



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS DE CURITIBA
DEPARTAMENTO DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
E DE MATERIAIS - PPGEM

PAULO CESAR JUNEK FILHO

OTIMIZAÇÃO DE TRAJETÓRIAS DE ROBÔS DE
SOLDAGEM APLICADOS A CÉLULAS DE
MANUFATURA

CURITIBA
DEZEMBRO – 2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

PAULO CESAR JUNEK FILHO

**OTIMIZAÇÃO DE TRAJETÓRIAS DE ROBÔS DE
SOLDAGEM APLICADOS A CÉLULAS DE
MANUFATURA**

Dissertação apresentada como requisito parcial à obtenção do título de Mestre em Engenharia, do Programa de Pós-Graduação em Engenharia Mecânica e de Materiais, Área de Concentração em Engenharia de Manufatura, do Departamento de Pesquisa e Pós-Graduação, do Campus de Curitiba, da UTFPR.

Orientador: Prof. Dr. Luiz Carlos de Abreu Rodrigues

**CURITIBA
DEZEMBRO – 2008**

TERMO DE APROVAÇÃO

PAULO CESAR JUNEK FILHO

OTIMIZAÇÃO DE TRAJETÓRIAS DE ROBÔS DE SOLDAGEM APLICADOS A CÉLULAS DE MANUFATURA

Esta Dissertação foi julgada para a obtenção do título de mestre em engenharia, área de concentração em engenharia de Manufatura, e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia Mecânica e de Materiais.

Prof. Giuseppe Pintaúde, Dr.
Coordenador de Curso

Banca Examinadora

Profa. Maria Teresinha Arns Steiner, Dr.
(UFPR)

Prof. João Antônio Palma Setti, Dr.
(UTFPR)

Prof. Eduardo Mauro Nascimento, Dr.
(UTFPR)

Curitiba, 16 de dezembro de 2008

AGRADECIMENTOS

Agradeço a todos aqueles que contribuíram direta ou indiretamente no desenvolvimento deste trabalho, em especial:

Ao prof. Dr. Luiz Carlos de Abreu Rodrigues, orientador, pela contribuição direta nesta dissertação;

Ao prof. Luís Paulo Laus, pelo material sobre robótica disponibilizado e utilizado neste trabalho;

Aos meus pais, pelo apoio e auxílio durante o desenvolvimento deste trabalho.

À minha esposa e filha (Simone e Alice) pela compreensão e apoio durante os momentos em que estive ausente do convívio familiar para o desenvolvimento deste trabalho.

JUNEK FILHO, Paulo Cesar, **Otimização de Trajetórias de Robôs de Soldagem Aplicados a Células de Manufatura**, 2008, Dissertação (Mestrado em Engenharia) - Programa de Pós-graduação em Engenharia Mecânica e de Materiais, Universidade Tecnológica Federal do Paraná, Curitiba, 94p.

RESUMO

Neste trabalho é abordado o problema de planejamento de trajetória otimizada de um robô industrial. Dispõe-se de um conjunto composto por um robô de seis eixos e um posicionador (mesa de posicionamento) de um eixo, que são aplicados a um processo de soldagem do eixo traseiro de um automóvel. Esta configuração, segundo os fabricantes de robôs, acrescenta flexibilidade e precisão ao processo, em comparação à operação de soldagem com um dispositivo estático. Porém, esta flexibilidade pode ser questionada, do ponto de vista de programação do robô; já que o processo de posicionamento de ambos (robô e posicionador) normalmente é realizado através do processo conhecido como ponto-a-ponto, no qual o programador precisa conduzir o robô através de cada ponto da trajetória a ser executada, sendo que cada ponto deve ser gravado na memória do robô. Esse método apresenta a desvantagem de obrigar a interrupção do processo de produção durante a programação, além de tornar o sistema de programação de um robô de soldagem complexo e demorado, principalmente na execução de peças com vários cordões a serem executados. Em problemas de planejamento de trajetória, o número de trajetórias possíveis entre as posições inicial e final para um robô é frequentemente muito grande, ou melhor, infinito, e o objetivo não é necessariamente determinar a melhor solução, mas obter uma que seja aceitável de acordo com certos requisitos e certas restrições. Visando apresentar um método mais eficiente de programação para o robô em questão, é apresentada uma alternativa através do método da Busca Tabu para a seleção da melhor alternativa de trajetória para o robô durante o processo de soldagem, através da avaliação da melhor combinação de valores de ângulos de junta para a pose de trabalho.

Palavras-chave: Planejamento de trajetória, Programação de robôs, Busca Tabu

JUNEK FILHO, Paulo Cesar, **Otimização de Trajetórias de Robôs de Soldagem Aplicados a Células de Manufatura**, 2008, Dissertação (Mestrado em Engenharia) - Programa de Pós-graduação em Engenharia Mecânica e de Materiais, Universidade Tecnológica Federal do Paraná, Curitiba, 94p.

ABSTRACT

In this work the problem of an industrial robot's optimized path planning is approached. It is composed by a robot of six axes and an indexing (positioning) table of an axis, that are applied to the welding process of the back axis of an automobile. This configuration increases flexibility and precision to the process, in comparison with the welding operation with a static device. However, the process of positioning of both robot and indexing table is usually accomplished through the process known as point-to-point programming. In point-to-point programming the programmer needs to lead the robot through each point of the path to be executed, and each point should be recorded in the robot's memory. This method presents the disadvantage of forcing the interruption of the production process during the programming, besides turning the programming of a welding robot system into a complex and slow process,. In problems of path planning, the number of possible paths among the initial and final positions for a robot is frequently big - or better - infinite. Due to this, the objective is not necessarily to determine the best solution, but to obtain one that is acceptable in agreement with certain requirements and restrictions. Seeking to present a more efficient method of programming for the robot in subject, an alternative approach is presented, through the use of Tabu Search method for the selection of the best alternative path for the welding process. This is done through the evaluation of the best combination of values of joint angles which form a working pose.

Keywords: Path Planning, Robot Programming, Tabu Search

SUMÁRIO

RESUMO.....	ii
ABSTRACT.....	iii
LISTA DE FIGURAS.....	vi
LISTA DE TABELAS.....	vii
LISTA DE ABREVIATURAS E SIGLAS.....	ix
1 INTRODUÇÃO.....	1
1.1 Objetivos.....	1
1.2 Justificativa.....	1
1.3 Estrutura da dissertação.....	2
2 DESCRIÇÃO DO PROBLEMA.....	3
3 MODELO CINEMÁTICO DE ROBÔS.....	6
3.1 Cinemática direta.....	7
3.1.1 Modelo cinemático do robô estudado.....	8
3.2 Cinemática inversa.....	12
3.3 Programação de robôs.....	13
3.4 Planejamento da trajetória de robôs.....	17
3.5 Programação de robôs de soldagem.....	19
4 TÉCNICAS DE OTIMIZAÇÃO	22
4.1 Busca tabu	23
4.2 Algoritmos genéticos	32
4.2.1 Cruzamento.....	33
4.2.2 Mutação.....	33
4.2.3 Reprodução.....	34
4.2.4 Um exemplo simples.....	35
4.3 <i>Simulated annealing</i>	36
5 METODOLOGIA	40
5.1 Trajetória linear.....	44
5.2 Trajetória circular.....	46
5.3 Geração de trajetória usando algoritmo do <i>simulated annealing</i>	51
5.4 Geração de trajetória usando o algoritmo da busca tabu.....	52
5.4.1 Busca na vizinhança.....	54
5.5 Análise da variância.....	54
5.5.1 Tratamento.....	55
5.5.2 Unidade experimental ou parcela.....	55
5.5.3 Repetição.....	56
5.5.4 Experimentos fatoriais.....	58

6	RESULTADOS.....	62
6.1	Análise dos resultados para o cordão linear.....	62
6.2	Análise dos resultados para o cordão circular.....	70
7	CONCLUSÕES.....	79
	REFERÊNCIAS.....	81

LISTA DE FIGURAS

Figura 2.1 - Eixo traseiro de um automóvel	3
Figura 2.2 - Exemplo de posicionador de um eixo (um grau de liberdade).....	4
Figura 2.3 - Exemplo de robô de seis eixos (seis graus de liberdade) para solda.....	4
Figura 2.4 - Detalhe da solda do eixo traseiro de automóvel.....	5
Figura 3.1 - Modelo cinemático robô de seis graus de liberdade.....	8
Figura 3.2 - Parâmetros usados na notação Denavit-Hartenberg.....	9
Figura 3.3 - Oito soluções para um robô de seis graus de liberdade.....	14
Figura 3.4 - Controle para programação de robôs ABB Flex-pendant.....	16
Figura 3.5 - Célula de soldagem robotizada.....	20
Figura 4.1 - Algoritmo da Busca Tabu.....	25
Figura 5.1 - Representação do cordão linear.....	44
Figura 5.2 - Valores adotados no presente estudo.....	45
Figura 5.3 - Diagrama ponto 1 da trajetória circular.....	46
Figura 5.4 - Diagrama ponto 2 da trajetória circular.....	47
Figura 5.5 - Diagrama ponto 3 da trajetória circular.....	48
Figura 5.6 - Diagrama ponto 4 da trajetória circular.....	49
Figura 5.7 - Diagrama ponto 5 da trajetória circular.....	50
Figura 5.8 - Algoritmo de <i>simulated annealing</i> (SA) usado neste trabalho.....	51
Figura 5.9 - Algoritmo de busca tabu adotado neste trabalho.....	53
Figura 5.10. Procedimento do Planejamento Sistemático de Layout (SLP).....	55

LISTA DE TABELAS

Tabela 3.1 – Relação cinemática entre as juntas.....	10
Tabela 3.2 - Componentes básicos de um sistema de soldagem robotizado.....	21
Tabela 4.1 - Conteúdo da memória após uma iteração.....	27
Tabela 4.2 - Conteúdo da memória após cinco iterações.....	28
Tabela 4.3 - Conteúdo da memória após seis iterações.....	29
Tabela 4.4 - Conteúdo da memória baseada em frequência para $h = 50$	30
Tabela 4.5 - Cinco seqüências aleatórias.....	35
Tabela 4.6 - Um típico experimento.....	35
Tabela 4.7 - Geração de novas soluções partindo de $T=100$	37
Tabela 4.8 - 2ª tentativa de geração de soluções partindo de $T=500$	38
Tabela 5.1 - Disposição das equações na Tabela ANOVA.....	58
Tabela 5.2 - Plano experimental para algoritmo SA.....	58
Tabela 5.3 - Tabela resumo da análise de fatores.....	61
Tabela 6.1 - Tabela ANOVA da função objetivo nos experimentos de SA para o cordão linear.....	63
Tabela 6.2 - Apresentação do valor da função objetivo nos 8 experimentos realizados usando <i>simulated annealing</i> para o cordão linear.....	64
Tabela 6.3 - Apresentação do número de iterações nos 8 experimentos realizados usando <i>simulated annealing</i> para o cordão linear.....	65
Tabela 6.4 - Tabela ANOVA do número de iterações nos experimentos de SA para o cordão linear.....	66
Tabela 6.5 - Valor da função objetivo nos experimentos usando BT para o cordão linear....	67
Tabela 6.6 - Tabela ANOVA da função objetivo nos experimentos de BT para o cordão linear.....	67
Tabela 6.7 - Número de iterações nos experimentos usando BT para o cordão linear.....	68
Tabela 6.8 - Tabela ANOVA do número de iterações nos experimentos de BT para o cordão linear.....	68

Tabela 6.9 - Comparação entre as abordagens SA e BT.....	69
Tabela 6.10 - Tabela ANOVA comparando as abordagens SA e BT.....	69
Tabela 6.11 - Tempo de execução dos algoritmos para o caso de cordão linear.....	70
Tabela 6.12 - Tabela ANOVA da função objetivo nos experimentos de SA para o cordão circular.....	71
Tabela 6.13 - Apresentação do valor da função objetivo nos 8 experimentos realizados usando <i>simulated annealing</i> para o cordão circular.....	72
Tabela 6.14 - Apresentação do número de iterações nos 8 experimentos realizados usando <i>simulated annealing</i> para o cordão circular.....	73
Tabela 6.15 - Tabela ANOVA do número de iterações nos experimentos de SA para o cordão circular.....	74
Tabela 6.16 - Valor da função objetivo nos experimentos usando BT para o cordão circular.....	75
Tabela 6.17 - Tabela ANOVA da função objetivo nos experimentos de BT para o cordão circular.....	75
Tabela 6.18 - Número de iterações nos experimentos usando BT para o cordão circular.....	76
Tabela 6.19 - Tabela ANOVA do número de iterações nos experimentos de BT para o cordão circular.....	76
Tabela 6.20 - Comparação entre as abordagens SA e BT.....	77
Tabela 6.21 - Tabela ANOVA comparando as abordagens SA e BT.....	77
Tabela 6.22 - Tabela ANOVA do número de iterações nos experimentos de BT para o cordão circular.....	78

LISTA DE ABREVIATURAS E SIGLAS

AG	<i>Genetic Algorithms</i> (Algoritmos Genéticos)
BT	<i>Tabu Search</i> (Busca Tabu)
DOF	<i>Degree of Freedom</i> (graus de liberdade)
MIG/MAG	<i>Metal Inerte Gás/ Metal Active Gás</i> (processo de soldagem com gás inerte/ gás ativo)
SA	<i>Simulated Annealing</i> (têmpera simulada)
SAT	<i>Satisfiability</i> (problema de satisfação de restrições)
TCP	<i>Tool Center Point</i> (ponto central da ferramenta)

1. INTRODUÇÃO

O uso de robôs em aplicações industriais tem aumentado nos últimos anos. Os robôs deixaram de ser privilégio das grandes indústrias e têm se disseminado em diferentes ramos e aplicações industriais. Porém, em muitos casos, o potencial de um robô aplicado a um processo não é completamente utilizado, devido à sua programação não permitir que o seu desempenho seja otimizado. Esse fato tem motivado diversos trabalhos científicos voltados para a otimização de processos industriais robotizados, com a aplicação de técnicas computacionais matemáticas e heurísticas (WU *et al*, 2000; ÇAKIR *et al*, 2006), algumas das quais serão abordadas neste trabalho. Estas últimas, principalmente, têm sido empregadas com grande intensidade em muitos problemas que envolvem alocação de recursos, embalagem e programação da produção, pois não necessitam de estruturas complexas de modelagem como os métodos baseados em programação matemática, por exemplo. Além disso, a adoção de métodos heurísticos é vantajosa em problemas que têm como características: dificuldade de modelagem e tempo de solução inviável do ponto de vista da capacidade de processamento da solução (RODRIGUES, 2006).

1.1 Objetivos

Neste trabalho buscou-se, através da aplicação do método da Busca Tabu, a obtenção da melhor alternativa de trajetória para o conjunto formado por um robô industrial de seis eixos e por uma “mesa posicionadora” de um eixo, sendo aplicados a um processo de soldagem do eixo traseiro de um automóvel. Para este problema de planejamento de trajetória, através da avaliação da melhor combinação de valores de ângulos de junta para a pose de trabalho do robô, identificou-se a trajetória otimizada para a execução dos cordões de solda necessários à união das diversas partes que, reunidas, formam o eixo em questão.

1.2 Justificativa

Neste tipo de problema, o processo de posicionamento de ambos (robô e posicionador) normalmente é realizado através do processo conhecido como ponto-a-ponto ou *leadthrough*, no qual o programador precisa conduzir o robô através de cada ponto da trajetória a ser executada, sendo que cada ponto deve ser gravado na memória do robô. Esse método apresenta a desvantagem de obrigar a interrupção do processo de produção durante a programação; além de tornar o sistema de programação de um robô de soldagem

complexo e demorado, principalmente na execução de peças com vários cordões a serem executados. Além disso, no processo ponto-a-ponto, cabe ao programador do robô a definição da posição e da orientação da tocha de soldagem e, também, da orientação do posicionador. Assim, uma definição inadequada da posição e das orientações implicará num maior tempo de processamento da operação de solda.

Visando apresentar um método mais eficiente de programação para o robô em questão, a proposta do trabalho é a aplicação do método da Busca Tabu para a seleção da melhor alternativa de trajetória para o robô durante o processo de soldagem.

1.3 Estrutura da Dissertação

Neste capítulo 1 são apresentados os objetivos e a justificativa do trabalho, além da indicação das técnicas selecionadas para a sua obtenção. No capítulo seguinte, descreve-se em detalhes o problema de planejamento de trajetória de robôs abordado. No capítulo 3, apresenta-se a teoria sobre cinemática direta e inversa de robôs, bem como os métodos de programação e técnicas relacionadas à aplicação de robôs em um processo de soldagem. No capítulo seguinte, são apresentados os principais métodos heurísticos de otimização (busca tabu, algoritmos genéticos e *simulated annealing*) e alguns exemplos de aplicação. No capítulo 5, descreve-se a metodologia proposta para a abordagem do problema de planejamento de trajetória, onde são apresentadas as equações utilizadas no método, buscando a obtenção da solução ótima. Finalmente, são apresentados os resultados no capítulo 6 e as conclusões obtidas com o trabalho no capítulo 7.

2. DESCRIÇÃO DO PROBLEMA

No presente trabalho é abordado o problema de planejamento otimizado de trajetória de um robô industrial. Dispõe-se de um conjunto composto por um robô de seis eixos e um posicionador (mesa de posicionamento) de um eixo, que são aplicados a um processo de soldagem do eixo traseiro de um automóvel, como o mostrado na figura 2.1.



Figura 2.1 - Eixo traseiro de automóvel.

O posicionador de um eixo de um grau de liberdade (figura 2.2) é usado para ajustar a orientação necessária e mais favorável à execução de cordões de solda através do processo MIG/MAG (*Metal Inerte Gás/Metal Active Gas*), enquanto que o robô de seis graus de liberdade (figura 2.3) é responsável por guiar a tocha de soldagem durante o processo. Esta configuração acrescenta flexibilidade e precisão ao processo, em comparação à operação de soldagem com um dispositivo estático. Porém, o processo de posicionamento de ambos (robô e posicionador) normalmente é realizado por meio do processo conhecido como ponto-a-ponto, no qual o programador precisa conduzir o robô através de cada ponto da trajetória a ser executada, gravando cada ponto na memória do robô. Esse método apresenta a desvantagem de obrigar a interrupção do processo de produção durante a programação, além de tornar o sistema complexo e demorado, principalmente na execução de peças com vários cordões.

Quando é necessária a elaboração de um planejamento de trajetória para robôs de múltiplos eixos (juntas), o perfeito sincronismo entre os mesmos é um fator determinante para que o caminho do Ponto Central da Ferramenta (*TCP – Tool Center Point*) não seja modificado em relação ao trajeto previamente programado. Para uma tarefa de movimentação de materiais, isto não representa um problema, pois, neste caso, busca-se, principalmente, que os pontos inicial e final da trajetória sejam atendidos, independente do

caminho que foi seguido. Porém, para aplicações como soldagem, pintura e montagem, por exemplo, é imprescindível assegurar que o robô siga a trajetória programada.



Figura 2.2. Exemplo de posicionador de um eixo (um grau de liberdade) (ABB, 2008).



Figura 2.3. Exemplo de robô de seis eixos (seis graus de liberdade) para solda (ABB, 2008).

Em problemas de planejamento de trajetória, o número de trajetórias possíveis entre as posições inicial e final para um robô é frequentemente muito grande, ou melhor, infinito, como será apresentado a seguir, e o objetivo não é necessariamente determinar a melhor solução, mas obter uma que seja aceitável de acordo com certos requisitos e certas

restrições. Vários métodos de busca têm sido desenvolvidos (por exemplo, métodos baseados em cálculos, algoritmos de busca aleatória, etc.) para o problema de planejamento de trajetória de robôs (GARG e KUMAR, 2002).

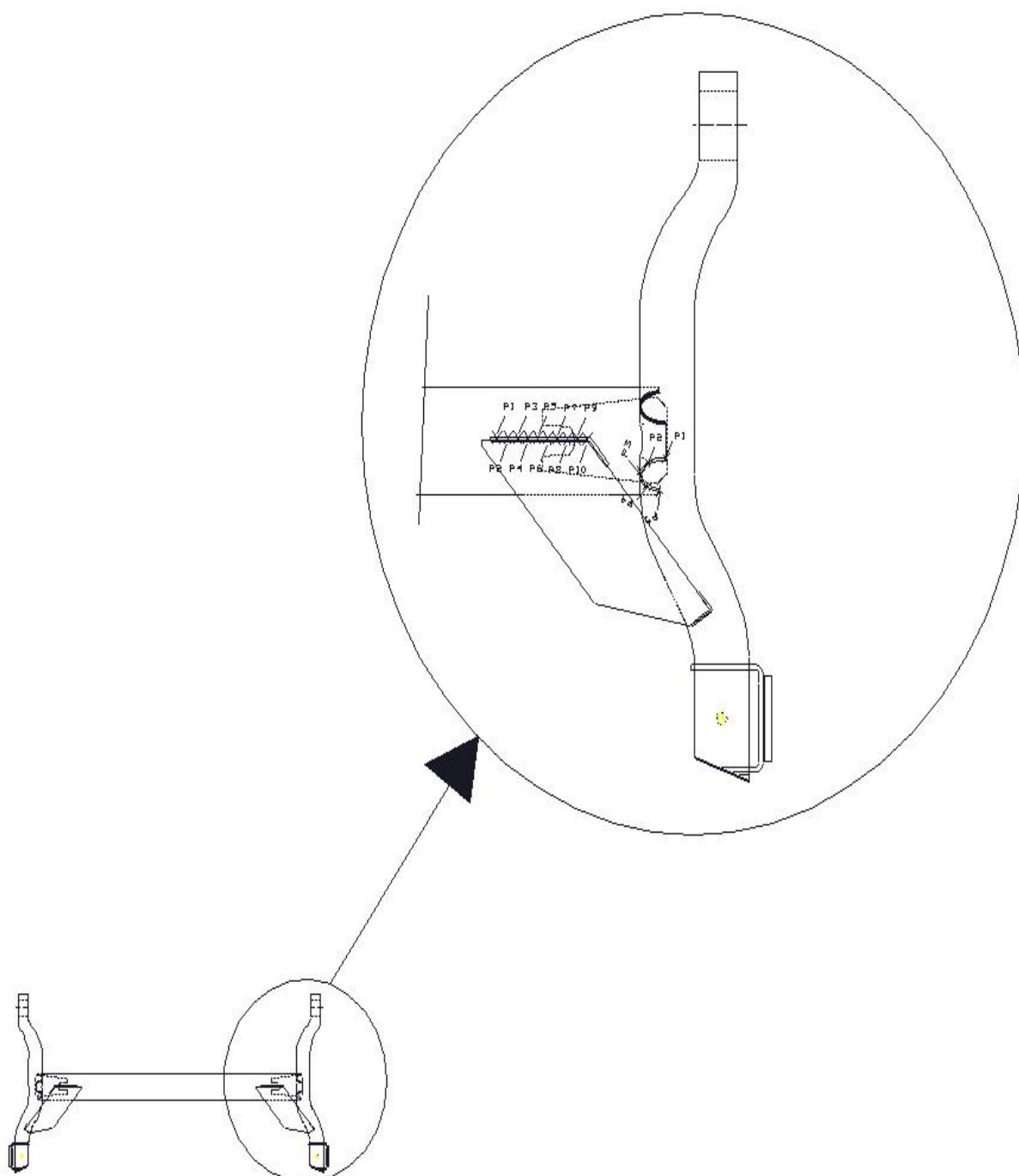


Figura 2.4 – Detalhe da solda do eixo traseiro de automóvel.

3. MODELO CINEMÁTICO DE ROBÔS

Um robô pode ser imaginado como sendo uma cadeia de corpos rígidos chamados de segmentos (*links*) unidos entre si por ligações móveis chamadas juntas (*joints*). Quando se tem dois corpos rígidos que podem se mover um em relação ao outro com a restrição de que o movimento se dê sobre uma superfície de contato, este conjunto de corpos recebe o nome de par inferior. Existem apenas seis tipos de pares inferiores: rotativo, prismático, cilíndrico, planar, fuso e esférico. Os pares inferiores do tipo rotativo e prismático permitem apenas um único grau de liberdade, já os demais permitem dois (cilíndrico e fuso) ou três (planar e esférico). Alguns autores consideram que o fuso também só possui um único grau de liberdade porque a rotação e a translação são restritas: um movimento de translação sempre implicará numa rotação e vice-versa. (ROMANO, 2002)

Em geral, cada junta é acionada por um atuador (um motor elétrico ou cilindro hidráulico) e, neste caso, só possui um único grau de liberdade. Por isso, as juntas da maioria dos robôs são rotativas (movimento angular) ou prismáticas (movimento linear). Eventualmente, são utilizadas juntas esféricas, também chamadas de joelhos, com o objetivo de manter a distância entre dois pontos constante.

Embora a cinemática seja o estudo do movimento sem levar em conta sua causa, em robótica o termo é usado para descrever as relações entre o movimento das juntas do robô e o movimento dos diversos segmentos que o compõem, sobretudo do último segmento livre que transporta a ferramenta (punho). Então, um conjunto chamado de variáveis de junta é definido, as quais expressam a posição de uma junta, seja rotativa ou prismática. O papel da cinemática é criar um mapeamento entre as variáveis de junta e a posição e orientação de cada segmento do robô. Surpreendentemente, este mapeamento é estático: posição para posição, e não há o envolvimento de velocidades (cinemática e dinâmica). Geralmente, só há interesse no último segmento, que é aquele que transporta a ferramenta de trabalho.

Têm-se, então, dois tipos de cinemática:

- Cinemática direta, a qual define a posição e orientação do punho em função das variáveis de junta;
- Cinemática inversa, responsável por determinar o valor das variáveis de junta para se atingir, com o punho, uma determinada posição e orientação.

Os dois tipos de junta (rotativa e prismática) de interesse são caracterizados por uma reta. No caso da junta rotativa a reta é o eixo de rotação. Para a junta prismática, a reta

indica a direção e sentido (se for considerado uma reta orientada) sobre os quais se dá o movimento (ROMANO, 2002). Neste capítulo é feita uma revisão sucinta de cinemática direta e inversa. O leitor interessado encontrará uma explicação detalhada de cinemática direta e inversa em CRAIG (1989) e em LAUS (2006).

3.1 Cinemática direta

Como dito anteriormente, o objetivo da cinemática direta de robôs é obter uma descrição da posição e orientação da ferramenta de trabalho em função dos ângulos (no caso de juntas rotativas) ou comprimento (no caso das juntas prismáticas) de junta do robô que são chamados de variáveis de junta. Esta descrição pode ser feita através de sistemas de coordenadas. Um local conveniente para o sistema de coordenadas situa-se no início ou no fim do segmento sobre uma das juntas, fazendo com que o eixo da junta coincida com um dos três eixos do sistema de coordenadas. Na robótica convencionou-se associar o eixo de junta ao versor \hat{z} e, neste trabalho, adotou-se fixar cada sistema de coordenadas no início do segmento sobre cada junta, no que é chamada *Notação Hartenberg Modificada*, como proposto por CRAIG (1989).

Na cinemática direta, para conhecer-se a posição/orientação da ferramenta de solda, deve-se definir o valor das variáveis de junta. Para se obter uma descrição do movimento do robô (posição e orientação variando no tempo) é preciso saber descrever a posição e orientação dos seus segmentos. Para isso, atrela-se um sistema de coordenadas a cada segmento. Assim, o problema da descrição da posição e orientação relativa dos segmentos do robô se transforma em um problema de descrição da posição e orientação relativa dos sistemas de coordenadas, que é solucionado com a construção de matrizes chamadas *transformações homogêneas*, que contém a informação da posição e orientação relativa entre dois sistemas de coordenadas. Uma transformação homogênea (${}^A_B T$) é uma matriz 4x4, conforme indicado na equação 3.1 que contém uma matriz de rotação 3x3 (${}^A_B R_{3x3}$), que definirá a orientação relativa entre dois sistemas de coordenadas, indicados a seguir como {A} e {B}, e um vetor-coluna 3x1 (${}^A O_B$) que identifica a origem de um sistema de coordenadas {B} em relação ao sistema de coordenadas {A}. A última linha da transformação homogênea será sempre igual a [0 0 0 1]. Na matriz de rotação (${}^A_B R_{3x3}$), a primeira coluna apresenta a projeção do versor \hat{x}_B em relação ao sistema de coordenadas {A}. Da mesma forma, a segunda e terceira colunas apresentam, respectivamente, as

projeções dos versores \hat{y}_B e \hat{z}_B em relação ao sistema de coordenadas {A} (CRAIG, 1989; LAUS, 2006).

$${}^A_B T = \begin{bmatrix} {}^A_B R_{3 \times 3} & {}^A O_{B3 \times 1} \\ 0 & 1 \end{bmatrix} \quad \text{Eq. 3.1}$$

Na construção do modelo cinemático é necessário partir-se de uma “pose” ou desenho do robô, com a representação dos tipos de junta e a relação entre elas. Perspectivas cônicas sofisticadas e com riqueza de detalhes geralmente não são úteis porque tornam o trabalho de interpretação do desenho mais difícil, já que escondem a relação entre as juntas e, muitas vezes, a própria localização de uma junta. Uma vez que se compreende a relação entre as juntas, formulando um modelo mental do robô, é possível construir-se um desenho utilizando-se uma representação simbólica das juntas, ao invés de uma perspectiva ou de uma projeção ortogonal mais acurada (LAUS *et al.*, 2006).

3.1.1 Modelo cinemático do robô estudado

O modelo cinemático de um robô de seis graus de liberdade, que foi usado neste trabalho, está representado de forma simbólica na figura 3.1, juntamente com a tabela 3.1 que representa a relação cinemática entre as juntas. No modelo, as letras \hat{x} e \hat{z} , juntamente com os seus índices, representam os versores dos eixos do sistema de coordenadas associado a cada junta do robô.

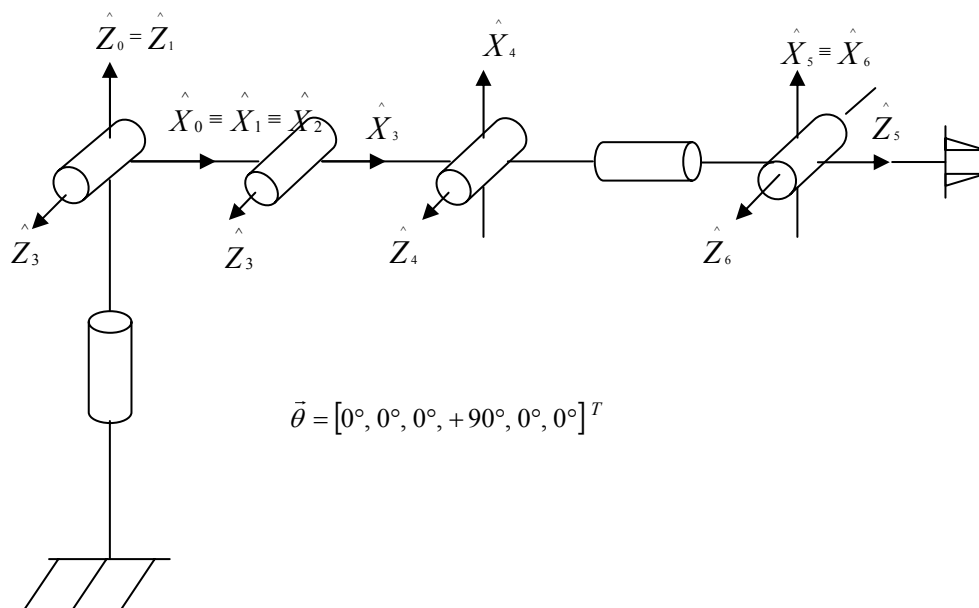


Figura 3.1 - Modelo cinemático robô de seis graus de liberdade (LAUS, 2006).

Note que o modelo da figura 3.1 também define a pose de referência que corresponde aos ângulos de junta $\vec{\theta} = [0^\circ, 0^\circ, 0^\circ, +90^\circ, 0^\circ, 0^\circ]^T$. Estes valores para os ângulos de junta, assim como os demais valores dos parâmetros do robô são indicados na tabela 3.1. Nesta tabela:

- a_{i-1} é a distância do versor \hat{z}_{i-1} até o versor \hat{z}_i ao longo do versor \hat{x}_{i-1} ;
- α_{i-1} é o ângulo do versor \hat{z}_{i-1} até o versor \hat{z}_i medido em relação ao versor \hat{x}_{i-1} ;
- d_i é a distância do versor \hat{x}_{i-1} até o versor \hat{x}_i ao longo do versor \hat{z}_i ;
- θ_i é o ângulo do versor \hat{x}_{i-1} até o versor \hat{x}_i medido em relação ao versor \hat{z}_i .

A figura 3.2 ilustra os parâmetros, apresentados na tabela 3.1, que são normalmente usados na representação cinemática de robôs industriais. Como mencionado anteriormente, a *Notação de Denavit-Hartenberg Modificada* é usada para gerar a representação da posição e orientação relativa entre dois sistemas de coordenadas adjacentes $\{i-1\}$ e $\{i\}$. Para isso, basta calcular a transformação homogênea indicada na equação 3.2, através da substituição dos valores de a_{i-1} , α_{i-1} , d_i e θ_i nesta equação.

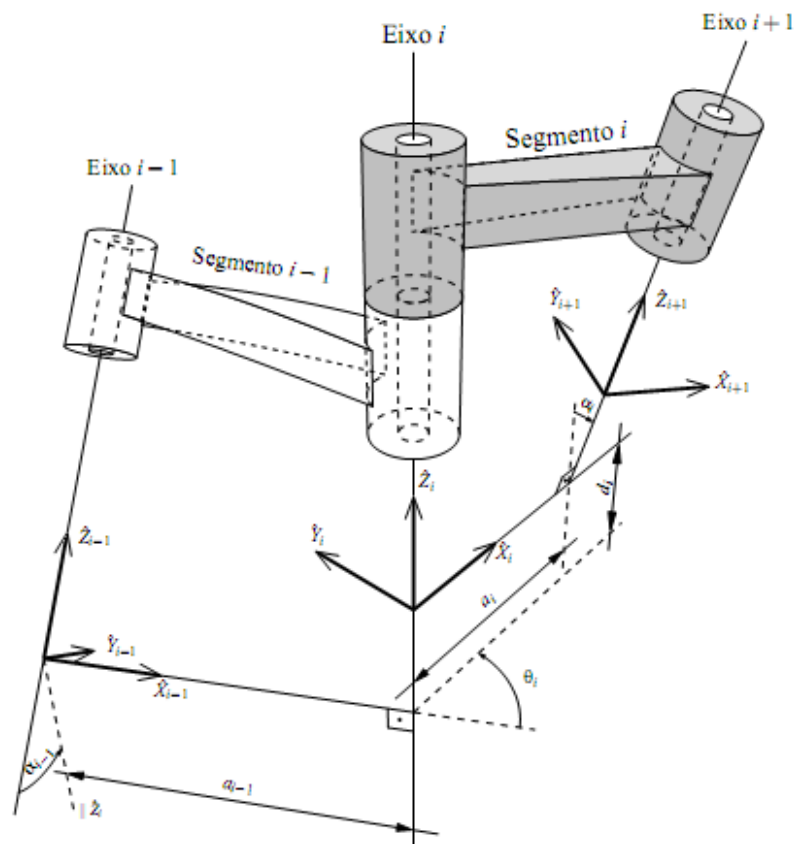


Figura 3.2. Parâmetros usados na notação Denavit-Hartenberg. (LAUS, 2006)

$${}^{i-1}_i T = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i & 0 & a_{i-1} \\ \cos \alpha_{i-1} \text{sen} \theta_i & \cos \alpha_{i-1} \cos \theta_i & -\text{sen} \alpha_{i-1} & -d_i \text{sen} \alpha_{i-1} \\ \text{sen} \alpha_{i-1} \text{sen} \theta_i & \text{sen} \alpha_{i-1} \cos \theta_i & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.2}$$

A partir da tabela 3.1 e da posterior substituição dos valores de cada linha i na equação 3.2, identificou-se a transformação homogênea (${}^{i-1}_i T$) para cada linha desta tabela, conforme indicado nas equações (3.3) a (3.8):

Tabela 3.1 - Relação cinemática entre as juntas (LAUS, 2006).

i	${}^{i-1}_i T$	α_{i-1}	a_{i-1}	θ_i	d_i
1	${}^0_1 T$	0	0	θ_1	0
2	${}^1_2 T$	90°	0	θ_2	0
3	${}^2_3 T$	0	a_2	θ_3	0
4	${}^3_4 T$	0	a_3	$90^\circ + \theta_4$	0
5	${}^4_5 T$	$+90^\circ$	0	θ_5	d_5
6	${}^5_6 T$	-90°	0	θ_6	0

$${}^0_1 T = \begin{bmatrix} \cos(\theta_1) & -\text{sen}(\theta_1) & 0 & 0 \\ \text{sen}(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.3}$$

$${}^1_2 T = \begin{bmatrix} \cos(\theta_2) & -\text{sen}(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \text{sen}(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.4}$$

$${}^2_3T = \begin{bmatrix} \cos(\theta_3) & -\text{sen}(\theta_3) & 0 & a_2 \\ \text{sen}(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.5}$$

$${}^3_4T = \begin{bmatrix} \cos(\theta_4) & -\text{sen}(\theta_4) & 0 & a_3 \\ \text{sen}(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.6}$$

$${}^4_5T = \begin{bmatrix} \cos(\theta_5) & -\text{sen}(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ \text{sen}(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.7}$$

$${}^5_6T = \begin{bmatrix} \cos(\theta_6) & -\text{sen}(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\text{sen}(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.8}$$

Multiplicando-se as transformações homogêneas obtidas, têm-se as equações (3.9) e (3.10), que permitem identificar a posição e orientação da sexta junta (onde estará acoplada a ferramenta de solda) em relação à base do robô (identificada como sistema de coordenadas $\{0\}$).

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T \quad \text{Eq. 3.9}$$

$${}^0_6T = \begin{bmatrix} c_6s_1s_5 - s_6c_1s_{234} + c_1c_5c_6c_{234} & -s_1s_5s_6 - c_1c_6s_{234} - c_1c_5s_6c_{234} & c_5s_1 - c_1s_5c_{234} & d_5c_1s_{234} + c_1(a_2c_2 + a_3c_{23}) \\ -c_6c_1s_5 - s_6s_1s_{234} + s_1c_5c_6c_{234} & c_1s_5s_6 - s_1c_6s_{234} - s_1c_5s_6c_{234} & -c_5c_1 - s_1s_5c_{234} & d_5s_1s_{234} + s_1(a_2c_2 + a_3c_{23}) \\ c_5c_6s_{234} + s_6c_{234} & -c_5s_6s_{234} + c_6c_{234} & -s_5s_{234} & -d_5c_{234} + a_2s_2 + a_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.10}$$

onde:

$$\begin{aligned} c_i &= \cos(\theta_i) \\ s_i &= \text{sen}(\theta_i) \\ c_{234} &= \cos(\theta_2 + \theta_3 + \theta_4) \\ s_{234} &= \text{sen}(\theta_2 + \theta_3 + \theta_4) \end{aligned}$$

Esta última transformação homogênea (eq. 3.10) descreve a posição (através da última coluna) e a orientação (através da submatriz de rotação) do punho do robô em função dos ângulos $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ e θ_6 que aparecem na equação na forma de senos e cossenos desses ângulos ou da soma deles. Os outros parâmetros da equação 3.10 (a_2, a_3 e d_5) normalmente são constantes conhecidas, pois podem ser obtidos do manual do robô, interpretando-se algum desenho ou por processo de calibração (identificação) do robô.

3.2 Cinemática inversa

O objetivo da cinemática inversa é obter o valor de todas as variáveis de junta para se atingir uma posição/orientação (meta) desejada. Há três problemas a se considerar:

- A existência de mais de um conjunto (vetor) de soluções;
- A inexistência de qualquer solução (por falta de alcance ou destreza do robô);
- A contenção de soluções (por falta de graus de liberdade no manipulador ou extremidade do robô).

A existência de mais de um conjunto (vetor) de soluções pode representar um problema adicional, já que pode haver um número finito (ou até infinito) de soluções e deve-se escolher uma delas. Podem existir infinitas soluções no caso de uma ou mais juntas do robô serem *redundantes*¹ para a tarefa ou se tornarem *redundantes* por estar trabalhando em uma posição/orientação *singular*². A figura 3.3 é apresentada para exemplificar um caso em que, para uma dada posição e orientação do manipulador (extremidade do robô), há oito soluções diferentes para os ângulos das juntas do robô.

A inexistência de qualquer solução pode ocorrer devido à posição/orientação estar fora do “espaço de trabalho” do robô. Define-se, então, espaço de trabalho como sendo o conjunto de posições cartesianas que o robô pode atingir. Neste caso, as únicas soluções para o problema são a substituição do robô por outro de espaço de trabalho adequado ou a aproximação do robô da meta (mover a base do robô para que ele fique mais próximo da meta). A inexistência de qualquer solução também pode ocorrer por falta de destreza do robô (o robô pode atingir a posição desejada, mas não com a orientação desejada). Diz-se,

¹ Um robô manipulador é chamado de redundante quando o número de graus de mobilidade do mecanismo é maior que o número de variáveis necessárias para realizar uma tarefa (SCIAVICCO e SICILIANO, 1996).

² A singularidade ocorre quando ocorre o alinhamento de dois ou mais eixos do sistema de coordenadas do manipulador (SCIAVICCO e SICILIANO, 1996).

neste caso, que a meta está fora do “espaço de trabalho destro”. Define-se, então, espaço de trabalho destro como sendo o conjunto de posições cartesianas que o robô pode atingir com qualquer orientação. Naturalmente, o espaço de trabalho destro é menor que o espaço de trabalho (total) já que há muitas posições (com o braço quase totalmente esticado ou recolhido) que o robô pode atingir, mas com restrição na orientação (não é possível uma orientação arbitrária) (LAUS, 2006).

Existem manipuladores no mercado com menos do que seis graus de liberdade e que, portanto, não podem atingir qualquer posição/orientação (o espaço de trabalho destro é um conjunto vazio) (SANTOS, 2007). Neste caso, a cinemática inversa deve ser calculada levando-se em conta que não é possível obter uma orientação arbitrária. Há duas técnicas para isso:

- Projetar a posição/orientação desejada sobre o espaço de trabalho do robô (implicando na perda de um ou mais graus de liberdade) e calcular a solução para a posição/orientação projetada;
- Desenvolver a cinemática inversa usando, por comparação, a cinemática do robô com a cinemática atingível (obtida, por exemplo, usando ângulos de Euler como um dos ângulos constantes). Esta técnica apresenta a vantagem de fornecer a cinemática direta em uma forma intuitiva para o programador, porém há a desvantagem da necessidade de que as outras formas de descrição sejam traduzidas para a forma usada e não para transformações homogêneas, o que é mais genérico.

De qualquer forma, há a necessidade da determinação de soluções específicas para cada manipulador ou conjunto manipulador/ferramenta.

3.3 Programação de robôs

A programação dos robôs industriais também pode receber o nome de ensino e pode envolver diferentes dispositivos de comunicação homem-máquina. Ela compreende, além do dispositivo ao qual se introduz a informação, a linguagem empregada na informação. Logo, entende-se por programação a forma e os meios que um robô dispõe para ser informado sobre a tarefa que deve realizar.

Um robô normalmente é programado através da inclusão de comandos na memória de seu controlador. Existem quatro métodos de programação de robôs (GROOVER, 1987):

- *Setup* manual - nesta forma, a “programação” é feita através da colocação de obstáculos físicos que são utilizados como barreiras para controlar os pontos finais de cada

movimento. No entanto, a simples colocação de obstáculos não corresponde precisamente a um método de programação, e sim a um controle mecânico;

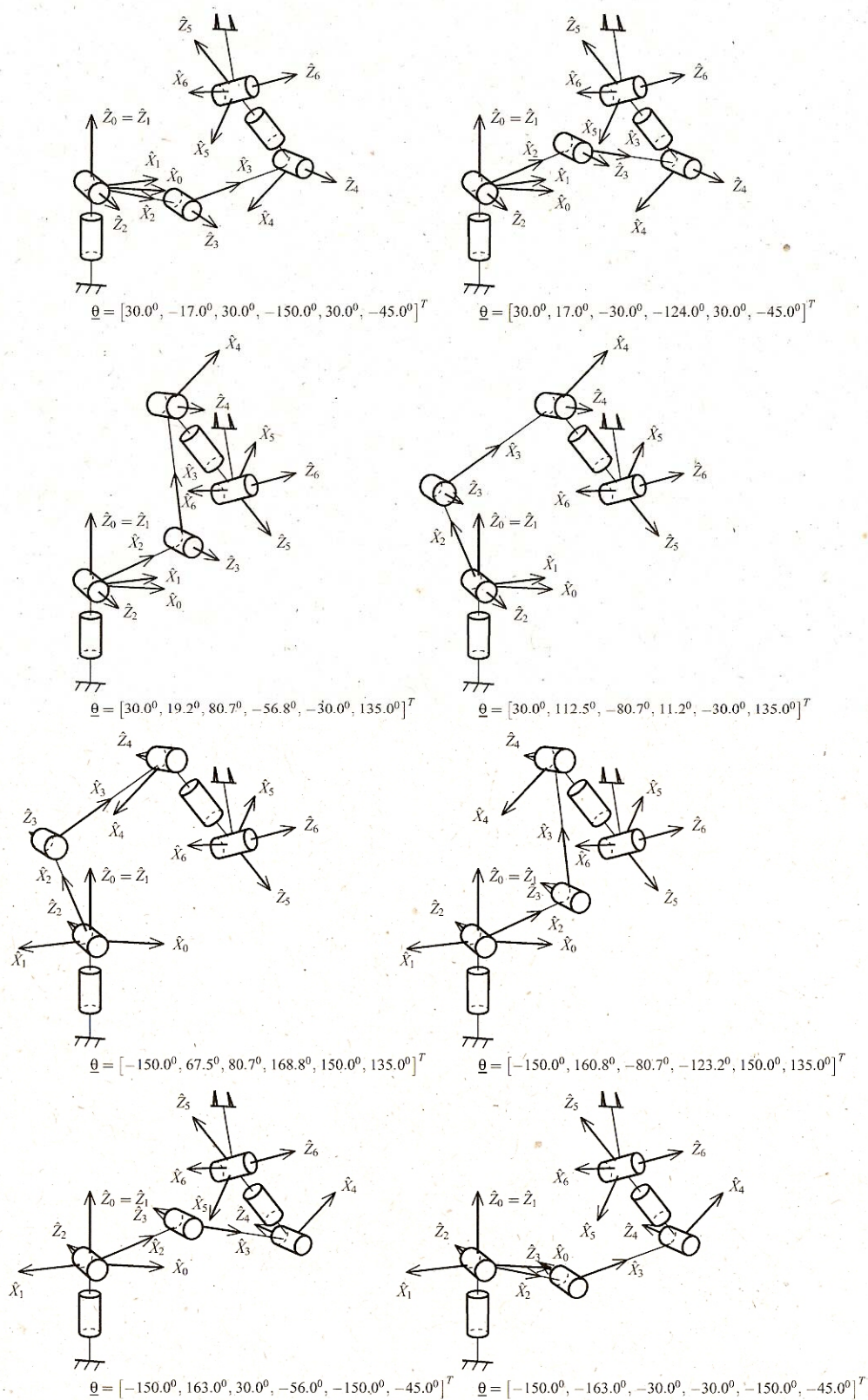


Figura 3.3. Oito soluções para um robô de seis graus de liberdade (LAUS, 2006).

- Programação ponto-a-ponto ou *leadthrough* - o programador precisa movimentar o robô de acordo com o movimento desejado durante um processo de aprendizado, incluindo, desta forma, os pontos por onde o robô deve se deslocar na memória do controlador. Esse método é mais adequado para tarefas de trajetória contínua, como: pintura por pulverização, limpeza, aplicação de cola ou solda a arco, cuja programação seria extremamente tediosa de outra forma. Quem programa dessa maneira deve conhecer bem a tarefa. Essa programação pode ser feita também de modo ponto a ponto, onde o robô apenas grava certos pontos e, na repetição, move-se entre eles em linha reta (ou trajetórias circulares, se assim for solicitado). Essa maneira é mais utilizada em tarefas como erguer e abaixar objetos, cuja posição é conhecida com exatidão. Este tipo de programação envolve o uso de um uma espécie de controle portátil para guiar o robô ao longo da trajetória desejada, dotado de um teclado e um "joystick", comumente chamado de "teach-box". Quando o programador especifica um trajeto contínuo ou ponto a ponto, usando coordenadas cartesianas, os sinais do programa vão para o controlador, que os transforma em coordenadas de juntas e envia os sinais apropriados para os atuadores moverem as juntas e produzirem o movimento desejado do robô. A figura 3.4 mostra o controle utilizado pelos robôs da *Asea Brown Boveri* (ABB), denominado por eles de *Flex-pendant*.

Apesar de ser um método simples de programar, possui algumas desvantagens como: a necessidade de interromper o processo de produção e esperar a programação; ser limitado em termos de incorporar mecanismos de tomada de decisão no programa e não ser compatível com tecnologias de computação existentes (CAD/CAM, comunicação e banco de dados);

- Programação direta em Linguagens de programação próprias - semelhante à programação de um computador, respeitando-se as características do robô. Neste caso, para a confecção do programa, não é necessário a utilização do robô. Apresenta várias vantagens como: um melhor tratamento dos sensores; capacidade de controlar dispositivos externos; aumento da capacidade de expressar a lógica do programa e possibilidade de comunicar com sistemas computacionais externos permitindo o acesso a vários recursos como bases de dados ou bases de conhecimento;
- Programação *off-line* – similar ao tipo anterior, utiliza-se uma linguagem de programação própria (com as vantagens já indicadas), mas o desenvolvimento não é feito diretamente sobre o robô e sim através de um sistema de computação normal, permitindo simular o programa que, somente quando pronto e correto, será enviado ao robô para execução. A grande vantagem deste método é o fato de não ser necessário interromper o processo de produção para construir e testar o programa.



Figura 3.4 - Controle para programação de robôs ABB *Flex-pendant* (ABB, 2004).

- Programação *off-line* – similar ao tipo anterior, utiliza-se uma linguagem de programação própria (com as vantagens já indicadas), mas o desenvolvimento não é feito diretamente sobre o robô e sim através de um sistema de computação normal, permitindo simular o programa que, somente quando pronto e correto, será enviado ao robô para execução. A grande vantagem deste método é o fato de não ser necessário interromper o processo de produção para construir e testar o programa.

Em geral, os métodos apresentados são fáceis de aplicar, mas apresentam um problema em comum: consomem muito tempo durante a programação. Se fosse necessário reprogramar 250 robôs a cada meia hora, considerando-se cerca de uma hora para a programação de cada um deles, o tempo gasto, além do custo da mão de obra de programadores, “amarrariam” demasiadamente a linha de produção, que ficaria indisponível para a produção durante o período de programação dos robôs. Felizmente, o ensino só consome tempo na primeira vez em que uma tarefa é programada. Uma vez que os movimentos estejam corretos, eles podem ser, rapidamente, transferidos por meios eletrônicos para um, vários ou todos os robôs da linha de produção. Quando os movimentos gravados não são mais necessários para a tarefa imediata que está sendo executada, eles podem ser salvos para uma chamada posterior fácil e rápida (SANTOS, 2001).

Uma grande desvantagem dos métodos de programação *leadthrough* e *off-line* é que eles não utilizam sinais de sensores. Às vezes, os robôs precisam de sensores para guiá-los em seu trabalho e programar sem sensores é complicado demais, considerando a tarefa que tem de ser feita. Por exemplo, suponha que se queira “treinar” um robô para

desempilhar pequenas caixas de uma plataforma. Mesmo que todas as caixas sejam do mesmo tamanho, elas podem estar orientadas de maneira diferente e empilhadas em diferentes níveis. Dever-se-ia usar programação por condução ou por aprendizagem para treinar o robô a remover caixas em pilhas de 25, então de 24 e depois de 23 caixas de altura? Pareceria mais rápido que alguém fizesse o trabalho e se esquecesse do robô. Não será possível encontrar um modo de ensinar o robô a empilhar caixas de qualquer tamanho, dispostas em qualquer orientação e empilhadas a qualquer altura? Certamente seria usado para essa tarefa um método de instrução que utilizasse sensores com um programa "*off-line*", interligando o robô a um sistema de visão (SANTOS, 2001).

Como pode ser visto, o uso de uma linguagem de programação, *off-line* ou não, traz muitas vantagens para a área de robótica. No entanto, estas linguagens de programação apresentam algumas características que prejudicam o desenvolvimento de sistemas robóticos, no que diz respeito à engenharia de *software*:

- As linguagens utilizadas são do paradigma imperativo, e algumas delas são do tipo *assembly*, ou seja, as linguagens utilizadas pertencem a um nível próximo da máquina, o que faz com que o programador desvie sua atenção da lógica do programa para aspectos do *hardware*;
- O controle do fluxo de execução é realizado através de desvios incondicionais (*go to*). Mesmo nos desvios condicionais, a execução é desviada para um ponto qualquer do programa definido por um rótulo. Isto prejudica os critérios da programação estruturada;
- Não existe portabilidade de código fonte, pois cada linguagem de programação é específica para um robô;
- Como consequência do item anterior, a reutilização só é possível entre programas feitos para as mesmas máquinas;
- Devido à não portabilidade de código fonte, os programadores precisam ser treinados sempre que houver alguma troca de equipamento.

3.4 Planejamento da trajetória de robôs

Segundo FU *et al.*(1987), para controlar um manipulador é necessário que a configuração do braço do robô nas posições inicial e final sejam especificadas antes que a trajetória de movimento seja planejada. As restrições de uma típica trajetória de junta são mostradas a seguir:

a. Restrições de posição

- Posição inicial: velocidade e aceleração são dadas (normalmente zero)
- Posição superior: movimentação contínua para pontos intermediários
- Posição inferior: similar à posição superior
- Posição final: velocidade e aceleração são dadas (normalmente zero)

b. Restrições físicas e geométricas

- Limites mecânicos de cada junta do manipulador (área de trabalho)

c. Considerações de tempo

- Segmentos inicial e final da trajetória: o tempo é baseado na taxa de aproximação do ponto central do punho (*TCP*)
- Pontos intermediários ou segmento da trajetória média: o tempo é baseado na máxima velocidade e aceleração das juntas e o máximo destes tempos é usado (neste caso, o tempo máximo da junta mais lenta é usado por normalização).

Baseado nestas restrições pode ser feita a seleção de uma classe de funções polinomiais de grau n ou menor, de modo que sejam contemplados cada um dos parâmetros definidos (posição de junta, velocidade e aceleração) das posições inicial, superior, inferior e final. Um exemplo de abordagem para esta situação é a equação polinomial (3.9) de grau sete (FU *et al.*, 1987):

$$q_i(t) = a_7t^7 + a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \quad \text{Eq. 3.9}$$

onde i representa o número de juntas do manipulador e os coeficientes a_j podem ser determinados a partir das posições conhecidas e condições de continuidade. Porém, o emprego de um polinômio de alto grau para interpolar os pontos dados pode não ser satisfatório. Uma abordagem alternativa é dividir a trajetória completa em diversos segmentos que podem ser interpolados através de polinômios de baixo grau, simplificando consideravelmente o cálculo envolvido. Há diferentes métodos de dividir uma trajetória de junta, e cada um deles apresenta diferentes propriedades. Os métodos mais comuns são os seguintes:

Trajectoria 4-3-4. Neste método, cada junta possui três segmentos de trajetória: o primeiro segmento é um polinômio de grau quatro que especifica a trajetória desde a posição inicial até a posição superior. O segundo segmento, por sua vez, é um polinômio de grau três que

especifica a trajetória da posição superior à posição inferior. O último segmento de trajetória é um polinômio de grau quatro que especifica a trajetória desde a posição inferior até à posição final.

Trajétória 3-5-3. Similar ao método de trajetória 4-3-4, porém utiliza polinômios de graus diferentes para cada segmento: um polinômio de grau três para o primeiro segmento, um polinômio de grau cinco para o segundo segmento e um polinômio de grau três para o último segmento.

Trajétória 5-Cúbica. Neste método, são utilizadas funções *spline* de polinômios de grau três para uma trajetória de cinco segmentos.

Note que a discussão apresentada é válida para cada trajetória de eixo; isto é, cada trajetória de eixo é dividida em trajetórias de três ou cinco segmentos. O número de polinômios para o método 3-4-3 de um manipulador de N -juntas terá N trajetórias de junta ou $N * 3 = 3N$ segmentos de trajetória e $7N$ coeficientes polinomiais para validar os segmentos de trajetória $3N$. Uma discussão detalhada dos métodos de trajetória 4-3-4 e 5-cúbico pode ser consultada em FU *et al.* (1987).

A principal característica que diferencia as linguagens de programação de robôs das linguagens de programação convencionais é a necessidade de marcar/planejar os pontos da trajetória. Isto é preciso porque o efetuador (extremidade) de um robô pode ser movimentado de um lugar para outro, e em cada lugar ele poderá fazer alguma tarefa (soldar, pintar, pegar, largar, *etc.*). Cada um destes lugares possui uma posição e uma orientação associadas a ele. Devido aos objetivos deste trabalho, será detalhada a seguir a programação de robôs voltados para a soldagem.

3.5 Programação de robôs de soldagem

Segundo ROMANO (2002), a utilização de robôs para soldagem industrial exige um conjunto de componentes básicos e a execução da programação das trajetórias, que o robô deverá executar durante o ciclo de trabalho. Normalmente, esses componentes são agrupados e confinados em células de soldagem. A figura 3.5 apresenta uma célula de soldagem robotizada. A tabela 3.2 destaca os componentes básicos de um sistema de soldagem industrial, bem como suas principais características.

A programação *leadthrough* de um robô para a realização de soldas obedece à seguinte seqüência:

- 1 Calibração da posição da tocha de soldagem, para assegurar que o robô irá trabalhar dentro de sua faixa de tolerância;
- 2 Localização dos componentes de trabalho (mesa, gabaritos, fixadores) e definição do tipo de fonte de energia;
- 3 Definição do caminho que será seguido pela tocha (ou seja, a trajetória de trabalho), assim como a localização dos cordões de solda;
- 4 Definição das condições de soldagem a serem utilizadas e que devem estar atreladas às coordenadas de movimentação do robô. Nessa etapa da programação, são inseridos dois tipos de dados: o primeiro no início da trajetória, onde será depositado o cordão, contendo informações para a abertura de arco e o segundo no final da trajetória, contendo informações para a extinção do arco.
- 5 Refinamento do programa através da verificação do desempenho e introdução de alguns controles básicos para soldagem.



Figura 3.5 - Célula de soldagem robotizada.

Por exemplo, para programar um robô para soldar um componente é feita a orientação da tocha de soldagem, posicionada no final do braço do robô (punho). A movimentação ou manipulação das juntas do robô, assim como a determinação do caminho a ser seguido, é realizada através do *teach-pendant* (figura 3.4). Os pontos ao longo da trajetória são armazenados na memória do controlador do robô (o programador guia fisicamente a tocha de soldagem ao longo da trajetória desejada, gravando pontos). O robô irá repetir essa trajetória tendo como referência as coordenadas da posição de calibração.

Tabela 3.2. Componentes básicos de um sistema de soldagem robotizado (ROMANO, 2002).

COMPONENTES BÁSICOS	PRINCIPAIS CARACTERÍSTICAS
Robô	Graus de liberdade, área de trabalho utilizável, precisão, sistema de controle, repetibilidade, velocidade e capacidade de carga
Controlador do robô	Interface de programação, capacidade de memória para armazenamento do sistema operacional e do programa de trabalho, e <i>software</i> de aplicação para soldagem
Fonte de energia para soldagem	Ciclo de trabalho (usualmente 100% é desejável), interface com o controle do robô, precisão, confiabilidade e robustez
Equipamentos e acessórios	Alimentador de arame, tocha de soldagem, estação de limpeza do bocal da tocha, ferramentas para alinhamento da tocha e equipamentos de segurança.

Caso a espessura ou as dimensões da peça sejam diferentes da utilizada durante a programação da trajetória e das condições de soldagem do robô, o cordão de solda será posicionado em uma posição diferente da desejada, ou seja, não cobrirá completamente o chanfro ou a junta, ou então o fará de forma incompleta. Consequentemente, a precisão do trabalho realizado pelo robô depende, além de sua precisão de posicionamento, da qualidade da fixação da peça pelos dispositivos e do controle rigoroso da tolerância dimensional dos componentes a serem soldados.

Devido às dificuldades apresentadas, torna-se cada vez mais evidente a necessidade de técnicas de auxílio à programação de robôs, permitindo aos programadores a possibilidade de criação de programas de robôs mais robustos, estruturados e concebidos de maneira a utilizar os recursos disponíveis do equipamento desde o primeiro momento, eliminando a necessidade de correções posteriores para a melhoria de desempenho nas tarefas desejadas. Algumas destas técnicas, bem como exemplos de aplicações, serão mostrados na seqüência.

4. TÉCNICAS DE OTIMIZAÇÃO

A grande incidência de problemas de otimização em áreas como telecomunicações, logística e produção vem motivando o desenvolvimento de técnicas de otimização, que usualmente são formadas pela adaptação de idéias de várias áreas de pesquisa. Em muitos problemas que envolvem alocação de recursos, embalagem e programação da produção, podem ser usados métodos baseados em programação matemática (programação linear e /ou inteira). Porém, esses métodos tradicionais, em geral, trabalham com estruturas complexas de modelagem, assumem quantidades determinísticas e têm problemas para lidar com qualquer efeito estocástico. Não há como negar que estes métodos têm produzido excelentes resultados nos últimos 40 anos, e precisamente por causa do seu sucesso, é usual modelar problemas usando esse tipo de estrutura para se chegar a uma solução. Ainda que esses modelos sejam eficientes para muitos casos, nos últimos anos, diversos métodos heurísticos têm sido desenvolvidos, possibilitando a modelagem de problemas sem a necessidade de utilização das mesmas estruturas. Há duas razões para a adoção de métodos heurísticos: dificuldade de modelar o problema usando programação matemática; o tempo de solução do problema ser extremamente alto usando programação matemática. Isso possibilitou a construção de modelos onde o objetivo deixou de ser a identificação de soluções “ótimas”, como ocorre na programação matemática, e passou a ser a identificação da melhor solução possível dentro do tempo ou critério de parada especificado (RODRIGUES, 2006).

No contexto da determinação de trajetórias satisfatórias para manipuladores robóticos, várias métodos têm sido propostos e pesquisados nos últimos anos. Por exemplo, PIRES e MACHADO (1999) têm usado AG (Algoritmos Genéticos) para a geração de trajetórias livres de colisão para manipuladores robóticos com o objetivo de minimizar a extensão do caminho percorrido. Os autores têm usado a cinemática direta para este propósito e têm apresentado resultados para diversos robôs manipuladores redundantes e não-redundantes. Similarmente, CHEN e ZALZALA (1997) têm aplicado a aproximação de AG para o planejamento de movimentação de sistemas de manipuladores móveis. Distância mínima de trajeto e percurso seguro foram considerados como os dois critérios para o planejamento do robô. A ênfase do estudo foi utilizar AG para buscar uma solução global ótima e resolver o problema de minimização da distribuição de torque do manipulador. Vários resultados de simulação para dois exemplos propostos em que foram utilizados a aproximação por AG se mostraram melhores que métodos convencionais de busca.

ÇAKIR *et al.* (2006) empregou em seu artigo AG para o planejamento de trajetória de robôs industriais com seis graus de liberdade (*DOF*), em adição com um tipo de álgebra alternativa chamado de teorema dos *quaternions*. Para empregar o método baseado em AG, cada cromossomo da população foi transformado em um vetor que indicava as coordenadas de posição utilizando valores reais no primeiro estágio, representando uma solução para o problema. Por sua vez, WU *et al* (2000) abordou um problema de coordenação entre um manipulador de seis eixos e uma mesa de posicionamento de três eixos aplicados a um sistema de soldagem a arco, empregando algoritmos genéticos para solucionar o problema de planejamento de trajetória do manipulador.

A revisão, apresentada a seguir, aborda três métodos heurísticos: busca tabu, algoritmos genéticos e *simulated annealing*. A filosofia da Busca Tabu (BT) se deriva e pode explorar uma coleção de mecanismos de memória dos movimentos executados e das soluções geradas anteriormente. Algoritmos genéticos (AG) buscam imitar o fenômeno biológico da reprodução e da evolução. O *simulated annealing* (SA) é baseado na simulação de um processo físico da metalurgia, a têmpera ou o recozimento, sendo também chamado de têmpera simulada ou recozimento simulado.

As heurísticas foram consideradas durante muito tempo modelos cognitivos. Elas se constituem de regras baseadas na experiência e na memória das soluções obtidas durante o processo de busca para a geração de novas soluções. Os métodos heurísticos diferem dos métodos exatos, como a programação matemática, porque estes últimos normalmente constroem as soluções durante o processo de busca, enquanto os métodos heurísticos usam a(s) solução(ões) vigente(s) para gerar nova(s) solução(ões) “vizinha(s)” da(s) vigente(s).

Uma técnica heurística (ou simplesmente, uma heurística) é um método de busca de soluções a um custo computacional aceitável, porém sem a garantia absoluta de otimalidade. Infelizmente, não é possível mesmo indicar quão perto da otimalidade uma solução heurística em particular está (SMITH *et al*, 1996). As técnicas heurísticas não asseguram a identificação da solução ótima, mas podem trabalhar tanto com soluções factíveis como infactíveis. No caso de usar soluções infactíveis, esta ação destina-se exclusivamente a permitir que a busca “fuja” de pontos de mínimo ou máximo local, necessitando-se garantir que a busca posteriormente convergirá para uma solução factível.

4.1 Busca tabu

A Busca Tabu (BT) é uma meta-heurística que guia um procedimento de busca heurística local, de forma a permitir que o espaço de solução alcance um ponto de

otimalidade local, mas sem ficar preso a esta região do espaço de busca. Um dos principais componentes da BT é o uso de diferentes mecanismos de memória, o que cria um comportamento de busca mais flexível. Há uma diversidade de aplicações da busca tabu, como no seu emprego na resolução de problemas de planejamento e programação de produção, alocação de recursos, *layout* de células, *etc.*

O método da Busca Tabu, qualificado como inteligente, baseia-se na premissa de que a solução de um problema deve incorporar mecanismos de memória e exploração responsiva (escolhas estratégicas). Uma boa analogia é a escalada de montanha, onde o alpinista deve lembrar seletivamente (“memorizar”) elementos chave do caminho percorrido e deve estar apto a fazer escolhas estratégicas ao longo do caminho. O uso da memória (dos últimos movimentos e/ou soluções geradas) na BT permite a implementação de procedimentos capazes de explorar uma grande porção do espaço de soluções com requisitos mínimos de memória.

O método de busca do algoritmo envolve três etapas a cada iteração: montagem da vizinhança, avaliação e escolha da próxima solução. Na montagem da vizinhança, realiza-se uma troca de posição de todos os elementos da solução corrente da iteração. Ou seja, gera-se todas as soluções vizinhas da solução vigente. Cada movimento representa um vizinho, que é uma solução viável. Na avaliação, identifica-se qual é a melhor solução da vizinhança analisada, para se tornar a solução vigente (s) no passo seguinte. Esse movimento é guardado na memória de curto prazo, também chamada lista tabu, a qual é responsável por guardar os movimentos realizados na mudança da solução corrente (ou vigente), entre cada iteração. Nessa mudança, o algoritmo faz com que este movimento seja tabu ou proibido por um número k de iterações. A cada iteração, atualiza-se a memória de curto prazo (lista tabu).

Para a escolha da próxima solução, verifica-se a memória (lista tabu) para se continuar a busca na próxima iteração. Para esta solução, verifica-se também se algum movimento tabu gera uma solução melhor que a melhor solução já encontrada até então (s^*). Caso ela seja melhor que s^* , o movimento tabu é aceito e esta solução é gravada como nova solução vigente. Este procedimento é chamado de **Critério ou Função de Aspiração**.

Em resumo, segundo MICHALEWICZ e FOGEL (2000), a idéia principal por trás da BT é muito simples. Uma “memória” força a busca a explorar novas áreas do espaço de busca. Pode-se memorizar os movimentos que tenham sido executados recentemente, e estes tornam-se “tabu” (proibidos) no processo de seleção da próximas k soluções, assumindo que qualquer movimento possa ficar tabu por k iterações. Note que a BT é basicamente determinística (opostamente ao *simulated annealing*), mas é possível adicionar alguns elementos probabilísticos a ela.

Na figura 4.1 é apresentada a estrutura básica da Busca Tabu. Uma forma de explicar os conceitos da BT é através de um exemplo. Suponha que se esteja resolvendo o problema SAT, como é conhecido na literatura (que visa a satisfação de todas as restrições do problema), com oito variáveis ($n = 8$). Assim, para uma dada solução, busca-se uma função que valide as oito variáveis, de tal maneira que a satisfação das restrições seja analisada. Assuma que a solução inicial, gerada aleatoriamente, é $\mathbf{x} = (x_1, \dots, x_8)$, onde $\mathbf{x} = (0,1,1,1,0,0,0,1)$. A geração da solução inicial está indicada na linha 1 do procedimento de busca tabu, na figura 4.1.

Procedimento <i>BT</i>	
1.	Seja s_0 solução inicial;
2.	$s^* \leftarrow s$; (Melhor solução obtida até então)
3.	$Iter \leftarrow 0$; (Contador do número de iterações)
4.	$MelhorIter \leftarrow 0$; (Iteração mais recente que forneceu s^*)
5.	Seja $Btmax$ o número máximo de iterações sem melhora em s^* ;
6.	$T \leftarrow \emptyset$; (Lista Tabu)
7.	Inicialize a função de aspiração A ;
8.	enquanto ($Iter - MelhorIter \leq Btmax$) faça
9.	$Iter \leftarrow Iter + 1$;
10.	Seja $s' \leftarrow s \oplus m$ o melhor elemento de $V \subseteq N(s)$ tal que o movimento m não seja tabu ($m \notin T$) ou s' atenda a condição de aspiração ($f(s') < A(f(s))$);
11.	$T \leftarrow T - \{\text{movimento mais antigo}\} + \{\text{movimento que gerou } s'\}$;
12.	$s \leftarrow s'$;
13.	se $f(s) < f(s^*)$ então
14.	$s^* \leftarrow s$;
15.	Atualize a função de aspiração A ;
16.	$MelhorIter \leftarrow Iter$;
17.	fim-se;
18.	fim-enquanto;
19.	Retorne s^* ;
20.	Fim <i>BT</i>

Figura 4.1. Algoritmo da Busca Tabu (GLOVER e LAGUNA, 1997).

Usualmente, necessita-se de alguma função de avaliação que forneça parâmetros para a busca. Por exemplo, pode-se calcular uma soma de pesos, onde cada peso depende da satisfação de uma regra. Neste caso, a função de avaliação deve ser maximizada (*i.e.*, busca-se satisfazer todas as regras). No exemplo, assumiu-se que a solução inicial tem o valor de 27. Na figura 4.1, a linha 2 indica que a solução inicial é assumida como a melhor solução obtida até então (s^*). As linhas 3 a 5 tratam, respectivamente, da inicialização do contador de iterações ($Iter$); da identificação da iteração ($MelhorIter$) que gerou a melhor

solução obtida até então, s^* ; da definição do parâmetro $BTmax$ que indica o número máximo aceitável de iterações sem que ocorra a melhora no valor da solução s^* . A aplicação deste parâmetro será melhor explicada nos comentários da linha 8. Nas linhas 6 e 7 são inicializadas, respectivamente, a lista Tabu e a função de Aspiração. Note que inicialmente a lista tabu está “zerada”, já que nenhum movimento foi realizado até então. Também vale lembrar que a função de Aspiração será executada a cada iteração. Neste caso, como a lista tabu está “zerada”, a função de Aspiração ficará sem efeito na iteração zero (geração da solução inicial).

A cada iteração examina-se a vizinhança de x , a qual consiste, para este exemplo, de oito outras soluções, cada qual podendo ser obtida pela troca (ou inversão de valor) de um único *bit* no vetor x . Todas as soluções são, então, avaliadas e seleciona-se a melhor delas. Até este estágio da busca, isto é o mesmo que um procedimento *hill-climbing* (MICHALEWICZ e FOGEL, 2000).

Suponha que a troca de valor da terceira variável gerou a melhor solução (de valor 31), então este novo vetor gera a melhor solução corrente. Agora é o momento de introduzir a nova faceta da busca tabu: a lista tabu. Com o objetivo de guardar um registro das ações (movimentos), são utilizadas algumas estruturas de memória. Pretende-se relembrar o índice da variável que foi trocada (ou o movimento realizado), assim como o “momento” (ou iteração) quando este movimento foi feito, para então serem diferenciados os movimentos antigos e os mais recentes. No caso de um problema SAT, guarda-se na memória, para cada posição do vetor, qual foi a última iteração em que uma solução corrente foi gerada através deste movimento (troca do valor desta posição). O valor da memória, associado a cada posição, fornecerá informações sobre quão recente foi o último movimento desta posição em particular. Assumindo que qualquer movimento possa ficar “tabu” (proibido) por k iterações, um vetor M de memória é inicializado com valores 0 e, então, a cada iteração j da busca, define-se a entrada $M(i) = j + k$; quando ocorreu um movimento no i -ésimo *bit*, ou seja, quando o i -ésimo *bit* foi alterado. Note que, se $M(i)$ é igual a zero, isto implica que o i -ésimo *bit* nunca foi movimentado. Neste caso, a cada iteração j , basta comparar se o valor de j é maior ou igual a $M(i)$. Se isto for verdadeiro, $j \geq M(i)$, o i -ésimo *bit* pode ser “movido”. Caso contrário, $j < M(i)$, o movimento no i -ésimo *bit* será tabu.

O processo de geração de novas soluções vigentes pode ser dividido em duas estratégias: intensificação e diversificação. A intensificação corresponde à exploração de uma região do espaço de busca. A lista tabu é um mecanismo da BT para desestimular a repetição de soluções já “visitadas”. Há duas formas de identificar que, apesar da lista tabu, a busca está presa num ponto ou região de ótimo local: *i*) contando o número de iterações executadas desde a última alteração da solução s^* (melhor solução encontrada até então);

ii) memorizar soluções geradas e contar a sua repetição. No procedimento da figura 4.1, utiliza-se a primeira forma; ou seja, conta-se o número de iterações executadas desde a última alteração da solução s^* . O parâmetro $BTmax$ indica o número máximo aceitável de iterações sem que ocorra a melhora no valor da solução s^* . Assim, o procedimento de intensificação é repetido entre as linhas 8 e 19 da Figura 4. A linha 8 controla o fim do procedimento de intensificação, quando a diferença entre a iteração atual ($Iter$) e a última alteração da solução s^* ($MelhorIter$) se torna maior que $BTmax$. A linha 9 atualiza o contador de iterações, $Iter$. A geração e seleção da melhor solução na vizinhança da solução vigente s , assim como a condição de aspiração, são executadas na linha 10. Nas linhas 11 e 12, respectivamente, são atualizadas a lista Tabu e a nova solução vigente s . Se a solução s é a melhor solução gerada até então, como indica a linha 13, atualiza-se a informação da melhor solução gerada (s^*), entre as linhas 15 e 17. Assumindo que qualquer movimento possa ficar "tabu" (proibido) por k iterações, poderá ser feita uma nova interpretação através de uma entrada $M(i) = j+k$, quando ocorre o movimento do i -ésimo *bit* na iteração j . Pode-se interpretar que o i -ésimo *bit* poderá ser movido novamente após k iterações. Neste caso, a lista tabu (M) é atualizada a cada iteração, subtraindo-se uma unidade de cada posição desta memória, a menos que o valor em uma dada posição já esteja zerado. Sob esta interpretação, se k é igual a 4, o conteúdo da estrutura de memória M na primeira iteração do exemplo é dado na tabela 4.1. Na primeira iteração, $M(3)$ será igual a 4; mas na segunda iteração, $M(3)$ será igual a 3, já que subtrai-se uma unidade de cada posição desta memória a cada iteração, até que este valor seja zerado. Logo, após 4 iterações (i.e., k iterações), a informação armazenada nesta posição da memória será zerada. Isto pode ser interessante pelo fato de que a principal diferença entre estas duas interpretações equivalentes é simplesmente a maneira de implementação. A última aproximação interpreta o valor como o número de iterações pela qual uma dada posição não está disponível para qualquer movimento.

Tabela 4.1. Conteúdo da memória após uma iteração.

Variável (i)	1	2	3	4	5	6	7	8
Valor $M(i)$	0	0	4	0	0	0	0	0

Exemplificando a lista tabu, após quatro iterações adicionais com a seleção do melhor vizinho (o qual não é necessariamente melhor que a solução vigente) esta memória tem o conteúdo mostrado na tabela 4.2. Os números presentes na memória (tabela 4.2) indicam que as variáveis 2, 3, 5 e 8 estão disponíveis para serem "movidas" a qualquer

momento. A variável 1 não estará disponível pelas próximas duas iterações, pois $M(1) = 2$; a variável 3 (que antes era tabu) já estará disponível para ser escolhida na próxima iteração, pois $M(3) = 0$; e a variável 4 (que acabou de ser movida) não estará disponível pelas próximas quatro iterações, já que o movimento mais recente, realizado na iteração 5, ocorreu na variável 4 (i.e., $M(4) = 4$). A variável 6 não estará disponível pelas próximas três iterações, pois $M(6) = 3$; e, por fim, a variável 7 não estará disponível por mais uma iteração, pois $M(7) = 1$.

Tabela 4.2. Conteúdo da memória após cinco iterações.

Variável (i)	1	2	3	4	5	6	7	8
Valor $M(i)$	2	0	0	4	0	3	1	0

Considera-se, após a quinta iteração do exemplo analisado, que a solução atual (ou vigente) será $\mathbf{x} = (1,1,0,0,0,1,1,1)$ e que sua função de avaliação equivale a 33. Ao examinar a vizinhança de \mathbf{x} , haverá oito soluções possíveis, correspondentes ao movimento de cada uma das oito variáveis:

- $\mathbf{x}_1 = (0,1,0,0,0,1,1,1)$;
- $\mathbf{x}_2 = (1,0,0,0,0,1,1,1)$;
- $\mathbf{x}_3 = (1,1,1,0,0,1,1,1)$;
- $\mathbf{x}_4 = (1,1,0,1,0,1,1,1)$;
- $\mathbf{x}_5 = (1,1,0,0,1,1,1,1)$;
- $\mathbf{x}_6 = (1,1,0,0,0,0,1,1)$;
- $\mathbf{x}_7 = (1,1,0,0,0,1,0,1)$;
- $\mathbf{x}_8 = (1,1,0,0,0,1,1,0)$.

Ao avaliar cada uma das soluções vizinhas, calcula-se o valor da sua função de avaliação. Os movimentos memorizados que tenham sido feitos recentemente serão tabu (proibidos) para seleção da nova solução. Consequentemente, até a próxima iteração (iteração 6), o movimento das variáveis tabu (1, 4, 6 e 7) serão proibidos. Estas soluções proibidas (tabu) serão consideradas pela função de avaliação e a próxima solução será selecionada entre as soluções vizinhas \mathbf{x}_2 , \mathbf{x}_3 , \mathbf{x}_5 e \mathbf{x}_8 . Assim, a BT utiliza a memória

para forçar a busca a explorar novas áreas do espaço de busca. Suponha que a melhor função de avaliação destas quatro possibilidades é dada pela solução x_5 , a qual tem valor 32. Note que este valor representa um decréscimo na avaliação entre a melhor solução atual e a nova solução candidata. Após a sexta iteração, o conteúdo da memória muda, como indicado a seguir. Todos os valores da lista Tabu diferentes de zero são decrescidos de um para refletir a atualização de todos os movimentos tabu memorizados. Em particular, o valor $M(7)$ igual a 1 é mudado para $M(7)$ igual 0; i.e., a sétima variável foi “removida” da lista Tabu depois de quatro iterações. Também, desde que a nova solução atual selecionada resultou do movimento da quinta variável, o valor de $M(5)$ é mudado de zero para quatro (para as próximas quatro iterações, esta posição é tabu). Assim, após a sexta iteração o conteúdo da memória agora é como mostrado na tabela 4.3. Outras iterações são similarmente executadas. Em qualquer estágio, há uma solução atual sendo processada a qual implica uma vizinhança, e a partir dela, soluções tabu são eliminadas de uma possível exploração.

Tabela 4.3. Conteúdo da memória após seis iterações.

Variável (i)	1	2	3	4	5	6	7	8
Valor $M(i)$	1	0	0	3	4	2	0	0

A proibição de realizar movimentos tabu pode ser considerada demasiadamente restritiva. Pode acontecer com um dos vizinhos considerados tabu, por exemplo, x_6 , que esta solução forneça um excelente resultado da função de avaliação, que é melhor que qualquer solução encontrada anteriormente. Neste caso, deve-se deixar a busca mais flexível; ou seja, quando uma solução tabu é melhor que qualquer solução encontrada anteriormente, deve-se ignorar a proibição de realizar este movimento. Pela necessidade de fazer a busca mais flexível, a busca tabu considera soluções de toda a vizinhança, avaliando todas elas, e sob circunstâncias “normais” seleciona uma solução não-tabu como a próxima solução vigente, mesmo que esta solução não-tabu não tenha melhor avaliação que a solução vigente anterior. Mas, em circunstâncias que não são “normais”, i.e., se uma excelente solução tabu é encontrada na vizinhança, melhor que qualquer solução encontrada anteriormente, tal solução poderá ser tomada como nova solução vigente. Este procedimento de análise de movimentos tabu é chamado de critério de aspiração.

É claro que há também outras possibilidades para aumentar a flexibilidade da busca. Por exemplo, segundo GLOVER e LAGUNA (1997) pode-se mudar o procedimento prévio de seleção determinística para um método probabilístico, onde as melhores soluções

tenham uma chance maior de serem selecionadas. Adicionalmente, pode-se mudar o horizonte de memória durante a busca, já que as vezes pode ser interessante “memorizar” mais (favorecendo a diversificação) e outras vezes “menos” (i.e., quando o algoritmo atinge uma área promissora do espaço de busca, favorecendo a intensificação). Pode-se também ajustar este horizonte de memória para o tamanho do problema (i.e., memorizando os últimos \sqrt{n} movimentos, onde n indica a medida do tamanho da instância do problema).

A estrutura da memória discutida até aqui, a lista Tabu, pode ser denominada memória baseada em recência, pois ela somente guarda algumas ações das últimas iterações. Esta estrutura pode ser estendida para a chamada memória baseada em frequência, a qual opera sobre um horizonte muito maior. Por exemplo (referindo-se ao problema SAT considerado anteriormente), um vetor H pode servir como uma memória de longo prazo. Este vetor é inicializado com valores iguais a zero e, em qualquer estágio da busca, a entrada “ $H(i) = k$ ” é interpretada como indicando que durante nas últimas h iterações do algoritmo, a i -ésima variável foi movida k vezes. Usualmente, o valor do horizonte h é “significativamente grande”, ao menos em comparação com o horizonte da memória baseada em recência. Assim, depois de 100 iterações com h igual a 50, a memória de longo prazo H pode ter o valor mostrado na tabela 4.4. Esta frequência de contagem mostra a distribuição dos movimentos ao longo das últimas 50 iterações. Como se pode usar esta informação? Os princípios da BT indicam que este tipo de memória pode ser útil para diversificar a busca. Por exemplo, a memória baseada em frequência fornece informações sobre quais movimentos têm sido subutilizados (i.e., menos frequentes) ou não representados totalmente, e pode-se diversificar a busca explorando estas possibilidades.

Tabela 4.4. Conteúdo da memória baseada em frequência para $h = 50$.

Variável (i)	1	2	3	4	5	6	7	8
Memória por frequência	5	7	11	3	9	8	1	6

O uso de memórias de longo prazo na Busca Tabu é normalmente restrito a algumas circunstâncias especiais. Por exemplo, pode-se encontrar uma situação onde todos os movimentos não-tabu levam a soluções inferiores, em relação à última solução. Assim, para tomar uma decisão sobre qual a próxima direção explorar, pode ser interessante consultar o conteúdo da memória de longo prazo. Há muitas possibilidades aqui para incorporar esta informação dentro de um processo de tomada de decisão. A aproximação mais típica torna os movimentos de maior frequência menos atrativos. Usualmente, o valor do resultado da avaliação é decrescido por alguma medida de penalidade que depende da frequência, e o

resultado final indica o movimento a ser realizado. Para ilustrar este fato através de um exemplo, assuma que o valor da solução atual s , para o problema SAT analisado, é 35. Os movimentos não-tabu (movimentos 2, 3 e 7) resultam em valores de 30, 33 e 31, respectivamente, e nenhum dos movimentos tabu resulta em um valor maior que 37 (o maior valor encontrado até então), então o critério de aspiração não terá efeito. Nestas circunstâncias, pode-se utilizar a memória baseada em frequência. Pode-se assumir que a fórmula de avaliação para uma nova solução s' , usada em certas circunstâncias, é:

$$F(s') = \text{aval}(s') - \text{penalidade}(s') \quad \text{Eq. 4.1}$$

onde $\text{aval}(s')$ retorna o valor original da função de avaliação (i.e., 30, 33 e 31, respectivamente, para soluções criadas pelo movimento da segunda, terceira e sétima variáveis), e:

$$\text{penalidade}(s') = 0,7 \cdot H(i) \quad \text{Eq. 4.2}$$

onde 0,7 serve como um coeficiente e $H(i)$ é o valor pego da memória de longo prazo H :

$H(2) = 7$, para uma solução criada movendo a segunda variável;

$H(3) = 11$, para uma solução criada movendo a terceira variável;

$H(7) = 1$, para uma solução criada movendo a sétima variável;

Os novos resultados para as três possíveis soluções são:

$$F(2) = 30 - 0,7 \cdot 7 = 25,1;$$

$$F(3) = 33 - 0,7 \cdot 11 = 25,3;$$

$$F(7) = 31 - 0,7 \cdot 1 = 30,3,$$

e, então, a terceira solução (isto é, movendo a sétima variável) é a que será selecionada.

A opção acima de inclusão de valores de frequência em uma medida de penalidade para avaliar soluções diversifica a busca. É claro, muitas outras opções podem ser consideradas em adição à BT (GLOVER e LAGUNA, 1997). Por exemplo, se é necessário selecionar um movimento tabu, pode-se usar uma regra adicional (então chamada “aspiração modificada”) para selecionar um movimento que é considerado o mais “antigo” dentre todos. Também pode ser uma boa idéia memorizar não somente o grupo de movimentos recentes, mas também se eles geraram ou não qualquer melhoria. Esta informação pode ser incorporada dentro das decisões de busca (então chamada “aspiração pela direção da busca”). Na seqüência, é interessante introduzir o conceito de influência, a qual mede o grau de mudança de uma nova solução qualquer em relação à distância entre a antiga e a nova solução, ou talvez a mudança na viabilidade da solução quando se trata de

um problema restrito. A intenção relacionada com influência é que um movimento particular tem uma grande influência se um passo “grande” correspondente foi executado da antiga solução em relação à nova. Esta informação pode também ser incorporada dentro da busca (então chamada “aspiração por influência”) (GLOVER e LAGUNA, 1997).

4.2 Algoritmos genéticos

Segundo SMITH *et al.* (1996), algoritmos genéticos (AG) têm sido recentemente utilizados quando se necessita de uma ferramenta computacional de busca de soluções para problemas complexos, como aprendizagem de máquinas, planejamento da movimentação de robôs, reconhecimento de padrões, detecção de imagem e muitas outras aplicações da engenharia. Eles oferecem um grupo um pouco diferente de comparações e contrastes com a BT e são baseados na seleção de subgrupos (usualmente pares) de soluções de uma população, chamados pais, e na combinação deles para a produção de novas soluções chamadas filhos. Regras de combinação para o desenvolvimento de filhos são baseadas na noção genética de cruzamento, que consiste no intercâmbio de valores de solução de variáveis particulares, junto com operações ocasionais como mudanças de valores aleatórios. Os filhos que passam no teste de sobrevivência, probabilisticamente influenciado para favorecer os de qualidade superior, são então disponibilizados para serem escolhidos como pais da geração seguinte. A escolha dos pais a serem emparelhados em cada geração é baseada na amostragem aleatória ou amostragem aleatória influenciada da população (em algumas versões paralelas executada em subpopulações separadas cujos melhores membros são periodicamente trocados ou compartilhados). Assim, o AG inicia com um grupo de soluções ao invés de uma solução única. A população consistiu de um grupo de cromossomos, que a partir da qual uma nova população foi desenvolvida por analogia dos processos específicos da genética fundamental: reprodução baseada em aptidão, cruzamento e mutação. A terminologia genética costumeiramente se refere a soluções como cromossomos, variáveis como genes e valores de variáveis como alelos. Por meio de convenções de código, os genes dos AGs podem ser comparados a atributos na busca tabu, ou mais precisamente a atributos na forma de memórias baseadas em frequência. A introdução de memória em AG para rastrear a história dos genes e seus alelos em subpopulações irão prover uma imediata e natural forma de criar um modelo híbrido com a BT.

4.2.1 Cruzamento

Segundo SMITH *et al.* (1996), o cruzamento é uma idéia simples: suponha que haja duas seqüências **a** e **b**, cada uma consistindo de 6 variáveis, por exemplo,

$$(a_1, a_2, a_3, a_4, a_5, a_6) \quad \text{e} \quad (b_1, b_2, b_3, b_4, b_5, b_6),$$

representando duas soluções para um problema. Um ponto de cruzamento é dado aleatoriamente a partir dos números de 1 a 5 e uma nova solução produzida pela combinação das variáveis dos “pais” originais. Por exemplo, se o ponto de cruzamento foi 2, então as soluções “descendentes” poderiam ser

$$(a_1, a_2, b_3, b_4, b_5, b_6) \quad \text{e} \quad (b_1, b_2, a_3, a_4, a_5, a_6).$$

Em muitas aplicações, o vetor componente, ou cromossomo, é simplesmente uma seqüência de zeros e uns. GOLDBERG (1989) sugere que há significativas vantagens se o cromossomo puder então ser estruturado, embora outros argumentos lancem alguma dúvida nisto. Contudo, grande parte do desenvolvimento teórico é fácil de ser compreendido se for analisado deste modo.

Continuando a analogia genética, variáveis são frequentemente chamadas de genes, os possíveis valores de uma variável são chamados de alelos, e a posição de uma variável em uma seqüência é chamada lócus. Em casos simples, o lócus de uma variável/gene é normalmente irrelevante, mas em problemas mais complexos ele torna-se importante. Uma distinção adicional é esboçada, em genética, entre o cromossomo (ou uma coleção de cromossomos) como genótipo, significando a atual estrutura (ou estruturas), e o fenótipo – a expressão física da estrutura (ou estruturas). Nos termos de um AG, pode-se interpretar o genótipo com a seqüência codificada que é processada pelo algoritmo, enquanto o grupo decodificado de parâmetros representa o fenótipo.

4.2.2 Mutação

Outro operador comumente usado é a mutação, o qual fornece a oportunidade de alcançar partes do espaço de busca que talvez não possam ser alcançadas pelo cruzamento somente. Cada gene de uma seqüência é examinado ao redor, e com uma pequena probabilidade seu alelo atual pode ser mudado. Por exemplo, a seqüência (011001) poderia tornar-se (010011) se o terceiro e quinto alelos forem mudados.

4.2.3 Reprodução

Cruzamento e mutação são as ferramentas básicas para a criação de novas soluções (reprodução). No entanto, a escolha de quais cromossomos serão selecionados como base para o passo reprodutivo (ou selecionados para a reprodução) é claramente crítica em relação ao que acontece com a população como um todo. HOLLAND (1992), que foi o responsável pelo desenvolvimento inicial dos conceitos de AG, sugere que ao menos um dos “pais” deve sempre ser escolhido pela sua “aptidão” – em termos de problemas combinatoriais. Isto implica em alguma função monótona no valor da função objetivo. Com isso, ele foi capaz de provar o importante Teorema do Esquema, o qual fornece uma maneira de entender como o AG trabalha. A idéia básica fundamental da tradicional teoria de AG será apresentada na seqüência.

Se forem adotadas duas seqüências como (011100) e (010011), pode-se notar que elas têm algo em comum, pois ambas têm um “0” como primeiro gene e um “1” como segundo. No jargão de AG, ambas são instâncias de um esquema (01****), onde o símbolo * é essencialmente usado como um caractere “não importa”. Esquematizado dessa maneira, elas representam similarmente subconjuntos de um hipercubo sobre o qual a busca está localizada. Neste caso o subconjunto consiste de todas as 16 (2^4) seqüências geradas pela substituição de valores particulares no lugar dos símbolos *. A ordem de um esquema é o número de genes definidos (0 ou 1), enquanto seu comprimento é a distância entre os genes externos definidos. No exemplo acima, a ordem é 2 e o comprimento é 1.

Usando desta maneira os agrupamentos de seqüências juntos, HOLLAND (1992) mostrou que, dado um número fixo de instâncias de um esquema s na população numa geração t , o número esperado de instâncias do esquema na geração $t+1$ poderia aumentar visto que sua razão de aptidão $f(s)$ (dada por sua aptidão dividida pela média da aptidão da população), onde a aptidão corresponde ao valor da função de avaliação, excederia:

$$f(s) = 1 + \frac{l(s)}{n-1} + P_m k(s) \quad \text{Eq. 4.3}$$

onde $l(s)$ é o comprimento e $k(s)$ a ordem de s , e P_m a probabilidade de mutação. O número atual de instâncias de s pode ser bem diferente do que o número esperado. Contudo, ele demonstra a base para uma explanação de como os AGs operam.

4.2.4 Um exemplo simples

Para apresentar a ação do AG de forma mais transparente, será usado o problema de maximização da função:

$$f(x) = x^3 - 60x^2 + 900x + 100 \quad \text{Eq. 4.4}$$

Suponha que sejam geradas cinco seqüências aleatórias com os resultados mostrados na tabela 4.5.

Tabela 4.5 - Cinco seqüências aleatórias (SMITH *et al.*, 1996)

NÚM.	SEQUÊNCIA	x	$f(x)$	$P[\textit{selecionado}]$
1	10011	19	2399	0.206
2	00101	5	3225	0.277
3	11010	26	516	0.044
4	10101	21	1801	0.155
5	01110	14	3684	0.317
MÉDIA DE APTIDÃO		2325		

Tratando os valores de $f(x)$ simplesmente como valores de aptidão (ou seja, da função de avaliação), a probabilidade de seleção de cada seqüência como “Pai 1” é diretamente proporcional ao seu valor, e é dado na coluna $P[\textit{selecionado}]$. Pela distribuição aleatória uniforme dos números entre 0 e 1, foi selecionado o “Pai 1” e, através da distribuição discreta uniforme, selecionou-se o “Pai 2”. Novamente, através da distribuição aleatória uniforme dos números, procedeu-se o processo de cruzamento (com $P_c = 1$) até um determinado local randômico, seguido pelo processo de mutação (com $P_m = 0.02$ em cada local). Os resultados de um experimento típico são apresentados na tabela 4.6.

Tabela 4.6 - Um típico experimento (SMITH *et al.*, 1996).

PASSO	PAI 1	PAI 2	PONTO DE CRUZAMENTO	MUTAÇÃO ?	DESCENDÊNCIA SEQUÊNCIA	
1	1	2	4	NNNYN	10001	2973
2	5	3	2	NNNNN	01010	4100
3	5	2	3	NNNNN	01101	3857
4	4	2	1	NYNNN	11101	129
5	2	5	4	NNNNN	00100	2804
MÉDIA DE APTIDÃO					2773	

A busca iria continuar pelo recálculo das probabilidades de seleção e prosseguiria com um novo ciclo de seleção, cruzamento e mutação. Este é um exemplo muito simples do procedimento de AG, mas foi substancialmente efetivo na elevação da média da população, encontrando a alternativa ótima no passo 2.

4.3 *Simulated annealing*

Segundo REEVES (2005), o *simulated annealing* (SA) é a técnica que primeiramente popularizou-se na década passada, e desde então têm se mostrado um método efetivo de busca para um grande número de problemas. Ela executa uma busca dentro do grupo de todas as possíveis soluções, reduzindo a chance de prender-se em um ponto de ótimo local “pobre” pela possibilidade de movimentos para soluções inferiores sob o controle de um esquema randomizado. Especificamente, se um movimento a partir de uma solução \mathbf{x} para outra vizinhança, porém de solução \mathbf{x}' inferior (ou pior que \mathbf{x}), resulta em uma mudança no valor da função de avaliação de Δc , tal que Δc tem valor positivo, onde

$$\Delta c = f(\mathbf{x}') - f(\mathbf{x}) \quad \text{Eq. 4.5}$$

o movimento para \mathbf{x}' é aceito, ainda que a solução seja inferior à anterior, se

$$R < \exp(-\Delta c/T) \quad \text{Eq. 4.6}$$

onde T é um parâmetro de controle, e $R \in [0,1]$ é um número aleatório. Sempre que a função de avaliação $f(\mathbf{x}')$ da nova solução \mathbf{x}' é melhor ou igual à função de avaliação $f(\mathbf{x})$ da solução anterior \mathbf{x} , tal que Δc tem valor menor ou igual a zero, a nova solução \mathbf{x}' será aceita como a nova solução vigente. Há ainda o Critério de Metrópolis, que permite a aceitação de uma solução pior que a solução vigente para fugir de ótimos locais. O parâmetro T é inicialmente alto, aumentando a chance de aceitação de soluções inferiores, e é vagarosamente reduzido até atingir um valor onde movimentos que geram soluções inferiores são quase sempre rejeitados. Há uma similar analogia entre este método e o processo termodinâmico de têmpera ou recozimento de materiais; e foi esta analogia que originalmente motivou o desenvolvimento do método.

As idéias que formam a base do SA foram primeiramente publicadas em 1953 em um algoritmo para simular o resfriamento de material em um banho líquido quente – um processo conhecido como têmpera. Se um material sólido é aquecido até o seu ponto de derretimento e então resfriado voltando ao seu estado sólido, as propriedades estruturais do

sólido resfriado dependem da taxa de resfriamento. O processo de têmpera pode ser simulado pela consideração do material como um sistema de partículas.

Como exemplo, seja considerada a maximização de:

$$f(x) = x^3 - 60x^2 + 900x + 100 \quad \text{Eq. 4.7}$$

onde x é codificado em um número binário inteiro de 5 bits no intervalo $[0, 31]$.

Para ilustrar o SA (Tabela 4.7), a seqüência inicial será (1 0 0 1 1), código binário para $x = 19$, e $f(x)$ será igual a 2399. Uma das soluções vizinhas (ou os *bits* para mudança) de x é escolhida aleatoriamente, como mostrado nesta tabela.

Tabela 4.7 – Geração de novas soluções partindo de $T=100$ (SMITH *et al*, 1996).

T	BIT	SEQUÊNCIA	x	$f(x)$	Δc	R	$e^{-\Delta c/T}$	MOVE R	NOVA SEQUÊNCIA
100	1	0 0 0 1 1	3	2287	112	0,65879	326×10^{-3}	N	1 0 0 1 1
90	3	1 0 1 1 1	23	1227	1172	0,87456	$2,2 \times 10^{-6}$	N	1 0 0 1 1
81	5	1 0 0 1 0	18	2692	< 0	0,32658	1	S	1 0 0 1 0
72.9	2	1 1 0 1 0	26	516	2176	0,44879	$1,09 \times 10^{-13}$	N	1 0 0 1 0
65.6	4	1 0 0 0 0	16	3236	< 0	0,22589	1	S	1 0 0 0 0
59.0	3	1 0 1 0 0	20	2100	1136	0,15879	$4,35 \times 10^{-9}$	N	1 0 0 0 0

Devido ao parâmetro T ser menor que o valor de Δc , no exemplo da tabela 4.3, a chance de aceitação de uma solução inferior (à anterior) diminuiu a medida que a razão entre eles ($\Delta c / T$) aumentou, como indicado na coluna $e^{-\Delta c/T}$. Note que o valor de R é definido aleatoriamente (tal que $R \in [0,1]$). Na tabela 4.7, é improvável que T seja alto o suficiente para que a busca saia de um ponto de ótimo local e, independente da taxa de variação de T pelas próximas 50 tentativas, a chance de alteração da nova seqüência (indicando a saída deste ponto de ótimo local) será muito baixa.

Na segunda tentativa (tabela 4.8) a temperatura inicial foi alta o suficiente para a busca escapar da região de atração do valor de $f = 3236$ (seqüência: 10000 da tabela 4.7) e o ponto ótimo global foi encontrado. Note, porém, que a busca escapou do local novamente, mas nas 150 tentativas seguintes a maioria dos “estados” visitados foram o ótimo ou um dos seus vizinhos imediatos. Outra observação importante é que o ponto de ótimo global era conhecido neste problema, mas o mais comum é que este ponto não seja conhecido. Assim, ainda que o ponto de ótimo global seja “visitado”, o SA não será capaz de identificá-lo e a

busca prosseguirá. É por este motivo que é tão importante dotar o métodos heurísticos de algum mecanismo para a fuga de pontos de ótimo local.

Tabela 4.8 – 2ª tentativa de geração de soluções partindo de $T=500$ (SMITH *et al*, 1996).

T	BIT	SEQUÊNCIA	x	$f(x)$	Δc	R	$e^{-\Delta c/T}$	MOVER	NOVA SEQUÊNCIA
500	1	0 0 0 1 1	3	2287	112	0,58979	799×10^{-3}	S	0 0 0 1 1
450	3	0 0 1 1 1	7	3803	< 0	0,88559	1	S	0 0 1 1 1
405	5	0 0 1 1 0	6	3556	247	0,21359	543×10^{-3}	S	0 0 1 1 0
364.5	2	0 1 1 1 0	14	3684	<0	0,69874	1	S	0 1 1 1 0
328.0	4	0 1 1 0 0	12	3988	<0	0,95784	1	S	0 1 1 0 0
295.2	3	0 1 0 0 0	8	3972	16	0,11236	947×10^{-3}	S	0 1 0 0 0
265.7	4	0 1 0 1 0	10	4100	<0	0,65589	1	S	0 1 0 1 0
239.1	5	0 1 0 1 1	11	4071	29	0,33259	885×10^{-3}	S	0 1 0 1 1
215.2	1	1 1 0 1 1	27	343	3728	0,26598	3×10^{-8}	N	0 1 0 1 1

A escolha da taxa de resfriamento e da temperatura inicial no exemplo apresentado foi feita arbitrariamente. A seguir serão definidas com detalhes as decisões genéricas necessárias na implementação do SA na prática.

a) Temperatura Inicial

Segundo REEVES (1995), o processo deve iniciar de modo que a maioria, senão todos os movimentos possam ser aceitos, por isso, a temperatura inicial deve ser “alta”. Na prática, isso pode exigir o conhecimento da magnitude das soluções da vizinhança; na falta de tal conhecimento, uma forma de determinação de um valor adequado é escolher um valor grande e aplicar o algoritmo por um curto espaço de tempo, observando a taxa resultante. Se a taxa for “apropriadamente alta”, este valor de T pode ser usado para iniciar o processo. A taxa “apropriadamente alta” aceitável poderá variar de uma situação para outra, mas em muitos casos uma taxa de aceitação entre 40% e 60% pode dar bons resultados. Métodos mais sofisticados são possíveis, porém frequentemente desnecessários. Apesar de a analogia com o processo de têmpera ou recozimento ser utilizada, o parâmetro temperatura T não tem nenhum significado físico no SA, devendo ser visto como um parâmetro adimensional.

b) Taxa de Resfriamento

Talvez o mais importante fator em aplicações práticas é a taxa de resfriamento. Aqui deve-se notar que há basicamente dois tipos de taxa, por analogia às cadeias *Markov* homogêneas e não homogêneas, respectivamente (REEVES, 1995) No caso homogêneo, a temperatura é conduzida a uma temperatura fixa até que o “equilíbrio” é alcançado. Uma vez julgado que este estado tenha sido atingido, a temperatura é reduzida, e o procedimento repetido. O número de tentativas de movimento a cada temperatura pode ser bastante grande, embora os intervalos de temperatura possam ser grandes também. No caso não homogêneo, a temperatura é reduzida (mas por uma taxa constante mínima) depois de cada movimento. Este caso é menos complicado que o caso homogêneo, e é o mais comumente usado na prática. Nos dois casos, o mais usual é tomar uma taxa de resfriamento constante α , onde $0 < \alpha < 1$; mas, normalmente, $0.9 \leq \alpha < 1$. Considerando-se T_k a temperatura na iteração atual e T_{k+1} a temperatura na próxima iteração, tem-se:

$$T_{k+1} = \alpha T_k \quad \text{Eq. 4.8}$$

c) Temperatura Final

Em teoria o procedimento deve ser continuado até que a temperatura final T_f seja próxima de zero, mas na prática é necessário que se prossiga até que a chance de aceitação de um movimento para cima torne-se desprezível.

d) Número de Iterações

Deve-se ter em mente que o número de iterações é efetivamente fixado pelas três escolhas acima descritas.

5 METODOLOGIA

Como foi apresentado anteriormente, neste trabalho é abordado o problema de planejamento de trajetória otimizado de um robô industrial de seis eixos, que é parte integrante do conjunto formado por uma “mesa posicionadora” de um eixo, sendo aplicados a um processo de soldagem do eixo traseiro de um automóvel. O posicionador ou mesa posicionadora de um eixo (de um grau de liberdade) é usado para ajustar a orientação necessária e mais favorável à execução de cordões de solda através do processo *MIG/MAG*, enquanto que o robô (de seis graus de liberdade) é responsável por guiar a tocha de soldagem durante o processo. Porém, o processo de posicionamento de ambos (robô e posicionador) normalmente é realizado através do processo conhecido como ponto-a-ponto ou *leadthrough*, no qual o programador precisa conduzir o robô através de cada ponto da trajetória a ser executada, sendo que cada ponto deve ser gravado na memória do robô. Esse método apresenta a desvantagem de obrigar a interrupção do processo de produção durante a programação; além de tornar o sistema de programação de um robô de soldagem complexo e demorado, principalmente na execução de peças com vários cordões a serem executados. Além disso, no processo ponto-a-ponto, cabe ao programador do robô a definição da posição e da orientação da tocha de soldagem e, também, da orientação do posicionador. Assim, uma definição inadequada da posição e das orientações implicará num maior tempo de processamento da operação de solda.

Visando apresentar um método mais eficiente de programação para o robô em questão, a proposta do trabalho é a aplicação do método da Busca Tabu para a seleção da melhor alternativa de trajetória para o robô durante o processo de soldagem. Isto é feito através da avaliação da melhor combinação de valores de ângulos de junta para a pose de trabalho. É importante ressaltar que, neste momento, o foco principal do trabalho é a determinação da melhor trajetória para o robô, não sendo contemplado o ajuste dos parâmetros de soldagem do processo.

Para a abordagem deste problema, será utilizado o modelo cinemático correspondente ao robô *Mitsubishi* modelo RV-M1 de cinco graus de liberdade, acrescido das considerações para a inclusão do sexto eixo. A transformação homogênea que descreve a posição e orientação do “punho” do robô citado é conhecida, ou seja, sabe-se o valor numérico de cada elemento da matriz.

A matriz de valores associados à transformação homogênea, indicada por ${}^0\tilde{T}_6$, é representada pela equação 5.1:

$${}^0\tilde{T}_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 5.1}$$

que deve ser igual à transformação homogênea 0T_6 que, por sua vez, é função das variáveis de junta mostradas na equação 5.1, ou seja,

$${}^0\tilde{T}_6 = {}^0T_6 \quad \text{Eq. 5.2}$$

Esta equação matricial fornece um sistema de equações com doze equações escalares que pode ser difícil de ser resolvido. Usando o artifício de procurar alguma simetria que surge da limitação de uma das juntas, pode-se observar que, neste caso, se a primeira junta for imobilizada, o punho só pode atingir pontos sobre um plano (muito embora a ferramenta possa atingir pontos fora deste plano). Este plano é chamado de plano do punho ou plano do robô. A limitação a um plano implica na perda de um grau de liberdade de posicionamento do robô. Assim, é de se esperar que a transformação homogênea que descreve a posição e orientação do punho em relação ao segmento 1 (1T_6) tenha um valor constante na última coluna (que representa a posição do punho). Esta constante pode ser usada para construir uma equação mais simples, em função de apenas uma variável de junta, conforme indicado na equação 5.3. Trabalhando no “plano do robô”, pode-se igualar a equação 5.3 à equação 5.4, obtendo-se a equação 5.5.

$${}^1\tilde{T}_6 = ({}^0T_1^{-1}) {}^0\tilde{T}_6 = \begin{bmatrix} r_{11}c_1 + r_{21}s_1 & r_{12}c_1 + r_{22}s_1 & r_{13}c_1 + r_{23}s_1 & p_x c_1 + p_y s_1 \\ r_{21}c_1 - r_{11}s_1 & r_{22}c_1 - r_{12}s_1 & r_{23}c_1 - r_{13}s_1 & p_y c_1 - p_x s_1 \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 5.3}$$

$${}^1T_6 = {}^1T_2 {}^2T_1 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad \text{Eq. 5.4}$$

$${}^1T_6 = \begin{bmatrix} c_5c_6c_{234} - s_6s_{234} & -c_5s_6c_{234} - c_6s_{234} & -s_5c_{234} & a_2c_2 + a_3c_{23} + d_5s_{234} \\ -c_6s_5 & s_6s_5 & -c_5 & 0 \\ c_5c_6s_{234} + s_6c_{234} & -c_5s_6s_{234} + c_6c_{234} & -s_5s_{234} & a_2s_2 + a_3s_{23} - d_5c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 5.5}$$

Comparando o elemento na segunda linha e na quarta coluna das duas matrizes nas equações 5.3 e 5.5 têm-se:

$$\begin{aligned} p_y \cos(\theta_1) - p_x \operatorname{sen}(\theta_1) &= 0 \\ p_y \cos(\theta_1) &= p_x \operatorname{sen}(\theta_1) \\ \frac{p_y}{p_x} &= \frac{\operatorname{sen}(\theta_1)}{\cos(\theta_1)} = \tan(\theta_1) \end{aligned}$$

Logo:

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right) \quad \text{Eq. 5.6}$$

Note que p_x , p_y e p_z representam as coordenadas cartesianas do ponto central do punho do robô, que por sua vez coincidem com o ponto central da ferramenta (TCP) presa ao punho ou, neste caso, a tocha de soldagem.

As matrizes apresentadas nas equações 5.3 e 5.5 permitem a obtenção das seguintes equações para o cálculo dos ângulos das juntas do robô (LAUS, 2006), indicados a seguir. As equações foram ordenadas na seqüência correta da definição dos ângulos.

Conhecendo-se o valor de θ_1 é possível a comparação do valor dos elementos (2,3) nas equações 5.4 e 5.5:

$$\begin{aligned} \cos(\theta_5) &= r_{13} \operatorname{sen}(\theta_1) - r_{23} \cos(\theta_1) \\ \theta_5 &= \arccos\left(r_{13} \operatorname{sen}(\theta_1) - r_{23} \cos(\theta_1)\right) \end{aligned} \quad \text{Eq. 5.7}$$

De forma análoga, conhecendo-se os valores de θ_1 e θ_5 é possível a comparação do valor dos elementos na primeira e segunda colunas da segunda linha de 1_6T e ${}^1_6\tilde{T}$ para se obter as expressões:

$$\begin{aligned} \operatorname{sen}(\theta_5) \cos(\theta_6) &= r_{11} \operatorname{sen}(\theta_1) - r_{21} \cos(\theta_1) \\ \operatorname{sen}(\theta_5) \operatorname{sen}(\theta_6) &= r_{22} \cos(\theta_1) - r_{12} \operatorname{sen}(\theta_1) \end{aligned} \quad \text{Eq. 5.8}$$

de onde pode-se concluir que, se $\theta_5 \neq 0^\circ$ ou se $\theta_5 \neq 180^\circ$:

$$\theta_6 = \arctan\left(\frac{r_{22} \cos(\theta_1) - r_{12} \text{sen}(\theta_1)}{r_{11} \text{sen}(\theta_1) - r_{21} \cos(\theta_1)}\right) \quad \text{Eq. 5.9}$$

Fazendo-se a comparação do valor dos elementos na primeira e terceira linhas da terceira coluna de ${}^1_6\mathcal{T}$ e ${}^1_6\tilde{\mathcal{T}}$ obtém-se o sistema de equações:

$$\begin{cases} \text{sen}(\theta_5) \cos(\theta_{234}) = -r_{13} \cos(\theta_1) - r_{23} \text{sen}(\theta_1) \\ \text{sen}(\theta_5) \text{sen}(\theta_{234}) = -r_{33} \end{cases} \quad \text{Eq. 5.10}$$

de onde se conclui que, se $\theta_5 \neq 0^\circ$ ou se $\theta_5 \neq 180^\circ$:

$$\theta_2 + \theta_3 + \theta_4 = \arctan\left(\frac{r_{33}}{r_{13} \cos(\theta_1) + r_{23} \text{sen}(\theta_1)}\right) \quad \text{Eq. 5.11}$$

A partir de valores arbitrados para p_x , p_y e p_z , que representam as coordenadas cartesianas da ferramenta de trabalho presa ao punho do robô (neste caso, a tocha de soldagem), o objetivo é calcular os valores dos ângulos das juntas para essas coordenadas. A partir destes valores, é possível a aplicação de algoritmos de otimização para a escolha da melhor combinação de valores dos ângulos das juntas, fornecido pela função de avaliação (C_s), indicada na equação 5.12:

$$C_s = \sum_j \sum_{k=1}^{n-1} |\theta_{j,k+1} - \theta_{jk}| \quad \text{Eq. 5.12}$$

onde θ_{jk} representa o ângulo da junta j quando o robô ocupa a posição k da trajetória de solda.

Neste trabalho é abordado o problema de planejamento de trajetória otimizado de um robô industrial de seis eixos, que é parte integrante do conjunto formado por uma “mesa posicionadora” de um eixo (ou seja, com um grau de liberdade), sendo aplicados a um processo de soldagem do eixo traseiro de um automóvel. O posicionador ou mesa posicionadora é usado para ajustar a orientação necessária e mais favorável à execução de cordões de solda através do processo *MIG/MAG*, enquanto que o robô (de seis graus de liberdade) é responsável por guiar a tocha de soldagem durante o processo. Serão utilizadas duas modalidades de trajetória: linear e circular.

5.1 Trajetória linear

Para a determinação dos parâmetros necessários à implementação do algoritmo para a geração de trajetória do cordão linear, foi utilizado o diagrama mostrado na figura 5.1.

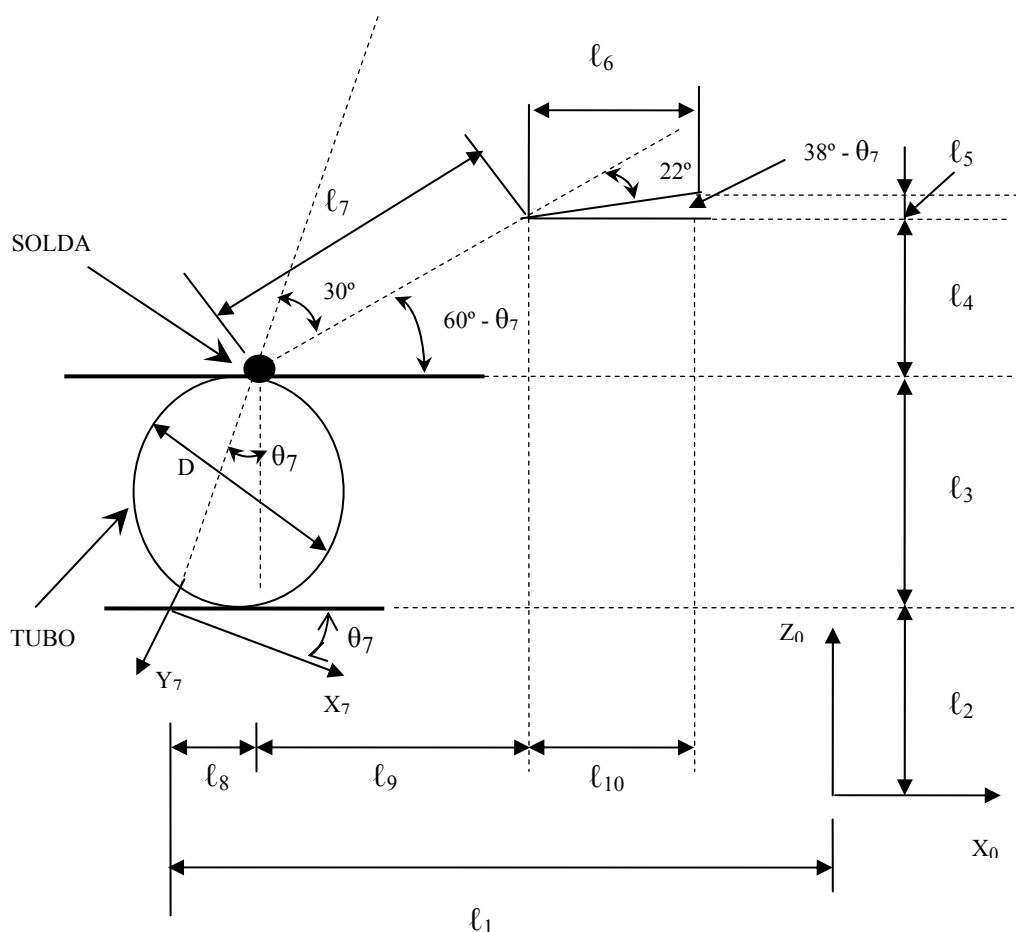


Figura 5.1. Representação do cordão linear.

No diagrama apresentado, o sistema de coordenadas, também conhecido como *frame7*³, (representado por \hat{x}_7 e \hat{y}_7) representa o grau de liberdade da mesa posicionadora (rotação). Os valores representados pelas letras D e l são parâmetros adotados para este estudo em particular. Portanto, a metodologia aplicada é válida para qualquer outro conjunto de parâmetros existentes. Neste caso, D representa o diâmetro do tubo onde será aplicado

³ A palavra *frame* indica um sistema de coordenadas que é associado a cada um dos graus de liberdade do problema estudado.

o cordão de solda e os valores de l são as distâncias resultantes da decomposição dos ângulos que incidem entre a tocha de soldagem e a peça. A seguir são apresentados os valores dos parâmetros, indicados na figura 5.2, que foram usados no problema estudado.

D	$= 150 \text{ mm};$
l_1	$= -200 \text{ mm};$
l_2	$= 150 \text{ mm};$
l_3	$= 150 \cos \theta_7$
l_4	$= 260 \text{ sen } (60^\circ - \theta_7);$
l_5	$= 100 \text{ sen } (38^\circ - \theta_7);$
l_6	$= 100 \text{ mm};$
l_7	$= 260 \text{ mm};$
l_8	$= 150 \text{ sen } \theta_7;$
l_9	$= 260 \cos (60^\circ - \theta_7);$
l_{10}	$= 100 \cos (38^\circ - \theta_7).$

Figura 5.2. Valores adotados no presente estudo.

A partir da análise dos parâmetros do diagrama acima, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (*TCP*). Neste caso, a matriz representa o TCP (tocha de soldagem), que está acoplada à sexta junta do robô, em relação à pose de referência mostrada na figura 5.1, em substituição à equação 5.1.

$${}^0_6 \tilde{T} = \begin{bmatrix} -\text{sen } (38^\circ - \theta_7) & \cos (38^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \cos (38^\circ - \theta_7) & \text{sen } (38^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.13)}$$

onde:

$$p_x = 150 \text{ sen } \theta_7 + 260 \cos (60^\circ - \theta_7) + 100 \cos (38^\circ - \theta_7) - 200;$$

p_y é uma variável independente;

$$p_z = 150 + 150 \cos \theta_7 + 260 \text{ sen } (60^\circ - \theta_7) + 100 \text{ sen } (38^\circ - \theta_7).$$

5.2 Trajetória circular

Similarmente ao processo adotado para a trajetória linear, na determinação dos parâmetros necessários à implementação do algoritmo para a geração de trajetória do cordão circular, foram gerados diagramas para cada ponto pertencente à trajetória, os quais são mostrados a seguir.

Ponto 1:

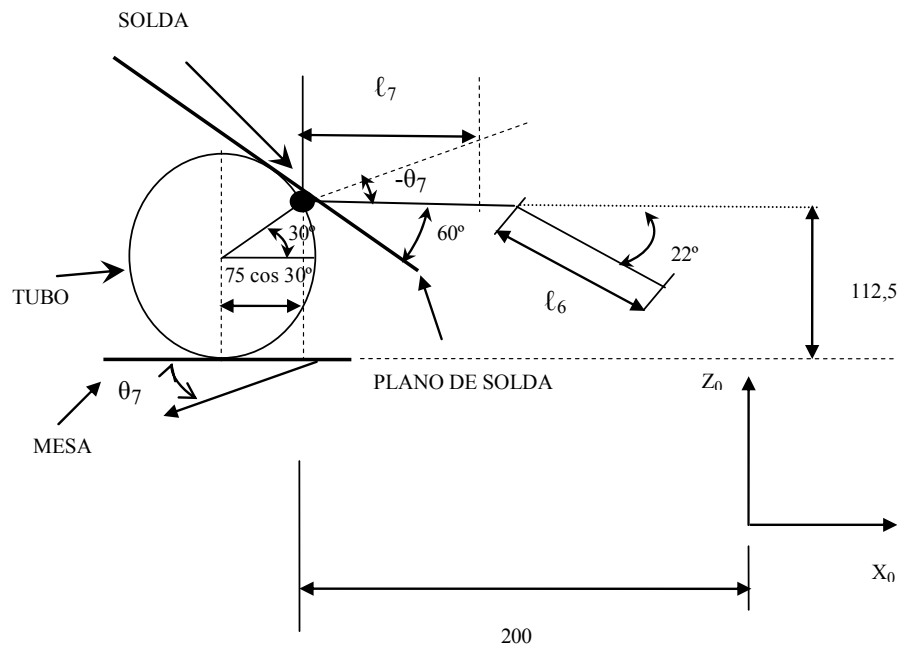


Figura 5.3: Diagrama ponto 1 da trajetória circular.

A partir da análise dos parâmetros do diagrama da figura 5.3, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (TCP). Neste caso, a matriz representa o TCP (tocha de soldagem), que está acoplada à sexta junta do robô, em relação à pose de referência mostrada na figura 5.3, em substituição à equação 5.1.

$${}^0_6\tilde{T} = \begin{bmatrix} -\sin(-22^\circ - \theta_7) & \cos(-22^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \cos(-22^\circ - \theta_7) & \sin(-22^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.14)}$$

onde:

$$p_x = 112,5 \sin \theta_7 + 200 \cos(\theta_7) + 100 \cos(22^\circ + \theta_7) - 200 + 75 \cos(30^\circ);$$

p_y é uma variável independente;

$$p_z = 150 + 112,5 \cos \theta_7 - 260 \sin \theta_7 - 100 \sin(22^\circ + \theta_7).$$

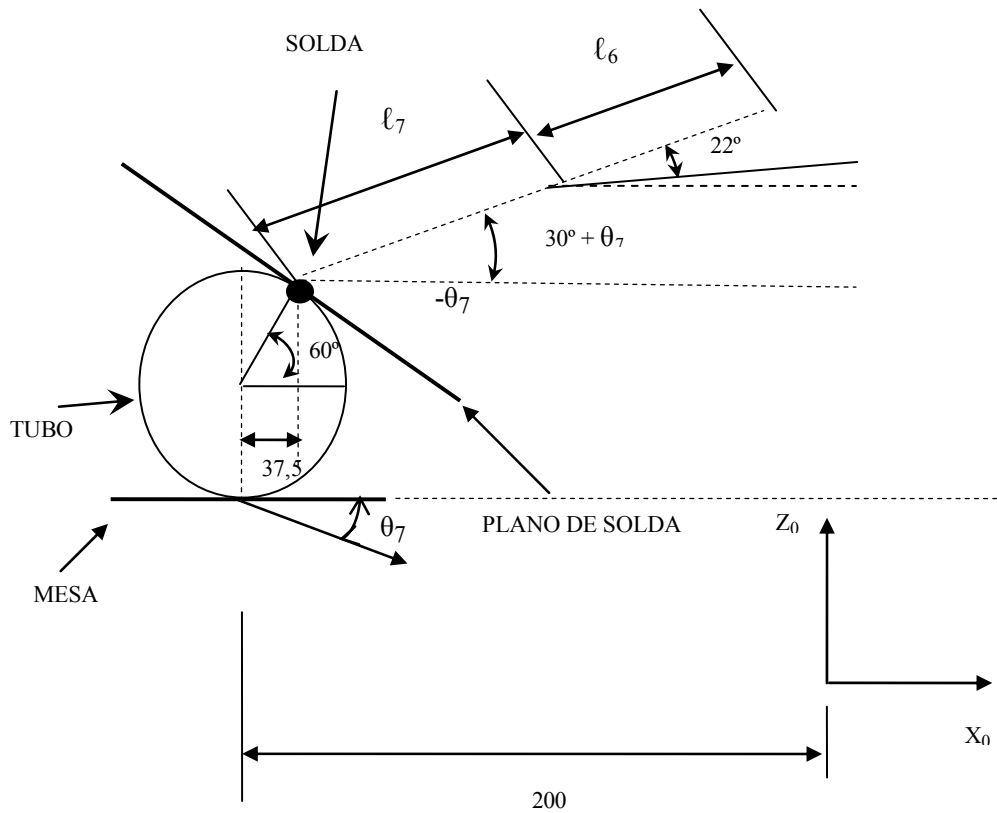
Ponto 2:

Figura 5.4. Diagrama ponto 2 da trajetória circular.

A partir da análise dos parâmetros do diagrama da figura 5.4, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (*TCP*). Neste caso, a matriz representa o *TCP* (tocha de soldagem), que está acoplada à sexta junta do robô, em relação à pose de referência mostrada na figura 5.4, em substituição à equação 5.1.

$${}^0_6\tilde{T} = \begin{bmatrix} -\text{sen}(8^\circ - \theta_7) & \text{cos}(8^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \text{cos}(8^\circ - \theta_7) & \text{sen}(8^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.15)}$$

onde:

$$p_x = 75 (1 + \text{cos } 60^\circ) \text{sen } \theta_7 + 260 \text{cos } (30^\circ + \theta_7) + 100 \text{cos } (8^\circ + \theta_7) - 200 + 37,5;$$

p_y é uma variável independente;

$$p_z = 150 + 75 (1 + \text{sen } 60^\circ) \text{cos } \theta_7 + 260 \text{sen } (30^\circ + \theta_7) + 100 \text{sen } (8^\circ + \theta_7).$$

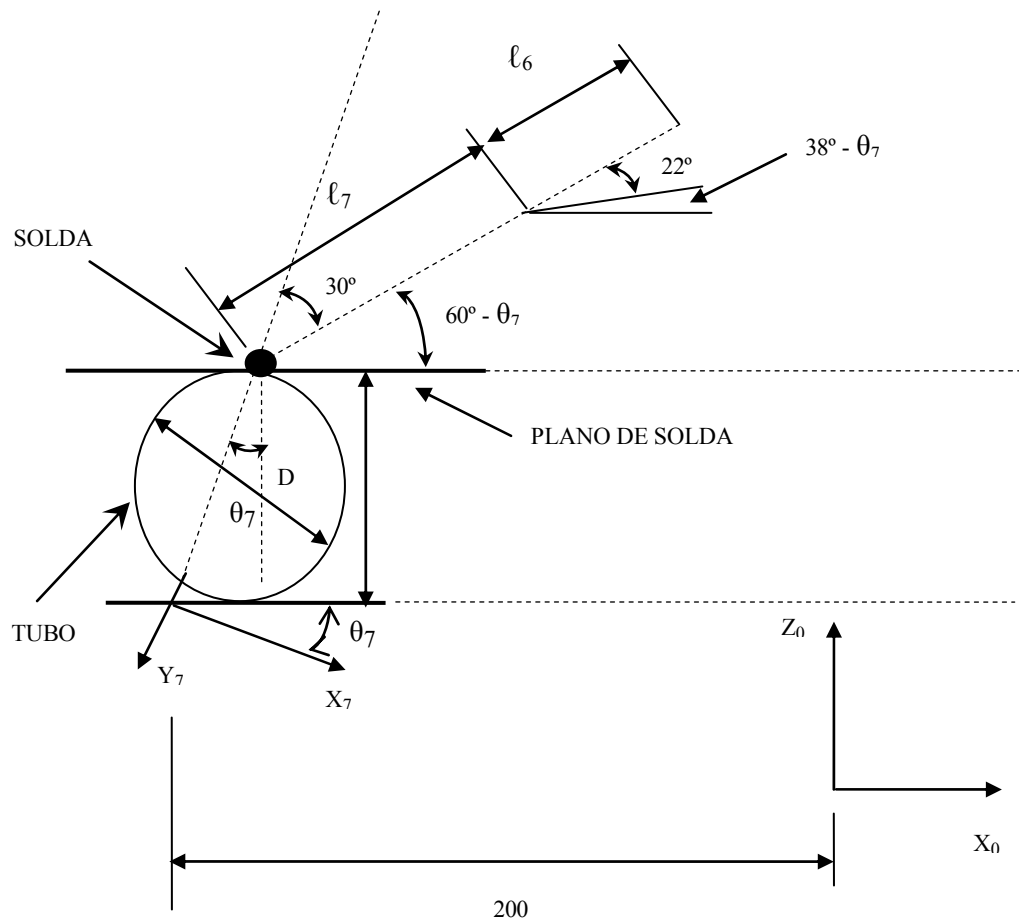
Ponto 3:

Figura 5.5. Diagrama ponto 3 da trajetória circular.

A partir da análise dos parâmetros do diagrama da figura 5.5, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (*TCP*). Neste caso, a matriz representa o *TCP* (tocha de soldagem), que está acoplada à sexta junta do robô, em relação à pose de referência mostrada na Figura 5.5, em substituição à equação 5.1.

$${}^0\tilde{T}_6 = \begin{bmatrix} -\text{sen}(38^\circ - \theta_7) & \text{cos}(38^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \text{cos}(38^\circ - \theta_7) & \text{sen}(38^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.16)}$$

onde:

$$p_x = 150 \text{ sen } \theta_7 + 260 \text{ cos } (60^\circ - \theta_7) + 100 \text{ cos } (38^\circ - \theta_7) - 200;$$

p_y é uma variável independente;

$$p_z = 150 + 150 \text{ cos } \theta_7 + 260 \text{ sen } (60^\circ - \theta_7) + 100 \text{ sen } (38^\circ - \theta_7).$$

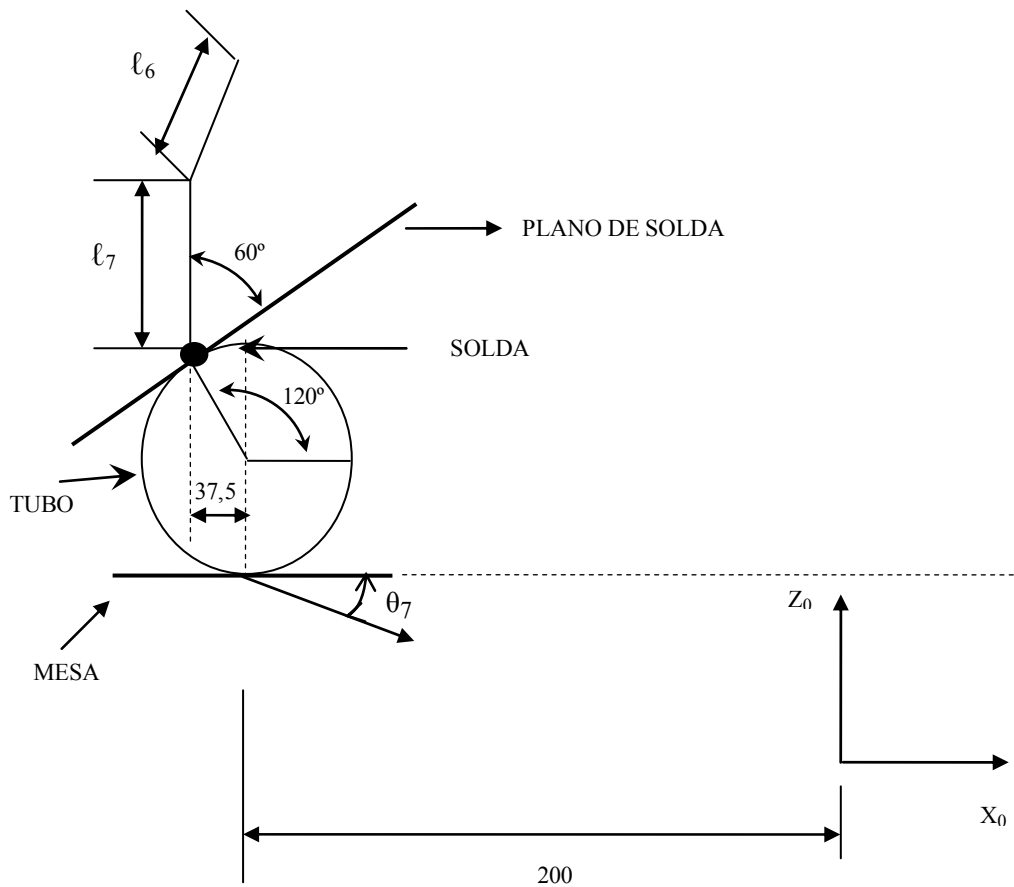
Ponto 4:

Figura 5.6. Diagrama ponto 4 da trajetória circular.

A partir da análise dos parâmetros do diagrama da figura 5.6, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (*TCP*). Neste caso, a matriz representa o *TCP* (tocha de soldagem), que está acoplada à sexta junta do robô, em relação à pose de referência mostrada na Figura 5.6, em substituição à equação 5.1.

$${}^0\tilde{T} = \begin{bmatrix} -\sin(68^\circ - \theta_7) & \cos(68^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \cos(68^\circ - \theta_7) & \sin(68^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.17)}$$

onde:

$$p_x = -75 (1 + \cos 60^\circ) \sin \theta_7 - 260 \sin \theta_7 + 100 \sin (22^\circ - \theta_7) - 237,5;$$

p_y é uma variável independente;

$$p_z = 150 + 75 (1 + \sin 60^\circ) \cos \theta_7 + 260 \cos \theta_7 + 100 \cos (22^\circ - \theta_7).$$

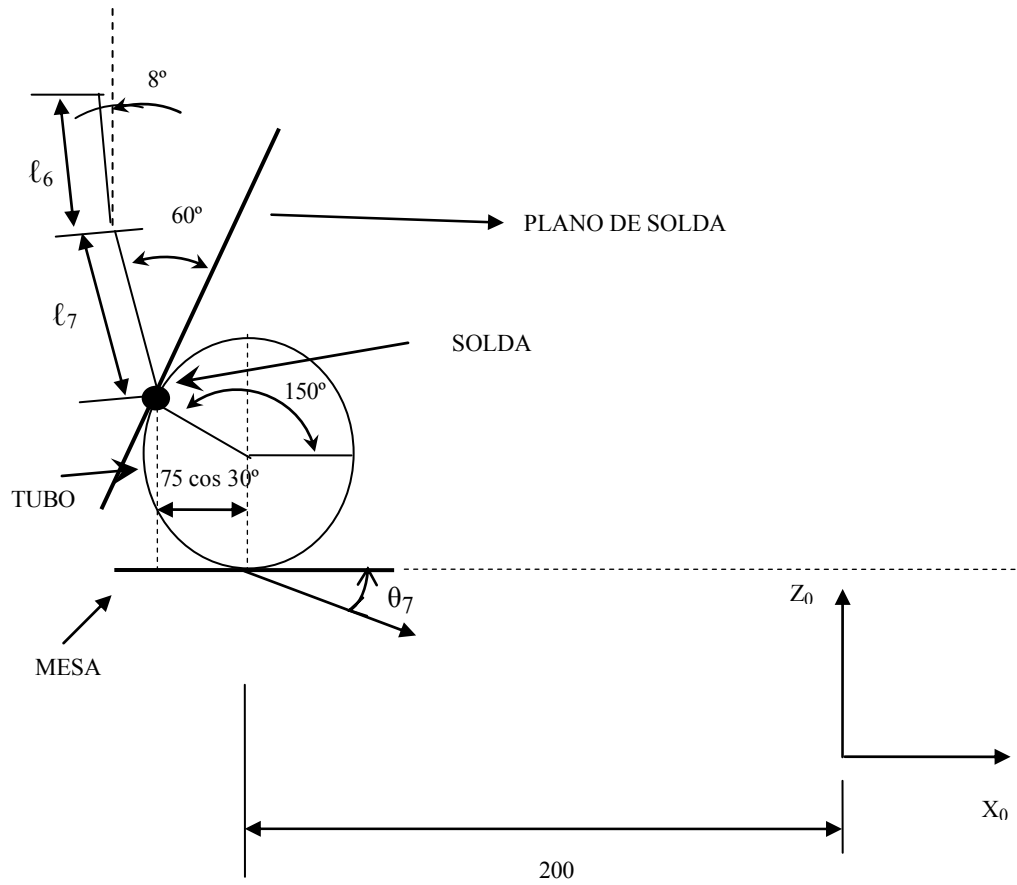
Ponto 5:

Figura 5.7. Diagrama ponto 5 da trajetória circular.

A partir da análise dos parâmetros do diagrama da figura 5.7, resulta uma matriz que representa a transformação homogênea do ponto central da ferramenta (*TCP*) que, neste caso, é representada pela tocha de soldagem, em relação à pose de referência mostrada na figura 5.7, em substituição à equação 5.1.

$${}^0_6\tilde{T} = \begin{bmatrix} -\sin(98^\circ - \theta_7) & \cos(98^\circ - \theta_7) & 0 & p_x \\ 0 & 0 & 1 & p_y \\ \cos(98^\circ - \theta_7) & \sin(98^\circ - \theta_7) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. (5.18)}$$

onde:

$$p_x = -112,5 \sin \theta_7 - 260 \sin(30^\circ + \theta_7) - 100 \sin(8^\circ + \theta_7) - 200 - 75 \cos(30^\circ);$$

p_y é uma variável independente;

$$p_z = 150 + 75(1 + \sin 30^\circ) \cos \theta_7 + 260 \cos(30^\circ + \theta_7) + 100 \cos(8^\circ + \theta_7).$$

5.3 Geração de trajetória usando algoritmo do *simulated annealing*

Para a obtenção dos ângulos das juntas do robô e da mesa posicionadora, foi implementado uma forma básica do algoritmo do *Simulated Annealing* (SA), conforme está descrito abaixo. Optou-se por implementar o SA para se dispor de um algoritmo para comparar com a implementação da Busca Tabu. O SA executa uma busca dentro do grupo de todas as possíveis soluções, buscando reduzir a chance de prender-se em um ponto de ótimo local “pobre”, pela possibilidade de movimentos sob o controle de um esquema estocástico de aceitação de soluções inferiores.

1	Gera solução inicial (s)
2	Faz melhor solução (s^*) = solução inicial (s)
3	Enquanto o critério de parada não é atingido:
4	Sorteia o incremento a ser usado nas soluções candidatas
5	Identifica a melhor solução candidata (s') vizinha da solução vigente (s)
6	Faz nova solução vigente (s) = melhor solução candidata (s')
7	Realiza busca local na nova solução vigente, checando se alguma atende ao Critério de Metrópolis.
8	Se melhor solução (s^*) > Nova solução vigente (s), então:
9	Faz melhor solução (s^*) = Nova solução vigente (s)
10	Retorna à linha 3
11	Escreve melhor solução (s^*)

Figura 5.8. Algoritmo de *simulated annealing* (SA) usado neste trabalho.

Inicialmente, a cada nova iteração, são sorteados, aleatoriamente, a variável a ser alterada e o incremento a ser utilizado. Para isso, foi implementado no código uma regra para que o incremento não fosse nulo, a fim de evitar-se uma falha na busca da melhor solução. Note que neste problema há variáveis independentes (p_y , θ_3 , θ_4 e θ_7) e dependentes (p_x , θ_1 , θ_2 , θ_5 e θ_6), e, portanto, apenas as variáveis independentes poderão ser sorteadas. As equações 5.19 a 5.22 são rerepresentadas (vide equações 5.6, 5.7, 5.9 e 5.11) como memorial do cálculo das variáveis dependentes. O cálculo de p_x é apresentado nas seções 5.1 (para trajetória linear) e 5.2 (para trajetória circular).

No cálculo das variáveis dependentes, se houver alteração em θ_7 ou p_y (em qualquer um dos pontos da trajetória), as coordenadas de p_x (apenas para o caso de alteração de θ_7) e θ_1 serão recalculadas. Posteriormente, a partir do valor recalculado de θ_1 , ocorrerão os cálculos de θ_5 e θ_6 . Por fim, ocorrerá o cálculo de θ_2 . Caso haja a alteração de θ_3 ou θ_4 (em qualquer um dos pontos da trajetória), apenas θ_2 será recalculado. Um passo importante após este cálculo é a verificação da factibilidade dos valores obtidos, através da constatação de que os valores dos ângulos encontram-se dentro dos limites da faixa de trabalho de cada

junta do robô. Na implementação, sempre que uma solução inactível é obtida, durante o processo de busca, esta solução candidata será descartada e ocorrerá novo sorteio de variável independente. Caso a solução inicial seja inactível, uma nova solução inicial deverá ser gerada, pois não são admitidas soluções inactíveis. A geração da solução inicial se dá sequencialmente sorteando-se os valores das variáveis independentes θ_7 e p_y . A partir daí são calculados os valores das variáveis dependentes, pela ordem, p_x , θ_1 , θ_5 e θ_6 . Após isso, sorteiam-se os valores das variáveis independentes θ_3 e θ_4 . Por fim, ocorrerá o cálculo da variável dependente θ_2 .

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right) \quad \text{Eq. 5.19}$$

$$\theta_5 = \arccos\left(r_{13} \text{sen}(\theta_1) - r_{23} \cos(\theta_1)\right) \quad \text{Eq. 5.20}$$

$$\theta_6 = \arctan\left(\frac{r_{22} \cos(\theta_1) - r_{12} \text{sen}(\theta_1)}{r_{11} \text{sen}(\theta_1) - r_{21} \cos(\theta_1)}\right) \quad \text{Eq. 5.21}$$

$$\theta_2 = \arctan\left(\frac{r_{33}}{r_{13} \cos(\theta_1) + r_{23} \text{sen}(\theta_1)}\right) - \theta_3 - \theta_4 \quad \text{Eq. 5.22}$$

Determinadas todas as variáveis de junta, passa-se para o cálculo da função objetivo e a aplicação do **Critério de Metrópolis**, caso a solução candidata seja pior que a solução vigente. Se a solução encontrada atende ao critério, que, resumidamente, permite a aceitação de uma solução pior que a solução vigente para fugir de ótimos locais, a mesma é aceita como solução candidata. A cada 500 iterações sem que uma nova melhor solução (s^*), um novo “reaquecimento” é feito.

5.4 Geração de trajetória usando o algoritmo da busca tabu

Na aplicação do método metaheurístico da Busca Tabu, primeiramente é feito o sorteio aleatório do incremento. Similarmente ao que foi implementado no algoritmo do SA, foram adotadas medidas para que o incremento sorteado não fosse nulo e, adicionalmente, que ele também avaliasse a variação do sinal (positivo ou negativo) do incremento, com o intuito de que o algoritmo realizasse uma busca mais completa. Novamente, de forma

similar ao SA, para cada solução candidata, são definidos os valores das variáveis de junta e verificado se estes valores são factíveis, para então, realizar-se o cálculo da função objetivo. A partir deste momento, há uma grande diferença de abordagem entre as duas técnicas para a busca da melhor solução. O algoritmo da BT realiza a busca do melhor valor de cada variável do problema. A cada iteração são testadas todas as variáveis independentes, considerando-se tanto a possibilidade de um incremento positivo como negativo. Após determinar qual alteração de variável proporcionou a melhor solução na vizinhança da solução vigente, esta solução candidata é aceita como nova solução vigente e a variável alterada é classificada como tabu. Na escolha da melhor solução (nova solução vigente) determina-se o melhor movimento não tabu ou o movimento tabu que gerou uma solução melhor que a melhor solução encontrada (entre todas as iterações executadas) até aquele momento.

Quando uma nova solução vigente (s) é definida, a busca concentra-se em verificar qual é o melhor valor da variável tabu que proporcionará a maior melhoria da solução vigente, constituindo-se numa busca local (passo 7 da figura 5.9) da variável recém alterada. Quando este valor é encontrado, é iniciada uma nova iteração e novamente a busca inicia a procura por outra variável. A variável assumida como tabu é memorizada (na lista tabu) e não pode ter o seu valor alterado por um período de k iterações. No critério de aspiração, basicamente, verifica-se a lista tabu. Se algum movimento tabu gerar uma solução melhor que a melhor solução já encontrada até então, ele é aceito e esta solução é gravada como nova solução vigente.

```

1  Gera solução inicial ( $s$ )
2  Faz melhor solução ( $s^*$ ) = solução inicial ( $s$ )
3  Enquanto o critério de parada não é atingido:
4    Sorteia o incremento a ser usado nas soluções candidatas
5    Identifica a melhor solução candidata ( $s'$ ) vizinha da solução vigente ( $s$ )
6    Faz nova solução vigente ( $s$ ) = melhor solução candidata ( $s'$ )
7    Realiza busca local na nova solução vigente, checando se o uso do incremento (positivo ou negativo), definido no passo 4, seguirá melhorando a solução vigente
8    Atualiza lista tabu
9    Se melhor solução ( $s^*$ ) > Nova solução vigente ( $s$ ), então:
10     Faz melhor solução ( $s^*$ ) = Nova solução vigente ( $s$ )
11  Retorna à linha 3
12  Escreve melhor solução ( $s^*$ )

```

Figura 5.9. Algoritmo da busca tabu adotado neste trabalho.

5.4.1 Busca na vizinhança

A busca na vizinhança possibilita ao algoritmo a exploração das soluções “vizinhas”, que podem melhorar a solução encontrada até a iteração corrente. A avaliação da vizinhança é, em geral, o passo que consome mais tempo do processo de busca.

No algoritmo, foi adotado como parâmetro um valor de amplitude do incremento para a realização da busca pela melhor solução. Quando o incremento é definido aleatoriamente a cada iteração, isso ocorre multiplicando-se um número aleatório (entre 0 e 1) pelo parâmetro “amplitude”. Com o intuito de refinar a busca, a cada vez que uma nova solução melhor for encontrada, o valor da amplitude torna-se igual a 10. Da mesma forma, no momento em que não é encontrada uma nova solução melhor que a vigente, este valor é reduzido para 1. Isto foi feito tentando-se favorecer uma busca mais detalhada na vizinhança da solução vigente.

A diversificação tem como objetivo fazer com que o algoritmo saia de ciclos e dê novas opções de busca, e assim possa encontrar soluções melhores do que aquelas já identificadas. Nesta implementação, a diversificação é executada pela função *SolucaoInicial*. Na primeira iteração, a solução inicial gerada é aleatória.

5.5 Análise da variância

Esta seção foi extraída de HIDESON (2004) e, a princípio, seria apresentada como um anexo. No entanto, ela foi incluída neste capítulo porque esta abordagem estatística será usada para comparar as diferentes abordagens usadas. A análise da variância (ANOVA) é um método estatístico no qual através de testes de igualdade de médias, verifica-se se os tratamentos produzem mudanças sistemáticas em alguma variável de interesse. É um procedimento usado na comparação de três ou mais tratamentos, existindo muitas variações devido aos diferentes tipos de experimentos que podem ser realizados. Para se comparar as médias de vários experimentos, dependendo do número de experimentos, pode existir um número muito grande de pares de experimentos a serem analisados para serem comparados. Por exemplo, em 8 experimentos tem-se: Número de pares = $[a(a - 1)]/2$ pares possíveis de experimentos a comparar; ou seja: $(8 \times 7)/2 = 28$ pares. Para estes casos é possível utilizar a análise da Variância, para testar a hipótese de igualdade entre as médias dos experimentos.

5.5.1 Tratamento

Um tratamento (pode ser chamado também de fator) é uma condição imposta ou objeto que se deseja medir ou avaliar na execução do experimento, sendo utilizados normalmente mais de um tratamento (i.e.: equipamentos de diferentes marcas, diferentes tamanhos de peças, diferentes valores atribuídos a uma variável de um algoritmo, entre outros). Os fatores podem ser variáveis quantitativas (quando dispostos em uma ordem; i.e.: níveis de temperatura, quantidade de lubrificantes, entre outros) ou qualitativas (quando não são dispostos em ordem; i.e.: marcas de equipamentos, variedade de plantas, entre outros.), sendo chamados de variáveis independentes (i.e.: se o fator for a temperatura, as diferentes temperaturas são os níveis do fator) e a variável resposta do fator é chamada de dependente. Na aplicação desta análise há duas suposições básicas a serem satisfeitas:

- a) As amostras devem ser aleatórias e independentes;
- b) As amostras devem ser extraídas de populações normais;

5.5.2 Unidade experimental ou parcela

Unidade experimental ou parcela é onde se faz a aplicação do tratamento, sendo então ela que fornece os dados para serem avaliados (i.e.: um motor, uma peça de motor, entre outros) podendo serem formadas por grupos ou indivíduos.

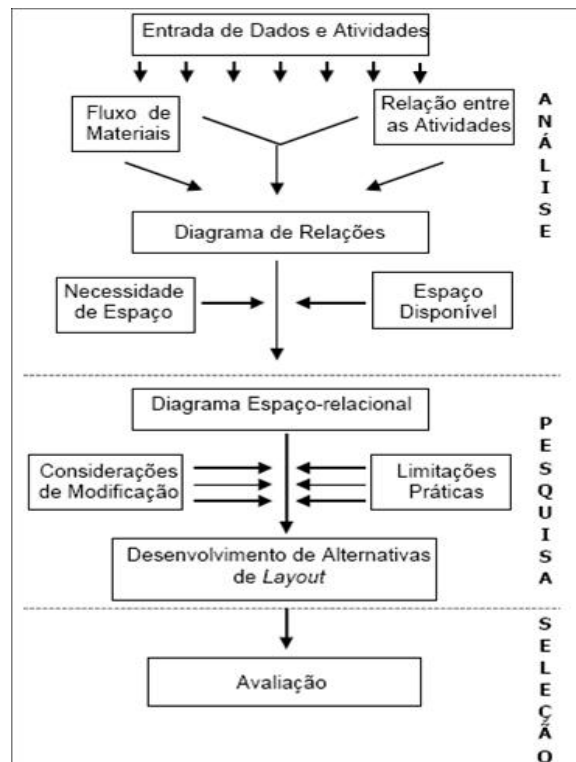


Figura 5.10. Procedimento do Planejamento Sistemático de *Layout* (SLP)(TREIN, 2001).

5.5.3 Repetição

A repetição é o número de vezes que um fator aparece no experimento, dependendo do tipo do experimento (delineamento), dos recursos disponíveis, da variabilidade ou da variável resposta. Estimar um número satisfatório de repetições pode ser algo difícil, existindo várias metodologias com essa finalidade, sendo muito importante a experiência do pesquisador sobre o algoritmo.

A média é calculada conforme a equação 5.23. A hipótese nula (H_0), na ANOVA, considera que não existe diferença entre as médias de diferentes amostras (ou tratamentos), isto é dado por:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

A hipótese alternativa é $H_A : \mu_i \neq \mu_j$, para pelo menos um par (i, j).

$$Y_{i+} = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i} \quad \text{Eq. 5.23}$$

A análise da variância para cada conjunto de resultados obtidos com os algoritmos é calculada conforme a equação 5.24:

$$S_i^2 = \frac{\sum_{j=1}^n (Y_{ij} - i+)^2}{n-1} \quad \text{Eq. 5.24}$$

Na equação 5.25 combinam-se as estimativas das variâncias:

$$S_R^2 = \frac{\sum (n_i - 1) S_i^2}{\sum (n_i - 1)} \quad \text{Eq. 5.25}$$

A variabilidade entre os tratamentos é calculado pela equação 5.26, onde S_T^2 é chamado de Quadrado Médio de Tratamento, ou Quadrado Médio entre Tratamentos. Na equação 5.26 o termo $\overline{Y_{i+}}$ representa a média do tratamento i e $\overline{Y_{++}}$ representa a média dos tratamentos.

$$S_T^2 = \frac{\sum_{i=1}^k \sum_{j=1}^i \left(\bar{Y}_{i+} - \bar{Y}_{++} \right)^2}{k-1} \quad \text{Eq. 5.26}$$

A comparação entre essas duas estimativas permite testar a hipótese de igualdade das médias do tratamento. A comparação é feita através da equação 5.27. Essa razão é denominada razão F ou F_{exp} .

$$F_{\text{exp}} = \frac{S_T^2}{S_R^2} \quad \text{Eq. 5.27}$$

Neste ponto há duas estimativas da variância do erro, uma baseada na variância dentro dos tratamentos (S_R^2) e outra baseada na variância entre os tratamentos (S_T^2). Se a hipótese H_0 for aceita, não se pode dizer que existem diferenças significativas entre as médias. Caso a hipótese seja diferente, a diferença entre elas pode ter sido causada pela diferença entre as médias analisadas. O valor esperado de F_{exp} é aproximadamente igual a 1 se H_0 for verdadeiro. Quanto maior o valor de F, maior é a evidência de que H_0 é falso. A estatística F_{exp} é o critério utilizado para testar a hipótese H_0 . Como o critério envolve a comparação de duas estimativas de variância, o procedimento é denominado de Análise da Variância (ANOVA).

Uma medida da variabilidade total das observações é dada pela decomposição da Soma de Quadrados Total, conforme a equação 5.28,

$$SQ_{\text{total}} = SQ_T + SQ_R \quad \text{Eq. 5.28}$$

onde :

- SQ_T é o numerador de S_T^2 , denominado soma de quadrados devido aos algoritmos;
- SQ_R é o numerador de S_R^2 , denominado Soma de Quadrado de Resíduo ou Soma de Quadrado do Erro;

A tabela 5.1 representa a distribuição das equações na tabela ANOVA.

Tabela 5.1 - Disposição das equações na Tabela ANOVA.

Fonte	Graus de Liberdade	Soma Dos Quadrados	Quadrados Médios	Teste F_{EXP}
Algoritmos	$k - 1$	SQ_T	QM_T	QM_T / QM_R
Resíduo	$n - K$	SQ_R	QM_R	
Total	$N - 1$	SQ_{TOTAL}		

Quanto maior o valor do F_{exp} , maior é a evidência contra H_0 .

5.5.4 Experimentos fatoriais

Em determinadas situações, busca-se a melhor combinação de níveis de fatores que maximizem o rendimento. Usa-se como fator para as variáveis que estão sendo estudadas, daí o nome de *experimento fatorial*, para a classe de planos experimentais que busca analisar o efeito de um conjunto de fatores em uma determinada resposta. Um experimento fatorial 2^k completo envolve um plano com k fatores, cada um com dois níveis. Como fatores têm todos dois níveis, adota-se a notação (-) para o nível “fator ausente” e (+) para o nível “fator presente”. A tabela 5.2 apresenta o plano experimental com a notação adotada.

Tabela 5.2 - Plano experimental para algoritmo SA.

Rodada	A	B	C	AB	AC	BC	ABC
1	-	-	-	+	+	+	-
2	-	-	+	+	-	-	+
3	-	+	-	-	+	-	+
4	-	+	+	-	-	+	-
5	+	-	-	-	-	+	+
6	+	-	+	-	+	-	-
7	+	+	-	+	-	-	-
8	+	+	+	+	+	+	+

A primeira informação desejada é obter o efeito de cada fator na resposta. O efeito de um fator é definido como a mudança que ocorre na resposta quando um fator muda de um nível para o outro (efeito principal do fator). Calcula-se a média do rendimento em cada um dos níveis do fator e calcula-se a diferença entre essas médias. Para o cálculo dos rendimentos são apresentadas as equações 5.29, 5.30, 5.31, 5.32, 5.33 e 5.34.

$$\sum_{i=1}^n \sum_{j=1}^p \sum_{k=1}^q \sum_{l=1}^r Y_{ijkl} = \sum_{i=1}^{20} \sum_{j=1}^2 \sum_{k=1}^2 \sum_{l=1}^2 Y_{ijkl} \quad \text{Eq. 5.29}$$

$$[Y] = \frac{\left(\sum_{i=1}^n \sum_{j=1}^p \sum_{k=1}^q \sum_{l=1}^r Y_{ijkl} \right)^2}{npqr} = \frac{\left(\sum_{i=1}^{20} \sum_{j=1}^2 \sum_{k=1}^2 \sum_{l=1}^2 Y_{ijkl} \right)^2}{160} \quad \text{Eq. 5.30}$$

$$[ABCS] = \sum_{i=1}^n \sum_{j=1}^p \sum_{k=1}^q \sum_{l=1}^r Y_{ijkl}^2 = \sum_{i=1}^{20} \sum_{j=1}^2 \sum_{k=1}^2 \sum_{l=1}^2 Y_{ijkl}^2 \quad \text{Eq. 5.31}$$

$$[A] = \sum_{j=1}^p \frac{\left(\sum_{i=1}^n \sum_{k=1}^q \sum_{l=1}^r Y_{ijkl} \right)^2}{nqr} = \sum_{j=1}^2 \frac{\left(\sum_{i=1}^{20} \sum_{k=1}^2 \sum_{l=1}^2 Y_{ijkl} \right)^2}{80} \quad \text{Eq. 5.32}$$

$$[B] = \sum_{k=1}^q \frac{\left(\sum_{i=1}^n \sum_{j=1}^p \sum_{l=1}^r Y_{ijkl} \right)^2}{npr} = \sum_{k=1}^2 \frac{\left(\sum_{i=1}^{20} \sum_{j=1}^2 \sum_{l=1}^2 Y_{ijkl} \right)^2}{80} \quad \text{Eq. 5.33}$$

$$[C] = \sum_{l=1}^r \frac{\left(\sum_{i=1}^n \sum_{j=1}^p \sum_{k=1}^q Y_{ijkl} \right)^2}{npq} = \sum_{l=1}^2 \frac{\left(\sum_{i=1}^{20} \sum_{j=1}^2 \sum_{k=1}^2 Y_{ijkl} \right)^2}{80} \quad \text{Eq. 5.34}$$

Se, por exemplo, o efeito médio de um fator foi estimado como sendo 23, esse é o valor a diferença em rendimento entre os níveis deste fator. A questão é: esse valor depende ou não dos níveis dos outros fatores? Se não depende, diz-se que não há interação, caso contrário, diz-se que há interação. O cálculo da interação entre os fatores é apresentado pelas equações 5.35, 5.36 e 5.37:

$$[AB] = \sum_{j=1}^p \sum_{k=1}^q \frac{\left(\sum_{i=1}^n \sum_{l=1}^r Y_{ijkl} \right)^2}{nr} = \sum_{j=1}^2 \sum_{k=1}^2 \frac{\left(\sum_{i=1}^{10} \sum_{l=1}^2 Y_{ijkl} \right)^2}{20} \quad \text{Eq. 5.35}$$

$$[AC] = \sum_{j=1}^p \sum_{l=1}^r \frac{\left(\sum_{i=1}^n \sum_{k=1}^q Y_{ijkl} \right)^2}{nr} = \sum_{j=1}^2 \sum_{l=1}^2 \frac{\left(\sum_{i=1}^{20} \sum_{k=1}^2 Y_{ijkl} \right)^2}{40} \quad \text{Eq. 5.36}$$

$$[BC] = \sum_{k=1}^q \sum_{l=1}^r \frac{\left(\sum_{i=1}^n \sum_{j=1}^p Y_{ijkl} \right)^2}{np} = \sum_{k=1}^2 \sum_{l=1}^2 \frac{\left(\sum_{i=1}^{20} \sum_{j=1}^2 Y_{ijkl} \right)^2}{40} \quad \text{Eq. 5.37}$$

A interação dos três fatores é um pouco mais complexa. Ela é a diferença entre o efeito da interação de dois fatores com respeito aos dois níveis do terceiro fator. Isto é apresentado na equação 5.38.

$$[ABC] = \sum_{j=1}^p \sum_{k=1}^q \sum_{l=1}^r \frac{\left(\sum_{i=1}^n Y_{ijkl} \right)^2}{n} = \sum_{j=1}^2 \sum_{k=1}^2 \sum_{l=1}^2 \frac{\left(\sum_{i=1}^{20} Y_{ijkl} \right)^2}{20} \quad \text{Eq. 5.38}$$

Cada fator tem apenas dois níveis, então diz-se que cada fator tem um grau de liberdade. Os graus de liberdade de uma interação é igual ao produto dos graus de liberdade de cada fator que compõe a interação. Como cada fator tem um grau de liberdade, cada interação tem também um grau de liberdade. O número de graus de liberdade de um experimento é igual ao número de observações menos um. Um experimento fatorial 2^k tem portanto $2^k - 1$ graus de liberdade. A tabela 5.3 apresenta um algoritmo para o cálculo dos efeitos principais e interações.

O desvio padrão é o mesmo para qualquer efeito fatorial. Isso é devido ao fato que o cálculo de qualquer efeito colateral usa todas as observações o mesmo número de vezes. O desvio padrão de um efeito colateral em um experimento 2^k que é replicado n vezes é dado pela equação 5.39.

$$D.P.(efeito) = \sqrt{\frac{1}{n2^{k-2}} S_R^2} \quad \text{Eq. 5.39}$$

Sendo S_R^2 calculado conforme a equação 5.40:

$$S_R^2 = \frac{1}{2^k(n-1)} \sum_{i=1}^{2^k} \sum_{j=1}^n (y_{ij} - \bar{y}_i)^2 \quad \text{Eq. 5.40,}$$

e y_{ij} a j -ésima replicação da i -ésima combinação dos fatores. Um efeito fatorial é significativo se o intervalo, dado pela equação 5.41, não contiver o número zero.

$$\text{Efeito} \pm 2 * D.P. \quad \text{Eq. 5.41}$$

Tabela 5.3 - Tabela resumo da análise de fatores.

	Graus de liberdade	SQ	MQ	F
A	$(p - 1) = 1$	$[A] - [Y]$	SQ_A / gl_A	$MQ_A / MQ_{Residuo}$
B	$(q - 1) = 1$	$[B] - [Y]$	SQ_B / gl_B	$MQ_B / MQ_{Residuo}$
C	$(r - 1) = 1$	$[C] - [Y]$	SQ_C / gl_C	$MQ_C / MQ_{Residuo}$
AB	$(p - 1)(q - 1) = 1$	$[AB] - [A] - [B] + [Y]$	SQ_{AB} / gl_{AB}	$MQ_{AB} / MQ_{Residuo}$
AC	$(p - 1)(r - 1) = 1$	$[AC] - [A] - [C] + [Y]$	SQ_{AC} / gl_{AC}	$MQ_{AC} / MQ_{Residuo}$
BC	$(q - 1)(r - 1) = 1$	$[BC] - [B] - [C] + [Y]$	SQ_{BC} / gl_{BC}	$MQ_{BC} / MQ_{Residuo}$
ABC	$(p - 1)(q - 1)(r - 1) = 1$	$[ABC] - [AB] - [AC] - [BC] + [A] + [B] + [C] - [Y]$	SQ_{ABC} / gl_{ABC}	$MQ_{ABC} / MQ_{Residuo}$
RESÍDUO	$pqr(n - 1) = 152$	$[ABCS] - [ABC]$	SQ_A / gl_A	
TOTAL	$pqrn - 1 = 159$	$[ABCS] - [Y]$		

6 RESULTADOS

As duas abordagens implementadas (de busca tabu e de *simulated annealing*) foram testadas para vários parâmetros diferentes. Por exemplo, a abordagem de busca tabu foi testada variando-se o número de iterações tabu de um movimento e a amplitude do incremento. Da mesma forma, o algoritmo de *simulated annealing* foi testado variando-se a taxa de resfriamento (α), a temperatura inicial (T_o) e a amplitude do incremento.

Devido ao grande número de experimentos realizados, adotou-se a Análise de Variância (ANOVA) como procedimento para a comparação entre experimentos e para a identificação dos fatores significativos. Ou seja, buscou-se a identificação dos fatores que, ao serem variados entre dois níveis ou valores distintos, produziram uma variação significativa entre os resultados do valor da função objetivo e do número de iterações realizadas até o fim da busca. Cada experimento foi repetido 20 vezes. Nos resultados apresentados, o valor calculado para o desvio padrão (ou erro) dos experimentos é comparado ao valor de F (na tabela ANOVA). Se o módulo do valor de F for maior que o valor do erro (desvio padrão dos experimentos), então pode-se considerar que este fator é significativo.

6.1 Análise dos resultados para o cordão linear

Nesta seção são apresentados os resultados para as duas abordagens implementadas, inclusive com a comparação entre as duas abordagens.

- **Análise do *Simulated Annealing* (SA) proposto**

Para o algoritmo o SA proposto testou-se os fatores:

- A (taxa de resfriamento), para os valores $\alpha = 0,9$ e $\alpha = 0,999$;
- B (temperatura inicial), para os valores $T_o = 100$ e $T_o = 500$;
- C (amplitude), para os valores: amplitude = 1 e amplitude = 10.

Analisando-se a tabela 6.1, que apresenta a tabela ANOVA destes experimentos, pode-se afirmar estatisticamente que todos os fatores (A, B e C) são significativos, ou melhor, pode-se afirmar que a interação entre os três fatores (ABC) foi significativa na obtenção de uma solução melhor para a função objetivo. Isso é verificado porque os valores de F são superiores ao erro (45,34). Apesar de não ser trivial, esta interação entre os três fatores pode ser comprovada tomando-se as médias dos experimentos tabela 6.2 e

associando a cada média o sinal correspondente no plano experimental da tabela 5.2 (na coluna referente à interação ABC), que resultará na expressão:

$$\text{Efeito} = (-376,598 + 257,917 + 514,1785 - 249,457 + 244,47 - 222,726 - 261,624 + 284,6055) / 4$$

Como o valor do *efeito* (interação ABC) é igual a 47,69, isto indica a expectativa estatística de que o valor da função objetivo seja menor quando a taxa de resfriamento (fator **A**) é 0,999, a temperatura inicial (fator **B**) é 100 e a amplitude (fator **C**) é 10 (vide BOX *et al.*, 1978).

A tabela 6.2 apresenta os valores da função objetivo nos 8 experimentos realizados usando *simulated annealing* para o cordão linear. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrados. Nota-se nesta tabela que o valor mínimo encontrado para a função objetivo em cada um dos experimentos é significativamente menor que a média dos valores encontrados. Esta é uma indicação da complexidade deste problema, devido à multiplicidade de soluções viáveis, conforme indicado na figura 3.3, bem como à existência de inúmeros pontos de ótimo local.

Baseado nos dados da tabela 6.2, foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 45,34 (conforme a equação 5.39 e 5.41).

Tabela 6.1. Tabela ANOVA da função objetivo nos experimentos de SA para o cordão linear.

	Graus de liberdade	SQ	MQ	F
A (Taxa de resfriamento)	1	-12708868,01	-12708868,01	-73,38
B (Temperatura inicial)	1	-12811773,50	-12811773,50	-73,98
C (Amplitude)	1	-12717653,29	-12717653,29	-73,43
AB	1	12827539,06	12827539,06	74,07
AC	1	12911754,58	12911754,58	74,56
BC	1	12837079,29	12837079,29	74,12
ABC	1	-12809610,41	-12809610,41	-73,97
RESÍDUO	152	26323843,66	173183,18	
TOTAL	159	13852311,38		

A tabela 6.3 apresenta o número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão linear. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrado.

Tabela 6.2. Apresentação do valor da função objetivo nos 8 experimentos realizados usando *simulated annealing* para o cordão linear.

Tratamento	1	2	3	4	5	6	7	8
Taxa Resfriamento (A)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)
Temper. Inicial (B)	100 (-)	100 (-)	500 (+)	500 (+)	100 (-)	100 (-)	500 (+)	500 (+)
Amplitude (C)	1 (-)	1 (-)	1 (-)	1 (-)	10 (+)	10 (+)	10 (+)	10 (+)
Repetições	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução
1	398,83	9,40	567,57	370,56	884,71	384,49	2,48	17,87
2	573,63	38,29	239,79	48,96	548,59	1,75	7,36	404,03
3	941,07	43,32	675,43	41,92	93,88	409,29	550,99	390,82
4	468,10	5,06	611,51	426,79	118,99	72,47	499,48	4,73
5	110,66	2,24	256,22	383,60	50,26	373,23	80,61	424,00
6	160,82	133,44	137,88	69,12	492,39	383,64	26,80	396,52
7	72,99	750,72	277,01	118,40	380,61	24,94	44,34	14,18
8	418,10	74,88	533,74	33,60	42,71	5,51	516,83	390,86
9	1067,41	2,04	197,34	375,38	400,25	369,48	194,08	402,93
10	319,18	371,82	263,49	879,30	30,81	8,07	13,04	5,56
11	113,10	888,34	1120,83	370,70	50,68	378,30	853,27	370,22
12	538,62	370,38	254,10	375,11	64,22	24,47	421,32	383,70
13	246,74	369,14	552,42	367,01	110,40	4,75	56,44	395,51
14	291,43	419,62	935,82	46,08	175,05	394,86	886,11	381,58
15	103,78	6,58	662,96	62,52	569,62	437,25	61,39	27,79
16	625,58	374,50	273,30	898,54	13,14	383,02	438,89	895,08
17	150,50	2,56	608,02	29,44	751,63	414,92	10,09	413,15
18	540,97	384,04	375,14	26,19	25,95	366,14	22,79	1,43
19	172,23	3,32	641,13	7,68	74,04	11,91	160,22	370,11
20	218,22	908,65	1099,87	58,24	11,47	6,03	385,95	2,04
Mínimo	72,99	2,04	137,88	7,68	11,47	1,75	2,48	1,43
Média	376,598	257,917	514,1785	249,457	244,47	222,726	261,624	284,6055
Variância	998888,4229	92126,49689	85472,61032	73233,03567	74185,90118	36637,65781	82311,56084	54979,87673
Desvio padrão	277,1307511	303,5234701	292,3569912	270,61603	272,370889	191,4096597	286,8999143	234,4778811

Tabela 6.3. Apresentação do número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão linear.

Tratamento	1	2	3	4	5	6	7	8
Taxa Resfriamento (A)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)
Temper. Inicial (B)	100 (-)	100 (-)	500 (+)	500 (+)	100 (-)	100 (-)	500 (+)	500 (+)
Amplitude (C)	1 (-)	1 (-)	1 (-)	1 (-)	10 (-)	10 (-)	10 (-)	10 (-)
Repetições	iterações	iterações	iterações	iterações	iterações	iterações	iterações	iterações
1	9097	47066	9731	25963	10587	71312	9706	40765
2	6489	46016	3216	109442	6309	42628	10857	99703
3	11848	33951	7706	40007	14774	35479	2784	86774
4	8247	68134	9125	29922	13311	61601	13647	34988
5	9973	51887	8293	79835	6807	49585	8405	63540
6	7073	65226	6944	23476	4572	8570	7777	57613
7	7729	52906	9002	112333	9751	19480	777	93600
8	7434	41970	9319	11818	7316	48987	6935	38444
9	3809	36322	8999	39853	14114	8848	5327	61992
10	8721	70281	8837	39971	7743	36634	12240	88203
11	8584	141257	12712	53340	5468	29474	13030	35664
12	9196	61044	8882	62158	9666	57671	18487	83384
13	6794	77532	6969	52708	8060	39304	10613	73390
14	8461	77956	8519	11273	4245	38819	2686	81253
15	9167	7913	6610	14266	5920	48193	9606	96707
16	9576	57686	6912	120160	9972	65625	9898	31530
17	7298	32372	6388	93793	870	30275	12365	34228
18	6800	47446	10303	13704	4954	28854	2819	23298
19	9719	22679	8976	11529	10161	31476	8780	72482
20	6871	143200	8983	103742	11615	59013	8114	48677
Mínimo	3809	7913	3216	11273	870	8570	777	23298
Média	8144,3	59142,2	8321,3	52464,65	8310,75	40591,4	8742,65	62311,75
Variância	2874466,853	1127142959	3582314,537	1426419091	12906759,04	306968089,4	18990698,03	621911964,3
Desvio padrão	1695,425272	33572,94982	1892,700329	37767,96381	3592,597812	17520,50483	4357,831804	24938,16281

Para o número de iterações, o valor calculado para o erro (desvio padrão dos experimentos) foi igual a 3546,02 (conforme a equação 5.39 e 5.41), baseado nas médias apresentadas na tabela 6.3, que apresenta o número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão linear. Analisando-se a tabela 6.4, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com menor número de iterações. Ainda que haja uma diferença entre as médias dos valores dos experimentos (combinações de fatores) testados, a variância associada aos experimentos levou a valores de F inferiores ao erro (igual a 3546,02).

Tabela 6.4. Tabela ANOVA do número de iterações nos experimentos de SA para o cordão linear.

	Graus de liberdade	SQ	MQ	F
A (Taxa de resfriamento)	1	81894355202,50	81894355202,50	186,08
B (Temperatura inicial)	1	612439282,23	612439282,23	1,39
C (Amplitude)	1	164669582,02	164669582,02	0,37
AB	1	520843673,02	520843673,02	1,18
AC	1	215829930,63	215829930,63	0,49
BC	1	2052457369,60	2052457369,60	4,66
ABC	1	1980071122,50	1980071122,50	4,50
RESÍDUO	152	66895130501,00	440099542,77	
TOTAL	159	154335796663,50		

- **Análise da Busca Tabu (BT) proposta**

A tabela 6.5 apresenta os valores da função objetivo nos experimentos realizados usando busca tabu para o cordão linear. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrado. Novamente, nota-se nesta tabela que o valor mínimo encontrado para a função objetivo em cada um dos experimentos é significativamente menor que a média dos valores encontrados.

Baseado nos dados da tabela 6.5, foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 7,75 (conforme a equação 5.39 e 5.41). Analisando-se a tabela 6.6, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com função objetivo melhor. Esta conclusão é obtida observando-se que não há valores superiores ao erro (7,75) na coluna de F .

Tabela 6.5. Valor da função objetivo nos experimentos usando *BT* para o cordão linear.

Tratamento	1	2	3	4
Amplitude (A)	1 (-)	10 (+)	1 (-)	10 (+)
Iter. tabu (B)	3 (-)	3 (-)	5 (+)	5 (+)
Repetições	Melhor solução	Melhor solução	Melhor solução	Melhor solução
1	29,8800	2,2400	1,7500	42,6100
2	42,3600	23,9500	43,0900	2,5000
3	16,7600	24,4700	66,5600	59,4500
4	62,7200	17,1400	2,8300	59,3700
5	138,4800	11,6400	9,8400	2,8300
6	14,0800	33,1800	64,1200	3,0100
7	28,7200	11,3500	33,5200	18,5400
8	67,6400	28,1600	6,6400	21,3100
9	21,3600	123,6700	3,1500	22,1000
10	7,1600	35,2000	50,6800	59,1000
11	67,0000	1,7200	62,5200	2,9500
12	128,4400	30,0400	15,1600	8,7100
13	63,9100	35,3900	17,0800	17,6500
14	66,7900	8,9600	44,9200	21,6100
15	61,4400	21,5600	11,1500	48,0100
16	2,6800	2,6800	10,0400	42,5800
17	33,2000	81,5000	5,5600	6,4000
18	55,6800	12,2800	9,2800	102,0300
19	2,6800	71,4300	14,7200	15,3700
20	18,8000	65,5100	28,1600	53,3700
Mínimo	2,68	1,72	1,75	2,5
Média	46,489	32,1035	25,0385	30,475
Variância	1412,432852	970,8930766	500,7219608	716,8083211
Desvio padrão	37,58234761	31,15915719	22,37681749	26,77327625

Tabela 6.6. Tabela ANOVA da função objetivo nos experimentos de *BT* para o cordão linear.

	Graus de liberdade	SQ	MQ	F
Amplitude (A)	1	400,42	400,42	0,44
Iter. tabu (B)	1	2663,20	2663,20	2,96
AB	1	1964,56	1964,56	2,18
RESÍDUO	76	68416,27	900,21	
TOTAL	79	73444,45		

Para o número de iterações, a tabela 6.7 apresenta estes resultados nos experimentos realizados usando busca tabu para o cordão linear. Baseado nos dados da tabela 6.7 (número de iterações), foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 330,62 (conforme a equação 5.39 e 5.41). Analisando-se a tabela 6.8, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com função objetivo melhor. Esta conclusão é obtida observando-se que não há valores superiores ao erro (330,62) na coluna de *F*.

Tabela 6.7. Número de iterações nos experimentos usando *BT* para o cordão linear.

Tratamento	1	2	3	4
Amplitude (A)	1 (-)	10 (+)	1 (-)	10 (+)
Iter. tabu (B)	3 (-)	3 (-)	5 (+)	5 (+)
Repetições	iterações	iterações	iterações	iterações
1	1073	1248	2936	164
2	954	2787	605	3222
3	4249	294	4163	896
4	2882	2278	3706	1443
5	1694	2872	3290	297
6	2538	1909	3804	2838
7	181	793	5143	4119
8	1253	1519	2640	885
9	199	610	1675	3388
10	942	2960	634	399
11	1619	184	2823	1502
12	2746	2183	2599	4081
13	3898	1703	2127	2181
14	1410	522	1687	773
15	3025	2040	340	103
16	2823	1079	2312	1123
17	1356	3130	400	532
18	725	325	4785	2170
19	1226	358	4813	2050
20	3603	2149	669	378
Mínimo	181	184	340	103
Média	1919,8	1547,15	2557,55	1627,2
Variância	1466459,221	980867,0816	2388208,366	1723217,537
Desvio padrão	1210,974492	990,3873392	1545,382919	1312,713806

Tabela 6.8. Tabela ANOVA do número de iterações nos experimentos de *BT* para o cordão linear.

	Graus de liberdade	SQ	MQ	F
Amplitude (A)	1	8489045,00	8489045,00	5,18
Iter. tabu (B)	1	2576184,20	2576184,20	1,57
AB	1	1555146,45	1555146,45	0,95
RESÍDUO	76	124616291,90	1639688,05	
TOTAL	79	137236667,55		

- **Comparação entre as abordagens de *simulated annealing* e *busca tabu***

Na comparação entre as abordagens, realizou-se primeiro a análise entre as abordagens de busca tabu e de *simulated annealing*, comparando-se o melhor experimento de cada uma das abordagens. Note que, do ponto de vista estatístico, não houve um melhor resultado para a *busca tabu*. Assim, para a *busca tabu*, tomou-se o experimento com 5 iterações tabu e amplitude igual a 1, por ter apresentado a melhor média para a função objetivo ($25,04 \pm 22,38$). Para o *simulated annealing*, tomou-se o experimento com taxa de resfriamento de 0,999, temperatura inicial igual a 100 e amplitude igual a 10, porque este

experimento foi estatisticamente superior (de acordo com a interação entre fatores ABC), cuja média dos valores da função objetivo foi $222,73 \pm 191,41$. A tabela 6.9 apresenta a comparação entre as duas abordagens, indicando os valores do mínimo, média, variância e desvio padrão para cada experimento. Ainda que ambas as abordagens tenham chegado ao mesmo valor mínimo (1,75), há uma diferença significativa na média dos dois experimentos. A tabela 6.10 apresenta a avaliação entre estes experimentos. Note que o valor de F (igual a 21,05) é superior ao valor de corte 4,1 (definido a partir do nível de confiança 95% e dos graus de liberdade dos tratamentos), indicando que a busca tabu foi superior ao *simulated annealing* para o caso de solda de um cordão linear.

Tabela 6.9. Comparação entre as abordagens SA e BT.

Tratamento	1	2
	Busca Tabu	<i>Simulated Annealing</i>
		Taxa de resfriamento = 0,999
	No. diversificações = 5	Temperatura inicial = 100
	Amplitude = 1	Amplitude = 10
Repetições	Melhor solução	Melhor solução
1	1,75	384,49
2	43,09	1,75
3	66,56	409,29
4	2,83	72,47
5	9,84	373,23
6	64,12	383,64
7	33,52	24,94
8	6,64	5,51
9	3,15	369,48
10	50,68	8,07
11	62,52	378,3
12	15,16	24,47
13	17,08	4,75
14	44,92	394,86
15	11,15	437,25
16	10,04	383,02
17	5,56	414,92
18	9,28	366,14
19	14,72	11,91
20	28,16	6,03
Mínimo	1,75	1,75
Média	25,04	222,73
Variância	500,72	36637,66
Desvio padrão	22,38	191,41

Tabela 6.10. Tabela ANOVA comparando as abordagens SA e BT.

	Graus de liberdade	SQ	MQ	F
Entre tratamentos	1	390803,48	390803,48	21,05
Dentro dos tratamentos	38	705629,22	18569,19	
TOTAL	39	1096432,69		

- **Análise dos tempos de processamento**

A tabela 6.11 apresenta os tempos de processamento medidos na execução de uma busca. Tais tempos devem ser considerados apenas como uma referência, já que o número de iterações e, conseqüentemente, os tempos de execução das duas abordagens ficaram em segundo plano na comparação entre as abordagens. O objetivo aqui foi de apenas indicar que as buscas ocorreram dentro de tempos de execução aceitáveis.

Tabela 6.11. Tempo de execução dos algoritmos para o caso de cordão linear.

Abordagem	Arranjo de parâmetros	Tempo de execução
<i>Simulated Annealing</i>	Amplitude = 1, temperatura = 100, $\alpha = 0,9$	48 seg
	Amplitude = 1, temperatura = 100, $\alpha = 0,999$	11 min e 21 seg
	Amplitude = 1, temperatura = 500, $\alpha = 0,9$	51 seg
	Amplitude = 1, temperatura = 500, $\alpha = 0,999$	14 min e 22 seg
	Amplitude = 10, temperatura = 100, $\alpha = 0,9$	56 seg
	Amplitude = 10, temperatura = 100, $\alpha = 0,999$	8 min e 6 seg
	Amplitude = 10, temperatura = 500, $\alpha = 0,9$	1 min e 5 seg
	Amplitude = 10, temperatura = 500, $\alpha = 0,999$	11 min e 11 seg
Busca Tabu	Amplitude = 1, iteração tabu = 3	6 min e 14 seg
	Amplitude = 1, iteração tabu = 5	6 min e 35 seg
	Amplitude = 10, iteração tabu = 3	4 min e 26 seg
	Amplitude = 10, iteração tabu = 5	5 min e 11 seg

6.2 Análise dos resultados para o cordão circular

Nesta seção são apresentados os resultados para as duas abordagens implementadas, inclusive com a comparação entre as duas abordagens.

- **Análise do *Simulated Annealing* (SA) proposto**

Para o algoritmo o SA proposto testou-se os fatores:

- A (taxa de resfriamento), para os valores $\alpha = 0,9$ e $\alpha = 0,999$;
- B (temperatura inicial), para os valores $T_o = 100$ e $T_o = 500$;
- C (amplitude), para os valores: amplitude = 1 e amplitude = 10.

A tabela 6.13 apresenta os valores da função objetivo nos 8 experimentos realizados usando *simulated annealing* para o cordão circular. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrado. Nota-se nesta tabela que o valor mínimo encontrado para a função objetivo em cada um dos experimentos é significativamente

menor que a média dos valores encontrados. Esta é uma indicação da complexidade deste problema, devido à multiplicidade de soluções viáveis, conforme indicado na figura 3.3, bem como à existência de inúmeros pontos de ótimo local.

Baseado nos dados da tabela 6.13, foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 23,98 (conforme a equação 5.39 e 5.41). Analisando-se a tabela 6.12, que apresenta a tabela ANOVA destes experimentos, pode-se afirmar estatisticamente que todos os fatores (A, B e C) são significativos, ou melhor, pode-se afirmar que a interação entre os três fatores (ABC) foi significativa na obtenção de uma solução melhor para a função objetivo. Isso é verificado porque os valores de F são superiores ao erro (23,98).

Tabela 6.12. Tabela ANOVA da função objetivo nos experimentos de SA para o cordão circular.

	Graus de liberdade	SQ	MQ	F
A (Taxa de resfriamento)	1	-17733584,38	-17733584,38	-112,82
B (Beta)	1	-17833614,22	-17833614,22	-113,46
C (Temperatura inicial)	1	-17830803,94	-17830803,94	-113,44
AB	1	17841598,66	17841598,66	113,51
AC	1	17920884,51	17920884,51	114,01
BC	1	17843736,00	17843736,00	113,52
ABC	1	-17812604,84	-17812604,84	-113,32
RESÍDUO	152	23891662,09	157181,99	
TOTAL	159	6287273,88		

A tabela 6.14 apresenta o número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão circular. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrado.

Para o número de iterações, o valor calculado para o erro (desvio padrão dos experimentos) foi igual a 2381,41 (conforme a equação 5.39 e 5.41), baseado nas médias apresentadas na tabela 6.14, que apresenta os valores do número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão circular. Analisando-se a tabela 6.15, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com menor número de iterações. Ainda que haja uma diferença entre as médias dos valores dos experimentos (combinações de fatores) testados, a variância associada aos experimentos levou a valores de F inferiores ao erro (igual a 2381,41).

Tabela 6.13. Apresentação do valor da função objetivo nos 8 experimentos realizados usando *simulated annealing* para o cordão circular.

Tratamento	1	2	3	4	5	6	7	8
Taxa Resfriamento (A)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)
Temper. Inicial (B)	100 (-)	100 (-)	500 (+)	500 (+)	100 (-)	100 (-)	500 (+)	500 (+)
Amplitude (C)	1 (-)	1 (-)	1 (-)	1 (-)	10 (+)	10 (+)	10 (+)	10 (+)
Repetições	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução	Melhor solução
1	691,4900	386,2400	475,3500	128,9600	196,9300	386,7200	717,4300	182,8400
2	480,7700	241,6000	235,1700	520,0000	205,2400	281,1300	178,9800	169,8400
3	288,4100	252,9200	1001,9500	585,9100	279,3000	279,1800	166,9000	522,2200
4	457,6700	280,5600	466,1600	365,7600	316,1500	322,8900	582,5900	296,0600
5	520,4800	509,4400	879,5700	307,2000	338,1500	262,2800	484,6900	206,0600
6	440,1200	405,1200	417,0600	514,2400	336,1400	479,1200	483,6600	373,0700
7	677,6300	145,7200	534,5000	124,8000	428,1800	497,4800	462,3800	512,2500
8	480,3800	168,3200	600,6700	325,8800	238,6000	434,9800	437,7000	218,8400
9	328,1700	293,1200	501,5200	349,1200	558,1200	538,0200	446,6200	422,8500
10	559,1100	248,6400	715,4200	334,5200	373,6600	156,4400	231,8200	315,4700
11	599,9300	525,4400	588,0800	302,4000	425,5100	242,1700	139,3000	342,4700
12	408,7700	258,5600	403,6400	247,6800	396,0500	408,7200	310,4300	329,0600
13	607,3400	540,4800	375,5100	395,8400	336,2100	363,6400	432,3500	321,0300
14	862,7400	328,7600	872,2700	507,2000	521,1800	279,7200	192,2900	537,9700
15	217,3200	230,0000	773,4500	369,9200	361,6300	212,9100	293,3400	280,4700
16	544,9300	346,1700	351,0300	251,2000	388,7600	382,9400	268,4200	227,2200
17	309,5700	396,1600	407,0700	300,9500	347,4600	534,0200	338,7100	359,6600
18	403,8400	332,4800	766,5200	372,1600	343,0700	202,9600	566,7700	561,0900
19	319,2500	482,2400	428,7300	211,8400	399,2000	376,8400	562,8700	290,6100
20	285,1400	134,4000	539,3900	384,6400	325,0800	334,4100	570,2100	330,1900
Mínimo	217,32	134,4	235,17	124,8	196,93	156,44	139,3	169,84
Média	474,153	325,3185	566,653	345,011	355,731	348,8285	393,373	339,9635
Variância	664178,3086	15185,82312	42003,53464	15073,59155	8139,498357	12341,95115	27488,60922	14072,71284
Desvio padrão	163,3344115	123,2307718	204,9476388	122,7745558	90,21916846	111,0943345	165,7968915	118,6284656

Tabela 6.14. Apresentação do número de iterações nos 8 experimentos realizados usando *simulated annealing* para o cordão circular.

Tratamento	1	2	3	4	5	6	7	8
Taxa Resfriamento (A)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)	0,9 (-)	0,999 (+)
Temper. Inicial (B)	100 (-)	100 (-)	500 (+)	500 (+)	100 (-)	100 (-)	500 (+)	500 (+)
Amplitude (C)	1 (-)	1 (-)	1 (-)	1 (-)	10 (-)	10 (-)	10 (-)	10 (-)
Repetições	iterações	iterações	iterações	iterações	iterações	iterações	iterações	iterações
1	11007	39435	9918	24056	7288	28700	5761	50588
2	9663	40222	8206	95643	5824	22561	7250	58021
3	10922	53111	10515	98698	4741	25223	5439	18184
4	11342	66040	11532	63926	6888	44620	3716	59986
5	7551	31262	12009	62767	7203	34231	7363	48701
6	11275	62966	12473	35329	6135	30383	6687	48722
7	12003	83164	9911	85620	4862	14017	6569	19953
8	8896	31544	13380	74110	4930	29500	6482	19665
9	8353	83548	11538	35950	5185	50843	7655	67455
10	12539	29288	14004	88282	5256	39591	5069	38805
11	10209	84549	8042	36716	5272	6319	5735	30622
12	10810	81487	8344	48855	7656	20516	6901	30076
13	10252	51020	10869	24489	7764	33716	5888	9515
14	11055	76321	9555	35495	5576	13277	4596	19653
15	10207	64262	12887	76554	6182	29478	7447	67527
16	10483	79260	9561	36674	4743	35297	3891	59850
17	8872	80174	11081	35116	8305	38296	5856	48383
18	11812	70727	11049	85171	4724	29161	5417	9924
19	14241	61881	12230	23724	5597	42109	4295	49425
20	7198	39799	10549	38427	8691	50779	7411	59235
Mínimo	7198	29288	8042	23724	4724	6319	3716	9515
Média	10434,5	60503	10882,65	55280,1	6141,1	30930,85	5971,4	40714,5
Variância	2871658,053	384517776,4	2833558,871	676842811,6	1647923,989	141941675,7	1489068,147	375768054,7
Desvio padrão	1694,596723	19609,12483	1683,317816	26016,20287	1283,714918	11913,9278	1220,2738	19384,73767

Tabela 6.15. Tabela ANOVA do número de iterações nos experimentos de SA para o cordão circular.

	Graus de liberdade	SQ	MQ	F
A (Taxa de resfriamento)	1	59289076003,60	59289076003,60	298,70
B (Beta)	1	58544641,60	58544641,60	0,29
C (Temperatura inicial)	1	7113529094,40	7113529094,40	35,84
AB	1	45845233,23	45845233,23	0,23
AC	1	3050803689,02	3050803689,02	15,37
BC	1	517586719,22	517586719,22	2,61
ABC	1	610304688,40	610304688,40	3,07
RESÍDUO	152	30170338021,50	198489065,93	
TOTAL	159	100856028090,98		

- **Análise da Busca Tabu (BT) proposta**

A tabela 6.16 apresenta os valores da função objetivo nos experimentos realizados usando busca tabu para o cordão circular. Esta tabela apresenta os valores obtidos para as 20 repetições de cada experimento, indicando o valor mínimo ocorrido, a média, a variância e o desvio padrão encontrado. Novamente, nota-se nesta tabela que o valor mínimo encontrado para a função objetivo em cada um dos experimentos é significativamente menor que a média dos valores encontrados.

Baseado nos dados da tabela 6.16, foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 14,89 (conforme a equação 5.39 e 5.41). Analisando-se a tabela 6.17, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com função objetivo melhor. Esta conclusão é obtida observando-se que não há valores superiores ao erro (14,89) na coluna de **F**.

Para o número de iterações, a tabela 6.18 apresenta estes resultados nos experimentos realizados usando busca tabu para o cordão circular. Baseado nos dados da tabela 6.18, foi calculado o valor para o erro (desvio padrão dos experimentos) na análise dos valores da função objetivo, que foi igual a 581,87 (conforme a equação 5.39 e 5.41). Analisando-se a tabela 6.19, que apresenta a tabela ANOVA destes experimentos, não se pode afirmar estatisticamente que qualquer um dos fatores (A, B e C) ou que a interação entre fatores (AB, AC, BC e ABC) foi significativa na obtenção de um experimento com função objetivo melhor. Esta conclusão é obtida observando-se que não há valores superiores ao erro (581,87) na coluna de **F**.

Tabela 6.16. Valor da função objetivo nos experimentos usando *BT* para o cordão circular.

Tratamento	1	2	3	4
Amplitude (A)	1 (-)	10 (+)	1 (-)	10 (+)
Iter. tabu (B)	3 (-)	3 (-)	5 (+)	5 (+)
Repetições	Melhor solução	Melhor solução	Melhor solução	Melhor solução
1	126,64	189,76	125,04	135,66
2	173,12	130,90	162,80	128,08
3	277,52	175,52	226,39	210,39
4	318,11	182,56	215,48	144,14
5	294,41	151,36	190,20	127,82
6	237,51	278,22	161,96	136,47
7	178,75	170,29	348,52	144,89
8	133,19	195,63	213,36	166,06
9	186,28	149,53	249,28	274,98
10	314,23	208,74	237,24	244,31
11	284,40	197,06	126,12	213,37
12	217,64	266,53	208,88	209,28
13	286,12	336,98	339,32	168,53
14	166,03	163,94	207,60	223,47
15	132,43	282,07	215,36	258,93
16	183,92	146,68	123,96	189,71
17	181,92	173,35	147,86	130,17
18	173,04	147,20	261,68	209,13
19	166,32	158,42	146,80	133,83
20	190,62	168,03	246,96	140,22
Mínimo	126,64	130,9	123,96	127,82
Média	211,11	193,6385	207,7405	179,472
Variância	3956,447032	3016,294792	4048,269521	2285,751196
Desvio padrão	62,90029437	54,92080473	63,62601293	47,80953039

Tabela 6.17. Tabela ANOVA da função objetivo nos experimentos de *BT* para o cordão circular.

	Graus de liberdade	SQ	MQ	F
Amplitude (A)	1	10460,74	10460,74	3,14
Iter. tabu (B)	1	1537,56	1537,56	0,46
AB	1	582,88	582,88	0,18
RESÍDUO	76	252828,49	3326,69	
TOTAL	79	265409,66		

Tabela 6.18. Número de iterações nos experimentos usando *BT* para o cordão circular.

Tratamento	1	2	3	4
Amplitude (A)	1 (-)	10 (+)	1 (-)	10 (+)
Iter. tabu (B)	3 (-)	3 (-)	5 (+)	5 (+)
Repetições	iterações	iterações	iterações	iterações
1	3666	665	3548	2393
2	941	1002	8833	3009
3	5725	1862	8624	3532
4	3921	2987	8191	4366
5	1720	2296	8116	6933
6	5834	3743	1405	297
7	814	4099	4795	761
8	5834	4873	1653	3058
9	2518	627	4073	2147
10	5749	3196	3091	881
11	6945	649	11178	529
12	6005	3742	6114	4929
13	7414	4190	5865	2334
14	4183	2093	4350	4959
15	1037	5096	10244	3422
16	5437	3040	3426	4936
17	7638	3852	3179	1299
18	6232	4728	10143	3208
19	1987	4879	2072	1737
20	5956	4255	9037	2165
Mínimo	814	627	1405	297
Média	4477,8	3093,7	5896,85	2844,75
Variância	5080932,905	2290835,379	9873907,187	3068829,987
Desvio padrão	2254,092479	1513,550587	3142,277389	1751,807634

Tabela 6.19. Tabela ANOVA do número de iterações nos experimentos de *BT* para o cordão circular.

	Graus de liberdade	SQ	MQ	F
Amplitude (A)	1	98399352,20	98399352,20	19,38
Iter. tabu (B)	1	6845670,05	6845670,05	1,35
AB	1	13911120,00	13911120,00	2,74
RESÍDUO	76	385975603,70	5078626,36	
TOTAL	79	505131745,95		

- **Comparação entre as abordagens de *simulated annealing* e *busca tabu***

Na comparação entre as abordagens para o cordão de solda circular, novamente, realizou-se primeiro a análise entre as abordagens de busca tabu e de *simulated annealing*, comparando-se o melhor experimento de cada uma das abordagens. Note que, do ponto de vista estatístico, não houve um melhor resultado para a *busca tabu*. Assim, para a *busca tabu*, tomou-se o experimento com 5 iterações tabu e amplitude igual a 1, cuja média para a função objetivo foi $207,74 \pm 63,63$, já que experimento foi tomado na comparação para o

cordão linear. Para o *simulated annealing*, tomou-se o experimento com taxa de resfriamento de 0,999, temperatura inicial igual a 100 e amplitude igual a 1, porque este experimento apresentou a média mais baixa dos valores da função objetivo, que foi $134,4 \pm 123,23$. A tabela 6.20 apresenta a avaliação entre as duas abordagens, indicando os valores do mínimo, média, variância e desvio padrão para cada experimento. Há uma diferença significativa na média dos dois experimentos. A tabela 6.21 apresenta a comparação entre estes experimentos. Note que o valor de F (igual a 14,38) é superior ao valor de corte 4,1 (definido a partir do nível de confiança 95% e dos graus de liberdade dos tratamentos), indicando que a busca tabu foi superior ao *simulated annealing* também para o caso de solda de um cordão circular.

Tabela 6.20. Comparação entre as abordagens SA e BT.

Tratamento	1	2
	Busca Tabu	Simulated Annealing
		Taxa de resfriamento = 0,999
	No. diversificações = 5	Temperatura inicial = 100
	Amplitude = 1	Amplitude = 1
Repetições	Melhor solução	Melhor solução
1	125,04	386,2400
2	162,80	241,6000
3	226,39	252,9200
4	215,48	280,5600
5	190,20	509,4400
6	161,96	405,1200
7	348,52	145,7200
8	213,36	168,3200
9	249,28	293,1200
10	237,24	248,6400
11	126,12	525,4400
12	208,88	258,5600
13	339,32	540,4800
14	207,60	328,7600
15	215,36	230,0000
16	123,96	346,1700
17	147,86	396,1600
18	261,68	332,4800
19	146,80	482,2400
20	246,96	134,4000
Mínimo	123,96	134,4
Média	207,74	325,3185
Variância	4048,27	15185,82312
Desvio padrão	63,63	123,2307718

Tabela 6.21. Tabela ANOVA comparando as abordagens SA e BT.

	Graus de liberdade	SQ	MQ	F
Entre tratamentos	1	138245,86	138245,86	14,38
Dentro dos tratamentos	38	365447,76	9617,05	
TOTAL	39	503693,62		

- **Análise dos tempos de processamento**

A tabela 6.22 apresenta os tempos de processamento medidos na execução de uma busca. Tais tempos devem ser considerados apenas como uma referência, já que o número de iterações e, conseqüentemente, os tempos de execução das duas abordagens ficaram em segundo plano na comparação entre as abordagens. O objetivo aqui foi de apenas indicar que as buscas ocorreram dentro de tempos de execução aceitáveis.

Tabela 6.22. Tabela ANOVA do número de iterações nos experimentos de *BT* para o cordão circular.

Abordagem	Arranjo de parâmetros	Tempo de execução
Simulated Annealing	Amplitude = 1, temperatura = 100, $\alpha = 0,9$	1 min e 14 seg
	Amplitude = 1, temperatura = 100, $\alpha = 0,999$	12 min e 9 seg
	Amplitude = 1, temperatura = 500, $\alpha = 0,9$	1 min e 21 seg
	Amplitude = 1, temperatura = 500, $\alpha = 0,999$	15 min e 2 seg
	Amplitude = 10, temperatura = 100, $\alpha = 0,9$	40 seg
	Amplitude = 10, temperatura = 100, $\alpha = 0,999$	6 min e 44 seg
	Amplitude = 10, temperatura = 500, $\alpha = 0,9$	42 seg
	Amplitude = 10, temperatura = 500, $\alpha = 0,999$	9 min e 18 seg
Busca Tabu	Amplitude = 1, iteração tabu = 3	11 min e 15 seg
	Amplitude = 1, iteração tabu = 5	15 min e 10 seg
	Amplitude = 10, iteração tabu = 3	8 min e 1 seg
	Amplitude = 10, iteração tabu = 5	9 min e 1 seg

7. CONCLUSÕES

Uma nova proposta de planejamento de trajetória otimizado para robôs foi apresentada, visando proporcionar a utilização mais efetiva do potencial destes equipamentos cada vez mais empregados e aplicados em processos industriais. Além disso, este trabalho visa difundir o método da Busca Tabu, ainda pouco explorado, como ferramenta na resolução de problemas comuns ao meio industrial. Assim, foram usadas duas abordagens heurísticas para resolver o problema proposto, usando o *simulated annealing* e a busca tabu. Com isso, foram feitos testes para cada uma das abordagens, visando a comparação entre as abordagens. Os testes foram executados considerando-se duas situações de solda distintas: cordão linear e circular. Para comparar as abordagens, adotou-se a Análise de Variância (ANOVA), que é um tratamento estatístico específico para este fim.

Nos resultados apresentados, pode-se observar que o valor mínimo encontrado para a função objetivo em cada um dos experimentos é significativamente menor que a média dos valores encontrados. Esta é uma indicação da complexidade deste problema, devido à multiplicidade de soluções viáveis, bem como à existência de inúmeros pontos de ótimo local.

Na comparação entre os valores obtidos para a função objetivo em cada uma das abordagens de busca tabu, ficou claro que, para as quatro condições de parâmetros testadas, não é possível rejeitar a hipótese de igualdade entre os experimentos. Isto pode ser observado tanto para o caso de cordão de solda linear como circular. Da mesma forma, para o número de iterações, tanto para o caso de cordão de solda linear como circular, também não foi possível rejeitar a hipótese de igualdade entre os experimentos.

Para o algoritmo de *simulated annealing*, tanto para o caso de cordão de solda linear como circular, a hipótese de igualdade entre os oito experimentos foi rejeitada, identificando-se uma interação entre os três fatores. Ou seja, identificou-se que o valor da função objetivo tenderá a ser menor quando a taxa de resfriamento (fator **A**) é 0,999, a temperatura inicial (fator **B**) é 100 e a amplitude (fator **C**) é 10. No entanto, para o número de iterações, não foi possível rejeitar a hipótese de igualdade entre os experimentos, tanto para o caso de cordão de solda linear como circular.

Na comparação entre as abordagens de busca tabu e *simulated annealing*, ficou claro, para os dois tipos de cordão de solda, que estatisticamente a busca tabu teve desempenho superior ao *simulated annealing*, ainda que os valores mínimos nos

experimentos indiquem que o *simulated annealing* tenha sido capaz de atingir resultados próximos dos obtidos com a busca tabu.

Não foi possível testar as duas abordagens propostas a alguma abordagem da literatura, já que os artigos relacionados a este problema foram muito pobres na apresentação de dados e também dos parâmetros usados. Assim, como trabalhos futuros, propõe-se a implementação de algoritmos genéticos e, eventualmente, outras abordagens heurísticas. O presente trabalho focou a determinação da trajetória ótima, o que permitirá que futuramente este trabalho seja estendido para a definição de *set-points* do problema de controle de robôs. Além disso, pretende-se testar esta abordagem em problemas reais.

REFERÊNCIAS

ABB. **Manual do operador – IRC5 com flexpendant.** 2004.

ABB. Descrição de produtos. Disponível em <http://www.abb.com.br/product/seitp327/5AA9D9DC6BC1C77CC1256F11004C80E4.aspx>
Acesso em: 04 Novembro de 2008.

ÇAKIR, M., BUTUN, E., KAYMAN, Y. **Effects of genetic algorithm parameters on trajectory planning for 6-DOF industrial robots.** Industrial Robot, Volume 33, issue 3, 2006, p 205-215.

CHEN, M., ZALZALA, A. **A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration.** Journal of Robotics Systems 14, 1997, p 529-544.

CRAIG, J. J. **Introduction to Robotics.** Ed. Addison-Wesley Publishing, Inc., 1989.

FU, K. S., GONZALEZ, R. C., LEE, C. S. G. **Robotics: control, sensing, vision and intelligence.** Ed. Mcgraw-Hill College, 1987, New York, NY, USA.

GLOVER, F., LAGUNA, M. **Tabu search.** Ed. Kluwer Academic Publishers, 1997, Boston, USA.

GARG, D. P., KUMAR, M. **Optimization techniques applied to multiple manipulators for path planning and torque minimization.** Engineering applications of artificial intelligence, Volume 15, Número 3, Junho de 2002 , pp. 241-252.

GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning.** Ed. Addison Wesley, 1989.

GROOVER, M. P. **Automation, Production, Systems, and Computer-Integrated Manufacturing.** Ed. Prentice Hall, Inc., 1987

HOLLAND, J. H. **Adaptation in natural and artificial systems.** Ed. MIT Press, Cambridge, MA, USA, 1992

LAUS, L.P. **Apostila da disciplina de robótica**. UTFPR, 2006. Disponível em <<http://www.dexter.damec.cefetpr.br/~laus/arquivos/introduc2.pdf>> Acesso em: 12 setembro de 2007.

MICHALEWICZ, Z., FOGEL, D.B. **How to solve it: modern heuristics**. Ed. Springer Verlag, 2000, Berlim, Germany,

PIRES, E., MACHADO, J. **A trajectory planner for manipulators using genetic algorithms**. Proceedings of the IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, 1999, p. 163-168.

REEVES, C.R. **Modern Heuristic Techniques for Combinatorial Problems**. Ed. McGraw Hill, 1995, New York, USA.

RODRIGUES, L. C. A. **Notas de aula**. Disciplina Métodos Heurísticos (PMN 15), 2006.

ROMANO, V. F. **Robótica industrial: aplicação na indústria de manufatura e de processos**. Ed. Edgar Blucher Ltda, 2002, São Paulo, Brasil, 256 pp.

SANTOS, W. E. **Apostila robôs industriais**. 2001. Disponível em: <<http://www.controle.eng.br/textos.htm>> Acesso em: 15 Agosto de 2007.

SCIAVICCO, L., SICILIANO, B. **Modeling and control of robot manipulators**. Ed. McGraw Hill, 1996, Naples.

SMITH, V. J. R., OSMAN, I. H., REEVES, C. R., SMITH, G.D. **Modern heuristic search methods**. Ed. John Wiley & Sons Ltd, 1996, West Sussex, England, 294 pp.

WU, L., CUI, K., CHEN, S.B. **Redundancy coordination of multiple robotic devices for welding through genetic algorithm**. National Key Laboratory of Advanced Welding Production Technology, Harbin Institute of Technology, Harbin, Hei Longjiang (People's Republic of China), Robotica Volume 18, issue 6, 2000, p.669-676.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)