

FLAVIO A. M. CIPPARRONE
Eng. Eletricista, Universidade de São Paulo, 1989

UM SIMULADOR DINÂMICO PARA ROBÔS
MANIPULADORES DE CADEIA ABERTA

Dissertação apresentada ao Depto. de
Eng. Mecânica da Escola Politécnica
da USP para obtenção do título de
Mestre em Engenharia.

Orientador: Prof. Dr. José Jaime da Cruz.

São Paulo, 1991

Sumário

Neste trabalho, realizamos o projeto e a implementação de um simulador dinâmico aplicável a qualquer robô de cadeia aberta. Este simulador compõe-se de módulos eficientes para cálculo da dinâmica inversa e sua linearização, dinâmica direta, bem como de módulos gráficos e de controle.

A entrada do simulador é feita a partir de arquivos em formato livre, que descrevem os parâmetros geométricos e dinâmicos do modelo de projeto e do "robô real". Este último é necessário para conseguirmos uma simulação mais realista de esquemas de controle. Pode-se facilmente construir e testar o desempenho de diversos destes esquemas.

O simulador permite ainda a visualização gráfica de diversas variáveis dinâmicas tais como forças, posições, velocidades, etc. . .

No texto apresentamos exemplos de utilização para o caso do pêndulo duplo e do robô Stanford.

Abstract

A dynamical simulator that can be applied to any open chain manipulator is developed in this work. The simulator is composed of modules that efficiently perform the computation of the direct dynamics as well as the inverse dynamics and its linearized form. Control and graphical modules are included. Its modular structure is such that one can easily build specific control blocks and evaluate their performances.

Data input is done through free format files describing dynamical and geometrical parameters from both the design and the actual manipulator models. The last one is useful for a more realistic evaluation of control systems.

The simulator allows the graphical visualization of the various dynamical variables involved such as forces, positions, velocities, etc. . .

Examples of utilization of the simulator are presented for both the double pendulum and the Stanford manipulator.

Conteúdo

1 INTRODUÇÃO	7
1.1 ASPECTOS BÁSICOS	7
1.1.1 Definição de Robô	7
1.1.2 Constituição e Classificação	8
1.1.3 Problemas Clássicos da Robótica	8
1.2 OBJETIVOS	12
1.3 COMPOSIÇÃO	12
1.4 CONSIDERAÇÕES SOBRE EFICIÊNCIA	13
1.5 APLICAÇÕES	13
1.6 CONTRIBUIÇÕES	14
1.7 ESTRUTURA DO TRABALHO	15
2 MODELAGEM DINÂMICA	16
2.1 PARÂMETROS DE DENAVIT-HARTENBERG	16
2.2 ALGUNS FATOS DA CINEMÁTICA	17
2.2.1 Matrizes Homogêneas	17
2.2.2 Simplificações das Derivadas Parciais	20
2.3 OBTENÇÃO DO MODELO DINÂMICO	22
2.3.1 Velocidade de um Ponto	22
2.3.2 Cálculo da Energia Cinética de um Ligamento	23
2.3.3 Cálculo da Energia Potencial de um Ligamento	25

2.3.4	Lagrangiano do Sistema	25
2.3.5	Equações Dinâmicas	26
2.4	UMA DEMONSTRAÇÃO ALTERNATIVA E OUTRAS RELAÇÕES . . .	29
2.4.1	Relações entre Coeficientes Dinâmicos - I	29
2.4.2	A Demonstração Alternativa	29
2.4.3	Relações Entre Coeficientes Dinâmicos - II	30
2.4.4	Cálculos Necessários	30
2.5	FORMAS MATRICIAIS	31
2.6	O CÁLCULO DAS FORÇAS GENERALIZADAS	33
3	O CÁLCULO EFICIENTE DOS MODELOS	35
3.1	HISTÓRICO	35
3.2	O ALGORITMO PARA O MDD	38
3.3	O ALGORITMO PARA O MDL	42
3.3.1	Cálculo de A	43
3.3.2	Cálculo de B	43
3.3.3	Cálculo de C	43
3.3.4	Simplificações	49
3.4	A UTILIZAÇÃO DOS SISTEMAS DE LIGAMENTO	50
3.4.1	Simplificações	52
3.5	O ALGORITMO PARA O MDI	53
3.5.1	O Algoritmo de Cholesky Modificado	53
3.5.2	Os Algoritmos de Runge - Kutta	54
4	SIMULAÇÃO	57
4.1	DESCRIÇÃO DO CONTROLADOR EXEMPLO	57
4.2	ALGORITMOS DINÂMICOS	59
4.2.1	Entrada de Parâmetros (EP)	59

CONTEÚDO

	3
4.2.2	Entrada de Pontos da Trajetória (EQ) 59
4.2.3	Primeiro algoritmo para cálculo do MDD (MDD1) 59
4.2.4	Primeiro algoritmo para cálculo do MDL (MDL1) 60
4.2.5	Segundo algoritmo para cálculo do MDD (MDD2) 61
4.2.6	Segundo algoritmo para cálculo do MDL (MDL2) 61
4.2.7	Cálculo do MDI (MDI) 62
4.2.8	Algoritmo de Controle (AC) 63
4.2.9	Algoritmo do Simulador (AS) 63
5	RESULTADOS 65
5.1	DESCRIÇÃO SUCINTA DO PROGRAMA 65
5.2	PÊNDULO DUPLO 67
5.2.1	Equações Analíticas 67
5.2.2	Arquivo de Dados do Pêndulo Duplo 70
5.2.3	Resultados Obtidos 72
5.3	ROBÔ STANFORD 89
5.3.1	Arquivo de Dados do Robô Stanford 89
5.3.2	Resultados Obtidos 90
6	CONCLUSÕES 99
A	RELAÇÕES MATRICIAIS IMPORTANTES 101
A.1	DEFINIÇÕES 101
A.2	TEOREMAS 102
B	DEMONSTRAÇÃO DE RELAÇÕES CINEMÁTICAS 104
B.1	PRIMEIRO MODO 104
B.1.1	Caso A - Juntas de Rotação 104
B.1.2	Caso B - Juntas de Translação 106

B.2	SEGUNDO MODO	106
B.2.1	Caso A - Juntas de Rotação	107
B.2.2	Caso B - Juntas de Translação	107

Lista de Figuras

1.1	Classificação dos Manipuladores	9
2.1	Atribuição de Eixos pelas Regras de Denavit-Hartenberg	18
2.2	Parâmetros de Denavit-Hartenberg	19
4.1	Esquema de Controle Utilizado	58
5.1	Exemplo de Telas do Programa	66
5.2	Esquema do Pêndulo Duplo	68
5.3	TRAJREF - Posição de Referência - PD (Ex. 1)	73
5.4	TRAJREF - Velocidade de Referência - PD (Ex. 1)	73
5.5	TRAJREF - Aceleração de Referência - PD (Ex. 1)	74
5.6	FEEDFORWARD - Torque Inercial - PD (Ex. 1)	74
5.7	FEEDFORWARD - Torque Centrípeto + Coriolis - PD (Ex. 1)	75
5.8	FEEDFORWARD - Torque Gravitacional - PD (Ex. 1)	75
5.9	FEEDFORWARD - Torque Resultante - PD (Ex. 1)	76
5.10	MODINV - Torque Aplicado - PD (Ex. 1)	76
5.11	MODINV - Posição do Robô - PD (Ex. 1)	77
5.12	MODINV - Velocidade do Robô - PD (Ex. 1)	77
5.13	MODINV - Aceleração do Robô - PD (Ex. 1)	78
5.14	TRAJREF - Posição de Referência - PD (Ex. 2)	78
5.15	TRAJREF - Velocidade de Referência - PD (Ex. 2)	79
5.16	TRAJREF - Aceleração de Referência - PD (Ex. 2)	79

Agradecimentos

Desejo externar meus mais sinceros agradecimentos,

ao prof. Dr. José Jaime da Cruz, orientador deste trabalho, pelo incentivo, dedicação e pelas valiosas sugestões;

ao prof. Dr. Normonds Alens, pelos bons conselhos e pelo constante e decisivo apoio;

ao colega Ricardo Matone, pelas interessantes discussões sobre robótica.

5.17 FEEDFORWARD - Torque Inercial - PD (Ex. 2)	80
5.18 FEEDFORWARD - Torque Centrípeto + Coriolis - PD (Ex. 2)	80
5.19 FEEDFORWARD - Torque Gravitacional - PD (Ex. 2)	81
5.20 FEEDFORWARD - Torque Resultante - PD (Ex. 2)	81
5.21 MODINV - Torque Aplicado - PD (Ex. 2)	82
5.22 MODINV - Posição do Robô - PD (Ex. 2)	82
5.23 MODINV - Velocidade do Robô - PD (Ex. 2)	83
5.24 MODINV - Aceleração do Robô - PD (Ex. 2)	83
5.25 SIMULADOR - Torque Aplicado - PD (Ex. 3)	85
5.26 TRAJREF - Posição de Referência - PD (Ex. 3)	85
5.27 TRAJREF - Velocidade de Referência - PD (Ex. 3)	86
5.28 TRAJREF - Aceleração de Referência - PD (Ex. 3)	86
5.29 SIMULADOR - Posição do Robô - PD (Ex. 3)	87
5.30 SIMULADOR - Velocidade do Robô - PD (Ex. 3)	87
5.31 SIMULADOR - Deltaq - PD (Ex. 3)	88
5.32 SIMULADOR - Deltaqpto - PD (Ex. 3)	88
5.33 Robô Manipulador STANFORD	91
5.34 TRAJREF - Posição de Referência - STANFORD	92
5.35 TRAJREF - Velocidade de Referência - STANFORD	93
5.36 TRAJREF - Aceleração de Referência - STANFORD	94
5.37 FEEDFORWARD - Força Inercial - STANFORD	95
5.38 FEEDFORWARD - Força Centrípeta + Coriolis - STANFORD	96
5.39 FEEDFORWARD - Força Gravitacional - STANFORD	97
5.40 FEEDFORWARD - Força Resultante - STANFORD	98
B.1 Sistemas de Coordenadas	105

Capítulo 1

INTRODUÇÃO

1.1 ASPECTOS BÁSICOS

1.1.1 Definição de Robô

Entre as várias definições de robô manipulador existentes na literatura, transcrevemos a seguinte, por considerarmos a mais detalhada [GOR 84].

Um robô manipulador é um conjunto formado por:

- Uma **estrutura mecânica** que suporta o **efetuador** (ou órgão terminal) a situar.
- **Acionadores** que agem sobre a estrutura precedente para modificar a configuração do robô manipulador como um todo e, em consequência, a situação (posição e orientação) do efetuador.
- **Captore**s (ou **sensores**) diversos necessários ao controle, entre os quais distinguem-se os captores proprioceptivos que fornecem informação acerca do estado mecânico do manipulador e os captores exteroceptivos que fornecem informação acerca do estado do ambiente de trabalho do robô manipulador.
- Um **sistema de controle** que dirige os acionadores do robô manipulador a partir da definição do movimento a executar transmitido por um sistema de decisão e pelas informações fornecidas pelos captores proprioceptivos.
- Um **sistema de decisão**, que elabora o movimento do robô manipulador a partir da definição da tarefa a executar transmitida pelo operador por meio de um sistema de comunicação (As funções básicas de um sistema de decisão são a interpretação das mensagens emitidas pelo operador, o planejamento de trajetória e a geração de mensagens ao operador, assim como o gerenciamento de bases de dados).
- Um **sistema de comunicação**, que gera as mensagens a serem transmitidas entre o sistema de decisão e o operador, a partir de um terminal (alfanumérico ou gráfico).

Deve-se notar, entretanto [GOR 84], que os robôs manipuladores atuais, na sua grande maioria, não possuem o sistema de decisão, o que acarreta a perda da possibilidade de auto-adaptação ao ambiente.

1.1.2 Constituição e Classificação

A estrutura mecânica de um robô manipulador é um conjunto de ligamentos (corpos rígidos na modelagem adotada), unidos dois a dois por intermédio de juntas.

Os ligamentos e as juntas podem ser dispostos a partir de uma base fixa ou móvel segundo diversos modos, classificados como (Fig. 1.1):

- Estrutura de cadeia **aberta** (ou simples).
- Estrutura de cadeia **arborescente**.
- Estrutura de cadeia **fechada** (ou complexa).

Esta última se caracteriza pela existência de “loops” mecânicos, ou seja, de caminhos que permitam-nos partir de um ligamento e retornar a este mesmo ligamento passando sempre por juntas diferentes.

OBS: Definimos **grau de mobilidade** (n) de um robô como sendo o número de juntas deste. Em [GOR 84], o termo **grau de liberdade do robô** é reservado para o número de componentes independentes de velocidade cartesiana geradas pela juntas, o que nos parece ser uma nomenclatura mais apropriada.

1.1.3 Problemas Clássicos da Robótica

Os problemas clássicos da robótica são:¹

Cinemática Direta

Por **Cinemática Direta** entende-se o problema geométrico de computar a posição e orientação do efetuador relativas a um sistema de referência, a partir do posicionamento das juntas do robô.

Existe também o problema da **Cinemática Direta Instantânea** ou **Cinemática Diferencial Direta**, que consiste em obter a velocidade (linear e angular) do efetuador a partir das velocidades das juntas e da configuração instantânea do robô.

¹Esta seção está baseada em [CRA 86]

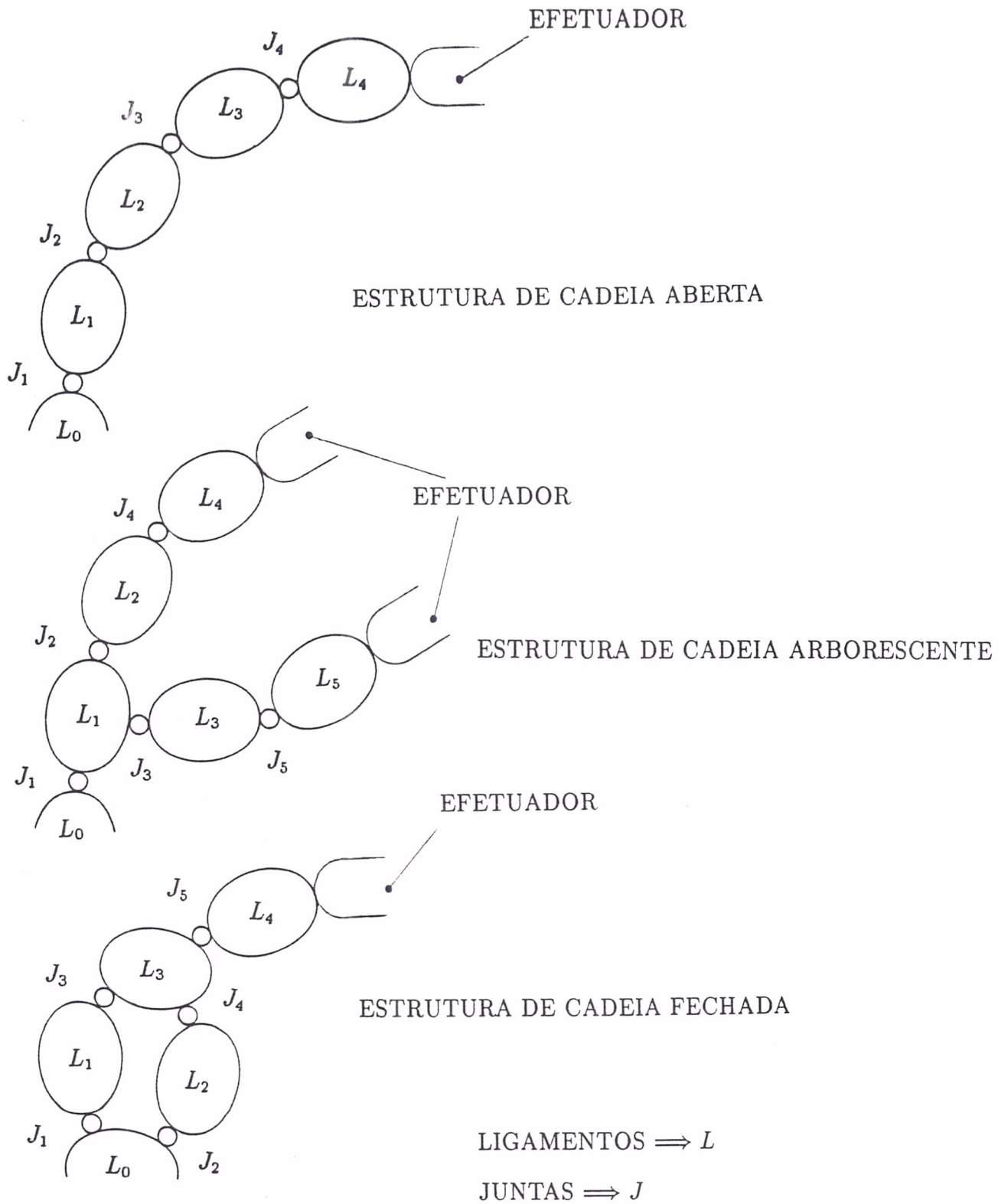


Figura 1.1: Classificação dos Manipuladores

Cinemática Inversa

Por **Cinemática Inversa** entende-se o problema de, dadas a posição e orientação do efetuador, encontrar conjuntos de ângulos de junta que nos levem a estas.

Analogamente à Cinemática Direta, existe também o problema da **Cinemática Inversa Instantânea** ou **Cinemática Diferencial Inversa**, que consiste em obter as velocidades das juntas a partir da velocidade (linear e angular) do efetuador e da configuração instantânea do robô.

Dinâmica

Mais adiante, definiremos e classificaremos os problemas de dinâmica, que serão o objetivo principal deste trabalho. Por enquanto, diremos apenas que a dinâmica de manipuladores é a parte da robótica que se ocupa do relacionamento entre os torques aplicados às juntas do robô e as posições, velocidades e acelerações destas.

Geração de Trajetória

O problema de sintetizar trajetórias (geralmente suaves) que as juntas devam seguir para que o robô trilhe certos pontos desejados é denominado **Geração de Trajetória**. Este problema, quando a trajetória é fornecida não a nível de posições de junta, mas a nível de posições cartesianas (Ex: seguir uma linha reta no espaço) é denominado **Geração de Trajetória Cartesiana**.

Em [GOR 84], temos uma subdivisão dos tipos de trajetória. Lá é utilizada uma terminologia mais genérica que em outros textos, sendo inclusive a palavra trajetória reservada para uma determinada classe de deslocamento do efetuador (veja a seguir). Utiliza-se o termo **operacional** para algo especificado em termos do espaço ordinário euclidiano em contraposição a **generalizado** para algo especificado em termos dos ângulos de junta.

O problema de geração de trajetórias é lá subdividido em três classes a saber:

TIPO I Neste problema, a evolução do efetuador é livre entre as situações inicial e final impostas.

TIPO II Neste problema, a evolução do efetuador é imposta entre as situações inicial e final, ou seja, o efetuador deve trilhar uma certa sequência de posições e orientações pré-especificada. A evolução do robô neste caso é denominada trajetória operacional.

TIPO III Aqui, o problema é exatamente igual ao tipo II com a ressalva de que a sequência mencionada é parametrizada no tempo. Este tipo de problema é denominado **movimento operacional**.

Controle de Posição

O objetivo do controle de posição é fazer com que o manipulador trilhe uma certa trajetória de referência dentro de determinada precisão. Devido às falhas de modelagem dinâmica do robô manipulador, mesmo que calculássemos os torques necessários para cada ponto da trajetória, o que é denominado “feedforward”, não conseguiríamos seguir perfeitamente esta trajetória. Por isto, faz-se necessário um esquema de controle, que permita automaticamente compensar a falha de modelagem, bem como suprimir perturbações de diversas origens tais como ruídos intervenientes no sistema, quantização (proveniente da utilização de controlador digital), etc...

Para realizar tal tarefa, sensores de posição e velocidade são monitorados pelo sistema de controle, que executa o cálculo (já tentando compensar os erros descritos anteriormente) dos torques que os acionadores do robô devem aplicar à estrutura mecânica.

Controle de Força

Algumas tarefas exigem o controle preciso da força de contato aplicada pelo efetuador. Quando o manipulador está movendo-se no espaço livre, apenas o controle de posição é importante, pois não há necessidade de reação contra superfícies. Porém quando o efetuador está tocando alguma superfície, utilizar apenas controle de posição poderia resultar em perda de contato ou em forças excessivas aplicadas à superfície. Como exemplo, se o efetuador estivesse cortando uma placa de vidro e não tivéssemos o controle de força, qualquer pequeno deslocamento poderia fraturar a placa. Deve-se mencionar também que, como os manipuladores raramente estão restritos pela superfície em todas as direções, faz sentido a utilização de um controle híbrido posição-força, com algumas direções controladas em posição e outras direções controladas em força.

Programação de Robôs

Os robôs diferenciam-se da automação fixa por serem programáveis, o que os torna flexíveis. Através da comunicação com um sistema de decisão, o robô pode “adaptar-se” a variações nas tarefas. Basicamente, devemos ser capazes de especificar as tarefas de um ou de múltiplos robôs através de uma interface homem-máquina adequada. Deve-se notar que linguagens específicas para robôs foram e estão sendo desenvolvidas. Como exemplos de linguagens ou de bibliotecas de procedimentos auxiliares na programação de robôs, podemos citar:

- PASRO
- VAL
- AL
- AR-BASIC

- AML
- RISE
- ROBOT-BASIC
- JARS-JPL

1.2 OBJETIVOS

Este trabalho trata do projeto e implementação de um simulador dinâmico que poderá ser utilizado com qualquer robô de cadeia aberta.

Este simulador nos permitirá obter as forças e torques necessários para o comando de robôs, servindo também como ferramenta auxiliar ao projeto de controladores avançados, pois, além de fornecer todos os coeficientes da equação dinâmica e de sua linearização, calculará vários parâmetros de sensibilidade que poderão ser úteis para análises mais detalhadas do mecanismo.

Também será implementado um módulo para computar, a partir de forças e torques aplicados como dados de entrada, qual a evolução do robô em termos de posição, velocidade e aceleração alcançados.

A entrada do programa será feita a partir de arquivos que descrevam o modelo e o robô (para simulação). Poder-se-á visualizar graficamente a evolução de diversas variáveis dinâmicas em operações que serão descritas posteriormente.

Forneceremos ainda alguns exemplos de utilização do sistema tais como: Simulação do Pêndulo Duplo, "Feedforward" do Robô STANFORD, etc...

Exemplificaremos o uso deste simulador através de gráficos ilustrativos.

1.3 COMPOSIÇÃO

O simulador constará basicamente de módulos para:

- Computar os coeficientes do modelo dinâmico direto (MDD), também conhecido como problema da dinâmica inversa [HOL 80,MEG 84], e que nos permitirá obter o valor das forças generalizadas f_i em função do valor das posições q_i , velocidades \dot{q}_i e acelerações \ddot{q}_i generalizadas.
- Computar os coeficientes do modelo dinâmico inverso (MDI), também conhecido como problema da dinâmica direta [HOL 80,MEG 84], e que nos permitirá obter o valor das acelerações generalizadas \ddot{q}_i em função das posições q_i , velocidades \dot{q}_i e forças generalizadas f_i .

- Computar os coeficientes do modelo dinâmico direto linearizado (MDL), em torno da trajetória de referência especificada, que nos permitirá obter o valor das perturbações nas forças generalizadas δf_i em função do valor das perturbações nas posições δq_i , velocidades $\delta \dot{q}_i$ e acelerações $\delta \ddot{q}_i$ generalizadas em um determinado ponto desta trajetória.
- Integrar os modelos acima através de um esquema de controle escolhido e verificar o seu desempenho. (Usaremos, como exemplo, um esquema de controle baseado em [GUO 89]).
- Realizar a entrada de parâmetros do robô a ser simulado e do modelo a ser utilizado no “feedforward”.
- Descrever a trajetória de referência, ou seja, o conjunto de pontos que o robô deverá atingir em determinados instantes (tempo simulado).
- Apresentar os resultados obtidos durante a simulação.

1.4 CONSIDERAÇÕES SOBRE EFICIÊNCIA

Na implementação do simulador, usaremos os algoritmos mais eficientes disponíveis atualmente, o que não é apenas uma questão de acelerar a execução do programa em alguns segundos, mas sim uma condição essencial, pois o problema de cálculo da dinâmica de robôs é muito complexo, resultando tipicamente em milhares de multiplicações e milhares de adições, isto sem contar o cálculo de senos e cossenos, as atribuições e as equações intermediárias envolvidas.

Além disso, como foi observado em [ANG 88]: “... *A falta de algoritmos para computação do modelo dinâmico que executassem em tempo real teve uma grande influência no projeto dos manipuladores atualmente existentes, resultando em sistemas (os comercialmente disponíveis) com falta de destreza e fraco desempenho ...*”

1.5 APLICAÇÕES

Algumas aplicações para estes modelos são:

- O MDD pode ser utilizado, por exemplo, para computar o “feedforward” e para a simulação dinâmica de robôs. No caso de obtermos os coeficientes da equação dinâmica em forma fechada, podemos utilizá-lo também no auxílio ao projeto de controladores avançados. O planejamento de trajetórias visando mínimo tempo de percurso é também realizado com base neste modelo.
- O MDI serve para simular um determinado robô, ou seja, dadas as posições q_i e velocidades \dot{q}_i iniciais do robô, e as forças generalizadas f_i em cada instante, podemos

calcular as acelerações generalizadas \ddot{q}_i e posteriormente integrá-las, obtendo assim a trajetória seguida.

- O MDL, por sua vez, possibilita a aplicação da teoria dos sistemas lineares ao problema de controle, podendo ser utilizado, por exemplo, para: [LI 89]
 1. Projetar controladores avançados, que seriam muito mais difíceis de realizar com base no MDD, pois neste temos equações diferenciais não lineares e altamente acopladas.
 2. Estudar propriedades cinemáticas e dinâmicas do robô, através do modelo de pequenos sinais obtido, ou seja, computar a variação da posição, velocidade e aceleração originadas por pequenas variações das forças e torques, ou vice-versa.
 3. Resolver problemas de regulação ótima (trilhar uma trajetória de referência) para manipuladores, pois, em geral, precisamos escrever as equações na forma canônica de estados, neste caso.

1.6 CONTRIBUIÇÕES

Como contribuições desta dissertação, podemos citar:

- Um método de dedução completa (ou seja, a partir dos parâmetros geométricos e dinâmicos dos robôs) das equações dinâmicas de robôs de cadeia aberta mais limpo e direto que o método normalmente utilizado (apresentado em [PAU 81]).
- A utilização do método de Cholesky para resolução de sistemas lineares com modificações, o que resultou em um algoritmo mais rápido que o convencional, pois evitamos o cálculo de raízes quadradas, que é uma operação relativamente demorada.
- Um método vetorial e muito rápido para dedução das derivadas parciais de T_p , que está descrito no apêndice.
- A utilização integrada de algoritmos sofisticados, tais como o de equações diferenciais utilizando passo variável (adaptativo), o de cálculo do modelo dinâmico e de sua linearização, o de controle dinâmico e de módulos de entrada-saída e gerador de gráficos, em um único programa.
- A descoberta e correção de alguns erros nos artigos e livros utilizados.

1.7 ESTRUTURA DO TRABALHO

Neste capítulo, foram descritos os aspectos gerais do trabalho, bem como definimos uma série de termos, que serão extensivamente utilizados daqui para a frente. No capítulo 2, abordamos o problema da obtenção do MDD e deduzimos várias relações entre os coeficientes do referido modelo. Estes tópicos serão de fundamental importância ao desenvolvimento de algoritmos eficientes, o que será feito no capítulo 3. No capítulo 4, descrevemos o esquema de controle que será utilizado como exemplo. Também neste mesmo capítulo, delineamos detalhadamente os algoritmos dinâmicos utilizados. O acompanhamento destes algoritmos permite-nos verificar, de forma clara, quais as relações que serão efetivamente utilizadas na implementação em computador do simulador dinâmico. No capítulo 5, exemplificamos através de gráficos alguns dos resultados que podem ser obtidos do simulador. Finalmente, no capítulo 6, tecemos as considerações finais acerca deste trabalho. Escrevemos também 2 apêndices, onde estão colocadas algumas definições e relações acessórias, porém imprescindíveis à compreensão do texto.

Capítulo 2

MODELAGEM DINÂMICA

2.1 PARÂMETROS DE DENAVIT-HARTENBERG

Para descrever as relações geométricas entre ligamentos adjacentes de um robô, é conveniente fixar a cada ligamento um sistema de coordenadas. Um conjunto de regras para isto foi proposta por **J. Denavit** e **R.S. Hartenberg** [ASA 85], sendo muito utilizado em robótica. A chamada notação (ou convenção, ou representação) de Denavit-Hartenberg estabelece as seguintes regras (vide Fig. 2.1 e Fig. 2.2):

- Os ligamentos do robô são numerados consecutivamente de 0 a n , onde o ligamento 0 corresponde à base do robô.
- As juntas do robô são numeradas de tal modo que a junta i seja a junta entre o ligamento $i - 1$ e o ligamento i .
- Um sistema **ortogonal** de coordenadas é fixado ao ligamento i , tomando o número i , e tendo os eixos fixados do seguinte modo:¹
 - z_i é fixado sobre o eixo da junta $i + 1$, podendo ter o sentido escolhido arbitrariamente.
 - x_i é fixado sobre a perpendicular comum entre z_{i-1} e z_i , apontando de z_{i-1} a z_i .²
 - y_i é o produto vetorial de z_i por x_i e completa o triedro ortogonal.
- A posição relativa de dois sistemas de coordenadas (afixados aos respectivos ligamentos) pode ser completamente especificada pelos seguintes parâmetros: (parâmetros de **Denavit-Hartenberg**)

¹Em geral, nos textos em francês [GOR 84,MEG 84,FER 84], a numeração dos sistemas começa de 1, ou seja, o sistema 1 está fixado ao ligamento 0.

²No caso de paralelismo entre z_{i-1} e z_i , ou seja, da não unicidade da perpendicular comum, é convenção podermos escolher qualquer uma delas.

- l_i é a distância (em módulo) entre as origens dos sistemas $i - 1$ e i , medida sobre o eixo x_i .
- d_i é a distância (algébrica) do eixo x_{i-1} ao eixo x_i , sendo medida sobre o eixo z_{i-1} .
- α_i é o ângulo (algébrico) entre o eixo z_{i-1} e o eixo z_i , sendo medido de acordo com o eixo x_i .
- θ_i é o ângulo (algébrico) entre o eixo x_{i-1} e o eixo x_i , sendo medido de acordo com o eixo z_{i-1} .

Se a junta i for de rotação (translação), $\theta_i(d_i)$ é o parâmetro variável, e é normalmente simbolizado por q_i (tanto para junta de rotação como translação). No capítulo anterior, já utilizamos q_i em tal contexto.

NOTA IMPORTANTE: Alguns livros não se preocupam muito com a questão do sinal dos parâmetros. No entanto, o sinal destes é de importância fundamental, já que nas matrizes homogêneas formadas com estes parâmetros, as grandezas entram com sinal.

2.2 ALGUNS FATOS DA CINEMÁTICA

2.2.1 Matrizes Homogêneas

Sendo A_i a matriz de transformação homogênea³ do sistema i em relação ao sistema $(i - 1)$ e T_i a matriz de transformação homogênea do sistema i em relação ao sistema de base (ou sistema 0), podemos obter [PAU 81]:

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & \vdots & l_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & \vdots & l_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & \vdots & d_i \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad (2.1)$$

Se denotarmos por:

R_i a matriz de rotação 3 x 3 do sistema i relativamente ao sistema de base.

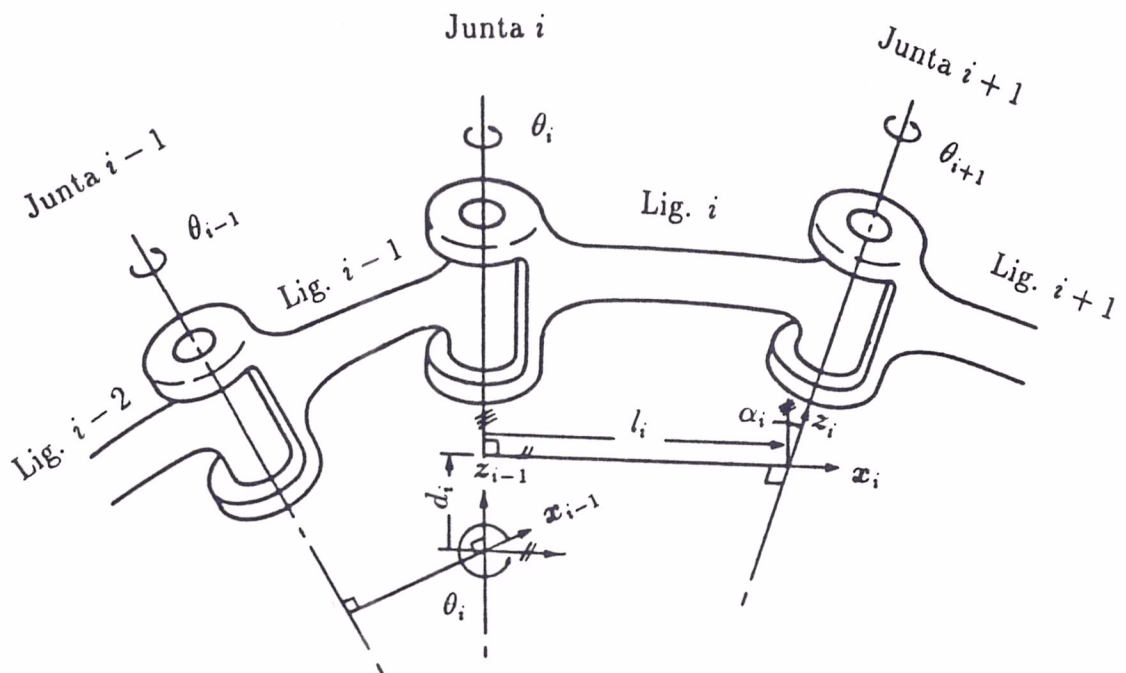
${}^{i-1}R_i$ a matriz de rotação 3 x 3 do sistema i relativamente ao sistema $(i - 1)$.

p_i o vetor posição 3 x 1 da origem do sistema i relativamente ao sistema de base.

${}^{i-1}p_i$ o vetor posição 3 x 1 da origem do sistema i relativamente ao sistema $(i - 1)$.

³Lembrar que o 4o. componente de um vetor homogêneo livre vale 0 enquanto que de um vetor homogêneo aplicado (no nosso caso, vetores descrevendo pontos do espaço) vale 1.

JUNTA DE ROTAÇÃO



JUNTA PRISMÁTICA

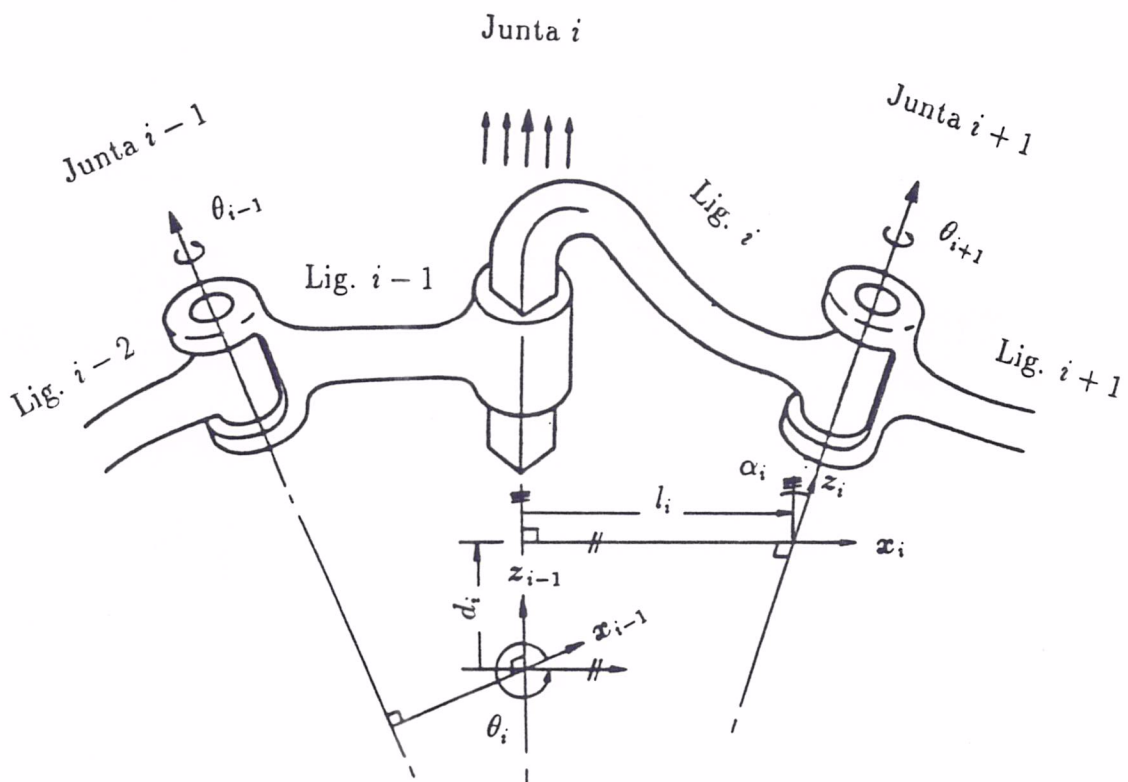


Figura 2.1: Atribuição de Eixos pelas Regras de Denavit-Hartenberg

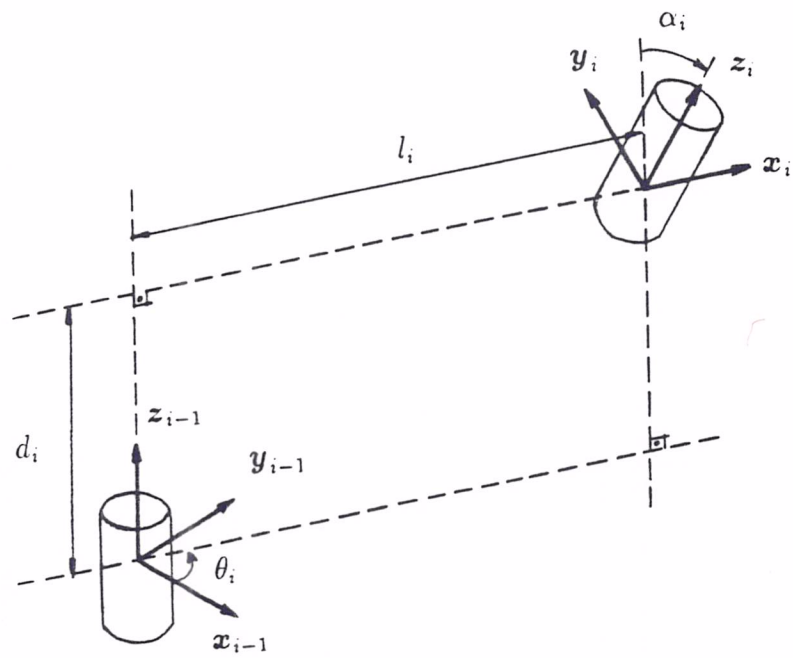


Figura 2.2: Parâmetros de Denavit-Hartenberg

teremos:

$$A_i = \begin{bmatrix} {}^{i-1}R_i & \vdots & {}^{i-1}p_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 1 \end{bmatrix}$$

bem como:

$$T_i = \begin{bmatrix} R_i & \vdots & p_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 1 \end{bmatrix} = \begin{bmatrix} n_i & \vdots & o_i & \vdots & a_i & \vdots & p_i \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 1 \end{bmatrix} = A_1 \cdots A_i \quad (2.2)$$

Como R_i é ortonormal, sua inversa é igual à transposta e T_i^{-1} fica:

$$T_i^{-1} = \begin{bmatrix} R_i^T & \vdots & -R_i^T p_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 1 \end{bmatrix}$$

Além disso, da relação $T_i = T_{i-1}A_i$, podemos concluir que:

$$R_i = R_{i-1} {}^{i-1}R_i \quad (2.3)$$

$$p_i = p_{i-1} + R_{i-1} {}^{i-1}p_i \quad (2.4)$$

com:⁴

$$R_0 = I_3$$

$$p_0 = 0$$

2.2.2 Simplificações das Derivadas Parciais

Sendo:⁵

$$\Delta_i \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{\delta}_i & \vdots & \lambda_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix}$$

com:

$$\delta_i = \begin{cases} a_{i-1} & \text{se a junta } i \text{ for de rotação} \\ 0 & \text{se a junta } i \text{ for de translação} \end{cases} \quad (2.5)$$

$$\lambda_i = \begin{cases} p_{i-1} \wedge a_{i-1} & \text{se a junta } i \text{ for de rotação} \\ a_{i-1} & \text{se a junta } i \text{ for de translação} \end{cases} \quad (2.6)$$

⁴O símbolo I_3 denota a matriz identidade de ordem 3.

⁵ $\tilde{\delta}_i$ é a matriz do pré-produto vetorial de δ_i (veja o apêndice A)

pode-se mostrar (veja Apêndice B) que:

$$\frac{\partial T_p}{\partial q_i} = \begin{cases} 0 & \text{se } i > p \\ \Delta_i T_p & \text{caso contrário} \end{cases} \quad (2.7)$$

$$\frac{\partial^2 T_p}{\partial q_i \partial q_j} = \begin{cases} 0 & \text{se } i > p \text{ ou } j > p \\ \Delta_i \Delta_j T_p & \text{caso contrário} \end{cases} \quad (2.8)$$

$$\frac{\partial^3 T_p}{\partial q_i \partial q_j \partial q_k} = \begin{cases} 0 & \text{se } i > p \text{ ou } j > p \text{ ou } k > p \\ \Delta_i \Delta_j \Delta_k T_p & \text{caso contrário} \end{cases} \quad (2.9)$$

onde temos sempre ($i \leq j \leq k$). Isto não é problema, pois a ordem de derivação de T_p não altera o resultado. Portanto, podemos sempre derivar seguindo a ordem crescente de índices.

2.3 OBTENÇÃO DO MODELO DINÂMICO

Muita pesquisa tem sido feita atualmente no tema da geração de modelos dinâmicos, como pode ser verificado em [HOL 80,MEG 84,FER 91,CHG 88,CHE 89], dentre outros.

Entretanto, como bem observa Ferreira [FER 91], é interessante notar que: “... *Existem diversos formalismos para a obtenção do modelo dinâmico de um robô manipulador, contudo a maioria deles só tem valor formal, pois não se prestam à implementação em computador, gerando, caso implementados, um modelo com um número muito grande de operações ...*”

Neste trabalho, para a obtenção do modelo dinâmico, escolhemos a formulação Lagrangiana pelas seguintes razões:

1. Permite obter as equações dinâmicas de sistemas de grande complexidade da maneira mais rápida possível [PAU 81].
2. Entre os métodos existentes, dos quais os mais usados são “Newton-Euler”⁶ e “Lagrange-Euler”, o que mais se presta ao projeto de sistemas de controle é o de “Lagrange-Euler” pois precisamos obter um conjunto de equações em forma fechada, ou melhor, precisamos dos valores dos coeficientes da equação dinâmica 2.22.

De acordo com [HOL 80], a formulação Lagrangiana padrão do modelamento dinâmico, empregando matrizes 4 x 4, foi utilizada inicialmente por Uicker, em sua tese de doutorado, para solucionar um problema mais geral que o de manipuladores de cadeia aberta. Esta abordagem foi posteriormente adaptada para o caso específico destes por Kahn.

A seguir, apresentaremos duas maneiras de se deduzir a equação do MDD para robôs de cadeia aberta, bem como ressaltaremos propriedades dos termos desta equação.

2.3.1 Velocidade de um Ponto

Tomando-se o ligamento i (fixado previamente), podemos descrever a posição de um ponto qualquer em relação à base por:

$$\rho = T_i {}^i \rho$$

onde ρ é o vetor posição homogêneo deste ponto, sendo que o índice **superior esquerdo** indica o sistema no qual estamos escrevendo o vetor (quando este índice for 0 podemos suprimi-lo).

⁶Embora o número de computações do algoritmo de “Newton - Euler” seja baixo, ele é um algoritmo recursivo e não fornece as equações na forma que desejamos.

Portanto, lembrando que \mathbf{T}_i só depende de (q_1, \dots, q_i) :

$$\dot{\rho} = \sum_{j=1}^i \left(\frac{\partial \mathbf{T}_i}{\partial q_j} \dot{q}_j \right) {}^i \rho \quad (2.10)$$

Conseqüentemente, indicando por \cdot o produto escalar e por tr a operação de traço de uma matriz, vem:⁷

$$\dot{\rho}^2 \stackrel{\text{def}}{=} \dot{\rho} \cdot \dot{\rho} = \text{tr} (\dot{\rho} \dot{\rho}^T)$$

Substituindo $\dot{\rho}$ de (2.10), temos:

$$\dot{\rho}^2 = \text{tr} \left[\sum_{j=1}^i \frac{\partial \mathbf{T}_i}{\partial q_j} \dot{q}_j {}^i \rho \sum_{k=1}^i \left(\frac{\partial \mathbf{T}_i}{\partial q_k} \dot{q}_k {}^i \rho \right)^T \right] = \text{tr} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial \mathbf{T}_i}{\partial q_j} {}^i \rho {}^i \rho^T \frac{\partial \mathbf{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right]$$

Vimos que surgiu ${}^i \rho {}^i \rho^T$ em consequência do uso da operação tr . Este agrupamento é conveniente, pois com ele conseguiremos (veja a seguir) exprimir as energias do sistema em função das massas e dos momentos de inércia dos ligamentos do robô.

Além disso, daqui já podemos constatar uma propriedade muito interessante e que viabiliza a aplicação, ao problema de dinâmica, de muitos métodos atuais de estimação de parâmetros, que é a linearidade do MDD nos parâmetros dinâmicos (veja a dedução abaixo).

2.3.2 Cálculo da Energia Cinética de um Ligamento

Tomando-se o ligamento i fixado anteriormente e simbolizando a energia cinética deste ligamento por K_i , podemos escrever:

$$dK_i = \frac{1}{2} \text{tr} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial \mathbf{T}_i}{\partial q_j} ({}^i \rho \, dm \, {}^i \rho^T) \frac{\partial \mathbf{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right]$$

Portanto:

$$K_i = \int_i dK_i = \frac{1}{2} \int_i \text{tr} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial \mathbf{T}_i}{\partial q_j} ({}^i \rho \, dm \, {}^i \rho^T) \frac{\partial \mathbf{T}_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right]$$

onde \int_i designa a integral realizada em todo o ligamento i .

⁷Deve-se notar que quando derivamos ρ , anulamos o 4o. componente de $\dot{\rho}$, que valia 1 no vetor homogêneo ρ .

Ou, permutando as somatórias com a integral,

$$K_i = \frac{1}{2} \text{tr} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \left(\int_i \rho \, dm \right) \frac{\partial T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (2.11)$$

A integral na fórmula 2.11 é denominada Matriz de Inércia Homogênea ou Matriz de Pseudo-Inércia, que denotaremos por \mathbf{J}_i .

Sendo:

$$I_{xx} = \int (y^2 + z^2) \, dm$$

$$I_{yy} = \int (x^2 + z^2) \, dm$$

$$I_{zz} = \int (x^2 + y^2) \, dm$$

$$I_{xy} = \int xy \, dm$$

$$I_{xz} = \int xz \, dm$$

$$I_{yz} = \int yz \, dm$$

$$m\bar{x} = \int x \, dm$$

$$m\bar{y} = \int y \, dm$$

$$m\bar{z} = \int z \, dm$$

podemos escrever:

$$\mathbf{J}_i = \begin{bmatrix} \frac{-I_{ixx} + I_{iyy} + I_{izz}}{2} & I_{ixy} & I_{ixz} & m_i \bar{x}_i \\ I_{ixy} & \frac{I_{ixx} - I_{iyy} + I_{izz}}{2} & I_{iyz} & m_i \bar{y}_i \\ I_{ixz} & I_{iyz} & \frac{I_{ixx} + I_{iyy} - I_{izz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

onde o índice i indica que os parâmetros são calculados com referência àquele ligamento. Note que \mathbf{J}_i é uma matriz simétrica.

Portanto, 2.11 fica:

$$K_i = \frac{1}{2} \text{tr} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \mathbf{J}_i \frac{\partial T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (2.12)$$

2.3.3 Cálculo da Energia Potencial de um Ligamento

Sendo γ o vetor homogêneo que descreve a aceleração da gravidade, $\bar{\rho}$ a posição do centro de massa⁸ de um corpo qualquer em relação a um referencial inercial e denotando a energia potencial por P , podemos escrever:

$$P_{\text{corpo}} = -m\gamma \cdot \bar{\rho}_{\text{corpo}}$$

Portanto, se ${}^j\bar{\rho}_i$ designar a posição do centro de massa do ligamento i em relação ao sistema j , resulta para o ligamento i :

$$P_i = -m_i\gamma^T T_i {}^i\bar{\rho}_i \quad (2.13)$$

2.3.4 Lagrangiano do Sistema

Cálculo da Energia Cinética Total K

Lembrando que a operação tr comuta com as somatórias, temos:

$$K = \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \frac{1}{2} \text{tr} \left[\frac{\partial T_i}{\partial q_j} J_i \frac{\partial T_i^T}{\partial q_k} \right] \dot{q}_j \dot{q}_k \quad (2.14)$$

Note que o termo entre colchetes só depende do vetor posição generalizada \mathbf{q} do manipulador.

Cálculo da Energia Potencial Total P

Temos:

$$P = - \sum_{i=1}^n m_i \gamma^T T_i {}^i\bar{\rho}_i \quad (2.15)$$

Cálculo do Lagrangiano do Sistema

O Lagrangiano do sistema é dado por:

$$L = K - P$$

⁸A barra designará sempre centro de massa.

Portanto, de 2.14 e 2.15:

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{tr} \left[\frac{\partial \mathbf{T}_i}{\partial q_j} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_k} \right] \dot{q}_j \dot{q}_k + \sum_{i=1}^n m_i \gamma^T \mathbf{T}_i \bar{\rho}_i \quad (2.16)$$

2.3.5 Equações Dinâmicas

Para sistemas materiais holônomos (aqueles contendo somente vínculos geométricos ou cinemáticos integráveis) a vínculos perfeitos [FRA 90], pode-se mostrar que vale a equação de Lagrange. Sendo este o caso aqui, podemos obter agora o modelo dinâmico do manipulador, utilizando esta equação, que é:

$$f_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (2.17)$$

onde f_i representa a força generalizada agente na junta i .

Vamos agora calcular os termos necessários a 2.17 a partir de 2.16. Então, temos:

$$\frac{\partial L}{\partial \dot{q}_p} = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^i \text{tr} \left[\frac{\partial \mathbf{T}_i}{\partial q_p} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_k} \right] \dot{q}_k + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \text{tr} \left[\frac{\partial \mathbf{T}_i}{\partial q_j} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_p} \right] \dot{q}_j$$

Utilizando a propriedade $\text{tr}(A^T) = \text{tr}(A)$ na primeira somatória, trocando j por k na segunda somatória, e observando que $\partial \mathbf{T}_i / \partial q_p = 0$ para $i < p$ (pois \mathbf{T}_i só depende de q_1, \dots, q_i), obtemos:

$$\frac{\partial L}{\partial \dot{q}_p} = \sum_{i=p}^n \sum_{k=1}^i \text{tr} \left[\frac{\partial \mathbf{T}_i}{\partial q_k} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_p} \right] \dot{q}_k$$

Portanto:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} &= \sum_{i=p}^n \sum_{k=1}^i \text{tr} \left[\frac{\partial \mathbf{T}_i}{\partial q_k} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_p} \right] \ddot{q}_k \\ &+ \sum_{i=p}^n \sum_{k=1}^i \sum_{s=1}^i \text{tr} \left[\frac{\partial^2 \mathbf{T}_i}{\partial q_k \partial q_s} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_p} \right] \dot{q}_k \dot{q}_s \\ &+ \sum_{i=p}^n \sum_{k=1}^i \sum_{s=1}^i \text{tr} \left[\frac{\partial^2 \mathbf{T}_i}{\partial q_p \partial q_s} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_k} \right] \dot{q}_k \dot{q}_s \end{aligned}$$

Por outro lado:

$$\begin{aligned} \frac{\partial L}{\partial q_p} &= \frac{1}{2} \sum_{i=p}^n \sum_{j=1}^i \sum_{k=1}^i \text{tr} \left[\frac{\partial^2 \mathbf{T}_i}{\partial q_j \partial q_p} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_k} \right] \dot{q}_j \dot{q}_k \\ &+ \frac{1}{2} \sum_{i=p}^n \sum_{j=1}^i \sum_{k=1}^i \text{tr} \left[\frac{\partial^2 \mathbf{T}_i}{\partial q_k \partial q_p} \mathbf{J}_i \frac{\partial \mathbf{T}_i^T}{\partial q_j} \right] \dot{q}_j \dot{q}_k \end{aligned}$$

$$+ \sum_{i=p}^n m_i \gamma^T \frac{\partial T_i}{\partial q_p} {}^i \bar{\rho}_i$$

Trocando j por k na segunda somatória acima, e depois trocando j por s na equação resultante, obtemos:

$$\begin{aligned} \frac{\partial L}{\partial q_p} &= \sum_{i=p}^n \sum_{s=1}^i \sum_{k=1}^i \text{tr} \left[\frac{\partial^2 T_i}{\partial q_p \partial q_s} J_i \frac{\partial T_i^T}{\partial q_k} \right] \dot{q}_s \dot{q}_k \\ &+ \sum_{i=p}^n m_i \gamma^T \frac{\partial T_i}{\partial q_p} {}^i \bar{\rho}_i \end{aligned}$$

Então, 2.17 fica:

$$\begin{aligned} f_p &= \sum_{i=p}^n \sum_{k=1}^i \text{tr} \left[\frac{\partial T_i}{\partial q_k} J_i \frac{\partial T_i^T}{\partial q_p} \right] \ddot{q}_k \\ &+ \sum_{i=p}^n \sum_{k=1}^i \sum_{s=1}^i \text{tr} \left[\frac{\partial^2 T_i}{\partial q_k \partial q_s} J_i \frac{\partial T_i^T}{\partial q_p} \right] \dot{q}_k \dot{q}_s \\ &- \sum_{i=p}^n m_i \gamma^T \frac{\partial T_i}{\partial q_p} {}^i \bar{\rho}_i \end{aligned}$$

Finalmente, para chegarmos na forma apresentada em [HOL 80], basta trocarmos p, i por i, j respectivamente, obtendo:

$$\begin{aligned} f_i &= \sum_{j=i}^n \sum_{k=1}^j \text{tr} \left[\frac{\partial T_j}{\partial q_k} J_j \frac{\partial T_j^T}{\partial q_i} \right] \ddot{q}_k \\ &+ \sum_{j=i}^n \sum_{k=1}^j \sum_{s=1}^j \text{tr} \left[\frac{\partial^2 T_j}{\partial q_k \partial q_s} J_j \frac{\partial T_j^T}{\partial q_i} \right] \dot{q}_k \dot{q}_s \\ &- \sum_{j=i}^n m_j \gamma^T \frac{\partial T_j}{\partial q_i} {}^j \bar{\rho}_j \end{aligned} \quad (2.18)$$

Definindo:

$$D_{ij} \stackrel{\text{def}}{=} \sum_{p=\max(i,j)}^n \text{tr} \left[\frac{\partial T_p}{\partial q_i} J_p \frac{\partial T_p^T}{\partial q_j} \right] \quad (2.19)$$

$$D_{ijk} \stackrel{\text{def}}{=} \sum_{p=\max(i,j,k)}^n \text{tr} \left[\frac{\partial T_p}{\partial q_i} J_p \frac{\partial^2 T_p^T}{\partial q_j \partial q_k} \right] \quad (2.20)$$

$$D_i \stackrel{\text{def}}{=} - \sum_{p=i}^n m_p \gamma^T \frac{\partial T_p}{\partial q_i} {}^p \bar{\rho}_p \quad (2.21)$$

podemos reescrever 2.18 como:

$$f_i = \sum_{j=1}^n D_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (2.22)$$

A forma 2.22 é mais adequada ao cálculo de f_i . Seus coeficientes têm denominações particulares que são listadas abaixo.

D_{ii} é a inércia efetiva da junta i .

D_{ij} é a inércia de acoplamento entre as juntas i e j .

D_{ijj} é o termo centrípeto (pois multiplica um fator do tipo \dot{q}_j^2) da junta i originário da velocidade da junta j .

D_{ijk} é o termo de Coriolis (pois multiplica um fator do tipo $\dot{q}_j \dot{q}_k$) da junta i originário das velocidades das juntas j e k .

D_i é a parcela de força generalizada, devida à gravidade, agente na junta i .

2.4 UMA DEMONSTRAÇÃO ALTERNATIVA E OUTRAS RELAÇÕES

2.4.1 Relações entre Coeficientes Dinâmicos - I

Das equações 2.19, 2.20 e 2.21 acima, é imediato que D_{ij}, D_{ijk} e D_i só dependem do vetor posição generalizada. Além disso, concluímos que:

$$D_{ij} = D_{ji} \quad (2.23)$$

$$D_{ijk} = D_{ikj} \quad (2.24)$$

2.4.2 A Demonstração Alternativa

Uma forma mais rápida e clara de se obter 2.20 (expressão dos D_{ijk}), e que também nos fornece relações entre os D_{ij} e os D_{ijk} é esquematizada abaixo.

A energia cinética 2.14 pode ser escrita, usando 2.19, como:

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n D_{jk} \dot{q}_j \dot{q}_k \quad (2.25)$$

OBS: Da relação acima, podemos deduzir [FRA 90] que a matriz $\mathbf{Y} \stackrel{\text{def}}{=} [D_{ij}]$, como matriz da parte quadrática em velocidade generalizada da energia cinética de um sistema, deve ser definida positiva (se escolhermos boas coordenadas, ou seja, havendo correspondência biunívoca entre coordenadas e configuração, o que é o caso).

Continuando, podemos escrever o Lagrangiano como:

$$L = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n D_{jk} \dot{q}_j \dot{q}_k - P \quad (2.26)$$

Então, 2.17 torna-se:

$$f_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial D_{ik}}{\partial q_j} - \frac{1}{2} \frac{\partial D_{jk}}{\partial q_i} \right) \dot{q}_j \dot{q}_k + \frac{\partial P}{\partial q_i}$$

ou, levando em conta a simetria na segunda somatória acima:

$$f_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{j=1}^n \sum_{k=1}^n \frac{1}{2} \left(\frac{\partial D_{ik}}{\partial q_j} + \frac{\partial D_{ij}}{\partial q_k} - \frac{\partial D_{jk}}{\partial q_i} \right) \dot{q}_j \dot{q}_k + \frac{\partial P}{\partial q_i}$$

o que nos leva a:

$$D_{ijk} = \frac{1}{2} \left(\frac{\partial D_{ik}}{\partial q_j} + \frac{\partial D_{ij}}{\partial q_k} - \frac{\partial D_{jk}}{\partial q_i} \right) \quad (2.27)$$

ou seja, os D_{ijk} são símbolos de Christoffel de primeira espécie, muito conhecidos da Análise Tensorial.

Note que, a partir da relação acima e dos D_{ij} que conhecemos (vide 2.19), podemos obter os D_{ijk} diretamente por derivação, o que nos leva a “outro” método de obtenção do modelo dinâmico. Os livros que adotam esta abordagem, em geral até fornecem 2.27, mas não chegam a deduzir os coeficientes dinâmicos na forma das equações 2.19, 2.20 e 2.21.

Da relação 2.27, podemos deduzir que:

$$D_{ijk} + D_{kji} = \frac{\partial D_{ik}}{\partial q_j} \quad (2.28)$$

2.4.3 Relações Entre Coeficientes Dinâmicos - II

Como o coeficiente dinâmico D_{ik} não depende dos ângulos de junta iguais ou anteriores à junta m (com $m = \min(i, k)$), podemos escrever:

$$\frac{\partial D_{ik}}{\partial q_j} = 0 \quad \text{se } j \leq i, k \quad (2.29)$$

Isto nos fornece duas relações, que são:

$$D_{ijk} = -D_{kji} \quad \text{se } j \leq i, k \quad (2.30)$$

$$D_{iji} = 0 \quad \text{se } j \leq i \quad (2.31)$$

2.4.4 Cálculos Necessários

Utilizando as relações 2.23, 2.24, 2.30 e 2.31, percebe-se que só precisamos calcular:

$$D_{ij} \quad \text{para } i \leq j \quad (2.32)$$

$$D_{ijk} \quad \text{para } i < k, j \leq k \quad (2.33)$$

2.5 FORMAS MATRICIAIS

É usual encontrarmos a equação dinâmica 2.22 expressa na seguinte forma: [MEG 84, LI 88]

$$\mathbf{f} = \mathbf{Y}(\mathbf{q}) \ddot{\mathbf{q}} + 2\mathbf{U}(\mathbf{q}) \dot{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{V}(\mathbf{q}) \dot{\mathbf{q}}^2 + \mathbf{e}(\mathbf{q}) \quad (2.34)$$

onde, levando em conta 2.23, 2.24, 2.30 e 2.31, podemos escrever:

$$\mathbf{Y}(\mathbf{q}) = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{12} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ D_{1n} & D_{2n} & \cdots & D_{nn} \end{bmatrix}$$

$$\mathbf{U} = [\mathbf{U}_1(\mathbf{q}); \mathbf{U}_2(\mathbf{q}); \cdots ; \mathbf{U}_{n-1}(\mathbf{q})]$$

$$\mathbf{U}_i(\mathbf{q}) = \begin{bmatrix} D_{1ii+1} & D_{1ii+2} & \cdots & D_{1in} \\ D_{2ii+1} & D_{2ii+2} & \cdots & D_{2in} \\ \vdots & \vdots & \cdots & \vdots \\ D_{iii+1} & D_{iii+2} & \cdots & D_{iin} \\ 0 & D_{i+1ii+2} & \cdots & D_{i+1in} \\ -D_{i+1ii+2} & 0 & \cdots & D_{i+2in} \\ \vdots & \vdots & \cdots & \vdots \\ -D_{i+1in} & -D_{i+2in} & \cdots & 0 \end{bmatrix}$$

$$\mathbf{V}(\mathbf{q}) = \begin{bmatrix} 0 & D_{122} & \cdots & D_{1nn} \\ -D_{112} & 0 & \cdots & D_{2nn} \\ \vdots & \vdots & \cdots & \vdots \\ -D_{11n} & -D_{22n} & \cdots & 0 \end{bmatrix}$$

$$\mathbf{e}(\mathbf{q}) = [D_1 \ D_2 \ \cdots \ D_n]^T$$

$$\ddot{\mathbf{q}} = [\ddot{q}_1 \ \ddot{q}_2 \ \cdots \ \ddot{q}_n]^T$$

$$\dot{\mathbf{q}}\dot{\mathbf{q}} = [\dot{q}_1\dot{q}_2 \ \dot{q}_1\dot{q}_3 \ \cdots \ \dot{q}_1\dot{q}_n \ \dot{q}_2\dot{q}_3 \ \dot{q}_2\dot{q}_4 \ \cdots \ \dot{q}_2\dot{q}_n \ \cdots \ \cdots \ \dot{q}_{n-1}\dot{q}_n]^T$$

$$\dot{\mathbf{q}}^2 = [\dot{q}_1^2 \ \dot{q}_2^2 \ \cdots \ \dot{q}_n^2]^T$$

Ou seja:

\mathbf{Y} é a matriz ($n \times n$) de inércia do manipulador (positiva definida).

\mathbf{U}_i é a submatriz ($n \times (n - i)$) de Coriolis (Note que nesta submatriz, da linha $i + 1$ à linha n , temos uma parte anti-simétrica e de diagonal nula).

U é a matriz ($n \times n(n-1)/2$) de Coriolis do manipulador.

V é a matriz ($n \times n$) centrífuga do manipulador.

$\ddot{\mathbf{q}}$ é o vetor ($n \times 1$) das acelerações de junta.

$\dot{\mathbf{q}}^2$ é o vetor ($n \times 1$) dos quadrados das velocidades de junta.

$\dot{\mathbf{q}}\dot{\mathbf{q}}$ é o vetor ($n(n-1)/2 \times 1$) dos termos quadráticos em velocidade de juntas diferentes.

Podemos reescrever também 2.22 em forma matricial como:

$$\mathbf{f} = \mathbf{Y}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{e}(\mathbf{q}) \quad (2.35)$$

onde incorporamos os coeficientes de velocidade em uma única matriz \mathbf{C} .

Podemos também escrever 2.35 como:

$$\mathbf{f} = \mathbf{Y}(\mathbf{q}) \ddot{\mathbf{q}} + \boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}}) \quad (2.36)$$

onde $\boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}})$ agrupa $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{e}(\mathbf{q})$.

2.6 O CÁLCULO DAS FORÇAS GENERALIZADAS

Para computar as forças generalizadas, usando apenas os coeficientes dinâmicos que calculamos (isto é, só os coeficientes não redundantes), devemos escrever a equação 2.22 numa forma mais apropriada. É isto que faremos a seguir.

Podemos reescrever 2.22, usando 2.23 e 2.24, como:

$$f_i = \sum_{j=1}^{i-1} D_{ji} \ddot{q}_j + \sum_{j=i}^n D_{ij} \ddot{q}_j + 2 \sum_{j=1}^{i-1} \sum_{k=j+1}^n D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=1}^n D_{ijj} \dot{q}_j^2 + D_i \quad (2.37)$$

Podemos desmembrar a terceira somatória do membro direito de 2.37 em:

$$\sum_{j=1}^{i-1} \sum_{k=j+1}^n D_{ijk} \dot{q}_j \dot{q}_k = \sum_{j=1}^{i-1} \sum_{k=j+1}^i D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=1}^{i-1} \sum_{k=i+1}^n D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=i}^{n-1} \sum_{k=j+1}^n D_{ijk} \dot{q}_j \dot{q}_k \quad (2.38)$$

Mas, usando 2.30 e 2.31, a primeira somatória do membro direito de 2.38 fica:

$$\sum_{j=1}^{i-1} \sum_{k=j+1}^i D_{ijk} \dot{q}_j \dot{q}_k = \sum_{j=1}^{i-2} \sum_{k=j+1}^{i-1} D_{ijk} \dot{q}_j \dot{q}_k = - \sum_{j=1}^{i-2} \sum_{k=j+1}^{i-1} D_{kji} \dot{q}_j \dot{q}_k \quad (2.39)$$

Por outro lado, usando 2.30 e 2.31, também obtemos:

$$\sum_{j=1}^n D_{ijj} \dot{q}_j^2 = \sum_{j=1}^{i-1} D_{ijj} \dot{q}_j^2 + \sum_{j=i+1}^n D_{ijj} \dot{q}_j^2 = - \sum_{j=1}^{i-1} D_{jji} \dot{q}_j^2 + \sum_{j=i+1}^n D_{ijj} \dot{q}_j^2 \quad (2.40)$$

Então, substituindo 2.38, 2.39 e 2.40 em 2.37 obtemos a fórmula final para o cálculo dos f_i :

$$\begin{aligned} f_i &= \sum_{j=1}^{i-1} D_{ji} \ddot{q}_j + \sum_{j=i}^n D_{ij} \ddot{q}_j - \sum_{j=1}^{i-1} D_{jji} \dot{q}_j^2 + \sum_{j=i+1}^n D_{ijj} \dot{q}_j^2 + D_i \\ &+ 2 \left[- \sum_{j=1}^{i-2} \sum_{k=j+1}^{i-1} D_{kji} \dot{q}_j \dot{q}_k + \sum_{j=1}^{i-1} \sum_{k=i+1}^n D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=i}^{n-1} \sum_{k=j+1}^n D_{ijk} \dot{q}_j \dot{q}_k \right] \quad (2.41) \end{aligned}$$

Desmembrando 2.41 em componentes, obtemos:

$$f_i = f_i^a + f_i^c + f_i^g \quad (2.42)$$

onde:

$$f_i^a = \sum_{j=1}^{i-1} D_{ji} \ddot{q}_j + \sum_{j=i}^n D_{ij} \ddot{q}_j \quad (2.43)$$

$$\begin{aligned}
 f_i^c = & -\sum_{j=1}^{i-1} D_{jji} \dot{q}_j^2 + \sum_{j=i+1}^n D_{ijj} \dot{q}_j^2 \\
 & + 2 \left[-\sum_{j=1}^{i-2} \sum_{k=j+1}^{i-1} D_{kji} \dot{q}_j \dot{q}_k + \sum_{j=1}^{i-1} \sum_{k=i+1}^n D_{ijk} \dot{q}_j \dot{q}_k + \sum_{j=i}^{n-1} \sum_{k=j+1}^n D_{ijk} \dot{q}_j \dot{q}_k \right] \quad (2.44)
 \end{aligned}$$

$$f_i^g = D_i \quad (2.45)$$

Capítulo 3

O CÁLCULO EFICIENTE DOS MODELOS

3.1 HISTÓRICO

De acordo com Hollerbach [HOL 80], as formulações mais antigas para cálculo do MDD, que são baseadas **diretamente** nos resultados anteriores (2.19, 2.20 e 2.21), devido ao grande número de cálculos redundantes, foram muito ineficientes.

Para se ter uma idéia, basta notar que para um robô com 6 juntas de rotação eram necessárias 66271 multiplicações e 51548 adições para calcular cada ponto da trajetória, ou seja, os torques dificilmente poderiam ser computados em tempo real.¹

Foram, então, propostos diversos esquemas para superar esta dificuldade, a saber:

- Simplificar a dinâmica, ignorando termos convenientes do modelo. Nesta abordagem, ignoram-se principalmente termos que contenham a velocidade, ou melhor, os termos centrífugos e de Coriolis (responsáveis pela maior parte da carga computacional). Isto é feito em [PAU 81] e justificado argumentando-se que “... as equações dinâmicas completas são muito complexas, contém milhares de termos, mas a simplificação é rápida e muito fácil de se fazer. As equações resultantes são simples, podem ser computadas em tempo real e nos dão grande percepção acerca do sistema a ser controlado ...”. Além disso, lá é dito que podemos desconsiderar os torques dependentes de velocidade por serem tipicamente pequenos quando comparados a outros torques do sistema. Afirma-se também que há muitos deles para se computar e que são importantes somente quando o manipulador esteja trabalhando em alta velocidade quando, então, a precisão seria de pouca importância.

¹Segundo Li [LI 88, LI 89], temos para os algoritmos referidos a ligamento (veja mais adiante): $3n^3/2 + 14n^2 + 9n/2 - 4$ multiplicações e $7n^3/6 + 10n^2 + 115n/3 - 10$ adições e, incluindo a linearização, $9n^4/8 + 19n^3/6 + 431n^2/24 - 541n/12 + 81$ multiplicações e $5n^4/4 + 11n^3/4 + 83n^2/6 - 53n/6 + 52$ adições o que nos fornece, para $n = 6$ juntas, 851 multiplicações e 832 adições e, incluindo a linearização, 2599 multiplicações e 2711 adições.

Contra esta argumentação, temos as seguintes:

[HOL 80] “... ignorar forças de Coriolis e centrífugas só funciona bem em baixas velocidades; em altas velocidades, as forças de Coriolis e centrífugas são um grande componente da dinâmica. Os erros no cálculo do torque ignorando estas forças não podem ser corrigidos com realimentação, por causa da excessiva demanda dos torques de correção ...”

[LI 88] “... o sistema de controle de robô projetado pelo modelo simplificado de Lagrange-Euler torna o controle acurado de posição do robô muito difícil, senão impossível ...”

[MEG 84] “... esta simplificação só é possível quando as velocidades são muito baixas. J. Luh et al. acharam erros grandes nos valores e mesmo nos sinais das forças generalizadas calculadas a partir do modelo simplificado de Bejczy quando as velocidades eram altas ...”

Pelos comentários acima, percebe-se que a tendência moderna é utilizar melhores algoritmos, que permitam calcular o modelo dinâmico sem negligenciar a parte de Coriolis e centrífuga.

- Tabular a solução, ou parte dela. Estas soluções tabulares são, no fundo, a realização do amplamente conhecido compromisso memória - tempo de processamento. Os pontos não tabulados seriam devidamente interpolados.

Lembrando as equações 2.22, 2.34, 2.35, 2.36 e que D_i , D_{ij} e D_{ijk} só dependem da posição \mathbf{q} do robô, temos como exemplos de possibilidades de tabulação:

- Tabular \mathbf{f} contra $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. Isto leva a um gasto enorme de espaço de memória.
- Eliminar $\ddot{\mathbf{q}}$ da tabela (utilizamos 2.36). Ou seja, tabulamos $\mathbf{Y}(\mathbf{q})$ e $\boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}})$.
- Tabular os D_i , D_{ij} e D_{ijk} , que só dependem de \mathbf{q} , calculando \mathbf{f} por 2.22.

A grande vantagem desta abordagem é que além do número de operações ser bastante reduzido, não necessita-se do cálculo dos senos e cossenos dos ângulos (que são, no caso, pré-computados).

As desvantagens são:

1. Gasta-se muito tempo calculando a tabela.
2. As tabelas necessitam ser inteiramente recalculadas se algum parâmetro dinâmico muda. Por exemplo, se o robô pegar algum objeto, já teremos alteração nestes parâmetros. Esta pode ser considerada a pior das desvantagens, no sentido de inviabilizar o método.
3. Quantidades muito grandes de memória são necessárias para criar-se uma tabela suficientemente refinada. Por exemplo [CRA 86], para o esquema mais dispendioso de tabulação (o primeiro esquema citado) e para apenas 10 valores discretos para quantizar os $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ de um manipulador de 6 juntas o número de células de quantização será de $(10 \times 10 \times 10)^6$. Mas cada célula contém 6 valores de torque. Portanto serão necessárias 6×10^{18} células na tabela. Se

cada número real (um por célula) gastar 6 bytes, teremos um total de 36×10^{18} bytes!!!.

4. Para cada ponto $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ dado devemos interpolar entre pontos próximos que estejam na tabela.
- Obter os modelos dinâmicos sob forma simbólica, utilizando-os diretamente nos cálculos. Esta abordagem é descrita em [MEG 84], sendo muito interessante, permitindo obter um modelo eficiente em cada **caso particular** de robô. Em [FER 91] temos uma breve descrição de um gerador de modelos simbólico para robôs de cadeia aberta (GMROB).
 - Utilizar um algoritmo numérico eficiente, que organize os cálculos de maneira a eliminar grande parte das redundâncias que estão presentes num cálculo direto de D_i , D_{ij} e D_{ijk} por 2.19, 2.20 e 2.21 acima. Esta é a abordagem que seguiremos neste trabalho, sendo válida para qualquer manipulador de cadeia aberta.

No MDI, devemos calcular termos que fazem parte também do MDD. Assim sendo, um algoritmo eficiente para o MDD pode ser adaptado ao MDI. (Para ser mais específico, devemos calcular $\mathbf{Y}(\mathbf{q})$ e $\boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}})$, os quais também deverão ser calculados no MDD). Adicionalmente, aproveitaremos o fato da matriz \mathbf{Y} ser simétrica definida positiva na obtenção de um melhor algoritmo para o MDI.

Quanto ao MDL, segundo Li [LI 89], entre todos os métodos existentes atualmente, o algoritmo proposto por ele é o mais eficiente. Também aqui, soluções tabulares podem ser tentadas, mas sofrem dos mesmos problemas que foram expostos para o caso do MDD. Por isso, o algoritmo de Li [LI 89] será adotado neste trabalho.

3.2 O ALGORITMO PARA O MDD

Neste capítulo, estaremos utilizando a seguinte notação:

- \mathbf{g} é a parte 3 x 1 do vetor homogêneo γ .
- $\bar{\mathbf{r}}_i$ é a parte 3 x 1 do vetor homogêneo ${}^i\bar{\rho}_i$, o que quer dizer que o índice de $\bar{\mathbf{r}}$ estará definindo tanto o ligamento em que calculamos o vetor posição do **centro de massa** como o sistema em que este vetor será expresso (que no caso é o sistema do próprio ligamento).
- $\bar{\mathbf{I}}_i$ é a matriz de inércia 3 x 3 do ligamento i calculada com relação ao **centro de massa** deste, utilizando o sistema i .

Ou seja,

$$\gamma = \begin{bmatrix} \mathbf{g} \\ \dots \\ 0 \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \\ g_z \\ 0 \end{bmatrix}$$

$${}^i\bar{\rho}_i = \begin{bmatrix} \bar{\mathbf{r}}_i \\ \dots \\ 1 \end{bmatrix} = \begin{bmatrix} r_{ix} \\ r_{iy} \\ r_{iz} \\ 1 \end{bmatrix}$$

$$\bar{\mathbf{I}}_i = \begin{bmatrix} \frac{-\bar{I}_{ixx} + \bar{I}_{iyy} + \bar{I}_{izz}}{2} & \bar{I}_{ixy} & \bar{I}_{ixz} \\ \bar{I}_{ixy} & \frac{\bar{I}_{ixx} - \bar{I}_{iyy} + \bar{I}_{izz}}{2} & \bar{I}_{iyz} \\ \bar{I}_{ixz} & \bar{I}_{iyz} & \frac{\bar{I}_{ixx} + \bar{I}_{iyy} - \bar{I}_{izz}}{2} \end{bmatrix}$$

No algoritmo do MDD, usaremos muito o particionamento de matrizes visando a eliminação de redundâncias nos cálculos. Assim sendo, utilizando o **Teorema dos Eixos Paralelos** [GIA 78], podemos escrever a matriz de inércia homogênea como:

$$\mathbf{J}_i = \begin{bmatrix} \bar{\mathbf{I}}_i + m_i \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^T & \vdots & m_i \bar{\mathbf{r}}_i \\ \dots & \dots & \dots \\ m_i \bar{\mathbf{r}}_i^T & \vdots & m_i \end{bmatrix}$$

Substituindo 2.7 em 2.19 e 2.21 e também 2.7 e 2.8 em 2.20, podemos escrever:

$$\begin{aligned} D_{ij} &= \sum_{p=j}^n \text{tr} \left[\Delta_i \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \Delta_j^T \right] \\ &= \text{tr} \left[\Delta_i \left(\sum_{p=j}^n \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \right) \Delta_j^T \right] \end{aligned} \quad (3.1)$$

$$\begin{aligned}
 D_{ijk} &= \sum_{p=k}^n \text{tr} \left[\Delta_i T_p J_p T_p^T \Delta_k^T \Delta_j^T \right] \\
 &= \text{tr} \left[\Delta_i \left(\sum_{p=k}^n T_p J_p T_p^T \right) \Delta_k^T \Delta_j^T \right]
 \end{aligned} \tag{3.2}$$

$$\begin{aligned}
 D_i &= - \sum_{p=i}^n m_p \left[\mathbf{g}^T \quad ; \quad 0 \right] \Delta_i T_p \begin{bmatrix} \bar{r}_p \\ \cdots \\ 1 \end{bmatrix} \\
 &= - \left[\mathbf{g}^T \quad ; \quad 0 \right] \Delta_i \sum_{p=i}^n m_p T_p \begin{bmatrix} \bar{r}_p \\ \cdots \\ 1 \end{bmatrix}
 \end{aligned} \tag{3.3}$$

Definindo:

$$\begin{bmatrix} \mathbf{L}_i & ; & \mathbf{h}_i \\ \cdots & \cdots & \cdots \\ \mathbf{h}_i^T & ; & M_i \end{bmatrix} \stackrel{\text{def}}{=} \sum_{p=i}^n T_p J_p T_p^T \tag{3.4}$$

com \mathbf{L}_i 3 x 3, \mathbf{h}_i 3 x 1, M_i 1 x 1 e desenvolvendo o membro direito, resulta:

$$\mathbf{L}_i = \sum_{j=i}^n \left[\mathbf{R}_j \bar{\mathbf{I}}_j \mathbf{R}_j^T + m_j (\mathbf{R}_j \bar{\mathbf{r}}_j + \mathbf{p}_j) (\mathbf{R}_j \bar{\mathbf{r}}_j + \mathbf{p}_j)^T \right] \tag{3.5}$$

$$\mathbf{h}_i = \sum_{j=i}^n m_j (\mathbf{R}_j \bar{\mathbf{r}}_j + \mathbf{p}_j) \tag{3.6}$$

$$M_i = \sum_{j=i}^n m_j \tag{3.7}$$

$$\tag{3.8}$$

Donde resulta que a matriz \mathbf{L}_i é simétrica.

Notando que 3.4 implica também:

$$\sum_{p=i}^n m_p T_p \begin{bmatrix} \bar{r}_p \\ \cdots \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_i \\ \cdots \\ M_i \end{bmatrix}$$

podemos escrever 3.1, 3.2 e 3.3 como:

$$\begin{aligned}
 D_{ij} &= \text{tr} \left\{ \begin{bmatrix} \tilde{\delta}_i & ; & \lambda_i \\ \cdots & \cdots & \cdots \\ 0 & ; & 0 \end{bmatrix} \begin{bmatrix} \mathbf{L}_j & ; & \mathbf{h}_j \\ \cdots & \cdots & \cdots \\ \mathbf{h}_j^T & ; & M_j \end{bmatrix} \begin{bmatrix} \tilde{\delta}_j^T & ; & 0 \\ \cdots & \cdots & \cdots \\ \lambda_j^T & ; & 0 \end{bmatrix} \right\} \\
 D_{ijk} &= \text{tr} \left\{ \begin{bmatrix} \tilde{\delta}_i & ; & \lambda_i \\ \cdots & \cdots & \cdots \\ 0 & ; & 0 \end{bmatrix} \begin{bmatrix} \mathbf{L}_k & ; & \mathbf{h}_k \\ \cdots & \cdots & \cdots \\ \mathbf{h}_k^T & ; & M_k \end{bmatrix} \begin{bmatrix} \tilde{\delta}_k^T & ; & 0 \\ \cdots & \cdots & \cdots \\ \lambda_k^T & ; & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_j^T & ; & 0 \\ \cdots & \cdots & \cdots \\ \lambda_j^T & ; & 0 \end{bmatrix} \right\}
 \end{aligned}$$

$$D_i = - \begin{bmatrix} \mathbf{g}^T & \vdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\delta}}_i & \vdots & \boldsymbol{\lambda}_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{h}_i \\ \cdots \\ M_i \end{bmatrix}$$

o que nos fornece:

$$\begin{aligned} D_{ij} &= \text{tr} \left\{ [\tilde{\boldsymbol{\delta}}_i \left(\mathbf{L}_j \tilde{\boldsymbol{\delta}}_j^T + \mathbf{h}_j \boldsymbol{\lambda}_j^T \right) + \boldsymbol{\lambda}_i \left(\mathbf{h}_j^T \tilde{\boldsymbol{\delta}}_j^T + M_j \boldsymbol{\lambda}_j^T \right)] \right\} \\ D_{ijk} &= \text{tr} \left\{ [\tilde{\boldsymbol{\delta}}_i \left(\mathbf{L}_k \tilde{\boldsymbol{\delta}}_k^T + \mathbf{h}_k \boldsymbol{\lambda}_k^T \right) + \boldsymbol{\lambda}_i \left(\mathbf{h}_k^T \tilde{\boldsymbol{\delta}}_k^T + M_k \boldsymbol{\lambda}_k^T \right)] \tilde{\boldsymbol{\delta}}_j^T \right\} \\ D_i &= -\mathbf{g}^T \left(\tilde{\boldsymbol{\delta}}_i \mathbf{h}_i + M_i \boldsymbol{\lambda}_i \right) \end{aligned}$$

Coloquemos, por definição:^{2 3}

$$\mathbf{U}_i \stackrel{\text{def}}{=} \overline{\overline{\mathbf{L}_i}} \quad (3.9)$$

$$\begin{aligned} \mathbf{V}_i &\stackrel{\text{def}}{=} \mathbf{L}_i \tilde{\boldsymbol{\delta}}_i^T + \mathbf{h}_i \boldsymbol{\lambda}_i^T \\ &= \left(\boldsymbol{\delta}_i \wedge \mathbf{L}_{i1} \quad \boldsymbol{\delta}_i \wedge \mathbf{L}_{i2} \quad \boldsymbol{\delta}_i \wedge \mathbf{L}_{i3} \right)^T + \mathbf{h}_i \boldsymbol{\lambda}_i^T \end{aligned} \quad (3.10)$$

$$\mathbf{W}_i \stackrel{\text{def}}{=} \overline{\overline{\mathbf{V}_i}} \quad (3.11)$$

Utilizando então A.1 e os teoremas A.2, A.6 e A.7 (no caso do D_{ij}) e também o teorema A.9 (no caso do D_{ijk}), obtemos o seguinte algoritmo para o cálculo dos coeficientes dinâmicos:

$$D_{ij} = \boldsymbol{\delta}_i^T (\mathbf{U}_j \boldsymbol{\delta}_j + \mathbf{h}_j \wedge \boldsymbol{\lambda}_j) + \boldsymbol{\lambda}_i^T (\boldsymbol{\delta}_j \wedge \mathbf{h}_j + M_j \boldsymbol{\lambda}_j) \quad (i \leq j) \quad (3.12)$$

$$D_{ijk} = \boldsymbol{\delta}_j^T [\mathbf{W}_k \boldsymbol{\delta}_i + (\boldsymbol{\delta}_k \wedge \mathbf{h}_k + M_k \boldsymbol{\lambda}_k) \wedge \boldsymbol{\lambda}_i] \quad (i < k, j \leq k) \quad (3.13)$$

$$D_i = -\mathbf{g}^T (\boldsymbol{\delta}_i \wedge \mathbf{h}_i + M_i \boldsymbol{\lambda}_i) \quad (3.14)$$

Se definirmos:

$$\boldsymbol{\phi}_i \stackrel{\text{def}}{=} \mathbf{U}_i \boldsymbol{\delta}_i + \mathbf{h}_i \wedge \boldsymbol{\lambda}_i \quad (3.15)$$

$$\boldsymbol{\psi}_i \stackrel{\text{def}}{=} \boldsymbol{\delta}_i \wedge \mathbf{h}_i + M_i \boldsymbol{\lambda}_i \quad (3.16)$$

podemos reescrever 3.12, 3.13, 3.14 em uma forma mais compacta como:

$$D_{ij} = \boldsymbol{\delta}_i^T \boldsymbol{\phi}_j + \boldsymbol{\lambda}_i^T \boldsymbol{\psi}_j \quad (i \leq j) \quad (3.17)$$

$$D_{ijk} = \boldsymbol{\delta}_j^T (\mathbf{W}_k \boldsymbol{\delta}_i + \boldsymbol{\psi}_k \wedge \boldsymbol{\lambda}_i) \quad (i < k, j \leq k) \quad (3.18)$$

$$D_i = -\mathbf{g}^T \boldsymbol{\psi}_i \quad (3.19)$$

² $\overline{\overline{\mathbf{A}}}$ denota o que denominamos matriz auxiliar de \mathbf{A} (veja o apêndice A).

³ \mathbf{L}_{ik} denota a k -ésima coluna da matriz \mathbf{L}_i .

Por outro lado, lembrando 3.5, 3.6 e 3.7, podemos obter o seguinte algoritmo recorrente:

$$\mathbf{L}_i = \mathbf{L}_{i+1} + \mathbf{R}_i \bar{\mathbf{I}}_i \mathbf{R}_i^T + m_i (\mathbf{R}_i \bar{\mathbf{r}}_i + \mathbf{p}_i) (\mathbf{R}_i \bar{\mathbf{r}}_i + \mathbf{p}_i)^T \quad (3.20)$$

$$\mathbf{h}_i = \mathbf{h}_{i+1} + m_i (\mathbf{R}_i \bar{\mathbf{r}}_i + \mathbf{p}_i) \quad (3.21)$$

$$M_i = M_{i+1} + m_i \quad (3.22)$$

com: $\mathbf{L}_{n+1} = 0$, $\mathbf{h}_{n+1} = 0$, $M_{n+1} = 0$.

3.3 O ALGORITMO PARA O MDL

Como:

$$f_i = f_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$$

a linearização de f_i em torno do ponto $(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0)$ fica:

$$\delta f_i = \sum_{j=1}^n \left[\left(\frac{\partial f_i}{\partial \ddot{q}_j} \right) \delta \ddot{q}_j + \left(\frac{\partial f_i}{\partial \dot{q}_j} \right) \delta \dot{q}_j + \left(\frac{\partial f_i}{\partial q_j} \right) \delta q_j \right] \quad (3.23)$$

onde as derivadas parciais são calculadas no referido ponto.

Se definirmos:

$$A_{ij} \stackrel{\text{def}}{=} \frac{\partial f_i}{\partial \ddot{q}_j} \quad B_{ij} \stackrel{\text{def}}{=} \frac{1}{2} \frac{\partial f_i}{\partial \dot{q}_j} \quad C_{ij} \stackrel{\text{def}}{=} \frac{\partial f_i}{\partial q_j} \quad (3.24)$$

podemos escrever 3.23 matricialmente como:

$$\delta \mathbf{f} = \mathbf{A} \delta \ddot{\mathbf{q}} + 2\mathbf{B} \delta \dot{\mathbf{q}} + \mathbf{C} \delta \mathbf{q} \quad (3.25)$$

onde então:

\mathbf{A} é a matriz de sensibilidade da força **inercial** em relação à **aceleração** generalizada.

\mathbf{B} é a matriz de sensibilidade das forças **centrípetas** e de **Coriolis** em relação à **velocidade** generalizada.

\mathbf{C} é a matriz de sensibilidade da força **generalizada** em relação à **posição** generalizada.

Podemos também escrever 3.25 na forma de estados:⁴

$$\begin{bmatrix} \delta \dot{\mathbf{q}} \\ \dots \\ \delta \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} 0 & \vdots & \mathbf{I}_n \\ \dots & \dots & \dots \\ -\mathbf{A}^{-1}\mathbf{C} & \vdots & -2\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \delta \mathbf{q} \\ \dots \\ \delta \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} 0 \\ \dots \\ \mathbf{A}^{-1} \end{bmatrix} \delta \mathbf{f}$$

Com base na equação 2.22, vamos efetuar o cálculo das referidas matrizes. Note que a expressão final destas deve estar baseada somente em coeficientes dinâmicos que foram necessários ao cálculo do vetor força generalizada.

Ou seja, usaremos extensivamente as relações 2.23, 2.24, 2.30 e 2.31. Em alguns casos, usaremos diretamente 2.41.

⁴O símbolo \mathbf{I}_n denota a matriz identidade de ordem n .

3.3.1 Cálculo de A

É imediato que:

$$A_{ij} = \frac{\partial f_i}{\partial \ddot{q}_j} = D_{ij}$$

o que implica a inversibilidade da matriz \mathbf{A} , já que a matriz \mathbf{Y} (dos D_{ij}) é simétrica definida positiva e portanto inversível.

Enfim:

$$A_{ij} = \begin{cases} D_{ij} & \text{se } (i \leq j) \\ A_{ji} & \text{se } (i > j) \end{cases} \quad (3.26)$$

3.3.2 Cálculo de B

Obtemos:

$$B_{ij} = \frac{1}{2} \frac{\partial f_i}{\partial \dot{q}_j} = \sum_{k=1}^n D_{ijk} \dot{q}_k$$

Usando 2.24, 2.30 e 2.31, obtemos:

$$B_{ij} = \begin{cases} \sum_{k=1}^j D_{ikj} \dot{q}_k + \sum_{k=j+1}^n D_{ijk} \dot{q}_k & \text{se } (i < j) \\ \sum_{k=i+1}^n D_{iik} \dot{q}_k & \text{se } (i = j) \\ - \sum_{k=1}^j D_{jkj+1} \dot{q}_k + \sum_{k=j+2}^n D_{j+1jk} \dot{q}_k & \text{se } (i = j + 1) \\ - \sum_{k=1}^j D_{jkj} \dot{q}_k - \sum_{k=j+1}^{i-1} D_{kji} \dot{q}_k + \sum_{k=i+1}^n D_{ijk} \dot{q}_k & \text{se } (i > j + 1) \end{cases} \quad (3.27)$$

3.3.3 Cálculo de C

Vamos subdividir o cálculo de C_{ij} nas três parcelas seguintes:

$$C_{ij} = \frac{\partial f_i}{\partial q_j} = C_{ij}^a + C_{ij}^c + C_{ij}^g \quad (3.28)$$

onde:

C_{ij}^a é a sensibilidade da força inercial em relação à posição generalizada.

C_{ij}^c é a sensibilidade das forças centrípeta e de Coriolis em relação à posição generalizada.

C_{ij}^g é a sensibilidade da força gravitacional em relação à posição generalizada.

Portanto, de 2.41, obtemos:

$$C_{ij}^a = \sum_{k=1}^{i-1} \frac{\partial D_{ki}}{\partial q_j} \ddot{q}_k + \sum_{k=i}^n \frac{\partial D_{ik}}{\partial q_j} \ddot{q}_k \quad (3.29)$$

$$C_{ij}^c = 2 \left[- \sum_{k=1}^{i-2} \sum_{s=k+1}^{i-1} \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=1}^{i-1} \sum_{s=i+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=i}^{n-1} \sum_{s=k+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right] \\ - \sum_{k=1}^{i-1} \frac{\partial D_{kki}}{\partial q_j} \dot{q}_k^2 + \sum_{k=i+1}^n \frac{\partial D_{ikk}}{\partial q_j} \dot{q}_k^2 \quad (3.30)$$

$$C_{ij}^g = \frac{\partial D_i}{\partial q_j} \quad (3.31)$$

Cálculo de C^a

Lembrando 2.28, podemos escrever:

$$C_{ij}^a = \sum_{k=1}^n (D_{ijk} + D_{kji}) \ddot{q}_k$$

o que implica:

$$C_{ij}^a = \begin{cases} \sum_{k=1}^{j-1} (D_{ikj} + D_{kij}) \ddot{q}_k + D_{ijj} \ddot{q}_j + \sum_{k=j+1}^n (D_{ijk} - D_{jik}) \ddot{q}_k & \text{se } (i < j) \\ \sum_{k=1}^{i-1} D_{kii} \ddot{q}_k & \text{se } (i = j) \\ \sum_{k=1}^{j-1} (D_{kji} - D_{jki}) \ddot{q}_k & \text{se } (i > j) \end{cases} \quad (3.32)$$

Cálculo de C^c

Com base em 3.30, podemos escrever:

$$C_{ij}^c = 2 \left\{ - \sum_{k=1}^{i-2} \sum_{s=k+1}^{i-1} \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=1}^{i-1} \left[\sum_{s=i+1}^j \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{s=j+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right] \right. \\ \left. + \sum_{k=i}^{j-1} \left[\sum_{s=k+1}^j \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{s=j+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right] + \sum_{k=j}^{n-1} \sum_{s=k+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right\} \\ - \sum_{k=1}^{i-1} \frac{\partial D_{kki}}{\partial q_j} \dot{q}_k^2 + \sum_{k=i+1}^j \frac{\partial D_{ikk}}{\partial q_j} \dot{q}_k^2 + \sum_{k=j+1}^n \frac{\partial D_{ikk}}{\partial q_j} \dot{q}_k^2 \quad \text{se } (i < j) \quad (3.33)$$

$$\begin{aligned}
 C_{ij}^c &= 2 \left[- \sum_{k=1}^{i-2} \sum_{s=k+1}^{i-1} \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=1}^{i-1} \sum_{s=i+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right. \\
 &+ \left. \sum_{k=i}^{n-1} \sum_{s=k+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right] - \sum_{k=1}^{i-1} \frac{\partial D_{kki}}{\partial q_j} \dot{q}_k^2 \\
 &+ \sum_{k=i+1}^n \frac{\partial D_{ikk}}{\partial q_j} \dot{q}_k^2 \quad \text{se } (i = j)
 \end{aligned} \tag{3.34}$$

$$\begin{aligned}
 C_{ij}^c &= 2 \left\{ - \sum_{k=1}^{j-1} \left[\sum_{s=k+1}^j \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{s=j+1}^{i-1} \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s \right] \right. \\
 &- \sum_{k=j}^{i-2} \sum_{s=k+1}^{i-1} \frac{\partial D_{ski}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=1}^{j-1} \sum_{s=i+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \\
 &+ \left. \sum_{k=j}^{i-1} \sum_{s=i+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s + \sum_{k=i}^{n-1} \sum_{s=k+1}^n \frac{\partial D_{iks}}{\partial q_j} \dot{q}_k \dot{q}_s \right\} - \sum_{k=1}^{j-1} \frac{\partial D_{kki}}{\partial q_j} \dot{q}_k^2 \\
 &- \sum_{k=j}^{i-1} \frac{\partial D_{kki}}{\partial q_j} \dot{q}_k^2 + \sum_{k=i+1}^n \frac{\partial D_{ikk}}{\partial q_j} \dot{q}_k^2 \quad \text{se } (i > j)
 \end{aligned} \tag{3.35}$$

Mas:

$$\begin{aligned}
 \frac{\partial D_{iks}}{\partial q_j} &= \sum_{p=\max(j,s)}^n \left[\text{tr} \left(\frac{\partial^2 \mathbf{T}_p}{\partial q_i \partial q_j} \mathbf{J}_p \frac{\partial^2 \mathbf{T}_p^T}{\partial q_k \partial q_s} \right) \right. \\
 &+ \left. \text{tr} \left(\frac{\partial \mathbf{T}_p}{\partial q_i} \mathbf{J}_p \frac{\partial^3 \mathbf{T}_p^T}{\partial q_j \partial q_k \partial q_s} \right) \right] \quad (i < s, k \leq s)
 \end{aligned} \tag{3.36}$$

Então, definindo:

$$\begin{aligned}
 a &\stackrel{\text{def}}{=} \min(i, j) \\
 b &\stackrel{\text{def}}{=} \max(i, j) \\
 c &\stackrel{\text{def}}{=} \min(j, k) \\
 d &\stackrel{\text{def}}{=} \min(s, \max(j, k)) \\
 e &\stackrel{\text{def}}{=} \max(j, s)
 \end{aligned}$$

podemos escrever 3.36 como:

$$\begin{aligned}
 \frac{\partial D_{iks}}{\partial q_j} &= \sum_{p=\max(j,s)}^n \left[\text{tr} \left(\Delta_a \Delta_b \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \Delta_s^T \Delta_k^T \right) + \text{tr} \left(\Delta_i \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \Delta_e^T \Delta_d^T \Delta_c^T \right) \right] \\
 &= \text{tr} \left[\Delta_a \Delta_b \left(\sum_{p=c}^n \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \right) \Delta_s^T \Delta_k^T \right] + \text{tr} \left[\Delta_i \left(\sum_{p=e}^n \mathbf{T}_p \mathbf{J}_p \mathbf{T}_p^T \right) \Delta_e^T \Delta_d^T \Delta_c^T \right]
 \end{aligned}$$

$$\begin{aligned}
 &= \text{tr} \left\{ \begin{bmatrix} \tilde{\delta}_a & \vdots & \lambda_a \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_b & \vdots & \lambda_b \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \right. \\
 &\times \left. \begin{bmatrix} L_e & \vdots & h_e \\ \cdots & \cdots & \cdots \\ h_e^T & \vdots & M_e \end{bmatrix} \begin{bmatrix} \tilde{\delta}_s^T & \vdots & 0 \\ \cdots & \cdots & \cdots \\ \lambda_s^T & \vdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_k^T & \vdots & 0 \\ \cdots & \cdots & \cdots \\ \lambda_k^T & \vdots & 0 \end{bmatrix} \right\} \\
 &+ \text{tr} \left\{ \begin{bmatrix} \tilde{\delta}_i & \vdots & \lambda_i \\ \cdots & \cdots & \cdots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} L_e & \vdots & h_e \\ \cdots & \cdots & \cdots \\ h_e^T & \vdots & M_e \end{bmatrix} \right. \\
 &\times \left. \begin{bmatrix} \tilde{\delta}_e^T & \vdots & 0 \\ \cdots & \cdots & \cdots \\ \lambda_e^T & \vdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_d^T & \vdots & 0 \\ \cdots & \cdots & \cdots \\ \lambda_d^T & \vdots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_c^T & \vdots & 0 \\ \cdots & \cdots & \cdots \\ \lambda_c^T & \vdots & 0 \end{bmatrix} \right\}
 \end{aligned}$$

Expandindo a equação acima, temos então:

$$\begin{aligned}
 \frac{\partial D_{iks}}{\partial q_j} &= \text{tr} \left\{ \tilde{\delta}_a \left[\tilde{\delta}_b \left(L_s \tilde{\delta}_s^T + h_s \lambda_s^T \right) + \lambda_b \left(\delta_s \wedge h_s + M_s \lambda_s \right)^T \right] \tilde{\delta}_k^T \right\} \\
 &+ \text{tr} \left\{ \left[\tilde{\delta}_i \left(L_s \tilde{\delta}_s^T + h_s \lambda_s^T \right) + \lambda_i \left(\delta_s \wedge h_s + M_s \lambda_s \right)^T \right] \tilde{\delta}_d^T \tilde{\delta}_c^T \right\} \\
 &\quad \text{se } (i < s, k \leq s, j \leq s)
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial D_{iks}}{\partial q_j} &= \text{tr} \left\{ \tilde{\delta}_i \left[\left(\tilde{\delta}_j L_j + \lambda_j h_j^T \right) \tilde{\delta}_s^T + \left(\delta_j \wedge h_j + M_j \lambda_j \right) \lambda_s^T \right] \tilde{\delta}_k^T \right\} \\
 &+ \text{tr} \left\{ \left[\tilde{\delta}_i \left(L_j \tilde{\delta}_j^T + h_j \lambda_j^T \right) + \lambda_i \left(\delta_j \wedge h_j + M_j \lambda_j \right)^T \right] \tilde{\delta}_s^T \tilde{\delta}_k^T \right\} \\
 &\quad \text{se } (i < s, k \leq s, j \geq s)
 \end{aligned}$$

Se definirmos:

$$\mathbf{X}_{ij} \stackrel{\text{def}}{=} \tilde{\delta}_i \left(L_j \tilde{\delta}_j^T + h_j \lambda_j^T \right) + \lambda_i \left(\delta_j \wedge h_j + M_j \lambda_j \right)^T \quad (i \leq j)$$

e lembrarmos que (3.10 e 3.16):

$$\begin{aligned}
 \mathbf{V}_j &\stackrel{\text{def}}{=} L_j \tilde{\delta}_j^T + h_j \lambda_j^T \\
 \boldsymbol{\psi}_j &\stackrel{\text{def}}{=} \delta_j \wedge h_j + M_j \lambda_j
 \end{aligned}$$

podemos escrever:

$$\mathbf{X}_{ij} = \left(\delta_i \wedge \mathbf{V}_{j1} \quad \delta_i \wedge \mathbf{V}_{j2} \quad \delta_i \wedge \mathbf{V}_{j3} \right) + \lambda_i \boldsymbol{\psi}_j^T \quad (i \leq j) \quad (3.37)$$

Sendo:

$$Y_{ij} \stackrel{\text{def}}{=} \overline{\overline{X_{ij}}} \quad (i \leq j) \quad (3.38)$$

definiremos também:

$$D_{ijk_s} \stackrel{\text{def}}{=} \text{tr}(\tilde{\delta}_i X_{j_s} \tilde{\delta}_k^T) = \delta_k^T Y_{j_s} \delta_i \quad (i \leq s, j \leq s, k \leq s) \quad (3.39)$$

OBS: Note que no artigo de Li [LI 89], aparentemente foi esquecido que, em uma somatória para cálculo de C_{ij}^c (veja algoritmo adiante), necessitamos de termos D_{ijk_s} em que o primeiro índice torna-se igual ao quarto índice. Isto pode ser (e foi) corrigido acima e no programa de computador que implementamos, onde calculamos os coeficientes necessários restantes. No mesmo artigo há alguns pequenos erros no apêndice, que provavelmente são de impressão pois, no texto principal, estes não aparecem.

Continuando, podemos escrever:

$$\begin{aligned} \frac{\partial D_{iks}}{\partial q_j} &= \text{tr}(\tilde{\delta}_a X_{b_s} \tilde{\delta}_k^T) + \text{tr}(X_{i_s} \tilde{\delta}_d^T \tilde{\delta}_c^T) \\ &= \text{tr}(\tilde{\delta}_a X_{b_s} \tilde{\delta}_k^T) + \text{tr}(\tilde{\delta}_c^T X_{i_s} \tilde{\delta}_d^T) \\ &= D_{abks} - D_{cids} \quad (i < s, j \leq s, k \leq s) \end{aligned}$$

$$\begin{aligned} \frac{\partial D_{iks}}{\partial q_j} &= \text{tr}(\tilde{\delta}_k X_{s_j} \tilde{\delta}_i^T) + \text{tr}(X_{i_j} \tilde{\delta}_s^T \tilde{\delta}_k^T) \\ &= D_{ksij} - D_{kij_s} \quad (i < s, k \leq s, j \geq s) \end{aligned}$$

Destas últimas relações, concluímos que:⁵

$$\frac{\partial D_{iks}}{\partial q_j} = \begin{cases} D_{ijk_s} - D_{jik_s} & (j \leq k \leq s) \\ D_{ijk_s} - D_{kij_s} & (k < j < s) \\ D_{ksij} - D_{kij_s} & (k \leq s \leq j) \end{cases} \quad (i \leq j) \quad (3.40)$$

$$\frac{\partial D_{iks}}{\partial q_j} = \begin{cases} 0 & (j \leq k \leq s) \\ D_{jjks} - D_{kjj_s} & (k < j < s) \end{cases} \quad (i = j) \quad (3.41)$$

$$\frac{\partial D_{iks}}{\partial q_j} = \begin{cases} 0 & (j \leq k \leq s) \\ D_{jik_s} - D_{kij_s} & (k < j < s) \end{cases} \quad (i > j) \quad (3.42)$$

⁵Note que só calcularemos as derivadas parciais acima para $i < s$, pois estas são as necessárias.

Então, podemos reescrever as expressões 3.33, 3.34 e 3.35 como:

$$C_{ij}^c = \begin{cases} 2 \left\{ \sum_{k=1}^{i-2} \sum_{s=k+1}^{i-1} (D_{ksij} - D_{kij}) \dot{q}_k \dot{q}_s + \right. \\ \left. \sum_{k=1}^{i-1} \left[\sum_{s=i+1}^j (D_{ksij} - D_{kij}) \dot{q}_k \dot{q}_s + \sum_{s=j+1}^n (D_{ijks} - D_{kij}) \dot{q}_k \dot{q}_s \right] + \right. \\ \left. \sum_{k=i}^{j-1} \left[\sum_{s=k+1}^j (D_{ksij} - D_{kij}) \dot{q}_k \dot{q}_s + \sum_{s=j+1}^n (D_{ijks} - D_{kij}) \dot{q}_k \dot{q}_s \right] + \right. \\ \left. \sum_{k=j}^{n-1} \sum_{s=k+1}^n (D_{ijks} - D_{jik}) \dot{q}_k \dot{q}_s \right\} + \sum_{k=1}^{i-1} (D_{kkij} - D_{kikj}) \dot{q}_k^2 + \\ \sum_{k=i+1}^j (D_{kkij} - D_{kikj}) \dot{q}_k^2 + \\ \sum_{k=j+1}^n (D_{ijkk} - D_{jik}) \dot{q}_k^2 \\ \text{se } (i < j) \\ \\ 2 \left[\sum_{k=1}^{i-2} \sum_{s=k+1}^{i-1} (D_{ksii} - D_{kisi}) \dot{q}_k \dot{q}_s + \sum_{k=1}^{i-1} \sum_{s=i+1}^n (D_{iiks} - D_{kii}) \dot{q}_k \dot{q}_s \right] + \\ \sum_{k=1}^{i-1} (D_{kkii} - D_{kiki}) \dot{q}_k^2 \\ \text{se } (i = j) \\ \\ 2 \left\{ \sum_{k=1}^{j-1} \left[\sum_{s=k+1}^j (D_{ksji} - D_{sjki}) \dot{q}_k \dot{q}_s + \sum_{s=j+1}^{i-1} (D_{ksji} - D_{jski}) \dot{q}_k \dot{q}_s \right] + \right. \\ \left. \sum_{k=1}^{j-1} \sum_{s=i+1}^n (D_{jik}) \dot{q}_k \dot{q}_s \right\} + \sum_{k=1}^{j-1} (D_{kkji} - D_{kjk}) \dot{q}_k^2 \\ \text{se } (i > j) \end{cases} \quad (3.43)$$

Cálculo de C^g

Derivando 2.21, obtemos:

$$C_{ij}^g = \frac{\partial D_i}{\partial q_j} = - \sum_{p=\max(i,j)}^n m_p \left[\mathbf{g}^T \quad ; \quad 0 \right] \frac{\partial^2 T_p}{\partial q_i \partial q_j} \begin{bmatrix} \bar{r}_p \\ \dots \\ 1 \end{bmatrix}$$

Desta relação, percebemos que C_{ij}^g é simétrica. Portanto calcularemos estes coeficientes somente para $i \leq j$.

Utilizando 2.8, obtemos:

$$\begin{aligned} C_{ij}^g &= - \sum_{p=j}^n m_p \left[\mathbf{g}^T \quad ; \quad 0 \right] \Delta_i \Delta_j T_p \begin{bmatrix} \bar{r}_p \\ \dots \\ 1 \end{bmatrix} \\ &= - \left[\mathbf{g}^T \quad ; \quad 0 \right] \Delta_i \Delta_j \sum_{p=j}^n m_p T_p \begin{bmatrix} \bar{r}_p \\ \dots \\ 1 \end{bmatrix} \\ &= - \left[\mathbf{g}^T \quad ; \quad 0 \right] \begin{bmatrix} \tilde{\delta}_i & \dots & \lambda_i \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \tilde{\delta}_j & \dots & \lambda_j \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{h}_j \\ \dots \\ M_j \end{bmatrix} \\ &= -\mathbf{g}^T [\delta_i \wedge (\delta_j \wedge \mathbf{h}_j + M_j \lambda_j)] \\ &= \delta_i^T (\mathbf{g} \wedge \psi_j) \quad \text{se } (i \leq j) \end{aligned} \quad (3.44)$$

Portanto:

$$C_{ij}^g = \begin{cases} \delta_i^T (g \wedge \psi_j) & \text{se } (i \leq j) \\ C_{ji}^g & \text{se } (i > j) \end{cases} \quad (3.45)$$

3.3.4 Simplificações

Analisando as expressões fornecidas anteriormente, podemos fazer as seguintes simplificações:

1. Se a junta i for de translação (o que implica $\lambda_i = 0$):

$$\phi_i = h_i \wedge \lambda_i \quad (3.46)$$

$$\psi_i = M_i \lambda_i \quad (3.47)$$

$$V_i = h_i \lambda_i^T \quad (3.48)$$

$$X_{ij} = \lambda_i \psi_j^T \quad (i \leq j) \quad (3.49)$$

$$D_{ij} = \lambda_i^T \psi_j \quad (i \leq j) \quad (3.50)$$

$$D_{ijk} = \delta_j^T (\psi_k \wedge \lambda_i) \quad (i < k, j \leq k) \quad (3.51)$$

$$D_{ijks} = 0 \quad (i < s, j \leq s, k \leq s) \quad (3.52)$$

$$C_{ij}^g = 0 \quad (i \leq j) \quad (3.53)$$

2. Se a junta j for de translação:

$$D_{ijk} = 0 \quad (i < k, j \leq k) \quad (3.54)$$

3. Se a junta k for de translação:

$$D_{ijks} = 0 \quad (i \leq s, j \leq s, k \leq s) \quad (3.55)$$

3.4 A UTILIZAÇÃO DOS SISTEMAS DE LIGAMENTO

É comum em robótica referenciar os cálculos aos sistemas de ligamento ao invés do sistema de base. Isto geralmente leva a um número um pouco menor de operações, pelo menos para o número de juntas que é utilizado normalmente (5, 6 ou 7). Assim sendo, colocaremos também aqui, as equações para este processo, que poderá ser utilizado na implementação de um segundo algoritmo para o MDD e o MDL do simulador.

Note que, nestas fórmulas, usamos a seguinte notação:

Para matrizes:

$${}^j A_i = R_j^T A_i R_j$$

Para vetores:

$${}^j \alpha_i = R_j^T \alpha_i$$

Podemos reescrever então as fórmulas 2.4, 2.5 e 2.6 como:

$${}^i p_i = \begin{cases} {}^{i-1} R_i^T {}^{i-1} p_{i-1} + \begin{bmatrix} l_i \\ d_i \sin(\alpha_i) \\ d_i \cos(\alpha_i) \end{bmatrix} & \text{se a junta } i \text{ for de rotação} \\ {}^{i-1} R_i^T ({}^{i-1} p_{i-1} + {}^{i-1} p_i) & \text{se a junta } i \text{ for de translação} \end{cases} \quad (3.56)$$

$${}^{i-1} \delta_i = \begin{cases} (0, 0, 1)^T & \text{se a junta } i \text{ for de rotação} \\ (0, 0, 0)^T & \text{se a junta } i \text{ for de translação} \end{cases} \quad (3.57)$$

$${}^{i-1} \lambda_i = \begin{cases} {}^{i-1} p_{i-1} \wedge {}^{i-1} \delta_i & \text{se a junta } i \text{ for de rotação} \\ (0, 0, 1)^T & \text{se a junta } i \text{ for de translação} \end{cases} \quad (3.58)$$

onde ${}^0 p_0 = 0$.

Também serão necessárias as seguintes relações recursivas em j :

$${}^j \delta_i = {}^{j-1} R_j^T {}^{j-1} \delta_i \quad (3.59)$$

$${}^j \lambda_i = {}^{j-1} R_j^T {}^{j-1} \lambda_i \quad (3.60)$$

Reescreveremos:⁶

$${}^{i-1} U_i = \overline{{}^{i-1} L_i} \quad (3.61)$$

$${}^{i-1} V_i = \left({}^{i-1} \delta_i \wedge {}^{i-1} L_{i1} \quad {}^{i-1} \delta_i \wedge {}^{i-1} L_{i2} \quad {}^{i-1} \delta_i \wedge {}^{i-1} L_{i3} \right)^T + {}^{i-1} h_i {}^{i-1} \lambda_i^T \quad (3.62)$$

⁶ ${}^P L_{ik}$ denota a k -ésima coluna da matriz ${}^P L_i$.

$${}^{i-1}\mathbf{W}_i = \overline{{}^{i-1}\mathbf{V}_i} \quad (3.63)$$

$${}^{i-1}\phi_i = {}^{i-1}\mathbf{U}_i {}^{i-1}\delta_i + {}^{i-1}\mathbf{h}_i \wedge {}^{i-1}\lambda_i \quad (3.64)$$

$${}^{i-1}\psi_i = {}^{i-1}\delta_i \wedge {}^{i-1}\mathbf{h}_i + M_i {}^{i-1}\lambda_i \quad (3.65)$$

Também definiremos:

$${}^i\mathbf{g} \stackrel{\text{def}}{=} {}^{i-1}\mathbf{R}_i^T {}^{i-1}\mathbf{g} \quad (3.66)$$

com ${}^0\mathbf{g} = \mathbf{g}$.

Utilizando então A.3, A.4 e A.5 podemos mudar as equações 3.17, 3.18 e 3.19 para:

$$D_{ij} = {}^{j-1}\delta_i^T {}^{j-1}\phi_j + {}^{j-1}\lambda_i^T {}^{j-1}\psi_j \quad (i \leq j) \quad (3.67)$$

$$D_{ijk} = {}^{k-1}\delta_j^T \left({}^{k-1}\mathbf{W}_k {}^{k-1}\delta_i + {}^{k-1}\psi_k \wedge {}^{k-1}\lambda_i \right) \quad (i < k, j \leq k) \quad (3.68)$$

$$D_i = -{}^{i-1}\mathbf{g}^T {}^{i-1}\psi_i \quad (3.69)$$

Ainda, o algoritmo recorrente deduzido fica:

$${}^i\mathbf{L}_i = {}^i\mathbf{L}_{i+1} + \bar{I}_i + m_i \left(\bar{\mathbf{r}}_i + {}^i\mathbf{p}_i \right) \left(\bar{\mathbf{r}}_i + {}^i\mathbf{p}_i \right)^T \quad (3.70)$$

$${}^{i-1}\mathbf{L}_i = {}^{i-1}\mathbf{R}_i {}^i\mathbf{L}_i {}^{i-1}\mathbf{R}_i^T \quad (3.71)$$

$${}^i\mathbf{h}_i = {}^i\mathbf{h}_{i+1} + m_i \left(\bar{\mathbf{r}}_i + {}^i\mathbf{p}_i \right) \quad (3.72)$$

$${}^{i-1}\mathbf{h}_i = {}^{i-1}\mathbf{R}_i {}^i\mathbf{h}_i \quad (3.73)$$

$$M_i = M_{i+1} + m_i \quad (3.74)$$

com ${}^n\mathbf{L}_{n+1} = 0$, ${}^n\mathbf{h}_{n+1} = 0$, $M_{n+1} = 0$.

Também:

$${}^{j-1}\mathbf{X}_{ij} = \left({}^{j-1}\delta_i \wedge {}^{j-1}\mathbf{V}_{j1} \quad {}^{j-1}\delta_i \wedge {}^{j-1}\mathbf{V}_{j2} \quad {}^{j-1}\delta_i \wedge {}^{j-1}\mathbf{V}_{j3} \right) + {}^{j-1}\lambda_i \wedge {}^{j-1}\psi_j^T \quad (i \leq j) \quad (3.75)$$

$${}^{j-1}\mathbf{Y}_{ij} = \overline{{}^{j-1}\mathbf{X}_{ij}} \quad (i \leq j) \quad (3.76)$$

Portanto:

$$D_{ijks} = {}^{s-1}\delta_k^T {}^{s-1}\mathbf{Y}_{js} {}^{s-1}\delta_i \quad (i \leq s, j \leq s, k \leq s) \quad (3.77)$$

E, finalmente:

$$C_{ij}^g = \begin{cases} {}^{j-1}\delta_i^T \left({}^{j-1}\mathbf{g} \wedge {}^{j-1}\psi_j \right) & \text{se } (i \leq j) \\ C_{ji}^g & \text{se } (i > j) \end{cases} \quad (3.78)$$

3.4.1 Simplificações

Analogamente ao executado anteriormente, podemos fazer as seguintes simplificações:

1. Se a junta i for de translação (o que implica $\lambda_i = 0$):

$${}^{i-1}\phi_i = {}^{i-1}h_i \wedge {}^{i-1}\lambda_i \quad (3.79)$$

$${}^{i-1}\psi_i = M_i {}^{i-1}\lambda_i \quad (3.80)$$

$${}^{i-1}V_i = {}^{i-1}h_i {}^{i-1}\lambda_i^T \quad (3.81)$$

$${}^{j-1}X_{ij} = {}^{j-1}\lambda_i {}^{j-1}\psi_j^T \quad (i \leq j) \quad (3.82)$$

$$D_{ij} = {}^{j-1}\lambda_i^T {}^{j-1}\psi_j \quad (i \leq j) \quad (3.83)$$

$$D_{ijk} = {}^{k-1}\delta_j^T ({}^{k-1}\psi_k \wedge {}^{k-1}\lambda_i) \quad (i < k, j \leq k) \quad (3.84)$$

$$D_{ijks} = 0 \quad (i < s, j \leq s, k \leq s) \quad (3.85)$$

$$C_{ij}^g = 0 \quad (i \leq j) \quad (3.86)$$

2. Se a junta j for de translação:

$$D_{ijk} = 0 \quad (i < k, j \leq k) \quad (3.87)$$

3. Se a junta k for de translação:

$$D_{ijks} = 0 \quad (i \leq s, j \leq s, k \leq s) \quad (3.88)$$

3.5 O ALGORITMO PARA O MDI

Como visto anteriormente, obtivemos algoritmos eficientes para o cálculo de todos os termos da equação dinâmica. Usaremos parte destes algoritmos no cálculo do MDI.

Lembrando que:

$$\mathbf{f} = \mathbf{Y}(\mathbf{q}) \ddot{\mathbf{q}} + \boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3.89)$$

onde $\boldsymbol{\sigma}$ agrupa $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{e}(\mathbf{q})$, podemos escrever:

$$\mathbf{Y}(\mathbf{q}) \ddot{\mathbf{q}} = \mathbf{f} - \boldsymbol{\sigma}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3.90)$$

Por outro lado, como é conhecido da análise numérica, o fato de \mathbf{Y} ser simétrica definida positiva, implica que podemos fatorá-la em:

$$\mathbf{Y} = \mathbf{L} \mathbf{L}^T$$

onde \mathbf{L} é uma matriz triangular inferior $n \times n$. Portanto o sistema 3.90 fica:

$$\mathbf{L} \mathbf{L}^T \ddot{\mathbf{q}} = \mathbf{f} - \boldsymbol{\sigma} \quad (3.91)$$

que é equivalente aos 2 sistemas seguintes:

$$\mathbf{L} \mathbf{x} = \mathbf{f} - \boldsymbol{\sigma} \quad (3.92)$$

$$\mathbf{L}^T \ddot{\mathbf{q}} = \mathbf{x} \quad (3.93)$$

onde \mathbf{x} é um vetor auxiliar $n \times 1$. Este sistema pode facilmente ser resolvido por substituição, ou seja, resolvemos \mathbf{x} de 3.92 e posteriormente $\ddot{\mathbf{q}}$ de 3.93. Ou então, simultaneamente resolvemos os 2 sistemas, o que é possível, se incorporarmos pequenas modificações no algoritmo de fatoração.

Podemos então integrar $\ddot{\mathbf{q}}$, usando um método apropriado.

OBSERVAÇÕES:

1. Pode-se fazer também a decomposição $\mathbf{L}^T \mathbf{L}$ ao invés de $\mathbf{L} \mathbf{L}^T$.
2. Usaremos, para a decomposição $\mathbf{L} \mathbf{L}^T$, o conhecido algoritmo de Cholesky, com pequenas modificações.

3.5.1 O Algoritmo de Cholesky Modificado

Sendo D_{i+1} o termo $f_i - \sigma_i$, o conhecido algoritmo de Cholesky para resolução de sistemas lineares [PRE 89] aplicado ao MDI fica:

$$v_{ii} \stackrel{\text{def}}{=} \left(D_{ii} - \sum_{k=1}^{i-1} v_{ki}^2 \right)^{\frac{1}{2}} \quad i = 1, \dots, n \quad (3.94)$$

$$v_{ij} \stackrel{\text{def}}{=} \frac{1}{v_{ii}} \left(D_{ij} - \sum_{k=1}^{i-1} v_{ki} v_{kj} \right) \quad i = 1, \dots, n; j = i + 1, \dots, n + 1 \quad (3.95)$$

$$\tilde{q}_i = \frac{1}{v_{ii}} \left(v_{in+1} - \sum_{k=i+1}^n v_{ik} \tilde{q}_k \right) \quad i = n, \dots, 1 \quad (3.96)$$

Fazendo as seguintes substituições:

$$\begin{aligned} v_{ii} &= \sqrt{s_{ii}} \\ v_{ij} &= s_{ij} \sqrt{s_{ii}} \end{aligned}$$

conseguimos eliminar as raízes quadradas (que são relativamente custosas em termos de tempo de cálculo).

Então, temos para o algoritmo “inversor” do MDI as seguintes equações:⁷

$$s_{ii} \stackrel{\text{def}}{=} D_{ii} - \sum_{k=1}^{i-1} s_{ki}^2 s_{kk} \quad i = 1, \dots, n \quad (3.97)$$

$$s_{ij} \stackrel{\text{def}}{=} \frac{1}{s_{ii}} \left(D_{ij} - \sum_{k=1}^{i-1} s_{ki} s_{kj} s_{kk} \right) \quad i = 1, \dots, n; j = i + 1, \dots, n + 1 \quad (3.98)$$

$$\tilde{q}_i = s_{in+1} - \sum_{k=i+1}^n s_{ik} \tilde{q}_k \quad i = n, \dots, 1 \quad (3.99)$$

3.5.2 Os Algoritmos de Runge - Kutta

Considerações Iniciais

Muitos problemas físicos podem ser expressos em termos de equações diferenciais ordinárias. Estas podem ser escritas como:

$$\frac{dy}{dx} = f'(x, \mathbf{y})$$

onde x é a variável independente e \mathbf{y} é a variável dependente.

Em geral, poucos são os tipos de equações diferenciais ordinárias que podem ser resolvidas por técnicas puramente analíticas. Na maior parte dos casos, precisamos recorrer a algoritmos numéricos.

Existem vários métodos de integração numérica voltados para equações diferenciais ordinárias. Entre os mais utilizados, podemos citar:

⁷Perceba que s_{ii} e s_{ij} são interdependentes, ou seja, não é possível calcular todos os s_{ii} e depois todos os s_{ij} .

- Método de Euler
- Método de Runge-Kutta
- Método de Runge-Kutta-Fehlberg
- Extrapolação de Richardson
- Método de Burlish-Stoer
- Método de Adams-Bashforth-Moulton

O Método Adotado

Neste trabalho, utilizaremos algoritmos de **Runge-Kutta** pelas seguintes razões:

1. Não necessitam do cálculo de derivadas.
2. Apresentam uma boa estabilidade.
3. Não necessitam de algoritmos auxiliares para sua iniciação.
4. Facilitam a mudança de passo entre iterações.

O algoritmo de integração **Runge-Kutta** utilizado neste trabalho será o de 4a. ordem, pois o compromisso entre o número de avaliações de f' e a precisão obtida leva-nos à escolha desta como o "breakpoint" natural entre as diversas ordens possíveis [PRE 89].

Este algoritmo pode ser escrito como:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\mathbf{k}_1}{6} + \frac{\mathbf{k}_2}{3} + \frac{\mathbf{k}_3}{3} + \frac{\mathbf{k}_4}{6} \quad (3.100)$$

onde:

$$\begin{aligned} \mathbf{k}_1 &= h \mathbf{f}'(x_n, \mathbf{y}_n) \\ \mathbf{k}_2 &= h \mathbf{f}'\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{\mathbf{k}_1}{2}\right) \\ \mathbf{k}_3 &= h \mathbf{f}'\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{\mathbf{k}_2}{2}\right) \\ \mathbf{k}_4 &= h \mathbf{f}'(x_n + h, \mathbf{y}_n + \mathbf{k}_3) \end{aligned}$$

O Passo Adaptativo

A necessidade do passo adaptativo provém do fato da função que se está integrando variar mais ou menos abruptamente dependendo do valor da variável independente. Passos fixos muito pequenos iriam ocasionar uma demora muito grande do algoritmo, enquanto que passos muito grandes iriam fornecer uma resposta muito pouco precisa. Além disto, o algoritmo de passo variável aqui utilizado automaticamente busca o melhor passo dentro da precisão imposta pelo usuário, coisa que o passo fixo não faz, ou seja, é muito difícil especificar um passo fixo conveniente sem uma boa idéia do comportamento da função a ser integrada.

O algoritmo que veremos aqui usa a técnica do “step doubling”, que consiste em integrar uma vez com o passo todo, em seguida duas vezes com meio passo e comparar a diferença y_d com uma tolerância pré-especificada, utilizando um conjunto de heurísticas matematicamente razoáveis para atualização do passo.

O número de avaliações da função neste algoritmo é 11, contra 8 do algoritmo de passo fixo, pois sempre utilizamos para o resultado a integração com 2 meio passos.

O algoritmo vem descrito a seguir:

A partir de uma tolerância relativa ϵ previamente especificada, executamos a cada etapa de integração o seguinte procedimento:

Calculamos

$$y_e = \epsilon \left(|y| + \left| h \frac{dy}{dx} \right| + a \right)$$

onde a é uma constante positiva e muito pequena em relação à precisão da máquina.

Em seguida, isolamos o máximo valor absoluto ρ entre os $y_d[i]/y_e[i]$, com i variando de 1 até a ordem (n) do sistema. Se ρ for menor ou igual a 1.0, o passo utilizado é informado ao subprograma integrador e o próximo passo a ser tentado será dado por $0.9 h \rho^{-0.2}$. Do contrário, o passo utilizado não é informado ao programa principal e tentamos um novo passo dado por $0.9 h \rho^{-0.25}$.

Apenas para finalizar, podemos observar que o valor 0.2 provém de considerações sobre a ordem do algoritmo de **Runge-Kutta** (4a.), enquanto que o valor 0.25 provém do fato de querermos, em geral, controlar também o erro global da integração e portanto, quando reduzimos o passo, reduzimos também a estimativa de erro local máxima admitida. A favor da segurança, adotamos 0.25 quando diminuimos o passo e 0.2 quando aumentamos o passo, ou seja, forçamos mais a diminuição que o aumento do passo.

Para maiores detalhes deste algoritmo, veja [PRE 89].

Capítulo 4

SIMULAÇÃO

4.1 DESCRIÇÃO DO CONTROLADOR EXEMPLO

Nas simulações deste trabalho, usaremos um controlador baseado em [GUO 89], com a diferença que vamos calcular as sensibilidades necessárias (Lá usou-se um esquema adaptativo que não calculava estas sensibilidades. Ao invés, tentava-se estimar as matrizes de ganho diretamente). O objetivo, aqui, é apenas ilustrar a utilização do simulador desenvolvido para a avaliação de opções de esquemas de controle.

No capítulo anterior, deduzimos que a linearização de f em torno de $(q_0, \dot{q}_0, \ddot{q}_0)$ pode ser escrita como:

$$\delta f = A \delta \ddot{q} + 2B \delta \dot{q} + C \delta q \quad (4.1)$$

onde as matrizes de sensibilidade foram lá definidas.

A idéia deste controlador é trazer o MDL para uma forma canônica conhecida (invariante), através de uma realimentação do tipo PD neste modelo (Este esquema pode ser chamado **controle por cancelamento linearizado**).

Ou seja, trazer o MDL para um sistema da forma:

$$\delta \ddot{q} + 2\Psi^T \Omega \Psi \delta \dot{q} + \Omega^2 \delta q = 0 \quad (4.2)$$

onde Ψ é uma matriz $n \times n$ adimensional e constante, denominada matriz de amortecimento do sistema e Ω é uma matriz $n \times n$ constante, denominada matriz de frequências naturais do sistema. Em [GUO 89] ressaltava-se que nesta abordagem, ao invés de tentarmos estabilizar o sistema (MDL) por **atribuição de pólos**, a estabilização é tentada via **atribuição de modelo**, e para isto, o fato de um sistema da forma 4.2 ser estável se os coeficientes matriciais forem definidos positivos é usado.

Podemos então escolher para Ψ uma matriz não-singular e para Ω uma matriz definida positiva, o que satisfará a condição acima. Por simplicidade escolheremos estas matrizes como diagonais.

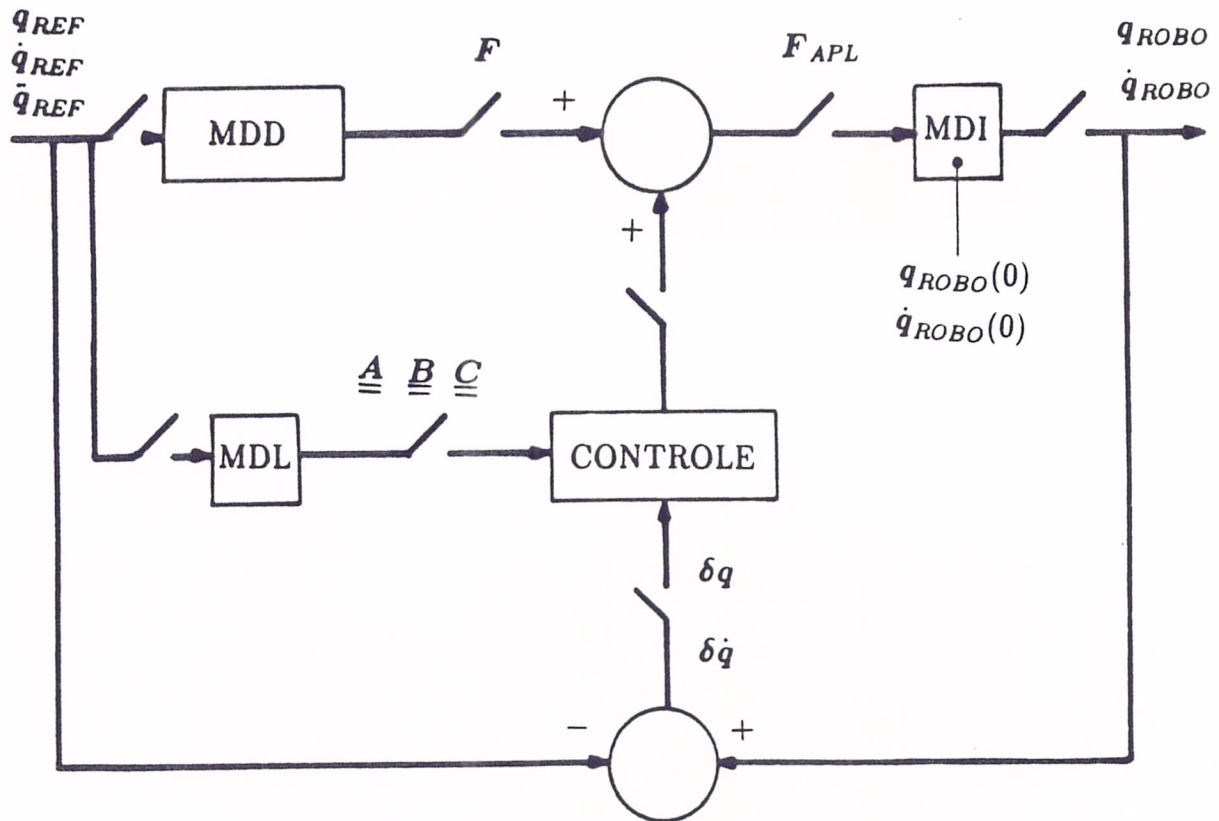


Figura 4.1: Esquema de Controle Utilizado

Suponha K_P e K_D duas matrizes de ganho variantes no tempo (que iremos deduzir). Impomos:

$$\delta f = - (K_P \delta q + K_D \delta \dot{q}) \quad (4.3)$$

Substituindo 4.3 em 4.1 nos fornece:

$$A \delta \ddot{q} + [2B + K_D] \delta \dot{q} + [C + K_P] \delta q = 0 \quad (4.4)$$

Portanto, para conseguirmos que o sistema fique na forma da expressão 4.2 devemos ter:

$$K_P = A\Omega^2 - C \quad (4.5)$$

$$K_D = 2A\Psi^T \Omega \Psi - 2B \quad (4.6)$$

Para uma visão geral do esquema de controle veja a Fig. 4.1.

4.2 ALGORITMOS DINÂMICOS

Listaremos agora os algoritmos do simulador.

4.2.1 Entrada de Parâmetros (EP)

1. Abra arquivo de dados.
2. Para $i := 1, \dots, n$ obtenha os parâmetros geométricos e dinâmicos de cada ligamento.
3. Feche arquivo de dados.
4. FIM DO ALGORITMO.

4.2.2 Entrada de Pontos da Trajetória (EQ)

1. Obtenha próximo ponto $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ da trajetória nominal.
2. FIM DO ALGORITMO.

4.2.3 Primeiro algoritmo para cálculo do MDD (MDD1)

1. Para $i := 1, \dots, n$ calcule \mathbf{R}_i e \mathbf{p}_i por: 2.3 e 2.4.
2. Para $i := 1, \dots, n$ calcule δ_i e λ_i por: 2.5 e 2.6.
3. Para $i := n, \dots, 1$ calcule \mathbf{L}_i , \mathbf{h}_i e \mathbf{M}_i por: 3.20, 3.21 e 3.22.
4. Para $i := 1, \dots, n$ calcule \mathbf{U}_i , \mathbf{V}_i e \mathbf{W}_i por: 3.9, 3.10 e 3.11.
5. Para $i := 1, \dots, n$ calcule ϕ_i e ψ_i por: 3.15 e 3.16.
6. Para $i, k := 1, \dots, n$ faça:
 - Se $i < k$ então calcule $(\mathbf{W}_k \delta_i + \psi_k \wedge \lambda_i)$.
7. Para $i := 1, \dots, n$ calcule D_i por: 3.19.
8. Para $i, j := 1, \dots, n$ faça:
 - Se $i \leq j$ então calcule D_{ij} por: 3.17.
9. Para $i, j, k := 1, \dots, n$ faça:
 - Se $i < k, j \leq k$ então calcule D_{ijk} por: 3.18.

10. Para $i := 1, \dots, n$ calcule f_{ai} , f_{ci} , f_{gi} e f_i por: 2.43, 2.44, 2.45 e 2.42, respectivamente.

11. FIM DO ALGORITMO.

4.2.4 Primeiro algoritmo para cálculo do MDL (MDL1)

OBS: Supõe MDD1 já executado.

1. Para $i, j := 1, \dots, n$ faça:
 - Se $i \leq j$ então calcule X_{ij} e Y_{ij} por: 3.37 e 3.38.
2. Para $i, j, k, s := 1, \dots, n$ faça:
 - Se $i \leq s, j \leq s, k \leq s$ então calcule $D_{ijk s}$ por: 3.39.
3. Para $i, j := 1, \dots, n$ calcule A_{ij} da seguinte maneira:
 - Se $i \leq j$ use: 3.26 - caso 1.
 - Se $i > j$ use: 3.26 - caso 2.
4. Para $i, j := 1, \dots, n$ calcule B_{ij} da seguinte maneira:
 - Se $i < j$ use: 3.27 - caso 1.
 - Se $i = j$ use: 3.27 - caso 2.
 - Se $i = j + 1$ use: 3.27 - caso 3.
 - Se $i > j + 1$ use: 3.27 - caso 4.
5. Para $i, j := 1, \dots, n$ calcule C_{ij}^a da seguinte maneira:
 - Se $i < j$ use: 3.32 - caso 1.
 - Se $i = j$ use: 3.32 - caso 2.
 - Se $i > j$ use: 3.32 - caso 3.
6. Para $i, j := 1, \dots, n$ calcule C_{ij}^c da seguinte maneira:
 - Se $i < j$ use: 3.43 - caso 1.
 - Se $i = j$ use: 3.43 - caso 2.
 - Se $i > j$ use: 3.43 - caso 3.
7. Para $i, j := 1, \dots, n$ calcule C_{ij}^g da seguinte maneira:
 - Se $i \leq j$ use: 3.45 - caso 1.
 - Se $i > j$ use: 3.45 - caso 2.
8. Para $i, j := 1, \dots, n$ calcule C_{ij} por: 3.28.
9. FIM DO ALGORITMO.

4.2.5 Segundo algoritmo para cálculo do MDD (MDD2)

1. Para $i := 1, \dots, n$ calcule ${}^i p_i$ por: 3.56.
2. Para $i, j := 1, \dots, n$ faça:
 - Se $i \leq j$ então calcule ${}^{j-1} \delta_i$ e ${}^{j-1} \lambda_i$ por: 3.59, 3.60 e 3.57, 3.58.
3. Para $i := n, \dots, 1$ calcule ${}^i L_i, {}^{i-1} L_i, {}^i h_i, {}^{i-1} h_i$, e M_i por: 3.70, 3.71, 3.72, 3.73 e 3.74.
4. Para $i := 1, \dots, n$ calcule ${}^{i-1} U_i, {}^{i-1} V_i$ e ${}^{i-1} W_i$ por: 3.61, 3.62 e 3.63.
5. Para $i := 1, \dots, n$ calcule ${}^{i-1} \phi_i$ e ${}^{i-1} \psi_i$ por: 3.64 e 3.65.
6. Para $i, k := 1, \dots, n$ faça:
 - Se $i < k$ então calcule $({}^{k-1} W_k {}^{k-1} \delta_i + {}^{k-1} \psi_k \wedge {}^{k-1} \lambda_i)$.
7. Para $i := 1, \dots, n$ calcule ${}^{i-1} g$ por: 3.66.
8. Para $i := 1, \dots, n$ calcule D_i por: 3.69.
9. Para $i, j := 1, \dots, n$ faça:
 - Se $i \leq j$ então calcule D_{ij} por: 3.67.
10. Para $i, j, k := 1, \dots, n$ faça:
 - Se $i < k, j \leq k$ então calcule D_{ijk} por: 3.68.
11. Para $i := 1, \dots, n$ calcule f_{ai}, f_{ci}, f_{gi} e f_i por: 2.43, 2.44, 2.45 e 2.42, respectivamente.
12. FIM DO ALGORITMO.

4.2.6 Segundo algoritmo para cálculo do MDL (MDL2)

OBS: Supõe MDD2 já executado.

1. Para $i, j := 1, \dots, n$ faça:
 - Se $i \leq j$ então calcule ${}^{j-1} X_{ij}$ e ${}^{j-1} Y_{ij}$ por: 3.75 e 3.76.
2. Para $i, j, k, s := 1, \dots, n$ faça:
 - Se $i \leq s, j \leq s, k \leq s$ então calcule D_{ijks} por: 3.77.
3. Para $i, j := 1, \dots, n$ calcule A_{ij} da seguinte maneira:
 - Se $i \leq j$ use: 3.26 - caso 1.

- Se $i > j$ use: 3.26 - caso 2.
4. Para $i, j := 1, \dots, n$ calcule B_{ij} da seguinte maneira:
 - Se $i < j$ use: 3.27 - caso 1.
 - Se $i = j$ use: 3.27 - caso 2.
 - Se $i = j + 1$ use: 3.27 - caso 3.
 - Se $i > j + 1$ use: 3.27 - caso 4.
 5. Para $i, j := 1, \dots, n$ calcule C_{ij}^a da seguinte maneira:
 - Se $i < j$ use: 3.32 - caso 1.
 - Se $i = j$ use: 3.32 - caso 2.
 - Se $i > j$ use: 3.32 - caso 3.
 6. Para $i, j := 1, \dots, n$ calcule C_{ij}^c da seguinte maneira:
 - Se $i < j$ use: 3.43 - caso 1.
 - Se $i = j$ use: 3.43 - caso 2.
 - Se $i > j$ use: 3.43 - caso 3.
 7. Para $i, j := 1, \dots, n$ calcule C_{ij}^g da seguinte maneira:
 - Se $i \leq j$ use: 3.78 - caso 1.
 - Se $i > j$ use: 3.78 - caso 2.
 8. Para $i, j := 1, \dots, n$ calcule C_{ij} por: 3.28.
 9. FIM DO ALGORITMO.

4.2.7 Cálculo do MDI (MDI)

OBS: Há dois algoritmos possíveis conforme passo 2. Note que utilizamos o algoritmo de Cholesky para resolver o sistema de equações.

1. Leia o vetor de estados inicial formado por: \dot{q} e q iniciais.
2. Obtenha nova força aplicada f (entrada)
3. Execute um algoritmo de MDD sem o último passo (MDD1 ou MDD2).
4. Para $i := 1, \dots, n$ calcule f_{ci} e f_{gi} por: 2.44, 2.45.
5. Para $i := 1, \dots, n$ calcule $D_{in+1} = f_i - f_{ci} - f_{gi}$.
6. Para $i := 1, \dots, n$ calcule s_{ii} por: 3.97.

- Para $j := i + 1, \dots, n + 1$ calcule s_{ij} por: 3.98.
7. Para $i := n, \dots, 1$ calcule \tilde{q}_i por: 3.99.
 8. Integre a derivada do vetor de estados formada por \tilde{q} e \dot{q} por Runge Kutta de ordem 4 (fórmula 3.100), obtendo os novos \dot{q} e q .¹
 9. FIM DO ALGORITMO.

4.2.8 Algoritmo de Controle (AC)

O algoritmo de controle que estamos utilizando fica: (Supõe lidas as matrizes Ψ e Ω .)

1. A partir de A , B e C calculados do MDL, forme as matrizes K_P e K_D por: 4.5 e 4.6.
2. Calcule δf por: 4.3.
3. Determine a nova força aplicada ao robô $f_{APL} = f_{MDD} + \delta f$.

4.2.9 Algoritmo do Simulador (AS)

O algoritmo descrito a seguir é uma combinação de todos os algoritmos descritos até aqui. A idéia é usar a força de referência (calculada pelo “feedforward”) em “escada”, ou seja, simular exatamente da maneira como implementaríamos este esquema com um controlador digital (em certos pontos da trajetória determinados pelo amostrador). Note que o robô estaria no lugar do bloco do MDI. Estamos fazendo estas observações em virtude do integrador do MDI que é usado como um subprograma do módulo de simulação, poder necessitar de valores da força de referência em vários instantes entre 2 amostragens. Aliás, a força de referência será calculada como a média entre a atual e a do instante posterior, ou seja, tentaremos gerar uma “escada” média (Percebemos em várias simulações que o desempenho do controlador melhora bastante com isto). O simulador usará o seguinte esquema de processamento.

1. Leia as matrizes Ψ e Ω usadas no controle.
2. Leia o vetor de estados inicial formado por: \dot{q} e q iniciais do robô.
3. Leia o novo ponto de referência e calcule a força de referência atual f (“feedforward”).
4. Para o número de pontos de simulação, faça:
 - Salve a f atual em um vetor f_{ANT} .

¹Se a escolha do usuário for o passo adaptativo, utilizamos adicionalmente as heurísticas para atualização de passo que foram descritas anteriormente.

- Calcule a nova força de referência \mathbf{f} .
- Calcule \mathbf{f}_{APL} por: $(\mathbf{f} + \mathbf{f}_{ANT})/2$.
- Calcule os erros de posição e de velocidade.
- Execute o módulo de controle para determinar $\delta\mathbf{f}$.
- Atualize \mathbf{f}_{APL} por: $\mathbf{f}_{APL} + \delta\mathbf{f}$.
- Com o estado do robô e com a força aplicada calcule $\ddot{\mathbf{q}}$ utilizando o algoritmo de Cholesky (veja o algoritmo do MDI acima).
- Integre a derivada do vetor de estados formada por $\ddot{\mathbf{q}}$ e $\dot{\mathbf{q}}$ por Runge Kutta de ordem 4 (fórmula 3.100), obtendo os novos $\dot{\mathbf{q}}$ e \mathbf{q} . Lembrar que se a escolha do usuário for o passo adaptativo, utilizamos adicionalmente as heurísticas para atualização de passo que foram descritas anteriormente.

5. FIM DO ALGORITMO.

Capítulo 5

RESULTADOS

5.1 DESCRIÇÃO SUCINTA DO PROGRAMA

O programa de computador desenvolvido permite várias opções, que são descritas em “menus” explicativos (Fig. 5.1). Iremos aqui listar as possibilidades disponíveis na versão atual.

As opções do “menu” principal são:

ARQUIVOS Primeiramente, o usuário especifica o nome dos arquivos do modelo e do robô, a serem utilizados nas opções **FEEDFORWARD** e **MODINV** respectivamente (Um exemplo de arquivo será fornecido adiante). O programa informa então qualquer erro sintático cometido pelo usuário, explicitando a linha e coluna onde o erro foi detectado e a palavra que se estava esperando naquele momento. O padrão adotado para estes arquivos exige uma certa sequência de palavras-chave, porém permitindo a introdução de comentários utilizando-se um asterisco (“*”) logo no início de uma linha. Detectamos também se o número de juntas do robô é igual ao do modelo (É o que exigimos de compatibilidade em relação aos dois arquivos).

FEEDFORWARD Neste módulo, calculam-se para o modelo as componentes de força inercial, centrípeta+Coriolis e estática gravitacional, bem como a força total resultante, a partir de uma certa trajetória de referência previamente especificada.

MODINV Aqui, a partir de uma referência de forças generalizadas em função do tempo aplicadas a cada junta do robô e de sua posição e velocidade iniciais, calcula-se qual a trajetória seguida pelas juntas do manipulador, ou melhor, pode-se visualizar força de referência, posição, velocidade e aceleração.

TRAJREF Neste módulo, calculam-se as curvas de posição, velocidade e aceleração de referência.

SIMULADOR No módulo simulador, estaremos verificando o desempenho de um particular esquema de controle, com possibilidade de variação do valor dos parâmetros

MENU PRINCIPAL	
ARQUIVOS	-----> DADOS DO MODELO E DO ROBO
FEEDFORWARD	
MODINV	
TRAJREF	
SIMULADOR	
PLOTAR	
GUARDAR	
ENCERRAR	

MODULO GRAFICO	
1 - POSICAO	
2 - VELOCIDADE	
3 - ACELERACAO	
4 - FORCA APLICADA	-----> PLOTA FORCA APLICADA AO ROBO
5 - SAIR	

Figura 5.1: Exemplo de Telas do Programa

do controlador. Podemos facilmente, adaptar o simulador para vários esquemas de controle que desejemos estudar. Aqui, as possibilidades de visualização são: posição e velocidade do robô, erros de posição e velocidade e força aplicada ao robô.

PLOTAR Faz-se aqui, a visualização das variáveis temporais calculadas em ítems anteriores. Conforme o “menu” que anteriormente ativamos, é chamado um outro “menu” específico que controla a chamada de um módulo gráfico de alta resolução. Podemos executar ainda “zooms” no gráfico obtido, bem como imprimí-lo com uma boa qualidade.

GUARDAR Neste módulo, salvamos em disco (arquivo texto) as variáveis temporais calculadas em ítems anteriores. Conforme o “menu” que anteriormente ativamos, salva-se um determinado conjunto de arquivos referentes àquele “menu”.

ENCERRAR Abandona o programa.

Além disto, podemos variar diversos parâmetros tais como passo de integração, posição inicial, se usa passo fixo ou adaptativo, tolerância relativa no integrador com passo adaptativo, etc...

5.2 PÊNDULO DUPLO

5.2.1 Equações Analíticas

A seguir, fornecemos as equações dinâmicas analíticas para o pêndulo duplo (vide Fig. 5.2).

Deve-se notar que em [PAU 81], houve um erro na dedução destas equações e as fórmulas lá contidas devem ser corrigidas para as apresentadas abaixo. Note a complexidade das equações, mesmo para um caso tão simples como este (massas concentradas, manipulador planar com grau de mobilidade 2).

Para encontrarmos o Lagrangiano do sistema, devemos computar as energias cinética e potencial.

Notemos primeiramente que, em relação ao sistema de base, temos:

$$\begin{aligned}x_2 &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\y_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)\end{aligned}$$

e que portanto:

$$\begin{aligned}\dot{x}_2 &= -l_1 \sin(\theta_1) \dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{y}_2 &= l_1 \cos(\theta_1) \dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)\end{aligned}$$

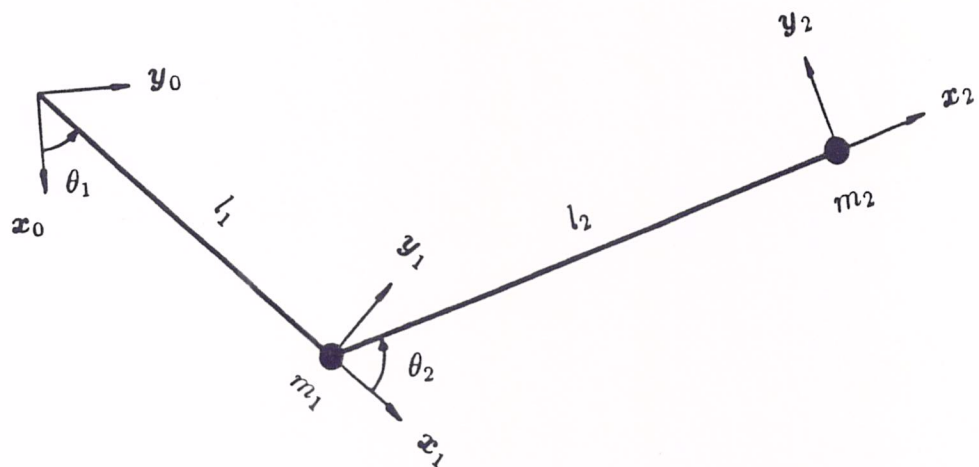


Figura 5.2: Esquema do Pêndulo Duplo

Então, o quadrado da velocidade de m_2 pode ser escrito como:

$$\begin{aligned} (\dot{x}_2^2 + \dot{y}_2^2) &= l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ &+ 2l_1 l_2 \cos(\theta_1) \cos(\theta_1 + \theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \\ &+ 2l_1 l_2 \sin(\theta_1) \sin(\theta_1 + \theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \\ &= l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) + 2l_1 l_2 \cos(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \end{aligned}$$

Portanto podemos formar as energias cinética e potencial como abaixo.

$$K_1 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 \quad (5.1)$$

$$P_1 = -m_1 g l_1 \cos(\theta_1) \quad (5.2)$$

$$K_2 = \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) + m_2 l_1 l_2 \cos(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \quad (5.3)$$

$$P_2 = -m_2 g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_1 + \theta_2) \quad (5.4)$$

O Lagrangiano do sistema então fica:

$$\begin{aligned} L &= \frac{1}{2} (m_1 + m_2) l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ &+ m_2 l_1 l_2 \cos(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) + (m_1 + m_2) g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_1 + \theta_2) \end{aligned}$$

Para obter agora as equações dinâmicas, aplicamos a equação de Lagrange, resultando:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} &= [(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(\theta_2)] \ddot{\theta}_1 \\ &+ [m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2)] \ddot{\theta}_2 \\ &- 2m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_2^2 \end{aligned} \quad (5.5)$$

$$\frac{\partial L}{\partial \theta_1} = -(m_1 + m_2) g l_1 \sin(\theta_1) - m_2 g l_2 \sin(\theta_1 + \theta_2) \quad (5.6)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \cos(\theta_2) \ddot{\theta}_1 - m_2 l_1 l_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 \quad (5.7)$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_2 \sin(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) - m_2 g l_2 \sin(\theta_1 + \theta_2) \quad (5.8)$$

Finalmente, podemos escrever os torques como:

$$\begin{aligned} \tau_1 &= [(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(\theta_2)] \ddot{\theta}_1 \\ &+ [m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2)] \ddot{\theta}_2 \end{aligned}$$

$$\begin{aligned}
& - 2m_2l_1l_2\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 - m_2l_1l_2\sin(\theta_2)\dot{\theta}_2^2 \\
& + (m_1 + m_2)gl_1\sin(\theta_1) + m_2gl_2\sin(\theta_1 + \theta_2)
\end{aligned} \tag{5.9}$$

$$\begin{aligned}
\tau_2 = & \left[m_2l_2^2 + m_2l_1l_2\cos(\theta_2) \right] \ddot{\theta}_1 + m_2l_2^2\ddot{\theta}_2 \\
& + m_2l_1l_2\sin(\theta_2)\dot{\theta}_1^2 + m_2gl_2\sin(\theta_1 + \theta_2)
\end{aligned} \tag{5.10}$$

Ou explicitando os componentes:

$$\tau_1 = D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_1\dot{\theta}_2 + D_1 \tag{5.11}$$

$$\tau_2 = D_{21}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_1\dot{\theta}_2 + D_2 \tag{5.12}$$

onde:

$$\begin{aligned}
D_{11} &= (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2\cos(\theta_2) \\
D_{22} &= m_2l_2^2 \\
D_{12} &= m_2l_2^2 + m_2l_1l_2\cos(\theta_2) \\
D_1 &= (m_1 + m_2)gl_1\sin(\theta_1) + m_2gl_2\sin(\theta_1 + \theta_2) \\
D_2 &= m_2gl_2\sin(\theta_1 + \theta_2) \\
D_{112} &= -m_2l_1l_2\sin(\theta_2) \\
D_{122} &= -m_2l_1l_2\sin(\theta_2)
\end{aligned}$$

e também:¹

$$\begin{aligned}
D_{21} &= D_{12} \\
D_{111} &= 0 \\
D_{222} &= 0 \\
D_{212} &= 0 \\
D_{221} &= 0 \\
D_{121} &= D_{112} \\
D_{211} &= -D_{112}
\end{aligned}$$

5.2.2 Arquivo de Dados do Pêndulo Duplo

Listamos a seguir o arquivo de entrada no formato que o programa exige. Note as palavras-chave. Os comentários explicam-nas.

¹Note que, lembrando as fórmulas 2.23, 2.24, 2.30 e 2.31, poderíamos adiantar estes resultados.

* OBSERVACOES IMPORTANTES
* Angulos serao dados de -180 a 180 graus senao serao recusados
* Os LL deverao ser valores ≥ 0 senao serao recusados
* ESTE ROBO ESTA EM PAUL P.159
* ELE SUPOS MASSAS CONCENTRADAS. DESTE MODO AS MATRIZES DE INERCIA
* HOMOGENEAS RESULTARAM MUITO ESPARSAS

* CONSTANTE GRAVITACIONAL DO PROBLEMA G

G 9.8 0 0

* NOME DO ROBO

ROBO PLANAR1

* NUMERO DE LIGACOES

NLIGS 2

* PARAMETROS GEOMETRICOS
* EXEMPLO: ALFA <ALFA1> <ALFA2>
* A PALAVRA VAR INDICA PARAMETRO VARIÁVEL
* NOTE QUE SE TETA1 = VAR ==> DD1 = NUMERO, OU SEJA, OU A JUNTA
* E ROTORIA OUX (OU EXCLUSIVO) PRISMÁTICA

GEOM

ALFA 0 0
TETA VAR VAR

DD 0 0
LL 0.10 0.20

* PARAMETROS DINAMICOS (MATRIZ DE INERCIA HOMOGENEA MODIFICADA - MIHMOD)
* ESTA MATRIZ, QUE E SIMETRICA, TEM O SEGUINTE FORMATO:

*
* I XY XZ X
* J YZ Y
* K Z
* M
*

* ONDE OS PARAMETROS SAO:

* I, J, K, XY, XZ, YZ SAO REFERIDOS AO SISTEMA DE COORDENADAS DA
* LIGACAO RESPECTIVA MAS COLOCADO NO CENTRO DE MASSA.
* X, Y, Z SAO AS COORDENADAS DO CENTRO DE MASSA EM
* RELACAO O SISTEMA DA LIGACAO.

* M E A MASSA DO LINK

MIHMOD

I 0 0
 J 0 0
 K 0 0
 M 1 2
 XY 0 0
 XZ 0 0
 YZ 0 0
 X 0 0
 Y 0 0
 Z 0 0

* FIM DE ESPECIFICACAO

5.2.3 Resultados Obtidos

Ilustraremos, a seguir, algumas possibilidades do simulador, através de exemplos na forma de gráficos.² Faremos três exemplos para o Pêndulo Duplo que serão denominados PD (Ex. 1), PD (Ex. 2) e PD (Ex. 3).

A diferença entre eles é basicamente o tipo de trajetória utilizada, bem como a magnitude de velocidade desta trajetória.

Nos gráficos, valem as seguintes afirmações:

- No **MODINV**, adotamos os valores iniciais $\mathbf{q} = 0$ e $\dot{\mathbf{q}} = 0$. No **SIMULADOR**, fizemos posição e velocidade iniciais iguais às da trajetória de referência.
- Em todos os exemplos onde necessitou-se de integração, utilizamos o algoritmo com passo adaptativo.³
- No controle, para simular um erro de identificação, adotamos no arquivo do robô massas 20% maiores que as fornecidas no exemplo de arquivo acima, que supomos ser o arquivo do modelo. O passo utilizado foi de 0.005. Adotamos ainda, para as matrizes diagonais, os seguintes valores: $\Psi_{11} = 0.95$, $\Psi_{22} = 0.95$, $\Omega_{11} = 50 \text{ rad/s}$ e $\Omega_{22} = 50 \text{ rad/s}$.
- O $\delta\mathbf{q}$ e $\delta\dot{\mathbf{q}}$ utilizados pelo controlador foram calculados como a diferença entre a referência do instante **posterior** (ou seja, para onde quero que o robô vá) e o valor de **onde o robô está** (em termos de posição ou velocidade).

²Note como o valor das velocidades afeta o valor da componente de torque centrípeto+Coriolis.

³A tolerância relativa especificada foi de 10^{-6} .

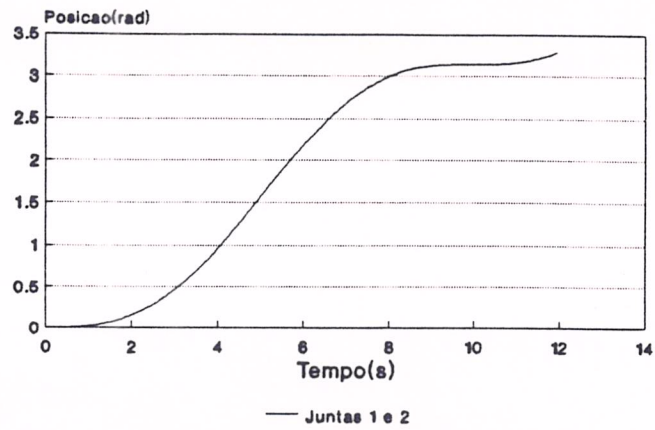


Figura 5.3: TRAJREF - Posição de Referência - PD (Ex. 1)

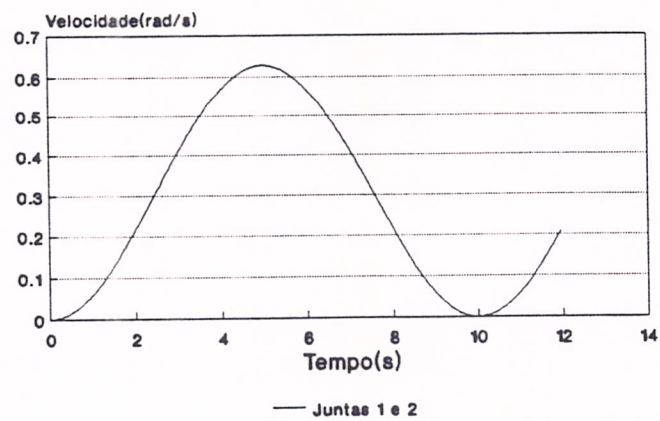


Figura 5.4: TRAJREF - Velocidade de Referência - PD (Ex. 1)

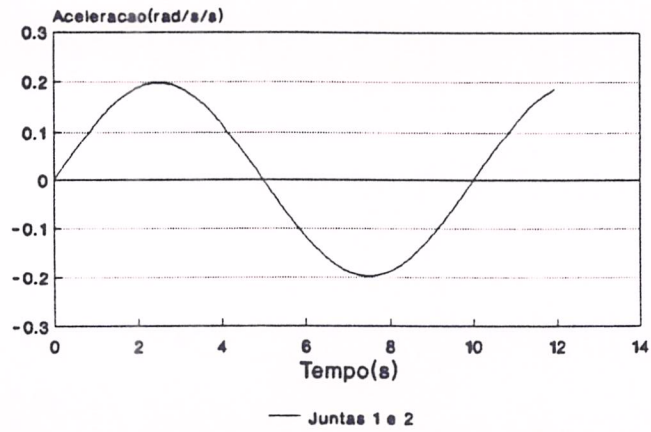


Figura 5.5: TRAJREF - Aceleração de Referência - PD (Ex. 1)

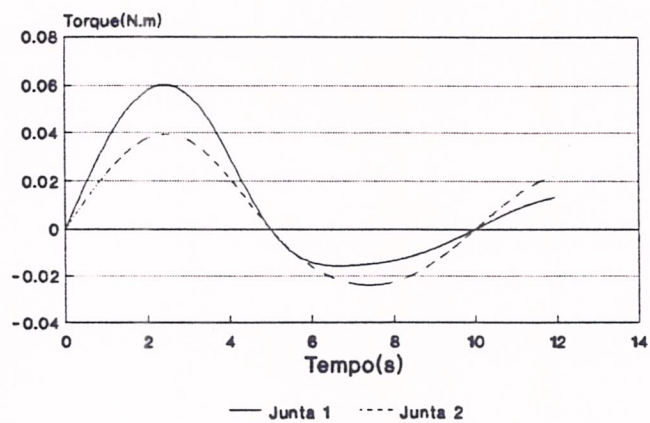


Figura 5.6: FEEDFORWARD - Torque Inercial - PD (Ex. 1)

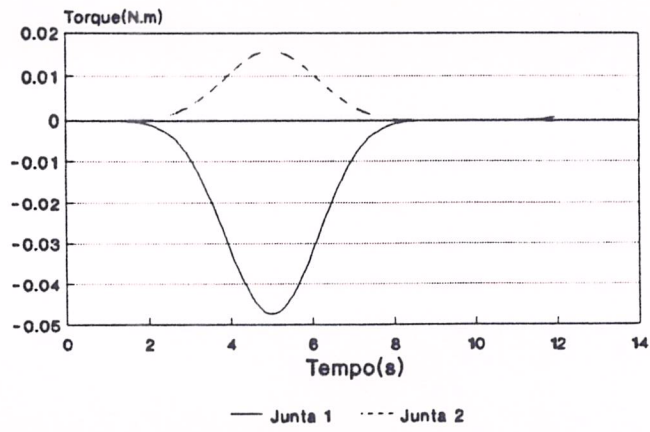


Figura 5.7: FEEDFORWARD - Torque Centrípeto + Coriolis - PD (Ex. 1)

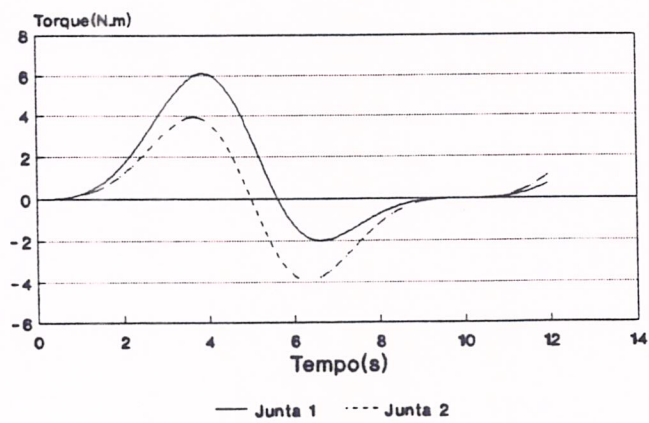


Figura 5.8: FEEDFORWARD - Torque Gravitacional - PD (Ex. 1)

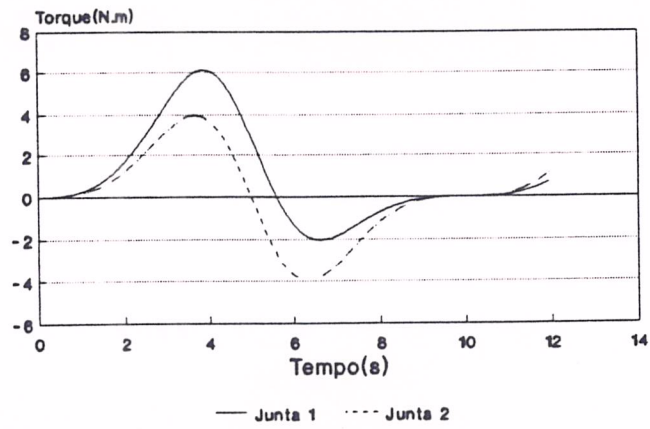


Figura 5.9: FEEDFORWARD - Torque Resultante - PD (Ex. 1)

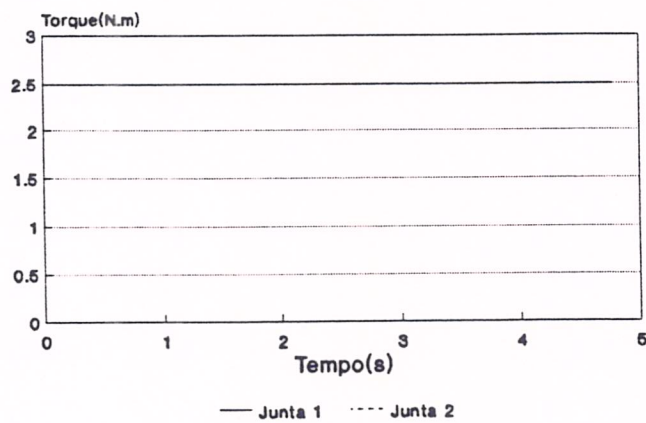


Figura 5.10: MODINV - Torque Aplicado - PD (Ex. 1)

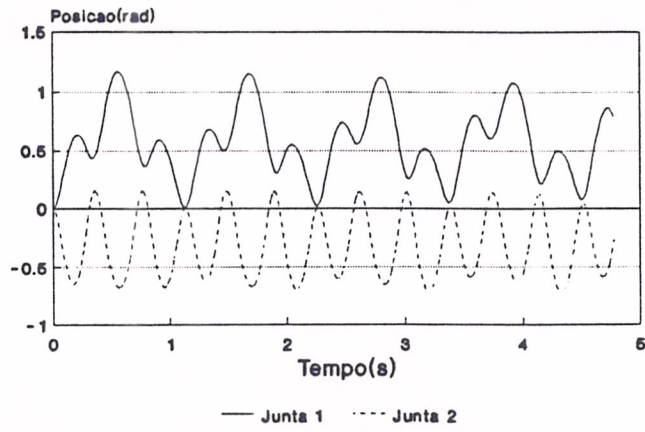


Figura 5.11: MODINV - Posição do Robô - PD (Ex. 1)

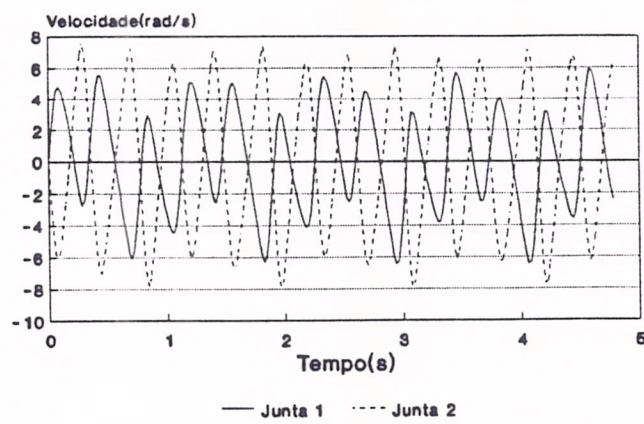


Figura 5.12: MODINV - Velocidade do Robô - PD (Ex. 1)

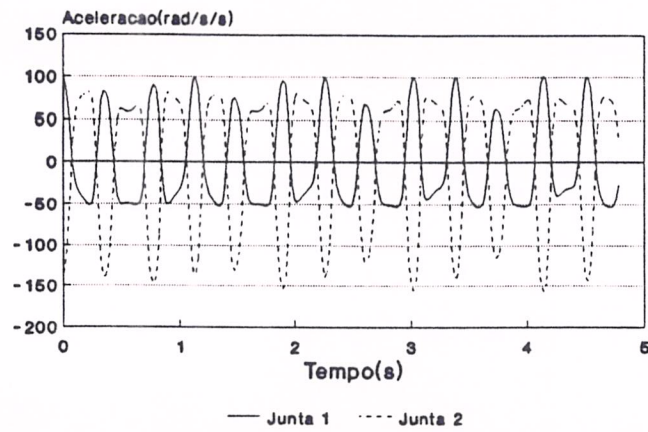


Figura 5.13: MODINV - Aceleração do Robô - PD (Ex. 1)

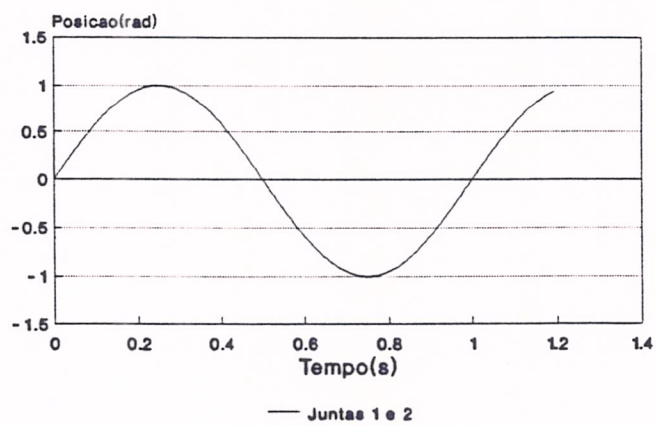


Figura 5.14: TRAJREF - Posição de Referência - PD (Ex. 2)

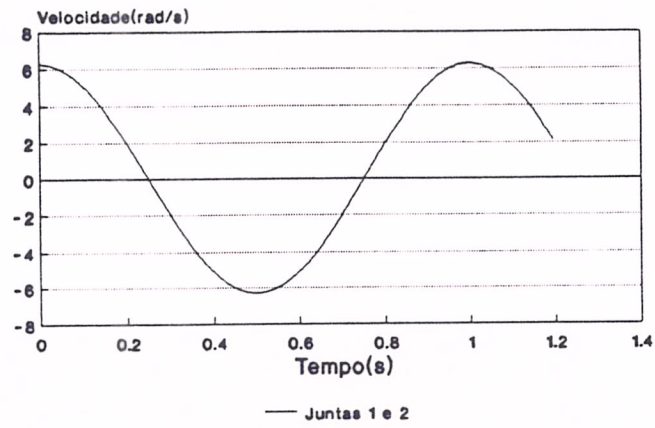


Figura 5.15: TRAJREF - Velocidade de Referência - PD (Ex. 2)

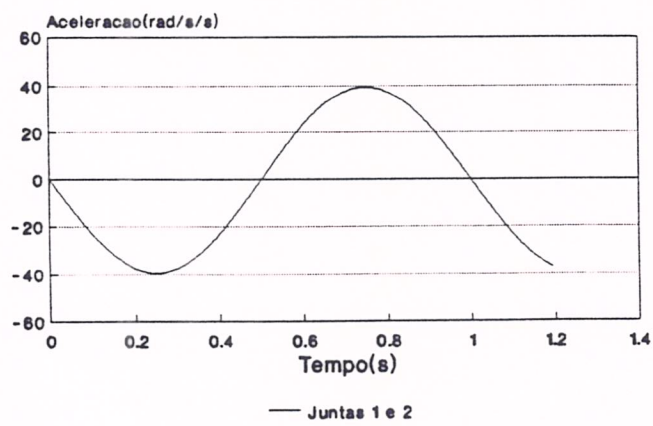


Figura 5.16: TRAJREF - Aceleração de Referência - PD (Ex. 2)

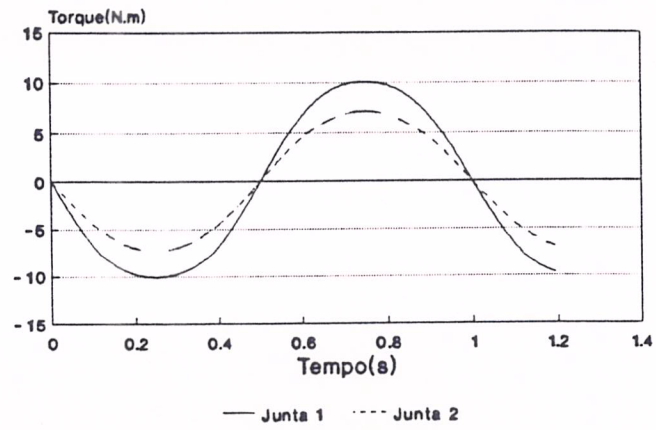


Figura 5.17: FEEDFORWARD - Torque Inercial - PD (Ex. 2)

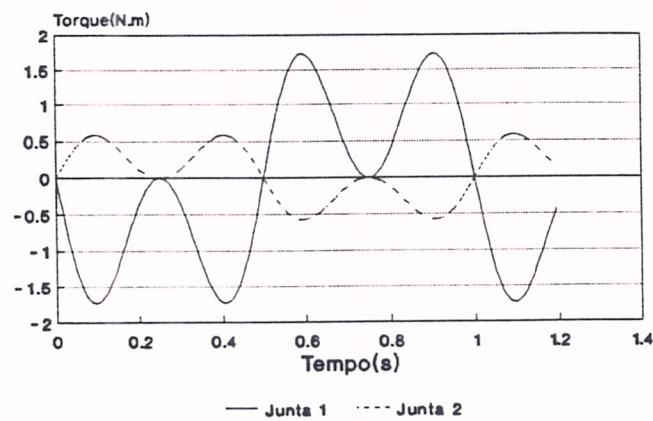


Figura 5.18: FEEDFORWARD - Torque Centrípeto + Coriolis - PD (Ex. 2)

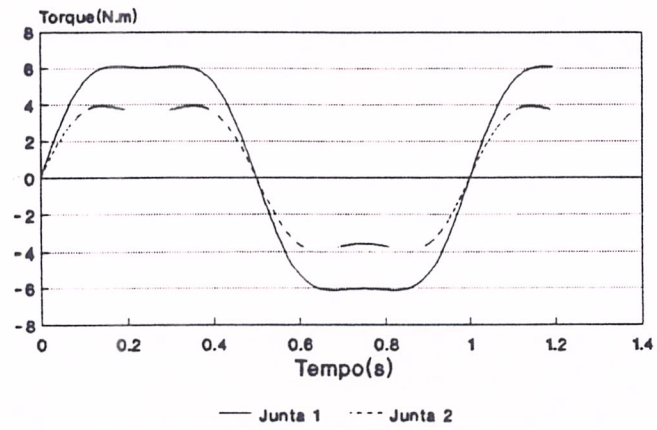


Figura 5.19: FEEDFORWARD - Torque Gravitacional - PD (Ex. 2)

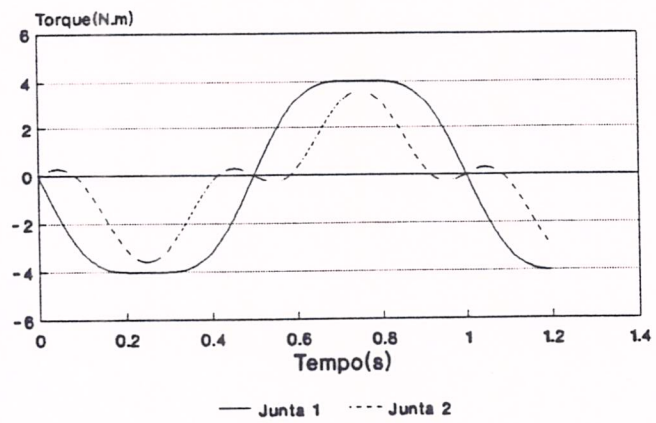


Figura 5.20: FEEDFORWARD - Torque Resultante - PD (Ex. 2)

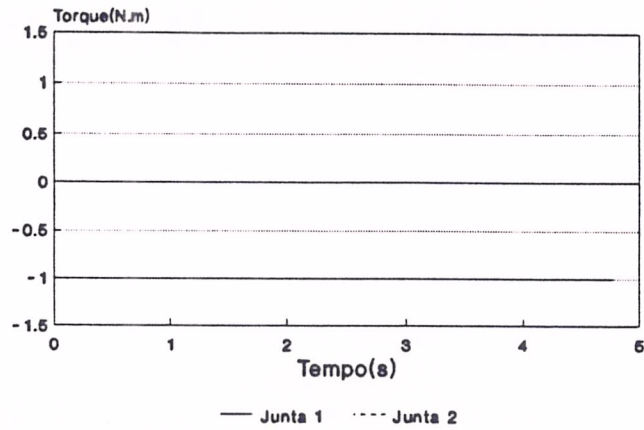


Figura 5.21: MODINV - Torque Aplicado - PD (Ex. 2)

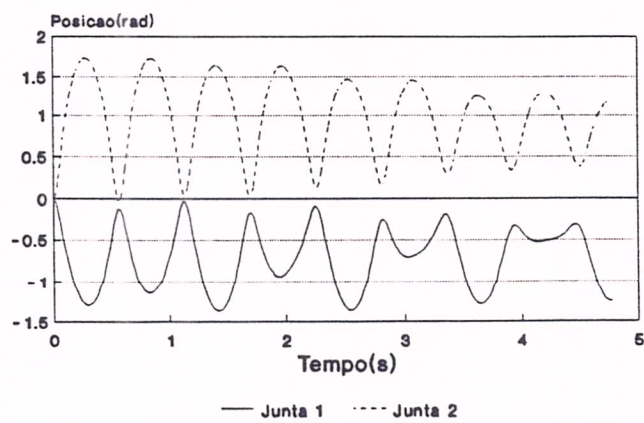


Figura 5.22: MODINV - Posição do Robô - PD (Ex. 2)

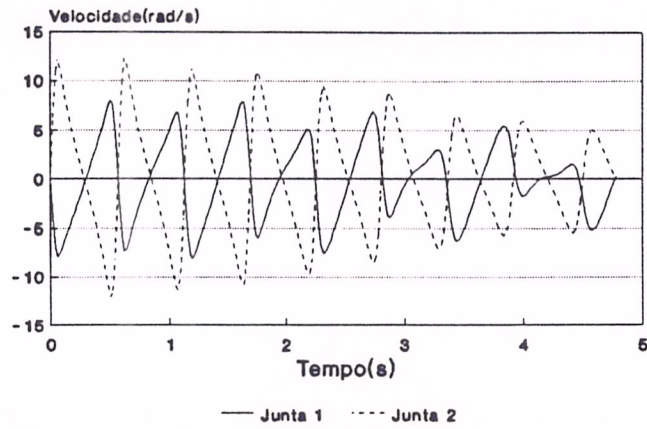


Figura 5.23: MODINV - Velocidade do Robô - PD (Ex. 2)

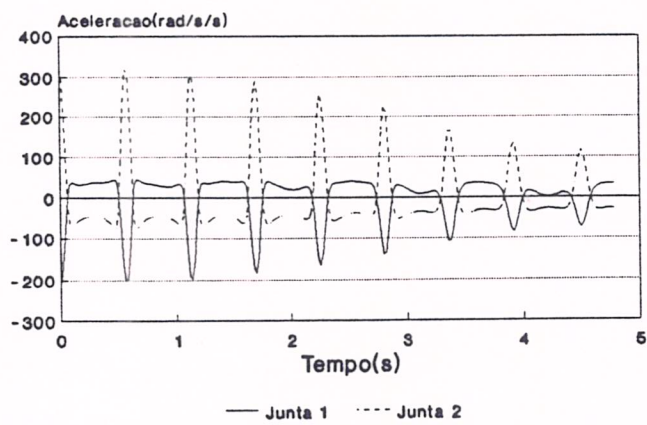


Figura 5.24: MODINV - Aceleração do Robô - PD (Ex. 2)

Na página seguinte, temos os exemplos da simulação utilizando o esquema de controle proposto. Deve-se notar que as matrizes do modelo linearizado são calculadas com base no arquivo do modelo. Além disso, os erros de quantização e a maneira de cálculo de δq e $\delta \dot{q}$ podem influenciar de modo dificilmente previsível os valores dos coeficientes das matrizes Ψ e Ω (que produzam bons resultados) utilizadas no esquema de controle. Perceba, através dos gráficos (principalmente de posição), o razoável desempenho do controlador (Lembre-se que δq e $\delta \dot{q}$ não são exatamente erros em relação à referência).

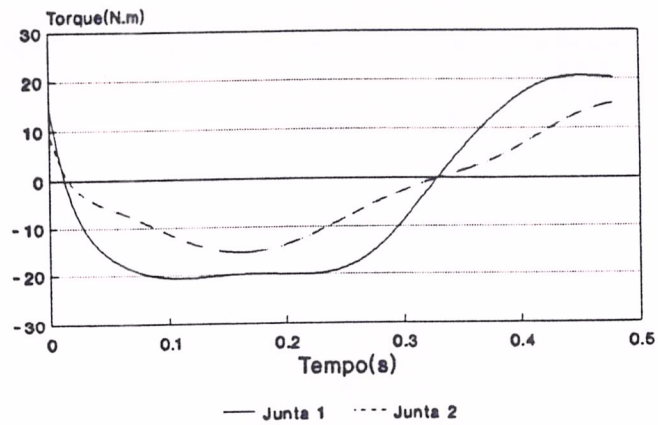


Figura 5.25: SIMULADOR - Torque Aplicado - PD (Ex. 3)

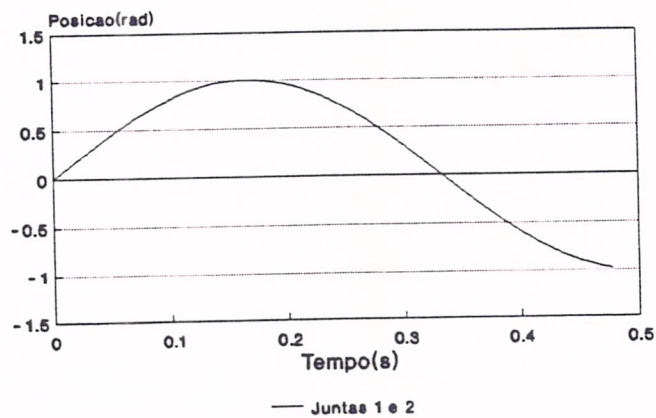


Figura 5.26: TRAJREF - Posição de Referência - PD (Ex. 3)

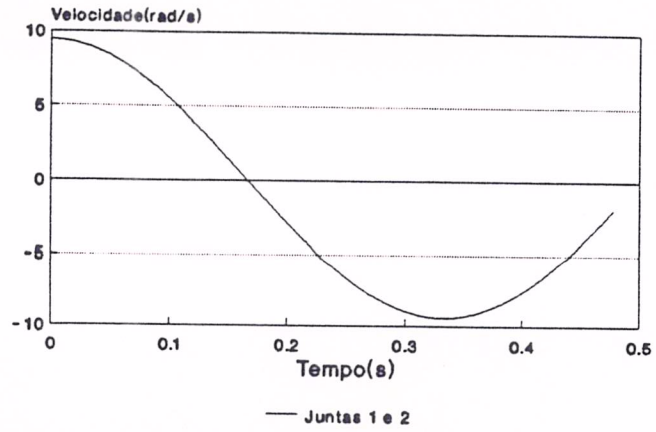


Figura 5.27: TRAJREF - Velocidade de Referência - PD (Ex. 3)

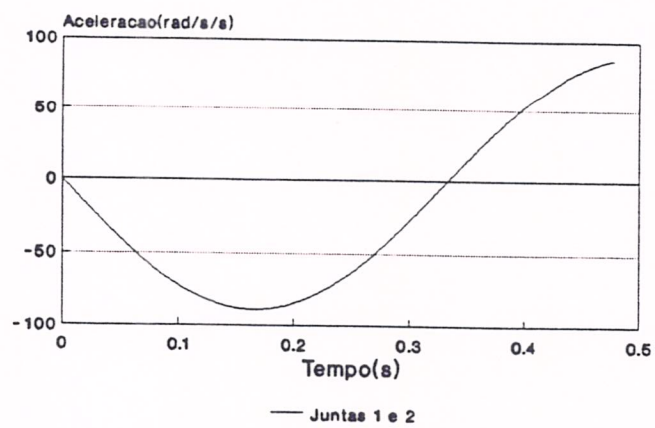


Figura 5.28: TRAJREF - Aceleração de Referência - PD (Ex. 3)

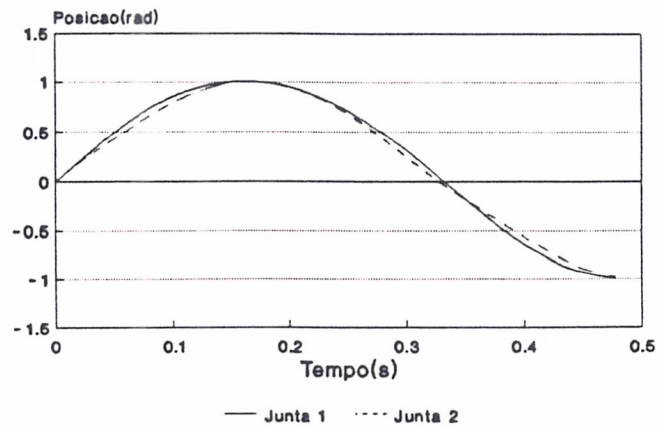


Figura 5.29: SIMULADOR - Posição do Robô - PD (Ex. 3)

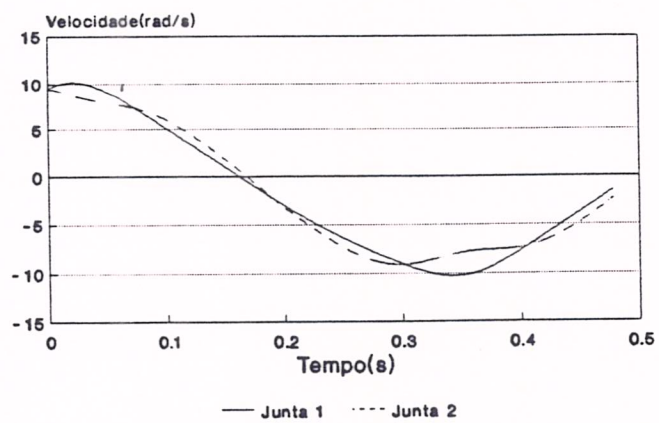


Figura 5.30: SIMULADOR - Velocidade do Robô - PD (Ex. 3)

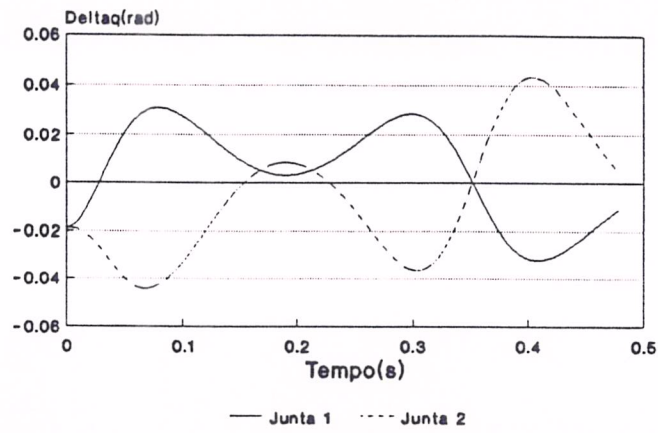


Figura 5.31: SIMULADOR - Deltaq - PD (Ex. 3)

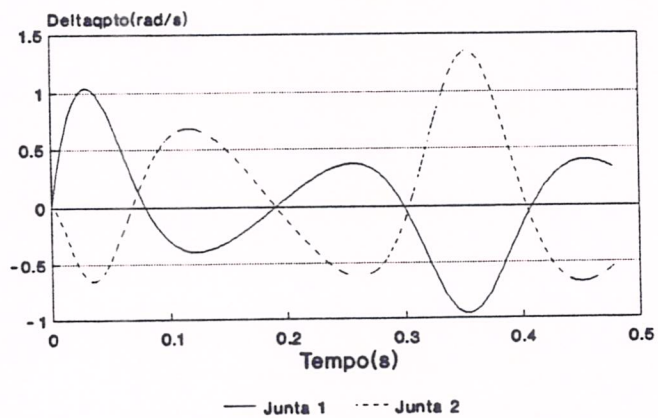


Figura 5.32: SIMULADOR - Deltaqpto - PD (Ex. 3)

Os resultados acima mostram concordância com resultados calculados a partir das fórmulas analíticas. No caso do exemplo de simulação (SIMULADOR), apesar de perturbarmos em 20% as massas do robô, a correção de controle somada ao “feedforward” do modelo sintetiza aproximadamente o “feedforward” do robô, ou seja, o controle compensa as falhas de identificação que colocamos propositalmente nesta simulação.

5.3 ROBÔ STANFORD

5.3.1 Arquivo de Dados do Robô Stanford

```

* OBSERVACOES IMPORTANTES
* Angulos serao dados de -180 a 180 graus senao serao recusados
* Os LL deverao ser valores >= 0 senao serao recusados
* ESTE ROBO ESTA EM PAUL P.56

* CONSTANTE GRAVITACIONAL DO PROBLEMA G

G 0 0 -9.8

* NOME DO ROBO

ROBO STANFORD

* NUMERO DE LIGAMENTOS

NLIGS 6

* PARAMETROS GEOMETRICOS
* EXEMPLO: ALFA <ALFA1> <ALFA2> ...
* A PALAVRA VAR INDICA PARAMETRO VARIÁVEL
* NOTE QUE SE TETA1 = VAR ==> DD1 = NUMERO, OU SEJA, OU A JUNTA
* E ROTORIA OUX (OU EXCLUSIVO) PRISMÁTICA

GEOM

ALFA  -90    90    0   -90    90    0
TETA  VAR   VAR   0   VAR   VAR   VAR
DD    0    0.25  VAR   0    0    0
LL    0    0    0    0    0    0

* PARAMETROS DINAMICOS (MATRIZ DE INERCIA HOMOGENEA MODIFICADA)
* ESTA MATRIZ, QUE E SIMETRICA, TEM O SEGUINTE FORMATO:
*
* I  XY  XZ  X

```

```
*      J  YZ  Y
*           K  Z
*           M
*
```

* ONDE OS PARAMETROS SAO:

* I, J, K, XY, XZ, YZ SAO REFERIDOS AO SISTEMA DE COORDENADAS DA

* LIGACAO RESPECTIVA MAS COLOCADO NO CENTRO DE MASSA.

* X, Y, Z SAO AS COORDENADAS DO CENTRO DE MASSA EM

* RELACAO O SISTEMA DA LIGACAO.

* M E A MASSA DO LINK

MIHMOD

I	0.276	0.108	2.51	0.002	0.003	0.013
J	0.2522	-0.0377	2.51	0.0009	0.003	0.013
K	-0.0424	0.1	-1.7065	0.001	-0.0016	-0.012
M	9.29	5.01	4.25	1.08	0.63	0.51
XY	0	0	0	0	0	0
XZ	0	0	0	0	0	0
YZ	0.018	0	0	0.0001	0	0
X	0	0	0	0	0	0
Y	0.0175	-0.1054	0	0.0092	0	0
Z	-0.1105	0	-0.6447	-0.0054	0.0566	0.1554

* FIM DE ESPECIFICACAO

5.3.2 Resultados Obtidos

Iremos a seguir, ilustrar o robô STANFORD (Fig. 5.33), através de seu "feedforward". Deve-se notar que o "feedforward" deste robô demorou aproximadamente 10s para 256 pontos da trajetória em um PC-AT 386 de 20 MHz (O "feedforward" do Pêndulo Duplo demorou aproximadamente 2.5s). Este dado é o mais importante para avaliar o desempenho do simulador, visto que todos os algoritmos utilizam o "feedforward" ou parte dele em seus procedimentos.

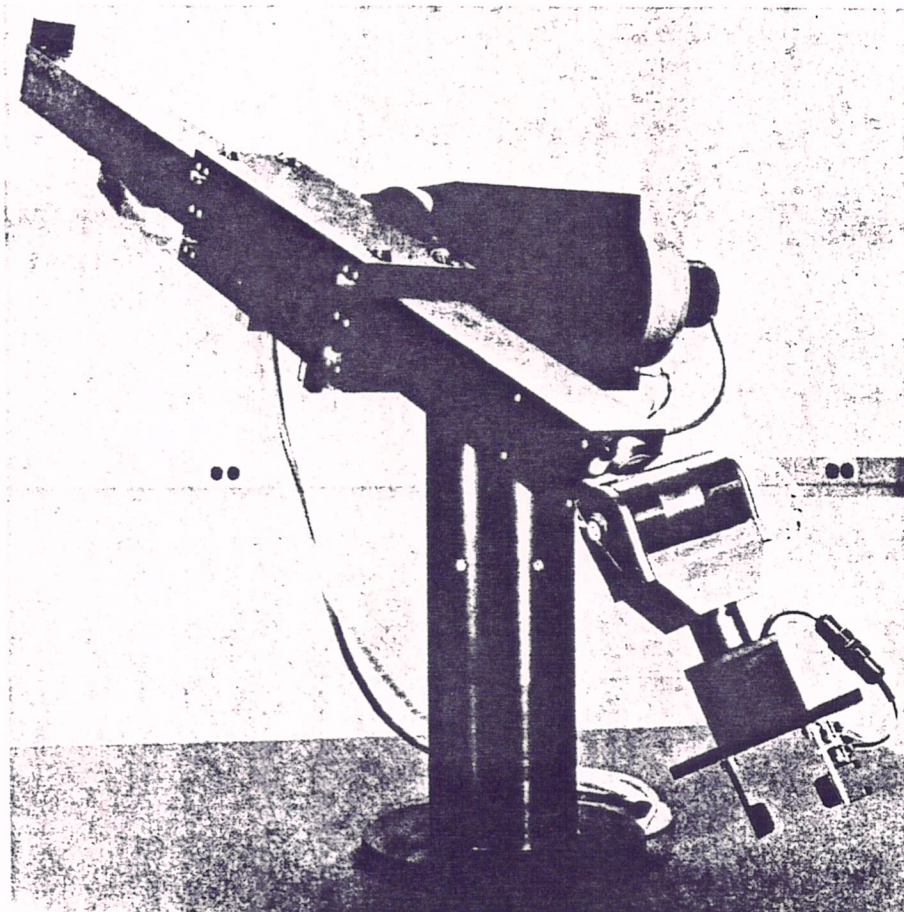
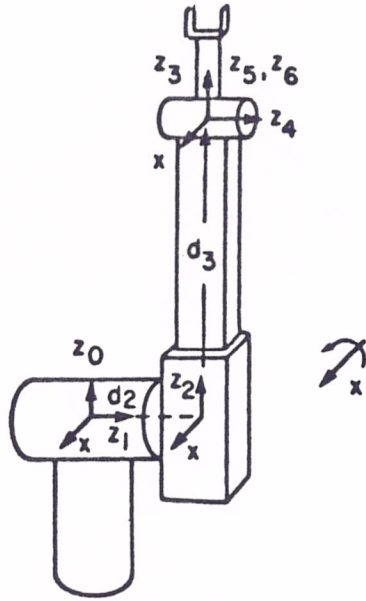


Figura 5.33: Robô Manipulador STANFORD

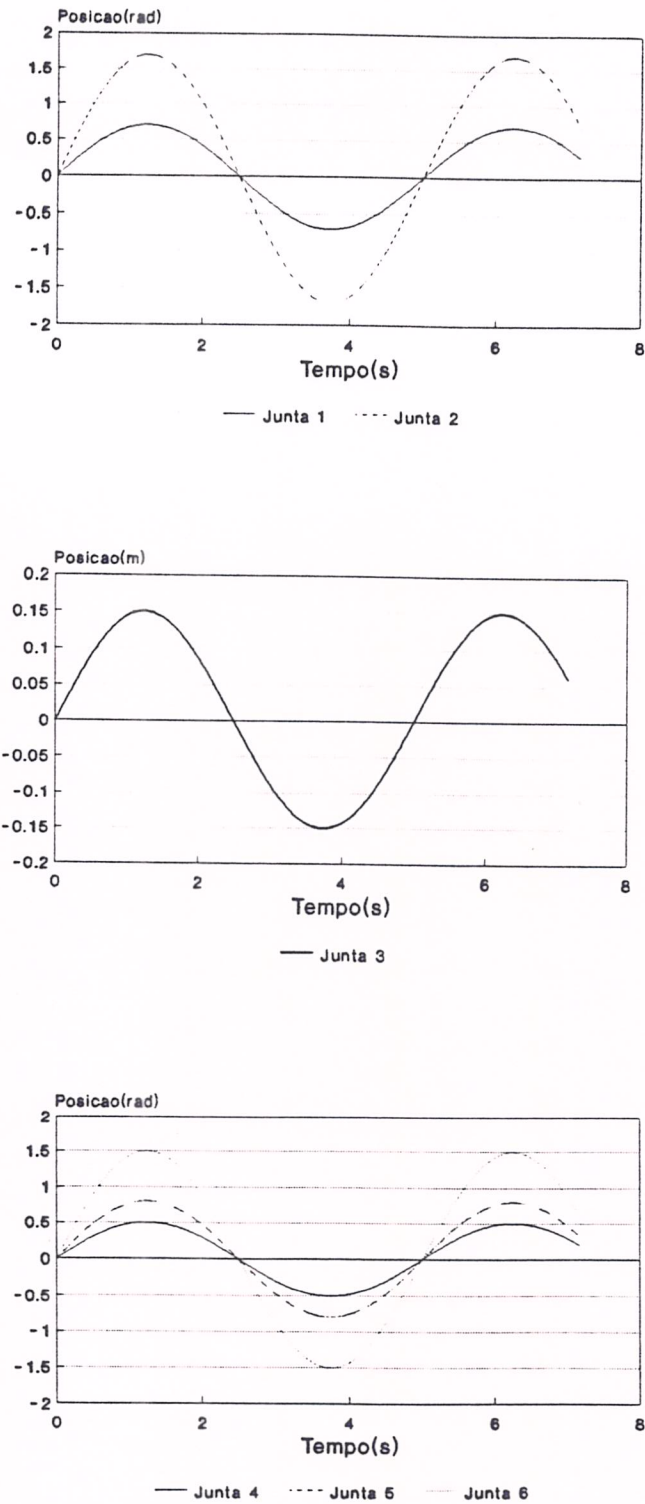


Figura 5.34: TRAJREF - Posição de Referência - STANFORD

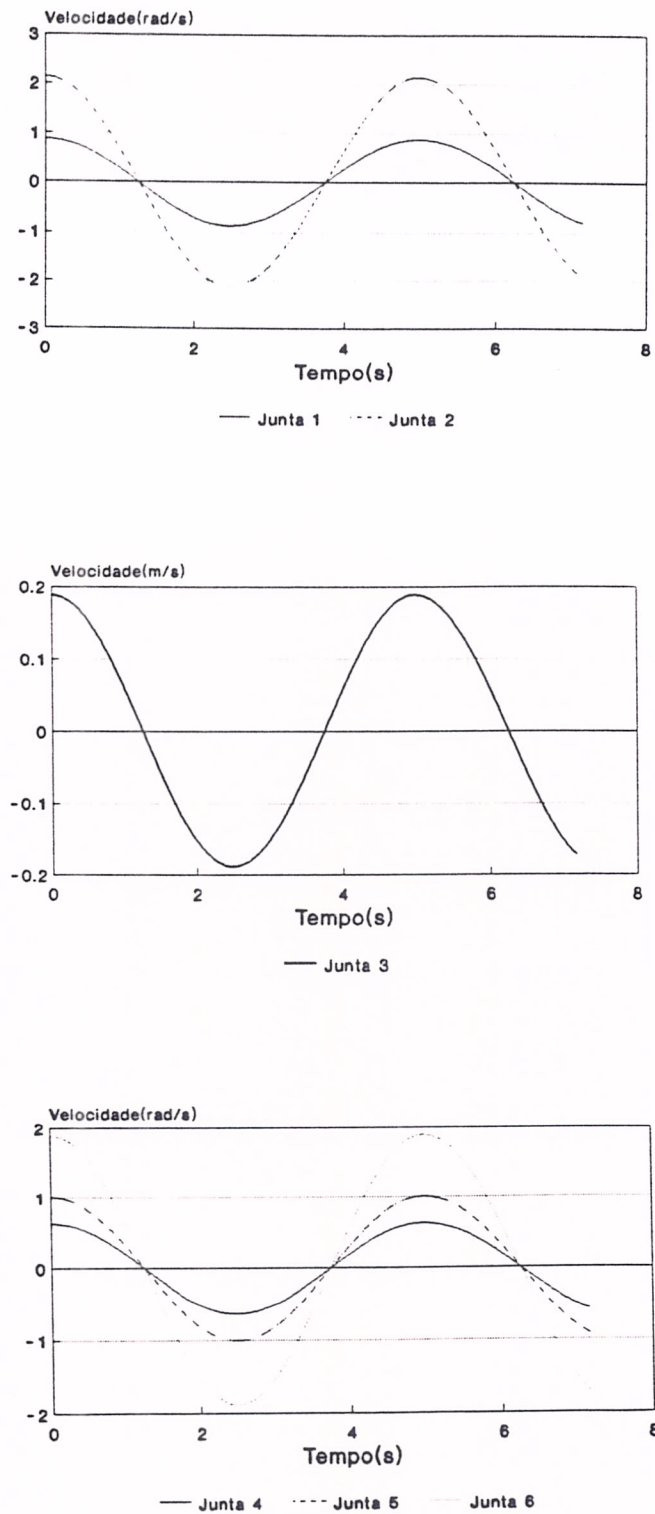


Figura 5.35: TRAJREF - Velocidade de Referência - STANFORD

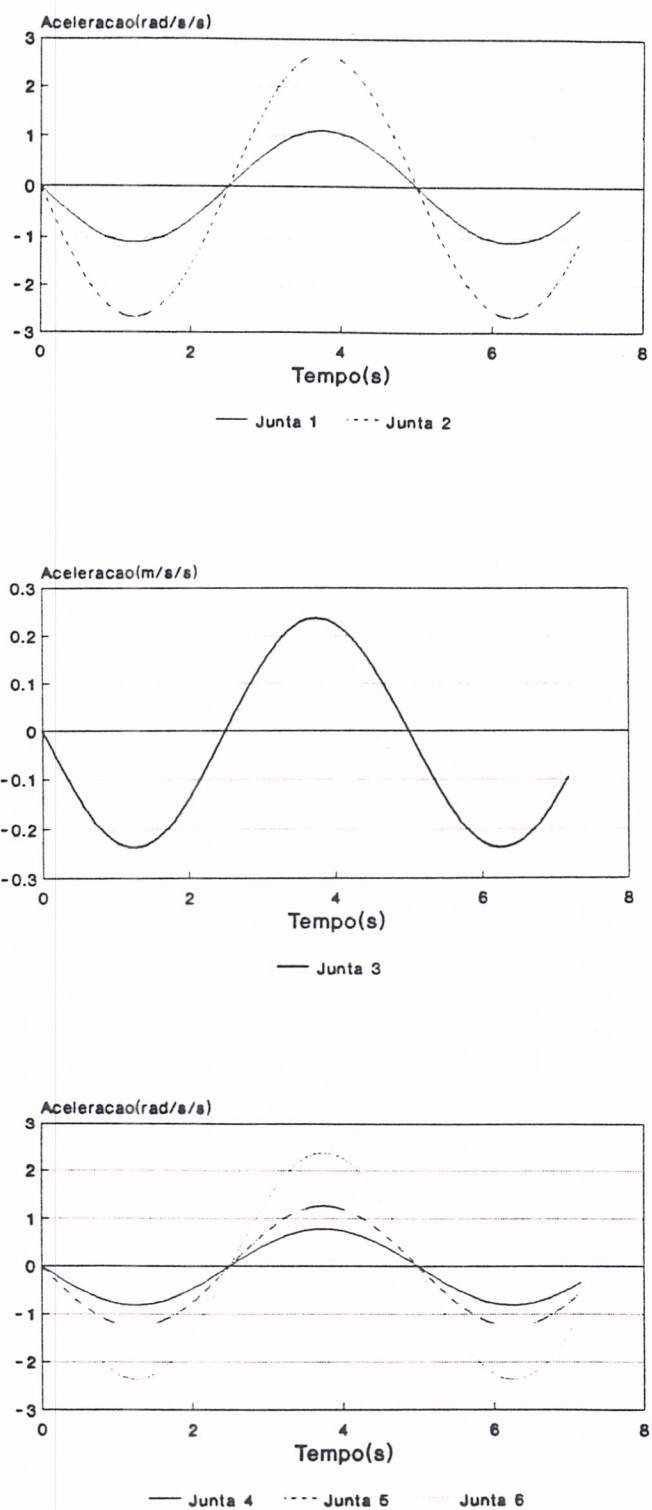


Figura 5.36: TRAJREF - Aceleração de Referência - STANFORD

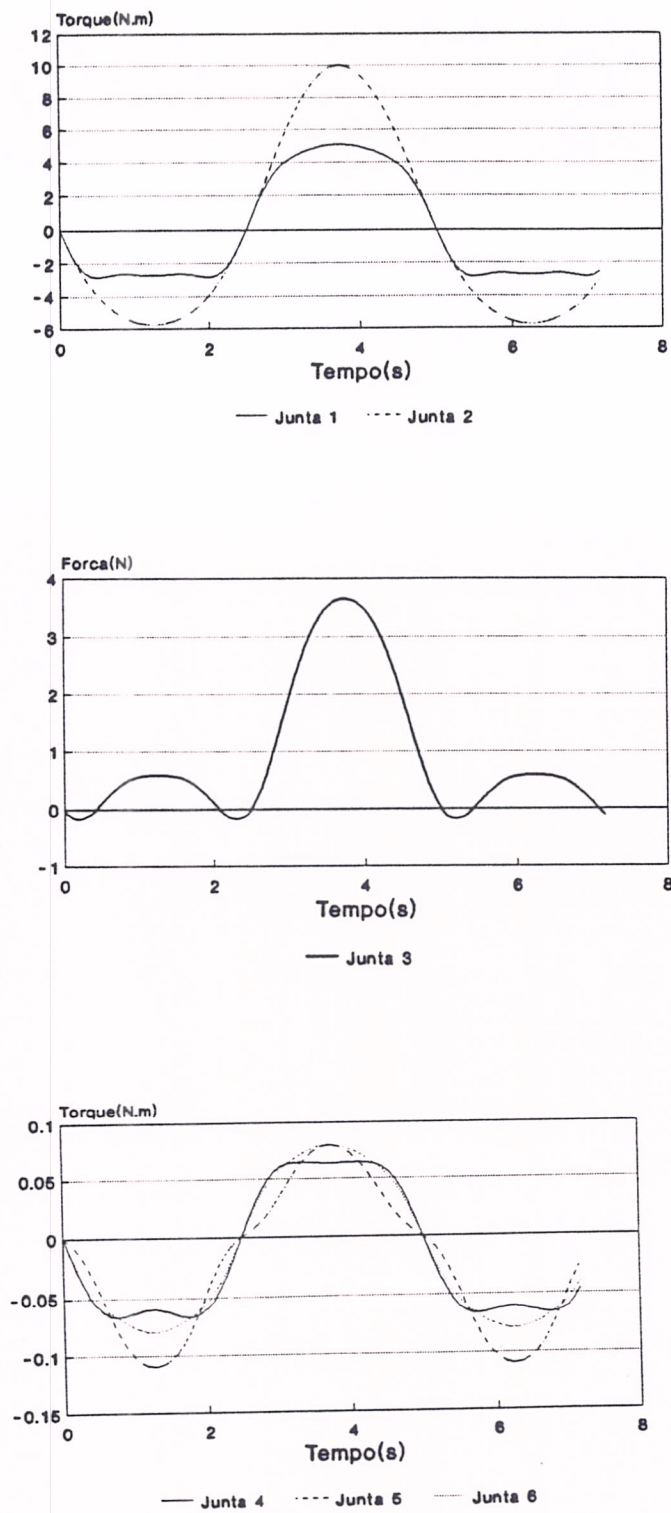


Figura 5.37: FEEDFORWARD - Força Inercial - STANFORD

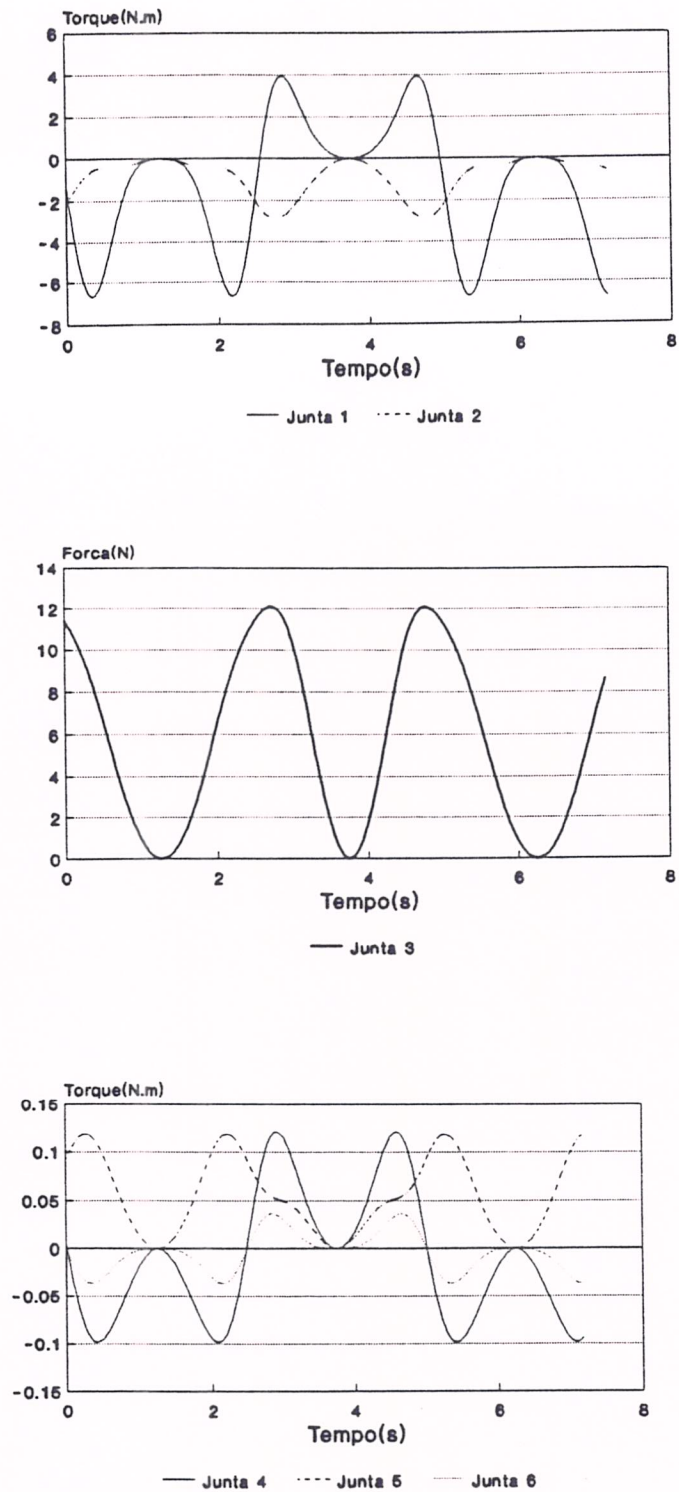


Figura 5.38: FEEDFORWARD - Força Centrípeta + Coriolis - STANFORD

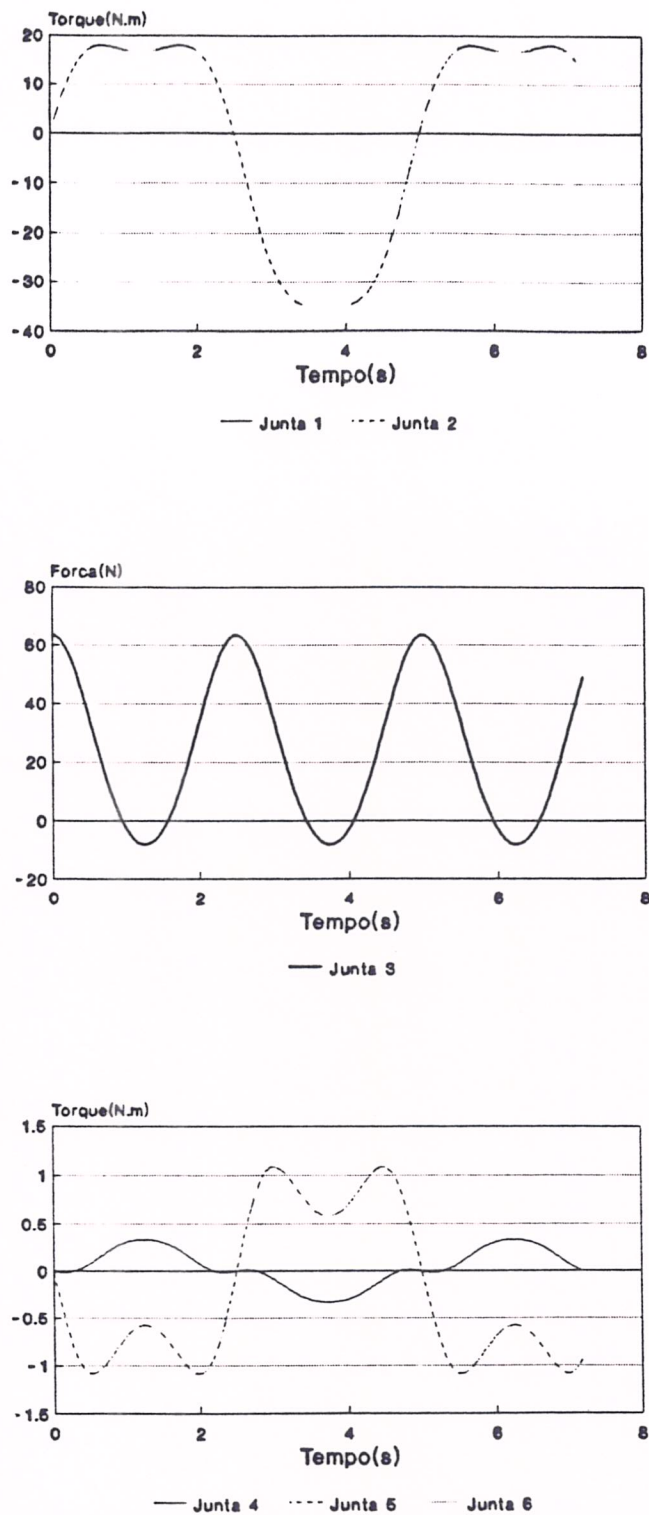


Figura 5.39: FEEDFORWARD - Força Gravitacional - STANFORD

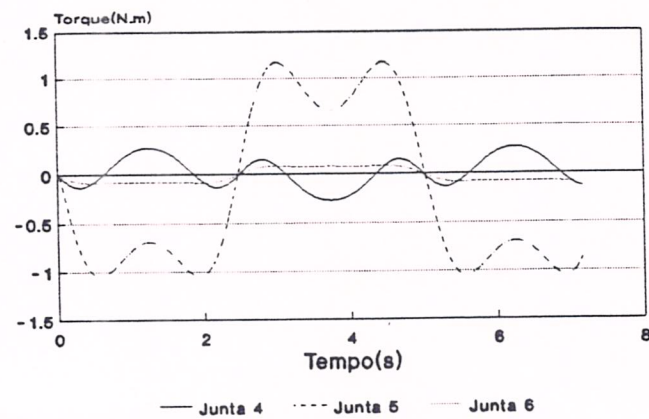
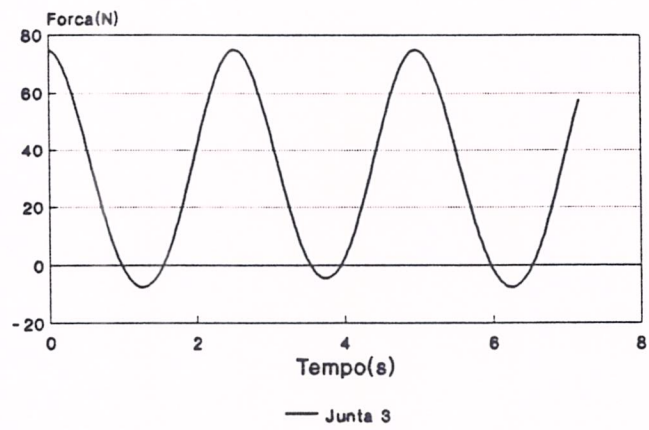
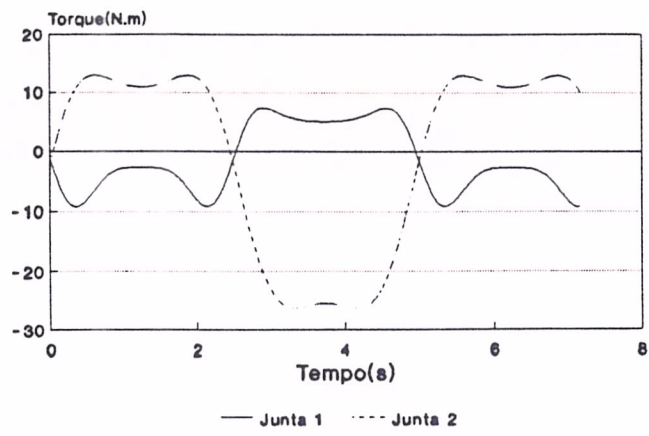


Figura 5.40: FEEDFORWARD - Força Resultante - STANFORD

Capítulo 6

CONCLUSÕES

Nesta dissertação, descreve-se um simulador para cálculo de modelos dinâmicos de manipuladores.

O programa computacional implementado calcula, de modo eficiente, os seguintes modelos:

- Modelo dinâmico direto (MDD).
- Modelo dinâmico inverso (MDI).
- Modelo dinâmico direto linearizado (MDL).

A eficiência, que norteou a realização deste trabalho, não foi motivada apenas pelo desejo de acelerar o algoritmo. Mais do que isso, trata-se de um fator determinante para viabilizar a utilização prática do programa computacional implementado e, com isso, permitir o desenvolvimento de robôs mais rápidos e precisos.

No simulador, o usuário pode examinar os torques necessários para uma dada trajetória de referência. Pode também, a partir de torques aplicados às juntas do robô, verificar o movimento realizado, monitorando diversas variáveis, tais como posição e velocidade de cada junta.

Foram implementados, ainda, um módulo gráfico em alta resolução que serve para visualizar e imprimir, no próprio ambiente do programa, a evolução das variáveis de interesse e um módulo para gravação dos arquivos de resultados em disco, para posterior utilização.

O programa opera com qualquer robô de cadeia aberta mostrando grande eficiência em termos de tempo de cálculo, para o que foi necessário pesquisar diversos algoritmos para realizar os módulos deste simulador.

Devemos enfatizar que a grande complexidade das equações dinâmicas de robôs torna imprescindível o uso de algoritmos velozes. Vimos que em um dos mais simples

casos de “manipulador”, como o do pêndulo duplo, já temos equações suficientemente complexas demonstrando esta exigência.

O trabalho aqui apresentado admite extensões, dentre as quais podemos citar:

- Integração com módulos cinemáticos (Cinemática Inversa). Neste caso, o usuário poderia especificar a trajetória diretamente em coordenadas operacionais [GOR 84]. Por exemplo, ele poderia especificar a posição em coordenadas cartesianas e a orientação em ângulos de **Bryant**.
- Inclusão de parâmetros do motor, saturação e de modelos de atrito (o atrito viscoso é facilmente modelado, mas o atrito estático não o é, principalmente a velocidades muito próximas de 0).
- Implementação de outros esquemas de controle, tais como: PID, “Sliding Mode”, Torque computado, etc. . .
- Integração com modeladores de sólidos para efeitos de animação. Com isto, conseguiremos visualizar em três dimensões o movimento do manipulador.
- Generalização para robôs de cadeia fechada, cuja modelagem utiliza outro conjunto de parâmetros geométricos (para cadeia aberta, utilizamos os parâmetros de **Denavit-Hartenberg**).
- Integração com programas simuladores de células de manufatura.

Também foi possível, no decorrer deste trabalho, desenvolver demonstrações mais claras e rápidas de algumas relações e detectar alguns enganos existentes em trabalhos de alto nível.

Finalmente, acreditamos que este simulador será um auxiliar valioso no desenvolvimento de novos manipuladores robóticos, bem como no ensino e pesquisa de métodos de modelagem dinâmica e controle.

Apêndice A

RELAÇÕES MATRICIAIS IMPORTANTES

A.1 DEFINIÇÕES

Dado o vetor $\alpha = (\alpha_x, \alpha_y, \alpha_z)^T$, seu produto vetorial com outro vetor pode ser feito multiplicando-se a matriz:

$$\tilde{\alpha} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\alpha_z & \alpha_y \\ \alpha_z & 0 & -\alpha_x \\ -\alpha_y & \alpha_x & 0 \end{bmatrix}$$

por este vetor. Ou seja, para um vetor qualquer β , temos:

$$\alpha \wedge \beta = \tilde{\alpha} \beta \tag{A.1}$$

Esta matriz anti-simétrica ($\tilde{\alpha}^T = -\tilde{\alpha}$) é denominada matriz do pré-produto vetorial de α .

Dada a matriz:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

definimos a matriz auxiliar $\overline{\mathbf{A}}$ por:

$$\overline{\mathbf{A}} \stackrel{\text{def}}{=} \begin{bmatrix} a_{22} + a_{33} & -a_{12} & -a_{13} \\ -a_{21} & a_{11} + a_{33} & -a_{23} \\ -a_{31} & -a_{32} & a_{11} + a_{22} \end{bmatrix}$$

A.2 TEOREMAS

Os 3 primeiros teoremas são de demonstração imediata, além de serem muito conhecidos. Assim sendo não apresentaremos a demonstração.

Teorema A.2.1 *Sejam α e β vetores 3×1 . Então, vale:*

$$\text{tr}(\alpha\beta^T) = \alpha^T\beta \quad (\text{A.2})$$

Teorema A.2.2 *Para qualquer matriz de rotação R e para quaisquer 2 vetores 3×1 α e β , vale:*

$$R^T R = I_3 \quad (\text{A.3})$$

$$R R^T = I_3 \quad (\text{A.4})$$

$$R^T(\alpha \wedge \beta) = (R^T\alpha) \wedge (R^T\beta) \quad (\text{A.5})$$

Teorema A.2.3 *Sejam α , β e γ vetores 3×1 . Então, vale:*

$$\alpha^T(\beta \wedge \gamma) = \beta^T(\gamma \wedge \alpha) = \gamma^T(\alpha \wedge \beta) \quad (\text{A.6})$$

Teorema A.2.4 *Sejam α e β vetores 3×1 e A uma matriz 3×3 . Então, vale:*

$$\text{tr}(\tilde{\alpha} A \tilde{\beta}^T) = \beta^T \overline{A} \alpha \quad (\text{A.7})$$

Demonstração:

Sendo:

$$A \tilde{\beta}^T = \begin{bmatrix} -a_{12}\beta_z + a_{13}\beta_y & a_{11}\beta_z - a_{13}\beta_x & -a_{11}\beta_y + a_{12}\beta_x \\ -a_{22}\beta_z + a_{23}\beta_y & a_{21}\beta_z - a_{23}\beta_x & -a_{21}\beta_y + a_{22}\beta_x \\ -a_{32}\beta_z + a_{33}\beta_y & a_{31}\beta_z - a_{33}\beta_x & -a_{31}\beta_y + a_{32}\beta_x \end{bmatrix} \quad (\text{A.8})$$

temos:

$$\begin{aligned} \text{tr}(\tilde{\alpha} A \tilde{\beta}^T) &= [(a_{22} + a_{33})\beta_x - a_{21}\beta_y - a_{31}\beta_z] \alpha_x \\ &+ [(a_{11} + a_{33})\beta_y - a_{12}\beta_x - a_{32}\beta_z] \alpha_y \\ &+ [(a_{11} + a_{22})\beta_z - a_{13}\beta_x - a_{23}\beta_y] \alpha_z \\ &= \beta^T \overline{A} \alpha \end{aligned}$$

Teorema A.2.5 *Sejam α , β e γ vetores 3×1 e A uma matriz simétrica 3×3 . Então:*

$$\text{tr}(\tilde{\alpha} A \tilde{\beta}^T \tilde{\gamma}^T) = \gamma^T A (\beta \wedge \alpha) \quad (\text{A.9})$$

Demonstração:

Por A.7, temos:

$$\text{tr}(\tilde{\alpha} A \tilde{\beta}^T \tilde{\gamma}^T) = \gamma^T \overline{\overline{A \tilde{\beta}^T}} \alpha$$

Pela simetria de A e usando a expressão da matriz A.8, obtemos:

$$\overline{\overline{A \tilde{\beta}^T}} = -A \tilde{\beta}^T$$

Lembrando a anti-simetria da matriz do pré-produto vetorial, ficamos com:

$$\text{tr}(\tilde{\alpha} A \tilde{\beta}^T \tilde{\gamma}^T) = \gamma^T A \tilde{\beta} \alpha = \gamma^T A (\beta \wedge \alpha)$$

Teorema A.2.6 *Sejam A uma matriz 3×3 qualquer e R uma matriz de rotação. Então:*

$$R^T \overline{\overline{A}} R = \overline{\overline{R^T A R}} \quad (\text{A.10})$$

Demonstração:

Para qualquer matriz B (3×3), temos:

$$\overline{\overline{B}} = -B + \text{tr}(B) I_3$$

onde I_3 é a matriz identidade de ordem 3.

Se fizermos $B = R^T A R$, então:

$$\text{tr}(B) = \sum_{j,k=1}^3 a_{jk} r_{j1} r_{k1} + a_{jk} r_{j2} r_{k2} + a_{jk} r_{j3} r_{k3}$$

Levando em conta o fato de R ser ortonormal, podemos escrever:

$$\text{tr}(B) = \sum_{j=1}^3 a_{jj} (r_{j1}^2 + r_{j2}^2 + r_{j3}^2) = \text{tr}(A)$$

Ou seja, o traço da matriz é invariante por esta transformação, o que já era esperado, pois trata-se de uma transformação de similaridade.

Portanto:

$$\overline{\overline{R^T A R}} = -R^T A R + \text{tr}(A) I_3 = R^T [-A + \text{tr}(A) I_3] R = R^T \overline{\overline{A}} R$$

Apêndice B

DEMONSTRAÇÃO DE RELAÇÕES CINEMÁTICAS

A seguir, é suposto $i \leq p$ (pois T_p só depende das juntas q_1, \dots, q_p).

Vamos então fornecer duas possíveis demonstrações para 2.7 e conseqüentemente para 2.8 e 2.9, pois daí basta derivar sucessivamente 2.7 lembrando que ($i \leq j \leq k$) e considerando de que juntas dependem os " Δ ".

B.1 PRIMEIRO MODO

Uma forma de demonstrar as relações acima é analisando como fica o sistema ortogonal p após um deslocamento infinitesimal δq_i , por considerações vetoriais. Por este modo chegamos aos resultados de uma maneira quase imediata, como será mostrado a seguir.

Com base na figura B.1, podemos escrever a variação diferencial em T_p por unidade de q_i como a seguir.

B.1.1 Caso A - Juntas de Rotação

Neste caso,

$$\frac{\partial T_p}{\partial q_i} = \begin{bmatrix} \mathbf{a}_{i-1} \wedge \mathbf{n}_p & \vdots & \mathbf{a}_{i-1} \wedge \mathbf{o}_p & \vdots & \mathbf{a}_{i-1} \wedge \mathbf{a}_p & \vdots & \mathbf{a}_{i-1} \wedge (\mathbf{p}_p - \mathbf{p}_{i-1}) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \end{bmatrix}$$

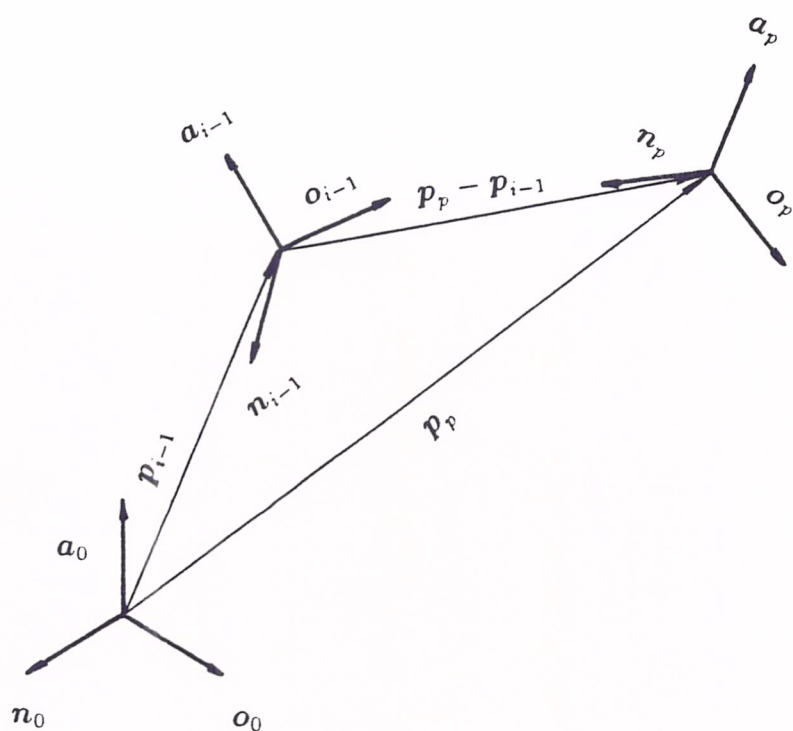


Figura B.1: Sistemas de Coordenadas

Ou seja,

$$\frac{\partial \mathbf{T}_p}{\partial q_i} = \begin{bmatrix} \tilde{\mathbf{a}}_{i-1} \mathbf{n}_p & \vdots & \tilde{\mathbf{a}}_{i-1} \mathbf{o}_p & \vdots & \tilde{\mathbf{a}}_{i-1} \mathbf{a}_p & \vdots & \tilde{\mathbf{a}}_{i-1} \mathbf{p}_p + \mathbf{p}_{i-1} \wedge \mathbf{a}_{i-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \end{bmatrix}$$

B.1.2 Caso B - Juntas de Translação

Neste caso, é imediato que:

$$\frac{\partial \mathbf{T}_p}{\partial q_i} = \begin{bmatrix} 0 & \vdots & 0 & \vdots & 0 & \vdots & \mathbf{a}_{i-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \end{bmatrix}$$

B.2 SEGUNDO MODO

Outra maneira de demonstrar, de uma forma puramente algébrica, é explicitada a seguir:

Como:

$$\frac{\partial \mathbf{T}_p}{\partial q_i} = \mathbf{A}_1 \cdots \mathbf{A}_{i-1} \frac{\partial \mathbf{A}_i}{\partial q_i} \mathbf{A}_{i+1} \cdots \mathbf{A}_p$$

e também:

$$\frac{\partial \mathbf{A}_i}{\partial q_i} = \mathbf{Q}_i \mathbf{A}_i$$

onde temos ${}^{i-1}\mathbf{a}_{i-1} = (0, 0, 1)^T$ e:

$$\mathbf{Q}_i \stackrel{\text{def}}{=} \begin{bmatrix} {}^{i-1}\tilde{\mathbf{a}}_{i-1} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \quad \text{se a junta } i \text{ for de rotação}$$

$$\mathbf{Q}_i \stackrel{\text{def}}{=} \begin{bmatrix} 0 & \vdots & {}^{i-1}\mathbf{a}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \quad \text{se a junta } i \text{ for de translação}$$

podemos escrever:

$$\frac{\partial \mathbf{T}_p}{\partial q_i} = \mathbf{A}_1 \cdots \mathbf{A}_{i-1} \mathbf{Q}_i \mathbf{A}_i \mathbf{A}_{i+1} \cdots \mathbf{A}_p = (\mathbf{T}_{i-1} \mathbf{Q}_i \mathbf{T}_{i-1}^{-1}) \mathbf{T}_p$$

Mas:

$$\Delta_i \stackrel{\text{def}}{=} (\mathbf{T}_{i-1} \mathbf{Q}_i \mathbf{T}_{i-1}^{-1})$$

Portanto, teremos os seguintes casos:

B.2.1 Caso A - Juntas de Rotação

Para este tipo de junta,

$$\begin{aligned}
 \Delta_i &= \begin{bmatrix} \mathbf{R}_{i-1} & \vdots & \mathbf{p}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 1 \end{bmatrix} \begin{bmatrix} {}^{i-1}\tilde{\mathbf{a}}_{i-1} & \vdots & 0 \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{i-1}^T & \vdots & -\mathbf{R}_{i-1}^T \mathbf{p}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{R}_{i-1} {}^{i-1}\tilde{\mathbf{a}}_{i-1} \mathbf{R}_{i-1}^T & \vdots & -\mathbf{R}_{i-1} {}^{i-1}\tilde{\mathbf{a}}_{i-1} \mathbf{R}_{i-1}^T \mathbf{p}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \\
 &= \begin{bmatrix} \tilde{\mathbf{a}}_{i-1} & \vdots & \mathbf{p}_{i-1} \wedge \mathbf{a}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix}
 \end{aligned}$$

B.2.2 Caso B - Juntas de Translação

Para este tipo de junta,

$$\begin{aligned}
 \Delta_i &= \begin{bmatrix} \mathbf{R}_{i-1} & \vdots & \mathbf{p}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 1 \end{bmatrix} \begin{bmatrix} 0 & \vdots & {}^{i-1}\mathbf{a}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{i-1}^T & \vdots & -\mathbf{R}_{i-1}^T \mathbf{p}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & \vdots & \mathbf{R}_{i-1} {}^{i-1}\mathbf{a}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & \vdots & \mathbf{a}_{i-1} \\ \dots & \dots & \dots \\ 0 & \vdots & 0 \end{bmatrix}
 \end{aligned}$$

Deste modo, ficam concluídas as demonstrações.

Bibliografia

- [ACT 70] ACTON, F.S. *Numerical methods that work*. New York, Harper and Row, 1970.
- [ALV 88] ALVES, J.B.M. *Controle de robô*. Campinas, Cartgraf, 1988.
- [ANG 88] ANGELES, J.; MA, O. Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement. *International Journal of Robotics Research*, Cambridge, v. 7, n. 5, pp. 32-47, Oct. 1988.
- [ASA 85] ASADA, H.; SLOTINE, J.J.E. *Robot analysis and control*. New York, John Wiley, 1985.
- [BEN 74] BEN-ISRAEL, A.; GREVILLE, T.N.E. *Generalized inverses: theory and applications*. New York, John Wiley, 1974.
- [BLU 72] BLUM, E.K. *Numerical analysis and computation: theory and practice*. New York, John Wiley, 1972.
- [BOI 88] BOISSONAT, J.D.; LAUMOND, J.P. *WORKSHOP ON GEOMETRY AND ROBOTICS*, n. 391, Toulouse, 1988. Proceedings, Berlin, Springer, 1989. (Lecture Notes in Computer Science).
- [BOU 71] BOULLION, T.L.; ODELL, P.L. *Generalized inverse matrices*. New York, Wiley-Interscience, 1971.
- [CAR 69] CARNAHAN, B.; LUTHER, H.A.; WILKES, J.O. *Applied numerical methods*. New York, John Wiley, 1969.
- [CHG 88] CHENG, P.Y.; WENG, C.I.; CHEN, C.K. Symbolic derivation of dynamic equations of motion for robot manipulators using piogram symbolic method. *IEEE Journal of Robotics and Automation*, New York, v. 4, n. 6, pp. 599-609, Dec. 1988.
- [CHE 89] CHENG, H. *Computational aspects of robot manipulation*. Urbana, 1989, 165p. Ph.D. Dissertation - University of Illinois.
- [COI 78] COIMBRA, A.L. *Lições de mecânica do contínuo*. São Paulo, EDUSP, 1978.
- [CRA 86] CRAIG, J.J. *Introduction to robotics: mechanics and control*. Menlo Park, Addison-Wesley, 1986.

- [DEM 73] DEMIDOVICH, B.P.; MARON, I.A. *Computational mathematics*. Moscow, Mir, 1973.
- [FEA 83] FEATHERSTONE, R. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research*, Cambridge, v. 2, n. 1, pp. 13-30, Spring 1983.
- [FER 84] FERREIRA, E. *Contribution a l'identification des paramètres e a la commande dynamique adaptative des robots manipulateurs*. Toulouse, 1984, 211p. Thèse Docteur-Ingénieur - Université Paul Sabatier.
- [FER 91] FERREIRA, E. *Robótica básica: modelagem de robôs*. V EBAI, Rio de Janeiro, 1991.
- [FRA 90] FRANÇA, L.N.F. *Mecânica analítica: notas de aula*. São Paulo, Escola Politécnica da USP, 1990.
- [GIA 78] GIACAGLIA, G.E.O. *Mecânica analítica*. Rio de Janeiro, Almeida Neves, 1978.
- [GOL 89] GOLUB, G.H.; VAN LOAN, C.F. *Matrix computations*. Baltimore, The Johns Hopkins University Press, 1989.
- [GOR 84] GORLA, B.; RENAUD, M. *Modèles des robots manipulateurs: application a leur commande*. Toulouse, Cepadues Editions, 1984.
- [GU 88] GU, Y.L.; LOH N.K. Dynamic modeling and control by utilizing an imaginary robot model. *IEEE Journal of Robotics and Automation*, New York, v. 4, n. 5, pp. 532-540, Oct. 1988.
- [GUO 89] GUO, L.; ANGELES, J. Controller estimation for the adaptive control of robotic manipulators. *IEEE Trans. Robotics and Automation*, New York, v. 5, n. 3, pp. 315-323, June 1989.
- [HEN 62] HENRICI, P. *Discrete variable methods in ordinary differential equations*. New York, John Wiley, 1962.
- [HOL 80] HOLLERBACH, J.M. A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Syst. Man Cybern.*, New York, v. SMC-10, pp. 730-736, Nov. 1980.
- [KAR 89] KARMANOV, G.V. *Mathematical programming*. Moscow, Mir, 1989.
- [KHO 89] KHOSLA, P.K. Categorization of parameters in the dynamic robot model. *IEEE Trans. Robotics and Automation*, New York, v. 5, n. 3, pp. 261-268, June 1989.
- [KOB 89] KOBRINSKI, A.A.; KOBRINSKI, A.E. *Bras manipulateurs des robots: architecture et théorie*. Moscow, Mir, 1989.

- [KOR 75] KORSHUNOV, Y.M. *Fundamentos matemáticos de la cibernética*. Moscow, Mir, 1975.
- [KUL 88] KULESHOV, V.S.; LAKOTA, N.A. *Remotely controlled robots and manipulators*. Moscow, Mir, 1988.
- [LAT 85] LATHROP, R.H. Parallelism in manipulator dynamics. *International Journal of Robotics Research*, Cambridge, v. 4, n. 2, pp. 80-102, Summer 1985.
- [LI 88] LI, C.J. A new method of dynamics for robot manipulators. *IEEE Trans. Syst. Man Cybern.*, New York, v. 18, n. 1, pp. 105-114, Jan./Feb. 1988.
- [LI 89] LI, C.J. An efficient method for linearization of dynamic models of robot manipulators. *IEEE Trans. Robotics and Automation*, New York, v. 5, n. 4, pp. 397-408, Aug. 1989.
- [LUR 68] LUR'E, L. *Mécanique analytique*. Tomes I, II. Paris, Masson e Cie, 1968.
- [MEG 84] MEGAHED, S. *Contribution a la modélisation géométrique et dynamique des robots manipulateurs à structure de chaîne cinématique simple ou complexe. Application a leur commande*. Toulouse, 1984, 203p. Thèse d'Etat - Université Paul Sabatier.
- [NAG 86] NAGY, F.N.; SIEGLER, A. *Engineering foundations of robotics*. Englewood Cliffs, Prentice-Hall International, 1986.
- [NEU 84] NEUMAN, C.P.; MURRAY, J.J. Linearization and sensitivity functions of dynamic robot models. *IEEE Trans. Syst. Man Cybern.*, New York, v. SMC-14, pp. 805-818, Nov./Dec. 1984.
- [NEU 85] NEUMAN, C.P. Discrete dynamic robot models. *IEEE Trans. Robotics and Automation*, New York, v. SMC-15, n. 2, pp. 193-204, Mar./Apr. 1985.
- [PAU 81] PAUL, R. *Robot manipulators: mathematics, programming and control*. Cambridge, MIT Press, 1981.
- [PIE 68] PIEPER, D.L. *The kinematics of manipulators under computer control*. Stanford, 1968, 157p. Ph.D. Dissertation - Stanford University.
- [PRE 89] PRESS, W.H.; FLANNERY, B.P.; TEULKOLSKY, S.A.; VETTERLING, W.T. *Numerical recipes in Pascal: the art of scientific computing*. Cambridge, Cambridge University Press, 1989.
- [REN 80] RENAUD, M. *Contribution a l'étude de la modélisation et de la commande des systèmes mécaniques articulés*. Toulouse, 1975, 184p. Thèse Docteur-Ingénieur - Université Paul Sabatier.
- [SAM 86] SAMARSKI, A.A. *Introducción a los métodos numéricos*. Moscow, Mir, 1986.

- [SPO 89] SPONG, M.W.; VIDYASAGAR, M. *Robot dynamics and control*. New York, John Wiley, 1989.
- [STU 88] STURGES, R.H. Analog matrix inversion. *IEEE Journal of Robotics and Automation*, New York, v. 4, n. 2, pp. 157-162, Apr. 1988.
- [TAR 66] TARG, S. *Éléments de mécanique rationnelle*. Moscow, Mir, 1966.
- [WAL 82] WALKER, M.W.; ORIN, D.E. Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamic Systems, Measurement and Control*, New York, v. 104, pp. 205-211, Sept. 1982.
- [WAL 88] WALKER, M.W. Manipulator kinematics and the epsilon algebra. *IEEE Journal of Robotics and Automation*, New York, v. 4, n. 2, pp. 186-192, Apr. 1988.