1. Publicação nº	2. Versão	3. Data	5. Distribuição
INPE-2778-PRE/346		*Junho, 1983*	☐ Interna ☒ Externa ☐ Restrita

4. Origem	Programa
DMC/DGC	*TEREAL*

6. Palavras chaves - selecionadas pelo(s) autor(es)

SIMULAÇÃO — *PROCESSAMENTO DISTRIBUÍDO*
SISTEMAS DE CONTROLE — *PROCESSOS CONCORRENTES*

7. C.D.U.: *681.5.01:519.876.5*

8. Título *INPE-2778-PRE/346*

SIMULAÇÃO DIGITAL DESCENTRALIZADA DE SISTEMAS DE CONTROLE EM SUBSISTEMAS AUTÔNOMOS PARA PROCESSAMENTO CONCORRENTE

10. Páginas: *11*

11. Última página: *10*

12. Revisada por

Wilson C.C. da Silva

9. Autoria *Décio Castilho Ceballos*

Assinatura responsável

13. Autorizada por

Nelson de Jesus Parada
Diretor

14. Resumo/Notas *O artigo propõe a utilização de processamento concorrente para simulação digital de sistemas de controle, através da subdivisão em sub sistemas autônomos como, por exemplo, simulador da dinâmica, do estimador, do controlador, onde cada subsistema tem a capacidade de emitir e receber mensa gens. De modo geral, um programa de computador para simulação de sistemas de controle, construído a partir de uma linguagem de alto nível, do tipo uso ge ral, ou para simulação, requer uma rotina de gerenciamento para o controle do processamento dos simuladores autônomos. Este programa pode apresentar carac terísticas desvantajosas, tais como: ser complexo; ser aplicável a uma única arquitetura de sistema de controle; não apresentar adaptabilidade para outras situações de simulação, por exemplo, em esquemas de simulação híbrida em tem po real; e não simular realisticamente a lógica de acionamento dos subsistemas. A utilização de um esquema de simulação concorrente elimina os problemas an teriores, através da exclusão da rotina de gerenciamento da simulação. Os sub sistemas programados, compilados e processados independentemente, poderiam ser executados em um único processador, ou dentro de um esquema de simulação híbrida; ou em um esquema que envolve equipamentos reais, ou até através dos próprios subsistemas do sistema real. Para ilustrar o esquema proposto, apre senta-se um simulador distribuído em subsistemas para processamento concorren te, programado em FORTRAN, utilizando, para comunicação entre processos, uma ferramenta disponível para os computadores Burroughs-6800 denominada "port file".*

15. Observações

Este trabalho será apresentado no IV Encontro Nacional de Automática, Be lém, Pará, de 6 a 13 de julho de 1983.

IV ENCONTRO NACIONAL DE AUTOMÁTICA
SOCIEDADE BRASILEIRA DE AUTOMÁTICA
BELÉM, PARÁ, 6 a 13, JULHO, 1983

SIMULAÇÃO DIGITAL DESCENTRALIZADA DE SISTEMAS DE CONTROLE EM SUBSISTEMAS AUTÔNOMOS PARA PROCESSAMENTO CONCORRENTE

Décio Castilho Ceballos
Instituto de Pesquisas Espaciais - INPE
Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq
C.P. 515 - 12200 - São José dos Campos - SP - Brasil

RESUMO

O artigo propõe a utilização de processamento concorrente para simulação digital de sistemas de controle, através da subdivisão em subsistemas autônomos como, por exemplo, simulador da dinâmica, do estimador, do controlador, onde cada subsistema tem a capacidade de emitir e receber mensagens. De modo geral, um programa de computador para simulação de sistemas de controle, construído a partir de uma linguagem de alto nível, do tipo uso geral, ou para simulação, requer uma rotina de gerenciamento para o controle do processamento dos simuladores autônomos. Este programa pode apresentar características desvantajosas, tais como: ser complexo; ser aplicável a uma única arquitetura de sistema de controle; não apresentar adaptabilidade para outras situações de simulação, por exemplo, em esquemas de simulação híbrida em tempo real; e não simular realisticamente a lógica de acionamento dos subsistemas. A utilização de um esquema de simulação concorrente elimina os problemas anteriores, através da exclusão da rotina de gerenciamento da simulação. Os subsistemas programados, compilados e processados independentemente, poderiam ser executados em um único processador, ou dentro de um esquema de simulação híbrida, ou em um esquema que envolve equipamentos reais, ou até através dos próprios subsistemas do sistema real. Para ilustrar o esquema proposto, apresenta-se um simulador distribuído em subsistemas para processamento concorrente, programado em FORTRAN, utilizando, para comunicação entre processos, uma ferramenta disponível para os computadores Burroughs-6800 denominada "port file".

CONTROL SYSTEMS DIGITAL DECENTRALIZED SIMULATION IN AUTONOMOUS SUBSYSTEMS FOR CONCURRENT PROCESSING

ABSTRACT

The articles proposes utilization of concurrent processing for the digital simulation of control systems through partitioning in autonomous subsystems. As an example, a simulator for the dynamics, the estimator or the controller, where each subsystem can send and receive messages.

simulação simplifica a construção do programa, pode desvinculá-lo da arquitetura de sistema de controle, mas continua dependente do ambiente de processamento para o qual foi projetado.

A utilização de um esquema descentralizado elimina os inconvenientes anteriores, a medida que cada subsistema pode ser elaborado quase independentemente da arquitetura do sistema de controle e do ambiente em que será processado sem necessidade da rotina de gerenciamento.

2 - SIMULAÇÃO EM SUBSISTEMAS AUTÔNOMOS

Os sistemas de controle possuem estruturas fisicamente distribuídas. Isto torna adequado a descentralização do "hardware" e do "software", tanto na implementação real como nas atividades de simulação.

Neste capítulo descrevem-se os requisitos computacionais necessários para um esquema de simulação em subsistemas e propõe-se a simulação em subsistemas autônomos como um passo no sentido do estabelecimento de um esquema integrado que envolve as várias fases de simulação e a implementação real.

2.1 - REQUISITOS COMPUTACIONAIS

As facilidades de comunicação entre os processos que simulam os subsistemas são fundamentais e devem, dentro do possível, seguir as características estruturais normalmente encontradas em sistemas de controle. O trabalho de Kramer et alii (1983) discute em destalhes o problema de comunicação entre processos com "hardware" distribuído.

As facilidades para implementação de multiprocessos são necessárias durante a realização, em um único ambiente computacional, de múltiplos subsistemas.

As facilidades para E/S dos processos em execução, para memória de massa, são necessárias, particularmente para o armazenamento dos resultados da simulação em formato livre. As características de simplicidade de implementação, generalidade e fidelidade de simulação ficariam prejudicadas sem estes recursos de E/S.

As facilidades para programação em alto nível com códigos portáteis para os diversos ambientes de processamento envolvidos, por exemplo os códigos de programação gerados para um subsistema filtro, seriam de aplica

ção genérica ao subsistema real, simulação totalmente digital em um grande computador, ou em um esquema híbrido.

2.2 - RECURSOS INTEGRADOS PARA SIMULAÇÃO E IMPLEMENTAÇÃO

Técnicas aplicáveis a diferentes sistemas de controle, utilizando recursos computacionais distribuídos, vêm sendo desenvolvidos através de linguagens com recursos especiais para tempo real e sistemas integrados de "software" e "hardware" (Kramer et alii, 1983). No entanto, parece existir uma lacuna no que se refere ao desenvolvimento de técnicas para integração dos recursos computacionais nas várias fases do desenvolvimento de um sistema de controle, desde as primeiras simulações até a implementação real.

O desenvolvimento integrado permitiria um aprimoramento na fidelidade dos processos de simulação, melhor qualificação pela utilização sempre de um mesmo "software", e eventualmente de um mesmo "hardware", simplicidade e redução dos custos.

O desenvolvimento de técnicas para simulação distribuída é um passo na direção do desenvolvimento integrado, por tornar mais simples a construção dos simuladores e tornar genérico para as variadas aplicações o "software" correspondente a um subsistema.

2.3 - CARACTERÍSTICAS PECULIARES

Um simulador digital para simulação de um sistema ou subsistema de controle, quando construído através de uma estrutura sequencial (Figura 2.1), utiliza uma rotina de gerenciamento, com o objetivo de coordenar as atividades de entrada e saída, monitoramento das chamadas dos simuladores dos subsistemas e término do processamento.

Um simulador construído dessa forma possui a vantagem de não requisitar nenhum recurso especial de computação. No entanto, pode apresentar características desvantajosas, tais como: complexidade e tempo de implementação; ser aplicável a uma única arquitetura de sistema de controle; não apresentar características de adaptabilidade para outras configurações de simulação; e não simular realisticamente a lógica de acionamento dos subsistemas.

A utilização de linguagens especiais para simulação, tais como, SIMSCRIPT, GPSS, CSSL, CSMP tornam o gerenciamento da simulação transparente para o usuário (Colella et alii, 1974), isto simplifica a construção de

simulador e o desvincula da arquitetura do sistema de controle, mas a existência, mesmo que transparente do gerenciamento centralizado, não permite generalidade para aplicações em outros ambientes de processamento.

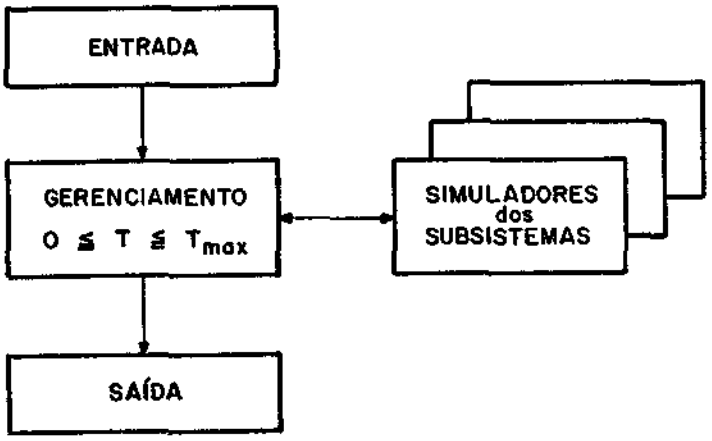

Fig. 2.1 - Estrutura para processamento sequencial.

A utilização de um esquema de simulação paralela elimina a necessidade de rotina de gerenciamento. Isto permite que os subsistemas sejam programados quase independentemente da arquitetura do sistema de controle e processados em qualquer ambiente de simulação, tais como em um único processador, dentro de um esquema de simulação híbrida, em um esquema que envolve equipamentos reais, ou até em subsistemas do sistema real.

3 - SIMULAÇÃO DESCENTRALIZADA UTILIZANDO "PORT FILE"

Neste capítulo são apresentados, a título de exemplo, alguns subsistemas de controle desenvolvidos para simulação descentralizada. Aplicam-se os subsistemas a uma arquitetura de sistema de controle. Discutem-se os aspectos associados à entrada e saída.

3.1 - "PORT FILE"

"Port file" é uma ferramenta de comunicação entre processos baseada em facilidades de entrada e saída, disponíveis nas construções das linguagens ALGOL, FORTRAN, PL/I e COBOL (Burroughs, 1980). Esta ferramenta é utilizada através da declaração de arquivos do tipo "port".

As operações de E/S podem ser especificadas nas opções "WAIT" e "DONTWAIT". Na opção "WAIT" as operações de E/S suspendem o processo enquanto a operação não puder ser realizada. Na opção "DONTWAIT" simplesmente a operação é ignorada quando não puder ser realizada de imediato. Por exemplo, para a opção "WAIT", se um comando "READ" é executado e não existe o registro de mensagem disponível, o programa é suspenso até que algum outro processo execute um comando "WRITE" correspondente. Neste mesmo exemplo, se a opção "DONTWAIT" for utilizada, o processo não é interrompido e a leitura é completada somente quando o registro for disponível de imediato. A princípio, um "port file" pode ter um ou mais "subports", embora esta facilidade não possa ser utilizada em FORTRAN, onde cada um deles pode ser conectado a diferentes processos. Para definição de um "port file", ou para retorno de informações de controle de um "port file" é utilizado uma série de atributos.

O atributo "TITLE" define o nome interno de um processo para o "port file". Este nome deve corresponder ao nome do "port file" do processo complementar para o estabelecimento da ligação. Por exemplo, se dois processos A e B são projetados para comunicação através de um "port file" PAB, então, para ambos os processos A e B, devem estar definidos o arquivo do tipo "port" e "TITLE=PAB". Detalhes sobre toda uma série de atributos podem ser vistos em Burroughs (1980).

3.2 - SUBSISTEMAS TÍPICOS IMPLEMENTADOS

Nesta seção apresenta-se a estrutura básica dos simuladores de cada um dos subsistemas da Figura 3.1, para implementação em estruturas ligadas através da ferramenta "port file".

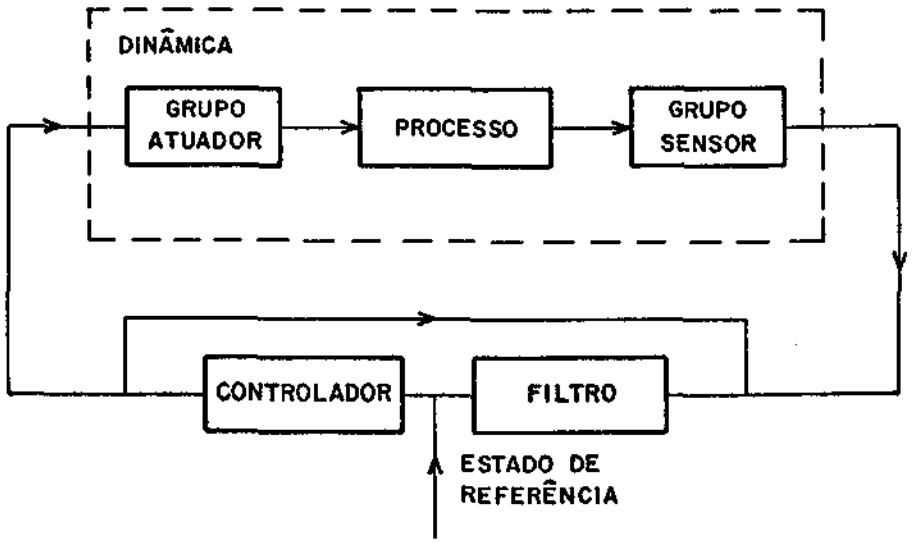

Fig. 3.1 - Um sistema malha fechada típico.

A Tabela 3.1 apresenta o algoritmo básico do simulador da dinâmica. Essencialmente, o programa simulador da dinâmica lê em um "port file" o vetor de controle, calcula os vetores de ruído da dinâmica e dos sensores, realiza operações de integração numérica das equações dinâmicas, imprime em um outro as variáveis de observações e imprime as variáveis de saída em um arquivo de saída. O processo é interrompido quando a variável *tempo* ultrapassa um valor máximo.

TABELA 3.1 - SIMULADOR DA DINÂMICA

```
C**
DEFINA - "Port file" ("TITLE" = DIN/CONTROLE);
DEFINA - "Port file" ("TITLE" = DIN/OBSERV);
C**
LEIA - Parâmetros do processamento em arquivo de dados de entrada;
C**
     REPITA ENQUANTO   TEMPO < TEMPO-FINAL;
          LEIA     - Vetor de controle no "Port file DIN/CONTROLE";
          CALCULE  - Vetores de ruído;
          REALIZE  - Integração numérica;
          CALCULE  - Vetor de observações de estado;
          IMPRIMA  - Vetor de observações de estado no "Port file DIN/OBSERV";
          IMPRIMA  - Tempo, estado, observações em um arquivo de saída;
     CONTINUE;
C**
FIM.
```

A Tabela 3.2 apresenta o algoritmo básico de um filtro sequencial. Essencialmente o programa filtro lê em um "por file" o vetor de controle, em outro o vetor de observações, realiza as operações referentes ao filtro, imprime em outro as variáveis de estado estimadas e imprime estas mesmas variáveis em um arquivo de dados de saída. O programa é interrompido quando algum outro processo, ligado através de "port file", for interrompido.

A Tabela 3.3 apresenta o algoritmo básico de um controlador. Essencialmente o programa controlador lê em um "Port file" o vetor de estado estimado, realiza operações referentes ao controlador, imprime em dois outros as variáveis de controle e imprime as variáveis de saída. O programa é interrompido quando algum outro processo ligado através de "port file" for interrompido.

TABELA 3.2 - PROGRAMA FILTRO

```
C**
DEFINA - "Port file" ("TITLE" = FIL/CONTROLE);
DEFINA - "Port file" ("TITLE" = FIL/OBSERV);
DEFINA - "Port file" ("TITLE" = FIL/ESTIM);
C**
LEIA - Parâmetros do processamento em arquivo de dados de entrada;
C**
     REPITA;
          LEIA     - Vetor de controle do "Port file FIL/CONTROLE";
          LEIA     - Vetor de observação no "Port file FIL/OBSERV";
          REALIZE  - Operações referentes ao Filtro;
          IMPRIMA  - Vetor de estado estimado no "Port file  FIL/ESTIM";
          IMPRIMA  - Estado estimado em um arquivo de saída;
     CONTINUE;
C**
FIM.
```

TABELA 3.3 - PROGRAMA CONTROLE

```
C**
DEFINA - "Port file" ("TITLE" = CONT/ESTIM);
DEFINA - "Port file" ("TITLE" = CONT/CONTROLE 1);
DEFINA - "Port file" ("TITLE" = CONT/CONTROLE 2);
C**
LEIA - Parâmetros de processamento em arquivo de dados de entrada;
C**
     REPITA;
          LEIA     - Vetor de estado estimado no "Port file CONT/ESTIM";
          REALIZE  - Operações referentes ao controlador;
          IMPRIMA  - Vetor de controle no "Port file CONT/CONTROLE 1;
          IMPRIMA  - Vetor de controle no "Port file CONT/CONTROLE 2;
          IMPRIMA  - Vetor de controle em um arquivo de saída;
     CONTINUE;
C**
FIM.
```

3.3 - ACOPLAMENTOS EM ESTRUTURAS

Os programas definidos na Seção 3.2 podem ser utilizados, quase independentemente da arquitetura do sistema de controle. Os simuladores de subsistemas apresentados foram projetados a partir da estrutura elementar típica apresentada na Figura 3.1, onde a dinâmica possui uma ligação para entrada e uma ligação de saída, o controlador uma ligação de entrada e duas de saída e o filtro duas de entradas e uma de saída. Eventualmente algumas arquiteturas de sistemas de controle possuem alguma diferença neste sentido, por exemplo, um programa dinâmico com duas ligações para variáveis de observações, ou um programa controlador com uma única ligação para o vetor de controle. Estas adaptações podem ser feitas utilizando "subports", programas para interface, ou através de pequenas alterações no programa básico.

A estrutura da Figura 3.1 é estabelecida fazendo as correspondências dos "port files" DIN/OBSERV e FIL/OBSERV, DIN/CONTROLE e CONT/CONTROLE 1, FIL/CONTROLE e CONT/CONTROLE 2, FIL/ESTIM e CONT/ESTIM. A simulação deve ser realizada em um ambiente de processamento paralelo, que pode ser conseguido com a utilização do comando "PROCESS", para o caso do computador Burroughs-6800.

4 - CONSIDERAÇÕES FINAIS

O esquema de simulação apresentado no Capítulo 3, vem sendo utilizado para simulações que envolvem estruturas de controle complexas em um desenvolvimento de pesquisa em compensadores de erros. A simplicidade e as características de adaptabilidade a diferentes estruturas do "software" desenvolvido indicam aplicações promissoras para processamento paralelo em simulações de sistema de controle.

A ferramenta "port file", do tipo multipropósitos, não satisfaz evidentemente todos os requisitos desejáveis do Capítulo 2, pois não inclui recursos para tempo real, recursos para comunicação com dispositivos externos, e particularmente é aplicada exclusivamente aos computadores Burroughs. No entanto, o desenvolvimento de ferramentas sem estas deficiências são atualmente viáveis e algumas, com características semelhantes, já foram desenvolvidas (Kramer et alii, 1982).

REFERÊNCIAS

Burroughs, Implementations of Port Files, D3650 General, B6000 series Mark 32, 1980.

Colella, A.M.; O'Sullivan, M.J.; Carlino, D.J. "Systems simulation". Lexington Books, London, 1974.

Ray, I.M. "GPSS/SIMSCRIPT - The dominant simulation languages", Proceedings of the Eighth Annual Simulation Symposium, IEEE Computer Socity, CA, 1975.

Kramer, K.; Magee, J.; Slomann, M.; Slister, A. "CONIC: an integrated approach to distributed computer control systems". IEEE Procedure, vol. 130, January 1983.

Korn, G.A.; Wait, J.V. "Digital continuous-system simulation". Prentice--Hall, Inc, New Jersey, 1978.

Markowitz, H.M.; Hausner, B.; Karr, H.W. "SIMSCRIPT A simulation programming language". Prentice-Hall, Inc, N.J., 1963.

Ricketts, I.W.; Dickie, A.A. "MIMESIS - A machine independent CSSL for minicomputer systems". Prentice-Hall, Inc, N.J. 1963.

Rook, P.E. "The EAI ECSSL hybrid compiler". Proceedings of the UKSC conference on computer simulation, IPC Science and Technology Press, England, 1978.

Spriet, J.A.; Vansteenkiste, "Computer-aided modelling and simulation". Academic Press, New York, 1982.