

ASSOCIAÇÃO ESCALÁVEL DE DSP's PARA PROCESSAMENTO DE UMA EQUAÇÃO DE ESTADO MULTIVARIÁVEL

FABRIZIO LEONARDI, MARCO A. A. DE MELO, GEORGE A. B. DE ARAÚJO

Faculdade de Engenharia Industrial FEI-FCA
Depto. de Eletricidade- Av. Humberto A. C. Branco, 3972
09850-901 – São Bernardo do Campo, SP, BRAZIL

E-mails: fabrizio@cci.fei.br, mant@cci.fei.br, georaugu@cci.fei.br

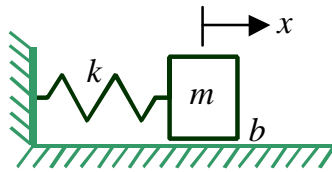
Resumo - Este trabalho discute sobre a síntese de sistemas dinâmicos através de equações de estado quando o número de entradas e saídas do meio de implementação é limitado. A solução empregada foi o particionamento em várias equações, mantendo a função entre as entradas e saídas, caracterizando assim uma implementação distribuída. Como resultado temos uma estrutura modular (escalável) para um número arbitrário de entradas e saídas. Como ilustração é apresentado o estudo de um caso de síntese de uma equação de estado linear de tempo discreto descentralizada em dois microprocessadores que juntos realizam o controlador multivariável de um pêndulo invertido. Aspectos pertinentes a implementação digital, como influência do tempo de processamento, são também abordados.

Abstract - This work deals with the synthesis of dynamic systems through state space equations when the number of inputs and outputs of the hardware are limited. The solution applied splits the equations, while keeps the whole transfer function, resulting in a distributed implementation. The result has a modular structure for an arbitrary number of inputs and outputs. A synthesis of an linear equation in two microprocessors that together are a multivariable controller of an inverted pendulum is used as an illustration. Others matters on the digital implementation, as influence of the time of processing, are also included.

Keywords – Multivariable Control; Digital Signal Processing; Distributed Control, Real Time Control

1. Introdução e Motivação.

O comportamento dinâmico de um sistema ou, mais precisamente, do modelo desse sistema, é descrito por equações diferenciais de variáveis reais, funções do tempo. Suponha um conjunto massa-mola-atrito a seguir:



Sistema massa-mola-atrito

O modelo do sistema pode ser representado de várias formas equivalentes, conforme ilustram as Eq. (i), (ii) e (iii):

$$m\ddot{x}(t) = -b\dot{x}(t) - kx(t) + u(t) \quad (i)$$

$$m\dot{v}(t) = -bv(t) - k \int v(t) dt + u(t) \quad (ii)$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) \quad (iii)$$

Nos casos (i) e (ii), o conhecimento do valor da respectiva variável dependente (x para (i) e v para (ii)), num determinado instante, não é suficiente para determinar, só com as informações daquele instante, os valores das demais grandezas. Com o

modelo no formato de espaço de estado (iii), o conhecimento numérico das variáveis de estado $x(t)$ e $v(t)$ num determinado instante arbitrário, permite determinar todas as demais variáveis internas do sistema. Além disso, como esta equação dá o conhecimento da derivada das variáveis, podemos antever o valor delas num instante futuro

Por esses motivos e por mais uma série de propriedades, é conveniente adotar a representação de estados não só para sua análise, mas também para sua síntese. As boas propriedades das equações de estado na síntese de controladores digitais é estudada por (Astrom, 1997) e vários problemas inerentes a síntese digital são também analisadas. Uma questão de ordem prática porém é deixada de lado: a limitação do número de entradas e saídas do *hardware* utilizado. É neste sentido que este trabalho apresenta sua contribuição.

A principal motivação deste trabalho surgiu quando no curso de Controle Multivariável da graduação incluímos no conteúdo do laboratório a síntese de controladores com os DSP's (*Digital Signal Processor*) disponíveis na instituição: *kits* de desenvolvimento dotados de 2 (duas) entradas analógicas e 1 (uma) saída analógica. O problema passou a ser então como realizar sistemas de um número arbitrário de entradas e saídas sem a necessidade de realizarmos um projeto de *hardware* ou a aquisição de novos *kits*. Não é comum se transmitir os dados de forma analógica entre DSPs, pois, além de diminuir a imunidade a ruídos do sistema, introduz-se erros de quantização e atrasos. Contudo, almejou-se que o resultado fizesse uso apenas da teoria de sistemas, por questões didáticas

do curso. A solução encontrada foi realizar uma associação modular dos DSP's, que é descrita em detalhes no item (2) deste trabalho. No item (3) descrevemos brevemente o DSP que será usado para a síntese da equação de estado. Para ilustrar a validade do método, incluímos no item (4) o estudo de um caso de controle de um pêndulo invertido cujo controlador resultou com 3 (três) entradas e 1 (uma) saída.

2. Equação de Estado Particionada.

Uma vez que pretende-se sintetizar as equações de estado digitalmente, todo o desenvolvimento teórico foi feito no domínio do tempo discreto. Para simplificar a notação, uma equação de estado será, quando necessário, representada abreviadamente pelas 4 matrizes na notação da Eq. (1).

$$\begin{aligned} x(k+1) &= A \cdot x(k) + B \cdot u(k) \\ y(k) &= C \cdot x(k) + D \cdot u(k) \end{aligned} \Leftrightarrow \begin{pmatrix} [A] & [B] \\ [C] & [D] \end{pmatrix} \quad (1)$$

Considere a equação de estado (Eq. 2) e a de saída (Eq. 3) com B , C e D na forma particionada.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = [A] \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + [B_1 \ B_2 \ \dots \ B_p] \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_p(k) \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_r(k) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_r \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & \dots & D_{1p} \\ D_{21} & D_{22} & \dots & D_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ D_{r1} & D_{r2} & \dots & D_{rp} \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_p(k) \end{bmatrix} \quad (3)$$

Este sistema possui uma matriz função de transferência F associada dada pela Eq. (4).

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \\ \vdots \\ Y_r(z) \end{bmatrix} = \begin{bmatrix} F_{11}(z) & F_{12}(z) & \dots & F_{1p}(z) \\ F_{21}(z) & F_{22}(z) & \dots & F_{2p}(z) \\ \vdots & \vdots & \ddots & \vdots \\ F_{r1}(z) & F_{r2}(z) & \dots & F_{rp}(z) \end{bmatrix} \cdot \begin{bmatrix} U_1(z) \\ U_2(z) \\ \vdots \\ U_p(z) \end{bmatrix} \quad (4)$$

O sistema de dimensão $p \times r$ pode ser fatorado (particionado) em equações de estado de forma que a matriz função de transferência global é idêntica a da equação original, pois cada saída pode ser escrita como mostra a Eq. (5).

$$\begin{aligned} Y_i &= [F_{i1} \ \dots \ F_{ij}] \cdot \begin{bmatrix} U_1 \\ \vdots \\ U_j \end{bmatrix} + [F_{i,j+1} \ \dots \ F_{ik}] \cdot \begin{bmatrix} U_{j+1} \\ \vdots \\ U_k \end{bmatrix} + \dots + \\ &+ \dots + [F_{iq} \ \dots \ F_{ip}] \cdot \begin{bmatrix} U_q \\ \vdots \\ U_p \end{bmatrix} \end{aligned} \quad (5)$$

Uma maneira de ser realizar tal particionamento é usando a mesma matriz dinâmica em todas as equações de estado (Eq. 6). Embora desta maneira o estado não seja propriamente fatorado, a realização é particionada no sentido de cada equação de estado é feita com um número menor de entradas e saídas, conforme motivação inicial.

$$\begin{aligned} &\left(\begin{bmatrix} [A] & [B_1] & \dots & [B_j] \\ [C_i] & [D_{i1}] & \dots & [D_{ij}] \end{bmatrix} \right), \left(\begin{bmatrix} [A] & [B_{j+1}] & \dots & [B_k] \\ [C_i] & [D_{i,j+1}] & \dots & [D_{ik}] \end{bmatrix} \right), \dots \\ &\dots, \left(\begin{bmatrix} [A] & [B_q] & \dots & [B_p] \\ [C_i] & [D_{iq}] & \dots & [D_{ip}] \end{bmatrix} \right) \end{aligned} \quad (6)$$

No particionamento, as saídas de uma equação de estado são entradas de outra. As Eq. (8) e (9) ilustram com isso pode ser feito, através de equações de estado de 2 entradas e 1 saída, para um sistema de 3 entradas e 1 saída (Eq. 7), exatamente como é utilizado no exemplo numérico da seção 4.

$$Y(z) = [F_1(z) \ F_2(z) \ F_3(z)] \cdot \begin{bmatrix} U_1(z) \\ U_2(z) \\ U_3(z) \end{bmatrix} \quad (7)$$

$$Y_1(z) = [F_1(z) \ F_2(z)] \cdot \begin{bmatrix} U_1(z) \\ U_2(z) \end{bmatrix} \quad (8)$$

Portanto,

$$Y(z) = [1 \ F_3(z)] \cdot \begin{bmatrix} Y_1(z) \\ U_3(z) \end{bmatrix} \quad (9)$$

2.1 Aspectos Práticos da Síntese.

Normalmente o tempo de amostragem de uma equação de tempo discreto aparece como coeficientes nos próprios elementos das matrizes A , B , C e D . Desta maneira, é necessário conhecer o tempo de execução da equação, antes de sua implementação. A forma mais comum para se resolver tal problema é a criação de um esquema de interrupção cujo período é superior ao máximo tempo de processamento da equação. Os dados terminam de serem processados antes do período de amostragem e são enviados ao mundo externo imediatamente ou no instante da interrupção. A estratégia de aguardar até o evento da interrupção normalmente é preferido uma vez que acarretará um atraso mais fácil de ser tratado matematicamente. Essa adoção implica, por causa do tempo de processamento, em multiplicar por z^{-1} as funções de transferência.

Especialmente para sistemas de controle, é importante considerar os atrasos de fase no sinal introduzidos. Neste trabalho, limita-se a estudar o problema do atraso computacional. Ao associar DSP's para a síntese da uma equação, os atrasos computacionais ocorrerão em cada um deles. Para simplificar a análise da influência que estes atrasos geram, buscou-se uma forma de alterar as equações de tal forma que os atrasos computacionais continuassem a serem vistos como simples atrasos z^{-m} na função de transferência global. Assim, alguns dos sinais que chegam a um DSP são forçosamente atrasados de um número de períodos de amostragem, dentro da própria equação de estado daquele DSP, que é modificada para tanto.

Outro cuidado relacionado com a implementação é a inicialização do sistema. Uma vez que a equação de estado é implementada de forma distribuída, o vetor inicial de estados deve ser carregado em cada DSP de acordo com o particionamento.

Na ilustração que veremos adiante, o módulo utilizado possui 2 entradas e 1 saída. Com ele, um sistema de dimensão pxr poder ser implementado com r sistemas $px1$, onde cada um desses pode ser implementado como ilustra o diagrama da Fig. 1.

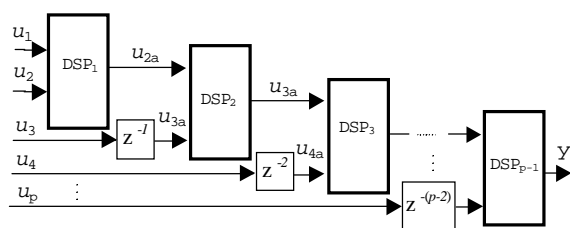


Figura 1. Síntese de um sistema $px1$

São necessários portanto, para este tipo de restrição às entradas e saídas, $(p-1) \cdot r$ processadores para a síntese da equação original.

3. O DSP TMS320C31 e sua organização

O sistema de processamento de uma equação de estado multivariável, aqui proposto, utiliza em cada módulo de processamento um TMS320C31 DSK (DSP Starter Kit) o qual é formado por um conjunto de subsistemas composto de: (i) um processador digital de sinais de propósito geral de 32 bits com unidades aritméticas em ponto flutuante (DSP TMS320C31); (ii) memória interna local de 2 KWords para armazenamento de programas e dados, e (iii) um circuito de interface analógica (AIC) constituído de um conversor analógico digital e um conversor digital analógico.

O TMS320C31 utilizado opera com ciclo de instrução de 40 ns e realiza 50 milhões de operações em ponto flutuante por segundo (MFLOPS) ou 25 milhões de instruções por segundo MIPS (desconsiderando-se aspectos como "wait-states"). Isto deve-se ao fato do processador dispor de duas unidades em ponto flutuante, sendo uma para soma e a outra para multiplicação, que podem operar em paralelo (Chassaing, 1999) e (Sorensen, 1997).

O processador possui um barramento de endereçamento 24 bits, o que possibilita acessar 16 MWords de programas, dados ou I/O, sendo que a comunicação do processador com o AIC, é feito pelo canal serial de comunicação existente no DSP. No sistema proposto, utiliza-se a saída de um conversor digital analógico para conectar a entrada de um outro conversor analógico digital.

O programa fonte é escrito em linguagem de montagem e depois convertido em linguagem de máquina por um montador (DSK3A).

O código de máquina é carregado nos processadores com o DSK3D. O programa é armazenado na memória de cada módulo de processamento para execução, em tempo real, da equação de estado (Eq. 1) e para controle do periférico AIC na leitura das duas entradas analógicas u_1 e u_2 e daí fornecer uma saída analógica y_1 . Observe que a comunicação de dados se dá pela transmissão de y_n (DSP $_n$) para a entrada u_1 do módulo de processamento seguinte (DSP $_{n+1}$).

Basicamente o programa, que é igual em cada um dos módulos de processamento, realiza as seguintes tarefas descritas no fluxograma da Fig. 2.

A organização dos módulos de processamento utilizada para solucionar o problema inicial da quantidade de dados de entrada e saída limitada em cada módulo de processamento, foi elaborada de maneira a se conseguir uma solução para qualquer número de entradas e saídas de um dado sistema, onde através da escalabilidade (Hwang, 1993) do número de processadores se pode aumentar linearmente o número de entradas para o sistema.

Em um modelo de computador paralelo escalar, para manter a eficiência de computação a um dado problema, tem-se um aumento do tamanho da máquina (n) proporcional ao aumento do tamanho do problema (s), onde n é o número de processadores e s é o número de entradas do nosso sistema. Com base no estudo da escalabilidade, que determina o grau de casamento entre um computador e uma aplicação algorítmica, chegamos, fundamentado no fato de que cada módulo de processamento tem duas entradas e uma saída, a configuração sugerida, mostrada na Fig. 1, para resolver o problema inicial.

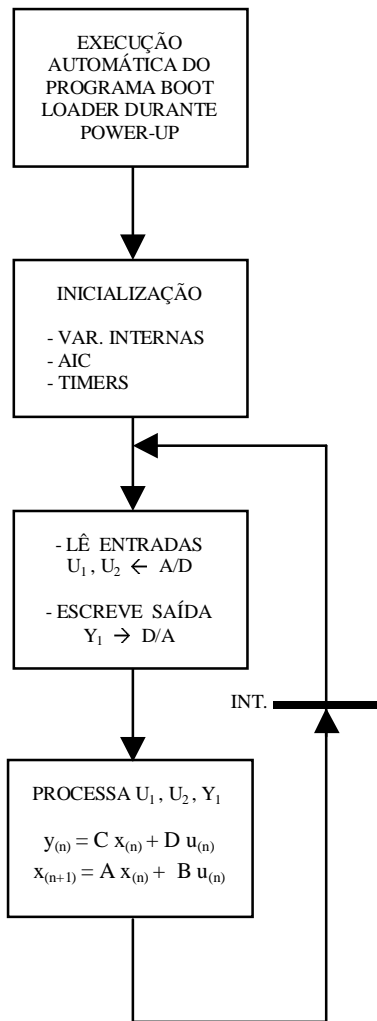


Figura 2. Fluxograma

4. Ilustração Numérica.

Um caso clássico no projeto controladores é o pêndulo invertido. Este sistema possui duas variáveis de controle (ângulo em relação a vertical e a posição do carrinho). Uma das maneiras para simplificar a estratégia de controle é criar uma variável de controle sobre a qual recaem os dois objetivos de controle, que são manter o ângulo nulo e o carro numa posição x arbitrária. Isso é conseguido com o controle da posição y da cabeça do pêndulo, com a restrição de que a solução seja estável. Isto implica num x fixo e no ângulo nulo. O controlador deste sistema tem três entradas: $y_{desejado}$, x_{medido} , θ_{medido} e uma saída: v .

Definindo,

$$u_1(k) \hat{=} \theta_{medido}$$

$$u_2(k) \hat{=} x_{medido}$$

$$u_3(k) \hat{=} y_{desejado}$$

$$e$$

$$y(k) \hat{=} v$$

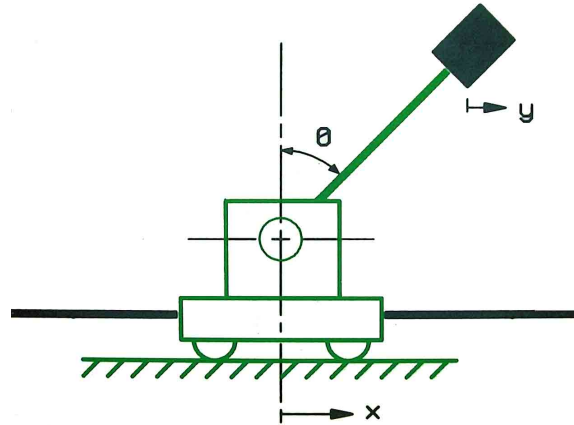


Figura 3. Pêndulo invertido

A equação de estado de um compensador capaz de controlar o pêndulo invertido da Bytronic International Ltd. (Bytronic) é dado pela equação a seguir, discretizado numa taxa de amostragem de 500 Hz.

$$x(k+1) = A \cdot x(k) + [B1 \quad B2 \quad B3] \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

$$y(k) = C \cdot x(k) + [D1 \quad D2 \quad D3] \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

numericamente,

$$G = \left(\begin{bmatrix} 1.0149 & -0.0149 \\ 0.0693 & 0.9307 \end{bmatrix} \begin{bmatrix} -0.0274 & -0.0740 & 0.0274 \\ 0.0419 & 0.1131 & -0.0419 \end{bmatrix} \right)$$

$$\left[\begin{matrix} 1.7723 & -1.7925 \\ 3.7000 & 9.9900 & -3.7000 \end{matrix} \right]$$

A síntese será feita por 2 DSP's de 2 entradas e 1 saída cada, como ilustra a Fig. 4, a seguir:

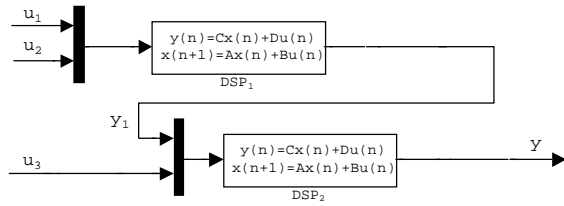


Figura 4. Síntese com dois DSP's

Este caso é simples e pode dar margem a encobrir as potencialidades da solução e induzir o leitor a crer que um dos DSP's faz a função apenas de somar as contribuições. Isto ficaria mais evidente se a equação de estado não fosse de ordem tão reduzida. De fato, cada DSP sintetiza uma equação dinâmica de ordem n .

As equações de estado de cada DSP, sem levar em consideração o atraso computacional, resultam como se segue:

$$DSP_1 = \begin{pmatrix} \begin{bmatrix} 1.0149 & -0.0149 \\ 0.0693 & 0.9307 \end{bmatrix} & \begin{bmatrix} -0.0274 & -0.0740 \\ 0.0419 & 0.1131 \end{bmatrix} \\ \begin{bmatrix} 1.7723 & -1.7925 \end{bmatrix} & \begin{bmatrix} 3.7000 & 9.9900 \end{bmatrix} \end{pmatrix}$$

$$w(k+1) = A \cdot w(k) + \begin{bmatrix} B1 & B2 \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

$$y_1(k) = C \cdot w(k) + \begin{bmatrix} D1 & D2 \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

A função de transferência implementada pelo DSP₁ é dada por:

$$Y_1(z) = F_1(z) \cdot U_1(z) + F_2(z) \cdot U_2(z)$$

Para o segundo DSP temos,

$$DSP_2 = \begin{pmatrix} \begin{bmatrix} 1.0149 & -0.0149 \\ 0.0693 & 0.9307 \end{bmatrix} & \begin{bmatrix} 0 & 0.0274 \\ 0 & -0.0419 \end{bmatrix} \\ \begin{bmatrix} 1.7723 & -1.7925 \end{bmatrix} & \begin{bmatrix} 1 & -3.7 \end{bmatrix} \end{pmatrix}$$

$$z(k+1) = A \cdot z(k) + \begin{bmatrix} 0 & B3 \end{bmatrix} \cdot \begin{bmatrix} y_1(k) \\ u_3(k) \end{bmatrix}$$

$$y(k) = C \cdot z(k) + \begin{bmatrix} 1 & D3 \end{bmatrix} \cdot \begin{bmatrix} y_1(k) \\ u_3(k) \end{bmatrix}$$

e a expressão da saída global em relação às entradas resulta:

$$Y(z) = 1 \cdot Y_1(z) + F_3(z) \cdot U_3(z) = \begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

Com a adoção de uma taxa de amostragem de 500 Hz e da estratégia de ler/enviar os dados para o mundo externo somente no instante de interrupção, implicam em atrasos computacionais individuais de 0,002s.

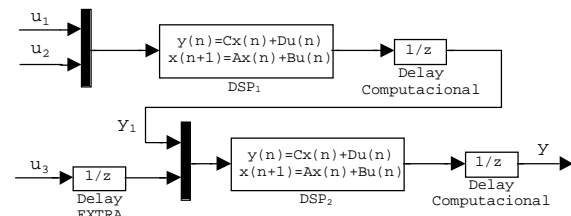


Figura 5. Atrasos normalizados

Como o DSP₁ contribui com 1 atraso computacional, então para que o DSP₂ tenha suas entradas igualmente atrasadas de 1 período de amostragem, sua equação deve ser alterada para incorporar tal atraso. O resultado é que as funções de transferência são as mesmas associadas à G , multiplicadas por z^{-2} .

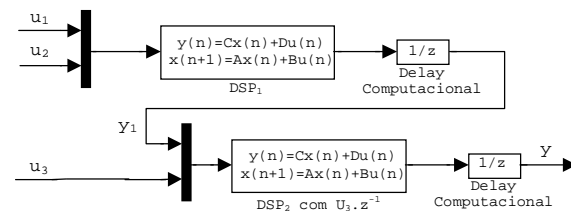


Figura 6. Síntese do atraso pelo DSP₂

A nova equação de estado do DSP₂ que incorpora o atraso somente na segunda entrada resulta:

$$DSP_2 = \begin{pmatrix} \begin{bmatrix} 1.0149 & -0.0149 & 0.0274 \\ 0.0693 & 0.9307 & -0.0419 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1.7723 & -1.7925 & -3.7000 \end{bmatrix} & \begin{bmatrix} 1 & 0 \end{bmatrix} \end{pmatrix}$$

Uma análise simples da equação original permite avaliar se o resultado da associação escalável de DSP's será efetiva. Neste caso basta analisarmos se o sistema de controle suporta o atraso de fase de z^{-2} (0,004s).

Neste exemplo, os atrasos computacionais são insignificantes face à dinâmica do processo a ser controlado. Assim, a implementação distribuída será equivalente à implementação ilustrada a seguir:

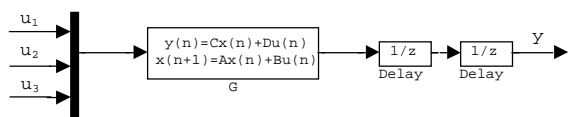


Figura 7. Implementação equivalente

Simulações confirmaram que a solução com dois DSP's é equivalente à implementação numa única equação.

Funções que ocorrem em instantes diferentes foram aplicadas simultaneamente às entradas dos dois sistemas. Restaria analisar as conseqüências da implementação digital. Como o processador a ser utilizado possui uma ULA que opera em ponto flutuante e com uma quantidade de bits elevada, espera-se que a simulação reflita a síntese com os DSP's, se forem tomados os cuidados necessários com o escalonamento de variáveis, os quais, para o caso da ilustração, podem ser feitos através de transformações lineares.

5. Comentários e Conclusões

Este trabalho discutiu sobre a associação de DSP's com um número limitado de entradas e saídas para a síntese de uma equação de estado com um número arbitrário de entradas e saídas. A solução sugerida foi o uso escalável de DSP's disponíveis, configurando um implementação descentralizada.

Para ilustrar didaticamente as possibilidades, mostramos como o controlador multivariável de um

pêndulo invertido dotado de 3 entradas e 1 saída pode ser implementado com 2 DSP's de 2 entradas e 1 saída.

6. Referências

- Astrom, K. J. and Wittenmark, B. (1997). Computer Controlled System: Theory and Design, *3rd Ed, Prentice Hall*.
- Bytronic International Ltd. Documentation for the Pendulum Control System, *England*
- Chassaing, R.(1999). Digital Signal Processing; Laboratory Experiments Using C and the TMS320C31 DSK, *John Wiley & Sons, Inc.*
- Hwang, K.(1993).Advanced Computer Architecture: Parallelism, Scalability, Programmability, *McGraw-Hill, Inc.*
- Sorensen, H. V., and Chen, Jianping. (1997). Digital Signal Processing Laboratory Using the TMS320C30, *Prentice Hall*

7. Agradecimentos

Gostaríamos de agradecer pelo custeio da bolsa de iniciação científica concedida pela Fundação de Ciências Aplicadas - FCA