

**INSTITUTO MILITAR DE ENGENHARIA**

**OLAVO LUPPI SILVA**

**SOLUÇÃO DA DINÂMICA INVERSA DE MECANISMOS DE CADEIA  
FECHADA COM ATUADORES REDUNDANTES ATRAVÉS DE  
CONTROLE ÓTIMO**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Mecânica do Instituto Militar de Engenharia como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Mecânica.

Orientador:

Prof. Luciano Luporini Menegaldo, D.C.

Rio de Janeiro

2007

# **Livros Grátis**

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

Milhares de livros grátis para download.

c2007

INSTITUTO MILITAR DE ENGENHARIA  
Praça General Tibúrcio, 80 - Praia Vermelha  
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

S586s Silva, Olavo Luppi  
SOLUÇÃO DA DINÂMICA INVERSA DE MECANISMOS  
DE CADEIA FECHADA COM ATUADORES REDUNDANTES  
ATRAVÉS DE CONTROLE ÓTIMO/ Olavo Luppi Silva. - Rio de  
Janeiro: Instituto Militar de Engenharia, 2007.  
198: il, graf., tab.

Dissertação: (mestrado) - Instituto Militar de Engenharia-  
Rio de Janeiro, 2007.

1. Dinâmica Inversa. 2. Controle Ótimo.  
3. Atuadores Redundantes  
I. Título. II. Instituto Militar de Engenharia

CDD 621

INSTITUTO MILITAR DE ENGENHARIA

OLAVO LUPPI SILVA

SOLUÇÃO DA DINÂMICA INVERSA DE MECANISMOS DE CADEIA  
FECHADA COM ATUADORES REDUNDANTES ATRAVÉS DE  
CONTROLE ÓTIMO

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia Mecânica do Instituto Militar de Engenharia como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia Mecânica.

Orientador: Prof. Luciano Luporini Menegaldo, D.C.

Aprovada em 3 de agosto de 2007 pela seguinte Banca Examinadora:

---

Luciano Luporini Menegaldo, D.C. - Presidente

---

Agenor de Toledo Fleury, D.C. Universidade de São Paulo

---

Clodio Alberto Pastro Sarzeto , D.C. Instituto Militar de Engenharia

---

Jorge Audrin Morgado de Gois, Dr. - Ing. Instituto Militar de Engenharia

Rio de Janeiro  
2007



## AGRADECIMENTOS

Agradeço à CAPES pela bolsa de mestrado e à FAPERJ pelo programa de Auxílio à Pesquisa que possibilitou a aquisição de recursos computacionais utilizados na elaboração desta dissertação.

Agradeço aos meus pais, João Baptista e Rita Luppi, cujo apoio, incentivo e compreensão foram fundamentais para pavimentar os caminhos por onde ando.

Agradeço ao prof. Luciano Menegaldo que me orientou nesta dissertação pela sua competência, paciência e bom senso em fazer da sua orientação o ofício de fazer o orientado a descobrir e trilhar seus próprios caminhos.

Agradeço aos professores Fleury, Audrin e Sarzeto por terem aceitado o convite para compor a banca; em especial ao prof. Sarzeto pela boa vontade que teve em dirimir minhas dúvidas sobre dinâmica de sistemas multi-corpos representada através de Equações Diferenciais Algébricas e ao prof. Fleury com quem tive as primeiras lições sobre Controle Ótimo.

Agradeço a Dra. Adriana Thomaz, amiga, médica e confidente, por ter me assistido durante todo o período de elaboração deste trabalho, me socorrendo nos períodos mais difíceis, ora com remédios, ora com o seu carinho.

Agradeço a Amana da Matta, por ter compartilhado seu círculo de amigos no meu retorno ao Rio de Janeiro, tornando os momentos de lazer fonte de inspiração para semana de trabalho.

Agradeço aos colegas de mestrado do IME, em especial à Talita Sono, Marcelo de Oliveira, Mauricio Britto e Achille Arantes por terem acompanhado mais de perto o desenvolvimento deste trabalho e dado todo o suporte que me deram.

Agradeço aos voluntários do Girasohl, ONG em que atuei durante este mestrado, dedicada a cuidar de pacientes fora de possibilidade de cura no Hospital da Lagoa, pelas discussões e ensinamentos acerca da morte e terminalidade que permitiram lançar outros olhares para a vida.

Agradeço a Fernando Neder pela oportunidade de discutir e confrontar a visão da engenharia com a visão da dança a respeito da biomecânica, permitindo enriquecer a minha visão e expandir horizontes sobre o assunto.

Agradeço também a Chico Buarque por sua canção *Apesar de Você*, cujo refrão “Amanhã, vai ser outro dia...” me fazia ter certeza de que o que parecia ser impossível no dia de hoje, seria conseguido naturalmente no dia seguinte.

E por fim agradeço ao “café”, por toda a concentração que me proporcionou e sem o qual esta dissertação não teria sido possível.

## SUMÁRIO

LISTA DE ILUSTRAÇÕES .....	10
LISTA DE TABELAS .....	13
LISTA DE ABREVIATURAS E SÍMBOLOS .....	16
<b>1 INTRODUÇÃO .....</b>	<b>19</b>
1.1 Importância para a Biomecânica .....	20
1.2 Importância em Manipuladores robóticos e Máquinas Operatrizes .....	25
1.3 Objetivos .....	26
1.4 Metodologia .....	27
1.5 Organização dos capítulos .....	28
<b>2 SOLUÇÃO DA DINÂMICA INVERSA DO MECANISMO 4 BARRAS COM ATUADORES REDUNDANTES POR EQUAÇÕES DIFERENCIAIS AL- GÉBRICAS .....</b>	<b>31</b>
2.1 Introdução .....	31
2.1.1 Emprego do termo “coordenadas generalizadas” .....	32
2.1.2 Tipos de coordenadas .....	36
2.1.2.1 Coordenadas generalizadas .....	36
2.1.2.2 Coordenadas relativas .....	37
2.1.2.3 Coordenadas cartesianas .....	38
2.1.2.4 Coordenadas naturais .....	40
2.1.2.5 Coordenadas mistas .....	41
2.1.3 Equações de restrição .....	45
2.2 Equação de Movimento .....	48
2.2.1 Equações Diferenciais Algébricas (DAE's) .....	48
2.2.2 Índice de uma DAE .....	50
2.2.3 Redução de ordem de uma DAE .....	50
2.2.4 Integração de DAE transformada em ODE .....	51
2.3 Dinâmica inversa: método tradicional .....	53
2.3.1 Equacionamento .....	53
2.3.2 Implementação Computacional Dinâmica Inversa .....	61
2.3.3 Implementação Computacional Dinâmica Direta .....	63



2.4	Dinâmica inversa: método modificado para atuadores redundantes .....	68
2.4.1	Modificação 1: introdução de restrições diretoras adicionais ao vetor $\Phi$ .....	68
2.4.2	Modificação 2: introdução da matriz de transmissão $[T]$ .....	70
2.5	Análise dos Resultados .....	73
2.5.1	Análise do torque máximo dos atuadores.....	73
2.5.2	Análise do trabalho dos atuadores .....	76
2.5.3	Verificação dos resultados .....	78
<b>3</b>	<b>SOLUÇÃO DA DINÂMICA INVERSA DO MECANISMO 4 BARRAS COM ATUADORES REDUNDANTES POR CONTROLE ÓTIMO .....</b>	<b>83</b>
3.1	Formulação genérica do problema de controle ótimo .....	83
3.2	Formulação específica do OCP para o 4-barras com atuadores redundantes .....	85
3.3	Dedução do modelo dinâmico .....	88
3.3.1	Equação de movimento (lado esquerdo) .....	89
3.3.2	Equação de movimento (lado direito).....	92
3.3.2.1	Termo “Força Generalizada” .....	92
3.3.3	Equação de movimento na forma matricial .....	98
3.4	Discussão e histórico dos diferentes casos simulados .....	100
3.4.1	CASO 1: 3 atuadores .....	101
3.4.2	CASO 2: 3 atuadores com molas nas extremidades .....	104
3.4.3	CASO 3: 1 atuador .....	106
3.4.4	CASO 4: 1 atuador com molas nas extremidades .....	106
3.4.5	CASO 5: 1 atuador com mola e amortecedor nas extremidades .....	108
3.4.6	CASO 6: 2 atuadores (articulações A e B) com mola e amortecedor nas extremidades .....	110
3.4.7	CASO 7: 3 atuadores com mola e amortecedor nas extremidades .....	111
3.5	Análise dos Resultados .....	113
3.5.1	Função Objetivo .....	115
3.5.2	Seleção dos atuadores pelo critério do torque máximo.....	115
3.5.3	Análise do trabalho dos atuadores .....	116
3.5.3.1	Trabalho total .....	116
3.5.3.2	Trabalho individual e trabalho negativo .....	119
<b>4</b>	<b>COMPARAÇÃO ENTRE OS MÉTODOS E CONCLUSÕES FINAIS .....</b>	<b>123</b>
4.1	Comparação dos resultados equivalentes: DAE $\times$ OCP.....	124

4.2	Sugestão para trabalhos futuros . . . . .	127
4.3	Considerações Finais . . . . .	130
<b>5</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> . . . . .	<b>133</b>
	<b>APÊNDICES</b> . . . . .	<b>139</b>
<b>A</b>	<b>EOM2SS: DEDUÇÃO DA EQUAÇÃO DE MOVIMENTO</b> . . . . .	<b>140</b>
A.1	Introdução . . . . .	140
A.2	Descrição do algoritmo . . . . .	140
A.3	Diferenciação . . . . .	141
A.4	Agrupamento dos termos . . . . .	142
A.5	Variáveis de estado . . . . .	143
A.6	Output do programa . . . . .	143
A.7	Futuras implementações . . . . .	144
<b>B</b>	<b>TABELAS DE HISTÓRICO DA SOLUÇÃO DO OCP</b> . . . . .	<b>145</b>
<b>C</b>	<b>CÓDIGO FONTE</b> . . . . .	<b>153</b>
C.1	Dinâmica Inversa Sem Redundância . . . . .	153
C.2	Dinâmica Inversa Com Redundância . . . . .	161
C.3	Dinâmica Direta . . . . .	167
C.3.1	Programa principal . . . . .	167
C.3.2	<i>Function</i> de integração - sem redundância . . . . .	173
C.3.3	<i>Function</i> de integração - com redundância . . . . .	175
C.4	Algoritmos do OCP . . . . .	176
C.4.1	Algoritmo principal . . . . .	176
C.4.2	Equações de estado: EOM2SS . . . . .	181
C.4.3	Exportação de vetores e matrizes em linguagem C . . . . .	188
C.4.4	Vínculos terminais . . . . .	189
C.4.5	Visualização dos resultados . . . . .	190
C.4.6	Código em linguagem C para ser compilado . . . . .	193

## LISTA DE ILUSTRAÇÕES

FIG.1.1	Mecanismo plano de 8 barras, 3 GDL com duas cadeias cinemáticas fechadas que representa o modelo biomecânico da pedalada .....	23
FIG.1.2	Algoritmo da solução tradicional da dinâmica inversa utilizado em SILVA et al. (2005) .....	24
FIG.1.3	Exemplos de manipuladores retirados de VALASEK e SIKKA (2001) .....	27
FIG.1.4	Exemplos de máquinas ferramenta projetadas dentro do princípio de cadeia cinemática fechada .....	28
FIG.2.1	Exemplos de coordenadas generalizadas .....	37
FIG.2.2	Exemplos de coordenadas relativas em um mecanismo 4 barras .....	39
FIG.2.3	Exemplo de coordenadas cartesianas em um mecanismo 4 barras .....	40
FIG.2.4	Exemplo de coordenadas naturais em um mecanismo 4 barras .....	41
FIG.2.5	Exemplos de coordenadas mistas em um mecanismo 4 barras .....	43
FIG.2.6	Norma das restrições de posição, velocidade e aceleração calculadas ponto a ponto. Sem estabilização: $\alpha = 0$ e $\beta = 0$ . Com estabilização: $\alpha = 1$ e $\beta = 10$ . .....	53
FIG.2.7	Mecanismo 4 barras - coordenadas cartesianas .....	54
FIG.2.8	Representação gráfica das equações de restrição cinemática do vetor $\Phi$ . .....	57
FIG.2.9	Diagrama de corpo livre do mecanismo 4 barras .....	60
FIG.2.10	Torque na articulação A necessário para fazer o 4 barras percorrer uma volta completa, com $\dot{\phi}_1 = 60$ rpm calculado na dinâmica inversa .....	62
FIG.2.11	Curvas dos $\lambda$ 's calculados pela dinâmica inversa no mecanismo 4 barras com atuador simples .....	63
FIG.2.12	Curvas dos elementos do vetor $F^C$ calculados pela dinâmica inversa no mecanismo 4 barras com atuador simples .....	64
FIG.2.13	Comparação da violação das restrições de posição, velocidade e aceleração entre os resultados da Dinâmica Inversa e Dinâmica Direta (4 barras com 1 atuador). .....	65
FIG.2.14	Comparação dos resultados da cinemática obtidos com a Dinâmica Inversa e a Dinâmica Direta (mecanismo 4 barras 1 atuador): Variáveis angulares ( $\phi_i$ ). .....	66
FIG.2.15	Comparação dos resultados da cinemática obtidos com a Dinâmica Inversa e a Dinâmica Direta (mecanismo 4 barras 1 atuador): Variáveis	

	cartesianas ( $x_i$ e $y_i$ ). . . . .	67
FIG.2.16	Diagrama de blocos do primeiro algoritmo proposto para resolver a dinâmica inversa do 4 barras com atuadores redundantes a partir da formulação DAE. . . . .	70
FIG.2.17	Comparação dos torques calculados com a dinâmica inversa formulada a partir das DAE's para o mecanismo 4 barras: 1 e 2 atuadores . . . . .	74
FIG.2.18	Comparação dos torques calculados com a dinâmica inversa formulada a partir das DAE's para o mecanismo 4 barras: 3 e 4 atuadores . . . . .	75
FIG.2.19	Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz $[T]$ e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 2 atuadores. . . . .	80
FIG.2.20	Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz $[T]$ e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 3 atuadores. . . . .	81
FIG.2.21	Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz $[T]$ e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 4 atuadores. . . . .	82
FIG.3.1	Modelo dinâmico do pêndulo triplo . . . . .	90
FIG.3.2	Esquema do pêndulo triplo com restrições: forças de mola, amortecedor e torques de controle . . . . .	93
FIG.3.3	Pêndulo triplo “aberto” indicando a relação entre as coordenadas independentes ( $\phi$ ) e auxiliares $r$ . . . . .	96
FIG.3.4	Indicação de forças externas atuando no modelo do pêndulo triplo com restrições . . . . .	97
FIG.3.5	Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 1 . . . . .	102
FIG.3.6	Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 2 . . . . .	105
FIG.3.7	Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 4 . . . . .	107
FIG.3.8	Trajetória das variáveis de estado e de controle na melhor tentativa para	

	o CASO 5 .....	108
FIG.3.9	Comparação do torque U1 calculado pelo controle ótimo no CASO 5 com o torque calculado através da dinâmica inversa .....	111
FIG.3.10	Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 6 .....	112
FIG.3.11	Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 7 .....	113
FIG.3.12	Comparação dos torques nos casos em que foram utilizados 3 atuadores .....	117
FIG.3.13	Gráfico das coordenadas relativas $r_i$ em função do tempo para 1 ciclo do 4 barras. ....	118
FIG.3.14	Comparativo do trabalho total e individual dos atuadores ao longo de um ciclo completo. $W_1$ , $W_2$ e $W_3$ são os trabalhos dos atuadores nas articulações A, B e C, respectivamente e $W_{tot} = W_1 + W_2 + W_3$ .....	119
FIG.4.1	Visão geral dos resultados obtidos .....	125
FIG.4.2	Comparação dos torques para o 4 barras entre os resultados com formu- lação DAE e OCP. ....	128

## LISTA DE TABELAS

TAB.2.1	Comparativo das características entre os principais tipos de coordenadas (adaptada de (NIKRAVESH, 1988, p.13))	44
TAB.2.2	Comparação dos torques máximos e mínimos dos atuadores durante 1 ciclo completo do mecanismo 4 barras na formulação DAE (Valores em Nm).	76
TAB.2.3	Trabalho dos atuadores durante 1 ciclo completo do mecanismo 4 barras nos casos estudados com a formulação DAE	78
TAB.3.1	Identificação de forças externas e suas respectivas coordenadas da Figura 3.4	95
TAB.3.2	Mapa de casos estudados sobre o modelo 4 barras	100
TAB.3.3	Torques máximos e mínimos dos atuadores durante 1 ciclo completo do mecanismo 4 barras	116
TAB.3.4	Trabalho dos atuadores durante 1 ciclo completo do mecanismo 4 barras nos casos estudados com a formulação OCP	119
TAB.3.5	Tabela geral comparativa de parâmetros usados e resultados obtidos. PT = Pêndulo Triplo, 1A/2A/3A = 1, 2 ou 3 Atuadores	121
TAB.4.1	Comparação entre os principais resultados entre as formulações DAE e OCP para os casos com 2 e 3 atuadores	126
TAB.B.1	Tabela de simulações do CASO 1	147
TAB.B.2	Tabela de simulações do CASO 2	148
TAB.B.3	Tabela de simulações do CASO 4	149
TAB.B.4	Tabela de simulações do CASO 5	150
TAB.B.5	Tabela de simulações do CASO 6	151
TAB.B.6	Tabela de simulações do CASO 7	152

## RESUMO

Mecanismos de cadeia fechada com atuadores redundantes são encontrados em modelos biomecânicos e em aplicações industriais. Pode-se citar o exemplo do mecanismo plano de 8 barras e 3 graus de liberdade utilizado para descrever o movimento da pedalada humana. Neste tipo de modelo, os torques resultantes (fornecidos pelos músculos) em cada articulação, são entendidos como sendo “torques externos” que realizam trabalho sobre o sistema. Além disso, o uso de mecanismo de cadeia cinemática fechada em máquinas-ferramenta é vantajoso pois melhora a rigidez mecânica, a precisão de trajetória e de posicionamento, aumenta a capacidade de carga e reduz a massa de partes móveis. No entanto esta alternativa traz a desvantagem de aumentar o número de posições singulares dentro do seu espaço de trabalho, o que dificulta o controle. Este problema pode ser superado com a introdução de redundância de atuadores. Em ambos os casos, por haver mais atuadores do que graus de liberdade, a dinâmica inversa destes sistemas não possui solução única.

Este trabalho apresenta e compara duas alternativas para abordar esse problema. A primeira consiste na descrição de sistemas multi-corpos a partir de Equações Diferenciais Algébricas utilizada em conjunto com a matriz pseudo-inversa de Moore Penrose. A segunda se baseia na Teoria de Controle Ótimo para resolver o problema da dinâmica inversa com o auxílio do software RIOTS. Estuda-se um mecanismo 4 barras, com até 4 atuadores de torque no primeiro caso e até 3 atuadores no segundo, durante uma volta completa da manivela, a uma velocidade constante de 60 rpm. Os dois métodos apresentaram diferentes padrões de curvas de torque, sendo que a solução obtida através do controle ótimo obteve resultados superiores em relação aos três critérios de comparação adotados: integral da soma quadrática dos torques, torque máximo e trabalho dos atuadores.

## ABSTRACT

Redundantly actuated closed kinematic chain mechanisms can be identified in both biomechanical models and industrial applications. The planar, 8 bars, 3 degree of freedom mechanism used to describe human pedaling is referred as an example. In such models, net joint torques are supposed to be external torques that exert work on the system. Besides, machine-tools with closed chain mechanism have several advantages because increase mechanical stiffness, improve positioning and trajectory accuracy, have higher load capacity and smaller mobile mass. However this alternative has a drawback of increasing the number of singular points inside the workspace, which difficulties the control. This problem can be overcome with actuators redundancy. Anyway, inverse dynamics in both cases has no unique solution because the number of actuators is greater than the system's degrees of freedom.

This work develops and compares two alternatives to address this problem. In the first approach, the multi-body system is described by a set of Differential Algebraic Equations and solved by Moore Penrose pseudo-inverse matrix. The second, is based on Optimal Control Theory to solve the inverse dynamics problem with the aid of RIOTS software. A 4 bar mechanism with up to 4 torque actuators in the first case and up to 3 actuators in the second case, is studied during a complete crank cycle with a 60 rpm constant velocity. Both methods presented different torque patterns. The solution obtained by optimal control approach presented better results relating to all criteria of comparison adopted: integral of torque quadratic sum, maximum torque and actuators work.



## LISTA DE ABREVIATURAS E SÍMBOLOS

### ABREVIATURAS

EPS	-	Tolerância de otimização
EPSNEQ-	-	Tolerância das restrições
DAE	-	Equação Diferencial Algébrica
GDL	-	Grau de Liberdade
LSODA	-	integrador de passo variável
OCP	-	Optimal Control Problem
ODE	-	Equação Diferencial Ordinária
RK4	-	integrador Runge-Kutta de 4 <sup>a</sup> ordem (RK4) e passo fixo

### SÍMBOLOS

$m$	-	Massa da barra
$nb$	-	Número de corpos
$\mathbf{n}$	-	Vetor de atuadores redundantes introduzido na Seção 2.4.2
$\mathbf{q}$	-	Vetor de coordenadas cartesianas
$\dot{\mathbf{q}}$	-	Vetor de velocidade das coordenadas cartesianas
$\ddot{\mathbf{q}}$	-	Vetor aceleração das coordenadas cartesianas
$t$	-	Tempo
$t_f$	-	Tempo final
$u_0$	-	Estimativa do controle inicial
$\mathbf{u}$	-	Vetor de variáveis de controle
$x$	-	Coordenada horizontal
$\mathbf{x}$	-	Vetor de variáveis de estado
$y$	-	Coordenada vertical
$\mathbf{G}$	-	Matriz definida pelo agrupamento dos termos $\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}^A$
$C$	-	Constante de amortecimento do amortecedor
$K$	-	Constante de rigidez da mola
$L$	-	Lagrangeano
$L_1$	-	Comprimento da barra 1: manivela ou <i>crank</i>
$L_2$	-	Comprimento da barra 2: acoplador ou <i>coupler</i>
$L_3$	-	Comprimento da barra 3: seguidor ou <i>follower</i>

$L_4$	- Comprimento da barra fixa ou <i>ground</i>
$I$	- Momento de inércia
$\mathbf{I}$	- Matriz identidade
$\mathbf{M}_i$	- Matriz diagonal de massa do corpo $i$
$\mathbf{M}$	- Matriz diagonal de massa do sistema composta pelas matrizes de massa $\mathbf{M}_i$
$\mathbf{P}$	- Matriz definida pela justaposição da matriz $\Phi_{\mathbf{q}}^T$ e a matriz de Transmissão $\mathbf{T}$
$\mathbf{Q}_i$	- Vetor com todas as forças (internas e externas) que atuam sobre corpo $i$
$\mathbf{Q}$	- Vetor de forças composto por todos os vetores $\mathbf{Q}_i$
$\mathbf{Q}^A_i$	- Vetor com as forças ativas (que realizam trabalho) que atuam sobre corpo $i$
$\mathbf{Q}^A$	- Vetor de forças ativas composto por todos os vetores $\mathbf{Q}^A_i$
$T$	- Energia cinética
$\mathbf{T}$	- Matriz de Transmissão usada na Seção 2.4.2
$U_1$	- Torque obtido da solução OCP na articulação A
$U_2$	- Torque obtido da solução OCP na articulação B
$U_3$	- Torque obtido da solução OCP na articulação C
$V$	- Energia potencial
$W_1$	- Trabalho do atuador da articulação A ao longo de 1 ciclo do mecanismo
$W_2$	- Trabalho do atuador da articulação B ao longo de 1 ciclo do mecanismo
$W_3$	- Trabalho do atuador da articulação C ao longo de 1 ciclo do mecanismo
$W_i^f$	- Trabalho total realizado por forças externas sobre um sistema entre os instantes inicial e final
$W_{nc}$	- Trabalho realizado por forças não conservativas
$W_{tot}$	- Trabalho total dos atuadores ao longo de 1 ciclo do mecanismo
$X_A$	- Coordenada horizontal conhecida do vértice A
$X_B$	- Coordenada horizontal conhecida do vértice B
$Y_A$	- Coordenada vertical conhecida do vértice A
$Y_B$	- Coordenada vertical conhecida do vértice B
$\phi$	- Símbolo geral usado para ângulo
$\dot{\phi}$	- Símbolo geral usado para velocidade angular
$\ddot{\phi}$	- Símbolo geral usado para aceleração angular
$\nu$	- Termo relativo ao cálculo das velocidades, pela abordagem DAE, definido como $\equiv -\Phi_t$
$\gamma$	- Termo relativo ao cálculo das acelerações, pela abordagem DAE, definido como $\equiv -(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})_{\mathbf{q}} \cdot \dot{\mathbf{q}} - 2\Phi_{t\mathbf{q}}\dot{\mathbf{q}} - \Phi_{tt}$
$\hat{\gamma}$	- Vetor $\gamma$ modificado pela estabilização de Baumgarte

- $\lambda$  - Vetor de multiplicadores de Lagrange associado às forças internas na formulação DAE
- $\omega$  - Velocidade angular prescrita em uma restrição diretora
- $\Phi$  - Vetor de restrições
- $\Phi^D$  - Sub-vetor de  $\Phi$  com as restrições cinemáticas
- $\Phi^K$  - Sub-vetor de  $\Phi$  com as restrições diretoras
- $\Phi_q$  - Matriz Jacobiana do vetor de restrições  $\Phi$  em relação ao vetor de coordenadas  $q$
- $\Phi_t$  - Vetor coluna relativo à primeira derivada do vetor  $\Phi$  em relação ao tempo
- $\Phi_{tt}$  - Vetor coluna relativo a segunda derivada do vetor  $\Phi$  em relação ao tempo
- $\Phi_{qt}$  - Matriz cujos elementos são a derivada temporal da matriz Jacobiana  $\Phi_q$  em relação ao tempo.
- $\Phi_{tq}$  - Matriz cujos elementos são a derivada parcial do vetor  $\Phi_t$  em relação ao vetor  $q$
- $\Delta T$  - Variação da energia cinética
- $\Delta V$  - Variação da energia potencial (gravitacional, elástica, etc.)

## 1. INTRODUÇÃO

Este texto pretende abordar o tema da dinâmica inversa de mecanismos de cadeia cinemática fechada com redundância de atuadores, ou seja, mecanismos cujo número de atuadores é superior ao número de graus de liberdade (GDL). Existem diversos exemplos no campo da dinâmica aplicada à biomecânica e a manipuladores robóticos em que isso ocorre. Este problema será formulado aqui como um Problema de Controle Ótimo (*Optimal Control Problem* - OCP) e comparado com uma outra teoria já estabelecida baseada em Equações Diferenciais Algébricas (*Differential Algebraic Equations* - DAE<sup>1</sup>). Apesar de já existirem alguns trabalhos que se dediquem a resolver a dinâmica inversa com redundância de atuadores no contexto de manipuladores robóticos, o emprego da Teoria de Controle Ótimo para este tipo de problema parece ser inédito<sup>2</sup>.

Dentro deste tema, escolheu-se por trabalhar apenas com o mecanismo 4 barras que, além de ser extensamente conhecido e estudado, é um dos mecanismos de cadeia fechada mais simples que existe. Isto permitirá que os métodos desenvolvidos nesta dissertação possam servir de *benchmark* para aplicações futuras em mecanismos mais complexos. Fixou-se também uma cinemática que compreende uma volta completa da manivela do mecanismo a uma velocidade angular constante de 60 rpm ( $= 2\pi$  rad/s). O problema da dinâmica inversa, se resume neste caso a encontrar o(s) torque(s) que deve(m) ser aplicado(s) à(s) articulação(ões) do mecanismo para que este reproduza a cinemática imposta. Por possuir 1 GDL, a solução deste problema, quando se considera apenas 1 atuador, é rotineiramente descrita nos livros de dinâmica de sistemas multi-corpos. No entanto, a geometria do mecanismo 4 barras permite considerar até 4 atuadores de torque operando coordenadamente para fazê-lo girar. Desta forma, além do caso simples (com 1 atuador) a dinâmica inversa será resolvida para os casos com 2, 3 e 4 atuadores.

HAUG (1989) foi adotado como texto-base para o cálculo da dinâmica inversa do 4

---

<sup>1</sup>Nesta dissertação optou-se por utilizar as siglas referentes ao termo em inglês (ex. “OCP” ou “DAE” e o termo por extenso em português (ex. “Problema de Controle Ótimo” ou “Equações Diferenciais Algébricas”).

<sup>2</sup>Uma busca pela ocorrência simultânea dos termos “*inverse dynamics*”, “*optimal control*”, “*closed chain*” e “*redundant*” na base de dados Compendex ([www.engineeringvillage2.org](http://www.engineeringvillage2.org)) feita em 09/07/2007 não retorna nenhum resultado. A mesma busca feita no Google retorna apenas uma página sobre os tópicos de pesquisa do *Institut de Recherche en Communications et Cybernétique de Nantes - IRCCyN*. Mesmo assim este instituto parece empregar a teoria de controle ótimo apenas na solução de sistemas com estruturas flexíveis mas não para sistemas com redundância de atuadores.

barras simples onde se introduz a modelagem de sistemas multi-corpos a partir de Equações Diferenciais Algébricas. Com isso, a adaptação proposta para resolver o problema da redundância de atuadores foi implementada de forma coerente com o formalismo apresentado por este autor. A teoria de controle ótimo será utilizada para uma formulação alternativa do problema, adotando-se para a solução numérica o software RIOTS. Foge do escopo deste trabalho a formulação e solução das equações adjuntas e as advindas da Condição de Transversalidade que aparecem nos textos clássicos de controle ótimo, tais como CITRON (1969).

Desta forma, pode-se enunciar as questões centrais dessa dissertação como:

*Quais são os torques para que um mecanismo 4 barras acionado com e sem redundância de até 4 atuadores, operando em condição de regime permanente, complete 1 volta a uma velocidade constante de 60 rpm?*

*Considerando que a redundância de atuadores introduz a possibilidade de haver mais de uma solução para o problema da dinâmica inversa, como a solução advinda do controle ótimo se diferencia da solução obtida por outros métodos?*

Ao longo desta dissertação, em especial na dedução dos modelos dinâmicos, foi necessário adotar algumas hipóteses simplificadoras. As principais são listadas a seguir.

Todos os modelos considerados são de mecanismos planos (bidimensionais).

Desconsidera-se força de atrito e de resistência do ar.

Todas as barras são corpos rígidos, delgados e com centro de massa no ponto médio.

Desconsidera-se efeitos da natureza ou dinâmica do atuador.

A principal motivação para a realização deste trabalho foi a sua aplicação na área de biomecânica, em especial no modelo que descreve a pedalada. (SILVA et al. (2005); HULL e JORGE (1985)). No decorrer da pesquisa, verificou-se que a redundância de atuadores possui vantagens em manipuladores robóticos e máquinas operatrizes. A seguir, discorre-se sobre essas duas aplicações.

## 1.1 IMPORTÂNCIA PARA A BIOMECÂNICA

A redundância de atuadores está presente em toda parte em sistemas biológicos (LEE et al., 2002). Ela ocorre sempre que o número de “entradas” ou “controles” presentes no

modelo é superior ao número de graus de liberdade. No âmbito dos sistemas músculo-esqueléticos, a redundância pode ser vista de duas formas distintas: (i) quando se considera os músculos como atuadores; (ii) quando se considera que existe um atuador de torque “imaginário” presente em cada articulação cujo efeito representa o somatório dos momentos que cada músculo aplica em torno da articulação. No primeiro caso, isto pode ser verificado no modelo tridimensional de corpo humano descrito por HATZE (2000) que considera 17 segmentos, 42 graus de liberdade e 240 atuadores músculo-esqueléticos. No segundo caso, dependendo do tipo de movimento considerado, pode haver redundância ou não. Por exemplo, no modelo para salto de trampolim de BLAJER e CZAPLICKI (2001) o número de atuadores é menor do que o número de GDL e nos modelos de postura humana utilizados por MENEGALDO e WEBER (1997), BARIN (1989) e CAHOUEY et al. (2002) o número de atuadores é igual ao número de GDL. Já nos modelos de pedalada (HULL e JORGE, 1985), de remo (LEE et al., 2005) e da fase de duplo apoio na marcha (PANDY e BERME, 1988), nos quais ocorre uma condição de cadeia cinemática fechada, o número de atuadores de torque supera o número de GDL. Em suma, a redundância em biomecânica é um problema desafiador pois, tal como HATZE (2000) afirma, não possui solução única.

SILVA et al. (2005) se depararam com esta situação ao aplicar uma abordagem tradicional para resolver a dinâmica inversa da pedalada. Nesta abordagem, descrita por WINTER (1990), é utilizado um diagrama de corpo livre e as forças internas e externas são identificadas em cada segmento. Deve-se também medir a cinemática de um número de coordenadas igual ao número de GDL do sistema e as forças que atuam no membro mais distal da cadeia cinemática<sup>3</sup>. Desta forma é possível encontrar as forças e momentos internos resolvendo-se recursivamente as equações de Newton-Euler do membro mais distal para o mais proximal. Um exemplo da utilização desta abordagem, também conhecida como *bottom-up*, na análise de marcha pode ser encontrada no site: [www.univie.ac.at/cga/teach-in/inverse-dynamics.html](http://www.univie.ac.at/cga/teach-in/inverse-dynamics.html).

A abordagem *bottom-up* rotineiramente usada no campo da biomecânica se baseia em métodos vetoriais na descrição de sistemas multi-corpos, cuja origem remonta aos tempos de Isaac Newton e Leonard Euler. Posteriormente, d’Alembert, considerando sistemas de corpos rígidos com restrições (cinemáticas) fez a distinção entre *forças aplicadas* e *forças de reação*. Uma formulação consistente do princípio de d’Alembert é atribuída a Joseph-Louis Lagrange que combinou a idéia fundamental de d’Alembert com o princípio do

---

<sup>3</sup>Na análise de marcha essa informação é registrada por meio de uma plataforma de força sobre a qual o indivíduo deve caminhar. Já no problema da pedalada costuma-se montar *strain-gauges* sobre os pedais para medir as forças normal e tangencial de reação (HULL e DAVIS (1981), BOYD et al. (1996))

trabalho virtual (SCHIEHLEN, 1997). DOUGHTY (1988, p.222) enuncia este princípio da seguinte maneira:

*“O Princípio do Trabalho Virtual para um sistema ideal<sup>4</sup> de corpos rígidos diz que este estará em equilíbrio desde que o trabalho virtual das **forças externas** ao sistema seja zero.”*

A distinção entre força aplicada (ou externa, na citação de DOUGHTY (1988)) e força de reação constatada por d’Alembert foi fundamental para enxergar o Princípio do Trabalho Virtual de uma outra forma quando o sistema não está em equilíbrio (i.e. está se movendo), tal como aparece em (ROBERSON e SCHWERTASSEK, 1988, p.166):

*“Um sistema mecânico sujeito a restrições realiza movimentos de tal forma que o trabalho virtual total das **forças e torques de restrição** seja zero.”*

O princípio do trabalho virtual deu origem a uma série de métodos escalares para a solução de problemas cuja natureza pode ser tanto de equilíbrio (Método dos Trabalhos Virtuais) quanto dinâmica (Equações de Lagrange e de Jourdain<sup>5</sup>) (MOON, 1998, p.42). Apesar da abordagem vetorial (Newtoniana) do estudo da dinâmica se mostrar mais natural e intuitiva, os métodos escalares, por não envolverem forças internas e de reação, permitem que o número de equações seja enormemente reduzido (DOUGHTY, 1988, p.217).

À luz dessa discussão, considere por exemplo o mecanismo plano de 8 barras e 3 GDL mostrado na Figura 1.1 que representa o modelo biomecânico da pedalada<sup>6</sup>. A abordagem tradicional para resolver a dinâmica inversa deste problema biomecânico requer que as forças de reação normal e tangencial dos pedais sejam medidas (HULL e JORGE (1985), REDFIELD e HULL (1986)). Isto representa “cortar” as duas cadeias cinemáticas fechadas nos pontos indicados na Figura e medir as forças internas. Portanto, estas forças não precisariam ser medidas se fosse adotada uma abordagem escalar para a formulação das equações de movimento. Neste caso seria necessário medir apenas o torque resistivo

---

<sup>4</sup>Sistema ideal é um sistema de corpos rígidos no qual não há absorção de energia nos seus pontos de interconexão.

<sup>5</sup>Também conhecido como Método das Potências Virtuais

<sup>6</sup>A suposição da descrição do movimento da pedalada se dar momentaneamente no plano sagital foi testada por UMBERGER e MARTIN (2001) que concluiu que este modelo é adequado para fins de análise de dinâmica inversa.

em torno do pé de vela, além de três ângulos (em geral o do pé de vela e os dos tornozelos) para determinar a configuração cinemática do sistema. Os demais torques externos<sup>7</sup> considerados neste modelo são os torques resultantes nas articulações do tornozelo, joelho e quadril devido à ação combinada dos músculos que atravessam essas articulações. Justamente por não ser possível medir esses torques diretamente, eles são as incógnitas do problema da dinâmica inversa.

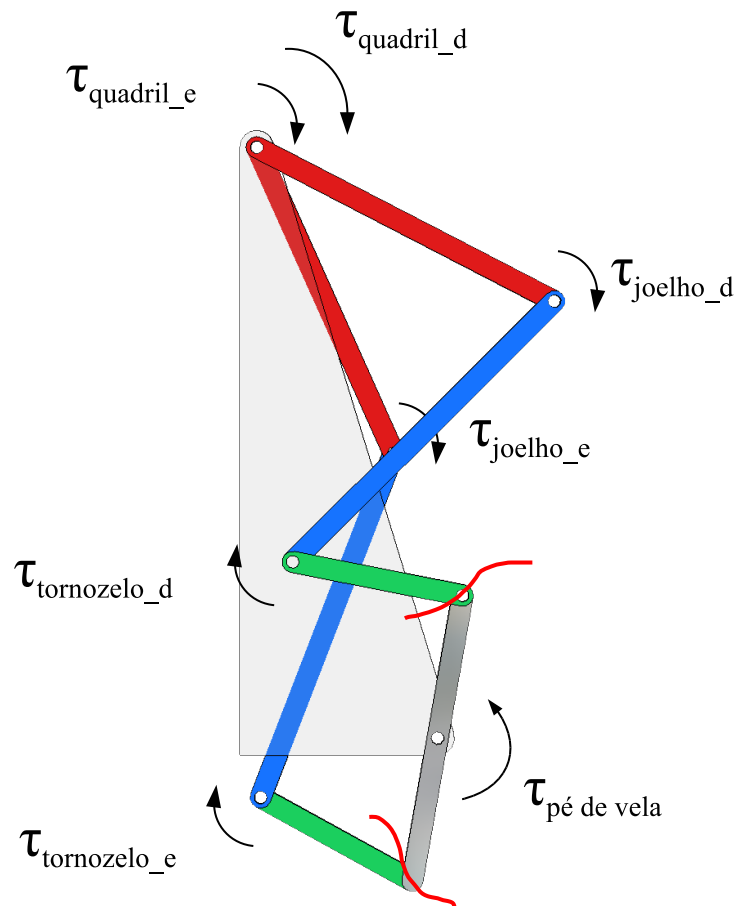


FIG. 1.1: Mecanismo plano de 8 barras, 3 GDL com duas cadeias cinemáticas fechadas que representa o modelo biomecânico da pedalada

Este modelo possuiria a princípio 6 incógnitas, mas por adotar uma abordagem escalar na dedução das equações de movimento, só estariam disponíveis 3 equações (devido aos 3 GDL do sistema); o que torna o problema indeterminado. Isto significa que para uma mesma cinemática imposta ou fixada (digamos, um ciclo completo da pedalada), haveria mais de um conjunto de curvas de torque no tornozelo, joelho e quadril que reproduziriam esta cinemática. Na abordagem vetorial-Newtoniana, haveria 21 equações: 3 por corpo,

<sup>7</sup>Nesta dissertação usa-se o termo “força externa” ou “torque externo” para se referir a forças ou torques que realizam trabalho sobre o sistema.



relativas ao somatório de forças horizontais, verticais e de momento. Uma vez calculadas as acelerações dos centros de massa de todos os corpos, e conhecendo-se as forças de reação no pedal, sobram como incógnitas as forças de reação no tornozelo. Analogamente, conhecendo-se as reações no tornozelo, calculam-se as reações no joelho, procedendo-se da mesma forma para as reações no quadril. Uma vez que todas as reações internas sejam conhecidas, elas podem ser substituídas nas equações de somatório de momentos. Isto levará a um sistema linear de solução única para cada posição do mecanismo. Este algoritmo foi implementado em SILVA et al. (2005) e aparece esquematizado na Figura 1.2.

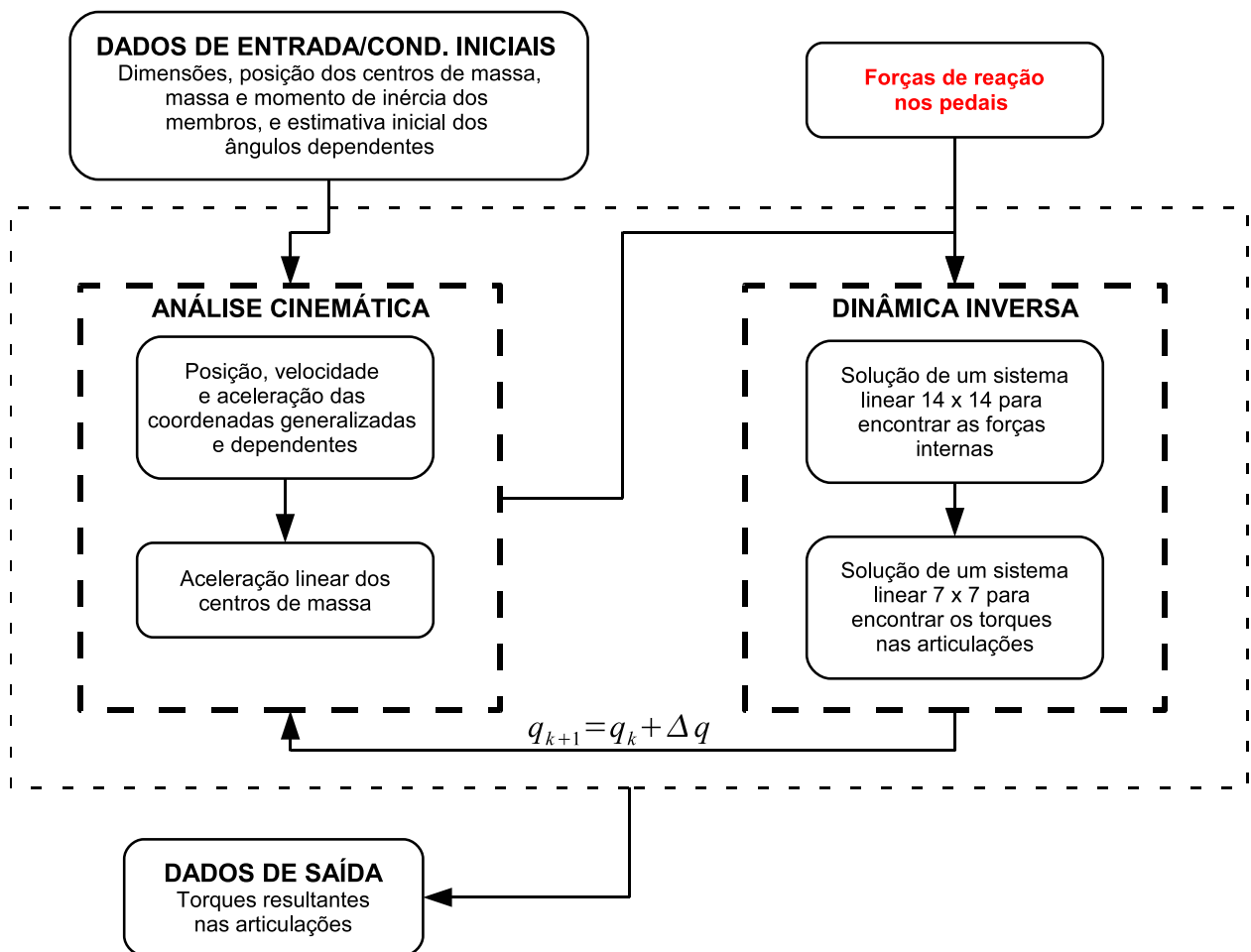


FIG. 1.2: Algoritmo da solução tradicional da dinâmica inversa utilizado em SILVA et al. (2005)

Pode-se alegar que a utilização dessa abordagem em problemas biomecânicos é salutar, pois permite individualizar as respostas obtidas. Uma vez que cada um pedala de uma maneira diferente, as curvas de reação nos pedais também seriam diferentes bem como os torques resultantes nas articulações. Além disso, também seria possível avaliar eventuais assimetrias na pedalada quando o ciclista possui uma perna mais forte do que a outra. No

entanto, ao enxergar o problema da dinâmica inversa como um problema de otimização, não se deseja criar modelos que possam ser individualizados para cada paciente ou atleta. Mas sim, revelar princípios de ótimo escondidos em tarefas relacionadas ao controle motor. Quanto dos torques articulares de cada indivíduo estaria escondido na informação das curvas de reação nos pedais? Estaria esta informação balizada ou regrada por algum princípio de otimização? Um outro problema que surge ao tratar a dinâmica inversa da pedalada como um problema de otimização é poder tratar o torque resistivo no pé de vela como uma incógnita. Ao invés de mensurá-lo, este torque seria obtido como resultado da própria dinâmica inversa. Que implicações isso teria e como isso modificaria os resultados? Se for possível encontrar uma função objetivo que, quando minimizada, leve aos mesmos resultados da dinâmica inversa tradicional, seria possível simplificar e baratear o *setup* experimental para avaliações biomecânicas da pedalada, prescindindo das medições de força ou torque.

## 1.2 IMPORTÂNCIA EM MANIPULADORES ROBÓTICOS E MÁQUINAS OPERATRIZES

Mecanismos de cadeia fechada têm experimentado uma crescente utilização em aplicações industriais e robóticas, como por exemplo em LIU et al. (2004) e LEE et al. (2002). Nestes casos é usual se referir a eles como Manipuladores Paralelos ou Manipuladores de Cadeia Fechada (*Parallel e Closed Chain Manipulators*, respectivamente). Este tipo de mecanismo, especialmente quando é usado para acionar manipuladores robóticos, possui diversas vantagens sobre seus equivalentes de cadeia aberta, tais como: alta rigidez mecânica (MILLER, 2001), alta precisão de trajetória (NAKAMURA e GHODOUSSI, 1989) e de posicionamento (DASGUPTA e MRUTHYUNJAYA, 1998), alta capacidade de carga (DASGUPTA e MRUTHYUNJAYA, 1998) e partes móveis de massa reduzida (MILLER, 2001).

Estas características fazem com que esse tipo de mecanismo seja preferido em máquinas de soldagem, corte e selagem e vedação (NAKAMURA e GHODOUSSI, 1989). Além disso, a existência de singularidades cinemáticas é um problema crucial na análise de mecanismos de cadeia fechada. Este problema se revela, por exemplo, quando o mecanismo atinge os limites do seu espaço de trabalho, condição esta que, segundo LIU et al. (2001) deve ser evitada na maioria dos casos. Quando um manipulador paralelo se move em direção a uma configuração singular, suas propriedades de rigidez e precisão se deterioram rapidamente (CHENG et al., 2003). Além do mais, a dinâmica de mecanismos de

cadeia fechada é mais complicada e não tem sido tão amplamente estudada quanto seus equivalentes de cadeia aberta (NAKAMURA e GHODOUSSI, 1989).

LIU et al. (2001) se dedicaram a estudar os efeitos indesejáveis da singularidade em manipuladores paralelos. Para solucionar esses efeitos, os autores sugerem introduzir três tipos de redundância ao manipulador:

Redundância cinemática: o número de GDL do manipulador deve ser maior do que o número de GDL do *end-effector*.

Redundância de restrições: aumentar o número de braços aumentando também o número de cadeias cinemáticas fechadas

Redundância de atuadores: colocar um atuador em cada um dos “braços” adicionais fazendo com que seu número supere o número de GDL do *end-effector*.

CHENG et al. (2003) concordam que a redundância de atuadores pode ser uma maneira eficaz para eliminar as singularidades no projeto de manipuladores paralelos, e desta forma melhorar sua *performance*. A Figura 1.3 mostra dois exemplos de como é possível modificar a estrutura de um manipulador plano de três graus de liberdade para que possa dispor de redundância de atuadores. VALASEK et al. (2004), apresentam um novo conceito no projeto de máquina-ferramenta baseado em mecanismos de cadeia fechada com redundância de atuação. Este conceito deu origem ao centro de usinagem TRI-JOINT 900H<sup>8</sup> mostrado na Figura 1.4(b). Ao introduzir mais atuadores do que GDL do *end-effector*, os efeitos de flexão na base de suporte da ferramenta são substituídos por tração/compressão, a massa das partes móveis é reduzida e a presença de folgas e demais imprecisões, comuns em cadeias cinemáticas abertas, são bastante reduzidas.

### 1.3 OBJETIVOS

O objetivo geral desta dissertação é desenvolver uma metodologia para o cálculo da dinâmica inversa de um mecanismo de cadeia fechada acionado por atuadores redundantes através da técnica de Controle Ótimo. Como objetivos específicos pode-se citar:

Estudar e fazer a implementação computacional de algoritmos para a análise cinemática, dinâmica inversa e direta baseados em formalismos para sistemas multi-corpos.

---

<sup>8</sup><http://www.kovosvit.cz/trijoint/english/eindex.php>

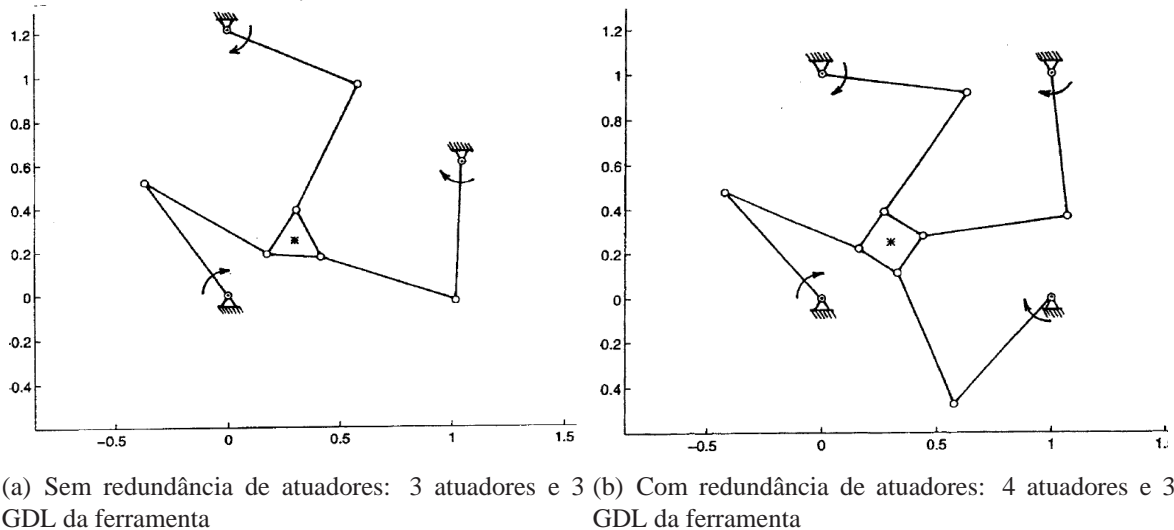


FIG. 1.3: Exemplos de manipuladores retirados de VALASEK e SIKKA (2001)

Propor modificação dos algoritmos estudados para que possam tratar do problema com redundância, visto que a teoria que os fundamenta, em geral não leva em conta esse caso particular.

Implementar um outro método para a solução do problema da redundância já proposto e estudado por outro autor tais como os que são apresentados por VALASEK e SIKKA (2001), CHENG et al. (2003), LEE et al. (2002), LEE et al. (2005), NAKAMURA e GHODOUSSI (1989) e GANOVSKI et al. (2004).

Verificar as diferenças dos resultados apresentados entre o Controle Ótimo e os demais métodos apresentados para calcular a dinâmica inversa de mecanismos de cadeia fechada com redundância de atuadores.

#### 1.4 METODOLOGIA

Esta dissertação se baseia essencialmente na apresentação e comparação de dois métodos para resolver um mesmo problema. Para o primeiro método, apresentado no Capítulo 2, adotou-se o livro de HAUG (1989) como bibliografia básica. Portanto, a maior parte da nomenclatura e simbologia adotada nesta dissertação se baseou nesta obra. Este livro apresenta a teoria relativa à análise cinemática e dinâmica de sistemas mecânicos multi-corpos dentro do contexto das Equações Diferenciais Algébricas. Outros textos que também abordam o assunto dentro desta ótica foram utilizados como apoio, tais como HAUG (1992), GARCÍA DE JALÓN e BAYO (1994) e STEJSKAL e VALASEK (1996).



(a) Máquina ferramenta de 5 eixos com manipulador paralelo (LIU et al., 2004)



(b) Centro de usinagem TRIJOINT 900H

FIG. 1.4: Exemplos de máquinas ferramenta projetadas dentro do princípio de cadeia cinemática fechada

O segundo método, apresentado no Capítulo 3, partiu da tese de doutorado de MENEGALDO (2001) para adaptar o problema de controle ótimo apresentado naquele texto para o problema aqui proposto. Os primeiros testes se basearam no modelo de pêndulo triplo adotado por MENEGALDO (2001) para o problema da postura humana. Posteriormente a rotina EOM2SS, apresentada no Apêndice A, foi escrita para tornar o programa originalmente desenvolvido por MENEGALDO (2001) mais geral e modular. Com ela, juntamente com o algoritmo apresentado na Seção C.4.1, será possível resolver OCP's usando o software RIOTS para sistemas de cadeia aberta com  $n$  corpos, fazendo apenas pequenas alterações no código fonte.

## 1.5 ORGANIZAÇÃO DOS CAPÍTULOS

Esta dissertação está organizada da seguinte forma: No Capítulo 1 apresenta-se e delimita-se o tema da dissertação, mostram-se os elementos que justificam a importância desta pesquisa, bem como a metodologia utilizada e os objetivos desta dissertação.

O Capítulo 2 inicia-se com uma revisão de conceitos relacionados à modelagem de sistema multi-corpos tais como: coordenada generalizada, tipos de coordenadas e equações de restrição. Depois, a dedução de equações de movimento através de Equações Diferenciais Algébricas é apresentada ao leitor, seguida da implementação dos algoritmos para

dinâmica inversa e direta de mecanismos sem redundância de atuadores. O Capítulo segue com proposição de uma modificação à teoria original para levar em conta a redundância de atuadores e da implementação do método da Matriz de Transmissão (VALASEK e SIKKA, 2001) para a solução da dinâmica inversa deste problema. No final do Capítulo, analisam-se os resultados obtidos.

No Capítulo 3 formula-se a mesma questão como um Problema de Controle Ótimo. Propõe-se um novo modelo dinâmico mais adequado para a implementação computacional e sua equação de movimento é deduzida. Em seguida estudam-se sete variações do OCP e analisa-se os resultados obtidos.

A dissertação se encerra no Capítulo 4 com a comparação entre os resultados obtidos nos Capítulos 2 e 3, apresentam-se as sugestões para trabalhos futuros e por fim as considerações finais.



## 2. SOLUÇÃO DA DINÂMICA INVERSA DO MECANISMO 4 BARRAS COM ATUADORES REDUNDANTES POR EQUAÇÕES DIFERENCIAIS ALGÉBRICAS

### 2.1 INTRODUÇÃO

A Análise Cinemática consiste no estudo do movimento de um sistema multicorpos independentemente das forças que o produzem. Tipicamente, em mecanismos de cadeia fechada, prescreve-se a posição, velocidade e aceleração de uma ou mais coordenadas e calculam-se estas grandezas para as coordenadas restantes. Para o cálculo de posição, deve-se resolver um sistema de equações não lineares, enquanto que para a determinação da velocidade e aceleração resolve-se um sistema de equações algébricas expresso na forma de uma equação linear do tipo  $[A][x] = [b]$ .

A Análise Dinâmica de um sistema mecânico, também chamada de Dinâmica Direta, se preocupa com o estudo do movimento devido à ação das forças que são aplicadas a ele. Neste tipo de problema deseja-se encontrar a posição, velocidade e aceleração de todos os corpos de um sistema mecânico uma vez que se conheça (ou se prescreva) o histórico das forças que atuam sobre o mesmo. Esse tipo de problema, dependendo da forma como se escolhe as coordenadas para descrever o sistema, pode levar a equações diferenciais ou a uma combinação de equações diferenciais com equações algébricas. Seja num caso ou no outro, essas equações são de difícil solução algébrica e em geral são resolvidas através de métodos numéricos.

Já na Dinâmica Inversa, como o próprio nome diz, deseja-se o contrário da Dinâmica Direta, isto é, uma vez que se conheça a posição, velocidade e aceleração de todos os corpos, deve-se encontrar as forças responsáveis por tal movimento. Em geral, para encontrar tais forças é necessário resolver simultaneamente as equações da análise cinemática e usar as informações de posição, velocidade e aceleração de todos os corpos nas equações de movimento de forma a encontrar as forças e/ou torques. Este tipo de problema leva a um sistema de equações algébricas que possui solução única quando o número de atuadores é igual ao número de graus de liberdade do sistema.

Este Capítulo dedica-se a apresentar os fundamentos teóricos necessários para resolver a dinâmica inversa de um mecanismo plano de cadeia fechada (no caso, o 4 barras) através da formulação das Equações Diferenciais Algébricas. Os princípios para a elaboração de um algoritmo que faça a Dinâmica Direta deste mecanismo também serão apresentados



como forma de verificar os resultados obtidos com a Dinâmica Inversa. Primeiramente, apresentam-se estas duas abordagens para o caso regular em que se considera o mecanismo de 4 barras com apenas 1 atuador. Depois, uma modificação na teoria original é proposta de forma a permitir o cálculo da dinâmica inversa deste mesmo mecanismo com até 4 atuadores de torque. Por fim, faz-se a simulação (Dinâmica Direta) do mecanismo 4 barras submetido aos torques dos atuadores redundantes para verificar os resultados obtidos com o algoritmo de dinâmica inversa modificado.

### 2.1.1 EMPREGO DO TERMO “COORDENADAS GENERALIZADAS”

Um conceito fundamental que aparece em toda a literatura de sistemas multi-corpos são as *coordenadas* utilizadas para se descrever a configuração de um determinado sistema. Antes de entrar em detalhes quanto aos diferentes tipos de coordenadas, cabe aqui fazer uma revisão sobre a utilização do termo *Coordenada Generalizada*. Apesar de ser amplamente empregado, a definição deste termo não é muito precisa. Alguns autores chegam, inclusive, a apresentar definições que se contradizem, principalmente com relação à dependência ou independência das coordenadas. DOUGHTY (1988), por exemplo, na página 4, define coordenada generalizada da seguinte maneira:

“A set of coordinates the is (1) *complete* in that they are sufficient to specify the system position fully and (2) each *independent* of the others is called a set of *generalized coordinates*. Generalized coordinates may include rectangular Cartesian coordinates, angles, arc lengths, or any other measures that contribute to the description of the system configuration, subject to the requirement that they form a complete, independent set. For the present, the number of generalized coordinates should be considered equal to the number of degrees of freedom.”

Este autor define coordenada generalizada como um conjunto de coordenadas que seja *completo*, no sentido em que este seja suficiente para descrever completamente a posição de um sistema e *independente* uma das outras. E ainda afirma que o número de coordenadas generalizadas deve ser igual ao número de graus de liberdade do sistema. STEJSKAL e VALASEK (1996), na página 260, vão no mesmo caminho:

“Independent coordinates, due to their significance for the formulation of equations of motion, are called generalized coordinates in analytical mechanics. These coordinates would be obtained from physical coordinates by a simple

choice, in which case they are called simple coordinates. Generalized coordinates with a general dependance on physical coordinates are called combined coordinates.”

Eles dizem que devido à importância que as coordenadas independentes têm para a formulação das equações de movimento, estas são chamadas de coordenadas generalizadas dentro da mecânica analítica. Depois, dão definições vagas para os termos *simple coordinates* e *combined coordinates* mas não os utiliza no decorrer do livro. NIKRAVESH (1988) também não apresenta uma definição formal de coordenada generalizada mas deixa claro, na página 10, que o número de coordenadas generalizadas é sempre igual ao número de graus de liberdade.

O livro de GOLDSTEIN (1970, p.11), considerado um clássico no estudo da mecânica e citado por muitos autores, aborda o tema da seguinte forma:

“In the case of holonomic constraints, the first difficulty<sup>9</sup> is solved by the introduction of *generalized coordinates*. So far we have been thinking implicitly in terms of cartesian coordinates. A system of  $N$  particles, free from constraints has  $3N$  independent coordinates or *degrees of freedom*. If there exist holonomic constraints, expressed in  $k$  equations in the form (1-35)<sup>10</sup>, then we may use these equations to eliminate  $k$  of the  $3N$  coordinates, and we are left with  $3N - k$  independent coordinates, and the system is said to have  $3N - k$  degrees of freedom. This elimination of the dependent coordinates can be expressed in another way, by the introduction of the new,  $3N - k$ , independent variables<sup>11</sup>  $q_1, q_2, \dots, q_{3N-k}$  in terms of which the old coordinates  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$  are expressed by equations of the form

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{r}_1(q_1, q_2, \dots, q_{3N-k}, t) \\ &\vdots \\ \mathbf{r}_N &= \mathbf{r}_N(q_1, q_2, \dots, q_{3N-k}, t) \end{aligned} \tag{1-36}$$

containing the constraints in them implicitly. These are *transformation* equations from the set of  $(\mathbf{r}_i)$  variables to the  $(q_i)$  set, or alternatively Eqs. (1-36) can be considered as parametric representation of the  $(\mathbf{r}_i)$  variables.”

---

<sup>9</sup>A dificuldade citada se refere ao fato das coordenadas não serem independentes, o que dificulta o problema mecânico. N.A.

<sup>10</sup>Equação (1-35):  $f(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, t) = 0$

<sup>11</sup>O contexto deixa claro que é esse conjunto de variáveis  $q_1, q_2, \dots$  que o autor chama de coordenada generalizada

O trecho anterior à citação faz referência à existência de restrições (*constraints*), que podem ser holonômicas ou não-holonômicas, em sistemas mecânicos. A presença dessas restrições introduz duas dificuldades. A primeira é que as coordenadas (cartesianas) que descrevem o sistema sujeito a equações de restrição deixam de ser independentes entre si. Seriam independentes, por exemplo, se fosse considerado um sistema de partículas. A segunda dificuldade é a de que as forças de restrição não são conhecidas *a priori*. É dentro desse contexto que GOLDSTEIN (1970) faz a sua introdução ao conceito de coordenada generalizada, de forma a vir resolver o problema dependência das coordenadas. Ele mostra que é possível usar *Equações de Transformação* (identificadas na citação com o número (1-36)), que relacionam um conjunto independente de coordenadas ( $q_l$ ) com um conjunto ( $\mathbf{r}_1$ ) de coordenadas dependentes.

Por outro lado, GREENWOOD (1965) nas páginas 231 e 232 afirma o contrário dos demais autores:

“No specific set of coordinates is uniquely suited to the analysis of a given mechanical system. Many coordinates are possible; in fact there is an infinite number. (...) But in any case, the number of coordinates is equal to the number of degrees of freedom **plus**<sup>12</sup> the number of independent equations of constraint.

(...) Any set of numbers<sup>13</sup> which serve to specify the configuration of a the system are examples of *generalized coordinates*. (...) Note that the term *generalized coordinates* can refer to any of the commonly used coordinate systems<sup>14</sup>, but it can also refer to any of an infinite variety of other sets of parameters which serve to specify the configuration of a system.

... it will become apparent that in many instances the mathematical analysis of a dynamical system is simplified by choosing a set of *independent* generalized coordinates.”

Primeiro, GREENWOOD (1965) começa dizendo que qualquer conjunto de coordenadas que represente um sistema de forma única é aceitável. E que em qualquer caso, o número de coordenadas necessárias para fazer essa representação deve ser igual ao número de GDL do sistema somado ao número de equações de restrição independentes. Depois afirma que qualquer um desses conjuntos aceitáveis de coordenadas são exemplos

---

<sup>12</sup>Grifo do autor

<sup>13</sup>Número de coordenadas. N.A.

<sup>14</sup>Cartesiano, cilíndrico, esférico, etc. N.A

de coordenadas generalizadas. No final ele admite que a análise matemática de sistemas dinâmicos é simplificada quando se escolhe um conjunto “independente” de coordenadas generalizadas.

HAUG (1989), ao definir o termo na página 49, segue a mesma linha de GREENWOOD (1965):

“Any set of variables that uniquely specifies the position and the orientation of all bodies in a mechanism, that is, the *configuration* of the mechanism, is called a set of *generalized coordinates*. Generalized coordinates may be *independent* (i.e., free to vary arbitrarily) or *dependent* (i.e., required to satisfy equations of constraint).”

Ele faz uma afirmação explícita de que as coordenadas generalizadas podem ser tanto independentes quanto dependentes e que qualquer conjunto de variáveis que especifique de forma única a posição e orientação de todos os corpos de um mecanismo, pode ser chamado de coordenada generalizada. Isto é demonstrado no Exemplo 2.4.3 (págs. 35 e 36) referente a um mecanismo biela-manivela. Na página 14, HAUG (1989) ainda cita outros dois termos: *Cartesian generalized coordinate* e *independent Lagrangian generalized coordinate*. O primeiro termo, pelo que tudo indica, parece se tratar simplesmente do que chamaremos de *coordenada Cartesiana* e que será explicado na Seção 2.1.2.3. Já o segundo, não foi possível encontrar alguma menção a este termo. NIKRAVESH (1988) chega a se referir ao termo “Lagrangian coordinate” e o distingue de “Cartesian coordinate” na página 38 dizendo que o primeiro termo permite uma definição da posição de um corpo relativo a um sistema de coordenadas móvel, enquanto que o segundo normalmente requer que a posição de cada corpo no espaço seja definida em relação a um sistema de coordenadas global fixo. Uma apresentação um pouco mais alongada sobre Cartesian e Lagrangian coordinates é apresentada em DABERKOW e KREUZER (1999).

ROBERSON e SCHWERTASSEK (1988, p.89) reconhecem a ambigüidade no uso do termo coordenada generalizada e citam dois autores que propõem utilizar o termo coordenada Lagrangeana para se referir a conjuntos de coordenadas dependentes e deixar o termo coordenada generalizada para coordenadas e velocidades independentes.

“In the dynamics literature one often uses the term generalized coordinates to suggest that coordinates other than Cartesian may be used to represent system motion. Often the term is used both for independent or interrelated sets of variables. To avoid this ambiguity it has been proposed, for example, in PARS

(1968) and PAUL (1979), to use the term *Lagrangian coordinates* when referring to any sets of interrelated variables and to reserve the term generalized coordinates and velocities (or state variables) for independent sets.”

## 2.1.2 TIPOS DE COORDENADAS

Dentro da polêmica do emprego do termo *Coordenada Generalizada*, optou-se por utilizar nesta dissertação a definição empregada por DOUGHTY (1988) na citação apresentada na página 32. Esta definição está de acordo, ou é complementada, pelas definições de STEJSKAL e VALASEK (1996), GOLDSTEIN (1970) e NIKRAVESH (1988). Dentro deste contexto, e com o devido esclarecimento com relação à problemática da nomenclatura, faz-se necessário apresentar os principais tipos de coordenadas existentes. O artigo de PENNESTRÌ e VITA (2005) e o livro de GARCÍA DE JALÓN e BAYO (1994) trazem revisões atualizadas sobre o assunto. No entanto, ambos os textos não deixam claro o emprego do termo coordenada generalizada dentro desse contexto. Apenas o livro de NIKRAVESH (1988) situa este termo em relação aos outros tipos de coordenadas. Por esta razão, optou-se por tomar este autor como base para a breve revisão dos diferentes tipos de coordenadas que será apresentada a seguir. Juntamente com esta revisão, mostram-se também as equações cinemáticas advindas de cada uma dessas formulações. Ao final desta Seção, apresenta-se a Tabela 2.1 que é uma adaptação da tabela apresentada por NIKRAVESH (1988) na página 13, acrescida dos dois outros tipos de coordenadas apresentadas por GARCÍA DE JALÓN e BAYO (1994).

### 2.1.2.1 COORDENADAS GENERALIZADAS

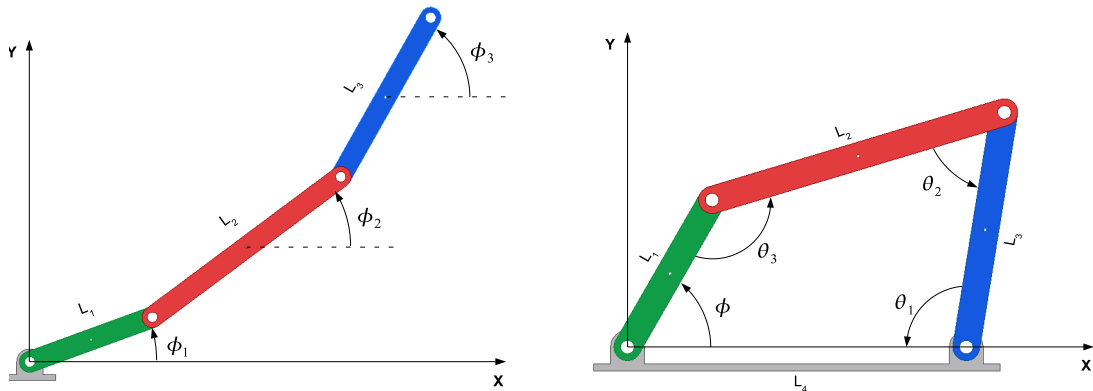
Uma ampla discussão sobre o termo coordenada generalizada já foi feita, bem como já foi definido que, por coordenada generalizada, se subentende um conjunto de coordenadas que seja *completo*, no sentido em que este seja suficiente para descrever completamente a posição de um sistema e seja composto por coordenadas *independentes* umas das outras. Desta forma o número de coordenadas generalizadas é igual ao número de graus de liberdade do sistema. Este tipo de coordenada é conveniente na descrição de sistemas mecânicos de cadeia cinemática aberta, tal como o que aparece no exemplo da Figura 2.1(a). Se as coordenadas são tomadas em relação a um referencial inercial, a descrição das posições do centro de massa, bem como de qualquer outro ponto do sistema, pode ser feita através de simples relações trigonométricas, uma vez que se imponha a trajetória das coordenadas generalizadas.

Coordenadas generalizadas também podem ser empregadas na descrição de mecanismos de cadeia cinemática fechada, tal como o mecanismo 4 barras apresentado na Figura 2.1(b). Pode-se perceber que coordenadas auxiliares (também chamadas de secundárias), apresentadas nesta Figura como  $\theta$ 's, são convenientes na descrição do mecanismo. Neste caso, as coordenadas auxiliares  $\theta$  serão dependentes da coordenada generalizada  $\phi$ . Em alguns casos, é possível encontrar equações algébricas desacopladas que podem ser resolvidas em seqüência, o que permite encontrar o valor de todas as coordenadas auxiliares, uma vez que se forneça o valor da coordenada generalizada. NIKRAVESH (1988) apresenta um exemplo dessas equações (Eqs. 2.1a - 2.1c) para o mecanismo 4 barras da Figura 2.1(b).

$$(L_1^2 + L_4^2 + L_3^2 - L_2^2) - 2L_1L_4 \cos(\phi) + 2L_4L_3 \cos(\theta_1) - 2L_1L_3 \cos(\phi - \theta_1) = 0 \quad (2.1a)$$

$$(L_1^2 + L_4^2 + L_3^2 - L_2^2) - 2L_1L_4 \cos(\phi) + 2L_2L_3 \cos(\theta_2) = 0 \quad (2.1b)$$

$$\phi + \theta_1 + \theta_2 + \theta_3 - 2\pi = 0 \quad (2.1c)$$



(a) Exemplo de coordenada generalizada em um pêndulo triplo

(b) Exemplo de coordenada generalizada ( $\phi$ ) e coordenadas secundárias ( $\theta$ ) em um mecanismo 4 barras

FIG. 2.1: Exemplos de coordenadas generalizadas

### 2.1.2.2 COORDENADAS RELATIVAS

GARCÍA DE JALÓN e BAYO (1994) definem coordenadas relativas como sendo aquelas capazes de definir a posição de cada elemento em relação ao elemento anterior numa cadeia cinemática, através do uso de parâmetros ou coordenadas correspondentes aos

graus de liberdade relativos ao tipo de junta que conecta os dois elementos. Estes autores apresentam um exemplo de emprego de coordenadas relativas no mecanismo 4 barras tal como mostrado na Figura 2.2(a). O uso de coordenadas relativas leva a um número mínimo de coordenadas dependentes. No caso de sistemas de cadeia cinemática aberta, esse número é igual ao número de graus de liberdade. Este tipo de coordenada é particularmente conveniente quando existe um motor ligado à junta que liga duas barras, pois permite o controle direto daquele grau de liberdade.

As equações cinemáticas do exemplo mostrado na Figura 2.2(a), ficariam assim:

$$L_1 \cos(\phi_1) + L_2 \cos(\phi_1 + \phi_2) + L_3 \cos(\phi_1 + \phi_2 + \phi_3) - L_4 = 0 \quad (2.2a)$$

$$L_1 \sin(\phi_1) + L_2 \sin(\phi_1 + \phi_2) + L_3 \sin(\phi_1 + \phi_2 + \phi_3) = 0 \quad (2.2b)$$

Por outro lado, NIKRAVESH (1988, p.11) considera como coordenada relativa, coordenadas que definam a orientação de um corpo tanto com relação a um referencial que se mova, quanto com relação a um que não se mova. Desta forma, ele exemplifica coordenada relativa no mecanismo 4 barras, tal como mostrado na Figura 2.2(b). Com as coordenadas definidas dessa maneira, as equações cinemáticas ficariam levemente modificadas:

$$L_1 \cos(\phi_1) + L_2 \cos(\phi_2) + L_3 \cos(\phi_3) - L_4 = 0 \quad (2.3a)$$

$$L_1 \sin(\phi_1) + L_2 \sin(\phi_2) + L_3 \sin(\phi_3) = 0 \quad (2.3b)$$

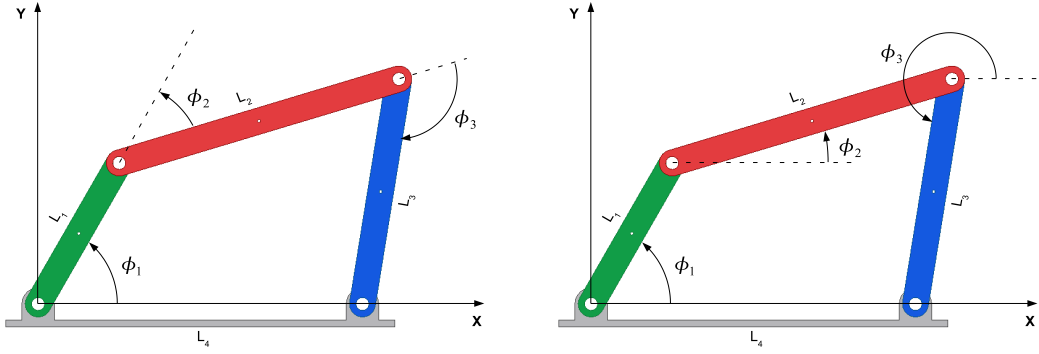
Vale lembrar que DOUGHTY (1988), apesar de não citar nada a respeito do termo coordenada relativa, também define as equações do *loop* de posição da mesma forma que NIKRAVESH (1988) (Eqs. 2.3a e 2.3b). Em ambos os casos, para se fazer a análise de posição, as Equações 2.2a/2.2b ou 2.3a/2.3b, devem ser resolvidas numericamente através de um algoritmo Newton-Raphson.

### 2.1.2.3 COORDENADAS CARTESIANAS

Coordenadas cartesianas procuram determinar a posição absoluta de um ponto de referência (em geral o centro de massa) bem como a orientação de um corpo no espaço. Portanto, no caso 2D são necessárias 3 coordenadas por corpo e no caso 3D, 6 coordenadas por corpo<sup>15</sup>. Definindo-se as coordenadas dessa maneira, obtém-se um conjunto

---

<sup>15</sup>Alguns autores (HAUG (1989), HAUG (1992), NIKRAVESH (1988)) utilizam 7 coordenadas, para evitar problemas numéricos, definindo a orientação do corpo no espaço através de parâmetros de Euler (4 coordenadas) ao invés de ângulos de Euler (3 coordenadas)



(a) Coordenada relativa segundo GARCÍA DE JALÓN e BAYO (1994)

(b) Coordenada relativa segundo NIKRAVESH (1988)

FIG. 2.2: Exemplos de coordenadas relativas em um mecanismo 4 barras

de coordenadas dependentes entre si, tal como acontece com as coordenadas relativas. A diferença fica por conta do número de equações necessárias para se definir a cinemática do mecanismo. No caso do 4 barras da Figura 2.2 são necessárias apenas duas equações, enquanto que no exemplo da Figura 2.3 utiliza-se um conjunto de 8 equações algébricas não-lineares (Eqs. 2.4a a 2.4h). Uma vez que se especifique o valor da coordenada independente (que pode ser  $\phi_1$ , neste exemplo), o valor das demais coordenadas pode ser obtido numericamente através do algoritmo Newton-Raphson.

$$x_1 - \frac{L_1}{2} \cos(\phi_1) = 0 \quad (2.4a)$$

$$y_1 - \frac{L_1}{2} \sin(\phi_1) = 0 \quad (2.4b)$$

$$x_1 + \frac{L_1}{2} \cos(\phi_1) - x_2 + \frac{L_2}{2} \cos(\phi_2) = 0 \quad (2.4c)$$

$$y_1 + \frac{L_1}{2} \sin(\phi_1) - y_2 - \frac{L_2}{2} \sin(\phi_2) = 0 \quad (2.4d)$$

$$x_2 + \frac{L_2}{2} \cos(\phi_2) - x_3 - \frac{L_3}{2} \cos(\phi_3) = 0 \quad (2.4e)$$

$$y_2 + \frac{L_2}{2} \sin(\phi_2) - y_3 - \frac{L_3}{2} \sin(\phi_3) = 0 \quad (2.4f)$$

$$x_3 - \frac{L_3}{2} \cos(\phi_3) - L_4 = 0 \quad (2.4g)$$

$$y_3 - \frac{L_3}{2} \sin(\phi_3) = 0 \quad (2.4h)$$

As matrizes que aparecem nas equações de movimento formuladas com esse conjunto de coordenadas são esparsas. Além do mais, as posições de cada elemento são diretamente determinadas, o que requer um menor esforço computacional de pré e pós-processamento. Isso faz com que esse tipo de coordenada seja preferido por *softwares* comerciais de simula-



ção de sistemas dinâmicos tais como ADAMS<sup>16</sup> e DADS<sup>17</sup> (PENNESTRÌ e VITA (2005)). No entanto, o alto número de coordenadas que se obtém quando se emprega coordenadas cartesianas pode ser um fator limitante quando se deseja considerar aplicações em que se exija a integração de equações de movimento em tempo real (GARCÍA DE JALÓN e BAYO (1994)).

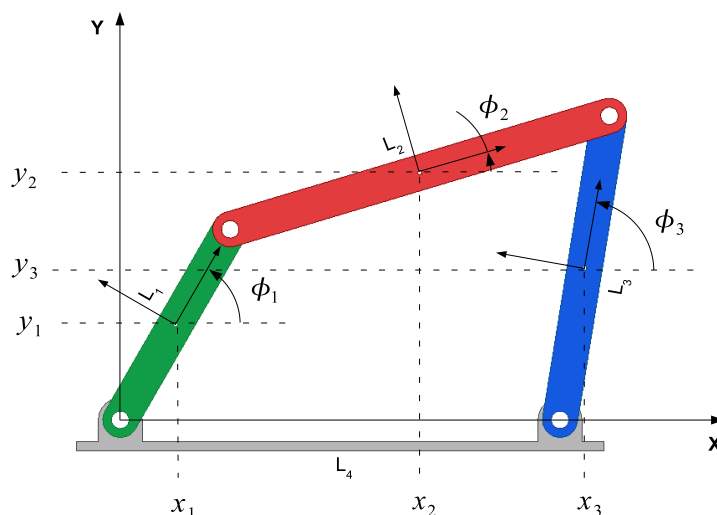


FIG. 2.3: Exemplo de coordenadas cartesianas em um mecanismo 4 barras

#### 2.1.2.4 COORDENADAS NATURAIS

Coordenadas naturais foram originalmente introduzidas por GARCÍA DE JALÓN (2007) há 25 anos e representam uma alternativa entre o uso de coordenadas relativas e cartesianas. Seu uso permite definir a configuração de um corpo no espaço com um conjunto de coordenadas dependentes, cujo número é superior à formulação com coordenadas relativas, mas inferior à abordagem com coordenadas cartesianas. A idéia central é a de que como cada corpo possui pelo menos dois pontos, sua posição e orientação pode ser determinada a partir das coordenadas cartesianas desses dois pontos. Como alguns desses pontos (as articulações) são compartilhados por pelo menos dois corpos, o número de coordenadas necessárias para descrever um mecanismo é inferior a 4 vezes o número de corpos. Nos exemplos mostrados nesta Seção, são necessárias 3 coordenadas relativas, ou 9 coordenadas cartesianas para descrever um mecanismo 4 barras. No entanto, ao optar por utilizar coordenadas naturais, são necessárias apenas 4 coordenadas para descrever o

<sup>16</sup><http://www.mscsoftware.com/products/adams>

<sup>17</sup><http://www.lmsintl.com/DADS>

mesmo sistema, tal como mostra a Figura 2.4.

Com esta formulação, as equações de restrição deixam de ser equações transcendentais e se tornam quadráticas ou lineares. Isso pode ser verificado nas Equações 2.5a-2.5c em que cada uma delas representa a condição de comprimento constante das barras.

$$(x_1 - X_A)^2 + (y_1 - Y_A)^2 - L_2^2 = 0 \quad (2.5a)$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_3^2 = 0 \quad (2.5b)$$

$$(x_2 - X_B)^2 + (y_2 - Y_B)^2 - L_4^2 = 0 \quad (2.5c)$$

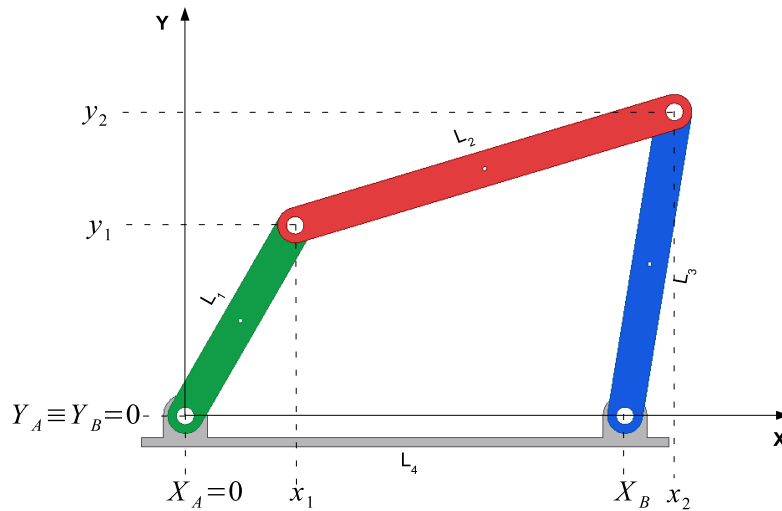


FIG. 2.4: Exemplo de coordenadas naturais em um mecanismo 4 barras

### 2.1.2.5 COORDENADAS MISTAS

Uma das desvantagens tanto das coordenadas cartesianas quanto das coordenadas naturais é o fato de não poder levar em conta diretamente o ângulo relativo entre as barras. O cálculo direto do ângulo relativo pode ser vantajoso quando se considera a presença de motores nas juntas. Esse problema pode ser contornado com o uso das chamadas “coordenadas mistas”. Nesse tipo de coordenada toma-se como base uma formulação de coordenada cartesiana ou natural e adiciona-se qualquer outro tipo de coordenada que seja conveniente para a descrição do problema em questão. Para cada coordenada adicional, deve-se determinar uma equação de restrição que a relacione com as demais.

No exemplo da Figura 2.5(a) apresenta-se a formulação com coordenadas cartesianas usando duas coordenadas angulares adicionais  $\phi_4$  e  $\phi_5$ , que representam o ângulo rela-

tivo entre as barras. Neste caso, as equações de restrição cinemática seriam dadas pelas Equações 2.6a a 2.6j. Repare que as oito primeiras equações, são as equações vindas da formulação com coordenadas cartesianas. Já as Equações 2.6i e 2.6j expressam relações trigonométricas a partir das quais pode-se calcular os ângulos desejados.

$$x_1 - \frac{L_1}{2} \cos(\phi_1) = 0 \quad (2.6a)$$

$$y_1 - \frac{L_1}{2} \sin(\phi_1) = 0 \quad (2.6b)$$

$$x_1 + \frac{L_1}{2} \cos(\phi_1) - x_2 + \frac{L_2}{2} \cos(\phi_2) = 0 \quad (2.6c)$$

$$y_1 + \frac{L_1}{2} \sin(\phi_1) - y_2 - \frac{L_2}{2} \sin(\phi_2) = 0 \quad (2.6d)$$

$$x_2 + \frac{L_2}{2} \cos(\phi_2) - x_3 - \frac{L_3}{2} \cos(\phi_3) = 0 \quad (2.6e)$$

$$y_2 + \frac{L_2}{2} \sin(\phi_2) - y_3 - \frac{L_3}{2} \sin(\phi_3) = 0 \quad (2.6f)$$

$$x_3 - \frac{L_3}{2} \cos(\phi_3) - L_4 = 0 \quad (2.6g)$$

$$y_3 - \frac{L_3}{2} \sin(\phi_3) = 0 \quad (2.6h)$$

$$\phi_4 - \phi_2 + \phi_1 = 0 \quad (2.6i)$$

$$\phi_5 - \phi_3 + \phi_2 - \pi = 0 \quad (2.6j)$$

$$(2.6k)$$

Já a Figura 2.5(b) mostra um exemplo de coordenadas mistas obtidas a partir da adição do ângulo  $\phi_1$  ao vetor de coordenadas naturais onde, por definição, não aparecem coordenadas angulares. No entanto, no caso do 4 barras, é conveniente a presença do ângulo  $\phi_1$  indicado na Figura para determinar a configuração do mecanismo.

$$(x_1 - X_A)^2 + (y_1 - Y_A)^2 - L_2^2 = 0 \quad (2.7a)$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_3^2 = 0 \quad (2.7b)$$

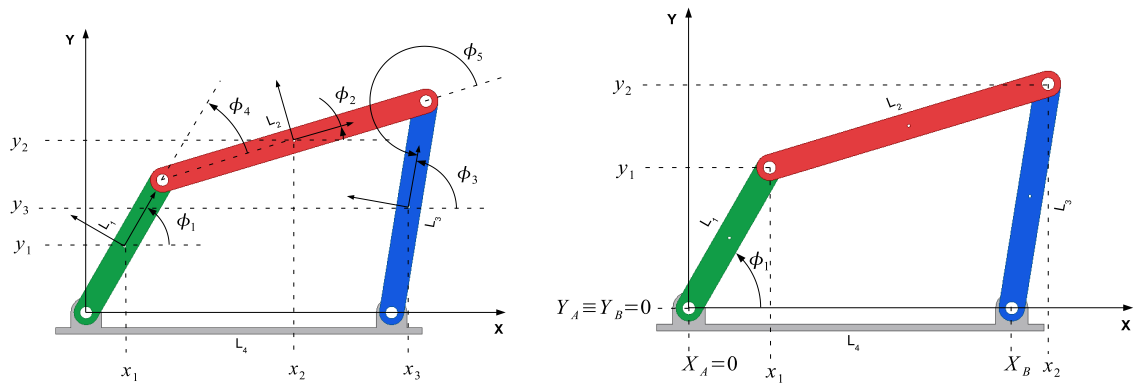
$$(x_2 - X_B)^2 + (y_2 - Y_B)^2 - L_4^2 = 0 \quad (2.7c)$$

$$(x_1 - X_A) - L_1 \cos(\phi_1) = 0 \quad (2.7d)$$

$$(y_1 - Y_A) - L_1 \sin(\phi_1) = 0 \quad (2.7e)$$

Segundo GARCÍA DE JALÓN e BAYO (1994) a equação de restrição relativa ao ângulo  $\phi_1$ , neste exemplo, pode ser deduzida a partir da relação do produto escalar ou do produto vetorial. Por exemplo: seja o vetor  $\vec{L}_1 = [x_1 - X_A; y_1 - Y_A; 0]$  e o versor

$\hat{i} = [1; 0; 0]$ , então o produto escalar entre eles é dado por  $\vec{L}_1 \cdot \hat{i} = L_1 \cdot 1 \cdot \cos(\phi_1) = (x_1 - X_A) \cdot 1 + (y_1 - Y_A) \cdot 0$ . Esta expressão quando simplificada nos leva a Equação 2.7d. Pela regra do produto vetorial  $|\hat{i} \times \vec{L}_1| = |\vec{L}_1| |\hat{i}| \sin \phi$ . Desenvolvendo esta expressão para o vetor  $\vec{L}_1$  e o versor  $\hat{i}$  definidos anteriormente, temos que  $y_1 - Y_A = L_1 \sin(\phi_1)$ , o que é a própria Eq. 2.7e. Este autor recomenda que se use a relação do produto escalar quando  $0^\circ < \phi_1 < 180^\circ$ , e a relação do produto vetorial quando  $-90^\circ < \phi_1 < 90^\circ$ , pois a relação do produto escalar deixa de ser válida quando  $\phi_1$  está próximo de  $0^\circ$  ou de  $180^\circ$  enquanto que a relação do produto vetorial encontra problemas para valores de  $\phi_1$  próximos a  $-90^\circ$  ou  $90^\circ$ . Se a natureza do problema imponha que  $\phi_1$  assuma qualquer valor, a análise de posição pode ser implementada com um algoritmo Newton-Raphson que seja capaz de alterar o vetor de restrições (e também seu Jacobiano) quando  $\phi_1$  se aproxime dos pontos críticos, ou usando as duas restrições (Eqs. 2.7d e 2.7e), obtendo-se desta forma um sistema de equações sobre-determinado (em inglês, *overdetermined*).



(a) Exemplo de coordenada mista formada por coordenadas Cartesianas com as coordenadas auxiliares  $\phi_1, \phi_2$  e  $\phi_3$   
 (b) Exemplo de coordenada mista formada por coordenadas naturais e a coordenada auxiliar  $\phi_1$

FIG. 2.5: Exemplos de coordenadas mistas em um mecanismo 4 barras

TAB. 2.1: Comparativo das características entre os principais tipos de coordenadas (adaptada de (NIKRAVESH, 1988, p.13))

	Coordenadas				
	Generalizadas	Relativas	Naturais	Cartesianas	Mistas
<b>Dimensão vetor [q]</b>	$[\phi_1]$	$[\phi_1 \phi_2 \phi_3]^T$	$[x_1 y_1 x_2 y_2]^T$	$[x_1 y_1 \phi_1 x_2 y_2 \phi_2 x_3 y_3 \phi_3]^T$	$[x_1 y_1 x_2 y_2 \phi_1]^T$ ou $[x_1 y_1 \phi_1 x_2 y_2 \phi_2 x_3 y_3 \phi_3 \phi_4 \phi_5]^T$
<b>Nº de coordenadas</b>	Mínimo	Pequeno	Médio	Alto	De médio a alto
<b>Dependência das coordenadas</b>	Independente	Dependente	Dependente	Dependente	Dependente
<b>Nº eqs. diferenciais de 2ª ordem</b>	Mínimo	Moderado	Moderado	Alto	De moderado a alto
<b>Nº eqs. algébricas de restrição</b>	Nenhuma	Moderado	Moderado	Alto	De moderado a alto
<b>Grau de não-linearidade</b>	Alto	Moderado	Muito Baixo	Baixo	Baixo
<b>Dedução das eqs. de movimento</b>	Difícil	De moderado a difícil	De moderado a simples	Simple	De moderado a simples
<b>Eficiência computacional</b>	Eficiente	Eficiente	Eficiente	Não muito eficiente	Varia
<b>Desenvolvimento de um programa de aplicação geral</b>	Difícil	De moderado a difícil	Fácil	Fácil	De moderado a fácil

### 2.1.3 EQUAÇÕES DE RESTRIÇÃO

Na Seção 2.1.2 foram apresentadas 5 formas diferentes de se escolher as coordenadas que descrevem um sistema mecânico. Junto com elas, mostraram-se também as equações necessárias para realizar a análise cinemática. Nos exemplos apresentados, apenas quando se utilizou coordenadas generalizadas foi possível encontrar um conjunto de equações desacopladas que podem ser resolvidas algebricamente, em sequência, de forma a encontrar o valor de todas as variáveis (no caso, chamadas de auxiliares). No entanto, quando se trata de um mecanismo de cadeia cinemática fechada qualquer, isto nem sempre é possível, especialmente quando se trata de mecanismos de cadeia cinemática fechada.

Quando se escolhe um conjunto de coordenadas dependentes, em geral obtém-se um conjunto de equações não-lineares acopladas que devem ser resolvidas simultaneamente. Estas equações de restrição são classificadas em dois grandes grupos: restrições cinemáticas (*kinematic constraints*)  $\Phi^K$  e restrições diretoras (*driving constraints*)  $\Phi^D$ , (HAUG, 1989).

As chamadas restrições cinemáticas representam as conexões físicas entre os corpos e, por este motivo, elas impõem restrições ao movimento relativo entre corpos que estão conectados. Esse tipo de restrição é função somente das coordenadas do sistema, neste caso cartesianas, e não dependem explicitamente do tempo. Já as restrições diretoras dependem explicitamente do tempo e das coordenadas cartesianas, que por sua vez estão associadas aos graus de liberdade do sistema. Seu papel é o de prover uma função de entrada para alguma coordenada absoluta ou relativa que deve ser fornecida por algum motor ou atuador. Esse tipo de comportamento é comum em robôs ou máquinas operatrizes onde um determinado tipo de movimento é prescrito pelo usuário.

HAUG (1989) no Capítulo 3, define uma série de restrições entre pares de corpos rígidos. Ele chama atenção ainda que, para a correta descrição do movimento de um mecanismo, a escolha das equações de restrição deve ser feita de tal maneira que estas equações “tenham como consequência” a geometria da junta considerada. Textualmente, “*The equations of constraint must imply the geometry of the joint*”. Portanto, para cada tipo de junta, e/ou movimento relativo entre dois corpos, HAUG (1989) define os seguintes tipos de equações de restrição.

Restrições Cinemáticas:  $\Phi^K$

- Restrição absoluta de distância
- Restrição absoluta de posição

- Restrição absoluta de ângulo
- Restrição relativa de distância
- Restrição relativa de posição (horizontal ou vertical)
- Restrição relativa de ângulo
- Restrição para junta de revolução
- Restrição para junta de translação
- Restrições para juntas compostas (revolução-revolução, translação-revolução)
- Restrições entre engrenagens (convexa-convexa, côncava-convexa, pinhão-cremalheira)
- Restrições entre cames (came-seguidor, ponto-seguidor)

Restrições Diretoras:  $\Phi^D$

- Restrição Diretora Absoluta
- Restrição Diretora Relativa

O primeiro passo para análise cinemática é montar um vetor de restrições composto (Eq. 2.8), ou simplesmente vetor de restrições, formado por um conjunto de equações de restrição cinemáticas e diretoras. Este vetor depende explicitamente das coordenadas cartesianas (que estão presentes no vetor  $\mathbf{q}$ ) e do tempo (devido à presença da restrição diretora  $\Phi^D$ ).

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^K(\mathbf{q}, t) \\ \Phi^D(\mathbf{q}, t) \end{bmatrix} \quad (2.8)$$

As equações de restrição são manipuladas de tal forma que do lado direito apareça um vetor nulo, tal com na Eq. 2.9. Assume-se que um método numérico será utilizado para resolver esta equação no tempo em intervalos discretos.

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (2.9)$$

No caso mais geral o vetor  $\mathbf{q}$  não é conhecido como uma função explícita do tempo e não pode ser diferenciado para obter  $\dot{\mathbf{q}}$  e  $\ddot{\mathbf{q}}$ . Uma alternativa é utilizar a Regra da Cadeia para derivar os dois lados da Eq. 2.9 com relação ao tempo de forma a obter a equação de velocidade:

$$\dot{\Phi}(\mathbf{q}, t) = \frac{\partial \Phi}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial \Phi}{\partial t} \frac{dt}{dt} = \mathbf{0} \quad (2.10)$$

Na primeira parcela da Eq. 2.10 aparece uma derivada parcial de um vetor coluna ( $\Phi$ ) em relação a outro ( $\mathbf{q}$ ). Este tipo de operação dá origem a uma matriz Jacobiana em que o elemento  $(i, j)$  desta matriz é a derivada parcial da equação  $i$  do vetor  $\Phi$  em relação à coordenada  $j$  do vetor  $\mathbf{q}$ . Neste capítulo, utiliza-se a notação  $\Phi_{\mathbf{q}}$  para indicar esta matriz Jacobiana. Já na segunda parcela da Eq. 2.10 aparece a derivada parcial de um vetor com relação ao tempo. O resultado desta operação é também um vetor-coluna, em que cada elemento representa a derivada parcial da equação de restrição  $i$  do vetor  $\Phi$  em relação ao tempo. Neste capítulo, utiliza-se a notação  $\Phi_t$  para indicar este vetor. Portanto, a Eq. 2.10 pode ser reescrita como:

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t = \mathbf{0} \quad (2.11)$$

É conveniente manipular a equação anterior e definir o termo  $\nu$  tal como aparece na Equação 2.12. Se a matriz Jacobiana  $\Phi_{\mathbf{q}}$  não for singular, pode-se isolar  $\dot{\mathbf{q}}$  na Eq. 2.12, de forma a obter as velocidades das coordenadas cartesianas.

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} = -\Phi_t \equiv \nu \quad (2.12)$$

De forma análoga, pode-se diferenciar os dois lados da Equação 2.10 com relação ao tempo usando a regra da cadeia, de forma a obter uma expressão para o cálculo da aceleração.

$$\ddot{\Phi}(\mathbf{q}, t) = \frac{\partial(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})}{\partial t} \frac{dt}{dt} + \frac{\partial \Phi_t}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial \Phi_t}{\partial t} \frac{dt}{dt} = \mathbf{0} \quad (2.13)$$

Na equação anterior o termo  $\partial(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})/\partial t$  pode ser ainda desdobrado como:

$$\ddot{\Phi}(\mathbf{q}, t) = \frac{\partial(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial \Phi_{\mathbf{q}}}{\partial t} \dot{\mathbf{q}} + \Phi_{\mathbf{q}} \frac{\partial \dot{\mathbf{q}}}{\partial t} + \frac{\partial \Phi_t}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} + \frac{\partial \Phi_t}{\partial t} \frac{dt}{dt} = \mathbf{0} \quad (2.14)$$

Na Eq. 2.14 aparece a derivada parcial da matriz jacobina  $\Phi_{\mathbf{q}}$  em relação ao tempo. O resultado desta operação – indicado como  $\Phi_{\mathbf{q}t}$  na Eq. 2.15 – é uma matriz cujos elementos são as derivadas temporais da matriz  $\Phi_{\mathbf{q}}$ . Nesta equação aparece também a derivada parcial do vetor  $\Phi_t$  em relação ao vetor  $\mathbf{q}$ . Esta operação dá origem a uma matriz indicada



na Eq. 2.15 como  $\Phi_{t\mathbf{q}}$ . Já o termo  $\partial(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})/\partial \mathbf{q}$  é também uma matriz, visto que o produto  $(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})$  é um vetor e este está sendo derivado também em relação a outro vetor. Portanto, a notação simplificada deste termo aparece na Equação 2.15 como  $(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})_{\mathbf{q}}$ . Por fim, o vetor  $\Phi_{tt}$  representa a segunda derivada de  $\Phi$  em relação ao tempo.

$$\ddot{\Phi}(\mathbf{q}, t) = (\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_{\mathbf{q}t} \dot{\mathbf{q}} + \Phi_{\mathbf{q}} \ddot{\mathbf{q}} + \Phi_{t\mathbf{q}} \dot{\mathbf{q}} + \Phi_{tt} = \mathbf{0} \quad (2.15)$$

É possível mostrar ainda que  $\Phi_{\mathbf{q}t} = \Phi_{t\mathbf{q}}$ <sup>18</sup>. Desta forma, a equação anterior se simplifica para:

$$\ddot{\Phi}(\mathbf{q}, t) = (\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} + 2\Phi_{t\mathbf{q}} \dot{\mathbf{q}} + \Phi_{\mathbf{q}} \ddot{\mathbf{q}} + \Phi_{tt} = \mathbf{0} \quad (2.16)$$

Tal como foi feito na dedução da equação de velocidade, pode-se retirar o termo  $\ddot{\Phi}(\mathbf{q}, t)$  de forma a reescrever a equação acima, e definir o vetor  $\gamma$ , tal como aparece na Equação 2.17. Assim, se a matriz Jacobiana  $\Phi_{\mathbf{q}}$  não for singular, é possível isolar  $\ddot{\mathbf{q}}$  na Equação 2.17 para se obter as acelerações das coordenadas cartesianas.

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}})_{\mathbf{q}} \cdot \dot{\mathbf{q}} - 2\Phi_{t\mathbf{q}} \dot{\mathbf{q}} - \Phi_{tt} \equiv \gamma \quad (2.17)$$

## 2.2 EQUAÇÃO DE MOVIMENTO

### 2.2.1 EQUAÇÕES DIFERENCIAIS ALGÉBRICAS (DAE'S)

Na Seção 2.1.3 foram deduzidas as equações para o cálculo numérico de  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  e  $\ddot{\mathbf{q}}$ . Além da análise cinemática, estas equações se prestam para a dedução das equações de movimento de sistemas planos com restrições. HAUG (1989), p.219 faz essa dedução a partir da equação variacional de movimento 2.18, para um  $\delta \mathbf{q}_i$  arbitrário, considerando

---

<sup>18</sup>APOSTOL (1969) mostra na página 255 que duas derivadas mistas parciais  $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$  de uma função escalar  $f$ , são iguais em um ponto desde que uma delas seja contínua na vizinhança deste ponto. Para estender esta propriedade para uma função vetorial ou matricial, a continuidade deve ser obedecida para cada elemento da matriz ou vetor. Portanto basta mostrar que  $\Phi_{\mathbf{q}t}$  ou  $\Phi_{t\mathbf{q}}$  é contínua para qualquer ponto. Uma prova geral de que uma destas matrizes é contínua para qualquer ponto não será desenvolvida aqui. No entanto, observando a matriz Jacobiana da Eq. 2.31 pode-se perceber que todos os elementos da matriz são funções contínuas, e que suas derivadas em relação ao tempo também são contínuas. Desta forma mostra-se que, todos os elementos da matriz  $\Phi_{\mathbf{q}t}$  são contínuos e que portanto, a identidade  $\Phi_{\mathbf{q}t} = \Phi_{t\mathbf{q}}$  é válida, pelo menos para o exemplo do mecanismo 4 barras.

que todas as forças (internas e externas) que atuam em cada corpo estão incluídas no vetor  $\mathbf{Q}_i$ .

$$\sum_{i=1}^{nb} \delta \mathbf{q}_i [\mathbf{M}_i \ddot{\mathbf{q}}_i - \mathbf{Q}_i] = 0 \quad (2.18)$$

Na Eq.2.18,  $\mathbf{q}_i = [x_i, y_i, \phi_i]^T$  são as coordenadas e  $\ddot{\mathbf{q}}_i = [\ddot{x}_i, \ddot{y}_i, \ddot{\phi}_i]^T$  são as acelerações das coordenadas cartesianas do corpo  $i$ ,  $\mathbf{M}_i = \text{diag}(m_i, m_i, J_i)$ , é a matriz diagonal de massa do corpo  $i$ , e  $\mathbf{Q}_i$  é um vetor de três elementos que representa todas as forças ou torques (internos e externos) que atuam sobre o corpo  $i$ , nas direções  $x$ ,  $y$  e  $\phi$ , respectivamente. Por fim,  $nb$  representa o número de corpos no sistema. Definindo os vetores compostos como:

$$\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T, \dots, \mathbf{q}_{nb}^T]^T \quad (2.19)$$

$$\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{nb}) \quad (2.20)$$

$$\mathbf{Q} = [\mathbf{Q}_1^T, \mathbf{Q}_2^T, \dots, \mathbf{Q}_{nb}^T]^T \quad (2.21)$$

é possível reescrever a equação de movimento na forma variacional, de forma mais compacta, tal como mostrado na equação 2.22.

$$\delta \mathbf{q}^T [\mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q}] = 0 \quad (2.22)$$

Utilizando o Teorema dos Multiplicadores de Lagrange (HAUG, 1992, p.109) e (HAUG, 1989, p.223) é possível provar que as forças internas que atuam sobre cada corpo podem ser escritas em função do Jacobiano das equações de restrição e de um vetor de incógnitas  $\lambda$ . Essas forças internas não realizam trabalho, e são desconhecidas, *a priori*. Desta forma, a equação de movimento de um sistema multi-corpos, no caso bidimensional, pode ser escrita como:

$$\mathbf{M} \ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda = \mathbf{Q}^A \quad (2.23a)$$

$$\Phi(\mathbf{q}, t) = 0 \quad (2.23b)$$

### 2.2.2 ÍNDICE DE UMA DAE

As Equações 2.23a e 2.23b são conhecidas como um sistema de *Equações Diferenciais-Algébricas* de índice 3, (HAUG (1989), BRENAN et al. (1989)). Neste texto chamamos este termo de DAE devido a sua sigla em inglês: “Differential-Algebraic Equations”. Chama-se de DAE a um conjunto de equações diferenciais que descrevem o comportamento dinâmico de um sistema que está acoplado a um conjunto de equações algébricas que devem ser satisfeitas para qualquer solução destas equações diferenciais<sup>19</sup>. DAE’s em geral não possuem solução algébrica e só podem ser resolvidas através de métodos numéricos. Mesmo assim, apesar de existirem alguns métodos numéricos capazes de resolver uma DAE de índice 3 (os chamados *DAE solvers*), como a da Eq. 2.23, como por exemplo, os apresentados por BRENAN et al. (1989), não estão disponíveis até o momento algoritmos suficientemente robustos e de uso geral capazes de resolver DAE’s de índices superiores (i.e.,  $> 2$ ). Os que existem são pouco eficientes numericamente e não estão livres de problemas de estabilidade (GARCÍA DE JALÓN e BAYO, 1994, p.163). Por este motivo, DAE’s são geralmente resolvidas através de técnicas de redução de índice.

O índice de uma DAE, na ausência de singularidades, se refere ao número de diferenciações analíticas necessárias para transformar uma DAE em um sistema de Equações Diferenciais Ordinárias (ODE), (MINGARDI e PENNESTRÌ (1999)). Derivando-se a Equação 2.23b duas vezes, isto é, utilizando-se a equação de aceleração (Eq. 2.17), aparece o vetor de aceleração das coordenadas cartesianas. Desta forma, é possível transformar a DAE de índice 3 numa DAE de índice 1, o que nada mais é do que uma ODE.

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A \\ \gamma \end{bmatrix} \quad (2.24)$$

### 2.2.3 REDUÇÃO DE ORDEM DE UMA DAE

A Equação 2.24 é uma equação diferencial ordinária de 2<sup>a</sup> ordem. Apesar de existirem alguns métodos numéricos para a integração de ODE’s de 2<sup>a</sup> ordem, é conveniente reduzir a ordem da ODE, pois os métodos existentes para integração de equações de 1<sup>a</sup> ordem são muito mais robustos e bem desenvolvidos. STEJSKAL e VALASEK (1996, p.357) mostram que ao fazer a seguinte transformação de coordenadas,

---

<sup>19</sup>MÜLLER (2000) chama de *Descriptor Systems* DAE’s nas quais o vetor  $\mathbf{Q}^A$  inclui forças de controle.

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{q} \xrightarrow{d/dt} \dot{\mathbf{y}}_1 = \dot{\mathbf{q}} = \dot{\mathbf{y}}_2 \\ \mathbf{y}_2 &= \dot{\mathbf{q}} \xrightarrow{d/dt} \ddot{\mathbf{q}} \end{aligned}$$

é possível reescrever a Equação 2.24 como:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \mathbf{0} & \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{y}}_1 \\ \dot{\mathbf{y}}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{y}_2 \\ \mathbf{Q}^A \\ \gamma \end{bmatrix} \quad (2.25)$$

PENNESTRÌ e VITA (2004) chamam a atenção de que alguns integradores requerem que o sistema seja escrito na forma:

$$[\mathbf{K}][\dot{\mathbf{y}}] = [\mathbf{f}(\mathbf{y})] \quad (2.26)$$

$$\text{onde } [\mathbf{K}] = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, [\mathbf{y}] = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} \text{ e } [\mathbf{f}(\mathbf{y})] = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda - \mathbf{Q}^A \\ \Phi \end{bmatrix}$$

#### 2.2.4 INTEGRAÇÃO DE DAE TRANSFORMADA EM ODE

Apesar de ser possível transformar uma DAE numa ODE, estas equações não podem ser tratadas como simples equações diferenciais ordinárias (HAUG (1989, p.243)). Segundo PETZOLD (1982), para algumas classes de problemas, DAE's podem ser integradas com algoritmos para resolver ODE's duras. No entanto, isto nem sempre é possível pois a mudança do passo de integração causa grandes erros na solução.

O maior problema na integração da Equação 2.24 é o que se chama de “estabilização de violação das restrições” (*drift*). Na formulação do problema original a DAE resultante é aquela mostrada na Equação 2.23, na qual está presente a restrição de posição  $\Phi = \mathbf{0}$ . No entanto, o que é efetivamente integrado é a Eq. 2.24, que contém a restrição de aceleração  $\Phi_{\mathbf{q}}\ddot{\mathbf{q}} = \mathbf{0}$ . Desta forma, apenas a restrição de aceleração é satisfeita dentro das tolerâncias integração. Já as restrições de posição não são necessariamente satisfeitas. Isso resulta no aumento progressivo da violação das restrições de posição à medida que a integração é realizada por tempos maiores.

Um solução popularmente utilizada e de fácil implementação é a chamada “Estabilização de Baumgarte” (HAUG (1989), STEJSKAL e VALASEK (1996)). Consiste na reformulação da Equação 2.17, definindo-se um novo  $\hat{\gamma}$ , que deve ser substituído na Equação 2.24, tal apresentado na Eq. 2.27. YU e CHEN (2000) afirmam que não há regras para a escolha dos parâmetros  $\alpha$  e  $\beta$ . Se esses parâmetros forem muito altos, a dinâmica do sistema será seriamente afetada e os estados obtidos na integração não irão descrever o comportamento do sistema. Por outro lado, se os valores forem muito pequenos as violações não poderão ser controladas apropriadamente. Portanto sua escolha depende de tentativa e erro e da experiência do usuário.

$$\Phi_{\mathbf{q}}\ddot{\mathbf{q}} = \gamma - 2\alpha(\Phi_{\mathbf{q}} \cdot \dot{\mathbf{q}} + \Phi_t) - \beta^2\Phi \equiv \hat{\gamma} \quad (2.27)$$

A Figura 2.6 mostra um exemplo do efeito que o uso da Estabilização de Baumgarte tem sobre a integração de uma DAE. Neste exemplo a equação de movimento de um mecanismo 4 barras com a manivela girando a velocidade constante de 60 rpm foi integrada durante 4s. A Figura mostra a norma Euclidiana das restrições de posição, velocidade e aceleração calculada em cada instante de tempo. Repare que quando se utiliza a estabilização de Baumgarte, a violação das restrições de posição se mantém abaixo de  $\|\Phi\| \leq 10^{-6}$ ; enquanto que, quando ela não é utilizada, existe uma tendência da violação aumentar continuamente. Comportamento parecido é observado com a norma da restrição de velocidade ( $\|\Phi_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_t\|$ ). Já com a restrição de aceleração, isto não acontece. Usando-se ou não estabilização, a norma Euclidiana da restrição de aceleração sem mantém, em média entre  $10^{-14} \leq \|\Phi_{\mathbf{q}}\ddot{\mathbf{q}} - \hat{\gamma}\| \leq 10^{-13}$ .

A utilização da estabilização de Baumgarte não é garantia de convergência numérica. HAUG (1989, p.257) afirma que quando o mecanismo se encontra na proximidade de configurações singulares, não é raro o algoritmo divergir. Outro método bastante utilizado para resolver o problema da estabilização das restrições é o *Coordinate Partitioning*, originalmente proposto por WEHAGE e HAUG (1982). A idéia central deste método é encontrar  $\ddot{\mathbf{q}}$  a partir da equação 2.24 e particioná-lo em um subvetor de coordenadas independentes  $\ddot{\mathbf{q}}_i$  e outro de coordenadas dependentes  $\ddot{\mathbf{q}}_d$ . A integração é feita apenas sobre o subvetor  $\ddot{\mathbf{q}}_i$ , encontrando-se  $\dot{\mathbf{q}}_i$  e  $\mathbf{q}_i$ . A cada passo da integração utiliza-se o algoritmo de Newton-Raphson para encontrar  $\dot{\mathbf{q}}_d$  e  $\mathbf{q}_d$ . A integração prossegue reconstruindo o vetor original  $\mathbf{q} = [\mathbf{q}_i, \mathbf{q}_d]^T$ . GARCÍA DE JALÓN e BAYO (1994) apresentam vários outros métodos tais como os baseados em matriz de projeção  $[R]$ , transformações de velocidade e em equações canônicas. UNDA et al. (1987) fazem um comparativo teórico e numérico

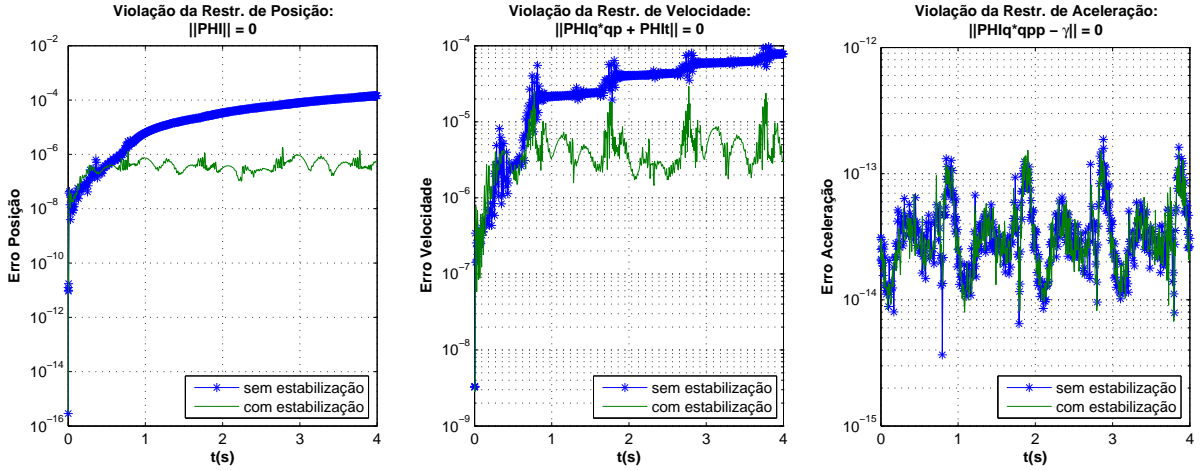


FIG. 2.6: Norma das restrições de posição, velocidade e aceleração calculadas ponto a ponto. Sem estabilização:  $\alpha = 0$  e  $\beta = 0$ . Com estabilização:  $\alpha = 1$  e  $\beta = 10$ .

entre esses diferentes métodos, considerando também o uso de coordenadas cartesianas e naturais. Uma revisão mais recente está disponível em NETO e AMBRÓSIO (2003). O desenvolvimento de métodos numéricos para integração de DAE's aplicadas à dinâmica de sistemas multi-corpos é ainda um tema que motiva recentes pesquisas, em especial quando existe a necessidade de se fazer a integração das equações de movimento em tempo real ARNOLD et al. (2007) e DEL CITTO et al. (2004), COSSALTER e LOT (2002) and BAYO et al. (1991).

## 2.3 DINÂMICA INVERSA: MÉTODO TRADICIONAL

### 2.3.1 EQUACIONAMENTO

Nesta seção será mostrado o procedimento para a análise cinemática e dinâmica inversa de um mecanismo 4 barras, utilizando-se coordenadas cartesianas e Equações Diferenciais Algébricas. A abordagem adotada baseia-se naquela utilizada por HAUG (1989).

O primeiro passo é fazer uma representação gráfica do mecanismo em estudo (Figura 2.7), identificando os elementos móveis (as barras), a posição de seus centros de massa juntamente com os sistemas de coordenadas fixados a eles, bem como os ângulos de orientação desses sistemas de coordenadas locais. Com isso é possível definir o vetor de coordenadas cartesianas que irão descrever, de forma única, a configuração do mecanismo (Eq. 2.28).

$$\mathbf{q} = [x_1, y_1, \phi_1, x_2, y_2, \phi_2, x_3, y_3, \phi_3]^T \quad (2.28)$$

Em geral, existe mais de uma maneira de se escolher o vetor de coordenadas cartesianas  $\mathbf{q}$ . No caso específico do 4 barras, poder-se-ia considerar o segmento  $\bar{AD}$  como uma quarta barra levando a um vetor  $\mathbf{q}$  com 12 elementos ou então substituir a barra 3 por uma restrição cinemática absoluta de distância (ver Seção 2.1.3), o que levaria a um vetor  $\mathbf{q}$  com apenas 5 elementos ( $\mathbf{q} = [\phi_1, x_2, y_2, \phi_2, \phi_3]^T$ ), tal como aparece no exemplo da pág. 88 de HAUG (1989).

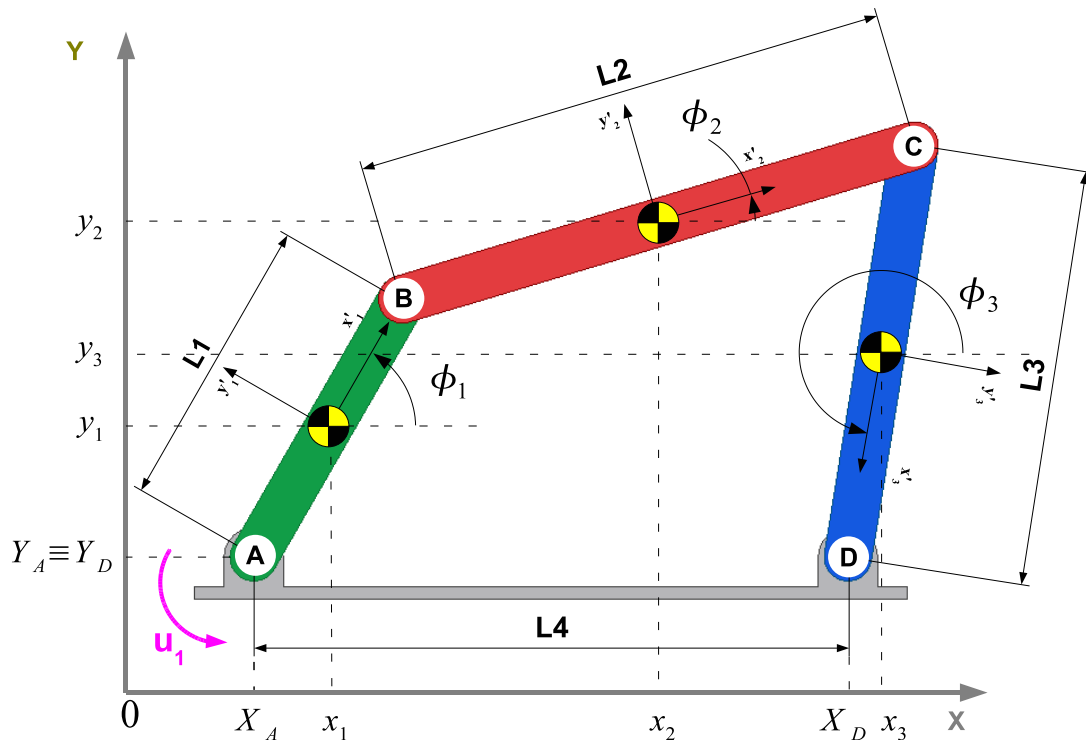


FIG. 2.7: Mecanismo 4 barras - coordenadas cartesianas

Em seguida, deve-se escrever as equações de restrição do vetor  $\Phi$ , mostradas na Eq. 2.29. As duas primeiras linhas deste vetor representam *restrições cinemáticas absolutas de posição*. Elas vinculam as coordenadas da barra 1 ( $\vec{r}_1 = [x_1, y_1]$ ) ao ponto conhecido  $\vec{A} = [X_A, Y_A]$ . O significado delas está representado graficamente na Figura 2.8(a). Da terceira à sexta linha do vetor  $\Phi$ , aparecem *restrições cinemáticas para junta de revolução*. Seu significado é o de impor que a posição do ponto B da barra 1 é a mesma do ponto B da barra 2. O mesmo raciocínio se aplica para o ponto C comum às barras 2 e 3. Essas duas restrições aparecem representadas graficamente nas Figuras 2.8(b) e 2.8(c). A sétima e oitava linhas do vetor  $\Phi$  são mais duas *expressões de restrição cinemáticas absolutas de posição* para o ponto D, tal como mostra a Figura 2.8(d). Por fim, a última expressão do vetor  $\Phi$  é uma *restrição diretora absoluta* cuja função é a de impor um

comportamento conhecido a uma das coordenadas cartesianas, visto que este mecanismo tem apenas 1 GDL. Isto se traduz pelo fato de que a restrição diretora depende explicitamente do tempo. Neste caso, seja-se que  $\phi_1$  gire a uma velocidade constante  $\omega$ , portanto,  $\dot{\phi}_1 = \omega t$ . Para tornar a expressão mais geral acrescenta-se o termo  $\phi_{1o}$  que representa uma posição inicial (conhecida) qualquer para a variável  $\phi_1$ .

$$\Phi = \begin{bmatrix} x_1 - \frac{1}{2}L_1 \cos(\phi_1) - X_A \\ y_1 - \frac{1}{2}L_1 \sin(\phi_1) - Y_A \\ x_1 + \frac{1}{2}L_1 \cos(\phi_1) - x_2 + \frac{1}{2}L_2 \cos(\phi_2) \\ y_1 + \frac{1}{2}L_1 \sin(\phi_1) - y_2 + \frac{1}{2}L_2 \sin(\phi_2) \\ x_2 + \frac{1}{2}L_2 \cos(\phi_2) - x_3 + \frac{1}{2}L_3 \cos(\phi_3) \\ y_2 + \frac{1}{2}L_2 \sin(\phi_2) - y_3 + \frac{1}{2}L_3 \sin(\phi_3) \\ x_3 + \frac{1}{2}L_3 \cos(\phi_3) - X_D \\ y_3 + \frac{1}{2}L_3 \sin(\phi_3) - Y_D \\ \phi_1 - \phi_{1o} - \omega t \end{bmatrix} = \mathbf{0} \quad (2.29)$$

O próximo passo é escrever os vetores e matrizes necessários para formar os vetores que permitem calcular a velocidade e a aceleração das coordenadas,  $\nu$  e  $\gamma$ , conforme as Equações 2.12 e 2.17 respectivamente. Neste problema, como  $\dot{\omega} = 0$ , o vetor  $\Phi_t$  será dado pela Eq. 2.30, enquanto que  $\Phi_{tt}$  será um vetor nulo de dimensão  $[1 \times 9]$  e  $\Phi_{qt}$  uma matriz nula  $[9 \times 9]$ . Já a matriz Jacobiana deste problema é mostrada na Equação 2.31.

$$\Phi_t = [0, 0, 0, 0, 0, 0, 0, 0, -\omega]^T \quad (2.30)$$



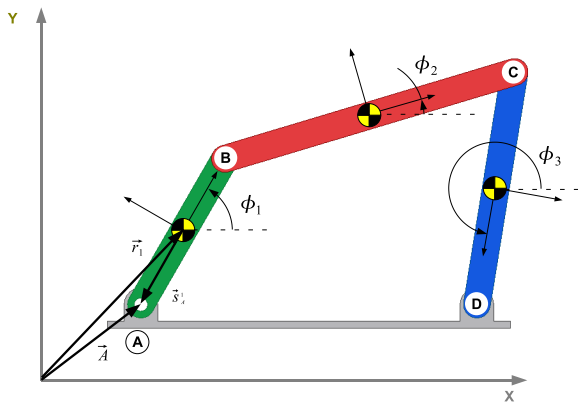
$$\Phi_{\mathbf{q}} = \begin{bmatrix} 1 & 0 & \frac{1}{2}L_1 \sin(\phi_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2}L_1 \cos(\phi_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -\frac{1}{2}L_1 \sin(\phi_1) & -1 & 0 & -\frac{1}{2}L_2 \sin(\phi_2) & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2}L_1 \cos(\phi_1) & 0 & -1 & \frac{1}{2}L_2 \cos(\phi_2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{2}L_2 \sin(\phi_2) & -1 & 0 & -\frac{1}{2}L_3 \sin(\phi_3) \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{2}L_2 \cos(\phi_2) & 0 & -1 & \frac{1}{2}L_3 \cos(\phi_3) \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\frac{1}{2}L_3 \sin(\phi_3) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2}L_3 \cos(\phi_3) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.31)$$

Com esses termos definidos, chega-se ao vetor  $\gamma$  através da Equação 2.17 que, para este problema, tem a forma da expressão a abaixo.

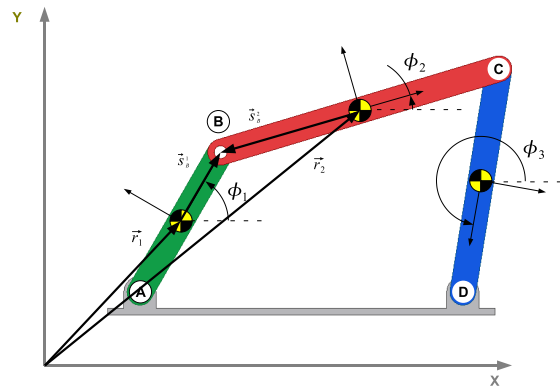
$$\gamma = \begin{bmatrix} -\frac{1}{2}L_1 \cos(\phi_1) \dot{\phi}_1^2 \\ -\frac{1}{2}L_1 \sin(\phi_1) \dot{\phi}_1^2 \\ \frac{1}{2}L_1 \cos(\phi_1) \dot{\phi}_1^2 + \frac{1}{2}L_2 \cos(\phi_2) \dot{\phi}_2^2 \\ \frac{1}{2}L_1 \sin(\phi_1) \dot{\phi}_1^2 + \frac{1}{2}L_2 \sin(\phi_2) \dot{\phi}_2^2 \\ \frac{1}{2}L_2 \cos(\phi_2) \dot{\phi}_2^2 + \frac{1}{2}L_3 \cos(\phi_3) \dot{\phi}_3^2 \\ \frac{1}{2}L_2 \sin(\phi_2) \dot{\phi}_2^2 + \frac{1}{2}L_3 \sin(\phi_3) \dot{\phi}_3^2 \\ \frac{1}{2}L_3 \cos(\phi_3) \dot{\phi}_3^2 \\ \frac{1}{2}L_3 \sin(\phi_3) \dot{\phi}_3^2 \\ 0 \end{bmatrix} \quad (2.32)$$

Resta ainda definir a matriz de massa  $\mathbf{M}$  e o vetor de forças generalizadas<sup>20</sup>  $\mathbf{Q}^A$ . A matriz  $\mathbf{M}$  é uma matriz diagonal  $[9 \times 9]$ , cujos elementos da diagonal principal são dados pela Eq. 2.33 e o vetor  $\mathbf{Q}^A$ , cujos elementos não nulos são as forças peso das três barras, é mostrado na Eq 2.34. Pode-se perceber que como o peso das barras atua na direção

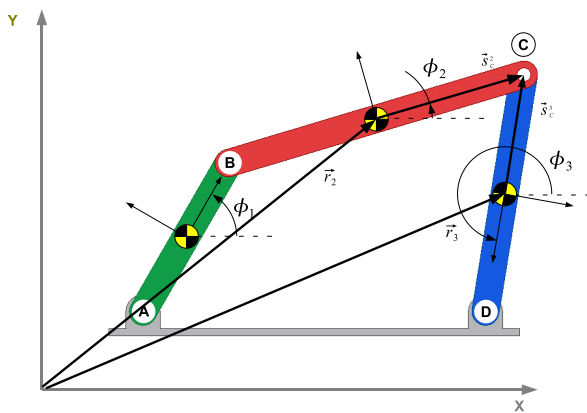
<sup>20</sup>Usa-se aqui o termo Força Generalizada pois é este termo o que HAUG (1989) emprega. Uma definição mais pormenorizada, bem como sua formalização matemática será feita na Seção 3.3.2.1.



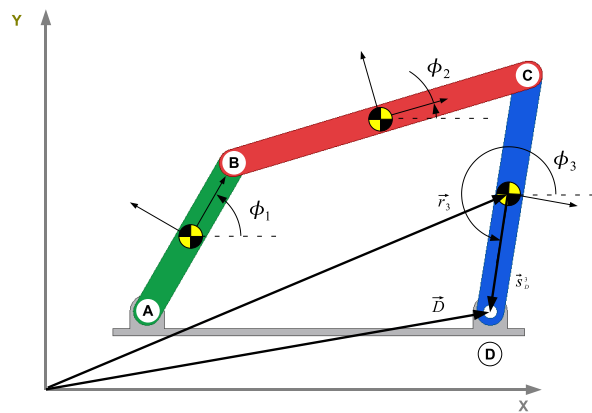
(a) Duas restrições absolutas de distância para determinar a posição do ponto A



(b) Duas restrições para a junta de revolução do ponto B



(c) Duas restrições para a junta de revolução do ponto C



(d) Duas restrições absolutas de distância para determinar a posição do ponto D

FIG. 2.8: Representação gráfica das equações de restrição cinemática do vetor  $\Phi$ .

negativa do eixo Y de coordenadas global, eles aparecem nas posições correspondentes a  $y_1$ ,  $y_2$  e  $y_3$  do vetor  $\mathbf{q}$  com o sinal negativo.

$$\mathbf{M} = \text{diag}(m_1, m_1, J_1, m_2, m_2, J_2, m_3, m_3, J_3) \quad (2.33)$$

$$\mathbf{Q}^A = [0, -m_1g, 0, 0, -m_2g, 0, 0, -m_3g, 0] \quad (2.34)$$

Com todas as matrizes e vetores definidos, é possível expandir a equação de movimento do sistema (Eq. 2.24) tal como mostrado abaixo. Na equação 2.35a o termo  $\ddot{\mathbf{q}}$  não é conhecido explicitamente, mas pode ser calculado, na fase da análise cinemática, através da Equação 2.35b. Portanto, a única incógnita desta equação seria o vetor  $\boldsymbol{\lambda}$ .

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}^A \quad (2.35a)$$

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = \boldsymbol{\gamma} \quad (2.35b)$$

Desta forma, se a matriz Jacobiana  $\Phi_{\mathbf{q}}$  não for singular, o vetor  $\boldsymbol{\lambda}$  pode ser calculado através da Equação 2.36.

$$\boldsymbol{\lambda} = \Phi_{\mathbf{q}}^{T^{-1}} (\mathbf{Q}^A - \mathbf{M}\ddot{\mathbf{q}}) \quad (2.36)$$

Para ajudar a compreender o papel dos Multiplicadores de Lagrange, é conveniente então expandir a Equação 2.24. Nas equações 2.37 é possível perceber que as nove primeiras equações se referem às equações de movimento propriamente ditas, isto é, somatório de forças é igual à massa vezes aceleração e somatório de momentos é igual ao momento de inércia vezes aceleração angular; e correspondem à expansão da Equação 2.35a. Já as nove últimas se referem às equações de restrição de aceleração expressa pela Equação 2.35b.

$$\begin{aligned}
m_1 \ddot{x}_1 + \lambda_1 + \lambda_3 &= 0 \\
m_1 \ddot{y}_1 + \lambda_2 + \lambda_4 + m_1 g &= 0 \\
J_1 \ddot{\phi}_1 + \frac{1}{2} \sin(\phi_1) L_1 \lambda_1 - \frac{1}{2} \cos(\phi_1) L_1 \lambda_2 - \frac{1}{2} \sin(\phi_1) L_1 \lambda_3 + \frac{1}{2} \cos(\phi_1) L_1 \lambda_4 + \lambda_9 &= 0 \\
m_2 \ddot{x}_2 - \lambda_3 + \lambda_5 &= 0 \\
m_2 \ddot{y}_2 - \lambda_4 + \lambda_6 + m_2 g &= 0 \\
J_2 \ddot{\phi}_2 - \frac{1}{2} \sin(\phi_2) L_2 \lambda_3 + \frac{1}{2} \cos(\phi_2) L_2 \lambda_4 - \frac{1}{2} \sin(\phi_2) L_2 \lambda_5 + \frac{1}{2} \cos(\phi_2) L_2 \lambda_6 &= 0 \\
m_3 \ddot{x}_3 - \lambda_5 + \lambda_7 &= 0 \\
m_3 \ddot{y}_3 - \lambda_6 + \lambda_8 + m_3 g &= 0 \\
J_3 \ddot{\phi}_3 - \frac{1}{2} \sin(\phi_3) L_3 \lambda_5 + \frac{1}{2} \cos(\phi_3) L_3 \lambda_6 - \frac{1}{2} \sin(\phi_3) L_3 \lambda_7 + \frac{1}{2} \cos(\phi_3) L_3 \lambda_8 &= 0 \\
\ddot{x}_1 + \frac{1}{2} \sin(\phi_1) L_1 \ddot{\phi}_1 + \frac{1}{2} \cos(\phi_1) L_1 \dot{\phi}_1^2 &= 0 \\
\ddot{y}_1 - \frac{1}{2} \cos(\phi_1) L_1 \ddot{\phi}_1 + \frac{1}{2} \sin(\phi_1) L_1 \dot{\phi}_1^2 &= 0 \\
\ddot{x}_1 - \frac{1}{2} \sin(\phi_1) L_1 \ddot{\phi}_1 - \ddot{x}_2 - \frac{1}{2} \sin(\phi_2) L_2 \ddot{\phi}_2 - \frac{1}{2} \cos(\phi_1) L_1 \dot{\phi}_1^2 - \frac{1}{2} \cos(\phi_2) L_2 \dot{\phi}_2^2 &= 0 \\
\ddot{y}_1 + \frac{1}{2} \cos(\phi_1) L_1 \ddot{\phi}_1 - \ddot{y}_2 + \frac{1}{2} \cos(\phi_2) L_2 \ddot{\phi}_2 - \frac{1}{2} \sin(\phi_1) L_1 \dot{\phi}_1^2 - \frac{1}{2} \sin(\phi_2) L_2 \dot{\phi}_2^2 &= 0 \\
\ddot{x}_2 - \frac{1}{2} \sin(\phi_2) L_2 \ddot{\phi}_2 - \ddot{x}_3 - \frac{1}{2} \sin(\phi_3) L_3 \ddot{\phi}_3 - \frac{1}{2} \cos(\phi_2) L_2 \dot{\phi}_2^2 - \frac{1}{2} \cos(\phi_3) L_3 \dot{\phi}_3^2 &= 0 \\
\ddot{y}_2 + \frac{1}{2} \cos(\phi_2) L_2 \ddot{\phi}_2 - \ddot{y}_3 + \frac{1}{2} \cos(\phi_3) L_3 \ddot{\phi}_3 - \frac{1}{2} \sin(\phi_2) L_2 \dot{\phi}_2^2 - \frac{1}{2} \sin(\phi_3) L_3 \dot{\phi}_3^2 &= 0 \\
\ddot{x}_3 - \frac{1}{2} \sin(\phi_3) L_3 \ddot{\phi}_3 - \frac{1}{2} \cos(\phi_3) L_3 \dot{\phi}_3^2 &= 0 \\
\ddot{y}_3 + \frac{1}{2} \cos(\phi_3) L_3 \ddot{\phi}_3 - \frac{1}{2} \sin(\phi_3) L_3 \dot{\phi}_3^2 &= 0 \\
\ddot{\phi}_1 &= 0
\end{aligned} \tag{2.37}$$

Comparando as nove primeiras linhas da Equação 2.37 com o digrama de corpo livre da Figura 2.9 é possível identificar a relação dos  $\lambda$ 's com as forças internas. Vê-se que, por esta análise, as forças internas são os próprios  $\lambda$ 's (em alguns casos com o sinal negativo), tal como é mostrado abaixo. Fica evidente também que o termo  $\lambda_9$  aparece isolado na terceira equação, e corresponde ao somatório de momentos em torno do centro de massa da barra 1. Isto significa que  $-\lambda_9$  é o torque que deve ser imposto sobre o centro de massa da barra 1 para que esta gire com uma velocidade  $\dot{\phi}_1$  constante.

$$\begin{aligned}
Fx_1 = -\lambda_1 &\rightarrow Fy_1 = -\lambda_2 \rightarrow P_1 = -m_1g \\
Fx_2 = +\lambda_3 &\rightarrow Fy_2 = +\lambda_4 \rightarrow P_2 = -m_2g \\
Fx_3 = -\lambda_5 &\rightarrow Fy_3 = -\lambda_6 \rightarrow P_3 = -m_3g \\
Fx_4 = +\lambda_7 &\rightarrow Fy_4 = +\lambda_8 \\
M_1 = -\lambda_9
\end{aligned}
\tag{2.38}$$

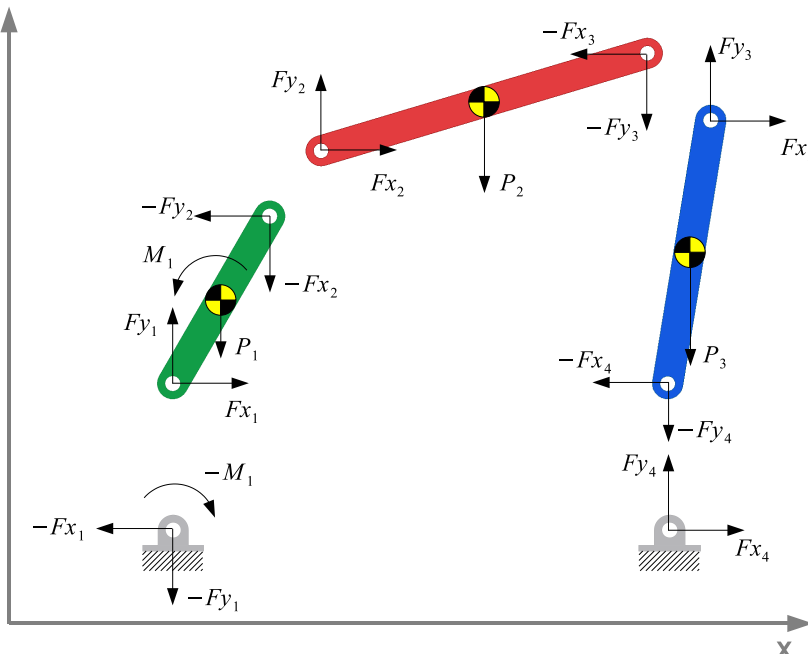


FIG. 2.9: Diagrama de corpo livre do mecanismo 4 barras

No entanto HAUG (1992, p.110) chama a atenção para que, às vezes, os textos sobre dinâmica se referem aos multiplicadores de Lagrange como sendo as forças de restrição, o que não seria correto afirmar. Na mesma seção, o autor usa o Teorema dos Multiplicadores de Lagrange para obter uma expressão explícita para as forças de restrição  $\mathbf{F}^C$  (internas) que não realizam trabalho, que é mostrada na Equação 2.39.

$$\mathbf{F}^C = -\Phi_{\mathbf{q}}^T \boldsymbol{\lambda}
\tag{2.39}$$

A conclusão feita anteriormente parece contradizer o que diz HAUG (1992, p.110). Para fins de ilustração, apresenta-se aqui o vetor  $\mathbf{F}^C$  para este problema. Pelo que pode-se perceber na Eq. 2.40, cada elemento de  $\mathbf{F}^C$  representa a força (ou momento) resultante

agindo sobre o centro de massa de cada barra nas direções  $x$ ,  $y$  ou  $\phi$  e não cada uma das forças internas agindo nas juntas de cada barra. O terceiro elemento do vetor  $\mathbf{F}^C$  por exemplo, como será visto na Seção 2.3.2, é numericamente nulo. Isto mostra que  $\lambda_9$ , neste caso, tem o papel de equilibrar o momento das forças internas  $Fx_1, Fy_1, Fx_2, Fy_2$  em torno do centro de massa.

$$\mathbf{F}^C = \begin{bmatrix} -\lambda_1 - \lambda_3 \\ -\lambda_2 - \lambda_4 \\ -\frac{1}{2} \sin(\phi_1)L_1\lambda_1 + \frac{1}{2} \cos(\phi_1)L_1\lambda_2 + \frac{1}{2} \sin(\phi_1)L_1\lambda_3 - \frac{1}{2} \cos(\phi_1)L_1\lambda_4 - \lambda_9 \\ \lambda_3 - \lambda_5 \\ \lambda_4 - \lambda_6 \\ \frac{1}{2} \sin(\phi_2)L_2\lambda_3 - \frac{1}{2} \cos(\phi_2)L_2\lambda_4 + \frac{1}{2} \sin(\phi_2)L_2\lambda_5 - \frac{1}{2} \cos(\phi_2)L_2\lambda_6 \\ \lambda_5 - \lambda_7 \\ \lambda_6 - \lambda_8 \\ \frac{1}{2} \sin(\phi_3)L_3\lambda_5 - \frac{1}{2} \cos(\phi_3)L_3\lambda_6 + \frac{1}{2} \sin(\phi_3)L_3\lambda_7 - \frac{1}{2} \cos(\phi_3)L_3\lambda_8 \end{bmatrix} \quad (2.40)$$

### 2.3.2 IMPLEMENTAÇÃO COMPUTACIONAL DINÂMICA INVERSA

A análise cinemática e o cálculo da dinâmica inversa foram implementados em Matlab cujo algoritmo é apresentado na Seção C.1. A diferença entre o algoritmo da análise cinemática e o da dinâmica inversa consiste no acréscimo de alguns comandos com os quais se calculam as forças responsáveis por provocar o movimento que foi imposto ao mecanismo. Por esse motivo, a análise cinemática é apresentada junto com a dinâmica inversa e referida ao longo do texto simplesmente como “dinâmica inversa”.

Na primeira parte do algoritmo calculam-se as matrizes e vetores definidos na Seção 2.3.1, utilizando-se os recursos de manipulação de expressões simbólicas do Matlab. Em seguida declara-se o valor numérico das constantes tais como comprimento das barras, parâmetros de inércia, etc. O próximo passo é fornecer uma estimativa inicial para o vetor de coordenadas cartesianas  $\mathbf{q}$ . Esta estimativa foi feita através de um desenho, utilizando-se o *software* de CAD *Autodesk Inventor*.

Até esta etapa, o que se faz é uma fase que pode ser chamada de pré-processamento. A partir daqui, começa o *loop* principal do algoritmo. Inicialmente, deve-se calcular dois

indicadores que irão servir como critério de parada para o algoritmo Newton-Raphson. Um deles é o `erro_eq`, que avalia a norma Euclidiana do vetor  $\Phi$  e o outro é o `erro_sol` que avalia a diferença entre a solução atual  $\Phi_k$  e a solução da iteração anterior  $\Phi_{k-1}$ . Deve-se também determinar o vetor correspondente à malha temporal (chamada de `tt`), que especifica o intervalo de tempo da análise cinemática bem como a sua discretização.

Dentro do *loop* principal, deve-se atualizar a variável de tempo e o elemento do vetor  $\Phi$  (que no algoritmo aparece como PHI) correspondente à restrição diretora. Com isso inicia-se o algoritmo Newton-Raphson que deverá encontrar um vetor  $\mathbf{q}$  que satisfaça o sistema de equações não-lineares  $\Phi(\mathbf{q}, t) = 0$ . O critério de parada estabelece que `erro_eq` e `erro_sol` devem ser menores do que um número pequeno especificado, no caso  $10^{-15}$ .

Em seguida calculam-se os vetores  $\nu$  e  $\gamma$  para calcular o vetor de velocidades (`qp`) e de acelerações (`qpp`). Nesta implementação, as variáveis precedidas da letra `s` são variáveis simbólicas (por exemplo: `sPHI`). Algumas delas estão em função das variáveis de posição (`x1`, `y1`, `phi1`, etc.) enquanto que outras estão em função das variáveis de posição e velocidade. Por isso, é necessário atualizar o valor dessas variáveis a cada iteração. A atualização das variáveis de posição é feita dentro do Newton-Raphson, logo após o “while” enquanto que a atualização das variáveis de velocidade é feita logo após o cálculo do vetor `qp`. Uma vez que o vetor de acelerações das coordenadas cartesianas já foi calculado, o vetor  $\lambda$  é encontrado calculando-se o valor numérico da matriz de massa `sM` e do vetor de forças externas `sQA` e resolvendo-se a Equação 2.36.

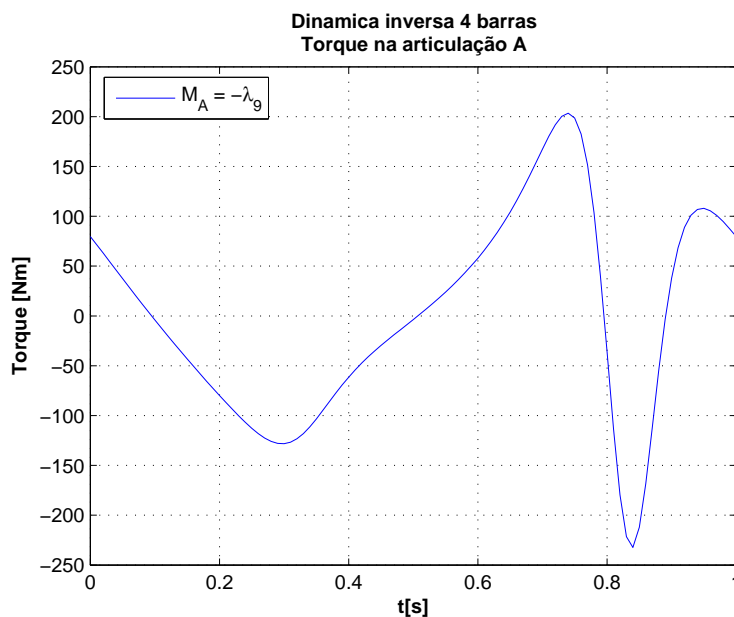


FIG. 2.10: Torque na articulação A necessário para fazer o 4 barras percorrer uma volta completa, com  $\dot{\phi}_1 = 60$  rpm calculado na dinâmica inversa

Os resultados da dinâmica inversa são apresentados nas Figuras 2.10, 2.11 e 2.12. A primeira apresenta o torque necessário para fazer o mecanismo de 4 barras percorrer uma volta completa, com velocidade angular constante  $\dot{\phi}_1 = 60$  rpm, e a segunda as curvas relativas a todos os elementos do vetor  $\boldsymbol{\lambda}$  e  $\mathbf{F}^C$  calculados a cada instante de tempo. Foi mostrado anteriormente (Eqs. 2.38) que o torque  $M_A$  mostrado na Figura 2.10 é igual ao negativo do 9º elemento do vetor  $\boldsymbol{\lambda}$ . Apesar deste torque ter sido obtido a partir da equação de somatório de momentos em torno do centro de massa da barra 1, ele pode ser entendido como o torque de um motor preso ao solo e à barra 1 no ponto A.

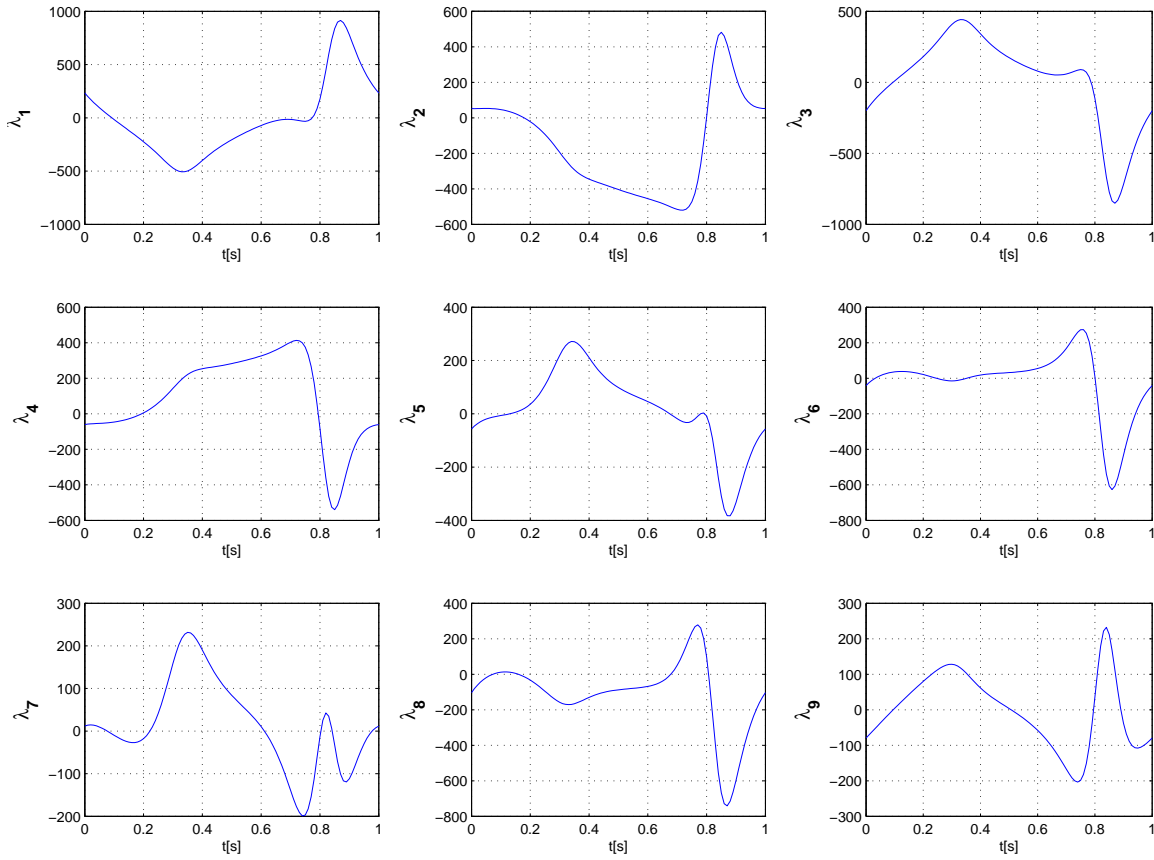


FIG. 2.11: Curvas dos  $\lambda$ 's calculados pela dinâmica inversa no mecanismo 4 barras com atuador simples

### 2.3.3 IMPLEMENTAÇÃO COMPUTACIONAL DINÂMICA DIRETA

Nesta Seção apresenta-se a implementação computacional do algoritmo para integração da DAE que descreve a dinâmica do mecanismo de 4 barras, transformada em ODE. Optou-se por implementar este algoritmo visando checar o resultado obtido na análise cinemática/dinâmica inversa. Isto é, na dinâmica inversa impõe-se um comportamento



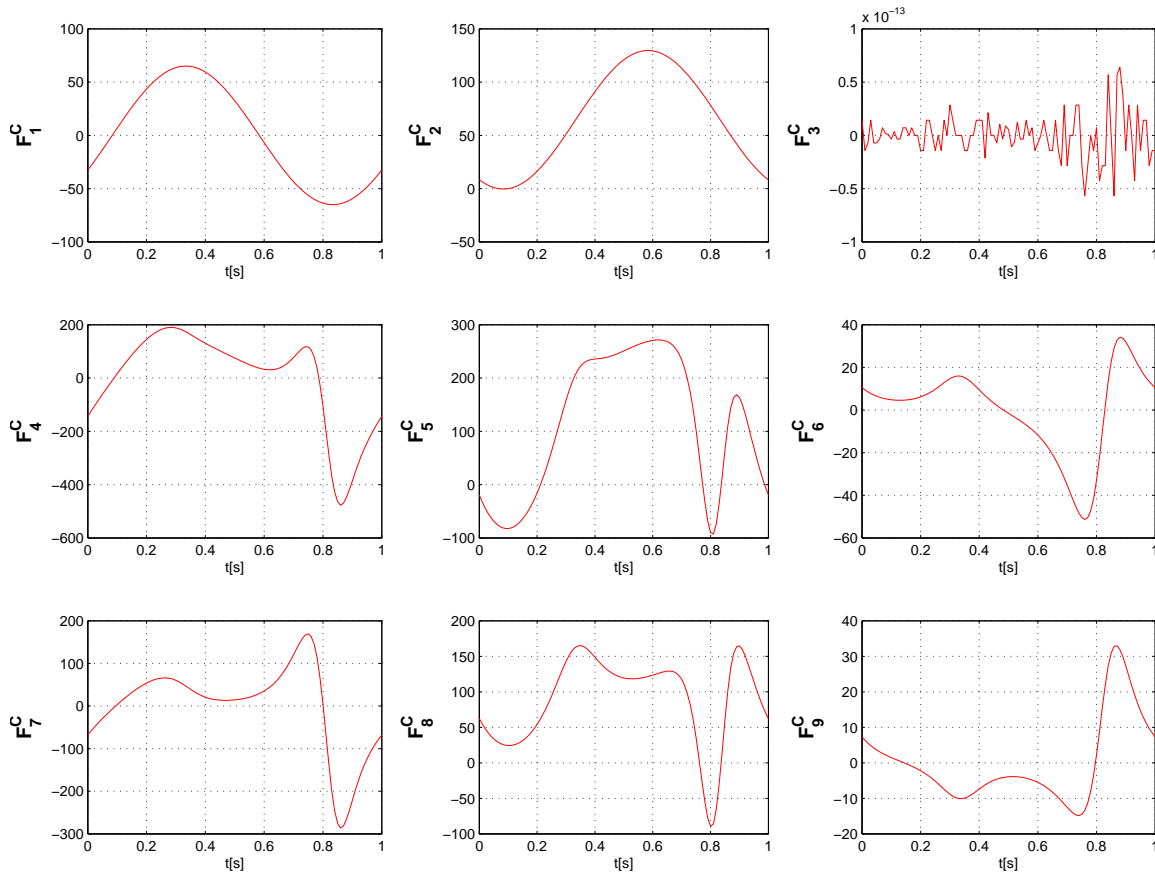


FIG. 2.12: Curvas dos elementos do vetor  $\mathbf{F}^C$  calculados pela dinâmica inversa no mecanismo 4 barras com atuador simples

para uma das variáveis cinemáticas, e a partir disso, calcula-se a posição, velocidade e aceleração das demais, bem como a força ou torque necessários para gerar esse movimento. Na dinâmica direta, impõe-se uma força conhecida e integra-se a equação diferencial com condições iniciais determinadas. Se os resultados da cinemática de ambos os casos são semelhantes, o resultado obtido na dinâmica inversa é consistente.

A primeira parte do algoritmo consiste em calcular os vetores e matrizes definidos na Seção 2.3.1, tal como foi feito para a dinâmica inversa. A diferença aqui, consiste no fato do vetor  $\Phi$  possuir apenas as restrições cinemáticas, isto é, a equação relativa à restrição diretora ( $\phi_1 - \phi_{1o} - \omega t$ ) foi retirada. Isto faz com que o 4 barras tenha um comportamento “pendular” uma vez que o vetor  $\mathbf{Q}^A$  não contém outras forças além do peso das barras. Em seguida os valores das constantes relativas à geometria e aos parâmetros de inércia são declarados. Depois carregam-se as informações obtidas da dinâmica inversa que serão os parâmetros de entrada para a integração: o tempo final, as condições iniciais, e também a curva do momento calculado na articulação A na dinâmica inversa (no caso,  $-\lambda_9$ ).

Com isso, procede-se à integração propriamente dita, utilizando-se o integrador de passo variável ode45 do Matlab. Após a integração, é conveniente proceder a uma etapa de pós-processamento na qual se calculam as acelerações das variáveis de estado (já que estas não são retornadas pelo integrador) e se armazenam os resultados em variáveis mnemônicas para posterior visualização das curvas correspondentes.

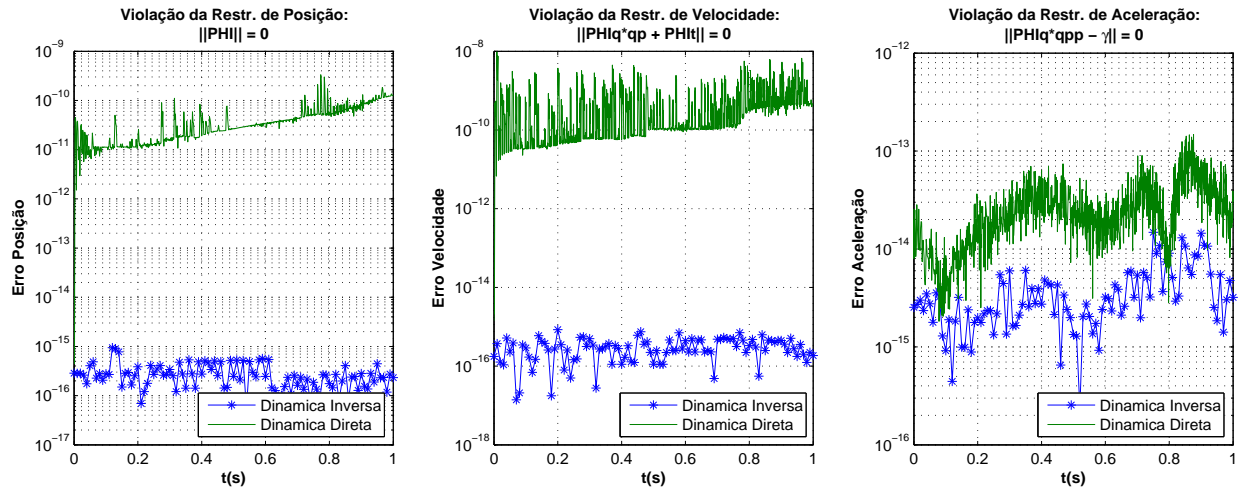


FIG. 2.13: Comparação da violação das restrições de posição, velocidade e aceleração entre os resultados da Dinâmica Inversa e Dinâmica Direta (4 barras com 1 atuador).

As Figuras 2.13, 2.14 e 2.15 mostram que a dinâmica direta conseguiu reproduzir de forma satisfatória a cinemática calculada na dinâmica inversa. Pode-se perceber, na Figura 2.13 que a violação das restrições<sup>21</sup> de posição, velocidade e aceleração na dinâmica inversa ficou abaixo dos  $10^{-14}$ . Isto se deve aos critérios de convergência adotados na solução por Newton-Raphson presente naquele algoritmo. Já na dinâmica direta, o pior resultado foi observado na violação das restrições de velocidade que foi inferior a  $10^{-8}$ , o que pode ser considerado satisfatório.

O que foi revelado na Figura 2.13 pode ser comprovado nas Figuras 2.14 e 2.15, onde se comparam os resultados cinemáticos obtidos na Dinâmica Inversa com aqueles da Dinâmica Direta. Apenas as curvas de  $\dot{\phi}_1$  e  $\ddot{\phi}_1$  obtidas na integração afastam-se um pouco das curvas da dinâmica inversa. O que chama mais atenção é o desvio da curva  $\ddot{\phi}_1$ . Pode-se atribuir isto ao método utilizado para estabilização da DAE durante a integração, pois o Método de Baumgarte é um dos mais limitados. Isto no entanto não invalida os resultados visto que o desvio de  $\dot{\phi}_1$  ficou dentro de uma faixa de  $\pm 1^\circ/s$  e o desvio de  $\phi_1$  foi ainda menor.

<sup>21</sup>Entende-se por violação das restrições como a norma Euclideana do vetor de restrição

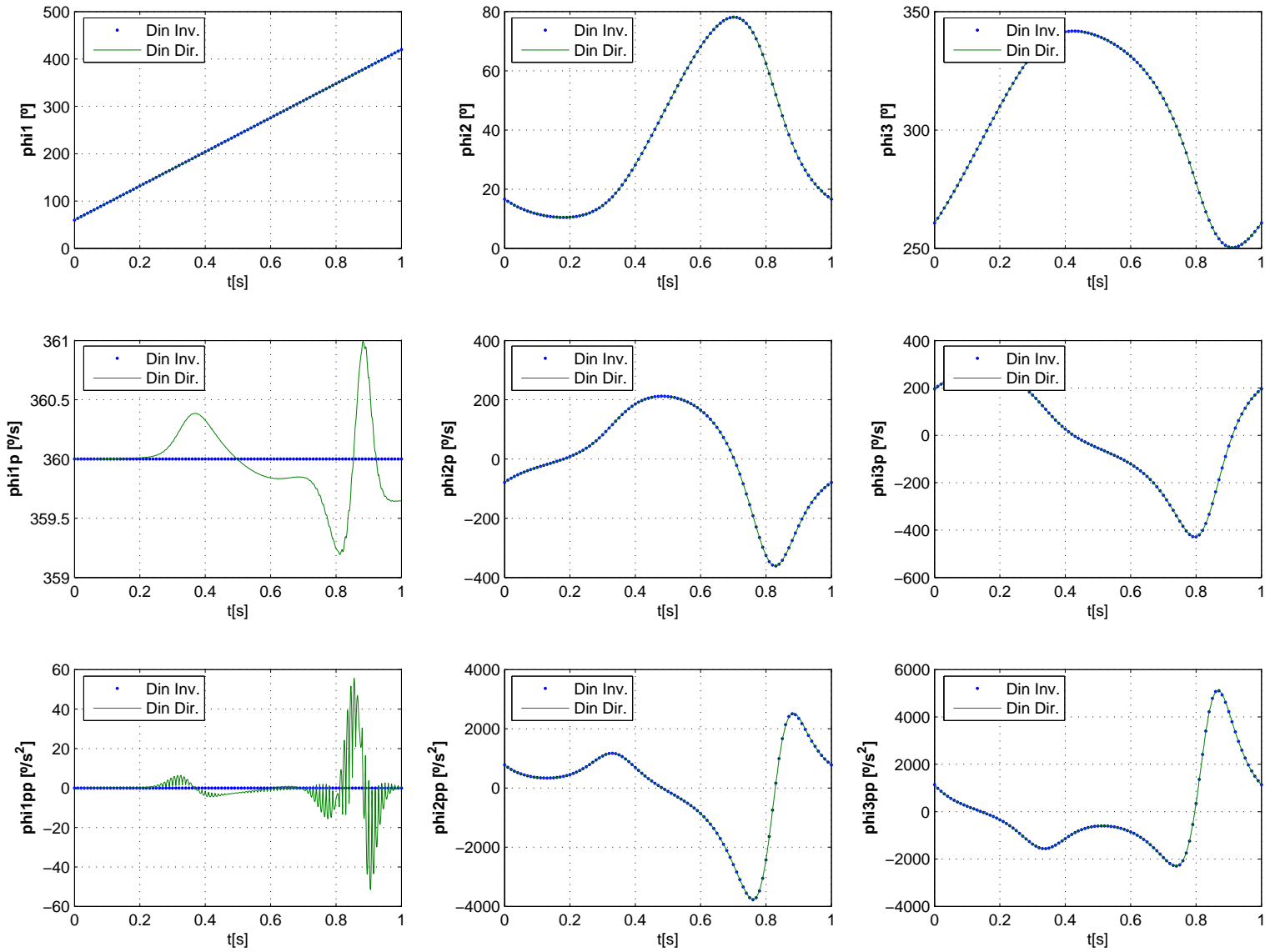


FIG. 2.14: Comparação dos resultados da cinemática obtidos com a Dinâmica Inversa e a Dinâmica Direta (mecanismo 4 barras 1 atuador):  
Variáveis angulares ( $\phi_i$ ).

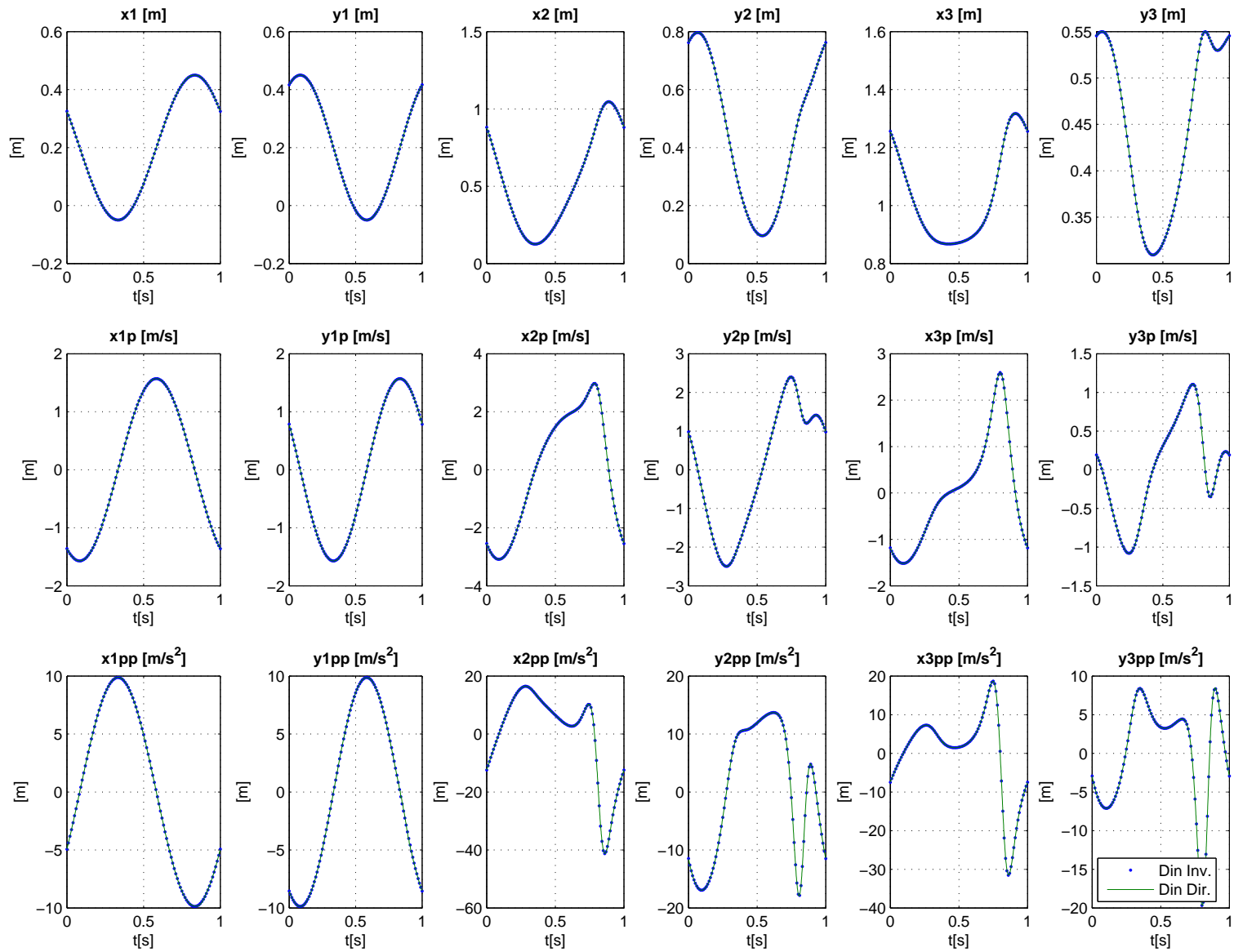


FIG. 2.15: Comparação dos resultados da cinemática obtidos com a Dinâmica Inversa e a Dinâmica Direta (mecanismo 4 barras 1 atuador): Variáveis cartesianas ( $x_i$  e  $y_i$ ).

## 2.4 DINÂMICA INVERSA: MÉTODO MODIFICADO PARA ATUADORES REDUNDANTES

Foi visto na Seção 2.3.1 que a introdução da restrição diretora no último elemento do vetor  $\Phi$  (Eq.2.29) introduz uma variável  $\lambda$  na equação de somatório de momentos em torno da barra 1, conforme foi verificado nas Eqs. 2.37. Este termo foi interpretado como sendo o torque que deve ser aplicado à articulação A para que o mecanismo gire na velocidade constante pré-determinada. Como a matriz  $\Phi_q$  não era singular e o vetor  $\ddot{q}$  já era conhecido da análise cinemática, foi possível encontrar o valor deste  $\lambda$  (no caso,  $\lambda_9$ ) em cada ponto da malha temporal resolvendo-se a Equação 2.36, pois o vetor  $\lambda$  era a única incógnita da equação.

No caso de mecanismos redundantes com atuadores de torque, outras incógnitas deveriam aparecer, relativas aos atuadores em excesso. Além disso, como os atuadores adicionais são fixados *entre* as barras, as equações de somatório de momentos devem exibir tanto a *ação* do torque na barra em que é aplicado, quanto a *reação* deste torque na barra onde o atuador é fixado.

### 2.4.1 MODIFICAÇÃO 1: INTRODUÇÃO DE RESTRIÇÕES DIRETORAS ADICIONAIS AO VETOR $\Phi$

A primeira modificação à metodologia do cálculo da dinâmica inversa a partir das Equações Diferenciais Algébricas, visando a formulação do problema com atuadores redundantes, partiu da hipótese de impor uma restrição diretora adicional ao vetor  $\Phi$ . Neste caso o vetor  $\lambda$  também aumentaria de tamanho e o termo adicional seria o torque (redundante) procurado. Por exemplo, para o caso do mecanismo 4 barras com 2 atuadores nas articulações A e B, o vetor  $\Phi$  teria 10 elementos tal como definido na Eq. 2.41. O nono elemento da Eq. 2.41 é o mesmo da Eq. 2.29, onde  $\omega = \omega_1$ . Já o décimo elemento, representa a restrição diretora adicional onde  $\omega_2 = \dot{\phi}_2$  é a velocidade angular de  $\phi_2$  calculada previamente na análise cinemática do mecanismo 4 barras regular.

$$\Phi_{redun} = \begin{bmatrix} x_1 - \frac{1}{2}L_1 \cos(\phi_1) - X_A \\ y_1 - \frac{1}{2}L_1 \sin(\phi_1) - Y_A \\ x_1 + \frac{1}{2}L_1 \cos(\phi_1) - x_2 + \frac{1}{2}L_2 \cos(\phi_2) \\ y_1 + \frac{1}{2}L_1 \sin(\phi_1) - y_2 + \frac{1}{2}L_2 \sin(\phi_2) \\ x_2 + \frac{1}{2}L_2 \cos(\phi_2) - x_3 + \frac{1}{2}L_3 \cos(\phi_3) \\ y_2 + \frac{1}{2}L_2 \sin(\phi_2) - y_3 + \frac{1}{2}L_3 \sin(\phi_3) \\ x_3 + \frac{1}{2}L_3 \cos(\phi_3) - X_D \\ y_3 + \frac{1}{2}L_3 \sin(\phi_3) - Y_D \\ \phi_1 - \phi_{1o} - \omega_1 t \\ \phi_2 - \phi_{2o} - \omega_2 t \end{bmatrix} = \mathbf{0} \quad (2.41)$$

Com base nesta alteração, foi implementado o algoritmo apresentado na Figura 2.16. Inicialmente, resolve-se a análise cinemática do mecanismo 4 barras tal como mostrado na Seção 2.3.2. Com isso obtém-se as curvas  $\mathbf{q}(t)$  e  $\dot{\mathbf{q}}(t)$ , que são apropriadamente armazenadas. Em seguida, é feita a derivação simbólica dos vetores e matrizes que constituem a equação de movimento do mecanismo 4 barras com atuadores redundantes (EQ.1 mostrada na Figura 2.16). Depois inicia-se o *loop* principal em que a matriz  $\Phi_{\mathbf{q}_{redun}}$  e o vetor  $\gamma_{redun}$  são avaliados com os valores contidos nos vetores  $\mathbf{q}$  e  $\dot{\mathbf{q}}$  calculados previamente, para cada ponto da malha temporal. Com isso é possível constituir a DAE na forma da Eq. 2.24 que pode ser resolvida como um sistema linear, isolando-se o vetor  $[\ddot{\mathbf{q}}_{redun} \ \lambda_{redun}]^T$ . Com isso, o vetor  $[\lambda_{redun}]$  conteria as forças internas bem como os dois torques nas articulações A e B.

Esta abordagem não funcionou devido a um problema de singularidade numérica. A matriz  $[A]$ , indicada na Figura 2.16, apesar de ter dimensão  $[19 \times 19]$ , apresentou posto igual a 18. Com isso, não era possível resolver o sistema linear  $[x] = [A]^{-1}[b]$ . Tentou-se ainda uma outra versão deste método em que apenas o valor de  $\dot{\phi}_2$  era armazenado na análise cinemática inicial e esse valor era passado para  $\omega_2$  num programa para resolver a dinâmica inversa, muito semelhante ao apresentado na Seção 2.3.2, onde o vetor  $\Phi$  é dado pela Eq. 2.41. Nesta abordagem, a violação das restrições de posição crescia rapidamente de tal forma que, logo nos primeiros instantes de tempo, a rotina de Newton-Raphson deixava de convergir.

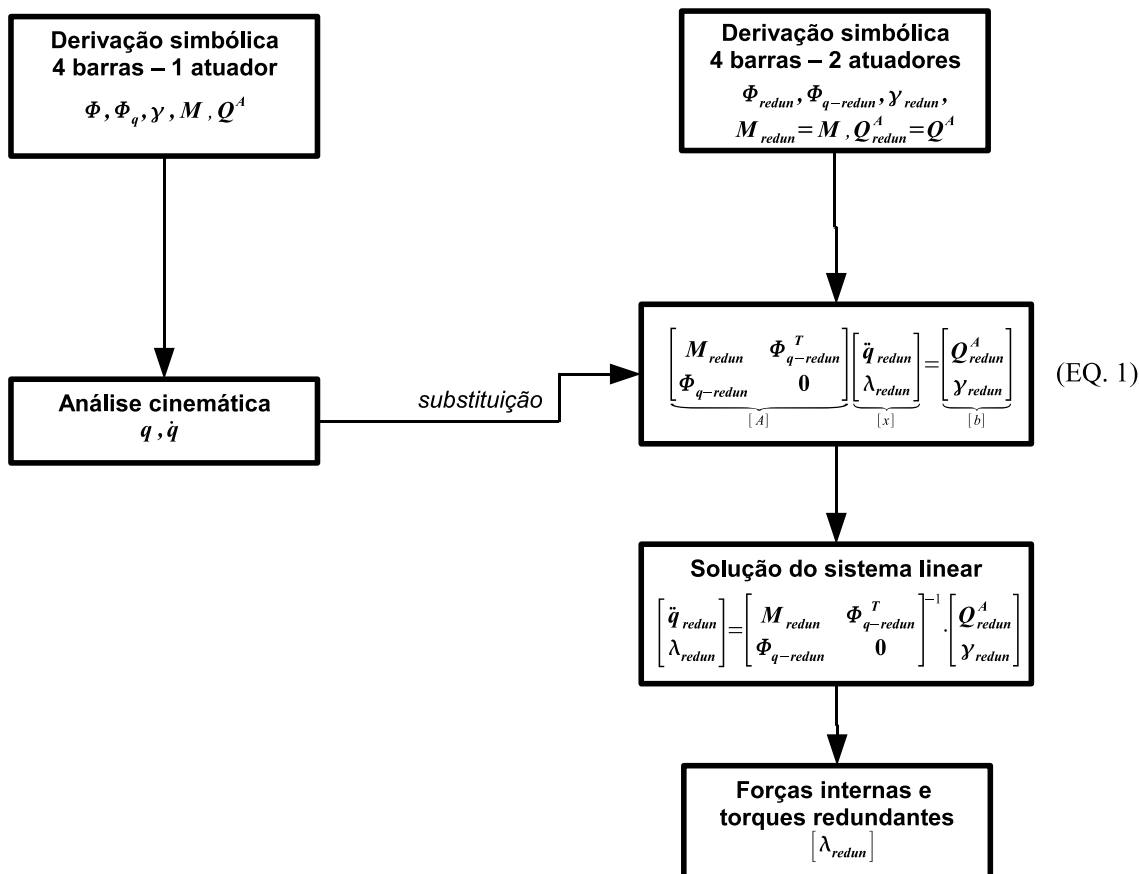


FIG. 2.16: Diagrama de blocos do primeiro algoritmo proposto para resolver a dinâmica inversa do 4 barras com atuadores redundantes a partir da formulação DAE.

#### 2.4.2 MODIFICAÇÃO 2: INTRODUÇÃO DA MATRIZ DE TRANSMISSÃO [T]

A segunda tentativa de modificação da DAE se baseou no artigo apresentado por VALASEK e SIKÁ (2001). Neste trabalho, os autores se ocupam em tratar o que chamam de “Problema Dinâmico Global” para sistemas multi-corpos com atuadores redundantes descritos através de DAE. O Problema Dinâmico Global seria uma terceira formulação, além das duas outras discutidas no início deste Capítulo (Dinâmica Direta e Inversa) que tem o objetivo de avaliar a “capacidade dinâmica” de um sistema mecânico. Ou seja, investigar os valores acessíveis das posições, velocidades e acelerações, bem como a relação desses valores acessíveis com as forças que devem ser fornecidas para se atingir esses valores.

Na solução do Problema Dinâmico Global, VALASEK e SIKÁ (2001) apresentam uma modificação da Equação 2.24 para levar em conta o excesso de atuadores. Esta modificação consiste em acrescentar o termo  $\mathbf{T} \mathbf{n}$  no lado direito da Equação 2.23a,

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}^A + \mathbf{T}\mathbf{n} \quad (2.42)$$

onde  $\mathbf{n}$  é o vetor de torques (ou forças) dos atuadores e  $\mathbf{T}$  é a chamada “Matriz de Transmissão”. Esta matriz depende do tipo de coordenada escolhida e também do posicionamento dos atuadores. Se o conjunto de coordenadas escolhido coincidir com as coordenadas sobre as quais os atuadores agem diretamente,  $\mathbf{T}$  é uma matriz identidade. Caso contrário, a matriz é definida por  $\mathbf{T} = \partial \mathbf{s} / \partial \mathbf{q}^T$ , onde  $\mathbf{s}$  é o vetor de coordenadas sobre as quais os atuadores agem diretamente e  $\mathbf{q}$  é o vetor de coordenadas para descrever o sistema. Apesar de VALASEK e SIKÁ (2001) apontarem o livro de STEJSKAL e VALASEK (1996) ao fazer referência a esta equação, o livro não aborda diretamente o problema da dinâmica inversa de mecanismos com atuadores redundantes. No entanto, o mesmo artigo mostra que é possível reescrever a Equação 2.42 na forma<sup>22</sup>:

$$[\Phi_{\mathbf{q}}^T, -\mathbf{T}] \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{n} \end{bmatrix} = \mathbf{Q}^A - \mathbf{M}\ddot{\mathbf{q}} \quad (2.43)$$

Pode-se perceber que procedendo desta maneira todas as incógnitas do problema ficam reunidas no mesmo vetor coluna. Assim, definindo-se a matriz  $[\mathbf{P}] = [\Phi_{\mathbf{q}}^T, -\mathbf{T}]$ , formada pela justaposição das matrizes  $[\Phi_{\mathbf{q}}]^T$  e  $[-\mathbf{T}]$ , e o vetor  $[\mathbf{G}] = \mathbf{Q}^A - \mathbf{M}\ddot{\mathbf{q}}$ , é possível encontrar as forças internas e o torque nos atuadores resolvendo-se o sistema:

$$\mathbf{X} = \mathbf{P}^{+1} \mathbf{G} \quad (2.44)$$

onde  $\mathbf{P}^{+1}$  é a matriz pseudo-inversa de Moore-Penrose de  $\mathbf{P}$  e pode ser calculada através da Equação 2.45.

$$\mathbf{P}^{+1} = \mathbf{P}^T (\mathbf{P} \cdot \mathbf{P}^T)^{-1} \quad (2.45)$$

Deve-se observar que a matriz  $\mathbf{P}$  possui mais colunas do que linhas e portanto o sistema linear da Equação 2.44 não pode ser resolvido pelos métodos convencionais. Uma propriedade importante da matriz pseudo-inversa é a de que a solução do sistema da Eq. 2.44 é equivalente ao problema de otimização definido na Eq. 2.47. Ou seja,  $\mathbf{X}$

---

<sup>22</sup>A forma da Eq. 2.43 aparece com alguns sinais trocados em relação a fórmula que consta em VALASEK e SIKÁ (2001) porque estes autores definem a Eq. 2.42 com o sinal do termo  $\Phi_{\mathbf{q}}^T \boldsymbol{\lambda}$  trocado.



minimiza a norma Euclidiana, ao mesmo tempo que obedece o sistema linear  $\mathbf{P} \cdot \mathbf{X} = \mathbf{G}$ . Uma revisão sucinta e objetiva sobre a definição, propriedades e métodos numéricos de obtenção da matriz pseudo-inversa está disponível em DE FALCO et al. (2005).

$$\min: g \equiv \|\mathbf{X}\|_2^2 \quad (2.46)$$

$$\text{s.a: } \mathbf{P} \cdot \mathbf{X} = \mathbf{G} \quad (2.47)$$

### Implementação Computacional

Com base nas equações apresentadas anteriormente, uma modificação do algoritmo apresentado na Seção 2.3.2 foi introduzida para que se calculasse a dinâmica inversa do mecanismo 4 barras com 1, 2, 3 e 4 atuadores. Desta forma, foi necessário definir as matrizes de transmissão  $\mathbf{T}_1 \dots \mathbf{T}_4$ , bem como o vetor  $\mathbf{n} = [n_1 \ n_2 \ n_3 \ n_4]^T$ . A matriz  $\mathbf{T}_1$  é uma matriz nula de dimensão  $[9 \times 4]$  enquanto que as demais são mostradas na Eq. 2.48.

$$\mathbf{T}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{T}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{T}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (2.48)$$

Os elementos não nulos das matrizes  $\mathbf{T}_2$ ,  $\mathbf{T}_3$  e  $\mathbf{T}_4$  foram selecionados de forma a explicitar os torques  $n_2$ ,  $n_3$  e  $n_4$  nas equações de somatório de momentos de cada uma das barras (Eq. 2.37). Por exemplo, o elemento  $\mathbf{T}_2(6, 2)$  é igual a 1 porque o atuador  $n_2$  (fixado na articulação B) aplica um torque positivo sobre a coordenada  $\phi_2$ , que descreve o ângulo da barra 2. Como o ângulo  $\phi_2$  é o sexto elemento do vetor  $\mathbf{q}$ , o elemento não nulo da matriz também deverá aparecer na sexta linha. Por outro lado, esse mesmo atuador que age sobre a barra 2, tem sua reação sobre a barra 1. Portanto, deve-se aplicar um torque na direção contrária sobre a barra 1, cuja orientação angular é determinada pelo ângulo  $\phi_1$  – não por acaso – o terceiro elemento de  $\mathbf{q}$ . Daí  $\mathbf{T}_2(3, 2) = -1$ . O mesmo raciocínio pode ser aplicado para explicar os demais elementos não nulos das matrizes  $\mathbf{T}_3$  e  $\mathbf{T}_4$ .

Percebe-se ainda que a primeira coluna nessas três matrizes é nula. Poder-se-ia perguntar por que não estariam presentes elementos não-nulos também na primeira coluna dessas matrizes, visto que esta se refere ao atuador na articulação A, que está presente em todos os casos. Na presente formulação isto não se faz necessário pois o torque em A é calculado através do vetor  $\boldsymbol{\lambda}$ , que é um sub-vetor de  $\mathbf{X}_i$ . Mais precisamente, o torque na articulação A aparece no termo  $\lambda_9$  (mostrado na Eq.2.49) do vetor  $\mathbf{X}_i$  que é comum em todos os casos.

$$\mathbf{X}_i = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_9 \\ n_1 \\ \vdots \\ n_4 \end{bmatrix} \quad (2.49)$$

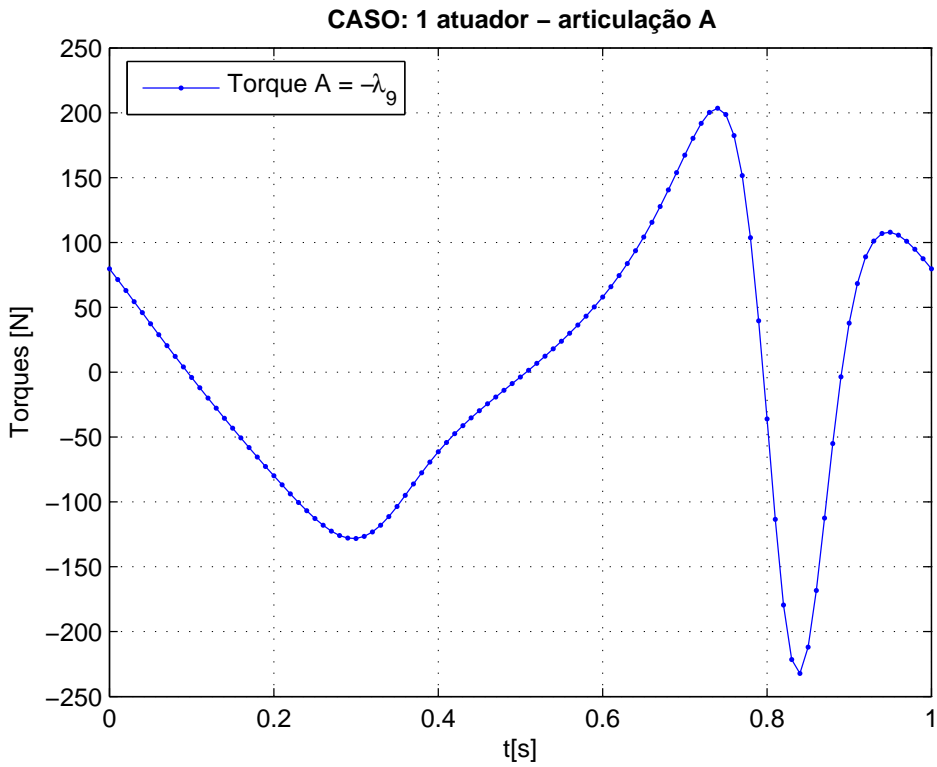
Assim, além de introduzir as matrizes  $\mathbf{T}_1 \dots \mathbf{T}_4$ , foi necessário apenas reescrever as matrizes do sistema na forma da Equação 2.43 e resolver a Equação 2.44 em cada instante de tempo para adaptar o algoritmo apresentado na Seção 2.3.2, de forma a obter uma solução da dinâmica inversa do mecanismo de 4 barras com até 4 atuadores redundantes. Os torques calculados desta maneira são apresentados nas Figuras 2.17 e 2.18 .

## 2.5 ANÁLISE DOS RESULTADOS

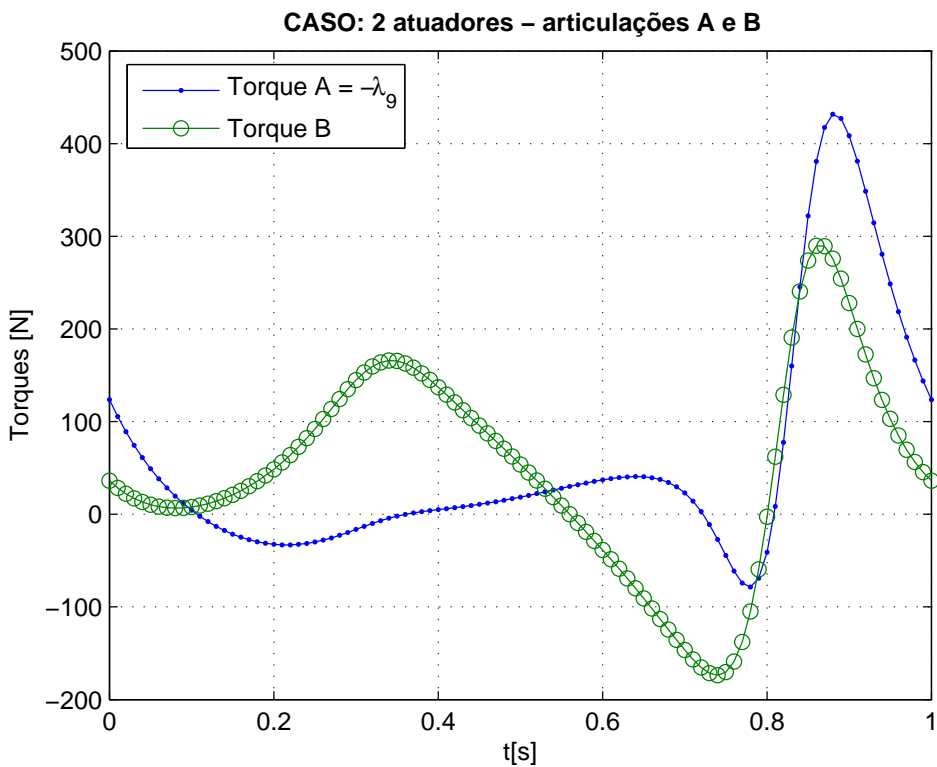
### 2.5.1 ANÁLISE DO TORQUE MÁXIMO DOS ATUADORES

Na Figura 2.17(a) apresenta-se o torque para o caso com apenas 1 atuador na articulação A. Deve-se ressaltar que o torque apresentado aqui para o caso com 1 atuador não é uma repetição do resultado apresentado na Figura 2.10. Enquanto que aquele torque foi calculado resolvendo-se a Eq. 2.36, a curva da Figura 2.17(a) foi obtida com a matriz pseudo-inversa da Eq. 2.44. Como a pseudo-inversa de uma matriz tem a propriedade de coincidir com a sua inversa quando ela é quadrada e de posto completo, a solução da Eq. 2.44 coincide com a solução da Eq. 2.36.

Pode-se perceber nas Figuras 2.17 e 2.18 que os torques máximos em geral aumentaram em relação ao caso em que havia apenas 1 atuador. Enquanto que para o 4 barras com 1 atuador o torque máximo (em módulo) foi de 232 Nm, os torques máximos para os

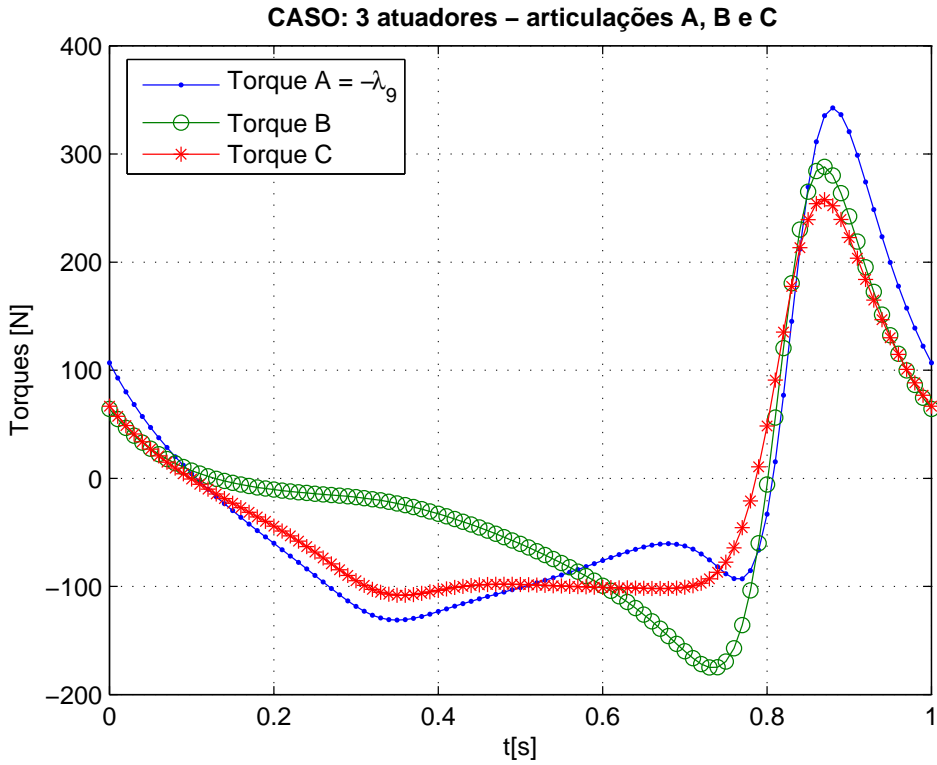


(a) Um atuador posicionado na articulação A

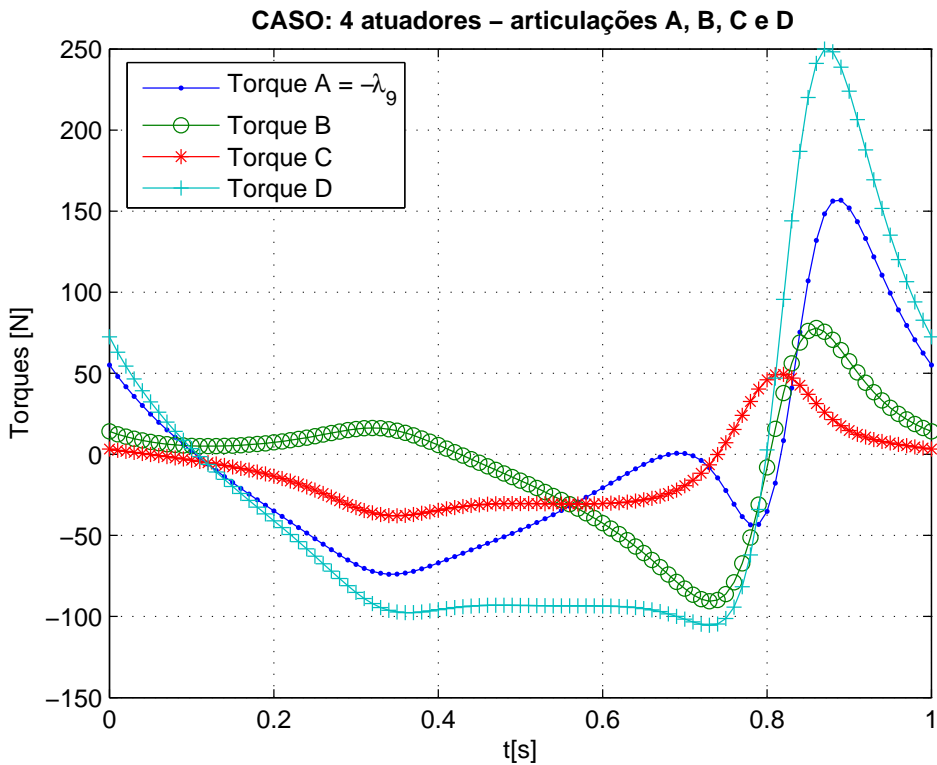


(b) Dois atuadores posicionados nas articulações A e B

FIG. 2.17: Comparação dos torques calculados com a dinâmica inversa formulada a partir das DAE's para o mecanismo 4 barras: 1 e 2 atuadores



(a) Três atuadores posicionados nas articulações A, B e C



(b) Quatro atuadores posicionados nas articulações A, B, C e D

FIG. 2.18: Comparação dos torques calculados com a dinâmica inversa formulada a partir das DAE's para o mecanismo 4 barras: 3 e 4 atuadores

mecanismos com 2, 3 e 4 atuadores foram de 431 Nm, 342 Nm e 249 Nm, respectivamente. Ou seja, o mecanismo com atuadores redundantes necessitaria de motores com uma especificação de torque máximo maior (e portanto, mais pesados) do que o caso tradicional. Portanto, **a solução do problema da dinâmica inversa através da formulação DAE modificada pela introdução da Matriz de Transmissão não apresentou vantagens do ponto de vista da redução dos torques máximos.** Os dados de torque máximo e mínimo nos quatro casos apresentados são detalhados na Tabela 2.2.

TAB. 2.2: Comparação dos torques máximos e mínimos dos atuadores durante 1 ciclo completo do mecanismo 4 barras na formulação DAE (Valores em Nm).

	$n_1$		$n_2$		$n_3$		$n_4$	
	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>
<b>1 atuador</b>	203	-232	-	-	-	-	-	-
<b>2 atuadores</b>	431	-78	289	-173	-	-	-	-
<b>3 atuadores</b>	342	-131	288	-174	257	-108	-	-
<b>4 atuadores</b>	156	-73	77	-90	49	-37	249	-105

## 2.5.2 ANÁLISE DO TRABALHO DOS ATUADORES

A segunda parte da análise dos resultados diz respeito ao trabalho executado pelos atuadores nos casos em questão. Antes de mostrar resultados e conclusões, faz-se necessário uma breve revisão e formalização das equações envolvidas no cálculo do trabalho de uma força ou torque.

### Cálculo do trabalho

Segundo MERIAN e KRAIGE (1999), o trabalho realizado por uma força é definido como a integral do produto escalar dessa força pelo deslocamento infinitesimal do seu ponto de aplicação, tal como mostra a Equação 2.50.

$$W_{1-2} = \int_1^2 \mathbf{F} \cdot d\mathbf{r} \quad (2.50)$$

Se a força em consideração for um torque (binário), considera-se apenas o produto simples entre este torque e o deslocamento angular infinitesimal (Eq. 2.51). Desta forma, se os sinais de  $M$  e  $d\theta$  são os mesmos, o torque é aplicado na direção do deslocamento, o trabalho é positivo e portanto a energia é “entregue” ao corpo. Mas se os sinais de  $M$  e

$d\theta$  são contrários, o torque é aplicado na direção oposta a do deslocamento, o que resulta num trabalho negativo e portanto a energia é “retirada” ao corpo. Neste caso, considera-se que o atuador está tendo um papel de “gerador”.

$$W_{1-2} = \int_1^2 M \cdot d\theta \quad (2.51)$$

APOSTOL (1967) define na pág. 115 três propriedades do trabalho como integral de uma força: a propriedade aditiva, a monotônica e a da fórmula elementar. Esta última diz que se a força  $f$  é constante (por exemplo,  $f(x) = c$ ), para todo  $x$  no intervalo aberto  $(a, b)$ , então o trabalho da força  $f$  de  $a$  até  $b$  é dado por  $W_a^b(f) = c \cdot (b - a)$ . Esta propriedade, juntamente com a propriedade aditiva, permite que este conceito seja estendido para qualquer número finito de intervalos. Ou seja, se a força considerada é uma função degrau  $s$  que toma valores constantes no intervalo aberto  $(x_{k-1}, x_k)$ , a propriedade da fórmula elementar diz que  $W_k = s_k \cdot (x_k - x_{k-1})$ , portanto teremos:

$$W_a^b(s) = \sum_{k=1}^n s_k \cdot (x_k - x_{k-1}) = \int_a^b s(x) \cdot dx \quad (2.52)$$

A definição do trabalho de uma força como um somatório, tal como na Equação 2.52 é adequada para o cálculo computacional desta grandeza. Portanto, considerando  $\text{phi1}$ ,  $\text{phi2}$ ,  $\text{phi3}$ ,  $\text{n1}$ ,  $\text{n2}$ ,  $\text{n3}$ ,  $\text{n4}$ , vetores com  $N$  elementos que representam respectivamente os ângulos das coordenadas generalizadas (em rad) e os torques nas articulações A, B, C e D (em Nm); os trabalhos dos atuadores durante 1 ciclo completo do mecanismo 4-barras podem ser calculados através do seguinte algoritmo:

```

r1 = phi1;
r2 = phi2 - phi1;
r3 = phi3 - phi2;
r4 = phi3;
W1 = 0; W2 = 0; W3 = 0; W4 = 0;
for i=1:(N-1)
    W1 = W1 + n1(i)*(r1(i+1) - r1(i));
    W2 = W2 + n2(i)*(r2(i+1) - r2(i));
    W3 = W3 + n3(i)*(r3(i+1) - r3(i));
    W4 = W4 + n4(i)*(r4(i+1) - r4(i));
end
Wtot = W1 + W2 + W3 + W4;
```

## Análise do resultado

A Tabela 2.3 mostra o resultado do trabalho executado por cada um dos atuadores em 1 ciclo completo ( $\phi_1 = 60^\circ$  até  $\phi_1 = 420^\circ$ ) do mecanismo 4 barras, calculado a partir deste algoritmo. A primeira linha da tabela mostra que o trabalho executado pelo atuador na articulação A, para o 4 barras regular, foi numericamente nulo. Este resultado é coerente com a relação Trabalho-Energia <sup>23</sup> (Eq. 2.53), pois a formulação do modelo na dinâmica inversa não incorpora nenhuma força dissipativa e considera-se um ciclo completo do mecanismo. Como posição e velocidade iniciais e finais são iguais, então  $\Delta T = 0$  e  $\Delta V = 0$ . Se o trabalho total realizado por uma força externa também foi nulo ( $W_i^f = 0$ ), isso implica que  $W_{nc} = 0$  de forma que não existem forças não conservativas atuando sobre o sistema. Portanto, por exclusão, no caso ideal pode-se assumir o torque do atuador do mecanismo 4 barras como sendo conservativo.

$$W_i^f = \Delta T + \Delta V + W_{nc} \quad (2.53)$$

Nos casos com 2, 3 e 4 atuadores a última coluna da Tabela 2.3 mostra que o trabalho total não foi desprezível. Como a variação de energia cinética e potencial entre a posição final e inicial do movimento é nula, então a Eq.2.53 mostra que o trabalho total dos atuadores deve ser atribuído a um trabalho de forças não conservativas, portanto,  $W_i^f = W_{nc} = W_{tot}$ . A Tabela mostra também que  $W_{tot}$  aumenta com o número de atuadores.

TAB. 2.3: Trabalho dos atuadores durante 1 ciclo completo do mecanismo 4 barras nos casos estudados com a formulação DAE

	<b>W<sub>1</sub></b> (J)	<b>W<sub>2</sub></b> (J)	<b>W<sub>3</sub></b> (J)	<b>W<sub>4</sub></b> (J)	<b>W<sub>tot</sub></b> (J)
<b>1 atuador</b>	6,84E-13	0,0	0,0	0,0	6,84E-13
<b>2 atuadores</b>	355,2	-342,3	0,0	0,0	12,9
<b>3 atuadores</b>	-72,7	-147,5	241,0	0,0	20,9
<b>4 atuadores</b>	-29,9	-11,5	37,6	27,1	23,3

### 2.5.3 VERIFICAÇÃO DOS RESULTADOS

Para checar se os resultados estão corretos, o algoritmo de integração apresentado na Seção 2.3.3 foi adaptado para integrar as equações de movimento do mecanismo de 4

<sup>23</sup>A relação Trabalho-Energia mostra que o trabalho total das forças externas que atuam sobre um sistema entre uma posição inicial e outra final quaisquer, deve ser igual à soma da variação da energia cinética mais a variação da energia potencial mais o trabalho das demais forças não conservativas que atuam sobre o sistema.

barras quando submetido aos torques apresentados nas Figuras 2.17 e 2.18. A principal alteração foi feita na *function*, utilizada pelo ODE45, que retorna a derivada do vetor de estados. Nela, o vetor **QA** foi corrigido com a adição do produto da matriz de transformação (Eqs. 2.48) pelo vetor de torques nos atuadores **n**. Este vetor recebe os torques previamente calculados pelo algoritmo da dinâmica inversa. Um exemplo desta *function* para o caso com 4 atuadores é apresentado na Seção C.3.3.

O resultado dessa checagem é apresentado nas Figuras 2.19, 2.20 e 2.21. Nelas, a cinemática das coordenadas angulares calculadas pelo algoritmo de dinâmica inversa modificado pela introdução da matriz **[T]** é confrontada com a cinemática das mesmas coordenadas obtidas através da integração da equação de movimento do mecanismo de 4 barras quando submetidas aos torques calculados pela dinâmica inversa. Pode-se perceber que, assim como na Figura 2.14, onde essa mesma análise foi feita para o caso com 1 atuador, só é possível distinguir alguma diferença nos resultados para as curvas de  $\dot{\phi}_1$  e  $\ddot{\phi}_1$ , nos três casos apresentados. Assim como aconteceu na Seção 2.3.3 a curva de  $\ddot{\phi}_1$  foi a que mais se desviou do esperado o que pode ser atribuído ao método de estabilização da DAE utilizado durante a integração da equação de movimento. Como o desvio máximo para  $\dot{\phi}_1$  foi de  $2,7^\circ/s$  na Figura 2.21, e o desvio para  $\phi_1$  foi desprezível, os resultados obtidos podem ser considerados aceitáveis.



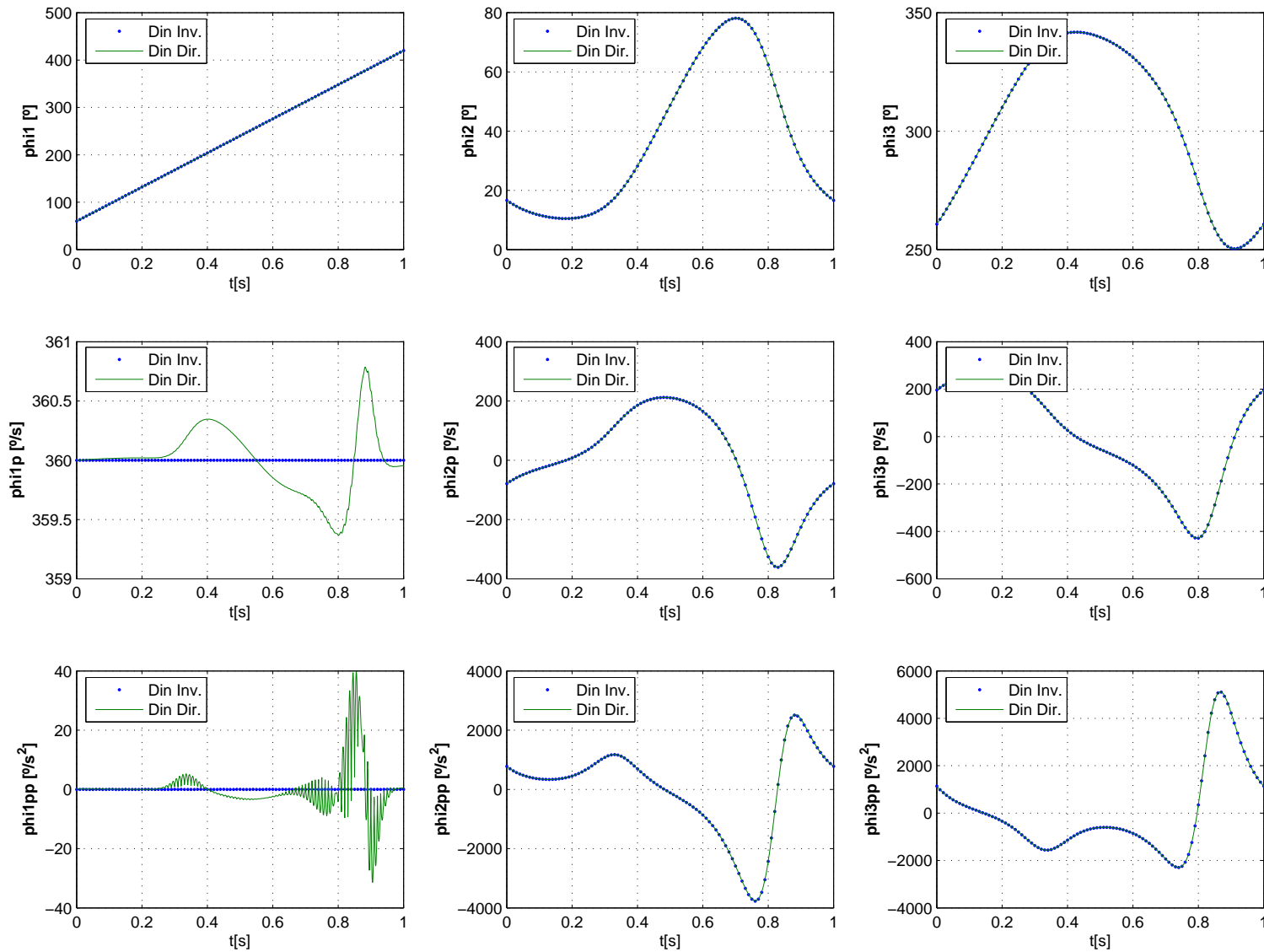


FIG. 2.19: Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz  $[T]$  e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 2 atuadores.

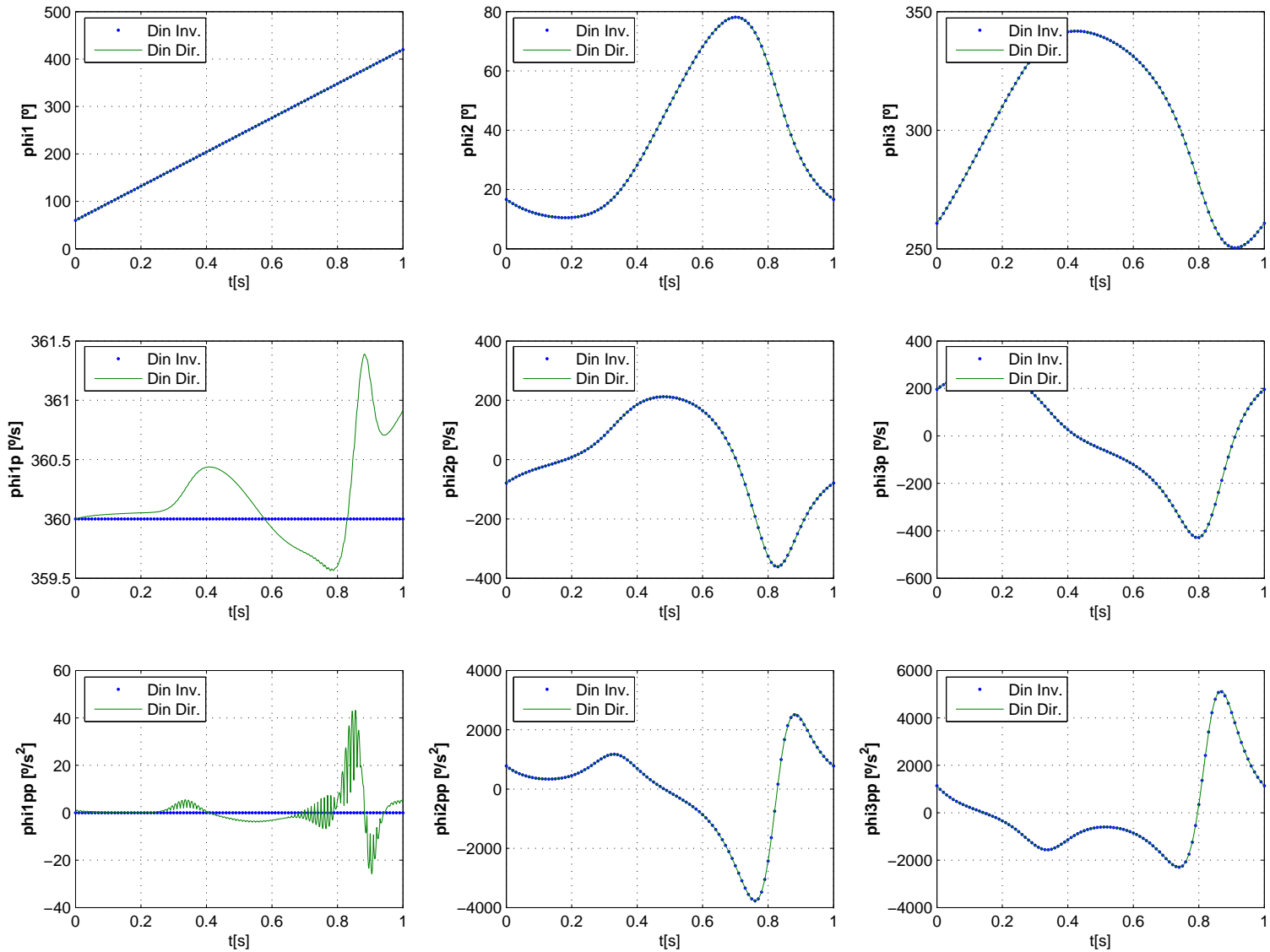


FIG. 2.20: Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz  $[T]$  e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 3 atuadores.

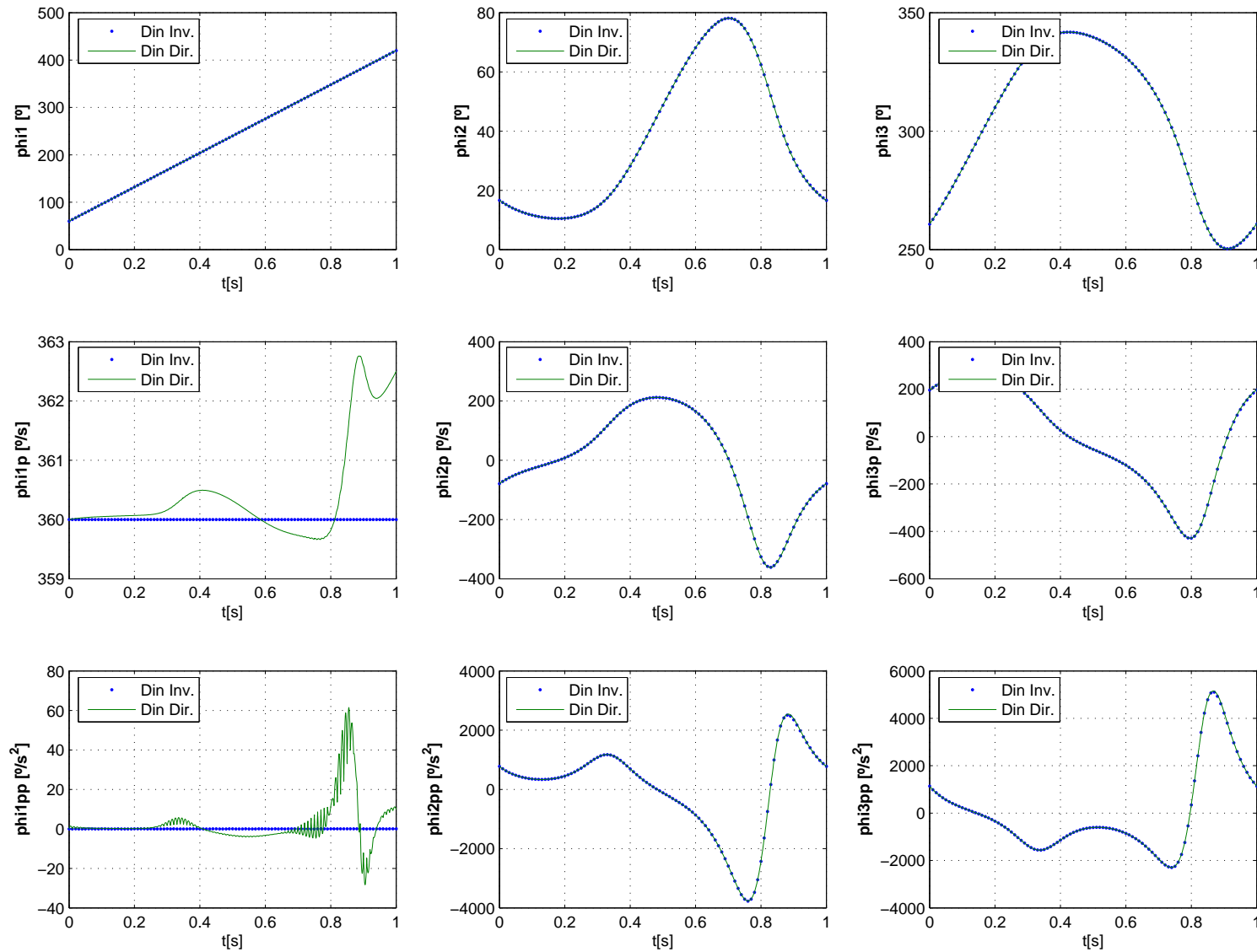


FIG. 2.21: Comparação da cinemática das coordenadas angulares entre a Dinâmica Inversa modificada pela introdução da matriz  $[T]$  e a Dinâmica Direta atuada pelos torques calculados com a Dinâmica Inversa: caso com 4 atuadores.

### 3. SOLUÇÃO DA DINÂMICA INVERSA DO MECANISMO 4 BARRAS COM ATUADORES REDUNDANTES POR CONTROLE ÓTIMO

#### 3.1 FORMULAÇÃO GENÉRICA DO PROBLEMA DE CONTROLE ÓTIMO

Um problema de controle ótimo (ou OCP, da sigla em inglês “Optimal Control Problem”) pode ser formulado segundo (CITRON (1969); MENEGALDO (2001); BOTTASSO e CROCE (2004)), como o problema da determinação dos estados ( $\mathbf{x}^*$ ) e controles ótimos ( $\mathbf{u}^*$ ) e eventualmente o tempo final  $T$  que minimiza uma função de custo  $G_0$  - Eq.(3.1a). Esta minimização deve obedecer às equações de estado Eq.(3.1b) que descrevem a dinâmica do sistema. Várias outras possíveis restrições podem estar presentes, dependendo do problema em questão, como por exemplo, restrições de trajetória das variáveis de estado Eq. (3.1f), condições de contorno Eqs.(3.1c) e Eq.(3.1d), e ainda limites superiores e inferiores para as variáveis de controle Eq. (3.1e).

**Função Objetivo:**

$$\min: G_o(\mathbf{x}, \mathbf{u}, t) = g_o(\mathbf{x}, \mathbf{u}, t) + \int_0^T f_o(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (3.1a)$$

**Equação de Movimento:**

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (3.1b)$$

**Condição Inicial:**

$$\mathbf{x}(0) = \mathbf{x}^0 \quad (3.1c)$$

**Condição Final:**

$$\mathbf{x}(T) = \mathbf{x}^T \quad (3.1d)$$

**Limites nas variáveis de controle:**

$$\mathbf{u}^{min} \leq \mathbf{u}(t) \leq \mathbf{u}^{max} \quad (3.1e)$$

**Restrição de trajetórias das variáveis de estado:**

$$\mathbf{x}^{min} \leq \mathbf{x}(t) \leq \mathbf{x}^{max} \quad (3.1f)$$

A solução do problema de controle ótimo pode ser de dois tipos: uma delas é chamada de *lei de controle ótimo* e a outra de *função de controle ótimo*. Na lei de controle ótimo, encontra-se uma relação entre os controles e os estados do tipo  $\mathbf{u}^* = \mathbf{f}(\mathbf{x}, t)$ . Ou seja, existe

uma “regra” que relaciona, a cada momento, os estados de um sistema com os controles que minimizam alguma função objetivo. Já na função de controle ótimo ( $\mathbf{u}^* = \mathbf{f}(t)$ ) procura-se encontrar uma função que não dependa diretamente dos estados, mas que efetivamente leve o sistema de um estado ao outro, respeitando os vínculos impostos e ao mesmo tempo minimizando uma função objetivo. O primeiro tipo é também chamado de *controle de realimentação*, ou *malha fechada* enquanto que o segundo é chamado de problema de *geração de trajetória* ou de *malha aberta*.

Um controle de realimentação ( $\mathbf{u}^*(\mathbf{x}, t)$ ) em geral é mais desejado pois permite que os estados sejam medidos continuamente e os controles corrigidos simultaneamente de forma a levar o sistema a uma trajetória ótima. No entanto, obter uma lei de controle ótimo é, em geral, uma tarefa difícil, especialmente quando aumenta-se o tamanho e a complexidade do sistema. Nesta dissertação tratamos, ao invés disso, de problemas de geração de trajetória ( $\mathbf{u}^*(t)$ ). Estes problemas podem ser resolvidos pelos chamados Métodos Indiretos ou Diretos. Os métodos indiretos procuram uma solução que satisfaça o Princípio do Máximo de Pontryagin, que quando aplicado, dá origem a um sistema de equações diferenciais com condições de contorno em dois pontos e por isso são chamados de “TPBVP - “*Two Point Boundary Value Problem*”. Nos Métodos Diretos, procura-se minimizar diretamente uma função objetivo. Há vários métodos diretos disponíveis para a solução de problemas de controle ótimo, tais como o Método do Gradiente, métodos baseados em Programação Dinâmica e ainda os que fazem uma *Parametrização do Controle*. Neste último grupo, destaca-se uma classe de algoritmos chamada de *Integração Iterativa*, que é utilizada por *softwares* tais como o *MISER3*<sup>24</sup> (JENNINGS et al. (1991)), *TOMP*<sup>25</sup> (KRAFT (1994)) e *RIOTS*<sup>26</sup> (SCHWARTZ (1996)). Uma ampla revisão dos métodos disponíveis para a solução de problemas de controle ótimo é apresentada nas teses de doutorado de SCHWARTZ (1996) e de MENEGALDO (2001).

O *software* RIOTS foi escolhido para realizar os cálculos apresentados nesta dissertação devido a (i) característica e complexidade do problema proposto estar dentro do conjunto de problemas que o *software* é capaz de resolver, (ii) haver uma licença disponível e (iii) pela experiência adquirida em outros trabalhos (MENEGALDO et al. (2003), MENEGALDO et al. (2006), TERCEIRO e FLEURY (2004)) por pesquisadores ligados a instituições brasileiras.

O RIOTS, cujo nome é uma abreviação para “*Recursive Integration Optimal Trajectory*

---

<sup>24</sup><http://www.cado.uwa.edu.au/miser/>

<sup>25</sup><http://www.netlib.org/toms/733>

<sup>26</sup><http://www.schwartz-home.com/adam/RIOTS/>

*Solver*” é um *toolbox* do Matlab, que possui um conjunto de *softwares* e utilitários, escritos em sua maior parte na linguagem C, capaz de resolver uma ampla gama de problemas de controle ótimo. Ele se baseia na Teoria das Aproximações Consistentes (POLAK (1993)) e foi desenvolvido durante o doutorado de SCHWARTZ (1996). Nesta abordagem, a solução obtida com o uso do RIOTS é a solução de um problema de controle ótimo discreto que é uma aproximação do problema de controle ótimo original no tempo contínuo. Ou seja, ele é construído a partir da discretização da dinâmica do sistema, levada a cabo através de um dos integradores numéricos disponíveis e pela parametrização do espaço de controle de dimensão infinita por um espaço de coeficientes da *spline* de dimensão finita. Deve-se ressaltar que o recurso apresentado pelo método de se descrever os controles por meio de *B-splines* permite que as variáveis de decisão deixem de ser os controles  $\mathbf{u}_k$  (forças, torques, etc) em cada um dos pontos da malha temporal e passem a ser os coeficientes  $\alpha_k$  da própria *B-spline*. Além da transformação do OCP no tempo contínuo em um OCP no tempo discreto, ocorre ainda uma segunda transformação do OCP no tempo discreto em um problema de Programação Matemática. Este problema de Programação Matemática é então resolvido com o pacote NPSOL (GILL et al. (1998)), que por sua vez é uma implementação do método SQP (Sequential Quadratic Programming).

Uma outra característica importante do algoritmo do RIOTS é a de utilizar métodos de Integração Iterativa. Sucintamente, isto significa que qualquer que seja a solução final apresentada, esta solução é fisicamente factível, pois é resultado da integração da equação de movimento do sistema sujeito ao vetor de controle ótimo. Para dar início à solução do problema, o RIOTS requer uma estimativa inicial da curva de controle. Tipicamente, essa curva pode ser um vetor nulo ( $u(t) = 0$ ) ou um outro valor constante qualquer. Com isso, a equação de movimento é integrada, o valor da função objetivo é calculado e a trajetória das variáveis de controle é determinada para todo o intervalo de tempo da simulação. Enquanto o critério de parada não é atingido, o RIOTS encontra uma nova curva de controle ( $u(t)$ ) que deverá diminuir o valor da função objetivo e usa esse controle para integrar mais uma vez as equações de movimento. Este processo irá se repetir até que as condições de parada do algoritmo sejam atingidas.

### 3.2 FORMULAÇÃO ESPECÍFICA DO OCP PARA O 4-BARRAS COM ATUADORES REDUNDANTES

A formulação de um problema de controle ótimo genérico foi apresentada na seção anterior. Nesta Seção deseja-se particularizar aquela formulação de forma a encontrar

os torques que fazem com que o modelo dinâmico do pêndulo triplo, apresentado na Seção 3.3, se comporte tal como um mecanismo 4-barras equivalente. A cinemática do 4-barras, durante uma volta completa do mecanismo, é conhecida *a priori* e foi calculada no Capítulo 2.

A formulação final do OCP, tal como apresentada a seguir, surgiu a partir de testes preliminares realizados durante a solução do problema que, posteriormente, foi chamado de CASO 1. Na função objetivo, buscou-se apenas a minimização dos torques dos atuadores ao longo de todo o ciclo. Como os torques podem assumir valores negativos, eles tiveram de ser elevados ao quadrado para que o módulo fosse minimizado. Assim, a função objetivo (Eq. 3.2) foi definida, onde  $w_1$ ,  $w_2$  e  $w_3$  assumem os valores 0 ou 1 de acordo com cada caso estudado, tal como mostra a Tabela 3.5. O valor do tempo final foi o mesmo em todos os casos ( $tf = 1s$ ), pois com  $\phi_1$  girando a 60 rpm, o 4-barras completa um ciclo em 1 segundo.

$$\min: f(\mathbf{u}) = \int_0^{tf} (w_1 U_1^2 + w_2 U_2^2 + w_3 U_3^2) dt \quad (3.2)$$

sujeito a:

*Equação de Movimento:*

$$\{\dot{\mathbf{x}}\} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \quad (3.3)$$

*Restrição de trajetória das variáveis de estado:*

$$l(\mathbf{x}, \mathbf{u}) = (f_1)^2 + (f_2)^2 + (f_3)^2 + (f_4)^2 + (f_5)^2 \leq EPSNEQ \quad (3.4)$$

onde:

$$f_1(\mathbf{x}, \mathbf{u}) = L_1 \cos(\phi_1) + L_2 \cos(\phi_2) + L_3 \cos(\phi_3) - L_4 \longrightarrow 0 \quad (3.5a)$$

$$f_2(\mathbf{x}, \mathbf{u}) = L_1 \sin(\phi_1) + L_2 \sin(\phi_2) + L_3 \sin(\phi_3) \longrightarrow 0 \quad (3.5b)$$

$$f_3(\mathbf{x}, \mathbf{u}) = -L_1 \dot{\phi}_1 \cos(\phi_1) - L_2 \dot{\phi}_2 \cos(\phi_2) - L_3 \dot{\phi}_3 \cos(\phi_3) \longrightarrow 0 \quad (3.5c)$$

$$f_4(\mathbf{x}, \mathbf{u}) = -L_1 \dot{\phi}_1 \sin(\phi_1) - L_2 \dot{\phi}_2 \sin(\phi_2) - L_3 \dot{\phi}_3 \sin(\phi_3) \longrightarrow 0 \quad (3.5d)$$

$$f_5(\mathbf{x}, \mathbf{u}) = \dot{\phi}_1 - 2\pi \longrightarrow 0 \quad (3.5e)$$

A Equação 3.3 representa o modelo dinâmico do sistema considerado colocado na forma de equações de estado. Este modelo representa um pêndulo triplo com atuadores de torques em suas articulações e forças de molas e amortecedores presentes na extremidade da barra mais distante do ponto fixado ao solo, cuja dedução aparece na Seção 3.3.

Uma vez que as equações de movimento desse sistema foram deduzidas, estas foram colocadas na forma de equações de estado através da rotina EOM2SS (SILVA et al. (2007)), desenvolvida pelo autor, cujo detalhamento aparece no Apêndice 1 desta dissertação.

A forma da equação de restrição de trajetória das variáveis de estado ( $l(\mathbf{x}, \mathbf{u})$ ), foi definida após uma série de testes preliminares onde diferentes opções foram experimentadas. Inicialmente considerou-se apenas os termos  $f_1$  e  $f_2$  (Eqs. 3.5a e 3.5b) na restrição 3.4, e ainda uma condição de vínculo terminal (não apresentada) em que o vetor de estado em  $t = 0$ s deve ser igual ao vetor de estado em  $t = 1$ s. As expressões  $f_1$  e  $f_2$  representam as equações do *loop* de posições do mecanismo 4 barras que devem ser satisfeitas em todas as posições que ele pode ocupar. Os vínculos terminais teriam o objetivo de forçar uma solução que representasse um movimento do 4 barras girando em regime permanente.

Verificou-se que quando cada uma das expressões  $f_1$  e  $f_2$  eram informadas ao RIOTS com duas expressões de restrição de trajetória, por exemplo,  $l_1(\mathbf{x}, \mathbf{u}) = f_1$  e  $l_2(\mathbf{x}, \mathbf{u}) = f_2$ , a solução do OCP era um pouco mais demorada do que quando as duas eram somadas na mesma inequação. Deste modo, optou-se por trabalhar com apenas 1 restrição de desigualdade para as variáveis de estado, composta pela soma de várias parcelas de expressões que, quando satisfeitas, devem ter valor nulo. No entanto, apesar de algumas simulações terem terminado normalmente, atendendo às restrições de posição, o perfil de velocidade das variáveis de estado fugia muito do que era esperado. Assim, resolveu-se acrescentar as expressões  $f_3$  e  $f_4$  (Eqs. 3.5c e 3.5d) à restrição  $l(\mathbf{x}, \mathbf{u})$ , que nada mais são do que a derivada temporal de  $f_1$  e  $f_2$ . Isto ainda não foi suficiente pois os resultados obtidos mostraram que a velocidade de  $\phi_1$  não era constante, e em alguns casos o mecanismo chegava a girar até a metade do caminho e depois voltava para a posição de saída. A solução para este problema foi acrescentar a expressão  $f_5$  à restrição  $l(\mathbf{x}, \mathbf{u})$  e remover o vínculo terminal.

A idéia de se introduzir no OCP restrições que representam a derivada do *loop* de posições foi inspirada na modelagem de sistemas multicorpos através de coordenadas cartesianas e Equações Diferenciais Algébricas. Apesar de se utilizar coordenadas generalizadas para a dedução do modelo de pêndulo triplo analisado neste capítulo, as expressões  $f_1 \dots f_5$  conceitualmente representam o vetor de restrições  $\Phi(\mathbf{q}, t)$  da Eq. 2.9 que foi introduzido no Capítulo 2 no contexto das coordenadas cartesianas.

As condições iniciais utilizadas em todos os casos que foram simulados foram:  $\phi_1 = 1.0472$  rad ( $60.0^\circ$ ),  $\phi_2 = 0.2907$  rad ( $16.5^\circ$ ),  $\phi_3 = 4.5513$  rad ( $260.7^\circ$ ),  $\dot{\phi}_1 = 6.2832$  rad/s ( $360^\circ/s$ ),  $\dot{\phi}_2 = -1.3760$  rad/s ( $78.8^\circ/s$ ) and  $\dot{\phi}_3 = 3.4240$  rad/s ( $196.1^\circ/s$ ). A condição inicial da posição das coordenadas generalizadas ( $\phi_1$ ,  $\phi_2$  e  $\phi_3$ ) foi obtida a partir de um



desenho feito em CAD e é a mesma utilizada no Capítulo anterior. Já as condições iniciais de velocidade utilizadas ( $\dot{\phi}_1$ ,  $\dot{\phi}_2$  e  $\dot{\phi}_3$ ), foram as velocidades obtidas para  $t = 1$  no final na análise cinemática do Capítulo anterior. Desta forma, procura-se formular o problema para que reproduza uma condição de operação do mecanismo em regime permanente.

Não foi necessário impor condições finais para as variáveis de estado (Eq.3.1d). Em geral, a imposição das condições finais dificultava a convergência numérica ou fazia com que o controle ótimo acabasse violando a restrição de trajetória das variáveis de estado para atender a condição de terminalidade. Quando a equação de restrição das variáveis de estado passou a incorporar as informações de velocidade (expressões  $f_3$ ,  $f_4$ ,  $f_5$ ) elas passaram a ser desnecessárias e não foram mais utilizadas. A Equação 3.1e para impor limites inferior ou superior às variáveis de controle também não foi utilizada. Como não estamos considerando atuadores de torque (motores elétricos) reais que possuem um torque máximo fornecido pelo fabricante, não faria sentido impor esta condição.

### 3.3 DEDUÇÃO DO MODELO DINÂMICO

O mecanismo 4 barras é caracterizado por ser um mecanismo plano, de cadeia fechada e 1 grau de liberdade. É formado por 4 barras que se articulam por juntas de pino em suas extremidades, sendo que uma delas (a chamada “barra fixa”) representa o segmento que une as duas barras que estão fixadas no solo. A barra que está fixada ao solo e que, em geral, recebe a ação de um motor, é denominada “manivela” (*crank*). A outra barra, que se fixa ao solo, mas que não recebe a ação do motor, chama-se “seguidor” (*follower*) enquanto que a barra intermediária, responsável por unir as duas outras recebe o nome de “acoplador” (*coupler*).

Teoricamente, a cinemática do mecanismo 4 barras poderia ser reproduzida por um outro sistema mecânico – o pêndulo triplo – desde que este seja submetido a forças externas de tal forma a manter fixa a extremidade da barra mais distante. O pêndulo triplo, por definição, se caracteriza por possuir 3 graus de liberdade e cadeia cinemática aberta. No entanto, se as forças externas forem forças passivas (e não controles), tais como a de molas e amortecedores, essas forças podem ser escolhidas apropriadamente de forma a “enrijecer” o sistema. Este recurso, ainda que não diminua o número de graus de liberdade do sistema, fará com que as três coordenadas sejam independentes entre si apenas para deslocamentos “pequenos”. Portanto, pode-se fazer com que estes deslocamentos se tornem pequenos o bastante, através do aumento das constantes de rigidez e amortecimento, de tal forma que a cinemática desse sistema de 3 GDL venha a representar a cinemática de um sistema de

1 GDL dentro de uma certa margem de tolerância.

Uma das formas mais sistemáticas e diretas de se deduzir equações de movimento é através do uso das Equações de Lagrange (Eq.3.6). No caso do pêndulo triplo plano, uma vez que se determine a origem do sistema fixo de coordenadas e se identifique um número de coordenadas absolutas e independentes entre si, é conceitualmente simples a dedução de tais equações. Basta que se escreva o vetor posição dos centros de massa de cada corpo rígido, em função das coordenadas independentes relativas a um referencial inercial. Desta forma é possível deduzir as equações de movimento apenas com derivações (totais e parciais) e operações aritméticas simples.

A dedução da equação de movimento será dividida em duas partes: primeiro será deduzido o lado esquerdo da Equação de Lagrange e em seguida o lado direito. O lado esquerdo da Equação de Lagrange (Eq.3.6) está associado ao comportamento da energia cinética e potencial dos corpos rígidos que constituem o sistema, representados na Figura 3.1. Já o lado direito da Equação de Lagrange diz respeito às forças externas (Figura 3.2) que atuam sobre ele. A Figura 3.2 representa o mesmo sistema mostrado na Figura 3.1, porém numa configuração diferente, mais adequada para explicitar a relação que o pêndulo triplo tem com o mecanismo 4-barras.

$$\frac{d}{dt} \frac{\partial L}{\partial \{\dot{q}\}} - \frac{\partial L}{\partial \{q\}} = \{Q\} \quad (3.6)$$

### 3.3.1 EQUAÇÃO DE MOVIMENTO (LADO ESQUERDO)

O esquema do modelo de pêndulo triplo é mostrado na Figura 3.1, onde a manivela, o acoplador e o seguidor possuem os comprimentos  $L_1 = 0,5$  m,  $L_2 = 0,9$  m e  $L_3 = 0,7$  m, respectivamente. O comprimento  $L_4 = 1,0$  m (não representado) é relativo à barra fixa. As massas de cada barra foram estimadas por um *software* de CAD para um material de densidade  $7,75g/cm^3$  e levaram aos valores de  $m_1 = 6,590$  kg,  $m_2 = 11,550$  kg e  $m_3 = 9,070$  kg. Já os momentos de inércia em torno do centro de massa foram calculados com a fórmula  $I = \frac{1}{12}mL^2$ , pois todos os centros de massa são considerados na metade de cada uma das barras. Desta forma, seus vetores-posição são dados por:

$$\mathbf{r}_1 = \begin{bmatrix} \frac{1}{2}L_1 \cos(\phi_1) \\ \frac{1}{2}L_1 \sin(\phi_1) \end{bmatrix} = \begin{bmatrix} R1_x \\ R1_y \end{bmatrix} \quad (3.7)$$

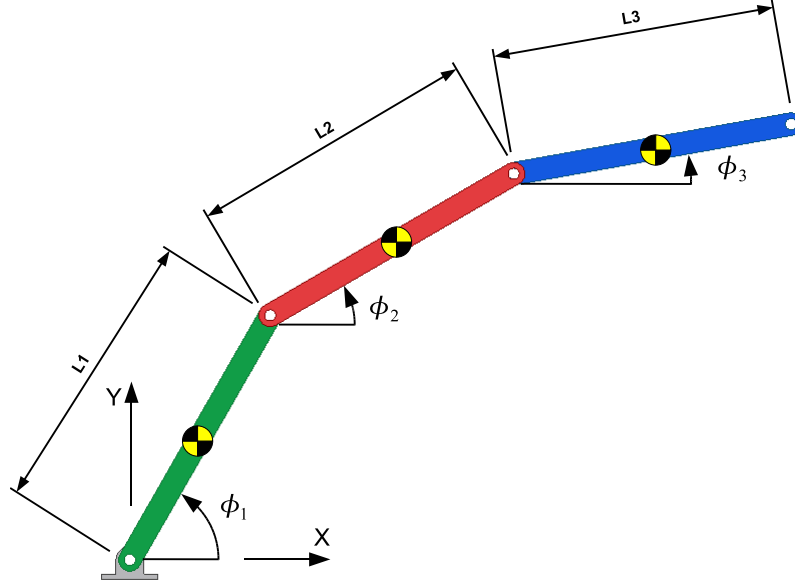


FIG. 3.1: Modelo dinâmico do pêndulo triplo

$$\mathbf{r}_2 = \begin{bmatrix} L_1 \cos(\phi_1) + \frac{1}{2} L_2 \cos(\phi_2) \\ L_1 \sin(\phi_1) + \frac{1}{2} L_2 \sin(\phi_2) \end{bmatrix} = \begin{bmatrix} R2_x \\ R2_y \end{bmatrix} \quad (3.8)$$

$$\mathbf{r}_3 = \begin{bmatrix} L_1 \cos(\phi_1) + L_2 \cos(\phi_2) + \frac{1}{2} L_3 \cos(\phi_3) \\ L_1 \sin(\phi_1) + L_2 \sin(\phi_2) + \frac{1}{2} L_3 \sin(\phi_3) \end{bmatrix} = \begin{bmatrix} R3_x \\ R3_y \end{bmatrix} \quad (3.9)$$

Enquanto que suas velocidades são dadas por:

$$\frac{d\mathbf{r}_1}{dt} = \begin{bmatrix} -\frac{1}{2} \dot{\phi}_1 L_1 \sin(\phi_1) \\ \frac{1}{2} \dot{\phi}_1 L_1 \cos(\phi_1) \end{bmatrix} = \begin{bmatrix} V1_x \\ V1_y \end{bmatrix} \quad (3.10)$$

$$\frac{d\mathbf{r}_2}{dt} = \begin{bmatrix} -\dot{\phi}_1 L_1 \sin(\phi_1) - \frac{1}{2} \dot{\phi}_2 L_2 \sin(\phi_2) \\ \dot{\phi}_1 L_1 \cos(\phi_1) + \frac{1}{2} \dot{\phi}_2 L_2 \cos(\phi_2) \end{bmatrix} = \begin{bmatrix} V2_x \\ V2_y \end{bmatrix} \quad (3.11)$$

$$\frac{d\mathbf{r}_3}{dt} = \begin{bmatrix} -\dot{\phi}_1 L_1 \sin(\phi_1) - \dot{\phi}_2 L_2 \sin(\phi_2) - \frac{1}{2} \dot{\phi}_3 L_3 \sin(\phi_3) \\ \dot{\phi}_1 L_1 \cos(\phi_1) + \dot{\phi}_2 L_2 \cos(\phi_2) + \frac{1}{2} \dot{\phi}_3 L_3 \cos(\phi_3) \end{bmatrix} = \begin{bmatrix} V3_x \\ V3_y \end{bmatrix} \quad (3.12)$$

Seja a energia cinética e a energia potencial de cada barra dada pelas Equações 3.13 e 3.14,

$$T_i = \frac{1}{2} m_i (V_{i_x}^2 + V_{i_y}^2) + \frac{1}{2} J_i \dot{\phi}_i^2 \quad (3.13)$$

$$V_i = m_i g R i_y \quad (3.14)$$

então a energia cinética total e a energia potencial total do sistema são dadas pelas Equações 3.15 e 3.16.

$$T = \sum_{i=1}^3 T_i \quad (3.15)$$

$$V = \sum_{i=1}^3 V_i \quad (3.16)$$

Uma vez obtidas as expressões de  $T$  e  $V$ , obtêm-se o chamado Lagrangeano subtraindo-se uma expressão da outra.

$$L = T - V \quad (3.17)$$

As equações de movimento são então obtidas fazendo-se duas derivações parciais do Lagrangeano – uma em relação à posição e a outra em relação à velocidade de cada uma das coordenadas. Sobre o resultado da derivada parcial em relação à velocidade da coordenada, deve-se fazer uma outra derivada (esta total, em relação ao tempo). Desta forma, pode-se obter o lado esquerdo das equações mostradas abaixo.

$$\begin{aligned} \text{eq1: } \frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}_1} - \frac{\partial L}{\partial \phi_1} &= Q_1 \\ \text{eq2: } \frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}_2} - \frac{\partial L}{\partial \phi_2} &= Q_2 \\ \text{eq3: } \frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}_3} - \frac{\partial L}{\partial \phi_3} &= Q_3 \end{aligned} \quad (3.18)$$

Rearrmando os termos da Equação 3.18, as equações de movimento do pêndulo triplo podem ser expressas a partir da Equação 3.19<sup>27</sup>.

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \end{bmatrix} + \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} \dot{\phi}_1^2 \\ \dot{\phi}_2^2 \\ \dot{\phi}_3^2 \end{bmatrix} + \begin{bmatrix} C \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \quad (3.19)$$

<sup>27</sup>Obs: Os elementos das matrizes  $[A]$ ,  $[B]$  e  $[C]$  são mostrados na Seção 3.3.3.

### 3.3.2 EQUAÇÃO DE MOVIMENTO (LADO DIREITO)

Para que a Equação 3.19 represente o comportamento de um pêndulo triplo ideal, (com todas as juntas sem atrito) e sem atuadores, basta que o lado direito da equação seja um vetor nulo. Nos modelos tratados nesta dissertação, ainda que se mantenha a hipótese de atrito nulo nas juntas, consideramos diversas opções de forças externas.

O primeiro conjunto de forças externas se refere aos torques nas articulações, também chamados de “torques de controle”, ou apenas “controle”. Fisicamente esses torques são fornecidos por motores presos às articulações do pêndulo triplo. O cálculo dessas curvas de controles é objeto das simulações levadas a cabo com o RIOTS. Como o pêndulo triplo possui três articulações, é possível considerar modelos atuados com um, dois ou três motores.

O segundo conjunto de forças consideradas aqui trata de forças devido à presença de molas e amortecedores que “amarram” a extremidade da última barra do pêndulo triplo (ponto D), de tal forma que este passe a se comportar como um mecanismo 4 barras. Essas forças não dependem diretamente dos controles, e sim dos estados. Portanto, nesta seção será feita a dedução do lado direito da equação de movimento (Eq.3.19) para o caso mais geral, no qual o vetor de forças generalizadas ( $\{Q\}$ ) inclui os termos das forças de molas e amortecedores (colocados nas posições horizontal e vertical), bem como os três atuadores de torque em cada uma das articulações.

#### 3.3.2.1 TERMO “FORÇA GENERALIZADA”

Os livros de dinâmica básica e avançada consultados durante a produção desta dissertação não trazem uma definição textual do termo “Força Generalizada”. Em geral definem Força Generalizada a partir da definição de “Deslocamento Virtual”, “Trabalho Virtual” e “Coordenada Generalizada”.

HAUG (1992) (p. 95) define o termo Deslocamento Virtual como “*um deslocamento pequeno (infinitesimal) hipotético em que o tempo é mantido constante*”. DOUGHTY (1988) (p.218) exemplifica esta idéia explicando a diferença entre o operador de diferenciação ( $d$ ) e o operador delta ( $\delta$ ) (também chamado de “variação” pois usualmente aparece em textos sobre Cálculo Variacional). Considere-se, por exemplo, uma função escalar  $f(x, y, z, t)$  que depende das coordenadas espaciais  $x, y, z$  e do tempo  $t$ .

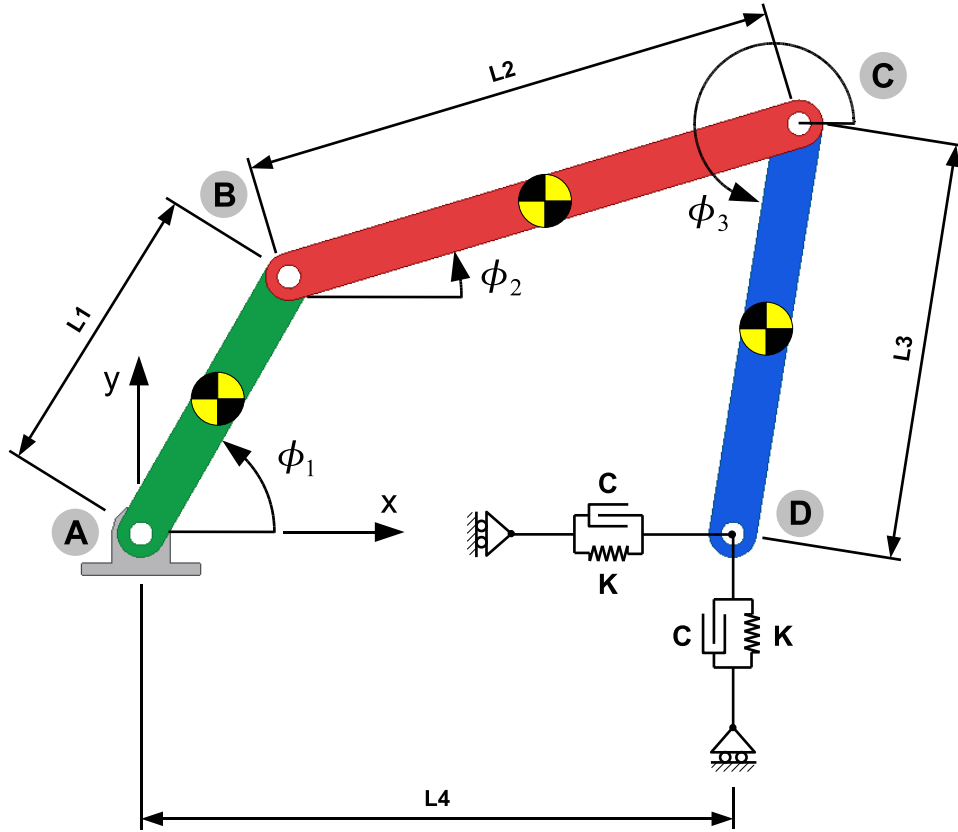


FIG. 3.2: Esquema do pêndulo triplo com restrições: forças de mola, amortecedor e torques de controle

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial t} dt \quad (3.20a)$$

$$\delta f = \frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial y} \delta y + \frac{\partial f}{\partial z} \delta z \quad (3.20b)$$

A diferença entre  $df$  e  $\delta f$  é que neste último não há a presença do termo relativo ao tempo. Ou seja, quando HAUG (1992) diz “*com o tempo mantido constante*” ele quer dizer que a variação do tempo  $\delta t \rightarrow 0$  é nula. Ele acrescenta ainda que o deslocamento virtual é *arbitrário* na medida em que este se mantenha compatível com as restrições cinemáticas impostas e ele. O Trabalho Virtual é portanto o trabalho resultante do produto escalar entre uma força finita  $\mathbf{F}$  aplicada a um deslocamento virtual  $\delta \mathbf{r}$ .

$$\delta W = \mathbf{F} \cdot \delta \mathbf{r} \quad (3.21)$$

Uma discussão detalhada sobre o termo Coordenada Generalizada já foi feita no Capítulo 2. Aqui, nos referimos a este termo como um conjunto de coordenadas independentes

que descrevem de forma única a configuração de um sistema. Segundo GOLDSTEIN (1970), uma vez que se estabeleça o que ele chama de Equações de Transformação (Transformation Equations), que relaciona uma coordenada qualquer de interesse  $\mathbf{r}_i$  com as coordenadas generalizadas  $q_j$ ,  $j = 1, 2 \dots N$ , o deslocamento virtual desta coordenada é dado por:

$$\delta \mathbf{r}_i = \sum_{j=1}^N \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j \quad (3.22)$$

Portanto, o Trabalho Virtual executado por todas as forças que agem sobre um sistema é o somatório do produto escalar de cada uma dessas forças pelo deslocamento virtual imposto por elas e pode ser expresso em função das forças e coordenadas generalizadas do sistema através da Equação 3.23d.

$$\delta W = \sum_{i=1}^M \mathbf{F}_i \cdot \delta \mathbf{r}_i \quad (3.23a)$$

$$= \sum_{i=1}^M \sum_{j=1}^N \mathbf{F}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j \quad (3.23b)$$

$$= \sum_{j=1}^N \sum_{i=1}^M \mathbf{F}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j \quad (3.23c)$$

$$= \sum_{j=1}^N Q_j \cdot \delta q_j \quad (3.23d)$$

Assim, trocando a ordem dos somatórios da Equação 3.23b, o j-ésimo componente do que se chama Vetor de Forças Generalizadas, que atua sobre um sistema multi-corpos com  $j = 1 \dots N$  graus de liberdade sobre o qual existem  $i = 1 \dots M$  forças externas atuando, é definido pela Equação 3.24.

$$Q_j = \sum_{i=1}^M \mathbf{F}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \quad (3.24)$$

Nesta Equação,  $\mathbf{F}_i$  é a i-ésima força que atua sobre o sistema,  $\mathbf{r}_i$  é a direção de atuação da i-ésima força e  $Q_j$  é o j-ésimo elemento do vetor de forças generalizadas. Para o cálculo da expressão 3.24, deve-se notar que as coordenadas auxiliares  $\mathbf{r}_i$  são funções das coordenadas generalizadas  $q_j$ , portanto  $\mathbf{r}_i = f_i(q_1, q_2, \dots, q_N)$ , tal como mostram as Equações 3.25a-3.25g. A Tabela 3.1 relaciona cada uma dessas coordenadas (que são

representadas nas Figuras 3.2 e 3.3) com as respectivas forças que atuam ao longo delas e as Equações 3.26a-3.26d, explicitam as expressões das forças horizontal e vertical devido à presença da mola e do amortecedor ( $F_x$ ,  $F_y$ ,  $C_x$  e  $C_y$ ).

$$r_1(\phi_1, \phi_2, \phi_3) = \phi_1 \quad (3.25a)$$

$$r_2(\phi_1, \phi_2, \phi_3) = \phi_2 - \phi_1 \quad (3.25b)$$

$$r_3(\phi_1, \phi_2, \phi_3) = \phi_3 - \phi_2 \quad (3.25c)$$

$$r_4(\phi_1, \phi_2, \phi_3) = L_1 \cos(\phi_1) + L_2 \cos(\phi_2) + L_3 \cos(\phi_3) \quad (3.25d)$$

$$r_5(\phi_1, \phi_2, \phi_3) = L_1 \sin(\phi_1) + L_2 \sin(\phi_2) + L_2 \sin(\phi_3) \quad (3.25e)$$

$$r_6(\phi_1, \phi_2, \phi_3) = r_4 \quad (3.25f)$$

$$r_7(\phi_1, \phi_2, \phi_3) = r_5 \quad (3.25g)$$

$$F_x = -K(L_1 \cos(\phi_1) + L_2 \cos(\phi_2) + L_3 \cos(\phi_3) - L_4) \quad (3.26a)$$

$$F_y = -K(L_1 \sin(\phi_1) + L_2 \sin(\phi_2) + L_2 \sin(\phi_3)) \quad (3.26b)$$

$$C_x = -C(-L_1 \dot{\phi}_1 \sin(\phi_1) - L_2 \dot{\phi}_2 \sin(\phi_2) - L_3 \dot{\phi}_3 \sin(\phi_3)) \quad (3.26c)$$

$$C_y = -C(L_1 \dot{\phi}_1 \cos(\phi_1) + L_2 \dot{\phi}_2 \cos(\phi_2) + L_3 \dot{\phi}_3 \sin(\phi_3)) \quad (3.26d)$$

TAB. 3.1: Identificação de forças externas e suas respectivas coordenadas da Figura 3.4

$r_i$ (Coord.)	$F_i$ (Força)	Expressão	Descrição
$r_1$	$F_1 = u_1$	-	Torque do atuador na articulação A
$r_2$	$F_2 = u_2$	-	Torque do atuador na articulação B
$r_3$	$F_3 = u_3$	-	Torque do atuador na articulação C
$r_4$	$F_4 = F_x$	$F_4 = -K(r_4 - L_4)$	Força horizontal da mola no ponto D
$r_5$	$F_5 = F_y$	$F_5 = -K(r_5)$	Força vertical da mola no ponto D
$r_6$	$F_6 = C_x$	$F_6 = -C(\dot{r}_4)$	Força horizontal do amortecedor no ponto D
$r_7$	$F_7 = C_y$	$F_7 = -C(\dot{r}_5)$	Força vertical do amortecedor no ponto D

Expandindo a Equação 3.24, teremos:



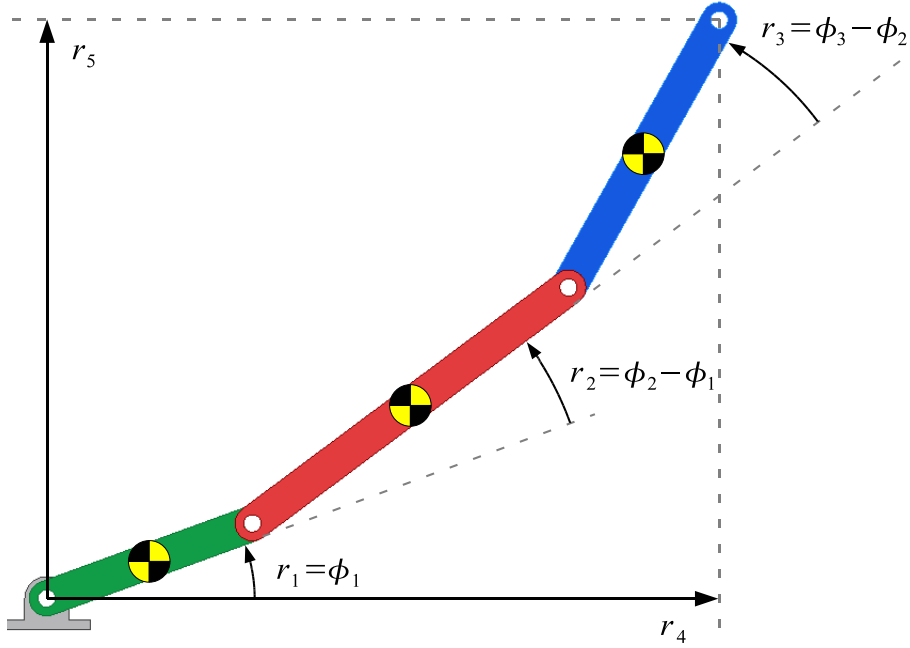


FIG. 3.3: Pêndulo triplo “aberto” indicando a relação entre as coordenadas independentes ( $\phi$ ) e auxiliares  $r$

$$\begin{aligned}
 Q_1 &= F_1 \frac{\partial r_1}{\partial \phi_1} + F_2 \frac{\partial r_2}{\partial \phi_1} + \dots + F_7 \frac{\partial r_7}{\partial \phi_1} \\
 Q_2 &= F_1 \frac{\partial r_1}{\partial \phi_2} + F_2 \frac{\partial r_2}{\partial \phi_2} + \dots + F_7 \frac{\partial r_7}{\partial \phi_2} \\
 Q_3 &= F_1 \frac{\partial r_1}{\partial \phi_3} + F_2 \frac{\partial r_2}{\partial \phi_3} + \dots + F_7 \frac{\partial r_7}{\partial \phi_3}
 \end{aligned} \tag{3.27}$$

A Equação 3.27 nos revela que há uma estrutura regular nas expressões que formam o vetor de forças generalizadas quando as elas atuam nas mesmas direções das coordenadas. As forças e coordenadas no modelo em questão são identificadas na Tabela 3.1. Ou seja, as forças  $F_j$  podem ser agrupadas num vetor de forças  $\mathbf{F}$  e as derivadas parciais, num Jacobiano  $\mathbf{J}$ . Uma demonstração semelhante é feita por MOON (1998), mostrando que o vetor de força generalizada é o produto do Jacobiano pelo vetor de forças externas; ainda que este autor insira a sua demonstração no contexto do Método das Potências Virtuais.

$$\begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_j \end{bmatrix} = \begin{bmatrix} \frac{\partial r_1}{\partial \phi_1} & \frac{\partial r_2}{\partial \phi_1} & \dots & \frac{\partial r_j}{\partial \phi_1} \\ \frac{\partial r_1}{\partial \phi_2} & \frac{\partial r_2}{\partial \phi_2} & \dots & \frac{\partial r_j}{\partial \phi_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_1}{\partial \phi_i} & \frac{\partial r_2}{\partial \phi_i} & \dots & \frac{\partial r_j}{\partial \phi_i} \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_j \end{bmatrix} \Rightarrow [\mathbf{Q}] = [\mathbf{J}] [\mathbf{F}] \tag{3.28}$$

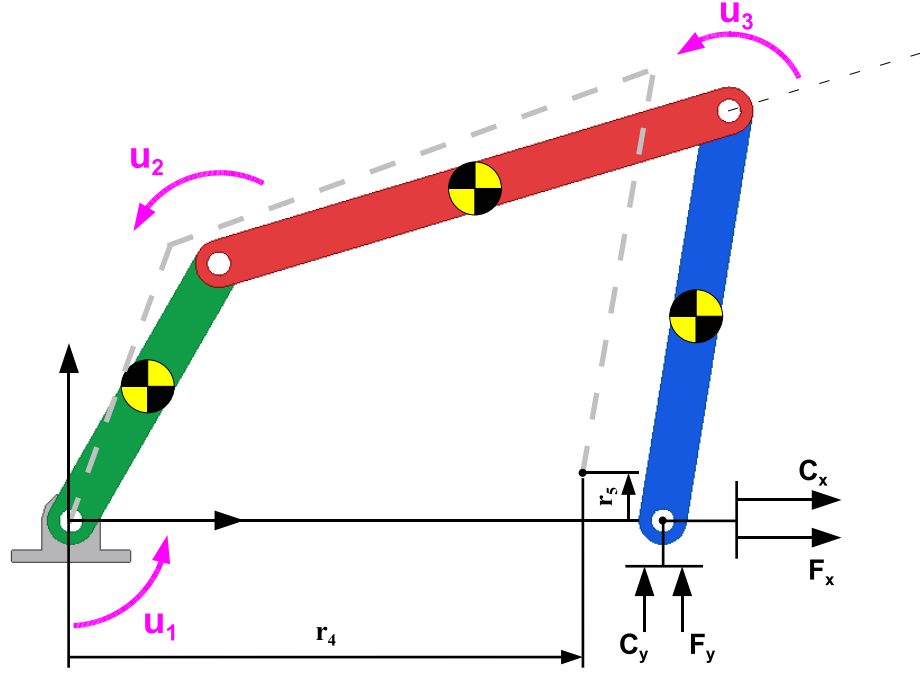


FIG. 3.4: Indicação de forças externas atuando no modelo do pêndulo triplo com restrições

Substituindo as forças  $\mathbf{F}_j$  e as coordenadas  $\mathbf{r}_j$  da Tabela 3.1 na Equação 3.28 e expandindo-a, obtemos as Equações 3.29a - 3.29c.

$$Q_1 = u_1 \frac{\partial \phi_1}{\partial \phi_1} + u_2 \frac{\partial(\phi_2 - \phi_1)}{\partial \phi_1} + u_3 \frac{\partial(\phi_3 - \phi_2)}{\partial \phi_1} + F_x \frac{\partial(r_4)}{\partial \phi_1} + F_y \frac{\partial(r_5)}{\partial \phi_1} + C_x \frac{\partial(r_4)}{\partial \phi_1} + C_y \frac{\partial(r_5)}{\partial \phi_1} \quad (3.29a)$$

$$Q_2 = u_1 \frac{\partial \phi_1}{\partial \phi_2} + u_2 \frac{\partial(\phi_2 - \phi_1)}{\partial \phi_2} + u_3 \frac{\partial(\phi_3 - \phi_2)}{\partial \phi_2} + F_x \frac{\partial(r_4)}{\partial \phi_2} + F_y \frac{\partial(r_5)}{\partial \phi_2} + C_x \frac{\partial(r_4)}{\partial \phi_2} + C_y \frac{\partial(r_5)}{\partial \phi_2} \quad (3.29b)$$

$$Q_3 = u_1 \frac{\partial \phi_1}{\partial \phi_3} + u_2 \frac{\partial(\phi_2 - \phi_1)}{\partial \phi_3} + u_3 \frac{\partial(\phi_3 - \phi_2)}{\partial \phi_3} + F_x \frac{\partial(r_4)}{\partial \phi_3} + F_y \frac{\partial(r_5)}{\partial \phi_3} + C_x \frac{\partial(r_4)}{\partial \phi_3} + C_y \frac{\partial(r_5)}{\partial \phi_3} \quad (3.29c)$$

Calculando as derivadas parciais indicadas nas Equações 3.29a - 3.29c, obtemos então a expressão final das forças generalizadas para o modelo dinâmico do pêndulo triplo com três atuadores de torques nas juntas e ainda molas e amortecedores na extremidade da barra mais distal.

$$Q_1 = u_1 - u_2 - L_1(C(L_1\dot{\phi}_1 + L_2\dot{\phi}_2 \cos(\phi_1 - \phi_2) + L_3\dot{\phi}_3 \cos(\phi_1 - \phi_3)) + K(L_4 \sin(\phi_1) - L_2 \sin(\phi_1 - \phi_2) + L_3 \sin(\phi_1 - \phi_3))) \quad (3.30a)$$

$$Q_2 = u_2 - u_3 - L_2(C(L_2\dot{\phi}_2 + L_1\dot{\phi}_1 \cos(\phi_1 - \phi_2) + L_3\dot{\phi}_3 \cos(\phi_2 - \phi_3)) + K(L_1 \sin(\phi_1 - \phi_2) + L_4 \sin(\phi_2) - L_3 \sin(\phi_2 - \phi_3))) \quad (3.30b)$$

$$Q_3 = u_3 - L_3(C(L_3\dot{\phi}_3 + L_1\dot{\phi}_1 \cos(\phi_1 - \phi_3) + L_2\dot{\phi}_2 \cos(\phi_2 - \phi_3)) + K(L_1 \sin(\phi_1 - \phi_3) + L_2 \sin(\phi_2 - \phi_3) + L_4 \sin(\phi_3))) \quad (3.30c)$$

### 3.3.3 EQUAÇÃO DE MOVIMENTO NA FORMA MATRICIAL

Feita a dedução dos lados direito e esquerdo da equação de Lagrange é possível agora reescrever a Equação 3.19 na forma matricial (Eq. 3.31). Repare que o vetor  $[Q]$  foi desmembrado, separando os termos presentes nas Equações 3.30a - 3.30c que dependem dos controles (e agrupando seus coeficientes na matriz  $[D]$ ) daqueles que dependem dos estados (e agrupando-os no vetor  $[E]$ ). Cada um dos elementos das matrizes e vetores da Equação 3.31 são mostrados a seguir, exceto para a matriz  $[D]$  que é diferente para cada um dos casos analisados e por isso é apresentada na Tabela 3.5.

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \end{bmatrix} + \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} \dot{\phi}_1^2 \\ \dot{\phi}_2^2 \\ \dot{\phi}_3^2 \end{bmatrix} + \begin{bmatrix} C \end{bmatrix} = \begin{bmatrix} D \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} E \end{bmatrix} \quad (3.31)$$

$$\begin{aligned}
A_{1,1} &= m_2 L_1^2 + m_3 L_1^2 + 1/4 m_1 L_1^2 + I_1 \\
A_{1,2} &= L_2 L_1 m_3 \sin(\phi_2) \sin(\phi_1) + \frac{1}{2} L_2 L_1 m_2 \cos(\phi_2) \cos(\phi_1) + \\
&\quad \frac{1}{2} L_2 L_1 m_2 \sin(\phi_2) \sin(\phi_1) + L_2 L_1 m_3 \cos(\phi_2) \cos(\phi_1) \\
A_{1,3} &= \frac{1}{2} m_3 L_3 L_1 \cos(\phi_3) \cos(\phi_1) + \frac{1}{2} m_3 L_3 L_1 \sin(\phi_3) \sin(\phi_1) \\
A_{2,1} &= L_2 L_1 m_3 \sin(\phi_2) \sin(\phi_1) + \frac{1}{2} L_2 L_1 m_2 \cos(\phi_2) \cos(\phi_1) + \\
&\quad \frac{1}{2} L_2 L_1 m_2 \sin(\phi_2) \sin(\phi_1) + L_2 L_1 m_3 \cos(\phi_2) \cos(\phi_1) \\
A_{2,2} &= I_2 + m_3 L_2^2 + 1/4 m_2 L_2^2 \\
A_{2,3} &= \frac{1}{2} m_3 L_3 L_2 \sin(\phi_3) \sin(\phi_2) + \frac{1}{2} m_3 L_3 L_2 \cos(\phi_3) \cos(\phi_2) \\
A_{3,1} &= \frac{1}{2} m_3 L_3 L_1 \cos(\phi_3) \cos(\phi_1) + \frac{1}{2} m_3 L_3 L_1 \sin(\phi_3) \sin(\phi_1) \\
A_{3,2} &= \frac{1}{2} m_3 L_3 L_2 \sin(\phi_3) \sin(\phi_2) + \frac{1}{2} m_3 L_3 L_2 \cos(\phi_3) \cos(\phi_2) \\
A_{3,3} &= \frac{1}{4} m_3 L_3^2 + I_3
\end{aligned}$$

$$\begin{aligned}
B_{1,1} &= 0 \\
B_{1,2} &= \frac{1}{2} L_2 L_1 m_2 \cos(\phi_2) \sin(\phi_1) - L_2 L_1 m_3 \sin(\phi_2) \cos(\phi_1) + \\
&\quad L_2 L_1 m_3 \cos(\phi_2) \sin(\phi_1) - \frac{1}{2} L_2 L_1 m_2 \sin(\phi_2) \cos(\phi_1) \\
B_{1,3} &= \frac{1}{2} m_3 L_3 L_1 \sin(\phi_1) \cos(\phi_3) - \frac{1}{2} m_3 L_3 L_1 \sin(\phi_3) \cos(\phi_1) \\
B_{2,1} &= \frac{1}{2} L_2 L_1 m_2 \sin(\phi_2) \cos(\phi_1) - \frac{1}{2} L_2 L_1 m_2 \cos(\phi_2) \sin(\phi_1) + \\
&\quad L_2 L_1 m_3 \sin(\phi_2) \cos(\phi_1) - L_2 L_1 m_3 \cos(\phi_2) \sin(\phi_1) \\
B_{2,2} &= 0 \\
B_{2,3} &= \frac{1}{2} m_3 L_3 L_2 \sin(\phi_2) \cos(\phi_3) - \frac{1}{2} m_3 L_3 L_2 \sin(\phi_3) \cos(\phi_2) \\
B_{3,1} &= -\frac{1}{2} m_3 L_3 L_1 \sin(\phi_1) \cos(\phi_3) + \frac{1}{2} m_3 L_3 L_1 \sin(\phi_3) \cos(\phi_1) \\
B_{3,2} &= -\frac{1}{2} m_3 L_3 L_2 \sin(\phi_2) \cos(\phi_3) + \frac{1}{2} m_3 L_3 L_2 \sin(\phi_3) \cos(\phi_2) \\
B_{3,3} &= 0
\end{aligned}$$

$$\begin{aligned}
C_{1,1} &= m_2 g L_1 \cos(\phi_1) + m_3 g L_1 \cos(\phi_1) + \frac{1}{2} m_1 g L_1 \cos(\phi_1) \\
C_{1,2} &= \frac{1}{2} m_2 g L_2 \cos(\phi_2) + m_3 g L_2 \cos(\phi_2) \\
C_{1,3} &= \frac{1}{2} m_3 g L_3 \cos(\phi_3)
\end{aligned}$$

$$\begin{aligned}
E_{1,1} &= -L_1(C(L_1\dot{\phi}_1 + L_2\dot{\phi}_2 \cos(\phi_1 - \phi_2) + L_3\dot{\phi}_3 \cos(\phi_1 - \phi_3)) \\
&\quad + K(L_4 \sin(\phi_1) - L_2 \sin(\phi_1 - \phi_2) + L_3 \sin(\phi_1 - \phi_3))) \\
E_{2,1} &= -L_2(C(L_2\dot{\phi}_2 + L_1\dot{\phi}_1 \cos(\phi_1 - \phi_2) + L_3\dot{\phi}_3 \cos(\phi_2 - \phi_3)) \\
&\quad + K(L_1 \sin(\phi_1 - \phi_2) + L_4 \sin(\phi_2) - L_3 \sin(\phi_2 - \phi_3))) \\
E_{3,1} &= -L_3(C(L_3\dot{\phi}_3 + L_1\dot{\phi}_1 \cos(\phi_1 - \phi_3) + L_2\dot{\phi}_2 \cos(\phi_2 - \phi_3)) \\
&\quad + K(L_1 \sin(\phi_1 - \phi_3) + L_2 \sin(\phi_2 - \phi_3) + L_4 \sin(\phi_3)))
\end{aligned}$$

### 3.4 DISCUSSÃO E HISTÓRICO DOS DIFERENTES CASOS SIMULADOS

Foram estudados sete diferentes casos do problema de controle ótimo. Eles diferem basicamente por utilizar diferentes variações do sistema mecânico apresentado na Equação 3.31. Estas variações consideram diferentes números de atuadores de torque, bem como a presença (ou não) de molas e amortecedores. A Tabela 3.2 mostra a nomenclatura adotada para cada um dos casos estudados em função do número de atuadores e da presença ou não de vínculos de mola ou amortecedor.

TAB. 3.2: Mapa de casos estudados sobre o modelo 4 barras

N° Atuadores	Sem Vínculo	Com Vínculo	
		Mola	Mola + Amortecedor
<b>1</b>	CASO 3	CASO 4	CASO 5
<b>2</b>	-	-	CASO 6
<b>3</b>	CASO 1	CASO 2	CASO 7

As seções a seguir descrevem em detalhes cada um dos casos estudados e apresentam as conclusões que foram inferidas no decorrer da formulação, implementação e solução de cada um dos problemas. A numeração adotada seguiu a ordem cronológica, ou seja, o CASO 2, por exemplo, só foi formulado uma vez que se obteve uma resposta satisfatória

para o CASO 1. Desta forma, muito do que se aprendeu das tentativas realizadas no caso anterior, foi incorporado ou aplicado nos casos seguintes<sup>28</sup>.

### 3.4.1 CASO 1: 3 ATUADORES

A primeira formulação a ser considerada foi a do pêndulo triplo com três atuadores de torque nas juntas e nenhum outro tipo de força presente. Este modelo pode ser obtido a partir da Equação 3.31 eliminando-se a matriz  $[E]$ . À primeira vista, este problema nos pareceu conceitualmente mais simples pois o número de atuadores é igual ao número de graus de liberdade do sistema. Fisicamente ele se assemelha ao problema de um braço robótico plano, composto por três segmentos, cuja extremidade (*end effector*) deve permanecer parada enquanto que o segmento próximo à base deve dar uma volta completa com velocidade angular constante.

A Figura 3.5 mostra a trajetória das variáveis de controle, isto é, os torques obtidos, ao longo de uma volta completa. Mostra também, a trajetória das variáveis de estado (posições angulares à esquerda, velocidades angulares à direita) indicadas por um asterisco (\*) em comparação com a cinemática dessas mesmas variáveis calculada na solução da dinâmica inversa do 4 barras no capítulo anterior (Figura 2.10), indicadas por linha cheia.

#### **Análise:**

Tentou-se introduzir uma condição análoga à da restrição diretora utilizada no Capítulo 2 – a de que a barra 1 deve girar com velocidade constante – através de duas formas diferentes. A primeira, incorporando a expressão  $(\dot{\phi}_1 - 2\pi)^2$  na função objetivo a ser minimizada; a segunda incorporando essa expressão na restrição de trajetória das variáveis de estado. A segunda alternativa deu resultados melhores que a primeira.

Foram feitas algumas tentativas introduzindo-se uma condição final (Eq. 3.1d). No entanto essas tentativas falharam pois, como a condição final era igual à condição inicial, o mecanismo começava a girar para um lado até um certo ponto e depois invertia o sentido de rotação para voltar ao estado de onde partiu.

---

<sup>28</sup>No Capítulo 2 foi adotada a nomenclatura *CASO 1*, *CASO 2*, etc. para identificar os diferentes casos de acordo com o número de atuadores utilizados. Neste Capítulo, adota-se a mesma nomenclatura, porém com significado diferente, tal como explicado. Quando for necessário se referir aos CASOS de mesmo nome, será feita menção ao Capítulo, ou ao método (DAE ou OCP) a que eles se referem.

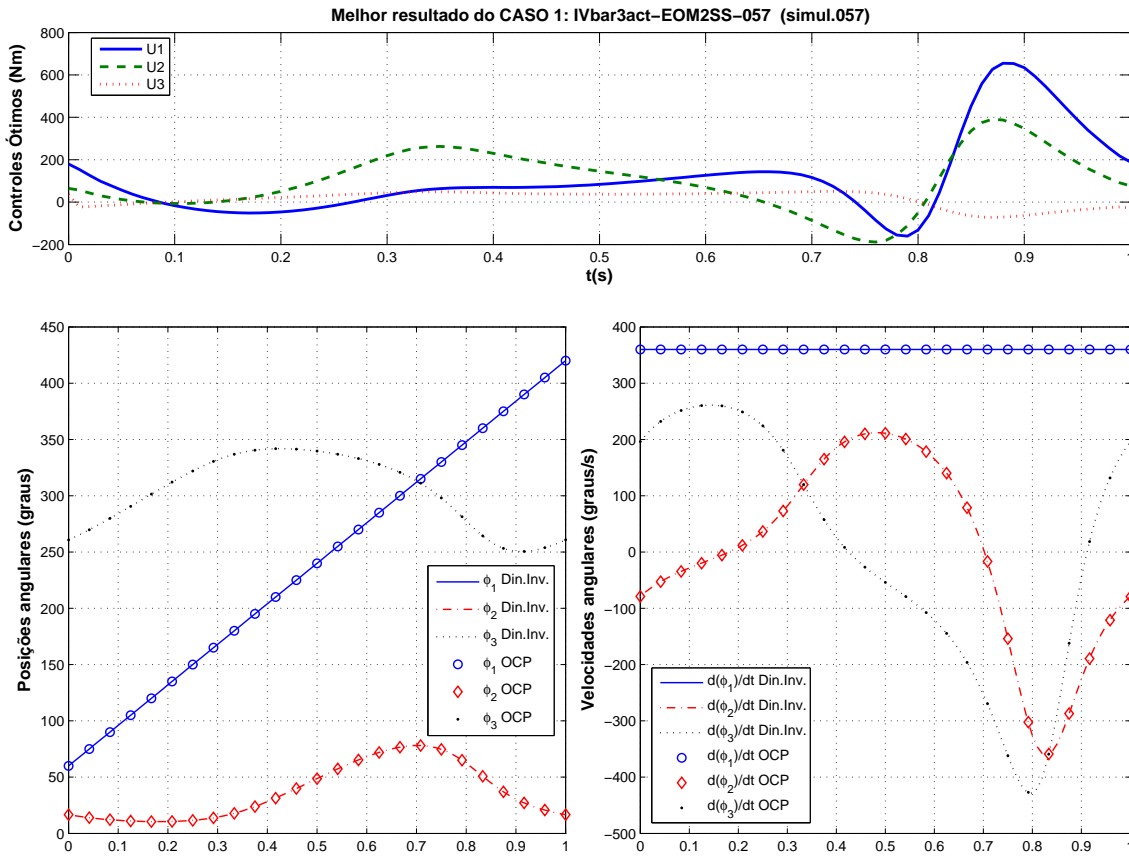


FIG. 3.5: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 1

Uma das limitações do RIOTS é a de que ele não é capaz de resolver problemas de controle ótimo com restrições de trajetória de igualdade de estado. Ou seja, encontrar uma curva de controle (ótimo) tal que faça o sistema obedecer uma equação na qual as variáveis de estado estejam envolvidas. No entanto, o RIOTS aceita restrições de desigualdade de estado. A primeira restrição que vem à mente quando pensamos em em mecanismo 4 barras, é o chamado “*loop* de posições”. O *loop* de posições é essencialmente uma equação vetorial que aparece em mecanismos de cadeia cinemática fechada e que deve ser obedecida para todas as posições. Como a equação do *loop* de posição envolve as coordenadas generalizadas (que são parte das variáveis de estado), esta pode ser transformada em uma restrição de desigualdade de estado. Isto é feito elevando-se o lado esquerdo da equação ao quadrado e impondo-se que o lado direito seja menor do que um número pequeno. Este número pequeno é chamado aqui de EPSNEQ.

Exemplo:

$$L_1 \cos(x_1) + L_2 \cos(x_2) + L_3 \cos(x_3) - L_4 = 0$$

↓

$$(L_1 \cos(x_1) + L_2 \cos(x_2) + L_3 \cos(x_3) - L_4)^2 \leq EPSNEQ$$

O RIOTS em geral, fica mais rápido quando há apenas uma única restrição de trajetória de desigualdade de estado. Portanto, no caso do 4 barras em que há pelo menos duas equações de restrição (a equação do *loop* de posições na direção x e na direção y), a melhor alternativa foi somar as duas equações, transformando-as em apenas uma única equação. Desta forma o número NLTIC<sup>29</sup> informado ao RIOTS foi igual a 1.

Nas primeiras tentativas de solução do problema do CASO 1, a restrição de desigualdade de estado contava apenas com as parcelas relativas ao *loop* de posição do mecanismo (Eq. 3.5a e Eq. 3.5b). Depois, introduziu-se a condição da restrição diretora (Eq. 3.5e). Mas mesmo assim a simulação não convergia, ou dava um resultado inesperado. O problema só começou a dar mostras de que seria possível chegar a uma solução satisfatória quando foram acrescentadas as equações da derivada do *loop* de posições (Eq. 3.5c e Eq. 3.5d). Com isso, chegou-se à formulação final do problema de controle ótimo, apresentada na Seção 3.2, que foi adotada em todos os casos.

Mesmo utilizando a restrição de desigualdade de estado da Equação 3.4, o problema não convergia. Possivelmente devido a instabilidade do pêndulo triplo associada à pobreza da estimativa do controle inicial, que em geral era um vetor nulo. O caminho para se chegar a uma simulação com terminação normal, foi começar com uma aceleração gravitacional (por exemplo  $g = 4m/s^2$ ) e com um tempo final pequenos. Em seguida, procurou-se aumentar primeiro o tempo final, e a cada simulação com término normal, utilizou-se a resposta (o controle ótimo propriamente dito) como estimativa do controle inicial para a próxima simulação. Este processo deve ser repetido até que se consiga chegar ao valor do tempo final desejado (no caso,  $t_f = 1$  s). Depois, deve-se fixar o tempo final e, a cada simulação, ir aumentando o valor da aceleração gravitacional até chegar em  $g = 9.81 \text{ m/s}^2$ . Esta estratégia de variação da

---

<sup>29</sup>NLTIC = Non Linear Trajectory Inequality Constraint – nomenclatura adotada pelo RIOTS para informar ao *software* o número de restrições não-lineares de desigualdade de estado



aceleração gravitacional foi utilizada com sucesso por MENEGALDO et al. (2004) para estabilizar um pêndulo triplo invertido que possui uma dinâmica bem mais instável que o modelo considerado aqui.

Existem duas tolerâncias que afetam diretamente a convergência da solução dos problemas tratados no RIOTS. Uma delas é a tolerância geral de otimização, chamada de EPS, enquanto que a outra, é uma tolerância de violação das restrições, chamada de EPSNEQ. No processo iterativo descrito no item anterior, verificou-se conveniente trabalhar com EPSNEQ bem reduzido, isto é, em torno de  $10^{-5}$ , e com EPS bem aberto (em torno de  $10^{-1}$ ). Isso facilita que a otimização tenha uma terminação normal, com uma boa reprodução da cinemática do mecanismo 4-barras que está imposta através da Equação 3.4.

Finalmente, adotando-se as estratégias colocadas nos itens anteriores, foi possível encontrar um conjunto de curvas de torque tais que façam com que o pêndulo triplo se mova como um mecanismo 4 barras (Figura 3.5). Se esse conjunto de curvas de torque fosse aplicado num mecanismo ideal de 4 barras com 3 atuadores, além de ter a propriedade de minimizar a integral dos torques ao longo do ciclo, também não induziria reações em uma das articulações que fica presa ao solo.

No processo de se ajustar os parâmetros da simulação, obteve-se uma série de tentativas com terminação normal. Verificando as curvas dos controles  $u_1(t)$  dessas simulações num mesmo gráfico, percebe-se que todas têm o mesmo aspecto, o mesmo acontecendo para  $u_2(t)$ . Somente as curvas relativas a  $u_3(t)$  exibiam um formato um pouco diferente entre si.

### 3.4.2 CASO 2: 3 ATUADORES COM MOLAS NAS EXTREMIDADES

Uma vez que foi obtido um resultado satisfatório com o CASO 1, um outro modelo foi proposto para simular o comportamento do mecanismo 4 barras. A idéia principal é formular as equações de um pêndulo triplo cuja extremidade livre está “amarrada” por meio de duas molas, uma na horizontal e outra na vertical. O comprimento de repouso dessas molas é ajustado de maneira tal que elas não exerçam força sobre o pêndulo triplo quando a extremidade da terceira barra (ponto D) estiver na posição da articulação correspondente da barra fixa, no mecanismo 4 barras. Este modelo é obtido a partir da Equação 3.31, atribuindo-se para K (constante de rigidez das molas) um valor bem alto (em torno de

$10^5$ ) e  $C = 0$  (coeficiente de amortecimento), para eliminar a força dos amortecedores. Os resultados obtidos são mostrados na Figura 3.6.

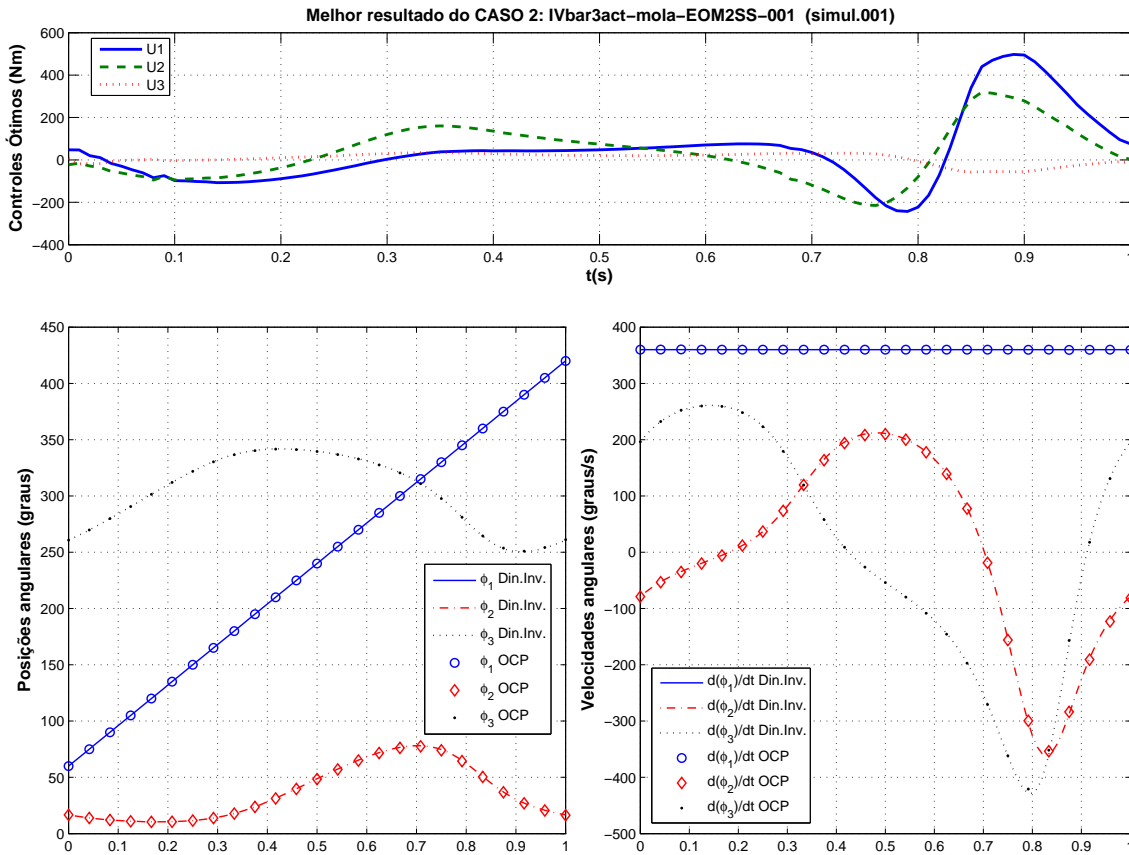


FIG. 3.6: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 2

### Análise:

A convergência deste modelo foi mais fácil do que no CASO 1. Por exemplo: usando  $g = 4,0 \text{ m/s}^2$  e  $t_f = 1,0 \text{ s}$ , o modelo sem molas (CASO 1) não convergiu. Foi necessário realizar uma simulação anterior, com um  $t_f$  menor e usar o controle ótimo dessa simulação como controle inicial na simulação com  $t_f = 1,0 \text{ s}$ . Da mesma forma, usando  $g = 9,81 \text{ m/s}^2$  e  $t_f = 1,0 \text{ s}$ , o modelo sem molas também não apresentava terminação normal. No CASO 2, ao contrário, logo na primeira tentativa usando-se  $g = 4,0 \text{ m/s}^2$  e  $t_f = 1,0 \text{ s}$  o RIOTS retornou uma terminação normal. Numa outra tentativa usando  $g = 9,81 \text{ m/s}^2$  e  $t_f = 1,0 \text{ s}$  e um vetor nulo como estimativa do controle inicial, conseguiu-se terminação normal. No entanto, na curva de controles, o torque nos primeiros centésimos de segundo ficou extremamente alto.

O melhor resultado, mostrado na Figura 3.6, foi obtido utilizando-se como estimativa

inicial do controle o resultado da primeira simulação em que se utilizou  $g = 4 \text{ m/s}^2$ . A introdução da força da mola possibilitou que com apenas duas simulações com terminação normal se conseguisse um resultado aceitável. Já no CASO 1, foram necessários mais de 10 simulações.

### 3.4.3 CASO 3: 1 ATUADOR

Nenhuma das simulações com o modelo de pêndulo triplo com 1 atuador, tentando reproduzir um mecanismo de 4 barras tradicional, foi bem sucedida. Isso já era esperado pois este sistema é incontrollável. Ou seja, o número de atuadores é menor do que o número de graus de liberdade do sistema. Pode-se observar que o que o RIOTS tenta fazer é estimar uma solução inicial de controle para o sistema e em seguida integra a equação de movimento. Uma vez que aquele controle inicial não é capaz de levar todas as variáveis de estado para o estado desejado (e ainda assim, respeitando as restrições e vínculos de estado), ele retorna uma mensagem do tipo *line search fail*.

### 3.4.4 CASO 4: 1 ATUADOR COM MOLAS NAS EXTREMIDADES

No CASO 4, formulou-se o problema de controle ótimo para o pêndulo triplo com apenas 1 atuador na barra 1 (*crank*) e molas atuando na extremidade da barra 3 (ponto D da Figura 3.2). Apesar do modelo continuar a ser essencialmente um sistema de 3 graus de liberdade, com a escolha de uma mola de alta rigidez é possível “amarrar” o pêndulo triplo de forma a fazer com que as 3 coordenadas generalizadas sejam independentes apenas para pequenos deslocamentos. Usando-se esse modelo, pretende-se formular um problema de controle ótimo que represente um problema típico de dinâmica inversa do mecanismo 4 barras. A solução deste problema é conhecida, pois pode ser obtida através de outros métodos mais consagrados, tal como o que foi mostrado no Capítulo 2. Desta forma, poderemos comparar a solução do controle ótimo com a solução da dinâmica inversa tradicional. Os resultados obtidos são mostrados na Figura 3.7.

#### **Análise:**

Foi difícil obter uma terminação normal no CASO 4. De início tentou-se reduzir a aceleração da gravidade para  $g = 0 \text{ m/s}^2$ . Depois reduziu-se o tempo final para um valor pequeno (correspondente a apenas 5% ou 10% da volta completa), mas nenhuma dessas estratégias funcionou.

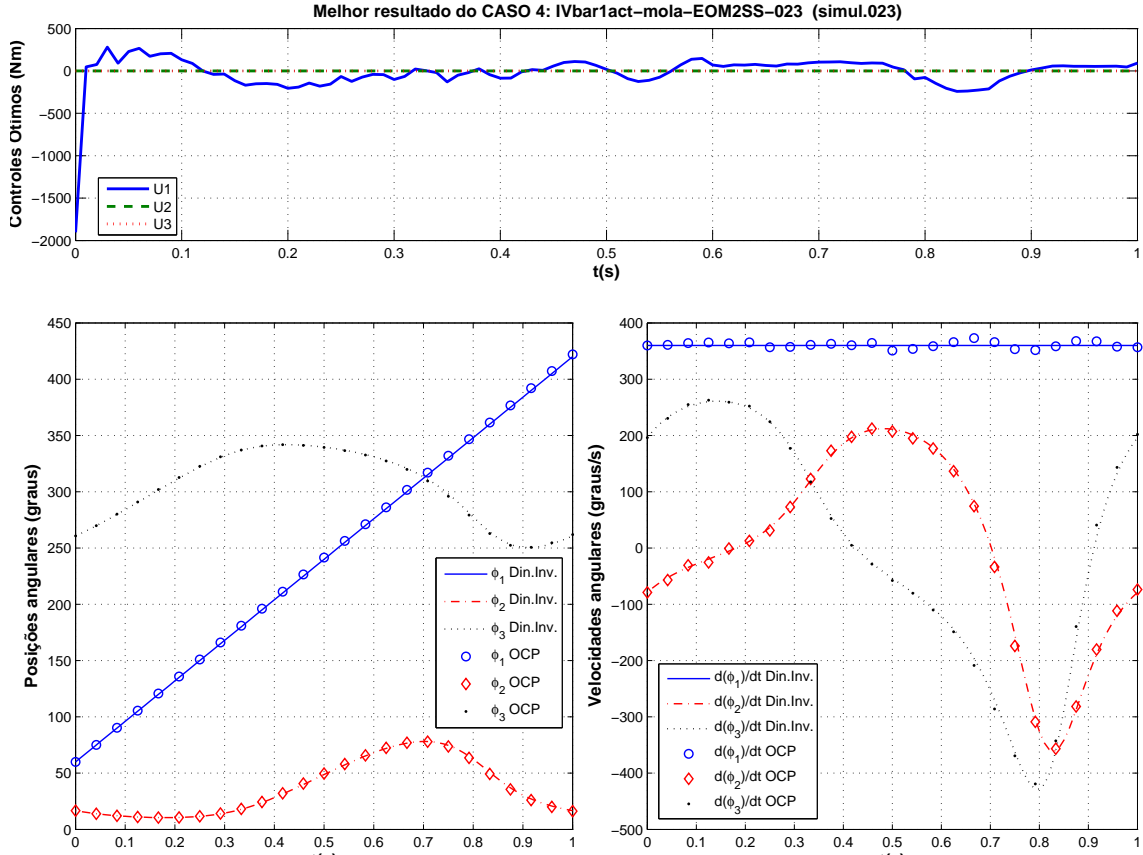


FIG. 3.7: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 4

Em todos os casos estudados até aqui, o integrador utilizado pelo RIOTS era o Runge-Kutta de 4ª ordem (RK4) e passo fixo. Nenhuma terminação normal foi obtida usando-se este integrador no CASO 4. A única terminação normal para este caso foi obtida utilizando-se o integrador LSODA (SCHWARTZ et al. (1997) e PETZOLD (1983)) que possui passo variável e é capaz de resolver problemas duros (“*stiff*”).

Nesta tentativa em que foi obtida terminação normal, foi utilizado:  $K = 10^7$  N/m,  $g = 0$  m/s<sup>2</sup>,  $tf = 1$  s,  $EPS = EPSNEQ = 0.1$ , controle inicial nulo e o integrador LSODA. Usando o resultado dessa simulação bem sucedida como estimativa do controle inicial para uma outra tentativa com  $g = 1$  m/s<sup>2</sup>, e todos os demais parâmetros mantidos iguais, o RIOTS retornou um *Line search fail*. As curvas da cinemática mostradas na Figura 3.7 ficaram razoavelmente próximas da cinemática exata, mas a força da mola oscilou muito e em alta frequência.

Como não foi possível conseguir terminações normais com valores crescentes da aceleração da gravidade, optou-se por abandonar o modelo do CASO 4. Portanto,

pode-se concluir que a imposição da mola dificultou a convergência numérica desse problema devido à componente oscilatória que introduz. Com uma mola de alta rigidez, aparece uma perturbação de alta frequência que o algoritmo numérico não consegue tratar. Já se uma mola de rigidez baixa é utilizada, o pêndulo triplo deixa de reproduzir a cinemática do mecanismo 4 barras.

### 3.4.5 CASO 5: 1 ATUADOR COM MOLA E AMORTECEDOR NAS EXTREMIDADES

Neste modelo acrescentam-se amortecedores em paralelo com as molas que haviam sido introduzidas no CASO 4. Esta configuração é mostrada na Figura 3.2. A necessidade de se acrescentar amortecedores veio da conclusão da Seção 3.4.4, de que a imposição da mola dificulta a convergência numérica devido à componente oscilatória que introduz. Os resultados obtidos são mostrados na Figura 3.8.

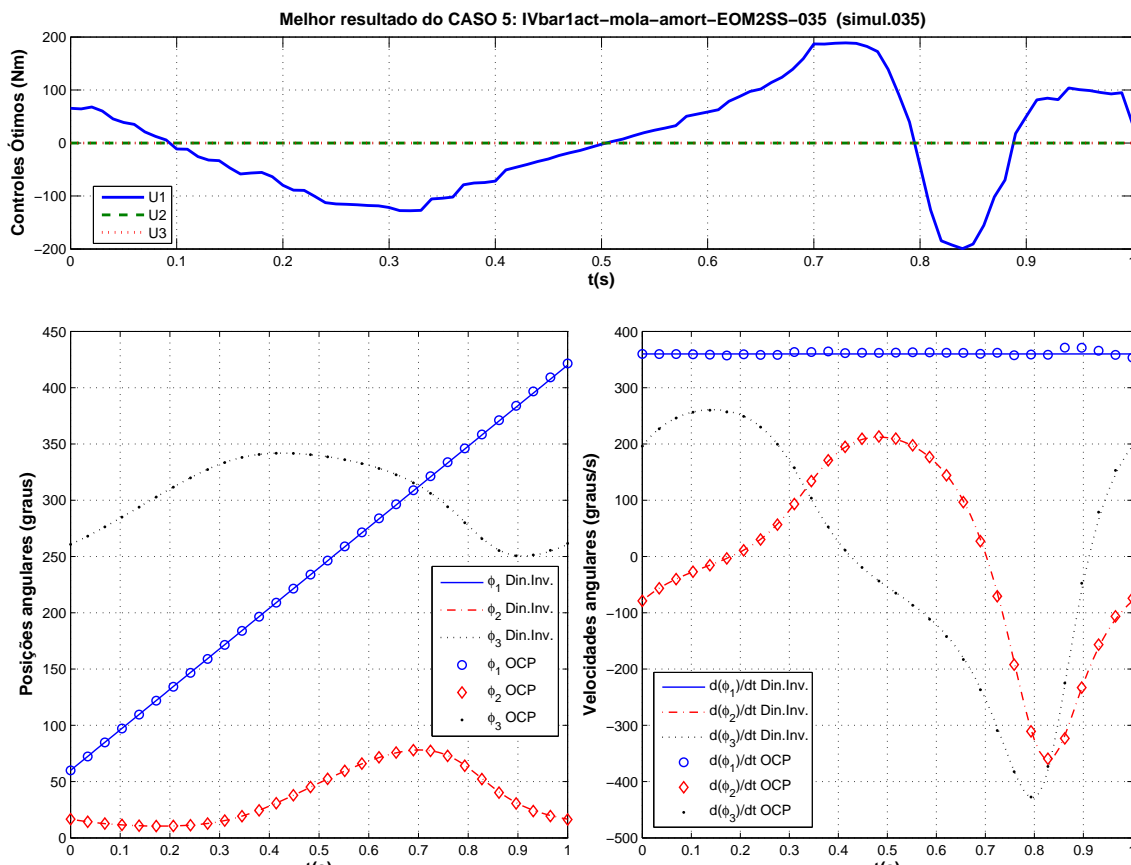


FIG. 3.8: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 5

#### Análise:

Introduzindo o amortecedor, a convergência numérica ficou um pouco mais fácil. Foi possível obter algumas terminações normais, sempre utilizando  $g = 0$  e tempos

finais pequenos. No entanto, utilizando esta estratégia, só foi possível avançar até  $tf = 0.3$  s.

O fato de só conseguir convergência com tempos finais pequenos era devido ao método de integração que estava sendo usado: o Runge-Kutta de 4ª ordem. Quando foi utilizado um integrador de passo variável (LSODA), foi possível resolver o OCP com o tempo final desejado ( $tf = 1$  s).

A partir daí, a estratégia utilizada foi fixar a rigidez e o coeficiente de amortecimento em valores altos ( $K = 5,0 \cdot 10^4$  e  $C = 1,0 \cdot 10^4$ ) e deixar as tolerâncias do RIOTS abertas ( $EPS = EPSNEQ = 0,1$ ). Desta forma, foi possível obter uma terminação normal com  $tf = 1,0$  s e  $g = 0$  m/s<sup>2</sup>, usando um vetor nulo como estimativa do controle inicial. O controle ótimo obtido nesta simulação serviu de estimativa de controle inicial para uma outra com  $g = 1$  m/s<sup>2</sup>; o que funcionou bem. O controle ótimo obtido com  $g = 1$  m/s<sup>2</sup> foi a estimativa de controle inicial para uma simulação com  $g = 2$  m/s<sup>2</sup> e assim sucessivamente, até atingir  $g = 9.81$  m/s<sup>2</sup>.

Curiosamente, o controle ótimo obtido para  $g = 0$  m/s<sup>2</sup> ficou com aspecto de “escadinha”. Os demais controles com  $g$  pequeno, também ficaram “pontudos” (descontinuidade na primeira derivada). À medida que se obtinham resultados bem sucedidos, com valores maiores da aceleração gravitacional, o controle ótimo foi sendo suavizado.

Depois que se chegou a uma resposta com terminação normal utilizando  $tf = 1,0$  s,  $g = 9.81$  m/s<sup>2</sup> e tolerâncias abertas ( $EPS = EPSNEQ = 0,1$ ), não foi possível obter outras simulações bem sucedidas utilizando valores menores de  $EPSNEQ$ . Nessas tentativas o RIOTS retornava o erro *No feasible point found for nonlinear constraints*, tal como aconteceu na simulação `IVbar1act-mola-amort-EOM2SS-034` (ver Tabela B.4).

As simulações bem sucedidas foram realizadas com constantes de rigidez e coeficientes de amortecimento de  $K = 5,0 \cdot 10^4$  N/m e  $C = 1,0 \cdot 10^4$  N.s/m, respectivamente. Em seguida, tentou-se aumentar esses coeficientes para ver até que ponto seria possível enrijecer o sistema sem prejudicar a convergência numérica. Foi possível conseguir simulações bem sucedidas com valores de até  $K = 5,0 \cdot 10^7$  N/m e  $C = 1,0 \cdot 10^7$  N.s/m. Quando as duas constantes foram aumentadas para  $K = 5,0 \cdot 10^8$  N/m e  $C = 1,0 \cdot 10^8$ , N.s/m o RIOTS apresentou o erro: *Kuhn-Tucker conditions satisfied*

*but sequence did not converge*, (simulação `IVbar1act-mola-amort-EOM2SS-036`, ver Tabela B.4).

Mesmo utilizando a máxima rigidez que o integrador LSODA era capaz de lidar ( $K = 5,0 \cdot 10^7$  N/m e  $C = 1,0 \cdot 10^7$  N.s/m), não foi possível conseguir uma terminação normal utilizando um vetor nulo como estimativa inicial do controle. Portanto, no CASO 5, a estratégia de se conseguir respostas com gravidade reduzida e ir utilizando os controles ótimos resultantes como estimativa inicial do controle para novas simulações com acelerações gravitacionais maiores, se mostrou imprescindível, (simulação `IVbar1act-mola-amort-EOM2SS-038`, ver Tabela B.4).

O resultado do controle  $u_1$  nas simulações `IVbar1act-mola-amort-EOM2SS-033` e `IVbar1act-mola-amort-EOM2SS-035`<sup>30</sup> deram resultados muito parecidos com a dinâmica inversa do 4 barras calculado pela metodologia utilizada por HAUG (1989), apresentada no Capítulo 2. A comparação desse resultado aparece na Figura 3.9. Pelo resultado da comparação, parece que o controle poderia ser melhorado se  $N$  fosse aumentado de 200 para 400. Mas isto exigiria que se fizesse cerca de 9 simulações adicionais começando com  $g = 0$  m/s<sup>2</sup> e indo até  $g = 9.81$  m/s<sup>2</sup>, usando o resultado do controle ótimo da simulação anterior como estimativa do controle inicial da próxima simulação.

### 3.4.6 CASO 6: 2 ATUADORES (ARTICULAÇÕES A E B) COM MOLA E AMORTECEDOR NAS EXTREMIDADES

Neste modelo considerou-se o mesmo modelo anterior, com molas e amortecedores nas extremidades, porém desta vez com dois atuadores. Um dos atuadores está fixado na manivela (articulação A) e sua reação vai para o solo, enquanto que o outro fica no acoplador (articulação B) e sua reação é recebida pela manivela. Os resultados obtidos são mostrados na Figura 3.10.

#### **Análise:**

A experiência acumulada nas simulações bem sucedidas dos casos anteriores foi aplicada à escolha dos parâmetros utilizados no CASO 6. Logo na primeira tentativa, foi obtida uma terminação normal para o tempo final e aceleração gravitacional desejados. Foram utilizados  $tf = 1.0$  s,  $g = 9.81$  m/s<sup>2</sup>,  $K = 5,0 \cdot 10^4$  N/m,  $C = 1,0 \cdot 10^4$

---

<sup>30</sup>As condições dessas simulações aparecem detalhadas na Tabela B.4

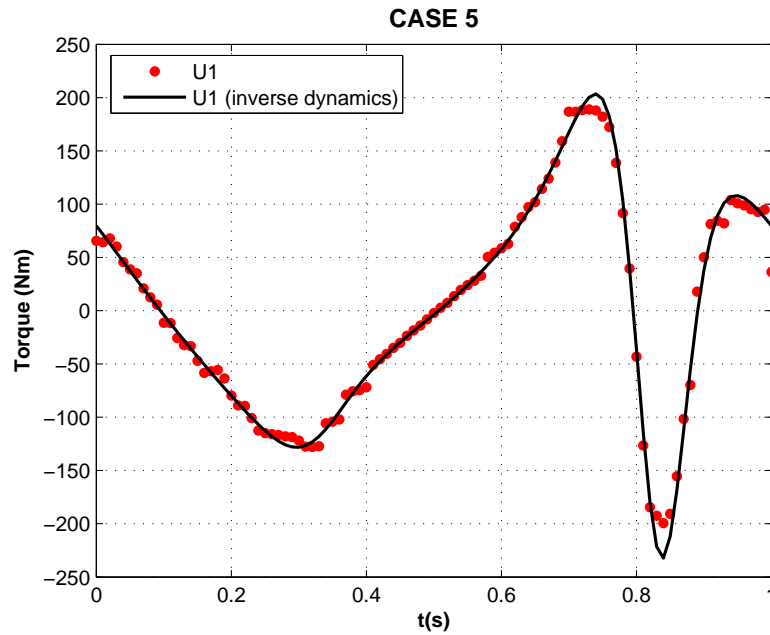


FIG. 3.9: Comparação do torque U1 calculado pelo controle ótimo no CASO 5 com o torque calculado através da dinâmica inversa

N.s/m,  $EPS = EPSNEQ = 0,1$ , o integrador de passo variável LSODA e um vetor nulo como estimativa inicial do controle. Em nenhum outro dos casos simulados anteriormente foi possível obter uma terminação normal com essa combinação de parâmetros. Com esse resultado pode-se concluir que a introdução do segundo atuador facilitou a convergência numérica, fazendo com que a estratégia de se iniciar as simulações com aceleração gravitacional reduzida se tornasse dispensável.

A discretização da malha temporal utilizada na primeira tentativa do CASO 6 foi de  $N = 200$  pontos. Este número de pontos vinha sendo mantido em todos os casos até este momento. No CASO 6, procurou-se explorar então a variação no número de pontos da malha temporal, utilizando  $N = 400, 100, 50$  e  $25$ . O melhor resultado, apresentado na Figura 3.10, foi obtido com  $N = 50$ .

### 3.4.7 CASO 7: 3 ATUADORES COM MOLLA E AMORTECEDOR NAS EXTREMIDADES

Como os casos onde foram obtidos os melhores resultados para 1 e 2 atuadores foram casos em que se contava com a presença de amortecedor, estudou-se também o CASO 7, onde ao modelo com 3 atuadores foram acrescentados molas e amortecedores no ponto D.

**Análise:**



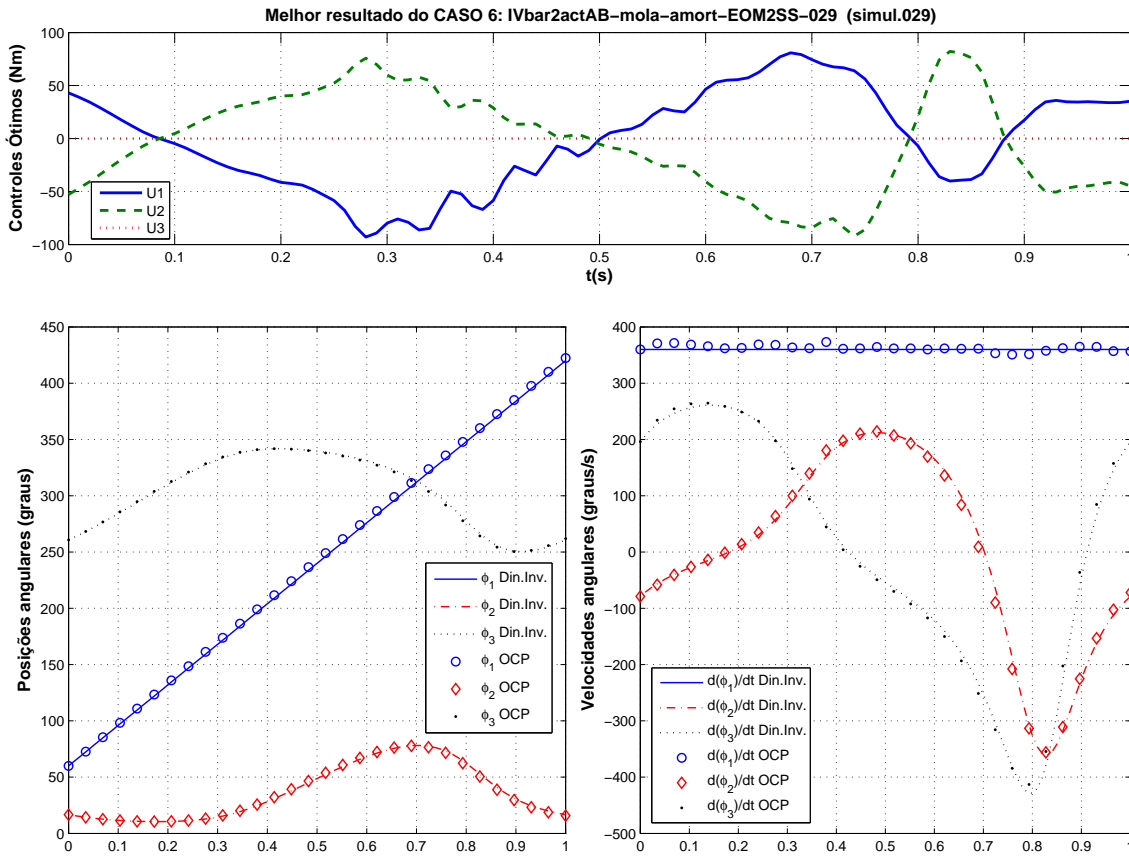


FIG. 3.10: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 6

A convergência do problema, sem dúvida foi mais fácil do que nos CASOS 1 e 2. Não foi necessário utilizar a estratégia de se obter soluções intermediárias utilizando  $tf$  ou  $g$  menores do que o real. Logo na primeira tentativa, o problema retornou uma terminação normal, mesmo utilizando o integrador Runge-Kutta de 4ª ordem. No entanto, o controle não foi tão suave quanto se desejava. A resposta foi melhorada aumentando-se a constante de amortecimento ( $C$ ), reduzindo-se a tolerância de otimização ( $EPS$ ) e principalmente trocando o integrador de passo fixo para o de passo variável (LSODA). Com isso conseguiu-se uma resposta razoável, (e foi publicada em SILVA e MENEGALDO (2007)) ainda que tenha sido menos suave do que a dos CASOS 1 e 2.

O resultado publicado na referência anterior foi obtido com uma discretização da malha temporal de  $N = 200$  pontos. As tentativas subsequentes para se melhorar o resultado, procurando-se suavizar as curvas de controle, se basearam em alterar o número de pontos dessa malha temporal. Na primeira tentativa, o número de pontos foi aumentado para 400 e verificou-se que o aspecto “ruidoso” do controle ficou ainda

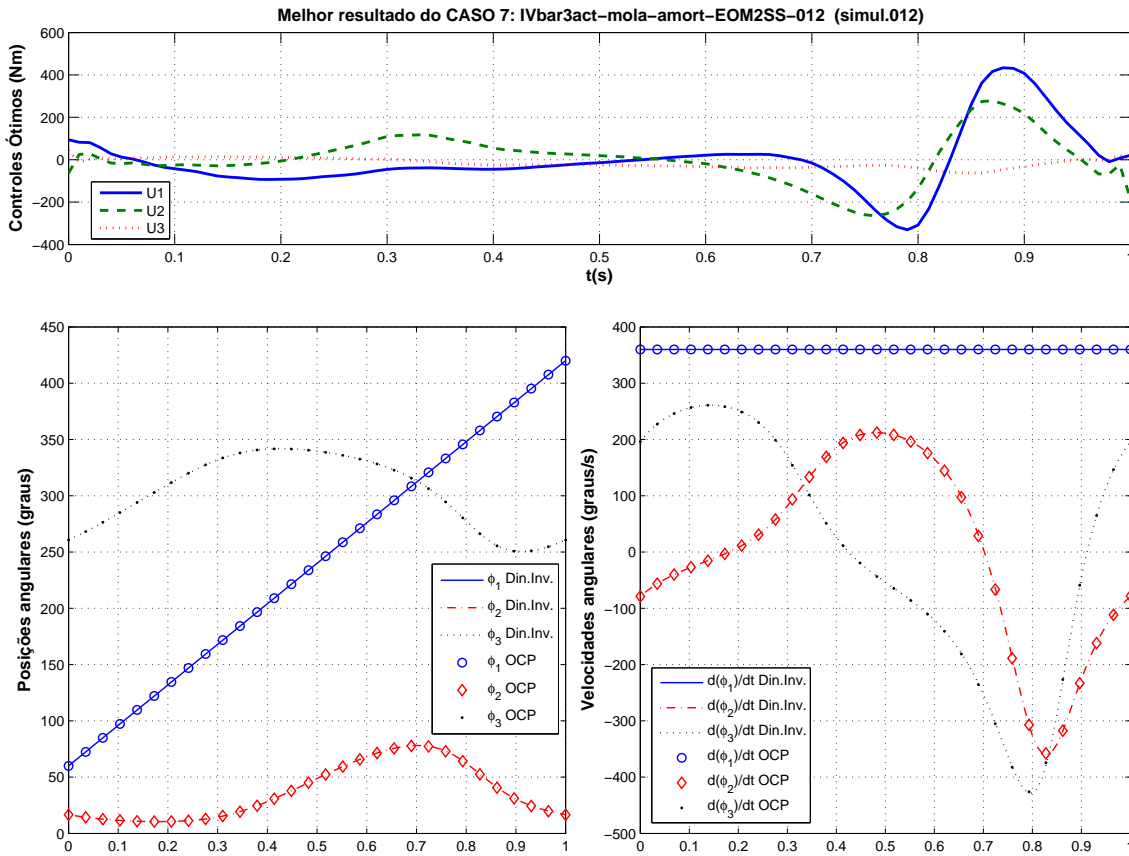


FIG. 3.11: Trajetória das variáveis de estado e de controle na melhor tentativa para o CASO 7

pior. Na segunda tentativa, o número de pontos foi diminuído para 50 pontos e a resposta se mostrou tão suave quanto nos CASOS 1 e 2, tal como é mostrado na Figura 3.11.

### 3.5 ANÁLISE DOS RESULTADOS

O processo de simulação e aprendizagem sobre o uso correto das rotinas de controle ótimo do RIOTS, percorreu o seguinte caminho: Primeiro partiu-se do pêndulo triplo, com três atuadores sem nenhum vínculo de mola e amortecedor (CASO 1). Uma vez conseguido sucesso nesta simulação, introduziu-se o vínculo de mola no sistema (CASO 2) e verificou-se que isto facilitou a convergência numérica. O próximo passo foi então verificar que o pêndulo triplo com apenas 1 atuador (CASO 3) não é capaz de reproduzir a cinemática do 4 barras. Tentando fazer com que o pêndulo triplo com apenas 1 atuador reproduzisse o comportamento do 4 barras, utilizou-se a estratégia que já havia sido bem sucedida para o caso com 3 atuadores, isto é, introduzir o vínculo da mola (CASO 4). No entanto, isso fez com que o problema se tornasse numericamente duro, o que dificultou a

integração numérica de forma que foi necessário mudar o integrador de passo fixo (Runge-Kutta de 4ª ordem) para um integrador de passo variável (LSODA). Além disso, a força oscilatória da mola, sem amortecimento, fez com que o controle não apresentasse uma curva bem condicionada. O próximo passo foi então introduzir um amortecedor (CASO 5) ao modelo anterior, o que funcionou muito bem, e mostrou que a curva de torque assim obtida é semelhante ao torque obtido pela dinâmica inversa tradicional do mecanismo 4 barras. Já que foi possível obter soluções satisfatórias com 1 e 3 atuadores, faltava verificar que curvas se obteria com um modelo de 2 atuadores com vínculos de mola e amortecedor (CASO 6). Por fim, retornou-se ao modelo do pêndulo com 3 atuadores (CASO 7), simulando-o sob o vínculo de molas e amortecedores para que esta resposta pudesse ser comparada com a resposta dos CASOS 1 e 2.

Em cada um dos casos, várias tentativas foram necessárias para se atingir os resultados “bons” mostrados nas Figuras 3.5 a 3.11. O critério para escolha do resultado considerado bom, se baseou em alguns fatores apresentados a seguir. A ordem de enumeração, reflete mais ou menos a ordem de importância de cada um deles.

- 1) Terminação normal do algoritmo de controle ótimo.
- 2) Correspondência entre a cinemática esperada e a cinemática obtida.
- 3) Ausência de picos na curva de controle que estivessem muito acima do valor médio da curva.
- 4) Suavidade da curva de controle.

A seguir, faremos a discussão dos resultados obtidos sob diversos aspectos. Dos sete casos apresentados, dois foram descartados dessa análise. Um deles, o CASO 3, simplesmente porque não houve resultado devido à natureza do sistema considerado. O outro, CASO 4, pelo fato de o melhor resultado obtido ter se baseado num modelo em que a aceleração da gravidade foi de  $g = 0 \text{ m/s}^2$ , o que portanto, não corresponde a um caso real.

Uma tabela geral com todos os principais resultados obtidos, bem como os parâmetros de entrada utilizados é apresentada no final deste capítulo (Tab. 3.5). Onde se fizer necessário, serão apresentadas tabelas adicionais, com informações complementares ou detalhes da tabela geral próximo ao trecho do texto que fizer referência a uma informação contida nela.

### 3.5.1 FUNÇÃO OBJETIVO

Da análise da expressão da função objetivo (Eq. 3.2), poder-se-ia supor que quanto maior o número de atuadores presentes, maior seria o valor desse índice de performance. No entanto, os valores obtidos para a função objetivo, mostrados na Tabela 3.5, evidenciam que esse comportamento seguiu um padrão mais complexo.

A ordem de grandeza dos valores obtidos nos permite dividir os casos analisados em dois grupos. No primeiro grupo, composto pelos CASOS 1, 2, 7, (em que se considera 3 atuadores) e ainda o caso da dinâmica inversa tradicional, chamada de *Din.Inv*, a ordem de grandeza da função objetivo ficou em torno de  $10^4$ . O valor da Equação 3.2, para o CASO *Din.Inv* foi estimado fazendo-se o cálculo da integral numericamente para fins de comparação. Dentre esses quatro casos, o menor valor obtido foi justamente do CASO *Din.Inv* ( $1,01 \cdot 10^4$ ), o que já era esperado. O segundo menor, foi obtido no CASO 7 ( $3,34 \cdot 10^4$ ), mostrando que a presença da mola e do amortecedor levou a uma redução da função objetivo em comparação com os casos em que não estava presente um ou ambos elementos passivos de força.

O segundo grupo, composto pelos CASOS 5 e 6, levou a resultados com uma ordem de grandeza na casa dos  $10^3$ . Como o número de atuadores presentes nos CASOS 5 e 6 é menor do que nos CASOS 1, 2 e 7; esse resultado também já era esperado. No entanto, surpreende o fato de que o valor da função objetivo do CASO 6 ter sido menor do que a do CASO 5. Aliás, o CASO 6 apresentou o menor valor entre todos. Isto é, mesmo possuindo um atuador a mais em comparação com o CASO 5, **a integral do torque quadrático (Eq. 3.2) ao longo de um ciclo do 4-barras foi menor para o caso com 2 atuadores do que quando comparado com os casos com 1 ou 3 atuadores.**

### 3.5.2 SELEÇÃO DOS ATUADORES PELO CRITÉRIO DO TORQUE MÁXIMO

Uma pergunta pertinente no contexto de atuação redundante em mecanismos como cadeia cinemática fechada é:

*“O aumento do número de atuadores (a introdução da redundância) permite selecionar atuadores que tenham uma especificação de torque máximo menor e que sejam, portanto, mais leves?”*

A análise da Tabela 3.3 pode ajudar a responder a esta questão. Nela são mostrados os valores máximos e mínimos nas curvas de torque em cada atuador para todos os casos analisados. O CASO *Din.Inv*. mostra que seria necessário utilizar um motor com torque máximo superior a 232 Nm para conduzir um mecanismo 4-barras com as características

adotadas neste trabalho. Nos CASOS 1, 2 e 7, em que se utilizou 3 atuadores, os torques máximos foram de duas a três vezes superiores a esse valor. No entanto, no CASO 6, o torque máximo obtido (em módulo) foi de 93 Nm. Portanto, **a solução do problema da dinâmica inversa através do OCP mostrou que é possível conduzir o mecanismo 4 barras com 2 atuadores utilizando-se motores com uma especificação de torque máximo cerca de 50% menor do que quando comparado com o 4 barras tradicional.**

TAB. 3.3: Torques máximos e mínimos dos atuadores durante 1 ciclo completo do mecanismo 4 barras

	U1		U2		U3	
	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>
<b>Din. Inv.</b>	203	-232	-	-	-	-
<b>CASO 1</b>	654,7	-160,4	391,1	-188,5	50,3	-72,1
<b>CASO 2</b>	497,5	-243,2	319,8	-214,9	34,0	-57,5
<b>CASO 5</b>	188,9	-199,4	-	-	-	-
<b>CASO 6</b>	80,9	-93,0	82,2	-92,1	-	-
<b>CASO 7</b>	433,5	-330,9	278,5	-264,4	41,4	-63,3

Três modelos (CASOS 1, 2 e 7) foram utilizados para descrever o mecanismo 4-barras com 3 atuadores. A Figura 3.12 permite comparar como os torques em cada atuador se comportaram quando havia a mola e o amortecedor presente, quando havia só a mola ou quando nenhum deles estava presente. De uma forma geral, as curvas mantiveram o mesmo aspecto nos três casos e apresentaram um *offset* para baixo à medida que se incorporavam molas e amortecedores ao modelo. Exceção feita para a curva  $u_3$  do CASO 7 (Fig. 3.5.2) que exhibe um comportamento diferente das demais.

### 3.5.3 ANÁLISE DO TRABALHO DOS ATUADORES

#### 3.5.3.1 TRABALHO TOTAL

A Tabela 3.4 mostra o resultado do trabalho executado por cada um dos atuadores em 1 ciclo completo ( $\phi_1 = 60^\circ$  até  $\phi_1 = 420^\circ$ ) do mecanismo 4 barras, calculado a partir deste algoritmo e a Figura 3.14 apresenta graficamente os dados desta tabela. Já a Figura 3.13 mostra as curvas dos ângulos relativos  $r_1, r_2$  e  $r_3$ , entre as barras, que foram utilizados para calcular o trabalho a partir do algoritmo apresentado.

Foi visto na Seção 2.5.2 que o trabalho executado por 1 atuador no caso do 4 barras regular é nulo. Isto pôde ser comprovado na primeira linha da Tabela 2.3, onde o trabalho

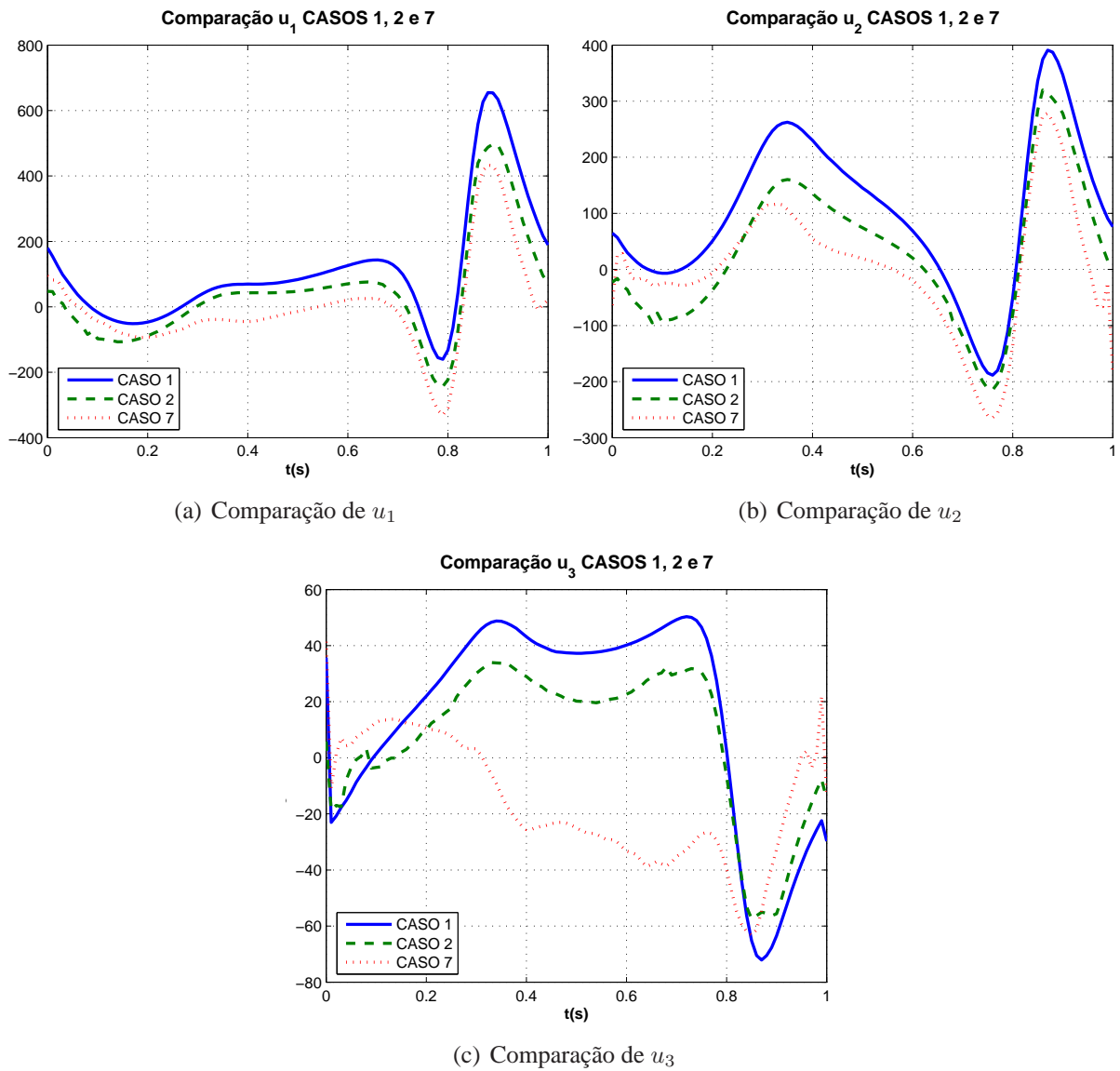


FIG. 3.12: Comparaç o dos torques nos casos em que foram utilizados 3 atuadores

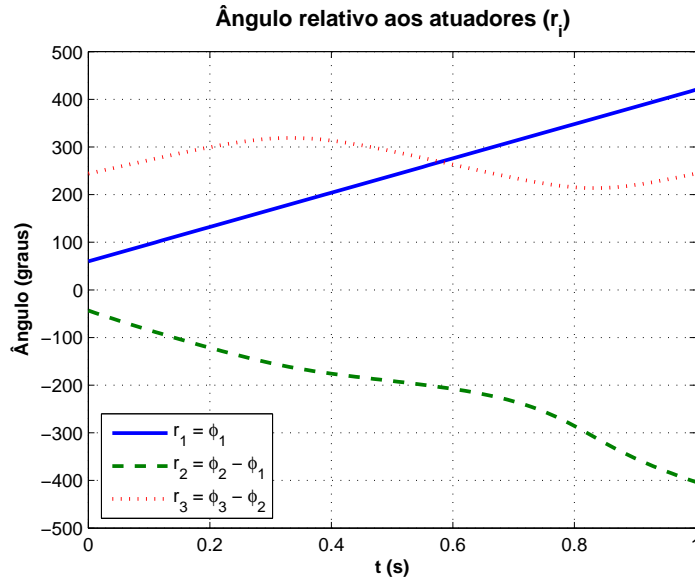


FIG. 3.13: Gráfico das coordenadas relativas  $r_i$  em função do tempo para 1 ciclo do 4 barras.

calculado foi numericamente nulo. No entanto, a última coluna da Tabela 3.4 mostra que o trabalho total foi diferente de zero. Em todos os casos, a variação da energia cinética e potencial entre as posições inicial e final foi desprezível, logo  $\Delta T = 0$  e  $\Delta V = 0$ . Neste caso, a Equação 2.53 ficaria  $W_i^f = W_{nc} = W_{tot}$ . Para os CASOS 5, 6 e 7, o fato de  $W_{tot}$  ter sido diferente de zero pode ser atribuído ao fato que nesses casos, pelo menos uma força não conservativa está presente: a força do amortecedor. Mesmo para o CASO 7 que teve o maior  $W_{tot}$ , 14,4 J representam uma quantidade de energia relativamente pequena. Este trabalho é equivalente ao trabalho necessário para se levantar 5,5 cm a massa total do sistema que é de 27.2 kg.

No entanto, no CASO 1 por exemplo, as únicas forças externas que agem sobre o sistema são os torques dos atuadores e mesmo assim  $W_{tot}$  foi de 9,1 J. Portanto, pela Equação 2.53 esse trabalho deve, necessariamente, ser devido a forças não conservativas. Isso nos leva a concluir que para este problema os torques que saíram da solução do problema de controle ótimo são torques não conservativos. Aparentemente, esta conclusão não entra em contradição com nenhuma hipótese simplificadora assumida anteriormente, pois na formulação do problema não há nenhuma informação a respeito da natureza (relação constitutiva) da força de controle. Portanto, parece ser admissível supor que o controle ótimo, na busca de minimizar a função objetivo fixada para este problema, encontre uma solução que não seja um torque conservativo. É interessante destacar ainda que o trabalho total dos casos analisados no Capítulo 3 em geral ficaram abaixo do trabalho total dos casos analisados no Capítulo 2.

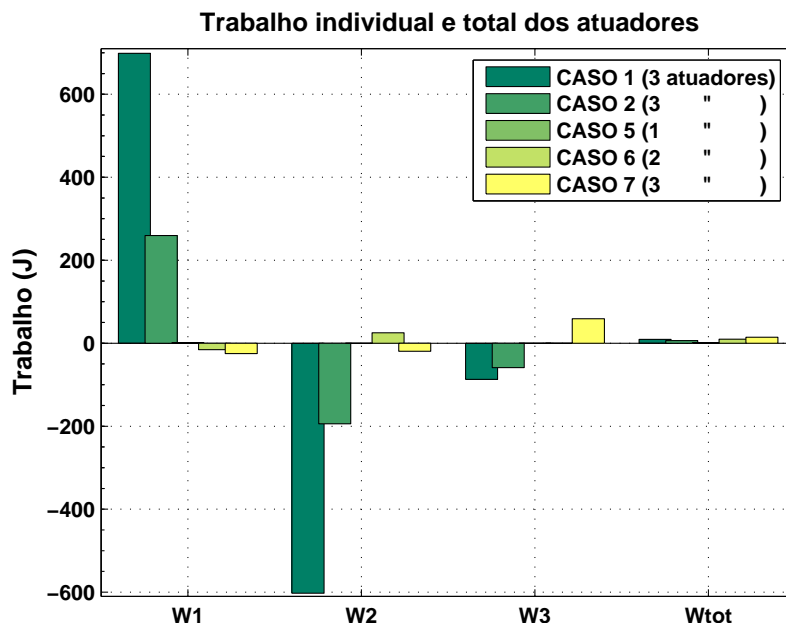


FIG. 3.14: Comparativo do trabalho total e individual dos atuadores ao longo de um ciclo completo.  $W_1$ ,  $W_2$  e  $W_3$  são os trabalhos dos atuadores nas articulações A, B e C, respectivamente e  $W_{tot} = W_1 + W_2 + W_3$

### 3.5.3.2 TRABALHO INDIVIDUAL E TRABALHO NEGATIVO

É interessante comparar os trabalhos individuais dos atuadores ( $W_1, W_2$  e  $W_3$ ) nos CASOS 1, 2 e 7, onde foi analisado o modelo do pêndulo triplo com 3 atuadores sem vínculo, com vínculo de mola e com vínculo de mola+amortecedor, respectivamente. Pode-se perceber que o trabalho individual de cada atuador no CASO 1 foi superior, em módulo, ao trabalho do atuador correspondente dos demais casos. Não coincidentemente, os torques máximos e mínimos do CASO 1, mostrados na Tabela 3.3, também são superiores aos demais. Isto pode ser explicado pelo fato de os atuadores serem responsáveis por suportar

TAB. 3.4: Trabalho dos atuadores durante 1 ciclo completo do mecanismo 4 barras nos casos estudados com a formulação OCP

	$W_1$ (J)	$W_2$ (J)	$W_3$ (J)	$W_{tot}$ (J)
<b>Din. Inv.</b>	-1,7e-15	0,0	0,0	-1,7e-15
<b>CASO 1</b>	698,7	-602,5	-87,1	9,1
<b>CASO 2</b>	259,5	-194,1	-59,0	6,5
<b>CASO 5</b>	1,4	0,0	0,0	1,4
<b>CASO 6</b>	-15,6	25,2	0,0	9,6
<b>CASO 7</b>	-25,2	-19,6	59,1	14,4



todo o peso das barras. Nos CASOS 2 e 7, parte do peso era suportado pelas molas e amortecedores presentes no ponto D.

Outro item que vale a pena ser comentado é quanto ao sinal do trabalho de alguns atuadores. Pela convenção de sinais usualmente adotada, quando o trabalho é positivo, energia está sendo fornecida ao sistema enquanto que quando o trabalho é negativo, energia está sendo retirada dele. Nos casos em que havia mais de um atuador presente, pelo menos um deles apresentou trabalho médio negativo. Isto significa que em alguma posição a barra se moveu numa direção e o torque foi aplicado na direção contrária. Portanto, este atuador que apresentou trabalho negativo teve o papel de “frear” o movimento. Na média, funcionando nesta condição, o motor estaria funcionando como um gerador.

Suponha-se que esse sistema de atuação seja capaz de armazenar essa energia gerada em um dos atuadores para posteriormente ser utilizada por outro atuador, ou mesmo por ele próprio, de forma perfeita e sem perdas. Então, a coluna  $\mathbf{W}_{\text{tot}}$  da Tabela 3.4, indica um limite inferior para o saldo líquido de energia que deveria ser fornecida ao sistema a cada ciclo. Por outro lado, se o sistema de atuação não fosse capaz de armazenar essa energia, o limite superior para energia que deveria ser fornecida ao sistema a cada ciclo seria o trabalho do atuador que apresentou trabalho positivo. Tomando como exemplo o CASO 6, seria necessário gastar no mínimo 9,6 J/ciclo e no máximo 25,2 J/ciclo por atuador para mantê-lo em funcionamento.

É importante ressaltar que o modelo utilizado nesta dissertação não foi deduzido com o objetivo de fazer uma análise de eficiência energética de mecanismos de cadeia fechada com atuadores redundantes. Ou seja, todos os atritos são considerados nulos, não se faz nenhuma menção ao modelo dinâmico do motor elétrico muito menos leva-se em conta a sua eficiência de conversão de energia. No entanto, os resultados mostrados nessa seção podem ser tomados como base para futura investigação acerca desse assunto.

TAB. 3.5: Tabela geral comparativa de parâmetros usados e resultados obtidos. PT = Pêndulo Triplo, 1A/2A/3A = 1, 2 ou 3 Atuadores

	<b>CASO 1</b> PT-3A s/vinculo	<b>CASO 2</b> PT-3A+mola	<b>CASO 5</b> PT-3A+mola/amort	<b>CASO 6</b> PT-2A+mola/amort	<b>CASO 7</b> PT-3A+mola/amort	<b>Din. Inv.</b> 4 Barras-1A
<b>Função Objetivo</b>	7,51E+04	4,36E+04	9,69E+03	4,27E+03	3,34E+04	1,01E+04 <sup>a</sup>
<b>Tempo de processamento (min)</b>	46,7	43,9	14,3	67,1	192,6	-
<b>W1 (J)</b>	698,7	259,5	1,4	-15,6	-25,2	-1,70E-015
<b>W2 (J)</b>	-602,5	-194,1	0,0	25,2	-19,6	-
<b>W3 (J)</b>	-87,1	-59,0	0,0	0,0	59,1	-
<b>Wtot (J)</b>	9,1	6,5	9,6	9,6	14,4	-1,70E-015
<b>U1 max/min (Nm)</b>	654,7/-160,3	497,4 /-243,1	188,8/-199,3	80,8/-93,0	433,4/-330,9	232,2/-203,4
<b>U2 max/min (Nm)</b>	391,0/-188,5	319,8/-214,9	-/-	82,1/-92,1	278,5/-264,3	-/-
<b>U3 max/min (Nm)</b>	50,3/-72,1	33,9/-57,4	-/-	-/-	41,4/-63,25	-/-
<b>Integrador</b>	RK4	RK4	LSODA	LSODA	LSODA	-
<b>Controle Inicial - (<math>u_0</math>)</b>	vetor nulo	vetor nulo	$u_{ol}$ simul. ant.	vetor nulo	$u_{ol}$ simul. ant.	-
<b>Discretizacao da malha temporal - N</b>	200	200	200	50	50	-
<b>Tol. de otimização - EPS</b>	1,00E-01	1,00E-01	1,00E-01	1,00E-01	1,00E-02	-
<b>Tol. das restrições - EPSNEQ</b>	1,00E-05	1,00E-03	1,00E-01	1,00E-01	1,00E-04	-
<b>Constante de Rigidez - K</b>	-	1,00E+03	5,00E+06	5,00E+04	1,00E+05	-
<b>Constante de Amortecimento - C</b>	-	-	1,00E+06	1,00E+04	1,00E+03	-
<b>Parâmetros da Função Objetivo -</b> (Eq. 3.2) - $w_1 / w_2 / w_3$	1 / 1 / 1	1 / 1 / 1	1 / 0 / 0	1 / 1 / 0	1 / 1 / 1	-
<b>Matriz [D] (Eq. 3.31)</b>	$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$	Idem CASO 1	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	Idem CASO 1	-

<sup>a</sup>Valor estimado da Eq. 3.2 para o caso da dinâmica inversa tradicional



#### 4. COMPARAÇÃO ENTRE OS MÉTODOS E CONCLUSÕES FINAIS

No decorrer desta dissertação foram investigadas diversas formas de se obter os torques necessários para fazer com que um mecanismo plano de cadeia fechada percorra uma cinemática prescrita, tanto com quanto sem redundância de atuadores. Particularmente, investigou-se o mecanismo 4 barras com atuadores de torque em suas articulações.

No Capítulo 2 utilizou-se a formulação oriunda da dinâmica de sistemas multi-corpos para tratar este problema. Dentre os diversos formalismos existentes na literatura sobre esse assunto, escolheu-se por trabalhar com as Equações Diferenciais Algébricas. Dentro deste contexto, foram expostos os principais conceitos e implementados algoritmos em linguagem Matlab para a solução dos problemas de análise cinemática, dinâmica direta e inversa de mecanismos planos presentes na literatura relativa a DAEs. Os livros básicos sobre o assunto, em geral, consideram apenas sistemas mecânicos em que o número de atuadores é igual ao número de graus de liberdade. Com base nisso, foi proposta uma modificação na formulação do vetor de restrições, introduzindo-se restrições diretoras adicionais, para poder realizar a dinâmica inversa de um mecanismo 4 barras com atuadores redundantes. Como esta formulação foi inconsistente, optou-se por implementar uma outra baseada na introdução da chamada Matriz de Transmissão (VALASEK e SIKKA (2001)). Esta abordagem permitiu encontrar as curvas de torque durante 1 ciclo completo do mecanismo para os casos com 1, 2, 3 e 4 atuadores. As curvas de torque obtidas com os algoritmos de dinâmica inversa foram checadas utilizando-as como torque externo conhecido na integração da equação de movimento do mecanismo. Como a cinemática produzida pela dinâmica direta reproduziu satisfatoriamente a cinemática obtida na dinâmica inversa, pode-se considerar que os resultados obtidos são factíveis. Por fim, foram comparadas as curvas de torque obtidas entre os casos com 1, 2, 3 e 4 atuadores.

Já no Capítulo 3 o problema da dinâmica inversa com atuadores redundantes foi encarado sob a ótica do controle ótimo. Uma apresentação geral da formulação de um problema de controle ótimo foi feita juntamente com uma breve descrição do *software* que foi utilizado para resolvê-lo. Em seguida apresentou-se a formulação específica do OCP abordado nesta dissertação. Nesta formulação, optou-se por trabalhar com o modelo dinâmico de um pêndulo triplo, e incorporar ao OCP restrições que o fizessem reproduzir a cinemática do mecanismo 4 barras. Sete casos foram identificados, simulados e analisados considerando-se diferentes números de atuadores e a presença ou não de molas e

amortecedores na extremidade do pêndulo, simulando as reações de apoio. O histórico da implementação/simulação desses casos foi contado enfocando os problemas numéricos (e suas soluções) encontrados. As curvas de torque desses sete casos foram analisadas e comparadas entre si sob o ponto de vista do torque máximo, trabalho individual dos atuadores, trabalho total líquido, bem como do valor da função objetivo.

#### 4.1 COMPARAÇÃO DOS RESULTADOS EQUIVALENTES: DAE $\times$ OCP

Uma visão geral sobre os problemas abordados e implementados nesta dissertação é mostrada na Figura 4.1. Nesta Figura mostra-se que a solução OCP com 1 atuador é similar à dinâmica inversa tradicional do mecanismo 4 barras (sem redundância de atuadores) obtida através da DAE<sup>31</sup>. Resta então fazer a comparação das curvas de torque obtidas para os casos com 2 e 3 atuadores (ver as setas curvas tracejadas na Figura 4.1), entre as abordagens DAE e OCP, visto que o modelo do pêndulo triplo não permite considerar o caso com 4 atuadores.

A Tabela 4.1 reúne as principais informações sobre os casos em que se considera 2 e 3 atuadores, contidas nas Tabelas 2.2, 2.3, 3.3, 3.4 e 3.5. Como no Capítulo 2 não havia nenhuma função objetivo explícita a ser minimizada, a Eq. 3.2 foi utilizada para calcular o valor da função objetivo para os casos com 2 e 3 atuadores utilizando-se a metodologia DAE modificada e os resultados foram colocados na primeira linha da Tabela 4.1. Como era de se esperar, o valor da função objetivo nos casos OCP ficaram inferiores ao da função objetivo dos casos DAE, sendo que quando o mecanismo possui 2 atuadores, essa diferença chegou a ser cerca de 10 vezes menor.

Com relação ao critério de torque máximo, a solução OCP para 2 atuadores se saiu claramente melhor do que a solução DAE, (vide Figura 4.2(a)). Enquanto que o torque máximo (em módulo) na solução DAE foi de 431 Nm e 239 Nm nas articulações A e B, os torques equivalentes na solução OCP foram de 93 Nm e 92.1 Nm. Já para o caso com 3 atuadores, a vantagem da solução OCP sobre a DAE, foi mais discreta. O torque máximo pela solução OCP foi superior na articulação A, semelhante na articulação B e inferior na articulação C. Isso dá uma pequena vantagem para o resultado obtido por OCP pois mesmo que o motor da articulação A tenha que ser mais pesado, esse motor não se moveria junto com as outras barras tal como aqueles que ficam em B e C.

---

<sup>31</sup>Neste Capítulo será adotada a sigla DAE, não apenas para abreviar “Equações Diferenciais Algébricas”, mas também para fazer referência aos resultados obtidos no Capítulo 2. Analogamente, OPC terá o significado tanto de “Optimal Control Problem” quanto uma alusão aos resultados do Capítulo 3

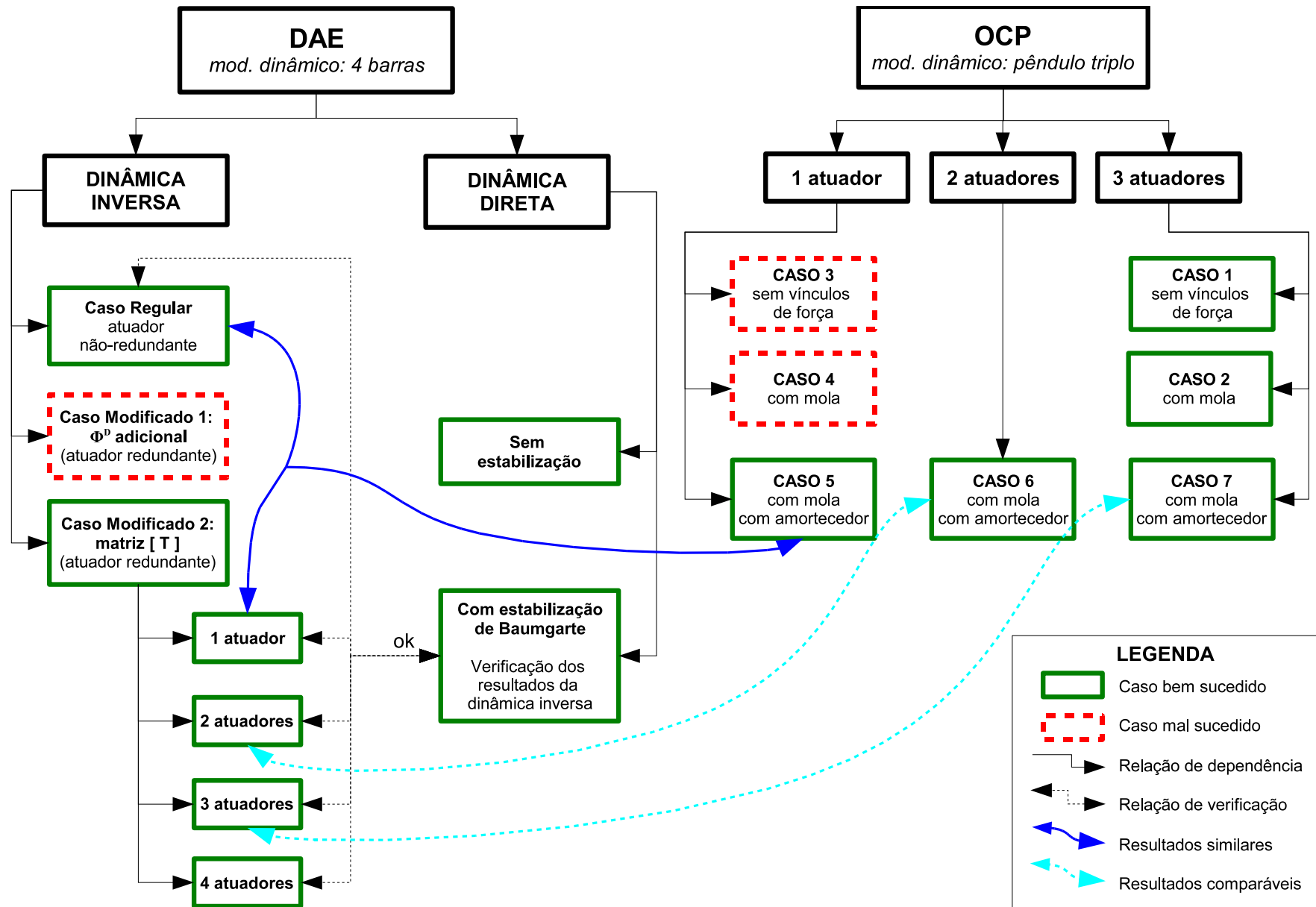


FIG. 4.1: Visão geral dos resultados obtidos

TAB. 4.1: Comparação entre os principais resultados entre as formulações DAE e OCP para os casos com 2 e 3 atuadores

	2 atuadores		3 atuadores	
	DAE	OCP	DAE	OCP
<b>Função objetivo</b>	3,25E+004	4,27E+003	4,11E+004	3,34E+004
<b>max</b> $\ \mathbf{u1}\ $ [Nm]	431	93	342	433,5
<b>max</b> $\ \mathbf{u2}\ $ [Nm]	289	92,1	288	278,5
<b>max</b> $\ \mathbf{u3}\ $ [Nm]	x	x	257	63,3
$\mathbf{W}_1$ [J]	355,2	-15,6	-72,7	-25,2
$\mathbf{W}_2$ [J]	-342,3	25,2	-147,5	-19,6
$\mathbf{W}_3$ [J]	x	x	241,0	59,1
$\mathbf{W}_{tot}$ [J]	12,9	9,6	20,9	14,4

Do ponto de vista do trabalho individual dos atuadores, as duas soluções obtidas por OCP também foram melhores. No caso de 2 atuadores, a solução OCP proporcionou um trabalho individual até 22 vezes menor que a solução DAE enquanto que no caso com 3 atuadores essa diferença chegou a ser de até sete vezes menor. A solução OCP também levou vantagem sobre a solução DAE no critério de trabalho total.

De uma forma geral, a solução OCP foi melhor em todos os três critérios de comparação (valor da função objetivo, torque máximo e trabalho dos atuadores). Direta ou indiretamente, esses três critérios dependem do valor do torque dos atuadores. Na formulação do OCP, existe uma função explícita que deve ser minimizada (Eq. 3.2), na qual estão presentes apenas os termos dos torques. Já na formulação DAE modificada pela Matriz de Transmissão, o cálculo da matriz pseudo-inversa implica na minimização do vetor  $\mathbf{X} = [\boldsymbol{\lambda}, \mathbf{n}]^T$ . A “função objetivo” que está implícita ao método da Seção 2.4.2 possui portanto, tanto os termos dos torques quanto os termos das forças internas. É possível então, que para minimizar ao máximo os torques, a formulação OCP tenha penalizado as forças internas do mecanismo, tornando-as maiores do que as encontradas na solução DAE. A seguir mostra-se um resumo das principais vantagens e desvantagens de ambos os métodos.

A abordagem DAE é mais fácil de implementar do que a OCP

O custo computacional da abordagem DAE é menor, o que permitiria até implementá-la para controle em tempo real.

A abordagem DAE não depende de um *software* específico, tal como o RIOTS e pode ser facilmente implementada em C ou Fortran.

A formulação OCP permite que a função objetivo seja modificada livremente pelo usuário.

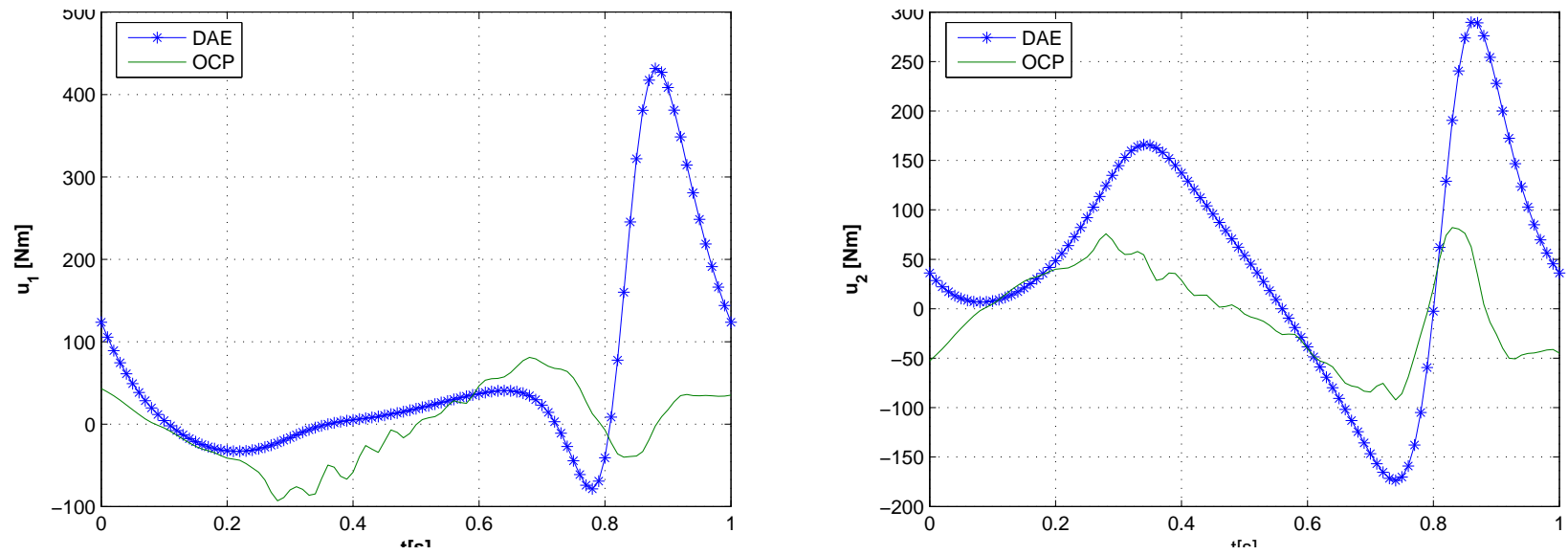
A solução OCP apresentou resultados melhores do que a solução DAE em praticamente todos os itens de comparação.

## 4.2 SUGESTÃO PARA TRABALHOS FUTUROS

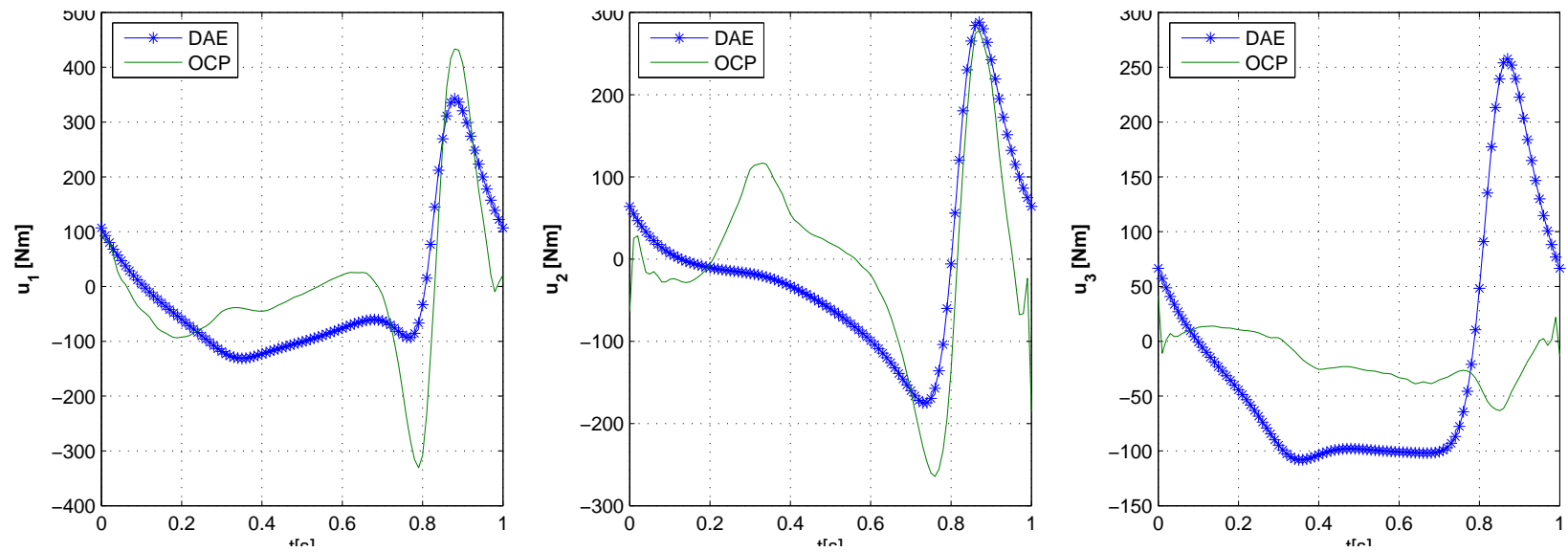
O desenvolvimento desta dissertação levantou uma série de questões e de problemas que não puderam ser ainda resolvidos. Listam-se aqui os principais tópicos que poderão servir de inspiração para futuros trabalhos.

1. A discussão da Seção 2.1.2.4 sobre coordenadas naturais mostrou que este sistema de coordenadas dá origem a equações de restrição quadráticas ou lineares, quando da formulação do sistema mecânico. Seria interessante explorar a implementação no RIOTS das equações de movimento usando coordenadas naturais, pois isso poderia levar a uma diminuição do custo computacional na solução do problema de controle ótimo.
2. Foi mostrado que o método DAE modificado pela Matriz de Transmissão funcionou para um mecanismo simples de cadeia fechada e este método possui algumas vantagens como facilidade de implementação e baixo custo computacional. Portanto, ele poderia ser utilizado para investigar o problema biomecânico da pedalada que é representado por um mecanismo de 3 GDL, 8 barras, e duas cadeias cinemáticas fechadas.
3. Por outro lado, o Capítulo 3 mostrou que a essência da construção do modelo dinâmico foi a abertura da cadeia cinemática fechada em um ponto e a substituição do vínculo cinemático por uma mola e amortecedor de constantes elevadas. Este mesmo princípio poderia ser aplicado ao problema da pedalada deduzindo as equações de movimento de dois pêndulos quádruplos com mesmo ponto fixo (articulação da coxa) e forças de mola e amortecedor para fixar as extremidades no eixo do pé de vela da bicicleta. Uma vez que esse modelo esteja pronto, poderá ser formulado um OCP semelhante ao do Capítulo 3 para encontrar os torques musculares resultantes em cada articulação.
4. Pode-se explorar diferentes funções objetivo da formulação do OCP incluindo, por exemplo, termos relativos à potência e trabalho dos atuadores. Essa modificação poderia ser aplicada aos CASOS 6 e 7 do Capítulo 3.





(a) 2 atuadores



(b) 3 atuadores

FIG. 4.2: Comparação dos torques para o 4 barras entre os resultados com formulação DAE e OCP.

5. Uma das formulações possíveis da função objetivo pode ser a inclusão das forças internas do mecanismo. Uma vez que seja possível a formulação deste problema, os resultados do OCP poderão ser comparados com os da DAE modificada pela Matriz de Transmissão.
6. Por outro lado, as soluções obtidas com o OCP para os casos com 2 e 3 atuadores poderiam servir como força externa conhecida na solução do problema de dinâmica direta cujo algoritmo foi apresentado na Seção 2.3.3. Isto permitiria calcular as forças internas presentes no 4 barras, quando este é forçado a girar com os torque que foram calculados pelo controle ótimo.
7. Tal como foi apontado na Seção 3.5, o CASO 6, onde se considera o 4 barras com 2 atuadores, apresentou resultados superiores aos demais. No entanto, no CASO 6 consideram-se os atuadores presentes nas articulações A e B. Falta ainda explorar se o posicionamento dos atuadores nas articulações B e C ou A e C trariam alguma vantagem sobre a configuração do CASO 6.
8. As equações de estado de motores elétricos podem ser incorporadas ao OCP de forma que sua solução seja expressa em unidades de tensão de armadura e não em torque. Isso seria um passo importante para futura implementação e comprovação experimental do método.
9. Na maioria das inúmeras tentativas feitas até se chegar aos resultados dos CASOS de 1 a 7 (Capítulo 3), utilizou-se um vetor nulo como estimativa inicial de controle, tal como pode ser visto nas tabelas do Apêndice B. Foi necessário utilizar a estratégia de resolver sucessivas vezes o OCP com valores crescentes da aceleração gravitacional (MENEGALDO et al., 2004), o que é bastante custoso computacionalmente. No CASO 1, por exemplo, o OCP chegou ser resolvido 14 vezes até chegar no resultado da Figura 3.5. A utilização dos torques obtidos através do método DAE modificado pela Matriz de Transmissão como estimativa inicial de controle para resolver o OCP pode ser uma alternativa para diminuir o número de tentativas.
10. Pode-se considerar atuadores musculares fictícios para o mecanismo 4 barras e empregar curvas de torque obtidas no Capítulo 2 ou 3 para testar a viabilidade de utilização do método “Inverse Dynamics Optimal Control - IDOC” (MENEGALDO et al., 2006) em sistemas de cadeia cinemática fechada.

11. A abordagem DAE pode ser rapidamente modificada para calcular os torques para diferentes velocidades da manivela. Com isso, o trabalho nos atuadores poderia ser calculado e os resultados analisados de forma a verificar se existe a mesma relação entre trabalho negativo e velocidade do pé de vela que NEPTUNE e HERZOG (1999) encontraram entre trabalho negativo e velocidade da pedalada.

### 4.3 CONSIDERAÇÕES FINAIS

Esta dissertação procurou apresentar a dinâmica inversa do mecanismo 4 barras com e sem redundância de atuadores nas formulações baseadas em Equações Diferenciais Algébricas e na Teoria de Controle Ótimo. Sua principal motivação se baseou na relação que este problema tem com a pedalada humana. Sob este aspecto foram levantadas questões tais como (i) se é possível revelar princípios de ótimo escondidos em tarefas relacionadas ao controle motor; (ii) quanto dos torques articulares de cada indivíduo estaria escondido na informação das curvas de reação nos pedais; (iii) estaria esta informação balizada por algum princípio de otimização ou mesmo (iv) se seria possível desenvolver um método de dinâmica inversa que usasse apenas informações relativas à cinemática do movimento como dado de entrada. Esta dissertação não tem a pretensão de querer responder questões em aberto como estas. No entanto foi possível traçar algumas conclusões que, apesar de estarem limitadas ao modelo do mecanismo 4 barras adotado, podem ter alguma analogia com o problema biomecânico.

A solução do problema da dinâmica inversa através da formulação DAE modificada pela introdução da Matriz de Transmissão não apresentou vantagens do ponto de vista da redução dos torques máximos quando compara-se o 4 barras redundante com o não redundante.

Já a solução do problema da dinâmica inversa através do OCP mostrou que é possível conduzir o mecanismo 4 barras utilizando-se motores com uma especificação de torque máximo menor do que quando comparado com o 4 barras sem redundância. Esta redução chegou a ser da ordem de 50% no caso de dois atuadores.

O cálculo do trabalho nos atuadores mostrou, tanto na solução OCP quanto na solução DAE, que a introdução da redundância aumenta o gasto energético a cada ciclo do mecanismo.

O importante ressaltar aqui é que o padrão das curvas de torque obtidas com o OCP foi diferente do padrão obtido com o método DAE. A explicação encontrada para o modelo

adotado foi que o método OCP procura minimizar apenas os torques enquanto que o método DAE modificado pela introdução da matriz de transmissão  $[\mathbf{T}]$  minimiza torques e reações internas. Portanto, seria interessante investigar qual dos dois métodos mais se aproxima de uma condição fisiológica, aplicando-os para um modelo biomecânico. A questão do trabalho nos atuadores também merece ser melhor investigada



## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- APOSTOL, T. M. *Calculus, One-Variable Calculus, With an Introduction to Linear Algebra*, volume I. John Willey and Sons, Inc, 2nd edition, 1967.
- APOSTOL, T. M. *Calculus, Multi-Variable Calculus and Linear Algebra, with Applications to Differential Equations and Probability*, volume II. John Willey and Sons, Inc, 2nd edition, 1969.
- ARNOLD, M., BURGERMEISTER, B. e EICHBERGER, A. **Linearly implicit time integration methods in real-time applications: DAEs and stiff ODEs** . *Multi-body System Dynamics*, 17(2-3):99–117, 2007.
- BARIN, K. **Evaluation of a Generalized Model of Human Postural Dynamics and Control in the Sagittal Plane** . *Biological Cybernetics*, 61(1):37–50, 1989.
- BAYO, E., GARCÍA DE JALÓN, J., AVELLO, A. e CUADRADO, J. **An efficient computational method for real time multibody dynamic simulation in fully cartesian coordinates**. *Computer Methods in Applied Mechanics and Engineering*, 92(3):377–395, 1991.
- BLAJER, W. e CZAPLICKI, A. **Modeling and inverse simulation of somersaults on the trampoline**. *Journal of Biomechanics*, 34(12):1619–1629, 2001.
- BOTTASSO, C. e CROCE, A. **On the Solution of Inverse Dynamics and Trajectory Optimization Problems for Multibody Systems**. *Multibody System Dynamics*, 11:1–22, 2004.
- BOYD, T., HULL, M. e WOOTTEN, D. **An improved accuracy six-load component pedal dynamometer for cycling** . *Journal of Biomechanics*, 29(8):1105–1110, 1996.
- BRENAN, K., CAMPBELL, S. e PETZOLD, L. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.
- CAHOUET, V., LUC, M. e DAVID, A. **Static optimal estimation of joint accelerations for inverse dynamics problem solution**. *Journal of Biomechanics*, 35(11):1507–1513, 2002.
- CHENG, H., YIU, Y.-K. e LI, Z. **Dynamics and Control of Redundantly Actuated Parallel Manipulators**. *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, 8(4):483–491, 2003.
- CITRON, S. *Elements of Optimal Control*. Holt, Rinehart and Winston, Inc, 2nd edition, 1969.
- COSSALTER, V. e LOT, R. **A Motorcycle Multi-Body Model for Real Time Simulations Based on the Natural Coordinates Approach**. *Vehicle System Dynamics*, 27(6):423–447, 2002.

- DABERKOW, A. e KREUZER, E. **An Integrated Approach for Computer Aided Design in Multibody System Dynamics.** *Engineering with Computers*, 15:155–170, 1999.
- DASGUPTA, B. e MRUTHYUNJAYA, T. **Force redundancy in parallel manipulators: theoretical and practical issues.** *Mechanism and Machine Theory*, 33(6): 727–742, 1998.
- DE FALCO, D., PENNISTRÌ, E. e VITA, L. **The udwadia-kalaba formulation: a report on its numerical efficiency in multibody dynamics simulations and on its teaching effectiveness.** MULTIBODY DYNAMICS, ECCOMAS Thematic Conference, 2005.
- DEL CITTO, F., PENNISTRÌ, E. e VITA, L. **Coordinates reduction and numerical integration of models of constrained multibody system.** European Congress on Computational Methods in Applied Sciences and Engineering, 2004.
- DOUGHTY, S. *Mechanics of Machines.* John Willey and Sons, Inc, 1988.
- GANOVSKI, L., FISETTE, P. e SAMIN, J. **Piecewise Overactuation of Parallel Mechanisms Following Singular Trajectories: Modeling, Simulation and Control.** *Multibody System Dynamics*, 12:317–343, 2004.
- GARCÍA DE JALÓN, J. **Twenty-five years of natural coordinates.** *Multibody System Dynamics*, 18(1):15–33, 2007.
- GARCÍA DE JALÓN, J. e BAYO, E. *Kinematic and Dynamic Simulation of Multibody Systems - the real-time challenge.* Springer-Verlag, New York NY, USA, 1994.
- GILL, P. E., MURRAY, W., SAUNDERS, M. A. e WRIGHT, M. H. *User's guide for NPSOL 5.0: A FORTRAN package for nonlinear programming* , 1998.
- GOLDSTEIN, H. *Classical Mechanics.* Addison-Wesley, 1970.
- GREENWOOD, D. T. *Principles of Dynamics.* Prentice-Hall Inc., 1965.
- HATZE, H. **The inverse dynamics problem of neuromuscular control.** *Biological Cybernetics*, 82(0):133–141, 2000.
- HAUG, E. J. *Computer Aided Kinematics and Dynamics of Mechanical Systems*, volume 1. Allyn and Bacon, Massachusetts, USA, 1989.
- HAUG, E. J. *Intermediate Dynamics.* Prentice Hall, Inc., New Jersey, USA, 1992.
- HULL, M. e DAVIS, R. **Measurement of pedal loading in bicycling: I. instrumentation.** *Journal of Biomechanics*, 14(12):843–856, 1981.
- HULL, M. e JORGE, M. **A method for biomechanical analysis of bicycle pedalling.** *Journal of Biomechanics*, 18(9):631–644, 1985.

- JENNINGS, L., FISHER, M., TEO, K. e GOH, C. **MISER3. Solving optimal control problems. An update.** *Advances in Engineering Software and Workstations*, 13(4): 190–196, 1991.
- KRAFT, D. **Algorithm 733 TOMP Fortran Modules for Optimal Control Calculations.** *ACM Transactions on Mathematical Software*, 20(3):262–281, 1994.
- LEE, S.-H., KIM, J., PARK, F. C., KIM, M., e BOBROW, J. E. **Newton-Type Algorithms for Dynamics-Based Robot Movement Optimization.** *IEEE TRANSACTIONS ON ROBOTICS*, 21(4):657–667, 2005.
- LEE, Y.-H., HAN, Y., IURASCU, C. C., e PARK, F. C. **Simulation-Based Actuator Selection for Redundantly Actuated Robot Mechanisms.** *Journal of Robotic Systems*, 19(8):379–390, 2002.
- LIU, G., WU, Y., WU, X., KUEN, Y. e LI, Z. **Analysis and Control of Redundant Parallel Manipulators.** Em *Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation*, págs. 3748–3754, 2001.
- LIU, X.-J., WANG, Q.-M. e WANG, J. **Kinematics, Dynamics and Dimensional Synthesis of a Novel 2-DoF Translational Manipulator.** *Journal of Intelligent and Robotic Systems*, 41(4):205–224, 2004.
- MENEGALDO, L. *Modelagem Biomecânica e Controle Ótimo da Postura Humana Através de Algoritmos Baseados na Teoria das Aproximações Consistentes.* Tese de Doutorado, PMC-EPUSP, São Paulo/SP, 2001.
- MENEGALDO, L., FLEURY, A. T. e WEBER, H. **A “cheap” optimal control method to estimate muscle forces in musculoskeletal systems.** *Journal of Biomechanics*, 2006.
- MENEGALDO, L., FLEURY, A. e WEBER, H. *Optimal control of human posture using algorithms based on consistent approximations theory*, págs. 425–450. CRC Press, 2004.
- MENEGALDO, L. e WEBER, H. *Modeling, Simulation and Control of Human Postural Dynamics with Induced Oscillatory Motion*, volume 1, págs. 271–274. Eds., Publ. AAM (American Academy of Mechanics) and ABCM (Associação Brasileira de Ciências Mecânicas), 1997.
- MENEGALDO, L. L., DE TOLEDO FLEURY, A. e WEBER, H. I. **Biomechanical modeling and optimal control of human posture.** *Journal of Biomechanics*, 36: 1701–1712, 2003.
- MERIAN, J. e KRAIGE, L. *Mecânica - Dinâmica.* LTC - Livros Técnicos e Científicos S.A, Rio de Janeiro, 4 edition, 1999.
- MILLER, K. **Dynamics of the new UWA robot.** Em *Proc. Australian Conference on Robotics and Automation*, 2001.



- MINGARDI, S. e PENNESTRÌ, E. **A Short Review of Numerical Methods and Software Tools for Solving Differential-Algebraic Equations.** Em *Atti Sixth Applied Mechanisms and Robotics Conference, Cincinnati, OH, Paper No. AMR99-053*, 1999. URL <http://www.ingegneriameccanica.org/papers/daesoni99.pdf>.
- MÜLLER, P. C. **Descriptor systems: pros and cons of system modelling by differential-algebraic equations.** *Mathematics and Computers in Simulation*, 2000.
- MOON, F. C. *Applied Dynamics: with applications to Multibody and Mechatronic Systems.* Wiley-Interscience, New York NY, USA, 1998.
- NAKAMURA, Y. e GHODOUSSI, M. **Dynamics computation of closed-link robot mechanisms with non-redundant and redundant actuators.** *IEEE Trans. on Robotics and Automation*, 5(3):294–302, 1989.
- NEPTUNE, R. e HERZOG, W. **The association between negative muscle work and pedaling rate.** *Journal of Biomechanics*, 32:1021–1026, 1999.
- NETO, M. A. e AMBRÓSIO, J. **Stabilization Methods for the Integration of DAE in the Presence of Redundant Constraints.** *Multibody System Dynamics*, 10: 81–105, 2003.
- NIKRAVESH, P. *Computer Aided Analysis of Mechanical Systems.* Prentice Hall Inc., 1988.
- PANDY, M. G. e BERME, N. **A Numerical Method for Simulating the Dynamics of Human Walking.** *Journal of Biomechanics*, 21(12):1043–1051, 1988.
- PARS, L. *A Treatise of Analytical Dynamics.* J. Willey & Sons, NY, 1968.
- PAUL, B. *Kinematics and Dynamics of Planar Machinery.* Prentice Hall, Englewood Cliffs, NJ, 1979.
- PENNESTRÌ, E. e VITA, L. **Strategies for the numerical integration of DAE systems in multibody dynamics.** *Computer Applications in Eng. Education*, 12(2): 106–116, 2004.
- PENNESTRÌ, E. e VITA, L. *Multibody dynamics in advanced education*, volume 2, págs. 345–370. Springer Netherlands, 2005.
- PETZOLD, L. **Differential/Algebraic Equations are not ODE's.** *SIAM Journal on Scientific and Statistical Computing*, 3(3):367–384, 1982. URL <http://link.aip.org/link/?SCE/3/367/1>.
- PETZOLD, L. **Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations.** *SIAM Journal on Scientific and Statistical Computing*, 4(1):136–148, 1983.
- POLAK, E. **On the use of consistent approximations in the solution of semi-infinite optimization and optimal control problems.** *Mathematical Programming*, 62(2):385–414, 1993.

- REDFIELD, R. e HULL, M. **On the relation between joint moments and pedaling rates at constant power in bicycling.** *Journal of Biomechanics*, 19(4):317–329, 1986.
- ROBERSON, R. e SCHWERTASSEK, R. *Dynamics of Multibody Systems*. Springer-Verlag, 1988.
- SCHIEHLEN, W. **Multibody System Dynamics: Roots and Perspectives.** *Multibody System Dynamics*, 1(2):149–188, 1997.
- SCHWARTZ, A. L. *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*. Tese de Doutorado, University of California at Berkley, 1996.
- SCHWARTZ, A. L., POLAK, E. e Y., C. *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.
- SILVA, O. L., DEDINI, F. G. e MENEGALDO, L. L. **Biomechanical Model for Net Joint Torques of Pedaling.** Em *Proc. 18th International Congress of Mechanical Engineering, Ouro Preto, Brazil*, 2005.
- SILVA, O. L. e MENEGALDO, L. L. **Inverse dynamics of redundantly actuated four-bar mechanism using an optimal control formulation.** Em *Proc. 19th International Congress of Mechanical Engineering, Brasilia, Brazil*, 2007.
- SILVA, O. L., MENEGALDO, L. L. e MOURA, C. L. A. **EOM2SS: Uma rotina para geração automática de equações de estado de sistemas biomecânicos.** Em *1º Encontro Nacional de Engenharia Biomecânica - ENEBI, Petrópolis, Rio de Janeiro*, 2007.
- STEJSKAL, V. e VALASEK, M. *Kinematics and Dynamics of Planar Machinery*. Marcel Dekker, Inc, 1996.
- TERCEIRO, E. e FLEURY, A. **Optimal Control Approach to the Vibrations of a Flexible One-Link Manipulator Carrying Moving Sliders.** Em *ABC M Symposium Series in Mechatronics*, volume 1, págs. 233–242, 2004.
- UMBERGER, B. R. e MARTIN, P. E. **Testing the planar assumption during ergometer cycling.** *Journal of Applied Biomechanics*, 17:55–62, 2001.
- UNDA, J., GARCÍA DE JALÓN, J., LOSANTOS, F. e ENPARANTZA, R. **A comparative study on some different formulations of the dynamic equations of constrained mechanical systems.** *Journal of Mechanisms, Transmissions, and Automation in Design*, 109:466–474, 1987.
- VALASEK, M., SIKÁ, Z., BAUMA, V. e VAMPOLA, T. **The innovative Potential of Redundantly Actuated PKM.** Em *Proc. of the 4th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany*, 2004.

- VALASEK, M. e SIKA, Z. **Evaluation of Dynamic Capabilities of Machines and Robots.** *Multibody System Dynamics*, 6(2):183–202, 2001.
- WEHAGE, R. A. e HAUG, E. J. **Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems.** *Transactions of ASME, Journal of Mechanical Design*, 104:247–255, 1982.
- WINTER, D. *Biomechanics and Motor Control of Human Movement.* Wiley-Interscience, Inc, 1990.
- YU, Q. e CHEN, I.-M. **A direct violation correction method in numerical simulation of constrained multibody systems.** *Computational Mechanics*, 26(1):52–57, 2000.

## APÊNDICES

## A. EOM2SS: DEDUÇÃO DA EQUAÇÃO DE MOVIMENTO

### A.1 INTRODUÇÃO

A simulação direta de sistemas mecânicos, também chamada de “dinâmica direta”, é tarefa corriqueira para o engenheiro que trabalha com sistemas multi-corpos. Para tal, há uma diversa gama de *softwares* comerciais disponíveis no mercado que funcionam como um sistema caixa-preta. Ou seja, quem modela e simula o sistema mecânico, não tem acesso aos algoritmos que rodam por trás do programa, nem às equações de movimento do sistema. Por um outro lado, se o engenheiro se propuser a deduzir as equações de movimento e desenvolver seus próprios programas, pode contar com ferramentas computacionais tais como o Matlab.

Nesta dissertação, foi necessário escrever as equações de movimento de sistemas de complexidade e tamanho de médio a grande. No entanto, até o presente momento, não existem comandos, nem *toolboxes* disponíveis “free of charge” na Internet, que facilitem a dedução das equações de movimento de um sistema mecânico. Uma outra dificuldade é que os integradores numéricos atualmente disponíveis, não são capazes de lidar com as equações de movimento na forma em que elas ficam quando são deduzidas, por exemplo, através das equações de Lagrange. Por se tratar de equações diferenciais de 2<sup>a</sup> ordem, elas devem ser transformadas em um sistema de equações diferenciais de 1<sup>a</sup> ordem.

No decorrer dessa dissertação, foi necessário desenvolver um programa que fosse capaz de deduzir as equações de movimento de um sistema mecânico, usando os recursos de manipulação de variáveis simbólicas do Matlab e em seguida reescrevê-las na forma de equações de estado. Dessa forma seria possível utilizar integradores numéricos de primeira ordem, tais como o ODE45, para fazer a simulação. Este apêndice mostra alguns detalhes do desenvolvimento do programa escrito pelo autor, batizado de EOM2SS (*Equation of Motion to State Space*), que foi utilizado para deduzir as equações de movimento do pêndulo triplo, utilizado no Capítulo 3.

### A.2 DESCRIÇÃO DO ALGORITMO

Existem diversas formas de se deduzir as equações de movimento do sistema. Uma das mais sistemáticas e diretas é através do uso das Equações de Lagrange. Uma vez que se determine a origem do sistema fixo de coordenadas e se identifique um número

de coordenadas absolutas e independentes entre si (que descrevam de forma única uma posição do sistema em questão). É conceitualmente simples a dedução de tais equações. Basta que se escreva o vetor posição dos centros de massa de cada corpo rígido, em função das coordenadas independentes, para que a equação de movimento possa ser deduzida apenas com derivações (totais e parciais) e operações aritméticas simples.

### A.3 DIFERENCIAÇÃO

Apesar da dedução da equação de movimento ser conceitualmente simples, sua implementação computacional não é tal trivial. Uma das dificuldades encontradas foi o cálculo das derivada total ( $\frac{d}{dt}$ ) que aparece tanto no cálculo das velocidades dos centros de massa quanto na primeira parcela da equação de Lagrange ( $\frac{d}{dt} \frac{\partial T}{\partial \dot{q}}$ ). O comando `diff` disponível no Matlab não é capaz de calcular a derivada de uma expressão com múltiplas variáveis, em relação a todas as variáveis da expressão. Este problema foi resolvido usando-se a *function* `fulldiff`<sup>32</sup>. O exemplo abaixo mostra a diferença do uso da *function* `diff` para a *function* `fulldiff`.

```
pos2 =
  1/2*L1*cos(phi1)+1/2*L2*cos(phi2)
  1/2*L1*sin(phi1)+1/2*L2*sin(phi2)
```

```
>> diff(pos2)
ans =
  -1/2*L1*sin(phi1)
   1/2*L1*cos(phi1)
```

```
>> fulldiff(pos2)
ans =
  -1/2*dphi1*L1*sin(phi1)
   1/2*dphi1*L1*cos(phi1)
```

```
>> fulldiff(pos2,{phi1,phi2})
ans =
```

---

<sup>32</sup>Disponível em [www.mathworks.com/matlabcentral/](http://www.mathworks.com/matlabcentral/)

$$\begin{aligned}
& -1/2*dphi1*L1*sin(phi1)-1/2*dphi2*L2*sin(phi2) \\
& 1/2*dphi1*L1*cos(phi1)+1/2*dphi2*L2*cos(phi2)
\end{aligned}$$

Como pode ser observado neste exemplo, a *function* `fulldiff` calcula a derivada total de uma expressão em relação ao tempo. Ou seja:

$$\frac{df}{dt} = \frac{df}{dx} \frac{dx}{dt} + \frac{df}{dy} \frac{dy}{dt} \tag{A.1}$$

#### A.4 AGRUPAMENTO DOS TERMOS

Outro problema encontrado foi como reescrever a equação de movimento automaticamente na forma de equações de estado. Uma equação diferencial de 2ª ordem que representa o comportamento de um sistema mecânico, em geral, possui termos que aparecem multiplicados pela 2ª derivada, pela 1ª derivada elevada ao quadrado e/ou pela 1ª derivada, pode possuir termos independentes que não são multiplicados por variável nenhuma e eventualmente, termos que aparecem multiplicados pela própria variável. Além disso, o lado direito da equação de Lagrange (Eq. 3.6), relativo ao termo de forças generalizadas, pode conter termos de forças de controle e forças que dependem das variáveis de estado (molas e amortecedores, por ex.). Portanto, esta estrutura pode ser expressa matricialmente da seguinte forma:

$$[A] \{\ddot{q}\} + [B] \{\dot{q}^2\} + [C] \{\dot{q}\} + [D] \{q\} + [E] = [F] \{u\} + [G] \tag{A.2}$$

Quando as equações de movimento têm tamanho reduzido, é possível reescrevê-las agrupando cada termo adequadamente de modo que se consiga identificar os elementos das matrizes  $[A],[B],[C],[D],[E]$ ,  $[F]$  e  $[G]$ . Fazer isso automaticamente para grandes equações foi um desafio. A estratégia adotada foi reagrupar cada uma das variáveis (de ordem 0, 1 e 2), recursivamente com o comando `collect`. Em seguida, com o auxílio da *function* `SymPoly2Sym`<sup>33</sup>, cada um dos elementos que apareciam multiplicados por uma das variáveis eram colocados na posição apropriada de cada uma das matrizes. Uma vez que as matrizes eram calculadas o vetor de acelerações poderia ser isolado da seguinte forma:

---

<sup>33</sup>Disponível em [www.mathworks.com/matlabcentral/](http://www.mathworks.com/matlabcentral/)

$$\{\ddot{q}\} = [A]^{-1} \cdot ([F] \{u\} + [G] - [B] \{\dot{q}^2\} - [C] \{\dot{q}\} - [D] \{q\} - [E]) \quad (\text{A.3})$$

## A.5 VARIÁVEIS DE ESTADO

Uma vez que o vetor  $\{\ddot{q}\}$  esteja calculado, basta apenas fazer a substituição das variáveis utilizadas no problema em variáveis de estado.

$$\begin{aligned} x1 &= \text{phi1} & \rightarrow & \text{dx1} = x4 \\ x2 &= \text{phi2} & \rightarrow & \text{dx2} = x5 \\ x3 &= \text{phi3} & \rightarrow & \text{dx3} = x6 \\ x4 &= \text{dphi1} & \rightarrow & \text{dx4} = \text{qpp}(1) \\ x5 &= \text{dphi2} & \rightarrow & \text{dx5} = \text{qpp}(2) \\ x6 &= \text{dphi3} & \rightarrow & \text{dx6} = \text{qpp}(3) \end{aligned}$$

e em seguida montar o vetor da derivada das variáveis de estado.

$$\dot{x} = \begin{pmatrix} x4 \\ x5 \\ x6 \\ \text{qpp}(1) \\ \text{qpp}(2) \\ \text{qpp}(3) \end{pmatrix} \quad (\text{A.4})$$

## A.6 OUTPUT DO PROGRAMA

A parte final do algoritmo permite que o *output* do programa seja de duas formas. A primeira, simplesmente retorna o vetor  $\{\dot{x}\}$  para o *workspace* do Matlab. A segunda escreve a *function* que deve ser fornecida ao comando ODE, em um arquivo *.m*. O exemplo abaixo, mostra como seria a estrutura desse arquivo.

```
function xp =teste_pend_triplo(t,x)
global L1 L2 L3 m1 m2 m3 I1 I2 I3 g
x1 = x(1);
x2 = x(2);
x3 = x(3);
```



```

x4 = x(4);
x5 = x(5);
x6 = x(6);
xp = [
x4
x5
x6
(-.318*U1+.318*U2+.341e-1*sin(-1.*x3+x1)*x6^2-.375*U3*cos(x3+x1-2.*x2)+ ...
(-.651*U2+.651*U3+.104*x4^2*sin(x2+x1-2.*x3)+.374*U1*cos(-1.*x2+x1)-...
(-1.97*U3-.113*sin(-1.*x3+x1)*x4^2-.375*U1*cos(x3+x1-2.*x2)+...
];

```

## A.7 FUTURAS IMPLEMENTAÇÕES

A presente implementação se baseou na dedução das equações de movimento do pêndulo triplo e na colocação delas na forma de equações de estado. Futuramente, o código pode ser alterado, retirando-se a dedução da equação de movimento, para que transforme qualquer sistema de equações do tipo da Eq. A.2, em um sistema de equações do tipo da Eq. A.4. O código fonte deste algoritmo bem como informações adicionais sobre sua utilização podem ser encontrados na Seção C.4.2.

## B. TABELAS DE HISTÓRICO DA SOLUÇÃO DO OCP

Tal como foi mencionado, os resultados apresentados no Capítulo 3 foram obtidos depois de diversas tentativas utilizando-se diferentes parâmetros nas simulações com o RIOTS. Este Apêndice apresenta as tabelas que foram utilizadas para registrar cada uma das tentativas dos sete casos apresentados. Os resultados apresentados no Capítulo 3, correspondem à tentativa que aparece em destaque. As abreviaturas utilizadas nas tabelas são apresentadas a seguir.

### Lista de abreviaturas utilizadas nas Tabelas B.1 a B.6

**Nº int.** - Número de iterações

**Stop** - Mensagem retornada pelo RIOTS relativo ao critério de parada

**tempo** - tempo de processamento do riots

**Função Objetivo** - Expressão adotada para a função objetivo (*integrando*) ou valor final obtido (*valor final*)

**g** - Aceleração da gravidade

**tf** - Tempo final

**tol.geral** - Overall optimization tolerance. default = 1E-6

**tol.const** - Nonlinear constraint tolerance. default = 1E-4

**Condição inicial (var. estado)** - Condição inicial adotada para as variáveis de estado

**Cond. inicial (var. controle)** - Condição inicial adotada para as variáveis de controle

**NLTIC** - Non Linear Trajectory Inequality Constraint = vínculo de desigualdade de estado

**NLEEC** - Non Linear Endpoint Equality Constraint (vínculos terminais não-lineares de igualdade)

**N** - Número de pontos da malha temporal

**NA** - Não está Ativo

**MzA** - Torque na articulação A calculado pela dinâmica inversa.

**$\Delta t$**  - Delta t =  $t_f/N$  (incremento de tempo)

**l/nl** - Linear / Non Linear

### **Lista de mensagens do RIOTS**

Durante as simulações o RIOTS apresentou diversas mensagens ao terminar o programa. A seguir, apresentam-se os códigos utilizados nas Tabelas B.1 a B.6 para identificá-las bem como as mensagens exibidas.

**Normal** - Normal termination

**MNI** - Completed maximum number of iterations.

**LSF** - Line search failed.

**SV** - "Segmentation violation detected" É um erro do mex file. O arquivo gerado fica gravado num diretório temporário.

**NFP** - No feasible point found for nonlinear constraints.

**FPE** - Floating point error during constraint gradient computation.

**CECG** - Couldn't evaluate constraint gradient.

**HUMMM** - confun: Backtracking the linesearch... Hum, I don't know about this message: -1.

**KKC** - Kuhn-Tucker conditions satisfied but sequence did not converge.



TAB. B.2: Tabela de simulações do CASO 2

CASO 2 - 3 atuadores + mola

Prefixo Nome: Ivbar3act\_mola\_EOM2SS\_

Sufixo	Nºint	Stop	tempo	Função Objetivo	K	g	tf	RK	tol.geral	tol.const	Condição inicial (var. estado)						Cond. Inicial (controle)			NLTIC	NLEEC	N	Δt			
arq.m			(min) valor final	integrando	N/m	m/s <sup>2</sup>	s	ialg	EPS	EPSNEQ	phi1	phi2	phi3	phi1p	phi2p	phi3p	U1	U2	U3	nº	expressão	nº	expr.			
001	294	Normal	43,8	4,36E+04	u1^2+u2^2+u3^2	1,0E+03	4	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	$(x^2-2\pi)^2 + (-L1\sin(x1) + L2\sin(x2) - L3\sin(x3) - L4)^2 + (L1\cos(x1) + L2\cos(x2) + L3\cos(x3))^2 + (-L1x^4\cos(x1) - L2x^5\cos(x2) - L3x^6\cos(x3))^2 + (-L1x^4\sin(x1) - L2x^5\sin(x2) - L3x^6\sin(x3))^2$	0	0	200	0,005
002	936	Normal	183,6	8,21E+05	u1^2+u2^2+u3^2	1,0E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	$(x^2-2\pi)^2 + (-L1\sin(x1) + L2\sin(x2) - L3\sin(x3) - L4)^2 + (L1\cos(x1) + L2\cos(x2) + L3\cos(x3))^2 + (-L1x^4\cos(x1) - L2x^5\cos(x2) - L3x^6\cos(x3))^2 + (-L1x^4\sin(x1) - L2x^5\sin(x2) - L3x^6\sin(x3))^2$	0	0	200	0,005
003	981	LSF	243,4	1,30E+08	u1^2+u2^2+u3^2	1,0E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	uol simul001	ud simul001	uol simul001	1(5)	$L3x^6\sin(x3)^2$	0	0	200	0,005
004	564	Normal	94,7	4,09E+04	u1^2+u2^2+u3^2	1,0E+05	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	uol simul001	ud simul001	uol simul001	1(5)	$kdam simul 001$	0	0	200	0,005
005	0	FPE	0	0,00E+00	u1^2+u2^2+u3^2	1,0E+06	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	uol simul001	ud simul001	uol simul001	1(5)	$kdam simul 002$	0	0	200	0,005
006	0	CECG	0	0,00E+00	u1^2+u2^2+u3^2	1,0E+07	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	uol simul001	ud simul001	uol simul001	1(5)	$kdam simul 002$	0	0	200	0,005
007	361	LSF	951,5	3,83E+04	u1^2+u2^2+u3^2	1,0E+05	9,81	1,00	5	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	$kdam simul 002$	0	0	200	0,005
008	124	Normal	300,5	9,35E+03	u1^2+u2^2+u3^2	1,0E+05	9,81	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	$kdam simul 002$	0	0	200	0,005
009	293	Normal	734,3	2,35E+04	u1^2+u2^2+u3^2	1,0E+05	9,81	1,00	5	1,00E-01	1,00E-02	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	$kdam simul 002$	0	0	200	0,005

0/1

TAB. B.3: Tabela de simulações do CASO 4

CASO 4- 1 atuador + mola

Prefixo Nome: IVbar3act\_mola\_EOM2SS\_

Sufixo	Nome	Nºint	Stop	tempo	Função Objetivo	K	g	tf	RK	tol.geral	tol.const	Condição inicial (var. estado)					Cond. Inicial (controle)			NLTIC	NLEEC	N	Δt		
arq.m				(min)	valor final	integrando	N/m	m/s²	s	ialg	EPS	EPSNEQ	phi1	phi2	phi3	phiip	phi3p	phi3p	U1	U2	U3	nº	expressão	nº	expr.
001	101	NFP		15,4	5,81E+07	u1²	1,E+05	4	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	L3*x5*sin(x3)/2	0	0 200 0,005
002	12	LSF		1	5,73E+02	u1²	1,E+05	4	0,20	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,001
003	100	NFP		14,8	1,12E+05	u1²	1,E+05	4	0,20	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,001
004	11	LSF		1	4,17E+03	u1²	1,E+03	4	0,20	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,001
005	75	LSF		11	3,14E+06	u1²	1,E+05	4	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
006	22	NFP		2,7	3,76E+05	u1²	1,E+06	4	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
007	8	LSF		0,4	2,86E+03	u1²	1,E+05	0	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
008	9	LSF		0,8	7,29E+03	u1²	1,E+06	9,81	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
009	78	NFP		12,2	1,27E+06	u1²	1,E+07	9,81	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
010	0	CECG		0	0,00E+00	u1²	1,E+09	9,81	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
011	111	NFP		20,2	8,63E+05	u1²	1,E+08	9,81	0,05	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
012	0	FPE		0	0,00E+00	u1²	1,E+08	9,81	1,00	4	1,00E-01	1,00E-05	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,00025
013	0	FPE		0	0,00E+00	u1²	1,E+06	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
014	197	NFP		38	8,53E+05	u1²	1,E+05	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
015	154	LSF		34,9	4,82E+07	u1²	1,E+04	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
016	187	LSF		29,3	2,53E+06	u1²	5,E+04	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
017	217	LSF		52,2	3,56E+07	u1²	5,E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
018	78	NFP		14	1,01E+06	u1²	1,E+06	4	1,00	4	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
019	32	NFP		3,6	6,70E+03	u1²	1,E+05	0	1,00	4	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
020	15	NFP		3	7,06E+03	u1²	1,E+05	0	1,00	4	1,00E-01	1,00E-02	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	u1 simul019	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
021	65	NFP		15,5	1,39E+06	u1²	1,E+05	0	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	u1 simul020	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
022	29	NFP		54,4	1,02E+04	u1²	1,E+05	0	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
023	39	Normal		301,4	1,07E+04	u1²	1,E+07	0	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
024	43	LSF		440,8	1,03E+07	u1²	1,E+07	0	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	u1 de IVbar3act_EOM2SS_057	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005
025	3	LSF		20,5	5,80E+03	u1²	1,E+07	1	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2632	-1,3760	3,4240	u1 simul023	zeros(1,N+1)	zeros(1,N+1)	1(5)	klem simul 001	0	0 200 0,005







TAB. B.6: Tabela de simulações do CASO 7

CASO 7 - 3 atuadores + mola + amortecedor

Prefixo Nome: Mbar3act\_mola\_amort\_EOM2SS\_

Sufixo	Nome	NºInt	Stop	tempo	Função Objetivo	K	C	g	tf	RK	tol.geral	tol.const	Condição inicial (var. estado)						Cond. Inicial (controle)			NTIC	NLEEC	N	Δt	
arq.m			(min)	valor final	integrando	N/m	N.s/m	m/s <sup>2</sup>	s	ialg	EPS	EPSNEQ	phi1	phi2	phi3	phi1p	phi2p	phi3p	U1	U2	U3	nº	expressão	nº	expr.	
001	642 Normal		123,3	5,39E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	(x/2-pi) <sup>2</sup> + (-L1*sin(x1) + L2*sin(x2) - L3*sin(x3) - L4) <sup>2</sup>	0	0 200	0,005
002	532 Normal		103,1	1,11E+05	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+04	1,0E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	+ (L1*cos(x1) + L2*cos(x2) + L3*cos(x3)) <sup>2</sup> + (-	0	0 200	0,005
003	0 CECG		0,0	0,00E+00	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+04	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	L1*x4*cos(x1) - L2*x5*cos(x2) - L3*x6*cos(x3)) <sup>2</sup> + (-	0	0 200	0,005
004	455 Normal		1387,8	4,68E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+04	9,81	1,00	5	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)	L1*x4*sin(x1) - L2*x5*sin(x2) - L3*x6*sin(x3)) <sup>2</sup>	0	0 200	0,005
005	160 Normal		455,2	2,21E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+04	9,81	1,00	5	1,00E-02	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul004	u2 simul004	u3 simul004	1(5)		0	0 200	0,005
006	5000 MNI		9863,0	4,42E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+04	9,81	1,00	5	1,00E-03	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul005	u2 simul005	u3 simul005	1(5)		0	0 200	0,005
007	224 LSF		362,0	3,74E+05	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	4	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)		0	0 400	0,0025
008	17 Normal		171,8	2,53E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)		0	0 400	0,0025
009	35 Normal		37,7	1,45E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-01	1,00E-01	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)		0	0 50	0,02
010	316 Normal		324,9	5,36E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-01	1,00E-03	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	zeros(1,N+1)	zeros(1,N+1)	zeros(1,N+1)	1(5)		0	0 50	0,02
011	234 Normal		220,6	3,65E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-01	1,00E-04	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul010	u2 simul010	u3 simul010	1(5)		0	0 50	0,02
012	208 Normal		192,5	3,34E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-02	1,00E-04	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul011	u2 simul011	u3 simul011	1(5)		0	0 50	0,02
013	1000 MNI		877,4	4,97E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-03	1,00E-04	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul012	u2 simul012	u3 simul012	1(5)		0	0 50	0,02
014	194 Normal		199,0	3,37E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	1,00E-02	1,00E-04	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul012	u2 simul012	u3 simul012	1(5)		0	0 50	0,02
015	1000 MNI		761,6	4,47E+04	u1 <sup>2</sup> +u2 <sup>2</sup> +u3 <sup>2</sup>	1,0E+05	1,0E+03	9,81	1,00	5	5,00E-03	1,00E-04	1,0472	0,2907	4,5513	6,2832	-1,3760	3,4240	u1 simul014	u2 simul014	u3 simul014	1(5)		0	0 50	0,02

152

## C. CÓDIGO FONTE

Neste apêndice apresenta-se o código fonte dos diversos algoritmos desenvolvidos e citados ao longo da dissertação.

### C.1 DINÂMICA INVERSA SEM REDUNDÂNCIA

Apresenta-se aqui o algoritmo para se resolver a dinâmica inversa do mecanismo 4 barras sem redundância de atuadores, tal como é abordado na Seção 2.3.2.

```
%% 4 BARRAS – ANÁLISE CINEMÁTICA & DINÂMICA INVERSA
% Instituto Militar de Engenharia, Olavo Luppi Silva – junho/2007

%% DESCRIÇÃO:
% Neste arquivo o problema cinemático de posição, velocidade e aceleração
% do mecanismo 4 barras, é resolvido com o algoritmo de Newton-Raphson bem
% como são calculados suas forças internas através dos lambdas. O torque
% que faz o mecanismo girar à velocidade prescrita está associado ao
% elemento do vetor lambda que contém a equação de "drive" cinemático (isto
% é, dependente do tempo).

%%
clc
clear all
close all

%% CÁLCULO SIMBÓLICO:
%% Definição de variáveis
syms L1 L2 L3 L4
syms x1 x2 x3 x4 x1p x2p x3p
syms y1 y2 y3 y4 y1p y2p y3p
syms phi1 phi1_0 phi2 phi3 phi1p phi2p phi3p
syms w t g
syms m1 m2 m3 J1 J2 J3
syms s1a s1b s1b s2c s3c s3d
syms ptoA ptoD
```

```

%% Matrizes de rotação
A1 = [cos(phi1) -sin(phi1); sin(phi1) cos(phi1)];
A2 = [cos(phi2) -sin(phi2); sin(phi2) cos(phi2)];
A3 = [cos(phi3) -sin(phi3); sin(phi3) cos(phi3)];

%% Vetor posição dos centros de massa:
r1 = [x1;y1];
r2 = [x2;y2];
r3 = [x3;y3];

%% Vetor posição dos pontos de interesse (articulações) em relação aos sist.
%% de coordenadas locais das barras
s1a = [-L1/2;0]; s1b = [L1/2;0];
s2b = [-L2/2;0]; s2c = [L2/2;0];
s3c = [-L3/2;0]; s3d = [L3/2;0];

%% Vetor de posição absoluta dos pontos fixos A e D:
XA = 0.2;      YA = 0.2;
XD = XA + L4;  YD = YA;
ptoA = [XA; YA];
ptoD = [XD; YD];

%% Vetor de restrições cinemáticas:
sPHI = [r1 + A1*s1a - ptoA           % Pto A é fixo
        r1 + A1*s1b - (r2 + A2*s2b)% Soma vetorial do CG1 até ptoB é igual a CG2 até pto
        r2 + A2*s2c - (r3 + A3*s3c)% Soma vetorial do CG2 até ptoC é igual a CG3 até pto
        r3 + A3*s3d - ptoD           % Pto D é fixo
        phi1 - phi1_0 - w*t]        % Driver cinemático

%% Vetor de coordenadas generalizadas e de velocidades
sq = [x1; y1; phi1; x2; y2; phi2; x3; y3; phi3]
sqp = [x1p; y1p; phi1p; x2p; y2p; phi2p; x3p; y3p; phi3p]

sPHIq = jacobian(sPHI,sq)
sPHIt = diff(sPHI,t)
sPHItt = diff(sPHIt,t)
sPHIqt = diff(sPHIq,t)
sgamma = -jacobian(sPHIq*sqp,sq)*sqp - 2*sPHIqt*sqp - sPHItt
sM = diag([m1,m1,J1,m2,m2,J2,m3,m3,J3])
sQA = [0; -m1*g; 0; 0; -m2*g; 0; 0; -m3*g; 0] % Esse vetor segue a ordem do
                                                % vetor de coord. generalizadas

```

```

%% INICIALIZAÇÃO DOS PARÂMETROS NUMÉRICOS:
%% Valores constantes:
L1 = 0.5; % Comprimento da barra 1 [m]
L2 = 0.9; % Comprimento da barra 2 [m]
L3 = 0.7; % Comprimento da barra 3 [m]
L4 = 1.0; % Comprimento da barra 4 [m]
w = 2*pi; % Velocidade angular da barra 1 [rad/s]
g = 9.81; % Aceleração gravitacional [m/s^2]
t = 0; % Tempo inicial [s]
m1 = 6.590; % Massa da barra 1 [Kg]
m2 = 11.550; % Massa da barra 2 [Kg]
m3 = 9.070; % Massa da barra 3 [Kg]
J1 = (1/12)*m1*L1^2; % Momento de inércia em torno do CG da barra 1 [Kg.m^2]
J2 = (1/12)*m2*L2^2; % Momento de inércia em torno do CG da barra 2 [Kg.m^2]
J3 = (1/12)*m3*L3^2; % Momento de inércia em torno do CG da barra 3 [Kg.m^2]

%% Valores iniciais estimados a partir do desenho do Inventor para phi1 = 60°
x1 = 0.125; y1 = 0.216; phi1 = 0; phi1_0 = 60*pi/180;
x2 = 0.681; y2 = 0.561; phi2 = 16.6538*pi/180;
x3 = 1.056; y3 = 0.345; phi3 = 261.4161*pi/180;

%% Avaliação inicial
q0 = eval(sq);
PHI = eval(sPHI);
PHIq = eval(sPHIq);
dq = -PHIq\PHI;
q = q0 + dq;

%% Quantificando o erro quadrático da solução
erro_eq = norm(PHI);
erro_sol = sqrt(sum((q0-q).^2));

tt = 0:0.01:2*pi/w; % tempo em segundos mas parametrizado pelo número
                    % de voltas (2pi = 1 volta).

%% INICIO DO LOOP PRINCIPAL
for i=1:length(tt)
    t = tt(i);
    PHI(9,1) = q(3) - w*t; % Quando for alterar o vetor sPHI deve-se prestar atenção
                            % nesta linha para fazer a modificação apropriada
%% Loop do Newton-Raphson

```

```

while norm(PHI)>10^-15 , sqrt(sum((q0-q).^2)) > 10^-15;
    x1 = q(1);  y1 = q(2);  phi1 = q(3);
    x2 = q(4);  y2 = q(5);  phi2 = q(6);
    x3 = q(7);  y3 = q(8);  phi3 = q(9);

    PHI = eval(sPHI);
    PHIq = eval(sPHIq);
    dq = -PHIq\PHI;
    q0 = q;
    q = q0 + dq;
    erro_eq = norm(PHI);
    erro_sol = sqrt(sum((q0-q).^2));
end

%% Cálculo da velocidade
PHIt = eval(sPHIt);
ni = -PHIt;
qp = PHIq\ni;
x1p = qp(1);  y1p = qp(2);  phi1p = qp(3);
x2p = qp(4);  y2p = qp(5);  phi2p = qp(6);
x3p = qp(7);  y3p = qp(8);  phi3p = qp(9);

%% Cálculo da aceleração
gamma = eval(sgamma);
qpp = PHIq\gamma;

%% Cálculo dos vértices:
ptoA = eval(r1+A1*s1a);
ptoB = eval(r1+A1*s1b);
ptoC = eval(r2+A2*s2c);
ptoD = eval(r3+A3*s3d);

%% Cálculo da dinâmica inversa:
QA = eval(sQA);
M = eval(sM);
lambda = inv(PHIq')*(QA - M*qpp);
Fint = -PHIq'*lambda;

%% Armazenamento
%Armazenadores gerais
R.t(i) = t; R.q(:,i) = q;  R.qp(:,i) = qp; R.qpp(:,i) = qpp;
R.lambda(i,:) = lambda';  R.Fint(i,:) = Fint';

```

```

%Eq.3.1.4 – position level constraint:
R.erro_pos(i) = norm(eval(sPHI));
%Eq.3.1.9 – velocity level constraint:
R.erro_vel(i) = norm(eval(sPHIq)*qp + eval(sPHIt));
%Eq.3.1.10 – acceleration level constraint:
R.erro_acel(i) = norm(eval(sPHIq)*qpp – eval(sgamma));
%—>Posição das articulações
R.ptoA(i,:) = ptoA;    R.ptoB(i,:) = ptoB;
R.ptoC(i,:) = ptoC;    R.ptoD(i,:) = ptoD;
%—>Posição das coordenadas
R.x1(i,:) = q(1);      R.y1(i,:) = q(2);      R.phi1(i,:) = q(3);
R.x2(i,:) = q(4);      R.y2(i,:) = q(5);      R.phi2(i,:) = q(6);
R.x3(i,:) = q(7);      R.y3(i,:) = q(8);      R.phi3(i,:) = q(9);
%—>Velocidade das coordenadas
R.x1p(i,:) = qp(1);    R.y1p(i,:) = qp(2);    R.phi1p(i,:) = qp(3);
R.x2p(i,:) = qp(4);    R.y2p(i,:) = qp(5);    R.phi2p(i,:) = qp(6);
R.x3p(i,:) = qp(7);    R.y3p(i,:) = qp(8);    R.phi3p(i,:) = qp(9);
%—>Aceleração das coordenadas
R.x1pp(i,:) = qpp(1);  R.y1pp(i,:) = qpp(2);
R.phi1pp(i,:) = qpp(3);
R.x2pp(i,:) = qpp(4);  R.y2pp(i,:) = qpp(5);
R.phi2pp(i,:) = qpp(6);
R.x3pp(i,:) = qpp(7);  R.y3pp(i,:) = qpp(8);
R.phi3pp(i,:) = qpp(9);

%% Animação
figure(1)
clf
hold on
plot(ptoA(1),ptoA(2), 'kx')
plot(ptoB(1),ptoB(2), 'kx')
plot(ptoC(1),ptoC(2), 'kx')
plot(ptoD(1),ptoD(2), 'kx')
plot(x1,y1, 'g*')
plot(x2,y2, 'r*')
plot(x3,y3, 'b*')
hold off
axis('equal')
axis([-0.3 1.5 -0.5 1.3])
line([ptoA(1),ptoB(1)], [ptoA(2),ptoB(2)], 'Color', 'g')
line([ptoB(1),ptoC(1)], [ptoB(2),ptoC(2)], 'Color', 'r')
line([ptoC(1),ptoD(1)], [ptoC(2),ptoD(2)], 'Color', 'b')

```

```

line([ptoD(1),ptoA(1)],[ptoD(2),ptoA(2)], 'Color', 'k')
grid
title(['\phi_1: ', num2str(round(phi1*180/pi)), '°,
t: ', num2str(tt(i)), 's'])
% Textos:
text(x1 - L1/10*sin(phi1), y1 + L1/10*cos(phi1), '1')
text(x2 - L2/10*sin(phi2), y2 + L2/10*cos(phi2), '2')
text(x3 - L3/10*sin(phi3), y3 + L3/10*cos(phi3), '3')
text(ptoA(1) - 0.05, ptoA(2) - 0.05, 'A')
text(ptoB(1) - L1/10*sin(phi1), ptoB(2) + L1/10*cos(phi1), 'B')
text(ptoC(1) - L2/10*sin(phi2), ptoC(2) + L2/10*cos(phi2), 'C')
text(ptoD(1) + 0.05, ptoD(2) - 0.05, 'D')
filme(:, i) = getframe;

%% Verificação
phil_graus = q(3)*180/pi;
Comp1 = norm(ptoB - ptoA);
Comp2 = norm(ptoC - ptoB);
Comp3 = norm(ptoD - ptoC);
Comp4 = norm(ptoA - ptoD);
home
disp('Verificação: ')
phil_graus
end

save Haug4B_9coord_1c R

%% Avaliação dos erros e posição do centro de massa
figure(1)
subplot(2,1,1)
semilogy(R.t, R.erro_pos, '-.', R.t, R.erro_vel, '-.', R.t, R.erro_acel, '-.')
xlabel('t(s)')
ylabel('Violação das Restrições')
title({'Norma dos vetores q deveriam ser nulos'; ...
'caso as restrições fossem identicamente obedecidas'})
legend('Restr.Posição: PHI = 0', 'Restr.Velocidade: PHIq*qp + PHIt = 0', ...
'Restr.Aceleração: PHIq*qp - gamma = 0', 3)

subplot(2,1,2)
plot(R.x1, R.y1, R.x2, R.y2, R.x3, R.y3, R.ptoA(:,1), R.ptoA(:,2), '*', R.ptoB(:,1), R.ptoB(:,2), .,
R.ptoC(:,1), R.ptoC(:,2), R.ptoD(:,1), R.ptoD(:,2), '*')
legend('CG1', 'CG2', 'CG3', 'ponto A', 'ponto B', 'ponto C', 'ponto D', 4)

```

```

axis('equal')
xlabel('x')
ylabel('y')
title('Lugar geométrico dos CGs e dos pontos de interesse','fontsize',10)
grid

%% Cinemática das variáveis angulares
figure(2)
R.angulos = [R.phil R.phi2 R.phi3 R.philp R.phi2p R.phi3p R.philpp R.phi2pp R.phi3pp];
angulos = struct('ang',{'phi1 [°]','phi2 [°]','phi3 [°]',...
                 'philp [°/s]','phi2p [°/s]','phi3p [°/s]',...
                 'philpp [°/s^2]','phi2pp [°/s^2]','phi3pp [°/s^2]'});

for j=1:9
    subplot(3,3,j)
    plot(tt,R.angulos(:,j)*180/pi)
    xlabel('t[s]')
    ylabel(char(angulos(j).ang),'FontWeight','bold')
    grid
end

%% Reações internas do mecanismo
figure(3) %Lambdas
for j=1:9
    subplot(3,3,j)
    plot(tt,R.lambda(:,j) )
    xlabel('t[s]')
    ylabel(['\lambda_' num2str(j)],'FontWeight','bold','FontSize',14)
    grid
end

figure(4) % FC
for j=1:9
    subplot(3,3,j)
    plot(tt,R.Fint(:,j),'r')
    xlabel('t[s]')
    ylabel(['F^C_' num2str(j)],'FontWeight','bold','FontSize',14)
    grid
end

figure(5)
plot(R.t,-R.lambda(:,9))
xlabel('t[s]','FontWeight','bold')

```



```

ylabel('Torque [Nm]', 'FontWeight', 'bold')
legend('M_A = -\lambda_9', 2)
title({'Dinamica inversa 4 barras'; 'Torque na articulação A'}, 'FontWeight', 'bold')
grid

```

```

%% Cinemática do centro de massa da barra 1

```

```

figure(6)
subplot(3,2,1); plot(R.phil*180/pi,R.x1); xlabel('phil[deg]');
ylabel('x1'); grid;
title('Cinemática do centro de massa da barra 1')
subplot(3,2,2); plot(R.phil*180/pi,R.y1); xlabel('phil[deg]');
ylabel('y1'); grid;
subplot(3,2,3); plot(R.phil*180/pi,R.x1p); xlabel('phil[deg]');
ylabel('x1p'); grid;
subplot(3,2,4); plot(R.phil*180/pi,R.y1p); xlabel('phil[deg]');
ylabel('y1p'); grid;
subplot(3,2,5); plot(R.phil*180/pi,R.x1pp); xlabel('phil[deg]');
ylabel('x1pp'); grid;
subplot(3,2,6); plot(R.phil*180/pi,R.y1pp); xlabel('phil[deg]');
ylabel('y1pp'); grid;

```

```

%% Cinemática do centro de massa da barra 2

```

```

figure(7)
subplot(3,2,1); plot(R.phil*180/pi,R.x2); xlabel('phil[deg]');
ylabel('x2'); grid;
title('Cinemática do centro de massa da barra 2')
subplot(3,2,2); plot(R.phil*180/pi,R.y2); xlabel('phil[deg]');
ylabel('y2'); grid;
subplot(3,2,3); plot(R.phil*180/pi,R.x2p); xlabel('phil[deg]');
ylabel('x2p'); grid;
subplot(3,2,4); plot(R.phil*180/pi,R.y2p); xlabel('phil[deg]');
ylabel('y2p'); grid;
subplot(3,2,5); plot(R.phil*180/pi,R.x2pp); xlabel('phil[deg]');
ylabel('x2pp'); grid;
subplot(3,2,6); plot(R.phil*180/pi,R.y2pp); xlabel('phil[deg]');
ylabel('y2pp'); grid;

```

```

%% Cinemática do centro de massa da barra 3

```

```

figure(8)
subplot(3,2,1); plot(R.phil*180/pi,R.x3); xlabel('phil[deg]');
ylabel('x3'); grid;
title('Cinemática do centro de massa da barra 3')

```

```

subplot(3,2,2); plot(R.phil*180/pi,R.y3);   xlabel('phil[deg]');
ylabel('y3');   grid;
subplot(3,2,3); plot(R.phil*180/pi,R.x3p);  xlabel('phil[deg]');
ylabel('x3p');  grid;
subplot(3,2,4); plot(R.phil*180/pi,R.y3p);  xlabel('phil[deg]');
ylabel('y3p');  grid;
subplot(3,2,5); plot(R.phil*180/pi,R.x3pp); xlabel('phil[deg]');
ylabel('x3pp'); grid;
subplot(3,2,6); plot(R.phil*180/pi,R.y3pp); xlabel('phil[deg]');
ylabel('y3pp'); grid;

```

## C.2 DINÂMICA INVERSA COM REDUNDÂNCIA

Apresenta-se aqui o algoritmo para se resolver a dinâmica inversa do mecanismo 4 barras com a redundância de atuadores, tal como é abordado na Seção 2.4.2.

```

%% 4 BARRAS – DINÂMICA INVERSA COM 1, 2, 3 e 4 ATUADORES
% Exemplo adaptado do Haug, implementando a dinâmica inversa no mecanismo 4
% barras com 4 atuadores utilizando redundância de atuadores
%% DESCRIÇÃO: POSIÇÃO, VELOCIDADE, ACELERAÇÃO E CÁLCULO DOS LAMBDA
% Neste arquivo é resolvido o problema cinemático de posição, velocidade e
% aceleração do mecanismo 4 barras, com o algoritmo de Newton–Raphson. Ele
% se baseia no arquivo Haug4B_5a.m em que toda a parte da cinemática foi
% resolvida com sucesso e incorpora o conceito de se fazer a derivação das
% matrizes simbolicamente e depois avaliá-la numericamente com o comando
% "eval". Também incorpora a forma de definir o vetor de restrição PHI, tal
% como é apresentado no Haug.
% O que há de novo aqui, é a implementação da decomposição SVD para o
% cálculo dos lambdas e dos torques resultantes, que é mostrada no artigo:
% "Evaluation of dynamic capability of machines and robots, Michel Valasek,
% Zbijnek Zika – Multibody Systems Mech, 2001, Vol.6, pag. 183–202
% Neste artigo, a DAE é definida como:
%  $[M] \cdot \{qpp\} - [PHIq]' \cdot \{\lambda\} = \{G\} + [T] \cdot \{n\}$ , onde:
% [M] é a matriz de massa,
% {qpp} é o vetor de acelerações das coord. usadas para descrever o sistema,
% [PHIq] é o jacobiano das restrições cinemáticas
% {G} é o vetor das forças generalizadas que atuam sobre o sistema
%     (representa forças elásticas, de Coriolis, de amortecimento, etc)
% [T] é a Matriz de Transmissão. É a matriz que relaciona onde cada torque
%     está atuando em cada barra. Como o mecanismo 4 barras tem 4

```

```

%      articulações, e está sendo modelado, neste exemplo, com 9
%      coordenadas, a dimensão de [T] é 9x4.
% {n} é o vetor de torques (redundantes) que atuam sobre o sistema. Neste
%      problema {n} é uma incógnita e tem dimensão 4x1.
% {lambda} é o vetor de multiplicadores de Lagrange que também é uma
%      incógnita. Sua interpretação está relacionada às forças internas do
%      mecanismo
%%
clc
clear all
close all

%% CÁLCULO SIMBÓLICO:
%% Definição de variáveis
syms L1 L2 L3 L4
syms x1 x2 x3 x4 x1p x2p x3p
syms y1 y2 y3 y4 y1p y2p y3p
syms phi1 phi1_0 phi2 phi3 phi1p phi2p phi3p
syms w t g
syms m1 m2 m3 J1 J2 J3
syms s1a s1b s1c s2c s3c s3d
syms ptoA ptoD

%% Matrizes de rotação
A1 = [cos(phi1) -sin(phi1); sin(phi1) cos(phi1)];
A2 = [cos(phi2) -sin(phi2); sin(phi2) cos(phi2)];
A3 = [cos(phi3) -sin(phi3); sin(phi3) cos(phi3)];
%% Vetor posição dos centros de massa:
r1 = [x1;y1];
r2 = [x2;y2];
r3 = [x3;y3];
%% Vetor posição dos pontos de interesse (articulações) em relação aos sist.
% de coordenadas locais das barras
s1a = [-L1/2;0]; s1b = [L1/2;0];
s2b = [-L2/2;0]; s2c = [L2/2;0];
s3c = [-L3/2;0]; s3d = [L3/2;0];
%% Vetor de posição absoluta dos pontos fixos A e D:
ptoA = [0.2*L4; 0.2*L4];
ptoD = ptoA + [L4;0];
%% Vetor de restrições cinemáticas:
sPHI = [r1 + A1*s1a - ptoA
        r1 + A1*s1b - (r2 + A2*s2b)

```

```

    r2 + A2*s2c - (r3 + A3*s3c)
    r3 + A3*s3d - ptoD
    phil - phil_0 - w*t]
%% Vetor de coordenadas generalizadas e de velocidades
sq = [x1; y1; phil; x2; y2; phi2; x3; y3; phi3]
sqp = [x1p; y1p; philp; x2p; y2p; phi2p; x3p; y3p; phi3p]
sPHIq = jacobian(sPHI,sq)
sPHIt = diff(sPHI,t)
sPHItt = diff(sPHIt,t)
sPHIqt = diff(sPHIq,t)
sgamma = -jacobian(sPHIq*sqp,sq)*sqp - 2*sPHIqt*sqp - sPHItt
alfa = 0; beta = 2*alfa;
sgammaB = sgamma - 2*alfa*(sPHIq*sqp + sPHIt) - beta^2*sPHI;
sM = diag([m1,m1,J1,m2,m2,J2,m3,m3,J3])
sQA = [0; -m1*g; 0; 0; -m2*g; 0; 0; -m3*g; 0]
% Matriz de transformação para 1 atuador
T1 = zeros(9,4);
% Matriz de transformação para 2 atuadores
T2 = zeros(9,4);
T2(3,1) = 0; T2(3,2) = -1;
T2(6,2) = 1;
% Matriz de transformação para 3 atuadores
T3 = zeros(9,4);
T3(3,1) = 0; T3(3,2) = -1;
T3(6,2) = 1; T3(6,3) = -1;
T3(9,3) = 1;
% Matriz de transformação para 4 atuadores
T4 = zeros(9,4);
T4(3,1) = 0; T4(3,2) = -1;
T4(6,2) = 1; T4(6,3) = -1;
T4(9,3) = 1; T4(9,4) = 1;

%% INICIALIZAÇÃO DOS PARÂMETROS NUMÉRICOS:
%% Valores constantes:
L1 = 0.5; %[m] Comprimento da barra 1
L2 = 0.9; %[m] Comprimento da barra 2
L3 = 0.7; %[m] Comprimento da barra 3
L4 = 1.0; %[m] Comprimento da barra 4
w = 2*pi; %rad/s Velocidade angular (constante) da barra 1
g = 9.81; %m/s^2
t = 0; %segundos
%% Valores iniciais estimados a partir do desenho do Inventor para phil = 60°

```

```

x1 = 0.125;          y1 = 0.216;          phi1 = 0;          phi1_0 = 60*pi/180;
x2 = 0.681;          y2 = 0.561;          phi2 = 16.6538*pi/180;
x3 = 1.056;          y3 = 0.345;          phi3 = 261.4161*pi/180;
m1 = 6.590;          m2 = 11.550;          m3 = 9.070;
J1 = (1/12)*m1*L1^2; J2 = (1/12)*m2*L2^2; J3 = (1/12)*m3*L3^2;

%% Avaliação inicial
q0 = eval(sq);
PHI = eval(sPHI);
PHIq = eval(sPHIq);
dq = -PHIq\PHI;
q = q0 + dq;
%% Quantificando o erro quadrático da solução
erro_eq = norm(PHI);
erro_sol = sqrt(sum((q0-q).^2));
tt = 0:0.01:2*pi/w;
%% INICIO DO LOOP PRINCIPAL
for i=1:length(tt)
    t = tt(i);
    PHI(9,1) = q(3) - w*t;
%% Loop do Newton-Raphson
    while norm(PHI)>10^-15 , sqrt(sum((q0-q).^2)) > 10^-15;
        x1 = q(1); y1 = q(2); phi1 = q(3);
        x2 = q(4); y2 = q(5); phi2 = q(6);
        x3 = q(7); y3 = q(8); phi3 = q(9);
        PHI = eval(sPHI);
        PHIq = eval(sPHIq);
        dq = -PHIq\PHI;
        q0 = q;
        q = q0 + dq;
        erro_eq = norm(PHI);
        erro_sol = sqrt(sum((q0-q).^2));
    end

%% Cálculo da velocidade
    PHIt = eval(sPHIt);
    ni = -PHIt;
    qp = PHIq\ni;
    x1p = qp(1); y1p = qp(2); phi1p = qp(3);
    x2p = qp(4); y2p = qp(5); phi2p = qp(6);
    x3p = qp(7); y3p = qp(8); phi3p = qp(9);

```

```

%% Cálculo da aceleração
gamma = eval(sgammaB);
qpp = PHIq\gamma;

%% Cálculo dos vértices:
ptoA = eval(r1+A1*s1a);
ptoB = eval(r1+A1*s1b);
ptoC = eval(r2+A2*s2c);
ptoD = eval(r3+A3*s3d);

%% Cálculo da dinâmica inversa:
Q = eval(sQA);      M = eval(sM);      G = Q - M*qpp;
P1 = [PHIq' -T1];  % Para o cálculo do caso com 1 atuador
P2 = [PHIq' -T2];  % Para o cálculo do caso com 2 atuadores
P3 = [PHIq' -T3];  % Para o cálculo do caso com 3 atuadores
P4 = [PHIq' -T4];  % Para o cálculo do caso com 4 atuadores
X1 = P1'*inv(P1*P1')*G; % No Matlab também pode ser calculada com o
X2 = P2'*inv(P2*P2')*G; % comando "pinv" -> pinv(P) = P'*inv(P*P')
X3 = P3'*inv(P3*P3')*G;
X4 = P4'*inv(P4*P4')*G;

%% Armazenamento
R.t(i) = t; R.q(:,i) = q;  R.qp(:,i) = qp; R.qpp(:,i) = qpp;
R.erro_pos(i) = norm(eval(sPHI));
R.erro_vel(i) = norm(eval(sPHIq)*qp + eval(sPHIt));
R.erro_acel(i) = norm(eval(sPHIq)*qpp - eval(sgamma));
R.X1(i,:) = X1; %Obs: Esses vetores "X" armazenam os valores dos
R.X2(i,:) = X2; %multiplicadores de Lagrange em seus primeiros elementos
R.X3(i,:) = X3; %e os valores dos torques resultantes da din. inv. nos
R.X4(i,:) = X4; %seus últimos elementos.
R.x1(i,:) = q(1);%-->Posição
R.y1(i,:) = q(2);
R.phi1(i,:) = q(3);
R.x2(i,:) = q(4);
R.y2(i,:) = q(5);
R.phi2(i,:) = q(6);
R.x3(i,:) = q(7);
R.y3(i,:) = q(8);
R.phi3(i,:) = q(9);
R.ptoA(i,:) = ptoA;
R.ptoB(i,:) = ptoB;
R.ptoC(i,:) = ptoC;

```

```

R.ptoD(i,:) = ptoD;%-->Posição
R.x1p(i,:) = qp(1);%-->Velocidade
R.y1p(i,:) = qp(2);
R.philp(i,:) = qp(3);
R.x2p(i,:) = qp(4);
R.y2p(i,:) = qp(5);
R.phi2p(i,:) = qp(6);
R.x3p(i,:) = qp(7);
R.y3p(i,:) = qp(8);
R.phi3p(i,:) = qp(9);%-->Velocidade
R.x1pp(i,:) = qpp(1);%-->Aceleração
R.y1pp(i,:) = qpp(2);
R.philpp(i,:) = qpp(3);
R.x2pp(i,:) = qpp(4);
R.y2pp(i,:) = qpp(5);
R.phi2pp(i,:) = qpp(6);
R.x3pp(i,:) = qpp(7);
R.y3pp(i,:) = qpp(8);
R.phi3pp(i,:) = qpp(9);%-->Aceleração
%% Verificação
    phil_graus = q(3)*180/pi;
    home
    disp('Verificação: ')
    phil_graus
end
figure(301)
plot(R.t,-R.X1(:,9),'.-')
legend('Torque A = -\lambda_9',2)
title('CASO: 1 atuador - articulação A','FontWeight','bold')
xlabel('t[s]')
ylabel('Torques [N]')
grid
figure(302)
plot(R.t,-R.X2(:,9),'.-',R.t,R.X2(:,11),'o-')
legend('Torque A = -\lambda_9','Torque B',2)
title('CASO: 2 atuadores - articulações A e B','FontWeight','bold')
xlabel('t[s]')
ylabel('Torques [N]')
grid
figure(303)
plot(R.t,-R.X3(:,9),'.-',R.t,R.X3(:,11),'o-',R.t,R.X3(:,12),'*-')
legend('Torque A = -\lambda_9','Torque B','Torque C',2)

```

```

title('CASO: 3 atuadores – articulações A, B e C','FontWeight','bold')
xlabel('t[s]')
ylabel('Torques [N]')
grid
figure(304)
plot(R.t,-R.X4(:,9),'.-',R.t,R.X4(:,11),'o-',R.t,R.X4(:,12),'*-',R.t,R.X4(:,13),'+-')
legend('Torque A = -\lambda_9','Torque B','Torque C','Torque D',2)
title('CASO: 4 atuadores – articulações A, B, C e D','FontWeight','bold')
xlabel('t[s]')
ylabel('Torques [N]')
grid

```

### C.3 DINÂMICA DIRETA

Apresenta-se aqui o algoritmo para se resolver a dinâmica direta do mecanismo 4 barras, tal como é abordado na Seção 2.3.3.

#### C.3.1 PROGRAMA PRINCIPAL

O início do programa principal para a dinâmica direta do mecanismo 4 barras é similar ao algoritmo apresentado para a dinâmica inversa. No entanto, há uma diferença fundamental: o vetor PHI não possui restrição diretora. Isto quer dizer que não há nada guiando o movimento de nenhuma das variáveis. Portanto, ao integrar as equações de movimento definidas aqui, mediante uma condição inicial, o 4 barras irá exibir um comportamento pendular.

```

%% 4 BARRAS REGULAR – (CRANK COM VELOCIDADE CONSTANTE) – Dinâmica Direta/ODE45
% Olavo Luppi Silva – junho de 2007

%% DESCRIÇÃO: (DINÂMICA DIRETA DO 4 BARRAS)
% Neste arquivo é resolvido o problema da dinâmica direta de um mecanismo 4
% barras submetido ao torque calculado pela dinamica inversa. Se o
% resultado da dinâmica inversa estiver correto, a cinemática das
% coordenadas cartesianas calculadas aqui, deve-se aproximar da cinemática
% calculada na dinâmica inversa.
%
% O algoritmo apresentado neste arquivo compreende as seguintes etapas:
% 1. Derivação simbólica (OBS: o vetor PHI não contém a expressão do
%     driver cinemático, já que a condição de velocidade philp = constante

```



```

%      é imposta pelo torque calculado na dinamica inversa.
%  2. Inicialização de parâmetros numéricos (geométricos e de inércia)
%  3. Carregamento das condições iniciais vindas da dinâmica inversa, bem
%      como da curva de torque na articulação A.
%  4. Integração
%  5. Pós processamento (cálculo das acelerações) e armazenamento dos
%      resultados
%  6. Visualização
clear all
close all
global sPHI sPHIq sPHIt sgammaB sM sQA g w phi1_0
global L1 L2 L3 L4
global m1 m2 m3 J1 J2 J3
global R

%% 1.DERIVAÇÃO SIMBÓLICA:
syms L1 L2 L3 L4
syms x1 x2 x3 x1p x2p x3p
syms y1 y2 y3 y1p y2p y3p
syms phi1 phi1_0 phi2 phi3 phi1p phi2p phi3p
syms w t g
syms m1 m2 m3 J1 J2 J3
syms s1a s1b s1c s2c s3c s3d s4d s4a
syms ptoA ptoD

% Matrizes de rotação
A1 = [cos(phi1) -sin(phi1); sin(phi1) cos(phi1)];
A2 = [cos(phi2) -sin(phi2); sin(phi2) cos(phi2)];
A3 = [cos(phi3) -sin(phi3); sin(phi3) cos(phi3)];

% Vetor posição dos centros de massa:
r1 = [x1;y1];
r2 = [x2;y2];
r3 = [x3;y3];

% Vetor posição dos pontos de interesse (articulações) em relação aos sist.
% de coordenadas locais das barras
s1a = [-L1/2;0]; s1b = [L1/2;0];
s2b = [-L2/2;0]; s2c = [L2/2;0];
s3c = [-L3/2;0]; s3d = [L3/2;0];
s4d = [-L4/2;0]; s4a = [L4/2;0];

```

```

% Vetor de posição absoluta dos pontos fixos A e D:
ptoA = [0.2*L4; 0.2*L4];
ptoD = ptoA + [L4;0];

% Vetor de restrições cinemáticas:
sPHI = [r1 + A1*s1a - ptoA           %Pto A é fixo
        r1 + A1*s1b - (r2 + A2*s2b)%Soma vet. do CG1 até ptoB = CG2 até pto B
        r2 + A2*s2c - (r3 + A3*s3c)%Soma vet. do CG2 até ptoC = CG3 até pto C
        r3 + A3*s3d - ptoD]         %Pto D é fixo

% Vetor de coordenadas generalizadas e de velocidades
sq = [x1; y1; phi1; x2; y2; phi2; x3; y3; phi3];
sqp = [x1p; y1p; phi1p; x2p; y2p; phi2p; x3p; y3p; phi3p];
sPHIq = jacobian(sPHI,sq);
sPHIt = diff(sPHI,t);
sPHItt = diff(sPHIt,t);
sPHIqt = diff(sPHIq,t);
sgamma = -jacobian(sPHIq*sqp,sq)*sqp - 2*sPHIqt*sqp - sPHItt;
alfa = 1; beta = 10*alfa; % Parâmetros para a estabilização de Baumgarte.
sgammaB = sgamma - 2*alfa*(sPHIq*sqp + sPHIt) - beta^2*sPHI;% gamma c/ estab de Baumgart
sM = diag([m1,m1,J1,m2,m2,J2,m3,m3,J3]); % Matriz de massa
sQA = [0; -m1*g; 0; 0; -m2*g; 0; 0; -m3*g; 0];% Esse vetor segue a ordem do
                                           % vetor de coord. generalizadas

%% 2. INICIALIZAÇÃO DE PARÂMETROS NUMÉRICOS:
% Valores constantes:
L1 = 0.5;   L2 = 0.9;   L3 = 0.7;   L4 = 1.0; %[m]
w = 2*pi;  %rad/s
g = 9.81; %m/s^2
t = 0;    %segundos
% Parâmetros de inércia
m1 = 6.590; m2 = 11.550;   m3 = 9.070; % [Kg]
J1 = (1/12)*m1*L1^2;   J2 = (1/12)*m2*L2^2;   J3 = (1/12)*m3*L3^2;% [Kg*m^2]

%% 3. DADOS DA ANÁLISE CINEMÁTICA / DINÂMICA INVERSA:
load Haug4B_9coord_1c R
N = length(R.x1);
int_tempo = [0 R.t(N)]; % Tempo da integração
q_ini = [R.x1(1) R.y1(1) R.phi1(1) ...
         R.x2(1) R.y2(1) R.phi2(1) ...
         R.x3(1) R.y3(1) R.phi3(1)]; % Condição inicial de posição
qp_ini = [ R.x1p(1) R.y1p(1) R.phi1p(1) ...

```

```

        R.x2p(1) R.y2p(1) R.phi2p(1) ...
        R.x3p(1) R.y3p(1) R.phi3p(1)]; % Condição inicial de velocidade

%% 4. INTEGRAÇÃO
% Condições iniciais:
cond_ini = [q_ini,qp_ini];
% Opções de integração do comando ode45:
% 'RelTol = tol_relativa e AbsTol = tol_absoluta
tol_relativa = 1e-8;
tol_absoluta = 1e-9;
options = odeset('RelTol',tol_relativa,...
                'AbsTol',[tol_absoluta*ones(1,length(q_ini))...
                tol_absoluta*ones(1,length(qp_ini))]); %
% Rotina de integração:
tic
[ttdir,y] = ode45(@quatrobarras_forcado_3b,int_tempo,cond_ini,options);
toc
disp('Dinâmica Direta calculada com sucesso')

%% 5. PÓS PROCESSAMENTO E ARMAZENAMENTO DE RESULTADOS
disp('Preparando para armazenamento e animação')
for i=1:length(y)
    yp(i,:) = quatrobarras_forcado_3b(ttdir(i),y(i,:))'; % derivada do vetor de estado:
                                                    % velocidades e acelerações

    % Atualização do vetor posição
    x1 = y(i,1); y1 = y(i,2); phi1 = y(i,3);
    x2 = y(i,4); y2 = y(i,5); phi2 = y(i,6);
    x3 = y(i,7); y3 = y(i,8); phi3 = y(i,9);
    q = [x1; y1; phi1; x2; y2; phi2; x3; y3; phi3];
    % Atualização do vetor velocidade
    x1p = y(i,10); y1p = y(i,11); phi1p = y(i,12);
    x2p = y(i,13); y2p = y(i,14); phi2p = y(i,15);
    x3p = y(i,16); y3p = y(i,17); phi3p = y(i,18);
    qp = [x1p; y1p; phi1p; x2p; y2p; phi2p; x3p; y3p; phi3p];
    % Atualização do vetor aceleração
    x1pp = yp(i,10); y1pp = yp(i,11); phi1pp = yp(i,12);
    x2pp = yp(i,13); y2pp = yp(i,14); phi2pp = yp(i,15);
    x3pp = yp(i,16); y3pp = yp(i,17); phi3pp = yp(i,18);
    qpp = [x1pp; y1pp; phi1pp; x2pp; y2pp; phi2pp; x3pp; y3pp; phi3pp];
    % Violação das restrições na solução da Din. Direta
    RD.erro_pos(i) = norm(eval(sPHI));
% position constraint

```

```

        RD.erro_vel(i) = norm(eval(sPHIq)*qp + eval(sPHIt));
% velocity constraint
        RD.erro_acel(i) = norm(eval(sPHIq)*qpp - eval(sgammaB));% acceleration constraint

end

% Armazenamento dos resultados individuais do resultado da Din. Direta na
% structure "RD" para facilitar a plotagem.
RD.t = ttdir;
RD.x1 = y(:,1); RD.y1 = y(:,2); RD.phi1 = y(:,3);
RD.x2 = y(:,4); RD.y2 = y(:,5); RD.phi2 = y(:,6);
RD.x3 = y(:,7); RD.y3 = y(:,8); RD.phi3 = y(:,9);
RD.q = y(:,1:9)';
RD.x1p = y(:,10); RD.y1p = y(:,11); RD.phi1p = y(:,12);
RD.x2p = y(:,13); RD.y2p = y(:,14); RD.phi2p = y(:,15);
RD.x3p = y(:,16); RD.y3p = y(:,17); RD.phi3p = y(:,18);
RD.qp = y(:,10:18)';
RD.x1pp = yp(:,10); RD.y1pp = yp(:,11); RD.phi1pp = yp(:,12);
RD.x2pp = yp(:,13); RD.y2pp = yp(:,14); RD.phi2pp = yp(:,15);
RD.x3pp = yp(:,16); RD.y3pp = yp(:,17); RD.phi3pp = yp(:,18);
RD.qpp = yp(:,10:18)';

%% 6.VISUALIZAÇÃO
figure(1)
plot( R.x1,R.y1,'b.',RD.x1,RD.y1,'b',...
      R.x2,R.y2,'g.',RD.x2,RD.y2,'g',...
      R.x3,R.y3,'r.',RD.x3,RD.y3,'r',...
      R.x1(1),R.y1(1),'b*',R.x2(1),R.y2(1),'g*',R.x3(1),R.y3(1),'r*')
legend('CG1 - Din.Inv', 'CG1 - Din.Dir',...
      'CG2 - Din.Inv', 'CG2 - Din.Dir',...
      'CG3 - Din.Inv', 'CG3 - Din.Dir',4)
title('Lugar geométrico dos CGs e dos pontos de interesse')
axis('equal')
xlabel('x')
ylabel('y')
grid

figure(2)
subplot(1,3,1)
semilogy(R.t,R.erro_pos,'-*',RD.t,RD.erro_pos)
xlabel('t(s)', 'FontWeight', 'bold')
ylabel('Erro Posição', 'FontWeight', 'bold')
title({'Violação da Restr. de Posição:', ' ||PHI|| = 0'}, 'FontWeight', 'bold')

```

```

legend('Dinâmica Inversa ', 'Dinâmica Direta', 4)
grid

subplot(1,3,2)
semilogy(R.t,R.erro_vel, '-*', RD.t, RD.erro_vel)
xlabel('t(s)', 'FontWeight', 'bold')
ylabel('Erro Velocidade', 'FontWeight', 'bold')
title({'Violação da Restr. de Velocidade:', ' ||PHIq*qp + PHIt|| = 0'}, ...
      'FontWeight', 'bold')
legend('Dinâmica Inversa ', 'Dinâmica Direta', 4)
grid

subplot(1,3,3)
semilogy(R.t,R.erro_acel, '-*', RD.t, RD.erro_acel)
xlabel('t(s)', 'FontWeight', 'bold')
ylabel('Erro Aceleração', 'FontWeight', 'bold')
title({'Violação da Restr. de Aceleração:', ' ||PHIq*qp - \gamma|| = 0'}, ...
      'FontWeight', 'bold')
legend('Dinâmica Inversa ', 'Dinâmica Direta', 4)
grid

%% Cinemática das variáveis angulares (dinâmica inversa)
figure(3)
angulos = struct('ang', {'phi1 [°]', 'phi2 [°]', 'phi3 [°]', ...
                       'philp [°/s]', 'phi2p [°/s]', 'phi3p [°/s]', ...
                       'philpp [°/s^2]', 'phi2pp [°/s^2]', 'phi3pp [°/s^2]'});
R.angulos = [R.phi1 R.phi2 R.phi3 R.philp R.phi2p R.phi3p R.philpp R.phi2pp R.phi3pp];
RD.angulos = [RD.phi1 RD.phi2 RD.phi3 RD.philp RD.phi2p RD.phi3p RD.philpp RD.phi2pp RD.
for j=1:9
    subplot(3,3,j)
    plot(R.t,R.angulos(:,j)*180/pi, '.', RD.t, RD.angulos(:,j)*180/pi)
    xlabel('t[s]')
    ylabel(char(angulos(j).ang), 'FontWeight', 'bold')
    grid
    legend('Din Inv.', 'Din Dir.', 2)
end

figure(5)
subplot(2,1,1)
plot(R.t,R.angulos(:,4)*180/pi, '.', RD.t, RD.angulos(:,4)*180/pi)
xlabel('t[s]')
ylabel(char(angulos(4).ang), 'FontWeight', 'bold')

```

```

    grid
    legend('Din Inv.', 'Din Dir.',2)
subplot(2,1,2)
    plot(R.t,R.angulos(:,7)*180/pi, '.',RD.t,RD.angulos(:,7)*180/pi)
    xlabel('t[s]')
    ylabel(char(angulos(7).ang), 'FontWeight', 'bold')
    grid
    legend('Din Inv.', 'Din Dir.',2)

figure(4)
coordenadas = struct('coord',...
{'x1 [m]', 'y1 [m]', 'x2 [m]', 'y2 [m]', 'x3 [m]', 'y3 [m]',...
'x1p [m/s]', 'y1p [m/s]', 'x2p [m/s]', 'y2p [m/s]', 'x3p [m/s]', 'y3p [m/s]',...
'x1pp [m/s^2]', 'y1pp [m/s^2]', 'x2pp [m/s^2]', 'y2pp [m/s^2]',...
'x3pp [m/s^2]', 'y3pp [m/s^2]'});
R.coor_cart = [R.x1 R.y1 R.x2 R.y2 R.x3 R.y3...
R.x1p R.y1p R.x2p R.y2p R.x3p R.y3p...
R.x1pp R.y1pp R.x2pp R.y2pp R.x3pp R.y3pp];
RD.coor_cart = [RD.x1 RD.y1 RD.x2 RD.y2 RD.x3 RD.y3...
RD.x1p RD.y1p RD.x2p RD.y2p RD.x3p RD.y3p...
RD.x1pp RD.y1pp RD.x2pp RD.y2pp RD.x3pp RD.y3pp];
for j=1:6*3
    subplot(3,6,j)
    plot(R.t,R.coor_cart(:,j), '.',RD.t,RD.coor_cart(:,j), 'LineWidth',2)
    title([char(coordenadas(j).coord)], 'FontWeight', 'bold')
    ylabel('[m]')
    xlabel('t[s]')
    if j==6*3
        legend('Din Inv.', 'Din Dir.',3)
    end
    grid
end
end

```

### C.3.2 FUNCTION DE INTEGRAÇÃO - SEM REDUNDÂNCIA

Esta é a *function* que é chamada pelo ODE45 durante o processo de integração, cuja função é retornar a derivada do vetor de estados. Esta *function* foi utilizada para verificar se o resultado obtido com a dinâmica inversa para o 4 barras sem redundância de atuadores era reproduzido com a dinâmica direta.

```

%=====
% Função que calcula a derivada do vetor de estados
%=====
function yp = quatrobarras_forcado_3b(t,q)
% DESCRIÇÃO DAS VARIÁVEIS:
% t: tempo
% q: variáveis de estado
% N: número de elementos das variáveis de estado
% q(1:N/2) -> posições
% q(N/2+1:N) -> velocidades
% x(1:N/2) -> acelerações
% x(N/2+1:N) -> lambdas
global sPHI sPHIq sPHIt sgammaB sM sQA g w phi1_0
global L1 L2 L3 L4
global m1 m2 m3 J1 J2 J3
global R

[M,N] = size(sPHIq); % Se q = 9 coord e ã há drive cinemático => M = 8, N = 9
x1 = q(1); y1 = q(2); phi1 = q(3);
x2 = q(4); y2 = q(5); phi2 = q(6);
x3 = q(7); y3 = q(8); phi3 = q(9);
x1p = q(10); y1p = q(11); phi1p = q(12);
x2p = q(13); y2p = q(14); phi2p = q(15);
x3p = q(16); y3p = q(17); phi3p = q(18);

% Modificação no vetor de forças externas com o torque calculado através da
% dinâmica inversa.
QA = eval(sQA);
QA(3) = interp1(R.t,-R.lambda(:,9),t);

% Matrizes do sistema:
A = [eval(sM) eval(sPHIq)'
     eval(sPHIq) zeros(M,M)];
b = [QA; eval(sgammaB)];

x = A\b;
yp(1:N) = q(N+1:2*N);
yp(N+1:2*N) = x(1:N);
yp = yp';

```

### C.3.3 FUNCTION DE INTEGRAÇÃO - COM REDUNDÂNCIA

Esta *function* foi utilizada para verificar se o resultado obtido com a dinâmica inversa para o 4 barras com redundância de atuadores era reproduzido com a dinâmica direta. O exemplo que é mostrado aqui considera o caso com 4 atuadores.

```
%=====
function yp = quatrobarras_forcado_2act_1a(t,q)

% DESCRIÇÃO DAS VARIÁVEIS:
% t: tempo
% q: variáveis de estado
% N: número de elementos das variáveis de estado
% q(1:N/2) -> posições
% q(N/2+1:N) -> velocidades
% x(1:N/2) -> acelerações
% x(N/2+1:N) -> lambdas

global sPHI sPHIq sPHIt sgammaB sM sQA g w phi1_0
global L1 L2 L3 L4
global m1 m2 m3 J1 J2 J3
global R T1 T2 T3 T4

[M,N] = size(sPHIq); % Se q = 9 coord e ã há drive cinemático => M = 8, N = 9
x1 = q(1); y1 = q(2); phi1 = q(3);
x2 = q(4); y2 = q(5); phi2 = q(6);
x3 = q(7); y3 = q(8); phi3 = q(9);
x1p = q(10); y1p = q(11); phi1p = q(12);
x2p = q(13); y2p = q(14); phi2p = q(15);
x3p = q(16); y3p = q(17); phi3p = q(18);

% Modificação no vetor de forças externas com o torque calculado através da
% dinâmica inversa.
QA = eval(sQA);
QA(3) = interp1(R.t,R.CAS04act_artA,t);
n = zeros(4,1);
n(2) = interp1(R.t,R.CAS04act_artB,t);
n(3) = interp1(R.t,R.CAS04act_artC,t);
n(4) = interp1(R.t,R.CAS04act_artD,t);

QA = QA + T4*n;
```



```

% Matrizes do sistema:
A = [eval(sM) eval(sPHIq)'
      eval(sPHIq) zeros(M,M)];
b = [QA; eval(sgammaB)];

x = A\b;
yp(1:N) = q(N+1:2*N);
yp(N+1:2*N) = x(1:N);
yp = yp';

```

## C.4 ALGORITMOS DO OCP

### C.4.1 ALGORITMO PRINCIPAL

Este é o algoritmo principal para a solução do problema de controle ótimo utilizando o RIOTS. O exemplo mostrado aqui se refere ao CASO 7 para o pêndulo triplo com 3 atuadores mais a mola e amortecedor nas extremidades.

```

%% CONTROLE ÓTIMO – MECANISMO 4 BARRAS
%% MODELO: 3 atuadores com mola e amortecedor na extremidade
% INSTITUTO MILITAR DE ENGENHARIA – IME
% MSc. Olavo Luppi Silva

%% DESCRIÇÃO:
% O objetivo desse arquivo é:
% – fazer a derivação das equações de movimento do sistema,
% – fazer a derivação dos jacobianos que devem ser fornecidos ao Riots,
% – criar o arquivo e .C eo .DLL que devem ser fornecidos ao Riots,
% – resolver o problema de controle ótimo
clc
clear global
clear all
global L1 L2 L3 L4 I1 I2 I3 L1cm L2cm L3cm m1 m2 m3 g
global xdot h_x h_u
syms xdot h_x h_u
syms x1 x2 x3 x4 x5 x6 U1 U2 U3 real
syms L1 L2 L3 L4 m1 m2 m3 g I1 I2 I3
syms phi1 phi2 phi3 dphi1 dphi2 dphi3 d2phi1 d2phi2 d2phi3

```

```

%% 1ª PARTE: DERIVAÇÃO DA EQUAÇÃO DE MOVIMENTO
% Parâmetros do sistema
m1 = 6.590;      % Massa da barra1 (crank)
m2 = 11.550;    % Massa da barra2 (coupler)
m3 = 9.070;;    % Massa da barra3 (follower)
L1 = 0.5;       % Comprimento da barra1 (crank)
L2 = 0.9;       % Comprimento da barra2 (coupler)
L3 = 0.7;       % Comprimento do barra3 (follower)
L4 = 1.0;       % Comprimento da barra4 (fixa)
L1cm = L1/2;    % Dist. da art. A ao CM da barra1
L2cm = L2/2;    % Dist. da art. B ao CM da barra2
L3cm = L3/2;    % Dist. da art. C ao CM da barra3
I1 = (1/12)*m1*L1^2; % Momento de inércia da barra1 (crank)
I2 = (1/12)*m2*L2^2; % Momento de inércia da barra2 (coupler)
I3 = (1/12)*m3*L3^2; % Momento de inércia do barra3 (follower)
g = 9.81;       % Constante gravitacional
pr = 16 % Precisão do truncamento numérico na geração das expr. algébricas

% Forças externas (controles e molas)
F = [1 -1 0 % Essa matriz "D" (de controles)
      0 1 -1 % Obs: termos negativos correspondem às reações dos torques na barra.
      0 0 1]; % Ex: M(1,2) = -1, pq é a reação de U2 sobre a barra 1.

u = [U1 % Este vetor "u" serve para o caso de 3 atuadores
      U2
      U3];

K = 1e5; % Rigidez da mola (admite-se a mesma rigidez para as 2 molas)
C = 1e3; % Coeficiente de amortecimento (mesmo coef. para os 2 amortecedores)

% Matriz correspondente ao vetor de forças generalizadas devido à presença da mola e do
G = [-L1*(C*L1*dphi1 + C*L2*dphi2*cos(phi1 - phi2) + C*L3*dphi3*cos(phi1 - phi3)...
      + K*L4*sin(phi1) - K*(L2*sin(phi1 - phi2) + L3*sin(phi1 - phi3)))
      -L2*(C*L2*dphi2 + C*L1*dphi1*cos(phi1 - phi2) + C*L3*dphi3*cos(phi2 - phi3)...
      + K*L1*sin(phi1 - phi2) + K*L4*sin(phi2) - K*L3*sin(phi2 - phi3))
      -L3*(C*L3*dphi3 + C*L1*dphi1*cos(phi1 - phi3) + C*L2*dphi2*cos(phi2 - phi3)...
      + K*L1*sin(phi1 - phi3) + K*L2*sin(phi2 - phi3) + K*L4*sin(phi3))];

% Equação de estado
xdot = eom2ss_1c(F,G,u);
xdot = simple(xdot);

```

```

% Restrição de desigualdade de estado:
l = [(L1*cos(x1) + L2*cos(x2) + L3*cos(x3) - L4)^2
      (L1*sin(x1) + L2*sin(x2) + L3*sin(x3))^2
      (-L1*x4*cos(x1) - L2*x5*cos(x2) - L3*x6*cos(x3))^2
      (-L1*x4*sin(x1) - L2*x5*sin(x2) - L3*x6*sin(x3))^2
      (x4 - 2*pi)^2];
l = l(1) + l(2) + l(3) + l(4) + l(5);
% OBS: A eq acima é uma forma de deixar o problema com apenas 1 restrição
% de desigualdade nas vars de estado para ficar como o problema exemplo
% "Parabola.c".

% Cálculo dos jacobianos da equação de movimento (EDM)
for i=1:length(xdot),
    for j=1:length(xdot), % Jacobiano da EDM em relação a "x"
        h_x(i,j) = vpa(diff(xdot(i),strcat('x',num2str(j))),pr);
    end
    for j=1:length(u), % Jacobiano da EDM em relação a "u"
        h_u(i,j)=vpa(diff(xdot(i),strcat('U',num2str(j))),pr);
    end
end

% Cálculo dos jacobianos da restrição das equações de restrição de
% desigualdade de estado
for i=1:length(l),
    for j=1:length(xdot), % Jacobiano da restrição em relação a "x"
        l_x(i,j) = vpa(diff(l(i),strcat('x',num2str(j))),pr);
    end

    for j=1:length(u), % Jacobiano da restrição em relação a "u"
        l_u(i,j)=vpa(diff(l(i),strcat('U',num2str(j))),pr);
    end
end

%% GERAÇÃO DO CÓDIGO EM C.
exportar_c(xdot, 'xdot', 'xdot.c')
exportar_c(l, 'l', 'l.c')
exportar_c(l_x, 'l_x', 'l_x.c')
exportar_c(l_u, 'l_u', 'l_u.c')
exportar_c(h_x, 'h_x', 'h_x.c')
exportar_c(h_u, 'h_u', 'h_u.c')

```

```

% Cria variável 'nome' com o nome deste arquivo
nome = mfilename % Comando útil:"fullfile": Build a full filename from parts
disp('Arquivos .c foram gerados com sucesso!')
disp(['Monte o arquivo principal .c com o nome: ' nome ' '])
disp('Aperte qualquer tecla quando terminar')

% Define diretórios de saída do Riots
dirsaida = pwd; % O dir. de saída deve ser o dir. onde arquivo atual está gravado

% Define diretórios de instalação do Riots
dirinstal = ('C:\user\olavo\Mestrado\Riots\toolbox\Riots_95\System');

% Cria variável com o nome do arquivo .c e dll que serão manipulados
nomec = strcat(nome, '.c');
nomedll = strcat(nome, '.dll');

% Apaga dll's antigas no diretório corrente e no diretório de instalação do Riots
delete(nomedll);
delete('simulate.dll');
cd(dirinstal);
delete(nomedll);

% Cria .bat no diretório de instalação que irá chamar o compilador do Matlab 5.1
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)

% Executa o script "cel.bat" na janela DOS que por sua vez irá compilar o
% arquivo com o nome "nome_deste_arquivo.c" em um dll de nome:
% "nome_deste_arquivo.dll".
disp('tecle cel.bat na janela dos')
pause
copyfile(nomedll, dirsaida)
cd(dirsaida);
copyfile(nomedll, 'simulate.dll')
delete(nomedll);

%% 2ª PARTE: ESCOLHA DOS PARÂMETROS DO SISTEMA E DE DISCRETIZAÇÃO
N = 50; % número de pontos da malha
tf = 1; % tempo final

```

```

t = [0:tf/N:tf];           % discretização do vetor de tempo
r = 4;                     % ordem da spline

% Condição inicial (ângulos em radianos):
phi = [1.0472 0.2907 4.5513];
phip = [6.2832 -1.3760 3.4240];
x0 = [phi'; phip'];

% Condição final:
% Retorna as coord (pos e vel), no sistema de coord. do 4barras, para um
% dado tempo "tf" (dado vindo da análise cinemática do mecanismo)
xf = pos_final(tf);
x1f = xf(1);  x2f = xf(2);  x3f = xf(3);  x4f = xf(4);  x5f = x0(5);
x6f = x0(6);

% Controle inicial (nº de controles (3) x nº de pontos da malha temporal)
u0 = zeros(3,N+r-1);
u0 = uol;
clear uol

% Limites máximo e mínimo nas variáveis de controle
uboundl = []; % Usa-se o valor default = -Inf
uboundh = []; % Usa-se o valor default = +Inf

% Parametros para serem repassados ao Riots
global params
params = [x1f x2f x3f x4f x5f x6f 0 0 0 0 0 0 0 0 0];

%% Solução do problema de controle ótimo
[info,simed] = simulate(0,params)
simulate(0)
[f,x] = simulate(1,x0,u0,t,5,2);

% Parâmetros de otimização do RIOTS
% EPS: Overall optimization tolerance. default = 1E-6
EPS = 5e-3;
% EPSNEQ: Nonlinear constraint tolerance. default = 1E-4.
EPSNEQ = 1e-4;
% OBJREP: Indicates function precision. default = 0.
OBJREP = 0;
% BIGBND: A number large than the largest magnitude expected for the decision
% variables. Default = 1E6.

```

```

BIGBND = 1E6;
% definição das tolerâncias de otimização global e de violação de vínculos
% de estado (p. 67 do manual)
cond = [EPS, EPSNEQ, OBJREP, BIGBND];

% Início da otimização propriamente dita
med_tempo = clock;
tic
[uo1,xo1,fo1] = riots(x0,u0,t,uboundl,uboundh,[params],1000,5,cond);
toc
tempo_cpu = etime(clock,med_tempo)/60

% Contagem de tempo
if tempo_cpu < 61
    disp(['O Riots levou ' num2str(tempo_cpu*60) ' segundos (= ' ...
        num2str(tempo_cpu) ' min.) para resolver este problema.'])
else
    disp(['O Riots levou ' num2str(tempo_cpu) ' minutos (= ' ...
        num2str(tempo_cpu/60) ' h) para resolver este problema.'])
end

save(nome)
plotagem_phi(nome)

```

#### C.4.2 EQUAÇÕES DE ESTADO: EOM2SS

Apresenta-se aqui o código fonte do programa EOM2SS - *Equation of Motion to State Space*, desenvolvido durante esta dissertação para deduzir as equações de movimento do pêndulo triplo, utilizado no Capítulo 3 e colocá-las na forma de equações de estado. Maiores detalhes sobre o desenvolvimento deste algoritmo aparecem no Apêndice A.

```

%% EOM2SS: Equation of Motion to State Space
% AUTOR: Olavo Luppi Silva – olavo@ime.eb.br
function xp = eom2ss_1c(F,G,u,pr,output_file)

%% DESCRIÇÃO:
% .O objetivo desta function é derivar a equação de movimento do pêndulo
% triplo, colocá-la na forma de estado e exportar uma function para no
% formato adequado para ser utilizado por algum ODE.

```

```

%% LIMITAÇÕES:
% 1. É esperado acontecer o erro: "Exceeded maximum number of terms (20).
% Inputs may be in invalid form (not a polynomial). To increase maximum
% number, use SymPoly2Sym(F,DV,Max#)". Este erro é devido à forma recursiva
% que se usa o comando SymPoly2Sym para montar as matrizes A, B, C, D e E.
% 2. Este programa fica mais rápido quando valores constantes,
% (como comprimentos, massa e momento de inércia) são declarados na
% função principal. Como consequência, os únicos valores não numéricos
% que aparecem no arquivo de saída ('output_file') são os das variáveis de
% estado (x1, x2,...x6). No entanto, é possível obter um arquivo de saída
% com todas as variáveis simbólicas se forem comentadas todas as linhas em
% que aparece o comando "vpa".

%% HISTÓRICO da versão 1b:
% 12/03/07: .Criação do arquivo a partir do eom2ss_1a.m.
%
% .Neste arquivo procura-se colocar a matriz "F" – que
% representa a matriz das forças externas – como um argumento de
% entrada já que o mesmo sistema pode ser simulado segundo um
% diferente conjunto de forças atuantes sobre ele.
%
% .Incorporado o uso do "vpa" para forçar que os valores
% numéricos do tipo 10/3 apareçam como 3.3333.
%
% .Incorporado o comando "simple" após calcular o vetor qpp para
% diminuir o tamanho do vetor "xp".

%% HISTÓRICO da versão 1c:
% 30/03/07: .Criação do arquivo a partir do eom2ss_1b.m.
%
% .Inserida a matriz G que irá representar as forças
% generalizadas da mola no problema do pêndulo triplo com molas
% na extremidade para simular o comportamento do 4 barras.
% 02/04/07: .Correção do bug na exportação do arquivo da function. Antes
% ele escrevia a primeira linha da function como:
%
% "function xp =pend_triplo_mola.m(t,x)" e o acréscimo do ".m"
% fazia que desse um erro na hora da integração.
syms L1 L2 L3 L4 m1 m2 m3 g I1 I2 I3
syms phi1 phi2 phi3 dphi1 dphi2 dphi3 d2phi1 d2phi2 d2phi3
syms U1 U2 U3
global L1 L2 L3 m1 m2 m3 g I1 I2 I3

%% Vetor posição dos centros de massa:
origem = [0;0];
pos1 = origem + ...
        [(L1/2)*cos(phi1); (L1/2)*sin(phi1)];% Posição do centro de massa da barral

```

```

pos2 = [L1*cos(phi1); L1*sin(phi1)] + ...
        [(L2/2)*cos(phi2); (L2/2)*sin(phi2)]; % Posição do centro de massa da barra2
pos3 = [L1*cos(phi1); L1*sin(phi1)] + [L2*cos(phi2); L2*sin(phi2)] + ...
        [(L3/2)*cos(phi3); (L3/2)*sin(phi3)]; % Posição do centro de massa da barra2
%% Vetor velocidade dos centros de massa:
vel1 = fulldiff(pos1, {phi1, phi2, phi3});
vel2 = fulldiff(pos2, {phi1, phi2, phi3});
vel3 = fulldiff(pos3, {phi1, phi2, phi3});
%% Energia Cinética de cada corpo
T1 = (1/2)*m1*(vel1(1)^2 + vel1(2)^2) + (1/2)*I1*dphi1^2;
T2 = (1/2)*m2*(vel2(1)^2 + vel2(2)^2) + (1/2)*I2*dphi2^2;
T3 = (1/2)*m3*(vel3(1)^2 + vel3(2)^2) + (1/2)*I3*dphi3^2;
%% Energia Potencial de cada corpo
V1 = m1*g*pos1(2);
V2 = m2*g*pos2(2);
V3 = m3*g*pos3(2);
%% Lagrangeano
T = T1 + T2 + T3;
V = V1 + V2 + V3;
Lag = T - V;

%% Diferenciação Eq1:
% Derivada parcial do lagrangeano em relação à posição da 1ª coord.
dL_phi1 = diff(Lag, phi1);
% Derivada parcial do lagrangeano em relação à velocidade da 1ª coord.
dL_dphi1 = diff(Lag, dphi1);
% Derivada total temporal da expressão anterior
d_dL_dphi1_dt = fulldiff(dL_dphi1, {phi1, phi2, phi3, dphi1, dphi2, dphi3});
% Montagem da primeira equação de Lagrange (em relação a 1ª coordenada)
eq1 = d_dL_dphi1_dt - dL_phi1;
%% Diferenciação Eq2:
% Derivada parcial do lagrangeano em relação à posição da 2ª coord.
dL_phi2 = diff(Lag, phi2);
% Derivada parcial do lagrangeano em relação à velocidade da 2ª coord.
dL_dphi2 = diff(Lag, dphi2);
% Derivada total temporal da expressão anterior
d_dL_dphi2_dt = fulldiff(dL_dphi2, {phi1, phi2, phi3, dphi1, dphi2, dphi3});
% Montagem da primeira equação de Lagrange (em relação a 2ª coordenada)
eq2 = d_dL_dphi2_dt - dL_phi2;
%% Diferenciação Eq3:
% Derivada parcial do lagrangeano em relação à posição da 3ª coord.
dL_phi3 = diff(Lag, phi3);

```



```

% Derivada parcial do lagrangeano em relação à velocidade da 3ª coord.
dL_dphi3 = diff(Lag,dphi3);
% Derivada total temporal da expressão anterior
d_dL_dphi3_dt = fulldiff(dL_dphi3,{phi1,phi2,phi3,dphi1,dphi2,dphi3});
% Montagem da primeira equação de Lagrange (em relação à 2ª coordenada)
eq3 = d_dL_dphi3_dt - dL_phi3;
%% Simplificação das equações
eq1 = simple(eq1);
eq2 = simple(eq2);
eq3 = simple(eq3);

%% Formatando a equação 1
eq1 = expand(eq1);
eq1 = collect(eq1,d2phi1); eq1 = collect(eq1,d2phi2);
eq1 = collect(eq1,d2phi3);
eq1 = collect(eq1,dphi1); eq1 = collect(eq1,dphi2);
eq1 = collect(eq1,dphi3);
eq1 = collect(eq1,dphi1^2); eq1 = collect(eq1,dphi2^2); eq1 = collect(eq1,dphi3^2);
eq1 = collect(eq1,phi1); eq1 = collect(eq1,phi2);
eq1 = collect(eq1,phi3);
%% Formatando a equação 2
eq2 = expand(eq2);
eq2 = collect(eq2,d2phi1); eq2 = collect(eq2,d2phi2);
eq2 = collect(eq2,d2phi3);
eq2 = collect(eq2,dphi1); eq2 = collect(eq2,dphi2);
eq2 = collect(eq2,dphi3);
eq2 = collect(eq2,dphi1^2); eq2 = collect(eq2,dphi2^2); eq2 = collect(eq2,dphi3^2);
eq2 = collect(eq2,phi1); eq2 = collect(eq2,phi2);
eq2 = collect(eq2,phi3);
%% Formatando a equação 3
eq3 = expand(eq3);
eq3 = collect(eq3,d2phi1); eq3 = collect(eq3,d2phi2);
eq3 = collect(eq3,d2phi3);
eq3 = collect(eq3,dphi1); eq3 = collect(eq3,dphi2);
eq3 = collect(eq3,dphi3);
eq3 = collect(eq3,dphi1^2); eq3 = collect(eq3,dphi2^2); eq3 = collect(eq3,dphi3^2);
eq3 = collect(eq3,phi1); eq3 = collect(eq3,phi2);
eq3 = collect(eq3,phi3);

%% MONTAGEM DO SISTEMA DE EQUAÇÕES:
% O sistema de equações do pêndulo triplo é do tipo:
% A*qp + B*qp^2 + C*qp + D*q + E = F*u + G, onde:(*atenção!

```

```

%
alteração na versão 1c aqui*)
% qpp = [d2phi1; d2phi3; d2phi3] % Acelerações das coordenadas
% qp = [dphi1; dphi3; dphi3] % Velocidades das coordenadas
% q = [phi1; phi3; phi3] % Posições das coordenadas
% u = [u1; u2; u3] % Torques externos (controles)
% A,B,C,D,E e F % Matrizes que agrupam cada um dos termos
% G % Matriz que irá formar o vetor de forças
% generalizadas (externas) devido às molas,
% amortecedores, etc.

% Montagem das matrizes
EQ = [eq1; eq2; eq3];
var = struct('matrixA', {'d2phi1', 'd2phi2', 'd2phi3'},...
            'matrixB', {'dphi1^2', 'dphi2^2', 'dphi3^2'},...
            'matrixC', {'dphi1', 'dphi2', 'dphi3'},...
            'matrixD', {'phi1', 'phi2', 'phi3'});

syms A B C D E
for j=1:3
    for i=1:3
        % MATRIX A
        a = SymPoly2Sym(EQ(j),var(i).matrixA);
        if length(a) == 2 % então a variável var(i).matrixA está na equação
            A(j,i) = a(1);
            EQ(j) = a(2);
        elseif length(a) == 1 % a variável var(i).matrixA não está e o
            A(j,i) = 0; % coeficiente da matriz deve ser zero
            EQ(j) = a(1);
        else
            A(j,i) = 0;
        end
    end
end
for i=1:3
    % MATRIX B
    b = SymPoly2Sym(EQ(j),var(i).matrixB);
    if length(b) == 2
        B(j,i) = b(1);
        EQ(j) = b(2);
    elseif length(b) == 1
        B(j,i) = 0;
        EQ(j) = b(1);
    else
        B(j,i) = 0;
    end
end

```

```

        end
    end
    for i=1:3
        % MATRIX C
        c = SymPoly2Sym(EQ(j),var(i).matrixC);
        if length(c) == 2
            C(j,i) = c(1);
            EQ(j) = c(2);
        elseif length(c) == 1
            C(j,i) = 0;
            EQ(j) = c(1);
        else
            C(j,i) = 0;
        end
    end
end
for i=1:3
    % MATRIX D
    d = SymPoly2Sym(EQ(j),var(i).matrixD);
    if length(d) == 2
        D(j,i) = d(1);
        EQ(j) = d(2);
    elseif length(d) == 1
        D(j,i) = 0;
        EQ(j) = d(1);
    else
        D(j,i) = 0;
    end
end
end
E = EQ;

%% Variable precision arithmetic
% Usado para que os valores numéricos apareçam como 3.3333 e não 10/3.
%pr = 16;
A = vpa(A,pr); B = vpa(B,pr); C = vpa(C,pr);
D = vpa(D,pr); E = vpa(E,pr); F = vpa(F,pr);

%% Montagem do vetor de acelerações:
q = [phi1; phi2; phi3];
qp = [dphi1; dphi2; dphi3];
qpp = inv(A)*(F*u + G - (B*qp.^2 + C*qp + D*q + E));
qpp = simple(qpp);

```

```

qpp = vpa(qpp,pr);

%% DEFINIÇÃO DAS VARIÁVEIS DE ESTADO
% x1 = phi1 -> dx1 = x4
% x2 = phi2 -> dx2 = x5
% x3 = phi3 -> dx3 = x6
% x4 = dphi1 -> dx4 = qpp(1)
% x5 = dphi2 -> dx5 = qpp(2)
% x6 = dphi3 -> dx6 = qpp(3)
syms x1 x2 x3 x4 x5 x6
qpp = subs(qpp,phi1,x1);
qpp = subs(qpp,phi2,x2);
qpp = subs(qpp,phi3,x3);
qpp = subs(qpp,dphi1,x4);
qpp = subs(qpp,dphi2,x5);
qpp = subs(qpp,dphi3,x6);
disp('Vetor derivada das variáveis de estado:')
xp = [ x4;
       x5;
       x6;
       qpp(1);
       qpp(2);
       qpp(3)];

%% EXPORTAÇÃO DO ARQUIVO
if nargin < 5
    return
else
    % Este trecho exporta o arquivo que deve ser utilizado pelo ODE
    function_name = output_file; %Bug fix de 02/04/07
    output_file = strcat(output_file, '.m')
    fid = fopen(output_file, 'w');
    texto = strcat('function xp = ', function_name, '(t,x) \n\n'); %BugFix de 02/04/07
    fprintf(fid, texto);
    fprintf(fid, 'global L1 L2 L3 m1 m2 m3 I1 I2 I3 g\n\n');
    for i=1:6
        texto = strcat('x', num2str(i), ' = x(', num2str(i), ');');
        fprintf(fid, '%s\n', texto);
    end
    fprintf(fid, '\n\nxp = [\n');
    for i=1:length(xp)
        texto = char(xp(i));

```

```

        fprintf(fid, '%s\n', texto);
    end
    fprintf(fid, ' ');
    fclose(fid);
    disp(['File "' output_file '" successfully created!'])
end
%% =====
% FIM DO PROGRAMA PRINCIPAL
%% =====

```

### C.4.3 EXPORTAÇÃO DE VETORES E MATRIZES EM LINGUAGEM C

O programa principal deriva as equações de movimento e calcula a matriz Jacobiana das equações de estado e de restrição, usando o seu manipulador simbólico. Estas expressões precisam ser exportadas para serem incorporadas ao arquivo em linguagem C que será posteriormente compilado e utilizado pelo RIOTS. Apresenta-se aqui uma pequena rotina para facilitar a exportação em C.

```

function exportar_c(var_exportada, nome_var, nome_arquivo)
% 14/12/2006: Olavo Luppi Silva
% Esta function recebe como entrada uma variável simbólica "x" na forma de vetor
% ou matriz e converte para um arquivo em .C com o nome "nome_do_arquivo".
% O resultado da exportação é um arquivo com a estrutura em .C já
% formatada, isto é, com as quebras de linha, pronto para ser copiado e
% colado no programa principal.
% INPUT:
% - "var_exportada": variável simbólica do workspace do Matlab que será
%   convertida em código .C.
% - "nome_var": string que define o nome que irá aparecer dentro do arquivo
%   em C.
% - "nome_arquivo": string que define o nome do arquivo texto que será
%   gravado no diretório corrente.
% OUTPUT:
% - para o workspace a function não retorna nenhuma saída.
%=====
% Versão 2 da function "exportar_c.m"
%=====
fid = fopen(nome_arquivo, 'w');
xc = ccode(var_exportada);
xc = strrep(xc, '\n', '\r');

```

```

xc = strrep(xc, 'var_exportada', nome_var);

%-----
% Este trecho verifica se é vetor linha ou coluna e exporta com o número
% correto de índices
%-----
[m,n] = size(var_exportada);
if n == 1 % Se for vetor linha...
    for i=0:(m-1)
        xc = strrep(xc,[nome_var '[' num2str(i) ']' '[0]' ],[nome_var '[' num2str(i) ']]);
    end
end
if m == 1
    for i=0:(n-1)
        xc = strrep(xc,[nome_var '[0]' '[' num2str(i) ']'],[nome_var '[' num2str(i) ']]);
    end
end
%-----
xc = strrep(xc, '↵', '');
fprintf(fid,xc);
fclose(fid);
disp(['O arquivo ' nome_arquivo ' foi criado no diretório ' pwd])

%=====
% Versão 1 da function "exportar_c.m"
%=====
% fid = fopen(nome_arquivo,'w');
% xc = ccode(var_exportada);
% xc2 = strrep(xc, '↵', '\r');
% xc2 = strrep(xc2, 'var_exportada', nome_var);
% xc2 = strrep(xc2, '↵', '');
% fprintf(fid,xc2);
% fclose(fid);
% disp(['O arquivo ' nome_arquivo ' foi criado no diretório ' pwd])

```

#### C.4.4 VÍNCULOS TERMINAIS

Nas primeiras tentativas do CASO 1 (Seção 3.4.1) tentou-se incluir vínculos terminais. Esta *function* tinha a função de retornar o valor das coordenadas angulares calculadas na análise cinemática para qualquer tempo  $t$ . Nas análises posteriores ela deixou de ser utilizada pois a abordagem que se mostrou mais adequada prescindiu de vínculos

terminais.

```
function q = pos_final(t)
% Esta function carrega os resultados da análise cinemática e retorna a
% posição e a velocidade das variáveis angulares do mecanismo 4 barras,
% no tempo desejado 't'
load Haug4B_6b tt R_q R_qp
phi1 = interp1(tt,R_q(3,:),t);
phi2 = interp1(tt,R_q(6,:),t);
phi3 = interp1(tt,R_q(9,:),t);
phi1p = interp1(tt,R_qp(3,:),t);
phi2p = interp1(tt,R_qp(6,:),t);
phi3p = interp1(tt,R_qp(9,:),t);
q = [phi1 phi2 phi3 phi1p phi2p phi3p];
```

#### C.4.5 VISUALIZAÇÃO DOS RESULTADOS

Esta *function* tem o papel de reunir os recursos gráficos de visualização dos resultados, tais como animação e gráficos, fazendo a comparação com a análise cinemática calculada no Capítulo 2.

```
function plotagem_phi(nome_mat)
% Nesta function concentra-se a visualização gráfica dos resultados obtidos
% com o RIOTS.

%% Carrega os dados na memória:
load(nome_mat)
%% ANIMAÇÃO
ptoA = [0.0; 0.0];
figure(1)
for i=1:N+1          % N = número de pontos da malha
    x1 = xol(1,i);
    x2 = xol(2,i);
    x3 = xol(3,i);
    x4 = xol(4,i);
    x5 = xol(5,i);
    x6 = xol(6,i);
    % Cálculo da força da mola
    Δ_x = L1*cos(x1) + L2*cos(x2) + L3*cos(x3) - L4;
    Δ_y = L1*sin(x1) + L2*sin(x2) + L3*sin(x3);
```

```

    Δ_xp = -L1*x4*sin(x1) - L2*x5*sin(x2) - L3*x6*sin(x3);
    Δ_yp = L1*x4*cos(x1) + L2*x5*cos(x2) + L3*x6*sin(x3);
    Fx(i) = -K*Δ_x;
    Fy(i) = -K*Δ_y;
    Cx(i) = -C*Δ_xp;
    Cy(i) = -C*Δ_yp;
    ptoB = ptoA + [L1*cos(x1); L1*sin(x1)];
    ptoC = ptoB + [L2*cos(x2); L2*sin(x2)];
    ptoD = ptoC + [L3*cos(x3); L3*sin(x3)];
    home
    dist1 = norm(ptoB - ptoA) % Verificação do comprimento das barras
    dist2 = norm(ptoC - ptoB)
    dist3 = norm(ptoD - ptoC)
    clf
    hold on
    plot(ptoA(1),ptoA(2), 'kx')
    plot(ptoB(1),ptoB(2), 'kx')
    plot(ptoC(1),ptoC(2), 'kx')
    plot(ptoD(1),ptoD(2), 'kx')
    hold off
    axis('square')
    axis([-2 2 -2 2])
    %axis('equal')
    line([ptoA(1),ptoB(1)],[ptoA(2),ptoB(2)], 'Color', 'g')
    line([ptoB(1),ptoC(1)],[ptoB(2),ptoC(2)], 'Color', 'r')
    line([ptoC(1),ptoD(1)],[ptoC(2),ptoD(2)], 'Color', 'b')
    %line([ptoD(1),ptoA(1)],[ptoD(2),ptoA(2)], 'Color', 'k')
    % Textos
    text(ptoA(1) - 0.05,ptoA(2) - 0.05, 'A')
    text(ptoB(1) - L1/10*sin(x1),ptoB(2) + L1/10*cos(x1), 'B')
    text(ptoC(1) - L2/10*sin(x2),ptoC(2) + L2/10*cos(x2), 'C')
    text(ptoD(1) + 0.05,ptoD(2) - 0.05, 'D')
    grid
    filme(:,i) = getframe;
end

% Carrega-se dados da análise cinemática para comparação com os resultados
% obtidos por controle ótimo
load Haug4B_6b R_q R_qp tt
X = [R_q([3 6 9],:); R_qp([3 6 9],:)];
X = X*180/pi;
xol = xol*180/pi;

```



```

%% GRÁFICOS
figure(2)
subplot(311)
plot(t,xol(1:3,:)*180/pi)
title('Estados calculados pelo Controle Ótimo (ÂNGULOS)')
legend('x1=\phi_1', 'x2=\phi_2', 'x3=\phi_3')
ylabel('x [deg]')
xlabel('t(s)')
grid
subplot(312)
plot(t,xol(4:6,:)*180/pi)
title('Estados calculados pelo Controle Ótimo (VELOCIDADES)')
legend('x4=d(\phi_1)/dt', 'x5=d(\phi_2)/dt', 'x6=d(\phi_3)/dt')
ylabel('xdot [deg/s]')
xlabel('t(s)')
grid
subplot(313)
Sp_plot(t,uol)
title('Controle Ótimo')
legend('U1', 'U2', 'U3')
ylabel('Torques [N.m]')
xlabel('t(s)')
grid

figure(3)
subplot(2,2,1)
plot( tt,X(1,:), 'b',tt,X(2,:), 'g',tt,X(3,:), 'r',...
      t,xol(1,:), 'b.',t,xol(2,:), 'g.',t,xol(3,:), 'r.')
title('Posições pend. triplo')
legend('\phi_1 OCP', '\phi_2 OCP', '\phi_3 OCP', '\phi_1 Haug',
       '\phi_2 Haug', '\phi_3 Haug',2)
ylabel('x')
xlabel('t(s)')
grid
subplot(2,2,2)
plot( tt,X(4,:), 'b',tt,X(5,:), 'g',tt,X(6,:), 'r',...
      t,xol(4,:), 'b.',t,xol(5,:), 'g.',t,xol(6,:), 'r.')
title('Velcidades pend. triplo')
legend('d(\phi_1)/dt OCP', 'd(\phi_2)/dt OCP', 'd(\phi_3)/dt OCP', 'd(\phi_1)/dt Haug',
       'd(\phi_2)/dt Haug', 'd(\phi_3)/dt Haug',3)
ylabel('x_p')
xlabel('t(s)')

```

```

grid
subplot(2,2,3)
[AX,H1,H2] = plotyy(t,Fx,t,Fy);
title('Força das molas')
set(get(AX(1),'Ylabel'),'String','Hor. Spring Force - Kx (N)')
set(get(AX(2),'Ylabel'),'String','Vert. Spring Force - Ky (N)')
xlabel('t(s)')
subplot(2,2,4)
[AX,H1,H2] = plotyy(t,Cx,t,Cy);
title('Força dos amortecedores')
set(get(AX(1),'Ylabel'),'String','Hor. Damper Force - Cx (N)')
set(get(AX(2),'Ylabel'),'String','Vert. Damper Force - Cy (N)')
xlabel('t(s)')

```

#### C.4.6 CÓDIGO EM LINGUAGEM C PARA SER COMPILADO

Apesar de ser possível utilizar o RIOTS totalmente dentro do ambiente Matlab, é possível conseguir um aumento de performance muito significativo se o usuário fornecer a descrição do problema de controle ótimo em código C. O arquivo abaixo é uma amostra de como deve ser a estrutura desse arquivo .C e foi criado com base no arquivo referente à simulação IVbar3act-mola-amort-EOM2SS-015 (Tabela B.6)

```

/* =====
MECANISMO 4 BARRAS - Dinâmica Inversa 12/12/2006
Cálculo do torque (u1) de apenas 1 atuador
=====
Descrição do problema:
* cost: int_0^tf u1^2 + u2^2 + u3^2 dt
* xp: eq. de movimento do pêndulo triplo (3GDL)
* x(0) = x(tf): estado inicial é igual ao estado final
* tf = 1: tempo final especificado
*/

#include <math.h>
#define F_NUM neq[4]
#define SQP_XU 0 /*Copiado do Parabola.c*/
#define NSTATES 6 /* Numero de estados*/
#define NINPUTS 3 /* Numero de inputs*/
#if SQP_XU /*Copiado do Parabola.c*/
#define LT_INEQ 0 /*Copiado do Parabola.c*/

```

```

#else          /*Copiado do Parabola.c*/
#define LT_INEQ 1 /*Copiado do Parabola.c*/
#endif        /*Copiado do Parabola.c*/
#define LTI    4 /*Copiado do Parabola.c*/
#define NLEEC  0 /*Non Linear Endpoint Equality Constraint
                 /* (vínculos igualdade terminais não-lineares)*/
#define NLTIC  1 /*Non Linear Trajectory Inequality Constraint*/
#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; */
/* A linha acima foi comentada (e a de baixo criada pq o comando ccode do Matlab7
   dá esses nomes de MapleGenVar para as variáveis auxiliares que ele precisa criar
   na hora de exportar para a linguagem C
   */
double MapleGenVar1, MapleGenVar2, MapleGenVar3, MapleGenVar4, MapleGenVar5, MapleGenVar6, MapleGenVar7;
static double x1f,x2f,x3f,x4f,x5f,x6f,Q11,Q22,Q33,Q44,Q55,Q66,R11,R22,R33 ;
/* parâmetros */

/*Contador de iterações)*/
int iter;
iter = 0;
void activate(message)
    char **message;
{
    *message = "IVbar3act_mola_amort_EOM2SS_001 testado em 24/04/2007";
}
void init(neq,params)
    int neq[];
    double *params;
{
    if ( params == 0 ) {

```

```

    neq[0] = NSTATES;
    neq[1] = NINPUTS;
    neq[2] = N_PARAMS;
    neq[6] = NLTIC; /*Non Linear Trajectory Inequality Constraint */
    neq[10] = NLEEC;
/* neq[0] = NSTATES; /*Esse bloquinho foi copiado do Parabola.c
    neq[1] = NINPUTS;
    neq[7] = LT_INEQ;
    neq[12] = LTI;*/
}
else {
    x1f = params[0];
    x2f = params[1];
    x3f = params[2];
    x4f = params[3];
    x5f = params[4];
    x6f = params[5];
    Q11 = params[6]; /*A principio não irei precisar destes parâmetros*/
    Q22 = params[7]; /*Então, devo lembrar de passar valores nulos a partir*/
    Q33 = params[8]; /*do 7º elemento do vetor de parâmetros que é definido*/
    Q44 = params[9]; /*no arquivo do matlab*/
    Q55 = params[10];
    Q66 = params[11];
    R11 = params[12];
    R22 = params[13];
    R33 = params[14];
return;
}
}

void h(neq,t,x,u,xdot) /* Equação de movimento na forma de estado */
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],xdot[NSTATES];
{

xdot[0] = x4;
xdot[1] = x5;
xdot[2] = x6;
MapleGenVar1 = /*MapleGenVar são variáveis auxiliares que o Matlab cria */
MapleGenVar5 = /*quando a expressão a ser exportada é muito grande. */
MapleGenVar4 = /*Aqui se substituiu o conteúdo original do arquivo por esse*/
MapleGenVar5 = /*comentário por uma questão de economia de espaço*/

```

```

MapleGenVar3 =
MapleGenVar4 =
MapleGenVar2 =
xdot[3] = MapleGenVar1*MapleGenVar2;
xdot[4] = MapleGenVar1*MapleGenVar2;
xdot[5] = MapleGenVar1*MapleGenVar2;

    /*Atualiza contador de iterações*/
    iter = iter+1;
    /*Mostra como evolui iterações, variáveis de estado e de controle*/
    printf("%i %f %f %f \r",iter,U1,U2,U3);
}

/*Jacobianos da equação de movimento*/
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];
{
/*Jacobiano em relação às variáveis de estado 'x': dh/dx */
h_x[0][0] = /*removido*/
h_x[0][1] = /*removido*/
h_x[0][2] = /*removido*/
/* Aqui continuam às demais declarações dos elementos do Jacobiano que foram
removidos por questão de economia de espaço. */
h_x[5][5] = /*removido*/

/*Jacobiano em relação às variáveis de controle 'u': dh/du*/
h_u[0][0] = /*removido*/
h_u[0][1] = /*removido*/
h_u[0][2] = /*removido*/
/* Aqui continuam as demais declarações dos elementos do Jacobiano que foram
removidos por questão de economia de espaço. */
h_u[5][2] = /*removido*/

}

/*A função "l" tem o papel de armazenar tanto a expressão da função objetivo*/
/*quanto a restrição de desigualdade de estado na variável zdot*/
double l(neq,t,x,u)
    int neq[];

```

```

    double *t,x[NSTATES],u[NINPUTS];
{
double zdot;

if ( neq[4] == 1 ) /*função objetivo*/
    zdot = u[0]*u[0] + u[1]*u[1] + u[2]*u[2]; /* Função objetivo */
else
    /* Restrição desigualdade de estado */
    zdot = pow(cos(x1)/2.0+9.0/10.0*cos(x2) + 7.0/10.0*cos(x3)-1.0,2.0) + ...
        pow(sin(x1)/2.0+9.0/10.0*sin(x2) + 7.0/10.0*sin(x3),2.0) + ...
        pow(-x4*cos(x1)/2.0-9.0/10.0*x5*cos(x2)-7.0/10.0*x6*cos(x3),2.0) + ...
        pow(-x4*sin(x1)/2.0-9.0/10.0*x5*sin(x2)-7.0/10.0*x6*sin(x3),2.0) +
        pow(x4-2.0*0.3141592653589793E1,2.0);
return zdot;
}

/*Jacobianos da função objetivo e da restrição desigualdade de estado */
double Dl(neq,t,x,u,l_x,l_u) /*Jacobiano da função objetivo (em rel. a "x" e a "u"), dl/
    int neq[];
    double *t,x[NSTATES],u[NINPUTS],l_x[NSTATES],l_u[NINPUTS];
{
/* Jacobiano da Função objetivo em relação às variáveis de estado (dl/dx)*/
if ( neq[4] == 1 ) {
    l_x[0] = 0;
    l_x[1] = 0;
    l_x[2] = 0;
    l_x[3] = 0;
    l_x[4] = 0;
    l_x[5] = 0;
/* Jacobiano da Função objetivo em relação às variáveis de controle (dl/du)*/
    l_u[0] = 2*u[0];
    l_u[1] = 2*u[1];
    l_u[2] = 2*u[2];
}
else
{
/* Jacobiano da restrição desigualdade de estado em relação */
/* às variáveis de estado (dl/dx)*/
l_x[0] = /*removido*/
l_x[1] = /*removido*/
l_x[2] = /*removido*/
l_x[3] = /*removido*/
l_x[4] = /*removido*/

```

```

    l_x[5] = /*removido*/
    /* Jacobiano da restrição desigualdade de estado em relação */
    /* às variáveis de controle (dl/du)*/
    l_u[0] = 0.0;
    l_u[1] = 0.0;
    l_u[2] = 0.0;
}
return 0.0;
}

/* Função que declara os vínculos terminais */
double g(neq,t,x0,xf)
    int neq[];
        double *t,x0[NSTATES],xf[NSTATES];
{
    return 0.0; /*Como não há vinculo terminal, a função retorna zero*/
}

/* Função que declara o Jacobiano 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];
{
    return 0.0; /*Como não há vinculo terminal, a função retorna zero*/
}

```

# Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

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

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)



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