

APLICAÇÕES EM CONTROLE DE UMA ARQUITETURA PARALELA BASEADA EM PROCESSADORES DIGITAIS DE SINAL*

Elder M. Hemerly
CTA - ITA - IEEE
Instituto Tecnológico de Aeronáutica
12228-900 - São José dos Campos - SP

RESUMO - Este trabalho apresenta aplicações em controle H^∞ e adaptativo de uma arquitetura empregando dois processadores digitais de sinal de terceira geração TMS320C30, com aritmética em ponto flutuante. Esta arquitetura permite a implementação, a baixo custo, de algoritmos de controle computacionalmente complexos, tais como aqueles resultados das técnicas H^∞ e adaptativas. Detalhes sobre a implementação em tempo real de um controlador H^∞ para foguete e de um controlador auto-sintonizável para controle de temperatura são apresentados. Em ambos os casos avalia-se o desempenho do sistema de controle e calcula-se os tempos de execução dos algoritmos, para comparação com os tempos obtidos com implementação em microcomputador tipo PC/AT-286.

ABSTRACT - This work describes applications of a DSP based architecture to H^∞ and adaptive control. This architecture employs two third generation TMS320C30 DSP's, thereby allowing the real time implementation of high performance control algorithms at reduced cost. Details concerning the implementations of an H^∞ controller for rocket guidance and a self-tuning temperature controller are presented. The performance of both control systems are investigated. The execution times of the algorithms are also evaluated and compared with those obtained with implementation in an IBM PC/AT-286 compatible computer.

1 - INTRODUÇÃO

A implementação em tempo real de estratégias de controle pode requer grande capacidade computacional, caso o algoritmo de controle seja complexo ou altas taxas de amostragens sejam necessárias. Embora o emprego de microprocessadores e microcontroladores viabilize a implementação em alguns casos menos restritivos, a utilização de processadores digitais de sinal parece indicada para satisfazer os requisitos computacionais da estratégias de controle de alto desempenho. Vide (Gurubasavaraj, 1989; Battilotti e Ulivi, 1990; Radivojevic et. al., 1991). Basicamente, esses processadores possuem arquitetura especialmente projetada para efetuar operações aritméticas, de modo rápido e eficiente. Para detalhes, vide (Lee, 1990).

Diversas aplicações de processadores digitais de sinal em filtragem e controle têm sido reportadas. Em (Tan e Kyriakopoulos, 1988) utiliza-se o DSP TMS32010, com aritmética inteira, para implementar um filtro de Kalman para rastrear um objeto móvel, concluindo-se que a versão tridimensional do filtro pode ser executada a 1,12 KHz. Apenas simulação é apresentada, e grande esforço de simulação é necessário para normalizar as variáveis, devido à aritmética inteira do DSP utilizado. Em (Yeh, 1991) considera-se a implementação de filtros de Kalman no DSP32C, da AT&T, que possui aritmética em ponto flutuante. Basicamente, avalia-se o tamanho do código e o tempo de execução em função do número de estados. Aplicações em tempo real, utilizando o TMS320C30 com aritmética em ponto flutuante da Texas

* - artigo submetido em 08/10/92

1ª revisão - 09-10-92 2ª revisão 19/05/93

aceito por recomendação do editor consultor Prof.Dr.Fernando A.C.Gomide

Instruments, podem ser encontradas em (Santos e Hemerly, 1992).

Aplicações em controle são sugeridas em (Battilotti e Ulivi, 1990; Radivojevic et.al.,1991), onde são comentados os potenciais benefícios de se utilizar processadores digitais de sinal. Contudo, nenhuma aplicação efetiva é descrita. Em (Gurubasavaraj,1989) descreve-se uma aplicação de controle adaptativo em tempo real, relativo ao controle de um servomecanismo DC de velocidade. A implementação é feita no TMS32010, com o inconveniente já citado de possuir aritmética inteira.

Neste trabalho utiliza-se a arquitetura multiprocessadora proposta em (Santos e Hemerly, 1992), utilizando dois DSP's tipo TMS320C30, para implementar em tempo real duas estratégias de controle. Mais especificamente, descreve-se a implementação de um controlador H^∞ para foguete e de controlador auto-sintonizável para controle de temperatura. As principais contribuições em relação a (Gurubasavaraj, 1989; Battilotti e Ulivi, 1990; Radivojevic et. al., 1991) são: 1) Utiliza-se processador digital de sinal de terceira geração, com aritmética em ponto flutuante; 2) Descreve-se aplicações realistas, avaliando-se também o desempenho do sistema de controle; 3) Determina-se o *speedup* em relação à implementação em microcomputador AT-286; 4) Ilustra-se a utilização de dois DSP's, operando em paralelo, que é relevante em aplicações nas quais um único DSP não consegue satisfazer as exigências computacionais.

Este trabalho está organizado conforme descrito a seguir. Na seção 2 faz-se um resumo da arquitetura utilizando DSP's proposta em (Santos e Hemerly, 1992). Na seção 3 é apresentado um resumo do projeto de um controlador H^∞ para foguete, objetivando manter bom desempenho face às variações paramétricas e perturbações externas. Na seção 4 é descrito o projeto de um controlador auto-sintonizável, tipo

GPC(*Generalized Predictive Control*), para controle de temperatura. Resultados da implementação em tempo real são mostrados e discutidos na seção 5.

2 - ARQUITETURA PARALELA UTILIZANDO DSP's

O diagrama de blocos da arquitetura proposta em (Santos e Hemerly, 1992) é mostrada na fig. 1. Com esta arquitetura objetiva-se proporcionar capacidade numérica para a implementação de algoritmos de controle sofisticados ou para controle de sistemas que requerem altas taxas de amostragem. Um microcomputador hospedeiro tipo IBM-PC é utilizado como interface com o usuário e geração de código para os processadores.

Além de dois DSP's tipo TMS320C30 (Texas Instruments, 1990a), cuja função é executar processamento numérico intensivo, a arquitetura emprega um microcontrolador PCB80C552(Philips, 1989), que pode ser utilizado para aquisição de dados, comunicação com o microcomputador e execução de tarefas que requerem pouco processamento numérico. Convém ressaltar que o TMS320C30 é um processador com aritmética em ponto flutuante (mantissa de 32 bits e expoente de 8 bits), com barramento diferentes para instruções e dados. Adicionalmente, possui um conjunto de instruções especiais para processamento digital de sinais, resultando alta velocidade de processamento e grande versatilidade. Vide (Texas Instruments, 1990a) para maiores detalhes.

A arquitetura mostrada na fig. 1, e utilizada neste trabalho, foi montada em *wire wrap* e diversos *software* para descarga de código, verificação da integridade do código descarregado, acesso e limpeza de memória foram desenvolvidos. A geração de código para o TMS320C30 foi efetuada empregando-se

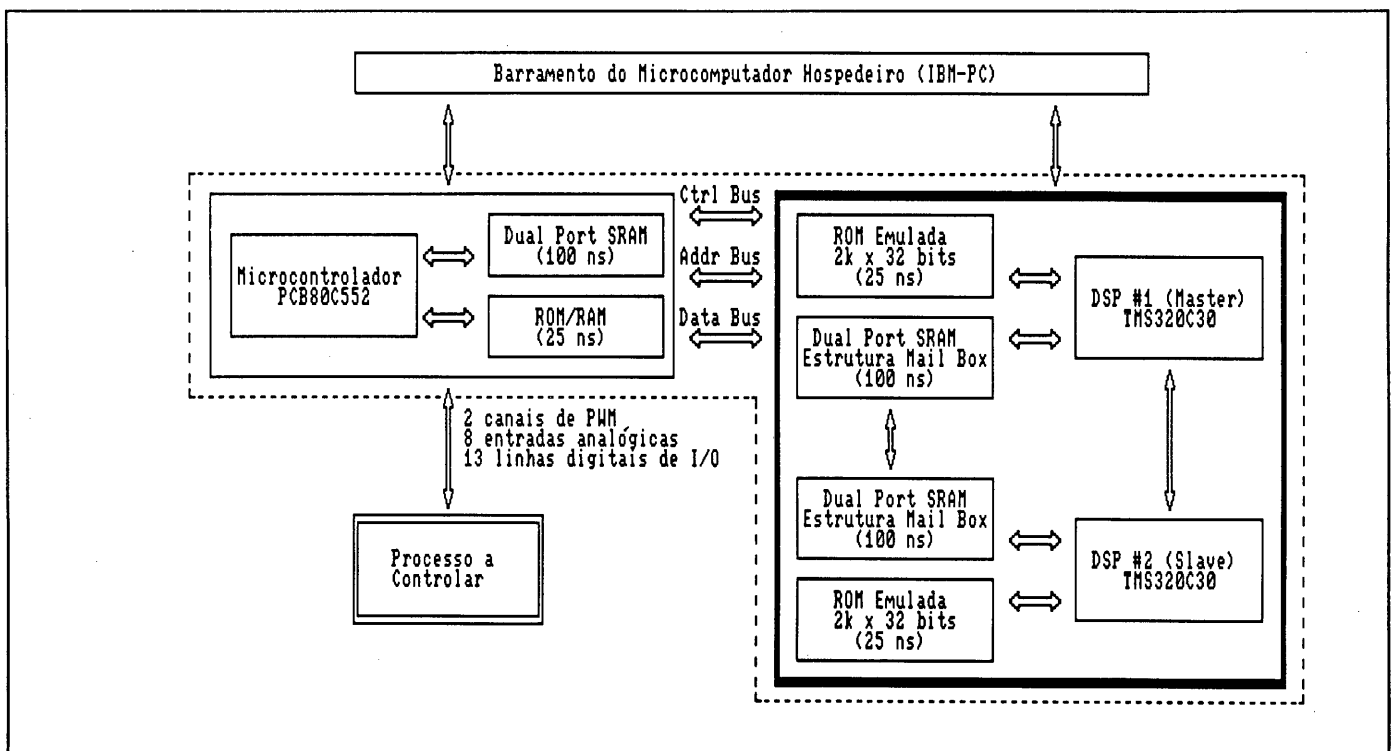


Figura 1 - Diagrama de Blocos da Arquitetura Paralela Baseada em DSP's

compilador C (Texas Instruments, 1990b), sendo a inicialização do processador e gerenciamento de interrupções escritas em *in line assembly*.

3 - CONTROLADOR H^∞ PARA FOGUETE

A técnica de controle H^∞ proporciona um procedimento formal para se projetar sistemas de controle multivariáveis robustos a variações paramétricas e/ou perturbações externas. Vide, por exemplo, (Francis, 1987; Francis e Doyle, 1987) para fundamentos teóricos e (Glover e Doyle, 1988) para detalhes sobre método eficiente de se determinar a solução do problema de otimização.

A seguir é apresentado um resumo do projeto de um controlador H^∞ para foguete, apresentado com detalhes em (Moreira e Hemerly, 1992; Moreira, 1992). Convém ressaltar que este exemplo é representativo, pois ilustra adequadamente a utilização da técnica H^∞ em problemas onde: a) o modelo dinâmico do processo controlado é multivariável e relativamente complexo; b) robustez é essencial, pois há variações paramétricas consideráveis e perturbações externas severas; e) o emprego de técnicas convencionais, tipo PID, não satisfaz os requisitos de projeto.

Conforme (Greensite, 1970; Leite Filho, 1987), a dinâmica linearizada do foguete para manobras de arfagem é mostrada na fig. 2. onde β = ângulo de deflexão do atuador, w = velocidade linear lateral, q = velocidade angular, θ = ângulo de atitude, θ_c = comando de atitude, r = distúrbio na medida de velocidade, V = perturbação de vento, β_p = desvio residual de empuxo.

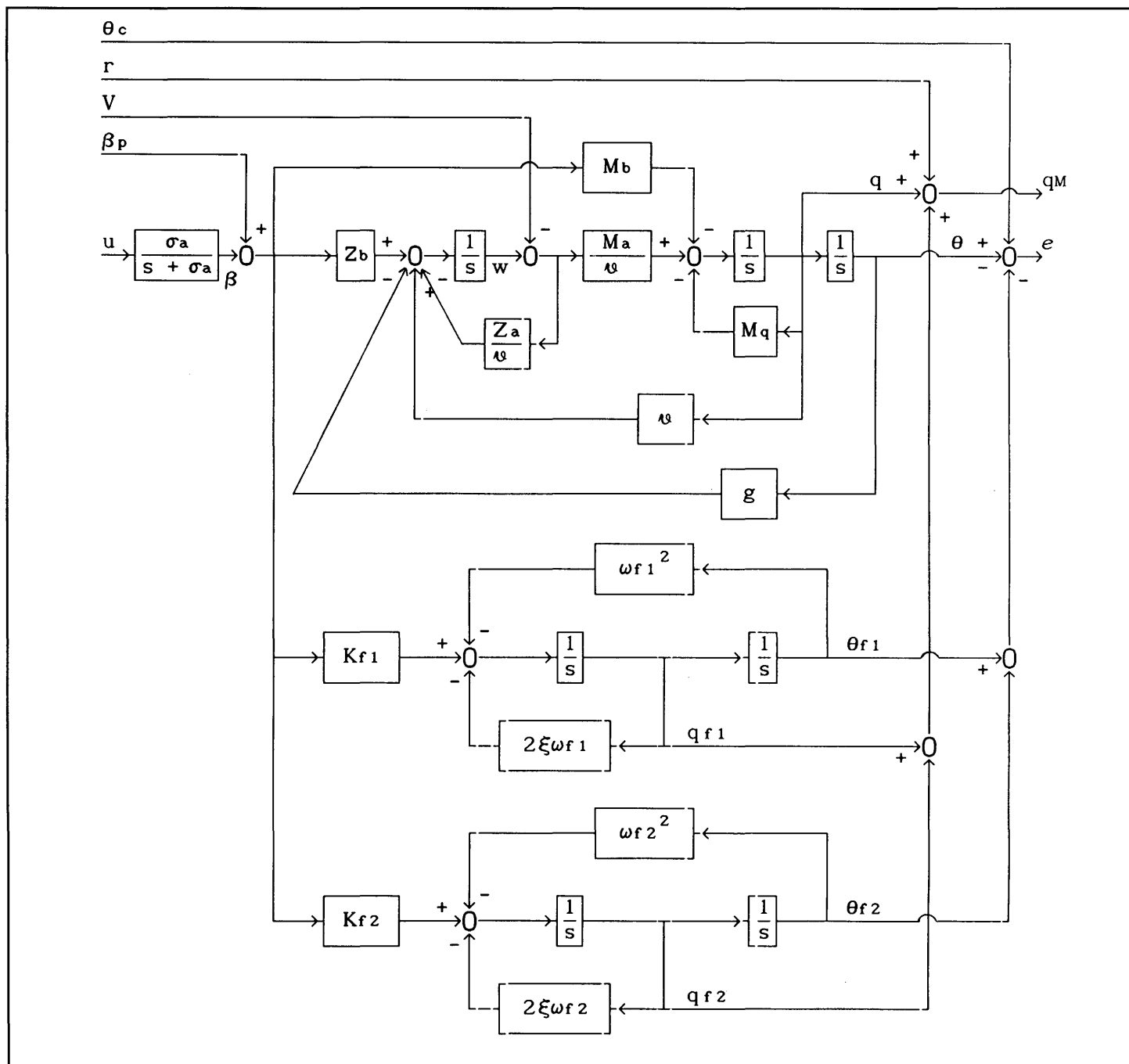


Figura 2 - Dinâmica do foguete, incluindo modos de flexão θ_{f1} e θ_{f2} , no plano de arfagem.

Para a utilização da técnica H^∞ , faz-se necessário reconfigurar o sistema da fig. 2 na forma geral (Francis e Doyle, 1987). Procedendo conforme em (Levine e Reichert, 1989), obtém-se a planta aumentada mostrada na fig. 3. As entradas exógenas são θ_c , r , V e β_p . As saídas reguladas são e , θ e u , que originam, após ponderação por W_1 , W_2 e W_3 , os sinais z_1 , z_2 e z_3 , respectivamente. Os sinais medidos são e e q_M , e o sinal de controle é u .

Utilizando-se os parâmetros dados em (Moreira, 1992), obtém-se o controlador $K(s)$, cuja representação em variáveis de estado é

$$\dot{x}_c(\tau) = A_c x_c(\tau) + B_c \begin{bmatrix} e(\tau) \\ q_M(\tau) \end{bmatrix} \quad (3.1)$$

$$u(\tau) = C_c x_c(\tau) \quad (3.2)$$

$$A_c = \begin{bmatrix} -0.09048 & -0.06957 & 0.10695 & 0.9213 & 0.0485 & 0.1971 \\ -0.03925 & -0.14174 & 0.35966 & 3.1408 & 0.2067 & 0.6904 \\ 0.06702 & 0.35966 & -1.9643 & -20.528 & -7.8751 & -4.3394 \\ 0.43044 & 3.0699 & -14.767 & -161.98 & -130.24 & -40.135 \\ -0.04573 & -0.18321 & 7.7388 & 128.00 & -0.4874 & -24690 \\ 0.11037 & 0.68767 & -4.2570 & -38.596 & 24.281 & -9.8118 \end{bmatrix}$$

(3.3)

$$B_c = \begin{bmatrix} 1.98800 & -1.0558 \\ 0.17815 & -0.9002 \\ -0.26836 & 1.3176 \\ -0.11588 & 11.543 \\ 0.40075 & -0.4415 \\ -0.28212 & 2.4514 \end{bmatrix}$$

$$C_c = [2.250 \quad 0.91762 \quad -1.3447 \quad -11.543 \quad -0.59628 \quad -2.4676]$$

onde

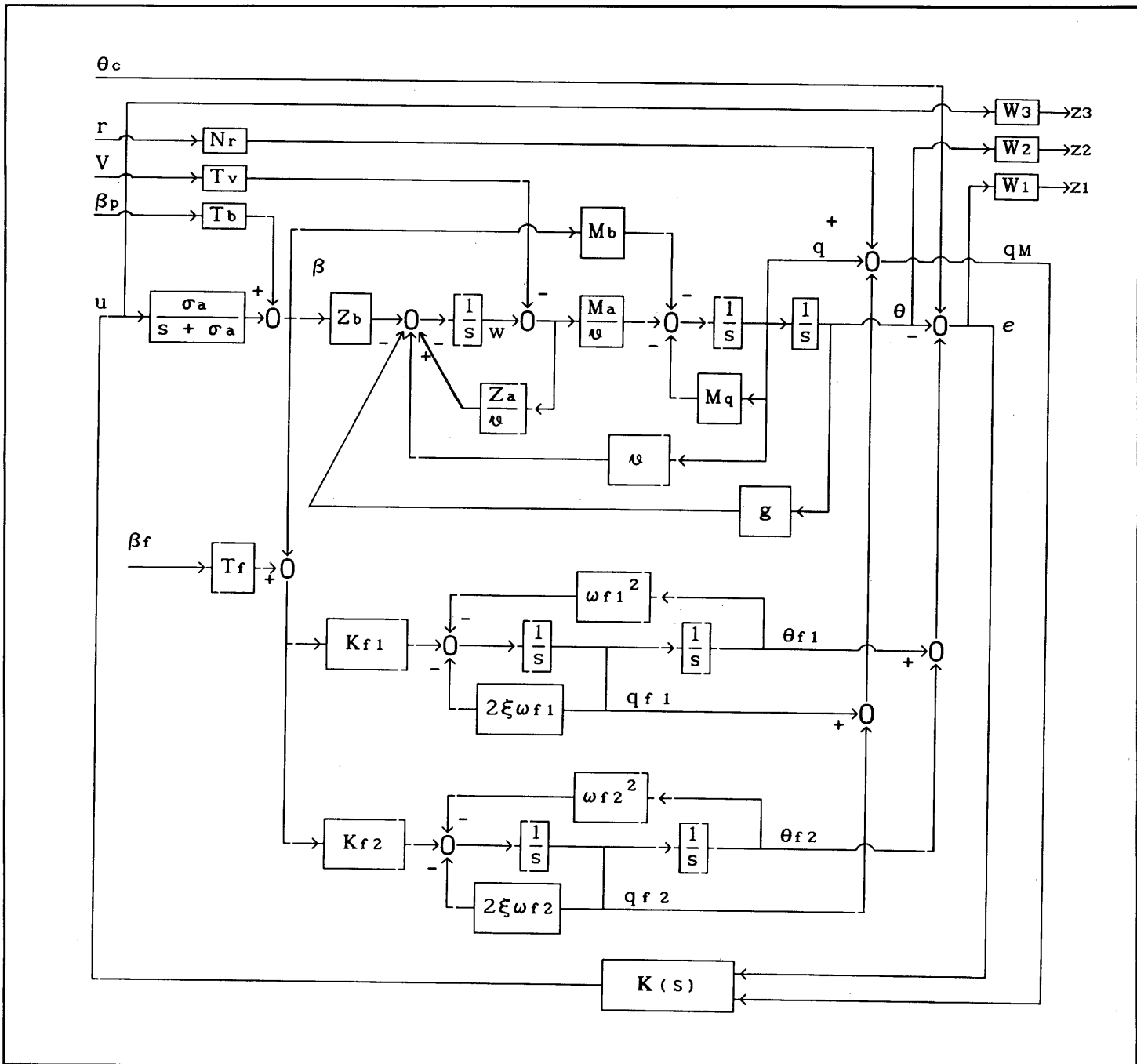


Figura 3 - Planta aumentada para a aplicação da técnica H^∞ .

Para efeito de implementação digital com período de amostragem T suficientemente curto, pode-se integrar (3.1) via método de Euler, resultando a versão discreta

$$x_c(t+1) = (I + TA_c)x_c(t) + TB_c \begin{bmatrix} e(t) \\ q_M(t) \end{bmatrix} \quad (3.4)$$

$$u(t) = C_c x_c(t) \quad (3.5)$$

Obviamente, o cálculo em tempo real de $u(t)$ via (3.4)-(3.5) requer processamento numérico considerável, devido a (3.3), motivando assim a implementação de (3.4)-(3.5) em processador digital de sinal.

4 - CONTROLADOR AUTO-SINTONIZÁVEL TIPO GPC

Um resumo da técnica de controle adaptativo tipo GPC (*Generalized Predictive Control*) é apresentado a seguir. Vide Clarke et. al. (1987) e Clake (1988) para detalhes. Esta técnica tem recebido especial atenção porque é menos sensível do que as técnicas tipo MV (*Minimum Variance*) e GMV (*Generalized Minimum Variance*) à especificação incorreta dos parâmetros de projeto, tais como atraso de transporte e ordem do sistema controlado. Isto porque no cálculo do controle utiliza-se a predição da saída diversos passos à frente, e não apenas K passos, onde K é o atraso de transporte.

Suponha que o processo controlado seja descrito pelo modelo

$$A(q^{-1})y(t+1) = B(q^{-1})u(t) + \frac{w(t+1)}{\Delta(q^{-1})}, \quad (4.1)$$

$$\Delta(q^{-1}) = 1 - q^{-1}$$

onde $A(q^{-1})$ e $B(q^{-1})$ são polinômios dados por

$$A(q^{-1}) = 1 - a_1 q^{-1} - \dots - a_{p_0} q^{-p_0} \quad (4.2)$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{q_0} q^{-q_0+1} \quad (4.3)$$

e $w(t)$ representa um distúrbio de natureza estocástica.

Considerando-se que a estratégia adaptativa tipo GPC baseia-se no princípio da equivalência à certeza (Goodwin e Sin, 1984), de início considera-se o problema de controle supondo-se conhecidos os parâmetros em (4.2)-(4.3), e a seguir substitui-se tais parâmetros por suas estimativas. Dada a referência $y_{ref}(t)$, o problema de controle resume-se em minimizar o custo.

$$J(N_1, N_2, N_u, \lambda) = E \left[\sum_{l=N_1}^{N_2} (y(t+l) - y_{ref}(t+l))^2 + \sum_{l=1}^{N_u} \lambda(l) (\Delta u(t+l-1))^2 \right] \quad (4.4)$$

onde

$$\Delta u(t+l-1) = u(t+l-1) - u(t+l-2) \quad (4.5)$$

e $\lambda(l)$ é uma seqüência de ponderação da intensidade do controle. A minimização de (4.4) requer a predição de $y(t)$ 1 passo à frente (Clarke et. al., 1987). Supondo-se $E[w(t)] = 0$ e $E[w(i)w(j)] = 0, \forall i \neq j$, pode ser mostrado que o preditor ótimo é dado por

$$y^0(t+l|t) = F_l(q^{-1})B(q^{-1})\Delta u(t+l-1) + G_l(q^{-1})y(t) \quad (4.6)$$

onde $F_l(q^{-1})$ e $G_l(q^{-1})$ são obtidos com base na equação polinomial

$$1 = F_l(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-l}G_l(q^{-1}) \quad (4.7)$$

com

$$F_l(q^{-1}) = f_{0,l} + f_{1,l}q^{-1} + \dots + f_{l-1,l}q^{-l+1} \quad (4.8)$$

$$G_l(q^{-1}) = g_{0,l} + g_{1,l}q^{-1} + \dots + g_{p_0,l}q^{-p_0} \quad (4.9)$$

Com base em (4.6)-(4.9) e definindo-se

$$H_l(q^{-1}) = F_l(q^{-1})B(q^{-1}) = h_{0,l} + h_{1,l}q^{-1} + \dots + h_{1+q_0-2,l}q^{-1-q_0+2} \quad (4.10)$$

pode ser mostrado (Clarke et. al., 1987) que a seqüência de controle

$$\Delta U(t, N_u) = [\Delta u(t) \ \Delta u(t+1) \ \dots \ \Delta u(t+N_u-1)]^T \quad (4.11)$$

que minimiza (4.4) é dada por

$$\Delta U(t, N_u) = (H^T(N_1, N_2, N_u) H(N_1, N_2, N_u) + \lambda I_{N_u \times N_u})^{-1} \cdot H^T(N_1, N_2, N_u) \cdot (Y_{ref}(t, N_1, N_2) - S(t, N_1, N_2)) \quad (4.12)$$

onde

$$H(N_1, N_2, N_u) = \begin{bmatrix} h_{N_1-1, N_1} & h_{N_1-2, N_1} & \dots & h_{N_1-N_u, N_1} & \dots \\ \vdots & \vdots & & & \\ h_{N_2-1, N_2} & h_{N_2-2, N_2} & \dots & \dots & h_{N_2-N_u, N_2} \end{bmatrix} \quad (4.13)$$

$$\text{com } h_{N-N_u, N} = 0 \quad \text{para } N < N_u$$

$$Y_{ref}(t, N_1, N_2) = [y_{ref}(t+N_1) \ \dots \ y_{ref}(t+N_2)]^T \quad (4.14)$$

e

$$S(t, N_1, N_2) = \begin{bmatrix} \sum_{i=N_1}^{N_1+q_0-2} \Delta u(t+N_1-1-i) + G_{N_1}(q^{-1})y(t) \\ \vdots \\ \sum_{i=N_1}^{N_2+q_0-2} \Delta u(t+N_1-1-i) + G_{N_1}(q^{-1})y(t) \end{bmatrix} \quad (4.15)$$

Assim, no instante t resolve-se (4.12) para $\Delta u(t)$ e aplica-se o controle

$$u(t) = u(t-1) + \Delta(t) \quad (4.16)$$

ao processo.

A versão adaptativa da lei de controle (4.12) é obtida estimando-se recursivamente os polinômios $A(q^{-1})$ e $B(q^{-1})$ em (4.1), resolvendo-se (4.7)-(4.10) com $A(q^{-1})$ e $B(q^{-1})$ substituídos por suas estimativas $\hat{A}(q^{-1})$ e $\hat{B}(q^{-1})$, e rescrevendo-se (4.12) na forma

$$\Delta U(t, N_u) = \left(\hat{H}^T(N_1, N_2, N_u) \hat{H}(N_1, N_2, N_u) + \lambda I_{N_u \times N_u} \right)^{-1} \cdot \hat{H}^T(N_1, N_2, N_u) \cdot (Y_{ref}(t, N_1, N_2) - \hat{S}(t, N_1, N_2)) \quad (4.17)$$

Além de requerer a estimação dos coeficientes dos polinômios $A(q^{-1})$, $B(q^{-1})$ e a avaliação de vários preditores (vide(4.10)), a determinação do controle em (4.17) requer a inversão, em tempo real, de uma matriz $N_u \times N_u$, onde N_u é uma estimativa do atraso de transporte do processo a controlar. Considerando-se que atrasos de transporte elevados são comuns em processos industriais (por exemplo, em Clarke(1988) apresenta-se uma aplicação com $N_u=10$), conclui-se que o cálculo do controle pode demandar processamento numérico considerável. Isto motiva o emprego de processadores digitais de sinal neste trabalho para implementar a técnica GPC.

Neste trabalho utiliza-se a técnica GPC para controlar o processo térmico PT326 da Feedback, cujo diagrama de blocos é mostrado na fig. 4, onde se mostra também a resposta de malha aberta para degrau de amplitude 2V, estando a escala horizontal em múltiplos do período de amostragem $T=0,15s$. Basicamente, há dois motivos justifi-

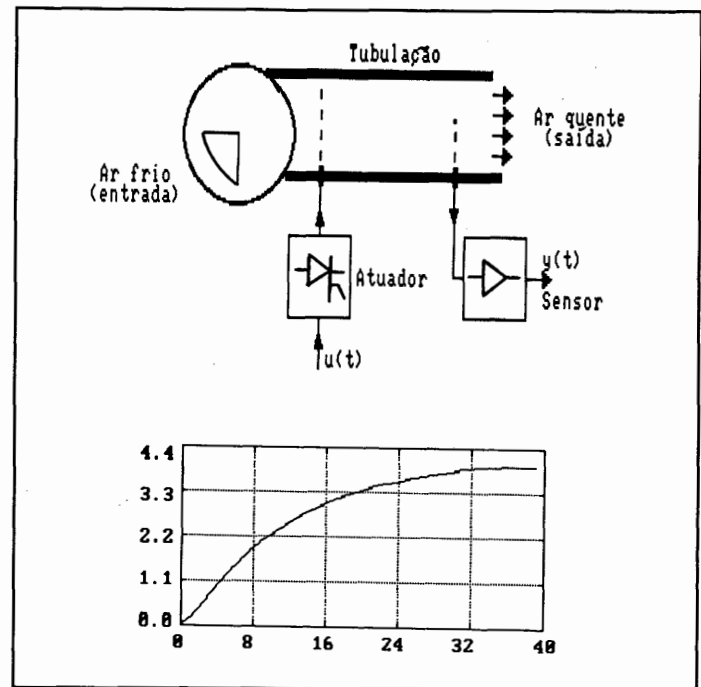


Figura 4 - Diagrama de blocos do processo térmico PT326 e resposta ao degrau $u(t)=2V$.

cando o emprego de controle adaptativo para controlar o PT326:a) O ganho varia com a temperatura ambiental, e b) atrasos de transporte variáveis podem se obtidos deslocando-se a posição do sensor de temperatura.

5 - EXEMPLOS DE APLICAÇÃO

Nesta seção são apresentados detalhes sobre a implementação em DSP dos algoritmos de controle descritos nas seções 3 e 4.

5.1 - Implementação do controlador H^∞

O controlador H^∞ definido por (3.3)-(3.5) foi implementado no DSP2 da fig. 1, que pode se comunicar diretamente com o microcomputador via memória de duplo acesso, não sendo necessário utilizar-se o microcontrolador. A dinâmica do

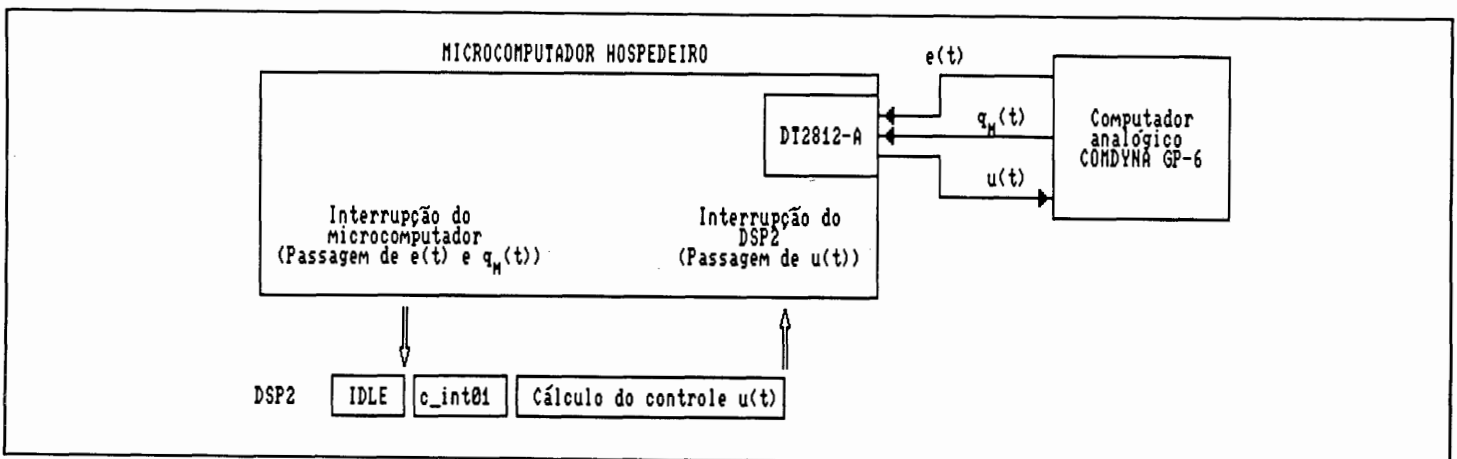


Figura 5 - Implementação do controlador H^∞ para foguete no DSP2.

foguete, caracterizada na fig.2, foi simulada no computador analógico COMDYNA GP-6. A interface com o microcomputador, no caso um PC AT-286, com o computador analógico é efetuada via cartão AD/DA de 12 bits, da Data Translation.

Utilizou-se tempo de amostragem $T=0,005s$ em (3.4). As conexões microcomputador-sistema simulado e microcomputador-DSP2 são mostradas na fig. 5.

O microcomputador gera a base de tempo, lê os valores de $e(t)$ e $q_M(t)$ e escreve tais valores em memória de duplo acesso compartilhada com o DSP2. A seguir o DSP2 é interrompido e atende a interrupção c_int01 , que efetua a leitura destes valores. A seguir o DSP2 calcula o controle $u(t)$ e interrompe o microcomputador, que lê o sinal de controle da memória de duplo acesso e o aplica ao processo simulado. Os valores de $\theta(t)$ e $u(t)$ são também armazenados pelo microcomputador, para geração de gráficos e avaliação do desempenho do sistema de controle.

Na fig. 6 são mostrados os sinais $\theta(t)$ e $u(t)$ para referência $\theta_c=1^\circ$, estando a escala horizontal em múltiplos do período de amostragem $T=0,005s$, logo o tempo total de controle é 4s, e a escala vertical em graus. Constata-se que o desempenho do sistema de controle é satisfatório.

Para determinar o tempo de execução do algoritmo de controle no DSP2, programou-se em *assembly* o temporizador 0 do TMS320C30, que é zerado no início do cálculo e bloqueado no final do mesmo. Para efeito de comparação, o algoritmo de controle foi também implementado em microcomputador AT-286 12MHz, com coprocessador, empregando-se o compilador C da Microsoft.

Os tempos de execução das implementações em DSP e no microcomputador, obtidos experimentalmente, foram:

Tempo de execução para implementação em DSP TMS320C30: 34 μ s

Tempo de execução para implementação em AT-286 12 MHz: 6,9ms

Conclui-se assim que o tempo de execução da implementação em DSP é aproximadamente 202 vezes menor que o tempo da implementação em microcomputador AT-286. Uma vez que se utilizou clock de 32MHz no TMS320C30, que é apenas 2,6 vezes maior que o clock do microcomputador, conclui-se que o desempenho do DSP TMS320C30 se deve principalmente à sua arquitetura e conjunto de instruções especialmente talhados para processamento digital de sinais. Convém também ressaltar que com o microcomputador seria

impossível controlar o foguete com período de amostragem $T=5ms$, pois o cálculo do sinal de controle requer 6,9ms.

De modo a ilustrar a robustez do controlador H^∞ , considera-se agora uma perturbação no ângulo do vetor de empuxo, em forma de degrau de $0,5^\circ$ de amplitude. Esta perturbação excita os modos de flexão, comprometendo o desempenho de controladores convencionais, como o PID por exemplo. Na fig. 7 mostra-se o sinal de saída e o controle para esta perturbação no empuxo, constatando-se que o controlador H^∞ é robusto a tal perturbação.

5.2 - Implementação do controlador auto-sintonizável utilizando 1 DSP

A estratégia de controle descrita na seção 4 foi implementada no DSP2 da fig. 1. Os seguintes parâmetros de projeto foram utilizados: $N_1=1$, $N_2=4$, $N_u=2$, $p_o=2$, $q_o=1$, $\lambda=0,01$, $st=0,15s$. O procedimento para a determinação do controle é mostrado na fig. 8. Basicamente, o microcomputador gera a base de tempo, lê o valor da saída $y(t)$ via conversor AD/DA tipo DT2812-A e interrompe o DSP2, passando o valor de $y(t)$. O DSP2 calcula o controle $u(t)$ e interrompe o microcomputador, que lê o sinal de controle da memória de duplo acesso e o aplica ao processo térmico. Os valores de $y(t)$ e $u(t)$ são armazenados pelo microcomputador, para avaliação do desempenho do sistema de controle.

A fig. 9 mostra uma realização típica da saída $y(t)$ e do sinal de controle $u(t)$, estando a escala vertical em Volts e a horizontal em múltiplos do período de amostragem, que é 0,15s. Logo, o tempo total de controle corresponde a 18s. O sinal de referência é $y_{ref}(t) = 3V, 0 \leq t < 40$; $y_{ref}(t) = 2V, 40 \leq t < 80$ e $y_{ref}(t) = 3V, 80 \leq t \leq 120$. Conclui-se da fig. 9 que o desempenho do sistema de controle é satisfatório. Por economia de espaço, o desempenho da fig. 9 não será comparado com o de outras estratégias auto-sintonizáveis, como a de variância mínima, por exemplo.

O tempo de execução do algoritmo de controle no DSP2 foi determinado de modo similar ao da seção 5.1. O algoritmo de controle foi também implementado no microcomputador AT-286 12MHz. Os tempos de execução das implementações em DSP e no microcomputador, obtidos experimentalmente, foram:

Tempo de execução para implementação em DSP TMS320C30: 363 μ s

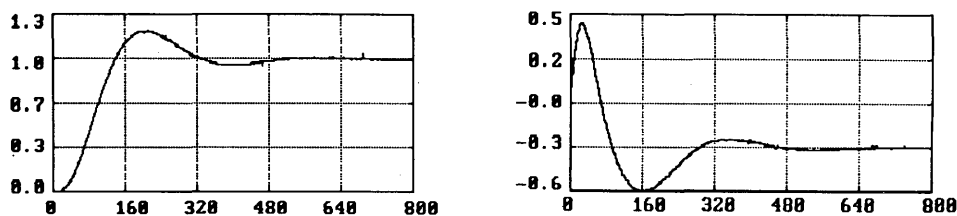


Figura 6- Saída $\theta(t)$ e controle $u(t)$ usando-se controlador H^∞ implementado em DSP.

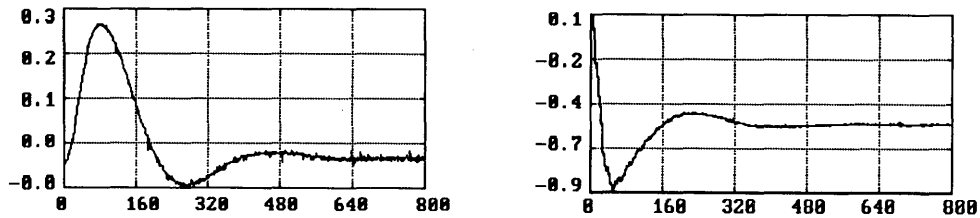


Figura 7 - Saída $\Theta(t)$ e controle $u(t)$ usando-se controlador H^∞ e perturbação no empuxo.

Tempo de execução para implementação em AT-286 12MHz: 18ms

ou seja, o tempo de execução da implementação em DSP é aproximadamente 49 vezes menor que o tempo da implementação em microcomputador AT-286. Convém lembrar que na seção 5.1 o tempo de execução da implementação em DSP foi aproximadamente 202 vezes menor que na implementação em PC AT-286. Esta diferença se deve ao fato de que o implementado na seção 5.1, equações (3.3)-(3.5), envolve basicamente produto interno e esta operação é efetuada de modo extremamente eficiente pelo DSP TMS320C30, utilizando por exemplo registro internos e instrução RPTS(Repeat Single). Vide (Texas Instruments, 1990a) para detalhes.

5.3 - Implementação do controlador auto-sintonizável utilizando 2 DSP's

Nesta seção utiliza-se os dois DSP's da arquitetura da fig.1 para implementar o controlador auto-sintonizável da seção 4. Um escalonamento natural de tarefas consiste em se separar as etapas de estimação de parâmetros e cálculo de controle, conforme sugerido pelo princípio da equivalência à certeza. Basicamente, o DSP1 se encarrega da estimação do vetor de parâmetros $\hat{\theta}(t)$, contendo as estimativas dos coeficientes dos polinômios $A(q^{-1})$ e $B(q^{-1})$, e o DSP2 efetua o cálculo do sinal de controle $u(t)$, compreendendo a geração dos vários preditores e a inversão de matriz. Este escalonamento é mostrado na fig. 10.

Para os parâmetros de projeto utilizados aqui e explicitados no início da seção 5.2, o cálculo efetuado pelo DSP2 é mais intensivo. Verificou-se experimentalmente, utilizando-se o temporizador 0 do DSP2, que o cálculo do sinal de controle $u(t)$ requer $245\mu s$. Resumindo, temos os seguintes tempos de execução:

Tempo de execução para implementação serial em 1 DSP TMS320C30: $363\mu s$

Tempo de execução para implementação paralela em 2 DSP's TMS320C30: $245\mu s$

O comportamento do sistema de controle com a implementação paralela é similar àquele mostrado na fig.9, conforme esperado, e portanto será omitido.

Deve ser ressaltado que o *speed up* obtido com a implementação paralela obviamente depende do grau de paralelismo inerente ao algoritmo de controle utilizado e do escalonamento eficiente das tarefas. Estratégias ótimas de escalonamento são em geral complexas, motivo pelo qual usualmente o escalonamento de tarefas é efetuada heurísticamente com base na análise minuciosa do fluxo de informação no algoritmo de controle de interesse. Para detalhes, vide por exemplo (Luh e Lin, 1982; Hwang e Briggs, 1985).

6 - CONCLUSÕES

Processadores digitais de sinal são utilizados neste trabalho para implementar algoritmos de controle em tempo real. A utilização de DSP's viabiliza a implementação de estratégias sofisticadas de controle, a baixo custo. Dois exemplos

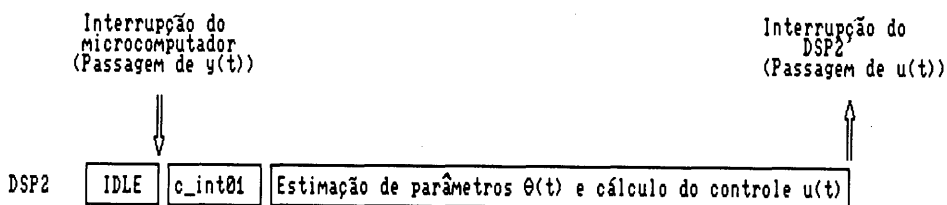


Figura 8 - Implementação serial do controlador auto-sintonizável no DSP2.

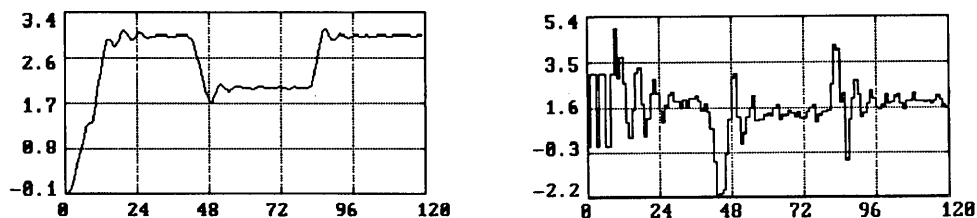


Figura 9 - Realização típica da saída $y(t)$ e sinal de controle $u(t)$ usando-se controlador auto-sintonizável tipo GPC implementado em DSP.

realistas de aplicação, relativos ao controle H^∞ de um foguete e controle auto-sintonizável de um processo térmico, empregado DSP de terceira geração TSM320C30 da Texas Instruments são apresentados e discutidos. Apresenta-se também um exemplo de implementação paralela da estratégia auto-sintonizável tipo GPC, utilizando-se dois DSP's.

As aplicações descritas neste trabalho sugerem que a elevada capacidade de processamento numérico dos DSP's de terceira geração, aliada ao seu baixo custo, viabiliza o emprego destes em três principais situações: a) altas taxas de amostragens são necessárias, como por exemplo em controle de sistemas eletromecânicos; b) a estratégia de controle é sofisticada, como por exemplo em controle adaptativo de manipuladores robóticos, e c) múltiplas malhas têm que ser controladas simultaneamente. Em aplicações mais complexas ou em situações nas quais o tempo de processamento é crítico, pode-se utilizar arquiteturas multiprocessadoras. Neste trabalho ilustrou-se a utilização de uma arquitetura com dois DSP's, que objetiva reduzir o tempo de processamento explorando o eventual paralelismo dos algoritmos de controle de interesse.

AGRADECIMENTO:

O autor agradece à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo auxílio financeiro, via Processo n°91/1009-0 e ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) via processo n°

500108/92-7.

REFERÊNCIAS

- Battilotti, S. and Ulivi, G., 1990, An Architecture for High Performance Control Using Digital Signal Processor Chips, IEEE Control Sys. Mag., Vol.10, N°7, pp. 20-23.
- Clarke, D.W.; Mohtadi, C. and Tuffs, P.S., 1987, Generalized predictive Control. Part 1: The Basic Algorithm, Automatica, Vol.23, N° 2, pp. 137-148.
- Clarke, D.W., 1988, Applications of Generalized Predictive Control to Industrial Processes, IEEE Control Syst. Mag., Vol.8, N°2, pp. 49-55.
- Francis, B. A., 1987, A Course in H^∞ Control Theory, Springer-Verlag, Berlin.
- Francis, B. A. and Doyle, J. C., 1987, Linear Control Theory in H^∞ Optimality Criterion, SIAM Journal on Control and Optimization, Vol. 25, N°4, pp.815-844.
- Glover, K. and Doyle, J. C., 1988, State-Space Formulae for All Stabilizing Controllers that Satisfy an H^∞ - Norm and Relations to Risk Sensitivity, Syst. and Contr. Letters, Vol.11, pp. 167-172.

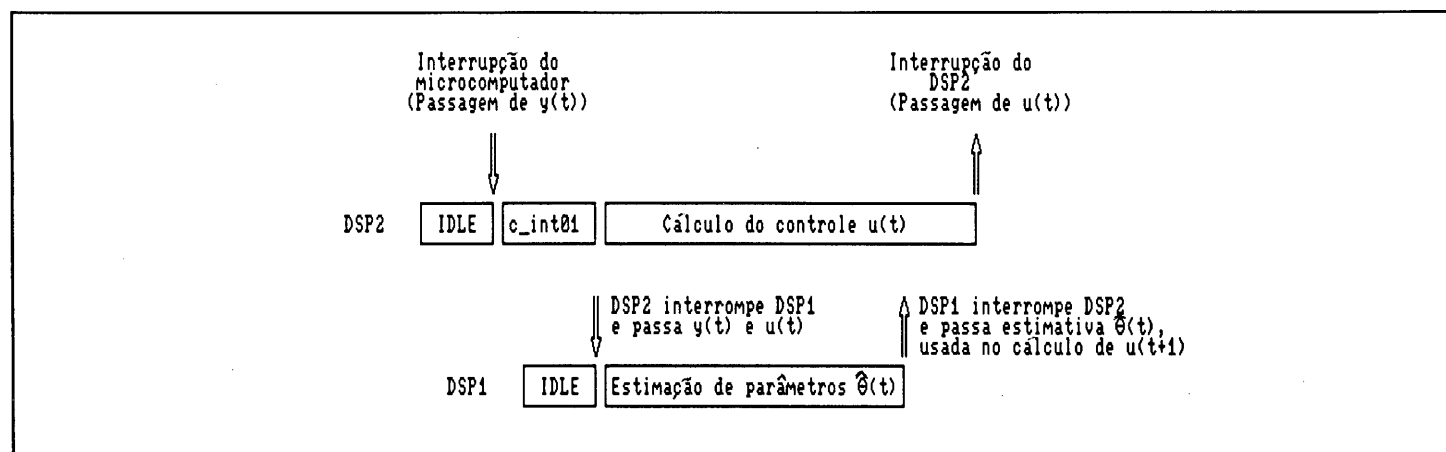


Figura 10 - Implementação paralela do controlador auto-sintonizável usando dois DSP's.

- Goodwin, G.C. and Sin, K.S., 1984, Adaptive Filtering, Prediction and Control, Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Gurubasavaraj, K.H., 1989, Implementation of a Self-Tuning Controller Using Digital Signal Processor Chips, IEEE Control Syst. Mag., Vol.9, N°4, pp.38-42.
- Hwang, K. and Briggs, F. A., 1985, Computer Architecture and Parallel Processing, McGraw-Hill, New York.
- Lee, E.A., 1988, Programmable DSP Architectures: Part I, IEEE Acoust. Speech Sig. Proc. Mag., Vol. 5, No. 4, pp. 4-19.
- Levine, W.S. and Reichert, R.T., 1990, An Introduction to H^∞ Control System Design, Proc.of the 29th Conference on Decision and Control, Honolulu, Hawaii, pp. 2966-2974.
- Luh, J.Y.S. and Lin, C.S., 1982, Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator, IEEE Trans. on Syst. Man and Cybernetics, Vol. 12, N°2.
- Moreira, F.J.O., 1992, Projeto de Autopiloto Robusto para Foguetes Utilizando a Teoria H^∞ , ITA, São José dos Campos, São Paulo.
- Moreira, F.J.O. e Hemerly, E.M., 1992, Projeto de um Autopiloto Robusto para um Foguete usando a Técnica H^∞ , 9° Congresso Brasileiro de Automática, Vitória, pp. 995-1000.
- Philips, 1989, Single-Chip 8-Bit Microcontroller PCB83C552/562 - User's Manual, Eindhoven.
- Radivojevic, I.; Herath, J. and Gray, W. S., 1991, High-Performance DSP Architectures for Intelligence and Control Applications, IEEE Control Syst.Mag., Vol. 11, N°4, pp. 49-55.
- Santos, W. A. and Hemerly, E. M., 1992, Implementation and Performance Evaluation of DSP Based Architectures for Filtering and Control Applications, Int. Conf. on Systems Science XI, Wroclaw, Poland.
- Santos, W. A. e Hemerly, E. M., 1992, Implementação Paralela do Filtro de Kalman para Aplicações em Tempo Real, IV Simpósio Brasileiro de Arquitetura de Computadores-Processamento de Alto Desempenho, São Paulo.
- Tan, J. and Kiriakopoulos, N., 1988, Implementation of a Tracking Kalman Filter on a Digital Signal Processor, IEEE Trans. Ind. Elect., Vol. 35, N°1, pp. 126-134.
- Texas Instruments, 1990b, TMS320C30 Optimizing C Compiler Reference Guide, Houston.
- Texas Instruments, 1990a, Third Generation TMS320 - User's Manual, Houston.
- Yeh, H. G., 1991, Processing Performance of Two Kalman Filter Algorithms with a DSP32C by Using Assembly and C Languages, IEEE Trans. Ind.Elect., Vol. 38, N°4, pp. 298-302.