

ecai

Universidade Federal de Santa
Catarina
Centro Tecnológico
Curso de Engenharia de Controle e
Automação Industrial

ufsc

***Controle Adaptativo de um Cilindro
Hidráulico para Aplicação em
Robôs Manipuladores***

*Monografia submetida à Universidade Federal de Santa
Catarina
como requisito para a aprovação da disciplina:
EEL 5901: Projeto de Fim de Curso*

Gilberto da Cunha

Florianópolis, Agosto de 1996

Controle Adaptativo de um Cilindro Hidráulico para Aplicação em Robôs Manipuladores

Gilberto da Cunha

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Prof. Raul Guenther, Dr.
Orientador Empresa

Prof. Edson Roberto de Pieri, Dr.
Orientador do Curso

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. Alexandre Trofino Neto, Dr., Avaliador

Fábio Alonso da Silva, Debatedor

Eduardo José Pereira, Debatedor

"A maioria das idéias fundamentais da ciência são essencialmente sensíveis e, regra geral, podem ser expressas em linguagem compreensível a todos."
Albert Einstein

Resumo

Este trabalho trata da aplicação do método de Controle Adaptativo Auto-Ajustável a um servosistema hidráulico de posicionamento, e o estudo da Placa Controladora de Ponto Flutuante-DS1102 para implementações práticas.

Abstract

This work presents the application of Self-Tuning Controller for hydraulic positioning systems, and a study of the Floating-Point Controller Board-DS1102 in practical implementations.

Sumário

1. CAPÍTULO I. COLOCAÇÃO DO PROBLEMA	7
1.1 PRINCÍPIO DE FUNCIONAMENTO	7
1.2 MODELAGEM DO CONJUNTO CILINDRO+SERVOVÁLVULA	7
2. CAPÍTULO II. SISTEMAS DE CONTROLE ADAPTATIVO.....	10
2.1 CONTROLADOR AUTO-AJUSTÁVEL(SELF-TUNING CONTROLLER STC)	11
2.2 CONTROLADOR ADAPTATIVO POR MODELO DE REFERÊNCIA(MRAC).....	11
2.3 IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS DESCRITOS POR EQUAÇÕES DE DIFERENÇAS	12
2.4 MÉTODO DOS MÍNIMOS QUADRADOS RECURSIVO (MQR)	13
2.5 ESTIMAÇÃO DE PROCESSOS VARIANTES NO TEMPO	14
3. CAPÍTULO III. CONTROLE ADAPTATIVO APLICADO AO POSICIONADOR HIDRÁULICO.....	16
3.1 SÍNTESE DO CONTROLADOR	16
3.1.1 <i>Simulação de Estimação dos Parâmetros</i>	17
3.1.2 <i>Controladores STC</i>	21
4. CAPITULO IV. COMO IMPLEMENTAR NA PRÁTICA.....	26
4.1 PREELIMINARES	26
4.2 SINAIS DE MEDIÇÃO E ATUAÇÃO.....	26
4.3 IMPLEMENTAÇÃO PRÁTICA	27
5. CAPÍTULO FINAL. CONCLUSÕES E PERSPECTIVAS	31
6. REFERÊNCIAS BIBLIOGRÁFICAS:.....	32
A. ANEXO I. ESTUDO DO HARDWARE E SOFTWARE DA DS1102.....	34
A.1 PLACA CONTROLADORA DE PONTO FLUTUANTE (DS1102)	34
A.1.1 <i>Arquitetura</i>	34
A.1.2 <i>A Pinagem do Conector Externo</i>	35
A.2 O COCKPIT	36
A.2.1 <i>Os Arquivos de Entrada e Saída do Cockpit</i>	37
A.2.2 <i>O Modo de Edição do Cockpit</i>	38
A.2.3 <i>Os Instrumentos</i>	38
A.3 O TRACE	41
A.3.1 <i>Arquivos de Entrada e Saída do Trace</i>	41
A.3.2 <i>Como Operar o Trace</i>	43
A.4 INTERFACE TEMPO REAL PARA O SIMULINK.....	47
A.4.1 <i>O Simulink</i>	47
A.4.2 <i>Exemplo de Familiarização</i>	48
B. ANEXO II. ALGORITMO DO MATLAB USADO NAS SIMULAÇÕES	52

Capítulo Inicial: Introdução

Uma forma de obter posicionamentos precisos em robôs manipuladores é através de técnicas de controle adaptativo.

Quando posicionamentos precisos são desejados, os resultados obtidos com estas técnicas ultrapassam os do controle clássico não só pelos erros menores, mas também por garantirem a estabilidade em malha fechada. Como consequência, usando controle adaptativo pode-se garantir melhores condições de segurança no funcionamento.

Os manipuladores acionados eletricamente constituem apenas uma parcela dos robôs industriais. Quando é necessária a movimentação de cargas elevadas os acionamentos hidráulicos passam a ser convenientes.

A utilização dos acionamentos hidráulicos depende, no entanto, do desenvolvimento de técnicas de controle que permitam obter boa performance no posicionamento com esses atuadores [Heintze 93].

Técnicas de controle clássico podem ser normalmente utilizadas, entretanto o desempenho do sistema é, em geral, prejudicado pela inexatidão do modelo e outros fatores. A obtenção de controladores clássicos foi feita por diversos autores [Heintze 93], onde são discutidas as principais dificuldades de aplicação.

O estudo e a aplicação de técnicas de controle adaptativo em robôs manipuladores com acionamentos hidráulicos pode criar caminhos para a ampliação da utilização dos robôs industriais, possibilitando seu emprego na movimentação precisa de grandes cargas.

Este trabalho trata da aplicação de técnicas de controle adaptativo vistas na graduação a um servossistema hidráulico para controle de posição. O modelo utilizado para as simulações é linear, e os resultados dependem de quanto este modelo corresponde a realidade. E também a dinâmica dos atuadores foi desprezada [Lewis93].

No capítulo I tem-se a colocação da problema e no capítulo III a aplicação do controle adaptativo ao posicionador, que dentre os dois principais métodos para construção de controladores adaptativos, (Método Auto-Ajustável (STC) e Controle Adaptativo por Modelo de Referência (MRAC)) utilizou-se o Método Auto-Ajustável embora tenha-se também feito estudos sobre o Método de Controle Adaptativo por Modelo de Referência [Slotine 91] que não serão apresentados aqui.

O capítulo IV mostra como utilizar a placa de aquisição de dados DS1102 para a implementação prática das técnicas de controle adaptativo discutidas.

A parte do estudo da placa de aquisição de dados DS1102 da DSPACE (Digital Signal Processing and Control Engineering) bem como os seus recursos de visualização do processo em tempo real e como utilizá-la para implementações práticas se encontram no anexo I. Estas informações sobre a DS1102 e seu software são gerais e possibilitam a sua aplicação em qualquer área.

Para os interessados fornecemos no final uma bibliografia para um maior aprofundamento no assunto.

1. Capítulo I. Colocação do Problema

Este capítulo apresenta uma breve descrição do servosistema hidráulico. Mostraremos o seu princípio de funcionamento, e uma possível modelagem que pode ser encontrada na literatura [Stringer 76].

1.1 Princípio de Funcionamento

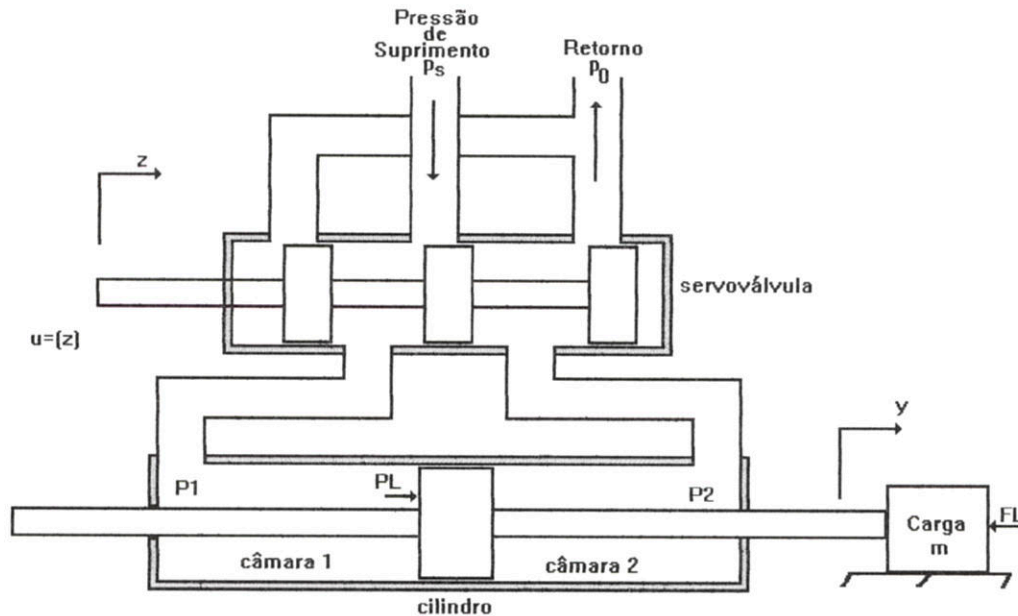


Figura 1. Servosistema de posicionamento hidráulico.

A figura 1 apresenta um conjunto cilindro+servoválvula, com a seguinte descrição de funcionamento:

Supondo o sistema em repouso em $z=0$, ou seja, as vias de suprimento e exaustão estão fechadas e a válvula encontra-se na posição central. Quando o carretel da válvula é deslocado para direita ($z>0$), como a pressão p_s é maior do que $P1$, produz uma vazão no sentido da pressão de suprimento para a câmara 1 do cilindro. Da mesma forma, como $P2$ é maior do que p_0 , há uma vazão no sentido da câmara 2 para o retorno. Provocando com isto o deslocamento do pistão para a direita ($y>0$).

1.2 Modelagem do Conjunto Cilindro+Servoválvula

Uma descrição deste sistema, pode ser encontrada na literatura [Stringer 76] na forma de equações de estado, é apresentada abaixo:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\mathbf{p})\mathbf{x} + \mathbf{b}(\mathbf{p})\mathbf{u} + \mathbf{e}(\mathbf{p})\mathbf{w} \\ \mathbf{y} &= \mathbf{c}(\mathbf{p})\mathbf{x} \end{aligned} \quad (1)$$

onde $x \in \mathbb{R}^3$, $u \in \mathbb{R}$ é a entrada, $y \in \mathbb{R}$ é a saída a ser regulada, $w \in \mathbb{R}$ é a perturbação e o modelo da planta é parametrizado pelos parâmetros p , $p \in P$.

Sendo:

x_1 é a posição do pistão (cm) (y na figura 1)

x_2 é a velocidade do pistão (cm/s) (\dot{y} na figura 1)

x_3 a pressão do óleo (PL na figura 1)

u é a vazão de entrada da servoválvula

y é a posição do pistão a ser regulada (y na figura 1)

w perturbação externa (FL na figura 1, $0 < ||FL|| < 800\text{daN}$)

p é a parametrização $p \in P$.

z posição do carretel.

Os dados para o problema são:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -(R + \frac{4}{\pi} R_c) / m & A / m \\ 0 & -\frac{4BA}{V} & \frac{-4B}{V} (\sigma + KL) \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 0 \\ \frac{4B}{V} \end{bmatrix}$$

$$e = \begin{bmatrix} 0 \\ -1/m \\ 0 \end{bmatrix}$$

$$c = [1 \ 0 \ 0]$$

onde os parâmetros B , m , r , R_c , KL , σ , V , A são:

B - módulo de compressibilidade (valor nominal de $B=14000\text{N/cm}^2$)

m - massa (valor nominal $m=0.1287\text{t}$)

A massa pode variar de: $0.1287 \leq m \leq 0.18\text{t}$

R - coeficiente de fricção ($R=0.150\text{daN/cm}$)

R_c - coeficiente de fricção de Coulomb ($R_c=0.01\text{daN/cm}$)

KL- coeficiente de fuga de óleo ($KL=0.002\text{cm}^5/\text{sdaN}$)

σ - constante hidráulica ($\sigma=0.24\text{cm}^5/\text{sdaN}$)

V- volume ($V=874\text{ cm}^3$)

A- área do pistão ($A=10.75\text{cm}^2$)

FL-força externa ($0 < FL < 800\text{ daN}$)

O problema consiste na aplicação de técnicas de controle adaptativo vistas na graduação a este sistema para o controle de posição e a discussão dos resultados obtidos.

2. Capítulo II. Sistemas de Controle Adaptativo

Nos últimos anos, o interesse em sistemas de controle adaptativo aumentou junto com o interesse e o progresso em robótica e outros campos de controle. O termo adaptativo tem uma variedade de significados específicos, mas normalmente implica que o sistema é capaz de se acomodar a mudanças imprevisíveis no sistema. Este conceito é de muito interesse para o projetista de sistema pois um sistema altamente adaptativo, também se acomodaria a erros de projeto em engenharia e compensaria pôr falhas ou incertezas moderadas de componentes secundários do sistema, aumentando portanto a confiabilidade do sistema.

Em algumas tarefas de controle, tais como manipulações de robôs, os sistemas a serem controlados apresentam incertezas nos parâmetros no começo da operação de controle. Exceto se tais incertezas são gradualmente reduzidas on-line por um mecanismo de adaptação ou estimação, isto pode causar imprecisão ou instabilidade nos sistemas de controle. Em muitas outras tarefas tais como em sistemas de potência, os sistemas dinâmicos são bem conhecidos no começo, mas variações de parâmetros (imprevisíveis) podem acontecer. Sem um contínuo "reprojeto" do controlador, o controlador projetado inicialmente de maneira apropriada pode não conseguir controlar as variações na planta. Geralmente, o objetivo do controle adaptativo é manter consistente o desempenho de um sistema na presença de incertezas ou variações não conhecidas nos parâmetros da planta. Como incertezas nos parâmetros ou variações ocorrem em muitos problemas práticos, controle adaptativo é vantajoso em muitas aplicações industriais.

Definição: " Sistema de controle adaptativo é aquele que estima, de forma contínua e automática, as características dinâmicas (tais como a função de transferência ou a equação de estado do processo), compara-as com as características dinâmicas desejadas e usa a diferença para variar parâmetros ajustáveis do sistema (normalmente características do controlador)". [Ogata 90]

Um controlador adaptativo difere de um controlador clássico, pelo fato de que os parâmetros do controlador são variáveis, e existe um mecanismo para ajuste destes parâmetros on-line, baseado nos sinais do sistema. Existem dois principais métodos para construção de controladores adaptativos, que são:

- Método Auto-Ajustável(STC);
- Controle Adaptativo por Modelo de Referência (MRAC).

2.1 Controlador Auto-Ajustável(Self-Tuning Controller STC)

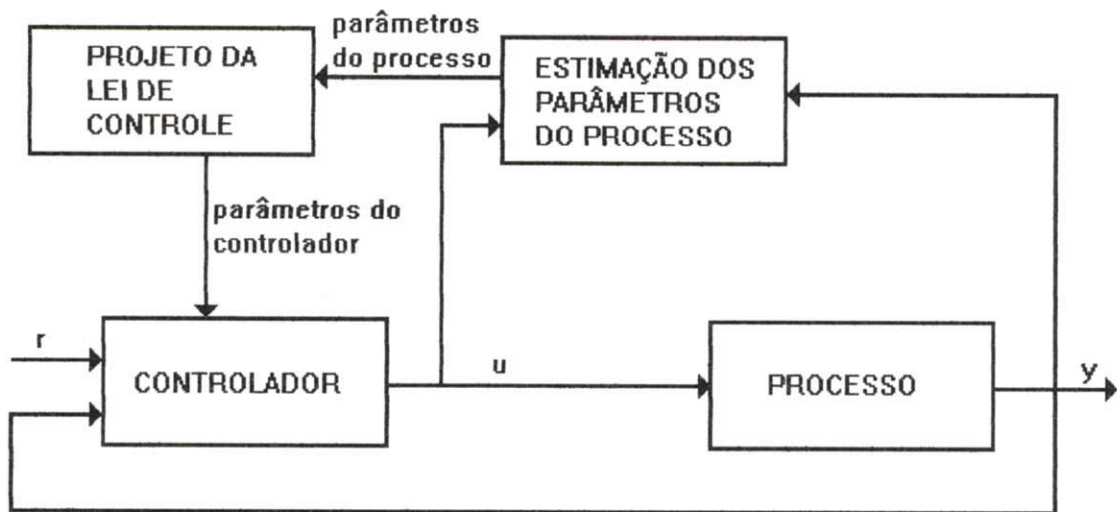


Figura 2. Algoritmo do controlador auto-ajustável.

A estrutura STC permite o estabelecimento de diferentes estratégias de controle adaptativo, simplesmente combinando-se convenientemente leis de controle e técnicas de identificação recursiva. Estes controladores baseiam-se no princípio da equivalência certa, isto é, calcula a lei de controle supondo os parâmetros do processo conhecidos para a seguir substituí-los pelos valores estimados.

2.2 Controlador Adaptativo por Modelo de Referência(MRAC)

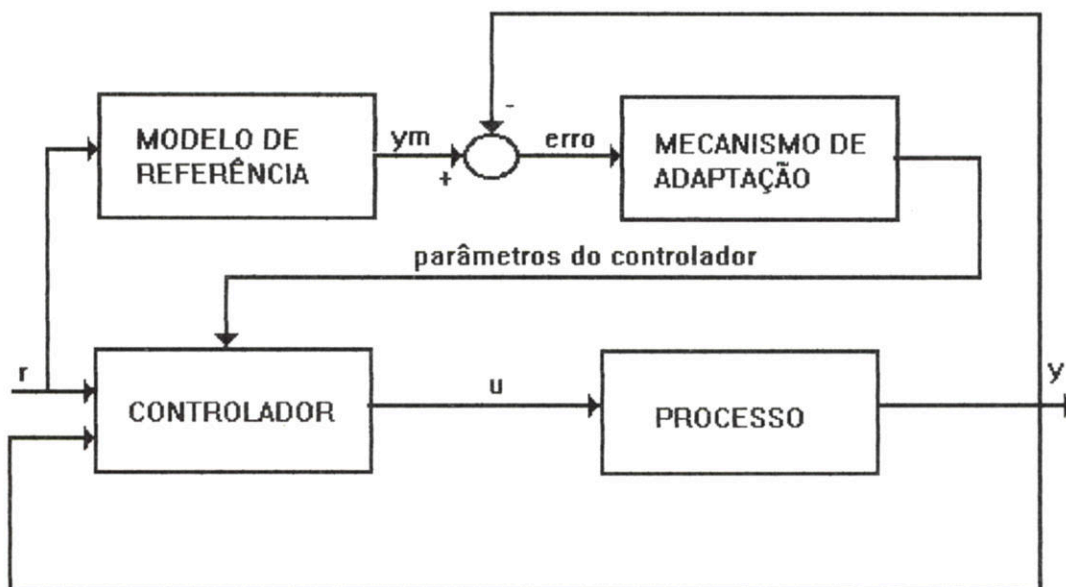


Figura 3. Controlador adaptativo por modelo de referência.

As especificações são dadas em termos de um modelo de referência, o qual informa como a saída do processo idealmente deveria responder a um

signal de comando (r -setpoint) e os parâmetros do controlador são ajustados de tal maneira que o erro ($ym-y$) seja mínimo.

Relações entre MRAC e STC

- Ambos os tipos de sistemas têm uma malha interna para controle e outra externa para estimação dos parâmetros.
- STC são mais flexíveis porque existe a possibilidade de se acoplar vários controladores com diferentes estimadores (isto é, a separação do controle e da estimação).
- A estabilidade e convergência dos controladores ST são geralmente muito difíceis de serem provadas, requerendo freqüentemente que o sinal no sistema seja suficientemente bom, de tal forma que os parâmetros estimados converjam para os parâmetros verdadeiros.
- Historicamente(origem):
MRAC - controle ótimo de servomecanismos determinísticos e geralmente tempo contínuo;
STC - Problemas estocásticos de regulação e geralmente tempo discreto.

2.3 Identificação de Sistemas Dinâmicos Descritos Por Equações de Diferenças

Entende-se por identificação a determinação de um modelo matemático do processo representando os seus aspectos essenciais de forma adequada para a sua utilização.

As técnicas de identificação por computador podem ser:

OFF-LINE: coleta-se uma certa quantidade de dados de entrada e saída que pode estar armazenando num computador. Em seguida os dados são processados para se estimar parâmetros do modelo.

ON-LINE: onde os dados de entrada e saída não precisam ser armazenados e um algoritmo recursivo é usado para ajustar os parâmetros do controlador após cada período de amostragem.

A identificação de um modo geral consiste de três etapas:

1. Determinação da estrutura;
2. Estimação dos parâmetros
3. Validação do modelo.

A fase que antecede a estimação dos parâmetros é a determinação da estrutura, que em sistemas monovariáveis, consiste na determinação do número de parâmetros a serem estimados, ou seja, a ordem das equações de diferenças que descrevem a dinâmica do sistema. Este aspecto é importante do ponto de vista de controle. A estrutura deve ser simples, de modo a ser entendida e manipulada sem esforço computacional excessivo.

A estimação de parâmetros é um procedimento numérico que determina os valores desconhecidos dos parâmetros do modelo. Obtém-se a estimativa dos parâmetros através do tratamento estatístico dos dados segundo um critério preestabelecido. Uma maneira possível de estimação dos parâmetros on-line é a técnica dos Mínimos Quadrados Recursiva.

Uma vez encontrados os valores dos parâmetros objetivando qualificar o desempenho do modelo estimado, métodos de validação de modelo são aplicados.

Estes métodos baseiam-se na comparação entre a resposta do sistema e a resposta obtida através do modelo. O modelo é tido como adequado se o erro cometido no ajuste estiver dentro de valores preestabelecidos, ou ainda, para uma determinada aplicação, a resposta do modelo deve refletir corretamente a resposta do sistema, não apresentando discrepância entre o real e o estimado.

2.4 Método dos Mínimos Quadrados Recursivo (MQR)

No STC o esquema do estimador dos parâmetros deve ser iterativo, onde o modelo do sistema é atualizado a cada período de amostragem, quando novos dados estão disponíveis. Para a estimação on-line de parâmetros uma técnica muito usada é a dos Mínimos quadrados Recursivo.

O algoritmo, dado uma matriz de variância $\mathbf{P}(\mathbf{0})$ inicial e um vetor de parâmetros $\hat{\theta}(\mathbf{0})$ inicial, consiste em:

- Medir a entrada e a saída do processo;

- Atualizar o vetor de medidas;

$$\varphi^T(\mathbf{t} + 1) = [-\mathbf{y}(\mathbf{t}) \quad -\mathbf{y}(\mathbf{t} - 1) \quad \dots \quad \mathbf{u}(\mathbf{t} - \mathbf{d} + 1) \quad \dots]$$

\mathbf{d} - é o atraso de transporte.

- Calcular o erro de previsão;

$$\varepsilon(\mathbf{t} + 1) = \mathbf{y}(\mathbf{t} + 1) - \varphi^T(\mathbf{t} + 1)\hat{\theta}(\mathbf{t})$$

- Calcular o ganho de estimação;

$$\mathbf{K}(\mathbf{t} + 1) = \frac{\mathbf{P}(\mathbf{t})\varphi(\mathbf{t} + 1)}{1 + \varphi^T(\mathbf{t} + 1)\mathbf{P}(\mathbf{t})\varphi(\mathbf{t} + 1)}$$

- Calcular o vetor de parâmetros estimados;

$$\hat{\theta}(\mathbf{t} + 1) = \hat{\theta}(\mathbf{t}) + \mathbf{K}(\mathbf{t})\varepsilon(\mathbf{t} + 1)$$

- Calcular a matriz de Covariância;

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \frac{\mathbf{P}(t)\boldsymbol{\varphi}(t+1)\boldsymbol{\varphi}^T(t+1)\mathbf{P}(t)}{1 + \boldsymbol{\varphi}^T(t+1)\mathbf{P}(t)\boldsymbol{\varphi}(t+1)}$$

ou então pela equação:

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{K}(t+1)[\mathbf{P}(t)\boldsymbol{\varphi}(t+1)]^T$$

- Retornar.

O erro de previsão $\varepsilon(t+1)$ no algoritmo se torna zero quando o parâmetros estimados ($\hat{\theta}$) convergem para o valor dos parâmetros do processo real (θ). A matriz de Covariância do estimador é a medida da qualidade do estimador. O tamanho dos elementos da diagonal estão relacionados com a variância dos elementos em $\hat{\theta}$.

Por exemplo, se o elemento $\mathbf{P}(1,1)$ é pequeno significa que a estimativa do primeiro elemento do vetor de parâmetros $\hat{\theta}(1,1)$ é adequada. Logo, a matriz de Covariância inicial deve ser alta (10^3 ou maior) se nenhum conhecimento do sistema está disponível.

Deve-se ressaltar aqui que um fator muito importante dos mínimos quadrados recursivo: este algoritmo deve ser persistentemente excitado, se não ele perde sua habilidade de rastrear os parâmetros. Um ruído de baixa amplitude muitas vezes é necessário para a estimação dos parâmetros. [Slotine 91]

2.5 Estimação de Processos Variantes no Tempo

Quando o processo é variante no tempo devemos utilizar técnicas que impeçam que o algoritmo dos MQR "adormeça", ou seja perca a capacidade de rastreamento.

Uma técnica muito utilizada para evitar este problema é a utilização de um Fator de Esquecimento no algoritmo dos MQR evitando que os elementos de $\mathbf{P}(t)$ tendam a zero e mantém o algoritmo sempre "alerta" para rastrear dinâmicas variantes. Abaixo mostramos as modificações no algoritmo:

$$\mathbf{K}(t+1) = \frac{\mathbf{P}(t)\boldsymbol{\varphi}(t+1)}{\lambda + \boldsymbol{\varphi}^T(t+1)\mathbf{P}(t)\boldsymbol{\varphi}(t+1)}$$

$$\mathbf{P}(t+1) = \frac{1}{\lambda} \left\{ \mathbf{P}(t) - \frac{\mathbf{P}(t)\boldsymbol{\varphi}(t+1)\boldsymbol{\varphi}^T(t+1)\mathbf{P}(t)}{1 + \boldsymbol{\varphi}^T(t+1)\mathbf{P}(t)\boldsymbol{\varphi}(t+1)} \right\}$$

Observamos que com $\lambda = 1$ temos o MQR anterior.

Na prática tem-se: $0.95 \leq \lambda \leq 1$, ou seja, ponderação maior para medidas mais recentes.

Outra técnica consiste em reinicializar a matriz de Covariância com a soma de uma matriz \mathbf{Q} cujos elementos dão um ganho abrupto no estimador assim que o seu traço seja menor que um determinado valor ($\text{tr} < \text{tr}_0$). O erro de estimação ε também pode ser observado para esta reinicialização. Esta técnica é chamada de "Covariance Resetting".

Se alguns parâmetros variam, podemos somar à matriz de Covariância um ganho somente nas linhas correspondentes a estes parâmetros, evitando que os outros oscilem quando aumentamos todos os valores do matriz \mathbf{P} . Esta técnica é chamada de "Random Walk".

3. Capítulo III. Controle Adaptativo Aplicado ao Posicionador Hidráulico

Neste capítulo são aplicadas as técnicas de Controle Auto-Ajustável (STC) vistas na graduação. Para tanto é necessário obter um modelo discreto do sistema.

Em seguida, são apresentados os resultados de simulação de estimação dos parâmetros e a síntese de controladores adaptativos mais comuns na literatura de STC.

3.1 Síntese do Controlador

Considerando $FL=0$ (sem perturbação externa) na equação (1) da seção 1.2 e substituindo os valores com $m=0.1287$, obtém-se a seguinte função de transferência:

$$G(s) = \frac{y(s)}{u(s)} = \frac{5351.9}{s^3 + 17s + 57552s}$$

Com um período de amostragem de $T=0.01s$, escolhido para não piorar a margem de fase do sistema mais de 5° , tem-se que:

$$B_0G(z) = \frac{z-1}{z} z \left\{ \frac{G(s)}{s} \right\} \Rightarrow$$

$$H(z) = B_0G(z) = \frac{6.45 \times 10^{-4} z^2 + 1.26 \times 10^{-3} z + 6.31 \times 10^{-4}}{z^3 + 0.36z^2 - 0.52z - 0.84}$$

Que será o modelo usado nas simulações de controle.

Por métodos clássicos, especificamente Métodos Freqüenciais, pode-se chegar a um controlador do tipo:

$$C(s) = \frac{70s + 420}{s + 5.35}$$

Usando a aproximação de Tustin com $T=0.01s$ obtém-se:

$$C(z) = \frac{70z - 65.9104}{z - 0.9479}$$

Em malha fechada obtém-se a seguinte resposta no tempo:

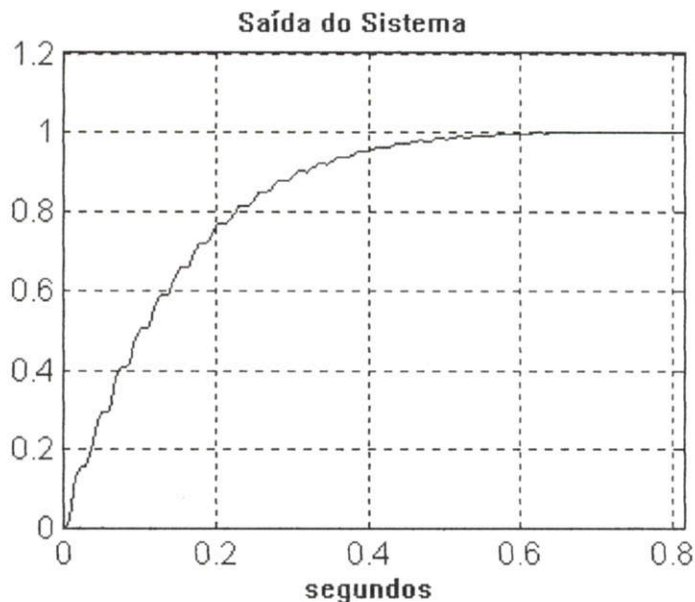


Figura 4. Resposta no tempo do servossistema hidráulico, com um degrau de referência aplicado em $t=0$, com o controlador proposto.

3.1.1 Simulação de Estimação dos Parâmetros

Na estimação dos parâmetros com o algoritmo dos MQR do modelo linear discreto do servossistema hidráulico com:

$$\hat{\theta}(0) = [0 \ 0 \ 0 \ 0 \ 0 \ 0],$$

$$P(0) = \begin{bmatrix} 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^6 \end{bmatrix}$$

e uma entrada com um ruído sobreposto como mostra a figura 5, obtém-se os resultados da figura 6.

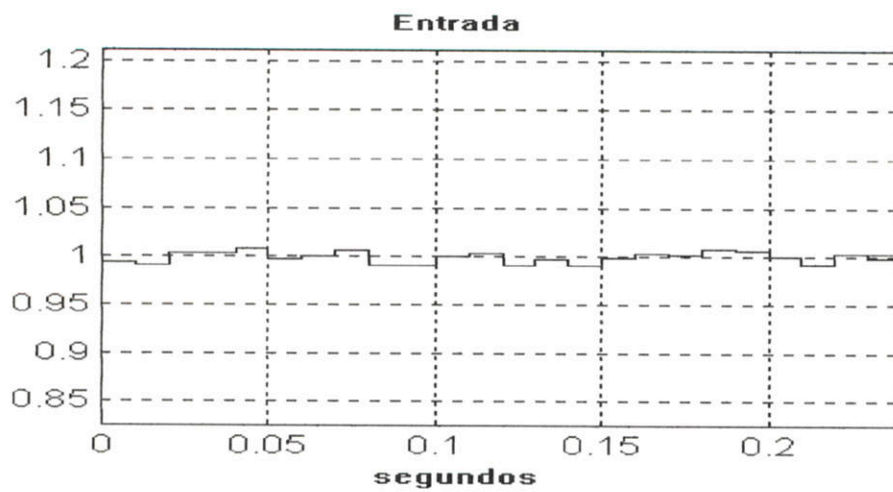
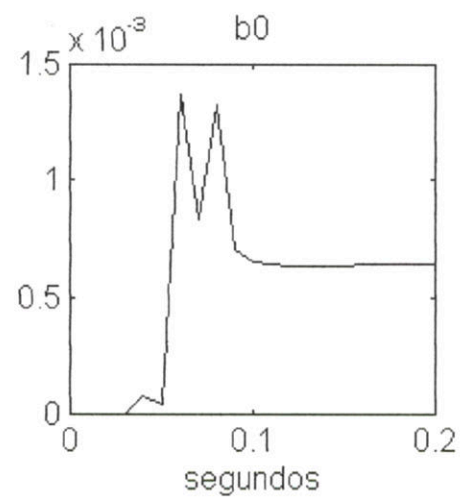
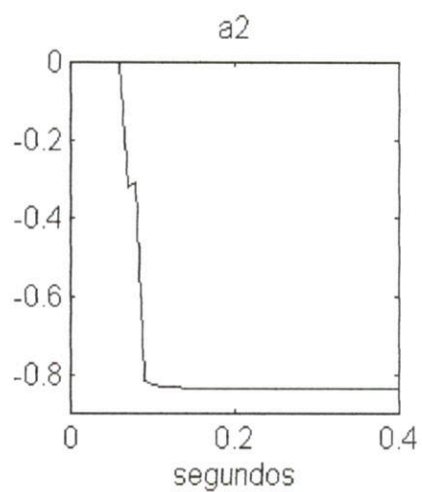
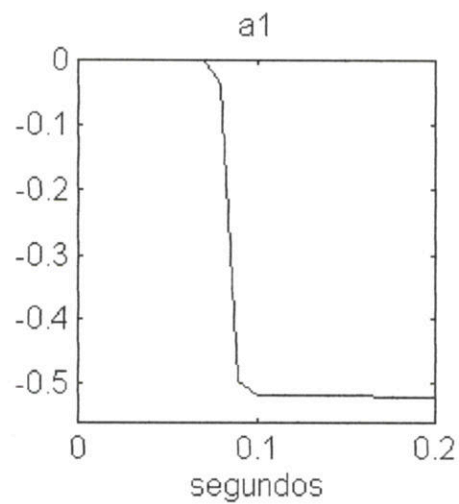
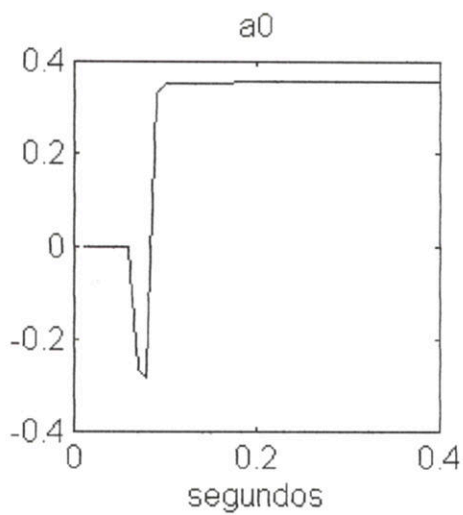


Figura 5. Entrada do sistema para a estimação dos parâmetros



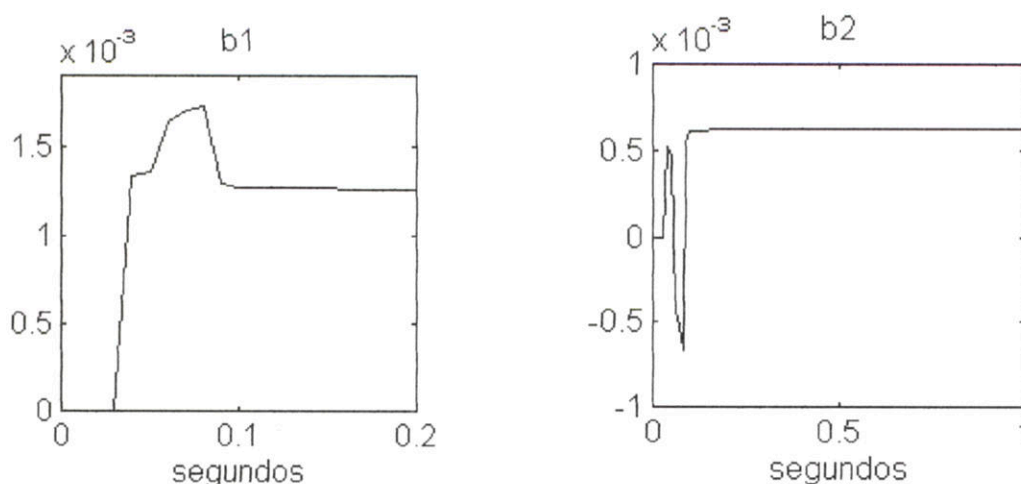


Figura 6. Parâmetros estimados do servosistema hidráulico

Onde:

$$H(z) = \frac{b_0z^3 + b_1z^2 + b_2z}{z^3 + a_0z^2 + a_1z + a_2}$$

Após 100 (1s) iterações do algoritmo obtém-se que:

$$\hat{\theta} = [0.36 \quad -0.5192 \quad -0.84 \quad 0.0006 \quad 0.0013 \quad 0.0006]$$

Na prática é necessário incluir no algoritmo de estimação uma análise do erro de estimação ε e sua variância para a validação do modelo estimado. A variância é dada por:

$$v(k) = \frac{1}{k} [(k-1)v(k-1) + (\varepsilon(k) - \bar{\varepsilon})^2]$$

Onde $\bar{\varepsilon}$ é o erro médio até o instante k .

Considera-se a seguir somente a variação do parâmetro m , embora possam ocorrer variações de outros parâmetros. Como, por exemplo, o módulo de compressibilidade (B) com a variação de temperatura do fluido.

Se algum parâmetro, como por exemplo a massa (m) do sistema, variar de $m=0.1287$ para $m=0.18$, obtém-se a seguinte função de transferência discreta :

$$H(z) = \frac{5 \times 10^{-4} z^2 + 1.5 \times 10^{-3} z + 5 \times 10^{-4}}{z^3 - 0.1885z^2 + 0.0372z - 0.84}$$

Usando "Covariance Resetting", com uma matriz de atualização de P dada por:

$$Q = \begin{bmatrix} 10^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^3 \end{bmatrix}$$

se o traço de $[P]$ for menor que 10^3 , obtêm-se resultados como mostra a figura 7 e $\hat{\theta} = [-0.188 \quad 0.0372 \quad -0.84 \quad 0.0005 \quad 0.0015 \quad 0.0005]$.

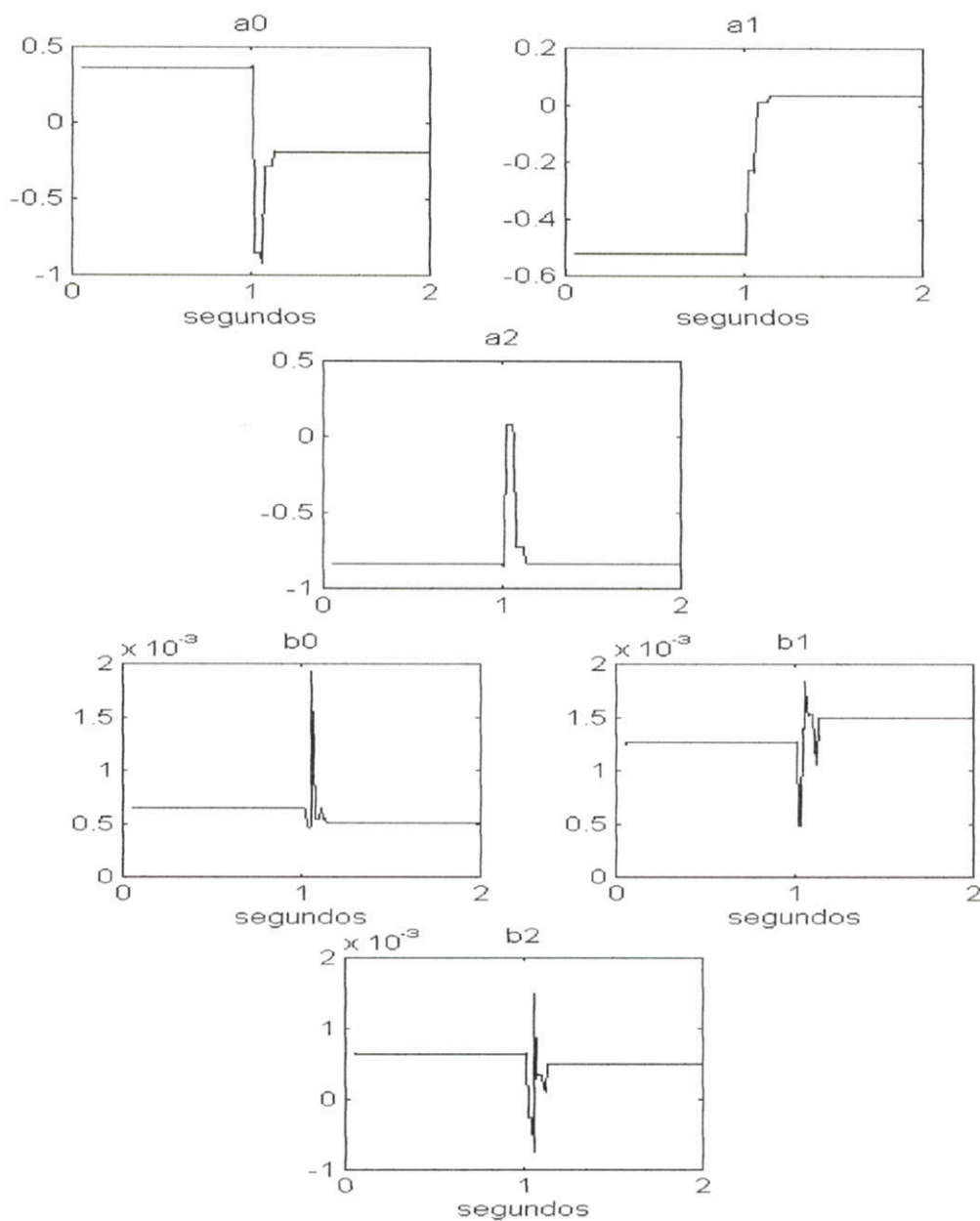


Figura 7. Estimação de parâmetros variáveis no sistema com "Covariance Resetting".

3.1.2 Controladores STC

Apresenta-se a seguir algumas técnicas de STC e os resultados das simulações obtidas para o servossistema hidráulico.

Na técnica por alocação de pólos faz-se com que os pólos de malha fechada do sistema sejam iguais a um polinômio $T(z^{-1})$, escolhido de acordo com as especificações desejadas do sistema.

Assim se a função de transferência do processo de fase mínima for:

$$G(z^{-1}) = \frac{z^{-k}B(z^{-1})}{A(z^{-1})}$$

e o controlador por alocação de pólos for do tipo

$$C(z^{-1}) = \frac{G(z^{-1})}{H(z^{-1})}$$

em malha fechada, condiciona-se que $A(z^{-1})H(z^{-1}) + z^{-k}B(z^{-1})G(z^{-1}) = T(z^{-1})$.

Para o servossistema hidráulico tem-se que:

$$\text{Com } G(z) = \frac{b_0z^3 + b_1z^2 + b_2z}{z^3 + a_0z^2 + a_1z + a_2} \text{ e com } T(z^{-1}) = 1 + d_0z^{-1} + d_1z^{-2} + d_2z^{-3}$$

obtem-se

$$\text{que } C(z) = \frac{g_0z^2 + g_1z + g_2}{b_0z^2 + b_1z + b_2}$$

Onde:

$$g_0 = d_0 - a_0$$

$$g_1 = d_1 - a_1$$

$$g_2 = d_2 - a_2$$

Na figura 8 e 9 encontramos a resposta do sistema para duas alocações de pólo para o sistema com $m=0.1287$ (em 0.1, 0.2, 0.3 e em 0.4, 0.5, 0.6 respectivamente). A referência é do tipo: 1 p/ $0 \leq t \leq 5$ e 0 p/ $5 \leq t \leq 10$.

O algoritmo foi inicializado com $P(0) = 10^6 \cdot I_{6 \times 6}$ e

$$\hat{\theta}(0) = [0.3 \quad -0.5 \quad -0.8 \quad 0.0006 \quad 0.0013 \quad 0.0006]$$

Observe que na ação de controle ocorre o fenômeno conhecido na literatura de controle como "Ringing", pelo fato dos pólos do controlador estarem nas vizinhanças de $z=-1$ ($-0.97 \pm 0.15i$ para $m=0.1287$). Para $m=0.18$ o sistema deixa de ser de fase mínima (zeros de malha aberta em -2.6180 e -0.3820) e

deve-se usar outro algoritmo para a lei de controle ou considerar no cálculo do controlador por alocação de pólos $\mathbf{B}(z^{-1}) = \mathbf{B}^d(z^{-1})\mathbf{B}^f(z^{-1})$. Separando os zeros estáveis de malha aberta (\mathbf{B}^d) dos instáveis (\mathbf{B}^f).

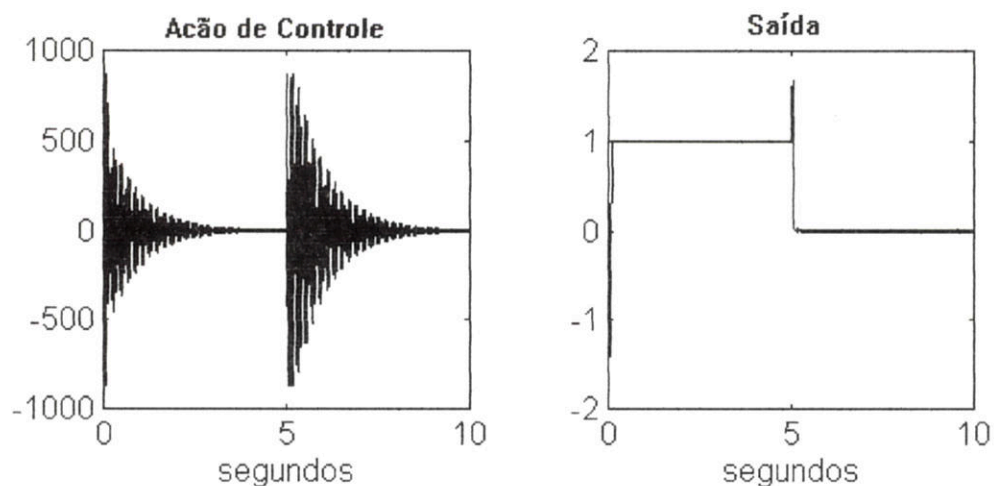


Figura 8. Controlador por alocação de pólos em **0.1, 0.2 ,0.3.**

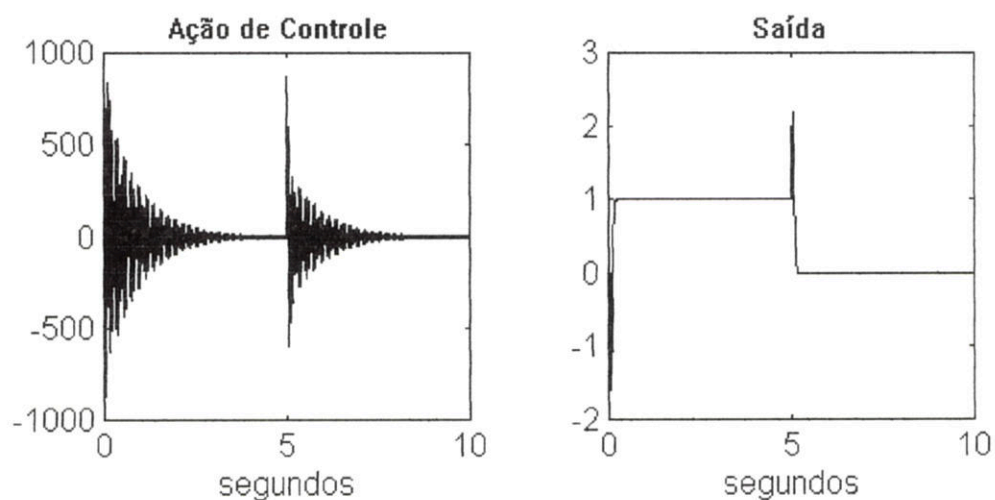


Figura 9. Controlador por alocação de pólos em **0.4, 0.5 ,0.6.**

Tendo em vista os problemas encontrados com estes controladores propomos um controlador do tipo:

$$C(z) = \frac{c_0z + c_1}{d_1z + d_0}$$

Com seus parâmetros dados por:

$$\begin{bmatrix} a0 & 1 & b0 & 0 \\ a1 & a0 & b1 & b0 \\ a2 & a1 & b2 & b1 \\ 0 & a2 & 0 & b2 \end{bmatrix} \begin{bmatrix} d1 \\ d0 \\ c0 \\ c1 \end{bmatrix} = \begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \end{bmatrix}$$

Sendo $e1$, $e2$, $e3$ e $e4$ os coeficientes do polinômio de malha fechada $z^4 + e1z^3 + e2z^2 + e3z + e4$.

Tomamos $e1=-0.5427$, $e2=-0.8155$, $e3=-0.3859$ e $e4=0.7546$ de forma que com $\hat{\theta} = [0.36 \quad -0.5192 \quad -0.84 \quad 0.0006 \quad 0.0013 \quad 0.0006]$ o controlador coincide com $C(z) = \frac{70z - 65.9104}{z - 0.9479}$, que é o controlador obtido anteriormente por métodos clássicos.

Na figura 10 tem-se o resultado da simulação. O algoritmo foi inicializado com $\hat{\theta}(0) = [0.3 \quad -0.5 \quad -0.8 \quad 0.0006 \quad 0.0013 \quad 0.0006]$, $P(0) = 10^6 \cdot I_{6 \times 6}$, supondo um degrau de referência aplicado em $t=3$ s.

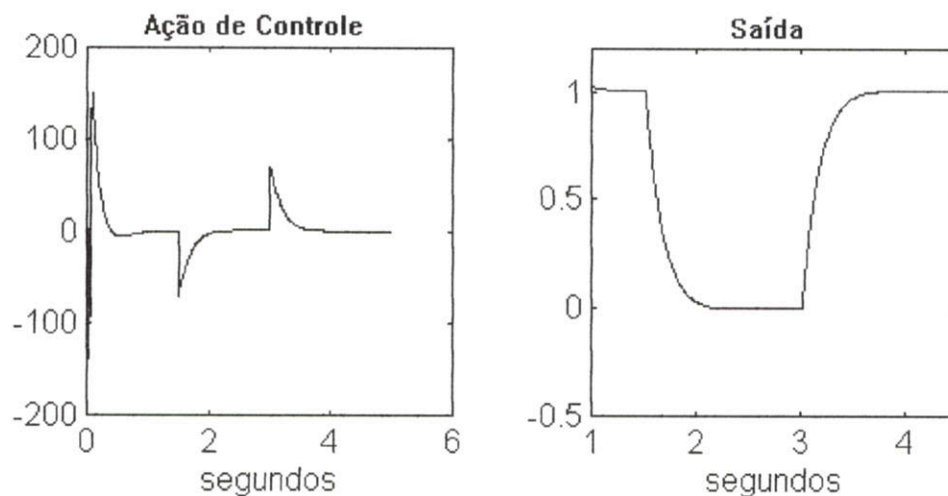


Figura 10. Resposta no tempo, para um degrau de referência, do sistema com o controlador proposto para $m=0.1287$.

Deseja-se neste ponto que o posicionador hidráulico com cargas de massa variando de $m=0.16$ a $m=.18$ siga a seguinte trajetória:

$$y_{ref} = A_m \sin(2\pi f)$$

com

$$A_m = 10 \text{ e } f = 0.003. \text{ (ver figura 11)}$$

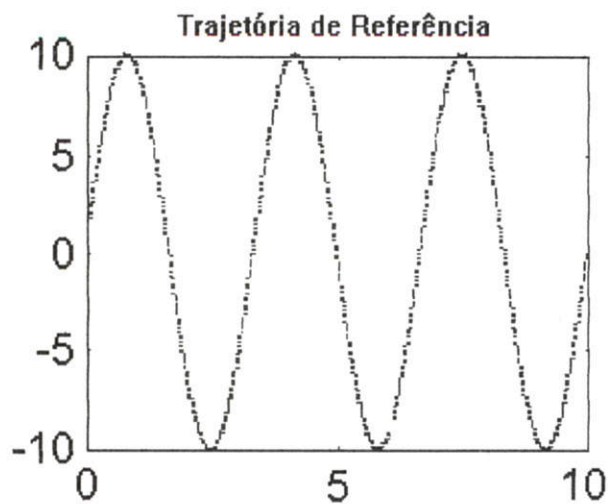


Figura 11. Trajetória de referência desejada para massas variando de $m=0.16$ a $m=0.18$.

Com o controlador adaptativo obteve-se as curvas de erros de seguimento de referência das figuras 12 e 13, para $m=0.16$ e $m=0.18$, respectivamente.

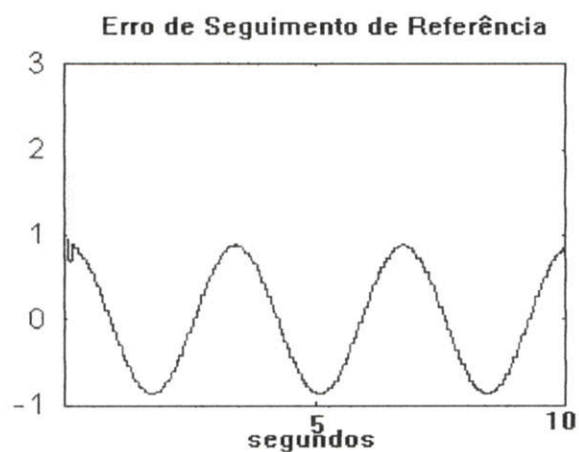


Figura 12. Erro de seguimento de trajetória com o Controlador Adaptativo com $m=0.16$.

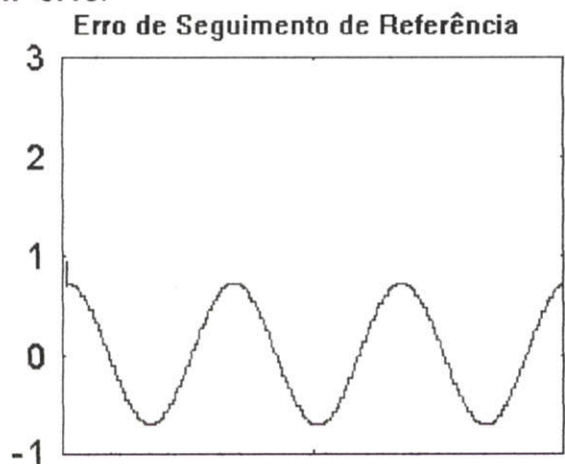


Figura 12. Erro de seguimento de trajetória com o Controlador Adaptativo para $m=0.18$.

Se o controlador não for adaptativo ($C(z) = \frac{70z - 65.9104}{z - 0.9479}$ constante para $0.16 \leq m \leq 0.18$), para $m=0.18$ observa-se um erro de seguimento de trajetória como mostra a figura 13.

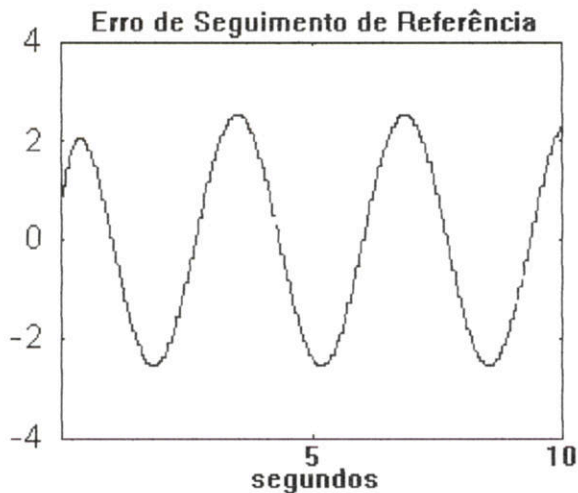


Figura 13. Erro de seguimento de trajetória com o controlador convencional.

Como pode-se perceber, com o controlador adaptativo obtêm-se erros de posicionamento menores com a variação da massa de $m=0.16$ a $m=0.18$. Deve-se para tanto, dentro da especificação proposta escolher um polinômio $z^4 + e_1z^3 + e_2z^2 + e_3z + e_4$ que atenda as exigências de erro com as variação paramétricas.

Escolhemos um controlador **PD** (Proporcional-Derivativo), por simplicidade de cálculo, mas podemos usar outros tipos de controladores. Como por exemplo controladores **PID** (Proporcional-Integral-Derivativo) ou de ordem maior, implicando porém em um aumento do esforço computacional necessário (maior número de equações para resolver).

A principal característica dos controladores **PD** é a de ser "antecipativo", ou seja, tende a "antecipar" o efeito da mudança de referência na saída do sistema e procura "corrigir" antes de que ocorra". Isto leva que o controlador exija do sistema uma "desaceleração da resposta" tendendo a diminuir o pico.

Dentre outras técnicas de STC tem-se o Controlador de Variância Mínima [Clarke 75] usado para referência constante, quando este varia lentamente ou é constante por longos períodos que não é o caso de manipuladores.

4. Capítulo IV. Como Implementar na Prática

4.1 Preliminares

Discutir-se-á aqui como deve ser a implementação prática dos algoritmos de controle adaptativo desenvolvidos, já que sua execução não foi possível devido ao atraso na entrega do "toolbox" RTW.

Na figura 14 temos um esquema simplificado da montagem para implementação.

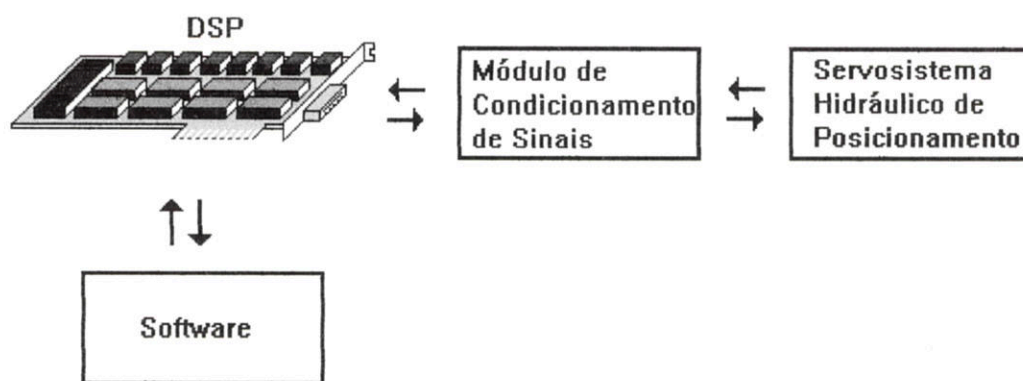


Figura 14. Esquema simplificado de execução prática.

Dentre estes, até o presente momento, o Módulo de Condicionamento de Sinais, a DSP, o Servosistema Hidráulico e parte do Software já se encontram disponíveis e montados no LASHIP (Laboratório de Sistemas Hidráulicos e Pneumáticos).

4.2 Sinais de Medição e Atuação

No Módulo de Condicionamento de Sinais estão disponíveis os seguintes sinais de medição e atuação condicionados de 0 a 10V:

Sinais de Medição

- Pressão em ambas as câmaras do cilindro (**P1** e **P2** na figura 1);
- Posição do pistão (**y** na figura 1);
- Temperatura do fluido.

Sinais de Atuação

- Corrente que regula a vazão de entrada da servoválvula (**u** na figura 1)

4.3 Implementação Prática

Nesta implementação prática, dos algoritmos de controle adaptativo usando a DSP, devemos primeiramente construir cada algoritmo em blocos para serem usados pelo Simulink, para em seguida estes serem codificados, compilados e carregados na DSP.

Como por exemplo o bloco de estimação dos parâmetros dos mínimos quadrados recursivo como mostra a figura 15.

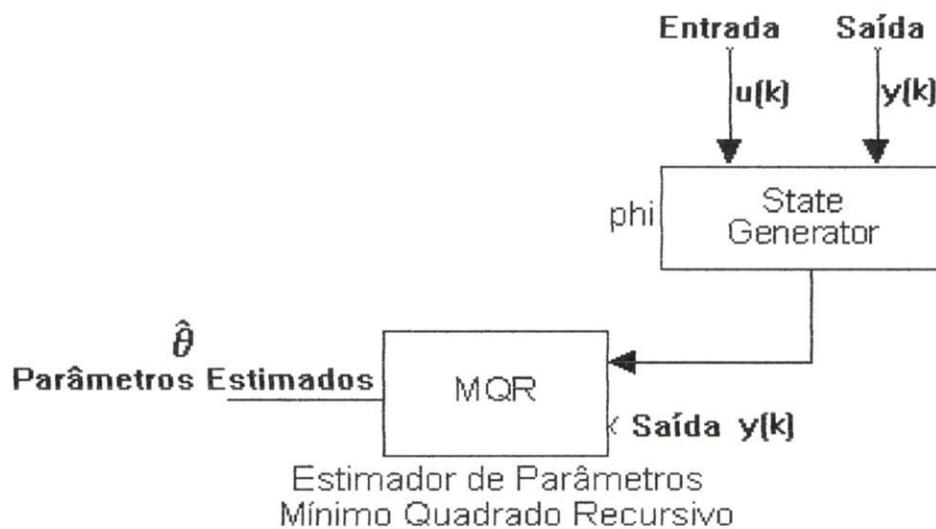


Figura 15. Implementação do algoritmo dos Mínimos Quadrados Recursivo no Simulink do Matlab.

No estimador dos MQR do Simulink podemos incluir outra técnica de estimação de parâmetros variantes (seção 2.5), já que as considerações feitas na seção 3.1.1 de que somente a massa (m) varia pode não ocorrer.

Além disso podem surgir outros problemas de ordem prática como por exemplo a consideração de que a amostragem é instantânea e que os valores das amostragens não contêm erros [Malik 91]. Para tanto devemos tomar as devidas precauções.

Depois de validado o algoritmo através de simulação, pode-se partir para a implementação prática, e construir no Simulink usando suas bibliotecas, o diagrama mostrado na figura 16.

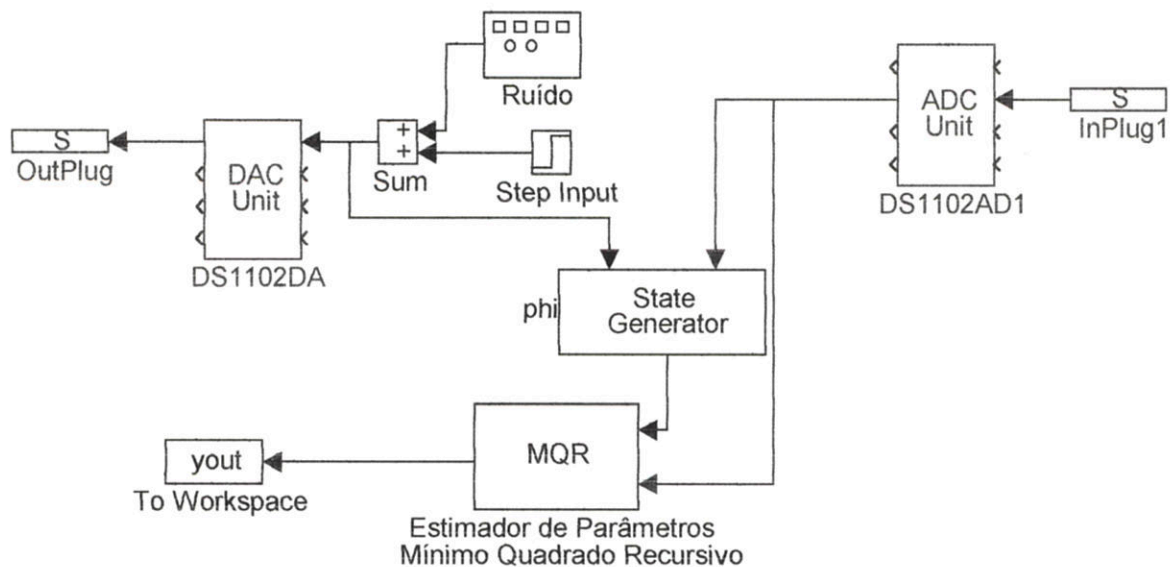


Figura 16. Diagrama do Simulink para estimação dos parâmetros do servossistema hidráulico.

Através do bloco **InPlug1** são coletadas as medidas da saída do sistema (posição do cilindro). E através de **OutPlug** aplica-se um degrau sobreposto de ruído ao sistema, obtendo-se então, no espaço de trabalho do Matlab a variável **yout**, que é a matriz de parâmetros estimados do sistema ($\hat{\theta}$). O diagrama após montado é carregado na placa para execução através da opção **Code** no Simulink como descrito em anexo.

Da mesma forma para o controlador adaptativo temos o diagrama mostrado na figura 17.

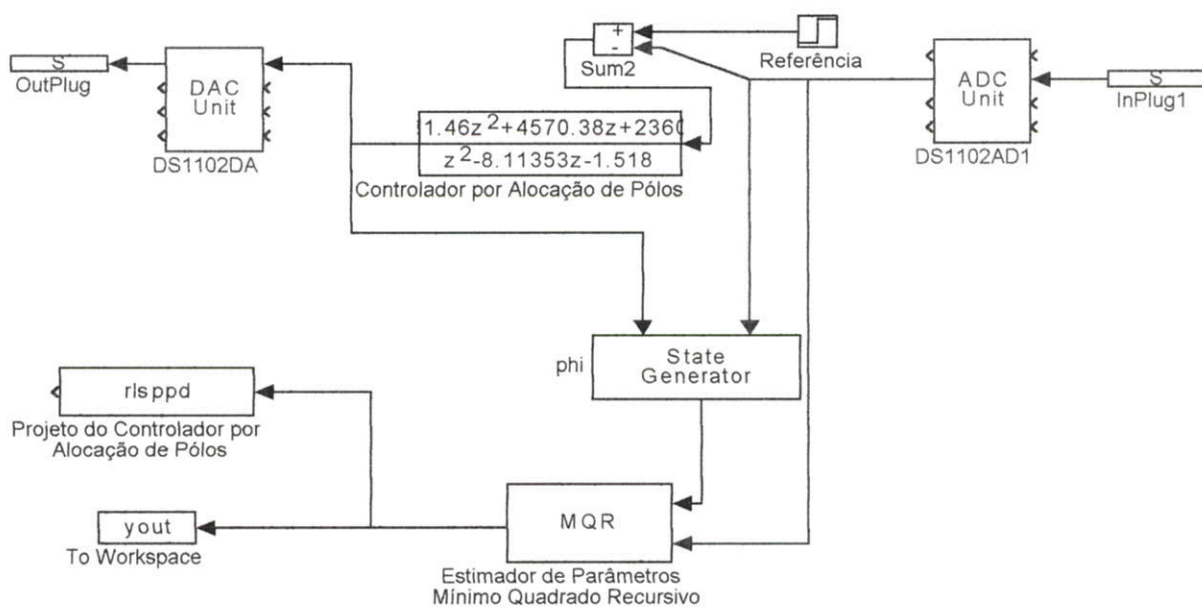


Figura 17. Implementação do controlador por Alocação de Pólos Adaptativo.

A maioria dos blocos usados aqui se encontram disponíveis nas bibliotecas do Simulink embora precisem de ajuste para a aplicação específica. A sintaxe de construção dos blocos não é complicada, e permite a criação de nossas próprias funções do Matlab.

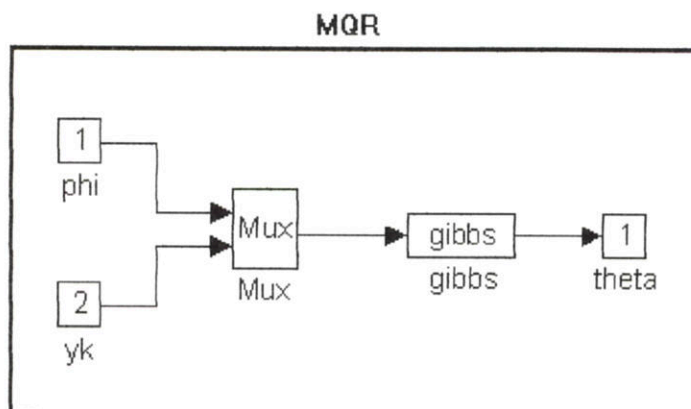


Figura 18. Bloco MQR do Simulink "desmascarado".

Na figura 18 temos o bloco MQR, utilizado nos diagramas anteriores, "desmascarado", ou seja representado na forma de componentes básicos do Simulink.

Após o carregamento na placa podemos acompanhar a evolução das variáveis escolhidas em tempo real no Trace e criar uma espécie de "software de supervisão" no Cockpit.

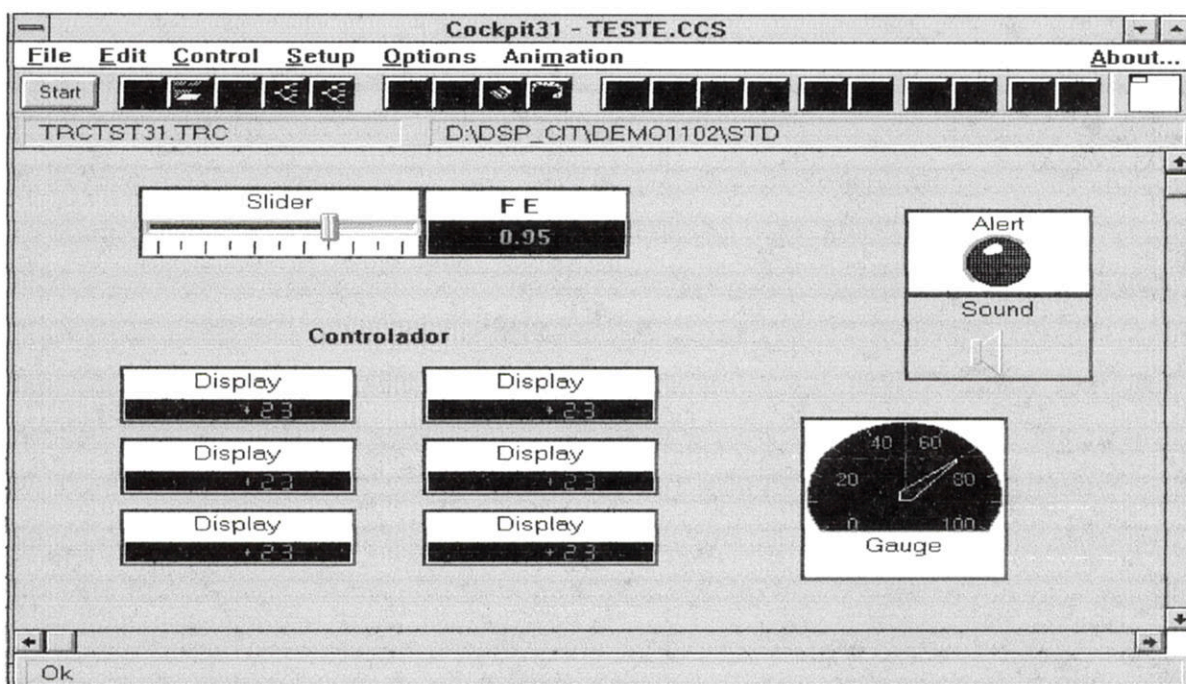


Figura 19. Possível "supervisor" para controle adaptativo implementado no Cockpit.

A figura 19 mostra um exemplo de diagrama construído no Cockpit, para "supervisionar" o controlador adaptativo. Nele podemos setar o Fator de Esquecimento (**0.95**), observar alguma variável importante através do medidor, acionar um alarme se uma variável ultrapassar uma condição e verificar os parâmetros do controlador.

5. Capítulo Final. Conclusões e Perspectivas

Embora o modelo linear apresentado aqui não corresponda aquele disponível na bancada, podemos utilizar a técnica de estimação do Mínimos Quadrados Recursivo (MQR) para o levantamento do mesmo. É importante destacar também que, além de contribuir para a diversificação dos métodos de acionamento para robôs manipuladores, ao longo do trabalho foi apresentado, a nível de simulação, alguns controladores STC e os problemas e vantagens inerentes a cada um deles. Facilitando assim trabalhos futuros nesta área.

No modelo considerou-se que as perturbações externas são nulas. Para uma análise mais detalhada da influência da perturbações podemos usar controle robusto.

No método Auto-Ajustável de controle adaptativo (STC) é imprescindível garantir no algoritmo de estimação que os parâmetros estimados convirjam para os verdadeiros, já que se o mesmo não acontecer, poderá gerar alguma instabilidade no sistema. Tendo isto em mente o Método de Controle Adaptativo por Modelo de Referência (MRAC) é mais "robusto" e mais próprio para a aplicação visto que a estabilidade é garantida mesmo que os parâmetros estimados do sistema não convirjam para o verdadeiro (prova de estabilidade por Lyapunov). Mesmo porque o MRAC é mais usado em controle ótimo de servomecanismos determinísticos e o STC para problemas estocásticos de regulação.

Já a DSP (Digital Signal Processing) pode ter as mais variadas aplicações a nível de controle, já que esta facilita, através de seus softwares, o projeto e reprojeto de controladores e a visualização de sinais do sistema em tempo real.

Mesmo que não se tenha chegado a resultados experimentais, com as indicações condensadas aqui sobre a DSP e seu software (que antes se encontravam dispersas em vários manuais), não se teria maiores dificuldades para uma implementação prática dos algoritmos dos MQR e STC.

O estudo da placa e seu ambiente contidas aqui podem servir de base para o desenvolvimento de projetos futuros. Como por exemplo um MRAC para o servosistema hidráulico de posicionamento.

Devemos ressaltar mais uma vez aos interessados em aprofundar num determinado assunto deste trabalho, o uso da seção de referências bibliográficas.

6. Referências Bibliográficas:

[Board 93]

Dspace, "Floating-Point Controller Board-DS1102", User's Guide, 1993.

[Clarke 75]

D.W Clarke. e P.G. Gauthrep, "Self-Tuning Controller", Prec. IEEE, 129(9), pp.929-934, 1975.

[Cockpit 96]

Dspace, "Cockpit-Instrument Panel", User's Guide, 1996.

[Heintze 93]

Heintze, J., G.V. Schothorst, A.J.J. Weiden and P.C. Teerhuis "Modeling and Control of an Industrial Hydraulic Rotary Vane Actuator". Proc. IEEE 32nd CDC, San Antonio, Texas, December, pp.1913-1918, 1993.

[Lewis 93]

F.L. Lewis, C.T. Abdallah. e D.M. Dawson, "Control of Robot Manipulators". Macmillan Publishing Company, 1993.

[Malik 91]

O.P. Malik, G.S. Hope e S.J. Cheng, "Some Issues on the Practical Use of Recursive Least Squares Identification in Self-Tuning Control", Int. J. Control, Vol. 53, Nº 5, pp. 1021-1033, 1991.

[Ogata 90]

Ogata Katsuhiko, "Engenharia de Controle Moderno", Prentice-Hall do Brasil, 1990.

[RTI 95]

Dspace, "Real-Time Interface to Simulink-RTI31", User's Guide, 1995.

[Slave-DSP 93]

Dspace, "Slave-DSP Applications", User's Guide, 1995.

[Slotine 91]

J. J.E. Slotine e W. Li, "Applied Nonlinear Control", Prentice-Hall, 1991.

[Stringer 76]

J. Stringer , "Hidraulic Systems Analysis". The Macmillan Press LTD, 1976

[Trace 96]

Dspace, "Real-Time Trace Module", User's Guide, 1996.

A. Anexo I. Estudo do Hardware e Software da DS1102

A.1 Placa Controladora de Ponto Flutuante (DS1102)

O DS1102 foi especificamente projetada para o desenvolvimento de controladores digitais multivariáveis de alta velocidade e para simulações em tempo real em vários campos como robótica, atuadores elétricos e servohidráulicos, controle de veículos e muitos outros.

A DS1102 é baseada no TMS320C31 da Texas Instruments que é um processador de sinal digital de ponto flutuante (DSP) de terceira geração. O DSP é acompanhado por um conjunto de periféricos freqüentemente usados em sistemas de controle digital: conversores analógicos digitais e digitais analógicos, várias entradas e saídas digitais e interfaces para sensores incrementais. Tais características permitem o controle digital de muitas tarefas.

As informações aqui contidas e sintetizadas para um melhor entendimento podem ser encontradas em maiores detalhes no [Board 93].

A.1.1 Arquitetura

A placa DS1102 é construída em torno do processador de sinal digital de ponto flutuante (DSP) da Texas Instruments TMS320C31. Ele possui 128 K words de memória RAM. O TMS320C31 suporta um total de memória de 16 M 32-bit words incluindo o programa, dados e o espaço de entrada e saída. A figura A1 apresenta o diagrama em bloco do DS1102 com todos os periféricos disponíveis no conector externo.

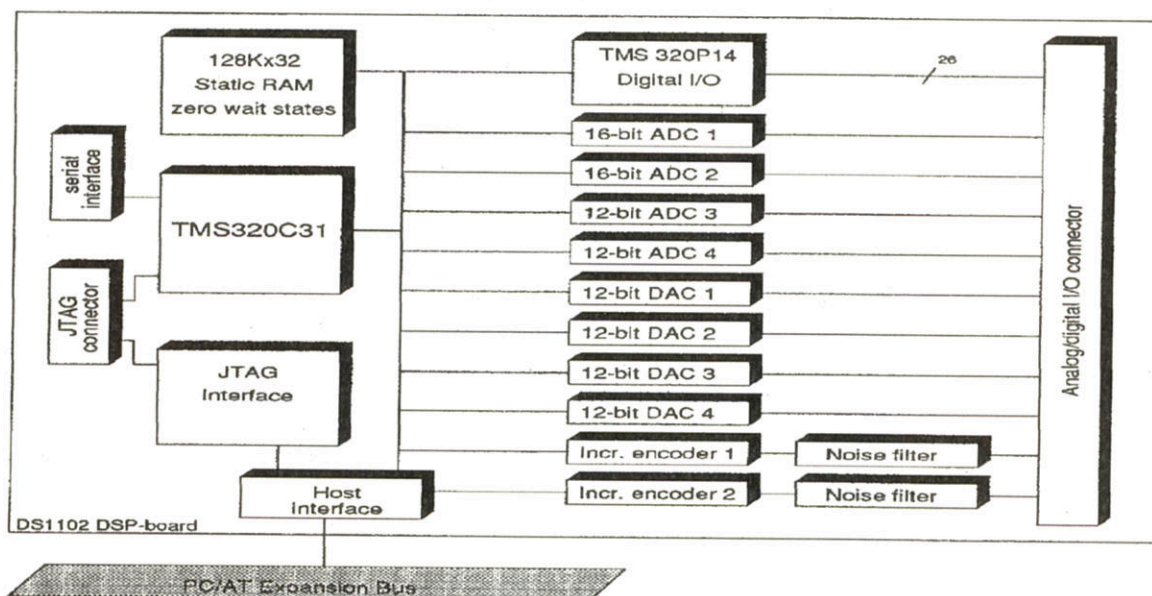


Figura A1. Diagrama em blocos do DS1102.

O subsistema digital de entrada e saída é baseado no microcontrolador TMS320P14. Este subsistema é referido aqui neste trabalho como DSP escravo. Sua função é basicamente colocar a disposição do usuário todos os periféricos deste microcontrolador para sua utilização em aplicações [Slave-DSP 93]. Para um aproveitamento total de suas funcionalidades, aconselha-se a consulta do Guia do Usuário do TMS320C14/ TMS320E14 da Texas Instruments. Para uma visão geral destas funcionalidades ver o [Board 93].

Os subsistemas seguintes referem-se ao conversor analógicos-digitais e digitais-analógicos.

O último subsistema é usado para a ligação de dois sensores incrementais óticos para controle de posição.

A.1.2 A Pinagem do Conector Externo

Pino	Sinal	Pino	Sinal	Pino	Sinal
1	ADC1	22	GND Analógico	43	DAC 1
2	ADC2	23	GND Analógico	44	DAC2
3	ADC3	24	GND Analógico	45	DAC3
4	ADC4	25	GND Analógico	46	DAC4
5	XFO	26	GND Digital	47	GND Digital
6	INTEXT	27	XF1	48	TXD
7	IOP1	28	IOP0	49	RCD
8	IOP3	29	IOP2	50	CAP0
9	IOP5	30	IOP4	51	CAP1
10	IOP7	31	IOP6	52	CAP2
11	IOP9	32	IOP8	53	CAP3
12	IOP11	33	IOP10	54	CMP0
13	IOP13	34	IOP12	55	CMP1
14	IOP15	35	IOP14	56	CMP2
15	GND Digital	36	/WDTINT	57	CMP3
16	Phi90 2	37	/Phi90 2	58	TCLK1
17	Phi0 2	38	/Phi0 2	59	TCLK2
18	Index 2	39	/Index 2	60	GND Digital
19	Phi90 1	40	/Phi90 1	61	Fonte 5V/200mA
20	Phi0 1	41	/Phi0 1	62	Fonte 5V/200mA
21	Index 1	42	/Index 1		

Tabela A1. Pinagem do conector de entrada e saída.

O conector de entrada e saída de 62 pinos apresenta a pinagem como mostra a tabela A1. Este conector possui vários grupos de sinais com as mais variadas funções, como mostra a tabela A2.

Os conversores possuem uma tensão de entrada ou de saída, conforme o tipo (AD ou DA), variando de $\pm 10V$. Os conversores analógicos-digitais de 16-bits apresentam um tempo de conversão de $10\mu s$ e os de 12-bits de $3\mu s$. Já os conversores digitais-analógicos que são de 12-bits tem um tempo de conversão de $4\mu s$. As funções referentes ao subsistema digital encontram-se indicadas como DSP escravo que não serão explicitadas.

Nome	Função
ADC 1 - 4	Entradas dos Conversores Analógicos-Digitais
DAC 1 - 4	Entradas dos Conversores Digitais-Analógicos
CAP 0 - 3	DSP escravo
CMP 0 - 3	DSP escravo
IOP 0 - 15	DSP escravo
GND Analógico	Linha de retorno analógica
GND Digital	Linha de retorno digital
INTEXT	Interrupção externa do usuário do TMS320C31(INT 0)
XF0, XF1	Pinos de entrada e saída de propósito geral do TMS320C31
RCD	DSP escravo
TXD	DSP escravo
WDTINT	DSP escravo
TCLK 1,2	DSP escravo
Phi 0, /Phi 0	Sensor incremental
Phi 90, /Phi 90	Sensor incremental
Index, /Index	Sensor incremental
Fontes de 5V/200mA	Fontes para o Sensor incremental

Tabela A2. Agrupamento dos diversos tipos de sinais

A.2 O Cockpit

O programa **Cockpit** é um painel de instrumentos, com o qual é possível obtermos valores gráficos de variáveis e uma modificação interativa das mesmas, para qualquer aplicação rodando na placa de processamento de sinal digital (DSP). É também possível modificar ou mostrar todas as variáveis representando-as como ponto flutuante de precisão simples ou como variáveis inteiras na memória da placa de processamento. A saída é formada por instrumentos como displays, medidores ou barras de movimento. As variáveis podem ser modificadas com vários tipos de botões ou através de uma entrada numérica no teclado.

O Cockpit fornece ao usuário uma série de controles, que podem ser usados para a construção de um layout específico para cada aplicação.

Devemos atentar para o fato de que o Cockpit não é usado para uma análise em tempo real, como o Trace, já que o acesso as variáveis não é feito em passos fixos de tempo.

As informações aqui contidas e sintetizadas para um melhor entendimento podem ser encontradas em maiores detalhes no [Cockpit 96].

A.2.1 Os Arquivos de Entrada e Saída do Cockpit

O Cockpit necessita de alguns arquivos de entrada que contém informações a respeito da aplicação rodando bem como alguns arquivos de configuração. Alguns destes são arquivos globais que aparecem apenas uma vez no ambiente e outros contém informação da aplicação específica.

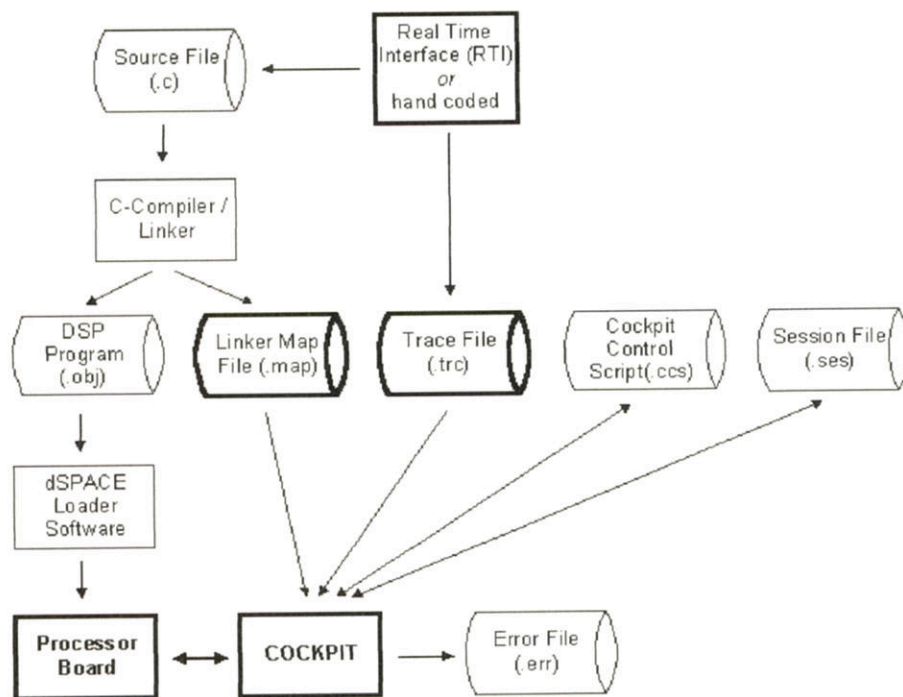


Figura A2. Arquivos de entrada e saída do Cockpit

Na figura A2 acima encontramos todos os arquivos de entrada e saída necessários para a utilização do Cockpit. O arquivos (.trc) e (.c) podem ser obtidos manualmente ou através de ferramentas automáticas de geração de código (RTI, IMPEX e RTW respectivamente). O arquivo (.trc) é um arquivo texto em ASCII contendo informações das variáveis do programa C que são necessárias para serem vistas e modificadas pelo Cockpit. O arquivo (.map) é um arquivo ASCII contendo informações das posições de memória usadas pelo programa DSP. Este arquivo é criado junto com o (.obj), que não é necessário como arquivo de entrada do Cockpit, através da compilação do arquivo (.c). O arquivo (.ccs) (Cockpit Control Script) contém informações do layout criado pelo usuário. Na figura A3 mostramos um exemplo de layout criado no Cockpit.

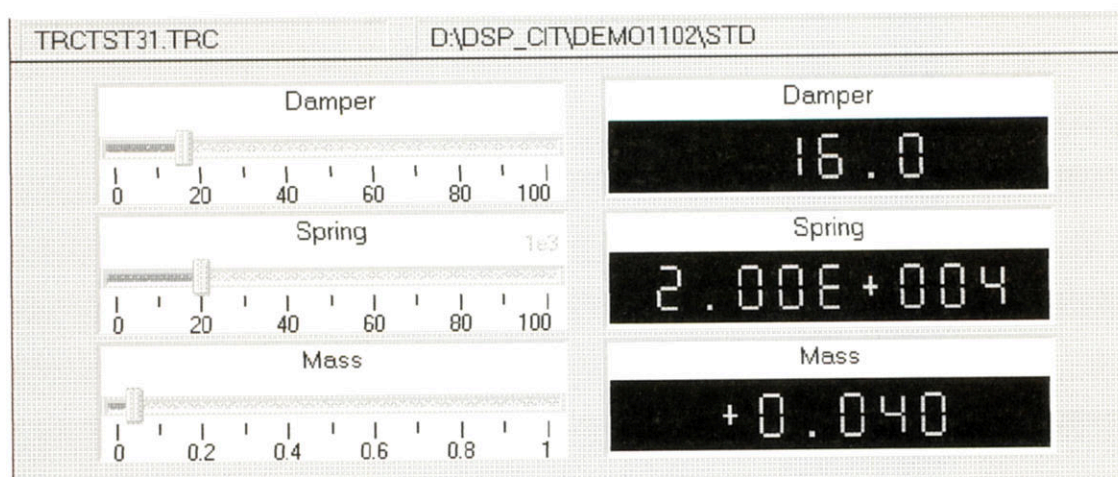


Figura A3. Exemplo de layout criado no Cockpit

O arquivo (.ses) é criado pelo Cockpit e contém somente a informação do path em que se encontra o arquivo com sufixo (.ccs) mais recentemente carregado. Este arquivo é automaticamente lido quando o Cockpit é reiniciado. Como o (.ccs) contém informações referentes ao último arquivo (.trc) usado ele é automaticamente recarregado. Então é possível suspender uma seção salvando o layout, e saindo do Cockpit. A seção será restaurada exatamente como ela estava quando o Cockpit é reiniciado. Para se evitar inconsistências o arquivo de seção (.ccs) não deve ser modificado manualmente.

A.2.2 O Modo de Edição do Cockpit

O Cockpit apresenta dois modos de operação: o modo de edição e o modo de animação. O layout de um painel de instrumentos é criado no modo de edição. Vários instrumentos podem ser escolhidos para se construir uma tela de layout de acordo com as necessidades do usuário. No modo de edição é possível colocar qualquer tipo de instrumento de qualquer tamanho em qualquer lugar do espaço de trabalho.

A.2.3 Os Instrumentos

Na Figura A4 encontramos uma lista de instrumentos disponíveis no Cockpit.

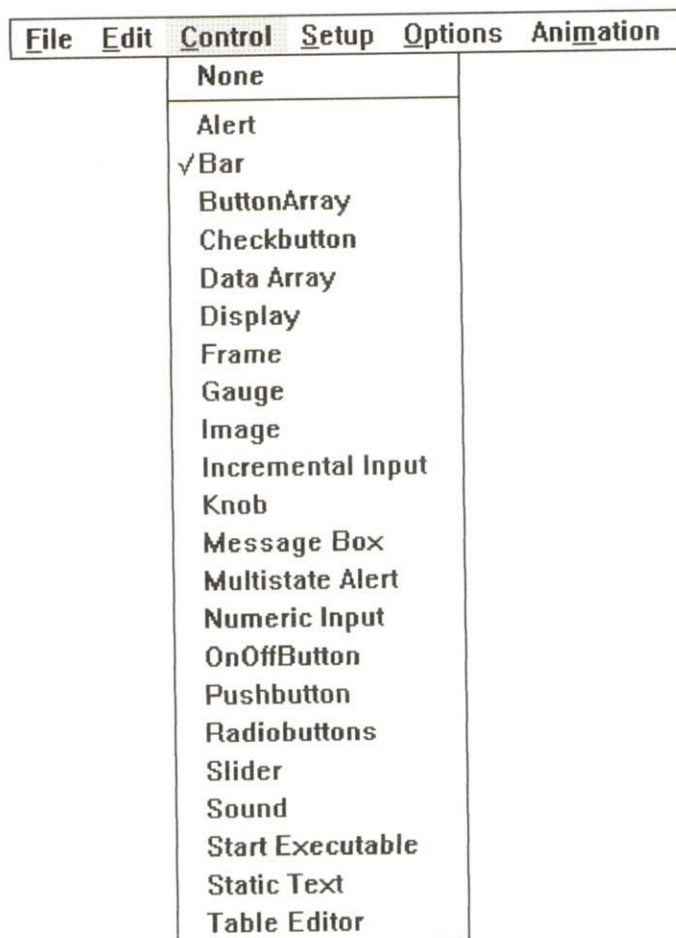


Figura A4. Lista de Instrumentos

Cada elemento possui um conjunto de parâmetros que podem ser ajustados. Um parâmetro básico muito importante é o parâmetro associado às variáveis a serem vistas e modificadas pelo Cockpit. O arquivo (.trc) define estas variáveis que aparecem na janela de diálogo, como mostra a figura A5.

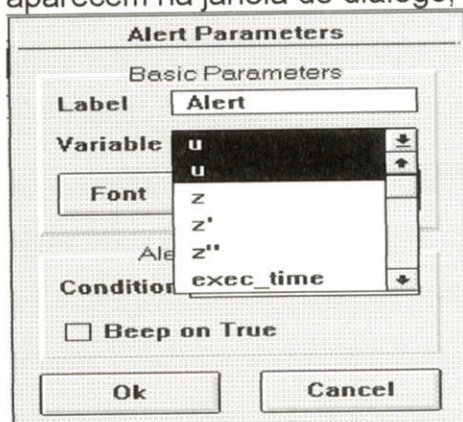


Figura A5. Variáveis presentes na janela de diálogo, definidas no arquivo (.trc).

Mas quando a Interface Tempo Real com o Matlab/ Simulink é usada a janela de diálogo contém uma Janela de Busca como mostrado na figura A6.

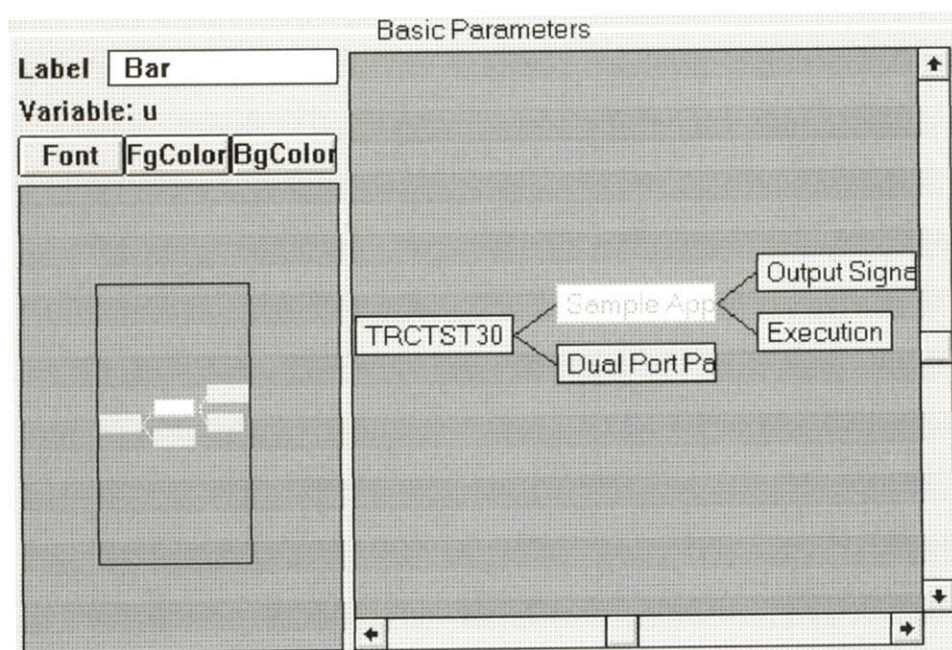


Figura A6. A Janela de Busca

A janela de busca é usada para mostrar a estrutura do arquivo (.trc) como uma árvore onde a raiz tem o nome do programa DSP.

A estrutura da árvore é mostrada duas vezes. A primeira a esquerda contém uma visão geral do painel de parâmetros básicos e contém um retângulo que mostra a parte visível da árvore, na parte direita da janela de busca.

Em cada nó temos um subconjunto de variáveis como mostrado na figura A7.

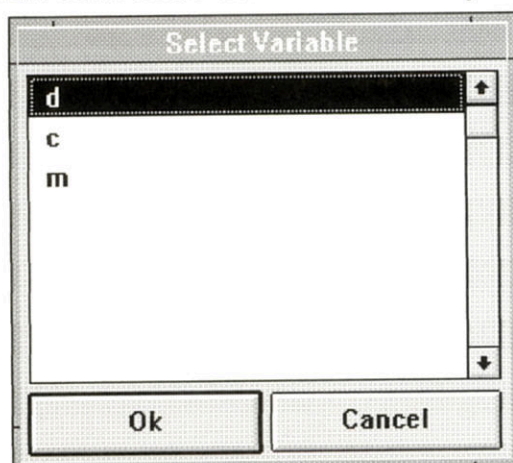


Figura A7. Seleção de variáveis de um subconjunto.

Estas podem ser seleccionadas para serem modificadas ou mostradas pelo Cockpit.

A seguir, na figura A8 mostramos alguns instrumentos contidos na lista de instrumentos na figura A4.

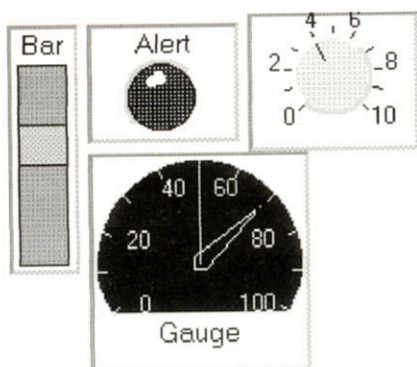


Figura A8. Exemplo de instrumentos disponíveis.

A.3 O Trace

O modulo **Trace** tem a capacidade de rastrear em tempo real aplicações rodando na DSP. O Trace grava e mostra graficamente todos os sinais como ponto flutuante de precisão simples ou inteiros na memória da placa de processamento, permitindo boa análise de qualquer aplicação enquanto que a aquisição de dados é feita em tempo real.

Mais informações sobre o Trace podem ser encontradas no [Trace 96].

A.3.1 Arquivos de Entrada e Saída do Trace

O Trace necessita de alguns arquivos de entrada contendo informações do programa que está atualmente rodando na DSP, bem como alguns arquivos de configuração. Alguns destes são arquivos globais que aparecem apenas uma vez no ambiente e outros contém informação da aplicação específica. O Trace também cria alguns arquivos de saída para armazenar várias informações.

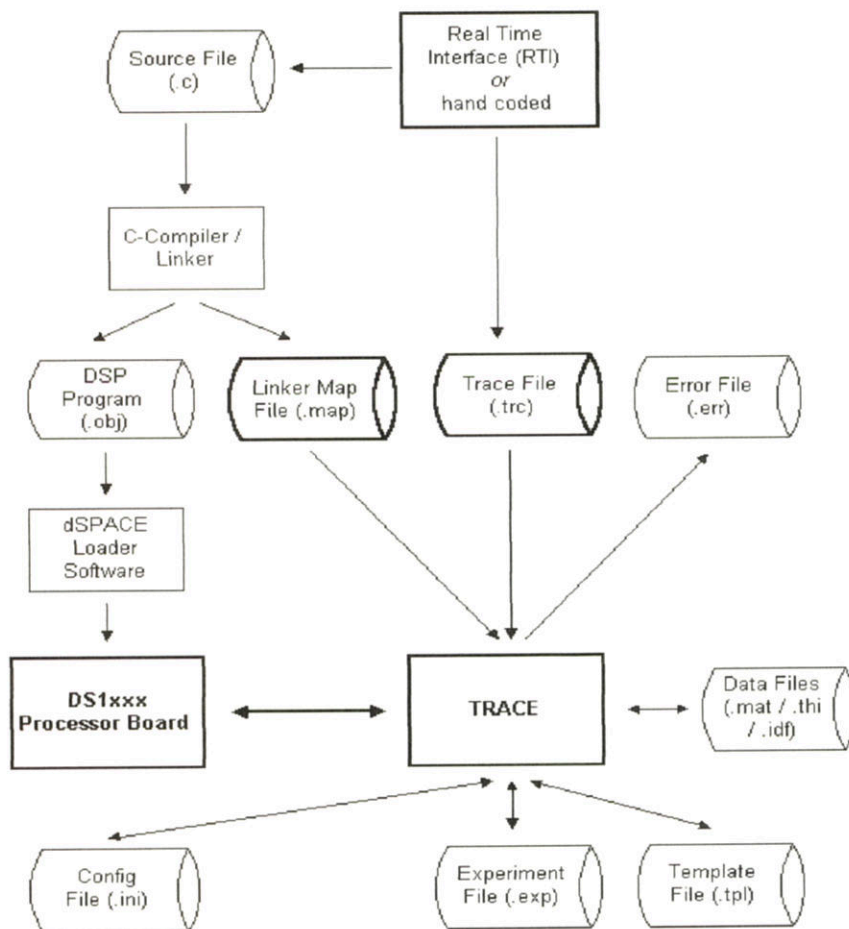


Figura A9. Arquivos de Entrada e saída do Trace.

Na figura A9 acima encontramos todos os arquivos de entrada e saída necessários para a utilização do Trace. O arquivos (.trc) e (.c) podem ser obtidos manualmente ou através de ferramentas automáticas de geração de código (RTI, IMPEX e RTW respectivamente). O arquivo (.trc) é um arquivo texto em ASCII contendo informações das variáveis do programa C a serem gravadas ou observadas graficamente pelo Trace. Todas estas variáveis devem ser variáveis globais no programa C. O arquivo (.map) é um arquivo ASCII contendo informações das posições de memória usadas pelo programa DSP. Este arquivo é criado junto com o (.obj), que não é necessário como arquivo de entrada do Cockpit, através da compilação do arquivo (.c).

Se houver algum erro no arquivo (.trc) ou (.map) um arquivo de erro (.err) é gerado.

O arquivo (.ini) é gerado automaticamente e contém informações como cores usadas nos gráficos e os paths do arquivos de experimentos carregados recentemente etc.

Os arquivos (.mat), (.thi) e (.idf) são arquivos que permitem armazenar dados no disco para carregá-los na memória posteriormente. Os (.mat) são do formato Matlab, os (.idf) do formato do Trace e os (.thi) que é uma tabela de dados em formato ASCII.

O arquivo (.tpl) contém informações sobre o layout da janela onde encontram-se os gráficos. O arquivo (.exp) apresenta informações sobre o ambiente do experimento como: as janelas dos Gráficos aberta atualmente, o arquivo (.trc) relacionado, o tamanho, posição e configurações do painel de controle, etc.

A.3.2 Como Operar o Trace

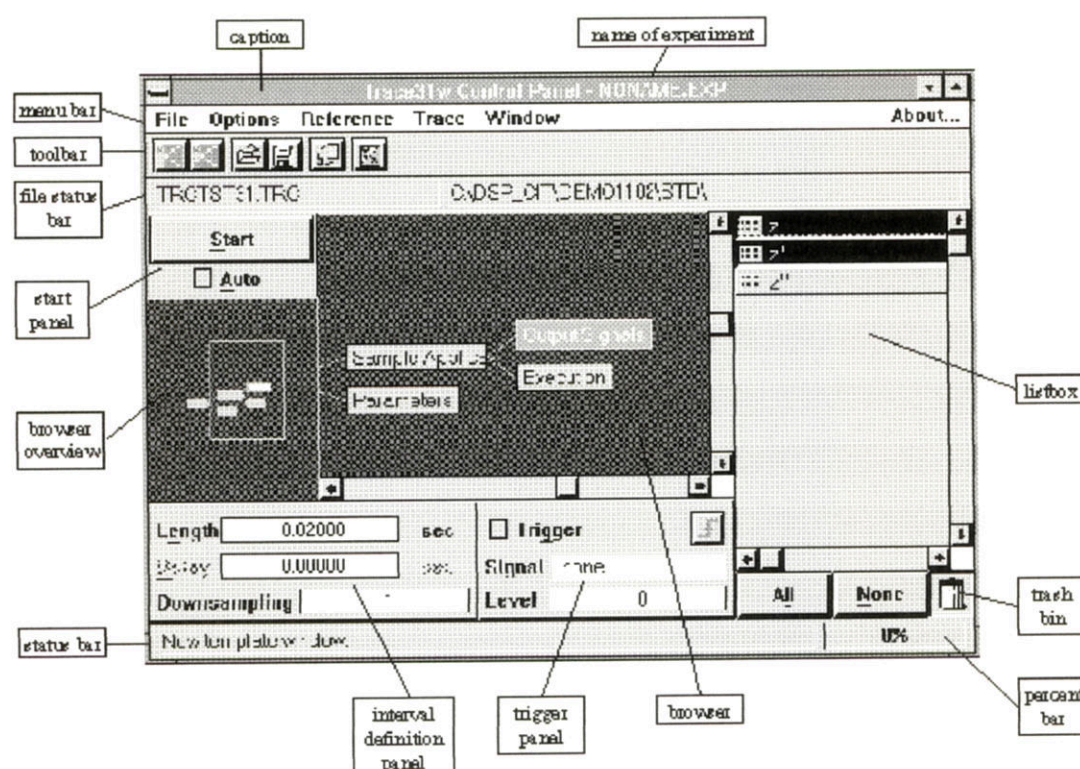


Figura A10. O painel de controle.

Antes de invocar o programa Trace devemos carregar alguma aplicação na DSP usando o comando **down31 name.c** no DOS. Este comando compila e carrega a aplicação na placa. Se a aplicação estiver implementada no SIMULINK através da caixa de diálogo obtida do menu **Code**, "clitando" sobre o botão **Build** (figura A19). A seguir podemos chamar o programa no Windows.

A figura A10 apresenta a janela do painel de controle do Trace. Este controla os parâmetros de captura do Trace, seleção de sinais, a definição e seleção de gráficos, o modo de aquisição de dados, etc.

O espaço principal do painel de controle é ocupado por uma janela de busca como já foi descrita anteriormente no Cockpit. A janela de busca é usada para selecionar os sinais a serem observados. Depois da escolha do nó, as variáveis serão listadas ao lado e a partir destas poderemos definir gráficos, a partir do item **Options - Define XY-Plot**, como mostra a figura A11.

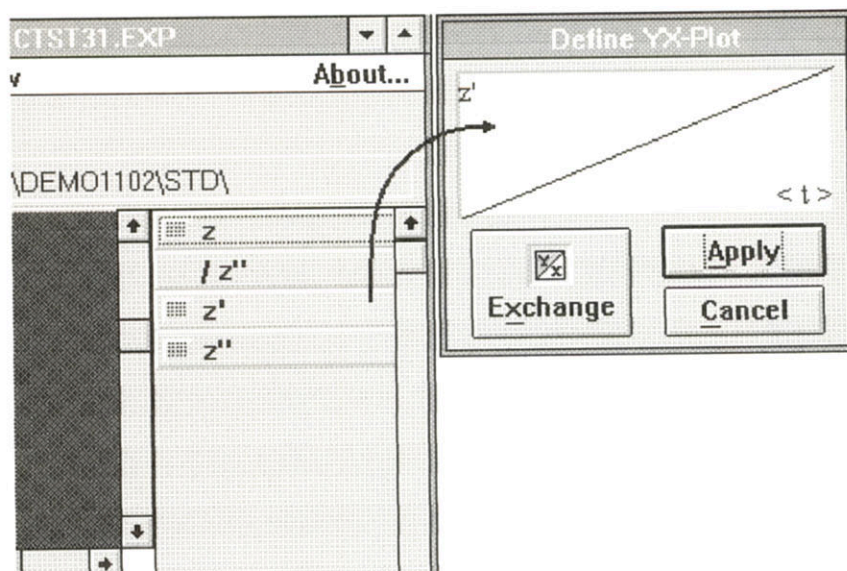


Figura A11. Janela de definição do gráfico.

A partir desta mesma lista de variáveis obtemos uma janela de gráficos, simplesmente "clicando" sobre estas. A figura A13 abaixo apresenta uma janela de gráficos que mostra o gráfico da evolução das variáveis em tempo real. Esta janela de gráficos oferece uma série de serviços que permitem uma melhor análise dos gráficos: como o "zoom" em um determinado ponto do gráfico (figura A12), a superposição de dois ou mais sinais, etc.

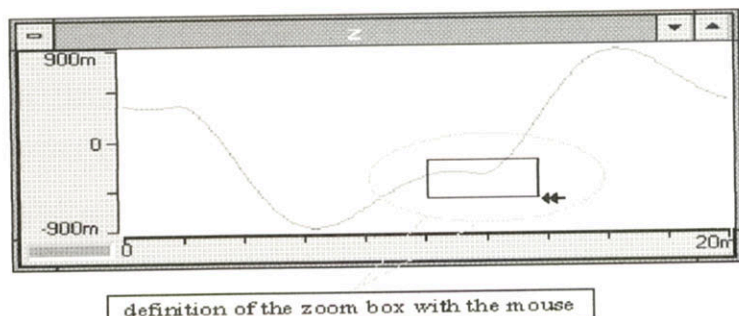


Figura A12. O definição do retângulo de "zoom" com o mouse.

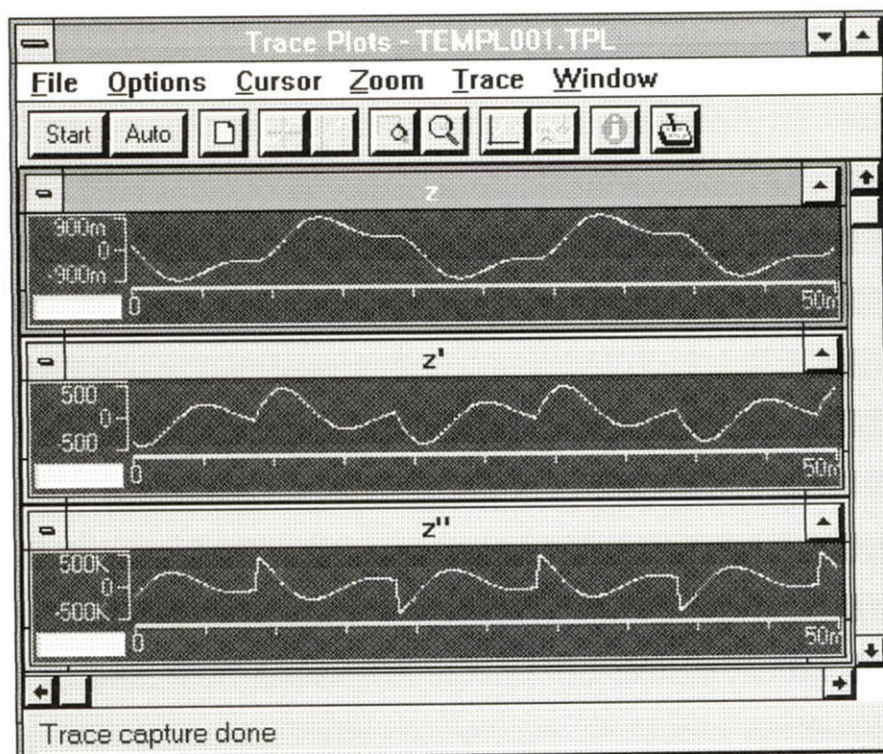


Figura A13. Janela de gráficos.

O intervalo de tempo de captura (amostragem) é definido pelo painel da figura A14, que se encontra a direita do painel de controle (figura A10), no campo denominado de **Length**. O campo **Delay** especifica o tempo de início da simulação a partir de um determinado evento "gatilho", se tal especificação não for necessária este deve se setado a zero. O campo **Downsampling** é sempre positivo e determina de quantos em quantos intervalos é feita a coleta de dados pelo Trace. Por exemplo se este campo for igual a **n** a coleta é feita cada **n**ésima amostragem.

Length	<input type="text" value="0.02000"/>	sec
Delay	<input type="text" value="0.00000"/>	sec
Downsampling	<input type="text" value="1"/>	

A14. Painel que define o tempo de captura.

O modo como é feita a aquisição de dados é invocado a partir do painel de controle **Options - Acquisition Mode**, como mostra a caixa de dialogo da figura A15. Nesta o primeiro painel **Mode**, possui três opções onde se define o modo de aquisição:

1. mostrando os dados logo após a captura;
2. mostrando apenas os dados já disponíveis, onde a captura já foi feita. Esta opção é usada em caso de processos lentos que devem ser observados por um longo período de tempo;
3. a aquisição feita de modo contínuo, é similar a segunda opção mais esta não tem fim.

O campo **Percent** define uma porcentagem de dados que devem ser coletados para ocorrer a atualização no gráfico. O campo a direita indica o valor absoluto equivalente da porcentagem do período de amostragem. Na figura A15 indica que a cada 10% de dados coletados ocorre a atualização do gráfico.

Figura A15. Caixa de dialogo de aquisição de dados.

A opção **Automatic Zoom into data** força ao eixo **x** a atualizar-se automaticamente conforme os dados vão sendo coletados. Esta atualização é definida pelo campo **Factor**. A figura A15 indica que a cada 30% de dados coletados o gráfico é atualizado. O equivalente em tempo de amostragem é indicado no campo ao lado.

As opções em **Disk Options** são usadas para salvar os dados automaticamente no disco.

A.4 Interface Tempo Real para o Simulink

A interface tempo real (RTI) conecta o Matlab, o Simulink e o RTW (Real Time Workshop) com o DSP para aplicações em tempo real. Permite, por exemplo, a implementação de protótipos de controladores (contínuos, discretos ou híbridos) para sistemas físicos, e sua simulação em tempo real. A figura A16 mostra esta relação do RTI com o Matlab.

Para maiores informações ver [RTI 95].

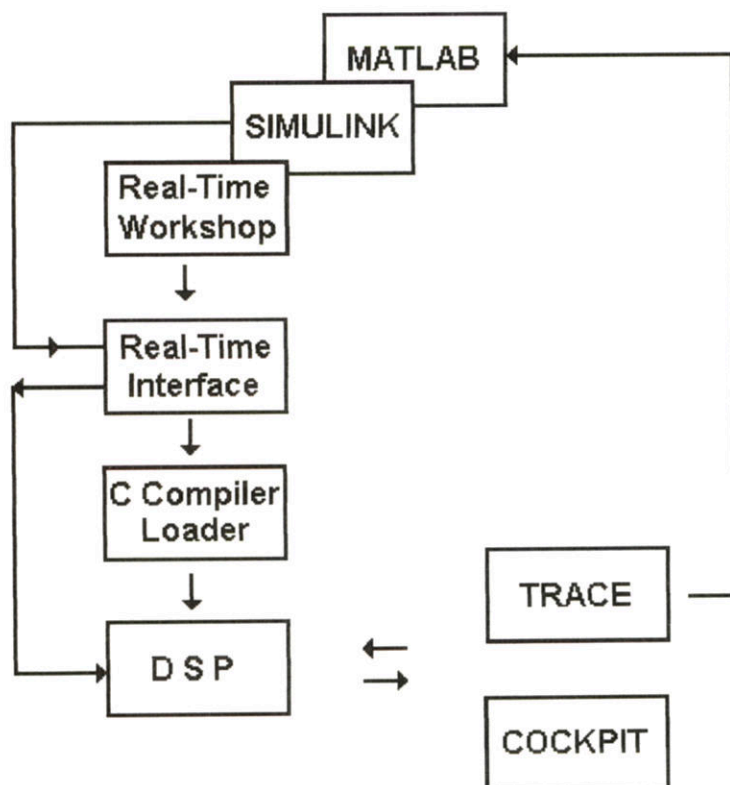


Figura A16. Relação do RTI com o Matlab.

Para nos familiarizarmos com o RTI e o RTW utilizaremos um exemplo na seção A.4.2.

A.4.1 O Simulink

O Simulink é um conjunto de funções gráficas usadas para a descrição e simulação de sistema. Este possui diversas bibliotecas com diversos subsistemas para a composição do sistema em questão. O Simulink é invocado digitando-se **Simulink** no prompt do Matlab.

Ver figura A16 abaixo.

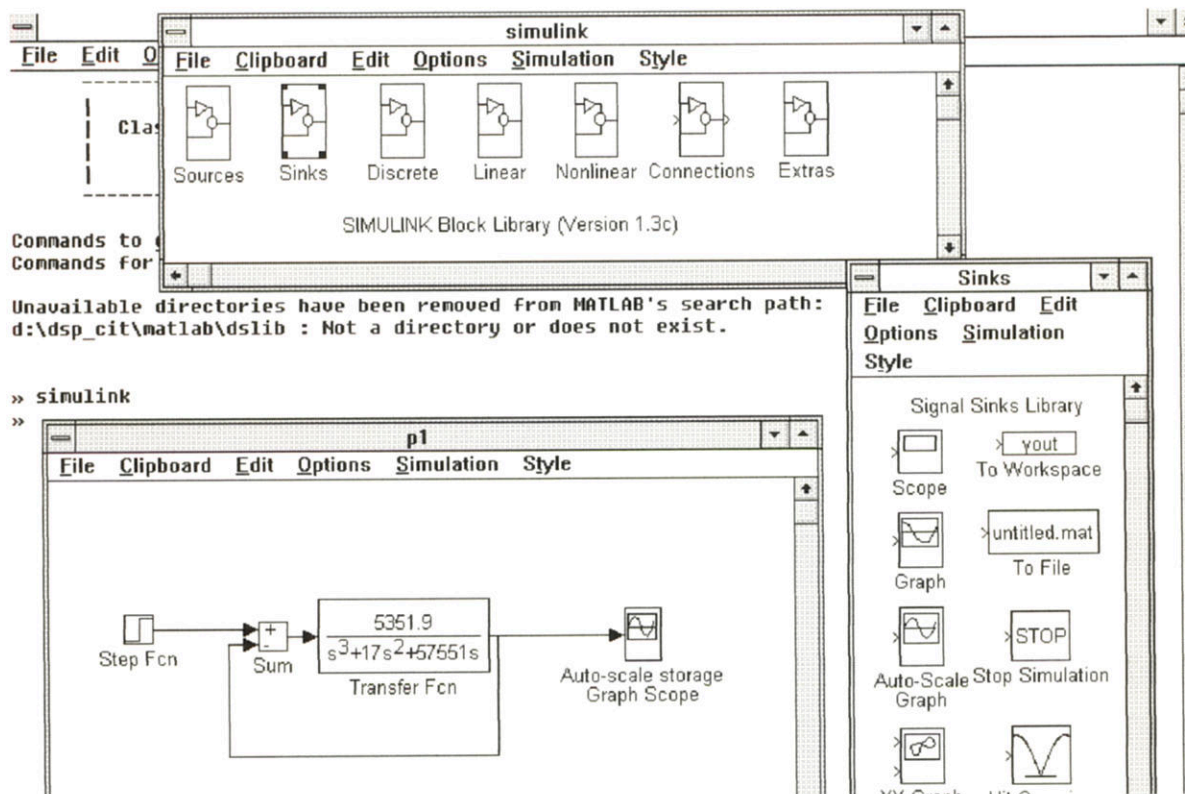


Figura A16. Implementação no Simulink de um sistema.

A figura indica na janela **p1** a representação do sistema e na janela **Sinks** uma biblioteca de "pontos de saída", que no caso de **p1** é um gráfico em auto-escala.

A.4.2 Exemplo de Familiarização

Na figura A17 encontramos um diagrama que implementa um modelo de segunda ordem com um gerador de sinal de entrada e um bloco para a geração de gráfico na saída. Este modelo não usa qualquer hardware de entrada e saída da DSP1102.

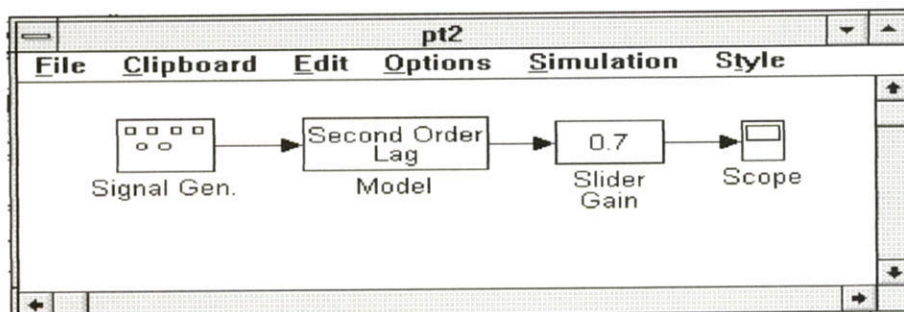


Figura A17. Modelo de segunda ordem implementado no Simulink.

A resposta da simulação "off-line" encontra-se na figura A18.

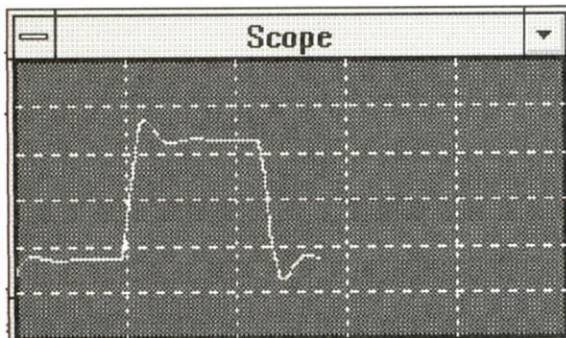


Figura A18. Resposta do sistema de 2ª ordem "off-line".

Para este sistema rodar em tempo real na placa usa-se a opção **Code** do menu (esta opção somente aparece com o RTW instalado). Veja a figura A19 abaixo.

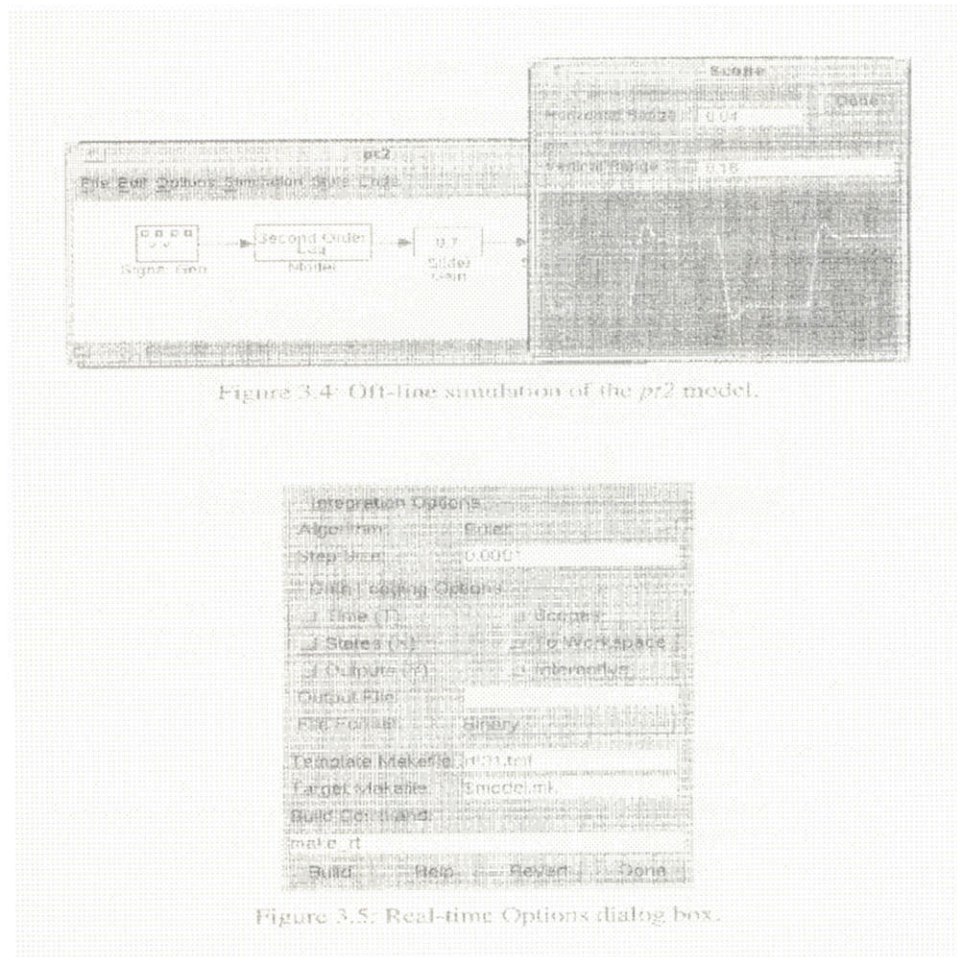


Figure 3.4: Off-line simulation of the *pr2* model.

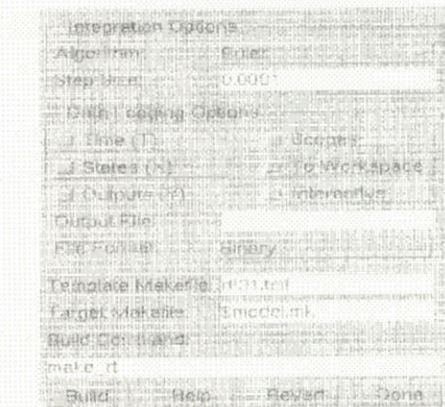


Figure 3.5: Real-time Options dialog box.

Figura A19. Simulink com opção **Code** e logo abaixo a caixa de diálogo desta opção.

O texto de diálogo da figura A19 **Template Makefile** deve ser sempre `rti31.tmf`. Na opção de integração pode-se selecionar o método de integração (**Euler** no caso) e o tamanho do passo de integração. O valor deste passo indica a base do tempo de amostragem para a aplicação executada em tempo real. Para sistema discretos temos a opção de não colocar valor algum ou o menor valor de tempo de amostragem usado no modelo.

No texto de diálogo **Target Makefile** cria-se um arquivo com extensão `(.mk)` e indica o nome da aplicação específica a ser criada.

O **Build Command** é sempre o programa `make_rt`.

Em **Data Logging Options** selecionamos se os dados de simulação devem ir para um arquivo ou gráfico. Estas opções não se encontram disponíveis em caso aplicações em tempo real. Para observarmos os dados de aplicações em tempo real usamos o Trace.

Após estas especificações serem feitas corretamente "clicamos" sobre o **Build** e se a implementação estiver correta aparecerá a mensagem **DOWNLOAD SUCCEEDED**.

Em seguida chamamos o programa Trace com seu respectivo arquivo `(.trc)` e obtemos os gráficos da simulação como foi descrito anteriormente

Na figura A20 mostramos um exemplo de sistema com entrada e saída externas.

Após seu carregamento na placa conectamos um gerador de sinais no primeiro conversor AD (pino 1) e um osciloscópio no primeiro conversor DA (pino 43) para observarmos sua saída.

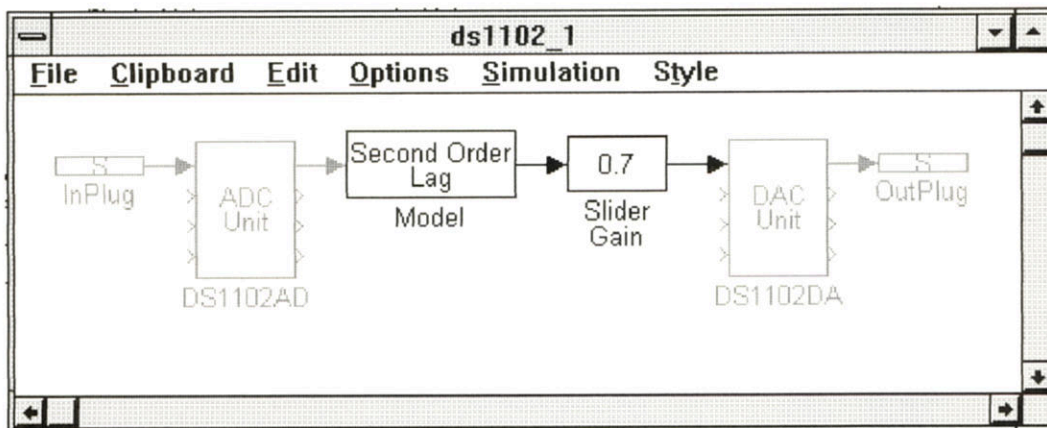


Figura A20. Sistema de 2ª ordem com entrada e saída externas.

Os blocos **InPlug**, **OutPlug**, **DS1102AD** e **DS1102DA** são obtidos de uma biblioteca do Simulink.

Um detalhe importante é que o modelo a ser desenvolvido no Simulink deverá ser condicionado para que as saídas (e entradas) tenham uma tensão variando de **-1 a +1 V** (ver figura A21). com este tipo de escala a resolução dos conversores é completamente explorada.

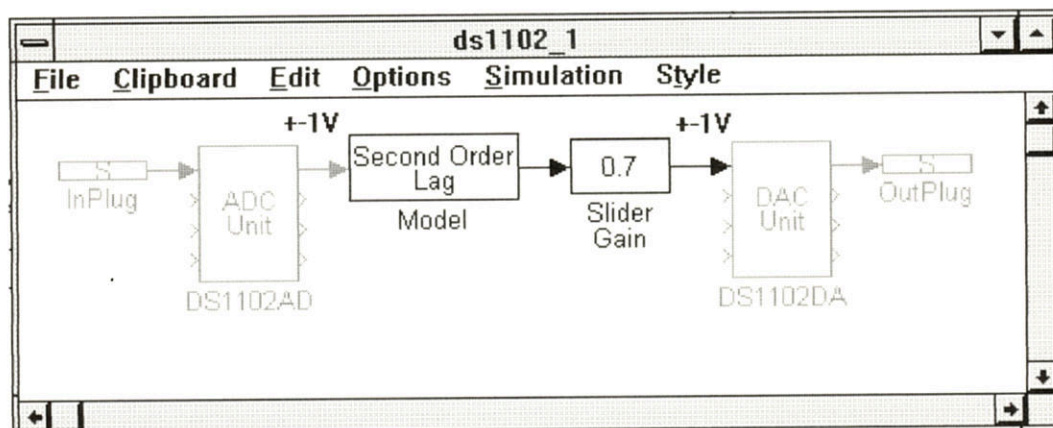


Figura A21. Condicionamento necessário para o modelo.

B. Anexo II. Algoritmo do Matlab Usado nas Simulações

```

%=== CONTROLADORES ADAPTATIVOS PARA UM SERVOSISTEMA
HIDRÁULICO===%
%=== ESTIMAÇÃO DOS PARÂMETROS COM MQR =====%
%=== VARIAÇÃO DOS PARÂMETROS =====%
%=== A ENTRADA DO SISTEMA É ||U||<=866 =====%

clear
clg

y=[0 0 0 0 0 0]
%y=[1 1 1 1 1 1]

u=[0 0 0 0 0 0]

%teta=[0 0 0 0 0 0]
%teta=[0.36 -0.52 -0.84 0.0006 0.0012 0.0006]
%m=0.18
%teta=[-0.18 0.0372 -0.84 0.0005 0.0015 0.0005]

% m=0.16
teta=[0.0079 -0.16 -0.84 0.0005 0.0015 0.0005]

nit=input('numero de iteracoes=');
pinic=input('matriz de covariancia inicial=');
for i=1:6
    for j=1:6
        p(i,j)=0.0;
    end
    p(i,i)=pinic;
end

u(1)=1;
u(2)=1;u(3)=1;
eref(2)=-1;eref(3)=-1;

r=rand(nit,1)/1000;

u=rand(nit,1);

v(3)=0;

```

```
trp0=1000;
```

```
fe=.96;
```

```
for k=4:nit
```

```
    yref(k)=10*sin(2*pi*.003*k);
```

```
    %if k>100 & k<=200
```

```
        % yref=0;
```

```
    %end
```

```
    %if k>1000 & k<=2000
```

```
        % yref=1;
```

```
    %end
```

```
    %if k>300 & k<=400
```

```
        %yref=1;
```

```
    %end
```

```
    %if k>400 & k<=500
```

```
        % yref=0;
```

```
    %end
```

```
    %m=.1287
```

```
    yaux=-0.36*y(k-1)+0.52*y(k-2)+0.84*y(k-3);
```

```
    uaux=6.45e-04*u(k-1)+1.26e-03*u(k-2)+6.31e-04*u(k-3);
```

```
    %m=0.16
```

```
    %yaux=-0.0079*y(k-1)+0.16*y(k-2)+0.84*y(k-3);
```

```
    %uaux=5e-04*u(k-1)+1.6e-03*u(k-2)+5e-04*u(k-3);
```

```
    %m=0.18
```

```
    %if k >100
```

```
        %yaux=0.19*y(k-1)-0.0372*y(k-2)+0.84*y(k-3);
```

```
        %uaux=5e-4*u(k-1)+1.5e-3*u(k-2)+5e-4*u(k-3);
```

```
    %end
```

```
    y(k)=yaux+uaux;
```

```
    vm=[-y(k-1) -y(k-2) -y(k-3) u(k-1) u(k-2) u(k-3)];
```

```
    yhat=vm*teta';
```

```
    e1(k)=y(k)-yhat;
```

```
    eaux=0;
```

```
    for i=1:k
```

```
        eaux=eaux+e1(i);
```

```
    end
```

```
    e1m=eaux/k;
```

```
    v(k)=1/k*(((k-1)*v(k-1))+(e1(k)-e1m)^2);
```

```

    ganho=(p*vm')/(1+vm*p*vm');
    %ganho=(p*vm')/(fe+vm*p*vm');
    teta=teta+(ganho)*e1(k);
    %p=(1/fe)*(p-ganho*(fe+vm*p*vm')*(ganho'));
    p=p-ganho*(1+vm*p*vm')*(ganho');

    %Cálculo do Traço de P
    trp=p(1,1)+p(2,2)+p(3,3)+p(4,4)+p(5,5)+p(6,6);
    trpa(k)=p(1,1)+p(2,2)+p(3,3)+p(4,4)+p(5,5)+p(6,6);

    %Random Walk
    %if trp<trp0 ,
    %qd=1000
        %qd=trp/2
    %q=[qd 0 0 0 0 0;0 qd 0 0 0 0;0 0 qd 0 0 0;0 0 0 qd 0 0;0 0 0 0 qd 0;0 0 0 0
0 qd];
    %p=p+q;
    %end;

    %Covariance Resetting
    %if trp<trp0
    %qd=10000;
        %qd=trp/6;
    %q=[0 0 0 0 0 0;0 0 0 0 0 0;0 0 0 0 0 0;0 0 0 qd 0 0;0 0 0 0 qd 0;0 0 0 0 0
qd];
    %p=p+q;
    % end

    a0(k)=teta(1,1);
    a1(k)=teta(1,2);
    a2(k)=teta(1,3);
    b0(k)=teta(1,4);
    b1(k)=teta(1,5);
    b2(k)=teta(1,6);
    eref(k)=-y(k)+yref(k);

%==== CONTROLADOR POR ALOCAÇÃO DE PÓLOS (0.4 0.5 0.6) ==== %
    % g2(k)=-0.12-a2(k);
    % g1(k)=0.74-a1(k);
    % g0(k)=-1.5-a0(k);

%==== CONTROLADOR POR ALOCAÇÃO DE PÓLOS ==== %
%==== POLOS EM 0.9 0.9 0.9      ===== %

```



```

%g2(k)=-0.729-a2(k);
%g1(k)=2.43-a1(k);
%g0(k)=-2.7-a0(k);

%==== CONTROLADOR POR ALOCAÇÃO DE PÓLOS (0.1 0.2 0.3)====%

%g2(k)=-0.006-a2(k);
%g1(k)=0.11-a1(k);
%g0(k)=-0.6-a0(k);

%====PD Auto Ajustável=====%
SS=[a0(k) 1 b0(k) 0
a1(k) a0(k) b1(k) b0(k)
a2(k) a1(k) b2(k) b1(k)
0 a2(k) 0 b2(k)];

CS=[-0.5427
-0.8155
-0.3859
0.7546];

DS=SS\CS

%u(k)=(((b1(k)*u(k-1)-b2(k)*u(k-2)+g0(k)*eref(k)+g1(k)*eref(k-1)+g2(k)*eref(k-2))/b0(k)));
%+r(k);

%u(k)=((DS(3)*eref(k)+DS(4)*eref(k-1)-DS(2)*u(k-1)))/DS(1);

u(k)=70*eref(k)-65.9104*eref(k-1)+.9479*u(k-1);
%m=.18;
%u(k)=(-2178*eref(k)+5093*eref(k-1)+2.1*u(k-1))/8.4;

%m=0.16;
%u(k)=(-629.1*eref(k)+1096*eref(k-1)+0.2*u(k-1))/2.2;

if u(k)>=866
u(k)=866;
end

if u(k)<=-866

```

```
u(k)=-866  
end
```

```
end
```

```
t=5:nit;  
subplot(221),plot(t*0.01,u(t)),title('ENTRADA'),xlabel('segundos');  
subplot(222),plot(t*0.01,y(t)),title('SAIDA'),xlabel('segundos');  
subplot(223),plot(t,eref(t)),title('ERRO'),xlabel('AMOSTRAGEM');  
subplot(224),plot(t,yref(t)),title('yref'),xlabel('AMOSTRAGEM');
```