

**Escola Politécnica
Da
Universidade de São Paulo**

**Departamento de Engenharia de Energia e
Automação Elétricas e Departamento de
Telecomunicações e Controle**



**Estudo e Implementação de Sistema de Controle Ótimo
para Elevadores**

**Marcelo Tarkieltaub
Felipe Vivacqua Pinto**

Projeto de Formatura 2001

**Escola Politécnica
Da
Universidade de São Paulo**

Departamento de Engenharia de Energia e
Automação Elétricas e Departamento de
Telecomunicações e Controle



Projeto de Formatura 2001

Estudo e Implementação de Sistema de Controle Ótimo
para Elevadores

Alunos: Marcelo Tarkieltaub
Felipe Vivacqua Pinto
Orientador: Clóvis Goldenberg
Coordenador: Luiz Cláudio R. Galvão

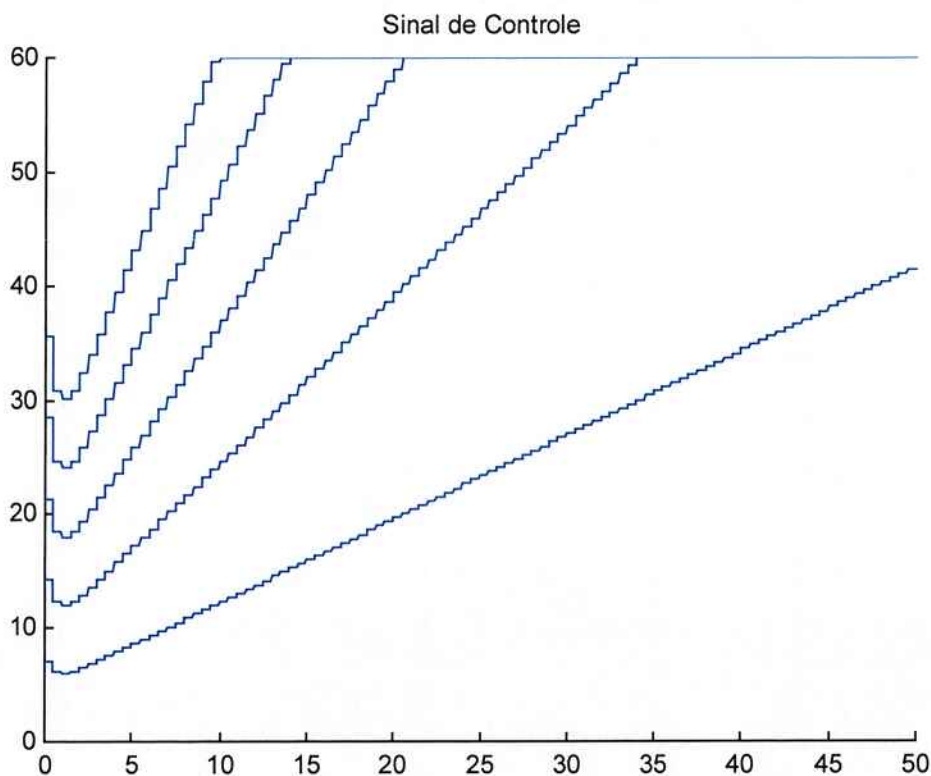
Escola Politécnica da Universidade de São Paulo
Projeto de Formatura 2001

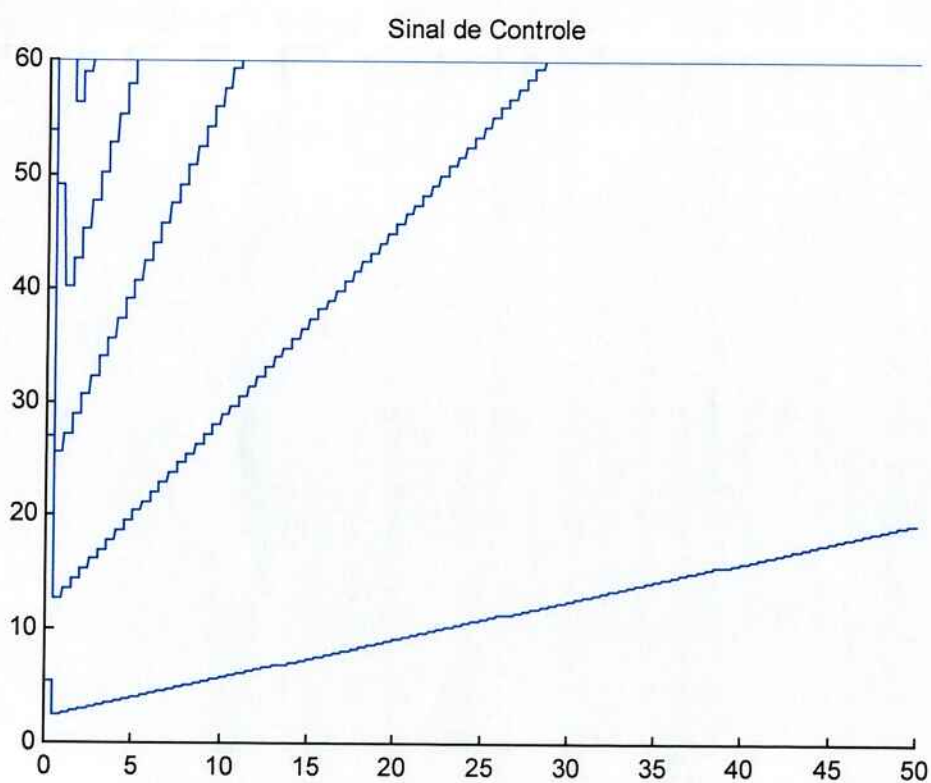
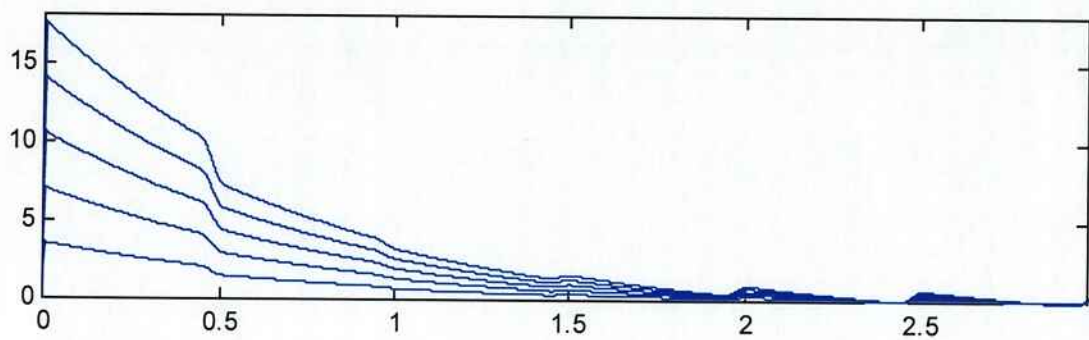
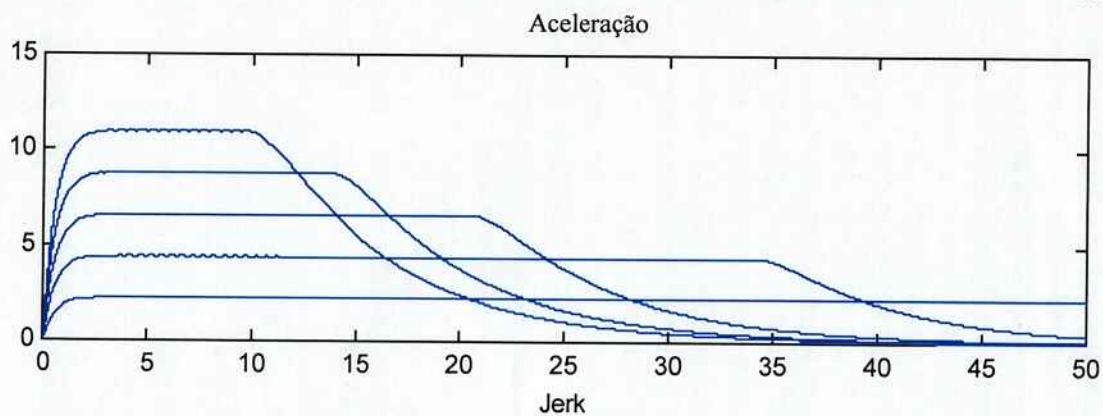
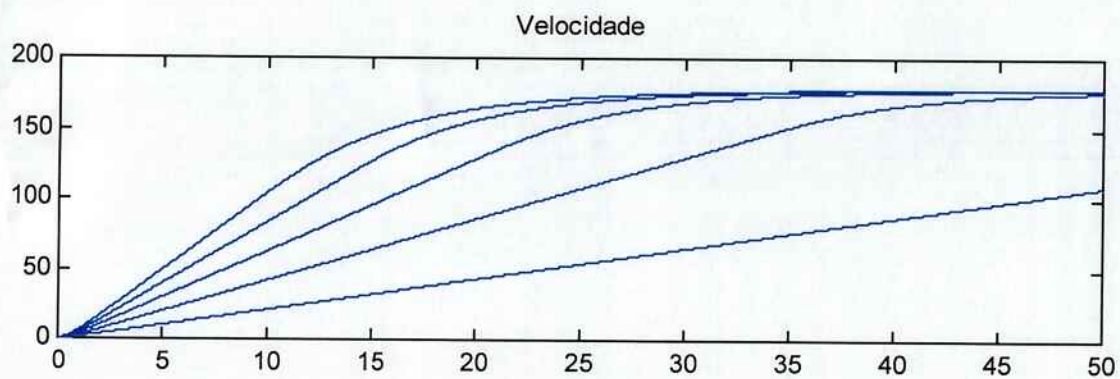
Estudo e Implementação de Sistema de Controle Ótimo para
Elevadores

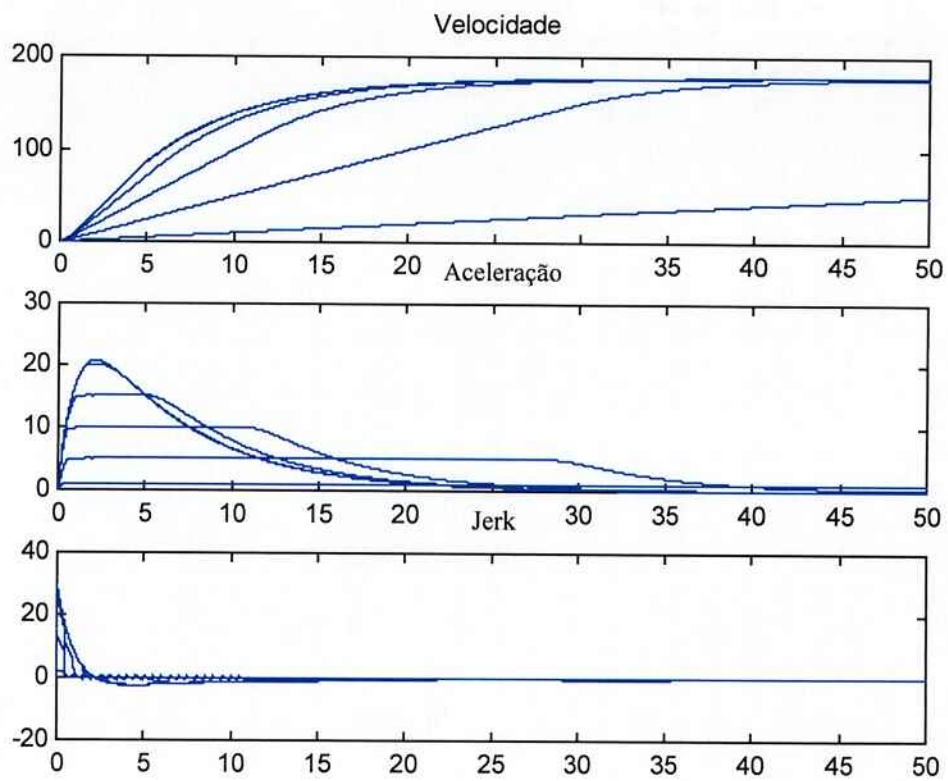
Felipe Vivacqua Pinto, Marcelo Tarkieltaub

ERRATA

Os gráficos abaixo substituem os existentes na seção de simulações do trabalho, tanto no paper técnico como no relatório final (páginas 22, 23, 24). O erro existente é devido a escolha incorreta da taxa de amostragem dos sinais de interesse nas simulações, o que levaram a obter-se resultados irreais. Os gráficos presentes nesta ERRATA possuem os resultados corretos. Os 2 primeiros gráficos são para limitações de “jerk”. O 3º e 4º são com limitações de aceleração.







Índice

1. Introdução.....	página 1
2. Objetivos.....	página 2
Metodologia.....	página 2
1 Implementação-Conceitos.....	página 2
3.2 Sinal de Referência.....	página 7
3.2.1 O problema de controle ótimo.....	página 7
3.2 Cálculo Γ 's.....	página 11
3.3 Restrições.....	página 13
4 A solução do problema.....	página 15
5 Algumas simulações.....	página 21
5 Obtenção dos parâmetros p, q, k para aproximação do sistema físico por um sistema de segunda ordem.....	página 27
3 Implementação-Especificação.....	página 29
3.1 Problemas Mecânicos.....	página 30
3.3.2 Especificação de Equipamentos(Inversor e CLP).....	página 31
3.3.3 Parametrização e Programação.....	página 33
4. Dados premissas e critérios utilizados.....	página 38
5. Resultados Obtidos.....	página 38
6. Conclusões.....	página 46
7. Bibliografia.....	página 48
8. Agradecimentos.....	página 48
9. Biografia.....	página 49
10. Anexos.....	página 49

1. Introdução

Transportar verticalmente objetos e pessoas é um problema muito antigo. A história registra que por volta de 1500a.C. foram desenvolvidos os primeiros sistemas de elevação, quando os egípcios procuravam uma solução para elevar as águas do Rio Nilo.

Desde então os rudimentares sistemas foram sendo aperfeiçoados até se chegar aos modernos elevadores de hoje.

Os primeiros elevadores eram exclusivamente de carga e movidos a vapor. Com o desenvolvimento de mecanismos de segurança surgiram os elevadores de passageiros.

Os americanos atribuem a Elisha Graves Otis a invenção em 1853, do elevador de segurança. No ano seguinte o invento foi submetido à prova no Palácio de Cristal, em Nova York. Diante da platéia, foi elevada sua plataforma e ordenado que se cortassem os cabos, para a surpresa geral, a plataforma não caiu, comprovando a sua eficácia. Em 1857, os primeiros elevadores para passageiros eram acionados por máquinas a vapor com velocidade de 12m/min.

A partir de então começaram a surgir edifícios cada vez mais altos e com eles a necessidade de equipamentos mais sofisticados.

O primeiro elevador elétrico foi instalado em Nova York, no mesmo ano da Proclamação da República no Brasil (1889). Hoje, graças ao desenvolvimento tecnológico, alguns elevadores podem atingir a velocidade de 550m/min.

O desenvolvimento da eletrônica, especialmente o da informática, revolucionou as máquinas em geral e com o elevador não foi diferente. O quadro de comando tradicional, composto de circuitos eletromecânicos, deu lugar a um sistema microprocessado que passou a coordenar, com perfeição, o funcionamento dos elevadores.

Dentro deste projeto de formatura será abordada uma aplicação específica para a solução de problema relacionado a elevadores com a aplicação de tecnologias atuais.

2. Objetivos

Pretende-se desenvolver estudo e posterior construção de um sistema semelhante àquele de um elevador tradicional.

Esse sistema consistirá essencialmente no movimento vertical de um corpo (de massa a ser definida) preso a um sistema mecânico que, por sua vez, estará acoplado a um motor elétrico.

O movimento vertical da massa desejado será tal que esta irá se deslocar entre dois pontos pré definidos (partida e chegada). As velocidades adquiridas pelo corpo ao longo dessa trajetória deverão seguir um perfil a ser estudado de tal forma que proporcione possíveis vantagens relacionadas ao tempo de percurso e a variações bruscas indesejadas de velocidade.

Para tanto será necessária a elaboração de uma lógica para o controle do motor elétrico de tal forma que o sistema como um todo opere da maneira desejada.

Na elaboração dessa solução tem-se como opção a utilização de tecnologias como as dos Controladores Lógico Programáveis e Inversores de Frequência, amplamente difundidas no mundo da automação e que se mostram extremamente interessantes dentro do escopo apresentado.

3. Metodologia

3.1 Implementação- Conceitos

Para a implementação de um sistema de controle ótimo para um elevador, analisou-se as condições nas quais este tipo de sistema se encontra.

De uma forma geral, elevadores são utilizados em edifícios residenciais ou prediais, que em sua totalidade são alimentados em corrente alternada.

Desta forma teria de se encontrar uma solução disponível no mercado, que possibilitasse a implementação do controle utilizando-se motores CA.

Dada esta necessidade, optou-se pela utilização da tecnologia de inversores de frequência para o controle primário do motor. Tal controle possibilita a simplificação do

modelo matemático que será utilizado e consequentemente a viabilidade da proposta de projeto como veremos posteriormente.

Os Motores CA possuem um campo magnético girante no entreferro do motor. A velocidade na qual este campo magnético gira, esta relacionada diretamente, à frequência de alimentação, e de características internas do motor, no caso, o número de pólos.

A produção de conjugado pelos motores CA, ocorre devido a interação do campo girante no entreferro, e as correntes que ele induz no rotor, que circulam de forma tal a gerar um conjugado que faz com que o rotor acompanhe o estator.

Sabe-se então, que para que se possa variar a velocidade de um motor AC, é necessário variar a frequência do estator, de forma a fazer com que o rotor acompanhe tal velocidade, e ao mesmo tempo, tem-se de manter o conjugado disponível máximo para qualquer frequência f . Para tal, tem de se manter o fluxo magnético constante no valor especificado pelo fabricante do motor.

Para se obter tal desempenho, existem no mercado duas soluções clássicas, o controle escalar, e o controle vetorial. Estaremos abordando o controle escalar com mais atenção, uma vez que esta foi a solução adotada.

Eletromagneticamente falando, a idéia do controle escalar, é a de que se possa manter na máquina um escorregamento s , baixo em qualquer frequência aplicada no motor, para que desta forma, se obtenha o máximo conjugado.

Este controle é extremamente eficiente quando as variações de velocidade não são extremamente bruscas, ou quando o tempo de reação não tem de ser muito alto.

Para a nossa aplicação este método de controle é extremamente adequado.

O controle escalar consiste basicamente na idéia de mantermos a relação tensão frequência constante, ou seja, $(V/f)=\text{constante}$.

Ele pode ser deduzido facilmente através de leis básicas do eletromagnetismo. Com elas é possível obter a equação clássica de conversão eletromecânica de Energia:

$$E=4,44.n.f.\phi_{\text{máx}};$$

Logo, se isolarmos $\phi_{\text{máx}}$, teremos:

$$\phi_{\text{máx}}=E/(4,44.n.f)=a.(E/f)$$

Onde a é uma constante.

Portanto, ao mantermos a relação entre tensão e frequência constantes, teremos fluxo máximo na máquina.

A única premissa colocada neste desenvolvimento é que o inversor de frequência consegue gerar ondas senoidais perfeitas, o que pode ser considerado bem razoável.

A partir disso, pode-se trabalhar como se o sistema se comportasse como o gráfico abaixo.

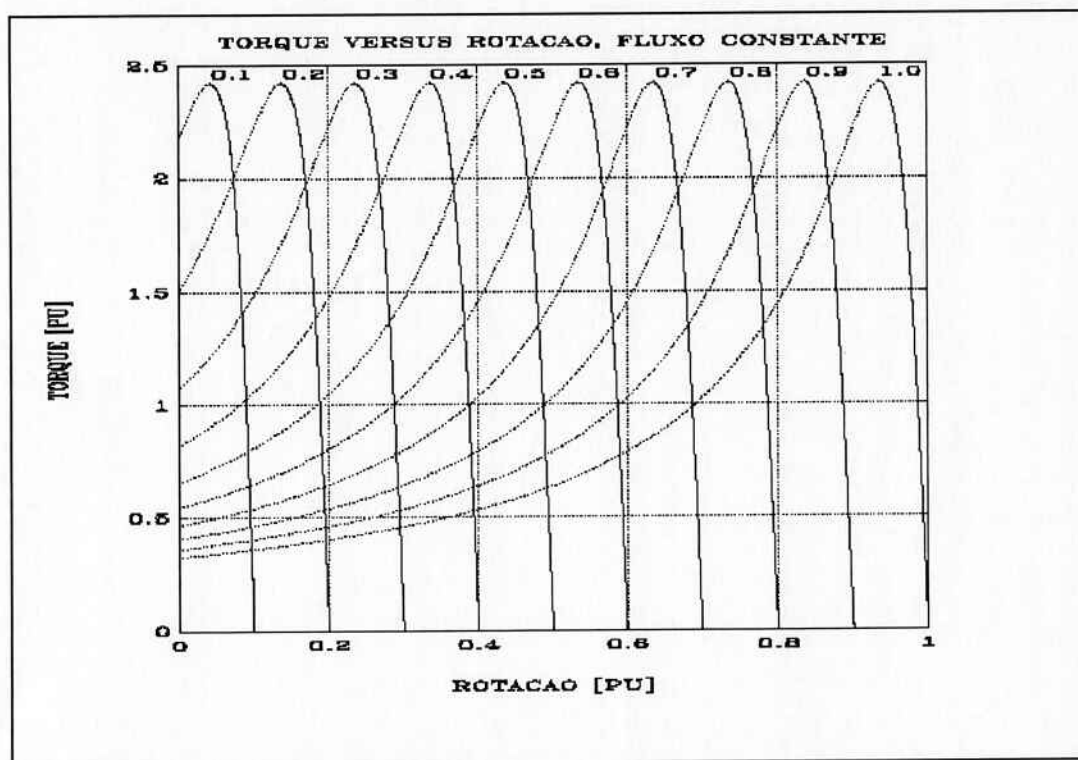


Figura 1- Gráfico Torque x Rotação

Ou seja, para qualquer frequência possui-se disponível torque máximo no eixo do motor.

Faz-se necessário uma explicação básica do funcionamento do inversor de frequência.

Para se sintetizar uma onda senoidal, é necessário a utilização de técnicas de modulação.

Para o caso de motores AC, quer se que o motor veja nele aplicada, uma tensão trifásica, senoidal e equilibrada. Como motores são “filtros” para altas frequências, pode-se dizer que ao se aplicar uma série de pulsos rápidos, o motor estará vendo a média dos mesmos. Dessa forma, se aplicarmos no motor, uma forma de onda semelhante a mostrada abaixo, ter-se-á no motor uma corrente muito próxima de uma senoidal.

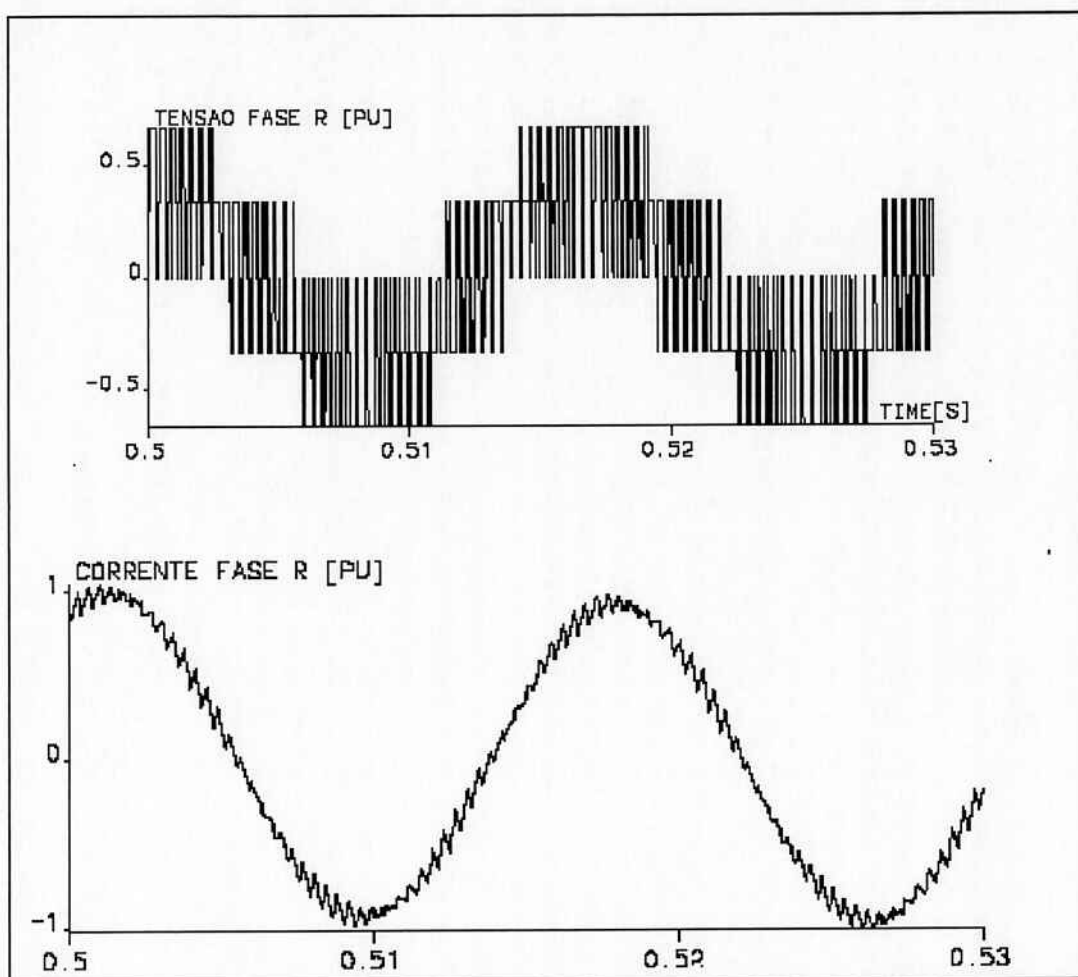


Figura 2- PWM de tensão e Corrente no motor

Para se gerar esta forma de onda, no caso do controle escalar, é necessário como “hardware” do inversor de frequência, uma ponte retificadora, um DC Link, uma ponte inversora, e um sistema de controle interno do inversor, que através da curva V/Hz, sintetiza o PWM desejado como pode ser visto abaixo:

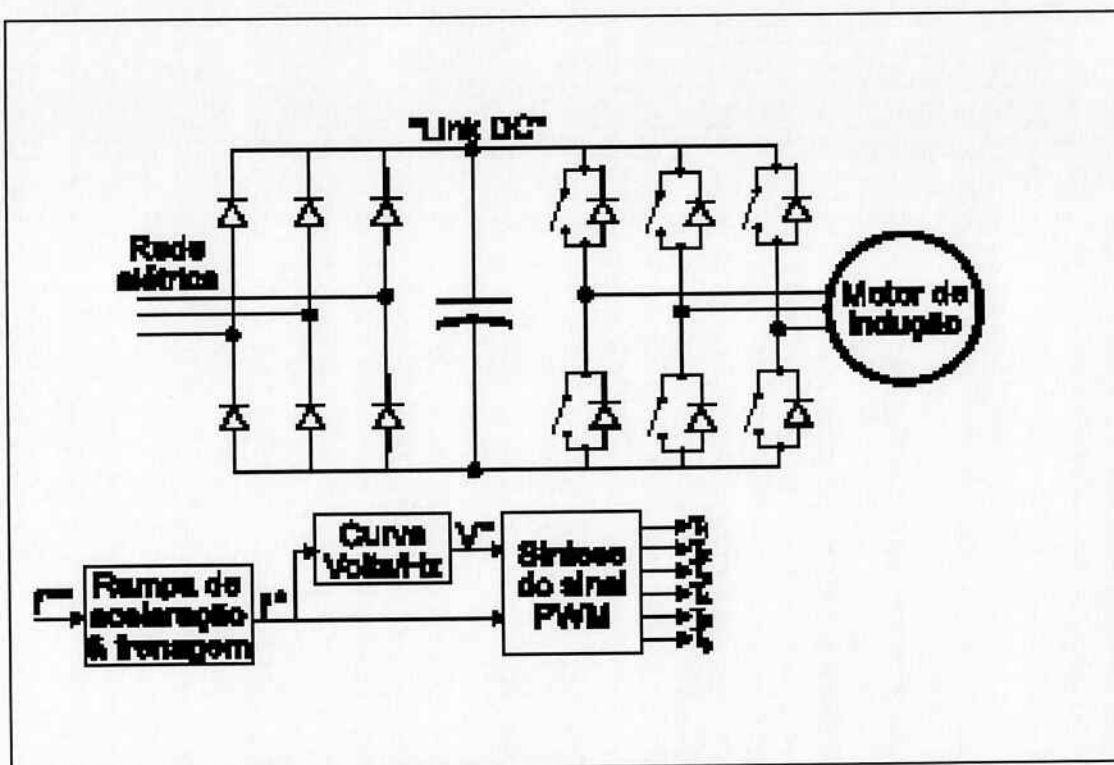


Figura 3- Esquema de um Inversor de Frequência

A partir deste hardware, mais os conceitos do controle escalar, pode-se dizer que já é possível implementar a solução.

Porém se fazia necessário gerar o sinal de controle para o inversor, que seria aplicado no “módulo” Rampa de aceleração e frenagem do esquema acima.

Para tal, decidiu-se por utilizar a tecnologia do CLP, amplamente difundida no mercado de automação industrial.

O CLP (Controlador Lógico Programável) funciona como um microcomputador, com interfaces de entradas e saídas, tanto analógicas quanto digitais de alta performance.

Com isso, pode-se implementar uma rotina, que gerará o sinal de referência para o inversor.

Dessa forma, o acoplamento motor+inversor+CLP pode ser modelado com uma função de transferência de segunda ordem como veremos mais a frente.

3.2 Sinal de Referência

3.2.1 O Problema de Controle Ótimo

Seguindo o escopo apresentado, pretende-se neste capítulo, identificar, formular e apresentar uma estratégia de solução teórica para o problema de controle ótimo em questão.

A tecnologia do inversor de frequência e do controlador lógico programável permite considerar o sistema, conforme já discutido, como sendo descrito pela seguinte equação diferencial:

$$\omega'' + p.\omega' + q.\omega = k.f \quad (1)$$

Onde ω representa a velocidade desenvolvida pelo motor, f a frequência aplicada como referência de torque (i.e. sinal de controle) e p, q, k constantes.

O problema que se deseja resolver tem como premissa a limitação nos valores de pico de algumas variáveis do sistema sendo analisado:

$$|\omega''| < \alpha \quad \text{restrição de jerk } (\omega'') \quad (2)$$

$$|\omega'| < \beta \quad \text{restrição de aceleração } (\omega') \quad (3)$$

A variável ω'' , denominada “jerk” e a variável ω' , conforme visto nas expressões (2) e (3) deverão estar sempre em um intervalo limitado de valores ao longo de todo o percurso que estiver sendo executado o movimento.

Assim temos então a necessidade de elaborar uma abordagem teórica para a identificação de um sinal de controle a ser aplicado no sistema em malha aberta (open-loop) de forma que, partindo do repouso, o motor irá atingir velocidade de regime (velocidade de referência) no menor tempo possível mantendo as variáveis “jerk” e torque dentro do intervalo de valores permitidos e estabelecidos previamente.

Utilizando-se das equações (1), (2) e (3) as restrições podem ser reescritas na forma:

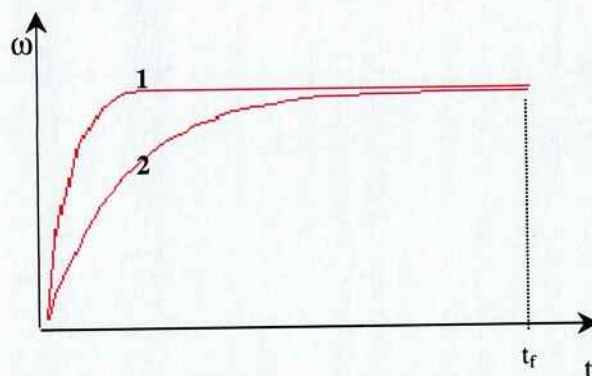
$$-\alpha < -p.\omega' - q.\omega + k.f < \alpha \quad (2')$$

$$-\beta < \omega' < \beta \quad (3')$$

Obtendo-se assim um conjunto de inequações que identificam limites a serem considerados.

Parte-se então para a definição precisa da abordagem do problema de controle ótimo, aplicando-se técnicas de programação linear. Conforme será explicado, a abordagem escolhida irá resultar em solucionar-se problema de programação linear, onde, a partir de uma expressão denominada “objetivo” e das diversas inequações que expressam as limitações de parâmetros desejadas, obtém-se os valores que maximizam ou minimizam a expressão objetivo.

Observe o gráfico abaixo:



O gráfico apresenta a resposta de um sistema qualquer de segunda ordem (open-loop) a dois sinais de entrada diferentes. É imediato perceber que a resposta número 1 é a que resulta em menor tempo de percurso, desde o início do movimento até ser atingido o valor de regime que pode ser facilmente identificado no gráfico. A abordagem escolhida para solução do problema de controle ótimo leva a tentar enxergar essa diferença entre os tempos que se atingiu o regime nos sinais 1 e 2 segundo outro enfoque: pode-se entender que o sistema mais rápido é aquele onde a área sob a curva de resposta do sistema de segunda ordem (i.e. as curvas do gráfico) tem maior valor.

Partindo-se dessa premissa, o princípio básico para a solução do problema de controle ótimo segundo técnica de programação linear consiste em maximizar a área sob a curva apresentada levando-se em considerações as restrições já apresentadas. É imediato notar que, a área máxima, caso não existissem restrições, seria aquela onde a curva de resposta (de saída do sistema de segunda ordem que foi excitado) se aproxima de uma partida direta, ou seja, corresponde a um degrau ocorrido no instante 0 segundos. É importante ressaltar que a velocidade final desejada é conhecida.

Pode-se então formular algumas expressões:

A expressão geral de cálculo da área média sob a curva é:

$$\bar{\omega} = \frac{1}{t_f} \int_0^{t_f} \omega(\tau) d\tau \quad (4)$$

Onde $\omega(\tau)$ representa a velocidade de saída desenvolvida pelo sistema.

A solução que será implementada, em equipamento eletrônico/digital, torna necessária discretização do problema.

Assim, o cálculo da área sob a curva em questão pode ser discretizado obtendo-se valor aproximado da área média, segundo a seguinte expressão:

$$\bar{\omega} = \frac{1}{t_f} [(\omega_1 + \omega_2 + \omega_3 + \dots + \omega_n) \cdot \frac{t_f}{n}] \quad (5)$$

Antes de continuarmos, abaixo é apresentada a estrutura geral de um problema de programação linear:

Função Objetivo: $Y(x_1, x_2, x_3, \dots, x_n) = a_1 x_1 + a_2 x_2 + a_3 x_3 + \dots + a_n x_n$

sujeita a k limitações:

$$\begin{aligned} f_1(x_1, x_2, x_3, \dots, x_n) &< L_1 \\ &\vdots \\ f_k(x_1, x_2, x_3, \dots, x_n) &< L_k \end{aligned}$$

Deseja-se maximizar a função objetivo Y segundo as variáveis $x_1, x_2, x_3, \dots, x_n$ (sendo a_1, a_2, a_n constantes) satisfazendo as limitações impostas.

De volta ao nosso problema, é imediato perceber que nossa função objetivo é dada pela expressão da área, ω , que deverá ser expressa em função dos valores do sinal de controle (sinal de entrada do sistema) para se enquadrar a estrutura de um problema de programação linear. Da mesma forma, as limitações deverão ser adequadamente reescritas de forma a se enquadrar ao modelo de problema de programação linear.

Continuando o desenvolvimento da formulação considera-se:

A resposta geral de um sistema linear é do tipo:

$$x' = Ax + b \cdot u(t) \quad (6)$$

$$x(t) = e^{A \cdot t} \cdot x(0) + \int_0^{t_f} e^{A(t_f - \tau)} \cdot b \cdot u(\tau) d\tau \quad (7)$$

onde $\underline{x}$ representa o vetor das variáveis de estado do sistema e u é o sinal de entrada. É imediato perceber que a expressão (7) é aquela que nos dá a expressão de $\omega(t)$.

Então surge uma primeira dificuldade. Se substituirmos a expressão (7) na função objetivo (4), percebe-se que os valores de u (sinal de entrada) fazem parte da integração, o que impossibilitaria a abordagem via programação linear. No entanto, a discretização do problema permite considerarmos o sinal de controle como sendo composto por uma sequência de diversos degraus (sendo aplicados na entrada do sistema) de amplitudes distintas e constantes por um certo intervalo de tempo, de tal forma que $u(\tau)$ pode ser "retirado" da integração já que é constante no intervalo de discretização.

De volta a expressão (7) e considerando-se condições iniciais nulas obtem-se:

$$x(t) = \begin{bmatrix} \omega(t) \\ \omega'(t) \end{bmatrix} = \int_0^{t_f} e^{A(t_f - \tau)} \cdot b \cdot u(\tau) d\tau \quad (8)$$

Continuando a formulação e de posse destas informações, faz-se necessário então obter-se os valores de $\omega_1, \omega_2, \omega_3, \dots, \omega_n$ (expressão (5)) em função dos sinais de entrada $u_1, u_2, \dots, u_n$ conforme a discretização proposta. Então teremos uma função chamada objetivo do tipo $\omega = \omega(u_1, u_2, \dots, u_n)$, que deverá ser maximizada segundo os valores de $u_1, u_2, \dots, u_n$.

Tomando-se a expressão (8), multiplicando-se pelo vetor linha $[1 \ 0]$ extrai-se a expressão de $\omega(t)$. Após a discretização obter-se-á:

$$\omega_n = [1 \ 0] \left(\int_0^{t_1} e^{A(t_1 - \tau)} \cdot b \cdot d\tau \cdot u_1 + \dots + \int_{t_{n-1}}^{t_n} e^{A(t_n - \tau)} \cdot b \cdot d\tau \cdot u_n \right) \quad (9)$$

Para simplificar o entendimento faz-se:

$$\Gamma_{k,n} = \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)} \cdot b \cdot d\tau \quad (10)$$

Substituindo-se a expressão (10) em (9) e reorganizando a expressão ter-se-á:

$$\bar{\omega} = [1 \ 0] \{ (\Gamma_{1,1} + \Gamma_{1,2} + \dots + \Gamma_{1,n}) u_1 + (\Gamma_{2,2} + \Gamma_{2,3} + \dots + \Gamma_{2,n}) u_2 + \dots + \Gamma_{n,n} u_n \} \quad (11)$$

Expressão essa que deverá ser maximizada segundo os parâmetros $u_1, u_2, \dots, u_n$.

Segundo todo o desenvolvimento realizado até então foi possível obter-se a expressão objetivo aos moldes de um problema de programação linear.

O próximo passo é calcular o valor dos diversos Γ 's da expressão (11).

3.2.2 Cálculo Γ 's

Para a obtenção Γ 's da expressão (11), faz-se necessário o cálculo da integral em (10). Sendo o sistema do tipo dado em (6), tem-se:

$$e^{At} = L^{-1}[(sI - A)^{-1}] \quad (12)$$

onde para o sistema de segunda ordem em questão de (6) tem-se:

$$A = \begin{bmatrix} 0 & 1 \\ -q & -p \end{bmatrix} \quad e \quad b = \begin{bmatrix} 0 \\ k \end{bmatrix} \quad (13)$$

Para o cálculo da anti-transformada de Laplace necessário conforme apresentado na expressão (12) faz-se necessário a divisão do problema. Inicialmente iremos considerar que o problema possui autovalores reais.

Assim, para o caso de autovalores reais:

$$e^{At} \cdot b = \begin{bmatrix} k \cdot A_1 \cdot e^{\lambda_1 t} + k \cdot B_1 \cdot e^{\lambda_2 t} \\ k \cdot A_2 \cdot e^{\lambda_1 t} + k \cdot B_2 \cdot e^{\lambda_2 t} \end{bmatrix} \quad (14)$$

É interessante notar que o cálculo da matriz inversa de $(sI - A)$ conforme sugere a expressão (11) foi omitido, já que esse não faz-se necessário para o bom entendimento da formulação do problema.

Os valores de A_1, A_2 e B_1, B_2 da expressão (14) são dados por:

$$B_1 = \frac{1}{\lambda_1 - \lambda_2} \quad A_1 = \frac{-1}{\lambda_1 - \lambda_2} \quad (15)$$

$$B_2 = \frac{\lambda_2}{(\lambda_2 - \lambda_1)} \quad A_2 = \frac{-\lambda_1}{(\lambda_2 - \lambda_1)} \quad (16)$$

Tais valores foram obtidos pelo método convencional de antitransformação de laplace. Os parâmetros λ_1, λ_2 são os autovalores do sistema.

Pode-se agora substituir em (14) os valores das expressões (15) e (16). Em seguida calcula-se a integral da expressão (10). Para simplificação da leitura esse cálculo foi omitido do texto.

Como resultado desse cálculo obtém-se:

$$\tilde{\Gamma}_{k,n} = \left[\begin{array}{c} \frac{k.A_1.e^{\lambda_1 \cdot (t_n - \tau)} + k.B_1.e^{\lambda_2 \cdot (t_n - \tau)}}{\lambda_1} + \frac{k.A_2.e^{\lambda_1 \cdot (t_n - \tau)} + k.B_2.e^{\lambda_2 \cdot (t_n - \tau)}}{\lambda_2} \end{array} \right]_{t'_{k-1}}^{t'_k} \quad (17)$$

Vamos agora considerar o caso onde os autovalores são imaginários:

Segue-se raciocínio análogo.

$$e^{At} \cdot b = \left[\begin{array}{c} 2.k.|A_1|e^{\text{Re}[\lambda_1]t} \cdot \cos(\text{Im}[\lambda_1]t + \text{angle}(A_1)) \\ 2.k.|A_2|e^{\text{Re}[\lambda_1]t} \cdot \cos(\text{Im}[\lambda_1]t + \text{angle}(A_2)) \end{array} \right] \quad (18)$$

Os valores de A_1, A_2 e B_1, B_2 são dados por:

$$A_1 = \frac{-1}{\lambda_1 - \lambda_2} \quad B_1 = A_1^* \quad (19)$$

$$A_2 = \frac{\lambda_1}{\lambda_1 - \lambda_2} \quad B_2 = A_2^* \quad (20)$$

Calculando novamente a integral em (10) (integração por partes) obtém-se:

$$\Gamma_{k,n} = \left[\frac{e^{-(t_n - \tau).a} \cdot \sin(\omega.(t_n - \tau) + \phi) \cdot \omega}{\omega^2 + a^2} - \frac{a \cdot e^{-(t_n - \tau).a} \cdot \cos(\omega.(t_n - \tau) + \phi)}{\omega^2 + a^2} \right]_{t'_{k-1}}^{t'_k} \cdot 2.k.|A_1| \quad (21)$$

com $\phi = \text{angle}(A_1)$ e $\omega = \text{Im}[\lambda_1]$ e $a = \text{Re}[\lambda_1]$.

$$\Gamma'_{k,n} = \left[\frac{e^{-(t_n - \tau).a} \cdot \sin(\omega.(t_n - \tau) + \phi) \cdot \omega}{\omega^2 + a^2} - \frac{a \cdot e^{-(t_n - \tau).a} \cdot \cos(\omega.(t_n - \tau) + \phi)}{\omega^2 + a^2} \right]_{t'_{k-1}}^{t'_k} \cdot 2.k.|A_1| \quad (22)$$

com $\phi = \text{angle}(A_2)$ e $\omega = \text{Im}[\lambda_1]$ e $a = \text{Re}[\lambda_1]$.

$$\text{onde } \Gamma_{k,n} = \begin{bmatrix} \Gamma_{k,n} \\ \Gamma'_{k,n} \end{bmatrix} \quad (23)$$

É imediato perceber que a expressão objetivo dada em (11), quando com os valores de (17) ou (22) e (23) se adapta perfeitamente a estrutura de programação linear necessário. Faz-se então necessária a adoção da mesma abordagem com as expressões (2) e (3), abordagem essa que segue.

3.2.3 Restrições

Cabe ainda obter-se expressões gerais para as restrições apresentadas em (2) e (3) que deverão ser atendidas ponto a ponto, segundo o número “n” de divisões consideradas na discretização conforme em (5).

A expressão (2) pode ser escrita com a utilização de matrizes:

$$-\alpha < [-q \quad -p] \begin{bmatrix} \omega_k(t) \\ \omega'_k(t) \end{bmatrix} + k.u_k < \alpha \quad (24)$$

Ou ainda, para cada instante t_k (da discretização), utilizando as equações (9) e (10) tem-se:

$$-\alpha < [-q \cdot \Gamma_{1,n} - p \cdot \Gamma'_{1,n} \dots - q \cdot \Gamma_{n-1,n} - p \cdot \Gamma'_{n-1,n} - q \cdot \Gamma_{n,n} - p \cdot \Gamma'_{n,n} + k] \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} < \alpha \quad (25)$$

A expressão (3) pode ser escrita como:

$$-\beta < [0 \ 1] \cdot \begin{bmatrix} \omega(t_k) \\ \omega'(t_k) \end{bmatrix} < \beta \quad (26)$$

Ou, utilizando as equações (9) e (10), a equação (26) torna-se:

$$-\beta < [\Gamma'_{1,n} \ \Gamma'_{2,n} \dots \Gamma'_{n,n}] \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} < \beta \quad (27)$$

Dessa forma obtém-se as equações (25) e (27), onde estão definidas as restrições para cada instante de tempo da discretização. É extremamente importante lembrar que estas equações devem ser calculadas para cada instante t da discretização. Dessa forma, se existirem 100 pontos resultantes da discretização, tempo final da simulação de 10 segundos, teremos discretização com 100 degraus de 0.1 segundo de duração e um total de 400 restrições (dadas as equações (25) e (27)).

Conclui-se então a formulação do problema. Parte-se agora para a apresentação da abordagem para a solução do problema proposto. Conforme proposto, obteve-se uma formulação dentro da estrutura de um problema de programação linear:

Função Objetivo: $Y(x_1, x_2, x_3, \dots, x_n) = a_1 x_1 + a_2 x_2 + a_3 x_3 + \dots + a_n x_n \rightarrow$ que para nosso problema é a equação (11):

sujeita a k limitações:

$$f_1(x_1, x_2, x_3, \dots, x_n) < L_1$$

⋮

$$f_k(x_1, x_2, x_3, \dots, x_n) < L_k$$

→ equações (25) e (27)

3.2.4 A Solução do Problema

De posse das equações (11),(17),(21),(22), (25) e (27) e dos parâmetros do sistema (i.e. p,q,k, ver equação (1)) nos resta então resolver o problema, obtendo o vetor u , sinal de controle que deverá ser aplicado ao sistema de forma a atender as restrições impostas e com valor máximo para a função objetivo. Vale lembrar que “valor máximo para a função objetivo” significa dizer partir o sistema de uma velocidade inicial nula e atingir velocidade de regime em tempo mínimo, dadas as restrições.

O software MATLAB da empresa MATHWORKS possui em sua biblioteca de funções a função “lp” (linear programming). Segue abaixo detalhes dessa função:

```
LP      Linear programming.
X=LP(f,A,b) solves the linear programming problem:

        min f'x      subject to:   Ax <= b
        x

X=LP(f,A,b,VLB,VUB) defines a set of lower and upper
bounds on the design variables, X, so that the solution is always in
the range VLB <= X <= VUB.

X=LP(f,A,b,VLB,VUB,X0) sets the initial starting point to X0.

X=LP(f,A,b,VLB,VUB,X0,N) indicates that the first N constraints
defined
by A and b are equality constraints.

X=LP(f,A,b,VLB,VUB,X0,N,DISPLAY) controls the level of warning
messages displayed. Warning messages can be turned off with
DISPLAY = -1.

[x,LAMBDA]=LP(f,A,b) returns the set of Lagrangian multipliers,
LAMBDA, at the solution.

[X,LAMBDA,HOW] = LP(f,A,b) also returns a string how that indicates
error conditions at the final iteration.

LP produces warning messages when the solution is either unbounded
or infeasible.
```

É imediato perceber que a função “lp” do MATLAB oferece uma alternativa para solução do problema de programação linear formulado.

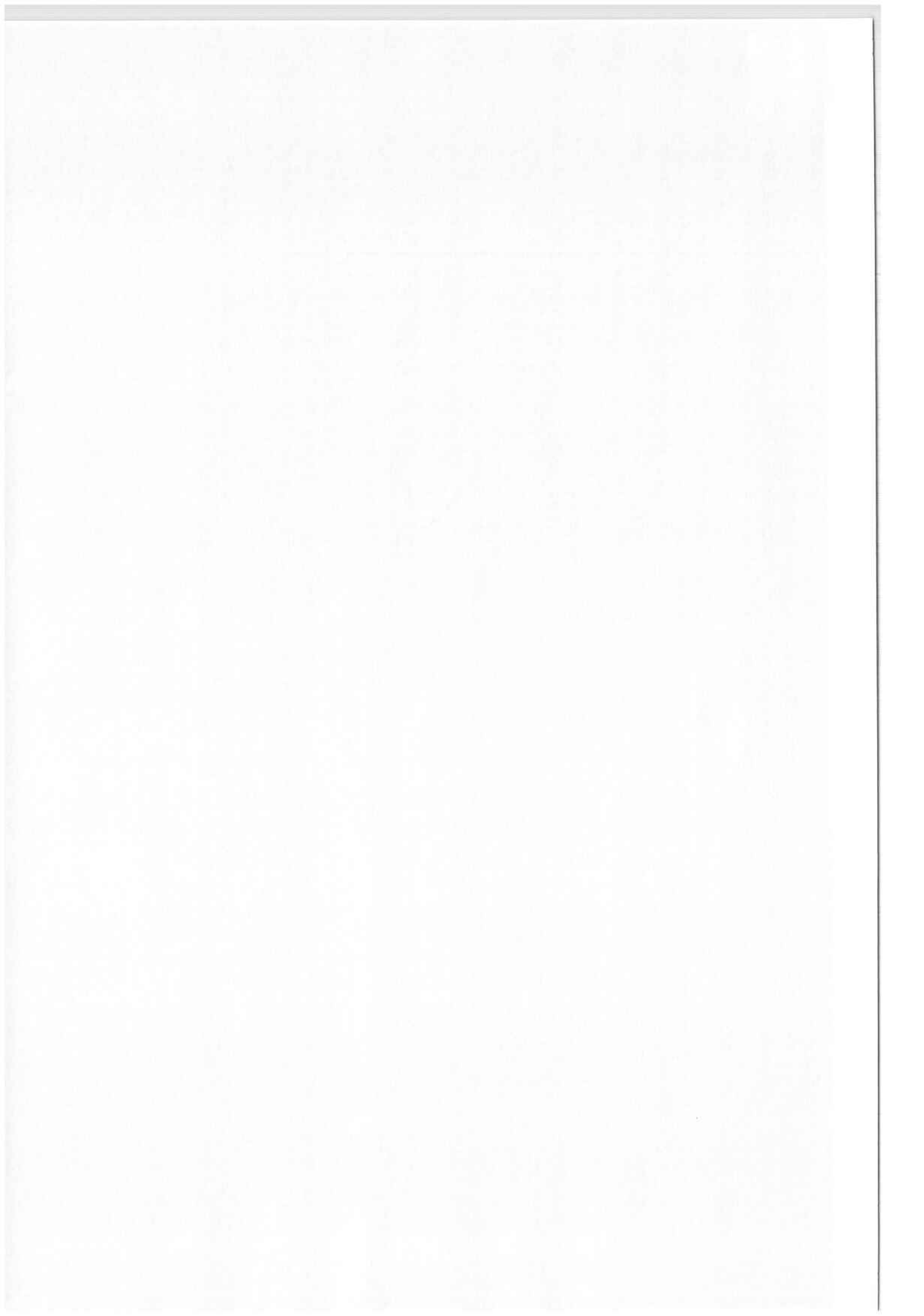
O esforço agora deve ser concentrado na elaboração do código “.m” que deverá ser implementado, onde se considera a utilização da função “lp”.

O programa deverá considerar os seguintes pontos:

- a possibilidade de existirem autovalores reais ou imaginários
- Cálculo Γ 's
- Obtenção das matrizes que serviram de entrada para a função “lp” (i.e. as matrizes das restrições e a matriz da função objetivo.
- Limitar o sinal de controle que deverá estar no intervalo de 0 à 60 HZ (possível graças a rotina “lp”).
- liberdade de alteração dos parâmetros do sistema e restrições para análise/ confronto de diferentes resultados.
- organização do resultado de forma a gerar sinal a ser utilizado em simulação no software SIMULINK.

Dadas as condições estabelecidas foi elaborado o código apresentado abaixo, para o caso de autovalores reais:

<pre>clear fator=32768/60; vlb=0; vub=60;</pre>	}	Definições iniciais. vlb e vub são os valores limite superior e inferior para o sinal de controle
<pre>r1=100 r2=15 n=125 Tf=25</pre>	}	Restrições. r1 e r2 são as restrições de jerk e torque, respectivamente. Tf é o tempo final considerado para controle.
<pre>autovalor1=1/6; autovalor2=1/(6/3); p=autovalor1+autovalor2; q=autovalor1*autovalor2; k=(178/60)*q; lambda1=(-p+sqrt(p^2-4*q))/2; lambda2=(-p-sqrt(p^2-4*q))/2; A=[0 0;-q -p]; B=[0 k]; t=0:Tf/n:Tf; passo=(Tf/(n))/10; A1=(-1/(lambda1-lambda2))*k; B1=(1/(lambda1-lambda2))*k; A2=(-lambda1/(-lambda1+lambda2))*k; B2=(lambda2/(lambda2-lambda1))*k; for linha = 2:1:n+1</pre>	}	São definidas aqui as características do sistema e parâmetros: p,q,k, autovalores e as matrizes A e b (equação (6)). As variáveis A1, B1,A2,B2 são aquelas obtidas nas equações (15) e (16).



```

for coluna = linha:1:n+1
    gama1(linha-1,coluna-1)=
    (A1*exp( lambda1*(t(coluna)-t(linha)) ))/lambda1 +
    (B1*exp( lambda2*(t(coluna)-t(linha)) ))/lambda2;
    gama1(linha-1,coluna-1)=
    gama1(linha-1,coluna-1)-
    ((A1*exp(lambda1*(t(coluna)-t(linha-1))))/lambda1+
    (B1*exp( lambda2*(t(coluna)-t(linha-1)) ))/lambda2);
end
end

```

Aqui é calculado o valor de Γ conforme a equação (17).

Os valores são armazenados na matriz gama1.

```

for linha = 2:1:n+1
    for coluna = linha:1:n+1
        gama2(linha-1,coluna-1)=
        (A2*exp( lambda1*(t(coluna)-t(linha)) ))/lambda1 +
        (B2*exp( lambda2*(t(coluna)-t(linha)) ))/lambda2;
        gama2(linha-1,coluna-1)=
        gama2(linha-1,coluna-1)-
        ((A2*exp( lambda1*(t(coluna)-t(linha-1)) ))/lambda1 +
        (B2*exp( lambda2*(t(coluna)-t(linha-1)) ))/lambda2);
    end
end

```

Aqui é calculado o valor de Γ' conforme a equação (17).

Os valores são armazenados na matriz gama2.

```

gamaBarra=0;
for linha = 2:1:n+1
    for coluna = linha:1:n+1
        if linha==coluna
            gamaBarra(linha-1)=gama1(linha-1,coluna-1);
        else
            gamaBarra(linha-1)=gamaBarra(linha-1)+
            gama1(linha-1,coluna-1);
        end
    end
end
end

```

De posse dos valores de Γ e Γ' , calcula-se a matriz da função objetivo conforme a equação (11). A matriz foi chamada no código de gamaBarra.

```

gamaBarra=-gamaBarra*(Tf/n);
for l=1:1:n
    for x=1:1:n
        if x==1
            restj(x,l)=-q*gama1(l,x)-p*gama2(l,x)+k;
        else
            restj(x,l)=-q*gama1(l,x)-p*gama2(l,x);
        end
    end
end
for x=n+1:1:2*n
    for l=x:1:2*n
        restj(l,x-n)=gama2(x-n,l-n);
    end
end
for l=2*n+1:1:3*n
    for x=1:1:3*n
        if x==1
            restj(x,l-2*n)=-(-q*gama1(l-2*n,x-2*n)-p*gama2(l-2*n,x-2*n)+k);
        else
            restj(x,l-2*n)=-(-q*gama1(l-2*n,x-2*n)-p*gama2(l-2*n,x-2*n));
        end
    end
end
end
for x=3*n+1:1:4*n
    for l=x:1:4*n
        restj(l,x-3*n)=-gama2(x-3*n,l-3*n);
    end
end
end

```

Aqui, é montada a matriz das restrições segundo as equações (25) e (27). Note que as matrizes das equações (25) e (27) são empilhadas em uma única matriz denominada no código de restj.

```

r_v=r1*ones(n,1);
r_v=[r_v ; r2*ones(n,1)];
r_v=[r_v ; r1*ones(n,1)];
r_v=[r_v ; r2*ones(n,1)];

```

Construção do vetor de restrições de jerk e torque. Equivalente aos parâmetros α e β nas equações (25) e (27)

```

vub_v=vub*ones(length(restj),1);
vub_v=vub*ones(length(restj),1);

```

Construção do vetor com os limites superior e inferior do sinal de controle

```

sc=lp(gamaBarra,restj,r_v,vub_v,vub_v)

```

De posse de todas as matrizes necessárias a rotina "lp" é chamada.

```

temp=0:passo:Tf/(n);
for x=2:1:length(sc)
    temp=[temp Tf/(n)*(x-1)+passo:passo:Tf/(n)*x];
end

cont=sc(1)*ones(10,1);
for x=2:1:length(sc)
    cont = [cont; sc(x)*ones(10,1)];
end
cont(length(cont)+1)=cont(length(cont));
controle=[temp' cont];

```

Construção do vetor de controle a partir dos resultados gerados pela rotina "lp". Essa construção consiste na obtenção de um vetor onde a primeira coluna é a base de tempo em intervalos segundo o intervalo de discretização definido e a segunda coluna é o sinal de controle gerado pela rotina "lp".

-----FIM DO CÓDIGO-----

Para o caso de autovalores complexos, procedeu-se de maneira análoga:

```

clear
vub=0
vub=60

```

Definições iniciais.

vub e vub são os valores limite superior e inferior para o sinal de controle


```

r1=100
r2=5
n=125
Tf=25

```

Restrições.
r1 e r2 são as restrições de jerk e torque, respectivamente.
Tf é o tempo final considerado para controle.

```

autovalor1=1/6;
autovalor2=1/(6/3);

p=autovalor1+autovalor2;
q=autovalor1*autovalor2;
k=(178/60)*q;
lambda1=(-p+sqrt(p^2-4*q))/2
lambda2=(-p-sqrt(p^2-4*q))/2

A=[0 0;-q -p]
B=[0 k]

t=0:Tf/n:Tf;

A1=-1/(lambda1-lambda2);
A2=lambda1/(lambda1-lambda2);

```

São definidas aqui as características do sistema e parâmetros: p,q,k, autovalores e as matrizes A e b (equação (6)).
As variáveis A1, B1,A2,B2 são aquelas obtidas nas equações (19) e (20).

```

w=abs(imag(lambda1));
e1=real(lambda1);
passo=(Tf/(n))/10;
for linha = 2:1:n+1
    for coluna = linha:1:n+1
        fi=angle(A1);
        gama1(linha-1,coluna-1)=
            ((exp(e1*t(linha))*exp(-t(coluna)*e1))*
            (sin(w*(t(coluna)-t(linha))+fi)*w/(w^2+e1^2)))-
            ((exp(e1*t(linha))*exp(-t(coluna)*e1))*(cos(w*(t(coluna)-
            t(linha))+fi)*e1/(w^2+e1^2)));
        gama1(linha-1,coluna-1)=gama1(linha-1,coluna-1)-
            ((exp(e1*t(linha-1))*exp(-t(coluna)*e1))*
            (sin(w*(t(coluna)-t(linha-1))+fi)*w/(w^2+e1^2)))-
            ((exp(e1*t(linha-1))*exp(-t(coluna)*e1))*
            (cos(w*(t(coluna)-t(linha-1))+fi)*e1/(w^2+e1^2)));
        gama1(linha-1,coluna-1)=gama1(linha-1,coluna-1)*2*k*abs(A1);
    end
end

```

Aqui é calculado o valor de Γ conforme a equação (21).

Os valores são armazenados na matriz gama1.

```

w=abs(imag(lambda1));
e1=real(lambda1);
for linha = 2:1:n+1
    for coluna = linha:1:n+1
        fi=angle(A2);
        gama2(linha-1,coluna-1)=((exp(e1*t(linha))*
        exp(-t(coluna)*e1))*
        (sin(w*(t(coluna)-t(linha))+fi)*w/(w^2+e1^2)))-
        ((exp(e1*t(linha))*exp(-t(coluna)*e1))*
        (cos(w*(t(coluna)-t(linha))+fi)*e1/(w^2+e1^2)));
        gama2(linha-1,coluna-1)=gama2(linha-1,coluna-1)-
            ((exp(e1*t(linha-1))*exp(-t(coluna)*e1))*
            (sin(w*(t(coluna)-t(linha-1))+fi)*w/(w^2+e1^2)))-
            ((exp(e1*t(linha-1))*exp(-t(coluna)*e1))*
            (cos(w*(t(coluna)-t(linha-1))+fi)*e1/(w^2+e1^2)));
        gama2(linha-1,coluna-1)=gama2(linha-1,coluna-1)*2*k*abs(A2);
    end
end
gamaBarra=0;

```

Aqui é calculado o valor de Γ conforme a equação (22).

Os valores são armazenados na matriz gama1.

```

for linha = 2:1:n+1
    for coluna = linha:1:n+1
        if linha==coluna
            gamaBarra(linha-1)=-gama1(linha-1,coluna-1);
        else
            gamaBarra(linha-1)=gamaBarra(linha-1)-
                gama1(linha-1,coluna-1);
        end
    end
end

```

De posse dos valores de Γ e Γ' , calcula-se a matriz da função objetivo conforme a equação (11). A matriz foi chamada no código de gamaBarra.

```

end
end
for l=1:1:n
    for x=1:1:n
        if x==1
            restj(x,l)=-q*gama1(l,x)-p*gama2(l,x)+k;
        else
            restj(x,l)=-q*gama1(l,x)-p*gama2(l,x);
        end
    end
end

```

```

for x=n+1:1:2*n
    for l=x:1:2*n
        restj(l,x-n)=gama2(x-n,l-n);
    end
end

```

```

for l=2*n+1:1:3*n
    for x=1:1:3*n
        if x==1
            restj(x,l-2*n)=-(-q*gama1(l-2*n,x-2*n)-p*gama2(l-2*n,x-2*n)+k);
        else
            restj(x,l-2*n)=-(-q*gama1(l-2*n,x-2*n)-p*gama2(l-2*n,x-2*n));
        end
    end
end

```

```

for x=3*n+1:1:4*n
    for l=x:1:4*n
        restj(l,x-3*n)=-gama2(x-3*n,l-3*n);
    end
end

```

```

r_v=r1*ones(n,1);
r_v=[r_v ; r2*ones(n,1)];
r_v=[r_v ; r1*ones(n,1)];
r_v=[r_v ; r2*ones(n,1)];

```

Construção do vetor de restrições de jerk e torque. Equivalente aos parâmetros α e β nas equações (25) e (27)

```

vlb_v=vlb*ones(length(restj),1);
vub_v=vub*ones(length(restj),1);

```

Construção do vetor com os limites superior e inferior do sinal de controle

```

sc=lp(gamaBarra,restj,r_v,vlb_v,vub_v)

```

De posse de todas as matrizes necessárias a rotina "lp" é chamada.

Aqui, é montada a matriz das restrições segundo as equações (25) e (27). Note que as matrizes das equações (25) e (27) são empilhadas em uma única matriz denominada no código de restj.

```

temp=0:passo:Tf/(n);
for x=2:1:length(sc)
    temp=[temp (Tf/(n)*(x-1))+passo:passo:Tf/(n)*x];
end

cont=sc(1)*ones((1/passo)*Tf/(n),1);
for x=2:1:length(sc)
    cont = [cont; sc(x)*ones((1/passo)*Tf/(n),1)];
end

cont(length(cont)+1)=cont(length(cont));
controle=[temp' cont];
-----FIM DO CÓDIGO-----
-----

```

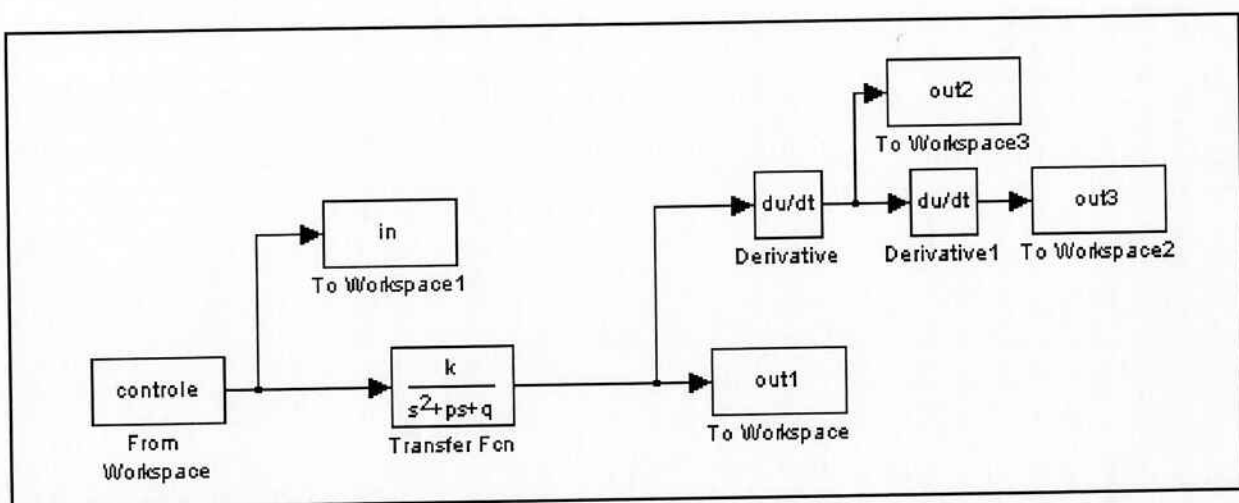
Construção do vetor de controle a partir dos resultados gerados pela rotina "lp". Essa construção consiste na obtenção de um vetor onde a primeira coluna é a base de tempo em intervalos segundo o intervalo de discretização definido e a segunda coluna é o sinal de controle gerado pela rotina "lp".

Uma observação importante é que, no caso de obter-se autovalores reais e iguais ou autovalores complexos puros far-se-ão necessárias pequenas alterações nos códigos propostos acima.

3.2.5 Algumas Simulações

Tendo em mãos o código apresentado no item anterior e as características de um sistema qualquer, pode-se simular o comportamento desse sistema quando excitado com o sinal de controle gerado pelo código. Inicialmente considera-se um sistema geral para ilustrar o comportamento da simulação. Em seguida serão apresentados resultados para o sistema implementado em laboratório, que é o sistema "foco" do projeto.

O software utilizado nas simulações foi o SIMULINK, presente no MATLAB. O diagrama de simulação é o seguinte:



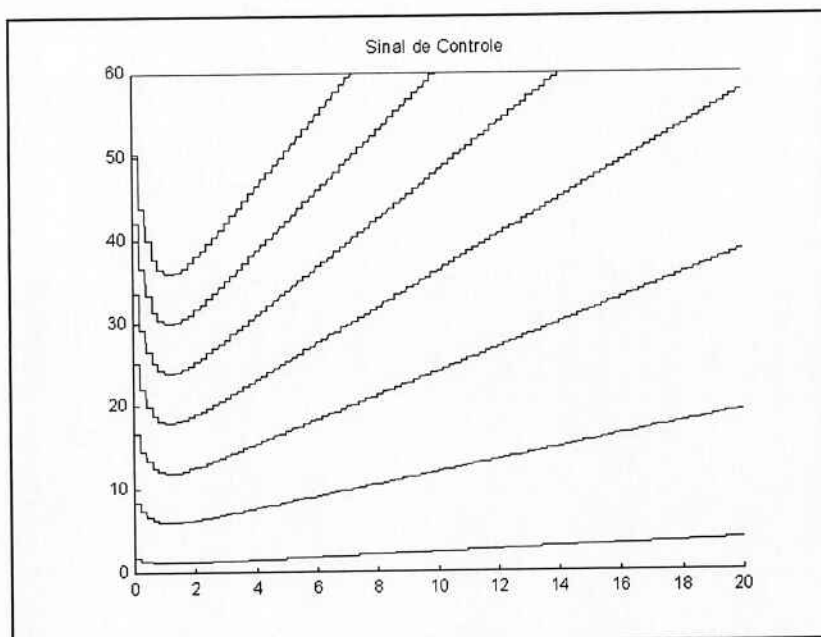
Alguns itens do diagrama merecem uma breve descrição para situá-los no contexto dos objetivos da simulação. O bloco “controle” é o bloco de ligação entre a simulação e os dados obtidos do código “.m” oriundos do Matlab. Basicamente, para o funcionamento deste bloco, deve ser passado o nom de uma matriz $n \times 2$ onde a primeira coluna estão os valores de tempo e na segunda coluna os valores de controle para os respectivos instantes. O próximo bloco “Transfer Fcn” nada mais é do que um bloco do tipo função de transferência linear onde são passados os parâmetros p e q . Os blocos “in”, “out1”, “out2” e “out3” são os blocos responsáveis pela obtenção dos resultados da simulação e sua passagem para o “Workspace” do Matlab. Note ainda a presença de dois blocos derivadores para poder-se obter os valores de ω'' e ω' .

Os parâmetros para a primeira simulação considerados são:

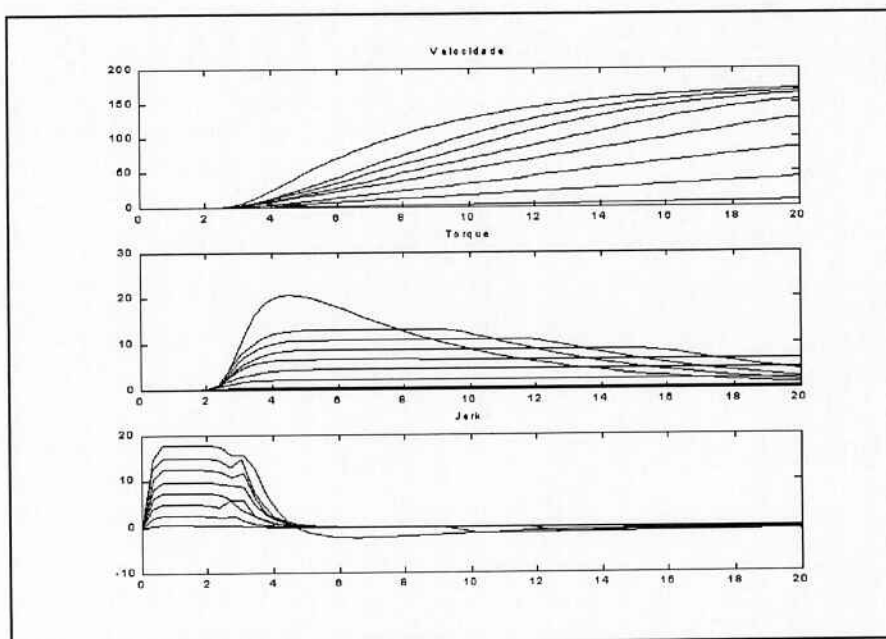
$$\begin{aligned} p &= 1.1667 \\ q &= 0.1667 \\ k &= 0.4944 \end{aligned}$$

A primeira simulação foi feita, limitando-se o Jerk (ω'') e deixando o torque (ω') assumir valores altos.

Desta forma foram obtidos os seguintes sinais de controle para distintas limitações de “Jerk” (1, 5, 10, 15, 20, 25, 30, 50). Lembre-se que os sinais de controle são obtidos via utilização do código “.m”

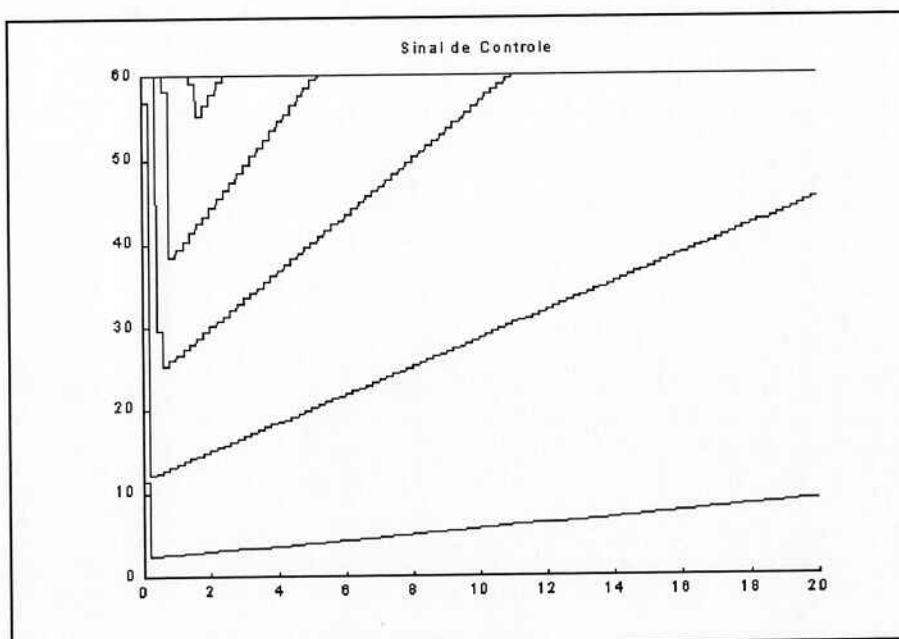


De posse dos sinais de controle, foi simulado o comportamento do sistema no MatLab (SIMULINK), conforme o diagrama de simulação apresentado, e obtidas as respostas ω , ω' , ω'' do sistema:

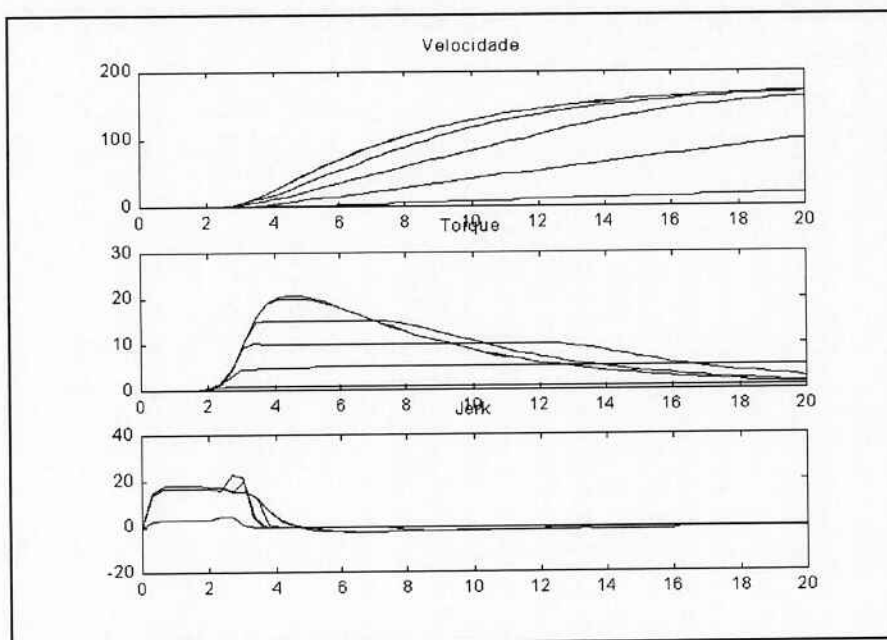


Uma segunda simulação, foi feita, porém agora como Jerk livre, podendo ter valores altos, enquanto o Torque era limitado.

Desta forma foram obtidos os seguintes sinais de controle para distintas limitações de Torque(1, 5, 10, 15, 20, 50)



De posse dos sinais de controle, foi simulado no MatLab (SIMULINK) e obtida as respostas do sistema , ou seja, obteve-se ω , ω' , ω'' :



Uma terceira simulação foi feita, aonde tanto o Jerk, quanto o torque foram “liberados” em valores altos.

Desta forma foi obtido um sinal de controle, o qual é um degrau de amplitude 60Hz.

Os resultados obtidos e apresentados demonstram que a metodologia de formulação e solução do problema realmente atendem as restrições especificadas. Um fato bastante notável é o observado no gráfico das velocidades ω . Quanto maior a restrição imposta, o sinal de controle gerado resulta em um tempo maior para a “partida” do sistema, isto é, é necessário mais tempo para atingir-se o valor da velocidade de regime do sistema.

Antes de prosseguir com a discussão dos resultados, é extremamente importante ser apresentada uma discussão com realização a discretização do sistema.

Vamos considerar o seguinte sistema que possui autovalores reais e negativos:

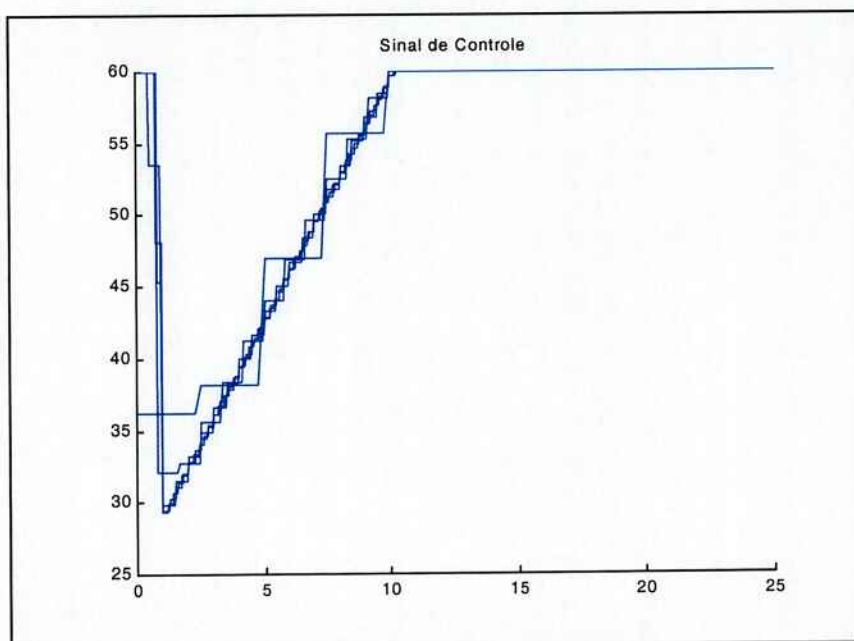
$$\omega'' + 0.6667.\omega' + 0.08333.\omega = 0.2472.f$$

As constantes de tempo são: $\tau_1 = 6$ e $\tau_2 = 2$. Nota-se que $\tau_1 > \tau_2$.

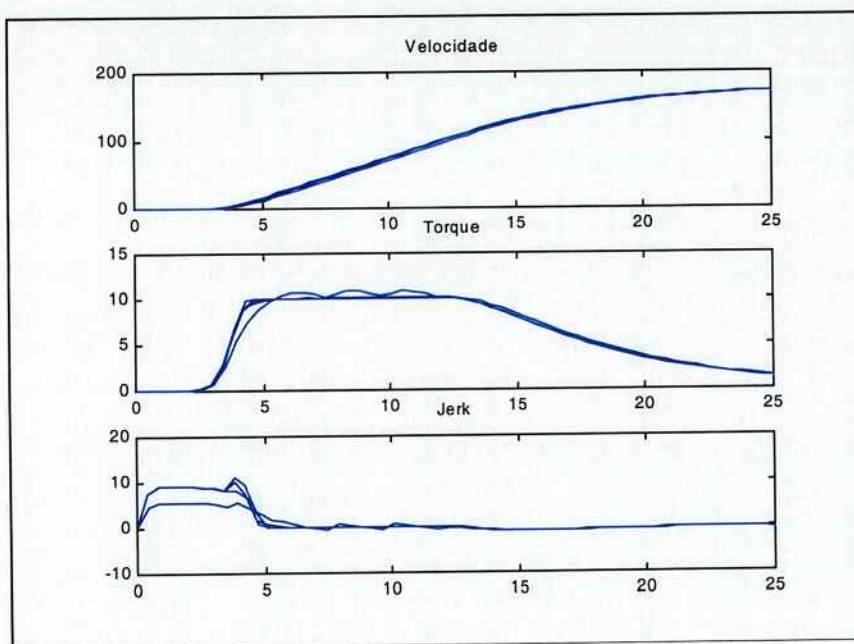
Roda-se a rotina “.m” com o “jerk” limitado em 100 (equivalente a livre) e o torque em 10, com número de pontos $n=10,50,75,100,125$ e tempo final = 25 . Isto significa que a rotina foi utilizada 5 vezes e foram obtidos 5 sinais de controle diferentes com os parâmetros variando conforme descrito:

Configuração 1	Configuração 2	Configuração 3	Configuração 4	Configuração 5
r1=100	r1=100	r1=100	r1=100	r1=100
R2=10	r2=10	r2=10	r2=10	r2=10
N=10	n=30	n=50	N=100	n=125
Tf=25	Tf=25	Tf=25	Tf=25	Tf=25
Tf/n=2.5	Tf/n=0.8333	Tf/n=0.5	Tf/n=0.25	Tf/n=0.2

Assim, os sinais de controle obtidos foram:



No gráfico acima apresentado podem ser visto plotados os 5 resultados. Os gráficos onde $n=10$ e $n=30$ apresentaram como resultado um sinal de controle bastante diferente daqueles onde tem-se valores de n maiores. Percebe-se também uma tendência ao resultado convergir a certo sinal de controle conforme o valor de n vai aumentando. Observe abaixo o resultado da simulação do sistema excitado com os cinco sinais diferentes:



É importante notar, a partir dos gráficos acima, que o o número de pontos utilizado na simulação afeta o resultado obtido no controle. Isso se deve ao fato de que os diversos valores de n , com o valor do tempo final t_f fixo, resultam em diferentes intervalos de discretização (i.e. t_f/n). Pelo critério de Nyquist, sabe-se que o valor ideal para o tamanho do intervalo de discretização é da ordem de 10 vezes menor que a constante de tempo mais rápida. De fato, os resultados convergem para aquele obtido com valores de n suficientemente elevados.

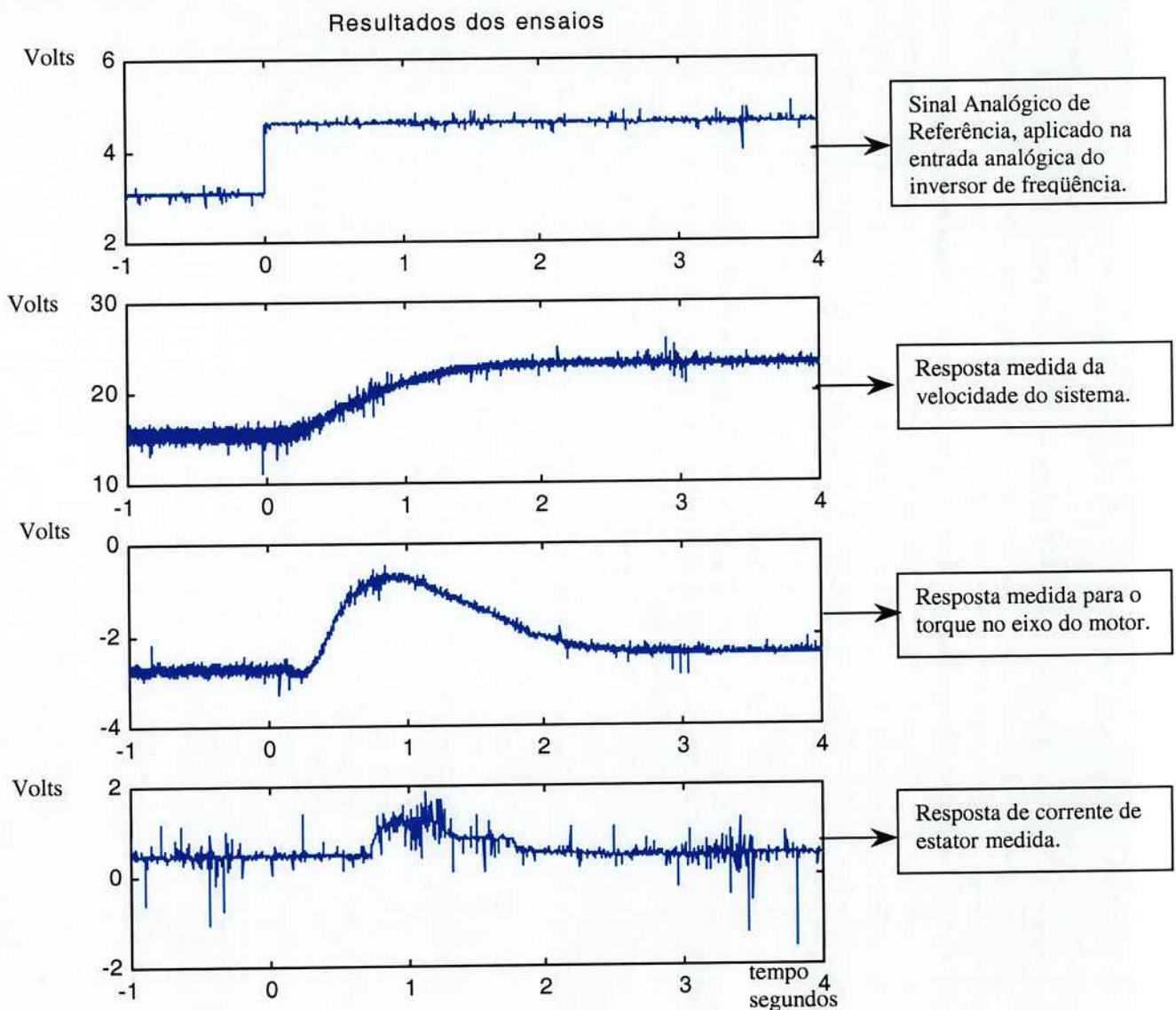
Outro fato que dever ser ressaltado é que no início das simulações, tem-se um valor de “jerk”, porém o torque aparece nulo. Na verdade, o que ocorre é que devido à escala, não é possível visualizar o valor de torque nos instantes iniciais devido ao fato de seu valor ser baixo, o que não significa que sua derivada (“jerk”) o seja. Deve-se notar que o valor do “jerk” é calculado como a derivada do torque pelo MatLab.

Conclui-se com isso a formulação e demonstração de que a metodologia e solução adotada possui bastante consistência e poderá ser usada com segurança para a solução do problema prático proposto considerado. Existe ainda a necessidade de se identificar de maneira razoável que o sistema físico real implementado em laboratório possui comportamento de um sistema de segunda ordem ou , pelo menos, razoavelmente próximo de um sistema de segunda ordem. Tal discussão é feita no item 3.3.4.

3.2.6 Obtenção dos parâmetros p, q e k para aproximação do sistema físico por um sistema de segunda ordem.

Prentende-se neste item, a partir de alguns ensaios realizados com o sistema físico, chegar-se a uma aproximação satisfatória da dinâmica do sistema, segundo a dinâmica de um sistema de segunda ordem. Os ensaios realizados seguiram o esquema de montagem apresentado no item 3.3.2. Conforme apresentado no item 3.3.3, a montagem permite, após instalada e com a lógica “ladder” implementada, enviar-se um “sinal” de controle via CLP que pode variar de 0 á 32767 (valores do sinal no software), sinal esse proporcionais aos valores enviados ao inversor de frequência, através das saída analógica do Controlador Lógico Programável.

O ensaio realizado consistiu basicamente na medição da resposta do sistema a uma variação do sinal de controle. São apresentados abaixo os gráficos dos sinais de interesse no ensaio:



Conforme será explicado no item 5, as medidas dos sinais apresentados acima foram obtidas através da utilização de diferentes instrumentos e métodos. Isso justifica obter-se medidas de velocidade e torque em volts, já que a medida foi retirada de um instrumento que indica o valor da variável em questão segundo saídas de tensão.

A partir dos resultados da simulação, obteve-se os parâmetros para o modelo de segunda ordem do tipo:

$$G(s) = \frac{\kappa \cdot \omega_n^2}{s^2 + 2 \cdot \omega_n \xi + \omega_n^2}$$

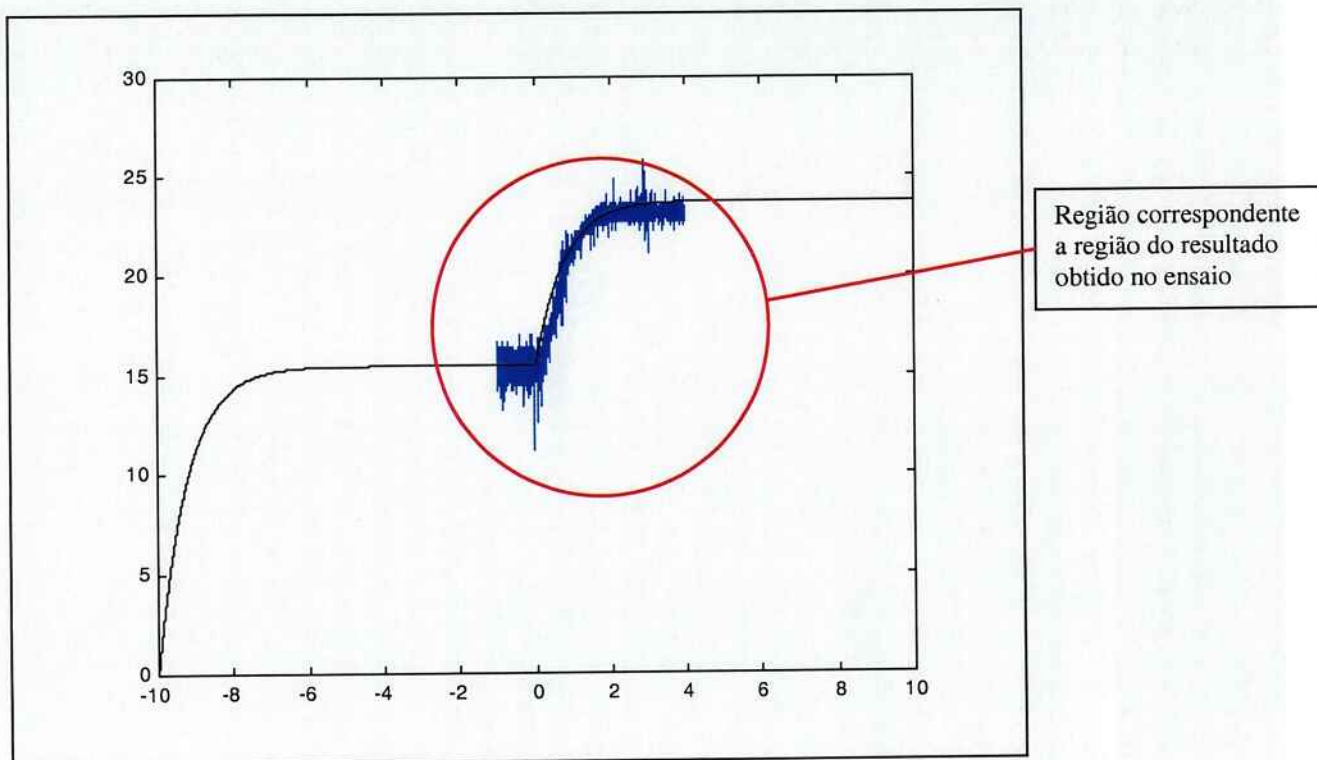
onde s é a variável de Laplace e $G(s)$ indica função de transferência para o sistema.

Vale ainda observar que o sistema foi colocado inicialmente em regime em um estado diferente do inicial (o que pode ser visto 2º gráfico apresentado acima) e então foi aplicado o degrau demonstrado no gráfico nº 1. Importante lembrar também que este ensaio considerou todo o sistema implementado, pode-se ter uma idéia do comportamento de toda a montagem junta.

Partindo-se de uma aproximação inicial para os parâmetros do modelo, e com os conhecimentos do comportamento de sistemas de segunda ordem conforme os valores de ω_n e ξ , pôde-se, após algumas iterações obter os seguintes parâmetros:

$$\omega_n = 70 \quad \xi = 27 \quad k=5.15$$

Para comprovar esses valores são apresentados abaixo os gráficos da resposta do modelo (utilizando-se o software “Simulink”) a uma mesma excitação que aquela aplicada no sistema durante o ensaio em laboratório, confrontada com os resultados obtidos em laboratório. Observe:



É fácil notar que o modelo obtido é bastante próximo da realidade, isto é, da dinâmica do sistema real considerado. Tal fato observado tem valor bastante expressivo para o projeto já que é uma prova real de que o acoplamento do motor AC com o inversor de frequência efetivamente nos permite considerar o sistema como sendo um sistema de segunda ordem linear, hipótese inicial básica assumida e, agora, comprovada como sendo razoável.

3.3 Implementação- Especificação

Após a escolha técnica de que a implementação seria feita utilizando-se corrente alternada, e por consequência, inversores de frequência, viu-se necessário, especificar, e decidir quais equipamentos seriam utilizados, bem como outros detalhes necessários para a implementação, mas que a princípio não seriam foco deste projeto.

3.3.1 Problemas Mecânicos

A construção do sistema do elevador envolve alguns problemas de cunho mecânico tais como redução de rotação do motor elétrico, torque, manuseio dos cabos de fixação entre outros.

Para a solução destes problemas, buscou-se encontrar no mercado soluções já prontas, e que onerassem da menor forma possível o projeto.

Ápos a realização de uma pesquisa com as soluções existentes no mercado, bem como os seus custos, viu-se que nenhuma das soluções disponíveis era compatível com o orçamento e com o tempo para a realização do projeto.

Então decidiu-se por utilizar um sistema já existente na EPUSP que pode simular um elevador.

Este sistema consiste de um motor elétrico que tem em seu eixo acoplado uma inércia.

Esta inércia terá a função de simular os efeitos que o elevador causaria ao motor e ao seu controle.

Dessa forma poderá ser construído o projeto sem nenhuma perda em termos de desenvolvimento do controle e de acionamentos.

Abaixo pode-se ver uma foto do motor e sua inércia.



Figura 4- Inércia

Este sistema também é constituído de um motor de Corrente Contínua e um Torquímetro, que são extrema importância na medição dos dados para a avaliação da performance do sistema.

3.3.2 Especificação dos Equipamentos(Inversor de Frequência e CLP)

Para se especificar o inversor de frequência, foi necessário analisar as condições de operação do sistema.

O primeiro ponto de especificação era de que o inversor deveria suportar o controle em modo escalar, uma vez que esta solução é extremamente adequada para esta aplicação, como já foi discutido anteriormente.

O segundo passo dentro da análise, era de que, a montagem de um elevador, exigia que o inversor fosse de “quatro quadrantes”, característica fundamental em elevadores.

Dentro destes dados, escolheu-se utilizar o inversor de frequência GD 4000(figura 5) da ALSTOM, recentemente doado à EPUSP, que já incorpora um CLP(Controlador Lógico Programável), e que contempla todas as características necessárias para a aplicação em elevadores, além de como citado acima, já possuir um controlador, o que nos traria benefícios de fiação e tempo de montagem.

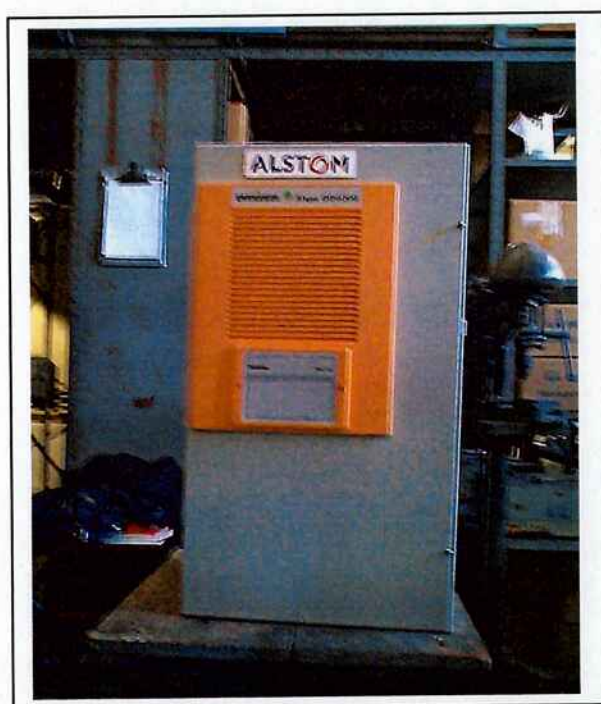


Figura 5-GD 4000 Fechado

Porém devido a problemas técnicos que surgiram durante o trabalho de parametrização(falta de um contator, que não poderia ser substituído em tempo hábil), teve de se alterar o inversor, e por consequência, o CLP utilizado.

Nesta fase do trabalho, já havia sido feita a alteração no projeto, na qual não seria realizada a construção física do elevador, e sim, tão somente a utilização do sistema motor+inércia comentado anteriormente.

Desta forma foi escolhido a utilização do conjunto inversor de frequência GV3000 da Reliance Electric(Figura 6) e do CLP SLC 500 da Allen-Bradley (Figura 7), ambos disponíveis na Escola Politécnica-PEA.



Figura 6- GV3000

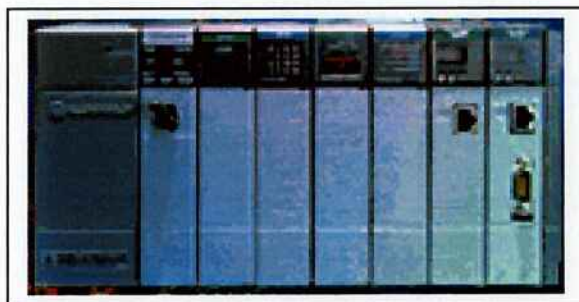


Figura 7- SLC 500

O GV3000 não opera em quatro quadrantes, mas em apenas dois.

Como a inércia conectada ao motor da nova montagem é de valor extremamente elevado e pelo fato do inversor não operar em quatro quadrantes, decidiu-se por medida de precaução e para viabilizar a implementação, que seria colocada uma desaceleração extremamente lenta para evitar um “trip” do equipamento durante a operação de parada.

Outra alteração com relação à montagem inicialmente planejada, seria a existência de fiação entre o CLP e o inversor. Esta fiação consiste na referência analógica de velocidade e do sinal de início do processo.

O esquema de montagem esta abaixo na figura 8.

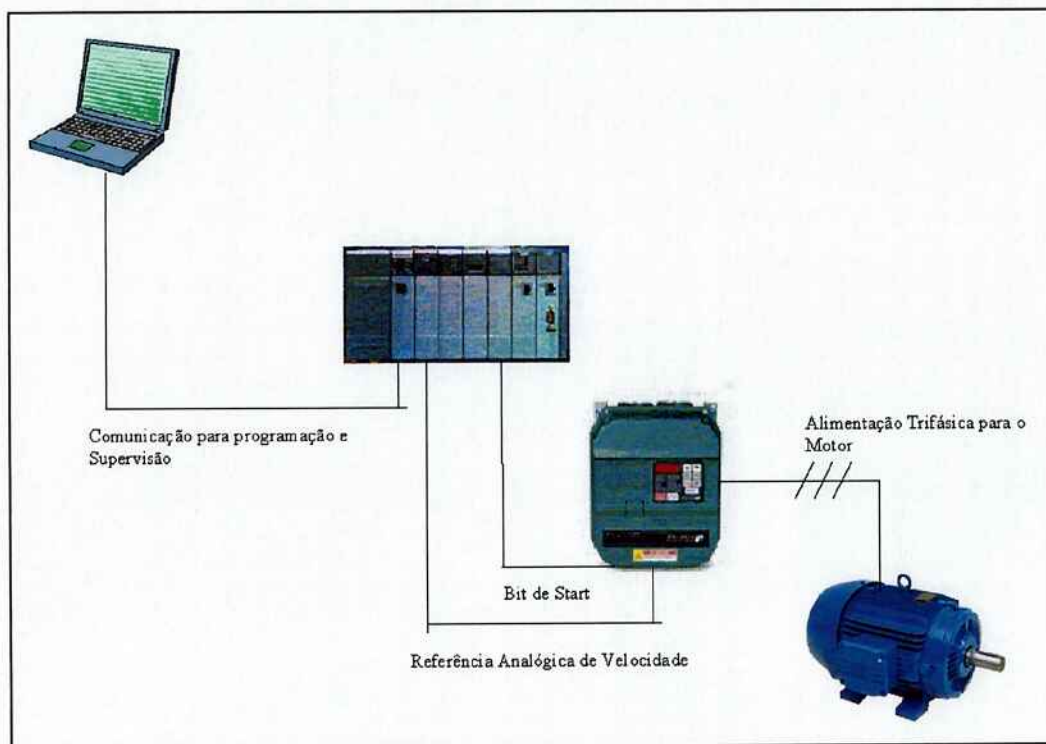


Figura 8-Esquema de Montagem

3.3.3 Parametrização e Programação

Para se realizar a implementação, era necessário a parametrização do inversor de frequência, para que desta forma, este possui-se os dados necessários para desenvolver a performance desejada.

Alguns dados elétricos relevantes do nosso sistema eram:

- Motor de alimentação trifásica, 220V
- Potência nominal do motor de 3HP
- Corrente Nominal de 9,4 A
- Frequência de Alimentação da rede de 60Hz

- Frequência nominal do Motor de 60Hz
- Modo de Controle: Volts/Hertz(escalar)

Numa primeira fase, parametrizou-se o inversor GD4000.

Este inversor possui algumas características peculiares que valem a pena serem comentadas.

Como este inversor já possui um CLP acoplado a ele, toda a parametrização é feita através deste, via comunicação serial, e todas as informações são passadas ao inversor junto do Ladder de programação deste CLP.

Outro detalhe relevante sobre este inversor possui um número elevado de parâmetros envolvidos. Sua parametrização envolve não só parâmetros elétricos do sistema em questão, como também os tempos de disparo nos tiristores, tempo de disparo nas pontes, valores de tensão de DCLink, acionamento dos contadores internos do inversor, tempos de alarmes de falhas entre muitos outros parâmetros.

Esta parametrização custou um longo tempo de trabalho, aonde chegou-se a parametrizar cerca de 1000 distintos parâmetros. Porém, não foi possível testar a operacionalidade deste inversor, uma vez que por problemas já descritos.

Já numa segunda fase, parametrizou-se o inversor GV3000.

Este inversor possui uma construção mais simples, uma vez que sua operação independe da utilização ou não de um CLP.

Desta forma, ele já possui de fábrica uma pré-configuração dos parâmetros não ligados ao sistema, fazendo com que o nosso foco fosse a parametrização de dados elétricos.

Todo este trabalho foi feito de forma manual, através da interface homem-máquina do aparelho.

Após o trabalho no inversor, iniciou-se o estudo para a implementação da rotina no CLP.

O CLP SLC 500 é programável em linguagem Ladder, através do software RSLogix 500.

A linguagem Ladder tem como objetivo programar de forma semelhante aos antigos diagramas elétricos.

Primeiramente configurou-se no software, o modelo da CPU utilizada. No caso uma CPU 1747-L524 com 4K de memória.

Depois configurou-se os cartões de entrada e saída conectados ao barramento, para o interfaciamento com o processo(figura 9).

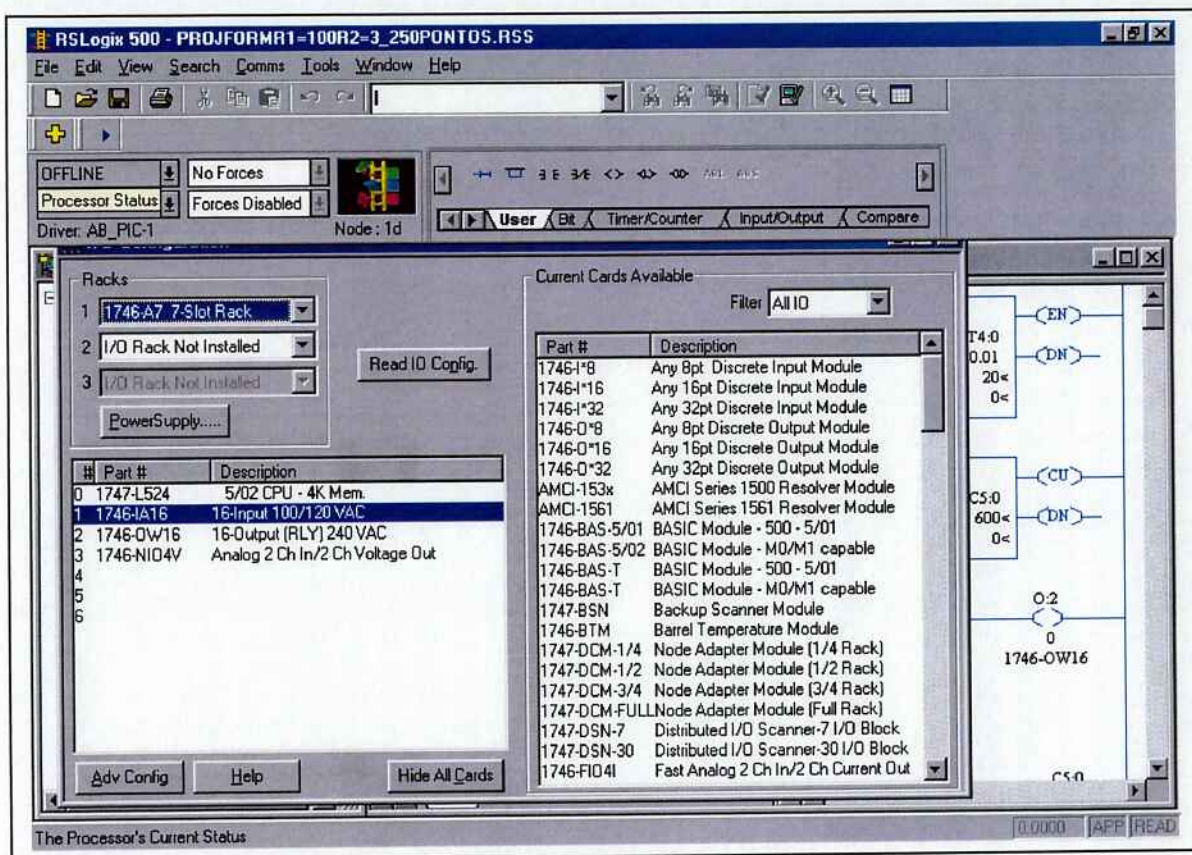


Figura 9- Configuração de entradas e saídas

Desta forma foi possível iniciar a implementação da rotina explanadas entre os itens 3.2.1 até o item 3.2.5.

A idéia básica é a de enviar ao inversor, em intervalos pré-definidos, uma série de valores de referências de velocidade. Por exemplo, a cada 0,5s, o CLP deve enviar uma nova referência de velocidade.

Estes valores foram calculados anteriormente a implementação, como já foi explicado, através de uma rotina em MatLab.

A rotina programada foi feita da seguinte forma:

Dado um bit de início, um temporizador inicia a contagem do período de tempo pré-estabelecido(figura 10).

A cada vez que esta contagem termina, um bit de "done" é sinalizado por este temporizador, que aciona um contador, que passa a contar progressivamente(figura 10).

O valor deste contador, é usado como comparação pela rotina, para que esta saiba qual o valor que deve ser enviado para a saída. Este valor é enviado a uma variável interna como mostrado na figura 11.

Finalmente este valor é enviado para a saída analógica do CLP(figura 12).

É importante ressaltar que os valores dos sinais vindos do MatLab são de 0 á 10V, mas o CLP, por trabalhar com 16 bits, envia para as suas saídas analógicas valores entre 0 e 32767. Devido a este fato, foi necessário a conversão deste fatores para esta escala.

As figuras seguem na sequência:

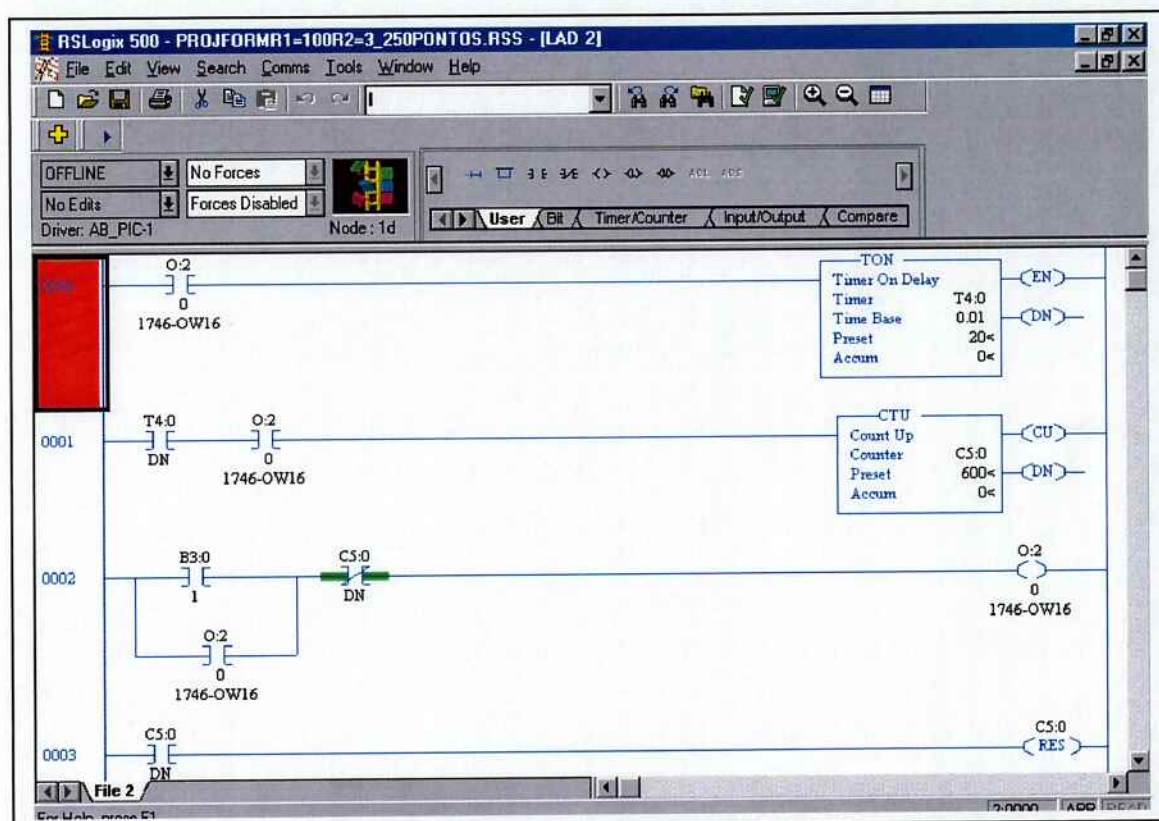


Figura 10- Ladder, Contador e Temporizador

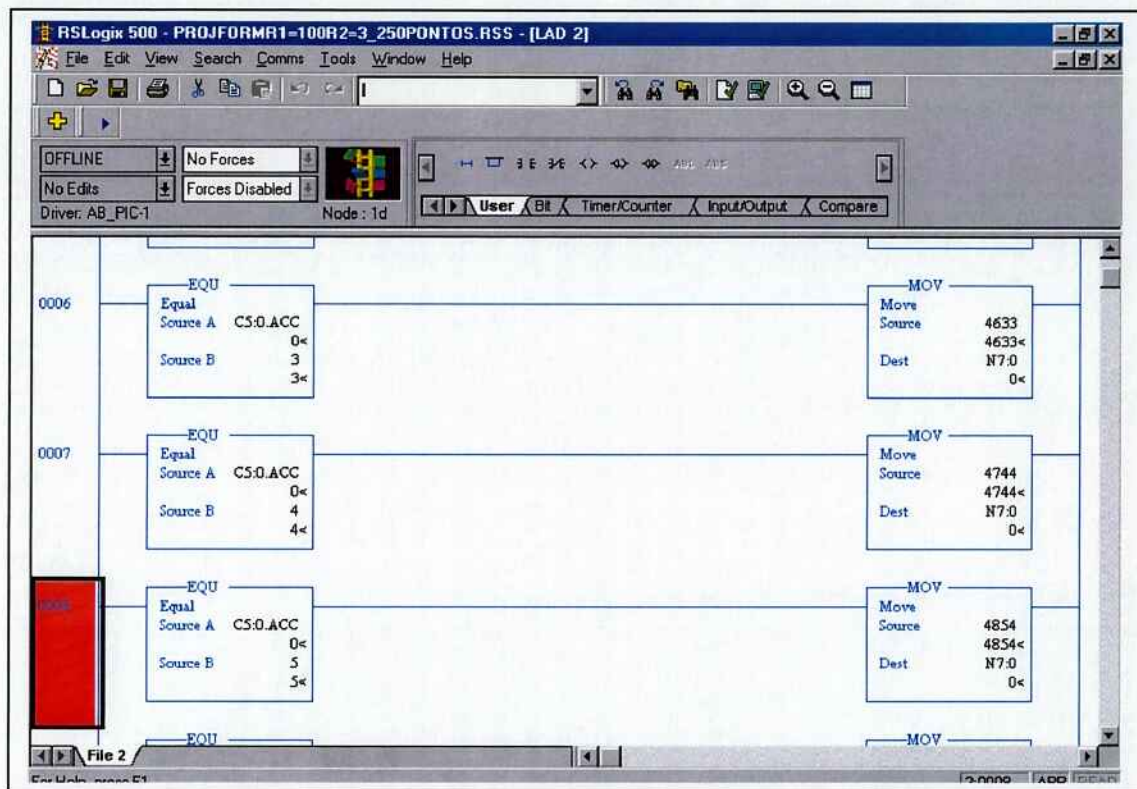


Figura 11-Comparação dos valores do contador

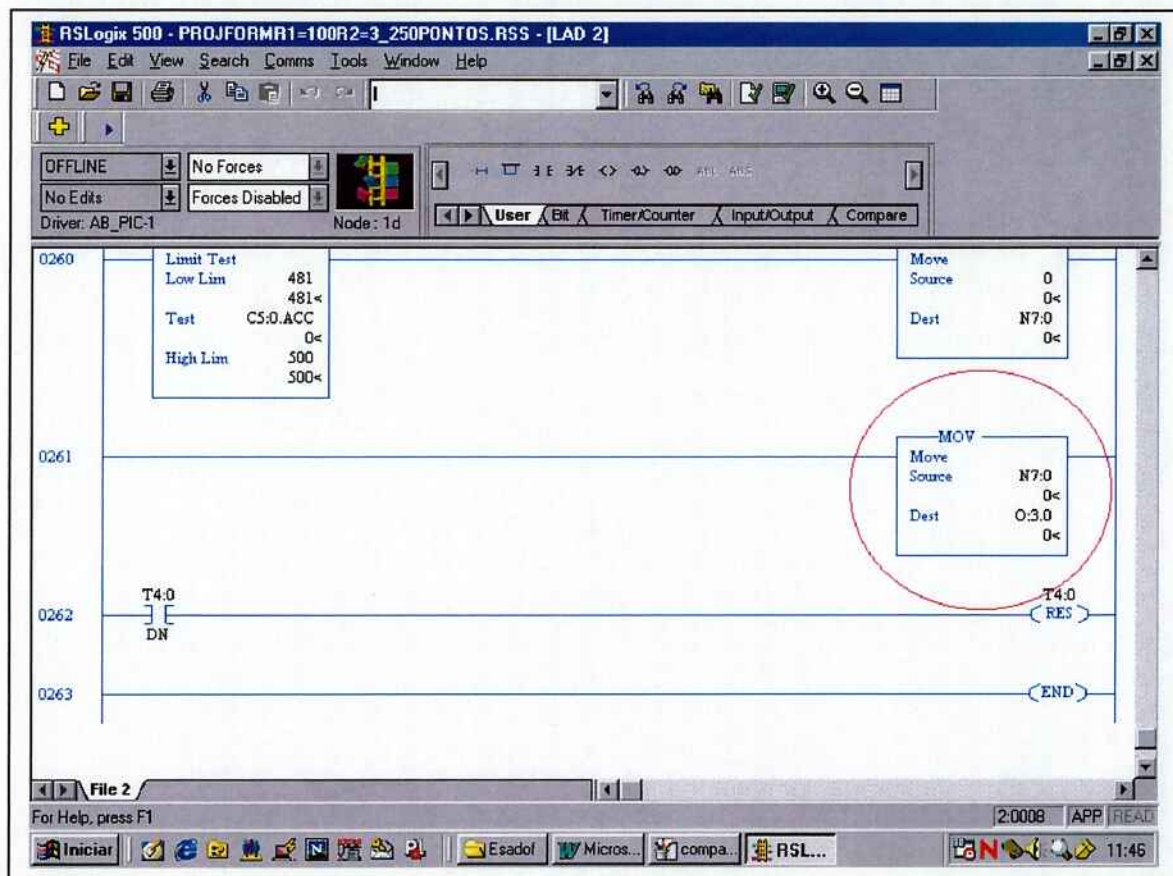


Figura 12-Envio do sinal ao inversor

Com o término do desenvolvimento forma realizados uma série de ensaios para aferir os resultados. Estes testes estão melhor descritos no item 5.

4. Dados, premissas e critérios utilizados

No desenvolvimento do estudo e implementação de sistema de controle para elevador, é necessário fazer algumas premissas.

Como se estará utilizando alguns equipamentos já desenvolvidos, como o CLP e o inversor de frequência, espera-se que estes tenham comportamento semelhante ao descrito em seus manuais, principalmente no que tange os aspectos de tempo de resposta a um comando programado.

5. Resultados Obtidos

Após o trabalho de desenvolvimento da solução do problema de controle ótimo, e de sua implementação, decidiu-se por comparar, os resultados de simulações do sistema, com os resultados obtidos através do sistema real, quando aplicado o algoritmo desenvolvido.

A montagem realizada no laboratório consistia, além do sistema já explicado no itens 3.3.2 e 3.3.3, de osciloscópio de 4 canais digital e microcomputador com software de aquisição de dados.

Seguem a seguir fotos da montagem experimental no laboratório.



Figura 13- Osciloscópio



Figura 14-Montagem Inversor+CLP+Micro

O ensaio consistiu na implementação, via Ladder, de três rotinas simuladas no MatLab conforme descrito mais abaixo neste item.

Foram medidos valores de velocidade, torque, corrente, além do sinal de referência de velocidade do CLP para o inversor

A velocidade foi medida através de um motor de corrente contínua acoplado ao motor AC controlado.

Isso é possível, pois a FCEM de um motor CC é proporcional a velocidade de rotação como se pode ver na equação abaixo:

$$E=k.\phi.\omega$$

Pode-se considerar k constante.

Dada que a excitação(fluxo) é independente e constante, pode-se afirmar que a tensão E é diretamente proporcional à rotação

Dessa forma, é possível se medir a forma da curva de velocidade para posterior comparação com a simulação do sistema.

O torque foi medido através de um torquímetro acoplado no eixo do motor AC. Esse torquímetro possui uma saída analógica de 0 á 10Volts, indicando o valor do torque.

A medição do valor da corrente foi feita através da saída analógica do inversor de frequência, também de 0 á 10Volts.

A referência de velocidade enviada do CLP para o inversor foi medida através da saída analógica do CLP, também um sinal de 0 á 10 Volts.

As rotinas implementadas para o sistema do laboratório, desenvolviam partidas em três distintos tempos, cinco, nove e quinze segundos.

Quanto menor o tempo, maior o limite de torque permitido para o desenvolvimento da aceleração.

Para o ensaio de partida em quinze segundos, foram colocados valores de limitação para Jerk de 10^{10} e para aceleração(proporcional ao torque) de 3.

Obteve-se o resultado abaixo:

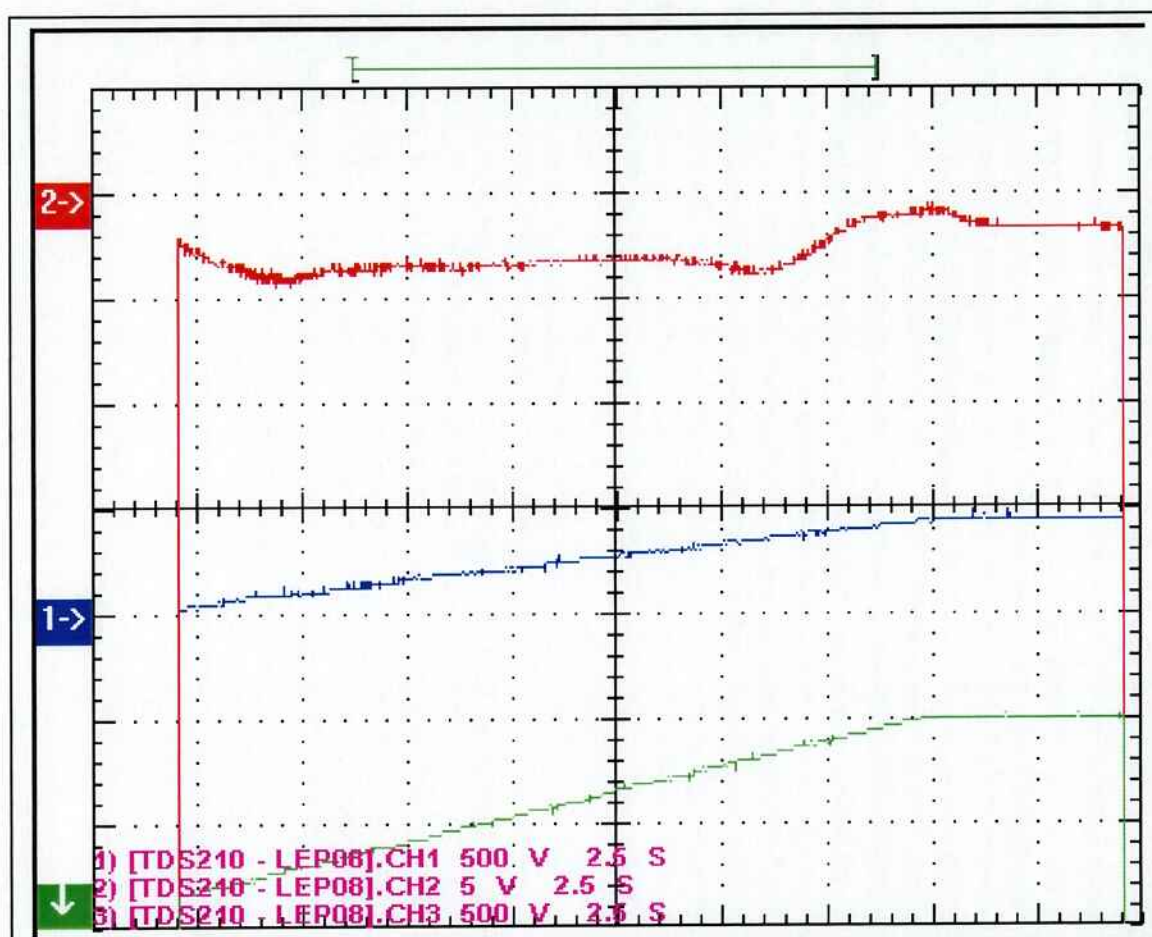


Figura15- Resultados dos ensaio para t=15 segundos

Aonde o canal 1(azul), é a velocidade do motor, o canal 2(vermelho) é o torque desenvolvido com o sinal invertido e o canal três(verde) é o sinal de referência vindo do CLP.

Nota-se que o torque se comporta praticamente constante durante toda a trajetória de aceleração.

Foi simulado no software Simulink o comportamento deste mesmo sistema, quando aplicado o sinal de controle ótimo.

Obteve-se como resposta os gráficos abaixo:

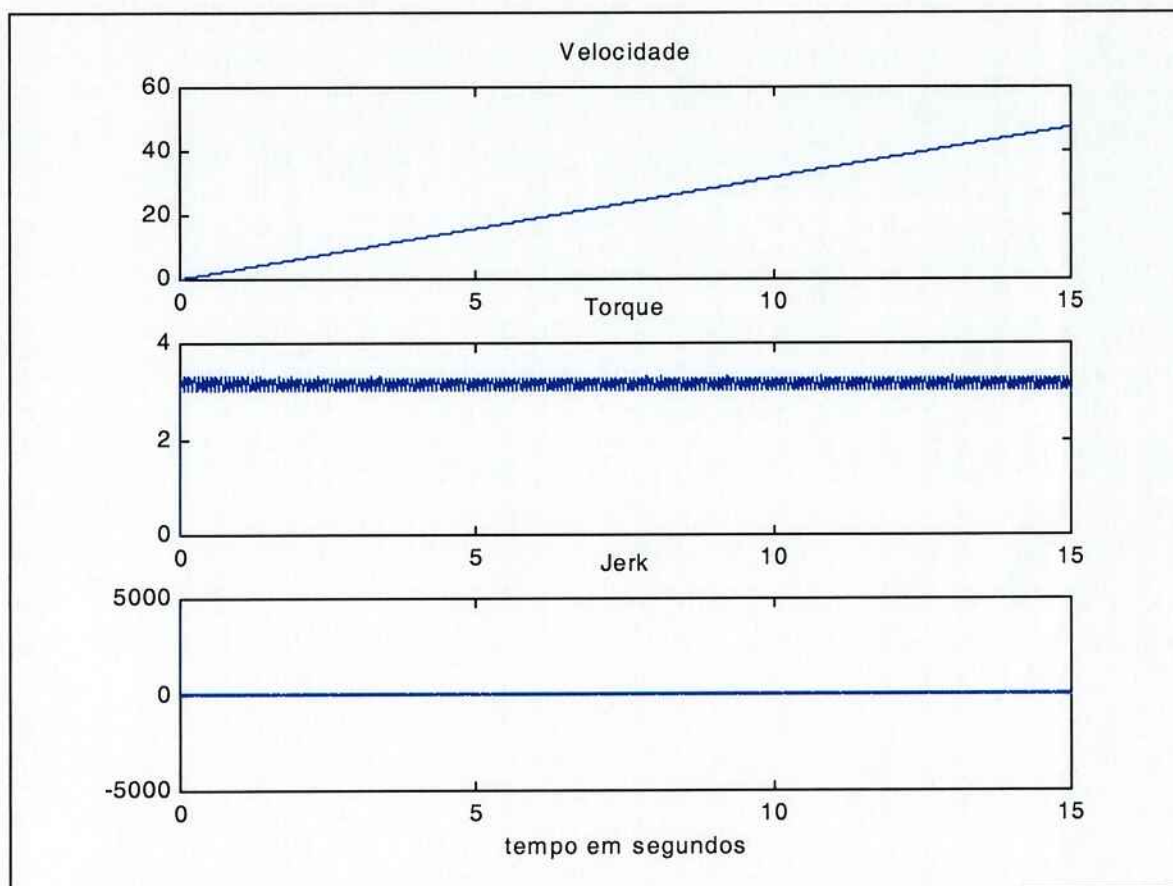


Figura 16-Saída da Simulação para partida em 15 segundos

Um segundo ensaio realizado foi o de partida em 9 segundos, cujo os resultados seguem abaixo:

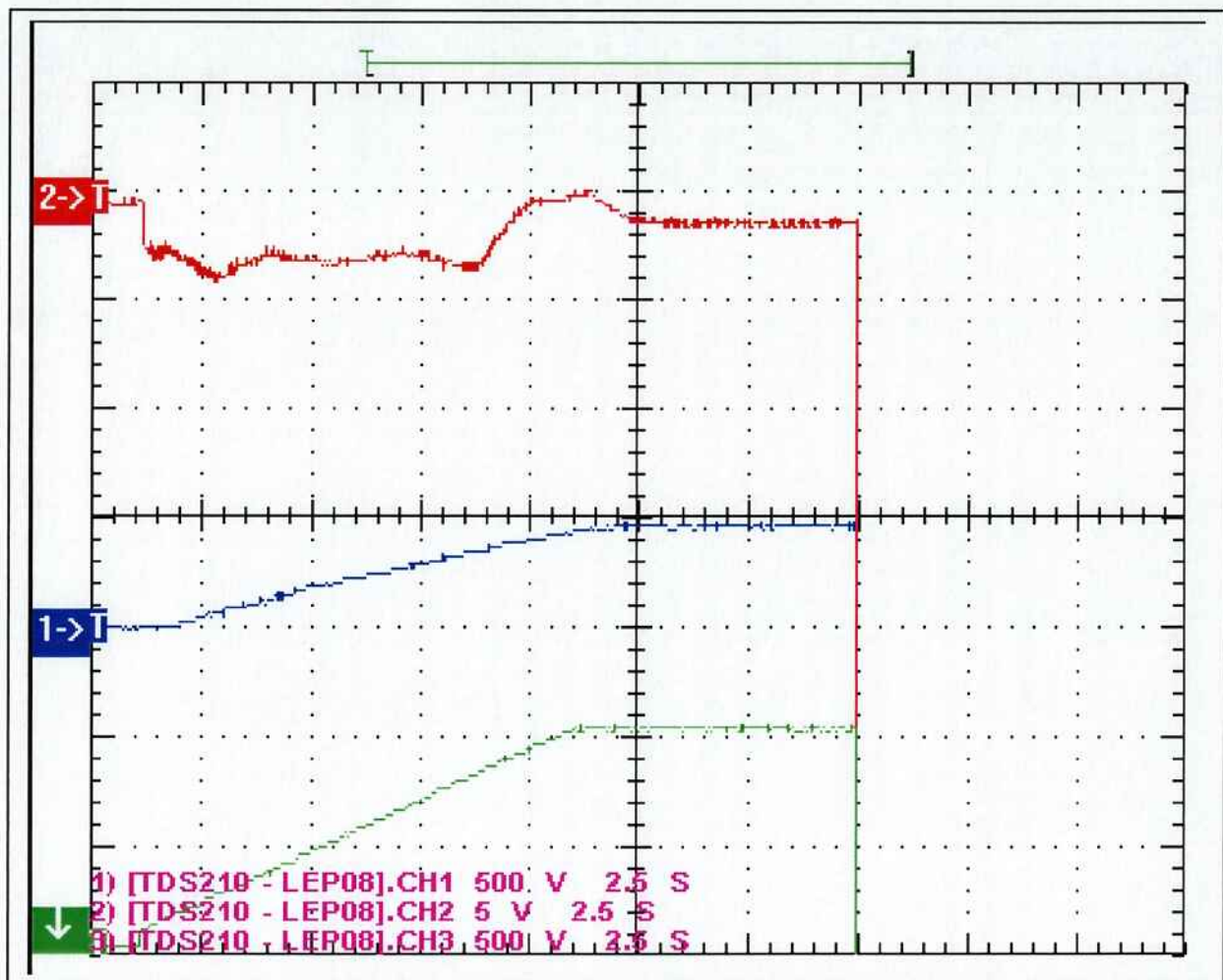


Figura 17- Resultados dos ensaios para $t=9$ segundos

Aonde o canal 1(azul), é a velocidade do motor, o canal 2(vermelho) é o torque desenvolvido com o sinal invertido e o canal três(verde) é o sinal de referência vindo do CLP.

Nota-se novamente que o torque se comporta praticamente constante durante toda a trajetória de aceleração.

Foi simulado no software Simulink o comportamento deste mesmo sistema, quando aplicado o sinal de controle ótimo.

Obteve-se como resposta os gráficos abaixo:

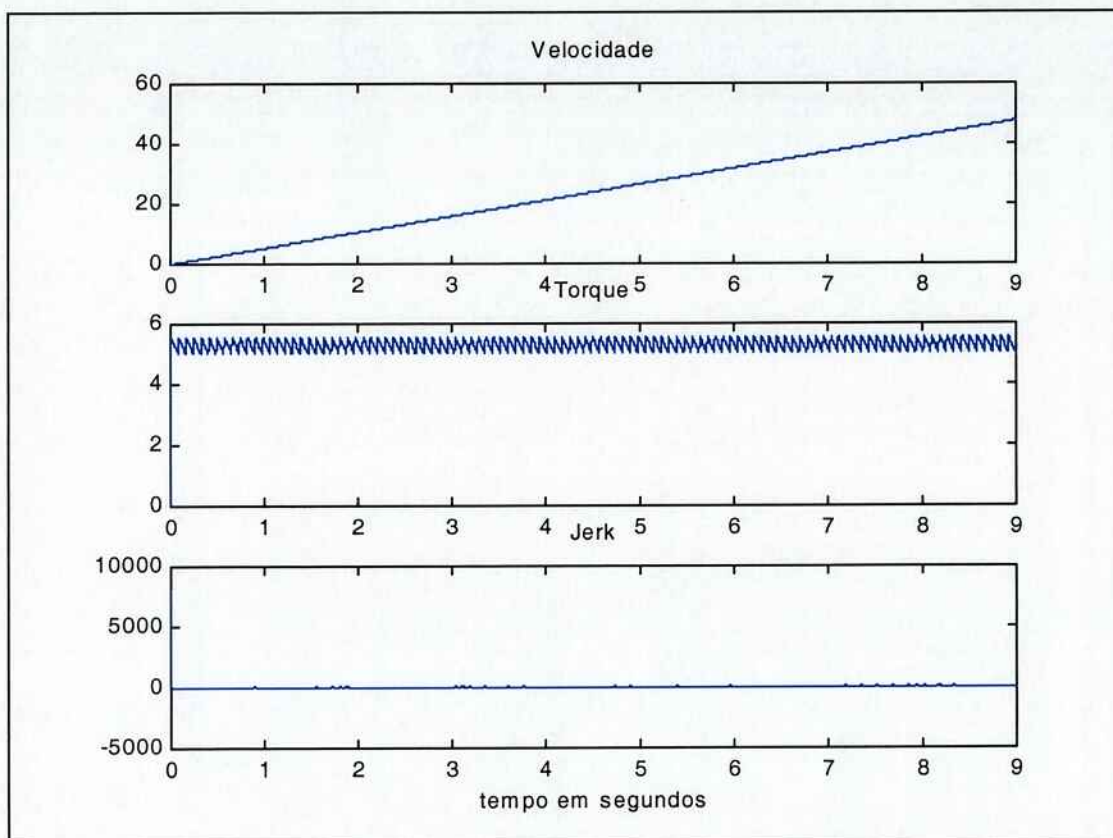


Figura 18- Saída da Simulação para partida em 9 segundos

O terceiro ensaio realizado, tinha como objetivo a partida do motor em cinco segundos.

Durante esta partida verificou-se uma série de esforços mecânicos indesejados muito provavelmente devido á rápida aceleração da máquina, além de vibrações devido a desbalanceamento mecânico do sistema.

Devido a isso a partida se realizou em pouco menos de 6 segundos, porém com o torque controlado como se pode observar abaixo:

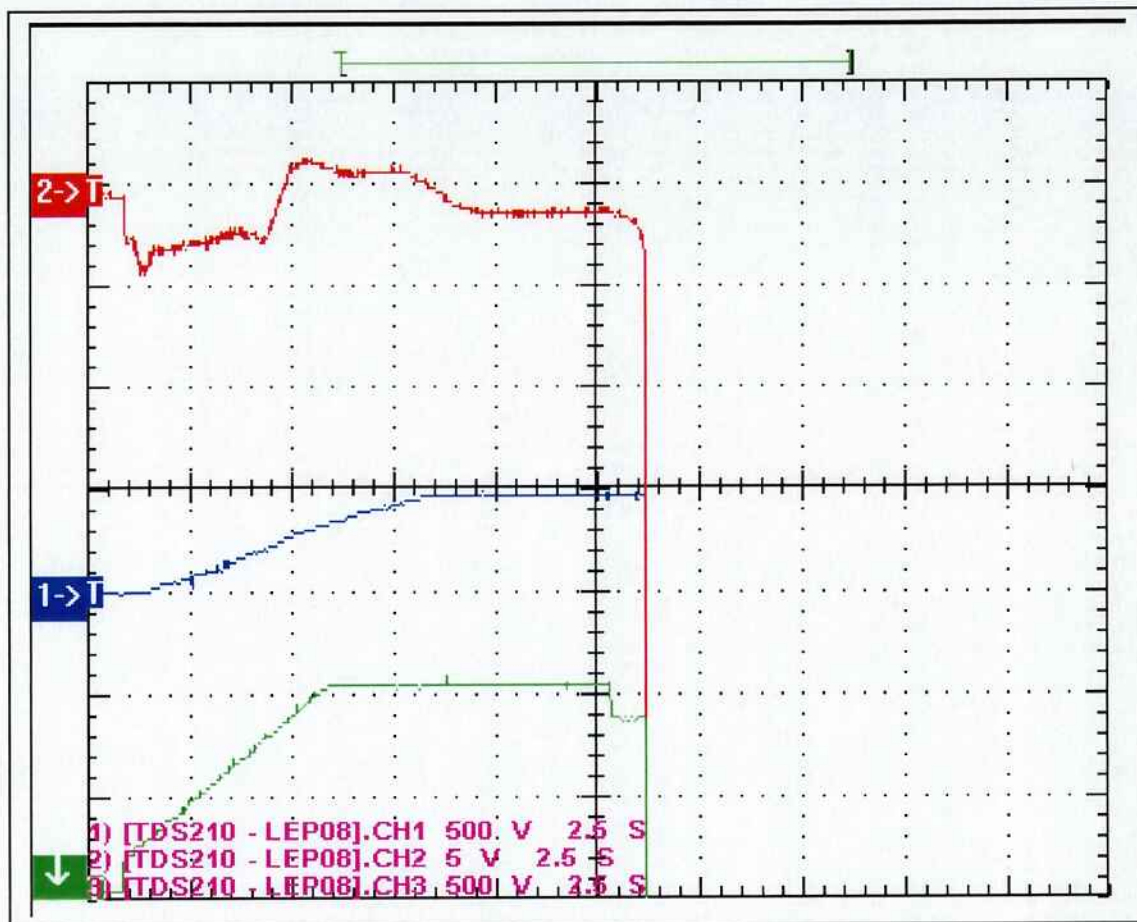


Figura 19- Resultados dos ensaio para $t=5$ segundos

Aonde o canal 1(azul), é a velocidade do motor, o canal 2(vermelho) é o torque desenvolvido com o sinal invertido e o canal três(verde) é o sinal de referência vindo do CLP.

Foi simulado no software Simulink o comportamento deste mesmo sistema, quando aplicado o sinal de controle ótimo.

Obteve-se como resposta os gráficos a seguir:

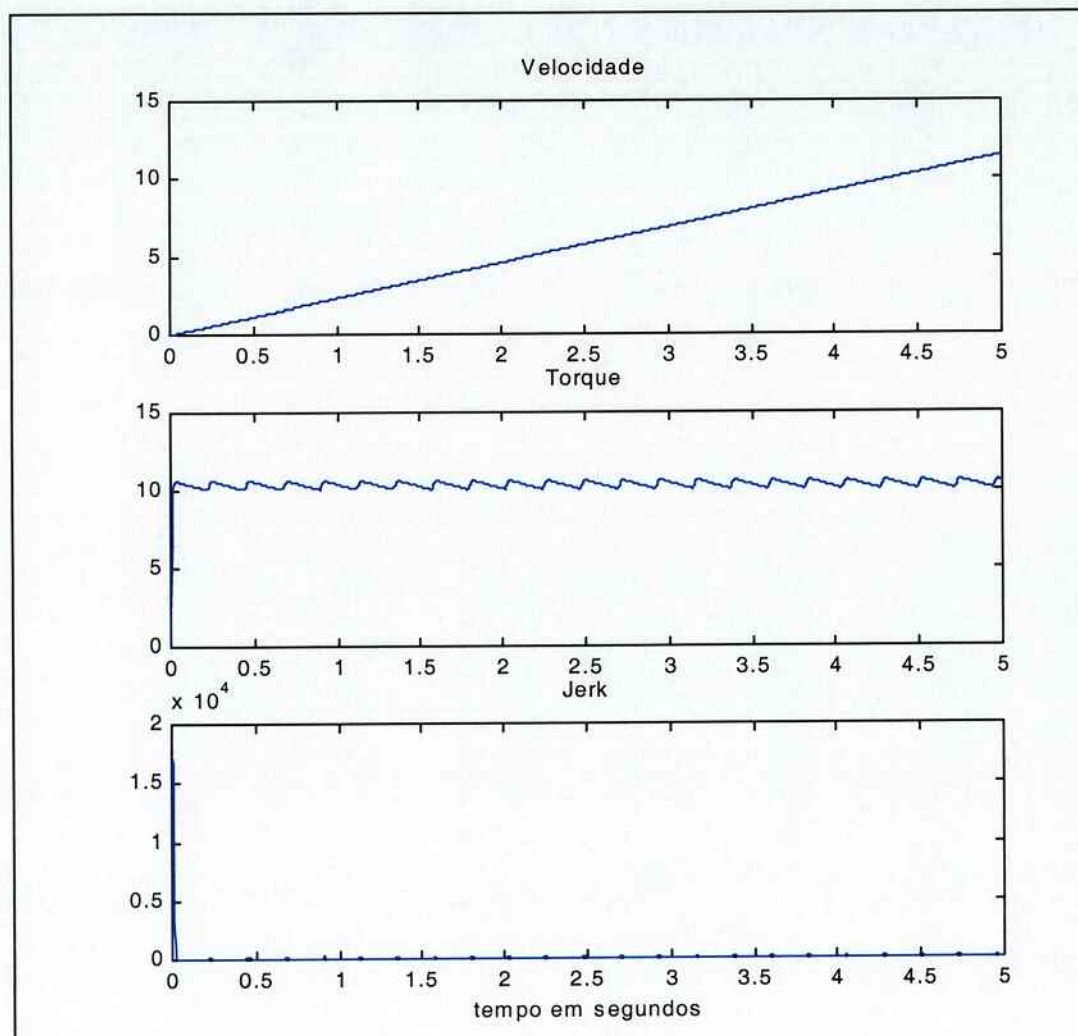


Figura 20- Saída da Simulação para partida em 5 segundos

Por último, para que se possa visualizar ainda melhor o comportamento do torque, mediu-se a velocidade e o torque durante uma partida direta.

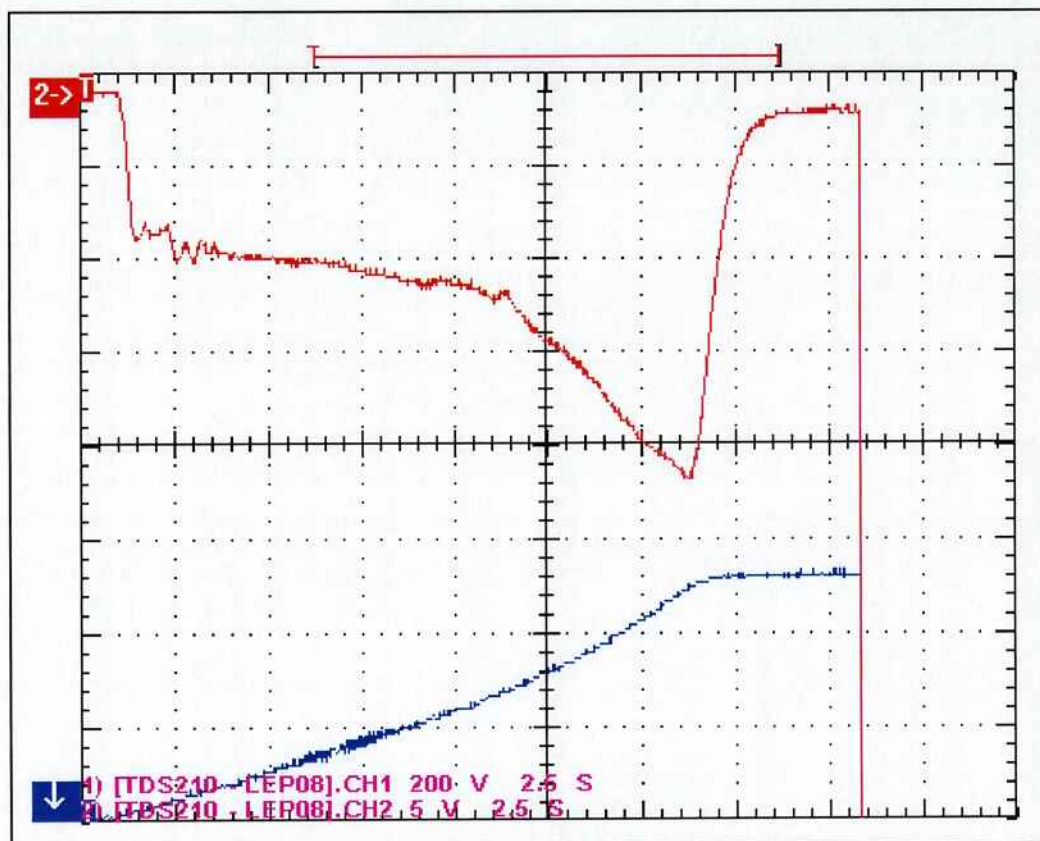


Figura 21- Partida Direta

É fácil de se notar no gráfico acima que quando não há nenhuma forma de controle sobre o motor, o torque não é “bem comportado”.

Devido á problemas no medidor de torque, não é possível saber exatamente o valor do torque em cada um dos resultados.

Para que se possa comprovar os resultados decidiu-se por estimar a grandeza do torque através da derivada da velocidade(proporcional ao torque).

Ao aproximarmos linearmente este sinal, obtemos para os sinais, as seguintes respostas:

-Partida em $t=15s$, $C \propto (185/7,5)=24,66$

-Partida em $t=9s$, $C \propto (100/2,5)=40$

-Partida em $t=5s$, $C \propto (200/2,5)=80$

Logo qualitativamente pode-se concluir que o torque medido para a partida em 9s, foi 1,62 vezes maior que para a partida em 15s, enquanto o esperado era de 1,66 vezes e que o torque para a partida em 5s foi duas vezes maior que para a partida em 9s, como era esperado.

Com isso pode-se concluir que os resultados obtidos foram extremamente satisfatórios dentro dos objetivos inicialmente traçados.

6. Conclusões

Primeiramente pode-se afirmar que o projeto atingiu as metas planejadas na sua concepção, ou seja, implementar uma lógica de controle para elevadores, com a característica diferencial de otimização de tempo de percurso, dadas as restrições de desempenho, e sua construção prática com inversores de frequência para que desta forma fosse possível a validação das premissas.

A partir do que foi desenvolvido, vislumbrou-se a possibilidade de implantação deste protótipo em um elevador real.

Para se gerar o sinal de controle é necessário se conhecer a cada início de viagem, as características mecânicas do elevador.

Sabe-se, que em um elevador residencial, por exemplo, o seu funcionamento é dividido em carga do elevador com os passageiros e percurso de sua origem ao seu destino. Percebe-se então que a carga do elevador só se altera nos inícios de viagem.

Logo, para se implementar a abordagem proposta no caso que a carga varia, segundo descrito acima, bastaria gerar um número n de sinais de controle ótimo, um para cada situação de carga(dadas as limitações inferior e superior do elevador), armazená-los na CPU do CLP e selecionar a melhor curva no início de cada viagem dada uma medição da situação mecânica naquele instante.

Desta forma seria possível a utilização desta técnica para se iniciar uma abordagem de implementação real implementação real.

Dadas todos os pontos discutidos, gostaríamos de sugerir possíveis continuações deste estudo, o que incluiria técnicas de “malha fechada” para aumentar a confiabilidade, assim como o estudo de métodos para a medição e posterior seleção da “curva” conforme já descrito.

7. Bibliografia

- [1] Ogata, Katsuhiko, 1990:Engenharia de Controle Moderno edição 3, Ed. PHB, Estados Unidos
- [2] Beer, Ferdinand P., Johnston, E. Russel, 1994 Mecânica Vetorial para Engenheiros 5a. Edição, Ed. McGrawHill
- [3] Manual de Técnico “GD 4000”, 1998, Publicação T1656, Alcatel Alstom
- [4] W. Leonhard, 1985 Control of Electrical Drives, Berlin Heidelberg New York Tokyo
- [5] Manual de Técnico “GV 3000”, 1997,Reliance Electric, Rockwell Automation
- [6] Goldemberg, Clóvis, Apostila de Acionamentos AC, versão 1.4, Novembro de 1999

8. Agradecimentos

Este projeto só foi possível graças a colaboração do professores doutores Clóvis Goldenberg, docente do Departamento de Energia e Automação Elétricas da Escola Politécnica da USP e orientador deste projeto e do professor Dr. José Jaime Cruz, docente do LAC da Escola Politécnica da USP.

Também colaboraram os técnicos responsáveis pelo almoxarifado do PEA, com seu suporte durante as experiências de Laboratório

Por último gostaríamos de agradecer a todos os professores, tanto do PEA, quanto do LAC pelo suporte, em especial aos envolvidos com a disciplina de Projeto de Formatura, que permitiram este projeto interdepartamental.

9. Biografia

Marcelo Tarkieltaub, natural de São Paulo, Brasil, nascido no dia 12 de novembro de 1979. Cursou o segundo grau no Colégio I.L. Peretz em São Paulo.

Aluno de graduação do curso de Engenharia de Energia e Automação Elétricas da Escola Politécnica da Universidade de São Paulo, formando em 2001. Estagiou no LPS da Escola Politécnica da Universidade de São Paulo, na Rockwell Automation do Brasil e, atualmente continua trabalhando na Rockwell Automation do Brasil

Felipe Vivacqua Pinto, natural de São Paulo, Brasil, nascido no dia 19 de maio de 1979. Cursou o segundo grau no Colégio Dante Alighieri em São Paulo.

Aluno de graduação do curso de Engenharia de Automação e Controle da Escola Politécnica da Universidade de São Paulo, formando em 2001. Estagiou no LSI da Escola Politécnica da Universidade de São Paulo, na Accenture do Brasil(antiga Andersen Consulting) e, atualmente trabalha no Banco Itaú S.A

10. Anexos

Anexo 1- Sistema Operando

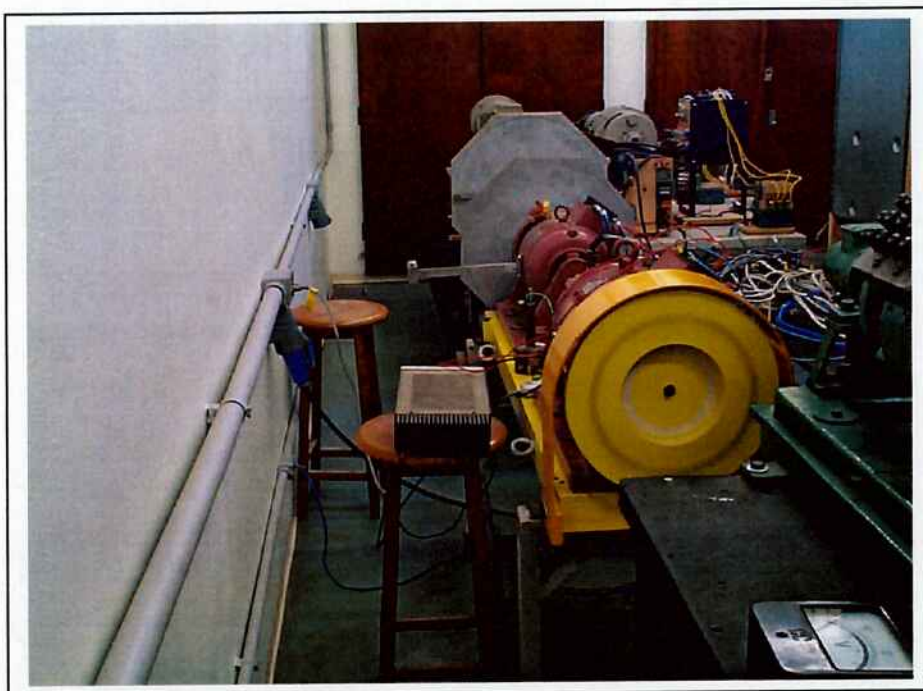


Figura22- Sistema Funcionando

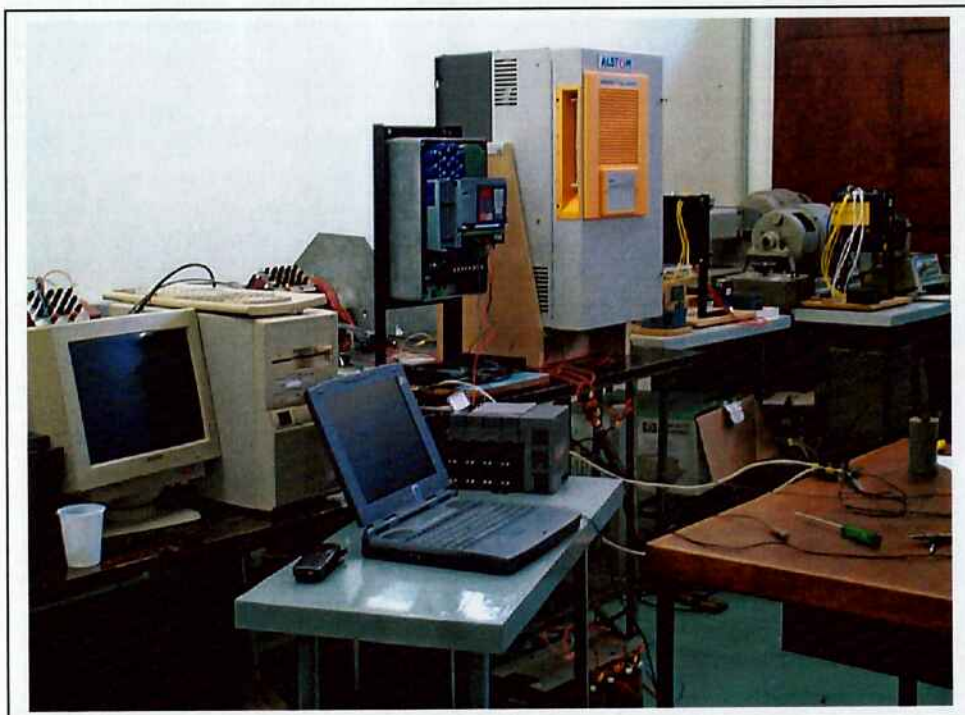


Figura 23- Montagem no Laboratório

Segue em anexo um exemplo da rotina “Ladder” implementada

