

LUCIANO LUPORINI MENEGALDO

**MODELAGEM BIOMECÂNICA E CONTROLE ÓTIMO
DA POSTURA HUMANA ATRAVÉS DE ALGORITMOS
BASEADOS NA TEORIA DAS APROXIMAÇÕES
CONSISTENTES**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Engenharia

São Paulo
2001

LUCIANO LUPORINI MENEGALDO

**MODELAGEM BIOMECÂNICA E CONTROLE ÓTIMO
DA POSTURA HUMANA ATRAVÉS DE ALGORITMOS
BASEADOS NA TEORIA DAS APROXIMAÇÕES
CONSISTENTES**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Engenharia

Área de Concentração:
Engenharia Mecânica

Orientador:
Agenor de Toledo Fleury

2001

Menegaldo, Luciano Luporini

Modelagem Biomecânica e Controle Ótimo da Postura Humana Através de Algoritmos Baseados na Teoria das Aproximações Consistentes. São Paulo, 2001

315 pp.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1. Biomecânica 2. Controle Ótimo 3. Postura Humana 4. Controle Motor I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II. t

Aos meus pais. Animus et Anima.

Agradecimentos

Agradecemos inicialmente à Fapesp pelo Projeto de Auxílio à Pesquisa concedido para a realização desse trabalho (Proc. N° 97/12626-7), ao CNPq pela Bolsa de Doutorado e ao Depto. de Engenharia Mecânica - EPUSP. Sem o auxílio dessas instituições, não seria viável realizar essa tese.

Agradecemos aos muitos amigos do PME e do PMR, pelos ombros amigos e pelos fraternais cafezinhos, generosamente oferecidos nos momentos de obnubilação. Especialmente, aos nossos colegas Melqui, Adinan, Fernando, Fred, Pacífico, Cap. Raul, Emerson, Alison, Walter, Flávio, Zé Augusto, James, Craig, Gerson e tantos outros.

Especialmente gratos somos ao Prof. Hans Weber e ao Prof. Raúl Gonzalez, pela amizade, confiança e ajuda de todos esses anos.

Agradecemos finalmente ao Prof. Agenor Fleury pela amizade, confiança e conhecimento exemplares que, mais uma vez, demonstrou ao orientar esse trabalho. Particularmente, pela paciência heróica ao suportar as nossas obtusidades e teimosias.

Por que a cabeça da gente é uma só, e as coisas que há e que estão para haver são demais de muitas, muito maiores diferentes, e a gente tem de necessitar de aumentar a cabeça, para o total.

Digo: o real não está na saída nem na entrada: ele se dispõe para a gente é no meio da travessia.

O senhor por hora mal me entende, se é que no fim me entenderá. Mas a vida não é entendível.

João Guimarães Rosa, Grande Sertão: Veredas

Gloria Dei celare verbum et gloria regum investigare sermonem.

Prov. 25, 2

Sumário

Capítulo 1.....	1
Introdução.....	1
Capítulo 2.....	13
Modelo geométrico do membro inferior.....	13
2.1 Revisão bibliográfica.....	15
2.2 Descrição do ambiente SIMM.....	31
2.2.1 Arquivos de ossos (<i>bone files</i>).....	31
2.2.2 Arquivos de músculos (<i>muscle files</i>).....	31
2.2.3 Arquivos de articulação (<i>joint files</i>).....	33
2.3 Determinação das equações de regressão para o comprimento e braços de momento dos atuadores músculo-tendíneos.....	34
Capítulo 3.....	50
Refinamento e linearização de um modelo da mecânica da contração muscular.....	50
Capítulo 4.....	65
Formulação do Problema de Controle Ótimo.....	65
4.1 Teoria das Aproximações Consistentes.....	75
4.2 Definição do problema de controle ótimo (OCP).....	79
4.3 Construção de uma família de problemas de aproximação para o problema de controle ótimo.....	83
4.3.1 Definição dos subespaços de controle de dimensão finita.....	83
4.3.2 Definição dos problemas de aproximação.....	93

4.4	Cálculo de gradientes e consistência das funções de otimalidade do problema de aproximação.....	95
4.5	Determinação de uma base ortonormal de coordenadas para o espaço de coeficientes	97
4.6	Definição dos problemas de aproximação baseados em splines.....	99
4.7	Cálculo dos gradientes das funções de custo e de vínculos no espaço de funções spline.....	102
4.8	Transformação de coordenadas com representações por splines.....	103
4.9	Formulação do problema discreto de controle ótimo e equivalência com o problema de programação matemática	104
4.10	Solução do problema de programação matemática com vínculos de desigualdade (<i>simple bounds</i>).....	108
4.11	Solução do problema de programação matemática com vínculos gerais (método SQP)	113
4.12	Utilização de integradores com passo variável e múltiplo	116
4.13	Seleção da ordem de integração e da aproximação por splines.....	118
4.14	Estimação dos erros de integração e aproximação e definição de estratégias de redistribuição da malha de discretização	122
4.15	Utilização do RIOTS_95	124
4.16	Considerações gerais e limitações do método	128
Capítulo 5.....		130
Testes preliminares		130
5.1	Pêndulo com um grau de liberdade, atuador de torque.....	131
5.1.1	Experimentos com tempo final aberto	140

5.2	Pêndulo triplo com atuadores de torque sem dinâmica e dinâmicos	142
5.2.1	Atuadores de torque sem dinâmica.....	142
5.2.2	Atuadores de torque dinâmicos lineares de 2ª ordem	157
5.3	Pêndulo triplo com atuadores musculares	162
5.3.1	Atuadores musculares lineares.....	162
5.3.2	Problema do tempo crítico.....	169
5.4	Pêndulo triplo com atuadores musculares não-lineares e braços de momento fixos	
	173	
5.4.1	Testes com tempo final 0.4 segundos e com tempo acrescido	176
5.4.2	Testes com tempo final 1 segundo e aceleração da gravidade variável....	185
	Capítulo 6.....	207
	Resultados com atuadores não lineares e braços de momento variáveis.....	207
6.1	Resultados com $t_f=0.4$ segundos.....	211
6.2	Resultados com $t_f= 1.0$ segundo, gravidade variável	242
6.3	Análise dos padrões de ativação obtidos	261
	Capítulo 7.....	264
	Problema da Realimentação.....	264
	Capítulo 8.....	271
	Discussão e perspectivas.....	271
	Referências Bibliográficas.....	277

Lista de Figuras

Figura 2.1: Modelo geométrico do membro inferior	14
Figura 2.2: Localização dos sistemas de referência da pélvis, fêmur, tíbia e tornozelo (Musculographics Inc., 1997)	17
Figura 2.3: Comprimento do <i>rectus femoris</i> em função das coordenadas generalizadas. A curva real (branca) e a que se refere ao resultado da interpolação (cinza) estão superpostas.....	48
Figura 2.4: Braço de momento do <i>rectus femoris</i> em relação ao centro da articulação do quadril em função das coordenadas generalizadas. Curva real :branca; interpolada: cinza.....	48
Figura 2.5: Braço de momento do <i>rectus femoris</i> em relação ao centro da articulação do joelho em função das coordenadas generalizadas. Curva real :branca; interpolada: cinza.....	49
Figura 3.1: Modelo viscoelástico da mecânica da contração muscular.....	51
Figura 3.2: Modelo da dinâmica da contração de Zajac (Zajac, 1989).	53
Figura 3.3: Relações força \times comprimento e força \times velocidade. No gráfico da esquerda, a linha contínua representa a relação força \times comprimento da componente ativa do músculo (para $a=1$), a linha tracejada a componente passiva e a outra linha a soma das duas anteriores.....	54
Figura 3.4: Curva teórica de rigidez do tendão (Zajac, 1989).....	57
Figura 3.5: Identificação do modelo linear da mecânica muscular.	62
Figura 3.6: Identificação do modelo linear da mecânica muscular	64
Figura 4.1: Árvore mostrando os métodos disponíveis de controle ótimo	67

Figura 5.1: Pêndulo com 1 grau de liberdade.	131
Figura 5.2: Pêndulo simples com tempo final fechado.....	139
Figura 5.3: pêndulo de 1 grau de liberdade e tempo final aberto	141
Figura 5.4: Modelo de três corpos rígidos da postura humana.....	144
Figura 5.5: Pêndulo 3gl, atuadores de torque, $Q=[10\ 10\ 10\ 1\ 1\ 1]$, $R=[1\ 1\ 1]$, tempo CPU=7.84 min.	154
Figura 5.6: Pêndulo 3gl, atuadores de torque, $Q=[1\ 1\ 1\ 1\ 1\ 1]$, $R=[1\ 1\ 1]$, tempo CPU=35.01 min.	155
Figura 5.7: Pêndulo 3gl, atuadores de torque, $Q=[0\ 0\ 0\ 0\ 0\ 0]$, $R=[1\ 1\ 1]$, tempo CPU=11.09 min.	156
Figura 5.8: Pêndulo triplo com atuadores de torque sem dinâmica e função de custo dada pelo centro de massa. Tempo de CPU = 1.54 min.....	157
Figura 5.9: Estados e controle do pêndulo triplo com atuadores de torque lineares de 2ª ordem e função de custo tipo LQR com $Q=[10\ 10\ 10\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$ e $R=[1\ 1\ 1]$. Tempo de CPU = 49.81 minutos, sem convergência.....	160
Figura 5.10: Estados e controle do pêndulo triplo com atuadores de torque lineares de 2ª ordem e função de custo = soma dos quadrados das derivadas temporais dos torques. Tempo de CPU = 4.06 minutos.	161
Figura 5.11: Comportamento do sistema controlado por malha de realimentação calculada por LQR. O gráfico superior mostra os estados do sistema em malha fechada, isto é, integrando as equações diferenciais associadas à matriz de ganhos. No gráfico central estão mostrados os sinais de controle calculados multiplicando-se os estados pelo matriz de ganhos. Em baixo, o sinal de controle foi introduzido no sistema em malha aberta.....	165

Figura 5.12: Estimativa inicial do controle calculada com RK1 utilizando vínculos de desigualdade. Tempo de CPU = 825.84 min (Pentium II 400Mhz).	166
Figura 5.13: Solução do pêndulo triplo com 8 atuadores lineares, RK1. Tempo de CPU = 1.045,8 minutos (Pentium II 400Mhz).....	167
Figura 5.14: Estimativa inicial do controle calculada com RK4 utilizando vínculos de desigualdade. Tempo de CPU = 1.126,1 minutos (Pentium II 400Mhz).	168
Figura 5.15 Solução do pêndulo triplo com 8 atuadores lineares, RK1. Tempo de CPU = 1274 minutos.....	169
Figura 5.16: Solução para $t > t_{crit}$, em dois intervalos, com,vínculos de desigualdade e RK1.....	171
Figura 5.17: Solução final para $t > t_{crit}$, com vínculos de igualdade, utilizando como estimativa inicial o controle mostrado na figura anterior. Tempo de cpu = apx. 100 horas num Pentium II 400 MHz.	172
Figura 5.18: Trajetória 12_9_4.....	189
Figura 5.19: Trajetória 13_9_4.....	191
Figura 5.20: Trajetória 25_9_1.....	193
Figura 5.21: Trajetória 24_9_5.....	195
Figura 5.22: Trajetória 24_9_3.....	197
Figura 5.23: Trajetória 25_10_1.....	199
Figura 5.24: Trajetória 25_11_3.....	201
Figura 5.25: Ativações neuromusculares 25_11_3.....	203
Figura 5.26: Forças musculares 25_11_3.....	204
Figura 5.27: Trajetória 7_11_1.....	205
Figura 6.1: Grupos musculares denotados pelos seus componentes mais importantes	208

Figura 6.2: Trajetória 2_12_1	216
Figura 6.3: Trajetória 2_12_2	218
Figura 6.4: Trajetória 3_12_4	220
Figura 6.5: Trajetória 4_12_1	222
Figura 6.6: Trajetória 30_11_5	224
Figura 6.7: Trajetória 1_12_1	226
Figura 6.8: Trajetória 5_12_3	228
Figura 6.9: Trajetória 6_12_6	230
Figura 6.10: Trajetória 11_12_2	232
Figura 6.11: Trajetória 11_12_3	234
Figura 6.12: Trajetória 13_12_2	236
Figura 6.13: Trajetória 13_12_3	238
Figura 6.14: Ativações neuromusculares 13_12_3	240
Figura 6.15: Forças musculares 13_12_3	241
Figura 6.16: Trajetória com controle inicial u_{00}	245
Figura 6.17: Trejetória 14_12_1	247
Figura 6.18: Trejetória 14_12_3	249
Figura 6.19: Trejetória 1_1_1	251
Figura 6.20: Trejetória 6_1_1	253
Figura 6.21: Trejetória 7_1_1	255
Figura 6.22: Trejetória 8_1_1	257
Figura 6.23: Ativações neuro-musculares 8_1_1	259
Figura 6.24: Forças musculares 8_1_1	260

Lista de Tabelas

Tabela 2.1: Características dos modelos de articulação utilizadas no modelo do membro inferior.	16
Tabela 2.2: Funções cinemáticas. Os valores mostrados na 2ª coluna (mostrados por segmentos de reta) são interpolados por uma spline cúbica (curva suave) pelo SIMM. As amplitudes admissíveis de movimento estão mostradas na Tabela 2.7.....	19
Tabela 2.3: Transformações entre os segmentos.	20
Tabela 2.4: Coordenadas dos eixos de rotação entre o primeiro e o segundo segmento de cada articulação, definidos na Tabela 2.3.	20
Tabela 2.5: Músculos utilizados, suas abreviaturas e coordenadas de origem e inserção. Duas coordenadas eventualmente definidas no mesmo segmento não alteram o L^{MT} com a variação das coordenadas generalizadas; configuram apenas o caminho (<i>path</i>) do atuador músculo-tendíneo, necessário entretanto para o cálculo correto de seu comprimento total.	24
Tabela 2.6: Localização dos sistemas de referência. Para todos os segmentos, o eixo x aponta anteriormente, y superiormente e z lateralmente.	25
Tabela 2.7: Coordenadas generalizadas e sua amplitude admissível de movimento.....	25
Tabela 2.8: Parâmetros musculares e trabalhos que foram utilizados para escolha dos parâmetros (1): Wickiewicz et al., 1993, (2): Brand et al., 1986, (3): Friederich e Brand, 1990.	30
Tabela 2.9: Cada coluna significa um grau de liberdade, sendo: HF=flexão do quadril (e=extensão, f=flexão); HA=adução do quadril (ad=adução, ab=abdução); HR=rotação do quadril (i=interna, e=externa) KA=ângulo do joelho (e=extensão,	

f=flexão); SA=ângulo subtalar (i=inversão, e=eversão); MA= ângulo metatarso-falangeal,(f=flexão, e=extensão).	35
Tabela 2.10: Grupos musculares selecionados com critério de possuírem parâmetros dependentes das mesmas coordenadas generalizadas.....	41
Tabela 2.11: Equações propostas para ajuste das curvas de L^{MT} e r em função das coordenadas generalizadas.....	46
Tabela 2.12: Seleção das equações de regressão com mínimo erro de ajuste, conforme Tabela 2.11 e Anexo 1 (x - parâmetro não definido para o músculo)	47
Tabela 3.1: Pontos utilizados na geração das curvas força × comprimento da porção ativa do músculo.....	56
Tabela 4.1: Erros de representação e de integração em função da ordem de integração e do nível de discretização (Schwartz, 1996). $\Delta_N = \max_k \Delta_{N, k}$	119
Tabela 4.2: Ordem possível de escolha das splines em função da ordem de integração.....	120
Tabela 4.3: Recomendações para escolha dos parâmetros de solução.	121
Tabela 5.1: Parâmetros musculares em posição anatômica. HF, KA e AA são as três coordenadas generalizadas relevantes no plano sagital: flexão do quadril, ângulo do joelho e ângulo do tornozelo. Os parâmetros r_1 , r_2 e r_3 são os braços de momento em relação a cada eixo de rotação para cada músculo. Os grupos na última coluna foram selecionados de acordo com a semelhança funcional dos músculos que os compõe.	174
Tabela 5.2: Grupos musculares selecionados e seus parâmetros.....	175
Tabela 5.3: Resultados sem tempo aumentado e com tempo aumentado, r fixo.....	183
Tabela 5.4: Resultados com tempo final 0.4 segundos e aceleração da gravidade variável	187

Tabela 6.1: Curvas médias dos braços de momento para os 10 grupos musculares selecionados. As superfícies mostradas na segunda parte da tabela se referem aos braços de momento de músculos biarticulares em relação a cada uma das articulações envolvidas. Para esses grupos, as curvas de cada componente não estão mostradas.

..... 210

Tabela 6.2: Simulações com braço de momento variável, tempo final 0.4 segundos .. 215

Tabela 6.3: Resultados com braço de momento variável, 1 segundo de simulação, gravidade variável..... 244

Lista de Abreviaturas

gmed1	<i>gluteus medius</i>
gmed2	<i>gluteus medius</i>
gmed3	<i>gluteus medius</i>
gmin1	<i>gluteus minimus</i>
gmin2	<i>gluteus minimus</i>
gmin3	<i>gluteus minimus</i>
semimem	<i>Semimembranosus</i>
semiten	<i>Semitendinosus</i>
bifemlh	<i>biceps femoris</i>
bifemsh	<i>biceps femoris</i>
sar	<i>Sartorius</i>
addlong	<i>adductor longus</i>
addbrev	<i>adductor brevis</i>
amag1	<i>adductor magnus</i>
amag2	<i>adductor magnus</i>
amag3	<i>adductor magnus</i>
tfl	<i>tensor fasciae latae</i>
pect	<i>Pectineus</i>
gra	<i>Gracilis</i>
gmax1	<i>gluteus maximus</i>
gmax2	<i>gluteus maximus</i>
gmax3	<i>gluteus maximus</i>
iliacus	<i>Iliacus</i>
psoas	<i>Psoas</i>
quadfem	<i>quadatus femoris</i>
gem	<i>Gemelli</i>
peri	<i>Periformis</i>
rf	<i>rectus femoris</i>
vasmed	<i>vastus medialis</i>
vasint	<i>vastus intermedius</i>
vaslat	<i>vastus lateralis</i>
medgas	<i>Gastrocnemius</i>
latgas	<i>Gastrocnemius</i>
sol	<i>Soleus</i>
tibpost	<i>tibialis posterior</i>
flexdig	<i>flexor digitorum longus</i>
flexhal	<i>flexor hallucis longus</i>
tibant	<i>tibialis anterior</i>
perbrev	<i>peroneus brevis</i>
perlong	<i>peroneus longus</i>
pertert	<i>peroneus tertius</i>
extdig	<i>extensor digitorum longus</i>
exthal	<i>extensor hallucis longus</i>
ligpat	<i>ligamento patelar</i>

SIMM – *Software for Interactive Musculoskeletal Modelling*

PCSA: Physiologica Cross-Sectional Area

HF=flexão do quadril (e=extensão, f=flexão)

HA=adução do quadril (ad=adução, ab=abdução)

HR=rotação do quadril (i=interna, e=externa)

KA=ângulo do joelho (e=extensão, f=flexão)

SA=ângulo subtalar (i=inversão, e=eversão)

MA= ângulo metatarso-falangeal,(f=flexão, e=extensão).

OCP: *Optimal Control Problem*

IP: índice de performance

TPBVP: *Two Point Boundary Value Problems*

RK1, ..., RK4: Runge-Kutta de 1^a, ..., 4^a ordem

SQP: *Sequential Quadratic Programming*

PLTE: erro local principal de truncamento

LTE: erro local de truncamento

Lista de Símbolos

Capítulo 1,2 e 3

$F^M, (\tilde{F}^M)$ = força muscular

$F^{M*}, (\tilde{F}^{M*})$ = força muscular quando o músculo está no comprimento ótimo

$F^{CE}, (\tilde{F}^{CE})$ = força no elemento contrátil

$F^{PE}, (\tilde{F}^{PE})$ = força no elemento passivo

$F^T, (\tilde{F}^T)$ = força no tendão

L^s = comprimento do sarcômero

$L^M, (\tilde{L}^M)$ = comprimento do músculo

$L^T, (\tilde{L}^T)$ = comprimento do tendão

$L^{MT}, (\tilde{L}^{MT})$ = comprimento do atuador músculo-tendíneo

$v^M, (\tilde{v}^M)$ = velocidade da fibra muscular

$v_a^M, (\tilde{v}_a^M)$ = velocidade de alongamento da fibra muscular

$v_e^M, (\tilde{v}_e^M)$ = velocidade encurtamento da fibra muscular

$v^T, (\tilde{v}^T)$ = velocidade do tendão

$v^{MT}, (\tilde{v}^{MT})$ = velocidade do atuador músculo-tendíneo

$t, (\tau)$ = tempo

$\sigma^T, (\tilde{\sigma}^T)$ = tensão no tendão

ε^T = deformação do tendão

$u(t), u(\tau)$ = excitação muscular

$a(t), a(\tau)$ = ativação muscular

α_M = ângulo da fibra muscular

L_α^M = comprimento das fibras musculares projetado no eixo do tendão

F_0^M = força isométrica máxima do músculo

L_0^M = comprimento ótimo das fibras musculares (desenvolvendo a força máxima)

α_0 = ângulo ótimo da fibra muscular quando $L^M = L_0^M$

ε_0^T = deformação no tendão quando $F^T = F_0^M$

σ_0^T = tensão no tendão quando $F^T = F_0^M$

$L_s^T, (\tilde{L}_s^T)$ = comprimento do tendão relaxado

$k^{SE}, (\tilde{k}^{SE})$ = rigidez das pontes cruzadas (SEE)

$k^T, (\tilde{k}^T)$ = rigidez do tendão

E^T = módulo de elasticidade do tendão

A^T = área da seção transversal do tendão

fl_a = componente ativa da relação força-comprimento

fl_p = componente passiva da relação força-comprimento

v_m = velocidade máxima de encurtamento das fibras musculares

τ_c = escala de tempo ($\tau_c \equiv L_0^M / v_M$)

$\tau_{act}, (\tilde{\tau}_{act}) =$ constante de tempo para contração

$\tau_{deact}, (\tilde{\tau}_{deact}) =$ constante de tempo para relaxamento

$\beta = \tau_{act} / \tau_{deact}$

$F^{PE} (\tilde{F}^{PE}) =$ Força no elemento elástico em paralelo

$F^{DE} (\tilde{F}^{DE}) =$ Força no elemento viscoso em paralelo

$F^{CE} (\tilde{F}^{CE}) =$ Força no elemento contrátil

$B(\tilde{B}) =$ Coeficiente de amortecimento do elemento viscoso

$k^{PE}, (\tilde{k}^{PE}) =$ rigidez do elemento elástico em paralelo (PE)

$a_m(d) =$ curva de recrutamento em função da largura de pulso da excitação

$T_c = \tau_{deact} =$ constante de tempo para relaxamento

Capítulo 4

$\mathbf{x}(t)$: vetor de estados do sistema

\mathbf{u}^* : controle ótimo

$\theta_N: H_N \rightarrow \mathfrak{R}$ função de otimalidade

$L_2^m[0,1]$: um espaço de funções quadraticamente integráveis no intervalo $[0,1]$

$\langle \mathbf{u}(t), \mathbf{v}(t) \rangle_2 = \int_0^1 \langle \mathbf{u}(t), \mathbf{v}(t) \rangle dt$: produto interno num espaço de funções contínuas

$\|\mathbf{u}(t)\|_2 = \left[\int_0^1 \langle \mathbf{u}(t), \mathbf{u}(t) \rangle dt \right]^{1/2}$: norma num espaço de funções contínuas

$L_{\infty,2}^m[0,1]$: espaço pré-hilbertiano de elementos de controle munido de norma e produto interno de $L_2^m[0,1]$.

$H_{\infty,2} = \mathfrak{R}^n \times L_{\infty,2}^m[0,1] = (\mathfrak{R}^n \times L_{\infty,2}^m[0,1], \langle \cdot, \cdot \rangle_H, |\cdot|_H)$: espaço pré-hilbertiano de controles

ótimos cujos elementos são pares de condições iniciais, variáveis de controle

$\eta = (\xi, u)$: pares de condições iniciais, variáveis de controle

$H_{\infty,2}$: um subespaço denso do espaço de Hilbert $H_2 = \mathfrak{R}^n \times L_2^m[0,1]$

$\langle \eta, \eta' \rangle_H = \langle \xi, \xi' \rangle + \langle u, u' \rangle_{L_2}$

$\|\eta\|_H^2 = \langle \eta, \eta \rangle_H = |\xi|^2 + \|u\|_2^2$

$U \subset B(0, \rho_{max}) = \{u \in \mathfrak{R}^m \mid \|u\| \leq \rho_{max}\}$: um conjunto compacto e convexo de vínculos de controle

$U = \{u \in L_{\infty,2}^m[0,1] \mid u(t) \in U, t \in [0,1]\}$: conjunto de controles admissíveis

$\mathbf{H} = \mathfrak{R}^n \times U \subset H_{\infty,2}$: conjunto de pares (ξ, u) admissíveis contido num conjunto maior

$\mathbf{B} = \mathfrak{R}^n \times \{u \in L_{\infty,2}^m[0,1] \mid u(t) \in B(0, \rho_{max}), t \in [0,1]\} \subset H_{\infty,2}$ dentro do qual são válidos os resultados das equações diferenciais

$\mathbf{q}_0 = \{1, 2, \dots, q_0\}$, q_0 número de funções objetivo

$\mathbf{q}_c = \{1, 2, \dots, q_c\}$, q_c número de funções de vínculo

$\mathbf{q} = \{1, 2, \dots, q\}$, $q = q_0 + q_c$, $\mathbf{q}_c + q_0 = \{1 + q_0, \dots, q_c + q_0\}$

$\mathbf{x}^n(1)$: solução no tempo final do sistema dinâmico $\dot{\mathbf{x}} = \mathbf{h}$ com a condição inicial η .

$f^v: \mathbf{B} \rightarrow \mathfrak{R}$, $f^v(\eta) = \zeta^v(\xi, x^n(1))$: solução da v -ésima função f no tempo final

$$H_N = \mathfrak{R}^n \times L_N \subset H_{\infty,2}$$

L_N : subespaço de dimensão finita de $L_{\infty,2}^m[0,1]$, gerado a partir de funções contínuas por partes

$$\bar{x}_{k+1} = \bar{x}_k + \Delta \sum_{i=1}^s b_i K_{k,i}, \quad \bar{x}_0 = x(0) = \xi, \quad k \in \{0, 1, \dots, N-1\}$$

$$\Delta = 1/N$$

$$t_k = k\Delta$$

$$\tau_{k,i} = t_k + c_i \Delta$$

$a_{i,j}$, c_i , b_i determinam o método RK utilizado, e podem ser organizados na chamada *matriz de Butcher* $\mathbf{A} = [c, \mathbf{A}, \mathbf{b}^T]$

\bar{x}_k : estimativa de $x(t_k)$. A chamada *malha de integração* ou *malha de discretização* corresponde à seqüência $\{t_k\}_{k=0}^N$.

$$\mathbf{I} = \{i_1, i_2, \dots, i_r\} = \{i \in \mathbf{s} \mid c_j \neq c_i, \forall j \in \mathbf{s}, j < i\},$$

$$\mathbf{I}_j = \{i \in \mathbf{s} \mid c_j = c_i, i_j \in \mathbf{I}\}, j \in \mathbf{r}$$

r : número de valores diferentes das componentes c_i do vetor de Butcher

\mathbf{r} : conjunto dos valores diferentes das componentes c_i do vetor de Butcher

$$\bar{\mathbf{u}} = (\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_{N-1}), \quad \bar{\mathbf{u}}_k \in \times_{\mathbf{r}} \mathfrak{R}^m$$

$$\bar{\mathbf{u}}_k = (\bar{u}_{k,1}, \dots, \bar{u}_{k,r}),$$

$$\bar{u}_{k,j} \in \mathfrak{R}^m, \quad j \in \mathbf{r}$$

cada $\bar{u}_{k,j}$ corresponde a uma amostra de controle $u[\tau_{k,ij}]$

$$\langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle_{l_2} = \sum_{N=0}^{N-1} \sum_{j=1}^r \langle \bar{u}_{k,j}, \bar{v}_{k,j} \rangle : \text{produto interno Euclidiano}$$

$\mathbf{1}_j$: vetor com comprimento igual a \mathbf{I}_j de elementos unitários

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_1 & & & \\ & \mathbf{1}_1 & & \\ & & \ddots & \\ & & & \mathbf{1}_r \end{bmatrix}$$

$$\omega_k = \bar{\mathbf{u}}_k \mathbf{G} = [\bar{u}_{k,1}, \dots, \bar{u}_{k,r}] \mathbf{G}$$

$$T_k^1 = [t_k, t_{k+1})$$

$$L_N^1 = \left\{ u \in L_2^m[0,1] \mid u(t) = \sum_{k=0}^{N-1} \sum_{j=1}^r \bar{u}_{k,j} \Phi_{N,k,j}^1(t), \quad \bar{u}_{k,j} \in \mathfrak{R}^m, \quad \forall t \in [0,1] \right\} : \text{subespaço de}$$

controle L_N^1 de dimensão finita

$\Phi_{N,k,j}^1(t)$ sendo a *função base* de L_N^1

$\phi_{N,k,j}(t)$ j -ésimo polinômio de Lagrange,

$$\bar{L}_N^1 = \left(\times_{N \times \mathbf{r}} \mathfrak{R}^m, \langle \cdot, \cdot \rangle_{\bar{L}_N^1}, \|\cdot\|_{\bar{L}_N^1} \right)$$

$V_{A,N}^1$ é um mapa $V_{A,N}^1 : L_N^r \rightarrow \times_{N,r} \mathfrak{R}^m$; $u = V_{A,N}^1(\bar{u})$

$W_{A,N} : H_N \rightarrow \bar{H}_N$, $W_{A,N}(\eta) = (\xi, V_{A,N}(u))$

$H_N = \mathfrak{R}^n \times V_{A,N}^{-1}(\bar{u}_N) \subset H_N$: conjunto de vínculos para o problema de aproximação no espaço de funções (polinomiais contínuas por partes)

$\bar{H}_N = \mathfrak{R}^n \times \bar{u}_N \subset \bar{H}_N$: idem, no espaço de coeficientes

$\Psi_{0,N}(\eta) = \max_{v \in \mathbf{q}_0} f_N^v(\eta)$

$\Psi_{c,N}(\eta) = \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f_N^v(\eta)$

$\bar{f}_N^v : \bar{H}_N \rightarrow \mathfrak{R}$, $\bar{f}_N^v = \zeta^v(\xi, \bar{x}_N^{\bar{\eta}})$, $v \in \mathbf{q}$, $\bar{\eta} = (\xi, \bar{u}) = W_{A,N}(\eta)$

$\nabla f_N^v(\eta) = (\nabla_\xi f_N^v(\eta), \nabla_u f_N^v(\eta))$: gradiente da função de custo

F_x e F_{w_j} : derivadas de $F(x,w)$ com relação a x e ao j -ésimo componente de w_j

$\tilde{L}_N = \times_{N,r} \mathfrak{R}^m$ espaço de coeficientes munido de norma e produto interno l_2

$Q: \bar{L}_N \rightarrow \tilde{L}_N$: uma transformação $\tilde{u} = Q(\bar{u}) = \bar{u}M_N^{1/2}$

$\tilde{\eta} = (\xi, \tilde{u})$

$\tilde{f}_N^v(\tilde{\eta}) = \bar{f}_N^v((\xi, \tilde{u}M_N^{-1/2}))$

$L_N^{(r)} = \left\{ u \in L_{\infty,2}^m[0,1] \mid u(t) = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t), \quad t \in [0,1] \right\}$: espaço de coeficientes de controle de

dimensão finita de B-splines

$t_N = \left\{ \frac{k}{N} \right\}_{k=-r+1}^{N+r-1}$ ou $t_N = \{t_k\}_{k=-r+1}^{N+r-1}$

$U_N^{(r)} = \{u \in L_N^{(r)} \mid \alpha_k \in U, k = 1, \dots, N+r-1\}$: conjuntos de vínculos de controle para o problema de aproximação definido por splines

$\bar{\alpha}_k = \{\alpha_k\}_{k=1}^{N+r-1} \in \bar{L}_N^{(r)}$, $\alpha_k \in \mathfrak{R}^m$: variáveis de controle $u = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t) \in L_N^{(r)}$ no espaço de coeficientes spline

$S_{N,r} = L_N^{(r)} \rightarrow \bar{L}_N^{(r)}$: mapa entre os elementos u e $\bar{\alpha}_k$

$I_u^v(t, x(t), u(t)) \leq 0$: vínculos de desigualdade nas trajetórias

$g_{ei}^v(\xi, x(b)) \leq 0$: vínculos de desigualdade terminais

$g_{ee}^v(\xi, x(b)) = 0$: vínculos de igualdade terminais

Resumo

Uma formulação visando o aprimoramento de modelos matemáticos da biomecânica da postura humana (Menegaldo, 1997; Menegaldo e Weber, 1998) é apresentada neste trabalho. A classe de modelos propostos contém sub-modelos envolvendo: múltiplos corpos rígidos, mecânica muscular, dinâmica da ativação neural e geometria do membro inferior. Aplicou-se aos modelos estudados uma metodologia para a determinação de controles ótimos em malha aberta utilizando algoritmos baseados na Teoria das Aproximações Consistentes (Polak, 1996; Schwartz, 1996), como uma tentativa de reproduzir tarefas posturais nos modelos biomecânicos propostos, bem como determinar funcionais fisiologicamente plausíveis para a solução do problema da redundância dos atuadores musculares. Aplicações futuras do trabalho prevêem a simulação do comportamento postural em indivíduos normais e em portadores de disfunções, candidatos a cirurgias corretivas no membro inferior.

Abstract

This work presents improvements on a biomechanical model of the human posture developed by the author (Menegaldo, 1997; Menegaldo e Weber, 1998). Several sub-models were analyzed, dealing with multi-body dynamics, muscle mechanics, neuromuscular activation dynamics and lower limb musculotendon geometry. An optimal control methodology developed by Schwartz (1996), based on Polak's Consistent Approximations Theory (Polak, 1996), has been used to find open-loop muscular activations patterns for postural tasks. Optimal cost functions were chosen and tested to solve the optimal control problem, at the same time that the actuator redundancy problem is solved in a physiologically feasible way. The resulting methodology is to be applied, in the future, for numerical simulation of the posture behavior in normal or impaired people who are candidate to orthopedic surgeries procedures.

Capítulo 1

Introdução

Modelos matemáticos de sistemas biológicos constituem um dos campos com maior crescimento no desenvolvimento científico atual. Apesar do enorme arcabouço de técnicas de modelagem matemática e simulação hoje disponíveis, sua aplicação generalizada em problemas prementes do ponto de vista clínico é ainda incipiente.

Com relação ao sistema músculo-esquelético, a disponibilidade de modelos mecânicos adequados pode fornecer contribuições significativas para o entendimento e tratamento de suas afecções. Entretanto, o desenvolvimento de tais modelos exige um aporte de esforços e recursos proporcional à sua complexidade. Uma análise introdutória da natureza dos modelos em questão revela que se tratam, na realidade, de conjuntos de sub-modelos integrados:

- Modelos de múltiplos corpos rígidos, semelhantes aos utilizados na robótica
- Modelos da mecânica muscular

- Modelos geométricos, capazes de relacionar as forças musculares com os torques nas articulações dos modelos de múltiplos corpos rígidos

Diversas características observadas neste sistema biológico introduzem um alto grau de complexidade aos modelos em questão: tratam-se, quase sempre, de modelos dinâmicos não-lineares, com múltiplas entradas e múltiplas saídas, muitos graus de liberdade redundantes, com fortes acoplamentos, etc. Acrescente-se ainda a grande dificuldade na determinação de parâmetros - variáveis alguns deles com o tempo, com outros parâmetros ou com variáveis de estado - para cada um dos sub-modelos, tanto de valores médios para uma determinada população quanto para indivíduos singulares.

Bons modelos biomecânicos, entretanto, não são suficientes para a reprodução, através de técnicas de simulação, da maior parte das operações efetuadas pelo sistema músculo-esquelético: o comportamento dinâmico destes modelos depende intrinsecamente do sistema de controle associado. Desde membros individuais até sistemas com múltiplas articulações, o comportamento mecânico só pode ser reprodutível por um modelo que contemple um sistema de controle adequado: uma abordagem que efetivamente reproduza o comportamento biológico do sistema músculo-esquelético com alguma riqueza não pode prescindir nem dos aspectos biomecânicos nem dos relativos ao sistema de controle neuronal.

Três abordagens principais de modelagem do complexo *neuro-mecânico* humano são encontrados na literatura: movimentos uniarticulares, postura e marcha. No primeiro caso, considera-se usualmente um modelo cinemático mais ou menos detalhado da articulação, na tentativa de determinar com precisão esforços tendíneos ou ligamentares (Riener et al., 1996; Balthasar et al., 1996); o problema da redundância de atuadores pode também ser tratada neste sistema mecânico de um grau de liberdade cinemático (Enoka,

1993; Kuo, 1996; Legreneur et al., 1996), concentrando a análise nas estratégias de recrutamento e somação das unidades motoras envolvidas na ação uniarticular específica. A análise da marcha, por sua vez, é a que pode fornecer os subsídios mais relevantes do ponto de vista da reabilitação de portadores de lesões musculoesqueléticas ou neuronais. Apesar disso, modelos detalhados da marcha humana são extremamente complexos, e a interpretação de informações clinicamente relevantes deve ser feita de maneira criteriosa, contando o analista com conhecimentos exaustivos do sistema biológico e das limitações do modelo.

Outro interessante objeto de modelagem do sistema neuro-mecânico são as tarefas posturais, ou a biomecânica da postura. Situa-se a meio caminho entre os sistemas uniarticulares e a marcha: ao mesmo tempo que requer um sistema de controle para várias articulações atuando em conjunto, pode ser formulado com poucos graus de liberdade, facilitando uma eventual análise de estratégias de controle das unidades motoras. Além disso, diversos modelos da biomecânica e do controle da postura, possuindo significativas diferenças de complexidade podem reproduzir, segundo aspectos particulares, comportamentos verificados experimentalmente (Barin, 1989). Isso possibilita a implementação *gradual e modular* dos modelos, permitindo verificações de sub-modelos particulares através de testes preliminares e a adequação aos recursos computacionais disponíveis.

A análise biomecânica da postura, não obstante sua importância no estudo deste mecanismo biológico, bem como de diversos problemas relacionados à teoria de controle motor e à dinâmica de sistemas, possui uma relevância clínica não menos importante: a postura, enquanto conjunto de forças exercidas pelo sistema muscular para controlar a posição relativa entre os segmentos corporais (Massion, 1992), é um pré-requisito para a

obtenção do equilíbrio. Entende-se, neste contexto, por equilíbrio o posicionamento ortostático do corpo, tomado como um único conjunto, em relação ao campo gravitacional.

Na prática clínica, um grande número de patologias são responsáveis direta ou indiretamente por déficits posturais ou de equilíbrio, sendo assim a avaliação destas funções um importante indicador de disfunções no aparelho vestibular e no sistema proprioceptivo (nos receptores periféricos e no sistema coluna dorsal - lemnisco medial). Outras doenças relacionadas com o funcionamento anormal dos núcleos da base, como o mal de Parkinson, são responsáveis ainda por alterações no equilíbrio. Plegias e paresias, provocadas por acidentes vasculares, lesões mecânicas e tumores, introduzem muitas vezes perdas de equilíbrio associadas a problemas posturais.

Como já foi dito, tentativas de simulação computacional do movimento humano, não podendo prescindir de bons modelos biomecânicos, tampouco podem deixar de lado o sistema de controle associado. Para tanto, é necessário em algum nível da modelagem abstrair-se do substrato físico em questão e lançar mão de estratégias de engenharia como uma tentativa de reproduzir o mesmo comportamento. Um exemplo clássico disso é a substituição da rede neuronal biológica, ou mais especificamente o córtex motor e seus principais tratos descendentes, responsáveis pela formulação de algumas estratégias e pela geração de sinais de controle motor, por um modelo de controle ótimo (Pandy et al., 1992; Kuo, 1995; Pandy et al., 1995; Menegaldo, 1997; Raash et al., 1997, Spägele et al., 1999; Kaplan e Heegaard, 1999).

Esta abordagem baseia-se na idéia seminal de Bernstein, de considerar que a distribuição de forças entre diversos músculos que cruzam uma articulação e que produzem uma determinada ação é, do ponto de vista matemático, um problema de otimização. Diversos pesquisadores (An et al., 1995; Yamaguchi et al., 1995; para uma

revisão, ver Tsirakos et al., 1997) utilizaram este princípio na análise de modelos da dinâmica inversa, partindo do comportamento cinemático do corpo humano determinado por meios opto-eletrônicos. Calculando os torques articulares, procura-se determinar, estaticamente, a contribuição de cada força muscular que cruza a articulação. Entretanto, este tipo de análise difere da abordagem através do controle ótimo, que visa de fato determinar as excitações neuro-musculares que levam o sistema de um estado para outro, partindo apenas das equações de movimento e de vínculos. No caso da otimização, procura-se apenas minimizar um funcional algébrico, calculando um conjunto de forças; já no caso do controle ótimo, o objetivo é determinar um conjunto de sinais de controle que executem o movimento respeitando as equações dinâmicas, ao mesmo tempo que é minimizada uma funcional, geralmente integral, que depende dos estados dinâmicos e das variáveis de controle ao longo de todo o movimento. Desde o ponto de vista matemático e numérico, o problema de otimização é um dos sub-problemas do controle ótimo, ao menos segundo os métodos de integração iterativa (Capítulo 4).

As idéias de otimização e controle ótimo aplicadas ao controle motor têm ganhado força nos últimos anos, porém diversas questões permanecem abertas. Se é razoável supor que o sistema neurofisiológico de controle motor busca a minimização de um funcional na geração de trajetórias para o movimento dos membros superior e inferior, várias alternativas de escolha deste funcional ou índice de performance são possíveis: a soma das forças musculares elevadas à potência n ($n=1,2,3..$), a soma das tensões musculares (força/área da seção muscular), das suas derivadas temporais, a soma do calor dissipado na contração muscular, entre outros (Pandy et al. 1995; Buchnan e Shreeve, 1996). A escolha do funcional pode ainda depender do tipo de tarefa que está sendo realizada ou que foi planejada pelo córtex cerebral: movimentos rápidos e pouco precisos apresentarão

provavelmente funcionais diferentes de movimentos lentos e delicados. Dentre os vários tipos de vínculos, alguns deverão ser obrigatoriamente respeitados para reproduzir a tarefa, enquanto outros serão menos relevantes. Alguns autores (Khang e Zajac, 1989; He et al., 1991; Kuo, 1995; Menegaldo, 1997) utilizaram, ao invés da metodologia de controle ótimo para o caso não-linear, a otimização estática associada a técnicas de controle linear com índice de performance quadrático; para cada aplicação, uma ou outra técnica poderá ser a mais indicada (Zajac e Gordon, 1989): no caso de uma implementação artificial do controle, para Estimulação Elétrica Neuro Muscular em portadores de lesão medular por exemplo, o tempo de processamento para o cálculo do sinal de controle é crítico, ao passo que na reprodução numérica de movimentos naturais não. Diversas questões poderiam ainda ser levantadas, relativas por exemplo à influência do número de graus de liberdade do modelo dinâmico no desempenho do sistema de controle (Kuo, 1996) e ao método de controle ótimo empregado.

As questões apontadas acima não serão respondidas provavelmente a curto prazo, assim como a prescrição de tratamentos e a formulação de testes clínicos com o auxílio de modelos matemáticos. Todas convergem, porém, para uma comum necessidade: a disponibilidade de bons modelos biomecânicos e de formulações adequadas de sistemas de controle, necessários também para a avaliação e validação experimental dos próprios modelos biomecânicos.

O Estado da Arte

Entre os poucos autores que se dedicaram ao estudo sistemático da aplicação de técnicas de controle ótimo a sistemas musculo-esqueléticos, estão o Prof. Marcus Pandy e seus colaboradores, da Universidade do Texas em Austin. Os trabalhos que têm sido

produzidos por esse grupo situam-se, no juízo do autor, no atual estado da arte da abordagem aqui empregada. A maior parte dos trabalhos publicados são referentes ao problema do salto vertical, ainda que movimentos não balísticos de postura e marcha também tenham sido estudados (Pandy et al., 1995; Anderson e Pandy, 1996).

Os primeiros trabalhos desse grupo (Pandy et al., 1990; Pandy e Zajac, 1991), fruto da tese de Doutorado de Pandy na Universidade de Stanford, no grupo de Felix Zajac, tratam de um modelo plano de 4 segmentos e 8 músculos, numa versão modificada do modelo de Zajac (1989) contendo um elemento elástico em série e sem elementos de dissipação viscosa. Em princípio, os braços de momento são variáveis em função dos ângulos articulares, mas as expressões empregadas e sua determinação não estão explicitadas nos artigos consultados. Foi empregado um método de controle ótimo tipo *bang-bang* baseado numa versão modificada do algoritmo de Polak e Mayne (1975), com controles assumindo valores 0 ou 1, sendo determinados os tempos de chaveamento de cada músculo. Um aspecto interessante desse trabalho, estendido aos demais, é a utilização de um método de otimização estática para determinar os valores iniciais dos estados, de maneira que o modelo se encontre em equilíbrio no início do processo de integração. Pandy e Zajac (1991) trazem comparações das simulações com estudos experimentais do problema do salto, tecendo considerações sobre os padrões de acionamento muscular observados.

A partir de 1992 (Pandy et al., 1992) até os trabalhos atuais, o problema de controle ótimo passou a ser tratado como um problema de otimização de parâmetros. Em linhas gerais, as equações diferenciais do sistema dinâmico são integradas empregando-se uma malha de 600 a 1000 pontos para tempos de simulação inferiores 0.5s., sobre a qual é estabelecida uma segunda malha bem mais esparsa de nós de controle, com até 16

elementos. Nos pontos intermediários da primeira malha, os valores das variáveis de controle são estimadas por interpolação linear. O problema de otimização é resolvido com o método de programação seqüencial quadrática proposto por Powell (1978). Uma particularidade da abordagem é a maneira como as matrizes jacobianas das funções de custo e vínculos são calculadas: ao redor de um determinado ponto de operação, é introduzida uma pequena perturbação em cada variável de controle, integrando-se o sistema com essa nova condição. Com esse resultado, as derivadas são então estimadas por diferenças finitas e o processo prossegue iterativamente, até que a variação da função de custo seja pequena o suficiente. Tendo em conta o grande número de variáveis envolvidas e a complexidade das equações diferenciais, o custo computacional dessa metodologia é bastante elevado, sendo necessário o emprego de supercomputadores, ainda que o algoritmo seja facilmente paralelizável (Anderson et al., 1996). Por outro lado, o pequeno número relativo de nós de controle fornece uma estimativa pouco refinada, do ponto de vista da resolução temporal, dos padrões de acionamento muscular obtidos.

Em 1995, Pandy et al. (1995) controlaram o movimento não balístico de levantar-se de uma posição semi-agachada e sentada, empregando o mesmo modelo biomecânico. Os autores mostraram a conveniência do emprego, nesses problemas, da função de custo que minimiza as derivadas das forças musculares. Foi mostrada a semelhança dessa função com a minimização do *jerk*, isto é, da derivada da aceleração, proposta inicialmente por Stein et al. (1996).

Nos trabalhos posteriores (Anderson et al., 1996; Anderson e Pandy, 1998; Anderson e Pandy, 1999), foram feitos estudos introduzindo um modelo biomecânico extremamente sofisticado: tridimensional com 23 graus de liberdade e 54 músculos. Esse

modelo foi empregado tanto na simulação do salto vertical quanto da fase de apoio da marcha.

Escopo do presente trabalho

Este trabalho, utilizando diversos subsídios de trabalhos anteriores (Menegaldo, 1997), visa:

1. Atingir o Estado da Arte na formulação de modelos matemáticos do sistema músculo-esquelético do membro inferior, segundo seus aspectos geométricos e da dinâmica muscular.
2. Utilização de métodos de controle ótimo baseados em teorias matemáticas formalmente bem fundamentadas e numericamente tratáveis na geração de sinais de excitação neural para tarefas posturais, levando o sistema de condições iniciais dadas até condições finais requeridas respeitando os vínculos necessários.

A partir do modelo de múltiplos corpos rígidos da postura humana deduzido em Menegaldo (1997) e do modelo da mecânica muscular estudado no mesmo trabalho, procurou-se obter um modelo biomecânico mais refinado, utilizando um *software* de modelagem do sistema músculo-esquelético e um modelo geométrico detalhado do membro inferior humano de domínio público. Foi elaborada, com base nesse modelo, uma formulação computacionalmente conveniente da geometria, a partir de curvas de regressão. Além disso, o modelo da mecânica muscular desenvolvido anteriormente sofreu algumas modificações, tendo em vista a solução de problemas numéricos na integração das equações diferenciais e a adequação do modelo a testes experimentais com animais (Menegaldo et al., 1999). No Capítulo 2 está descrito em pormenores o modelo

geométrico do membro inferior estudado. Procurou-se apresentá-lo de maneira que ficassem facilmente distinguíveis todos os seus elementos e as hipóteses que foram adotadas na sua elaboração. Conhecendo desta maneira o modelo, assim como as fontes bibliográficas para a determinação de parâmetros do sistema músculo-esquelético, a tarefa de apresentar propostas metodológicas para a obtenção de parâmetros de sujeitos particulares *in vivo*, essencial em futuras aplicações clínicas do modelo matemático, pode ser facilitada. Neste mesmo Capítulo, foi implementado um método para o ajuste de curvas multidimensionais, que permitem expressar os braços de momento e o comprimento dos atuadores músculo-tendíneos através de expressões polinomiais, dependentes das coordenadas generalizadas do modelo.

O problema de controle, por sua vez, foi abordado utilizando algoritmos de discretização e solução do problema de programação não-linear associado, desenvolvidos por Elijah Polak e seus colaboradores. Este tratamento do problema de controle ótimo, desenvolvido formalmente dentro da Teoria das Aproximações Consistentes proposta por esse autor, possui diversos atrativos que justificaram seu emprego no presente trabalho. Entre eles, a possibilidade de tratar problemas relativamente grandes, altamente não-lineares, com diversos tipos de vínculo; além disso, o intenso formalismo matemático com base na teoria de análise em espaços de Hilbert, garantem a consistência analítica do método. O método de discretização, a partir de uma integração tipo Runge-Kutta da dinâmica do sistema é adequada à sua não-linearidade, e os métodos de programação não-linear, ou otimização, estão entre os mais robustos e confiáveis dentre os disponíveis na extensa literatura a respeito. Os autores da Teoria desenvolveram também um *software* compatível com ambiente Matlab chamado RIOTS (*Recursive Integration Optimal Trajectory Solver*), capaz de tratar problemas de controle ótimo com relativa eficiência e

confiabilidade. No Capítulo 4, está formulado o problema de controle ótimo segundo os princípios da Teoria das Aproximações Consistentes. Alguns princípios gerais considerados relevantes para sua compreensão estão também expostos e, na medida do possível, explicados de maneira mais acessível a leitores não familiarizados com essa Teoria, assim como alguns conceitos de análise matemática. Foram omitidos, entretanto, aspectos demasiadamente formais e demonstrações, que o leitor poderá encontrar na bibliografia citada.

São mostrados em seguida resultados preliminares da utilização do método e do *software* de controle em modelos com diversos e decrescentes tipos de simplificações, tendo em vista a solução gradual dos problemas numéricos encontrados. Estes resultados mostram curvas de trajetória dos estados (deslocamentos e velocidades angulares) e das variáveis de controle (ativações neuro-musculares) de um sujeito partindo de uma posição inicial de flexão do joelho até a postura ereta. A partir de um modelo de pêndulo simples controlado por atuadores de torque, foi desenvolvida progressivamente uma metodologia para formulação e solução dos sistemas de controle ótimo usando modelos de crescente complexidade: pêndulo triplo com atuadores de torque sem dinâmica, dinâmicos lineares, atuadores musculares lineares e, finalmente, atuadores musculares não-lineares. Algumas funções de custo foram testadas, assim como selecionados os parâmetros de discretização, integração e representação das trajetórias de controle para cada modelo. Foram discutidas e implementadas soluções para diversos problemas numéricos que se apresentaram, principalmente aqueles relacionados com a escolha da estimativa inicial do vetor de controles e com o tempo de simulação. No capítulo seguinte, são mostrados os resultados finais obtidos com o modelo contendo músculos não-lineares e braços de momento variáveis

Uma característica desejável seria a de que o sistema de controle oferecesse robustez diante de perturbações aleatórias que ocorram durante o movimento. O autor supunha que essa propriedade poderia ser obtida, não fossem alguns problemas numéricos discutidos à frente, formulando-se um problema de Controle Ótimo Linear em malha fechada com Critério de Desempenho Quadrático, associado ao problema original não-linear. Entretanto, não foi possível solucionar o problema formulado desta maneira. Não obstante, foi incluído um capítulo mostrando as tentativas adotadas neste trabalho e discutindo a sua ineficiência.

Os resultados obtidos e discutidos nas Conclusões demonstram a viabilidade da utilização da técnica de controle ótimo empregada para os modelos biomecânicos propostos, assim como as diretrizes que podem ser sugeridas na implementação de problemas da mesma natureza.

Capítulo 2

Modelo geométrico do membro inferior

Neste capítulo está descrito o modelo anatômico do membro inferior a ser utilizado no presente trabalho. Consiste num modelo de domínio público (ver Figura 2.1) desenvolvido por Scott Delp, da Northwestern University, Chicago, EUA¹ (Delp et al., 1990). Este modelo, da maneira como é apresentado, pode ser utilizado num *software* de modelagem geométrica do sistema músculo-esquelético desenvolvido pelo próprio autor do modelo e colaboradores, conhecido como SIMM – *Software for Interactive Musculoskeletal Modelling*² (Delp e Loan, 1995). Nos próximos itens será apresentada

¹ O modelo está disponível na *home page* da ISB – International Society of Biomechanics, no endereço <http://isb.ri.ccf.org/data/delp/>, tendo sido desenvolvido na Veterans Affairs R&D Center, Palo Alto, CA. Esta mesma página traz uma lista de artigos em que o modelo foi empregado para diversas aplicações.

² Comercializado pela empresa Musculographics Inc., Evanston, IL (<http://www.musculographics.com/>)

uma revisão das principais hipóteses assumidas na formulação deste modelo, assim como dos trabalhos precedentes que forneceram os parâmetros por ele utilizado.

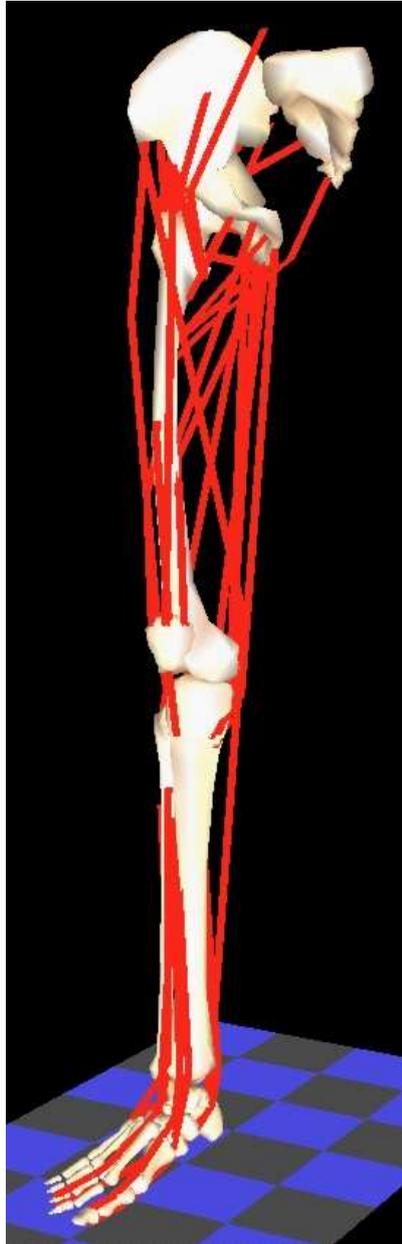


Figura 2.1: Modelo geométrico do membro inferior

Entretanto, a maneira como o *software* é utilizado normalmente não é adequada para a simulação de modelos integrados da biomecânica do sistema músculo-esquelético³, controlados ainda por um sistema de controle das ativações musculares. Assim, a partir de informações geométricas obtidas através do SIMM, foram ajustadas curvas de interpolação polinomial, que fornecem valores dos braços de momento e do comprimento de atuadores músculo-tendíneos, em função de coordenadas generalizadas. Com isso, a inclusão dessas grandezas numa rotina de simulação e controle do sistema músculo-esquelético pode ser feita de maneira computacionalmente econômica.

2.1 Revisão bibliográfica

As unidades básicas de definição do modelo são suas articulações. Dada uma articulação, definem-se inicialmente as suas características cinemáticas, isto é, o movimento em seis graus de liberdade (três rotações e três translações) que realiza em função da variação das coordenadas generalizadas que configuram o seu estado. Isto é feito a partir de *funções cinemáticas*, configurando o movimento, numa dada direção, de um corpo rígido em relação a outro. Por exemplo, sabe-se que na articulação do joelho, a tíbia apresenta uma translação na direção proximal com a extensão, juntamente com uma rotação externa. Entretanto, esses movimentos estão vinculados diretamente ao próprio ângulo de flexão, sendo este uma coordenada generalizada. Um sumário das principais características dos modelos de articulação utilizadas para quadril, joelho e tornozelo está mostrado na Tabela 2.1.

³ Existe um produto comercializado pela Musculographics Inc. chamado *Dynamics Pipeline*, que gera equações de movimento através de um *software* de manipulação simbólica conhecido como SD/FAST, utilizando também suas rotinas de integração. Entretanto, o custo de tais produtos é bastante elevado e sua versatilidade limitada, não sendo possível, a princípio, a utilização das rotinas de controle ótimo empregadas neste trabalho.

Articulação	Características
Quadril	Junta esférica com três graus de liberdade (Delp et al., 1990)
Joelho	Junta plana de um grau de liberdade, baseada em Yamaguchi e Zajac (1989). Leva em conta o ponto de contato variável da articulação tíbio-femoral, a cinemática patelo-femoral e o efeito do aumento do braço de momento do tendão do quadríceps (Delp et al., 1990)
Tornozelo, subtalar e metatarso-falangeal	Modeladas como juntas tipo pino (com um grau de liberdade), com eixos localizados e orientados segundo Inman (1976) com pequenas alterações na orientação da articulação metatarso-falangeal (Delp et al., 1990)

Tabela 2.1: Características dos modelos de articulação utilizadas no modelo do membro inferior.

As funções cinemáticas são definidas a partir de uma tabela de valores de translação ou de ângulo de rotação, em função de uma coordenada generalizada. As funções cinemáticas estão na Tabela 2.2. Entretanto, a definição total do conjunto de transformações lineares de um sistema de coordenadas para outro deve levar em conta todas as translações e rotações⁴ que ocorrem entre os segmentos. Estas, por sua vez, podem ser funções cinemáticas ou simplesmente valores constantes. Na Tabela 2.3 estão mostradas as translações e rotações entre os referenciais, cujas localizações e orientações estão mostradas na Figura 2.2. As rotações entre um segmento e outro ocorrem ao redor de eixos também definidos. A Tabela 2.4 traz as coordenadas dos eixos de rotação entre o primeiro e o segundo segmento de cada articulação.

⁴ Todas as unidades de comprimento do modelo estão expressas em metros e os ângulos em graus. Esta convenção é adotada ao longo de todo capítulo, a menos que se indique o contrário.

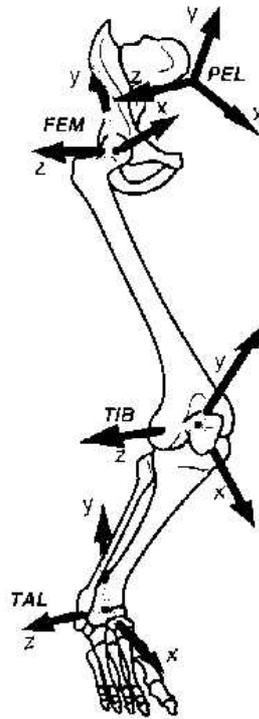


Figura 2.2: Localização dos sistemas de referência da pélvis, fêmur, tíbia e tornozelo (Musculographics Inc., 1997)

Função	Tabela de valores	Função interpolada
f1	-360.0, -360.0 360.0, 360.0	ângulo de flexão do quadril x rot. eixo z
f2	-360.0, -360.0 360.0, 360.0	ângulo de adução do quadril x rot. eixo x
f3	-360.0, -360.0 360.0, 360.0	ângulo de rotação do quadril x rot. eixo y
f4	-360.0, -360.0 360.0, 360.0	ângulo de flexão do joelho x rot. eixo z
f5	-120.0, -0.00320 -100.0, 0.00179 -80.0, 0.00411 -60.0, 0.00410 -40.0, 0.00212 -20.0, -0.00100 -10.0, -0.00310 0.0, -0.00525	deslocamento dx da tíbia x ângulo de flexão do joelho

<p>f6</p>	<p>-120.0, -0.4226 -70.0, -0.4082 -30.0, -0.3990 -20.0, -0.3976 -10.0, -0.3966 0.0, -0.3960</p>	<p>deslocamento dy da tibia x ângulo de flexão do joelho</p>
<p>f7</p>	<p>-120.0, 0.0173 -80.0, 0.0324 -60.0, 0.0381 -40.0, 0.0430 -20.0, 0.0469 -10.0, 0.0484 0.0, 0.0496</p>	<p>deslocamento dx do referencial tibial em relação ao patelar x ângulo de flexão do joelho</p>
<p>f8</p>	<p>-120.0, -0.0219 -90.0, -0.0202</p>	<p>deslocamento dy do referencial tibial em relação ao</p>

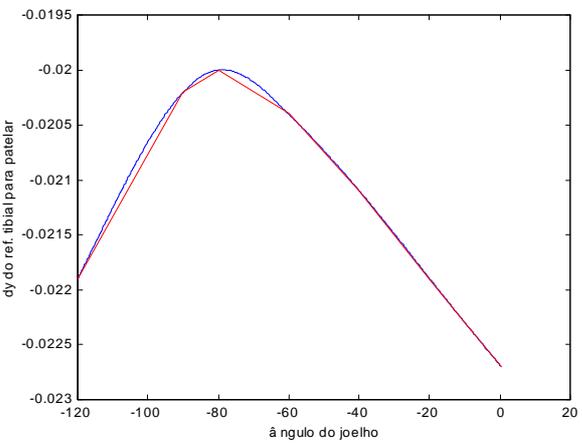
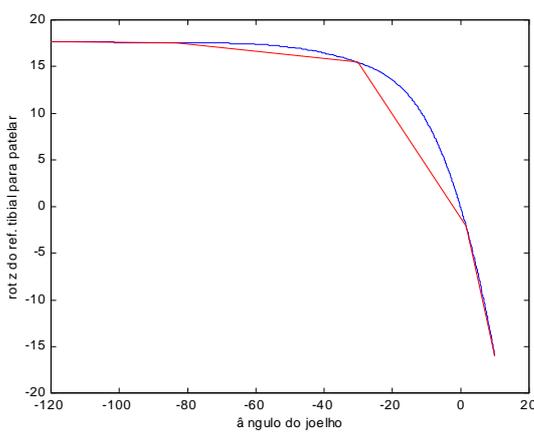
	-80.0, -0.0200 -60.0, -0.0204 -40.0, -0.0211 -20.0, -0.0219 -10.0, -0.0223 0.286,-0.0227	patelar x ângulo de flexão do joelho 
f9	-120.00, 17.65 -114.59, 17.65 -83.51, 17.55 -30.16, 15.48 1.60, -2.12 10.00, -16.04	rotação do referencial tibial em relação ao patelar x ângulo de flexão do joelho 
f10	-360.0, -360.0 360.0, 360.0	ângulo do tornozelo x rot. eixo z
f11	-360.0, -360.0 360.0, 360.0	ângulo de inversão (subtalar) x rot eixo x
f12	-360.0, -360.0 360.0, 360.0	ângulo dos dedos (metatarso-falangeal) x rot. eixo x
f13		progressão da perna

Tabela 2.2: Funções cinemáticas. Os valores mostrados na 2ª coluna (mostrados por segmentos de reta) são interpolados por uma spline cúbica (curva suave) pelo SIMM. As amplitudes admissíveis de movimento estão mostradas na Tabela 2.7.

Articulação	Seg.1	seg.2	ordem	tx	ty	tz	r1	r2	r3
quadril	Pelvis	femur	t r3 r1 r2	-.0707	-.0661	.0835	f1	f2	f3
femoral-tibial	Femur	tibia	t r3 r1 r2	f5	f6	0	f4	0	0
tíbio-patelar	Tíbia	patelar	t r3 r1 r2	f7	f8	0.0024	f9	0	0
tornozelo	Tíbia	talus	t r3 r1 r2	0	-0.43	0	f10	0	0
subtalar	Talus	calcaneus	t r1 r2 r3	-.04877	-.04195	.00792	f11	0	0
dedos	Calcaneus	dedos	t r1 r2 r3	0.1788	-0.0020	0.00108	0	0	0
ref. inercial	Ref. inercial	pelvis	t r1 r2 r3	f13	0	0	0	0	0

Tabela 2.3: Transformações entre os segmentos.

Articulação	eixo 1	eixo 2	eixo 3
Quadril	1.0 0.0 0.0	0.0 1.0 0.0	0.0 0.0 1.0
femoral-tibial	1.0 0.0 0.0	0.0 1.0 0.0	0.0 0.0 1.0
tíbio-patelar	1.0 0.0 0.0	0.0 1.0 0.0	0.0 0.0 1.0
Tornozelo	1.0 0.0 0.0	0.0 1.0 0.0	-0.10453 -0.17365 0.97925
Subtalar	0.781 0.60 -0.12	0.0 1.0 0.0	0.0 0.0 1.0
Metatarso-falangeal	0.5808 0.0 -0.8140	0.0 1.0 0.0	0.0 0.0 1.0
ref. Inercial	1.0 0.0 0.0	0.0 1.0 0.0	0.0 0.0 1.0

Tabela 2.4: Coordenadas dos eixos de rotação entre o primeiro e o segundo segmento de cada articulação, definidos na Tabela 2.3.

Uma importante grandeza geométrica a ser determinada é o comprimento dos atuadores músculo-tendíneos (L^{MT}) em função das coordenadas generalizadas. Dispondo-se dessas relações, é possível realizar as simulações do sistema músculo-esquelético e de controle sem considerar necessariamente a hipótese de que as contrações musculares sejam isométricas. O cálculo de L^{MT} para cada atuador é feito a partir da distância entre sua origem e inserção tendo em conta, porém, os desvios que podem ocorrer quando o atuador se dobra sobre estruturas ósseas. Nestes casos, os chamados *wrapping points*, ou

pontos de contorno, são incluídos na definição da trajetória do músculo; entretanto, tais pontos podem ou não estar ativos, dependendo do valor da coordenada generalizada. Por exemplo, para o músculo *rectus femoris*, considera-se que existe um ponto de contorno na porção proximal do fêmur quando o ângulo de flexão do joelho está entre -120.0° e -83.65° (-120° corresponde a extensão máxima no modelo geométrico utilizado).

As coordenadas das origens e inserções musculares foram determinadas por Brand et al. (1982), a partir de três cadáveres (feminino 1,63 m, masculino 1,72 m e masculino 1,83 m). As coordenadas foram determinadas com o auxílio de marcadores opacos posicionados nas regiões de inserção e origem muscular, através de Raio-X. Os resultados apresentados pelos autores correspondem às médias dos três espécimes, utilizando sistemas de coordenadas sujeitos a fatores de escala dependentes de dimensões anatômicas (para a definição dessas coordenadas e fatores, ver por exemplo Menegaldo, 1997, Cap. 5). Na Tabela 2.5 estão mostradas as coordenadas das inserções e origens para os 43 músculos que compõe o modelo. Nesta mesma tabela estão relacionados os nomes abreviados dos músculos que serão utilizados ao longo do trabalho.

Nome	Nº	Abreviação	Coordenadas				segmento	Observações
			x	y	z			
<i>gluteus medius</i>	1	gmed1	-0.0408	0.0304	0.1209	pelvis	compartimento anterior	
			-0.0218	-0.0117	0.0555	femur		
<i>gluteus medius</i>	2	gmed2	-0.0855	0.04450	0.0766	pelvis	compartimento médio	
			-0.0258	-0.0058	0.0527	femur		
<i>gluteus medius</i>	3	gmed3	-0.1223	0.0105	0.0648	pelvis	compartimento posterior	
			-0.0309	-0.0047	0.0518	femur		
<i>gluteus minimus</i>	4	gmin1	-0.0467	-0.0080	0.1056	pelvis	compartimento anterior	
			-0.0072	-0.0104	0.0560	femur		
<i>gluteus minimus</i>	5	gmin2	-0.0633	-0.0065	0.0991	pelvis	compartimento médio	
			-0.0096	-0.0104	0.0560	femur		
<i>gluteus minimus</i>	6	gmin3	-0.0834	-0.0063	0.0856	pelvis	compartimento posterior	
			-0.0135	-0.0083	0.0550	femur		
<i>Semimembranosus</i>	7	semimem	-0.1192	-0.1015	0.0695	pelvis		
			-0.0243	-0.0536	-0.0194	tíbia		
<i>Semitendinosus</i>	8	semiten	-0.1237	-0.1043	0.0603	pelvis		
			-0.0314	-0.0545	-0.0146	tíbia		
			-0.0113	-0.0746	-0.0245	tíbia		
			0.0027	-0.0956	-0.0193	tíbia		
<i>biceps femoris</i>	9	bifemlh	-0.1244	-0.1001	0.0666	pelvis	cabeça longa	
			-0.0081	-0.0729	0.0423	tíbia		
<i>biceps femoris</i>	10	bifemsh	0.0050	-0.2111	0.0234	femur	cabeça curta	
			-0.0101	-0.0725	0.0406	tíbia		
<i>Sartorius</i>	11	sar	-0.0153	-0.0013	0.1242	pelvis		
			-0.0030	-0.3568	-0.0421	femur		
			-0.0056	-0.0419	-0.0399	tíbia		
			0.0060	-0.0589	-0.0383	tíbia		
			0.0243	-0.0840	-0.0252	tíbia		
<i>adductor longus</i>	12	addlong	-0.0316	-0.0836	0.0169	pelvis		
			0.0050	-0.2111	0.0234	femur		
<i>adductor brevis</i>	13	addbrev	-0.0587	-0.0915	0.0164	pelvis		
			0.0009	-0.1196	0.0294	femur		
<i>adductor magnus</i>	14	amag1	-0.0732	-0.1174	0.0255	pelvis	compartimento superior	
			-0.0045	-0.1211	0.0339	femur		
<i>adductor magnus</i>	15	amag2	-0.0831	-0.1192	0.0308	pelvis	compartimento médio	
			0.0054	-0.2285	0.0227	femur		
<i>adductor magnus</i>	16	amag3	-0.0771	-0.1181	0.0276	pelvis	compartimento inferior	
			0.0070	-0.3837	-0.0266	femur		
<i>tensor fasciae latae</i>	17	tfl	-0.0311	0.0214	0.1241	pelvis		
			0.0294	-0.0995	0.0597	femur		
			0.0054	-0.4049	0.0357	femur		
			0.0060	-0.0487	0.0297	tíbia		
<i>Pectineus</i>	18	pect	-0.0431	-0.0768	0.0451	pelvis		
			-0.0122	-0.0822	0.0253	femur		
<i>Gracilis</i>	19	gra	-0.0563	-0.1038	0.0079	pelvis		
			-0.0154	-0.0475	-0.0358	tíbia		
			0.0060	-0.0836	-0.0228	tíbia		
<i>gluteus maximus</i>	20	gmax1	-0.1195	0.0612	0.0700	pelvis	compartimento superior	
			-0.1291	0.0012	0.0886	pelvis		
			-0.0457	-0.0248	0.0392	femur		
			-0.0277	-0.0566	0.0470	femur		

<i>gluteus maximus</i>	21	gmax2	-0.1349 0.0176 0.0563 - 0.1376 -0.0520 0.0914 -0.0426 -0.0530 0.0293 -0.0156 -0.1016 0.0419	pelvis pelvis femur femur	compartimento médio
<i>gluteus maximus</i>	22	gmax3	-0.1556 -0.0314 0.0058 -0.1529 -0.1052 0.0403 -0.0299 -0.1041 0.0135 -0.0060 -0.1419 0.0411	pelvis pelvis femur femur	compartimento inferior intelecta tuberosidade isquiática acima de 60° de flexão do quadril
<i>Iliacus</i>	23	iliacus	-0.0674 0.0365 0.0854 -0.0218 -0.0550 0.0851 0.0017 -0.0543 0.0057 -0.0193 -0.0621 0.0129	pelvis pelvis femur femur	
<i>Psoas</i>	24	psoas	-0.0647 0.0887 0.0289 -0.0238 -0.0570 0.0759 0.0016 -0.0507 0.0038 -0.0188 -0.0597 0.0104	pelvis pelvis femur femur	
<i>quadratus femoris</i>	25	quadfem	-0.1143 -0.1151 0.0520 -0.0381 -0.0359 0.0366	pelvis femur	
<i>Gemelli</i>	26	gem	-0.1133 -0.0820 0.0714 -0.0142 -0.0033 0.0443	pelvis femur	
<i>Periformis</i>	27	peri	-0.1396 0.0003 0.0235 -0.1193 -0.0276 0.0657 -0.0148 -0.0036 0.0437	pelvis pelvis femur	
<i>rectus femoris</i>	28	rf	-0.0295 -0.0311 0.0968 0.0334 -0.4030 0.0019 0.0121 0.0437 -0.0010	pelvis femur patela	ativa de -120° a -83.65° de flexão do joelho
<i>vastus medialis</i>	28	vasmed	0.0140 -0.2099 0.0188 0.0356 -0.2769 0.0009 0.0370 -0.4048 -0.0125 0.0274 -0.4255 -0.0131 0.0063 0.0445 -0.0170	femur femur femur femur patela	ativa de -120° a -69.33° ativa de -120° a -101.99° de flexão do joelho
<i>vastus intermedius</i>	29	vasint	0.0290 -0.1924 0.0310 0.0335 -0.2084 0.0285 0.0343 -0.4030 0.0055 0.0058 0.0480 -0.0006	femur femur femur patela	ativa de -120° a -81.36° de flexão do joelho
<i>vastus lateralis</i>	30	vaslat	0.0048 -0.1854 0.0349 0.0269 -0.2591 0.0409 0.0361 -0.4030 0.0205 0.0253 -0.4243 0.0184 0.0103 0.0423 0.0141	femur femur femur femur patela	ativa de -120° a -69.33° ativa de -120° a -110.01°
<i>Gastrocnemius</i>	31	medgas	-0.0127 -0.3929 -0.0235 -0.0239 -0.4022 -0.0258 -0.0217 -0.0487 -0.0295 0.0044 0.0310 -0.0053	femur femur tibia calcaneus	cabeça medial ativa de -44.12° a 5.73° de flexão do joelho
<i>Gastrocnemius</i>	32	latgas	-0.0155 -0.3946 0.0272 -0.0254 -0.4018 0.0274 -0.0242 -0.0481 0.0235 0.0044 0.0310 -0.0053	femur femur tibia calcaneus	cabeça lateral ativa de -44.12° a 5.73° de flexão do joelho
<i>Soleus</i>	33	sol	-0.0024 -0.1533 0.0071 0.0044 0.0310 -0.0053	tibia calcaneus	
<i>tibialis posterior</i>	34	tibpost	-0.0094 -0.1348 0.0019 -0.0144 -0.4051 -0.0229 0.0417 0.0334 -0.0286 0.0772 0.0159 -0.0281	tibia tibia calcaneus calcaneus	

<i>flexor digitorum longus</i>	35	flexdig	-0.0083 -0.2046 -0.0018 -0.0154 -0.4051 -0.0196 0.0436 0.0315 -0.0280 0.0708 0.0176 -0.0263 0.1658 -0.0081 0.0116 -0.0019 -0.0078 0.0147 0.0285 -0.0071 0.0215 0.0441 -0.0060 0.0242	tibia tibia calcaneus calcaneus calcaneus dedos dedos dedos	
<i>flexor hallucis longus</i>	36	flexhal	-0.0079 -0.2334 0.0244 -0.0186 -0.4079 -0.0174 0.0374 0.0276 -0.0241 0.1038 0.0068 -0.0256 0.1726 -0.0053 -0.0269 0.0155 -0.0064 -0.0265 0.0562 -0.0102 -0.0181	tibia tibia calcaneus calcaneus calcaneus dedos dedos	
<i>tibialis anterior</i>	37	tibant	0.0179 -0.1624 0.0115 0.0329 -0.3951 -0.0177 0.1166 0.0178 -0.0305	tibia tibia calcaneus	
<i>peroneus brevis</i>	38	perbrev	-0.0070 -0.2646 0.0325 -0.0198 -0.4184 0.0283 -0.0144 -0.4295 0.0289 0.0471 0.0270 0.0233 0.0677 0.0219 0.0343	tibia tibia tibia calcaneus calcaneus	
<i>peroneus longus</i>	39	perlong	0.0005 -0.1568 0.0362 -0.0207 -0.4205 0.0286 -0.0162 -0.4319 0.0289 0.0438 0.0230 0.0221 0.0681 0.0106 0.0284 0.0852 0.0069 0.0118 0.1203 0.0085 -0.0184	tibia tibia tibia calcaneus calcaneus calcaneus calcaneus	
<i>peroneus tertius</i>	40	pertert	0.0010 -0.2804 0.0231 0.0229 -0.4069 0.0159 0.0857 0.0228 0.0299	tibia tibia calcaneus	
<i>extensor digitorum longus</i>	41	extdig	0.0032 -0.1381 0.0276 0.0289 -0.4007 0.0072 0.0922 0.0388 -0.0001 0.1616 0.0055 0.0130 0.0003 0.0047 0.0153 0.0443 -0.0004 0.0250	tibia tibia calcaneus calcaneus dedos dedos	
<i>extensor hallucis longus</i>	42	exthal	0.0012 -0.1767 0.0228 0.0326 -0.3985 -0.0085 0.0970 0.0389 -0.0211 0.1293 0.0309 -0.0257 0.1734 0.0139 -0.0280 0.0298 0.0041 -0.0245 0.0563 0.0034 -0.0186	tibia tibia calcaneus calcaneus calcaneus dedos dedos	
<i>ligamento patelar</i>	43	ligpat	0.0390 -0.0822 0.0000 0.0021 0.0015 0.0001	tibia patela	

Tabela 2.5: Músculos utilizados, suas abreviaturas e coordenadas de origem e inserção. Duas coordenadas eventualmente definidas no mesmo segmento não alteram o L^{MT} com a variação das coordenadas generalizadas; configuram apenas o caminho (*path*) do atuador músculo-tendíneo, necessário entretanto para o cálculo correto de seu comprimento total.

Os sistemas de referência estão localizados segundo a Tabela 2.6 e a Figura 2.2 ; as coordenadas generalizadas são relacionadas na Tabela 2.7.

Sistema de Referência	Localização
Pelvis	Ponto médio da linha que conecta as duas ASIS (espinhas ilíacas ântero-superiores)
Fêmur	Centro da cabeça do fêmur
Tíbia	Ponto médio entre os dois epicôndilos femorais, com joelho na posição anatômica
Patela	Ponto mais distal do ápice da patela
Tálus	Ponto médio entre os maléolos lateral e medial
Calcâneo	Ponto mais distal (e inferior) da superfície posterior do calcâneo
Dedos	Base do 2º metatarso

Tabela 2.6: Localização dos sistemas de referência. Para todos os segmentos, o eixo x aponta anteriormente, y superiormente e z lateralmente.

Nome da coordenada generalizada	Código	mínimo (graus)	máximo (graus)
flexão do quadril	HF	-10 extensão	95 flexão
adução do quadril	HA	-50 abdução	15 adução
rotação do quadril	HR	-20 rot. interna	20 rot. externa
ângulo do joelho	KA	-120 flexão	0 (extensão)
ângulo do tornozelo	AA	-30 flexão plantar	30 dorsiflexão
ângulo subtalar	SA	-20 inversão	20 eversão
ângulo metatarso-falangeal	MA	-30 flexão	30 extensão

Tabela 2.7: Coordenadas generalizadas e sua amplitude admissível de movimento.

Dado um músculo cuja linha de ação passa através dessa articulação, existe uma distância mínima entre a sua linha de ação e o centro instantâneo de velocidade nula da articulação no caso plano⁵. Esta distância é conhecida como braço de momento (r) do músculo em relação à articulação (Menegaldo, 1997, Cap. 5). A maneira apresentada por

⁵ Considerando o movimento no espaço, o lugar geométrico em relação ao qual a velocidade de translação é nula é sempre uma reta, conhecida como *eixo instantâneo de rotação*.

Delp et al.(1990) para cálculo do braço de momento⁶ considera que uma reta de tamanho r , sofrendo um deslocamento angular infinitesimal $d\theta$ em torno de uma de suas extremidades descreverá, na outra extremidade, um arco de comprimento $rd\theta$. Assim, considerando uma mudança de comprimento dL^{MT} do atuador músculo-tendíneo, pode-se dizer que

$$rd\theta = dL^{MT} \quad \text{eq. 2.1}$$

ou seja, o braço de momento do músculo i em relação à coordenada generalizada θ_j é calculada como

$$r_{ij} = \frac{dL_i^{MT}}{d\theta_j} \quad \text{eq. 2.2}$$

Utilizando a equação acima com uma aproximação por diferenças finitas o cálculo do braço de momento é feito de maneira simples, se L^{MT} tiver sido previamente calculado⁷.

A mecânica muscular é tratada pelo SIMM a partir de uma versão estática do modelo adimensional tipo Hill de Zajac (Zajac, 1989). Com isso, foi realizada pelos autores do modelo uma coletânea de parâmetros necessários para definir cada músculo, isto é, força máxima (F_0^M), comprimento ótimo da fibra muscular (L_0^M) e ângulo de empenamento (α). Nesta etapa do presente trabalho, quando se pretende apenas o cálculo de r e L^{MT} , tais parâmetros não são utilizados. Entretanto, o modelo da dinâmica da contração muscular que será utilizado adiante os necessita, além de outros mais. Estes parâmetros foram obtidos a partir de três trabalhos, que serão descritos sucintamente a seguir. As referências para a obtenção de cada um dos parâmetros dos músculos estudados estão mostradas na Tabela 2.8.

⁶ Uma outra maneira para o cálculo de r é apresentada em Delp e Loan (1995), utilizando a dinâmica de Kane (Kane e Levinson, 1985). Esse método é adequado na situação em que as funções cinemáticas são conhecidas através de expressões analíticas.

No trabalho apresentado por Wickiewicz et al. (1983) foram utilizadas três pernas de cadáveres seccionadas em hemipelvectomy e fixadas em formalina. As articulações do quadril e do joelho foram mantidas em extensão máxima, enquanto o tornozelo em flexão plantar máxima. Cada músculo foi dissecado fora do membro, sendo limpo de gordura e tecido conjuntivo, e colocado em banho de formalina a 10% por dois a três dias. Foram lavados posteriormente numa solução tampão de fosfato de sódio e colocados em solução a 15% de ácido sulfúrico para amolecer e retirar restos de tecido conjuntivo. Depois, os músculos eram limpos em solução tampão de fosfato e armazenados numa solução a 50% de glicerol.

O ângulo de empenamento era medido com um transferidor, segundo Gans et al. (1965). Feixes de 10 a 20 fibras musculares foram dissecados de várias regiões do músculo e medidos os comprimentos de cada feixe com precisão de 1 mm. Calculava-se área da seção transversal do músculo (PCSA), em cm^2 , como

$$\text{PCSA} = \frac{\text{massa}[\text{g}] \cos(\alpha)}{\text{comprimento da fibra}[\text{cm}] \rho_{\text{músculo}}[\text{g}/\text{cm}^3]}$$

sendo a densidade do músculo $\rho_{\text{músculo}} = 1,056 \text{ g}/\text{cm}^3$. Brand et al. (1986) e Friederich e Brand (1990) utilizaram dois cadáveres embalsamados em posição anatômica, um do sexo masculino de 1,83 m de altura, 91 kg, 37 anos e outro do sexo feminino, 1,63 m, 59 Kg, 63 anos. Com uma régua, foram medidos os comprimentos dos músculos, desde o centróide da origem ao da inserção, e os ângulos de empenamento com um goniômetro. O comprimento das fibras musculares foi medido para cada músculo segundo o método reportado por Spector et al. (1980), deixando-os em solução salina de 1 a 5 dias e posteriormente em ácido nítrico, de 24h a 48h, com o objetivo de macerar o tecido. Cada

⁷ O cálculo da diferença de L^{LM} nesta equação pode ser feito considerando apenas a variação da distância

músculo foi dissecado em vários feixes e, de cada feixe 10 a 20 fibras, eram separadas num microscópio de dissecação. Com isso, medindo-se o comprimento de cada fibra foi calculado o comprimento médio das fibras de cada músculo. A área da seção transversal foi calculada a partir do volume do músculo, medido através de deslocamento de líquido, dividido pelo comprimento médio das fibras.

Tomando o momento máximo gerado experimentalmente pelo grupo extensor do joelho como referência (Spector et al., 1980), foi utilizado um modelo estático (Hoy et al., 1990) no qual a área da seção transversal dos músculos era multiplicada por um fator de escala. Quando o momento gerado na simulação foi igual ao determinado experimentalmente, variando esse fator, seu valor ficava associado à tensão muscular máxima. Devido às diferenças de preparação, e conseqüentemente de encolhimento dos espécimens, foram ajustados tensões musculares máximas de 73 N/cm^2 por Wickiewicz et al. (1983) e 30 N/cm^2 por Brand et al. (1986) e Friederich e Brand (1990). O comprimento ótimo das fibras musculares foi estimado por Hoy et al. (1990) por um fator (2.8/2.2), correspondente à razão do comprimento ótimo do sarcômero⁸, de acordo com a teoria microscópica da mecânica muscular (Gordon et. al, 1966), e o utilizado por Wickiewicz.

Além desses parâmetros, foi necessário ainda determinar o comprimento do tendão relaxado⁹ (L^{ST}), o que foi feito segundo dois critérios (Delp et al., 1990). No primeiro, utilizando o SIMM, considerava-se o comprimento do atuador músculo-tendíneo correspondente ao ângulo da articulação no qual, experimentalmente, pode ser verificado o

entre duas coordenadas que são contíguas e estão em segmentos diferentes (ver Tabela 2.5) .

⁸ Wickiewicz et al., (1983) utilizam no seu trabalho o comprimento ótimo do sarcômero de $2,2 \mu\text{m}$, resultado obtido por Gordon et. al. (1966) para o músculo do sapo aplicando a teoria microscópica dos filamentos deslizantes da contração muscular. Entretanto, se forem utilizados os valores encontrados em humanos dos comprimentos dos miofilamentos grossos e finos, a teoria mesma teoria fornece o valor de $2,8 \mu\text{m}$ para o comprimento ótimo do sarcômero (Hoy et al., 1990).

⁹ Mais precisamente, comprimento acima do qual o atuador músculo-tendíneo começa a exercer algum momento passivo na articulação.

início da geração do momento passivo. Tal comprimento, subtraído do comprimento ótimo (supostamente coincidente com o comprimento acima do qual o músculo passa a exercer força passiva), fornecia valores de L^{ST} . No segundo critério, variava-se os valores de L^{ST} até que o ângulo referente ao pico de momento ativo medido no SIMM coincidissem com valores experimentais, tomando assim este valor de L^{ST} .

Músculo	origem dos parâmetros			valores utilizados dos parâmetros			
	F_0^M	L_0^M	α	F_0^M	L_0^M	α	L^{ST}
gmed1	2	3	3	546.0	0.0535	8.0	0.0780
gmed2	2	3	3	382.0	0.0845	0.0	0.0530
gmed3	2	3	3	435.0	0.0643	19.0	0.0530
gmin1	2	3	3	180.0	0.0680	10.0	0.0160
gmin2	2	3	3	190.0	0.0560	0.0	0.0260
gmin3	2	3	3	215.0	0.0380	1.0	0.0510
semimem	1	1	1	1030.0	0.0800	15.0	0.3590
semiten	1	1	1	328.0	0.2010	5.0	0.2620
bifemlh	1	1	1	717.0	0.1090	0.0	0.3410
bifemsh	1	1	1	402.0	0.1730	23.0	0.1000
sar	1	1	1	104.0	0.5790	0.0	0.0400
addlong	2	3	3	418.0	0.1380	6.0	0.1100
addbrev	2	3	3	286.0	0.1330	0.0	0.0200
amag1	2	3	3	346.0	0.0870	5.0	0.0600
amag2	2	3	3	312.0	0.1210	3.0	0.1300
amag3	2	3	3	444.0	0.1310	5.0	0.2600
tfl	1	1	1	155.0	0.0950	3.0	0.4250
pect	2	3	3	117.0	0.1330	0.0	0.0010
gra	1	1	1	108.0	0.3520	3.0	0.1400
gmax1	2	3	3	382.0	0.1420	5.0	0.1250
gmax2	2	3	3	546.0	0.1470	0.0	0.1270
gmax3	2	3	3	368.0	0.1440	5.0	0.1450
iliacus	2	3	3	429.0	0.1000	7.0	0.0900
psoas	2	3	3	371.0	0.1040	8.0	0.1300
quadfem	2	3	3	254.0	0.0540	0.0	0.0240
gem	2	3	3	109.0	0.0240	0.0	0.0390
peri	2	3	3	296.0	0.0260	10.0	0.1150
rf	1	1	1	779.0	0.0840	5.0	0.3460
vasmed	1	1	1	1294.0	0.0890	5.0	0.1260
vasint	1	1	1	1365.0	0.0870	3.0	0.1360
vaslat	1	1	1	1871.0	0.0840	5.0	0.1570
medgas	2	1	1	1113.0	0.0450	17.0	0.4080
latgas	2	1	1	488.0	0.0640	8.0	0.3850
sol	1	2	2	2839.0	0.0300	25.0	0.2860
tibpost	1	1	1	1270.0	0.0310	12.0	0.3100
flexdig	1	1	1	310.0	0.0340	7.0	0.4000
flexhal	1	1	1	322.0	0.0430	10.0	0.3800
tibant	1	1	1	603.0	0.0980	5.0	0.2230
perbrev	1	1	1	348.0	0.0500	5.0	0.1610
perlong	1	1	1	754.0	0.0490	10.0	0.3450
pertert	2	3	3	90.0	0.0790	13.0	0.1000
extdig	1	1	1	341.0	0.1020	8.0	0.3450
exthal	1	1	1	108.0	0.1110	6.0	0.3050

Tabela 2.8: Parâmetros musculares e trabalhos que foram utilizados para escolha dos parâmetros (1): Wickiewicz et al., 1993, (2): Brand et al., 1986, (3):Friederich e Brand, 1990.

2.2 Descrição do ambiente SIMM

Uma vez definido conceitualmente o modelo do membro inferior e determinados os parâmetros necessários, resta descrevê-lo através de uma linguagem computacional adequada para sua visualização e análise. Isso pode ser feito com grande eficiência pelo SIMM, cujos módulos de entrada de informações sobre o modelo e seus parâmetros estão descritos sumariamente a seguir.

2.2.1 Arquivos de ossos (*bone files*)

Como um auxílio à visualização dos modelos, o SIMM utiliza arquivos de ossos digitalizados. Cada osso é parametrizado a partir de uma lista de coordenadas de vértices de polígonos, estabelecendo as conexões entre os vértices, cujas coordenadas podem ser obtidas através de um sistema de digitalização tridimensional, por exemplo o Polhemus (Delp et al., 1990). A geração dos *bone files* propriamente ditos é feita através de uma rotina específica incorporada ao SIMM chamada `norm` (Musculographics Inc., 1997, Cap. 5).

2.2.2 Arquivos de músculos (*muscle files*)

Nos arquivos de músculos, define-se inicialmente as propriedades invariantes dos músculos, isto é, as relações de força x comprimento, força x velocidade, nas porções ativa e passiva. As relações fornecidas pelo modelo geométrico do SIMM não foram empregadas. Na dinâmica da contração muscular, foram utilizadas as relações já utilizadas em Menegaldo (1997).

Definição das propriedades do músculo padrão

```

beginmuscle defaultmuscle

begintendonforcelengthcurve
/* (tendon strain, normalized force) */
(-10.000,0.0000)
(-0.0020,0.0000)
.
.
(9.20200,345.0)
(20.0000,345.0)
endtendonforcelengthcurve

beginactiveforcelengthcurve
/* (norm length, norm force) */
(-5.00000,0.000000)
(0.000000,0.000000)
(0.401000,0.000000)
.
.
(2.200000,0.000000)
(5.000000,0.000000)
endactiveforcelengthcurve

beginpassiveforcelengthcurve
/* (norm length, norm force) */
(-5.00000,0.000000)
.
(5.000000,2.000000)
endpassiveforcelengthcurve

beginforcevelocitycurve
/* velocity, normalized force */
(-1.000000,0.000000)
(-0.950000,0.010417)
.
(0.950000,1.773155)
(1.000000,1.774455)
endforcevelocitycurve

max_contraction_velocity 10.0 /* fiberlengths/second */
endmuscle

```

As propriedades de cada músculo são então inseridas definindo o seu nome, o caminho, o grupo funcional a que pertence, F_0^M , L_0^M , L^{ST} e α . Na existência de pontos de contorno na definição do caminho do músculo, que podem ou não estar ativos em função de coordenadas generalizadas, utiliza-se o comando `ranges`, seguido do intervalo em que está ativo.

Exemplo: *vastus medialis*

```

beginmuscle vasmed /* vastus medialis */
beginpoints
0.0140 -0.2099 0.0188 segment femur
0.0356 -0.2769 0.0009 segment femur
0.0370 -0.4048 -0.0125 segment femur ranges 1 knee_angle (-150.0, -69.33)
0.0274 -0.4255 -0.0131 segment femur ranges 1 knee_angle (-150.0, -101.99)

```

```

0.0063 0.0445 -0.0170 segment patella
endpoints
beginngroups knee_ext endgroups
max_force 1294.0 /* source: Wickiewicz */
optimal_fiber_length 0.0890 /* source: Wickiewicz */
tendon_slack_length 0.1260 /* source: Delp */
pennation_angle 5.0 /* source: Wickiewicz */
endmuscle

```

2.2.3 Arquivos de articulação (*joint files*)

Os arquivos de articulação possuem a seguinte estrutura básica:

1. Definição de objetos globais, isto é, da cor e textura dos ossos e do chão, assim como do nome do modelo.
2. Definição de cada articulação, isto é, seu nome, os segmentos que a compõe, a ordem de transformações lineares que reproduz a sua cinemática, a orientação dos sistemas de referência e as funções cinemáticas de translação e rotação.

Exemplo: articulação do quadril

```

beginjoint hip
segments pelvis femur /* defines joint between pelv and femur */
order t r3 r1 r2 /* translation, flexion, adduction, rotation */
axis1 1.0 0.0 0.0
axis2 0.0 1.0 0.0
axis3 0.0 0.0 1.0
tx constant -.0707 /* from midpt. between ASIS to hip ctr. */
ty constant -.0661
tz constant .0835
r3 function f1(hip_flexion)
r1 function f2(hip_adduction)
r2 function f3(hip_rotation)
endjoint

```

3. Definição das coordenadas generalizadas e sua amplitude.

Exemplo:

```

beginngcoord hip_flexion /* hip flexion/extension */
range -10.0 95.0
endngcoord

```

4. Lista dos arquivos de ossos que são utilizados em cada segmento.

Exemplo:

```

beginsegment pelvis
beginfiles
pelvis.bin
sacrum.bin
endfiles
endsegment

```

5. Definição das funções cinemáticas

Exemplo: Translação tx da tíbia em relação ao fêmur, em função do ângulo de flexão

(função f5)

```

/* tx for femoral-tibial joint */
beginfunction f5 /* range of motion: 120 deg.flexion to 0 deg.ext */
/* knee angle, x-translation from femural to tibial frame */
(-120.0, -0.00320)
(-100.0, 0.00179)
( -80.0, 0.00411)
( -60.0, 0.00410)
( -40.0, 0.00212)
( -20.0, -0.00100)
( -10.0, -0.00310)
(  0.0, -0.00525)
endfunction

```

2.3 Determinação das equações de regressão para o comprimento e braços de momento dos atuadores músculo-tendíneos

Para cada músculo definido no *joint file* foram determinadas curvas de regressão múltipla de valores do comprimento dos atuadores e de seus braços de momento em relação às coordenadas generalizadas. Os músculos analisados dependiam destas coordenadas segundo a Tabela 2.9:

Músculo	HF	HA	HR	KA	AA	SA	MA
gmed1	f	ab	i				
gmed2		ab					
gmed3	e	ab	e				
gmin1	f	ab	i				
gmin2		ab					
gmin3	e	ab	e				
semimem	e	ad		f			
semiten	e	ad		f			
bifemlh	e	ad		f			
bifemsh				f			
sar	f	ab		f			
addlong	e	ad	f				
addbrev	e	ad					
amag1	e	ad					
amag2	e	ad					
amag3	e	ad					
tfl	f	ab	i				
pect	f	ad					
gra	f	ad		f			
gmax1	e	ab					
gmax2	e						
gmax3	e						
iliacus	f		i				
psoas	f		i				
quadfem			e				
gem			e				
peri		ab	e				
rf	f			e			
vasmed				e			
vasint				e			
vaslat				e			
medgas				f	fp		
latgas				f	fp		
sol					fp		
tibpost					fp	i	
flexdig					fp	i	f
flexhal					fp	i	f
tibant					df	i	
perbrev					fp	e	
perlong					fp	e	
pertert					df	e	
extdig					df	e	e
exthal					df	i	e
ligpat				e			

Tabela 2.9: Cada coluna significa um grau de liberdade, sendo: HF=flexão do quadril (e=extensão, f=flexão); HA=adução do quadril (ad=adução, ab=abdução); HR=rotação do quadril (i=interna, e=externa) KA=ângulo do joelho (e=extensão, f=flexão); SA=ângulo subtalar (i=inversão, e=eversão); MA= ângulo metatarso-falangeal.(f=flexão, e=extensão).

Foi implementada um rotina em Matlab[®] responsável pela geração de arquivos *musculo.in*, compatíveis com o formato de arquivos do *plot maker*¹⁰ do SIMM. Esta rotina possui a seguinte estrutura:

1ª parte: geração de vetores de coordenadas generalizadas.

Dada a amplitude de movimento para cada coordenada generalizada, tal intervalo angular é dividido em vinte partes iguais.

```
clear all
% geração de arquivo de entrada para o simm
% cálculo de r1, r2, r3 e Lmt para músculos especificados no arquivo
muscles_delp.in
dis=20; %número de pontos da malha de geração das coordenadas generalizadas

%hip_flexion          /* hip flexion/extension */
min=-10.0;
max=95.0;
HF=min:(max-min)/(dis-1):max;

%hip_adduction        /* hip ab/adduction */
min=-50.0;
max=15.0;
HA=min:(max-min)/(dis-1):max;

%hip_rotation         /* hip internal/external rotation */
min=-20.0;
max=20.0;
HR=min:(max-min)/(dis-1):max;

%knee_angle           /* knee flexion */
min=-120.0;
max=5.0;
KA=min:(max-min)/(dis-1):max;

%ankle_angle          /* ankle motion */
min=-30.0;
max=30.0;
AA=min:(max-min)/(dis-1):max;

%subt_angle           /* subtalar motion */
min=-20.0;
max=20.0;
SA=min:(max-min)/(dis-1):max;

%mtp_angle            /* toe motion */
```

¹⁰ Rotina responsável pelo cálculo de r e L^{MT} .

```
min=-30.0;
max=30.0;
MA=min:(max-min)/(dis-1):max;
```

2ª parte: geração de um vetor de pontos de entrada, cuja primeira coluna é o número do ponto e a segunda o valor da coordenada generalizada (exemplo da 2ª porção do músculo *gluteus medius*)

```
%gmed2
clear gmed2
for i=1:dis,
    gmed2=[gmed2;HA(i)];
end
c1=gmed2(:,1);
c0=1:length(c1);
minr=c0(1);
maxr=c0(length(c1));
```

3ª parte: geração do arquivo musculo.in

```
hdl=char('#include "simmkeys.h"');
musc=char('gmed2');
datacolumns=size(gmed2,2)+1;
datarows=size(gmed2,1)
fid = fopen('gmed2.in','w');
fprintf(fid,'%10s\n',hdl);
fprintf(fid,'%s\n',' ');
fprintf(fid,'%s %s\n','name',musc);
fprintf(fid,'%s %i\n','datacolumns',datacolumns);
fprintf(fid,'%s %i\n','datarows',datarows);
fprintf(fid,'%s %i %i\n','range',minr,maxr');
fprintf(fid,'%s\n','wrap');
fprintf(fid,'%s\n','endheader');
fprintf(fid,'%s\n',' ');
fprintf(fid,'%s\t %s\t\n','leg_tx','hip_adduction');
fprintf(fid,'%s\n',' ');
for i=1:datarows,
    fprintf(fid,'%i\t %f\t\n',c0(i),c1(i));
end
fclose(fid);
```

Arquivo gerado:

```
#include "simmkeys.h"

name gmed2
datacolumns 2
datarows 20
range 1 20
wrap
endheader

leg_tx hip_adduction
```

```

1      -50.000000
2      -46.578947
3      -43.157895
4      -39.736842
5      -36.315789
6      -32.894737
7      -29.473684
8      -26.052632
9      -22.631579
10     -19.210526
11     -15.789474
12     -12.368421
13     -8.947368
14     -5.526316
15     -2.105263
16     1.315789
17     4.736842
18     8.157895
19     11.578947
20     15.000000

```

No caso de um músculo cujo comprimento e braços de momento dependem de mais coordenadas generalizadas (2 ou 3), os arquivos de entrada são gerados de maneira que, mantendo o valor inicial de duas coordenadas generalizadas, varia-se a terceira dentro da amplitude de movimento correspondente. Posteriormente, toma-se o segundo valor da 2ª coordenada generalizada, variando a terceira; mais adiante, a 1ª coordenada é também alterada pelo seu segundo valor. Procedendo de maneira sucessiva, são gerados 400 (2 graus de liberdade) ou 8000 (três graus de liberdade) pontos de conjuntos de valores das coordenadas generalizadas, percorrendo toda a amplitude de movimento.

Exemplo: 1ª porção do músculo *gluteus medius* (3 coordenadas generalizadas)

```

%gmed1
clear gmed1
for i=1:dis,
    for j=1:dis,
        for k=1:dis,
            gmed1=[gmed1;HF(i) HA(j) HR(k)];
        end
    end
end
end
c1=gmed1(:,1);
c2=gmed1(:,2);
c3=gmed1(:,3);
c0=1:length(c1);
minr=c0(1);

```


26	-10.000000	-46.578947	-9.473684
27	-10.000000	-46.578947	-7.368421
28	-10.000000	-46.578947	-5.263158
29	-10.000000	-46.578947	-3.157895
30	-10.000000	-46.578947	-1.052632
31	-10.000000	-46.578947	1.052632
32	-10.000000	-46.578947	3.157895
33	-10.000000	-46.578947	5.263158
34	-10.000000	-46.578947	7.368421
35	-10.000000	-46.578947	9.473684
36	-10.000000	-46.578947	11.578947
37	-10.000000	-46.578947	13.684211
38	-10.000000	-46.578947	15.789474
39	-10.000000	-46.578947	17.894737
40	-10.000000	-46.578947	20.000000
41	-10.000000	-43.157895	-20.000000
42	-10.000000	-43.157895	-17.894737
43	-10.000000	-43.157895	-15.789474
44	-10.000000	-43.157895	-13.684211
45	-10.000000	-43.157895	-11.578947
46	-10.000000	-43.157895	-9.473684
47	-10.000000	-43.157895	-7.368421
48	-10.000000	-43.157895	-5.263158
49	-10.000000	-43.157895	-3.157895
50	-10.000000	-43.157895	-1.052632
51	-10.000000	-43.157895	1.052632
52	-10.000000	-43.157895	3.157895
53	-10.000000	-43.157895	5.263158
54	-10.000000	-43.157895	7.368421
55	-10.000000	-43.157895	9.473684
56	-10.000000	-43.157895	11.578947
57	-10.000000	-43.157895	13.684211
58	-10.000000	-43.157895	15.789474
59	-10.000000	-43.157895	17.894737
60	-10.000000	-43.157895	20.000000
.	.	.	.
.	.	.	.
.	.	.	.
7990	95.000000	15.000000	-1.052632
7991	95.000000	15.000000	1.052632
7992	95.000000	15.000000	3.157895
7993	95.000000	15.000000	5.263158
7994	95.000000	15.000000	7.368421
7995	95.000000	15.000000	9.473684
7996	95.000000	15.000000	11.578947
7997	95.000000	15.000000	13.684211
7998	95.000000	15.000000	15.789474
7999	95.000000	15.000000	17.894737
8000	95.000000	15.000000	20.000000

Os arquivos gerados neste formato podem então ser importados pelo SIMM. O comprimento de cada atuador músculo-tendíneo e os braços de momento em relação às

coordenadas generalizadas são então calculados segundo as hipóteses estabelecidas para o modelo adotado.

Em função dos graus de liberdade que determinam a geometria de cada músculo, foram definidos 14 grupos (Tabela 2.10). Para cada grupo são gerados arquivos de saída contendo uma seqüência de valores de L_{mt} , r_1 , r_2 e r_3 (ou apenas r_1 ou r_1 e r_2). Uma vez calculados esses parâmetros, foi utilizado o Método dos Mínimos Quadrados (por exemplo, Press et al., 1992) para ajustar parâmetros de curvas de regressão propostas para cada uma das grandezas geométricas calculadas, em função das coordenadas generalizadas.

Grupos	Nº de coord. gen.	Músculos
1	3	gmed1, gmed3, gmin1, gmin3, addlong, tfl
2	1	gmed2, gmin2
3	3	semimem, semiten, bifemlh, sar, gra
4	1	bifemsh, vasint, vaslat, ligpat
5	2	adbrev, amag1, pect, gmax1
6	1	gmax2, gmax3
7	2	iliacus, psoas
8	1	quadfem, gem
9	2	peri
10	2	rf
11	2	medgas, latgas
12	1	sol
13	2	tibpost, tibiant, perbrev, perlong, pertert
14	3	flexdig, flexhal, extdig, exthal

Tabela 2.10: Grupos musculares selecionados com critério de possuírem parâmetros dependentes das mesmas coordenadas generalizadas.

O método de ajuste consiste em, dada uma seqüência de pontos (x_i, y_i) , ajustar os coeficientes a_k da equação

$$y(x) = \sum_{i=1}^m a_k X_k(x) \quad \text{eq. 2.3}$$

sendo $X_k(x)$ funções base não lineares.

Seja uma função de mérito

$$\chi^2 = \sum_{i=1}^n \left[\frac{y_i - \sum_{k=1}^m a_k X_k(x_i)}{\sigma_i} \right]^2 \quad \text{eq. 2.4}$$

tal que n é o número de medidas e σ_i é o desvio padrão de cada medida x_i . Normalmente, $n \gg m$.

A *matriz de definição* A ($n \times m$) do problema de ajuste de curvas pode ser construída a partir das m funções base X_k calculadas em n amostras x_i , e dos n σ_i .

$$A = \begin{bmatrix} \frac{X_1(x_1)}{\sigma_1} & \frac{X_2(x_1)}{\sigma_1} & \dots & \frac{X_m(x_1)}{\sigma_1} \\ \frac{X_1(x_2)}{\sigma_2} & \frac{X_2(x_2)}{\sigma_2} & \dots & \frac{X_m(x_2)}{\sigma_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{X_1(x_n)}{\sigma_n} & \frac{X_2(x_n)}{\sigma_n} & \dots & \frac{X_m(x_n)}{\sigma_n} \end{bmatrix} \quad \text{eq. 2.5}$$

Sejam ainda os vetores \mathbf{a} de m parâmetros a serem ajustados e $\mathbf{b} = \{y_i/\sigma_i\}$ de dimensão m . Uma condição necessária para mínimo da função de mérito é que

$$\frac{d\chi^2}{da_k} = 0 \quad \text{eq. 2.6}$$

É possível mostrar (Press et al., 1992) que esta condição fornece a seguinte equação matricial

$$(\mathbf{A}^T \mathbf{A}) \mathbf{a} = \mathbf{A}^T \mathbf{b} \quad \text{eq. 2.7}$$

Cuja solução pode ser escrita como

$$\mathbf{a} = (\mathbf{A}^T \mathbf{A})^+ \mathbf{A}^T \mathbf{b} \quad \text{eq. 2.8}$$

sendo $(\cdot)^+$ a matriz pseudo-inversa.

O método pode ser aplicado para o problema do ajuste das equações de regressão de L_{mt} , r_1 , r_2 e r_3 em função das coordenadas generalizadas. Supondo $\sigma_i = 1$ e tomando como exemplo um músculo do Grupo 1, pode-se propor uma equação do tipo:

$$L_{mt,r_1,r_2,r_3}(HF, HA, HR) = a_1 + a_2 HF + a_3 HA + a_4 HR + a_5 HF^2 + a_6 HA^2 + a_7 HR^2 \quad \text{eq. 2.9}$$

Lembrando que HF, HA HR são respectivamente os ângulos de flexão, adução e rotação do quadril, define-se a matriz A como

$$A = \begin{bmatrix} 1 & HF(1) & HA(1) & HR(1) & HF(1)^2 & HA(1)^2 & HR(1)^2 \\ 1 & HF(2) & HA(2) & HR(2) & HF(2)^2 & HA(2)^2 & HR(2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & HF(8000) & HA(8000) & HR(8000) & HF(8000)^2 & HA(8000)^2 & HR(8000)^2 \end{bmatrix} \quad \text{eq. 2.10}$$

e os vetores \mathbf{a} de coeficientes a determinar e \mathbf{b} de valores de L_{MT}

$$\mathbf{a} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7] \quad \text{eq. 2.11}$$

$$\mathbf{b} = [L_{mt}(1) \quad L_{mt}(2) \quad \dots \quad L_{mt}(8000)] \quad \text{eq. 2.12}$$

é possível calcular os valores de \mathbf{a} utilizando a equação normal dos mínimos quadrados

$$\mathbf{a} = (\mathbf{A}^T \mathbf{A})^+ \mathbf{A}^T \mathbf{b}.$$

Foi implementado um programa em linguagem MATLAB para o cálculo dos parâmetros a_i , assim como do erro médio e do desvio padrão da média com relação às curvas originais geradas pelo SIMM.

Programa regrasc1.m:

```
clear all
clear global
dis=20; %número de pontos da malha de geração das coordenadas generalizadas
%hip_flexion      /* hip flexion/extension */
min=-10.0;
max=95.0;
HF=min:(max-min)/(dis-1):max;
%hip_adduction   /* hip ab/adduction */
min=-50.0;
```

```

max=15.0;
HA=min:(max-min)/(dis-1):max;
%hip_rotation      /* hip internal/external rotation */
min=-20.0;
max=20.0;
HR=min:(max-min)/(dis-1):max;
%knee_angle        /* knee flexion */
min=-120.0;
max=5.0;
KA=min:(max-min)/(dis-1):max;
%ankle_angle       /* ankle motion */
min=-30.0;
max=30.0;
AA=min:(max-min)/(dis-1):max;
%subt_angle        /* subtalar motion */
min=-20.0;
    max=20.0;
SA=min:(max-min)/(dis-1):max;
%mtp_angle         /* toe motion */
min=-30.0;
max=30.0;
MA=min:(max-min)/(dis-1):max;

%grupo 1
load gl.asc;
Lmt=gl(:,2:7);
Lmtgl=zeros(size(Lmt,2),dis,dis,dis);
%cria matriz 3d do Lmt(HF,HA,HR) e recria vetor de entrada do arquivo .in, que
será utilizado na interpolação
for h=1:size(Lmt,2),
    for i=1:dis,
        for j=1:dis,
            for k=1:dis,
                Lmtgl(i,j,k,h)=Lmt( (i-1)*dis*dis + (j-1)*dis + k , h );
                glr=[glr;HF(i) HA(j) HR(k)];
            end
        end
    end
end
c1=glr(:,1);
c2=glr(:,2);
c3=glr(:,3);

% Algoritmo de ajuste de curvas
% função 1 para Lmt,
Lmt(HF,HA,HR)=a1+a2*HF+a3*HA+a4*HR+a5*HF^2+a6*HA^2+a7*HR^2+a8*HF^3+a9*HA^3+a10*H
R^3
% função 2 para Lmt,
Lmt(HF,HA,HR)=a1+a2*HF+a3*HA+a4*HR+a5*HF*HA+a6*HF*HR+a7*HA*HR
% função 3 para Lmt, Lmt(HF,HA,HR)=a1+a2*HF+a3*HA+a4*HR
% função 4 para Lmt,
Lmt(HF,HA,HR)=a1+a2*HF+a3*HA+a4*HR+a5*HF^2+a6*HA^2+a7*HR^2+a8*HF*HA*HR

%determinação da matriz A e do vetor b
b1=zeros(Lmt,size(Lmt,2));
b2=zeros(Lmt,size(Lmt,2));
b3=zeros(Lmt,size(Lmt,2));
b4=zeros(Lmt,size(Lmt,2));
for h=1:size(Lmt,2),
    for i=1:length(Lmt),
        A1(i,:)= [1 c1(i) c2(i) c3(i) c1(i)^2 c2(i)^2 c3(i)^2 c1(i)^3 c2(i)^3
c3(i)^3];
        b1(i,h)=[Lmt(i,h)];
        A2(i,:)= [1 c1(i) c2(i) c3(i) c1(i)*c2(i) c1(i)*c3(i) c2(i)*c3(i)];
        b2(i,h)=[Lmt(i,h)];
        A3(i,:)= [1 c1(i) c2(i) c3(i)];
    end
end

```

```

        b3(i,h)=[Lmt(i,h)];
        A4(i,:)= [1 c1(i) c2(i) c3(i) c1(i)^2 c2(i)^2 c3(i)^2 c1(i)*c2(i)*c3(i)];
        b4(i,h)=[Lmt(i,h)];
    end
end
%cálculo do vetor a de parâmetros ajustados por mínimos quadrados e decomposição
SVD
for h=1:size(Lmt,2),
    a1(:,h)=pinv(A1'*A1)*A1'*b1(:,h);
    a2(:,h)=pinv(A2'*A2)*A2'*b2(:,h);
    a3(:,h)=pinv(A3'*A3)*A3'*b3(:,h);
    a4(:,h)=pinv(A4'*A4)*A4'*b4(:,h);
end

%verificação da função de ajuste
erro=zeros(4,size(Lmt,2));
for h=1:size(Lmt,2),
    for i=1:dis,
        for j=1:dis,
            for k=1:dis,
                Lmtglr1(i,j,k,h)=a1(1,h)+a1(2,h)*HF(i)+a1(3,h)*HA(j)+a1(4,h)*HR(k)+a1(
                    5,h)*HF(i)^2+a1(6,h)*HA(j)^2+a1(7,h)*HR(k)^2+a1(8,h)*HF(i)^3+a1(9,h)*H
                    A(j)^3+a1(10,h)*HR(k)^3;
                Lmtglr2(i,j,k,h)=a2(1,h)+a2(2,h)*HF(i)+a2(3,h)*HA(j)+a2(4,h)*HR(k)+a2(
                    5,h)*HF(i)*HA(j)+a2(6,h)*HF(i)*HR(k)+a2(7,h)*HA(j)*HR(k);
                Lmtglr3(i,j,k,h)=a3(1,h)+a3(2,h)*HF(i)+a3(3,h)*HA(j)+a3(4,h)*HR(k);
                Lmtglr4(i,j,k,h)=a4(1,h)+a4(2,h)*HF(i)+a4(3,h)*HA(j)+a4(4,h)*HR(k)*a4(
                    5,h)*HF(i)^2+a4(6,h)*HA(j)^2+a4(7,h)*HR(k)^2+a4(8,h)*HF(i)*HA(j)*HR(k)
                    ;
                erro(1,h)=erro(1,h)+abs(Lmtgl(i,j,k,h)-Lmtglr1(i,j,k,h));
                erro(2,h)=erro(2,h)+abs(Lmtgl(i,j,k,h)-Lmtglr2(i,j,k,h));
                erro(3,h)=erro(3,h)+abs(Lmtgl(i,j,k,h)-Lmtglr3(i,j,k,h));
                erro(4,h)=erro(4,h)+abs(Lmtgl(i,j,k,h)-Lmtglr4(i,j,k,h));
            end
        end
    end
end
erro

for h=1:size(Lmt,2),
    erro(1,h)=erro(1,h)/length(Lmt);
    erro(2,h)=erro(2,h)/length(Lmt);
    erro(3,h)=erro(3,h)/length(Lmt);
    erro(4,h)=erro(4,h)/length(Lmt);
end
erro

desv=zeros(4,size(Lmt,2));
for h=1:size(Lmt,2),
    for i=1:dis,
        for j=1:dis,
            for k=1:dis,
                desv(1,h)=desv(1,h)+(abs(Lmtgl(i,j,k,h)-Lmtglr1(i,j,k,h))-
                    erro(1,h))^2;
                desv(2,h)=desv(2,h)+(abs(Lmtgl(i,j,k,h)-Lmtglr2(i,j,k,h))-
                    erro(2,h))^2;
                desv(3,h)=desv(3,h)+(abs(Lmtgl(i,j,k,h)-Lmtglr3(i,j,k,h))-
                    erro(3,h))^2;
                desv(4,h)=desv(4,h)+(abs(Lmtgl(i,j,k,h)-Lmtglr4(i,j,k,h))-
                    erro(4,h))^2;
            end
        end
    end
end
desv=(desv/(length(Lmt)-1)).^0.5

```

Equações de regressão:

São propostas três estruturas de equações de regressão, para músculos que dependem de uma, duas ou três coordenadas generalizadas (Tabela 2.11). Para as duas primeiras, quatro equações, com diversos graus de complexidade, e portanto de diferentes custos computacionais foram ajustadas. Para terceira estrutura, trabalhou-se com três equações.

Estrutura	Nº de coord. generaliz.	Eq. Nº	Equações propostas
1	3	1	$L_{mt, r_1, r_2, r_3}(Q_1, Q_2, Q_3) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_3 + a_5 Q_1^2 + a_6 Q_2^2 + a_7 Q_3^2 + a_8 Q_1^3 + a_9 Q_2^3 + a_{10} Q_3^3$
		2	$L_{mt, r_1, r_2, r_3}(Q_1, Q_2, Q_3) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_3 + a_5 Q_1 \cdot Q_2 + a_6 Q_1 \cdot Q_3 + a_7 Q_2 \cdot Q_3$
		3	$L_{mt, r_1, r_2, r_3}(Q_1, Q_2, Q_3) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_3$
		4	$L_{mt, r_1, r_2, r_3}(Q_1, Q_2, Q_3) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_3 + a_5 Q_1^2 + a_6 Q_2^2 + a_7 Q_3^2 + a_8 Q_1 \cdot Q_2 \cdot Q_3$
2	2	1	$L_{mt, r_1, r_2}(Q_1, Q_2) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_1^2 + a_5 Q_2^2 + a_6 Q_1^3 + a_7 Q_2^3$
		2	$L_{mt, r_1, r_2}(Q_1, Q_2) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_1 \cdot Q_2$
		3	$L_{mt, r_1, r_2}(Q_1, Q_2) = a_1 + a_2 Q_1 + a_3 Q_2$
		4	$L_{mt, r_1, r_2}(Q_1, Q_2) = a_1 + a_2 Q_1 + a_3 Q_2 + a_4 Q_1^2 + a_5 Q_2^2 + a_6 Q_1 \cdot Q_2$
3	1	1	$L_{mt, r_1}(Q_1) = a_1 + a_2 Q_1 + a_3 Q_1^2 + a_4 Q_1^3$
		2	$L_{mt, r_1}(Q_1) = a_1 + a_2 Q_1 + a_3 Q_1^2$
		3	$L_{mt, r_1}(Q_1) = a_1 + a_2 Q_1$

Tabela 2.11: Equações propostas para ajuste das curvas de L^{MT} e r em função das coordenadas generalizadas.

Uma vez calculados os parâmetros a_i das curvas ajustadas, os erros de ajuste e os desvios padrão, procurou-se selecionar, para cada grandeza de cada músculo qual das curvas de interpolação propostas fornecia os menores erros de ajuste. Os resultados de tal seleção se encontram na Tabela 2.12, ao passo que os parâmetros de ajuste, erros e desvios padrão se encontram no Anexo 2. Ilustrando a forma e o ajuste das curvas de regressão obtidas, na Figura 2.3, Figura 2.4 e Figura 2.5 estão mostrados, para o músculo *rectus femoris*, o comprimento do atuador músculo-tendíneo e os braços de momento em relação à articulação do quadril e do joelho, em função das coordenadas generalizadas.

Músculo	L^{MT}	r_1	r_2	r_3
gmed1	2	2	2	2
gmed2	1	1	x	x
gmed3	2	2	1	2
gmin1	2	2	2	1
gmin2	1	1	x	x
gmin3	2	2	1	1
semimem	2	1	2	1
semiten	2	1	2	1
bifemlh	1	1	2	1
bifemsh	1	1	x	x
sar	1	1	2	1
addlong	2	2	1	1
addbrev	4	4	1	x
amag1	4	4	1	x
amag2	4	4	1	x
amag3	4	4	4	x
tfl	2	2	2	2
pect	4	4	4	x
gra	1	2	1	1
gmax1	4	4	4	x
gmax2	1	1	x	x
gmax3	1	1	x	x
iliacus	1	1	1	x
psoas	1	1	1	x
quadfem	1	1	x	x
gem	1	1	x	x
peri	1	4	4	x
rf	1	4	1	x
vasmed	1	1	x	x
vasint	1	1	x	x
vaslat	1	1	x	x
medgas	1	1	1	x
latgas	1	1	1	x
sol	1	1	x	x
tibpost	4	4	4	x
flexdig	2	2	2	1
flexhal	1	2	2	1
tibant	4	4	4	x
perbrev	4	4	4	x
perlong	4	4	4	x
pertert	4	4	4	x
extdig	1	1	1	1
exthal	1	1	1	1
ligpat	1	1	x	x

Tabela 2.12: Seleção das equações de regressão com mínimo erro de ajuste, conforme Tabela 2.11 e Anexo 1 (x - parâmetro não definido para o músculo)

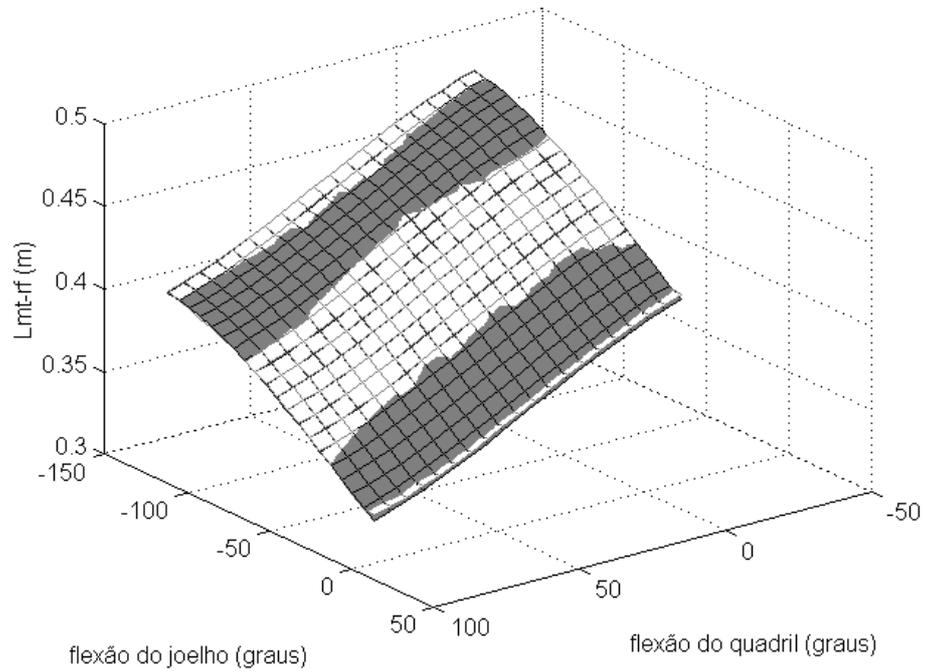


Figura 2.3: Comprimento do *rectus femoris* em função das coordenadas generalizadas. A curva real (branca) e a que se refere ao resultado da interpolação (cinza) estão superpostas.

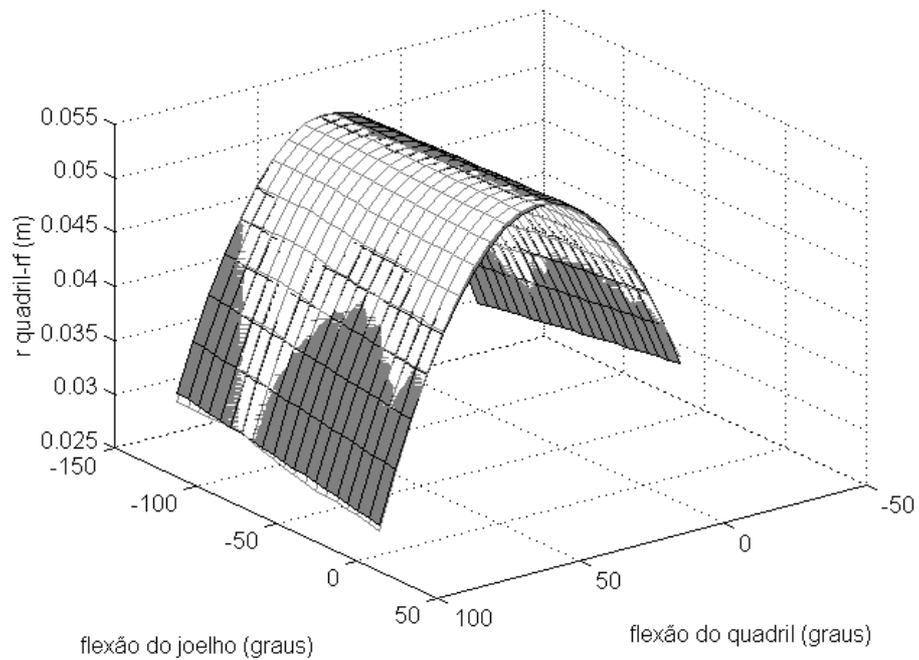


Figura 2.4: Braço de momento do *rectus femoris* em relação ao centro da articulação do quadril em função das coordenadas generalizadas. Curva real :branca; interpolada: cinza.

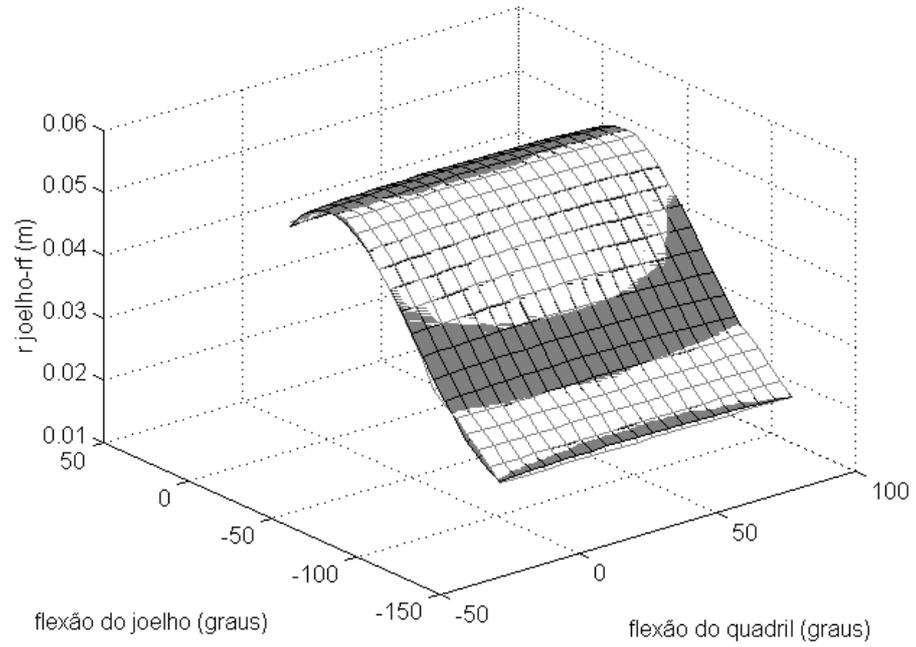


Figura 2.5: Braço de momento do *rectus femoris* em relação ao centro da articulação do joelho em função das coordenadas generalizadas. Curva real :branca; interpolada: cinza.

Capítulo 3

Refinamento e linearização de um modelo da mecânica da contração muscular

Em trabalhos anteriores (Menegaldo, 1997; Menegaldo e Weber, 1998), foi feita a formulação de um modelo tipo Hill, ou viscoelástico, da dinâmica da contração muscular, baseada no modelo de Zajac (1989). Apesar de apresentar diversas vantagens em relação a outros modelos do mesmo tipo, apresenta problemas numéricos na deativação, em faixas de operação do músculo com baixos níveis de ativação e em músculos com tendões longos. Para saná-los, são introduzidos elementos elásticos e viscosos em paralelo aos elementos contráteis (Schutte et al., 1993). Neste capítulo, é apresentada, de maneira sintética, uma formulação aprimorada em relação aos trabalhos anteriores, guardando, porém, os mesmos princípios. Informações mais detalhadas podem ser encontradas nos

trabalhos já citados e em Menegaldo et al. (1999), onde está mostrada também uma validação experimental do modelo com o músculo *lastissimus dorsi* do cão.

Os elementos que compõem o modelo proposto estão representados na Figura 3.1.

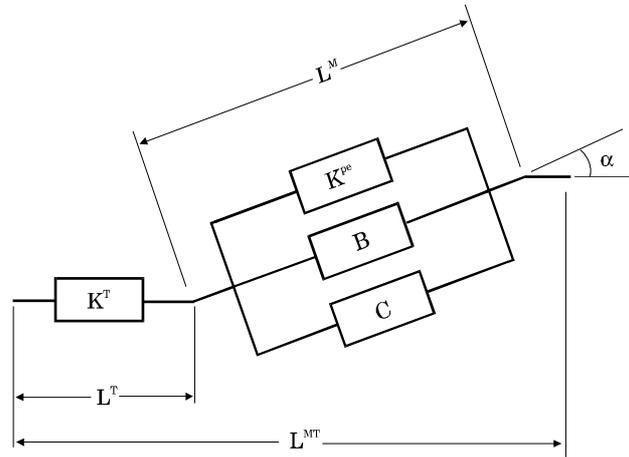


Figura 3.1: Modelo viscoelástico da mecânica da contração muscular

Este modelo obedece a nomenclatura a seguir, sendo os símbolos entre parêntesis, usualmente assinalados com \sim , referentes à grandeza física de força, comprimento ou tempo, divididas pelos parâmetros de adimensionalização: força máxima (ou força no comprimento ótimo) F_0^M , comprimento ótimo L_0^M e parâmetro de escala de tempo τ_c , respectivamente.

$F^M, (\tilde{F}^M)$ = força muscular

$F^{M*}, (\tilde{F}^{M*})$ = força muscular quando o músculo está no comprimento ótimo

$F^{CE}, (\tilde{F}^{CE})$ = força no elemento contrátil

$F^{PE}, (\tilde{F}^{PE})$ = força no elemento passivo

$F^T, (\tilde{F}^T)$ = força no tendão

L^s = comprimento do sarcômero

$L^M, (\tilde{L}^M)$ = comprimento do músculo

$L^T, (\tilde{L}^T)$ = comprimento do tendão

$L^{MT}, (\tilde{L}^{MT})$ = comprimento do atuador músculo-tendíneo

$v^M, (\tilde{v}^M)$ = velocidade da fibra muscular

$v_a^M, (\tilde{v}_a^M)$ = velocidade de alongamento da fibra muscular

- $v_e^M, (\tilde{v}_e^M)$ = velocidade encurtamento da fibra muscular
 $v^T, (\tilde{v}^T)$ = velocidade do tendão
 $v^{MT}, (\tilde{v}^{MT})$ = velocidade do atuador músculo-tendíneo
 $t, (\tau)$ = tempo
 $\sigma^T, (\tilde{\sigma}^T)$ = tensão no tendão
 ε^T = deformação do tendão
 $u(t), u(\tau)$ = excitação muscular
 $a(t), a(\tau)$ = ativação muscular
 α_M = ângulo da fibra muscular
 L_α^M = comprimento das fibras musculares projetado no eixo do tendão
 F_0^M = força isométrica máxima do músculo
 L_0^M = comprimento ótimo das fibras musculares (desenvolvendo a força máxima)
 α_0 = ângulo ótimo da fibra muscular quando $L^M = L_0^M$
 ε_0^T = deformação no tendão quando $F^T = F_0^M$
 σ_0^T = tensão no tendão quando $F^T = F_0^M$
 $L_s^T, (\tilde{L}_s^T)$ = comprimento do tendão relaxado
 $k^{SE}, (\tilde{k}^{SE})$ = rigidez das pontes cruzadas (SEE)
 $k^T, (\tilde{k}^T)$ = rigidez do tendão
 E^T = módulo de elasticidade do tendão
 A^T = área da seção transversal do tendão
 fl_a = componente ativa da relação força-comprimento
 fl_p = componente passiva da relação força-comprimento
 v_m = velocidade máxima de encurtamento das fibras musculares
 τ_c = escala de tempo ($\tau_c \equiv L_0^M / v_M$)
 $\tau_{act}, (\tilde{\tau}_{act})$ = constante de tempo para contração
 $\tau_{deact}, (\tilde{\tau}_{deact})$ = constante de tempo para relaxamento
 $\beta = \tau_{act} / \tau_{deact}$
 $F^{PE}, (\tilde{F}^{PE})$ = Força no elemento elástico em paralelo
 $F^{DE}, (\tilde{F}^{DE})$ = Força no elemento viscoso em paralelo
 $F^{CE}, (\tilde{F}^{CE})$ = Força no elemento contrátil
 $B(\tilde{B})$ = Coeficiente de amortecimento do elemento viscoso
 $k^{PE}, (\tilde{k}^{PE})$ = rigidez do elemento elástico em paralelo (PE)
 $a_m(d)$ = curva de recrutamento em função da largura de pulso da excitação
 $T_c = \tau_{deact}$ = constante de tempo para relaxamento

O aspecto central deste modelo é o funcionamento do elemento contrátil. A partir do modelo de Zajac, pode-se determinar, da relação entre a deformação deste elemento e

da deformação do tendão, a primeira das duas equações diferenciais de 1ª ordem – também chamada dinâmica da contração - que descrevem o comportamento mecânico do músculo. Segundo o modelo de Zajac, mostrado na Figura 3.2, o comprimento do músculo relaxado, correspondente a uma situação de equilíbrio, é dado por:

$$\tilde{L}^M \cos \alpha + \tilde{L}^{ST} = \tilde{L}^{MT} \quad \text{Eq. 3.1}$$

Supondo a contração do músculo, a expressão acima fica:

$$\tilde{L}^M \cos \alpha + \tilde{L}^{ST} + \frac{\dot{F}^T}{\tilde{k}^T} = \tilde{L}^{MT} \quad \text{Eq. 3.2}$$

Derivando os dois lados da equação acima em relação ao tempo adimensional τ ,

$$\dot{\tilde{v}}^M \cos \alpha + \frac{\dot{F}^T}{\tilde{k}^T} = \dot{\tilde{v}}^{MT} \quad \text{Eq. 3.3}$$

isto é:

$$\boxed{\dot{F}^T = \tilde{k}^T (\dot{\tilde{v}}^{MT} - \dot{\tilde{v}}^M \cos \alpha)} \quad \text{Eq. 3.4}$$

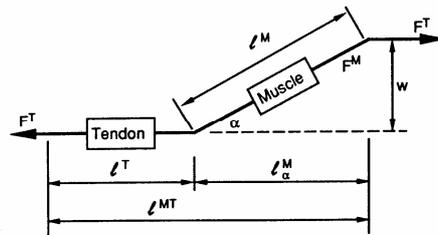


Figura 3.2: Modelo da dinâmica da contração de Zajac (Zajac, 1989).

Duas relações constitutivas definem o comportamento mecânico do elemento contrátil, as relações força \times comprimento e força \times velocidade. Essas curvas foram levantadas inicialmente por Hill para o músculo *gastrocnemius* do sapo, mas verificou-se que seu comportamento poderia ser estendido para a maior parte dos músculos

esqueléticos, tendo em conta alguns parâmetros de escala. Tais relações estão mostradas na Figura 3.3.

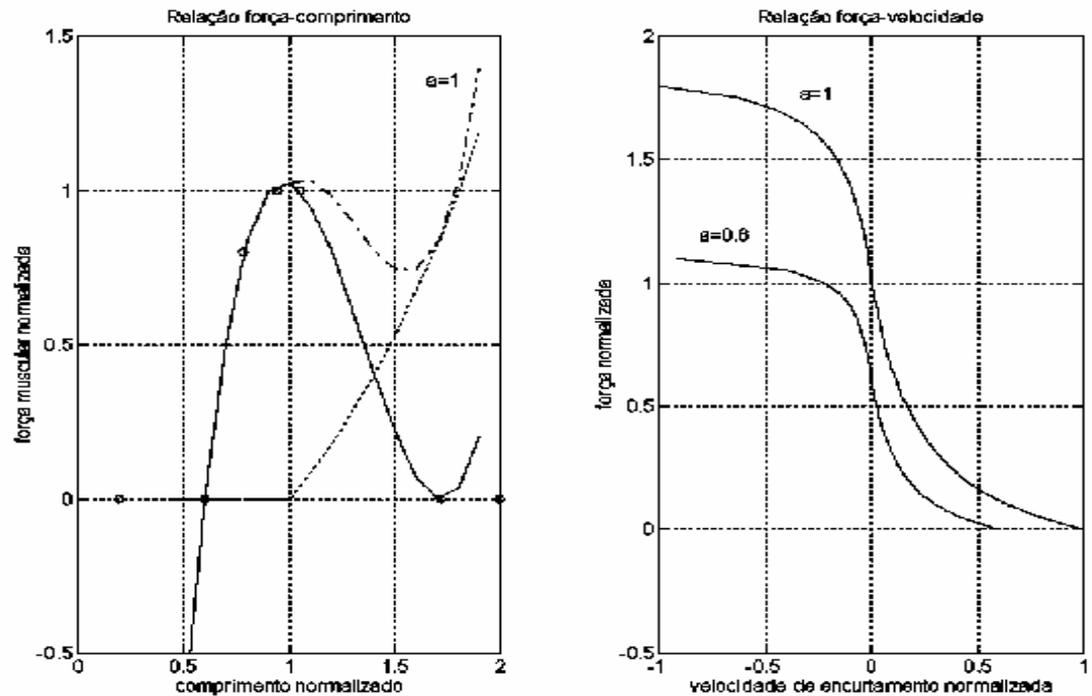


Figura 3.3: Relações força \times comprimento e força \times velocidade. No gráfico da esquerda, a linha contínua representa a relação força \times comprimento da componente ativa do músculo (para $a=1$), a linha tracejada a componente passiva e a outra linha a soma das duas anteriores.

Relação força \times comprimento¹¹:

Componente ativa

¹¹ Esses polinômios foram determinados utilizando os dados apresentados no modelo de Delp disponível em <http://isb.ri.ccf.org/data/delp/>. Os dados originais são utilizados pelo SIMM para determinar funções spline da relação força – comprimento. Aqui, os dados foram um pouco modificados, para que a forma final da relação polinomial fornecesse um aspecto plausível do ponto de vista fisiológico. Os dados utilizados estão mostrados na Tabela 3.1

$$fl_a = \tilde{F}_a^M = 1.9523 \cdot \tilde{L}^{M5} - 5.8369 \cdot \tilde{L}^{M4} + 2.8726 \cdot \tilde{L}^{M3} + 3.1788 \cdot \tilde{L}^{M2} - 1.2007 \cdot \tilde{L}^M - 0.0777 \quad \text{eq. 3.5}$$

Componente passiva

$$fl_p = \tilde{F}_p^M = 1.9660 \cdot \tilde{L}^{M5} - 5.2151 \cdot \tilde{L}^{M4} + 4.8491 \cdot \tilde{L}^{M3} - 1.8220 \cdot \tilde{L}^{M2} + 0.2242 \cdot \tilde{L}^M - 0.0004 \quad \text{eq. 3.6}$$

$$fl(\tilde{L}^M, a) = fl_a(\tilde{L}^M) \cdot a + fl_p(\tilde{L}^M) \quad \text{eq. 3.7}$$

Relação força × velocidade

- Contração

$$\left(\frac{\tilde{F}^{M*}}{a} + \frac{1}{4} \right) \left(\frac{\tilde{v}_e^M}{a} + \frac{1}{4} \right) = \frac{5}{16} \quad \text{eq. 3.8}$$

$$\tilde{v}_e^M = -\tilde{v}^M$$

$$\tilde{v}_e^M = \frac{a^2 - a\tilde{F}^{M*}}{4\tilde{F}^{M*} + a} \quad \text{eq. 3.9}$$

$$\tilde{F}^M = fl \cdot \tilde{F}^{M*} \quad \text{eq. 3.10}$$

$$\boxed{\tilde{v}_e^M = \frac{a^2 fl - a\tilde{F}^M}{4\tilde{F}^M + afl}} \quad \text{eq. 3.11}$$

- Alongamento ($\tilde{v}_a^M = -\tilde{v}^M$)

$$\frac{\tilde{F}^{M*}}{a} = 1,8 - 0,8 \left(\frac{v_m + \frac{v^{M*}}{a}}{v_m - \frac{7,6v^{M*}}{a}} \right) \quad \text{eq. 3.12}$$

$$\boxed{\tilde{v}_a^M = \frac{\tilde{F}^M a - a^2 fl}{7,6\tilde{F}^M - 14,48a \cdot fl}} \quad \text{eq. 3.13}$$

	fl ativa	fl passiva
pontos	0,0	0.00000,0.000000
utilizados na	0.1,0	0.2,0
geração das	0.15,0	
curvas força ×	0.2,0.1	0.4,0
comprimento,	0.25,0.2	0.6,0
nas sua	0.3,0.1	0.8,0
componentes	0.401000,0.000000	0.998000,0.000000
ativa e passiva	0.402000,0.000000	0.999000,0.000000
(comprimento,	0.403500,0.000000	1.000000,0.000000
força)	0.527250,0.226667	1.100000,0.035
	0.628750,0.636667	1.200000,0.120
	0.628750,0.636667	1.300000,0.260
	0.718750,0.856667	1.400000,0.550
	0.718750,0.856667	1.500000,1.170
	0.861250,0.950000	1.600000,2.000000
	1.045000,0.993333	1.601000,2.000000
	1.217500,0.770000	1.602000,2.000000
	1.438750,0.246667	
	1.618750,0.000000	
	1.620000,0.000000	
	1.621000,0.000000	
	1.7,0.0	
	1.8,0.0	

Tabela 3.1: Pontos utilizados na geração das curvas força × comprimento da porção ativa do músculo

Com relação à mecânica dos tendões, foi adotado um modelo elástico linear, supondo que a curva de rigidez apresenta o comportamento mostrado na Figura 3.4.

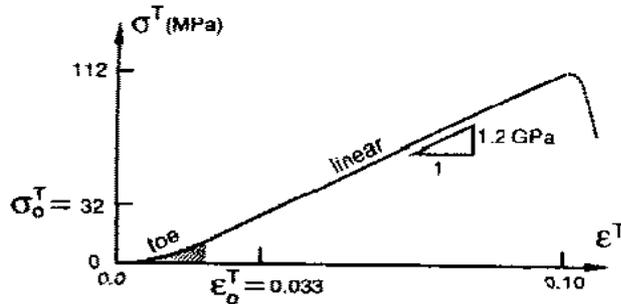


Figura 3.4: Curva teórica de rigidez do tendão (Zajac, 1989)

Sejam a tensão $\sigma^T = \frac{F^T}{A^T}$ e a deformação $\epsilon^T = \frac{\Delta L^T}{L_s^T} = \frac{(L^T - L_s^T)}{L_s^T}$.

Pela lei de Hooke, $\sigma^T = E^T \epsilon^T$. Se $\frac{F^T}{A^T} = E^T \frac{\Delta L^T}{L_s^T}$, temos que a força no tendão pode ser calculada como:

$$F^T = \frac{E^T A^T}{L_s^T} (L^T - L_s^T)$$

que, expressa em termos adimensionais, fica:

$$\tilde{F}^T = \frac{E^T A^T}{L_s^T} \left(\frac{L_0^M}{F_0^M} \right) (\tilde{L}^T - \tilde{L}_s^T) \quad \text{eq. 3.14}$$

Com isso, o coeficiente de rigidez do tendão é

$$\tilde{k}^T = \frac{E^T A^T}{L_s^T} \left(\frac{L_0^M}{F_0^M} \right) \quad \text{eq. 3.15}$$

Nos níveis máximos de tensão isométrica, a tensão de trabalho do tendão é:

$$\sigma_0^T = \frac{F_0^M}{A^T} = 32 \text{ MPa} \quad (1/3 \text{ tensão de ruptura})$$

Sendo, segundo Butler et al. (1984), $E^T = 1.2 \text{ GPa}$, e $\tilde{L}_s^T = L_s^T / L_0^M$, a rigidez de qualquer tendão pode, de acordo com este modelo, ser calculada como

$$\tilde{k}^T = \begin{pmatrix} E^T \\ \sigma_0^M \end{pmatrix} \begin{pmatrix} 1 \\ \tilde{L}_s^T \end{pmatrix} = \begin{pmatrix} 1,2 \times 10^9 \\ 32 \times 10^6 \end{pmatrix} \begin{pmatrix} 1 \\ \tilde{L}_s^T \end{pmatrix}$$

$$\boxed{\tilde{k}^T = \frac{37,5}{\tilde{L}_s^T}}$$

eq. 3.16

No ponto de união entre os elementos elástico em paralelo, viscoso e contrátil com o tendão, o equilíbrio de forças permite afirmar que:

$$\cos \alpha (\tilde{F}^{PE} + \tilde{F}^{DE} + \tilde{F}^{CE}) = \tilde{F}^T \quad \text{eq. 3.17}$$

Com relação ao elemento elástico em paralelo, de comportamento supostamente linear, aplicando a Lei de Hooke:

$$\tilde{F}^{PE} = -\tilde{k}^{PE} \Delta \tilde{L}^M \quad \text{eq. 3.18}$$

Por outro lado, a deformação dos três elementos em paralelo é a mesma, podendo-se calculá-la, na hipótese de que a contração é isométrica, a partir da deformação do tendão, isto é:

$$\Delta \tilde{L}^M = -\frac{\tilde{F}^T}{\tilde{k}^T \cos \alpha} \quad \text{eq. 3.19}$$

A força no elemento viscoso pode ser calculada, supondo também que a força varia linearmente com a velocidade de contração, como:

$$\tilde{F}^{DE} = -\tilde{B} \tilde{v}_e^M \quad \text{eq. 3.20}$$

Substituindo as três expressões anteriores na eq. 3.17,

$$-\tilde{B} \tilde{v}_e^M + \frac{fl \cdot a^2 - \tilde{v}_e^M a \cdot fl}{4\tilde{v}_e^M + a} = \frac{\tilde{F}^T}{\cos \alpha} \left(1 - \frac{\tilde{k}^{PE}}{\tilde{k}^T} \right) = Q \quad \text{eq. 3.21}$$

$$\tilde{v}_e^{M^2}(-4\tilde{B}) + \tilde{v}_e^M(-\tilde{B}a - a \cdot fl - 4Q) + (-Qa + a^2 fl) = 0 \quad \text{eq. 3.22}$$

cuja solução algébrica é:

$$\tilde{v}_e^M = \frac{-b_b - \sqrt{b_b^2 - 4a_b c_b}}{2a_b} \quad (\text{a solução } + \Delta \text{ não é consistente})$$

$$a_b = -4\tilde{B}, b_b = -(\tilde{B}a + afl + 4Q), c_b = a^2 fl - Qa \quad \text{eq. 3.23}$$

Substituindo $\tilde{v}_e^M = -\tilde{v}^M$ na Eq. 3.3, pode-se encontrar a 1ª equação diferencial da dinâmica da contração:

$$\dot{\tilde{F}}^T = \tilde{k}^T \left(\tilde{v}^{MT} + \frac{-b_b - \sqrt{b_b^2 - 4a_b c_b}}{2a_b} \cos \alpha \right) \quad \text{eq. 3.24}$$

A segunda equação diferencial do modelo é a da dinâmica da ativação, representando o processo de geração da despolarização da membrana muscular a partir da chegada do potencial de ação na junção neuro-muscular. A formulação apresentada por Zajac (1989) e adotada em Menegaldo (1997) apresentava problemas de integração numérica, tendo-se aqui optado pela formulação proposta por Piazza e Delp (1996),

$$da/dt = (u - a)(k_1 u + k_2) \quad \text{eq. 3.25}$$

em que $T_{act} = 1/(k_1 + k_2)$ e $T_{deac} = 1/k_2$. Valores utilizados: $T_{act} = 0.020$ e $T_{deac} = 0.050$. T_{act} e T_{deact} são as constantes de tempo de ativação e deativação, respectivamente

Para gerar os modelos específicos, tendo em conta os parâmetros de cada músculo, foi utilizado um programa Matlab com manipulador algébrico. O programa abaixo gera três músculos específicos para serem inseridos num modelo de pêndulo triplo: os estados x_7 a x_9 correspondem às expressões da dinâmica da ativação e x_{10} a x_{12} à mecânica da

contração. Os parâmetros musculares utilizados nos testes a seguir não correspondem a nenhum músculo específico, mas foram escolhidos dentro de faixas plausíveis.

alg_test_mus.m

```
%2/6/99
%Problema do modelo de músculo
%testes do músculo não-linear e determinação do modelo linear
%específico de cada músculo
%utilizar matlab 5.3 para cálculos algébricos

clear global
clear all

pr=16; %precisão numérica dos resultados

syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 U1 U2 U3 U4 U5 U6
real

syms Ftti Vetí Vmtti kpeti ktíi Bti a Lom Fom fl alfo u k1 k2 real

Q=Ftti/cos(alfo)*(1+kpeti/ktíi);
eq2=Veti^2*(4*Bti)+Veti*(a*Bti+a*fl+4*Q)+(Q*a-a^2*fl);
sol2=(solve(eq2,Veti));
Veti=sol2(1);

%equações diferenciais:
Vmtti=0; %velocidade do músculo+tendão (imposta)
ap=(u-a)*(k1*u+k2); %da/dt=f(1)
Fttip=ktíi*(Vmtti+Veti*cos(alfo)); %dFtti/dtal=f(2)

%parâmetros da dinâmica da ativação
Talc=0.1;
%expressões para a o comprimento e velocidade dos atuadores músculo-tendíneos
%parâmetros da dinâmica da contração
Lmt=[0.4 0.4 0.4];
Vmtti=[0 0 0];
Fom=[8000 8000 8000];
Lst=[0.1 0.1 0.1];
alfo=pi/180*[5 5 5];
Lom=[0.3 0.3 0.3];
B=[225 225 225];
kpe=[1500 1500 1500];
Lstti=Lst./Lom;
ktíi=37.5./Lstti; %rigidez do tendão
kt=ktíi.*Fom./Lom;
Lmtti=Lmt./Lom;
Bti=B.*Lom./(Fom); % elemento viscoso em paralelo
kpeti=kpe.*Lom./(Fom); % elemento elástico em paralelo

%parâmetros da dinâmica da ativação
Tact=0.020;
Tdeact=0.050;
k1=1/Tact-1/Tdeact;
k2=1/Tdeact;

%substituição de parâmetros adimensionais

%dinâmica da ativação
x7p=subs(ap,{'a','k1','k2','u'},{'x7',k1,k2,U1});
x8p=subs(ap,{'a','k1','k2','u'},{'x8',k1,k2,U2});
x9p=subs(ap,{'a','k1','k2','u'},{'x9',k1,k2,U3});

%dinâmica da contração
i=1;
x10p=subs(Fttip,'fl',1); % relação força-comprimento cte.=1
x10p=subs(x10p,'a','x7');
x10p=subs(x10p,'Ftti','x13');
x10p=subs(x10p,'ktíi',ktíi(i));
```

```

x10p=subs(x10p,{ 'Lmtti', 'Lstti' }, {Lmtti(i),Lstti(i)});
x10p=subs(x10p, 'Bti', Bti(i));
x10p=(subs(x10p, { 'alfo', 'kpeti', 'Lom', 'Fom' }, {alfo(i),kpeti(i),Lom(i),Fom(i)}));
x10p=vpa(x10p,pr)

i=2;
x11p=subs(Fttip, 'f1', 1);
x11p=subs(x11p, 'a', 'x8');
x11p=subs(x11p, 'Ftti', 'x14');
x11p=subs(x11p, 'ktti', ktti(i));
x11p=subs(x11p, { 'Lmtti', 'Lstti' }, {Lmtti(i),Lstti(i)});
x11p=subs(x11p, 'Bti', Bti(i));
x11p=(subs(x11p, { 'alfo', 'kpeti', 'Lom', 'Fom' }, {alfo(i),kpeti(i),Lom(i),Fom(i)}));
x11p=vpa(x11p,pr)

i=3;
x12p=subs(Fttip, 'f1', 1);
x12p=subs(x12p, 'a', 'x9');
x12p=subs(x12p, 'Ftti', 'x15');
x12p=subs(x12p, 'ktti', ktti(i));
x12p=subs(x12p, { 'Lmtti', 'Lstti' }, {Lmtti(i),Lstti(i)});
x12p=subs(x12p, 'Bti', Bti(i));
x12p=(subs(x12p, { 'alfo', 'kpeti', 'Lom', 'Fom' }, {alfo(i),kpeti(i),Lom(i),Fom(i)}));
x12p=vpa(x12p,pr)

```

Para identificar modelos lineares que representassem a mecânica muscular da melhor maneira possível, foi utilizado o método dos mínimos quadrados. Essas aproximações lineares da mecânica muscular foram utilizadas em modelos simplificados da postura e na geração de estimativas iniciais dos vetores de controle, através de um regulador LQR, no Problema de Controle Ótimo.não-linear

Inicialmente, é gerado um sinal de excitação senoidal do tipo

$$u = 0.5*(1+\sin(2*2*\pi*t))$$

Este sinal é aplicado ao modelo dinâmico do músculo, calculando as ativações e

forças musculares:

```

tid=[0:0.001:1];
y0=[0.5 0.5]';
[t,y] = ode23s('ident_musc',tid,y0);
figure(1)
clf
plot(t,y(:,1), 'k', t,y(:,2), 'k:', t,0.5*(1+sin(2*2*pi*t)), 'k--')
xlabel('tempo (s)')
ylabel('u(t), a(t), F(t)')
grid

```

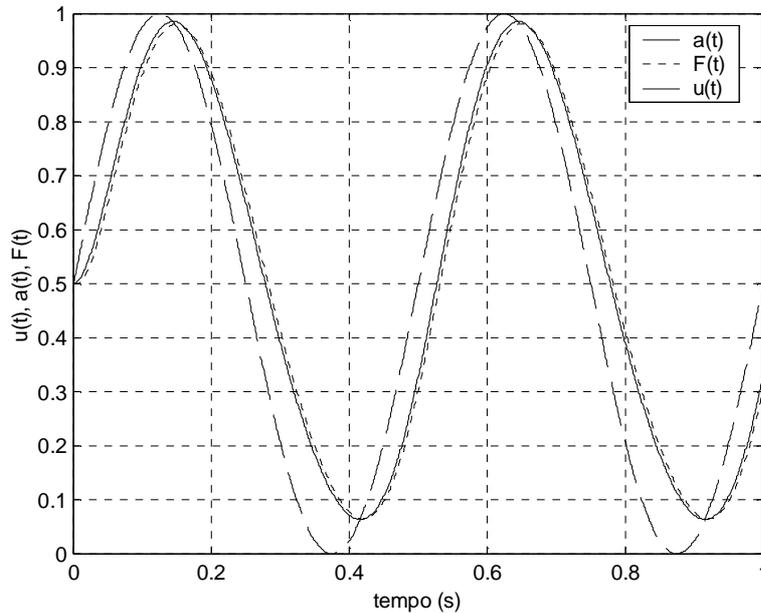


Figura 3.5: Identificação do modelo linear da mecânica muscular.

Sendo a função `ident_musc.m`

```
function yp=ident_musc(t,y)
Talc=0.1;
Tact=0.020;
Tdeact=0.050;
k1=1/Tact-1/Tdeact;
k2=1/Tdeact;

%U=0.5;
U=0.5*(1+sin(2*2*pi*t));

%modelo din. contração adimensional, com dimensionalização do tempo
yp(1)=(U-y(1))*(k1*U+k2);
yp(2)=1/Talc*(-6670.*y(2)-1674.333484761491*y(1)+14.81481481481481*
(14468.57267751850*y(1)^2 +100063.6739979000*y(1)*y(2)+202702.5506250000*y(2)^2)^(1/2));

%caso limite de contração excêntrica
if Ftti/cos(alfo) > 1.4, Vmti=-0.15; end;
yp=yp';
```

Utilizando o método proposto por Press et al. (1992), descrito no Capítulo 2 para caso de ajuste de funções algébricas multi-variáveis, são calculadas matrizes de projeto $[A]$ e $[AF]$ para a ativação e para a força muscular:

$$[A] = \begin{bmatrix} a(1) & \dot{a}(1) \\ a(2) & \dot{a}(2) \\ \vdots & \vdots \\ a(n) & \dot{a}(n) \end{bmatrix} \quad [AF] = \begin{bmatrix} F(1) & \dot{F}(1) \\ F(2) & \dot{F}(2) \\ \vdots & \vdots \\ F(n) & \dot{F}(n) \end{bmatrix} \quad \text{eq. 3.26}$$

```

%determinação do modelo de músculo linear equivalente
%identificação de um sistema linear de 1a. ordem para a dinâmica da ativação e da
contração
t=tid;
for i=1:length(tid),
    yy=ident_musc(t(i),y(i,:));
    yp1(i)=yy(1);
    yp2(i)=yy(2);
end
%excitação
b=0.5*(1+sin(2*2*pi*tid)); %senoidal
%b=1*ones(length(tid),1)'; %degrau
%matriz de projeto
clear A
clear AF
for i=1:length(tid),
%   A(i,:)=[1 y(i,1) yp1(i)]; %modelo linear com termo constante
%   AF(i,:)=[1 y(i,2) yp2(i)]; %modelo linear sem termo constante
    A(i,:)=[y(i,1) yp1(i)]; %ativação
    AF(i,:)=[y(i,2) yp2(i)]; %força
end

```

Tendo-se em conta a excitação (vetor **b**), são calculados pelas expressões

$$\mathbf{aa} = ([A]^T [A])^+ [A]^T \mathbf{b}^T \quad \text{e} \quad \mathbf{aaf} = ([AF]^T [AF])^+ [AF]^T \mathbf{b}^T \quad \text{eq. 3.27}$$

os vetores de coeficientes **aa** e **aaf** das equações lineares identificadas, respectivamente da dinâmica da ativação e da contração. $[\cdot]^+$ significa a matriz pseudo-inversa. Foram testadas duas estruturas de equações lineares, uma com termo constante e outra sem. Para a dinâmica da ativação essas equações são:

$$\text{ou} \quad \begin{aligned} \mathbf{aa}(3) \dot{a} + \mathbf{aa}(2) a + \mathbf{aa}(1) &= u \\ \mathbf{aa}(2) \dot{a} + \mathbf{aa}(1) a &= u \end{aligned} \quad \text{eq. 3.28}$$

Optou-se, porém, pela equação sem termo constante, que reproduzia melhor a dinâmica não-linear, especialmente em baixas ativações. Os resultados da identificação estão mostrados na Figura 3.5. Os coeficientes identificados para o conjunto de parâmetros adotados foram: $\mathbf{aa} = [0.9860 \ 0.0296]$ e $\mathbf{aaf} = [1.0037 \ 0.0045]$.

```

%vetor de coeficientes
global aa aaf

```

```

aa=pinv(A'*A)*A'*b';
aaf=pinv(AF'*AF)*AF'*y(:,1);

%teste do músculo linear identificado
[t,al]=ode23s('alin',tid,y0);
figure(1)
hold on
plot(tid,al(:,1),'r',tid,al(:,2),'m')
grid
hold off

```

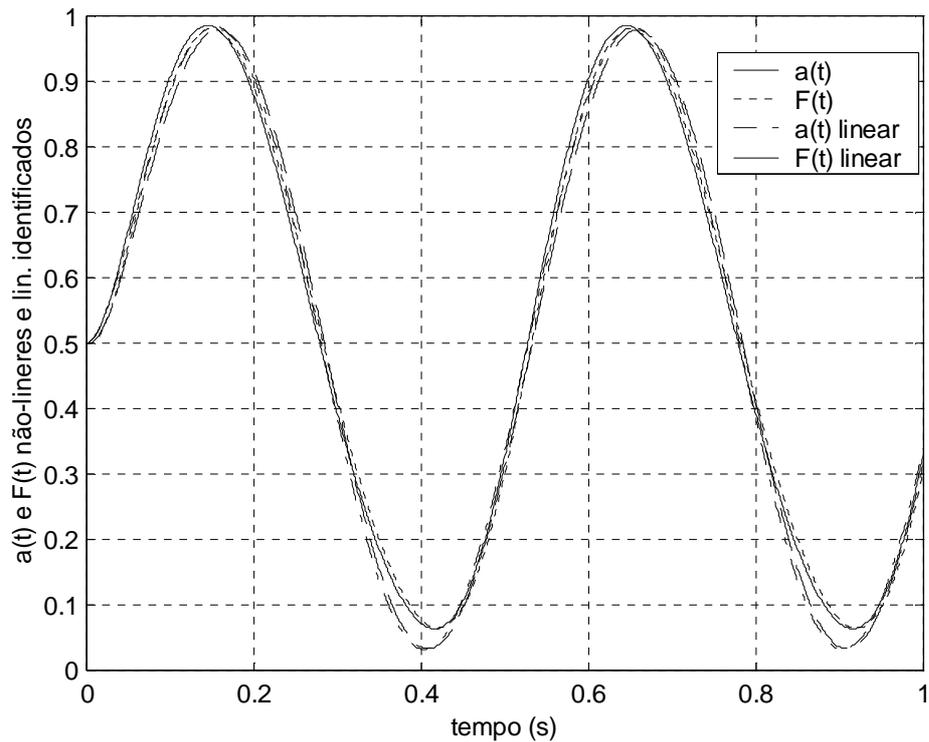


Figura 3.6: Identificação do modelo linear da mecânica muscular

Utilizando a metodologia descrita acima, foram identificados modelos lineares específicos de músculos, posteriormente vinculados às equações da dinâmica de corpo rígido descritas nos próximos capítulos.

Capítulo 4

Formulação do Problema de Controle Ótimo

Dado um sistema dinâmico expresso por equações diferenciais ordinárias não-lineares, o problema de controle ótimo (*Optimal Control Problem, OCP*) consiste na determinação de uma trajetória factível para as variáveis de controle que leve o sistema de uma certa condição inicial até uma condição terminal, ao mesmo tempo que é minimizada uma função objetivo ou índice de performance (IP) e respeitado um conjunto de vínculos, sejam as próprias equações diferenciais do sistema dinâmico, vínculos terminais e de trajetória de desigualdade e/ou igualdade, assim como limites laterais (também chamados vínculos simples, ou *simple bounds*) nas variáveis de controle.

A literatura traz diversos métodos para resolver OCPs, com diferentes vantagens e desvantagens no que diz respeito aos custos computacionais, às propriedades de convergência, dificuldades teóricas de demonstração, abrangência de tipos de vínculos, precisão, etc. Deste modo, a seleção do método mais adequado deve levar em conta o problema específico que se deseja resolver, assim como os recursos disponíveis.

O objetivo principal da solução de OCP seria obter trajetórias de controle através de uma *lei de controle* do tipo:

$$\mathbf{u}^* = \mathbf{f}(\mathbf{x}(t), t)$$

em que $\mathbf{x}(t)$ é o vetor de estados do sistema e \mathbf{u}^* é o controle ótimo. Neste caso, o controle é chamado de **realimentação** (*feedback*) ou **malha fechada**, existindo uma regra definida que relaciona a trajetória da variável de controle e os estados do sistema a cada momento. A função \mathbf{f} pode ser linear ou não linear, variável ou invariante com o tempo. Entretanto, em alguns casos, a grande complexidade do problema permite apenas determinar uma trajetória de controle que não depende diretamente dos estados do sistema ao longo do tempo, mas que, de fato, leva o sistema de um estado a outro, respeitando os vínculos existentes. Trata-se, nesse caso, de um problema de **geração de trajetória** ou **malha aberta**. Cada uma dessas classes de problemas possui metodologias próprias para sua solução, ainda que estejam intimamente relacionados, tanto do ponto de vista teórico – nos métodos variacionais ou indiretos, por exemplo, a realimentação é um caso limite da geração de trajetórias – quanto prático, pois perturbações que se introduzem no sistema podem levá-lo a que se afaste da trajetória desejada. Assim, pode ser conveniente que se introduzam, num problema de seguimento de trajetória previamente gerada, leis de controle de realimentação formuladas com base em modelos simplificados do sistema

dinâmico, por exemplo, linearizando-o ao longo da trajetória de movimento. Metodologias específicas para as duas classes de problema estão mostradas nos itens que se seguem.

Problema da geração de trajetórias

Uma tentativa de classificação dos métodos de controle disponíveis na literatura é apresentada por Schwartz (1996), situando-os em relação ao seu próprio método, empregado no presente trabalho. Este Capítulo traz essencialmente uma síntese da teoria apresentada por este autor na sua tese de doutorado¹². Uma árvore de métodos de controle ótimo, procurando situar o aqui utilizado em relação aos disponíveis na literatura, está mostrada na Figura 4.1

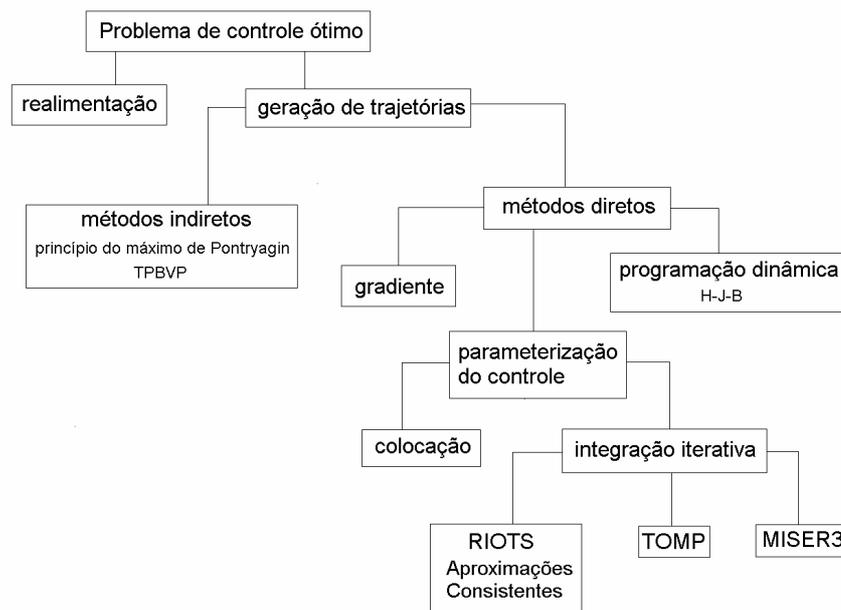


Figura 4.1: Árvore mostrando os métodos disponíveis de controle ótimo

¹²Esse trabalho se encontra disponível no *site* <http://www.accesscom.com/~adam/RIOTS/>

Os chamados **métodos indiretos** baseiam-se na solução de condições relacionadas ao Princípio do Máximo de Pontryagin que levam à minimização indireta da função objetivo:

$$G_0(\mathbf{u}, \mathbf{x}(t)) = \phi_0(\mathbf{x}(t_f)) + \int_t^{t_f} g_0(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad \text{eq. 4.1}$$

sujeita a vínculos dinâmicos expressos por equações diferenciais ordinárias

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \quad \text{eq. 4.2}$$

a vínculos de contorno

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}^0 \\ h(\mathbf{x}(t_f)) &= 0 \end{aligned} \quad \text{eq. 4.3}$$

e a vínculos de desigualdade nas variáveis de controle:

$$\mathbf{u}^{\min} \leq \mathbf{u}(t) \leq \mathbf{u}^{\max} \quad \text{eq. 4.4}$$

tal que $\mathbf{x}(t) = [x_1(t), \dots, x_n(t)] \in \mathfrak{R}^n$ é um vetor de variáveis de estado, $\mathbf{u}(t) = [u_1(t), \dots, u_m(t)] \in \mathfrak{R}^m$ é um vetor de variáveis de controle, $\mathbf{f}, \mathbf{h}, \mathbf{x}^0 \in \mathfrak{R}^n$ e $\phi, g_0 \in \mathfrak{R}$. t_f pode ser ou não especificado.

Escrevendo-se o Hamiltoniano do sistema como uma função que depende das equações diferenciais do sistema, do integrando da função objetivo e de variáveis de co-estado (ou multiplicadores de Lagrange),

$$H(t, \mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = g_0 + \boldsymbol{\lambda}^T \mathbf{f} \quad \text{eq. 4.5}$$

uma condição necessária de primeira ordem para otimalidade é que o gradiente dessa função em relação à variável de controle seja nula:

$$\nabla_{\mathbf{u}} H = 0 \quad \text{eq. 4.6}$$

Outra condição, que o gradiente do Hamiltoniano em função do vetor de estados seja igual ao negativo da derivada da variável de co-estado:

$$\dot{\lambda}^T = -\nabla_x H \quad \text{eq. 4.7}$$

Essas novas equações, por sua vez, possuem condições finais a serem satisfeitas:

$$\lambda(t_f)^T = \frac{\partial \phi(\mathbf{x}(t_f))}{\partial \mathbf{x}(t_f)} \quad \text{eq. 4.8}$$

O cálculo de $\mathbf{u}(t, \lambda, \mathbf{x})$ através da equação eq. 4.6, substituído na equação eq. 4.2 sujeita aos vínculos eq. 4.3 junto com as equação eq. 4.7 sujeita aos vínculos eq. 4.8 dá origem a um sistema de equações diferenciais com condições de contorno em dois pontos (*Two Point Boundary Value Problems*), que pode ser resolvido por um método conveniente¹³.

Soluções de OCPs baseadas no Cálculo Variacional são historicamente as mais importantes e, em problemas de dimensões relativamente pequenas, podem ser a primeira opção. Entretanto, o esforço algébrico cresce rapidamente com a dimensão do problema; ainda, as propriedades de robustez dos métodos numéricos associados ao TPBVP são pobres, sendo freqüentemente necessária a escolha de uma trajetória inicial bastante próxima da solução. Além disso, propriedades de otimalidade de 2ª ordem que garantem ser o ponto singular da solução obtida pelo método de 1ª ordem de fato um mínimo, pelo menos local, da hipersuperfície da função de custo requerem a solução do Problema do Mínimo Acessório (ver por exemplo Citron 1969 item 2-11).

A segunda grande classe de métodos de solução do OCP são os **métodos diretos**. Procura-se, a partir destes, minimizar diretamente a função objetivo sem estabelecer aprioristicamente que sua primeira variação seja nula, condição atingida apenas no limite

de convergência. O **método do gradiente** (*steepest descent*) (Bryson e Ho, 1975, item 7.4; Fleury, 1978; Citron, 1969, cap.6) é o método direto de 1ª ordem clássico. Sua principal característica é a boa convergência nas primeiras iterações, não requerendo estimativas iniciais do controle muito próximas da solução; entretanto, próximo à solução suas propriedades de convergência são pobres; além disso, sua formulação é bastante complexa. Existem também métodos de segunda ordem associados ao método do gradiente, baseados na condição de assegurar que a Hessiana da função objetivo é positiva-definida (Bryson e Ho, 1975, item 7.5). De qualquer maneira, Schwartz (1996) observa que, pelo fato dos algoritmos conceituais requererem operações numéricas de alta precisão, como cálculo de integrais, mesmo nas primeiras iterações, a precisão numérica dos resultados é baixa.

Outros métodos diretos estão baseados em algoritmos de **programação dinâmica**, com os quais as equações diferenciais do sistema dinâmico e da função objetivo são transformados em equações a diferenças finitas, utilizando equações recursivas baseadas no Princípio de Otimalidade de Bellman¹⁴ (Kirk, 1970; Bertsekas, 1995). O problema de controle ótimo, que nesta abordagem se torna um problema de realimentação ótima, pode ser também formulado como a solução de uma equação diferencial parcial sujeita a condições de contorno, a chamada Equação de Hamilton - Jacobi - Bellman (H-J-B). Bryson e Ho (1975, Cap. 4) não recomendam a utilização deste algoritmos para sistemas com número grande de estados, devido a consumo excessivo de memória.

¹³ Os métodos mais utilizados para resolver o TPBVP em controle ótimo são os do tipo *point-and-shoot*. Implementações destes algoritmos podem ser encontrados em biblioteca de funções matemáticas para Fortran IMSL/MATH, ou no Numerical Recipes (Press et al., 1992).

¹⁴ Calculando-se caminhos de mínimo custo entre os pontos da malha de discretização, Bellman mostrou que o custo total, isto é, a soma dos custos parciais dos vários intervalos da malha, fornece uma solução que é ótima entre o início e o fim do processo.

Pelo fato de apresentarem grande complexidade teórica, exigirem um elevado esforço computacional e só convergirem para a solução ótima no limite, a aplicação de algoritmos diretos tem levado a propostas que utilizam recursos da **programação matemática, métodos de otimização** ou **programação não-linear**, os chamados métodos de **aproximações consistentes**.

Nos algoritmos baseados nas aproximações consistentes, o controle ótimo é obtido pela solução de uma seqüência de OCPs de dimensão finita (ou discretos), o que pode ser feito por métodos convencionais de programação matemática de dimensão finita. As aproximações consistentes, por sua vez, podem ainda ser realizadas a partir de duas classes de algoritmos: os **métodos de colocação** e os de **integração iterativa**. Nos primeiros (por exemplo Cuthrell e Biegler, 1987), as equações diferenciais são substituídas por relações que representam condições de colocação que devem ser satisfeitas para um número finito de pontos no tempo. O problema de programação matemática gerado passa a ser responsável por encontrar, simultaneamente aos parâmetros de controle, estados do sistema ao longo dos pontos da malha de discretização. Tais métodos são convenientes para problemas com dinâmica instável, sistemas de equações diferenciais duras (*stiff*)¹⁵ e vínculos nas variáveis de estado. Entretanto, o número de variáveis de decisão multiplica-se com o nível de discretização, levando em alguns casos a soluções pouco precisas, eventualmente refináveis por métodos indiretos.

Já os métodos de integração iterativa procuram resolver numericamente as equações diferenciais e expressar as variáveis de controle num subespaço funcional de dimensão finita do espaço de dimensão infinita do problema original. Posteriormente, o problema de programação matemática é tratado com algum algoritmo apropriado, tendo como resultado

os valores das variáveis discretas de controle. A teoria e as implementações computacionais apresentadas por Schwartz (1996), utilizadas no presente trabalho para solução do problema de geração de trajetórias de excitação muscular da postura humana, introduz uma série de contribuições para essa classe de algoritmos:

- Determinação das condições para parâmetros de integração Runge-Kutta capazes de garantir a convergência de pontos estacionários do problema discretizado para os pontos estacionários do problema original.
- Dedução de uma métrica não-euclidiana utilizada no problema de otimização associado e de uma transformação de coordenadas que permite o emprego de uma métrica euclidiana associada, evitando problemas de mau-condicionamento numérico.
- Determinação de expressões para limites superiores dos erros de integração e de aproximação do controle por funções spline, em função do nível de discretização, em problemas com e sem vínculos.
- Um novo algoritmo de programação matemática baseado no método de Newton.
- Um novo método de solução para OCPs singulares que minimiza oscilações nas soluções numéricas.
- Um software em ambiente MATLAB chamado RIOTS (*Recursive Integration Optimal Trajectory Solver*), capaz de resolver OCPs a partir da teoria apresentada.

Dada a complexidade da abordagem matemática para a dedução da metodologia e sua aplicação, bem como da sua relativa novidade, nos próximos itens deste Capítulo será

¹⁵ Sistemas de equações diferenciais que apresentam simultaneamente constantes de tempo muito grandes e muito pequenas.

mostrado um breve resumo da teoria e da implementação de métodos de controle ótimo baseados em integração RK e aproximações por splines das trajetórias de controle, desenvolvidas por aquele autor. A maior parte das demonstrações e detalhes formais foi omitida, por se situarem fora do escopo deste trabalho. Concentra-se assim a exposição sobre os princípios gerais da teoria, os teoremas mais importantes e as implicações práticas relacionadas com a escolha de parâmetros da solução. Ao mesmo tempo, foram incluídas no texto algumas explicações procurando elucidar conceitos de análise matemática e alguns pontos mais obscuros, ainda que importantes, da teoria estudada. Para a compreensão formal da teoria e de diversas questões analíticas e numéricas associadas, recomenda-se a leitura da tese de Schwartz (1996).

Além do RIOTS, existem outros softwares de integração iterativa disponíveis na literatura. O TOMP (Kraft, 1994), desenvolvido por Dieter Kraft, da Fachhochschule München, é um software com código aberto¹⁶ escrito em Fortran 77, que transforma o OCP em um problema de programação não-linear, resolvendo-o por um método SQP (ver item 4.11). Suas características peculiares são a utilização de uma 'malha de comunicação' de pontos onde valem e são avaliados vínculos de trajetória, a opção de otimizar simultaneamente parâmetros de projeto e o vetor de controles e a possibilidade de se utilizar diversas representações do controle, inclusive funções spline. Além disso, apresenta vários algoritmos de integração, entre os quais métodos Runge-Kutta de ordem 8. Entre as suas limitações estão: avaliação das derivadas numéricas apenas por diferenças finitas, não permitindo o uso opcional de fórmulas analíticas, documentação escassa (apenas 1 artigo e um livro esgotado) e restrição de no máximo 6 variáveis de controle. Entretanto, já foi testado com sucesso em diversos problemas e, tratando-se de um código

¹⁶ Disponível em <http://www.netlib.org/toms/733>

aberto, poderia ser aprimorado, principalmente no sentido de aumentar o número de variáveis de controle.

Outro software em Fortran 77 para solução de OCPs baseado na discretização do sistema dinâmico e na solução de um problema de otimização é o MISER3¹⁷, desenvolvido por Leslie Jennings e outros pesquisadores da University of Western Australia (Jennings et al., 1997). Utiliza, o integrador LSODA (ver item 4.12) e dois algoritmos possíveis de otimização tipo SQP: NLPQL¹⁸ e o FFSQP¹⁹ (Zhou e Tits, 1997). Sua principal característica é o tratamento formal dos vínculos que envolvem variáveis de estado ou funções não-lineares do controle que são transformadas para uma forma canônica padrão. Além disso, possui métodos para a suavização de funções de custo não diferenciáveis, que podem surgir em alguns problemas de biomecânica²⁰ (Jennings et al., 1991; Jennings et al., 1993). Entre outras características, utiliza apenas funções constantes ou lineares por partes na representação do controle, permite que a malha de discretização do controle não seja igualmente espaçada e mesmo diferente para cada controle, podendo ainda utilizar derivadas analíticas ou numéricas das funções de custo e controle.

Existe também um software comercial chamado SOCS²¹ (*Sparce Optimal Control Software*), desenvolvido por John Betts (Betts, s.d.), da Boeing Corporation. Parece ser um software bastante refinado, possibilitando o uso de diversas rotinas numéricas. Carece, entretanto, de literatura disponível ao público acadêmico, além de ter alto custo (US\$ 1.600,00 com 80% de desconto acadêmico, licença anual). Entre os recursos disponíveis, possui métodos de refinamento da malha para alcançar uma certa precisão especificada

¹⁷ Manual disponível em <http://cado.uwa.edu.au/miser/>

¹⁸ <http://www.uni-bayreuth.de/departments/math/org/mathe5/staff/memb/kschittkowski/nlpql.htm>

¹⁹ <http://www.isr.umd.edu/Labs/CACSE/FSQP/fsqp.html>

²⁰ Por exemplo, na mudança de parâmetros de modelos da mecânica muscular com contração excêntrica e concêntrica, ou na marcha, cujas equações dinâmicas são diferentes dependendo da fase.

pelo usuário, 10 diferentes tipos de discretização e permite também, além do método direto, a solução de TPBVPs através de equações adjuntas. Possui ainda algoritmos de programação quadrática esparsa para problemas de grande porte e métodos SQP convencionais para problemas de tamanho moderado, utilizando algoritmos específicos para a detecção da esparsidade das matrizes Jacobiana e Hessiana.

4.1 Teoria das Aproximações Consistentes

O objetivo da teoria das aproximações consistentes é indicar de maneira formalmente rigorosa as condições capazes de assegurar que as soluções e pontos estacionários do problema de aproximação convergem para as soluções e pontos estacionários do problema original. Dadas essas condições, é possível então determinar a trajetória de controle desejada num determinado subespaço de coeficientes, através de técnicas convencionais de programação matemática. Daniel (1971) apresenta os primeiros avanços na tentativa de estabelecer uma teoria de consistência de aproximações em OCPs, assegurando a convergência dos minimizadores do problema de aproximação para os minimizadores do problema original. Polak (1993) introduziu o conceito de aproximações consistentes incluindo condições provenientes de *funções de otimalidade* e garantindo a convergência de pontos estacionários.

Formalmente, a teoria das aproximações consistentes pode ser introduzida considerando H um espaço linear normado e $\mathbf{B} \subset H$ um conjunto convexo. Uma família de problemas \mathbf{P} pode ser formulada como:

$$\mathbf{P} \quad \min_{\eta \in \mathbf{F}} \psi(\eta) \quad \text{eq. 4.9}$$

²¹ Maiores informações em <http://www.boeing.com/assocproducts/socs/index.html>

sendo a função $\psi: \mathbf{B} \rightarrow \mathfrak{R}$ pelo menos semi-contínua inferiormente²² e $\mathbf{F} \subset \mathbf{B}$ um conjunto factível. η é um elemento, eventualmente multidimensional, de \mathbf{F} .

Seja $\mathbf{N} = \{1, 2, 3, \dots\}$ um conjunto enumerável de \mathbf{N} (números naturais) e $\{H_N\}_{N \in \mathbf{N}}$ uma família de subespaços de dimensão finita de H .

Considere-se agora a família de problemas de aproximação \mathbf{P}_N

$$\mathbf{P}_N \quad \min_{\eta \in \mathbf{F}_N} \psi_N(\eta) \quad \text{eq. 4.10}$$

de maneira que $\psi_N: H_N \rightarrow \mathfrak{R}$ é pelo menos semi-contínua inferiormente e $\mathbf{F}_N \subset H_N \cap \mathbf{B}$.

Polak (1993) estabelece duas concepções de consistência dos problemas de aproximação \mathbf{P}_N , uma baseada no conceito de *epiconvergência* (ver Definição 1) e outra na consideração de que pontos estacionários são zeros de uma *função de otimalidade* (ver Definição 3).

Definição 1: Os problemas de aproximação da classe \mathbf{P}_N convergem epigraficamente²³ (ou epiconvergem) para \mathbf{P} se

²² Segundo Polak (1997), proposição 5.1.19, se v é um espaço real munido de norma e S um subconjunto convexo de v ,

Uma função $f: v \rightarrow \mathfrak{R}$ é contínua em x^* se e somente se, para qualquer seqüência $\{x_i\}_{i=0}^{\infty}$ em v tal que $x_i \rightarrow x^*$ quando $i \rightarrow \infty$, $f(x_i) \rightarrow f(x^*)$ quando $i \rightarrow \infty$.

Uma função $f: v \rightarrow \mathfrak{R}$ é semi-contínua inferiormente em x^* se e somente se, para qualquer seqüência $\{x_i\}_{i=0}^{\infty}$ em v tal que $x_i \rightarrow x^*$ quando $i \rightarrow \infty$, $\underline{\lim} f(x_i) \geq f(x^*)$ quando $i \rightarrow \infty$.

Além disso, segundo a definição 5.1.8 de Polak (1997), se $\{\alpha_i\}_{i=0}^{\infty}$ é uma seqüência limitada de números reais e S o conjunto de todos os pontos de acumulação (ver nota de rodapé 24) da seqüência, os limites superior e inferior são definidos como:

$$\overline{\lim} \alpha_i = \max \{ \alpha \mid \alpha \in S \}$$

$$\underline{\lim} \alpha_i = \min \{ \alpha \mid \alpha \in S \}$$

A seqüência $\{\alpha_i\}_{i=0}^{\infty}$ converge para α^* se e apenas se $\alpha^* = \overline{\lim} \alpha_i = \underline{\lim} \alpha_i$

²³ Isto é, os epígrafos do problema de aproximação convergem para os epígrafos do problema original. Segundo a definição apresentada por Bazaraa et al. (1993, def. 3.2.1), o epígrafo é um par de subconjuntos (x, y) do domínio e da imagem de uma função, respectivamente, tal que $y \geq f(x)$.

- a. para todo $\eta \in \mathbf{F}$ existe uma seqüência $\{\eta_N\}_{N \in \mathbf{N}}$, com $\eta_N \in \mathbf{F}_N$, tal que $\eta_N \rightarrow \eta$ e $\overline{\lim} \psi_N(\eta_N) \leq \psi(\eta)$.
- b. Para toda seqüência infinita $\{\eta_N\}_{N \in \mathbf{K}}$, $\mathbf{K} \subset \mathbf{N}$, satisfazendo $\eta_N \in \mathbf{F}_N$ para todo $N \in \mathbf{K}$ e $\eta_N \rightarrow \eta$, $\eta \in \mathbf{F}$ e $\underline{\lim}_{N \in \mathbf{K}} \psi_N(\eta_N) \geq \psi(\eta)$.

Pode-se chegar, com a Definição 1, ao Teorema 2, que garante que os pontos minimizadores locais e globais do problema de aproximação convergem para os minimizadores locais e globais do problema original.

Teorema 2 (Polak, 1993): Suponha-se que \mathbf{P}_N converge epigraficamente para \mathbf{P}

- a. Se para $N \in \mathbf{N}$, $\hat{\eta}_N$ é um minimizador global de \mathbf{P}_N e $\hat{\eta}$ um ponto de acumulação²⁴ qualquer da seqüência $\{\eta_N\}_{N \in \mathbf{N}}$, então $\hat{\eta}$ é um minimizador global de \mathbf{P} .
- b. Se para $N \in \mathbf{N}$, $\hat{\eta}_N$ é um minimizador estritamente local de \mathbf{P}_N cujo raio de atração é não-nulo e $\hat{\eta}$ um ponto de acumulação qualquer da seqüência $\{\eta_N\}_{N \in \mathbf{N}}$, então $\hat{\eta}$ é um minimizador local de \mathbf{P} .

Através desse teorema, o ponto de mínimo do problema contínuo pode ser estimado através de um ponto de acumulação de uma seqüência, definida no espaço discreto associado. Entretanto, a simples epiconvergência do problema de aproximação não garante que pontos estacionários de \mathbf{P}_N convirjam necessariamente para pontos estacionários de \mathbf{P} , em função do nível de discretização e da definição dos conjuntos de

²⁴ Polak (1997) Definição 5.1.6-c: Seja v um espaço real munido de norma. Um ponto x^* é chamado ponto de acumulação de uma seqüência $\{x_i\}_{i \in \mathbf{N}}$ em v se existe um subconjunto infinito $\mathbf{K} \subset \mathbf{N}$ tal que, para a subseqüência $\{x_i\}_{i \in \mathbf{K}}$, $\lim_{i \rightarrow \infty, i \in \mathbf{K}} |x_i - x^*| = 0$.

vínculos²⁵. Assim, Polak (1993) utiliza as funções de otimalidade que introduzem convergência de primeira ordem e restringem os domínios \mathbf{F} e \mathbf{F}_N .

Definição 3: $\theta: \mathbf{B} \rightarrow \mathfrak{R}$ é uma *função de otimalidade* para o problema \mathbf{P} se:

- i. $\theta(\cdot)$ é pelo menos semi-contínuo inferiormente
- ii. $\theta(\eta) \leq 0$ para todo $\eta \in \mathbf{B}$.
- iii. para $\eta \in \mathbf{F}$, $\theta(\hat{\eta}) = 0$ se $\hat{\eta}$ for um minimizador local para \mathbf{P} .

e de maneira equivalente, $\theta_N: H_N \rightarrow \mathfrak{R}$ é uma função de otimalidade para \mathbf{P}_N se:

- i. $\theta_N(\cdot)$ é pelo menos semi-contínuo inferiormente
- ii. $\theta_N(\eta_N) \leq 0$ para todo $\eta_N \in H_N$.
- iii. se $\eta_N \in \mathbf{F}_N$ é um mínimo local para \mathbf{P}_N , $\theta_N(\hat{\eta}_N) = 0$

A partir da Definição 3 chega-se à formulação de aproximação consistente:

Definição 4: Os pares (\mathbf{P}_N, θ_N) na seqüência $\{(\mathbf{P}_N, \theta_N)\}_{N \in \mathbf{N}}$ são *aproximações consistentes* do par (\mathbf{P}, θ) se

- i. \mathbf{P}_N converge epigraficamente para \mathbf{P}
- ii. para qualquer seqüência $\{\eta_N\}_{N \in \mathbf{K}}$, $\mathbf{K} \subset \mathbf{N}$ com $\eta_N \in \mathbf{F}_N$ para todo $N \in \mathbf{K}$, tal que $\eta_N \rightarrow \eta$, as funções de otimalidade satisfazem a condição $\overline{\lim} \theta_N(\eta_N) \leq \theta(\eta)$

Deste modo, se η é um ponto estacionário de \mathbf{P} , $\theta(\eta) \leq 0$ (e de maneira semelhante para θ_N). Assim, se $\overline{\lim} \theta_N(\eta_N) \leq \theta(\eta)$, elimina-se a possibilidade de que pontos

²⁵ Schwartz (1996) aborda esse problema, apresentando um exemplo, no comentário ao seu Teorema 2.2.

estacionários do problema de otimização não sejam também pontos estacionários do problema original.

4.2 Definição do problema de controle ótimo (OCP)

É possível formular o problema de controle ótimo, no contexto da teoria das aproximações consistentes, seguindo o raciocínio exposto nos passos enumerados a seguir.

1. Definição das equações diferenciais que descrevem o comportamento dinâmico do problema:

$$\dot{x}(t) = h(x(t), u(t)) \quad \text{eq. 4.11}$$

$$t \in [0, 1], x(0) = \xi, x(t) \in \mathfrak{R}^n \text{ e } u(t) \in \mathfrak{R}^m, h: \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$$

2. Seja $L_2^m[0,1]$ um espaço de funções quadraticamente integráveis no intervalo $[0,1]$ em \mathfrak{R}^m , cujo produto interno²⁶ e norma são definidos respectivamente como

$$\langle u(t), v(t) \rangle_2 = \int_0^1 \langle u(t), v(t) \rangle dt \quad \text{e} \quad \|u(t)\|_2 = \left[\int_0^1 \langle u(t), u(t) \rangle dt \right]^{1/2}. \quad \text{Define-se, assim, um}$$

²⁶ Num espaço de funções contínuas de x com domínio Ω , a norma l_2 de um elemento u e o produto interno l_2 entre dois elementos u e v são dados por, respectivamente (Reddy, 1986): $\|u\|_{l_2} = \left[\int_{\Omega} |u|^2 dx \right]^{1/2}$ e $\langle u, v \rangle_{l_2} = \int_{\Omega} uv dx$. É possível também definir um espaço funcional l_2 , que corresponde ao espaço com todas as funções quadráticas u (não necessariamente contínuas) integráveis no domínio Ω , possuindo assim a propriedade $\int_{\Omega} |u|^2 dx < \infty$

espaço pré-hilbertiano²⁷ de elementos de controle $L_{\infty,2}^m[0,1]$ munido de norma e produto interno de $L_2^m[0,1]$.

3. Definição de outro espaço pré-hilbertiano de controles ótimos $H_{\infty,2} = \mathfrak{R}^n \times L_{\infty,2}^m[0,1]$
 $= (\mathfrak{R}^n \times L_{\infty,2}^m[0,1], \langle \cdot, \cdot \rangle_H, \|\cdot\|_H)$ cujos elementos são pares de condições iniciais e variáveis de controle: $\eta = (\xi, u)$. $H_{\infty,2}$ é um subespaço denso²⁸ do espaço de Hilbert $H_2 = \mathfrak{R}^n \times L_2^m[0,1]$.

4. Definição do produto interno e norma $\langle \cdot, \cdot \rangle_H, \|\cdot\|_H$ em H_2 e $H_{\infty,2}$. Considerando dois pares $\eta = (\xi, u)$ e $\eta' = (\xi', u') \in H_2$,

$$\langle \eta, \eta' \rangle_H = \langle \xi, \xi' \rangle + \langle u, u' \rangle_{L_2} \quad \text{eq. 4.12}$$

$$\|\eta\|_H^2 = \langle \eta, \eta \rangle_H = \|\xi\|^2 + \|u\|_2^2 \quad \text{eq. 4.13}$$

sendo $\langle \xi, \xi' \rangle$ o produto interno euclidiano e o produto interno l_2 $\langle u, u' \rangle_2 = \int_0^1 \langle u(t), u'(t) \rangle dt$

5. Definição de um conjunto compacto²⁹ e convexo³⁰ de vínculos de controle

²⁷Um espaço pré-hilbertiano é um espaço munido de produto interno e norma. Um espaço de Hilbert é aquele que, além disso, é completo, isto é, que toda seqüência convergente tem um ponto limite (Hönig, 1970; Reddy, 1986) dentro do próprio espaço. Isso equivale a dizer que toda seqüência de Cauchy é convergente. Seqüência de Cauchy é aquela em que a distância entre dois pontos subseqüentes pode ser sempre reduzida, ainda que tenha um ponto limite fora do espaço em que a seqüência é definida (White, 1973).

²⁸Dado um espaço métrico M , um subconjunto $A \subset M$ se diz denso em M se o fecho de A $\bar{A} = M$ ($\bar{A} = A \cup A'$, sendo A' o complemento de A). Isso significa que para todo ponto $p \in M$ existe um ponto $a \in A$ arbitrariamente próximo (Domingues, 1982). Por exemplo, o conjunto dos pontos racionais Q é denso em \mathfrak{R} , ou o conjunto de todas as funções polinomiais é denso em l_2 .

²⁹Um conjunto S num espaço real munido de norma e produto interno é compacto se for fechado (que contem as suas fronteiras) e limitado. Isso equivale a dizer que qualquer seqüência $\{x_i\}_{i=0}^{\infty} \subset S$ possui uma subseqüência convergente com um limite dentro de S , ou que tem pelo menos um ponto de acumulação dentro de S (Bazaraa, 1993, Apêndice A; Polak, 1997, Def. 5.1.13).

³⁰Um conjunto $S \subset \mathfrak{R}^n$ é dito convexo se para quaisquer x_1 e $x_2 \in S$, $\lambda \in [0,1]$, o segmento de linha $\lambda x_1 + (1-\lambda) x_2$ pertence a S . Uma função $f: S \rightarrow \mathfrak{R}$ é dita convexa em S se $f[\lambda x_1 + (1-\lambda) x_2] \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ para cada $x_1, x_2 \in S$ e cada $\lambda \in [0,1]$ (Bazaraa et al., 1993).

$$U \subset B(0, \rho_{max}) = \{u \in \mathfrak{R}^m \mid \|u\| \leq \rho_{max}\}$$

supondo que todos os controles possíveis estão contidos numa bola fechada de raio ρ_{max} .

6. Definição de um conjunto de controles admissíveis:

$$U = \{u \in L_{\infty,2}^m[0,1] \mid u(t) \in U, t \in [0,1]\}$$

e de um conjunto de pares (ξ, u) admissíveis $\mathbf{H} = \mathfrak{R}^n \times U \subset H_{\infty,2}$ contido num conjunto maior:

$$\mathbf{B} = \mathfrak{R}^n \times \{u \in L_{\infty,2}^m[0,1] \mid u(t) \in B(0, \rho_{max}), t \in [0,1]\} \subset H_{\infty,2}$$

dentro do qual são válidos os resultados das equações diferenciais.

7. Definição numa forma canônica do problema de controle ótimo com vínculos

$$\text{CP} \quad \min_{\eta \in \mathbf{H}} \{\psi_0(\eta) \mid \psi_c(\eta) \leq 0\} \quad \text{eq. 4.14}$$

sendo a função objetivo $\psi_0: \mathbf{B} \rightarrow \mathfrak{R}$, $\psi_0(\eta) = \max_{v \in \mathbf{q}_0} f^v(\eta)$ e a função de vínculos de estados

terminais $\psi_c: \mathbf{B} \rightarrow \mathfrak{R}$, $\psi_c(\eta) = \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f^v(\eta)$.

Seja ainda a v -ésima função $f^v: \mathbf{B} \rightarrow \mathfrak{R}$, $f^v(\eta) = \zeta^v(\xi, x^n(1))$, sendo $\zeta: \mathfrak{R}^n \times \mathfrak{R}^n \rightarrow \mathfrak{R}$, $\mathbf{q}_0 = \{1, 2, \dots, q_0\}$, $\mathbf{q}_c = \{1, 2, \dots, q_c\}$, q_0 e q_c inteiros positivos, (denotando o número de funções objetivo e o número de funções de vínculo, respectivamente). Sejam dados também o conjunto $\mathbf{q} = \{1, 2, \dots, q\}$, $q = q_0 + q_c$, $\mathbf{q}_c + \mathbf{q}_0 = \{1 + q_0, \dots, q_c + q_0\}$ e $x^n(1)$ correspondente à solução no tempo final do sistema dinâmico $\dot{x} = h$ com a condição inicial η . Nesta forma canônica é possível transcrever diversas classes de problemas não autônomos, com tempo final aberto, com IP integral, com e sem vínculos terminais e de controle.

8. Estabelecimento das hipóteses necessárias de definição das funções (Schwartz 1996,

Hipótese 3.1 e Teorema 3.2)

- A função h na eq. 4.11 é continuamente diferenciável e limitada, segundo as hipóteses 3.5a-c e 3.6a-c de Schwartz e Polak (1996).
- As funções ζ^v e suas derivadas com relação a x e ξ são Lipschitz contínuas³¹ em conjuntos limitados.

9. Definição de uma função de otimalidade (Schwartz e Polak, 1996, eq. 3.9) $\theta: \mathbf{B} \rightarrow \mathfrak{R}$.

Para todo $\eta, \eta' \in \mathbf{B}$ e $v \in \mathbf{q}$ pode-se definir uma aproximação quadrática para $f^v(\cdot)$ em

η

$$\tilde{f}^v(\eta, \eta') = f^v(\eta) + \langle \nabla f^v(\eta), \eta' - \eta \rangle_H + \frac{1}{2} \|\eta' - \eta\|_H^2 \quad \text{eq. 4.15}$$

Seja $\sigma > 0$. A função de otimalidade é definida como

$$\theta(\eta) = \min_{\eta \in H} \max \left\{ \max_{v \in \mathbf{q}_0} \tilde{f}^v(\eta, \eta') - \psi_0(\eta) - \sigma \psi_c(\eta)_+, \max_{v \in \mathbf{q}_c + \mathbf{q}_0} \tilde{f}^v(\eta, \eta') - \psi_c(\eta)_+ \right\} \quad \text{eq. 4.16}$$

em que $\psi_0(\eta) = \max_{v \in \mathbf{q}_0} f^v(\eta)$, $\psi_c(\eta) = \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f^v(\eta)$, $\psi_c(\eta)_+ = \max\{0, \psi_c(\eta)\}$.

Uma tentativa de interpretação do significado dessa função seria a seguinte:

supondo $\theta(\eta)=0$, isto é, que η é um mínimo pelo menos local de CP, a eq. 4.16 pode ser escrita como:

$$\theta(\eta) = \min_{\eta \in H} \max \left\{ \max_{v \in \mathbf{q}_0} \tilde{f}^v(\eta, \eta') - \max_{v \in \mathbf{q}_0} f^v(\eta) - \sigma \max \left\{ \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f^v(\eta), 0 \right\}, \max_{v \in \mathbf{q}_c + \mathbf{q}_0} \tilde{f}^v(\eta, \eta') - \max \left\{ \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f^v(\eta), 0 \right\} \right\}$$

eq. 4.17

Supondo que os vínculos são satisfeitos em η com tolerância ε , isto é,

³¹ Sejam um espaço métrico V , S um sub-espaço seu, e $T: S \rightarrow V$ um operador não-linear $u \mapsto Tu$. Se existir um escalar α tal que, para todo $u, v \in S$, $|Tu - Tv| \leq \alpha |u - v|$, se diz que o operador é *Lipschitz contínuo* (Reddy, 1986). Essa é uma condição suficiente, mas não necessária, para que as derivadas de uma função existam e sejam limitadas no domínio (Greenberg, 1978).

$$\max_{v \in Q_c + Q_0} f^v(\eta) \approx \varepsilon$$

a eq. 4.17 pode ser interpretada, não formalmente, como

$$\theta(\eta) = \min_{\eta \in H} \max \left\{ \text{erro aprox. linear da f.obj.} - \sigma\varepsilon, \max_{v \in Q_c + Q_0} \tilde{f}^v(\eta, \eta') - \varepsilon \right\}$$

$$\theta(\eta) = \min_{\eta \in H} \max \{ \text{erro aprox. linear da f.obj.} - \sigma\varepsilon, \text{erro aprox. linear da violação de vínculos} \}$$

A expressão acima mostra que a função de otimalidade para CP procura o η que minimiza o máximo entre o erro devido à aproximação linear da função objetivo, ponderado pela tolerância da violação de vínculos através de um parâmetro arbitrário σ , e o erro da aproximação linear da própria violação dos vínculos. A compreensão do significado das funções de otimalidade não é trivial; a demonstração formal destas condições de otimalidade para diversas formas canônicas para problemas de controle ótimo pode ser encontrada no item 4.2 de Polak (1997).

4.3 Construção de uma família de problemas de aproximação para o problema de controle ótimo

Uma vez estabelecidos os pressupostos teóricos de consistência do problema de aproximação para o controle ótimo, o próximo passo consiste na construção de famílias de subespaços de dimensão finita de $H_{\infty,2}$. Em seguida, serão estabelecidas aproximações para as funções de custo e para o conjunto de vínculos.

4.3.1 Definição dos subespaços de controle de dimensão finita

São descritos agora os passos necessários para definição formal dos subespaços de controle de dimensão finita compatíveis com as hipóteses da Teoria das Aproximações Consistentes.

Primeira parte

1. Definição de um espaço de dimensão finita $H_N = \mathfrak{R}^n \times L_N \subset H_{\infty,2}$, onde L_N são subespaços de dimensão finita de $L_{\infty,2}^m[0,1]$, gerado a partir de funções contínuas por partes para as quais pode ser aplicado um método tipo Runge-Kutta (RK).
2. Estabelecimento das condições necessárias para gerar L_N , isto é,
 - Seleção de um método de integração RK que gere erro de integração limitado, condição para provar a epiconvergência do problema de aproximação (Schwartz, 1996).
 - Os dados usados pelo RK são uma condição inicial ξ e um conjunto de amostras de controle, onde cada amostra corresponde a um único elemento de $u \in L_N$.
3. Definição de um método RK genérico, com tamanho de passo explícito e fixo, com s estágios de integração.

Seja uma equação diferencial do tipo:

$$\dot{x}(t) = \tilde{h}(t, x(t), u(t)), \quad x(0) = \xi, \quad t \in [0, 1] \quad \text{eq. 4.18}$$

$\tilde{h} : \mathfrak{R} \times \mathfrak{R}^n \rightarrow \mathfrak{R}^n$, contínua em t e Lipschitz-contínua em x .

O método RK calcula uma solução aproximada da eq. 4.18 avaliando a seguinte equação de diferenças³²:

³² \bar{x}_k é a estimativa de $x(t_k)$. A chamada *malha de integração* ou *malha de discretização* corresponde à seqüência $\{t_k\}_{k=0}^N$.

$$\bar{x}_{k+1} = \bar{x}_k + \Delta \sum_{i=1}^s b_i K_{k,i}, \quad \bar{x}_0 = x(0) = \xi, \quad k \in \{0, 1, \dots, N-1\} \quad \text{eq. 4.19}$$

com $\Delta = 1/N$, $t_k = k/\Delta$,

$$K_{k,1} = \tilde{h}(\tau_{k,1}, \bar{x}_k)$$

$$K_{k,i} = \tilde{h}(\tau_{k,i}, \bar{x}_k + \Delta \sum_{j=1}^{i-1} a_{i,j} K_{k,j}), \quad i = 2, \dots, s \quad \text{eq. 4.20}$$

e o termo $\tau_{k,i} = t_k + c_i \Delta$ eq. 4.21

Os parâmetros $a_{i,j}$, c_i , b_i nas equações acima determinam o método RK utilizado, e podem ser organizado na chamada *matriz de Butcher* $\mathbf{A} = [c, A, b^T]$:

$$\begin{array}{c|cccc} c_1 & 0 & & & \\ c_2 & a_{2,1} & 0 & & \\ \dots & \dots & & & \\ c_s & a_{s,1} & \dots & a_{s,s-1} & 0 \\ \hline & b_1 & \dots & b_{s-1} & b_s \end{array}$$

Segundo Schwartz (1996) (Hipótese 4.1 e Observação 4.2) a condição de que para todo i

$\in \{1, 2, \dots, s\}$, $b_i > 0$, $\sum_{i=1}^s b_i = 1$ é satisfeita por todos os métodos RK convergentes. Por

exemplo, para o método Runge-Kutta de 4ª ordem (RK4), a matriz de Butcher fica:

$$\begin{array}{c|cccc}
 1 & 0 & 0 & 0 & 0 \\
 1/2 & 1/2 & 0 & 0 & 0 \\
 1/2 & 0 & 1/2 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 \\
 \hline
 & 1/6 & 1/3 & 1/3 & 1/6
 \end{array}$$

E as recursões:

$$\bar{x}_0 = x(0) = \xi$$

$$\bar{x}_{k+1} = \bar{x}_k + \Delta \left(\frac{1}{6} \mathbf{K}_{k,1} + \frac{1}{3} \mathbf{K}_{k,2} + \frac{1}{3} \mathbf{K}_{k,3} + \frac{1}{6} \mathbf{K}_{k,4} \right)$$

$$\begin{cases}
 \tau_{k,i} = t_k + c_i \Delta \\
 \tau_{k,1} = t_k \\
 \tau_{k,2} = t_k + \frac{1}{2} \Delta \\
 \tau_{k,3} = t_k + \frac{1}{2} \Delta \\
 \tau_{k,4} = t_k
 \end{cases}$$

$$\begin{cases}
 \mathbf{K}_{k,1} = \tilde{h}(t_k, \bar{x}_k) \\
 \mathbf{K}_{k,2} = \tilde{h}\left(t_k + \frac{1}{2} \Delta, \bar{x}_k + \frac{\Delta}{2} \mathbf{K}_{k,1}\right) \\
 \mathbf{K}_{k,3} = \tilde{h}\left(t_k + \frac{1}{2} \Delta, \bar{x}_k + \frac{\Delta}{2} \mathbf{K}_{k,2}\right) \\
 \mathbf{K}_{k,4} = \tilde{h}(t_k, \bar{x}_k + \Delta \mathbf{K}_{k,3})
 \end{cases}$$

Considerando a equação $\tilde{h}(t, x) = h(x(t), u(t))$ discretizada, os termos $u(\tau_k)$ deverão ser definidos não apenas nos nós k ($\tau_k = t_k$) da malha de discretização, mas também

quando $\tau_k = t_k + \Delta/2$. Com isso, os valores $u(\tau_{k,i})$ podem ser expressos em termos de amostras de controle $u[\tau_{k,i}]$, isto é, como limites à esquerda $u[\tau_{k,i}] = \lim_{t \uparrow \tau_{k,i}} u(t)$. Tratando-se de $u(\cdot)$ avaliado no próprio ponto da malha de discretização, pode-se usar $u[\tau_{k,i}] = u(\tau_{k,i})$.

Para cada passo de integração k , define-se um vetor $\omega_k = (u[\tau_{k,1}], u[\tau_{k,2}], \dots, u[\tau_{k,s}])$, reescrevendo-se a recursão definida nas eq. 4.19, eq. 4.20 e eq. 4.21 em termos desses novos vetores (Schwartz 1996, eqs. 2.4.3a – d).

4. Para manter a correspondência um-a-um dos elementos do subespaço de controle L_N com cada uma das amostras de controle $u[\tau_k + c_i \Delta]$, definem-se os conjuntos

$$I = \{i_1, i_2, \dots, i_r\} = \{i \in \mathbf{s} \mid c_j \neq c_i, \forall j \in \mathbf{s}, j < i\},$$

$$I_j = \{i \in \mathbf{s} \mid c_j = c_i, i_j \in I\}, j \in \mathbf{r}$$

O parâmetro r corresponde ao número de valores diferentes das componentes c_i do vetor de Butcher (para o RK4, $\mathbf{c} = \{0, 1/2, 1/2, 1\}$, $\mathbf{r} = \{0, 1/2, 1\}$ $r=3$, $I = \{1, 2, 4\}$, $I_1 = \{1\}$, $I_2 = \{2, 3\}$, $I_3 = \{4\}$). Assim, os r tempos de amostragem distintos no intervalo $[t_k, t_{k+1}]$, $k \in \{0, 1, \dots, N-1\}$ são dados por $\tau_{k,i}$, $j \in \mathbf{r}$, $i_j \in I$. A cada tempo de amostragem corresponde uma amostra de controle $u[\tau_{k,i}] \in \mathfrak{R}^m$. Agrupando as amostras em vetores com N blocos, obtém-se:

$$\bar{\mathbf{u}} = (\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_{N-1}), \quad \bar{\mathbf{u}}_k \in \times_r \mathfrak{R}^m \quad \text{eq. 4.22}$$

$$\bar{\mathbf{u}}_k = (\bar{\mathbf{u}}_{k,1}, \dots, \bar{\mathbf{u}}_{k,r}), \quad \text{eq. 4.23}$$

$$\bar{\mathbf{u}}_{k,j} \in \mathfrak{R}^m, \quad j \in \mathbf{r}$$

Onde cada $\bar{\mathbf{u}}_{k,j}$ corresponde a uma amostra de controle $u[\tau_{k,i}]$.

5. Com o intuito de simplificar operações algébricas posteriores, a coleção de termos³³

$\bar{u} \in \times \times \mathfrak{R}^m$ pode ser substituída por uma matriz $m \times Nr$

$$[\bar{u}_{0,1} \quad \cdots \quad \bar{u}_{0,r} \quad \cdots \quad \bar{u}_{N-1,1} \quad \cdots \quad \bar{u}_{N-1,r}]$$

e o produto interno Euclidiano $\langle \bar{u}, \bar{v} \rangle_{l_2} = \sum_{N=0}^{N-1} \sum_{j=1}^r \langle \bar{u}_{k,j}, \bar{v}_{k,j} \rangle$

6. Definindo uma matriz G ($r \times s$), onde cada termo $\mathbf{1}_j$ corresponde a um vetor com comprimento igual a I_j de elementos unitários:

$$G = \begin{bmatrix} \mathbf{1}_1 & & & \\ & \mathbf{1}_1 & & \\ & & \ddots & \\ & & & \mathbf{1}_r \end{bmatrix}$$

eq. 4.24

Para o RK4, essa matriz fica:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

os vetores ω utilizados nas equações de recursão modificadas do método RK podem ser escrito como $\omega_k = \bar{u}_k G = [\bar{u}_{k,1}, \dots, \bar{u}_{k,r}] G$.

Segunda Parte

Uma vez estabelecida uma maneira conveniente de expressar a dinâmica do sistema num espaço discreto construído a partir de uma integração numérica, as variáveis de

controle num grupo de matrizes, assim como as operações de produto interno e norma, é introduzida uma representação possível de um subespaço de controle $L_N \subset L_{\infty,2}^m[0,1]$ de dimensão $N \times r \times m$ tal que a união $\cup_{N=1}^{\infty} L_N$ seja densa em $L_{\infty,2}^m[0,1]$. Schwartz apresenta duas representações de tais subespaços, uma baseada em polinômios contínuos por partes (L_N^1) e outra em funções constantes por partes (L_N^2). A seguir, será mostrada apenas a primeira representação, por tratar-se da mais genérica e que inclui funções tipo spline.

1. **1ª Representação:** Polinômios de ordem r contínuos por partes (subespaço L_N^1)

Hipótese: para todo $i \in \mathbf{s}$, $c_i \in [0, 1]$ (Schwartz, 1996, Hipótese 2.4.3)

Tendo em conta cada ponto $k \in \{0, 1, \dots, N-1\}$, é possível definir subintervalos

$$T_k^1 = [t_k, t_{k+1}) \quad \text{eq. 4.25}$$

e funções pulso

$$\prod_{N,k}^1(t) = \begin{cases} 1, & \text{se } t \in T_k^1 \\ 0 & \text{se não} \end{cases} \quad \text{eq. 4.26}$$

O subespaço de controle L_N^1 de dimensão finita é então definido como

$$L_N^1 = \left\{ \mathbf{u} \in L_2^m[0,1] \mid \mathbf{u}(t) = \sum_{k=0}^{N-1} \sum_{j=1}^r \bar{u}_{k,j} \Phi_{N,k,j}^1(t), \quad \bar{u}_{k,j} \in \mathfrak{R}^m, \quad \forall t \in [0,1] \right\} \quad \text{eq. 4.27}$$

sendo a *função base*

$$\Phi_{N,k,j}^1(t) = \phi_{N,k,j}(t) \prod_{N,k}^1(t) \quad \text{eq. 4.28}$$

$$k \in \{0, 1, \dots, N-1\}$$

O j -ésimo polinômio de Lagrange,

³³ $\times_N \mathfrak{R}$ significa um espaço gerado por N cópias do espaço \mathfrak{R} , ou $\mathfrak{R}^N \times \mathfrak{R}$.

$$\phi_{N,k,j}(t) = \prod_{\substack{l=1 \\ l \neq j}}^r \frac{(t - \tau_{k,i_l})}{(\tau_{k,i_j} - \tau_{k,i_l})} \quad \text{eq. 4.29}$$

definido para os pontos $\{\tau_{k,i_j}\}$, $j = 1, \dots, r$, $i_j \in I$, $k \in \{0, 1, \dots, N-1\}$, possui ainda a propriedade assumir valor 1 se $l = j$ ou 0 se $l \neq j$. A função $\phi_{N,k,j}(t)$ é um polinômio de ordem r que interpola as amostras de controle no intervalo $[t_k, t_{k+1}]$. Finalmente, as amostras de controle são expressas no subespaço L_N^1 como

$$u[\tau_{k,i}] = \begin{cases} u(\tau_{k,i}) & \text{se } \tau_{k,i} \in T_k^1 \\ \lim_{t \uparrow \tau_{k,i}} & \text{se } \tau_{k,i} = t_{k+1} \end{cases} \quad k \in \{0, 1, \dots, N-1\}, i \in I \quad \text{eq. 4.30}$$

2. Entretanto, as funções de base definidas acima não são ortogonais com relação a norma e produto interno l_2 . Assim, é conveniente que seja introduzido um novo espaço de coeficientes, definidos de maneira tal que o produto interno e a norma calculados nesse espaço sejam iguais ao produto interno e norma l_2 no espaço L_N^1 :

$$\bar{L}_N^1 = \left(\times_{N \text{ r}} \mathfrak{R}^m, \langle \cdot, \cdot \rangle_{\bar{L}_N^1}, \|\cdot\|_{\bar{L}_N^1} \right) \quad \text{eq. 4.31}$$

Neste novo espaço \bar{L}_N^1 , é desejável obter uma expressão sintética para o produto interno e a norma, de maneira que apresentem os mesmos resultados do produto interno e norma entre dois elementos de L_N . Isto é:

$$\langle u, v \rangle_{\bar{L}_N^1} = \langle u, v \rangle_2$$

Para determinar o produto interno e a norma nesse novo espaço, deve-se ter em conta que cada parte $\sum_{j=1}^r \bar{u}_{k,j} \Phi_{N,k,j}^1(t)$ (eq. 4.27) de $u \in L_N^1$ pode ser representada por uma série de potências $[\alpha_k] P(t-t_k)$, sendo $[\alpha_k]$ uma matriz de coeficientes $m \times r$ e $P : \mathfrak{R} \rightarrow \mathfrak{R}^r$. Seja:

$$P(t) = [1 \quad t/\Delta \quad \dots \quad (t/\Delta)^{r-1}]^T \quad \text{eq. 4.32}$$

$$P(c_{i_j} \Delta) = \begin{bmatrix} 1 & \frac{c_{i_j} \Delta}{\Delta} & \left(\frac{c_{i_j} \Delta}{\Delta}\right)^2 & \dots & \left(\frac{c_{i_j} \Delta}{\Delta}\right)^{r-1} \end{bmatrix} = \begin{bmatrix} 1 & c_{i_j} & c_{i_j}^2 & \dots & c_{i_j}^{r-1} \end{bmatrix} \quad \text{eq. 4.33}$$

Assim,

$$\bar{u}_{k,j} = [\alpha_k] P(c_{i_j} \Delta), j \in r \quad i_j \in I \quad \text{eq. 4.34}$$

e

$$\bar{u}_k = [\bar{u}_{k,1} \quad \dots \quad \bar{u}_{k,r}] = [\alpha_k] T^{-1} \quad \text{eq. 4.35}$$

em que a chamada matriz de Vandermonde T é definida de forma que:

$$T^{-1} = \begin{bmatrix} P(c_{i_1} \Delta) & P(c_{i_2} \Delta) & \dots & P(c_{i_r} \Delta) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ c_{i_1} & c_{i_2} & \dots & c_{i_r} \\ \vdots & \vdots & & \vdots \\ c_{i_1}^{r-1} & c_{i_2}^{r-1} & \dots & c_{i_r}^{r-1} \end{bmatrix}_{r \times r} \quad \text{eq. 4.36}$$

Se T^{-1} é não-singular, $[\alpha_k] = \bar{u}_k T$. Assim, para cada k,

$$u(t) = \bar{u}_k T P(t - t_k), \text{ para todo } t \in [t_k, t_{k+1}). \quad \text{eq. 4.37}$$

Com o RK4, a eq. 4.34 para $r = 1, 2, 3$ fica:

$$\bar{u}_{k,1} = [\alpha_k] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \bar{u}_{k,2} = [\alpha_k] \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \quad \bar{u}_{k,3} = [\alpha_k] \begin{bmatrix} 1 \\ 4 \\ 16 \end{bmatrix}$$

e a eq. 4.35,

$$\bar{u}_k = [\alpha_k] T^{-1} = [\alpha_k] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{bmatrix}$$

$$\text{Assim, } \mathbf{T}^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{bmatrix}$$

2. Determinação do produto interno e norma em \bar{L}_N^1 entre dois vetores³⁴ $\bar{\mathbf{u}}$ e $\bar{\mathbf{v}} \in$

$$\bar{L}_N^1, \mathbf{u} = \left(\mathbf{V}_{A,N}^1\right)^{-1}(\bar{\mathbf{u}}) \quad \text{e} \quad \mathbf{v} = \left(\mathbf{V}_{A,N}^1\right)^{-1}(\bar{\mathbf{v}}).$$

Uma vez que $\sum_{j=1}^r \bar{u}_{k,j} \Phi_{N,k,j}^1(t) = [\alpha_k] P(t-t_k)$, no intervalo $[t_k, t_{k+1})$, a eq. 4.35 permite afirmar que:

$$\mathbf{u}(t) = \bar{\mathbf{u}}_k \mathbf{T} P(t-t_k) \quad \text{eq. 4.38}$$

O produto interno entre dois elementos de \bar{L}_N^1 pode então ser expresso em termos de um somatório de produtos internos euclidianos de elementos de L_N ao longo dos intervalos entre os nós da malha de discretização:

$$\begin{aligned} \langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle_{\bar{L}_N^1} &= \langle \mathbf{u}, \mathbf{v} \rangle_2 = \sum_{k=0}^{N-1} \int_0^{\Delta} \langle \mathbf{u}(t_k + t), \mathbf{v}(t_k + t) \rangle dt \\ &= \sum_{k=0}^{N-1} \int_0^{\Delta} \langle \bar{\mathbf{u}}_k \mathbf{T} P(t), \bar{\mathbf{v}}_k \mathbf{T} P(t) \rangle dt \\ &= \Delta \sum_{k=0}^{N-1} \text{tr} \left(\bar{\mathbf{u}}_k \mathbf{T} \frac{1}{\Delta} \left(\int_0^{\Delta} P(t) P(t)^T dt \right) \mathbf{T}^T \bar{\mathbf{v}}_k^T \right) \end{aligned}$$

Com isso, o produto interno fica:

$$\langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle_{\bar{L}_N^1} = \langle \mathbf{u}, \mathbf{v} \rangle_2 = \Delta \sum_{k=0}^{N-1} \text{tr}(\bar{\mathbf{u}}_k \mathbf{M}_1 \bar{\mathbf{v}}_k^T) \quad \text{eq. 4.39}$$

\mathbf{M}_1 é uma matriz simétrica positiva-definida de dimensão $r \times r$

³⁴ $\mathbf{V}_{A,N}^1$ é um mapa $\mathbf{V}_{A,N}^1 : L_N^1 \rightarrow \times_{N,r} \mathfrak{R}^m$; $\mathbf{u} = \mathbf{V}_{A,N}^1(\bar{\mathbf{u}})$

$$\mathbf{M}_1 = \mathbf{T} \left[\frac{1}{\Delta} \int_0^{\Delta} P(t) P(t)^T dt \right] \mathbf{T}^T = \mathbf{T} \text{Hilb}(r) \mathbf{T}^T \quad \text{eq. 4.40}$$

$$\text{Hilb}(r) = \begin{bmatrix} 1 & 1/2 & 1/3 & \cdots & 1/r \\ 1/2 & 1/3 & 1/4 & \cdots & 1/(r+1) \\ 1/3 & 1/4 & 1/5 & \cdots & 1/(r+2) \\ \vdots & \vdots & \vdots & & \vdots \\ 1/r & 1/(r+1) & 1/(r+2) & \cdots & 1/(2r-1) \end{bmatrix} \quad \text{eq. 4.41}$$

Apesar de Hilb(r) e T serem mal condicionadas, \mathbf{M}_1 é bem condicionada e positiva-definida. Para facilitar o cálculo do produto interno definido acima, define-se uma matriz diagonal \mathbf{M}_N , para $\Delta_k = t_{k+1} - t_k = 1/N$

$$\mathbf{M}_N = \begin{bmatrix} \Delta_0 \mathbf{M}_1 & & & & \\ & \Delta_1 \mathbf{M}_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \Delta_{N-1} \mathbf{M}_1 \end{bmatrix} \quad \text{eq. 4.42}$$

E finalmente³⁵,

$$\langle \bar{\mathbf{u}}, \bar{\mathbf{v}} \rangle_{\bar{L}_N} = \langle \bar{\mathbf{u}} \mathbf{M}_N, \bar{\mathbf{v}} \rangle_2 = \text{tr}(\bar{\mathbf{u}} \mathbf{M}_N \bar{\mathbf{v}}^T) \quad \text{eq. 4.43}$$

4.3.2 Definição dos problemas de aproximação

Uma vez determinado os subespaços³⁶ L_N e \bar{L}_N , bem como as operações algébricas de produto interno e norma neles definidos, deseja-se estender os resultados para os conjuntos $H_N = \mathfrak{R}^n \times L_N$ e $\bar{H}_N = \mathfrak{R}^n \times \bar{L}_N$. Como $H_N \subset H_{\infty,2}$, o produto interno

³⁵ A norma nesse novo espaço é dada simplesmente calculando-se o produto interno com $\bar{\mathbf{v}} = \bar{\mathbf{u}}$.

³⁶ A partir deste momento o sub-espaço L_N^1 passa a ser tratado por L_N apenas.

$\langle \cdot, \cdot \rangle_{H_N}$ e a norma $\|\cdot\|_{H_N}$ são os mesmos da eq. 4.12 e da eq. 4.13. No caso do espaço de coeficientes, pares $\bar{\eta} = (\xi, \bar{u})$ e $\bar{\eta}' = (\xi', \bar{u}')$ fornecem produto interno $\langle \bar{\eta}, \bar{\eta}' \rangle_{\bar{H}_N} = \langle \xi, \xi' \rangle + \langle \bar{u}, \bar{u}' \rangle_{\bar{L}_N}$. Pode-se ainda, através de um mapa $W_{A,N} : H_N \rightarrow \bar{H}_N$, $W_{A,N}(\eta) = (\xi, V_{A,N}(u))$, relacionar esses dois espaços.

Outro passo necessário para chegar numa formulação genérica do problema de aproximação para os OCPs é a definição dos conjuntos de vínculos de controle. Seja $U \subset B(0, \rho_{\max})$, $U = \{u \in \mathfrak{R}^m \mid |u| \leq \rho_{\max}\}$ um conjunto compacto e convexo de vínculos de controle e $\mathbf{U} = \{u \in L_{\infty,2}^m[0,1] \mid u(t) \in U, t \in [0, 1]\}$ um conjunto de controles admissíveis. O conjunto de vínculos do problema de aproximação por polinômios contínuos por partes, para³⁷ $\mathbf{k}_U < \infty$, pode ser definido como:

$$\bar{\mathbf{U}}_N^1 = \left\{ \bar{u} \in \bar{L}_N^1 \mid \bar{u}_{k,j} \in U, j \in \mathbf{r}, \left| \bar{u}_k \mathbf{T}_j \right|_{\infty} \leq \frac{\Delta}{(j-1)(k-1)} \mathbf{k}_U, j=2, \dots, \mathbf{r}, \forall k \in \{0, 1, \dots, N-1\} \right\}$$

eq. 4.44

onde \mathbf{T}_j é j -ésima coluna da matriz \mathbf{T} (eq. 4.36), $\Delta=1/N$.

Define-se então o conjunto de vínculos para o problema de aproximação no espaço de funções (polinomiais contínuas por partes):

$$\mathbf{H}_N = \mathfrak{R}^n \times V_{A,N}^{-1}(\bar{\mathbf{u}}_N) \subset H_N \quad \text{eq. 4.45}$$

que no espaço de coeficientes fica

$$\bar{\mathbf{H}}_N = \mathfrak{R}^n \times \bar{\mathbf{u}}_N \subset \bar{H}_N \quad \text{eq. 4.46}$$

e os problemas de aproximação nesses espaços formulados, respectivamente, como:

³⁷ A condição $\mathbf{k}_U < \infty$ foi introduzida para assegurar a continuidade Lipschitz dos polinômios de base no intervalo $[t_k, t_{k+1})$, implicando que a precisão do integrador RK aumenta pelo menos linearmente com a diminuição de Δ (Schwartz, 1996, Observação 4.9).

$$\text{CP}_N \quad \min_{\eta \in \mathbf{H}_N} \{ \psi_{0,N}(\eta) \mid \psi_{c,N}(\eta) \leq 0 \} \quad \text{eq. 4.47}$$

em que $\psi_{0,N}(\eta) = \max_{v \in \mathbf{q}_0} f_N^v(\eta)$, $\psi_{c,N}(\eta) = \max_{v \in \mathbf{q}_c + \mathbf{q}_0} f_N^v(\eta)$ e $f_N^v : \mathbf{H}_N \rightarrow \mathfrak{R}$, $f_N^v = \zeta^v(\xi, \bar{x}_N^\eta)$, $v \in \mathbf{q}$, $\eta = (\xi, u)$.

$$\overline{\text{CP}}_N \quad \min_{\bar{\eta} \in \overline{\mathbf{H}}_N} \{ \bar{\psi}_{0,N}(\bar{\eta}) \mid \bar{\psi}_{c,N}(\bar{\eta}) \leq 0 \} \quad \text{eq. 4.48}$$

em que $\bar{\psi}_{0,N}(\bar{\eta}) = \max_{v \in \mathbf{q}_0} \bar{f}_N^v(\bar{\eta})$, $\bar{\psi}_{c,N}(\bar{\eta}) = \max_{v \in \mathbf{q}_c + \mathbf{q}_0} \bar{f}_N^v(\bar{\eta})$ e $\bar{f}_N^v : \overline{\mathbf{H}}_N \rightarrow \mathfrak{R}$, $\bar{f}_N^v = \zeta^v(\xi, \bar{x}_N^{\bar{\eta}})$, $v \in \mathbf{q}$, $\bar{\eta} = (\xi, \bar{u}) = W_{A,N}(\eta)$.

Supondo que para todo $\eta \in \mathbf{H}$ tal que $\psi_c(\eta) \leq 0$, existe uma seqüência $\{\eta_i\}_{i=1}^\infty$ tal que $\eta_i \in \mathbf{H}$, $\psi_c(\eta_i) < 0$ e $\eta_i \rightarrow \eta$ quando $i \rightarrow \infty$, Schwartz (1996, item 2.4.3) mostra que, mediante certas condições, os problemas $\{\overline{\text{CP}}_N\}_{N \in \mathbf{N}}$ convergem epigraficamente para o problema CP quando $N \rightarrow \infty$.

4.4 Cálculo de gradientes e consistência das funções de otimalidade do problema de aproximação

Os gradientes das funções de custo podem ser calculados através do Teorema 2.5.1 de Schwartz (1996)³⁸. Para $f_N^v(\eta) : \mathbf{H}_N \rightarrow \mathfrak{R}$ e cada $v \in \mathbf{q}$, $\nabla f_N^v : \mathbf{H}_N \rightarrow \mathbf{H}_N$, o gradiente da função de custo é calculada como:

$$\nabla f_N^v(\eta) = (\nabla_\xi f_N^v(\eta), \nabla_u f_N^v(\eta)) = (d_\xi f_N^v(\eta), V_{A,N}^{-1} (d_{\bar{u}} \bar{f}_N^v(\eta)) M_N^{-1}) \quad \text{eq. 4.49}$$

em que

³⁸ No *software* RIOTS Schwartz utilizou, alternativamente, um conjunto de fórmulas determinadas por Hager (1976) para o cálculo dos gradientes (ver Schwartz, 1996, pg. 195).

$$d_{\xi} \bar{f}_N^v(\eta) = \nabla_{\xi} \zeta(\eta, \bar{x}_N^{\bar{\eta}}) + \bar{p}_0^{v, \bar{\eta}} \quad \text{eq. 4.50}$$

$$d_{\bar{u}} \bar{f}_N^v(\eta)_{k,j} = F_{w_j}(\bar{x}_k^{\bar{\eta}}, \bar{u}_k)^T \bar{p}_{k+1}^{v, \bar{\eta}}, \quad k \in \{0, 1, \dots, N-1\}, \quad j \in \mathbf{r}. \quad \text{eq. 4.51}$$

$\bar{p}_k^{v, \bar{\eta}}$ é determinado pela equação adjunta

$$\bar{p}_k^v = F_x(\bar{x}_k^{\bar{\eta}}, \bar{u}_k)^T \bar{p}_{k+1}^v \quad \bar{P}_N^v = \zeta_x^v(\xi, \bar{x}_N^{\bar{\eta}}), \quad k \in \{0, 1, \dots, N-1\} \quad \text{eq. 4.52}$$

F_x e F_{w_j} são as derivadas de $F(x, w)$ com relação a x e ao j -ésimo componente de w_j (ver equação 2.5.3a de Schwartz, 1996). A demonstração deste teorema pode estudada através do Corolário 5.6.9 de Polak (1997), para o problema equivalente contínuo. Sendo $f^v = \zeta^v(\xi, \bar{x}^{\bar{\eta}})$ o valor da função de custo na qual foi aplicado o vetor de controle u e a condição inicial ξ , Polak define uma função de custo aumentada, escrevendo-a entretanto com a parte original integrada (ζ^v) e as equações de vínculo dinâmico ponderadas com multiplicadores de Lagrange dentro da integral. Isto é:

$$f^v = \zeta^v(\xi, \bar{x}^{\bar{\eta}}) + \int_0^t \langle p(s), h(x^{\bar{\eta}}(s), u(s)) - \dot{x}^{\bar{\eta}}(s) \rangle ds \quad \text{eq. 4.53}$$

O Hamiltoniano do sistema dinâmico é definido como:

$$H(x, u, p) = \langle p, h(x, u) \rangle \quad \text{eq. 4.54}$$

Expandindo por partes a integral da eq. 4.53 e calculando a derivada direcional em relação a $\eta = (\xi, u)$ desta nova $f^v(\eta)$, aplicando as transformações de coordenadas mostradas na eq. 4.49, chega-se em fórmulas análogas às eq. 4.50 e eq. 4.51. Para o caso discreto, a demonstração pode ser encontrada no Teorema 5.6.19 de Polak (1997). A

equação adjunta eq. 4.52 e sua condição inicial representam a equação de Euler-Lagrange para os multiplicadores de Lagrange do cálculo variacional (Bryson e Ho, 1975, sec. 2.3)³⁹

A direção de procura do mínimo, dada por $-\nabla f_N^v(\eta)$ calculado com a eq. 4.49, fornece a direção de procura adequada no espaço funcional H_N , evitando problemas de mau condicionamento numérico. Estes gradientes são calculados a partir de funções objetivo aumentadas com os vínculos dinâmicos e multiplicadores de Lagrange.

Schwartz (1996, item 2.5.1) apresenta versões válidas em \mathbf{H}_N e em $\overline{\mathbf{H}}_N$ da função de otimalidade apresentada na eq. 4.16. Em seguida, no item 2.5.2, demonstra que a nova função de otimalidade $\theta_N(\eta)$ converge para a função de otimalidade do problema original, constituindo os pares (CP_N, θ_N) aproximações consistentes de (CP, θ) , independentemente do espaço funcional ser expresso numa representação de funções contínuas por partes ou constantes por partes.

4.5 Determinação de uma base ortonormal de coordenadas para o espaço de coeficientes

Para encontrar a solução de \overline{CP}_N é necessário lançar mão de algum método de otimização para cálculo do mínimo da função de custo respeitando os vínculos estabelecidos. Entretanto, os métodos usuais são formulados para espaços Euclidianos, sendo assim necessário definir uma mudança de coordenadas que transforme o espaço de coeficientes num espaço Euclidiano ortonormal.

³⁹ Para a otimalidade, deve-se acrescentar a esta equação a equação de controle. Ambas podem ser obtidas do "Princípio do Máximo de Pontryagin", que estabelece que o controle que minimiza a função de custo torna máxima a Hamiltoniana. Isso implica que o gradiente da Hamiltoniana em função do controle deve ser nulo e a Hessiana definida negativa.

O gradiente $\nabla_{\mathbf{u}} f_N^v(\boldsymbol{\eta})$ pode ser calculado segundo a eq. 4.49. $\nabla_{\mathbf{u}} \bar{f}_N^v(\boldsymbol{\eta}) = \mathbf{V}_{A,N}(\nabla_{\mathbf{u}} f_N^v(\boldsymbol{\eta}))$, com a propriedade de que

$$\left\langle \nabla_{\mathbf{u}} f_N^v(\boldsymbol{\eta}), \delta \mathbf{u}_N \right\rangle_2 = \left\langle \nabla_{\mathbf{u}} \bar{f}_N^v(\bar{\boldsymbol{\eta}}), \delta \bar{\mathbf{u}} \right\rangle_{\bar{L}_N} = \left\langle \mathbf{d}_{\mathbf{u}} \bar{f}_N^v(\boldsymbol{\eta}), \delta \bar{\mathbf{u}} \right\rangle_{1_2} \quad \text{eq. 4.55}$$

$$\delta \mathbf{u} \in \mathbf{H}_N \text{ e } \delta \bar{\mathbf{u}} = \mathbf{V}_{A,N}(\delta \mathbf{u})$$

Considerando a introdução de um novo espaço de coeficientes

$\tilde{L}_N = \times_{N,r} \times \mathfrak{R}^m$ munido de norma e produto interno 1_2 e uma transformação $Q: \bar{L}_N \rightarrow \tilde{L}_N$

$$\tilde{\mathbf{u}} = Q(\bar{\mathbf{u}}) = \bar{\mathbf{u}} \mathbf{M}_N^{1/2} \quad \text{eq. 4.56}$$

Sendo $\tilde{\boldsymbol{\eta}} = (\xi, \tilde{\mathbf{u}})$ e para todo $v \in \mathbf{q}$, $\tilde{f}_N^v: \mathfrak{R}^n \times \tilde{L}_N \rightarrow \mathfrak{R}$,

$$\tilde{f}_N^v(\tilde{\boldsymbol{\eta}}) = \bar{f}_N^v((\xi, \tilde{\mathbf{u}} \mathbf{M}_N^{-1/2})) \quad \text{eq. 4.57}$$

Assim, sendo $\bar{\boldsymbol{\eta}} = (\xi, Q^{-1}(\tilde{\mathbf{u}}))$,

$$\nabla_{\bar{\mathbf{u}}} \tilde{f}_N^v(\tilde{\boldsymbol{\eta}}) = Q^{-1}(\nabla_{\bar{\mathbf{u}}} \bar{f}_N^v(\bar{\boldsymbol{\eta}})) = \mathbf{d}_{\bar{\mathbf{u}}} \bar{f}_N^v(\boldsymbol{\eta}) \mathbf{M}_N^{-1/2} \quad \text{eq. 4.58}$$

valendo a propriedade, com $\delta \tilde{\mathbf{u}} = Q(\delta \bar{\mathbf{u}})$,

$$\left\langle \nabla_{\bar{\mathbf{u}}} \tilde{f}_N^v(\tilde{\boldsymbol{\eta}}), \delta \tilde{\mathbf{u}} \right\rangle_{1_2} = \left\langle \nabla_{\bar{\mathbf{u}}} \bar{f}_N^v(\bar{\boldsymbol{\eta}}), \delta \bar{\mathbf{u}} \right\rangle_{\bar{L}_N} = \left\langle \nabla_{\mathbf{u}} f_N^v(\boldsymbol{\eta}), \delta \mathbf{u} \right\rangle_2 \quad \text{eq. 4.59}$$

Ou seja, a transformação Q cria uma base ortonormal para o espaço de coeficientes, já que o resultado do produto interno em \bar{L}_N passa a ser igual ao do produto interno 1_2 . Schwartz (1996, item 2.6) mostra para um problema de minimização linear quadrática que, a partir da introdução dessa transformação de coordenadas o número de iterações necessárias para atingir o mínimo cai da ordem de 5 vezes em relação ao problema original (idem, Tabela 6.1 e 6.2); através da Proposição 6.2, o autor introduz

uma regra estabelecendo que, se o método de integração RK é de 4ª ordem ($O(\Delta)^4$), o erro na representação do controle $|V_{A,N}(u^*) - V_{A,N}(u_N^*)|$ será da ordem de $(\Delta)^3$. Deste modo, duplicando o número de pontos da malha de discretização, pode-se obter uma redução de 8 vezes no erro de representação do controle. Deve-se ter em conta, ainda, que os vínculos de controle são também afetados pela transformação Q, de maneira que $\bar{u} = \tilde{u}(M_N^{-1/2})$. Com isso, os novos vínculos de controle serão combinações lineares de $\tilde{u}_{k,j}$, $j \in \mathbf{r}$ (Schwartz, 1996, Observação 2.6.1)

4.6 Definição dos problemas de aproximação baseados em splines

Uma possível classe de funções contínuas por partes para a construção de um espaço de coeficientes de controle de dimensão finita é a de B-splines. Para construir tal espaço, considere-se que, para $r \in \mathbf{N}$, $r \geq 1$,

$$L_N^{(r)} = \left\{ u \in L_{\infty,2}^m[0,1] \mid u(t) = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t), \quad t \in [0,1] \right\} \quad \text{eq. 4.60}$$

em que

$$\alpha_k \in \mathfrak{R}^m$$

$$\phi_k : [0,1] \rightarrow \mathfrak{R}, \quad \phi_k(t) = B_{k,r,t_N}(t)$$

r é a ordem (grau+1) dos polinômios que compõe os pedaços da spline

t_N é a seqüência de nós sobre os quais as splines são definidas, podendo ser uma seqüência:

$$\text{Uniforme:} \quad t_N = \left\{ \frac{k}{N} \right\}_{k=-r+1}^{N+r-1} \quad \text{eq. 4.61}$$

Geral:
$$t_N = \{t_k\}_{k=-r+1}^{N+r-1} \quad \text{eq. 4.62}$$

Dada uma seqüência de nós, as B-splines constituem uma base para um espaço de dimensão $N+r-1$ de splines de ordem r , $r-2$ vezes continuamente diferenciáveis, com pontos de quebra definidos numa seqüência $\{t_k\}_{k=0}^N$.

As amostras de controle $u[\tau_{k,j}]$, $k = \{0, 1, \dots, N-1\}$, $j \in \mathbf{r}$ (ordem do RK, neste caso) podem ser relacionados com os coeficientes spline como

$$u[\tau_{k,j}] = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(\tau_{k,j}) \quad \text{eq. 4.63}$$

Com isso, as splines podem então ser escritas em termos da recursão

$$B_{k,r+1,t_{N(t)}} = \frac{t - t_{k-r-1}}{t_{k-1} - t_{k-r-1}} B_{k-1,r,t_{N(t)}} + \frac{t_k - t}{t_k - t_{k-r}} B_{k,r,t_{N(t)}} \quad r \geq 1$$

$$B_{k,1,t_{N(t)}} = \begin{cases} 1 & \text{se } t_{k-1} \leq t \leq t_k \\ 0 & \text{senão} \end{cases} \quad \text{eq. 4.64}$$

tendo em conta que as splines utilizadas são normalizadas

$$\sum_{k=1}^{N+r-1} B_{k,r,t_{N(t)}} = 1 \quad \text{para todo } t \in [0,1] \quad \text{eq. 4.65}$$

O próximo passo é a definição dos conjuntos de vínculos de controle para o novo problema de aproximação, isto é

$$U_N^{(r)} = \{u \in L_N^{(r)} \mid \alpha_k \in U, k = 1, \dots, N+r-1\} \quad \text{eq. 4.66}$$

onde o conjunto U é definido de modo que as variáveis de controle são limitadas

$$U = \{v = [v^1, \dots, v^m]^T \in \mathfrak{R}^m \mid -\infty < a_i \leq v^i \leq b_i < \infty, i = 1, \dots, m\} \quad \text{eq. 4.67}$$

Com essa nova base de funções para a representação do controle, chega-se ao seguinte problema de aproximação⁴⁰:

$$CP_N \quad \min_{u \in U_N^{(r)}} \{ \psi_{0,N}(u) \mid \psi_{c,N}(u) \leq 0 \} \quad \text{eq. 4.68}$$

em que as variáveis de decisão são os coeficientes spline α_k , $k = 1, \dots, N + r - 1$, em número significativamente inferior à formulação no espaço L_N^1 genérico, onde era necessário definir as Nr amostras de controle $u[\tau_{k,j}]$, $k = 0, \dots, N-1$; $j = 1, \dots, r$. Uma vez definido o problema de aproximação, pode-se demonstrar sua epiconvergência para CP (Schwartz, 1996, Teorema 2.7.5).

É necessário ainda calcular gradientes das funções de custo e vínculo, bem como demonstrar que os as novas funções de otimalidade satisfazem a Definição 3.

Seja um espaço de coeficientes spline

$$\bar{L}_N^{(r)} = \left(\times_{N+r-1} \mathfrak{R}, \quad \langle \cdot, \cdot \rangle_{\bar{L}_N^{(r)}} \quad \left\| \cdot \right\|_{\bar{L}_N^{(r)}} \right). \quad \text{eq. 4.69}$$

Para expressar as variáveis de controle $u = \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t) \in L_N^{(r)}$ neste novo espaço de coeficientes, designados por $\bar{\alpha}_k = \{ \alpha_k \}_{k=1}^{N+r-1} \in \bar{L}_N^{(r)}$, $\alpha_k \in \mathfrak{R}^m$, define-se um mapa $S_{N,r} = L_N^{(r)} \rightarrow \bar{L}_N^{(r)}$ entre os elementos u e $\bar{\alpha}_k$.

É possível agora definir o produto interno e a norma. Dados α e $\beta \in \bar{L}_N^{(r)}$, $u = S_{N,r}^{-1}(\alpha)$ e $v = S_{N,r}^{-1}(\beta)$,

$$\begin{aligned} \langle \alpha, \beta \rangle_{\bar{L}_N^{(r)}} &= \langle u, v \rangle_{L_2} = \int_0^1 \left\langle \sum_{k=1}^{N+r-1} \alpha_k \phi_k(t), \sum_{l=1}^{N+r-1} \beta_l \phi_l(t) \right\rangle dt = \\ &= \sum_{k=1}^{N+r-1} \sum_{l=1}^{N+r-1} \langle \alpha_k, \beta_l \rangle \int_0^1 \phi_k(t) \phi_l(t) dt = \langle \alpha M_\alpha, \beta \rangle_{L_2} \end{aligned} \quad \text{eq. 4.70}$$

⁴⁰ Neste item, o par $\eta=(\xi,u)$ será substituído pela variável de controle u apenas.

Os termos k, l da matriz M_α de dimensão $(N+r-1 \times N+r-1)$ são

$$M_\alpha = \int_0^1 \phi_k(t)\phi_l(t) dt \quad \text{eq. 4.71}$$

Por outro lado, como $L_N^{(r)} \subset L_N^1$, é possível propor um mapa composto capaz de determinar as amostras de controle a partir dos coeficientes spline $\alpha = S_{N,r}(u)$

$$\bar{u} = V_{A,N} \circ S_{N,r}^{-1}(\alpha) = \alpha \Phi_{A,N}^T \quad \text{eq. 4.72}$$

sendo $\Phi_{A,N}^T$ é uma matriz de dimensão $Nr \times N+r-1$ cujo termo $(lr+j,k)$, $l=0, \dots, N-1$, $j=1, \dots, r$, $k=1, \dots, N+r-1$, é $\phi_k(\tau_{l,i,j})$, $i_j \in I$. Com esses elementos, pode-se propor uma outra maneira de se calcular M_α :

$$\begin{aligned} \langle u, v \rangle_{l_2} &= \langle S_{N,r}(u)M_\alpha, S_{N,r}(v) \rangle_{l_2} = \langle V_{A,N}(u)M_N, V_{A,N}(v) \rangle_{l_2} = \\ &= \langle V_{A,N} \circ S_{N,r}^{-1} \circ S_{N,r}(u)M_N, V_{A,N} \circ S_{N,r}^{-1} \circ S_{N,r}(v) \rangle_{l_2} = \langle S_{N,r}(u)\Phi_{A,N}^T M_N, S_{N,r}(v)\Phi_{A,N}^T \rangle_{l_2} \end{aligned} \quad \text{eq. 4.73}$$

Pós-multiplicando o produto obtido acima por $\Phi_{A,N}$, de maneira que o segundo termo fique igual a β , e comparando-o com a eq. 4.70.

$$M_\alpha = \Phi_{A,N}^T M_N \Phi_{A,N} \quad \text{eq. 4.74}$$

4.7 Cálculo dos gradientes das funções de custo e de vínculos no espaço de funções spline

Seja a derivada de $\bar{f}_N(u)$ em relação a α

$$d_\alpha \bar{f}_N(u) = \left(\frac{d}{d_{\alpha_1}} \bar{f}_N(S_{N,r}(u)) \quad \dots \quad \frac{d}{d_{\alpha_{N+p-1}}} \bar{f}_N(S_{N,r}(u)) \right) \quad \text{eq. 4.75}$$

O diferencial de $f_N: L_N^{(r)} \rightarrow \mathfrak{R}$ pode ser representado através de um operador linear

$$\langle \nabla_u f_N(u), \delta u \rangle = \langle S_{N,r} (\nabla_u f_N(u)) M_\alpha, \delta \alpha \rangle_{l_2} = \langle d_\alpha \bar{f}_N(u), \delta \alpha \rangle_{l_2} \quad \text{eq. 4.76}$$

Aplicando a regra da cadeia⁴¹ a $d_\alpha \bar{f}_N(u)$, com a eq. 4.72,

$$d_\alpha \bar{f}_N(u) = d_{\bar{u}} \bar{f}_N(u) \frac{d\bar{u}}{d\alpha} = d_{\bar{u}} \bar{f}_N(u) \Phi_{A,N} \quad \text{eq. 4.77}$$

Da eq. 4.76, calcula-se o gradiente de f_N em relação a u

$$\nabla_u f_N(u) = S_{N,r}^{-1} (d_{\bar{u}} \bar{f}_N(u) \Phi_{A,N} M_\alpha^{-1}) \quad \text{eq. 4.78}$$

No Teorema 2.7.6, Schwartz (1996) mostra que para $N = \{2^n\}_{n=1}^\infty$, com função de otimalidade θ_N definida na sua equação 2.5.8a, os pares (CP_N, θ_N) , $N \in \mathbb{N}$, constituem aproximações consistentes do problema original (CP, θ) . O autor apresenta ainda no item 2.6 exemplos de funções base de splines e de cálculos da matriz M_α .

4.8 Transformação de coordenadas com representações por splines

Com o objetivo de encontrar uma base ortogonal para o subespaço de controles representado por splines, é introduzida uma transformação de coordenadas

$$\tilde{\alpha} = \alpha M_\alpha^{1/2} \quad \text{eq. 4.79}$$

$$\tilde{f}_N(\tilde{\alpha}) = \bar{f}_N(\tilde{\alpha} M_\alpha^{-1/2}) \quad \text{eq. 4.80}$$

O gradiente é calculado como (lembrando que $u = S_{N,r}^{-1}(\tilde{\alpha} M_\alpha^{-1/2})$)

$$\nabla_{\tilde{\alpha}} f_N(u) = \frac{(\tilde{f}_N(\tilde{\alpha}))}{d\alpha} \frac{d\alpha}{d\tilde{\alpha}} = d_\alpha \bar{f}_N(\tilde{\alpha} M_\alpha^{-1/2}) M_\alpha^{-1/2} \quad \text{eq. 4.81}$$

Resta ainda a necessidade de uma fórmula adequada para o cálculo de $M_\alpha^{1/2}$. Uma possibilidade consiste na utilização da decomposição de Schur, transformando uma matriz simétrica em um produto de matrizes unitárias U e uma matriz diagonal D : $M_\alpha = UDU^T$ e

⁴¹ Para uma transcrição mais explícita desta fórmula, ver Schwartz (1996, pg. 195).

$M_\alpha^{\pm 1/2} = UD^{\pm 1/2} U^T$. Esse passo pode ser evitado se a matriz M_α calculada for diagonal. Utilizando a eq. 4.74, Schwartz (1996) propõe um método que utiliza a matriz M_N determinada a partir da representação L_N^2 dos subespaços funcionais de dimensão finita, isto é, de funções constantes por partes, não mostradas nesse texto. Segundo testes preliminares (idem, Tabela 2.7.1), essa metodologia mostrou-se a mais econômica do ponto de vista computacional quando utilizava-se um método de integração RK de 2ª ordem em problemas com vínculos de controle.

4.9 Formulação do problema discreto de controle ótimo e equivalência com o problema de programação matemática

Uma vez estabelecida a estratégia específica introduzida por Schwartz para a discretização do sistema dinâmico e da representação do subespaço de controle, é necessário relacioná-la com o problema genérico da equivalência entre o problema de controle ótimo e o de programação matemática. Esta equivalência é um dos principais pontos de partida da teoria de controle ótimo segundo a formulação de Canon, Polak e seus colaboradores. Desta maneira, qualquer problema de controle ótimo pode, a princípio, ser reduzido a um problema de otimização discreta, solucionável por diversos tipos de algoritmos. A seguir, será mostrada a passagem do problema de controle ótimo contínuo para o problema discreto e, a seguir, do problema discreto para o problema de programação matemática. Por razões didáticas, o texto a seguir utiliza a abordagem introduzida por Canon et al. (1970) e Polak (1971), para o controle discretizado por funções constantes por partes e para a dinâmica integrada pelo método de Euler. Em seguida, a formulação será transcrita de acordo com a discretização por funções spline e integrador RK.

Seja um problema de controle ótimo contínuo minimizar a função de custo

$$f^v(u) = g_o^v(x(b)) + \int_a^b l_o^v(t, x, u) dt \quad \text{eq. 4.82}$$

sujeita a:

vínculos dinâmicos do tipo $\dot{x} = h(t, x, u)$, $x(a) = \xi$, $t \in [a, b]$

vínculos de controle $s(u(t)) \leq 0$

vínculos de desigualdade nas trajetórias $l_{ii}^v(t, x(t)) \leq 0$, $v \in \mathbf{q}_{ii}$, $t \in [a, b]$

vínculos de igualdade nas trajetórias $l_{ie}^v(t, x(t)) = 0$, $v \in \mathbf{q}_{ie}$, $t \in [a, b]$

Além disso, $x(t) \in \mathfrak{R}^n$, $u(t) \in \mathfrak{R}^m$ e as funções $h: \mathfrak{R} \times \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$, $l_o: \mathfrak{R} \times \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}$ são continuamente diferenciáveis em x e u ; $g_o: \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $l: \mathfrak{R} \times \mathfrak{R}^n \rightarrow \mathfrak{R}$ continuamente diferenciáveis em x ; $s: \mathfrak{R}^m \rightarrow \mathfrak{R}^m$ continuamente diferenciável em u . h , l_o , suas derivadas em relação a x e a u , l e suas derivadas em relação a x são contínuas por partes em t .

A partir do problema definido acima, o problema de controle ótimo discreto pode ser formulado restringindo as funções de controle $u(\cdot)$ a funções constantes por partes com $N-1$ nós equidistantes. Seja o tamanho de passo $T=(b-a)/N$ e $a=0$,

$$u(t) = u_i \quad \text{para } t \in [iT, (i+1)T), \quad i = 0, 1, \dots, N-1 \quad \text{eq. 4.83}$$

O vínculo de controle fica assim

$$s_i(u_i) = s(u_i) \leq 0 \quad \text{para } i = 0, 1, \dots, N-1 \quad \text{eq. 4.84}$$

E os vínculos de trajetória

$$l_{ie_i}^v(x_i, a + Ti) = 0, \quad v \in \mathbf{q}_{ie}, \quad t \in [a, b] \quad \text{eq. 4.85}$$

$$l_{ii_i}^v(x_i, a + Ti) \leq 0, \quad v \in \mathbf{q}_{ii}, \quad t \in [a, b]$$

O problema discreto pode ser formulado como o de minimizar o somatório

$$\sum_{i=0}^{N-1} l_{o_i}(x_i, u_i) + g_o(x_N) \quad \text{eq. 4.86}$$

sujeito a equações a diferenças, que representam o sistema dinâmico discreto

$$x_{i+1} - x_i = h_i(x_i, u_i), i = 0, 1, \dots, N-1 \quad \text{eq. 4.87}$$

e aos vínculos eq. 4.84 e eq. 4.85, de maneira que possa ser encontrada uma seqüência de controles $U = \{u_0, u_1, \dots, u_{N-1}\}$ e uma trajetória correspondente $X = \{x_0, x_1, \dots, x_N\}$.

Para calcular l_{o_i} da eq. 4.86 e h_i da eq. 4.87, considere-se $x_i(t)$ para $i = 0, 1, \dots, N-1$ a solução do sistema dinâmico h correspondente a $u(t) = u_i$, para o intervalo de tempo $t \in [iT, (i+1)T)$ satisfazendo $x(iT) = x_i$. Além disso, para $i = 0, 1, \dots, N-1$, $x_{i+1} = x_i((i+1)T)$. Assim,

$$x_{i+1} - x_i = \int_{iT}^{(i+1)T} h(x_i(t), u_i, t) dt \quad \text{eq. 4.88}$$

O que permite definir

$$h_i(x_i, u_i) = \int_{iT}^{(i+1)T} h(x_i(t), u_i, t) dt \quad \text{eq. 4.89}$$

$$l_{o_i}(x_i, u_i) = \int_{iT}^{(i+1)T} l_o(x_i(t), u_i, t) dt \quad \text{eq. 4.90}$$

Uma vez apresentado o problema discreto de controle ótimo, Polak (1971) introduz duas transcrições deste problema em termos de um problema de otimização do tipo:

Dadas funções continuamente diferenciáveis $l_o: \mathfrak{R}^n \rightarrow \mathfrak{R}$, $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ e $r: \mathfrak{R}^n \rightarrow \mathfrak{R}^l$ (l é o número de vínculos de estados de igualdade), o problema de otimização consiste em encontrar um vetor \hat{z} dentro do conjunto $\Omega = \{z | f(z) \leq 0, r(z) = 0\}$ tal que para todo $z \in \Omega$, $l_o(\hat{z}) \leq l_o(z)$. Este problema pode ser escrito como:

$$\min\{l_o(z) | f(z) \leq 0, r(z) = 0\} \quad \text{eq. 4.91}$$

Na primeira transcrição apresentada por Polak (1971, item 1.1.13), o vetor z é escrito em termos das variáveis de estado e dos controles discretizados, o que implica num grande número de variáveis de decisão para o problema de otimização. Na segunda (idem, 1.1.17), mais adequada para a solução numérica do problema, o vetor z é definido como:

$$z = (x_0, u_0, u_1, \dots, u_{N-1})$$

e a solução do problema da eq. 4.86, no tempo i , $x_i = (x_0, U)$, $U = \{u_0, u_1, \dots, u_{N-1}\}$. Assim, as funções de custo e de vínculos do problema de otimização definido na eq. 4.91 são:

$$l_o(z) = \sum_{i=0}^{N-1} l_{o_i}(x_i(x_0, U), u_i) + g_o(x_N(x_0, U)) \quad \text{eq. 4.92}$$

$$r(z) = \begin{pmatrix} l_{te_0} \\ \vdots \\ l_{te_k}(x_N(x_0, U)) \end{pmatrix} \quad \text{eq. 4.93}$$

$$f(z) = \begin{pmatrix} l_{ti_0} \\ l_{ti_1}(x_1(x_0, U)) \\ \vdots \\ l_{ti_k}(x_N(x_0, U)) \\ s_o(u_0) \\ \vdots \\ s_{N-1}(u_{N-1}) \end{pmatrix} \quad \text{eq. 4.94}$$

Para expressar o problema de controle ótimo, e de otimização deste derivado, segundo o método de discretização introduzido por Schwartz (1996), deve-se considerar, inicialmente, que a malha de discretização passa a ser definida pelos nós das funções spline em cujo espaço funcional são expressos os controles (eq. 4.61 e eq. 4.62). Forma-se assim uma seqüência de índices $i = 0, 1, \dots, N+r-1$. Nestes nós, as variáveis de controle da

eq. 4.84 são expressas em termos dos coeficientes spline $\bar{\alpha}_i = \{\alpha_i\}_{i=1}^{N+r-1} \in \bar{L}_N^{(r)}$, $\alpha_i \in \mathfrak{R}^m$ e os controles $u[\tau_{i,j}]$, $i = \{0, 1, \dots, N-1\}$, $j \in \mathbf{r}$ (eq. 4.63). Ao mesmo tempo, a dinâmica integrada genericamente segundo a eq. 4.88, deve ser expressa em termos da recursão Runge - Kutta da eq. 4.19.

4.10 Solução do problema de programação matemática com vínculos de desigualdade (*simple bounds*)

Feitas as considerações do item anterior, o problema de controle ótimo pode ser tratado como um problema de programação matemática do tipo

$$P \quad \min_{x \in \mathfrak{R}^n} f(x) \quad \text{sujeito a } x^i \geq 0, i = 1, \dots, n \quad \text{eq. 4.95}$$

$f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ é uma função continuamente diferenciável e x um vetor $x = (x^1, x^2, \dots, x^n)$.

A partir do problema P, isto é, uma minimização com vínculos de desigualdade, é possível exprimir o problema de programação matemática da eq. 4.91 à eq. 4.94, e portanto do OCP discreto, aumentando a função de custo (eq. 4.92) com os vínculos de igualdade (terminais ou de trajetória) da eq. 4.93 e multiplicadores de Lagrange. Os vínculos de desigualdade de controle do problema (eq. 4.95), que são equivalentes aos vínculos da eq. 4.84 e da eq. 4.85, podem ser transformados em vínculos gerais do tipo $u_{\min} \leq u \leq u_{\max}$ (para o algoritmo PD, que será explicado a seguir, através da equação 3.2.44 de Schwartz, 1996).

Schwartz analisou detidamente o problema P apresentado e testou um algoritmo capaz de tratá-lo (algoritmo PD), de maneira eficiente, com função objetivo não linear e vínculos de desigualdade lineares (Schwartz, 1996, cap. 3; Schwartz e Polak, 1997). O tratamento de vínculos terminais, que do ponto de vista numérico introduz dificuldades

significativas, foi incorporado, posteriormente, na rotina `aug_lagrng.m` do RIOTS; entretanto, esta rotina não foi exaustivamente testada e otimizada pelos autores. Para o caso de vínculos gerais, o RIOTS incorpora um algoritmo de otimização tipo SQP (*Sequential Quadratic Programming*), chamado NPSOL (Gill et al., 1998), que será descrito no próximo item.

O algoritmo PD apresentado por Schwartz consiste num método tipo gradiente-conjugado com busca linear inexata tipo Armijo, identificando os vínculos ativos e procedendo a minimização apenas na região factível. Foram utilizados ainda três métodos para determinar a direção de procura; método do gradiente, gradiente conjugado projetado de Polak-Ribière e L-BFGS, um algoritmo quasi-Newton com memória limitada. O algoritmo PD é especialmente adequado para problemas de grande escala, uma vez que reduz a dimensão do problema à região factível. Além disso, com o L-BFGS, a matriz Hessiana é atualizada apenas na sua porção sem vínculo, com significativa economia computacional.

Dado um conjunto factível f

$$f = \{x \in \mathfrak{R}^n \mid x^i \geq 0, i = 1, \dots, n\} \quad \text{eq. 4.96}$$

Sendo $g(x) = \nabla f(x)$ (calculado com as fórmulas apresentadas no item 4.7), para $x \in f$, os vínculos ativos podem ser identificados através do conjunto de índices $A(x)$:

$$A(x) = \{i \in 1, \dots, n \mid 0 \leq x^i \leq \varepsilon(x), g^i(x) > 0\} \quad \text{eq. 4.97}$$

e seu complemento $I(x)$

$$I(x) = \{i \in 1, \dots, n \mid x^i > \varepsilon(x) \text{ ou } g^i(x) \leq 0\} \quad \text{eq. 4.98}$$

Define-se ainda um fator de atualização $w(x)$

$$w(x) = \left| x - [x - Dg(x)]_+ \right| \quad \text{eq. 4.99}$$

em que $[\cdot]_+$ é um operador de projeção em que dado qualquer $z \in \mathfrak{R}^n$

$$[z]_+ = \begin{bmatrix} \max\{0, z^1\} \\ \vdots \\ \max\{0, z^n\} \end{bmatrix} \quad \text{eq. 4.100}$$

e D uma matriz diagonal positiva-definida.

Dado um número $\varepsilon > 0$, $\varepsilon(x)$ utilizado na definição do conjunto $A(x)$ é

$$\varepsilon(x) = \min \{ \varepsilon, w(x) \} \quad \text{eq. 4.101}$$

No conjunto $A(x)$, se $\varepsilon = 0$, $x^i = 0$ e $g^i(x) > 0$. Isso significa que x é um ponto estacionário para o problema P e está no seu valor mínimo, isto é, próximo de 0; como o gradiente é positivo, qualquer perturbação em x fará com que a função de custo aumente de valor e assim x^i não deve ser alterado. Outra situação é quando $\varepsilon > 0$, incluindo tantos índices de variáveis quanto a dimensão de $\{A(x)\}$ que estão próximas da fronteira, podendo estas variáveis encontrar a fronteira quando forem atualizadas. Schwartz (1996) no item 3.2 apresenta, assim, o algoritmo PD que identifica o conjunto de vínculos ativos num número finito de iterações. Uma vez identificadas, as variáveis ativas permanecem inalteradas, enquanto para as outras é calculado um tamanho de passo e uma direção de procura a cada iteração.

Algoritmo PD

Dados: $\alpha, \beta \in (0, 1)$, $M \in \mathbf{N}$, $\sigma_1 \in (0, 1)$, $\sigma_2 \in (0, \infty)$, $x_0 \in f$

Passo 0: $k = 0$

Passo 1: $g_k = \nabla f(x_k)$

$$A_k = A(x_k), I_k = I(x_k)$$

se $|g_k|_{I_k} = 0$ ou $|g_k|_{I_k} < \varepsilon$ e $x_k^i = 0$ para todo $i \in A_k$, PARE

Passo 2: selecionar m_k^i , $i \in A_k$ e direção de busca d_k que satisfazem as condições⁴²

$$a) d_k^i = -m_k^i g_k^i, \quad \sigma_1 \leq m_k^i \leq \sigma_2, \quad \forall i \in A_k$$

$$b) \langle d_k, g_k \rangle_{I_k} \leq -\sigma_1 |g_k|_{I_k}^2$$

$$c) |d_k|_{I_k} < \sigma_2 |g_k|_{I_k}$$

Passo 3: Calcular o tamanho de passo $\lambda_k = \beta^m$ em que m é o menor número inteiro maior que $-M$. λ_k deve satisfazer a regra tipo Armijo:

$$f(x_k(\lambda_k, d_k)) - f(x_k) \leq \alpha(\lambda_k \langle g_k, \lambda_k \rangle_{I_k} - \langle g_k, x_k - x_k(\lambda_k, d_k) \rangle_{A_k})$$

$$\text{Calcular } x_{k+1} = x_k(\lambda_k, d_k) = [x_k + \lambda_k d_k]_+$$

Passo 4: Substituir k por $k+1$ e voltar ao Passo 1

O algoritmo é capaz, após um número N de iterações, de identificar as variáveis que pertencem ao conjunto $B(\hat{x}) = \{i \mid \hat{x}^i = 0\}$, em que \hat{x} é um ponto de acumulação da seqüência gerada pelo algoritmo PD, continuando o processo de otimização ($k \geq N+1$) com as componentes $x_k^i \notin B(\hat{x})$, $x_k^i > 0$. Se $m_k^i = 1$ para todo $i \in A_k$ a direção de procura no subespaço de vínculos ativos é do máximo gradiente.

Pode ser conveniente, entretanto, que uma vez identificada a porção sem vínculo $d_k^i, i \in I_k$ da direção de procura, seja empregado outro algoritmo de minimização mais preciso, com outras condições (em relação ao Passo 3 do Algoritmo PD) de cálculo do tamanho de passo. Pode-se assim introduzir uma regra modificada para cálculo do tamanho de passo:

⁴² Essas condições (a), (b) e (c) garantem que $\sigma_1 |g_k| \leq |d_k| \leq \sigma_2 |g_k|$ e que d_k não fica ortogonal a g_k (Schwartz 1996, Observação 3.2.3).

Sejam $\sigma_3 \in (0,1]$, $\sigma_4 \in (0,\infty)$ e k a iteraç o atual do Algoritmo PD. Sendo λ_k o tamanho de passo que satisfaz o Passo 3 e $\lambda_k' \in [0, \sigma_4]$ outro tamanho de passo. Se $\lambda_k' \geq \sigma_3 \lambda_k$ ent o $\tilde{x} = x_k(\lambda_k', d_k)$. Sen o,

$$\tilde{x}^i = \begin{cases} x_k^i(\lambda_k', d_k) & \text{se } i \in I_k \\ x_k^i(\sigma_3 \lambda_k, d_k) & \text{se } i \in A_k \end{cases} \quad \text{eq. 4.102}$$

Se $f(\tilde{x}) \leq f(x_k(\lambda_k, d_k))$ ent o $x_{k+1} = \tilde{x}$

Sen o, $x_{k+1} = x_k(\lambda_k, d_k)$

A formulaç o de P apresentada na eq. 4.95 pode ser estendida para o caso em que os v nculos se d o na forma $b_l^i \leq x^i \leq b_u^i$, sendo b_l^i e b_u^i limites inferior e superior de x^i .

Nesse caso, dados o conjunto fact vel $\mathbf{f} = \{x \in \mathfrak{R}^n \mid b_l^i \leq x^i \leq b_u^i, i = 1, \dots, n\}$ e $A_k = A(x_k)$, $A(x_k) = \{i \mid b_l^i \leq x_k^i \leq b_l^i + \varepsilon(x_k) \text{ e } g_k^i > 0, \text{ ou } b_u^i - \varepsilon(x_k) \leq x_k^i \leq b_u^i \text{ e } g_k^i < 0\}$ e recai-

se no esquema anterior. Define-se ent o o operador $[\cdot]_{\#}$ em lugar de $[\cdot]_{+}$ (eq. 4.100). Dado $z \in \mathfrak{R}^n$ e $i = 1, \dots, n$

$$[\cdot]_{\#}^i = \begin{cases} b_l^i & \text{se } z^i \leq b_l^i \\ z^i & \text{se } b_l^i < z^i < b_u^i \\ b_u^i & \text{se } z^i \geq b_u^i \end{cases} \quad \text{eq. 4.103}$$

Atrav s de um exemplo de OCP com v nculos nas vari veis de controle, IP quadr tico e v nculos din micos n o-lineares, Schwartz (1996, item 3.3) prop e e testa os tr s m todos para o c lculo de direç es de procura d_k no subespaço sem v nculo. O primeiro   a utilizaç o direta do vetor gradiente ($d_k = -g_k$), o segundo   um algoritmo tipo gradiente conjugado e o terceiro o L-BFGS. Este  ltimo m todo foi o que apresentou menor custo computacional, seguido pelo gradiente conjugado.

4.11 Solução do problema de programação matemática com vínculos gerais (método SQP)

Além dos métodos tipo gradiente, como o que foi descrito no item anterior, o problema de programação matemática da eq. 4.95 pode ser tratado com outros métodos e ferramentas da programação não-linear, como a utilização de funções de barreira e penalidade, funções de mérito e métodos tipo Lagrangeano projetado (Gill et al., 1981). Estas ferramentas, se utilizadas com arte, podem ser extremamente úteis no tratamento de problemas numericamente difíceis. Entre os métodos tipo Lagrangeano projetado, estão os SQP (*Sequential Quadratic Programming*), que realizam aproximações sucessivas da função objetivo por uma aproximação quadrática e substituem os vínculos não-lineares por funções lineares. Estes sub-problemas de programação quadrática (QP) gerados são então resolvidos, encontrando-se uma direção de procura satisfatória num subespaço de direções factíveis. Schwartz e Polak (1996) vincularam ao software RIOTS, inicialmente, duas rotinas de otimização deste tipo, a FSQP (*Feasible SQP*, Zhou et al., 1997) e a NPSOL (Gill et al., 1998). Entretanto, na versão mais recente do RIOTS (Schwartz et al., 1997) utiliza apenas o NPSOL, por ser mais rápido e robusto do que o FSQP, com as limitações de não ser de domínio público e de permitir apenas uma função objetivo (não é capaz de resolver problemas tipo *minimax*).

O NSPOL resolve problemas de programação não-linear com n variáveis que visam:

$$\text{NP} \quad \min_{x \in \mathfrak{R}^n} f(x) \quad \text{eq. 4.104}$$

Sujeito a m vínculos do tipo:

$$l \leq r(x) \leq u \quad \text{eq. 4.105}$$

$$r(x) = \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \quad \text{eq. 4.106}$$

em que: x são vínculos simples (*simple bounds*), Ax são vínculos lineares e $c(x)$ não-lineares. l e u são limites inferior e superior, respectivamente. As funções $f(x)$ e $c(x)$ são suaves e, pelo menos, duas vezes diferenciáveis. Pode-se ainda definir vínculos de igualdade fazendo o limite superior igual ao inferior. Entretanto, o NPSOL considera como 'ativos' os vínculos que estão dentro de uma faixa de tolerância determinável pelo usuário; se estiver dentro da solução viável, o vínculo é considerado 'inativo' e, se passar a faixa de tolerância, 'violado'. Segundo Gill et al. (1998), a quantidade de memória utilizada pode ser estimada por $24n(n+m)$, em que n é o número de variáveis e m o número de vínculos.

Definindo as componentes do vetor gradiente da função de custo $g(x)$ e a matriz jacobiana dos vínculos $J(x)$ como:

$$g_j(x) = \frac{\partial f(x)}{\partial x_j}$$

$$J_{ij}(x) = \frac{\partial r_i}{\partial x_j} \quad \text{eq. 4.107}$$

Pode-se afirmar que um ponto factível x satisfaz as condições de otimalidade de 1ª ordem para o problema NP se

1. Existe um vetor de multiplicadores de Lagrange λ tal que o gradiente do

Lagrangeano é nulo:

$$\nabla_x L = \frac{\partial}{\partial x} (f(x) - \lambda^T r(x)) = 0 \quad \text{eq. 4.108}$$

isto é, $g(x) = J(x)^T \lambda$

2. O multiplicador de Lagrange λ_j satisfaz: $\lambda_j = 0$ se $l_j < r_j(\mathbf{x}) < u_j$

$$\lambda_j \geq 0 \text{ se } l_j = r_j(\mathbf{x})$$

$$\lambda_j \leq 0 \text{ se } r_j(\mathbf{x}) = u_j$$

$$\lambda_j \text{ pode ter qualquer valor se } l_j = u_j$$

O método utilizado pelo NPSOL para que as soluções convirjam, respeitando as condições de otimalidade, divide o processo iterativo entre *iterações maiores* e *iterações menores*. As maiores procuram determinar um tamanho de passo α e gerar uma seqüência de pontos

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{p} \quad \text{eq. 4.109}$$

em que \mathbf{p} é a direção de procura, correspondente à solução do sub-problema de programação quadrática (QP), no qual as funções de custo e vínculos são substituídas por suas respectivas séries de Taylor de 1º ordem ao redor do ponto considerado:

$$\text{QP} \quad \min_{\mathbf{p} \in \mathcal{R}^n} \left(f(\mathbf{x}) + \mathbf{g}(\mathbf{x})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{H} \mathbf{p} \right) \quad \text{eq. 4.110}$$

Sujeito a vínculos do tipo:

$$l \leq \mathbf{r}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{p} \leq \mathbf{u}; \quad \text{eq. 4.111}$$

As iterações menores, por sua vez, procuram solucionar o QP, utilizando uma rotina específica para esse problema, conhecida como LSSOL, que procura minimizar, inicialmente, a soma das violações dos vínculos. Esta é a chamada *fase de factibilidade*, que determina um ponto factível. Posteriormente, na *fase de otimalidade*, constroi-se a direção de procura de maneira que o valor dos vínculos não se alterem ao longo desta direção, de acordo com as eqs. 6.3 e 6.4 de Gill et al. (1998). Caso ocorra a ativação de vínculos simples dentro do subespaço viável, a componente correspondente da direção de procura é fixa em zero (idem, item 6.5).

Uma vez que p tenha sido calculada, as iterações maiores procuram determinar o tamanho de passo α que produza suficiente decréscimo numa *função de mérito Lagrangiana*, que mede a qualidade de cada iteração. Esta função consiste na própria função de custo $f(x)$ aumentada com termos correspondentes aos vínculos não-lineares, multiplicados por estimativas dos multiplicadores de Lagrange e por uma função quadrática de penalidade (Gill et al., 1998, eq. 6.5). Em seguida, a estimativa positiva-definida da matriz Hessiana da função Lagrangiana é atualizada através de um método quasi-Newton BFGS.

4.12 Utilização de integradores com passo variável e múltiplo

Além dos integradores tipo RK de passo fixo, é possível também empregar, na discretização do OCP e na integração das equações diferenciais, outros métodos de integração numérica, com passo variável e múltiplo⁴³. Em problemas de controle ótimo, os métodos de passo fixo são na maior parte dos casos, ao contrário de problemas de valor inicial, a opção mais indicada. Do ponto de vista do erro de global da solução em problemas de controle ótimo parametrizados, o tipo de representação do subespaço de controle deve também ser levada em conta, e não apenas a precisão do integrador. Além disso, o cálculo do tamanho de passo da direção de procura do algoritmo de otimização, de uma iteração para a outra, fica alterado pela mudança do tamanho de passo do integrador. Schwartz (1996, item 4.6.1) realizou alguns testes comparativos entre soluções de OCPs com passo fixo e variável, percebendo o desempenho inferior do método de passo variável (Schwartz, 1996, Tabela 4.6.1).

⁴³ Algoritmo que utiliza aproximações por funções polinomiais da equação diferencial, passando por vários pontos da malha de discretização. Para uma discussão sobre a implementação e eficiência desse método, ver Press et al. (1992, item 16.7). Os métodos RK, diferentemente, utilizam apenas os passos k e $k+1$ a cada iteração, ainda que a equação diferencial seja avaliada em pontos intermediários entre esses dois passos.

Entretanto, em problemas duros ou altamente não lineares, a utilização de integradores de passo variável e múltiplo pode ser a única opção, pela inviabilidade de se utilizar passos fixos muito pequenos, que acarretam alto custo computacional, grande quantidade de variáveis para o problema de otimização e grandes erros globais de integração⁴⁴. O RIOTS incorpora a rotina LSODA, parte do pacote de domínio público ODEPACK desenvolvido por Alan Hindmarsh no Lawrence Livermore National Laboratory (Byrne e Hindmarsh, 1987). O LSODA incorpora dois algoritmos de integração, um para equações diferenciais duras e outro para não-duras. Para equações suaves, utiliza o método de passo múltiplo variável explícito Adams-Bashforth na primeira estimativa de x_{k+1} , e o método implícito Adams-Moulton para as estimativas subsequentes⁴⁵, utilizando passos menores para assegurar a convergência (Parker e Chua, 1987). Um algoritmo desenvolvido por Petzold (1993) detecta se a integração apresenta ou não características de sistemas duros, alternando durante a simulação do método de Adams para o método Gear, que utiliza apenas fórmulas de diferenciação implícitas e resolve um problema linear resultante com fatoração LU. Com isso, a região de estabilidade do integrador se amplia em função do tamanho de passo, que pode ser ajustado a cada iteração para obedecer um critério de erro mínimo de integração (ver comentários no código LSODA).

Entretanto, o cálculo dos gradientes das funções de custo e vínculos não pode ser feito com as fórmulas apresentadas no item 4.4 se for utilizado o LSODA. Não é possível

⁴⁴ Erros que se acumulam no final da integração, resultado da soma de erros locais das aproximações numéricas (erro de truncamento) e da representação computacional dos números reais (erro de aproximação). Ver Parker e Chua (1989, item 4.2).

⁴⁵ O método Adams-Bashford é um integrador explícito (as diferenças finitas para aproximação da derivada são do tipo $(x_{k+1} - x_k) / h$) de implementação simples, ao passo que o Adams-Moulton é implícito, com derivadas aproximadas por $(x_k - x_{k-1})/h$ e $x_{k+1} = f(x_k, x_{k+1})$ (Parker e Chua, 1989, item 4.1). Combinando os dois algoritmos, pode-se obter a solução da iteração implícita, dando origem a um método tipo *preditor-corretor*. Outro método possível para resolver a iteração implícita é utilizar o algoritmo de Newton-Raphson.

calcular os gradientes exatos no espaço discretizado, sendo necessário estimar esses valores a partir das equações diferenciais contínuas do problema, segundo as fórmulas apresentadas em Schwartz et al. (1997, item 5.5).

4.13 Seleção da ordem de integração e da aproximação por splines

O problema de aproximação que foi formulado nos itens acima possui como principais parâmetros de solução o nível de discretização N , a ordem de integração do método RK e a ordem da representação spline do controle. Cada um desses parâmetros produz duas fontes distintas de erros. O erro de integração é gerado pela discretização das equações diferenciais do sistema dinâmico. O erro de representação, por sua vez, acontece devido à parametrização do espaço de controle de dimensão infinita por um espaço de coeficientes spline, de dimensão finita. Deseja-se, com isso, que a escolha dos parâmetros de solução leve a um equilíbrio entre as contribuições de cada uma dessas fontes de erro, de maneira que não ocorra desperdício de esforço computacional, representando com alta precisão controles de um sistema dinâmico pobremente discretizado ou vice versa. Deve-se ainda ter em conta que, no balanço entre o nível de discretização N e a ordem de integração s , o aumento da ordem permite até certo ponto o aumento do tamanho de passo sem prejuízo à estabilidade da integração. Além disso, a utilização de uma ordem de representação polinomial muito elevada pode levar à sobre-parametrização da representação, com diminuição da precisão.

Tratando inicialmente o problema de aproximação sem vínculos no controle, Schwartz (1996) conseguiu relacionar, através do Teorema 2.2 (neste texto, Teorema 5) e de testes numéricos, a ordem dos erros de representação (Tabela 4.1) e de integração em função da ordem de integração RK e da ordem da representação spline.

Método RK	$\max_{k,j} \left \bar{u}_{N,k,j}^* - u^*(\tau_{k,j}) \right $	$\max_k \left \bar{x}_k^* - x^*(t_k) \right $
1ª ordem	$O(\Delta_N^1)$	$O(\Delta_N^1)$
2ª ordem	$O(\Delta_N^2)$	$O(\Delta_N^2)$
3ª ordem	$O(\Delta_N^2)$	$O(\Delta_N^3)$
4ª ordem	$O(\Delta_N^3)$	$O(\Delta_N^4)$

Tabela 4.1: Erros de representação e de integração em função da ordem de integração e do nível de discretização (Schwartz, 1996). $\Delta_N = \max_k \Delta_{N,k}$

Teorema 5: Seja $u^* \in C^{(\rho)}[a,b]$ a solução⁴⁶ local do problema P. Considerando o método RK na Tabela 4.1, ρ é o expoente a que Δ_N está elevado na primeira coluna. Seja ainda uma seqüência $\{u_N\}$, distribuída em nós não necessariamente uniformes, a solução local do problema P discretizado (P_N). Supondo a existência e continuidade para cada N de $d_{\alpha}^2 \bar{f}_N(\cdot)$, que $d_{\alpha}^2 \bar{f}_N(\alpha_N^*) > 0$ e $\Delta_N \left| d_{\alpha}^2 \bar{f}_N(\cdot)^{-1} \right|$ é uniformemente limitado em relação a N , então

$$\left| u^* - u_N^* \right|_{\infty} \approx O(\Delta_N^{\rho})$$

Para problemas com vínculos, a Tabela 4.1 e o Teorema 5 fornecem limites inferiores de erro. A Tabela 4.2 mostra as ordens de representação viáveis para cada método RK.

⁴⁶ $C^{(\rho)}[a,b]$: funções contínuas ρ vezes diferenciáveis definidas no intervalo fechado $[a, b]$. Por exemplo, uma spline de ordem ρ .

Método RK	Ordem de representação spline
1ª ordem (Euler)	1
2ª ordem	1 ou 2
3ª ordem	1 ou 2
4ª ordem	1, 2 ou 3

Tabela 4.2: Ordem possível de escolha das splines em função da ordem de integração

Em problemas com vínculos de controle, freqüentemente as trajetórias de controle são não-suaves, e mesmo descontínuas, tornando os limites do erro de representação pouco sensíveis ao aumento de ordem da spline. Entretanto, splines de primeira ordem (na realidade, funções constantes por partes) têm sua escolha preterida em relação às de segunda ordem que, em certas condições de suavidade e convexidade das funções de custo e vínculos, alcançam soluções mais precisas.

Com relação à ordem de integração, a escolha ótima seria aquela capaz de minimizar o trabalho computacional mantendo uma precisão pré-especificada. Essa escolha depende, entretanto, de diversos fatores, como do grau de não-linearidade, das características de estabilidade do sistema dinâmico, da suavidade da solução, da quantidade de trabalho computacional utilizada pelo algoritmo de otimização a cada iteração, etc.

Do que foi dito acima e em outras considerações levantadas pelo autor do método, uma série de recomendações práticas podem ser enunciadas, auxiliando o usuário na escolha dos parâmetros adequados (Tabela 4.3)

Recomendações para escolha da ordem de discretização, de integração e de representação por splines

- 1 Para RK de 2^a, 3^a e 4^a ordem e vínculos de controle e/ou estados (provavelmente não-suaves) utilizar splines de 2^a ordem, eventualmente 1^a.
- 2 Em problemas sem vínculos de controle, presumivelmente suaves, utilizar splines de 3^a ordem e RK de 4^a ordem, possibilitando convergência $O(\Delta_N^3)$.
- 3 Se for utilizada a transformação de coordenadas para splines (eq. 4.79 e eq. 4.80), pode ser necessária uma quantidade tal de cálculos para aplicar a transformação que exceda o custo computacional do algoritmo de integração. Se, por exemplo, se estiverem sendo usadas splines de 3^a ordem, recomenda-se substituí-las por splines de 2^a ordem com maior nível de discretização.
- 4 Em baixos níveis de discretização, é provável que os erros de integração superem largamente os de representação. Nestes casos, recomenda-se aumentar a ordem de integração RK (por exemplo 4^a ordem) e splines lineares.
- 5 De um modo geral, RK 4^a ordem é a melhor opção, justamente por permitir a diminuição do nível de discretização. Exceção deve ser feita em dois casos:
 Em problemas com dinâmica linear bem comportada e vínculos de controle, bastam normalmente splines com $\rho=2$ e método de integração de 2^a ordem.
 Em problemas que apresentam vínculos de estado deve-se calcular gradientes nos pontos da malha de discretização em que os vínculos estejam definidos, sendo o trabalho para calcular esses gradientes proporcional a sN^2 , onde s é a ordem de integração. Assim, diminuir se possível a ordem de integração

Tabela 4.3: Recomendações para escolha dos parâmetros de solução.

4.14 Estimação dos erros de integração e aproximação e definição de estratégias de redistribuição da malha de discretização

Uma vez formulado o problema de aproximação, provada sua epiconvergência e apontada uma estratégia adequada para a solução do problema de programação matemática obtido, é de grande utilidade que se disponha de ferramentas capazes de estimar os erros obtidos ao longo das operações matemáticas efetuadas.

Além da utilidade imediata de avaliação da solução obtida, a estimação de erros de integração permite que possam ser introduzidas estratégias de refinamento e redistribuição da malha de discretização, com o intuito de reduzir o erro global da solução, refinando ou aumentando a malha, conforme a necessidade, em locais específicos. Serão mostradas a seguir estratégias para estimação do erro de integração, assim como estratégias de refinamento de malha para a minimização de cada uma dessas fontes de erro.

Seja um passo de integração RK de ordem s :

$$\bar{x}_{N,k+1} = x_{N,k}(t_{k+1}) + e_{N,k} \Delta_{N,k}^{s+1} + O(\Delta_{N,k}^{s+2}) \quad \text{eq. 4.112}$$

A solução real da equação $\dot{x} = h$ é $x_{N,k}(t_{k+1})$. Serão chamados assim:

$$e_{N,k} \Delta_{N,k}^{s+1} = \text{erro local principal de truncamento (PLTE)}$$

$$\bar{x}_{N,k+1} - x_{N,k}(t_{k+1}) = \text{erro local de truncamento (LTE)}$$

Expandindo a expressão do LTE numa série de Taylor ao redor de $\bar{x}_{N,k}$ é possível obter uma expressão para o coeficiente do PLTE $e_{N,k}$, resultado que pode ser obtido através de outras estimativas (Schwartz, 1996 item 4.3.1; Hosea e Shampine, 1994). Deste modo, é possível avaliar locais da malha com altos e baixos erros de integração, apontando

assim para o refinamento ou o aumento dos intervalos da malha, de forma que o máximo PLTE ao longo dos intervalos de integração seja minimizado.

Schwartz apresenta duas estratégias para tal refinamento da malha. Na primeira, de **nós móveis**, uma vez obtida a solução do OCP sem qualquer refinamento, determina-se uma nova malha, com uma nova seqüência e número de pontos, garantindo que o PLTE a cada iteração k é o mesmo para todo k . Utilizando a segunda estratégia, chamada de **nós fixos**, novos pontos podem ser adicionados através da divisão do intervalo em subintervalos, ou pontos da malha original eliminados, mas não mudar de lugar. Pontos da malha são eliminados se uma estimativa do LTE for menor que um parâmetro σ . Definindo uma *constante de quase-uniformidade* $\bar{\delta}$, correspondente ao maior intervalo entre dois pontos da malha, calcula-se, para as duas estratégias, uma estimativa que é proporcional ao benefício introduzido pelo refinamento.

Aplicando os dois métodos para o OCP de Rayleigh⁴⁷ com subespaço de funções constantes por partes para representar o controle, RK 2ª ordem, $N=50$, $\sigma = 1/4$, observou-se uma queda acentuada dos erros de integração. A estratégia 1 reduziu o erro total de integração em 7 vezes sem aumento do número de pontos da malha, enquanto a 2 aproximadamente 13 vezes, reduzindo o erro assintoticamente a zero, numa nova malha com $N=64$. No problema Bang-Bang⁴⁸, a queda do erro total de integração foi da ordem de 100 vezes com a estratégia (Schwartz, 1996, Tabela 3.5).

⁴⁷ Este problema consiste em $\min_u J(u) = \int_0^{2.5} x_1^2 + u^2 dt$, sujeito a

$$\begin{cases} \dot{x}_1(t) = x_2(t) & x_1(0) = -5 \\ \dot{x}_2(t) = -x_1(t) + [1.4 - 0.14x_2^2(t)] + 4u(t) & x_2(0) = -5 \end{cases}$$

⁴⁸ Ou seja, $\min_{u,T} J(u, T) = T$ sujeito a

Schwartz (1996, item 4.4) faz a demonstração de duas fórmulas, para problemas com e sem vínculos de controle, capazes de fornecer uma estimativa do erro da solução, isto é, da distância entre o vetor de controles calculados para o problema de aproximação η_{N+1} e o vetor de controles ótimos η^* do problema original, dado um conjunto de condições iniciais ξ . O autor realizou testes com diversos OCPs, utilizando RK de 2ª e 4ª ordem, splines $\rho=2$ e $\rho=3$ (idem, Tabela 4.4.1) . A comparação dos erros da solução aproximados, calculados a partir das fórmulas desenvolvidas, com soluções obtidas através de malhas altamente refinadas ou passo de integração variável e baixa tolerância, mostrou-se bastante satisfatória.

4.15 Utilização do RIOTS_95

A partir da Teoria das Aproximações Consistentes e das estratégias de discretização e parametrização do Problema de Controle Ótimo, Schwartz (1996, Cap. 5) desenvolveu e implementou um *software* para tratamento de tais problemas. Tendo recebido o nome de RIOTS (*Recursive Integration Optimal Trajectory Solver*), consiste num conjunto de rotinas (*toolbox*) escritas parte em C e parte em *Matlab* que permite ao usuário definir seu OCP, resolvê-lo utilizando diversas rotinas de otimização e de redistribuição da malha de discretização e estimar os erros da solução. Praticamente toda teoria desenvolvida por seu autor e parcialmente descrita nos itens precedentes do presente trabalho, assim como outros tópicos relacionados, foi incorporada no *software*, que permite a definição de diversos tipos de vínculos, utilização de rotinas de integração

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases}, \quad \begin{cases} x_1(0) = 0 & x_1(T) = 300 \\ x_2(0) = 0 & x_2(T) = 0 \end{cases} \quad e$$

$$-2 \leq u(t) \leq 1, \forall t \in [0, T]$$

numérica com passo fixo e variável, transformação de coordenadas para geração de base ortonormal no subespaço de coeficientes spline, etc. A forma básica de expressar o OCP tendo em vista sua solução pelo RIOTS é a seguinte:

$$\text{OCP} \quad \min_{(u, \xi) \in L_{\infty, 2}^m[a, b] \times \mathfrak{R}^n} \left\{ \max_{v \in \mathbf{q}_o} \left\{ f^v(u, \xi) = g_o^v(\xi, x(b)) + \int_a^b l_o^v(t, x, u) dt \right\} \right\}$$

Sujeito a:

Sistema de equações diferenciais que descrevem a dinâmica do problema, com vetor de condições iniciais ξ

$$\dot{x} = h(t, x, u)$$

$$x(a) = \xi$$

$$t \in [a, b]$$

Vínculos de controle

$$u_{\min}^j(t) \leq u^j(t) \leq u_{\max}^j(t), \quad j = 1, \dots, m$$

Limites das condições iniciais⁴⁹ (quando livres)

$$\xi_{\min}^j \leq \xi^j \leq \xi_{\max}^j, \quad j = 1, \dots, n$$

Vínculos de desigualdade nas trajetórias⁵⁰

$$l_{ii}^v(t, x(t), u(t)) \leq 0, \quad v \in \mathbf{q}_{ii}, \quad t \in [a, b]$$

Vínculos de desigualdade terminais

$$g_{ei}^v(\xi, x(b)) \leq 0, \quad v \in \mathbf{q}_{ei}$$

Vínculos de igualdade terminais

$$g_{ee}^v(\xi, x(b)) = 0, \quad v \in \mathbf{q}_{ee}$$

Quanto às funções definidas acima,

$$x(t) \in \mathfrak{R}^n$$

$$u(t) \in \mathfrak{R}^m$$

$$g: \mathfrak{R}^n \times \mathfrak{R}^n \rightarrow \mathfrak{R}$$

⁴⁹ O resultado final da otimização realizada pelo RIOTS é uma seqüência $\eta^* = (u, \xi) \in H_2 = L_{\infty, 2}^m[a, b] \times \mathfrak{R}^n$.

⁵⁰ Vínculos desse tipo podem ser tratados como vínculos terminais, adicionando funções de penalidade no IP ou utilizando um estado aumentado (Schwartz, 1996, item 5.2).

$$l: \mathfrak{R}_x \mathfrak{R}^n \rightarrow \mathfrak{R}$$

$$h: \mathfrak{R}_x \mathfrak{R}^n \mathfrak{R}^m \rightarrow \mathfrak{R}^n$$

Outros tipos de OCPs podem ser tratados com a mesma formulação apresentada acima. Por exemplo, um problema eventualmente não-autônomo⁵¹ com tempo final aberto:

$$\min_{u, T} \tilde{g}(T, x(T)) + \int_a^{a+T} l(t, u, x) dt$$

$$\text{sujeito a } \dot{x} = \tilde{h}(t, x, u), x(a) = \xi, t \in [a, a+T],$$

pode ser convertido num problema com tempo definido, utilizando um vetor de estados aumentado

$$x_A = \begin{Bmatrix} \{x\} \\ x_{n+1} \\ x_{n+2} \end{Bmatrix} = \begin{Bmatrix} \{x\} \\ a + (t-a)\beta \\ \beta \end{Bmatrix}$$

$$x_A(a) = \alpha = \begin{Bmatrix} \{x\} \\ a \\ \beta \end{Bmatrix}$$

$$\min g(\alpha, x_A(b)) + \int_a^b l(t, x_A, u) dt$$

$$\text{Sujeito a } \dot{x}_A = h(t, x_A, u) = \begin{Bmatrix} x_{n+2} h(x, x_{n+1}, u) \\ x_{n+2} \\ 0 \end{Bmatrix}$$

sendo:

$$g(\alpha, x_A(b)) = \tilde{g}((a+T)\beta, x(b))$$

$$l(t, x_A, u) = \tilde{l}(x_{n+1}, x, u)$$

⁵¹ Para um sistema autônomo com tempo final aberto não é necessário x_{n+1} .

$b = a + T$, $T = b - a$: Tempo nominal de duração da trajetória

β : fator de escala de duração

$t_f = x_{n+1}(b) = a + (t - a) \beta$. A duração ótima será $T^* = t_f - a = (b - a)\beta = T\beta$

Se $a = 0$ e $b = 1$, $t_f = T^* = \beta$

Existem no *software* três grupos de rotinas: de simulação, otimização e rotinas auxiliares. As rotinas de simulação estão disponíveis através das várias formas do comando `simulate`, podendo ser utilizados algoritmos RK de várias ordens com passo fixo ou algoritmos com passo variável. A partir de diversos parâmetros de entrada, como um vetor de tempo, estados iniciais e finais, o vetor de controle, etc., são calculadas a trajetória do sistema, o valor do IP alcançado, diversos gradientes, valores ao longo do tempo das funções de vínculo, erros de integração entre outras grandezas.

Dependendo das características do OCP, estão disponíveis várias rotinas de otimização. A mais genérica chama-se `riots` e está baseada no pacote comercial de otimização NPSOL (Gill et al., 1986). Está desenvolvida para resolver OCPs com todos os tipos de vínculos definidos acima e uma função objetivo. Entretanto, é computacionalmente cara, sendo viável apenas a problemas de médio porte sem malhas de discretização muito refinadas. A rotina menos genérica, porém mais econômica, é a `pdmin`, capaz apenas de trabalhar com vínculos de controle simples e limites nas condições iniciais. É indicada para problemas de grande porte que podem ser formulados desta maneira, e oferece diversas opções na escolha do algoritmo de otimização: L-BFGS, gradiente e gradiente projetado (Schwartz et al., 1997). Uma versão modificada de `pdmin` é a rotina `aug_lagrng`, que pode incluir também vínculos terminais de igualdade. A subrotina `outer` utiliza `riots` várias vezes, redistribuindo a malha de

discretização, através do comando `distribute`, até atingir uma precisão especificada. Existem ainda outras rotinas auxiliares, que estimam erros de integração e da aproximação do controle, aplicam transformações de coordenadas etc.

Dado um OCP, o usuário escreve um conjunto de oito funções, em Matlab ou c: `init`, `acti`, `h`, `l`, `g`, `Dh`, `Dl`, `Dg`. Nas duas primeiras são definidos as características do OCP e nas outras as equações diferenciais, as funções `l`, as `g`, seguidas de suas derivadas. As derivadas podem ser eventualmente omitidas, pois o programa pode calculá-las automaticamente através de diferenças finitas.

4.16 Considerações gerais e limitações do método

Como já foi dito no item 4.12, em problemas de integração de equações diferenciais através de métodos numéricos, as propriedades de estabilidade \times custo da solução são bastante superiores quando se utiliza passo de integração variável. Em OCPs, entretanto, essa assertiva não é necessariamente verdadeira, já que o erro de integração depende também da capacidade de aproximação que possui a representação paramétrica do controle. Assim, se o problema crítico é reduzir o erro, a opção é utilizar discretizações com tamanho de passo fixo ao longo das diversas tentativas de solução, aplicando ao final alguma estratégia de refinamento da malha.

Em problemas com vínculos de igualdade ou desigualdade nas variáveis de estado, é possível que não exista uma solução, para um dado nível de discretização, que respeite todos esses vínculos. Recomenda-se assim que as tolerâncias na violação dos vínculos sejam aumentadas, caso a discretização não seja suficientemente refinada, evitando cálculos desnecessários na tentativa de satisfazer tais vínculos.

Entre as principais limitações atuais do *software* apontadas pelo seu autor, está a incapacidade de trabalhar com problemas de grande porte com alto nível de discretização e

vínculos genéricos, principalmente desigualdades de estados. Problemas com dinâmica altamente instável requerem uma boa estimativa do vetor de controles inicial; além disso, não é possível tratar até o momento problemas com atrasos na sua dinâmica. Entretanto, tais limitações são compartilhadas por praticamente todos os algoritmos de tratamento numérico de controle ótimo desenvolvidos até o momento.

Capítulo 5

Testes preliminares

Inicialmente, a metodologia proposta no Capítulo 4 para tratamento do problema de controle ótimo foi implementada num pêndulo invertido com um grau de liberdade. A partir deste modelo bastante simplificado, foram realizadas implementações do OCP (Optimal Control Problem) no ambiente RIOTS, utilizando funções de custo de vínculos em Matlab e em C, tendo em vista comparações de desempenho computacional. Para o mesmo modelo foram estudados, ainda que de maneira não exaustiva, problemas com tempo aberto e funções de penalidade. Posteriormente, o OCP foi formulado para um sistema de 3 corpos rígidos no plano unidos por juntas tipo articulação. Foram realizados testes com diversos tipos de atuadores de torque, com e sem dinâmica, atuadores musculares com dinâmica linear e atuadores musculares lineares. A partir dos resultados obtidos com esses modelos simplificados, foi possível avaliar as principais dificuldades numéricas encontradas na implementação de modelos mais complexos

5.1 Pêndulo com um grau de liberdade, atuador de torque

As equações de movimento do pêndulo com um grau de liberdade, mostrado na Figura 5.1, podem ser obtidas facilmente a partir da Mecânica Newtoniana.

Aplicando o Teorema do Momento Angular em relação ao ponto O:

$$\begin{aligned} \sum M_o &= I\ddot{\theta} \\ -mgL \cos(\theta) + M &= I\ddot{\theta} \end{aligned} \quad \text{eq. 5.1}$$

em que:

M_o é o momento das forças externas em relação ao eixo de rotação do corpo rígido

M é um momento externo aplicado

I é o momento de inércia de rotação em relação à articulação O

m é a massa do corpo

g é a aceleração da gravidade

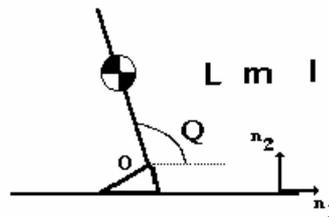


Figura 5.1: Pêndulo com 1 grau de liberdade.

Dividindo pelo momento de inércia I , fazendo uma transformação de coordenadas do tipo

$\theta = x + \pi/2$, e tomando o momento externo aplicado como variável de controle u ,

$$\ddot{x} - \frac{mgL}{I} \text{sen}(x) = u \quad \text{eq. 5.2}$$

ou seja, em variáveis de estado,

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{mgL}{I} \text{sen}(x_1) + u \end{cases} \quad \text{eq. 5.3}$$

Com o objetivo de determinar uma lei de controle em malha fechada para o sistema, que fornece uma estimativa inicial para o vetor de controle na solução do OCP em malha aberta, o sistema dinâmico é linearizado a partir de uma expansão em série de Taylor de 1ª ordem. Seja:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} f_1(x, u, t) \\ f_2(x, u, t) \end{pmatrix},$$

a expansão de Taylor em torno de uma posição \bar{x}, \bar{u}, t fica

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} \cong \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \Big|_{\bar{x}, \bar{u}, t} + \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} \Big|_{\bar{x}, \bar{u}, t} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{pmatrix} \Big|_{\bar{x}, \bar{u}, t} u \quad \text{eq. 5.4}$$

que permite chegar ao sistema linear

$$\begin{aligned} \{\dot{x}\} &= [A]\{x\} + \{B\}u \\ y &= [C]\{x\} \end{aligned} \quad \text{eq. 5.5}$$

no qual

$$[A] = \begin{pmatrix} 0 & 1 \\ \frac{mgL}{I} \cos(\bar{x}) & 0 \end{pmatrix}, \quad \{B\} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad [C] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

O problema de controle ótimo fica definido então como:

OCP: determinar o controle $u(t)$ que minimiza o funcional

$$f(u, \xi) = g_0(u, x(t_f)) + \int_0^{t_f} l_0(t, x, u) dt$$

$$\text{sujeito a:} \quad \dot{x} = h(t, x, u), \quad x(0) = \xi, \quad t \in [0, t_f]$$

$$g_{ee}(x(t_f)) = 0$$

A função objetivo foi definida como:

$$f(u, x(t_0)) = \int_0^2 100x_1^2 + 100x_2^2 + u^2 dt$$

Os vínculos terminais de igualdade g_{ee} foram estabelecidos como $x_i(t_f) = 0$, $i = 1, 2$ (ou seja, o pêndulo invertido deverá ficar parado e ereto no tempo final) e o sistema dinâmico, $h(t, x, u)$, na eq. 5.3 Neste problema, não foi estabelecido nenhum vínculo de controle. Os parâmetros do pêndulo foram adotados como: $I=20 \text{ N.m}^2$, $m=70 \text{ kg}$, $l=0.8 \text{ m}$, $g=9.81 \text{ m/s}^2$.

Para efeitos de comparação do custo computacional, as funções que definem o OCP, em ambiente RIOTS, foram implementadas em Matlab e em C. No primeiro caso, as funções definidas foram:

`t1glol.m: definição do problema`

```
clear global
clear all
global m g L I K

% tese: pendulo com 1 gl, acionador de torque, malha aberta
!copy e:\user\luciano\riots\Riots_95\systems\simulate0.dll
!rename simulate0.dll, simulate.dll
m=70;
g=9.81;
L=0.8;
I=20;
x1f=0;
x2f=0;
tf=2;
global params
params=[m g L I x1f x2f];

N=100; %número de pontos da malha
T=tf; %tempo final
t=[0:T/N:T];
r=2; %ordem da spline
u0=zeros(1,N+r-1);
x0=[pi/4;0];
%uboundl=[-1;-1];
%uboundh=[1;1];
uboundl=[];
uboundh=[];

[info,simed]=simulate(0,params)
simulate(0)
[f,x]=simulate(1,x0,u0,t,4,2)
figure(1)
zoom on

%cálculo de um vetor de controles de feed-back para chute inicial do controle open-loop
%através de LQR
E_op=0; %ângulo de operação na linearização
A=[0 1
    (m*g*L/I)*cos(E_op) 0];
B=[0;1];
```

```

Q=[1 0
   0 1];
R=1;
[K,S,E]=lqr(A,B,Q,R);
[t,x_aux] = ode23('Faux_u0',t,x0);
u0_aux=-K*x_aux';
t=t';
subplot(211)
plot(t,x_aux*180/pi)
title('estados - só realimentação')
ylabel('E, Ep')
xlabel('t(s)')
grid
subplot(212)
plot(t,u0_aux)
title('estimativa inicial do controle')
ylabel('n.m')
xlabel('t(s)')
grid

%cálculo da trajetória
med_tempo=cputime;
[u,x,f]=riots(x0,u0_aux,t,uboundl,uboundh,params,1000,4);
tempo_cpu=cputime-med_tempo

sys_init.m

function neq = sys_init(params)

global params
% Here is a list of the different system information paramters.
% neq = 1 : number of state variables.
% neq = 2 : number of inputs.
% neq = 3 : number of parameters.
% neq = 4 : reserved.
% neq = 5 : reserved.
% neq = 6 : number of objective functions.
% neq = 7 : number of nonlinear trajectory constraints.
% neq = 8 : number of linear trajectory constraints.
% neq = 9 : number of nonlinear endpoint inequality constraints.
% neq = 10 : number of linear endpoint inequality constraints.
% neq = 11 : number of nonlinear endpoint equality constraints.
% neq = 12 : number of linear endpoint equality constraints.
% neq = 13 : 0 => nonlinear, 1 => linear, 2 => LTI, 3 => LQR, 4 => LQR and LTI.

% The default value is 0 for all except neq = 6 which defaults to 1.

neq=[1 2 ; 2 1 ; 3 6 ; 11 2 ; 13 0];
global sys_params
sys_params = params;

%end

sys_act.m

function message = sys_activate
% This is a good time to allocate and set global variabls.
message = 'tlglol: acionador de torque, um grau de liberdade, open-loop';

sys_h.m, sys_dh.m: definição do sistema dinâmico

function xdot = sys_h(neq,t,x,u)
global sys_params
% xdot must be a column vectore with n rows.
m=sys_params(1);
g=sys_params(2);
L=sys_params(3);
I=sys_params(4);
xdot(1)=x(2);
xdot(2)=(m*g*L/I)*sin(x(1))+u;
xdot=xdot';

% function [h_x,h_u] = sys_Dh(neq,t,x,u)
function [h_x,h_u] = sys_dh(neq,t,x,u)

```

```

global sys_params
m=sys_params(1);
g=sys_params(2);
L=sys_params(3);
I=sys_params(4);
h_x=[0 1
      (m*g*L/I)*cos(x(1)) 0];
h_u=[0;1];

```

sys_g.m, sys_dg.m: definição dos vínculos terminais e g_0

```

function J = sys_g(neq,t,x0,xf)
global sys_params
x1f=sys_params(5);
x2f=sys_params(6);
F_NUM = neq(5);
if F_NUM == 1
    J = 0;
elseif F_NUM == 2
    J = xf(1)-x1f;
elseif F_NUM == 3
    J = xf(2)-x2f;
end

```

```

function [J_x0,J_xf,J_t] = sys_dg(neq,t,x0,xf)
global sys_params
x1f=sys_params(5);
x2f=sys_params(6);
F_NUM = neq(5);
J_x0 = [0 0];
J_xf = [0 0];
if F_NUM == 2
    J_xf(1) = 1;
elseif F_NUM == 3
    J_xf(2) = 1;
end

```

sys_l.m, sys_dl.m: definição da função de custo

```

function z = sys_l(neq,t,x,u)
global sys_params
z=100*x(1)^2+100*x(2)^2+u^2;

function [l_x,l_u,l_t] = sys_dl(neq,t,x,u)
global sys_params
l_x=[100*2*x(1) 100*2*x(2)];
l_u=[2*u];

```

Faux_u0.m: equações dinâmicas para teste da estimativa inicial do controle

```

function x_auxp=Faux_u0(t,x_aux);
global m g L I K
u=-K*[x_aux(1) x_aux(2)]';
x_auxp(1)=x_aux(2);
x_auxp(2)=(m*g*L/I)*sin(x_aux(1))+u;
x_auxp=x_auxp';

```

Para definir o OCP num arquivo em linguagem C, posteriormente vinculado a rotinas residentes em C, Fortran e DLLs (Bibliotecas de vínculo dinâmico do sistema Windows), foram escritas duas rotinas em Matlab (de inicialização) e Watcom C 10.5:

t1glol2.c: definição do problema

```

% 1ª parte: GERAÇÃO DO DLL

clear global
clear all
nome=char('t1glol2');

```

```

dirsaída=('e:\user\luciano\riots\tese\tlglolc');
dirinstal=('e:\user\luciano\riots\riots_95\systems');
nomec=strcat(nome, '.c');
nomedll=strcat(nome, '.dll');
eval(['delete ', nomedll]);
delete simulate.dll
eval(['chdir ', dirinstal]);
eval(['delete ', nomedll]);
comando=char(['c:\matlab\bin\mex -v4 -v ', dirsaída, '\', nomec, ' simulate.obj utility.obj
..\f2c\libf2c.lib ..\drivers\drivers.lib'])
fid=fopen('cel.bat', 'w')
fprintf(fid, '%c', comando)
fclose(fid)
disp('tecle cel.bat na janela dos')
pause
eval(['!copy ', nomedll, ' ', dirsaída])
eval(['chdir ', dirsaída])
eval(['!rename ', nomedll, ' simulate.dll'])

%2ª PARTE: ESCOLHA DOS PARÂMETROS DO SISTEMA E DE DISCRETIZAÇÃO

global m g L I K

m=70;
g=9.81;
L=0.8;
I=20;

x1f=0;
x2f=0;

tf=2;

Q11=1;
Q22=1;
R11=1;

params=[m g L I x1f x2f 100 100 R11];

N=100; %número de pontos da malha
T=tf; %tempo final
t=[0:T/N:T];
r=2; %ordem da spline
u0=zeros(1,N+r-1);
x0=[pi/4;0];
uboundl=[];
uboundh=[];
simulate(0)
[info,simed]=simulate(0,params)
[f,x]=simulate(1,x0,u0,t,4,2)

%cálculo de um vetor de controles de feed-back para chute inicial do controle open-loop
%através de LQR
E_op=0; %ângulo de operação na linearização
A=[0 1
(m*g*L/I)*cos(E_op) 0];
B=[0;1];
Q=[Q11 0
0 Q22];
R=R11;
[K,S,E]=lqr(A,B,Q,R);
[t,x_aux] = ode23('Faux_u0',t,x0);
u0_aux=-K*x_aux';
t=t';
figure(1)
subplot(211)
plot(t,x_aux*180/pi)
title('estados - só realimentação')
ylabel('E, Ep')
xlabel('t(s)')
grid
subplot(212)
plot(t,u0_aux)
title('estimativa inicial do controle')

```

```

ylabel('n.m')
xlabel('t(s)')
grid

%cálculo da trajetória
med_tempo=cputime;
[u,x,f]=riots(x0,u0_aux,t,uboundl,uboundh,[params],1000,4);
tempo_cpu=(cputime-med_tempo)/60;
figure(2)
zoom on
subplot(211)
plot(t,180/pi*x)
title('estados')
ylabel('E, Ep')
xlabel('t(s)')
grid
subplot(212)
sp_plot(t,u)
title('controle malha aberta')
ylabel('n.m')
xlabel('t(s)')
grid

```

tlglol2.c: funções do OCP

```

/*
pêndulo invertido
controlado por torque
lg1
3/4/99
*/

#include <math.h>

#define F_NUM neq[4]

#define NSTATES 2
#define NINPUTS 1
#define NLEEC 2 /* vínculos igualdade terminais não-lineares */
#define N_PARAMS 9 /* número de parâmetros */
#define FNUM neq[4] /* reservado */

static double m,Gr,L,I,x1f,x2f,Q11,Q22,R11; /* parâmetros */

void activate(message)
char **message;
{
  *message = "tlglol";
}

void init(neq,params)
int neq[];
double *params;
{
  if ( params == 0 ) {
    neq[0] = NSTATES;
    neq[1] = NINPUTS;
    neq[2] = N_PARAMS;
    neq[10] = NLEEC;
  }
  else {
    m = params[0];
    Gr = params[1];
    L = params[2];
    I = params[3];
    x1f = params[4];
    x2f = params[5];
    Q11 = params[6];
    Q22 = params[7];
    R11 = params[8];
    return;
  }
}

```

```

void h(neq,t,x,u,xdot)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],xdot[NSTATES];
{
    xdot[0] = x[1];
    xdot[1] = (m*Gr*L/I)*sin(x[0])+u[0];
}

void Dh(neq,t,x,u,h_x,h_u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS];
    double h_x[NSTATES][NSTATES],h_u[NSTATES][NINPUTS];
{
    h_x[0][1] = 1.0;
    h_x[1][0] = (m*Gr*L/I)*cos(x[0]);

    h_u[1][0] = 1.0;
}

double l(neq,t,x,u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS];
{
    return Q11*x[0]*x[0]+Q22*x[1]*x[1]+R11*u[0]*u[0];
}

double Dl(neq,t,x,u,l_x,l_u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],l_x[NSTATES],l_u[NINPUTS];
{
    l_x[0] = 2*Q11*x[0];
    l_x[1] = 2*Q22*x[1];
    l_u[0] = 2*R11*u[0];
}

double g(neq,t,x0,xf)
    int neq[];
    double *t,x0[NSTATES],xf[NSTATES];
{
    switch (FNUM) {
    case 1:
        return 0.0;
        break;
    case 2:
        return xf[0] - x1f;
        break;
    case 3:
        return xf[1] - x2f;
        break;
    }
}

double Dg(neq,t,x0,xf,J_x0,J_xf)
    int neq[];
    double *t,x0[NSTATES],xf[NSTATES],J_x0[NSTATES],J_xf[NSTATES];
{
    if (F_NUM == 2)
        J_xf[0] = 1.0;
    if (F_NUM == 3)
        J_xf[1] = 1.0;
    return 0.0;
}

```

O ângulo inicial do pêndulo e sua velocidade angular foram definidos nas simulações como $\pi/4$ e 0. A estimativa inicial do controle foi determinada a partir de uma lei de realimentação tipo LQR com matrizes de ponderação $Q=[1 \ 0;0 \ 1]$ e $R=1$. Foram

utilizadas splines de 2ª ordem para definição do espaço de controle e método Runge-Kutta de 4ª ordem (RK4) para integração das equações dinâmicas. Os resultados estão mostrados na Figura 5.2. Para maiores detalhes a respeito da sintaxe de implementação do OCP, bem como dos fundamentos matemáticos do método, remetemos a Schwartz et al. (1997) e ao Capítulo 4 deste trabalho. Foram utilizadas as tolerâncias numéricas padrão da rotina tipo SQP NPSPOL (Gill et al., 1998) no RIOTS, isto é, 10^{-6} para a otimização e 10^{-4} para a violação de vínculos não-lineares.

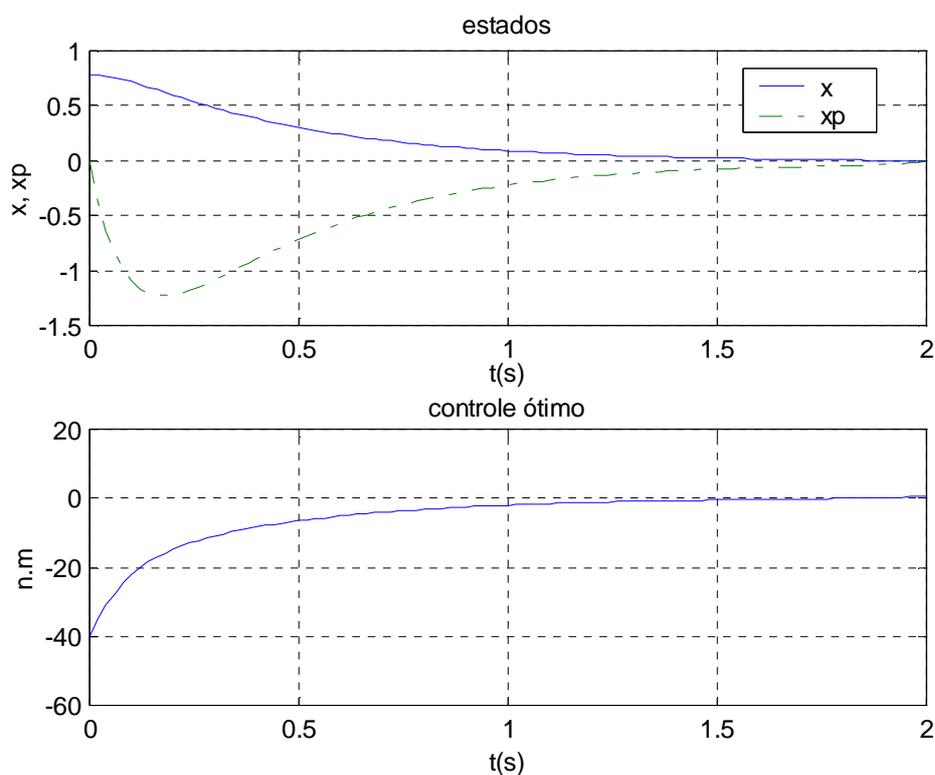


Figura 5.2: Pêndulo simples com tempo final fechado

Os testes comparativos de desempenho realizados num microcomputador Pentium III 600Mhz indicaram 2,5287 minutos de CPU para o OCP escrito em Matlab e 0,0973 minutos em C. Desse modo, o ganho de desempenho em tempo de processamento

utilizando a linguagem C é da ordem de **26 vezes**. Por isso, nas demais simulações realizadas neste trabalho, optou-se preferencialmente pelo C.

5.1.1 Experimentos com tempo final aberto

Pode-se definir o OCP de tal maneira que o tempo final da trajetória seja uma variável a ser otimizada, condição especialmente necessária quando a função de custo depende do tempo, como, por exemplo, nos problemas de tempo mínimo. Nestes casos, a formulação do OPC original deve sofrer algumas modificações, segundo o que foi indicado no item 4.15. Assim, o problema do pêndulo simples passa a ser definido pelo seguinte conjunto de equações diferenciais:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \left(\frac{mgL}{I} \sin(x_1) + u \right) \\ \dot{x}_3 = 0 \end{cases} \quad \text{Eq. 5.6}$$

em que x_3 é o fator de escala β que multiplica o tempo final, isto é, $t_f = t \beta$. Utilizando as mesmas condições de discretização, interpolação, função de custo e tolerância do problema com tempo final especificado em 2 s, os resultados estão mostrados na Figura 5.3. Escrevendo as funções de definição do OCP apenas em Matlab, o tempo de CPU foi 70 minutos e a função de custo ficou em $f = 262.7421$. O fator β ficou em 0.9395. Entretanto, o OCP não convergiu segundo a tolerância especificada, mesmo após 4200 iterações. No caso com tempo final definido em 2 s, que demorou 2.47 minutos e convergiu em 256 iterações, a função de custo ficou em $f = 243.4944$. Pode-se observar, entretanto, uma redução, no problema com tempo final aberto, do momento máximo no início do movimento. De qualquer modo, a introdução do tempo final aberto, mesmo em sistemas simples, parece trazer consigo diversos problemas de ordem numérica, que em sistemas complexos podem causar dificuldades de tratamento.

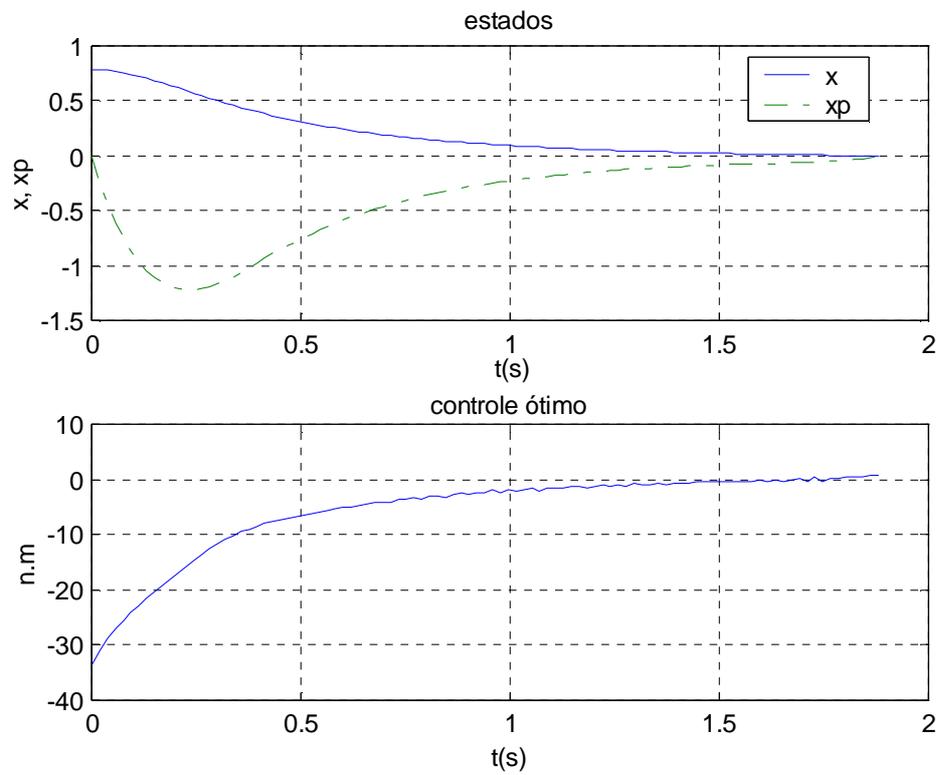


Figura 5.3: pêndulo de 1 grau de liberdade e tempo final aberto

5.2 Pêndulo triplo com atuadores de torque sem dinâmica e dinâmicos

5.2.1 Atuadores de torque sem dinâmica⁵²

Utilizando agora sistemas de múltiplos corpos rígidos para tratar uma versão simplificada do problema do controle postural, foram realizados testes com um pêndulo triplo, no plano sagital, invertido, tendo como articulações o tornozelo, o joelho e o quadril (Figura 5.4). Detalhes sobre a formulação deste modelo podem ser encontradas em Menegaldo (1997). Sua formulação está mostrada abaixo:

$$[M]\ddot{\mathbf{x}} + [C]\dot{\mathbf{x}}^2 + \mathbf{g} = [D]\boldsymbol{\tau} \quad \text{eq. 5.7}$$

em que as matrizes de massa $[M]$, de termos centrípetos $[C]$ e de transformação de torques articulares para momentos de corpo rígido $[D]$, assim como o vetor de termos gravitacionais \mathbf{g} são dados por:

$$[M] = \begin{pmatrix} A_1 & A_2 \cos(x_1 - x_2) & A_3 \cos(x_1 - x_3) \\ A_2 \cos(x_1 - x_2) & A_4 & A_5 \cos(x_2 - x_3) \\ A_3 \cos(x_1 - x_3) & A_5 \cos(x_2 - x_3) & A_6 \end{pmatrix}$$

$$[C] = \begin{pmatrix} 0 & B_1 \sin(x_1 - x_2) & B_2 \sin(x_1 - x_3) \\ -B_1 \sin(x_1 - x_2) & 0 & B_3 \sin(x_2 - x_3) \\ -B_2 \sin(x_1 - x_3) & -B_3 \sin(x_2 - x_3) & 0 \end{pmatrix}$$

$$[D] = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{g} = \begin{pmatrix} -G_1 \sin(x_1) \\ -G_2 \sin(x_2) \\ -G_3 \sin(x_3) \end{pmatrix}$$

⁵² O termo *atuador sem dinâmica* significa que a variável de controle $u(t)$ é o próprio torque externo na articulação, sem o atraso a que estão sujeitos os atuadores *dinâmicos*.

Os coeficientes indicados são:

$$A_1 = I_A + L_{Acm}^2 m_A + L_A^2 m_B + L_A^2 m_C$$

$$A_2 = L_A (L_B m_C + L_{Bcm} m_B)$$

$$A_3 = L_A L_{Ccm} m_C$$

$$A_4 = I_B + L_{Bcm}^2 m_B + L_B^2 m_C$$

$$A_5 = L_B L_{Ccm} m_C$$

$$A_6 = I_C + L_{Ccm}^2 m_C$$

$$B_1 = L_A (L_B m_C + L_{Bcm} m_B)$$

$$B_2 = L_A L_{Ccm} m_C$$

$$B_3 = L_B L_{Ccm} m_C$$

$$G_1 = g (L_{Acm} m_A + L_A m_B + L_B m_C)$$

$$G_2 = g (L_{Bcm} m_B + L_B m_C)$$

$$G_3 = g L_{Ccm} m_C$$

Em espaço de estados, pode-se sintetizar as equações de movimento por:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ M(x_1, x_2, x_3)^{-1} \left([D]\tau - [C(x_1, x_2, x_3)] \begin{pmatrix} x_4^2 \\ x_5^2 \\ x_6^2 \end{pmatrix} - G(x_1, x_2, x_3) \right) \end{pmatrix} \quad \text{eq. 5.8}$$

Utilizando atuadores de torque sem dinâmica, as variáveis de controle são os próprios elementos do vetor τ : $\tau = [U_1 \ U_2 \ U_3]^T$.

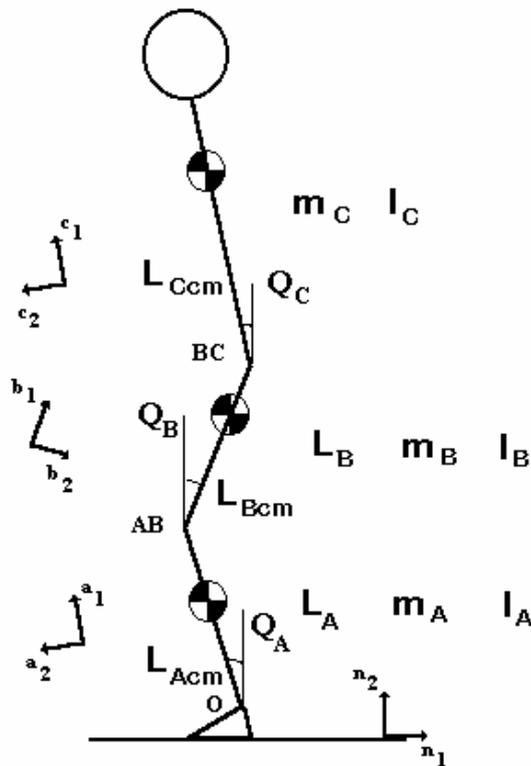


Figura 5.4: Modelo de três corpos rígidos da postura humana

O OCP pode então ser formulado como:

OCP: determinar o controle $u(t)$ que minimiza o funcional

$$f(u, \xi) = \int_0^{t_f} Q_1 x_1^2 + Q_2 x_2^2 + Q_3 x_3^2 + Q_4 x_4^2 + Q_5 x_5^2 + Q_6 x_6^2 + R_1 U_1^2 + R_2 U_2^2 + R_3 U_3^2 dt$$

sujeito a: $\dot{x} = h(t, x, u), \quad x(0) = \xi, \quad t \in [0, t_f]$

$$g_{ee}(x(t_f)) = 0$$

em que $h(t, x, u)$ é definido pela eq. 5.8 e $x_i(t_f) = 0, i = 1, 6$

Para definir o problema computacionalmente, utiliza-se inicialmente uma rotina em Matlab onde se definem os parâmetros do sistema a ser controlado, que gera suas equações diferenciais e as lineariza. Esta rotina utiliza o Matlab Symbolic Processing Toolbox, tanto na geração das equações quanto na sua transformação para a sintaxe da linguagem C.

alg6.m

```
% Determinação das equações algébricas do problema do pêndulo triplo controlado por atuadores de torque sem dinâmica
clear global
clear all
```

Aqui são definidos os parâmetros antropométricos do indivíduo a partir do qual é formulado o modelo (ver Menegaldo, 1997)

```
mA=6.7064;
mB=14.3536;
mC=44.6303;
lA=0.4750;
lB=0.3950;
lAcm=0.2746;
lBcm=0.2417;
lCcm=0.3085;
IA=0.1295;
IB=0.2640;
IC=2.2528;
g=9.81;

Termos das equações do sistema dinâmico

A1=(IA+lAcm^2*mA+lA^2*mB+lA^2*mC);
A2=lA*(lB*mC+mB*lBcm);
A3=lA*lCcm*mC;
A4=(IB+lBcm^2*mB+lB^2*mC);
A5=lB*lCcm*mC;
A6=(IC+lCcm^2*mC);

B1=lA*(mC*lB+lBcm*mB);
B2=lA*lCcm*mC;
B3=lB*mC*lCcm;

G1=g*(lAcm*mA+lA*mB+lA*mC);
G2=g*(lBcm*mB+lB*mC);
G3=g*lCcm*mC;

lC=1.8*lCcm;
lP1=-0.20; %comprimento do pé (p/frente, a partir do centro do tornozelo)
lP2=0.05; %p/trás
mtot=70; %massa total do indivíduo

Geração das equações do sistema dinâmico

pr=16; %precisão do truncamento numérico na geração das expressões algébricas

syms x1 x2 x3 x4 x5 x6 U1 U2 U3 real

% matrizes do sistema dinâmico
M=vpa([A1 A2*cos(x1-x2) A3*cos(x1-x3)
```

```

A2*cos(x1-x2) A4 A5*cos(x2-x3)
A3*cos(x1-x3) A5*cos(x2-x3) A6],pr);

C=vpa([0 B1*sin(x1-x2) B2*sin(x1-x3)
-B1*sin(x1-x2) 0 B3*sin(x2-x3)
-B2*sin(x1-x3) -B3*sin(x2-x3) 0],pr);

G=vpa([-G1*sin(x1)
-G2*sin(x2)
-G3*sin(x3)],pr);

u=[U1
U2
U3];

Mi=inv(M);

D=[1 -1 0
0 1 -1
0 0 1];

xp=vpa([x4
x5
x6
Mi*(D*u-C*[x4^2 x5^2 x6^2]'-G)],pr);

xp1=vpa([x4
x5
x6
Mi*(-C*[x4^2 x5^2 x6^2]'-G)],pr);

xp2=vpa([0 0 0
0 0 0
0 0 0
Mi*D],pr);

%Determinação do sistema linear equivalente
%cáculo do jacobiano

for i=1:length(xp),
for j=1:length(xp),
T(i,j)=vpa(diff(xp(i),strcat('x',num2str(j))),pr);
T1(i,j)=vpa(diff(xp1(i),strcat('x',num2str(j))),pr);
end
end

for i=1:length(xp),
for j=1:length(u),
Jxp_u(i,j)=vpa(diff(xp(i),strcat('U',num2str(j))),pr);
end
end

A=T1
B=xp2;
Jxp=T;
save res_analit A B xp Jxp Jxp_u u
save params_mod A1 A2 A3 A4 A5 A6 B1 B2 B3 G1 G2 G3 g mtot lC lP1 lP2 lA lB lAcm lBcm lCcm
mA mB mC

break

Geração das equações em sintaxe C

xpc=ccode(xp)
fid=fopen('xp_c','w')
fprintf(fid,xpc)
fclose(fid)

Jxpc=ccode(Jxp)
fid=fopen('Jxp_c','w')
fprintf(fid,Jxpc)
fclose(fid)

Jxp_uc=ccode(Jxp_u)

```

```
fid=fopen('Jxp_uc','w')
fprintf(fid,Jxp_uc)
fclose(fid)
```

A seguir, outra rotina em Matlab define os parâmetros de discretização e controle do problema:

t3glolc.m

```
% tese: pendulo invertido com 3 gl, acionador de torque, malha aberta
% 1ª parte: GERAÇÃO DO DLL

Esta parte do programa copia os arquivos necessários até o diretório do RIOTS onde são
vinculados e compilados. Posteriormente, o arquivo gerado é copiado de volta para o
diretório de trabalho.

clear global
clear all
nome=char('t3glol');
dirsaida=('e:\user\luciano\riots\tese\t3glolc3');
dirinstal=('e:\user\luciano\riots\riots_95\systems');
nomec=strcat(nome,'.c');
nomedll=strcat(nome,'.dll');
eval(['delete ',nomedll]);
delete simulate.dll
eval(['chdir ',dirinstal]);
eval(['delete ',nomedll]);
comando=char(['c:\matlab\bin\mex -V4 -v ',dirsaida,'\ ',nomec,' simulate.obj utility.obj
..\f2c\libf2c.lib ..\drivers\drivers.lib'])
fid=fopen('cel.bat','w')
fprintf(fid,'%c',comando)
fclose(fid)
disp('tecle cel.bat na janela dos')
pause
eval(['!copy ',nomedll,' ',dirsaida])
eval(['chdir ',dirsaida])
eval(['!rename ',nomedll,' simulate.dll'])

clear global

%2ª PARTE: ESCOLHA DOS PARÂMETROS DO SISTEMA E DE DISCRETIZAÇÃO

global A1 A2 A3 A4 A5 A6 B1 B2 B3 G1 G2 G3 g mtot lC lP1 lP2 lA lB lAcm lBcm lCcm mA mB mC
global xp Jxp Jxp_u

load res_analit %arquivos gerados por alg6.m
load params_mod

Parâmetros de discretização e interpolação, condições iniciais e vínculos de controle

N=100; %número de pontos da malha
tf=0.6; %tempo final
T=tf;
t=[0:T/N:T];
r=4; %ordem da spline
u0=zeros(3,N+r-1);

x0=[pi/4 -pi/4 0 0 0 0]';
%uboundl=[-1;-1];
%uboundh=[1;1];
uboundl=[];
uboundh=[];
```

```

figure(1)
clf
title('posição do VRS e do CM')
axis([-1.05 1.05 -0.1 2])
[d0,tornozelo,joelho,quadril,cabeca]=VRS(x0);
line([tornozelo(1) joelho(1) quadril(1) cabeca(1)],[tornozelo(2) joelho(2) quadril(2)
cabeca(2)]);
line([tornozelo(1) lP1 lP2],[tornozelo(2) 0 0])
[xcm0,ycm0]=CM(x0)
hold on
plot(d0,0,'ro')
plot(xcm0,ycm0,'go')
grid
hold off

```

Cálculo das matrizes [A] e [B] da equação linear de estados

```

x_op=[0 0 0 0 0 0]'; %ângulo de operação na linearização
x1=x_op(1);
x2=x_op(2);
x3=x_op(3);
x4=x_op(4);
x5=x_op(5);
x6=x_op(6);

```

```

for i=1:length(xp),
    for j=1:length(xp),
        Al(i,j)=eval(A(i,j));
    end
end

```

```

for i=1:length(xp),
    for j=1:length(u),
        Bl(i,j)=eval(B(i,j));
    end
end

```

```

global K Q R

```

Geração da estimativa inicial do vetor de controles utilizando LQR

```

Q=1*[10 0 0 0 0 0
      0 10 0 0 0 0
      0 0 10 0 0 0
      0 0 0 1 0 0
      0 0 0 0 1 0
      0 0 0 0 0 1];

```

```

R=[1 0 0
   0 1 0
   0 0 1];

```

```

[K,S,E]=lqr(Al,Bl,Q,R);
[t,x_aux] = ode23('Faux_u0',t,x0);
u0_aux=-K*x_aux';
t=t';
figure(1)
subplot(211)
plot(t,x_aux(:,1:6)*180/pi)
title('estados - só realimentação')
ylabel('x, xp')
xlabel('t(s)')
grid
subplot(212)
plot(t,u0_aux)
title('estimativa inicial do controle')
ylabel('n.m')
xlabel('t(s)')
zoom on
grid

```

Vínculos terminais

```

xlf=0;

```

```

x2f=0;
x3f=0;
x4f=0;
x5f=0;
x6f=0;

global params

params=[x1f x2f x3f x4f x5f x6f Q(1,1) Q(2,2) Q(3,3) Q(4,4) Q(5,5) Q(6,6) R(1,1) R(2,2)
R(3,3)];
[info,simed]=simulate(0,params)
simulate(0)
[f,x]=simulate(1,x0,u0_aux,t,4,2)

Solução do OCP

med_tempo=cpu_time;
[uol,xol,fol]=riots(x0,u0_aux,t,uboundl,uboundh,[params],1000,4);
tempo_cpu=(cpu_time-med_tempo)/60

save resp t Q R tf xol uol fol g N r tempo
save uol

```

t3glol.c

```

#include <math.h>
#define F_NUM neq[4]
#define NSTATES 6
#define NINPUTS 3
#define NLEEC 6 /* vínculos igualdade terminais não-lineares */
#define N_PARAMS 15 /* número de parâmetros */
#define FNUM neq[4] /* reservado */
#define x1 x[0]
#define x2 x[1]
#define x3 x[2]
#define x4 x[3]
#define x5 x[4]
#define x6 x[5]
#define U1 u[0]
#define U2 u[1]
#define U3 u[2]
double s1,s2,s3,s4,s5,s6,s7;
static double x1f,x2f,x3f,x4f,x5f,x6f,Q11,Q22,Q33,Q44,Q55,Q66,R11,R22,R33; /* parâmetros */

void activate(message)
char **message;
{
    *message = "t3glol";
}

void init(neq,params)
int neq[];
double *params;
{
    if ( params == 0 ) {
        neq[0] = NSTATES;
        neq[1] = NINPUTS;
        neq[2] = N_PARAMS;
        neq[10] = NLEEC;
    }
    else {
        x1f = params[0];
        x2f = params[1];
        x3f = params[2];
        x4f = params[3];
        x5f = params[4];
        x6f = params[5];
        Q11 = params[6];
        Q22 = params[7];
        Q33 = params[8];
        Q44 = params[9];
        Q55 = params[10];
    }
}

```

```

    Q66 = params[11];
    R11 = params[12];
    R22 = params[13];
    R33 = params[14];
    return;
}
}

```

Equações diferenciais

```

void h(neq,t,x,u,xdot)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],xdot[NSTATES];
{
    xdot[0] = x4;
    xdot[1] = x5;
    xdot[2] = x6;
    s1 = (-0.5243171829129E2+0.2957768233188618E2*pow(cos(x2-x3),2.0))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(U1-U2-0.100216609695E2*sin(x1-
x2)*x5*x5-0.654001258625E1*sin(x1-x3)*x6*x6+0.2929161027114E3*sin(x1));
    s3 = -(-0.6514446492291323E2*cos(x1-x2)+0.3556809900669858E2*cos(x1-x3)*cos(x2-x3))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(U2-U3+0.100216609695E2*sin(x1-
x2)*x4*x4-0.543853678225E1*sin(x2-x3)*x6*x6+0.2069736718122E3*sin(x2));
    s4 = -(0.5450317180186495E2*cos(x1-x2)*cos(x2-x3)-0.5275150566824476E2*cos(x1-x3))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(U3+0.654001258625E1*sin(x1-
x3)*x4*x4+0.543853678225E1*sin(x2-x3)*x5*x5+0.1350684704655E3*sin(x3));
    s2 = s3+s4;
    xdot[3] = s1+s2;

    etc...
}

```

Matrizes Jacobianas

```

void Dh(neq,t,x,u,h_x,h_u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS];
    double h_x[NSTATES][NSTATES],h_u[NSTATES][NINPUTS];
{
    h_x[0][0] = 0.0;
    h_x[0][1] = 0.0;
    h_x[0][2] = 0.0;
    h_x[0][3] = 1.0;
    h_x[0][4] = 0.0;
    h_x[0][5] = 0.0;
    h_x[1][0] = 0.0;
    h_x[1][1] = 0.0;
    h_x[1][2] = 0.0;
    h_x[1][3] = 0.0;
    h_x[1][4] = 1.0;
    h_x[1][5] = 0.0;
    h_x[2][0] = 0.0;
    h_x[2][1] = 0.0;
    h_x[2][2] = 0.0;
    h_x[2][3] = 0.0;
    h_x[2][4] = 0.0;
    h_x[2][5] = 1.0;
    s3 = -(-0.5243171829129E2+0.2957768233188618E2*pow(cos(x2-x3),2.0))/pow(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0),2.0)*(U1-U2-0.100216609695E2*sin(x1-
x2)*x5*x5-0.654001258625E1*sin(x1-x3)*x6*x6+0.2929161027114E3*sin(x1))*(-
0.1305711482993843E4*cos(x1-x2)*sin(x1-x2)+0.7129028591494857E3*sin(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.7129028591494857E3*cos(x1-x2)*sin(x1-x3)*cos(x2-x3)-
0.689991022027918E3*cos(x1-x3)*sin(x1-x3));
    s4 = (-0.5243171829129E2+0.2957768233188618E2*pow(cos(x2-x3),2.0))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-

```

```

x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(-0.100216609695E2*cos(x1-
x2)*x5*x5-0.654001258625E1*cos(x1-x3)*x6*x6+0.2929161027114E3*cos(x1));
    s2 = s3+s4;
    s3 = s2;
    s5 = -(0.6514446492291323E2*sin(x1-x2)-0.3556809900669858E2*sin(x1-x3)*cos(x2-x3))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(U2-U3+0.100216609695E2*sin(x1-
x2)*x4*x4-0.543853678225E1*sin(x2-x3)*x6*x6+0.2069736718122E3*sin(x2));
    s6 = 1.0*(-0.6514446492291323E2*cos(x1-x2)+0.3556809900669858E2*cos(x1-x3)*cos(x2-
x3))/pow(-0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0),2.0)*(U2-U3+0.100216609695E2*sin(x1-
x2)*x4*x4-0.543853678225E1*sin(x2-x3)*x6*x6+0.2069736718122E3*sin(x2))*(-
0.1305711482993843E4*cos(x1-x2)*sin(x1-x2)+0.7129028591494857E3*sin(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.7129028591494857E3*cos(x1-x2)*sin(x1-x3)*cos(x2-x3)-
0.689991022027918E3*cos(x1-x3)*sin(x1-x3));
    s4 = s5+s6;
    s1 = s3+s4;
    s2 = s1-0.100216609695E2*(-0.6514446492291323E2*cos(x1-
x2)+0.3556809900669858E2*cos(x1-x3)*cos(x2-x3))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*cos(x1-x2)*x4*x4;
    s3 = s2-(-0.5450317180186495E2*sin(x1-x2)*cos(x2-x3)+0.5275150566824476E2*sin(x1-
x3))/(-0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*(U3+0.654001258625E1*sin(x1-
x3)*x4*x4+0.543853678225E1*sin(x2-x3)*x5*x5+0.1350684704655E3*sin(x3));
    s4 = s3;
    s6 = 1.0*(0.5450317180186495E2*cos(x1-x2)*cos(x2-x3)-0.5275150566824476E2*cos(x1-
x3))/pow(-0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0),2.0)*(U3+0.654001258625E1*sin(x1-
x3)*x4*x4+0.543853678225E1*sin(x2-x3)*x5*x5+0.1350684704655E3*sin(x3))*(-
0.1305711482993843E4*cos(x1-x2)*sin(x1-x2)+0.7129028591494857E3*sin(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.7129028591494857E3*cos(x1-x2)*sin(x1-x3)*cos(x2-x3)-
0.689991022027918E3*cos(x1-x3)*sin(x1-x3));
    s7 = -0.654001258625E1*(0.5450317180186495E2*cos(x1-x2)*cos(x2-x3)-
0.5275150566824476E2*cos(x1-x3))/(-0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0))*cos(x1-x3)*x4*x4;
    s5 = s6+s7;
    h_x[3][0] = s4+s5;

    etc....

    h_u[0][0] = 0.0;
    h_u[0][1] = 0.0;
    h_u[0][2] = 0.0;
    h_u[1][0] = 0.0;
    h_u[1][1] = 0.0;
    h_u[1][2] = 0.0;
    h_u[2][0] = 0.0;
    h_u[2][1] = 0.0;
    h_u[2][2] = 0.0;
    h_u[3][0] = (-0.5243171829129E2+0.2957768233188618E2*pow(cos(x2-x3),2.0))/(-
0.731078497251155E3+0.4124146271772962E3*pow(cos(x2-
x3),2.0)+0.6528557414969214E3*pow(cos(x1-x2),2.0)-0.7129028591494857E3*cos(x1-x2)*cos(x1-
x3)*cos(x2-x3)+0.344995511013959E3*pow(cos(x1-x3),2.0));

    etc....
}

```

Função objetivo

```

double l(neq,t,x,u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS];
{
return Q11*x1*x1+Q22*x2*x2+Q33*x3*x3+Q44*x4*x4+Q55*x5*x5+Q66*x6*x6+R11*U1*U1+R22*U2*U2
+R33*U3*U3;
}

```

Derivadas da função objetivo

```
double Dl(neq,t,x,u,l_x,l_u)
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],l_x[NSTATES],l_u[NINPUTS];
{
    l_x[0] = 2*Q11*x1;
    l_x[1] = 2*Q22*x2;
    l_x[2] = 2*Q33*x3;
    l_x[3] = 2*Q44*x4;
    l_x[4] = 2*Q55*x5;
    l_x[5] = 2*Q66*x6;
    l_u[0] = 2*R11*U1;
    l_u[1] = 2*R22*U2;
    l_u[2] = 2*R33*U3;
}
```

Vínculos terminais

```
double g(neq,t,x0,xf)
    int neq[];
    double *t,x0[NSTATES],xf[NSTATES];
{
    switch (FNUM) {
    case 1:
        return 0.0;
        break;
    case 2:
        return xf[0] - x1f;
        break;
    case 3:
        return xf[1] - x2f;
        break;
    case 4:
        return xf[2] - x3f;
        break;
    case 5:
        return xf[3] - x4f;
        break;
    case 6:
        return xf[4] - x5f;
        break;
    case 7:
        return xf[5] - x6f;
        break;
    }
}
```

Derivadas dos vínculos terminais

```
double Dg(neq,t,x0,xf,J_x0,J_xf)
    int neq[];
    double *t,x0[NSTATES],xf[NSTATES],J_x0[NSTATES],J_xf[NSTATES];
{
    if (F_NUM == 2)
        J_xf[0] = 1.0;
    if (F_NUM == 3)
        J_xf[1] = 1.0;
    if (F_NUM == 4)
        J_xf[2] = 1.0;
    if (F_NUM == 5)
        J_xf[3] = 1.0;
    if (F_NUM == 6)
        J_xf[4] = 1.0;
    if (F_NUM == 7)
        J_xf[5] = 1.0;
    return 0.0;
}
```

Existem inúmeras possibilidades de escolha dos parâmetros de discretização, interpolação, integração etc. Os resultados mostrados foram obtidos com 100 pontos de discretização, RK4, splines de 3ª ordem e com as mesmas condições iniciais. Não foi possível, neste estágio, trabalhar com tempos de simulação superiores a 0.6s, por razões que serão discutidas à frente. As figuras a seguir mostram resultados⁵³ obtidos com funções de custo quadrática da forma:

$$f(\mathbf{u}, \xi) = \int_0^{t_f} Q_1 x_1^2 + Q_2 x_2^2 + Q_3 x_3^2 + Q_4 x_4^2 + Q_5 x_5^2 + Q_6 x_6^2 + R_1 u_1^2 + R_2 u_2^2 + R_3 u_3^2 dt$$

utilizando $Q=[10 \ 10 \ 10 \ 1 \ 1 \ 1]$, $R=[1 \ 1 \ 1]$; $Q=[1 \ 1 \ 1 \ 1 \ 1 \ 1]$, $R=[1 \ 1 \ 1]$; $Q=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$, $R=[1 \ 1 \ 1]$. Em geral, Q e R são matrizes quadradas diagonais. Neste texto, essas matrizes são expressas através dos vetores que formam a sua diagonal principal.

⁵³ A menos que se indique o contrário, as simulações foram realizadas num microcomputador Pentium III 600MHz, 512 MB RAM, Windows NT 4.0.

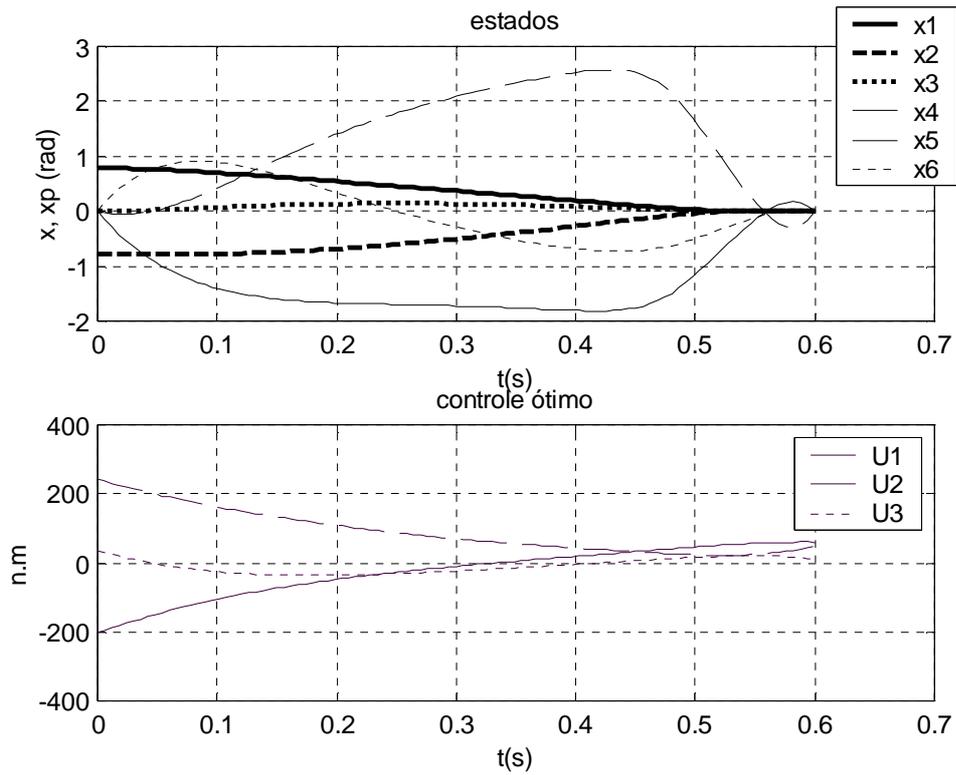


Figura 5.5: Pêndulo 3gl, atuadores de torque, $Q=[10 \ 10 \ 10 \ 1 \ 1 \ 1]$, $R=[1 \ 1 \ 1]$, tempo CPU=7.84 min.

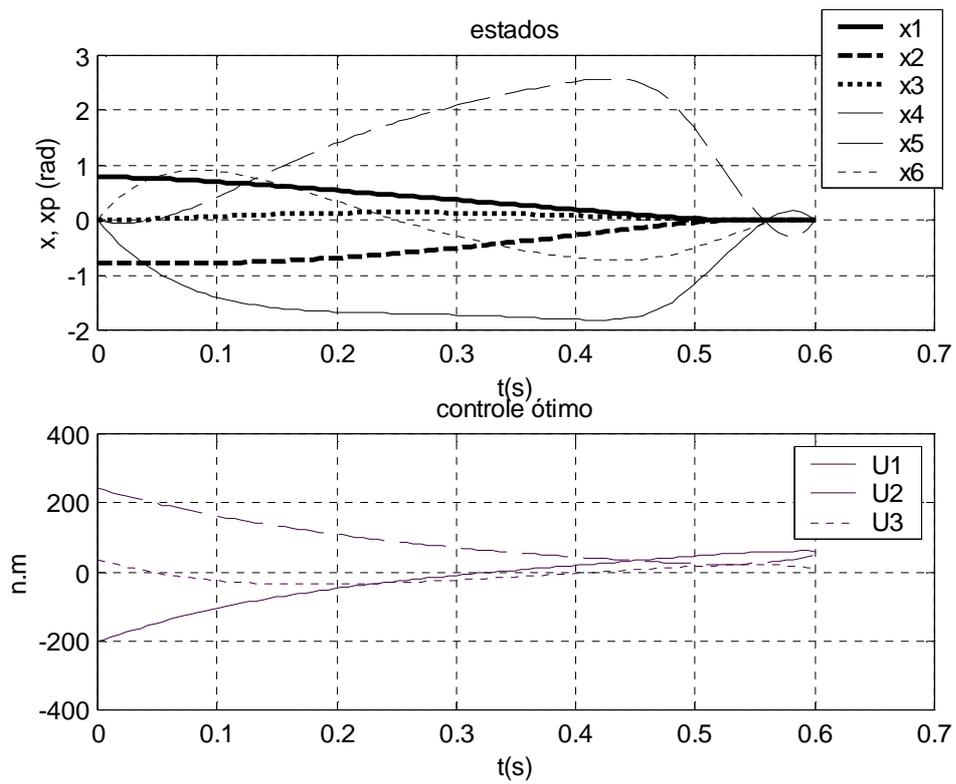


Figura 5.6: Pêndulo 3gl, atuadores de torque, $Q=[1 \ 1 \ 1 \ 1 \ 1 \ 1]$, $R=[1 \ 1 \ 1]$, tempo CPU=35.01 min.

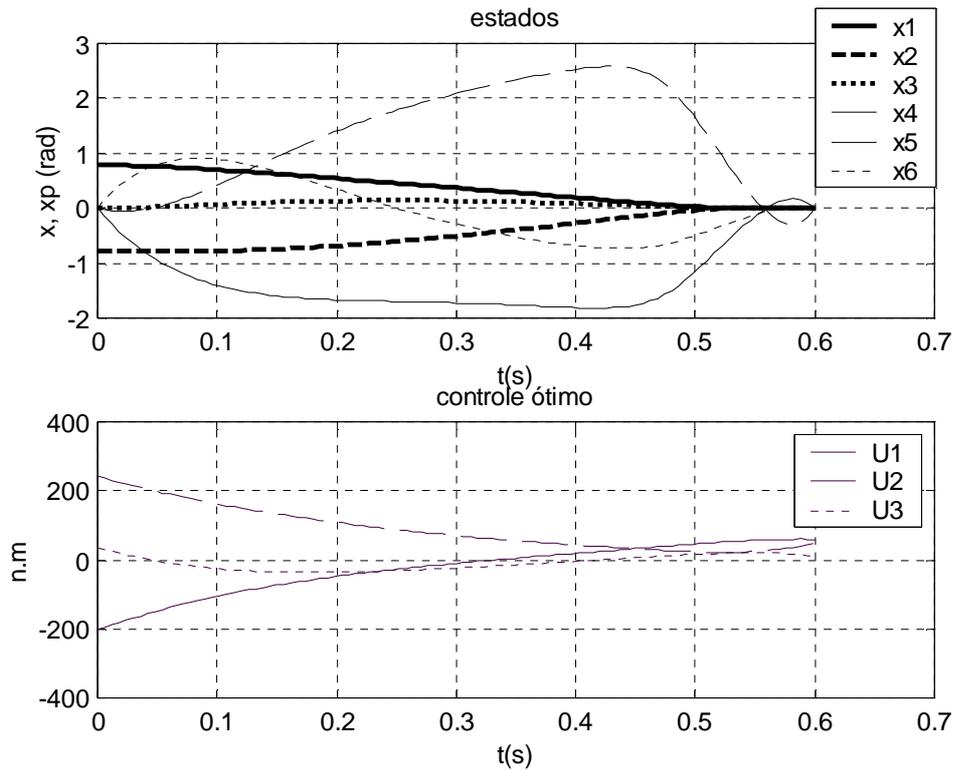


Figura 5.7: Pêndulo 3gl, atuadores de torque, $Q=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$, $R=[1 \ 1 \ 1]$, tempo CPU=11.09 min.

Existem outras possibilidades de escolha da função de custo. Por exemplo, pode-se maximizar a altura do centro de massa do corpo, ao mesmo tempo que se minimiza o seu deslocamento horizontal. Deste modo, a função de custo seria calculada como:

$$x_{cm} = -0.4545409282649036 \sin(x_1) - 0.3211773065429751 \sin(x_2) -$$

$$0.2095963566919316 \sin(x_3)$$

$$y_{cm} = 0.4545409282649036 \cos(x_1) + 0.3211773065429751 \cos(x_2) +$$

$$0.2095963566919316 \cos(x_3)$$

$$f(u, \xi) = \int_0^{t_f} -(y_{cm}/|y_{cm}|)y_{cm}^2 + x_{cm}^2 + R_1 U_1^2 + R_2 U_2^2 + R_3 U_3^2 dt$$

Os resultados obtidos com esta função de custo, nas mesmas condições do caso anterior a menos das condições iniciais e são mostrados na Figura 5.8. É interessante notar

o baixo custo computacional empregado quando esta função de custo foi utilizada. Entretanto, esta função possui o inconveniente, ao menos teórico, de não ser convexa em todo o domínio.

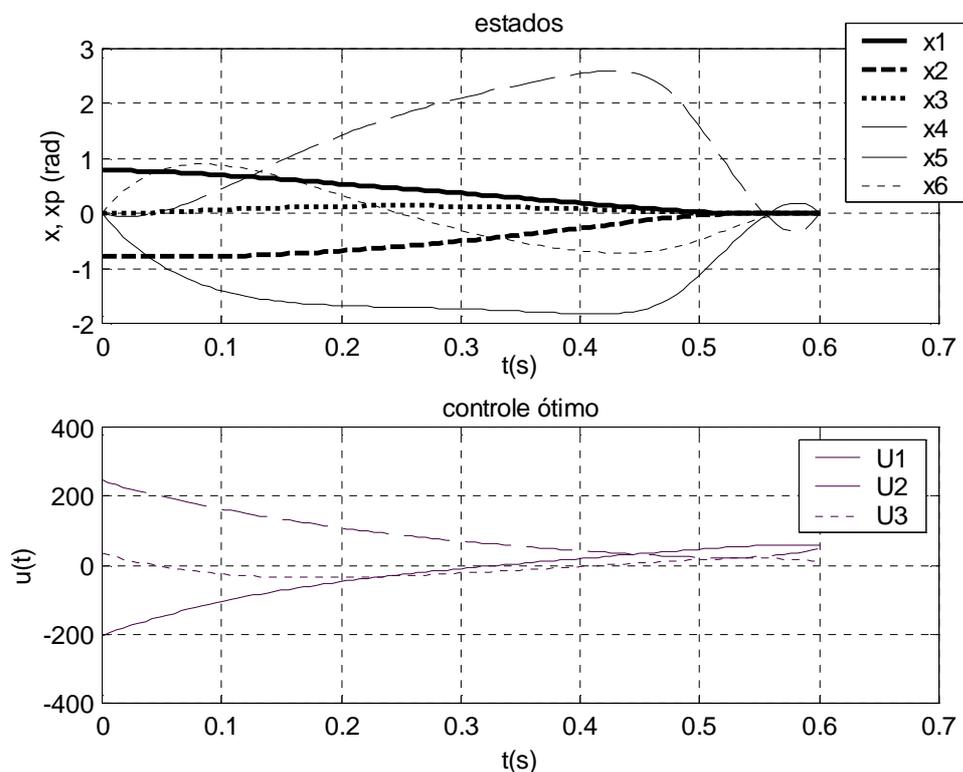


Figura 5.8: Pêndulo triplo com atuadores de torque sem dinâmica e função de custo dada pelo centro de massa. Tempo de CPU = 1.54 min.

5.2.2 Atuadores de torque dinâmicos lineares de 2ª ordem

A partir do modelo com atuadores de torque sem dinâmica, foi implementado um modelo com atuadores lineares de torque de 2ª ordem, isto é, contendo mais seis equações diferenciais, três da dinâmica da ativação e três da contração. Ao mesmo tempo, a variável de controle deixou de ser o próprio torque e passou a ser uma *ativação neuro-muscular equivalente*. Deste modo, foi possível abordar alguns dos problemas numéricos

introduzidos com esta nova dinâmica, sem entretanto ter a necessidade tratar, agora, os problemas relativos à redundância e saturação de atuadores.

Foram realizados diversos testes utilizando atuadores com a mesma estrutura e alguns dos parâmetros identificados no Capítulo 3, estabelecendo porém que os atuadores teriam 9300 N de força máxima e 7 cm de braço de momento. Os parâmetros de discretização, interpolação, integração, bem como as condições iniciais – a menos da estimativa inicial do vetor de controle – foram os mesmos utilizados na seção anterior. A formulação de estados deste modelo é a seguinte:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{pmatrix} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ M(x_1, x_2, x_3)^{-1} \left([D][rF_{om1} \ x_1 \ rF_{om2} \ x_2 \ rF_{om3} \ x_3] - [C(x_1, x_2, x_3)] \begin{pmatrix} x_4^2 \\ x_5^2 \\ x_6^2 \end{pmatrix} - G(x_1, x_2, x_3) \right) \\ -1/aa_2(aa_1 x_7) \\ -1/aa_2(aa_1 x_8) \\ -1/aa_2(aa_1 x_9) \\ 1/(2 aaf_2)(x_7 - aaf_1 x_{10}) \\ 1/(2 aaf_2)(x_8 - aaf_1 x_{11}) \\ 1/(2 aaf_2)(x_9 - aaf_1 x_{12}) \end{pmatrix}$$

eq. 5.9

As matrizes $[M]$, $[D]$, $[C]$, $\{G\}$ e os seus coeficientes são os mesmos da eq. 5.7. Os produtos $r_1 F_{om}$ foram definidos com o valor 651 m.N para os três eixos. Os parâmetros aa e aaf são os mesmos identificados no Capítulo 4.

Foram testadas diversas funções quadráticas tipo LQR, definidas anteriormente. Entretanto não foi possível em nenhum dos casos testados convergir para a tolerância padrão do RIOTS, mesmo após 2000 ou 3000 iterações. Um destes resultados está

mostrado na Figura 5.9, para $Q=[10 \ 10 \ 10 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ e $R=[1 \ 1 \ 1]$. Neste caso, não foram impostos limites ao controle, cujo módulo, a princípio, não deveria ultrapassar a unidade. Porém, os resultados apontam níveis de controle, especialmente no início do movimento, bastante acima deste limite.

Outra função de custo, bastante mais promissora, consiste na soma da derivada temporal dos momentos, isto é:

$$f(u, \xi) = \int_0^{t_f} \dot{\tau}_1^2 + \dot{\tau}_2^2 + \dot{\tau}_3^2 dt$$

utilizando as próprias expressões de derivadas de torque das equações de estado. Esta função de custo minimiza a variação do torque, maximizando assim sua suavidade (Pandy et. al., 1995). Os resultados obtidos com esta função de custo estão mostrados na Figura 5.10. É notável a economia computacional desta função de custo em relação à anterior.

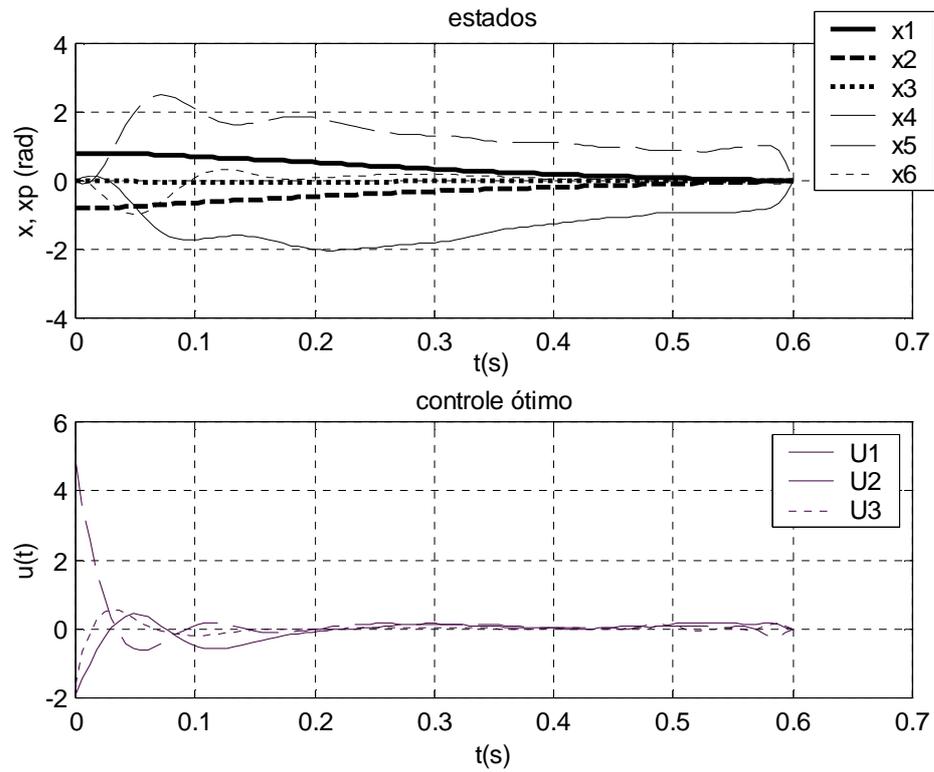


Figura 5.9: Estados e controle do pêndulo triplo com atuadores de torque lineares de 2ª ordem e função de custo tipo LQR com $Q=[10 \ 10 \ 10 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ e $R=[1 \ 1 \ 1]$. Tempo de CPU = 49.81 minutos, sem convergência.

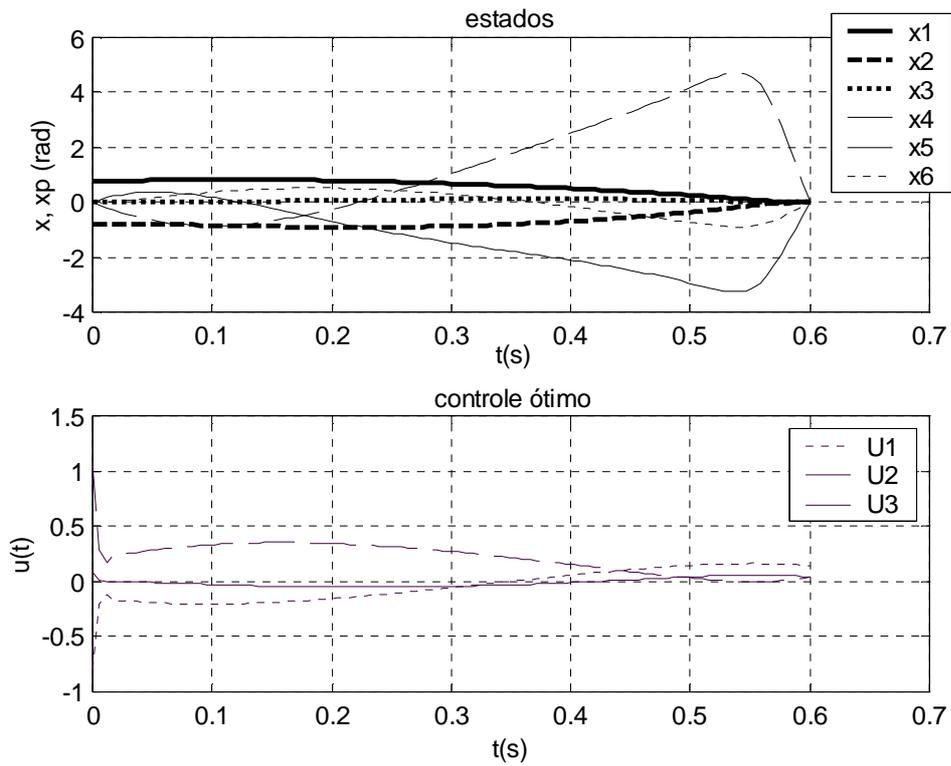


Figura 5.10: Estados e controle do pêndulo triplo com atuadores de torque lineares de 2ª ordem e função de custo = soma dos quadrados das derivadas temporais dos torques. Tempo de CPU = 4.06 minutos.

5.3 Pêndulo triplo com atuadores musculares

Uma vez determinada a metodologia para o problema dos atuadores dinâmicos de torque, a introdução de atuadores musculares, por sua própria natureza redundantes do ponto de vista mecânico, trouxe consigo dificuldades numéricas adicionais. Serão discutidos e mostrados a seguir alguns resultados obtidos com 8 atuadores lineares com parâmetros arbitrários mas plausíveis, sendo dois deles bi-articulares. Será abordado em seguida o problema do tempo crítico da simulação e mostrada uma estratégia para sua superação. Finalmente, serão mostrados os resultados obtidos utilizando atuadores não lineares, cujo emprego no problema em questão implica em problemas numéricos relativamente importantes.

5.3.1 Atuadores musculares lineares

Tendo em vista a realização de testes do algoritmo de controle, foi criado um modelo com 8 atuadores lineares, possuindo a seguinte matriz de braços de momento:

$$r = \begin{bmatrix} r_{11} & r_{12} & 0 & 0 & 0 & 0 & r_{17} & 0 \\ 0 & 0 & r_{23} & r_{24} & 0 & 0 & r_{27} & r_{28} \\ 0 & 0 & 0 & 0 & r_{35} & r_{36} & 0 & r_{38} \end{bmatrix}$$

cujos valores numéricos adotados foram:

$$r = \begin{bmatrix} 0.0630 & -0.0560 & 0 & 0 & 0 & 0 & 0.0960 & 0 \\ 0 & 0 & 0.0910 & -0.0560 & 0 & 0 & 0.0880 & -0.04000 \\ 0 & 0 & 0 & 0 & 0.0644 & -0.0756 & 0 & -0.0300 \end{bmatrix}$$

As forças máximas dos atuadores de 1 a 6 foi de 9300 N e dos dois últimos 8000 N e 9000 N respectivamente. Multiplicando-se a matriz de braços de momento por uma matriz 8x3 contendo três cópias do vetor das forças máximas, obtém-se a matriz rFom,

representado as efetivas capacidades de cada músculo produzir momento em cada uma dos corpos rígidos:

$$rFom = \begin{bmatrix} 585.9 & -520.8 & 0 & 0 & 0 & 0 & 768.0 & 0 \\ 0 & 0 & 846.3 & -520.8 & 0 & 0 & 704.0 & -360.0 \\ 0 & 0 & 0 & 0 & 598.92 & -703.08 & 0 & -270.0 \end{bmatrix}$$

Foram utilizados os parâmetros lineares da dinâmica da contração e da ativação obtidos no Capítulo 3 para todos os oito músculos. Depois de diversas tentativas preliminares, optou-se pela seguinte função de custo:

$$f(u, x(t_0)) = \int_0^2 \sum_{i=1}^8 \left(\dot{\tilde{F}}_i^2 + k u_i^2 \right) dt, \quad k=0.2$$

em que $\dot{\tilde{F}}_i$, $i=1, \dots, 8$ são as derivadas temporais de cada força muscular, isto é, a própria expressão da equação de estado correspondente, e u_i são os sinais de controle.

Entretanto, não foi possível aplicar diretamente a mesma metodologia e os mesmos parâmetros de discretização, interpolação e integração do problema dos atuadores dinâmicos de torque. Percebeu-se que, mesmo aumentando ao limite da memória da máquina a discretização da malha, com RK4 e diversas ordens de spline, o sistema não convergia. Uma das razões que possivelmente tenha levado a essas dificuldades é a utilização da estimativa inicial do controle baseada na lei de realimentação LQR. Estabelecendo um modelo contendo uma matriz de ganhos, durante a integração das equações diferenciais ao longo da solução do OCP, os métodos RK4, RK3 e RK1 (ver Press et al., 1992) fazem avaliações destas equações não só nos pontos t_k da malha de discretização, senão também em pontos intermediários $t_k + \Delta/2$, em que Δ é o intervalo de integração. Neste caso, o cálculo da variável de controle, a partir da matriz de ganhos, é feita a cada avaliação da equação diferencial pelo método RK. Em seguida, a trajetória final do controle pode ser calculada multiplicando-se a trajetória de estados pela matriz de

ganhos. Introduzindo esta trajetória final de controle como um sinal de malha aberta no mesmo sistema dinâmico, verificou-se que o comportamento do sistema divergia daquele observado com o sistema em malha fechada. Este efeito deve-se, provavelmente, à aproximação realizada pelo integrador implementado no RIOTS, que toma os valores do controle em pontos intermediários aos nós como os valores no nó à esquerda (Schwartz e Polak, 1996, Sec. 4). Na Figura 5.11 verifica-se um início de divergência das velocidades angulares.

Uma maneira empregada para tentar superar este efeito foi a utilização do RK1, ou método de Euler, que não requer a avaliação da equação diferencial fora dos nós da malha de discretização. De fato, utilizando este integrador, o que implica no emprego de uma malha mais refinada e na interpolação do sinal de controle apenas por funções constantes por partes, as respostas em malha aberta e fechada coincidem. Mesmo assim, o sistema não convergia.

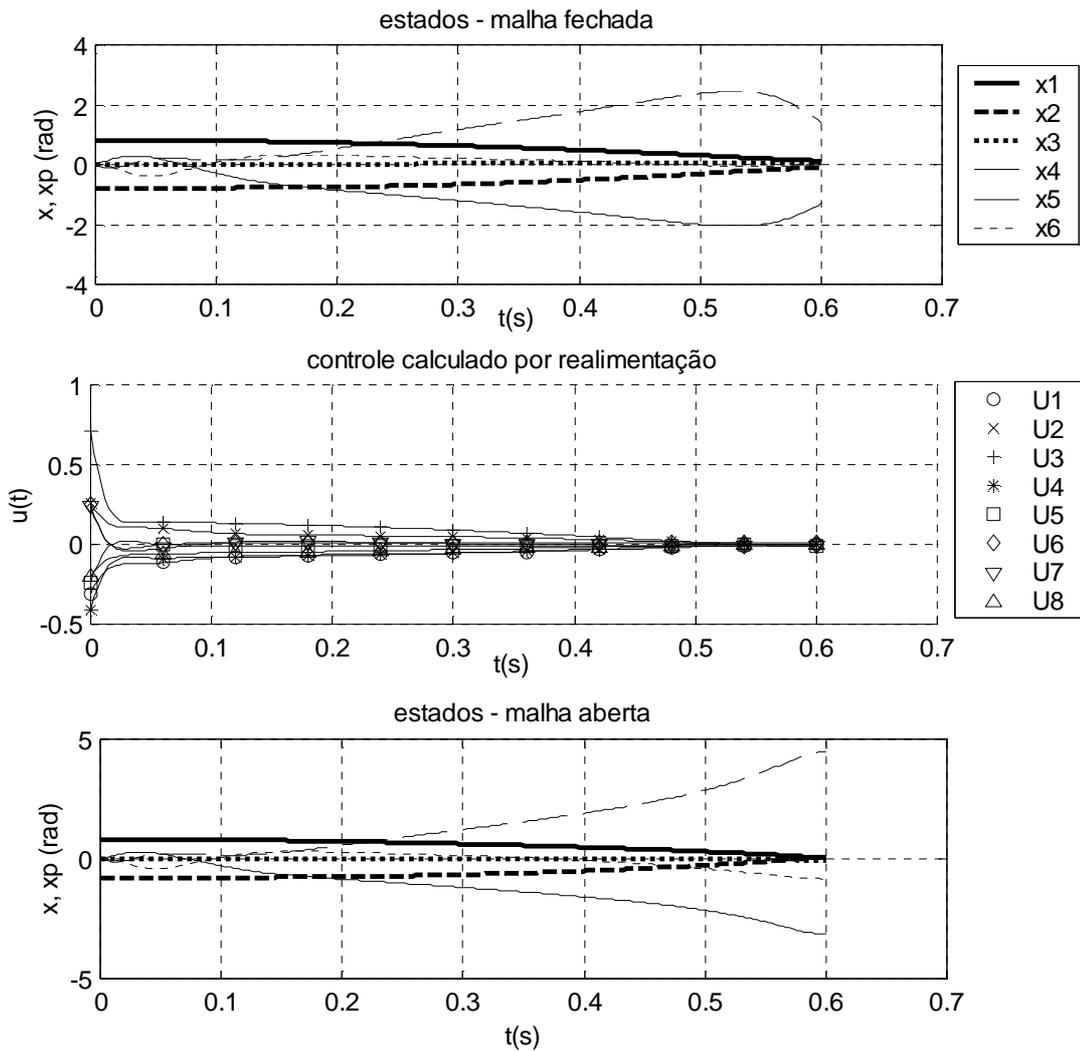


Figura 5.11: Comportamento do sistema controlado por malha de realimentação calculada por LQR. O gráfico superior mostra os estados do sistema em malha fechada, isto é, integrando as equações diferenciais associadas à matriz de ganhos. No gráfico central estão mostrados os sinais de controle calculados multiplicando-se os estados pelo matriz de ganhos. Em baixo, o sinal de controle foi introduzido no sistema em malha aberta.

Utilizando o RK1 e 200 pontos de discretização, foi possível calcular uma nova estimativa inicial para o controle, impondo vínculos terminais de **desigualdade** ($|x_1|, \dots, |x_6| \leq \pi/5$). Esta nova estimativa está mostrada na Figura 5.12. O controle ótimo obtido com as mesmas condições iniciais da Figura 5.12 e, finalmente, estabelecendo os vínculos

terminais de igualdade, estão mostrados na Figura 5.13. Para a condição inicial, foi utilizada a função de custo contanto apenas com as derivadas das forças musculares. Com esta função, entretanto, não era possível a convergência para os vínculos terminais de igualdade sem que fossem empregados os termos $0.2 U_i$, $i = 1, \dots, 8$.

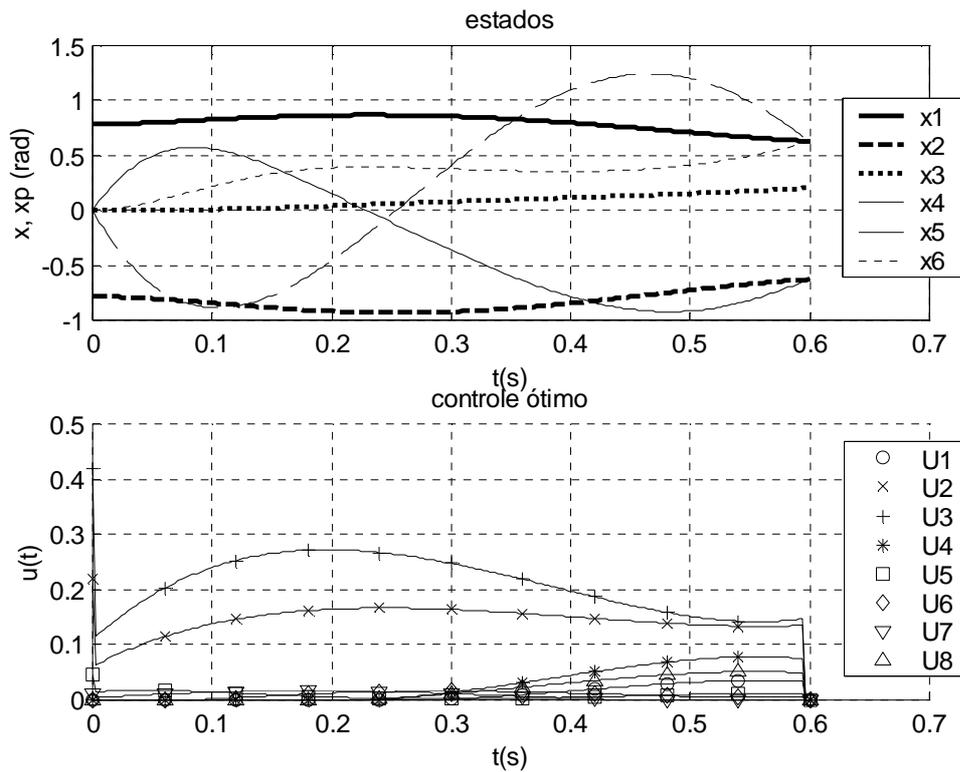


Figura 5.12: Estimativa inicial do controle calculada com RK1 utilizando vínculos de desigualdade. Tempo de CPU = 825.84 min (Pentium II 400Mhz).

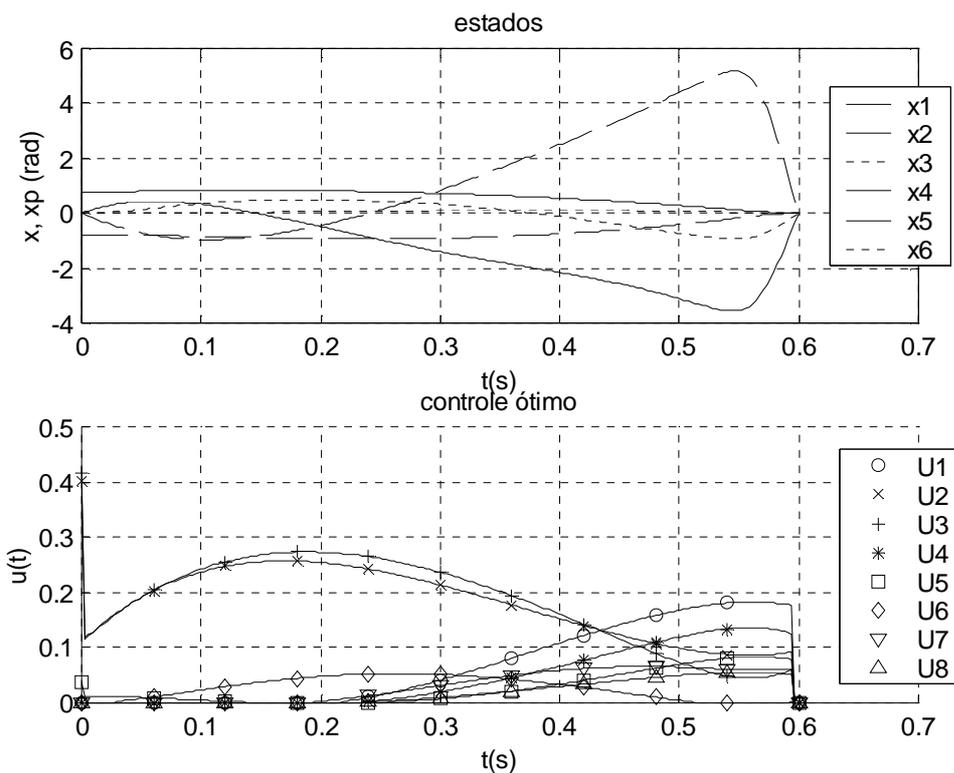


Figura 5.13: Solução do pêndulo triplo com 8 atuadores lineares, RK1. Tempo de CPU = 1.045,8 minutos (Pentium II 400Mhz).

Utilizando esta estimativa inicial a partir dos vínculos de desigualdade, foi possível empregar o RK4, mesmo com a degradação da condição inicial por LQR para o cálculo da condição inicial com vínculo de desigualdade (vide Figura 5.14 e Figura 5.15). Neste caso, percebeu-se um aumento significativo do tempo de CPU, tanto para a estimativa inicial quanto para a solução final. Entretanto, a função de custo que com o RK1 tinha ficado em 1.4262, reduziu-se para 1.3456 com o RK4.

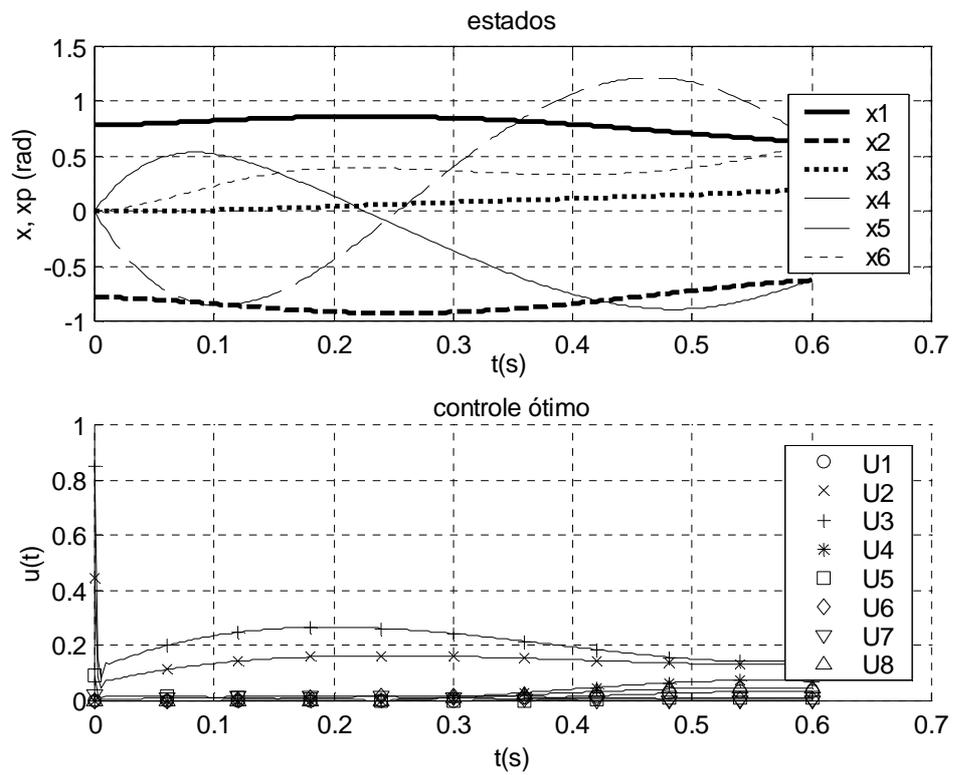


Figura 5.14: Estimativa inicial do controle calculada com RK4 utilizando vínculos de desigualdade. Tempo de CPU = 1.126,1 minutos (Pentium II 400Mhz).

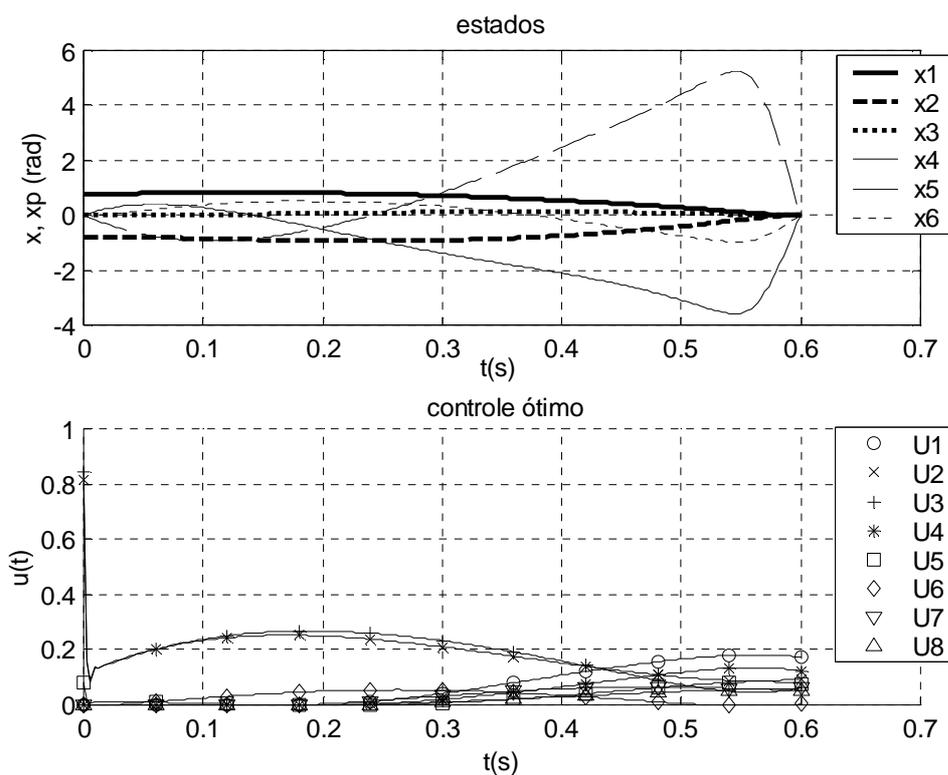


Figura 5.15 Solução do pêndulo triplo com 8 atuadores lineares, RK1. Tempo de CPU = 1274 minutos

5.3.2 Problema do tempo crítico

Em todos os modelos de pêndulo triplo estudados, seja com atuadores de torque sem dinâmica ou musculares dinâmicos, não foi possível estabelecer tempos de simulação superiores a um certo tempo crítico (t_{crit}), em geral, por volta de 0.6 segundos. As razões desta limitação não foram totalmente compreendidas, mas supõe-se que acima de t_{crit} os problemas causados pela inviabilidade das condições iniciais são agravados pelo comportamento dinâmico do sistema divergir da bacia de atração que o permitisse chegar aos estados correspondentes aos vínculos terminais que representam um ponto de equilíbrio instável. Em tempos de menor duração, o sistema pode em algumas iterações do processo de otimização apresentar soluções instáveis, mas sem terem tempo de simulação

suficiente para o pêndulo *cair*, a ponto de não poder recuperar-se novamente. Neste caso, a solução do OCP tornava-se inviável e, devido a imperfeições de implementação do RIOTS, a simulação apresentava um erro fatal de memória que requeria o encerramento do processo.

Para o caso dos atuadores musculares lineares, foi possível estabelecer uma metodologia que permitia a simulação do sistema por tempos superiores a t_{crit} . A solução adotada foi dividir o intervalo de simulação em dois ou mais, desde que o tempo de cada um fosse menor do que t_{crit} . Para o primeiro intervalo, eram estabelecidos vínculos de desigualdade terminais que representavam uma fração das condições iniciais. Por exemplo, para condições iniciais de $\pi/4$ e $-\pi/4$ para os ângulos do tornozelo e do joelho, respectivamente, estabelecia-se um vínculo terminal de desigualdade em que todos os estados, de deslocamento e velocidade, deveriam ter seus valores absolutos inferiores a $\pi/5$. A partir das condições finais do primeiro intervalo, era calculada uma nova solução num segundo intervalo de tempo, estabelecendo vínculos terminais de desigualdade mais estreitos, na faixa de $\pi/30$ para os mesmos estados. O resultado obtido com este processo (Figura 5.16) mostra que o sistema converge para os vínculos terminais estabelecidos, mas verifica-se uma transição abrupta do sinal de controle no encontro entre as duas soluções, pouco plausível do ponto de vista fisiológico.

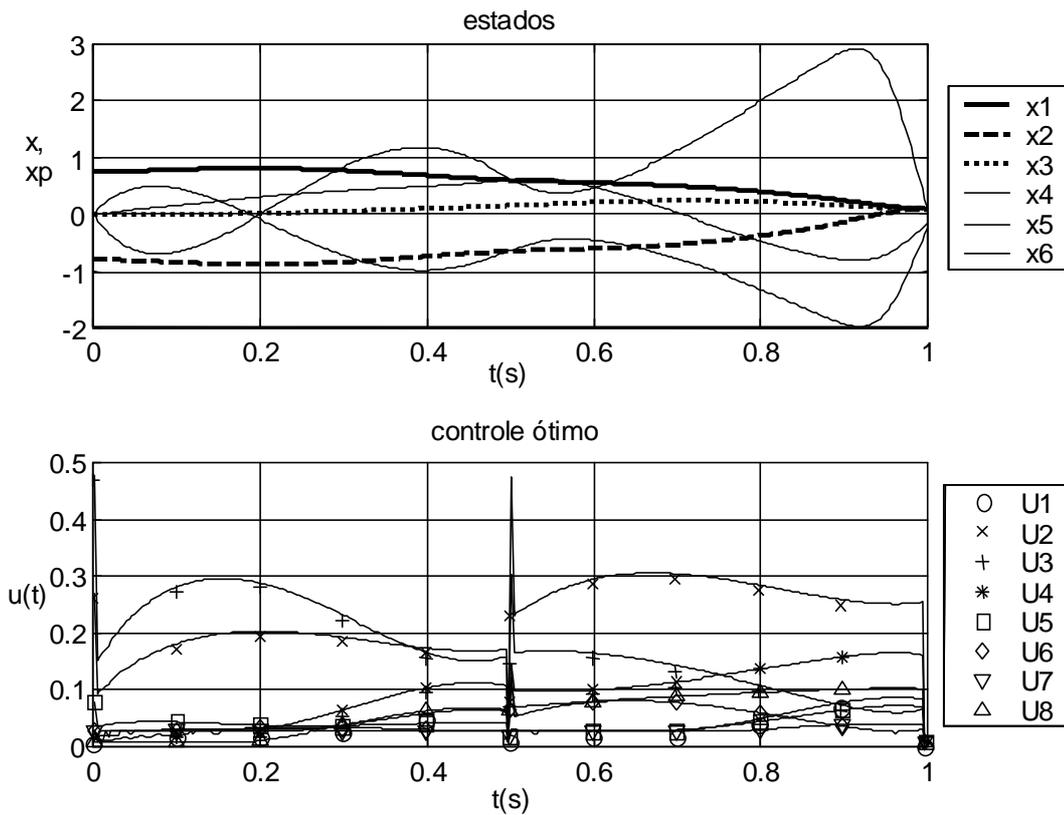


Figura 5.16: Solução para $t > t_{crit}$, em dois intervalos, com vínculos de desigualdade e RK1.

É importante observar que estes resultados foram obtidos com integrador RK1. Caso contrário, utilizando-se RK de ordem superior, é possível obter uma solução aparentemente semelhante. Entretanto, quando foi feita uma nova integração das equações diferenciais com este mesmo sinal de controle, a solução se tornava instável no segundo trecho, afastando-se dos vínculos terminais. Este fenômeno pode ser atribuído novamente aos termos do RK calculados em intervalos de tempo diferentes dos nós da malha, isto é, em $t_k + \Delta/2$. Considerando que o primeiro ponto do segundo intervalo é o mesmo do último ponto do primeiro intervalo, isto é, a condição inicial do 2º é a final do 1º, este último ponto foi calculado levando em conta as informações do nó precedente, já que o RK utiliza as informações do ponto $i-1$ para calcular o valor da equação diferencial integrada

no ponto i . No caso do primeiro ponto do segundo intervalo, a mera repetição do ponto anterior significa uma inconsistência para o método naquela iteração, introduzindo uma perturbação numérica fatal para a estabilidade das equações, de natureza altamente instável e não-linear. Com o RK1 esta inconsistência não se verifica.

Utilizando o controle calculado acima como condição inicial para a solução do OCP no tempo completo, pode-se obter o resultado mostrado na Figura 5.17. Deve-se observar, entretanto, o alto custo computacional desta solução.

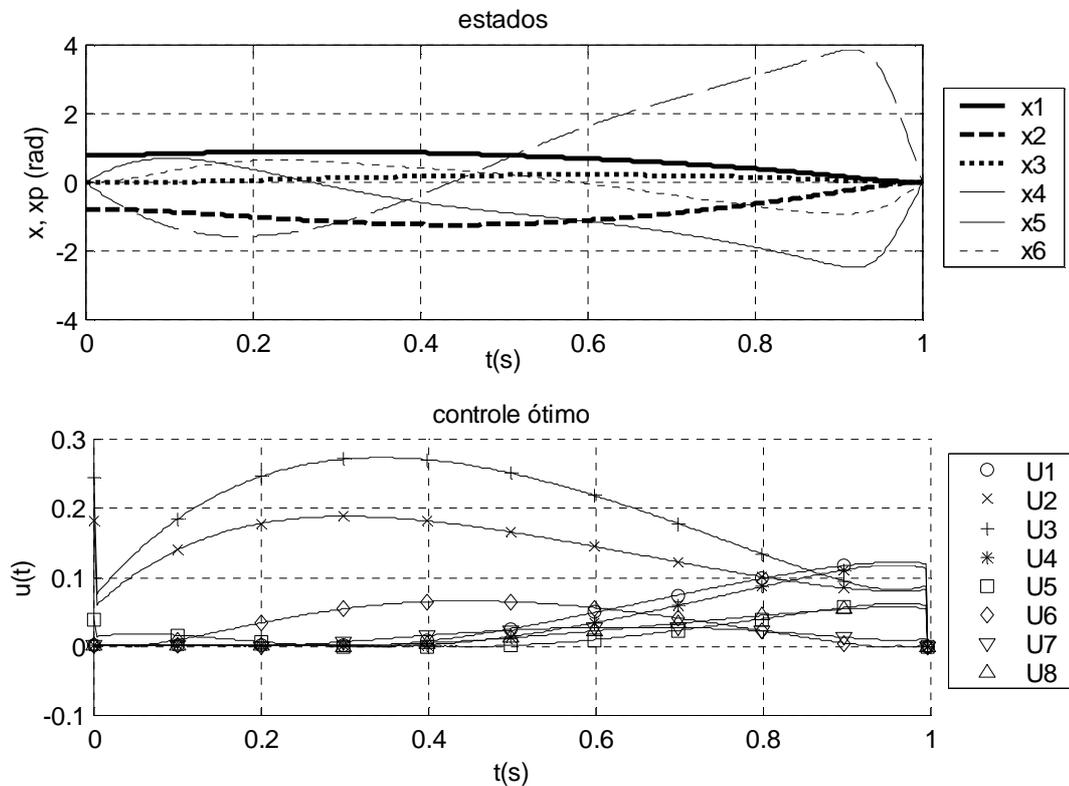


Figura 5.17: Solução final para $t > t_{\text{crit}}$, com vínculos de igualdade, utilizando como estimativa inicial o controle mostrado na figura anterior. Tempo de cpu = apx. 100 horas num Pentium II 400 MHz.

5.4 Pêndulo triplo com atuadores musculares não-lineares e braços de momento fixos

Uma vez estabelecidas algumas diretrizes metodológicas para o controle do modelo biomecânico da postura simplificado em diversos níveis, resta aplicá-las a um modelo que contemple a maior parte das informações anatômicas e fisiológicas relevantes a uma análise efetiva do problema proposto.

Deste modo, foram introduzidos finalmente no modelo atuadores musculares não lineares. Nesta etapa, foram consideradas algumas hipóteses simplificadoras:

- A contração foi admitida isométrica. Os comprimentos dos atuadores foram fixados no valores apresentados na posição anatômica.
- Os braços de momento foram admitidos constantes em função da variação das coordenadas generalizadas e tomados na mesma posição dos comprimentos.
- A relação força x comprimento da fibra muscular foi considerada constante e igual a 1.
- A dinâmica da ativação possuía as mesmas constantes de tempo para todos os músculos.
- Dos 40 músculos analisados e mostrados no Capítulo 2, foram criados 10 grupos musculares (chamados de **gm**⁵⁴), eliminando alguns músculos e agrupando outros, conforme o critério descrito abaixo

Em posição anatômica, os parâmetros mecânicos e geométricos dos 40 músculos descritos no Capítulo 2, a partir do modelo criado por Scott Delp e seus colaboradores, (Delp et al., 1990; Delp e Loan, 1995), estão mostrados na Tabela 5.1.

⁵⁴ Não confundir com os grupos definidos no Capítulo 2 com base nas coordenadas generalizadas do modelo geométrico.

Músculo	r_1 (HF)	r_2 (KA)	r_3 (AA)	F_0^M	$r_1 F_0^M$	$r_2 F_0^M$	$r_3 F_0^M$	L^{MT}	grupos
gmed1	-0.0125			546	-6.825			0.12	1
gmed2	-0.0225			382	-8.595			0.13	1
gmed3	-0.0250			435	-10.875			0.11	1
gmin1	-0.002			180	-0.36			0.08	
gmin2	-0.005			190	-0.95			0.08	
gmin3	-0.010			215	-2.15			0.09	
semimem	-0.055	-0.040		1030	-56.65	-41.20		0.42	2
semiten	-0.065	-0.042		328	-21.32	-13.78		0.47	2
bifemlh	-0.065	-0.055		717	-46.605	-39.435		0.45	2
bifemsh		-0.050		402		-20.10		0.21	3
sar	0.05	-0.020		104	5.20	-2.08		0.56	
addlong	0.04			418	16.72			0.20	
addbrev	0.005			286	1.43			0.13	
amag1	-0.01			346	-3.46			0.12	
amag2	-0.02			312	-6.24			0.20	
amag3	-0.015			444	-6.66			0.33	
tfl	0.04			155	6.20			0.52	
pect	0.02			117	2.34			0.10	
gra	0.015	-0.035		108	1.62	-3.78		0.41	
gmax1	-0.04			382	-15.28			0.20	4
gmax2	-0.05			546	-27.30			0.21	4
gmax3	-0.07			368	-25.76			0.24	4
iliacus	0.036			429	15.44			0.20	5
psoas	0.034			371	12.61			0.26	5
rf	0.047	0.0295		779	36.61	22.98		0.45	6
vasmed		0.0316		1294		40.89		0.23	7
vasint		0.0300		1365		40.95		0.25	7
vaslat		0.0305		1871		57.06		0.26	7
medgas		-0.0150	-0.0395	1113		-16.69	-43.96	0.41	8
latgas		-0.0125	-0.0405	488		-6.10	-19.76	0.41	8
sol			-0.0384	2839			-109.02	0.30	9
tibpost			-0.010	1270			-12.70	0.35	9
flexdig			-0.010	310			-3.10	0.44	
flexhal			-0.014	322			-4.51	0.42	
tibant			0.0426	603			25.69	0.30	10
perbrev			-0.004	348			1.39	0.21	
perlong			-0.008	754			6.03	0.40	10
pertert			0.0274	90			2.47	0.17	
extdig			0.040	341			13.64	0.44	10
exthal			0.042	108			4.54	0.40	10

Tabela 5.1: Parâmetros musculares em posição anatômica. HF, KA e AA são as três coordenadas generalizadas relevantes no plano sagital: flexão do quadril, ângulo do joelho e ângulo do tornozelo. Os parâmetros r_1 , r_2 e r_3 são os braços de momento em relação a cada eixo de rotação para cada músculo. Os grupos na última coluna foram selecionados de acordo com a semelhança funcional dos músculos que os compõe.

Multiplicando cada um dos braços de momento pela força máxima de cada músculo, é possível obter um estimador da relevância mecânica deste músculo em relação às coordenadas generalizadas. Tendo em conta esta informação, foram eliminados do modelo alguns músculos com capacidade muito pequena. Por outro lado, músculos com a mesma função biomecânica e com parâmetros semelhantes foram selecionados e reunidos em 10 grupos musculares gm_i , $i = 1, \dots, 10$ (última coluna da Tabela 5.1). Na Tabela 5.2 estão mostrados os músculos e os parâmetros adotados para os grupos musculares.

Músculo	grupos	L_0^M	α	L^{ST}	F_0^M	$r_1 F_0^M$	$r_2 F_0^M$	$r_3 F_0^M$	L^{MT}
gmed1	1	0.0535	8.0	0.0780	546	-6.825			0.12
gmed2	1	0.0845	0.0	0.0530	382	-8.595			0.13
gmed3	1	0.0643	19.0	0.0530	435	-10.875			0.11
gm 1		0.0681	9.9344	0.0595	1363	-26.295			0.1191
semimem	2	0.0800	15.0	0.3590	1030	-56.65	-41.20		0.42
semiten	2	0.2010	5.0	0.2620	328	-21.32	-13.78		0.47
bifemlh	2	0.1090	0.0	0.3410	717	-46.605	-39.435		0.45
gm2		0.1108	7.5027	0.3363	2075	-124.58	-94.415		0.4398
bifemsh	3	0.1730	23.0	0.1000	402		-20.10		0.21
gm3		0.1730	23.0	0.1000	402		-20.10		0.21
gmax1	4	0.1420	5.0	0.1250	382	-15.28			0.20
gmax2	4	0.1470	0.0	0.1270	546	-27.30			0.21
gmax3	4	0.1440	5.0	0.1450	368	-25.76			0.24
gm4		0.1448	3.0026	0.1333	1296	-68.34			0.2191
iliacus	5	0.1000	7.0	0.0900	429	15.44			0.20
psoas	5	0.1040	8.0	0.1300	371	12.61			0.26
gm5		0.1018	7.4496	0.1080	800	28.05			0.2270
rf	6	0.0840	5.0	0.3460	779	36.61	22.98		0.45
gm6		0.0840	5.0	0.3460	779	36.61	22.98		0.45
vasmed	7	0.0890	5.0	0.1260	1294		40.89		0.23
vasint	7	0.0870	3.0	0.1360	1365		40.95		0.25
vaslat	7	0.0840	5.0	0.1570	1871		57.06		0.26
gm7		0.0857	4.3788	0.1407	4530		139.90		0.2464
medgas	8	0.0450	17.0	0.4080	1113		-16.69	-43.96	0.41
latgas	8	0.0640	8.0	0.3850	488		-6.10	-19.76	0.41
gm8		0.0507	14.3097	0.4011	1601		-22.79	-63.72	0.41
sol	9	0.0300	25.0	0.2860	2839			-109.02	0.30
tibpost	9	0.0310	12.0	0.3100	1270			-12.70	0.35
gm9		0.0301	23.6436	0.2885	4109			-121.72	0.3052
tibant	10	0.0980	5.0	0.2230	603			25.69	0.30
perlong	10	0.0490	10.0	0.3450	754			6.03	0.40
extdig	10	0.1020	8.0	0.3450	341			13.64	0.44
exthal	10	0.1110	6.0	0.3050	108			4.54	0.40
gm10		0.1006	6.0362	0.2649	1052			43.87	0.3539

Tabela 5.2: Grupos musculares selecionados e seus parâmetros.

Os comprimentos ótimos, os comprimentos dos atuadores, os ângulos de empenamento e os comprimentos de tendão relaxado dos grupos foram calculados a partir da média ponderada pelo produto $\mathbf{r} F_0^M$ de cada músculo em cada coordenada generalizada.

Em testes preliminares, usando um modelo de 6 músculos não lineares com parâmetros arbitrários, verificou-se a necessidade de relaxar as tolerâncias de otimização e violação de vínculos do algoritmo de otimização tipo SQP (Sequential Quadratic Programming) (NPSOL) utilizado pelo RIOTS. Utilizando os valores padrão destas tolerâncias, 10^{-6} e 10^{-4} , o OCP não convergia. Assim, esses parâmetros foram aumentados para valores próximos a 0.1, ainda que tolerâncias um pouco mais estreitas permitissem convergência em alguns casos.

5.4.1 Testes com tempo final 0.4 segundos e com tempo acrescido

Com a introdução dos 10 atuadores com parâmetros fisiológicos, percebeu-se que, em função das condições iniciais testadas ($\pi/6$ de flexão para os ângulos do tornozelo e do joelho), dos vínculos terminais, do tempo de simulação e dos vínculos de controle, não era possível que o sistema convergisse numa única solução obedecendo por completo os vínculos. Foi realizada uma seqüência de simulações, cujos parâmetros estão mostrados na Tabela 5.3, visando a determinação de *via de soluções* que permitissem chegar, ao final, numa resposta convergente. Os vínculos terminais foram tratados sempre como vínculos de desigualdade, mesmo na resposta final, quando os valores especificados dos vínculos eram da mesma ordem de grandeza da tolerância para a violação de vínculo.

Uma tentativa de estender o tempo de simulação foi realizada nesse série de simulações: a uma solução convergente obtida até 0.4 segundos eram acrescentados novos pontos à malha, com algum padrão de controle escolhido para estimativa inicial desse

novo trecho. Em geral, valores constantes para todos os controles. A partir de uma primeira solução com esses novos pontos, mantendo-se inicialmente os vínculos terminais em valores largos ($\pi/5$), os vínculos terminais eram progressivamente estreitados, como nas simulações sem pontos acrescidos.

Mantendo inicialmente o tempo final fixo em 0.4 segundos, foi realizada uma 1ª série de simulações utilizando 300 pontos na malha de discretização, splines de 4ª ordem e integrador com passo variável LSODA, conveniente para sistemas duros e altamente não-lineares. Além disso, as estimativas de partida do vetor de controle para a seqüência de soluções foram escolhidas de maneira tal que fossem ativados músculos extensores, preferencialmente uniarticulares. Todas as condições utilizadas estão especificadas na Tabela 5.3. Nesta série, as soluções convergentes obtidas e a seqüência de soluções foram:

12_9_3 \Leftarrow 12_9_1b \Leftarrow u00

12_9_4 (Figura 5.18) \Leftarrow 12_9_2 \Leftarrow 12_9_2a \Leftarrow 12_9_1 \Leftarrow u00

13_9_1 \Leftarrow 12_9_4 \Leftarrow 12_9_2 \Leftarrow 12_9_2a \Leftarrow u00

13_9_2 \Leftarrow 12_9_4 \Leftarrow 12_9_2 \Leftarrow 12_9_2 \Leftarrow 12_9_2a \Leftarrow 12_9_1 \Leftarrow u00

13_9_4 (Figura 5.19) \Leftarrow 12_9_4 \Leftarrow 12_9_2 \Leftarrow 12_9_2 \Leftarrow 12_9_2a \Leftarrow 12_9_1 \Leftarrow u00

Pôde-se observar, nessas respostas, um significativo nível de ruído numérico nas funções de controle, apesar de ser verificável sua redução com a diminuição da tolerância, tanto de otimização quanto de violação de vínculos. A seqüência específica adotada em cada simulação influencia na resposta e no valor obtido da função de custo: pode-se notá-lo comparando as respostas 12_9_3 e 12_9_4 (Figura 5.18). Foi realizado um teste utilizando uma resposta obtida com o LSODA (14_9_4) como estimativa inicial para outra simulação (22_9_1) com RK4, fornecendo uma diminuição da função de custo (de 10.0583 para 7.6575) e aumento do ruído. Em todos os testes, também nas outras seqüências que serão analisadas adiante, foram utilizadas splines de 4ª ordem. Testes com splines de menor ordem (11_9_2) apresentaram erros na busca linear. Do ponto de vista

cinemático, a resposta que apresentou menores valores máximos de velocidades dos membros foi a 13_9_1. As respostas 13_9_2 e 13_9_4 (Figura 5.19) apresentaram aumento das velocidades máximas e a resposta 13_9_4 foi a que forneceu o menor valor da função de custo e o menor nível de ruído. Diminuindo as tolerâncias a partir dessas respostas (14_9_1), a simulação apresentava um erro fatal de memória.

Com o mesmo tempo final mantido em 0.4 segundos, foi realizada uma nova série de simulações, desta vez utilizando integrador RK4 e menores níveis de discretização, isto é, 200 e, na maior parte dos testes, 150 nós. Testes com 100 pontos produziram erros fatais de memória (25_9_2). As seqüências de simulações que levaram a resultados convergentes foram:

200 pontos: 24_9_4 \Leftarrow 24_9_2 \Leftarrow 24_9_1 \Leftarrow u00

150 pontos: 25_9_1 (Figura 5.20) \Leftarrow 24_9_5 (Figura 5.21) \Leftarrow 24_9_3 (Figura 5.22) \Leftarrow u00

Comparando os tempos totais de uso de CPU das duas séries de simulações, com 150 pontos foram gastos 98 minutos, ao passo que com 200 pontos 198 minutos, mostrando que um aumento de 33% no refinamento da discretização levou a tempos de CPU ao redor de 2 vezes mais longos, não obstante os controles obtidos diferirem pouco. O uso de 150 pontos levou a uma função de custo um pouco menor ($f = 9.3786$) do que 200 pontos ($f = 9.4710$), menores velocidades máximas dos membros e ruídos numéricos maiores para alguns músculos e menores para outros. Testes com LSOSDA e 150 pontos (23_10_1) levaram a resultados equivalentes ao RK4, tanto no que diz respeito aos padrões de controle, aos tempos de CPU e aos níveis de ruído numérico.

Uma tentativa de estender o tempo de simulação acima do tempo crítico observado foi feita acrescentando novos pontos a uma solução previamente obtida com tempo inferior ao crítico. O vetor de controle original, acrescido do novo conjunto de pontos, era

utilizado como estimativa inicial para uma nova seqüência de simulações. Foram feitas diversas tentativas, com diferentes integradores, níveis de discretização, valores de tolerâncias, valores iniciais dos pontos do vetor de controle acrescentado e número de pontos acrescentados (25, 50, 90, 70). Porém, a única série que forneceu ao final um resultado convergente foi a seguinte:

25_10_1 (Figura 5.23) \Leftarrow 24_10_3 \Leftarrow 24_10_2 \Leftarrow 24_10_1 \Leftarrow 22_10_4 \Leftarrow 22_10_1 \Leftarrow u00

Nestas simulações, foi empregado o resultado de uma simulação inicial com 150 pontos, LSODA, acrescentando 50 pontos ao vetor de controle (correspondendo a um aumento de 33% no tempo de simulação, ou 0.53 segundos). Foram empregados, no total, 469 minutos de CPU, e a função de custo decresceu significativamente em relação ao caso com tempo final 0.4 segundos, de 9.3786 para 6.8569. É interessante observar que na última passagem da seqüência de simulações, de uma resposta que apresentava níveis de controle de no máximo 1.5, para o resultado final com vínculo de controle $U_{\max}=1$, a função de custo aumentou 14%, assim como o ruído numérico do controle.

simul.	ci ⁱ	v. term. ⁱⁱ	tol. otm.	tol. cont.	U _{inf}	U _{sup}	RK ⁱⁱⁱ	N	Naug ^{iv}	Uoaug ^v	término ^{vi}	n° its.	min.cpu ^{vii}	f	fig. n.	obs.
05_9_1	u00a	$\pi/15$	0.1	0.1	0	10	5	300	0	0	1	42	951	686.84		
05_9_2	u00a	$\pi/30$	0.1	0.1	0	1	5	300	0	0	2	1	73	8.7737		
06_9_1	05_9_1	$\pi/30$	0.1	0,1	0	2	5	300	0	0	3	2	852	287.72		
07_9_1	u00a	$\pi/15$	0.1	0.1	0	3	5	300	0	0	4	-	951	-		
07_9_2	u00a	$\pi/30$	0.1	0.1	0	3	5	300	0	0	4	-	951	-		
11_9_1	u00	$\pi/5$	0.1	0.1	0	10	5	300	0	0	1	112	700	3.5915		
11_9_2	u00	$\pi/5$	0.1	0.1	0	10	5 ^{viii}	300	0	0	3	-	277	13.5313		
12_9_1b	u00	$\pi/30$	0.1	0.1	0	10	5	300	0	0	1	17	192	9.9804		
12_9_2a	11_9_1	$\pi/5$	0.1	0.1	0	1	5	300	0	0	1	1	43	-		
12_9_2	12_9_2a	$\pi/15$	0.1	0.1	0	1	5	300	0	0	1	18	233	8.9430		
12_9_3	12_9_1b	$\pi/30$	0.1	0.1	0	1	5	300	0	0	1	14	201	9.6492		
12_9_4	12_9_2	$\pi/30$	0.1	0.1	0	1	5	300	0	0	1	8	103	9.2596	Figura 5.18	
13_9_1	12_9_4	$\pi/30$	0.01	0.1	0	1	5	300	0	0	1	150	1364	10.4078		
13_9_2	12_9_4	$\pi/30$	0.1	0.01	0	1	5	300	0	0	1	22	194	10.0812		
13_9_4	12_9_4	$\pi/30$	0.01	0.01	0	1	5	300	0	0	1	46	330	10.0583	Figura 5.19	
14_9_1	12_9_4	$\pi/30$	0.001	0.001	0	1	5	300	0	0	5	0	0	-		
14_9_1	12_9_4	$\pi/30$	0.001	0.01	0	1	5	300	0	0	5	0	0	-		
14_9_1	12_9_4	$\pi/30$	0.01	0.001	0	1	5	300	0	0	5	0	0	-		
14_9_1	12_9_4	$\pi/30$	0.005	0.01	0	1	5	300	0	0	5	0	0	-		
14_9_1	12_9_4	$\pi/30$	0.01	0.005	0	1	5	300	0	0	5	0	0	-		
22_9_1	12_9_4	$\pi/30$	0.1	0.1	0	10	4	300	0	0	1	1	35	7.6575		
22_9_2	22_9_1	$\pi/30$	0.1	0.1	0	10	4	300	10	0.2	1	4	47	7.8544		
22_9_3	22_9_1	$\pi/30$	0.1	0.1	0	10	4	300	30	0.2	1	62	472	7.0650		
23_9_1	22_9_1	$\pi/30$	0.1	0.1	0	10	4	300	130	0.2	1	13	460	5.9625		lim.mem
24_9_1	u00	$\pi/5$	0.1	0.1	0	10	4	200	0	0	1	24	96	3.6373		
24_9_2	24_9_1	$\pi/15$	0.1	0.1	0	1	4	200	0	0	1	22	74	8.8661		
24_9_3	24_9_1	$\pi/5$	0.1	0.1	0	10	4	150	0	0	1	34	43	3.4314	Figura 5.22	
24_9_4	24_9_2	$\pi/30$	0.1	0.1	0	1	4	200	0	0	1	6	28	9.4710		
24_9_5	24_9_3	$\pi/15$	0.1	0.1	0	1	4	150	0	0	1	30	47	9.2629	Figura 5.21	
25_9_1	24_9_5	$\pi/30$	0.1	0.1	0	1	4	150	0	0	1	2	8	9.3786	Figura 5.20	
25_9_2	u00	$\pi/5$	0.1	0.1	0	10	4	100	0	0	5	-	-	-		
25_9_3	25_9_1	$\pi/15$	0.1	0.1	0	10	4	150	90	0.2	3	72	343	-		

25_9_4	25_9_1	$\pi/30$	0.1	0.1	0	10	4	150	90	0.2	3	48	165	13.7261		
25_9_5	25_9_1	$\pi/10$	0.1	0.1	0	10	4	150	90	0.2	3	82	293	12.6994		
25_9_6	25_9_1	$\pi/15$	0.2	0.2	0	10	4	150	90	0.2	2	21	103	10.7151		
26_9_1	25_9_1	$\pi/10$	0.1	0.1	0	10	4	150	90	0.2	6	200	211	13.6241		
26_9_2	25_9_1	$\pi/7$	0.1	0.1	0	10	4	150	90	0.2	2	234	190	11.9227		
26_9_3	25_9_1	$\pi/5$	0.1	0.1	0	10	4	150	90	0.2	1	535	313	5.0363		
26_9_4	26_9_3	$\pi/15$	0.1	0.1	0	10	4	150	90	0.2	6	1000	583	6.6143		
27_9_1	25_9_1	$\pi/5$	0.1	0.1	0	10	4	150	50	0.2	1	894	244	13.0026		
27_9_2	25_9_1	$\pi/15$	0.1	0.1	0	10	4	150	50	0.2	3	281	153	22.2637		
27_9_3	25_9_1	$\pi/5$	0.1	0.1	0	10	4	150	50	0.15	7	0	-	-		
27_9_4	25_9_1	$\pi/5$	0.1	0.1	0	10	4	150	25	0.2	1	170	54	8.7449		
28_9_1	27_9_4	$\pi/5$	0.1	0.01	0	10	4	150	25	0.2	1	719	702	4.6236		
28_9_2	28_9_1	$\pi/15$	0.1	0.1	0	10	4	150	25	0.2	3	1	4	5.1172		
29_9_1	27_9_4	$\pi/15$	0.1	0.1	0	10	4	150	25	0.2	3	102	158	9.7429		
29_9_2	25_9_1	$\pi/4$	0.1	0.1	0	50	4	150	75	0.2	3	32	103	18.4097		
29_9_3	26_9_3	$\pi/15$	0.1	0.1	0	10	4	150	50	0.2	1	1025	640	6.6100		
30_9_1	26_9_3	$\pi/15$	0.1	0.01	0	10	4	150	50	0.2	1	770	693	6.9916		
30_9_2	29_9_3	$\pi/30$	0.1	0.1	0	1	4	150	50	0.2	7	0	0	-		
1_10_1	30_9_1	$\pi/30$	0.1	0.01	0	10	4	150	50	0.2	7	0	0	-		
1_10_2	26_9_3	$\pi/30$	0.1	0.01	0	10	4	150	50	0.2	3	979	1358	9.5787		
3_10_1	30_9_1	$\pi/30$	0.1	0.1	0	10	4	150	50	0.2	3	1	6	8.8738		obs.1
3_10_2	30_9_1	$\pi/30$	0.2	0.2	0	10	4	150	50	0.2	3	1	7	8.8070		
3_10_3	26_9_3	$\pi/15$	0.1	0.01	0.2	10	4	150	50	0.2	2	73	202	8.9989		caiu
4_10_1	30_9_1	$\pi/30$	0.1	0.1	0	10	4	150	50	0.2	3	768	876	11.8477		obs.2
10_10_1	26_9_3	$\pi/5$	0.1	0.01	0	10	4	150	50	0.2	3	776	555	3.7941		
11_10_1	26_9_3	$\pi/5$	0.1	0.05	0	10	4	150	50	0.2	3	17	48	5.2147		
11_10_2	26_9_3	$\pi/5$	0.3	0.05	0	10	4	150	50	0.2	3	17	48	5.2147		
11_10_3	26_9_3	$\pi/5$	0.1	0.15	0	10	4	150	50	0.2	3	23	55	5.2036		
11_10_4	26_9_4	$\pi/4.5$	0.1	0.02	0	10	4	150	50	0.2	1	499	282	5.0286		
13_10_1	11_10_4	$\pi/15$	0.1	0.02	0	10	4	150	50	0.2	3	721	775	13.2390		
16_10_1	11_10_4	$\pi/15$	0.1	0.1	0	10	4	150	50	0.2	1	474	587	9.9374		
16_10_2	11_10_4	$\pi/15$	0.1	0.07	0	10	4	150	50	0.2	1	639	476	21.4488		
17_10_1	11_10_4	$\pi/15$	0.1	0.05	0	10	4	150	50	0.2	3	212	300	11.8018		

17_10_2	16_10_2	$\pi/30$	0.1	0.07	0	10	4	150	50	0.2	3	97	285	24.4738		
18_10_1	16_10_2	$\pi/20$	0.1	0.07	0	10	4	150	50	0.2	3	173	300	11.8018		
18_10_2	16_10_2	$\pi/18$	0.1	0.1	0	10	4	150	50	0.2	3	172	128	22.0898		
19_10_1	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	70	0.2	7	-	-	-		
19_10_1	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	0.25	3	1	17	13.9760		
19_10_1	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	0.2	3	-	-	-		
19_10_1	24_9_4	$\pi/5$	0.2	0.1	0	10	4	200	50	0.25	3	1	17	13.9760		
19_10_1	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	0.28	2	45	7	12.6274		
19_10_1	24_9_4	$\pi/5$	0.1	0.2	0	10	4	200	50	0.25	3	1	17	13.9760		
19_10_1	24_9_4	$\pi/5$	0.1	0.2	0	10	4	200	50	0.25	3	1	18	13.9670		bbnd10 ²
19_10_2	24_9_1	$\pi/5$	0.1	0.2	0	10	4	200	50	0.25	3	1	13	8.3692		
19_10_3	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.3	3	1	11	12.7808		
19_10_4	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.4	1	281	262	6.4595		
20_10_1	19_10_4	$\pi/15$	0.1	0.1	0	10	4	200	50		7	-	-	-		
20_10_1	24_9_4	$\pi/10$	0.1	0.1	0	10	4	200	50	obs.5	2	10	65	12.7808		
20_10_2	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.6	2	5	61	11.8413		
20_10_3	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.7	2	4	43	12.7080		
20_10_3	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.8	3	1	16	17.3583		
20_10_4	24_9_4	$\pi/5$	0.1	0.1	0	10	4	200	50	obs.9	3	1	15	13.0828		
21_10_1	u00	$\pi/5$	0.1	0.1	0	10	4	300	0		1	104	498	3.6027		
21_10_2	21_10_1	$\pi/15$	0.1	0.1	0	10	4	300	0		7	-	-	-		
22_10_1	u00	$\pi/5$	0.1	0.1	0	10	5	150	0		1	32	43	3.4314		
22_10_3	25_9_1	$\pi/30$	0.1	0.1	0	1	5	150	0		1	3	10	9.6158		
22_10_4	22_10_1	$\pi/15$	0.1	0.1	0	10	5	150	0		1	30	46	9.2629		
22_10_5	22_10_3	$\pi/30$	0.1	0.1	0	1	4	150	0		7	-	-	-		
23_10_1	22_10_4	$\pi/30$	0.1	0.1	0	1	5	150	0		1	2	7	9.3786		
23_10_2	23_10_1	$\pi/30$	0.1	0.1	0	1	4	150	0		7	-	-	-		
24_10_1	22_10_4	$\pi/5$	0.1	0.1	0	10	5	150	50	0.2	1	52	158	3.5541		
24_10_2	24_10_1	$\pi/15$	0.1	0.1	0	10	5	150	50	0.2	1	18	50	5.5443		
24_10_3	24_10_2	$\pi/30$	0.1	0.1	0	10	5	150	50	0.2	1	5	28	6.0173		
25_10_1	24_10_3	$\pi/30$	0.1	0.1	0	1	5	150	50	0.2	1	33	144	6.8569	Figura 5.23	
25_10_2	25_10_1	$\pi/5$	0.1	0.1	0	10	5	200	80	0.2	3	53	470	5.6380		
26_10_1	22_10_1	$\pi/5$	0.1	0.1	0	10	5	150	50	0.2	3	4	24	5.3989		

26_10_2	22_10_4	$\pi/5$	0.1	0.1	0	10	5	150	50	0.2	3	2	29	10.2097		
26_10_3	25_10_1	$\pi/5$	0.1	0.1	0	10	5	200	50	0.2	3	3	39	4.5740		
26_10_4	25_10_1	$\pi/5$	0.1	0.1	0	10	5	200	25	0.2	3	3	43	4.4479		
26_10_5	25_10_1	$\pi/15$	0.1	0.1	0	10	5	200	25	0.2	3	4	54	5.4846		
26_10_6	25_10_1	$\pi/15$	0.1	0.2	0	10	5	200	25	0.2	3	2	22	6.2164		
27_10_1	23_10_1	$\pi/5$	0.1	0.1	0	10	5	150	100	0.1	3	3	55	10.6388		
27_10_2	23_10_1	$\pi/5$	0.1	0.1	0	10	5	150	80	0.15	3	7	106	19.2782		
27_10_3	23_10_1	$\pi/5$	0.1	0.1	0	10	5	150	50	0.05	3	2	26	10.6035		
27_10_4	25_10_1	$\pi/15$	0.1	0.1	0	10	5	200	25	0.2	3	4	54	5.4846		obs.10
29_10_1	25_10_1	$\pi/15$	0.1	0.1	0	10	5	200	25	0.2	3	4	53	5.4846		obs.11

Tabela 5.3: Resultados sem tempo aumentado e com tempo aumentado, r fixo

```

i u00a=ones(size(u00));
u00a=0.5*[0.8*u00(1,:);
0.05*u00(2,:);
0.05*u00(3,:);
0.05*u00(4,:);
0.5*u00(5,:);
1*u00(6,:);
1*u00(7,:);
1*u00(8,:);
1*u00(9,:);
0.05*u00(10,:)];
u00=ones(size(u00));
u00=0.05*[2*u00(1,:);
2*u00(2,:);
2*u00(3,:);
2*u00(4,:);
20*u00(5,:);
10*u00(6,:);
2*u00(7,:);
2*u00(8,:);
10*u00(9,:);
2*u00(10,:)];
ii Vínculo de desigualdade  $x_f^2 \leq v.term^2$ .
iii RK5=LOSODA
iv número de pontos a mais
v valor adotado em todos os controles na ci do tempo aumentado
vi
1:término normal
2:não foi encontrado ponto factível para os vínculos não lineares
3:falha na busca linear

```

4:não é possível calcular o gradiente dos vínculos
 5:erro de memória
 6:completado número máximo de iterações
 7:erro de ponto flutuante durante cálculo do gradiente do vínculo

^{vii} Pentium III 600Mhz.

^{viii} testes com splines de 1a e 2a ordem

obs.1: estava apresentando erro de ponto flutuante (provavelmente div. por zero) devido a cis próximas de zero. Tentativas de alterar controle inicial eliminando valores muito pequenos.
 obs.2: if u000(i,j)^2 <= 0.0001, u000(i,j)=0.2;
 obs.3: u0aug=0.10*ones(10,Naug).*rand(10,Naug)*4
 obs.4: u0aug=0.10*ones(10,Naug).*rand(10,Naug)*3
 obs.5: Quando o u0aug é muito baixo, aparecem problemas de div. por zero (floating point error). Se for muito alto, o sistema cai. Utilizando um controle variável baixos níveis de controle não geram erro de div. por zero
 for i=1:10,
 u0aug(i,:)=0.2+0.15*(sin(2*pi*40*taug(length(t):length(taug)-1)));
 end
 obs.6: u0aug(i,:)=0.2+0.15*(sin(-2*pi*20*taug(length(t):length(taug)-1)))
 obs.7: u0aug(i,:)=0.2+0.15*(sin(-2*pi*10*taug(length(t):length(taug)-1)))
 obs.8: u0aug(i,:)=0.2+0.2*(sin(-2*pi*10*taug(length(t):length(taug)-1)));
 obs.9: u0aug(i,:)=0.2+0.2*(sin(-2*pi*60*taug(length(t):length(taug)-1)));
 obs.10: mudança no arquivo npsol.opt: Line search tolerance = 1; Minor iteration limit = 5000
 obs.10: mudança no arquivo npsol.opt: Line search tolerance = 5;

5.4.2 Testes com tempo final 1 segundo e aceleração da gravidade variável

Uma tentativa de obter simulações com tempos superiores ao tempo crítico de 0.4 segundos foi utilizar valores pequenos da aceleração da gravidade para gerar as primeiras estimativas do vetor de controle. Considerando inicialmente a aceleração da gravidade como 2 m/s^2 e utilizando uma estimativa inicial u_{00} conforme descrito na Tabela 5.4, foi possível controlar o modelo para vínculos terminais $\pi/5$. A partir dessa resposta, a aceleração da gravidade foi aumentada para 5, 7.5 e, finalmente, 9.81 m/s^2 , utilizando a resposta anterior como estimativa inicial, segundo a seguinte seqüência de simulações:

25_11_3 (Figura 5.24, Figura 5.25 e Figura 5.26) \Leftarrow 14_11_1 \Leftarrow 10_11_1 \Leftarrow 9_11_1 \Leftarrow 8_11_1 \Leftarrow 7_11_1 (Figura 5.27) \Leftarrow u_{00}

Foram utilizados 400 pontos na malha de discretização e integrador RK4, e entre as simulações 8_11_1, 9_11_1, 10_11_1 e 14_11_1 foram feitos pequenos ajustes no vetor de controle da simulação anterior, tendo em vista a obtenção de respostas um pouco mais próximas da região de convergência. Essencialmente, o aumento da gravidade requer maiores níveis de ativação dos extensores, o que pode ser feito por tentativa e erro até que os valores máximos das variáveis de estado se afastem o mínimo possível dos vínculos terminais. Os ajustes adotados estão mostrados na Tabela 5.4. Para a simulação 25_11_3, as ativações neuromusculares e as forças musculares estão mostradas na Figura 5.25 e na Figura 5.26.

É notável a diminuição da função de custo em relação à condição com $t_f = 0.4$ segundos, atingindo agora 4.0576. Ao mesmo tempo, as velocidades máximas dos membros se reduziram, denotando a obtenção de um movimento mais suave. Entretanto, o tempo de CPU foi bastante maior, atingindo no total 58.5 horas. Percebeu-se, nessa série de simulações, a conveniência de se adotar um limite inferior na variável de controle

maior que zero, e uma tolerância de violação de vínculos inferior a esse limite mas não nulo, evitando assim erros de divisão por zero, apontado pelo NPSOL como erro de ponto flutuante no cálculo do gradiente do vínculo. Além da resposta obtida na simulação 25_11_3, com tolerância violação de vínculo 0.12, foi obtida também a resposta 25_11_2, com 0.15 para essa tolerância. Percebe-se, comparando as duas respostas, a suavização de alguns picos atribuíveis ao ruído numérico, além de um ligeiro aumento na função de custo e no tempo de simulação, este da ordem de 15%.

simul.	ci ⁱ	v. term. ⁱⁱ	tol. otm.	tol. cont.	U _{inf}	U _{sup}	RK ⁱⁱⁱ	N	tf	g	término ^{iv}	n° its.	min.cpu ^v	f	fig. n.	obs
31_10_1	u00	$\pi/5$	0.1	0.1	0	10	5	300	0.8	3	4	1				
31_10_1	u00	$\pi/5$	0.1	0.01	0	10	5	300	0.8	3	1	48	470	0.1607		
31_10_2	31_10_1	$\pi/15$	0.1	0.01	0	1	5	300	0.8	3	5 LSODA	-	-	-		
1_11_1	u00	$\pi/5$	0.01	0.005	0	10	4	400	1.0	2	1	140	1473	0.0551		
2_11_1	1_11_1	$\pi/10$	0.01	0.005	0	10	4 e 5	400	1.0	2	5 LSODA	-	-	-		
2_11_1	1_11_1	$\pi/10$	0.001	0.00001	0	10	4	400	1.0	2	4	-	-	-		
2_11_1	1_11_1	$\pi/10$	0.01	0.00001	0.05	10	5	400	1.0	2	8	-	4.9 dias	-		
7_11_1	u00	$\pi/5$	0.05	0.02	0.04	10	4	400	1.0	2	1	131	1359	0.2201		
8_11_1	7_11_1	$\pi/15$	0.05	0.02	0.04	10	4	400	1.0	2	1	18	300	0.2855		
8_11_2	8_11_1	$\pi/30$	0.05	0.02	0.04	10	4	400	1.0	2	1	19	315	0.3344		
8_11_3	8_11_2	$\pi/60$	0.05	0.02	0.04	10	4	400	1.0	2	1	29	589	0.3579		
9_11_1	8_11_1	$\pi/5$	0.05	0.02	0.04	10	4	400	1.0	5	1	21	173	0.7261		obs.1
9_11_2	9_11_1	$\pi/15$	0.05	0.02	0.04	10	4	400	1.0	5	1	32	586	0.9998		
10_11_1	9_11_1	$\pi/5$	0.05	0.02	0.04	10	4	400	1.0	7.5	1	21	323	1.5527		obs.2
11_11_1	10_11_1	$\pi/5$	0.05	0.02	0.04	10	4	400	1.0	9.81	1	52	514	2.7376		obs.3
12_11_1	11_11_1	$\pi/15$	0.05	0.02	0.04	10	4	400	1.0	9.81	3	111	1453	3.5777		
13_11_1	10_11_1	$\pi/15$	0.1	0.1	0.04	10	4	400	1.0	9.81	2	6	522	4.4354		caiu
14_11_1	10_11_1	$\pi/15$	0.1	0.035	0.04	10	4	400	1.0	9.81	1	40	1040	3.8670		
15_11_1	14_11_1	$\pi/30$	0.1	0.035	0.04	10	4	400	1.0	9.81	3	32	2582	3359		caiu
20_11_1	14_11_1	$\pi/30$	0.1	0.1	0.04	10	4	400	1.0	9.81	2	2	70	3.8800		caiu
21_11_1	14_11_1	$\pi/20$	0.09	0.05	0.06	10	4	400	1.0	9.81	?	-	-	-		
22_11_1	14_11_1	$\pi/30$	0.15	0.035	0.04	10	4	400	1.0	9.81	6	2	295	4.6784		caiu
23_11_1	14_11_1	$\pi/30$	0.01	0.035	0.04	10	4	400	1.0	9.81	8	-	-	-		caiu
23_11_2	14_11_1	$\pi/30$	0.1	0.1	0	10	4	400	1.0	9.81	6	2	158	3.8511		não caiu
23_11_3	14_11_1	$\pi/30$	0.1	0.1	0	10	4	400	1.0	9.81	3	73	712	3.8016		não caiu
24_11_1	14_11_1	$\pi/30$	0.01	0.1	0	10	4	400	1.0	9.81	3	73	735	3.8016		
25_11_1	14_11_1	$\pi/30$	0.1	0.05	0	10	4	400	1.0	9.81	3	55	388	3.8020		
25_11_2	14_11_1	$\pi/30$	0.1	0.15	0	10	4	400	1.0	9.81	1	16	271	4.0267		
25_11_3	14_11_1	$\pi/30$	0.1	0.12	0	10	4	400	1.0	9.81	1	37	313	4.0576		
27_11_1	14_11_1	$\pi/30$	0.05	0.12	0	1	4	400	1.0	9.81	2	236	2546	3.6224		

Tabela 5.4: Resultados com tempo final 0.4 segundos e aceleração da gravidade variável

Obs.1: correção na estimativa inicial do controle:

```
load resp8_11_1;
u000(1,:)=3*uol(1,:);
u000(2,:)=uol(2,:);
u000(3,:)=uol(3,:);
u000(4,:)=3*uol(4,:);
u000(5,:)=uol(5,:);
u000(6,:)=uol(6,:);
u000(7,:)=1.85*uol(7,:);
u000(8,:)=uol(8,:);
u000(9,:)=1.5*uol(9,:);
u000(10,:)=uol(10,:);
```

obs.2

```
u000(1,:)=2.5*uol(1,:);
u000(2,:)=uol(2,:);
u000(3,:)=uol(3,:);
u000(4,:)=2*uol(4,:);
u000(5,:)=uol(5,:);
u000(6,:)=uol(6,:);
u000(7,:)=1.4*uol(7,:);
u000(8,:)=uol(8,:);
u000(9,:)=1.3*uol(9,:);
u000(10,:)=uol(10,:);
```

obs.3

```
u000(1,:)=2*uol(1,:);
u000(2,:)=uol(2,:);
u000(3,:)=uol(3,:);
u000(4,:)=1.5*uol(4,:);
u000(5,:)=uol(5,:);
u000(6,:)=uol(6,:);
u000(7,:)=1.2*uol(7,:);
u000(8,:)=uol(8,:);
u000(9,:)=1.35*uol(9,:);
u000(10,:)=uol(10,:);
```

```
i u00=ones(size(u00));
u00=0.02*[2*u00(1,:)
          2*u00(2,:)
          2*u00(3,:)
          2*u00(4,:)
          5*u00(5,:)
          4*u00(6,:)
          2*u00(7,:)
          2*u00(8,:)
          4*u00(9,:)
          2*u00(10,:)];
```

ⁱⁱ Vínculo de desigualdade $x_i^2 \leq v \cdot \text{term}^2$.

ⁱⁱⁱ RK5=LOSODA

^{iv} 1:término normal

lineares 2:não foi encontrado ponto factível para os vínculos não

3:falha na busca linear

4:não é possível calcular o gradiente dos vínculos

5:erro de memória

6:completado número máximo de iterações

vínculo 7:erro de ponto flutuante durante cálculo do gradiente d

8:processo interrompido

^v Pentium III 600Mhz.

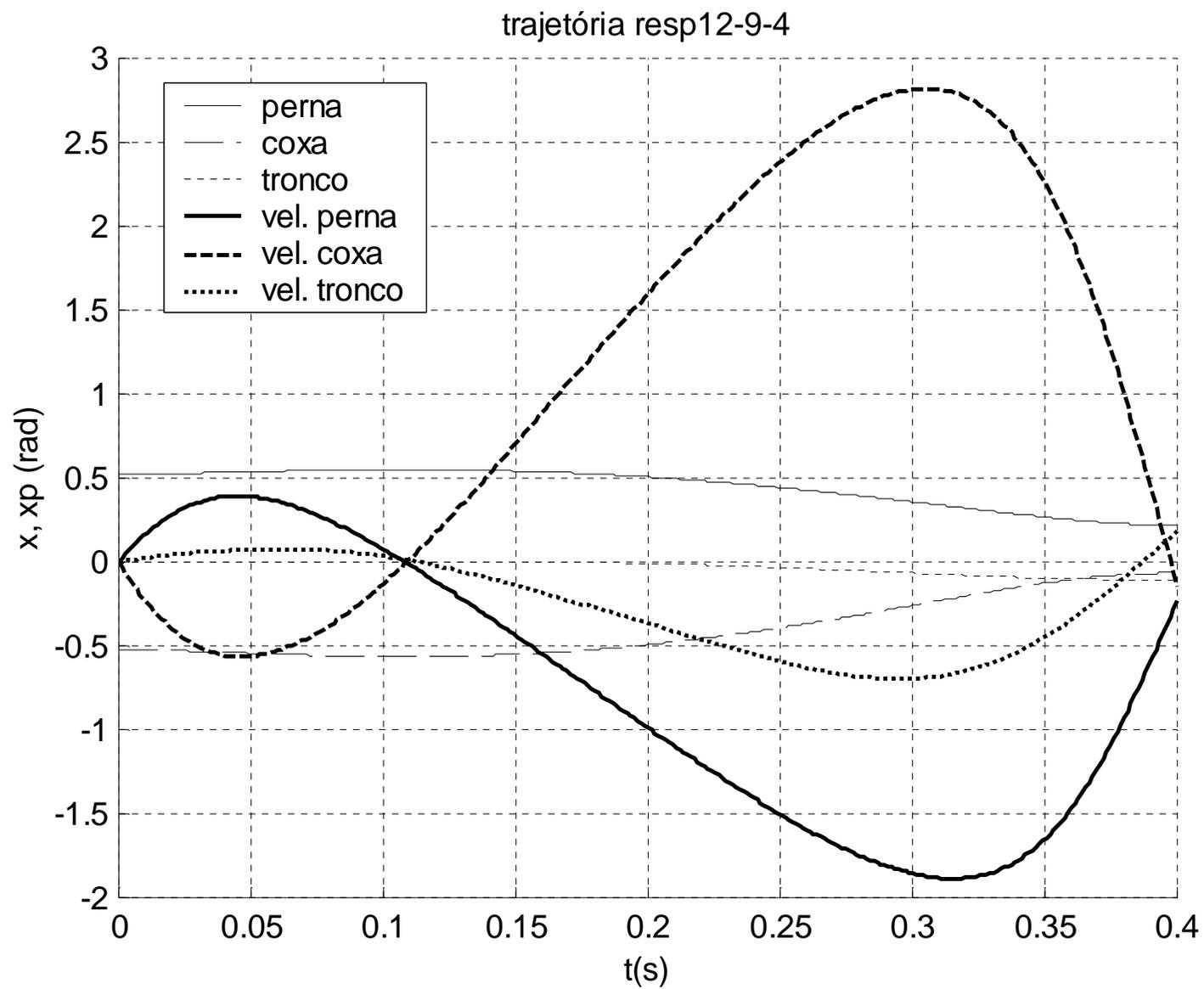


Figura 5.18: Trajetória 12_9_4

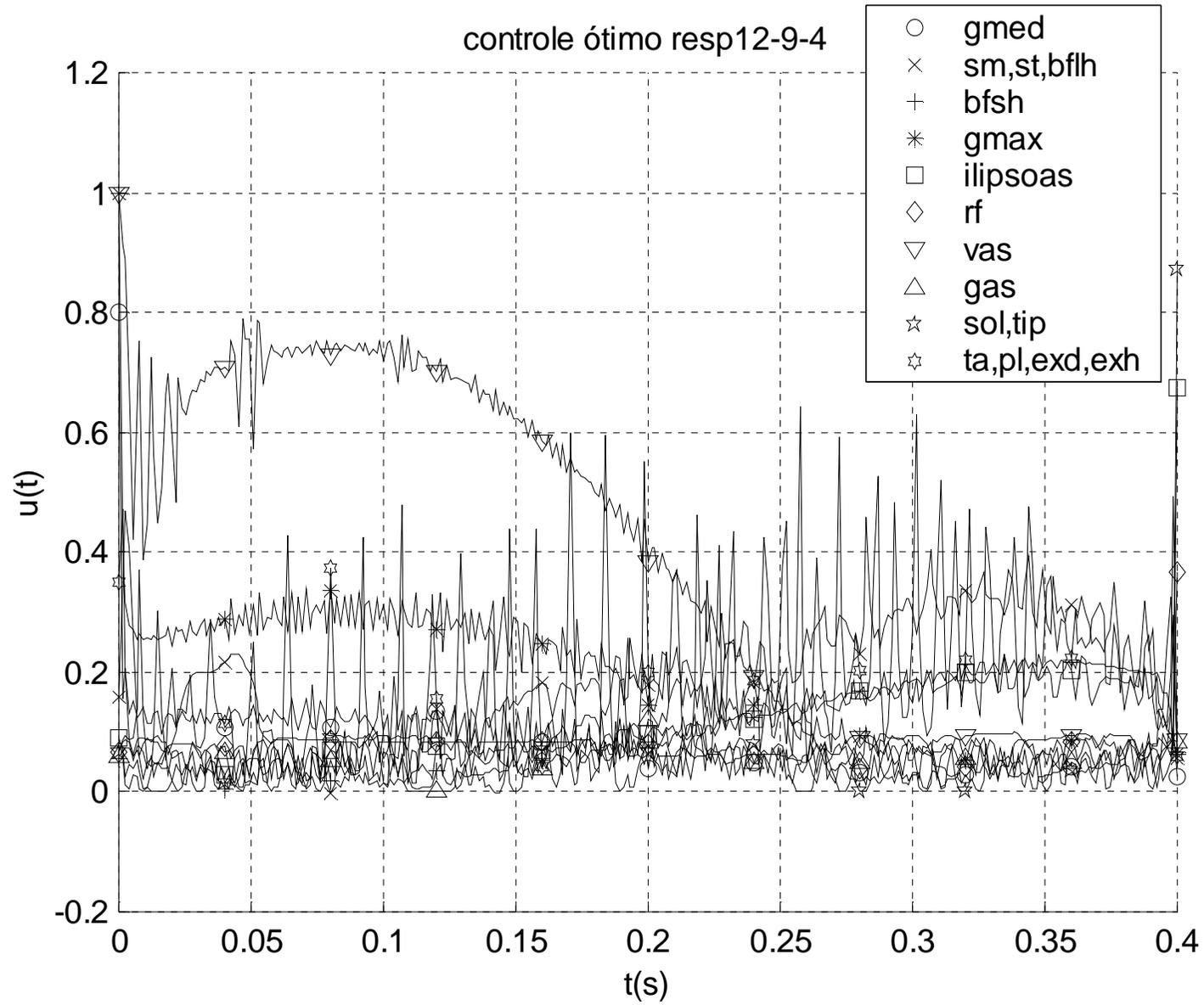


Figura 5.18: Controle 12_9_4

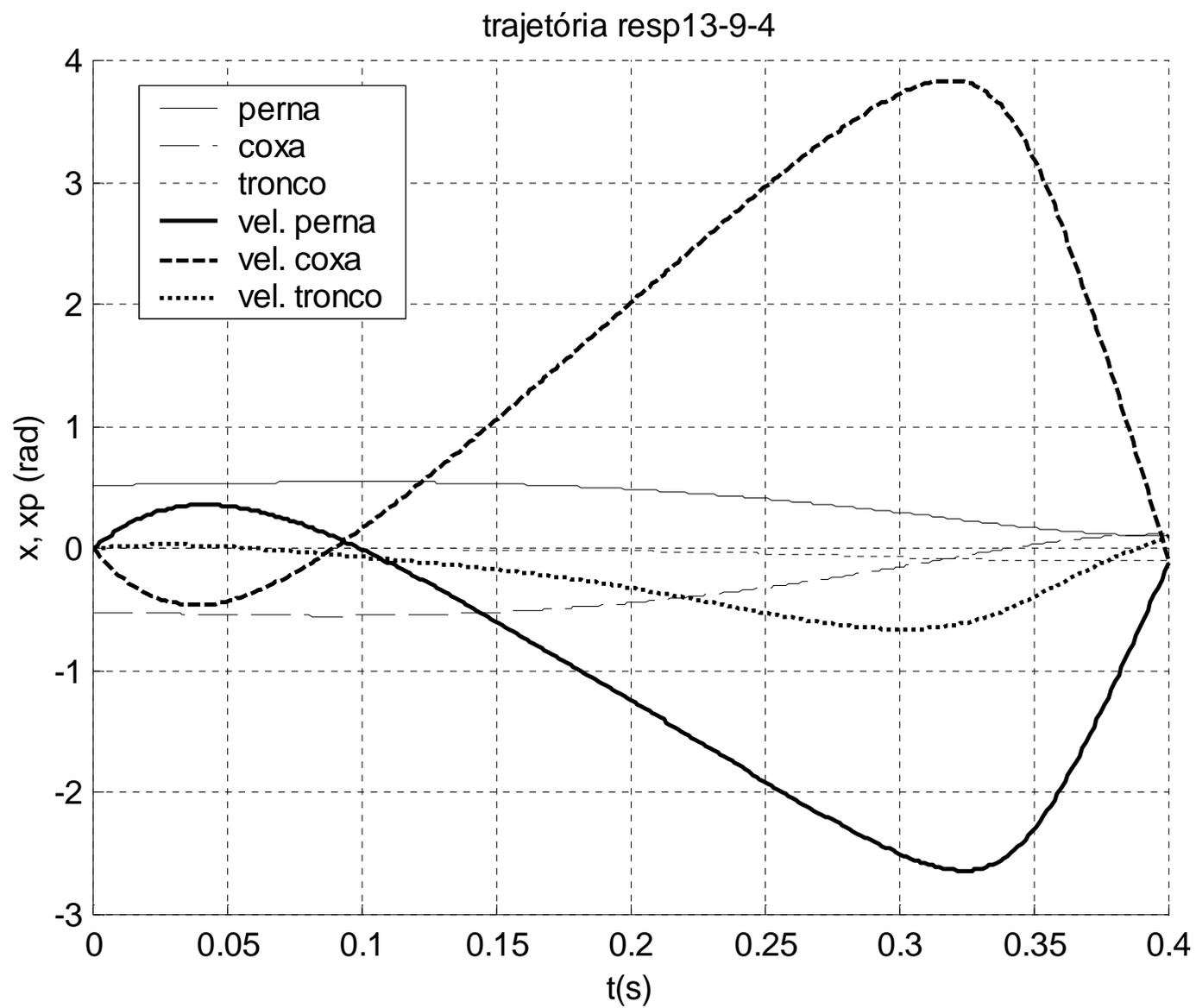


Figura 5.19: Trajetória 13_9_4

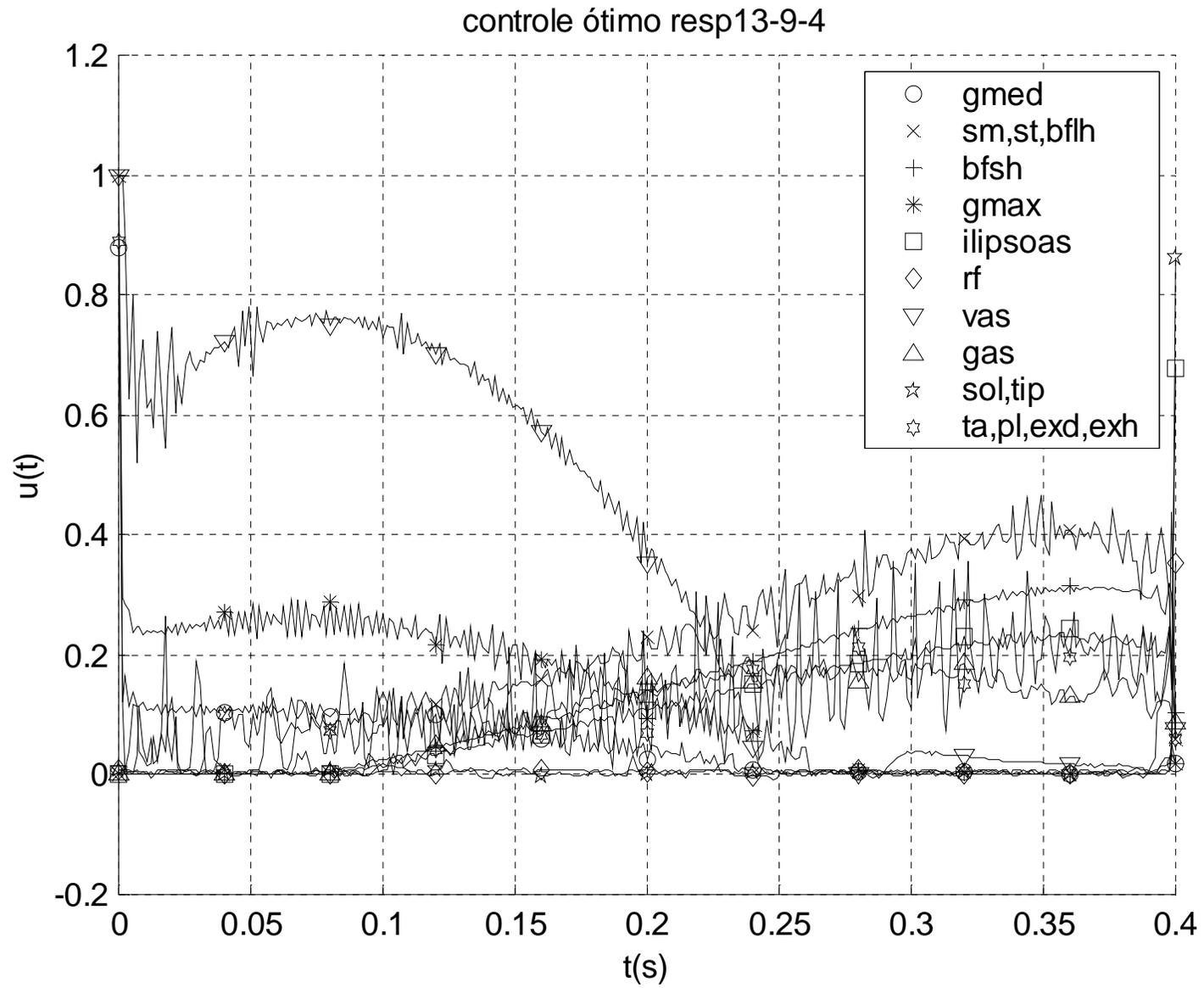


Figura 5.19: Controle 13_9_4

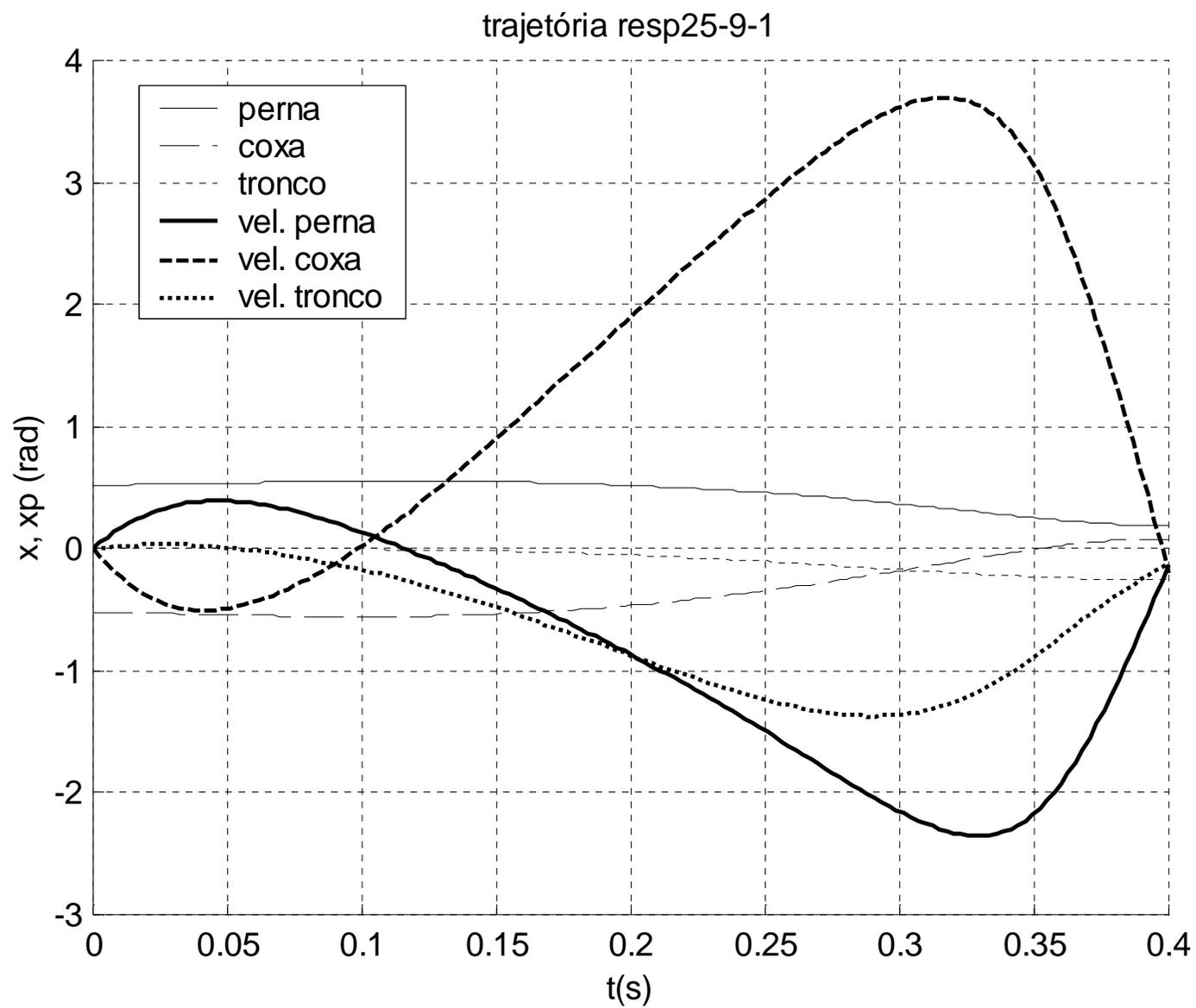


Figura 5.20: Trajetória 25_9_1

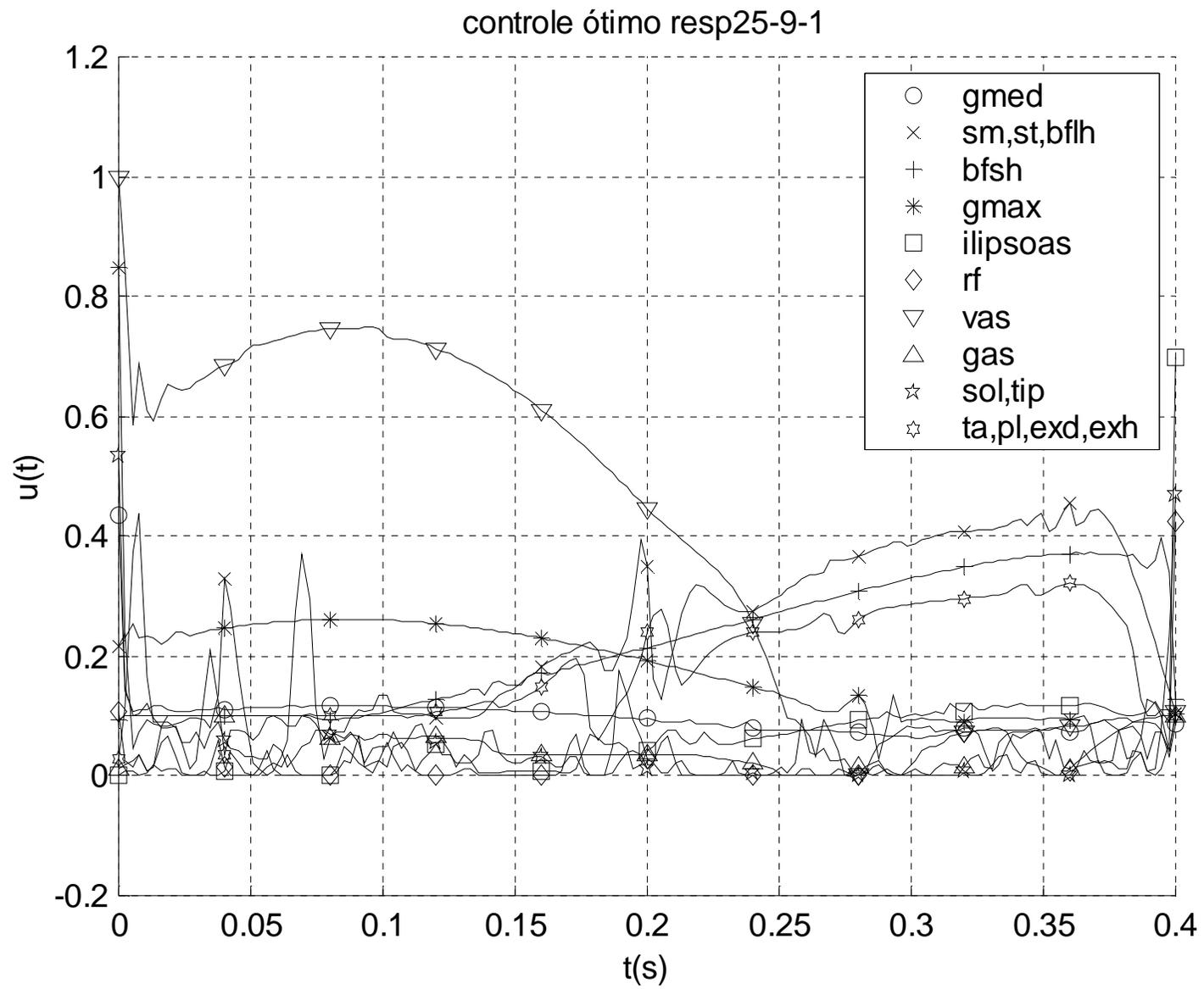


Figura 5.20: Controle 25_9_1

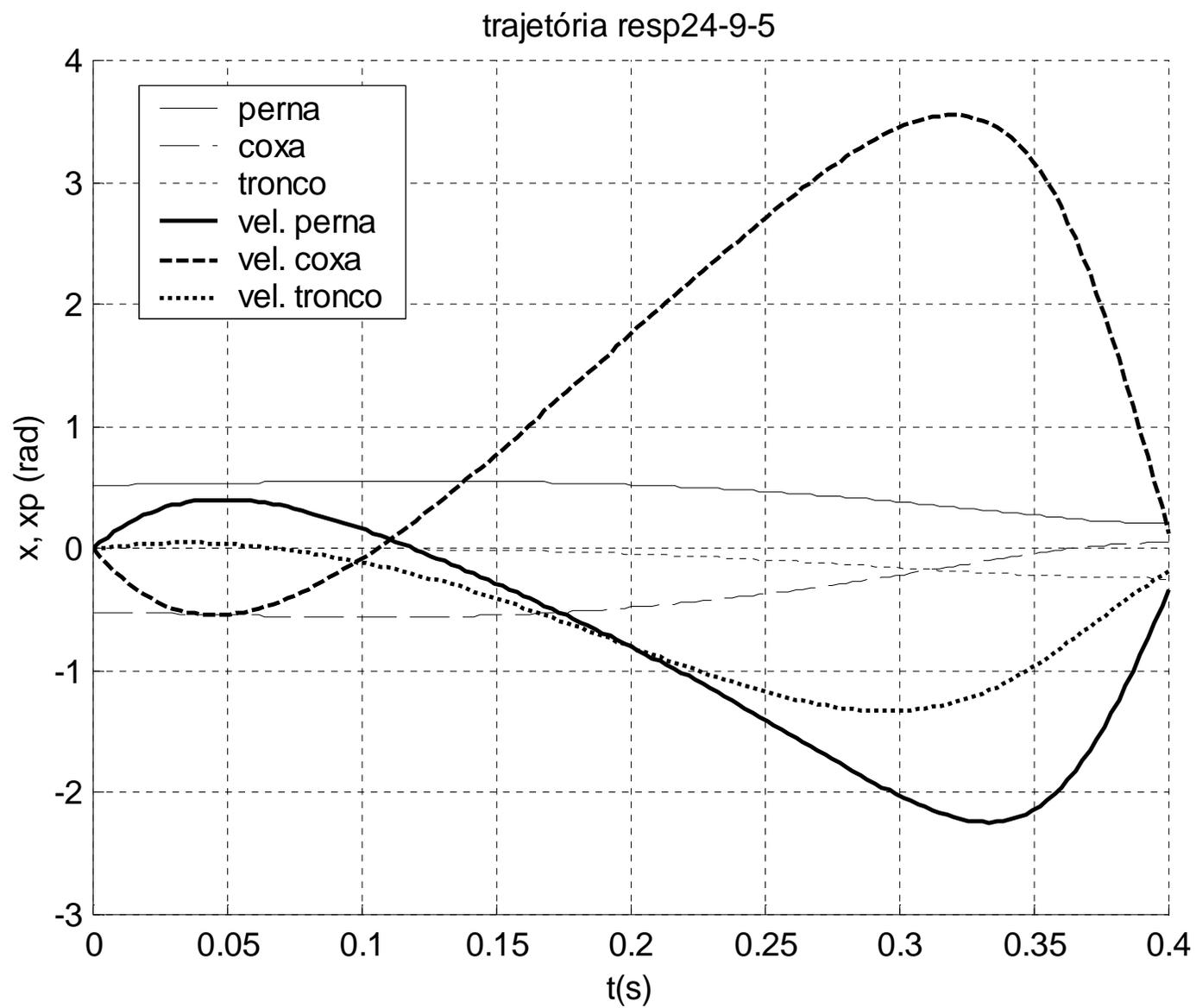


Figura 5.21: Trajetória 24_9_5

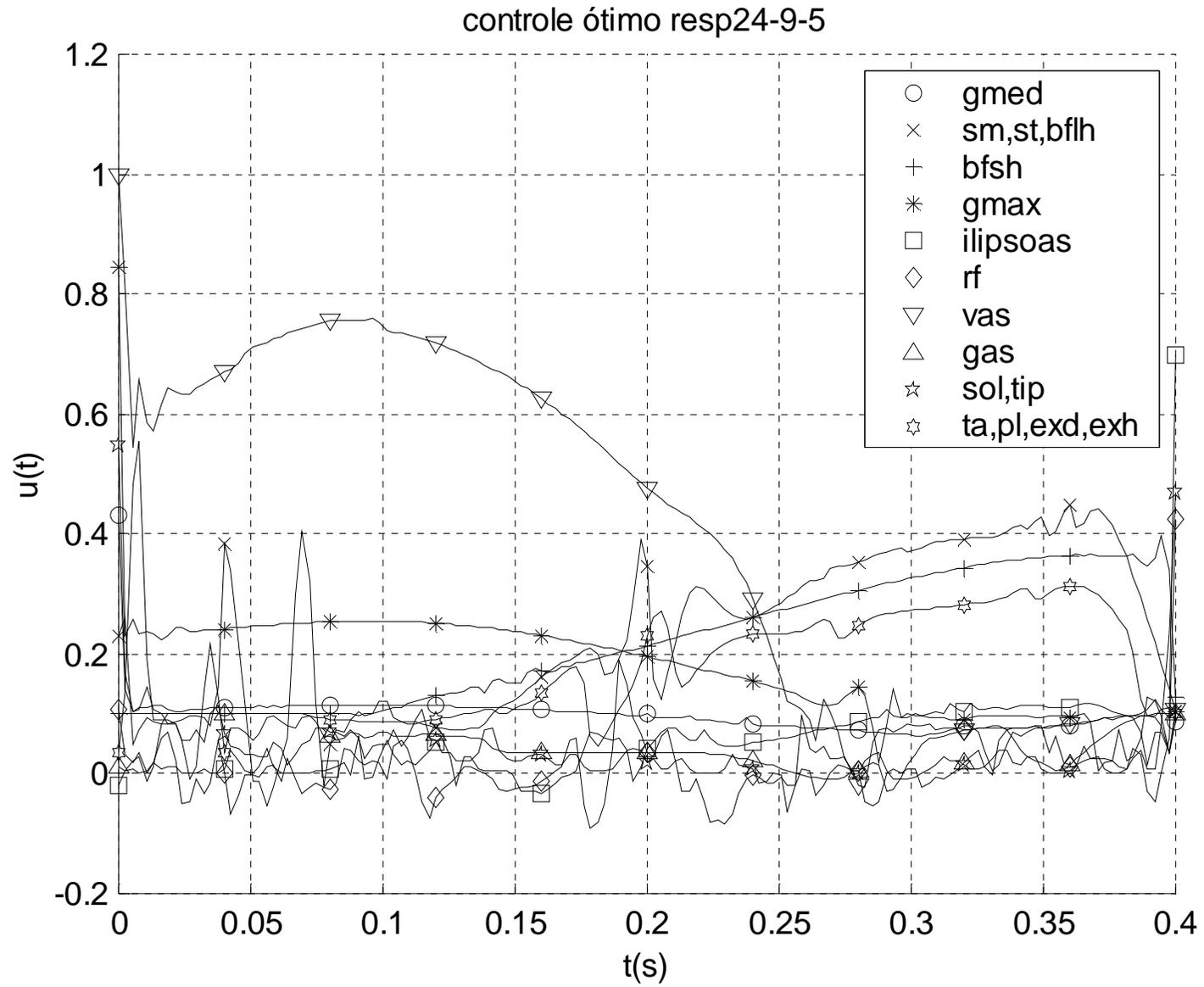


Figura 5.21: Controle 25_9_4

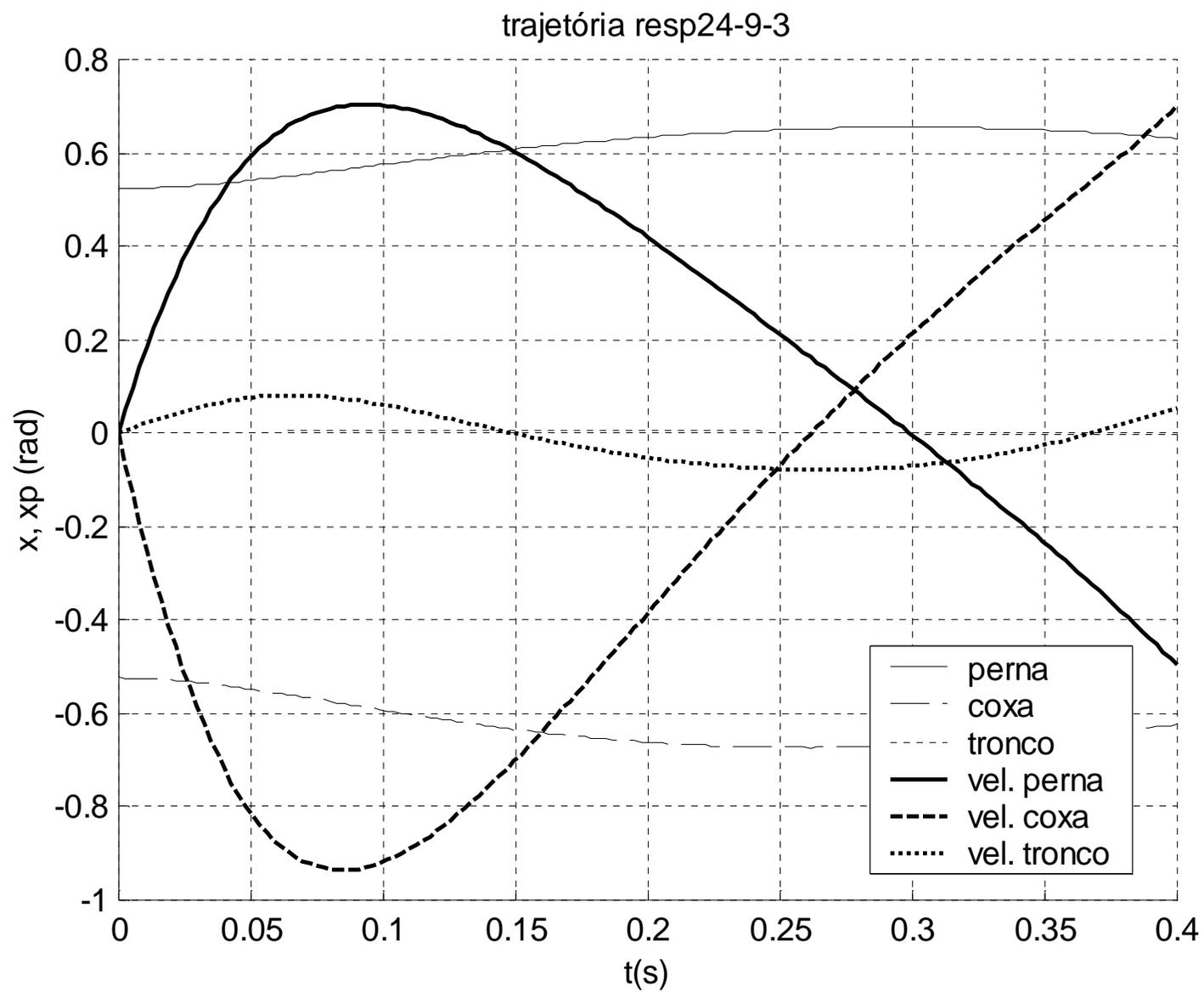


Figura 5.22: Trajetória 24_9_3

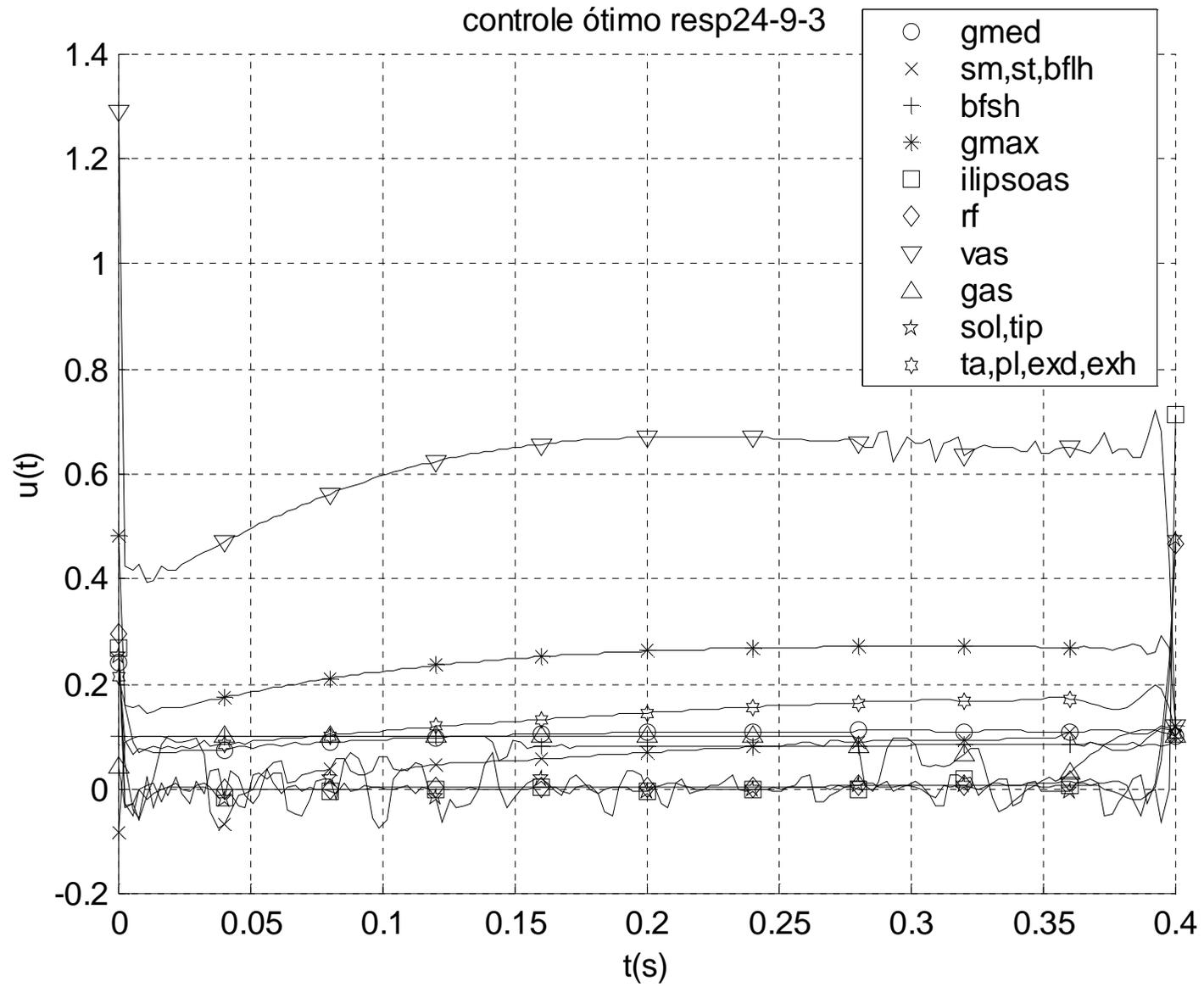


Figura 5.22: Controle 24_9_3

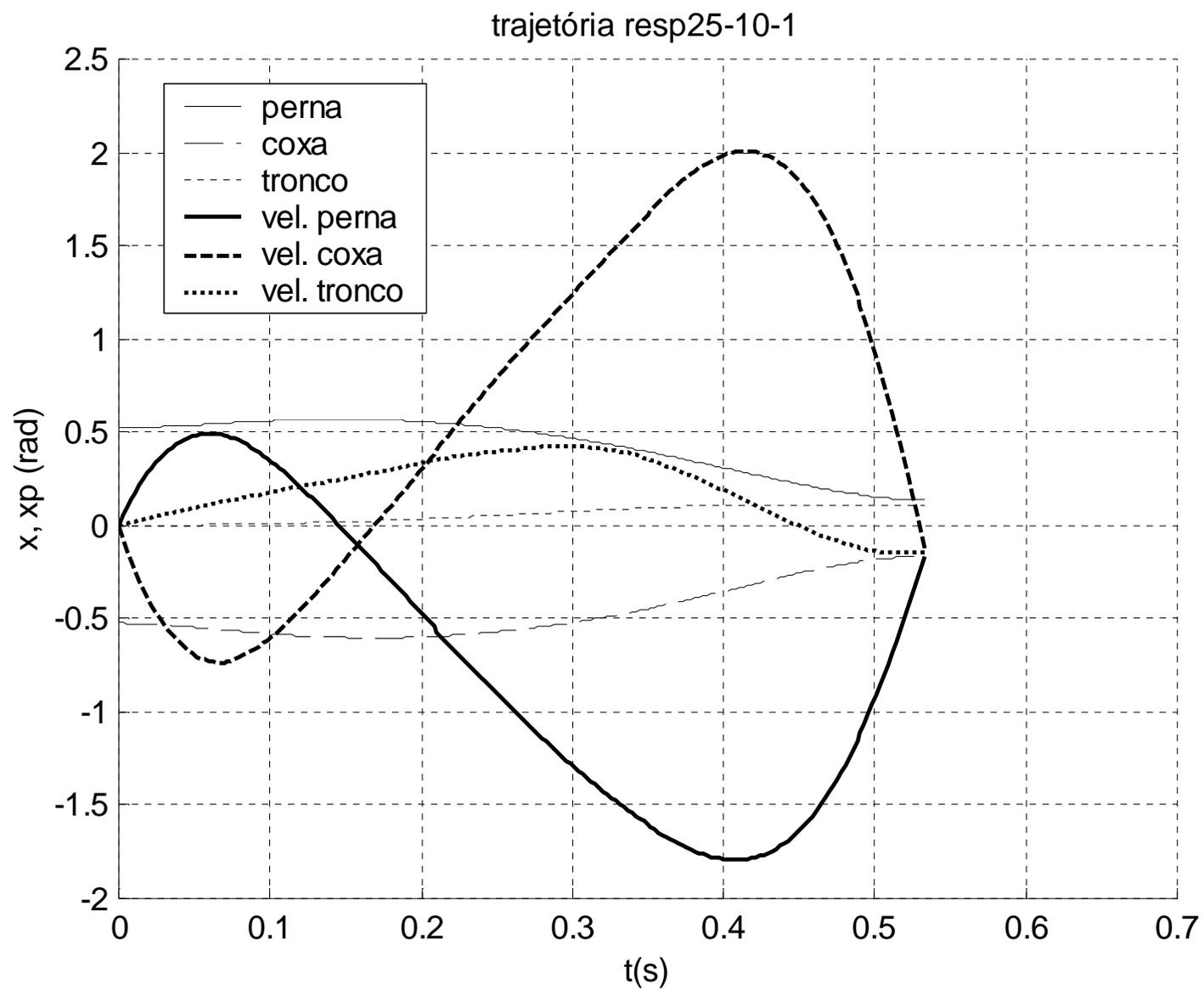


Figura 5.23: Trajetória 25_10_1

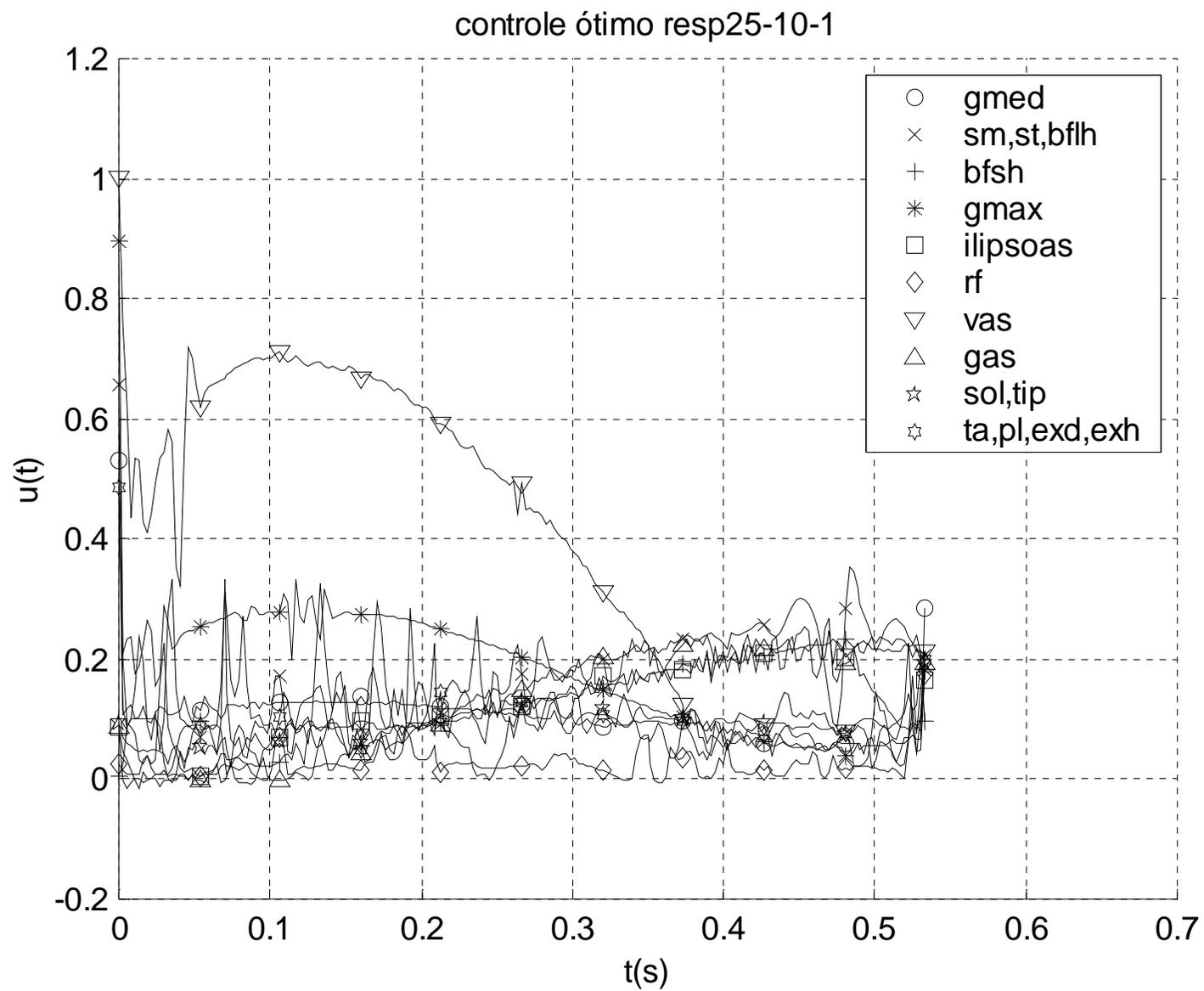


Figura 5.23: Controle 25_10_1

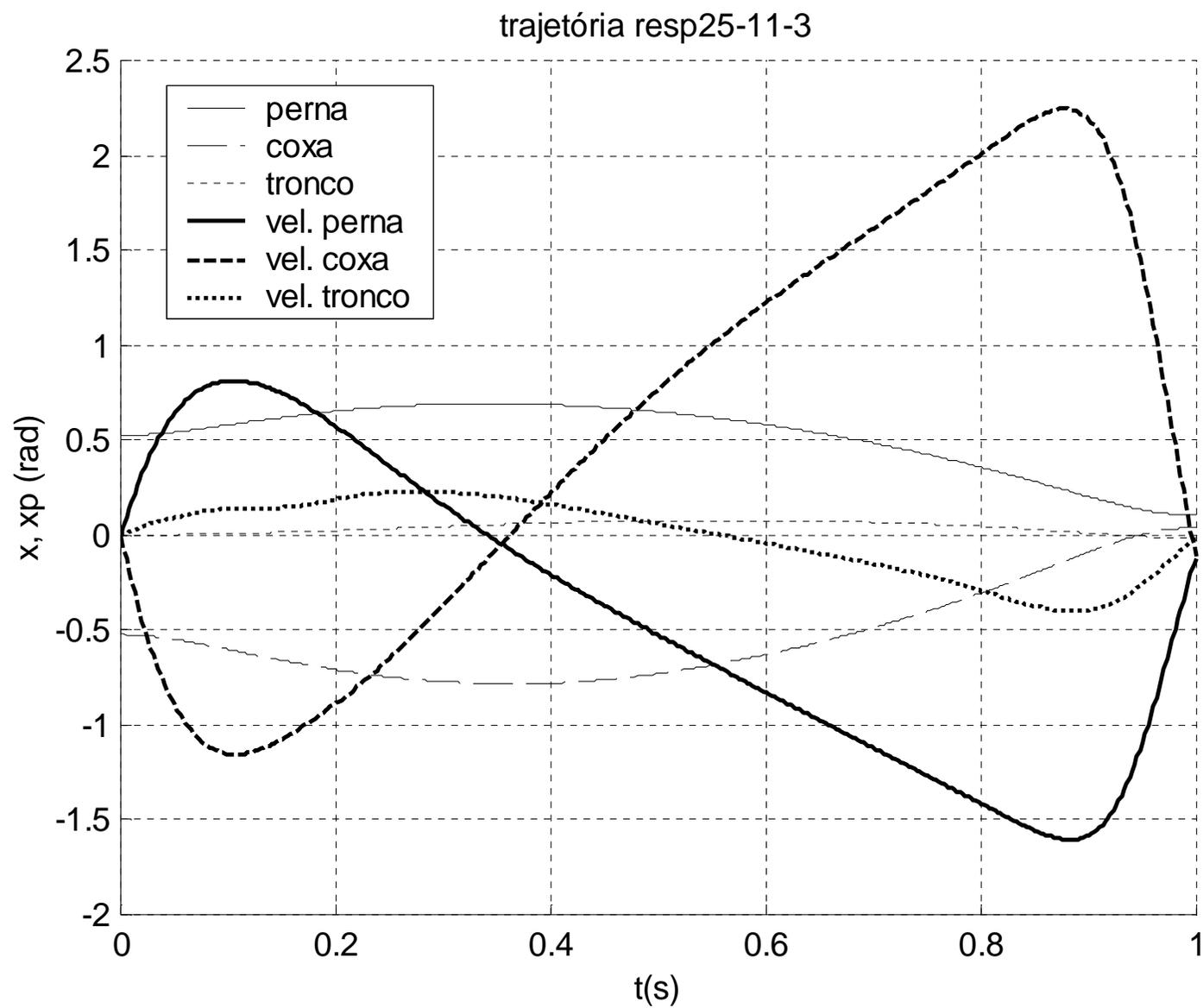


Figura 5.24: Trajetória 25_11_3

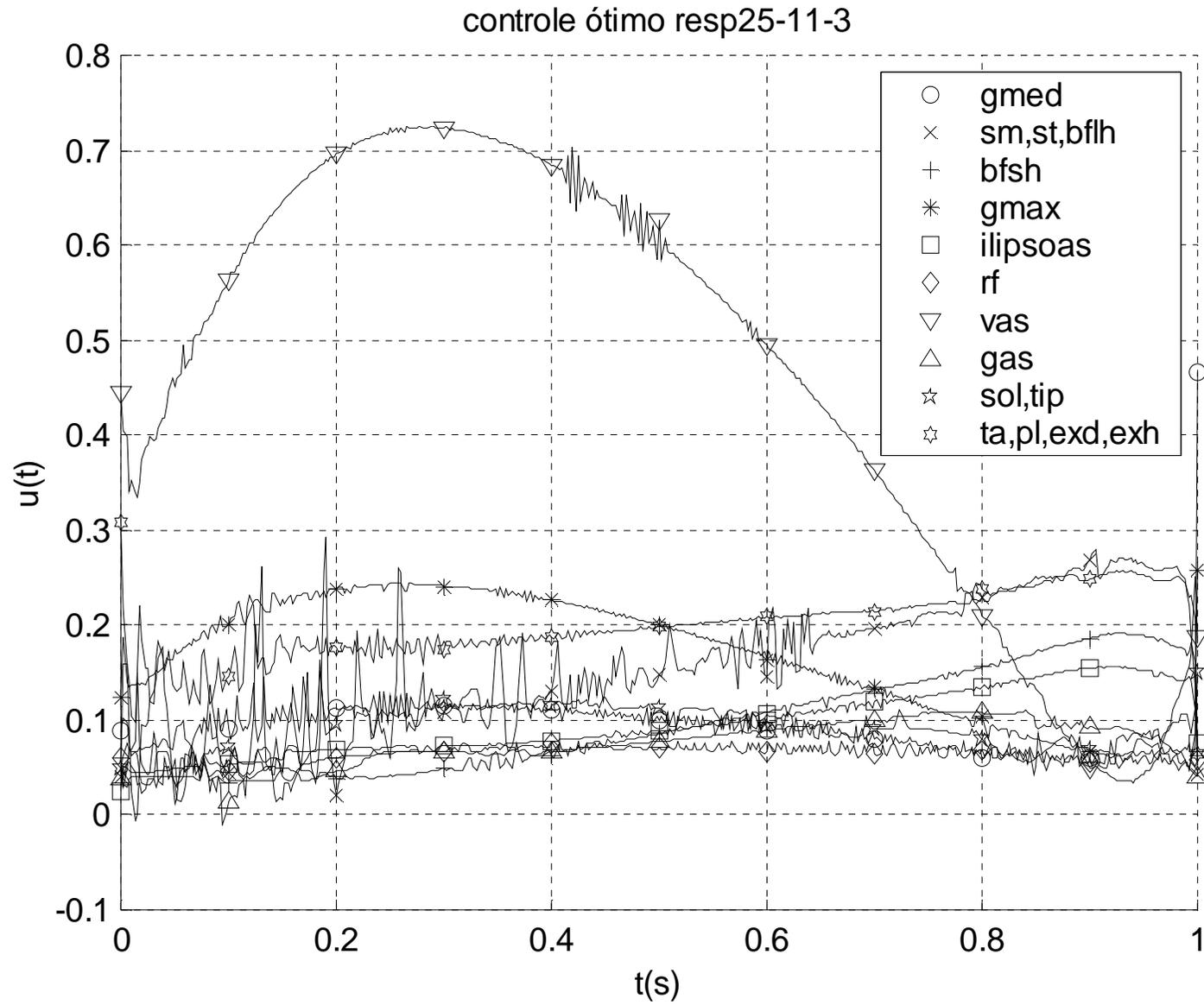


Figura 5.24: Controle 25_11_3

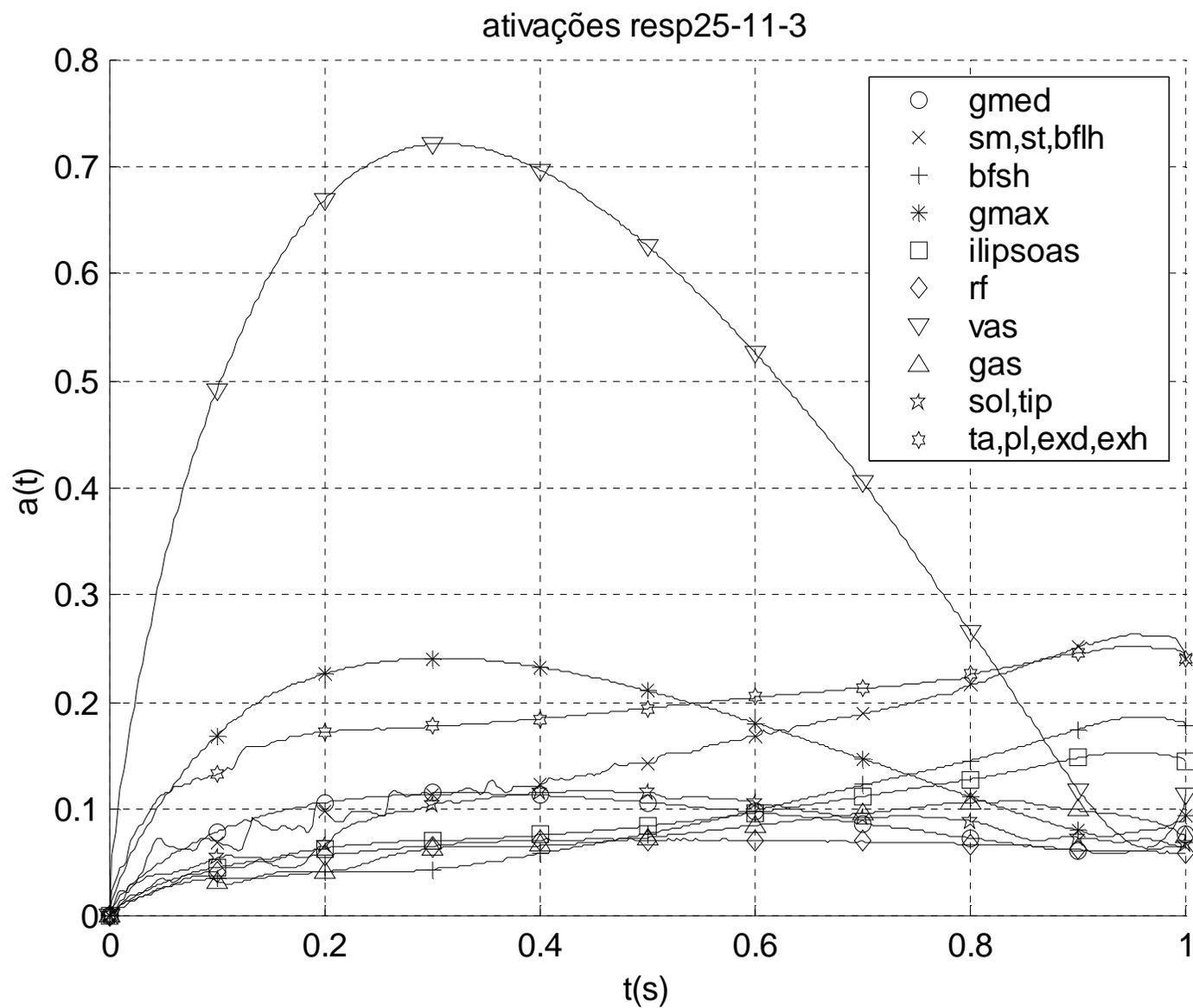


Figura 5.25: Ativações neuromusculares 25_11_3

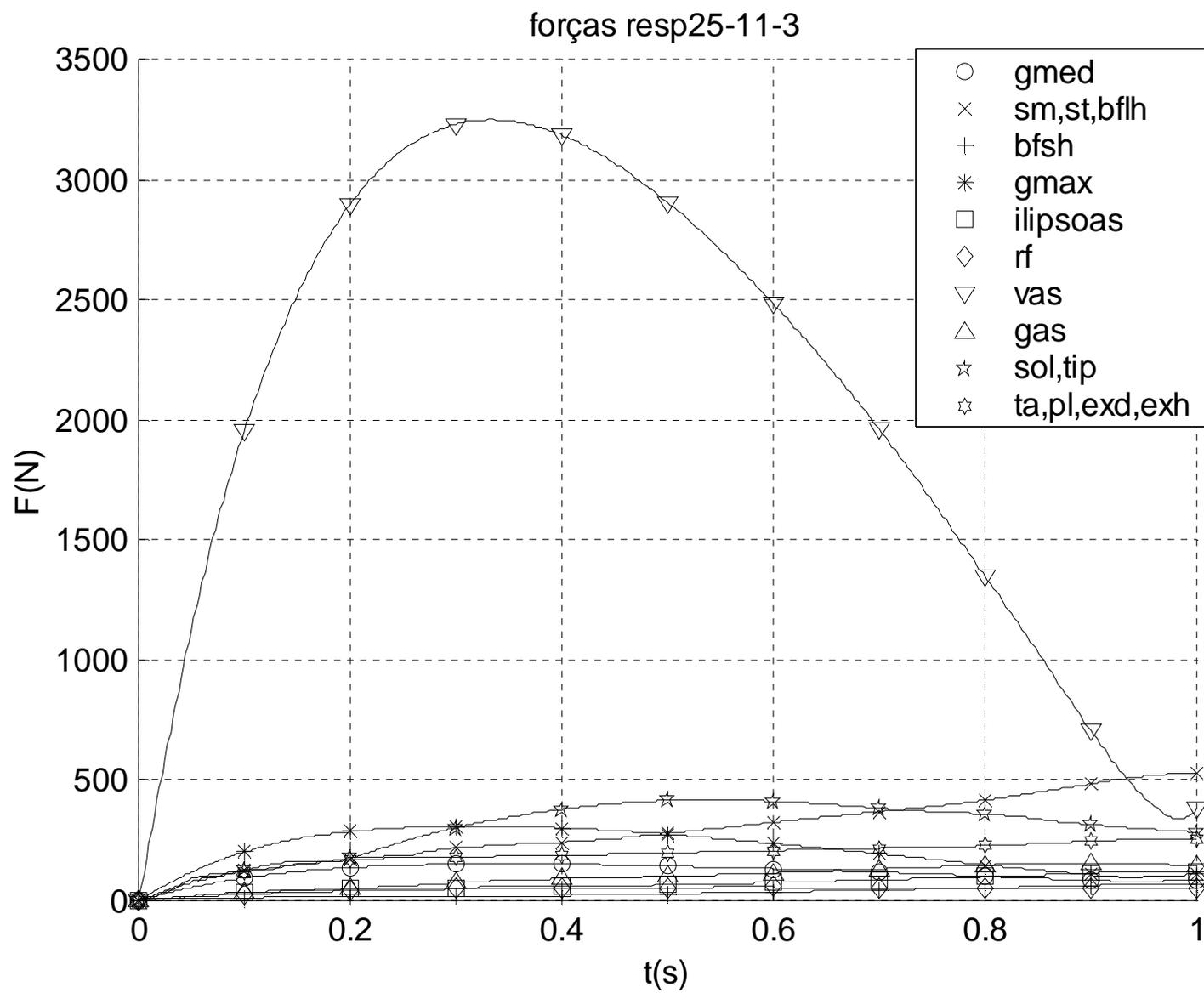


Figura 5.26: Forças musculares 25_11_3

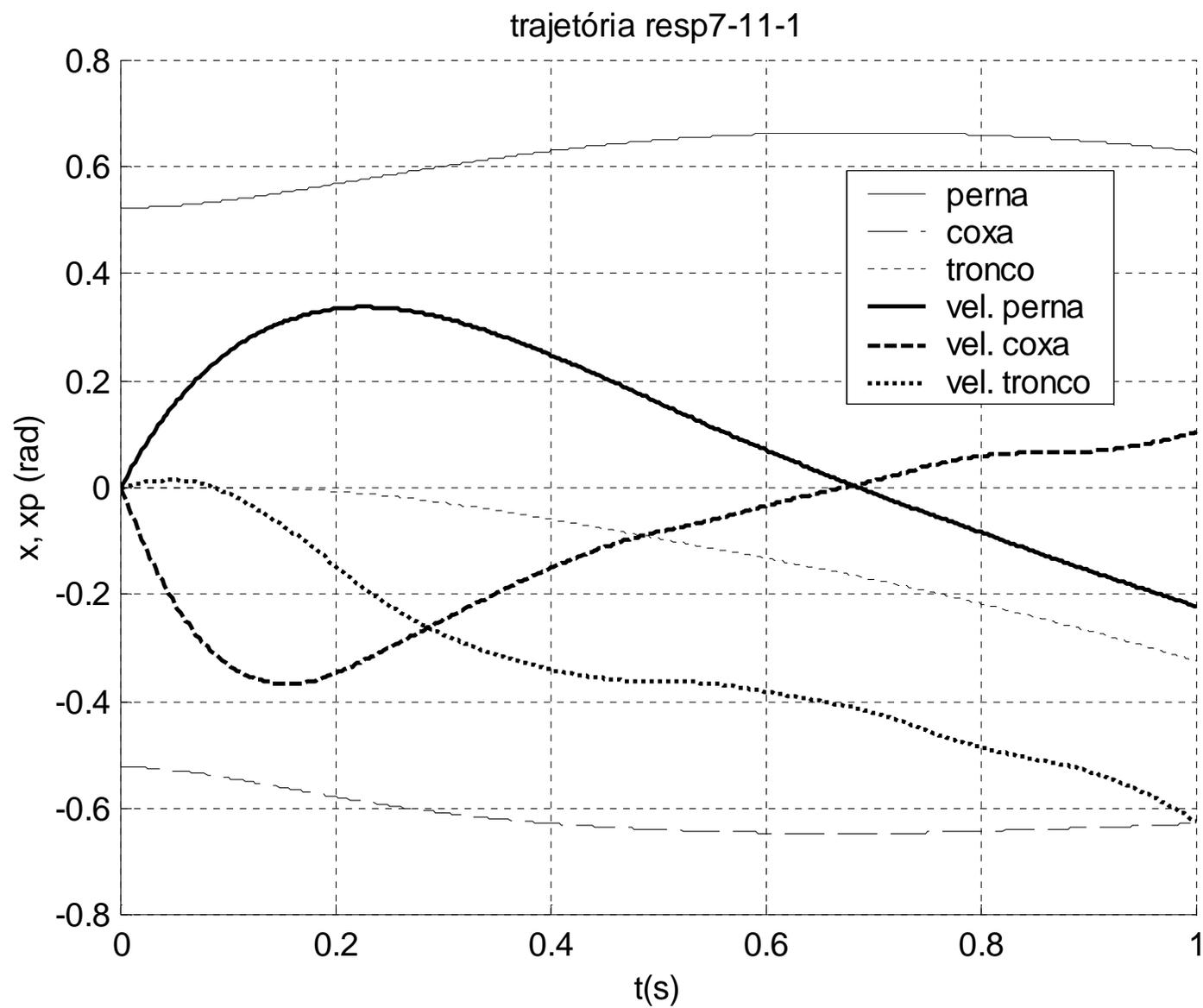


Figura 5.27: Trajetória 7_11_1

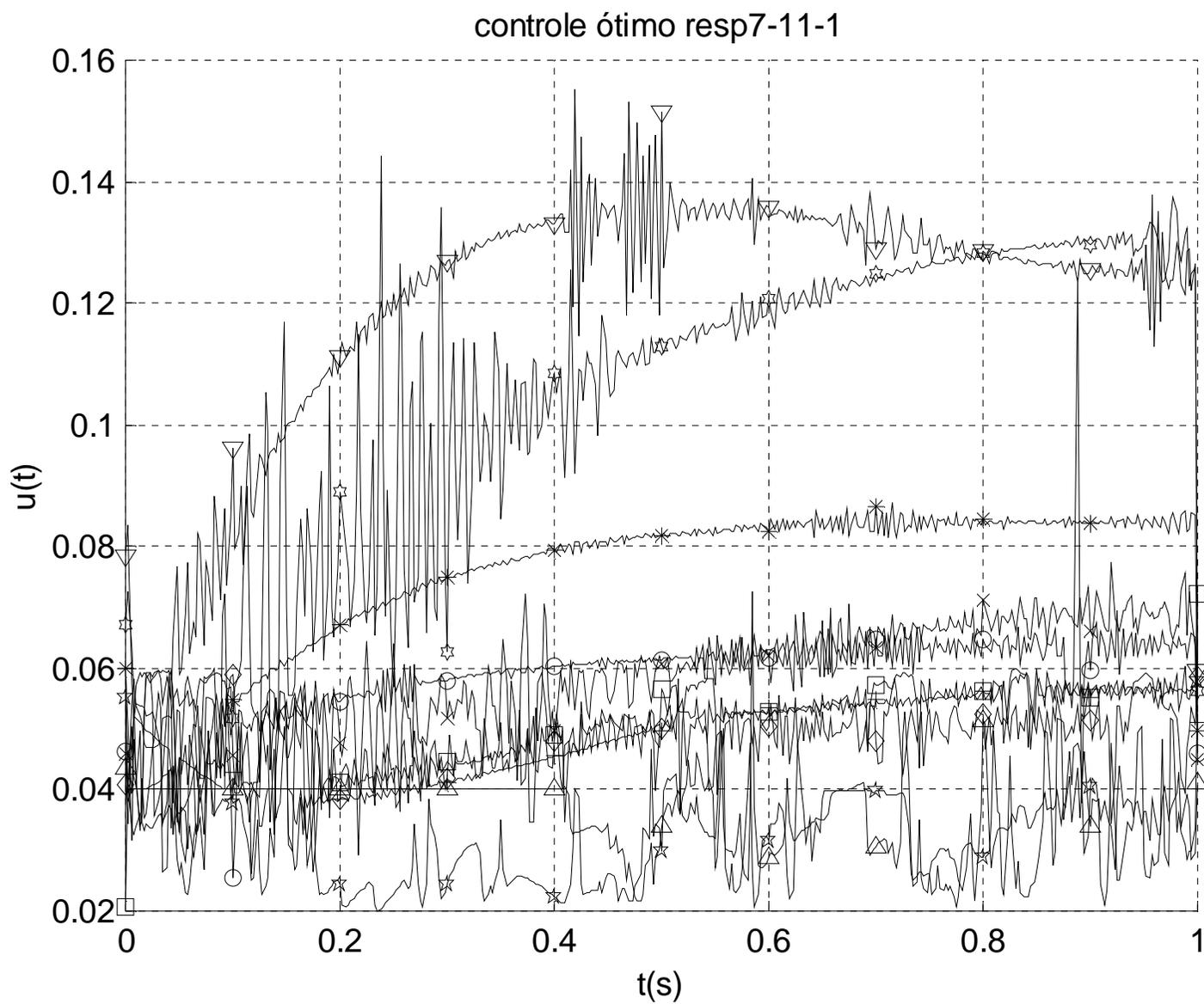


Figura 5.27: Controle 7_11_1

Capítulo 6

Resultados com atuadores não lineares e braços de momento variáveis

Uma vez realizados os testes preliminares do Capítulo 5, onde as principais dificuldades numéricas foram levantadas e, na medida do possível, solucionadas em modelos simplificados, este Capítulo traz os resultados das simulações de um modelo contendo 10 músculos não-lineares e braços de momento variáveis. Para incluir essa nova característica no modelo, foram empregadas as equações de regressão obtidas no Capítulo 2. Deste modo, as estimativas das forças musculares obtidas podem ser consideradas bastante mais realistas, ainda que as curvas de controle obtidas possam diferir, por limitações do modelo da mecânica muscular, mais sensivelmente dos valores eventualmente observados *in vivo*.

Inicialmente, aos valores dos parâmetros dos grupos musculares obtidos no Capítulo anterior, foram acrescentadas curvas médias dos braços de momento em relação às coordenadas generalizadas. Estas curvas foram obtidas a partir das curvas de regressão genéricas do Anexo 1, ponderando cada componente do grupo muscular pela sua força máxima. A partir daí, a interpolação era refeita para a curva média, e novos coeficientes de

regressão eram obtidos. Procurou-se utilizar as curvas de menor erro para cada um dos grupos musculares. As curvas médias obtidas estão mostradas na Tabela 6.1. As posições dos grupos musculares, denotadas pelo seu componente mais importante, estão mostradas na Figura 6.1.

Uma vez determinadas as curvas médias dos braços de momento, suas expressões polinomiais foram inseridas na matriz de braços de momento, substituindo os valores constantes utilizados anteriormente. Posteriormente, as equações de movimento e os gradientes necessários para a solução do OCP foram calculadas da mesma maneira que nos modelos anteriores, fazendo uso do pacote de manipulação algébrica do Matlab v5.3.

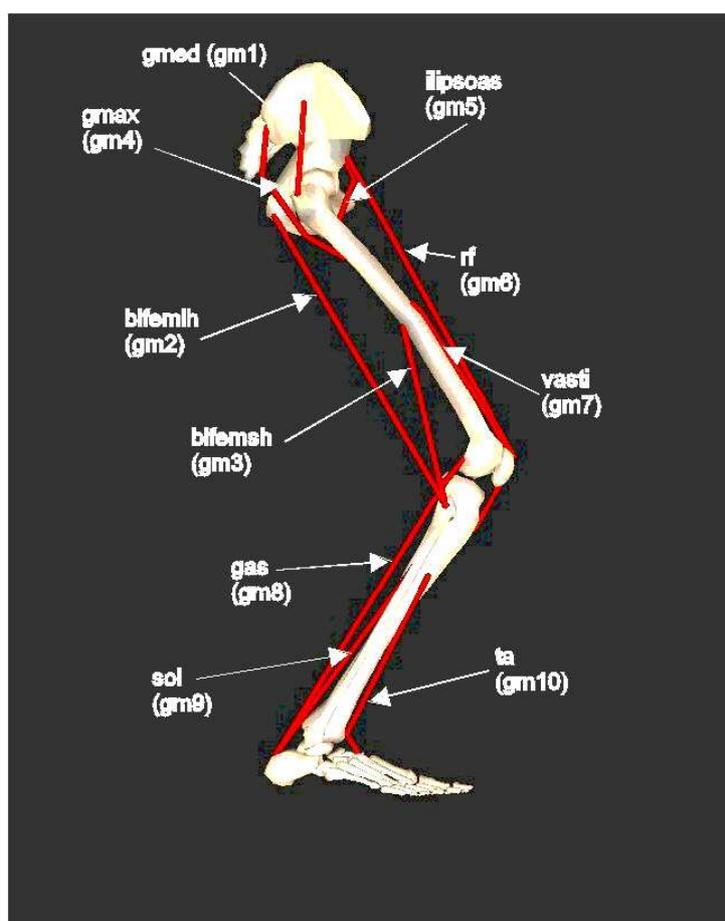
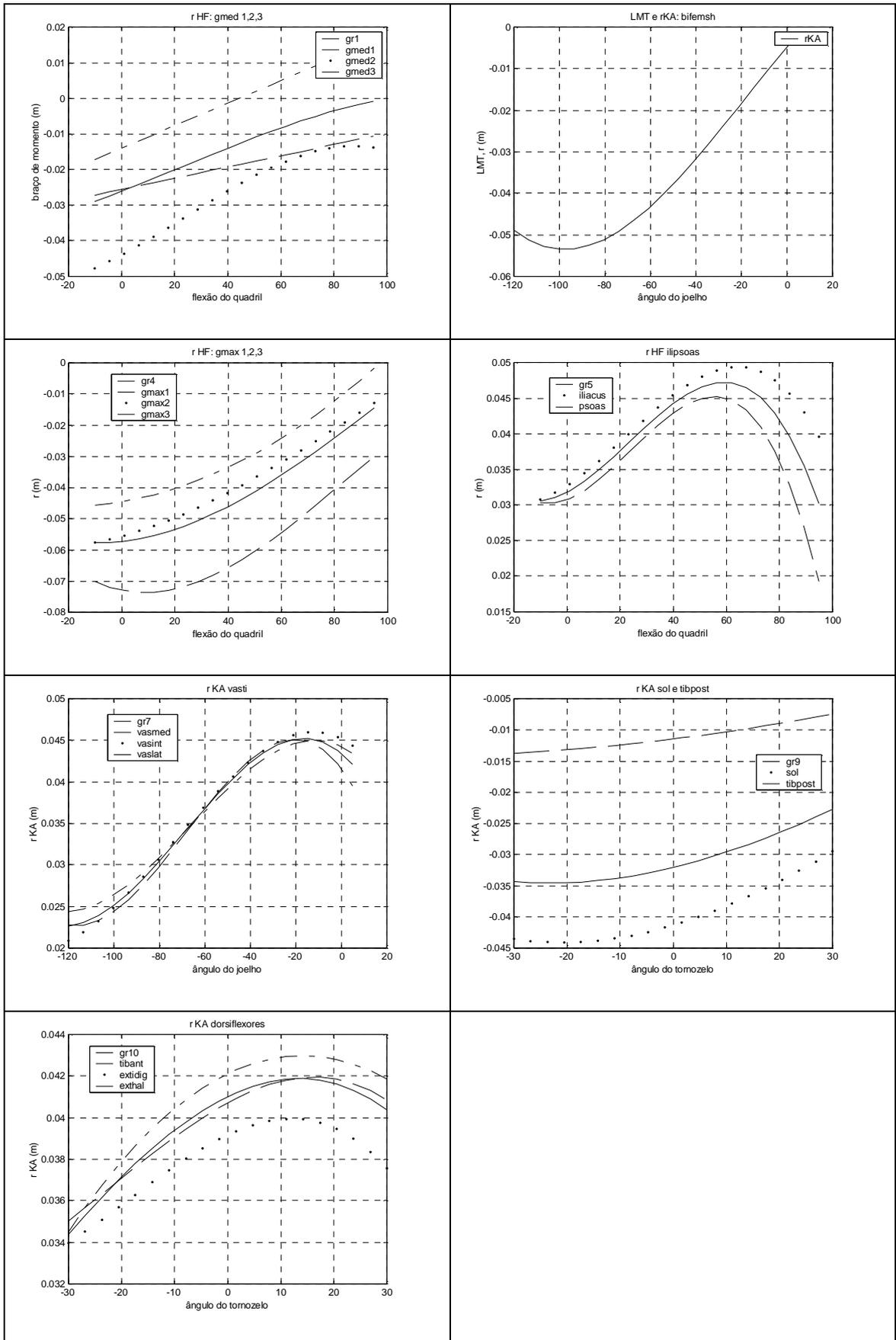


Figura 6.1: Grupos musculares denotados pelos seus componentes mais importantes



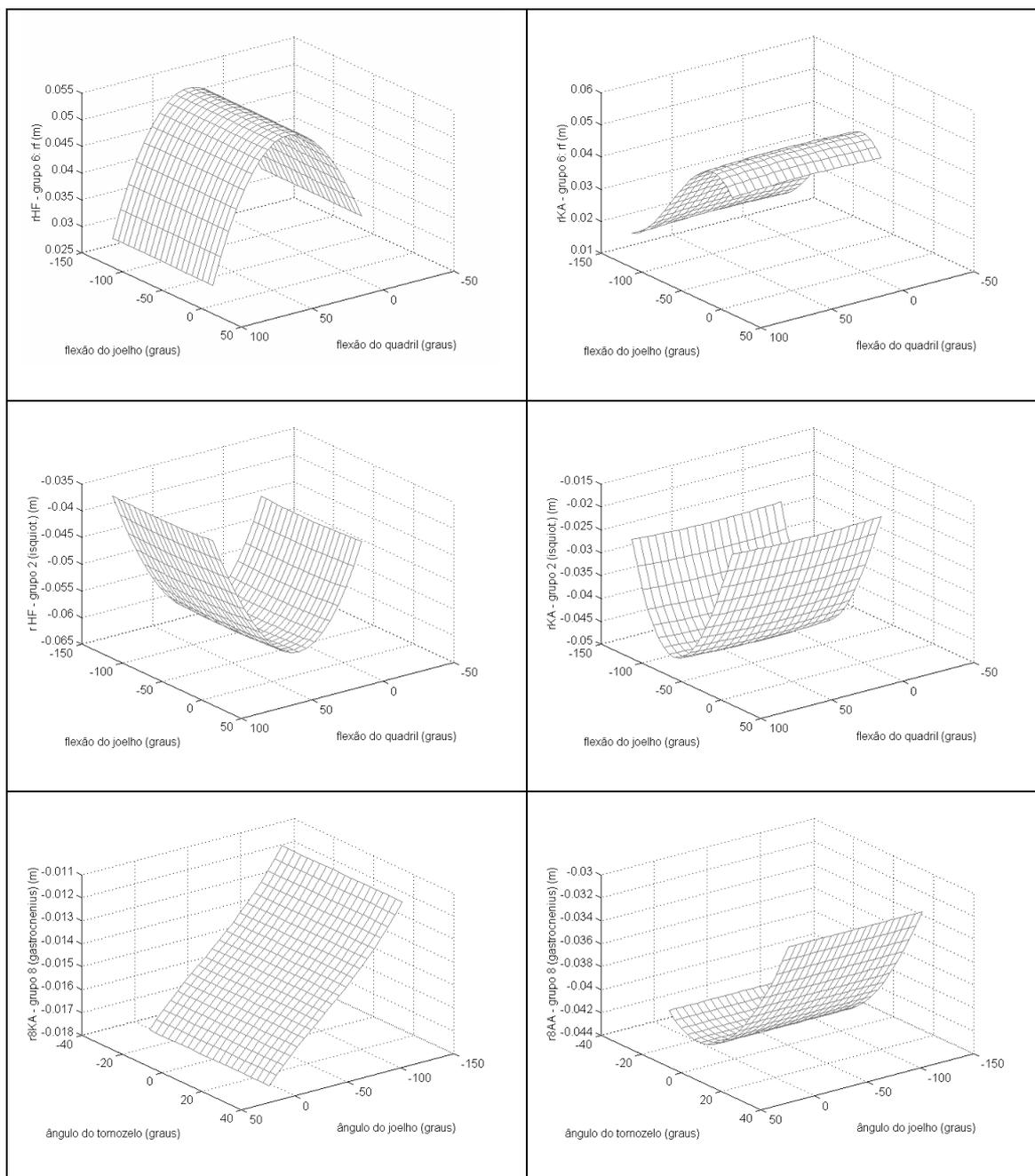


Tabela 6.1: Curvas médias dos braços de momento para os 10 grupos musculares selecionados. As superfícies mostradas na segunda parte da tabela se referem aos braços de momento de músculos biarticulares em relação a cada uma das articulações envolvidas. Para esses grupos, as curvas de cada componente não estão mostradas.

6.1 Resultados com $t_f=0.4$ segundos

A primeira parte dos resultados se refere a simulações realizadas sem o emprego da estratégia da aceleração da gravidade variável (ver Tabela 6.2 para todas as simulações realizadas nesse item). Foram utilizados 150 pontos na malha de discretização e integrador RK4. A função de custo foi mantida a mesma do modelo simplificado, isto é, a soma das derivadas das forças musculares, tanto nesse como no próximo item. Algumas tentativas iniciais, utilizando a mesma estimativa inicial u_{00} do caso com braço de momento constante, não foram bem sucedidas; em seguida, foi empregado para o u_{00} o valor constante de $u=0.2$ para todos as variáveis de controle, e desta vez foram obtidas apenas algumas respostas convergentes para o vínculo terminal de $\pi/5$, extremamente ruidosas e sem um padrão de excitação claramente identificável. Uma segunda série de simulações, utilizando uma nova estimativa de u_{00} , u_{00b} , forneceram respostas convergentes, de acordo com a seqüência:

4_12_1 (Figura 6.5) \Leftarrow 3_12_4 (Figura 6.4) \Leftarrow 2_12_2 (Figura 6.3) \Leftarrow 2_12_1 (Figura 6.2) \Leftarrow u_{00b}

Na primeira simulação, 2_12_1, a função de controle e a trajetória obtidas parecem plausíveis, assim como o valor da função de custo ($f = 4.4688$). Entretanto, quando o vínculo terminal foi reduzido para $\pi/15$, a função de custo apresentou um aumento desusado para 25.0614, verificando-se uma intensa coativação do *gastrocnemius* e do *tibialis anterior*, dois músculos antagonistas. Quando o vínculo terminal restringiu-se para $\pi/30$ (3_12_4), mantendo o vínculo superior de controle em 10 (ainda que os valores máximos apresentados foram de 2.23), a função de custo atingiu 36.4585. Quando U_{sup} foi reduzido para 2 (4_12_1), notou-se um aumento significativo do ruído, bem como da própria função de custo ($f = 51.9406$). Por outro lado, o padrão de ativação mostrou-se

bastante curioso, com uma forte ativação do *bíceps femoris caput brevis* e diminuição da atividade do *gastrocnêmius*. Estes resultados sugerem que o algoritmo de otimização atingiu um mínimo local que se afasta de um padrão fisiologicamente razoável.

Na tentativa de escapar deste mínimo local, procurou-se realizar uma terceira seqüência de simulações, desta vez mantendo o vínculo superior de controle num valor reduzido desde as primeiras simulações. Procedendo desta maneira, foi possível obter a seguinte seqüência de simulações, que culminou num resultado convergente:

13_12_2 (Figura 6.12, Figura 6.14, Figura 6.15) e 13_12_3 (Figura 6.13) \Leftarrow 11_12_3 (Figura 6.11) \Leftarrow 11_12_2 (Figura 6.10) \Leftarrow 6_12_6 (Figura 6.9) \Leftarrow 6_12_5 \Leftarrow 6_12_1 \Leftarrow 5_12_3 (Figura 6.8) \Leftarrow 1_12_1 (Figura 6.7) \Leftarrow 30_11_5 (Figura 6.6) \Leftarrow u00

Da primeira (30_11_5) para a segunda (1_12_1) simulação o vínculo superior de controle foi reduzido de 10 para 2 e logo para 1.5 na terceira simulação (5_12_3). Na seguinte, (6_12_1), o vínculo foi reduzido para 1.4 e a tolerância de violação de controle aumentada para 0.04, ou o NPSOL não era capaz de achar pontos factíveis para os vínculos não-lineares. Na simulação 6_12_5, o vínculo superior de controle foi reduzido para 1.3 e o inferior para 0.03 e na seguinte (6_12_6), o vínculo superior atingiu finalmente 1. A partir daí, os vínculos terminais foram reduzidos para $\pi/10$, as tolerâncias de otimização e violação de vínculos para 0.001 e 0.006 respectivamente e o vínculo inferior de controle para 0.01. Reduzindo os vínculos terminais para $\pi/20$ (11_12_3), foi necessário aumentar a tolerância de otimização para 0.01, mantida para a próxima simulação, desta vez já com os vínculos terminais em $\pi/30$ (13_12_2). Foi possível nesse ponto reduzir a tolerância de otimização para 0.001 (13_12_3), atenuando significativamente o ruído numérico (Figura 6.12 e Figura 6.13), ainda que as funções de custo diferissem pouco. O valor alcançado da função de custo para a simulação 13_12_3 ($f = 7.0458$) foi bastante inferior ao valor obtido na segunda série de simulações, e os

padrões de controle bem mais plausíveis. Na Figura 6.14 estão mostradas as ativações neuromusculares para a simulação 13_12_3 e na Figura 6.15 as forças musculares.

simul.	ci ⁱ	v. term. ⁱⁱ	tol. otm.	tol. cont.	U _{inf}	U _{sup}	RK ⁱⁱⁱ	N	Naug ^{iv}	Uoaug ^v	término ^{vi}	n° its	min.cpu ^{vii}	f	fig. n.	obs.
30_11_1	u00	$\pi/5$	0.1	0.1	0	10	4	150	-	-	1	343	79	6.1981		
30_11_2	30_11_1	$\pi/5$	0.05	0.1	0	1	4	150			3	1	0.6	10.5807		
30_11_3	30_11_1	$\pi/5$	0.05	0.1	0	10	4	150			1	143	47	8.5670		
30_11_4	u00	$\pi/5$	0.01	0.1	0	1	4	150			3	1	0.73	2.9957		
30_11_5	u00	$\pi/5$	0.05	0.1	0	10	4	150			1	388	80	6.2346	Figura 6.6	obs.1
30_11_3b	30_11_1	$\pi/5$	0.05	0.1	0	10	4	150			1	60	16	8.5958		
1_12_1	30_11_5	$\pi/5$	0.01	0.01	0	2	4	150			1	412	212	5.9225	Figura 6.7	
1_12_2	1_12_1	$\pi/5$	0.005	0.005	0	1	4	150			7	0				
1_12_2	1_12_1	$\pi/5$	0.005	0.005	0.01	1	4	150			7	96	108	15.5547		
2_12_1	u00b	$\pi/5$	0.01	0.01	0.05	10	4	150			1	498	304	4.4688	Figura 6.2	
2_12_2	2_12_1	$\pi/15$	0.01	0.01	0.05	10	4	150			1	707	465	25.0614	Figura 6.3	
3_12_1	2_12_1	$\pi/15$	0.01	0.04	0.05	1	4	150			3	10	60	7.0479		
3_12_2	2_12_1	$\pi/15$	0.01	0.04	0.05	10	4	150			1	619	254	68.3408		
3_12_3	2_12_2	$\pi/15$	0.01	0.01	0.05	1.5	4	150			7	7	22	3.3070		
3_12_4	2_12_2	$\pi/30$	0.01	0.01	0.05	10	4	150			1	643	503	36.4585	Figura 6.4	
4_12_1	3_12_4	$\pi/30$	0.1	0.01	0.05	2	4	150			1	776	746	51.9406	Figura 6.5	
4_12_2	2_12_4	$\pi/30$	0.01	0.01	0.05	2	4	150			3	1204	839	29.6083		
5_12_1	u00b	$\pi/5$	0.1	0.1	0.05	1	4	150			3	1	0.3	0.1617		
5_12_2	u00b	$\pi/5$	0.1	0.04	0.05	1	4	150			3	127	45	8.9447		
5_12_3	1_12_1	$\pi/5$	0.01	0.01	0.05	1.5	4	150			1	338	120	2.9627	Figura 6.8	
5_12_4	5_12_3	$\pi/5$	0.01	0.01	0.05	1	4	150			3	201	145	3.4153		
5_12_5	5_12_3	$\pi/5$	0.01	0.01	0.05	1.4	4	150			2	132	146	9.9130		
5_12_6	5_12_3	$\pi/5$	0.01	0.01	0.05	1.2	4	150			3	32	87	25.7699		
6_12_1	5_12_3	$\pi/5$	0.01	0.04	0.05	1.4	4	150			1	532	136	5.6279		
6_12_2	5_12_3	$\pi/5$	0.06	0.04	0.05	1.4	4	150			1	510	132	5.6193		
6_12_3	6_12_1	$\pi/5$	0.01	0.04	0.05	1.2	4	150			2	31	36	10.8272		
6_12_4	6_12_1	$\pi/5$	0.1	0.04	0.05	1.2	4	150			3	1	49	2.8694		
6_12_5	6_12_1	$\pi/5$	0.06	0.02	0.03	1.2	4	150			1	109	96	4.5115		
6_12_6	6_12_5	$\pi/5$	0.06	0.02	0.03	1	4	150			1	169	75	4.5169	Figura 6.9	
6_12_7	6_12_6	$\pi/10$	0.06	0.02	0.03	1	4	150			2	41	96	4.4952		
6_12_8	6_12_6	$\pi/15$	0.06	0.02	0.03	1	4	150			2	31	130	6.3407		
7_12_1	6_12_6	$\pi/15$	0.01	0.02	0.03	1	4	150			2	31	96	4.4952		

7_12_2	6_12_6	$\pi/15$	0.1	0.02	0.03	1	4	150			2	31	96	4.4952		
7_12_3	6_12_6	$\pi/15$	0.01	0.05	0.06	1	4	150			5	-	-	-		
7_12_4	6_12_6	$\pi/15$	0.01	0.09	0.1	1	4	150			5	-	-	-		
9_12_1	6_12_6	$\pi/10$	0.1	0.05	0.06	1	4	150			5	-	-	-		
9_12_2	6_12_6	$\pi/10$	0.1	0.02	0.03	1	4	150			2	41	122	6.3407		
11_12_1	6_12_6	$\pi/10$	0.001	0.02	0.03	1	4	150			5	-	-	-		
11_12_2	6_12_6	$\pi/10$	0.001	0.006	0.01	1	4	150			1	742	390	8.2805	Figura 6.10	
11_12_3	11_12_2	$\pi/20$	0.001	0.006	0.01	1	4	150			1	1016	794	11.7078		
11_12_4	11_12_2	$\pi/15$	0.001	0.006	0.01	1	4	150			3	1122	964	5.2248		
12_12_1	11_12_3	$\pi/30$	0.001	0.006	0.01	1	4	150			3	807	724	10.4693		
12_12_2	11_12_3	$\pi/40$	0.001	0.006	0.01	1	4	150			3	126	176	14.7865		
12_12_3	11_12_3	$\pi/30$	0.01	0.006	0.01	1	4	150			3	513	589	9.7860	Figura 6.11	
13_12_1	11_12_2	$\pi/30$	0.001	0.006	0.01	1	4	150			3	62	151	5.1911		
13_12_2	11_12_3	$\pi/30$	0.1	0.0001	0.001	1	4	150			1	1246	1235	7.0756	Figura 6.12	
13_12_3	11_12_3	$\pi/30$	0.001	0.0001	0.001	1	4	150			1	1771	1033	7.0458	Figura 6.13	

Tabela 6.2: Simulações com braço de momento variável, tempo final 0.4 segundos

```

obs.1: alterado npsol.opt (para valores os mais estreitos) e sem -mf no
compilador
obs.2: if u000(i,j)^2 <= 0.0001, u000(i,j)=0.2;
obs.3: u0aug=0.10*ones(10,Naug).*rand(10,Naug)*4
obs.4: u0aug=0.10*ones(10,Naug).*rand(10,Naug)*3
obs.5: Quando o u0aug é muito baixo, aparecem problemas de div. por zero
(floating point error). Se for muito alto, o sistema cai. Utilizando um
controle variável baixos níveis de controle não geram erro de div. por zero
    for i=1:10,
        u0aug(i,:)=0.2+0.15*(sin(2*pi*40*taug(length(t):length(taug)-1)));
    end
obs.6: u0aug(i,:)=0.2+0.15*(sin(-2*pi*20*taug(length(t):length(taug)-1)))
obs.7: u0aug(i,:)=0.2+0.15*(sin(-2*pi*10*taug(length(t):length(taug)-1)))
obs.8: u0aug(i,:)=0.2+0.2*(sin(-2*pi*10*taug(length(t):length(taug)-1)));
obs.9: u0aug(i,:)=0.2+0.2*(sin(-2*pi*60*taug(length(t):length(taug)-1)));
obs.10: mudança no arquivo npsol.opt: Line search tolerance = 1; Minor
iteration limit = 5000
obs.10: mudança no arquivo npsol.opt: Line search tolerance = 5;

```

ⁱ Vínculo de desigualdade $x_i^2 \leq v \cdot \text{term}^2$.

ⁱⁱ RK5=LOSODA

ⁱⁱⁱ número de pontos a mais

^{iv} valor adotado em todos os controles na ci do tempo aumentado

^v

- 1: término normal
- 2: não foi encontrado ponto factível para os vínculos não lineares
- 3: falha na busca linear
- 4: não é possível calcular o gradiente dos vínculos
- 5: erro de memória
- 6: completado número máximo de iterações
- 7: erro de ponto flutuante durante cálculo do gradiente do vínculo

^{vi} Pentium III 600MHz.

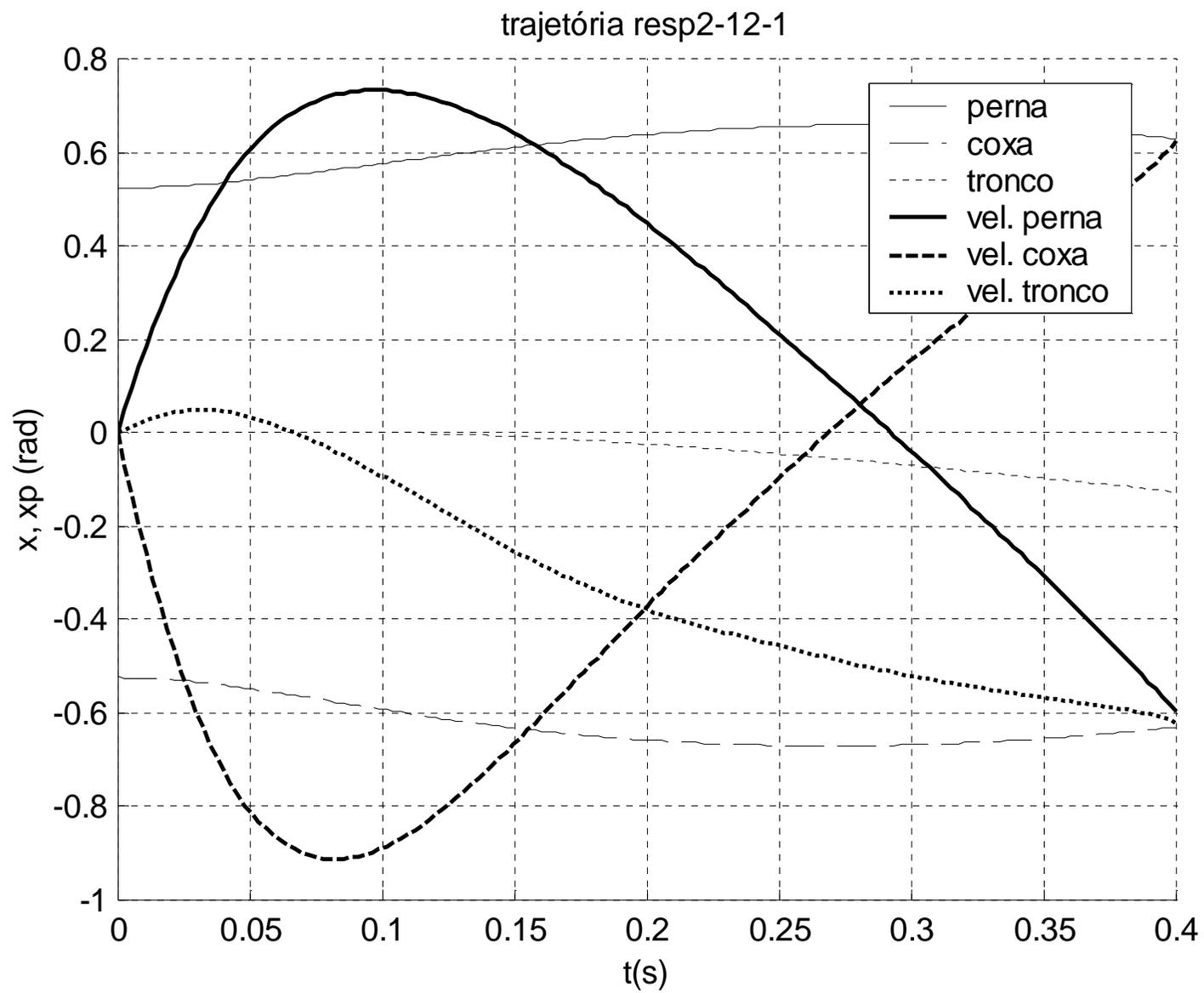


Figura 6.2: Trajetória 2_12_1

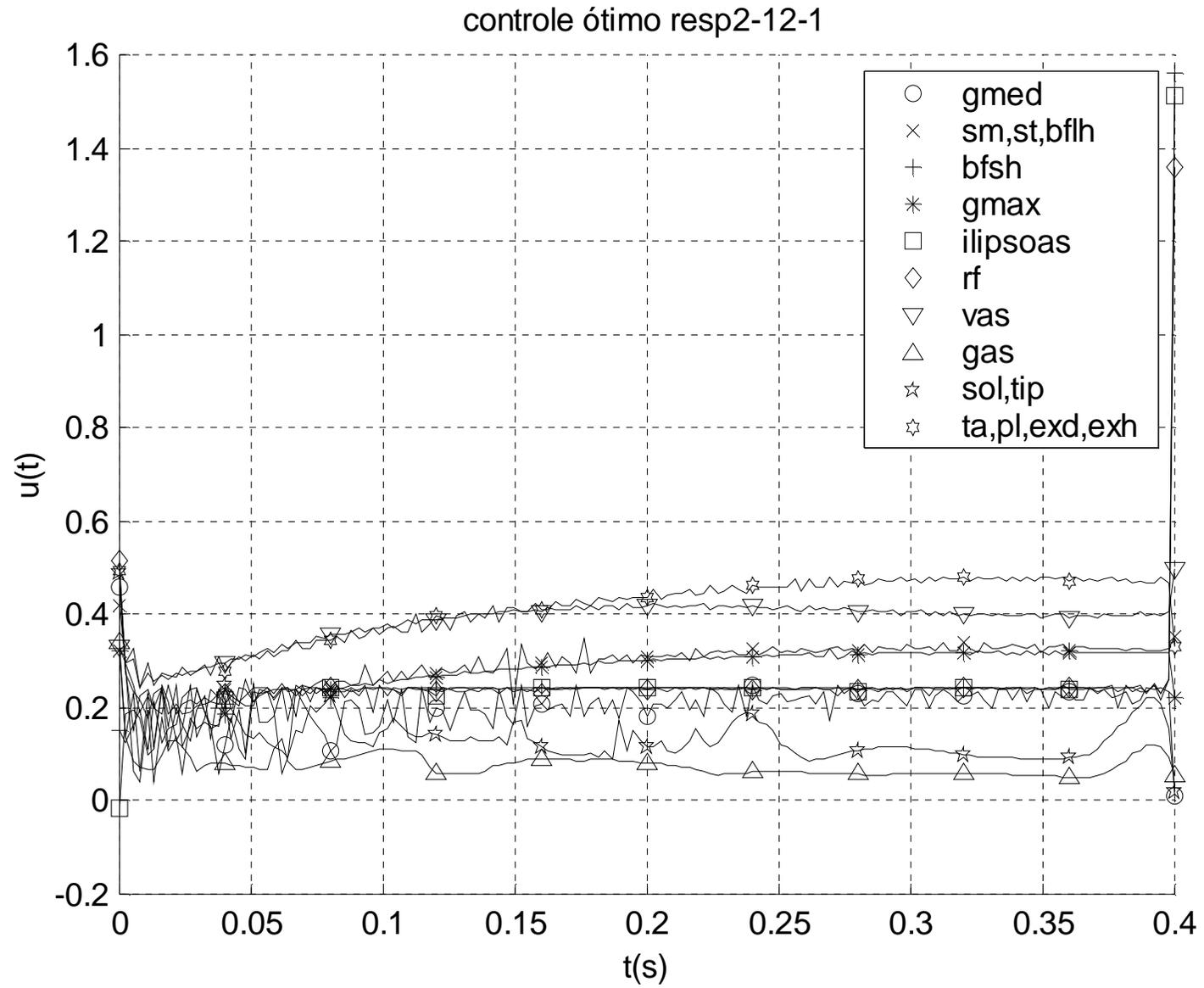


Figura 6.2: Controle 2_12_1

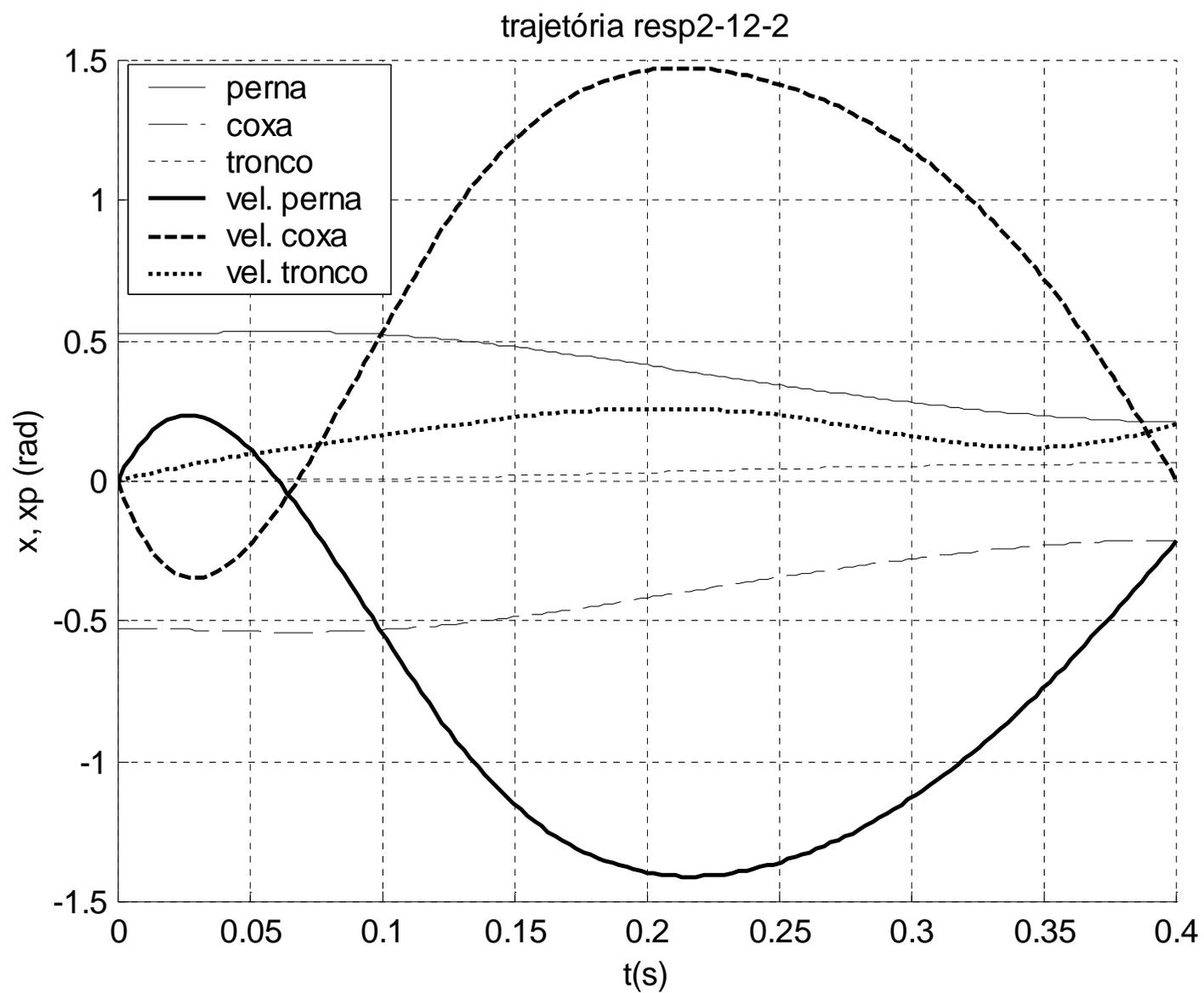


Figura 6.3: Trajetória 2_12_2

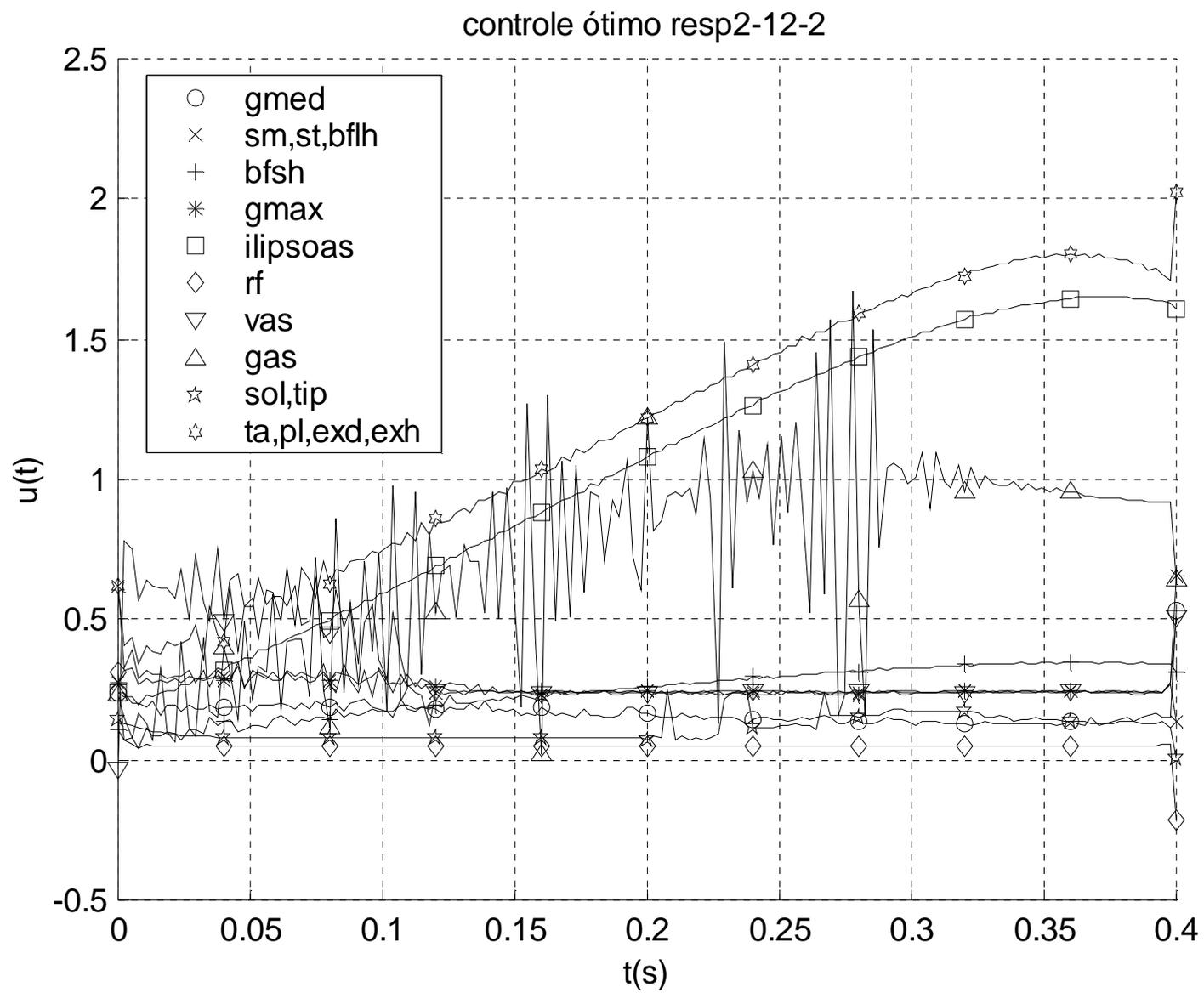


Figura 6.3: Controle 2_12_2

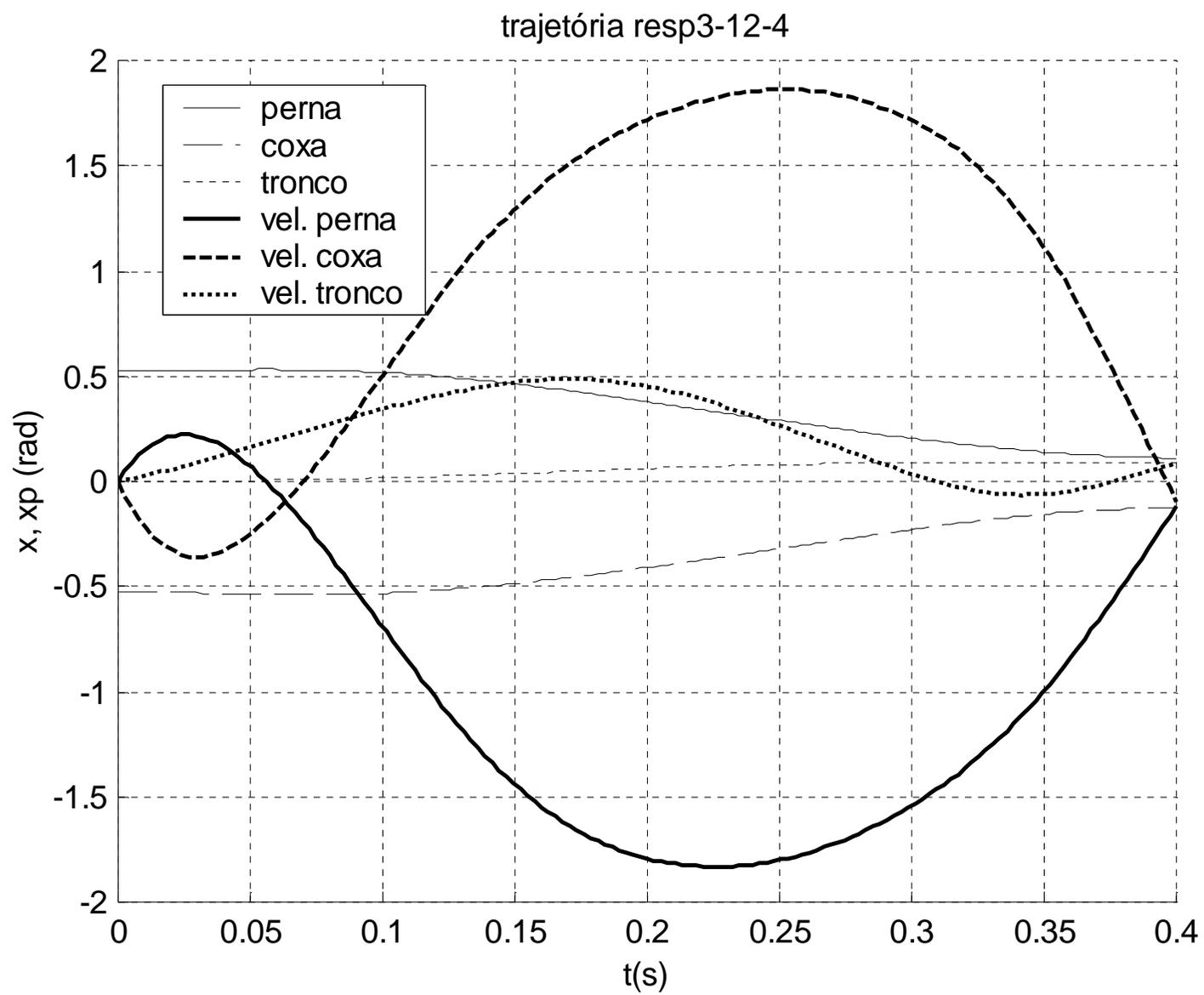


Figura 6.4: Trajetória 3_12_4

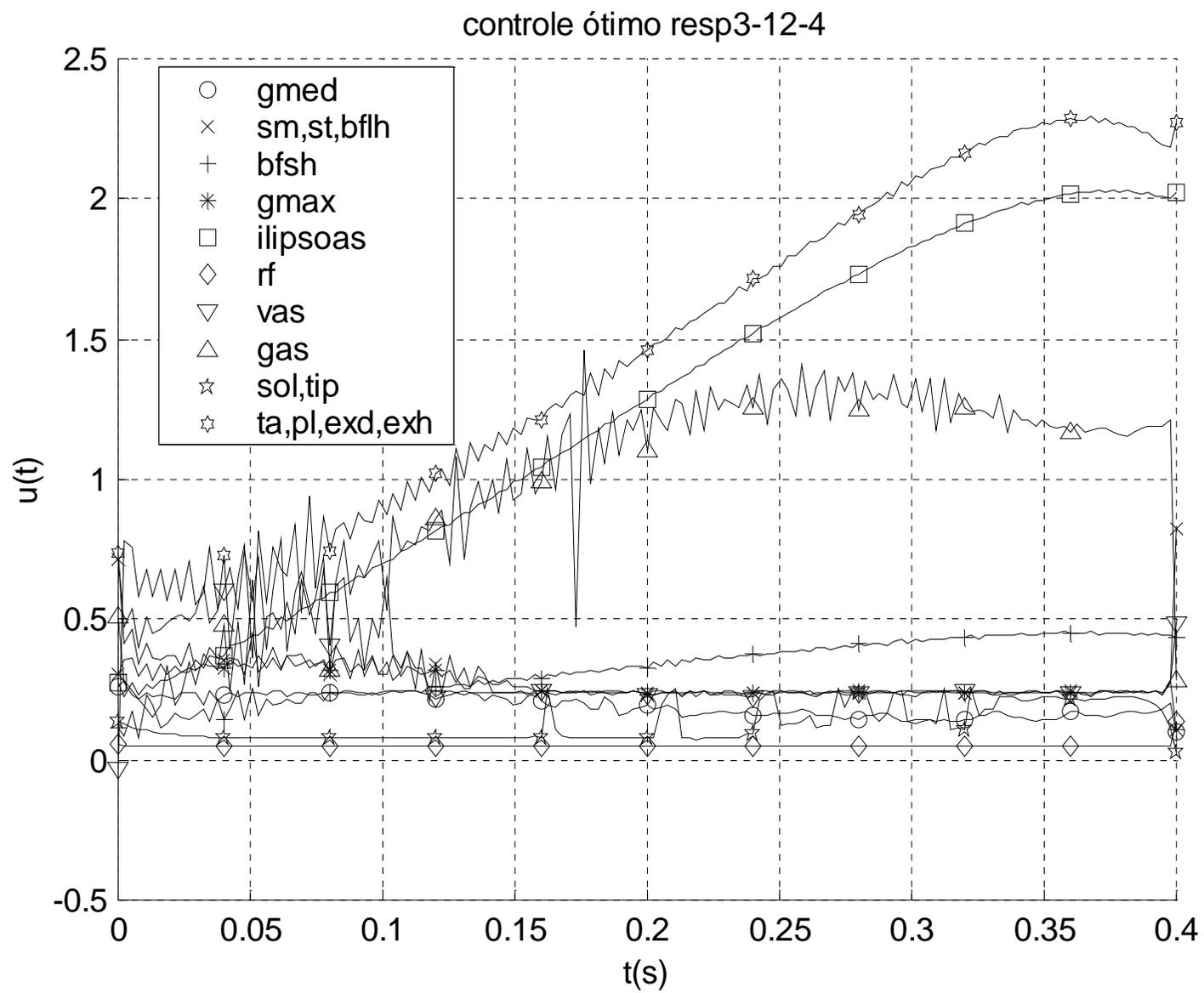


Figura 6.4: Controle 3_12_4

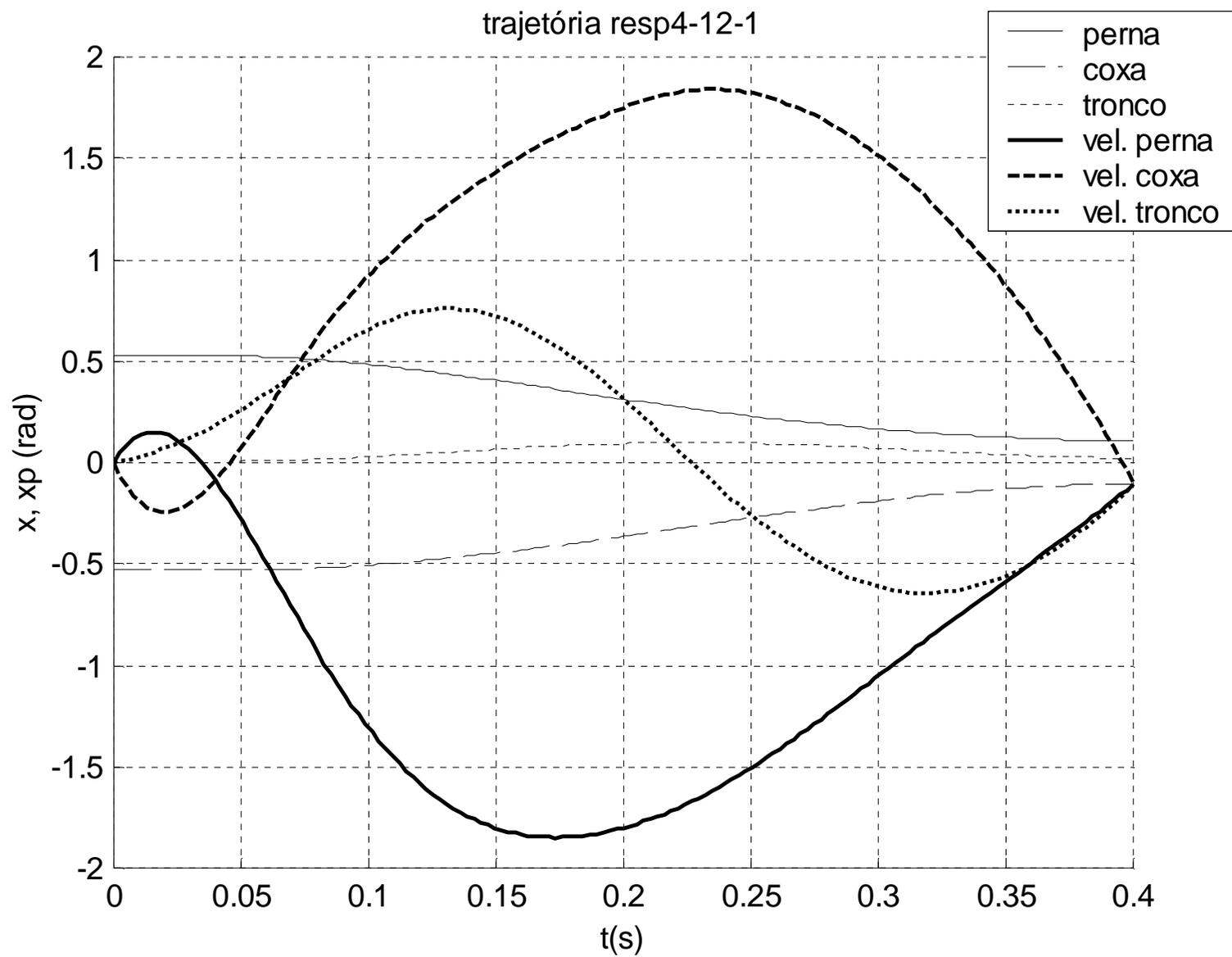


Figura 6.5: Trajetória 4_12_1

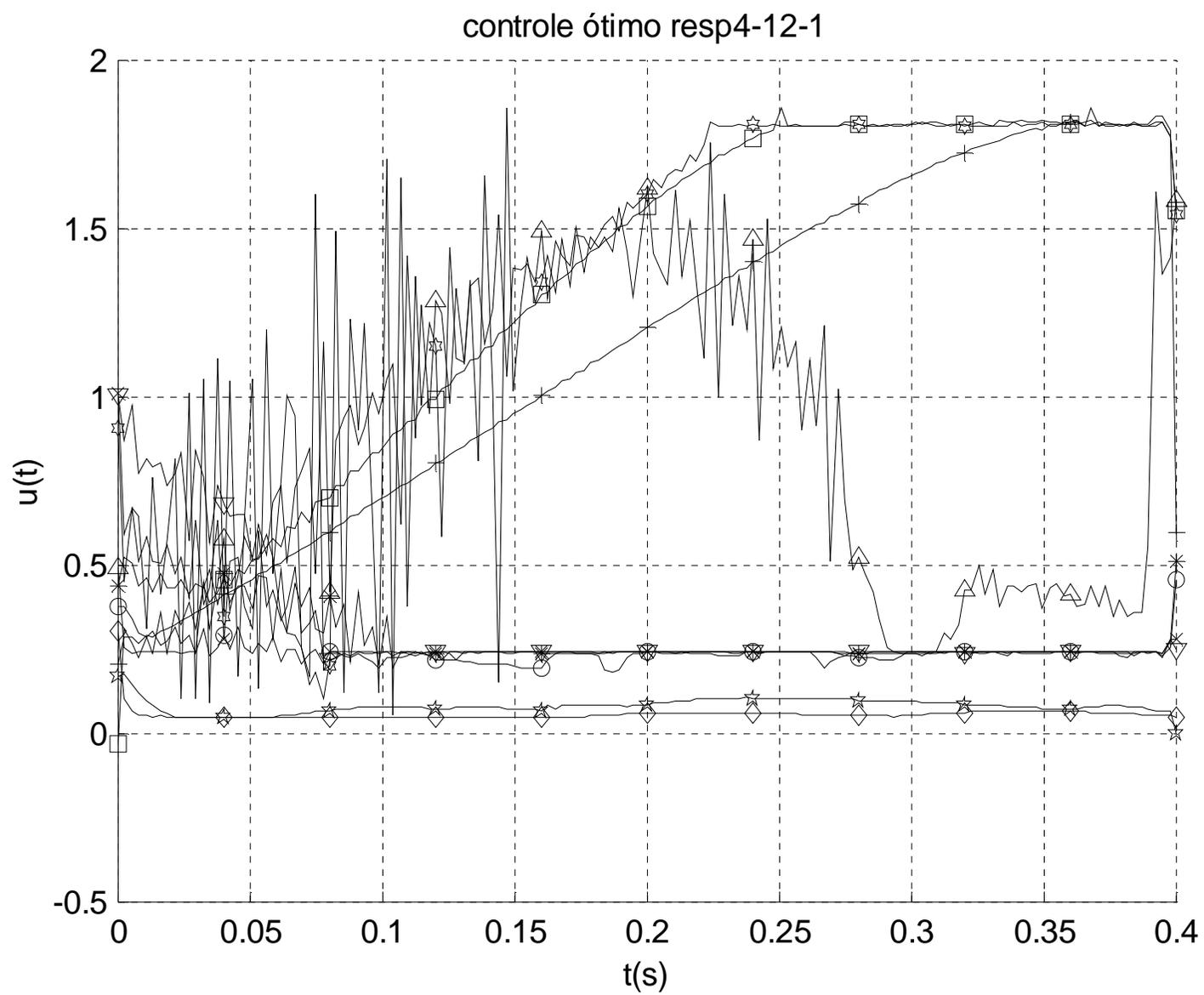


Figura 6.5: Controle 4_12_1

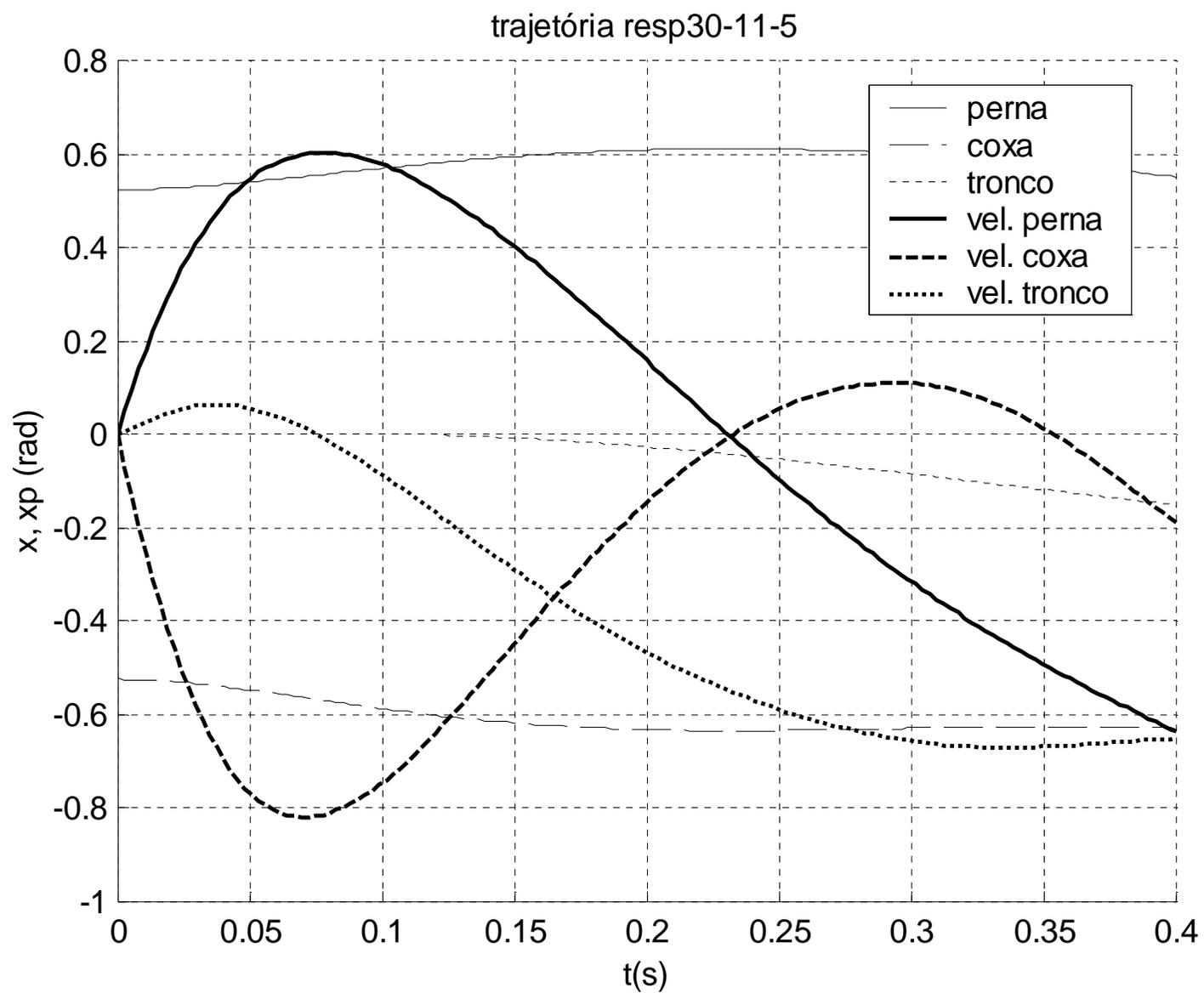


Figura 6.6: Trajetória 30_11_5

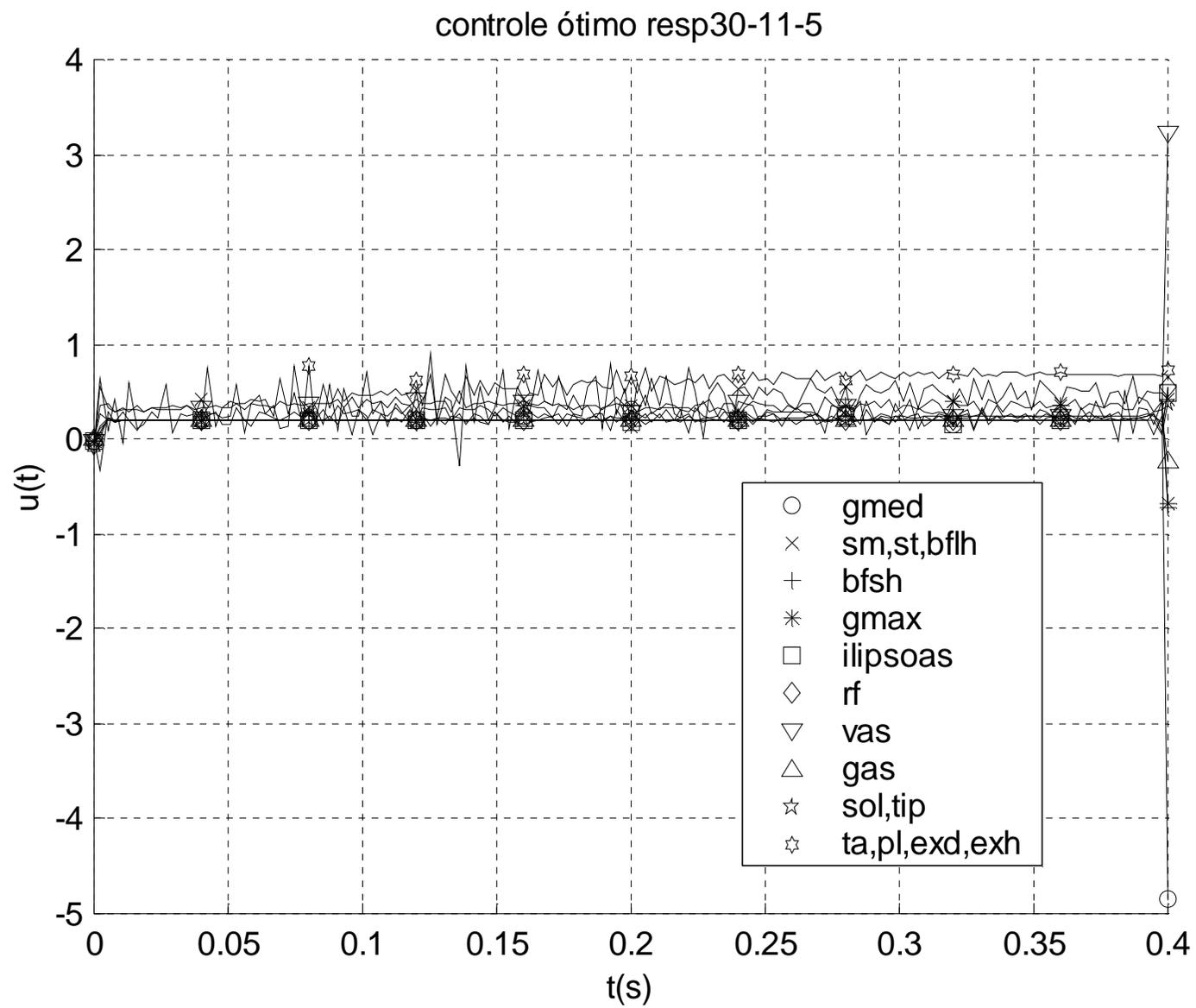


Figura 6.6: Controle 30_11_5

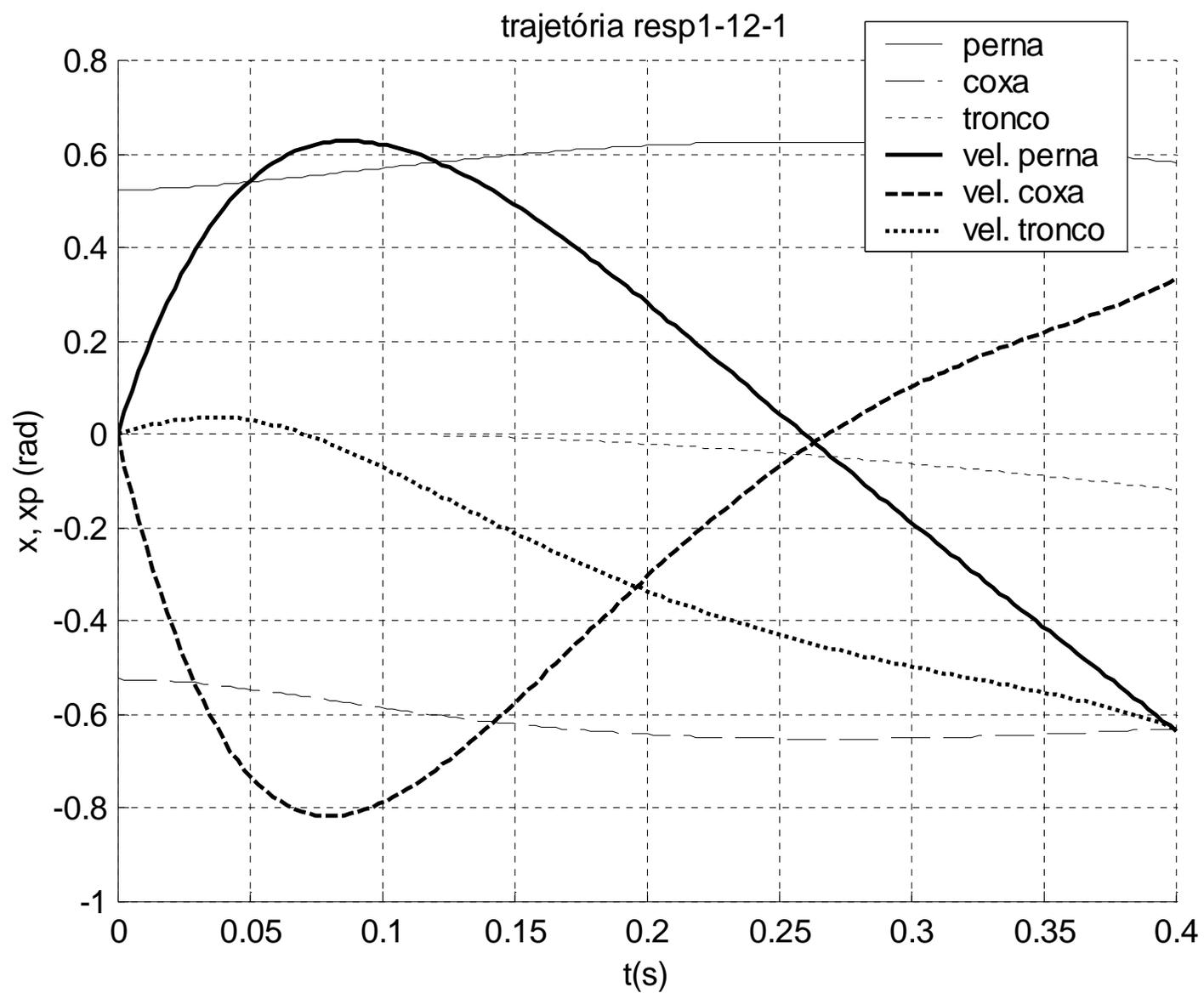


Figura 6.7: Trajetória 1_12_1

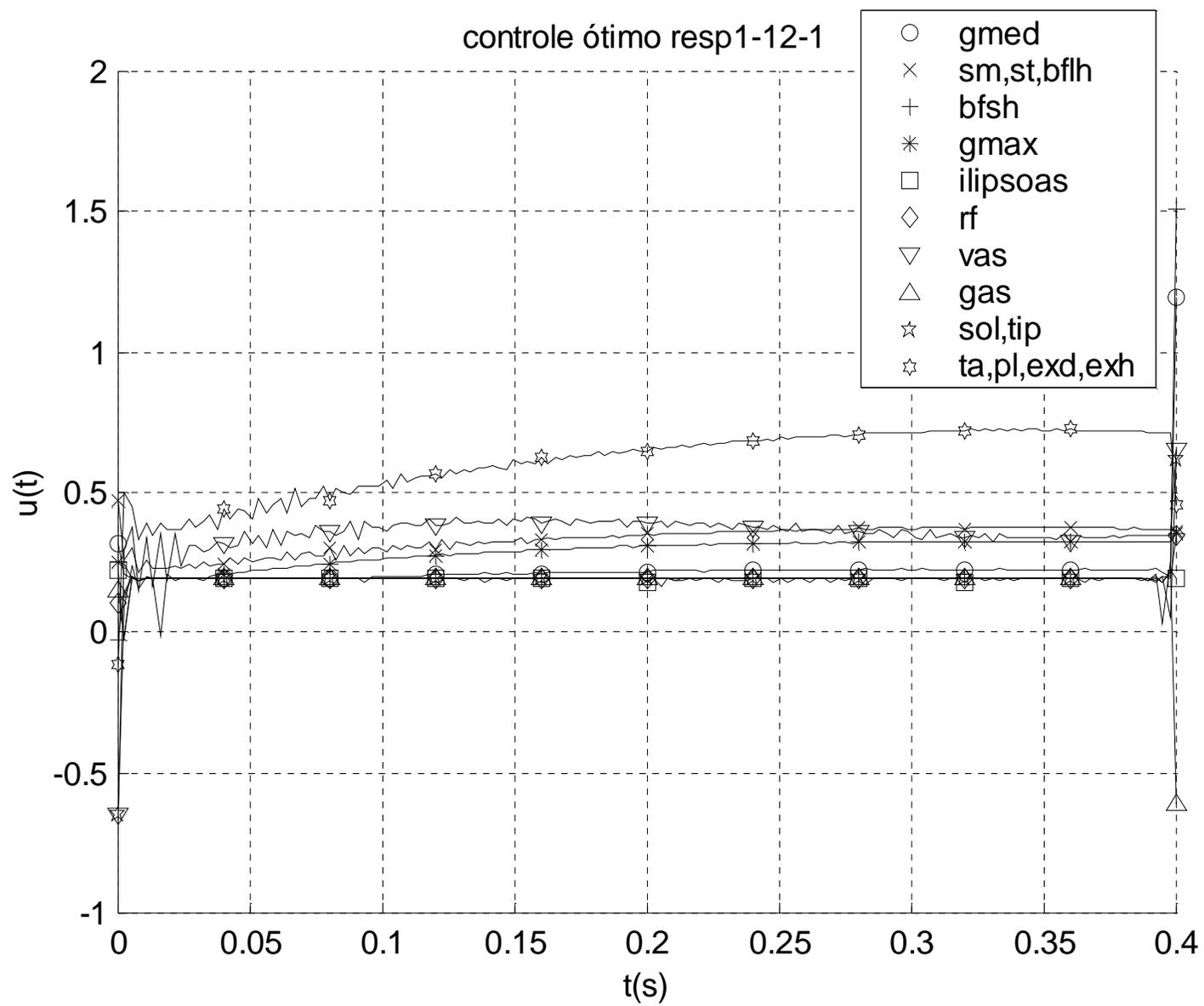


Figura 6.7: Controle 1_12_1

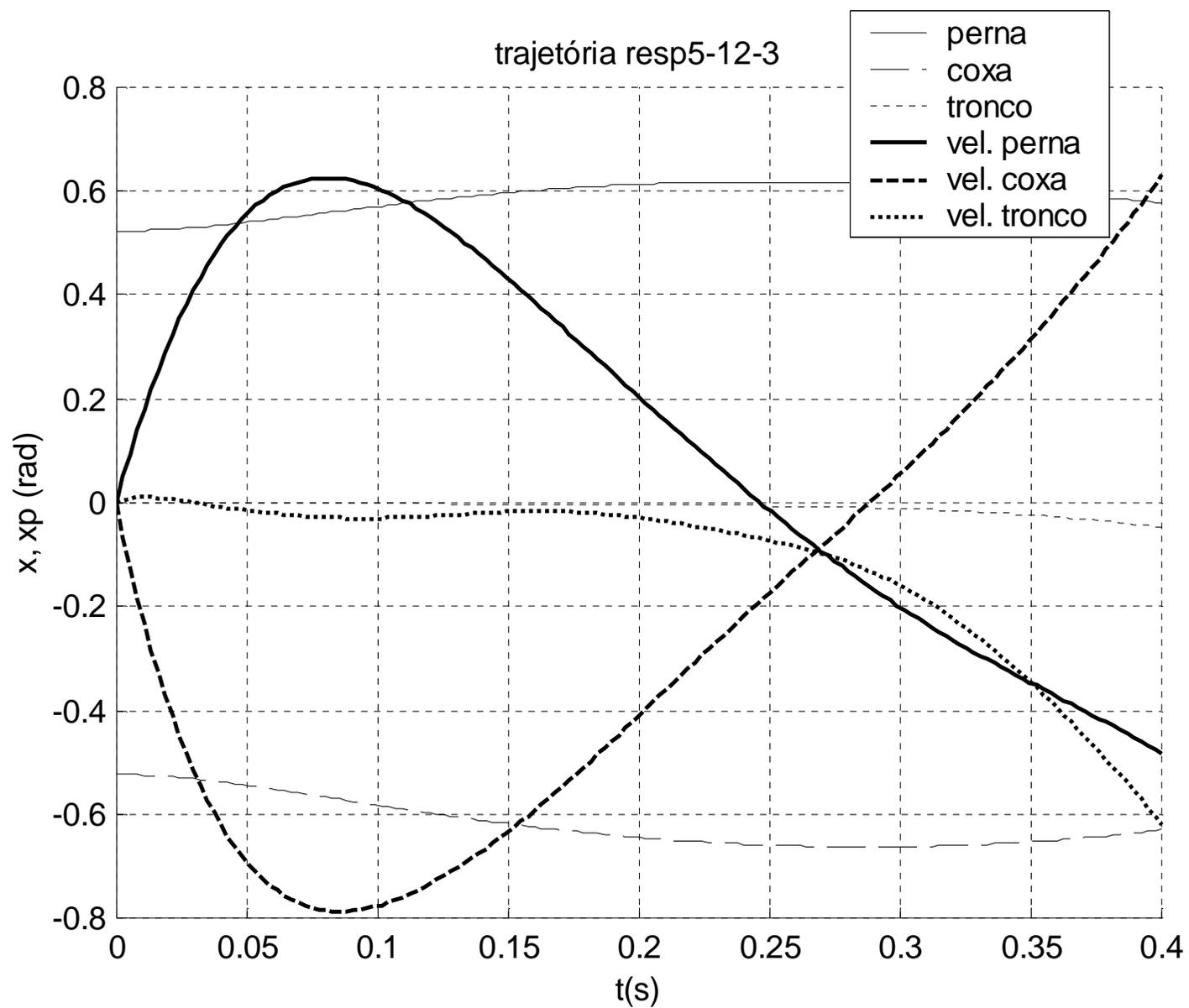


Figura 6.8: Trajetória 5_12_3

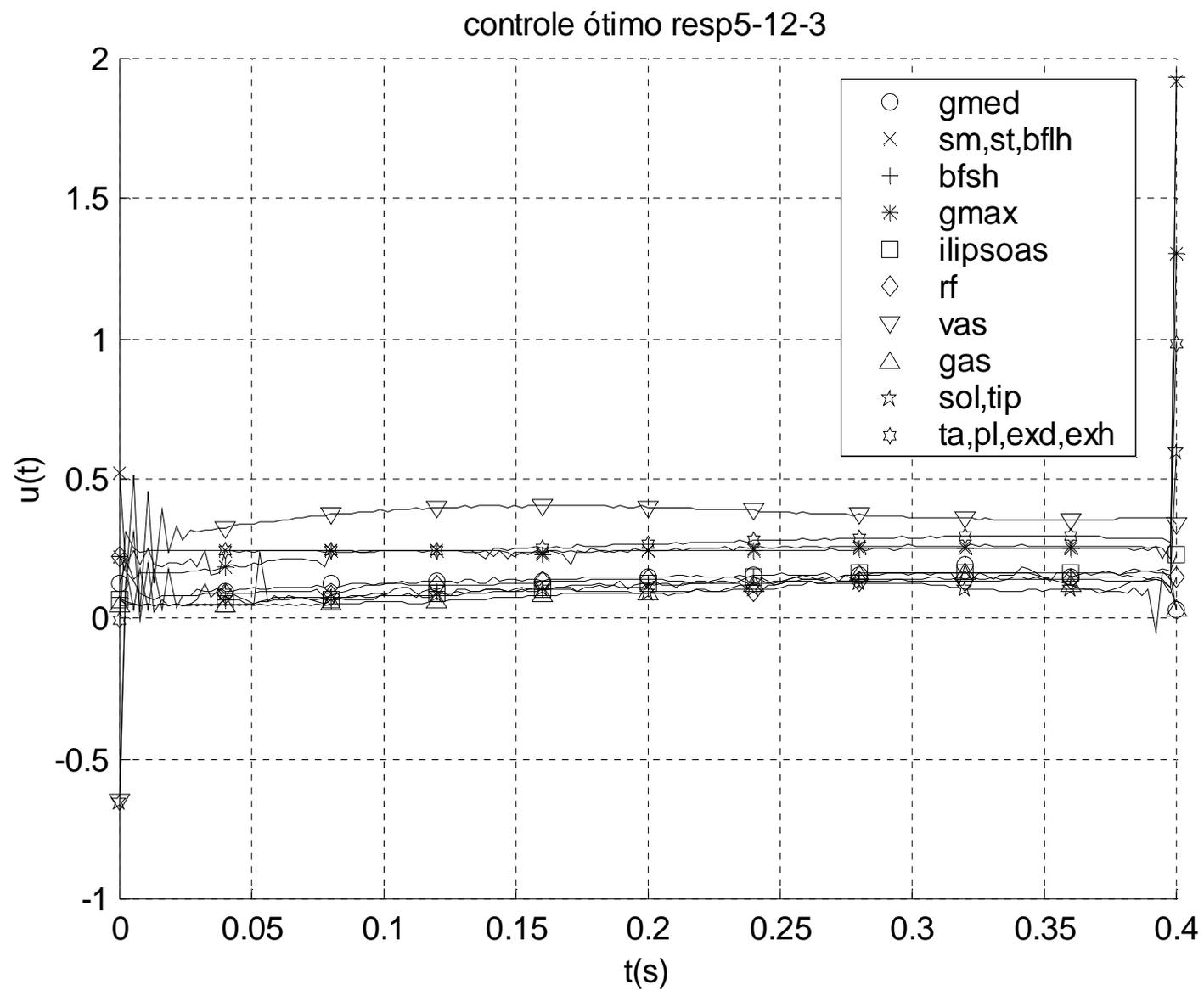


Figura 6.8: Controle 5_12_3

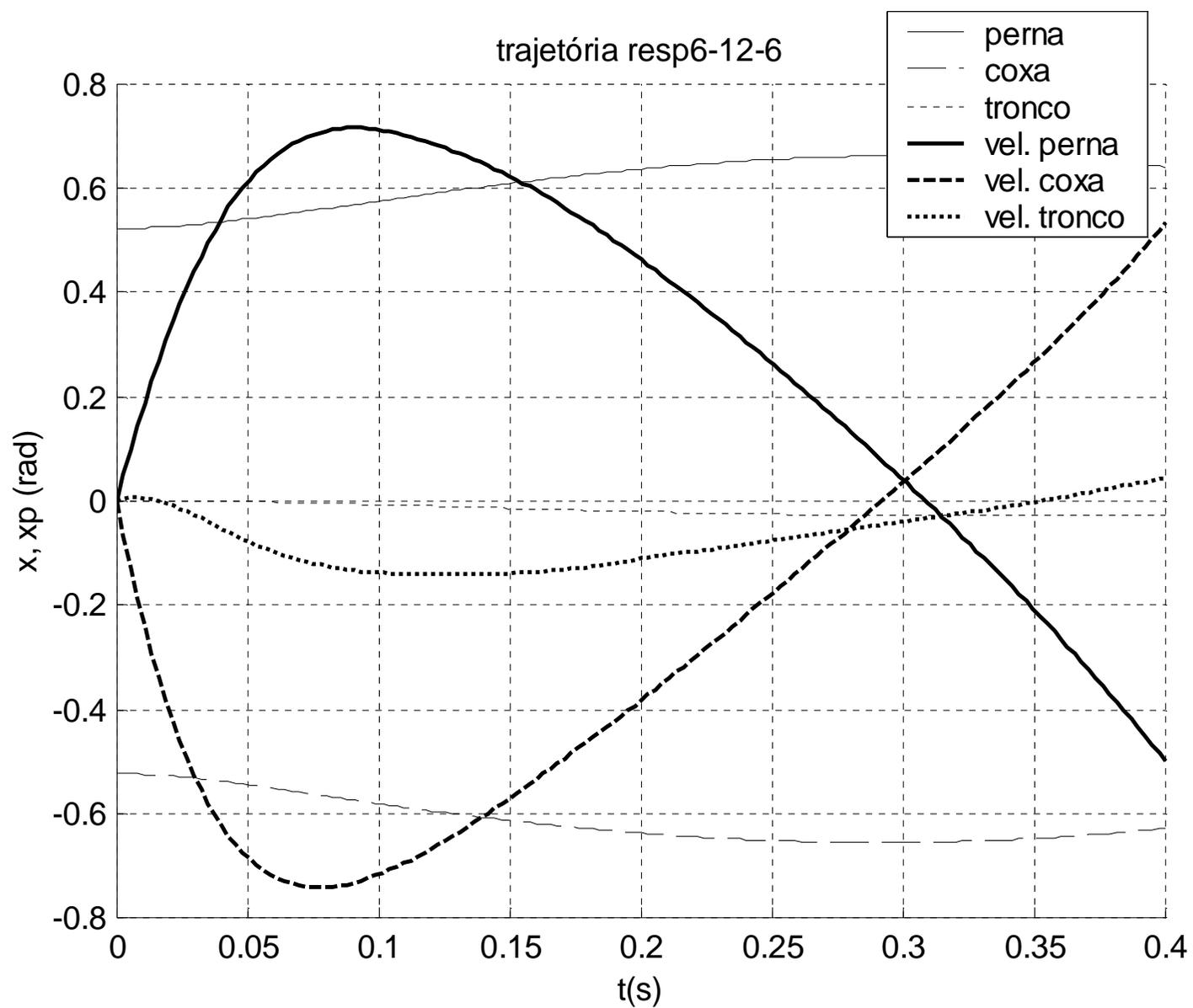


Figura 6.9: Trajetória 6_12_6

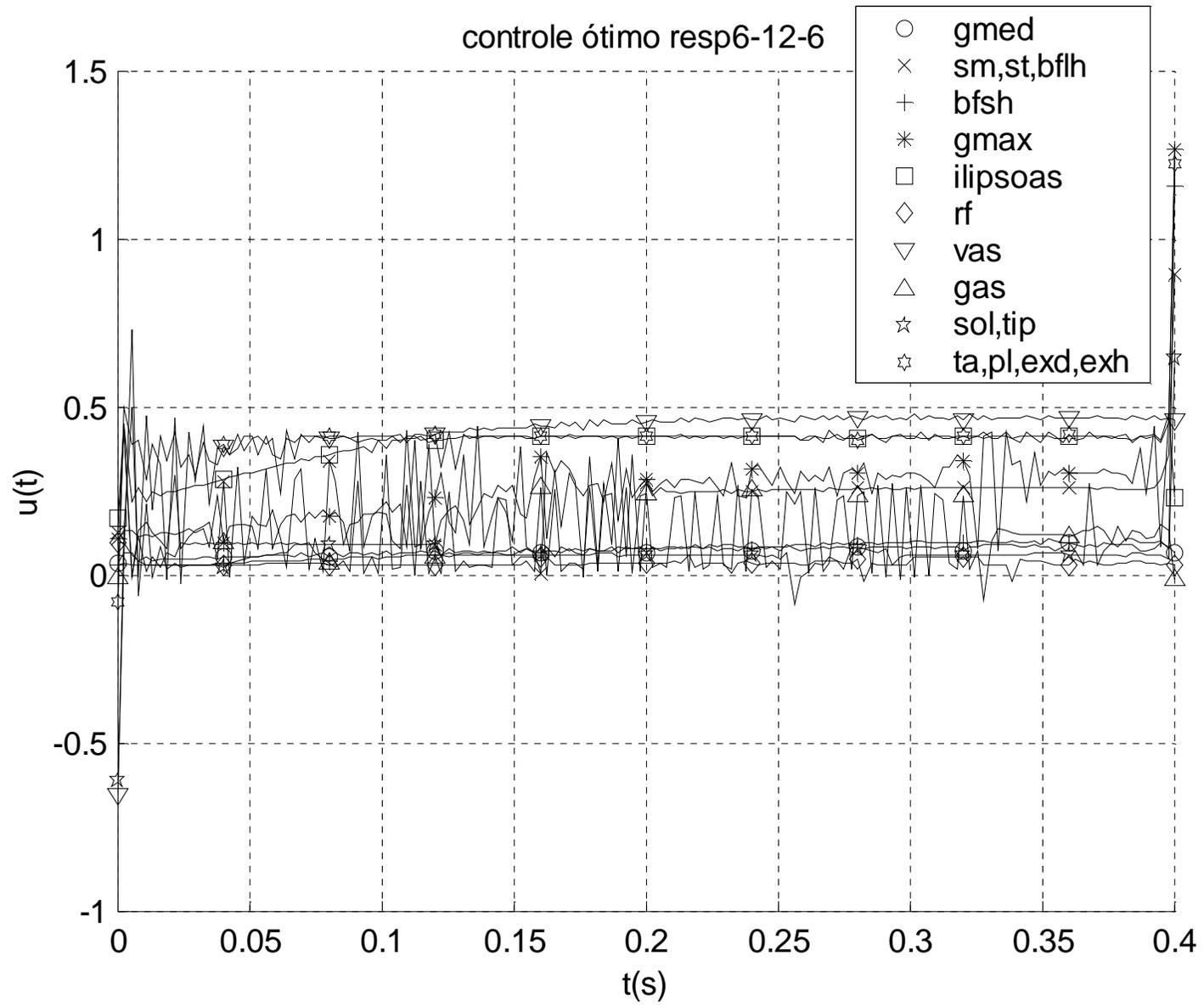


Figura 6.9: Controle 6_12_6

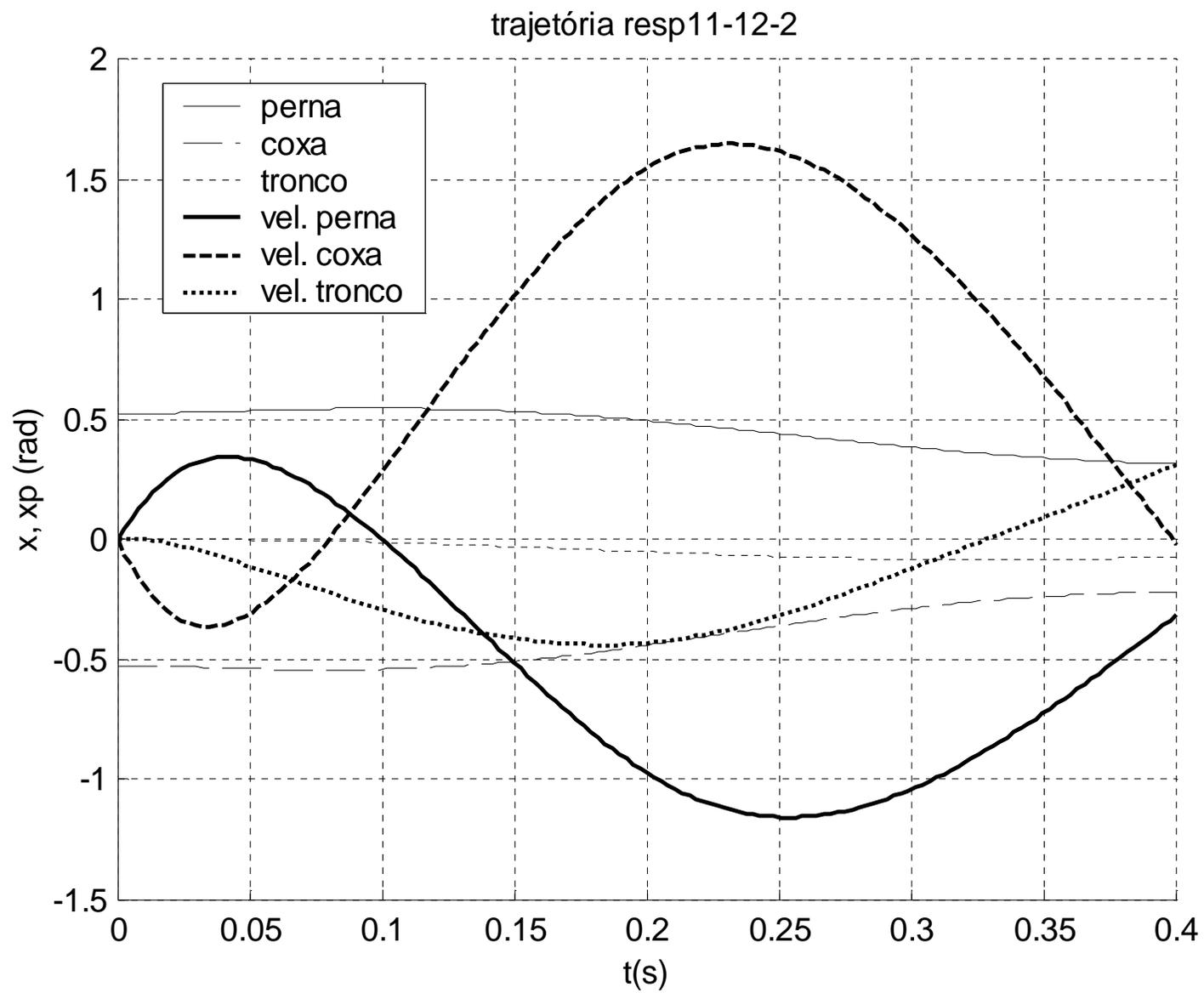


Figura 6.10: Trajetória 11_12_2

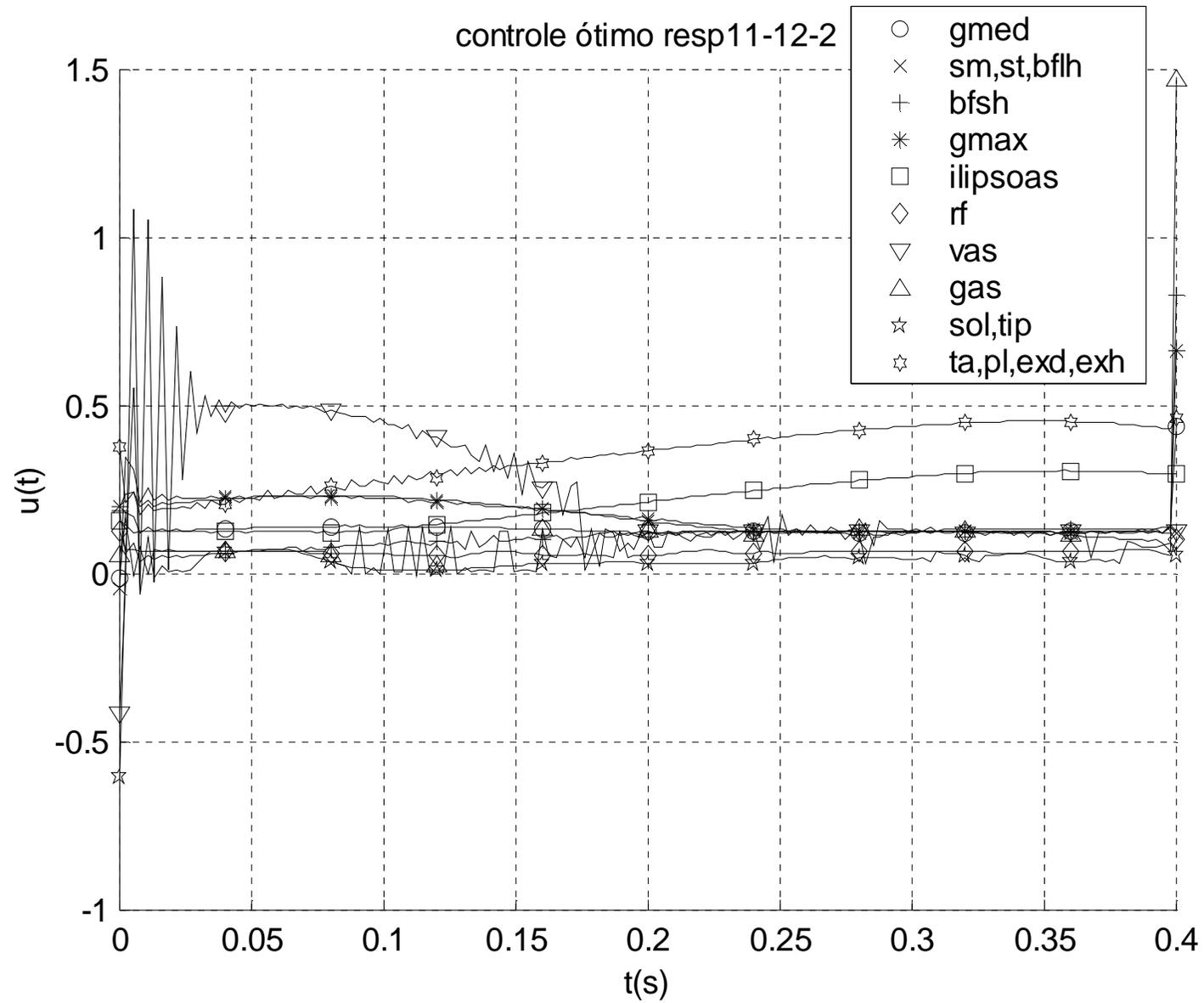


Figura 6.10: Controle 11_12_2

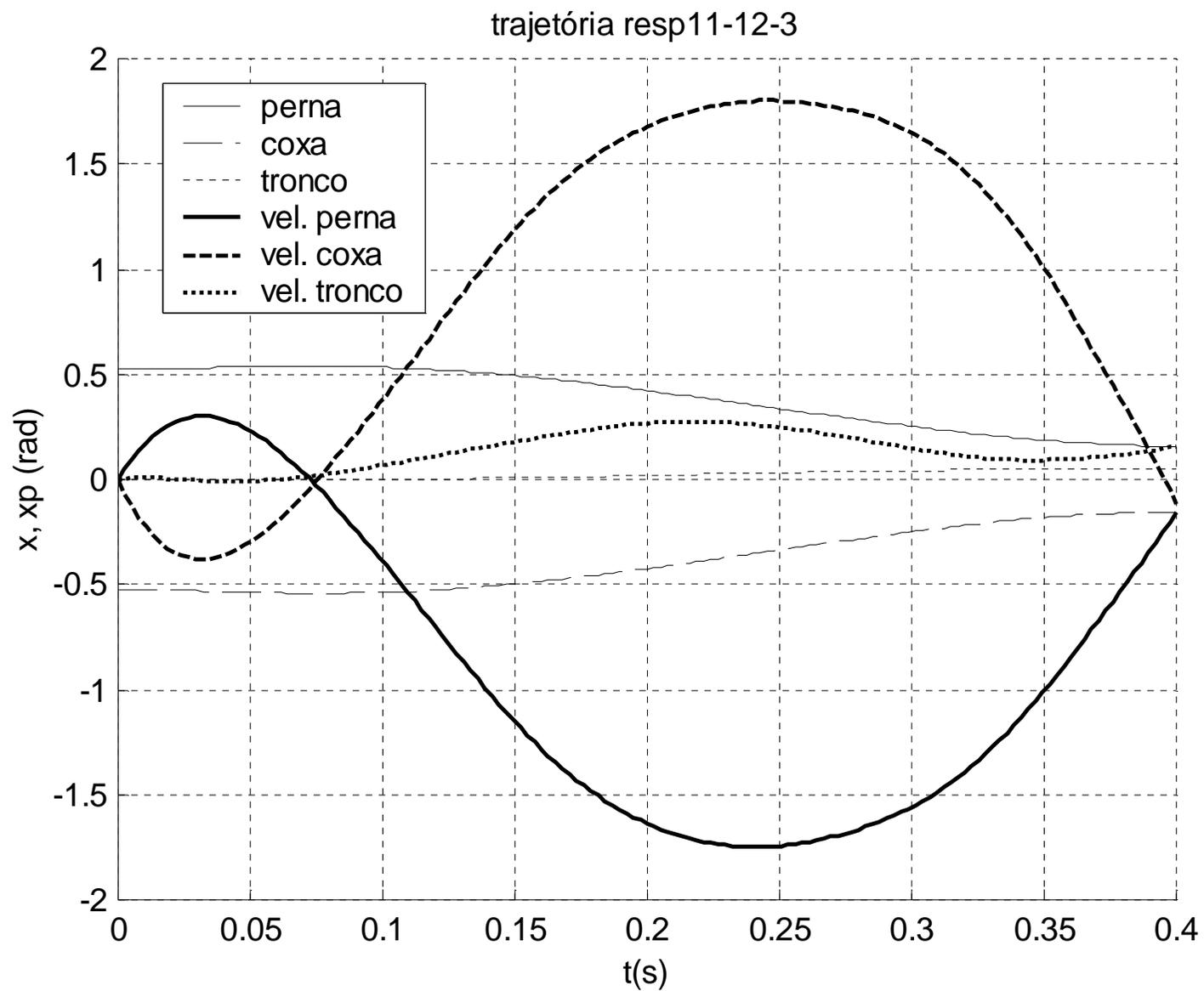


Figura 6.11: Trajetória 11_12_3

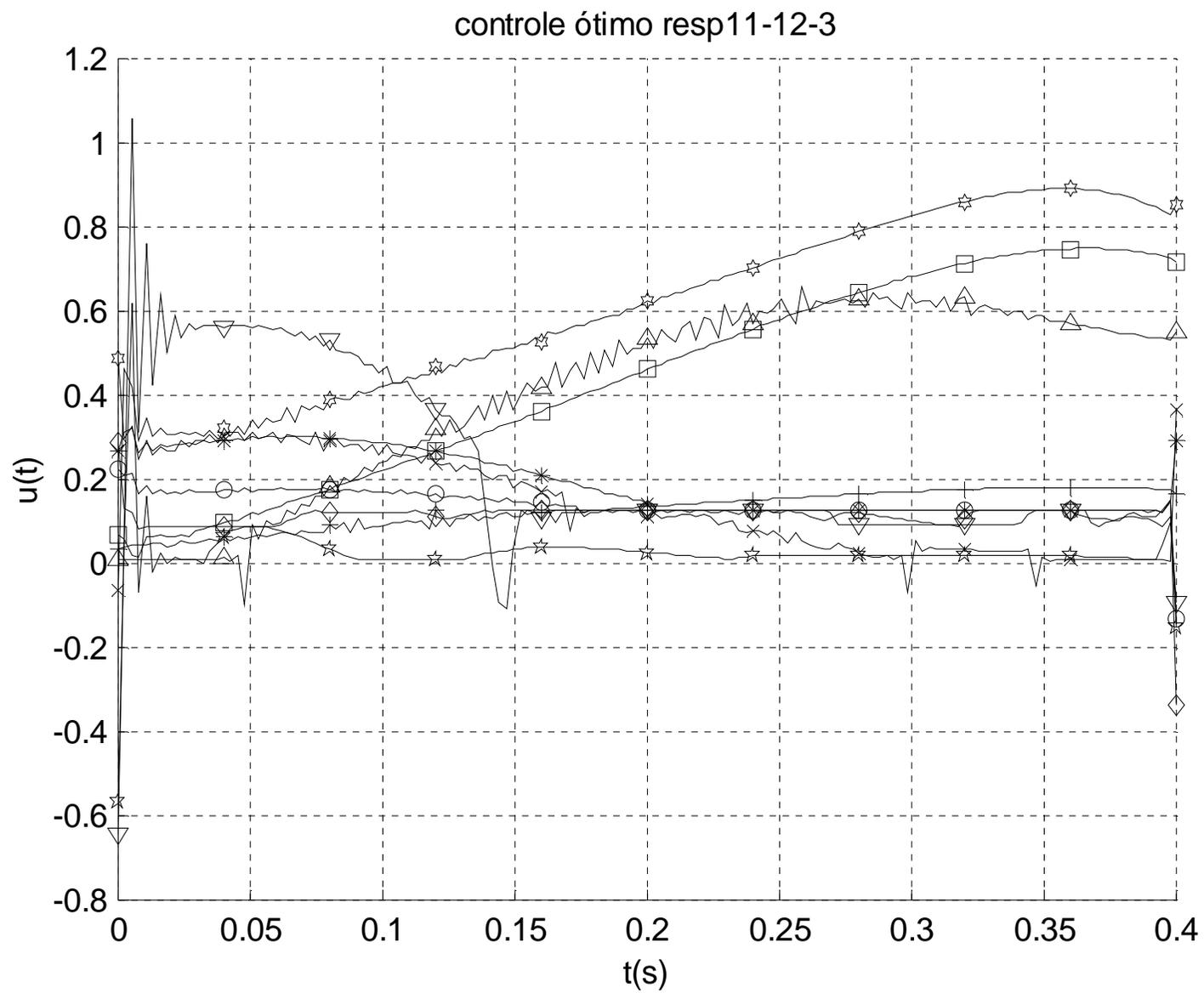


Figura 6.11: Controle 11_12_3

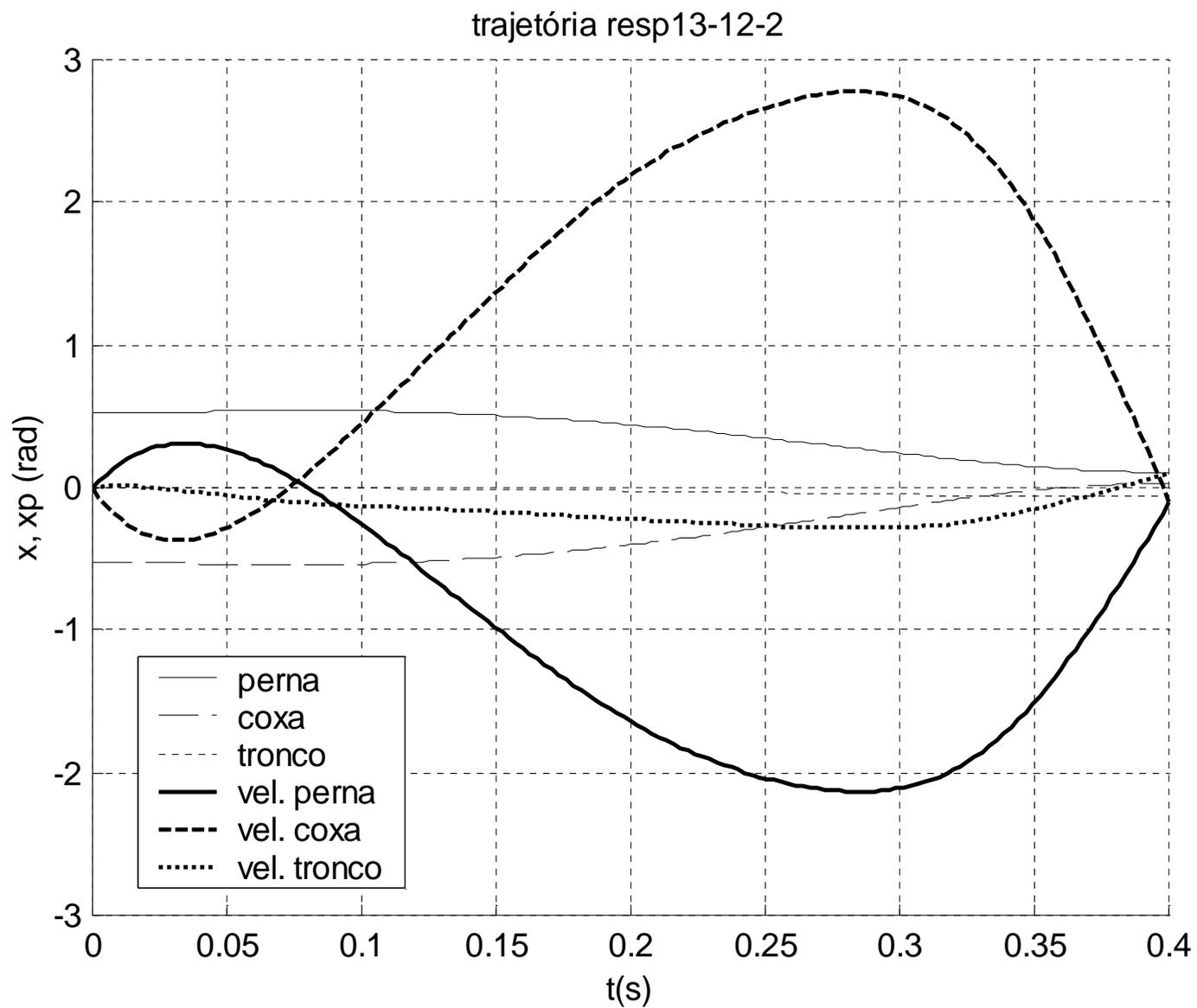


Figura 6.12: Trajetória 13_12_2

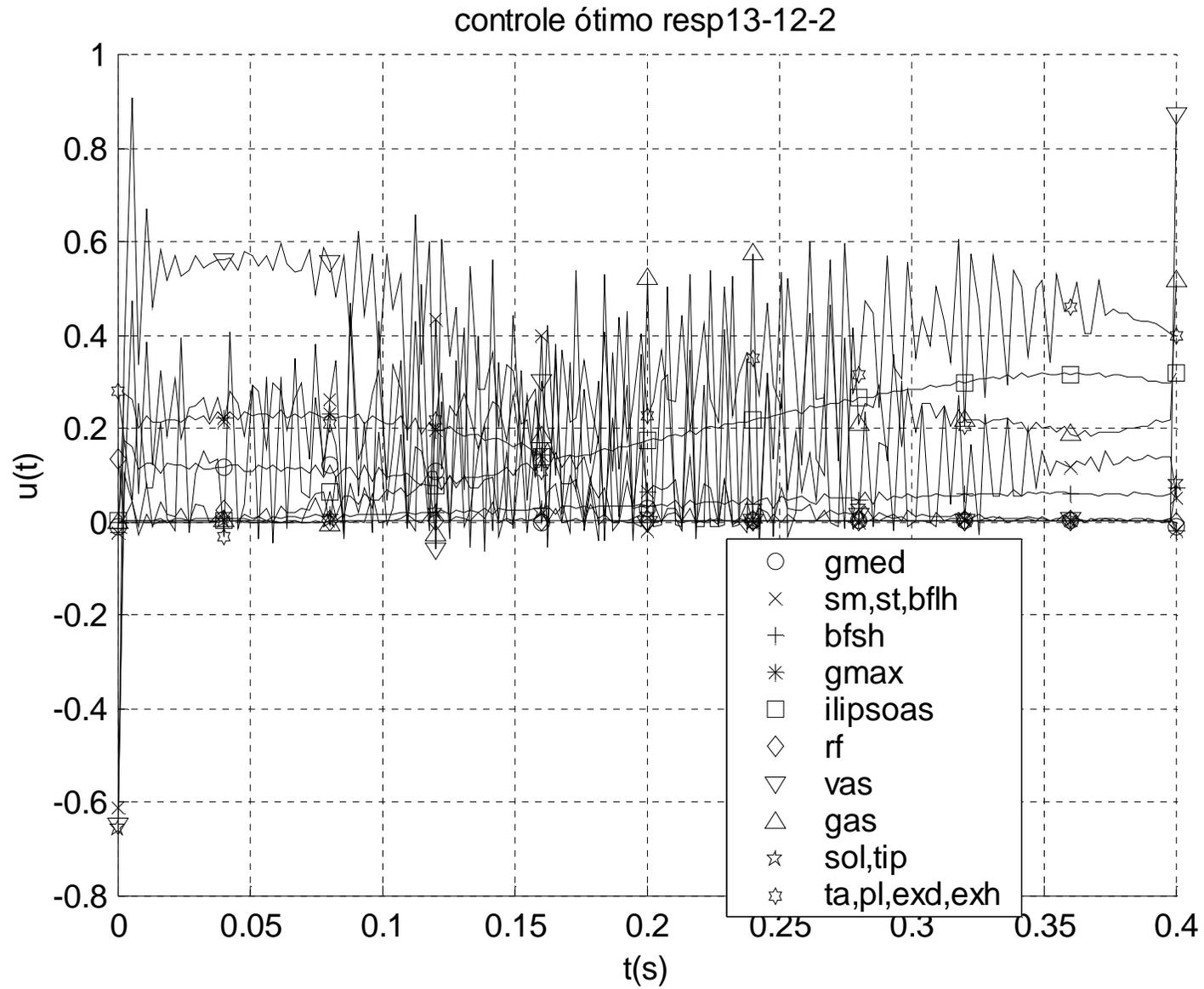


Figura 6.12: Controle 13_12_2

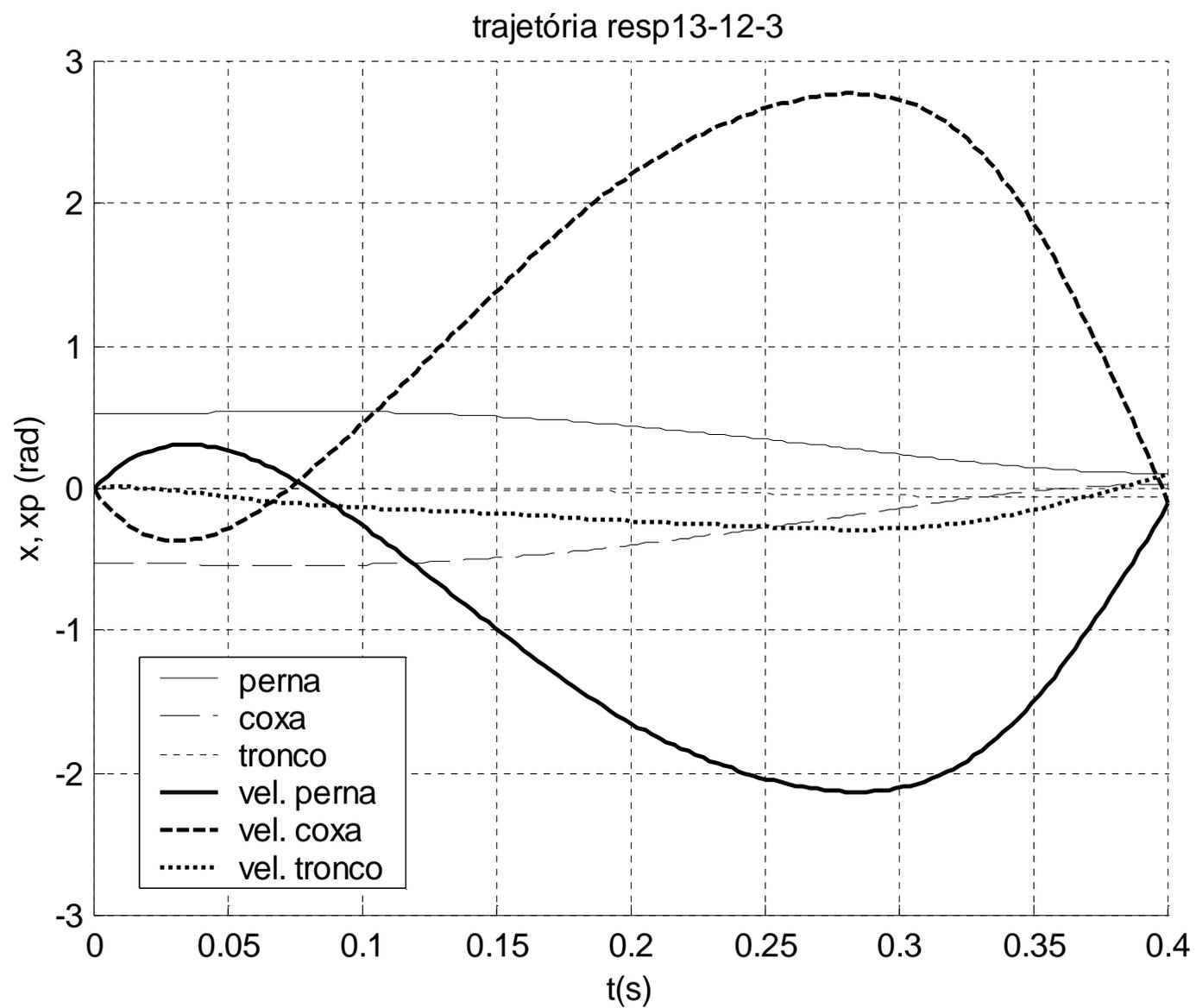


Figura 6.13: Trajetória 13_12_3

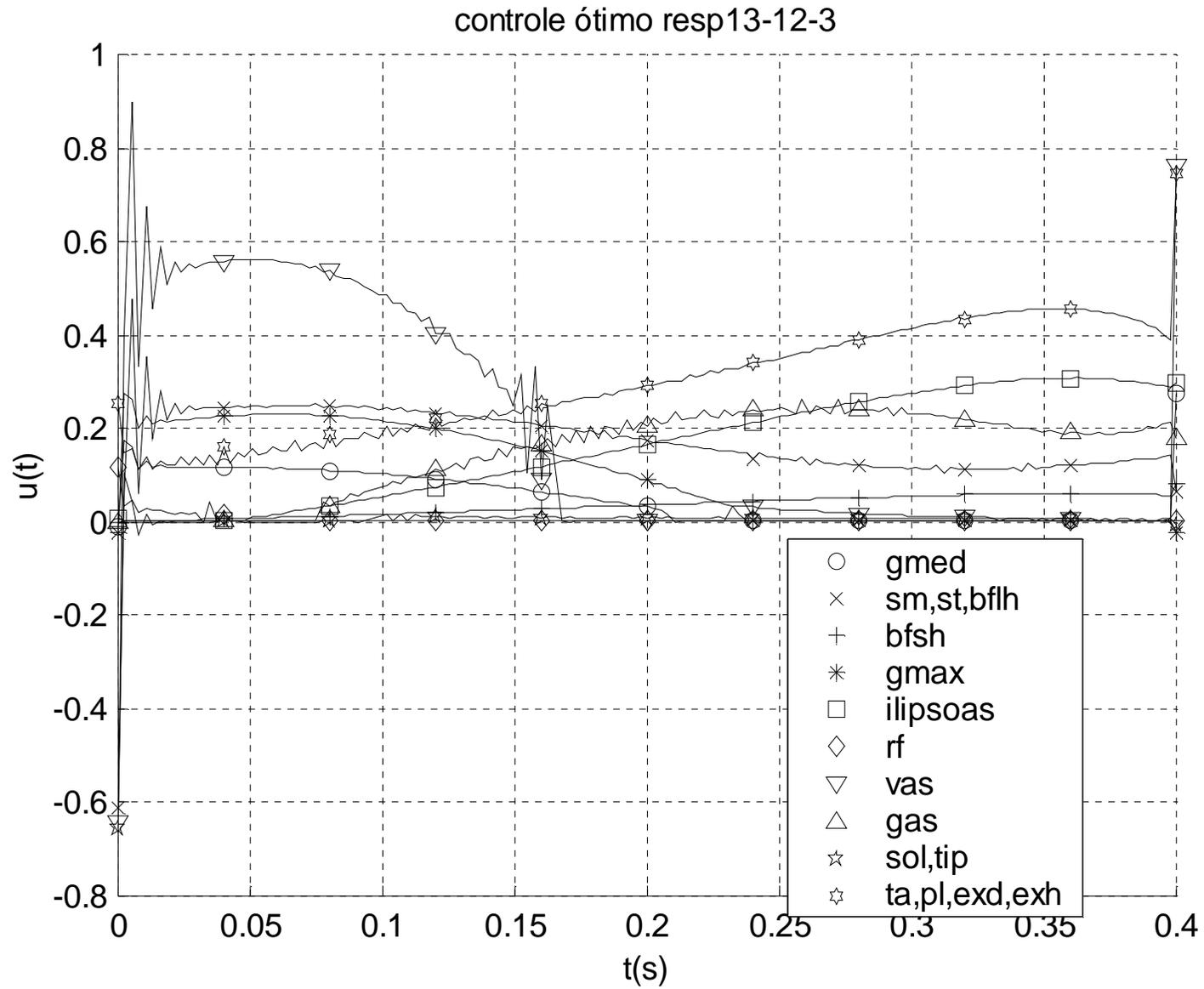


Figura 6.13: Controle 13_12_3

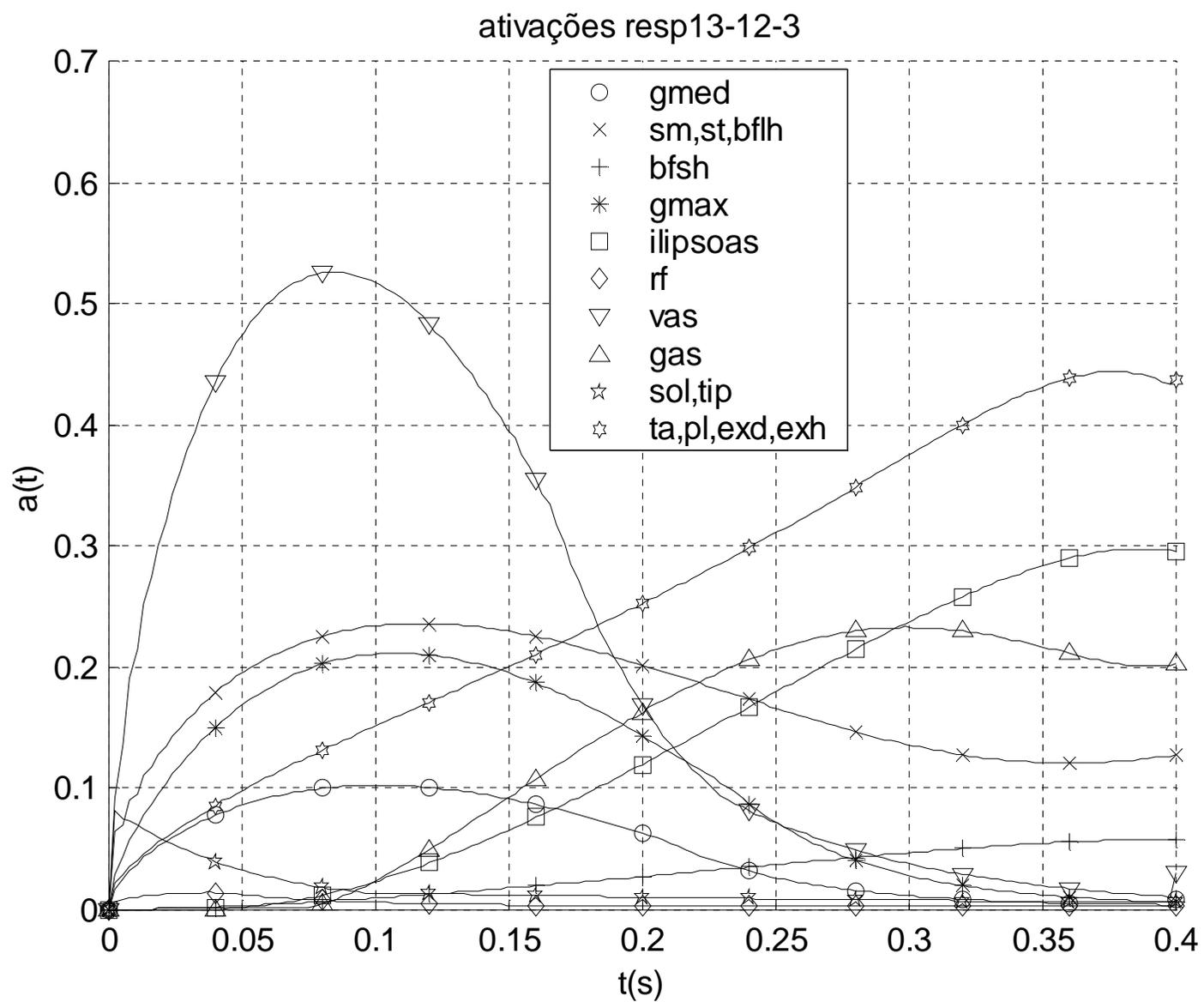


Figura 6.14: Ativações neuromusculares 13_12_3

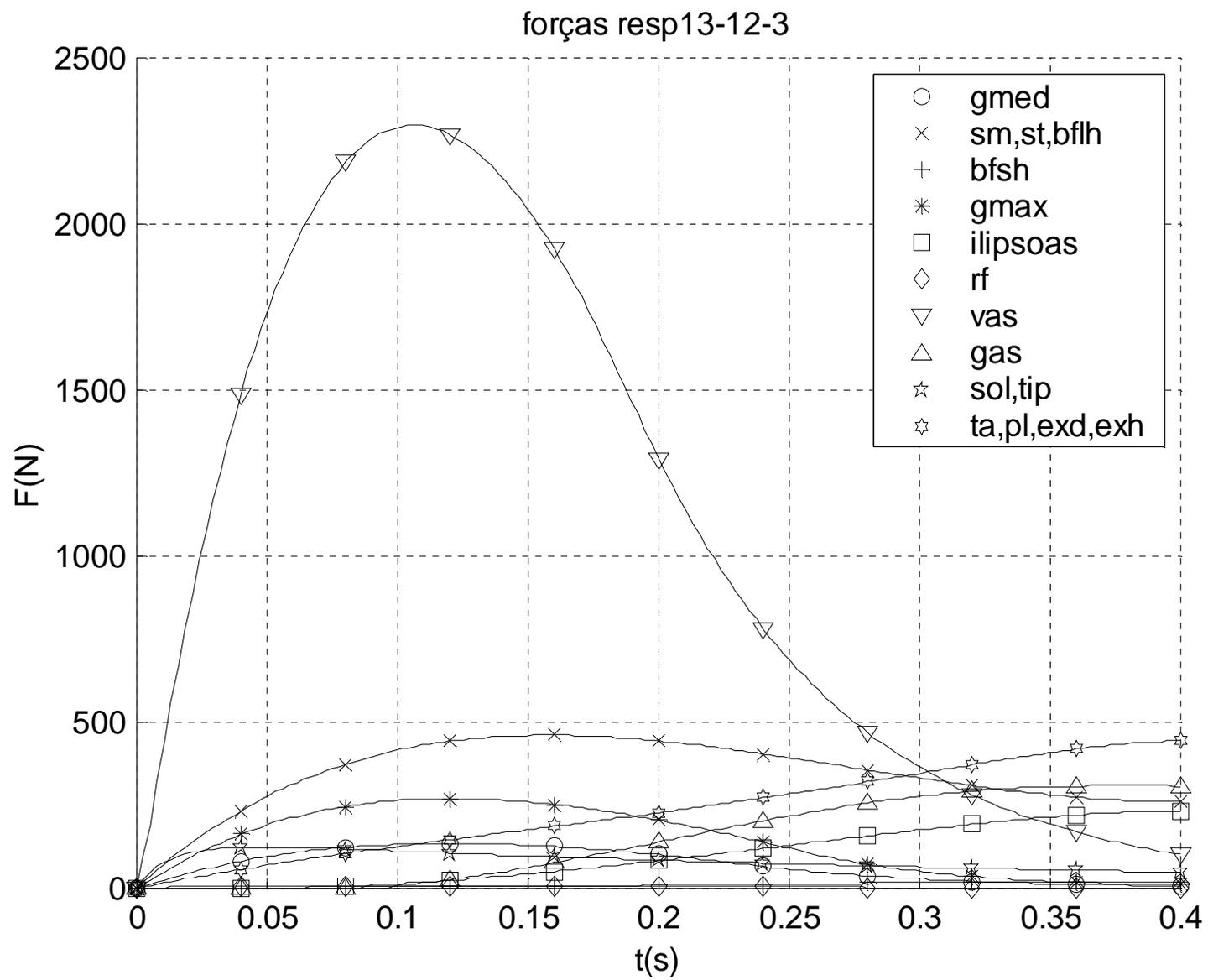


Figura 6.15: Forças musculares 13_12_3

6.2 Resultados com $t_f = 1.0$ segundo, gravidade variável

Aplicando a técnica da aceleração da gravidade com valores reduzidos nas primeiras simulações, foi possível controlar o modelo com braços de momento variáveis num movimento de 1 segundo. A seqüência de simulações, mostrada na Tabela 6.3, que levou a resultados convergentes foi:

8_1_1 (Figura 6.22, Figura 6.23 e Figura 6.24) \Leftarrow 7_1_1 (Figura 6.21) \Leftarrow 6_1_1 (Figura 6.20) \Leftarrow 3_1_2 \Leftarrow 1_1_1 (Figura 6.19) \Leftarrow 30_12_2 \Leftarrow 27_12_1 \Leftarrow 23_12_1 \Leftarrow 14_12_3 (Figura 6.18) \Leftarrow 14_12_1 (Figura 6.17) \Leftarrow u00 (Figura 6.16)

Nesta série de simulações, procurou-se evitar o uso de correções através de heurísticas das ativações musculares quando a gravidade era alterada. Deste modo, a metodologia seria menos suscetível ao nível de domínio do comportamento biomecânico do modelo por parte do usuário. Entretanto, foi necessária uma pequena correção da ativação dos *vasti* na simulação 1_1_1, quando a aceleração da gravidade atingiu o valor natural. Foram assim utilizadas mais etapas no aumento da gravidade, isto é, de $2 \rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow 9 \rightarrow 9.81 \text{ m/s}^2$. O valor final da função de custo ficou bastante mais baixo do que todas as simulações anteriores ($f = 1.8382$), sugerindo que o tempo de 1 segundo é bastante mais próximo do tempo ótimo para realização da manobra e que as funções que definem naturalmente os braços de momento dos músculos concorrem para a economia da função de custo adotada. As ativações e forças musculares obtidas para a simulação 8_1_1 estão mostradas na Figura 6.23 e Figura 6.24, respectivamente.

As tolerâncias de otimização e violação dos vínculos empregadas na maior parte dos testes foram de 0.05 e 0.0001, respectivamente, ao passo que o valor inferior do vínculo de controle ficou fixado em 0.001. Com isso, obteve-se uma tolerância bastante estreita para a violação de vínculos, sem que o algoritmo apresentasse erros de ponto flutuante no cálculo dos gradientes (ver o tratamento de tolerâncias no manual do NPSOL,

Gill et al., 1998). Uma parte do ruído numérico poderia ser atribuída à utilização, em algumas simulações intermediárias, de resultados que não atingiram as tolerâncias especificadas (vide Tabela 6.3), mas apenas o número máximo especificado de 400 iterações. Nas últimas simulações, uma vez atingidos os vínculos terminais de $\pi/30$, o ruído numérico foi atenuado com a redução da tolerância de otimização, não tanto de 0.05 para 0.005 quanto de 0.005 para 0.001. As simulações com gravidade variável foram realizadas apenas com vínculos terminais largos de $\pi/5$, reduzido apenas quando a gravidade atingiu o valor total. Economiza-se assim tempo de processamento, uma vez que a resposta se degrada sensivelmente quando a gravidade é alterada. O tempo total de CPU para a realização de toda a seqüência atingiu a considerável marca de 419.3 horas, ou 17.5 dias. Cada simulação nessas condições ocupava ao redor de 490 MB de memória RAM.

simul.	ci ⁱ	v. term. ⁱⁱ	tol. otm.	tol. cont.	U _{inf}	U _{sup}	RK ⁱⁱⁱ	N	tf	g	término ^{iv}	n° its.	min.cpu ^v	f	fig. n ^a	obs.
14_12_1	u00	$\pi/5$	0.001	0.0001	0.001	1	4	400	1	2	1	64	574	0.0275	Figura 6.17	obs. 1
14_12_3	14_12_1	$\pi/5$	0.001	0.0001	0.001	1	4	400	1	4	1	108	1734	0.1420	Figura 6.18	
16_12_1	14_12_3	$\pi/5$	0.001	0.0001	0.001	1	4	400	1	6	8		1 seman.			
23_12_1	14_12_3	$\pi/5$	0.05	0.0001	0.001	1	4	400	1	6	6	400	70.5 h	0.3859		
27_12_1	23_12_1	$\pi/5$	0.05	0.0001	0.001	1	4	400	1	8	6	400	74.1h	0.8135		
30_12_1	27_12_1	$\pi/5$	0.05	0.0001	0.001	1	4	400	1	9.81	2	4	284	0.8976		
30_12_1	27_12_1	$\pi/5$	0.05	0.0001	0.001	1	4	400	1	9	1	389	44.7h	1.0335		
1_1_1	30_12_1	$\pi/5$	0.05	0.0001	0.001	1	4	400	1	9.81	6	400	32.2h	1.2712	Figura 6.19	obs. 2
3_1_2	1_1_1	$\pi/15$	0.05	0.0001	0.001	1	4	400	1	9.81	1	219	48.7h	1.7151		
6_1_1	3_1_2	$\pi/30$	0.05	0.0001	0.001	1	4	400	1	9.81	1	185	32.7h	1.8380	Figura 6.20	
7_1_1	6_1_1	$\pi/30$	0.005	0.0001	0.001	1	4	400	1	9.81	1	102	12.2h	1.8376	Figura 6.21	
8_1_1	7_1_1	$\pi/30$	0.001	0.0001	0.001	1	4	400	1	9.81	1	1123	105.7h	1.8382	Figura 6.22	

Tabela 6.3: Resultados com braço de momento variável, 1 segundo de simulação, gravidade variável.

Obs. 1: u00 mostrado na Figura 6.16

Obs. 2: empregado fator de 1.025 no músculo 7 (vasti)

ⁱ u00=ones(size(u00));

u00=0.005*[3*u00(1,:)

3*u00(2,:)

3*u00(3,:)

3*u00(4,:)

5*u00(5,:)

4*u00(6,:)

1*u00(7,:)

3*u00(8,:)

10*u00(9,:)

3*u00(10,:)];

ⁱⁱ Vínculo de desigualdade $x_f^2 \leq v.term^2$.ⁱⁱⁱ RK5=LOSODA^{iv}

1:término normal

2:não foi encontrado ponto factível para os vínculos não lineares

3:falha na busca linear

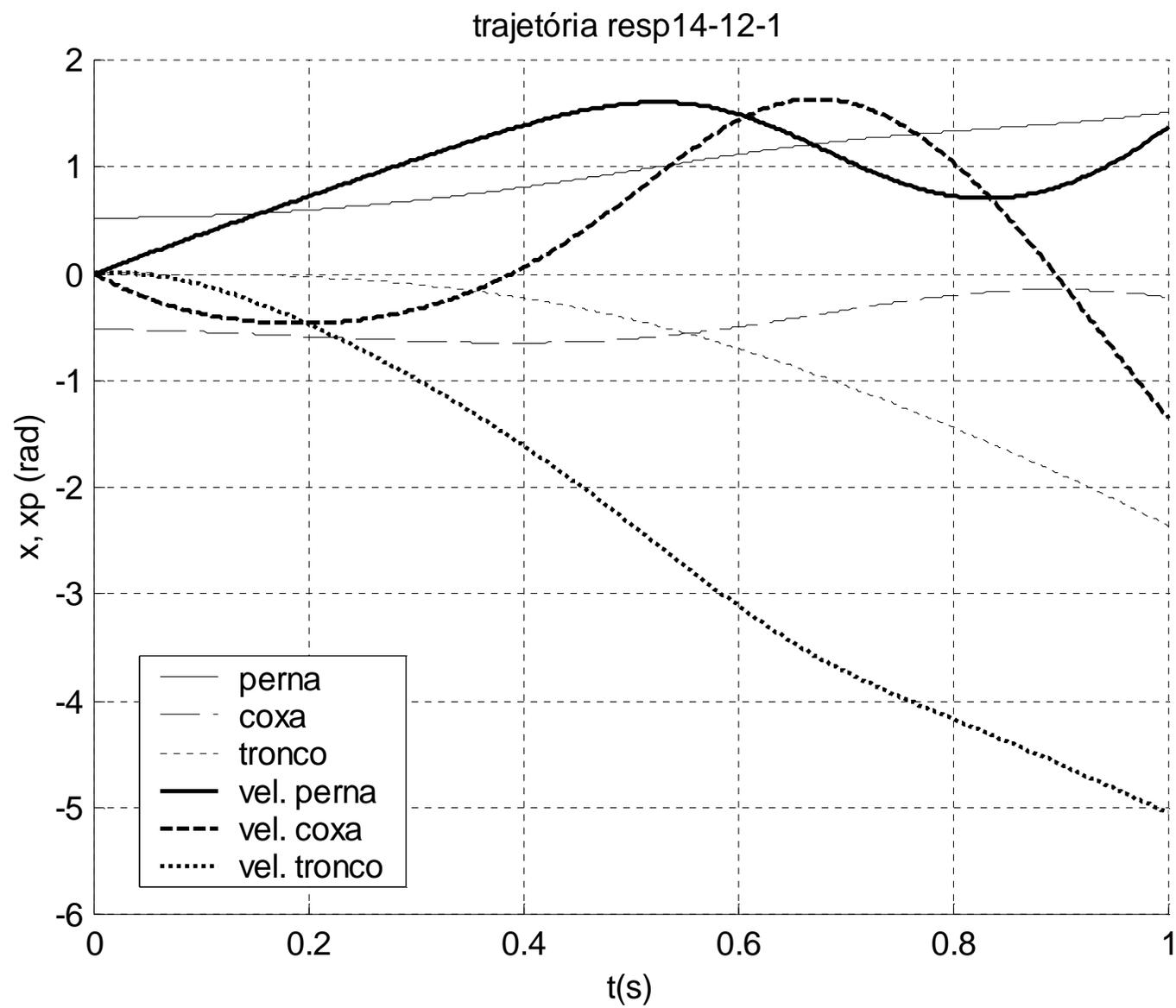
4:não é possível calcular o gradiente dos vínculos

5:erro de memória

6:completado número máximo de iterações

7:erro de ponto flutuante durante cálculo do gradiente do vínculo

^v Pentium III 600Mhz.

Figura 6.16: Trajetória com controle inicial u_{00}

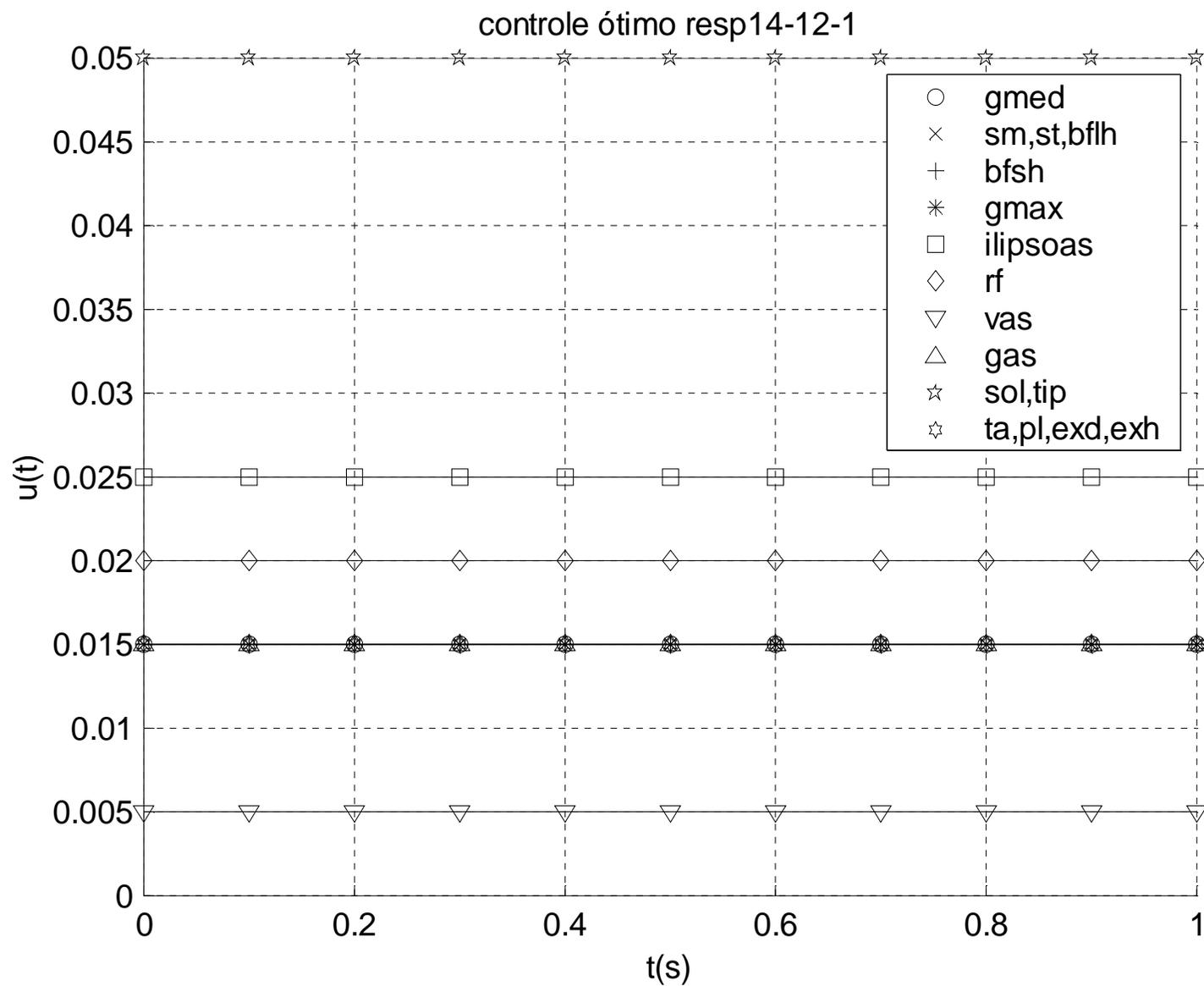


Figura 6.16: Controle inicial u_{00}

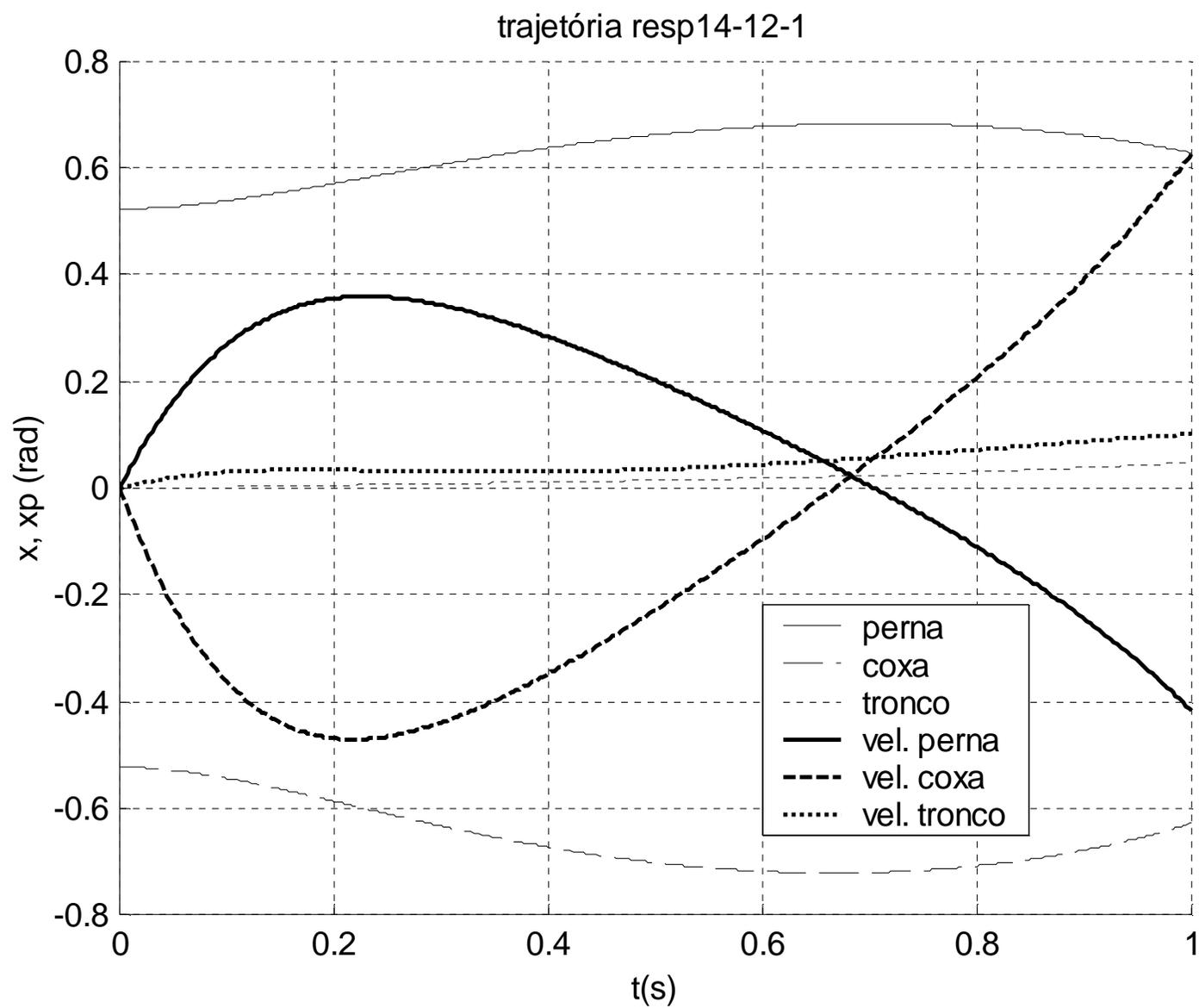


Figura 6.17: Trajetória 14_12_1

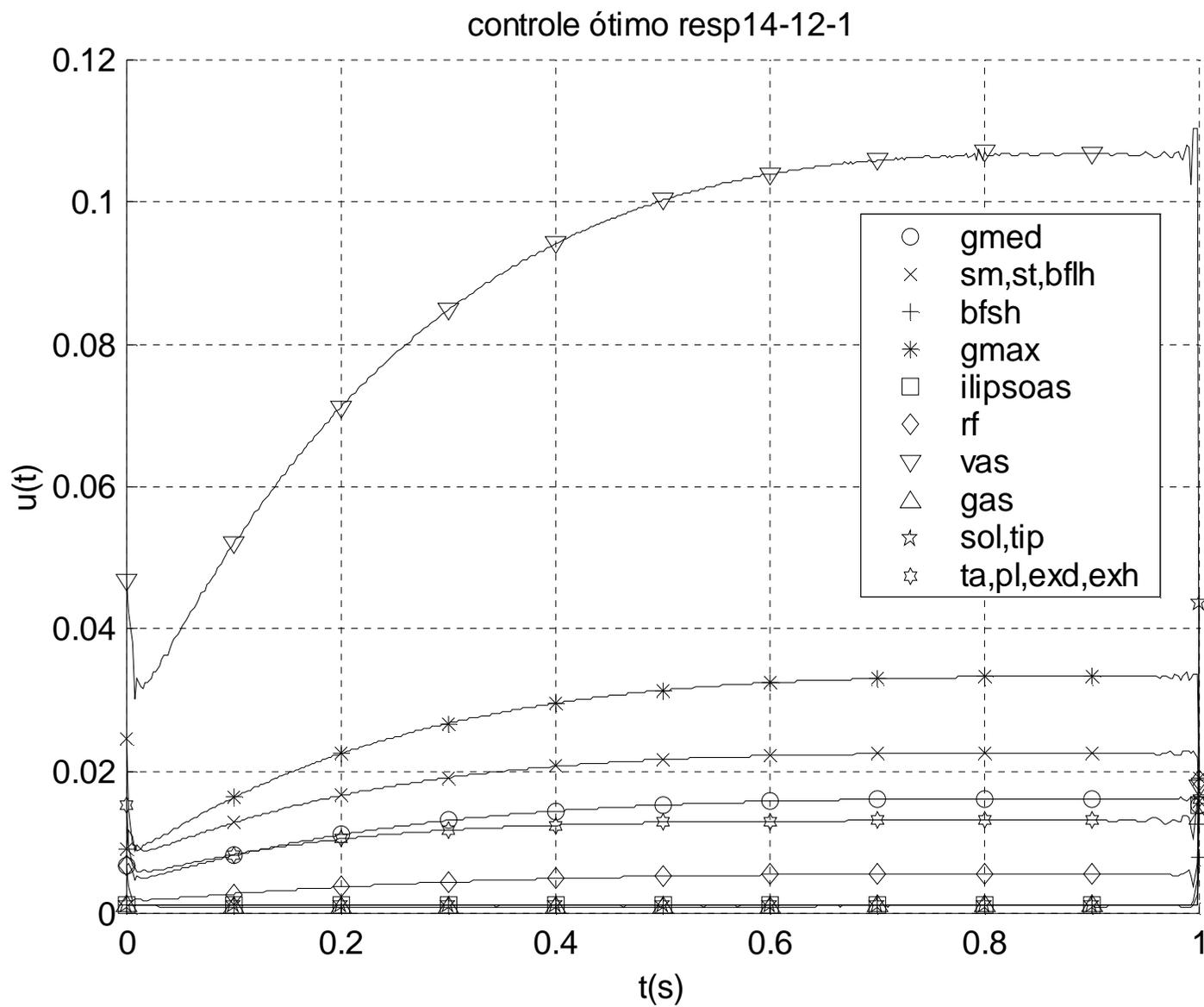


Figura 6.17: Controle 14_12_1

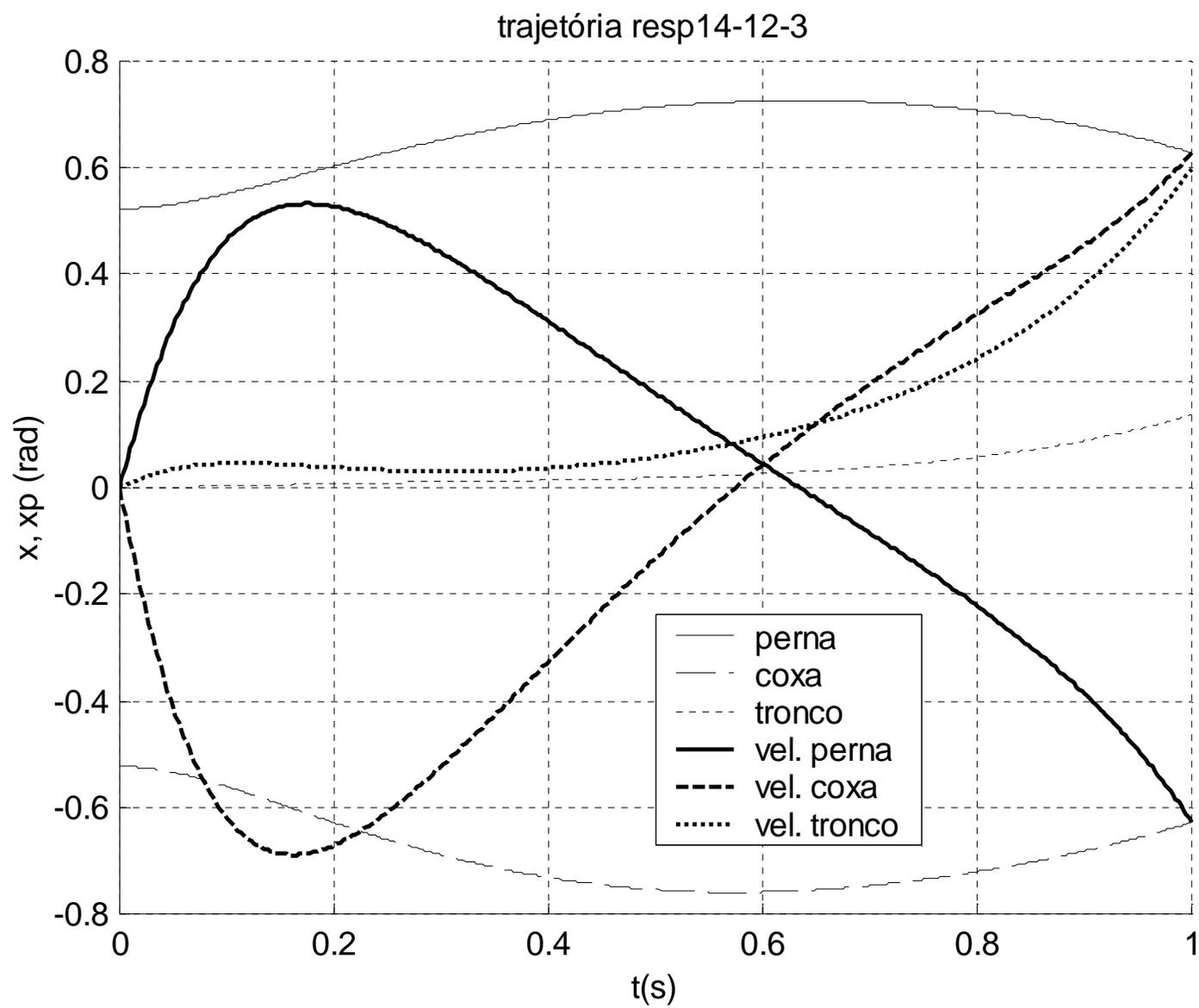


Figura 6.18: Trejetória 14_12_3

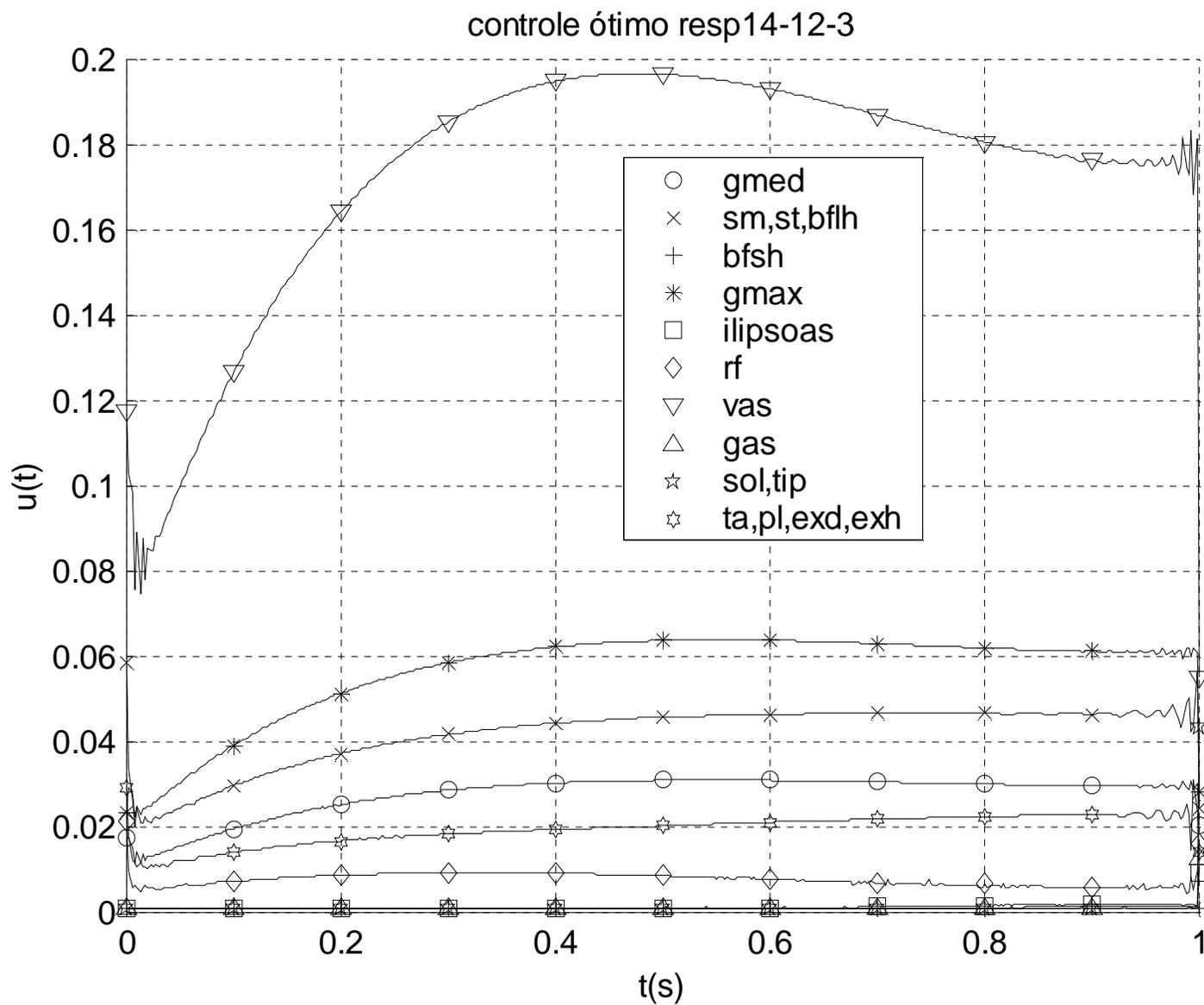


Figura 6.18: Controle14_12_3

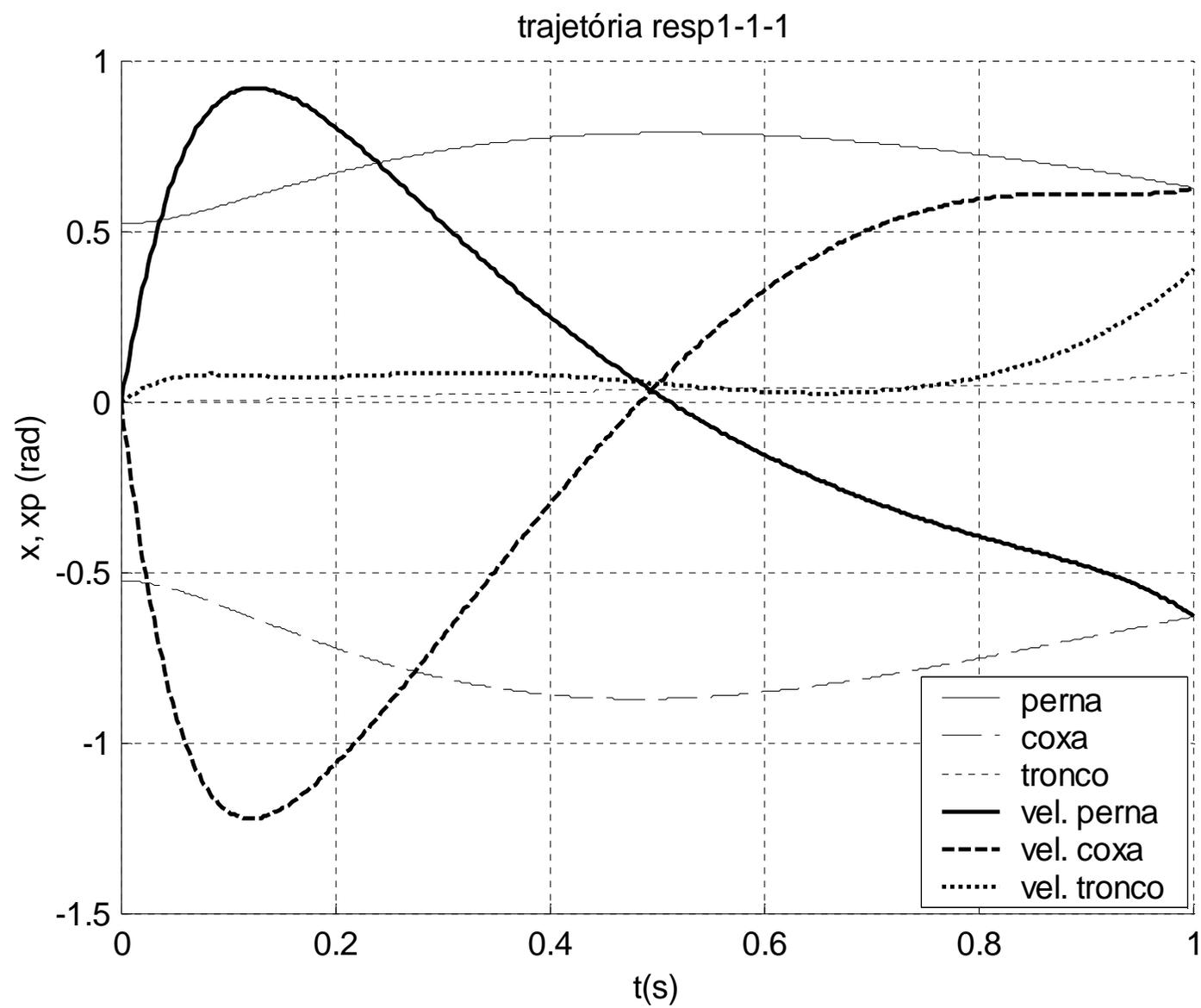


Figura 6.19: Trejetória 1_1_1

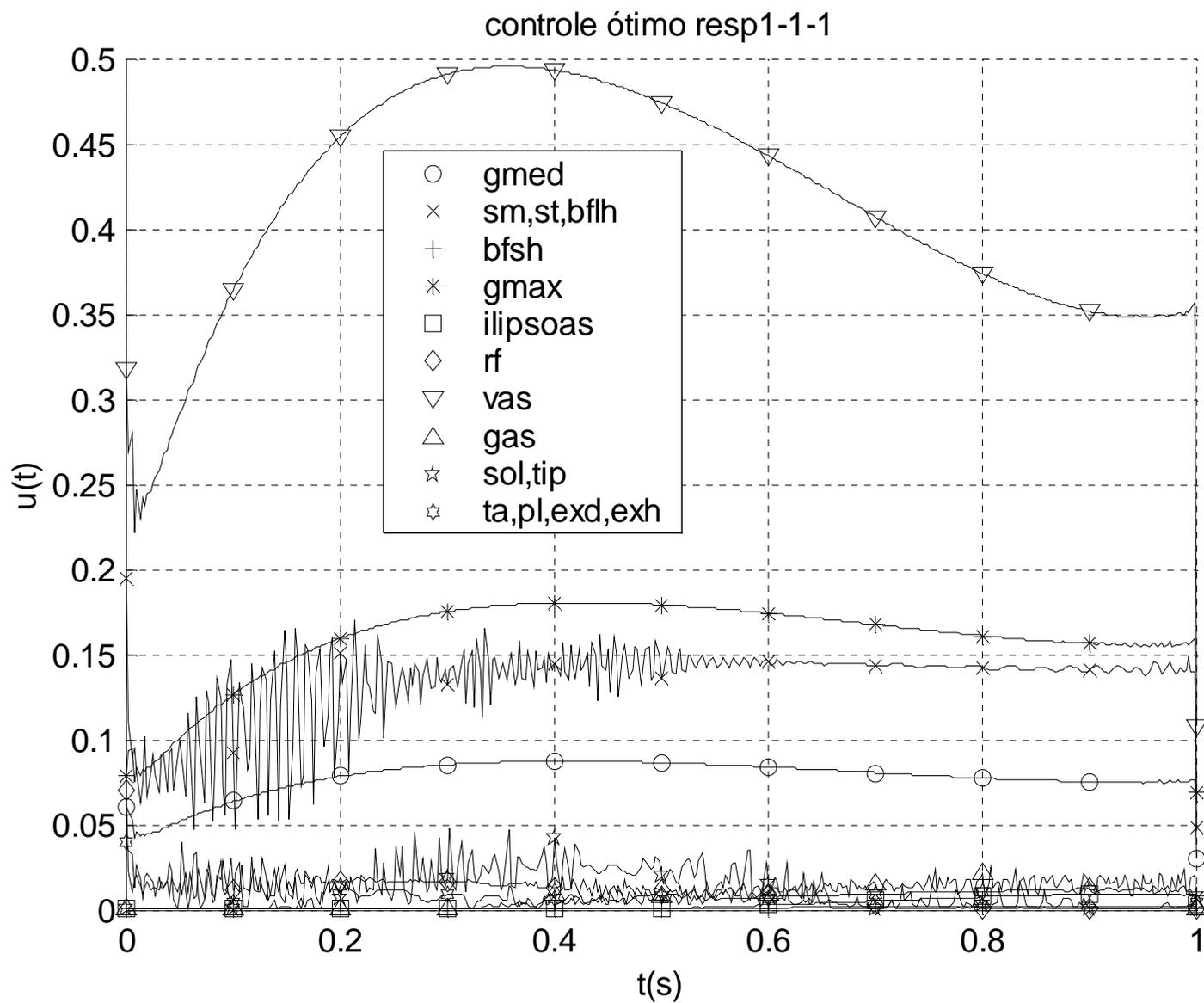


Figura 6.19: Controle 1_1_1

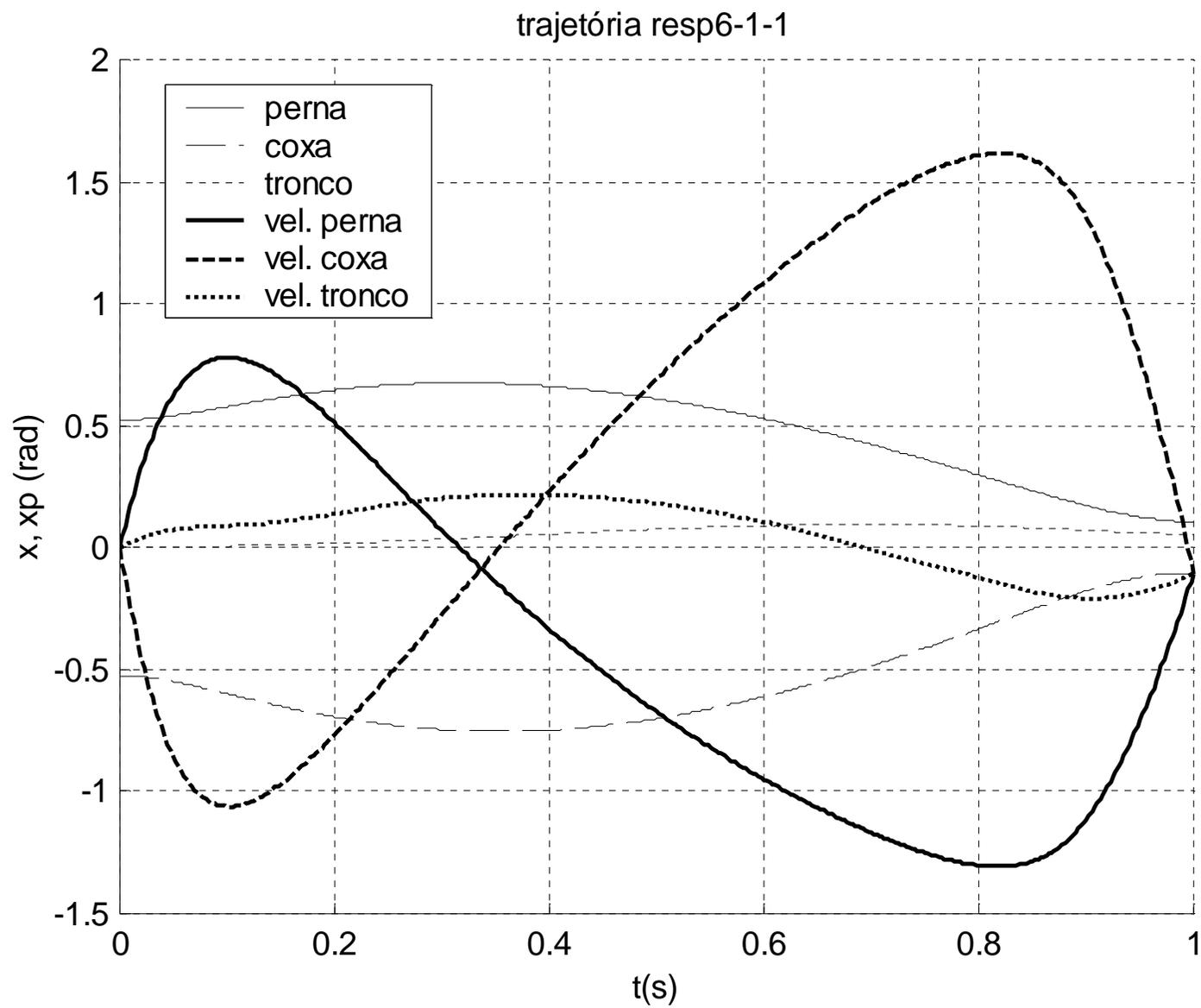


Figura 6.20: Trejetória 6_1_1

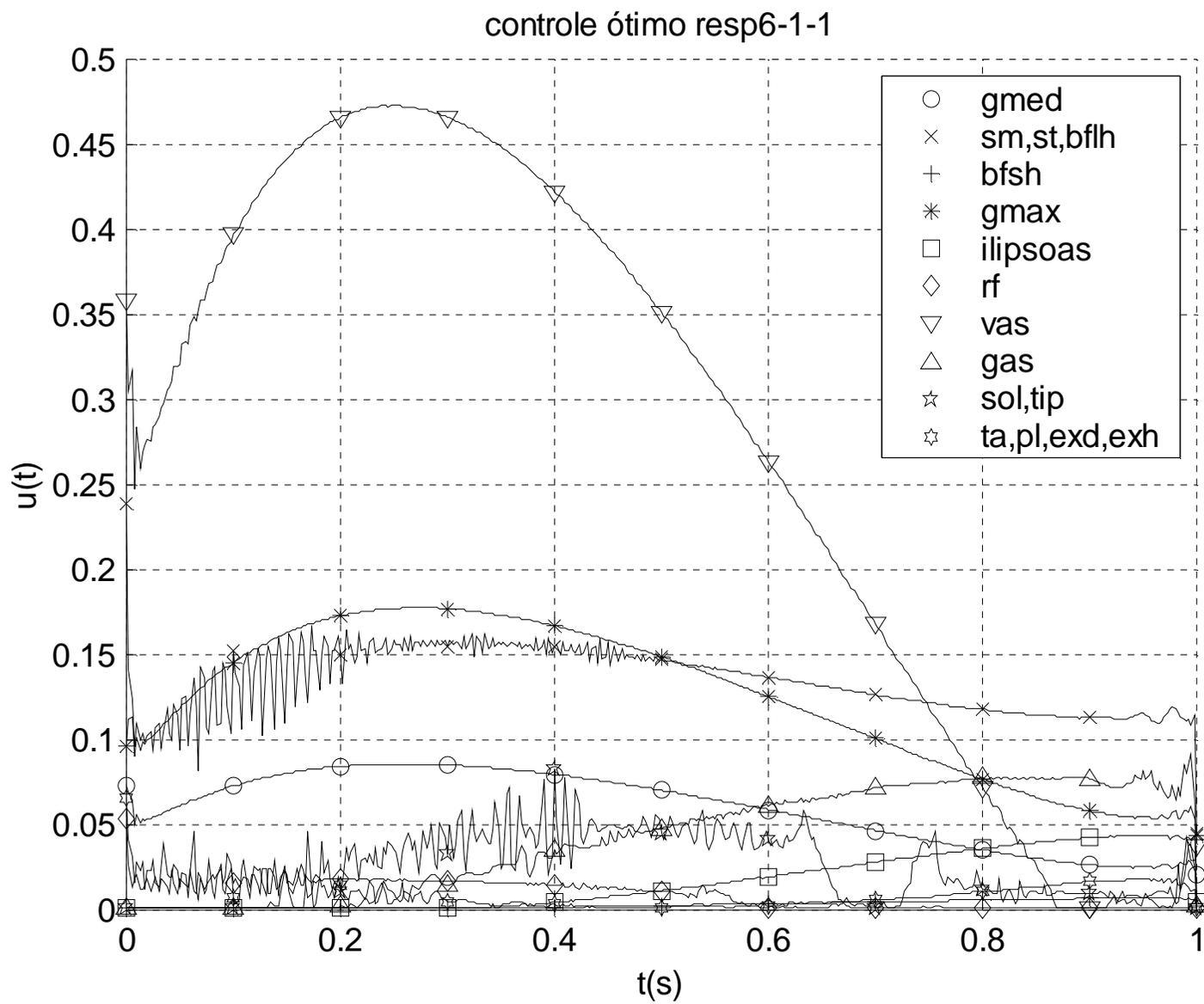


Figura 6.20: Controle 6_1_1

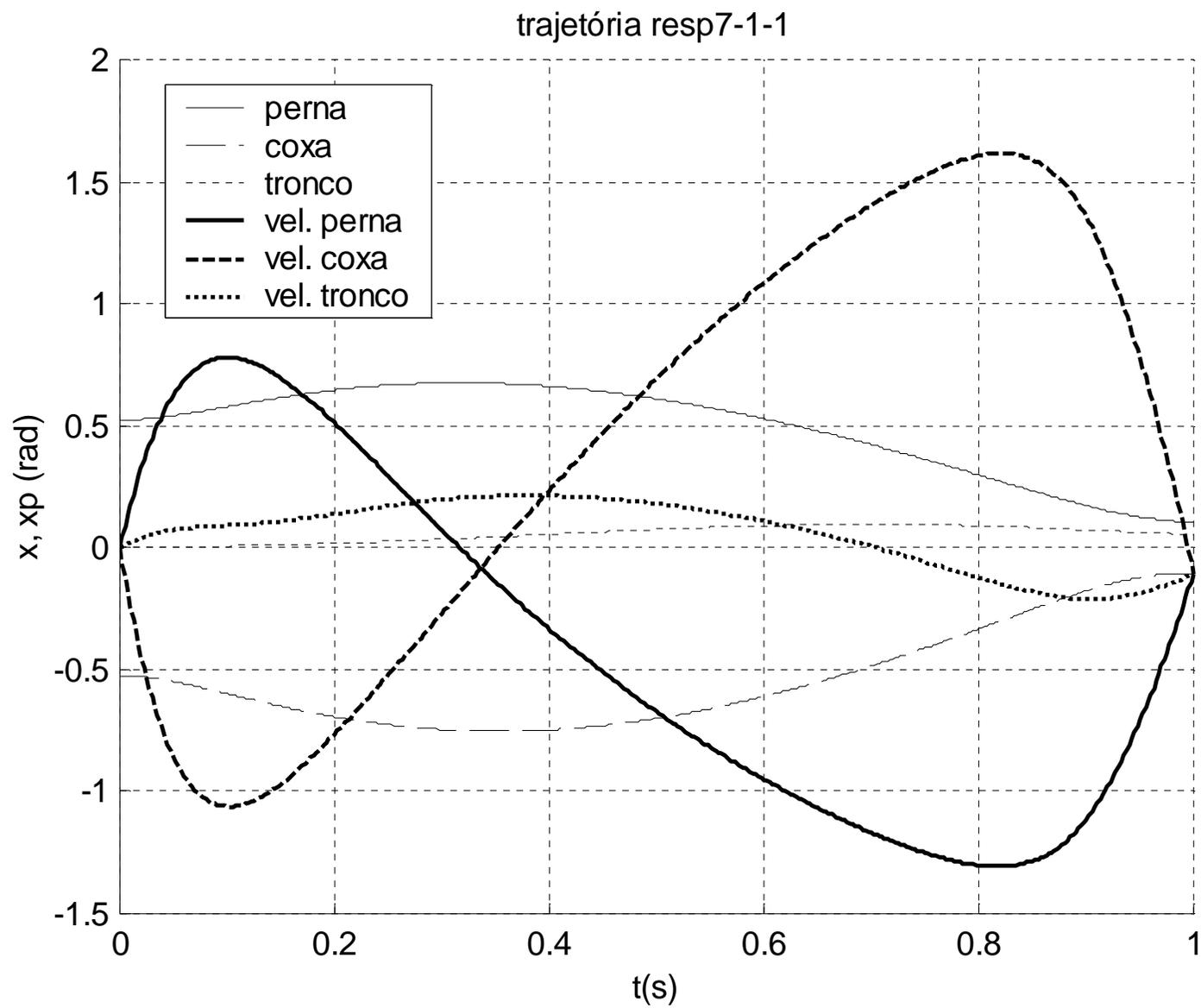


Figura 6.21: Trajetória 7_1_1

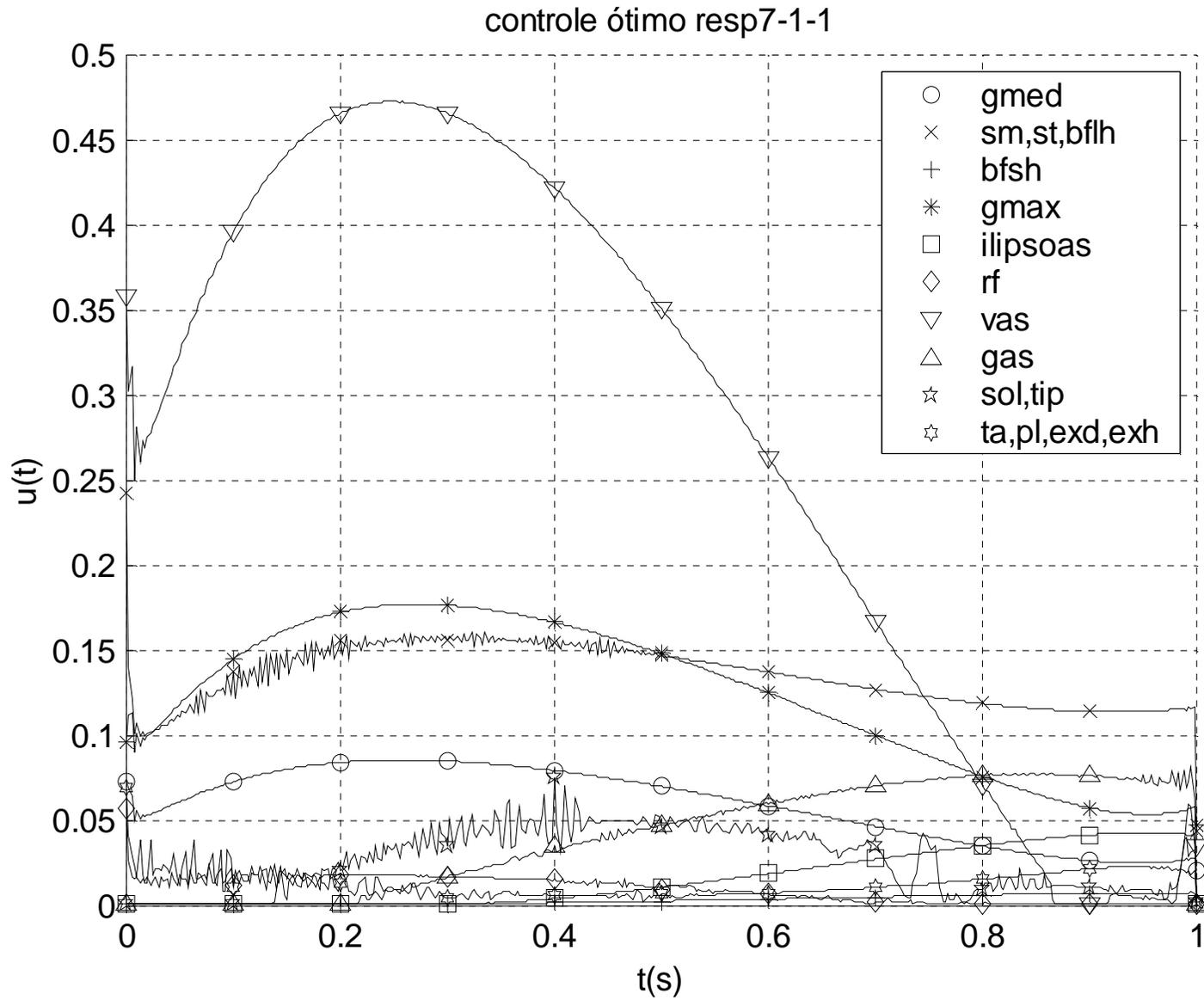


Figura 6.21: Controle 7_1_1

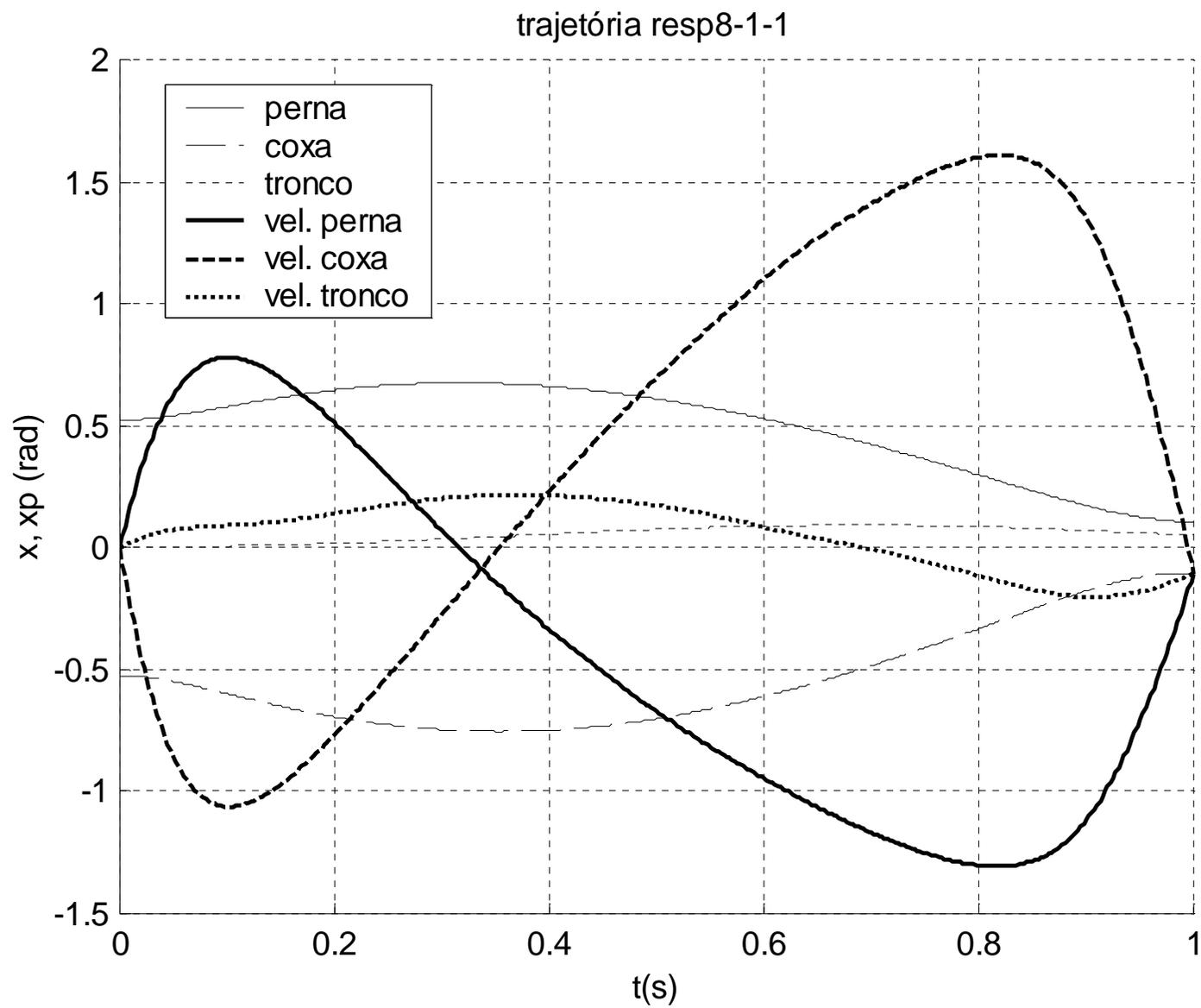


Figura 6.22: Trejetória 8_1_1

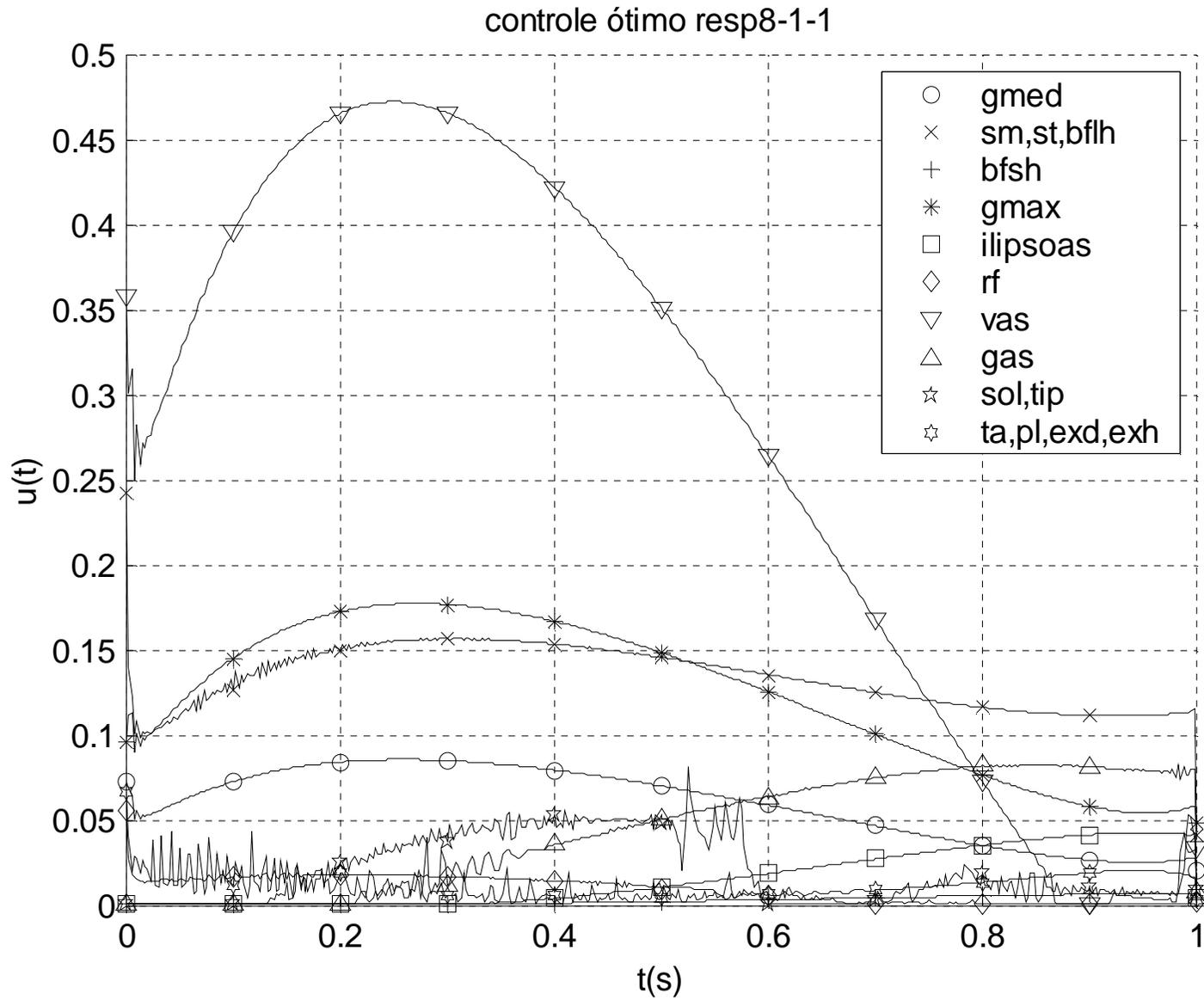


Figura 6.22: Controle 8_1_1

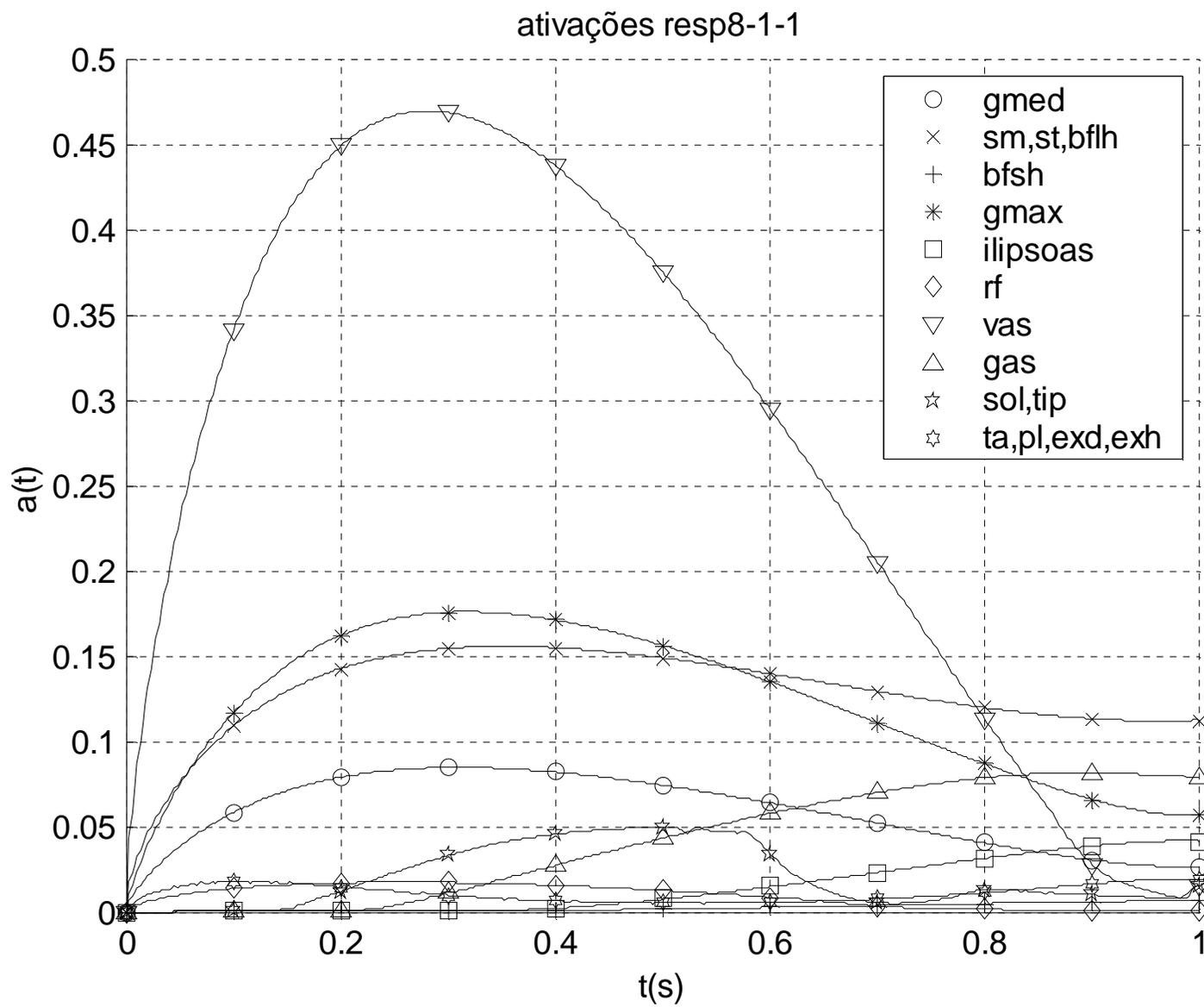


Figura 6.23: Ativações neuro-musculares 8_1_1

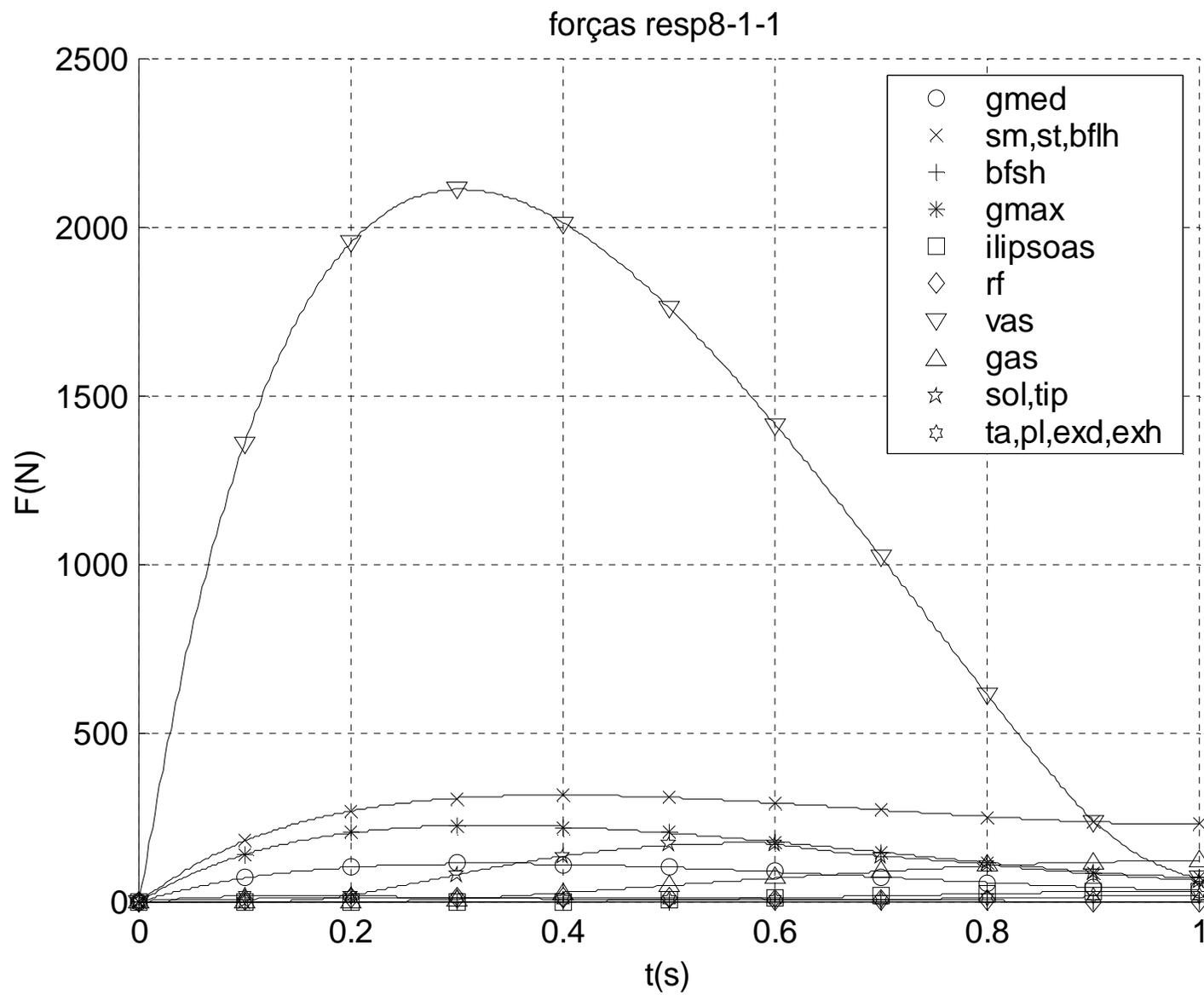


Figura 6.24: Forças musculares 8_1_1

6.3 Análise dos padrões de ativação obtidos

Observando as respostas obtidas nas simulações mostradas no item 6.1, pode-se perceber, inicialmente, uma distinção entre duas fases temporalmente distintas do padrão de ativação e, com algum atraso em relação a este, do padrão cinemático. A estas fases poderiam ser atribuídas as designações de *aceleração* e *frenagem*. Além disso, comparando as curvas de excitação (Figura 6.13), ativação (Figura 6.14) e força muscular (Figura 6.15), percebe-se claramente um padrão de filtragem passa-baixa, com atenuação das componentes de alta frequência e atraso do sinal.

Do ponto de vista do padrão de excitação muscular observado, pode-se atribuir a maior parte do esforço de levantamento do centro de massa do corpo, especialmente na primeira parte do movimento, aos *vasti*⁵⁵, que é um grupo muscular de elevada força máxima. Na assim chamada frenagem, os flexores *tibialis anterior*, *gastrocnemius* (flexor do joelho) e *ilipsoas* são ativados de maneira mais intensa, ainda que os níveis máximos de excitação sejam inferiores. O *tibialis anterior* é acionado para frear a extensão do tornozelo, realizada não tanto pelo seu extensor uniarticular, o *soleus*, como pelo *gastrocnemius*, que é utilizado para frear a extensão do joelho. O tronco, que inicialmente estava ereto, tende a inclinar-se para frente, num movimento de flexão. Para evitá-lo, os músculos *gluteus medius* e *gluteus maximus* são ativados. Na segunda parte do movimento, o *ilipsoas* trás o tronco diminuindo a velocidade angular negativa. Comparando a atividade do *biceps femoris caput longus* (bifemlh, ou gm₂) com o *caput brevis* (bifemsh, gm₃), este é excitado na segunda parte do movimento, agindo provavelmente como um flexor do joelho sinérgico ao *gastrocnemius*, ao passo que o primeiro é também um

extensor do quadril, cooperando, na *aceleração*, com os *glutei*. O *rectus femoris* praticamente não é ativado. Sendo um extensor do joelho, é também um flexor do quadril. Na primeira parte do movimento, o quadril deve ser estendido, e na segunda a extensão do joelho atenuada, e assim esse músculo fica silente.

Comparando esses resultados com os obtidos quando o tempo final era mantido em 1 segundo⁵⁶ (item 6.2 e Tabela 6.3), percebem-se inicialmente padrões semelhantes aos anteriores, mas com níveis de excitação (Figura 6.22), ativação (Figura 6.23) e, principalmente, forças musculares (Figura 6.24) mais baixas. Além disso, as velocidades máximas observadas na segunda parte do movimento são menores, ainda que se observe nitidamente, com 1 segundo, uma pequena *queda* dos membros no início do movimento, com maiores velocidades de flexão na perna e na coxa. Comparando padrão dos *glutei* com o caso anterior, percebe-se agora um atraso na sua ativação, assim como do *ilipsoas*, entrando apenas no terceiro *terço* do movimento, quando o tronco já se flexionou, estendeu e finalmente retorna à posição ereta. Esta oscilação mais pronunciada do tronco difere dos resultados observados com 0.4 segundos. A ativação dos *vasti* é mais distendida ao longo do movimento, não obstante atingir picos menores; tal tendência se repete no seu antagonista *biceps femoris caput longus*, mostrando um padrão de *coativação*, ainda que o *caput brevis* não seja excitado. Além disso, não se observa uma atividade muito significativa do *tibialis anterior* ao longo do movimento, senão no seu *terço* médio. As ativações do *gastrocnemius* e *ilipsoas* são notadamente inferiores ao caso precedente. Nota-se, entretanto, atividade discretamente maior do *rectus femoris*.

⁵⁵ Evidentemente, esta análise se refere aos resultados do modelo biomecânico cujos parâmetros estão definidos na Tabela 5.2, com bases nos grupos musculares gm_i , $i = 1, \dots, 10$. Para facilitar a análise dos resultados, esses grupos serão designados pelos seu componente mais característico.

Comparando ainda os resultados obtidos com o modelo simplificado com braços de momento fixos (Cap. 5), percebe-se que a introdução das curvas de regressão no modelo biomecânico levou a movimentos mais suaves, com menores velocidades máximas. As excitações e, principalmente, forças musculares foram bastante reduzidas. Nos *vasti*, por exemplo, a redução no pico de força atingiu 54%. O mesmo pode-se afirmar da própria função de custo, caindo de 4.0576 para 1.8382. As excitações mínimas com braços de momento fixos foram bastante superiores, denotando um padrão de coativação que pode ter contribuído para o aumento do custo, além do que as tolerâncias numéricas não foram as mesmas em ambos os casos. De qualquer modo, o limite inferior da variável de controle em r fixo foi zero.

Observando os padrões de ativação e excitação muscular, o *tibialis anterior* foi utilizado de maneira bem mais intensa com r fixo, ao passo que o *biceps femoris caput longus* menos. O *soleus* foi solicitado no terço médio do movimento de maneira bem mais intensa com r fixo. Isso pode ser devido ao braço de momento deste músculo ser bastante maior em 30° de flexão do que na posição ortostática, de acordo com a Tabela 6.1.

⁵⁶ As comparações entre essas duas séries de experimentos devem ser feitas com a ressalva de que os métodos de geração das estimativas iniciais do controle foram diferentes, bem como as tolerâncias numéricas empregadas.

Capítulo 7

Problema da Realimentação

Uma vez determinada a metodologia para a geração das trajetórias de excitação neuro-muscular em malha aberta, representando, de certo modo, a função do córtex motor, resta ainda abordar a seguinte questão: como incluir no modelo do controle postural proposto mecanismos de manutenção da trajetória ótima gerada em malha aberta através do fechamento da malha, usando regras de controle baseados nos estados atuais do sistema. Estes mecanismos, que do ponto de vista fisiológico correspondem a circuitos neuronais medulares e, em parte, do tronco cerebral, entram em ação, principalmente, para compensar perturbações impostas ao sistema, procurando fazer que as trajetórias planejadas em malha aberta sejam de fato seguidas. Deste modo, desvios nas variáveis de estado e controle são avaliados ao longo do movimento e são tomadas decisões em tempo real, a partir de regras simples. Do ponto de vista da teoria de sistemas dinâmicos, a introdução da realimentação visa aumentar as margens de estabilidade e as velocidades de resposta do sistema.

Tratando-se de um sistema altamente não linear como a postura humana, o estabelecimento de uma lei de realimentação direta não é uma tarefa simples. Para se poder usar técnicas lineares de projeto, deve-se ter em conta as não-linearidades do sistema, que só permitem aproximações lineares apenas em pequenos trechos da faixa de operação. Ainda mais, são de difícil tratamento os vínculos de controle $0 \leq u_i(t) \leq 1$, $i=1, \dots, m$, principalmente o inferior.

Uma das metodologias básicas testadas para a determinação de uma lei de realimentação parte da linearização do sistema num ponto determinado de operação. A partir das matrizes $[A]$, $[B]$ e $[C]$ obtidas por linearização calcula-se, considerando a hipótese de um controlador linear quadrático determinístico (LQR), uma matriz de ganhos $[K]$ e, a partir desta, implementa-se a lei de controle $\{u(x)\} = - [K] \{x(t)\}$. Para modelos do pêndulo triplo com atuadores sem dinâmica e dinâmicos de torque, sem vínculos de controle, a matriz de ganhos determinada a partir de um ponto de operação onde as coordenadas generalizadas são nulas é, em geral, capaz de controlar o sistema, levando-o desde uma condição inicial semelhante às mostradas acima, para o problema da malha aberta, até a posição de equilíbrio com as coordenadas generalizadas novamente nulas. Fazendo uma abstração e utilizando modelos com atuadores musculares lineares, sem vínculos de controle mas redundantes, a matriz de ganhos determinada também controla o sistema.

O problema surge quando se introduzem os vínculos de controle, principalmente o inferior. No caso do vínculo superior, $u(t) \leq 1$, alterando por tentativa e erro as matrizes Q e R , é possível encontrar leis de controle mais econômicas para as variáveis de controle que apresentam violação desse vínculo. Já o vínculo $0 \leq u(t)$ não permite esse tipo de

tratamento. Sua própria existência implica numa alteração importante da estratégia do controlador.

Foram analisadas neste projeto diversas estratégias para o tratamento do problema levantado. A primeira, baseada na metodologia empregada em Menegaldo (1997), que utilizava um modelo integrado da dinâmica da ativação para estudos de estimulação elétrica artificial do músculo, consistia nos seguintes passos:

1. Determinação de uma lei de controle baseada num modelo com atuadores de torque sem dinâmica.
2. Divisão dos torques de controle calculados em (1) entre várias forças musculares, através de uma otimização por mínimos quadrados, utilizando a matriz de braços de momento como ponderação. A função objetivo era tal que a soma vetorial das forças musculares fosse mínima para um dado torque requerido determinado pela lei de controle.
3. Alteração sucessiva da matriz de braços de momento, penalizando com valores constantes altos os braços de momento dos músculos que estivessem com forças negativas e refazendo o processo de otimização. No limite, todas as forças musculares estavam com valores superiores a um $-\epsilon$.
4. Aplicação de um modelo inverso da mecânica muscular para cálculo das excitações $u(t)$ capazes de gerar as forças musculares requeridas, calculadas no item anterior.

Para o problema com atuadores musculares lineares e não-lineares, esta abordagem, sugerida por Zajac e Gordon (1989) e utilizada por Riener e Fuhr (1998) não se mostrou eficaz. Num dado instante t , a excitação $u(t)$ calculada a partir dos valores atuais das derivadas das variáveis de estado, ao longo da integração numérica, não era capaz, no próprio instante t , de gerar a força muscular requerida, senão apenas algum

tempo depois, devido ao atraso da resposta do atuador (da ordem de 100 ms). Nas simulações realizadas, a medida em que o sistema ia divergindo, o controlador percebia o aumento do erro e reagia, requerendo altos valores de controle, que conseguiam fazer apenas que o sistema oscilasse antes de cair totalmente. Com isso, essa metodologia não foi mais utilizada.

Uma outra metodologia testada foi utilizar uma solução modificada da equação de Riccati apresentada por Bryson e Ho (1975, prob. 7 sec. 5.4). Novamente aqui, se partia de uma solução do pêndulo com atuadores de torque, neste caso dinâmicos e com constantes de tempo relativamente elevadas. Essa passa a ser uma trajetória de referência para os atuadores musculares: uma lei de controle variável no tempo procura fazer com que o vetor de forças musculares, multiplicado pela matriz de braços de momento, siga os torques determinados no modelo com atuadores de torque. Assim, para o sistema:

$$\dot{\mathbf{F}} = [\mathbf{A}]\mathbf{F} + [\mathbf{B}]\mathbf{u}, \quad \text{eq. 7.1}$$

o vetor \mathbf{F} representa o conjunto de ativações e forças musculares. Com atuadores que obedecem a relação

$$\mathbf{M}(t) = [\mathbf{r}]\mathbf{F}, \quad \text{eq. 7.2}$$

onde $[\mathbf{r}]$ é aumentada com uma matriz de zeros até atingir o mesmo tamanho que a matriz de braços de momento, para que sua multiplicação pelos estados, que incluem as ativações, seja consistente.

Deseja-se determinar uma lei de controle:

$$\mathbf{u} = -[\mathbf{C}]\mathbf{F} + \mathbf{w}(t), \quad \text{eq. 7.3}$$

que minimize a função de custo quadrática

$$J = \frac{1}{2} \int_{t_0}^{t_f} \{ \mathbf{M}(t) - [\mathbf{r}]\mathbf{F} \}^T [\mathbf{Q}] \{ \mathbf{M}(t) - [\mathbf{r}]\mathbf{F} \} + \mathbf{u}^T [\mathbf{R}] \mathbf{u} \, dt$$

Fazendo $[C] = [R]^{-1}[B]^T[S]$ e $w(t) = -[R]^{-1}[B]^T g$

obtem-se o sistema:

$$\begin{cases} \dot{[S]} = -[S][A] - [A]^T[S] + [S][B][R]^{-1}[B]^T[S] - [r]^T[Q][r] \\ \dot{g} = -([A]^T - [S][B][R]^{-1}[B]^T)g + [r]^T[Q]M(t) \end{cases}$$

que deve ser integrado com as condições finais $S(t_f)=0$ e $g(t_f)=0$.

Numa aproximação inicial, a primeira equação, e portanto $[C]$, pode ser calculada utilizando a solução da equação de Riccati com $\dot{[S]}=0$, por exemplo, através da função `are.m` do *Control Systems Toolbox* do Matlab. A segunda equação é calculada determinando-se inicialmente a seqüência de vetores $[r]^T[Q]M(t)$ em função das curvas de torque calculadas anteriormente. Com essa informação, a equação em \dot{g} é integrada a partir do seu vínculo terminal de trás para a frente, utilizando um RK4 passo fixo adaptado.

Os resultados obtidos para um sistema com 8 músculos, dois deles biarticulares, mostraram que o conjunto de músculos reproduzia a curva de torque com certa precisão, tanto maior quanto mais fina a discretização. Entretanto, o problema dos vínculos de controle ainda não havia sido solucionado. Para tanto, foi implementado um algoritmo que identificava para cada ponto da malha de discretização quais eram os vínculos que haviam sido violados (quais controles eram negativos) e esta informação guardada numa *matriz de sinais*. Em função dessa matriz, eram determinados diversos trechos que mantinham o mesmo padrão da matriz de sinais, e para cada um deles atribuía-se uma matriz $[R]$ específica, aumentando os pesos, e portanto o custo do controle, daquelas variáveis que violavam os vínculos. Assim, para cada trecho, as equações matriciais e a integração de trás para a frente de \dot{g} eram feitas independentemente. Com isso eram encontradas matrizes $[C]$ e w variantes no tempo, que levavam em conta as tais variações de $[R]$.

Entretanto, quando as soluções de controle encontradas eram introduzidas no sistema dinâmico, seu comportamento dinâmico divergia rapidamente daquele observado com os sinais de torque originais. Num caso extremo, em que a metodologia foi aplicada simplesmente na decomposição de um sinal de torque num conjunto de forças estáticas, utilizando 10000 pontos na malha de discretização, a distância entre os torques originais e reproduzidos eram da ordem de 10^{-3} N.m. Mesmo assim, a solução com os torques reproduzidos divergia.

É difícil explicar com suficiente propriedade a ocorrência deste comportamento. Atribuímo-lo, entretanto, à natureza altamente não linear – com comportamento em certas condições caótico – do pêndulo triplo. Sabe-se que nestas condições, pequenas variações nas condições iniciais e na excitação do sistema podem levar a grandes divergências de soluções, fenômeno que pode ser medido através do coeficiente de Liapunov (Strogatz, 1994). Desse modo, a pequena diferença numérica que decorre da reconstrução das curvas de torque faz com que o sistema se afaste rápida e definitivamente do ponto de equilíbrio instável que caracteriza os vínculos terminais do sistema em malha aberta.

Outra tentativa de solucionar o problema da realimentação, semelhante à anterior, considerou a utilização do próprio pacote RIOTS, ou seja, o problema foi formulado como um OCP não-linear. Após diversos testes, foi possível reproduzir razoavelmente as curvas de torque, utilizando, num modelo com 6 músculos não-lineares, o método de otimização do gradiente conjugado, integrador de passo variável e um grande número de iterações (acima de 3000). Pelas mesmas razões apontadas na abordagem anterior, o sistema não convergia quando os valores calculados foram substituídos no modelo global

A partir destes estudos, o sentimento é que o problema em malha fechada dificilmente será resolvido a partir de tentativas de reprodução de uma curva de torques

nominais. Não é possível afirmar no presente momento qual metodologia alternativa seria adequada, mas será muito provavelmente baseada na determinação direta das excitações musculares.

Capítulo 8

Discussão e perspectivas

Com base na metodologia proposta e nos resultados apresentados nos capítulos precedentes, pode-se tecer algumas considerações a respeito da viabilidade do método proposto, de suas atuais limitações e de perspectivas de aprimoramento, tendo em conta sua futura aplicação clínica.

Uma das características mais notáveis do problema em questão, e não diferentemente da solução proposta, é a sua complexidade, bem como as infinitas possibilidades de escolha de parâmetros e hipóteses de trabalho, tanto do modelo biomecânico quanto da solução numérica. Com isso, a *política* adotada ao longo do trabalho privilegiou a exploração das soluções viáveis que, dentre as muitas testadas, se apresentaram. Desse modo, testes sistemáticos não foram levados a cabo, senão em torno destas mesmas soluções mais promissoras. Dado o caráter, até o que se sabe, inédito da

utilização do método de controle ótimo estudado em problemas de biomecânica músculo-esquelética, e sua complexidade, entende-se esta opção.

Do ponto de vista da escolha dos parâmetros dos métodos numéricos e da complexidade do modelo biomecânico, pode-se fazer três observações principais:

1. **A escolha da estimativa inicial do controle é central para a convergência das simulações.** Deve-se, assim, sempre que possível, optar por uma estimativa que respeite os vínculos de controle, ou os viole pouco, e que não afaste o sistema, no final da simulação, demasiadamente dos vínculos terminais. Muitas vezes é necessário resolver OCPs parciais com vínculos menos estreitos e utilizar a solução como estimativa inicial do controle.
2. **A escolha do tempo de simulação é crítico para a convergência das simulações.** É de senso comum que o Sistema Nervoso Central pode modular voluntariamente a velocidade do movimento, realizando a tarefa proposta no tempo desejado, ainda que exista provavelmente um tempo preferencial. Tal modulação não é trivial no problema numérico, devendo-se lançar mão de estratégias específicas para realizá-la num modelo computacional. Quando o tempo de simulação supera o tempo crítico, é necessário utilizar o RK1 nos modelos com músculos lineares e estimativa inicial do controle por regulador LQR. No modelo com atuadores musculares não-lineares a estratégia da gravidade variável, com integrador RK4 e splines de 4^a ordem, mostrou-se especialmente promissora. A justaposição de novos nós à malha de simulação só foi funcional para um aumento de 30% no tempo de simulação, dentro das condições empregadas.

3. A determinação até onde for possível precisa das forças e excitações musculares requer o emprego de modelos não lineares da mecânica muscular e braços de momento variáveis

Tendo em conta especificamente o modelo com atuadores musculares não-lineares, além da estabilidade numérica do integrador, as principais questões aparecem na escolha das tolerâncias de otimização e violação de vínculos do método SQP. Não foi possível explicar, neste trabalho, de que exata maneira as tolerâncias influenciam na estabilidade do algoritmo, mas procurou-se, entretanto, apontar diretrizes básicas para sua escolha bem sucedida, em cada um dos modelos estudados. Por outro lado, o integrador Runge-Kutta e splines de 4^a parecem ser as mais indicadas para qualquer dos modelos, quando os algoritmos baseados na Teoria das Aproximações Consistentes são utilizados na solução de problemas como o aqui proposto. Recomenda-se ainda a aplicação gradativa e na etapa correta dos vínculos terminais (principalmente com braços de momento variáveis), sob pena de não atingir resultados convergentes ou chegar a mínimos locais da função de custo com escasso significado fisiológico. Os vínculos de controle também devem ser impostos de maneira gradual ainda que, para as tarefas posturais aqui estudadas, dificilmente os vínculos superiores de controle sejam atingidos. Alguns testes realizados (mas não expostos nesse texto) com metade da capacidade muscular nominal do modelo, indicam que, em tarefas posturais de maior solicitação muscular, tal imposição gradual dos vínculos superiores é determinante.

Apesar dos algoritmos de controle ótimo terem sido testados em diversos problemas acadêmicos por seus autores, e possuírem uma base analítica altamente consistente do ponto de vista da análise matemática, seu desenvolvimento é ainda recente,

bem como seu emprego em problemas reais e complexos de controle ótimo. Seria assim recomendável que os modelos aqui desenvolvidos fossem controlados a partir de outros algoritmos de controle ótimo, o que poderá ser feito no futuro. Além disso, o RIOTS possui diversos erros de implementação, que se manifestam quando o sistema dinâmico se instabiliza. A solução de tais problemas precisa ainda ser levada a cabo, e constituiram um dificuldade considerável na realização do presente trabalho.

A função de custo utilizada, minimizando a integral da soma dos quadrados das derivadas temporais das forças musculares ao longo do tempo de simulação, mostrou-se viável, levando a soluções plausíveis com custos computacionais elevados mas não inviáveis, como seria, provavelmente, a minimização do *jerk*, devido às extensas expressões analíticas que seriam obtidas. As funções tipo LQR não se mostram promissoras, com dificuldades de convergência e alto custo computacional, não obstante sua simplicidade e convexidade.

A partir destas observações, o desenvolvimento futuro deste trabalho procurará abordar alguns problemas não resolvidos:

1. Testar comparativamente novas funções de custo como, por exemplo, a maximização da altura do centro de massa. Deve-se, ainda, testar ainda outras condições iniciais dos estados.
2. Realizar estudos do movimento de levantar-se de uma cadeira e da marcha, extremamente relevantes no problema de excitação artificial de músculos neurologicamente paralizados.

3. Realizar testes com comprimento muscular \times força máxima não constante, utilizando as expressões analíticas do comprimento de atuadores musculares em função de coordenadas generalizadas, determinadas no Capítulo 2. Com a introdução destas expressões, deixa de vigorar a hipótese de contrações musculares isométricas.
4. Introduzir no modelo biomecânico expressões de momento passivo gerado por ligamentos do joelho (Montandon et al., 1998).
5. Realizar novos testes com tempo final aberto, ainda que os testes preliminares com o pêndulo simples indiquem dificuldades numéricas importantes.
6. Implementar uma metodologia para determinação das condições iniciais das variáveis de estado através de otimização estática (Pandy, 1990)

Tendo em conta o Estado da Arte comentado na Introdução, pode-se considerar, salvo melhor juízo, que o presente trabalho forneceu, na solução do problema de controle ótimo, as seguintes contribuições:

1. Uma formulação de maneira matematicamente rigorosa e numericamente econômica do problema de controle ótimo de sistemas neuro-musculo-esqueléticos.
2. Determinação de diretrizes para o emprego dos algoritmos baseados na Teoria das Aproximações Consistentes em problemas de controle postural.
3. Obtenção de sinais de controle com a mesma resolução temporal da dinâmica discretizada.
4. Dedução de fórmulas compactas para cálculo, durante as simulações, dos braços de momento e comprimento dos atuadores músculo-tendíneos.

A médio prazo, desenvolvimentos deste trabalho poderão abordar novamente a questão da realimentação, propondo novos métodos de controle mais adequados. Além disso, a forte demanda computacional dos métodos numéricos utilizados e, ao mesmo tempo, a sua viabilidade na solução do problema proposto, poderia justificar sua re-implementação em linguagens computacionais mais portáteis, prescindindo do Matlab, para utilização em computadores de grande porte, além da utilização de algoritmos paralelos de otimização. A confirmação dos padrões de ativação muscular obtidos poderia ser feita mediante testes experimentais em laboratórios de marcha com sistemas de aquisição da cinemática e de atividade EMG. Utilizações clínicas do método, por exemplo na simulação funcional de cirurgias ortopédicas, requererão ainda o estabelecimento de metodologias e protocolos para a determinação de parâmetros musculares individuais.

Referências Bibliográficas

An, K.N., Kaufman, K.R., Chao, E.Y-S., Estimation of Muscle and Joint Forces, in *Three-Dimensional Analysis of Human Movement*, P.Allard, I.A.F.Stokes, J-P., Blachi Eds., 1st. Edition, Human Kinetics, 1995.

Anderson, F. C., Pandy, M. G. A dynamic optimization solution for vertical jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering* Vol. 2, pp. 201-231, 1999.

Anderson, F. C., Pandy, M. G. A dynamic optimization solution for jumping in three dimensions, *Computer Methods in Biomechanics and Biomedical Engineering* –2, Middleton, J., Jones M. L., Pande, G. N., eds., Gordon and Breach Science Publishers, pp. 11-18, 1998.

Anderson, F. C., Ziegler, J. M., Pandy, M. G., and Whalen, R. T. Solving large-scale optimal control problems for human movement using supercomputers. In Witten, J. and Vincent, D.J. (eds.): *Building a Man in the Machine: Computational Medicine, Public Health, and Biotechnology*. Part 2, World Scientific, pp. 1088-1118, 1996

Baltahsar, M., Mountian, I., Willems, P.Y., On the Neuro-Muscular Control of Trochlean Joints, *ASME Journal of Biomechanical Engineering*, Vol. 118, pp. 349-356, 1996.

Barin, K., Evaluation of a Generalized Model of Human Postural Dynamics and Control in the Saggital Plane, *Biological Cybernetics*, Vol. 61, pp. 37-50, 1989.

Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Vol.1, Athena Scientific, Belmont, Mass., 1995.

Betts, J. T., Very Low Thrust Trajectory Optimization, *preprint*, Boeing Information and Support Services, s.d.

Brand, R.A., Crowninshield, R. D., Wittstock, C. E., Pederson, D. R., Clark, C. R., van Krieken, F. M, A Model of Lower Extremity Muscular Anatomy, *ASME Journal of Biomechanical Engineering*, Vol.104, pp.304-310, 1982.

Brand, R.A., Pedersen, D.R., Friederich, J.A., The Sensitivity of Muscle Force Predictions to Changes in the Physiologic Cross-Sectional Area, *Journal of Biomechanics*, Vol.19, No.8, pp. 589-596, 1986.

Bryson, A.E. Jr., Ho, Y.C., *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere Pub. Corp., 1975.

Buchnan, T.S., Shreeve D.A., An Evaluation of Optimization Techniques for the Prediction of Muscle Activation Patterns During Isometric Tasks, *ASME Journal of Biomechanical Engineering*, Vol.118, pp.565-574, 1996.

Butler, D.L., Grood, E.S., Noyes, F.R., Zernicke, R.F., Brackett, K., Effects on Structure and Strain Measurement Technique on the Material Properties of Young Human Tendon and Fascia, *Journal of Biomechanics*, Vol. 17, pp. 579-596, 1984

Byrne, G. D., Hindsmarsh, A. C., Stiff ODE Solvers: A Review of Current and Coming Attractions, *Journal of Computational Physics*, Vol. 70, pp. 1-62, 1987

Canon, M. D., Cullum Jr., C. D., Polak, E., *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill Series in Systems Science, McGraw-Hill Inc., 1970.

Citron, S.J., *Elements of Optimal Control*, Holt, Rinehart and Winston Inc., 1969.

Cuthrell, J. E., Biegler, L.T., On the Optimization of Differential-Algebraic Process Systems, *AIChE Journal*, Vol.33, No.8, pp.1257-1270, 1987.

Daniel, J. W., *The Approximate Minimization of Functionals*, Prentice-Hall, 1971.

Delp, S.L. and Loan, J.P.: A software system to develop and analyze models of musculoskeletal structures, *Computers in Biology and Medicine*, vol. 25, pp. 21-34, 1995.

Delp, S.L., Loan, J.P., Hoy, M.G., Zajac, F.E., Topp, E.L., Rosen, J.M., An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures, *IEEE Transactions on Biomedical Engineering*, Vol. 37, pp. 757-767, 1990.

Domingues, H.H., *Espaços Métricos e Introducao à Topologia*, Ed. Atual e Edusp, 1982.

Fleury, A. T., *Otimização da Distribuição de Massa e da Frequência Fundamental de uma Viga em Balanço*, Tese de Mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, 1978.

Friederich, J.A., Muscle Fiber Architecture in the Human Lower Limb, *Journal of Biomechanics*, Vol. 23, No.1, pp. 91-95, 1990.

Gans, C., Bock, W.J., The Functional Significance of Muscle Architecture: A Theoretical Analysis, *Ergeb. Anat. Entwickl. Gesch.*, Vol. 38, No.115, 1965

Gill, P. E., Murray, W., Saunders, M. A., Wright, M., *User's Guide for NPSOL 5.0: a Fortran Package for Nonlinear Programming*, Systems Optimization Laboratory, Stanford University, Technical Report SOL 86-1, Revised, 1998.

Gill, P. E., Murray, W., Wright, M., *Practical Optimization*, Academic Press Limited, London, 1981.

Greenberg, M. D., *Foundations of Applied Mathematics*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1978

Hager, W.W., Rates of Convergence for Discrete Approximations to Unconstrained Control Problems, *SIAMM Journal of Numerical Analysis*, Vol. 13, No. 4, 1976.

He,J., Levine,W.S., Loeb,G.E., Feedback Gains for Correcting Small Perturbations to Standing Posture, *IEEE Transactions on Automatic Control*, Vol. 36, No. 3, pp. 232-332, 1991.

Hönig, C.S., *Análise Funcional e Aplicações*, Publ. IME-USP, 1970

Hosea M.E., Shampine,L.F., Estimating the Error of Classical Runge-Kutta Formula, *Applied Mathem. and Comp.*, Vol.66, pp. 227-226, 1994.

Inman,V.T., *The Joints of the Ankle*, Willians and Wilkins, 1973

Jennings, L. S., Fisher, M. E., Teo, K. L., Goh, C. J., *Miser3: Optimal Control Software, Version 2.0, Theory and User Manual*, Department of Mathematics, University of Western Australia, 1997.

Jennings, L. S., Teo, K. L., Fisher, M. E., , Goh, C. J., Optimal Control Software MISER3 in biomechanics, *Abstracts of 3rd International Symposium on Computer Simulation in Biomechanics*, University of Western Australia, Perth, Australia, 1991.

Jennings, L. S., Wong, K. H., Teo, K. L., An Optimal Control Problem in Biomechanics, *IFAC World Congress*, Vol. IV, pp. 311-314, Ed. Institute of Engineers of Australia, Sidney, Australia, 1993.

Kaplan, M. L., Heegaard, J. H., An Efficient Optimal Control Algorithm for Human Locomotion, *Applied Mechanics in the Americas*, Vol. 6, pp. 31-34, Gonçalves, B., Jasiuk, I., Pamplona, D., Steele, C.R., Weber, H.I., Bevilacqua, C., Editors, American Academy of Mechanics e Brazilian Society of Mechanical Sciences, Rio de Janeiro, 1999.

Kirk, D., *Optimal Control Theory;an Introduction*, Prentice-Hall, 1970.

Kraft, D., Algorithm 733: TOMP - Fortran Modules for Optimal Control Calculations, *ACM Transactions on Mathematical Software*, Vol., 20, No. 3, pp. 262-281, 1994.

Kraft, D. : *TOMP -- FORTRAN Modules for Optimal Control Calculations*, Fortschritt-Berichte VDI Reihe 8 No. 254, VDI-Verlag, Düsseldorf, 1991.

Kuo, A., A Mechanical Analysis of Force Distribution between Redundant, Multiple Degree-of-Freedom Actuators in the Human: Implications for the Central Nervous System, Submitted to *Human Movement Science*, December 1996.

Kuo, A., An Optimal Control Model for Analyzing Human Postural Balance, *IEEE Transactions on Biomedical Engineering*, Vol. 42, No. 1, 1995.

Legreneur,P., Morlon,B., Van Hoecke,J., Simulation of *in situ* Soleus Isometric Force Output as a Function of Neural Excitation, *Journal of Biomechanics*, Vol. 29, No. 11, pp. 1455-1463, 1996.

Massion,J., Movement, Posture and Equilibrium: Interaction and Coordination, *Progress in Neurobiology*, Vol. 38, pp. 35-56, 1992.

Menegaldo, L.L., I.A. Cestari, A.C.S. Carvalho Jr., A. T. Fleury, H.I., Weber, A model of Skeletal Muscular Dynamics: Analytical Development and Experimental Testing in Canine Lastissimus Dorsi, *Applied Mechanics in the Americas*, Vol. 6, pp. 11-14, Gonçalves, B., Jasiuk, I., Pamplona, D., Steele, C.R., Weber, H.I., Bevilacqua, C., Editors, American Academy of Mechanics e Brazilian Society of Mechanical Sciences, Rio de Janeiro, 1999.

Menegaldo, L.L., *Modelagem Matemática, Simulação e Controle Artificial da Postura em Seres Humanos*, Dissertação de Mestrado, Faculdade de Engenharia Mecânica - UNICAMP, 1997.

Menegaldo, L.L., Weber, H.I., Biomechanics of Upright Standing in Humans: a Simulation Study, *Computer Methods in Biomechanics and Biomedical Engineering -2*, Middleton, J., Jones M. L., Pande, G. N., eds., Gordon and Breach Science Publishers, pp. 775-782, 1998.

Monatndon, M.T., Menegaldo, L.L, Fleury, A.T., *Determinação de Braços de Momento, do Comprimento de Ligamentos e do Centro de Rotação do Joelho Humano através de um Software de Modelagem do Sistema Músculo-esquelético*, Relatório de Atividades, Bolsa de IC FAPESP (proc. n. 97/12625-0), 1998.

MusculoGraphics Inc., SIMM Software for Interactive Musculoskeletal Modeling, Version 1.2.5, *User Manual*, may 1997.

Pandy, M. G., Zajac, F. E., Sim, E., Levine, W. S., An Optimal Control Model for Maximum-Height Human Jumping, *Journal of Biomechanics*, Vol. 23, No. 12, pp. 1185-1198, 1990.

Pandy, M. G., Zajac, F. E., Optimal Muscular Coordination Strategies for Jumping, *Journal of Biomechanics*, Vol. 24, No.1, pp.1-10, 1991.

Pandy,M.G., Anderson,F.C., Hull,D.G., A Parameter Optimization Approach for the Optimal Control of Large-Scale Musculoskeletal Systems, *ASME Journal of Mechanical Engineering*, Vol. 114, pp. 450-460, 1992.

Pandy,M.G., Garner,B.A., Anderson,F.C., Optimal Control of Non-Ballist Muscular Movements: A Constraint-Based Performance Criterion for Rising From a Chair, *ASME Journal of Biomechanical Engineering* Vol.117, pp.15-26, February 1995.

Parker, T. S., Chua, L. O., *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag, New York, 1989.

Petzold, L. R., Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Differential Equations, *SIAMM Journal of Scientific and Statistical Computing*, Vol. 4, pp. 136-148, 1983

Piazza,S.J., Delp,S.L., The Influence of Muscles on Knee Flexion During the Swing Phase of Gait, *Journal of Biomechanics*, Vol. 29, No. 6, pp. 723-733, 1996.

Polak, E., *Computational Methods in Optimization - A Unified Approach*, Mathematics in Science and Engineering Vol. 77, Academic Press Inc., New York, 1971.

Polak, E., Mayne, D.Q., First-order Strong Variation Algorithms for Optimal Control, *Journal of Optimization Theory Appl.*, Vol. 16, No. 3., pp. 277-301, 1975.

Polak, E., *Optimization: Algorithms, and Consistent Approximations*, Marsden, J.E., Sirovich, L., Applied Mathematical Series, Springer-Verlag, New York, 1997.

Polak,E., On the Use of Consistent Approximations in the Solution of Semi-Infinite Optimization and Optimal Control Problems, *Math. Progr.*, Vol.62, pp.385-415, 1993.

Powell, M. J. D., A Fast Algorithm for Nonlinearly Constrained Optimization Calculations, *Numerical Analysis: Lecture Notes in Mathematics*, Matson, G. A., (Ed.), Springer-Verlag, Vol. 630, pp. 144-157, 1978.

Press, W.H., Teukolsky, S.A., W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd ed., Cambridge University Press, 1992.

Raash,C.R., Zajac, F.E., Baoming, M, Levine, W.S., Muscle coordination of maximum-speed pedaling, *Journal of Biomechanics*, Vol. 30, No. 6, pp. 559-602, 1997.

Reddy, J.M., *Applied Functional Analysis and Variational Methods in Engineering*, McGrall-Hill, 1986

Riener, R., Fuhr, T., Patient-Driven Control of FES Supported Standing Up: a Simulation Study, *IEEE Transactions on Rehabilitation Engineering*, Vol.6, No. 2, June 1998.

Schwartz, A. L., Polak, E. Chen Y., *RIOTS - Recursive Integration Optimal Control Trajectory Solver. A MATLAB toolbox for solving optimal control problems*. Version 1.0, Dept. of Electrical Engineering and Computer Science, University of California at Berkley, 1997.

Schwartz, A. L., Polak, E., Consistent Approximations for Optimal Control Problems Based on Runge-Kutta Integration, *SIAMM Journal of Control and Optimization*, Vol. 34, No. 4, pp.1235-1296, July, 1996.

Schwartz, A. L., *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*, Ph.D. dissertation, University of California at Berkley, 1996.

Sirovich, L., John, F., Eds., Applied Mathematical Series, Vol. 124, Springer-Verlag, New York, NY, 1997.

Spägle, T., Kistner, A., Gollhofer, A., Modelling, Simulation and Optimization of a Human Vertical Jump, *Journal of Biomechanics*, Vol. 32, pp. 521-530, 1999.

Spector, S.A., Gardiner, P.F., Zernicke, R.F., Roy, R.R., Edgerton, V. R., Muscle Architecture and the Force-Velocity Characteristics of the Cat Soleus and Medial Gastrocnemius: Implications for Motor Control, *Journal of Neurophysiology* Vol.44, No. 951, 1980.

Stein, R. B., Oguztoreli, M. N., Capaday, C., What is Optimized in Muscular Movements?, Jones, N. L., McCartney, N., McComas, A. J., eds., *Human Muscle Power*, Human Kinetics Publishers, Champaign IL, pp. 131-150, 1986.

Strogatz. S. H., *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry and Engineering*, Addison-Wesley Publishing Company, Reading, MA, 1994.

Tsirakos, D., Baltzopoulos, V., Bartlett, R., Inverse Optimization: Functional and Physiological Considerations Related to the Force-Sharing Problem, *Critical Reviews in Biomedical Engineering*, Vol. 25, No. 4&5, pp. 371-407, 1997.

White, A. J., *Análise Real: uma Introdução*, Ed. Edgard Blücher/Edusp, São Paulo, 1973.

Wickiewicz, T.L., Roy, R.R., Powell, P.L., Edgerton, V.R., Muscle Architecture of the Human Lower Limb, *Clinical Orthopaedics and Related Research*, No.179, pp.275-283, 1983.

Yamaguchi, G.T., Moran, D.W., Si, J., A Computationally Efficient Method for Solving the Redundant Problem in Biomechanics., *Journal of Biomechanics*, Vol.28, pp. 999-1005, 1995.

Zajac, F.E., Gordon, M.E., Determining Muscle's Force and Action in Multi-Articular Movement, *Exercise and Sport Science Reviews*, Pandolf, K. ed., Vol.17, pp. 187-230, Williams & Wilkins, Baltimore, 1989.

Zajac, F.E., Muscle and Tendon: properties, Models, Scaling and Application to Biomechanics and Motor Control, *CRC Critical Reviews in Biomedical Engineering*, Vol.17, No.4, pp. 359-411, 1989.

Zhou, J. L., Tits, A. L., Lawrence, C. T., User's Guide for FFSQP Version 3.7 : A Fortran Code for Solving Optimization Programs, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates, Institute for Systems Research, University of Maryland, Technical Report SRC-TR-92-107r5, College Park, MD 20742, 1997.