

UNIVERSIDADE DO PORTO
FACULDADE DE ENGENHARIA

PRODEP - Medida 4.3/7/6/92/92

Estágio 3.9 - "Simulação Computacional de Processos Industriais"

António Barros Ferreira



4

66(042 3)/LCG 1032/FCR2
13 10 09

REFERENCIA DO PROJECTO -
PRODEP - Medida 4.3/7.6/92/93

PARECER TÉCNICO

Acompanhei os trabalhos de estágio realizados no Instituto de Sistemas e Robótica pelo aluno António Manuel Gomes P. B. Ferreira, bem como apreciei os trabalhos produzidos e o relatório oportunamente apresentado.

Com base no acompanhamento referido, considero que aqueles trabalhos reflectem o trabalho efectivamente realizado, revelando empenhamento do estagiário.

Sou portanto de parecer que foram plenamente atingidos os objectivos inicialmente propostos.

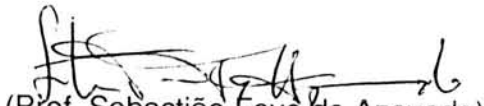
Porto, 30 de Setembro de 1993

O Supervisor do Estágio no Instituto de Sistemas e Robótica



(Prof. Jorge Leite Martins de Carvalho)

O Supervisor do Estágio na Faculdade de Engenharia da Universidade do Porto



(Prof. Sebastião Feyo de Azevedo)

Simulação Computacional de Processos Industriais

O presente trabalho, realizado no âmbito do Subprograma 4, Medida 4.3 do PRODEP, e desenvolvido no Polo do Porto do Instituto de Sistemas e Robótica, consiste no estudo por computador de processos dinâmicos. O estágio foi efectuado sob orientação do Prof. Dr. Sebastião Feye de Azevedo (Prof. Associado da FEUP) e Prof. Dr. Jorge Martins de Carvalho (Prof. Catedrático e Director da FEUP).

Foi desenvolvido um banco de dados de processos industriais e académicos para simulação dinâmica em tempo real, utilizando o MIMOSA (Multiple Input-Multiple Output System analysis), que é um programa desenvolvido no Departamento de Engenharia Química da Faculdade de Engenharia da Universidade do Porto.

Neste relatório, apresenta-se uma descrição sumária do MIMOSA, assim como alguns dos casos desenvolvidos.

António Manuel Barros Ferreira, Sebastião Feye de Azevedo

92/7/1 a 92/9/30

Departamento de Engenharia Química
Faculdade de Engenharia da Universidade do Porto

CASOS DO MIMOSA

Introdução

Com este trabalho pretendeu-se criar um conjunto de casos que simulem processos dinâmicos, tanto para problemas teóricos como para casos práticos.

Este 'banco de casos' engloba problemas que se podem dividir em três partes distintas:

- Casos SISO - sistemas lineares de uma entrada e uma saída.
- Casos MIMO - sistemas lineares de múltiplas entradas e saídas.
- Casos não lineares.

No presente trabalho apenas foram desenvolvidos casos MIMO e casos SISO. O resumo dos trabalhos propostos encontra-se na tabela 1.

Tabela 1 - Casos desenvolvidos para o MIMOSA.

CASO	DESCRIÇÃO
P1L1	Processo de 1ª ordem e Carga de 1ª ordem
P1L2	Processo de 1ª ordem e Carga de 2ª ordem
P1L3	Processo de 1ª ordem e Carga de 3ª ordem
P2L1	Processo de 2ª ordem e Carga de 1ª ordem
P2L2	Processo de 2ª ordem e Carga de 2ª ordem
P2L3	Processo de 2ª ordem e Carga de 3ª ordem
P3L1	Processo de 3ª ordem e Carga de 1ª ordem
P3L2	Processo de 3ª ordem e Carga de 2ª ordem
P3L3	Processo de 3ª ordem e Carga de 3ª ordem
I1L1	Integrador de 1ª ordem e Processo de 1ª ordem
I1L2	Integrador de 1ª ordem e Processo de 2ª ordem
I2L1	Integrador de 2ª ordem e Processo de 1ª ordem
I2L2	Integrador de 2ª ordem e Processo de 2ª ordem
IPL1	Integrador Puro e Carga de 1ª ordem
IPL2	Integrador Puro e Carga de 2ª ordem
IR1L1	Resposta Inversa (caso I) e Carga de 1ª ordem
IR1L2	Resposta Inversa (caso I) e Carga de 2ª ordem
IR2L1	Resposta Inversa (caso II) e Carga de 1ª ordem
IR2L2	Resposta Inversa (caso II) e Carga de 2ª ordem
CC12L11	Cascata (ciclo interno de 1ª ordem, ciclo externo de 2ª ordem e perturbações de 1ª ordem)
DISTCOL	Coluna de Destilação Binária
BOILER	Reebulidor
SHELLCON	Problema de Controlo da Shell
ROLLMILL	Moinho de Compressão a Frio
MISSILE	Controlo do Autopiloto de um Míssil
HIPLANT	Planta de ordem elevada

Cada um destes casos simula a dinâmica de um determinado processo englobando as dinâmicas dos sensores e elementos finais de controlo. Todos eles foram desenvolvidos em ambiente 'MIMOSA' que é um package de software desenvolvido no Departamento de Engenharia Química da Faculdade de Engenharia da Universidade do Porto. A descrição deste software pelos autores [1] é dada em seguida.

O MIMOSA

Os processos reais nas áreas de química, bioquímica e indústrias alimentares são na sua maioria sistemas MIMO (Multiple Input, Multiple Output) não lineares. As suas dinâmicas e controlo são difíceis de estudar, ambos por razões teóricas e práticas. Do ponto de vista teórico é conhecido que a maior parte das análises de estabilidade e de controlo se referem a sistemas lineares, em muitos casos sistemas SISO (Single Input, Single Output). As estratégias de controlo baseiam-se em técnicas adaptativas com a contínua identificação de modelos simplificados. Na prática, experiências com processos industriais reais são difíceis de executar por razões de economia e segurança.

A arte no que se refere à modelização e simulação de processos é mais brilhante. A modelização de processos é uma disciplina bem estabelecida. Para um número significativo de processos determinísticos podem ser escritos modelos que, com parâmetros apropriados determinados por estimativas de parâmetros não lineares, representem bem o comportamento dos processos. A capacidade computacional necessária para a computação em tempo real, com todas as suas características interactivas, está hoje disponível a preços baixos (máquinas 80386, 80486 ou equivalentes).

Simuladores para treino e desenvolvimento estão actualmente a ser propostos e utilizados por um número de companhias e grupos de investigação (Shewchuk et al.; Machietto and Kessianides; 1992) como meio de aumentar e melhorar os processos e o conhecimento do controlo dos processos.

Em todos os casos em que os tempos de operação o tornam fiável, a simulação e o treino em tempo real é de particular interesse quer para engenheiros de controlo quer para os de produção. Isto dá a visão interna e uma sensibilidade adicional da relevância do tempo em operações processuais. Um número de outras características devem ser incorporadas de modo a tornar um simulador próximo de um processo real, tal como os sinais analógicos reais de saída, simulação das medidas com ruídos e atrasos de transporte; a entrada de sinais analógicos (controles) para actuar nos elementos finais de controlo; a simulação dos parâmetros variáveis no tempo, todos definidos pelo utilizador e podendo ser mudados em linha.

As questões teóricas e práticas levantadas conduziram ao objectivo de criar um ambiente de fácil manipulação para simulação de processos, onde casos baseados em modelos determinísticos possam ser desenvolvidos e que incorporem características práticas de operações processuais importantes.

Definições e Conceitos

'MIMOSA - Multiple Input Multiple Output System Analysis' - é um package que simula em tempo real, em 'tempo escalonado' ('scaled time') ou em 'tempo simulado', o comportamento de processos representados matematicamente por conjuntos de equações algébrico-diferenciais não lineares. 'Tempo escalonado' é uma forma opcional de operação onde um segundo de tempo real corresponde a 's' segundos de simulação, sendo 's' um parâmetro definido pelo utilizador. A opção de 'tempo simulado' corresponde à forma de operação onde não há tempo de controlo.

REPRESENTAÇÃO PROCESSUAL

A figura 1 representa o processo tal como é simulado pelo MIMOSA. As variáveis relevantes com o propósito de implementação de processos estão agrupadas nos seis principais vectores seguintes:

- $c[]$ - entradas do controlo
- $u[]$ - cargas (perturbações)
- $ud[]$ - perturbações com atrasos
- $p[]$ - parâmetros do processo
- $y[]$ - 'variáveis do sistema'
- $f[]$ - derivadas das 'variáveis do sistema'

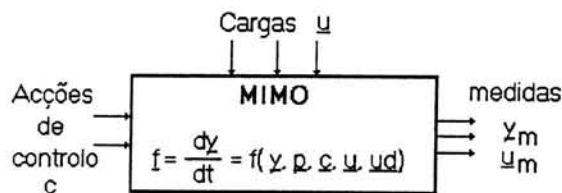


Fig 1 - Representação de um processo MIMO

O 'vector do sistema' - $y[]$ é um conceito específico do MIMOSA. Ele é composto por todas as variáveis que têm a primeira derivada explicitamente representada no conjunto das equações do modelo. Isto inclui as variáveis de estado formais e todas as outras variáveis que resultam de se adicionar ao conjunto de equações a dinâmica de sensores e elementos finais de controlo. Todas as variáveis de estado do sistema podem ser enviadas para o exterior através de uma interface D/A ou por uma porta RS232, dependendo da versão do package que se está a utilizar. A saída é constituída pelas variáveis de estado medidas - y_m - e as cargas medidas - u_m . Ambas são subconjuntos de y .

As seguintes características gerais devem ser tomadas em conta:

1. Todas as entradas de controlo ($c[]$) podem ser monitorizadas. Tais variáveis podem ser mudadas manualmente através do teclado (com as Comunicações desligadas) ou de um 'controlador externo' (via carta A/D ou RS232).
2. O utilizador poderá monitorizar todas as variáveis declaradas como perturbações ($u[]$). Essas variáveis podem ser mudadas manualmente com o teclado. Isto inclui funções degrau, seno, rampa ou introdução de ruídos.
3. Todos os parâmetros designados como $p[]$ nas equações do modelo podem ser monitorizados, e os seus valores mudados em linha (isto inclui ruídos e tempos mortos definidos num ficheiro, que o MIMOSA automaticamente adicionará à lista de parâmetros e como tal estará à disposição para mudanças em linha).
4. Todas as 'variáveis de estado do sistema' ($y[]$) podem ser monitorizadas e podem ser todas escolhidas como variáveis de saída, com ruído aleatório sobreposto.
5. O programa MIMOSA permite a implementação de atrasos associados às medidas (u_m e y_m), às entradas das acções de controlo e às perturbações. Esta caracterização independente permite a simulação de efeitos simultâneos de 'atrasos de transporte' (geralmente associados a medidas) e de tempos mortos inerentes ao processo (estes últimos associados a processos de parâmetros distribuídos). Tal característica aumenta significativamente a gama de aplicações a usar com o MIMOSA.

CONFIGURAÇÃO DO HARDWARE

O conceito que presidiu ao seu desenvolvimento foi o de 'comunicações' via meios analógicos. Tal solução é apresentada na Fig. 2a. Tem a limitação de ser cara se um número elevado de sinais de entrada/saída tiver que ser usado e também necessita que se escreva um 'driver' específico para cada carta A/D ou D/A diferente. Tem a vantagem de que controladores industriais reais possam ser empregues quer para treino quer para desenvolvimento. Com tais controladores apenas as aplicações em tempo real têm interesse. Foi desenvolvido um package alternativo, que emprega comunicações via RS232 bidirecionais. Isso está representado na Fig 2b. É de grande importância para a investigação e desenvolvimento dado que permite pelo mesmo custo e em formas standardizadas qualquer número de sinais de entrada/saída e permite também estudos em 'tempo escalonado' e em 'tempo de simulação' sem perda de características importantes.

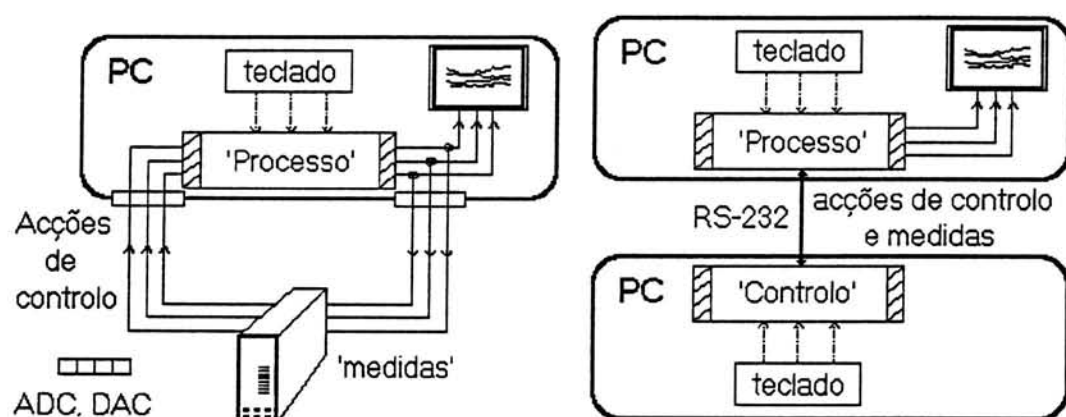


Figura 2 - MIMOSA com a) um controlador industrial ou com b) um computador.

ASPECTOS ALGORÍTMICOS E DA IMPLEMENTAÇÃO EM COMPUTADORES E INTERACÇÃO COM O OPERADOR

A solução geral do MIMOSA (codificado em MS C Vs 5.1) está baseada na integração em tempo real do conjunto de equações algébrico-diferenciais que constituem o modelo do processo. Isso é conseguido na presente versão, empregando um algoritmo de passo variável baseado no de Fehlberg de 4ª/5ª ordem embebido, a fórmula é do tipo Runge-Kutta (Chapra e Canale, 1989).

Um ciclo básico de trabalho consiste nos seguintes passos - (i) Lêr a acção de controlo; (ii) Continuar com a integração; (iii) Apresentar os resultados no ecrã e em ficheiro; (iv) Medidas externas; (v) Verificar o teclado para interacção enquanto se controla o tempo. Correntemente, com uma máquina 80486, um sistema bem condicionado de 20 equações diferenciais pode ser processado em 0.25 segundos, isto é, um observador pode 'ver' o processo todos os 0.25 segundos.

O conceito de base para se criar uma interface para o utilizador foi o de 'todos-em-um' GUI que significa Interface Gráfica do Utilizador ('Graphics User Interface') onde todas as informações são apresentadas apenas numa página (ecrã). Esta é a melhor característica para simplicidade de operação.

A Fig. 3 mostra o método 'todos-em-um'. O ecran está essencialmente dividido em duas zonas. Na metade superior uma utilidade gráfica simula um registador de quatro canetas. Na metade inferior as principais variáveis são monitorizadas e estão à disposição para serem mudadas em linha. Para além destas duas zonas o operador tem em qualquer momento a indicação das acções possíveis, tais como colocar a comunicação em estado on/off, mudar as variáveis a serem traçadas no registador, iniciar/parar a gravação de resultados em disco, iniciar/parar a impressão, fazer uma impressão do ecran,...

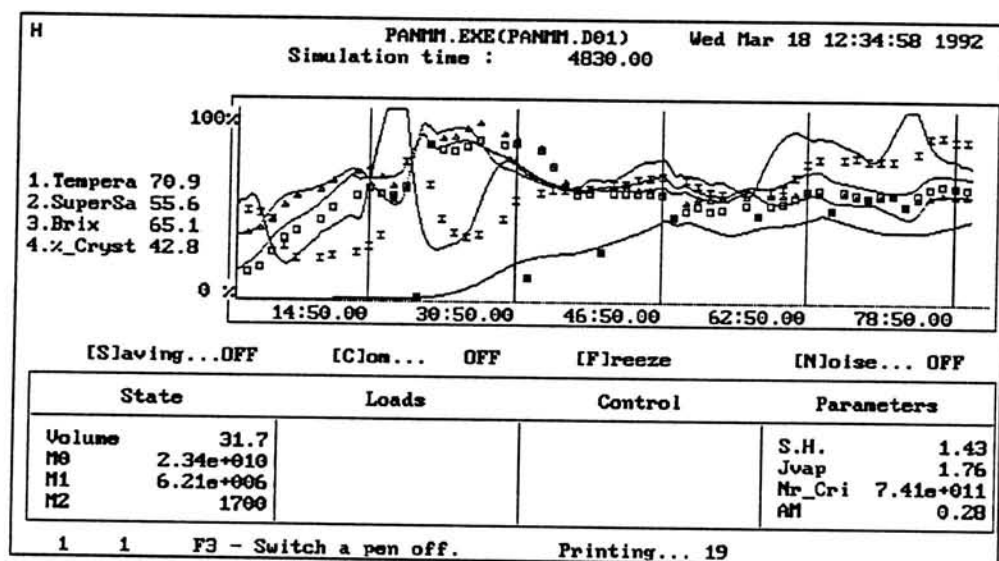


Fig. 3. - Ecran do MIMOSA para o estudo da cristalização.

A estrutura da aplicação prática

O package MIMOSA (Fig.4) é composto de três ficheiros, que são:

- O builder ('construtor') - responsável pela criação e computação de casos;
- O kernel - que por acção do builder será lincado a um ficheiro objecto correspondente a um caso específico, dando origem a um ficheiro executável;
- O ficheiro de dados com os parâmetros de controlo da integração.

Conclusões

O MIMOSA (Multiple Input, Multiple Output Systems Analysis) é um package de software para a simulação em tempo real de sistemas MIMO não lineares. Contém um número de características que permitem ao utilizador criar o ambiente de um processo, com as suas linearidades ou não linearidades, as suas acções de controlo, suas possíveis cargas, suas variáveis (mesuráveis ou não), e os seus parâmetros de processo que podem variar no tempo. Constitui essencialmente um laboratório à escala do computador para ensino e investigação em sistemas dinâmicos e controlo.

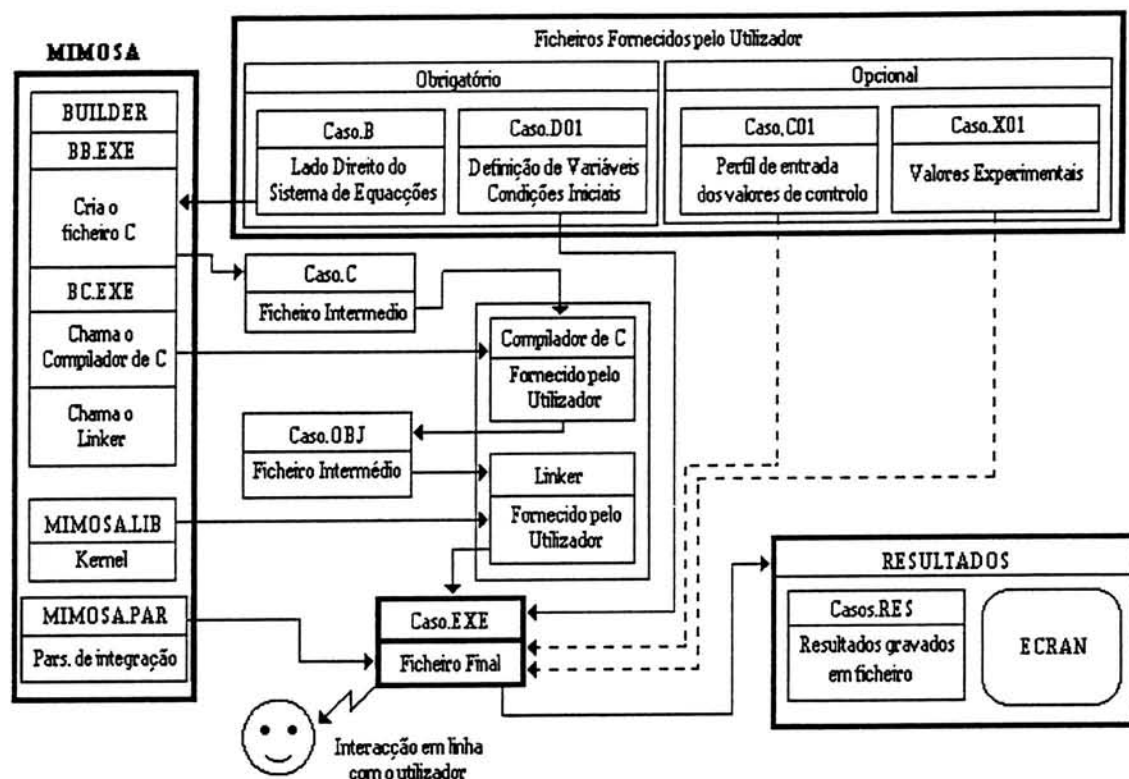
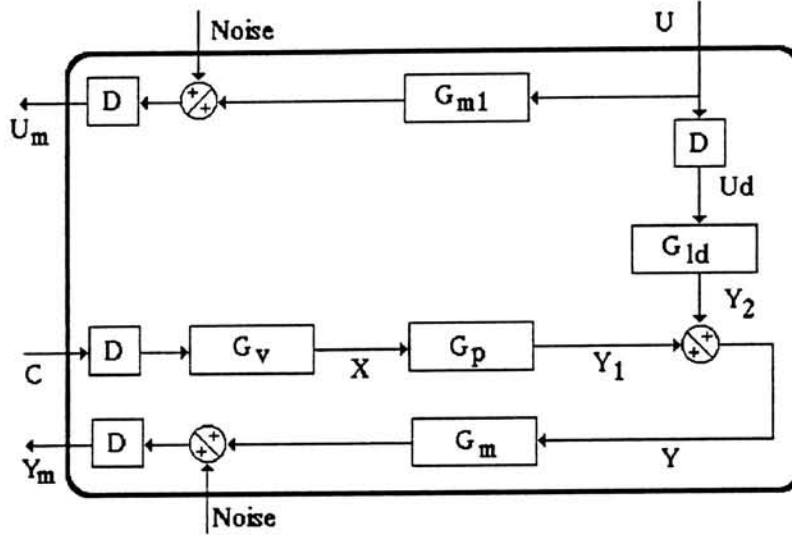


Fig. 4 - Os ficheiros e as aplicações do MIMOSA.

REFERÊNCIAS

- [1] Pedro C. C. Pimenta, Sebastião Feyo de Azevedo,
 "Real-Time Dynamic Simulation of Non-Linear MIMO Systems";
 Departamento de Engenharia Química, Faculdade de Engenharia da Universidade do Porto,
 Outubro, 1992

1. The process diagram is shown below:



2. The set of transfer functions in the Laplace domain is written as:

$$G_v = \frac{K_v}{\tau_v s + 1} \quad (1)$$

$$G_m = \frac{K_m}{\tau_m s + 1} \quad (4)$$

$$G_p = \frac{K_p (\tau_1 s + 1)}{\tau_p s + 1} \quad (2)$$

$$G_{m1} = \frac{K_{m1}}{\tau_{m1} s + 1} \quad (5)$$

$$G_{ld} = \frac{K_{ld}}{(\tau_{ld} s + 1)(\tau_{ld1}^2 s^2 + 2\xi_{ld}\tau_{ld1}s + 1)} \quad (3)$$

D - represents delays in inputs and outputs

3. The equivalent equations in the time domain are:

$$\frac{dx}{dt} = (K_v c - x) / \tau_v \quad (6)$$

$$\frac{dy_a}{dt} = (x - y_a) / \tau_p \quad (7)$$

$$y_1 = K_p \left(\tau_1 \frac{dy_a}{dt} + y_a \right) \quad (8)$$

$$\frac{dy_2}{dt} = y_b \quad (9)$$

$$\frac{dy_b}{dt} = y_c \quad (10)$$

$$\frac{dy_c}{dt} = [K_{ld} u_d - y_2 - (\tau_{ld} + 2\xi_{ld}\tau_{ld1}) y_b - (2\xi_{ld}\tau_{ld1}\tau_{ld} + \tau_{ld1}^2) y_c] / (\tau_{ld}\tau_{ld1}^2) \quad (11)$$

$$y = y_1 + y_2 \quad (12)$$

$$\frac{dy_m}{dt} = (K_m y - y_m) / \tau_m \quad (13)$$

$$\frac{du_m}{dt} = (K_{m1} u - u_m) / \tau_{m1} \quad (14)$$

IMPORTANT NOTE: Time delays are not represented explicitly in any of the equations above. However it is a feature of MIMOSA to assign automatically such delays to all inputs and outputs. The initial values for such parameters are given by the user in the .D01 file.

4. User-defined files:

The P1L3.B file:

```
/* 1st order process & 3rd order process load
Antonio Ferreira, Dez. 91 */
```

```
/* Valve */
if(p[1]<1e-3)
{
    y[0]=p[0]*c[0];
    f[0]=0.;
}
else f[0]=(p[0]*c[0]-y[0])/p[1];
```

```
/* Process */
f[1]=(y[0]-y[1])/p[3];
y[2]=p[2]*(p[4]*f[1]+y[1]);
f[2]=0.;
```

```
/* Process load */
f[3]=y[4];
f[4]=y[5];
```

```
f[5]=(p[5]*ud[0]-y[3]-(p[6]+2.*p[8]*p[7])*y[4]-
(2*p[8]*p[6]*p[7]+p[7]*p[7])*y[5])/(p[6]*p[7]*p[7]);
```

```
/* Output variable */
p[13]=y[2]+y[3];
```

```
/* Feedback control sensor */
if(p[10]<1e-3)
{
    y[6]=p[9]*p[13];
    f[6]=0.;
}
else f[6]=(p[9]*p[13]-y[6])/p[10];
```

```
/* Feedforward control sensor */
if(p[12]<1e-3)
{
    y[7]=p[11]*u[0];
    f[7]=0.;
}
else f[7]=(p[11]*u[0]-y[7])/p[12];
```

The P1L3.D01 file:

C
C Initial conditions for the solution of system P1L3

C
C N. of differential equations
8

C Name	Starting Value	Min	Max	Variances to noise)	Measuring Delays
x	50.	0.	100.	3.	100
ya(aux)	50.	0.	100.		
y1	50.	0.	100.		
y2	50.	0.	100.		
yb(aux)	50.	0.	100.		
yc(aux)	0.	0.	100.		
ym	50.	0.	100.		
um	50.	0.	100.		

C
C
C N. of loads/disturbances
1

C Name	Starting Value	Load Delay (sec/100)
Load	50.	100.

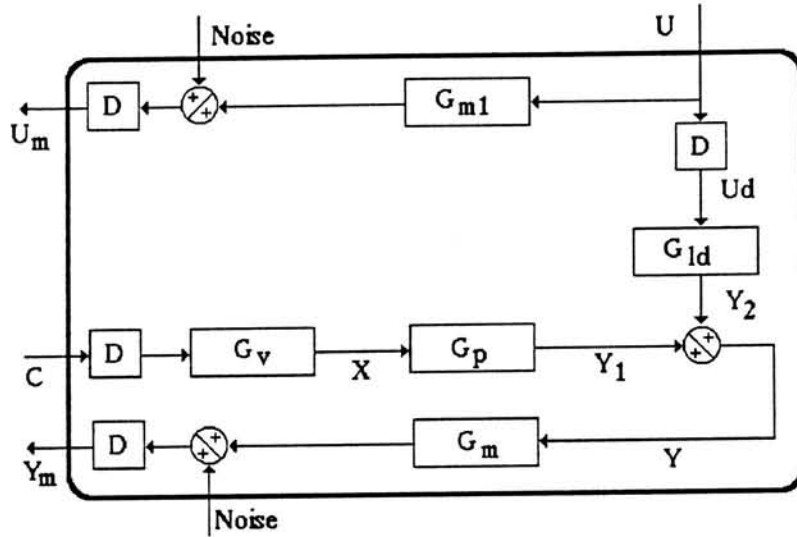
C N. of control actions
1

C Name	Starting Value	Control Delay (sec/100)
Control	50.	40

C
C
C N. of parameters
14

C Name	Starting Value
Kv	1.
Tv	1.
Kp	1.
Tp	1.
Tl	1.
Kld	1.
Tld	1.
Tld1	1.
Csi_ld	1.
Km	0.5
Tm	1.
Km1	1.
Tm1	1.
y	100.

1. The process diagram is shown below:



2. The set of transfer functions in the Laplace domain is written as:

$$G_v = \frac{K_v}{\tau_v s + 1} \quad (1)$$

$$G_m = \frac{K_m}{\tau_m s + 1} \quad (4)$$

$$G_p = \frac{K_p (\tau_1 s + 1)}{(\tau_p s + 1) (\tau^2 s^2 + 2\zeta\tau s + 1)} \quad (2)$$

$$G_{m1} = \frac{K_{m1}}{\tau_{m1} s + 1} \quad (5)$$

$$G_{ld} = \frac{K_{ld}}{(\tau_{ld} s + 1) (\tau_{ld1}^2 s^2 + 2\zeta_{ld}\tau_{ld1} s + 1)} \quad (3)$$

3. The equivalent equations in the time domain are :

$$\frac{dx}{dt} = (K_v c - x) / \tau_v \quad (6)$$

$$\frac{dy_a}{dt} = y_b \quad (7)$$

$$\frac{dy_b}{dt} = y_c \quad (8)$$

$$\frac{dy_c}{dt} = [x - y_a - (\tau_p + 2\xi\tau) y_b - (2\xi\tau\tau_p + \tau^2) y_c] / (\tau_p\tau^2) \quad (9)$$

$$y_1 = K_p \left(\tau_1 \frac{dy_a}{dt} + y_a \right) \quad (10)$$

$$\frac{dy_2}{dt} = y_d \quad (11)$$

$$\frac{dy_d}{dt} = y_e \quad (12)$$

$$\frac{dy_e}{dt} = [K_{ld} u_d - y_2 - (\tau_{ld} + 2\xi_{ld}\tau_{ld1}) y_d - (2\xi_{ld}\tau_{ld1}\tau_{ld} + \tau_{ld1}^2) y_e] / (\tau_{ld}\tau_{ld1}^2) \quad (13)$$

$$y = y_1 + y_2 \quad (14)$$

$$\frac{dy_m}{dt} = (K_m y - y_m) / \tau_m \quad (15)$$

$$\frac{du_m}{dt} = (K_{m1} u - u_m) / \tau_{m1} \quad (16)$$

IMPORTANT NOTE: Time delays are not represented explicitly in any of the equations above. However it is a feature of MIMOSA to assign automatically such delays to all inputs and outputs. The initial values for such parameters are given by the user in the .D01 file.

4. User-defined files:

The P3L3.B file:

```
/* 3rd order process & 3rd order process load
Antonio Ferreira, Dez. 91 */
```

```
/* Valve */
if(p[1]<1e-3)
{
    y[0]=p[0]*c[0];
    f[0]=0.;
}
else f[0]=(p[0]*c[0]-y[0])/p[1];

/* Process */
f[1]=y[2];
f[2]=y[3];
f[3]=(y[0]-y[1]-(p[3]+2.*p[5]*p[4])*y[2]-
(2*p[5]*p[3]*p[4]+p[4]*p[4])*y[3])/(p[3]*p[4]*p[4]);
y[4]=p[2]*(p[6]*f[1]+y[1]);
f[4]=0.;

/* Process load */
f[5]=y[6];
f[6]=y[7];
```

```
f[7]=(p[7]*ud[0]-y[5]-(p[8]+2.*p[10]*p[9])*y[6]-
(2*p[10]*p[8]*p[9]+p[9]*p[9])*y[7])/(p[8]*p[9]*p[9]);
```

```
/* Output variable */
p[15]=y[4]+y[5];

/* Feedback control sensor */
if(p[12]<1e-3)
{
    y[8]=p[11]*p[15];
    f[8]=0.;
}
else f[8]=(p[11]*p[15]-y[8])/p[12];

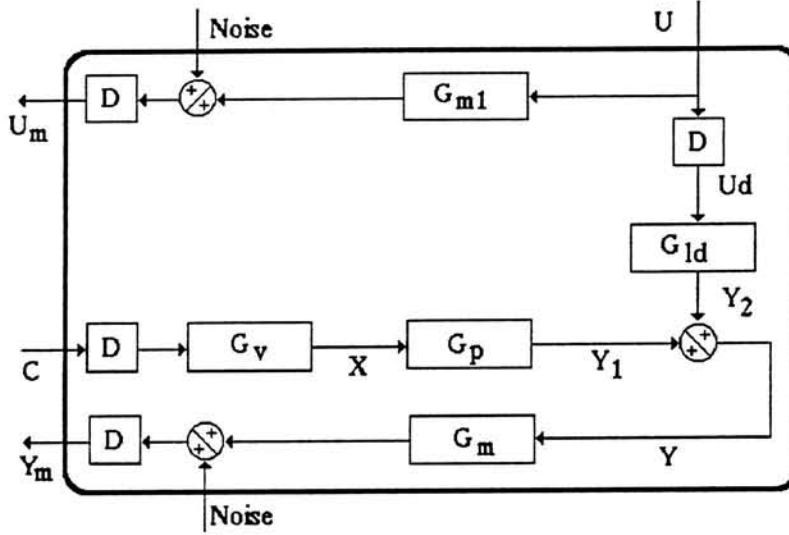
/* Feedforward control sensor */
if(p[14]<1e-3)
{
    y[9]=p[13]*u[0];
    f[9]=0.;
}
else f[9]=(p[13]*u[0]-y[9])/p[14];
```

The P3L3.D01 file:

C
 C Initial conditions for the solution of system P3L3
 C
 C N. of differential equations
 10
 C
 C Starting Variances Measuring
 C Name Value Min Max (to noise) Delays
 C
 x 50. 0. 100. 3. 100
 ya(aux) 50. 0. 100.
 yb(aux) 0. 0. 100.
 yc(aux) 0. 0. 100.
 y1 50. 0. 100.
 y2 50. 0. 100.
 yd(aux) 50. 0. 100.
 ye(aux) 0. 0. 100.
 ym 50. 0. 100.
 um 50. 0. 100.
 C
 C N. of loads/disturbances
 1
 C Starting Load Delay
 C Name Value (sec/100)
 Load 50. 100
 C
 C N. of control actions
 1

C
 C Starting Control Delay
 C Name Value (sec/100)
 Control 50. 40
 C
 C N. of parameters
 16
 C
 C Starting
 C Name Value Comments
 Kv 1. p0
 Tv 1. p1
 Kp 1. p2
 Tp 1. p3
 T 1. p4
 Csi 1. p5
 Tl 1. p6
 Kld 1. p7
 Tld 1. p8
 Tld1 1. p9
 Csi_ld 1. p10
 Km 0.5 p11
 Tm 1. p12
 Km1 1. p13
 Tm1 1. p14
 y 100. p15

1. The process diagram is shown below:



2. The set of transfer functions in the Laplace domain is written as:

$$G_v = \frac{K_v}{\tau_v s + 1} \quad (1)$$

$$G_m = \frac{K_m}{\tau_m s + 1} \quad (4)$$

$$G_p = \frac{K_i}{s} * \frac{K_a (\tau_1 s + 1)}{(\tau_a s + 1)} \quad (2)$$

$$G_{m1} = \frac{K_{m1}}{\tau_{m1} s + 1} \quad (5)$$

$$G_{ld} = \frac{K_{ld}}{\tau_{ld}^2 s^2 + 2\xi_{ld}\tau_{ld}s + 1} \quad (3)$$

The process given by G_p can be expressed as the second order transfer function $G_p(s) = \frac{K_p (\tau_1 s + 1)}{\tau_p^2 s^2 + 2\xi_p \tau_p s}$, where the parameters are:

$$K_p = K_a * K_i$$

$$\tau_p = \sqrt{\tau_a}$$

$$\xi_p = \frac{1}{2\sqrt{\tau_a}}$$

D - represents delays in inputs and outputs

3. The equivalent equations in the time domain are:

$$\frac{dx}{dt} = (K_v c - x) / \tau_v \quad (6)$$

$$\frac{dy_2}{dt} = y_c \quad (10)$$

$$\frac{dy_a}{dt} = y_b \quad (7)$$

$$\frac{dy_c}{dt} = (K_{ld} u_d - y_2 - 2\xi_{ld}\tau_{ld} y_c) / \tau_{ld}^2 \quad (11)$$

$$\frac{dy_b}{dt} = (x - 2\xi_p\tau_p y_b) / \tau_p^2 \quad (8)$$

$$y = y_1 + y_2 \quad (12)$$

$$y_1 = K_p \left(\tau_1 \frac{dy_a}{dt} + y_a \right) \quad (9)$$

$$\frac{dy_m}{dt} = (K_m y - y_m) / \tau_m \quad (13)$$

$$\frac{du_m}{dt} = (K_{m1} u - u_m) / \tau_{m1} \quad (14)$$

IMPORTANT NOTE: Time delays are not represented explicitly in any of the equations above. However it is a feature of MIMOSA to assign automatically such delays to all inputs and outputs. The initial values for such parameters are given by the user in the .D01 file.

4. User-defined files:

The I1L2.B file:

```
/* 1st order integrating process & 2nd order process load
Antonio Ferreira, Dez. 91 */
```

```
/* Parameters definition */
```

```
p[2]=p[13]*p[14];
p[3]=sqrt(p[15]);
p[4]=1/(2.*p[3]);
```

```
/* Valve */
```

```
if(p[1]<1e-3)
```

```
{
    y[0]=p[0]*c[0];
    f[0]=0.;
}
```

```
else f[0]=(p[0]*c[0]-y[0])/p[1];
```

```
/* Process */
```

```
f[1]=y[2];
f[2]=(y[0]-2*p[4]*p[3]*y[2])/(p[3]*p[3]);
y[3]=p[2]*(p[5]*f[1]+y[1]);
f[3]=0.;
```

```
/* Process load */
```

```
f[4]=y[5];
f[5]=(p[6]*ud[0]-y[4]-2.*p[8]*p[7]*y[5])/(p[7]*p[7]);
```

```
/* Output variable */
```

```
p[16]=y[3]+y[4];
```

```
/* Feedback control sensor */
```

```
if(p[10]<1e-3)
```

```
{
    y[6]=p[9]*p[16];
    f[6]=0.;
}
else f[6]=(p[9]*p[16]-y[6])/p[10];
```

```
/* Feedforward control sensor */
```

```
if(p[12]<1e-3)
```

```
{
    y[7]=p[11]*u[0];
    f[7]=0.;
}
else f[7]=(p[11]*u[0]-y[7])/p[12];
```

The I1L2.D01 file:

C
 C Initial conditions for the solution of system I1L2
 C
 C N. of differential equations
 8
 C

C Name	Starting Value	Min	Max	Variance (to noise)	Measuring Delays
x	50.	0.	100.	3.	100
ya(aux)	50.	0.	100.		
yb(aux)	0.	0.	100.		
y1	50.	0.	100.		
y2	50.	0.	100.		
yc(aux)	50.	0.	100.		
ym	50.	0.	100.		
um	50.	0.	100.		

 C
 C N. of loads/disturbances
 1
 C

C Name	Starting Value	Load Delay (sec/100)
Load	50.	100

 C
 C N. of control actions
 1

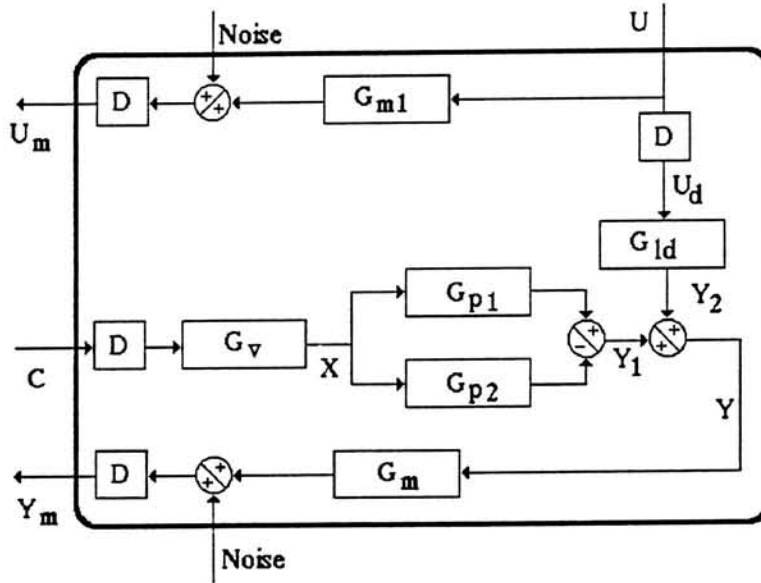
C Name	Starting Value	Control Delay (sec/100)
Control	50.	40

C
 C
 C N. of parameters
 17
 C

C Name	Starting Value	Comments
Kv	1.	p0
Tv	1.	p1
Kp	1.	p2
Tp	1.	p3
Csi_p	1.	p4
T1	1.	p5
Kld	1.	p6
Tld	1.	p7
Csi_ld	1.	p8
Km	.5	p9
Tm	1.	p10
Km1	1.	p11
Tm1	1.	p12
Ki	1.	p13
Ka	.1	p14
Ta	1.	p15
y	100.	p16

 Note: p15 > 0

1. The process diagram is shown below:



2. The set of transfer functions in the Laplace domain is written as:

$$G_v = \frac{K_v}{\tau_v s + 1} \quad (1)$$

$$G_{ld} = \frac{K_{ld}}{\tau_{ld}^2 s^2 + 2\xi_{ld}\tau_{ld}s + 1} \quad (4)$$

$$G_{p1} = \frac{K_a}{s} \quad (2)$$

$$G_m = \frac{K_m}{\tau_m s + 1} \quad (5)$$

$$G_{p2} = \frac{K_b}{\tau_b s + 1} \quad (3)$$

$$G_{m1} = \frac{K_{m1}}{\tau_{m1} s + 1} \quad (6)$$

The process given by G_{p1} and G_{p2} can be expressed as the second order transfer function $G_p(s) = \frac{K_p(\tau_1 s + 1)}{\tau_p^2 s^2 + 2\xi_p \tau_p s + 1}$

where the parameters are:

$$K_p = K_a \quad \tau_p = \sqrt{\tau_b} \quad \xi_p = \frac{1}{2\sqrt{\tau_b}} \quad \tau_1 = \frac{K_a \tau_b - K_b}{K_a}$$

There will be inverse response when $K_a \tau_b < K_b$

D - represents delays in inputs and outputs

3. The equivalent equations in the time domain are:

$$\frac{dx}{dt} = (K_v c - x) / \tau_v \quad (7)$$

$$\frac{dy_2}{dt} = y_c \quad (11)$$

$$\frac{dy_a}{dt} = y_b \quad (8)$$

$$\frac{dy_c}{dt} = (K_{ld} u_d - y_2 - 2\xi_{ld}\tau_{ld} y_c) / \tau_{ld}^2 \quad (12)$$

$$\frac{dy_b}{dt} = (x - 2\xi_p\tau_p y_b) / \tau_p^2 \quad (9)$$

$$y = y_1 + y_2 \quad (13)$$

$$y_1 = K_p (\tau_1 \frac{dy_a}{dt} + y_a) \quad (10)$$

$$\frac{dy_m}{dt} = (K_m y - y_m) / \tau_m \quad (14)$$

$$\frac{du_m}{dt} = (K_{m1} u - u_m) / \tau_{m1} \quad (15)$$

IMPORTANT NOTE: Time delays are not represented explicitly in any of the equations above. However it is a feature of MIMOSA to assign automatically such delays to all inputs and outputs. The initial values for such parameters are given by the user in the .D01 file.

4. User-defined files:

The IR2L2.B file:

```
/* inverse response process (case II) & 2nd order process
load Antonio Ferreira, Dez. 91 */
```

```
/* Parameters definition */
p[2]=p[13];
p[3]=sqrt(p[15]);
p[4]=1/(2.*p[3]);
p[5]=(p[13]*p[15]-p[14])/p[13];
```

```
/* Valve */
if(p[1]<1e-3)
{
    y[0]=p[0]*c[0];
    f[0]=0.;
}
else f[0]=(p[0]*c[0]-y[0])/p[1];
```

```
/* Process */
f[1]=y[2];
f[2]=(y[0]-2*p[4]*p[3]*y[2])/p[3]*p[3];
y[3]=p[2]*(p[5]*f[1]+y[1]);
f[3]=0.;
```

```
/* Process load */
```

```
f[4]=y[5];
f[5]=(p[6]*ud[0]-y[4]-2.*p[8]*p[7]*y[5])/p[7]*p[7];
```

```
/* Output variable */
p[16]=y[3]+y[4];
```

```
/* Feedback control sensor */
```

```
if(p[10]<1e-3)
{
    y[6]=p[9]*p[16];
    f[6]=0.;
}
else f[6]=(p[9]*p[16]-y[6])/p[10];
```

```
/* Feedforward control sensor */
```

```
if(p[12]<1e-3)
{
    y[7]=p[11]*u[0];
    f[7]=0.;
}
else f[7]=(p[11]*u[0]-y[7])/p[12];
```

The IR2L2.D01 file:

```

C
C Initial conditions for the solution of system IR2L2
C
C N. of differential equations
8
C
C      Starting      Variances      Measuring
C Name  Value      Min Max      to noise      Delays
C
  x      50.        0. 100.        3.        100
  ya(aux) 50.        0. 100.
  yb(aux) 0.         0. 100.
  y1      50.        0. 100.
  y2      50.        0. 100.
  yc(aux) 50.        0. 100.
  ym      50.        0. 100.
  um      50.        0. 100.
C
C
C N. of loads/disturbances
1
C
C      Starting      Load Delay
C Name  Value      ( sec/100 )
  Load   50.        100
C
C N. of control actions
1
C

```

```

C      Starting      Control Delay
C Name  Value      ( sec/100 )
  Control 50.        40
C
C N. of parameters
17
C
C      Starting
C Name  Value      Comments
  Kv      1.        p0
  Tv      1.        p1
  Kp      1.        p2
  Tp      1.        p3
  Csi_p    1.        p4
  T1      1.        p5
  Kld      1.        p6
  Tld      1.        p7
  Csi_ld    1.        p8
  Km       .5       p9
  Tm       1.       p10
  Km1      1.       p11
  Tm1      1.       p12
  Ka       .01      Note: for inverse response  $K_a \cdot T_b < K_b$ 
  Kb       5.       p14
  Tb       1.       p15
  y       100.      p16

```

1. Problem description:

This problem has been proposed by Davison [1], [2]. It is desired to simulate a binary distillation column with pressure variation. The system is multivariable, with 3 inputs and 3 outputs and includes one disturbance input.

For a binary distillation column with n plates the proposed linearized model is represented, in matrix form, by:

$$\frac{dx}{dt} = A x + B u + E w \quad (1)$$

$$y = C x \quad (2)$$

where x is the system state vector, u is the vector of control inputs, w is the vector of disturbances to the process and y represents the vector of outputs of the system which can be measured. A , B , E and C are constants coefficient matrices. The latter is the matrix of system observability.

In the present case, for a column with 8 plates, we have: $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, $u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ and $w = w_1$,

where:

- F is the feed flow,
- L_c and L_{b+1} are the liquid streams on the top and bottom of the column, respectively,
- P_A and P_B are the top and bottom products, respectively,
- Q_1 is the heat transferred to the reboiler,
- Q_2 is the heat removed from the condenser,
- u_1 is the reboiler steam temperature,
- u_2 is the condenser coolant temperature,
- u_3 is the controlled reflux,
- V_{c-1} and V_b are the vapour streams in the top and bottom of the column, respectively,
- x_1 represents the composition of more volatile component in condenser,
- x_2 through x_9 are the compositions of more volatile components in plates 1 through 8,
- x_{10} is the composition of more volatile component in reboiler,
- x_{11} is the pressure,
- w_1 is the change of input feed concentration,
- y_1 is the composition of more volatile component in plate 8,
- y_2 is the composition of more volatile component in top product,
- y_3 represents the pressure.

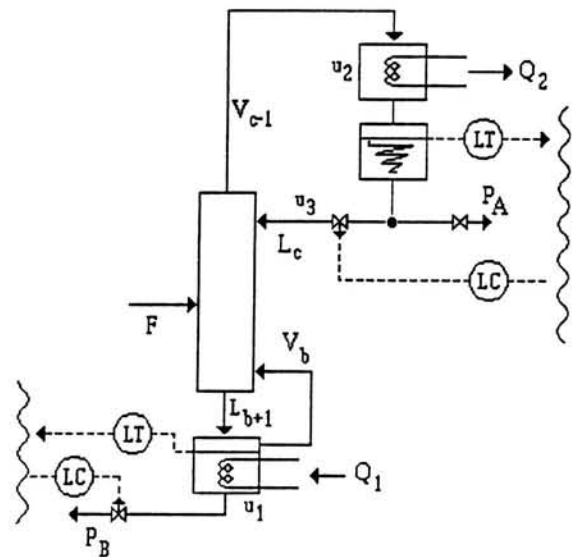


FIG. 1 - Control of a distillation column with pressure variation.

The present system is subject to the constraints:

$$\begin{aligned} \text{for } t > 0, \quad & |u_1| < 2.5 \\ & |u_2| < 2.5 \\ & |u_3| < 0.30 \\ & |w_1| < 1 \end{aligned}$$

IMPORTANT NOTE: Time delays are not represented explicitly in any of the equations above. However it is a feature of MIMOSA to assign automatically such delays to all inputs and outputs. The initial values for such parameters are given by the user in the .D01 file.

Three important comments should be made concerning the system above:

- (i) All variables are deviation variables;
- (ii) Direct readings, i.e. without dynamics and with unit gain are assumed for the three measured variables
- (iii) It is clear that in control practice flowrates rather than temperatures are the manipulated variables.
According to Davison [2] the present model only differs from the 'realistic' model by a constant factor which is included in the matrix coefficients. As such, we decided to keep the problem in the precise terms of the original work.

2. Data

Matrix A

$$\begin{bmatrix} -1.40\text{e-}2 & 4.30\text{e-}3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9.50\text{e-}3 & -1.38\text{e-}2 & 4.60\text{e-}3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.00\text{e-}4 \\ 0 & 9.50\text{e-}3 & -1.41\text{e-}2 & 6.30\text{e-}3 & 0 & 0 & 0 & 0 & 0 & 0 & 2.00\text{e-}4 \\ 0 & 0 & 9.50\text{e-}3 & -1.58\text{e-}2 & 1.10\text{e-}2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.50\text{e-}3 & -3.12\text{e-}2 & 1.50\text{e-}2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.02\text{e-}2 & -3.52\text{e-}2 & 2.20\text{e-}2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.02\text{e-}2 & -4.22\text{e-}2 & 2.80\text{e-}2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.02\text{e-}2 & -4.82\text{e-}2 & 3.70\text{e-}2 & 0 & 2.00\text{e-}2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.02\text{e-}2 & -5.72\text{e-}2 & 4.20\text{e-}2 & 5.00\text{e-}2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.02\text{e-}2 & -4.83\text{e-}2 & 5.00\text{e-}4 \\ 2.55\text{e-}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.55\text{e-}2 & -1.85\text{e-}2 \end{bmatrix}$$

Matrix B

$$\begin{bmatrix} 0 & 0 & 0 \\ 5.0\text{e-}6 & -4.0\text{e-}5 & 2.5\text{e-}3 \\ 2.0\text{e-}6 & -2.0\text{e-}5 & 5.0\text{e-}3 \\ 1.0\text{e-}6 & -1.0\text{e-}5 & 5.0\text{e-}3 \\ 0 & 0 & 5.0\text{e-}3 \\ 0 & 0 & 5.0\text{e-}3 \\ -5.0\text{e-}6 & 1.0\text{e-}5 & 5.0\text{e-}3 \\ -1.0\text{e-}5 & 3.0\text{e-}6 & 5.0\text{e-}3 \\ -4.0\text{e-}5 & 5.0\text{e-}6 & 2.5\text{e-}3 \\ -2.0\text{e-}5 & 2.0\text{e-}6 & 2.5\text{e-}3 \\ 4.6\text{e-}4 & 4.6\text{e-}4 & 0 \end{bmatrix}$$

Matrix E

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1.0\text{e-}2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Matrix C

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Translation to MIMOSA's nomenclature:

According to the rules of MIMOSA the equivalence of variables is as follows:

Variables in the model	Variables in MIMOSA	Variables in the model	Variables in MIMOSA
u_1	$c[0]$	x_7	$y[6]$
u_2	$c[1]$	x_8	$y[7]$
u_3	$c[2]$	x_9	$y[8]$
x_1	$y[0]$	x_{10}	$y[9]$
x_2	$y[1]$	x_{11}	$y[10]$
x_3	$y[2]$	y_1	$y[0]$
x_4	$y[3]$	y_2	$y[7]$
x_5	$y[4]$	y_3	$y[10]$
x_6	$y[5]$	w_1	$u[0]$

4. User-defined files:

The DISTCOL.B file:

```
/* Binary Distillation Column
Antonio Ferreira, Fev. 92
*/
```

```
int dum;
```

```
/* Restrictions */
```

```
c[0]=(c[0]-50)*5./100.;
if(c[0]>2.5) c[0]=2.5;
if(c[0]<-2.5) c[0]=-2.5;
c[1]=(c[1]-50)*5./100.;
if(c[1]>2.5) c[1]=2.5;
if(c[1]<-2.5) c[1]=-2.5;
c[2]=(c[2]-50)*5./100.;
if(c[2]>2.5) c[2]=2.5;
if(c[2]<-2.5) c[2]=-2.5;
if(u[0]>1.) u[0]=1.;
if(u[0]<-1.) u[0]=-1.;
```

```
/* Process */
```

```
f[0]=-1.4e-2*y[0]+4.3e-3*y[1];
f[1]=9.5e-3*y[0]-1.38e-2*y[1]+4.6e-3*y[2]+5e-4*y[10]
+5e-6*C[0]-4e-5*C[1]+2.5e-3*C[2];
f[2]=9.5e-3*y[1]-1.41e-2*y[2]+6.3e-3*y[3]+2e-4*y[10]
+2e-6*C[0]-2e-5*C[1]+5e-3*C[2];
f[3]=9.5e-3*y[2]-1.58e-2*y[3]+1.1e-2*y[4]+1e-6*C[0]
-1e-5*C[1]+5e-3*C[2];
f[4]=9.5e-3*y[3]-3.12e-2*y[4]+1.5e-2*y[5]+5e-3*C[2]
+1e-2*ud[0];
f[5]=2.02e-2*y[4]-3.52e-2*y[5]+2.2e-2*y[6]+5e-3*C[2];
f[6]=2.02e-2*y[5]-4.22e-2*y[6]+2.8e-2*y[7]-5e-6*C[0]
+1e-5*C[1]+5e-3*C[2];
f[7]=2.02e-2*y[6]-4.82e-2*y[7]+3.7e-2*y[8]+2e-4*y[10]-
1e-5*C[0]+0.3e-5*C[1]+5e-3*C[2];
```

```
f[8]=2.02e-2*y[7]-5.72e-2*y[8]+4.2e-2*y[9]+5e-4*y[10]
-4e-5*C[0]+5e-6*C[1]+2.5e-3*C[2];
f[9]=2.02e-2*y[8]-4.83e-2*y[9]+5e-4*y[10]-2e-5*C[0]
+2e-6*C[1]+2.5e-3*C[2];
f[10]=2.55e-2*y[0]+2.55e-2*y[9]-1.85e-2*y[10]
+4.6e-4*C[0]+4.6e-4*C[1];
```

```
/* Auxiliary variables */
```

```
y[11]=c[0]*40.;
f[11]=0.;
y[12]=c[1]*40.;
f[12]=0.;
y[13]=c[2]*1000./3.;
f[13]=0.;
```

```
/* Compositions */
```

```
dum=0;
do
{
    if((y[dum]>1.) && (f[dum]>0.));
    {
        y[dum]=1.;
        f[dum]=0.;
    }
    if((y[dum]<0.) && (f[dum]<0.));
    {
        y[dum]=0.;
        f[dum]=0.;
    }
}
while (dum++<10);
```

The DISTCOL.D01 file:

```

C
C Initial conditions for the solution of system DISTCOL
C
C N. of differential equations
  15
C
C Starting      Variances Measuring
C Name Value   Min Max   ( to noise )   Delays
C
x(cond)  0.    -1.  1.      0.  0
x1       0.    -1.  1.      0.  0
x2       0.
x3       0.
x4       0.
x5       0.
x6       0.
x7       0.
x8       0.    -1.  1.      0.  0
x(reeb)  0.
P0.      0.     1.  0.      0
Contr-1  4.     0. 100.     0.  0
Contr-2  4.     0. 100.     0.  0
Contr-3  50     0. 100.     0.  0

```

```

C
C N. of loads/disturbances
  1
C
C Starting      Process Intrinsic Delay
C Name Value    ( sec/100 )
W       0.5      0
C
C N. of control actions
  3
C
C Starting      Control Delay
C Name Value    ( sec/100 )
T(Reeb)  1.      0
T(Cond)  1.      0
R        .15
C
C N. of parameters
  0
C
C Starting
C Name Value
C

```

5. References:

- [1] Davison, E. J., "Control of a distillation column with pressure variation",
Trans. Institute of Chemical Engineers,
vol. 45, 1967,
pp T229-250
- [2] Davidson, E. J., "Benchmark Problems for Control System Design",
International Federation of Automatic Control,
Problem #90-01,
May 1990



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000101478