de sua cinemática, da linguagem de programação, da detecção de colisões, da compensação de erros em trajetórias e gerador de cinemática inversa. Neste trabalho é apresentado um exemplo de aplicação para o robô PUMA, mostrando que no interior do espaço de trabalho do robô existem pontos onde a trajetória é não controlável.

Storn e McWaters (1987) apresentam um artigo de revisão sobre a programação *off-line* de robôs industriais; analisando estratégias relativas a programação, aplicação, simulação, e ajuda para a programação direcionada ao CIM (CIM Computer Integrated Manufacturing).

Knapp (1994) desenvolve um sistema de simulação virtual do robô VRS (VRS Virtual Robot Simulation) no qual pode-se criar uma célula de trabalho virtual com robô onde o operador pode gerar rapidamente a programação *off-line* do robô para uma tarefa específica.

Troy (1995) propôs uma metodologia de programação *off-line* em VR (VR Virtual Reality) e mostraram as seguintes vantagens:

- 1) Visualização da célula robotizada antes de sua aquisição,
- 2) Criação e teste dos dispositivos de controle fora do sistema de produção,
- 3) Programação *off-line* de diferentes robôs usando um dispositivo gráfico comum e uma interface comum de programação.
- 4) Programação *off-line* rápida e modificações e teste,
- 5) Treinamento e educação do usuário a baixo custo e,
- 6) Melhoria da segurança.

Dafe (1995) da Universidade de Leeds - UK projetou um ambiente de simulação e programação *off-line* para o robô PUMA, desenvolvendo as seguintes tarefas: cinemática direta e inversa, dinâmica direta, geração de trajetórias, programação e controle do robô, interface gráfica e modelagem geométrica do robô e seu ambiente de atuação.

Jiang, Seneviratne, e Earles (1997) criaram um algoritmo para programação dinâmica e planejamento (scheduling) de dois robôs industriais em tarefas de montagem de circuitos impressos. Com a equação dinâmica Lend e Chen (1997) desenvolveram um método para planejamento de trajetórias de robôs e simulação, próxima da ótima. Consideram as singularidades da configuração e limitações de colisões. Steffen e Saramago (1997) usaram duas técnicas para otimizar o tempo nas trajetórias dos robôs através de programação *off-line*.

A realidade virtual VR (VR Virtual Reality) é uma metodologia apropriada para o desenvolvimento da programação *off-line* de robôs industriais. Boud e Steiner (1997)

desenvolvem uma metodologia de aprendizagem de tarefas de montagem robotizado baseado em três módulos:

- 1) Modulo de mapeio do ambiente do robô,
- 2) Modulo de interpretação das tarefas ao nível de comando baseado em DataGlobe
- 3) Modulo de re-planejamento das trajetórias baseado em VR-PAD (VR Point – and - Direct). Finalmente, com os três módulos, propõem uma metodologia de programação *off-line* de robôs, com os seguintes componentes:
  - a) ambiente do operador com dispositivos mecatrônicos e computadorizados,
  - b) interface de estação computadorizada - célula de trabalho do robô,
  - c) sistema de programação *off-line* com funções de planejamento de trajetórias,
  - d) tratamento de colisões e calibração,
  - e) célula de trabalho do robô.

Em 1999, Ana Djuric utiliza o software Workspace para desenvolver sua dissertação de mestrado em Windsor University, Canadá, sobre modelagem simulação e projeto de células automatizadas industriais a baixo custo. O desenvolvimento dos projetos de manufatura inclui sistemas totalmente robotizados com mais de dois robôs e sistemas híbridos de interação de sistemas robotizados com atuação manual dos operadores.

Ahuactzin e Gupta (1999) desenvolvem um sistema de programação *off-line* para robôs redundantes de manutenção de aviões, através da integração de três metodologias:

- 1) Metodologia de planejamento de trajetórias com curvas *spline*, onde o polinômio desta curva é utilizado para o controle do robô;
- 2) Metodologia de configuração do planejamento de acordo com as curvas *splines* satisfazendo requerimentos de otimização da energia e tratamento de singularidades.
- 3) Metodologia de planejamento do movimento do robô ao longo da reta, considerando as restrições cinemáticas e dinâmicas.

As três metodologias integram-se numa metodologia para programação *off-line*, desenvolvendo os seguintes passos:

- 1) Início.
- 2) Modelagem da célula robotizada.
- 3) Programação orientada às tarefas.
- 4) Planejamento de trajetórias.
- 5) Planejamento da configuração das trajetórias.
- 6) Planejamento do movimento.
- 7) Simulação do movimento.
- 8) Sistemas de correção.
- 9) Compilação para linguagem de programação do robô.
- 10) Transmissão do programa compilado ao sistema de controle do robô.
- 11) Fim.

Rentería e Rivas (2000) afirmam que o software de simulação é a chave para otimizar projetos de ambientes robotizados antes de implantá-los na realidade, porque esse procedimento representa uma economia de tempo e dinheiro. Assim, a modelagem de células robotizadas de forma virtual começa a partir da análise dos seguintes elementos: alcance do robô, colisões, tempos de ciclo, CAD e distribuição do equipamento em planta. O processo de simulação se desenvolve através de três passos: modelagem geométrica, planejamento de trajetórias e simulação.

Zha e Du (2001) apresentam uma nova aproximação para geração e otimização da posição e orientação de trajetórias no espaço cartesiano, e simulação e programação *off-line* virtual baseado num sistema CAD. O sistema desenvolvido chamado ROBOSIM é integrado no ambiente virtual CAD para o planejamento das trajetórias e simulação do robô onde a programação *off-line* é gerada automaticamente.

O pós-processamento é um complemento importante na programação *off-line* porque permite uma tradução das expressões do simulador à linguagem do robô. Tipicamente, a seqüência de movimento do robô pode ser transmitida através de interface paralela ou serial, entre o computador dedicado a programação *off-line* e o controlador do robô.

Chan e Kwan (2003) apresentam as diferentes metodologias de pós-processamento e sua generalização, desenvolvem exemplos com GRASP e programam os robôs com VALII e propõem duas metodologias para programação *off-line* chamadas hierarquia “Top - Down” e hierarquia “Bottom - Up” que processam os dados baseados no modelo CAD. Cada uma das metodologias pode ser desenvolvida com dois módulos; um de pré-processamento e outro de pós-processamento.

Huang e Lin (2003) desenvolveram uma célula de trabalho com dois robôs, cada um de seis graus de liberdade, o sistema UNIGRAPHICS CAD / CAM foi usado para a modelagem geométrica dos componentes e trajetórias dos robôs, o software IGRIP foi aplicado para simulação das trajetórias dos robôs e evasão das colisões, e um sistema de visão com câmara CCD (CCD Charge - couple device) foi usada para a calibração dos robôs. A arquitetura da célula tem os seguintes componentes:

- 1) Modelagem CAD da célula.
- 2) Geração das trajetórias.
- 3) Simulação dos robôs e calibração.
- 4) Pós-processamento (Programação *Off-line*).
- 5) Controlador dos robôs: PC Pentium III.
- 6) Robôs IRB 1400 e IRB 2000.
- 7) Outros dispositivos complexos.

A modelagem da cinemática dos robôs esta baseada na metodologia de Denavit – Hartenberg para obter a matriz de transformação homogênea MTH resultante. Esta célula possui três configurações.

A validação de cada uma dessas configurações permite conhecer aquela de melhor desempenho. Com o sistema CAD / CAM se identificam os sistemas de coordenadas da célula para o planejamento das trajetórias e a geração da programação *off-line* acontece automaticamente com um modulo desenvolvido em ambiente Windows.

O princípio da programação *off-line* está baseado nas correspondentes relações entre as linguagens dos robôs (ARLA para IRB 2000 e RAPID para IRB 1400) e a posição e orientação das ferramentas dos robôs geradas por pós-processamento. A simulação acontece usando UNIGRAPHICS. Os níveis de programação *off-line* desenvolvidos para os dois robôs são:

- 1) Nível independente, onde os dois robôs realizam suas trajetórias em paralelo.
- 2) Nível sincronizado, onde os dois robôs realizam suas trajetórias em cooperação enviando e recebendo sinais por sincronização.
- 3) Nível coordenado, onde os dois robôs realizam suas trajetórias ao mesmo tempo.

Os resultados desta pesquisa mostram que as células com robôs duais são flexíveis e eficientes para o desenvolvimento de tarefas complexas ao nível industrial.

Perez e McCarthy (2004) apresentam uma metodologia com Quaternions duais para a síntese cinemática de sistemas robóticos restringidos. As equações da cinemática são reformuladas em termos da posição e orientação para as juntas, baseada na metodologia de Denavit - Hartenberg. Então, os Quaternions duais definem as transformações sucessivas para simplificar a estrutura de equações de desenho. Finalmente, formula-se em Quaternions uma síntese da cinemática do robô paralelo RPRP.

## 2. 4 - FMS

A implementação cada vez mais freqüente da programação *off-line* está associada especialmente ao crescimento da automatização flexível na produção através de lotes de tamanho médio, crescimento de tarefas complexas e a necessidade de reduzir o tempo de espera do robô no sistema produtivo. A evolução da pesquisa em programação *off-line* de robôs é direcionada geralmente da simulação por computador em direção a implementação dos CIM.

Uma FMC para tarefas de montagem de sistemas mecânicos foi desenvolvida através da modelagem, simulação, programação *off-line* e implementada por Rhodes,

Stenson e Blanchfield (1985). Ao mesmo tempo, o projeto de uma célula de manufatura robotizada com programação *off-line* do robô foi implementada por Weck, Niehaus e Osterwinter (1987). Este último trabalho desenvolve as seguintes etapas no projeto da célula: definição da tarefa de manufatura, seleção de equipamentos, *layout* das células, e programação *off-line*. O pacote final do software desenvolvido tem quatro módulos: modelagem, simulação, programação *off-line* e interface com o robô.

Nos sistemas de manufatura flexíveis multi-máquinas MMFMS (MMFMS Mul Machine Flexible Manufacturing System) e sistemas de manufatura flexíveis multi - células MCFMS (MCFMS Multi Celd Flexible Manufacturing System) é comum termos mais de um robô cooperando em tarefas de manufatura e montagem. Suh e Shin (1989) utilizaram dois robôs PUMA 560<sup>TM</sup> para deslocar peças rígidas, grandes, longas e pesadas. As tarefas dos dois robôs foi movimentar o corpo rígido de um ponto a outro, ao longo de uma trajetória definida previamente dentro do espaço de trabalho dos robôs, e com limitações nas juntas. O robô principal realiza a operação de apreensão de um extremo do corpo da peça e o robô secundário realiza a apreensão o outro extremo do corpo da peça.

Uchiyama e Yamashita (1991) identificaram os vetores que descrevem a tarefa cooperativa de dois robôs, apresentando as condições de força, posição e contato com equações lineares através de um algoritmo.

Num outro sistema utilizando dois robôs cooperativos Il Kim e Zheng (1991) estudaram as duas fases do processo de distribuição de carga de um corpo sólido: na primeira fase a carga transportada pela ferramenta terminal do robô é estimada a partir da utilização de sensores enquanto que numa segunda fase é compensada a força onde um robô principal eleva a carga, através da utilização de um robô secundário. Zuofeng, Tzyh-Jong e Bejczy (1991), descrevem a solução do problema de calcular o torque / força máxima de um sistema cooperativo de robôs, e a partir destas informações foram planejadas as tarefas cooperativas utilizando o controle de torque / força.

Stokic, Vukobratovic e Lekovic (1991), desenvolveram um software para simulação de FMC. Através deste software configuraram uma célula de manufatura flexível com dois robôs, três máquinas ferramentas CNC (CNC Computer Numerical Control), dois sistemas de armazenamento, duas esteiras elétricas e vários tipos de sensores.

O Sistema de supervisão e controle foi projetado utilizando Redes de Petri (PN – Petri Nets). Através da modelagem da cinemática dos robôs foi definida a configuração da célula, e a partir do modelo dinâmico foram otimizadas as velocidades de operação e sincronização, e finalmente utilizando as leis de controle dinâmico e compensações nas juntas foram otimizadas conforme a precisão das trajetórias e tempos de ciclo.

Quando um objeto apresenta grandes dimensões e pesos superiores aos suportados pela garra de um robô, exige-se o desenvolvimento de novas ferramentas de preensão. Vigia *et al.* (1991), estudaram este caso sobre múltiplos braços robôs cooperativos interagindo no mesmo espaço de trabalho. Encontraram que a interação entre o corpo rígido e os robôs se caracteriza por limitações unilaterais e configuram uma cadeia cinemática fechada. As limitações na operação e as equações não lineares do sistema transformam sua abordagem num problema complexo. Finalmente, desenvolveram as equações dinâmicas com limitações unilaterais no ponto de contato e projetaram um sistema de controle de força em malha fechada.

Quando dois robôs movem o mesmo objeto cooperativamente, eles integram uma cadeia cinemática fechada; isto exige o controle de posição, da velocidade e da força. Tao e Luh (1991), consideraram o comportamento dinâmico do sistema, implementando para esta abordagem um sistema de controle adaptativo, a partir da utilização de duas estratégias de controle: PMRC (PMRC Position Model Referenced Control) e FMRC (FMRC Force Model Referenced Control) onde o PMRC apresentou uma melhor estabilidade dinâmica.

Xiaoping (1991) analisa a modelagem dinâmica e controle de dois robôs cooperativos para a situação onde os mesmos interagem entre si, empurrando um sólido. O controle de força interativo impede que o objeto deslize e se deteriore através da pressão. O problema de controle se formula no espaço de estado e o controle da coordenação dos robôs utiliza uma malha fechada de controle, para separar os sistemas de controle de força e movimento.

Tarefas de robôs cooperativos são muito complexas, aumentando ainda mais esta complexidade quando as tarefas são definidas a partir da programação *off-line*. Chi - Keng (1991) reporta esta complexidade na programação *off-line* de robôs duais para três tipos de movimentos: simultâneo, coordenados e independente.

Li - Chung e Ming (1994), desenvolveram uma aproximação sistemática para a formulação virtual de limitações no deslocamento e equações dinâmicas para cooperação entre robôs múltiplos, unificando um método para a dinâmica inversa e a capacidade de carga dinâmica. Este método considera conhecidas às coordenadas do centro de massa da pré-carga e do elemento terminal do robô. A partir da utilização do princípio de D'Alembert são definidas toda a dinâmica do corpo rígido.

A programação *off-line* de um robô que opera em uma célula para enrolamento de cabos, foi desenvolvida por Scholliers e Van Brussel (1996) utilizando a seguinte abordagem: desenvolvimento da modelagem geométrica do robô, movimentação do robô dentro de seu espaço de trabalho, enrolamento do cabo, planejamento da trajetória do robô, simulação e controle do robô.

Jiank, K. *et al.* (1997) apresentam, uma proposta para planejamento de seqüências de montagem, utilizando uma célula de trabalho constituída por dois robôs, através da otimização dos tempos.

Rehg (2000) apresenta critérios práticos e descreve a integração de robôs industriais em FMS, mostrando a necessidade de desenvolvimento de hardware e software dedicados para a implementação de células de trabalho automatizado orientadas aos CIM.

A programação das FMCs não é fácil e atualmente é um área de pesquisa e desenvolvimento. Pires e Sá da Costa (2003) apresentam uma programação orientada aos objetos. Foi aplicada ao robô IRB 1400/S4 individualmente e depois o robô foi implantado na FMC para o desenvolvimento das trajetórias na indústria automotiva.

A comunicação do usuário com o robô acontece na forma remota, usando comandos especialmente criados para desenvolver as funções do robô. A FMC foi configurada com o robô IRB 1400 no centro e três esteiras elétricas: O robô desenvolve tarefas de montagem, pega uma peça de vidro da primeira esteira elétrica e a posiciona no alojamento de um painel da terceira esteira elétrica, logo pega uma segunda peça da segunda esteira elétrica e posiciona sobre alojamento de outro painel da terceira esteira elétrica.

## **2. 5 - Metodologias para o Projeto e a Modelagem de FMS**

Askin e Strandridge (1993) destacam ao nível estratégico, o critério econômico para desenvolver o projeto de FMS's, propondo a seguinte seqüência para sua implementação:

- 1) Estabelecer objetivos estratégicos e econômicos,
- 2) Criar um sistema de capacitação para os trabalhadores,
- 3) Selecionar máquinas e definir estratégias para o sistema de controle,
- 4) Planejar as peças para processamento, com base na capacidade do sistema,
- 5) Projetar a configuração do FMS.

O sistema de movimentação de materiais MHS estabelece as bases para a configuração do FMS. Groover (1996) (2000) diferencia cinco tipos de configurações:

- 1) Em linha,

- 2) Em malha fechada,
- 3) Em escada,
- 4) Aberta,
- 5) FMC.

Os MHS baseados em esteiras elétricas enquadram-se nos primeiros três tipos, os MHS baseados em AGVs (AGV Automatically Guided Vehicle) enquadram-se nos primeiros quatro tipos.

A modelagem gráfica e a simulação computadorizada são as metodologias mais usuais no projeto de FMS. Outras técnicas empregadas são: Programação matemática, Redes neurais, Métodos heurísticos, Arranjos em grupo, Teoria de aproximação de controle, Inteligência artificial, Sistemas especialistas, Redes de Petri, Carvajal (2000).

Nos sistemas de manufatura tradicionais numerosas técnicas podem ser aplicadas para a configuração do sistema produtivo. Muther (1973) apresentou uma distribuição em planta sistemática SLP (SLP Systematic Layout Plant) como uma aproximação a um projeto manual. A aproximação ao problema de assinação quadrático QAP (QAP Quadratic Assigment Problem) calcula os custos mínimos de N departamentos a M locações. Para encontrar uma solução ótima ou perto da ótima, se desenvolveram as seguintes metodologias: problema de assinação linear LAP (LAP Linear Approximation Problem), gráficos da planta PG (PG Plant Graphs), maximização de gráficos de planta MPG (MPG Maximum Plant Graphs), e maximização por seu peso de gráficos de planta MWPG (MWPG Maximum Weigh Plant Graphs).

O projeto de FMS baseado na tecnologia de grupo GP (GT Group Technology) pode desenvolver-se nas seguintes etapas:

- 1) Seleção da família de peças,
- 2) Determinação de seu processamento,
- 3) Seleção das máquinas,
- 4) Seleção dos componentes do MHS,

- 5) Configuração final do FMS,
- 6) Definição do sistema de controle e sistema operativo.

Aly e Subramaniam (1993), Dekleva e Gaber (1994) e Tezlaff (1995) descrevem metodologias para a seleção de máquinas de processamentos usando diretamente a modelagem matemática e algoritmos computacionais.

Mellichamp et al (1990) descrevem uma metodologia para a seleção das máquinas de processamento através da utilização de um sistema especialista ES (ES Expert System). Egbelu (1993) propõe uma metodologia para configurar FMS baseada nos AGVs como sistema de transporte. Mahadevan e Narendran (1994) propõem a modelagem híbrida, através de processo gráfico - matemático, para projeto de um MHS baseado em AGVs para um FMS.

Segundo Chao e Wang (1995), as PN são técnicas de modelagem gráfica e matemática de sistemas concorrentes, assíncronos, distribuídos, paralelos, estocásticos; e propõem um sistema CAD, baseado na síntese das PN, permitindo a criação e simulação de um FMS.

Jeng (1995) propõe uma PN modular para a modelagem de FMS baseado na sincronização do controle do processo de manufatura, ou seja, sincronização entre as máquinas e o robô.

Rehnhart e Köhne (1996) propõem uma metodologia para o projeto de sistemas de manufatura autônomos e FMS, baseado na demanda e capacidade de funcionamento. Mertins e Wieneke - Toutaui (1991), apresentam uma metodologia computadorizada para a modelagem do funcionamento e topologia dos FMS.

Kaku e Rachamadugu (1992), baseados em duas das configurações mais utilizadas em FMS, em malha fechada e linear, e desenvolvem um modelo matemático. Das (1993) apresenta uma metodologia heurística para a configuração de um FMS.

A modelagem através de sistemas CAD e a simulação virtual são as ferramentas mais utilizadas no projeto de FMC complexos. O arranjo dos componentes para a configuração da FMC é a fase do projeto a ser seguida, após a definição dos equipamentos. A variável que lidera o projeto é a minimização dos custos de manufatura, seguida de uma maximização da utilização U (U Utilization) dos equipamentos.

A GT é fundamental para o projeto e implementação de FMC em forma modular. O projeto e a modelagem de FMS podem se desenvolver quando à estratégia da modernização e automação da indústria é um imperativo para competir nos mercados internacionais, Carvajal (2000) (2001).

## **2. 6 – Detecção e Tratamento de Colisões**

A detecção e tratamento de colisões são importantes para obtermos uma programação *off-line* eficiente. Um dos primeiros pesquisadores que analisa a detecção de obstáculos foi Lozano - Perez (1981) que mapeou os obstáculos no espaço da juntas para evitar colisões durante a realização de trajetórias.

Brooks (1983) considera o problema de detecção de obstáculos, apresentando o conceito de caminho livre, no espaço cartesiano e no espaço das juntas, onde o sistema é habilitado para que o robô possa atuar, evitando possíveis colisões, em ambientes perigosos.

Angeles (1997) propõe trajetórias com polinômios de alta ordem que definem os sistemas de coordenadas intermediárias no espaço cartesiano. Essas coordenadas conectam o ponto inicial com o ponto final e são importantes na evasão de obstáculos. Este método utiliza a cinemática inversa do robô.

No ambiente de modelagem podem ser utilizadas várias técnicas de detecção de colisões. Stobar (1987) descreve técnicas de detecção de colisões com base no modelador

BUILD™. Neste caso as instalações do robô têm que ficar otimizadas para melhorarmos a sua precisão.

Na programação *off-line* do robô o programador precisa fornecer as informações detalhadas sobre o movimento dos objetos no espaço de trabalho da FMC. As colisões são exibidas em forma gráfica num computador, e usualmente acontecem por desvio do robô de sua trajetória de trabalho previamente planejada. As colisões podem ser facilmente identificadas utilizando técnicas de realidade virtual, entretanto no mundo real, os efeitos dinâmicos inerentes às características não lineares de um robô contribuem aleatoriamente na sua movimentação.

Cameron (1984) propõe duas técnicas de detecção de colisões com base no modelador de sólidos com primitivas:

- 1) Através da tomada de uma seqüência de posições sucessivas utilizando o modelo do robô em 3D, com intervalos de tempo regulares para um movimento específico, se analisa sua interferência com os objetos ao seu redor, verificando que não ocupem o mesmo volume no mesmo instante de tempo.

- 2) Considera-se o tempo como uma quarta dimensão no movimento do robô, conseqüentemente, as colisões são detectadas quando se verifica interferência entre o volume do objeto e o volume de trabalho do robô.

Cameron (1984) menciona um terceiro método para detecção de colisões que foi desenvolvido por Martín (1985), onde o volume ocupado pelo robô em movimento é computado e simultaneamente são verificados a existência de interferência com o volume do objeto estático.

Muller (2004) propõe-se uma metodologia para resolver o problema da cinemática inversa de robôs seriais redundantes, considerando o tratamento das colisões, baseado num algoritmo que anuncia a perturbação do obstáculo e corrige a trajetória do robô. Este algoritmo pode ser aplicado na programação *off-line* do robô e na sua implementação em

tempo real. Sua aplicação pode ser estendida aos robôs paralelos. Uma especial atenção deverá ser dada ao desenvolvimento do ambiente do robô com sistema CAD.

## 2.7 - Ensino

O ensino de robótica através de uma FMC real envolve altos custos e em muitos casos condições extremas de segurança, que podem ser evitados utilizando técnicas de VR.

Li, Ho e Li, (2000) desenvolveram um ambiente de simulação de realidade virtual, onde os estudantes praticam, exploram e predizem as situações perigosas. Construir células robotizadas virtuais pode facilitar o estudo do comportamento do robô em funções de manufatura e montagem. Neste trabalho os autores incluem interação dinâmica entre os componentes da realidade virtual e simulação. Este projeto foi desenvolvido para o ensino de robótica.

Tzafestas et al (2004) projetaram, estruturaram e desenvolveram um laboratório remoto virtual para ensino de robôs usando tecnologia Java como interfase, incorporando:

- 1) representação gráfica das configurações dos robôs em 2D,
- 2) vídeo em tempo real para exibição da movimentação dos robôs,
- 3) interfase interativa entre o usuário e a estação computadorizada, com todas as funções para programação *off-line* do robô.

O usuário pode criar programação *off-line* para o robô e modificar-lo à vontade. Este laboratório virtual está baseado na arquitetura tele robótica cliente – servidor, com varias estações computadorizadas integradas numa rede LAN (LAN Local Área Network) ou pela internet, contribuindo ao usuário diretamente na tele operação e indiretamente na tele programação. Os robôs usados na experimentação são SCARA e Adepth I e linguagem de programação V+

Rawat e massiha (2004) e Meng et al (2004) trabalharam no desenvolvimento e implementação de laboratórios para o ensino e pesquisa em robótica e automação. Criaram laboratórios com todas as ferramentas necessárias do projeto e da simulação de células robotizadas, para estudantes de graduação e de pós-graduação em diferentes áreas de engenharia e administração.

## **2. 8. A Robótica na UNICAMP**

No Brasil, a Universidade Estadual de Campinas tem mais de uma década de desenvolvimento de pesquisas em robótica. Podem identificar-se três orientações fundamentais:

- a) Desenvolvimento de configurações e ferramentas do elemento terminal.
- b) Desenvolvimento de sistemas de sensoriamento, controle e supervisão.
- c) Estudos das trajetórias, modelagem, simulação, programação e calibração.

1) Em configurações, Madrid (1988) desenvolveram a estrutura de um robô manipulador mecânico TRRR para posicionamento espacial com controle digital hierárquico a microprocessadores, Ohfugi (1991) apresentou garras articuladas para robôs manipuladores, com análise da cinemateca e de forças, e também desenvolveu o sistema de sensoriamento e controle de posição e esforços na pressão de objetos, Fernandez (1997) desenvolve um robô manipulador, baseada na estrutura do paralelogramo, e em dispositivos do sistema de movimentação, Da Silva (1998) contribui com a seleção de atuadores e acionamentos integrados à modelagem de robôs industriais.

2) Em sistemas de controle para robôs tem-se variadas alternativas. Miranda (1992) apresenta um sistema de controle para um servomecanismo baseado em micro computador dedicado, Fayan (1992) desenvolve um estudo e especificação de um supervisor de controle para um robô industrial, Souza (1992) cria um procedimento automático para aquisição e tratamento do movimento de um robô, Dias (1993) implementa experimentalmente um supervisor de controle para robôs industriais, Coutinho (1993) gera um ambiente integrado de desenvolvimento de software para

robótica, Takita (1995) apresenta um sistema de sensoriamento de robôs industriais no espaço das juntas e no espaço cartesiano, Nogueira (1995) projeta um sistema de controle de posição e orientação de manipuladores através de mouse especial, Vieira (1996) desenvolve e implementa um sistema de controle de trajetória contínua para um robô industrial de alto desempenho, Cerqueira (1996) analisa uma classe de neurônios artificiais para aplicações em sistemas robóticas, Hermini (1996) contribui ao estado de não linearidades na modelagem e controle de manipuladores, Hervella (1997) implementa controle de manipuladores e robôs industriais, baseado nos controladores lógicos programáveis; Nobre (1997) contribui ao projeto e análise de controladores nebulosos e sua aplicação para controle de juntas robóticas, Sa (2000) desenvolve e implementa um programa computacional para supervisão e controle de manipuladores robóticos, Cerqueira (2001) identifica sistemas dinâmicos para aplicações em robótica usando redes neuronais, Souza (2001) implementa algoritmos preditivos para controle de juntas robóticas, Cosso (2002) implementa um sistema de supervisão e controle num sistema tele operado para robótica, Oliveira (2003) propõe um sistema de controle didático baseado na modelagem dinâmica de robôs industriais e Pimenta (2003) implementa um sistema de controle de juntas robóticas usando controlador preditivo generalizado adaptativo direto.

(3) A terceira orientação apresenta diversificação da pesquisa em robótica, Souza (1992) aplica um procedimento automático para aquisição e tratamento da movimentação de um robô, Cruz (1993) projeta e desenvolve um sistema de geração automática de trajetórias para manipuladores, Campos (1993) implementa um algoritmo para estimativa de parâmetros do modelo cinemático de robôs, Saramago (1993) projeta e desenvolve um sistema de calibração e medida de precisão para robôs industriais, Madrid (1994) desenvolve controle de trajetórias contínuas por seccionamento em sub trajetórias, usando inteligência artificial num robô multi-tarefas, David (1996) desenvolve modelagem, simulação e controle de robôs flexíveis e Sa (1996) implementa métodos numéricos para a resolução do problema cinemático inverso de robôs com ênfase em controle de posição.

## 2. 9. Contribuições desta Pesquisa

Existem diversos sistemas de simulação e programação *off-line* de robôs, mas poucas aplicações com metodologias bem definidas. As aplicações mais significativas são direcionadas as aplicações de:

- 1) soldagem na industria aeroespacial,
- 2) pintura pulverizada na industria automobilística,
- 3) soldagem por pontos na industria automotiva,
- 4) corte de vidro,
- 5) inserção de componentes eletrônicos,
- 6) manutenção do casco dos aviões de grande porte e
- 7) prototipagem rápida.

Os sistemas de simulação e programação *off-line* mais conhecidos são: GRASP, CimStation, IGRIP, Workspace, RobotStudio, EasyRob, Cosimir e todos têm em comum três ferramentas:

- 1) Visualização e a modelagem gráfica da célula de trabalho em sistema CAD.
- 2) Planejamento e simulação das trajetórias dos robôs em sistema CAD.
- 3) Tratamento de colisões.

Entretanto não existe uma metodologia geral integrada direcionada ao desenvolvimento das FMC que possa ser implementada com flexibilidade, a baixo custo e em mínimo tempo.

Para cada robô industrial precisa do desenvolvimento de um algoritmo para a modelagem da cinemática e este algoritmo integrar-se como modulo adicional ao sistema de simulação e programação *off-line*. Este algoritmo da modelagem cinemática é conhecido pelo nome de simulação realista do robô (RRS Robot Realistic Simulation).

Este trabalho de pesquisa apresenta as seguintes contribuições:

1) Proposta de uma metodologia geral de simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados em FMC, baseada em três tarefas:

- a) a modelagem do robô, do seu ambiente e da sua cinemática
- b) planejamento de trajetórias e simulação da movimentação dos robôs e mecanismos, e
- c) programação *off-line* dos robôs e mecanismos.

A Figura 1.1 mostra a arquitetura da metodologia proposta da simulação e programação *off-line* de robôs integrados em FMC.

2) Definem-se quatro níveis de complexidade na programação *off-line*:

Nível 1. Programação *off-line* independente: para um robô

Nível 2. Programação *off-line* simultânea: para dois robôs com movimentação simultânea.

Nível 3. Programação *off-line* seqüencial: para dois robôs com movimentação em série, enviando e recebendo sinais digitais sincronizados.

Nível 4. Programação *off-line* integrada: para um ou dois robôs com movimentação integrada com a movimentação dos mecanismos, enviando e recebendo sinais digitais sincronizados. A Figura 1.2 mostra os níveis de complexidade na programação *off-line* dos robôs industriais.

(3) Aplica-se a metodologia desenvolvida e os níveis de programação *off-line* para validação através de cinco exemplos práticos, desenvolvidos com o software Workspace no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP:

Exemplo 1: FMC com dois robôs de soldagem e mecanismo MB - MR.

Exemplo 2: FMC com um robô de soldagem, um robô de posicionamento e mecanismo de rotação.

Exemplo 3: FAC com dois robôs.

Exemplo 4: FMC com um robô de posicionamento e duas máquinas CNC: Três variantes.

Exemplo 5: FMC com um robô de posicionamento e cinco máquinas CNC.

(4) Configura-se uma FMC de quinze graus de liberdade proposta para ensino de robôs na UNICAMP, com a metodologia proposta da simulação e programação *off-line* de robôs e mecanismos. Também, apresentam-se as modelagens da FMC em PN para validação da:

- a) Programação *off-line* independente.
- b) Programação *off-line* simultânea.
- c) Programação *off-line* seqüencial.
- d) Programação *off-line* integrada.

Na Figura 1.3. mostra-se a FMC de 15 gdl a implementar-se na UNICAMP com o robô IRB 1400, o robô IRB 1500 e mecanismo MR + MB.

(5) Desenvolve-se uma metodologia para a obter o sistema de coordenadas com Quaternions das juntas de um robô serial de seis graus de liberdade.

(6) Desenvolve-se uma metodologia e equação para obter a modelagem da posição em Quaternions para a garra ou ferramenta de um robô serial de seis graus de liberdade.

(7) Desenvolve-se uma metodologia e equação para obter a modelagem da rotação em Quaternions da garra ou ferramenta de um robô serial de seis graus de liberdade.

(8) Desenvolve-se uma metodologia para obter a modelagem da cinemática com Quaternions para o robô serial IRB 1400 de ABB de seis graus de liberdade.

(9) Desenvolve-se a cinemática direta e inversa de um robô industrial serial 6R baseada na sistemática de Denavit – Hartenberg e matrizes de transformação homogêneas.

(10) Desenvolve-se uma metodologia para obter a modelagem da cinemática diferencial direta e inversa baseado nas MTH para o robô serial IRB 1400 de ABB de seis graus de liberdade.

(11) Desenvolve-se uma metodologia do projeto mecatrônico de robôs industriais baseada na integração sinérgica e concorrente de três tarefas fundamentais:

a) Modelagem da configuração em ambiente CAD, modelagem da cinemática, modelagem da dinâmica, e modelagem do sistema de controle.

b) Simulação da movimentação do robô e simulação da estratégia de controle.

c) Integração sinérgica e concorrente de componentes mecânicos, eletrônicos e computadorizados para configurar o novo robô.

(12) A metodologia apresentada nesse projeto de pesquisa é uma ferramenta direcionada aos sistemas de manufatura de pequeno e médio porte, e pode transformar-se em FMC ou para que se possam configurar novas FMC a partir de equipamento disponível no mercado, permitindo que as indústrias possam competir a nível internacional, com flexibilidade, com alto nível de produtividade, e com qualidade na produção.

## Capítulo 3

### Modelagem CAD de Robôs e de Dispositivos Mecatrônicos

A Mecatrônica, segundo Shetty e Kilk (1997), pode ser definida como a integração de conceitos da Engenharia Mecânica, Elétrica e Eletrônica baseada em controle através do computador para o projeto e manufatura de produtos e processos, enquanto a Engenharia Concorrente (CE), segundo Yazdani (1996), pode ser definida como a integração entre concorrência, super posicionamento do projeto e desenvolvimento de produtos e seu Processo de Manufatura.

A metodologia para o desenvolvimento de projetos de sistemas mecatrônicos é multi-disciplinar e não sequencial baseada na concorrência das disciplinas de engenharia, procurando uma aproximação ao projeto de produtos mais sinérgicos. O objetivo da CE é reduzir os tempos na implementação do projeto desenho e manufatura por superposição e integração das atividades.

Neste capítulo são apresentados conceitos que fundamentam e descrevem metodologias encaminhadas ao desenvolvimento da modelagem com sistemas CAD e PN de FMC. A Figura 3.1 é apresentada a integração da mecatrônica e CE para configurar FMC para o desenvolvimento da modelagem, simulação e integração para programação *off-line* de todos os mecanismos. Os requisitos preliminares são aqueles que permitem dar início ao desenvolvimento da FMC a partir de uma de duas opções: Conversão de uma

estação de processamento de um sistema produtivo em FMC ou a criação de uma nova FMC a para um novo produto.

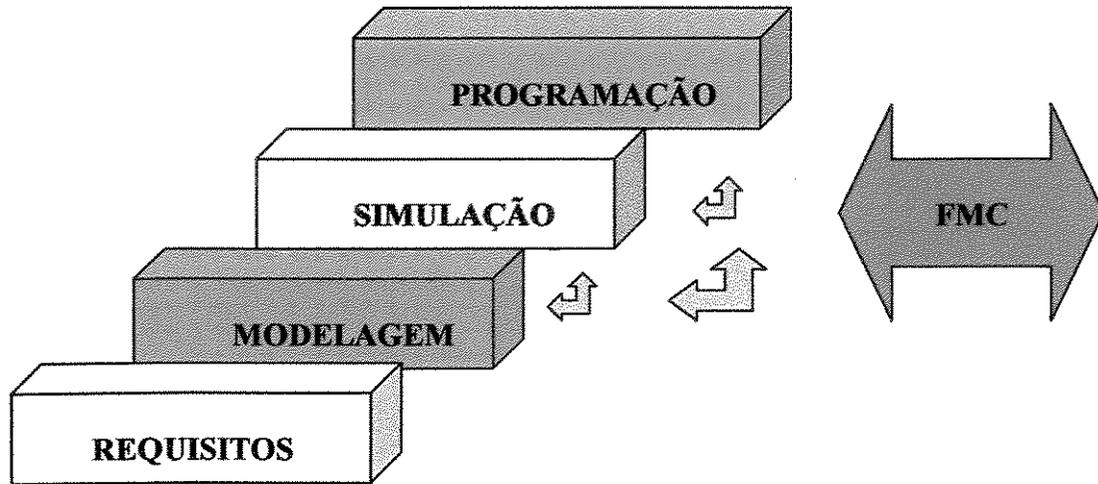


Figura 3.1: Mecatrônica e CE em FMC.

CIMOSA significa arquitetura aberta para sistema CIM (Open System Architecture for CIM). É uma arquitetura para definição, especificação e implementação de sistemas CIM. Seu objetivo é oferecer uma estrutura conceitual para analisar os requisitos de uma empresa e convertê-los num sistema que habilite e integre as funções que satisfaçam esses requisitos. Esta estrutura, aliada a uma infra-estrutura computacional para integração, possibilita descrever funcionalmente uma empresa como um conjunto enumerável de processos executados por entidades funcionais num ambiente distribuído, interagindo e compartilhando objetos físicos e objetos de informação, por intermédio de troca de eventos e de vistas de objetos. A Figura 3.2. mostra a integração mecatrônica dos componentes para configurar FMC apoiada pela integração entre as engenharias que também podem ser base para o desenvolvimento dos CIM baseados na metodologia CIMOSA.

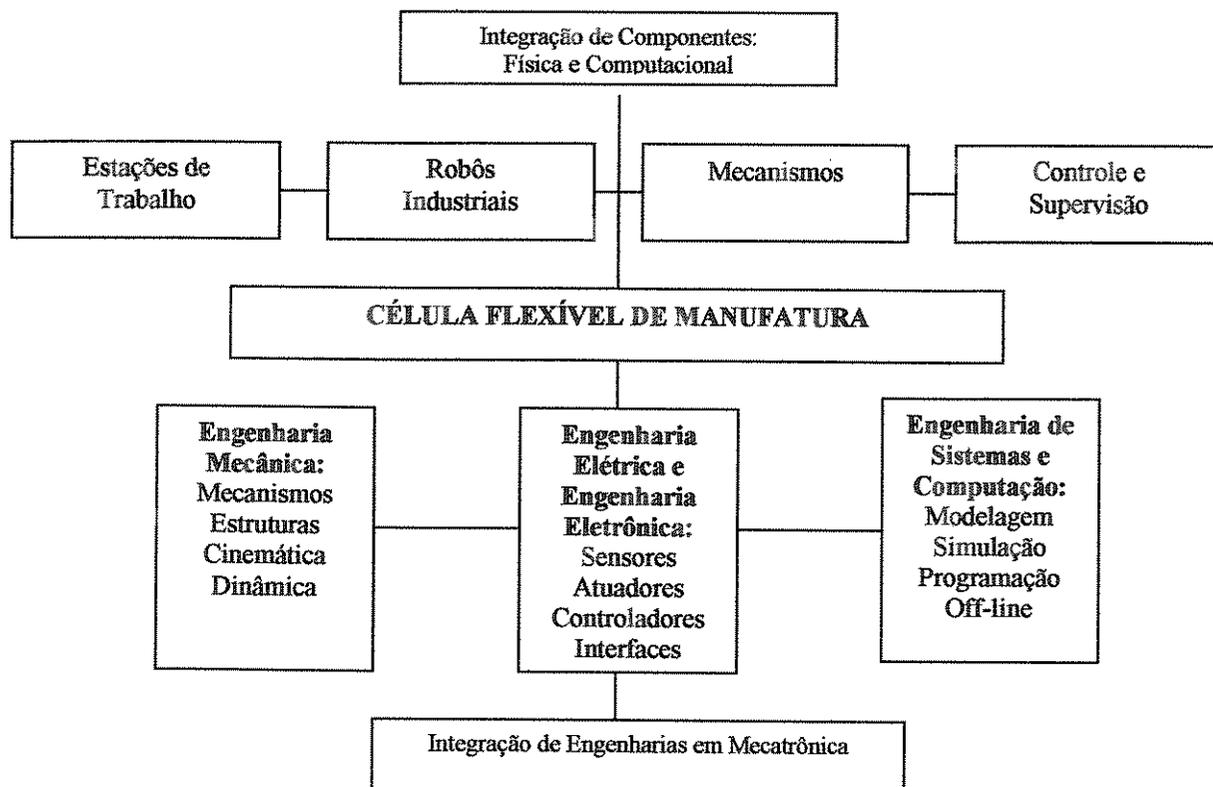


Figura 3.2: Arquitetura aberta de integração mecatrônica para o projeto de FMCs direcionados aos CIM

Segundo Pereira (2001) a metodologia CIMOSA tem como base quatro princípios e esses princípios visam tornar os aplicativos independentes das plataformas nas quais são executados, em face da redução dos ciclos de vida dos sistemas e a dificuldade de reutilização de seus aplicativos para a geração de novos subsistemas. São eles:

1) Isolamento entre aplicativos e a infra-estrutura de suporte. Segundo esse princípio o aplicativo não depende dos formatos dos dados, dos meios de acesso, de armazenamento e de transmissão presentes na infra-estrutura na qual é executado. Este isolamento é recomendado pela arquitetura CIMOSA visando a migração direta dos aplicativos de uma plataforma para outra; as entidades funcionais dessa plataforma são os seus aplicativos.

2) Isolamento entre dados e aplicativos. Esse princípio considera que um aplicativo nunca possui dados e que esses estariam numa base capaz de ser modificada sem requer alteração nos aplicativos. Essa base possibilitaria aos aplicativos operem nas mesmas estruturas de dados.

3) Isolamento entre funções e comportamento. Significa separar as funções elementares de um sistema daquelas que definem o comportamento, expresso como procedimento do processo, e criando novos aplicativos, sem mudar a maioria de seus componentes.

4) Conceito de generalidade. Enfatiza que qualquer objeto pode ser projetado como uma instância de um objeto mais genérico existente em modelos públicos ou privados. Os primeiros são supostamente comercialmente disponíveis, enquanto os privados são de propriedade de empresas. Essa instanciação propicia o pleno aproveitamento do que já foi modelado e com isso torna mais fácil a construção do modelo que através da derivação de um arquétipo, já que neste caso necessita-se descrever as especificações do objeto sob considerado.

### **3. 1. Conceitos Básicos Direcionados a FMC**

#### **3.1.1. Sistemas Automáticos de Produção SAP**

O processo de automatização industrial pode ser dividido em diferentes níveis de complexidade, conforme ilustra a figura 3.3.

O primeiro nível, considerado o mais simples, é o de dispositivo, sendo constituído de atuadores e sensores, que são integrados em malhas de controle individual para cada variável.

O segundo nível integra os sistemas de controle ao nível de dispositivo, para configurar uma máquina, como uma máquina CNC, um robô, um AGV, ou uma esteira.

Neste nível também se controla a seqüência de um processo ou a seqüência de operações de uma máquina.

O terceiro nível integra as equipes do segundo nível de automatização, como máquinas CNC, com robôs e diferentes mecanismos que configuram uma FMC. Integrando-se mais de dois FMC se configura um FMS. Neste nível se controla a coordenação de operação entre as equipes integradas.

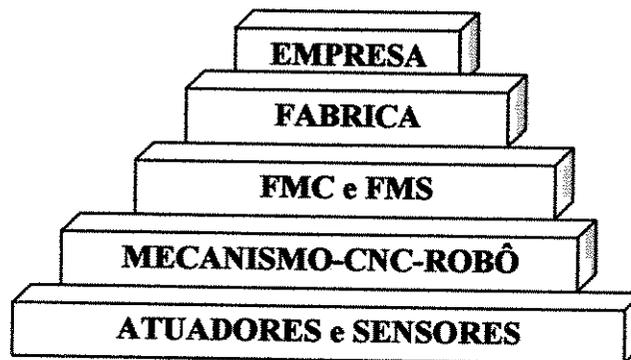


Figura 3.3: Níveis de Automação Industrial.

O quarto nível desenvolve-se na fábrica com instruções da administração para criar planos operacionais de produção, como processamento de pedidos, planejamento de processos, controle de inventários, planejamento de requerimento de materiais, controle na planta e controle de qualidade.

O quinto nível envolve toda a empresa, constituindo o sistema de informação com todas as funções ao nível de corporação: mercado, vendas, contabilidade, desenho, pesquisa, planejamento estratégico e planejamento geral da produção. A Figura 3.3 ilustra os conceitos descritos.

Os FMS têm como principal finalidade a transformação de matérias primas em produtos finais acabados e as atividades fundamentais para seu desenvolvimento, conforme mostra a figura 3.4 são:

- 1) Operações de processamento e montagem,

- 2) Movimentação de materiais,
- 3) Controle de qualidade e,
- 4) Supervisão e controle.

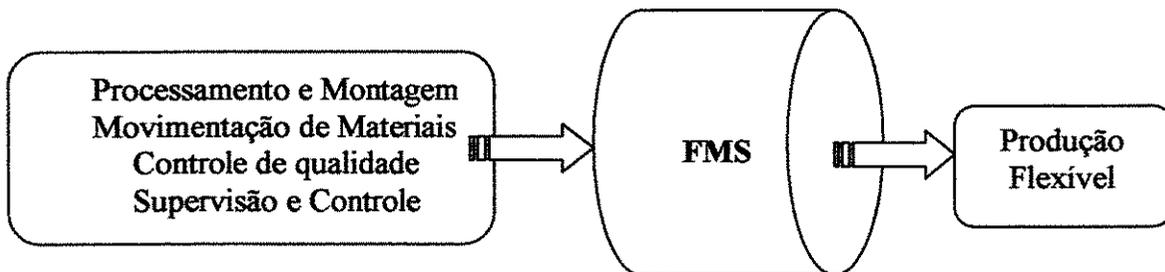


Figura 3.4: Funções nos FMS.

As *operações de processamento* usam trabalho e energia para transformar matéria prima em peças com valor agregado. Em geral uma determinada peça precisa de mais de uma operação de processamento, elas podem ser classificadas de acordo com o tipo de operação realizada, tais como:

- 1) Para dar forma usando energia mecânica ou térmica ou a combinação entre elas,
- 2) Para melhoramento das propriedades mecânicas ou químicas do material e,
- 3) Processamento da superfície.

As *operações da montagem* juntam duas ou mais peças para configurar uma nova peça com forma permanente o semi-permanente. O *sistema de movimentação* de materiais move as peças entre estações de processamento e / ou montagem, e precisa ser eficiente para reduzir tempos de espera das peças. O robô é um sistema eficiente de movimentação de peças entre estações de trabalho. A *inspeção e validação* são atividades de controle de qualidade que definem se o produto se ajusta a especificações de qualidade e funcionalidade. A *supervisão* e controle na manufatura permitem a regulagem das operações de processamento e montagem.

A alteração dos SAP para os FMS implica em três tarefas fundamentais: incorporação dos robôs industriais, integração física dos componentes, sistema de controle integrado e flexibilidade.

### **3.1.2 Tecnologia de Grupo (GT) e Manufatura Celular (CM)**

Segundo Sudhakarapeian, Jayakumar, e Sornakumar (2000) a Tecnologia de Grupo (GT) é uma estratégia da manufatura para identificar e explorar economicamente as semelhanças e similitudes geométricas das peças e processos, para otimizar a produção na fábrica. Peças similares são agrupadas em famílias de peças para configurar equipes de produção em grupos ou células de máquinas. Cada uma das células está dedicada a uma família de peças. As células de manufatura baseadas na GT podem ser classificadas assim:

- 1) Célula de máquinas simples,
- 2) Célula de máquinas em grupo com operação manual,
- 3) Célula de máquinas em grupo com operação semi-automática e,
- 4) FMS ou FMC.

O desenvolvimento da GP implica: formação de um número mínimo de células de processamento para criar o fluxo de produção, definição do número de células e seu tamanho, e otimização da utilização das máquinas. O caráter especial da CM é integrar o equipamento em grupos que operam independentemente com base em regras pré-estabelecidas, para obter os seguintes benefícios:

- 1) Redução do tempo de preparação do equipamento,
- 2) Redução do trabalho em processo WIP (WIP Work In Process),
- 3) Grande flexibilidade,
- 4) Maior aproveitamento do espaço de trabalho na fábrica,
- 5) Melhor qualidade na produção e,
- 6) Planejamento da produção simplificado (Scheduling).

### **3.1.3 Flexibilidade dos Sistemas de Manufatura**

A flexibilidade em um sistema de produção é sua habilidade para identificar e distinguir entre as peças a serem processadas no sistema, com mudança rápida de software e de hardware.

Para desenvolver o conceito de flexibilidade em um sistema de manufatura automatizado, por exemplo, pode-se considerar uma célula com duas máquinas CNC que são carregadas e descarregadas por um robô industrial a partir de uma esteira elétrica. A célula opera sem a assistência humana por longo tempo.

Periodicamente, um trabalhador descarrega as peças processadas situadas na esteira elétrica e são substituídas por novas peças. Por definição, esta é uma célula de manufatura automatizada, e para que seja uma FMC ela tem que cumprir quatro critérios, que representam condições aceitáveis de flexibilidade:

- 1) Existência de uma grande variedade de peças,
- 2) Possibilidade de alteração do ritmo e estratégia de produção,
- 3) Calibração mecânica do sistema e,
- 4) Possibilidades de ser realizada uma nova peça.

A flexibilidade para a indústria significa disposição de mudar as novas condições do mercado internacional. Ao mesmo tempo, significa flexibilizar a produção, automatização dos processos, melhoramento da qualidade da produção, confiabilidade no processo de produção e encontrar equilíbrio entre a produtividade e flexibilidade. Do ponto de vista dos negócios, a flexibilidade significa a habilidade de satisfazer as necessidades do mercado, sem muito custo, sem muito esforço, bom atendimento e flexibilidade na organização empresarial.

A necessidade de flexibilizar uma indústria de manufatura tem sua origem nos clientes. Dentro de um ambiente competitivo os clientes demandam produtos personalizados, com mudanças rápidas no desenho, linhas de produção adaptáveis para

uma variedade de produtos, tamanhos dos lotes de produção medianos, níveis múltiplos de qualidade, entrega dos pedidos a tempo e variedade nos níveis dos preços; que são elementos externos da flexibilidade na indústria.

Os elementos internos da flexibilidade são as funções e operações na fabricação dos produtos, como, flexibilidade nas seqüências, na rota das seqüências, no processo e as tecnologias envolvidas, na manipulação das materiais e peças, no produto, no plano de produção, no controle de manufatura, no volume da produção e na produção.

A automatização flexível é uma extensão da automatização programável. Um sistema automatizado flexível pode produzir uma variedade de peças em curtos períodos de tempo e desenvolver vários programas e planos de produção simultaneamente. Este tipo de automatização é apropriado para lotes de produção de médio tamanho. As FMC são as unidades fundamentais da automatização flexível e ocupam um lugar intermediário na automatização industrial.

#### **3.1.4. FMS**

O controle computadorizado central de todos os componentes é uma característica comum dos FMS. Maccarthy e Liu (1993) definem um FMS como um sistema de produção que pode produzir uma variedade de tipos de peças e consiste de máquinas NC ou CNC, conectadas por um sistema automatizado de movimentação de materiais AMHS (AMHS Automatically MHS).

As operações são controladas através de um computador central. A partir desta definição geral se identificam quatro tipos FMS:

1) Uma máquina flexível simples (SFM Single Manufacturing Machine) NC ou CNC com capacidade automatizada para trocar ferramentas e peças e capacidade de armazenagem.

2) Uma FMC, configurada com um grupo de dois ou três SFM, com mecanismo comum de movimentação de materiais, que pode ser um robô centralizado.

3) Um sistema de multi máquina de sistema de manufatura flexível MMFMS, configurado por varias SFM e conectadas por um AMHS que pode ser uma frota AGVs.

4) Uma multi-célula de sistema de manufatura flexível MCFMS, configurada por varias FMC e / ou vários SFM, se for necessário.

Para Dekleva e Gaber (1994), os FMS são uma transformação gradual dos sistemas de produção celular CPS (CPS Cellular Production System). A diferença entre CPS e FMS é que os primeiros são FMS operados manualmente e os segundos são totalmente automatizados.

Reinhart e Khone (1996), trabalharam com sistemas de manufatura autônomos MAS (AMS Autonomous Manufacturing Systems), configurados por várias células de manufatura autônomas AMC (AMC Autonomous Manufacturing Cells). Enquanto todos os elementos constituintes de um FMS são integrados através de um sistema de controle computadorizado, uma AMC têm um sistema de controle híbrido, ou seja, alguns componentes têm seu próprio sistema de controle enquanto outros são integrados.

Groover (2000) define um sistema de manufatura FMS como uma célula de maquinas baseado na tecnologia de grupo GP altamente automatizada, configurada por um grupo de máquinas, usualmente máquinas CNC, integradas por um AMHS e controladas por um sistema de computador distribuído.

O FMS é chamado flexível porque pode processar uma variedade de peças simultaneamente em varias estações de trabalho, e as misturas de estilos de peças e quantidades de produção são consequência da demanda do mercado. Groover (2000) também classifica os FMS de acordo ao número de máquinas e nível de automatização:

- 1) Máquina simples,
- 2) Multi-máquinas com movimentação de materiais manual,
- 3) Multi-máquinas com movimentação de materiais mecanizada,
- 4) FMC, e
- 5) FMS.

Moore e Gupta (1996) definem FMS ou sistema de manufatura automatizado MAS (MAS Manufacturing Automatically System), como um sistema de máquinas CNC integrado por um computador, que simultaneamente podem processar uma variedade de tipos de peças.

Um termo mais apropriado para FMS pode ser sistema de manufatura flexível automatizado. O uso do termo *automatizado* distingue esta tecnologia de produção de outros sistemas de manufatura que são flexíveis, mas não automatizados como as células de processado GT. Ao mesmo tempo, o termo *flexível* distingue aos FMS de outro sistema de manufatura altamente automatizado, mas não é flexível como a linha de transferência convencional.

Um FMS ou um FAS pode ser definido como um SAP com automatização flexível, integrado através de três componentes, conforme mostra a Figura 3.5:

- 1) Estações de processamento ou montagem, como máquinas CNC, equipamento auxiliar de processamento e robôs.
- 2) Mecanismos mecatrônicos de movimentação de materiais e peças entre estações de trabalho, que pode ser um robô.
- 3) Supervisão e controle computadorizado central de todos os componentes.

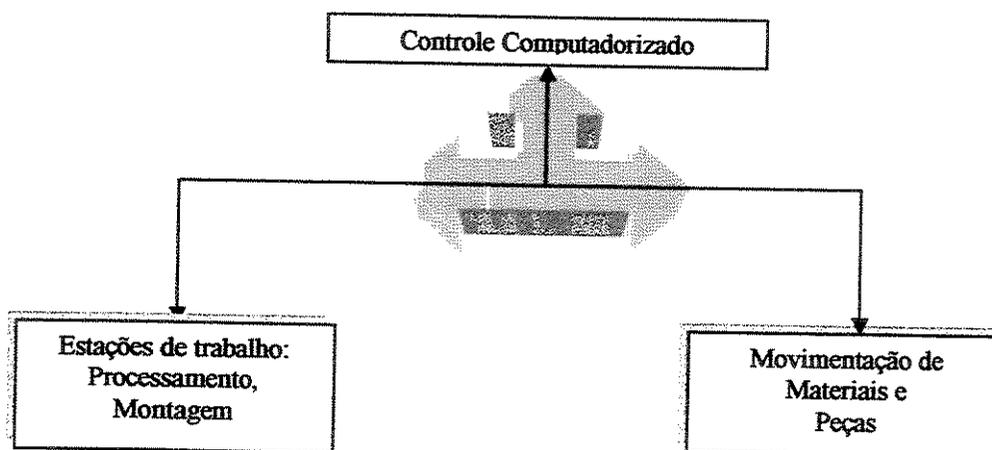


Figura 3.5. Equipamento básico de uma FMC.

A integração caracteriza aos FMS, integração de todos os componentes a um computador central onde se desenvolvem tarefas de programação, supervisão e controle, e integração física com um sistema de movimentação de materiais. A flexibilidade permite variar a configuração do hardware e também trocar a programação o software de cada um dos componentes do hardware.

Normalmente um projeto de FMS é direcionado para aplicações específicas, uma família específica de peças e processos específicos, sendo único. Um esquema de classificação precisa considerar as operações de processamento ou operações de montagem.

O projeto de FMS realiza uma destas duas funções, mas raramente as duas simultaneamente. Também, os FMS são projetados para peças prismáticas, para peças cilíndricas ou uma mistura delas, devendo considerar também o número de máquinas e a flexibilidade. Um esquema de classificação pode ter as seguintes categorias:

**ESTAÇÃO SIMPLES DE MANUFATURA FLEXÍVEL FMSS (FMSS Flexible Manufacturing Station Single).** Configurado por uma estação de processamento, um mecanismo de movimentação de materiais e peças, integrados com um sistema de controle e programação através de computador, sem a intervenção humana. As peças de trabalho terminadas são periodicamente descarregadas para a unidade de armazenagem, e novos materiais são carregados. Os FMSS podem-se configurar para operar em modo lote ou em modo flexível ou uma combinação dos dois. Quando opera em modo lote, a estação processa peças de um tipo somente em tamanhos específicos de lote, depois muda ao tipo seguinte de peça e lote específico para a segunda peça e assim sucessivamente. Quando opera em modo flexível, precisa satisfazer três das quatro habilidades de flexibilidade: diferentes peças, diferentes planos de produção e diferentes programas de peças.

**FMC.** Configurada com duas ou três estações de processamento e um sistema de movimentação de materiais integrados a um sistema de controle e programação através

de computador. O sistema de movimentação de materiais está ligado a uma estação de carregamento / descarregamento. Ademais, o sistema de movimentação inclui um sub sistema com capacidade limitada para armazenagem temporal das peças. Este tipo de FMS e precisa cumprir as quatro condições de flexibilidade.

**FMS.** Tem quatro ou mais estações de processamento integradas mecanicamente por um sistema de movimentação de peças e eletronicamente com um sistema de controle distribuído através de computador.

As principais diferenças entre uma FMC e um FMS são:

- 1) Número de estações de processamento. Uma FMC tem dois ou três estações de trabalho e um FMS tem quatro o mais.
- 2) Um FMS pode incluir outras estações que não são necessariamente estações de processamento de manufatura, mas são estações de apoio ou estações com equipamento auxiliar, como estações com máquinas de medição computadorizadas CMM (CMM Computer Machine Measurement) e outras.
- 3) Finalmente, os FMS têm um sistema de controle computadorizado maior e sofisticado, com funções adicionais, como supervisão e seguimento.

### **3.2 Modelagem com Sistemas CAD**

Os sistemas CAD são usados no projeto de peças, componentes e sistemas mecânicos em 2D e 3D, constituídos por comandos, e sua flexibilidade permite-lhes achar muitas variantes para criar e editar a modelagem geométrica. Tem três tipos de ferramentas para a modelagem em:

- 1) malha (wireframes);
- 2) superfícies (surfaces);
- 3) sólidos (solids).

### 3.2.1 Linguagem

A modelagem em malha é dos mais simples e são composta de pontos, linhas, arcos, círculos, e curvas. Os sistemas CAD têm métodos para definir pontos, linhas, arcos, círculos, elipses e parábolas. Também tem ferramentas matemáticas de parametrização de linhas retas e linhas curvas, úteis para o planejamento de trajetórias e definir a resolução das trajetórias de movimentação dos robôs.

As curvas paramétricas spline se definem como curvas polinomiais variáveis em continuidade. As curvas paramétricas spline cúbicas utilizam para interpolação entre duas curvas e não para o desenho de formas como as curvas Bezier e B - splines.

As curvas Bezier e B - splines são curvas baseadas em técnicas de aproximação com pontos de controle ao longo de sua trajetória. As curvas Bezier geram-se com quatro pontos de aproximação, as curvas B - splines são umas generalizações das curvas Bezier com mais pontos de aproximação. A modelagem em malha está baseada em comandos splines e elipses reais.

A modelagem em superfícies é mais completo que a modelagem em malha porque melhora sua precisão. Em sua elaboração consideram-se comandos e entidades analíticas como: superfícies planas geradas por interpolação entre duas linhas (ruled surface), superfícies geradas por o giro de uma poli linha assimétrica ao redor de um eixo (surface of revolution), superfície gerada por translação de uma curva em uma superfície ao longo de uma distancia específica na direção de um eixo (tabuled cylinder).

A modelagem através de sólidos são representações mais complexas, possuindo os seguintes comandos ou entidades analíticas:

- 1) BOX: através desse comando criam-se prismas retangulares em 3D a partir de bases quadradas ou retangulares, apoiadas no plano XY do espaço de trabalho. Considera-se a altura positiva na direção +Z e negativa na direção -Z.

- 2) **CONE**: através desse comando são gerados cones a partir de uma base que pode ser circular ou elíptica, sendo possível somente a geração de cones retos. A base do cone está no plano XY.
- 3) **CYLINDER**: através desse comando são gerados cilindros de forma similar ao cone;
- 4) **SPHERE**: através desse comando são gerados sólidos em forma de esfera, indicando o seu centro e o seu raio ou diâmetro.
- 5) **WEDGE**: através desse comando são obtidas cunhas que são paralelepípedos cortados por um plano diagonal.
- 6) **TORUS**: através desse comando são gerados toróides por revolução de um círculo em redor de um eixo posicionado no mesmo plano.

A utilização de sólidos são sempre as melhores ferramentas para a modelagem geométrica. Para obter uma visualização realística da geometria tri-dimensional, utiliza-se o comando **RENDER** para ocultar linhas no plano posterior. O comando **SHADED** é o primeiro passo para a modelagem com sólidos, onde o mesmo transforma em opaca a face posterior e visualiza graduações nos cores do modelo.

### **3. 2. 2. Álgebra Booleana**

As operações booleanas associadas a comandos representam a base do método para obtenção de sólidos compostos (figura 3.6). Os principais comandos utilizados são descritos a seguir:

1) **UNION**: dois ou mais sólidos se unem para configurar um só, absorvendo a parte comum ou superposta. É possível unir sólidos que não se cortem, ou seja, sólidos separados sem contatos.

2) **SUBSTRACT**: obtêm-se um sólido depois de subtrair um conjunto de sólidos ao outro conjunto de sólidos.

3) **INTERSECT**: obtêm-se um novo sólido configurado com a parte comum ou interseção dos sólidos interceptados.

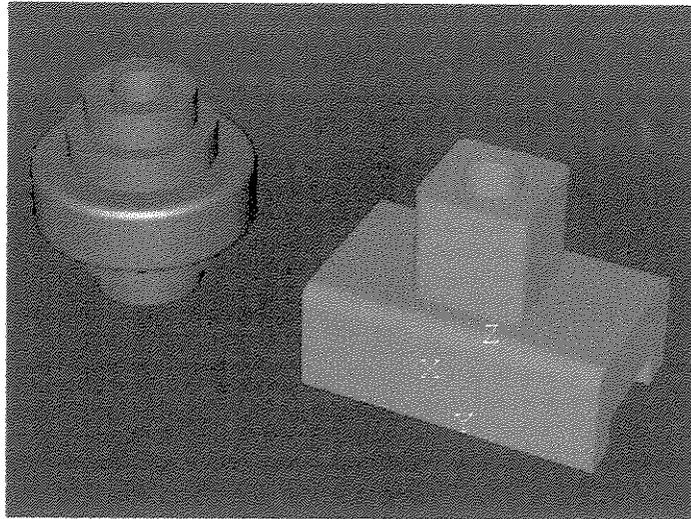


Figura 3. 6. Modelagem CAD de peças cilíndricas e peças prismáticas.

### 3. 3. Células Robotizadas

#### 3.3.1 - Operações dos Robôs Industriais

Uma das primeiras aplicações industriais dos robôs foi sua integração em um processo de fundição, Engelbert (1980). O robô era usado para descarregar a peça da matriz porque o ambiente da fundição não é apropriado para os trabalhadores. Atualmente, os robôs são usados na indústria de manufatura para:

- 1) Movimentação de materiais e peças,
- 2) Operações de processamento,
- 3) Montagem e controle de qualidade.

**MOVIMENTAÇÃO.** Nas aplicações de movimentação ou transferência, o robô move peças de um ponto a outro através de suas ferramentas terminais. Em algumas situações o posicionamento está acompanhado de orientação. Esta aplicação se conhece como operação *pick – and - place* e precisa de um robô de baixa tecnologia. Uma operação mais complexa de transferência é o empilhamento (ou a *paletização*), onde o robô recupera peças em um ponto para depositá-las em outro ponto como um *pallet* com

múltiplas posições. Como as posições são diferentes, têm-se uns aumentos do nível de complexidade e dificuldade na operação.

Outra operação similar é a *despaleitização*, onde são retiradas as peças de um arranjo em um *pallet* e se depositam em outro ponto. A operação de *empilhamento* uma peça é colocada sobre outra peça e o nível de dificuldade está nas diferentes posições verticais de posicionamento.

Na operação de *inserção*, o robô insere uma peça dentro de outra peça previamente preparada, onde o nível de dificuldade está na posição de posicionamento. Durante o processo de carregamento / descarregamento de máquinas ferramentas, o robô transfere peças aos pontos de trabalho e fora, como: maquinado, fundição, moldagem em plástico, conformado através de forjamento ou compressão, estampado e tratamentos térmicos.

**PROCESSAMENTO.** O robô realiza uma operação de processamento sobre a peça. Nesta operação tem-se uma ferramenta no elemento terminal para o processamento sobre a peça, como soldagem por arco elétrico, soldagem por pontos, pintura por pulverização, polido, curte com jato de água, e curte com laser.

**MONTAGEM E CONTROLE DE QUALIDADE.** A montagem e controle de qualidade são operações combinadas das anteriores: movimentação e processamento. A configuração SCARA e cartesiana são apropriadas para a montagem por sua precisão na operação.

### 3. 3. 2. Projeto Mecatrônico de Robôs

Para a modelagem CAD do robô pode-se escolher uma de duas opções: projetado ou selecionado de uma base de dados. No primeiro caso se apresenta uma metodologia esquematizada a desenvolver em três passos: (1) Modelagem, (2) Simulação e (3) Integração. Esta metodologia é sintetizada na Figura 3.7.

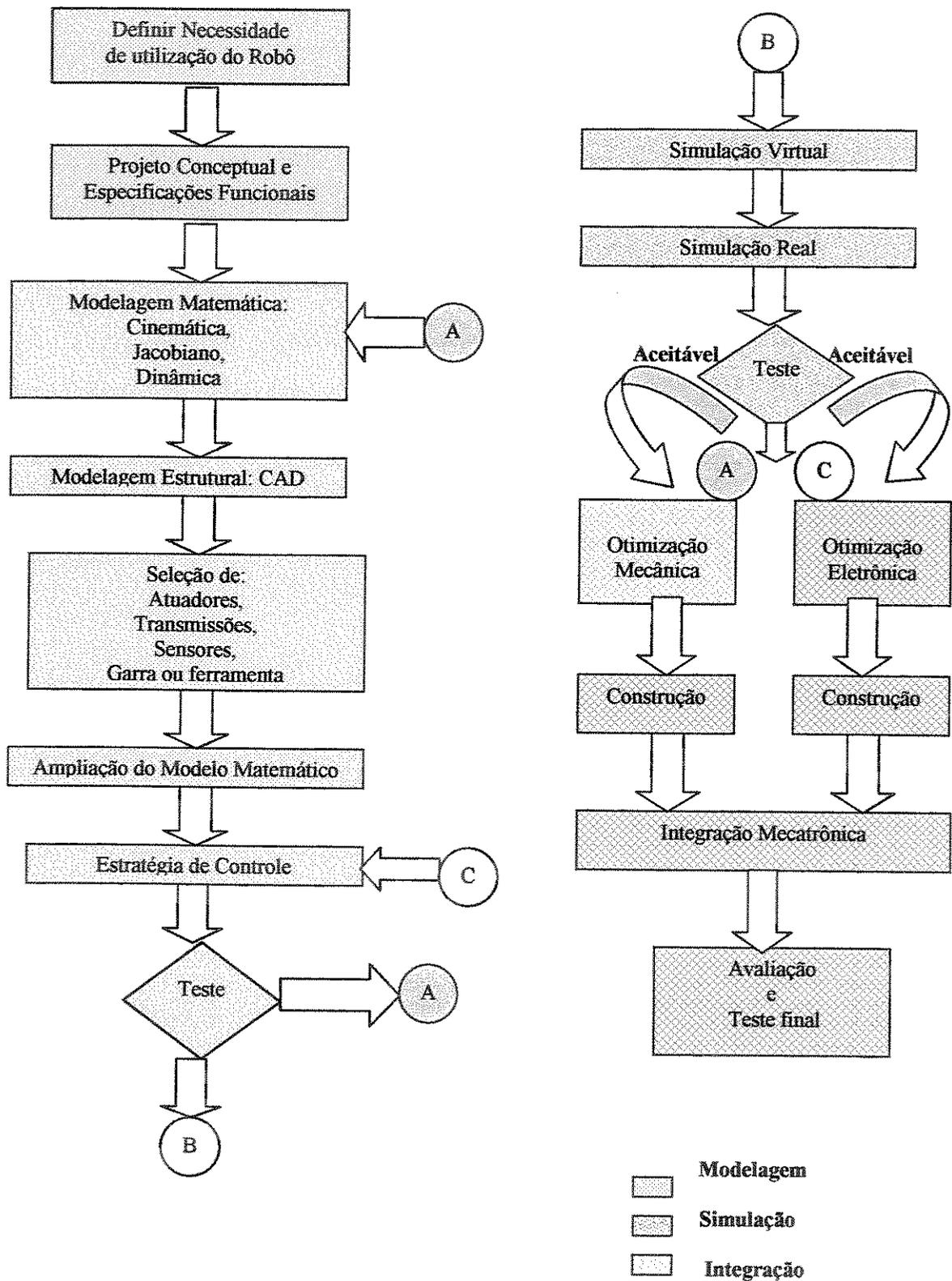


Figura 3.7: Metodologia para o projeto mecatrônico de robôs industriais.

### 3. 3. 2. 1. Modelagem

Para modelagem completa do sistema os seguintes passos deverão ser atendidos:

**Passo 1: Reconhecimento de uma necessidade:** Os robôs são componentes dos FMS e dos FAS. O passo inicial no projeto mecatrônico de robôs é definir sua função, sua estrutura e seu espaço de trabalho para projetar a FMC.

**Passo 2: Projeto conceitual e especificações funcionais:** Especificar condições de funcionamento, distâncias entre pontos de posicionamento, velocidades e acelerações na movimentação, cargas à transferir, estrutura definitiva, seleção e análise de materiais para os elementos, análise e configuração da seção dos elementos, análise de esforços, especificação de atuadores e transmissões, projeto de juntas, faixa de atuação das variáveis nas juntas e dimensões do robô.

**Passo 3: Modelagem Matemática:** Modelagem da cinemática, do Jacobiano, e da dinâmica do robô.

**Passo 4: Modelagem estrutural:** Definir estrutura básica do robô, selecionar materiais dos elementos, definir seção dos elementos, definir mecanismo das juntas, projetar punho, análise de cargas estática e dinâmicas, análise de inércia, tolerâncias dimensionais e configuração CAD.

**Passo 5: Seleção de atuadores, transmissões, sensores e ferramentas:** A partir da modelagem da dinâmica do robô, calcular forças e / ou torques nas juntas para selecionar atuadores, transmissões, sensores e ferramentas.

**Passo 6: Ampliação da modelagem matemático:** Incluir na modelagem matemática os modelos dos atuadores, transmissões e sensores para o projeto da estratégia de controle.

**Passo 7: Modelagem do sistema de controle:** Com a modelagem matemática ampliado, projeta-se a estratégia de controle. Pesquisar a estratégia de controle que cumpra os requerimentos de funcionamento aceitável e seja economicamente possível.

**Passo 8: Otimização do projeto:** Ajuste dos parâmetros e variáveis do robô.

### **3.3.2.2 - Simulação**

Para atender o problema da simulação do sistema os seguintes passos deverão ser atendidos:

**Passo 9: Simulação.** A Modelagem e simulação são tarefas superpostas na estratégia de controle para conhecer o comportamento do robô nas diferentes ações de controle. Normalmente a simulação precisa de um software especial.

**Passo 10. Otimização:** Baseados na simulação ajustam-se os parâmetros estáticos e dinâmicos do robô antes da integração de seus componentes em uma nova unidade.

### **3.3.2.3 - Integração**

Finalmente, na fase de integração os seguintes passos deverão ser atendidos:

**Passo 11. Integração:** A integração do hardware do sistema mecânico mais hardware do sistema eletrônico mais hardware do sistema de controle para obter uma nova unidade mecatrônica.

**Passo 12. Otimização e testes finais:** O ciclo de vida do novo robô depende do ciclo de vida de seus componentes mecatrônicos. A otimização do projeto mecatrônico do robô implica em ajustar detalhes mecânicos, eletrônicos e computadorizados integrados.

A validação final pode incluir o cálculo da resolução de controle CR (CR Control Resolution), a precisão e a repetibilidade. Com estes testes se caracterizam as principais funcionalidades do protótipo e se comparam com as propostas inicialmente. A Figura 3.2 mostra a modelagem CAD do robô projetado com seus parâmetros funcionais.

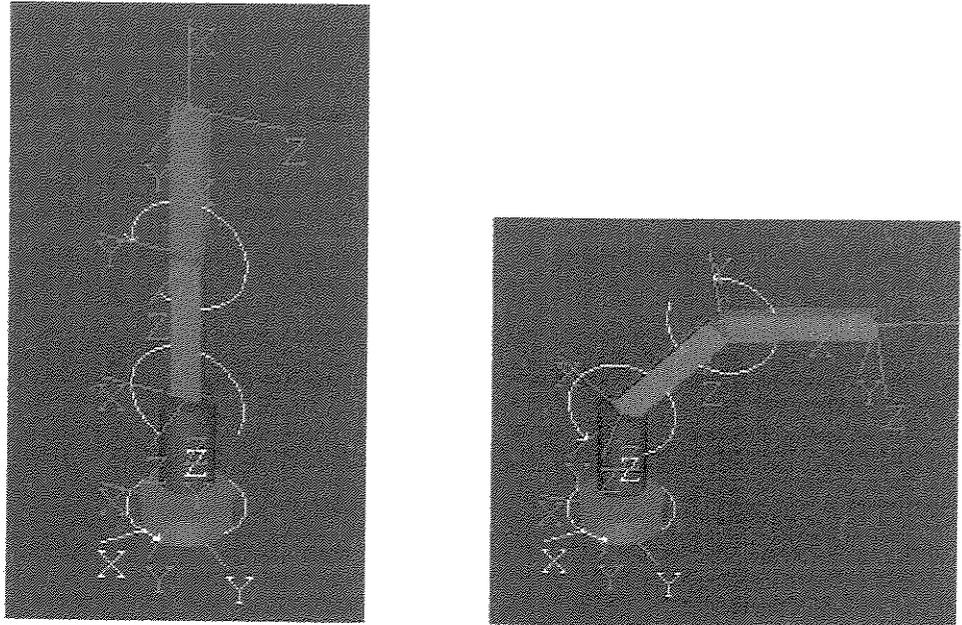


Figura 3.8: Modelagem CAD de robô de configuração angular RRR de 3 GL

### 3.3.3. Critérios de Seleção de Robôs

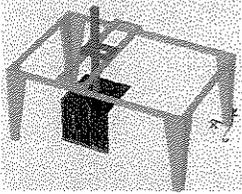
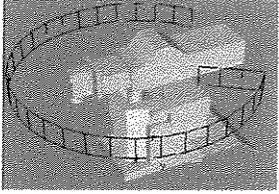
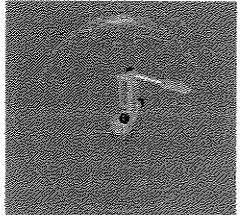
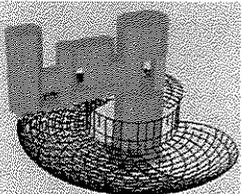
Na seleção de um robô de uma base de dados os seguintes aspectos deverão ser considerados:

- 1) Configuração,
- 2) Tipo de junta,
- 3) Garra ou ferramenta e,
- 4) Tamanho.

A configuração está relacionada com a operação do robô, a junta depende do tipo de configuração, a garra ou ferramenta depende da operação do robô e o tamanho depende das operações do robô na FMC.

Existem quatro configurações fundamentais de robôs comerciais: (1) Configuração cartesiana como o Robô Cincinnati Milacron T3886. (2) configuração cilíndrica como o Robô Umi RT3100. (3) Configuração angular como o Robô Motoman SK16-6. (4) Configuração SCARA como o Robô IBM SR 5428.

Tabela 3.1. Configurações fundamentais de robôs comerciais

| Nome       | Volume  | Espaço de Trabalho  |
|------------|---|---|
| Cartesiana | $L^3$<br>Quando L é o deslocamento de cada elemento     |    |
| Cilíndrica | $3\pi L^3$<br>Quando L é o Radio Do cilindro            |    |
| Angular    | $(32/3)\pi L^3$<br>Quando L é a longitude Dos elementos |  |
| SCARA      | $4\pi L^3$<br>Quando L é a longitude Dos elementos      |  |

A configuração cartesiana tem três juntas prismáticas (PPP) é muito usada em pórticos para movimentação de cargas de volume. A especificação da posição de um ponto se define em coordenadas cartesianas X, Y, Z e os valores das variáveis das juntas correspondem diretamente aos coordenadas da garra ou ferramenta. Com esta configuração o sistema de controle é simples.

A configuração cilíndrica tem duas juntas prismáticas e uma de rotação RPP. A posição é especificada através de coordenadas cilíndricas. Esta configuração é apropriada em FMC com um robô no centro atendendo a várias estações de trabalho. O volume de trabalho desta configuração RPP ou PRP, considerando um raio de rotação de 360 graus e faixa de atuação de deslocamento  $L$ , é um toróide de seção quadrada de raio interior  $L$  e raio exterior  $2L$  e o volume final é de  $3\pi L^3$

A configuração angular é uma estrutura com três juntas de rotação RRR. A posição da garra ou ferramenta se especifica em coordenadas angulares. A estrutura tem um melhor acesso aos espaços fechados, usa-se em tarefas complexas, sua configuração é apropriada no ensino e pesquisa. Com esta configuração obtém-se um grande volume de trabalho. Se a longitude de cada um dos três elementos é  $L$  e se tem um raio de giro de 360 graus, seu volume de trabalho é o de uma esfera de raio  $2L$ , ou seja,  $(32/3)\pi L^3$

A configuração SCARA foi especialmente projetada para desenvolver tarefas de montagem em um plano. Tem duas juntas de revolução com referência a dois eixos paralelos, e uma junta de deslocamento na direção perpendicular ao plano RRL. Se a longitude de seus elementos é  $L$ , o raio de giro de 360 graus e faixa de deslocamento  $L$ , então, seu volume de trabalho é  $4\pi L^3$

### **3.3.4. Ferramenta Terminal**

#### **3.3.4.1. Punho do Robô**

O punho é o mecanismo no extremo livre da cadeia cinemática da estrutura do robô. Está projetada para acoplar o extremo livre com a garra ou ferramenta. O punho permite a flexibilidade da estrutura do robô, porque uma mesma configuração pode desempenhar operações de manufatura ou movimentação de peças.

No punho os três eixos de orientação se interceptam ortogonalmente. Os três movimentos do punho são RPY (Roll, Pitch, Yaw) e identificam a orientação da garra ou

ferramenta no espaço de trabalho. Na Figura 3. 9 apresenta uma das configurações mais utilizadas em punhos para conectar a garra ou ferramenta à estrutura do robô e a Figura 3.10 apresenta o sistema de coordenadas.

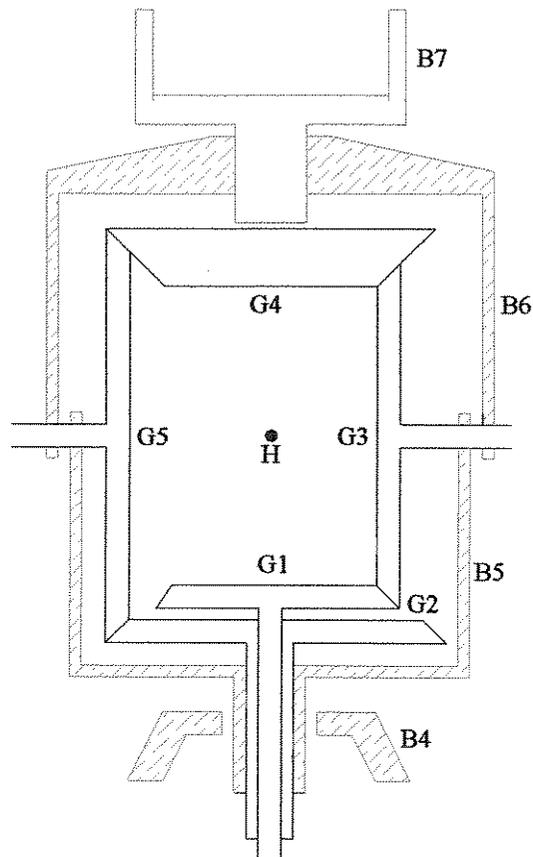


Figura 3. 9: Cadeia cinemática do punho esférico.

Este desenho está baseado num sistema de engrenagens cônico e a estrutura básica tem os corpos  $B_5$ ,  $B_6$  e  $B_7$ . Os graus de liberdade são  $\theta_4$  entre  $B_4$  e  $B_5$ ,  $\theta_5$  entre  $B_5$  e  $B_6$ , e  $\theta_6$  entre  $B_6$  e  $B_7$ ; e são movimentados com os engrenagens cônicos retos  $G_1G_3G_4$  e  $G_2G_5G_4$ .

Uma característica desta transmissão é o desenho do engrenagem  $G_4$  ao longe do eixo 6. Todos os eixos dos engrenagens se interceptam no ponto H que é o centro do punho. A movimentação da transmissão do punho é remoto a través dos tubos conectados

nos engrenagens  $G_1$  e  $G_2$  e ao corpo  $B_5$ .  $\phi_1$ ,  $\phi_2$  y  $\phi_3$  são as variáveis entre os tubos coaxiais.

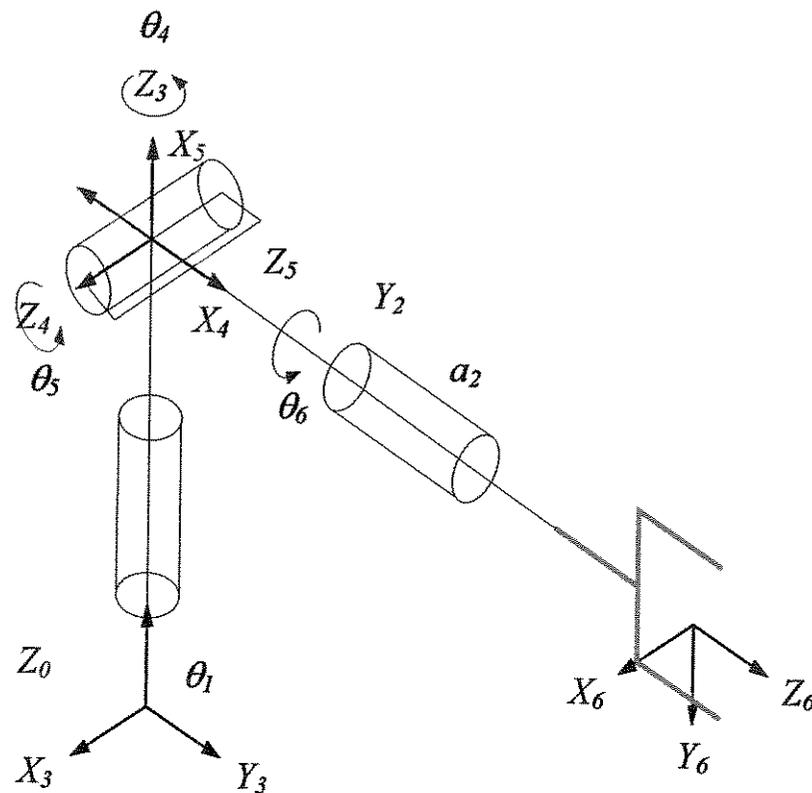


Figura 3.10: Configuração ortogonal do punho esférico.

### 3.3.4.2. Garra do Robô e Sistemas de Coordenadas

A garra ou ferramenta terminal permite o desenvolvimento das tarefas do robô dentro da FMC, tendo dois tipos de coordenadas: coordenadas iniciais e coordenadas finais. Os sistemas de simulação e programação *off-line* de robôs, baseados em sistemas CAD facilitam a modelagem da garra.

A modelagem através de CAD do mecanismo da garra identifica o Gripper como um mecanismo e define as juntas para cada um dos dedos, como é mostrado na Figura 3.4. O mecanismo da garra Gripper tem o sistema de coordenadas iniciais  $O_G(X_G, Y_G, Z_G)$  referente ao sistema de coordenadas da FMC (WCS World Coordinates Systems).

O sistema de coordenadas iniciais da garra tem  $-13$  unidades de deslocamento na direção X e  $90$  graus em torno de Y. O sistema de coordenadas finais da garra tem  $80$  unidades de deslocamento na direção X e  $90$  graus em torno de Y. Na robótica a rotação é positiva no sentido anti-horário. O conhecimento sobre os novos sistemas de coordenadas iniciais e finais é importante na programação *off-line* do robô. Na montagem da garra no punho da estrutura é preciso ajustar a modelagem da cinemática para o novo sistema de coordenadas finais.

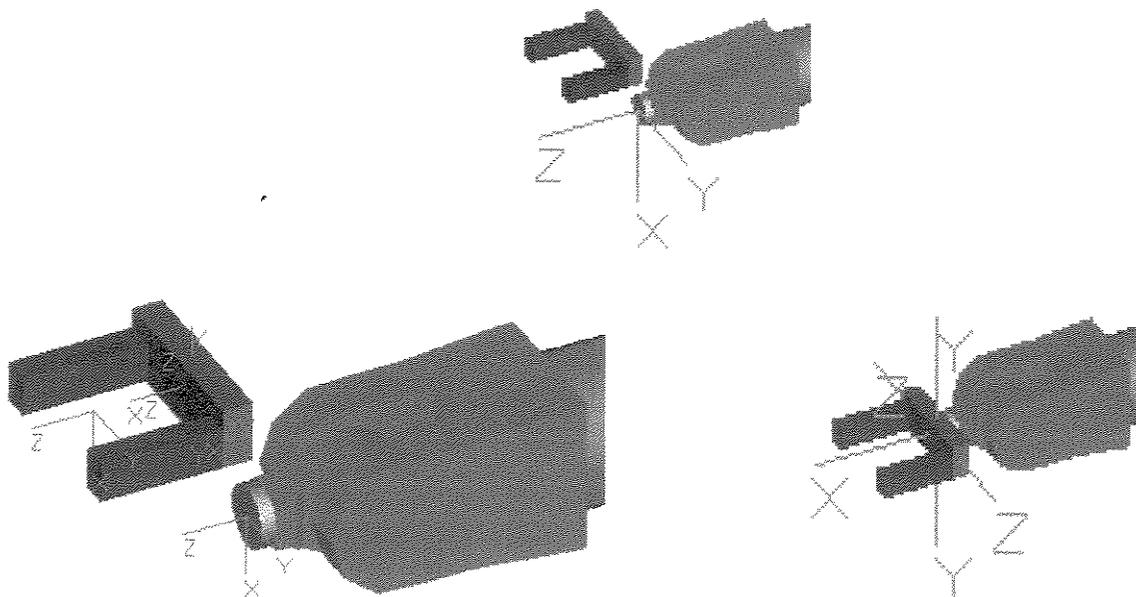


Figura 3.11: Mecanismo de garra e sistema de coordenadas

### 3.3.4.3. Ferramenta do Robô e Sistemas de Coordenadas

A seleção da ferramenta de uma base de dados depende da operação de manufatura do robô, como:

- 1) Soldagem por arco contínuo.
- 2) Soldagem por pontos.

- 3) Corte por jato de água.
- 4) Pintura por pulverização.
- 5) Montagem e inspeção.
- 6) Mecanizado.

Quando a ferramenta é instalada no punho do robô, o sistema de coordenadas do mesmo muda para a programação *off-line*. Conseqüentemente, o novo sistema de coordenadas pode ser definido como:

- 1) Sistema de coordenadas do punho  $O_P(X_P, Y_P, Z_P)$ .
- 2) Sistema de coordenadas iniciais da ferramenta  $O_{IF}(X_{IF}, Y_{IF}, Z_{IF})$ .
- 3) Sistema de coordenadas finais da ferramenta  $O_{FF}(X_{FF}, Y_{FF}, Z_{FF})$ .

Uma combinação destes três sistemas de coordenadas permite obter de forma automática através do software de programação *off-line* as novas coordenadas de programação. Na Figura 3. 12 são apresentadas alguns exemplos de ferramentas disponíveis no mercado.

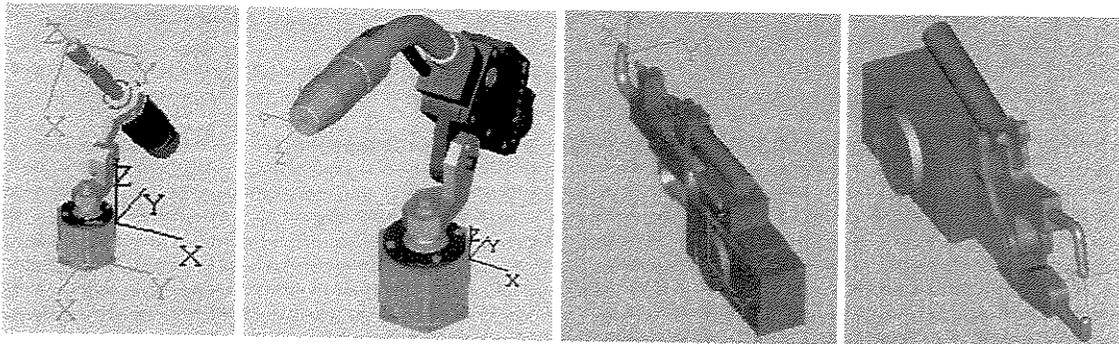


Figura 3. 12: Ferramentas para soldagem de arco elétrico e soldagem por pontos.

### 3.3.5. Espaço de Trabalho do Robô

O espaço de trabalho de um robô são todos os pontos que podem alcançar o sistema de coordenadas finais  $O_{FF}(X_{FF}, Y_{FF}, Z_{FF})$ . O conhecimento do espaço de

trabalho de um robô é importante para:

- 1) Definição da melhor configuração de um robô.
- 2) Definição do tamanho de uma FMC.
- 3) Localização e dimensionamento dos dispositivos mecatrônicos.
- 4) Apoiar o planejamento de trajetórias.

Dois espaços que configuram o espaço de trabalho total de um robô podem ser identificados:

- 1) O espaço primário: é um subconjunto do espaço total que contem os pontos onde todas as posições e orientações do sistema de coordenadas finais da garra ou ferramenta são alcançáveis.
- 2) Espaço secundário: é um subconjunto do espaço total que contem os pontos onde as posições e orientações do sistema de coordenadas finais da garra ou ferramenta não são alcançáveis.

Segundo Gupta (1999) três métodos podem ser utilizados para calcularmos o espaço de trabalho de um robô:

- 1) Análise direto: consiste de um análise iterativo dos pontos da periferia do espaço de trabalho para identificar singularidades.
- 2) Método da geração: consiste em gerar o espaço de cada junta, iniciando por a junta do último elemento do robô e finalizando com a junta do elemento da base do robô, e se configura o espaço total por partes.
- 3) Método por capas radiais RSL: tem seu desenvolvimento similar ao anterior, usando comandos CAD em 2D e se configura o espaço por capas sucessivas.

Nesta pesquisa, a modelagem do espaço de trabalho é gerada automaticamente com sistema CAD para dois primeiros eixos ou três primeiros eixos da configuração. Na Figura 3.13 se mostra o espaço de trabalho para as três primeiras juntas do robô IRB 1400 de ABB independente e o espaço de trabalho da FMC com robô e estação de trabalho; usando o software Workspace.

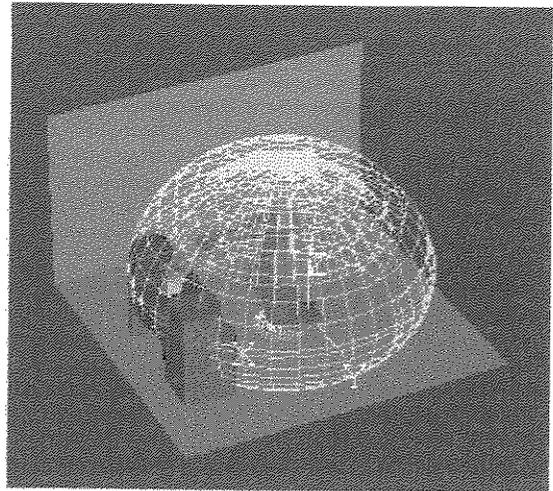
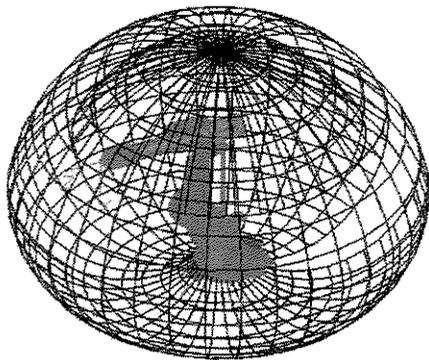
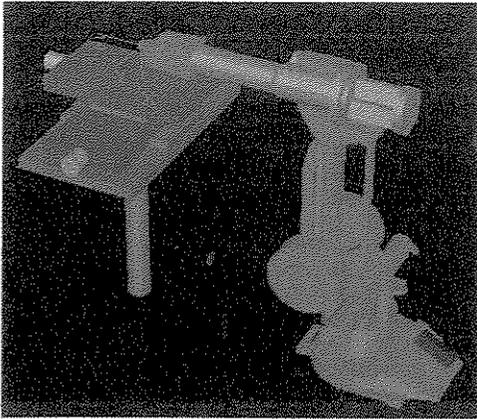


Figura 3. 13. Espaço de trabalho do robô IRB1400 de ABB e da FMC.

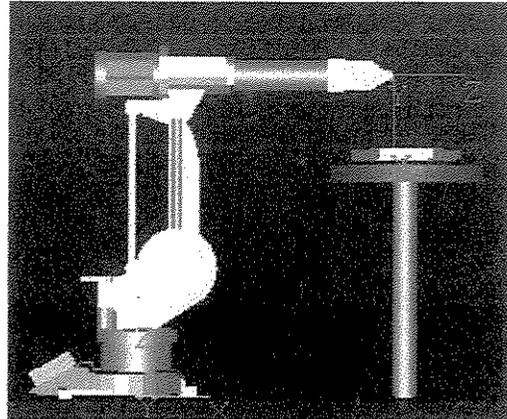
### 3.4. Modelagem dos Mecanismos

Os mecanismos na FMC são dispositivos modulares especiais para orientação e posicionamento das peças com referência ao sistema de coordenadas da garra ou ferramenta do robô. Os mecanismos também podem apreender, localizar, orientar, apoiar e direcionar as peças dentro do sistema de coordenadas da FMC (Figura 3.14). Neste trabalho de pesquisa foram utilizados os seguintes mecanismos:

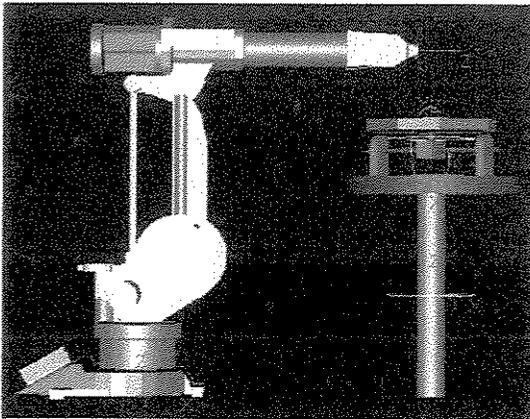
- 1) Posicionamento XY.
- 2) Rotativo MR.
- 3) Basculante MB.
- 4) Basculante MB e Rotativo MR Integrados.
- 5) Multi-Posição de Cinco Eixos MP5E.



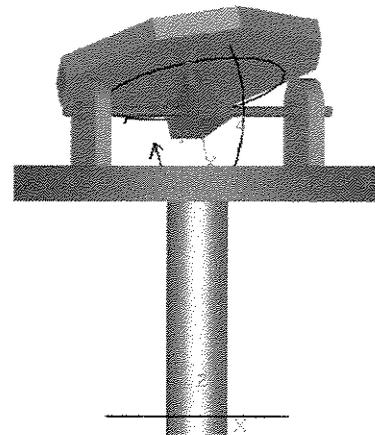
1) Posicionamento XY



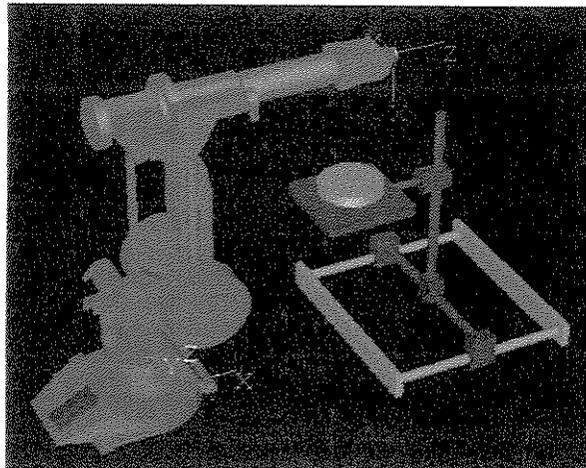
2) Rotativo MR



3) Basculante MB



4) MR + MB Integrados



5) Multi - Posição

Figura 3. 14: Modelagem dos mecanismos.

O mecanismo de posicionar XY com dois graus de liberdade é útil em estações de trabalho como máquinas de torner CNC e máquinas de fresar CNC. Tem três componentes: mesa inferior ou base, mesa intermédia e mesa superior. A mesa inferior é uma mesa fixa conectada a estrutura da máquina CNC e é apoio para as guias. A mesa intermediária tem deslocamento na direção X e é base para a mesa superior. A mesa superior tem movimento em Y com sistema de sujeição para a peça. A movimentação do mecanismo pode ser PTP ou CP dependendo do sistema de controle do mecanismo.

O MR de dois graus de liberdade é aplicado na movimentação rotacional de peças em estações de trabalho de máquinas ferramentas CNC. Ou como mecanismo de manipulação da peça durante o processamento de soldagem. Permite agilizar o processamento das peças complexas para o melhoramento da precisão de posicionamento angular. Pode ser utilizado em combinação com outros mecanismos modulares.

O MB é de apoio para o MR, porque permite adicionar um grau mais de liberdade para obter um posicionamento espacial e poder desenvolver trabalhos de processamento a movimentação complexa.

O MB permite o posicionamento espacial do MR para desenvolver mecanização complexa que necessitam posicionamento e movimentação espacial precisos. Também no processamento de soldagem o mecanismo integrado MB e MR de três graus de liberdade ortogonais PRR ou RRR facilita a integração da programação *off-line* dos robôs com a programação *off-line* do mecanismo.

O MP5E é de utilidade em funções de controle de qualidade. Com a montagem da peça sobre a mesa o sistema de controle de qualidade pode identificar o dimensionamento e precisão do acabado da peça. Este mecanismo tem deslocamento tri-dimensional com duas rotações. Ele pode ser aplicado em operações complexas de soldagem robotizada, desenvolvendo programação *off-line* integrada dos robôs de processamento com a programação *off-line* do mecanismo.

O desenvolvimento e modelagem modular de mecanismos virtuais para programação *off-line* de robôs em FMC, facilita a integração dos mecanismos simples para configurar mecanismos mais complexos que aumentam os graus de liberdade e poder desenvolver operações de manufatura e / ou montagem complexas.

Os mecanismos na FMC são dispositivos modulares especiais para orientação e posicionamento das peças com referência ao sistema de coordenadas da garra ou da ferramenta terminal do robô. Os mecanismos também podem sujeitar, localizar, orientar, apoiar e direcionar as peças dentro do sistema de coordenadas da FMC.

### 3. 5. Integração de Robôs e Mecanismos na FMC

Na Figura 3.15, mostra-se a modelagem da FMC apresentando a integração dos seguintes componentes:

- 1) MB integrado ao MR para configurar um novo mecanismo MB – MR com três graus de liberdade PRR.
- 2) Seleção do robô IRB 1400 de ABB para soldagem por arco elétrico de seis graus de liberdade RRR.RRR e do robô IRB 1500 de ABB para soldagem por arco elétrico com seis graus de liberdade RRR.RRR, da base de dados do software de simulação e programação *off-line*.
- 3) Seleção da base de dados do mesmo software da ferramenta CAT241D de soldagem por arco elétrico, para montagem no elemento terminal de cada um dos robôs.

Todos esses componentes são integrados na infra-estrutura física da FMC para configurar um sistema mecatrônico integrado de quinze graus de liberdade e é a configuração proposta para o desenvolvimento como FMC existente no Laboratório de Automação Integrada e Robótica - LAR da Faculdade de Engenharia Mecânica da UNICAMP. Estes três componentes da FMC integram-se a uma estação computadorizada para programação *off-line* e controle.

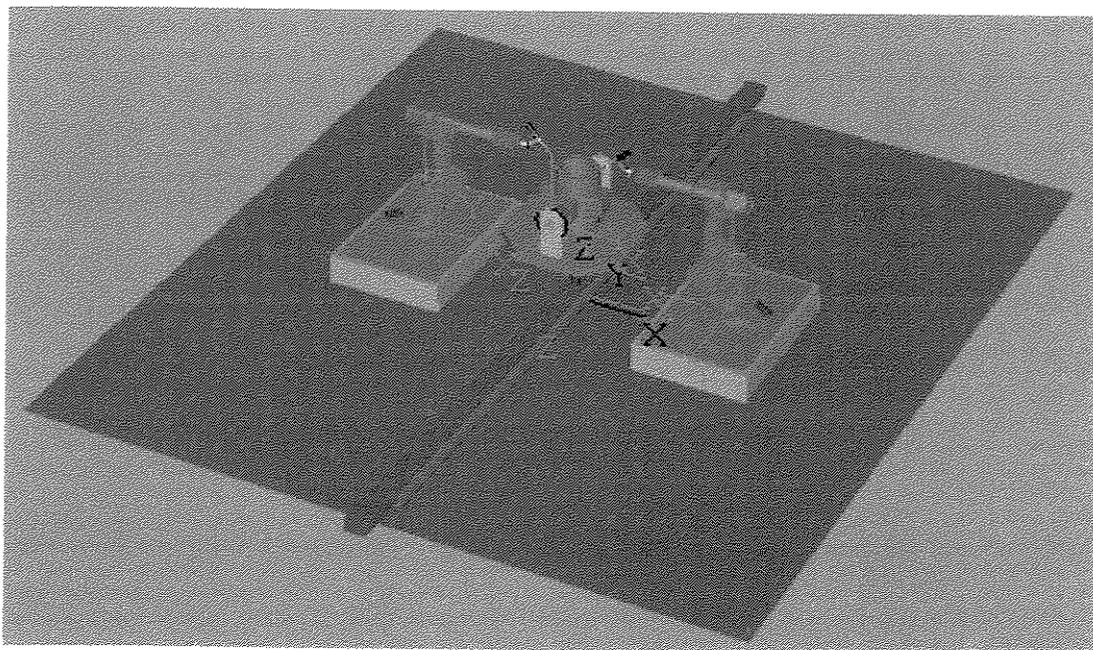


Figura 3. 15. FMC com dois Robôs e Mecanismo MB e MR

No capítulo 6 deste trabalho será apresentada detalhadamente toda a metodologia desenvolvida para simulação e programação *off-line* apresentada neste capítulo.

### 3. 6. Modelagem com PN das FMCs

As Redes de Petri (PN – Petri Nets) são as ferramentas preliminares utilizadas nesta pesquisa para validação do funcionamento da FMC. PN são grafos direcionados constituídos de dois elementos básicos: lugares e transições no qual um lugar esta seguida por uma transição e vice-versa.

Essa representação gráfica normalmente leva a soluções por simulação e será o caso considerado. Na representação gráfica, a estrutura da PN é definida por três conjuntos: Um conjunto de lugares P; um conjunto de transições T; e um conjunto de arcos direcionados A. Um arco conecta transição a um lugar ou um lugar a uma transição. Um lugar é uma entrada para uma transição se existe algum arco da transição para o lugar.

Uma definição pode-se escrever, segundo Dicesare, Harhalakis, Proth, Silva, e Vernadat (1993) e Zhou (1995):

$$PN = (P, T, A)$$

$$P = \{p_1, p_2, \dots, p_n\}$$

$$T = \{t_1, t_2, \dots, t_n\}$$

$$A = A_i \cup A_0$$

$$A_i \subseteq P \times T$$

$$A_0 \subseteq T \times P$$

Onde  $p_1, p_2, \dots, p_n$  são lugares e  $t_1, t_2, \dots, t_n$  são transições.

Na representação gráfica, lugares são simbolizados por círculos e transições por barras; Arcos são vetores de um lugar para uma transição ou de uma transição para um lugar. A dinâmica do sistema físico na PN é representada por marcas nos lugares da Rede. O número e as posições das marcas mudam para representar as mudanças do sistema físico. Lugares são pequenas bolas no interior dos círculos que representam os lugares. Um lugar contendo uma ou mais marcas é chamado de lugar marcado. Uma Rede contendo marcas é chamada de Rede marcada. A marcação de uma PN é a designação de marcas aos lugares da Rede.

Uma FMC é um SAP integrado por componentes mecatrônicos e sua modelagem em PN se encaminha à validação de sua função de processamento.

O processamento pode ser *seqüencial* quando as operações são em série, *concorrente* quando dois processos são paralelos e concorrem na mesma transição, *conflitivo* quando duas operações podem seguir rotas diferentes, mas as duas transições saem do mesmo lugar, *cíclico* quando a operação desenvolve um circuito fechado, e mutuamente *excludentes* quando duas operações não podem realizasse ao mesmo tempo, sem considerar que compartilhem o mesmo recurso físico.

Dicesare, Harhalakis, Proth, Silva e Vernadat (1993), e Zhou (1995) apresentam uma metodologia para a modelagem de sistemas de manufatura com Redes de Petri PN que pode estender sua aplicação a FMC. Esta metodologia se resume em função das seguintes operações, conforme passos descritos a seguir:

**Passo 1: Identificação das operações e recursos:** Descrever o SAP, identificar os eventos principais, operações, processos, recursos, condições, informação de rotas, etc.

**Passo 2: Identificação de relações:** As relações entre eventos identificados no passo 1. Definir políticas para compartilhar recursos, definir rotas, definir estrutura da PN.

**Passo 3: Desenho da PN:** Desenhar e marcar os lugares e / ou transições que representam os eventos, operações e / os processos, configurar a PN de acordo as relações no passo 2, inserir lugares e transições necessárias.

**Passo 4: Modificação da PN.**

## Capítulo 4

### Integração da Cinemática de Robôs em FMC

Um robô é uma cadeia cinemática aberta, programável e re-programável, o que faz que seja flexível em software; e sua configuração permite acomodar-se a suas necessidades de operação, o que a torna flexível em hardware.

O robô IRB 1400 de ABB apresentado na Figura 4.1 é um robô de seis eixos de rotação desenvolvido especificamente para aplicações direcionadas a indústria de manufatura que utilizam automação baseada em sistemas robotizados e possui uma estrutura aberta adaptada especialmente para flexibilidade de uso, podendo comunicar-se com sistemas externos. Será utilizado como base para desenvolver a modelagem da cinemática e a modelagem do Jacobiano.

A modelagem da cinemática direta, cinemática inversa e o Jacobiano do robô estão baseadas nos conceitos descritos pelos autores Paul (1981), Craig (1986), Rivin (1986), Fu *et al* (1988), Spong (1989), Stadler (1995), Sciavicco e Siciliano (1997), Barrientos *et al* (1997) e Carvajal (2000). Neste trabalho é desenvolvida uma metodologia original para obtenção da cinemática baseada em MTH para o robô IRB 1400 de ABB. Sendo o desenvolvimento da modelagem da cinemática com Quaternions uma importante contribuição do autor para o ensino e pesquisa nesta área.

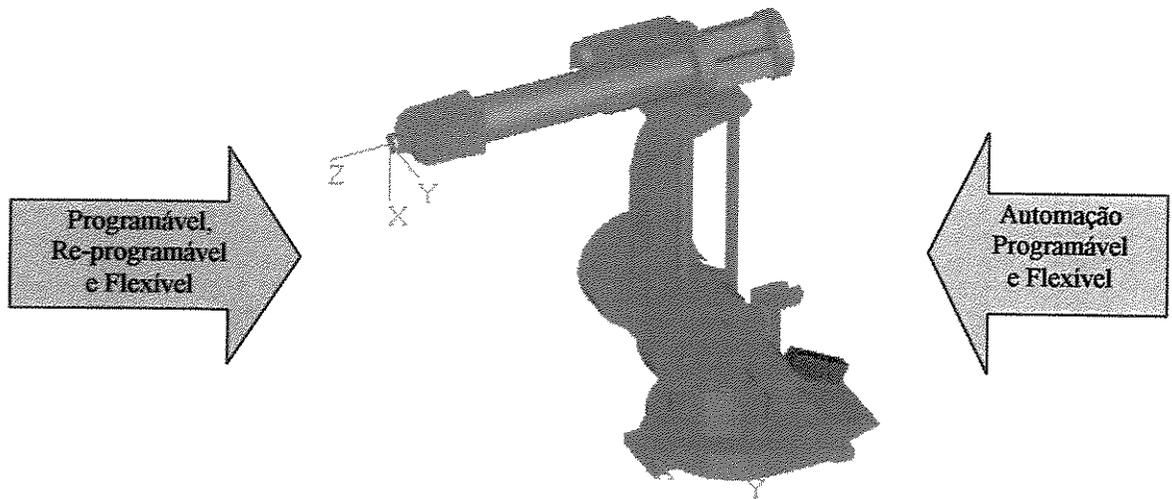


Figura 4.1: Robô industrial IRB 1400 de ABB.

#### 4. 1. Modelagem da Cinemática Direta com MTH

A modelagem da cinemática direta implica na representação do sistema de coordenadas da garra ou ferramenta terminal (TCP)  $O_n(X_n, Y_n, Z_n)$  em relação ao sistema de coordenadas da base do robô  $O_0(X_0, Y_0, Z_0)$  através da seguinte expressão:

$$T_n^0(q) = A_1^0(q_1) \cdot A_2^1(q_2) \cdot \dots \cdot A_n^{n-1}(q_n) \quad (4.1)$$

$$T^0(q) = \begin{bmatrix} n_x(q) & s_x(q) & a_x(q) & p_x \\ n_y(q) & s_y(q) & a_y(q) & p_y \\ n_z(q) & s_z(q) & a_z(q) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

Onde:

$q$  representa as variáveis da junta;

$n$ ,  $s$ , e  $a$  são vetores unitários do sistema de coordenadas da garra ou ferramenta e

$p$  é seu vetor posição em relação ao sistema de coordenadas da base  $O_0(X_0, Y_0, Z_0)$ .

A Figura 4.2 ilustra o sistema de coordenadas do TCP referente ao sistema da base do robô, é a representação da cinemática direta.

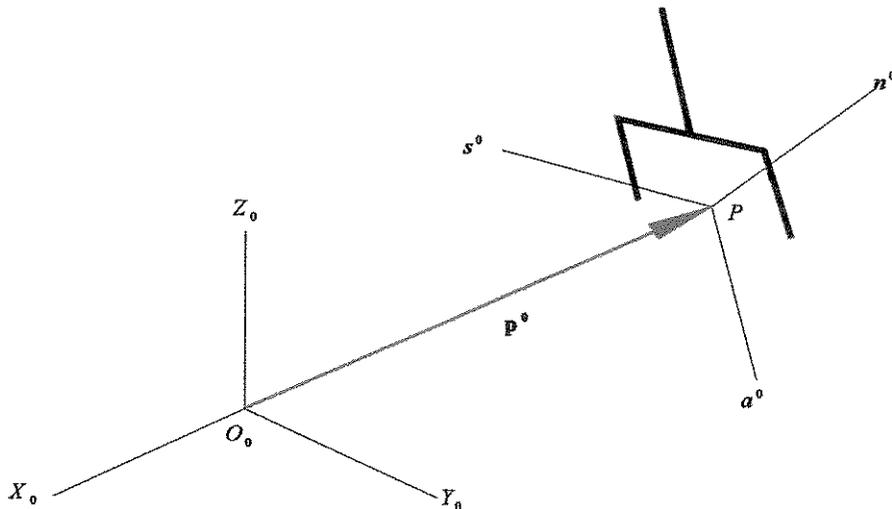


Figura 4. 2: Rotação e translação das coordenadas da garra ou ferramenta em função das coordenadas da base do robô.

A seguir será apresentada uma metodologia para desenvolver a modelagem da cinemática direta baseada na sistemática de Denavit – Hartenberg (1955):

**Passo 1:** Definir as direções dos eixos  $Z_0, Z_1, \dots, Z_{n-1}$ , a partir da base.

**Passo 2:** Definir a origem  $O_0$  do sistema de coordenadas da base no eixo  $Z_0$ . Os eixos  $X_0$  e  $Y_0$  escolhidos de acordo com a regra da mão direita.

**Passo 3:** Definir a origem de  $O_i$  na intercessão  $Z_i$  com a normal comum entre os eixos  $Z_{i-1}$  e  $Z_i$ . Se os eixos  $Z_{i-1}$  e  $Z_i$  são paralelos e a junta  $i$  é de revolução, então, localizar  $O_i$  para que  $d_i = 0$ ; se a junta  $i$  é prismática, localizar  $O_i$  em uma posição de referência para estabelecer um limite mecânico.

**Passo 4:** Escolher o eixo  $X_i$  ao longo da normal comum aos eixos  $Z_{i-1}$  e  $Z_i$  com direção

da junta  $i$  à junta  $i+1$ .

**Passo 5:** Escolher o eixo  $Y_i$ , para que com  $X_i$  se cumpra a regra da mão direita.

**Passo 6:** Escolher o sistema de coordenadas da garra ou ferramenta onde  $X_n$  seja normal à  $Z_{n-1}$ ; Se a junta da garra ou ferramenta é de revolução, então, alinhar  $Z_n$  com a direção de  $Z_{n-1}$ .

**Passo 7:** Para  $i = 1, 2, \dots, n$  construir a Tabela com os parâmetros  $a_i, d_i, \alpha_i, \theta_i$ .

**Passo 8:** Baseados nos parâmetros da tabela anterior calcular as MTH  $A_i^{i-1}(q_i)$  para  $i = 1, 2, \dots, n$ .

**Passo 9:** Definir a modelagem da cinemática direta  $T_n^0(q) = A_1^0(q_1)A_2^1(q_2)\dots A_n^{n-1}(q_n)$  para a posição e orientação do sistema de coordenadas da garra ou ferramenta, em relação ao sistema de coordenadas da base do robô.

Na Figura 4. 3 mostram-se os sistemas de coordenadas das juntas e na Tabela 4. 1 os parâmetros e variáveis do Robô IRB 1400 da ABB<sup>TM</sup>. Baseados nesta informação obtém-se as seguintes MTH para cada uma das juntas do robô IRB 1400, apresentada na Tabela 4.2.

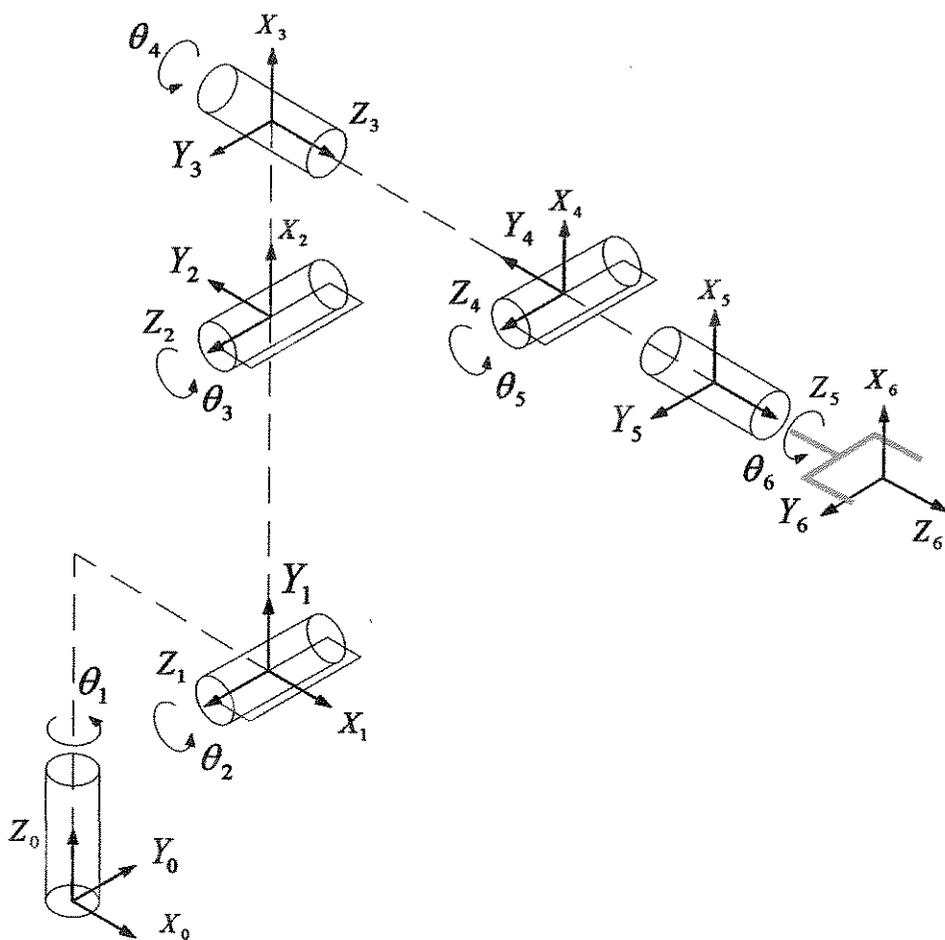


Figura 4. 3: Sistemas de coordenadas.

Tabela 4. 1. Parâmetros e variáveis de Denavit – Hartenberg.

| <b>J</b> | $\alpha_i$ | $\theta_i$ | $d_i$ | $a_i$ |
|----------|------------|------------|-------|-------|
| <b>1</b> | 90         | 0          | d1    | a1    |
| <b>2</b> | 0          | 90         | 0     | a2    |
| <b>3</b> | 90         | 0          | 0     | a3    |
| <b>4</b> | -90        | 0          | d4    | 0     |
| <b>5</b> | 90         | 0          | 0     | 0     |
| <b>6</b> | 0          | 0          | d6    | 0     |

Tabela 4. 2. Matrizes de transformação de coordenadas homogêneas.

|  |  |
|--|--|
| $\mathbf{A}_1^0(\theta_1) = \begin{bmatrix} c_1 & 0 & s_1 & a_1c_1 \\ s_1 & 0 & -c_1 & a_1s_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | $\mathbf{A}_2^1(\theta_2) = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| $\mathbf{A}_3^2(\theta_3) = \begin{bmatrix} c_3 & 0 & s_3 & a_3c_3 \\ s_3 & 0 & -c_3 & a_3s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$   | $\mathbf{A}_4^3(\theta_4) = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$        |
| $\mathbf{A}_5^4(\theta_5) = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$             | $\mathbf{A}_6^5(\theta_6) = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$         |

A modelagem da cinemática direta do robô pode ser decomposta em duas partes: (1) estrutura mecânica  $\mathbf{T}_3^0(q_A)$  e (2) garra ou ferramenta  $\mathbf{T}_6^3(q_B)$ . Os dois modelos integram-se em um só modelo para obter a modelagem final da cinemática direta:

$$\mathbf{T}_6^0(q) = \mathbf{T}_3^0(q_A) \cdot \mathbf{T}_6^3(q_B) \quad (4.3)$$

$$\mathbf{T}_3^0(q_A) = \mathbf{A}_1^0(\theta_1) \cdot \mathbf{A}_2^1(\theta_2) \cdot \mathbf{A}_3^2(\theta_3) \quad (4.4)$$

$$\mathbf{T}_3^0(q_A) = \begin{bmatrix} c_1 & 0 & s_1 & a_1c_1 \\ s_1 & 0 & -c_1 & a_1s_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & 0 & s_3 & a_3c_3 \\ s_3 & 0 & -c_3 & a_3s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_3^0(q_A) = \begin{bmatrix} c_{123} - c_1 s_{23} & s_1 & c_{13} s_2 + c_{12} s_3 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} + a_3 s_{23} c_1 \\ c_{23} s_1 - s_{123} & -c_1 & c_3 s_{12} + c_2 s_{13} & a_1 s_1 + a_2 c_2 s_1 + a_3 c_{23} s_1 - a_3 s_{123} \\ c_3 s_2 + c_2 s_3 & 0 & -c_{23} + s_{23} & a_2 s_2 + a_3 s_3 c_2 + d_1 + a_3 c_3 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

$$\mathbf{T}_6^3(q_B) = \mathbf{A}_4^3(\theta_4) \cdot \mathbf{A}_5^4(\theta_5) \cdot \mathbf{A}_6^5(\theta_6) \quad (4.5)$$

$$\mathbf{T}_6^3(q_B) = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_6^3(q_B) = \begin{bmatrix} c_{456} - s_{46} & -c_6 s_4 - c_{45} s_6 & c_4 s_5 & c_4 s_5 d_6 \\ c_{56} s_4 + c_4 s_6 & c_{46} - c_5 s_{46} & s_{45} & s_{45} d_6 \\ -c_6 s_5 & & c_5 & c_5 d_6 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

A modelagem integrada da cinemática direta pode escrever-se:

$$\mathbf{T}_6^0(q) = \begin{bmatrix} n_x(q) & s_x(q) & a_x(q) & p_x \\ n_y(q) & s_y(q) & a_y(q) & p_y \\ n_z(q) & s_z(q) & a_z(q) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$$n_x = -s_5 c_6 (c_{13} s_2 + c_{12} s_3) + s_1 (c_{56} s_4 + c_4 s_6) + (c_{123} - c_1 s_{23}) (c_{456} - s_{46})$$

$$n_y = -s_5 c_6 (c_3 s_{12} + c_2 s_{13}) - c_1 (c_{56} s_4 + c_4 s_6) + (c_{23} s_1 - s_{123}) (c_{456} - s_{46})$$

$$n_z = -s_5 c_6 (-c_{23} + s_{23}) + (c_3 s_2 + c_2 s_3) (c_{456} - s_{46})$$

$$s_x = s_{56} (c_{13} s_2 + c_{12} s_3) + (c_{123} - c_1 s_{23}) (-c_6 s_4 - c_{45} s_6) + s_1 (c_{46} - c_5 s_{46})$$

$$s_y = s_{56} (c_3 s_{12} + c_2 s_{13}) + (c_{23} s_1 - s_{123}) (-c_6 s_4 - c_{45} s_6) - c_1 (c_{46} - c_5 s_{46})$$

$$s_z = s_{56}(-c_{23} + s_{23}) + (c_3s_2 + c_2s_3)(-c_6s_4 - c_{45}s_6)$$

$$a_x = c_5(c_{13}s_2 + c_{12}s_3) + c_4s_5(c_{123} - s_{23}c_1) + s_{145}$$

$$a_y = c_5(c_3s_{12} + c_2s_{13}) + c_4s_5(c_{23}s_1 - s_{123}) - c_1s_{45}$$

$$a_z = c_5(-c_{23} + s_{23}) + c_4s_5(c_3s_2 + c_2s_3)$$

$$p_x = a_1c_1 + a_2c_{12} + a_3c_{123} - a_3c_1s_{23} + (d_4 + c_5d_6)(c_{13}s_2 + c_{12}s_3) + c_4s_5d_6(c_{123} - c_1s_{23}) + s_{145}d_6$$

$$p_y = a_1s_1 + a_2c_2s_1 + a_3c_{23}s_1 - a_3s_{123} + (d_4 + c_5d_6)(c_3s_{12} + c_2s_{13}) + c_4s_5d_6(c_{23}s_1 - s_{123}) - c_1s_{45}d_6$$

$$p_z = a_2s_2 + a_3c_2s_3 + d_1 + a_3c_3s_2 + (d_4 + c_5d_6)(-c_{23} + s_{23}) + c_4s_5d_6(c_3s_2 - c_2s_3)$$

Se a posição e orientação do sistema de coordenadas da garra ou ferramenta terminal do robô  $O_n(X_n, Y_n, Z_n)$  calculado com a expressão (4. 18) não é correta, então, cria-se uma MTH de ajuste, porque o sistema mecânico do robô pode gerar erros e o sistema eletrônico pode gerar erros na etapa de montagem pelo fabricante, então, o robô precisa de calibração para ajustar seus parâmetros antes de entrar em utilização.

## 4. 2. Modelagem da Cinemática Inversa com MTH

A cinemática inversa consiste em encontrar as configurações possíveis das juntas, correspondente a uma posição e orientação específica da garra ou ferramenta. Sua solução é importante para o planejamento das trajetórias sendo complexo porque:

- 1) As equações a desenvolver em geral são normalmente não lineares;
- 2) Podem encontrar-se soluções múltiplas (redundantes);
- 3) Podem existir soluções infinitas;
- 4) Podem existir soluções não aceitáveis devido às singularidades do robô.

A existência de três juntas de revolução consecutivas com eixos interceptados em um só ponto para todas as possíveis configurações se conhece como a condição de resolubilidade proposta por Pieper (1968). A maioria dos robôs cumpre esta condição, como se mostra na Figura 3. 14, e Figura 3. 15. Para o robô IRB 1400 no ponto W da

Figura 4. 4 interceptam-se os eixos  $Z_3$ ,  $Z_4$  e  $Z_5$  que correspondem ao sistema de coordenadas  $O_4(X_4, Y_4, Z_4)$ .

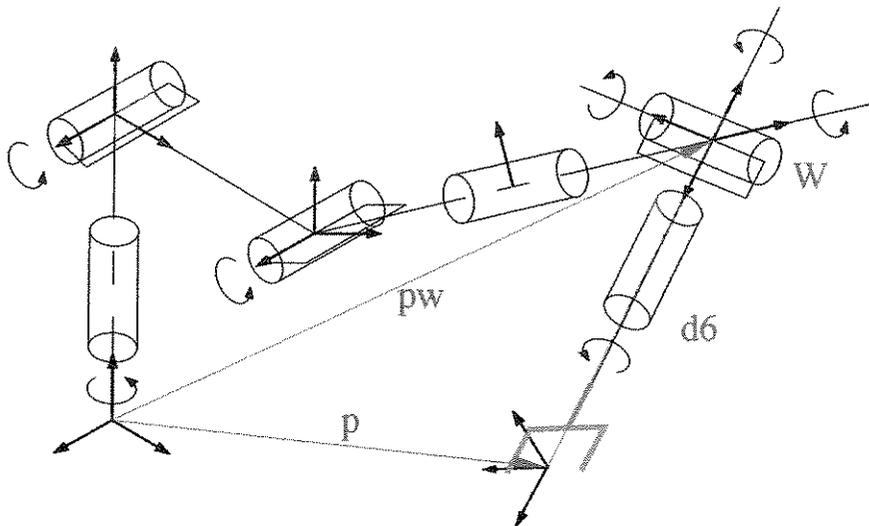


Figura 4. 4: Condição de resolubilidade de Pieper (1968)

Uma metodologia para calcular a cinemática inversa consiste em encontrar o ponto da configuração do robô, que possa representar-se como uma função da posição e orientação da garra ou ferramenta e como uma função de um número reduzido de variáveis de junta. O que significa propor o problema da cinemática inversa em duas partes:

- 1) Cinemática inversa da estrutura mecânica e
- 2) Cinemática inversa do punho esférico.

Esta metodologia pode desenvolver-se com os seguintes passos:

**Passo 1:** Calcular a posição do punho  $P_W$  ( $q_1, q_2, q_3$ ).

**Passo 2:** Solucionar a cinemática inversa para ( $q_1, q_2, q_3$ ).

**Passo 3:** Calcular  $R_3^0$  ( $q_1, q_2, q_3$ ).

**Passo 4:** Calcular  $\mathbf{R}_6^3(\theta_4, \theta_5, \theta_6) = \mathbf{R}_3^{0T} \mathbf{R}$

**Passo 5:** Resolver a cinemática inversa para a orientação  $(\theta_4, \theta_5, \theta_6)$

Para o robô IRB 1400 tem-se a seguinte expressão da cinemática direta:

$$\mathbf{T}_6^0(q) = \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \cdot \mathbf{A}_3^2(q_3) \cdot \mathbf{A}_4^3(q_4) \cdot \mathbf{A}_5^4(q_5) \cdot \mathbf{A}_6^5(q_6) \quad (4.8)$$

$\mathbf{T}_6^0(q)$  é conhecida e representa a cinemática direta do robô, em tanto que cada uma das MTH da direita da equação representa a transformação de cada junta. Para aplicar a metodologia de Pieper (1968) é necessário obter as inversas das MTH  $\mathbf{A}_i^{i-1}(q_i)$ . Assim:

$$\mathbf{T} = \mathbf{pR}^T \quad (4.9)$$

$$[\mathbf{T}]^{-1} = \{[\mathbf{p}][\mathbf{R}^T]\}^{-1} = \{[\mathbf{R}^T]\}^{-1}[-\mathbf{p}] = [\mathbf{R}^T]^{-1}[-\mathbf{p}] \quad (4.10)$$

$$[\mathbf{T}]^{-1} = \begin{bmatrix} n_x & n_y & n_z & 0 \\ s_x & s_y & s_z & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[\mathbf{T}]^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T \cdot p \\ s_x & s_y & s_z & -s^T \cdot p \\ a_x & a_y & a_z & -a^T \cdot p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

A idéia da metodologia de Pieper (1968) é que a posição W seja função de  $\theta_1, \theta_2$  e  $\theta_3$ . Como a posição W pode ser obtida também a partir da posição e orientação da garra ou ferramenta terminal do robô, então,  $\theta_1, \theta_2$  e  $\theta_3$  também podem ser determinados da

posição e orientação da garra ou ferramenta. A metodologia da cinemática inversa para o Robô IRB 1400 pode ser obtida da posição  $W \equiv O_4$  e pode escrever-se segundo uma análise geométrico:

$$\mathbf{p}_W = \mathbf{p} - d_6 \cdot \mathbf{a} \quad (4.12)$$

$$\begin{bmatrix} p_{wx} \\ p_{wy} \\ p_{wz} \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - d_6 \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (4.13)$$

$$\mathbf{p} - d_6 \cdot \mathbf{a} = \left\{ \mathbf{T}_4^0 \right\}_4 = \mathbf{p}_W \quad (4.14)$$

$$\mathbf{A}_1^0(\theta_1) \cdot \mathbf{A}_2^1(\theta_2) \cdot \mathbf{A}_3^2(\theta_3) \cdot \mathbf{A}_4^3(\theta_4) = \mathbf{p}_W \quad (4.15)$$

$$\left\{ \mathbf{A}_2^1(\theta_2) \cdot \mathbf{A}_3^2(\theta_3) \cdot \mathbf{A}_4^3(\theta_4) \right\} = \left\{ \mathbf{A}_1^0 \right\}_1^{-1} \cdot \mathbf{p}_W \quad (4.16)$$

$$\left\{ \mathbf{A}_3^2(\theta_3) \cdot \mathbf{A}_4^3(\theta_4) \right\} = \left\{ \mathbf{A}_2^1 \right\}_2^{-1} \cdot \left\{ \mathbf{A}_1^0 \right\}_1^{-1} \cdot \mathbf{p}_W \quad (4.17)$$

$$\left\{ \mathbf{A}_4^3(\theta_4) \right\}_4 = \left\{ \mathbf{A}_3^2 \right\}_3^{-1} \cdot \left\{ \mathbf{A}_2^1 \right\}_2^{-1} \cdot \left\{ \mathbf{A}_1^0 \right\}_1^{-1} \cdot \mathbf{p}_W \quad (4.18)$$

Das equações (4. 15) à (4. 18) obtêm-se as seguintes expressões simplificadas:

$$p_{wx} = a_1 c_1 + a_2 c_{12} + a_3 c_{123} - a_3 c_1 s_{23} + d_4 (c_{13} s_2 + c_{12} s_3) \quad (4.19)$$

$$p_{wy} = a_1 s_1 + a_2 c_2 s_1 + a_3 c_{23} s_1 - a_3 s_{123} + d_4 (c_3 s_{12} + c_2 s_{13}) \quad (4.20)$$

$$p_{wz} = a_2 s_2 + a_3 c_2 s_3 + d_1 + a_3 c_3 s_2 + d_4 (-c_{23} + s_{23}) \quad (4.21)$$

$$c_1 p_{wx} + s_1 p_{wy} = d_4 (c_3 s_2 + c_2 s_3) + a_2 c_2 + a_3 c_{23} - a_3 s_{23} \quad (4.2)$$

$$p_{wz} = d_4 (-c_{23} + s_{23}) + a_2 s_2 + a_3 s_3 c_3 + a_3 c_3 s_3 \quad (4.23)$$

$$0 = s_1 p_{wx} - c_1 p_{wy} \quad (4.24)$$

$$a_1 c_2 + a_2 + a_2 c_2 + a_3 c_{23} - a_3 s_{23} + d_1 s_2 = c_{12} p_{wx} + c_2 s_1 p_{wy} - d_1 s_2 + s_2 p_{wz} \quad (4.25)$$

$$-a_1 s_2 + a_2 s_2 + a_3 c_3 s_2 + a_3 c_2 s_3 + d_1 c_2 = c_2 p_{wz} - c_1 s_2 p_{wx} - s_{12} p_{wy} \quad (4.26)$$

$$-a_3 + a_2 c_3 + a_3 (c_{23} - s_{23}) = p_{wz} (c_3 s_2 + c_2 s_3) + c_3 p_{wx} (c_{23} - s_{23}) + p_{wy} s_3 (c_{23} - s_{23}) \quad (4.27)$$

$$0 = -c_3 p_{wy} + p_{wx} s_3 \quad (4.28)$$

$$+a_2 s_3 + a_3 (c_3 s_2 + c_2 s_3) + d_4 = c_3 p_{wx} (c_3 s_2 + c_2 s_3) + p_{wy} s_3 (c_3 s_2 + c_2 s_3) + p_{wz} (-c_{23} + s_{23}) \quad (4.29)$$

As incógnitas do sistema  $q = (q_1, q_2, q_3, q_4, q_5, q_6)^T$  é inferior ao número de equações, logo o sistema tem solução. Para as primeiras três variáveis de posição:

$$\theta_1 = \text{atan2} \left( \frac{-p_{wx}}{p_{wy}} \right) + \text{atan2} \left[ \pm \left( \frac{(p_{wy}^2 + p_{wx}^2 - d_2^2)^{0.5}}{d_2} \right) \right] \quad (4.30)$$

$$\theta_2 = \text{atan2} \left[ \frac{(c_1 p_{wx} + s_1 p_{wy}) \cdot (-a_3 s_3 + d_4 c_3) - p_{wz} \cdot (a_2 + a_3 c_3 + d_4 s_3)}{(c_1 p_{wx} + s_1 p_{wy}) \cdot (a_2 + a_3 c_3 + d_4 s_3) + p_{wz} \cdot (-a_3 s_3 + d_4 c_3)} \right] \quad (4.31)$$

$$\theta_3 = \text{atan2} \left( \frac{d_4}{a_3} \right) + \text{atan2} \left[ \pm \left( \frac{(a_3^2 + d_4^2 - \lambda^2)^{0.5}}{\lambda} \right) \right] \quad (4.32)$$

A segunda parte da solução do problema da cinemática inversa considera a MTH  $T_6^0(\theta)$  da cinemática direta em função da posição e orientação da garra ou ferramenta:

$$T_6^0(\theta) = T_3^0(\theta) \cdot T_6^3(\theta) \quad (4.33)$$

$$(T_3^0)^{-1} \cdot T_6^0 = T_6^3 \quad (4.34)$$

$$\begin{bmatrix} c_1 c_{23} & s_1 c_{23} & -s_{23} & -a_2 c_3 - a_3 \\ -s_1 & c_1 & 0 & -d_2 \\ c_1 s_{23} & s_1 s_{23} & c_{23} & -a_2 s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 & d_6 c_4 s_5 \\ c_4 c_5 c_6 + s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & s_4 s_5 & d_6 c_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 & d_6 c_5 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para a solução do problema inverso de orientação igualam-se os três primeiro termos das terceiras colunas da anterior expressão, de onde se obtém as seguintes equações:

$$\theta_4 = a \tan 2 \left[ \pm \frac{s_1 a_x - c_1 a_y}{s_{23} a_z - c_1 c_{23} a_x - s_1 c_{23} a_y} \right] \quad (4.35)$$

$$\theta_5 = a \tan 2 \left\{ \pm \frac{\left[ (c_1 a_y - s_1 a_x)^2 + (c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} z_z)^2 \right]^{0.5}}{c_1 s_{23} a_x + s_1 s_{23} a_y + c_{23} a_z} \right\} \quad (4.36)$$

$$\theta_6 = a \tan 2 \left[ \pm \frac{(c_1 s_{23} s_x + s_1 s_{23} s_y + c_{23} s_z)}{-(c_1 s_{23} n_x + s_1 s_{23} n_y + c_{23} n_z)} \right] \quad (4.37)$$

Estas três últimas equações definem duas configurações cada uma delas para  $\theta_4$ ,  $\theta_5$  e  $\theta_6$ . Um robô com seis graus de liberdade ou seis eixos, pode admitir um máximo de oito soluções, quatro possíveis configurações para a posição e quatro soluções para a orientação.

### 4. 3. Quaternions: Números Hiper - complexos

Hamilton (1843), (1844) (1969) inventou os chamados números hiper - complexos de dimensão 4 conhecidos como Quaternions, que se usam em sistemas aeroespaciais, há mais

de quatro décadas, Chow (1992), e em robótica para calcular sua cinemática e dinâmica, segundo Kuipers (1999) e Barrientos, Peñin, Balaguer, e Aracil (1997).

Os Quaternions podem ser definidos como a soma  $q = q_0 + \mathbf{q}$  com a seguinte representação  $q = q_0 + iq_1 + jq_2 + kq_3$ . Seu conjugado é similar ao conjugado de um número complexo  $q^* = q_0 - iq_1 - jq_2 - kq_3$  que é igual ao inverso do Quaternion  $q^{-1} = q^*$ .

O produto de Quaternions é complicado e define-se como o produto especial fundamental que satisfaz as seguintes regras  $i^2 = j^2 = k^2 = -1 = ijk$ .

É possível relacionar um ângulo com um Quaternion, similar à associar um ângulo com uma matriz de rotação. Então, o Quaternion que define o operador é sempre um Quaternion normalizado ou Quaternion unitário  $q^2 + |\mathbf{q}|^2 = 1$ ; análogo à  $\cos^2 \theta + \sin^2 \theta = 1$  onde  $\theta$  é definido e satisfaz a restrição  $-\pi < \theta \leq \pi$ . O produto especial de Quaternions quando  $p = \cos \alpha + \mathbf{K} \sin \alpha$  e  $q = \cos \beta + \mathbf{K} \sin \beta$  se escreve:

$$pq = p_0q_0 - \mathbf{p} \circ \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} \quad (4.38)$$

O produto triplo de Quaternion  $q \circ \mathbf{v} \circ q^*$  interpreta-se como operadores Quaternions de rotação. Para aplicar os operadores à um vetor em  $R^3$  é necessário que o vetor  $\mathbf{v}$  seja um Quaternion puro:  $\mathbf{v} = 0 + \mathbf{v}$ . Para definir o Quaternion operador de rotação  $L_q$  associado com o Quaternion  $q$  e aplicado ao vetor  $\mathbf{v} \in R^3$  tem-se a seguinte expressão:

$$\mathbf{W} = L_q(\mathbf{v}) = q \circ \mathbf{v} \circ q^* = (q + \mathbf{q})(0 + \mathbf{v})(q_0 + \mathbf{q})$$

$$\mathbf{W} = (q_0^2 - |\mathbf{q}|^2)\mathbf{v} + 2(\mathbf{q} \circ \mathbf{v})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{v}) \quad (4.39)$$

O operador representa uma rotação em  $R^3$ , o eixo de rotação é a parte vetorial de  $q$  e o ângulo de rotação é duas vezes o ângulo associado com o Quaternion  $q$ .

A primeira propriedade algébrica do operador é sua linearidade  $L_q(k\mathbf{a} + \mathbf{b}) = kL_q(\mathbf{a}) + L_q(\mathbf{b})$  e a segunda propriedade algébrica é que sua Norma ou longitude não muda no operador Quaternion  $|L_q(\mathbf{v})| = |\mathbf{v}|$ .

Esta propriedade é necessária, se o operador é para descrever uma rotação. Os três termos da expressão podem representar-se em forma expandida com matrizes; como  $\mathbf{W} = q(\mathbf{v})q^* = \mathbf{Q}\mathbf{v}$ . Onde  $\mathbf{Q}$  é a matriz. Baseado nisso, estabelece-se o seguinte teorema:

“Para qualquer Quaternion  $q = q_0 + \mathbf{q} = \cos\theta + \mathbf{u}\sin\theta$  e para qualquer vetor  $\mathbf{v} \in R^3$  a ação do operador  $L_q(\mathbf{v}) = q\mathbf{v}q^*$  sobre  $\mathbf{v}$  pode interpretar-se geometricamente como a rotação do vetor  $\mathbf{v}$  um ângulo  $2\theta$  em torno de  $\mathbf{q}$  como eixo fixo de rotação.” Kuipers (1999)

#### 4. 4. Sistemas de Coordenadas com Quaternions

Os sistemas de coordenadas para desenvolver a cinemática com Quaternions, basa-se, no seguinte: (1) Sistema de coordenadas baseadas na metodologia de Denavit - Hartenberg da Figura 4. 3 para o robô IRB 1400, (2) parâmetros  $d_1, a_1, a_2, d_4, a_3$  e  $d_6$  da Tabela 4. 1 e (3) variáveis  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$  da Tabela 4. 1

Agora, os sistemas de coordenadas da Figura 4. 3 chamam-se sistemas de coordenadas objetivo e criam-se novos sistemas de coordenadas intermédios como se mostra na Figura 4. 5. A razão para criar os novos sistemas intermédios é porque um Quaternion pode representar um deslocamento e uma rotação. Mais de um deslocamento ou mais de uma rotação precisa-se de outro Quaternion. Para obter o novo sistema de coordenadas apresenta-se a seguinte metodologia:

**Passo 1:** Deslocamento de  $O_0$  uma distancia  $d_1$  ao longo de  $Z_0$  chega-se ao sistema intermediário  $O_1: O_0 \rightarrow O_1, \mathbf{T}(Z_0, d_1), \mathbf{Rot}(Z_0, \theta_1 = 0)$

**Passo 2:** Deslocamento de  $O_1$  uma distancia  $a_1$  ao longo de  $X_1$  e um ângulo de rotação de  $+90^\circ$  ao redor de  $X_1$  chega-se ao sistema objetivo  $O_2: O_1 \rightarrow O_2, \mathbf{T}(X_1, a_1), \mathbf{Rot}(X_1, +90^\circ)$

**Passo 3:** Deslocamento de  $O_2$  uma distancia  $a_2$  ao longo de  $Y_2$  e um ângulo de rotação de  $+90^\circ$  ao redor de  $Z_2$  chega-se ao sistema objetivo  $O_3: O_2 \rightarrow O_3, \mathbf{T}(Y_2, a_2), \mathbf{Rot}(Z_2, \theta_2 = 90)$

**Passo 4:** Deslocamento de  $O_3$  uma distancia  $a_3$  ao longo de  $X_3$  chega-se ao sistema intermediário  $O_4: O_3 \rightarrow O_4, \mathbf{T}(X_3, a_3), \mathbf{Rot}(Z_3, \theta_3 = 0)$

**Passo 5:** Deslocamento nulo de  $O_4$ , porque  $O_4$  e  $O_5$  são coincidentes, e com um ângulo de rotação de  $+90^\circ$  redor de  $X_4$  chega-se ao sistema objetivo  $O_5: O_4 \rightarrow O_5, \mathbf{T}(Nulo), \mathbf{Rot}(X_4, +90^\circ)$

**Passo 6:** Deslocamento de  $O_5$  uma distancia  $d_4$  ao longo de  $Z_5$  chega-se ao sistema intermediário  $O_6: O_5 \rightarrow O_6, \mathbf{T}(Z_5, d_4), \mathbf{Rot}(Z_5, \theta_4 = 0)$

**Passo 7:** Deslocamento nulo de  $O_6$ , porque  $O_6$  e  $O_7$  são coincidentes, e com um ângulo de rotação de  $-90^\circ$  ao redor de  $X_6$  chega-se ao sistema objetivo  $O_7: O_6 \rightarrow O_7, \mathbf{T}(Nulo), \mathbf{Rot}(X_6, -90^\circ)$

**Passo 8:** Deslocamento nulo de  $O_7$  e com um ângulo de rotação  $\theta_5 = 0$  ao redor de  $Z_7$  chega-se ao sistema intermediário  $O_8$ :  $O_7 \rightarrow O_8$ ,  $\mathbf{T}(Nulo)$ ,  $\mathbf{Rot}(Z_7, \theta_5 = 0)$ . Este Passo é incluído porque aparece a variável  $\theta_5$ ,

**Passo 9:** Deslocamento nulo de  $O_8$  e com um ângulo de rotação de  $+90^\circ$  ao redor de  $Z_8$  chega-se ao sistema objetivo  $O_9$ :  $O_8 \rightarrow O_9$ ,  $\mathbf{T}(Nulo)$ ,  $\mathbf{Rot}(X_8, +90^\circ)$

**Passo 10:** Deslocamento de  $O_9$  uma distancia  $d_6$  ao longo de  $Z_9$  chega-se ao sistema objetivo  $O_{10}$ :  $O_9 \rightarrow O_{10}$ ,  $\mathbf{T}(Z_9, d_6)$ ,  $\mathbf{Rot}(Z_9, \theta_6 = 0)$

Os deslocamentos  $\mathbf{T}$  definem os seguintes vetores de posição:  $\mathbf{r}_1 = 0,0,d_1$ ,  $\mathbf{r}_2 = a_1,0,0$ ,  $\mathbf{r}_3 = 0,a_3,0$ ,  $\mathbf{r}_4 = a_3,0,0$ ,  $\mathbf{r}_5 = 0,0,0$ ,  $\mathbf{r}_6 = 0,0,d_4$ ,  $\mathbf{r}_7 = 0,0,0$ ,  $\mathbf{r}_8 = 0,0,0$ ,  $\mathbf{r}_9 = 0,0,0$ ,  $\mathbf{r}_{10} = 0,0,d_6$

As rotações  $\mathbf{Rot}$  definem os seguintes Quaternions para  $\hat{C}_i = \text{Cos} \frac{\theta_i}{2}$  e  $\hat{S}_i = \text{Sen} \frac{\theta_i}{2}$ :

$$\begin{aligned} Q_1 &= (\hat{C}_1, 0, 0, \hat{S}_1) & Q_2 &= \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0 \right) & Q_3 &= (\hat{C}_3, 0, 0, \hat{S}_3) & Q_4 &= (\hat{C}_4, 0, 0, \hat{S}_4) \\ Q_5 &= \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0 \right) & Q_6 &= (\hat{C}_6, 0, 0, \hat{S}_6) & Q_7 &= \left( \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0, 0 \right) \\ Q_8 &= (\hat{C}_8, 0, 0, \hat{S}_8) & Q_9 &= \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0 \right) & Q_{10} &= (\hat{C}_{10}, 0, 0, \hat{S}_{10}) \end{aligned}$$

As operações com Quaternions precisam utilizar uma das seguintes alternativas:

**Alternativa 1.** Quando se aplica uma translação representada por o vetor  $\mathbf{r}$  seguida por uma rotação representada por um Quaternion  $Q$  à um sistema  $O_i(X_i, Y_i, Z_i)$  resulta um

novos sistema  $O_{i+1}(X_{i+1}, Y_{i+1}, Z_{i+1})$  tal que as coordenadas de um vetor de posição  $\mathbf{p}$ , no sistema  $O_i(X_i, Y_i, Z_i)$  conhecidas no sistema  $O_{i+1}(X_{i+1}, Y_{i+1}, Z_{i+1})$ , pode calcular-se com a seguinte expressão, segundo Kuipers (1999):

$$\mathbf{T} = (0, \mathbf{p}_{i-1}) = Q_i \circ (0, \mathbf{p}_i) \circ Q_i^* + (0, \mathbf{r}_i) \quad (4.40)$$

**Alternativa 2.** Quando se aplica uma rotação  $Q$  seguida por uma translação e uma distância vetorial  $\mathbf{r}$  ao sistema  $O_0(X_0, Y_0, Z_0)$  gera-se um novo sistema  $O_1(X_1, Y_1, Z_1)$  tal que as coordenadas de um vetor  $\mathbf{p}$ , no sistema  $O_0(X_0, Y_0, Z_0)$  conhecidas em  $O_1(X_1, Y_1, Z_1)$ , pode calcular-se com a seguinte expressão, segundo Kuipers (1999):

$$(0, \mathbf{p}_0) = Q \circ (0, \mathbf{p}_1 + \mathbf{r}_0) \circ Q^* \quad (4.41)$$

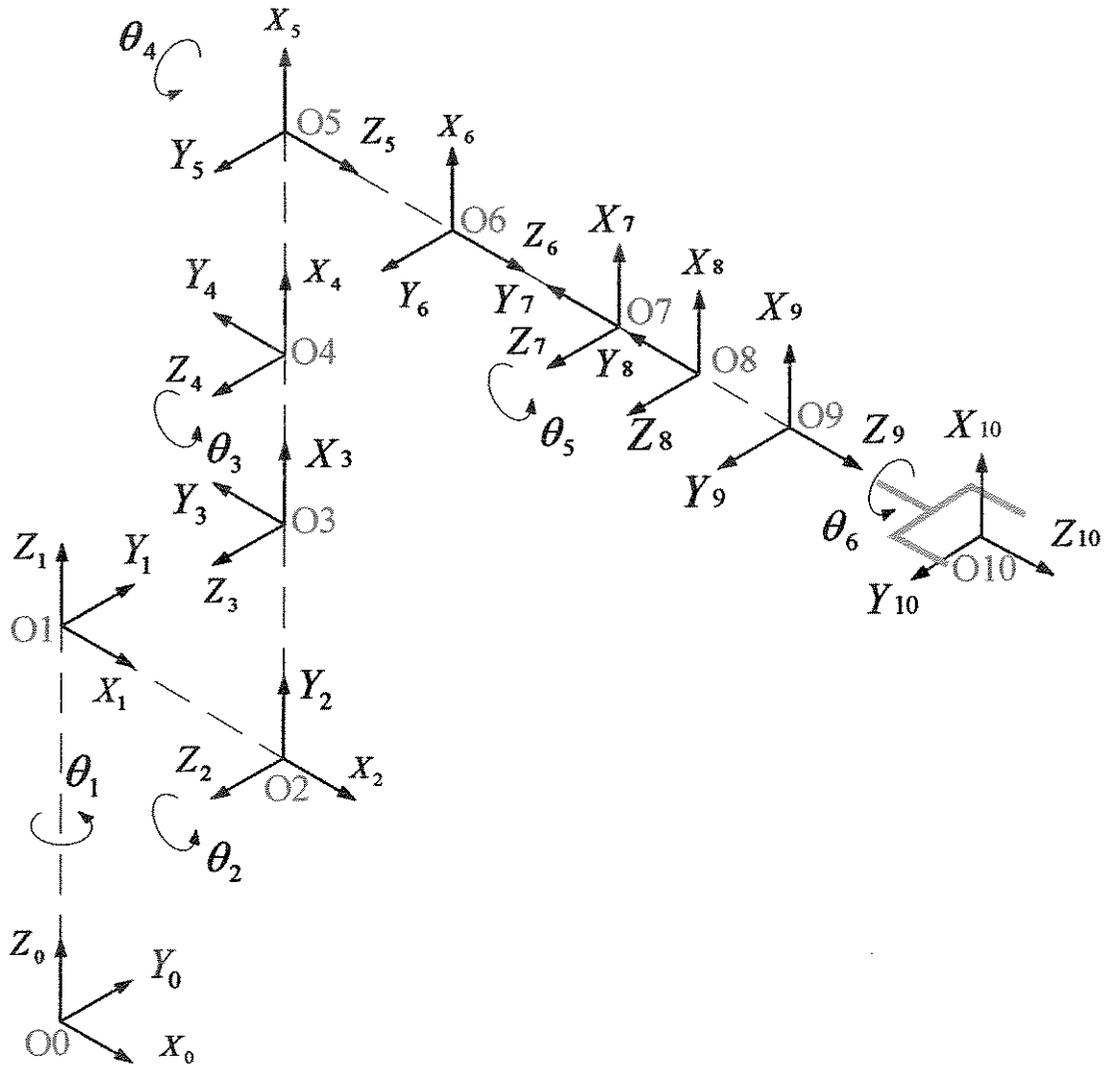


Figura 4. 5: Sistema de coordenadas do robô IRB 1400 para a modelagem com Quaternions

A rotação para as duas alternativas se expressa:

$$\text{Rot} = \mathbf{R}_{i-1} = \mathbf{Q}_i \mathbf{R}_i \quad (4. 42)$$

#### 4. 4. 1. Quaternions de Posição

Para o sistema de coordenadas da Figura 4. 7 aplica-se a alternativa 1 para calcular a posição e orientação da garra ou ferramenta do robô com respeito ao sistema de coordenadas da base. A posição pode ser calculada com a expressão (4. 40), com  $i = 1, 2, \dots, m$  e  $m = 10$ . Conseqüentemente, através de substituições sucessivas poderão ser criados conjuntos de dez equações, cuja resolução poderá ser obtida através do modelo matemático apresentado a seguir:

$$[P_0] = [Q]_{i=1}^m \circ [P_m] \circ [Q^*]_{i=1}^m + \sum_1^m (Q_1^{m-i} \circ r_m^i \circ Q^{*1}_{m-i}) \quad (4. 43)$$

$$P_0 = w + \sum_1^m W \quad (4. 44)$$

$[P_0]$ : Quaternion de posição do sistema de coordenadas da garra ou ferramenta com respeito ao sistema de coordenadas da base fixa.

$[Q]_{i=1}^m$ : Produto Quaternion de rotação de  $i = 1, 2, \dots, 10$  à  $m = 10$ .

$[P_m]$ : Quaternion de posição do sistema objetivo com respeito ao sistema  $m$ . Se o sistema objetivo coincide com o sistema  $m$ , então  $P = 0, 0, 0, 0$

$[Q^*]_{i=1}^m$ : Produto Quaternion conjugado de  $m = 10, 9, \dots, i$  quando  $i = 1$ .

$\sum_1^m W$ : Somatório produto do operador Quaternion.

Para  $w = [Q]_{i=1}^m \circ [P_m] \circ [Q^*]_{i=1}^m$  e  $\sum_1^m W = \sum_1^m (Q_1^{m-i} \circ r_m^i \circ Q^{*1}_{m-i})$

#### 4. 4. 2. Quaternions de Rotação

As relações entre os Quaternions que definem a rotação do sistema de coordenadas da garra ou ferramenta terminal  $O_{10}$  com respeito ao sistema de coordenadas da base  $O_0$  obtidas através da composição de Quaternions, tem a seguinte expressão geral:

$$\mathbf{R}_{i-1} = Q_i \mathbf{R}_i \quad (4. 45)$$

Se  $i = 1, 2, \dots, m$  e  $m = 10$  então, tem-se um conjunto de  $m$  equações e por substituições sucessivas, chega-se à:

$$\mathbf{R}_0 = Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 Q_7 Q_8 Q_9 Q_{10} \mathbf{R}_{10} = Q_{12345678910} \mathbf{R}_{10}$$

$$\mathbf{R}_0 = \sum_i^m Q_i \mathbf{R}_m \quad (4. 46)$$

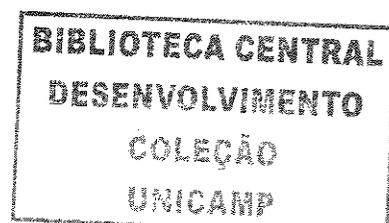
#### 4. 5. Modelagem da Cinemática com Quaternions

Para a modelagem da cinemática com Quaternions, tem os seguintes passos:

**Passo 1:** Com a sistemática de Denavit – Hartenberg obtém-se os sistemas de coordenadas objetivo das juntas.

**Passo 2:** Para os eixos de juntas consecutivas e paralelos precisa-se de um Quaternion para representar um deslocamento e uma rotação, que possa incluir a variável da junta.

**Passo 3:** Para eixos de juntas consecutivas que se cruzam ou que se curtam, se precisa adicionar um sistema de coordenadas de transição ou intermediário para incluir a variável da junta.



**Passo 4:** Adicionar aos sistemas de junta objetivo os sistemas transição necessários desde o sistema de coordenadas da base até o sistema de coordenadas da garra ou ferramenta.

**Passo 5:** Numerar todos os sistemas de coordenadas, desde a base até a garra ou ferramenta.

**Passo 6:** Definir o deslocamento e a rotação entre sistemas de coordenadas consecutivas, iniciando se desde o sistema de coordenadas da base.

**Passo 7:** Escrever os vetores de posição e os Quaternions de rotação, iniciando na base.

**Passo 8:** Escolher entre a alternativa 1 ou alternativa 2 para desenvolver o produto de Quaternions.

**Passo 9:** Apresentar resultados que descrevem a posição e a orientação da garra ou ferramenta com Quaternions.

#### **4. 6. De MTH a Quaternions e Vice-versa**

A transformação de matrizes de transformação homogêneas MTH a Quaternions e vice-versa, pode-se deduzir usando os sistemas de coordenadas intermédias ou de transição, os deslocamentos e as rotações. A seguir, se apresentam as relações finais, sem considerar o desenvolvimento, Hamilton (1969) e Beggs (1983).

A representação da matriz de transformação homogênea  $T$  em função dos componentes de um Quaternion, será dada através da seguinte matriz, também conhecida como relação direta:

$$\mathbf{T} = 2 \begin{bmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_3 q_0 & q_1 q_3 + q_2 q_0 & 0 \\ q_1 q_2 + q_3 q_0 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 + q_1 q_0 & 0 \\ q_1 q_3 + q_2 q_0 & q_2 q_3 + q_1 q_0 & q_0^2 + q_3^2 - \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.47)$$

A relação inversa pode obter-se igualando a trace e os elementos da diagonal principal da matriz (4. 58) com a matriz (4. 11):

$$\begin{aligned} q_0 &= \frac{1}{2} \sqrt{(n_x + o_y + a_z + 1)} & q_1 &= \frac{1}{2} \sqrt{(n_x - o_y - a_z + 1)} \\ q_2 &= \frac{1}{2} \sqrt{(-n_x + o_y - a_z + 1)} & q_3 &= \frac{1}{2} \sqrt{(-n_x - o_y + a_z + 1)} \end{aligned} \quad (4.48)$$

#### 4. 7. Cinemática Diferencial Direta

A cinemática diferencial direta ou a cinemática geométrica direta expressa a relação entre a velocidade da garra ou da ferramenta terminal, e as correspondentes velocidades subministradas por os atuadores das juntas do robô. Em operações de processamento, como soldagem de arco elétrico ou corte por jateamento de água, a ferramenta precisa movimentar-se com velocidade pré-determinada no espaço cartesiano, mas a movimentação acontece no espaço da junta, então, os atuadores subministram determinada velocidade para obter uma velocidade integrada resultante.

##### 4. 7. 1. Diferencial de Translação

Considere-se que a posição instantânea de um elemento do robô com respeito ao sistema de coordenadas da base é  $\mathbf{p}_0 = \mathbf{c}_i + \mathbf{p}_i$ ; onde:

$\mathbf{c}_i$  representa a posição do sistema de coordenadas do elemento  $O_i(X_i, Y_i, Z_i)$  em relação ao sistema de coordenadas da base  $O_0(X_0, Y_0, Z_0)$ ,

$\mathbf{p}_i$  representa a posição de um ponto P do elemento em relação ao sistema de coordenadas do elemento  $O_i(X_i, Y_i, Z_i)$  e

$\mathbf{p}_0$  é a resultante o posição de P em relação ao sistema de coordenadas da base  $O_0(X_0, Y_0, Z_0)$ .

Conseqüentemente, a translação do elemento do robô em relação ao sistema de coordenadas da base, se expressa:

$$\mathbf{p}_0 + d\mathbf{p}_0 = d\mathbf{c}_i + \mathbf{p}_i \quad (4.49)$$

$$\mathbf{p}_0 = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & b_i \\ 0 & 0 & 1 & c_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

$$d\mathbf{p}_0 = \begin{bmatrix} 0 & 0 & 0 & da_i \\ 0 & 0 & 0 & db_i \\ 0 & 0 & 0 & dc_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} da_i \\ db_i \\ dc_i \\ 1 \end{bmatrix}$$

#### 4.7.2. Diferencial de Rotação

Considere-se que a rotação instantânea de um elemento do robô respeito ao sistema de coordenadas da base é  $\omega_i dt = d\mathbf{k}_i = d\alpha_i \cdot \hat{\mathbf{i}}_0 + d\phi_i \cdot \hat{\mathbf{j}}_0 + d\theta_i \cdot \hat{\mathbf{k}}_0$  e sua representação com uma matriz anti - simétrica, se expressa:

$$d\mathbf{k}_i = \begin{bmatrix} 0 & -d\theta_i & -d\phi_i \\ d\theta_i & 0 & -d\alpha_i \\ -d\phi_i & d\alpha_i & 0 \end{bmatrix} \quad (4.50)$$

### 4. 7. 3. Diferencial de Translação + Diferencial de Rotação

A velocidade linear  $V_L$  e a velocidade angular  $V_A$  da garra ou ferramenta são funções das velocidades das juntas  $\dot{q}$ :

$$\dot{\mathbf{p}} = \mathbf{J}_p(q) \cdot \dot{\mathbf{q}} = V_L \quad (4. 51)$$

$$\omega = \mathbf{J}_0(q) \cdot \dot{\mathbf{q}} = V_A \quad (4. 52)$$

Na primeira equação  $\mathbf{J}_p$  é uma matriz 3xn relativa à contribuição das velocidades das juntas  $\dot{q}$  à velocidade linear da garra ou ferramenta  $V_L$ , de modo que na segunda equação  $\mathbf{J}_0$  seja uma matriz 3xn relativa à contribuição das velocidades das juntas  $\dot{q}$  ao velocidade angular da garra ou ferramenta  $V_A$ . Em forma compacta a velocidade da garra ou ferramenta se pode representar:

$$\mathbf{V} = \begin{bmatrix} \dot{\mathbf{p}} \\ \omega \end{bmatrix} = \mathbf{J}(q) \cdot \dot{\mathbf{q}} \quad \text{para} \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_0 \end{bmatrix} \quad (4. 53)$$

que é a equação da cinemática diferencial direta para calcular a velocidade da garra ou ferramenta terminal do robô, onde  $\mathbf{J}$  é a matriz Jacobiano.

### 4. 7. 4. A Matriz Jacobiana

Na Figura 4. 6 mostra-se a translação infinitesimal da garra ou ferramenta com um vetor  $d\mathbf{r}_6$  e a rotação infinitesimal da garra ou ferramenta com um vetor  $d\Phi_6$ ; definidos conforme sistema de coordenadas da base  $Oo(Xo, Yo, Zo)$  e se escreve de forma simplificada

$$\mathbf{V} = \mathbf{J}(q) \cdot \dot{\mathbf{q}} \quad \text{onde} \quad \mathbf{V} = \begin{bmatrix} V_L \\ V_A \end{bmatrix} \quad \text{e} \quad \dot{\mathbf{q}} = [\dot{q}_1, \quad \dot{q}_2, \quad \dot{q}_3, \dots, \quad \dot{q}_n]$$

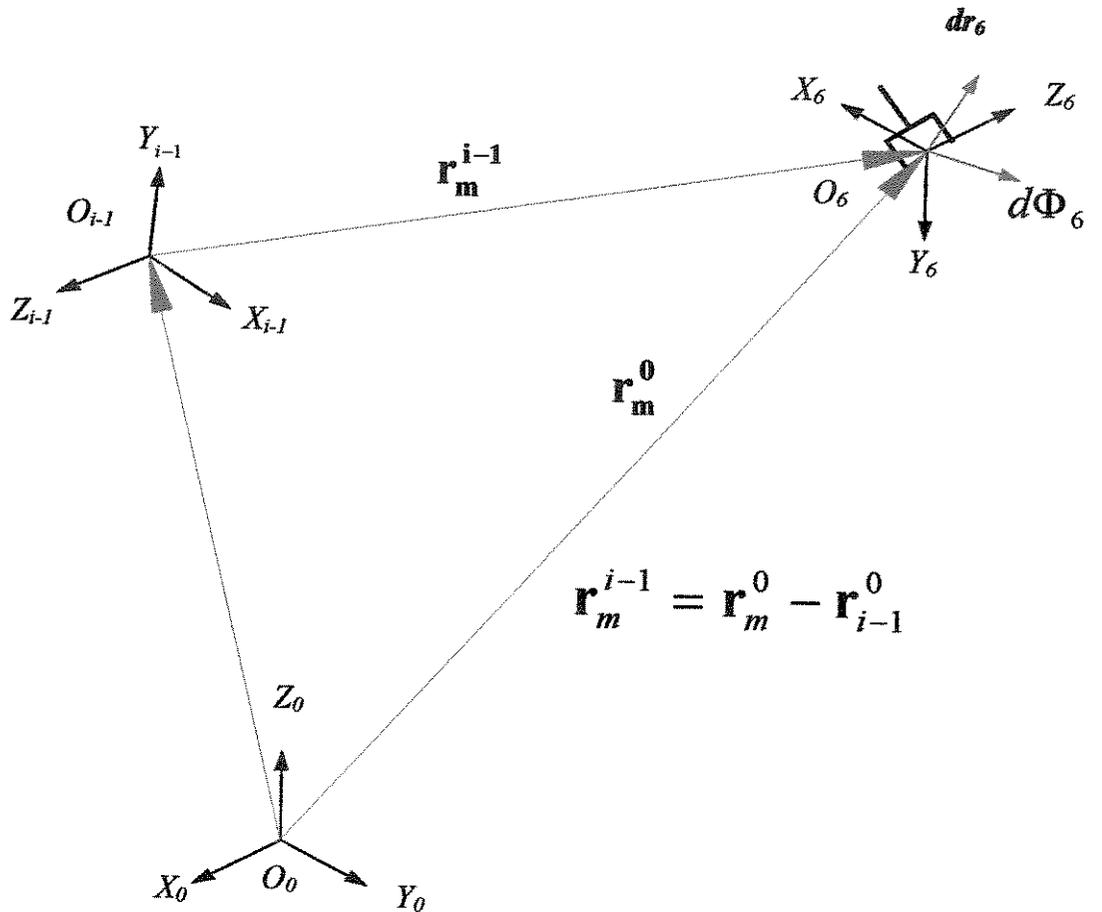


Figura 4. 6: Cinemática diferencial direta.

A dimensão da matriz  $\mathbf{J}$  é de  $6 \times 6$  onde as três primeiras filas representam a velocidade linear  $V_L$  e as três últimas linhas representam a velocidade angular  $V_A$ ; cada coluna representa a velocidade linear  $\mathbf{J}_{L_i}$  e a velocidade angular  $\mathbf{J}_{A_i}$  gerada por cada junta  $i$ . Então, se pode escrever para um robô com seis graus de liberdade:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{L1} & \mathbf{J}_{L2} & \mathbf{J}_{L3} & \mathbf{J}_{L4} & \mathbf{J}_{L5} & \mathbf{J}_{L6} \\ \mathbf{J}_{A1} & \mathbf{J}_{A2} & \mathbf{J}_{A3} & \mathbf{J}_{A4} & \mathbf{J}_{A5} & \mathbf{J}_{A6} \end{bmatrix} \quad (4.54)$$

A velocidade linear resultante  $V_L$  para a garra ou ferramenta se calcula com a seguinte expressão:

$$\mathbf{V}_L = \mathbf{J}_{L1} \cdot \dot{q}_1 + \mathbf{J}_{L2} \cdot \dot{q}_2 + \dots + \mathbf{J}_{Ln} \cdot \dot{q}_n \quad (4.55)$$

Se a  $i$ -ésima junta é prismática a velocidade será linear. Seja  $\mathbf{Z}_{i-1}$  o vetor unitário na direção característica da junta  $i$ , caso  $\dot{d}_i$  é a velocidade em essa direção, teremos:

$$\mathbf{J}_{Li} \cdot \dot{q}_i = \mathbf{Z}_{i-1} \cdot \dot{d}_i \quad (4.56)$$

Si a  $i$  ésima junta é de revolução, a velocidade angular efetiva de giro do elemento é:

$$\mathbf{J}_{Li} \cdot \dot{q}_i = \mathbf{Z}_{i-1} \cdot \dot{\theta}_i = \omega_i \quad (4.57)$$

Esta velocidade angular também gera uma velocidade linear na garra ou ferramenta. Seja  $\mathbf{r}_6^{i-1}$  o vetor posição desde  $O_{i-1}$  até  $O_6(X_6, Y_6, Z_6)$ . Então, a velocidade linear gerada através da velocidade angular é  $\mathbf{V} = \mathbf{r} \cdot \omega = \omega \cdot \mathbf{r}$ , ou seja:

$$\mathbf{J}_{Li} \cdot \dot{q}_i = \omega_i \cdot \mathbf{r}_6^{i-1} = (\mathbf{Z}_{i-1} \cdot \dot{\theta}_i) \cdot \mathbf{r}_6^{i-1} = (\mathbf{Z}_{i-1} * \mathbf{r}_6^{i-1}) \cdot \dot{\theta}_i \quad (4.58)$$

A velocidade angular da garra ou ferramenta se calcula com a seguinte expressão:

$$\mathbf{V}_A = \omega_6 = \mathbf{J}_{A1} \cdot \dot{q}_1 + \mathbf{J}_{A2} \cdot \dot{q}_2 + \dots + \mathbf{J}_{An} \cdot \dot{q}_n \quad (4.59)$$

Quando a junta é prismática não existe velocidade angular:

$$\mathbf{J}_{Ai} \cdot \dot{q}_i = \mathbf{Z}_{i-1} \cdot \omega_i = 0 \quad (4.60)$$

Quando a junta é de revolução, a velocidade angular induz uma velocidade angular na garra ou ferramenta que se calcula assim:

$$\mathbf{J}_{Ai} \cdot \dot{q}_i = \omega_i = \mathbf{Z}_{i-1} \cdot \dot{\theta}_i \quad (4.61)$$

As expressões (4. 55), (4. 57), (4. 59) e (4.61) permitem obter a matriz Jacobiano dum robô de n juntas:

$$\text{Para uma junta prismática tem-se a matriz Jacobiano: } \begin{bmatrix} \mathbf{J}_{Li} \\ \mathbf{J}_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{i-1} \\ \mathbf{0} \end{bmatrix} \quad (4. 62)$$

$$\text{Para uma junta de revolução tem-se a matriz Jacobiano: } \begin{bmatrix} \mathbf{J}_{Li} \\ \mathbf{J}_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{i-1} * \mathbf{r}_6^{i-1} \\ \mathbf{Z}_{i-1} \end{bmatrix} \quad (4. 63)$$

A direção do vetor  $\mathbf{Z}_{i-1}$  se representa por um vetor unitário  $\mathbf{Z}_{i-1}^{i-1} = [0 \ 0 \ 1]^T$  com respeito ao sistema de coordenadas  $\mathbf{O}_{i-1}(\mathbf{X}_{i-1}, \mathbf{Y}_{i-1}, \mathbf{Z}_{i-1})$  e em relação ao sistema de coordenadas da base. Com a seguinte expressão obtém-se a composição de matrizes de rotação  $\mathbf{R}_j^{i-1}(q_j)$ :

$$\mathbf{Z}_{i-1}^0 = \mathbf{R}_1^0(q_1) \cdot \mathbf{R}_2^1(q_2) \dots + \mathbf{R}_{i-1}^{i-2}(q_{i-1}) \cdot \mathbf{Z}_{i-1}^{i-1} \quad (4. 64)$$

O vetor posição  $\mathbf{r}_6^{i-1}$  em relação ao sistema de coordenadas da base, determina-se com MTH  $\mathbf{A}_j^{i-1}(q_j)$ . Seja  $\mathbf{r}_6^{i-1}$  o vetor aumentado 4x1 correspondente à  $\mathbf{r}_6^{i-1}$  e  $\mathbf{r}_{i-1}^{i-1} = [0 \ 0 \ 0 \ 1]^T$  o vetor posição aumentado correspondente ao sistema de coordenadas  $\mathbf{O}_{i-1}(\mathbf{X}_{i-1}, \mathbf{Y}_{i-1}, \mathbf{Z}_{i-1})$ . Então, pode escrever-se:

$$\mathbf{r}_6^{i-1} = \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \dots \mathbf{A}_n^{n-1}(q_n) \cdot \mathbf{r}_6^{n-1} - \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \dots \mathbf{A}_{i-1}^{i-2}(q_{i-1}) \cdot \mathbf{r}_{i-1}^{i-1} \quad (4. 65)$$

O primeiro termo do segundo membro da expressão, representam ao vetor posição de  $O_0$  à  $O_6$  e o segundo termo representam ao vetor posição intermediário de  $O_0$  à  $O_{i-1}$ .

#### 4. 7. 5. Modelagem da Matriz Jacobiano

Baseado nas MTH da modelagem da cinemática direta do IRB 1400 apresenta-se a seguinte metodologia para a modelagem da matriz Jacobiana:

**JUNTA 1:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 1$ :

$$\begin{bmatrix} \mathbf{J}_{L1} \\ \mathbf{J}_{A1} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_0 * \mathbf{r}_6^0 \\ \mathbf{Z}_0 \end{bmatrix}$$

Para  $\mathbf{Z}_0 = [0 \ 0 \ 1]^T$  e o vetor posição da garra ou ferramenta  $\mathbf{r}_6^0 = [p_x \ p_y \ p_z]^T$  cujos termos estão definidos na MTH (4. 7) desenvolve-se o produto vetorial para as expressões (4. 64) e (4. 65), assim:

$$\mathbf{Z}_0 * \mathbf{r}_6^0 = (0\hat{\mathbf{i}}_0 + 0\hat{\mathbf{j}}_0 + \hat{\mathbf{k}}_0) * (p_x\hat{\mathbf{i}}_0 + p_y\hat{\mathbf{j}}_0 + p_z\hat{\mathbf{k}}_0) = -p_y\hat{\mathbf{i}}_0 + p_x\hat{\mathbf{j}}_0$$

Escrito em forma compacta para a primeira coluna:

$$\mathbf{J}_{L1} = \begin{bmatrix} -p_y \\ p_x \\ 0 \end{bmatrix}, \quad \mathbf{J}_{A1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

**JUNTA 2:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 2$ :

$$\begin{bmatrix} \mathbf{J}_{L2} \\ \mathbf{J}_{A2} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_1 * \mathbf{r}_6^1 \\ \mathbf{Z}_1 \end{bmatrix}$$

Para  $O_1(X_1, Y_1, Z_1)$  se tem  $\mathbf{Z}_1^1 = [0 \ 0 \ 1]^T$ . Desenvolvendo o produto vetorial para as expressões (4. 64) e (4. 65) se tem:

$$\mathbf{Z}_1 = \mathbf{R}_1^0(q_1) * \mathbf{Z}_1^1 = \begin{bmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

$$\mathbf{r}_6^1 = \mathbf{r}_6^0 - \mathbf{r}_1^0 = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 0 \end{bmatrix} - \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ -1 \end{bmatrix}$$

$$\mathbf{Z}_1 * \mathbf{r}_6^1 = (s_1 \hat{\mathbf{i}}_1 - c_1 \hat{\mathbf{j}}_1 + 0 \hat{\mathbf{k}}_1) * (p_x \hat{\mathbf{i}}_1 + p_y \hat{\mathbf{j}}_1 + p_z \hat{\mathbf{k}}_1)$$

$$\mathbf{Z}_1 * \mathbf{r}_6^1 = -c_1 p_z \hat{\mathbf{i}}_1 + s_1 p_z \hat{\mathbf{j}}_1 + (s_1 p_y + c_1 p_x) \hat{\mathbf{k}}_1$$

Que poderá ser escrito em forma compacta para a segunda coluna:

$$\mathbf{J}_{L2} = \begin{bmatrix} -c_1 p_z \\ s_1 p_z \\ (s_1 p_y + c_1 p_x) \end{bmatrix}, \quad \mathbf{J}_{A2} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

**JUNTA 3:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 3$ :

$$\begin{bmatrix} \mathbf{J}_{L3} \\ \mathbf{J}_{A3} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_2 * \mathbf{r}_6^2 \\ \mathbf{Z}_2 \end{bmatrix}$$

Para  $O_2(X_2, Y_2, Z_2)$  se tem  $\mathbf{Z}_2^2 = [0 \ 0 \ 1]^T$ . Desenvolvendo o produto vetorial para as expressões (4. 64) e (4. 65) se tem:

$$\mathbf{Z}_2 = \mathbf{R}_1^0 * \mathbf{R}_2^1 * \mathbf{Z}_2^2 = \begin{bmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

$$\mathbf{r}_6^2 = \mathbf{r}_6^0 - \mathbf{r}_2^0 = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} - \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{r}_6^2 = \mathbf{r}_6^0 - \mathbf{r}_2^0 = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} - \begin{bmatrix} a_1 c_1 + a_2 c_1 c_2 \\ a_1 s_1 + a_2 s_1 c_2 \\ a_2 s_2 + d_1 \\ 1 \end{bmatrix}$$

$$\mathbf{Z}_2 * \mathbf{r}_6^2 = (s_1 \hat{\mathbf{i}}_2 - c_1 \hat{\mathbf{j}}_2 + 0 \hat{\mathbf{k}}_2) * [(p_x - a_1 c_1 - c_1 c_2 a_2) \hat{\mathbf{i}}_2 + (p_y - a_1 s_1 - s_1 c_2 a_2) \hat{\mathbf{j}}_2 + (p_z - a_2 s_2 - d_1) \hat{\mathbf{k}}_2]$$

Que poderá ser escrito em forma compacta para a terceira coluna:

$$\mathbf{J}_{L3} = \begin{bmatrix} -c_1(p_z - a_2 s_2 - d_1) \\ s_1(p_y - a_1 s_1 - a_2 s_1 c_2) \\ s_1 p_y + s_1(-a_1 s_1 - a_2 s_1 c_2) + c_1 p_x + c_1(-a_2 c_1 - a_2 c_1 c_2) \end{bmatrix}, \quad \mathbf{J}_{A3} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

**JUNTA 4:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 4$ :

$$\begin{bmatrix} \mathbf{J}_{L4} \\ \mathbf{J}_{A4} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_3 * \mathbf{r}_6^3 \\ \mathbf{Z}_3 \end{bmatrix}$$

Para  $O_3(X_3, Y_3, Z_3)$  se tem  $Z_3^3 = [0 \ 0 \ 1]^T$ . Desenvolvendo o produto vetorial para as expressões (4. 64) e (4. 65) se tem:

$$Z_3 = R_1^0 * R_2^1 * R_3^2 * Z_3^3 = \begin{bmatrix} T \\ R \\ Q \end{bmatrix}$$

$$r_6^3 = r_6^0 - r_3^0 = \begin{bmatrix} p_x - L \\ p_y - M \\ p_z - N \end{bmatrix}$$

Escrito em forma compacta:

$$J_4 = \begin{bmatrix} R(p_z - N) - Q(p_y - M) \\ T(p_z - N) - Q(p_x - L) \\ T(p_y - M) - R(p_x - L) \\ T \\ R \\ Q \end{bmatrix}$$

**JUNTA 5:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 5$ :

$$\begin{bmatrix} J_{15} \\ J_{A5} \end{bmatrix} = \begin{bmatrix} Z_4 * r_6^4 \\ Z_4 \end{bmatrix}$$

Para  $O_4(X_4, Y_4, Z_4)$  se tem  $Z_4^4 = [0 \ 0 \ 1]^T$ . Desenvolvendo o produto vetorial para as expressões (4. 64) e (4. 65) se tem:

$$Z_4 = R_1^0 * R_2^1 * R_3^2 * R_4^3 * Z_4^4 = \begin{bmatrix} I \\ J \\ K \end{bmatrix}$$

$$\mathbf{r}_6^4 = \mathbf{r}_6^0 - \mathbf{r}_4^0 \begin{bmatrix} p_x - G \\ p_y - H \\ p_z \end{bmatrix}$$

Que poderá ser escrito em forma compacta:

$$\mathbf{J}_5 = \begin{bmatrix} p_z J - K(p_y - H) \\ p_z I - K(p_x - G) \\ I(p_y - H) - J(p_x - G) \\ I \\ J \\ K \end{bmatrix}$$

**JUNTA 6:** A junta é de revolução, então se aplica a expressão (4. 63) para determinar a primeira coluna do Jacobiano quando  $i = 6$ :

$$\begin{bmatrix} \mathbf{J}_{L6} \\ \mathbf{J}_{A6} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_5 * \mathbf{r}_6^5 \\ \mathbf{Z}_5 \end{bmatrix}$$

Para  $O_5(X_5, Y_5, Z_5)$  se tem  $\mathbf{Z}_5^5 = [0 \ 0 \ 1]^T$ . Desenvolvendo o produto vetorial para as expressões (4. 64) e (4. 65) se tem:

$$\mathbf{Z}_5 = \mathbf{R}_1^0 * \mathbf{R}_2^1 * \mathbf{R}_3^2 * \mathbf{R}_4^3 * \mathbf{R}_5^4 * \mathbf{Z}_5^5$$

$$\mathbf{r}_6^5 = \mathbf{r}_6^0 - \mathbf{r}_5^0 = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 0 \end{bmatrix} - \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \cdot \mathbf{A}_3^2(q_3) \cdot \mathbf{A}_4^3(q_4) \cdot \mathbf{A}_5^4(q_5) \cdot \mathbf{r}_{i-1}^{i-1}$$

Efetuada substituições e escrito em forma compacta:

$$\mathbf{J}_6 = \begin{bmatrix} B(p_z - F) - C(p_y - E) \\ A(p_z - F) - C(p_x - D) \\ A(p_y - E) - B(p_x - D) \\ A \\ B \\ C \end{bmatrix}$$

Onde:

$$A = (s_1 s_4 s_5 + c_1 (c_2 (c_5 s_3 + c_3 c_4 s_5) - s_2 (-c_3 c_5 + c_4 s_3 s_5)))$$

$$B = (-c_1 s_4 s_5 + s_1 (c_2 (c_5 s_3 + c_3 c_4 s_5) - s_2 (-c_3 c_5 + c_4 s_3 s_5)))$$

$$C = (s_2 (c_5 s_3 + c_3 c_4 s_5) + c_2 (-c_3 c_5 + c_4 s_3 s_5))$$

$$D = (a_1 c_1 + c_1 (a_2 c_2 - (a_3 s_3 - c_4 d_4) s_2 + c_2 (a_3 c_3 + d_4 s_3)))$$

$$E = (a_1 c_1 + s_1 (a_2 c_2 - (a_3 s_3 - c_4 d_4) s_2 + c_2 (a_3 c_3 + d_4 s_3)))$$

$$F = (a_2 s_2 + d_1 c_2 (a_3 s_3 - c_3 d_4) + s_2 (a_3 c_3 + d_4 s_3))$$

$$G = (a_2 c_2 - (a_3 s_3 - c_3 d_4) s_2 + c_2 (a_3 c_3 + d_4 s_3))$$

$$H = (a_2 s_2 + c_2 (a_3 s_3 - c_3 d_4) + s_2 (a_3 c_3 + d_4 s_3))$$

$$I = ((c_4 s_1 + c_1 (-c_2 c_3 s_4 + s_2 s_3 s_4))$$

$$J = (-c_1 c_4 + s_1 (-c_2 c_3 s_4 + s_2 s_3 s_4))$$

$$K = (-c_3 s_2 s_4 - c_2 s_3 s_4)$$

$$L = (a_1 c_1 + c_1 (a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2))$$

$$M = (a_1 s_1 + s_1 (a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2))$$

$$N = (a_2 s_2 + a_3 s_3 c_2 + d_1 + a_3 c_3 s_2)$$

$$T = c_1 (c_3 s_2 + c_2 s_3)$$

$$R = s_1 (c_3 s_2 + c_2 s_3)$$

$$Q = (-c_2 c_3 + s_2 s_3)$$

#### 4. 8. Modelagem da Cinemática Diferencial Direta

Conhecidas as velocidades de cada junta, pode-se calcular a cinemática diferencial direta para a garra ou ferramenta do robô IRB 1400 com a seguinte expressão:

$$\mathbf{V} = \mathbf{J}(q) \cdot \dot{\mathbf{q}}$$

$$\begin{bmatrix} \mathbf{V}_X \\ \mathbf{V}_Y \\ \mathbf{V}_Z \\ \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{L1X} & \mathbf{J}_{L2X} & \mathbf{J}_{L3X} & \mathbf{J}_{L4X} & \mathbf{J}_{L5X} & \mathbf{J}_{L6X} \\ \mathbf{J}_{L1Y} & \mathbf{J}_{L2Y} & \mathbf{J}_{L3Y} & \mathbf{J}_{L4Y} & \mathbf{J}_{L5Y} & \mathbf{J}_{L6Y} \\ \mathbf{J}_{L1Z} & \mathbf{J}_{L2Z} & \mathbf{J}_{L3Z} & \mathbf{J}_{L4Z} & \mathbf{J}_{L5Z} & \mathbf{J}_{L6Z} \\ \mathbf{J}_{A1X} & \mathbf{J}_{A2X} & \mathbf{J}_{A3X} & \mathbf{J}_{A4X} & \mathbf{J}_{A5X} & \mathbf{J}_{A6X} \\ \mathbf{J}_{A1Y} & \mathbf{J}_{A2Y} & \mathbf{J}_{A3Y} & \mathbf{J}_{A4Y} & \mathbf{J}_{A5Y} & \mathbf{J}_{A6Y} \\ \mathbf{J}_{A1Z} & \mathbf{J}_{A2Z} & \mathbf{J}_{A3Z} & \mathbf{J}_{A4Z} & \mathbf{J}_{A5Z} & \mathbf{J}_{A6Z} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{bmatrix} \quad (4. 66)$$

A seguir a metodologia para a modelagem da cinemática diferencial direta:

**Passo 1:** Definir tipo de junta:

$$\text{Se a junta é prismática aplicar: } \begin{bmatrix} \mathbf{J}_{Li} \\ \mathbf{J}_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{i-1} \\ \mathbf{0} \end{bmatrix}$$

$$\text{Se a junta é de revolução aplicar: } \begin{bmatrix} \mathbf{J}_{Li} \\ \mathbf{J}_{Ai} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{i-1} * \mathbf{r}_6^{i-1} \\ \mathbf{Z}_{i-1} \end{bmatrix}$$

**Passo 2:** Calcular os vetores:

$$\mathbf{Z}_{i-1}^0 = \mathbf{R}_1^0(q_1) \cdot \mathbf{R}_2^1(q_2) \dots + \mathbf{R}_{i-1}^{i-2}(q_{i-1}) \cdot \mathbf{Z}_{i-1}^{i-1}$$

$$\mathbf{r}_6^{i-1} = \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \dots \mathbf{A}_n^{n-1}(q_n) \cdot \mathbf{r}_{ET}^{n-1} - \mathbf{A}_1^0(q_1) \cdot \mathbf{A}_2^1(q_2) \dots \mathbf{A}_{i-1}^{i-2}(q_{i-1}) \cdot \mathbf{r}_{i-1}^{i-1}$$

**Passo 3:** Calcular a operação vetorial quando a junta é de revolução:  $\mathbf{Z}_{i-1}^0 * \mathbf{r}_6^{i-1}$

**Passo 4:** Organizar o matriz Jacobiano com os termos do Passo 3:

$$\begin{bmatrix} \mathbf{J}_{L1X} & \mathbf{J}_{L2X} & \mathbf{J}_{L3X} & \mathbf{J}_{L4X} & \mathbf{J}_{L5X} & \mathbf{J}_{L6X} \\ \mathbf{J}_{L1Y} & \mathbf{J}_{L2Y} & \mathbf{J}_{L3Y} & \mathbf{J}_{L4Y} & \mathbf{J}_{L5Y} & \mathbf{J}_{L6Y} \\ \mathbf{J}_{L1Z} & \mathbf{J}_{L2Z} & \mathbf{J}_{L3Z} & \mathbf{J}_{L4Z} & \mathbf{J}_{L5Z} & \mathbf{J}_{L6Z} \\ \mathbf{J}_{A1X} & \mathbf{J}_{A2X} & \mathbf{J}_{A3X} & \mathbf{J}_{A4X} & \mathbf{J}_{A5X} & \mathbf{J}_{A6X} \\ \mathbf{J}_{A1Y} & \mathbf{J}_{A2Y} & \mathbf{J}_{A3Y} & \mathbf{J}_{A4Y} & \mathbf{J}_{A5Y} & \mathbf{J}_{A6Y} \\ \mathbf{J}_{A1Z} & \mathbf{J}_{A2Z} & \mathbf{J}_{A3Z} & \mathbf{J}_{A4Z} & \mathbf{J}_{A5Z} & \mathbf{J}_{A6Z} \end{bmatrix}$$

**Passo 5:** Calcular a cinemática diferencial direta com a expressão:

$$\mathbf{V} = \begin{bmatrix} V_L \\ V_A \end{bmatrix} = \mathbf{J}(q) \cdot \dot{\mathbf{q}} \text{ para } \dot{\mathbf{q}} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6)$$

#### 4. 9. Modelagem da Cinemática Diferencial Inversa

Com a expressão  $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{q}}$  se pode calcular a velocidade na garra ou ferramenta em função das velocidades das juntas. Para um robô com seis graus de liberdade, o Jacobiano é uma matriz quadrada de 6 x 6. Se a matriz Jacobiano não é singular, ou seja,  $|\mathbf{J}| \neq 0$  a matriz inversa  $\mathbf{J}^{-1}$  existe. Por tanto se tem:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{p}} \tag{4. 67}$$

Que é a expressão para calcular a velocidade nas juntas conhecida a velocidade na garra ou ferramenta.

O Jacobiano depende da configuração que o robô assuma e muda com as variáveis das juntas. Então, podem-se definir singularidades a um conjunto de pontos no espaço da junta onde o determinante Jacobiano seja igual a zero  $|\mathbf{J}| = 0$ . Considerando a inversa do Jacobiano, onde  $Adj \mathbf{J}$  é a matriz adjunta, se escreve:

$$\mathbf{J}^{-1} = \frac{Adj\mathbf{J}}{|\mathbf{J}|} \quad (4.68)$$

Quando a configuração do robô se aproxima a uma singularidade, alguns elementos de  $\mathbf{J}^{-1}$  assumem valores excessivamente grandes porque  $Adj\mathbf{J}$  sempre considera valores finitos. Baseado no anterior pode-se afirmar que as velocidades requeridas nas juntas para mover a garra ou ferramenta terminal em algumas direções, são excessivamente grandes, aproximando-se ao infinito quando  $|\mathbf{J}| = 0$ . Neste caso limite, se pode dizer que o robô assume uma configuração impossível porque a liberdade de movimento está restringida ou existem singularidades.

#### 4. 10. Modelagem da Cinemática da FMC

As janelas da Figura 4. 7 à Figura 4. 9 foram implementadas através do software Workspace<sup>TM</sup>. Quando se seleciona um robô de uma base de dados, disponível no software, necessita-se de um conhecimento das metodologias para o cálculo do modelo cinemático direto e cinemático inverso, a modelagem da cinemática diferencial direta e inversa; tendo em vista a necessidade de verificarmos se a modelagem cinemática existente neste software está em conformidade com o modelo calculado. Quando o software do modelo de robô selecionado não faz parte desta base de dados disponível neste software, então é necessário carregá-lo para obter bom desempenho do robô dentro da FMC.

A Figura 4. 7, apresenta uma tabela de velocidades e as acelerações de cada uma das juntas do robô IRB 1400 de ABB<sup>TM</sup>, obtida através do software Workspace<sup>TM</sup>. Uma possível alteração nos dados desta janela significa uma mudança no modelo cinemático direto do robô. Caso a modelagem da cinemática não esteja carregada no software é necessário inseri-la para que o robô possa ter um funcionamento controlável.

| Joint | Max Vel | Max Acc | Monitor |
|-------|---------|---------|---------|
| 1     | 120.00  | 670.00  | Yes     |
| 2     | 120.00  | 660.00  | Yes     |
| 3     | 120.00  | 1130.00 | Yes     |
| 4     | 280.00  | 3290.00 | Yes     |
| 5     | 280.00  | 3290.00 | Yes     |
| 6     | 280.00  | 3290.00 | Yes     |

Figura 4.7: Velocidade e aceleração nas juntas em Workspace.

A janela apresentada na Figura 4. 8 mostra as principais características das juntas quando se desenvolve o projeto para um novo robô. É também apresentada a característica cinemática das juntas para desenvolver as trajetórias dentro da FMC.

Figura 4. 8: Propriedades das juntas do IRB 1400 da ABB em Workspace.

Na janela da Figura 4. 9 apresentam-se os parâmetros de Denavit – Hartenberg de um grupo de robôs a ser selecionado. Quando se desenvolve um projeto de FMC com software Workspace<sup>TM</sup>, torna-se imprescindível uma consulta se a modelagem do robô selecionado, encontra-se na base de dados, caso contrario, é preciso carregar os parâmetros de Denavit – Hartenberg e a modelagem da cinemática do robô.

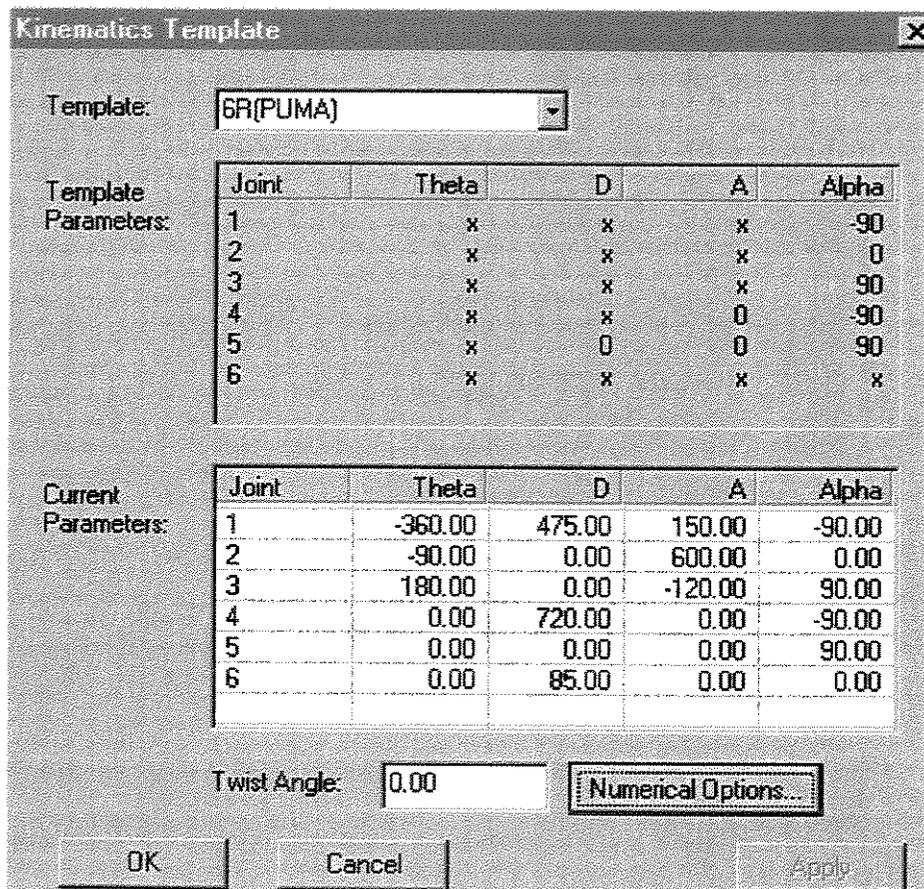


Figura 4. 9: Parâmetros de Denavit - Hartenberg do robô PUMA<sup>TM</sup> em Workspace<sup>TM</sup>.

O software Workspace<sup>TM</sup> somente tem um algoritmo de aproximação da cinemática para todos seus robôs na sua base de dados, então, necessita-se desenvolver a modelagem da cinemática do robô selecionado e inseri-lo no software para uma aplicação específica de simulação e programação *off-line* dentro de uma FMC.

## Capítulo 5

### Simulação e Programação *Off-line* de FMC

Após a realização da modelagem CAD de componentes mecatrônicos e integrados na FMC, e da cinemática dos robôs, a próxima etapa a ser realizada é a simulação e programação *off-line* necessárias para o desenvolvimento das trajetórias dos robôs e dos mecanismos com o nível de precisão normalmente requerido. As trajetórias podem ser interpoladas no espaço das juntas ou em relação a ferramenta terminal do robô (espaço cartesiano)

Quando as trajetórias do robô acontecem no espaço das juntas, as variáveis são conhecidas e o sistema de coordenadas da garra ou da ferramenta terminal são simples de serem calculados, aplicando um algoritmo com a cinemática direta. Entretanto, se as trajetórias acontecerem no espaço cartesiano, o processo é mais complexo de ser calculado, onde através do conhecimento do sistema de coordenadas da garra ou da ferramenta terminal, se aplica um algoritmo com a cinemática inversa para calcular a posição e orientação, em cada uma das juntas.

Existem muitos sistemas de simulação e programação *off-line*, os mais conhecidos são: GRASP, CimStation, IGRIP, Workspace, RobotStudio, EasyRob, Cosimir e todos têm em comum três ferramentas:

- (1) Visualização e a modelagem gráfica da célula de trabalho em sistema CAD.

(2) Planejamento e simulação das trajetórias dos robôs em sistema CAD.

(3) Tratamento de colisões.

Nesta pesquisa aplica-se o software Workspace<sup>TM</sup> como ferramenta de simulação e programação *off-line*. O software Workspace<sup>TM</sup> é um sistema de simulação sofisticado para robótica industrial que pode desenvolver as seguintes tarefas:

- 1) Importar dados CAD de outros sistemas CAD.
- 2) Modelagem CAD de células de trabalho.
- 3) Uso de robôs e ferramentas de sua base de dados.
- 4) Geração automática de trajetórias.
- 5) Edição gráfica de programas de robôs.
- 6) Otimização de ciclos de tempo e tratamento de colisões.
- 7) Geração de programação *off-line* de robôs industriais.
- 8) Descarregamento de programação *off-line* ao controlador do robô quando se necessita de um pós-processamento.

Workspace<sup>TM</sup> é um software que opera em ambiente Windows da Microsoft e usa o sistema ACIS CAD para criar alta qualidade, alta velocidade e imagens em 3D. Inclui o aplicativo visual Basic VBA (VBA Visual Basic Application) para simulação e sofisticação dos programas dos robôs e inclui também linguagens e apresentação gráfica da programação *off-line*.

Workspace<sup>TM</sup> utiliza seu próprio algoritmo planejador de trajetórias para todas as simulações de movimentação. Cada robô industrial possui um modelo cinemático único que precisa desenvolver-se e integrar-se como algoritmo num módulo do software de simulação e programação *off-line*. Esse módulo é conhecido como RRS (RRS Realistic Robot Simulation). Workspace<sup>TM</sup> aplica um RRS próprio que é uma aproximação de simulação de todos seus robôs, incluindo o robô IRB 1400 de ABB, ou quaisquer outro robô.

## **5. 1. Simulação**

A metodologia analítica de planejamento de trajetórias fundamenta a metodologia gráfica antes de realizar a simulação, que deverá ser implementada através dos seguintes passos:

- 1) Criar Pontos Geométricos PGs.
- 2) Interpolação das trajetórias.
- 3) Geração das trajetórias.
- 4) Detecção e evasão de colisões.
- 5) Ajuste e otimização das trajetórias.
- 6) Simulação e animação gráfica.

## **5. 2. PGs**

Os PGs definem uma determinada trajetória do sistema de coordenadas do sistema de agarre ou ferramenta terminal, designado geralmente pelo fabricante do robô como TCP (TCP Tool Center Point). Um PG pode ser visualizado nos sistema CAD como GP0001.

Os PGs podem pertencerem ou não a uma determinada trajetória, entretanto fazem parte do ambiente de trabalho do robô, sendo considerados como acessíveis quando se encontram dentro do espaço de trabalho do robô e não acessíveis quando se encontram fora do mesmo. O software Workspace<sup>TM</sup> apresenta varias opções para criar PGs no espaço de trabalho do robô, que deverão ser descritas detalhadamente a seguir.

### **5. 2. 1. Obtenção dos PGs a Partir das Coordenadas**

A partir do menu Criar, selecionar o comando GP, criar o GP e posicionar-o de acordo aos eixos X e Y, como é mostrado na Figura 5. 1.

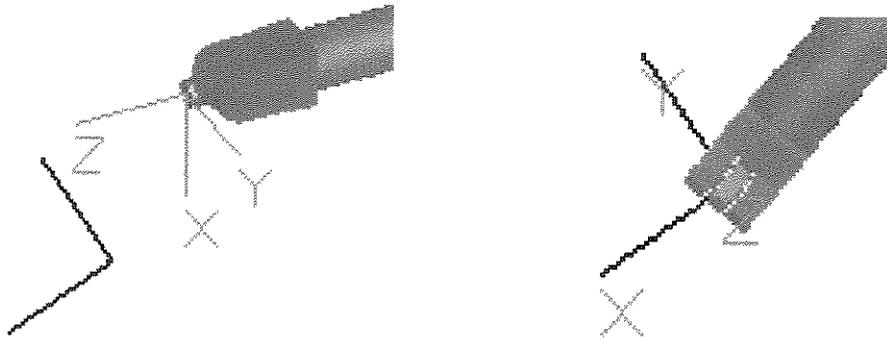


Figura 5. 1. Obtenção do PG por coordenadas

### 5. 2. 2. Criação de PGs Através da Aprendizagem

Aplicando o comando Learn GP poderão ser escritos os valores para cada junta do robô, caso seja necessário, utilizando o seguinte procedimento:

1) Selecionar Pendant e dentro da janela selecionar Learn GP, devemos escrever um valor para o novo PG chamado NewPG. Assim: Junta 1= 30, Junta 2= -20, Junta 3= -30

2) A seguir através do comando Learn GP. Do ponto de Simulação para o robô IRB 1400 no braço GP poderá ser lido: NewGP, NewGP001, NewGP002, NewGP003. O robô pode mover-se por os pontos criados e integrados em uma trajetória, como é mostrado na Figura 5. 2.

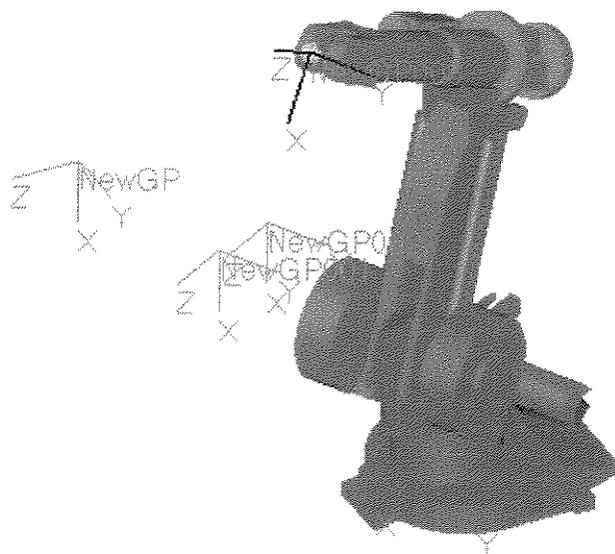


Figura 5. 2. Obtenção do PG através de aprendizagem

### 5. 2. 3. Obtenção do PGs em Arestas dos Objetos CAD

Para criarmos os PGs sobre as arestas dos objetos CAD, tem-se uma caixa dentro do espaço de trabalho do robô e do menu Edit selecionar comando Selection Mode e Edge. Sobre o modelo CAD devemos selecionar a aresta externa de 250 onde se posicionam os PGs.

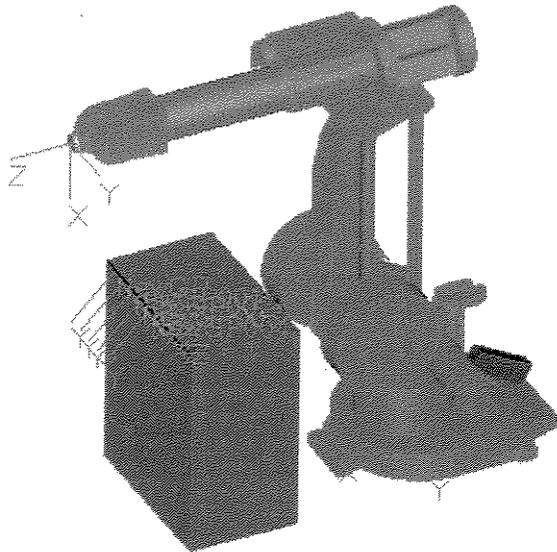


Figura 5. 3. PG em arestas

Após isso, deve-se selecionar o comando GP e Create GP e Relativa a Edges, escrevendo valores sucessivos de 50 depois do comando Create, até chegar ao extremo da aresta, criando: NewGP, NewGP001, NewGO002, NewGP003, NewGP004 correspondentes a 0, 50, 100, 150, 200, 250 na aresta selecionada, conforme mostra a Figura 5.3.

### 5. 2. 4. Obtenção do PGs em Interseção de Arestas

Para criarmos os PGs sobre as arestas das faces dos objetos CAD, tem-se uma caixa no espaço de trabalho do robô e do menu Edit selecionar o comando Selection Mode e Face. Sobre o modelo CAD selecionar a face superior do objeto e nas arestas do contorno da face criar os PGs.

Através de o menu selecionar, o comando GP e Create GP e na janela Relativa a Face, escrevem-se valores sucessivos para os parâmetros U e V da janela,  $U = 0$  e  $V = 0$ ,  $U = 1$  e  $V = 0$ ,  $U = 0$  e  $V = 1$ ,  $U = 1$  e  $V = 1$  para marcar quatro pontos correspondentes as quatro bordas. Com o comando Create da janela fixam-se os novos pontos: NewGP, NewGP001, NewGP002, NewGP003, como mostra a Figura 5. 4.

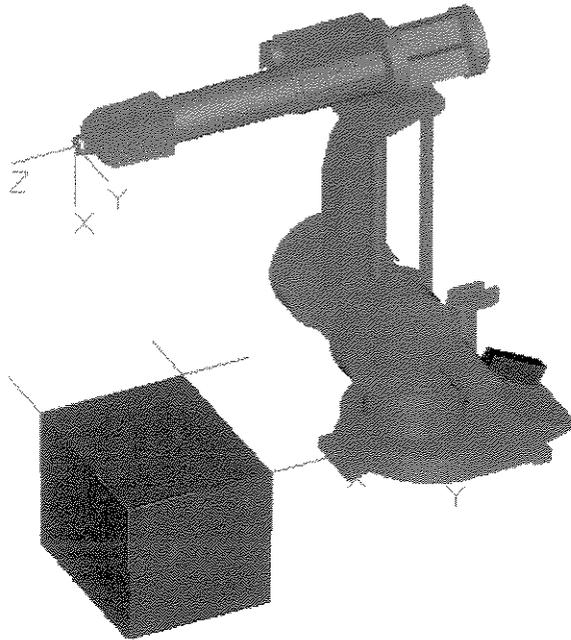


Figura 5. 4. Obtenção do PGs em intersecção de arestas

#### 5. 2. 5. Obtenção do PGs em Intersecção de Superfícies

O comando Section cria uma linha de intersecção comum a duas superficies de um objeto CAD. Através do menu Edit selecionar o comando Selection mode a Edge, com Ctrl-Shift-X o modelo CAD muda de rendering a shade e com Ctrl-Alt-E vice-versa.

Após selecionar a linha de intersecção e de Create selecionar GP e Create GP on section e têm-se as duas caras adjuntas. Da janela Relativa a Edge escrevem-se os valores em Distancia de arista de inicio: NewGP = 50, NewGP001 = 100, NewGP002 = 150, NewGP003 = 250, NewGP004 = 350 y NewGP005 = 500, que são PG sobre a linha de intersecção dos cilindros ortogonais, como se mostra na Figura 5. 5.

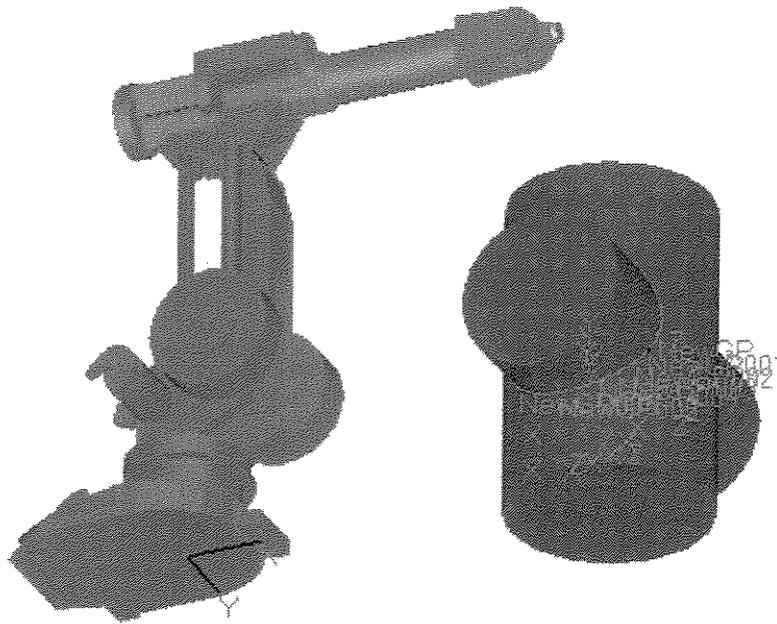


Figura 5. 5. Obtenção do PGs em intersecção de superfícies

### 5. 3. Interpolação das Trajetórias

As trajetórias especificam posições e orientações dos pontos de início e fim, e intermediários, sem importar a rota descrita pela garra ou ferramenta. Cada junta tem movimento independente e a trajetória resultante é complexa e imprevisível. O planejamento de trajetórias deverá incluir os seguintes passos:

- 1) Transformação dos pontos cartesianos inicial, final e intermediário à valores nas juntas, com ajuda da cinemática inversa.
- 2) Interpolação dos valores nas juntas para obter transições suaves entre pontos.
- 3) Movimentação independente em cada junta de acordo à trajetória planejada.

Na seção 2. 2. do capítulo 2, é apresentado uma revisão sobre o tratamento das trajetórias dos robôs. A descrição matemática das trajetórias, ajuda no entendimento do algoritmo que permite gera-las com sistemas de simulação e programação *off-line*. Nesta seção é apresentada a simbologia e equações desenvolvidas pelos autores Torres, Pomares, Gil, Puente, Aracil (2002) com conceitos desenvolvidos dentro desse projeto de pesquisa.

### 5. 3. 1. Interpolação com Pontos Início e Fim

Uma determinada trajetória considera somente os pontos iniciais e finais, conseqüentemente, obtêm-se trajetórias diversas, como se mostra na Figura 5. 6. Para conseguir movimentos suaves, quatro condições de contorno deverão ser impostas:

$$q(t_{início}) = q(0) = q_{início} \quad (5. 1)$$

$$q(t_{fim}) = q_{fim} \quad (5. 2)$$

$$\dot{q}(t_{início}) = \dot{q}(0) = 0 \quad (5. 3)$$

$$\dot{q}(t_{fim}) = 0 \quad (5. 4)$$

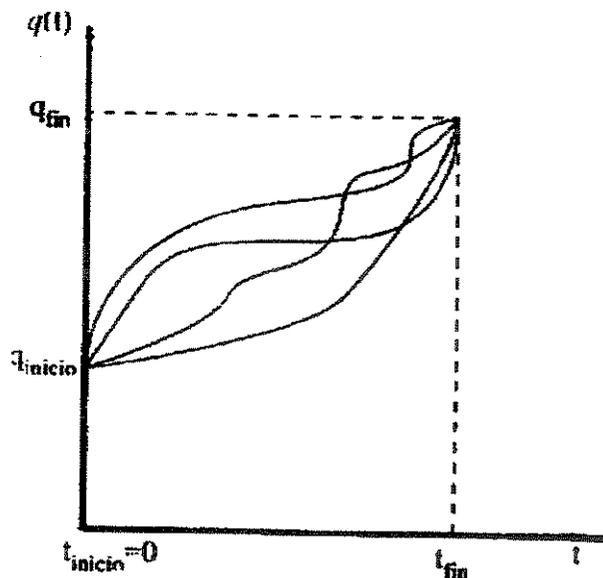


Figura 5. 6. Diferentes trajetórias para uma junta.

As duas primeiras condições deverão assegurar que a trajetória se inicia e se finaliza nos pontos definidos, e as duas últimas garantem que as velocidades inicial e final são nulas. Estabelecidas estas condições deveremos aplicar métodos de interpolação.

A seguir descreveremos alguns métodos de interpolação frequentemente utilizados em aplicações industriais.

### 5.3.1.1. Interpolação Cúbica.

As quatro expressões de contorno (5. 1), (5. 2), (5. 3) e (5. 4) podem-se satisfazer com um polinômio de terceiro grau, onde a equação que rege a trajetória é descrita por:

$$q(t) = a + b \cdot t + c \cdot t^2 + d \cdot t^3 \quad (5. 5)$$

onde:

$$a = q_{início}, \quad b = 0, \quad c = \frac{3(q_{fim} - q_{início})}{t_{fim}^2}, \quad d = \frac{-2(q_{fim} - q_{início})}{t_{fim}^3}$$

### 5.3.1.2. Interpolação Linear com Ajuste Parabólico

Um segundo método de interpolação muito utilizado industrialmente consiste na realização de uma interpolação linear entre os pontos  $q_{início}$  e  $q_{fim}$ . Através deste ajuste a velocidade permanece constante o que implica acelerações infinitas em  $t_{início}$  e  $t_{fim}$ .

Para resolver este problema, uma determinada trajetória é implementada em três trechos: dois trechos parabólicos e um trecho linear. O primeiro trecho parabólico, permite desenvolver uma velocidade na junta com aceleração constante, a seguir no trecho linear, o de maior movimentação do robô, a velocidade é mantida constante, e finalmente com o segundo trecho parabólico, se produz uma desaceleração constante até que a velocidade seja nula, justo em  $q_{fim}$ , como é apresentado na Figura 5. 7.

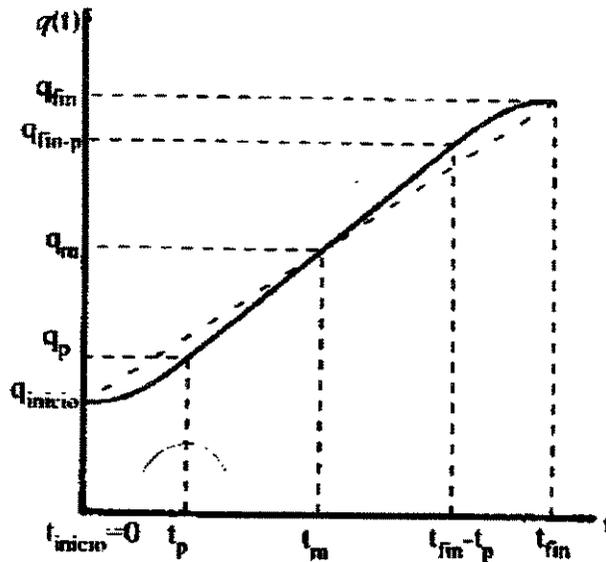


Figura 5. 7. Interpolação linear com ajuste parabólico.

Para simplificar os cálculos, considere-se de igual duração os trechos parabólicos  $t_p$  para que as soluções sejam simétricas com respeito a um ponto intermédio  $t_m$ . O primeiro trecho tem a seguinte equação:

$$q(t) = a + b \cdot t + c \cdot t^2 \quad (5.6)$$

onde:  $a = q_{início}$ ,  $b = 0$ ,  $c = \frac{\ddot{q}}{2}$ .

Logo, a equação do primeiro trecho parabólico é:

$$q(t) = q_{início} + \frac{\ddot{q}}{2} \cdot t^2 \quad (5.7)$$

O mesmo modelo matemático rege o trecho linear que tem velocidade constante e que corresponde ao mesmo valor da velocidade final do primeiro trecho parabólico, possui os seguintes valores para a, b e c:

$$a = q_{fim} + \ddot{q} \cdot t_{fim}^2 - \frac{\ddot{q}}{2} \cdot t_{fim}^2 = q_{fim} + \frac{\ddot{q}}{2} \cdot t_{fim}^2, \quad b = -\ddot{q} \cdot t_{fim}, \quad c = \frac{\ddot{q}}{2}$$

### 5. 3. 2. Interpolação de uma Trajetória Articular com Pontos Intermediários

Em alguns casos, além os pontos inicial e final da trajetória, se têm pontos intermediários.

#### 5. 3. 2. 1. Interpolação Cúbica

Neste caso a interpolação poderá ser utilizada substituindo as condições do contorno expressas em 5.3 e 5.4 por outras que expressem valores de velocidades das juntas, conhecidos para os pontos inicial e final da trajetória:

$$\dot{q}(t_{início}) = \dot{q}_{início} \quad (5. 8)$$

$$\dot{q}(t_{fim}) = \dot{q}_{fim} \quad (5. 9)$$

Com estas condições têm-se os seguintes valores para a, b, c, d :

$$a = q_{início},$$

$$b = \dot{q}_{início},$$

$$c = \frac{3(q_{fim} - q_{início})}{t_{fim}^2} - \frac{2 \cdot \dot{q}_{início} + \dot{q}_{fim}}{t_{fim}},$$

$$d = \frac{-2(q_{fim} - q_{início})}{t_{fim}^3} + \frac{\dot{q}_{fim} + \dot{q}_{início}}{t_{fim}^2}$$

### 5. 3. 2. 2. Interpolação Lineal com Ajuste Parabólico

Para gerar uma trajetória com pontos intermédios através desta interpolação, conforme mostra a Figura 5. 8, o procedimento muda em relação ao procedimento usado com pontos de inicio e fim, pelas novas condições de contorno de velocidade expressas nas equações 5.8 e 5.9. Agora os trechos parabólicos entre os pontos de passagem não podem ser simétricos, como acontece no caso anterior, porque as velocidades são diferentes em cada ponto de passagem.

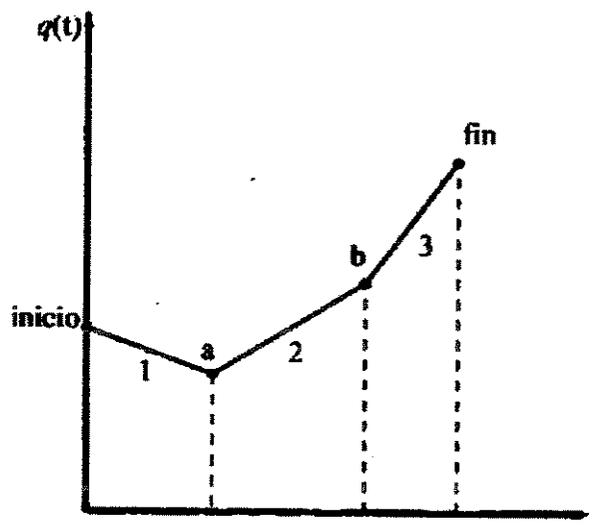


Figura 5. 8. Trajetória com pontos intermédios.

Na Figura 5. 9 mostra-se a forma geral de um interpolador linear com ajuste parabólico na junta e podemos observar que próximo aos pontos de passagem da função linear é substituída por uma função parabólica que ajusta a velocidade em dois trechos lineares contínuos.

O período de tempo de ajuste parabólico para o ponto de passo  $q_n$  chama-se  $t_{p_n}$ . O período de tempo entre os pontos de passo  $q_{n-1}$  e  $q_n$  chama-se  $t_{n-1,n}$ . A partir do conhecimento do tempo da trajetória entre os pontos  $q_{n-1}$  e  $q_n$  que é  $t_n - t_{n-1}$ , pode-se determinar que a duração do trecho linear entre os dois pontos se expressa através da relação:

$$t_{n-1,n} = (t_n - t_{n-1}) - 0.5t_{p_n} - 0.5t_{p_{n-1}} \quad (5.10)$$

Na Figura 5. 9 se representa com  $q_{n-1,n}$  a velocidade do trecho linear entre os pontos  $q_{n-1}$  e  $q_n$ . Esta velocidade determina-se com a seguinte expressão:

$$\dot{q}_{n-1,n} = \frac{q_n - q_{n-1}}{t_n - t_{n-1}} \quad (5.11)$$

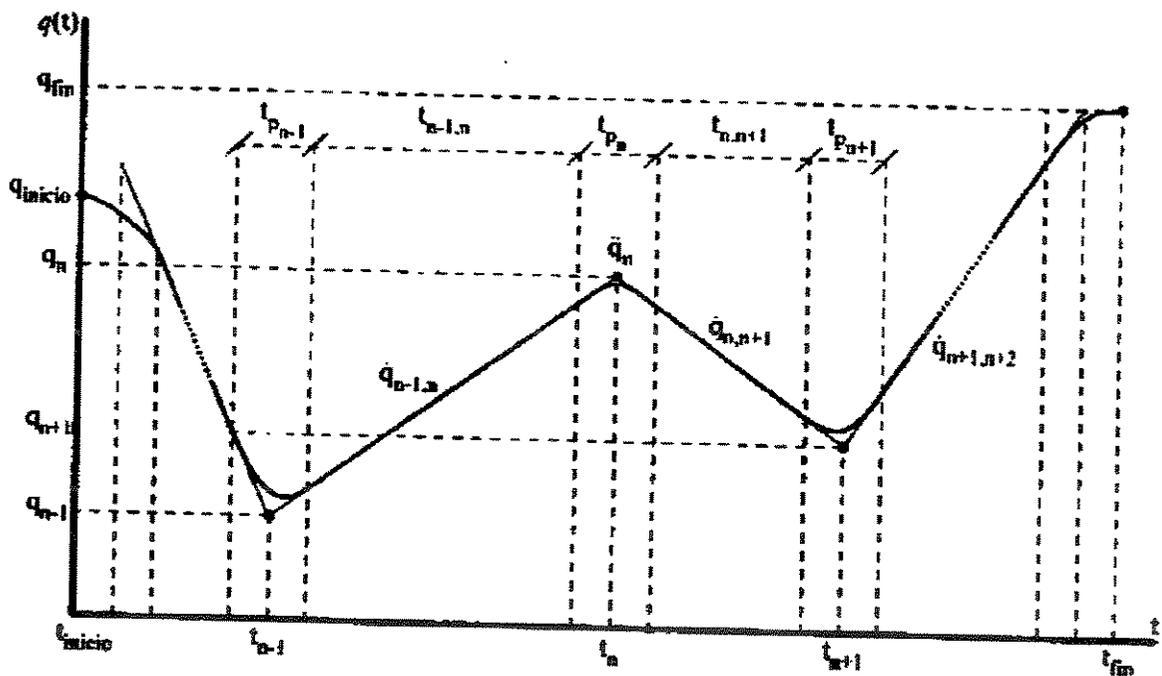


Figura 5. 9. Interpolação linear com ajuste parabólico com vários pontos intermédios

A informação requerida para gerar a interpolação deriva-se dos pontos de passagem e seu tempo. Para o ponto  $q_n$ , ou trecho parabólico, o tempo é  $t_{p_n}$  calcula-se com a seguinte expressão:

$$t_{p_n} = \frac{\dot{q}_{n,n+1} - \dot{q}_{n-1,n}}{\ddot{q}_n} \quad (5.12)$$

Os trechos inicial e final da trajetória são trechos parabólicos completos, e desta forma o tempo do trecho linear entre o primeiro e segundo ponto da trajetória é:

$$t_{1,2} = t_2 - t_{p_1} - 0.5t_{p_2} \quad (5.13)$$

E a velocidade pode-se calcular com a seguinte expressão:

$$\dot{q}_{1,2} = \frac{q_2 - q_1}{t_2 - 0.5t_{p-1}} \quad (5.14)$$

#### 5.4. Geração de Trajetórias

Uma trajetória é composta pela relação de todos os PGs que representam um determinado caminho para o robô, e podem ser representados graficamente como uma série de linhas que se conectam os PGs com orientação na direção do movimento.

No caso dos PGs pertencerem a uma determinada trajetória, então é um PG que se move, ou seja, o mesmo tem orientação e parâmetros de movimentação. Cada um dos PGs pertencentes a mesma trajetória poderão ser modificados ou movidos à uma nova posição ou uma nova trajetória quando são inacessíveis para o robô.

A geração das trajetórias pode-se desenvolver em forma manual ou em forma automática. No mesmo instante quando se criam os PGs, eles podem ser integrados a uma determinada trajetória, se assim é desejável ou depois podem criar-se as trajetórias de acordo com as necessidades de operação do robô.

Os PGs são entidades que podem fazer parte de mais de uma trajetória em programação *off-line* integrada. A trajetória no espaço da junta tem rota PTP e a trajetória no espaço cartesiano tem trajetória contínua linear TCL ou trajetória contínua circular TCC.

Numa trajetória PTP um robô se movimenta entre os PGs estabelecidos previamente, não importando como é descrita a trajetória. Por outro lado, numa trajetória contínua linear TCL, um robô se move passando através de todos os PGs estabelecidos, acarretando numa redução de velocidade, enquanto que na trajetória contínua circular TCC o robô descreve um movimento que inclui os PGs intermediários entre o PG inicial e o PG final.

A Figura 5. 10 é apresentado a TCL descrita pelo TCP do robô com os pontos inicial e final, numa determinada direção linear, entanto que na Figura 5. 11 mostra-se a TCC incluindo os PG intermediários por onde o TCP descreve uma trajetória circular.

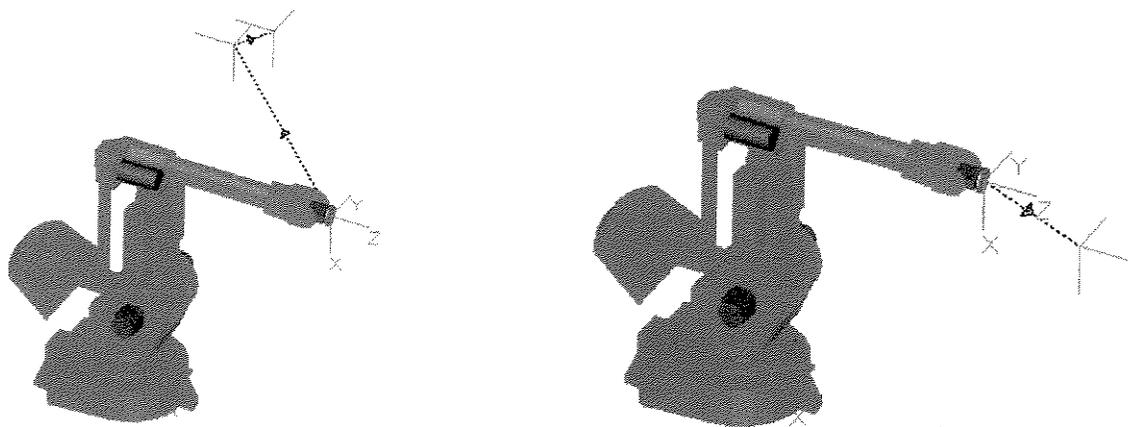


Figura 5. 10. Trajetória contínua linear TCL

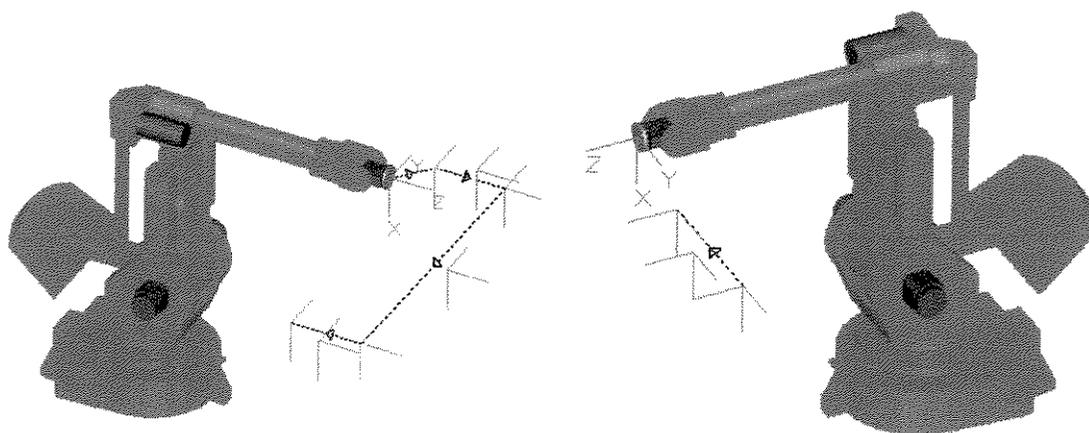
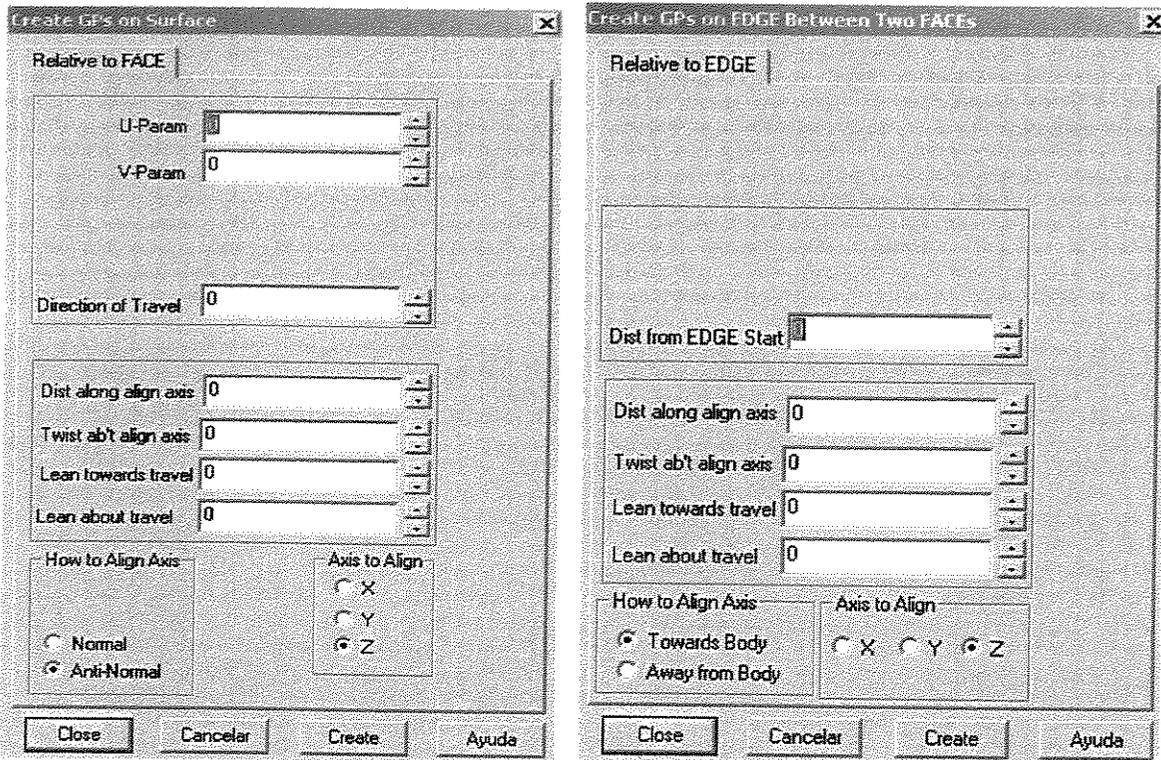


Figura 5. 11. Trajetória contínua circular TCC

Uma interpolação significa criar pequenos trechos de reta que possam ser programados e controlados pela unidade de controle do robô. Esta distância mínima entre PGs pode ser estabelecida matematicamente através de um algoritmo de interpolação ou graficamente, como mostra as telas apresentadas nas Figura 5. 12.



a) Interseção de arestas

b) Intersecção de superficies

Figura 5. 12. Algoritmo de interpolação PG.

Esta pesquisa não está encaminhada a desenvolver software para simulação e programação *off-line* de robôs e todas suas tarefas relacionadas, porque existem muitos de eles. Esta pesquisa está encaminhada a criar uma metodologia que permita utilizar com eficiência os softwares disponíveis no mercado e desenvolver sistemas flexíveis de manufatura.

Através das janelas apresentadas na Figuras 5. 12 uma metodologia gráfica da interpolação das trajetórias é aplicada, onde é considerado o robô IRB 1400™ desenvolvendo uma operação de soldagem elétrica em uma trajetória ao longo de uma aresta da face superior de uma peça prismática CAD, e um robô IRB 2400™ desenvolvendo uma operação de soldagem elétrica em uma trajetória ao longo de uma aresta da face superior de um cilindro CAD, e finalmente, o robô IRB 1500™ desenvolvendo uma operação de soldagem elétrica numa trajetória ao longo de uma aresta da face superior de uma peça prismática CAD. O seguinte procedimento foi utilizado para a sua implementação:

- (1) Criar FMC com os três robôs
- (2) Ativar robô com comando Set as active
- (3) Criar objeto CAD no espaço de trabalho do robô
- (4) Definir modo de Interpolação da trajetória Em Edit: Face ou Edge
- (5) Selecionar a cara ou a arista de trabalho
- (6) Criar PGs: Em Face ou Edge
- (7) Criar trajetória com Select Path e New
- (8) Nomear trajetória em Create Path
- (9) Fechar janela OK
- (10) Interpolação da trajetória em Relative to face ou Relative to Edge com valores para U e para V em uma trajetória seqüencial sobre o modelo CAD, onde é fundamental a orientação apropriada das coordenadas da ferramenta.
- (11) Fechar janela com Close
- (12) Sair e fim

Esta metodologia de criação de trajetórias em Mode Face e Mode Edge são apropriadas para a criação de TCL ou TCC. Através do comando Play Simulate pode-se visualizar cada uma das trajetórias criadas através da metodologia apresentada, como é mostrado na Figura 5. 13

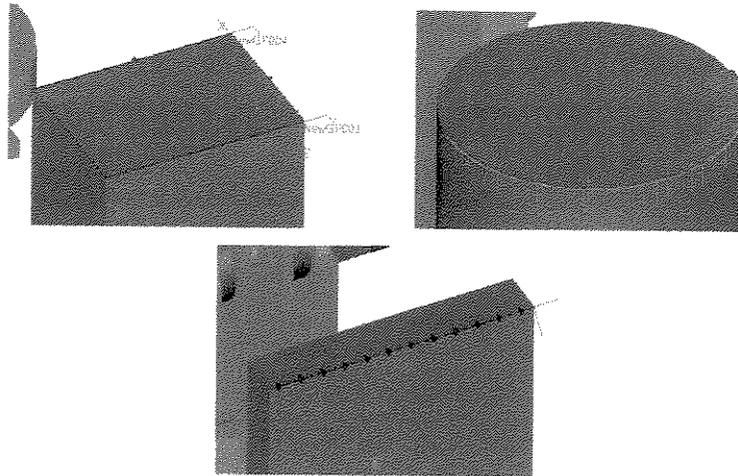


Figura 5. 13. Trajetórias com três tipos de PGs

### 5. 5. Detecção e Tratamento de Colisões

O planejamento e execução de trajetórias livres de obstáculos e colisões é uma função de interesse em robôs, exigida por requerimentos de segurança, funcionamento e confiabilidade. A detecção de colisões é baseada na informação da posição do robô, sua carga e os objetos CAD dentro de seu espaço de trabalho, sendo planificada em tempo real.

O software Workspace<sup>TM</sup> tem a função de detecção de colisões que opera com comando Collision Detection como se mostra na janela da Figura 5. 14: **Options, List e Report**. Da janela **Options** apresentam-se quatro possibilidades de opções:

- (1) Options.
- (2) Robots.
- (3) Near-miss.
- (4) Warnings.

A opção *Options* apresenta três métodos de detecção de colisões:

- (1) Detecção de colisões para todos os objetos CAD dentro da FMC.
- (2) Detecção de colisões baseados em uma lista específica.
- (3) Detecção de colisões baseado em o controle do espaço de um objeto CAD.

A opção *Robots* apresenta dois métodos de detecção de colisões:

- (1) Controle de auto colisão do robô.
- (2) Controle de colisões com objetos CAD.

A opção *Near-miss* permite definir uma distancia que é uma extensão da área de colisões para um objeto CAD, enquanto a opção *Warnings* apresenta três de métodos de prevenção:

- (1) Mudança da cor do objeto CAD de colisão.
- (2) Interrupção da movimentação do robô quando ocorre a colisão.
- (3) Alarme de colisão.

A opção *List* permite a criação de uma lista dos objetos CAD da FMC considerados na função de detecção de colisões; e finalmente, a opção *Report* apresenta a informação sobre os objetos CAD específicos na função de detecção de colisões que são armazenados sob a forma de relatório na execução da simulação.

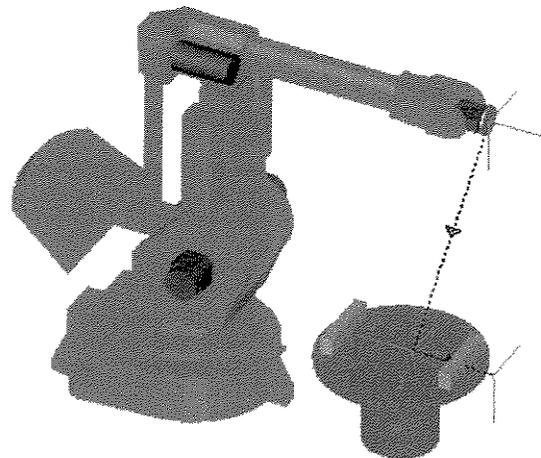
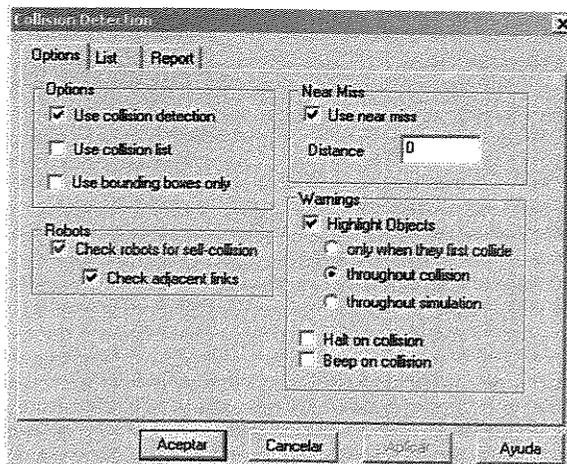


Figura 5. 14. Detecção de colisões.

A Figura 5. 14 apresenta um a FMC configurada com um robô IRB 6400, um mecanismo CAD rotativo e duas peças CAD prismáticas, onde são aplicadas as funções de detecção e tratamento de colisões.

## 5. 6. Ajuste e Otimização das Trajetórias

Uma vez aplicada à metodologia para obter a simulação *off-line* do robô dentro da FMC, podem-se encontrar imprecisões durante a movimentação do robô. Nesse caso é preciso fazer determinados ajustes e otimização das trajetórias do robô, sendo necessário, aplicarmos os seguintes pontos à metodologia desenvolvida para a simulação:

- (1) Re-configuração do Modelo CAD da FMC.
- (2) Criação de novos PG
- (3) Modificação de PG
- (4) Re-Interpolação das Trajetórias
- (5) Ajuste de velocidades.
- (6) Ajuste das Sequências.
- (7) Análise de Colisões.

## 5. 7. Simulação e Animação

A partir da geração dos PGs, e através deles as trajetórias, a próxima etapa é a simulação dos movimentos com o sistema de coordenadas do dispositivo de apreensão ou ferramenta terminal, que se move através de cada um dos PGs ao longo da trajetória. Se um PG é inacessível, a movimentação não deverá acontecer.

No planejamento de trajetórias para FMC, os parâmetros da simulação podem ser modificados ou personalizados. Os parâmetros da simulação encontram-se por default na primeira simulação, mas poderão ser alterados a critério do usuário.

A janela Simulate Options do software Workspace da Figura 5. 15 apresenta alguns dos parâmetros utilizados na simulação: *Visible simulation*, *Simulation log*, *Create path from track*, *Display GPs during simulation*, *Reset Model at end of Simulation*, *Single track debug mode*.

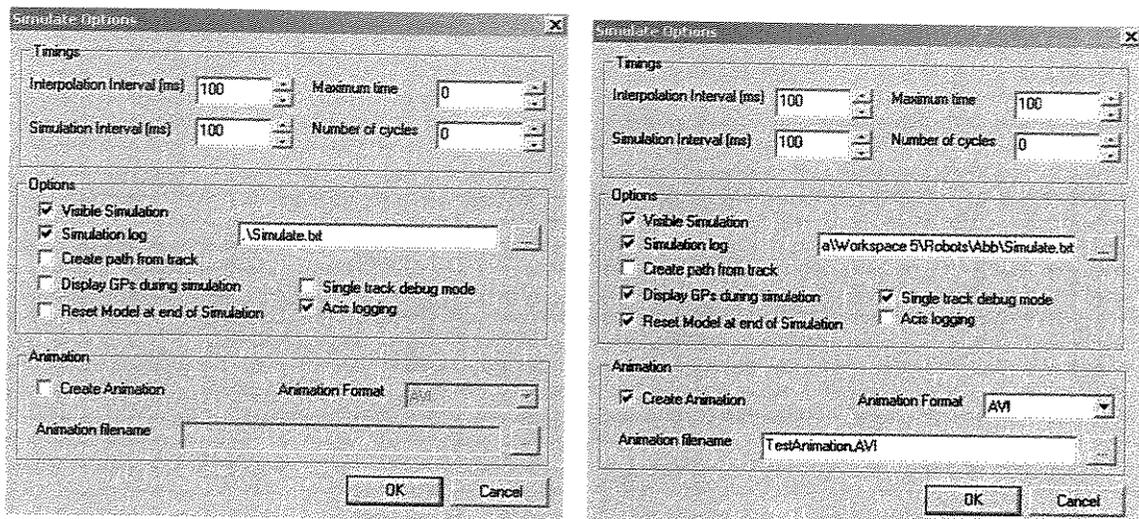


Figura 5. 15. Opções de simulação e animação

A simulação gráfica tem um ciclo de tempo em segundos, e através dela poderemos gerar uma simulação em «tempo real» através de um arquivo de animação (.AVI).

### 5. 8. Programação *Off-line*

A programação *off-line* de robôs é uma metodologia para desenvolver programas, parcial ou completamente, sem requerer o uso do robô. Ela inclui a geração de coordenadas, geração de funções e ciclos lógicos, segundo Nof (1985). A programação *on-line* de robôs com o *teaching box* é muito lenta e o tempo de programação aumenta com as complexidades das tarefas; então, o tempo útil do robô em operações de produção ou movimentação de materiais reduz-se significativamente ; conseqüentemente aumentam os custos de produção.

A programação *off-line* é apropriada em sistemas de produção de pequeno e médio tamanho. As vantagens da programação podem-se resumir assim:

- (1) Redução do tempo improdutivo do robô.
- (2) Proteção dos programadores dos ambientes perigosos.
- (3) Programação simples.
- (4) Integração com sistemas CAD / CAM.
- (5) Simplificação das tarefas complexas.

(6) Verificação dos programas.

Dois procedimentos podem aplicar-se para a programação *off-line* com Workspace :

- (1) Importar uma trajetória e programa do robô real para gerar a trajetória dentro da FMC virtual.
- (2) Exportar uma trajetória do robô da FMC virtual e gerar uma trajetória de robô com a linguagem de programação ao robô real.

### 5. 8. 1. Preparação do Robô, Linguagem e Célula

A metodologia de programação *off-line* tem os seguintes passos:

- (1) Seleção do robô.
- (2) Seleção da linguagem.
- (3) Seleção da trajetória.
- (4) Simulação.
- (5) Gerar programa automaticamente.
- (6) Editar programa.

É preciso ter a certeza que o robô é « perfeito » antes de proceder com a programação *off-line*, ou seja, definir que os parâmetros físicos do robô e os parâmetros funcionais são corretos. Existem dois caminhos para desenvolver programações *off-line*, que são descritos a seguir.

Quando a programação *off-line* do robô acontece no ambiente virtual para depois aplicá-la no ambiente real, desenvolvem-se as seguintes tarefas:

- (1) selecionar o robô de uma base de dados,
- (2) selecionar a linguagem de uma base de dados,
- (3) criar trajetórias,
- (4) simular e otimizar trajetórias,
- (5) desenvolver programação *off-line* e
- (6) exportar a programação *off-line* ao robô real.

Quando a programação do robô for realizada *on-line* e precisa-se da otimização antes de aplicar-la no ambiente real, é necessário o desenvolvimento das seguintes tarefas:

- (1) modelagem virtual do ambiente do robô,
- (2) calibração do robô real com o robô virtual,
- (3) importar trajetórias ao ambiente virtual,
- (4) editar trajetórias,
- (5) ajuste e otimização das trajetórias,
- (6) programação *off-line*, e
- (7) implantar programação *off-line* no ambiente real.

### **5. 8. 2. Tipos de Programação *Off-line***

A dificuldade da programação *off-line* POL aumenta com o número de robôs, o número de mecanismos e o número de graus de liberdade da FMC, assim mesmo, com a integração da programação *off-line* de todos os dispositivos mecatrônicos da FMC. Podem-se encontrar as seguintes situações para uma FMC com um robô, Figura 5. 16:

- (1) Robô Fixo e sem Mecanismo.
- (2) Robô com Deslocamento e sem Mecanismo.
- (3) Robô Fixo e Eixo Auxiliar para o Mecanismo.
- (4) Robô com Deslocamento e Eixo Auxiliar para o Mecanismo.
- (5) Robô Fixo e Mecanismo independente.
- (6) Robô com deslocamento e Mecanismo Independente.
- (7) Robô com Deslocamento e Eixo Auxiliar e Mecanismo Independente.

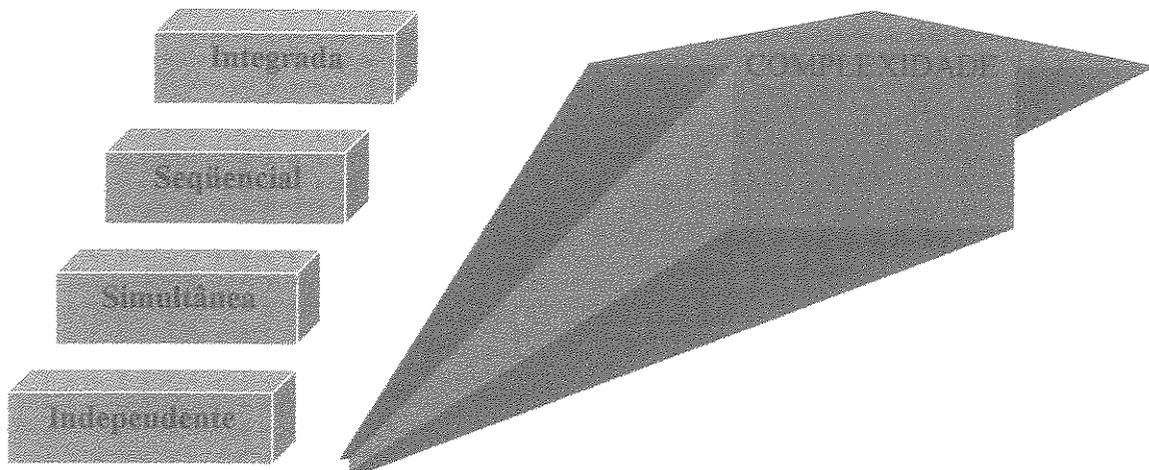


Figura 5. 16. Complexidade na programação *off-line*.

Pode-se identificar as seguintes opções de programação *off-line* na FMC:

- (1) Movimento Independente: Para um robô ou um mecanismo.
- (2) Movimento Simultâneo: Dois ou mais robôs se movem no mesmo tempo.
- (3) Movimento Sequencial: Os robôs se movem um depois de outro ou em serie.
- (4) Movimento Integrado: Os movimentos dos robôs e dos mecanismos são misturados e se podem integrar por PG e por trajetórias.

A FMC é um sistema mecatrônico complexo por a variedade de componentes integrados. Esta complexidade influi na programação *off-line*. Uma ferramenta útil na programação *off-line* integrada são sinais digitais porque permitem integrar programas ou PGs das trajetórias, desenvolvendo as seguintes tarefas:

- (1) Criar trajetórias para cada um dos componentes da FMC.
- (1) Configuração de sinais digitais entre trajetórias.
- (2) Conexão de sinais digitais entre PGs das trajetórias.
- (3) Definir 1 e 0 para ativar – desativar ou enviar – receber.
- (3) Implementação de sinais digitais.
- (4) Re - inicialização para cada sinal digital.

## 5. 9. Metodologia da Simulação e Programação *Off-line* das FMCs

Ao iniciar um projeto de célula de manufatura flexível FMC duas opções podem ser apresentadas:

- (1) Re - engenharia de antiga fábrica para integrar a FMC ou
- (2) Criação de uma nova FMC.

Em cada um dos casos pode-se desenvolver a modelagem da FMC com PN para avaliar seu funcionamento. O diagnóstico pode incluir:

- (1) O Processo produtivo: seqüências, trajetórias, rotas.
- (2) O Sistema Produtivo: tecnologias envolvidas e equipamento auxiliar.
- (3) Revisão de funções de fábrica para a sua integração. Isto serve para avaliar o estado das tecnologias e recomendação de seu emprego e atualização de novos equipamentos, e também para análise do processo de produção e ajuste.

Nesse projeto de pesquisa apresenta-se uma metodologia da modelagem, simulação e programação *off-line* de robôs e outros dispositivos mecatrônicos, encaminhada à configuração de FMC. A seguir, são identificadas as tarefas para o desenvolvimento da metodologia:

### 5.9.1 Modelagem CAD de Componentes Mecatrônicos

- |                 |  |
|-----------------|--|
| <b>Passo 1.</b> | Das peças de produção ou CAD / CAM das peças de produção     |
|                 | 1. 1. Prismáticas e / ou cilíndricas                         |
| <b>Passo 2.</b> | Do robô  |
|                 | 2. 1. Definir a configuração: Projeto ou seleção             |
|                 | 2. 2. Definir a garra ou a ferramenta: Projeto ou seleção    |
|                 | 2. 3. Calcular o espaço de trabalho                          |
| <b>Passo 3.</b> | Dos mecanismos   |
|                 | 3. 1. Definir graus de liberdade                             |
|                 | 3. 2. Definir a movimentação dos eixos: Linear ou rotacional |

**Passo 4.** Integração de componentes mecatrônicos

4. 1. Configurar a FMC
4. 2. Definir o tamanho da FMC

### **5.9.2 Modelagem da Cinemática do Robô**

**Passo 5.** Calcular a cinemática direta e inversa:

5. 1. Com MTH e / ou com Quaternions

**Passo 6.** Calcular a cinemática diferencial direta e inversa:

6. 1. Com MTH

### **5.9.3 Simulação**

**Passo 7.** Definição de PGs

**Passo 8.** Planejamento de trajetórias

**Passo 9.** Interpolação de trajetórias

**Passo 10.** Geração de trajetórias

**Passo 11.** Simulação e / ou animação

### **5.9.4 Programação *Off-line***

**Passo 12.** Seleção da linguagem

**Passo 13.** Edição da programação

## Capítulo 6

### Validação de Resultados

#### Simulação e Programação *Off-Line* de FMCs

Neste capítulo são apresentados exemplos de aplicação e validação da metodologia proposta para modelagem, simulação e programação *off-line* de robôs integrados em FMC. São casos reais da indústria internacional, estudados pelos pesquisadores, sem modelagem virtual da FMC, sem simulação gráfica e sem programação *off-line*.

Este trabalho considera o processo de manufatura desenvolvido e o equipamento utilizado nos estudos pelos pesquisadores, e a partir destas condições aplica-se à metodologia para modelagem, simulação e programação *off-line* da FMC, com novas condições de funcionamento.

Os exemplos de validação foram implementados utilizando a infra-estrutura existente no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP, tais como:

**Exemplo 1:** FMC com dois robôs de soldagem e mecanismo MB - MR.

**Exemplo 2:** FMC com um robô de soldagem, um robô de posicionamento e mecanismo de rotação.

**Exemplo 3:** FAC com dois robôs de montagem.

**Exemplo 4:** FMC com um robô de posicionamento e duas máquinas CNC utilizando três variantes.

**Exemplo 5:** FMS com um robô de posicionamento e cinco máquinas CNC.

O exemplo 1 proposto para validação foi implementado detalhadamente com o objetivo de para mostrar passo a passo a aplicação da metodologia proposta nesta pesquisa para simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados para configurar FMC. Os quatro exemplos seguintes são desenvolvidos utilizando a mesma metodologia, apresentando as principais vantagens da aplicação desta metodologia em estações robotizadas de linhas de produção (exemplo2), sistemas de montagem robotizadas (exemplo 3), flexibilidade na criação das novas FMC (exemplo 4) e conversão de sistemas produtivos robotizados em FMS.

#### **6. 1. Exemplo1: FMC com dois robôs de soldagem e mecanismo MB e MR**

FMC com quinze graus de liberdade configurada com os seguintes elementos:

- (1) mecanismo basculante MB integrado ao mecanismo rotativo MR com três graus de liberdade PRR,
- (2) um robô IRB 1400 para soldagem por arco elétrico de seis graus de liberdade RRR.RRR e
- (3) um robô IRB 1500 para soldagem por arco elétrico com seis graus de liberdade RRR.RRR.

Esta configuração foi proposta para o desenvolvimento da FMC no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP. Para validação de seu funcionamento, a seguir apresentam-se as modelagens da FMC em PN para:

- (1) Programação *off-line* independente.
- (2) Programação *off-line* simultânea.
- (3) Programação *off-line* seqüencial.
- (4) Programação *off-line* integrada.

### 6. 1. 1. PN para FMC com Programação *Off-line* Independente

FMC configurada com robô industrial IRB 1400 para desenvolver operações de soldagem de arco elétrico e dispositivo mecatrônico com três graus de liberdade que transporta uma peça da estação de carregamento / descarregamento à estação de trabalho do robô, depois da operação de soldagem transporta-se à peça à estação de carregamento / descarregamento, conforme é apresentado na Figura 6. 1.

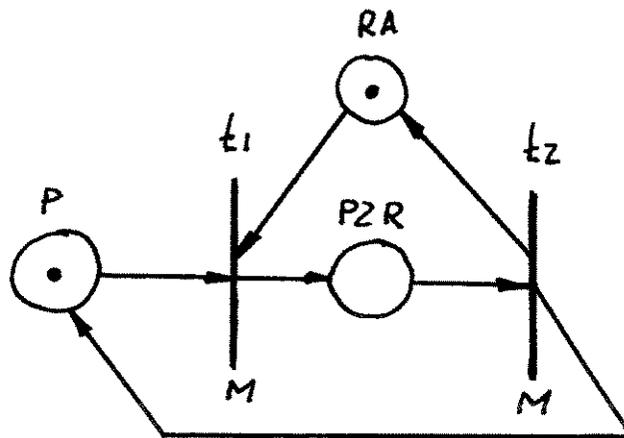


Figura 6. 1. Modelagem em PN para uma FMC, com programação *off-line* independente

Interpretação da simbologia:

P: Estação de carregamento / descarregamento

P2R: Estação de soldagem robotizada

RA: Robô disponível

$t_1$ : Mecanismo que posiciona peça ao estação de soldagem

$t_2$ : Mecanismo que posiciona peça na estação de carregamento / descarregamento

### 6. 1. 2. PN para FMC com Programação *Off-line* Simultânea

A FMC é configurada com dois robôs IRB 1400 para desenvolver operações de soldagem de arco elétrico e um dispositivo mecatrônico com três graus de liberdade que transporta uma peça da estação de carregamento / descarregamento para a estação de trabalho dos robôs e após as operações de soldagem transporta a peça para a estação de carregamento / descarregamento. O tempo de soldagem de cada um dos robôs sobre a peça é indiferente, conforme é mostrado na Figura 6. 2.

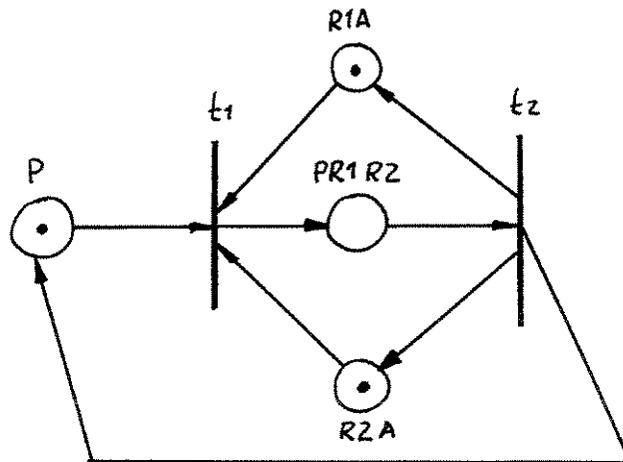


Figura 6. 2. Modelagem em PN para uma FMC com programação *off-line* simultânea

Interpretação da simbologia:

P: Estação de carregamento / descarregamento

PR1R2: Soldagem simultânea com R1 e R2

R1A: Robô 1 disponível

R2A: Robô 2 disponível

$t_1$ : Mecanismo que posiciona peça para soldagem com R1R2

$t_2$ : Mecanismo que posiciona peça na estação de carregamento / descarregamento

### 6. 1. 3. – PN para FMC com Programação *Off-line* Seqüencial

Neste exemplo a FMC é configurada com dois robôs (IRB 1400 e IRB 1500) e um dispositivo mecânico com três graus de liberdade para movimentação da peça. Cada um dos robôs desenvolve uma operação diferente de soldagem de arco elétrico sobre a peça, o mecanismo transporta a peça da estação de carregamento / descarregamento à estação de trabalho dos robôs e após as operações de soldagem transporta-se a peça à estação de carregamento / descarregamento. É indiferente o tempo de soldagem de cada um dos robôs sobre a peça, conforme é mostrado na Figura 6. 3.

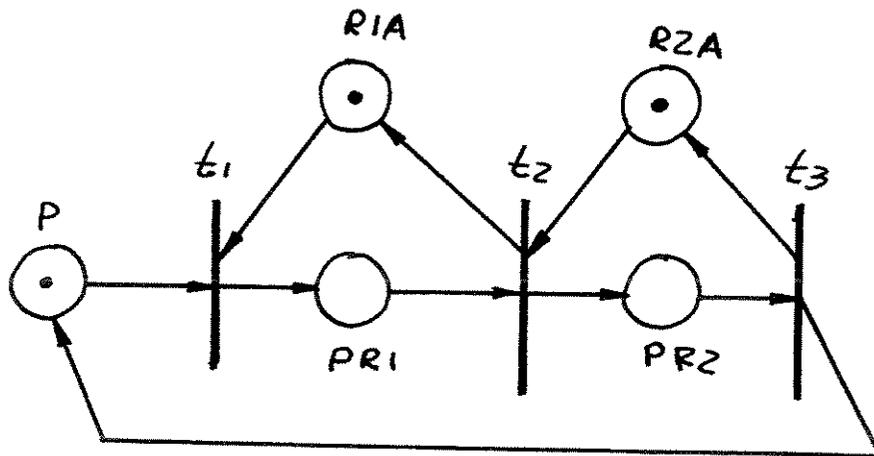


Figura 6. 3. Modelagem em PN para uma FMC com programação *off-line* seqüencial

Interpretação da simbologia:

P: Estação de carregamento / descarregamento

PR1A: Soldagem com R1

PR2A: Soldagem com R2

R1A: Robô 1 disponível

R2A: Robô 2 disponível

$t_1$ : Mecanismo que posiciona peça para soldagem com R1

$t_2$ : Mecanismo que gira peça para soldagem com R2

$t_3$ : Mecanismo que posiciona peça na estação de carregamento / descarregamento

#### 6. 1. 4. – PN para FMC com Programação *Off-line* Integrada

Neste exemplo a FMC é configurada com dois robôs (IRB 1400 e IRB 1500) e um dispositivo mecânico com três graus de liberdade. O mecanismo transporta uma peça da estação de carregamento / descarregamento à estação de trabalho dos robôs e após as operações de soldagem transporta a peça à estação de carregamento / descarregamento. Cada um dos robôs desenvolve uma operação de soldagem de arco elétrico sobre a peça. O tempo de soldagem em cada um dos robôs é diferente, conforme é mostrado na Figura 6. 4.

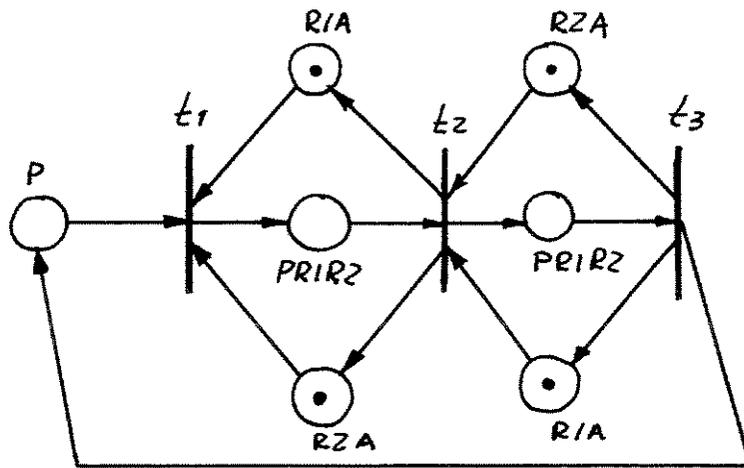


Figura 6. 4. Modelagem em PN para uma FMC através da programação *off-line* integrada

Interpretação da simbologia:

P: Estação de carregamento / descarregamento

PR1R2: Soldagem simultânea côm R1 e R2 (duas operações seqüenciais diferentes)

R1A: Robô 1 disponível

R2A: Robô 2 disponível

$t_1$ : Mecanismo que posiciona peça para soldagem com R1R2

$t_2$ : Mecanismo que gira peça para soldagem com R2R1

$t_3$ : Mecanismo que posiciona peça na estação de carregamento / descarregamento

## 6.2. Modelagem CAD de Dispositivos Mecatrônicos

**Passo 1:** Modelagem CAD da peça de produção.

**Passo 2:** Seleção da configuração CAD dos robôs da base de dados de acordo a sua função, seleção do modelo CAD das ferramentas de soldagem de arco elétrico da base de dados e determinação automática do espaço de trabalho de cada um dos robôs.

**Passo 3:** modelagem CAD do mecanismo de posicionamento e movimentação da peça de trabalho.

**Passo 4:** Integração mecatrônica dos robôs, ferramentas, mecanismo e peça para configurar a FMC. O tamanho da FMC é função do espaço de trabalho dos robôs cooperativos.

A Figura 6. 5 mostra a integração descrita no passo 4 para configurar a FMC, na Figura 6. 6 mostra o tamanho e vista superior da FMC derivada do espaço de trabalho dos robôs, e na Figura 6. 7 mostra a vista isométrica da mesma FMC e finalmente a Figura 6. 8. mostra vista superior dos robôs cooperativos e mecanismo dentro da FMC.

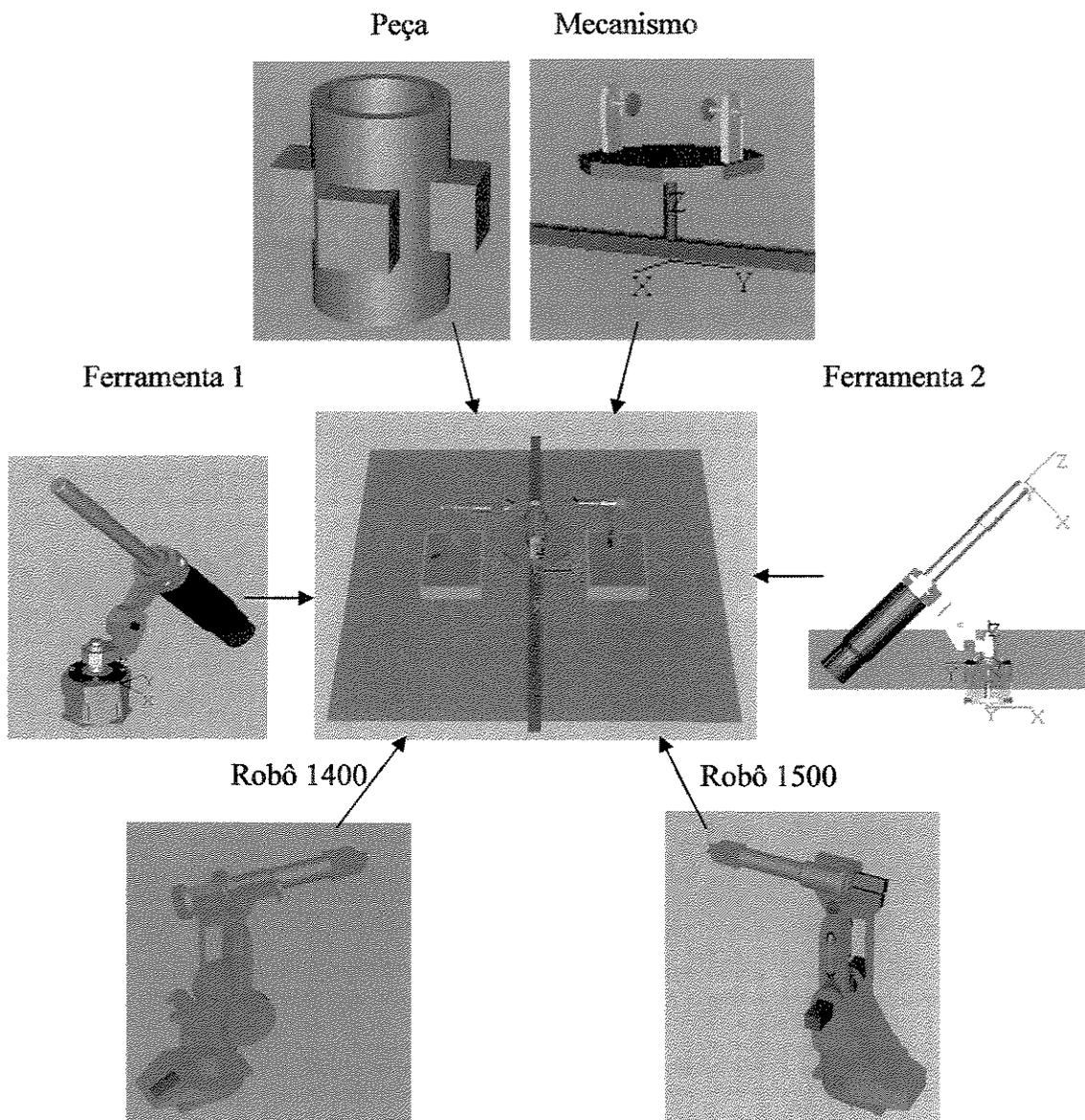


Figura 6. 5. Integração mecatrônica entre os componentes da FMC

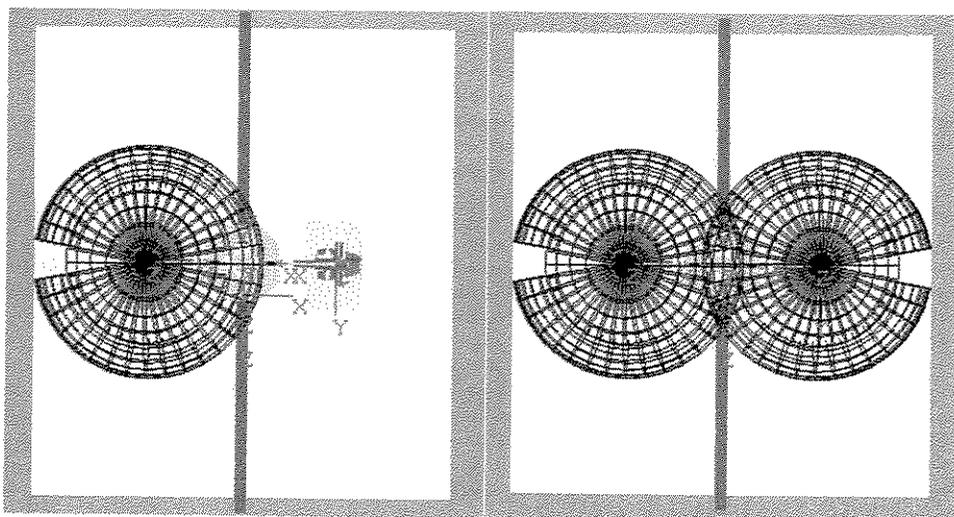


Figura 6. 6. Vista superior do espaço de trabalho dos robôs da FMC

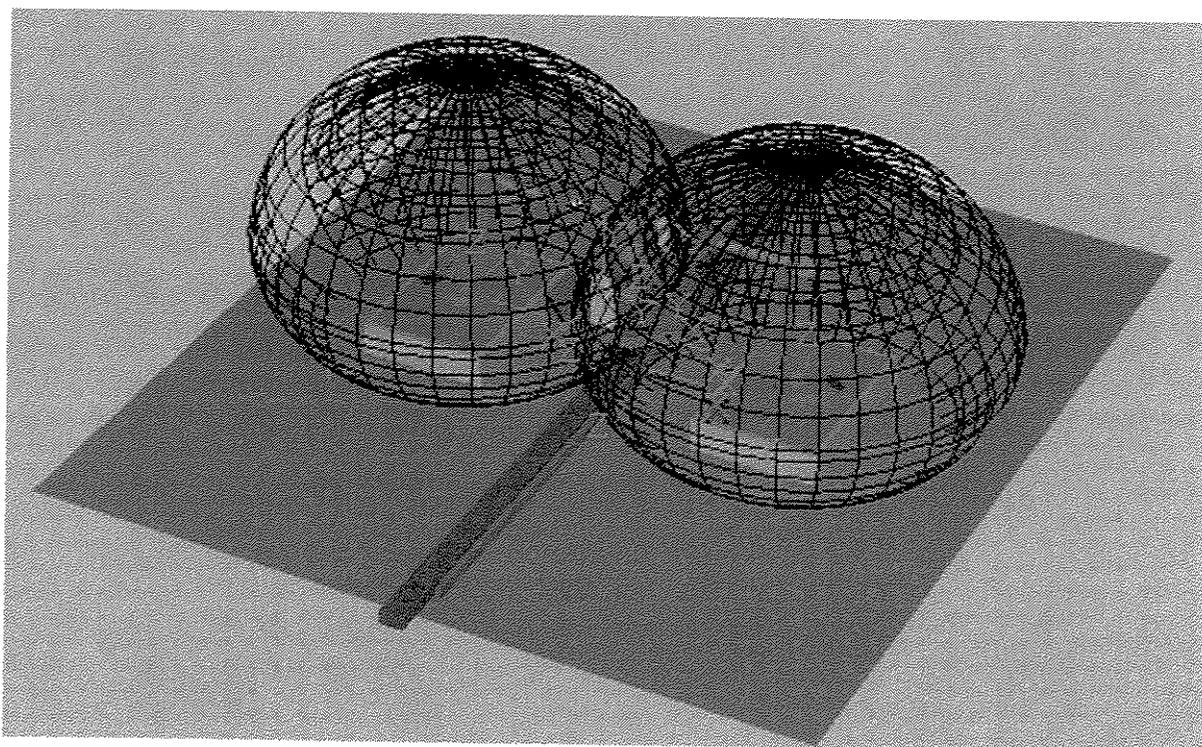


Figura 6. 7. Modelagem CAD do espaço de trabalho da FMC

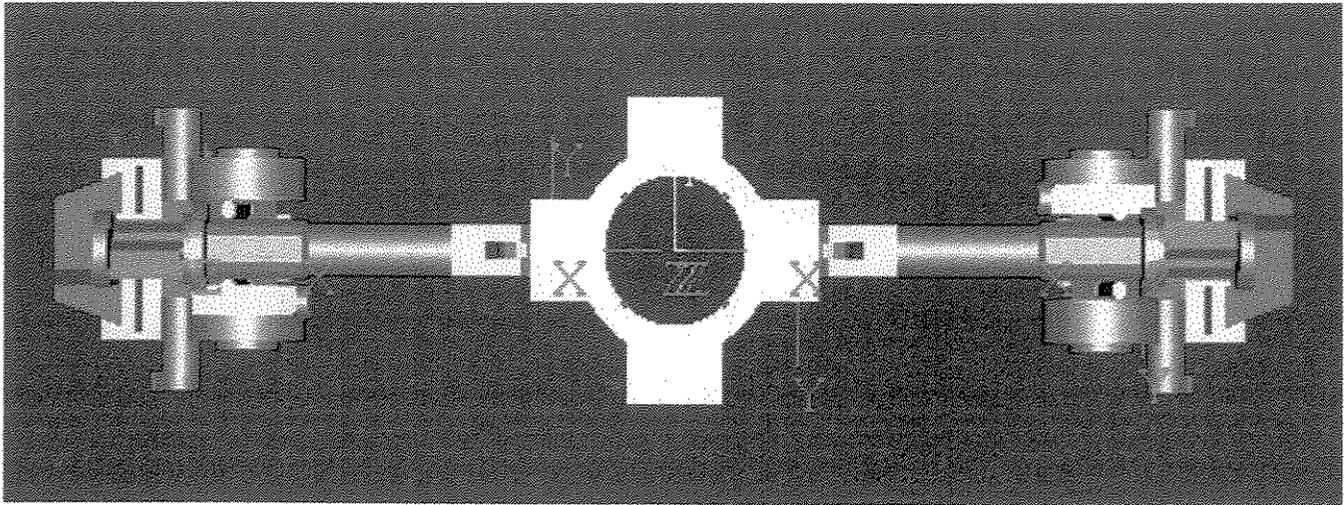


Figura 6. 8. Dois robôs cooperativos e mecanismo dentro da FMC

### 6. 3. Modelagem da Cinemática do Robô

**Passo 5:** As MTH são a ferramentas mais utilizadas para o cálculo da cinemática direta e inversa, mas os robôs de ABB expressão as posições em coordenadas cartesianas e as orientações em Quaternions como é apresentada na Figura 6. 9.

| Special        |                                       |                   |             |
|----------------|---------------------------------------|-------------------|-------------|
| <b>Jogging</b> |                                       | <b>Robot Pos:</b> |             |
| Unit:          | IRB                                   | x:                | 964.4 mm    |
| Motion:        | Linear                                | y:                | 712.4 mm    |
|                |                                       | z:                | 525.7 mm    |
| Coord:         | <u>World</u> <input type="checkbox"/> | q1:               | 0.27186     |
| Tool:          | tool0 ...                             | q2:               | -0.25529    |
| WObj:          | wobj0 ...                             | q3:               | 0.98949     |
|                |                                       | q4:               | -0.18372    |
| Joystick lock: | None <input type="checkbox"/>         |                   |             |
| Incremental:   | No <input type="checkbox"/>           |                   |             |
|                |                                       | x                 | z           |
|                |                                       | ↑                 | ↻           |
|                |                                       |                   | ←           |
| <b>World</b>   | <b>Base</b>                           | <b>Tool</b>       | <b>WObj</b> |

Figura 6. 9: Posição em coordenadas cartesianas e orientação em Quaternions dos robôs

A modelagem da cinemática direta e inversa de cada robô é única e necessita ser desenvolvida e formulada para programação *off-line* e controle do robô real. Os sistemas de simulação e programação *off-line* são ferramentas para visualizar a modelagem geométrica, simulação e projeto de novas células FMC, e a movimentação é baseada na utilização de algoritmos de aproximação.

Para um robô específico (como no caso do IRB 1400 de ABB) é necessário desenvolver seu próprio algoritmo da modelagem da cinemática e carregá-lo no software de simulação e programação *off-line* ou compará-lo com o do fabricante do robô num módulo conhecido como RRS. O software Workspace é somente uma ferramenta para modelagem geométrica, simulação e programação *off-line* de robôs, cuja cinemática está baseada num algoritmo de aproximação, mas não tem fornecido a modelagem da cinemática de todos os robôs disponíveis na sua base de dados ou disponíveis no mercado.

Neste trabalho de pesquisa são desenvolvidas metodologias para a modelagem da cinemática de robôs baseada nas MTH e Quaternions que necessitam serem instalados no software de simulação e programação *off-line*, no caso de implementação de uma FMC real.

**Passo 6.** A velocidade da operação da ferramenta de soldagem pode ser calculada com a cinemática diferencial direta da expressão 4. 64 e as velocidades das juntas podem-se calcular com a cinemática diferencial inversa da expressão 4. 78. O software Workspace apresenta uma tabela para escrever as velocidades nas juntas (Figura 4. 9) e apresenta uma tabela para entradas dos parâmetros de Denavit – Hartenberg (Figura 4. 11). Estas tabelas funcionam com o algoritmo de aproximação de Workspace. No caso que precise implementar-se a FMC real é necessário instalar a modelagem da cinemática diferencial direta e inversa, além da cinemática direta e inversa.

A FMC tem o sistema de coordenada em  $O_{FMC}(X_{FMC}, Y_{FMC}, Z_{FMC})$  ou  $O_{FMC}(0,0,0)$  o IRB 1400 em  $O_{1400}(-1500,0,300)$  e o IRB 1500 em  $O_{1500}(+1500,0,300)$ . Os sistemas de coordenadas das ferramentas são apresentados na Figura 6. 10, respectivamente, para o primeiro e segundo robôs.

A Figura 6. 11, apresenta as tabelas com as propriedades do robô IRB 1400. O software Workspace 5 não tem modulo RRS para os robôs da ABB, o qual significa que a simulação acontece com à ajuda do seu próprio algoritmo de aproximação RRS. Para obter uma simulação que corresponda ao modelo do robô IRB 1400 é necessário desenvolver a modelagem da cinemática deste robô específico e instalá-lo no software de simulação e programação *off-line*, neste caso instalar-lo no software Workspace antes de implementar a FMC real. Na Figura 6. 12 apresentam-se as propriedades da posição e orientação do mecanismo.

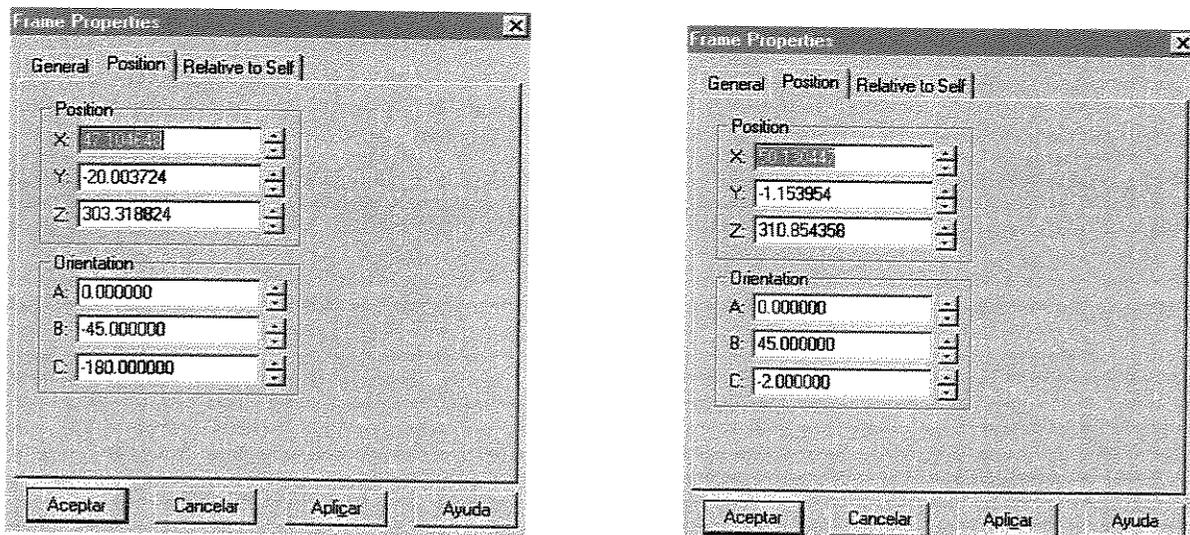


Figura 6. 10: Posição e orientação das ferramentas de soldagem elétrico CAT241D

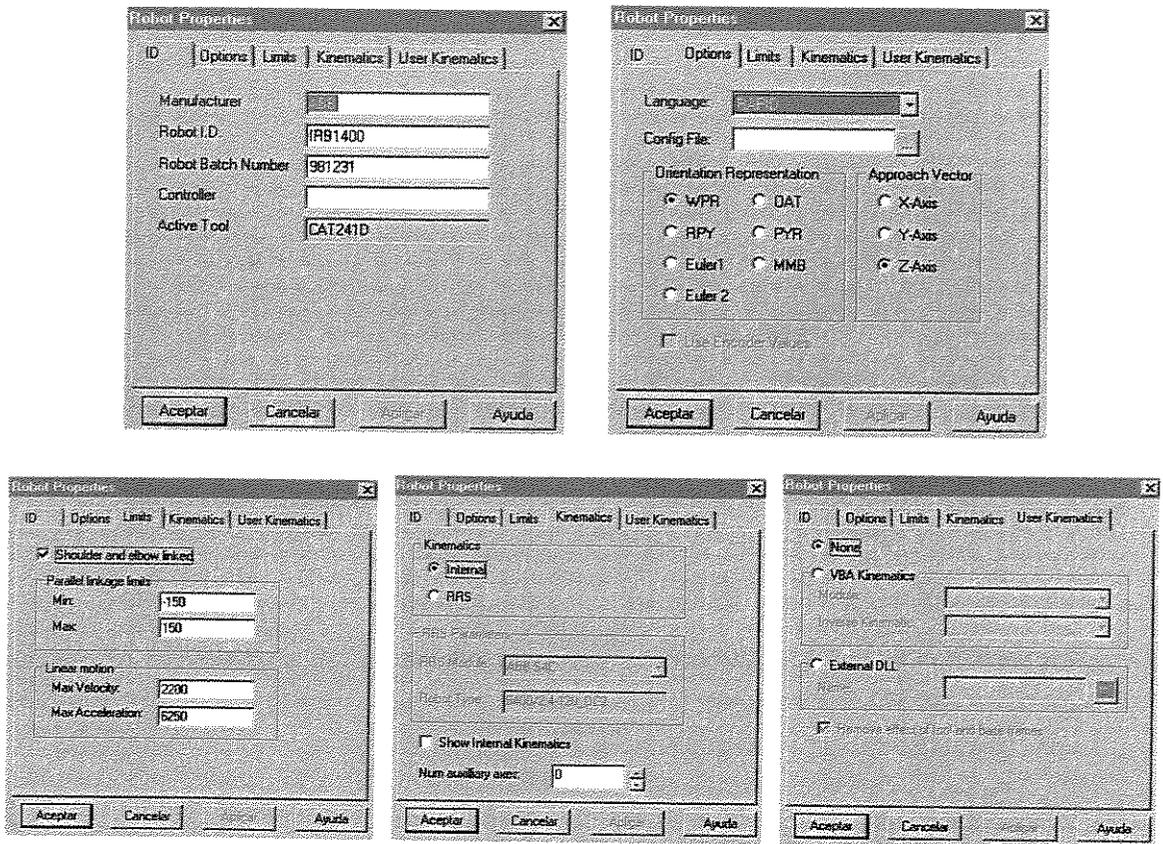


Figura 6. 11. Propriedades do robô IRB 1400.

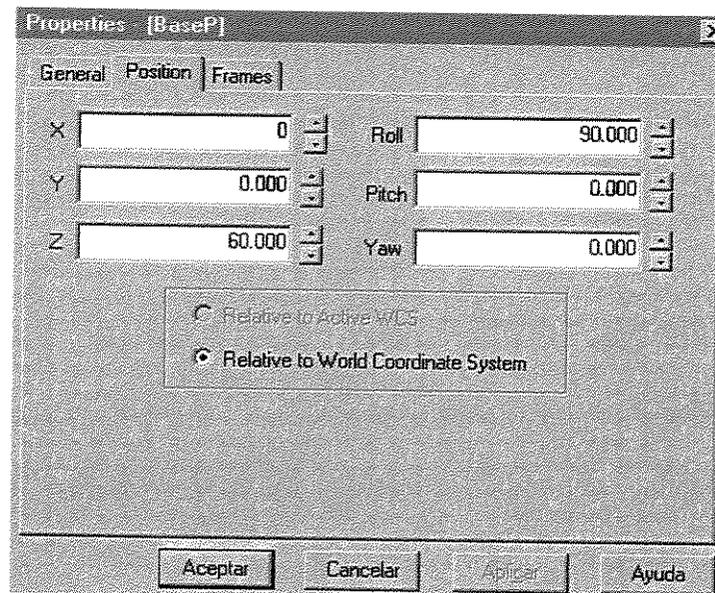


Figura 6. 12. Propriedades de posição e orientação do mecanismo

## 6. 4. Simulação

- Passo 7.** Definição de PGs.
- Passo 8.** Planejamento de trajetórias.
- Passo 9.** Interpolação de trajetórias.
- Passo 10.** Geração de trajetórias.
- Passo 11.** Simulação e / ou animação.

A Figura 6. 13 apresenta a definição dos PGs para o robô IRB 1400 e IRB 1500, e a Figura 6. 14 mostra os PGs integrados numa trajetória e a modelagem da trajetória. Com esta informação pode-se gerar simulação gráfica no software Workspace desta trajetória independente para o caso de um robô. Na seção 5. 1 foi desenvolvida a metodologia da simulação da FMC.

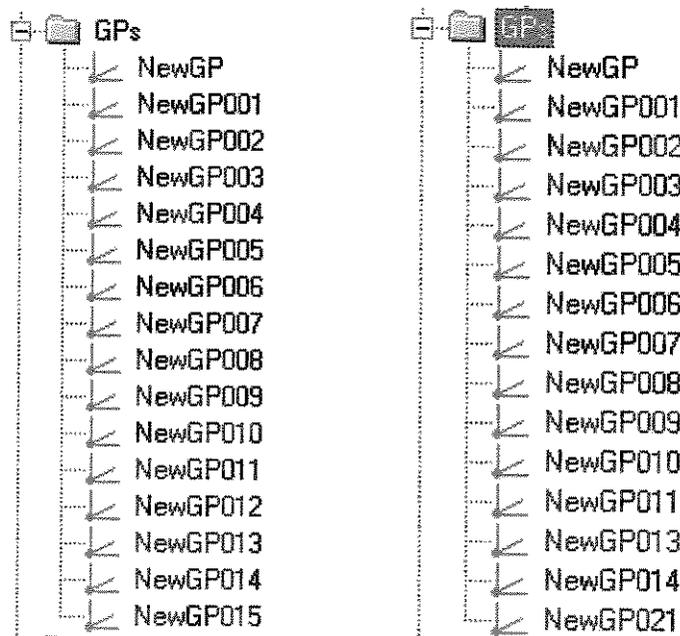


Figura 6. 13. PGs para os robô IRB 1400 e IRB 1500.

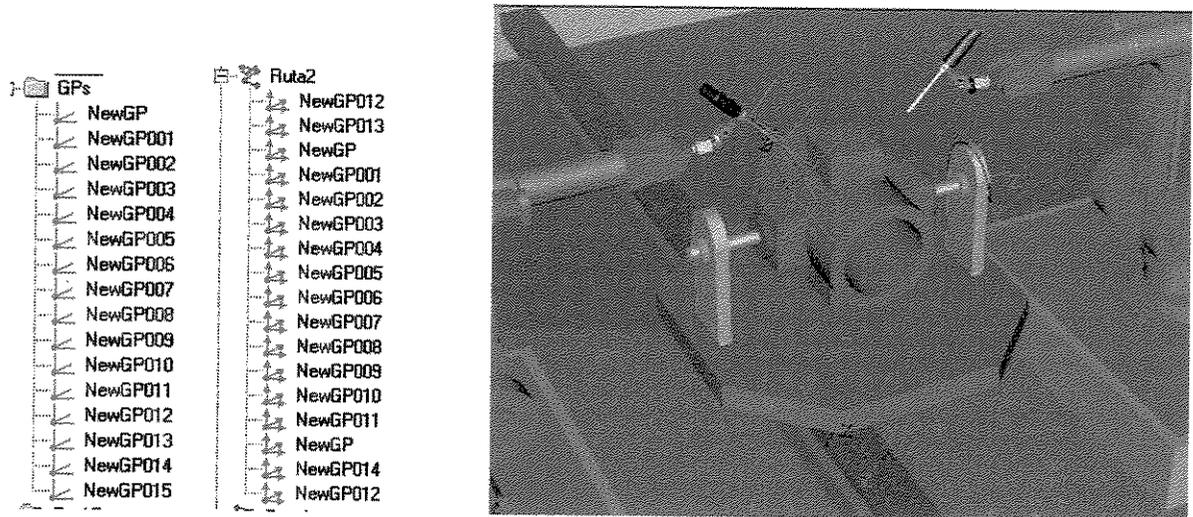


Figura 6. 14. PGs, trajetória e POL Independente para IRB 1400 de ABB.

## 6. 5. Simulação e Programação *Off-line* Independente

**Passo 12.** Seleção da linguagem.

**Passo 13.** Edição da programação *off-line*.

Na seção 5. 8. 1 foi desenvolvida a metodologia para preparação do robô, seleção do linguagem e programação *off-line* da FMC. Na tabela 6.1, é apresentado o programa para movimentação da trajetória do robô desenvolvido a partir do software Workspace.

Tabela 6.1: Programa em RAPID para a trajetória do robô IRB 1400

```
MoveJ NewGP012,v1000,fine,t_CAT241D;  
MoveJ NewGP013,v1000,fine,t_CAT241D;  
MoveJ NewGP,v1000,fine,t_CAT241Dlvido  
MoveL NewGP001,v1000,fine,t_CAT241D;  
MoveL NewGP002,v1000,fine,t_CAT241D;  
MoveL NewGP003,v1000,fine,t_CAT241D;  
MoveL NewGP004,v1000,fine,t_CAT241D;  
MoveL NewGP005,v1000,fine,t_CAT241D;  
MoveL NewGP006,v1000,fine,t_CAT241D;  
MoveL NewGP007,v1000,fine,t_CAT241D;  
MoveL NewGP008,v1000,fine,t_CAT241D;  
MoveL NewGP009,v1000,fine,t_CAT241D;  
MoveL NewGP010,v1000,fine,t_CAT241D;  
MoveL NewGP011,v1000,fine,t_CAT241D;  
MoveJ NewGP,v1000,fine,t_CAT241D;  
MoveJ NewGP014,v1000,fine,t_CAT241D;  
MoveJ NewGP012,v1000,fine,t_CAT241D;  
ENDPROC
```

## 6. 6. Simulação e Programação *Off-Line* Simultânea

**Passo 12.** Seleção da linguagem.

**Passo 13.** Edição da programação *off-line*.

Repitam-se os Passos 12 e 13 para programação *off-line* simultânea. Na Figura 6. 15 mostram-se os PGs criados para o robô IRB 1500, os PGs integrados numa trajetória e a configuração das trajetórias dos robôs IRB 1400 e IRB 1500.

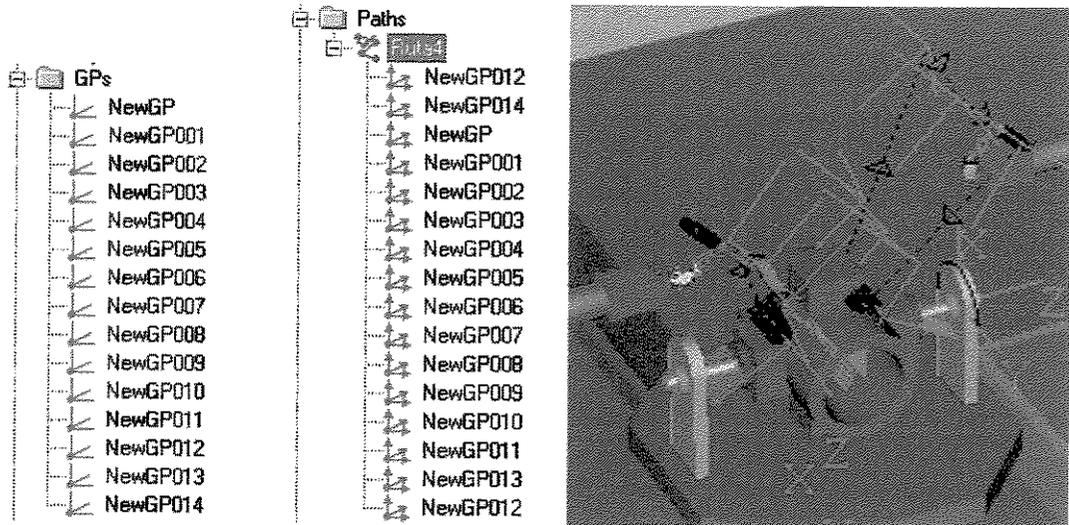


Figura 6. 15. PGs, trajetória e POL Simultânea para IRB 1500

Na Figura 6. 16 são apresentadas as propriedades e linguagem do robô IRB 1500, a linguagem RAPID da ABB, permitindo selecionar os limites das juntas, velocidade do TCP, a cinemática baseada no algoritmo interno de aproximação. Através da janela da Figura 6. 17 definam-se os limites da velocidade e aceleração das juntas da estrutura do robô. O mesmo procedimento foi utilizado para o robô IRB 1400.

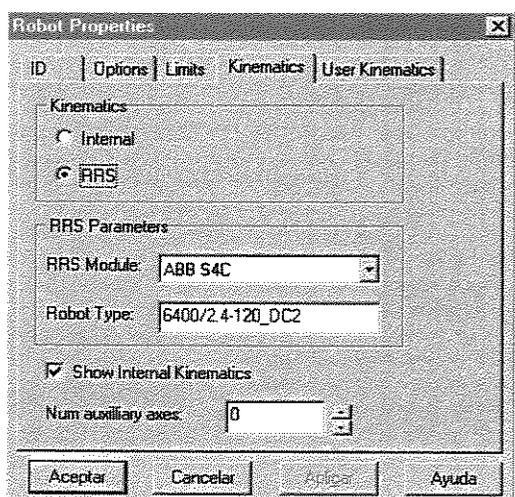
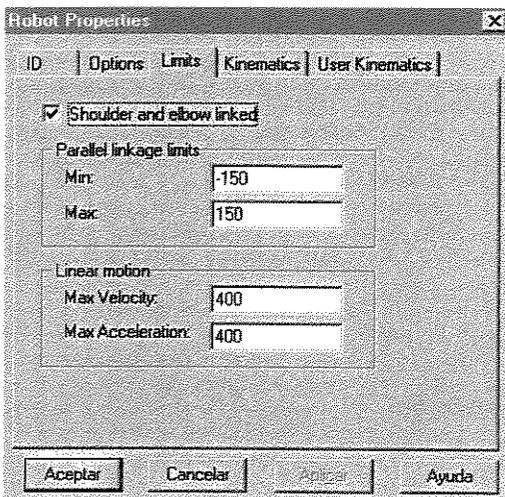
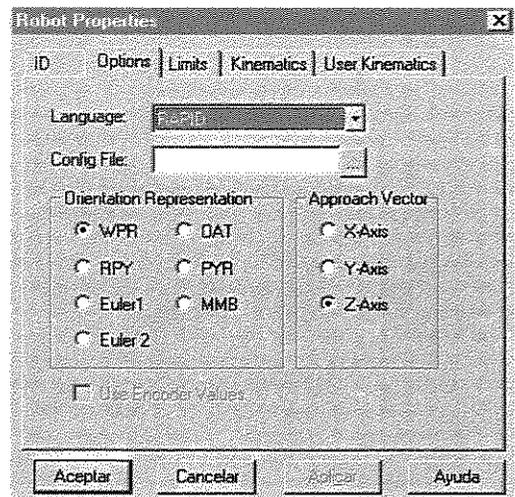
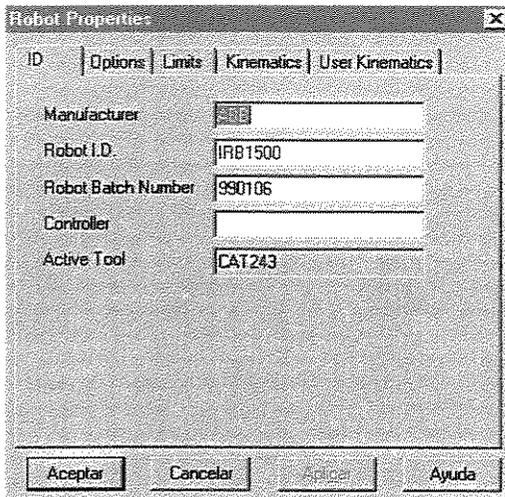


Figura 6. 16. Propriedades e linguagem do robô IRB 1500

| Joint | Max Vel | Max Acc | Monitor |
|-------|---------|---------|---------|
| 1     | 200.00  | 400.00  | Yes     |
| 2     | 200.00  | 400.00  | Yes     |
| 3     | 200.00  | 400.00  | Yes     |

Figura 6. 17. Limites e cinemática dos robôs

Na tabela 6.2 são apresentados o programa para movimentação da trajetória do robô criado *off-line* com o software Workspace.

Tabela 6.2: Programa em RAPID para a trajetória do robô IRB 1500

```
MoveJ NewGP012,v1000,fine,t_CAT243;  
MoveJ NewGP014,v1000,fine,t_CAT243;  
MoveJ NewGP,v1000,fine,t_CAT243;  
MoveL NewGP001,v1000,fine,t_CAT243;  
MoveL NewGP002,v1000,fine,t_CAT243;  
MoveL NewGP003,v1000,fine,t_CAT243;  
MoveL NewGP004,v1000,fine,t_CAT243;  
MoveL NewGP005,v1000,fine,t_CAT243;  
MoveL NewGP006,v1000,fine,t_CAT243;  
MoveL NewGP007,v1000,fine,t_CAT243;  
MoveL NewGP008,v1000,fine,t_CAT243;  
MoveL NewGP009,v1000,z1,t_CAT243;  
MoveL NewGP010,v1000,fine,t_CAT243;  
MoveL NewGP011,v1000,fine,t_CAT243;  
MoveJ NewGP013,v1000,fine,t_CAT243;  
MoveJ NewGP012,v1000,fine,t_CAT243;  
ENDPROC
```

### 6. 7. Simulação e Programação *Off-Line* Seqüencial

**Passo 12.** Seleção da linguagem.

**Passo 13.** Edição da programação *off-line*.

Os Passos 12 e 13 são repetidos para programação *off-line* seqüencial. Na Figura 6. 18 mostram-se as trajetórias integradas e sinais digitais para programação *off-line* seqüencial dos robôs IRB 1400 de ABB e IRB 1500. Esta programação *off-line* dos robôs já foi descrita anteriormente nas seções 6. 5, 6. 6 e 6. 7

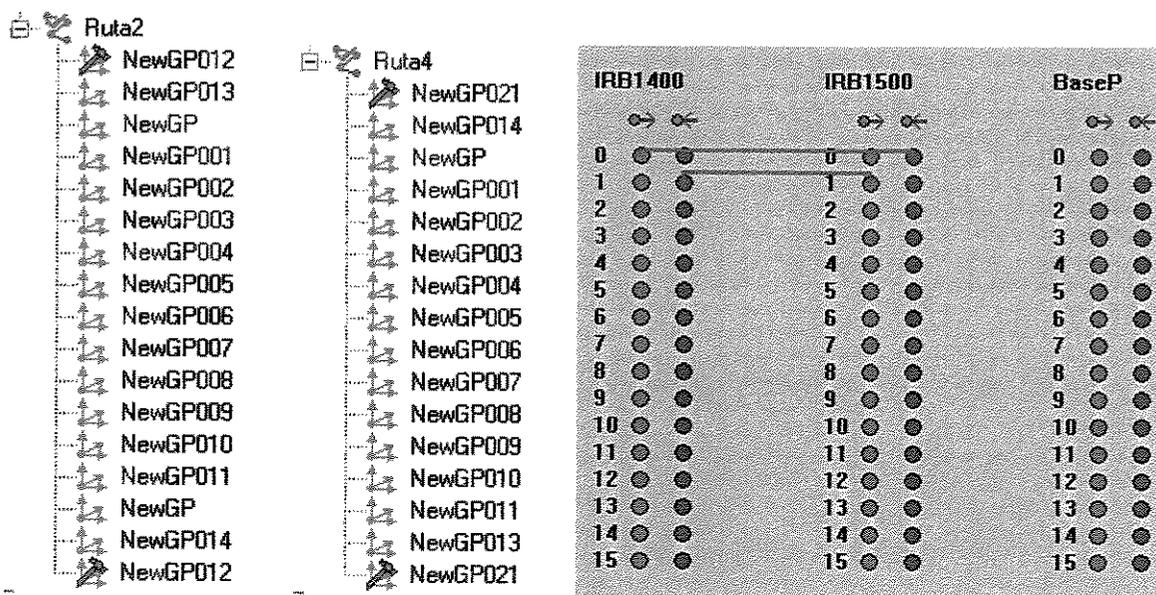


Figura 6. 18. Trajetórias e Sinais Digitais para POL seqüencial dos robôs IRB 1400 de ABB e IRB 1500

## 6. 8. Simulação e Programação *Off-Line* Integrada

**Passo 12.** Seleção da linguagem.

**Passo 13.** Edição da programação *off-line*.

Para a programação *off-line* integrada são repetidos os Passos 12 e 13. Na Figura 6. 19 apresenta-se a trajetória do mecanismo de movimentação com um ícone no ambiente Workspace representando um martelo nos PGs NewGP, NewGP62 e NewGP28, que significa que a programação *off-line* do mecanismo esta integrada com a programação *off-line* dos robôs IRB 1400 e IRB 1500 descrito na seção 6. 7 para um ciclo de trabalho.

Assim para implementarmos a programação *off-line* para os dois robôs integrados a outros mecanismos dentro da FMC. Na Figura 6. 20 é apresentada a ligação dos sinais digitais de simulação e programação *off-line* integrada entre os diferentes componentes da FMC.

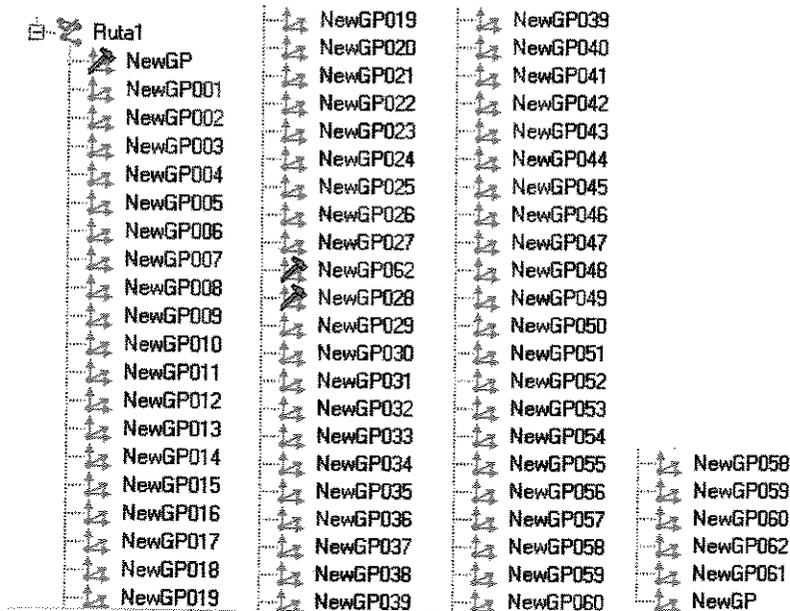


Figura 6. 19. Trajetórias integradas do mecanismo MR + MB de 3 GL.

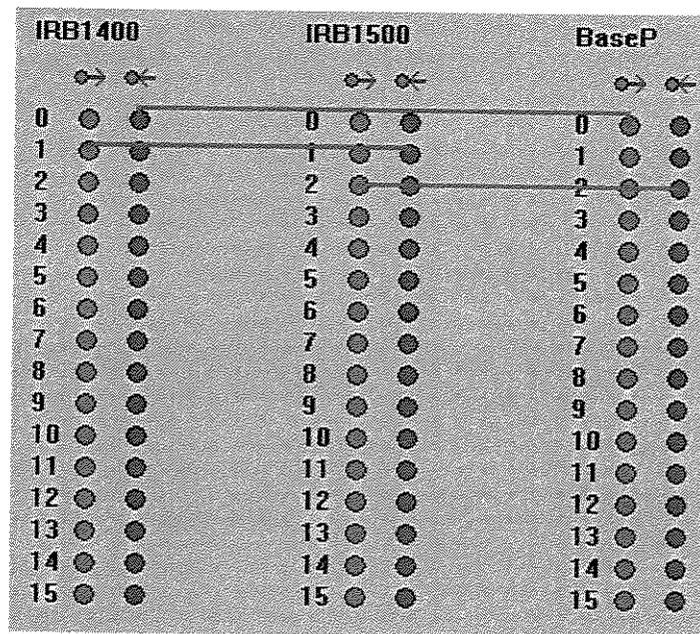


Figura 6. 20. Sinais digitais de simulação e programação *off-line* integrada.

Na Figura 6. 21 são apresentadas a visualização da FMC virtual desenvolvida em três diferentes projeções, utilizando a metodologia proposta neste trabalho de pesquisa sobre simulação e programação *off-line* de robôs e dispositivos mecâtrônicos integrados, com dois robôs IRB de ABB e mecanismo MR + MB, com 15 graus de liberdade proposto para ser implementado no Laboratório de Automação Integrada e Robótica da UNICAMP.

## **6. 9. Célula Robotizada da UNICAMP**

Na Figura 6. 22 é apresentada a célula robotizada implementada no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP constituída de um robô IRB 1400 da ABB (Figura 6. 23) e de um robô IRB 140 da ABB (Figura 6. 23).

A implementação da FMC com a metodologia desenvolvida nesta pesquisa para simulação e programação *off-line*, implica a integração dos robôs com o mecanismo de movimentação em forma física e a integração em forma virtual numa estação computadorizada que inclui o software de programação e controle.

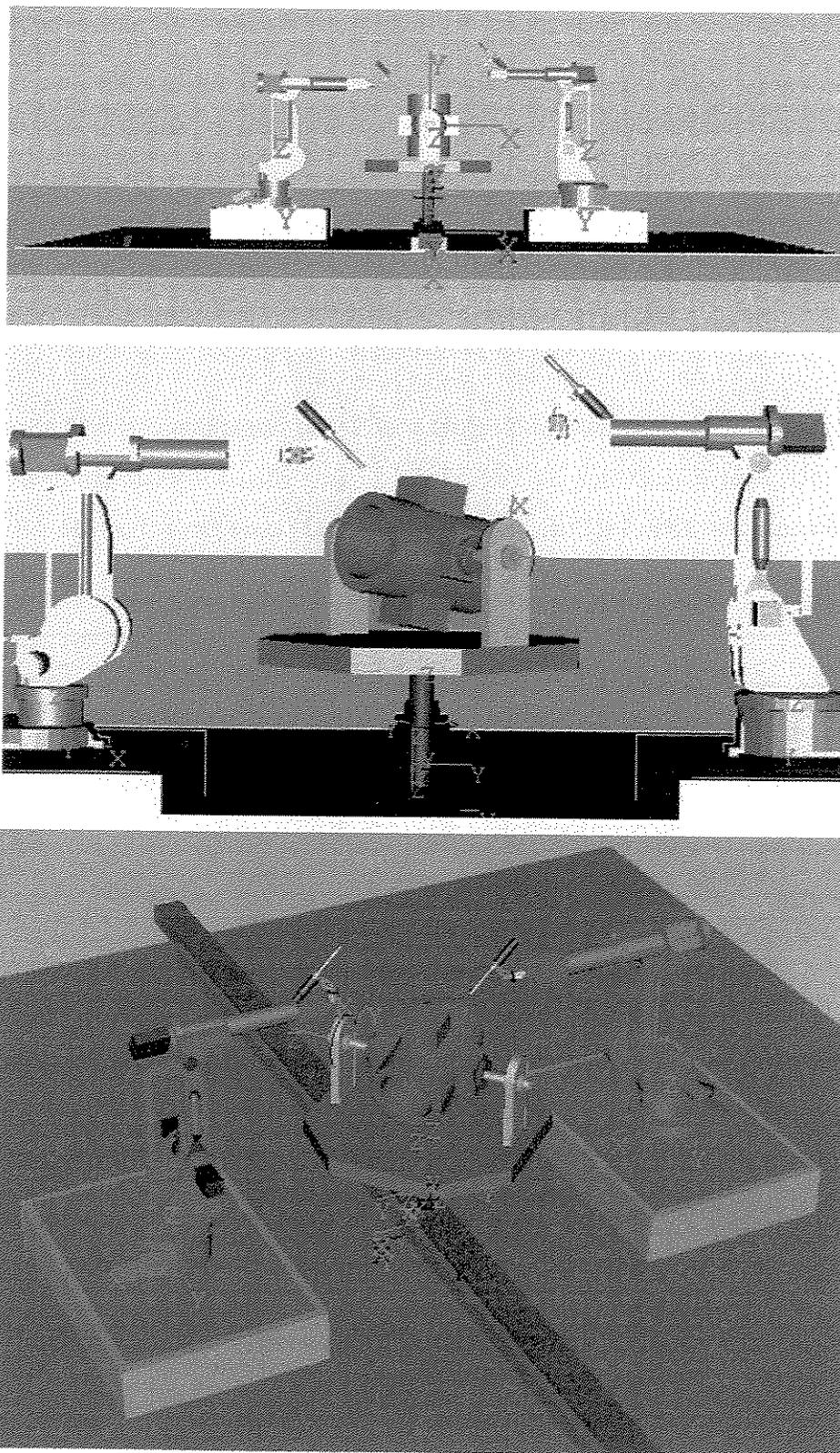


Figura 6. 21. FMC implementa na UNICAMP de 15 GL com dois robôs de ABB e mecanismo MR + MB

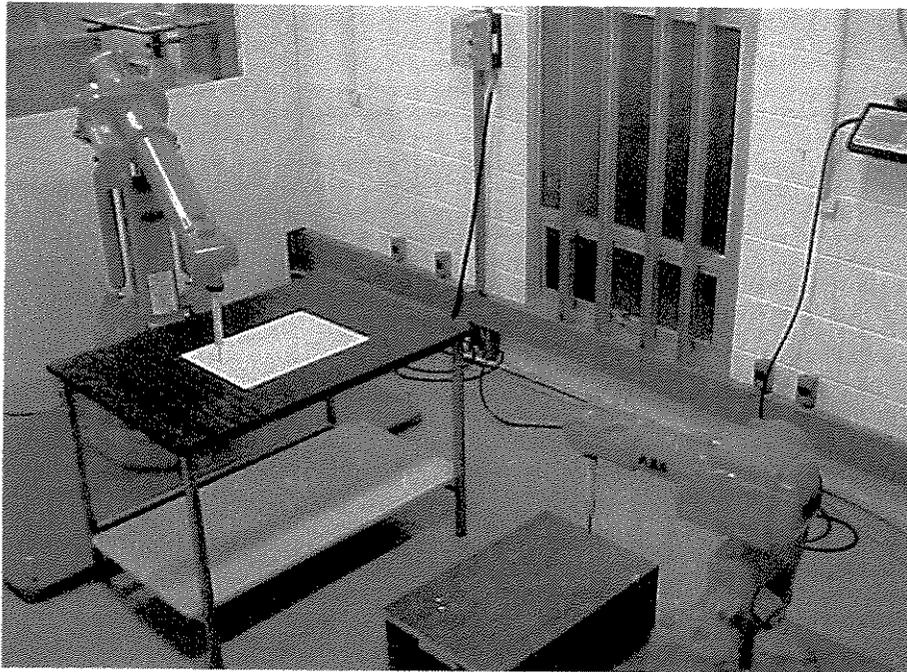


Figura 6. 22. Célula robotizada da UNICAMP.

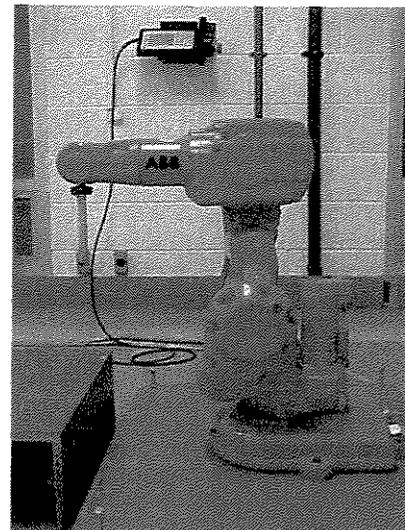
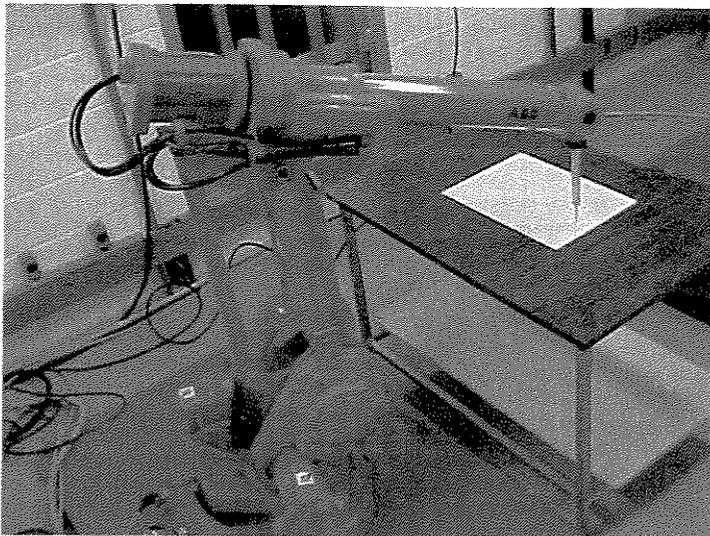


Figura 6. 23. Robô IRB 1400 e IRB 140 da UNICAMP.

## 6. 10. Exemplo 2, FMC de Soldagem com dois Robôs e Mecanismo de Rotação.

William (1994) descreve uma estação de trabalho como parte de uma linha de produção automotiva, baseada em dois robôs IRB de ABB, uma esteira elétrica, uma mesa rotativa, um sistema de armazenamento (buffer), e um trabalhador. Cada um dos robôs desenvolve a operação de soldagem por arco para soldar dois tipos de dobradiças nas portas dos veículos.

Os robôs de soldagem trabalham ao redor da mesa rotatória, que tem três posições. Pode-se dizer posição X para o trabalhador, posição Y para o robô 1 e posição Z para o robô 2. O trabalhador inspeciona e tira uma porta de uma esteira elétrica e a posiciona em X da mesa rotatória. A partir de um sistema de armazenagem adjunto pega um tipo A de dobradiça e a posiciona sobre o lugar 1 da porta, depois pega um segundo tipo B de dobradiça e a posiciona sobre o lugar 2 da porta. A porta com as duas dobradiças gira até a posição Y, onde o robô 1 realiza a operação de soldagem no primeiro tipo A de dobradiça, onde novamente a mesa gira até a posição Z, e o robô 2 realiza a operação de soldagem no segundo tipo B de dobradiça. Finalmente, a mesa gira de volta até à posição de origem X, onde o trabalhador inspeciona a porta com as duas dobradiças em soldagem, tira-la e posiciona-la na esteira elétrica para continuar o processo na linha de produção.

No modelo de William (1994) o trabalhador desenvolve as seguintes operações:

- (1) Carga porta da esteira elétrica e descarregar-la na posição X da mesa rotatória,
- (2) Carga uma dobradiça tipo A e descarregar-la na posição 1 da porta,
- (3) Carga uma dobradiça tipo B e descarregar-la na posição 2 da porta,
- (4) Inspeccionar qualidade das soldaduras em cada uma das duas dobradiças,
- (5) Carga da porta com as dobradiças soldadas e descarregar-la na esteira elétrica para continuar com o processamento na linha de produção

Neste exemplo, não existe uma FMC, só existe um posto de trabalho numa linha de produção com mistura de trabalho manual, trabalho mecanizado e trabalho automatizado com dois robôs.

O trabalhador desenvolve cinco operações, entando que cada um dos robôs só desenvolve uma operação de soldagem. O tempo de trabalho do operador é muito maior que o tempo de trabalho dos robôs. Pode-se afirma que, o posto de trabalho do operador é um gargalo e têm-se os dois robôs por debaixo de seu potencial de trabalho.

Agora, baseado na metodologia proposta neste trabalho de pesquisa, para a simulação e programação *off-line* de robôs integrados numa FMC, propõe-se o seguinte ajuste da estação: Utilizando os mesmos componentes configura-se e modela-se uma FMC, como se mostra na Figura 6. 24:

- (1) dois robôs IRB 1400 de ABB para soldagem de arco e
- (2) dispositivo mecatrônico de rotação para posicionamento de um grau de liberdade.

Este dispositivo mecatrônico tem duas posições: uma posição X de carregamento / descarregamento e outra posição Y de soldagem. As seguintes operações são efetuadas:

- a) Na primeira posição X o robô 1 desempenha a função de movimentação (*pick-and-place*), através do carregamento da porta da esteira elétrica e descarregamento na posição X da mesa, de um sistema de armazenamento A (*buffer*) carrega a dobradiça tipo A, descarregando a seguir na posição A da porta,
- b) Um sistema de armazenamento B (*buffer*) carrega a dobradiça tipo B e descarrega na posição B da porta. A mesa gira 180 graus e chega à posição Y, onde o robô 2 realiza a operação de soldagem na dobradiça tipo A e depois realiza a operação de soldagem na dobradiça tipo B.
- c) O robô 2 termina sua operação de soldagem das dobradiças sobre a porta e a mesa gira 180 graus e volta a o estação X de carregamento / descarregamento, onde o robô 1 carga a porta com as dobradiças e descarregar-la na esteira elétrica.
- d) Os robôs desenvolvem suas tarefas simultaneamente. Nesta situação, o robô 1 desenvolve a função do antigo trabalhador é este agora desenvolve a suave função de inspeção e controle de qualidade das operações de soldagem dos robôs 1 e 2.

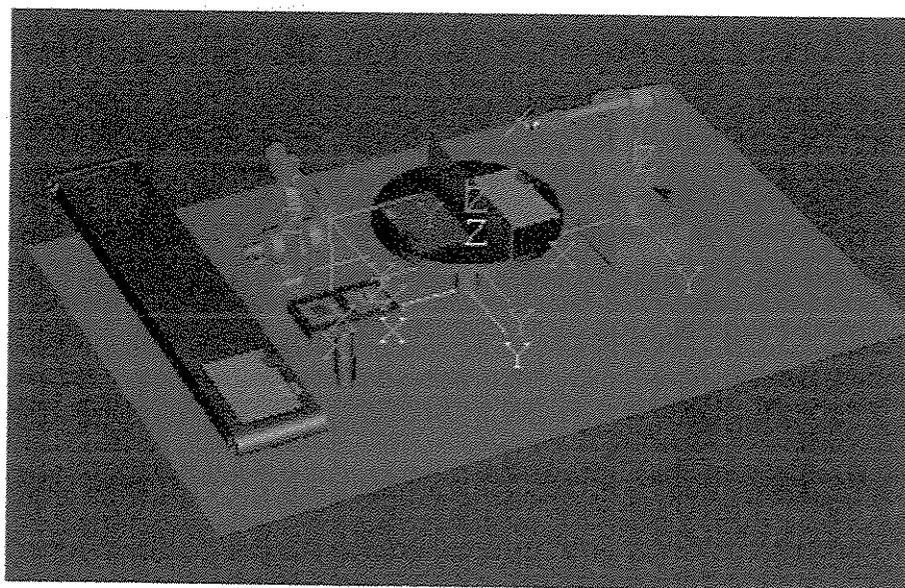
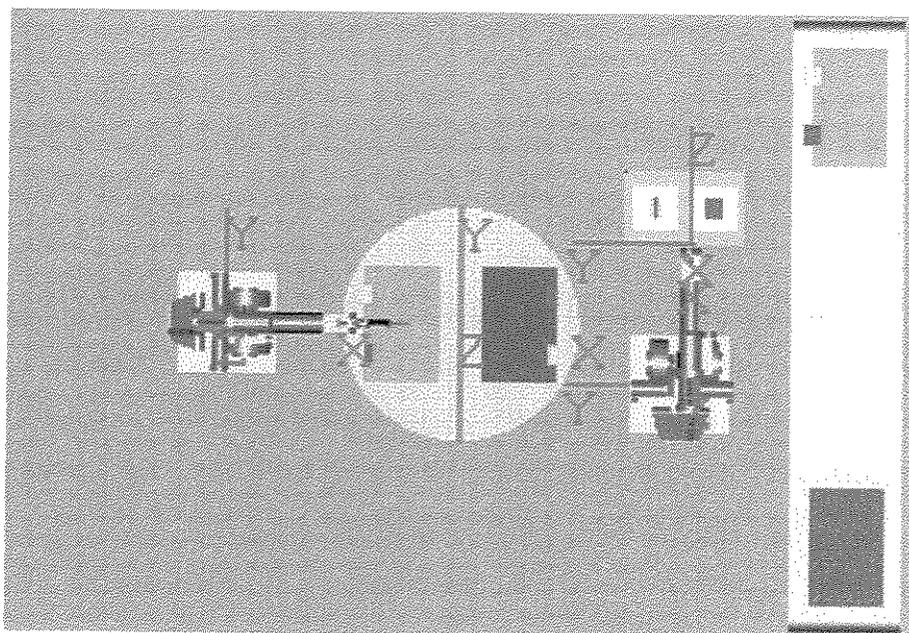


Figura 6. 24. FMC de soldagem com dois robôs

Esta FMC tem as seguintes vantagens:

- (1) Projeta-se uma FMC com dois robôs e um mecanismo de posicionamento.
- (2) O robô 1 destina-se a função de movimentação:
  - a) Carregamento da porta da esteira elétrica ao estação 1 da mesa rotativa,
  - b) movimentação da dobradiça tipo A,

- c) movimentação da dobradiça tipo B,
  - d) descarregamento da porta com as dobradiças da mesa ao esteira elétrica.
- (3) O robô 2 desenvolve a soldagem das duas dobradiças na porta.
  - (4) O mecanismo tem só duas posições.
  - (5) O trabalhador desenvolve a suave tarefa de inspeção e controle de qualidade.
  - (6) Na FMC o tempo do trabalho é minimizado.
  - (7) Na FMC a precisão do trabalho é maximizada.
  - (8) Na FMC incrementa-se a utilização dos robôs.
  - (9) Na FMC incrementa-se a produtividade e a flexibilidade.
  - (10) Suprime-se o trabalho repetitivo e fatigoso por um trabalho suave e de qualidade.

### 6. 11. Exemplo 3, FAC com dois Robôs e Mecanismo de Rotação.

Buxbaum (1995) analisa o incremento nas aplicações dos robôs em sistemas de produção com lotes de pequeno tamanho e variedade de peças, que caracteriza a produção flexível. Nestas aplicações o robô é um componente essencial e descreve o conceito de células de trabalho de manufatura flexível baseada em robôs. Apresenta uma célula de trabalho típica com multi-robôs em tarefas de montagem, para encaminhar o desenvolvimento do conceito como estratégia da automação da fábrica. Entretanto essa abordagem, não apresenta modelagem, simulação e programação *off-line* da FAC, como ferramenta de avaliação.

Assim, neste trabalho aplicaremos a metodologia desenvolvida nesta pesquisa, sobre simulação e programação *off-line* de robôs integrados com mecanismos em FMC, se identifica que a flexibilidade das FAC é muito maior que a flexibilidade das FMC. Conseqüentemente, será configurado e modelado em sistema CAD uma FAC típica, apresenta-se validação do tratamento de colisões dos robôs e finalmente simula-se sua movimentação. A Figura 6. 25 apresenta a configuração da nova FAC, incluindo:

- (1) dois robôs de montagem IRB 1400,
- (2) uma estação de montagem e um sistema de armazenamento (buffer) com variadas peças para inserção, realizando três funções de simulação: a) identificação das

peças, b) movimentação das peças por os robôs e c) inserção das peças por os robôs.

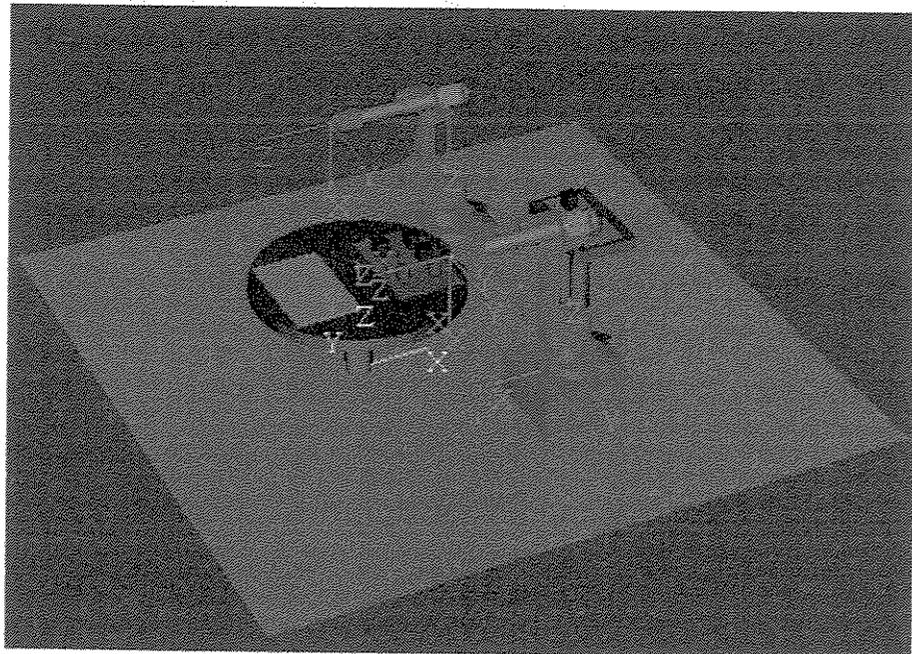
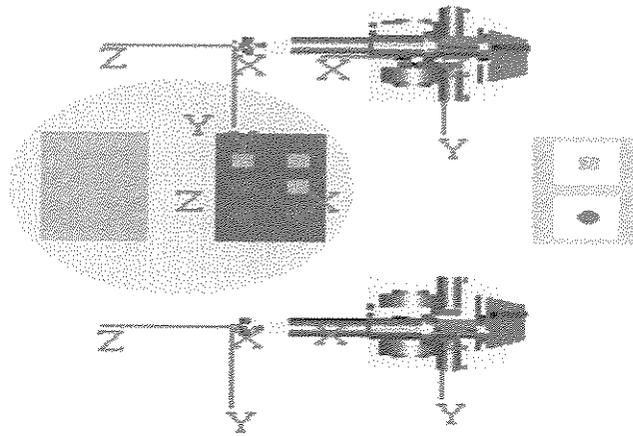


Figura 6. 25. FMC de montagem com dois robôs

Na estação de carregamento / descarregamento são identificados as peças (cilíndricas ou prismáticas); o robô 1 pega uma peça cilíndrica para inserção e o robô 2 pega uma peça prismática para inserção. Depois da inserção, o mecanismo de rotação gira 180 graus para ser descarregado na nova unidade por um trabalhador. A movimentação dos robôs acontece simultaneamente e com tratamento de colisões.

Aplicando a metodologia desta pesquisa têm-se as seguintes funções:

- (1) Carga e inserção da peça cilíndrica por o robô 1
- (2) Carga e inserção da peça prismática por o robô 2
- (3) Repita-se três vezes as duas operações anteriores
- (4) Descarregamento / carregamento do pallet por o trabalhador

A nova FAC apresenta as seguintes vantagens:

- (1) A configuração pode-se mudar com facilidade e rapidez.
- (2) A flexibilidade da FAC é maior.
- (3) Minimiza-se o tempo de montagem pela inserção simultânea dos robôs.
- (4) Minimiza-se o tempo de produção da FAC.
- (5) Visualiza-se a evasão das colisões antes de implantar-se a FAC.

#### **6. 12. Exemplo 4, FMC com um Robô e duas Máquinas CNC: Três variantes.**

Cutkovsky, Fussell, e Milligan (1990) apresentam três variantes de configuração do projeto conceitual de uma FMC para lotes de pequeno e médio tamanho, para mostrar a flexibilidade nas configurações dos sistemas produtivos robotizados. A FMC conceitual resultante é um sistema integrado de componentes que se pode modificar ou expandir de acordo com as necessidades do mercado e se tornar ainda mais complexa. Os autores não apresentam modelagem em sistema CAD das configurações, não apresentam simulação para validação da flexibilidade das configurações e não apresentam programação *off-line* dos robôs para validação da melhor opção.

É preciso insistir que Cutkovsky, Fussell, e Milligan (1990) desenvolvem as configurações em forma conceitual. Com a aplicação da metodologia proposta neste

trabalho de pesquisa, são criadas três variantes das configurações com sistemas CAD para visualização virtual, desenvolve-se simulação da movimentação dos robôs e mecanismos para cada uma das variantes e finalmente, desenvolve-se programação *off-line* dos robôs para validação do desempenho de cada uma das variantes. As variantes das FMC configuram-se com os mesmos componentes propostos pelos autores. Identificam-se as seguintes operações:

- (1) Posicionamento e inspeção de peças pelo trabalhador.
- (2) Pegar peças pelo robô da estação de carregamento / descarregamento.
- (3) Pegar peças pelo robô depois de processamento em CNC.
- (4) Carregar centro de usinagem CNC.
- (5) Processamento no centro de usinagem CNC.
- (6) Descarregar centro de usinagem CNC.
- (7) Carregar o torno CNC.
- (8) Processamento no torno CNC.
- (9) Transferir peça entre torno CNC e o centro de usinagem CNC pelo robô.

A primeira variante da FMC está configurada com um centro de usinagem CNC de eixo vertical, uma máquina ferramenta CNC de torneamento, um robô centralizado IRB 1400 que desempenha a função de carregamento / descarregamento, um computador central de controle e programação *off-line* das máquinas CNC e do robô. As peças são posicionadas na estação de carregamento / descarregamento por um trabalhador e são movimentadas pelo robô e posicionadas nos pontos de trabalho CNC (Figura 6. 26).

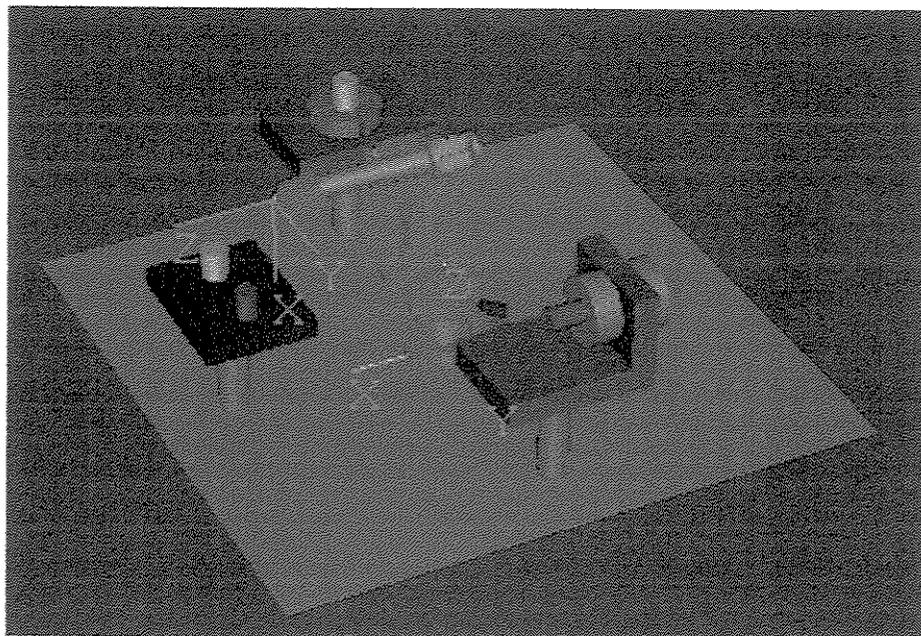
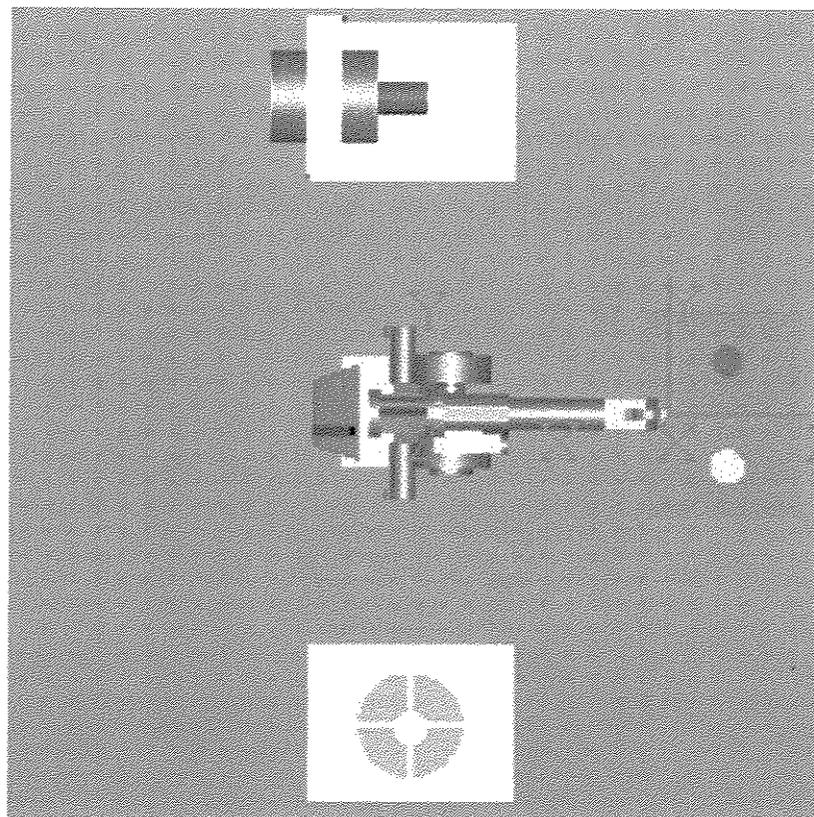


Figura 6. 26. Variante 1: FMC com robô centralizado

A segunda variante da FMC tem a mesma configuração da variante anterior com a diferença que o robô centralizado tem deslocamento entre as duas estações de trabalho CNC. A Flexibilidade desta configuração aumenta o espaço de trabalho do robô e pode-se minimizar a evasão de colisões, conforme é apresentado na Figura 6. 27.

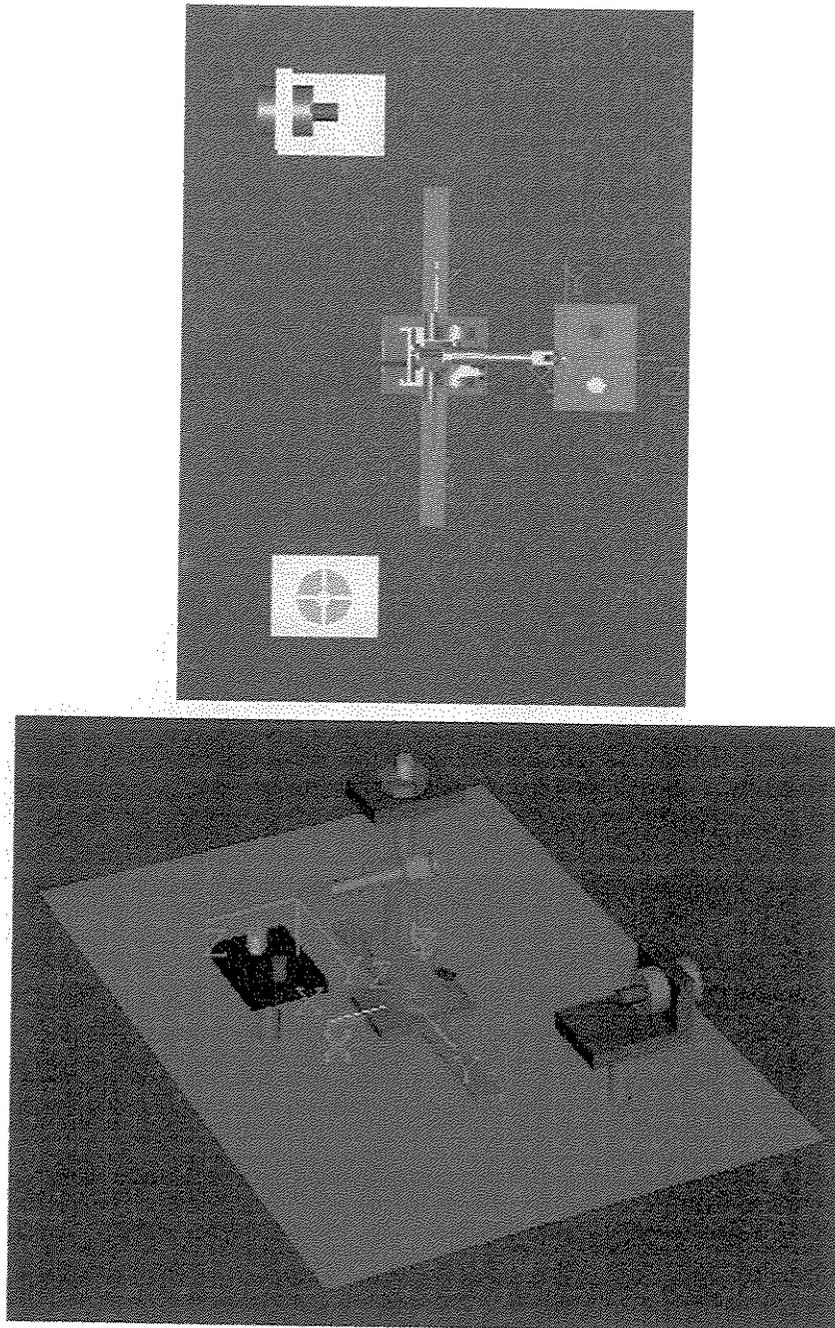


Figura 6. 27. Variante 2: FMC com robô centralizado e deslocamento

Na terceira variante configura-se a FMC com dois robôs de carregamento / descarregamento, uma esteira elétrica e duas estações de processamento CNC1 e CNC2. Uma peça chega por uma esteira elétrica e o robô 1 carrega / descarrega a peça entre a esteira elétrica e a estação CNC1 (e vice-versa depois de processada em CNC1), a peça é movimentada a seguinte estação de processamento pela esteira elétrica, onde o robô 2 carrega / descarrega a peça entre a esteira elétrica e a estação CNC2 (e vice-versa depois de processada em CNC2), e finalmente a peça é posicionada pelo robô2 na esteira elétrica para continuar seu processamento. Esta variante tem dois robôs, cada um dedicado a cada uma das estações de processamento CNC1 ou CNC2 é apresentada na Figura 6. 28

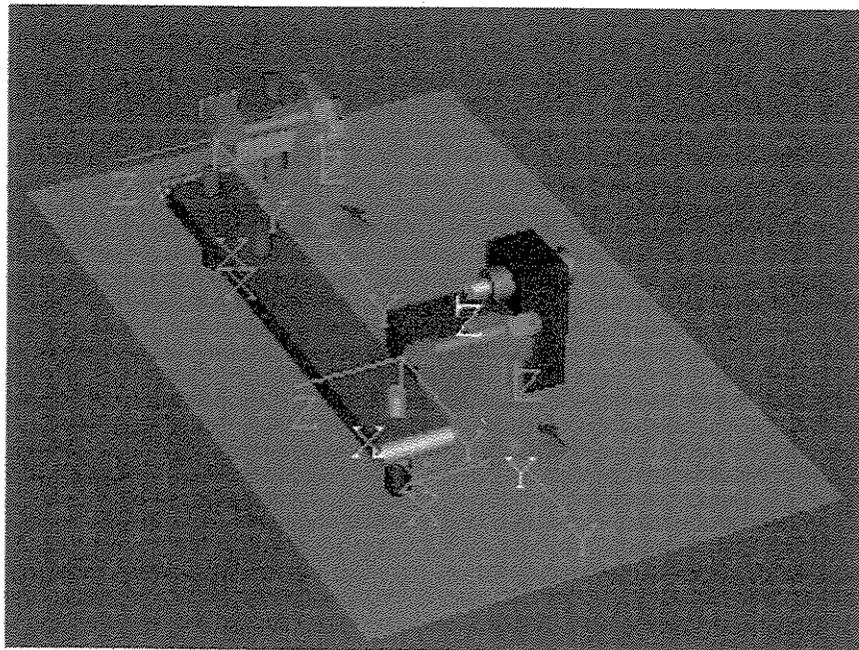
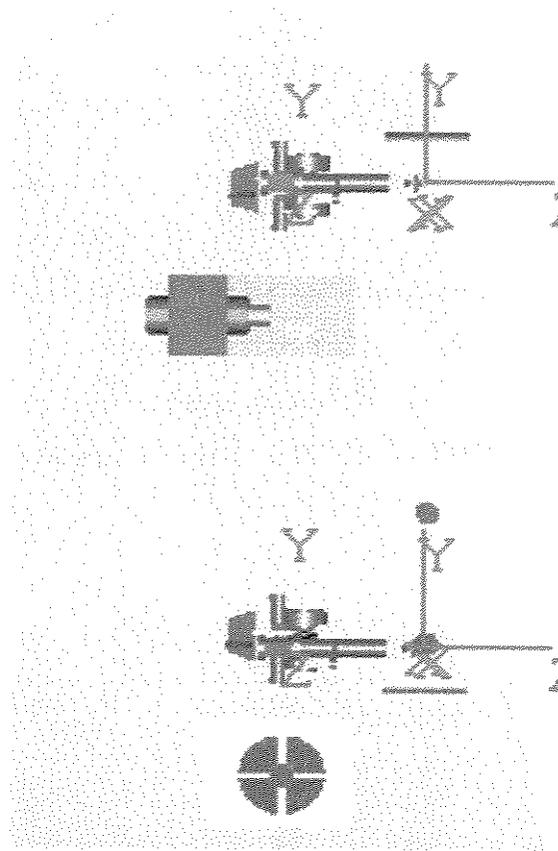


Figura 6. 28. Variante 3: FMC com dois robôs dedicados

As vantagens da aplicação da metodologia de simulação e programação *off-line* proposta, neste trabalho de pesquisa, podem ser resumidas como:

- (1) Podem ser configuradas diversas opções de FMC para o mesmo sistema produtivo a baixo custo antes de selecionar-se a configuração ótima ou perto da ótima.
- (2) Podem descrever infinitas trajetórias dos robôs antes de selecionar-se a trajetória ótima ou próxima da ideal.
- (3) Podem visualizar-se infinitas simulações da FMC antes de tomar-se uma decisão para implantar a FMC selecionada.
- (4) Pode desenvolver-se planejamento estratégico da FMC para avaliar os custos de investimentos antes de implementar-se uma solução definitiva.
- (5) Podem desenvolver-se estudos de flexibilidade, produtividade e qualidade da produção.

### **6. 13. Exemplo 5, FMS com um Robô e cinco Máquinas NC.**

Park (1995) apresenta simulação da movimentação do robô baseada no análise matemática das trajetórias dentro de um sistema produtivo de caminhões. A função do robô é movimentar peças entre as cinco estações NC. O principal objetivo é determinar a melhor regra de movimentação do robô baseada em estudos estatística e não – estatísticos. O sistema produtivo está configurado como:

- (1) um robô cilíndrico de três graus de liberdade posicionado mais ou menos no centro,
- (2) cinco estações de maquinado NC,
- (3) duas esteiras elétricas (uma de carregamento e outra de descarregamento),
- (4) uma câmara de visão.

O sistema produtivo é modelado usando linguagem de simulação SLAMII e a programação do robô com linguagem FORTRAN 77 no IBM / PC 386

O robô transporta uma peça de cada vez e pode se movimentar à esquerda da estação de carregamento e a direita da estação de descarregamento. Cada máquina NC pode realizar

um número de diferentes operações sobre as peças. Cada estação de processamento tem uma armazenagem temporal, depois a peça é transferida a seguinte estação para continuar sua seqüência de processamento. As peças esperam nas estações temporais de cada máquina e são movidas à próxima estação de acordo com a regra de prioridade: first – in – first – served.

As peças chegam em forma estocástica ao sistema produtivo pela esteira elétrica, fixadas no pallet. Um interruptor limite no final da esteira elétrica reconhece a presença da peça. Um código de barras em frente do pallet identifica o tipo de peça para definir seqüência de operação de processamento. O código de barras é lido por uma câmara de visão instalada na parte superior do robô. Uma peça processada por uma máquina CNC pode ser transferida pelo robô, quando a próxima máquina CNC está liberada o espaço de armazenagem temporal está liberado também. Processam-se cinco diferentes peças de trabalho e cada uma tem sua própria seqüência de maquina.

Park (1995) apresenta seis regras básicas de decisão de movimentação do robô:

**Regra 1 (first – in – first - out FIFO):** O robô move a primeira peça que necessita de movimentação entre todas as peças possíveis.

**Regra 2 (shortest move time first SMTF):** O robô move a peça com o tempo mais curto de movimentação entre todas as peças possíveis.

**Regra 3 (shortest remaining process time first SRPF):** O robô move a peça que tem o tempo total de processamento mais curto por realizar, entre todas as peças possíveis.

**Regra 4 (longest remaining process time first LRPF):** O robô move a peça com o tempo total de processamento mais longe por realizar, entre todas as peças possíveis.

**Regra 5 (smallest process sequence order first SPOF):** O robô move a peça com seqüência de processamento a realizar é mais curta, entre todas as peças possíveis.

**Regra 6 (largest process sequence order first LPOF):** O robô move a peça com seqüência de processamento a realizar é mais longe, entre todas as peças possíveis.

Antes de aplicar a metodologia proposta nesta pesquisa, é preciso especificar que a configuração de uma FMC acontece com máquinas ferramentas CNC com controle centralizado e não com máquinas ferramentas NC que tem controle descentralizado. Também que, uma FMC pode ter até três máquinas ferramentas CNC, acima deste número é um FMS. E também é necessário definir que, os FMS caracterizam-se porque tem integração física com um sistema de movimentação de materiais e integração computadorizada central para programação *off-line* e controle.

Agora, aplica-se a metodologia desenvolvida nesta pesquisa, sobre simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados em FMC:

(1) Converter as máquinas ferramentas NC em máquinas ferramentas CNC. O que significa integrar a programação e controle numa estação centralizada. Assim, pode-se configurar um FMS com cinco estações CNC em redor do robô e controle centralizado de todos os componentes.

(2) Modelagem CAD da configuração do FMS integrado por: (1) Robô IRB 1400 para movimentação de seis graus de liberdade, (2) cinco estações CNC, (3) Estação de carregamento de peças, (4) estação de descarregamento de peças, (5) armazenagem temporal das peças junto aos CNC em espera de processamento, (6) câmara de visão.

(3) Aplica-se a regra SRPF que representa: (A) trajetória perto da ótima para o variável tempo meio de fluxo mínimo das peças dentro do FMS e (B) trajetória perto da ótima para o variável tempo mínimo de movimentação do robô ou Makespan.

(4) Validação da regra SRPF a través da simulação e programação *off-line* do robô IRB 1400 em linguagem RAPID.

Na Figura 6. 29 são apresentados gráficos que ilustram a configuração do novo FMS proposto neste exemplo.

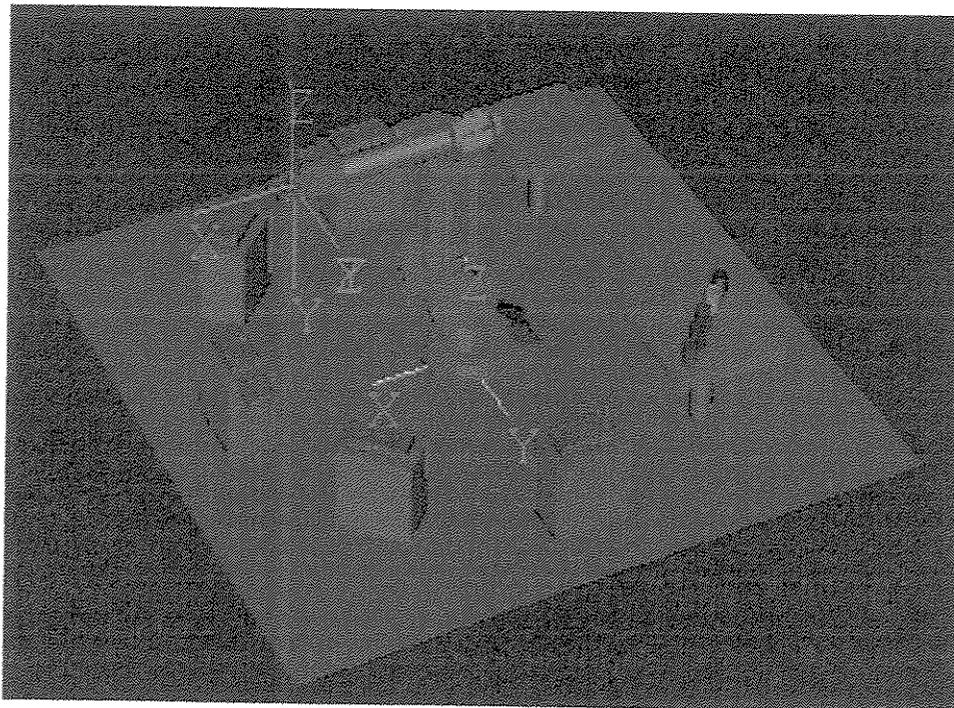
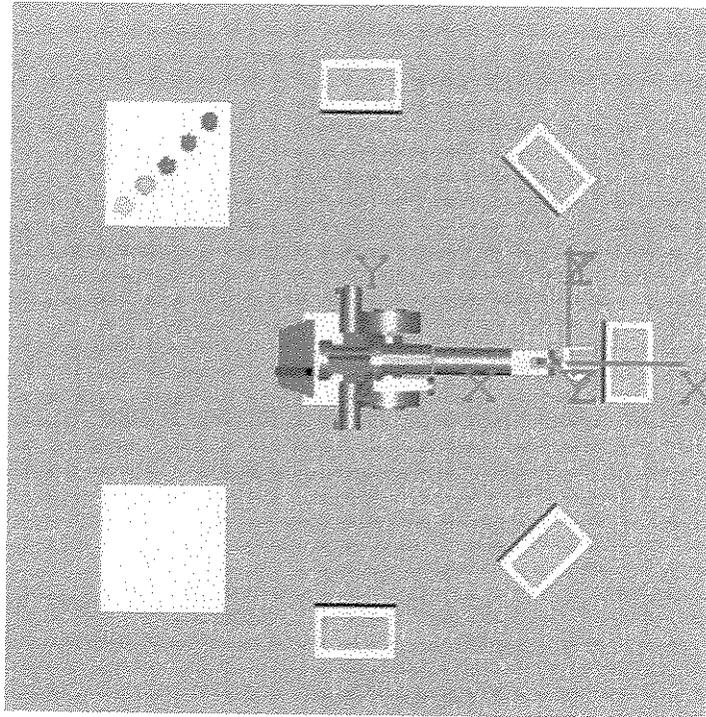


Figura 6. 29. FMC com um robô e cinco máquinas NC

Aplicando a metodologia proposta neste trabalho de pesquisa sobre simulação e programação *off-line* do robô dentro do FMS tem-se as seguintes vantagens:

(1) Pode projetar-se um FMS com todos os dispositivos centralizados numa estação computadorizada para programação *off-line* do robô integrado na programação *off-line* dos sistemas CAD / CAM das máquinas ferramentas CNC e controle centralizado.

(2) Com o sistema CAD do Workspace modela-se o FMS e pode visualizar-se numa estação gráfica antes de sua implantação real.

(3) Aplicando o diagrama de GANTT pode-se identificar a rota ótima de movimentação do robô.

(4) Com o algoritmo de movimentação e simulação do Workspace pode-se movimentar a trajetória ótima do robô antes de carregar-se no sistema real.

(5) Com o sistema de programação *off-line* do Workspace, pode-se obter em forma automática a programação da rota ótima do robô dentro do FMS em linguagem RAPID para o robô IRB 1400.

(6) Com os dados estatísticos da pesquisa previa pode desenvolver-se planejamento estratégico da produção e simulação gráfica com Workspace.

(7) A aplicação da metodologia desenvolvida nesta pesquisa para a modelagem, simulação e programação *off-line* dos robôs dentro dos FMS é uma ferramenta flexível e econômica para conhecer o funcionamento dos sistemas produtivos em forma rápida e visual, antes de projetar-se em forma real.

(8) A aplicação da metodologia desenvolvida nesta pesquisa para a modelagem, simulação e programação *off-line* dos robôs dentro dos FMS, otimiza a produtividade e competitividade das indústrias de médio e pequeno porte, no mercado internacional.

A Tabela 6. 3 apresenta um resumo do desenvolvimento de FMC baseadas na metodologia para simulação e programação *off-line* de robôs e dispositivos mecatrônicos integrados.

Tabela 6. 3. Resumo de desenvolvimento de FMC baseadas na metodologia para simulação e programação *off-line* de robôs

| <b>Exemplo</b> | <b>Robôs</b>                   | <b>Operação</b>                           | <b>Mecanismo</b> | <b>Dispositivo</b>  | <b>GDL FMC</b> | <b>Cinemática</b>        | <b>POL</b>                   | <b>Língua-gem</b> |
|----------------|--------------------------------|---|------------------|---------------------|----------------|--------------------------|------------------------------|-------------------|
| 1              | IRB1400<br>6R<br>IRB1500<br>6R | Soldagem<br>Soldagem                      | PRR              | C/D                 | 15             | MTH-<br>Quarter<br>nions | Ind.<br>Sim.<br>Seq.<br>Int. | RAPID             |
| 2              | IRB1400<br>6R<br>IRB1400<br>6R | Movimen-<br>tação<br>Soldagem             | 1R               | Esteira<br>Elétrica | 13             | MTH-<br>Quarter<br>nions | Sim.                         | RAPID             |
| 3              | IRB1400<br>6R<br>IRB1400<br>6R | Movimen-<br>tação e<br>Inserção<br>juntos | 1R               | C/D                 | 13             | MTH-<br>Quarter<br>nions | Sim.                         | RAPID             |
| 4a             | IRB1400<br>6R                  | Movimen-<br>tação                         | RPP<br>RRR       | C/D                 | 12             | MTH-<br>Quarter<br>nions | Ind.                         | RAPID             |
| 4b             | IRB1400<br>7R                  | Movimen-<br>tação                         | RPP<br>RRR       | C/D                 | 12             | MTH-<br>Quarter<br>nions | Ind.                         | RAPID             |
| 4c             | IRB1400<br>6R<br>IRB1400<br>6R | Movimen-<br>tação<br>Movimen-<br>tação    | RPP<br>RRR       | Esteira<br>Elétrica | 18             | MTH-<br>Quarter<br>nions | Sim.                         | RAPID             |
| 5              | IRB1400                        | Movimen-<br>tação                         | 5(RRR)           | C/D                 | 21             | MTH-<br>Quarter<br>nions | Ind.                         | RAPID             |

## **Capítulo 7**

### **Conclusões e sugestões**

#### **7. 1. Conclusões**

Os robôs são máquinas mecatrônicas programáveis, re-programáveis, e flexíveis que podem classificar - se ao nível da automatização programável e flexível. Também representam o passo para que os SAP se transformem em FMS. Pode-se afirmar, que um FMS é um SAP com robôs e sistema de controle centralizado computadorizado.

A metodologia apresentada nesse projeto de pesquisa é uma ferramenta direcionada aos sistemas de manufatura de pequeno e médio porte, e pode transformar-se em FMC ou para que se possam configurar novas FMC a partir de equipamento disponível no mercado, permitindo que as indústrias possam competir a nível internacional, com flexibilidade, com alto nível de produtividade, e com qualidade na produção.

A integração dos robôs nos sistemas de manufatura automatizados é uma realidade na indústria dos países desenvolvidos. É preciso que os países em desenvolvimento e processo de crescimento possam integrar e estender o campo das aplicações em robótica para a indústria, de modo a tornar o trabalho humano mais produtivo e menos exaustivo.

## 7. 2. Sugestões para trabalhos futuros

### 7. 2. 1. Desenvolvimento da CIM

Como sugestão de trabalhos futuros, visando uma continuidade deste projeto de pesquisa, torna-se necessário o desenvolvimento da manufatura integrada por computador (CIM) em ambiente virtual, através da integração da programação *off-line* de robôs e programação *off-line* dos sistemas CAD / CAM em FMC. Neste caso, propõe-se a implementação dos robôs numa estação computadorizada para sua programação *off-line* e controle e integração das máquinas ferramentas CNC em outra estação computadorizada para sua programação *off-line* e controle, separadamente.

Ao mesmo tempo, as duas estações computadorizadas dedicadas deverão se integrar a uma terceira estação computadorizada para efetuar programação *off-line*, controle e supervisão dos processos e planejamento estratégico ou *scheduling*, conforme mostra a Figura 7.1.

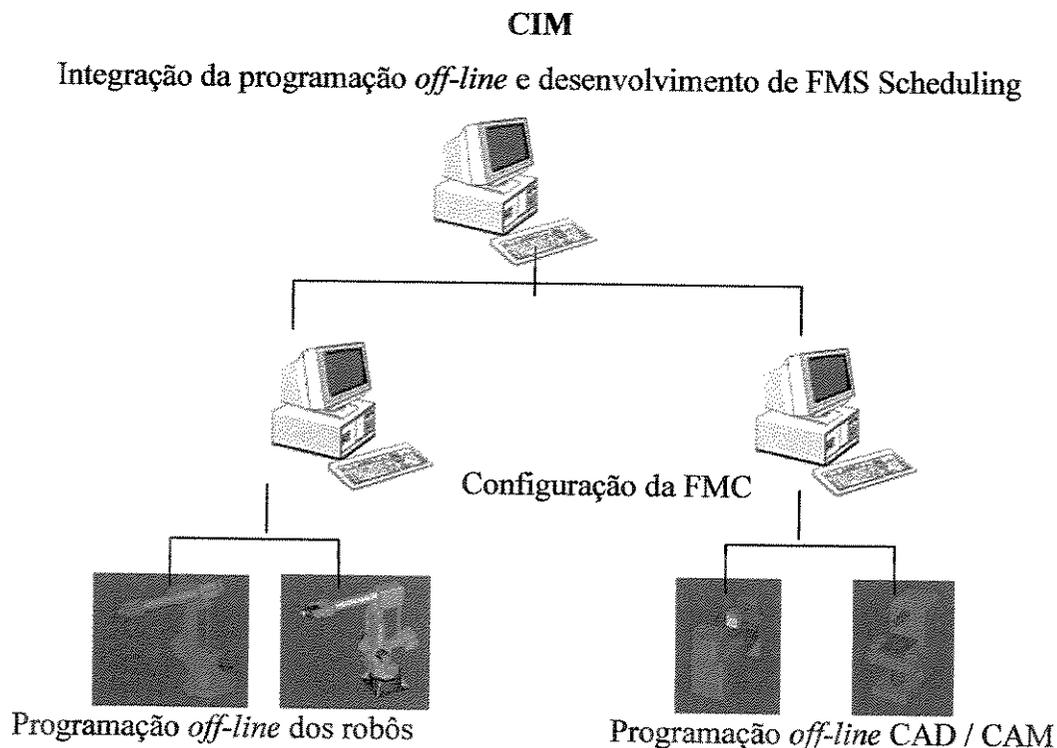


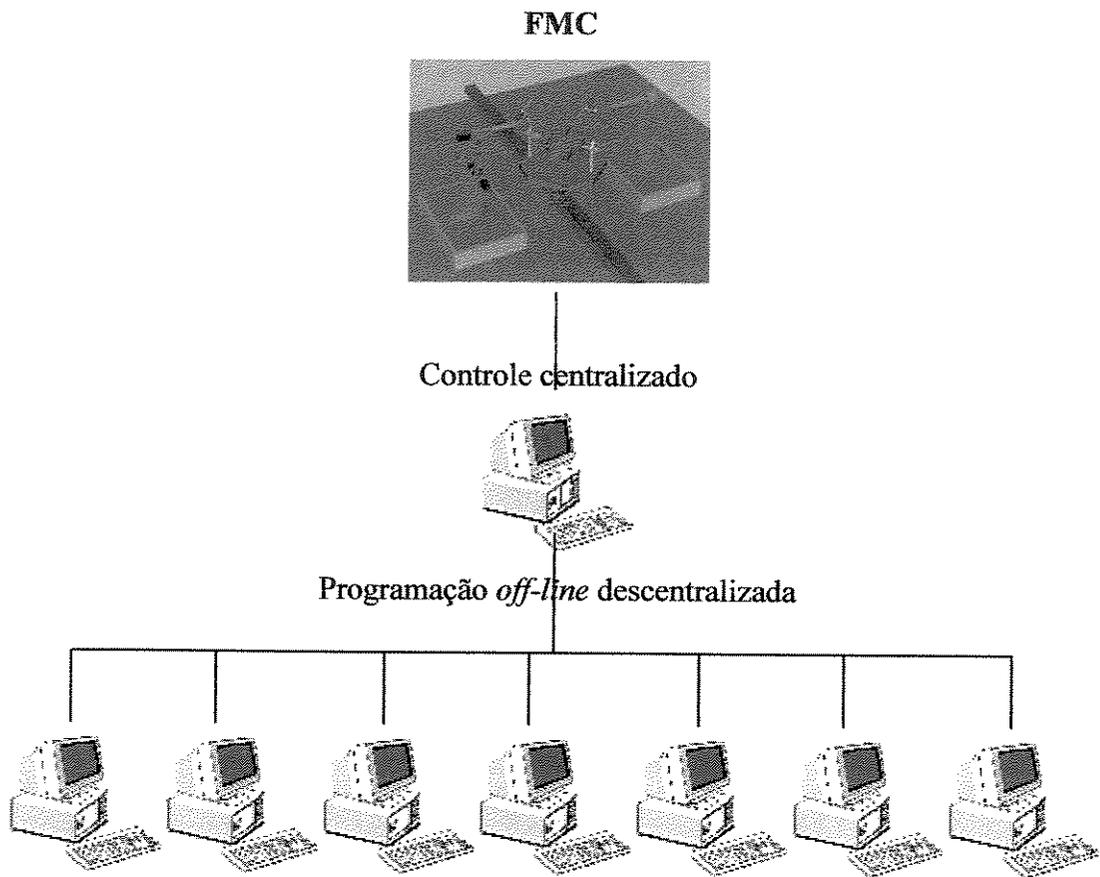
Figura 7. 1. Integração - CIM

### 7. 2. 2. Ensino de Robótica e FMS no Ambiente Virtual

Outra sugestão é a integração do computador central da FMC da UNICAMP com as estações computadorizadas do Laboratório de Automatização Integrada e Robótica - LAR da Faculdade de Engenharia Mecânica da UNICAMP através da rede LAN. Desta forma integram-se todos os componentes do LAR com os componentes da FMC para desenvolver ensino, palestras e pesquisas no ambiente virtual com arquitetura aberta encaminhada ao desenvolvimento da CIM.

A Figura 7. 2 mostra uma arquitetura para o desenvolvimento desta proposta. Neste ambiente virtual podem desenvolver-se:

- (1) Modelagem e projetos mecatrônicos de robôs e da FMC.
- (2) Simulação e programação *off-line* de robôs industriais.
- (3) Simulação e programação *off-line* dos CAD / CAM.
- (4) Planejamento estratégico dos FMS (FMS Scheduling).
- (5) Robótica avançada e novas aplicações.
- (6) Manufatura integrada por computador CIM.



Tarefas, aulas e palestras que podem desenvolver-se:

- (1) Modelagem e Projeto de novos robôs e novas FMC.
- (2) Simulação e Programação *off-line* dos robôs industriais.
- (3) Simulação e Programação *off-line* dos sistemas CAD / CAM.
- (4) Planejamento dos FMS (FMS Scheduling)
- (5) Manufatura integrada por computador CIM.

Figura 7. 2. Ensino de robótica no ambiente virtual

## Referências Bibliográficas

- Ahuactzin, J. M., Gupta, K. K., The kinematics road map: a motion planning based global approach for inverse kinematics for redundant robots. *IEEE Transactions on Robotics and Automation*. Vol. 15, Nº 4, August, 1999.
- Aly, A. A., Subramaniam, M., Design of an FMS Decision Support System. *Int. J. Prod. Res.*, Vol. 31, No. 10, pp. 2257-2273, 1993.
- Angeles, J., Alivizatos, A., Zsombor-Murray, P. J., The synthesis of smooth trajectories for pick-and-place operations. *IEEE Trans. Syst., Man., Cybern.*, Vol. 18 pp. 173-178, Jan/Feb, 1988.
- Angeles, J., *Fundamentals of Robotics Mechanical Systems: Theory Methods and Algorithms*. ISBN 0-387-85 540-7. Springer - Verlag. New York, 1997.
- Askin, R., Standridge, R., *Modeling and manufacturing systems*, John Wiley and Sons, 1993.
- Barrientos, A., Peñin, L., Balaguer, C., Aracil, R., *Fundamentos de Robótica*. Madrid McGraw Hill. 1997.
- Beggs, J. S., *Kinematics*. Springer - Verlag, Berlin, 1983.
- Billingsley, J., *Robots and automated manufacture*. Short Run Press Ltda., England, 1985.

- Booney, M. C., Marshall, R. L., Green, J. L., *Off Line Programming Using the Grasp Robot Simulation System. Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. pp. 171-180. (North Holland), 1987.
- Boud, A. C., Steiner, S. J., A new method for off-line programming: applications and limitations using a virtual environment. *Factory 2000 - The Technology Exploitation Process, Fifth International Conference on Factory 2000*, pp 450-455, Cambridge, UK, 2000.
- Brooks, R. A., Planning collision free motions for pick and place operations. *Int. J. Robot.* Vol. 2, N° 4. pp. 19-44, 1983.
- Buxbaum, H. J., The robot-based flexible manufacturing work cell as a strategic factory automation concept. *ISPE/EFACT International Conference of CAD/CAM, Robotics and Factories of the Future*, Pereira, Colombia, pp. 45-50, 1995.
- Cameron, S., *Modeling solids in motion*. University of Edinburgh, 1984. Tese (Doutorado).
- Campos, R. de., Implementação de um algoritmo para estimativa de parâmetros do modelo cinemático de robôs. Universidade Estadual de Campinas, 1993. Tese (Mestrado)
- Carvajal, J. H., Design and Modeling of Flexible Manufacturing Systems FMS: A review. *International Symposium on Computer Aided Process Engineering ISCAPE 2000*, Cartagena, Colombia, January 25-28, 2000.
- Carvajal, J. H., *Robótica: Aproximación al diseño mecatrónico*. Universidad del Atlántico. Colombia, Barranquilla, 2000.
- Carvajal, J. H., Sistemas de Manufactura Flexibles: Revisión analítica de métodos de diseño. *Segundo Congreso Nacional de Ingeniería Mecánica*, Universidad Nacional de Colombia, Bogotá, Octubre 24-27, 2001.

- Cerqueira, J. de J. F., Análise de uma classe de neurônios artificiais para aplicações em sistemas robóticos. Universidade Estadual de Campinas, 1996. Tese (Mestrado)
- Cerqueira, J. de J. F., Identificação de sistemas dinâmicos usando redes neurais artificiais: Uma aplicação a manipuladores robóticos. Universidade Estadual de Campinas, 2001. Tese (Doutorado)
- Chan, S. F., Kwan, R., Post-processing methodologies for off-line robot programming within computer integrated manufacture. *Journal of Materials Processing Technology*. Vol. 139, Issue 1-3, pp 8-14, 20 August, 2003.
- Chao, D. Y., Wang, D. T., XPN-FMS: a CAD tool for FMS modeling, analysis, animation, and simulation using Petri Nets and X window. *Inter. J. of FMS*, pp.339-360, 1995.
- Chi-Keng, T., Multiple robot coordination and programming. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*. Sacramento, California, pp. 978-985, 1991.
- Chow, J. C. K., Quaternion Kinematic and Dynamic Differential Equations. *IEEE Transactions on Robotics and Automation*, Vol. 8, Nº. 1, February, 1992.
- Cosso, S. G., Integração de ferramentas de automação direcionadas a aplicações de telerobótica: Implementação de um sistema de supervisão e controle num sistema teleoperado. Universidade Estadual de Campinas, 2002. Tese (Mestrado)
- Coutinho, L. A. de F., Um ambiente integrado de desenvolvimento de software aplicado a robótica. Universidade Estadual de Campinas, 1993. Tese (Mestrado)
- Craig. J. J., *Introduction to Robotics. Mechanics and Control*. Reading Massachusetts. Addison-Wesley Publishing Company, 1986.

- Cruz, J. M. D de., Projeto e desenvolvimento de um sistema de geração automática de trajetórias para manipuladores. Universidade Estadual de Campinas, 1993. Tese (Mestrado)
- Cutkovsky, M. R., Fussell, P. S., Milligan, R.Jr. The design of a flexible machining cell for a small batch production. *Journal of Manufacturing Systems*, Vol. 3, No. 1, pp.39-50, 1990.
- Dafe, O. L., Off line Programming and Simulation. *Computer Based Learning Unit*, University of Leeds, UK, 1995.
- Das, S. K., A facility layout method for FMS, *Int. J. Prod.*, Vol. 31, No. 2, pp.279-297, 1993
- David, S. A., Modelagem, simulação e controle de robôs flexíveis. Universidade Estadual de Campinas, 1996. Tese (Mestrado)
- Dekleva, J., Gaber, M., Two-phase method for designing FMS, *Int. J. Prod.* Vol. 32, No. 3, pp. 683-691, 1994.
- Denavit J., Hartenberg R. S., A kinematics notation for lower-pair mechanism based on matrices. *ASME J. Appl. Mech.*, pp. 215-221, June 1955.
- Dias, C. H., Implementação experimental de um supervisor de controle para robôs industriais. Universidade estadual de Campinas, 1993. Tese (Mestrado)
- Dicesare, F., Harhalakis, G., Proth, J. J., Silva, M., Vernadat, F. B., *Practice of Petri Nets In Manufacturing*. Chapman & Hall, London 1993.
- Djuric, Ana., *Economical Industrial Workcell Modeling Simulation and Layout Design*. University of Windsor, Windsor, Ontario, Canada. 1999. Tese (Mestrado).

- Edkins M., Smith C. R. T., The Paractical Problems Involved in Off line Programming a Robot from CAD Systems. *IEE Control Engineering Series 28*. Series Editors: Nicholson and Swanick. Peter Peregrinus Ltda. pp. 29-40, London, UK, 1985.
- Edwall, C. W., Ho, C. Y., Pottinger, H. J., Trajectory generation and control of a robot arm using splines functions. *Robots VI*. pp. 421-445, 1982.
- Egbelu, P. J., Economic design of unit load-based FMSs employing AGVs for transport. *Int. J. Prod. Res.* Vol. 31, No. 12, pp. 2753-2775, 1993
- Engelbert, J. F., *Robotics in practice*, AMACOM American Management Association, New York 1980.
- Fayan, B. L., Estudo e especificação de um supervisor de controle para um robô industrial. Universidade Estadual de Campinas, 1992. Tese (Mestrado)
- Fernandez, V. T., Analise e proposta de desenvolvimento de um robô manipulador utilizando a estrutura do paralelogramo. Universidade Estadual de Campinas, 1997. Tese (Mestrado)
- Fu, K. S., Gonzalez, R. C., Le, C. S. G., *Robotics: Control, Sensing, Vision and Intelligence*. NYC. , McGraw Hill, Inc. 1988.
- Fuller, J., *Robotics: Introduction, Programming, and Projects*. Prentice Hall, 2/e. 1999
- Groover, M. P., *Automation, Production Systems, and Computer-Integrated Manufacturing*. New Jersey: Prentice Hall, 2000. 856p.
- Groover, M., *Fundamentals of Modern Manufacturing: materials, processes, and systems*. Prentice Hall. NYC. 1996.
- Gupta, K. C., *Mechanics and Control of robots*. Springer. 1999.

- Hamilton, W. R., *Elements of Quaternions*, Chelsea Publishing Co. New York, 1969.
- Hamilton, W. R., On a new Species of Imaginary Quantities connected with a theory of Quaternions. *Proceeding of the Royal Irish Academy*, Vol. 2, pp. 424-434, 1843.
- Hamilton, W. R., On Quaternions. *Proceeding of the Royal Irish Academy*, Vol. 3, pp.1-16, 1844.
- Hermini, H. A., Uma contribuição ao estado de mão-lineariedades na modelagem e controle de manipuladores. Universidade estadual de Campinas, 1996. Tese (Mestrado)
- Hervella, C., Projeto e desenvolvimento de um controlador lógico programavel flexivel para controle de manipuladores e robôs industriais. Universidade Estadual de Campinas, 1997. Tese (Mestrado)
- Huang, H-K, Lin, G. C. I., Rapid and flexible prototyping through a dual-robot workcell. *Robotics and Computer Integrated Manufacturing* Vol 19, pp 263 - 272, 2003.
- Il Kim, K., Zheng, Y. F., Unknown load distribution of two industrial robots. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*. Sacramento, California, pp. 992-997.
- Jeng, M., Modular synthesis of Petri Nets for modeling FMS, *Inter. J. of FMS*, (7) pp.287-310, 1995.
- Jiang, K. *et al.*, Assembly scheduling for an integrated two-robot work cell. *Robotics and Computer Integrated Manufacturing*, Vol.13, N<sup>o</sup>. 2, pp. 121-129, 1997.
- Kaku, B., Rachamadugu, R., Layout design for FMS. *Europe. J. Oper. Res.* No.57, pp.224-230. 1992

- Knapp, D. S. Use of virtual reality in Off-line robot programming. Iowa State University. 1994.  
Thesis (Doctorate)
- Kuipers, J. B., *Quaternions and Rotation Sequences*. Princeton University Press. Princeton. New Jersey. 1999.
- Lend, D. Y., Chen, M., Robot trajectory planning using simulation. *Robotics and Computer Integrated Manufacturing*, Vol. 13, N° 2, pp. 121-129, 1997.
- Li, Y. F., Ho, J., Li, N., Development of a physically behaved robot work cell in virtual reality for task teaching. *Robotics and Computer Integrated Manufacturing*. Vol 16 pp. 91 –101, 2000.
- Li-Chung T. W., Ming J. K., Dynamic load – carrying capacity and inverse dynamics of multi cooperating robots manipulators. *IEEE Trans. On Robotics and Automation*, Vol. 10, N° 1, pp. 71-77. Feb. 1994.
- Lin, C. S., Chang, P. R., Luh, J. Y. S., Formulation and optimization of cubic polynomial joint trajectories of industrial robots. *IEEE Trans. Contr.*, Vol. AC-28, pp. 1966-1073, Dec. 1983.
- Lozano-Perez, T., Automatic planning of manipulator transfer movement. *IEEE Trans. Syst., Man., Cybern.*, Vol. SMC-11, pp. 661-698, 1981.
- Luh, J. Y. S., Lin, C. S., Approximate joint trajectories for control of industrial robots along Cartesian path. *IEEE Trans. Syst., Man., Cybern.*, Vol. SMC-14, pp. 444-450, May/June 1984.
- Maccarthy, B. L., Liu, J., A new classification scheme for FMS. *Int. J. Prod. Res.*, Vol. 31, No.2, pp. 299-309, 1990.

- Madrid, M. K., Controle de trajetórias contínuas por seccionamento em sub-trajetórias usando inteligência artificial num robô multitarefas. Universidade Estadual de Campinas, 1994. Tese (Doutorado)
- Madrid, M. K., Robô manipulador mecânico TRRR para posicionamento espacial com controle digital hierárquico a microprocessadores. Universidade Estadual de Campinas, 1988. Tese (Mestrado)
- Mahadevan B., Narendran T.T., A hybrid modeling approach to the design of an AGV-based material handling system for an FMS. *Int. J. Prod. Res.*, Vol. 32, No. 9, pp.2015-2030, 1994.
- Martin R. R., Aspects of the interference problems. *Proceedings of the joint Anglo-Hungarian Seminar on CAD*, 1985.
- Mellichamp. J. M., Kwon O-J., Wahab A. F. A., FMS Designer: An expert system for FMS design. *Int. J. Prod. Res.*, Vol. 28, No. 11, pp. 2013-2024, 1990.
- Meng, G., Husman, L., Vullum, N. Friesel, A., Project in robotics at the Copenhagen University College of Engineering; *IEEE International Conference on Robotics and Automation, 2004*. Proceedings. ICRA '04. 2004
- Mertins, K., Wieneke-Toutaoui, B., State of the art in FMS design. *Production planning & control*, Vol. 2, No.2, pp. 155-159, 1991.
- Miranda. M. F., Controle de um servomecanismo por um microcomputador dedicado: uma contribuição ao estado de controladores para robôs industriais. Universidade Estadual de Campinas, 1992. Tese (Mestrado)
- Moore, K. E., Gupta, M., Petri Nets models of flexible and automated manufacturing systems: a survey. *Inter. J. Of Prod. Res.*, Vol. 34, No. 11, 1996.

- Muller, A. Collision avoiding continuation method for the inverse kinematics of redundant manipulators. *IEEE International Conference on Robotics and Automation, 2004*.  
Procedeeng. ICRA '04. 2004
- Muther, R., *Systematic Layout Planning*, Canners Books, Boston 1973.
- Nobre, F. S. E., Projeto e análise de controladores nebulosos e sua aplicação para controle de juntas robóticas. Universidade Estadual de Campinas, 1997. Tese (Mestrado)
- Nof, S., *Handbook of Industrial Robotics*. Jhon Wiley and Sons. NY. 1985.
- Nogueira, R. G., Contole de posição e orientação de um manipulador atraves de um mause especial. Universidade Estadual de Campinas, 1995. Tese (Mestrado)
- Ohfugi, A. S., Garras articuladas para robôs manipuladores: Analise cinamática e de foças, sensoreamento e controle de posição e esforços da pressão de objetos. Universidade Estadual de Campinas, 1991. Tese (Mestrado)
- Oliveira, C. de., Proposta de implementação de um sistema de controle didactico baseado na modelagem dinâmica de robôs industriais. Universidade Estadual de Campinas, 2003. Tese (Doutorado)
- Park, Y. B., A simulation analysis of robot service movements in FMC. *Int. J. Produd. Res.* Vol. 33, No. 2, pp.539-58, 1995.
- Passeto, F., Desenvolvimento de um aplicativo para simulação e controle de manipuladores robóticos com ênfase em aplicações didaticas. Universidade Estadual de Campinas, 2000. Tese (Mestrado)

- Paul, R. P., *Robot manipulators: mathematics, programming and control*. MIT Press, Mass. Cambridge 1981.
- Paul, R., Manipulator Cartesian path control. *IEEE Trans. Syst., Man, Cybern.* Vol. SMC-9, pp.702-711, Nov/Dec 1979.
- Pereira, F. E. D., Modelagem Empresarial: Uma abordagem de Integração de Empresas – Aplicação da Metodologia CimOsa/rg à Plataforma PIPEFA. Universidade Estadual de Campinas, 2001. Tese (Mestrado)
- Perez, A, McCarthy, J. M. Dual Quaternions Synthesis of Constrained Robotic Systems. *Journal of Mechanical Design*. Vol. 126, Issue 3, pp. 425-435. May, 2004.
- Pfeiffer, F., Rainer, J. A., Concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, Vol. RA-3, N° 2, pp. 115-123, 1987.
- Pieper, D. L., Roth, B., The kinematics of manipulators under computer control. *Proc II Intel. Congr. Theory of Machines and Mechanisms*, Vol. 2, pp. 159-168, 1968.
- Pimenta, K. B., Controle de juntas robóticas usando controlador preditivo generalizado adaptativo direto. Universidade Estadual de Campinas, 2003. Tese (Doutorado)
- Pire, J. N., Sá da Costa, J. M. G. Objected – oriented and distributed approach for programming robotic manufacturing cells. *Robotics and Computer Integrated Manufacturing*. Vol 16, pp 29 – 42 , 2003.
- Pollman W., Dzembitzki H., Off – line Programming of Industrial Robots. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP. pp. 7-19, 1987.

- Pritschow G., Storr A., Grulher G., Schumacher H., Off – line Programming System With Geometrical Data Recording by Manually Guided Industrial Robot. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP, pp. 53-64, 1987.
- Rawat, K. S., Massiha, G. H. A hands-on laboratory based approach to undergraduate robotics education. *IEEE International Conference on Robotics and Automation, 2004*, Proceedings. ICRA '04. 2004.
- Rehg J., *Introduction to Robotics in CIMs*. Prentice Hall, 4/e, 440p. 2000.
- Reinhart, G., Köhne, T., Design of Autonomous Manufacturing Systems. *Proceeding CARS & FOF*. Pereira, Colombia 1996.
- Renteria, A., Rivas, M., *Robótica industrial: Fundamentos y aplicaciones*. Mc Graw Hill, EVE Ente Vasco de la Energía e Iberdiola. Madrid 2001.
- Rhodes D. J., Stenson E. H., Blanchfield P., Collision avoidance between robots operating in the same cell. *IEE Control Engineering Series 28*. Series Editors: Nicholson and Swanick. Peter Peregrinus Ltda. pp. 99-110. London, UK. 1985.
- Rivin, E., *Mechanical Design of Robots*. NYC. McGraw Hill Book Company. 1986.
- Sa, C. E. A. de., Desenvolvimento e implementação de um programa computacional para a supervisão e controle de manipuladores robóticos. Universidade Estadual de Campinas, 2000. Tese (Doutorado)
- Sa, C. E. A. de., Implementação de métodos numéricos para la resolução do problema cinemático inverso de robôs com ênfase em controle de posição. Universidade Estadual de Campinaas, 1996. Tese (Mestrado)

- Saramago, M. A. P., Projeto e desenvolvimento de um sistema de calibração e medida de posição para robôs industriais. Universidade Estadual de Campinas, 1993. Tese (Mestrado)
- Sata T., Kimura F., Hiraoka H., Suzuki H., Fujita T., Comprehensive Modeling of Machine Assembly for Off – line Programming of Industrial Robots. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP, pp. 19–33, 1987.
- Scholliers, J., Van Brussel, H., Design and off – line programming of a robotic tape winding cell. *Robotics and Computer Integrated Manufacturing*, V.12, Nº. 1, pp. 93-98. 1996.
- Sciavicco, L., Siciliano, B., *Modelling and Control of Robots Manipulators*. McGraw Hill, Inc. London 1997.
- Shan Y., Koren Y., Obstacle accommodation motion planning. *IEEE Transactions on Robotics and Automation*. Vol. 11, Nº 1, February 1995.
- Shetty, D., Kilk, R., *Mechatronics System Design*. PWS Publishing Company. Boston 1997.
- Silva, N. C. da., Seleção de atuadores e acionamento integrado a modelagem de manipuladores robóticos industriais, Universidade Estadual de Campinas, 1998. Tese (Doutorado)
- Souza, J. P. de., Implementação de algoritmos preditivos para controle de juntas roboticas. Universidade Estadual de Campinas, 2001. Tese (Doutorado)
- Souza, J. P. de., Procedimento automatico para aquisição e tratamento do movimento de um robô. Universidade Estadual de Campinas, 1992. Tese (Mestrado)
- Spong, M. W., Vidyasagar, M., *Robot dynamics and control*. John Wiley and Sons. NY. 1989.

- Spur G., Duelen, G., Krause F. L., An Integrated Approach Toward Off Line Robot programming. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP, pp. 181-190. 1987.
- Stadler, W., *Analytical Robotics and Mechatronics*. NYC. McGrawHill. 1995.
- Steffen, V., Saramago, S. F. P., Optimization techniques for off-line trajectories planning of robot manipulators. *Nonlinear dynamics, chaos, control and their applications to engineering sciences*. Published by AAM and ABCM. Vol. 3, pp. 363-368.
- Stobar R. K., Daily, C., The Use of the Simulation in the Off line Programming of Robots. Robots and Automated Manufacture. *IEE Control Engineering Series 28*. Series Editors: Nicholson and Swanick. Peter Peregrinus Ltda. pp. 11-28, London, UK. 1985.
- Stobart R. K. Collision Detection for Off Line Programming. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP, pp. 107-118, 1987.
- Stokic, D. M., Vukobratovic, M. K., Lekovic, D.B., Simulations of robots in Flexible Manufacturing Cells. *Robotics and Computer Integrated Manufacturing*, Vol. 8, N<sup>o</sup>. 1, pp. 1-8. 1997.
- Stone, H. W., *Kinematic modeling, identification, and control of robotic manipulators*. Kluwer Academic Publisher. Boston, Mass. 1987.
- Storr A., Schumacher, H., Programming Methods for Industrial Robots. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP. pp. 1-4., 1987.
- Sudhakarapandian, R., Jayakumar, V., Sornakumar, T., Cellular Manufacturing. A solution to layout problem. *CARS & FOF 2000*. Trinidad and Tobago.

- Suh, I. H., Shin, K. G., Coordination of dual robot arms using kinematics redundancy. *IEEE Trans. On Robotics and Automation*, Vol. 5, N° 2, pp. 236-245. April 1989.
- Takita, K., Sistema de sensoreamento de robôs industriais: espaço de juntas e espaço cartesiano. Universidade Estadual de Campinas, 1995. Tese (Doutorado)
- Tao, J. M., Luh, J. Y. S., Position and force control for two coordinating robots. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*, pp. 170-175. Sacramento, California.
- Taylor R. H. Planning and execution of straight line manipulator trajectories. *IBM J. Res. Develop.* Vol. 23, no. 4, pp. 424-436, 1979.
- Tetzlaff, U. A. W., A Model for the Minimum Cost Configuration Problem in FMS. *The International Journal of FMS*, Vol. 7, pp. 127-146, 1995.
- Torres, F., Pomares, J., Gil, P., Puente, S., Aracil, R., *Robots y Sistemas Sensoriales*. Prentice Hall, Madrid, 2002.
- Troy, J. J., Vanderploeg, M. J. Off-line robot programming in virtual environment. *ASME Design Engineering*. pp 963-968, 1995.
- Tzafestas, C. S., Alifragis, M., Palaiologou, N., Thomopoulos, C. S. A., Brahman, M., Exarchou, A. E. Development and Experimental Evaluation of a Remote Laboratory Platform for Teaching Robot Manipulator Programming. *International Conference on Engineering Education 2004, (ICEE 2004)*, Florida, USA, October 16-21, 2004.
- Uchiyama, M., Yamashita, T., Adaptive load sharing for hybrid controlled two cooperative manipulators. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*. pp. 978-985. Sacramento, California, 1991.

- Vieira, A. Do L., Desenvolvimento e implementação de controle de trajetória contínua em robô industrial de alto desempenho. Universidade Estadual de Campinas, 1996. Tese (Doutorado)
- Vigia, Kumar *et al.*, Control of contact conditions for manipulations with multiple robotic systems. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*. pp. 170-175. Sacramento, California.
- Weck M., Niehaus Th., Osterwinter M., An Interactive Model Based Robot Programming and Simulation Workstation. *Off-line Programming of Industrial Robots*. Storr & McWaters Editors. Elsevier Science Publishers B. V. (North Holland). IFIP, pp. 41-52. Sacramento, California, 1987.
- Westkämper, E., Schraft, R. D., Schweizer, M., Task-oriented programming of large redundant robot motion. *Robotics & Computer Integrated Manufacturing* Vol. 14. pp. 363 – 375, 1998.
- William, D. J., *Manufacturing Systems: An introduction to the technologies*. Second edition. Chapman & Hall, London, 1994.
- Workspace 5.01: *User Manual*. Flow Software Technologies.
- Xiaoping, Y., Coordination of two-arm pushing. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*. pp. 182-187. Sacramento, California, 1991.
- Xue, Q., Maciejewski, A. A., Sep, P. C., Determining the collision-free joint space graph for two cooperating robot manipulator. *IEEE Trans. Syst., Man., Cybern.* Vol. 23, pp. 285-293, 1993.
- Yazdani, B., Demystification of Concurrent Engineering. *CARS & FOF'96*. 1996.

Zha, X. F., Du, H., Generation and Simulation of Robot Trajectories in a Virtual CAD-Based Off-Line Programming Environment. *The International Journal of Advanced Manufacturing Technology*. Springer-Verlag London Ltd, Vol. 17, Number 8, pp 610-624, April, 2001

Zhou, M., *Petri Nets in Flexible and Agile Automation*. Kluwer Academic Publisher. London, 1995.

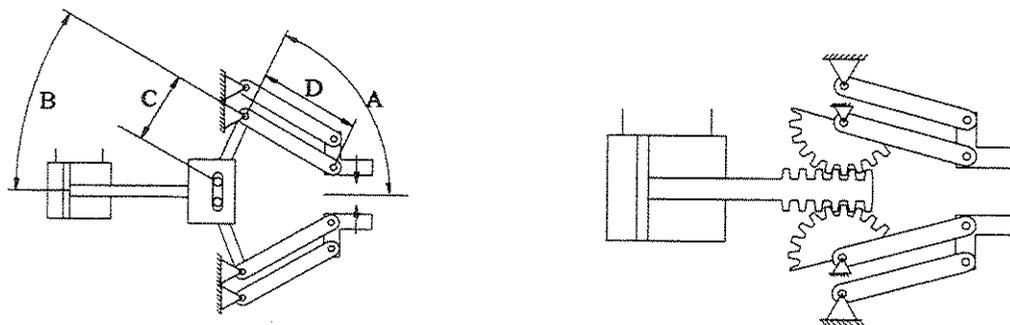
Zuofeng L., Tzyh-Jong T., Antal K. B., Dynamic workspace analysis of multiple cooperating robot arms. *IEEE Trans. Robotics and Automation*, Vol. 7, N<sup>o</sup>. 5, pp. 589-598, Oct. 1991.

## ANEXO I

### MECANISMOS PARA GARRA DE ROBÔS

#### Mecanismo de Quatro Barras Paralelas

O mecanismo faz que os dedos estejam em paralelo sempre. Esta é uma excelente configuração para agarrar uma variedade de peças cilíndricas por seu diâmetro externo ou diâmetro interno. Têm-se dois variantes: (1) Com guia. (2) Com pinhão e cremalheira. Com a segunda variante melhora a longitude de deslocamento dos dedos.



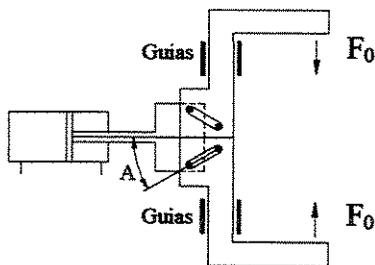
Mecanismo de quatro barras.

### Mecanismo com Levas em Paralelo

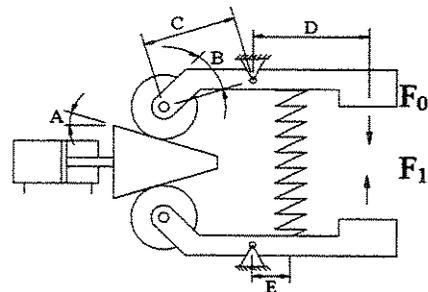
A garra tem um bloco com dois pinos que deslizam dentro das ranhuras das levas. As guias fazem que as levas tenham movimentação de abrir-fechar os dedos que dependem do ângulo entre as ranhuras. Os dedos estão normalmente abertos e para que estejam normalmente fechados se instala uma mola de compressão.

### Mecanismo com Leva Cônica

Os dedos giram ao redor de um pivô. No extremo livre do pistão monta-se a leva cônica que em sua movimentação de avance fecha os dedos. A mola abre os dedos quando a leva tem movimentação de retrocesso. Os dedos têm rodilhos que rodam sobre a leva para minimizar a fricção.



Mecanismo com levas em paralelo.



Mecanismo com leva cônica.

### Mecanismo com Deslocamento em Paralelo, Simples.

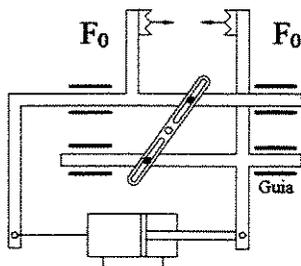
O mecanismo funciona com duas barras como guias e tem rolamentos para gerar movimentação em paralelo entre os dedos. Funciona com um cilindro pneumático de simples efeito. As barras - guias estão conectadas ao cilindro; uma ao pistão e a outra ao corpo do cilindro.

Quando o pistão se move em sua movimentação de retrocesso à barra conectada em seu extremo se move simultaneamente. Uma barra com ranhura conecta as duas barras - guias e transmite à a outra barra - guia um movimento na direção oposta, então,

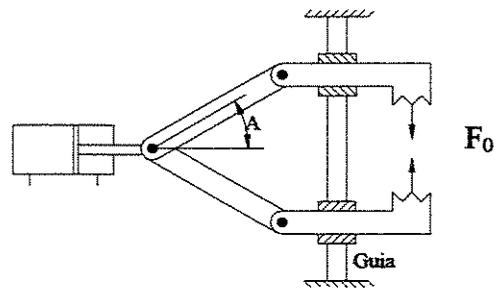
simultaneamente se movem os dedos para agarrar com movimentação em paralelo. Este movimento é de alta precisão.

### Mecanismo com Deslocamento em Paralelo - Dobre

Tem dois elementos acoplados aos dedos e uma barra - guia para o deslocamento em paralelo dos dedos. Quando o pistão do cilindro avança, os dedos têm movimentações uniformes, guiadas por a barra em paralelo. Este desenho apresenta boa longitude de abertura dos dedos, mas sua força é muito variável. Com aumento do angulo A se aumenta a força do mecanismo.



Mecanismo de deslocamento simples.



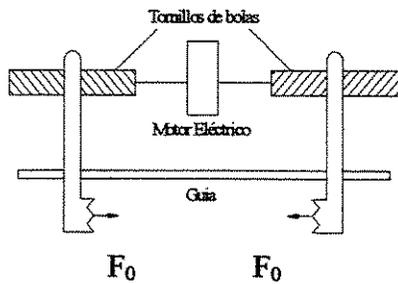
Mecanismo de deslocamento duplo.

### Braçadeira com Parafusos de Bolas

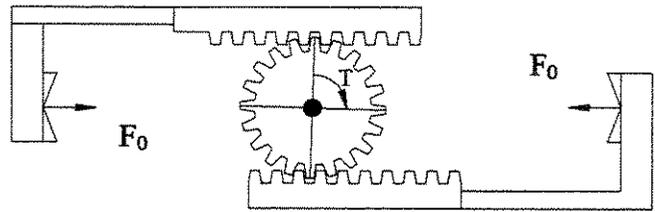
Os mecanismos têm dois parafusos de bolas, instalados nos eixos paralelos com giro em sentido contrario de um servo motor elétrico e tem uma barra - guia para facilitar a movimentação em paralelo dos dedos.

### Braçadeira com Dobre Cremalheira e Pinhão

O mecanismo usa um atuador pneumático rotacional ou servo motor elétrico para mover o pinhão e as duas cremalheiras. As cremalheiras estão apoiadas em rolamentos para obter movimentação paralela dos dedos da garra.



Braçadeira e parafusos de bolas.



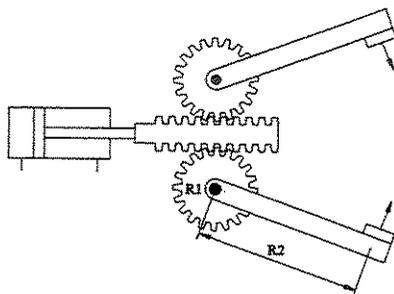
Braçadeira e dobre cremalheira e pinhão.

### Braçadeira com Cremalheira e Dobre Pinhão

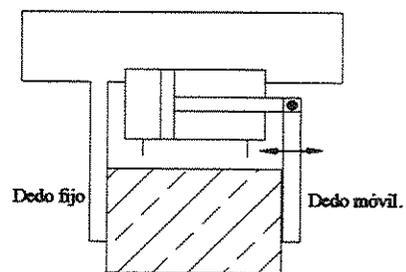
O mecanismo é uma variante do mecanismo anterior que usa dois pinhões para converter o movimento lineal que recebe a cremalheira do pistão do cilindro, em movimento rotacional, para mover os dedos em arco. Um cilindro hidráulico é utilizado como atuador.

### Braçadeira de Ação Simples

O mecanismo tem um cilindro pneumático como atuador com uma mordaza no extremo do pistão que desempenha a função de um dedo móvel e o outro dedo fixo.



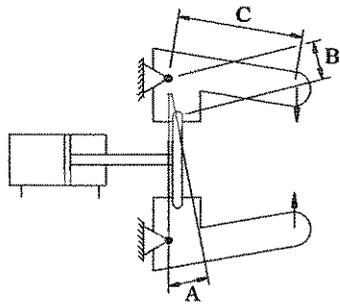
Braçadeira de cremalheira e dobre pinhão.



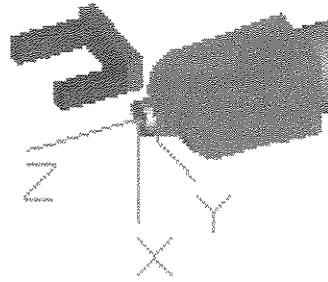
Braçadeira de ação Simples

### Mecanismo com Dois Pivôs

No extremo livre do pistão tem uma T que move os dedos para abrir - fechar.



Mecanismo com elementos pivôs.

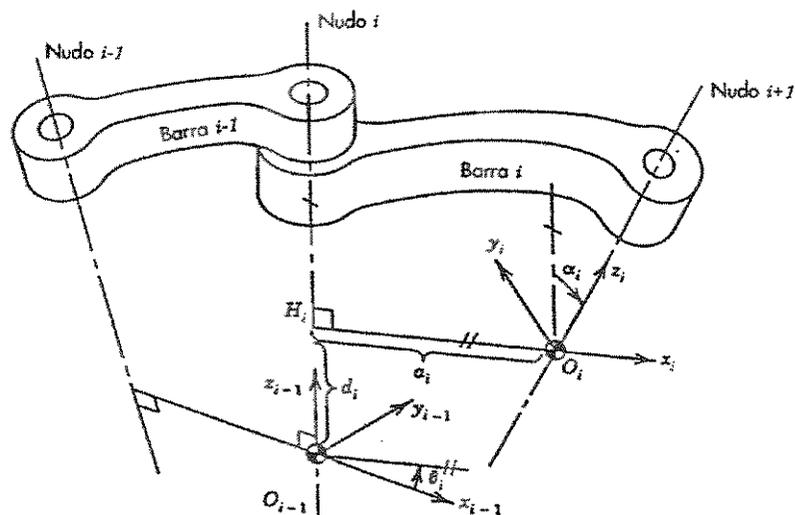


Garra virtual com três elementos.

## ANEXO 2

### METODOLOGIA DE DENAVIT - HARTENBERG

Com a metodologia Denavit–Hartenberg (1955) se definem os sistemas de coordenadas nas juntas de um robô, as variáveis da junta e os parâmetros estáticos da estrutura. É importante para a modelagem da cinemática e modelagem do Jacobiano do robô, aplicar corretamente esta metodologia.



Notação de Denavit e Hartenberg.

Considere-se a Figura 4.3 onde o eixo  $i$  identifica a conexão dos elementos  $i-1$  à  $i$ ; e se define o sistema de coordenadas do elemento  $i$ :

**Passo 1:** Escolher o eixo  $Z_i$  no eixo da junta  $i+1$ . Localizar o origem  $O_i$ , na interseção do eixo  $Z_i$  com a normal comum aos eixos  $Z_{i-1}$  e  $Z_i$ . Também se localiza  $H_i$  na interseção da normal comum com o eixo  $Z_{i-1}$ .

**Passo 2:** Escolher o eixo  $X_i$  na direção comum aos eixos  $Z_{i-1}$  e  $Z_i$  com orientação da junta  $i$  ao junta  $i+1$ .

**Passo 3:** Escolher o eixo  $Y_i$  que cumpra a regra da mão direita.

A metodologia Denavit - Hartenberg (1955) apresenta ambigüidade nos seguintes casos:

- 1) Para o elemento da base do robô o sistema  $O_0$  tem sô a direção do eixo  $Z_0$ , então,  $X_0$  e  $Y_0$  tem direção arbitrária.
- 2) Para a garra o ferramenta  $n$ , sô se escolhe o eixo  $X_n$  normal ao eixo  $Z_{n-1}$  e pode ter direção arbitrária.
- 3) Se dois eixos consecutivos são paralelos existem infinitas normais.
- 4) Se dois eixos consecutivos interceptam-se, a direção de  $X_i$  é arbitrária.
- 5) Se a junta  $i$  é prismática, sô a direção de  $Z_{i-1}$  está determinada.

Definidos os sistemas de coordenadas das juntas, do sistema  $i$  em relação ao sistema  $i-1$ , se determinam as variáveis e parâmetros da estrutura, assim:

**Passo 4:**  $a_i$ : Distância entre  $O_i$  e  $H_i$  ao longe de  $X_i$

**Passo 5:**  $d_i$ : Coordenada de  $H_i$  ao longe de  $Z_{i-1}$

**Passo 6:**  $\alpha_i$ : Ângulo entre os eixos  $Z_{i-1}$  e  $Z_i$  redor do eixo  $X_i$ , positivo à direita.

**Passo 7:**  $\theta_i$ : Ângulo entre os eixos  $X_{i-1}$  e  $X_i$  redor do eixo  $Z_{i-1}$ , positivo à direita.

Os parâmetros  $a_i$  y  $\alpha_i$  são constantes e dependem da geometria de conexão entre juntas consecutivas. Se a junta  $i$  é de revolução a variável é  $\theta_i$  e se a junta  $i$  é prismática a variável é  $d_i$ .

Para representar a MTH entre duas juntas de revolução consecutivas  $i$  e  $i-1$  se aplica a seguinte metodologia:

**Passo 1:** Escolher um sistema de coordenadas alinhado com  $i-1$ .

**Passo 2:** Transladar o sistema de coordenadas uma distancia  $d_i$  ao longo do eixo  $Z_{i-1}$  e girar um angulo  $\theta_i$  em torno do eixo  $Z_{i-1}$ ; e se escreve:

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

**Passo 3:** Transladar o sistema de coordenadas alinhado com  $i'$  uma distancia  $a_i$  na direção do eixo  $X_i$  e girar um angulo  $\alpha_i$  em torno do eixo  $X_i$ ; e se escreve:

$$\mathbf{A}_i^i = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

**Passo 4:** A coordenada de transformação resultante obtém-se pela pós-multiplicação das

transformações simples:

$$\mathbf{A}_i^{i-1}(q_i) = \mathbf{A}_i^{i-1} \cdot \mathbf{A}_i^i \quad (4.7)$$

$$\mathbf{A}_i^{i-1}(q_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

Para representar a MTH entre duas juntas prismáticas consecutivas  $i$  e  $i-1$  se aplica a mesma metodologia, e obtém-se:

$$\mathbf{A}_i^{i-1}(q_i) = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\alpha_i s\theta_i & 0 \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

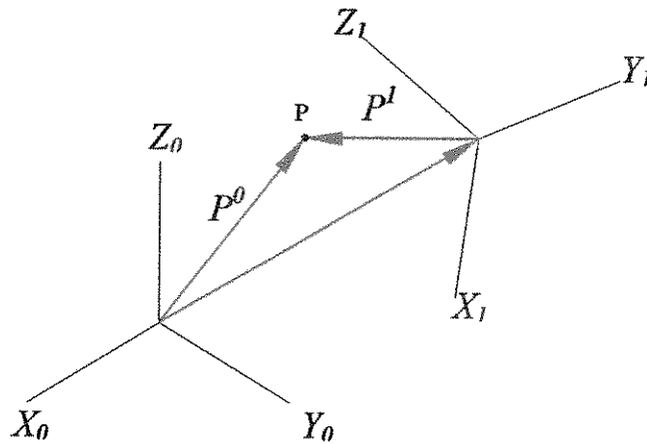
### ANEXO 3.

## MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEAS MTH

Denavit e Hartenberg (1955) introduziram o conceito de coordenadas homogêneas à análise dos mecanismos. Um ponto P de um corpo rígido no espaço pode ser representada fisicamente por uma matriz  $\mathfrak{R}^3$  e em coordenadas homogêneas por uma matriz  $\mathfrak{R}^4$  ou MTH, que representa a posição e orientação da garra ou ferramenta do robô. Com quatro partes: (1) Matriz de rotação  $\mathbf{R}_{3 \times 3}$  (2) Vetor de translação  $\mathbf{p}_{3 \times 1}$  (3) Transformação em perspectiva  $\mathbf{O}_{1 \times 3}$  (4) escalamento  $\mathbf{E}_{1 \times 1}$

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{O}_{1 \times 3} & \mathbf{E}_{1 \times 1} \end{bmatrix}$$

A Figura ilustra um sistema de coordenadas arbitrárias para representar a idéia de MTH de um ponto P no espaço. Seja o vetor  $\mathbf{p}^0$  cujo sistema de coordenadas é  $O_0(X_0, Y_0, Z_0)$  e o vetor  $\mathbf{p}^1$  cujo sistema de coordenadas é  $O_1(X_1, Y_1, Z_1)$ . Seja o vetor  $\mathbf{r}_1^0$  que descreve a posição do referencial  $O_1(X_1, Y_1, Z_1)$  com respeito ao referencial  $O_0(X_0, Y_0, Z_0)$  e seja também  $\mathbf{R}_1^0$  a matriz de rotação do referencial 1 respeito ao referencial 0.



Representação de um ponto com respeito a dois sistemas de coordenadas.

Com uma análise geométrica, tem-se:

$$\mathbf{p}^0 = \mathbf{r}_1^0 + \mathbf{R}_1^0 \mathbf{p}^1$$

$$\mathbf{p}^0 = \mathbf{R}_1^0 \mathbf{p}^1 + \mathbf{r}_1^0$$

A posição e orientação do ponto P são representadas com a seguinte MTH

$$\mathbf{A}_1^0 = \begin{bmatrix} \mathbf{R}_1^0 & \mathbf{r}_1^0 \\ \mathbf{O}^T & \mathbf{1} \end{bmatrix}$$

A transformação inversa obtém-se pré - multiplicando os dois lados da anterior expressão por  $\mathbf{R}_1^{0T}$  porque  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ , então:

$$\mathbf{R}_1^{0T} \mathbf{p}^0 = \mathbf{R}_1^{0T} \mathbf{r}_1^0 + \mathbf{R}_1^{0T} \mathbf{R}_1^0 \mathbf{p}^1$$

$$\mathbf{I}\mathbf{p}^1 = -\mathbf{R}_1^{0T} \mathbf{r}_1^0 + \mathbf{R}_1^{0T} \mathbf{p}^0$$

$$\mathbf{p}^1 = -\mathbf{R}_0^1 \mathbf{r}_1^0 + \mathbf{R}_0^1 \mathbf{p}^0$$

## ANEXO 4

### SIMULAÇÃO UTILIZANDO O APLICATIVO VISUAL BASIC™ VBA

Visual Basic™ Application VBA é um modulo existente no software Workspace™. As trajetórias desenvolvidas com VBA são uma metodologia alternativa de simulação; criam-se PGs, trajetórias com PGs, macros com PGs e gera-se uma codificação para aceder ao linguagem do robô em Workspace. Entre as duas metodologias existem diferencias: (1) As trajetórias em VBA escrevem-se no Integrated Development Environment IDE. (2) As trajetórias em VBA não estão conectadas necessariamente à um robô e podem ser usadas quando o robô não esta presente na FMC. (3) O Workspace Robot Language WRL pode ser usado em comandos do robô e com o modulo VBA. (4) WRL pode-se personalizar.

Integrated Development Environment IDE, é um editor VBA integrado a Workspace que orienta ao usuário a desenvolver e usar o modulo VBA e aceder à Workspace Object Model WOM. Também com WOM pode-se descrever VBA e pode-se chamar funções e dados dentro de Workspace. Worspace Robot Language WRL é a linguagem de programação *off line* de Workspace para simulação com VBA e pode ser personalizado.

Workspace usa uma linguagem do robô de alto nível em VBA para escrever a programação *off line* das trajetórias. Este linguagem oculta a complexidade do Realistic

Robot Simulation RRS. O usuário pode modificar esta linguagem modificando o módulo que tem a linguagem do robô. Workspace Robot Language WRL é o módulo que contém RRS e podem-se planificar as seguintes trajetórias:

- (1) Movimento no espaço da junta e adicionar um módulo.
- (2) Movimento linear e adicionar um módulo.
- (3) Movimento circular e adicionar um módulo.

## **Modelagem**

Cria-se uma FMC com um robô IRB 1400 e três mecanismos de rotação CAD associados como eixos auxiliares do robô. O primeiro mecanismo representa uma estação de subministro de peças prismáticas e peças cilíndricas, alternativamente, o segundo mecanismo representa uma estação de processamento de peças prismáticas e o terceiro mecanismo representa uma estação de processamento de peças cilíndricas. Precisa-se planificar a trajetória do robô que pega uma peça prismática da primeira estação e a posiciona na segunda estação, volta por a mesma rota ao primeira estação e pega a peça cilíndrica e a posiciona na terceira estação, finalmente, volta por a mesma rota ao primeira estação para terminar seu ciclo de trabalho. O robô em sua movimentação da primeira estação à segunda estação, simultaneamente gira 180 graus os três mecanismos, e em sua movimentação da terceira estação à primeira estação, simultaneamente gira os três mecanismos 180 graus, para liberar as locações para as seguintes peças prismáticas e cilíndricas, respectivamente.

A modelagem da FMC inclui: O robô no Working Coordinate System WCS, cada um dos mecanismos de rotação constituídos por dois cilindros CAD, o cilindro CAD base está conectada ao sistema de coordenadas da base do robô. Os sistemas de coordenadas dos mecanismos com respeito ao WCS são W1 (1000, 0, 630), W2 (0, 1000, 630) y W3 (0, -1000, 630). Figura 1 mostra a modelagem da FMC e as propriedades dos mecanismos.

Com o VBA se troca o nome do projeto à AuxiliaryAxesModule e revela-se o código com as funções dos mecanismos criados, que se mostram a seguir :

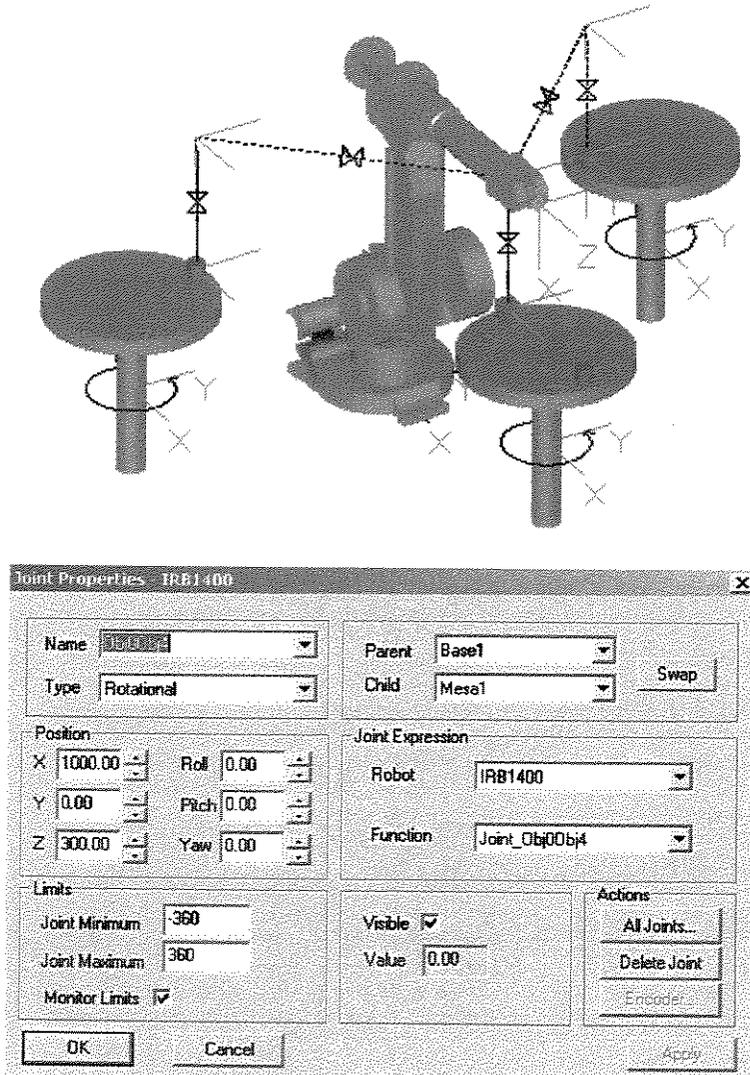


Figura 1. FMC com um robô e três mecanismos de rotação e propriedades dos mecanismos

```
Function Joint_Base3Mesa3(RRobot As Robot) As Double
```

```
    Joint_Base3Mesa3 = RRobot.AxisPos(7)
```

```
End Function
```

```
Function Joint_Base2Mesa2(RRobot As Robot) As Double
```

```
    Joint_Base2Mesa2 = RRobot.AxisPos(7)
```

```
End Function
```

```
Function Joint_Base1Mesa1(RRobot As Robot) As Double
```

```
    Joint_Base1Mesa1 = RRobot.AxisPos(7)
```

```

End Function
Function Joint_IRB1400LINK1(RRobot As Robot) As Double
    Joint_IRB1400LINK1 = RRobot.AxisPos(1)
End Function
Function Joint_LINK3PLINK5(RRobot As Robot) As Double
    Joint_LINK3PLINK5 = -RRobot.AxisPos(3)
End Function
Function Joint_LINK3LINK4(RRobot As Robot) As Double
    Joint_LINK3LINK4 = RRobot.AxisPos(4)
End Function
Function Joint_PLINK5PLINK1(RRobot As Robot) As Double
    Joint_PLINK5PLINK1 = RRobot.AxisPos(3)
End Function
Function Joint_LINK5LINK6(RRobot As Robot) As Double
    Joint_LINK5LINK6 = RRobot.AxisPos(6)
End Function
Function Joint_LINK4LINK5(RRobot As Robot) As Double
    Joint_LINK4LINK5 = RRobot.AxisPos(5)
End Function
Function Joint_LINK2LINK3(RRobot As Robot) As Double
    Joint_LINK2LINK3 = RRobot.AxisPos(3)
End Function
Function Joint_LINK1LINK2(RRobot As Robot) As Double
    Joint_LINK1LINK2 = RRobot.AxisPos(2)
End Function

```

No VBA troca-se as três funções Base1Mesa1, Base2Mesa2 e Base3Mesa3 controladas por as variáveis da junta (AxisPos)7, (AxisPos) 8 e (AxisPos)9.

## Trajetórias

Com Workspace criam-se PGs com respeito à WCS e para definir a trajetória R1 empleá-se os eixos X, Y, Z para o posicionamento com o comando Pendant como se mostra na Figura 2

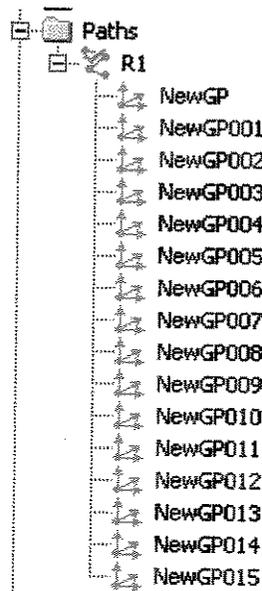


Figura 2. PG em trajetória com Workspace

Agora, com VBA importar WorkspaceRobotLanguage à Visual Basic Options como um novo módulo com o nome AuxiliaryAxesModule e dentro escrever o seguinte código para criar os pontas TP1 e TP2 da trajetória:

```
'Intended for use with the ABB_ird_1400
Option Explicit
'Handle for the robot
Public rRobot As Robot
Sub AuxiliaryAxesTrack()
Dim TP1 As WsTeachpoint
Dim TP2 As WsTeachpoint
Set rRobot = ThisDocument.GetRobot("IRB-1400")
```

```

SetCartesianTeachpoint rRobot, TP1, 955, 0, 1195, 0, 90, 0
LUNB , 10, 20, 30
SetJointAngleTeachpoint rRobot, TP2, 100, 0, 0, 0, 0, 0, 30, 20, 10
SetMotionType rRobot, Joint
MoveTo rRobot, TP1
MoveTo rRobot, TP2
End Sub

```

As instruções criaram uma trajetória para o robô que opera com os mecanismos de rotação.

```

SetCartesianTeachpoint rRobot, TP1, 955, 0, 1195, 0, 90, 0
LUNB , 10, 20, 30

```

Esta linha do código usa a função SetCartesianTeachpoint de Workspace Robot Language e põe posições cartesianas à variável TP1. Os valores 955, 0, 1195, 0, 90, 0 representam o origem do sistemas de coordenadas da garra (x, y, z, A, B, C) e 10, 20, 30 representam os valores para cada uma das juntas dos mecanismos (Auxiliary Axis 1, Auxiliary Axis 2 y Auxiliary Axis 3)

```

SetJointAngleTeachpoint rRobot, TP2, 100, 0, 0, 0, 0, 0, 30, 20, 10

```

Esta linha usa a função SetJointAngleTeachpoint e põe os ângulos de posição das juntas do robô IRB 1400 de ABB (rRobot) à variável TP2 que é uma variável WsTeachpoint. O nove valores reapresentam os ângulos de posição para cada junta do robô, as juntas 7, 8 e 9 representam os eixos dos mecanismos conectados na base do robô.

O modulo AuxiliaryAxesModule com o comando Add integrar-se ao VBA.

## **Simulação**

Com Workspace chama-se o comando Play Simulate e pode-se movimentar o robô e os mecanismos, simultaneamente, na trajetória descrita por os PG integrados na rota R1.

Com o modulo VBA se há criado uma trajetória com os pontos TP1 e TP2, e simultaneamente move-se cada um dos eixos dos mecanismos com angulo diferentes: 10, 20 e 30 grados. Para assegurar que o robô se move de acordo com a trajetória criada com VBA, e não com Workspace, desconecta-se a função Simulate do projeto onde se especifica o robô IRB1400. Seleciona-se o comando Play simulation para correr a simulação da programação em VBA.

O module VBA é uma ferramenta integrada ao Workspace para melhorar um projeto de modelagem, simulação e programação off line de FMC. Baseada em uma codificação específica que Workspace tem em sua base de dados. Com um conhecimento total do modulo VBA em um projeto melhora-se a criatividade, a flexibilidade, e a eficiência na utilização de Workspace

## **ANEXO V**

### **SIMULAÇÃO E**

### **PROGRAMAÇÃO COM APRENDIZAGEM VIRTUAL**

Considere-se uma FMC configurada com um robô IRB 1400 equipado com uma garra para posicionamento pick – and – place de uma peça CAD sobre uma mesa de trabalho com sistema de coordenadas  $O_M(X_M, Y_M, Z_M)$  como se mostra na Figura 1. A modelagem da garra do robô pode consultar na secção 3. 8. Para abrir ou fechar os dedos seleccionar a função Open e Close ao junta da garra e seleccionar a função Grasp ou Ungrasp, para agarrar ou soltar a peça CAD, respectivamente. Verifica-se que a superfície da mesa de trabalho está dentro da faixa de coordenadas cartesianas do robô, como se mostra na Figura 2 e a Tabela 1.

O robô realiza a função Pick – and - Place com trajetória Ponto – a - Ponto PTP no sistema de coordenadas da junta. Os PGs definam-se com a metodologia de aprendizagem com o comando Pendant, similarmente como si fora o Teaching Box com o robô real. Os seguintes são os valores para os PGs que correspondas aos juntas 1, 2, 3, 4, 5 y 6, respectivamente, originados das posições X, Y, Z com o Pendant e em Figura 3 como trajetória:

P0: Home

P1: 0, -15.06, 13.74, 0, -13.74, 900, 0, 1000

P2: -34.70, 0.21, 15.36, -69.07, -37.56, 64.24, 900, -500, 1000

P3: -34.70, 4.89, 31.41, -53.04, -45.44, 43, 900, -500, 800

P4: -34.70, 10.94, 42.14, -45.91, 52.44, 32.18, 900, -500, 670

P5: 34.70, 0.21, 15.36, -110.36, 37.56, 115.76, 900, 500, 1000

P6: 34.70, 4.89, 31.41, -126.96, 45.44, 137.00, 900, 500, 800

P7: 34.70, 10.94, 42.14, -134.09, 52.44, 147.82, 900, 500, 670

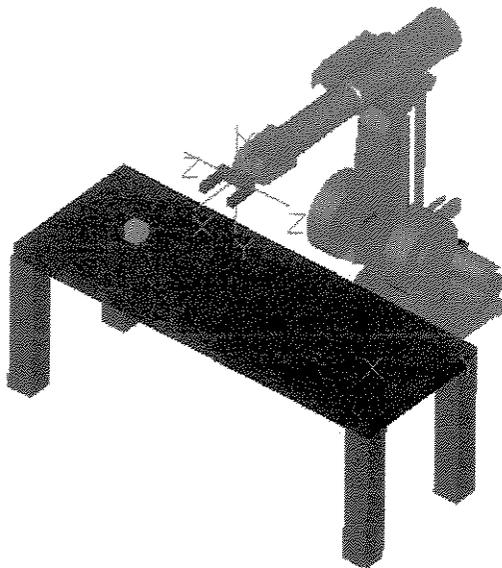


Figura 1. FMC com robô para posicionamento de peças

Tabela 1. Coordenadas cartesianas do robô IRB 1400 de ABB

| Coordenada | Positivo | Negativo |
|------------|----------|----------|
| X          | +900     | -900     |
| Y          | +1348    | -348     |
| Z          | +1495    | -5       |

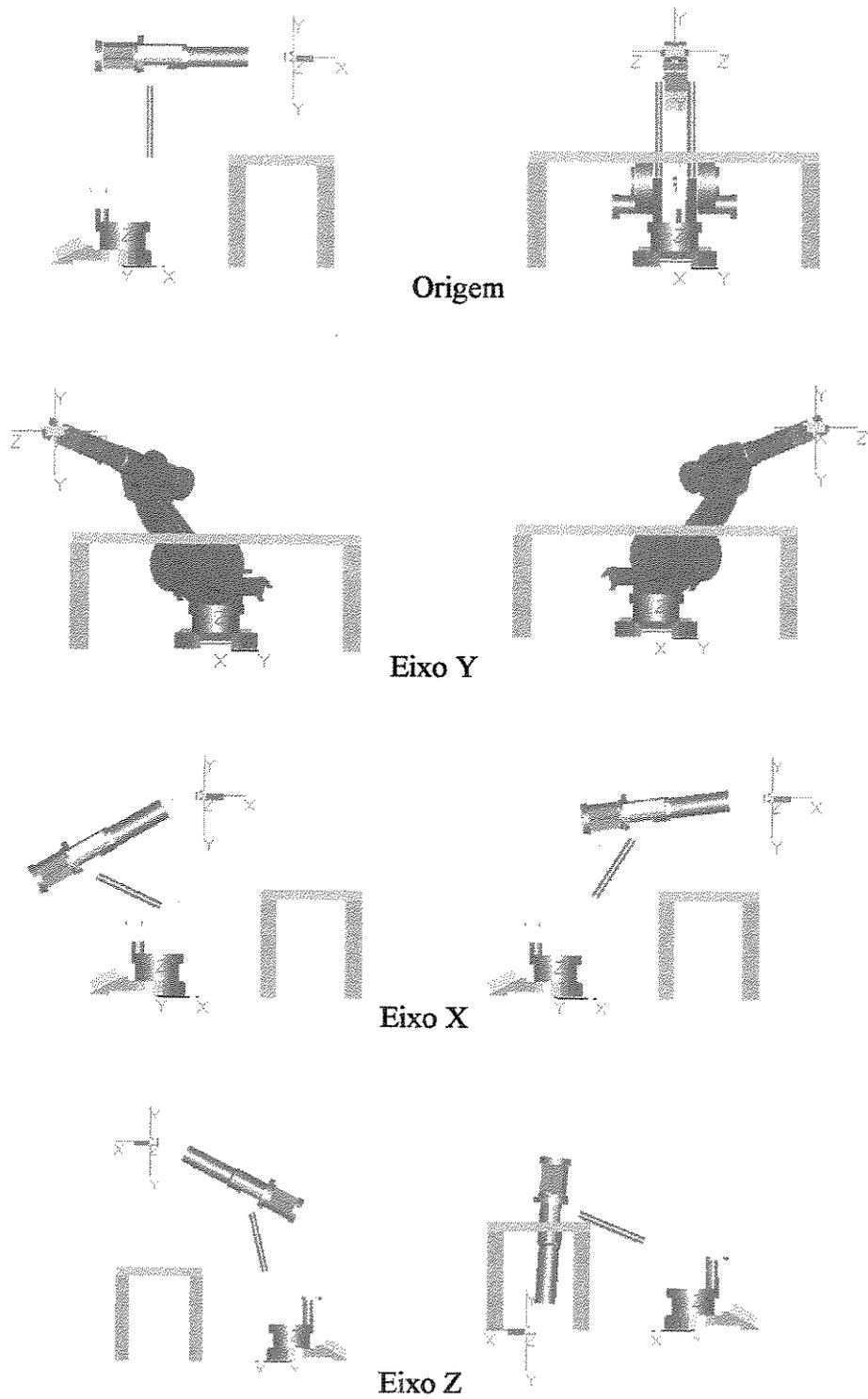


Figura 2. Coordenadas cartesianas do robô IRB 1400 de ABB.

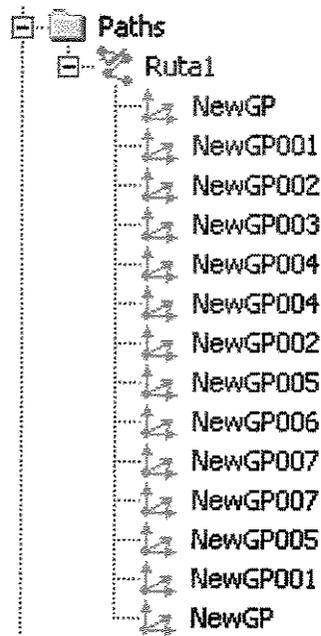


Figura 3. Rota de trajetória ponto – à - ponto

Com o modulo VBA desenvolve-se as melhoras para a garra : (1) Abrir e fechar o mecanismo da garra. (2) Apreensão e soltar. Do diretório de Workspace seleccionar o arquivo GripperBehavior.cls e do braço do projeto Worspace.IRB1400.Gripper utiliza-se a seguinte codificação :

Código para melhorar a garra ou gripper:

```
Private MyInterpolator As Interpolator
```

```
Private ToolName As String
```

```
Const OpenPosition = 0
```

```
Const ClosePosition = 45
```

```
Const OpenCloseTime = 500
```

```
Private GraspedObject As String
```

Código para melhorar o método de iniciar simulação:

```
Public Sub OnStartSimulation(Obj As String)
```

```
    ToolName = Obj
```

```
    Set MyInterpolator = New Interpolator
```

```
GraspedObject = ""  
End Sub
```

Código para melhorar finalizar simulação :

```
Public Sub OnEndSimulation()  
    Set MyInterpolator = Nothing  
End Sub
```

Con este código se reset el interpolador.

Código de interpolar intervalo:

```
Public Sub OnInterpolationInterval()  
    If MyInterpolator.Interpolating Then  
        Dim JointPos As New RCSJointPos  
        Dim Gripper As Robot  
        MyInterpolator.Interpolate  
ThisDocument.Simulation.InterpolationInterval  
        JointPos.AxesFormat = 1  
        JointPos.SetAxesValue 0, MyInterpolator.Value  
        Set Gripper = ThisDocument.GetRobot(ToolName)  
        Gripper.UpdateJoints JointPos  
    End If  
End Sub
```

Código de abrir a garra e código de fechar a garra:

```
Public Sub OpenGripper()  
    MyInterpolator.Start OpenCloseTime, ClosePosition, OpenPosition  
    DelayRobot  
End Sub  
Public Sub CloseGripper()  
    MyInterpolator.Start OpenCloseTime, OpenPosition, ClosePosition  
    DelayRobot
```

End Sub

Código de sujeitar a peça e código de soltar a peça:

```
Public Sub Grasp(GraspObj As String)
    ' Attaches part to gripper
    ThisDocument.Attach ToolName, GraspObj
    ' Store the name of the grasped object
    GraspedObject = GraspObj
End Sub
```

```
Public Sub UnGrasp()
    If Len(GraspedObject) Then
        ThisDocument.Detach ToolName, GraspedObject
        GraspedObject = ""
    End If
End Sub
```

Modulo de reset:

```
Sub ResetCADObject()
    ThisDocument.CadObject("Part").MoveAbsolute 900, -500, 670
End Sub
```

Em Workspace 5 as funções OpenGripper e CloseGripper estão trocadas; correspondem a fechar garra e abrir garra.