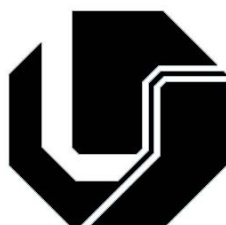


WERLEY ROCHERTER BORGES FERREIRA

**PLANEJAMENTO DE TRAJETÓRIAS ROBÓTICAS UTILIZANDO
B-SPLINES**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA MECÂNICA

2011

WERLEY ROCHERTER BORGES FERREIRA

**PLANEJAMENTO DE TRAJETÓRIAS ROBÓTICAS UTILIZANDO
B-SPLINES**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Sólidos e Vibrações.

Orientador: João Carlos Mendes Carvalho

UBEERLÂNDIA – MG

2011

AGRADECIMENTOS

A forma mais simples de incentivar uma pessoa a progredir é dando exemplos. Agradeço à Veridiana por ser para mim um exemplo de irmã, mãe e profissional. Assim como o Ricardo por ser exemplo de amigo, pai e profissional.

Agradeço ao João Carlos pela orientação ao longo de todos esses anos. Sendo paciente desde a minha iniciação científica à conclusão desta dissertação. Além de orientador é um dos exemplos que tento seguir.

Ao CNPq pelo apoio financeiro.

À Faculdade de Engenharia Mecânica.

Agradeço todos aqueles que ajudaram ao longo deste mestrado.

FERREIRA, W. R. B. **Planejamento de Trajetórias Robóticas Utilizando B-splines**. 2011. 122 f. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

Planejamento de trajetórias robóticas consiste em especificar o caminho que o elemento terminal do robô deve passar para realizar uma determinada tarefa. O planejamento de trajetórias pode ser realizado no domínio da tarefa ou no domínio das articulações. Seja qual for o domínio que as trajetórias são planejadas, elas devem ser definidas por uma função de ajuste de curva, tais como aproximação ou interpolação dos pontos que as definem. Muitos estudos de ajuste de curva têm sido realizados, procurando obter curvas suaves, de forma que a curva de referência promova o movimento do robô o mais suave possível. A utilização de curvas suaves em trajetórias robóticas evita vibrações do robô ao longo da trajetória, desgaste dos atuadores e erro de “tracking”. Para que uma trajetória seja suave ela deve possuir, pelo menos, continuidade de posição, velocidade e aceleração. Curvas como polinomiais, Hermite por partes e splines são utilizadas frequentemente em trajetórias de sistemas multi-eixos. Para trajetórias multipontos a spline é a curva comumente utilizada, porém se um ponto desta curva for alterado o perfil de toda a curva se altera. Para evitar este tipo de problema, alguns autores utilizam B-spline, pois ela possui controle local, isto é, com a mudança de um ponto de controle, a curva se altera somente em uma determinada região em função de seu grau de continuidade. Além de controle local, este tipo de curva possui comportamento previsível porque ela sempre estará contida dentro do polígono formado pelos seus pontos de controle. Esta característica da curva é denominada propriedade do fecho convexo. Nesta dissertação, é realizado o estudo de curvas utilizadas em planejamento de trajetórias robóticas com maior ênfase em B-spline e NURBS devido sua grande flexibilidade. Assim, a partir das curvas estudadas foi realizada uma análise do tipo de curva utilizada pelo robô industrial Motoman HP6. Com o modelo cinemático do robô Motoman HP6 foi proposto o planejamento de

trajetórias utilizando B-spline, pois devido à propriedade do fecho convexo, simulações mostram que a trajetória no domínio das articulações possui comportamento previsível.

FERREIRA, W. R. B. **Robotics Trajectory Planning Using B-spline** 2011. 122 f.
Master Thesis, Universidade Federal de Uberlândia, Uberlândia.

ABSTRACT

Robot trajectory planning is the specification of the path the terminal element of the robot should move to accomplish a certain task. The trajectory planning can be done in the task domain or in the joints domain. For any domains that the trajectories are planned, it should be defined by a curve fitting function, such as approximation or interpolation of the points that define them. Many studies of curve fitting have been accomplished, in order to obtain smoother curves, so that the reference curve promoting the movement of the robot as smoothest as possible. The use of smooth curves in robotic avoid vibration of the robot along the path, wear of actuators and tracking error. For a trajectory be smooth, it must have at least position, velocity and acceleration continuity. Curves like polynomial, piecewise Hermite and splines are often used in trajectories of multi-axes machines. For the trajectories multipoint spline curve is commonly used, but if a point of this curve is changed, the entire profile of the curve is changed. To avoid this problem, some authors have been used B-spline, because it has local control, i.e., with the change of a control point, the curve changes only in a given region according to their continuity degree. Moreover, this type of curve has predictable behavior because it will always be contained within the polygon formed by its control points. This characteristic curve is called the convex hull property. In this master thesis is done the study of curves for robotic trajectory planning with concerning with B-spline and NURBS due to its great flexibility. Thus, from the curves study, was analyzed the type of curve used by industrial robot Motoman HP6. With the kinematic model of the robot Motoman HP6, it was proposed a trajectory planning using B-spline, since due to the convex hull property, simulations show that the trajectory in the joints domain has predictable behavior.

Keywords: Trajectory Planning. Industrial Robo. Curve fitting. B-splines.

LISTA DE FIGURAS

Figura 2.1: Cam Hammer.	6
Figura 2.2: Trajetória do seguidor em função da posição angular da came; b) Conjunto came seguidor, (SCARAMBONI, 2003).	7
Figura 2.3: Came duplo para movimento no plano, adaptado de (ROTHBART, 2004).	7
Figura 2.4: Primeiro robô Unimate instalado na planta industrial da <i>General Motors</i> , 10	
Figura 2.5: (a) Representação de uma estrutura de cadeia cinemática aberta; (b) Robô industrial ABB.	11
Figura 2.6: (a) Representação de uma estrutura de cadeia cinemática fechada; (b) Robô paralelo <i>FlexPiker</i> ABB.	11
Figura 3.1: a) Interpolação; b) Aproximação.	22
Figura 3.2: Interpolação polinomial realizada para cinco pontos: polinômio do quarto grau.	26
Figura 3.3: Interpolação polinomial realizada para sete pontos: polinômio do sexto grau.	27
Figura 3.4: Interpolação polinomial realizada para nove pontos: polinômio do oitavo grau.	27
Figura 3.5: Interpolação polinomial realizada para onze pontos: polinômio do décimo grau.	27
Figura 3. 6: Interpolação polinomial realizada para treze pontos: polinômio do décimo segundo grau.	28
Figura 3.7: Erro devido ao truncamento em uma interpolação polinomial de 11 pontos.	29
Figura 3. 8: (a) Curva formada por dois segmentos com continuidade G^0 no ponto de união. (b) Curva formada por dois segmentos com continuidade G^1 no ponto de união.	30
Figura 3.9: Comparação entre a continuidade G^1 e a $C,^1$ (ROGERS, 2001).	31
Figura 3.10: Interpolação linear por partes.	32
Figura 3.11: Interpolação quadrática por partes.	32
Figura 3. 12: Interpolação cúbica por partes.	33
Figura 3. 13: Interpolação cúbica de Hermite por partes.	34

Figura 3.14: Interpolação por spline.....	34
Figura 3.15: Interpolação com spline cúbica.....	37
Figura 3.16: Interpolação com spline quadrática. Interpolação com os pontos originais (linha contínua), interpolação com a mudança de do sétimo ponto P (linha tracejada).....	37
Figura 3.17: B-spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto P_4	38
Figura 3.18: Spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto P_4	38
Figura 3.19: Funções base de grau zero, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).....	41
Figura 3.20: Funções base do primeiro grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).....	42
Figura 3.21: : Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).....	43
Figura 3.22: Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).....	46
Figura 3.23: B-spline cúbica com $U = \{0,..., 0, 1,..., 1\}$, i.e., Curva de Bézier.....	49
Figura 3.24: Propriedade do fecho convexo, (ROGERS, 2001).....	50
Figura 3.25: Controle local da B-spline.....	51
Figura 3. 26: B-spline do quarto grau.....	52
Figura 3.27: Primeira derivada da curva em relação a u , (a) componente em x , (b) componente e y	52
Figura 3.28: Terceira derivada da curva em relação a u , (a) componente em x , (b) componente e y	53
Figura 3.29: Interpolação por B-spline cúbica usando parametrização por comprimento de corda e vetor nós obtido pelo método de <i>averaging</i>	56
Figura 3.30: Interpolação com diferentes metodologias de parametrização do vetor U	57
Figura 3.31: Interpolação com diferentes metodologias de obtenção do vetor U	58
Figura 3.32: Interpolação com B-spline cúbica com derivadas nos extremos igual a zero.....	60
Figura 3.33: Influência do peso no comportamento da curva.....	62

Figura 4.1: Espaço de trabalho do robô Motoman HP6 com suas cotas em <i>mm</i> (MOTOMAN, 2009).	66
Figura 4.2: Eixos de movimentação do Motoman HP6 (MOTOMAN, 2009).	67
Figura 4.3: Diagrama unifilar do robô HP6 com seus respectivos referenciais.	69
Figura 4. 4: Trajetória planejada no domínio das articulações por spline cúbica com o vetor tempo igualmente espaçado.....	73
Figura 4. 5: Trajetória planejada no domínio das articulações por spline cúbica com o vetor tempo proporcional ao comprimento de corda.....	74
Figura 4.6: Planejamento de trajetórias para a articulação q_1	77
Figura 4.7: Planejamento de trajetórias para a articulação q_2	77
Figura 4.8: Planejamento de trajetórias para a articulação q_3	78
Figura 4. 9: Planejamento de trajetórias para a articulação q_4	78
Figura 4. 10: Planejamento de trajetórias para a articulação q_5	79
Figura 4. 11: Planejamento de trajetórias para a articulação q_6	79
Figura 4. 12: Montagem do experimento.	80
Figura 4.13: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline quadrática no domínio das articulações (linha tracejada).	81
Figura 4.14: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline cúbica no domínio das articulações (linha tracejada).	81
Figura 4.15: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline quadrática no domínio das articulações (linha tracejada).	82
Figura 4.16: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline cúbica no domínio das articulações (linha tracejada).	82
Figura 4.17: Planejamento de trajetória no domínio da tarefa com diferentes métodos de interpolação.....	83
Figura 4.18: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline cúbica no domínio das articulações (linha tracejada).	84
Figura 4.19: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline quadrática no domínio das articulações (linha tracejada).	84
Figura 4.20: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline cúbica no domínio das articulações (linha tracejada).	85
Figura 4.21: Perfil “S” obtido por uma B-spline quadrática.....	86

Figura 4.22: Curva com três segmentos em que o primeiro e o último são definidos por parábolas.....	87
Figura 4.23: Curva com cinco segmentos em que o primeiro e o último são definidos por parábolas.....	88
Figura 4.24: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).....	90
Figura 4.25: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).....	90
Figura 4.26: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).....	91
Figura 5.1: Simulação trajetórias no espaço das articulações e da tarefa.	96
Figura 5.2: Desvio entre a trajetória no espaço das articulações e da tarefa.	96
Figura 5.3: Erro de orientação do elemento terminal.	97
Figura 5.4: Trajetória tridimensional.	98
Figura 5.5: Projeção da trajetória tridimensional no plano XY.....	98
Figura 5. 6: Projeção da trajetória tridimensional no plano XZ.	99
Figura 5.7: Projeção da trajetória tridimensional no plano YZ.	99
Figura 5.8: Erro absoluto entre a trajetória no espaço da tarefa e a no espaço das articulações.	100
Figura 5. 9: Desvio entre a orientação do elemento terminal e a orientação desejada.	100
Figura 5.10: Orientação do elemento terminal ao longo da trajetória.	101
Figura 5.11: Trajetória tridimensional do exemplo 2 (Fig. 5.4) com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$	102
Figura 5. 12: Projeção da trajetória tridimensional no plano XY com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$	102
Figura 5.13: Projeção da trajetória tridimensional no plano XZ com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$	103
Figura 5.14: Projeção da trajetória tridimensional no plano YZ com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$	103
Figura 5. 15: Erro absoluto entre a trajetória no espaço da tarefa e a no espaço das articulações.	104

Figura 5.16: Desvio entre a orientação do elemento terminal e a orientação desejada.

..... 104

LISTA DE SÍMBOLOS

\bar{u}_k	Parâmetro de interpolação da B-spline.
$N_{i,p}^{(k)}$	k -ésima derivada da função de base da B-spline.
\bar{U}	Vetor contendo todos os \bar{u}_k .
\vec{v}	Vetor da direção do elemento terminal.
$\nabla^n y_i$	Diferença dividida de ordem n .
“S”	Forma do perfil de uma curva.
${}^A p$	vetor que descreve as coordenadas da posição da origem do referencial B em relação ao referencial A .
${}^A R_B$	Matriz que representa a orientação do referencial A em relação ao B .
$C(u)$	Curva paramétrica.
C^k	Continuidade paramétrica de ordem k .
$C_n(x)$	Polinômio de grau n .
d	Comprimento total de corda.
$D^{(k)}(x_i)$	k -ésima derivada no ponto de interpolação.
d_c	Somatório das raízes quadradas dos comprimentos de corda.
G^k	Continuidade geométrica de ordem k .
$N_{i,p}(u)$	Função de base da B-spline de grau p no vão i .
p	Grau da curva B-spline.

P_i	Pontos que definem uma curva, onde $(i = 1, \dots, n)$.
q	Coordenada articular.
Q_k	Pontos de interpolação de uma B-spline.
$R_{i,p}(u)$	Função de base da NURBS de grau p no vão i .
$S_i(x)$	i -ésimo segmento de uma spline.
T	Tempo total do trajeto de um robô.
t	Tempo.
u	Parâmetro de uma curva paramétrica.
U	Vetor dos nós da B-spline.
ω_i	Peso associado ao ponto P_i em uma curva NURBS.
ψ	Rotação em torno do eixo x.
θ	Rotação em torno do eixo z.
φ	Rotação em torno do eixo y.

SUMÁRIO

CAPÍTULO I	1
INTRODUÇÃO.....	1
CAPÍTULO II.....	5
REVISÃO BIBLIOGRÁFICA.....	5
2.1. Introdução.....	5
2.2. Trajetórias realizadas por came-seguidor.....	6
2.3. Cames eletrônicos.....	8
2.4. Planejamento de trajetória realizada por sistemas multi-eixos.....	8
2.4.1. Robôs industriais	9
2.4.2. Robôs.....	12
2.5. Planejamento de trajetórias robóticas	12
2.5.1. Trajetória definida em tempo ótimo	13
2.5.2. Desvio de obstáculos	14
2.5.3. Realização de trabalho com trajetórias específicas	14
2.6. Planejamento de trajetória no espaço da tarefa	15
2.7. Planejamento de trajetória no espaço das articulações.....	15
2.8. Curvas utilizadas em planejamento de trajetórias robóticas.....	16
2.8.1. Segmentos lineares concordantes por polinomiais.....	16
2.8.2. Trajetórias ponto a ponto definida por polinomiais.....	17
2.8.3. Trajetórias definidas por spline	19
2.8.4. Trajetórias definidas por B-spline e NURBS	20
2.9. Conclusão	21
CAPÍTULO III	22

AJUSTE DE CURVA	22
3.1. Introdução	22
3.2. Interpolação polinomial	23
3.2.1. Coeficientes Indeterminados	23
3.2.2. Interpolação de Lagrange	24
3.2.3. Interpolação de Newton	24
3.2.4. Interpolação de Hermite	25
3.2.5. Problemas numéricos com polinômios de alta ordem	26
3.3. Polinomial por Partes	29
3.3.1. Continuidade	29
3.3.2. Interpolação linear por partes	31
3.3.3. Interpolação quadrática por partes	32
3.3.4. Interpolação cúbica por partes	32
3.3.5. Interpolação de Hermite por partes	33
3.4. Spline	34
3.5. B-spline	37
3.5.1. Funções de base da B-spline	39
3.5.2. Derivadas das funções de base B-spline	43
3.5.3. Propriedades importantes das funções de base B-spline	44
3.5.4. Curvas B-spline	48
3.5.5. Propriedades das curvas B-splines	49
3.5.6. Derivada da curva B-spline	51
3.5.7. Interpolação com B-spline	54
3.5.8. Interpolação com especificação das derivadas nos extremos	58
3.5.9. Alteração de curvas B-spline	60
3.6. NURBS (Non Uniform Rational B-spline)	60

3.7. Conclusão	63
CAPÍTULO IV	65
ANÁLISE DA TRAJETÓRIA DESCRITA POR UM ROBÔ INDUSTRIAL	65
4.1. Introdução	65
4.2. Robô Motoman HP 6	66
4.3. Modelo cinemático direto	68
4.4. Modelo cinemático inverso	70
4.5. Planejamento de trajetórias para o robô HP6	72
4.5.1. Planejamento de trajetórias no domínio das articulações	72
4.5.2. Simulação numérica e comparação com os resultados experimentais	75
4.5.3. Planejamento de trajetórias no domínio da tarefa	83
4.5.4. Interpolação realizada pelo Motoman HP6	85
4.6. Erro de seguimento	91
4.7. Conclusão	92
CAPÍTULO V	93
PLANEJAMENTO DE TRAJETÓRIAS USANDO B-SPLINE PARA O ROBÔ MOTOMAN HP6	93
5.1. Introdução	93
5.2. Exemplo 1 – Trajetória no plano	95
5.3. Exemplo 2 – Trajetória espacial	97
5.4. Exemplo 3 – Inclusão de pontos para a redução do erro	101
5.5. Conclusão	105
CAPÍTULO VI	106
CONCLUSÕES	106
6.1. Conclusões	106
6.2. Sugestões para trabalhos futuros	107
REFERÊNCIAS BOBLIOGRÁFICAS	108

ANEXOS 117

CAPÍTULO I

INTRODUÇÃO

Ao longo da história o homem vem aprimorando suas técnicas e habilidades para executar tarefas com maior rapidez e melhor qualidade. Essa evolução pode ser notada desde os períodos Pré-Históricos como a Idade da Pedra, onde os utensílios agrícolas e armas eram feitos de pedra, até a evolução para a Idade dos Metais, em que o ser humano começava a dominar a técnica de fundição, na qual foi possível construir ferramentas mais eficientes e resistentes.

Neste processo evolutivo a tecnologia passou por várias transformações e foram desenvolvidas máquinas que permitiram o aumento da produtividade, podendo ser citadas a máquina a vapor e a máquina de tear após a Revolução Industrial do século XVIII. Com isso o mercado tornou-se mais competitivo e a exigência de produtos de qualidade produzidos em um curto período de tempo levou a processos de automação cada vez mais sofisticados incluindo computadores, sensores e atuadores de alta velocidade.

No ápice da automação industrial surgiu o robô que, diferentemente das outras máquinas automáticas, pode ser reprogramado para executar diversas tarefas sem alterar a planta da empresa. Um robô é um sistema que pode ser pneumático, hidráulico ou eletromecânico projetado a fim de executar tarefas com maior rapidez e precisão que os seres humanos. Eles podem substituir o trabalho humano em ambientes perigosos e em tarefas repetitivas e cansativas como corte de chapas, pintura, solda, colagem e manipulação de objetos.

Para cada tarefa que o robô deve realizar, a trajetória do seu elemento terminal precisa ser definida em sua programação. Esta trajetória pode ser descrita em termos das coordenadas

articulares (espaço das articulações) ou em função das coordenadas que definem o trajeto do elemento terminal no espaço Cartesiano (espaço da tarefa). O planejamento de tarefas no espaço das articulações é aplicado em ambientes de trabalho onde não existem obstáculos e é desejável que o movimento ocorra o mais rápido possível. Já o planejamento no espaço da tarefa é muito utilizado em operações onde o trajeto especificado deve ser seguido corretamente.

Diversos autores (FANG et al., 1998) e (MACFARLANE, 2001) afirmam que as trajetórias definidas no espaço das articulações não mantêm a orientação do elemento terminal do robô. Com isso, para operações em que é necessário uma orientação constante, como transporte de fluidos, a trajetória deve ser definida no espaço da tarefa. Isso se deve à relação matemática que descreve o comportamento do elemento terminal em função da movimentação das articulações/atuadores.

Em geral, para que uma trajetória seja seguida, o operador deve posicionar o elemento terminal do robô nos pontos que a definem, onde cada ponto é armazenado na memória do controlador. A partir dos pontos armazenados, existem duas formas de gerar a trajetória: uma é a trajetória planejada no domínio da tarefa, a qual utiliza uma metodologia de ajuste de curva para determinar a trajetória do robô no espaço Cartesiano. A outra é no espaço das articulações, que realiza o ajuste de curva nas coordenadas articulares do robô.

O ajuste de curva pode ser por aproximação ou interpolação. Seja qual for a forma de ajuste de curva utilizada, é sempre desejável que as trajetórias robóticas sejam as mais suaves possíveis, isto é, deve ter continuidade de posição, velocidade e aceleração. Muitos autores consideram também a continuidade do jerk (derivada da aceleração) para garantir uma trajetória ainda mais suave.

As trajetórias de um robô podem ser definidas ponto a ponto ou de forma contínua. Em trajetórias do tipo ponto a ponto, a cada dois pontos define-se uma curva de interpolação. Neste procedimento, para cada ponto, o robô parte do repouso e para no próximo ponto. Em trajetórias contínuas, todos os pontos ‘ensinados’ ao robô são utilizados para definir uma curva a ser seguida.

Para gerar trajetórias contínuas pode-se utilizar uma polinomial única porém, esse procedimento não é recomendado para trajetórias com muitos pontos, visto que polinomiais de alta ordem possuem comportamento indesejado devido a problemas numéricos como instabilidade e erros de truncamento.

O procedimento mais adequado para evitar polinomiais de alta ordem em planejamento de trajetórias robóticas é empregar polinomiais por partes. Em planejamento de trajetórias robóticas a spline cúbica, que consiste em uma polinomial por partes, é muito utilizada porque possui continuidade de posição, velocidade e aceleração. Porém, no primeiro e último ponto da trajetória a spline cúbica não possui continuidade de aceleração que, como consequência, essa trajetória não possui uma partida e chegada suaves.

Para evitar o problema da descontinuidade da aceleração da spline cúbica, os segmentos extremos podem ser modificados por um polinômio do quinto grau. Assim, a curva resultante terá continuidade de posição, velocidade e aceleração em todo o trajeto.

Outra alternativa para obter curvas suaves é a utilização de B-spline. Em uma B-spline pode-se definir o grau de continuidade para cada trajetória. Diferentemente da spline cúbica, com a B-spline cúbica é possível definir uma curva com continuidade nos extremos de posição, velocidade e aceleração. Além disso, a B-spline possui a propriedade de controle local, isso é, com a alteração de um ponto de controle, a curva mudará somente na região próxima ao ponto alterado.

Curva B-spline é um caso particular de curva NURBS (*Non Uniform Rational B-spline*). A NURBS é um ajustador de aproximação que possui grande flexibilidade, podendo ser alterada localmente com a mudança de um ponto de controle, ou do peso associado a cada ponto de controle, ou do vetor de nós, que são utilizados para calcular as funções de base. Se todos os pesos da NURBS forem iguais, a curva resultante é uma B-spline.

Diversos trabalhos tratam da aplicação das splines, sobretudo a spline cúbica, no planejamento de trajetórias robóticas. Sua grande vantagem consiste na relativa facilidade de resolução matemática, facilitando inclusive sua aplicação em controladores industriais. No entanto, caso um ponto do trajeto seja alterado, todo o sistema matemático deve ser resolvido novamente alterando todo o perfil da curva.

Nesta dissertação é apresentado um estudo da aplicação das curvas B-splines no planejamento de trajetórias robóticas, visto sua grande vantagem no controle local da curva. O estudo consiste no planejamento no espaço das articulações e da tarefa, tanto em termos da posição como de orientação do elemento terminal.

Além do estudo teórico, são realizadas simulações aplicadas a um robô industrial de seis graus de liberdade.

Com a previsão da aplicação futura do planejamento de trajetórias utilizando B-spline no citado robô industrial, foi realizado um estudo teórico-experimental de modo a verificar que

tipo de planejamento que seu controlador realiza no modo denominado pelo fabricante de “spline quadrática”.

Para que os objetivos deste trabalho sejam atingidos, no Capítulo II é apresentada uma revisão sobre o planejamento de trajetórias robóticas; no Capítulo III é feito um estudo das metodologias de ajuste de curva, com a ênfase à B-spline e NURBS, visando sua aplicação; no Capítulo IV simulações e experimentos são realizados para permitir determinar o tipo de curva utilizada nas trajetórias do robô Motoman HP6 disponível no Laboratório de Automação e Robótica; no Capítulo V são realizadas simulações de trajetórias aplicando B-spline no robô Motoman HP6; finalmente, no Capítulo VI, as conclusões e considerações para trabalhos futuros.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

2.1. Introdução

Trajetória é o percurso que um móvel descreve no tempo, ou seja, é a relação trajeto-tempo. Na robótica, é comum especificar a trajetória que o elemento terminal do robô deve descrever, que é conhecido como planejamento de trajetória robótica.

Para qualquer tarefa executada automaticamente é necessário planejar o movimento dos dispositivos do sistema. Isso é evidente tanto nas máquinas mais simples, como nas complexas. Por exemplo, nas máquinas de cortar papel ou tecido as quais devem ter um sincronismo entre a esteira e as lâminas. Caso contrário, os pedaços cortados não terão o padrão exigido. Nas máquinas multieixos a necessidade do planejamento de seus movimentos é evidente em função de seus vários eixos programáveis.

Se a trajetória de um robô não for planejada corretamente, há o risco dele não realizar a tarefa da forma desejada. Além disso, existe a possibilidade de uma colisão entre o robô e seu ambiente de trabalho.

Neste capítulo é apresentada uma revisão geral do estudo de trajetórias, desde seu início com o uso de cames, a trajetórias complexas de máquinas multi-eixos.

2.2. Trajetórias realizadas por came-seguidor

Mecanismos com utilização de cames têm uma longa história. Embora alguns autores remontam sua origem no Paleolítico, certamente Leonardo da Vinci pode ser considerado como um dos pioneiros em projetos ‘modernos’ de cames (ROTHBART, 2004). Leonardo da Vinci deixou documentado alguns de seu projetos, sendo um deles o martelo automático, ou *cam hammer*, mostrado na Fig. 2.1. No projeto do martelo automático, conforme a manivela gira, o martelo descreve um trajeto ascendente definido por um setor circular, até que em uma determinada posição o martelo retorna em queda livre para sua posição de equilíbrio. Este mecanismo simula o movimento da martelada produzido pelo braço humano.

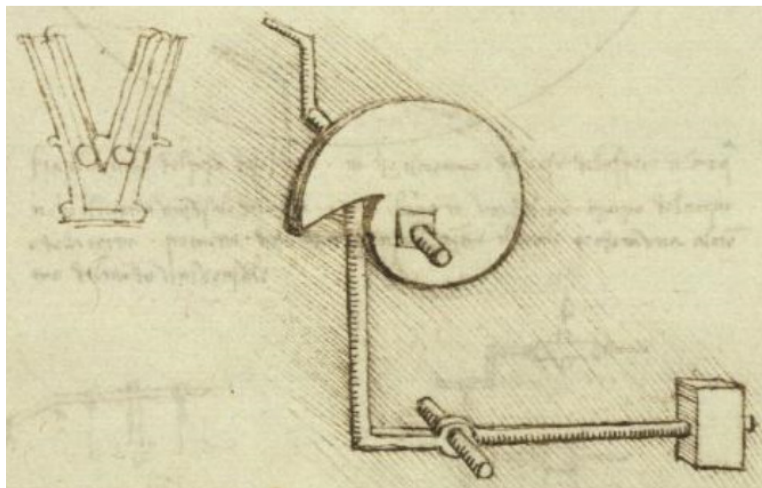


Figura 2.1: Cam Hammer.

<http://dlxs2.library.cornell.edu/cgi/t/text/pagevieweridx?c=kmoddl;cc=kmoddl;rgn=full%20text;idno=kmod020;didno=kmod020;view=image;seq=22;node=kmod020%3A2;page=root;size=s;frm=frameset>

Em muitas máquinas automáticas a trajetória é definida pelo conjunto came-seguidor. O came é um elemento de máquina que determina o movimento do seguidor, assim o came deve ser projetado de acordo com a trajetória necessária para que o seguidor realize a tarefa. Na Figura 2.2 é mostrado o perfil de um came em função do movimento do seguidor.

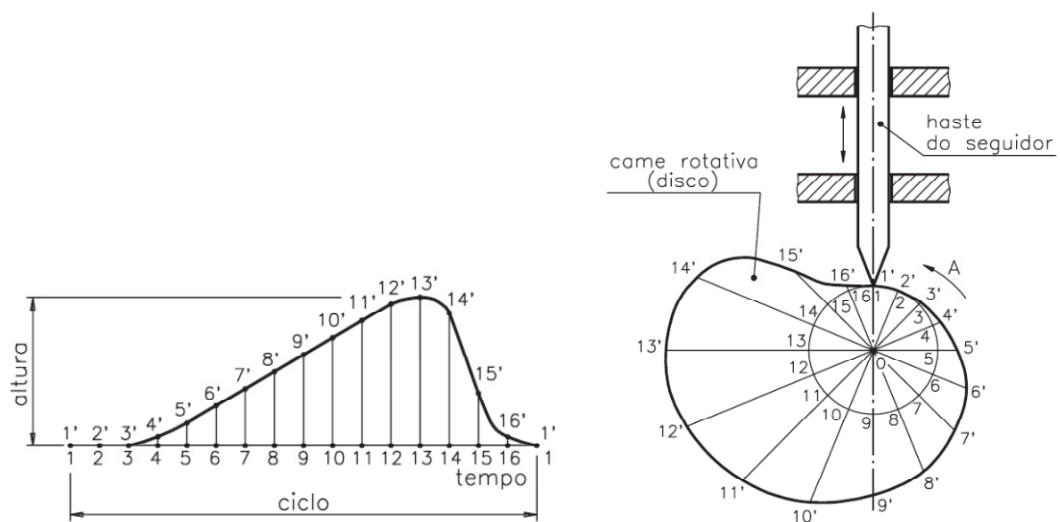


Figura 2.2: Trajetória do seguidor em função da posição angular da came; b) Conjunto came seguidor, (SCARAMBONI, 2003).

Além de trajetórias unidimensionais um sistema pode trabalhar com mais de um conjunto came-seguidor para realizar movimento em um plano, este é o caso do *Double cam*, mostrado na Fig. 2.3.

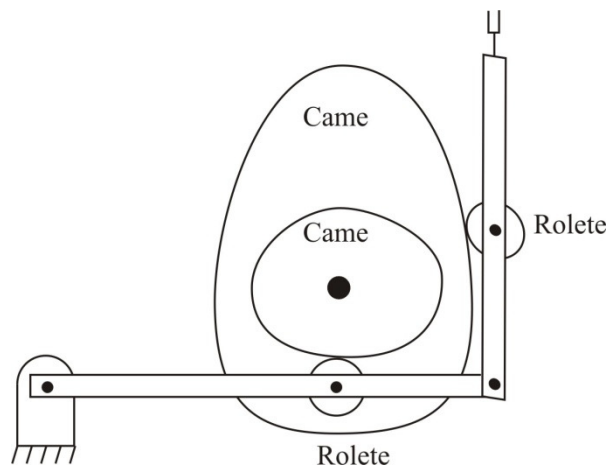


Figura 2.3: Came duplo para movimento no plano, adaptado de (ROTHBART, 2004).

Nas trajetórias utilizando came, se um novo movimento for necessário, um novo came deve ser projetado e confeccionado (WOELFEL, 1999). Isso faz com que as trajetórias com came sejam realizadas em sistemas dedicados sem muita flexibilidade.

O came mecânico é um dispositivo essencial em máquinas automáticas, sendo um elemento de máquina que vem sendo utilizado por muitos anos e, mesmo com o surgimento de novas tecnologias como o came eletrônico, em algumas aplicações o came mecânico

continua sendo mais vantajoso, principalmente pelo seu baixo custo e baixa manutenção quando comparados com os sistemas eletromecânicos.

2.3. Cames eletrônicos

Um dos problemas fundamentais dos sistemas de came mecânico é que uma vez construído o came, caso deseje-se modificar o perfil do movimento do seguidor, um novo came deve ser criado. A flexibilidade pode ser obtida com o “came eletrônico”.

Came eletrônico é um equipamento composto por controlador, driver, memória e servomotor para simular a função do came mecânico. Este sistema possui a característica de flexibilidade de movimentos, alta repetibilidade e pequeno erro de segmento do sinal de controle (*traking*). Mas, para cumprir precisamente a exigência do ajuste de curva, o came eletrônico precisa de muitos parâmetros, tornando complexo o processamento para obter uma trajetória (JIAZHONG; LEI; MING, 2009).

Cada componente do sistema de came eletrônico possui uma função para simular o came mecânico. A memória armazena os dados referentes ao deslocamento (sinal de referência), o controlador comanda o movimento do servomotor com o objetivo de seguir o sinal de referência e o driver amplifica o sinal do controlador para a alimentação do servo. Dessa forma, o planejamento da trajetória de um came eletrônico é menos complicado que o came mecânico, pois ele é realizado por programação, além de ser possível planejar diferentes trajetórias para um único came, pois o sistema seguirá a trajetória que está armazenada na memória (WOELFEL, 1999) e (JIAZHONG; LEI; MING, 2009).

2.4. Planejamento de trajetória realizada por sistemas multi-eixos

Embora no caso de uma máquina composta por vários motores, cada um possa ser programado e controlado de forma independente (controle no espaço da articulação), muitas aplicações exigem uma coordenação entre os diferentes eixos de movimento com a finalidade de obter uma trajetória no espaço da tarefa da máquina. Este é o caso de máquinas ferramenta usadas para cortar, perfurar ou polir uma determinada peça, ou de robôs que devem executar

tarefas tais como solda, manipulação de objetos, colagem, etc (BIAGIOTTI; MELCHIORRI, 2008).

Nestes equipamentos, cada eixo é controlado utilizando funções que definem sua movimentação, ou seja, o trajeto que cada atuador deve descrever, seja de rotação ou de translação, deve ser definido em função do tempo de movimentação e respeitando as restrições cinemáticas e dinâmicas de todo o sistema.

A trajetória planejada é convertida e enviada ao controlador que a “traduz” em um sinal que deve ser seguido pelo atuador. Este sistema é usualmente denominado servomecanismo que, na realidade, consiste em um tipo de came eletrônico mais “inteligente”.

Dessa forma, o planejamento de trajetória consiste não só em planejar a movimentação da ferramenta que vai executar a tarefa, mas também o planejamento da movimentação de cada eixo/atuator da máquina.

Atualmente, o sistema multi-eixo mais flexível, e que pode ser aplicado em diversas atividades do cotidiano ou industrial é, sem dúvida, o robô.

2.4.1. Robôs industriais

De acordo com a ISO (*International Organization for Standardization*), robô industrial é um mecanismo automático, servo-controlado, reprogramável, multifuncional, programável em três ou mais eixos, podendo ser fixo ou móvel para utilização em operações de automação industrial (CECCARELLI, 2004).

Em 1961, George Devol conseguiu a patente de uma máquina reprogramável de aplicação geral denominada de “*Programmed Article Transfer*”. Diferentemente das máquinas da época, que tinham seus movimentos definidos por cames mecânicos, a máquina proposta por Devol, mostrada na Fig. 2.4, podia ser servo-controlada por dispositivos hidráulicos ou elétricos. Isso garantiu que ela fosse rapidamente adaptada para uma grande diversidade de aplicações (DEVOL, 1961). Assim, pode-se dizer que somente a partir de 1961 começaram a surgir os primeiros robôs industriais que se enquadrassem na definição atual da ISO.

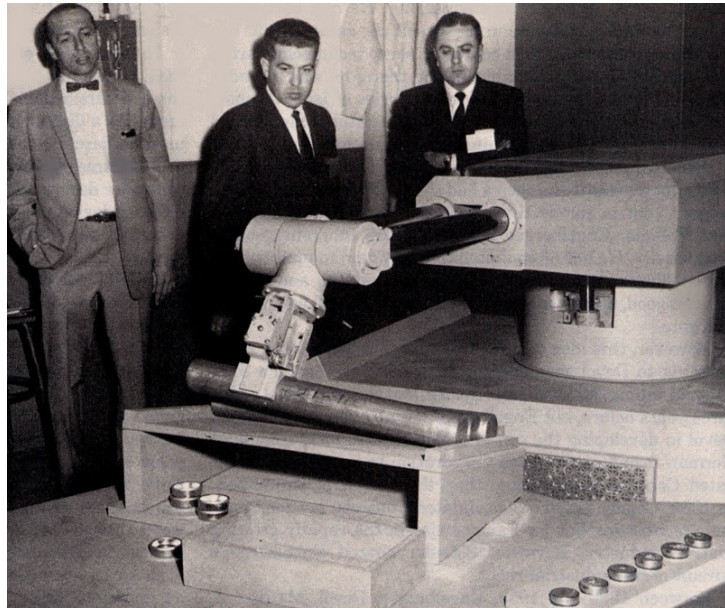


Figura 2.4: Primeiro robô Unimate instalado na planta industrial da *General Motors*, <<http://spectrum.ieee.org/automaton/robotics/industrial-robots/george-devol-a-life-devoted-to-invention-and-robots>> acessado em 3 de outubro de 2011.

Depois do “*Programmed Article Transfer*” foram construídos diferentes tipos de manipuladores. Em função da variedade de formas e tipos de acionamento, várias classificações foram propostas conforme (ELGELBERGER, 1980), (CECCARELLI, 2004) e (TSAI, 1999) . A que tem interesse direto nesse trabalho é a que classifica o robô em função de sua forma topológica podendo ser serial, paralelo ou misto. A estrutura serial, ou cadeia cinemática aberta, ou antropomórfica, é composta por *links* conectados em série por articulações, como representado na Fig. 2.5. Neste caso, todos os atuadores associados a cada articulação são ativos. Esta estrutura é a comumente utilizada em robôs manipuladores industriais.

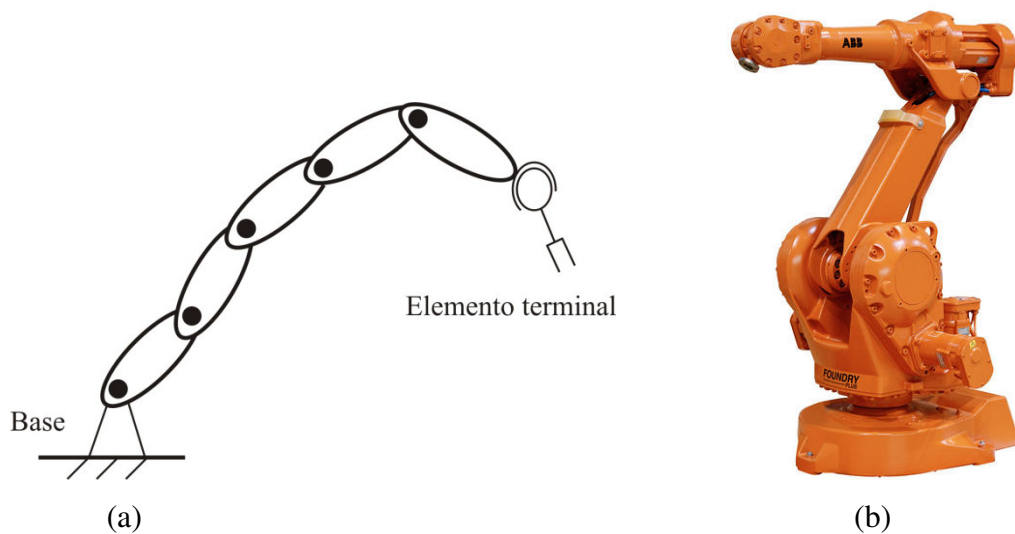


Figura 2.5: (a) Representação de uma estrutura de cadeia cinemática aberta; (b) Robô industrial ABB.

A estrutura paralela, ou estrutura de cadeia cinemática fechada é um manipulador que controla os movimentos do elemento terminal por meio de pelo menos duas cadeias cinemáticas, em que, cada cadeia cinemática sai da base e chega ao elemento terminal como mostrado na Fig. 2.6 (IONESCU, 2003).

A estrutura mista consiste na associação da estrutura serial e a paralela. Embora alguns autores dizem que ela pode acoplar as vantagens das duas estruturas anteriores, pouquíssimos trabalhos tem sido apresentados sobre elas.

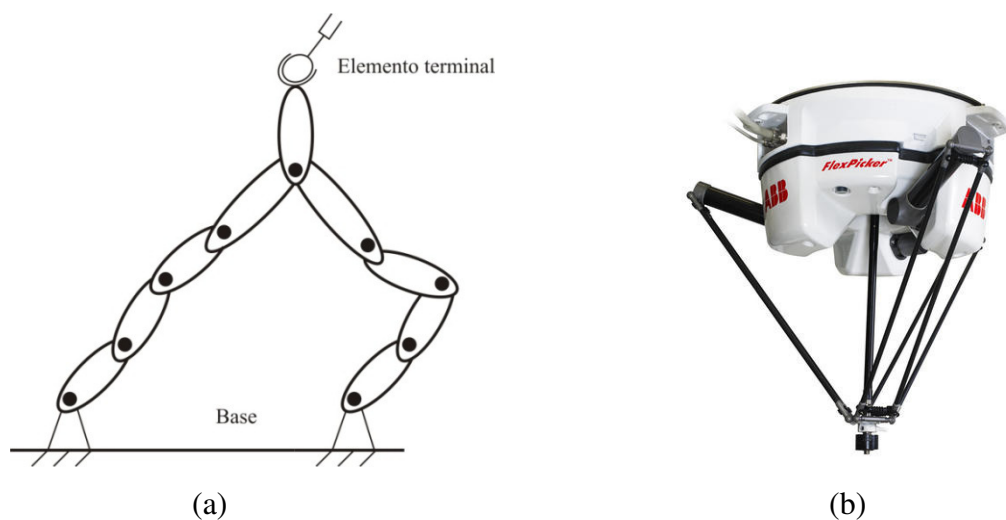


Figura 2.6: (a) Representação de uma estrutura de cadeia cinemática fechada; (b) Robô paralelo *FlexPiker* ABB.

Os robôs industriais têm sido utilizados em diversas aplicações tais como colagem, soldagem, pintura e corte, sendo que para cada uma o caminho e a orientação do elemento terminal, em função do tempo, devem ser especificados com precisão (MACFARLANE, 2001). A especificação do caminho que o robô deve seguir é chamado de planejamento de trajetória.

2.4.2. Robôs móveis

Robôs que desempenham tarefas de locomoção são denominados robôs móveis (IONESCU, 2003). Os robôs móveis podem ser utilizados em diversas aplicações tais como exploração do ambiente, segurança, entretenimento e trabalhos domésticos.

Para que um robô autônomo navegue pelo ambiente, além dos sensores de identificação de obstáculos e de posicionamento, é preciso ter uma referência de trajetória a ser seguida. Geralmente a referência é obtida por meio de uma curva que interpola pontos de controle que definem a trajetória.

2.5. Planejamento de trajetórias robóticas

O planejamento de trajetórias robóticas pode ser realizado no domínio da tarefa ou no domínio das articulações. Trajetórias planejadas no domínio da tarefa são realizadas com a especificação do caminho que o robô deve percorrer corretamente, isto é, a trajetória geralmente, é definida no espaço Cartesiano por uma função geométrica. Mesmo que o trajeto seja definido entre dois pontos, é definida uma função entre eles.

No planejamento de trajetórias no domínio das articulações, as coordenadas articulares são interpoladas para que o robô realize seu trabalho. Neste caso, nem sempre se pode definir uma função entre os valores das coordenadas articulares definidas entre dois instantes. Muitas vezes, a função que define a variação temporal da coordenada articular consiste em uma característica eletromecânica do sistema de motorização da articulação. Assim, no planejamento da trajetória do robô no domínio das articulações, não é possível prever, a priori, o trajeto do elemento terminal no espaço Cartesiano em função das relações matemáticas existentes entre o movimento dos atuadores e do elemento terminal.

Independentemente de como a trajetória seja definida, o seu planejamento é realizado em função do que se deseja, ou seja, aumento da produtividade, desvio de obstáculos ou simplesmente realizar uma determinada tarefa entre duas situações previamente definidas.

2.5.1. Trajetória definida em tempo ótimo

A alta velocidade de operação é um dos requisitos essenciais para minimizar o tempo de ciclo de produção em muitos sistemas de produção automatizados, principalmente nas unidades robotizadas. Assim, o desenvolvimento de um método que obtenha o movimento rápido de robôs tem sido amplamente estudado na área da robótica. No entanto, tal problema é bastante complexo na aplicação típica de robótica industrial, pois envolve uma série de restrições que devem ser satisfeitas. Por exemplo, a trajetória deve satisfazer não só as restrições cinemáticas como deslocamento, velocidade máxima, e outras que normalmente são impostas pelo usuário, mas também as limitações dinâmicas, tais como os torques máximos dos motores e engrenagens de redução, (KIM et al., 2010).

Robôs são usados em muitas indústrias para executar tarefas de pegar e colocar (“pick-and-place”). Um exemplo é a indústria de alimentos, onde um requisito é a capacidade de pegar objetos a partir de uma correia transportadora e colocá-los em embalagens em alta velocidade. Os fabricantes do setor estão procurando usar robôs cada vez mais rápidos, a fim de maximizar a produtividade e reduzir os custos de produção. A velocidade de “pick and place” de robôs industriais tem aumentado continuamente, sendo que velocidades entre 80 e 120 ciclos por minuto, superior ao de um operador humano, estão se tornando comuns. Nessas altas velocidades, a maneira pela qual um movimento do robô é planejado torna-se cada vez mais importante (MASEY et al., 2009).

Com o aumento da velocidade dos robôs, consequentemente há o aumento do consumo de energia, pois a potência requerida pelos atuadores é alta para produzir grandes acelerações. Isto justifica os diversos estudos sobre a otimização temporal aplicados ao planejamento de trajetórias robóticas. Por exemplo, em (SANTOS; STEFFEN; SARAMAGO, 2010), é apresentada uma estratégia de otimização de trajetória que busca o tempo ideal de trajeto e a energia mecânica mínima dos atuadores. O tempo de trajeto e a energia mecânica dos atuadores são considerados em conjunto para formar uma função multiobjetivo a ser minimizada ao longo do processo.

2.5.2. Desvio de obstáculos

Um requisito frequente analisado na determinação da trajetória é o desvio de obstáculos. O problema do desvio de obstáculos em robótica pode ser dividido em três etapas: o mapeamento do ambiente, a determinação da distância entre o manipulador e os demais objetos do ambiente e a decisão de como movimentar o manipulador de forma a evitar da melhor forma possível o contato com os outros objetos (SANTOS, 2007).

O planejamento de trajetória nestes casos pode considerar a presença de obstáculos fixos ou móveis. Dentre os obstáculos móveis, aqueles que têm o percurso previamente conhecido em todos instantes de tempo, juntamente com aqueles que são fixos, permitem a análise preliminar do movimento do robô antes de sua efetiva realização (SANTOS, 2007).

Como exemplo, em (SAMARAGO; STEFFEN, 1997), (SAMARAGO; STEFFEN, 1998), (SAMARAGO; STEFFEN, 1999), (SAMARAGO; STEFFEN, 2000), (SARAMAGO; STEFFEN, 2001), são apresentadas estratégias de otimização de trajetória com a presença de obstáculos, onde a função multiobjetivo a ser minimizada é composta pelo tempo de trajeto; função da energia mecânica dos atuadores, e a função de distância mínima admissível das partes do robô entre os obstáculos para que não houvesse colisão.

2.5.3. Realização de trabalho com trajetórias específicas

Para aplicações como colagem, soldagem, pintura e corte, o caminho e a orientação do elemento terminal devem ser especificados com precisão. Neste caso, a trajetória deve ser completamente especificada por uma função geométrica, ou seja, tanto a posição como a orientação do elemento terminal devem ser definidas (MACFARLANE, 2001).

Independentemente do objetivo e do servomecanismo utilizado, a movimentação suave e contínua é desejável em todas as aplicações.

2.6. Planejamento de trajetória no espaço da tarefa

Em operações em que o trajeto especificado deve ser seguido corretamente é melhor planejar toda a trajetória no espaço da tarefa do que aproximar o trajeto por interpolações entre os pontos no espaço das articulações. Movimentos planejados no espaço da tarefa também são mais previsíveis isto é, é mais fácil para o usuário dizer se o elemento terminal do manipulador vai tocar em algum obstáculo (TAYLOR, 1979).

O planejamento no espaço da tarefa possui uma dificuldade adicional que consiste no planejamento da orientação do elemento terminal ao longo da trajetória. A orientação não pode ser tratada como um vetor, sendo que a suavidade dos ângulos que a definem (em geral os ângulos de Euler) não necessariamente garante um movimento suave do elemento terminal (FANG et al., 1998).

Do ponto de vista tecnológico, a implementação de um sistema de controle de movimento no espaço Cartesiano é muito difícil, principalmente porque a obtenção de uma medição precisa direta da posição do elemento terminal é uma tarefa muito complexa. Assim, em casos práticos de movimento do elemento terminal, as coordenadas Cartesianas são convertidas em coordenadas articulares através da aplicação da cinemática inversa, sendo o controle realizado no espaço das articulações (VISIOLI, 2000).

Em trajetórias onde o trajeto que o robô realiza não é importante, no qual somente a posição inicial e final é de interesse, a trajetória pode ser planejada no domínio das articulações.

2.7. Planejamento de trajetória no espaço das articulações

O método de planejamento de trajetória no espaço das articulações é realizado por etapas. Primeiro, um número de pontos adequado deve ser selecionado no caminho especificado do espaço Cartesiano (espaço da tarefa). Em seguida, esses pontos são transformados em conjuntos de coordenadas articulares. Para cada ponto especificado haverá N coordenadas articulares, onde N é o número de graus de liberdade do robô. Finalmente, cada conjunto de coordenadas articulares deve ser interpolado por algum método de geração de trajetória suave para ser utilizado do comando dos atuadores (ZHENG; FRENG; CHAO, 2002). A quantidade

de pontos definidos em uma trajetória depende da tarefa realizada pelo robô. Em tarefas onde o trajeto intermediário não é importante define-se somente a posição dos pontos que o robô deve atingir.

Movimentos planejados no espaço das articulações atingirão os pontos definidos no espaço da tarefa com melhor acuracidade, visto que erros devido à solução dos modelos cinemático direto e inverso não são introduzidos no sistema. (FISHER; MUJTABA, 1988) verificam que, descrevendo um trajeto com trechos lineares no espaço articular permite um movimento suave e bem comportado do elemento terminal, cujo trajeto é invariável com a velocidade do manipulador. Isto significa que o robô seguirá o mesmo trajeto tanto em alta como em baixa velocidade de operação desde que as restrições cinemáticas e dinâmicas sejam respeitadas.

Um trajeto no espaço das articulações não é intuitivamente previsível. Além disso, para certas aplicações é necessário manter o elemento terminal com uma orientação constante tais como no transporte de fluidos. Isto pode ser facilmente especificado no espaço da tarefa, ao passo que no espaço das articulações não se tem esta garantia. Então, a carga computacional adicional para resolver a cinemática inversa, a cada intervalo de controle, é justificável (MACFARLANE, 2001).

2.8. Curvas utilizadas em planejamento de trajetórias robóticas

No planejamento de trajetórias robóticas seja no domínio da tarefa ou das articulações, a referência de trajetória é uma curva calculada a partir dos pontos ‘ensinados’ ao robô. Muitos perfis de curvas são empregados em trajetórias, sendo apresentados os mais comuns a seguir.

2.8.1. Segmentos lineares concordantes por polinomiais

O perfil de trajetória normalmente utilizado em vários controladores comerciais consiste em segmentos lineares com concordância parabólica (LSPB - *Linear Segments with Parabolic Blends*). Neste método a velocidade possui um perfil trapezoidal, mas a aceleração tem um perfil retangular, o que introduz transientes no movimento, provocando vibrações estruturais

que resulta em deficiência de acompanhamento do trajeto e aumentando o desgaste do atuador (SCIAVICCO; SICILIANO, 1996), (FISHER; MUJTABA, 1988), (BOBROW et al., 1985), (SHIN; MCKAY, 1985), (CASTAIN; PAUL, 1984) e (RED, 2000).

Trajetórias LSPB são geralmente utilizadas em manipuladores industriais e máquinas ferramentas por permitir um movimento rápido e também devido à baixa complexidade computacional. Nestas trajetórias, a aceleração sempre está no seu valor máximo, no zero ou no mínimo para permitir o movimento ótimo no tempo, respeitando os limites de velocidade impostos. Entretanto, este tipo de movimento tem jerk infinito e, como consequência, não é facilmente traçado, resultando em desvios do trajeto previsto.

Alguns autores como (CASTAIN; PAUL, 1984) modificaram o perfil LSPB fazendo com que a aceleração tenha um perfil trapezoidal. Para os movimentos entre os pontos, o perfil de aceleração trapezoidal é integrado duas vezes, assim, o perfil de posição terá duas regiões definidas por polinômios cúbicos e uma por uma parábola. A trajetória obtida é suave, sem movimentos bruscos, e com menor erro de posicionamento.

A fim de obter um perfil de aceleração contínuo na concordância de trajetórias compostas por segmentos de retas, (LLOYD; HAYWARD, 1991) propuseram a interpolação quártica (do quinto grau) de Hermite para o polinômio concordante. Isso implica que são impostas seis restrições, posição, velocidade e aceleração no início e final da concordância.

2.8.2. Trajetórias ponto a ponto definida por polinomiais

Além de trajetórias compostas por segmentos de retas, muitos autores estudaram trajetórias compostas por polinomiais. Autores como (ANGELES, 2003) e (BORYGA; GRABOS, 2008) apresentam interpolação polinomial para o tipo de operação ponto a ponto, onde são consideradas polinomiais de quinto, sétimo e nono grau.

Sabe-se que uma polinomial possui infinitas derivadas. Porém, em trajetórias robóticas o robô deve partir suavemente do ponto inicial e chegar suavemente no ponto final. Para que a restrição de suavidade de movimento seja garantida, o robô parte com velocidade nula e chega ao ponto final também com velocidade nula. Conforme as leis da dinâmica de corpos rígidos, para que não ocorram forças/torques, que podem provocar vibrações, é necessário que as acelerações nestes pontos também sejam nulas. Além disso, considerando que as taxas de

aceleração e desaceleração representam impactos, estas taxas (jerk) também devem ser nulas no início e final do movimento.

Trajetórias definidas por polinômios do quinto grau garantem continuidade de posição, velocidade e aceleração no início e fim do movimento, porém o Jerk é descontínuo. Para garantir a continuidade C^0 do Jerk, pode-se utilizar uma polinomial do sétimo grau. Porém, a sua derivada é descontínua. Para garantir continuidade C^1 do jerk, são introduzidas mais duas restrições de continuidade, sendo o polinômio final do nono grau.

A interpolação polinomial com vários pontos não é adequada para o planejamento de trajetórias robóticas, pois a ordem da polinomial depende da quantidade de pontos a serem interpolados, ou seja, na interpolação com $n+1$ pontos obtêm-se um polinômio de grau n . Em (KAHANER; MOLER; NASH, 1989), (QUARTERONI; SACCO; SALERI, 2000) e no Capítulo III é apresentado que, conforme o número de pontos a serem interpolados aumenta, pior é o comportamento do polinômio.

Uma alternativa para trabalhar com trajetórias multipontos é definir uma polinomial a cada dois ou mais pontos respeitando a condição de estabilidade em relação à quantidade de pontos. O mais comum é a utilização de dois e de quatro pontos. Com dois pontos define-se uma interpolação cúbica de Hermite e com quatro pontos uma aproximação cúbica de Bézier por partes.

Em (CHAUDHRY; GULREZ; ZIA, 2010) é apresentada uma metodologia com curvas de Bézier para o planejamento de trajetórias de robô móveis autônomos com rodas. Nesse trabalho o robô deve chegar ao destino sem ter um caminho prévio definido. Para isso, o robô utiliza um sistema de varredura a laser que identifica obstáculos. A partir dos obstáculos identificados, o algoritmo de navegação cria os próximos quatro melhores pontos para que o próximo segmento da trajetória seja seguido.

(OKAWA; NONAKA, 2010) utilizam interpolação cúbica por partes para o planejamento de trajetória de robôs móveis. Os coeficientes das polinomiais são obtidos por um algoritmo de otimização, o qual busca o valor mínimo de coeficientes que multiplica o termo cúbico e quadrático do polinômio. Segundo os autores esse processo garante uma curva suave.

(WAGNER et al., 2010) utilizam interpolação cúbica de Hermite para o planejamento de trajetórias para robôs móveis, isso porque a interpolação cúbica de Hermite permite definir a derivada no ponto de partida do robô e no ponto de chegada. Assim, para trajetórias definidas por muitos pontos a derivada do início de um segmento é igual a derivada do final do segmento anterior.

Outra metodologia de interpolação por partes é a spline e que é muito utilizada em planejamento de trajetórias robóticas, sendo descrita a seguir.

2.8.3. Trajetórias definidas por spline

A spline é uma curva de interpolação composta por polinomiais por partes. A cada dois pontos a interpolação é realizada por um polinômio, cujo grau é definido a partir da ordem de continuidade da curva. No Capítulo III é apresentado em detalhes o procedimento para o cálculo de uma spline.

Para uma trajetória definida por mais de dois pontos é comum a utilização da spline cúbica obtida por polinomiais por partes, visto que uma polinomial única gera problemas de estabilidade de movimento.

Em (BAZAZ; TONDU, 1999), (CAO; DODDS; IRWIN, 1997), (PLESSIS; SNYMAN, 2003) é realizado o planejamento de trajetórias no domínio das articulações para manipuladores industriais utilizando spline. Para determinar os coeficientes dos segmentos da spline são utilizados métodos de otimização considerando o tempo mínimo para que a trajetória seja realizada e as restrições cinemáticas dos atuadores sejam respeitadas.

Em (VISIOLI, 2000) é apresentada uma estratégia para planejamento de trajetórias por spline cúbica. A estratégia aplicada para obter este tipo de curva é fazer uma alteração no primeiro e último segmento da interpolação, pois nas extremidades do trajeto não é possível garantir a continuidade da aceleração. Assim, foi proposto modificar os polinômios da extremidade para o quinto grau a fim de satisfazer as condições de continuidade.

As splines também são muito aplicadas em planejamento de trajetórias em robótica móvel. (EREN; FUNG; EVANS, 1999) utilizam spline cúbica para gerar trajetórias de robôs. Apesar da spline cúbica possuir a suavidade necessária para uma trajetória robótica, este tipo de curva não é adequada para ser utilizada em trajetórias com a presença de obstáculos móveis, visto que com a alteração de um ponto de controle, toda curva se altera.

(OLABI et al., 2010) aplicam interpolação por spline quíntica no planejamento de trajetórias de robôs industriais, pois este tipo de curva possui continuidade de posição, velocidade, aceleração e jerk, o que propicia uma trajetória mais suave.

Curvas spline são relativamente fáceis de serem calculadas porém, para trajetórias na presença de obstáculos elas não possuem um desempenho satisfatório. Com a mudança de um ponto de controle, toda a curva é alterada. Para que a alteração de um ponto de controle não

mude o perfil de toda a curva, alguns autores empregam B-spline e NURBS no planejamento de trajetórias robóticas.

2.8.4. Trajetórias definidas por B-spline e NURBS

Curvas B-spline e NURBS são versões da spline que implementam o controle local da curva, de forma que a alteração de um ponto de controle modificará a curva somente na região dos pontos vizinhos mais próximos em função da ordem de continuidade. Estas curvas são ajustadores de aproximação, isto é, não passam pelos pontos de controle. No entanto, os pontos de controle podem ser calculados para que a interpolação seja realizada nos pontos desejados. No Capítulo III é apresentado em detalhes o procedimento para o cálculo de B-spline e NURBS.

Apesar da spline cúbica ser uma boa alternativa para o planejamento de trajetórias multipontos, pesquisadores como (LARTIGUE; THIEBAUT; MAEKAWA, 2001), (LEI; WANG, 2009), (CAN; UNIVAR, 2009), (LAI, 2010) e (EMAMI; AREZOO, 2010) empregam NURBS para a realização da trajetória da ferramenta de máquinas CNC (Comando Numérico Computadorizado). Segundo os autores, a NURBS foi escolhida porque é uma curva que oferece grande flexibilidade geométrica.

Em (SARAMAGO; STEFFEN, 1999) e (SARAMAGO et al., 2005) é utilizado B-spline cúbica para o planejamento de trajetórias robóticas. Assim como a spline cúbica, este tipo de curva possui continuidade de posição, velocidade e aceleração. Porém, com a B-spline é possível realizar modificações locais na curva.

Em (OZAKI et al., 2003) e (CAIGNY et al., 2007) é apresentado o planejamento de trajetórias robóticas utilizando B-spline do quinto grau, isto é, continuidade até do jerk. Nos seus experimentos este tipo de curva garantiu uma melhor acuracidade no seguimento da referência da trajetória desejada.

No planejamento de trajetórias no domínio da tarefa, a curva geométrica do trajeto é definida em função dos pontos que são ‘ensinados’ ao robô. Dessa forma, quanto menor o erro desejado para que o robô siga determinado perfil, maior a quantidade de pontos devem ser ‘ensinados’. O planejamento de trajetórias robóticas no domínio da tarefa utilizando B-spline reduz significativamente a quantidade de pontos que são necessários no ‘ensino’ do robô em relação ao uso de spline. Com isso há um aumento na produtividade no processo de

‘ensino’, pois com a redução dos pontos, o tempo necessário na programação também é reduzido (HORSCH; JÜTTLER, 1998).

Autores como (FLORES; MILAN, 2006) utilizam NURBS para o planejamento de trajetórias para UAV (*Unmanned Aerial Vehicle*). Devido à quantidade de parâmetros de controle da NURBS, como pontos de controle e seus pesos associados, este tipo de curva se apresentou como uma boa opção para trajetórias *on-line* na presença de obstáculos.

Em (ALEOTTI; CASELLI; MACCHEROZZI, 2005) é realizado um estudo com a utilização de NURBS em trajetórias de robôs industriais na presença de obstáculos. Os resultados mostraram que este tipo de curva possui melhor desempenho devido à sua grande flexibilidade.

2.9. Conclusão

Sem dúvida, os came tiveram um papel primordial no desenvolvimento dos sistemas eletro-mecânicos para que as máquinas multi-eixos chegassem ao grau de complexidade que atualmente possuem.

A dificuldade em reprojeter um came a cada nova tarefa que o sistema tinha que realizar e associado ao desenvolvimento tecnológico, provocaram o surgimento de sistemas mais ‘inteligentes’ como os came eletrônicos. Assim, com a utilização da eletrônica e da programação foi possível a definição de trajetórias mais complexas.

O dispositivo de Devol “*Programmed Article Transfer*” foi baseado na idéia de came eletrônico. Assim, a partir deste dispositivo, a robótica tem se desenvolvido continuamente possibilitando os robôs realizar diversas tarefas.

Diversas formas de planejamento de trajetórias robóticas têm sido pesquisadas de forma a permitir que o robô execute sua tarefa descrevendo movimentos suaves e em tempo reduzido, respeitando suas restrições cinemáticas e dinâmicas, além de evitar colisões no ambiente de trabalho.

Para que o robô realize uma tarefa de forma suave e rápida, sua trajetória deve ser planejada cuidadosamente com alguma metodologia de ajuste de curva. Assim, um estudo detalhado de ajuste de curva se faz necessário quando o objetivo é buscar uma trajetória mais suave e flexível.

CAPÍTULO III

AJUSTE DE CURVA

3.1. Introdução

Ajuste de curva é o processo matemático para a obtenção de uma função que define uma curva a partir de dois ou mais pontos. Segundo (PIEGL; TILLER, 1997) existem duas formas de ajuste de curva: aproximação e interpolação. A curva obtida por interpolação passa por todos os pontos dados, diferentemente da aproximação a qual não necessariamente passa pelos pontos, como mostrado na Fig.3.1.

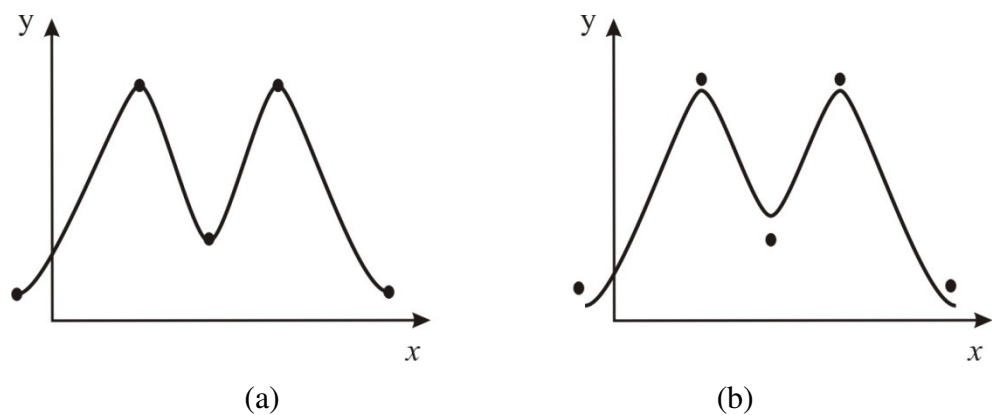


Figura 3.1: a) Interpolação; b) Aproximação

Sendo que o planejamento de trajetórias robóticas é feito a partir de um conjunto de pontos em que o elemento terminal do robô deve passar para executar a tarefa, um método adequado de ajuste de curva precisa ser aplicado. Deve-se salientar que a trajetória robótica, qualquer que seja a metodologia utilizada para seu planejamento, deve ser a mais suave possível.

Assim, neste capítulo são apresentadas diferentes metodologias de ajuste de curva que podem ser utilizadas em planejamento de trajetórias, bem como suas vantagens, desvantagens e características específicas que as definem.

O estudo não pretende ser exaustivo, mas com claro direcionamento à aplicação de B-spline e NURBS (*Non – Uniform Rational B-splines*) ao planejamento de trajetórias robóticas.

3.2. Interpolação polinomial

O tipo de interpolação mais conhecida é a interpolação polinomial, a qual consiste em obter uma função definida por um polinômio $C(x)$ que passa por um conjunto de $n+1$ pontos, $P_0, P_1, P_2, \dots, P_n$.

Com $n+1$ pontos distintos conhecidos, é possível obter um único polinômio $C_n(x)$ de grau n que passa por esses pontos. Assim, se são conhecidos os pontos P_i definidos por (x_i, y_i) , $(i=0, \dots, n)$, e com $x_i < x_{i+1}$ e $x_i \neq x_j$ ($i, j = 0, 1, \dots, n$), pode-se obter o polinômio de grau n para $x \in [x_0, \dots, x_n]$ dado por:

$$C_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x^1 + a_0 \quad (3.1)$$

Existem muitos métodos para a obtenção do polinômio interpolante, em (KAHANER; MOLER; NASH, 1989), (RUGGIERO; LOPES, 1996) e (QUARTERONI; SACCO; SALERI, 2000) pode-se encontrar as metodologias em detalhes.

Os métodos são apresentados de forma resumida a seguir.

3.2.1. Coeficientes Indeterminados

Este método obtém os coeficientes de um polinômio por meio da resolução de um sistema linear. O procedimento é realizado pela substituição do conjunto de pontos dados na Eq. (3.2). Assim, para $n+1$ pontos é possível obter um sistema linear de $n+1$ equações linearmente independentes.

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \dots + a_2 x_0^2 + a_1 x_0^1 + a_0 = y_0 \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_2 x_1^2 + a_1 x_1^1 + a_0 = y_1 \\ \dots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \dots + a_2 x_n^2 + a_1 x_n^1 + a_0 = y_n \end{cases} \quad (3.2)$$

3.2.2. Interpolação de Lagrange

O polinômio interpolador de Lagrange é uma soma ponderada de polinômios, e é considerado o método mais prático e compacto de ser aplicado.

O procedimento é descrito pela equação:

$$C_n(x) = \sum_{i=0}^n y_i \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} \quad (3.3)$$

3.2.3. Interpolação de Newton

É um método baseado no conceito de diferenças divididas. A diferença dividida de 1ª ordem consiste em uma aproximação da derivada primeira e pode ser denotada como:

$$C[x_{i+1}, x_i] = \frac{C(x_{i+1}) - C(x_i)}{x_{i+1} - x_i} \quad (3.4)$$

Com a diferença dividida de 1ª ordem pode-se obter a diferença dividida de 2ª ordem do seguinte modo:

$$C[x_i, x_{i+1}, x_{i+2}] = \frac{C[x_{i+1}, x_{i+2}] - C[x_i, x_{i+1}]}{x_{i+2} - x_i} \quad (3.5)$$

Da mesma forma a diferença dividida de 3ª ordem é dada por:

$$C[x_i, x_{i+1}, x_{i+2}, x_{i+3}] = \frac{C[x_{i+1}, x_{i+2}, x_{i+3}] - C[x_i, x_{i+1}, x_{i+2}]}{x_{i+3} - x_i} \quad (3.6)$$

Genericamente, a diferença dividida de ordem n é dada por:

$$\nabla^n y_i = C[x_i, x_{i+1}, \dots, x_{i+n}] = \frac{C[x_{i+1}, x_{i+2}, \dots, x_{i+n}] - C[x_i, x_{i+1}, \dots, x_{i+n-1}]}{x_{i+n} - x_i} \quad (3.7)$$

A diferença dividida de ordem zero é definida como $C[x_i] = [x_i] = \nabla^0 y_i = y_i$

Dessa forma o polinômio interpolador de Newton é dado por:

$$C_n(x) = y_0 - \nabla^1 y_0(x - x_0) + \nabla^2 y_0(x - x_1)(x - x_0) + \nabla^3 y_0(x - x_2)(x - x_1)(x - x_0) + \dots + \nabla^n y_0(x - x_{n-1}) \dots (x - x_1)(x - x_0) \quad (3.8)$$

3.2.4. Interpolação de Hermite

A interpolação de Hermite é uma extensão da interpolação polinomial que relaciona as informações de um conjunto de pontos e as derivadas até a k -ésima ordem nestes pontos. O procedimento para obter o polinômio interpolador de Hermite é definido como:

$$C(x_i) = y_i \quad i=0,1,\dots,n \quad \rightarrow n+1 \text{ restrições} \quad (3.9)$$

$$C'(x_i) = D(x_i) \quad i=0,1,\dots,n \quad \rightarrow n+1 \text{ restrições} \quad (3.10)$$

$$C''(x_i) = D^2(x_i) \quad i=0,1,\dots,n \quad \rightarrow n+1 \text{ restrições} \quad (3.11)$$

$$C^{(k)}(x_i) = D^{(k)}(x_i) \quad i=0,1,\dots,n \quad \rightarrow n+1 \text{ restrições} \quad (3.12)$$

Onde, $D^{(k)}(x_i)$ é definido como a k -ésima derivada no ponto de interpolação.

Então, tem-se um total de $(k+1).(n+1)$ restrições, necessitando um polinômio geral de grau $[(k+1).(n+1)-1]$.

A equação geral de um polinômio com o total de $(k+1).(n+1)$ restrições pode ser escrita como:

$$C(x) = \sum_{i=0}^{(k+1).(n+1)-1} a_i x^i \quad (3.13)$$

O procedimento de obtenção da interpolação de Hermite consiste basicamente em três etapas:

- a) A montagem do polinômio interpolador;
- b) A implementação das restrições;
- c) Resolver o sistema para os coeficientes indeterminados a_i , com $i=0,1,\dots,(k+1)(n+1)-1$.

Pode-se observar que a interpolação de Lagrange é um caso especial da interpolação de Hermite, quando $k=0$ (i.e., não são consideradas as derivadas).

É também possível obter um caso especial de interpolação de Hermite onde não são considerados todos os valores da função e/ou das derivadas em todos os pontos suportes.

3.2.5. Problemas numéricos com polinômios de alta ordem

No planejamento de trajetórias robóticas, a interpolação polinomial pode apresentar dois aspectos indesejados: instabilidade e erro de truncamento. A instabilidade é quando a curva possui oscilações de amplitudes elevadas nas regiões das extremidades. Isso ocorre devido à disposição dos pontos de controle associado com o aumento da ordem do polinômio. As Figuras 3.2, 3.3, 3.4 e 3.5 mostram a evolução da instabilidade com o aumento da quantidade de pontos.

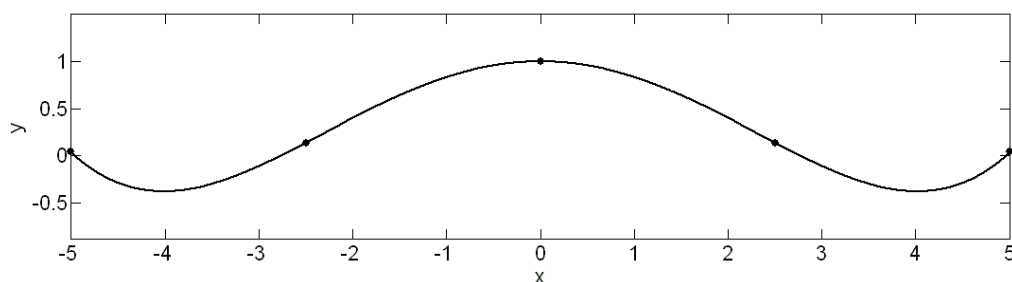


Figura 3.2: Interpolação polinomial realizada para cinco pontos: polinômio do quarto grau.

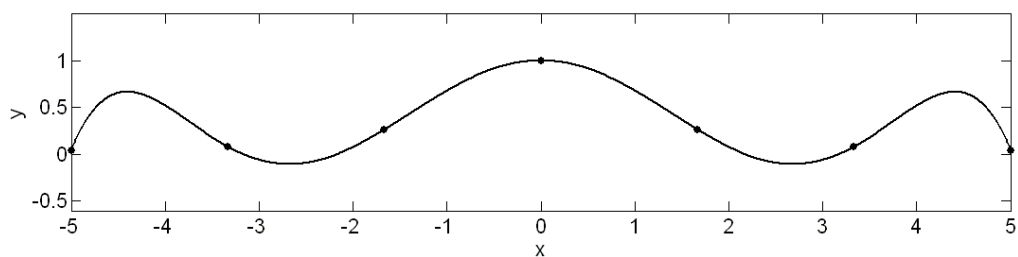


Figura 3.3: Interpolação polinomial realizada para sete pontos: polinômio do sexto grau.

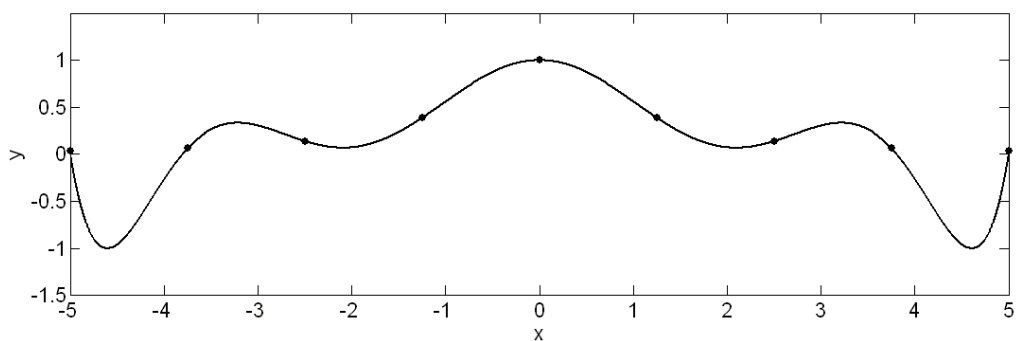


Figura 3.4: Interpolação polinomial realizada para nove pontos: polinômio do oitavo grau.

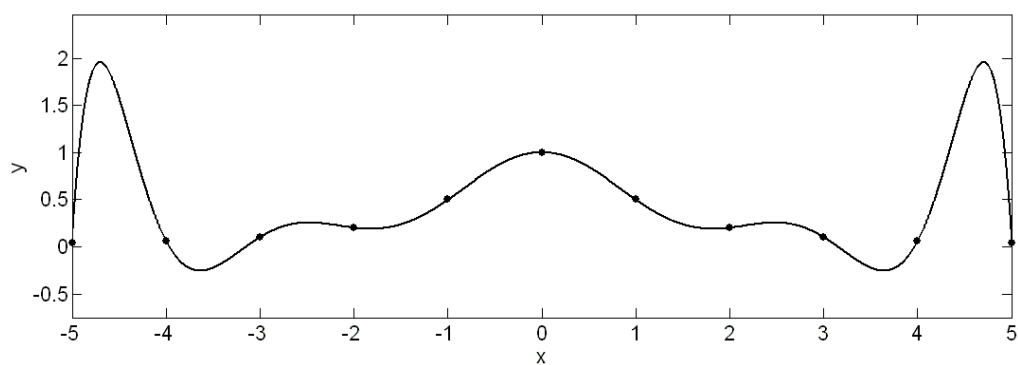


Figura 3.5: Interpolação polinomial realizada para onze pontos: polinômio do décimo grau.

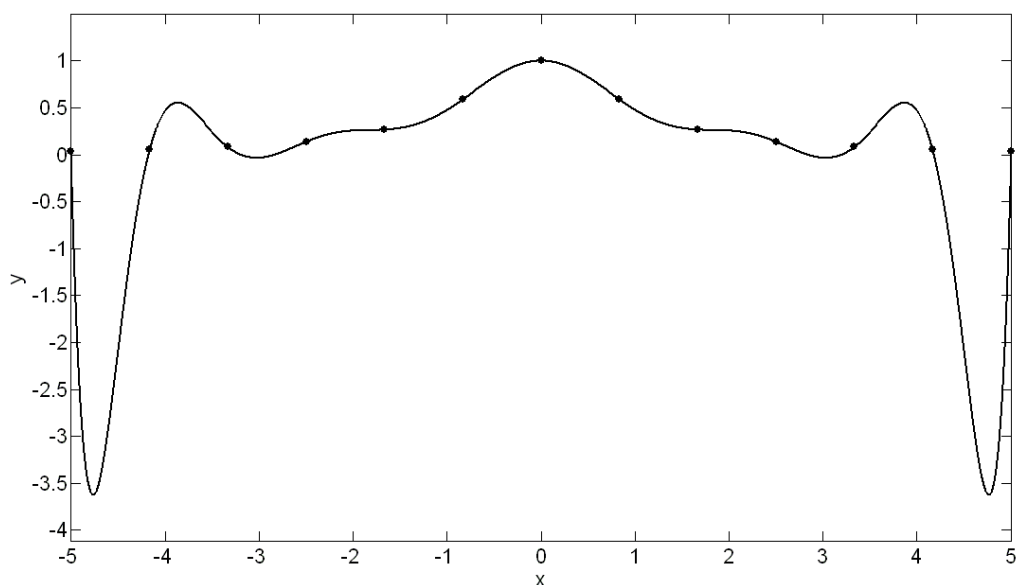


Figura 3. 6: Interpolação polinomial realizada para treze pontos: polinômio do décimo segundo grau.

Em sistemas computacionais trabalha-se com um número determinado de casas decimais. Assim, se um número possui infinitas casas decimais, ele será representado por um número limitado de casas decimais devido à disponibilidade de memória para representá-lo. Com o aumento do grau de um polinômio, a tendência é que os valores de seus coeficientes diminuam.

Supondo que um sistema esteja configurado para trabalhar com nove casas decimais, caso os coeficientes do polinômio calculado esteja na ordem da nona casa decimal, a interpolação terá um erro e divergirá como mostrado na Fig. 3.7.

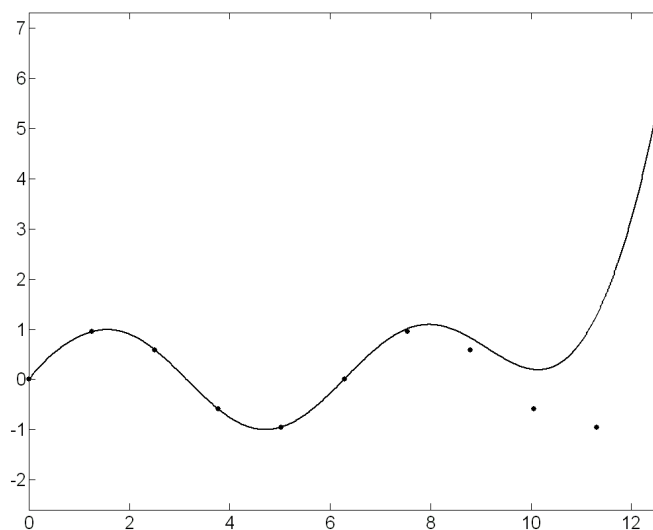


Figura 3.7: Erro devido ao truncamento em uma interpolação polinomial de 11 pontos.

Uma alternativa para contornar os problemas de instabilidades numéricas, é trabalhar com polinômios de graus menores. Para uma trajetória multipontos pode-se calcular um polinômio para cada segmento definido por dois pontos garantindo a continuidade desejada. Esse processo é denominado interpolação por partes.

3.3. Polinomial por Partes

Uma alternativa para a interpolação polinomial para um conjunto de pontos consiste em obter polinômios interpolantes para subconjuntos de dados. Este procedimento é conhecido por interpolação polinomial por partes.

As polinomiais são unidas nos extremos de cada intervalo definido por pontos respeitando certo grau de continuidade da curva nesses pontos.

Existem diversas formas de realizar a interpolação por polinomial por partes. Neste capítulo são apresentadas as formas de continuidade bem como as interpolações por partes.

3.3.1. Continuidade

Antes de definir continuidade de uma curva é necessário entender a formulação de curvas paramétricas. Em geral, as equações matemáticas são utilizadas na sua forma implícita. Por exemplo, uma curva descrita no plano xy tem sua forma implícita do tipo $f(x,y) = 0$. Isto significa que a equação descreve uma relação implícita entre as coordenadas x e y dos pontos pertencentes à curva.

Na forma paramétrica, cada coordenada do ponto da curva é representada separadamente como uma função explícita de um parâmetro independente. Então, uma função $C(u)$ pode ser descrita como,

$$C(u) = [x(u), y(u), z(u)] \quad (a \leq u \leq b)$$

$$\begin{aligned}
x(u) &= a_{xn}u^n + a_{x(n-1)}u^{n-1} + \dots + a_{x2}u^2 + a_{x1}u^1 + a_{x0} \\
y(u) &= a_{yn}u^n + a_{y(n-1)}u^{n-1} + \dots + a_{y2}u^2 + a_{y1}u^1 + a_{y0} \\
z(u) &= a_{zn}u^n + a_{z(n-1)}u^{n-1} + \dots + a_{z2}u^2 + a_{z1}u^1 + a_{z0}
\end{aligned} \tag{3.14}$$

Embora o intervalo $[a, b]$ seja arbitrário, freqüentemente o parâmetro u é normalizado de zero a um, isto é, $0 \leq u \leq 1$.

Existem dois tipos de continuidade associado a uma curva: continuidade geométrica e continuidade paramétrica.

A continuidade geométrica está relacionada com a suavidade geométrica da curva. Para uma curva formada somente pela união de segmentos, sua continuidade no ponto de união é G^0 , isto é, no ponto de união a curva não possui continuidade na derivada. Na união de segmentos em que a direção do vetor tangente no ponto final de um segmento possui a mesma direção do vetor tangente do ponto inicial do próximo segmento, a curva resultante possui continuidade G^1 no ponto de união. A continuidade geométrica de uma curva pode ser definida até a $(p-1)$ -ésima ordem G^k ($k = 1, \dots, p-1$), sendo k a k -ésima derivada das curvas. A Figura 3.8 mostra as curvas com continuidade G^0 e G^1 .

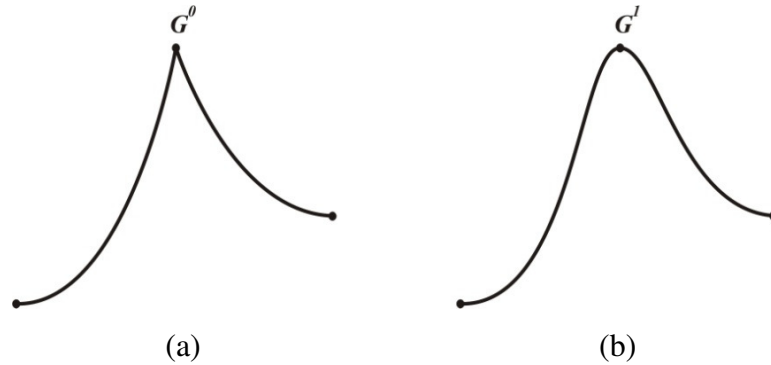


Figura 3. 8: (a) Curva formada por dois segmentos com continuidade G^0 no ponto de união. (b) Curva formada por dois segmentos com continuidade G^1 no ponto de união.

Curva na forma explícita é possível definir somente continuidade geométrica, porém curvas paramétricas possuem além da continuidade geométrica a continuidade paramétrica. Para que uma curva possua continuidade paramétrica, a direção do vetor tangente no ponto final de um segmento deve possuir a mesma direção do vetor tangente do ponto inicial do próximo segmento, além dos módulos serem iguais, neste caso a continuidade da curva é C^1 .

A continuidade paramétrica de uma curva pode ser definida até a $(p-1)$ -ésima ordem C^k ($k = 1, \dots, p-1$). Toda curva de continuidade C^k possui continuidade G^k , mas nem toda curva de continuidade G^k possui continuidade C^k .

A Figura 3.9 permite visualizar o comportamento das curvas no caso de continuidade geométrica e paramétrica de ordem 1.

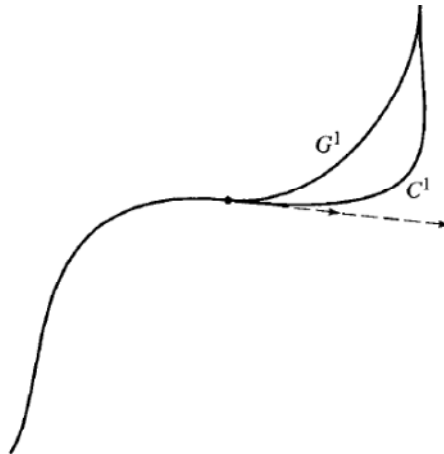


Figura 3.9: Comparação entre a continuidade G^1 e a C^1 (ROGERS, 2001).

Em uma trajetória robótica definida do espaço da tarefa a continuidade G^k representa a suavidade da curva que define a trajetória. A continuidade C^k representa a suavidade que a trajetória é realizada pelo robô, em termos da continuidade da velocidade C^1 , da aceleração C^2 e do Jerk C^3 .

3.3.2. Interpolação linear por partes

A interpolação linear por partes consiste em conectar sucessivamente o conjunto de pontos por segmentos de retas. Neste caso ocorrem vértices em cada ponto, isto é, continuidade C^0 , como pode ser observado na Fig. 3.10.

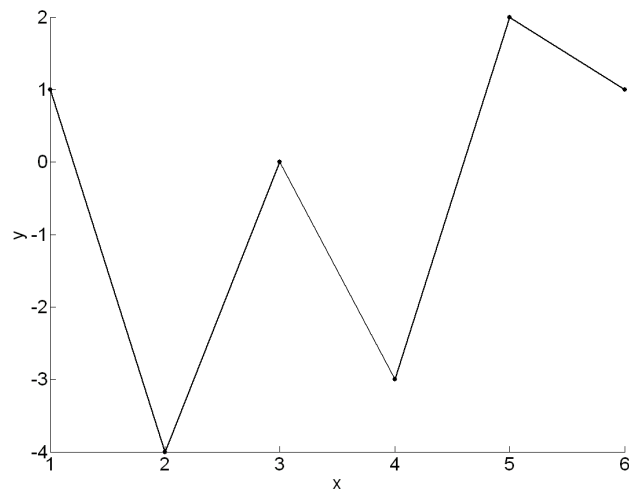


Figura 3.10: Interpolação linear por partes.

3.3.3. Interpolação quadrática por partes

A interpolação quadrática por partes usa pares de intervalos adjacentes no qual uma função interpolante quadrática é construída a partir de três pontos. Embora a curva seja contínua, não garante que ela seja diferenciável nos extremos de cada intervalo, porque consiste na união de duas funções quadráticas, Fig. 3.11.

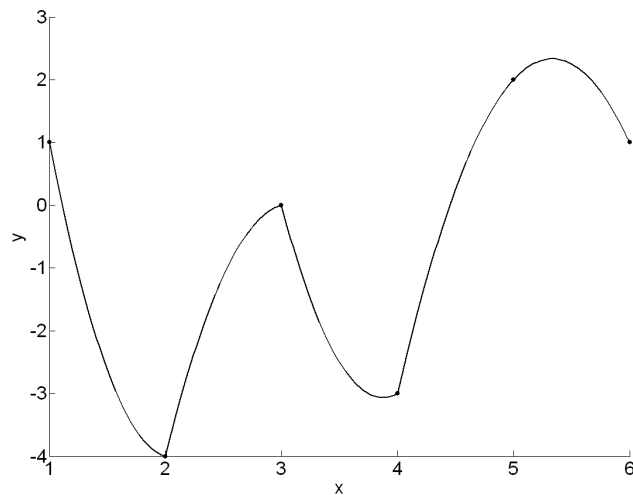


Figura 3.11: Interpolação quadrática por partes.

3.3.4. Interpolação cúbica por partes

Neste caso, são utilizados três subintervalos adjacentes no qual é construído um polinômio cúbico interpolante através de quatro pontos. Novamente, apesar da curva ser

continua não garante que ela seja diferenciável nos extremos dos intervalos como mostrado na Fig. 3.12.

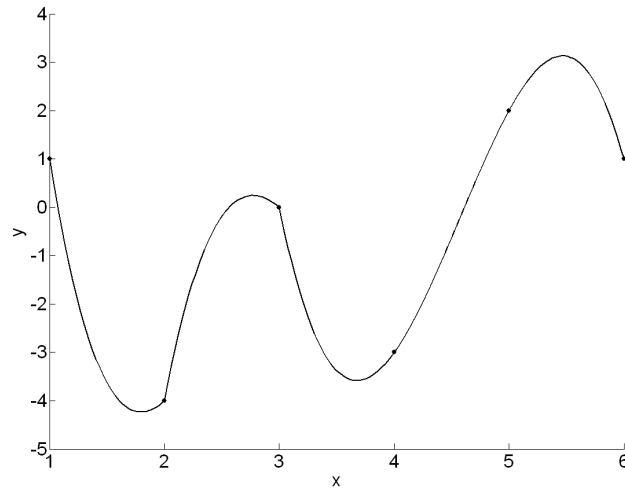


Figura 3. 12: Interpolação cúbica por partes.

3.3.5. Interpolação de Hermite por partes

Como mostrado anteriormente, a interpolação por partes é a união de polinômios sem a preocupação com a ordem de continuidade nos extremos de cada subintervalo. A interpolação de Hermite por partes considera a continuidade desejada em cada ponto. Assim a cada subintervalo existem $(2k+2)$ restrições para que a continuidade da curva seja C^k .

Em trajetórias robóticas contínuas, uma curva com continuidade C^0 não é adequada. Dessa forma, a curva deve possuir ao menos uma continuidade C^1 , ou seja, para a continuidade C^1 existem quatro restrições para seu cálculo, resultando em uma curva cúbica. A idéia é construir um polinômio interpolante cúbico em cada subintervalo que seja contínuo e diferenciável nos extremos de cada intervalo.

O procedimento básico consiste em obter polinômio cúbico interpolante nos subintervalos sucessivos $[x_0, x_1)$, $[x_1, x_2)$, ..., $[x_{n-1}, x_n]$ através da interpolação polinomial de Hermite da esquerda para a direita em todos os subintervalos, Fig. 3.13.

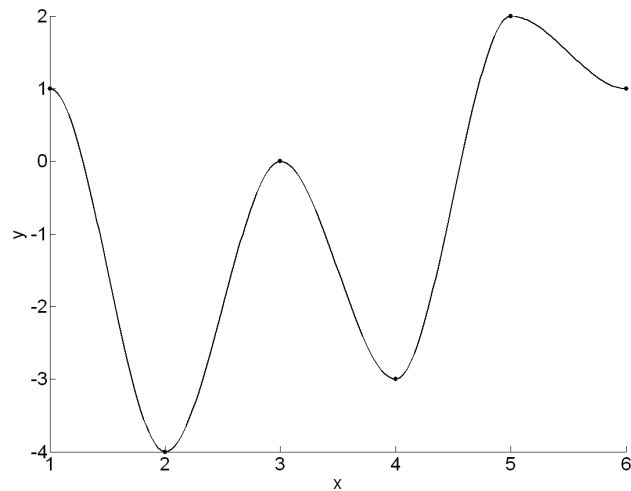


Figura 3. 13: Interpolação cúbica de Hermite por partes.

3.4. Spline

O princípio básico da spline é a interpolação de Hermite por partes. Enquanto na Interpolação de Hermite por partes a derivada em cada ponto é imposta, na spline é estabelecida somente a condição de continuidade. Isso significa que a cada dois segmentos consecutivos da curva as derivadas de ordem k do final de um segmento e do início do segmento seguinte são iguais.

Considerando um conjunto de $n+1$ pontos, é possível definir uma spline que os interpola. Portanto, para obter a spline deve-se encontrar as polinomiais, $S_i(x)$ ($i = 0, \dots, n-1$) do p -ésimo grau conforme a Eq. (3.15), para os n segmentos como mostrado na Fig. 3.14. Dado um polinômio do p -ésimo grau, sabe-se que ele possui $(p+1)$ coeficientes, dessa forma é preciso resolver um sistema de equações para encontrar $n(p+1)$ coeficientes da spline.

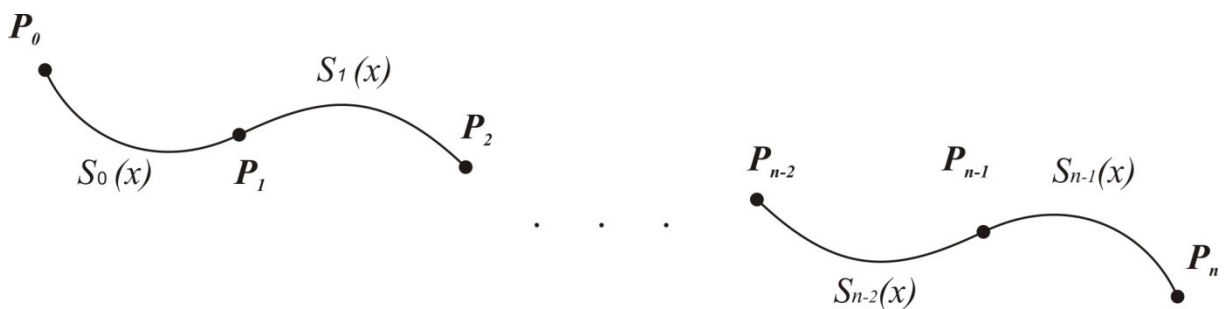


Figura 3.14: Interpolação por spline.

$$S_i(x) = a_{p,i}x^p + a_{p-1,i}x^{p-1} + \dots + a_{1,i}x + a_{0,i} \quad (3.15)$$

Assim, as $n(p+1)$ equações devem ser formuladas para que os coeficientes dos polinômios sejam encontrados. Essas equações são obtidas pelas condições de contorno dadas por:

- Continuidade C^0 na união dos segmentos $\rightarrow 2(n-1)$ equações

$$\begin{aligned} S_0(x_1) &= y_1 \\ S_1(x_1) &= y_1 \\ &\vdots \\ &\vdots \\ &\vdots \\ S_{n-2}(x_{n-1}) &= y_{n-1} \\ S_{n-1}(x_{n-1}) &= y_{n-1} \end{aligned} \quad (3.16)$$

- Continuidade C^1 à C^{p-1} na união dos segmentos $\rightarrow (p-1)(n-1)$ equações

Na união dos segmentos da spline define-se a continuidade desejada. Para isso as derivadas de um segmento devem ser iguais as dos seus segmentos vizinhos até a p -ésima ordem.

Condição de continuidade na derivada primeira

$$\begin{aligned} S'_0(x_1) &= S'_1(x_1) \\ S'_1(x_2) &= S'_2(x_2) \\ &\vdots \\ &\vdots \\ &\vdots \\ S'_{n-2}(x_{n-1}) &= S'_{n-1}(x_{n-1}) \end{aligned} \quad (3.17)$$

Condição de continuidade na derivada segunda

$$S''_0(x_1) = S''_1(x_1)$$

$$S''_1(x_2) = S''_2(x_2)$$

·
·
·

$$S''_{n-2}(x_{n-1}) = S''_{n-1}(x_{n-1}) \quad (3.18)$$

O processo é realizado sucessivas vezes até a condição da $(p-1)$ -ésima derivada,

$$S^{p-1}_0(x_1) = S^{p-1}_1(x_1)$$

$$S^{p-1}_1(x_2) = S^{p-1}_2(x_2)$$

·
·
·

$$S^{p-1}_{n-2}(x_{n-1}) = S^{p-1}_{n-1}(x_{n-1}) \quad (3.19)$$

- **Valor da curva nos pontos extremos → 2 equações**

$$S_0(x_0) = y_0$$

$$S_{n-1}(x_n) = y_n \quad (3.14)$$

- **Derivadas nos pontos extremos → $(p-1)$ equações**

Neste caso são impostas as condições de contorno necessárias para completar as equações do sistema. Por exemplo, se $p = 3$, pode-se calcular a spline com a derivada segunda nula nos extremos, neste caso a curva é conhecida como spline cúbica natural.

$$S''_0(x_0) = 0$$

$$S''_{n-1}(x_n) = 0 \quad (3.15)$$

Com essas condições de contorno é possível construir um sistema com $n(p+1)$ equações, cuja solução são os coeficientes dos polinômios da spline.

Utilizando os mesmos pontos da curva na Fig. 3.6, foi obtida uma spline cúbica natural mostrada na Fig. 3.15. Pode-se observar que a curva da Figura 3.15 é mais suave que a curva da Fig. 3.6, além de não possuir instabilidades. No Anexo I é apresentado o algoritmo em Matlab[®] para a interpolação por spline cúbica natural.

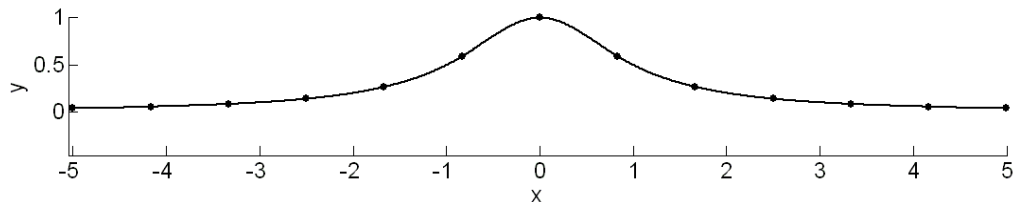


Figura 3.15: Interpolação com spline cúbica.

Apesar de ser possível definir a continuidade desejada de uma spline, a alteração de um ponto deste tipo de curva pode modificar o perfil de toda a curva, como mostrado na Fig. 3.16, para a alteração do ponto P .

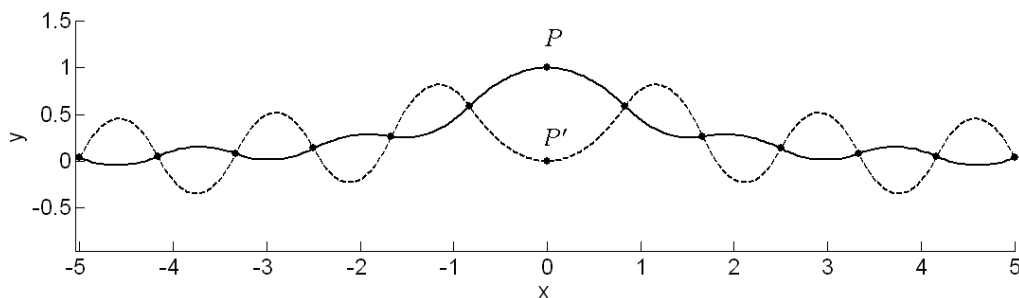


Figura 3.16: Interpolação com spline quadrática. Interpolação com os pontos originais (linha contínua), interpolação com a mudança de do sétimo ponto P (linha tracejada).

3.5. B-spline

Autores como (PIEGL; TILLER, 1997), (DE BOOR, 2000) e (ROGERS, 2001) definem a B-spline como uma versão da spline que implementa o controle local da curva, de forma que a alteração de um ponto de controle modificará a curva somente na região dos pontos vizinhos

mais próximos em função da ordem de continuidade, como mostrado na Fig. 3.17. Ao contrário da spline que a alteração de um ponto de controle pode modificar o formato de toda a curva, como mostrado na Fig. 3.18.

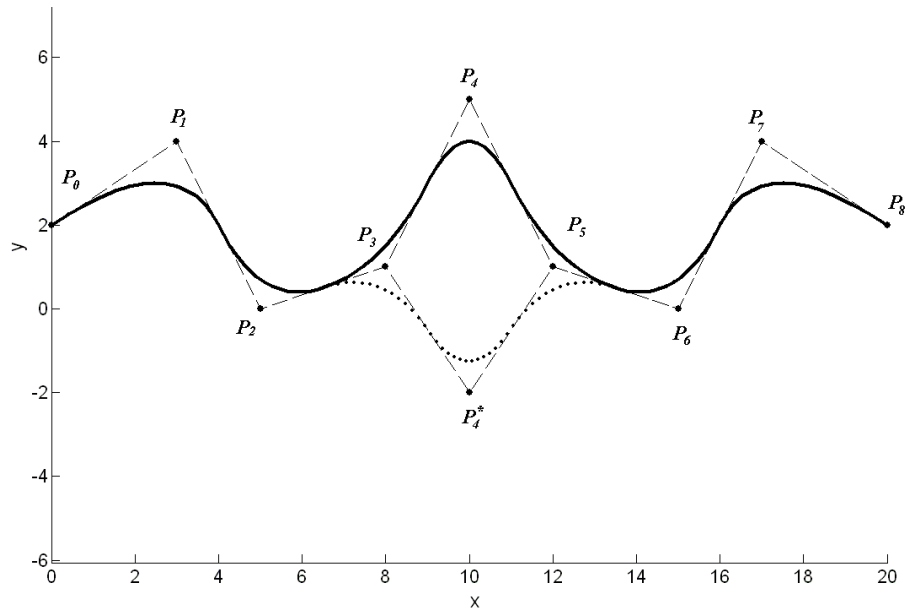


Figura 3.17: B-spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto P_4 .

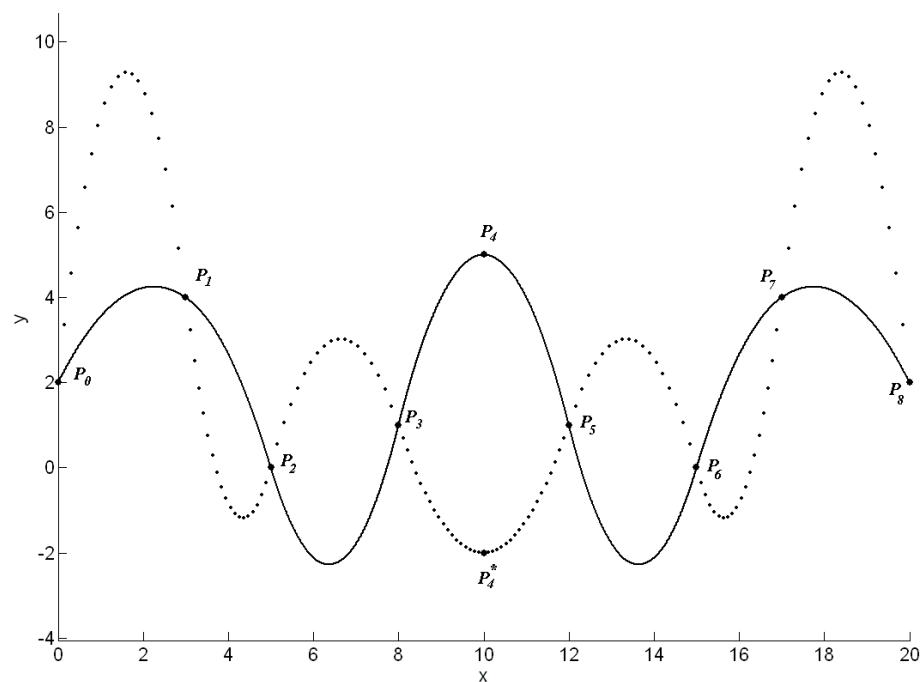


Figura 3.18: Spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto P_4 .

Curvas B-spline são calculadas a partir de funções de base que são apresentadas a seguir.

3.5.1. Funções de base da B-spline

Existem vários métodos para definir as funções de base da B-spline, porém o utilizado neste trabalho é o método de recorrência de Cox-deBoor, visto que ele é muito útil para implementação computacional. Como mostrado em (PIEGL; TILLER, 1997), seja $U=\{u_0, \dots, u_m\}$ uma seqüência de números reais não decrescente, i.e., $u_i \leq u_{i+1}$, $i = 0, 1, \dots, (m-1)$. Os u_i são denominados nós (knots), e U o vetor dos nós (*knot vector*). A i -ésima função de base B-spline de grau p (ordem $p+1$), denotado por $N_{i,p}(u)$, é definida como:

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{outros casos} \end{cases} \quad (3.16)$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \quad (3.17)$$

Pode-se observar que:

- a) $N_{i,0}(u)$ é uma função pulso, é sempre igual a zero exceto no intervalo aberto $u \in [u_i, u_{i+1})$;
- b) Para $p > 0$, $N_{i,p}(u)$ é uma combinação linear de duas funções de base de grau $p-1$;
- c) O cálculo de um conjunto de funções de base requer a especificação de um vetor de nós U , e o grau p ;
- d) A Equação (3.17) pode levar a quocientes do tipo $0/0$. Esses quocientes são definidos como zero;
- e) $N_{i,p}(u)$ são polinomiais por partes, definidos em todo domínio real. Em geral somente o intervalo $[u_0, u_m]$ é de interesse;
- f) O intervalo aberto, $[u_i, u_{i+1})$, é denominado de i -ésimo intervalo do nó. Ele pode ter comprimento nulo, desde que os nós não necessitem ser distintos;
- g) O cálculo das funções de grau p gera uma tabela triangular truncada, como a seguir:

$$\begin{array}{ccccccc}
& & & & N_{0,0} & & \\
& & & & & N_{0,1} & \\
& & & & & & N_{0,2} \\
& & N_{1,0} & & & & & N_{0,3} \\
& & & N_{1,1} & & & & \\
& & & & N_{1,2} & & & \\
& N_{2,0} & & & & & & \\
& & N_{2,1} & & & N_{1,3} & & \\
& & & N_{2,2} & & \vdots & & \\
& N_{3,0} & & & N_{3,1} & & \vdots & \\
& & & \vdots & & & & \\
& & N_{4,0} & & & & & \\
& & & \vdots & & & & \\
& & & & & & &
\end{array}$$

Para uma melhor compreensão do procedimento de cálculo das funções de base é apresentado um exemplo extraído de (PIEGL; TILLER, 1997):

Considere $U=\{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 2, u_5 = 3, u_6 = 4, u_7 = 4, u_8 = 5, u_9 = 5, u_{10} = 5\}$ e $p = 2$. As funções de base de zero, primeiro e segundo grau são calculadas. As funções de base não nulas são mostrados nas Figs. 3.19, 3.20 e 3.21, respectivamente. Nas figuras, no eixo das ordenadas, cada traço representa a quantidade de vezes em que o nó aparece no vetor nó.

$$N_{0,0} = N_{1,0} = 0 \quad \text{para } -\infty < u < \infty$$

$$N_{2,0} = \begin{cases} 1 & 0 \leq u < 1 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{3,0} = \begin{cases} 1 & 1 \leq u < 2 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{4,0} = \begin{cases} 1 & 2 \leq u < 3 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{5,0} = \begin{cases} 1 & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{6,0} = 0 \quad \text{para } -\infty < u < \infty$$

$$N_{7,0} = \begin{cases} 1 & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{8,0} = N_{9,0} = 0 \quad \text{para } -\infty < u < \infty$$

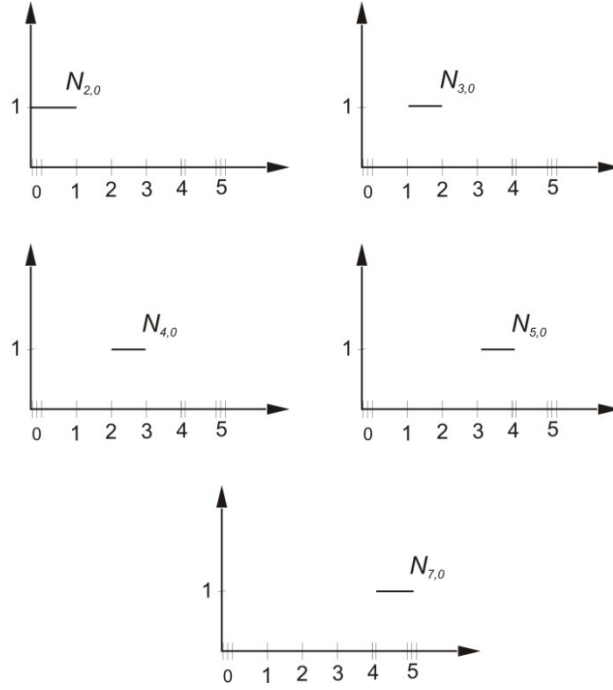


Figura 3.19: Funções base de grau zero, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).

$$N_{0,1} = \frac{u-0}{0-0} N_{0,0} + \frac{0-u}{0-0} N_{1,0} = 0 \quad -\infty < u < \infty$$

$$N_{1,1} = \frac{u-0}{0-0} N_{1,0} + \frac{1-u}{1-0} N_{2,0} = \begin{cases} 1-u & 0 \leq u < 1 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{2,1} = \frac{u-0}{1-0} N_{2,0} + \frac{2-u}{2-1} N_{3,0} = \begin{cases} u & 0 \leq u < 1 \\ 2-u & 1 \leq u < 2 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{3,1} = \frac{u-1}{2-1} N_{3,0} + \frac{3-u}{3-2} N_{4,0} = \begin{cases} u-1 & 1 \leq u < 2 \\ 3-u & 2 \leq u < 3 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{4,1} = \frac{u-2}{3-2} N_{4,0} + \frac{4-u}{4-3} N_{5,0} = \begin{cases} u-2 & 2 \leq u < 3 \\ 4-u & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{5,1} = \frac{u-3}{4-3} N_{5,0} + \frac{4-u}{4-4} N_{6,0} = \begin{cases} u-3 & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{6,1} = \frac{u-4}{4-4} N_{6,0} + \frac{5-u}{5-4} N_{7,0} = \begin{cases} 5-u & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{7,1} = \frac{u-4}{5-4} N_{4,0} + \frac{5-u}{5-5} N_{8,0} = \begin{cases} u-4 & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{8,1} = \frac{u-5}{5-5} N_{8,0} + \frac{5-u}{5-5} N_{9,0} = 0 \quad -\infty < u < \infty$$

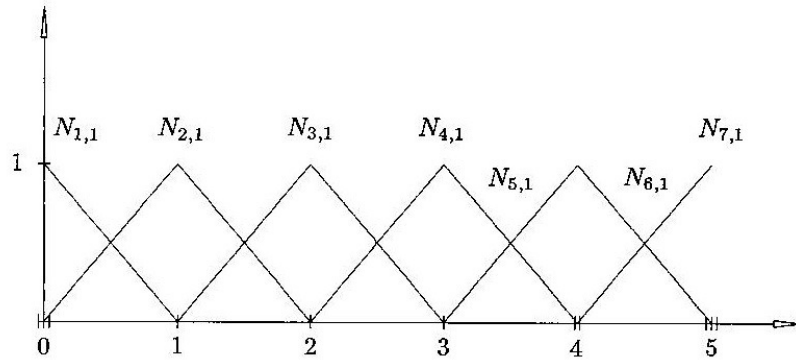


Figura 3.20: Funções base do primeiro grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).

Todos os $N_{i,2}$ seguintes são iguais a zero, exceto nos intervalos especificados, tal que:

$$N_{0,2} = \frac{u-0}{0-0} N_{0,1} + \frac{1-u}{1-0} N_{1,1} = (1-u)^2 \quad 0 \leq u < 1$$

$$N_{1,2} = \frac{u-0}{1-0} N_{1,1} + \frac{2-u}{2-0} N_{2,1} = \begin{cases} 2u - \frac{3}{2}u^2 & 0 \leq u < 1 \\ \frac{1}{2}(2-u)^2 & 1 \leq u < 2 \end{cases}$$

$$N_{2,2} = \frac{u-0}{2-0} N_{2,1} + \frac{3-u}{3-1} N_{3,1} = \begin{cases} \frac{1}{2}u^2 & 0 \leq u < 1 \\ -\frac{3}{2} + 3u - u^2 & 1 \leq u < 2 \\ \frac{1}{2}(3-u)^2 & 2 \leq u < 3 \end{cases}$$

$$N_{3,2} = \frac{u-1}{3-1} N_{3,1} + \frac{4-u}{4-2} N_{4,1} = \begin{cases} \frac{1}{2}(u-1)^2 & 1 \leq u < 2 \\ -\frac{11}{2} + 5u - u^2 & 2 \leq u < 3 \\ \frac{1}{2}(4-u)^2 & 3 \leq u < 4 \end{cases}$$

$$N_{4,2} = \frac{u-2}{4-1} N_{4,1} + \frac{4-u}{4-3} N_{5,1} = \begin{cases} \frac{1}{2}(u-2)^2 & 2 \leq u < 3 \\ -16 + 10u - \frac{3}{2}u^2 & 3 \leq u < 4 \end{cases}$$

$$N_{5,2} = \frac{u-3}{4-3} N_{5,1} + \frac{5-u}{5-4} N_{6,1} = \begin{cases} (u-3)^2 & 2 \leq u < 3 \\ (5-u)^2 & 3 \leq u < 4 \end{cases}$$

$$N_{6,2} = \frac{u-4}{5-4} N_{6,1} + \frac{5-u}{5-4} N_{7,1} = 2(u-4)(5-u) \quad 4 \leq u < 5$$

$$N_{7,2} = \frac{u-4}{5-4} N_{7,1} + \frac{5-u}{5-5} N_{8,1} = (u-4)^2 \quad 4 \leq u < 5$$

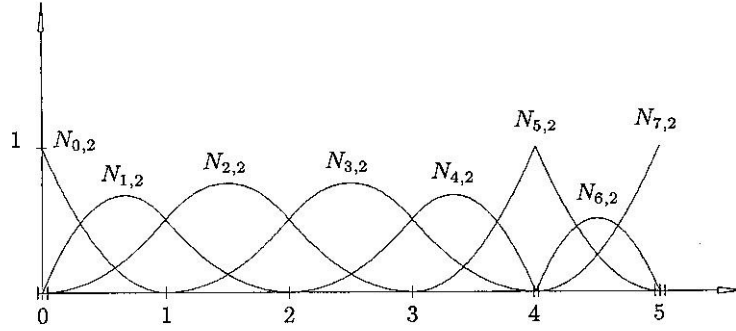


Figura 3.21: : Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).

3.5.2. Derivadas das funções de base B-spline

A derivada da função de base é dada por (PIEGL; TILLER, 1997):

$$N'_{i,p} = \frac{p}{u_{i+p}-u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \quad (3.18)$$

A diferenciação repetida da Eq. (3.18) permite obter a equação geral da derivada de ordem k :

$$N_{i,p}^{(k)}(u) = p \left(\frac{N_{i,p-1}^{(k-1)}}{u_{i+p}-u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1}-u_{i+1}} \right) \quad (3.19)$$

Outra generalização que permite obter a derivada de ordem k de $N_{i,p}(u)$ em termos das funções $N_{i,p-k}(u)$, ..., $N_{i+k,p-k}(u)$ é dada por:

$$N_{i,p}^{(k)} = \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k} \quad (3.20)$$

Onde,

$$a_{0,0} = 1$$

$$a_{k,0} = \frac{a_{k-1,0}}{u_{i+p-k+1}-u_i}$$

$$a_{k,j} = \frac{a_{k-1,j}-a_{k-1,j-1}}{u_{i+p+j-k+1}-u_{i+j}} \quad j = 1, \dots, k-1$$

$$a_{k,k} = \frac{-a_{k-1,k-1}}{u_{i+p+1}-u_{i+k}} \quad (3.21)$$

Deve-se observar que: k não deve exceder p (as derivadas de alta ordem são nulas); os denominadores envolvendo as diferenças entre os nós podem ser nulos (neste caso o quociente é definido como zero).

Uma equação adicional para o cálculo das derivadas das funções de base B-splines, em função da k -ésima derivada de $N_{i,p-1}(u)$ e $N_{i+1,p-1}(u)$ é:

$$N_{i,p}^{(k)}(u) = \frac{p}{p-k} \left(\frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}^{(k)} - \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}^{(k)} \right) \quad k = 0, \dots, p-1 \quad (3.22)$$

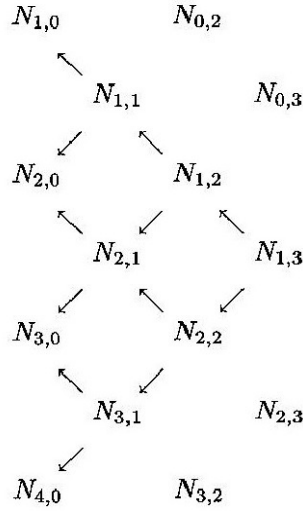
3.5.3. Propriedades importantes das funções de base B-spline

As propriedades das funções de base B-spline que serão apresentadas determinam várias características geométricas desejáveis das curvas B-splines.

Considerando a ordem p e o vetor nó $U = \{u_0, \dots, u_m\}$.

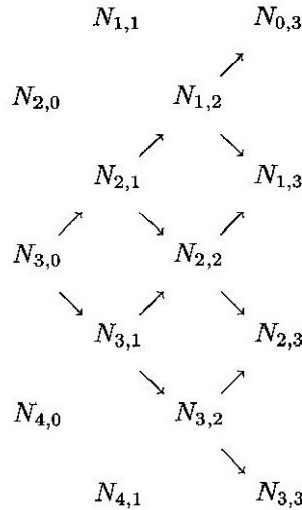
a) Propriedade 1- Propriedade de Suporte Local

$N_{i,p}(u) = 0$ se u é fora do intervalo $[u_i, u_{i+p+1})$. Esta propriedade é ilustrada pelo esquema triangular a seguir. Pode-se observar que $N_{1,3}$ é a combinação de $N_{1,0}$, $N_{2,0}$, $N_{3,0}$ e $N_{4,0}$. Então, $N_{1,3}$ é somente não nula para u pertencente à $[u_1, u_5)$.



b) Propriedade 2

Em um dado intervalo de nó, $[u_j, u_{j+1})$, no máximo $p+1$ das $N_{i,p}(u)$ são não nulas, mais especificamente as funções $N_{j-p,p}, \dots, N_{j,p}$. Por exemplo, no intervalo $[u_3, u_4)$ a única função não nula de ordem zero é $N_{3,0}$. Então, as únicas funções cúbicas não nulas no intervalo $[u_3, u_4)$ são $N_{0,3}, \dots, N_{3,3}$. Esta propriedade é ilustrada pelo triangulo a seguir:



c) Propriedade 3 - Propriedade de Não Negatividade

$N_{i,p}(u) \geq 0$ para todo i, p e u . Esta propriedade pode ser provada pela indução em p . Isso é verdade para $p=0$; Assumindo que também seja verdade para $p-1, p \geq 0$, com i e u arbitrários. Por definição :

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \quad (3.23)$$

Pela *propriedade 1*, $N_{i,p-1}(u) = 0$ se $u \notin [u_i, u_{i+p})$. Mas $u \in [u_i, u_{i+p})$ implica que: $(u - u_i)/(u_{i+p} - u_i)$ seja não-negativa. Tendo assumido que $N_{i,p-1}(u)$ seja não-negativa, então o primeiro termo da Eq.(3.23) é não negativa. O mesmo é verdadeiro para o segundo termo e então os $N_{i,p}(u)$ são não-negativos.

d) Propriedade 4

Para um intervalo arbitrário de nós, $[u_i, u_{i+1})$, $\sum_{j=i-p}^i N_{j,p}(u) = 1$ para todos u pertencente $[u_i, u_{i+1})$ (partição da unidade).

e) Propriedade 5

Todas derivadas de $N_{i,p}(u)$ existem no interior do intervalo de nós (onde ela é uma polinomial). No nó $N_{i,p}(u)$ é $(p-k)$ continuamente diferenciável, onde k é a multiplicidade do nó. Então, aumentando a ordem, aumenta a continuidade, e aumentando a multiplicidade do nó, reduz a continuidade, Fig. 3.22.

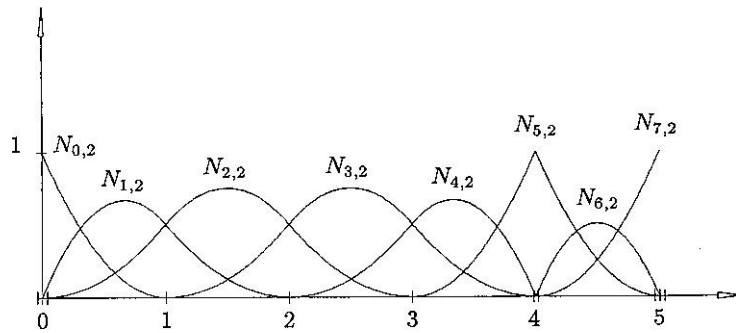


Figura 3.22: Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997).

É importante entender o efeito de múltiplos nós. Considerando as funções $N_{0,2}$, $N_{1,2}$, $N_{2,2}$, $N_{5,2}$ e $N_{6,2}$ da Fig. 3.22 e sabendo que $U=\{0,0,0,1,2,3,4,4,5,5,5\}$. Da Equação 3.17 e da propriedade 1 podemos verificar que estas funções são obtidas nos seguintes intervalos de nós e que são nulas fora destes intervalos:

$$N_{0,2}: \{0,0,0,1\}$$

$$N_{1,2}: \{0,0,1,2\}$$

$$N_{2,2}: \{0,1,2,3\}$$

$$N_{5,2}: \{3,4,4,5\}$$

$$N_{6,2}: \{4,4,5,5\}$$

O termo “multiplicidade” é entendido de duas formas:

- A multiplicidade do nó no vetor nó;
- A multiplicidade de um nó em relação a uma função base específica.

No exemplo, $u=0$ tem multiplicidade 3 no vetor nó U . Mas com relação às funções $N_{0,2}$, $N_{1,2}$, $N_{2,2}$ e $N_{5,2}$, $u=0$ é um nó de multiplicidade 3,2,1 e 0, respectivamente.

Então, a continuidade destas funções em $u=0$ é: $N_{0,2}$ é descontínua; $N_{1,2}$ tem continuidade C^0 ; $N_{2,2}$ tem continuidade C^1 e $N_{5,2}$ é totalmente inafetada, ou seja, todas suas derivadas são nulas em $u=0$, em ambos os lados. $N_{1,2}$ “ve” $u=0$ como um nó duplo, então ela tem continuidade C^0 . $N_{2,2}$ “vê” todos seus nós com multiplicidade 1, então tem continuidade C^1 . Outro efeito da multiplicidade dos nós, como visto pelas funções, é reduzir a quantidade de “intervalos aparentes” nas quais a função é não-nula, ou seja, $N_{6,2}$ é não nula somente em u pertencente ao intervalo $[4,5)$, e tem somente continuidade C^0 em $u=4$ e $u=5$.

Um vetor nó $U=\{u_0, \dots, u_m\}$ é definido como sendo *uniforme* se todos os nós de seu interior são igualmente espaçados, isto é, se existe um número real, d , tal que $d=u_{i+1}-u_i$ para todo $p \leq i \leq m-p-1$. Caso contrário é dito não uniforme.

O exemplo apresentado para $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, $p = 2$, o vetor nó é não uniforme porque existe um nó duplo em $u = 4$. Neste caso, a análise da uniformidade do vetor nó é feita no intervalo $2 \leq i \leq (10-2-1)$, ou seja, $\{0,1,2,3,4,4\}$

f) Propriedade 6

Exceto para o caso em que $p=0$, $N_{i,p}(u)$ possui exatamente um valor de máximo.

g) Propriedade 7

Um vetor nó na forma $U = \{0, \dots, 0, 1, \dots, 1\}$, tal que, a quantidade de zeros e de uns é $p+1$, leva à obtenção das polinomiais de Bearstein de grau p .

h) Propriedade 8

Seja $m+1$ o número de nós. Então, existem $n+1$ funções de base, sendo $n=m-p-1$; $N_{0,p}(a) = 1$ e $N_{n,p}(b) = 1$. Por exemplo, $N_{0,p}(a) = 1$ vem do fato de que $N_{0,0}, \dots, N_{p-1,0} = 0$, desde que isto implica que $N_{0,p}(a) = N_{p,0}(a) = 1$. Da propriedade 4 sabe-se que $N_{i,p}(a) = 0$ para $i \neq 0$, e $N_{i,p}(b) = 0$ para $i \neq m$.

3.5.4. Curvas B-spline

A função B-spline é um ajustador de aproximação, pois a curva gerada não passa necessariamente pelos pontos de controle, diferentemente da spline que realiza a interpolação dos pontos dados, como mostrado nas Figs. 3.17 e 3.18.

Curvas B-splines utilizam o mesmo conceito de curva de ponderação para a mistura dos pontos de controle, isto é, a curva é obtida pelo somatório das funções de base não nulas que são multiplicadas pelos pontos de controle, Eq. (3.24).

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad a \leq u \leq b \quad (3.24)$$

Onde P_i ($i = 0, \dots, n$) são os pontos de controle e $N_{i,p}(u)$ são as funções de base de grau p definidas no vetor de nós.

De acordo com (PIEGL; TILLER, 1997), para o cálculo de uma curva B-spline são necessárias três etapas:

- Encontrar o vão em que u está no vetor de nós U (Algoritmo no Anexo II);
- Calcular as funções base não nulas (Algoritmo no Anexo III);
- Multiplicar as funções base não nulas pelos seus respectivos pontos de controle (Algoritmo no Anexo IV).

3.5.5. Propriedades das curvas B-splines

De acordo com (PIEGL; TILLER, 1997), (DE BOOR, 2000) e (ROGERS, 2001), as propriedades mais importantes da B-spline são:

a) Se $n = p$ e $U = \{0, \dots, 0, 1, \dots, 1\}$ então $C(u)$ é uma curva de Bézier, Fig. 3.23.

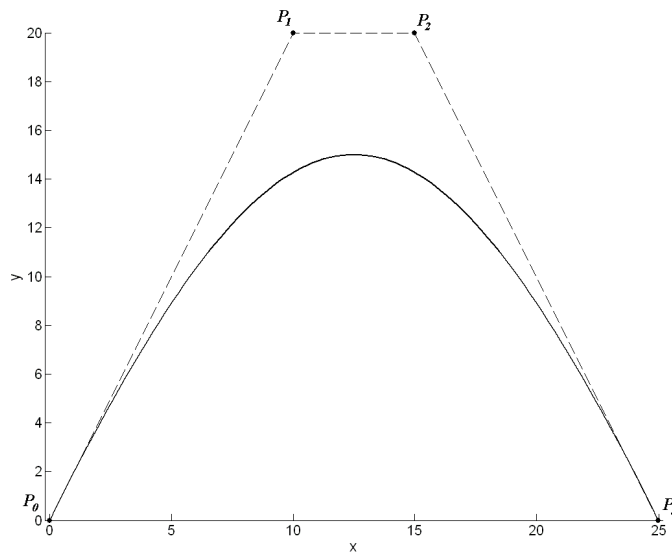


Figura 3.23: B-spline cúbica com $U = \{0, \dots, 0, 1, \dots, 1\}$, i.e., Curva de Bézier.

b) $C(u)$ é uma polinomial por partes; o grau p , o número de pontos $n+1$ e o número de nós $m+1$ são relacionados da seguinte forma:

$$m = p + n + 1 \quad (3.25)$$

c) Os pontos dos extremos de $C(u)$ são iguais aos pontos de controle: $C(0) = P_0$ e $C(1) = P_n$.

d) Propriedade do fecho convexo: fecho convexo é o polígono formado a cada $p+1$ pontos de controle da curva. Esta propriedade garante que a curva sempre estará no seu interior como mostrado na Fig. 3.24.

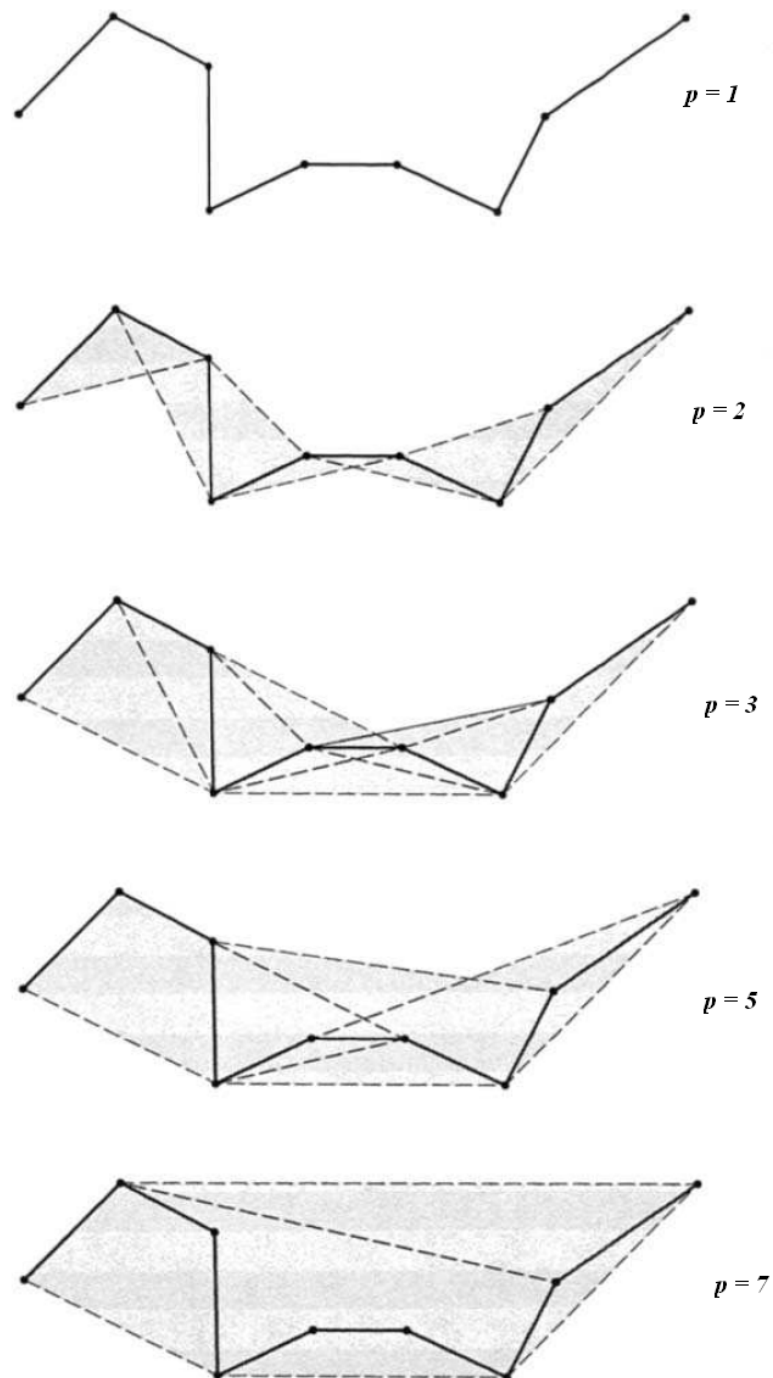


Figura 3.24: Propriedade do fecho convexo, (ROGERS, 2001).

e) Suporte local: com a modificação de um ponto P_i da B-spline, a mudança na curva será somente no intervalo $[u_i, u_{i+p+1})$. Esta região corresponde a porção da curva definida entre os $(p+1)/2$ segmentos de retas a esquerda e a direita do P_i modificado. Na Figura 3.25 é mostrada a propriedade do suporte local para uma B-spline cúbica com a modificação do ponto P_4 .

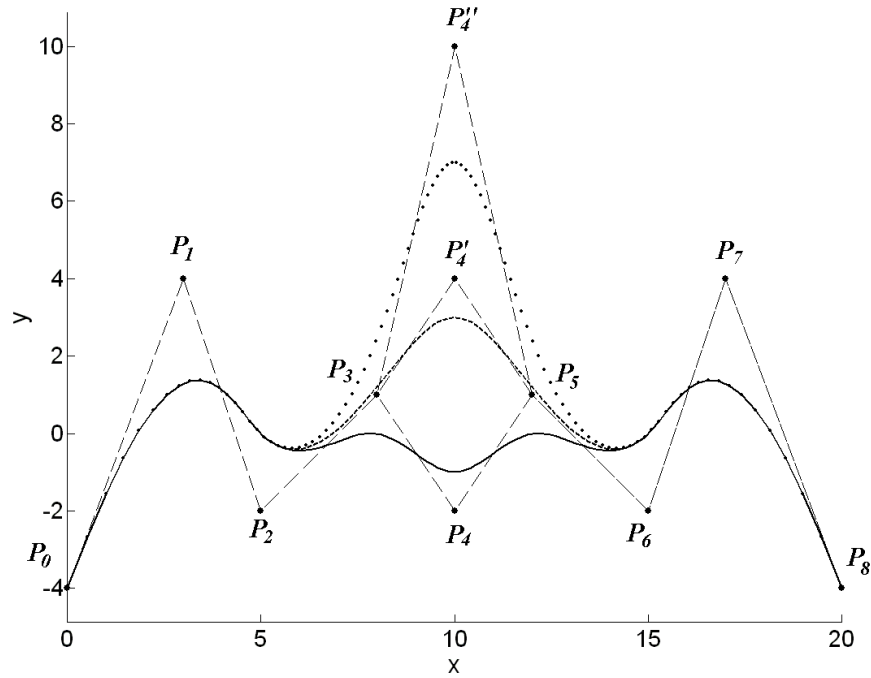


Figura 3.25: Controle local da B-spline.

f) A continuidade de $C(u)$ depende da continuidade de $N_{i,p}(u)$. No interior do intervalo de nós (u_i, u_{i+1}) a curva é infinitamente diferenciável e é $p-k$ vezes diferenciável nos nós com multiplicidade k .

3.5.6. Derivada da curva B-spline

A derivada da curva B-spline é obtida por meio da derivada da função de base e é dada por:

$$C^{(k)}(u) = \sum_{i=0}^n N_{i,p}^{(k)}(u) P_i \quad (3.26)$$

A Figura 3.26 mostra uma B-spline de quinta ordem e sua derivada primeira, segunda e terceira são apresentadas em Figs. 3.27, 3.28 e 3.29, respectivamente.

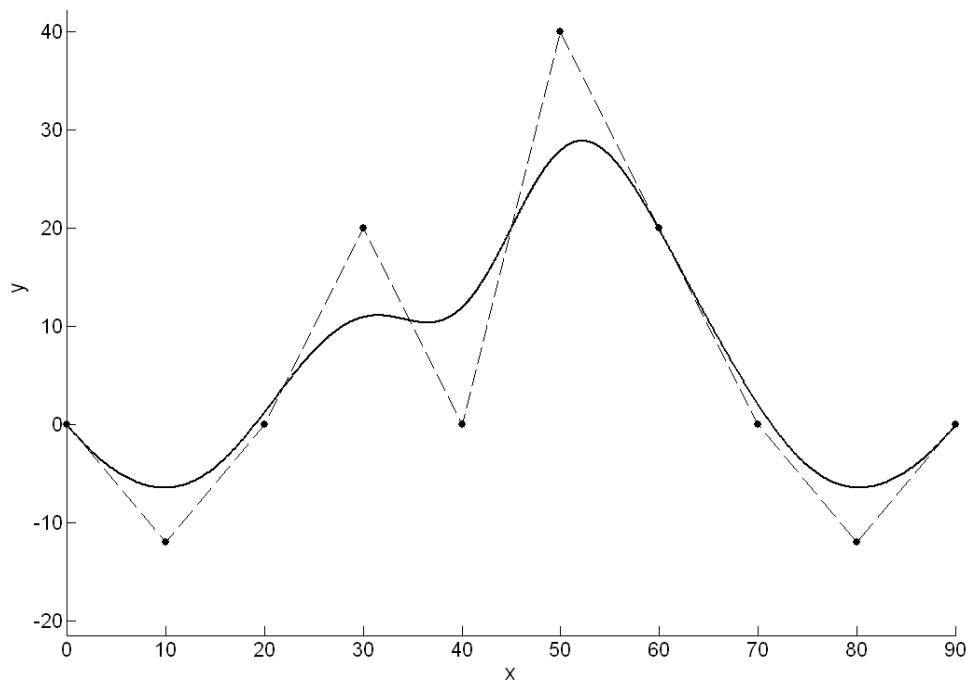


Figura 3. 26: B-spline do quarto grau.

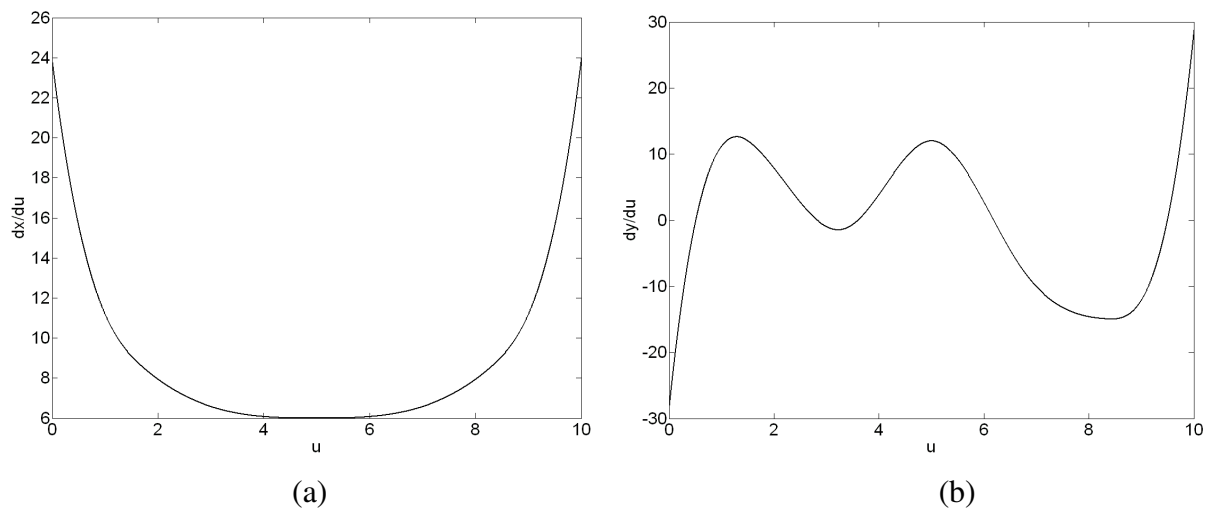


Figura 3.27: Primeira derivada da curva em relação a u , (a) componente em x , (b) componente y .

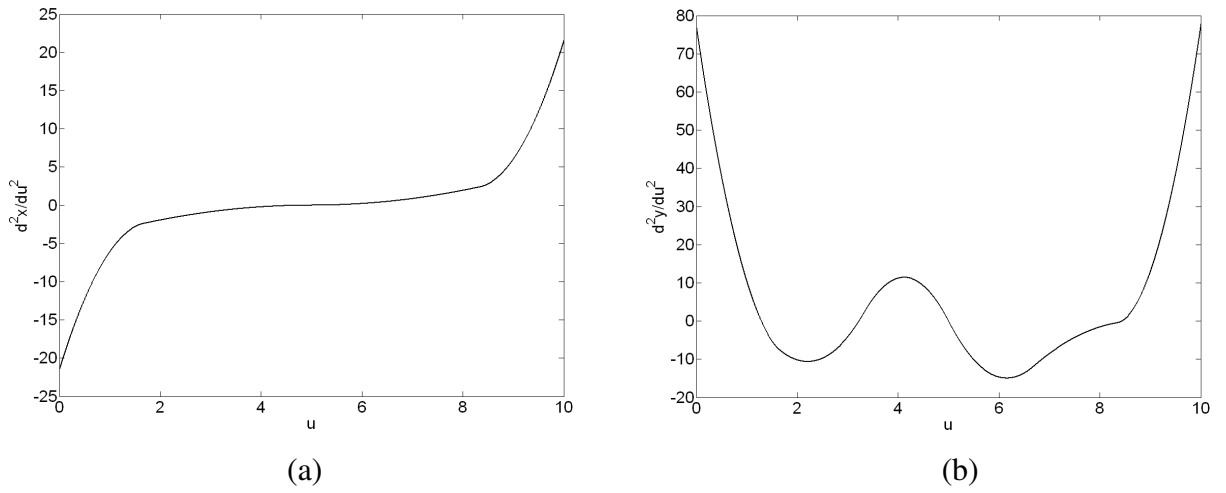


Figura 3.28: Segunda derivada da curva em relação a u , (a) componente em x , (b) componente y .

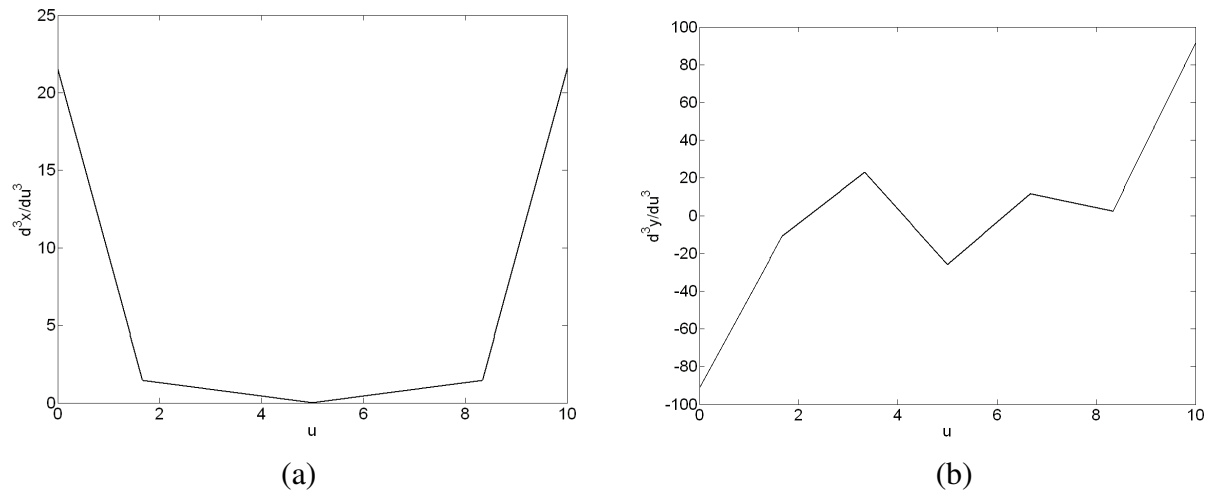


Figura 3.28: Terceira derivada da curva em relação a u , (a) componente em x , (b) componente y .

Como apresentado nas propriedades da B-spline, a continuidade de $C(u)$ depende da continuidade de $N_{i,p}(u)$. No interior do intervalo de nós (u_i, u_{i+1}) a curva é infinitamente diferenciável e é $(p-k)$ vezes diferenciável nos nós com multiplicidade k . Dessa forma, a curva de quarto grau pode ser derivada 3 vezes em relação a seu parâmetro.

3.5.7. Interpolação com B-spline

Sabe-se que quando se define uma B-spline ela não passa pelos pontos de controle. Porém, se for necessário que a curva passe por pontos pré-determinados é preciso calcular um conjunto de pontos de controle e um novo vetor nó a partir dos pontos dados, para poder traçar a curva.

Considerando que seja conhecido um conjunto de pontos $\{Q_k\}$, ($k = 0, \dots, n$) e deseja obter uma B-spline que passe por eles. Se para cada Q_k da curva existe um nó \bar{u}_k correspondente, o vetor nó $U = \{u_0, \dots, u_m\}$ pode ser escolhido para o cálculo das funções base. Assim, um sistema de $(n+1)$ equações pode ser resolvido por:

$$Q_k = C(\bar{u}_k) = \sum_{i=1}^n N_{i,p}(\bar{u}_k) P_i \quad (3.27)$$

Os pontos $\{P_i\}$ são as $(n+1)$ incógnitas.

A partir dos pontos de controle obtidos calcula-se a B-spline normalmente com a Eq. (3.24). Dessa forma a curva resultante passará por todos os pontos $\{Q_i\}$.

Um dos problemas a ser resolvido é a escolha de u_k e U , visto que suas escolhas afetam a forma da curva. A seguir são apresentadas algumas metodologias para a definição dos nós u_k :

- **Igualmente espaçados:**

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1$$

$$\bar{u}_k = \frac{k}{n} \quad k = 1, \dots, n-1$$

Esse método não é recomendado, pois pode causar formas indesejadas da B-spline.

- **Comprimento de corda:**

Suponha d o comprimento total de corda, dado por:

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}| \quad (3.28)$$

Então,

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad k = 1, \dots, n-1 \quad (3.29)$$

Este é o método mais utilizado, pois permite obter uma curva mais comportada.

- **Método Centrípeto**

Considerando que,

$$d_c = \sum_{k=1}^n \sqrt{|Q_k - Q_{k-1}|} \quad (3.30)$$

Então,

$$\bar{u}_0 = 0 \quad \bar{u}_n = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d_c} \quad k = 1, \dots, n-1 \quad (3.31)$$

Cada parâmetro de interpolação \bar{u}_k ($k = 0, \dots, n$) corresponde a um ponto Q_k por onde a B-spline deve passar. Para obter a B-spline, é necessário calcular os pontos de controle P_i ($i = 0, \dots, n$) que definem o fecho convexo, obtidos pela Eq. (3.27). Para o traçado da B-spline é necessário então, definir o vetor nó $U = \{u_0, \dots, u_m\}$ que permite obter as funções base. Dois métodos têm sido utilizados para a definição do vetor nó. O primeiro consiste em uma distribuição uniforme, ou seja:

$$u_0 = \dots = u_p = 0 \quad u_{m-p} = \dots = u_m = 1$$

$$u_{j+p} = \frac{j}{n-p+1} \quad j = 1, \dots, n-p \quad (3.32)$$

O segundo método é denominado *averaging* e que, a partir do parâmetro de interpolação \bar{u}_k , pode ser obtido por:

$$\begin{aligned} u_0 = \dots = u_p = 0 & \quad u_{m-p} = \dots = u_m = 1 \\ u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i & \quad j = 1, \dots, n-p \end{aligned} \quad (3.33)$$

Na Figura 3.30 é apresentado um exemplo de interpolação dos pontos $\{Q_i\}$ ($i = 0, \dots, 9$). O vetor \bar{U} foi obtido com a parametrização por comprimento de corda e o vetor de nós U calculado pelo método *averaging*.

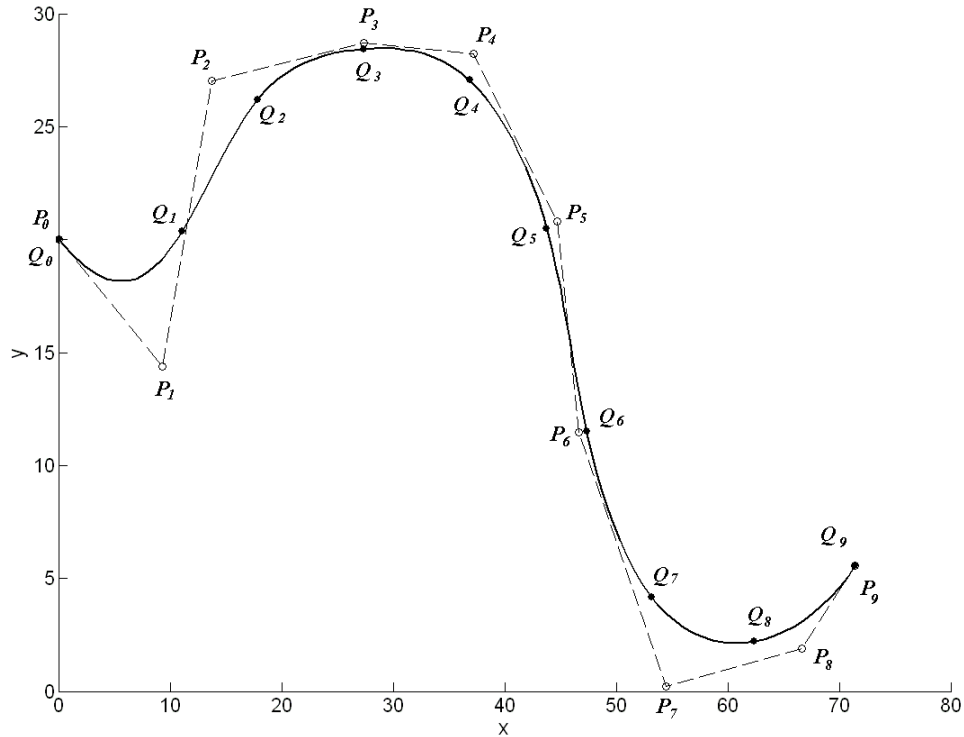


Figura 3.29: Interpolação por B-spline cúbica usando parametrização por comprimento de corda e vetor nós obtido pelo método de *averaging*.

A parametrização do vetor \bar{U} , assim como a ordem, possuem influência na forma da curva. Isso pode ser observado na Fig. (3.31) em que é realizada a interpolação para os três tipos de parametrização mais conhecido, sendo que o vetor U é obtido por *averaging*. Na a parametrização de \bar{U} com nós igualmente espaçados a curva possui comportamento indesejado em planejamento de trajetórias robóticas. Com esta metodologia a curva torna-se

menos suave; em regiões que os pontos estão próximos, a curva tende a ser “encolhida”, o que provoca curvaturas mais fechadas; em regiões que possuem pontos mais distantes a curva tende a ser “esticada”. A parametrização centrípeta garante uma curva suave, porém com a parametrização por comprimento de corda a curva possui um melhor comportamento.

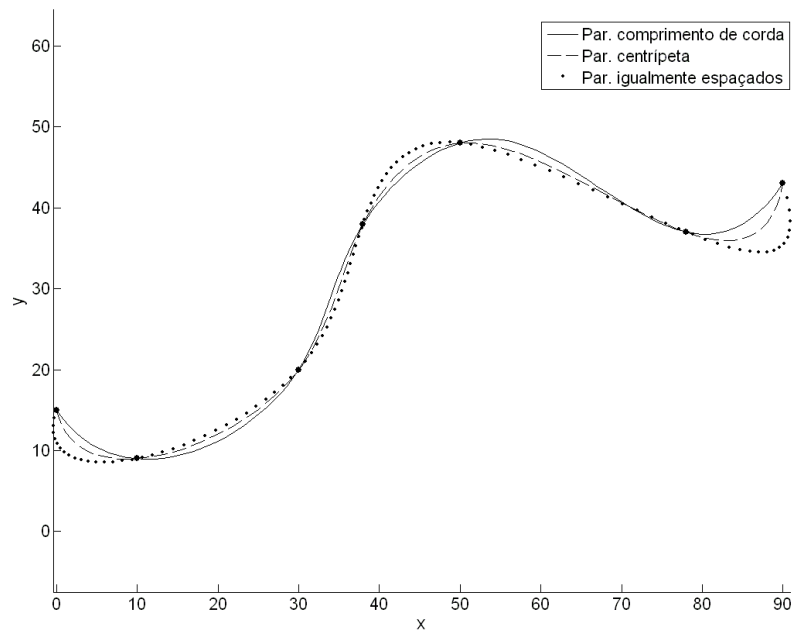


Figura 3.30: Interpolação com diferentes metodologias de parametrização do vetor \bar{U} .

Um elemento muito importante no formato da curva é a escolha do vetor nó U , o qual é responsável pela obtenção das funções de base. Na Figura 3.32 são apresentadas as curvas obtidas pelos mesmos pontos da Fig. 3.31. Neste exemplo, o vetor \bar{U} é obtido com parametrização por comprimento de corda, e o vetor U é calculado igualmente espaçado e utilizando a técnica *averaging*.

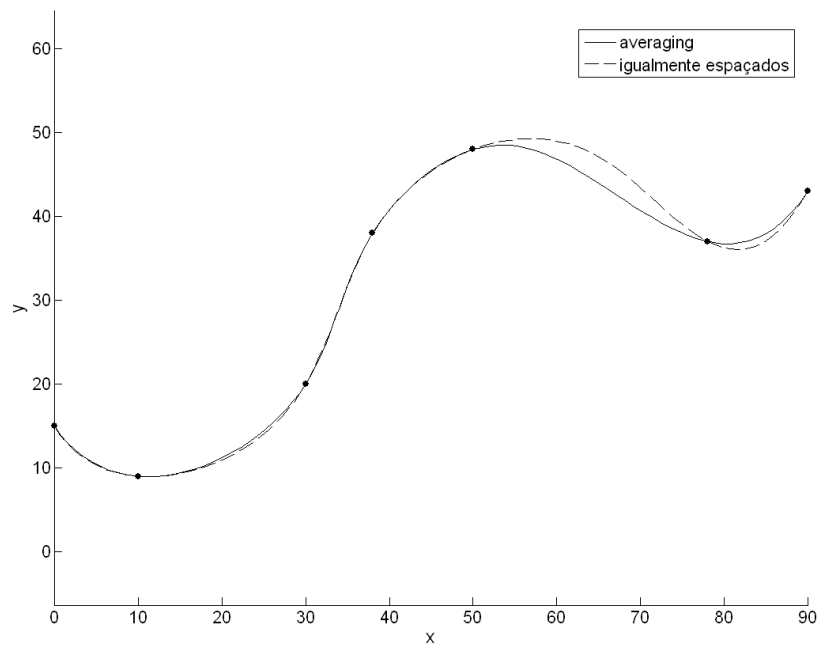


Figura 3.31: Interpolação com diferentes metodologias de obtenção do vetor U .

3.5.8. Interpolação com especificação das derivadas nos extremos

O procedimento é similar ao de interpolação sem derivada:

- 1) Cálculo de \bar{u}_k correspondente aos pontos Q_k ;
- 2) Cálculo do vetor nó U ;
- 3) Avaliação da função base para construir o sistema de equações, com os pontos de controle como variáveis;
- 4) Resolução do sistema.

Cada derivada dada aumenta um nó e um ponto de controle, conseqüentemente aumenta mais uma equação no sistema linear. As equações das derivadas são usadas para encontrar as equações adicionais.

Considerando $\{Q_k\}$, $k = 0, \dots, n$, sendo os pontos e assumindo que D_0 e D_n são as derivadas nos extremos da curva, os pontos podem ser interpolados com uma curva de p -ésimo grau, de acordo com a equação:

$$Q_k = C(\bar{u}_k) = \sum_{i=0}^{n+2} N_{i,p}(\bar{u}_k) P_i \quad (3.34)$$

Que resulta em um sistema linear de $(n+3)$ incógnitas e $(n+1)$ equações.

Como antes, \bar{u}_k , $k = 0, \dots, n$, pode ser calculado por nós igualmente espaçados, comprimento de corda ou método centrípeto.

Utilizando as Equações (3.20) e (3.26), tem-se que para uma B-spline, as derivadas nos extremos são dadas por:

$$C'(0) = \frac{p}{u_{p+1}} (P_1 - P_0) \quad (3.35)$$

$$C'(1) = \frac{p}{1-u_{m-p-1}} (P_{n+2} - P_{n+1}) \quad (3.36)$$

Dessa forma pode-se reescrever:

$$P_1 - P_0 = \frac{u_{p+1}}{p} D_0 \quad (3.37)$$

$$P_{n+1} - P_{n+2} = \frac{1-u_{m-p-1}}{p} D_n \quad (3.38)$$

Onde D_0 e D_n são as derivadas no ponto inicial e final da interpolação.

As Equações (3.37) e (3.38) são utilizadas juntamente com a Eq. (3.34), para completar o sistema de $(n+3)$ equações e $(n+3)$ incógnitas P_i . Na Figura 3.33 é apresentada a interpolação com derivadas nulas nos extremos.

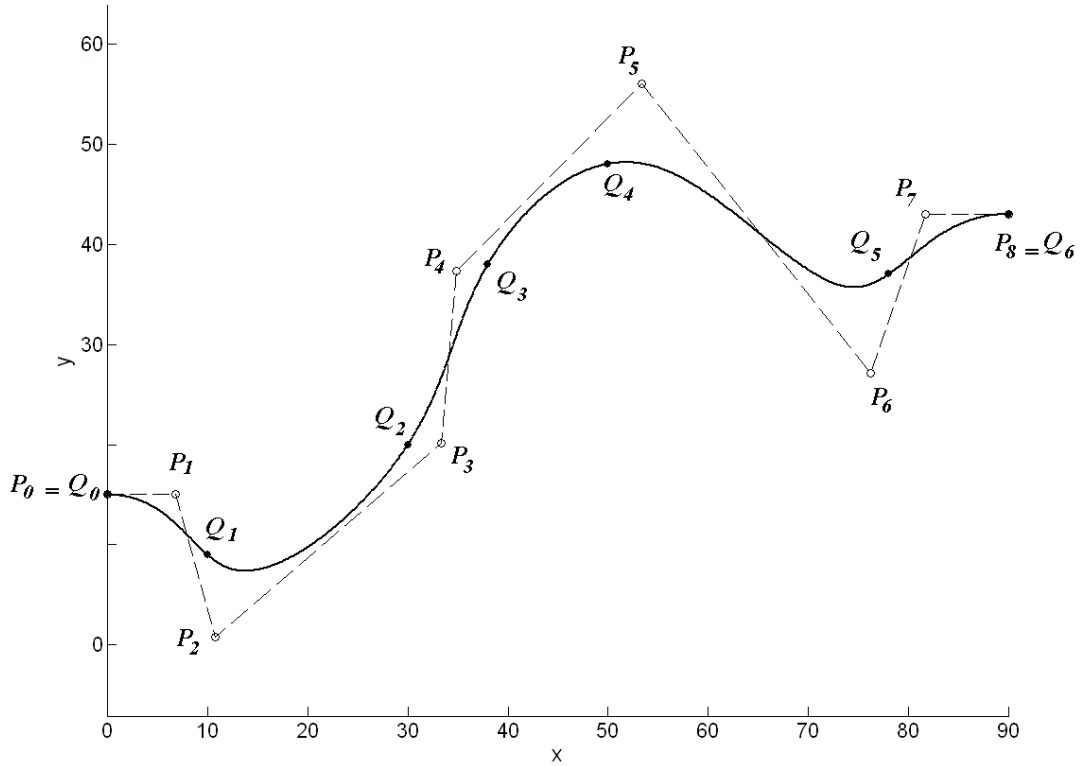


Figura 3.32: Interpolação com B-spline cúbica com derivadas nos extremos igual a zero.

3.5.9. Alteração de curvas B-spline

Em uma spline, as formas que existem de mudar o perfil da curva são os pontos de controle e sua ordem. A B-spline permite a mudança do seu perfil garantindo o controle local, com os pontos de controle, o vetor nó que conseqüentemente muda as funções base, a ordem da curva e a definição da derivada em cada ponto a qual não será abordada neste trabalho.

3.6. NURBS (Non Uniform Rational B-spline)

De acordo com Piegl e Tiller (1997) a NURBS de grau p é definida como:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \omega_i P_i}{\sum_{i=0}^n N_{i,p}(u) \omega_i} \quad a \leq u \leq b \quad (3.39)$$

Onde $\{P_i\}$ são os pontos que formam o polígono de controle, $\{\omega_i\}$ são os pesos e $\{N_{i,p}(u)\}$ são as funções de base de grau p definidas por um vetor de nós não periódicos $U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$. E assumindo que $a = 0$, $b = 1$ e $\omega_i > 0$ para todo i .

A NURBS possui uma função de base racional, obtida por meio de uma relação entre funções de base de B-spline, que é dada por:

$$R_{i,p}(u) = \frac{N_{i,p}(u)\omega_i}{\sum_{i=0}^n N_{i,p}(u)\omega_i} \quad (3.40)$$

Assim, a Eq.(3.39) pode ser reescrita da seguinte forma:

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i \quad (3.41)$$

A função base racional $R_{i,p}(u)$ possui as propriedades:

- 1- Não negatividade: $R_{i,p}(u) \geq 0$ para todo i, p e $u \in [0; 1]$;
- 2- Partição de unidade: $\sum_{i=0}^n R_{i,p}(u) = 1$ para todo $u \in [0; 1]$;
- 3- $R_{0,p}(0) = R_{n,p}(1) = 1$;
- 4- Para $p > 0$, todo $R_{i,p}(u)$ atinge exatamente um máximo no intervalo $u \in [0; 1]$;
- 5- Suporte Local: $R_{i,p}(u) = 0$ para $u \notin [u_i, u_{i+p+1})$. Além disso, em qualquer intervalo de nós (vão) dado, $R_{i,p}(u)$ será não nula no máximo $p+1$ vãos.
- 6- Todas as derivadas de $R_{i,p}(u)$ existem no interior do vão de nós. No nó, $R_{i,p}(u)$ é $p-k$ vezes diferenciável, onde k é a multiplicidade do nó.
- 7- Se $\omega_i = 1$ para todo i , então $R_{i,p}(u) = N_{i,p}(u)$ para todo i .
- 8- $C(0) = P_0$ e $C(1) = P_n$;

9- Propriedade do fecho convexo: se $u \in [u_i; u_{i+1})$, então $C(u)$ está dentro do fecho convexo;

10- Todas as derivadas de $C(u)$ existem no interior do vão de nós. No nó, $C(u)$ é $p-k$ vezes diferenciável, onde k é a multiplicidade do nó.

11- Aproximação Local: Se o ponto de controle P_i é movido, ou o peso ω_i é trocado, isso afeta somente a porção da curva no intervalo $u \in [u_i; u_{i+p+1})$.

A propriedade 11 é muito importante para a alteração de forma da curva. A Figura 3.33 mostra o efeito da modificação do peso. Assumindo um intervalo $u \in [u_i; u_{i+p+1})$, com o aumento de ω_i a curva tende a se aproximar do ponto P_i , já com a redução do fator peso a curva se afasta.

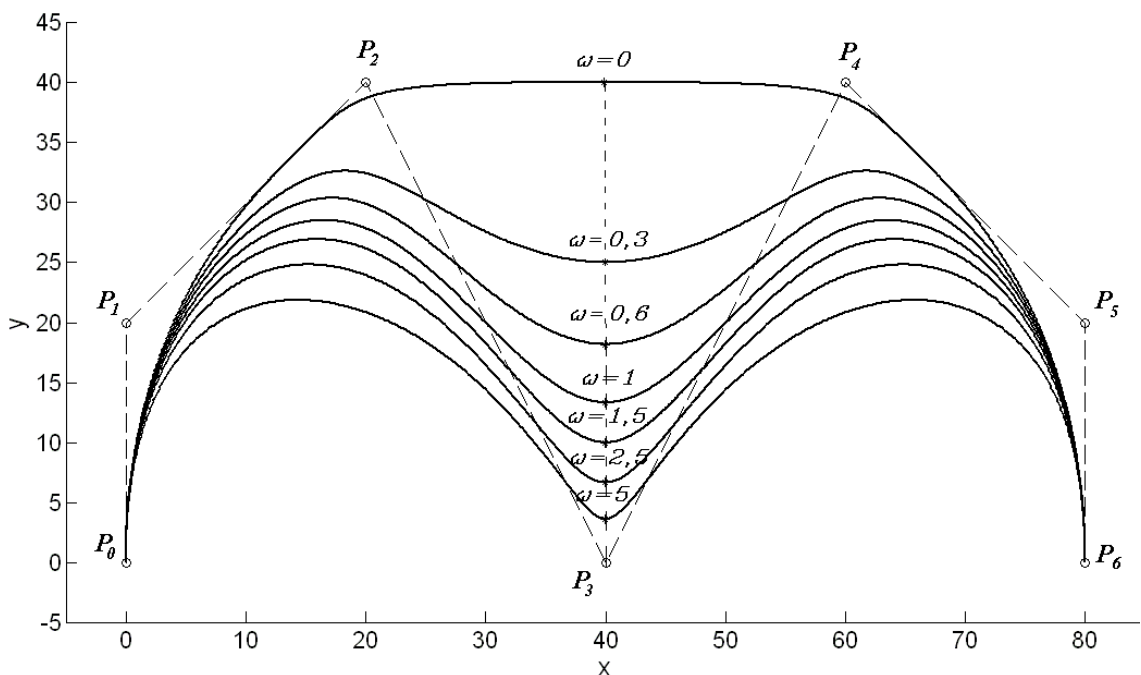


Figura 3.33: Influência do peso no comportamento da curva.

Pode ser observado um comportamento interessante na NURBS, conforme ω_i é alterado para um dado u fixo. Os pontos correspondentes a cada ω_i são colineares, isso pode ser observado com a linha tracejada entre $C(u, \omega=0)$ e P_3 .

A reta formada pelos pontos correspondentes a pesos diferentes passa por P_3 , e para $0 \leq \omega \leq \infty$ o ponto da curva sempre estará sobre a reta. Para $\omega = \infty$, a curva passará sobre P_3 , e para $\omega = 0$ o ponto P_3 não possui influência sobre a curva.

Há três formas de alterar a forma da curva NURBS: mudança do ponto de controle, alteração do vetor nó e mudança do vetor peso. Alteração da curva por meio da mudança do vetor nó é muito complicada, visto que não é intuitivo, assim os métodos mais utilizados são mudança do vetor peso e dos pontos de controle.

3.7. Conclusão

Neste capítulo foram apresentadas as formas de planejamento de trajetórias robóticas definidas a partir de um conjunto de pontos que permitem definir curvas por meio de interpolação ou aproximação. Em qualquer situação, a trajetória deve ser a mais suave possível.

Uns dos fatores que garante a suavidade de uma trajetória robótica é seu sinal de referência. Se o sinal for suave a trajetória será suave. Dessa forma, é importante especificar derivadas nulas no início e no final de uma trajetória definida no espaço das articulações.

Tendo em vista a suavidade da curva, a interpolação polinomial de Hermite pode ser utilizada em planejamento de trajetórias. Porém, não poderá ser utilizada para definir trajetórias multipontos, pois com a instabilidade e a divergência o robô não executará a tarefa da forma desejada e há o risco de colisão em seu ambiente de trabalho.

Para evitar problemas numéricos em trajetórias multipontos, a spline pode ser definida com a continuidade necessária tal que a trajetória seja suave. Em robótica industrial a spline cúbica é comumente usada, pois utiliza sistemas de equações relativamente fáceis de serem resolvidos e apresenta continuidade de posição, velocidade e aceleração. Mas a alteração de um ponto de controle afeta o perfil de toda a curva.

Uma alternativa consiste em utilizar a B-spline que possui o mesmo grau de suavidade, além de garantir controle local da curva. Isso permite uma mudança da trajetória somente em uma região determinada, ao contrário da spline que altera o perfil de todo o trajeto.

Curvas NURBS possuem maior flexibilidade que as B-splines, pois além dos pontos de controle a curva pode ser alterada com a mudança de pesos. Em trajetórias robóticas não é

justificável a interpolação com NURBS, visto que se o peso relacionado a um ponto de controle for alterado, a curva não realizará interpolação. Portanto, a NURBS é mais adequada para aproximação. A curva NURBS é interessante em trajetórias com a presença de obstáculos móveis devido sua grande flexibilidade o que permite o desvio com maior facilidade.

CAPÍTULO IV

ANÁLISE DA TRAJETÓRIA DESCRITA POR UM ROBÔ INDUSTRIAL

4.1. Introdução

Atualmente, os robôs industriais executam desde tarefas pesadas como soldagem de chassis de caminhão, a pequenas incisões cirúrgicas na medicina. Por mais que estas tarefas sejam diferentes, o planejamento da trajetória dos robôs é realizado utilizando as mesmas metodologias.

O planejamento de trajetórias de robôs pode ser realizado no domínio das articulações ou da tarefa. Em trajetórias planejadas no domínio das articulações, um conjunto de pontos ao longo da trajetória deve ser escolhido e transformado em coordenadas articulares por meio do modelo cinemático inverso. A partir das coordenadas articulares, utiliza-se um método de interpolação para a geração do sinal de referência para cada atuador. Assim, o controlador comanda os atuadores para seguir os sinais de referência obtidos pela interpolação. Neste método de planejamento de trajetória, o robô passa pelos pontos definidos, porém não é possível prever a trajetória entre eles.

Para que o robô siga exatamente um caminho especificado, sua trajetória deve ser planejada no domínio da tarefa. Neste caso, a trajetória do robô deve ser definida por uma função geométrica que, em geral, é descrita do espaço Cartesiano. Apesar da trajetória ser descrita no espaço Cartesiano, o controle do robô é realizado sobre os atuadores. Dessa forma, o cálculo da cinemática inversa deve ser realizado ao longo de toda a função geométrica que define a trajetória.

Em trajetórias robóticas contínuas é comum utilizar spline cúbica para seu planejamento, pois apresenta continuidade de posição, velocidade e aceleração. Além disso, os coeficientes

Os eixos e sentido de movimentação do robô Motoman HP6 estão representados na Fig. 4.2 e as especificações técnicas do HP6 são apresentadas na Tab.1.

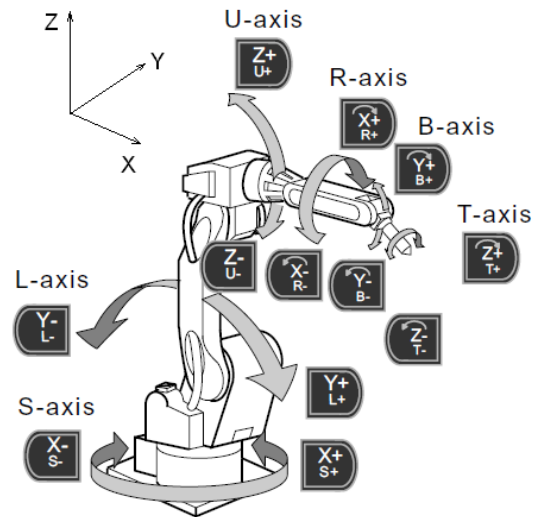


Figura 4.2: Eixos de movimentação do Motoman HP6 (MOTOMAN, 2009).

Tabela 1: Especificações técnicas do robô Motoman HP6 (MOTOMAN, 2009).

ROBOT SPECIFICATIONS		
		HP6 Standard
Structure		Vertical jointed-arm type
Controlled Axes		6
Payload		6 kg (13.2 lbs.)
Vertical Reach		2,403 mm (94.6")
Horizontal Reach		1,378 mm (54.25")
Repeatability		±0.08 mm (0.003")
Maximum Motion Range	S-Axis (Turning/Sweep)	±170°
	S-Axis (Wall Mount)	±30°
	L-Axis (Lower Arm)	+155°/-90°
	U-Axis (Upper Arm)	+250°/-175°
	R-Axis (Wrist Roll)	±180°
	B-Axis (Bend/Pitch/Yaw)	+225°/-45°
	T-Axis (Wrist Twist)	±360°
Maximum Speed	S-Axis	150°/s
	L-Axis	160°/s
	U-Axis	170°/s
	R-Axis	340°/s
	B-Axis	340°/s
	T-Axis	520°/s
Approximate Mass		130 kg (286.7 lbs.)
Brakes		All axes
Power Consumption		1.5 kVA
Allowable Moment	R-Axis	11.8 N • m
	B-Axis	9.8 N • m
	T-Axis	5.9 N • m
Allowable Moment of Inertia	R-Axis	0.24 kg • m²
	B-Axis	0.17 kg • m²
	T-Axis	0.06 kg • m²
Internal User I/O Cable		16 conductors + ground
Internal User Air Line		PT 3/8" connector

4.3. Modelo cinemático direto

Para obter o modelo cinemático do robô é utilizado o método das coordenadas equipolentes. Conforme apresentado em (TSAI, 1999), a matriz de transformação homogênea é dada por:

$${}^A T_B = \begin{bmatrix} {}^A R_B(3 \times 3) & \vdots & {}^A p(3 \times 1) \\ \dots\dots & \vdots & \dots\dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad (4.1)$$

Onde:

${}^A R_B$ é a matriz que representa a orientação do referencial A em relação ao B que pode ser escrita como:

Rotação de ψ em torno do eixo x

$${}^A R_B(x, \psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix} \quad (4.2)$$

Rotação de φ em torno do eixo y

$${}^A R_B(y, \varphi) = \begin{bmatrix} c\varphi & 0 & s\varphi \\ 0 & 1 & 0 \\ -s\varphi & 0 & c\varphi \end{bmatrix} \quad (4.3)$$

Rotação θ em torno do eixo z

$${}^A R_B(z, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

E, ${}^A p$ é o vetor que descreve as coordenadas da posição da origem do referencial B em relação ao referencial A .

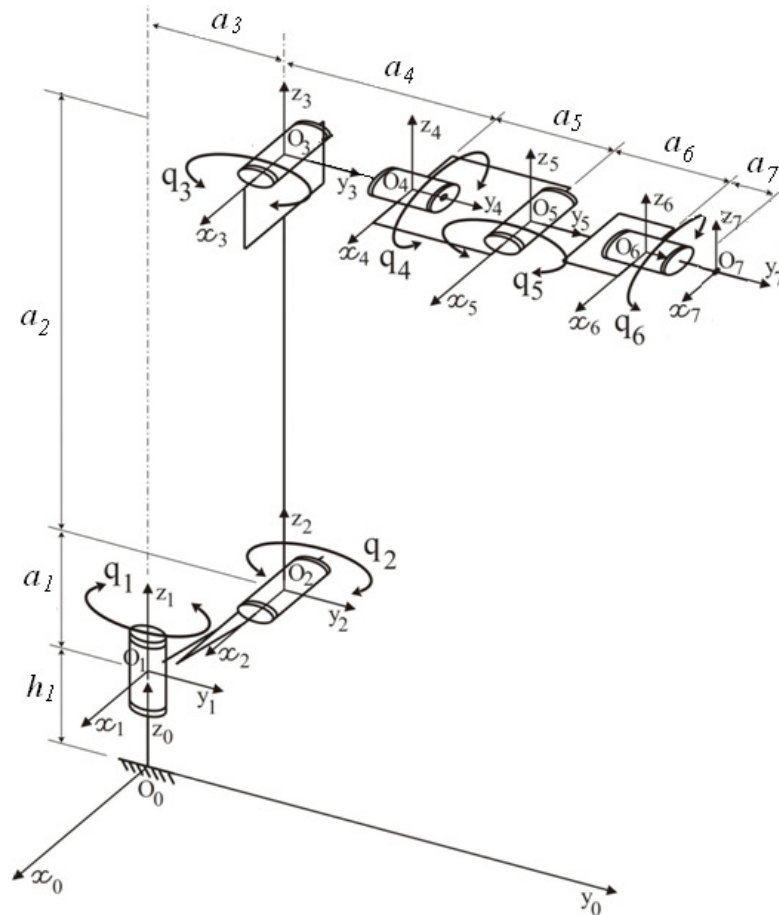


Figura 4.3: Diagrama unifilar do robô HP6 com seus respectivos referenciais.

Sendo as matrizes homogêneas entre cada referencial dadas por:

$$\begin{aligned} {}^0T_1 &= \begin{bmatrix} cq_1 & -sq_1 & 0 & 0 \\ sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & ; & \quad {}^1T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_2 & -sq_2 & a_3 \\ 0 & sq_2 & cq_2 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^2T_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_3 & -sq_3 & 0 \\ 0 & sq_3 & cq_3 & a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & ; & \quad {}^3T_4 = \begin{bmatrix} cq_4 & 0 & sq_4 & 0 \\ 0 & 1 & 0 & a_4 \\ -sq_4 & 0 & cq_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

$${}^4T_5 = \begin{bmatrix} cq_5 & -sq_5 & 0 & 0 \\ sq_5 & cq_5 & 0 & a_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad {}^5T_6 = \begin{bmatrix} cq_6 & 0 & sq_6 & 0 \\ 0 & 1 & 0 & a_6 \\ -sq_6 & 0 & cq_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^6T_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a_7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

A transformação entre o referencial associado ao elemento terminal e a base é dado por:

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \quad (4.6)$$

A partir Eq. (4.6) as coordenadas Cartesianas relativas ao elemento terminal são dadas por:

$$x = sq_1 \{-c(q_2 + q_3)[cq_5(a_6 + a_7) + a_4 + a_5] + s(q_2 + q_3)a_3 + sq_2a_2 - a_1\} + sq_5(a_6 + a_7)[cq_1sq_4 + cq_4sq_1s(q_2 + q_3)] \quad (4.7)$$

$$y = cq_1\{c(q_2 + q_3)[cq_5(a_6 + a_7) + a_4 + a_5] - s(q_2 + q_3)a_3 - sq_2a_2 + a_1\} + sq_5(a_6 + a_7)[sq_1sq_4 - cq_4cq_1s(q_2 + q_3)] \quad (4.8)$$

$$z = s(q_2 + q_3)[cq_5(a_6 + a_7) + a_4 + a_5] + c(q_2 + q_3)[cq_4sq_5(a_6 + a_7) + a_3]cq_2a_2 + h_1 \quad (4.9)$$

4.4. Modelo cinemático inverso

O cálculo da cinemática inversa do robô permite transformar as coordenadas Cartesianas do elemento terminal, em coordenadas articulares. Isso pode ser obtido numericamente ou analiticamente. Muitas vezes, em planejamento de trajetória robótica é necessário garantir a posição e a orientação do elemento terminal. Dessa forma, no cálculo da cinemática inversa,

além das coordenadas Cartesianas, é necessário definir a matriz de orientação para todo o trajeto.

O sistema de orientação mais utilizado é a representação em ângulos de Euler, que considera três rotações sucessivas em torno dos eixos coordenados. Neste trabalho são considerados os ângulos de Euler $w-u-w$, onde é considerada a primeira rotação φ em torno do eixo z do sistema de coordenadas inercial, uma segunda rotação θ em torno do eixo x do novo sistema de referência móvel e uma terceira rotação ψ em torno do eixo z do último sistema de referência obtido. Assim, a matriz de orientação em cossenos diretores é dada por:

$${}^A R_B = R(w, \varphi) R(u, \theta) R(w, \psi)$$

$${}^A R_B = \begin{bmatrix} c(\varphi)c(\psi) - s(\varphi)c(\theta)s(\psi) & -c(\varphi)s(\psi) - s(\varphi)c(\theta)c(\psi) & s(\varphi)s(\theta) \\ s(\varphi)c(\psi) + c(\varphi)c(\theta)s(\psi) & s(\varphi)s(\psi) - c(\varphi)c(\theta)c(\psi) & -c(\varphi)s(\theta) \\ s(\theta)s(\psi) & s(\theta)c(\psi) & c(\theta) \end{bmatrix}$$

Ou pode-se escrever:

$${}^A R_B = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \quad (4.10)$$

Igualando a Equação (4.10) com a matriz de orientação definida na Eq.(4.6) e resolvendo o sistema de equações, é possível obter as coordenadas articulares dadas por:

$$q_1 = \text{atan2}[v_x(a_6 + a_7) - x, y - v_y(a_6 + a_7)] \quad (4.11)$$

$$q_2 = \text{atan2}\left[\frac{-B \pm \sqrt{B^2 - 4AC}}{2AC}, k_2 \frac{-B \pm \sqrt{B^2 - 4AC}}{2AC} - k\right] \quad (4.12)$$

$$q_3 = \text{atan2}\left\{\left[(a_4 + a_5) \frac{Na_3 + M(a_4 + a_5)}{a_3^2 + (a_4 + a_5)^2} - M\right] / a_3, \frac{Na_3 + M(a_4 + a_5)}{a_3^2 + (a_4 + a_5)^2}\right\} \quad (4.13)$$

$$q_4 = \text{atan2}[v_x c q_1 + v_y s q_1, v_x s q_1 s(q_2 + q_3) + v_y s q_1 c(q_2 + q_3) + v_z c(q_2 + q_3)] \quad (4.14)$$

$$\begin{cases} sq_5 = [cq_1sq_4 - sq_1cq_4s(q_2 + q_3)]v_x + [sq_1sq_4 - cq_1cq_4s(q_2 + q_3)]v_y + [cq_4c(q_2 + q_3)]v_z \\ cq_5 = -sq_1c(q_3 - q_2)v_x + cq_1c(q_2 + q_3)v_y + s(q_2 + q_3)v_z \end{cases}$$

$$q_5 = \text{atan2}(sq_5, cq_5) \quad (4.15)$$

$$\begin{cases} sq_6 = [cq_1cq_4 - sq_1sq_4s(q_2 + q_3)]w_x + [sq_1cq_4 + cq_1sq_4s(q_2 + q_3)]w_y - sq_4c(q_3 - q_2)w_z \\ cq_6 = [cq_1cq_4 - sq_1sq_4s(q_2 + q_3)]u_x + [sq_1cq_4 + cq_1sq_4s(q_2 + q_3)]u_y - sq_4c(q_3 - q_2)u_z \end{cases}$$

$$q_6 = \text{atan2}(sq_6, cq_6) \quad (4.16)$$

Onde,

$$k_1 = -xsq_1 + ycq_1 - a_1 - (a_6 + a_7)[v_ycq_1 - v_xsq_1]$$

$$k_2 = z - h_1 + v_z(a_6 + a_7)$$

$$k = [(a_4 + a_5)^2 - a_2^2 + a_3^2 - (k_1^2 + k_2^2)] / -2a_2$$

$$A = k_1^2 + k_2^2; \quad B = -2kk_2; \quad C = k^2 - k_1^2$$

$$M = k_1cq_2 + k_2sq_2; \quad N = k_1sq_2 + k_2cq_2 - a_2 \quad (4.17)$$

4.5. Planejamento de trajetórias para o robô HP6

4.5.1. Planejamento de trajetórias no domínio das articulações

O planejamento no domínio das articulações pode ser realizado em quatro etapas descritas a seguir, sendo conhecido um conjunto de pontos que definem a trajetória.

1) A primeira etapa consiste no cálculo da cinemática inversa para os pontos dados. Assim, para cada ponto têm-se seis coordenadas articulares.

2) Na segunda etapa deve-se obter o vetor tempo que representa o intervalo de tempo para a movimentação de cada articulação. Na interpolação das coordenadas articulares o tempo é utilizado como abscissa para formar o par ordenado tempo e posição angular da articulação, (t_i, q_i^j) para $(i = 0, \dots, n)$ e $(j = 1, \dots, 6)$ sendo que i representa as coordenadas articulares e j as articulações. A escolha do tempo para a articulação atingir o valor calculado pelo modelo cinemático inverso influencia completamente a trajetória articular. O intervalo de tempo para o robô realizar o trajeto entre dois pontos não deve ser considerado constante porque há o risco da curva apresentar oscilações, isto é, entre dois pontos a trajetória pode ter um comprimento muito maior que o necessário. Em consequência disso, o elemento terminal se afasta do próximo ponto para depois se aproximar, como mostrado na Fig. 4.4.

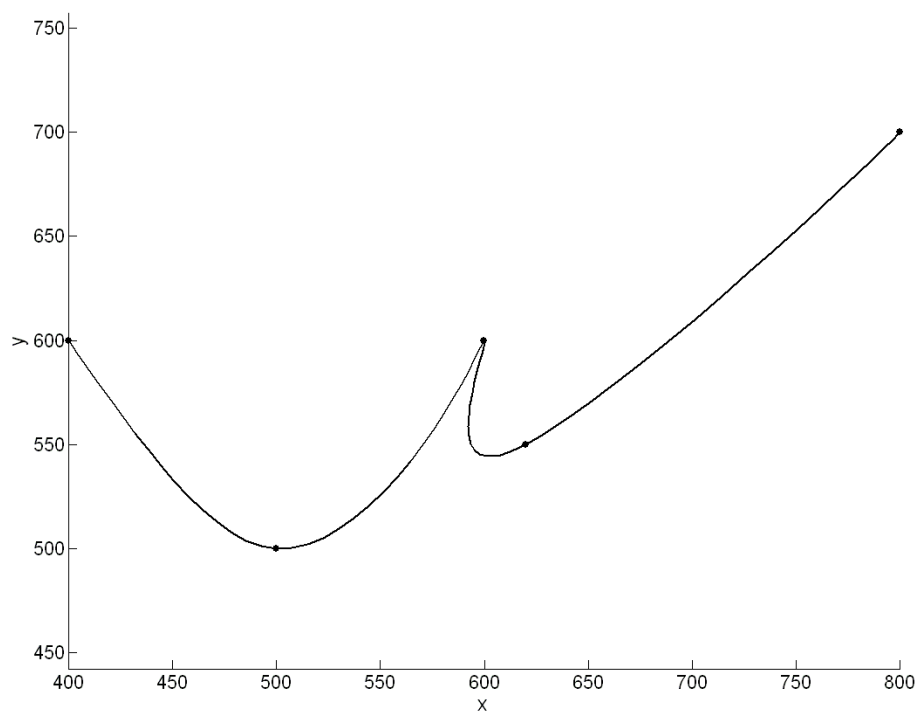


Figura 4. 4: Trajetória planejada no domínio das articulações por spline cúbica com o vetor tempo igualmente espaçado.

Para minimizar este efeito, o intervalo de tempo no trajeto entre dois pontos pode ser calculado proporcionalmente à distância entre dois pontos vizinhos, podendo escrever:

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}| \quad (4.18)$$

$$t_k = t_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} T \quad k = 1, \dots, n-1 \quad (4.19)$$

Onde: $t_0 = 0$ e T é o tempo necessário para o robô realizar todo o trajeto.

O resultado para uma trajetória planejada com o tempo calculado proporcionalmente à distância entre dois pontos vizinhos é mostrado na Fig. 4.5. Pode-se notar que a trajetória é mais suave que a trajetória da Fig. 4.4.

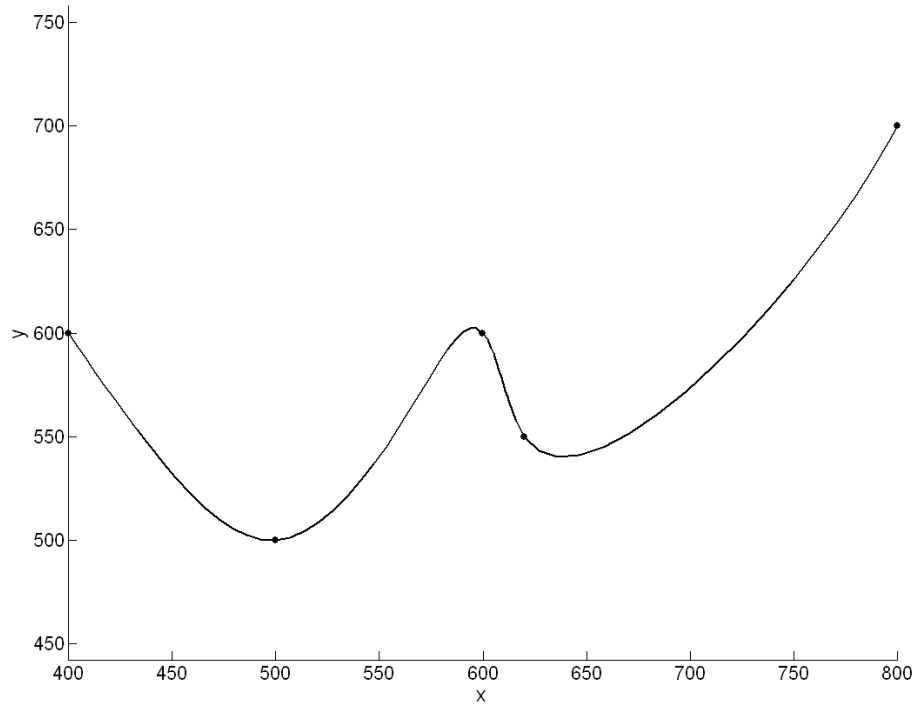


Figura 4. 5: Trajetória planejada no domínio das articulações por spline cúbica com o vetor tempo proporcional ao comprimento de corda.

3) A terceira etapa consiste na interpolação das coordenadas articulares. Neste caso, todas coordenadas referentes a uma articulação são interpoladas. As curvas obtidas são utilizadas como referência para o controle das articulações.

4) Na quarta etapa deve-se efetuar o controle dos motores para que sigam seus respectivos sinais de referência.

O método de interpolação das coordenadas articulares influencia diretamente na trajetória realizada pelo elemento terminal do robô. Nos experimentos são realizadas simulações numéricas com interpolação por spline quadrática, spline cúbica, B-spline quadrática e B-spline cúbica.

4.5.2. Simulação numérica e comparação com os resultados experimentais

Como apresentado no Capítulo III, a ordem da spline depende do grau de continuidade desejado e é necessário estabelecer condições de contorno nas extremidades da curva para completar as equações do sistema linear. No caso da spline quadrática somente uma condição é suficiente. Então, a primeira derivada é imposta nula no início da trajetória e a derivada do último ponto da trajetória não é definida. Fisicamente isso significa que as articulações partem do repouso e, no fim do trajeto como o robô deve parar, mas existe um valor de velocidade das articulações, o robô pára abruptamente.

Na spline cúbica, a condição de contorno de derivada nula é imposta no início e final do movimento. Dessa forma o robô parte do repouso e chega ao último ponto com velocidade nula.

A B-spline quadrática é uma curva mais suave que a spline quadrática, além disso, com ela é possível obter uma curva interpolante com as derivadas nos extremos nulas, diferentemente da spline quadrática que é possível definir a derivada somente em um extremo. Com a B-spline cúbica também é possível definir as derivadas nos extremos e sua continuidade é C^2 enquanto a B-spline quadrática é C^1 .

Nas simulações de planejamento de trajetória para o Motoman HP6 foram traçadas curvas que interpolam os pontos $Q_0(400, 430, 300)$, $Q_1(460, 400, 300)$, $Q_2(520, 430, 300)$, $Q_3(535, 490, 300)$, $Q_4(490, 640, 300)$, $Q_5(655, 655, 300)$, $Q_6(760, 730, 300)$ e $Q_7(850, 640, 300)$. A orientação do elemento terminal em cada ponto é constante e foi definida pela matriz de orientação em ângulos de Euler $w-u-w$:

$${}^A R_B = \begin{bmatrix} c(\varphi)c(\theta)c(\psi) - s(\varphi)s(\psi) & -c(\varphi)c(\theta)s(\psi) - s(\varphi)c(\psi) & c(\varphi)s(\theta) \\ c(\varphi)c(\theta)c(\psi) + c(\varphi)s(\psi) & -s(\varphi)c(\theta)s(\psi) + c(\varphi)c(\psi) & s(\varphi)s(\theta) \\ -s(\theta)c(\psi) & s(\theta)s(\psi) & c(\theta) \end{bmatrix}$$

Onde: $\psi = 0$ $\theta = -\frac{\pi}{2}$ e $\varphi = 0$.

Na Tabela 4.2 são apresentados os valores das coordenadas articulares referentes aos pontos definidos, utilizando-se o modelo cinemático inverso do robô.

Tabela 4.2: Coordenadas articulares referentes aos pontos da trajetória.

	$q_1(rad)$	$q_2(rad)$	$q_3(rad)$	$q_4(rad)$	$q_5(rad)$	$q_6(rad)$
Q_0	-0,7493	-0,2481	-1,0919	0	-0,2309	0,0723
Q_1	-0,8551	-0,2747	-1,0522	0	-0,2439	-0,1393
Q_2	-0,8799	-0,3504	-0,9347	0	-0,2857	-0,1889
Q_3	-0,8293	-0,4080	-0,8412	0	-0,3216	-0,0877
Q_4	-0,6534	-0,4990	-0,6882	0	-0,3836	0,2639
Q_5	-0,7854	-0,6382	-0,4451	0	-0,4875	0
Q_6	-0,8055	-0,7974	-0,1582	0	-0,6152	-0,0403
Q_7	-0,9254	-0,8110	-0,1334	0	-0,6264	-0,2800

Nas Figuras 4.6, 4.7, 4.8, 4.9, 4.10 e 4.11 são apresentados os diferentes tipos de interpolações para as seis articulações do robô. As interpolações realizadas por B-splines foram obtidas considerando a parametrização por comprimento de corda e o tempo total de percurso de 10s.

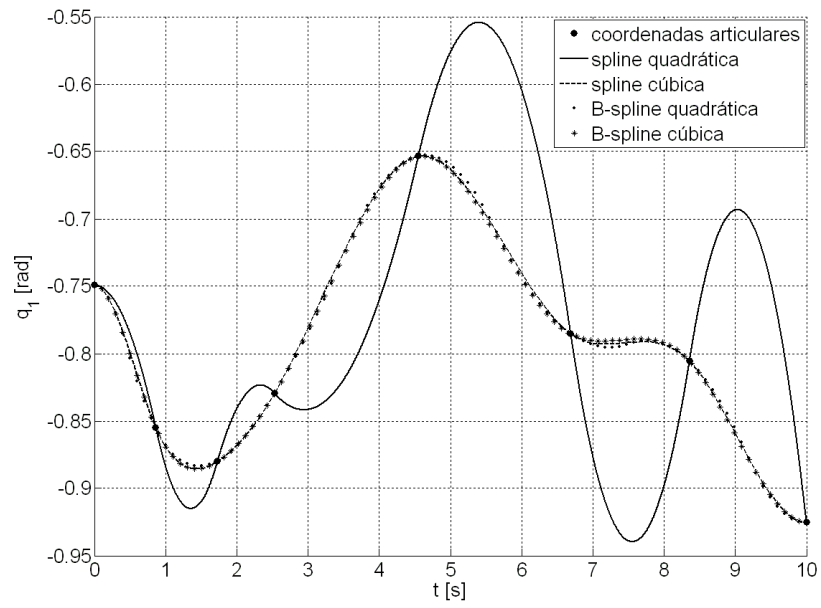


Figura 4.6: Planejamento de trajetórias para a articulação q_1 .

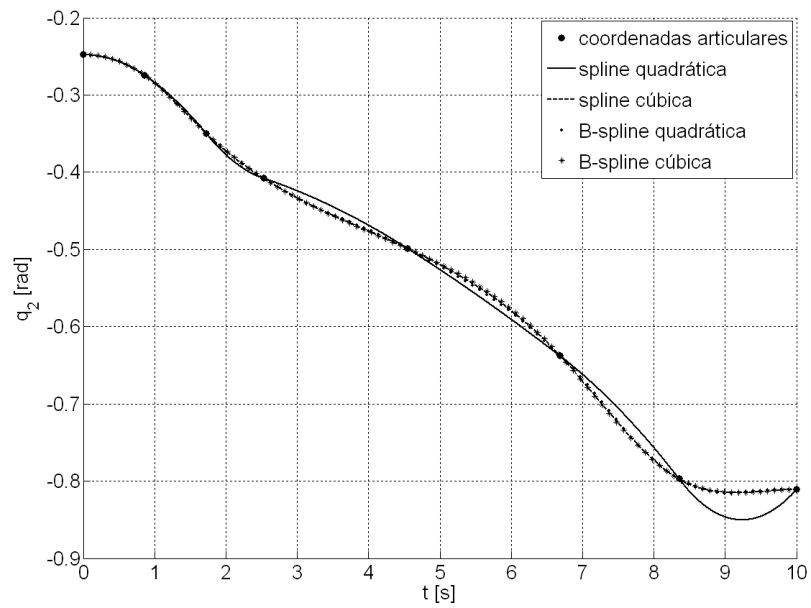


Figura 4.7: Planejamento de trajetórias para a articulação q_2 .

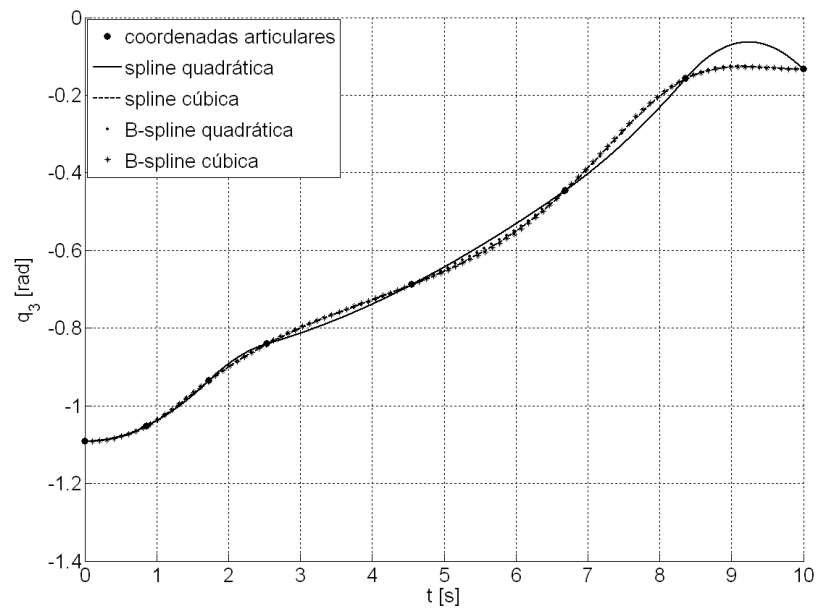


Figura 4.8: Planejamento de trajetórias para a articulação q_3 .

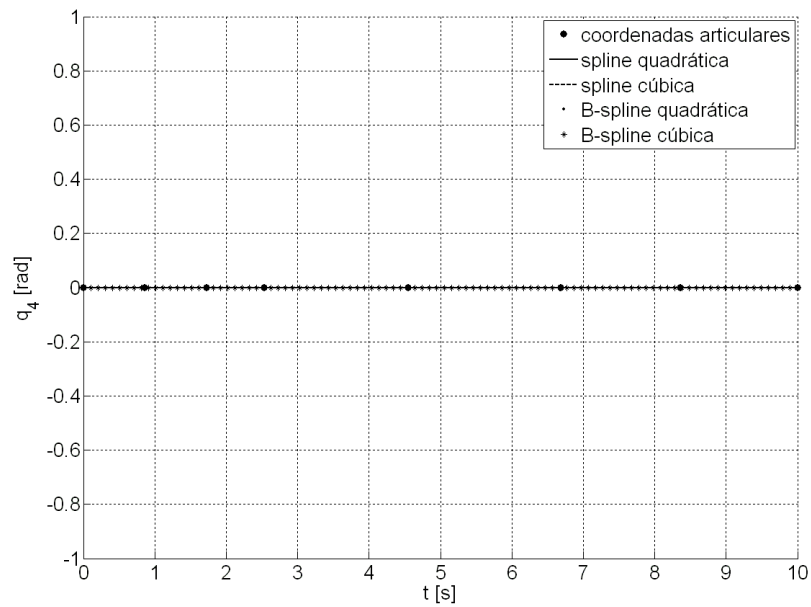


Figura 4. 9: Planejamento de trajetórias para a articulação q_4 .

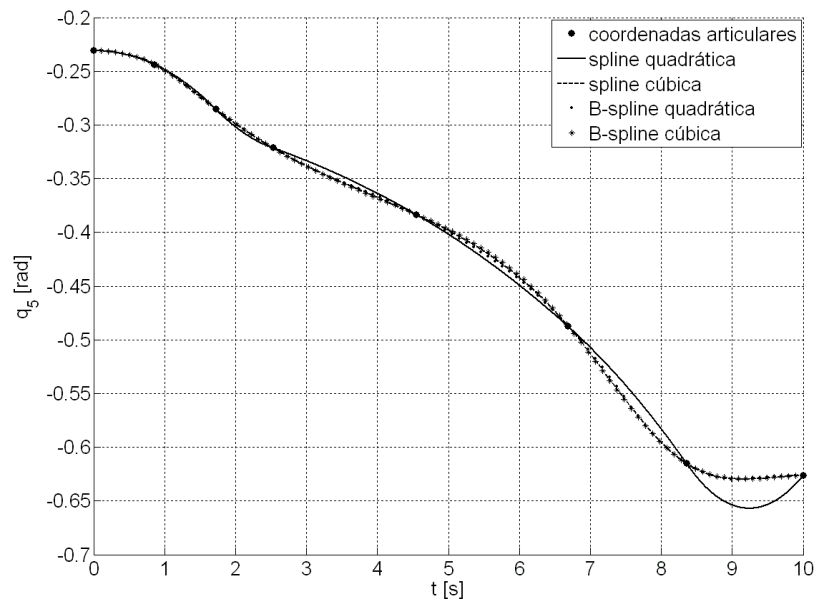


Figura 4. 10: Planejamento de trajetórias para a articulação q_5 .

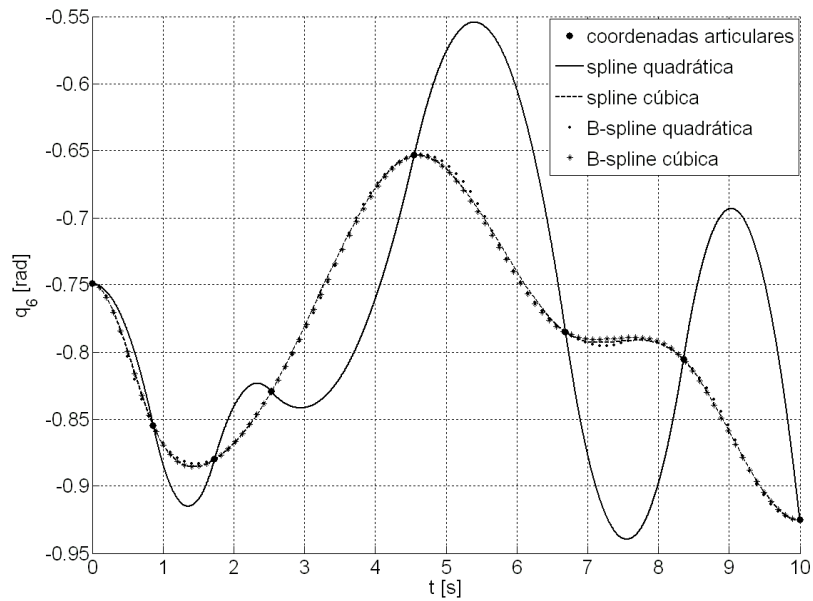


Figura 4. 11: Planejamento de trajetórias para a articulação q_6 .

As trajetórias descritas pelas articulações geram uma trajetória do elemento terminal do robô. A partir de seu modelo cinemático direto, a trajetória no domínio da tarefa é obtida para os diferentes métodos de interpolação. Estas trajetórias foram impressas em uma folha de papel para a comparação com a trajetória realizada pelo robô.

Para analisar o comportamento da trajetória descrita pelo elemento terminal do robô, permitindo sua comparação com as trajetórias simuladas, foi construído um elemento terminal que permitisse visualizar o traçado da trajetória descrita. As trajetórias simuladas numericamente foram impressas em papel e fixado na bancada de trabalho do robô, conforme a Fig. 4.12.

Em seguida, os pontos definidos foram ‘ensinados’ ao robô para o devido traçado do trajeto. Em função da diferença de escala entre os parâmetros do robô e da trajetória impressa, nas figuras traçadas utiliza-se uma unidade de medida simbólica definida por u.m (unidade métrica). Nas Figuras 4.13, 4.14, 4.15 e 4.16 são apresentadas as trajetórias obtidas.



Figura 4. 12: Montagem do experimento.

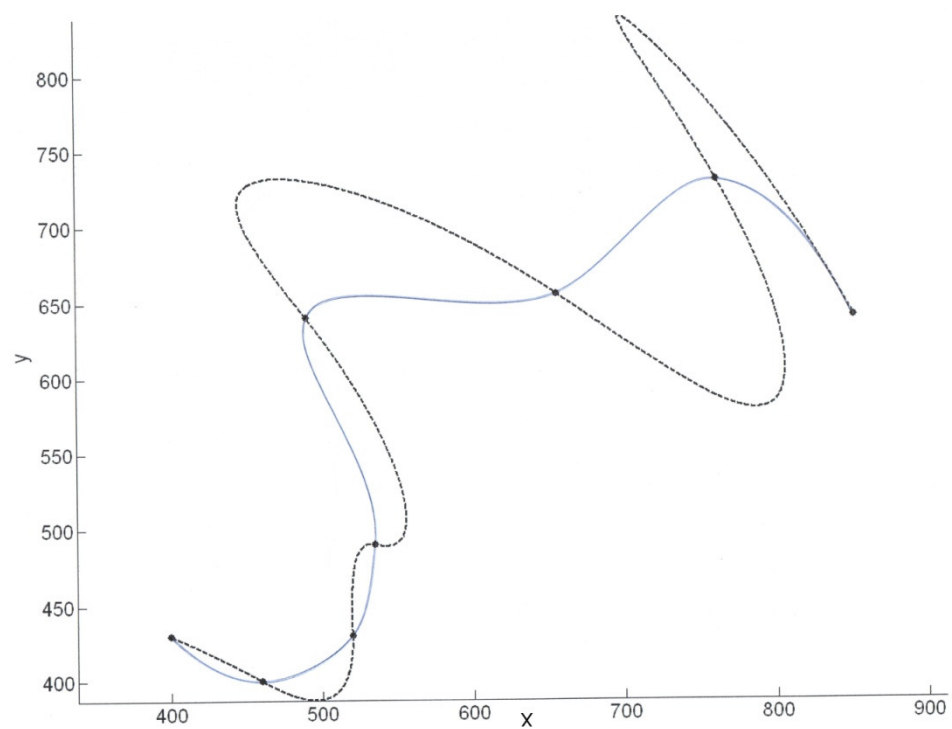


Figura 4.13: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline quadrática no domínio das articulações (linha tracejada).

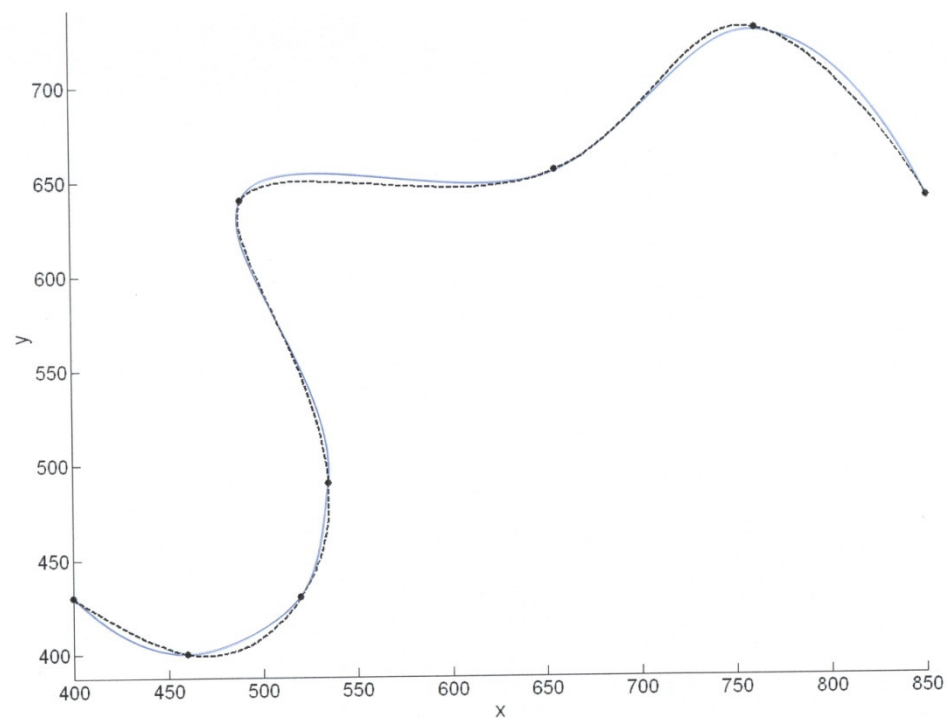


Figura 4.14: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline cúbica no domínio das articulações (linha tracejada).

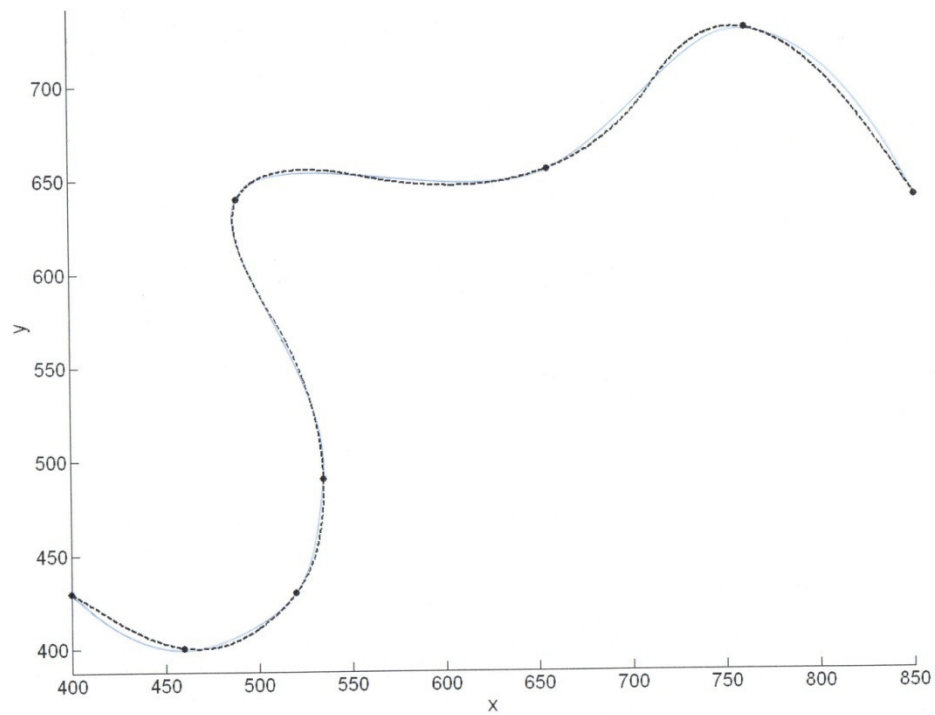


Figura 4.15: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline quadrática no domínio das articulações (linha tracejada).

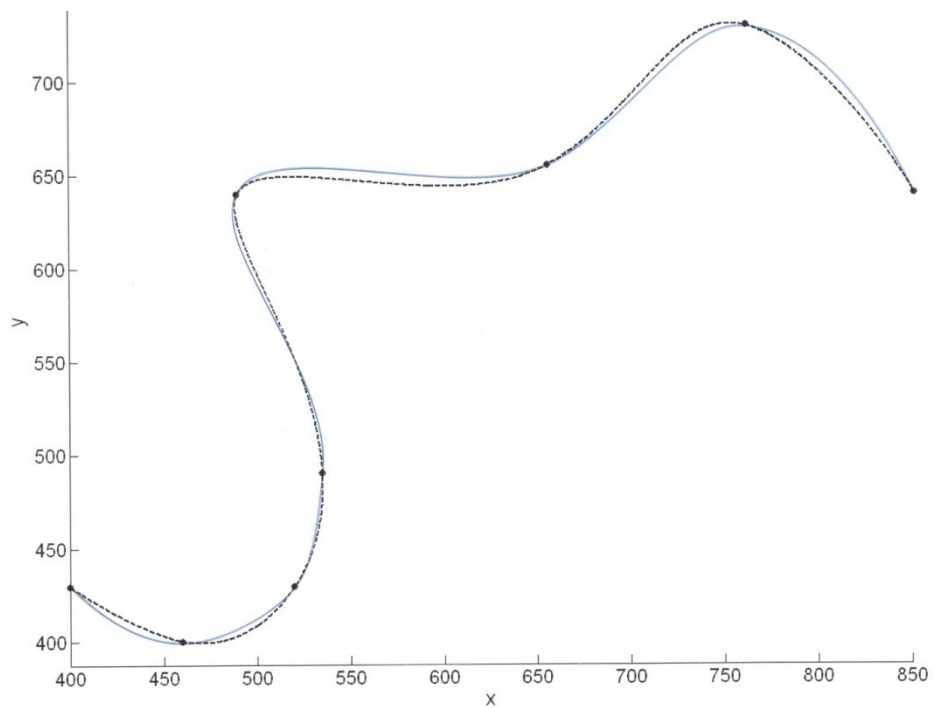


Figura 4.16: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline cúbica no domínio das articulações (linha tracejada).

Pode-se verificar pelas Figs. 4.13 a 4.16 que todas trajetórias apresentam erros, necessitando então, de novas simulações e experimentos.

4.5.3. Planejamento de trajetórias no domínio da tarefa

No planejamento de trajetória no domínio da tarefa a interpolação é realizada diretamente nos pontos conhecidos. O sinal de controle dos motores é obtido a partir da cinemática inversa para toda a curva. Na Figura 4.17 é apresentada a trajetória definida pelos mesmos pontos das trajetórias anteriores com o planejamento por spline quadrática e cúbica, B-spline quadrática e cúbica.

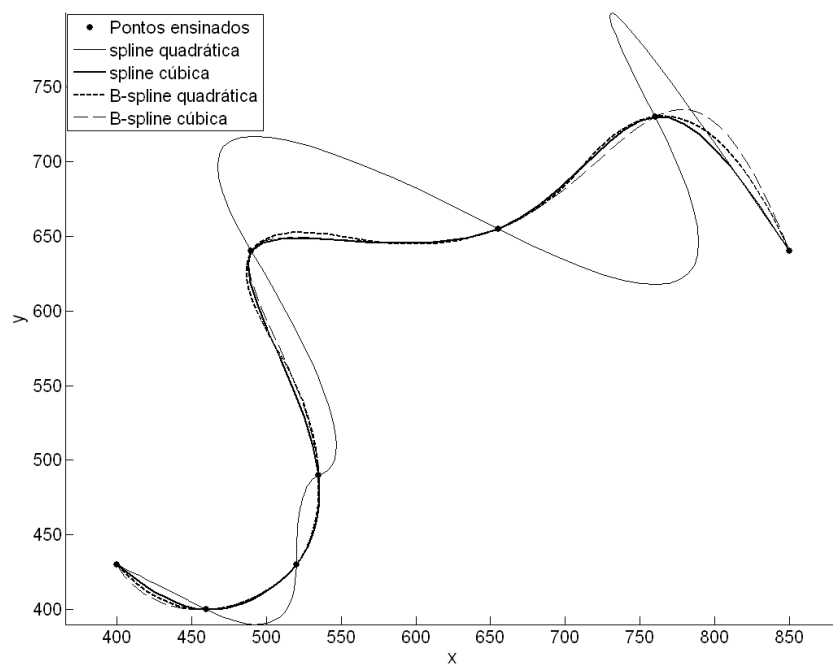


Figura 4.17: Planejamento de trajetória no domínio da tarefa com diferentes métodos de interpolação.

A comparação entre a trajetória planejada e a trajetória seguida não foi realizada utilizando spline quadrática, pois a trajetória desenvolvida pelo robô é muito diferente deste tipo de curva como mostrado nas trajetórias anteriores. Os resultados são apresentados nas Fig. 4.18, 4.19 e 4.20.

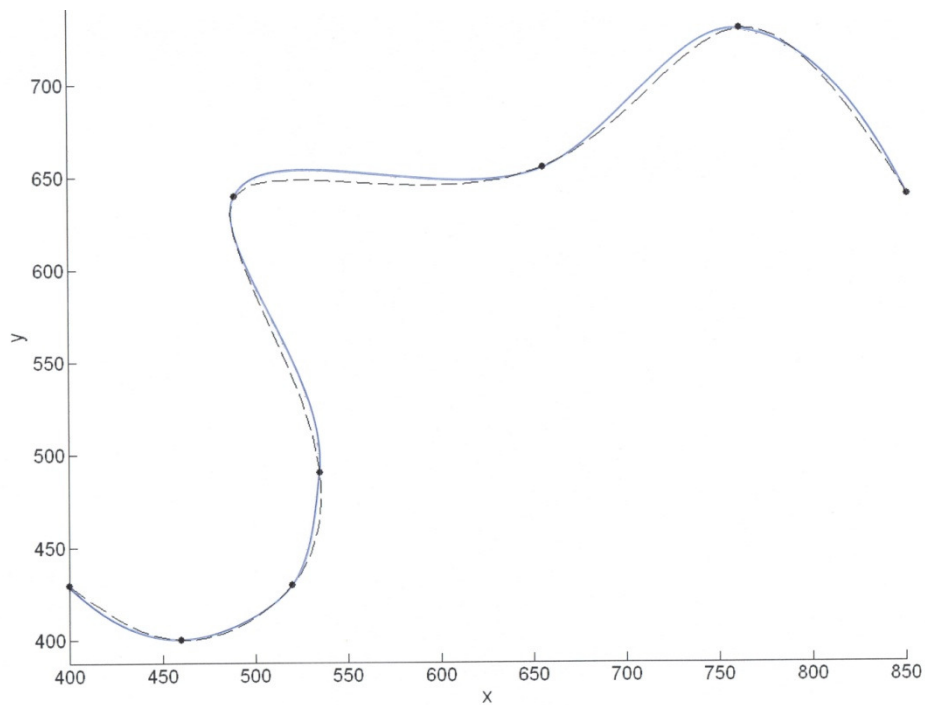


Figura 4.18: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por spline cúbica no domínio da tarefa (linha tracejada).

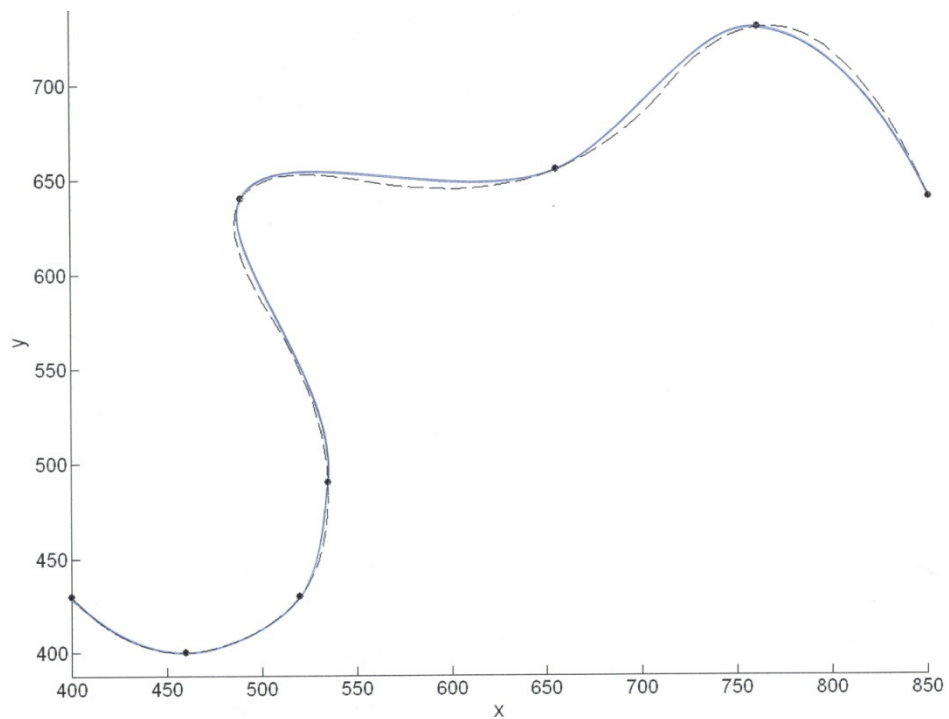


Figura 4.19: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline quadrática no domínio da tarefa (linha tracejada).

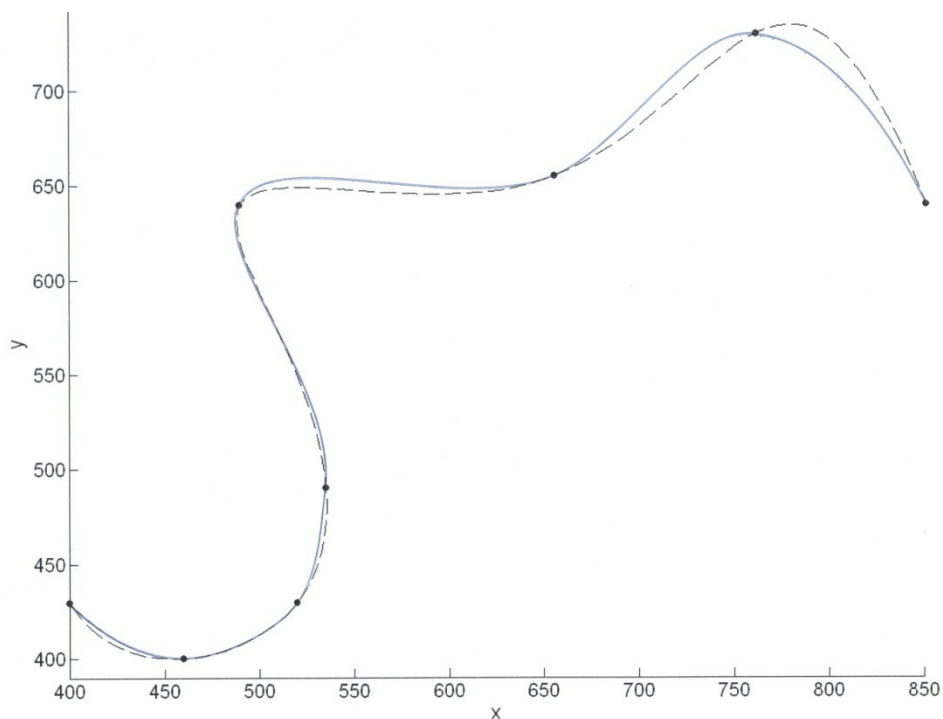


Figura 4.20: Trajetória realizada pelo robô (linha contínua) e trajetória planejada por B-spline cúbica no domínio da tarefa (linha tracejada).

As Fig. 4.18, 4.19 e 4.20 deixam claro que nenhuma dessas formas de planejamento de trajetórias corresponde à trajetória descrita pelo robô. Então, novas análises devem ser realizadas.

4.5.4. Interpolação realizada pelo Motoman HP6

Curvas splines e B-splines são tipos de interpolação polinomial por partes. Na spline, a cada dois pontos é calculado um polinômio cujo grau depende da ordem de continuidade imposta no início e no final de cada segmento. Nas curvas traçadas pelo robô podem-se observar trechos com perfil “S” entre dois pontos consecutivos: uma polinomial com esse perfil deve ser no mínimo de terceiro grau. Na B-spline é possível obter um perfil “S” com polinômios de segundo grau. Isso porque os polinômios são obtidos entre os nós, diferentemente da spline que é entre os pontos de controle. Dessa forma, entre os pontos da interpolação por B-spline pode haver um perfil “S” pois este segmento é composto por duas

parábolas. Na Figura 4.21 é apresentado um perfil “S” entre os pontos P_1 e P_2 , sendo composto por partes de duas parábolas.

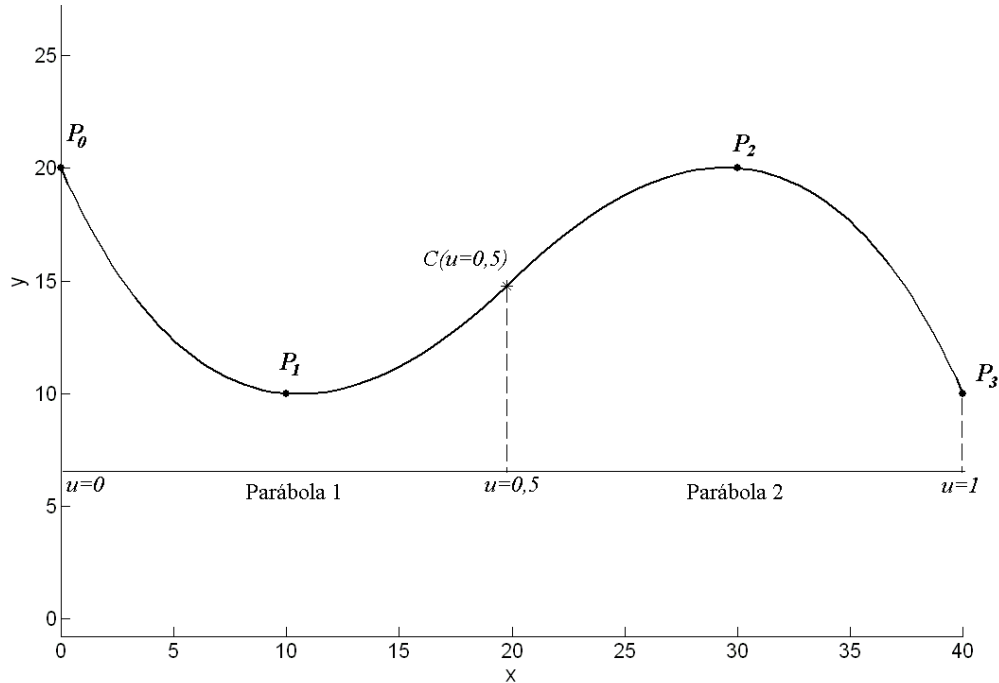


Figura 4.21: Perfil “S” obtido por uma B-spline quadrática.

De acordo com os experimentos, a curva de interpolação do robô pode apresentar perfil “S”, a qual não pertence a uma spline cúbica e nem a uma B-spline, conforme pode ser observado nas Figs. 4.13 a 4.20.

Assim, deve-se analisar a trajetória descrita sob nova perspectiva, ou seja, supondo que ela seja uma composição de curvas sem a caracterização de que seja uma spline ou uma B-spline.

Pode-se observar que o primeiro e o último segmento das Figs. 4.13 a 4.20 são similares a parábolas. Então, em uma primeira aproximação pode-se considerar que nestes segmentos a curva seja uma polinomial quadrática. Para sua definição, a primeira polinomial deve ser definida pelos três primeiros pontos e a última polinomial pelos três últimos pontos.

Para verificar se a trajetória descrita pelo robô no primeiro e último segmento é realmente uma parábola, foram realizados diversos experimentos. Na Figura 4.22 foram considerados quatro pontos. A parábola do primeiro segmento é definida pelos pontos P_0 , P_1 e P_2 e a parábola do último segmento pelos pontos P_1 , P_2 e P_3 . A trajetória descrita pelo robô parte do

ponto P_0 e vai até o ponto P_3 . Pode-se observar que tanto no primeiro segmento como no último, a trajetória descrita pelo robô coincide com as parábolas calculadas.

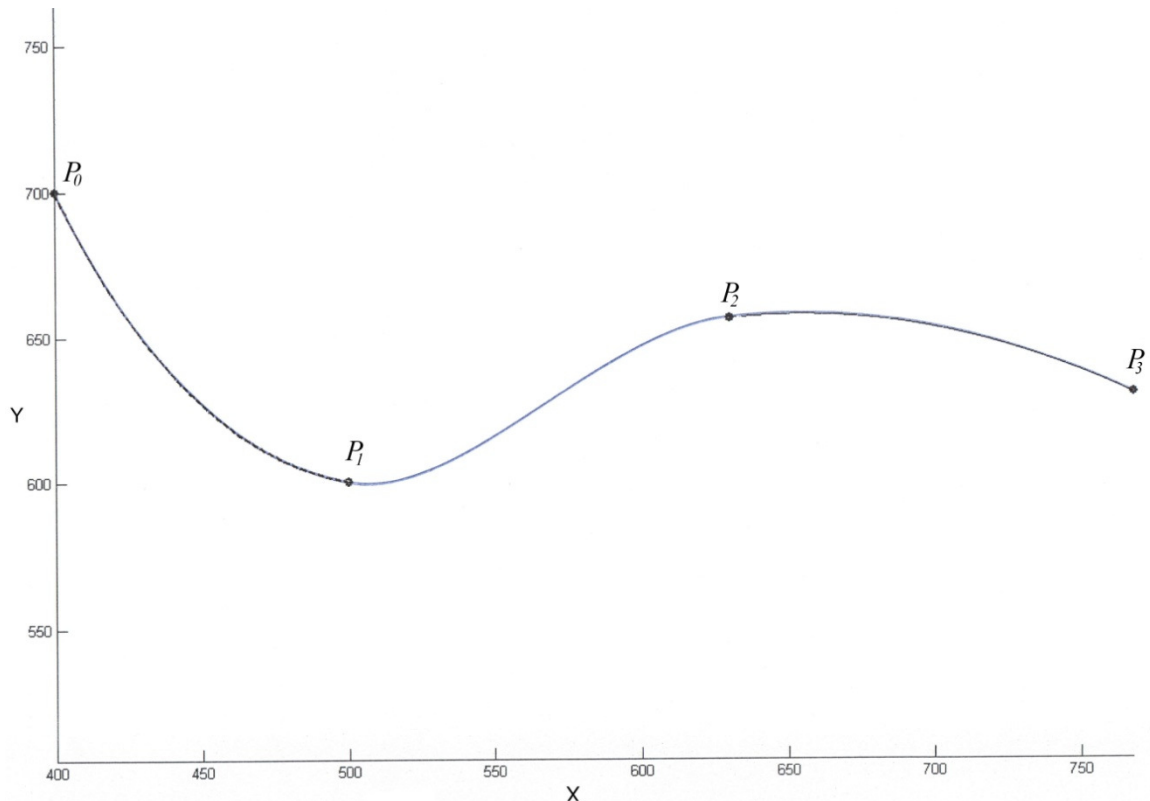


Figura 4.22: Curva com três segmentos em que o primeiro e o último são definidos por parábolas.

Um segundo exemplo, apresentado na Fig. 4.23, utiliza seis pontos para definir a trajetória. Neste caso, a parábola do segmento inicial é definida pelos pontos P_0 , P_1 e P_2 e do último segmento por P_3 , P_4 e P_5 . Novamente, pode-se verificar a coincidência das parábolas com a trajetória descrita.

Portanto, pode-se concluir que o primeiro e o último segmento da trajetória são parábolas.

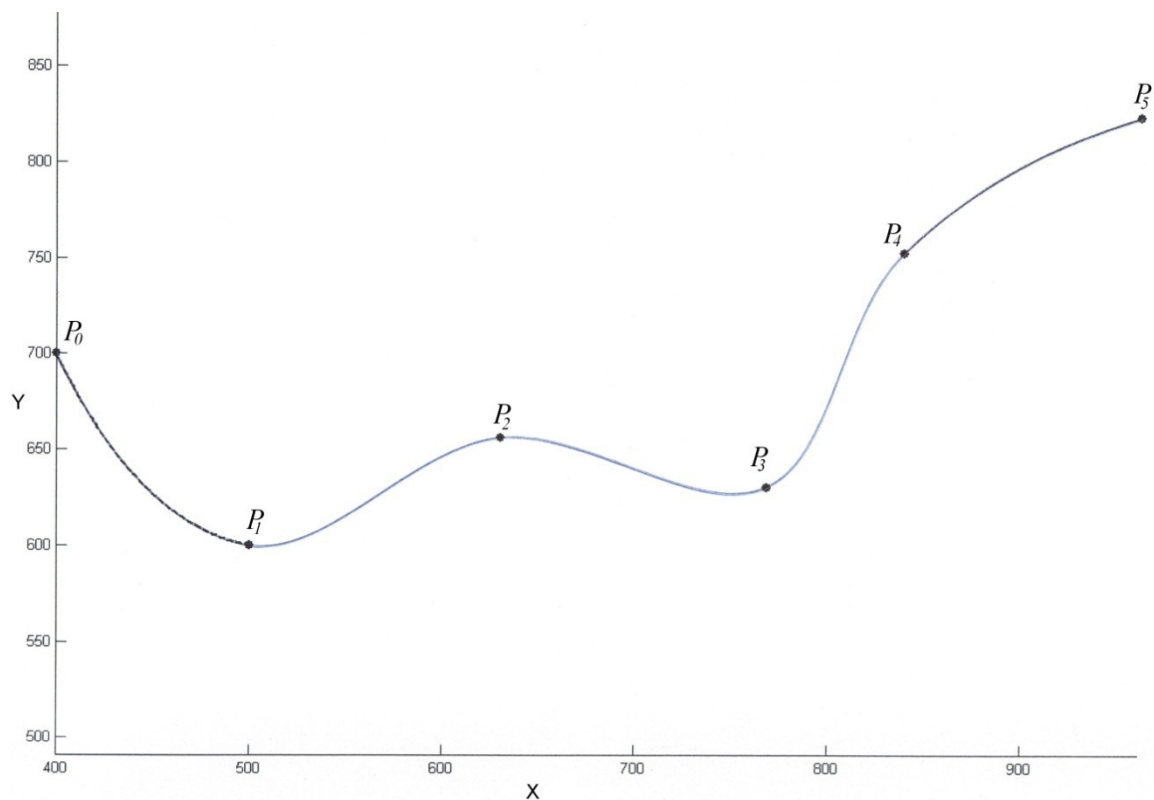


Figura 4.23: Curva com cinco segmentos em que o primeiro e o último são definidos por parábolas.

Dos exemplos apresentados pode-se verificar que as curvas correspondentes aos segmentos intermediários possuem perfil “S” que concordam nos seus extremos. Este comportamento faz crer que os segmentos podem ser definidos por Hermite cúbica ou quártica.

Considerando que na robótica industrial são utilizadas equações de baixa ordem de forma a simplificar o processamento matemático, permitindo sua implementação no controlador do robô, pode-se considerar que a curva utilizada seja uma Hermite cúbica.

Para obter a Hermite cúbica são necessárias quatro condições. A partir do primeiro segmento (trecho parabólico) e generalizando tem-se os pontos P_i e P_{i+1} , onde a Hermite cúbica será definida, e a tangente no ponto P_i . Falta então, uma condição a ser determinada. Sabe-se que a curva é contínua em P_{i+1} , o que é garantido pela derivada primeira da Hermite nos segmentos P_iP_{i+1} e $P_{i+1}P_{i+2}$. O problema a ser resolvido consiste então em obter o valor da derivada no ponto P_{i+1} , definindo a quarta condição para calcular a Hermite cúbica do segmento P_iP_{i+1} .

A partir das simulações e testes experimentais pôde-se verificar que a tangente no ponto P_{i+1} é definida pela tangente à uma parábola obtida a partir dos pontos P_i , P_{i+1} e P_{i+2} .

Assim, para obter a trajetória descrita pelo robô a partir de um conjunto de pontos conhecidos, P_i ($i = 0, \dots, n$), pode-se utilizar a seguinte lógica.

- a) Cálculo da parábola para o primeiro segmento definido por P_0 e P_1 , utilizando os três primeiros pontos P_0 , P_1 e P_2 .
- b) Cálculo da derivada da parábola no ponto P_1 .
- c) Cálculo dos coeficientes da parábola formada pelos pontos P_1 , P_2 e P_3 .
- d) Cálculo da derivada da parábola, obtida do item 'c', no ponto P_2 .
- e) A interpolação cúbica de Hermite para o segundo segmento, P_1P_2 , é realizada a partir dos pontos P_1 e P_2 e das derivadas em P_1 e P_2 obtidas nos itens 'b' e 'd', respectivamente.
- f) A lógica se repete a partir do item 'c' para obter a Hermite cúbica para todos os segmentos até o ponto P_{n-1} .
- g) O último segmento é definido pela parábola obtida pelos três últimos pontos.

No Anexo IX é apresentado o programa que executa este algoritmo.

Nas Figuras 4.24, 4.25 e 4.26 são apresentados os resultados obtidos com a trajetória simulada, conforme o algoritmo elaborado, e a trajetória traçada pelo robô.

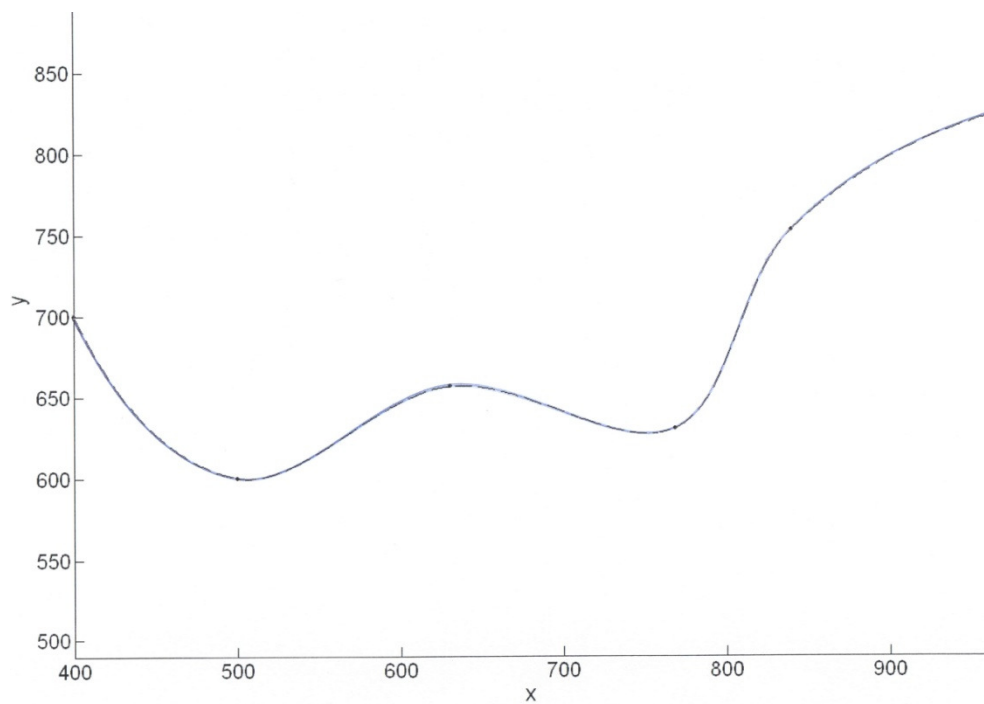


Figura 4.24: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).

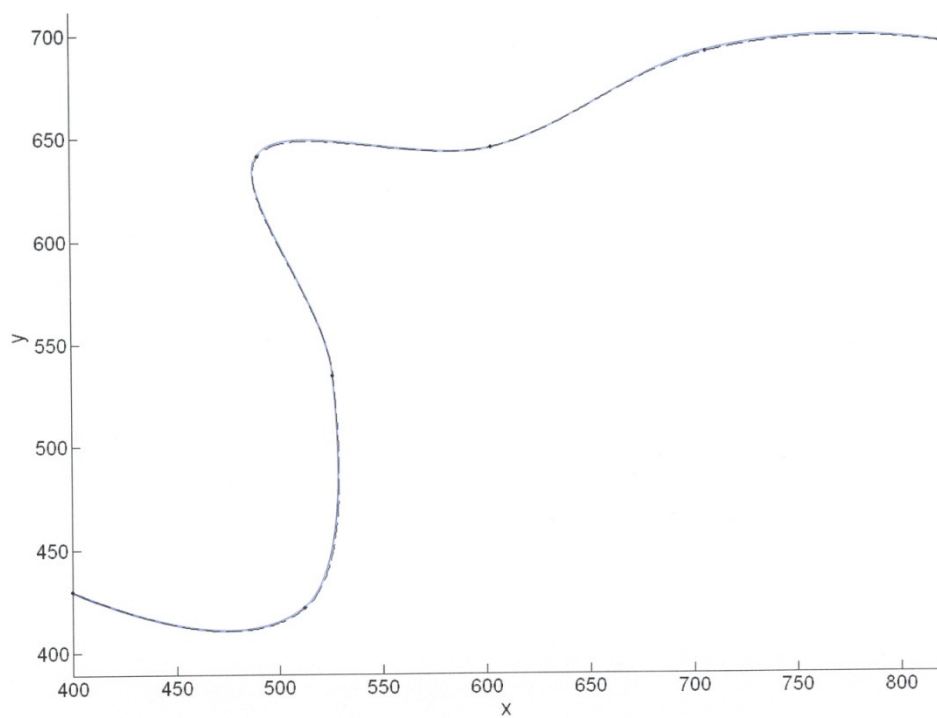


Figura 4.25: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).

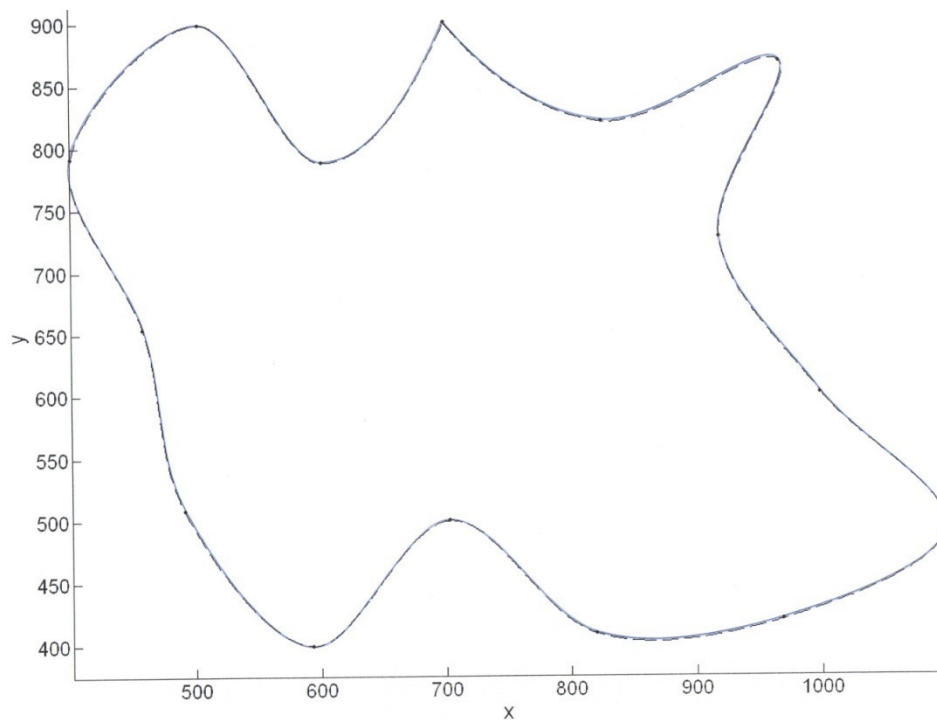


Figura 4.26: Comparação entre trajetória real (Linha contínua) e trajetória simulada (Linha tracejada).

Conforme pode ser observado nas Figuras 4.24 a 4.26 a trajetória descrita pelo robô e a simulada são coincidentes. Isto significa que a trajetória descrita pelo robô não consiste em uma spline quadrática e sim uma composição de polinomiais quadráticas e cúbicas por partes, com continuidade C^1 .

4.6. Erro de seguimento

Nos experimentos apresentados, a velocidade do robô foi ajustada para 11mm/s. Com o aumento da velocidade para 750 mm/s o robô não passa sobre a curva calculada, porém passa sobre os pontos definidos. Isso se deve ao erro de segmento ou *tracking*.

Segundo (OZAKI; et. al, 2003), robôs industriais possuem uma boa precisão no controle de posição, porém o controle de *tracking* não é tão preciso. Ou seja, o elemento terminal passa

pelos pontos ensinados com um erro muito pequeno (no HP6 pode-se garantir um erro menor que 0,08 mm), mas ao longo da trajetória entre dois pontos o erro pode ser bem maior.

De acordo com (JIANG; ISHIDA, 2007), os controladores tradicionais de robôs industriais não garante resultados precisos de *tracking* para determinadas trajetórias dinâmicas.

4.7. Conclusão

Sabendo que um dos requisitos de planejamento de trajetória é uma curva suave e previsível, a spline quadrática não é adequada para ser empregada em trajetórias robóticas. Pois a curva é composta por parábolas a cada dois pontos de controle. Dessa forma, a trajetória definida por spline quadrática possui oscilações indesejadas.

Para que a curva seja suave, dois fatores devem ser analisados: o tipo de ajuste de curva e o tipo de parametrização utilizada no tempo. Curvas spline e B-spline cúbicas possuem comportamento suave desde que o tempo seja parametrizado de forma correta como a parametrização por comprimento de corda.

A partir de simulações numéricas e testes experimentais foi demonstrado que o robô industrial Motoman HP6 não utiliza spline quadrática para planejamento de trajetórias, mas sim uma composição de curvas para a realização da interpolação com polinomiais por partes dos pontos ‘ensinados’ ao robô. A composição de parábolas no primeiro e último trajeto e Hermite cúbica nos segmentos interiores garante uma curva com continuidade C^1 em todo o trajeto.

CAPÍTULO V

PLANEJAMENTO DE TRAJETÓRIAS USANDO B-SPLINE PARA O ROBÔ MOTOMAN HP6

5.1. Introdução

Neste capítulo são apresentados os resultados do planejamento de trajetórias utilizando B-splines, aplicado ao robô industrial Motoman HP6.

A conceituação teórica das curvas B-splines usadas na interpolação foi apresentada no Capítulo III e o modelo cinemático do robô no Capítulo IV.

Para definir uma B-spline são necessários:

- Um conjunto de pontos de controle;
- O vetor de nós;
- O grau das funções de base.

Da mesma forma, para alterar uma curva B-spline pode-se mudar um ou mais desses parâmetros.

Para o planejamento de trajetórias robóticas, em geral os pontos de controle são conhecidos (ou os pontos por onde o elemento terminal deve passar); o grau das funções de base deve permitir simplicidade no modelo matemático, garantindo a suavidade de movimento. Neste caso, funções cúbicas podem ser usadas. Então, é possível analisar uma parametrização adequada do vetor nó.

Conforme simulações previamente realizadas e apresentadas no Capítulo III, uma parametrização adequada pode ser obtida pela proporcionalidade ao comprimento de cordas e o vetor nó obtido pela técnica de “*averaging*”.

Assim, do Capítulo III, sendo conhecido um conjunto de pontos Q_k ($k = 0, \dots, n$) e, para cada ponto um nó correspondente \bar{u}_k , obtido pela proporcionalidade ao comprimento de corda, ou seja:

$$\bar{u}_0 = 0$$

$$\bar{u}_n = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad k = 1, \dots, n-1 \quad (5.1)$$

e,

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}| \quad (5.2)$$

Pode-se, a partir da equação que define a B-spline e, sabendo que a curva deve passar sobre os pontos Q_k , resolver o sistema composto por $(n+1)(n+1)$ equações para obter os pontos de controle P_i ($i = 0, \dots, n$), ou seja:

$$Q_k = C(\bar{u}_k) = \sum_{i=1}^n N_{i,p}(\bar{u}_k) P_i \quad (5.3)$$

Então, o vetor nó $U = \{u_0, \dots, u_m\}$ correspondente aos pontos de controle P_i , pode ser obtido pela técnica do “*averaging*” como:

$$u_0 = \dots = u_p = 0$$

$$u_{m-p} = \dots = u_m = 1$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i \quad j = 1, \dots, n-p \quad (5.4)$$

Para a B-spline cúbica tem-se que $p = 3$. Então, se um ponto de controle é alterado, a região da curva alterada é composta por quatro segmentos da curva: dois antes do ponto de controle e dois depois.

Utilizando a parametrização do tempo proporcional ao comprimento de corda, da mesma forma que \bar{u}_k , pode-se escrever o tempo como uma função do tempo total de movimento T como:

$$t = T\bar{U} = T[\bar{u}_0, \dots, \bar{u}_m] \quad (5.5)$$

A seguir são apresentados exemplos de simulações realizadas no espaço da tarefa e no espaço das articulações para efeito de comparação.

Para isso, foram definidos os pontos por onde o elemento terminal deve passar e a correspondente orientação. Utilizando a modelagem matemática apresentada, foi obtida uma curva B-spline que o elemento terminal deve descrever. Esta curva, para efeito de comparação, é considerada a curva desejada e que nas figuras a seguir é denominada de “espaço da tarefa”.

Utilizando o modelo cinemático inverso do robô, a partir dos pontos conhecidos no espaço da tarefa, foram obtidas as coordenadas articulares correspondentes, sendo então utilizadas para planejar a trajetória no espaço das articulações, também utilizando B-spline. A trajetória do elemento terminal é obtida pelo modelo cinemático direto.

A análise foi feita para tanto a posição como para a orientação.

5.2. Exemplo 1 – Trajetória no plano

Neste exemplo os pontos por onde o elemento terminal deve passar são: Q_0 (400, 800, 200), Q_1 (480, 750, 200), Q_2 (560, 700, 200), Q_3 (480, 600, 200), Q_4 (600, 550, 200), Q_5 (720, 600, 200), Q_6 (640, 700, 200), Q_7 (720, 800, 200) e Q_8 (800, 800, 200) e a orientação do elemento terminal foi considerada constante e definida pelos ângulos de Euler como: $\varphi = -0,6109 \text{ rad}$, $\theta = -0,3840 \text{ rad}$ e $\psi = 0,2618 \text{ rad}$. Com esses ângulos, o vetor da direção do elemento terminal é dado por $\vec{v} (0,3017 \quad 0,8821 \quad -0,3618)$.

A Figura 5.1 apresenta o resultado das trajetórias obtidas.

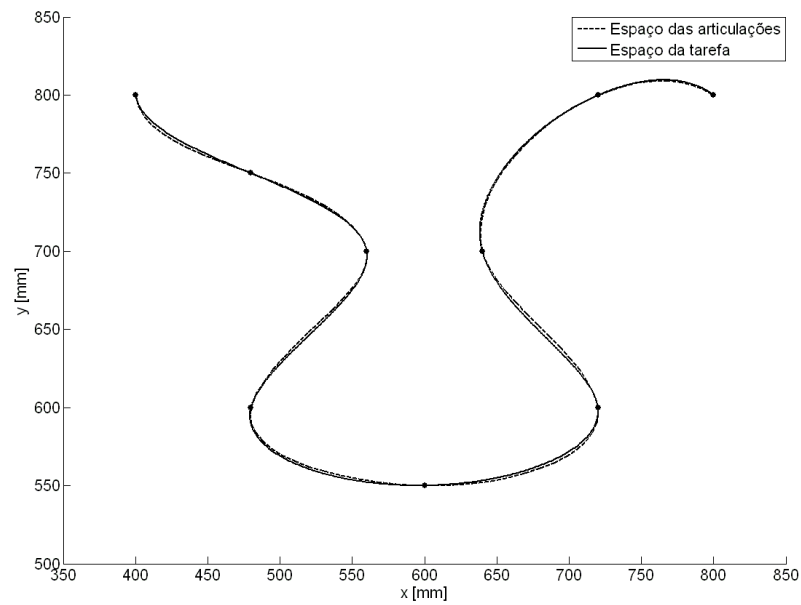


Figura 5.1: Simulação trajetórias no espaço das articulações e da tarefa.

O erro (desvio) entre as duas trajetórias é apresentado na Fig. 5.2. Pode-se verificar que a curva passa sobre os pontos que definem a trajetória.

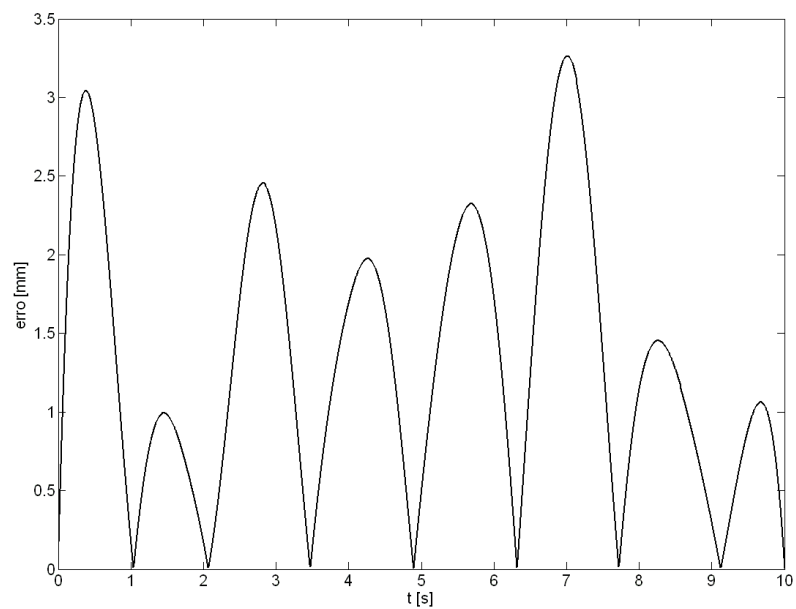


Figura 5.2: Desvio entre a trajetória no espaço das articulações e da tarefa.

Na Figura 5.3 é apresentado o comportamento do erro da orientação do elemento terminal que é obtido pelo produto escalar entre o vetor unitário da orientação desejada (\vec{v}) pelo vetor

unitário da orientação que o elemento terminal se encontra. Assim o resultado do produto escalar é o cosseno diretor entre os dois vetores.

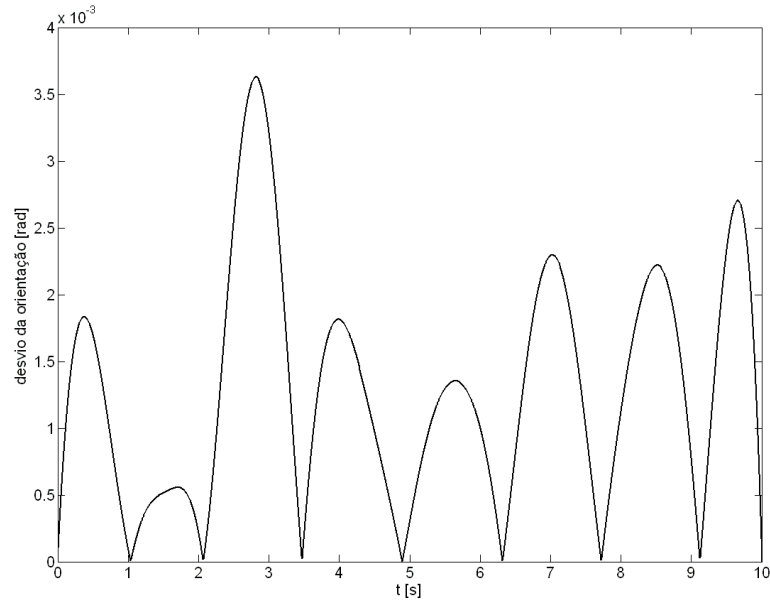


Figura 5.3: Erro de orientação do elemento terminal.

Trajetórias planejadas no espaço das articulações são imprevisíveis, isto é, não é possível prever o caminho que o elemento terminal descreverá entre dois pontos da trajetória. Porém, com a B-spline é possível prever a região em que o elemento terminal passará. Isso é possível devido à propriedade do fecho convexo, pois as referências das articulações ficarão restritas aos polígonos de controle.

5.3. Exemplo 2 – Trajetória espacial

As coordenadas dos pontos no espaço da tarefa são definidas por: $Q_0(400, 430, 252)$, $Q_1(460, 400, 252)$, $Q_2(520, 430, 276)$, $Q_3(535, 490, 303)$, $Q_4(490, 640, 354)$, $Q_5(655, 655, 393)$, $Q_6(760, 730, 444)$ e $Q_7(850, 640, 426)$; e a orientação do elemento terminal em termos dos ângulos de Euler definida por $\varphi = 0,6981 \text{ rad}$, $\theta = 0,5236 \text{ rad}$ e $\psi = 0,2618 \text{ rad}$, cujo vetor da direção do elemento terminal $\vec{v} (-0,7360 \quad 0,4744 \quad 0,4830)$, foi considerado constante ao longo do trajeto.

A Figura 5.4 apresenta as trajetórias espaciais e as Figs. 5.5, 5.6, e 5.7 as suas projeções nos planos XY, XZ e YZ, respectivamente.

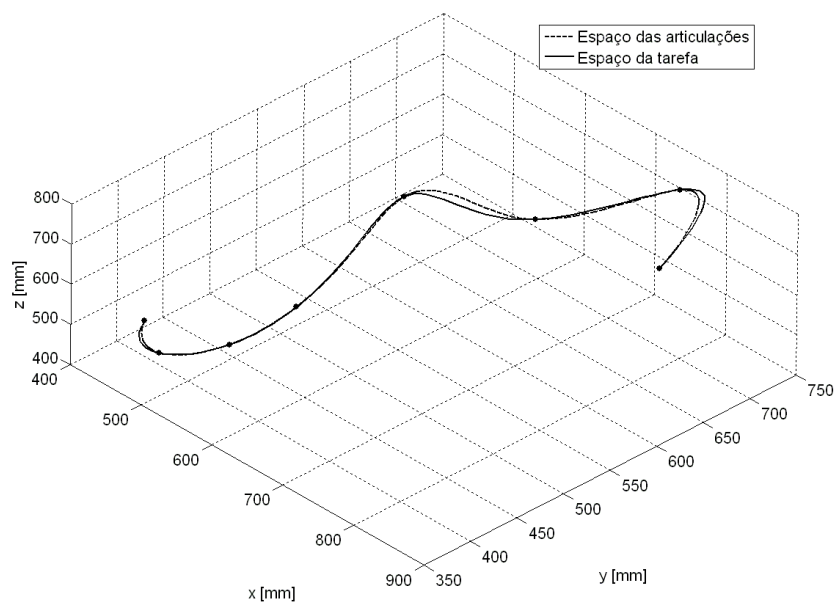


Figura 5.4: Trajetória tridimensional.

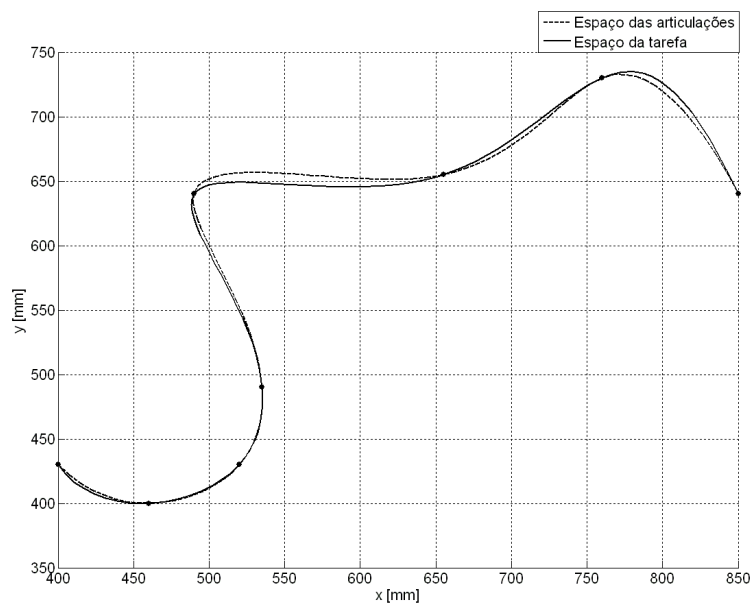


Figura 5.5: Projeção da trajetória tridimensional no plano XY.

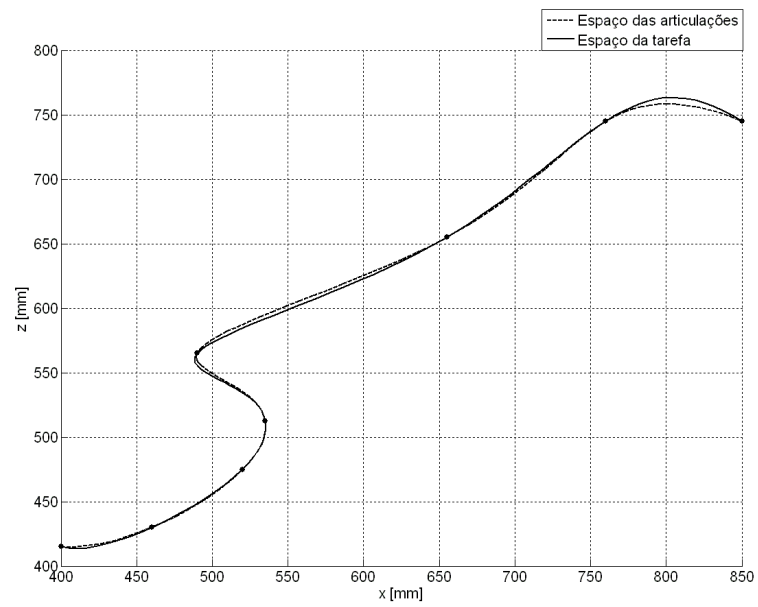


Figura 5.6: Projeção da trajetória tridimensional no plano XZ.

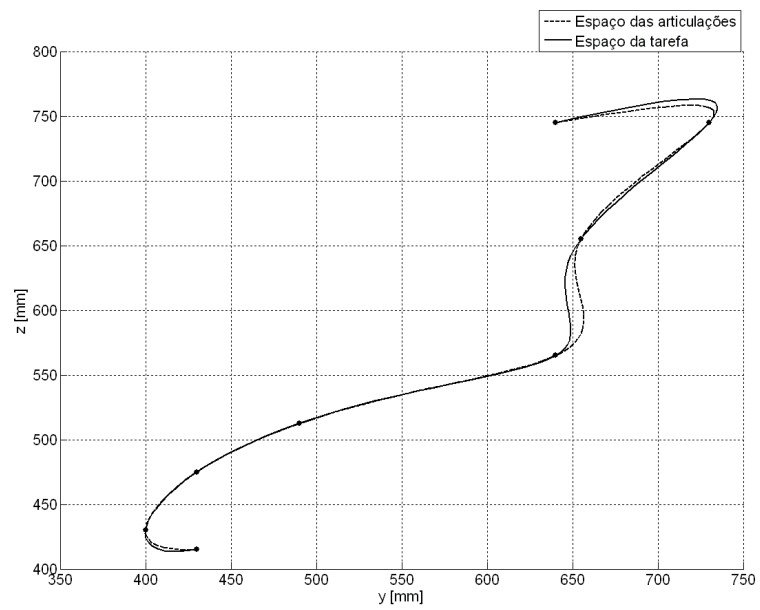


Figura 5.7: Projeção da trajetória tridimensional no plano YZ.

O erro da entre as trajetórias é apresentado na Fig. 5.8.

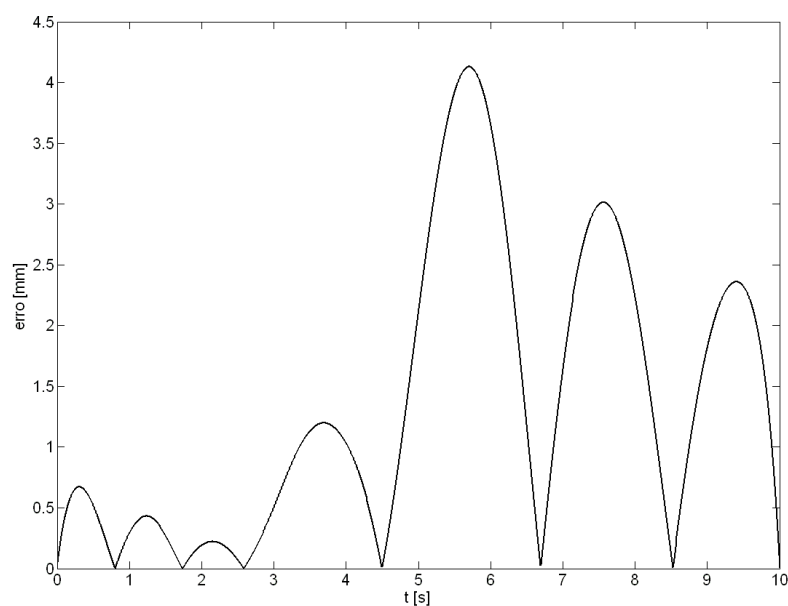


Figura 5.8: Erro absoluto entre a trajetória no espaço da tarefa e a no espaço das articulações.

A representação espacial do comportamento espacial da orientação é um problema não completamente resolvido. Com o intuito de verificar seu comportamento, além da representação do comportamento do erro, apresentado na Fig. 5.9, foi elaborada a Figura 5.10 onde a orientação é representada por vetores de mesmo comprimento.

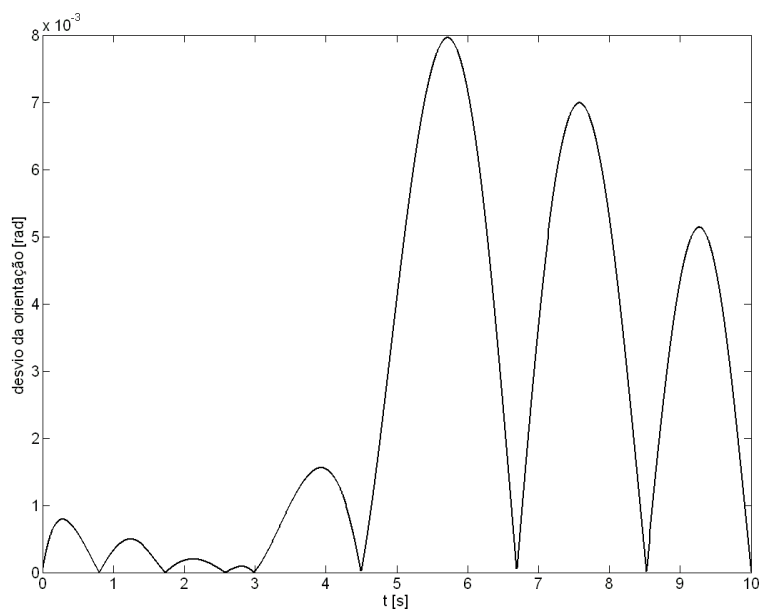


Figura 5.9: Desvio entre a orientação do elemento terminal e a orientação desejada.

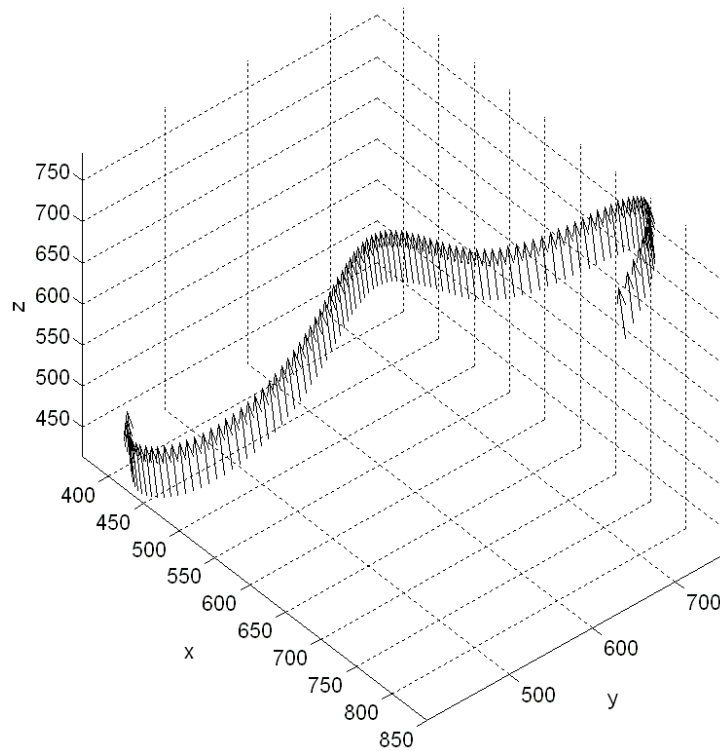


Figura 5.10: Orientação do elemento terminal ao longo da trajetória.

5.4. Exemplo 3 – Inclusão de pontos para a redução do erro

Este exemplo mostra claramente que a inclusão de pontos intermediários reduz consideravelmente o erro entre as trajetórias.

Para este exemplo foi utilizado os mesmos pontos do exemplo 2, com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$, cujas trajetórias espaciais estão apresentadas na Fig. 5.11 e as Figs. 5.12, 5.13, e 5.14 apresentam suas projeções nos planos XY, XZ e YZ, respectivamente.

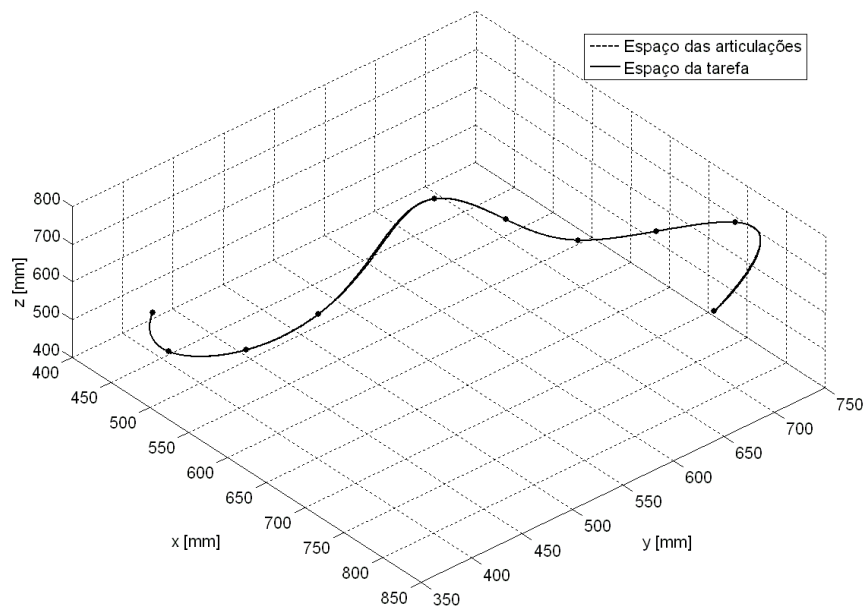


Figura 5.11: Trajetória tridimensional do exemplo 2 (Fig. 5.4) com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$.

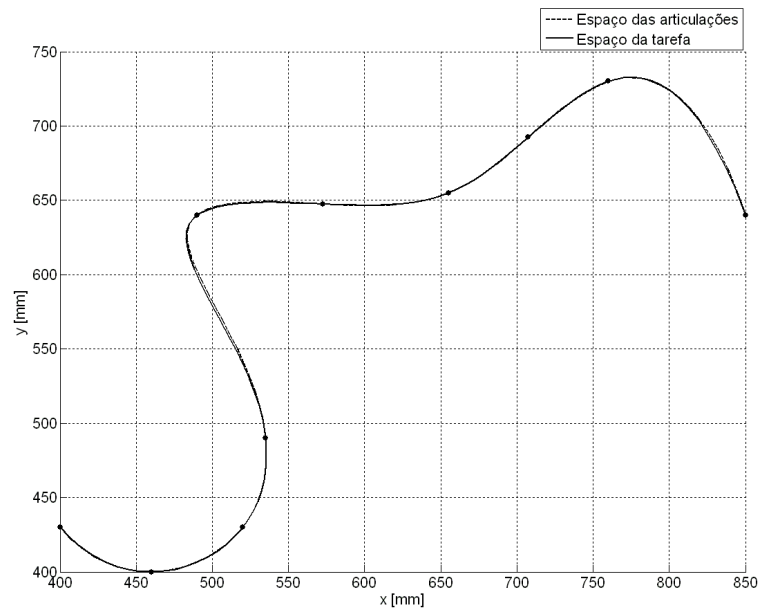


Figura 5.12: Projeção da trajetória tridimensional no plano XY com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$.

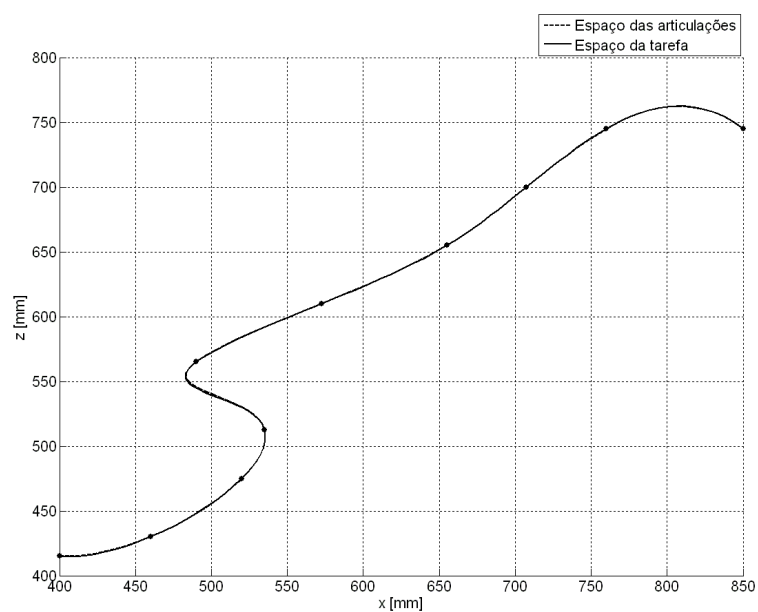


Figura 5.13: Projeção da trajetória tridimensional no plano XZ com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$.

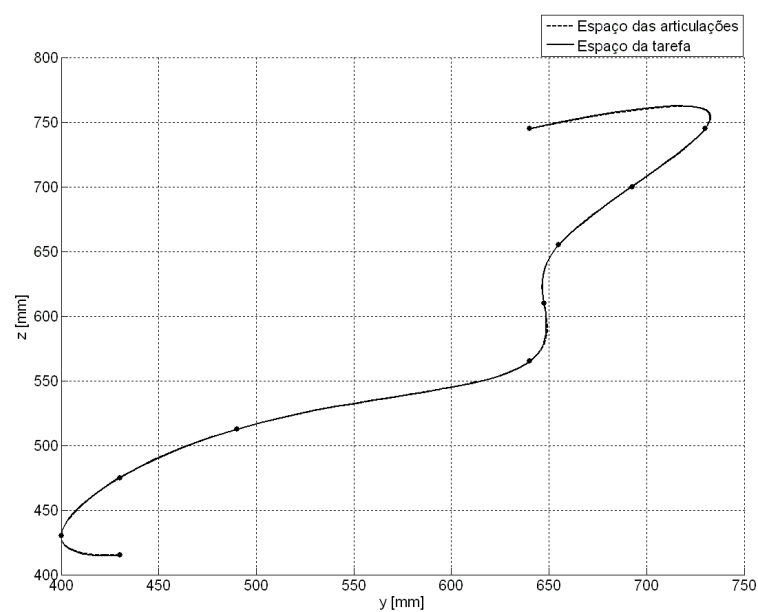


Figura 5.14: Projeção da trajetória tridimensional no plano YZ com a inclusão dos pontos $Q(572,5 \ 647,5 \ 610)$ e $Q(707,5 \ 692,5 \ 700)$.

Nas Figs. 5.15 e 5.16 estão representados os erros de posição e os erros de orientação ao longo da trajetória.

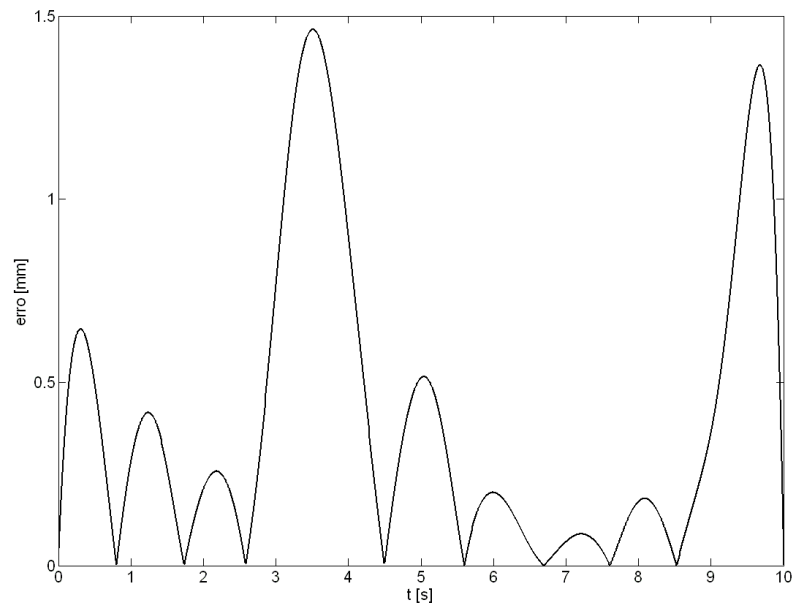


Figura 5.15: Erro absoluto entre a trajetória no espaço da tarefa e a no espaço das articulações.

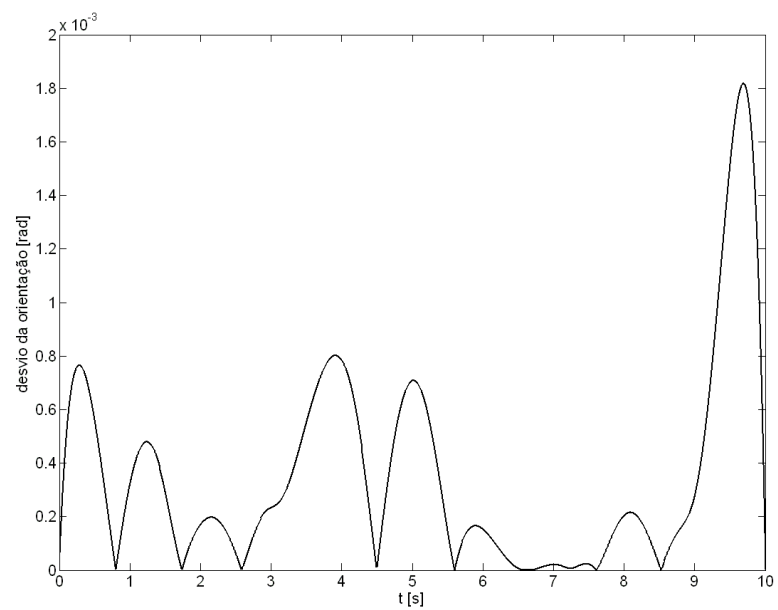


Figura 5.16: Desvio entre a orientação do elemento terminal e a orientação desejada.

Pode-se verificar uma redução importante dos erros quando comparados àqueles das Figs. 5.8 e 5.9.

5.5. Conclusão

Na prática, para que o robô siga uma determinada trajetória, os pontos que definem um caminho devem ser ‘ensinados’ manualmente. Assim, quanto mais próximo a trajetória deve se aproximar de um perfil, mais pontos devem ser ensinados. Na literatura é comum encontrar que a trajetória é planejada no domínio da tarefa quando se deseja previsibilidade no movimento do robô entre dois pontos ‘ensinados’, e se a trajetória for ensinada no domínio das articulações, o robô atingirá todos os pontos ensinados, porém o movimento entre dois pontos não pode ser previsto, podendo provocar colisões com o robô e seu ambiente de trabalho.

No entanto, as simulações realizadas e apresentadas neste capítulo não condizem com o dito sobre o planejamento da trajetória no espaço das articulações quando se utiliza as curvas B-spline.

Esse comportamento previsível quando se utiliza B-spline é garantido pela propriedade do fecho convexo, ou seja, a curva obtida está sempre contida dentro do polígono de controle. Dessa forma, com a parametrização adequada do tempo, o planejamento de trajetórias no domínio das articulações não terá oscilações entre dois pontos de controle. Além disso, a orientação do elemento terminal terá um erro relativamente pequeno para muitas aplicações do robô, como manipulação de objetos líquidos, colagem e pintura.

Apesar das simulações apresentarem trajetórias no espaço das articulações próximas às no espaço da tarefa, nesse trabalho o que se pode concluir é que devido à propriedade do fecho convexo a trajetória é previsível.

Estão previstos testes com o robô industrial citado para comprovar os resultados das simulações.

CAPÍTULO VI

CONCLUSÕES

6.1. Conclusões

O estudo de planejamento de trajetórias iniciou com o projeto de cames mecânicos, pois a cada perfil de movimento que o seguidor deve descrever, o perfil do came deve ser reprojetoado respeitando as condições de suavidade de movimento. Devido à dificuldade de construir um novo came para uma nova trajetória, surgiu o came eletrônico, que é um servosistema que pode seguir as trajetórias programadas e armazenadas em sua memória.

A partir dos cames eletrônicos, foi possível o surgimento dos robôs, os quais possuem vários eixos programáveis, possibilitando o seguimento de trajetórias mais complexas e conseqüentemente a realização de tarefas complexas.

Para cada tarefa que o robô deve realizar, sua trajetória deve ser definida. O tipo de curva utilizada influencia diretamente no comportamento da trajetória. Com o estudo de diversas curvas aplicadas à planejamento de trajetórias, foi possível verificar que o robô Motoman HP6, no planejamento de curvas no domínio da tarefa, utiliza interpolação por Hermite por partes associada com parábolas. Esta metodologia garante um perfil de trajetória previsível.

Devido à propriedade do fecho convexo, a B-spline apresentou ser uma curva aplicável em trajetórias planejadas no domínio das articulações. Sabendo que a curva de referência de cada articulação calculada por B-spline estará sempre no interior do polígono de controle, movimentos imprevisíveis no elemento terminal não ocorrerão. Além disso, definindo que a orientação do elemento terminal deve ser constante, o erro de orientação do elemento terminal é praticamente insignificante para muitas aplicações industriais.

6.2. Sugestões para trabalhos futuros

Para a validação dos resultados do planejamento de trajetórias no espaço das articulações apresentados nesta dissertação, é necessário realizar o planejamento de trajetórias em um robô industrial. O fabricante do Motoman HP6 não fornece os dados para a aplicação de trajetórias diferentes das definidas como padrão do controlador. Assim, para trabalhos futuros são sugeridos:

- a) Pesquisa do protocolo de comunicação dos sinais de comando e controle do controlador HP6 e seu funcionamento;
- b) Aplicação das B-splines no planejamento de trajetórias no espaço da tarefa e no espaço das articulações no robô Motoman HP6;
- c) Verificação do efeito da continuidade do sinal de referência das articulações na movimentação do robô.

REFERÊNCIAS BOBLIOGRÁFICAS

ALEOTTI, J.; CASELLI, S.; MACCHEROZZI, G.; Trajectory Reconstruction with NURBS Curves for Robot Programming by Demonstration. International Symposium on Computational Intelligence in Robotics and Automation, Finlândia, jun. 2005.

ANGELES, J., 1997, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Springer-Verlag, New York.

BAZAZ, S. A.; TONDU, B.; Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators. **Robotics and Autonomous Systems**. v. 29, n. 5, p. 257-268, mai, 1999.

BIAGIOTTI, L.; MELCHIORRI, C. **Trajectory Planning for Automatic Machines and Robots**. Springer-Verlag Berlin Heidelberg, 2008.

BOBROW, J. E., DUBOWSKY, S., GIBSON, J. S., 1985, "Time-Optimal Control of Robotic Manipulators Along Specified Paths", Int. J. of Robotics Research, Vol. 4, no 3, pp: 3-17.

BORIGA, M.; GRABOS, A. Planning of Manipulator Motion Trajectory with Higher-degree Polynomials use. **Journal of Mechanism and Machine Theory**. V. 44, n. 4, p. 1400-1419, dez. 2008.

CAIGNY, J.; DEMEULENAERE, B.; SWEVERS, J.; SCHUTTER, J.; Optimal Design of Spline-Based Feedforward for Trajectory Tracking. AMERICAN CONTROL CONFERENCE, New York City, USA, Jul. 2007.

CAN, A.; ÜNÜVAR, A.; Five-axis tool path generation for 3D curves created by projection on B-spline surfaces. **The International Journal of Advanced Manufacturing Technology**, v. 42, n. 9, p. 1047-1057, September 2009.

CAO, B.; DODDS, G. I.; IRWIN, G. W.; Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. **Control Theory & Applications**. v.144, n. 5, p. 467 – 475, sep. 1997.

CASTAIN, R. H., PAUL, R. P., 1984, “An On-Line Dynamic Trajectory Generator”, The Int. J. of Robotics Research, Vol. 3, no 1, pp: 68-72.

CECCARELLI, M., 2004, **Fundamentals of Mechanics of Robotic Manipulation**, Kluwer Academic Publishers, The Netherlands.

CHAUDHRY, T.; GULREZ, T.; ZIA, A.; Bézier Curve based Dynamic Obstacle Avoidance and Trajectory Learning for Autonomous Mobile Robots. **Intelligent Systems Design and Applications (ISDA), 10th International Conference on**. P. 1059-1065. Cairo, 2011.

DE BOOR, C. **A Practical Guide to Spline**. 1. ed. N.Y: Springer, 2000. 241 p.

ELGELBERGER, J. F. **Robotics in Practice**, AMACOM, New York, 1980.

EMAMI, M. M.; AREZOO, B.; A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. V. 42, n. 7, ago. 2010. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S001044851000062X> > . Acessado em: 10 nov. 2011.

EMAMI, M., M.; AREZOO, B.; A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. **Computer-Aided Design**, v. 42, n. 7, July 2010.

EREN, H.; FUNG, C. C.; EVANS, J.; Implementation of the spline method for mobile robot path control . **Instrumentation and Measurement Technology Conference**, v. 2, p. 739 – 744, Venice , Italy, 1999.

FANG, Y. C., HSIEH, C. C., KIM, M. J., CHANG, J. J., WOO, T. C., 1998, “Real Time Motion Fairing With Unit Quaternions”, *Computer-Aided Design*, Vol. 30, no 3, pp: 191-198.

FANG, Y. C., HSIEH, C. C., KIM, M. J., CHANG, J. J., WOO, T. C., Real Time Motion Fairing With Unit Quaternions, *Computer-Aided Design*, Vol. 30, no 3, pp: 191-198 , 1998

FISHER, W. D., MUJTABA, M. S. Minimum Ratio-Locked Profile Times for Robot Trajectories, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp: 1054-1060. , 1988.

FLORES, M. E.; MILAM, M. B.; Trajectory Generation for Differentially Flat Systems via NURBS Basis functions with Obstacle Avoidance. In: *AMERICAN CONTROL CONFERENCE*, 1996, Minneapolis, USA. p. 5769-5775.

G. C. DEVOL, *Programmed Article Transfer*. U.S.2988237, 10 December, 1954, 13 June, 1961.

HORSCH, T.; JÜTTLER, B.; Cartesian spline interpolation for industrial robots. **Computer-Aided Design**, v. 30, n. 3, p. 217 – 224, mar. 1998.

IONESCU, T. G. Terminology for the Mechanism and Machine Science. **Mechanism and Machine Theory**, vol. 38, p. 597–605 , 2003.

JIANG, Z. H.; ISHIDA, T.; Trajectory Tracking Control of Industrial Robot manipulators Using a Neural Network Controller. Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference, 2007.

JIAZHONG, X.; LEI, Z.; MING, Q. Research on Electronic Cam Based on Nurbs Interpolation Algorithm. IEEE. The Ninth International Conference on Electronic Measurements & Instruments.

KAHANER, D.; MOLER, C.; NASH, S., **Numerical Methods and Software**. 1.ed. New Jersey: Prentice-Hall, 1989. 483 p.

KIM, F.; KIM, S.; KIM, S.; KIM, D. A Practical Approach for Minimum-Time Trajectory Planning for Industrial Robots. **Industrial Robot: An International Journal**. v. 37, n.1 p. 51-61, 2010.

LAI, Y.; L.; Tool-path generation of planar NURBS curves. Robotics and Computer-Integrated Manufacturing. V. 26, p. 471–482, 2010.

LAI, Y. L.; Tool-path generation of planar NURBS curves. Revista eletrônica de Robotics and Computer-Integrated Manufacturing. v. 26, n.5, out. 2010. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S0736584510000165> > . Acessado em: 10 nov. 2011.

LARTIGUE, C.; THIEBAUT, F.; MAEKAWA, T.; CNC tool path in terms of B-spline curves. **Computer-Aided Design**. v. 33, n. 4, p. 307 – 319, April 2001.

LEI, W. T.; WANG, S. B.; Robust real-time NURBS path interpolators. v. 49, n. 7, jan. 2009. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S089069550900011X> > . Acessado em: 10 nov. 2011.

LEI, W. T.; WANG, S. B.; Robust real-time NURBS path interpolators. **International Journal of Machine Tools and Manufacture**. V. 49, n. 7, p. 625-633, june 2009.

LLOYD, J., HAYWARD, V., 1993, "Trajectory Generation for Sensor-Driven and Time-Varying Tasks", Int. J. of Robotics Research, Vol. 12, no 4, pp: 380-393.

MACFARLANE, S. On-Line Smooth Trajectory Planning for Manipulators. 2001. 164 f. M.Sc. Thesis, The University of British Columbia, Columbia Britânica, Canadá.

MACFARLANE, S. On-Line Smooth Trajectory Planning for Manipulators. 2001. 164 f. M.Sc. Thesis, The University of British Columbia, Columbia Britânica, Canadá.

MASEY, R. J. M; GRAY, J. O.; DODD, T. J.; CALDWELL, D. G., Elliptical Point to Point Trajectory Planning using Electronic Cam Motion Profiles for High Speed Industrial Pick and Place Robots. 14TH IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES & FACTORY AUTOMATION, 2009, Piscataway, USA.

OKAWA, I.; NONAKA, K.; Optimal Online Generation of Obstacle Avoidance Trajectory running on a Low Speed Embedded CPU for Vehicles. **Multi-Conference on Systems and Control**, Yokohama, Japan, set., 2010.

OLABI, A.; BEAREE, R.; NYIRI, E.; GIBARU, O.; Enhanced Trajectory Planning For Machining With Industrial Six-Axis Robots.

OZAKI, H.; HIRANO, K.; IWAMURA, M.; LIN, C.; SHIMOGAWA, T.; Improvement of Trajectory Tracking for Industrial Robot Arms by Learning Control with B-Spline. International Symposium on Assembly and Task Planning. n. 5, 2003, Besançon, p. 264 – 269.

PIEGL, L.; TILLER, W., **The NURBS Book**. 2. ed. N.Y: Springer, 1997. 641 p.

PLESSIS, L. J.; SNYMAN, J. A.; Trajectory-planning through interpolation by overlapping cubic arcs and cubic splines. *International Journal for Numerical Methods in Engineering*, v. 57

QUARTERONI, A.; SACCO, R.; SALERI, F., **Numerical Mathematics**. 1.ed. N.Y: Springer, 2000. 647 p.

RED, E., 2000, “A dynamic Optimal Trajectory Generator for Cartesian Path Following”, *Robotica*, Vol.18, Part 5, pp: 451-458.

ROGERS, D. F., **An Introduction to NURBS: With Historical Perspective**. 1. ed. San Diego: Morgan Kaufmann Publishers, 2001. 309 p.

ROTHBART, H. A. *Cam Desing Handbook*. McGRAW-Hill, 2004, 606p.

RUGGIERO, M. A. G.; LOPES, V. L. R., **Cálculo Numérico: Aspectos Teóricos e Computacionais**. 2. ed. São Paulo: Pearson Education, 1996. 401 p.

SAHAR G., HOLLERBACH, J.M., 1986, “Planning of Minimum-Time Trajectories for Robot Arms”, *The Int. J. of Robotics Research*, vol. 5, no 3, pp: 90-100.

SANTOS, R. R. *Contribuição ao Planejamento de Trajetórias de Robôs através de Elementos de Dinâmica, Controle e Otimização*. 2007. 165 f. Tese de Doutorado – Universidade Federal de Uberlândia, Uberlândia.

SANTOS, R. R., STEFFEN Jr, V., SARAMAGO, S. F. P., Optimal Task Placement of a Serial Robot Manipulator for Manipulability and Mechanical Power Optimization. **Intelligent Information Management**, v. 2, n. 9, p. 512-525, 2010.

SANTOS, R. R.; SARAMAGO, S. F. P.; STEFFEN Jr, V. Robot Path Planning: Avoiding Obstacles. In: 18th International Congress of Mechanical Engineering, 2005, Ouro Preto - MG. Proceedings of the 18th International Congress of Mechanical Engineering, 2005. p. 1-10.

SARAMAGO, S. F. P. ; OLIVEIRA, P. J. ; CARVALHO, J. C. M. ; CECCARELLI, Marco. Trajectory Modeling of CaPaMan (Cassino Parallel Manipulator) by using 4th order B-splines. In: 18th International Congress of Mechanical Engineering, 2005, Ouro Preto. COBEM05, 2005. v. 1. p. 1-8.

SARAMAGO, S. F. P. ; STEFFEN Jr, V. Dynamic Optimization for the Trajectory Planning of Robot Manipulators in the Presence of Obstacles. Journal of the Brazilian Society of Mechanical Sciences, v. XXI, p. 371-383, 1999.

SARAMAGO, S. F. P. ; STEFFEN Jr, V. Optimal Trajectory Planning of Robot Manipulators in the Presence of Moving Obstacles. Machine and Mechanism Theory, Oxford, v. 35, n. 8, p. 1079-1094, 2000.

SARAMAGO, S. F. P. ; STEFFEN JR, V. Trajectory Modeling of Robot Manipulators in the Presence of Obstacles. Journal of Optimization Theory and Applications **JCR**, USA, v. 110, n. 1, p. 17-34, 2001.

SARAMAGO, S. F. P. ; STEFFEN Jr, V. Using B-Splines for Optimal Trajectory Planning. Science and Engineering Journal, v. 1, p. 106-114, 1999

SARAMAGO, S. F. P. ; STEFFEN Jr, V. Using B-Splines for Optimal Trajectory Planning. Science and Engineering Journal, v. 1, p. 106-114, 1999

SARAMAGO, S. F. P.; STEFFEN Jr, V. Dynamic Optimization for the Trajectory Planning of Robot Manipulators in the Presence of Obstacles. In: PROC. 7TH INTERNATIONAL

CONFERENCE ON DYNAMIC PROBLEMS IN MECHANICS - DINAME 97, 1997, Angra dos Reis RJ. 7TH INTERNATIONAL CONFERENCE ON DYNAMIC PROBLEMS IN MECHANICS - DINAME 97, 1997. p. 130-132.

SARAMAGO, S. F. P.; STEFFEN Jr, V. Optimal Path Planning of Robot Manipulators in the Presence of Moving Obstacles. In: Applied Mechanics in the Americas - Proceedings of PACAM 6- VIII Diname, 1999. v. 6. p. 283-286.

SARAMAGO, S. F. P.; STEFFEN Jr, V. Trajectory Modeling of Robot Manipulators in the Presence of Obstacles. Journal of Optimization Theory and Applications, USA, v. 110, n. 1, p. 17-34, 2001.

SARAMAGO, S.F.P, STEFFEN JR, V., 2001, “Trajectory Modeling of Robot Manipulators in the Presence of Obstacles”, Journal of Optimization Theory and Applications, Vol. 110, nº 1, pp: 17-34.

SCARAMBONI, A.; Elementos de Máquinas. Telecurso 2000. v. 2. Rio de Janeiro: Fundação Roberto Marinho, 2003.

SCARAMBONI, A.; Elementos de Máquinas. Telecurso 2000. v. 2. Rio de Janeiro: Fundação Roberto Marinho, 2003.

SCIAVICCO L., SICILIANO, B., 1996, *Modeling and Control of Robot Manipulators*, McGraw-Hill Companies, Inc

SHIN, K. G., MCKAY, N. D., 1985, “Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints”, IEEE Trans. on Autom. Control, Vol. AC-30, no 6, pp: 531-541.

SHIN, K. G., MCKAY, N. D., 1986, "Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators", IEEE Trans. on Autom. Control, Vol. AC-31, no 6, pp: 491- 500.

SHUMAKER, L. L., **Spline Functions: Basic Theory**. 3. Ed. Cambridge, 2007. 582

TAYLOR, R. H. Planning and Execution of Straight Line Manipulator Trajectories, **IBM Journal of Research and Development**, Vol. 23, no 4, pp: 424- 436, 1979

TSAI, L.W. **Robot Analysis: The Mechanics of Serial and Parallel Manipulators**, John Wiley & Sons, New York, 1999.

VISIOLI, A. Trajectory Planning of Robot Manipulators by Using Algebraic and Trigonometric Splines. **Journal Robotica**. New York, v. 18, n. 21, p. 611-631, nov. 2000.

WAGNER, P.; KOTZIAN, J.; KORDAS, J.; MICHNA, V.; Path Planning and Tracking for Robots Based on Cubic Hermite Splines in Real-Time. Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on. P. 1-8. Settembre 2010.

WOELFEL, M. Introduction to Electronic Cam. Assembly Automation. Minnesota v. 19, n. 1, p. 17-24.

ZHENG, T. G.; FRENG, L.; CHAO, W. Y. Real-time accurate hand path tracking and joint trajectory planning for industrial robots. **Journal of Central South University of Technology**, v. 9, n. 3, p. 191-196, 2002.

ANEXOS

Anexo I – Função que Calcula Spline Natural Cúbica

```
function [xx yy] = SplineCubica(x,y)
```

```
%Esta função realiza a interpolação por spline cúbica natural
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Entradas%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%x é o vetor das ordenadas dos pontos que se deseja interpolar
```

```
%y é o vetor das abscissas dos pontos que se deseja interpolar
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Saídas%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% xx, vetor das ordenadas da spline
```

```
%yy, vetor das abscissas da spline
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
n = length(x);
```

```
%Número de pontos
```

```
A = zeros(4*(n-1));
```

```
%Para n pontos, (n-1) vão ser interpolados. Assim, tem-se (n-1)
```

```
%polinômios cúbicos. Para cada polinômio são encontrados 4 coeficientes,
```

```
%então o sistema linear deve ser  $[4*(n-1)] \times [4*(n-1)]$ . A matriz é organizada
```

```
%de tal forma que a cada 4 colunas são armazenados os dados de cada
```

```
%polinômio. Nas primeiras 4 são os dados do primeiro polinômio e nas
```

```
%últimas 4, do último polinômio.
```

```
j = 1;
```

```
c = 0;
```

```

for i = 1:(n-2)
%Sub-rotina para garantir a continuidade C1 da spline. A derivada de um
%segmento em um ponto deve ser igual a derivada do outro segmento no mesmo
%ponto. Esta condição existe somente nos pontos interiores. Então no
%sistema linear, a diferença das duas derivadas deve ser nula. A cada
%interação o a matriz é preenchida.
    c = c+1;
    A(i,j:(j+7)) = [3*x(c+1)^2 2*x(c+1) 1 0 -3*x(c+1)^2 -2*x(c+1) -1 0];
    j = j+4;
end
j = 1;
c = 0;
for i = (n-1):2*(n-2)
%Sub-rotina para garantir a continuidade C2 da spline. A derivada segunda de um
%segmento em um ponto deve ser igual a derivada segunda do outro segmento no mesmo
%ponto. Esta condição existe somente nos pontos interiores. Então no
%sistema linear, a diferença das duas derivadas deve ser nula. A cada
%interação o a matriz é preenchida.
    c = c+1;
    A(i,j:(j+7)) = [6*x(c+1) 2 0 0 -6*x(c+1) -2 0 0];
    j = j+4;
end
j = 1;
c = 2;
f = 0;
i = (2*n-3);
while (i <= (4*n-8))
%Sub-rotina para garantir a continuidade C0 da spline.
    A(i,j:(j+3)) = [x(c)^3 x(c)^2 x(c) 1];
    A(i+1,(j+4):(j+7)) = [x(c)^3 x(c)^2 x(c) 1];
    i = i+2;
    c = c+1;
    j = j+4;
end

```

```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A((4*n-7),1:4) = [x(1)^3 x(1)^2 x(1) 1];
A((4*n-6),(end-3):end) = [x(end)^3 x(end)^2 x(end) 1];
%Valor da spline nas extremidades
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A((4*n-5),1:2) = [6*x(1) 2];
A((4*n-4),(end-3):(end-2)) = [6*x(end) 2];
%Condição de contorno para a spline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y_aux = [];
for i = 2:(n-1)
    y_aux = [y_aux y(i) y(i)];
end
Y = [zeros(1,(2*(n-2))) y_aux y(1) y(end) 0 0]';
%Vetor que contém os valores do lado direito da igualdade do sistema

B = inv(A)*Y;
%Vetor que contém os valores dos coeficientes dos polinômios
j = 1;
xx = [];
yy = [];

for i = 1:(n-1)

    xx1 = linspace(x(i),x(i+1),100);
    yy1 = B(j)*xx1.^3+B(j+1)*xx1.^2+B(j+2)*xx1+B(j+3);
    j = j+4;
    xx = [xx xx1];

```



```

yy = [yy yy1];
end

```

Anexo II – Função que Encontra o Vão em que o Nó se Encontra

```

function i = EncontraVao(n,p,u,U)

%Determina o índice do vão
%%%%%%%%Entradas:
%n - número de pontos menos um
%p - grau da curva
%u - parametro
%U - vetor de nós
%%%%%%%%
if u==U(n+2)
    i = n+1;
    mid = i;
else
    low = p+1;
    high = n+2;
    mid = fix((low+high)/2);

    while ((u<U(mid))||(u>=U(mid+1)))
        if(u<U(mid))
            high = mid;

        else
            low = mid;

        end
        mid = fix((low+high)/2);
    end
end

```

```
end
i = mid-1;
```

Anexo III – Função que Calcula as Funções de Base não Nulas

```
function N = FuncaoBase(i,u,p,U)

i = i+1;
% Continuar e fazer as mudanças para os índices começando em 1.
%Não haverá mudança nas funções de base, somente os índices ficarão
%deslocados
%%%%%%%%Entradas
%i - vão de análise
%u - nó de cálculo
%p - grau da função de base
%U - Vetor de nós
%%%%%%%%
%N(0) = 1, mas não existe índice zero
N_0 = 1;
for j = 1:p
    % Continuar e fazer as mudanças para os índices começando em 1.
    %Não haverá mudança nas funções de base, somente os índices ficarão
    %deslocados
    esquerdo(j) = u-U(i+1-j);
    direito(j) = U(i+j)-u;
    salvo = 0;

    for r = 0:(j-1)
        if (r==0)
            temp = N_0/(direito(1)+esquerdo(j));
            N_0 = salvo+direito(1)*temp;
```

```

    salvo = esquerdo(j)*temp;
    else
    temp = N(r)/(direito(r+1)+esquerdo(j-r));
    N(r) = salvo+direito(r+1)*temp;
    salvo = esquerdo(j-r)*temp;
    end
end
N(j) = salvo;
end

N = [N_0 N];

```

Anexo IV – Função que Calcula o Ponto da B-spline

```

function C = PontoCurva(n,p,U,P,u)
%Calcula o ponto na curva referente ao nó u
vao = EncontraVao(n,p,u,U);
N = FuncaoBase(vao,u,p,U);
C = 0;
for i = 1:(p+1)
    C = C+N(i)*P(vao-p+i);
    %Eq. (5.20)
end

```