

Marcelo C. A. Costa
Sílvio R. Micheloto

8.0 (oit)

15m

Identificação de Parâmetros para um Modelo de Atrito em um Braço Flexível em Rotação

Trabalho de formatura para graduação em
Engenharia Mecatrônica no Departamento
de Engenharia Mecânica da EPUSP

Orientador: Prof. Dr. Raul Gonzalez Lima

São Paulo

1998

AGRADECIMENTOS

Agradecemos a todos que colaboraram com a realização deste trabalho, direta ou indiretamente, e em especial ao professor Raul Gonzalez Lima, pelo trabalho de orientação.

RESUMO

O presente trabalho tem por objetivo o desenvolvimento de um modelo de atrito no eixo de um motor que rotaciona uma barra flexível. Utilizou-se uma bancada experimental desenvolvida em anos anteriores, sendo que foram necessárias algumas alterações de hardware que consumiram grande parte do tempo disponível.

A bancada utilizava, basicamente, um microcomputador para a geração de sinais via software, uma placa conversora AD/DA, um amplificador, uma fonte, um motor de corrente contínua, um encoder e uma barra flexível de alumínio engastada no eixo do motor.

Espera-se que este trabalho sirva a outras linhas de pesquisa dentro da EPUSP, como a de controle de estruturas flexíveis, por exemplo.

ABSTRACT

The aim of this work is to develop a friction model in a shaft of a motor which rotates a flexible beam. A experimental set, developed some years ago, was used and some hardware modifications were made and were responsible for large waste of time.

The experimental set used, basically, a microcomputer for signal generation by software, a AD/DA converter board, an amplifier, an electric DC motor, an encoder and an aluminum flexible beam attached to the motor's shaft.

We're hopeful that this work will be useful in other researches at EPUSP, as, for example, the studies about the control of flexible structures.

ÍNDICE

1. INTRODUÇÃO	2
2. MODELAGEM.....	4
2.1. MODELAGEM DINÂMICA DA BARRA	4
2.1.1. MODELAGEM DINÂMICA DO BRAÇO FLEXÍVEL.....	4
2.2. MODELOS DE ATRITO.....	5
2.2.1. MODELO DE ATRITO LINEAR DE UM MOTOR DE CORRENTE CONTÍNUA.....	6
2.2.2. MODELO DE MOTOR CC COM ATRITO NÃO-LINEAR.....	7
2.2.3. STICK-SLIP FRICTION	8
3. ANÁLISE DO PROBLEMA UTILIZANDO O MODELO DE ATRITO VISCOZO	10
4. MESMA ANÁLISE COM A INTRODUÇÃO DE DOIS MODOS DE VIBRAR	11
5. MODIFICAÇÕES DE HARDWARE	14
5.1. COLOCAÇÃO DE NOVA BARRA.....	14
5.2. MONTAGEM DO CONECTOR DB25	14
5.3. AJUSTE DE OFFSET	15
6. MODIFICAÇÕES DE SOFTWARE.....	16
6.1. TESTE DOS CANAIS UTILIZADOS	16
6.2. TESTE DO MODO DE EXPANSÃO DA PLACA CAD 12/36.....	16
6.3. SOFTWARE PARA GERAÇÃO DO TORQUE.....	17
7. PROCEDIMENTO EXPERIMENTAL E RESULTADOS	18
7.1. DETERMINAÇÃO DA ENTRADA	18
8. IMPLEMENTAÇÃO DO MODELO DE ATRITO.....	23
9. BIBLIOGRAFIA	25
APÊNDICE A – SOFTWARES UTILIZADOS (TURBO PASCAL)	26
A.1. SLEW6.PAS	26
A.2. OFFSET.....	33
APÊNDICE B – ROTINAS EM MATLAB	34
B.1. ROTINA TORQUE2.M.....	34
B.2. TORQUE.M.....	34
B.3. F5.M.....	35
APÊNDICE C – MEDIÇÕES OBTIDAS.....	38
C.1. ARQ_POSM.DAT	38

1. INTRODUÇÃO

O presente trabalho tem como proposta o desenvolvimento de um modelo de atrito no eixo de acionamento uma barra flexível.

Os recentes avanços de, por exemplo, robôs manipuladores e posicionadores de grande velocidade e precisão, têm demandado o desenvolvimento de estruturas cada vez mais leves e flexíveis. Entretanto, se por um lado estruturas flexíveis representam uma redução no custo total e no consumo de energia, além de maior velocidade no deslocamento de estruturas, por outro geram maiores níveis de vibração, com um maior tempo de amortecimento na realização de operações se comparadas a estruturas "rígidas". Entre as aplicações práticas deste trabalho encontram-se, como já dito anteriormente, o desenvolvimento de robôs posicionadores e manipuladores, além de leitores de disco rígido, apêndices de satélites, etc..

Para reduzir estes níveis de vibração a padrões aceitáveis, propõe-se um controle, de malha fechada, da oscilação da barra flexível ao longo de uma manobra angular.

O problema a princípio gira em torno da seleção de um modelo de atrito para o eixo do motor DC utilizado para prover o torque de giro da estrutura flexível que melhor se aproxime do caso real. A obtenção deste modelo de atrito é essencial para a linearização do modelo do motor que será usado na malha de controle que pode ser desenvolvida posteriormente.

Para este fim, uma bancada experimental ilustrada abaixo foi utilizada:

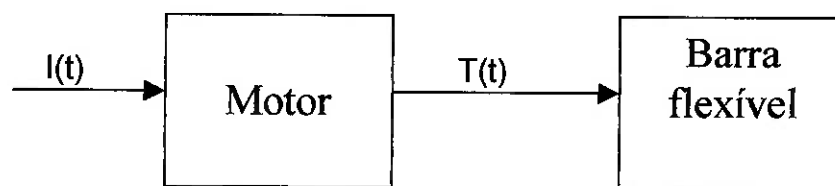


Figura 1: representação esquemática da bancada experimental

em que $i(t)$ é a corrente entregue pelo driver do motor e $T(t)$ o torque mecânico provido pelo motor. A bancada foi inicialmente desenvolvida pelo

PET e é composta de um motor de corrente contínua, encoder, amplificador de potência (do sinal que sai do microcomputador para o motor), barra flexível, além de um microcomputador e uma placa de aquisição de sinais.

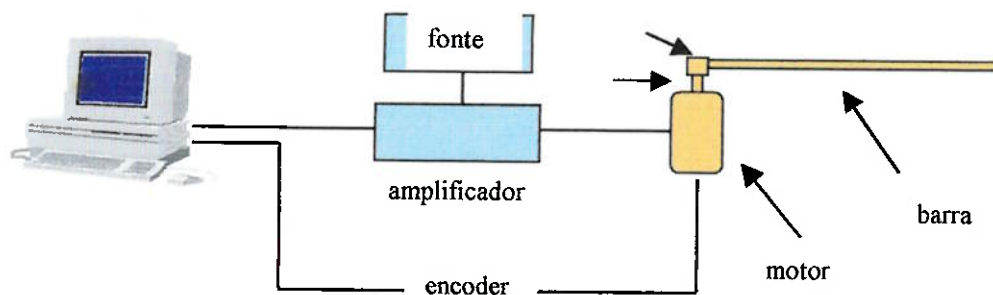


Figura 2: representação esquemática dos componentes da bancada

Deve-se ressaltar que a viabilidade prática deste trabalho já foi testada anteriormente pelo professor Raul G. Lima e que tal estudo (o desenvolvimento do modelo de atrito) pode ser útil a uma série de outros trabalhos já iniciados no Departamento de Engenharia Mecânica.

2. MODELAGEM

2.1. MODELAGEM DINÂMICA DA BARRA

2.1.1. MODELAGEM DINÂMICA DO BRAÇO FLEXÍVEL

O modelo matemático da dinâmica da barra já havia sido desenvolvido em Gildin et al.[1] e utilizou o princípio extendido de Hamilton e que pode ser expresso da seguinte forma:

$$\int_{t_1}^{t_2} (\delta T - \delta V + \delta W_{nc}) dt \quad (01)$$

em que δ indica a primeira variação, t_1 e t_2 são os limites de tempo de integração, T e V são respectivamente a Energia Cinética e Potencial da barra e W_{nc} é o trabalho das forças não conservativas do sistema.

A partir deste princípio e utilizando as condições de contorno e algumas simplificações, deriva-se o modelo discreto da vibração transversal da viga por superposição de autofunções,

$$y(x,t) = \sum_{r=1}^{\infty} \eta_r(t) \phi_r(x) \quad (02)$$

onde η_r representa a coordenada modal e ϕ_r a função de forma.

Desenvolvendo analiticamente a equação (01) e utilizando a hipótese de movimento síncrono, é possível derivar as matrizes de estado do sistema barra (A, B e C):

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx \\ x &= \begin{pmatrix} \theta \\ \dot{\theta} \\ \eta_1 \\ \dot{\eta}_1 \\ \eta_2 \\ \dot{\eta}_2 \end{pmatrix} \\ A &= \begin{pmatrix} 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & -737.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & -7782.9 & 0 \end{pmatrix} \\ B &= \begin{pmatrix} 0 \\ 9.434 \\ 0 \\ -29.9156 \\ 0 \\ -59.2337 \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4735 & 0 & -0.2217 & 0 \end{pmatrix}\end{aligned}$$

A vetor x representa as variáveis de estado do objeto de controle: barra flexível.

Deve-se ressaltar que a modelagem acima proposta foi usada na primeira parte deste trabalho (PMC 580 – Projeto Mecânico I), sendo que as matrizes foram alteradas no decorrer dos experimentos, como pode-se constatar em itens subseqüentes.

2.2. MODELOS DE ATRITO

Observando-se os modelos de atrito que poderiam representar o atrito no eixo de acionamento do motor estudados, nota-se a dependência da obtenção da velocidade angular em suas equações (dq/dt ou $d\theta/dt$). Logo, uma

das preocupações para se anular o atrito no eixo motriz da barra deve ser a estimativa, via modelagem teórica, ou medição, via placa de aquisição de dados, da velocidade de giro do eixo.

2.2.1. MODELO DE ATRITO LINEAR DE UM MOTOR DE CORRENTE CONTÍNUA

O modelo matemático que caracteriza o comportamento de um motor CC com ímã permanente pode ser descrito tipicamente pelas seguintes equações (vide referência [2]):

$$\begin{aligned}T &= K_a i_a \\v &= R_a i_a + L_a \frac{di_a}{dt} + e_b \\e_b &= K_b \dot{q}_m \\q_m &= r q\end{aligned}$$

em que K_a = constante do motor (Nm/A)

R_a = resistência de armadura (Ω)

L_a = indutância de armadura (H)

K_b = constante de força contra-eletromotriz (Vs/rad)

T = torque no eixo do motor (Nm)

i_a = corrente de armadura (A)

e_b = força contra-eletromotriz (V)

q_m = posição angular do eixo do motor (rad)

q = posição angular do eixo da carga (rad)

r = relação de redução das engrenagens

v = voltagem de armadura (V)

T_l = torque imposto pela carga

Por outro lado, a equação de movimento para o sistema é:

$$J_m \ddot{q}_m = T - f_m(\dot{q}_m) - \frac{T_l}{r}$$

em que T_l é o torque de carga aplicado depois de engrenagens sobre o eixo da carga, J_m é a inércia do rotor e $f_m(dq_m/dt)$ é o torque devido à fricção do rotor com seus suportes.

Do ponto de vista dos sistemas dinâmicos, o motor CC pode ser visto como um dispositivo cuja entrada é a voltagem v e a saída é o torque T a ser aplicado depois da caixa de engrenagens. Para a obtenção do modelo dinâmico que relaciona v com T , manipulam-se as equações acima, obtendo-se:

$$\frac{K_a}{rR_a}v = \frac{L_a}{rR_a} \frac{d}{dt} \ddot{q} + \left(J_m + \frac{L_a}{R_a} \frac{\delta f_m(r\dot{q})}{\delta(r\dot{q})} \right) \ddot{q} + \frac{1}{r} f_m(r\dot{q}) + \frac{K_a K_b}{r} \dot{q} + \frac{T}{r^2} + \frac{L_a}{r^2 R_a} \dot{T}$$

Em muitas aplicações, tradicionalmente a indutância de armadura L_a é desprezível, de tal forma que a equação acima toma a seguinte forma:

$$\frac{K_a}{rR_a}v = J_m \ddot{q} + \frac{1}{r} f_m(r\dot{q}) + \frac{K_a K_b}{r} \dot{q} + \frac{T}{r^2}$$

No caso do modelo linear, como $f_m(dq_m/dt) = f_m dq_m/dt$, temos:

$$\frac{K_a}{rR_a}v = J_m \ddot{q} + \left(f_m + \frac{K_a K_b}{r} \right) \dot{q} + \frac{T}{r^2}$$

2.2.2. MODELO DE MOTOR CC COM ATRITO NÃO-LINEAR

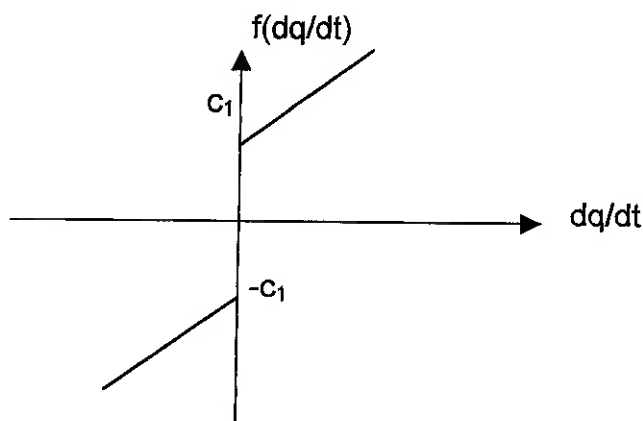


Figura 2: fricção não-linear

Um modelo mais realista para caracterizar a fricção no motor é o seguinte:

$$f_m(\dot{q}_m) = f_m \dot{q}_m + c_1 \sin(\dot{q}_m)$$

em que f_m e c_1 são constantes positivas.

Portanto, a relação entre voltagem aplicada na armadura do motor e torque exercido na carga fica:

$$\frac{K_a}{rR_a} v = J_m \ddot{q} + \left(f_m + \frac{K_a K_b}{r}\right) \dot{q} + \frac{c_1}{r} \sin(r \dot{q}) + \frac{T}{r^2}$$

2.2.3. STICK-SLIP FRICTION

Stick-slip friction está presente, em algum grau, em quase todos os atuadores e mecanismos e é quase sempre responsável por limitações de performance. Sua modelagem pode ser considerada bastante difícil devido ao seu comportamento fortemente não-linear quando a velocidade relativa das superfícies de contato tende a zero.

Um modelo que poderia ser utilizado (Karnopp [3]) é representado no gráfico abaixo:

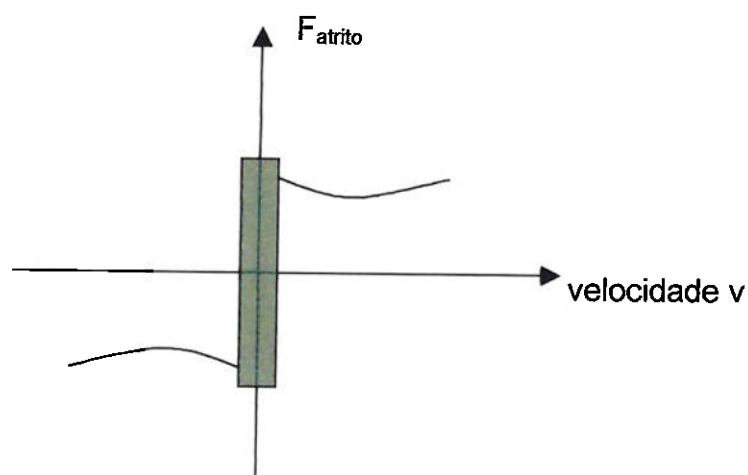


Figura 3: stick-slip friction

Quando o sistema escorrega (slips), a velocidade relativa das superfícies de contato em estudo é diferente de zero e a força de atrito é uma função de v .

Em $v=0$ ou suas proximidades, o sistema “adere” (sticks) e a força de atrito assume valores determinados por outros elementos do sistema. Se a força de atrito excede um determinado “patamar de escape”, o sistema volta para o “modo de escorregamento”.

3. ANÁLISE DO PROBLEMA UTILIZANDO O MODELO DE ATRITO VISCOSO

O primeiro modelo de atrito testado experimentalmente foi o de atrito viscoso onde tem-se: $F_{\text{atrito}} = C \dot{q} / dt$.

O primeiro passo desta análise consistiu em determinar analiticamente a função torque $T(t)$ necessária para se executar uma manobra da barra considerada rígida. Para tanto utilizou-se a função torque simplificada, dada pela equação:

$$I\ddot{\theta} + B\dot{\theta} = T(t)$$

Verificou-se que para uma entrada degrau de posição angular θ é necessária uma função triangular para a velocidade angular ω dada por:

$$\begin{aligned} \omega(t) &= 0 & 0 \leq t \leq a \\ \frac{t-a}{b} & & a \leq t \leq a+b \\ \frac{t-a}{b} - \frac{2(t-a-b)}{b} & & a+b \leq t < a+2b \\ \frac{t-a}{b} - \frac{2(t-a-b)}{b} + \frac{t-a-2b}{b} & & a+2b \leq t \end{aligned}$$

Utilizando transformada de Laplace, obtém-se:

$$\omega(s) = \frac{e^{-as}}{bs^2} - \frac{2e^{-(a+b)s}}{bs^2} + \frac{e^{-(a+2b)s}}{bs^2}$$

$$\tau(s) = (sI + B)\omega(s)$$

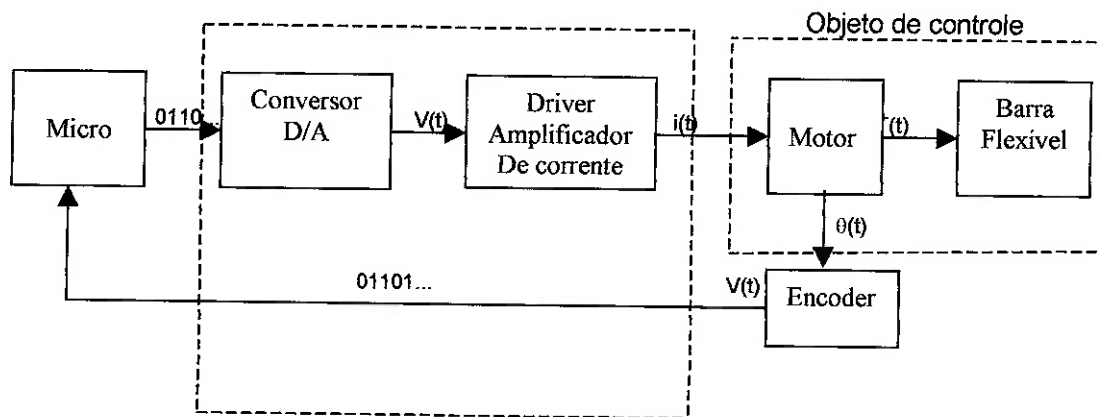
Finalmente, no domínio do tempo:

$$T(t) = \frac{I}{b} [u(t-a) - 2u(t-a-b) + u(t-a-2b)] + \frac{B}{b} [r(t-a) - 2r(t-a-b) + r(t-a-2b)]$$

onde $u(t)$ e $r(t)$ são respectivamente: funções degrau e rampa unitária

A função torque encontrada acima é implementada por software e acoplada à seguinte montagem experimental:

Placa de aquisição



Os resultados obtidos utilizando este modelo de atrito se encontram em anexo.

Verifica-se que o formato da curva $\theta(t)$ obtida experimentalmente apresenta semelhança com o esperado. Porém, percebem-se ainda muitos pontos em que o torque do motor não deslocou a estrutura angularmente, ou seja, percebe-se a presença de stick slip friction.

4. MESMA ANÁLISE COM A INTRODUÇÃO DE DOIS MODOS DE VIBRAR

Sugeriu-se que os erros verificados acima se deveram ao fato de não ser considerado, na função torque, os dois primeiros modos de vibrar da estrutura.

Iniciou-se então, o processo de determinação de uma função torque que possuísem estas parcelas.

Para tanto foi necessária uma análise matricial do problema.

Seja o objeto de controle modelado pelas seguintes equações:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

em que as matrizes x e B são iguais às mostradas no item 2.1. Na matriz A , porém, deve ser acrescentado a componente de atrito viscoso como mostra abaixo:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -737.5 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -7782.9 & -c_2 \end{pmatrix}$$

em que B é o coeficiente de atrito viscoso e I_{total} o momento de inércia total do objeto de controle (Motor + Barra).

A variável de estado de interesse é ω , pois a partir dela é possível se determinar a função torque necessária para se ter uma resposta a um degrau de posição que seja linear. Assim faz-se a matriz C de forma que:

$$y = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \\ \eta_1 \\ \dot{\eta}_1 \\ \eta_2 \\ \dot{\eta}_2 \end{pmatrix}$$

Sendo que $y(k)$ discretizado no tempo é equivalente a:

$$y(k) = \sum_{i=1}^k h(k-i)u(i)$$

Sendo h a resposta do sistema a um impulso retangular.

Para determinar a sequência $h(k)$, basta resolver a equação diferencial :

$$\dot{x} = Ax + Bu$$

fazendo $u(k)$ uma entrada retangular unitária. Desta forma, tenho y, uma vez que $y=Cx$.

A todo instante, a velocidade angular da viga pode ser avaliada, consequentemente tem-se uma estimativa do torque proveniente por atrito.

Esta estimativa pode ser adicionada ao torque de referência. A esta técnica dá-se o nome de feedforward.

Usando esta técnica para a implementação do modelo de atrito, eliminam-se eventuais erros que poderiam surgir se o modelo fosse feito por meio de uma realimentação negativa (feedback). Erros estes inerentes a conflitos de parâmetros do modelo de atrito e da malha de controle, onde invariavelmente utiliza-se o feedback.

5. MODIFICAÇÕES DE HARDWARE

Ao utilizarmos uma bancada previamente desenvolvida, não necessariamente para o mesmo fim, deparamo-nos com alguns problemas de adaptação do equipamento ao nosso estudo específico, além do fato de todo o hardware ter ficado desmontado e inativo durante algum tempo. Apresentam-se, a seguir, as modificações mais significativas efetuadas.

5.1. Colocação de Nova Barra

Foi necessária a substituição da barra utilizada anteriormente na bancada, cujas características eram:

- material: alumínio;
- comprimento: 0,70 m;
- largura: 0,0254 m;
- espessura: 0,001 m;

por uma barra “lisa”, isto é, que não tivesse extensômetros a ela ligados (utilizados em experimentos anteriores). As características da nova barra, que provocam alterações na modelagem das equações diferenciais que governam seu movimento, eram:

- material: alumínio;
- comprimento: 0,67 m;
- largura: 0,0238 m;
- espessura: 0,002 m;

5.2. Montagem do Conector DB25

Para que se efetuasse a conexão do encoder de posição, foi necessária a montagem de um conector DB25 à entrada digital da placa conversora AD/DA CAD 12/36.

O encoder utilizado é do tipo incremental e utiliza alimentação fornecida pela fonte do computador pelos pinos 20(Vcc) e 19(Gnd) do conector 5 da placa AD/DA. Além destes, utiliza-se dois bits do canal digital 0 da placa como entradas digitais dos canais A e B, pinos 12 e 11 respectivamente do mesmo conector 5. Com isto, a leitura do encoder é feita diretamente sem a necessidade de conversão por dispositivos intermediários, como contadores. Isso é possível devido a recursos de interrupções e escrita direta em memória (DMA) da placa, que possibilitou amostragem contínua do sinal do encoder sem ocupar a CPU do microcomputador.

5.3. Ajuste de Offset

Nos testes de funcionamento da placa conversora, foram medidos os sinais entregues ao motor e que eram gerados através de rotinas feitas em turbo pascal (vide apêndices). No entanto, percebeu-se que havia um *offset* do sinal em relação ao que era gerado, ou seja, havia um sinal maior que o pretendido, sendo que o sinal era não nulo mesmo quando não se estava enviando nenhum sinal para a placa. Tentou-se fazer o ajuste deste *offset* através da alteração nos potenciômetros da própria placa, medida que se revelou ineficaz.

6. MODIFICAÇÕES DE SOFTWARE

6.1. *Teste dos Canais Utilizados*

Sendo a conversão AD/DA realizada pela placa CAD 12/36 fundamental para a geração de sinais e aquisição dos dados do encoder, testou-se o seu funcionamento. Isto foi feito através da utilização do software `seno4.pas`, que gerava uma senóide. Conforme verificação através do osciloscópio, tinha-se um sinal discretizado (digital) que reproduzia de fato uma curva de seno.

Por meio deste software, foi possível também verificar o funcionamento do amplificador LM12 que drenava corrente suficiente para o disparo do motor que movimentava a barra flexível. Por meio de um resistor de 1 Ohm em série com o motor, visualizou-se no osciloscópio a tensão sobre este que pela lei de Ohm seria numericamente igual à corrente que passava pelo motor. Este artifício foi usado posteriormente para se estimar o torque do motor entregue ao seu eixo.

6.2. *Teste do Modo de Expansão da Placa CAD 12/36*

Foi detectado ainda um problema para adaptar completamente o hardware às necessidades da simulação proposta: determinar o modo de expansão D/A em que a placa estava configurada.

Esta configuração é feita por hardware, ou seja, por meio de jumpers dispostos adequadamente. Como a placa já estava conectada internamente ao slot do microcomputador PC, não era possível o acesso visual da mesma para verificação da ligação dos jumpers. Optou-se então, por construir uma pequena rotina em pascal considerando que a conversão da placa é de 12 bits deslocados de 4 à esquerda: 4 bits menos significativos no byte A, endereço base+8 e os oito restantes, MSB, no byte B, endereço base+9. Com uma palavra (16 bits) que poderia ser modificada por teclado em tempo real,

disposta adequadamente nesses endereços, foi possível por meio de um osciloscópio verificar que o número 2048 correspondia a um nível analógico convertido de 0 volts, 4096 a um nível de 5 volts 0 a um de -5 volts. Portanto a conversão era do tipo Binário/Bipolar.

6.3. Software para Geração do Torque

Para a geração do sinal à entrada do motor que levaria à geração do torque desejado, adaptou-se um programa previamente desenvolvido pelo professor Raul G. Lima, *slew6.pas*, presente no apêndice A.

Feito o debug das rotinas de inicialização da placa e leitura dos dados do encoder, foi necessário modificar a forma de onda de saída para a geração de um degrau de torque positivo e negativo, ambos com intervalos de tempos iguais.

Inicialmente, o programa deveria gerar um sinal retangular que reproduziria uma seqüência de vários degraus pretendidos para se analisar o sinal da corrente na saída do amplificador e consequentemente visualizar o formato do torque.

Feito esta primeira experiência, verificou-se o melhor tempo da manobra com o qual se conseguiria obter uma ótima retratação do atrito no gráfico de posição angular da barra.

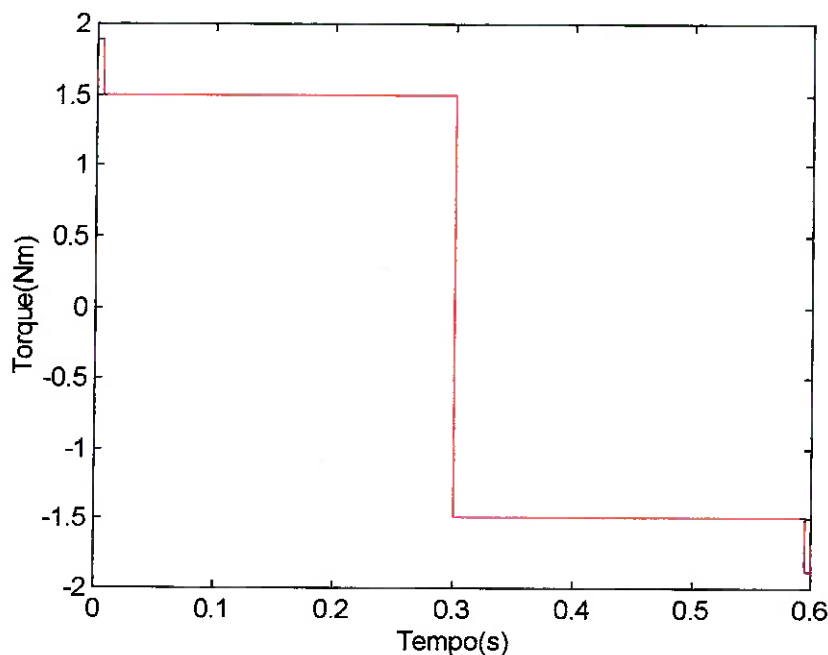
7. PROCEDIMENTO EXPERIMENTAL E RESULTADOS

7.1. Determinação da Entrada

Para a identificação de um modelo de atrito no eixo do motor de corrente contínua, era necessário avaliar a resposta do sistema (posição angular e velocidade angular da barra) face à entrada de torque fornecida, comparando a resposta obtida através de simulações no MatLab com a resposta medida pelo encoder e gravada no arquivo `arq_posM.dat` (apêndice C). Para a determinação e ajuste experimental do torque elétrico entregue ao motor, adotou-se o seguinte procedimento:

1. Medição da corrente média entregue pelo amplificador ao motor;
Utilizou-se um osciloscópio para a medida da mesma e, através de um resistor de $1\ \Omega$ em série com o motor, pôde-se medir a tensão sobre ele.
2. Calculou-se o torque fornecido ao eixo do motor utilizando-se sua constante de torque K_t ($K_t = 0,1\ \text{Nm/A}$).

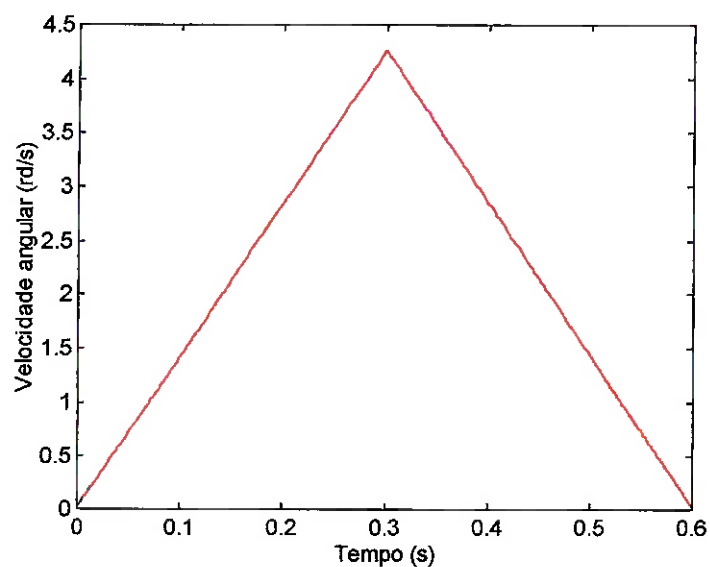
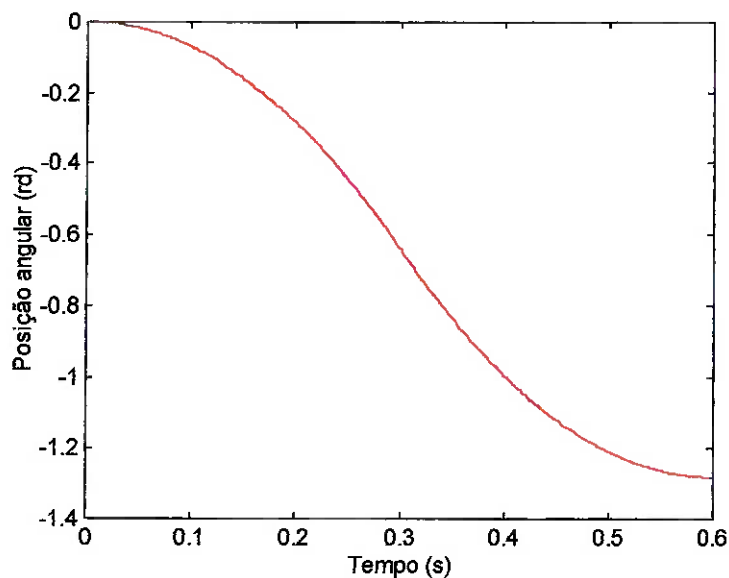
Conhecido o torque gerado à entrada, criou-se a rotina `torque2.m` em MatLab para a geração de um degrau positivo seguido de um degrau negativo de torque, como mostra a figura:



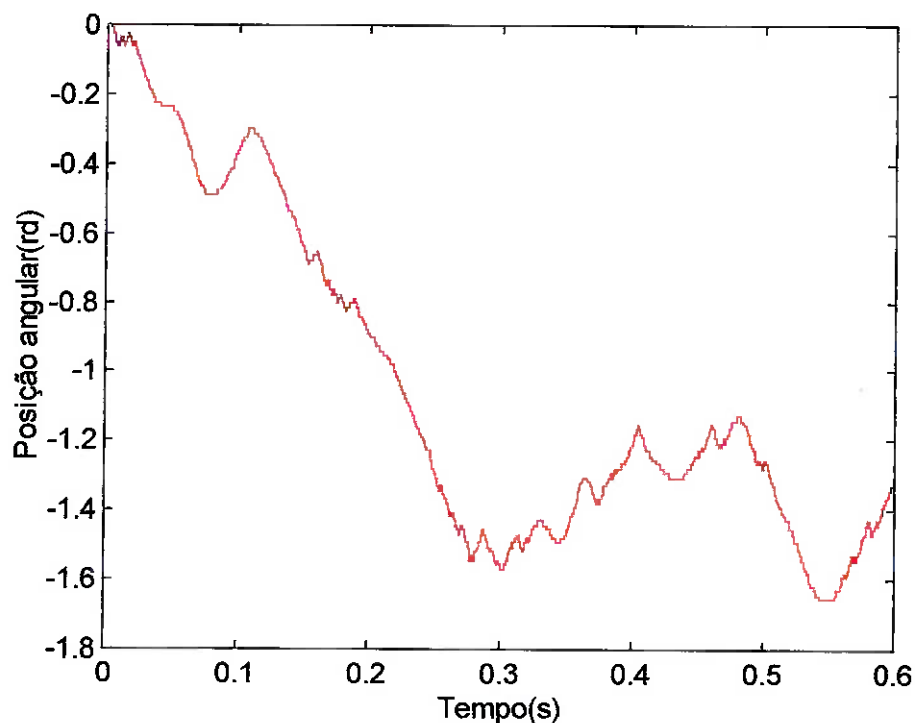
A rotina torque2.m encontra-se no apêndice B.

Deve-se ressaltar que é possível constatar na figura a inclusão do *overshooting* percebido através de medições no osciloscópio. Por não termos trabalhado com um osciloscópio de memória, não foi possível estimarmos quantitativamente com precisão este fenômeno, de forma que se considerou como sendo de 25%.

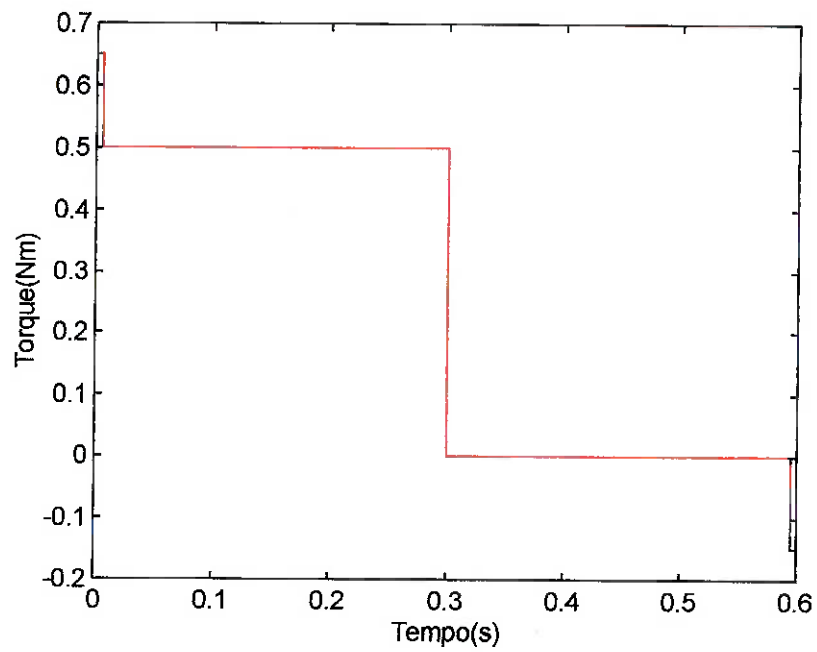
As saídas posição e velocidade angular da barra obtidas para essa entrada foram:



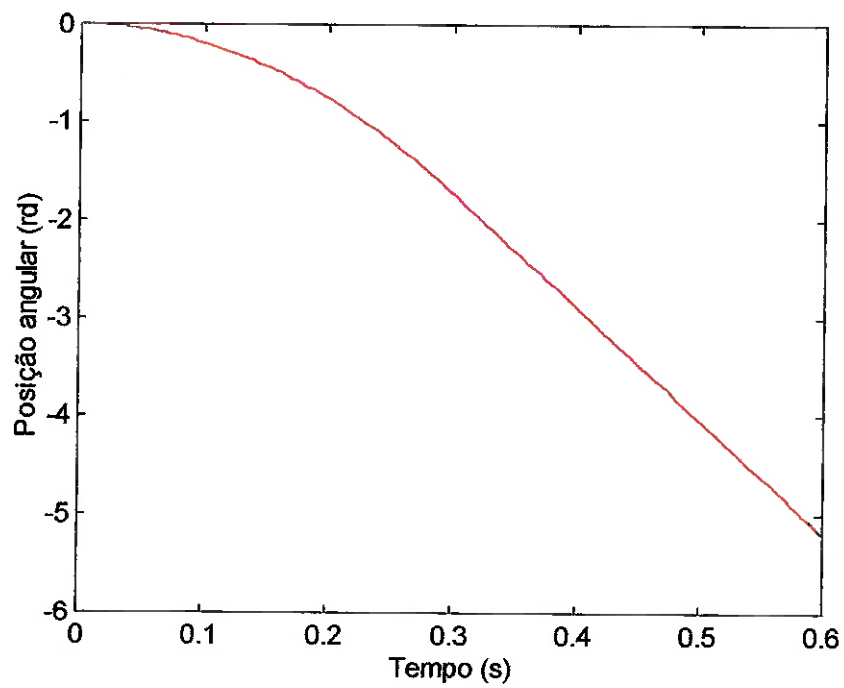
Verifica-se que a resposta simulada no MatLab para essa entrada teórica (degrau de torque) corresponde a uma aceleração da rotação barra em um sentido seguida de uma desaceleração. Entretanto, a saída experimental obtida foi a seguinte:

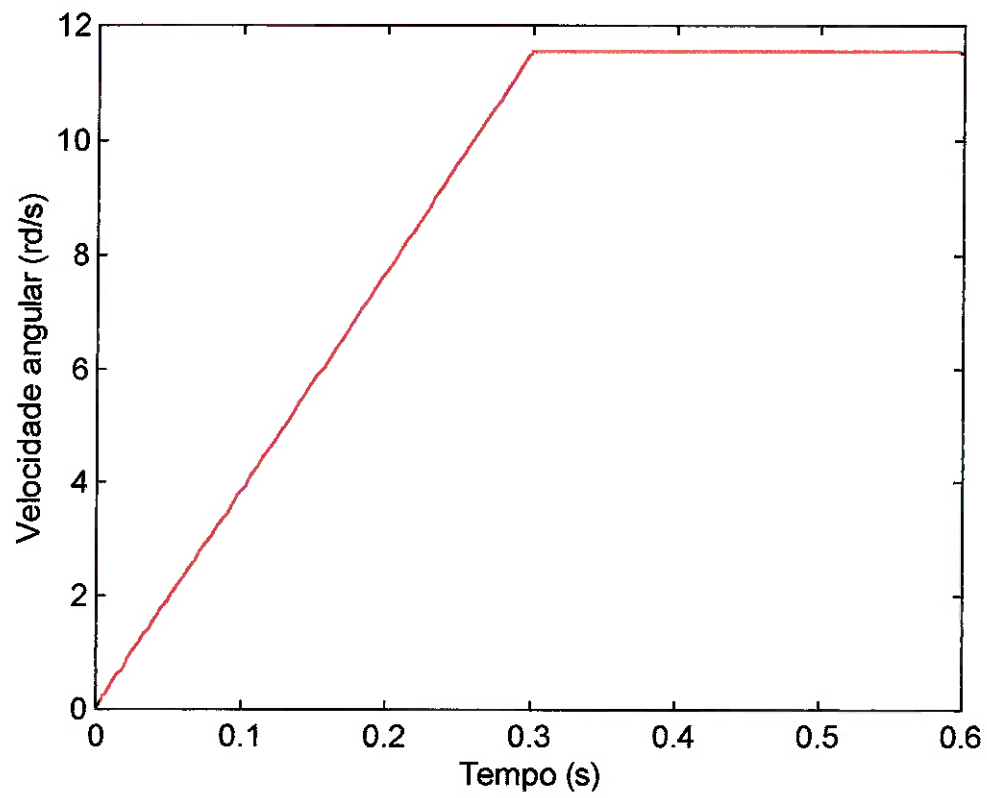


Há um grande número de oscilações na curva obtida, dificultando a análise dos resultados, apesar de mostrar claramente o efeito de um atrito não-linear na posição angular em função do tempo (percebe-se os pontos em que o eixo 'sticks', ou seja, pára de rotacionar). Por isso, adotou-se uma outra entrada, com apenas um degrau de torque positivo, conforme a rotina torque.m e a figura:



Para esta entrada, os novos resultados obtidos foram:





8. IMPLEMENTAÇÃO DO MODELO DE ATRITO

Para a simulação de um atrito do tipo *stick-slip*, usou-se o algoritmo:

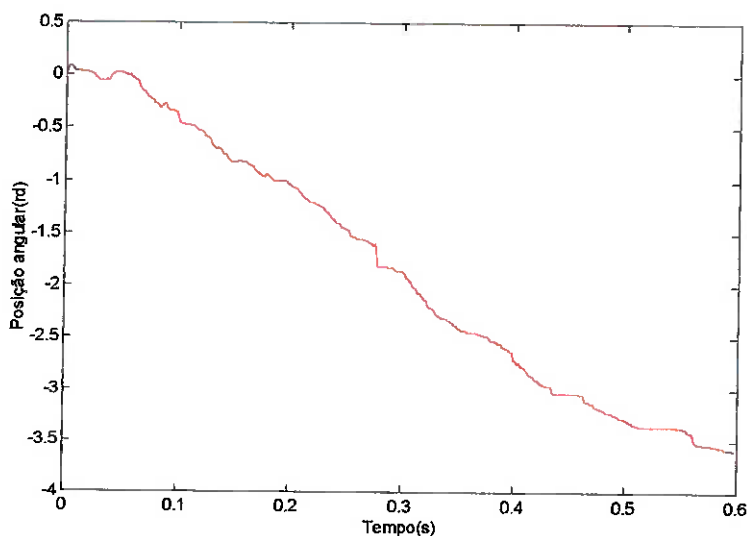
- se $w \sim 0$ então
 - Torque resultante=0
- senão
 - Torque resultante=Torque motor - $c \cdot w$

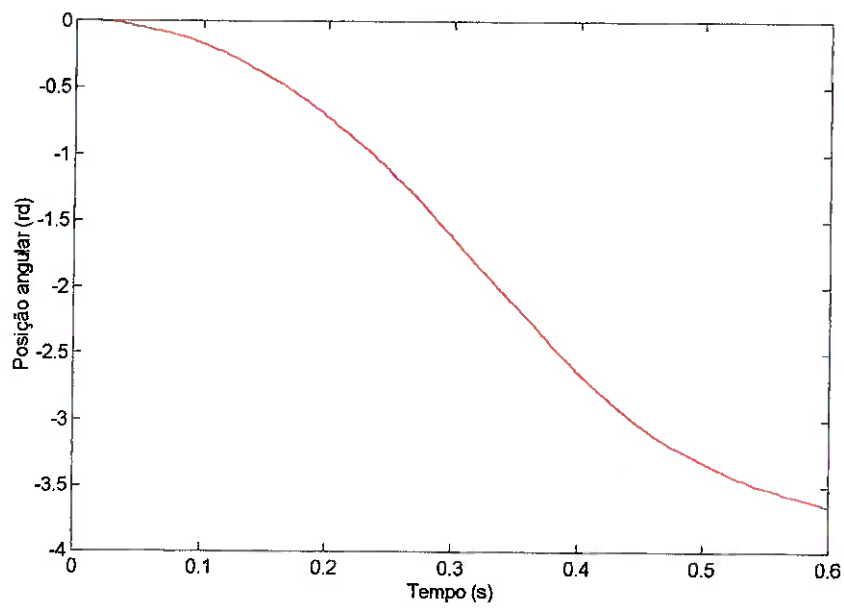
em que w = velocidade angular da barra
 c =constante de atrito viscoso

A implementação de um modelo de atrito que correspondesse aos resultados obtidos experimentalmente foi feita com o auxílio da rotina `f5.m`, feita em MatLab e que se encontra no apêndice B. Tal rotina contém a modelagem da barra flexível, na qual mudavam-se os parâmetros de forma que o modelo se aproximasse do caso real (as matrizes agora apresentadas diferem das contidas na seção (2)). Dessa forma, foi possível fazer uma estimativa do valor de c . A principal constatação ficou por conta do fato da constante c ser variável. Assim, assumiu-se um valor de c para cada um dos períodos de tempo considerados.

$C=0.7$ para t pertencente a $[0.0, 0.4s[$

$C=7.5$ para t pertencente a $[0.4, 0.6s]$





Os gráficos acima mostram, então, as saídas obtidas com os dados experimentais (obtidos via medição do encoder) e com a simulação do modelo considerado, respectivamente.

9. BIBLIOGRAFIA

- [1]. Gildin, E., Moscato, L.A. e Gonzalez-Lima, R.; Desenvolvimento de um Controlador Adaptativo para Manipuladores Flexíveis com Incertezas de Cargas
- [2]. Martinez, R.K.; Control de Movimiento de Robots Manipuladores
- [3]. Karnop, D.; Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems; Journal of Dynamic Systems, Measurement, and Control, vol. 107, março 1995.
- [4]. Tworzydło, W.W.; Becker, E.B. e Oden, J.T.; Numerical Modeling of Friction-Induced Vibrations and Dynamic Instabilities; ASME Reprint N° AMR 147, 1994.
- [5]. Wood, S.; Turbo Pascal – Guia do Usuário; McGraw-Hill, 1987.
- [6]. Departamento de Documentação ; Manual do Usuário e de Referência da Placa Conversora A/D CAD 12/36; Linx Tecnologia Eletrônica Ltda.;; Engenharia de Hardware; versão 1.2, 1993.

APÊNDICE A – Softwares Utilizados (Turbo Pascal)

A.1. Slew6.pas

```

Uses Dos,
    Crt;

{ ===== }
{ =          Declaracao de Constantes          = }
{ ===== }

Const
    Placa      = $380; { Endereco base da placa CAD12/36 }
    IntUsada   = 3;    { Interrupcao utilizada           }

    Limite     = 0;    { End. secundario da CAD12/36 - Reg. de
    }
    SecPonteiro = 1;    { End. secundario da CAD12/36 - Ponteiro
Memoria      }
    SecComAD    = 2;    { End. secundario da CAD12/36 - Comando
Conv. A/D    }
    SecRM       = 3;    { End. secundario da CAD12/36 - Reg. de
Modo         }
    SecMemoria  = 4;    { End. secundario da CAD12/36 - Escrita
Memoria      }
    SecAutoCal  = 6;    { End. secundario da CAD12/36 - Auto
Calibracao   }
    IntAck      = $20; { Registrador de reconhecimento de
interrupcao  }
    IntMask     = $21; { Registrador da mascara de interrupcoes
}

    CadCtr0     = Placa;
    CadCtrl1    = Placa + 1;
    CadCtrl2    = Placa + 2;
    CadModo     = Placa + 3;
    CadStatus   = Placa + 3;
    ByteA       = Placa + 4;
    ByteB       = Placa + 5;
    CadEnd      = Placa + 4;
    CadDado     = Placa + 5;
    CadESD0     = Placa + 6;
    CadESD1     = Placa + 7;

{ ===== }
{ =          Declaracao de Variaveis          = }
{ ===== }

Var Contador : integer;
    dadoant  : byte;
    Divisor  : word;

    Coll, Lin1: integer;
    VetLeit   : array[0..15] of longint; { vetor com leituras}
    Ch        : char;

```

Flg : boolean;

```
{ ===== }
{ =          Rotina  IniciaEnderecoHardware          = }
{ ===== }
```

Procedure IniciaEnderecoHardware;

begin
end;

```
{ ===== }
{ =          Rotina  EscreveRegSecundario          = }
{ ===== }
```

Procedure EscreveRegSecundario (EndReg, Dado: byte);

begin
 Port [CadEnd] := EndReg;
 Port [CadDado] := Dado;
end;

```
{ ===== }
{ =          Rotina  EscreveMemCanais          = }
{ = OBS.: Os Bits do dado sao invertidos pois o = }
{ = chip utilizado e o 74LS289                 = }
{ ===== }
```

Procedure EscreveMemCanais (Posicao, Dado: byte);

begin
 EscreveRegSecundario (SecPonteiro, Posicao); { Endereco da
Memoria de Canais }
 EscreveRegSecundario (SecMemoria, not(Dado)); { Canal e Ganho
correspondente }
end;

```
{ ===== }
{ =          Rotina  AutoCalibracao          = }
{ ===== }
```

Procedure AutoCalibracao;

begin
 EscreveRegSecundario (SecAutoCal, 0);
 Delay (2000);
end;

```
{ ===== }
{ =          Rotina  LimpaFIFO          = }
{ ===== }
```

Procedure LimpaFIFO;
 Var i, Dado: integer;

begin

```

    for i := 1 to 16 do
    begin
        Dado := Port [ByteB];
        Dado := Port [ByteA];
    end;
end;

```

```

{ ===== }
{ =          Rotina ProgramaTimer          = }
{ ===== }

```

```

Procedure ProgramaTimer (Contador, Modo: byte; Valor: word);
    Var EndCtr: word;

```

```

begin
    EndCtr := Placa + Contador;
    Port [CadModo] := Contador * 64 + 48 + Modo * 2;
    Port [EndCtr] := Lo (Valor);
    Port [EndCtr] := Hi (Valor);
end;

```

```

{ ===== }
{ =          Rotina InibeINT          = }
{ ===== }

```

```

Procedure InibeINT;
    Var Mask : byte;

```

```

begin
    Mask := Port [IntMask];
    Port [IntMask] := Mask or ($01 shl IntUsada);
end;

```

```

{ ===== }
{ =          Rotina HabilitaINT          = }
{ ===== }

```

```

Procedure HabilitaINT;
    Var Mask : byte;

```

```

begin
    Mask := Port [IntMask];
    Port [IntMask] := Mask and not ($01 shl IntUsada);
    Port [IntAck] := $60 + IntUsada; { limpa reg. de interrupcoes }
    dadoant:=port[placa+6];
    contador:=0;
end;

```

```

{ ===== }
{ =          Rotina InibeGateComINT          = }
{ ===== }

```

```

Procedure InibeGateComINT;

```

```

begin
    EscreveRegSecundario (SecRM, $0c);
                                { inibe timer }

```



```

{ usa timer 0 }
{ interrupcao ao final do burst }
{ modo burst ligado }
{ modo DMA desligado }
end;

```

```

{ ===== }
{ = Rotina HabilitaGateComINT = }
{ ===== }

```

Procedure HabilitaGateComINT;

begin

```

  EscreveRegSecundario (SecRM, $4c);
  { habilita timer }
  { usa timer 0 }
  { interrupcao ao final do burst }
  { modo burst ligado }
  { modo DMA desligado }

```

end;

```

{ ===== }
{ = Rotina ProgramaMemoriaCanais = }
{ ===== }

```

Procedure ProgramaMemoriaCanais;

Const

```

  G1 = 7*16; { constantes para programar o ganho }
  G2 = 6*16;
  G5 = 5*16;
  G100 = 3*16;

```

```

  Bipolar = $80; { faixa bipolar }

```

```

  Unipolar = $00; { faixa unipolar }

```

Var

```

  i: integer;

```

begin

```

  EscreveRegSecundario (SecLimite, $f); { Carrega 15 no Reg. Limite }

```

```

  EscreveMemCanais (0, Bipolar + G1+ 0); { memoria 0 com canal 0,
bipolar e ganho 1 }

```

```

  EscreveMemCanais (1, Bipolar + G2+ 1); { memoria 1 com canal 1,
bipolar e ganho 2 }

```

```

  EscreveMemCanais (2, Bipolar + G5+ 2); { memoria 2 com canal 2,
bipolar e ganho 5 }

```

```

  EscreveMemCanais (3, Bipolar + G1+ 3); { memoria 3 com canal 3,
bipolar e ganho 1 }

```

```

  EscreveMemCanais (4, Bipolar + G1+ 4); { memoria 4 com canal 4,
bipolar e ganho 1 }

```

```

  EscreveMemCanais (5, Bipolar + G1+ 5); { memoria 5 com canal 5,
bipolar e ganho 1 }

```

```

  EscreveMemCanais (6, Bipolar + G1+ 6); { memoria 6 com canal 6,
bipolar e ganho 1 }

```

```

  EscreveMemCanais (7, Unipolar+ G1+ 7); { memoria 7 com canal 7,
unipol. e ganho 1 }

```

```

  for i:= 8 to 15 do

```

```

    EscreveMemCanais (i, Bipolar+ G1+ i); { memoria i com canal i,
    unipol. e ganho 1 }
end;

```

```

{ ===== }
{ =                Rotina  TrataInt                = }
{ ===== }
const

```

```

    bitA=$1;
    bitB=$2;
    bitI=$80;

```

```

type
    position = record
        contador : Integer;
    end;

```

```

{$f+}
var times:longint;
    freq:extended;

```

```

Procedure TrataInt; interrupt;assembler;
asm
    mov dx,Cadstatus {limpa pedido de interrupcao da placa}
    in al,dx

```

```

    add word[times],1 {incrementa contador de tempo}
    adc word[times+2],0

```

```

    mov dx,placa+6
    in al,dx          { le porta}
    mov ah,al
    xor al,[dadoant]
    and al,3
    jz @NaoMudou
    and al,bitA      {alteracao em A => R=1}
    mov bl,ah
    and bl,bitA      {estado do Bit A}
    xor al,bl
    mov bl,ah
    and bl,bitB      {estado do Bit B}
    shr bl,1
    xor al,bl        {A xor B xor R}
    jz @diminui
    inc [Contador]
    jmp @aumentou {nome somente para nao usar outro}
@diminui:
    dec [Contador]
@aumentou:
    mov [dadoant],ah

```

```

@NaoMudou:

```

```

    mov al,$60+Intusada
    out IntAck,al      { limpando o Registrador de interrupcoes }
end;

```

```

{$f-}

```

```

var
  tk:word;
  tk8,tk9:byte;
  MUL,I,J: integer;
  k:char;
  oldpointer:pointer;
  amostra:array[0..255] of longint;
  po:byte;
  pos_dat: text;
  B,amp,temp,aux,PER,off:extended;
begin
  Divisor:= 1;          { A/D de 12 bits = 16 }
                        { A/D de 14 bits = 4 }
                        { A/D de 16 bits = 1 }
  IniciaEnderecoHardware; { Determina os enderecos de
I/O da placa CAD12/36 }
  EscreveRegSecundario (SecRM, 0); { Zera Registrador de Modo da
CAD12/36 }
  InibeINT; { Inibe a interrupcao usada
pela placa }

  freq:=1000; {Hz}

  ProgramaTimer (0, 2,round(2e6/freq)); { Programa Timer 0 no modo 2
e a 50 Hz = 2e6/4e4 }
  ProgramaTimer (1, 1,2); { Programa Timer 1 no modo 1
(nao gera pulsos) }
  ProgramaTimer (2, 1,2); { Programa Timer 2 no modo 1
(nao gera pulsos) }

  AutoCalibracao; { Comando de Auto-Calibracao
do Conversor A/D }
  ProgramaMemoriaCanais; { Programa a memoria de
canais }
  EscreveRegSecundario (SecLimite, 7); { Carrega numero de canais
menos 1 no Reg. Limite }
  EscreveRegSecundario (SecPonteiro, 0); { Zera Ponteiro da memoria de
canais }
  LimpaFIFO; { Limpa a memoria FIFO da
CAD12/36 }
  GetIntVec(IntUsada+8,Oldpointer); { Inicia vetor de interrupcao
}
  SetIntVec(IntUsada+8,@trataint); { observe que o vetor usado
e' IntUsada + 8 }

  HabilitaInt;
  I:= Port[CadStatus]; { limpa FF de pedido de interrupcao
}
  Ch:= 'A'; { Ch vai ser incrementado por TrataInt
}
  Flg:= false; { flag para indicar ocorrencia de
interrupcao }
  HabilitaGateComINT; { a partir deste ponto, as interrupcoes sao
ativadas }

  clrscr;
  directvideo:=true;
  checksnow:=false;
  po:=0;
  temp:=0;

```

```

PER:=0;
MUL:=1;
aux:=800;
off:=2040;
Assign(pos_dat,'c:\arq_pos.dat');
Rewrite(pos_dat);
writeln('*****APLICACAO DE IMPULSO
RETANGULAR*****');

{repeat
repeat
gotoxy(24,3);write(off:15:5); }

repeat {1 periodo do sinal}
writeln(pos_dat,temp,' ',contador,' ',tk);
temp:=times/freq;
if (temp<0.0) then
begin
tk:=round(off);
tk8:=Lo(tk shl 4);
port[placa+8]:=tk8;
tk9:=Hi(tk shl 4);
port[placa+9]:=tk9;
{ portw[placa+8]:=0+off;
tk:=off; }
end
else
if (temp<0.3) then
begin
tk:=round(off)+round(aux/2);
tk8:=Lo(tk shl 4);
port[placa+8]:=tk8;
tk9:=Hi(tk shl 4);
port[placa+9]:=tk9;
{ portw[placa+8]:=tk; }
end
else
if (temp<0.6) then
begin
tk:=round(off)-round(aux/2);
tk8:=Lo(tk shl 4);
port[placa+8]:=tk8;
tk9:=Hi(tk shl 4);
port[placa+9]:=tk9;
end

{ portw[placa+8]:=tk;}

else
begin
portw[placa+8]:=0+off;}
tk:=round(off);
tk8:=Lo(tk shl 4);
port[placa+8]:=tk8;
tk9:=Hi(tk shl 4);
port[placa+9]:=tk9;
end;
until temp>0.6;

InibeInt; { inibe a interrupcao }

```

```
SetIntVec(IntUsada+8,oldpointer);      { Restaura anterior }

end.
```

A.2. OFFSET

```
uses crt;
CONST placa=$380;
var off:extended;
    tk,tk12:word;
    tk8,tk9:word;
    k:char;
begin
port[placa+3]:=$00;
tk8:=0;
tk9:=0;
off:=2048;
ClrScr;
writeln('Offset      [up/down]      :');
repeat
    gotoxy(24,3);write(off:15:5);
    repeat
        tk:=round(off);
        tk8:=Lo(tk shl 4);
        port[placa+8]:=tk8;
        tk9:=Hi(tk shl 4);
        port[placa+9]:=tk9;
    until keypressed;
    k:=readkey;if k=#0 then k:=readkey;
    case k of
        'H':if off+10<=$7fff then off:=off+10 else off:=$7fff;
        'P':if off-10>=-$7fff then off:=off-10 else off:=0;
    end;
until k=#27;
end.
```

APÊNDICE B – Rotinas em MatLab

B.1. Rotina Torque2.m

```
%Torque elétrico fornecido ao eixo do motor
%Corrente na saída do amplificador medida pelo osciloscópio p/
%Degrau positivo e negativo de torque
%Torque=lout*Kt

n=1;
B=0;
for t=0:0.001:0.6
    if t<=0.005
        u=1.88;

    elseif t>0.005 & t<=0.3
        u=1.5;
    elseif t<0.595
        u=-1.5;
    else
        u=-1.88;
    end;
    Conjugado(n)=u;
    n=n+1;
end;
t=0:0.001:0.6;
figure;
plot(t,Conjugado,'r');
xlabel('Tempo(s)');
ylabel('Torque(Nm)');
```

B.2. Torque.m

```
%Torque elétrico fornecido ao eixo do motor, de grau positivo
%Corrente na saída do amplificador medida pelo osciloscópio
%Torque=lout*Kt

n=1;
B=0;
for t=0:0.001:0.6
    if t<=0.005
```

```

    u=0.65;
elseif t>0.005 & t<=0.3
    u=0.5;
elseif t<0.595
    u=0;
else
    u=-0.15;
end;
Conjugado(n)=u;
n=n+1;
end;
t=0:0.001:0.6;
figure;
plot(t,Conjugado,'r');
xlabel('Tempo(s)');
ylabel('Torque(Nm)');

```

B.3. f5.m

%Equações diferenciais para modelo de barra flexível
function xp=f(t,x)

```

if t<0.005
% u=0.65;
    u=20;

%Condição de atrito
if t<=0.0065
    A=[0 -1 28.0492 0 54.6318 0;
        0 0 0 0 0;
        0 0 0 1 0 0;
        0 0 -8275 0 0 0;
        0 0 0 0 1;
        0 0 0 0 -6474.7 0];

    B=[0;0;0;-28.0492;0;-54.6318];

else
    A=[0 -1 28.0492 0 54.6318 0;
        0 -0.7 0 0 0 0;
        0 0 0 1 0 0;
        0 0 -8275 0 0 0;
        0 0 0 0 1;
        0 0 0 0 -6474.7 0];

```

```
B=[0;76.5614;0;-28.0492;0;-54.6318];
end;

elseif t>0.005 & t<=0.3
    u=0.5;

    %Condição de atrito
    if t<=0.0113
        A=[0 -1 28.0492 0 54.6318 0;
            0 0 0 0 0 0;
            0 0 0 1 0 0;
            0 0 -8275 0 0 0;
            0 0 0 0 0 1;
            0 0 0 0 -6474.7 0];

        B=[0;0;0;-28.0492;0;-54.6318];

    else
        A=[0 -1 28.0492 0 54.6318 0;
            0 -0.35 0 0 0 0;
            0 0 0 1 0 0;
            0 0 -8275 0 0 0;
            0 0 0 0 0 1;
            0 0 0 0 -6474.7 0];

        B=[0;76.5614;0;-28.0492;0;-54.6318];
    end;

elseif t>0.3 & t<=0.595
    % u=-0.495;
    u=0;

    %Condição de atrito
    if t>0.4
        A=[0 -1 28.0492 0 54.6318 0;
            0 -7.5 0 0 0 0;
            0 0 0 1 0 0;
            0 0 -8275 0 0 0;
            0 0 0 0 0 1;
            0 0 0 0 -6474.7 0];

        B=[0;0;0;-28.0492;0;-54.6318];

    else
        A=[0 -1 28.0492 0 54.6318 0;
            0 -0.7 0 0 0 0;
            0 0 0 1 0 0;
            0 0 -8275 0 0 0;
```



```
0 0 0 0 0 1;  
0 0 0 0 -6474.7 0];
```

```
B=[0;76.5614;0;-28.0492;0;-54.6318];  
end;
```

```
else
```

```
u=-0.65;
```

```
% u=-200;
```

```
A=[0 -1 28.0492 0 54.6318 0;  
0 0 0 0 0 0;  
0 0 0 1 0 0;  
0 0 -8275 0 0 0;  
0 0 0 0 0 1;  
0 0 0 0 -6474.7 0];
```

```
B=[0;0;0;-28.0492;0;-54.6318];
```

```
end;
```

```
xp=A*x+B*u;
```

APÊNDICE C – Medições Obtidas

C.1. Arq_posM.dat

9.900000000000000E-0002	0	2040	2.200000000000000E-0001	-22	2440
1.000000000000000E-0001	6	2440	2.230000000000000E-0001	-22	2440
1.210000000000000E-0001	6	2440	2.230000000000000E-0001	-23	2440
1.210000000000000E-0001	6	2440	2.250000000000000E-0001	-25	2440
1.230000000000000E-0001	4	2440	2.270000000000000E-0001	-24	2440
1.240000000000000E-0001	3	2440	2.270000000000000E-0001	-23	2440
1.260000000000000E-0001	3	2440	2.280000000000000E-0001	-22	2440
1.270000000000000E-0001	3	2440	2.300000000000000E-0001	-25	2440
1.280000000000000E-0001	3	2440	2.320000000000000E-0001	-26	2440
1.290000000000000E-0001	2	2440	2.340000000000000E-0001	-28	2440
1.310000000000000E-0001	2	2440	2.350000000000000E-0001	-27	2440
1.330000000000000E-0001	2	2440	2.360000000000000E-0001	-27	2440
1.340000000000000E-0001	2	2440	2.380000000000000E-0001	-28	2440
1.350000000000000E-0001	2	2440	2.390000000000000E-0001	-29	2440
1.370000000000000E-0001	1	2440	2.410000000000000E-0001	-31	2440
1.380000000000000E-0001	1	2440	2.420000000000000E-0001	-37	2440
1.390000000000000E-0001	0	2440	2.520000000000000E-0001	-37	2440
1.410000000000000E-0001	-2	2440	2.540000000000000E-0001	-37	2440
1.430000000000000E-0001	-2	2440	2.550000000000000E-0001	-38	2440
1.430000000000000E-0001	-5	2440	2.580000000000000E-0001	-38	2440
1.560000000000000E-0001	-5	2440	2.580000000000000E-0001	-38	2440
1.580000000000000E-0001	-5	2440	2.590000000000000E-0001	-39	2440
1.590000000000000E-0001	-5	2440	2.610000000000000E-0001	-39	2440
1.600000000000000E-0001	-4	2440	2.620000000000000E-0001	-39	2440
1.620000000000000E-0001	-4	2440	2.630000000000000E-0001	-40	2440
1.640000000000000E-0001	-5	2440	2.650000000000000E-0001	-41	2440
1.740000000000000E-0001	-1	2440	2.670000000000000E-0001	-42	2440
1.770000000000000E-0001	0	2440	2.690000000000000E-0001	-42	2440
1.780000000000000E-0001	1	2440	2.700000000000000E-0001	-43	2440
1.800000000000000E-0001	2	2440	2.710000000000000E-0001	-44	2440
1.810000000000000E-0001	2	2440	2.730000000000000E-0001	-47	2440
1.820000000000000E-0001	2	2440	2.750000000000000E-0001	-47	2440
1.830000000000000E-0001	2	2440	2.750000000000000E-0001	-48	2440
1.850000000000000E-0001	1	2440	2.770000000000000E-0001	-49	2440
1.860000000000000E-0001	1	2440	2.800000000000000E-0001	-53	2440
1.870000000000000E-0001	0	2440	2.810000000000000E-0001	-54	2440
1.900000000000000E-0001	-1	2440	2.830000000000000E-0001	-56	2440
1.920000000000000E-0001	0	2440	2.840000000000000E-0001	-55	2440
1.930000000000000E-0001	-1	2440	2.860000000000000E-0001	-56	2440
1.940000000000000E-0001	-2	2440	2.870000000000000E-0001	-56	2440
1.950000000000000E-0001	-4	2440	2.890000000000000E-0001	-59	2440
1.970000000000000E-0001	-4	2440	2.970000000000000E-0001	-60	2440
1.980000000000000E-0001	-5	2440	2.990000000000000E-0001	-60	2440
2.070000000000000E-0001	-9	2440	3.000000000000000E-0001	-62	2040
2.080000000000000E-0001	-11	2440	3.020000000000000E-0001	-64	2040
2.100000000000000E-0001	-12	2440	3.030000000000000E-0001	-66	2040
2.110000000000000E-0001	-13	2440	3.050000000000000E-0001	-66	2040
2.130000000000000E-0001	-15	2440	3.060000000000000E-0001	-66	2040
2.140000000000000E-0001	-17	2440	3.070000000000000E-0001	-66	2040
2.160000000000000E-0001	-18	2440	3.100000000000000E-0001	-66	2040
2.170000000000000E-0001	-19	2440	3.100000000000000E-0001	-65	2040
2.180000000000000E-0001	-20	2440	3.120000000000000E-0001	-65	2040

8.030000000000000E-0001 -268 2040
8.040000000000000E-0001 -268 2040
8.050000000000000E-0001 -268 2040
8.070000000000000E-0001 -268 2040
8.090000000000000E-0001 -268 2040
8.090000000000000E-0001 -268 2040
8.110000000000000E-0001 -268 2040
8.130000000000000E-0001 -268 2040
8.130000000000000E-0001 -268 2040
8.170000000000000E-0001 -268 2040
8.180000000000000E-0001 -268 2040
8.200000000000000E-0001 -268 2040
8.210000000000000E-0001 -268 2040
8.220000000000000E-0001 -268 2040
8.240000000000000E-0001 -269 2040
8.330000000000000E-0001 -269 2040
8.350000000000000E-0001 -269 2040
8.360000000000000E-0001 -269 2040
8.380000000000000E-0001 -270 2040
8.390000000000000E-0001 -270 2040
8.400000000000000E-0001 -271 2040
8.420000000000000E-0001 -272 2040
8.440000000000000E-0001 -272 2040
8.440000000000000E-0001 -273 2040
8.460000000000000E-0001 -278 2040

8.540000000000000E-0001 -280 2040
8.580000000000000E-0001 -280 2040
8.600000000000000E-0001 -281 2040
8.610000000000000E-0001 -281 2040
8.630000000000000E-0001 -281 2040
8.640000000000000E-0001 -282 2040
8.650000000000000E-0001 -282 2040
8.670000000000000E-0001 -282 2040
8.740000000000000E-0001 -282 2040
8.760000000000000E-0001 -282 2040
8.770000000000000E-0001 -283 2040
8.780000000000000E-0001 -283 2040
8.800000000000000E-0001 -283 2040
8.820000000000000E-0001 -283 2040
8.830000000000000E-0001 -284 2040
8.850000000000000E-0001 -284 2040
8.850000000000000E-0001 -284 2040
8.870000000000000E-0001 -284 2040
8.880000000000000E-0001 -285 2040
8.910000000000000E-0001 -285 2040
8.920000000000000E-0001 -285 2040
8.930000000000000E-0001 -285 2040
8.950000000000000E-0001 -285 2040
8.960000000000000E-0001 -286 2040
8.970000000000000E-0001 -286 2040