

UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE ENGENHARIA MECÂNICA

PROGRAMA DE PÓS-GRADUAÇÃO

**ESTUDO DE UMA ESTRUTURA COM TRÊS GRAUS DE LIBERDADE
DE ORIENTAÇÃO VISANDO SEU USO COMO SIMULADOR DE VÔO**

**DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA
CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
MECÂNICA**

JOEL LACERDA

FLORIANÓPOLIS, JUNHO DE 1997.

**ESTUDO DE UMA ESTRUTURA COM TRÊS GRAUS DE LIBERDADE
DE ORIENTAÇÃO VISANDO SEU USO COMO SIMULADOR DE VÔO**

JOEL LACERDA

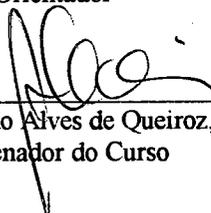
ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE

MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA MECÂNICA, ÁREA DE CONCENTRAÇÃO FABRICAÇÃO,
APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA
MECÂNICA.



Prof. Carlos Alberto Martin, Dr. Ing.
Orientador



Prof. Abelardo Alves de Queiroz, PhD.
Coordenador do Curso

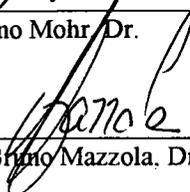
BANCA EXAMINADORA



Prof. Edson da Rosa, Dr. Eng. Mec.



Prof. Hari Bruno Mohr, Dr.



Prof. Vitorio Bruno Mazzola, Dr.

Nada do que ainda não existe pode ser feito, a exceção do que ainda não foi tentado.

Francys Bacon

DEDICATÓRIA

Dedico este trabalho às pessoas de bem - em especial àquelas de poucos bens.

AGRADECIMENTOS

Aos que me antecederam - não tivessem construído o conhecimento, eu nada saberia; não tivessem vivido, e eu sequer existiria

Ao Departamento de Engenharia Mecânica por acolher alunos de tempo parcial em seu Programa de Pós-Graduação - não fosse assim, profissionais como eu não teriam oportunidade de capacitação como a que tive.

Ao professor Carlos Alberto Martin, pela forma amigável e compreensiva da sua orientação.

Aos colegas do Núcleo de Eletrônica da Escola Técnica Federal de Santa Catarina, cujos horários de trabalho ficaram um pouco piores.

À minha família - que me suportou nos momentos de tensão e mau humor.

Ao grupo Companheiros de Jugo, pela amizade e estímulo.

A todos os que de alguma forma ajudaram e aos que não se opuseram

o meu muito obrigado.

RESUMO

No presente trabalho faz-se o estudo prospectivo de uma estrutura com três graus de liberdade de orientação visando seu uso como simulador de vôo. São feitos estudos da cinemática e apresentada a transformação cinemática inversa; da dinâmica e apresentado o modelo dinâmico da estrutura utilizado nas simulações para determinação dos torques necessários para acionar as juntas. Também são apresentadas possíveis estratégias de controle e soluções de engenharia para problemas de implementação.

ABSTRACT

In the present work a study is made on a structure with three degrees of freedom that is pretended to be used as a flight simulator. Studies in kinematics are made and present the inverse kinematics transformation. Studies in dynamics are also made to provide the structure's dynamic model. The model is used in simulations to determine the actuators torques. There have been presented some strategies of control and engineering solutions to the problems in implementation.

SUMÁRIO

Resumo	
Abstract	
Capítulo 1. Introdução	1
Capítulo 2. Cinemática	3
2.1 Introdução	3
2.2 Ângulos de guinada, arfagem e rolamento	4
2.3 Ângulos X,Y,Z de Euler	5
2.4 Posmultiplicação de matrizes de rotação	7
2.5 Notação de Denavit-Hartenberg	8
2.6 Cinemática direta e inversa	10
2.7 Velocidade	11
2.8 Equacionamento cinemático da estrutura do simulador	16
Capítulo 3. Dinâmica	20
3.1 Introdução	20
3.2 Formulação de Newton - Euler	20
3.3 Formulação por Lagrange	22
3.4 Esforços generalizados	26
3.5 Modelagem e simulação da estrutura do simulador	27
Capítulo 4. Controle	
4.1 Introdução	43
4.2 Caracterização da estrutura	43
4.3 Dinâmica dos atuadores	44
4.4 Compensador PD	47
4.5 Controle no espaço de estados	49
4.6 Torque computado	49
4.7 Controle multivariável	52
4.8 Controle por <i>sliding mode</i>	57
4.9 Controle por passividade	59

4.10 Controle por saturação	61
4.11 controladores adaptativos	63
4.12 Controle de estruturas flexíveis	66
4.13 Controle nas singularidades	68
Capítulo 5 Processadores utilizáveis no controle	72
5.1 Introdução.	72
5.2 Requisitos de desempenho do <i>hardware</i>	72
5.3 Considerações sobre o <i>hardware</i> de controle	78
5.4 Processadores utilizáveis no controle de movimento.	79
5.5 Processadores CISC e RISC	89
5.6 Tendências atuais	95
5.7 Conclusões	96
Capítulo 6. Contribuições à construção	
6.1 Acionamento da cabine	98
6.2 Construção da carenagem	99
6.3 Posição do piloto	99
6.4 Passagem de sinais e energia pelas juntas	100
6.5 Balanceamento de massas	101
6.6 Diminuição da troca de fiação entre os segmentos	102
6.7 Acionamento	103
6.8 Uso de giroscópio	103
6.9 Conclusão	104
6.10 Referências bibliográficas	107
Anexo	A-1

CAPÍTULO 1

1. INTRODUÇÃO

A crescente sofisticação, o alto custo e a necessidade de um treinamento cada vez maior de pilotos, apresentados por certos aparelhos, exigiram o desenvolvimento de simuladores. Nestes aparatos, em condições as mais realistas possíveis, o piloto em treinamento pode, a um custo muito menor e com segurança máxima, realizar extensivos treinamentos, mesmo de manobras altamente arriscadas quando realizadas com um veículo real. Nestes equipamentos um *software* que simula as características e reações do veículo real controla simultaneamente os movimentos de uma estrutura mecânica, responsável pelos movimentos que deverão impressionar os sentidos tátil e de equilíbrio, e a apresentação de áudio e vídeo, de forma a criar um ambiente de aparente realidade.

Os simuladores profissionais existentes, dedicados, em especial, ao treinamento para o comando de aviões comerciais, como os grandes jatos, são estruturas grandes e pesadas acionadas por conjuntos de cilindros hidráulicos e não se prestam ao treinamento para pilotagem em manobras rápidas, como aviões de acrobacia ou caça, capazes de girar em torno de qualquer um de seus eixos principais. O barateamento e crescente capacidade dos processadores permite que simuladores de veículos, os mais variados, sejam construídos também para fins de entretenimento.

No presente trabalho estuda-se a estrutura mecânica sugerida para uso como simulador de veículos de manobra rápida de três eixos, evidenciando os problemas de engenharia a serem superados e apontando as soluções possíveis.

A estrutura é composta por quatro segmentos: base, garfo, anel externo e cabine fig. 1.1 Como os eixos de giro são mutuamente interceptantes, a cabine tem todos os movimentos de uma junta esférica, ou seja, tem três graus de liberdade de orientação. Também foi dado como requisito que a estrutura possa executar movimentos ilimitados e reversíveis em qualquer um dos três eixos. O simulador compõem-se, além da estrutura mecânica, objeto deste trabalho, também de sistemas eletrônicos, computacionais e ópticos, necessários a criação do ambiente de realidade virtual e interatividade. Estas últimas características colocam a operação do simulador como um problema a ser resolvido em tempo real, com todas as consequências que tal implica. No capítulo 2 será visto o tratamento da cinemática da estrutura, precedido por uma introdução teórica desenvolvida com base na cinemática do punho esférico dos manipuladores robóticos. A teoria da dinâmica e o modelamento dinâmico da estrutura compõem o capítulo 3 onde também são feitas simulações. O capítulo 4 aborda as várias estratégias de controle empregadas em robôs e que são passíveis de serem utilizadas no controle da estrutura mecâni-

ca do simulador. Nas principais abordagens dá-se um tratamento mais extenso, nas demais remete-se para referências onde o assunto é explorado de forma especializada, uma vez que o objetivo deste capítulo não é o de desenvolver os algoritmos de controle, mas apenas levantar os problemas e estratégias que poderão ser melhor avaliados quando da construção de um protótipo. No capítulo 5 são levantadas plataformas de *hardware* com potencial para utilização no controle da estrutura mecânica do simulador. Algumas sugestões e contribuições a aspectos construtivos e soluções de engenharia são apresentadas no capítulo 6.

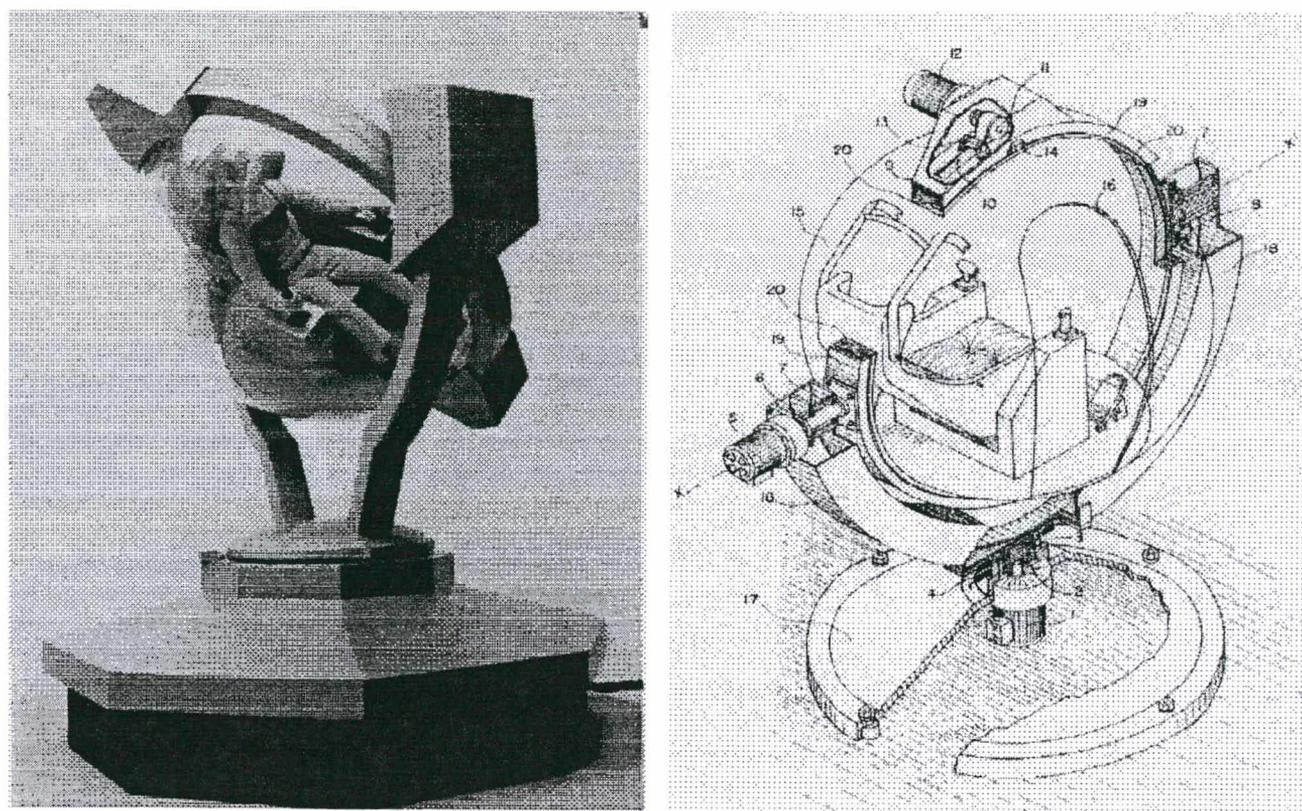


Figura 1.1 Maquete e esquema da estrutura proposta para o simulador.

CAPÍTULO 2. CINEMÁTICA

2.1 INTRODUÇÃO

No estudo da cinemática de um sistema, como estabelecido na Física, tem-se interesse apenas nas variáveis tempo e distância e as derivadas desta última. Quando se trata de um corpo livre para mover-se no espaço tridimensional, aparece ainda, além da noção de posição, ligada aos conceitos de distância e velocidade linear (translação), a noção de orientação, ligada aos conceitos de rotação e velocidade angular. Como as grandezas citadas são de caráter relativo, há sempre a necessidade do estabelecimento de um referencial, e, por estarem envolvidos movimentos de rotação, o referencial já não pode ser um ponto, mas sim um sistema de coordenadas em relação ao qual se possam estabelecer não apenas distâncias (posição), como também ângulos (orientação) de um corpo.

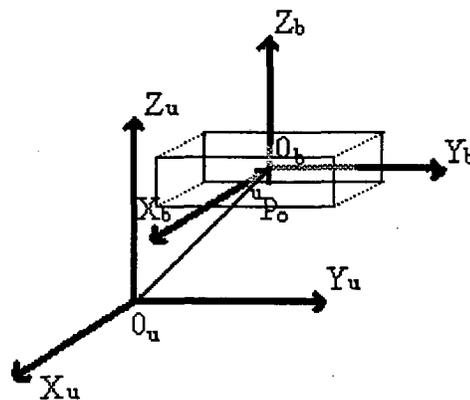


Figura 2.1 Sistemas de coordenadas para descrever posição e orientação de um corpo, b, em relação ao referencial, u.

Como visto na figura 2.1, a completa descrição da situação de um corpo no espaço em relação a um dado referencial é feita pelo vetor, uP_o , que liga a origem do sistema de coordenadas do referencial, O_u , à origem do sistema de coordenadas do corpo, O_b , fixo ao corpo (posição do corpo) mais uma matriz, uR_b , que especifica a rotação do sistema de coordenadas fixo ao corpo em relação a cada um dos eixos do sistema de coordenadas do referencial (orientação do corpo).

A esta matriz chama-se **matriz de rotação**, que quando aplicada a um vetor dado em função do sistema de coordenadas fixo ao corpo, transforma este vetor, de forma que ele passa a ter suas componentes relativas ao sistema de coordenadas do referencial.

$${}^uP = [{}^uR_b] {}^bP \quad (2.1)$$

Há algumas formas diferentes de se construírem as matrizes de rotação, como apresentado por Craig [1], que embora produzam o mesmo resultado, podem ter sua compreensão e aplicação facilitadas dependendo do problema em questão.

2.2 ÂNGULOS DE GUINADA, ARFAGEM E ROLAMENTO

Neste método, a rotação de um sistema de coordenadas qualquer em relação ao sistema de coordenadas do referencial é dada pela especificação dos ângulos que o primeiro deve girar sobre cada um dos eixos do referencial, para que, partindo de uma situação em que esteja paralelo ao sistema de coordenadas do referencial, se chegue à orientação especificada. Os ângulos de giro nos eixos X, Y e Z são chamados respectivamente de **rolamento**, **guinada** e **arfagem** (*roll, yaw, pitch*).

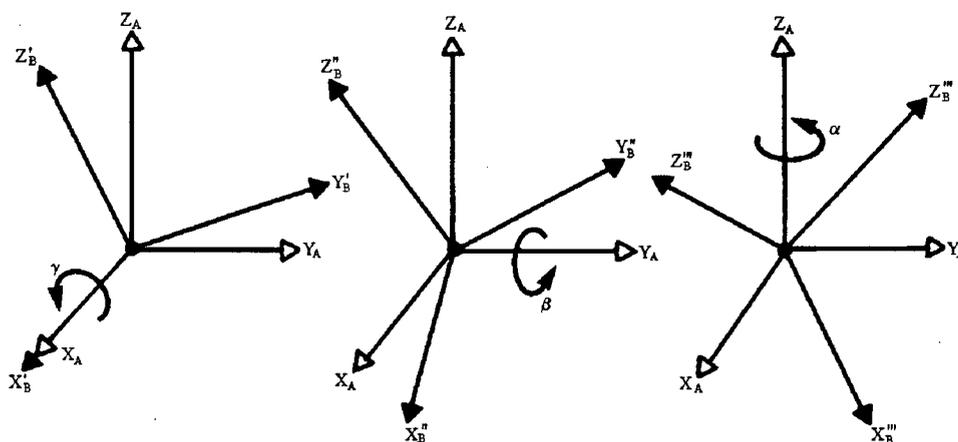


Figura 2.3 Especificação da orientação de um corpo através dos ângulos de rolamento, guinada e arfagem.

A matriz de rotação que leva de um sistema de coordenadas qualquer, B, para um referencial U, ${}^U_B R$, é assim obtida:

Rotação de um ângulo α no eixo Z:

$$= \begin{vmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.2)$$

Rotação de um ângulo β no eixo Y:

$$= \begin{vmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{vmatrix} \quad (2.3)$$

Rotação de um ângulo γ no eixo X:

$$= \begin{vmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{vmatrix} \quad (2.4)$$

A matriz de rotação que leva de B para U é então obtida pela posmultiplicação das três matrizes acima, resultando:

$${}^U_B R = \begin{vmatrix} c\alpha \cdot c\beta & c\alpha \cdot s\beta \cdot s\gamma - s\alpha \cdot c\gamma & c\alpha \cdot s\beta \cdot c\gamma + s\alpha \cdot s\gamma \\ s\alpha \cdot c\beta & s\alpha \cdot s\beta \cdot s\gamma + c\alpha \cdot c\gamma & s\alpha \cdot s\beta \cdot c\gamma - c\alpha \cdot s\gamma \\ -s\beta & c\beta \cdot s\gamma & c\beta \cdot c\gamma \end{vmatrix} \quad (2.5)$$

2.3. ÂNGULOS DE EULER

Neste outro método de construção da matriz de rotação parte-se também com o sistema de coordenadas B coincidente com o referencial U, entretanto, agora as rotações serão efetuadas em relação aos eixos do sistema de coordenadas B. Assim, gira-se primeiro de um ângulo α no eixo Z, em seguida gira-se de um ângulo β no eixo Y que, note-se, já teve sua posição modificada, por último, gira-se de um ângulo γ no eixo X, este também já com sua posição alterada pelas duas rotações anteriores. Um ponto de interesse especial neste tipo de construção da matriz de rotação é o fato de que os giros acontecem do ponto de vista de um observador colocado no corpo e não no referencial inercial.

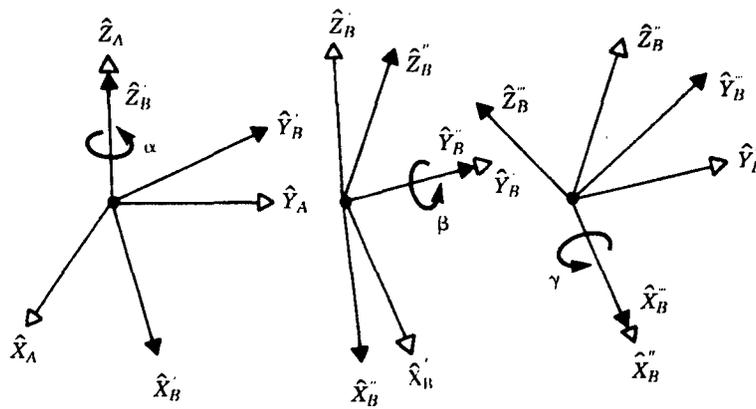


Figura 2.1 Especificação da orientação de um corpo através dos ângulos de Euler.

Obtenção da matriz de rotação pelo métodos dos ângulos de Euler:

Giro de um ângulo α no eixo Z de B:

$$= \begin{vmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.6)$$

Giro de um ângulo β no eixo Y de B:

$$= \begin{vmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{vmatrix} \quad (2.7)$$

Giro de um ângulo γ no eixo X de B:

$$= \begin{vmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{vmatrix} \quad (2.8)$$

A matriz de rotação, $[{}^uR]$, que leva de B para U é então dada pela posmultiplicação das três matrizes acima, resultando:

$${}^U_B R = \begin{vmatrix} c\alpha \cdot c\beta & c\alpha \cdot s\beta \cdot s\gamma - s\alpha \cdot c\gamma & c\alpha \cdot s\beta \cdot c\gamma + s\alpha \cdot s\gamma \\ s\alpha \cdot c\beta & s\alpha \cdot s\beta \cdot s\gamma + c\alpha \cdot c\gamma & s\alpha \cdot s\beta \cdot c\gamma - c\alpha \cdot s\gamma \\ -s\beta & c\beta \cdot s\gamma & c\beta \cdot c\gamma \end{vmatrix} \quad (2.9)$$

As matrizes de rotação podem ser encaradas como transformações que mapeiam vetores de um sistema de coordenadas em um outro sistema de coordenadas, desde que estes tenham uma origem comum. Desta forma, são suficientes para a descrição de movimentos de rotação pura e o mapeamento de vetores livres. Entende-se por vetores livres aqueles que não necessitam do estabelecimento de um ponto de aplicação para estarem completamente descritos, como, por exemplo o vetor velocidade angular.

Já para fazer o mapeamento de vetores que necessitam da indicação de um ponto de aplicação para estarem completamente especificados, como o vetor força, e se tratando de sistemas de coordenadas cujas origens não são coincidentes, há que se levar em conta uma translação, implícita na distância que separa as origens dos dois sistemas. A transformação capaz de mapear vetores genéricos de um sistema de coordenadas qualquer, B, em um sistema de coordenadas, U, é a **matriz homogênea** formada pela matriz de rotação ${}^U_B R$, pelo vetor que liga as origens dos dois sistemas de coordenadas, e por um fator de escala que modifica a magnitude do vetor, neste trabalho sempre unitário [4], como apresentado esquematicamente abaixo:

$${}^U_B T = \begin{vmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.10)$$

2.4. POSMULTIPLICAÇÃO DE MATRIZES DE ROTAÇÃO

Até agora o uso das transformações foi pensado como ferramenta para mapear um sistema de coordenadas qualquer atrelado a um corpo, em um sistema de coordenadas referencial que se imagina inercial, mas o processo pode ser generalizado para fazer o mapeamento entre dois sistemas de coordenadas quaisquer.

Assim, no estudo de estruturas segmentadas formadas por uma cadeia de corpos, pode-se aplicar a transformação para mapear o sistema de coordenadas de um corpo para o de seu vizinho, normalmente do corpo que está mais próximo da extremidade da cadeia para aquele que o antecede,

o que pode ser repetido iterativamente a cada segmento da cadeia, até que se alcance o referencial da base, quando se poderá mapear a posição e a orientação da cadeia em função do referencial inercial, o que é feito pela simples posmultiplicação das transformações entre os vários segmentos. Por exemplo, em uma estrutura formada por três segmentos A, B e C, o mapeamento de C no sistema de coordenadas referencial, U, seria feito por:

$${}^U T_C = {}^U T_A {}^A T_B {}^B T_C \quad (2.11)$$

É importante lembrar que transformações como a do exemplo acima mudam a cada alteração da configuração da cadeia pois são função das posições e orientações relativas entre os segmentos.

2.5. A NOTAÇÃO DE DENAVIT-HARTENBERG PARA ESTRUTURAS SEGMENTADAS

A fim de facilitar seu estudo, as estruturas segmentadas podem ser consideradas como sendo formadas por **segmentos** rígidos, unidos um ao outro por **juntas** com um único grau de liberdade. O modelamento de juntas com mais de um grau de liberdade é conseguido pelo artifício de se considerá-la como sendo formada por várias juntas ligadas por segmentos de comprimento nulo. Para a completa especificação de uma cadeia não basta que se fixe um certo número de juntas e segmentos, pois os segmentos podem ter tamanhos e formas diversas e as juntas podem ter orientações quaisquer. De acordo com a notação de **Denavit-Hartenberg**, uma cadeia pode ser completamente especificada atribuindo-se dois parâmetros a cada segmento e um parâmetro e uma variável a cada junta, conforme a figura 2.4 onde a e α são respectivamente o **comprimento** e a **torção** do segmento e d e θ respectivamente o **desvio** e o **ângulo** da junta. Nas juntas prismáticas (deslizantes) θ é fixo e d é a variável de junta. Nas juntas rotacionais d é fixo como parâmetro e θ é a variável de junta [1], [2].

Com base na notação de Denavit-Hartenberg um mecanismo como o da figura 2.5 pode ser completamente descrito através de uma tabela como a da mesma figura.

Para obter-se posicionamento e orientação em um espaço tridimensional (cartesiano) são necessários seis graus de liberdade, três para o posicionamento e três para a orientação, donde a necessidade de uma cadeia com seis juntas. Isto pressupõe ainda a correta orientação das juntas de

modo a evitar redundância, pois se as juntas forem paralelas, o mecanismo só poderá seguir posições contidas em um plano, embora possua ainda seis graus de liberdade.

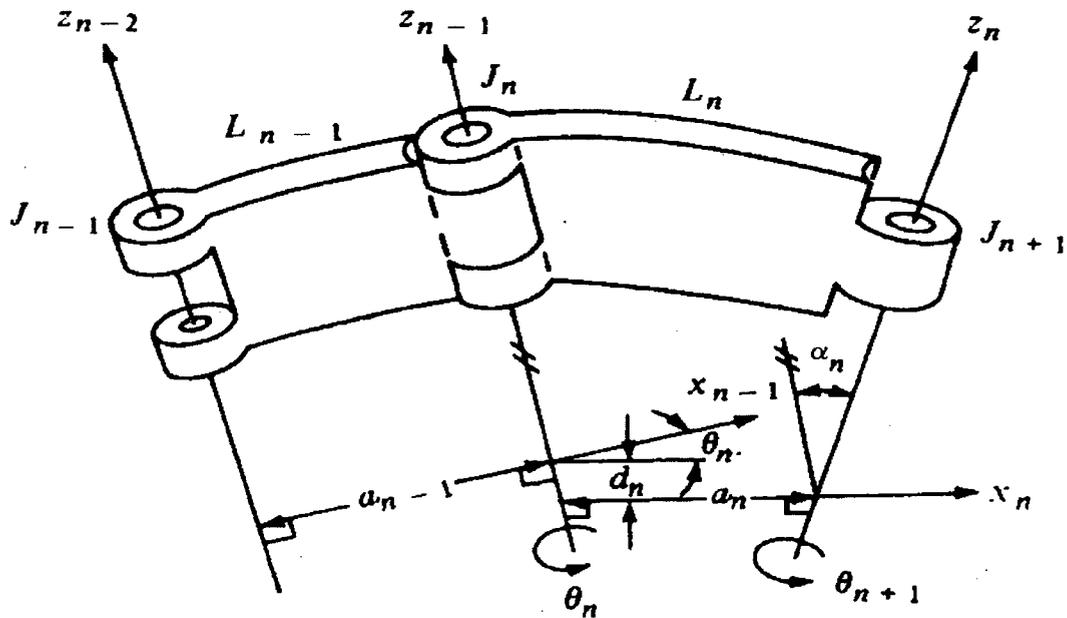
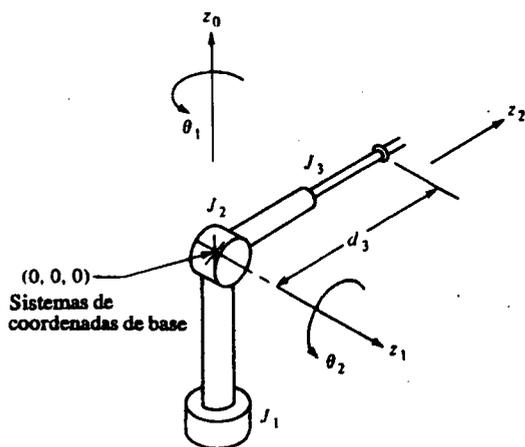


Figura 2.4 Parâmetros de Denavit-Hartenberg para juntas e segmentos



SEGMENTO	α_i	a_i	d_i	θ_i
1	-90°	0	0	θ_1
2	90°	0	0	θ_2
3	0	0	d_3	0

Figura 2.5 Estrutura especificada por Denavit-Hartenberg e respectiva tabela de parâmetros

A uma cadeia que executa posicionamento e orientação em um espaço utilizando para tal, o número mínimo de juntas (graus de liberdade), chama-se **estrutura não redundante** [2].

2.5.1 OBTENÇÃO DA TRANSFORMAÇÃO A PARTIR DA CONVENÇÃO DE DENAVIT-HARTENBERG

A transformação que leva do sistema de coordenadas de uma junta para o sistema de coordenadas da junta que a antecede, ${}^{i-1}T_i$, é dada segundo os parâmetros de Denavit-Hartenberg por [2]:

$${}^{i-1}T_i = \begin{vmatrix} c\theta & -s\theta.c\alpha_i & s\theta.s\alpha_i & a_i.c\theta \\ s\theta & c\theta.c\alpha_i & -c\theta.s\alpha_i & a_i.s\theta \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.12)$$

Assim, para se obter a transformação que leva de uma junta para que a antecede, basta introduzir na matriz (2.12) os parâmetros tirados da tabela de Denavit-Hartenberg para a junta i em questão. Repetindo o processo para cada junta até atingir a base, ter-se-ão n transformações de juntas que posmultiplicadas, darão uma transformação que leva da extremidade da cadeia para a base. Como as únicas variáveis são os ângulos θ e os deslocamentos d que medem a deflexão em cada junta, basta substituir seus valores para que se obtenham em cada uma das três primeiras colunas da matriz da transformação os vetores do sistema de coordenadas da extremidade, dados por suas projeções no sistema de coordenadas da base, e na última coluna o vetor que posiciona a origem do sistema de coordenadas da extremidade em relação ao sistema de coordenadas da base.

2.6. CINEMÁTICA DIRETA E INVERSA

Ao processo de entrar os valores dos ângulos de junta e obter a posição e orientação da extremidade da cadeia chama-se **cinemática direta**. Ao processo inverso, encontrar os ângulos de deflexão de juntas que levam a uma posição e orientação específicas, denomina-se **cinemática inversa**.

Na cinemática direta, uma vez definidos os ângulos de juntas, θ , a posição e a orientação da extremidade da cadeia estão unicamente definidos. Já na cinemática inversa, tanto pode haver multiplicidade como inexistência de soluções. A multiplicidade de soluções ocorrerá no caso de estruturas redundantes, nas quais, mesmo fixadas uma posição e uma orientação para a extremidade da cadeia, haverá sempre um par de juntas que poderão assumir infinitos ângulos - tome-se como exemplo o braço humano. A inexistência de soluções, por sua vez, está ligada a condições de

singularidade. A condição de singularidade pode ocorrer pelas limitações físicas da estrutura, como comprimento dos segmentos e deflexões máximas das juntas, ou pelo alinhamento de juntas cujos eixos sejam interceptantes, situação que provoca a perda de um grau de liberdade da estrutura, impedindo-a de mover-se em uma certa direção enquanto estiver na configuração de singularidade.

O cálculo da cinemática inversa é feito pelo desmembramento da equação matricial ${}^0T = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n$ em um sistema com n equações - ditas equações cinemáticas- e n incógnitas, sempre que for possível encontrar um sistema fechado de equações algébricas [2]. Estruturas para as quais é possível encontrar uma solução algébrica fechada chamam-se **estruturas resolvíveis**. Estruturas com até seis graus de liberdade são resolvíveis, desde que possuam três juntas rotacionais consecutivas cujos eixos se interceptem mutuamente [2].

2.7. VELOCIDADE

Até agora só foi tratado o problema de se estabelecer relações entre os deslocamentos da extremidade da cadeia as deflexões das juntas que os produzem. No entanto, de igual importância é o equacionamento das relações entre as velocidades da extremidade da cadeia e as velocidades de junta correspondentes. Inicialmente deve-se considerar os deslocamentos infinitesimais da extremidade em cada uma das direções ortogonais, como resultado de deflexões infinitesimais em cada uma das juntas que contribuem para aquele deslocamento, ou algebricamente:

$$dx = \frac{\partial x}{\partial \theta_1}(\theta_1, \theta_2)d\theta_1 + \frac{\partial x}{\partial \theta_2}(\theta_1, \theta_2)d\theta_2 \quad (2.13)$$

$$dy = \frac{\partial y}{\partial \theta_1}(\theta_1, \theta_2)d\theta_1 + \frac{\partial y}{\partial \theta_2}(\theta_1, \theta_2)d\theta_2 \quad (2.14)$$

Na forma matricial pode-se escrever:

$$dX = Jd\Theta \quad (2.15)$$

onde J é a matriz de derivadas parciais dos deslocamentos em relação às deflexões de juntas e é denominado **jacobiano** da estrutura.

$$J = \begin{vmatrix} \frac{\delta x}{\delta \theta_1} & \frac{\delta x}{\delta \theta_2} & \frac{\delta x}{\delta \theta_3} \\ \frac{\delta y}{\delta \theta_1} & \frac{\delta y}{\delta \theta_2} & \frac{\delta y}{\delta \theta_3} \\ \frac{\delta z}{\delta \theta_1} & \frac{\delta z}{\delta \theta_2} & \frac{\delta z}{\delta \theta_3} \end{vmatrix} \quad (2.16)$$

Para obter a relação entre as velocidades da extremidade da cadeia e as velocidades angulares das juntas basta que se dividam ambos os lados da equação 2.15 por dt :

$$\frac{dx}{dt} = J \frac{d\Theta}{dt} \quad (2.17)$$

o que equivale a:

$$V = J\dot{\Theta} \quad (2.18)$$

Logo, a relação entre a velocidade da extremidade da cadeia e as velocidades angulares das juntas é dada pelo jacobiano da estrutura, onde cada coluna representa a velocidade impressa à extremidade por cada uma das juntas. A velocidade total da extremidade pode ser obtida pela soma vetorial das contribuições devidas a cada junta. Para completar a generalização falta ainda o tratamento da orientação da extremidade da cadeia, já que até agora foi feito apenas o tratamento para o posicionamento, o que implica no tratamento de velocidades angulares. No equacionamento das relações entre deflexões (rotações) da estrutura foram utilizadas as matrizes de rotação, porém, deflexões finitas são essencialmente diferentes de deflexões infinitesimais, uma vez que a soma de deflexões infinitesimais é comutativa, logo vetorial, enquanto que a soma de deflexões finitas não o é [2]. Há então que se redefinir as matrizes de rotação para aplica-las as deflexões infinitesimais. A matriz de rotação que mapeia rotações infinitesimais no eixo x é:

$$R_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & c(d\theta_x) & -s(d\theta_x) \\ 0 & s(d\theta_x) & c(d\theta_x) \end{vmatrix} \quad (2.19)$$

como $d\theta_x$ é infinitesimal, a matriz acima pode ser aproximada para

$$R_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & -d\theta_x \\ 0 & d\theta_x & 1 \end{vmatrix} \quad (2.20)$$

Matrizes semelhantes podem ser encontradas para os eixos y e z , resultando em uma matriz de rotação em x,y,z dada pela produto das três primeiras, onde as derivadas de ordem superior foram desprezadas:

$$R_{xyz} = \begin{vmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{vmatrix} \quad (2.21)$$

Como as matrizes de rotação dependem apenas de deflexões por ângulos infinitesimais, a ordem das rotações torna-se irrelevante, e a multiplicação pode ser feita comutativamente.

A velocidade total da extremidade da cadeia em suas componentes de translação e rotação - velocidade linear (matriz V) e velocidade angular (matriz W) - pode ser tratada pela aplicação do raciocínio dos movimentos infinitesimais simultaneamente às duas componentes da velocidade, combinando os dois vetores tridimensionais dx e $d\Theta$, dados em relação ao sistema de coordenadas da base, formando um vetor de sexta dimensão, dP :

$$dP = \begin{vmatrix} dX_e \\ d\Theta_e \end{vmatrix} \quad (2.22)$$

onde o índice e denota tratar-se de movimentos da extremidade da cadeia. Dividindo-se ambos os lados por dt obtem-se:

$$\dot{P} = \begin{vmatrix} V_e \\ W_e \end{vmatrix} \quad (2.23)$$

Novamente a relação entre as velocidades de juntas e a velocidade da extremidade da cadeia é dada pelo jacobiano da estrutura que apresenta agora ordem $6 \times n$.

$$\dot{P} = J\dot{Q} \quad (2.24)$$

As três primeiras linhas da matriz jacobiana, $[J_{Li}]$, estão associadas à velocidade linear da extremidade, as últimas três, $[J_{Ai}]$, à velocidade angular, enquanto que cada coluna expressa a contribuição de cada junta para o movimento, esquematicamente:

$$J = \begin{vmatrix} J_{L1} & J_{L2} & \dots & J_{Ln} \\ J_{A1} & J_{A2} & \dots & J_{An} \end{vmatrix} \quad (2.25)$$

A velocidade linear da extremidade pode então ser escrita como:

$$V_e = J_{L1}\dot{q}_1 + \dots + J_{Ln}\dot{q}_n \quad (2.26)$$

Chamando-se b ao vetor que dá a orientação do eixo da junta i , e \dot{d}_i ao módulo da velocidade naquela direção, a contribuição de uma junta **prismática** à velocidade linear da extremidade será:

$$J_{Li}q_i = b_{i-1}\dot{d}_i \quad (2.27)$$

No caso de uma junta **rotacional** girando com velocidade angular w_i , esta velocidade se propaga por todos os segmentos da estrutura, de i a n , e é dada por:

$$W_i = b_{i-1} \dot{\theta}_i \quad (2.28)$$

Mas esta velocidade angular gera também uma componente linear na velocidade da extremidade dada pelo produto vetorial:

$$J_{Li} \dot{q}_i = w_i \times r_{i-1,e} = (b_{i-1} \times r_{i-1,e}) \dot{\theta}_i \quad (2.29)$$

onde $r_{i-1,e}$ é o vetor que posiciona a extremidade da cadeia em relação ao sistema de coordenadas da junta $i-1$ fig 2.6. Da mesma forma, a velocidade angular da extremidade, W_e , pode ser obtida pela combinação linear das contribuições de cada junta:

$$W_e = J_{A1} \dot{q}_1 + \dots + J_{An} \dot{q}_n \quad (2.30)$$

Quando as juntas são prismáticas não contribuem para a velocidade angular da extremidade, logo:

$$J_{Ai} \dot{q}_i = 0 \quad (2.31)$$

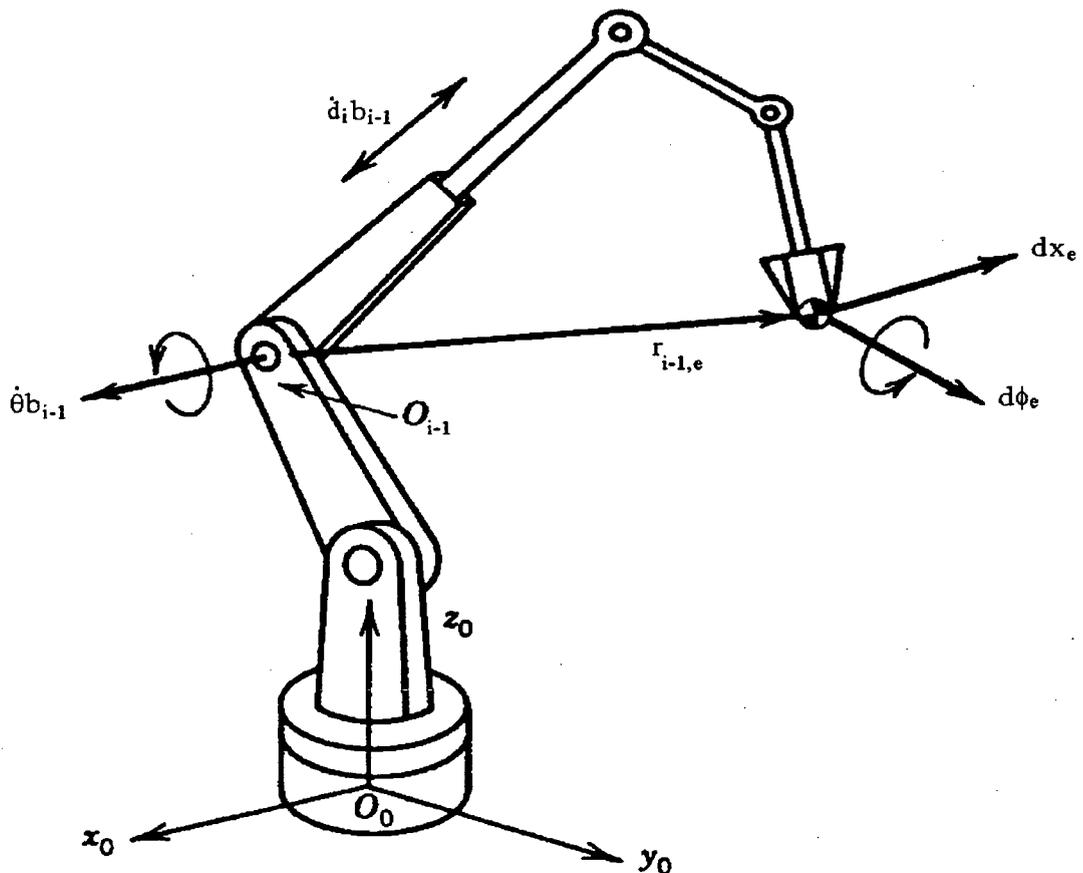


Figura 2.6 Posição da extremidade em relação ao sistema de coordenadas de uma junta.

De forma sumarizada:

$$\begin{vmatrix} J_{Li} \\ J_{Ai} \end{vmatrix} = \begin{vmatrix} b_{i-1} \\ 0 \end{vmatrix} \quad \text{para juntas prismáticas} \quad (2.32)$$

$$\begin{vmatrix} J_{Li} \\ J_{Ai} \end{vmatrix} = \begin{vmatrix} b_{i-1} X r_{i-1,e} \\ b_{i-1} \end{vmatrix} \quad \text{para juntas rotacionais} \quad (2.33)$$

onde os vetores b_{i-1} e $r_{i-1,e}$ podem ser calculados usando-se as matrizes de rotação, R , e transformações homogêneas, T , já vistas. Assim, sendo b o vetor da direção do eixo da junta $i-1$, com $b = |0,0,1|$ (porque o sistema de coordenadas da junta é fixado com o eixo Z coincidente com o eixo da junta), b_{i-1} mapeado no sistema de coordenadas da base será:

$$b_{i-1} = {}^0_1R \dots {}^{i-2}_{i-1}R b \quad (2.34)$$

O cálculo de $r_{i-1,e}$ é feito pela diferença $r_{i-1,e} = p_{0,e} - p_{0,i-1}$ à qual se aplicam as transformações homogêneas 4×4 :

$$\bar{r}_{i-1,e} = {}^0_1T \dots {}^{n-1}_n T \bar{p} - {}^0_1T \dots {}^{i-2}_{i-1} T \bar{p} \quad (2.35)$$

onde $\bar{r}_{i-1,e}$ é o vetor $r_{i-1,e}$ aumentado e \bar{p} é o vetor aumentado da origem do sistema de coordenadas da junta $i-1$, $[0,0,0,1]^T$.

Já se tem agora como calcular o jacobiano de uma cadeia a partir de seus parâmetros físicos, lembrando que, por depender das matrizes de rotação e das transformações homogêneas, que são funções dos ângulos de junta, o jacobiano de uma estrutura é dependente da configuração que ela assume, variando o valor dos elementos permanentemente com o seu movimento.

2.7.1. CINEMÁTICA INSTANTÂNEA INVERSA

Encontrar o vetor de velocidades de juntas que gerem uma determinada velocidade da extremidade é resolver a equação

$$\dot{Q} = J^{-1}\dot{P} \quad , \quad (2.36)$$

o que requer o cálculo da inversa de J. Porém, como J é dependente da configuração instantânea da cadeia, pode ocorrer de não existir a inversa, pois, nos pontos de singularidade o determinante de J se anula.

A solução deste problema pode ser obtida pelo aumento do número de graus de liberdade da estrutura, o que por outro lado dificulta a determinação devido à multiplicidade de soluções que requer o uso de funções de otimização das soluções [2]. Um fator complicador do problema das singularidades é a necessidade de se computar a matriz jacobiana em tempo real, em aplicações onde as tarefas não podem ser pré definidas. Maiores detalhes sobre a solução do problema das singularidades serão dados no capítulo 4.

2.8. EQUACIONAMENTO CINEMÁTICO DA ESTRUTURA DO SIMULADOR

2.8.1 Fixação dos sistemas de coordenadas aos segmentos da estrutura

A fixação dos sistemas de coordenadas aos segmentos da estrutura foi feito segundo a figura 2.7.

2.8.2 Geração da tabela com os parâmetros de Denavit-Hartenberg

i	α_i	a_i	d_i	θ_i
1	-90	0	0	θ_1
2	90	0	0	θ_2
3	0	0	0	θ_3

2.8.3 Inserção, para cada uma das juntas, dos parâmetros da tabela na matriz de transformação genérica de junta.

$${}^{i-1}T_i = \begin{vmatrix} c\theta & -s\theta.c\alpha_i & s\theta.s\alpha_i & a_i.c\theta \\ s\theta & c\theta.c\alpha_i & -c\theta.s\alpha_i & a_i.s\theta \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.37)$$

$${}^0_1T = \begin{vmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad {}^1_2T = {}^1_2R = \begin{vmatrix} c\theta_2 & 0 & s\theta_2 \\ s\theta_2 & 0 & -c\theta_2 \\ 0 & 1 & 0 \end{vmatrix} \quad {}^2_3T = {}^2_3R = \begin{vmatrix} c\theta_3 & -s\theta_3 & 0 \\ s\theta_3 & c\theta_3 & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (2.38)$$

Como não há movimento de translação, as transformações são as próprias matrizes de rotação

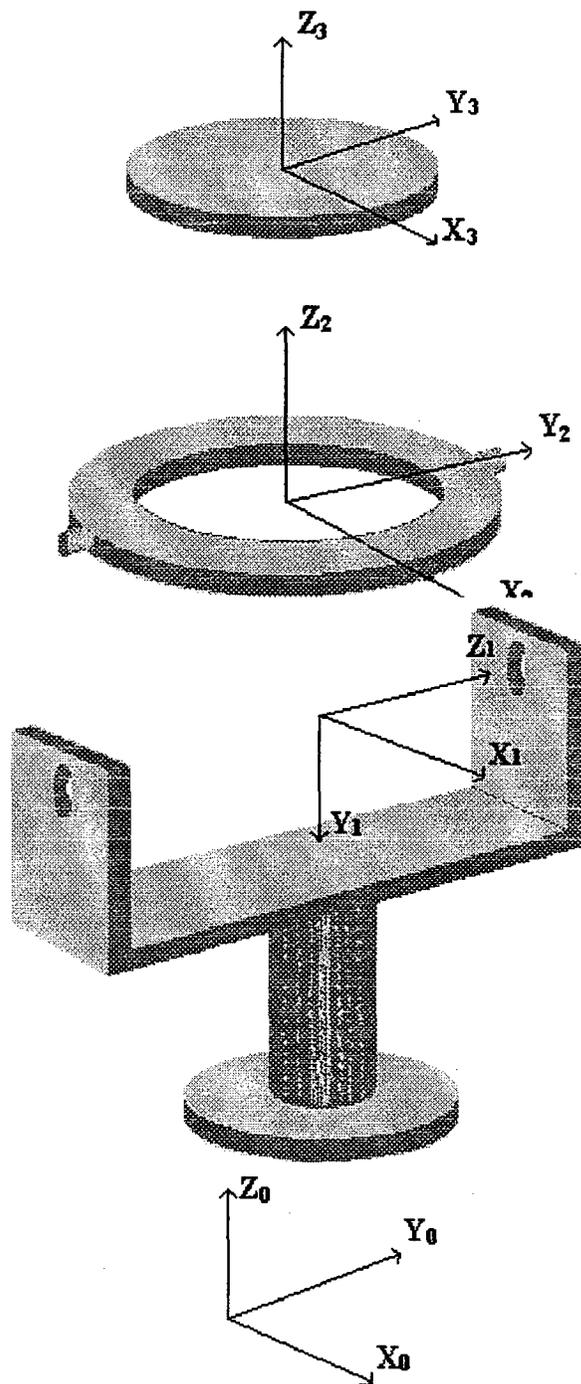


Figura 2.7 Fixação dos sistemas de coordenadas aos segmentos da estrutura.

2.8.4. Posmultiplicação das matrizes de rotação de cada uma das juntas para obtenção da matriz que leva da extremidade da cadeia para a base

$${}^0T_3 = {}^0R_1 {}^1R_2 {}^2R_3 \quad (2.39)$$

2.8.5. Obtenção dos ângulos de rolamento, arfagem e guinada a serem utilizados no sincronismo com a parte gráfica.

$$\begin{aligned} \gamma &= A \tan(r_{32}, r_{33}) \\ \beta &= A \tan(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \end{aligned} \quad (2.40)$$

$$\alpha = A \tan(r_{21}, r_{11})$$

Onde mantém-se os dois argumentos da função arco-tangente para que se possa fazer identificação do quadrante. Os argumentos são os elementos da matriz de rotação 0R_3 e os índices indicam a posição do elemento na matriz.

2.8.6 Inversão (transposição) das matrizes de rotação de cada uma das juntas para obtenção do jacobiano da estrutura relativo ao sistema de coordenadas da extremidade da estrutura (cabine).

$${}^3R_2 = \begin{vmatrix} c\theta_3 & s\theta_3 & 0 \\ -s\theta_3 & c\theta_3 & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad {}^2R_1 = \begin{vmatrix} c\theta_2 & s\theta_2 & 0 \\ 0 & 0 & 1 \\ s\theta_2 & -c\theta_2 & 0 \end{vmatrix} \quad {}^0R_1 = \begin{vmatrix} c\theta_1 & s\theta_1 & 0 \\ 0 & 0 & -1 \\ -s\theta_1 & c\theta_1 & 0 \end{vmatrix} \quad (2.41)$$

$$b_2 = \begin{vmatrix} {}^3R_2 \\ 0 \\ 1 \end{vmatrix} \quad b_1 = \begin{vmatrix} {}^2R_1 \\ 0 \\ 1 \end{vmatrix} \quad b_0 = \begin{vmatrix} {}^0R_1 \\ 0 \\ 1 \end{vmatrix} \quad (2.42)$$

onde b_2, b_1, b_0 , são os vetores das direções dos eixos das juntas dados no sistema de coordenadas do segmento 3 (cabine). O jacobiano da estrutura dado no sistema de coordenadas da cabine, 3J , é a matriz 3 x 3 formada pelos vetores coluna b_2, b_1, b_0 .

$${}^3J = \begin{vmatrix} -s\theta_2 \cdot c\theta_3 & s\theta_3 & 0 \\ s\theta_2 \cdot s\theta_3 & c\theta_3 & 0 \\ c\theta_2 & 0 & 1 \end{vmatrix} \quad (2.43)$$

2.8.7 Inversão do jacobiano para uso, em tempo real, na obtenção das acelerações angulares de cada um dos eixos da estrutura a partir de acelerações dadas segundo o referencial da cabine.

$$\ddot{\Theta} = {}^3J^{-1} \dot{W} \quad (2.44)$$

2.9. Conclusão

Neste capítulo foi feito o estudo genérico da cinemática das estruturas segmentadas estabelecendo as transformações que levam as posições, orientações e suas derivadas no tempo, do espaço de trabalho para o espaço de juntas. Ao final, foram estabelecidas as transformações para o caso específico da estrutura do simulador.

CAPÍTULO 3. DINÂMICA

3.1. INTRODUÇÃO

Sob o enfoque da dinâmica trata-se a relação entre entes da cinemática, como velocidades e acelerações, e as forças e torques necessários para que estas velocidades e acelerações sejam impressas à estrutura móvel.

O modelamento da dinâmica de uma estrutura mecânica possibilita o projeto de controladores ao estabelecer características como massas, inércias, ação da gravidade e coeficientes de atrito. O modelamento dinâmico serve também para a simulação do funcionamento da estrutura em computador, o que permite antever o comportamento dinâmico da estrutura, antes da sua implementação real, possibilitando, entre outras coisas, a determinação dos torques e potências requeridos para o acionamento da estrutura em situações variadas, inclusive as extremas. Adiante, neste capítulo, serão apresentados resultados da simulação da dinâmica da estrutura em estudo utilizando o modelo elaborado por Silva [5].

Há duas abordagens para o equacionamento da dinâmica de uma estrutura, a de Newton-Euler e a de Lagrange.

Na formulação de Newton-Euler as equações baseiam-se essencialmente na segunda lei de Newton adaptada por Euler para aplicação aos movimentos de rotação e as equações diferenciais são obtidas a partir de equações de balanço de forças e momentos calculados para cada segmento da estrutura. Tais balanços incluem tanto as forças e momentos motrizes como aqueles que apenas suportam parte dos esforços de sustentação da estrutura.

Na formulação lagrangeana o equacionamento é feito com base no balanço de energia usando coordenadas generalizadas. Dessa forma, todos os esforços que não realizam trabalho são automaticamente eliminados do balanço, levando a equações que relacionam diretamente o esforço em cada junta com a sua deflexão, isto é, forças e momentos segundo os graus de liberdade da estrutura.

3.2 A FORMULAÇÃO DE NEWTON-EULER

Quando, no estudo do movimento de um corpo, há interesse não só em sua posição (translação), mas também em sua orientação (rotação), além do movimento de translação do centro de

massa em relação a um referencial, tem-se também de considerar os movimentos de rotação do corpo em relação ao seu próprio centro de massa.

O início de uma modelação segundo N-E é o estabelecimento de um diagrama de corpo livre do segmento para o qual se deverá fazer os balanços de forças e momentos:

$$f_{-1,i} - f_{i,i+1} + m_i g - m_i \dot{V}_{ci} = 0 \quad \text{para } i = 1, \dots, n \quad (3.1)$$

onde as duas primeiras parcelas correspondem às forças de interação com os segmentos anterior e posterior, respectivamente, $m_i g$ corresponde à parcela gravitacional e a última parcela está ligada aos esforços inerciais, sendo V_{ci} a velocidade do centro de massa do segmento.

O mesmo tipo de equacionamento deve ser feito para os movimentos rotacionais na forma de um balanço de momentos, com a diferença de que agora a massa do segmento não pode mais ser descrita como puntualmente concentrada em um centro de massa, havendo a necessidade da descrição da sua distribuição espacial, o que é feito pelo **tensor de inércia, I**.

O tensor de inércia é uma matriz 3 X 3 cuja diagonal principal é formada pelos momentos de inércia do corpo em relação a cada um dos eixos do sistema de coordenadas da base e os demais elementos são produtos de inércia entre os eixos. Como se pode deduzir, o tensor de inércia de um corpo é dependente da sua orientação, variando com ela.

Aplicação do balanço de momentos a um corpo:

$$N_{i-1,i} - N_{i,i+1} + r_{i,ci} X f_{i,i+1} - r_{i-1,ci} X f_{i-1,i} - I_i \dot{\omega}_i - \omega_i X (I_i \omega_i) = 0 \quad \text{para } i = 1, \dots, n \quad (3.2)$$

onde:

$N_{i-1,i}$ e $N_{i,i+1}$ são respectivamente os torques transmitidos pelos segmentos anterior e posterior

$r_{i,ci} X f_{i,i+1}$ e $r_{i-1,ci} X f_{i-1,i}$ são os torques devidos às forças transmitidas pelos segmentos posterior e anterior cujas direções não passam pelo centro do segmento em questão.

$I_i \dot{\omega}_i$ é a componente inercial equivalente ao produto $m.a$ dos movimentos lineares

$\omega_i X(I_i \omega_i)$ é a parcela de torques devidos a efeitos não lineares de componentes centrífugas e componentes giroscópicas de Coriolis.

Estas duas equações (3.1 e 3.2) na forma em que estão apresentadas tem pouca utilidade [2] para a análise dinâmica e o projeto de controladores, uma vez que não explicitam as relações entre os esforços nas juntas como uma função das coordenadas generalizadas, q , da estrutura. Para se adequarem ao fim que se tem em mente, haveria necessidade de um tratamento matemático adicional baseado em análises físicas de difícil algoritmização.

3.3. A FORMULAÇÃO POR LAGRANGE

O equacionamento por Lagrange é feito a partir da função de Lagrange, L , que representa a diferença a cada instante, entre as quantidades de energia cinética, T , e potencial, U , armazenadas na estrutura.

$$L(q_i, \dot{q}_i) = T - U \quad (3.3)$$

Na equação 3.3 já se pode perceber a vantagem do uso do equacionamento por Lagrange uma vez que L é função das coordenadas generalizadas q . Por Lagrange, as equações de movimento de um sistema dinâmico ficam:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \quad i = 1, \dots, n \quad (3.4)$$

onde Q_i é o vetor de esforços generalizados atuando nas juntas. Tem-se assim, um sistema fechado que relaciona diretamente as coordenadas generalizadas aos esforços atuantes nas juntas de uma estrutura em cadeia.

3.3.1 ENERGIA CINÉTICA ARMAZENADA NA ESTRUTURA

A energia cinética armazenada em um único segmento da estrutura é:

$$T_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} \omega_i^T I_i \omega_i \quad (3.5)$$

onde:

v_{ci} e ω_{ci} são dois vetores 3×1 que representam respectivamente a velocidade linear do centro de massa e a velocidade angular do segmento em relação ao seu centróide dados no sistema de coordenadas da base.

m_i é a massa do segmento

I_i é o tensor de inércia do segmento no centro de massa expresso no sistema de coordenadas da base.

O primeiro termo do lado direito da equação 3.5 representa a energia cinética do segmento devida ao movimento de translação, enquanto que a segunda parcela corresponde à energia devida ao movimento de rotação do segmento em relação ao seu próprio centróide.

Como a energia pode ser somada de forma escalar, a energia cinética total armazenada na estrutura é:

$$T = \sum_{i=1}^n T_i \quad (3.6)$$

Como as velocidades linear, v_{ci} e angular, ω_i são dependentes das velocidades dos segmentos antecedentes na cadeia, v_{ci} e ω_i são relacionados às coordenadas generalizadas, q , através do jacobiano da estrutura com as colunas de índice maior ou igual a i zeradas, já que as juntas posteriores não contribuem para o movimento do segmento i .

Assim:

$$T = \frac{1}{2} \sum_{i=1}^n (m_i \dot{q}^T J_L^{(i)T} J_L^{(i)} \dot{q} + \dot{q}^T J_A^{(i)T} I_i J_A^{(i)}) \quad (3.7)$$

onde $J_L^{(i)}$ e $J_A^{(i)}$ são respectivamente as linhas da matriz jacobiana que determinam as velocidades linear e angular de cada um dos segmentos da estrutura. As propriedades da massa de toda a estrutura podem ser sumarizadas em uma matriz, H , $n \times n$, chamada **tensor de inércia da cadeia**:

$$H = \sum_{i=1}^n (m_i J_L^{(i)T} J_L^{(i)} + J_A^{(i)T} I_i J_A^{(i)}) \quad (3.8)$$

A equação 3.7 pode então ser reescrita de forma compacta como:

$$T = \frac{1}{2} \dot{q}^T H \dot{q} \quad (3.9)$$

ou na forma escalar:

$$T = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{i,j} \dot{q}_i \dot{q}_j \quad (3.10)$$

onde $H_{i,j}$ corresponde a cada um dos elementos da matriz H , sendo, portanto, também função de q .

3.3.2 ENERGIA POTENCIAL ARMAZENADA NA ESTRUTURA

A energia potencial armazenada na estrutura segmentada é dada pela equação 3.11 que representa a soma das energias potenciais de cada um dos segmentos.

$$U = \sum_{i=1}^n m_i g^T r_{o,ci} \quad (3.11)$$

onde:

m_i é a massa do segmento

$r_{o,ci}$ é a posição do centróide do segmento dada no sistema de coordenadas da base

g é o vetor 3×1 que representa a aceleração da gravidade no sistema de coordenadas da base.

U é dependente de q , uma vez que ci é dependente da configuração assumida pela estrutura.

3.3.3 EQUAÇÃO DE MOVIMENTO

Colocando os resultados anteriormente obtidos para T e U equação (3.4) tem-se: em relação à primeira parcela:

$$\frac{d}{dt} \left(\frac{\delta T}{\delta \dot{q}_i} \right) = \frac{d}{dt} \left(\sum_{j=1}^n H_{ij} \dot{q}_j \right) = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \frac{dH_{ij}}{dt} \dot{q}_j \quad (3.12)$$

a derivada em relação ao tempo de H_{ij} faz aparecer um somatório sobre um novo índice:

$$\frac{dH_{ij}}{dt} = \sum_{k=1}^n \frac{\delta H_{ij}}{\delta q_k} \frac{dq_k}{dt} = \sum_{k=1}^n \frac{\delta H_{ij}}{\delta q_k} \dot{q}_k \quad (3.13)$$

A segunda parcela fica:

$$\frac{\delta T}{\delta q_i} = \frac{\delta}{\delta q_i} \left(\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n H_{jk} \dot{q}_j \dot{q}_k \right) = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\delta H_{jk}}{\delta q_i} \dot{q}_j \dot{q}_k \quad (3.14)$$

Fazendo-se:

$$A = \frac{\delta H_{ij}}{\delta q_k} \quad \text{e} \quad B = \frac{\delta H_{jk}}{\delta q_i} \quad (3.15)$$

pode-se escrever reagrupando os somatórios:

$$\sum_{j=1}^n H_{i,j} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i = Q_i \quad (3.16)$$

onde: $h_{ijk} = A - B$

G_i é o termo gravitacional obtido pela derivada parcial da energia potencial em relação a q_i .

$$G_i = \frac{\delta U}{\delta q_i} = \sum_{j=1}^n m_j g^T \frac{\delta r_{0,ci}}{\delta q_i} = \sum_{j=1}^n m_j g^T J_{Li}^{(j)} \quad (3.17)$$

O primeiro termo da equação (3.16) de movimento corresponde aos esforços de origem inercial, o segundo termo é devido às componentes de torque por efeito centrífugo e de Coriolis e G_i está relacionado aos esforços para vencer a gravidade.

Uma vez que H_{ij} , h_{ijk} e G_i são todos dependentes de q , a equação acima pode ser reescrita como uma equação de estados da dinâmica da estrutura:

$$H(q) \ddot{q} + h(q, \dot{q}) \dot{q} + G(q) = Q \quad (3.18)$$

3.4 ESFORÇOS GENERALIZADOS

Os esforços generalizados, Q , incluem todos os esforços que agem na estrutura, excetuando-se os esforços devidos à inércia e à gravidade, estes já computados como responsáveis pelas quantidades de energia cinética e potencial da estrutura, T e U . Assim, em uma estrutura em cadeia, os esforços generalizados são compostos pelos torques nas juntas exercidos pelos atuadores, $\tau = [\tau_1, \dots, \tau_n]$ e pelas forças externas agindo sobre a extremidade da cadeia.

Para se chegar ao cálculo destes esforços usando as equações de Lagrange, tem-se de trabalhar com o trabalho virtual por eles realizado. Como demonstrado por [2], os esforços externos a que está submetida a extremidade da cadeia se refletem nos esforços nas juntas pela relação:

$$\tau = J^T F_{ext} \quad (3.19)$$

logo, o trabalho virtual realizado pelos esforços generalizados será dado por:

$$\delta W = (\tau^T + J^T F_{ext}) \delta q \quad (3.20)$$

como

$$\delta W = Q^T \delta q \quad (3.21)$$

conclui-se que

$$\tau = Q - J^T F_{ext} \quad (3.22)$$

Se a estrutura não operar contra uma força externa resistente, os esforços generalizados reduzem-se aos torques exercidos pelos atuadores das juntas.

Na equação de movimento obtida falta ainda inserir os esforços necessários para vencer o atrito. Os parâmetros de atrito, devido ao seu comportamento não linear e múltipla dependência, são difíceis de serem estabelecidos com exatidão. Um procedimento que pode ser adotado consiste em estabelecer valores nominais para os coeficientes de atrito e fazer a sua validação ensaiando a planta após sua construção.

3.5 MODELAGEM E SIMULAÇÃO DA ESTRUTURA DO SIMULADOR

A estrutura proposta para o simulador é formada basicamente por três segmentos móveis e uma base, como pode ser visto esquematicamente na figura 2.7. A base é formada por seções radiais em chapa de aço dobrado e parafusado e suporta o mancal de apoio do eixo 1 e o motor que aciona este eixo. O garfo é construído em chapa de aço dobrada e soldada em suas extremidades estão colocados os mancais e o motor que aciona o eixo, ao qual está solidário o anel externo. O anel externo é uma peça usinada em alumínio tendo na face interna os roletes que apoiam o anel interno. O anel externo contém ainda o motor que através de correia dentada aciona o anel interno. O terceiro segmento é também um anel usinado em alumínio tendo na face externa pistas para apoiar nos roletes e superfície dentada para encaixe da correia e na face interna a cabine (figura 1.1).

O modelo dinâmico da estrutura do simulador foi elaborado por [4] usando a formulação de Lagrange, devido as vantagens já citadas, e a documentação encontra-se em anexo. O modelamento foi feito com base na estrutura tal como proposta inicialmente, ou seja, com um motor em cada segmento e acionamento do anel interno por roletes de fricção.

Como a estrutura não age contra esforços resistentes externos, os esforços generalizados são os próprios torques motores das juntas, m_1, m_2, m_3 .

$$m_1 = mi_1 + mc_1 + ma_1$$

$$m_2 = mi_2 + mc_2 + ma_2 + mg_2 \quad (3.23)$$

$$m_3 = mi_3 + mc_3 + ma_3 + mg_3$$

onde:

mi representa as parcelas devidas aos esforços inerciais e correspondem em cada eixo a:

$$mi_1 = H_{11} \cdot \ddot{q}_1 + H_{12} \cdot \ddot{q}_2 + H_{13} \cdot \ddot{q}_3$$

$$mi_2 = H_{12} \cdot \ddot{q}_1 + H_{22} \cdot \ddot{q}_2 + H_{23} \cdot \ddot{q}_3 \quad (3.34)$$

$$mi_3 = H_{13} \cdot \ddot{q}_1 + H_{23} \cdot \ddot{q}_2 + H_{33} \cdot \ddot{q}_3$$

mc representa as parcelas devidas aos esforços centrífugos e de Coriolis:

$$mc_1 = h_{112} \cdot \dot{q}_1 \cdot \dot{q}_2 + h_{113} \cdot \dot{q}_1 \cdot \dot{q}_3 + h_{122} \cdot \dot{q}_2 \cdot \dot{q}_2 + h_{123} \cdot \dot{q}_2 \cdot \dot{q}_3 + h_{132} \cdot \dot{q}_3 \cdot \dot{q}_2$$

$$mc_2 = h_{211} \cdot \dot{q}_1 \cdot \dot{q}_1 + 2 \cdot h_{212} \cdot \dot{q}_1 \cdot \dot{q}_2 + 2 \cdot h_{213} \cdot \dot{q}_1 \cdot \dot{q}_3 + h_{223} \cdot \dot{q}_2 \cdot \dot{q}_3 \quad (3.25)$$

$$mc_3 = h_{311} \cdot \dot{q}_1 \cdot \dot{q}_1 + h_{312} \cdot \dot{q}_1 \cdot \dot{q}_2 + h_{321} \cdot \dot{q}_2 \cdot \dot{q}_1 + h_{322} \cdot \dot{q}_2 \cdot \dot{q}_2$$

ma representa as parcelas de torque necessárias para vencer o atrito:

$$ma_1 = \text{pesototal} \cdot \text{raio} \cdot \text{fatordeatrito}$$

com: pesototal = 4890 N (garfo 1490N, anel externo 1050N e anel interno 2350)

raio = 0,045 m (raio do mancal que apoia o eixo do garfo - eixo 1)

fa = 0,0015 (fator de atrito típico para o mancal utilizado)

$$ma_2 = 4 \cdot f \cdot rb \cdot fa$$

com: f = peso dos anéis + reações de momentos

rb = raio do rolamento = 0,03 m

fa = 0,0015

$$ma_3 = fa \cdot f$$

com: fa = 0,0015

f = resultante do peso do anel interno + momentos de reação

mg representa as parcelas de torque necessárias para vencer a gravidade:

$$mg_1 = 0$$

$$mg_2 = g_2 \cdot rg_2 + g_3 \cdot rg_3$$

com: g_2 = peso do anel externo = 1050 N

g_3 = peso do anel interno com carga = 2350 N

rg_2 e rg_3 = respectivamente as excentricidades dos pesos dos anéis externo e interno vistos da direção do eixo 2

$$mg_3 = g_3 \cdot rg_3$$

com: g_3 = peso anel interno com carga = 2350 N

rg_3 = excentricidade do anel interno

modelo acima foi implementado no programa de simulação SIMNOM tendo como entradas as acelerações e como saídas os torques por eixo da estrutura. Em todas as plotagens o eixo das abcissas é o tempo dado em segundos. Em alguns casos foi plotado, junto com a variável de interesse, uma função seno, deslocada no eixo vertical e sem escala, do ângulo de deflexão da junta, para que se possa ter idéia imediata da sua posição. O passo de tempo utilizado foi escolhido para otimizar a relação precisão / tempo de simulação. Além das corridas plotadas foram feitas outras com tempo expandido para verificação das diferentes configurações da estrutura. Em alguns casos foram utilizadas velocidades fracionárias para evitar efeito harmônico, obtendo-se uma varredura das poses da estrutura em menor tempo.

As seções de simulação tem como objetivos:

- determinar torques e potências devidas ao atrito
- determinar torques e potências devidos às componentes de produto de velocidades
- determinar torques e potências máximos para especificação do acionamento

3.5.1 DETERMINAÇÃO DOS TORQUES DEVIDOS AO ATRITO

Para a determinação da componente de atrito a ser vencida pelo torque motor dos acionamentos simulou-se o acionamento de cada um dos eixos da estrutura com as excentricidades de massa zeradas e com velocidade angular constante, mantendo os demais eixos parados. As figuras 3.1, 3.2 e 3.3 mostram respectivamente os torques de atrito nos eixos 1, 2 e 3 sob velocidade máxima.

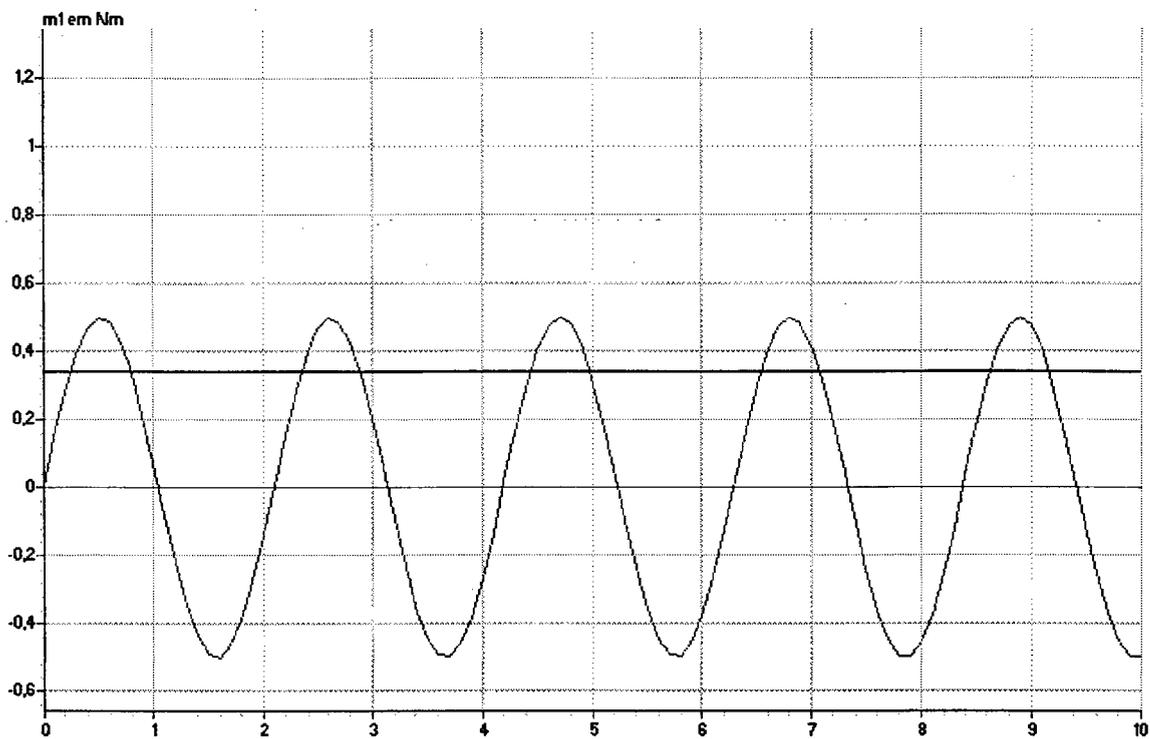


Figura 3.1 Torque de atrito no eixo 1 (a linha tracejada dá idéia da posição do segmento)

t em seg

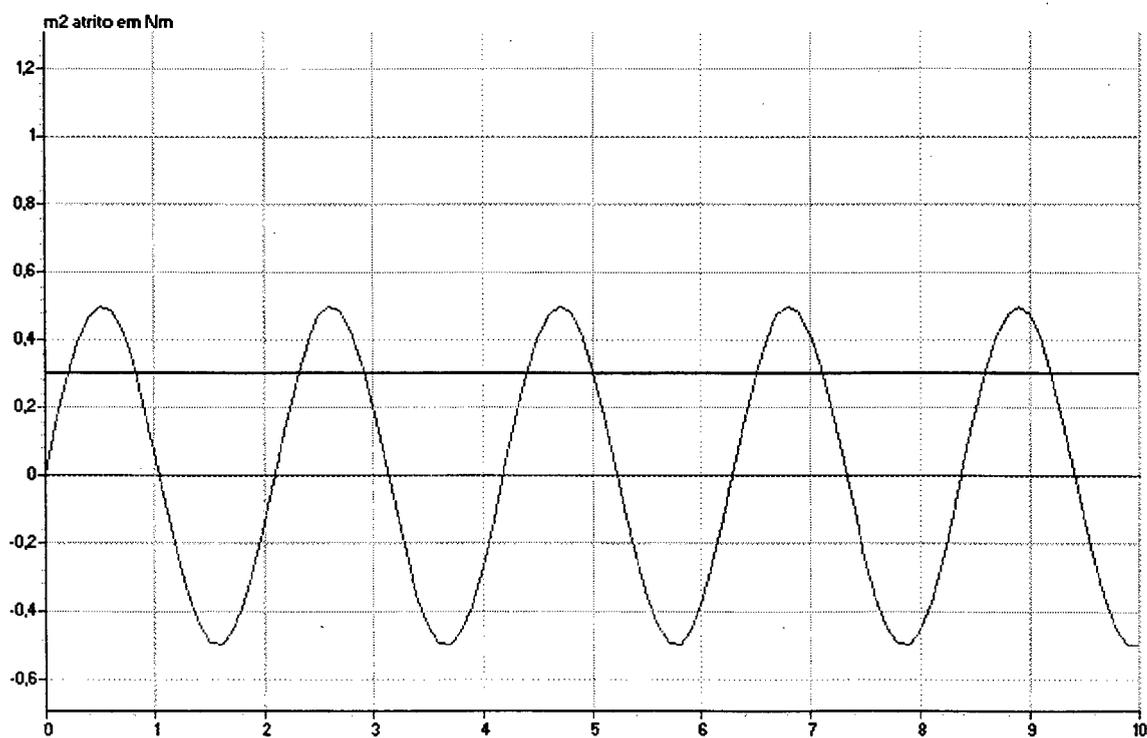
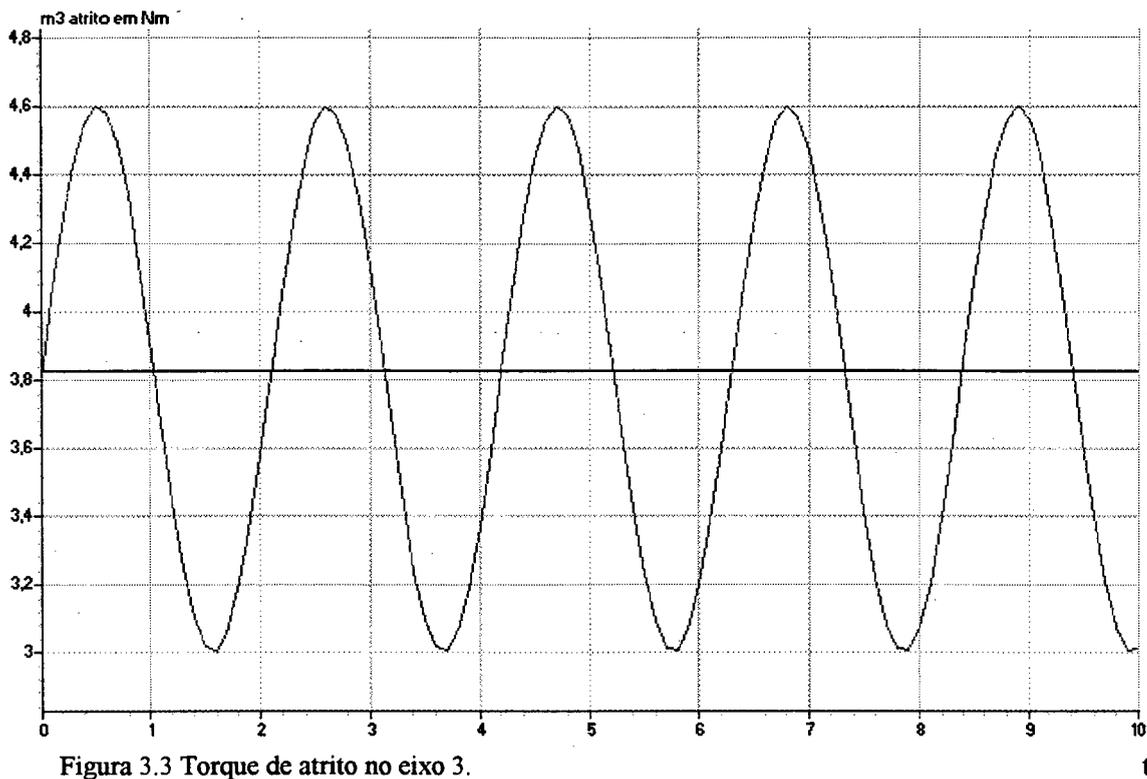


Figura 3.2 Torque de atrito no eixo 2.

t em seg



3.5.2 DETERMINAÇÃO DOS TORQUES E POTÊNCIAS DEVIDOS AOS ESFORÇOS CENTRÍFUGOS E DE CORIÓLIS

Para a determinação destes torques foram acionados dois dos eixos da estrutura a cada vez, com velocidades de 3 rd/s e 3,33 rd/s, respectivamente. Escolheu-se estas velocidades por serem próximas da metade de uma revolução por segundo, limite de velocidade que se considera adequado aos propósitos da estrutura, e não serem valores com um múltiplo inteiro o que evita manutenção de fase entre os movimentos dos eixos, fazendo uma varredura de todas as combinações de posições possíveis.

As figuras de 3.4 a 3.18 mostram torques e potências sobre cada um dos eixos quando são acionados um eixo com velocidade de 3 rd/s e um outro eixo com velocidade de 3,33 rd/s. As potências requeridas dos motores das juntas não acionadas não são mostradas por serem todas nulas, pois

apesar dos acionadores destas juntas terem de resistir aos torques de Coriolis gerados pelas juntas acionadas, não há movimento para gerar o produto torque x velocidade angular.

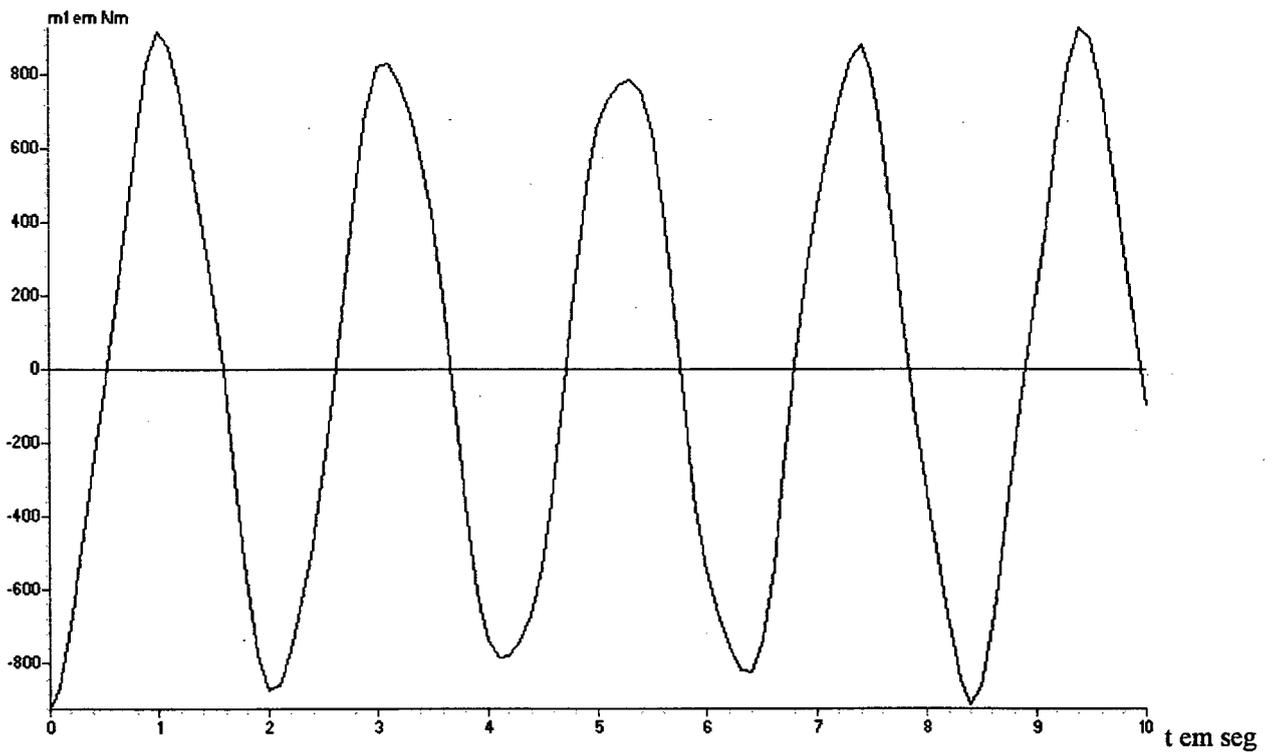


Figura 3.4 Torque na junta 1 quando 2 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

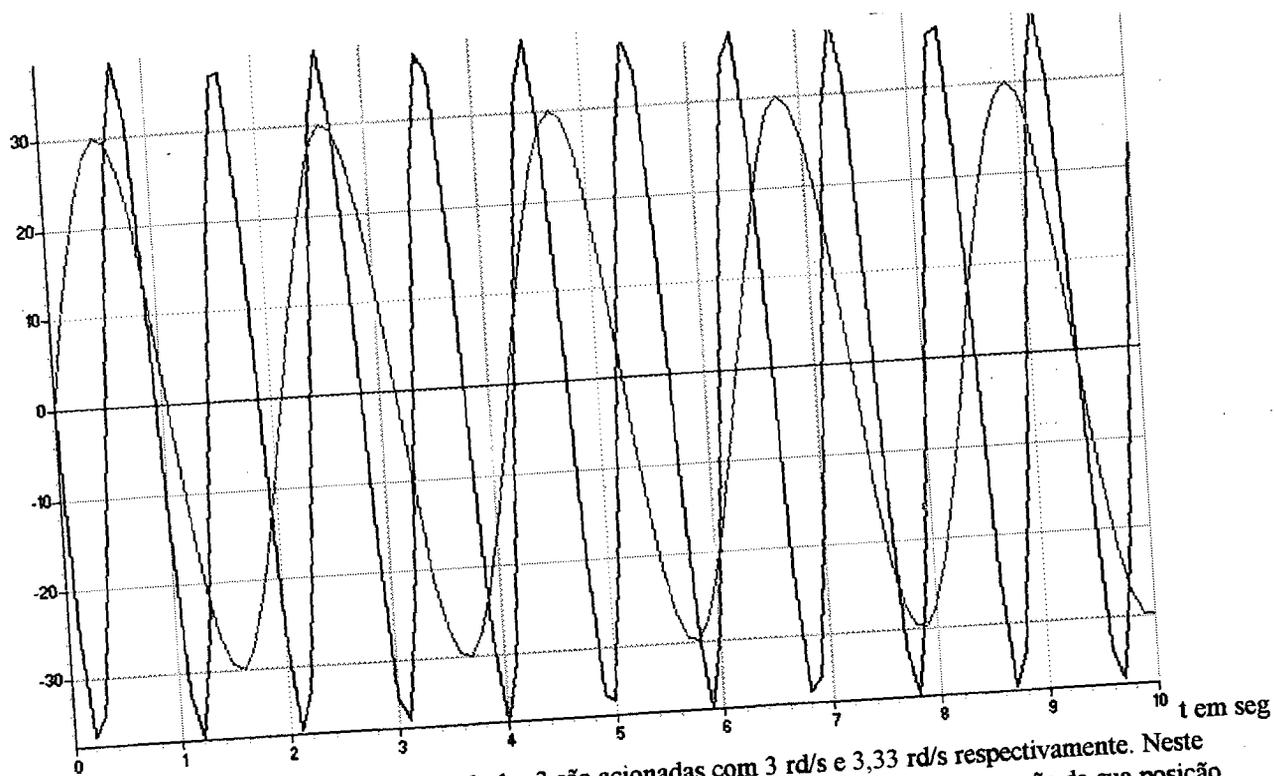


Figura 3.5 Torque na junta 2 quando 1 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente. Neste caso, e em alguns outros, foi plotado também o seno do ângulo da junta, para se ter noção da sua posição.

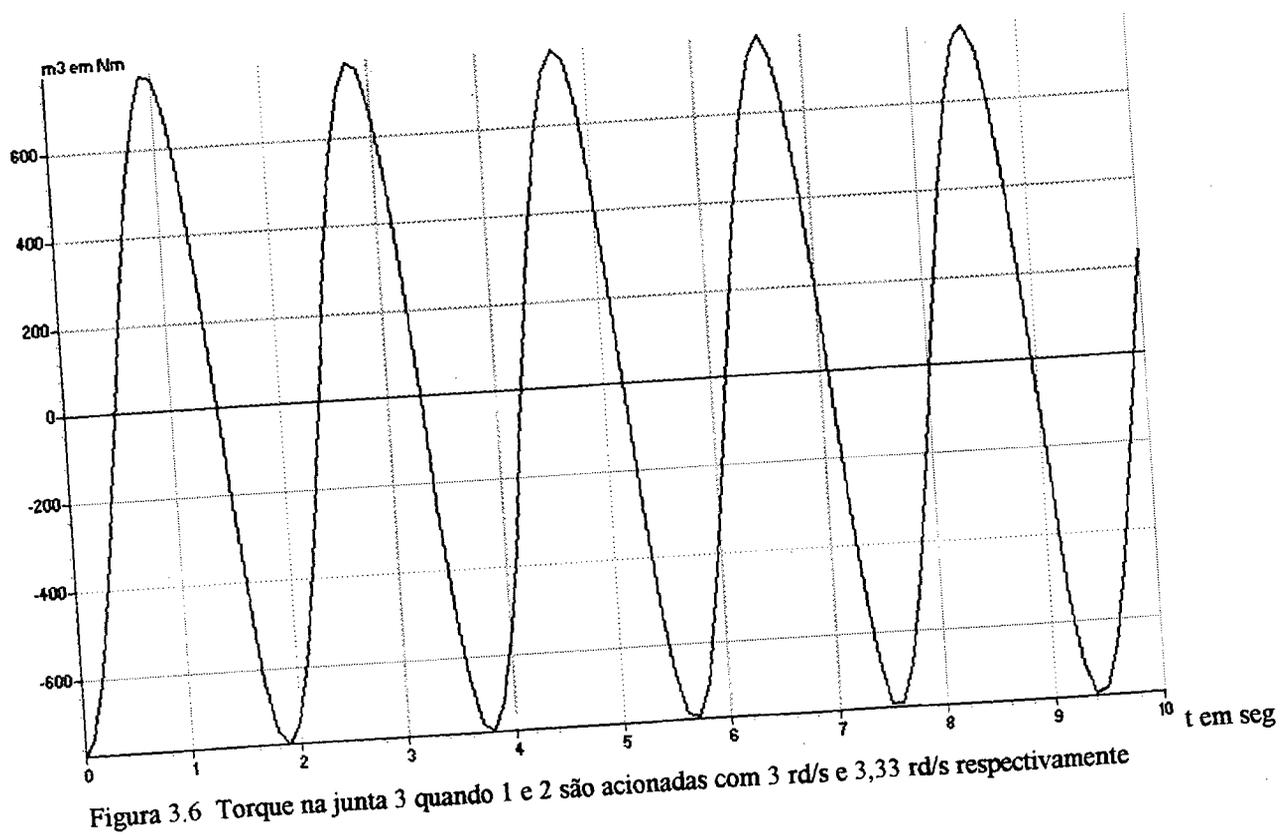


Figura 3.6 Torque na junta 3 quando 1 e 2 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

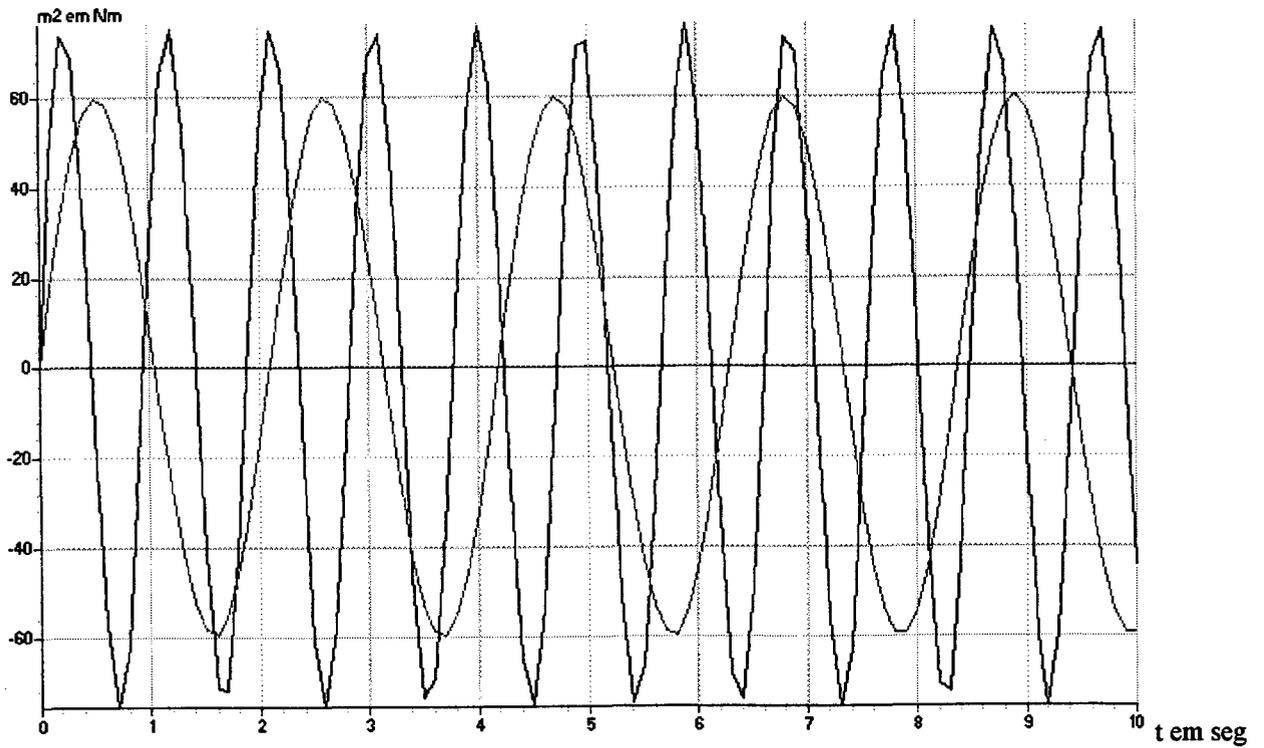


Figura 3.7 Torque na junta 2 em Nm quando 2 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

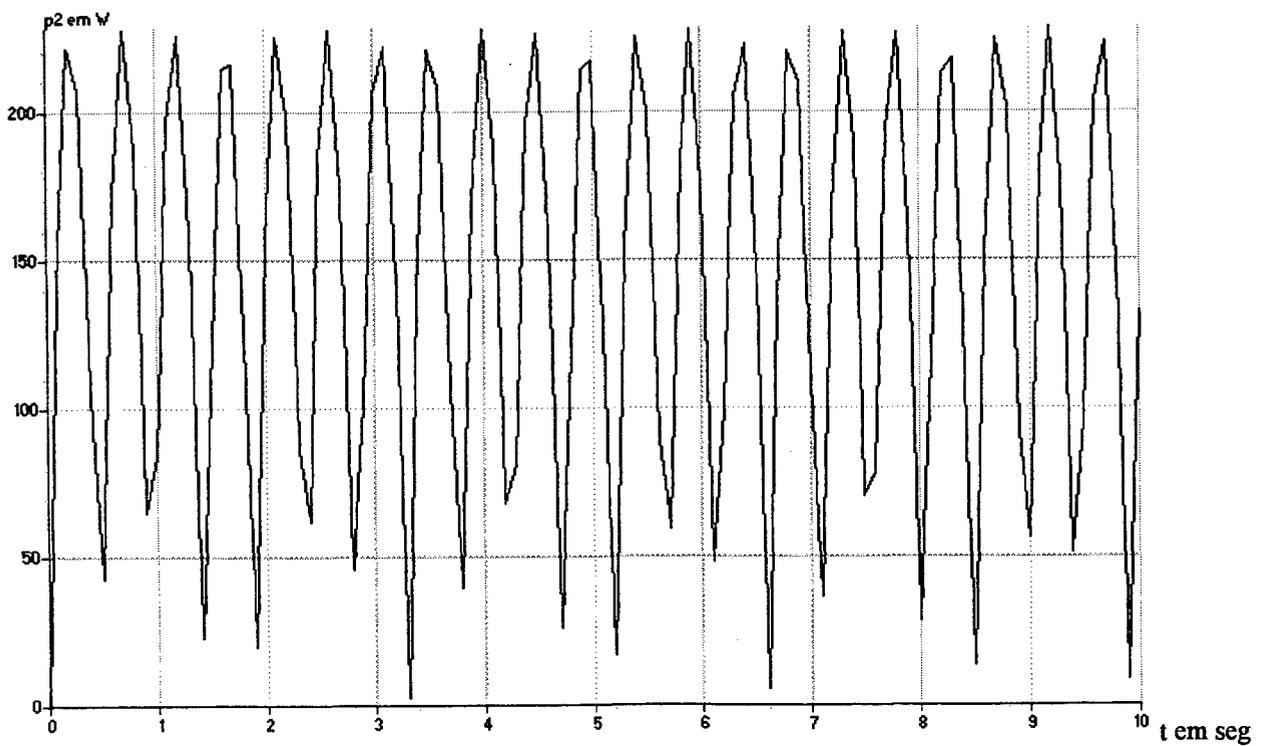


Figura 3.8 Potência na junta 2 quando 2 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

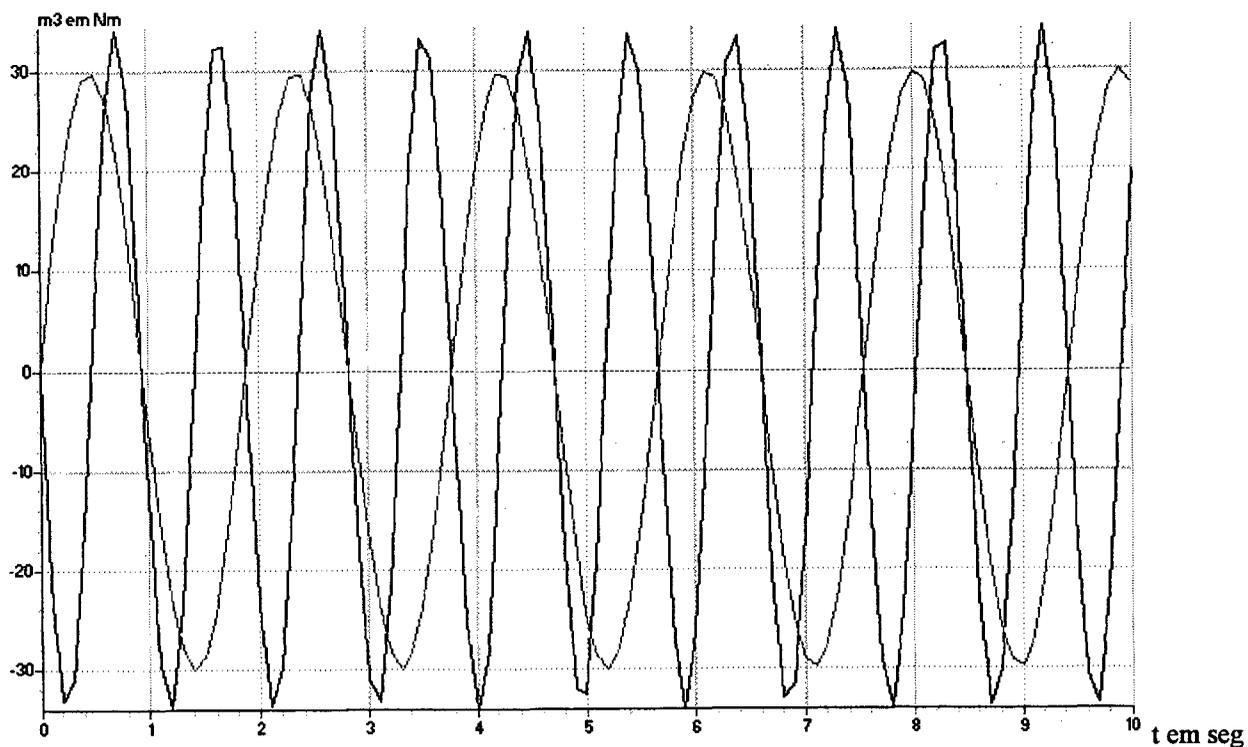


Figura 3.9 Torque na junta 3 quando 2 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

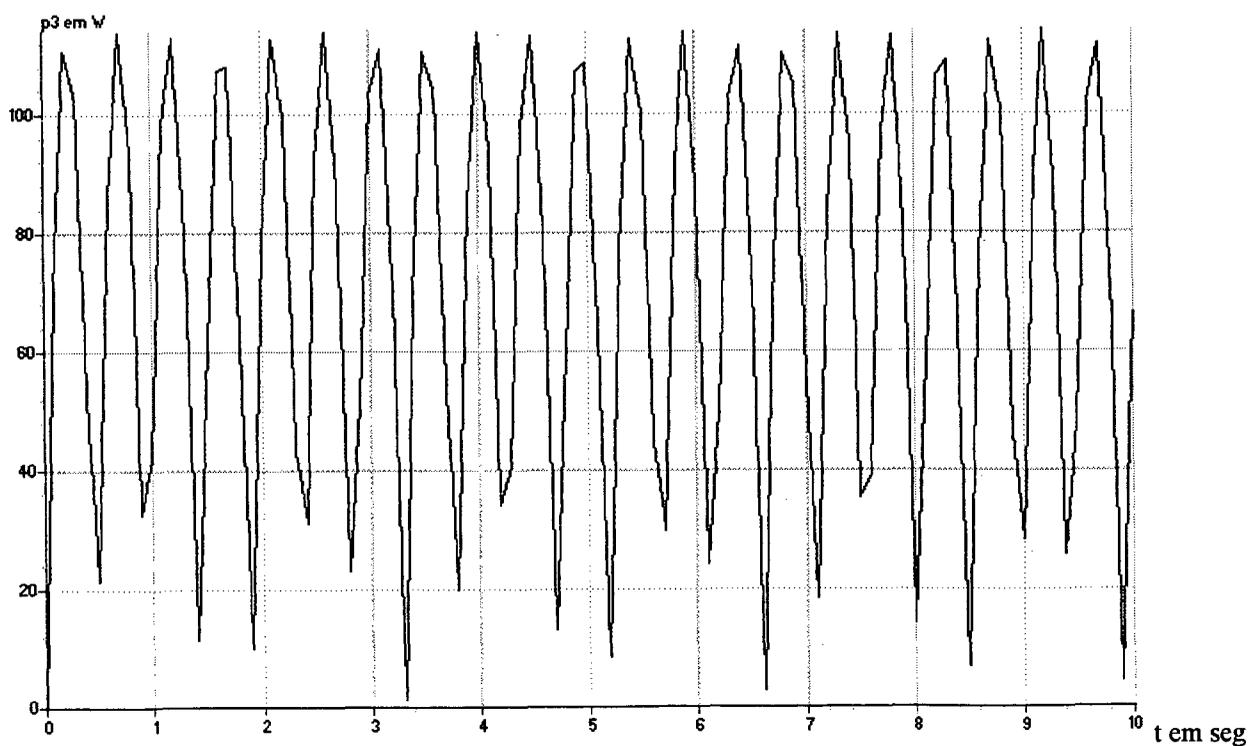


Figura 3.10 Potência na junta 3 quando 2 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

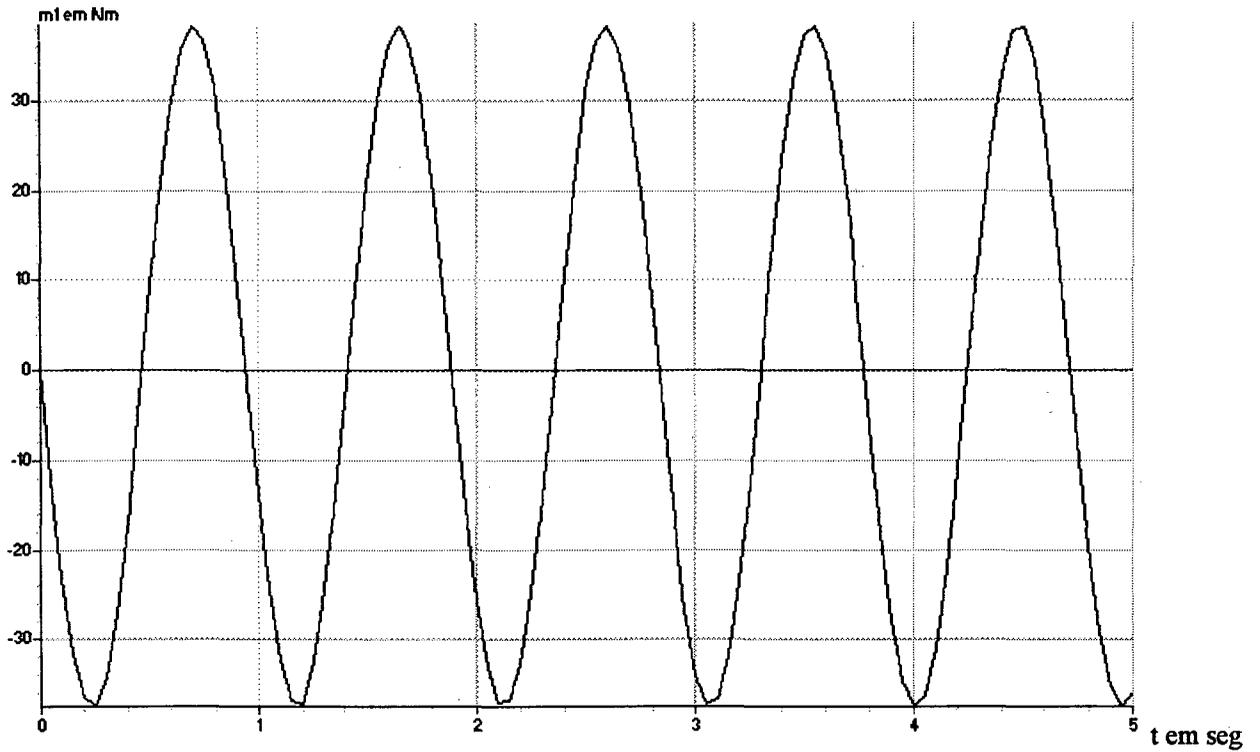


Figura 3.11 Torque na junta 1 quando as juntas 1 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

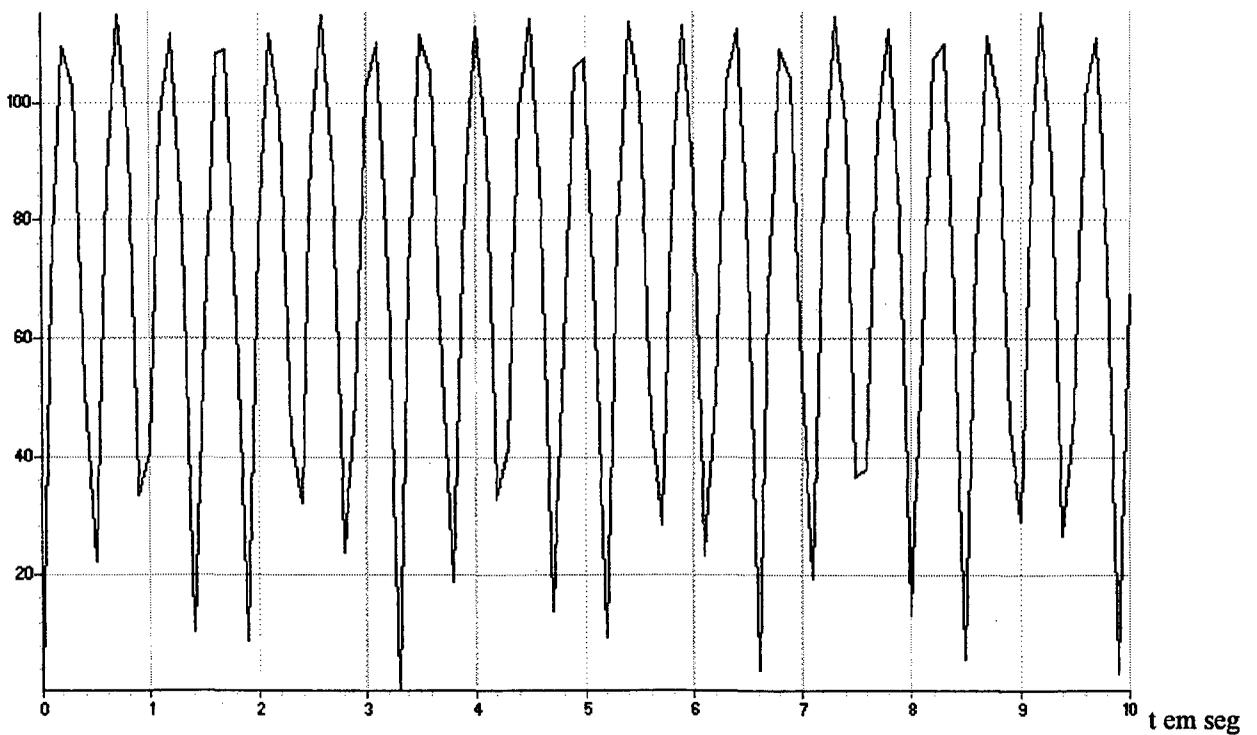


Figura 3.12 Potência na junta 1 quando as juntas 1 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

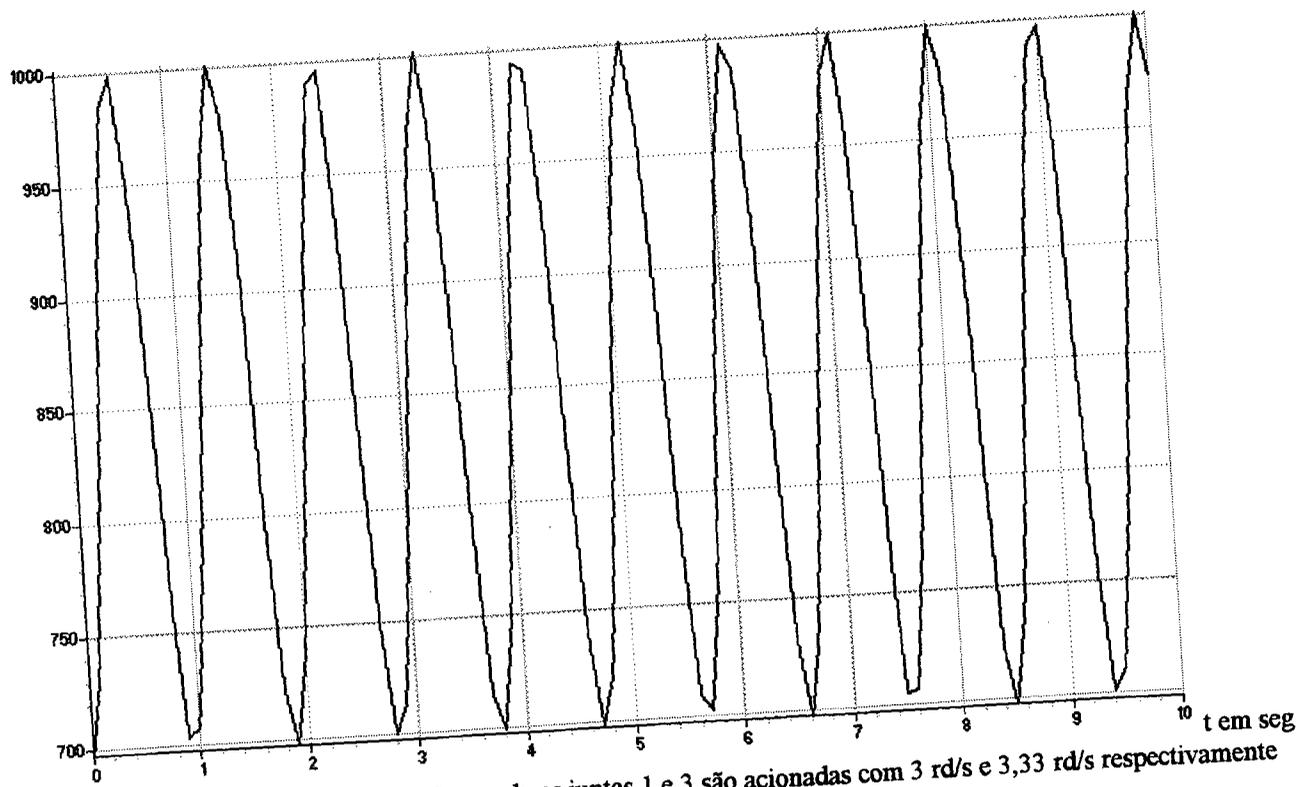


Figura 3.13 Torque na junta 3 quando as juntas 1 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

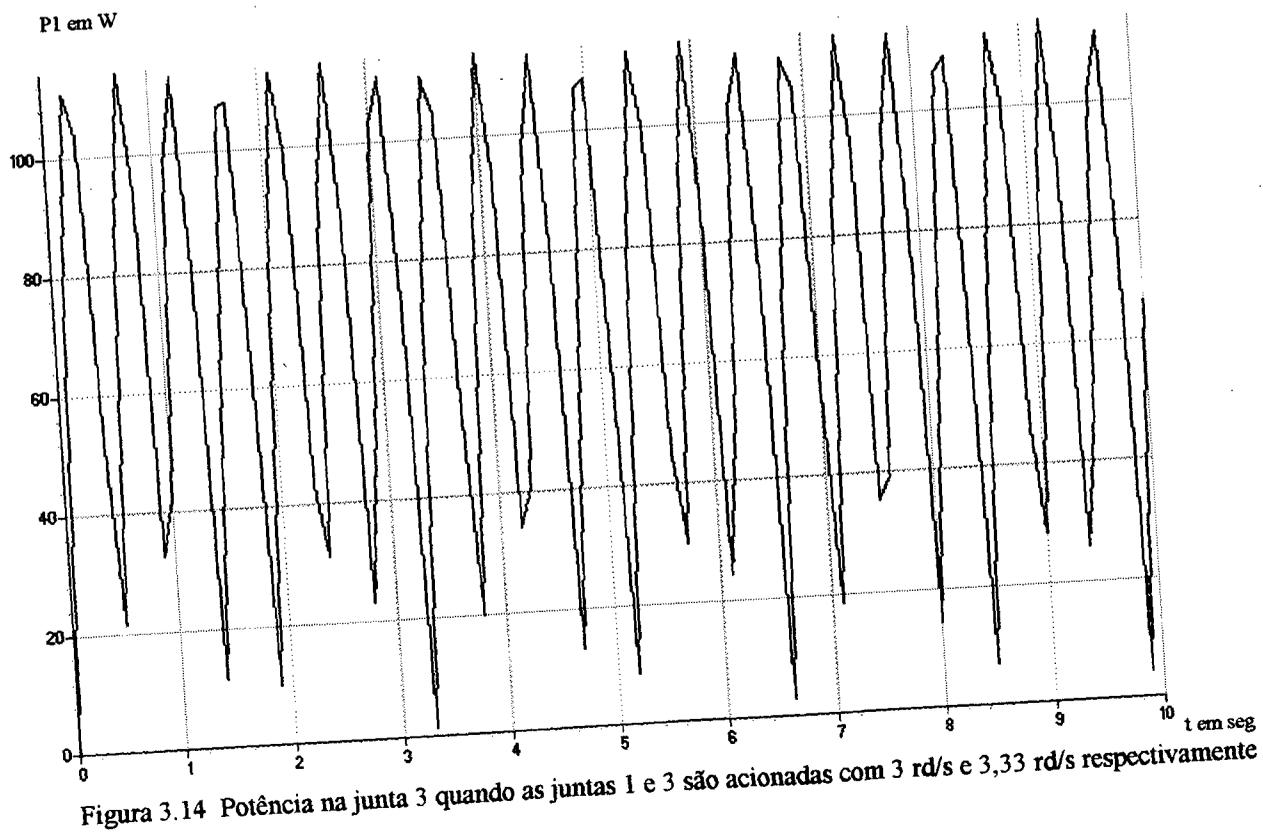


Figura 3.14 Potência na junta 3 quando as juntas 1 e 3 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

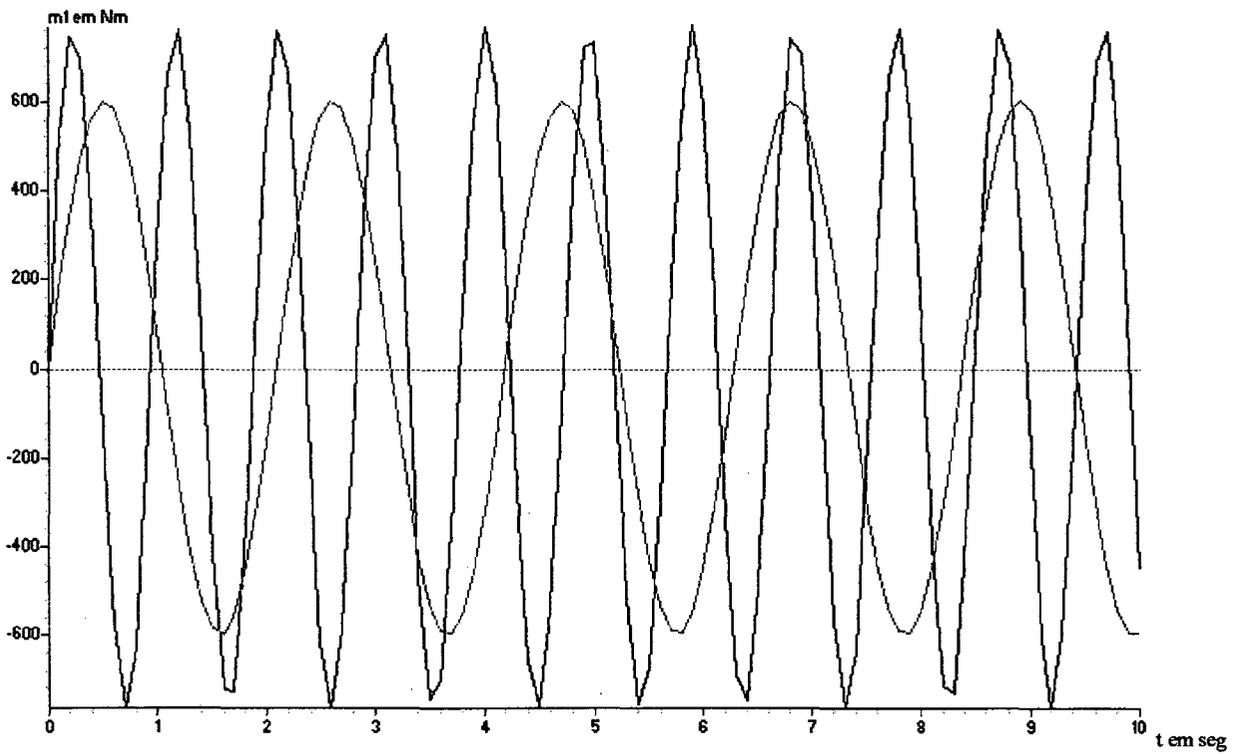


Figura 3.15 Torque na junta 1 quando as juntas 1 e 2 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

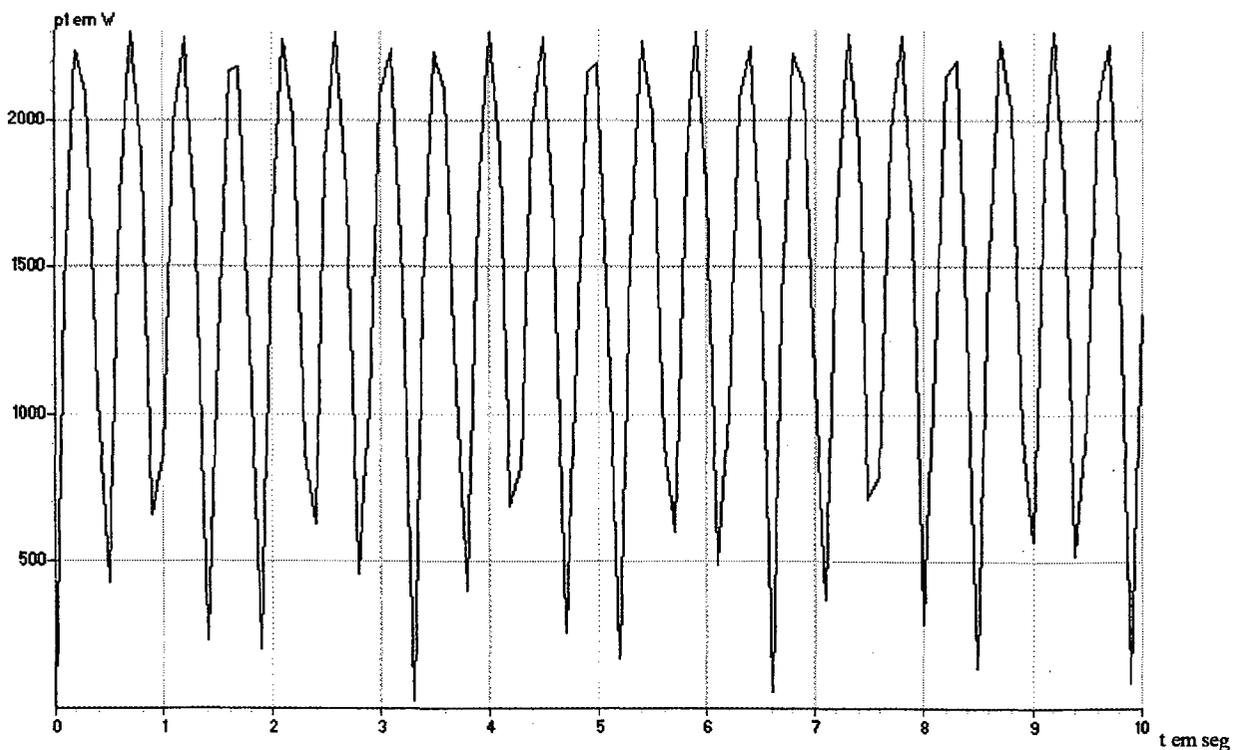


Figura 3.16 Potência na junta 1 quando as juntas 1 e 2 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

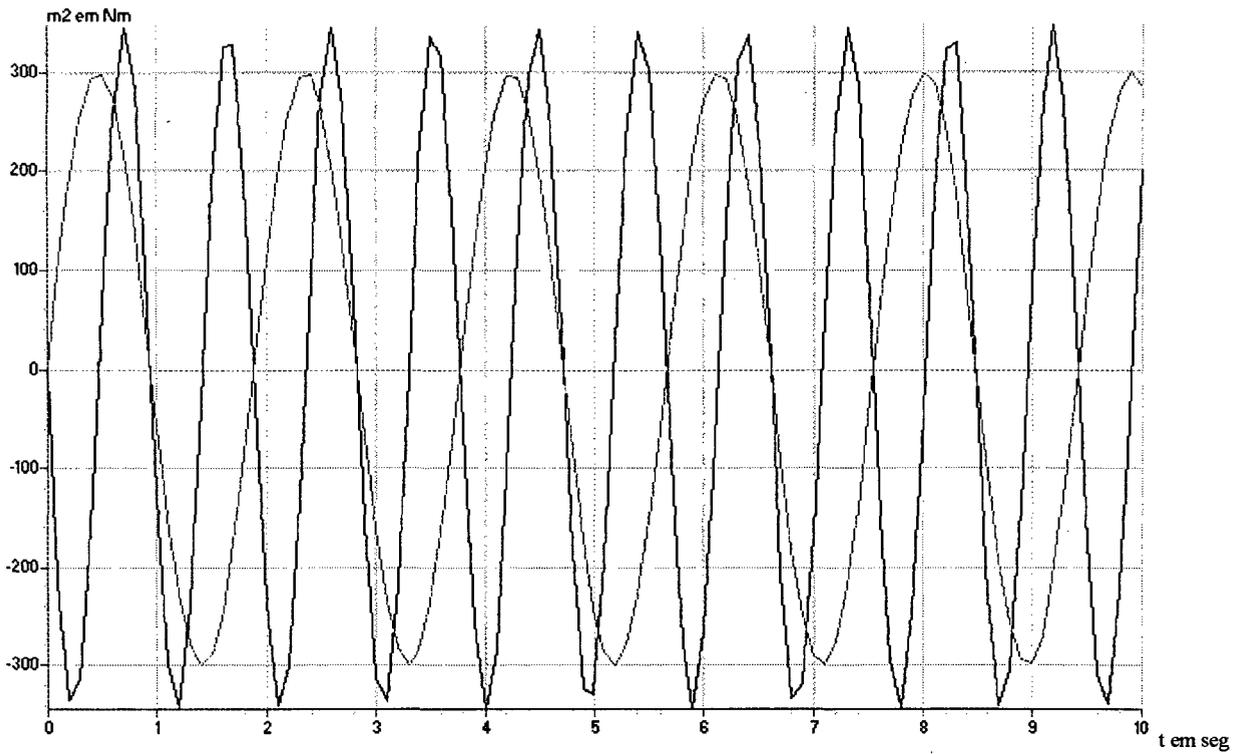


Figura 3.17 Torque na junta 2 quando as juntas 1 e 2 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

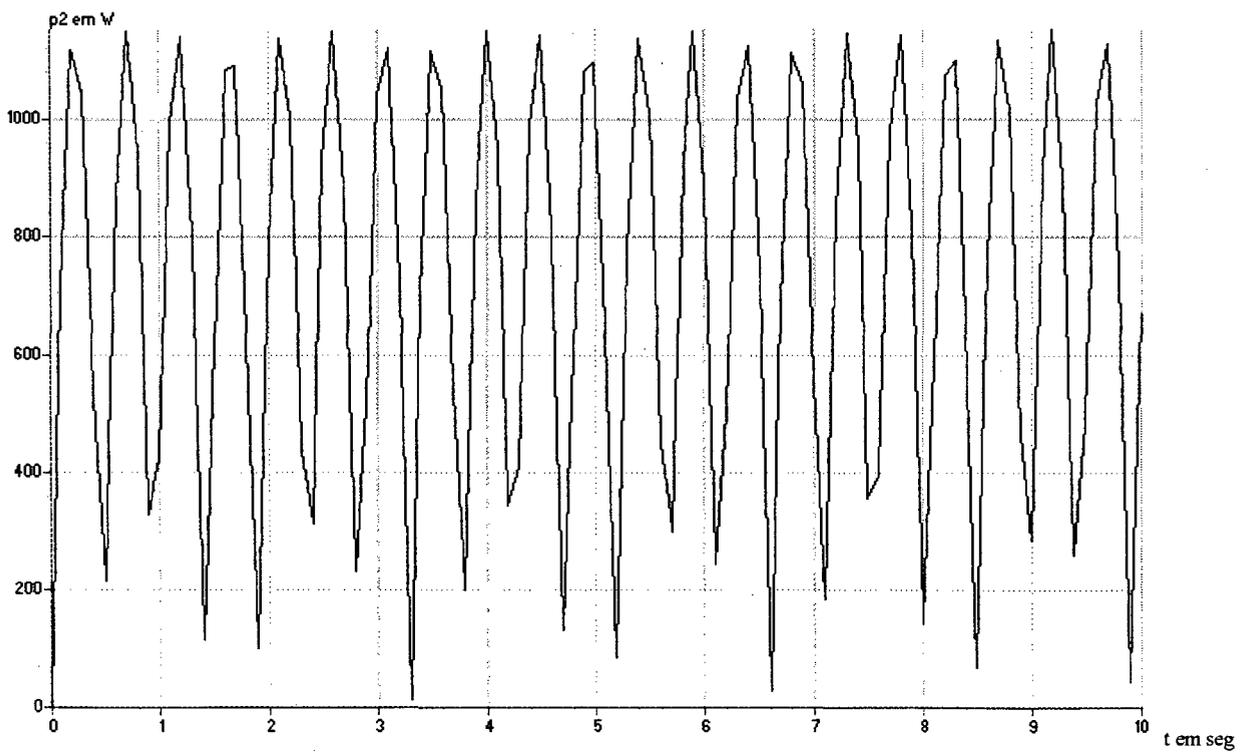


Figura 3.18 Potência na junta 2 quando as juntas 1 e 2 são acionadas com 3 rd/s e 3,33 rd/s respectivamente

3.5.3 TORQUES INERCIAIS

Para a verificação da magnitude dos torques devidos às componentes inerciais foram zerados os valores dos fatores de atrito e excentricidades e cada uma das juntas foi acionada a seu tempo com uma aceleração angular de 2rd/s^2 , considerado valor limite de projeto. O tempo de simulação será o suficiente para que a velocidade limite de 3.14rd/s seja alcançada. É importante lembrar que os torques e potências verificados nas juntas 1 e 2 não correspondem apenas aos esforços para movimentar os segmentos adjacentes, mas também os posteriores, ou seja, o torque na junta 1, por exemplo, gira o garfo levando consigo a estrutura dos anéis (juntas 1 e 2). As figuras 3.19 a 3.21 mostram os resultados destas simulações.

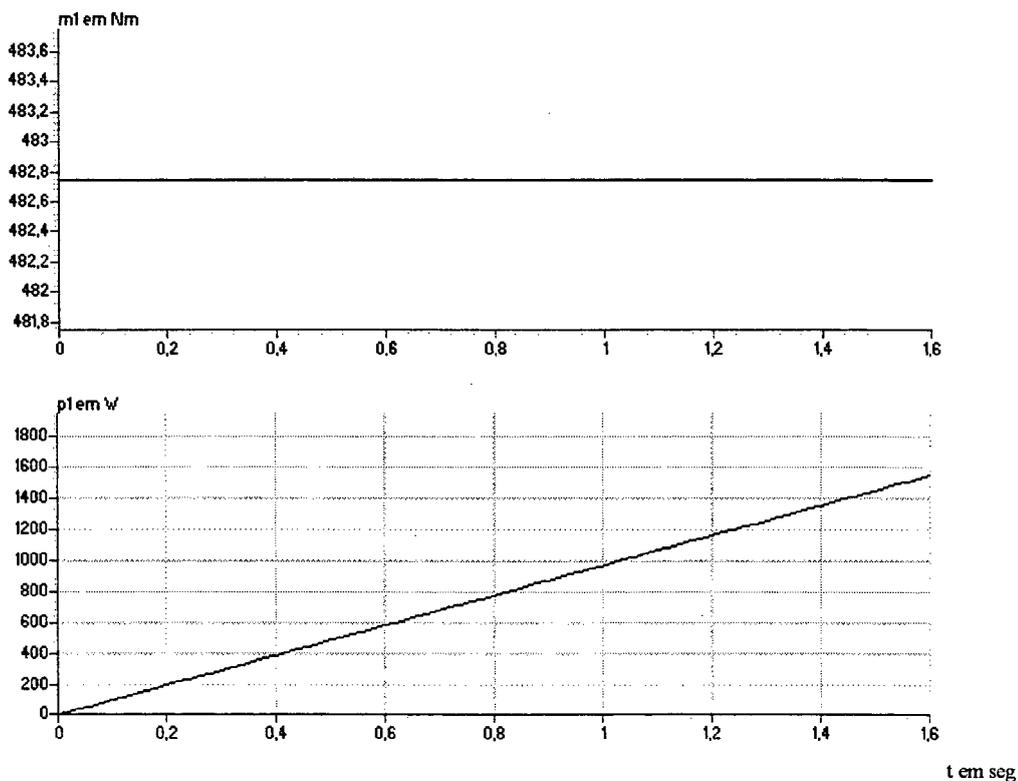


Figura 3.19 Superior: componente inercial do torque na junta 1.
Inferior: potência requerida na junta 1 para vencer o torque inercial a 2rd/s^2 .

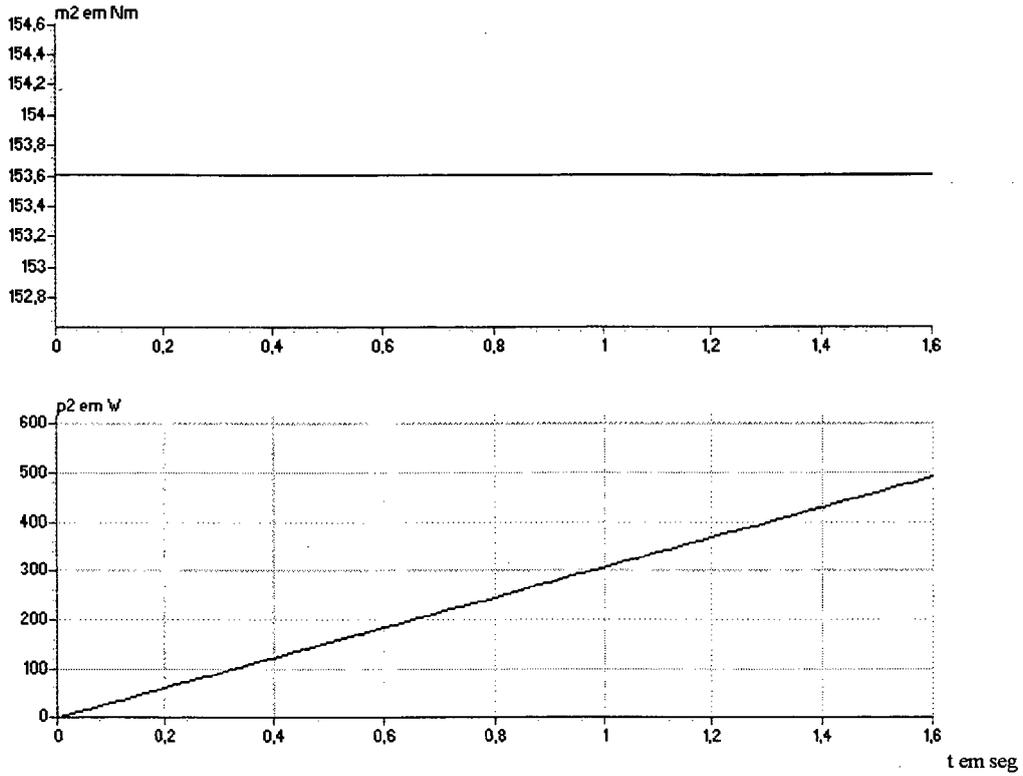


Figura 3.20 Superior: componente inercial do torque na junta 2.
Inferior: potência requerida na junta 2 para vencer o torque inercial a 2 rd/s^2 .

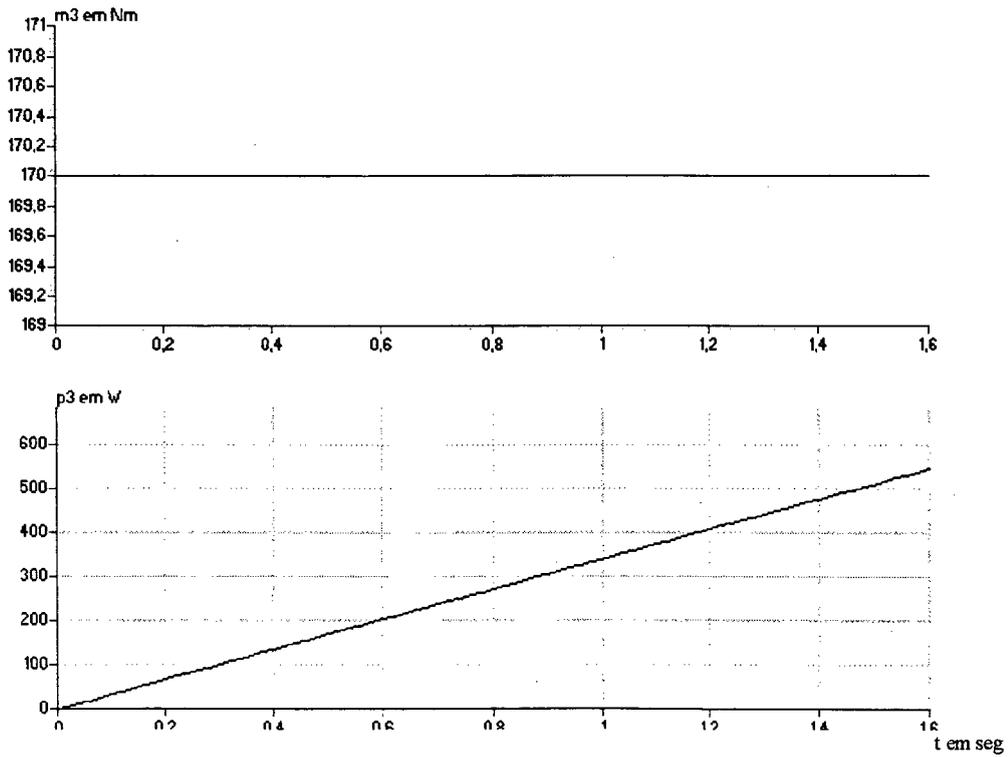


Fig. 3.21 Superior: componente inercial do torque na junta 1. Inferior: potência requerida na junta 1 para vencer o torque inercial a 2 rd/s^2 .

3.5.3 DETERMINAÇÃO DOS TORQUES E POTÊNCIAS MÁXIMOS

Para a determinação dos torques e potências máximos foi feita uma varredura através de uma bateria de simulações que, a cada vez, considerava dois eixos com velocidade de $3,14 \text{ rd/s}$ e o terceiro com aceleração de 2 rd/s^2 . Para cada um destes três casos foram investigadas diferentes posições com passo de variação de trinta graus.

Os maiores esforços encontrados podem ser observados na figura 3.22. As variações mais importantes acontecem para diferentes configurações do eixo 2, que orienta a estrutura formada pelos dois anéis.

Do ponto de vista da sua magnitude, os esforços mais importantes são aqueles devidos às componentes centrífugas e de Coriolis, seguidos pelos torques de origem inercial, gravitacional e de atrito, nesta ordem, considerando os limites de velocidade e aceleração dados.

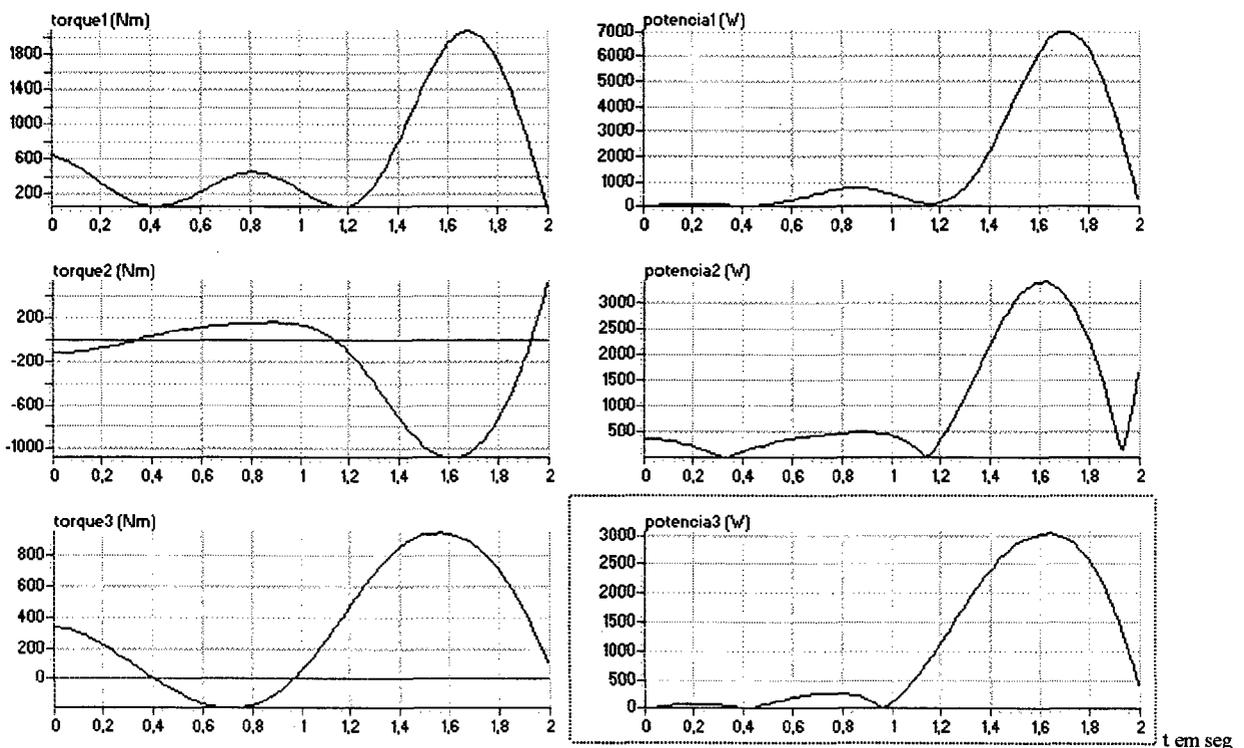


Figura 3.22 Valores máximos de torque e potência encontrados respeitando os limites de projeto de acelerações de 2 rd/s^2 e velocidades de $3,14 \text{ rd/s}$.

3.6. Conclusão: Neste capítulo foi estudado o equacionamento dinâmico da estrutura do simulador por Lagrange e utilizado o modelo estabelecido por [5] para simulação em computador identificando as parcelas de torques inercial, de Coriolis, gravitacionais e de atrito a serem vencidas pelo acionamento.

CAPÍTULO 4. CONTROLE

4.1 INTRODUÇÃO

Este capítulo tem como objetivo caracterizar a estrutura mecânica do simulador do ponto de vista do seu controle. Mais uma vez serão usadas as teoria desenvolvidas para uso na robótica, fazendo-se agora uma analogia da estrutura do simulador com os três primeiros graus de liberdade de um manipulador com juntas rotativas, ou seja, os graus de liberdade responsáveis, no robô, pelo posicionamento do pulso. Serão também apresentadas as principais estratégias de controle que vem sendo utilizadas ou desenvolvidas para o controle de estruturas segmentadas rígidas ou flexíveis.

4.2 CARACTERIZAÇÃO DA ESTRUTURA

A estrutura foi modelada como sendo uma cadeia de corpos rígidos ligados por juntas também rígidas. Foi considerado também que os corpos que constituem cada um dos segmentos da estrutura estão ligados às juntas de forma a girarem segundo um de seus eixos principais de inércia, o que levou à obtenção de tensores de inércia representados por matrizes diagonais, ou seja, contendo apenas os momentos de inércia e não os produtos de inércia.

O modelamento do atrito contém uma incerteza bastante grande e necessita de validação ou levantamento após a construção do protótipo. Estas incertezas do modelamento podem inviabilizar certas estratégias de controle, por provocar erros inaceitáveis ou mesmo instabilidades. Tal pode acontecer devido a esforços adicionais não levados em conta no modelamento como os gerados, por exemplo, por algum tipo de atrito, produto de inércia, ou à excitação de modos naturais da planta, que embora considerada rígida, apresenta certa flexibilidade.

A dificuldade do projeto de um controlador ou mesmo de uma estratégia definida para o controle da planta, deve-se, além das incertezas de modelamento acima citadas, também à incerteza dos índices de desempenho requeridos, limites de erro toleráveis e frequência mínima de aplicação da lei de controle, pois não dependem de um requisito mecânico facilmente quantificável mas sim da sensibilidade do piloto em perceber diferenças de comportamento entre o simulador e o veículo real. Quanto à frequência de aplicação da referência, espera-se que uma provável flexibilidade das juntas e

segmentos da planta seja suficiente para filtrar pequenas descontinuidades de trajetória, permitindo frequências um pouco menores, o que só será definitivamente estabelecido após a construção e ensaio do protótipo, uma vez que o modelamento de tal característica, além de complexo, carregaria consigo elevado grau de incertezas, além do que, sempre restaria a dúvida de como o piloto sentiria a suavidade da trajetória.

Embora esteja sendo usadas abordagens desenvolvidas para a robótica para o estudo da estrutura do simulador, há entre esta e os manipuladores robóticos certas diferenças que tornam ainda mais crítica a já difícil tarefa de controlá-la, quais sejam:

- O grande porte da estrutura, o que implica em elevada inércia e peso.
- A grande variação da carga útil (piloto)
- Operação em altas velocidades tornando relevantes esforços de atrito e, principalmente, componentes não lineares de acoplamento entre os eixos.
- Trajetória definida *on-the-fly*, o que impede qualquer tipo de pré processamento, elevando o peso da computação de tempo real.
- Características da trajetória, que pode ser totalmente irregular, aumentando o risco de que limites toleráveis de erro sejam ultrapassados ou modos naturais da estrutura sejam excitados.
- Tem-se maior interesse nas acelerações e velocidades do que nas posições, embora estas últimas tenham de ser controladas para que haja reação aos esforços gravitacionais, centrífugos e de Coriolis.

4.3 DINÂMICA DOS ATUADORES

Os modelos dos manipuladores tem normalmente como entrada o torque nas juntas. No entanto, ao se fechar a malha de controle esta deve incluir a dinâmica dos atuadores e das reduções. Será considerada nesta seção a dinâmica dos motores CC [3] que será o tipo de atuador utilizado no projeto.

O torque motor, τ_M , se relaciona com a corrente por:

$$\tau_M = K_1 \cdot \phi \cdot i_a \quad (4.1)$$

onde: τ_M é o torque do motor em Nm

φ é o fluxo magnético em Webers

K_1 é uma constante física dependente das características construtivas do motor.

Considerando φ constante, o que é razoável, especialmente no caso de motores de ímã permanente, então

$$\tau_M = K_M \cdot i_a \quad (4.2)$$

i_a é feita proporcional à potência de saída do motor pela força contra-eletromotriz V_b , produzida pelo movimento do rotor imerso no campo magnético, dada por

$$V_b = K_2 \varphi \omega$$

onde V_b é a força contra-eletromotriz induzida dada em volts

ω é a velocidade angular do rotor em rd/s

K_2 é uma constante de proporcionalidade que leva em conta, entre outros fatores, o número de espiras do bobinamento da armadura.

Novamente para φ constante pode-se escrever

$$V_b = K_b \omega \quad (4.3)$$

onde K_b é a constante de força contraeletromotriz. A equação da corrente de armadura é

$$L \frac{di_a}{dt} + Ri_a = V - V_b \quad (4.4)$$

onde L e R são respectivamente a indutância e a resistência dos bobinamentos da armadura

V é a tensão de alimentação do motor.

Até agora foi visto o modelo elétrico do motor, mas é necessária também a sua modelação dinâmica, para tal, o motor será considerado como estando acoplado a uma redução de relação 1:r. No projeto da estrutura do simulador r é da ordem de 1/150 em todos os eixos, ou seja, a redução de velocidade é da ordem de 150 vezes. A equação de movimento do conjunto motor-redução é

$$J_M = \frac{d^2 \theta_M}{dt^2} + B_M \frac{d\theta_M}{dt} = \tau_M - r\tau \quad (4.5)$$

onde J_M é a soma das inércias do motor e da redução vistos no eixo do motor

B_M é o atrito na redução

r é o inverso do fator de redução

θ_M é a posição angular do motor

τ é o torque transmitido à estrutura e τ_M é o torque gerado pelo motor dado por

$$\tau_M = K_M i_a \quad (4.6)$$

escritas no domínio da frequência as equações elétricas e dinâmicas ficam, respectivamente

$$(sL + R) I_a(s) = V(s) - K_b(s)\theta_M(s) \quad (4.7)$$

$$(s^2 J_M + s B_M) \theta_M(s) = K_M I_a(s) - r \tau(s)$$

que leva ao diagrama de blocos da figura 4.1

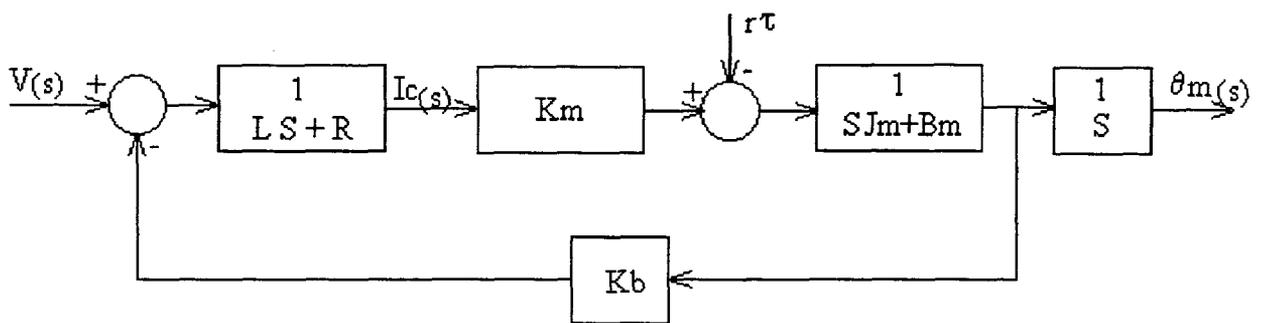


Figura 4.1 Diagrama em blocos do acionador elétrico.

Pelo diagrama da figura 4.1 pode-se perceber a vantagem, do ponto de vista de controle, de se ter uma elevada redução pois assim os torques de perturbação da carga entram no sistema (malha de controle) bastante atenuados.

Para facilitar o processamento e devido ao fato de normalmente a constante de tempo elétrica do motor ser bem menor do que a constante de tempo mecânica, ou seja

$$\frac{J_M}{B_M} \gg \frac{L}{R} \quad (4.8)$$

a indutância da armadura, L , pode ser desprezada, o que leva ao diagrama simplificado da figura 4.2.

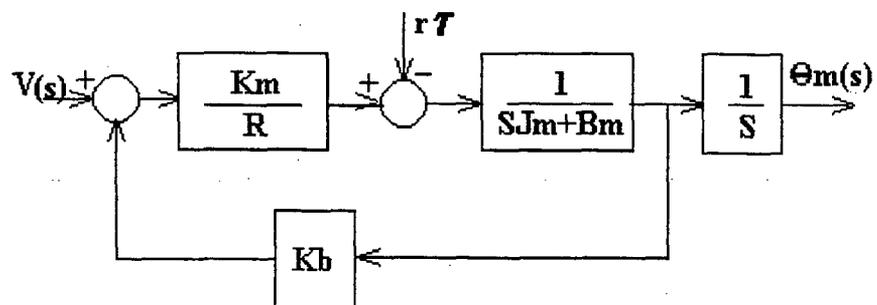


Figura 4.2 Malha de controle da posição do motor desprezando sua constante de tempo elétrica.

Se o diagrama da figura 4.2 fosse usado para estabelecer a malha para controlar θ_M em cada um dos eixos da estrutura todo o acoplamento da dinâmica da estrutura estaria sendo negligenciado e seria encarado pelo controle de cada um dos eixos como perturbações introduzidas pelo torque de carga τ . Assim, pode-se dizer que se

$$\tau_K = \sum_{j=1}^n H_{i,j} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j + G_i \quad (4.9)$$

é o torque visto pela saída da redução em cada um dos eixos k (com $k = 1, \dots, n$), então o atuador (motor) pode ser controlado por

$$J_M \ddot{\theta}_{M_k} + (B_M + K_b + K_M/R) \dot{\theta}_{M_k} = (K_M/R) V - r_k \tau_k \quad \text{com } k=1, \dots, n \quad (4.10)$$

A grande vantagem desta estratégia de controle é a sua simplicidade, pois uma vez que os acoplamentos causadores de fortes não-linearidades são negligenciados, o controle do atuador resulta linear. Mas para tanto ainda é necessário fugir à dependência que certos parâmetros tem em relação à configuração instantânea assumida pela estrutura. J e B são então tomados como uma média dos valores que assumem ao longo do volume de trabalho. Então pode-se escrever a equação 4.10 na forma [1] e [3]:

$$J_{ef} \ddot{\theta}_{M_k} + B_{ef} \dot{\theta}_{M_k} = K V - r_k \tau_k \quad (4.11)$$

onde J_{ef} é um valor médio de $J_M + r_k d_{kk}(q)$
 B_{ef} é $B_M + K_b + K_M/R$
 K é K_M/R

4.4 COMPENSADOR PROPORCIONAL DERIVATIVO

Um compensador PD é implementado aplicando-se uma lei de controle do tipo

$$U_{(s)} = K_P (\theta_{(s)}^d - \theta_{(s)}) - K_D S \theta_{(s)} \quad (4.12)$$

onde K_P e K_D são respectivamente os ganhos proporcional e derivativo

θ^d e θ são respectivamente as trajetórias desejada e real.

diagrama de blocos da figura 4.3 mostra a implementação de uma malha PD no controle do atuador.

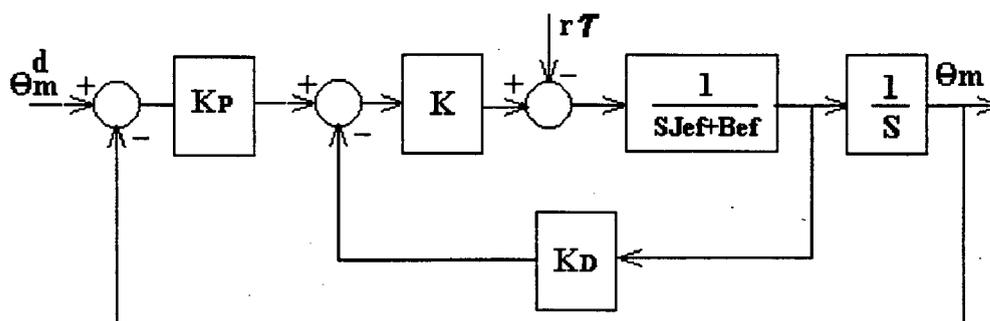


Figura 4.3 Controle PD para uma junta.

Este esquema de controle é estável para K_p e K_d positivos e perturbações limitadas. Sob uma perturbação $D(s)$ constante o erro de regime e_{ss} é dado por

$$e_{ss} = -rD / KK_p \quad (4.13)$$

donde se depreende que o erro pode ser feito pequeno pelo aumento de K_p ou do fator de redução $1/r$. No entanto, este procedimento tem limites construtivos, no caso de $1/r$, ou de saturação, no caso de K_p .

Uma forma de se obter menores erros de regime sem apelar para altos ganhos é pelo uso do controle integral pela adição de um termo K_i/s conforme o diagrama em blocos da figura 4.4.

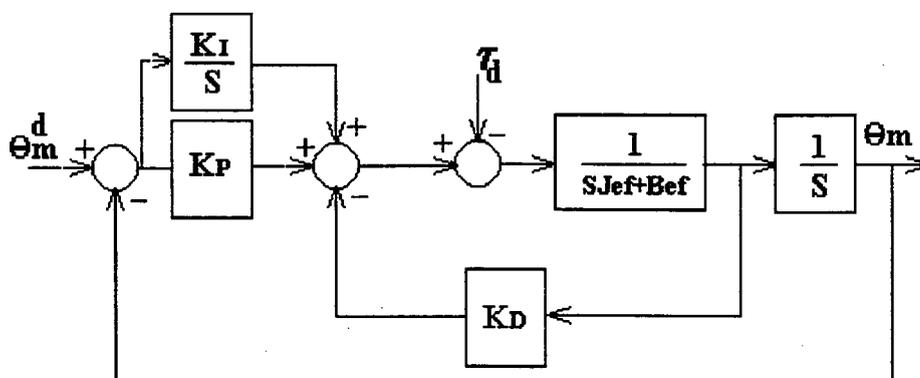


Figura 4.4 Controle PID de uma junta.

4.5 CONTROLE NO ESPAÇO DE ESTADOS

Na estratégia de controle PID o desempenho tem de ser conseguido pelo ajuste dos valores de apenas três parâmetros, K_p , K_i , K_d , o que limita o projeto dos compensadores.

Uma forma alternativa para o projeto dos compensadores onde o projetista pode agir sobre um maior número de parâmetros para estabelecer o desempenho do sistema é o projeto no **espaço de estados** que tem a vantagem adicional de não requerer as transformações do domínio do tempo para o da frequência e vice-versa.

Nesta estratégia de controle as variáveis de interesse são as posições e suas derivadas, sendo o número de variáveis dependente do grau do sistema a ser controlado. Uma apresentação completa desta abordagem é apresentada em [48] enquanto que a aplicação ao controle de posição é feita por Spong [3].

Os métodos vistos até agora são os mais utilizados no controle de robôs industriais, porém só funcionam bem em baixa velocidade ou em tarefas de posicionamento ponto a ponto onde sobrepassagens relativamente grandes podem ser toleradas desde que a posição alvo seja alcançada dentro de um certo limite de tempo. No entanto, quando se tem especial interesse na trajetória e, mais ainda, se a aplicação exige altas velocidades, os acoplamentos entre eixos começam a gerar perturbações grandes demais para serem rejeitadas apenas pelo uso de ganhos elevados e uma escolha crítica da alocação dos pólos do sistema. A solução, a princípio, é usar as equações dinâmicas do sistema para desacoplar e linearizar o controle. O mais conhecido dos métodos por dinâmica inversa é o do **torque computado**.

4.6 TORQUE COMPUTADO

Para se seguir trajetórias complexas em alta velocidade os métodos por realimentação são insuficientes. O método do torque computado, ao contrário, é um método *feedforward*, ou seja, procura antecipar as perturbações. Isto é possível porque, se as perturbações externas são imprevisíveis, aquelas devidas aos acoplamentos entre eixos e à gravidade podem ser previstas com base nas equações dinâmicas da estrutura. O esquema de controle por torque computado pode ser visto na figura 4.5 onde se percebe a inclusão de

$$d^d = \sum_{j=1}^n H_{i,j} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i \quad (4.14)$$

A intenção é que o termo d^d (que tem unidade de torque, donde o nome do método) cancele o termo D em suas componentes de acoplamentos não lineares de inércia, coriólis, centrífugas e gravitacionais. No entanto, o erro de *tracking* só se anulará na condição ideal de uma modelagem perfeita e computação sem truncamentos. O esforço computacional aliás, é o grande porém desta estratégia. No caso de robôs industriais com tarefas pré determinadas, d^d pode ser pré-computado, o que não é o caso do controle de um simulador interativo.

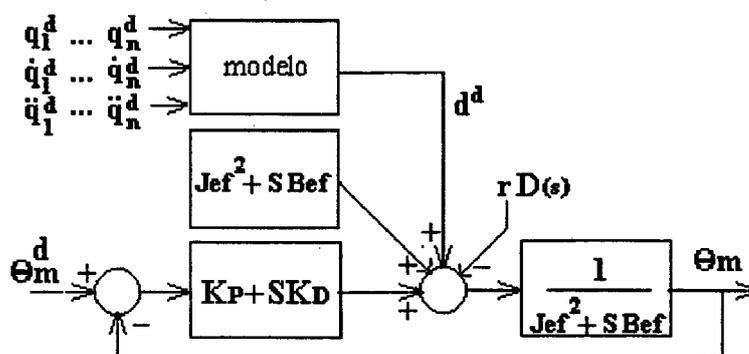


Figura 4.5 Esquema de controle por junta baseado em torque computado.

4.6.1 MODELAGEM DA DINÂMICA PARA USO EM TEMPO REAL

Para uso no projeto e simulação de uma estrutura, visando seu dimensionamento, otimização e escolha dos acionadores, a formulação ideal para obtenção do modelo é a de Lagrange, por permitir a individualização dos efeitos de inércia, centrífugos e de coriólis, gravitacionais e de atrito. No entanto, para uso em tempo real em sistemas de controle utilizando dinâmica inversa, o modelamento por Lagrange torna-se muito pesado computacionalmente sendo, a depender da capacidade computacional disponível, preferível o uso da formulação de Newton-Euler iterativa ou de algoritmos dela adaptados.

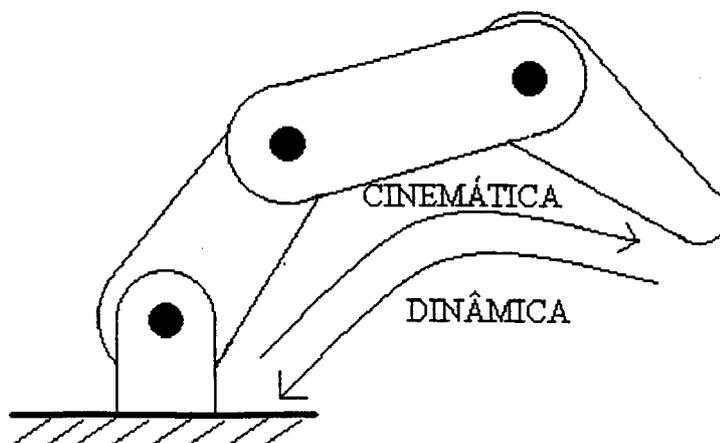
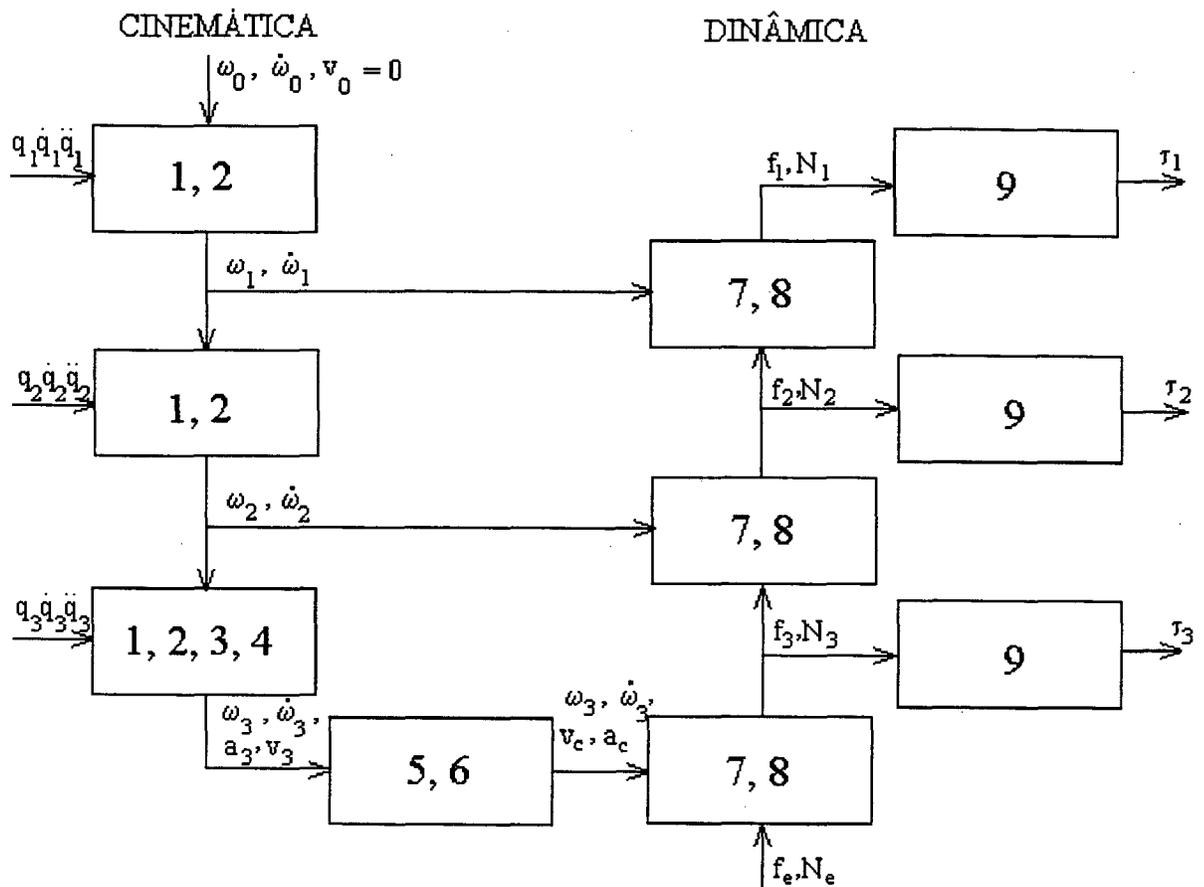


Figura 4.6 Sentidos dos cálculos iterativos da cinemática e da dinâmica em uma estrutura em cadeia.

Em [2] é apresentado o importante algoritmo de Luh-Walker-Paul (figura 4.7) que minimiza de uma proporção de n^4 para n o número de operações de multiplicação e soma necessárias para o cálculo da dinâmica inversa de uma estrutura com n graus de liberdade, pela integração dos cálculos da cinemática aos da dinâmica, o que possibilita um melhor aproveitamento dos resultados intermediários obtidos. Assim, enquanto a cinemática é calculada a partir da base da cadeia em direção à sua extremidade, determinando velocidades e acelerações em cada segmento, a dinâmica é computada em sentido oposto, da extremidade para a base, determinando os esforços descarregados sobre cada junta pelos segmentos que a sucedem (figura 4.6)



- 1) $\omega_{i+1} = \omega_i + \dot{q}_{i+1} b_i$
- 2) $\dot{\omega}_{i+1} = \dot{\omega}_i + \ddot{q}_{i+1} b_i + \omega_i \times \dot{q}_i b_i$
- 3) $v_{i+1} = v_i + \omega_{i+1} \times r_{i,i+1}$
- 4) $a_{i+1} = a_i + \dot{\omega}_i \times r_{i,i+1} + \omega_{i+1} \times (\omega_i \times v_{i,i+1})$
- 5) $v_c = v_i + \omega_i \times r_{i,c}$
- 6) $\dot{v}_c = \dot{v}_i + \dot{\omega}_i \times r_{i,c} + \omega_i \times (\omega_i \times r_{i,c})$
- 7) $f_{i-1,j} = f_{i,j+1} - m_i \cdot g + m_i \cdot a_c$
- 8) $N_{i-1,j} = N_{i,j+1} - r_{i,c} \times f_{i,j+1} + r_{i-1,i} \times f_{i-1,j} + I_i \dot{\omega}_i + \omega_i \times (I_i \cdot \omega_i)$
- 9) $\tau_i = b_{i-1}^T \cdot N_{i-1,i}$

Figura 4.7 Fluxograma do algoritmo LWP para a estrutura do simulador.

Dhanaraj [8] apresenta um método modificado do algoritmo de Luh-Walker-Paul usando computação simbólica que transforma operações com matrizes em operações escalares. Por exemplo, onde aparece no algoritmo LWP uma expressão do tipo

$$a \times r + w \times w \times r \quad (4.14)$$

ela poderá ser substituída por

$$[\lambda] r \quad (4.15)$$

onde $[\lambda]$ é uma matriz pré computada cujos elementos são funções de w e a .

Do ponto de vista de custo computacional para calcular a dinâmica inversa de uma junta rotacional, o algoritmo modificado emprega 92 multiplicações e 76 adições enquanto que o LWP original utiliza 135 multiplicações e 123 adições.

Os métodos de controle baseados em referência modelo ou dinâmica inversa tem como principal desvantagem o grande custo computacional devido à necessidade do cálculo da dinâmica a cada vez que a lei de controle é aplicada; além disso, como sempre haverá uma margem de incerteza nos parâmetros do modelo, estes levarão ao não cancelamento total das não linearidades e, por consequência, a erros de rastreo por vezes inaceitáveis.

Assim, ao invés de se tentar um refinamento cada vez maior do modelo dinâmico, que se tornará cada vez mais dispendioso computacionalmente, pode-se optar por um controlador robusto ou adaptativo. Na primeira filosofia, que busca a robustez, a idéia é criar um controlador capaz de corrigir os erros de rastreo resultantes das falhas de modelamento. Na segunda, o objetivo é criar um controlador baseado numa lei de adaptação de forma a buscar a minimização dos erros de rastreo a cada instante pelo constante atualização de parâmetros do modelo ou do compensador.

4.7 CONTROLE MULTIVARIÁVEL

Até aqui cada um dos eixos da estrutura foi tratado como um sistema SISO - com uma única entrada e uma única saída, no qual os acoplamentos entre eixos eram tratados como perturbações externas. Uma outra forma de tratar o controle de estruturas em cadeia é o uso de controle multivariável, ou seja, com várias entradas e saídas onde os efeitos não lineares devidos aos acoplamentos entre os eixos são considerados. Esta forma dá um tratamento mais rigoroso ao controle de estruturas

segmentadas que são, na verdade, sistemas multivariáveis não lineares, pois os efeitos de acoplamento não são mais tomados como uma média constante e sim calculada a cada vez que se aplica a lei de controle. A expressão vetorial 4.16 pode então ser usada para o controle

$$(D_{(q)} + J)\ddot{q} + C_{(q,\dot{q})}\dot{q} + B\dot{q} + G_{(q)} = u \quad (4.16)$$

onde J é uma matriz diagonal dos elementos $(1/r_k^2)J_{mk}$ e o vetor de entrada u tem componentes $u_k = (Km/r_k R) V_k$.

Um controle PD, ainda por junta, pode ser implementado por

$$u = Kp(q^d - q) - Kd.\dot{q} \quad (4.17)$$

Pode ser demonstrado que a aplicação de u leva um rastreo assintótico das posições de referência na ausência de gravidade. Se houver um termo gravitacional, u deve tornar-se

$$u = Kp(q^d - q) - Kd.\dot{q} + G_{(q)} \quad (4.18)$$

de forma a compensar os efeitos de gravidade.

4.7.1 LINEARIZAÇÃO PELA DINÂMICA INVERSA

A equação 4.16 pode ser reescrita de forma condensada como

$$M_{(q)}\ddot{q} + h_{(q,\dot{q})} = u \quad (4.19)$$

onde $M=D+J$ e $h = C\dot{q} + B\dot{q} + g$

O objetivo da linearização pela dinâmica inversa é encontrar uma lei não linear

$$u = f(q, \dot{q}) \quad (4.20)$$

de modo que colocada em 4.19 resulte em uma malha fechada linear e desacoplada. Segundo Spong [3] uma escolha razoável para u é fazer-la

$$u = M_{(q)}v + h_{(q,\dot{q})} \quad (4.21)$$

onde v torna-se uma nova entrada do sistema, agora linear. Ainda em [3] pode-se encontrar o desenvolvimento da determinação da função v . O sistema de controle resultante possui duas malhas, uma interna de linearização e desacoplamento e uma externa do controle propriamente dito (fig.4.9)

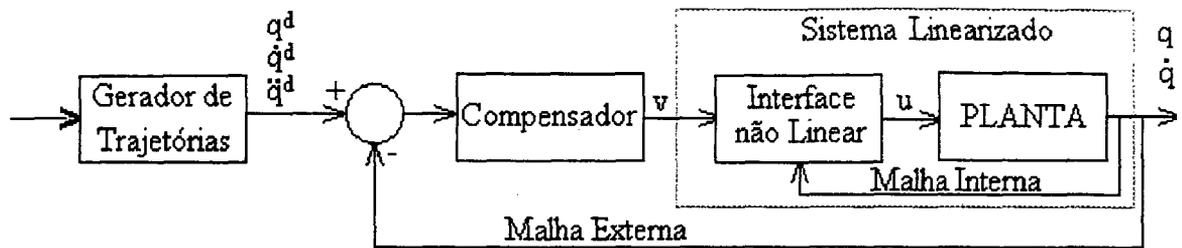


Figura 4.9 Esquema de controle mostrando as malhas interna - linearização - e externa - compensação.

Um esquema de controle como o que acaba de ser visto não avança muito além do plano teórico uma vez que é praticamente impossível fazer um perfeito cancelamento dos efeitos não lineares, pois sempre há um certo grau de incerteza nos parâmetros do modelo. Também em benefício da realizabilidade do controle pode ser interessante a simplificação do modelo a ser utilizado. Assim, uma estratégia interessante para o controle de sistemas dinâmicos complexos é a utilização de esquemas de controle que possam rastrear um sinal de referência com um erro tolerável a despeito de inexatidões de modelamento. A esta estratégia denomina-se controle robusto. Segundo Mulders [7], no projeto de controladores robustos o objetivo é construir um controlador cuja performance tenha a menor sensibilidade possível às incertezas do modelamento, incluindo as dinâmicas de alta frequência não modeladas. Asada [2] apresenta o gráfico (fig.4.10) onde pode-se ter uma idéia qualitativa de que estratégia é mais favorável para o controle de um sistema tomando como parâmetro o erro de rastreamento em função do nível de incertezas do modelo. A medida que suportes computacionais mais potentes tornam-se disponíveis, o ponto de mudança, α , desloca-se para mais próximo do eixo vertical. No extremo um esquema robusto seria capaz de controlar uma planta não linear e acoplada usando um modelo simples e desacoplado, ficando todo o esforço de controle por conta do compensador que poderá ter de recorrer a sinais de grande amplitude e potência para manter o erro de rastreamento dentro de limites aceitáveis, o que pode provocar saturação e *chattering*. O usual é que se recorra aos efeitos de robustez aliados a um modelamento menos rigoroso da planta, ou seja, desconsiderando efeitos de menor monta ou utilizando parâmetros nominais ao invés de reais. Uma abordagem comumente aceita para o projeto de um controlador robusto para uma estrutura em cadeia é definir inicialmente um controle a partir de parâmetros nominais para depois adicionar a parte responsável pela robustez da lei de controle como um todo. O método do torque computado é freqüentemente utilizado para implementar o controle nominal.

Retomando a equação 4.21 pode-se reescreve-la como

$$u_{(t)} = \hat{M}_{(q)}v + \hat{h}_{(q,\dot{q})} \quad (4.22)$$

sendo \hat{M} e \hat{h} os valores nominais

$$\Delta M := \hat{M}_{(q)} - M_{(q)} \quad (4.23)$$

e

$$\Delta h = \hat{h}_{(q,\dot{q})} - h_{(q,\dot{q})} \quad (4.24)$$

representam as incertezas ou erros de modelamento.

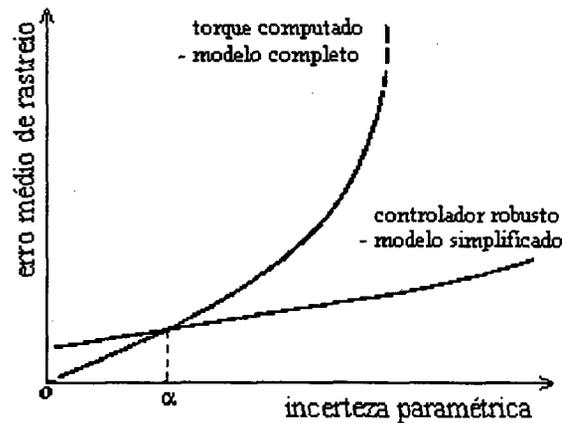


Figura 4.10 Desempenho da estratégia de controle em função da incerteza paramétrica [2].

Há várias abordagens para a implementação de controle robusto de estruturas com dinâmica acoplada desenvolvidas para uso em manipuladores, a maioria ainda em desenvolvimento. Em [9] é feito um levantamento de cinco abordagens para a obtenção de controle robusto de robôs. São elas

- Multivariável Linear
- Controladores a Estrutura Variável
- Controladores Baseados na Passividade
- Controladores por Saturação
- Controladores Adaptativos Robustos

O desenvolvimento de uma abordagem de controle robusto multivariável é apresentado em [3] e [9] onde a lei da malha externa 4.19 é substituída por

$$u = \hat{M}_{(q)}v + \hat{h}_{(q,\dot{q})} \quad (4.25)$$

o que resulta no sistema

$$\ddot{q} = v + \eta_{(v,q,\dot{q})} \quad (4.26)$$

onde a incerteza η é dada por

$$\eta = E_{(q)}v + M^{-1}\Delta h$$

sendo $E = M^{-1}\hat{M} - I$. No espaço de estados 4.26 pode ser escrita

$$\dot{X} = AX + B(v + \eta) \quad (4.27)$$

onde

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

Os vetores de erros no espaço de estados em relação a uma trajetória desejada $q_{(t)}^d$ são

$$e_1 = X_1 - X_1^d = q - q^d$$

$$e_2 = X_2 - X_2^d = \dot{q} - \dot{q}^d$$

os erros de rastreo satisfazem

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = \ddot{q} - \ddot{q}^d$$

$$\dot{e}_2 = v + \eta - \ddot{q}^d$$

vetorialmente

$$\dot{e} = Ae + B(v + \eta - \ddot{q}^d) \quad (4.28)$$

Um esquema como o da figura 4.11 pode ser usado para o controle de estruturas com juntas rotacionais satisfeitas algumas limitações nas incertezas [3] e [9], onde $C_{(s)}$ é um compensador linear.

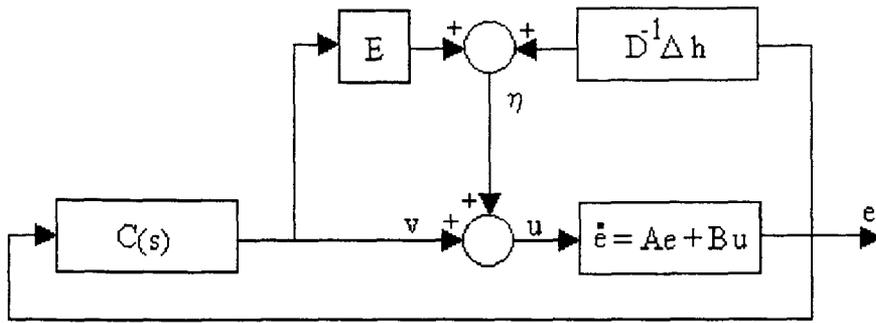


Figura 4.11 Possível esquema de controlador multivariável [9]

4.8 CONTROLE POR SLIDING MODE

Seja o sistema dinâmico SISO

$$\dot{x}_{(t)}^{(n)} = f_{(x)} + b_{(x)} u_{(t)} + d_{(t)} \quad (4.29)$$

onde $u_{(t)}$ é a variável de controle

x é a variável de saída

$$X = [x, \dot{x}, \dots, x^{(n-1)}]^T$$

$d_{(t)}$ é a perturbação

No âmbito do controle robusto $f_{(x)}$ não é conhecida com exatidão, no entanto o grau de incerteza é limitado por uma função de X que é conhecida. Também as perturbações não são conhecidas mas sabe-se que são contidas por função contínua de X e t . Controlar o sistema será então achar o estado X que rastreie um estado de referência $X_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]$ mesmo na presença das incertezas e perturbações. Para que tal seja possível utilizando um controle u finito X_d deve ser tal que o erro inicial seja nulo, ou seja

$$X_{d(t=0)} = X_{(t=0)}$$

sendo \tilde{X} o vetor de erros

$$\tilde{X} = [x - x_d, \dot{x} - \dot{x}_d, \dots]$$

pode-se definir uma superfície $S_{(t)}$ no espaço de estados dada pela equação escalar

$$S_{(x,t)} = 0$$

tendo por definição

$$S_{(x,t)} = \left(\frac{d}{dt} + \lambda \right)^{n-1} \tilde{x} \quad (4.30)$$

sendo λ uma constante escolhida como critério de projeto do controlador.

Fazer $S_{(x,t)} = 0$ é o mesmo que fazer uma equação diferencial linear ter como única solução $\tilde{X} = 0$, dadas as condições iniciais citadas. Assim, fazer o rastreio de um vetor de entrada, X_d , equivale a manter o valor escalar S igual a zero. Pode-se então usar uma lei de controle, u , que fora de $S(t)$ imprima a condição

$$\frac{1}{2} \frac{d}{dt} S^2_{(x,t)} \leq -\eta |S| \quad (4.31)$$

onde η é uma constante positiva que dá a margem tolerável dentro da qual o erro tenderá assintoticamente para zero, a denominada condição de deslizamento. Esta margem também permite que mesmo que as condições iniciais $\tilde{X} = 0$ não sejam verificadas exatamente, ainda assim a superfície $S(t)$ será alcançada em um tempo finito

$$t < |S_{(t=0)}| / \eta \quad (4.32)$$

O projeto de um controlador por modo de deslizamento é feito em duas etapas. Primeiro usa-se uma lei de controle u para a qual a condição de deslizamento 4.31 é verificada. Esta lei ao rastrear uma trajetória pode criar descontinuidades conforme o erro torne-se positivo ou negativo, num efeito conhecido por *chattering*. Este chaveamento além de exigir excessivo trabalho do controle pode excitar modos naturais de alta frequência na planta.

Para diminuir os efeitos nocivos do *chattering* procede-se um segundo passo no projeto do controlador de forma a dar-lhe uma atuação mais contínua, o que na verdade é a otimização do compromisso entre a largura de faixa e a precisão do rastreio. No primeiro passo consegue-se robustez em relação às incertezas e erros de modelamento, enquanto que o segundo passo tem como objetivo tornar o controle imune a problemas devidos a modos estruturais de alta frequência não modelados. Um exemplo de lei adotada no primeiro passo é

$$u = \hat{u} - K \operatorname{sgn}(S) \quad (4.32)$$

onde \hat{u} é a melhor aproximação de uma lei contínua capaz de garantir $\dot{S} = 0$,

K é o ganho de descontinuidade função do vetor de estados

$\text{sgn}(S)$ é 1 se $S > 0$ e -1 se $S < 0$.

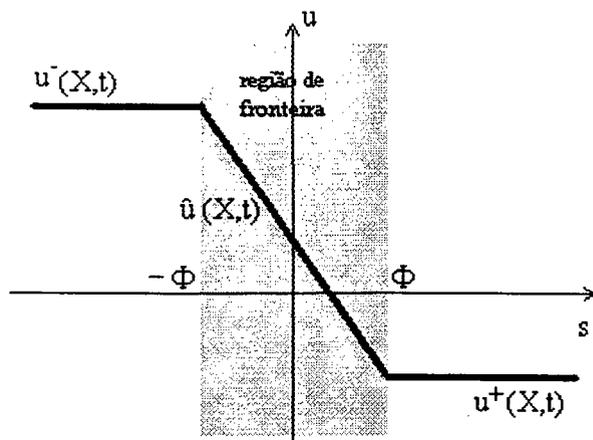


Figura 4.12 Interpolação da lei de controle na região de fronteira [2].

No segundo passo o controle é tornado contínuo em uma faixa de vizinhança em torno de $S=0$. A largura desta faixa pode ser variante no tempo de modo a explorar a máxima faixa passante possível do controle (fig. 4.13)

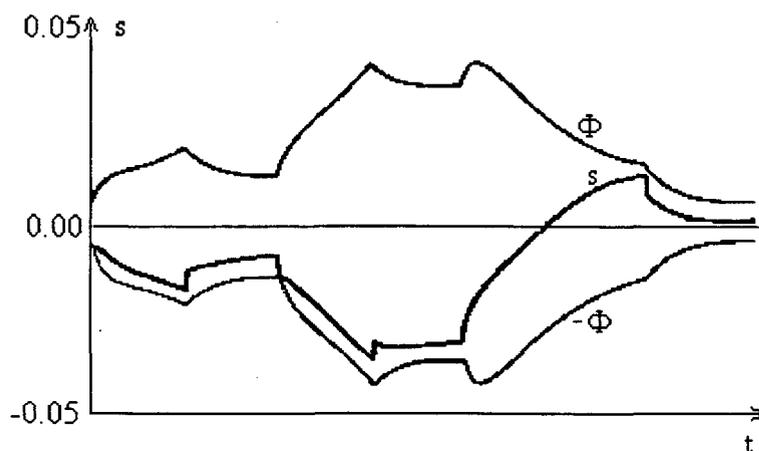


Figura 4.13 Exemplo de trajetória s com margem Φ adaptável [2].

4.9 CONTROLE POR PASSIVIDADE

Esta abordagem baseia-se na passividade das estruturas rígidas como as descritas por 4.33

$$D_{(q)}\ddot{q} + C_{(q,\dot{q})}\dot{q} + G_{(q)} = \tau \quad (4.33)$$

onde as características das matrizes D e C permitem que a inequação 4.35 seja posta como teorema [9]

$$\int_0^t \dot{q} \tau \, dt \geq -\beta \quad (4.35)$$

para qualquer t finito e $\beta > 0$, o que fisicamente significa que o trabalho desenvolvido pelos atuadores é irreversível. O teorema assegura que \dot{q} é um mapeamento passivo de τ o que por sua vez garante a estabilidade da malha fechada desde que as entradas sejam limitadas (*bounded*).

A passividade do sistema que mapeia τ em q e por consequência τ no erro e pode ser demonstrada usando o vetor r obtido por filtragem de e

$$r = -[sI + \frac{K(s)}{s}] e \quad (4.36)$$

$$r = -F(s)^{-1} e \quad (4.37)$$

considere-se então a lei de controle

$$\tau = -\Lambda(s) \dot{e} + u \quad (4.38)$$

aplicando 4.38 a 4.33 tem-se (Fig. 4.14)

$$r = -\Lambda(s) \dot{e} \quad (4.40)$$

ou

$$\dot{e} = -\Lambda(s)^{-1} r \quad (4.41)$$

que garante a estabilidade de \dot{e} porém não garante a estabilidade de e diante de trajetórias variantes no tempo, assegura apenas a sua limitação (*boundedness*). Uma solução proposta é a adoção de um compensador PD com ganhos dependentes de q , o que tem a desvantagem de tornar o controle dependente de valores singulares de $D(q)$ e do conhecimento desta última. Em [10] é apresentada uma descrição mais completa desta estratégia de controle.

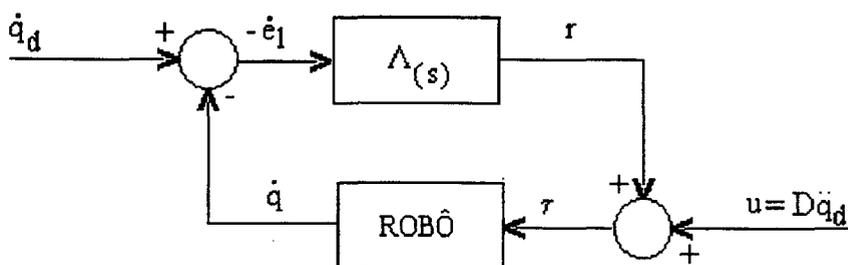


Figura 4.14 Esquema do controle que explora a passividade da planta do robô.

4.10 CONTROLE POR SATURAÇÃO

O controle por saturação assemelha-se ao controle por *sliding mode*, onde há também uma fronteira que delimita a aplicação de duas leis de controle diferentes. Neste caso, no entanto, segundo proposto por [6], a largura da faixa de fronteira não determina o erro em estado de regime. Ela apenas serve para ligar o efeito integral depois que o erro se aproxima de zero, evitando o efeito integral sobre grandes erros, o que costuma causar grandes sobrepassagens. Dessa forma o parâmetro determinante da largura da camada de fronteira pode ser maior, diminuindo os efeitos nocivos dos altos ganhos.

Este controle é dito robusto porque depende do conhecimento das limitações mas não do valor dos parâmetros exatos da planta.

Na proposta de Liu [6] o efeito de saturação é adicionado a um controle nominal

$$\hat{\tau} = \hat{D}_{(q)}\mathbf{a} + \hat{C}_{(q,\dot{q})}v + \hat{G}_{(q)} \quad (4.42)$$

onde

$$e := q - q^d$$

$$\mathbf{a} = \ddot{q} - 2\lambda \dot{e} - \lambda^2 e \quad \lambda > 0$$

$$v := \dot{q} - \dot{e} - \lambda e$$

Por parametrização linear pode-se escrever 4.42 como

$$\hat{\tau} = Y_{(q,\dot{q},v,a)} \hat{p} \quad (4.43)$$

com parâmetros reais tem-se

$$\tau = D_{(q)}\mathbf{a} + C_{(q,\dot{q})}v + G_{(q)} = Y_{(q,\dot{q},v,a)} p \quad (4.44)$$

então

$$\tau = \hat{\tau} + Y_{(q,\dot{q},v,a)} u = Y_{(q,\dot{q},v,a)} (\hat{p} + u) \quad (4.45)$$

onde o termo u é um termo de controle que visa obter robustez contra incertezas paramétricas representadas por $\hat{p} - p$. Substituindo a lei de controle 4.45 na equação do modelo dinâmico 4.16

$$\tau = D_{(q)}\ddot{a} + C_{(q,\dot{q})}v + G_{(q)} = Y_{(q,\dot{q},v,a)}(\hat{p} + u) \quad (4.46)$$

subtraindo 4.46 de 4.44

$$D_{(q)}(\ddot{e} + 2\lambda \dot{e} + \lambda^2 e) + C_{(q,\dot{q})}(\dot{e} + \lambda e) = Y_{(q,\dot{q},v,a)}(\tilde{p} + u) \quad (4.47)$$

Fazendo $r = \dot{e} + \lambda e$ tem-se

$$D_{(q)}\dot{r} + \lambda D_{(q)}r + C_{(q,\dot{q})}r = Y_{(q,\dot{q},v,a)}(\tilde{p} + u) \quad (4.48)$$

define-se então a faixa de fronteira

$$\|Y_{(q,\dot{q},v,a)}^T \cdot r\| = \varepsilon \quad (4.49)$$

onde ε é uma constante positiva.

Assumindo-se que o erro entra na faixa num tempo t_c , então a lei de controle pode ser definida como

$$u = -\rho \frac{Y_{(q,\dot{q},v,a)}^T \cdot r}{\|Y_{(q,\dot{q},v,a)}^T \cdot r\|} \quad \text{se } t < t_c \quad (4.50)$$

$$u = -\frac{\rho}{\varepsilon} Y_{(q,\dot{q},v,a)}^T \cdot r - K_I \int_{t_c}^t Y_{(q,\dot{q},v,a)}^T \cdot r \, dt \quad \text{se } t > t_c \quad (4.51)$$

onde K_I é uma matriz diagonal.

Um exemplo de desenvolvimento para um manipulador com dois graus de liberdade é apresentado por Liu [6].

Abdallah [9] dá os seguintes critérios para a escolha do controle robusto a ser utilizado em uma aplicação

- O controle linear multivariável é interessante quando o desempenho do controle é dado em índices do controle linear como razão de amortecimento e porcentagem de sobrepassagem. Pode levar ao uso de ganhos elevados para conseguir diminuir o erro de rastreamento.
- Controles passivos são fáceis de implementar mas não exploram as características físicas das estruturas em cadeia.
- Os controles por estrutura variável, incluindo aqueles por *sliding mode*, não são aconselháveis quando a flexibilidade dos segmentos da estrutura não pode ser desprezada, pois pode excitar dinâmicas de alta frequência.
- Os controles por saturação são mais adequados quando um certo erro transitório é tolerado.

4.11 CONTROLADORES ADAPTATIVOS

Uma outra forma de lidar com os erros de rastreamento de uma trajetória devidos aos erros de modelamento e incertezas paramétricas é o uso de controladores adaptativos. Segundo Moltchakov [23] “controle adaptativo é interessante quando é necessário controlar um objeto nas condições de existência de indeterminações na descrição deste objeto, do ambiente onde o objeto está funcionando ou ainda quando as interações do ambiente com o objeto são pouco conhecidas”. Ou seja, o controle adaptativo é interessante quando:

- o modelo de um sistema não é bem determinado.
- o levantamento do modelo do sistema tem custo proibitivo
- as perturbações que agem no sistema são pouco conhecidas
- os parâmetros do sistema são a princípio desconhecidos
- os parâmetros do sistema variam no tempo segundo uma lei desconhecida.

É neste último caso onde reside a principal vantagem do controle adaptativo que é a de compensar variações nos parâmetros dinâmicos.

O grau de adaptabilidade de um controlador varia de estratégia para estratégia. Em algumas parte-se de um modelo *a priori* do sistema sobre cujos parâmetros agirão as leis de adaptação. Em oposição, há estratégias que dispensam qualquer modelamento anterior, levantando o modelo e seus parâmetros por aprendizado *on line*. O critério de escolha de qual estratégia utilizar está no fato de que quanto maior o grau de adaptabilidade maior será a quantidade de computação em tempo real para aplicação das leis de adaptação. Este fato limita o desempenho do sistema como um todo, já que na aplicação de esquemas adaptativos parte-se do pressuposto de que a frequência de aplicação das leis de adaptação é bem mais rápida do que a dinâmica da planta [11].

Um dos esquemas mais intuitivos de controlador adaptável é apresentado por Craig [1] onde um controlador baseado em modelo tem os parâmetros deste modelo adaptados através de leis de adaptação que têm como entradas os estados da planta e os respectivos erros de rastreo. O projeto de um destes controladores tem como principal problema o fato do estabelecimento das leis de adaptação e seus pesos serem obtidos basicamente por tentativa e erro.

Em [12] são apresentados os controladores adaptativos das figuras 4.15a e 4.15b. No controlador da figura 4.15a usa-se um modelo de referência tipo torque computado e um compensador com uma matriz de ganhos ótimos, \hat{k} , obtida a partir de uma lei que minimiza uma função de custo especificada através de um algoritmo de mínimos quadrados recursivos (RLS). Já no controlador da figura 4.15b, ao invés de um modelo de referência, é utilizado um modelo identificado recursivamente. O modelo identificado recursivamente utiliza um dinâmica linearizada do tipo

$$\tau = Y_{(q,\dot{q})} p \quad (4.52)$$

Resultados obtidos por Goes [12] mostram que o controlador com modelo adaptativo tem melhor desempenho geral do que o não adaptativo mas apresenta oscilações durante o período de aprendizado, o que pode ser minimizado pela adoção de valores iniciais mais próximos da realidade.

Uma outra área de desenvolvimento de controladores adaptativos aplicáveis ao controle de estruturas em cadeia, como os manipuladores robóticos, é a dos controladores baseados em redes neurais. Em [13] é apresentado um estudo comparativo do desempenho de controladores adaptativos a redes neurais com aprendizado *on line* em relação aos que utilizam os parâmetros conhecidos *a priori*, embora com grandes incertezas, para dar partida ao sistema e tornar o aprendizado mais rápido. O controle de um manipulador utilizando redes neurais é apresentado em [14]. Neste caso foram utilizadas duas redes neurais, uma para o cálculo da cinemática inversa e uma segunda para o cálculo da dinâmica

inversa, o fechamento da malha de controle fica por conta de um compensador PI (fig. 4.16). Em [13] é apresentado o controlador híbrido neuro-adaptivo cujo esquema é mostrado na fig. 4.17.

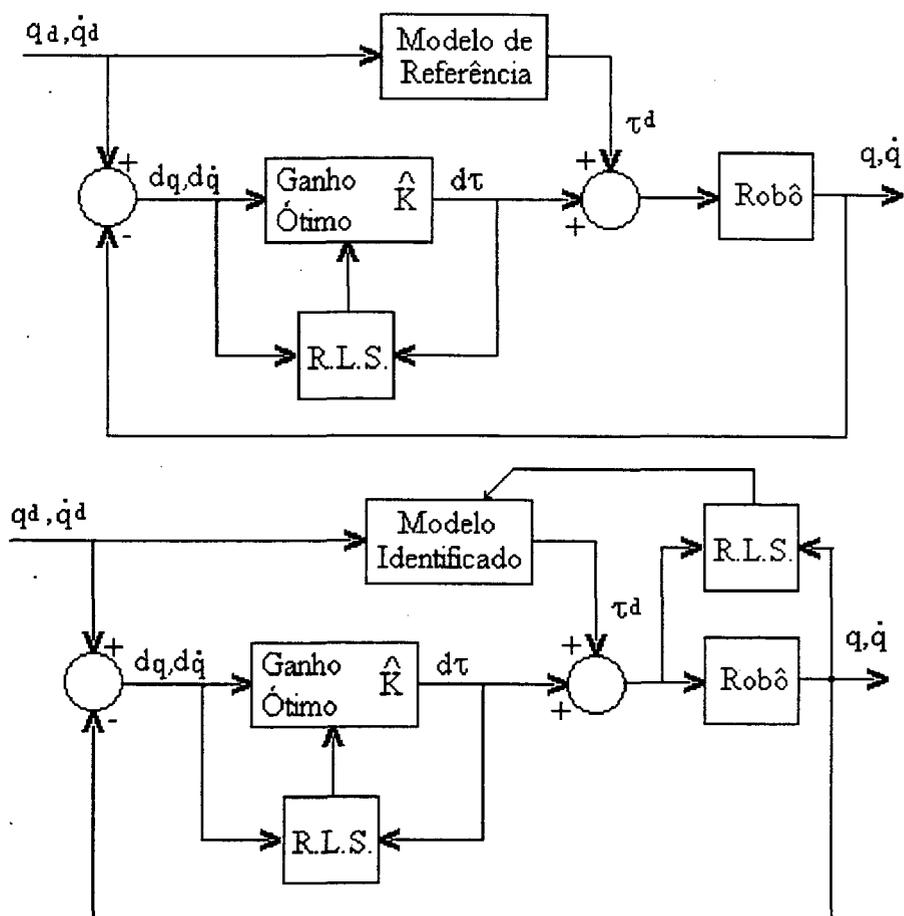


Figura 4.15 Esquema de controle adaptativo baseado em modelo de referência - a, e em modelo identificado - b, propostos por Goes [12].

Outras formas de controle adaptativo podem ser encontradas em [15] que apresenta o modelamento difuso (*fuzzy*) de sistemas dinâmicos não lineares e em [23] que apresenta o controle extremal de objetos inerciais.

No extremo da abordagem de controle adaptativo está o método de projeto de controle não paramétrico. Neste tipo de controle parte-se sem o conhecimento prévio de qualquer característica da planta. O modelo é levantado pelo método da resposta ao impulso onde pulsos são aplicados sequencialmente às entradas do sistema e as suas saídas, neste caso deflexão das juntas, são medidas.

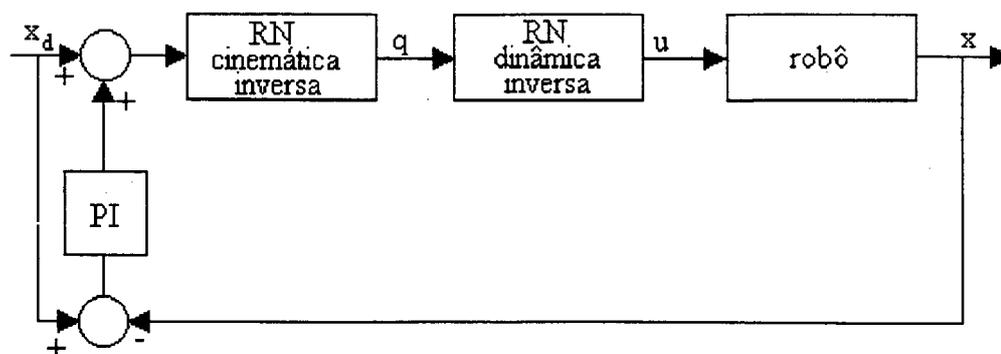


Figura 4.16 Esquema de controlador baseado em RN proposto por [14].

Desta forma, o modelo levantado engloba todos os componentes do sistema desde os *drivers* dos motores até os sensores. Tan [15] apresenta ainda um exemplo de aplicação da lei de controle a um manipulador com dois graus de liberdade, que consiste em cerca de 200 operações de multiplicação e adição e é calculada a intervalos de 10 ms, que pelo peso computacional pode ser implementada até mesmo por um microcomputador do tipo 386 com coprocessador aritmético.

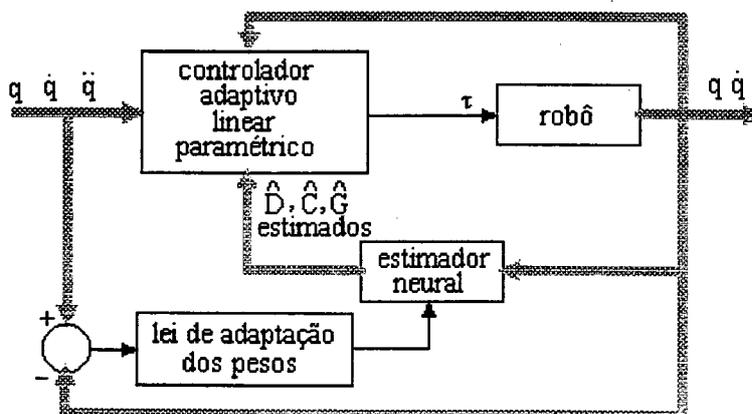


Figura 4.17 Controlador neuro-adaptativo híbrido [13]

4.12 CONTROLE DE ESTRUTURAS FLEXÍVEIS.

Embora na maioria dos casos as estruturas em cadeia, como as que formam os manipuladores robóticos, sejam tratadas e modeladas como um conjunto de corpos rígidos ligados por juntas que

também não apresentam qualquer flexibilidade, na verdade nenhum corpo ou estrutura tem rigidez infinita. Assim, o que determina se uma dada estrutura pode ou não ser modelada como sendo rígida ou não, é, normalmente, o grau de exigência que se faz em termos de erro de rastreo permitido e do tipo de trajetória a ser rastreada.

Estas considerações são relevantes no caso da estrutura do simulador porque, embora do ponto de vista da cinemática o comprimento dos segmentos seja nulo, devido ao fato dos eixos serem mutuamente interceptantes, estruturalmente os segmentos tem dimensões bastante grandes, especialmente os braços do garfo e o anel externo que suporta a cabine. Estas dimensões avantajadas aliadas ao fato de não se poder superdimensionar a estrutura sob pena de prejudicar a sua dinâmica, fazem com que se possa esperar que ela apresente alguma flexibilidade. O tamanho da estrutura também faz com que pequenas deflexões angulares apareçam como deslocamentos bastante grandes nas extremidades. Dessa forma é interessante prever as estratégias de controle que poderiam vir a serem adotadas caso a flexibilidade apresentada pela estrutura venha mesmo a prejudicar seu desempenho como simulador.

O estudo da flexibilidade das estruturas comumente as dividem em dois casos, juntas flexíveis e segmentos flexíveis. As juntas flexíveis transformam um problema de posicionamento de dimensão n em $2n$ uma vez que as posições dos segmentos não são mais solidárias às posições dos atuadores. Também um termo $K(q_a - q_s)$ relativo à energia potencial acumulada na constante de mola k é adicionado à equação do modelo dinâmico da estrutura. Em [3] é feito o modelamento das flexibilidades das juntas e uma lei de controle do tipo PD é sugerida. Brogliato [16] apresenta um estudo comparativo entre vários esquemas de controle para manipuladores com juntas flexíveis.

Segmentos flexíveis transformam o problema de rastreo em um problema com infinitas dimensões devido aos vários modos de flexão que apresentam. Um estudo para controle de trajetórias de manipuladores com segmentos flexíveis usando dinâmica inversa é apresentado por Xi [17] e também por Singh [18]. Boyer [19] apresenta o modelamento simbólico por newton-euler para manipuladores flexíveis. Ainda em [20] são apresentadas soluções de controle para manipuladores com juntas flexíveis e com segmentos flexíveis.

Na planta do simulador os pontos de maior probabilidade de aparecimento de flexibilidades são:

- A junta 3, entre o anel externo e a cabine, devido ao acoplamento por correia dentada além da redução mecânica.

- O segmento formado pelo anel externo, que além de ter grande dimensão, é acionado em apenas dos apoios, suportando esforços de torção ao longo do eixo de rotação que podem chegar a 800 Nm.

A consideração destas flexibilidades nas leis de controle implica na instalação de *encoder* solidário à cabine (detecção de erro de rastreamento na junta) e acelerômetros (detecção de oscilações do segmento) no anel externo. Ambos são de difícil realização mecânica e complicam o processamento do modelo. Inicialmente, deve-se apenas prever faixas de passagem para os compensadores largas o suficiente para que o controle possa corrigir os erros causados pelo acoplamento das oscilações a outros eixos.

4.13 CONTROLE NAS SINGULARIDADES

A estrutura proposta para o simulador tem três graus de liberdade de orientação obtidos pelo uso de três eixos de giro mutuamente interceptantes. Esta disposição à primeira vista permite que sejam simuladas quaisquer atitudes de um móvel livre no espaço tridimensional como um avião em vôo. Isto é verdade para a maior parte das orientações à exceção das duas situações onde os eixos 1 e 3, ligados respectivamente ao garfo e à cabine ficam alinhados. Nestas duas situações pontuais a estrutura perde um dos seus graus de liberdade e fica impedida de simular um dos movimentos de orientação de um corpo livre. O problema é ainda maior porque, embora as condições de singularidade sejam pontuais, no entorno dos pontos singulares a obtenção de pequenas velocidades no grau de liberdade que caminha para a singularidade exige grandes velocidades dos atuadores para serem realizadas. Isto pode ser melhor entendido usando-se o exemplo da escada que desliza em uma parede fig4.18. A medida que a escada desce e se aproxima da singularidade, posição horizontal, maior velocidade é necessária em Y , V_y , para manter a velocidade em X , V_x , até que na posição horizontal a velocidade em Y não é capaz de contribuir para a velocidade em X , mesmo que a primeira seja infinita.

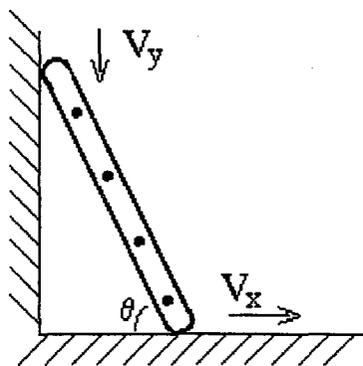


Figura 4.18 A medida que o ângulo θ se aproxima de 0° , maior velocidade é necessária em y para manter a velocidade em x constante.

O problema dos pontos de singularidade não é apenas mecânico-funcional. No ponto de singularidade a matriz jacobiana não admite inversão, o que gera problemas computacionais. Estes empecilhos podem ser facilmente contornados em sistemas onde as trajetórias são previamente definidas. Neste caso basta que a trajetória evite aproximar-se além de uma certa margem de proximidade dos pontos de singularidade. No caso deste trabalho, no entanto, como o projeto do simulador pressupõe interatividade com o piloto, não há como determinar antecipadamente as trajetórias. Também por não haver graus de liberdade redundantes estes não podem ser utilizados para realizar a trajetória especificada em substituição a um grau de liberdade degenerado por um ponto de singularidade.

Chiaverini [21] compara três métodos usados para contornar o problema do movimento em direções degeneradas pela singularidade e os problemas devidos ao defeito no posto da matriz jacobiana. São eles:

- Método da jacobiana transposta
- Método da pseudo-inversa da jacobiana
- Método dos mínimos quadrados amortecido

Este último também abordado por Egeland [22]. Todos buscam um mapeamento alternativo a

$$\tau = J_{(q)}^{-1} \dot{p} \quad (4.53)$$

que é indefinido quando J se torna deficiente.

4.13.1 MÉTODO DA JACOBIANA TRANSPOSTA

A transformação que mapeia os esforços externos aplicados à extremidade de uma cadeia nos torques de juntas é

$$\tau = J_{(q)}^T F \quad (4.54)$$

se o erro de posicionamento da extremidade for escrito como

$$e = p_d - p \quad (4.55)$$

então pode-se escrever o mapeamento

$$\dot{q} = J_{(q)}^T K e \quad (4.56)$$

onde $K e$ equivale a uma força elástica que tende a levar a extremidade para a posição desejada p_d .

Este método é bastante simples e computacionalmente barato mas por ser iterativo só é interessante para posicionamento ou para rastreamento lento, quando pode dar soluções exatas. Para trajetórias de alta velocidade o método pode degradar a precisão do rastreamento. Idealmente K deve ser adaptado em função da configuração.

4.13.2 MÉTODO DA PSEUDO-INVERSA DA JACOBIANA

Este método usa o mapeamento

$$\dot{q} = J_{(q)}^T (J_{(q)} J_{(q)}^T)^{-1} \dot{p} \quad (4.57)$$

ou

$$\dot{q} = J_{(q)}^\dagger \dot{p} \quad (4.58)$$

onde $J_{(q)}^\dagger$ é a pseudo-inversa de $J_{(q)}$.

Este mapeamento é definido mesmo nos pontos de singularidade mas o problema das altas velocidades de junta próximo das singularidades ainda persiste.

4.13.3 MÉTODO DOS MÍNIMOS QUADRADOS AMORTECIDO

Usa o mapeamento

$$\dot{q} = J_{(q)}^T (J_{(q)} J_{(q)}^T + \lambda^2 I)^{-1} \dot{p} \quad (4.59)$$

que satisfaz a condição de minimização

$$\min \left\| \dot{p} - J_{(q)} \dot{q} \right\|^2 + \lambda^2 \left\| \dot{q} \right\|^2 \quad (4.60)$$

Esta abordagem procura comprometer a precisão de rastreamento com a realizabilidade das velocidades de juntas necessárias para gerar as velocidades desejadas na extremidade da cadeia. Este compromisso é conseguido pela adequada escolha de λ . Se λ é pequeno, tem-se soluções precisas mas altas velocidades de juntas, se λ é grande, perde-se em precisão e ganha-se em ter velocidades de junta mais aceitáveis.

A principal desvantagem desta estratégia é o custo computacional, especialmente se se pensa em aplicações de tempo real.

Em [22] é apresentado um exemplo de solução da cinemática inversa por mínimos quadrados amortecido para um pulso esférico, problema semelhante, do ponto de vista da cinemática, ao do simulador.

4.12. Conclusão

Neste capítulo a estrutura do simulador foi caracterizada do ponto de vista do seu controle e foram levantadas estratégias de controle utilizáveis em estruturas segmentadas.

CAPÍTULO 5. PROCESSADORES UTILIZÁVEIS NO CONTROLE

5.1 INTRODUÇÃO

Um dos maiores problemas com que se defronta o projetista de sistemas de controle de movimento é a definição do *hardware* computacional [20]. Há casos, como o relatado em [24], em que toda uma equipe de engenharia de eletrônica embarcada levou mais de dois anos para definir qual plataforma de *hardware* usar em um determinado projeto. Tanto mais crítica será a escolha quanto mais estreitas forem as exigências de desempenho e de relação custo / benefício.

Uma preocupação também sempre presente é a da constante necessidade de atualização e aumento de desempenho dos sistemas bem como a rápida obsolescência do *hardware* disponível à época do projeto. Tais fatos levam à procura de arquiteturas que facilitem a introdução de aperfeiçoamentos e expansões do sistema sem necessidade de mudanças na estrutura do sistema.

Sendo os sistemas de controle de movimento já de natureza multidimensional devido aos vários graus de liberdade das plantas controladas, o espaço de controle tem número de dimensões ainda maior por ter de considerar, além da posição, velocidade e, eventualmente, derivadas de ordem maior em cada uma das direções [7]. Dessa forma, o número de dimensões do espaço de controle pode rapidamente passar a exigir a computação de grandes vetores ou matrizes tornando interessante o uso de processadores capazes de executar instruções sobre tais arranjos de dados.

O fato de que o sistema de controle deve operar em tempo real também é um gerador de restrições ao *hardware* a ser utilizado. Dependendo da tarefa desejada, o requisito de velocidade de processamento pode proibir o uso de certos processadores mais lentos ou mesmo tornar impossível a realização da tarefa por um único processador, no atual estado da arte, passando a requerer o uso de estrutura multiprocessadores[25].

Levanta-se a seguir algumas figuras de mérito que devem ser consideradas no momento da escolha da plataforma de *hardware* que dará suporte ao sistema de controle.

5.2 REQUISITOS DE DESEMPENHO DO HARDWARE

1. Velocidade de Processamento

2. Capacidade de Empilhamento de Interrupções
3. Capacidade de Memória de Programa e Dados
4. Capacidade de Comunicação

CARACTERÍSTICAS DESEJÁVEIS

5. Adequação ao Algoritmo a Ser Processado
6. Dispositivos Específicos para Interface
7. Suporte de Desenvolvimento
8. Facilidade de Gerenciamento
9. Custo e Confiabilidade

5.2.1. VELOCIDADE DE PROCESSAMENTO

A velocidade de processamento de um *hardware* não é ditada apenas pelo número de instruções executadas por unidade de tempo, pois além da efetividade desta medida depender do que faz uma instrução em cada tipo de arquitetura, há que se considerar também, principalmente quando se trata de um sistema de controle, os tempos gastos no atendimento a interrupções,

mudanças de contexto de operação, comunicação, bem como o tempo despendido para o enchimento e esvaziamento de possíveis *pipelines* existentes na arquitetura [26]. Dessa forma, o real desempenho em termos de velocidade de processamento de um dado *hardware* só é corretamente avaliado quando executando a tarefa para a qual é pretendido ou *benchmarks* adequadamente escolhidos [27].

A velocidade de processamento deve ser definida em função dos quesitos:

- Complexidade do algoritmo da lei de controle e frequência com que esta deve ser aplicada à planta. Estas, por sua vez serão definidas pelo modelo matemático e pelo grau de sujeição que se deseja imprimir à planta, respectivamente. De forma resumida pode-se dizer que a velocidade de processamento está diretamente ligada à faixa passante (*bandwidth*) do sistema de controle.

- Uma maior velocidade de processamento também pode permitir uma migração das implementações do hardware para o software, o que pode gerar economia nos custos da etapa de produção.

- Parâmetros físicos do sistema de controle ou da planta. No caso do uso de acionadores comandados por PWM, há a limitação de velocidade imposta pelo tempo de chaveamento dos dispositivos CMOS de potência, atualmente por volta dos 50 μ s [28].

5.2.2. CAPACIDADE DE TRATAMENTO DE INTERRUPÇÕES

Uma das formas usuais de se implementar a comunicação entre processadores e entre estes e os dispositivos de entrada / saída é pelo uso de interrupções. Dependendo da filosofia adotada, a incidência de interrupções pode ser bastante grande. Nestes casos, o uso de dispositivos com pilhas muito curtas, como alguns DSPs que possuem apenas quatro posições, dificulta o aninhamento de interrupções, que só poderá ser ampliado a custa de artifícios de software, com prejuízos para a velocidade do sistema. Importante também é o tempo demandado pela ar-

quietura em questão para efetuar uma mudança de contexto, que se muito longo prejudica, como já visto, a velocidade de processamento do sistema.

Dependendo de como se deseje implementar o *hardware* periférico, pode ser importante a observação do número de linhas externas de interrupção disponíveis para implementações pelo usuário.

5.2.3 CAPACIDADE DE MEMÓRIA DE PROGRAMA E DADOS

A necessidade de memória de programa e dados será função, basicamente, do tamanho do algoritmo a ser processado e do número de constantes e variáveis necessárias ao processamento.

Dependendo do processador empregado, o tamanho de memória de programa necessária é ainda dependente do *set* de instruções. Assim, processadores com um *set* de instruções mais restrito tendem a tornar necessária uma maior quantidade de memória de programa[24].

Em aplicações onde é requerida alta velocidade de processamento, torna-se importante levar em conta o tamanho da memória *cache* disponível no *chip* ou na placa, pois sendo uma memória de alta velocidade de acesso permite à *CPU* buscar instruções e variáveis com menor atraso no processamento. Uma outra implicação da quantidade de memória *cache* disponível é a possibilidade de implementar os bancos de memória externos com dispositivos com maior tempo de acesso, logo, mais baratos.

5.2.4 TAMANHO DA PALAVRA

O tamanho da palavra é ditado pela faixa dinâmica dos sinais a serem manipulados pelo controlador, especialmente nos processadores que executam aritmética de ponto fixo. Palavras com maior número de *bits* possibilitam trabalhar com maiores resoluções sem custo adicional no tempo de processamento e diminuem o efeito indesejado do liga-desliga do *bit* menos significativo.

5.2.5 CAPACIDADE DE COMUNICAÇÃO

A importância da capacidade de comunicação dependerá da intensidade da troca de informações entre os vários elementos do sistema que, por sua vez é função do grau de *frouxidão*

(looseness) da arquitetura geral do sistema e da sua hierarquização[25]. Pode, então, ser relevante a disponibilidade de canais de comunicação paralelos ou seriais, dependendo do caso, por parte dos vários elementos processadores do sistema, de forma a permitir comunicação eficiente sem a necessidade do uso de lógicas de adaptação adicionais (glue logic).

É interessante que a arquitetura do *hardware* disponha de características que tornem o gerenciamento da comunicação o mais transparente possível ao contexto do processamento principal, de modo que não provoque degeneração no desempenho computacional.

5.2.6 ADEQUAÇÃO AO ALGORITMO A SER PROCESSADO

É considerada uma tendência atual das filosofias de projeto de arquitetura de sistemas, fazer a arquitetura do hardware ter estrutura assemelhada àquela do software e da própria forma dos dados[29]. Assim, a distribuição de inteligência pelos vários níveis do sistema deverá ser função da complexidade das tarefas em que o problema como um todo foi dividido, para aquele nível, bem como ter estrutura adequada ao tratamento dos dados na forma em que se apresentam naquele nível. Por exemplo, ao nível de controle de acionadores pode ser interessante o uso de microcontroladores com características específicas para tal, enquanto que em um nível superior, onde se requer tratamento numérico de sinais, o dispositivo ideal pode ser um processador digital de sinais - DSP.

Também do ponto de vista horizontal, deve-se procurar uma arquitetura que se adeque ao problema, caso haja, por exemplo, a necessidade de processamento em ramos paralelos de um algoritmo simultaneamente, a arquitetura deve prever esta possibilidade, ou pelo menos serem escolhidos processadores que facilitem a sua implementação.

5.2.7 DISPOSITIVOS ESPECÍFICOS PARA INTERFACE

Alguns processadores, em especial aqueles desenvolvidos para uso como controladores, podem incorporar blocos como temporizadores, conversores A/D e D/A, moduladores PWM e decodificador para encoder, entre outros, que facilitam sobremaneira a implementação da malha de controle de acionadores, o que os torna bastante atrativos para uso em controle de movimento.

5.2.8 SUPORTE AO DESENVOLVIMENTO

Deve-se também levar em conta a disponibilidade de certas facilidades de desenvolvimento como kits para avaliação de desempenho, softwares para *assemblagem*, compilação e simulação, bem como literatura de aplicação fornecida pelo próprio fabricante, além da *amigabilidade* de cada sistema de desenvolvimento. Dependendo do nível e da escala de aplicação que

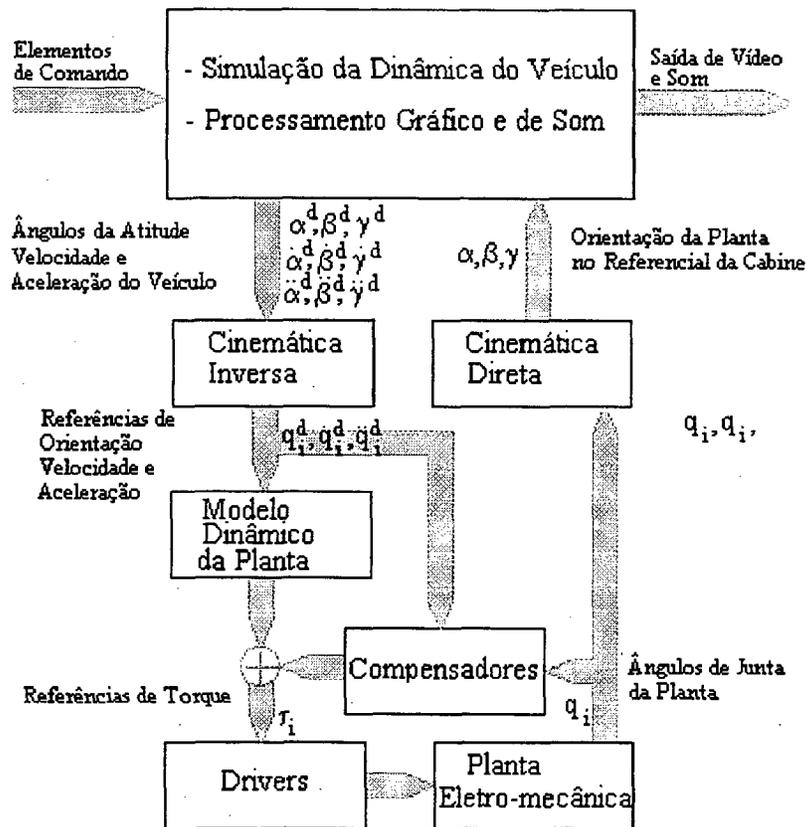


Fig. 5.1 Esquema de controle mostrando as necessidades de processamento

o *hardware* em questão virá a ter, pode ser interessante considerar o emprego de placas comerciais contendo os processadores e periféricos necessários a dada aplicação. Em muitos casos as considerações devem ser feitas em termos de características de placas ao invés de características de *chips* processadores[30] sobretudo quando a quantidade a ser produzida não justifica o desenvolvimento de uma placa própria com todos os custos e tempos envolvidos.

5.2.9 FACILIDADE DE GERENCIAMENTO

Arquiteturas mais complexas requerem, para a operacionalização de todo o seu potencial, de um gerenciamento eficiente de todos os seus recursos. Caso não se possa contar com um compilador com inteligência suficiente para fazer ele próprio o gerenciamento dos recursos do hardware, esta tarefa será bastante penosa para a engenharia de desenvolvimento.

5.2.10 CUSTO E CONFIABILIDADE

Estes dois fatores são estabelecidos por critérios que podem se antagonizar. O custo por exemplo, sendo feito o mais baixo possível durante o desenvolvimento, pode levar a custos maiores durante toda a fase de produção. Também é conveniente procurar a divisão mais razoável entre os custos do *hardware* e do *software*, é normal que um maior investimento inicial na elaboração dos programas leve a uma economia que se estenderá por todo o tempo de produção e mesmo durante a utilização do produto.

A confiabilidade, por sua vez, oscila entre a adoção de tecnologias recém estabelecidas e aquelas de desempenho comprovado por vasta utilização. Decisões como esta, e de que outras precauções adotar, como o grau de redundância do sistema -por exemplo, devem ser feitas em função do custo e do nível de segurança requerido.

5.3 ALGUMAS CONSIDERAÇÕES SOBRE A ARQUITETURA DO *HARDWARE* DE CONTROLE

Duas características fundamentam as filosofias das arquiteturas avançadas de controle atualmente: Elas são abertas e hierarquizadas. As arquiteturas abertas tem como característica a modularidade, possibilitando a incorporação de módulos de diferentes fabricantes de modo a permitir a integração de programas de aplicação, sensores e processadores sem que o sistema tenha

de ser redefinido. Em [31] é descrito um *hardware* de controle com estas características para aplicação em uma máquina ferramenta CNC. Já a hierarquização é a característica das arquiteturas onde a inteligência do sistema, e por conseguinte, o processamento, está distribuído pelos vários níveis de controle, nos quais podem ser usadas diferentes técnicas[32]. Sob esta filosofia todo o sistema coopera para a execução de uma tarefa que, sendo requerida no nível mais alto da hierarquia, vai sendo subdividida em sub-tarefas até se transformarem em simples comandos a serem aplicados a atuadores. Cada um dos níveis tem a seguinte seqüência lógica de ações [32]:

1. Receber a tarefa dada pelo nível mais elevado
2. Ler dados dos sensores e estimar os modelos relevantes
3. Realizar a sub-tarefa correspondente usando os dados disponíveis
4. Repassar as ações a serem executadas em níveis subordinados
5. Monitorar as respostas e em caso de necessidade, revisar a estratégia
6. Finalmente dar resposta ao nível superior.

Em um sistema hierarquizado de controle de um robô pode-se distinguir os seguintes níveis[32]:

1. Nível de tarefa: Transforma uma tarefa completa do robô em séries de movimentos elementares
2. Movimentos elementares: Transforma os comandos vindos do nível de tarefa em seqüências de poses intermediárias
3. Primitivas: Responsável pela geração de trajetórias suaves, eficientes e temporalmente sincronizadas entre as poses impostas pelo nível anterior
4. Nível de servo controle: Gera os sinais para os atuadores de acordo com as trajetórias computadas pelo nível de primitivas.

O uso de uma arquitetura hierarquizada em controle possibilita, entre outras, as seguintes vantagens:

- Fácil adaptação a outras plantas
- Maior transparência do *hardware* de controle do ponto de vista do usuário, o que torna o sistema mais amigável e facilita a execução de diferentes programas de aplicação

- Permite a utilização, em cada nível, da solução a ele mais adequada tanto do ponto de vista da técnica quanto da capacidade computacional
- Como o processamento encontra-se distribuído pelos vários níveis, a capacidade de processamento total é aumentada, o que é desejável em sistemas que operam em tempo real.

5.4. PROCESSADORES UTILIZÁVEIS EM CONTROLE DE MOVIMENTO

“Qualquer processador é capaz de executar qualquer tarefa desde que lhe seja dado tempo suficiente. No entanto, diferenças na arquitetura dos processadores ditarão quão fácil será programar tal tarefa e quão rapidamente ela será executada” [27]. Em se tratando de controle de movimento, no entanto, é crucial que o processamento seja feito em tempo real, o que cria severas restrições para a capacidade dos processadores empregados.

Em algumas aplicações como um simulador de vôo multidimensional, por exemplo [25], a capacidade computacional exigida para a operação em tempo real é de tal ordem que impossibilita sua realização pelo emprego de um único processador, mesmo o mais rápido do atual estado da arte. Há, no entanto, uma faixa de aplicações para as quais a decisão do uso de arquitetura mono ou multiprocessadores é assunto para uma análise atenciosa. Alguns critérios podem então ser considerados:

“O usuário estará sempre melhor servido por uma única *CPU* rápida do que por um arranjo de *CPUs* mais lentas, pois além de ser mais simples evita o desperdício de tempo com a comunicação entre processadores”[26].

Uma *CPU* única também pode ser mais barata do que o conjunto que ela substitui, quando, por exemplo se utiliza um processador *RISC* ao invés de vários *CISC*, mesmo considerando os custos adicionais com programação e periféricos.

A complexidade do projeto de sistemas multiprocessados também é um fator que não deve ser negligenciado, bem como a distribuição das tarefas pelos vários processadores, quer arranjados em *pipeline*, quer de forma paralela [33].

O multiprocessamento pode se tornar atrativo também por motivos outros que não a necessidade de incremento da capacidade computacional. É o que ocorre, por exemplo, em aplicações como a robótica onde pode ser preferível a alocação de um processador próximo a cada junta comunicando-se serialmente, do que ter de passar os vários sinais de sensores e atuadores fisicamente através das juntas[33].

5.4.1 *HARDWARE* MONOPROCESSADOR

i) Microcontroladores

São processadores com um *set* de instruções que pode facilmente ultrapassar uma centena delas, construído com uma arquitetura de barramento único e que privilegia as interfaces de entrada e saída. Devido a este tipo de construção, os microcontroladores, por mais rápidos que sejam, tem limitações de aplicação onde é requerida alta velocidade de processamento. Isto porque gastam boa parte do tempo gerenciando o sistema de *I/O*, restando pouco tempo de manipulação efetiva de dados. Para evitar computações mais pesadas uma técnica freqüentemente utilizada nos sistemas microcontrolados consiste nas *lookup tables* que têm o inconveniente de ocuparem muita memória e restringirem o número de variáveis do algoritmo. Um outro ponto fraco dos microcontroladores é o fato de suas *ALUs* executarem multiplicações pelo processo de adições sucessivas. Tais limitações do *hardware* resultam em uma redução da taxa de amostragem e por conseguinte da faixa de passagem *bandwidth* [34].

Um outro problema que freqüentemente aparece em sistemas de controle a microprocessadores é o basculamento permanente do bit menos significativo (*cycling*) devido ao tamanho pequeno de suas palavras e a aritmética de ponto fixo que utilizam.

Não obstante tais limitações, os microcontroladores tem seu nicho de utilização garantido em situações onde não são requeridas alta velocidade de processamento nem processamento de algoritmos complexos. Estas situações são freqüentemente encontradas em pequenas aplicações ou nos níveis mais baixos de arquiteturas de controle hierarquizadas onde é exigido pouco trabalho computacional e os periféricos característicos de um microcontrolador são bastante atraentes.

O uso de microcontroladores também é atrativo pela simplicidade da sua arquitetura que acaba estendendo-se ao *software* e à construção do restante do sistema ainda mais facilitados pela existência de auxílios ao desenvolvimento como simuladores, emuladores e placas para protótipos para quase todas as famílias, bem como versões para desenvolvimento com *EPROM on chip* em alguns casos.

O principal atrativo dos microprocessadores, na verdade, reside na sua facilidade de interfaceamento devido a multiplicidade de dispositivos de *I/O* e a existência de periféricos específicos para controle como conversores *A/D*, contadores, temporizadores e geradores de sinal *PWM*.

Como exemplo de microcontrolador será tomado o *chip* 80C196KC da família de microcontroladores *embedded* de 16 *bits* da Intel, do qual serão evidenciadas algumas características a seguir.

Microcontrolador 80C196KC

É um dispositivo de tecnologia CHMOS que opera com palavra de 16 *bits* e *clock* de até 16 M Hz e está baseado em uma *CPU* com operação *register to register* sobre uma pilha de registradores de 232 *bytes* cujos registradores podem funcionar indistintamente como acumuladores. A estes registradores podem ser somados outros 256 *bytes* em *RAM* mapeados pela técnica de *register windowing*, do que resultam mudanças de contexto mais rápidas pois há menos conteúdos a serem salvos.

Memória

O microcontrolador 80C196KC pode endereçar até 64 *kbytes* de memória sendo que a maior parte pode ser utilizada pelo usuário.

Operandos

O 80C196KC pode operar sobre os seguintes tipo de dados:

- * *bytes*: variáveis de 8 *bit* sem sinal com valores entre 0 e 255
- * *palavras*: variáveis de 16 *bit* sem sinal com valores entre 0 e 65535
- * *inteiros curtos*: variáveis de 8 *bit* com sinal com valores entre -128 e +127
- * *inteiros*: variáveis de 16 *bit* com sinal com valores entre -32768 e +32767
- * *bits*: operandos de um único *bit* que pode assumir os valores booleanos 0 e 1
- * *palavras duplas*: variáveis de 32 *bit* sem sinal com valores entre 0 e 4.294.967.295
- * *inteiros longos*: variáveis de 32 *bit* com sinal com valores entre -2.147.483.648 e +2.147.483.647.

Periféricos

Saída *PWM* (D/A)

O 80C196KC possui 3 saídas *PWM*. A saída *PWM* é uma forma de onda com razão cíclica variável com período selecionável de 256 ou 512 ciclos de estado do *clock* que devidamente modulada pode acionar diferentes tipos de servo motores. Também pode ser utilizada para conver-

são D/A pela produção de um nível *DC* obtido por filtragem, com a resolução da modulação que é de 256 degraus. A modulação é feita escrevendo-se um valor de oito *bit* nos registradores *PWM*.

Temporizadores

No 80C196KC estão disponíveis dois temporizadores de 16 *bit*. O *TIMER1* incrementa a cada oito ciclos do *clock*, enquanto que o *TIMER2* está disponível para contagens externas e pode ser configurado para diversos tipos de contagens.

Entradas e Saídas de Alta Velocidade

Quatro entradas cujo tipo de sensibilidade pode ser configurado independentemente, ao serem ativadas fazem com que o conteúdo do *TIMER1* seja salvo juntamente com quatro *bits* que indicam qual entrada provocou a ativação. As saídas de alta velocidade podem gerar eventos de 15 tipos diferentes em valores especificados de *TIMER1* ou *TIMER2* com um mínimo de perda de tempo.

Porta Serial

Opera em um modo síncrono e três assíncronos, estes em *full duplex*. Há *buffers* para 2 *bytes* em ambos os sentidos. A taxa de transmissão é estabelecida por contador específicos de 15 *bit*, e pode chegar a 4,0 Mbaud com clock de 16 M Hz.

Conversor A/D

O bloco do conversor A/D no 80C196KC consiste de um *sample-and-hold*, um multiplexador de oito canais e um conversor por aproximações sucessivas de 10 *bit*.

Portas de Entrada/Saída

As entradas e saídas do 80C196KC estão organizadas em 5 portas de 8 *bit*, algumas só de entrada, algumas só de saída e outras com funções múltiplas. A porta 0, por exemplo é uma porta de entrada e, ao mesmo tempo, entrada analógica para o conversor A/D [35].

ii) Processador Digital de Sinais (*DSP*)

Os *DSPs* são processadores que ao invés da arquitetura de Von Neumann, que armazena dados e instruções sequencialmente em uma mesma memória, utiliza arquitetura Harvard, onde dados e instruções são armazenados em diferentes memórias e são transportados também por diferentes barramentos, um para dados outro para instruções. Disto resulta ganho de tempo computacional pois pode haver busca simultânea de dado e instrução além de não ser mais necessário identificar o que é dado e o que é instrução. Toda a concepção do *DSP* é voltada para a alta velocidade de processamento necessária ao tratamento de sinais em tempo real, por isso a arquitetura interna é estruturada de modo a executar a maioria das instruções em um único ciclo de instrução (dois ciclos de *clock*) o que só é possível porque a maioria das instruções são implementadas em *hardware* não acarretando, portanto, em dispêndio de tempo na execução de microcódigos [27]. O *set* de instruções é bastante restrito, principalmente no tocante a modos de endereçamento. A ênfase é dada às instruções de multiplicação/acumulação (*MACs*) que estão para o processamento digital de sinais assim como os amplificadores operacionais estão para o processamento analógico[34]. Dessa forma, os *DSPs* podem computar algoritmos de controle avançados como controle adaptativo, filtros de Kalman e controladores de estado em tempo real, aparecendo como opções naturais em aplicações onde os microcontroladores e outros processadores convencionais não apresentam capacidade computacional suficiente [36].

Em aplicações mecânicas podem ser implementados *notch filters* capazes, por exemplo, de atenuar ressonâncias com alta seletividade [34], ou efetuar todo o controle de um motor de automóvel [36] com computação de algoritmos em tempo real ao invés soluções aproximadas armazenadas em *lookup tables*. Os *DSPs* também abriram espaço para o controle *single chip* de sistemas multieixos em tempo real. Dessa forma os controles puderam avançar do domínio analógico para o totalmente digital - os microprocessadores são, na verdade uma solução de meio termo - uma vez que fazem uma abordagem numérica do problema utilizando suas *MACs* para implementar pólos e zeros[34], [27].

Por outro lado, em oposição ao poderio computacional os *DSPs* tendem a ser pobres em recursos para interfaceamento e exigem trabalho mais extensivo em *software* se comparados, por exemplo, aos microcontroladores. Esta última característica se bem explorada pode, no entanto, ser interessante do ponto de vista de custos, já que tendo o *DSP* velocidade suficiente, elementos antes construídos em *hardware* podem ser implementados em *software* com vantagens adicionais de estabilidade, imunidade a ruídos e flexibilidade (adaptabilidade).

Estando os *DSPs* apresentando-se como solução de processamento em sistemas de controle, torna-se praticamente inevitável sua comparação com os microcontroladores e microprocessadores convencionais.

Em termos de velocidade de processamento um *DSP* pode processar o algoritmo de um filtro *FIR* de ordem 64 mil vezes mais rápido do que um processador de 16 *bit* com coprocessador matemático [37]. Em [34] é apresentada a tabela 1 que compara os desempenhos de um *DSP* da Texas, um microcontrolador Intel e um microprocessador *CISC* da Motorola, usando como *benchmarks* alguns algoritmos importantes em controle.

<i>Benchmark</i>	320C14 @ 26 MHz	80C196KB @ 12 MHz	68000 @ 16 MHz
PID ordem 3	3,2	27,0	17,0
Polinomial ordem 4	4,6	29,7	45,4
Multiplicação P.F.	5,2	19,8	25,6
Busca de pivô	23,3	51,5	31,0
Hex para decimal	91,3	90,8	113,0
Int. <i>overhead</i> (máx)	7,0	14,8	15,1
Int. <i>overhead</i> (mín)	1,6	6,7	4,0

Tab 5.1 Tempos de execução (μ s)

Em termos de capacidade de memória e movimentação de grandes quantidades de dados os microcontroladores levam vantagem, uma vez que os *DSPs* dispositivos de operação de bases de dados. Seu *set* de instruções não privilegia operações de carga e armazenamento, ao contrário são estruturados para operarem com fluxo permanente de dados (*datastream*), típico do processamento em tempo real.

Facilidade de programação é um outro ponto em que os microcontroladores vencem. O desenvolvimento de *software* é notadamente mais fácil e demanda menos especialização nos microprocessadores, o que é especialmente importante em cronogramas de trabalho apertados.

O custo pode ser favorável aos *DSPs* se suas características forem exploradas agressivamente, substituindo *hardware* por *software* [27],[34],[36].

As implementações com *DSPs* podem se dar de três formas básicas: usando *DSPs single-chip* de uso geral, usando blocos fornecidos pelos fabricantes na forma de *sets* configuráveis,

ou usando dispositivos customizados ou semicustomizados disponibilizados por alguns fabricantes.

Pode-se também escolher entre dispositivos com aritmética de ponto-fixo ou de ponto flutuante. Os *DSPs* de ponto fixo são mais rápidos e mais baratos mas podem vir a apresentar problemas com a faixa dinâmica. Neste sentido alguns *DSPs* de ponto-fixo possuem um registrador circular (*barrel register*) que tem por finalidade girar uma palavra à esquerda ou à direita mudando seu peso por potências inteiras de dois (*scalling*) aumentando a faixa dinâmica sem perda maior de velocidade.

A seguir dá-se como exemplo algumas características do *DSP* 56000 da Motorola.

O 56000 é um *DSP* de ponto-fixo com palavra de 24 *bit* cuja unidade central é composta de uma *MAC* que realiza instruções do tipo $24 \times 24 \pm 56$ e de giro, e de dois acumuladores de 56 *bit* cada um. A memória interna está organizada em um bloco de memória de programa e dois blocos de memória de dados (X e Y) acessadas por barramentos específicos, o que permite buscas simultâneas de uma instrução e dois dados. Há ainda toda uma série de controladores, multiplexadores e chaveadores de barramentos. A frequência de *clock* admitida varia de versão para versão podendo alcançar 100 MHz.

Memória

As memórias internas têm capacidade de 256 palavras cada - memórias de dados- e 512 palavras - memória de programa. Externamente o 56000 pode endereçar 128 kpalavras de dados e 64 kpalavras de programa.

Operandos

Palavra: variáveis fracionárias com sinal, com comprimento de 24 *bit*, cujo valor pode variar de $1-2^{*-23}$ (maior valor), a -1. Faixa dinâmica de 144 dB.

Palavra longa: variáveis fracionárias com sinal, com comprimento de 48 *bit*, cujo valor pode variar de $1-2^{*-47}$ (maior valor), a -1. Faixa dinâmica de 298 dB.

Acumulador: variável de 56 *bit* sendo um para o sinal, oito para a parte inteira e os demais para a parte fracionária cujo valor pode variar de $256-2^{*-47}$ (maior valor), a $-256-2^{*-47}$. Faixa dinâmica de 336 dB.

Periféricos

Host Interface

É uma interface em oito *bit* com suporte para *DMA* que oferece uma forma simples de conectar o 56000 com outro *DSP*, *hardwares* de *DMA* ou qualquer outro processador .

Interface de Comunicação Serial

É uma interface que permite comunicação serial assíncrona *full duplex* com outros processadores usando protocolos *UART*. Também pode operar em modo síncrono com *baudrate* estabelecido por gerador programável .

Interface Serial Síncrona

Opera em *full duplex* permitindo comunicação serial síncrona com dispositivos de padrão codec, *DSPs*, microprocessadores e periféricos seriais. Possui modo rede usado tipicamente para multiplexação de tempo em redes estrela ou anel.

Temporizador / Contador de Eventos.

É um contador de 24 *bit* com resolução máxima de metade da frequência do *clock* que pode gerar sinalização externa a partir da contagem de eventos internos ou gerar interrupções pela contagem de eventos externos[38, 39].

iii) Transputer

O transputer é um processador *RISC* com uma arquitetura concebida segundo uma filosofia que procura privilegiar características como alta velocidade de processamento, processamento paralelo e multiprocessamento. A alta velocidade é obtida pelo uso de instruções de um ciclo, formação de um *pipeline* por alguns blocos internos do processador que permite a execução simultânea de várias instruções e pelo pequeno tempo de mudança de contexto, resultado do reduzido número de registros usados, a serem salvos. O processamento paralelo é facilitado pelo uso de canais de comunicação entre os quatro níveis de processamento. No caso de multiprocessamento, vários transputers podem ser interligados diretamente através de quatro *links* seriais de alta velocidade. A comunicação tanto entre contextos executando em um mesmo processador como em processadores diferentes é estabelecida da mesma forma através dos canais, existam

links físicos entre estes processadores ou não, o que facilita bastante a programação. A comunicação entre transputers não adjacentes em uma rede ocorre através dos *links* dos transputers intermediários mas de forma quase transparente a cada processador já que cada um deles conta com um processador interno específico para o controle da comunicação.

Uma outra característica dos transputers é o fato de terem sido desenvolvidos para serem programados através de linguagens de alto nível, contando com um *set* de instruções que levou isto em conta ao ser implementado. Entre outras possibilidades pode ser usada a linguagem OCCAM que procura otimizar a execução dos programas em função do *hardware* disponível.

Internamente o transputer também conta com um *hardware* agrupador de instruções que agrupa as instruções de forma a serem executadas tirando o maior desempenho possível da arquitetura.

Outras características interessantes dos transputers é a necessidade de *clock* externo de apenas 5 MHz, elevado depois internamente e a disponibilidade de dispositivos de interligação de processadores em forma de matrizes *crossbar*, quando as características de conectividade do transputer não for por si só suficiente para o grau de multiprocessamento desejado.

Em aplicações de controle além da evidente vantagem devida à alta velocidade de processamento, os transputers oferecem, pela sua concepção, a possibilidade de cuidar simultaneamente vários aspectos do controle da planta. Num automóvel, por exemplo diferentes níveis de processamento poderiam cuidar respectivamente do sistema de ignição, de injeção de combustível e freios. Já sua alta capacidade de comunicação serial possibilita a diminuição da cabeleagem entre sensores, atuadores e elementos de processamento pela colocação estratégica de transputers no sistema de controle, o que evita além dos inconvenientes como montagem de chicotes e sua movimentação, também problemas de natureza elétrica como *crosstalk*, deformação de sinais e indução de ruídos. Caso típico são os robôs, onde os sinais precisam chegar a cada *link* passando através das juntas móveis.

A seguir serão dadas algumas características do transputer IMS9000 da InMOS.

Transputer IMS9000

Fisicamente o IMS9000 é construído sobre uma pastilha de 180 mm**2 contendo cerca de 2 milhões de transistores - o dobro de um IBM 486 [29]. Em termos de blocos funcionais tem-se um *pipeline* - formado pela *cache* de trabalho, dois geradores de endereços uma unidade lógica e aritmética, operando com inteiros de 32 *bit*, e uma unidade de ponto flutuante de 64 *bit* -, uma *cache* de instruções e dados de 16 *kbyte*, uma interface de memória programável, além

de vários periféricos, tudo interligado por uma estrutura *crossbar* de 32 *bit*. As operações são executadas sobre uma pilha de três registros - Areg, Breg e Creg - na CPU e outros três registros de 64 *bit* - FAreg, FBreg e FCreg.

Memória

O IMS900 pode endereçar até 4 *Gbyte* de memória diretamente com suporte total para uso de *DRAMS* e podendo inclusive misturar memórias com diferentes comprimentos de palavra. Um bloco chamado interface programável de memória é o responsável pela flexibilidade e também pelo correto compartilhamento da memória entre os processos que estão sendo executados simultaneamente.

Periféricos

Processador de canal virtual: responsável pela comunicação entre os processos que estão sendo executados em um mesmo transputer, ou diferentes unidades cuidando de fazer o *bypass* das informações vindas de um processador com destino a um terceiro com o qual não tem um *link* físico de forma transparente a execução dos processos.

Links Seriais: em número de quatro, fazem comunicação entre transputers em taxas de até 80 *Mbyte/Seg* bidirecionalmente sem necessidade de qualquer lógica externa.

Canal de Eventos: contém quatro linhas que desempenham papel semelhante ao das interrupções nos outros processadores. São usados também para sincronizar processos.

Links de Controle: Os dois *links* de controle funcionam com protocolos semelhantes aos dos *links* de dados. Um deles recebe comandos e envia informações de *status* enquanto que o outro provê uma saída para cascadeamento que possibilita a formação de sistemas em *dayse-chain*.

Timer: O IMS9000 conta com dois *timers* de 32 *bit*, um para o processo de alta prioridade e o outro para o processo de baixa prioridade. Ambos contam eventos internos. O primeiro completa um ciclo em aproximadamente 4295 segundos, o outro em cerca de 76 horas. Também ambos trabalham associados a dois registradores. Um dos registradores armazena o valor da

contagem de estouro, o outro um endereço do espaço de trabalho onde haverá o desvio do programa.

5.4 PROCESSADORES CISC E RISC

Uma tendência observada no desenvolvimento dos microprocessadores foi a de dotá-los de *sets* de instruções cada vez mais poderosos, ou seja, com instruções que desencadeiam a execução de verdadeiros microprogramas residentes em *ROM*. Disto resultam vantagens como: Uma aproximação da linguagem de programação do microprocessador de linguagens de nível mais alto, diminuição do trabalho de programação, necessidade de menos memória. Apoiada nestes pontos a filosofia *CISC* (*Complex Instruction Set Computer*) foi dominante até poucos anos atrás.

Com o avanço da tecnologia de semicondutores que diminuiu o custo das memórias e permitiu aumento de ordens de grandeza na escala de integração e frequência do *clock*, a economia de memória se tornou menos importante ao passo que a possibilidade aberta de se ter todo o processador em um único *chip*, evitando atrasos nos barramentos externos, passou a ser valorizada. Também do ponto de vista de utilização chegou-se à conclusão de que as instruções mais utilizadas são as mais simples.

Dessa forma, entrou em evidência a arquitetura *RISC* que assim se caracteriza:

- Instruções simples realizadas em um único ciclo de máquina, logo, rápidas
- Grande incidência de operações registro-registro, dessa forma muitos operandos em *cache*.
- Modos de endereçamento simples; os mais complexos podem ser implementados em *software*
- Instruções de formato simples com tamanho limitado ao da palavra, o que possibilita busca e decodificação simultâneas de operandos.

Destas características resultam outras vantagens como a possibilidade de formação de *pipelines*, possibilidade de se criarem compiladores capazes de maior otimização na geração dos códigos de máquina, resposta mais eficiente a interrupções [29]. Tais vantagens ficam ainda mais evidentes em tarefas de controle onde normalmente não se utilizam modos rebuscados de endereçamento ou funções matemáticas complexas, principalmente quando feita abordagem digital, e onde atendimento eficiente a interrupções pode ser crucial. Do ponto de vista de custos

para sistemas *embedded* a solução *RISC* promete um desempenho de velocidade de duas a cinco vezes o da *CISC*, para um mesmo custo [24].

Por tudo isto é que se o mercado de placas até o início dos anos 90 era dominado por produtos baseados em processadores *CISC*, em especial o 68000 da Motorola, atualmente a tendência é a de usar o Intel i960 ou 29000 da AMD em aplicações de pequena a média exigências e o R3000 da MIPS ou Sparc da Sun em aplicações militares ou de alta performance[30].

A escolha entre as duas filosofias, entretanto não está definida já que, quando amigabilidade, facilidade e tempo de desenvolvimento estão em jogo os processadores *CISC* ainda podem ser os mais indicados. Em seguida apresentamos algumas características dos processadores i960 (*RISC*) e 68000 (*CISC*).

O Processador *Embedded RISC* 80960CA

O processador 80960CA é uma das versões construídas em torno da arquitetura i960 com opções de *clock* de 16, 25, 33 e 40 MHz. Com *clock* de 33 MHz o processador é capaz de sustentar uma velocidade de processamento de 66 MIPS. O corpo central do 80960CA é um *pipeline* formando um módulo de pré busca de instruções, um *cache* de instruções de 1kbyte e um preparador de instruções paralelas, além das unidades de multiplicação e divisão, e execução. O processamento é feito sobre uma pilha de registradores de 6 portas que se comunica com a unidade de multiplicação e divisão, unidade de execução e controlador programável de interrupção por três barramentos (portas) de 64 *bit* e com os módulos de gerenciamento de memória por dois barramentos (portas) de 128 *bit*, um para operações de leitura, outro para operações de armazenamento. Há ainda uma porta de 32 *bit* para comunicação com uma *ULA* que processa endereços.

Tal arquitetura permite ao 80960CA executar múltiplas instruções a cada ciclo de *clock*, garantindo o desempenho citado.

Memória

O 80960CA pode endereçar diretamente 4 *Gbytes* de memória, a maior parte da qual disponível para o usuário. Internamente há também uma memória *RAM* de alta velocidade.

Operandos

Inteiros:

<i>byte integer</i>	8 bit	-2^{**7} a $2^{**7} - 1$
<i>short integer</i>	16bit	-2^{**15} a $2^{**15} - 1$
<i>integer</i>	32bit	-2^{**31} a $2^{**31} - 1$
<i>long integer</i>	64bit	-2^{**63} a $2^{**63} - 1$

Ordinais:

<i>byte ordinal</i>	8bit	0 a $2^{**8} - 1$
<i>short ordinal</i>	16bit	0 a $2^{**16} - 1$
<i>ordinal</i>	32bit	0 a $2^{**32} - 1$
<i>long ordinal</i>	64bit	0 a $2^{**64} - 1$

Não numéricos (apenas para *manipulação* e transporte de dados):”

<i>bit</i>	1bit
<i>bit field</i>	1 a 32 bit
palavra tripla	96 bit
<i>palavra quádrupla</i>	128 bit

Periféricos

Unidade de controle de barramento: gerencia o interfaceamento do 80960CA com a memória externa e periféricos com taxa de transferência de até 132 Mbytes/Seg em 32 bit, gerando também os tempos de espera para uso de diversos tipos de memória sem necessidade de lógica adicional.

Controlador de DMA

É um controlador de quatro canais que permite transferência entre periféricos e memória em taxas de até 66 Mbytes/Seg.

Controlador de interrupção: gerência a prioridade das interrupções ativadas por 8 entradas externas, 4 provindas do DMA e mais a entrada de interrupção não mascarável. Pela configuração destas 8 entradas podem ser suportadas 248 fontes de interrupção. É a operação do

controlador de interrupções que permite a baixa latência do atendimento das interrupções tipicamente inferior a 1μ .

A tabela 5.2 compara as principais figuras de mérito dos processadores citados.

5.3.1.2 SOLUÇÕES DE *HARDWARE* MULTIPROCESSADAS

i) Pipelined Vector Processors

São uma maneira natural de se implementar computadores vetoriais, que por sua vez se caracterizam por apresentar um *set* de instruções que operam sobre vetores, além daquelas que operam sobre escalares. São comuns entre suas aplicações a manipulação de vetores com 50.000 ou mais elementos [26]. Como perdem desempenho computacional com o tempo demandado para encher e esvaziar o *pipeline*, este tipo de arquitetura só apresenta todo o seu potencial processando longos arranjos de dados como em algumas aplicações científicas caracterizadas por processamento massivo. Os exemplos mais conhecidos desta classe de processadores são o CRAY e o CIBER

ii) Array Processors

São um arranjo paralelo de elementos processadores que operam de forma síncrona sob a supervisão de uma unidade de controle [9], que pode ser semelhante aos demais processadores. Os *array processors* constituem-se numa forma barata de implementação de processadores vetoriais e, sendo processadores específicos, normalmente trabalham acoplados através da unidade de controle a um *host*, responsável pelo gerenciamento, bibliotecas e I/O. Como os *array processors*. São constituídos por *CPUs* escalares, o processamento de vetores é baseado em bibliotecas elaboradas para tal fim e armazenadas no *host*. Uma das vantagens desta solução é o fato de se poder agrupar um número de elementos processadores suficiente para a aplicação em vista, além poder ser configurada fisicamente conforme a arquitetura mais adequada.

5.5 TENDÊNCIAS ATUAIS

Há uma tendência facilmente percebida nas tecnologias atuais de migrarem do domínio analógico para o digital, e por consequência de abordagem numérica, e do domínio do *hardware* para o do *software*. Tal tendência privilegia soluções que tenham como características alta velocidade computacional que possa permitir a programação em *software* de dispositivos antes implementados em *hardware* e, ao mesmo tempo, sejam eficientes no trato com algoritmos numéricos. Estes requisitos parecem apontar diretamente para os processadores *RISC*, em especial para os processadores de sinal digital (*DSPs*), com suas instruções (*MAC*) executadas em um único ciclo. No entanto ao se tratar de controle de movimento, há a limitação de velocidade pelos dispositivos eletrônicos de potência e de características inerentes à própria aplicação, como a máxima velocidade em que a pena de um *plotter* de alto desempenho consegue traçar adequadamente. Pode então surgir a situação de que seja interessante deixar o controle ao nível de atuadores ao cargo de um processador de menor velocidade porém com maiores facilidades de interfaceamento. Começam a surgir, então, os processadores que associam características de *DSPs* ou mesmo *RISCs* de uso geral com periféricos típicos de microcontroladores, ou microcontroladores com coprocessadores *DSP*[34]. Um exemplo é o *DSP 320C10* da Texas que recebeu periféricos típicos de microcontroladores na versão *320C14*.

Os fabricantes, sensíveis à necessidade de aproximar seu produto das reais necessidades dos usuários, tem lançado, além de versões de desempenho melhorado, também versões onde integram periféricos necessários em dadas aplicações, como também versões *stripped*, nas quais elementos nem sempre úteis em controle, como gerenciadores de *DMA*, são retirados. Assim, o usuário pode dispor do núcleo básico do processador para ao redor dele implementar os dispositivos que satisfaçam suas necessidades de forma otimizada.

Uma outra possibilidade de adequação do processador à aplicação é o uso de dispositivos customizados, normalmente oferecidos através de bibliotecas de células *ASIC*[7]. Há três tipos de tecnologias *ASIC* com possibilidades e custos diferenciados que podem se tornar atrativas ou não em função desempenho requerido e da quantidade de dispositivos a serem fabricados. A tecnologia mais barata é de *gate array*, mas tem escala de integração limitada pelo pela impossibilidade de fazer uso efetivo de todo o arranjo. A segunda possibilidade é o uso de células padrão que já conta com elementos como *ROM*, *PLA* e mesmo alguns núcleos de processadores. A terceira tecnologia é o *silicon compiler* onde dispositivos totalmente customizados são implementados e integrados [28].

Fig. de mér./ solução	μ controlador	DSP
Velocidade de processamento	10 Mips (80C196 @ 12MHz)	6,4 Mips (320C14 @ 25MHz)
Capacidade de interrupção	16 aninhamentos 4 linhas externas	4 aninhamentos (320C14) 16 aninhamentos (56000)
Capacidade de memória	64kbytes externa - EPROM interna opcional	dado: 256 palavras programa: 4 kpalavras (320C14) dado: 128 kpalavras programa: 64 kpalavras (56000)
Tamanho da palavra	16 bit (80C196)	16 bit (320C14) ; 24 bit (56000)
Comunicação	porta serial (80C196)	USART, I/O uso geral (320C14) 8 pinos da PB podem ser configura- dos como interface DMA, interf. serial com gerador de Baud rate, bus de dados, bus de endereços, interface CODEC (56000)
Interfaceamento - I/O	PWM, 2 timers, I/O alta velo- cidade (80C196)	PWM, I/O uso geral, timer (320C14) 3 portas de I/O (PA, PB e PC) com 24 bit cada.
Suporte ao desenvolvimento	placa EV80196KC, assem- bler, compilador, emulador <i>in</i> <i>circuit</i> (80C196)	placa Power 14 (320C14) placa DSP56001 ADM, software de desenvolvimento, simulador (56000)
Facilidade de gerenciamento	gerenciamento facilitado pela simplicidade da arquitetura e pelo set de instruções	programa monitor para placa Power 14 ; não há compilador (320C14)
Adequação ao algoritmo	dificuldade para processa- mento sobre matrizes	podem executar algoritmos de filtros digitais em tempo real

Transputer	RISC embedded proces.	ASIC
200 Mips de pico @ 5 MHz clock elevado internamente	66Mips (80C960CA) @ 25M Hz	tempo de micro instrução : 200 ns @ 20 M Hz
250.000 interrupções /s	248 aninhamentos = tabela com 248 vetores de interrupção	depende do processador ao qual está conectado
externa: 8 Mbytes com endereçamento direto interna: 4 kbytes	cache onchip 1k bytes externa: 4 Gbytes endereçamento direto (80C960)	depende do processador ao qual está conectado
32-bit (IMS9000)	32-bit (80C960CA)	16 bit
links seriais de alta velocidade - 50 Mpalavras/s	bus multiplexado de 32 bit	bus de 16 bit
2 timers de 32 bit 4 links seriais	bus multiplexado de 32 bit	ADC, PWM, contador, I/O de 8 bit
evaluating board, compilador, OCCAM	evaluating board, emulador in circuit, compilador, debugger	experimental
agrupador de instruções em hardware (IMS9000)	compilador em C, assembler otimizador	concebido para uso acoplado a um processador de uso geral que o gerencia
é um processador superescolar cuja grande vantagens são a velocidade e facilidade de formação de arrays multiprocessadores	seu set de instruções restrito não possui instruções específicas para para tratamento de vetores	arquitetura interna adequada ao controle de servo motores de posicionamento em tempo real

Em [28] também é apresentado o desenvolvimento de um processador para controle de atuadores baseado em *ASIC* para uso acoplado a um processador de uso geral.

Em [24] é apresentado um diagrama (tab. 5.3) mostrando a tendência de uso dos processadores à medida que as aplicações vão se tornando mais exigentes, de acordo com a tecnologia escolhida.

5.6 CONCLUSÕES

- No momento da escolha do *hardware* de um sistema de controle pode ser mais importante pensar em termos de características de placas do que *chips*.
- O aspecto custo, de um *hardware* para controle, pode estar muito mais ligado aos periféricos do que ao processador em si.

<i>RISC</i> de alta performance	<i>CISC</i> de alta performance	<i>CISCstripped</i>	<i>CISC</i> de alta integração	<i>RISC</i> de alta performance+ custom <i>ASIC</i>	<i>CISC</i> + custom <i>ASIC</i>
SPARC					
R3000	68030	68EC030	6830XX		
29000	68020	68EC020	6820XX	29000+ custom <i>ASIC</i>	
i960CA	68010	68EC010	6810XX	i960+ custom <i>ASIC</i>	68000+ custom <i>ASIC</i>

Tabela 5.3 Processadores em ordem crescente de desempenho (baixo para cima) em cada tecnologia.

Na escolha *CISC* x *RISC* a disponibilidade de tempo para desenvolvimento e a sensibilidade do projeto ao fator custo podem ser mais determinantes do que os fatores tecnológicos.

- Os *DSPs* são uma opção interessante em controle se se pretende dar ao problema uma abordagem digital (discreta).
- A utilização de processamento vetorial ou paralelo só é conveniente se a tarefa não pode ser executada por um *hardware* monoprocessador.
- Em algumas situações há limitações por parte da planta controlada que possibilitam o uso de processadores mais lentos sem prejuízo do desempenho global.
- A completa adequação do *hardware* controlador à aplicação em vista pode ser conseguida pela utilização de produtos customizados

CAPÍTULO 6. CONTRIBUIÇÕES À CONSTRUÇÃO

Neste capítulo serão apresentados algumas considerações que deverão ser tomadas em conta quando da construção do protótipo do simulador.

6.1 ACIONAMENTO DA CABINE

A solução inicialmente sugerida para acionamento do eixo 3, solidário à cabine, usava um rolete de material flexível como roda de atrito para transmitir o movimento do motor, fixo ao anel externo, à cabine. Esta solução implicava em um alto coeficiente de atrito que seria gerado pela deformação do rolete sob a grande pré-carga necessária para minimizar o escorregamento. Apesar disto, não haveria como garantir a posição da cabine pois sempre seria possível um certo escorregamento especialmente em trajetórias em zigue-zague realizadas em alta velocidade. Este problema admite duas soluções, colocar o sensor de posição após a transmissão, ou adotar um sistema de transmissão que torne a posição da cabine solidária à posição da redução e, por conseguinte, do motor.

A primeira solução é construtivamente inadequada pois além de obrigar o uso de sensores de posição separados do motor, o que em si já é uma desvantagem, é ainda dificultada pelas características da planta. Em primeiro lugar porque é complicado alojar um *encoder*, por exemplo, em um anel acionado pela periferia, segundo, porque como o simulador admite movimentos ilimitados, o sinal de retorno do sensor de posição demandaria mais esforço de engenharia para fazê-lo passar pela junta rotacional entre o anel externo a cabine.

A segunda solução, que é mais simples construtivamente, admite o uso de corrente ou correia dentada para fazer a transmissão. O uso de corrente tem a vantagem de que ela pode ser emendada em comprimentos bastante próximos do requerido necessitando apenas do uso de um esticador de pequeno curso. Porém seu uso não é recomendado quando se tem uma engrenagem de pequeno diâmetro com alta velocidade, o que acontece na saída do acionamento do eixo 3, onde tem-se uma redução ainda pequena da velocidade do motor [42].

A solução adotada no projeto foi então a da correia dentada, que é mais silenciosa e por ser mais leve não contribui significativamente para o aumento da inércia. Sua desvantagem está na

necessidade provável de ter de ser obtida por encomenda. Em [43] pode-se encontrar o tratamento teórico da distribuição de esforços em correias dentadas.

6.2 CONSTRUÇÃO DA CARENAGEM

A construção da carenagem do simulador e a previsão do local de acesso para embarque e desembarque devem levar em conta as configurações que a estrutura pode assumir. Tal como o pulso esférico de um manipulador, a estrutura do simulador pode assumir a mesma atitude com dois *sets* de ângulos de junta diferentes (fig. 6.1). Assim, ao simular um avião, o piloto pode entrar em vôo invertido girando o eixo 2 de 90 graus, sair do vôo invertido girando o eixo 3 de -90 graus e aterrissar em direção oposta à que estava ao decolar.

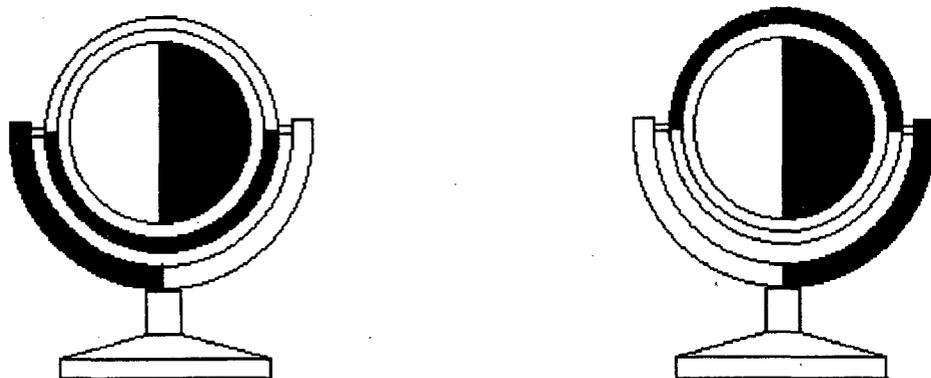


Figura 6.1 uma mesma orientação da cabine pode ser conseguida por duas configurações diferentes.

Para evitar a complicação da criação de rotinas de controle que trariam a cabine de volta a posição de desembarque, que ainda apresenta a desvantagem de distanciar a simulação da realidade, o ideal é que se adote um desenho de carenagem que possibilite o embarque e o desembarque em qualquer atitude em que a cabine esteja em orientação horizontal não invertida.

6.3 POSIÇÃO DO PILOTO

A proposta inicial do simulador previa a posição de pilotagem com o piloto voltado para o anel que suporta a cabine. Por observação da maquete, a posição de pilotagem foi mudada colocando-

se o piloto agora voltado para o centro de um dos hemisférios da cabine. Embora a mudança não tenha sido feita por motivos ligados à dinâmica da estrutura, ela acabou por ser bastante beneficiada neste aspecto. A maioria dos objetos a serem simulados, em especial os aviões, tem sua dinâmica mais rápida no eixo de rolamento. Em alguns modelos de avião a relação entre a aceleração de rolamento e as demais chega a mais de 10 vezes. Dessa forma, pela simples mudança da posição da poltrona do piloto mudou-se o movimento de rolamento para o eixo 3, o último da estrutura, logo o de menor inércia, necessitando portanto de menores torques de acionamento para sua aceleração. Por estar no eixo 3, o movimento de rolamento não depende da configuração da estrutura; desta forma, nunca necessitará ser composto por movimentos simultâneos de duas juntas, nem será degradado por condições de singularidade e poderá receber tratamento privilegiado com acionamentos e controle que lhe garantam o desempenho dinâmico requerido.

6.4 PASSAGEM DE SINAIS E ENERGIA PELAS JUNTAS

Uma das características que diferenciam o projeto do simulador, objeto deste estudo, de conceitos já existentes é a capacidade de movimento angular ilimitado. Para tanto é necessária a solução do problema de engenharia, de se fazer passar sinais de controle e energia de um segmento da estrutura para o outro.

A solução mais natural é o uso de conjuntos anel/escova (*slip-rings*) que, pela simplicidade e robustez, parecem bastante adequados para a condução da energia, com o inconveniente de poder gerar ruído suficiente para afetar os sinais de controle que trafeguem nas proximidades. Já a transmissão dos sinais de controle o uso de conjuntos anel/escova, mesmo que se procure aumentar a segurança usando anéis com múltiplas escovas e uma escolha criteriosa de materiais, podem aparecer ruídos perigosos pelo fato de se trabalhar com uma malha de controle com sinais de alta velocidade e possíveis ganhos derivativos.

Uma técnica promissora é a que usa transformadores rotativos como o esquematizado na figura 6.2. Teste realizado em um modelo normalmente empregado em gravadores profissionais de vídeo mostrou que sinais de 30 MHz conseguiam passar com atenuação desprezível. Mesmo a transmissão de energia pode ser feita através de transformadores rotativos. [45] apresenta o projeto de transformadores de força rotativos desenvolvidos a princípio para uso espacial que, caso necessário podem ser adaptados ao projeto do simulador.

6.5 BALANCEAMENTO DE MASSAS

Como apresentado em [46], pode-se demonstrar a viabilidade de acrescentar massa à uma estrutura de forma a balancear os seus segmentos. As vantagens do balanceamento em si são várias. A eliminação da componente gravitacional permite o uso do torque disponível para promover maiores acelerações e diminuir a incidência de perturbações sobre o acionador. Pode-se também fazer o balanceamento de modo a eliminar assimetrias de massa de difícil modelagem e que poderiam dar origem a

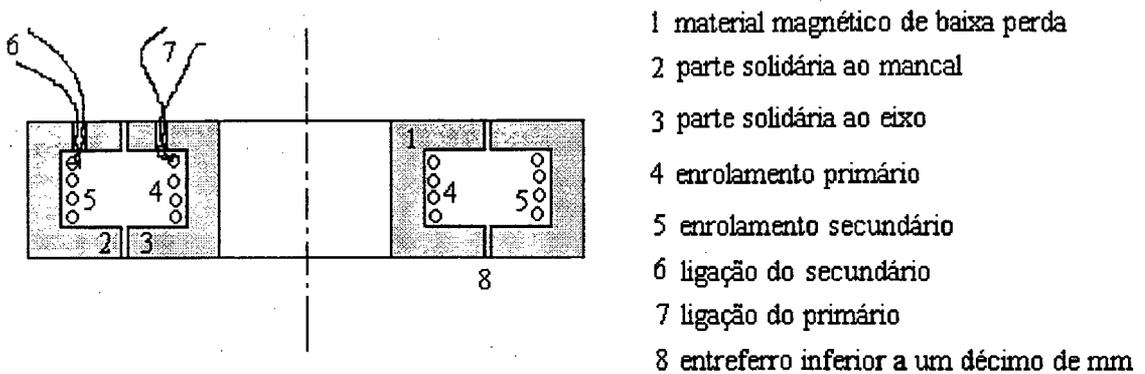


Figura 6.2 esquema em corte de transformador rotativo para transferência de sinais entre partes móveis

torques de precessão. A desvantagem do balanceamento pelo acréscimo de massa está no previsível aumento da componente de torque inercial, componente responsável pela aceleração. A vantagem ou não do uso desta estratégia é determinada, a princípio, pela comparação das componentes de torque inercial e gravitacional, podendo-se acrescentar massa de balanceamento até que ambos se igualem. Como, no entanto, as estruturas experimentam várias situações diferentes durante a operação, como variação da inércia ao longo do volume de trabalho, a comparação de torques deve ser feita sobre uma média que represente as tarefas típicas da estrutura, e deve-se ter sempre em mente a economia de energia por um lado e o desempenho da estrutura por outro.

No caso do simulador este questionamento pode ser feito sobre o anel externo, devido ao desbalanceamento provocado pelo conjunto que aciona a cabine e também no caso da cabine devido à colocação excêntrica dos equipamentos. Como com o modelo que se dispõe é impossível verificar a viabilidade por simulação, uma decisão pelo uso ou não de massas de balanceamento só poderá ser tomada quando se tiver construído um protótipo com completa definição dos parâmetros de distribuição de massa e melhor clareza das tarefas típicas a serem realizadas.

Com base no que se descreveu acima, deve-se projetar a estrutura seguindo os seguintes critérios, na ordem, conforme se tenha maior refinamento das informações de desempenho requerido e do modelo da estrutura:

- ter a menor massa possível
- balancear a distribuição de massa na estrutura massa
- acrescentar massa à estrutura para conseguir o balanceamento.

6.6 DIMINUIÇÃO DA FIAÇÃO ENTRE OS SEGMENTOS

Com a colocação dos motores proposta inicialmente, um fixo à base, um fixo ao garfo e o terceiro fixo ao anel externo, há a necessidade do uso de, no mínimo, 24 pares anel-escova somente para levar alimentação trifásica aos motores e trazer os sinais dos *encoders* para o controlador, afora os sinais que deverão trafegar entre o processador que estará na cabine lendo os controles, o processador que efetuará o controle de baixo nível fechando as malhas e o computador principal, responsável pelo *software* de simulação e geração de imagens.

No que diz respeito aos acionamentos, o número de condutores para a passagem de energia e sinais pelas juntas pode ser substancialmente diminuído se os três motores e o processador que fecha as malhas de controle estiverem todos alocados no garfo. Neste caso, o motor que movimenta o garfo ao invés de estar solidário à base estará fixo ao garfo obtendo seu movimento por reação da base. A alteração mais complexa é a do acionamento da cabine, que deve ser feito apoiando o anel externo em um eixo vazado pelo qual passará a árvore de transmissão que acoplará movimento do motor, figura 6.3. Além da economia, a redução do número de conjuntos anel-escova torna o sistema elétrico menos complexo, com menor necessidade de manutenção e, sobretudo, mais seguro, pois, como todo o sistema de controle de baixo nível está colocado em um único segmento, o garfo, não haverá o risco de abertura acidental das malhas de controle.

Do ponto de vista da inércia, não haverá variação significativa devida à mudança do motor da base para o garfo, pois este continuaria em posição bastante central. Já a inércia do anel externo seria diminuída com a transposição do motor que aciona da cabine para o garfo. Contra esta última solução pesam as desvantagens de uma mecânica mais complexa e o pequeno acoplamento cinético que apareceria entre os movimentos do anel externo e da cabine. Uma outra possibilidade para a diminuição do número de pares anel/escova utilizados para transmissão de energia através das juntas é pelo emprego de alimentação CC a dois fios. Um retificador trifásico pode ser colocado na base, tirando proveito das vantagens da retificação trifásica, enquanto que conversores CC/CC adequam o

nível da tensão aos requeridos pelos equipamentos da cabine como processador, ventilação e iluminação.

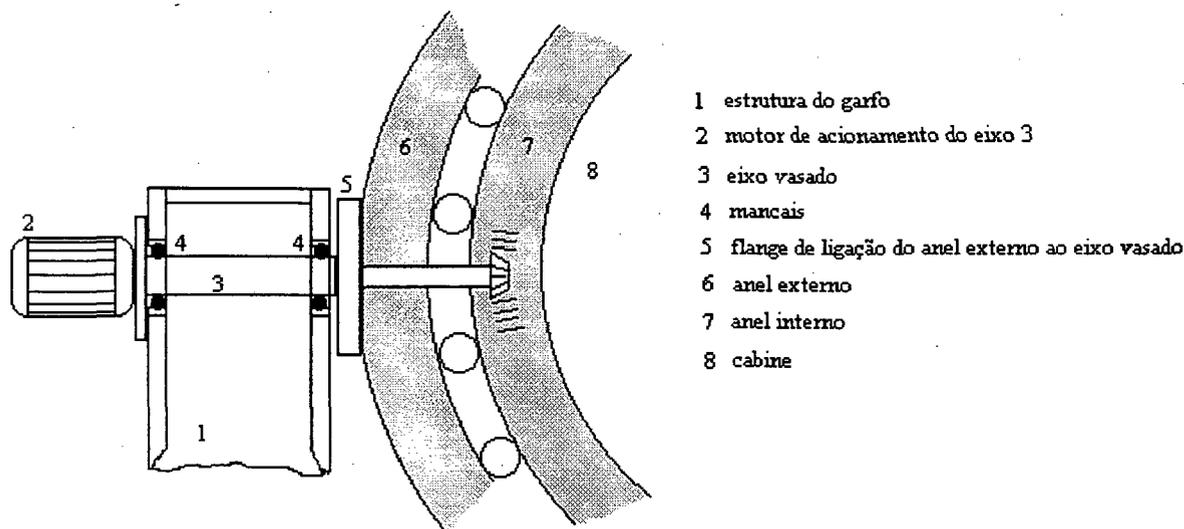


Figura 6.3 Esquema conceitual do acionamento da cabine com motor colocado solidário ao garfo

6.7 ACIONAMENTO

A princípio foram cogitados dois tipos de acionamento para a estrutura, hidráulico e elétrico. O hidráulico apresenta a vantagem de não dissipar energia enquanto a estrutura fica estacionária em posição que desafia a gravidade, e é capaz de promover grandes torques sob velocidades altas. No entanto foi logo descartado pelo fato de que a estrutura deve realizar movimentos ilimitados, o que demandaria sistemas de transmissão do fluido difíceis de serem implementados com suficiente desempenho. Prevaleceu então o uso do acionamento elétrico.

Pela magnitude dos torques envolvidos é impossível encontrar acionadores do tipo *direct drive*, que também tornam mais difícil a tarefa do controle. Assim, nos eixos 2 e 3 optou-se pela especificação de servo-motores e no eixo 1, devido a baixa velocidade, foi especificado um motor de torque.

6.8 USO DE GIROSCÓPIO

Uma solução alternativa ao uso de *encoders* nos eixos dos motores que, embora mais cara, pode evitar o transito de fiação de um para outro segmento é o uso de giroscópios, um para cada direção, localizados na cabine. Os giroscópios podem ser do tipo eletromecânico ou a laser. Este último tem a vantagem da alta precisão - detectam velocidades inferiores a uma revolução por ano - e inexistência de partes móveis [49].

CAPÍTULO 7 CONCLUSÃO

Neste trabalho são abordados todos os aspectos relativos ao uso da estrutura proposta como um simulador de vôo, procurando evidenciar alguns dos problemas de engenharia a serem superados e suas possíveis soluções, citando, quando necessário, referências bibliográficas onde o assunto é tratado com maior profundidade.

Do ponto de vista da cinemática, caracterizou-se a estrutura como sendo semelhante ao pulso esférico de um manipulador robótico, com a diferença de que o jacobiano converte as velocidades do sistema de coordenadas da base para o sistema de coordenadas do segmento extremo da estrutura, pois as velocidades pretendidas são vistas do ponto de vista do piloto que está no segmento extremo (cabine), e não na base.

Sob o enfoque da dinâmica, a estrutura mecânica do simulador pode ser analisada com as teorias desenvolvidas para o modelamento de manipuladores com três graus de liberdade, com a simplificação de que este não possui movimentos de translação. Foi utilizado um modelo obtido por formulação de Lagrange que permite melhor separação das componentes de torque facilitando a interpretação de resultados.

Os resultados das simulações revelaram que para as velocidades e acelerações máximas requeridas para a simulação de um avião de treinamento de caça as componentes de torque são, por ordem de valor por eixo:

- eixo 1: coriolis, inercial, atrito
- eixo 2 : coriolis, gravitacional, inercial, atrito
- eixo 3 : inercial, gravitacional, coriolis, atrito

A componente inercial depende da quantidade de massa e sua distribuição na estrutura e da aceleração que se quer imprimir a ela, tornando-se, logo, tanto maior quanto maior for a aceleração requerida. Pode ser diminuída pela redução da massa nas regiões dos segmentos que se encontram mais afastadas dos eixos de giro. As componentes centrífugas e de coriolis dependem da geometria da estrutura e da sua velocidade. Quanto maiores forem as velocidades impressas à estrutura mais significativas serão as componentes de torque de acoplamento. A parcela gravitacional depende da massa e de sua distribuição na estrutura. Pode ser diminuída por balanceamento de massas. O torque de atrito depende do tipo de contato entre as superfícies com movimento relativo entre si e de possíveis deformações dinâmicas. Pode ser diminuída pelo uso de mancais rolamentados em locais adequados e pelo uso de materiais convenientes na fabricação dos roletes de sustentação da cabine.

A solução que surge tanto do ponto de vista do *hardware* como de estratégias de controle, em especial para a construção do protótipo, é a utilização do pacote PMAC. Além de *hardware* de alta velocidade baseado em DSP, este pacote já possui as interfaces necessárias implementadas em ASIC e algoritmos tanto de controle como de análise da planta para auxílio no desenvolvimento de controladores otimizados para a estrutura.

Recomenda-se que ao ser construído o protótipo da estrutura mecânica, o sistema de aquisição do pacote PMAC seja utilizado para validação do modelo dinâmico especialmente os parâmetros de atrito. Estes últimos podem ser obtidos pelo acionamento, a velocidade constante, de um dos eixos a cada vez medindo-se o torque aplicado e a velocidade.

A escolha final da estratégia de controle a ser adotada depende da completa definição dos parâmetros físicos da estrutura, do seu comportamento dinâmico (ausência ou não de oscilações e vibrações), das características típicas e extremas das trajetórias a serem simuladas, de decisões de como enfrentar problemas como a variação do peso do piloto e sua posição (correção mecânica ou controle adaptativo) e mesmo da aplicação do simulador (profissional ou entretenimento).

Tal como no controle de manipuladores robóticos, também no controle do simulador tem-se três níveis hierárquicos respectivamente no robô / simulador:

- gerador de trajetórias / simulador de comportamento do avião
- cinemática inversa / cinemática inversa
- fechamento das malhas de controle / fechamento das malhas de controle.

O primeiro nível deverá processar o modelo aerodinâmico e de propulsão do avião, devendo ser executado em *hardware* com desempenho suficiente para processar também a parte gráfica do simulador. Este processador poderá ser uma unidade separada dos níveis mais baixos hierarquicamente. Os dois últimos níveis mais baixos devem ser executados em processador com capacidade suficiente para o cálculo das transformações cinemáticas e algoritmos de fechamento das malhas de controle. Devido a intensa troca de dados entre estes dois níveis é recomendável que ocupem um mesmo processador.

Como próximas etapas da continuidade do trabalho propõe-se:

- Modelação do acionamento.
- Comparação por simulação das principais estratégias de controle.

- Implementação da cinemática inversa comparando os resultados obtidos em cada um dos casos: inversa da matriz jacobiana, pseudo inversa da jacobiana, transposta da jacobiana e método dos mínimos quadrados amortecido.
- Construção de protótipo para validação de resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] - Craig, J. J. , Introduction to robotics: mechanics and control, Addison-Wesley Publishing Company, 1986.
- [2] - Asada, H. & Slotine, J-J. E. , Robot analysis and control. John Wiley and Sons, Inc, 1986.
- [3] - Spong, M. W. & Vidyasagar, M. , Robot dynamics and control, John Wiley and Sons, Inc, 1989.
- [4] - Silva, J. C. , "Desenvolvimento de um sistema para análise e modelamento de robôs industriais." Dissertação de mestrado, Florianópolis, setembro 1990.
- [5] - Silva, J. C. , "Relatório do projeto simulador". Florianópolis, julho 1995
- [6] - Liu, G.-J. & Goldenberg, A. A. , "Asyntotically stable robust control of robot manipulators." Mechanical machines theory, Pergamon, vol 31, 5 , 1996. pp. 607-618.
- [7] - Mulders, P. C. , *et alli* , "Optimal trajectory control of a linear robotarm by a state space method." Annals of the CIRP, vol 38, jan. 1989.
- [8] - Dhanaraj, C. & Sharan, A. M. , "Efficient modeling of rigid link robot dynamic problems whith friction." Mechanical machines theory, Pergamon, vol 30, 5 , 1995. pp 749 a 764.
- [9] - Abdallah, D. D. , *et alli* , "Survey of robust control for rigid robots." IEEE control systems, fev 1995. pp 24 a 30.
- [10] - Arimoto, S. , "Passivity of robot dynamics implies capability of motor program learning." Lecture notes in control and information sciences - advanced robot control, Springer- Verlag, vol 162, nov. 1990. pp 49 a 68.
- [11] - Serafi, K. El & Khalil, W. , "Energy based robots controler." Lecture notes in control and information sciences - advanced robot control, Springer- Verlag, vol 162, nov. 1990. pp 30 a 48.
- [12] - Goes, L. C. S. , "Controle adaptativo de um manipulador robótico." Anais do 8º. congresso brasileiro de automática, Belém 1989. pp 118 a 122.
- [13] - Behera, L. , *et alli* , "Neuro adaptive hybrid controller for robot-manipulator tracking control." IEE proc. - control theory appl. , vol 143, No 5, mai 1996. pp 270 a 275.
- [14] - Shoram, M. , *et alli* , "neural network control of robot arms." Annals of the CIRP, vol. 41, jan 1992. pp 407 a 410.

- [15] - Tan, S. , & Yu, Y. , "Adaptive fuzzy modeling of nonlinear dynamical systems." *Automatica*, vol. 32, No. 4, 1996. pp 637 a 643.
- [16] - Brogliato, B. , *et alli*, "Global tracking controllers for flexible-joint manipulators : a comparative study." *Automatica*, vol. 31, No. 7, 1995. pp 941 a 956.
- [17] - Xi, F. , "Trajectory tracking of a spatial flexible link manipulator using inverse dynamics method." *Mechanical machines theory*, vol. 30, No. 8, 1995. pp 1113 a 1126.
- [18] - Singh, S. & Schy, A. A. , "Elastic robot control : nonlinear inversion and linear stablization." *IEEE trans. on aerop. and eletronic systems*, vol. AES-22, No. 4, jul 1986. pp 340 a 347.
- [19] - Boyer, F. & Coiffet, P. , "Symbolic modeling of a flexible manipulator via assembling of its generalized newton-euler model." *Mechanical machines theory*, vol. 31, No. 1, 1996. pp. 45 a 56.
- [20] - ElMaraghy, W. H. *et alli*, "design and control of robots with flexibilities." *Annals of the CIRP*, vol. 43, jan 1994. pp. 359 a 362.
- [21] - Chiaverini, S. *et alli*, "Control of robotic systems through singularities." *Lecture notes in control and information sciences - advanced robot control*, Springer- Verlag, vol 162, nov. 1990. pp 285 a 295.
- [22] - Egeland, O. & Spangelo, I. , " Manipulator control in singular configurations - motions in degenerate directions." *Lecture notes in control and information sciences - advanced robot control*, Springer- Verlag, vol 162, nov. 1990. pp 296 a 306.
- [23] - Motchalov, I. A. , " Sistemas de controle extremal dos objetos inerciais. " *Notas de aula*, Caxias do Sul 1995.
- [24] - Child, J. & Wilson, D. , " RISC champions challenge Moto in embedded control." *Computer design*, out. 1991. Pp. 98 a 116.
- [25] - Andrews, W. , "Applications determines best approach to multiprocessing." *Computer design*, dez. 1990. pp. 72 a 86.
- [26] - Hwang & Brigs, *Designing efficient algorithms for paralel computers*, Mc Graw Hill - University of California, 1989.
- [27] - Bucella, T. , "Comparing DSPs to microprocessors in motion control systems: some real world data." *Intelligent motion - proceedings*, out. 1990.
- [28] - Schepper, F. de, *et alli*, "Development of na ASIC performing high speed current loop processing of servo motor control for mechatronics applications." *Annals of the CIRP*, vol. 38, jan 1989.
- [29] - Mazola, V. , "Organização de sistemas computacionais." *Notas de aula*, Florianópolis, 1994.

- [30] - Andrews, W. , "RISC-based boards make headway in real time applications." Computer design, out. 1991. pp. 69 a 80.
- [31] - Altintas, Y. & Munasinghe, W. K. , "A hierarchical open-architecture CNC system for machine tools." Annals of the CIRP, vol. 43, jan. 1994. pp. 349 a 354.
- [32] - Duelen, G. *et alli*, "Na advanced control system for manufacturing processes." Annals of the CIRP, vol. 40, jan. 1991.
- [33] - Wilson, R. , "Higher speeds push embedded systems to multiprocessing." Computer design, jul. 1989. pp. 72 a 83.
- [34] - Bucella, T. & Ahmed, I. , "Taking control with DSPs." Machine design, out. 1989. pp. 73 a 80.
- [35] - " Board offers High performance DSP for STD bus." Computer design, out. 1991.
- [36] - Andrews, W. , "Distinction blur between DSP solutions." Computer design, mai. 1989. pp. 86 a 99.
- [37] - Jain, Y. , "Tapping the power of DSPs." Machine design, jun. 1990. pp. 73 a 80.
- [38] - "56002- na evolution of excellence." DSP news, Motorola, vol. 8, num. 1, mai. 1995.
- [39] - Motorola technical training- DSP560000/1 course notes, Motorola, 1992.
- [40] - Mitchell, H. J. , 32 bits - microprocessor, McGraw-Hill, 1988.
- [41] - Hwang, K. & Briggs, F. A. , Computer architecture and parallel processing, Mc Graw-Hill, 1989.
- [42] - L'Hote, F. *et alli*, Robot technology - robot components and systems, Prentice-Hall, 1983.
- [43] - Karolev, N. A. & Gold, P. W., "Load distribution of timing belt drives transmitting variable torques." Mechanical machines theory, vol. 30, mai. 1995.
- [44] - Stuart, T. A. *et alli*, "Rotary transformer design with fixed magnetizing / or leakage inductances." IEEE transactions on aerospace and electronics, vol. AES 22, num. 5, set. 1986.
- [45] - Diken, H. , "Effect of mass balancing on the actuator torques of a manipulator." Mechanical machines theory, vol. 30, num. 4, out. 1995.
- [46] - Dunkel, W. , "An improved model of a propeller-driven type aircraft." Mathematical modelling of systems, vol. 1, num. 1, 1995, pp. 47 a 62.
- [47] - Ogata, K. , Engenharia de controle moderno, Prentice/Hall do Brasil, 1982.
- [48] - Werneck, Transdutores, Editora, LTC, 1995

ANEXO

Este anexo consiste nos arquivos de entrada de dados, arquivos de saída gráfica e um arquivo macro do programa SIMNOM , utilizado nas simulações da dinâmica da estrutura do simulador segundo modelo levantado por [4].

CONTINUOUS SYSTEM atritob

" Version: 1.0

" Abstract: SISTEMA DO SIMULADOR GIROSDÓPICO

" Description: Este arquivo visa o calculo dos

" termos de atrito e gravitacionais

" Revision: 1.0

" Author: Jonny Carlos da Silva

" Created: 30/10/95

" Revised: 08/11/95

" Inputs and outputs:

INPUT a1 a2 a3 xs1 xs2 xs3 vs1 vs2 vs3

OUTPUT ma2 mg2

" Respectivamente este termos representam os torques

" de atrito e gravitacional nas junta 2

" States, derivates and time:

"STATE x1 v1 x2 v2 x3 v3 " ângulos e velocidades do giroscópico

"DER dx1 dv1 dx2 dv2 dx3 dv3

"A definicao de estados e derivadas não é necessária

"tendo em vista que os valores de posicao, velocidade

"e aceleracao são inputs do arquivo giro1.t

"TIME t

dv1=a1 "estas variaveis são saidas do

dv2=a2 "arquivo giro.t

dv3=a3

x1=xs1

x2=xs2

x3=xs3

v1=vs1

v2=vs2

v3=vs3