

Trabalho de Graduação

Desenvolvimento e Implementação de Estratégias de Controle para a Etapa Fermentativa da Produção da Vitamina C.

Metodologia: Controle Linear Quadrático Gaussiano Associado à um Modelo tratado pelo Método das Perturbações Singulares.

Gerson Obara
Maurício Nogueira Soufen

Orientador: Prof. Dr. Agenor de Toledo Fleury

Dedicatória

Dedicamos o presente trabalho a nossas queridas mães.

Agradecimentos

Gerson, Maurício

Sashimis da Elea, Tatiana

Saco do Fabio, Freddie Mercury, Beethoven, PiCi, Ivan, Flash, Laurinda, Luciana, Leninha, Esperidião, Palmeirense, Yuri, Júnior, Cassio, Luís Fabiano, Maria - Regiane, Bigode, a Pracinha, Marcia, Luciano Meu, Luís Henrique do IPT, Mannes, Vladimir, Pessoal da EngSci, Ana Flávia, Simone, Fernanda, Cristina...

Índice

Resumo.....	3
Introdução.....	4
Embasamento Teórico.....	6
A Planta - Modelagem e Problemas.....	8
Método das Perturbações Singulares.....	22
Controlador Linear Quadrático Gaussiano.....	25
Controle Linear Quadrático - Resultados.....	29
Implementação - Projeto e Simulação.....	44
Resultados.....	97
Discussão.....	158
Conclusão.....	160
Bibliografia.....	161

Resumo

Esta dissertação registra trabalho realizado em conjunto ao agrupamento de Controle do Instituto de Pesquisa Tecnológico no sentido de se conhecer melhor métodos de controle multivariável no tempo. Assim, pesquisou-se a ação da teoria de Controle Linear Quadrático Gaussiano em uma planta de fermentação para produção de Vitamina C. Este trabalho dá, então continuidade ao já realizado no IPT, onde já se havia experimentado a utilização do Controle Linear Quadrático nessa mesma planta.

Ao longo da pesquisa o contato com esse tipo de teoria submetida ao controle de processo fermentativos revelou muitas dificuldades, devido a complexibilidade da planta. Alguns progressos foram feitos no sentido de se conhecer melhor a reação da planta frente a diversas estratégias adotadas.

Para a obtenção dos resultados se aproveitou uma estrutura já montada pelo IPT, adaptando-a para essa nova situação. O ferramental utilizado são softwares desenvolvidos em Fortran e MatLab.

Neste relatório os principais resultados e conclusões encontrados são descritos de uma forma resumida, onde se poderá perceber a dificuldade de se analisar resultados obtidos sobre um tipo de controle e de planta extremamente específicos e mal conhecidos. Esse é apenas o começo de uma pesquisa, onde se tem muito a descobrir.

Introdução

A proposta do projeto que é relatado neste trabalho é conseguir resultados satisfatórios para o controle de uma planta de produção de Vitamina C por processos fermentativos, modelada pelo Método das Perturbações Singulares. A estratégia inicial é a implementação de um controlador - observador Linear Quadrático Gaussiano.

Faz-se então uma breve descrição sobre a planta e a modelagem inicial dos processos químicos envolvidos na fermentação. As equações que dão origem ao modelo adotado, estão apresentadas em trabalho realizado sobre processos fermentativos pelo Agrupamento de Biotecnologia da Divisão de Química do IPT.

O passo seguinte trata-se de mostrar os conceitos de controle multivariável envolvidos, tais como o Método das Perturbações Singulares e o Controle Linear Quadrático Gaussiano. Nesta etapa é importante, para o leitor, ter disponível a bibliografia do ramo para uma total compreensão da teoria envolvida.

Faz parte deste relatório, ainda, uma breve descrição sobre o projeto propriamente dito e a simulação do controlador - observador. São registrados, então, os principais resultados obtidos e as principais estratégias adotadas no sentido de demonstrar a reação da planta de processos fermentativos frente a mudanças no controle aplicado. Não se chegou a uma conclusão definitiva sobre a reação desse tipo de sistema, como poderá se verificar ao longo deste relatório.

O leitor deve compreender a motivação do desenvolvimento e implementação de tais métodos de controle, isso se deve ao fato de processos fermentativos serem de difícil medição, seu modelo ser não linear, multivariável e de parâmetros mal conhecidos. Assim compreende-se o interesse pela área, pois controlar tais processos, não só melhoraria o rendimento da produção de Vitamina C, quanto da maioria dos processos fermentativos. Assim, existe ainda muito a se pesquisar nesta área, até que se chegue a resultados que possam ser utilizados industrialmente.

O leitor deve compreender também, a evolução que esta se dando a essa pesquisa e o porque da escolha das teorias de controle adotadas. Dentre as diversas abordagens que poderia ser dada ao projeto do controlador preferiu-se, inicialmente, recorrer ao modelo Linear Quadrático por ser uma teoria que se tem um relativo conhecimento o que permite maior certeza dos rumos que esta se dando ao projeto.

Frente às dificuldades apresentadas pela planta, a adoção de conceitos mais robustos de controle se tornou necessária, dentre as diversas opções, preferiu-se a inclusão de um observador probabilístico, ou seja adoção de um controlador Linear Quadrático Gaussiano, baseado em um modelo de Perturbações Singulares.

Outra dificuldade que pode ser citada no controle de tal tipo de planta, se deve ao fato de esta apresentar, após a linearização em torno do ponto ótimo, variáveis muito rápidas e outras muito lentas. A esta dificuldade a solução adotada é o tratamento do modelo que é proposto, através do Método das Perturbações Singulares. A idéia básica é dividirmos o modelo em dois subsistemas, um rápido e outro lento e após este desacoplamento tentarmos obter melhores resultados no controle.

A escolha da metodologia Gaussiana se deve ao citada fato do excesso de incertezas em nosso modelo e da dificuldade encontrada nas medições das variáveis envolvidas no processo.

Ao final deste trabalho se disserta a respeito das conclusões que se chegou, principais soluções encontradas e o grau de satisfação dos objetivos propostos a que se chegou.

A abordagem dada neste relatório diz respeito aos aspectos de controle, assim não se dará ênfase a aspectos químicos envolvidos no processo. Maiores informações sobre qualquer assunto discorrido neste trabalho podem ser encontrados facilmente na bibliografia indicada.

Embasamento Teórico

Abaixo estão citados alguns conceitos de controle importantes para compreensão do problema.

A partir do modelo proposto, que não está na forma linear necessária para projetarmos o controlador, o primeiro passo que deve ser seguido, se refere a linearização de tal modelo em torno de um ponto que nos é conveniente.

- Linearização

A linearização nada mais é que a expansão em Taylor das equações que temos, em torno de um ponto escolhido.

- Cálculo da Controlabilidade

Um sistema é controlável em $t = t_0$ se for possível construir um vetor de controle, $u(t)$ que transfira o sistema do estado inicial $x(t_0)$ para qualquer estado final $x(t_f)$ em um intervalo finito de tempo $t_0 \leq t \leq t_f$. Se qualquer estado inicial $x(t_0)$ é controlável, então o sistema é dito completamente controlável.

Praticamente, para se verificar a controlabilidade, deve-se seguir a seguinte seqüência:

Para o sistema:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

onde x é o vetor de variáveis de estado, $n \times 1$;

u é o vetor de entradas de controle, $m \times 1$;

y é o vetor de medidas (números), $r \times 1$;

A , B e C são matrizes de parâmetros do sistema, $n \times n$, $n \times m$,
 $r \times n$.

tem-se a seguinte matriz de controlabilidade ($n \times m$):

$$\tau = [B: AB: A^2B: \dots: A^{n-1}B]$$

Caso tal matriz tenha posto (rank) n , o que significa que é possível extrair de τ uma matriz $n \times n$ com determinante não nulo, então o sistema é controlável.

Para se ter uma visão mais prática do que se refere a controlabilidade pode-se fazer a seguinte imagem. As variáveis de estado da planta que estão propostas no modelo, fornecem condições para que se faça com que a planta atinja um estado desejado. As variáveis que não estão expressas neste modelo não impedem a controlabilidade.

É importante frisar que o sistema em questão se mostrou controlável.

- Observabilidade

Para o mesmo sistema proposto ao definirmos controlabilidade, podemos dizer que o mesmo é completamente observável se todo o estado $x(t)$ pode ser obtido a partir das observações $y(t)$ ao longo de um intervalo finito de tempo $t_0 \leq t \leq t_f$. Em outras palavras, o sistema é completamente observável se todos os estados afetam cada elemento do vetor de saídas.

Para verificarmos a observabilidade temos que o vetor $(m \times n)$:

$$\sigma = [C^T : A^T C^T : \dots : (A)^{n-1} C^T]$$

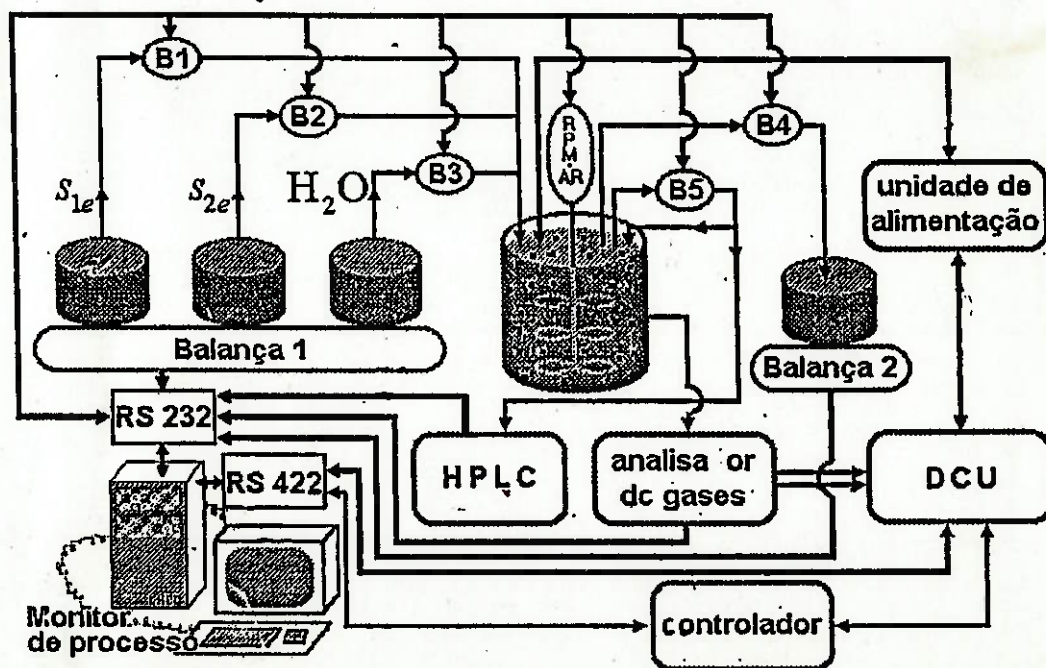
Frisamos que nosso modelo original, após a linearização, não é observável o que mais uma vez sugere que alguma alteração no modelo deve ser feita para que possamos obter bons resultados.

A Planta - Modelagem e Problemas

O esquema de controle proposto obedece aos requisitos mais atuais quanto à sua concepção: é baseado em modelo matemático (descrito a seguir), multivariável e implementado em microcomputador. Os controladores PID disponíveis no equipamento devem ser usados apenas para cuidar de variáveis importantes não incluídas no modelo, como temperatura e pH, admitidos constantes.

O esquema básico de automação e controle da etapa fermentativa do processo de produção de vitamina C é mostrado na figura abaixo. O fermentador Braun, que incorpora a DCU (Digital Control Unit) e a unidade de alimentação, é o centro do sistema. O sistema de agitação e ajuste de vazão de ar que fornece a variável de controle $K_L a$, taxa de transferência de oxigênio, não está representado na figura. As outras variáveis de controle são obtidas pelas bombas B1, B2, B3. O sistema de medidas é composto por uma sonda de oxigênio e por um cromatógrafo líquido de alto desempenho (HPLC). O analisador de gases, incluído na malha e já operacional, será utilizado para fornecer medidas de concentração celular no futuro.

A medida da concentração de O_2 dissolvido é obtido em ciclos de 15 s. Do cromatógrafo são obtidas com intervalos de 20 min., medidas de concentração de sorbitol e sorbose. Estes intervalos de amostragem, somados às dificuldades impostas pelos poucos sensores disponíveis, trazem grandes dificuldades ao projeto do controlador e faz com que novas alternativas tenham que ser pesquisadas.



O Modelo Matemático

O processo de transformação de sorbitol em sorbose, validado experimentalmente, pode ser descrito pelo seguinte sistema de equações diferenciais não lineares:

$$\dot{x} = D \cdot (-x\mu) + \mu \cdot x$$

$$\dot{p} = D \cdot (-p) + [\alpha \cdot x + \beta] \cdot x$$

$$\dot{s}_1 = D \cdot (s_{1a} - s_1) + \left[\left(-\frac{1}{Y_{xs1}} - \frac{\alpha}{Y_{p1}} \right) \cdot \mu + \left(-\frac{1}{Y_{ps1}} \right) \cdot \beta \right] \cdot x$$

$$\dot{s}_2 = D \cdot (s_{2a} - s_2) + \left[\left(-\frac{1}{Y_{xs2}} \right) \cdot \mu \right] \cdot x$$

$$\dot{o} = D \cdot (c_g - o) + \left[\left(-\frac{1}{Y_{xo}} - \frac{\alpha}{Y_{po}} \right) \cdot \mu + \left(-\frac{1}{Y_{po}} \right) \cdot \beta \right] \cdot x + K_1 a \cdot (c_g - o)$$

com:

$$\mu = \mu_0 \cdot s_1 / \left(K_{s1} + s_1 + \frac{s_1^2}{w} \right) \cdot s_2 / (K_{s2} + s_2) \cdot o / (K_o + o) \cdot e^{-K_p \cdot p}$$

$$\beta = \beta_0 \cdot s_1 / (K_{\beta s1} + s_1) \cdot o / (K_{\beta o} + o)$$

e onde se tem as seguintes variáveis de estado:

x concentração de bactérias na solução [g/l];

p concentração de sorbitol (produto) dissolvido [g/l];

s_1 concentração de sorbose dissolvida [g/l];

s_2 concentração de extrato de levedura [g/l];

o concentração de oxigênio dissolvido [mg/l];

μ representa a velocidade específica de crescimento celular [h^{-1}], e β a velocidade específica de produção de sorbose não associada ao crescimento celular [h^{-1}].

Como variáveis de controle pode-se dispor de:

D vazão específica [h^{-1}];

$K_1 a$ coeficiente de transferência de oxigênio [h^{-1}];

s_{1a} concentração de sorbitol na alimentação [g/l];

s_{2a} concentração de extrato de levedura na alimentação [g/l].

Os parâmetros do processo são estequiométricos ou ajustados experimentalmente:

α parâmetro de formação de sorbose associado ao crescimento [18.6 g/g];

Y_{xs1} fator de conversão sorbitol célula [0.795 g/g];

Y_{ps1} fator estequiométrico de conversão sorbitol/sorbose [0.989 g/g];
 Y_{xs2} fator de conversão levedura/célula [0.656 g/g];
 c_g oxigênio dissolvido em equilíbrio [7.50 mg/l];
 Y_{xo} fator de conversão oxigênio/célula [0.000664 g/mg];
 Y_{po} fator estequiométrico de conversão oxigênio/sorbose [0.01125 g/mg];
 μ_0 velocidade específica máxima de crescimento celular sem inibição pelo substrato [0.664 h⁻¹];
 K_{s1} parâmetro de saturação do crescimento pelo sorbitol [3.93 g/l];
 W parâmetro de inibição de crescimento pelo sorbitol [204.0 g/l];
 K_{s2} parâmetro de saturação do crescimento pela levedura [0.994 g/l];
 K_o parâmetro de saturação do crescimento pelo oxigênio [0.45 mg/l];
 K_p parâmetro de inibição exponencial do crescimento pelo produto [0.00554 l/g];
 β_0 velocidade específica máxima de produção de sorbose não associada ao crescimento [g/g-h];
 $K_{\beta s1}$ parâmetro de saturação da produção de sorbose não associada ao crescimento [4.91 g/l];
 $K_{\beta o}$ parâmetro de saturação por oxigênio da produção de sorbose não associada ao crescimento [0.544 mg/l].

Variáveis importantes no processo, como pH e temperatura, são controladas por dispositivos PID disponíveis nos fermentadores, mas fora da malha central descrita.

O objetivo deste trabalho é estabelecer um procedimento de controle em torno de um ponto de operação (equilíbrio) para o sistema descrito, de modo que permaneça próximo a este ponto com um mínimo de gasto de controle.

O ponto de equilíbrio é dado pela minimização de uma função objetivo F. O. que atenda a vários requisitos do processo:

$$FO = -w_1 \cdot D \cdot p / (s_{1a} - s_1) - w_3 \cdot p + w_4 \cdot s_1 + w_5 \cdot s_{21} + w_6 \cdot K_t a$$

onde os parâmetros w_i são ponderações:

$$w_1 = 30.0 \quad w_4 = 1.0$$

$$w_2 = 10.0 \quad w_5 = 30.0$$

$$w_3 = 1.2 \quad w_6 = 0.5$$

Assim a minimização de F.O. implica na maximização da velocidade de formação, rendimento e concentração final de produto, e a minimização de insumos inaproveitados e de gastos de aeração do processo. Os resultados obtidos pelo

algoritmo OPT3, que utiliza o método do gradiente reduzido vinculado à satisfação de vínculos algébricos, para o conjunto de pesos escolhidos, são os seguintes:

$$\begin{aligned}x_e &= 3.238 \quad o_e = 1.597 \quad s_{1ae} = 422.1 \\p_e &= 392.9 \quad \mu_e = D_e \quad s_{2ae} = 7.975 \\s_{1e} &= 20.79 \quad \beta_e = 1.123 \quad K_{1ae} = 269.6 \\s_{2e} &= 3.038 \quad D_e = 0.040\end{aligned}$$

Determinado o ponto de equilíbrio a aproximação linear do sistema em torno deste ponto fica na forma:

$$\begin{aligned}\delta \dot{x} &= A \cdot \delta x + B \cdot \delta u \\y &= C \cdot \delta x \\\delta x &\equiv x - x_e \\\delta u &\equiv u - u_e\end{aligned}$$

onde $x = [x \quad p \quad s_1 \quad s_2 \quad o]$ e $u = [D \quad s_{1a} \quad s_{2a} \quad K_1 a]$.

As matrizes A e B são obtidas da linearização do modelo no ponto de equilíbrio e C é definida em função das medidas dos estados disponíveis ($p, s_1, e o$).

O teste simples das características de controlabilidade e observabilidade do sistema antecipa problemas: o posto da matriz de observabilidade é 3, e o de controlabilidade é 5 para as condições estabelecidas. Tem-se portanto um sistema controlável mas não observável, assim não há como prever o comportamento do sistema: as variáveis não observáveis podem instabilizar o sistema sem que o controlador se dê conta do que está ocorrendo.

Isto está ligado fortemente a dois motivos: o primeiro é a grande diferença na ordem de grandeza dos autovalores. Enquanto quatro deles se situam em valores próximos a D ($\approx 10^2$). Esse motivo associado ao fato de não se dispor de medidas de todos os estados, implica na não observabilidade do sistema de forma direta.

Além disso a implementação de um controlador para este sistema é complicada pelo fato de as medidas estarem muito defasadas no tempo. O produto p e o substrato s_1 são medidos com atrasos de 20 min. e o oxigênio com atrasos de 15 s em vista da instrumentação disponível.

Um aspecto importante desta planta reside no fato de que se trata de um sistema estável, assim pode se verificar que mesmo sem controle ela converge para o ponto de equilíbrio, como mostra as simulações dispostas nas próximas páginas (tais simulações estão representadas através de gráficos cartesianos, onde no eixo horizontal está o tempo de simulação e no eixo vertical está a porcentagem de desvio em relação ao ponto de equilíbrio).

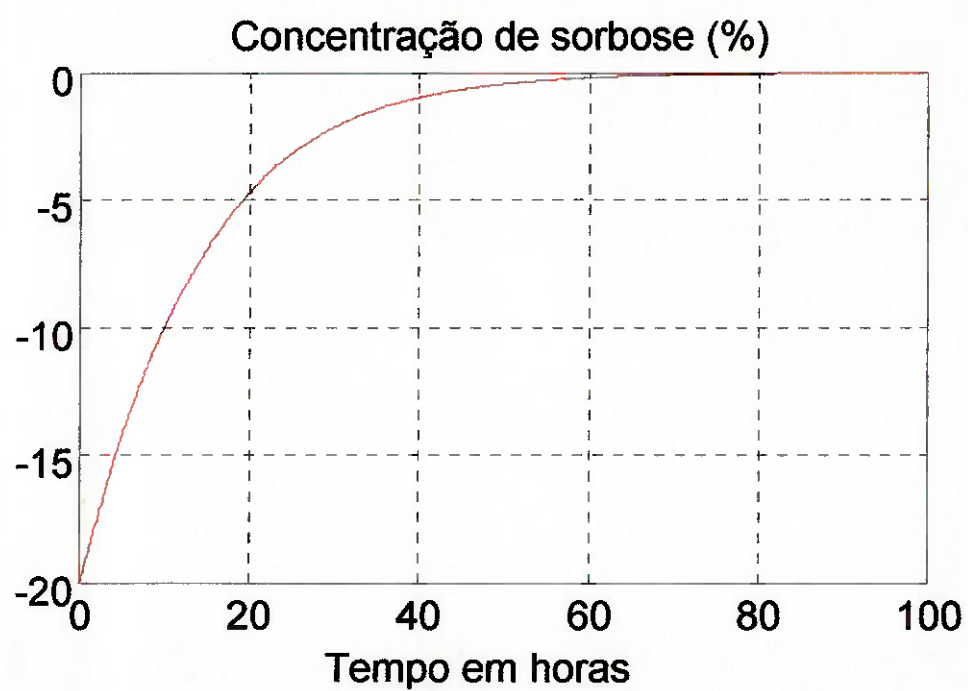
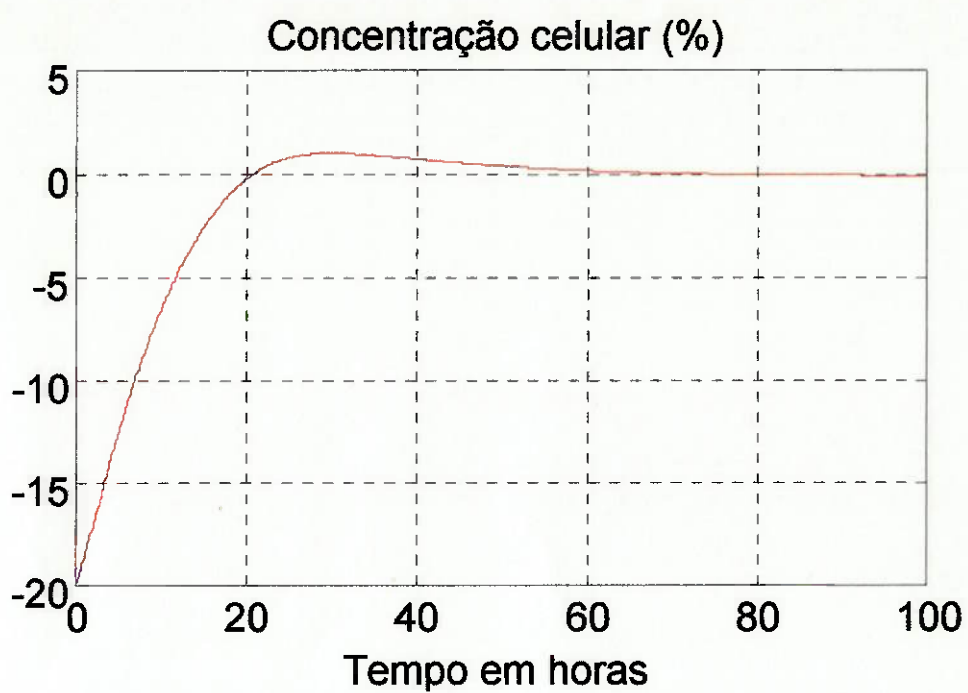
É importante frisar que estes gráficos são obtidos através de um modelo que ainda não inclui a presença de um ruído que é sabido que existe devido a imprecisões neste modelo e na instrumentação disponível.

O fato da malha ser estável é bastante relevante devido ao fato de que uma das estratégias de controle adotada é deixar a planta evoluir durante algumas horas sem controle para que ela atinja um ponto onde o controle seja facilitado.

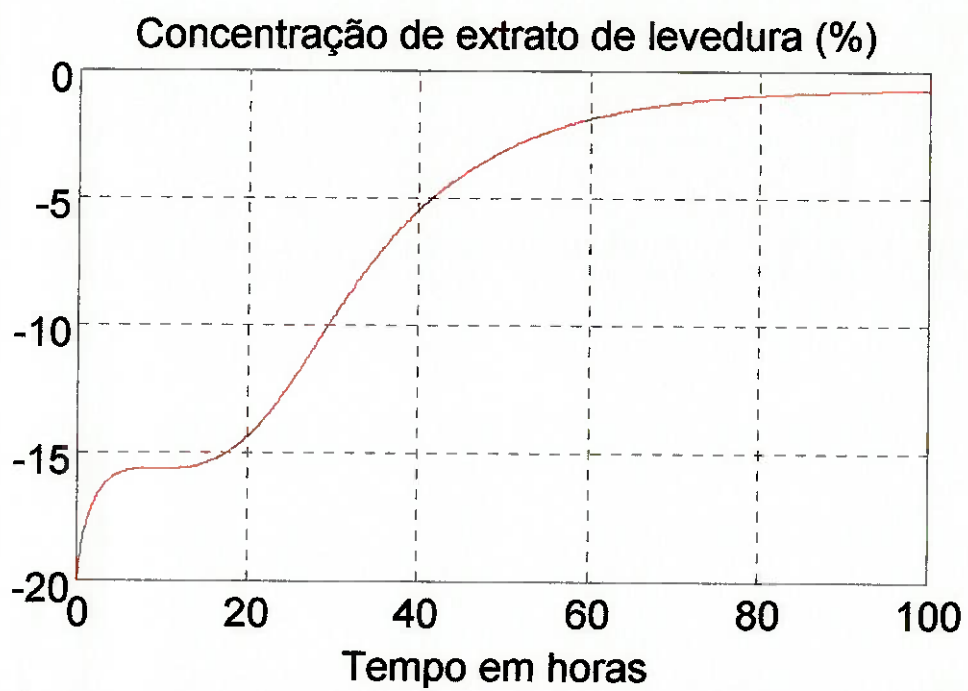
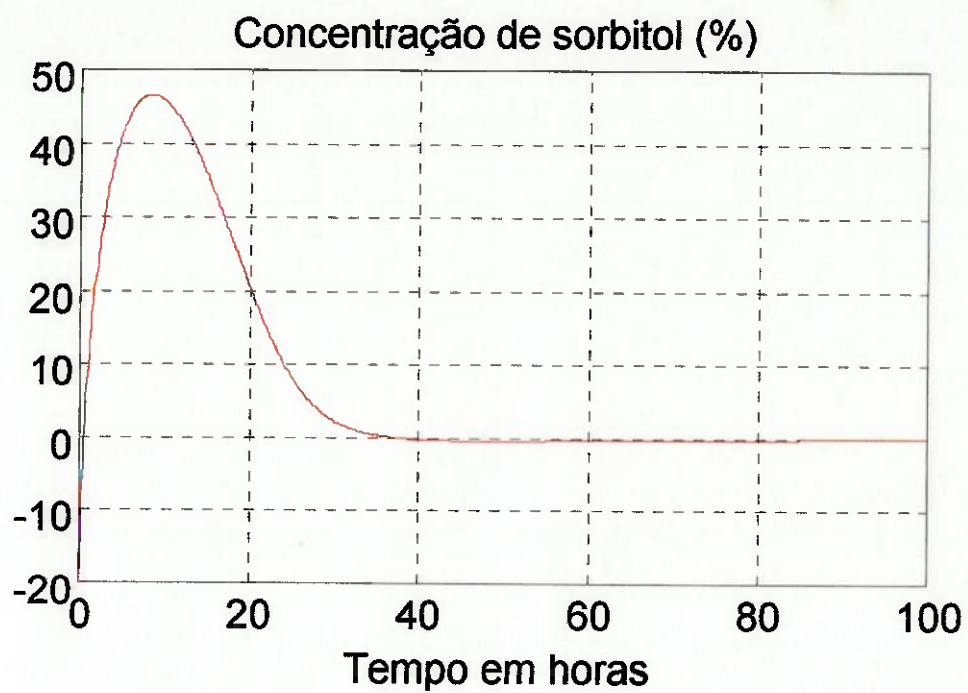
Uma análise superficial da simulação sem controle mostra que esse sistema converge para ponto de equilíbrio após 45 horas de simulação, a esta altura a maioria das variáveis já está em torno do ponto onde há 0 % de desvio. Outro fato a ser notado é o comportamento do oxigênio (Medida de Oxigênio) que é claramente mais “rápido” que o comportamento das demais variáveis.

Evolução da Planta Sem Controle

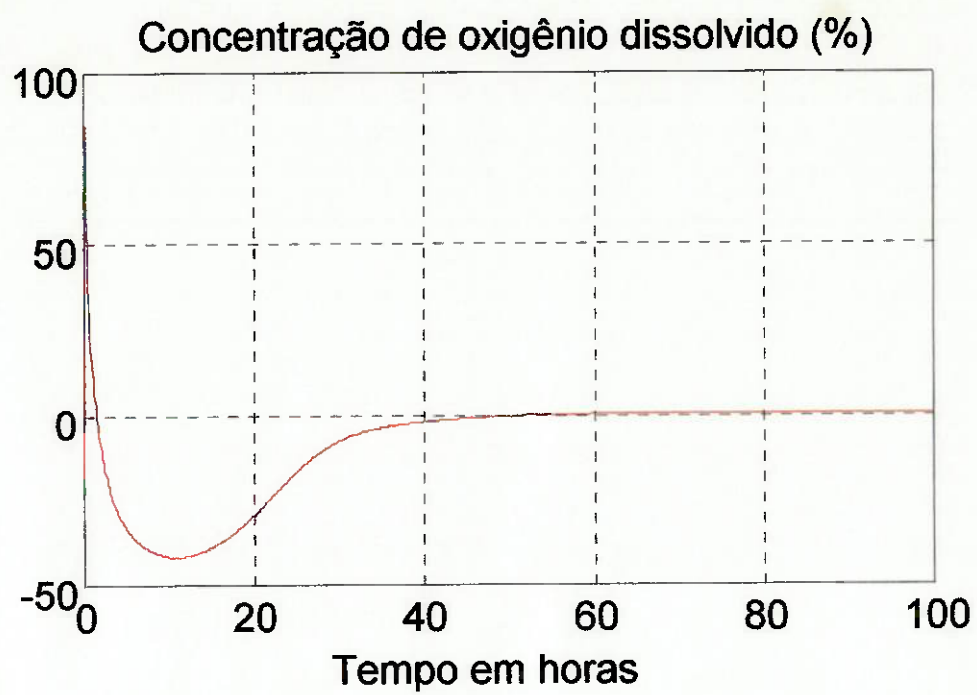
Variáveis de Estado:



Simulação sem controle.

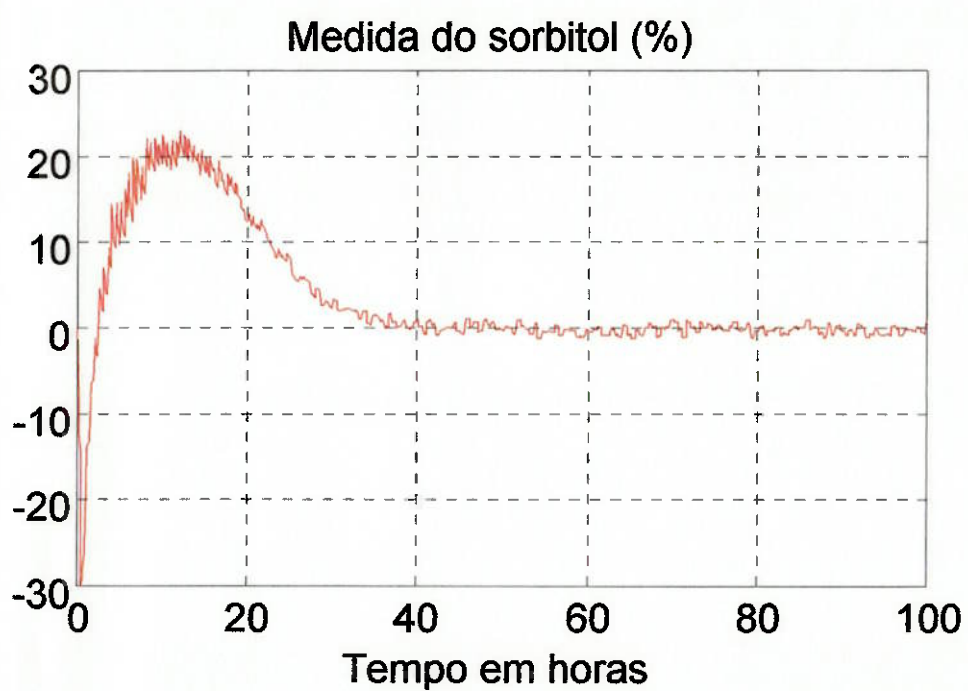
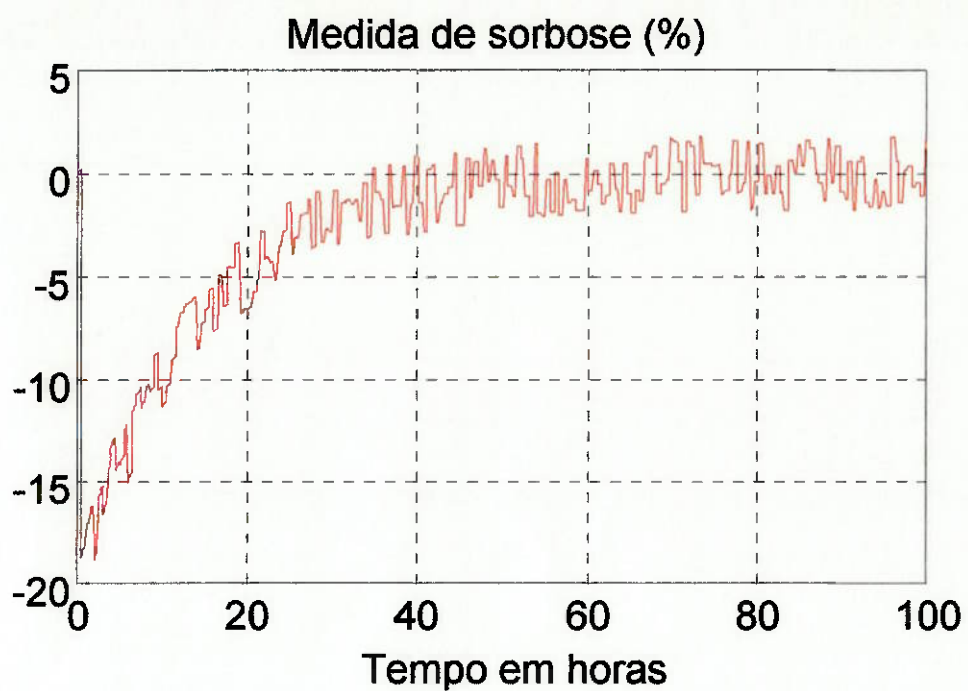


Simulação sem controle.

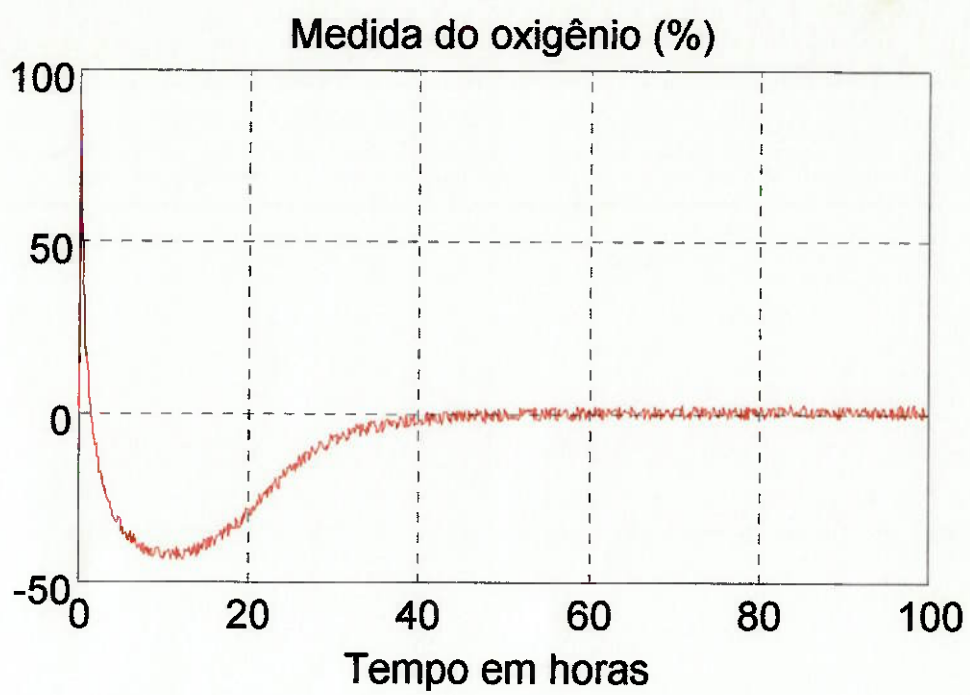


Simulação sem controle.

Variáveis Medidas:



Simulação sem controle.



Simulação sem controle.

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xe - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00 -.123794160E-02 .707919687E-01 .176010163E-01 .812277235E-01
.393306572E+01 -.115528924E+00 .359136491E+01 .505149169E+00 .558033286E+01
-.431723616E+01 .411919255E-01 -.401255173E+01 -.585665554E+00 -.598804914E+01
-.847457627E-01 .131137881E-02 -.749914922E-01 -.986451444E-01 -.860463173E-01
-.955666447E+03 .125364720E+02 -.855535230E+03 -.178243181E+03 -.143892113E+04

Matriz Fi

.999981172E+00 -.784903394E-06 .421348740E-04 .111597328E-04 .393510318E-04
.198542078E-02 .999920663E+00 .183598939E-02 .180172673E-03 .270530063E-02
-.221106331E-02 .161531388E-04 .997897628E+00 -.229664841E-03 -.290284112E-02
-.506743625E-04 .831465460E-06 -.446344004E-04 .999921514E+00 -.416854150E-04
-.463296555E+00 .608081778E-02 -.414739698E+00 -.864568839E-01 .302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01 .000000000E+00 .000000000E+00 .000000000E+00
-.116870000E+03 .000000000E+00 .000000000E+00 .000000000E+00
.128284000E+03 .800000000E-01 .000000000E+00 .000000000E+00
.251000000E+01 .000000000E+00 .800000000E-01 .000000000E+00
.693600000E+01 .000000000E+00 .000000000E+00 .693600000E+01

Matriz Gama

-.197813202E-02 .155751693E-08 .403827121E-09 .136024765E-06
-.972653142E-01 .716993845E-07 .819115086E-08 .934885760E-05
.106764081E+00 .665853199E-04 -.100006689E-07 -.100316676E-04
.208880533E-02 -.164991200E-08 .666640167E-04 -.144094031E-06
-.230530652E-01 -.165312790E-04 -.344534729E-05 .336908405E-02

Matriz C - Matriz de Medições de estados do modelo linearizado

.000000000E+00 .684521263E-01 .000000000E+00 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .466200466E+01 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .000000000E+00 .000000000E+00 .709219858E+00

Simulação sem controle.

Valor de dtO (em minutos)
.500000000E-01

Matriz Fimod

.998408956E+00	.161353277E-04	.688520478E+00	-.201173160E-03	-.168371751E-13
.546551183E-02	.999858945E+00	-.946268595E+00	.717784813E-03	.889796309E-13
-.167076841E-02	.129312546E-04	.165462509E+00	-.226657664E-03	-.131228805E-13
-.589977772E-03	.672747039E-05	.224734071E+00	.999847343E+00	-.640291789E-14
-.403489958E+00	.529604490E-02	-.360806729E+00	-.752980383E-01	.211647305E+00

Matriz Gamamod

.452972452E-01	.294831367E-04	-.493193159E-08	-.405262064E-16	.000000000E+00	-
.198619912E-05	-.689940279E+00	.000000000E+00	-.213914593E-02		
-.162410338E+00	-.405482411E-04	.214888946E-07	.293356861E-15	.000000000E+00	
.239070080E-04	.951295093E+00	.000000000E+00	.768516348E-02		
.496479757E-01	.309606583E-04	-.970570221E-08	-.719152253E-16	.000000000E+00	
.301050769E-05	.832984571E+00	.000000000E+00	-.231744786E-02		
.175244757E-01	.962412917E-05	.666621570E-04	-.165027682E-16	.000000000E+00	-
.679311360E-06	-.225279282E+00	.000000000E+00	-.792074690E-03		
-.211021288E-01	-.150387435E-04	-.313670867E-05	.293442728E-02	.000000000E+00	-
.681094289E-07	-.393993433E-03	.000000000E+00	.180712830E+00		

Dados iniciais para simulação (arquivo VIT_C.DAT)

-20.0	-20.0	-20.0	-20.0	-20.0	estados iniciais
0.00	0.00	0.00	0.00	0.00	observador inicial
0.00	tempo inicial (hora)		hoje 27/11/95 14h50m		
100.00	tempo final (hora)				
3.00	duração do filtro do atuador (min)				
0.00	atraso para início do controle (horas)				
400.00	atraso para melhoria da medição (horas)				
1.e-6	precisão				
1.0e-5	passo inicial (chute)				
1.e-8	menor passo admissível (hora)				
-1.0	tempo entre resultados armazenados (segundos)				
400000	número máximo de passos				
0	=1 simula-se até o máximo de passos; =0 não há limite				
1	=1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)				
2	=1 observador é o estado real; =2 obs discreto (despro)				
0	numero inteiro de dtO's para atraso dos contr na memorex (shift)				
0.0	erro máximo aleatório para observadores iniciais (%)				
360000.0	tempo de aleatorização dos parâmetros (min);				
50.0	50.0	50.0	50.0	valores máximos de controle (%)	
-80.0	-80.0	-80.0	-50.0	valores mínimos de controle (%)	
0.0	30.0	30.0	0.0	-0.25	atrasos das medidas (min)
1.0	1.0	1.0	1.0	1.0	fatores de ajuste da propagação
0.5	0.5	0.5	0.0	0.5	deadband (%) p/ atuadores(4) e
0.0	0.0	0.0	0.0	0.0	_μ0, _Ks1, _Ks2, _W, __ (%)
0.0	0.0	0.0	0.0	0.0	_B0, _Bs1, _BO, _KO, _Yxs1 (%)
0.0	0.0	0.0	0.0	0.0	_Yxs2, _Kp, _YxO, ruído, ruído

Simulação sem controle.

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

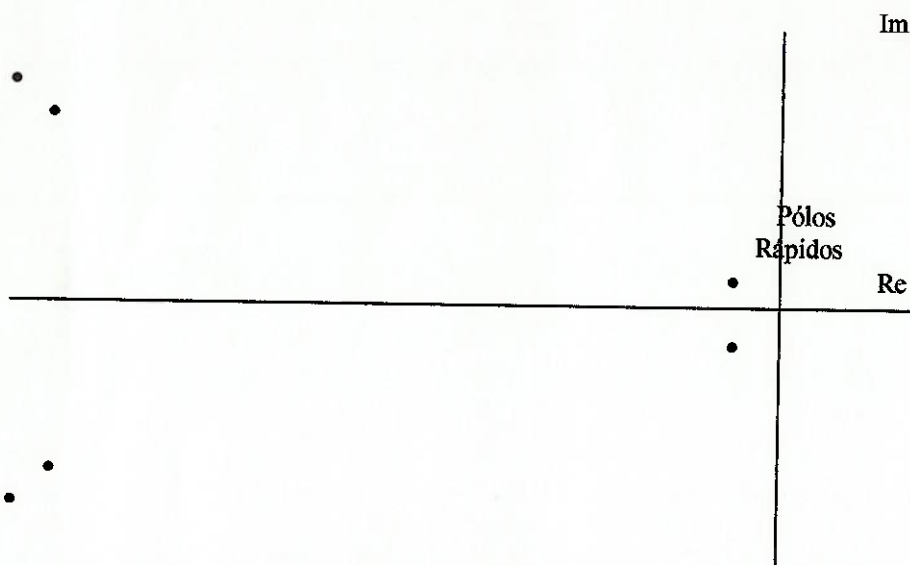
Simulação realizada em 0h04m26.01s.

Final normal da simulação

Simulação sem controle.

Método das Perturbações Singulares

Alguns sistemas apresentam uma configuração de pólos dispostos de uma certa forma que dão a esse sistema características distintas no que se refere a velocidade de suas variáveis de estado. Sistemas com indutâncias, inertâncias e capacitâncias multiplicando termos derivativos são comuns, mas se em particular os valores desses coeficientes são pequenos ou apresentam grande diferença de magnitude isso leva o modelo matemático a ter modos rápidos e lentos. A teoria do Método das Perturbações Singulares trata da separação desses modos para seu posterior tratamento.



Disposição dos pólos em sistemas com variáveis lentas e rápidas.

Sistemas com a configuração de pólos apresentada acima tem um comportamento de difícil controle pois os pólos rápidos agem sobre os lentos como se fossem ruído. Assim pretende-se modelar o sistema de forma a desacoplar as variáveis lentas das rápidas. No caso do nosso sistema como verificou-se a sua controlabilidade e a não observabilidade, o que leva a adoção deste tipo de teoria.

Basicamente o que se faz é tratar as variáveis rápidas como se fossem instantâneas em relação às lentas e tratar as lentas como se fossem constantes em relação às rápidas, como se segue.

Assim do sistema original:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Obtém-se um subsistema rápido

$$\varepsilon \cdot \dot{x}_2 = A_{21} \cdot x_1 + A_{22} \cdot x_2 + B_2 \cdot u$$

e um subsistema lento:

$$\dot{x}_1 = A_{11} \cdot x_1 + A_{12} \cdot x_2 + B_1 \cdot u$$

onde ε é um coeficiente que nos dá a ordem de grandeza entre a variável mais rápida e a mais lenta.

No caso do controlador implementado foi adotado ε como sendo a divisão entre os autovalores mais rápido e o mais lento.

No caso A_{11} , A_{12} , A_{21} , A_{22} , B_1 e B_2 estão definidos como abaixo.

$$A = \begin{bmatrix} A_{11} & \vdots & A_{12} \\ \dots & \vdots & \dots \\ A_{21} & \vdots & A_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 \\ \dots \\ B_2 \end{bmatrix}$$

$$\dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t)$$

$$\varepsilon \dot{x}_2(t) = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)$$

$$y(t) = C_1x_1(t) + C_2x_2(t)$$

Para montarmos o subsistema lento, procedemos da seguinte forma.

$$\dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t)$$

$$0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t)$$

$$y(t) = C_1x_1(t) + C_2x_2(t)$$

Fazemos $\varepsilon=0$ o que indica que temos um estado quasi-estático, disto temos:

$$\dot{x}_2(t) = -A_{22}^{-1}[A_{21}x_1(t) + B_2u(t)]$$

$$y_s(t) = C_0x_s(t) + D_0u_s(t)$$

onde

$$C_0 = C_1 - C_2A_{22}^{-1}A_{21}$$

$$D_0 = -C_2A_{22}^{-1}B_2$$

Para o sistema rápido temos:

$$x_1(t) = \text{const}$$

$$\dot{x}_f(t) = A_{22}x_f(t) + B_2u_f(t)$$

$$y_f(t) = C_2x_f(t)$$

Com isso desacoplamos as variáveis rápidas das lentas e temos dois subsistemas. Assim se projetará dois controladores um para cada subsistema, mas como na realidade a planta é uma só, é importante fazer uma composição criteriosa das leis de controle.

Isso resolve alguns problemas que a planta de processos fermentativos apresenta, no que se refere a diferença de escala de tempo de algumas variáveis, o sistema que não era observável após essa transformação se tornou observável. Mas ainda temos problemas de controle devido ao excesso de aproximações feitas durante a modelagem, o modelo original já incorpora aproximações, o que se acentua após a linearização.

Em resposta a isso é que se procura adotar novos conceitos de controle onde se incorpore um tratamento a resposta do sistema, como L.Q.G..

Controlador Linear-Quadrático Gaussiano

O controle Linear Quadrático Gaussiano tem como princípio básico a hipótese de que o controlador e o observador podem trabalhar independentemente, assim baseia-se no princípio da Separação para se começar a formular a teoria de projeto.

Tal teoria é inserida no contexto das teorias de controle a partir de modelos muito bem definidos, mesmo que não muito exatos, e é bom frisar que estes conceitos de controle só podem ser aplicados caso este modelo se mostre controlável e observável.

A literatura sobre o assunto cita que com a implantação deste tipo de controle deve-se chegar a margens de fase da ordem de 60° à 90° e margens de ganho que tendem a infinito.

A lei de controle que obteremos ao final do projeto será da seguinte forma:

$$u(x, t) = -k\hat{x}(t)$$

onde,

u - representa o sinal de controle;

$\hat{x}$ - representa o vetor das variáveis de estado estimadas pelo observador;

- Controle Linear Quadrático

A idéia básica deste tipo de controlador, que esta inserido dentro do contexto do controle ótimo, é minimizar o seguinte índice de desempenho:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

onde Q e R são os pesos definidos pelo projetista.

Assim o problema estará muito bem definido pois tem-se um modelo linear e uma função quadrática que se cruzam, deve-se, assim, achar o ponto de mínimo deste cruzamento.

Como o nosso sistema está subdividido em rápido e lento temos que minimizar dois índices de desempenho,

Para o sistema lento:

$$J = \frac{1}{2} \int_0^{\infty} (y_s^T y_s + u_s^T R u_s) dt, \quad R > 0$$

Para o sistema rápido:

$$J = \frac{1}{2} \int_0^{\infty} (y_f^T y_f + u_f^T R u_f) dt, \quad R > 0$$

onde os índices s e f se referem ao sistema lento e rápido, respectivamente.

Consideremos o seguinte sistema:

$$\begin{bmatrix} \dot{x} \\ \varepsilon \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \quad \begin{bmatrix} x(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} X_0 \\ Z_0 \end{bmatrix}$$

$$y = \begin{bmatrix} M_1 & M_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$$

Para se obter a lei de controle que satisfaça tal requisito usamos as equações de Riccati para o sistema lento e rápido.

$$0 = -K_s(A_0 - B_0 R_0^{-1} N_0 M_0) - (A_0 - B_0 R_0^{-1} N_0^T M_0)^T K_s$$

Para o sistema lento:

$$+ K_s B_0 R_0^{-1} B_0^T K_s - M_0^T (I - N_0 R_0^{-1} N_0^T) M_0$$

onde, M_0, A_0, B_0 - parâmetros do sistema lento

$$A_0 = A_{11} - A_{12} A_{22}^{-1} A_{21}$$

$$B_0 = B_1 - A_{12} A_{22}^{-1} B_2$$

$$M_0 = M_1 - M_2 A_{22}^{-1} A_{21}$$

$$N_0 = -M_2 A_{22}^{-1} B_2$$

$$R_0 = R + N_0^T N_0$$

Assim através desta equação de Riccati obtemos K_s que nos fornecerá a lei de controle para o sistema lento:

$$u_s = -R_0^{-1} (N_0^T M_0 + B_0^T K_s) x_s = G_0 x_s$$

Para o sistema rápido temos a seguinte equação de Riccati:

$$0 = -K_f A_{22} - A_{22}^T K_f + K_f B_2 R^{-1} B_2^T K_f - M_2^T M_2$$

de onde podemos obter K_f para formarmos a lei de controle:

$$u_f = -R^{-1} B_2^T K_f z_f = G_2^{(0)}$$

Para obtermos a lei de controle quasi-ótima composta por u_s e u_f usamos a seguinte equação:

$$u = [(I + G_2 A_{22}^{-1} B_2) G_0 + G_2 A_{22}^{-1} A_{21}] x + G_2 z$$

Assim conclui-se os passos para projeto do controlador L.Q. associado com uma modelagem segundo o Método das Perturbações Singulares. Segue-se que esse controlador não atinge todas as especificações que se deseja da malha, devido ao excesso de incertezas incluídas no modelo adotado. A partir disto decidiu-se projetar um controlador que incluisse um estimador para melhorar a ação do observador.

- Estimador Gaussiano - O filtro de Kalman

O controlador L.Q.G. considera que existe um ruído que prejudica a observação do sistema, para então projetar um filtro (Kalman) que melhora o desempenho do observador.

A adoção deste tipo de observador se deve, principalmente, ao fato dos parâmetros serem muito mal conhecidos e não se ter disponíveis todas as medições necessárias para se realizar o controle.

O princípio desta teoria se dá na inclusão de um 'ruído branco' no modelo, que simularia as incertezas que tem-se em relação a este. Assim o filtro de Kalman fará as estimativas das variáveis de estado, considerando a presença deste ruído.

A ação do filtro de Kalman está dividida em duas etapas: Propagação e Atualização.

A fase de propagação nada mais é que a estimativa, a partir do ultimo valor que se tem das variáveis de estado, do valor futuro das mesmas. Assim esta se fazendo uma projeção do comportamento da planta.

A atualização se dá com a correção do valor projetado durante a propagação, a partir da medição realizada do valor atual, correção é realizada segundo um ganho de projeto.

Outro aspecto importante é o fato de que também o filtro de Kalman baseia sua ação no modelo adotado, assim verifica-se que este deve ser formulado com a maior acuracidade possível.

No caso do Método das Perturbações Singulares teremos dois filtros um para o subsistema rápido outro para o lento. Para tanto devemos modificar o modelo que estávamos adotando para incluir a citada perturbação como sendo um ruído branco.

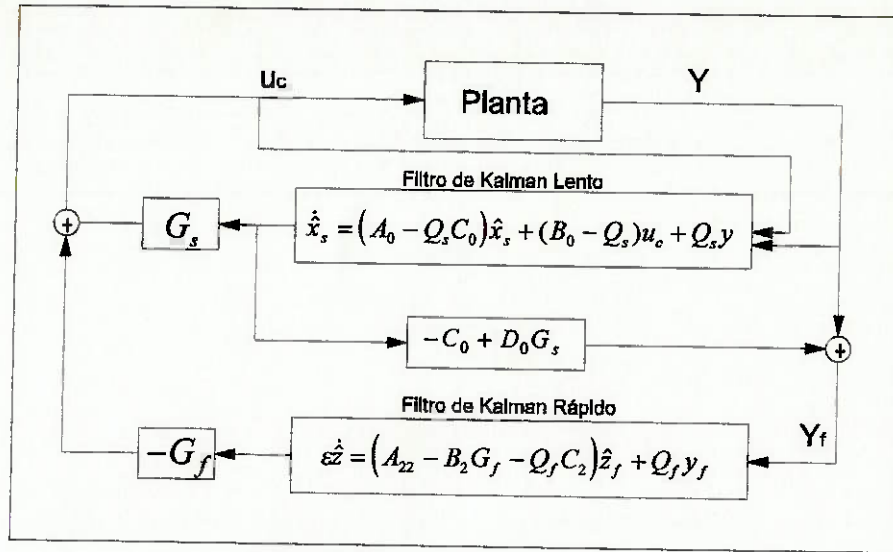
$$\dot{x}(t) = A_{11}x(t) + A_{12}z(t) + F_1 w(t), \quad x(0) = x_0$$

$$\dot{z}(t) = A_{21}x(t) + A_{22}z(t) + F_2 w(t), \quad z(0) = z_0$$

$$y(t) = C_1 x(t) + C_2 z(t) + V v(t)$$

onde $Vv(t)$ e $Fw(t)$ representam o ruído branco com distribuição Gaussiana.

Para realizarmos este tipo de controle devemos montar a seguinte malha:



Representação da Malha de Controle L.Q.G.

Como vemos a lei de controle para esse controlador fica:

$$u(t) = -[G_s(\varepsilon)\hat{x}(t) + G_f(\varepsilon)\hat{z}(t)]$$

onde $\hat{x}$ e $\hat{z}$ são as estimativas promovidas pelo filtro de Kalman.

G_s e G_f são os ganhos do controlador definidos no item anterior,

e Q_s e Q_f são os ganhos do filtro de Kalman.

Temos que:

$$A_0 = A_{11} - A_{12}A_{22}^{-1}A_{21},$$

$$C_0 = C_1 - C_2A_{22}^{-1}A_{21},$$

$$F_0 = F_1 - A_{12}A_{22}^{-1}F_2,$$

$$V_0 = V + S_0S_0^T,$$

$$S_0 = -C_2A_{22}^{-1}F_2,$$

$$Q_s = (P_s C_0^T + F_0 S_0^T) V_0^{-1}$$

$$Q_f = P_f C_2^T V^{-1}$$

onde P_f e P_s são resultado das equações de Riccati expostas abaixo.

$$0 = [A_0 - F_0 S_0^T V_0^{-1} C_0] P_s + P_s [A_0 - F_0 S_0^T V_0^{-1} C_0]^T$$

$$+ F_0 [I - S_0^T V_0^{-1} S_0] F_0^T - P_s C_0^T V_0^{-1} C_0 P_s$$

$$0 = [A_{22} P_f + P_f A_{22}^T + F_2 F_2^T - P_f C_2^T V^{-1} C_2 P_f]$$

Controle Linear Quadrático - Resultados

O Método das Perturbações Singulares (MPS) foi utilizado no projeto do controlador, uma vez que trata-se de um modelo testado e comprovado.

Analisando os autovetores verifica-se que o autovetor associado ao autovalor rápido ($-4.811E+2$) está ligado ao estado O , concentração de oxigênio, sendo pequenas as influências nos demais estados. Para os autovalores restantes a situação é oposta, com as componentes do estado O bastante pequenas.

O Método das Perturbações Singulares pressupõe a separação do sistema original em sub-sistemas, rápido e lento, e projeto de observadores de estados e controladores ótimos para cada subsistema. O controlador global será sub-ótimo, com desvio em relação ao ótimo da ordem de ε^2 , onde ε é o pequeno parâmetro utilizado para separar os modos (vide Kokotovic, 1975).

No caso em estudo, ε foi escolhido como a relação entre o menor e o maior auto-valores associados à dinâmica do processo, $\varepsilon = 8.314E-5$.

Assim consegue-se uma separação, $\mathbf{x}_1 = [\delta x \ \delta p \ \delta s_1 \ \delta s_2]^T$ para os modos lentos e $\mathbf{x}_2 = [\delta o]$ para o modo rápido, onde δ_i indica a diferença entre um estado i e seu valor de equilíbrio.

$$\dot{\mathbf{x}}_1 = \mathbf{A}_{11} \cdot \mathbf{x}_1 + \mathbf{A}_{12} \cdot \mathbf{x}_2 + \mathbf{B}_1 \cdot \mathbf{u}$$

$$\varepsilon \cdot \dot{\mathbf{x}}_2 = \mathbf{A}_{21} \cdot \mathbf{x}_1 + \mathbf{A}_{22} \cdot \mathbf{x}_2 + \mathbf{B}_2 \cdot \mathbf{u}$$

$$\mathbf{y} = \mathbf{C}_1 \cdot \mathbf{x}_1 + \mathbf{C}_2 \cdot \mathbf{x}_2$$

O observador identidade associado ao sistema é dado por Porter (1977):

$$\dot{\hat{\mathbf{x}}}_1 = \mathbf{A}_{11} \cdot \hat{\mathbf{x}}_1 + \mathbf{A}_{12} \cdot \hat{\mathbf{x}}_2 + \mathbf{B}_1 \cdot \mathbf{u} + \mathbf{P}_1 \cdot \{\mathbf{y} - \mathbf{C}_1 \cdot \hat{\mathbf{x}}_1 - \mathbf{C}_2 \cdot \hat{\mathbf{x}}_2\}$$

$$\varepsilon \cdot \dot{\hat{\mathbf{x}}}_2 = \mathbf{A}_{21} \cdot \hat{\mathbf{x}}_1 + \mathbf{A}_{22} \cdot \hat{\mathbf{x}}_2 + \mathbf{B}_2 \cdot \mathbf{u} + \mathbf{P}_2 \cdot \{\mathbf{y} - \mathbf{C}_1 \cdot \hat{\mathbf{x}}_1 - \mathbf{C}_2 \cdot \hat{\mathbf{x}}_2\}$$

No caso, $\hat{\mathbf{x}}_1$ e $\hat{\mathbf{x}}_2$ são as estimativas dos vetores de estado $\mathbf{x}_1$ e $\mathbf{x}_2$ das equações (2). As matrizes $\mathbf{P}_1$ e $\mathbf{P}_2$ de ganhos do observador foram projetadas de modo a satisfazer todas as condições de observabilidade, controlabilidade e detetabilidade exigidas para a aplicação do método, e alocando pólos da mesma ordem de grandeza do pólo mais rápido do sistema; pode-se obter:

$$\mathbf{P}_1 = \begin{bmatrix} 8.53E+0 & -3.74E+3 & -9.25E+2 \\ 2.65E+3 & -1.09E+2 & -7.75E+1 \\ -2.98E-1 & 2.40E+2 & 7.94E+1 \\ 1.62E+1 & 5.54E+4 & -9.88E+2 \end{bmatrix}$$

$$\mathbf{P}_2 = [0.00E+0 \quad 0.00E+0 \quad 6.39E-1]$$

A metodologia para projeto de controladores L.Q. (lineares quadráticos) segue os passos estabelecidos por Chow e Kokotovic (1976) e visam a minimização de

$$J = \int_0^{\infty} \{ \mathbf{x}^t \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{u}_c^t \cdot \mathbf{R} \cdot \mathbf{u}_c \} \cdot dt$$

onde $\mathbf{Q} \propto \mathbf{C}^t \cdot \mathbf{C}$ e $\mathbf{R}$ são as matrizes de ponderação de estado e de controle.

O controle composto $\mathbf{u}_c$, deve-se lembrar, contém as parcelas obtidas com a separação do funcional nos modos lentos e rápidos, com a dinâmica dada pelas equações (2), e pode ser sintetizado em:

$$\mathbf{u}_c = k_0 \cdot \mathbf{u}_s + k_f \cdot \mathbf{u}_f$$

onde k_0 e k_f são matrizes de ganhos constantes associadas às soluções das equações de Riccati dos problemas L.Q. dos subsistemas rápido e lento; $\mathbf{u}_s$ e $\mathbf{u}_f$ correspondem aos controles lentos e rápidos, respectivamente. Maiores detalhes podem ser obtidos em Chow e Kokotovic (1986).

$$k_0 = \begin{bmatrix} 5.25E-3 & 6.47E-4 & -1.28E-2 & -1.06E-5 \\ 1.65E+1 & -6.43E+0 & -1.48E+2 & 1.95E+0 \\ -2.96E+0 & 1.97E-2 & 1.39E-3 & -4.94E-1 \\ 1.09E+1 & -1.77E-1 & 2.63E+0 & 9.24E-1 \end{bmatrix}$$

$$k_f = \begin{bmatrix} -3.61E-6 \\ 0.00E+0 \\ 0.00E+0 \\ -1.09E+1 \end{bmatrix}$$

O projeto pressupõe medidas sem erros, disponíveis em tempo real e atuadores instantâneos, o que não ocorre na prática. Numa primeira fase dois problemas foram tratados:

- 1) as medidas de s_1 e p são realizadas a intervalos e com atraso de 20 minutos, e a medida de O a cada 15 segundos, com atraso de mesmo valor. As medidas são ainda admitidas perfeitas, embora a inclusão de estimadores probabilísticos (Filtros de Kalman) seja uma extensão natural do trabalho;
- 2) as bombas controlam as vazões s_{1a} , s_{2a} e D não atuam em intervalos menores que um minuto, respondendo a valores médios neste intervalo, e têm uma sensibilidade mínima para atuação ("dead band").

A restrição 1 acima foi equacionada da seguinte forma: o sistema linear contínuo (equação 2) foi discretizado com uma taxa de amostragem 5 vezes menor do que o menor intervalo de medida realizada, obtendo-se assim as matrizes Φ e Γ , correspondentes a $\mathbf{A}$ e $\mathbf{B}$. Quando uma medida está disponível, leva-se em conta que esta medida refere-se a um evento passado; e faz-se então uma "grande propagação"

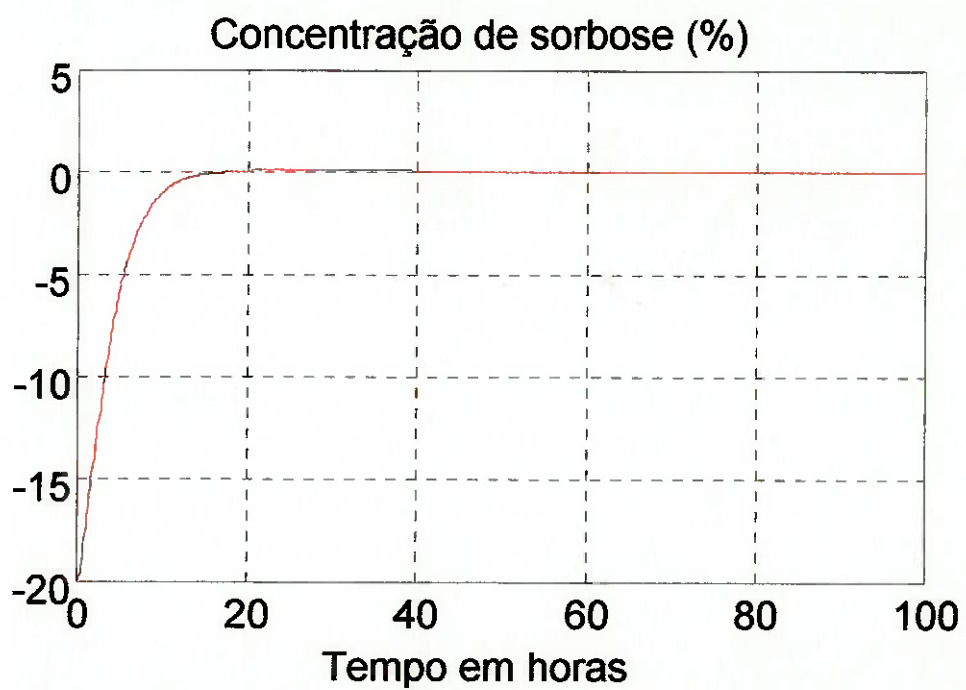
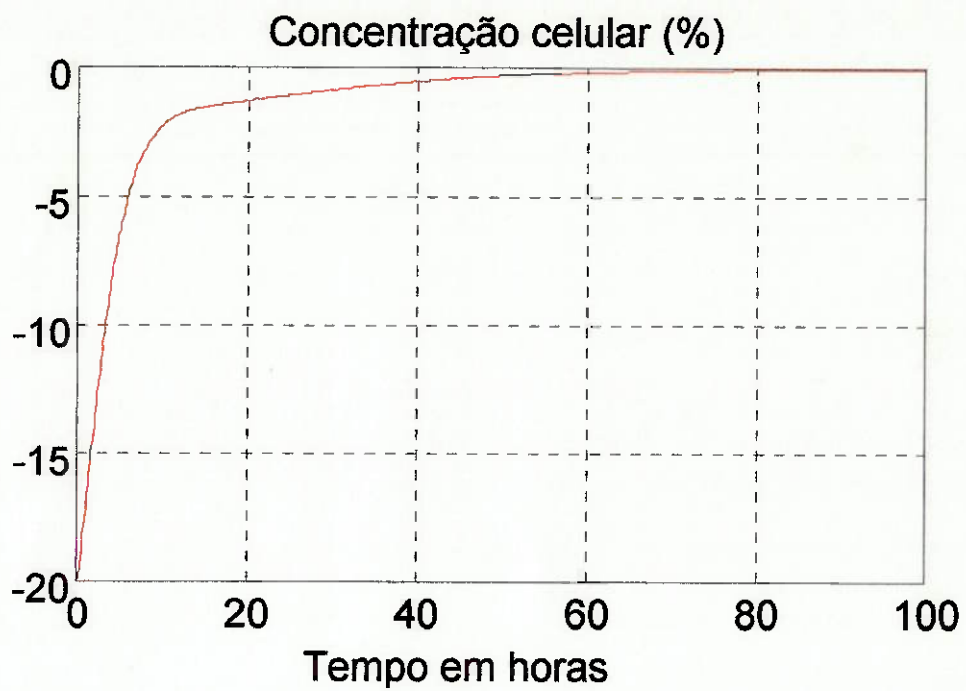
ou atualização da mesma, desde o instante de coleta da amostra (ocorrência do evento) até o instante presente (disponibilidade de medição). Enquanto se esperam novas medidas, fazem-se “pequenas propagações” a cada intervalo de discretização para se poder alimentar o observador e gerar as ações de controle.

Os resultados obtidos quando da utilização deste método de controle se mostraram bastante satisfatórios. Se comparados à evolução da planta sem controle vemos que há uma melhora no desempenho, principalmente no que se refere ao tempo de convergência. Estão demonstrados a seguir resultados referentes a esse tipo de simulação, quando submetidos a desvios iniciais de 20%.

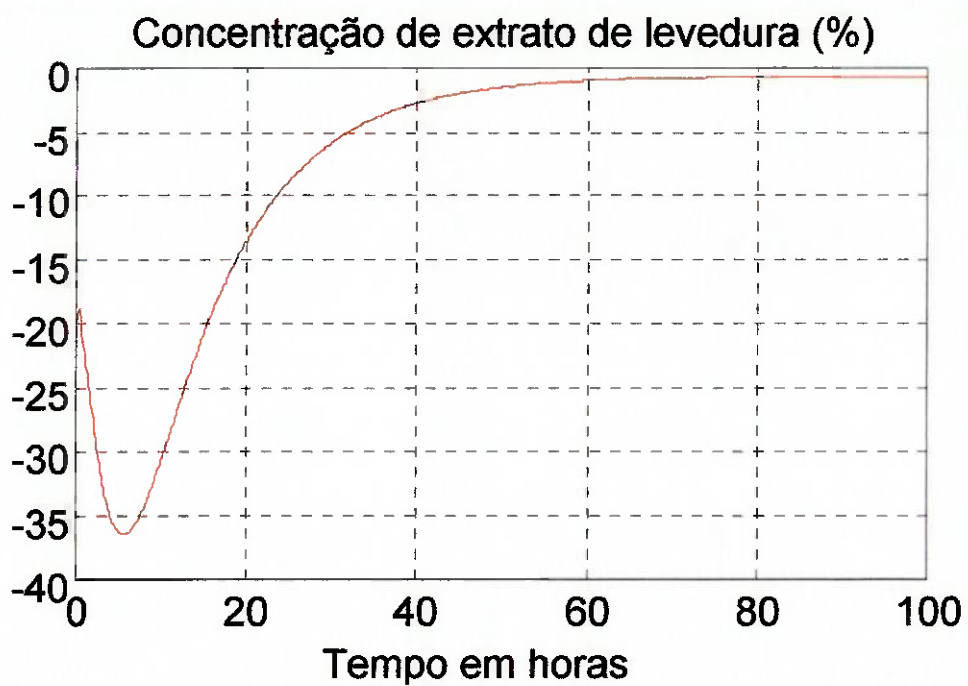
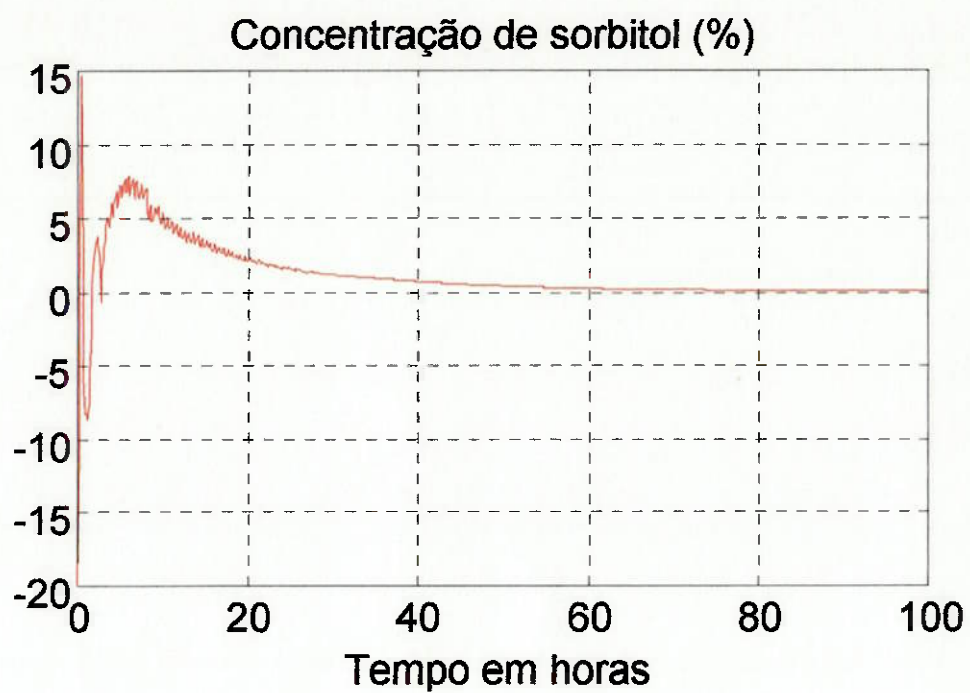
As concentrações de sorbitol e sorbose dissolvidos apresentam repostas bastante rápidas, muito parecidas com o caso ideal de realimentação de todos os estados do sistema, isto é, supondo que medidas de todas as variáveis estejam sempre disponíveis. Com esses resultados se esgota as possibilidades de melhora de desempenho com esse modelo e com essa estratégia de controle. O próximo passo para se conhecer melhor a planta era tentar melhorar o modelo proposto e se conseguir um controle mais robusto.

Resultado Controle LQ

Variáveis de Estado:



Simulação com controle Linear Quadrático - sem ruído



Simulação com controle Linear Quadrático - sem ruído

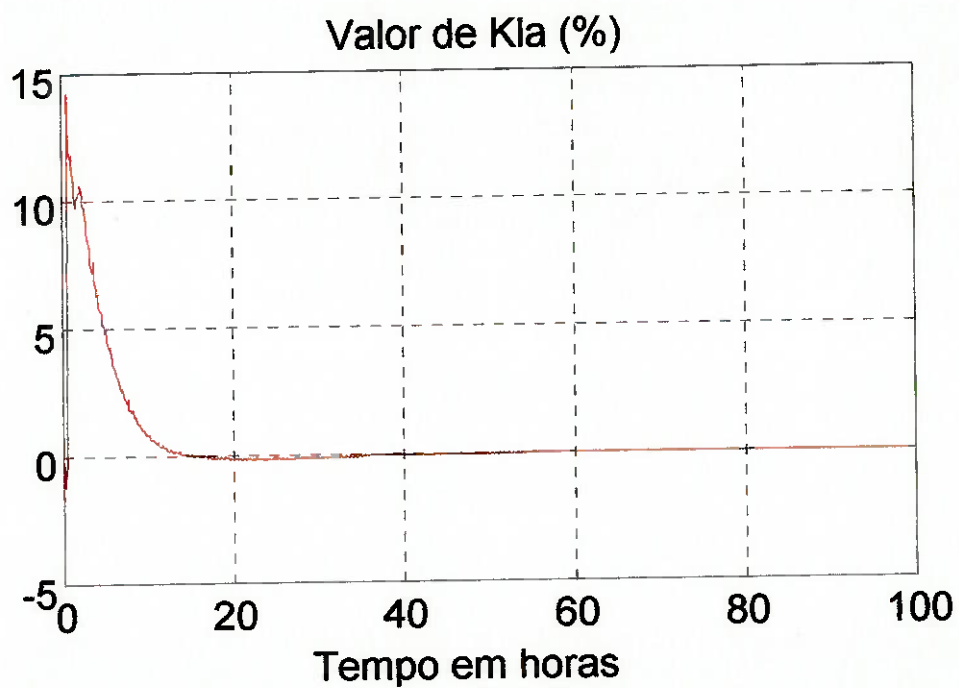
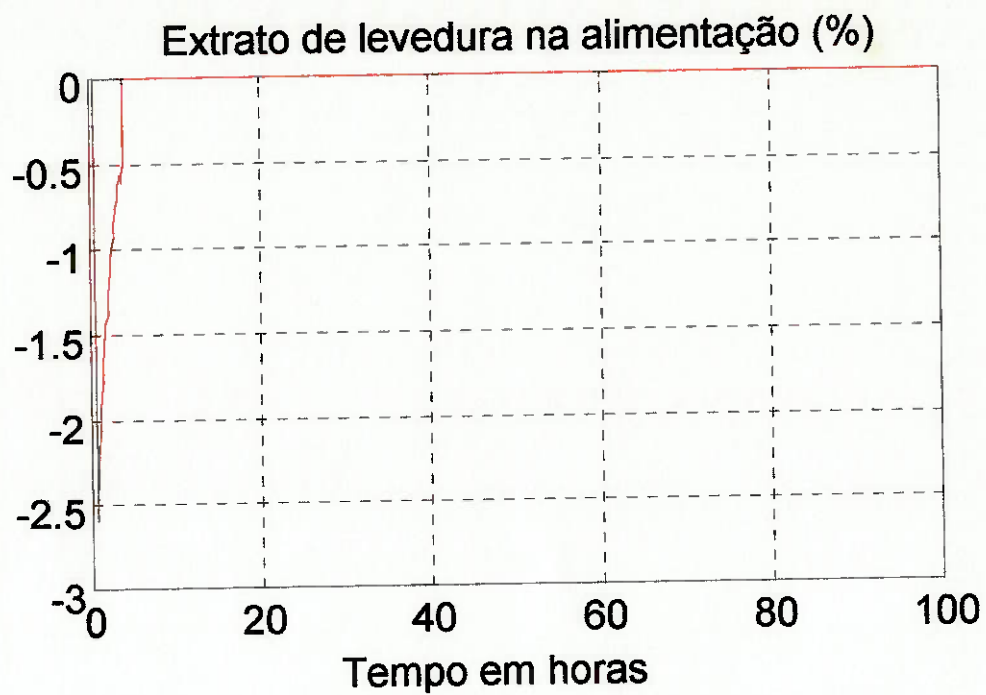


Simulação com controle Linear Quadrático - sem ruído

Variáveis de Controle:

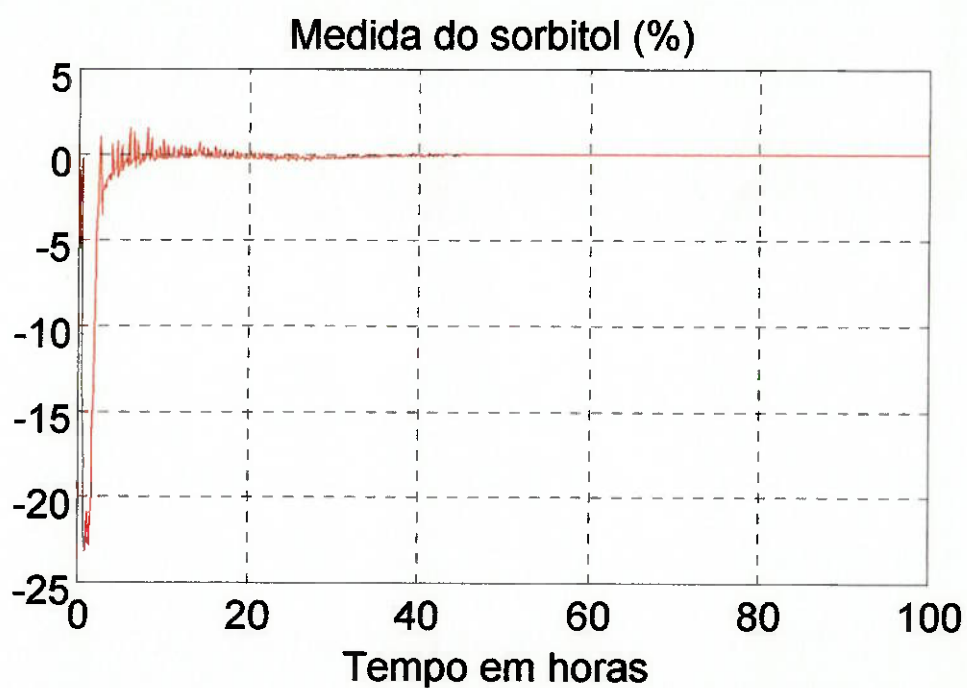
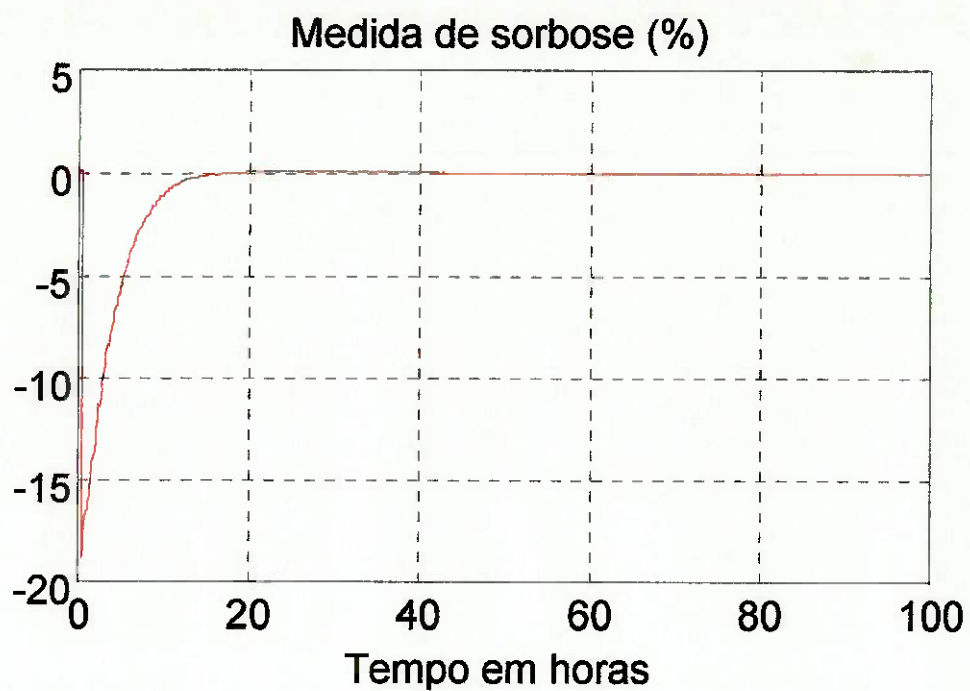


Simulação com controle Linear Quadrático - sem ruído



Simulação com controle Linear Quadrático - sem ruído

Variáveis Medidas:



Simulação com controle Linear Quadrático - sem ruído



Simulação com controle Linear Quadrático - sem ruído

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xe - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00 -.123794160E-02 .707919687E-01 .176010163E-01 .812277235E-01
.393306572E+01 -.115528924E+00 .359136491E+01 .505149169E+00 .558033286E+01
-.431723616E+01 .411919255E-01 -.401255173E+01 -.585665554E+00 -.598804914E+01
-.847457627E-01 .131137881E-02 -.749914922E-01 -.986451444E-01 -.860463173E-01
-.955666447E+03 .125364720E+02 -.855535230E+03 -.178243181E+03 -.143892113E+04

Matriz Fi

.999981172E+00 -.784903394E-06 .421348740E-04 .111597328E-04 .393510318E-04
.198542078E-02 .999920663E+00 .183598939E-02 .180172673E-03 .270530063E-02
-.221106331E-02 .161531388E-04 .997897628E+00 -.229664841E-03 -.290284112E-02
-.506743625E-04 .831465460E-06 -.446344004E-04 .999921514E+00 -.416854150E-04
-.463296555E+00 .608081778E-02 -.414739698E+00 -.864568839E-01 .302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01 .000000000E+00 .000000000E+00 .000000000E+00
-.116870000E+03 .000000000E+00 .000000000E+00 .000000000E+00
.128284000E+03 .800000000E-01 .000000000E+00 .000000000E+00
.251000000E+01 .000000000E+00 .800000000E-01 .000000000E+00
.693600000E+01 .000000000E+00 .000000000E+00 .693600000E+01

Matriz Gama

-.197813202E-02 .155751693E-08 .403827121E-09 .136024765E-06
-.972653142E-01 .716993845E-07 .819115086E-08 .934885760E-05
.106764081E+00 .665853199E-04 .100006689E-07 -.100316676E-04
.208880533E-02 -.164991200E-08 .666640167E-04 -.144094031E-06
-.230530652E-01 -.165312790E-04 -.344534729E-05 .336908405E-02

Matriz C - Matriz de Medições de estado do modelo linearizado

.000000000E+00 .273808505E+00 .000000000E+00 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .466200466E+01 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .000000000E+00 .000000000E+00 .141843972E+01

Simulação com controle Linear Quadrático - sem ruído

Matriz L - Ganho do Filtro de Kalman

.104881160E+01 -.274675463E+00 -.119420802E+02
.604973762E+01 -.603777988E-01 -.405158003E+02
-.187952727E+01 .331197761E+00 .470703219E+02
-.737245372E+01 .350571984E+00 -.532643417E+00
.000000000E+00 .000000000E+00 .101443940E+05

Matriz G1 - Ganho do controlador para o subsistema lento

.983624843E-03 .169438436E-02 -.176304821E-01 -.809878259E-03
-.406824781E+01 -.376669283E+01 -.148684744E+03 -.104750287E+01
-.790870254E-01 .470430688E-02 -.610903673E-03 -.234560217E-01
.117327725E+01 -.260776743E+01 .280598891E+02 .171027867E+01

Matriz G2 - Ganho do controlador para o subsistema rápido

-.481173524E-06
.000000000E+00
.000000000E+00
-.322456941E+01

Valor de magO - Velocidade do observador

.800000000E+01

Valor de magR - Magnitude do controlador

.400000000E+01

Vetor PesoR

.160000000E+02
.320000000E+01
.400000000E+01
.400000000E+02

Vetor PesoC

.320000000E+02
.800000000E+01
.800000000E+00

Valor de dtO (em minutos)

.500000000E-01

Matriz Fimod

.999208668E+00 -.229714585E-03 .416995801E-03 -.132861872E-03 .107189653E-02
.345000404E-03 .998562860E+00 .601684779E-03 -.125782811E-03 .397581276E-02
-.213953414E-03 .418255689E-03 .998399728E+00 .142841309E-03 -.458832401E-02
-.100922771E-03 .168212137E-02 -.145032089E-02 .999911854E+00 .545465696E-04
-.603206003E-01 .765389063E-03 -.539784866E-01 -.112598799E-01 .187714874E-03

Matriz Gamamod

```

-.193834157E-02 .155475610E-07 -.408762349E-08 .573449897E-05 .000000000E+00
.239067201E-03 -.106647874E-02 .000000000E+00 -.219558844E-02
-.972589844E-01 .261497916E-07 -.292603017E-08 .212564343E-04 .000000000E+00
.137936868E-02 -.234247117E-03 .000000000E+00 -.374061369E-02
.106762007E+00 .666062882E-04 .329963012E-08 -.245297694E-04 .000000000E+00 -
.428316504E-03 .128579700E-02 .000000000E+00 .469263385E-02
.193255030E-02 -.482959646E-07 .666637446E-04 .258870932E-06 .000000000E+00 -
.168066652E-02 .136083708E-02 .000000000E+00 -.172429299E-03
-.487842731E-02 -.332772482E-05 -.693734450E-06 .439118398E-03 .000000000E+00
.265588584E-04 -.190561085E-04 .000000000E+00 .908963538E+00

```

Dados iniciais para simulação (arquivo VIT_C.DAT)

```

-20.0 -20.0 -20.0 -20.0 -20.0 estados iniciais
0.00 0.00 0.00 0.00 0.00 observador inicial
0.00 tempo inicial (hora) hoje 27/11/95 14h50m
100.00 tempo final (hora)
3.00 duração do filtro do atuador (min)
0.00 atraso para início do controle (horas)
400.00 atraso para melhoria da medição (horas)
1.e-6 precisão
1.0e-5 passo inicial (chute)
1.e-8 menor passo admissível (hora)
-1.0 tempo entre resultados armazenados (segundos)
400000 número máximo de passos
0 =1 simula-se até o máximo de passos; =0 não há limite
2 =1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)
2 =1 observador é o estado real; =2 obs discreto (despro)
0 numero inteiro de dto's para atraso dos contr na memorex (shift)
0.0 erro máximo aleatório para observadores iniciais (%)
360000.0 tempo de aleatorização dos parâmetros (min);
50.0 50.0 50.0 50.0 valores máximos de controle (%)
-80.0 -80.0 -80.0 -50.0 valores mínimos de controle (%)
0.0 30.0 30.0 0.0 -0.25 atrasos das medidas (min)
1.0 1.0 1.0 1.0 1.0 fatores de ajuste da propagaçã
0.5 0.5 0.5 0.0 0.5 deadband (%) p/ atuadores(4) e
0.0 0.0 0.0 0.0 0.0 _μ0, _Ks1, _Ks2, _W, __ (%)
0.0 0.0 0.0 0.0 0.0 _β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0 0.0 0.0 0.0 0.0 _Yxs2, _Kp, _YxO, ruído, ruído

```

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

Simulação realizada em 0h05m19.77s.

Final normal da simulação

Implementação - Projeto e Simulação

- Projeto

O projeto do Controlador-Observador foi implementado em MatLab seguindo a seguinte estrutura.

- Entrada de Dados: Arquivo DadoVC.dat

Este arquivo contém os parâmetros de projeto, tais como as matrizes do sistema linearizado, pesos de projeto do controlador e nível de ruído.

- Programa Présing

Trata-se de um programa de inicialização de parâmetros que carrega o programa Singgaus.

- Programa Singgaus

Este programa, inicialmente carrega o arquivo DadoVC.dat e faz o projeto propriamente dito, do controlador e do observador, seguindo o seguinte roteiro.

Calcula as Matrizes Linearizadas A, B, C

Calcula as Matrizes do Sistema Desacoplado (M. P. S.)

Calcula as Matrizes de Ganho do Controlador (Eq. de Ricatti) - Rápido e Lento

Calcula as Matrizes de Ganho do Observador (Eq. de Ricatti) - Rápido e Lento

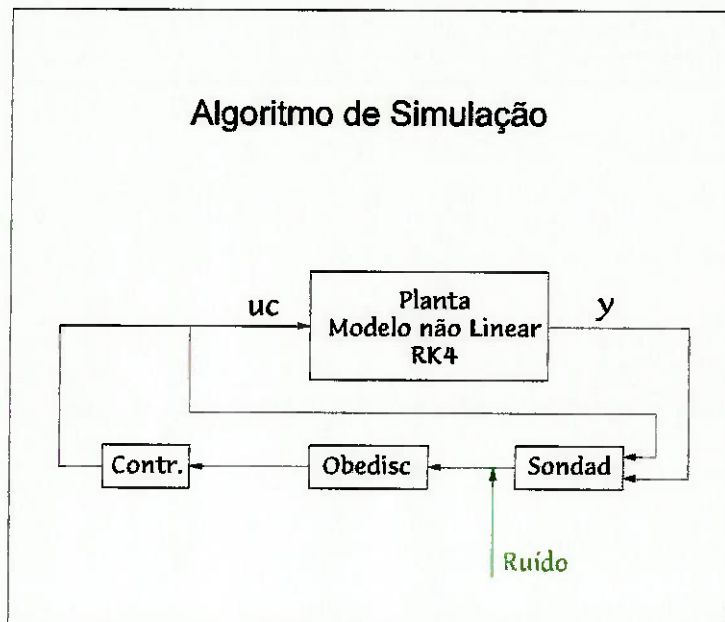
Salva um arquivo com as Matrizes calculadas - Dadvitc

- Arquivo Dadvitc

Este arquivo contém os valores projetados em MatLab e é usado como entrada o programa de Simulação Calvic.

- Simulação

A simulação do desempenho da planta foi implementado em fortran em um programa (Calvic) que tem a estrutura mostrada abaixo.



- Rotina RK4

Esta rotina simula o desempenho da planta através da resolução do modelo não linear por um algoritmo de Runge-Kuta de quarta ordem.

- Sondad

Os valores que são calculados pelo RK4 são atualizados pela rotina Sondad. Isso é necessário porque a planta produz medidas da concentração de O_2 dissolvido em ciclos de 15 s e as medidas de concentração de sorbitol e de sorbose são obtidas com intervalo de 20 min. Assim esta rotina irá produzir, através do modelo linear, medidas atualizadas de 3 em 3s.

Uma característica importante desta rotina é que o tratamento que se dá aos dados é um tratamento discreto.

- Obedisc

Tal rotina nada mais é que o observador discreto, ou seja o filtro de Kalman discreto.

- Contr

Esta rotina calcula a partir das estimativas feitas pela rotina Obedisc as entradas de controle da planta.

- Memorex

Esta rotina apenas armazena os dados para que se possa analisa-los posteriormente.

- Entradas do Programa de Simulação
 - Arquivo Dadvite.mat - contém os parâmetros de controle
 - Arquivo vit_c.dat - contém as características da simulação que se quer fazer, como tempo de simulação, nível de ruído, ausência ou presença de controle, etc.
- Saída do Programa de Simulação: O programa produz dois arquivos de saída com nome escolhido pelo usuário. O primeiro com extensão .mat com os resultados da simulação, que devem ser tratados através do MatLab. O outro arquivo tem o mesmo nome do primeiro, mas com extensão .txt, que contém um protocolo dos dados da simulação.

Todos os arquivos e programas citados acima estão impressos nas páginas seguintes para informação mais detalhada do leitor. Pode-se, então, verificar o que se especifica em cada um dos arquivos citados.

**Arquivos de Simulação
e
Projeto**

Programa DADOVC;

```

Cg      = 7.50      ; %* DADOVC.M *****
mimax   = .696      ; %* Arquivo de constantes e pontos de      *
Ks1     = 3.08      ; %* equilíbrio para simulação da fer-      *
Ks2     = .102      ; %* mentação no projeto "Vitamina C".      *
w       = 298      ; %***** Ver 13/6/95 16h15m*****
alfa    = 28.7      ; %*
betam   = 9.91      ; %*
Kbetas1 = 3.78      ; %* ATENÇÃO *****
Kbetao  = 0.502     ; %* Estes dados são lidos pelo programa      *
Ko      = .179      ; %* Calvic, os valores numéricos devem      *
Yxs1    = .235      ; %* ser todos reais, entre as colunas      *
Yps1    = .989      ; %* 10 e 25 inclusive                        *
Yxs2    = .944      ; %*****
Kp      = .006510   ; %*
Yxo     = .000132   ; %*
Ypo     = .011250   ; %*
u1      = .0800     ; % D *****
u2      = 130.00    ; % SE1 * Entradas para ponto ótimo (controles) *
u3      = 3.5100    ; % SE2 * conforme 13/6/95                      *
u4      = 327.45    ; % K1a *****
X1      = 2.3770    ; % X *****
X2      = 116.87    ; % P * Ponto de equilíbrio ótimo (estados)      *
X3      = 1.7160    ; % S1 * conforme 13/6/95                      *
X4      = 1.0000    ; % S2 *****
X5      = 0.5640    ; % O

Deltaes = 0.12      ; %      *Ruído nos Estados                      *
Deltao  = 0.07      ; %      *Ruído no Oxigênio                      *
Deltas1 = 0.05      ; %      *Ruído no Sorbitol e Sorbose          *
Deltap  = 0.05      ;

```

```

%
%
%           Projeto Vitamina C
%           IPT-DME/SS-DQ/AB
%
%           programa PRESING.M
%
% Programa para inicializar parâmetros usados no SINGGAUS.M
%
%
clear; % 26/8 sem o clear, rodar 2 vezes consecutivas este programa
% acaba transformando os vetores coluna em vetores linha!
dtO = 0.05/60;
magO = input('Velocidade do observador "magO"(>1.0): ');
magR = input('Magnitude do controlador "magR"(0.05): ');
%magO = 8.0 ;
%magR = 4.0 ;
PesoR( 1,1 ) = input('Peso do Controle R(1,1) "D"(4): ');
PesoR( 2,1 ) = input('Peso do Controle R(2,2) "S1(0.8)": ');
PesoR( 3,1 ) = input('Peso do Controle R(3,3) "S2"(2): ');
PesoR( 4,1 ) = input('Peso do Controle R(4,4) "Kla(20)": ');
% PesoR = [ 4 .8 2 20 ];
PesoC( 1,1 ) = input('Ganho da Medida C(1,2) "P"(8): ');
PesoC( 2,1 ) = input('Ganho da Medida C(2,3) "S1"(8): ');
PesoC( 3,1 ) = input('Ganho da Medida C(3,5) "O"(0.4): ');
% PesoC = [ 8 8 .4 ];
disp(' ');
disp(' Agora executando o SINGGAUS... ');
disp(' ');
singgaus;

```

```

%
%
%           Projeto Vitamina C
%           IPT-DME/SS-DQ/AB
%
%           programa SINGGAUS.M
% Programa para projetar observador de estado e controlador (baseado em
% observador); usa-se método de perturbações singulares.
% Gera também um arquivo bin rio com dados para uso no controle do fermen-
% tador Braun.
%
%
%           Versão 14/7/94 15h18m
% *****
%
format short e
%
% Constantes
if exist( 'magO' ) ~= 1 ,
    disp( '' );
    disp( '    Anta microcéfala! Rode o "PRESING" antes!' );
    disp( '' );
    pause;
    break;
end
um = [ 1 1 1 1 1 ]';
%
% Carega constantes do sistema e respectivos pontos de equilibrio
dadovc;
%
%           Entradas para ponto ótimo (controles)
Ue = [u1 u2 u3 u4]';    % vetor de controle (valores no equilíbrio)
%
%           Ponto de equilibrio ótimo
Xe = [ X1 X2 X3 X4 X5 ]'; % vetor de estados (valores no equilíbrio)
%
% Ponderacao da matriz de medicao C
C = zeros(3,5)        ;
% medida de P (X2)
if PesoC(1) > 0
    C(1,2) = 1/X2 * PesoC(1) ;
else
    C(1,2) = 1;
end
% medida de S1 (X3)
if PesoC(2) > 0
    C(2,3) = 1/X3 * PesoC(2) ;
else
    C(2,3) = 1;
end
% medida de O (X5)
if PesoC(3) > 0
    C(3,5) = 1/X5 * PesoC(3) ;
else
    C(3,5) = 1;
end
end

```

Programa SINGGAUS.M

```

% mi = mimax * X3/(Ks1+X3+(X3^2)/w) * X4/(Ks2+X4) * X5/(Ko+X5) * exp(-Kp*X2)
beta = betam * X3/(Kbetas1+X3) * X5/(Kbetao+X5);
mi = u1;
X1e = 0.;
Pe = 0.;
%
%----- Matrizes Linearizadas de Entrada B e D
%
b11 = (X1e-X1);
b21 = (Pe -X2);
b31 = (u2 -X3);
b41 = (u3 -X4);
b51 = (Cg -X5);
b32 = u1 ;
b43 = u1 ;
b54 = (Cg -X5);
%
B= [b11 0. 0. 0.;b21 0. 0. 0.;b31 b32 0. 0.;b41 0. b43 0.;b51 0. 0. b54];
%
a(1) = 1. ;
a(2) = alfa;
a(3) = -(1/Yxs1 + alfa/Yps1);
a(4) = -(1/Yxs2);
a(5) = -(1/Yxo + alfa/Ypo );
%
b(1) = 0.;
b(2) = 1.;
b(3) = -(1/Yps1);
b(4) = 0.;
b(5) = -(1/Ypo );
%
mil(1) = 1/X1;
mil(2) = -Kp ;
mil(3) = (Ks1/X3-X3/w)/(Ks1+X3+X3^2/w);
mil(4) = (Ks2/X4)/(Ks2+X4);
mil(5) = (Ko /X5)/(Ko +X5);
%
betal(1) = 1/X1;
betal(2) = 0.;
betal(3) = (Kbetas1/X3)/(Kbetas1+X3);
betal(4) = 0.;
betal(5) = (Kbetao /X5)/(Kbetao +X5);

%----- Matrizes Linearizadas A e C
%
for i=1:5
    for j=1:5
        A(i,j) = (a(i)*mil(j)*mi + b(i)*betal(j)*beta) * X1;
    end
    A(i,i) = A(i,i) - u1;
end
A(5,5) = A(5,5) - u4;
%
[Fi, Gama] = c2d(A, B, dtO);
[T,Av] = eig(A);

```



```

%           Método das perturbações singulares (SPM)
maxi = max ( abs( Av * um ) );
mini = min ( abs( Av * um ) );
epsf = mini / maxi ;
%
A11 = A(1:4, 1:4);
A12 = A(1:4, 5);
A21 = A( 5, 1:4) * epsf;
A22 = A( 5, 5) * epsf;
B1 = B(1:4, :);
B2 = B( 5, :) * epsf;
C1 = C( :, 1:4);
C2 = C( :, 5);
%
Ainv = inv( A22 );
A0 = A11 - A12 * Ainv * A21;
B0 = B1 - A12 * Ainv * B2;
C0 = C1 - C2 * Ainv * A21;
D0 = - C2 * Ainv * B2;
%
% Av0 = eig(A0);
%
%           Projeto do controlador do modo lento
% escolha da matriz de ponderação do índice de desempenho (U'*R*U)
R = eye( 4 ); % (por exemplo)
if magR <= 0,
    magR = 1;
end
i = 1;
while i < 5,
    if PesoR( i ) > 0 ,
        R(i,i) = magR * PesoR(i) / Ue(i) / Ue(i); % 17/8/93
    else
        R(i,i) = 1;
    end
    i = i + 1;
end
%keyboard
% cálculo da matriz R0 (eq. 7.83b)
R0 = R + D0'*D0;
R0inv = inv( R0 );
%
% cálculo das matrizes parâmetros de lqr (eq. de Riccati)
A_Rics = A0 - B0*R0inv*D0'*C0;
B_Rics = B0;
dimenD0 = size(D0);
Ip = eye( dimenD0(1) ); % (por exemplo)
Pre_Q = Ip - D0*R0inv*D0';
Q_Rics = C0'*Pre_Q*C0;
%keyboard; (Pode-se verificar que Q_Rics é positiva definida)
R_Rics = R0;
[KKs, Ks] = lqr2( A_Rics, B_Rics, Q_Rics, R_Rics ); %lqr2 = lqr modificado
%
% Matriz de ganho (modos lentos)
Gs = -R0inv * ( D0'*C0 + B0'*Ks );

```

```

%
% Projeto do controlador do modo rápido
% cálculo das matrizes parâmetros de lqr (eq. de Riccati), cálculo de Kf
A_Ricf = A22;
B_Ricf = B2;
Q_Ricf = C2'* C2;
R_Ricf = R;
[KKf, Kf] = lqr( A_Ricf, B_Ricf, Q_Ricf, R_Ricf);
%
% Matriz de ganho (modos rápidos)
Gf = -inv(R) * B2'* Kf;
%
% Síntese do controlador composto G = f(Gs, Gf)
%
G1 = ( eye(4) + Gf*Ainv*B2) * Gs + Gf * Ainv * A21;
G2 = Gf;

% Calculo dos autovalores do sistema controlado LENTO
%[T,Av] = eig( A0 + B0*Gs );
%um = [ 1 1 1 1 ]';
%maxi = max ( abs( Av * um ) );
%mini = min ( abs( Av * um ) );

% PROJETO DO OBSERVADOR
% F0 = A11 - A12*inv(A22-P*C2)*(A21-P*C1)
% COMO P = 0 => F0 = A0
Fdiag = [Deltaes Deltaes Deltaes Deltaes Deltaes];
Ruidoes = diag(Fdiag);
Vdiag = [Deltap Deltas1 Deltao];
V = diag(Vdiag);
Vinv = inv(V);

F1 = Ruidoes(1:4,1:5);
F2 = Ruidoes(5,1:5)*(1/epsf);
F0 = F1 - A12*Ainv*F2;
S0 = -C2*Ainv*F2;
V0 = V + S0*S0';
V0inv = inv(V0);
Id = [1 1 1 1 1];
I = diag(Id);

A_RicKs = A0 - F0*S0'*V0inv*C0;
Q_RicKs = F0*(I-S0'*V0inv*S0)*F0';
R_RicKs = V0inv;
B_RicKs = C0';
[Gks,Ps] = lqr2(A_RicKs,B_RicKs,Q_RicKs,R_RicKs);

A_RicKf = A22;
Q_RicKf = F2*F2';
R_RicKf = V0inv;
B_RicKf = C2';

```

```

[Gkf,Pf] = lqr2(A_RicKf,B_RicKf,Q_RicKf,R_RicKf);

% Calculo dos ganhos Qs e Qf

Qs = (Ps*C0'+F0*S0')*V0inv;

Qf = Pf*C2'*Vinv;

% H0 = C1 - C2 *inv(A22-P*C2)*(A21-P*C1)
% COMO P = 0 => H0 = C0
%H0 = C0;
%
% GANHO DO OBSERVADOR DO MODO RAPIDO L2
% p g. 312 n.3: precisamos garantir (A22-P2*C2) est vel, e em nosso caso
% QQC TEM QUE SER > 1. SE A22 > 0 CASO CONTRARIO QQC = 0
%QQC = 10.;
%QQC = abs ( QQC * A22 / ( C2' * C2 ) );
%P2 = QQC * C2' ;
%L2 = P2;

% GANHO DO OBSERVADOR DO MODO LENTO L1
%if magO >= 1./epsf, %esse if tinha sentido quando referenciavamos os polos em relacao
% magO = 1./epsf; %ao sistema original. Agora referenciamos cos relacao ao sistema
%end %lento e realimentado.
%Peig = [-1.0 -1.5 -2.0 -2.5]' * magO * mini ; % os polos do obs sao realacionados
%P0 = ( place(F0', H0', Peig) )' ; % ao polo mais lento do sistema
%L1 = P0 * (eye(3) - C2 * Ainv * P2) + A12 * Ainv * P2;
%
% Preparar arquivo para simulacao (CalVic) (arquivo dadvitec.mat)
L = Qs*1;
L(5,:) = Qf*1;
%
% Matrizes Fimod e Gamamod (Fi e Gama modificadas) para calculo dos estados
% estimados pelo programa do fermentador (sistema MFCS)
Lc = L * C;
Amod = A - Lc;
Bmod = [ B Lc ];
[Fimod, Gamamod] = c2d(Amod, Bmod, dtO);

% Gravação de arquivo para uso no controle do fermentador Braun (MFCS)
save c:\bio\biomfcs Ue Xe G1 G2 dtO Fimod Gamamod Fi Gama A B L;

% Gravação do arquivo para o programa de simulacao CALVIC
save c:\bio\dadvitec Ue Xe A Fi B Gama C L G1 G2 magO magR PesoR PesoC dtO Fimod Gamamod;
%
%clear magO;

```

```

c *****
c *****
c *****

```

```

c
c
c      Projeto Vitamina C
c      IPT-DME/SS-DQ/AB
c      1993

```

```

c      Programa CALVIC.EXE

```

```

c
c      Programa para simulação da etapa fermentativa da produção de vitamina C
c      sintética (conversão de sorbitol a sorbose). Trata-se, basicamente, de
c      resolver as equações diferenciais do modelo, que compreende o sistema do
c      fermentador com controlador, observador de estados, filtros digitais, etc.
c      Utiliza-se Runge-Kutta de 4ª ordem, com passo adaptativo.
c      Este programa foi idealizado para ser utilizado em conjunto com o
c      Matlab, onde se faz o projeto propriamente dito do controlador (cálculo dos
c      seus parâmetros, passados ao CALVIC por meio de um arquivo) e onde se geram
c      os gráficos do resultado da simulação.
c      O programa é composto por quatro módulos: CALPRIN.FOR -- este módulo --,
c      CALEDO.FOR, CAL-ES.FOR e CALMODEL.FOR.

```

```

c
c
c      Módulo CALPRIN.FOR

```

```

c
c      Neste módulo temos o programa principal (CALVIC) e a sub-rotina DELTAT.
c      O programa principal:
c      1 - "inicializa" alguns dados para a simulação, lidos em arquivo;
c      2 - chama rotinas dos outros módulos (para leitura de dados vindos
c          do Matlab, solução das equações do modelo);
c      3 - grava arquivo de resultados (a ser lido pelo Matlab) e fecha o
c          "arquivo de relatório da simulação";
c      4 - chama DELTAT.
c      A rotina DELTAT é chamada para o cálculo da duração total do processamento.

```

```

c *****
c *****

```

```

$declare

```

```

$debug

```

```

    program calvic

```

```

c
c      versão 6/10/95 16h17m

```

```

c 15/9/93: eliminado algoritmo relativo a "delaym", e o cálculo de tf
c 24/9: pedido de nome de arquivo de resultados logo no início do programa,
c      antes de se começar a contagem do tempo
c 29/9: nome (prefixo) para resultados volta a ser pedido no console
c 27/10: introdução de fatpond
c 28/10: mudança de fatpond para pesoprop e pesomed
c 11/11: abertura de unit 50 permitindo acesso compartilhado, abertura de
c      unit 11 no (novo) diretório \mat\xpss\BIO
c 2/12: introdução da variável atraso2
c 23/6/94: poderemos ter delay( i ) negativo
c 27/6/94: eliminada a variável "adj" (milk-shake)

```

Listagem do programa de simulação CALVIC.

- c 5/10/94: permitirei que "atraso" possa ser 0.0, ver abaixo
c 6/10: arredondamento de ndto eliminado, comentário abaixo com data

```

double precision yi( 1:10 ), ti, tf, eps, h1, hmin, dxsav,
a   xp( 1:750 ), yp( 1:10, 1:750 ), ndxsav, up( 1:4, 1:750 )
integer nok, nbad, i, j, kmax, kount, maxstp, observ, despro,
a   nest, rseed, status, dia_i, dia_f, mes, ano, hora, minuto,
b   segund, tempo( 1:4, 1:3 ), ode_st, verstp,
c   intd, maxvet, shift
double precision a( 1:5, 1:5 ), b( 1:5, 1:4 ), gs( 1:4, 1:4 ),
a   gf( 1:4 ), c( 1:3, 1:5 ), l( 1:5, 1:3 ), xe( 1:5 ),
b   ue( 1:4 ), mago, magr, pesor( 1:4 ), pesoc( 1:3 ), atraso,
c   atraso2, delay( 1:5 ), delaym, delays, umax( 1:4 ),
d   umin( 1:4 ), yl( 1:5 ), ylp( 1:5, 1:750 ), fi( 1:5, 1:5 ),
e   gama( 1:5, 1:4 ), dto, ndto, ruido, ruido2, ruido1
double precision cg, mimax, ks1, ks2, w, alfa, betam, kbetas1,
1   kbetao, ko, yxs1, yps1, yxs2, kp, yxo, ypo,
2   pesoprop( 1:5 ), pesomed( 1:5 ), deadband( 1:4 )

real frand, rand, trand
real a_mimax, a_ks1, a_ks2, a_w, a_alfa, a_betam, a_kbetas1,
1   a_kbetao, a_ko, a_yxs1, a_yxs2, a_kp, a_yxo

character*80 linha
character*8 nomearq

COMMON /PATH/ KMAX,KOUNT,DXSAV,XP,YP, MAXSTP, ylp, verstp
common /cup/ up
common /matri1/ a, b, c, l
common /matri2/ gs, gf
common /valeq0/ xe
common /valeq1/ mago, magr, pesor, pesoc
common /valeq2/ ue, deadband
common /control/ observ, atraso, umax, umin
common /estimad/ despro
common /csonda/ delay, ruido, ruido2, ruido1
common /csonda2/ delaym
common /cmaxvet/ maxvet
common /meds/ yl
common /deriva/ cg, mimax, ks1, ks2, w, alfa, betam, kbetas1,
1   kbetao, ko, yxs1, yps1, yxs2, kp, yxo, ypo,
2   trand
common /discret/ fi, gama
common /cdto/ dto
common /cndto/ ndto, shift
common /cadj/ pesoprop, pesomed
common /catraso2/ atraso2
common /ampli/ a_mimax, a_ks1, a_ks2, a_w, a_alfa, a_betam,
1   a_kbetas1, a_kbetao, a_ko, a_yxs1, a_yxs2, a_kp,
2   a_yxo

external derivs
external rkqc
data kmax /750/

```

Listagem do programa de simulação CALVIC.

c yi - vetor de estados iniciais, lidos em arquivo, (inclui estados e
 c observador de estados) **** VALORES PERCENTUAIS de erros em relação
 c ao equilíbrio
 c ti - tempo inicial para a integração
 c tf - tempo final
 c eps - precisão para cálculos
 c h1 - passo inicial (chute)
 c hmin - passo mínimo para integração
 c dxsav - intervalo mínimo de tempo entre valores a serem armazenados (h)
 c xp - vetor com valores armazenados da variável independente
 c yp - idem, variáveis dependentes
 c ndxsav - fração (ou múltiplo) de segundos, dxsav vale ndxsav segundos;
 c caso =-1., o programa calcula o intervalo adequado
 c up - vetor com valores armazenados do vetor de controle
 c nok - passos de integração que resultaram em "bons" resultados
 c nbad - idem, resultados "ruins"
 c i - uso geral
 c j - uso geral
 c kmax - número máximo de resultados a serem armazenados. Se houver mais
 c resultados que kmax, ficam guardados só os primeiros e o último
 c kount - número de resultados intermediários armazenados
 c maxstp - número máximo de passos para a solução do problema
 c observ - se =1 o controle é por realimentação de estados estimados,
 c senão por estados reais
 c despro - se =1 não nos preocupamos com o observador de estados
 c nest - número de estados a resolver (inclui observador se despro + 1)
 c tempo(i, j) - matriz de tempos: i varia de 1 a 4 para hora, minuto,
 c segundo e centésimo, e j varia de 1 a 3 respectivamente
 c para tempo inicial, final e tempo transcorrido entre 1 e 2
 c ode_st - indica "status" da rotina odeint. Se =0, final normal da rotina,
 c se =1 final por excesso de passos, se =2 final por interrupção
 c solicitada, se=3 final por passo não-significativo.
 c ndto - duração do filtro de controles (min)
 c verstp - se 1 utiliza-se "maxstp", se 0, a solução do sistema de equações
 c diferenciais não tem limitação de passos
 c a - matriz do sistema linearizado
 c b - matriz de controle (linearizada)
 c c - matriz de medidas
 c l - matriz de ganhos para o observador: note que inclui as partes
 c "lentas" e "rápida", esta já dividida por " μ "
 c xe - vetor de valores dos estados do sistema no equilíbrio
 c ue - vetor de valores dos controles do sistema no equilíbrio
 c atraso - atraso (em horas) para início da ação de controle
 c atraso2 - atraso (em horas) para início da melhoria da medição
 c delay - vetor com demoras ("atrasos") para realização de medidas dos
 c estados
 c delaym - infimo dos delay's acima
 c delays - supremo dos delay's acima
 c umax - vetor com valores maximos possiveis para os controles
 c umin - vetor com valores minimos possiveis para os controles
 c yl - vetor com valores atuais das medidas
 c ylp - vetor com valores armazenados das medidas
 c fi - matriz discretizada de a
 c gama - matriz discretizada de b
 c dto - tempo de amostragem para discretização (valor em horas)

Listagem do programa de simulação CALVIC.

c cg - concentração máxima de oxigênio dissolvido
 c mimax - máximo valor de "mi"
 c ks1 - constante para S1
 c ks2 - idem, S2
 c w - Pergunte ao Bonomi!
 c alfa -
 c betam -
 c kbetas1 -
 c kbetao -
 c ko -
 c yxs1 -
 c yps1 -
 c yxs2 -
 c kp -
 c yxo -
 c ypo -
 c linha - "string" de caracteres para cópia de arquivos, etc.
 c nomearq - nome para criação dos arquivos de resultados
 c pesoprop - peso ("ponderação") para valor de medida obtido por propagação
 c (matrizes de transferência discretizadas) (atualização)
 c pesomed - peso ("ponderação") para valor de medida obtido na medida,
 c sem atualização.
 c deadband - zona morta para as variáveis de controle

```

write( 6, 901 )
write( 6, 50 )
50 format( 5x, 'Nome do arquivo de saída -> ', \ )
read( 6, 51 ) nomearq
51 format( a8 )
open( unit = 10, file = 'bio\vit_c.dat', status = 'old',
a form = 'formatted' )
c read( 10, 51 ) nomearq
c abertura do arquivo de perturbações de parâmetros do modelo --> DEVE-SE
c usar unit = 50
open( unit = 50, file = 'bio\perturb1.dat', status = 'old',
1 form = 'formatted', mode = 'read', share = 'denywr',
2 err = 9000 )
go to 9001
9000 write( 6, 52 )
52 format( 5x, 'Er r o na abertura de perturb1. da !' )
stop ' Final anormal do programa.'
9001 call getdat( ano, mes, dia_i )
call gettim( hora, minuto, segund, i )
open( unit = 8, file = 'c:\bio\//nomearq//.txt',
1 form = 'formatted' )

c leitura dos dados do sistema: valores A, Fi, B, Gama, C, Gs, Gf, L, dtO,
c todos gravados pelo matlab no arquivo \bio\dadvitc.mat
c open ( unit = 8, file = 'c:\bio\dadosmat.txt' )
write( 8, 901 )
write( 8, 900 ) dia_i, mes, ano, hora, minuto, segund
write( 6, 901 )
write( 6, 900 ) dia_i, mes, ano, hora, minuto, segund
write( 6, 901 )
900 format(5x,'Data: ',i2.2,'/',i2.2,'/',i4,' às ',i2.2,'h',i2.2,'m',

```

Listagem do programa de simulação CALVIC.

```

1      i2.2, 's' )
901  format( 1x )

      call ledata( 8 )

      tempo( 1, 1 ) = hora
      tempo( 2, 1 ) = minuto
      tempo( 3, 1 ) = segund
      tempo( 4, 1 ) = i
c leitura dos dados da simulação no arquivo \bio\vit_c.dat
      read( 10, 100 ) ( yi( i ), i = 1, 5 )
      read( 10, 100 ) ( yi( i ), i = 6, 10 )
100  format( 7f10.3 )
101  format( f10.3 )
102  format( i10 )
103  format( 9x, f16.4 )
      read( 10, 101 ) ti
      read( 10, 101 ) tf
      read( 10, 101 ) ndto
      read( 10, 101 ) atraso
      read( 10, 101 ) atraso2
      read( 10, 101 ) eps
      read( 10, 101 ) h1
      read( 10, 101 ) hmin
      read( 10, 101 ) ndxsav
      read( 10, 102 ) maxstp
      read( 10, 102 ) verstp
      read( 10, 102 ) observ
      read( 10, 102 ) despro
      read( 10, 102 ) shift
      read( 10, 101 ) frand
      read( 10, 101 ) trand
      read( 10, 100 ) ( umax( i ), i = 1, 4 )
      read( 10, 100 ) ( umin( i ), i = 1, 4 )
      read( 10, 100 ) ( delay( i ), i = 1, 5 )
      read( 10, 100 ) ( pcsoprop( i ), i = 1, 5 )
      read( 10, 100 ) ( dcadband( i ), i = 1, 4 )
      read( 10, 100 ) a_mimax, a_ks1, a_ks2, a_w, a_alfa
      read( 10, 100 ) a_betam, a_kbetas1, a_kbetao, a_ko, a_yxs1
      read( 10, 100 ) a_yxs2, a_kp, a_yxo, ruido, ruido2
      read( 10, 101 ) ruido1
      a_mimax = a_mimax / 100.
      a_ks1   = a_ks1   / 100.
      a_ks2   = a_ks2   / 100.
      a_w     = a_w     / 100.
      a_alfa  = a_alfa  / 100.
      a_betam = a_betam / 100.
      a_kbetas1 = a_kbetas1 / 100.
      a_kbetao = a_kbetao / 100.
      a_ko    = a_ko    / 100.
      a_yxs1  = a_yxs1  / 100.
      a_yxs2  = a_yxs2  / 100.
      a_kp    = a_kp    / 100.
      a_yxo   = a_yxo   / 100.
      ruido   = ruido   / 100.

```

Listagem do programa de simulação CALVIC.

```

ruído2 = ruído2 / 100.
open( unit = 12, file = 'mat\xopss\dadovc.m', status = 'old',
a    form = 'formatted' )
read( 12, 103 ) Cg
read( 12, 103 ) mimax
read( 12, 103 ) Ks1
read( 12, 103 ) Ks2
read( 12, 103 ) w
read( 12, 103 ) alfa
read( 12, 103 ) betam
read( 12, 103 ) Kbetas1
read( 12, 103 ) Kbetao
read( 12, 103 ) Ko
read( 12, 103 ) Yxs1
read( 12, 103 ) Yps1
read( 12, 103 ) Yxs2
read( 12, 103 ) Kp
read( 12, 103 ) Yxo
read( 12, 103 ) Ypo
c conversão dos atrasos de minutos para horas
c  ndto = int(ndto/60./dto)*dto  eliminado em 6/10/95, 15h07m
ndto = ndto/60.
delaym = tf
delays = ti
maxvet = 600
do 10 i = 1, 5
    delay( i ) = delay( i )/60.
c (22-23/6/94) delay negativo significará atraso de medida e eliminar a
c propagação desta medida
    if ( delay( i ) .eq. 0. ) then
c nao faco coisa alguma
        elseif ( abs(delay( i )) .lt. dto ) then
            write( 6, 915 )
            stop ' Verifique seus dados, anta!'
915      format( 5x, 'Tempo de amostragem incompatível com',
1          ' "delays"! Programa terminado ' )
        else
c conversão de delay( i ) para múltiplo inteiro de dto
            intd = int( delay( i ) / dto + sign( 0.5, delay( i ) ) )
            if ( abs(intd) .gt. maxvet ) then
c                intd = maxvet*sign( 1, intd )
                write( 6, 916 )
                stop ' Aviso às antas muito exigentes! '
916      format( 5x, 'Atraso de HPLC incompatível com',
1          ' "memorex"! . . ' )
            endif
            delay( i ) = dto * intd
            if ( abs(delay( i )) .lt. delaym ) delaym = abs(delay( i ))
            if ( abs(delay( i )) .gt. delays ) delays = abs(delay( i ))
        endif
10 continue
c Determinacao dos atrasos de controle e medicao
delays = 6. * delays
c  if( atraso .lt. delays ) atraso = delays  desligado em 5/10/95
if( atraso2 .lt. 6*atraso ) atraso2 = 6.*atraso

```

Listagem do programa de simulação CALVIC.

```

c Determinacao de dxsav
  if ( ndxsav.eq.-1. ) then
    dxsav = ( tf - ti )/kmax
  else
    dxsav = 1./3600. * ndxsav
  endif
c (o tempo é dado em horas, vamos armazenar a cada ndxsav segundos)
  trand = trand / 60.
  if ( trand .lt. dto ) then
    trand = dto
    write( 6, 917 )
    stop ' Verifique seus dados, capadocio!'
917   format( 5x, 'Tempo de perturbação incompatível com',
1     ' "dto"! P r o g r a m a t e r m i n a d o ' )
  endif

  if ( frand.gt.0. ) then
    CALL GETTIM( status, status, status, rseed )
    frand = frand / 100.
    do 2 i = 1, 5
      CALL SEED( rseed )
      CALL RANDOM( rand )
      yi( i+5 ) = frand * ( rand - .5 ) * 2.
2    continue
  endif
  rewind( 10 )
  write( 8, 999 )
999   format( 5x, 'Dados iniciais para simulação (arquivo VIT_C.DAT)' )
3   read( 10, 1000, end = 1002 ) linha
1000  format( a80 )
      write( 8, 1001 ) linha
1001  format( 1x, a80 )
      go to 3
1002  continue
      if ( frand.gt.0. ) then
        write( 6, 1004 )
        write( 8, 1004 )
1004   format( 3x, '***** Atenção: valores aleatórios calculados ',
a       'para o estimador: *****' )
        write( 6, 1005 ) ( yi( i+5 ), i = 1, 5 )
        write( 8, 1005 ) ( yi( i+5 ), i = 1, 5 )
1005   format( 5x, 7( f10.5, 3x ) )
      endif
      close( 10 )
      close( 12 )

c cálculo dos valores absolutos de erro nos estados/observadores iniciais
  do 1 i = 1, 5
    yi( i ) = yi( i ) / 100. * xe( i )
    yi( i+5 ) = yi( i+5 ) / 100. * xe( i )
1   continue
c cálculo dos valores absolutos de saturação do controle
  do 4 i = 1, 4
    umax( i ) = ( umax( i ) / 100. ) * ue( i )
    umin( i ) = ( umin( i ) / 100. ) * ue( i )

```

Listagem do programa de simulação CALVIC.

```

4  continue
   nest = 10
   call odeint( yi, nest, ti, tf, eps, h1, hmin, nok, nbad, derivs,
a      rkqc, ode_st )

   open( unit = 11, file = 'c:\mat\xopss\bio\//nomearq//'.m',
1     form = 'formatted' )
   write( 11, 200 )
200  format( 1x, 'temp = [' )
   write( 11, 201 ) ( xp( i ), i = 1, kount )
201  format( 1x, f15.5)
   write( 11, 202 )
202  format( 1x, ' ] ;' )
   write( 11, 300 )
300  format( 1x, 'est = [' )
c vamos escrever os valores percentuais
   write( 11, 301 ) ( ( yp( j, i ) / xe( j ) * 100.,
a      j = 1, 5), i = 1, kount )
301  format( 1x, 5( e15.5, 1x ) )
   write( 11, 202 )
   write( 11, 400 )
400  format( 1x, 'obs = [' )
   write( 11, 301 ) ( ( yp( j, i ) / xe( j - 5 ) * 100.,
a      j = 6, 10), i = 1, kount )
   write( 11, 202 )
   write( 11, 500 )
500  format( 1x, 'ctr = [' )
   write( 11, 501 ) ( ( up( j, i ) / ue( j ) * 100.,
a      j = 1, 4), i = 1, kount )
501  format( 1x, 4( e15.5, 1x ) )
   write( 11, 202 )
   write( 11, 600 )
600  format( 1x, 'med = [' )
   write( 11, 301 ) ( ( ylp( j, i ) / xe( j ) * 100.,
a      j = 1, 5), i = 1, kount )
   write( 11, 202 )
c vamos escrever um nome para o "workspace" do MatLab
   linha = 'nome = '//nomearq//'';
   write( 11, 1001 ) linha

   close( 11 )

   call gettim( tempo( 1, 2 ), tempo( 2, 2 ), tempo( 3, 2 ),
1     tempo( 4, 2 ) )
   call getdat( ano, mes, dia_f )
   call deltat( tempo )
   tempo( 1, 3 ) = tempo( 1, 3 ) + 24*( dia_f-dia_i )
c (se ocorrer mudança do mês... precisamos melhorar o algoritmo!)
   write( 8, 901 )
   write( 8, 1090 ) dia_f, mes, ano, ( tempo( i, 2 ), i = 1, 3 )
   write( 8, 1100 ) ( tempo( i, 3 ), i = 1, 4 )
   write( 6, 901 )
   write( 6, 1090 ) dia_f, mes, ano, ( tempo( i, 2 ), i = 1, 3 )
   write( 6, 1100 ) ( tempo( i, 3 ), i = 1, 4 )
1090 format( 5x, 'Final da simulação em ', i2.2, '/', i2.2, '/', i4,

```

Listagem do programa de simulação CALVIC.

```

1      'às', i2.2, 'h', i2.2, 'm', i2.2, 's' )
1100 format( 5x, 'Simulação realizada em ', i3, 'h', i2.2, 'm', i2.2,
1      ' ', i2.2, 's.' )
      if ( ode_st.eq.0 ) then
          write( 6, 1101 )
          write( 8, 1101 )
      else if ( ode_st.eq.1 ) then
          write( 6, 1102 )
          write( 8, 1102 )
      else if ( ode_st.eq.2 ) then
          write( 6, 1103 )
          write( 8, 1103 )
      else if ( ode_st.eq.3 ) then
          write( 6, 1104 )
          write( 8, 1104 )
      endif
1101 format( 5x, 'Final normal da simulação' )
1102 format( 5x, 'Simulação interrompida: número de passos excessivo' )
1103 format( 5x, 'Simulação interrompida a pedido do usuário' )
1104 format( 5x, 'Simulação interrompida: 50 passos não-significati',
1      'vos consecutivos!' )

      close( 8 )
      close( 50 )
      stop ' Que tal um pró-labore para a turma da Seq'õ de Sistemas?'
      end

c *****
c *****
      subroutine deltat( tempo )
c calcula um intervalo de tempo entre dois instantes dados
c                                     versão 27/8/93 15h38m

      integer tempo( 1:4, 1:3 ), ajuste( 2:4 ), tempol( 1:4 ), i

      data ajuste / 2*60, 100 /

      do 50 i = 1, 4
          tempol( i ) = tempo( i, 1 )
50  continue
      do 100 i = 4, 1, -1
          tempo( i, 3 ) = tempo( i, 2 ) - tempo( i, 1 )
          if ( ( tempo( i, 3 ).lt.0 ).and.( i.gt.1 ) ) then
              tempo( i, 3 ) = tempo( i, 3 ) + ajuste( i )
              tempo( i-1, 1 ) = tempo( i-1, 1 ) + 1
          endif
100 continue
      do 150 i = 1, 4
          tempo( i, 1 ) = tempol( i )
150 continue
      return
      end

c *****
c *****

```

Listagem do programa de simulação CALVIC.


```

c *****
c *****
c *****
c
c
c      Projeto Vitamina C
c      IPT-DME/SS-DQ/AB
c
c      Programa CALVIC.EXE
c
c
c      Módulo CALED0.FOR
c
c   Neste módulo temos as sub-rotinas ODEINT, RK4 e RKQC.
c   Os algoritmos foram adaptados de "Numerical Recipes: the Art of
c   Scientific Computing", de William H. Press, B. P. Flannery e outros,
c   Cambridge University Press, (1986), onde estão documentadas.
c   As adaptações efetuadas permitem melhor controle da integração numérica,
c   considerando-se as peculiaridades do sistema.
c
c *****
c *****
$declare
$debug

      interface to subroutine parar[c] ( codpar )
c sub-rotina chamada em ODEINT ; interface versão 24/9/93
      integer codpar [reference]
      end

c *****
c *****
      SUBROUTINE ODEINT( YSTART,NVAR,X1,X2,EPS,H1,HMIN,NOK,NBAD,DERIVS,
*          RKQC, ode_st )

c                                versão 6/10/95 14h10m
c 8/9/93: introdução de "ode_st", atualização de "x" considerando "delay"
c 21/9: introdução de "verstp"
c 24/9: correção no cálculo de H para considerar as medições (antes calculava
c      x...); funcionamento satisfatório da sub-rotina PARAR
c 19-20/7/94: aperfeiçoamento para observador de estados discreto no tempo
c 22/12: limitador de impressão "imp"
c 18/4/95: mudança no limitador de impressão, algoritmo para imindto
c 4/7/95: retirados os estagios da integracao
c 6/10: passo "h1" negativo significa passo constante -h1

      double precision ystart, x1, x2, eps, h1, hmin, dxsav, xp, yp, x,
1      h, y, yscal, dydx, two, zero, tiny, xsav, hdid, hnext, up,
2      u, ylp, yl, delay, uxd( 1:5 ), dto, xant, xe, Ylei( 1:10 ),
3      ruido, ruido1, ruido2
      integer nvar, nok, nbad, maxstp, nmax, kmax, kount, i, nstp,
1      ode_st, verstp, codpar, despro, imp
      character*1 respos
      logical podeimp, imindto
      real rano

```

Listagem do programa de simulação CALVIC.

c ode_st - indica "status" da rotina odeint. Se =0, final normal da rotina,
 c se =1 final por excesso de passos, se =2 final por interrupção
 c solicitada (a ser implementada), se=3 final por passo não-signi-
 c ficativo.
 c uxd - vetor com o valor de "x" da última medida do estado "y(i)", no caso
 c de haver atraso nesta medida
 c respos - inicial de resposta (Sim ou Não)
 c codpar - valor devolvido pela função PARAR

```

PARAMETER (NMAX=10,TWO=2.0,ZERO=0.0,TINY=1.E-30)
COMMON /PATH/ KMAX,KOUNT,DXSAV,XP(750),YP(10,750), MAXSTP,
1      ylp( 5, 750 ), verstp
common /cup/ up( 4, 750 )
common /co/ u( 4 )
common /meds/ yl( 5 )
common /csonda/ delay( 5 )
common /cdto/ dto
common /estimad/ despro
common /dtocontr/ imindto
common /valeq0/ xe( 5 )
common /cso7nda/ ruido, ruido1, ruido2

```

```

DIMENSION YSTART(NMAX), YSCAL(NMAX), Y(NMAX), DYDX(NMAX)
EXTERNAL DERIVS
EXTERNAL V RANO
ode_st = 0
X=X1
xant = x1
podeimp = .false.
imp = 10
do 19 i = 1, 5
  uxd( i ) = x1
19 continue
H=SIGN(H1,X2-X1)
NOK=0
NBAD=0
KOUNT=0
DO 11 I=1,NVAR
  Y(I)=YSTART(I)
11 CONTINUE
IF (KMAX.GT.0) XSAV=X-DXSAV*TWO
111 DO 16 NSTP=1,MAXSTP
  CALL DERIVS(X,Y,DYDX)
  DO 12 I=1,NVAR
    YSCAL(I)=ABS(Y(I))+ABS(H*DYDX(I))+TINY
12 CONTINUE
  call parar( codpar )
  if ( codpar.eq.1 ) then
    ode_st = 2
    return
  endif
IF(KMAX.GT.0)THEN
  if ( h1 .lt. 0. ) h = -h1
  if ( (despro.eq.2) .and. ( (x+h).ge.(xant+dto) ) ) then

```

Listagem do programa de simulação CALVIC.

```

c      (observador de estados discretizado no tempo)
        h = abs( xant+dto - x )
        imindto = .true.
    else
        imindto = .false.
    endif
    IF(ABS(X-XSAV).GT.ABS(DXSAV)) THEN
        IF(KOUNT.LT.KMAX-1)THEN
            if ( imp.eq.10 ) then
                podeimp = .true.
c                write( 6, 100 ) x*3600.
                imp = 10
            else
c                imp = imp + 1
                imp = 10
            endif
100        format( t180, '+', 22x, 'Tempo = ', f12.2, ' segundos. ' )
            KOUNT=KOUNT+1
            XP(KOUNT)=X
            DO 13 I=1,NVAR
                YP(I,KOUNT)=Y(I)
13            CONTINUE
            DO 114 I=1,4
                up(I,KOUNT) = u(I)
114            CONTINUE
            DO 124 I=1,5
                ylp(I,KOUNT) = yl(I)
124            CONTINUE
            XSAV=X
        ENDIF
    ENDIF
ENDIF
20  continue
    IF((X+H-X2)*(X+H-X1).GT.ZERO) H=X2-X
    CALL RKQC( Y,DYDX,NVAR,X,H,EPS,YSCAL,HDID,HNEXT,DERIVS, ode_st )
        if ( ode_st.eq.3 ) return
        Y(1)=Y(1)+xe(1)*ruido*ran(1)
        Y(2)=Y(2)+xe(2)*ruido*ran(2)
        Y(3)=Y(3)+xe(3)*ruido*ran(3)
        Y(4)=Y(4)+xe(4)*ruido*ran(4)
        Y(5)=Y(5)+xe(5)*ruido*ran(5)
    IF(HDID.EQ.H)THEN
        NOK=NOK+1
        if ( imindto ) then
            call sondad( y, yl, x )
            Ylei(1)=Y(1)
            Ylei(2)=Y(2)+xe(2)*ruido1*ran(2)
            Ylei(3)=Y(3)+xe(3)*ruido1*ran(3)
            Ylei(4)=Y(4)
            Ylei(5)=Y(5)+xe(5)*ruido2*ran(5)
            Ylei(6)=Y(6)
            Ylei(7)=Y(7)
            Ylei(8)=Y(8)
            Ylei(9)=Y(9)
            Ylei(10)=Y(10)

```

Listagem do programa de simulação CALVIC.

```

        call obsdisc( Ylei, dydx )
        call contrs( x, Ylei, u )
        call memorex( u, x, y )
        imindto = .false.
        xant = xant + dto
    endif
ELSE
    NBAD=NBAD+1
ENDIF
    if (podeimp) then
c        xant = xant + dto
        write( 6, 100 ) x*3600.
        podeimp = .false.
    endif
    IF((X-X2)*(X2-X1).GE.ZERO)THEN
        DO 14 I=1,NVAR
            YSTART(I)=Y(I)
14        CONTINUE
        IF(KMAX.NE.0)THEN
            KOUNT=KOUNT+1
            XP(KOUNT)=X
            write( 6, 100 ) x*3600.
            DO 15 I=1,NVAR
                YP(I,KOUNT)=Y(I)
15            CONTINUE
            do 115 i = 1, 4
                up( i, kount ) = u( i )
115            continue
            do 125 i = 1, 5
                ylp( i, kount ) = yl( i )
125            continue
            ENDIF
            RETURN
        ENDIF
        IF(ABS(HNEXT).LT.HMIN) write( 6, 101 ) x*3600.
101    format( t180, '+', 'Passo menor que o mínimo, t = ', f12.2,
1        ' segundos. ' )
        H=HNEXT
16    CONTINUE
c se verstp = 0 não há limite de passos para a integração
    if ( verstp.eq.0 ) goto 111
    write( 6, 102 )
    ode_st = 1
102    format( 10x, 'Número de passos excessivo. ' )
    write( 6, 103 )
59    write( 6, 104 )
103    format( 5x, 'Quer continuar ou encerrar a simulação? ' )
104    format( 5x, 'Tecla "c" para continuar ou "e" para encerrá-la.' )
    read( 6, 105 ) respos
105    format( a1 )
    if ( (respos.eq.'c').or.(respos.eq.'C') ) then
        ode_st = 0
        go to 111
    elseif ( (respos.ne.'e').and.(respos.ne.'E') ) then
        write( 6, 106 )

```

Listagem do programa de simulação CALVIC.

```

106  format( 5x, ' Respostas v á l i d a s s ã o "c" ou "e"! ' )
      go to 59
endif

```

```

RETURN
END

```

```

c *****
c *****
SUBROUTINE RK4(Y,DYDX,N,X,H,YOUT,DERIVS)

```

```

c                                     versão 4/7/95 10h15m

```

```

c 5/8/94: chamada de perturb2 apenas na mudança de DTOs
c 4/7/95: retirado o controle do estagio3

```

```

double precision y, dydx, x, h, yout, yt, dyt, dym, hh, h6, xh
integer n, nmax, i

```

```

PARAMETER (NMAX=10)
DIMENSION Y(NMAX), DYDX(NMAX), YOUT(NMAX)
DIMENSION YT(NMAX), DYT(NMAX), DYM(NMAX)
HH=H*0.5
H6=H/6.
XH=X+HH
DO 11 I=1,N
  YT(I)=Y(I)+HH*DYDX(I)
11 CONTINUE
CALL DERIVS(XH,YT,DYT)
DO 12 I=1,N
  YT(I)=Y(I)+HH*DYT(I)
12 CONTINUE
CALL DERIVS(XH,YT,DYM)
DO 13 I=1,N
  YT(I)=Y(I)+H*DYM(I)
  DYM(I)=DYT(I)+DYM(I)
13 CONTINUE
CALL DERIVS(X+H,YT,DYT)
DO 14 I=1,N
  YOUT(I)=Y(I)+H6*(DYDX(I)+DYT(I)+2.*DYM(I))
14 CONTINUE
RETURN
END

```

```

c *****
c *****
SUBROUTINE RKQC( Y,DYDX,N,X,HTRY,EPS,YSCAL,HDID,HNEXT,DERIVS,
1  ode_st )

```

```

c                                     versão 4/7/95 10h14m

```

```

c 24/9/93: ode_st = 3 após 50 passos não significantes (antes contava até 10),
c      e impressão do tempo da ocorrência
c 10/11: limitação de passo no label 1 para "compensar" o GOTO 1 desligado
c      pois na simulação sem controle e sem observador o sistema estava
c      divergindo (passo muito grande)
c 20/7/94: introdução dos flags estagio1-3 para melhor controle da

```

Listagem do programa de simulação CALVIC.

c integração (observador discreto), yaux
 c 12/12: reativado o loop para label 1
 c 12/4/95: desligado o loop para label 1
 c 1/6: retirada a variável YAUUX (não usada)
 c 4/7: acabaram os estagios 1 a 3 (não estavam sendo usados) => simplificações

```
double precision fcor, one, safety, errcon, pgrow, pshrnk,
1  h, hh, errmax, ytemp, ysav, dysav, xsav, y, dydx, x, htry,
2  eps, yscal, hdid, hnext
```

```
integer n, nmax, i, insigp, ode_st
```

```
c insigp - contador de passos não-significativos consecutivos
PARAMETER (NMAX=10,FCOR=.0666666667,
* ONE=1.,SAFETY=0.9,ERRCON=6.E-4)
EXTERNAL DERIVS
DIMENSION Y(NMAX), DYDX(NMAX), YSCAL(NMAX)
DIMENSION YTEMP(NMAX), YSAV(NMAX), DYSAV(NMAX)
PGROW=-0.20
PSHRNK=-0.25
XSAV=X
DO 11 I=1,N
  YSAV(I)=Y(I)
  DYSAV(I)=DYDX(I)
11 CONTINUE
H=HTRY
1  if ( h.gt.10./3600. ) h = 10./3600.
  HH=0.5*H
  CALL RK4(YSAV,DYSAV,N,XSAV,HH,YTEMP,DERIVS)
  X=XSAV+HH
  CALL DERIVS(X,YTEMP,DYDX)
  CALL RK4(YTEMP,DYDX,N,X,HH,Y,DERIVS)
  X=XSAV+H
  IF(X.EQ.XSAV) then
    write( 6, 100 ) x*3600.
    insigp = insigp + 1
    if ( insigp.eq.50 ) then
      ode_st = 3
      return
    endif
  else
    insigp = 0
  endif
100 format( t180, '+', 'Passo insignificante, tempo = ', f12.2,
1  ' segundos. ')
  CALL RK4(YSAV,DYSAV,N,XSAV,H,YTEMP,DERIVS)
  ERRMAX=0.
  DO 12 I=1,N
    YTEMP(I)=Y(I)-YTEMP(I)
    ERRMAX=MAX(ERRMAX,ABS(YTEMP(I)/YSCAL(I)))
12 CONTINUE
  ERRMAX=ERRMAX/EPS
  IF(ERRMAX.GT.ONE) THEN
    H=SAFETY*H*(ERRMAX**PSHRNK)
    GOTO 1
```

Listagem do programa de simulação CALVIC.


```

ELSE
  HDID=H
  IF(ERRMAX.GT.ERRCON)THEN
    HNEXT=SAFETY*H*(ERRMAX**PGROW)
  ELSE
    HNEXT=4.*H
  ENDIF
ENDIF
DO 13 I=1,N
  Y(I)=Y(I)+YTEMP(I)*FCOR
13 CONTINUE
RETURN
END
C *****
C *****
C *****

```

```

c *****
c *****
c *****
c
c
c      Projeto Vitamina C
c      IPT-DME/SS-DQ/AB
c
c      Programa CALVIC.EXE
c
c
c      Módulo CAL-ES.FOR
c
c  Este módulo engloba as sub-rotinas LEDATA, LOADMAT e PERTURB1.
c  LEDATA lê (e grava no arquivo de relatório da simulação) os valores pas-
c  sados ao CALVIC pelo Matlab. Cada variável (escalar, matriz, vetor) é lida
c  e individualmente pela sub-rotina LOADMAT. PERTURB1 lê e interpola linear-
c  mente uma tabela (arquivo) de perturbações nos parâmetros para o modelo
c  não-linear.
c
c *****
c *****
$declare
$debug
  subroutine ledata( leunit )
c      versão 14/7/94 15h33m
c  Lê dados (matrizes/vetores do sistema linearizado, e para a
c  lei de controle e observador de estado, além dos valores no equilíbrio dos
c  estados e dos controles ) em um arquivo gravado pelo Matlab, e grava estas
c  mesmas matrizes num outro arquivo para relatório

c 14/7/94: leitura de Fimod e de Gamamod

    double precision a( 1:5, 1:5 ), b( 1:5, 1:4 ), gs( 1:4, 1:4 ),
a    gf( 1:4 ), c( 1:3, 1:5 ), l( 1:5, 1:3 ), vetr( 50 ),
b    veti( 50 ), xc, ue, mago, magr, pesor, pesoc,
c    fi( 1:5, 1:5 ), gama( 1:5, 1:4 ), dto, deadband,
e    fimod( 1:5, 1:5 ), gamamod( 1:5, 1:9 )
c  vetr - vetor para leitura de variáveis (valores reais)
c  veti - vetor para leitura de variáveis (valores imaginários)

    integer*4 MLtipo, MLnlin, MLncol, MLimag, MLncar

    character MLnome( 21 )
c  MLnome - vetor para armazenar nome de variável (incluindo espaço para
c  "null" final)

    integer i, j, k, MLok, leunit

    common /matril/ a, b, c, l
    common /matri2/ gs, gf
    common /discret/ fi, gama
    common /discretmod/ fimod, gamamod
    common /cdto/ dto
    common /valeq0/ xe( 5 )

```

Listagem do programa de simulação CALVIC.

```

common /valeq0/ xc( 5 )
common /valeq1/ mago, magr, pesor( 4 ), pesoc( 3 )
common /valeq2/ ue( 4 ), deadband( 4 )

open( unit = 9, file = 'c:\bio\dadvitc.mat', status = 'old',
1   form = 'binary' )

c leitura do vetor "Ue"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1   vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'U').or.(MLnome(2).ne.'e')
a   .or. (MLok.ne.0) ) then
    write( 6, 903 )
903   format( 1x, 'O vetor não é "Ue"!, ou houve erro )
    stop 'Chame o LuísHCQ'
  endif
  do 913 i = 1, MLnlin
    ue( i ) = vetr( k )
    k = k + 1
913 continue
  write( leunit, 199 )
  write( leunit, 900 )
900 format( 16x, 'Vetor Ue' )
  do 930 i = 1, MLnlin
    write( leunit, 201 ) ue( i )
930 continue

c leitura do vetor "Xe"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1   vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'X').or.(MLnome(2).ne.'e')
a   .or. (MLok.ne.0) ) then
    write( 6, 803 )
803   format( 1x, 'O vetor não é "Xe"!, ou houve erro )
    stop 'Chame o LuísHCQ'
  endif
  do 813 i = 1, MLnlin
    xe( i ) = vetr( k )
    k = k + 1
813 continue
  write( leunit, 199 )
  write( leunit, 800 )
800 format( 16x, 'Matriz Xe' )
  do 830 i = 1, MLnlin
    write( leunit, 201 ) xe( i )
830 continue

c leitura da matriz "A"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1   vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'A').or. (MLok.ne.0) ) then
    write( 6, 100 )

```

```

100  format( 1x, 'A matriz não é "A"!, ou houve erro )
      stop 'Chame o LuísHCQ'
endif
do 10 i = 1, MLncol
  do 20 j = 1, MLnlin
    a( j, i ) = vetr( k )
    k = k + 1
20  continue
10  continue
    write( leunit, 199 )
    write( leunit, 200 )
199  format( 1x )
200  format( 16x, 'Matriz A' )
    do 230 i = 1, MLnlin
      write( leunit, 201 ) ( a( i, j ), j = 1, MLncol )
201  format( 8( 3x, e15.9 ) )
230  continue

c leitura da matriz "Fi"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'F').or.(MLnome(2).ne.'i')
a      .or. (MLok.ne.0) ) then
    write( 6, 150 )
150  format( 1x, 'A matriz não é "Fi"!, ou houve erro )
      stop 'Chame o LuísHCQ'
endif
do 105 i = 1, MLncol
  do 205 j = 1, MLnlin
    fi( j, i ) = vetr( k )
    k = k + 1
205  continue
105  continue
    write( leunit, 199 )
    write( leunit, 3205 )
3205  format( 16x, 'Matriz Fi' )
    do 235 i = 1, MLnlin
      write( leunit, 251 ) ( fi( i, j ), j = 1, MLncol )
251  format( 8( 3x, e15.9 ) )
235  continue

c leitura da matriz "B"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'B') .or. (MLok.ne.0) ) then
    write( 6, 101 )
101  format( 1x, 'A matriz não é "B"!, ou houve erro )
      stop 'Chame o LuísHCQ'
endif
do 11 i = 1, MLncol
  do 21 j = 1, MLnlin
    b( j, i ) = vetr( k )
    k = k + 1

```

```

21  continue
11  continue
    write( leunit, 199 )
    write( leunit, 6300 )
6300 format( 16x, 'Matriz B' )
    do 6330 i = 1, MLnlin
        write( leunit, 201 ) ( b( i, j ), j = 1, MLncol )
6330 continue

c leitura da matriz "Gama"
    call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
    k = 1
    if ( (MLnome(1).ne.'G').or.(MLnome(2).ne.'a')
a      .or. (MLnome(3).ne.'m').or.(MLnome(4).ne.'a')
b      .or. (MLok.ne.0) ) then
        write( 6, 6100 )
6100  format( 1x, 'A matriz não é "Gama"!, ou houve erro )
        stop 'Chame o LuisHCQ'
    endif
    do 5100 i = 1, MLncol
        do 5200 j = 1, MLnlin
            gama( j, i ) = vetr( k )
            k = k + 1
5200  continue
5100  continue
        write( leunit, 199 )
        write( leunit, 5233 )
5233 format( 16x, 'Matriz Gama' )
        do 5230 i = 1, MLnlin
            write( leunit, 5201 ) ( gama( i, j ), j = 1, MLncol )
5201  format( 8( 3x, e15.9 ) )
5230  continue

c leitura da matriz "C"
    call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
    k = 1
    if ( (MLnome(1).ne.'C') .or. (MLok.ne.0) ) then
        write( 6, 104 )
104  format( 1x, 'A matriz não é "C"!, ou houve erro )
        stop 'Chame o LuisHCQ'
    endif
    do 14 i = 1, MLncol
        do 24 j = 1, MLnlin
            c( j, i ) = vetr( k )
            k = k + 1
24  continue
14  continue
        write( leunit, 199 )
        write( leunit, 600 )
600  format( 16x, 'Matriz C' )
        do 630 i = 1, MLnlin
            write( leunit, 201 ) ( c( i, j ), j = 1, MLncol )
630  continue

```

```

c leitura da matriz "L"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'L') .or. (MLok.ne.0) ) then
    write( 6, 107 )
107    format( 1x, 'A matriz não é "L"!, ou houve erro )
    stop 'Chame o LuísHCQ'
  endif
  do 15 i = 1, MLncol
    do 25 j = 1, MLnlin
      l( j, i ) = vetr( k )
      k = k + 1
25    continue
15  continue
  write( leunit, 199 )
  write( leunit, 700 )
700  format( 16x, 'Matriz L' )
  do 730 i = 1, MLnlin
    write( leunit, 201 ) ( l( i, j ), j = 1, MLncol )
730  continue

c leitura da matriz "G1"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'G').or.(MLnome(2).ne.'1')
a    .or. (MLok.ne.0) ) then
    write( 6, 102 )
102    format( 1x, 'A matriz não é "G1"!, ou houve erro )
    stop 'Chame o LuísHCQ'
  endif
  do 12 i = 1, MLncol
    do 22 j = 1, MLnlin
      gs( j, i ) = vetr( k )
      k = k + 1
22    continue
12  continue
  write( leunit, 199 )
  write( leunit, 400 )
400  format( 16x, 'Matriz G1' )
  do 430 i = 1, MLnlin
    write( leunit, 201 ) ( gs( i, j ), j = 1, MLncol )
430  continue

c leitura do vetor "G2"
  call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
  k = 1
  if ( (MLnome(1).ne.'G').or.(MLnome(2).ne.'2')
a    .or. (MLok.ne.0) ) then
    write( 6, 103 )
103    format( 1x, 'A matriz não é "G2"!, ou houve erro )
    stop 'Chame o LuísHCQ'

```



```

endif
do 13 i = 1, MLnlin
    gf( i ) = vetr( k )
    k = k + 1
13 continue
write( leunit, 199 )
write( leunit, 500 )
500 format( 16x, 'Matriz G2' )
do 530 i = 1, MLnlin
    write( leunit, 201 ) gf( i )
530 continue

c leitura do escalar "magO"
call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
k = 1
if ( (MLnome(1).ne.'m').or.(MLnome(2).ne.'a')
a      .or. (MLnome(3).ne.'g').or.(MLnome(4).ne.'O')
b      .or. (MLok.ne.0) ) then
    write( 6, 1003 )
1003 format( 1x, 'O escalar não é "magO"!, ou houve erro )
    stop 'Chame o LuísHCQ'
endif
do 1013 i = 1, MLnlin
    magO = vetr( k )
    k = k + 1
1013 continue
write( leunit, 199 )
write( leunit, 1000 )
1000 format( 16x, 'Valor de magO' )
write( leunit, 201 ) mago

c leitura do escalar "magR"
call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
k = 1
if ( (MLnome(1).ne.'m').or.(MLnome(2).ne.'a')
a      .or. (MLnome(3).ne.'g').or.(MLnome(4).ne.'R')
b      .or. (MLok.ne.0) ) then
    write( 6, 1103 )
1103 format( 1x, 'O escalar não é "magR"!, ou houve erro )
    stop 'Chame o LuísHCQ'
endif
do 1113 i = 1, MLnlin
    magr = vetr( k )
    k = k + 1
1113 continue
write( leunit, 199 )
write( leunit, 1150 )
1150 format( 16x, 'Valor de magR' )
write( leunit, 201 ) magr

c leitura do vetor "PesoR"
call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )

```

```

k = 1
if ( (MLnome(1).ne.'P').or.(MLnome(2).ne.'e')
a   .or. (MLnome(3).ne.'s').or.(MLnome(4).ne.'o')
b   .or. (MLnome(5).ne.'R').or.(MLok.ne.0) ) then
    write( 6, 1203 )
1203   format( 1x, 'O vetor não é "PesoR"!, ou houve erro )
        stop 'Chame o LuísHCQ'
    endif
    do 1213 i = 1, MLnlin
        pesor( i ) = vetr( k )
        k = k + 1
1213   continue
        write( leunit, 199 )
        write( leunit, 1250 )
1250   format( 16x, 'Vetor PesoR' )
        do 1253 i = 1, MLnlin
            write( leunit, 201 ) pesor( i )
1253   continue

c leitura do vetor "PesoC"
    call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1          vetr, veti, 9, MLok )
    k = 1
    if ( (MLnome(1).ne.'P').or.(MLnome(2).ne.'e')
a   .or. (MLnome(3).ne.'s').or.(MLnome(4).ne.'o')
b   .or. (MLnome(5).ne.'C').or.(MLok.ne.0) ) then
        write( 6, 1303 )
1303   format( 1x, 'O vetor não é "PesoC"!, ou houve erro )
            stop 'Chame o LuísHCQ'
        endif
        do 1313 i = 1, MLnlin
            pesoc( i ) = vetr( k )
            k = k + 1
1313   continue
            write( leunit, 199 )
            write( leunit, 1350 )
1350   format( 16x, 'Vetor PesoC' )
            do 1353 i = 1, MLnlin
                write( leunit, 201 ) pesoc( i )
1353   continue

c leitura do escalar "dtO"
    call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1          vetr, veti, 9, MLok )
    k = 1
    if ( (MLnome(1).ne.'d').or.(MLnome(2).ne.'t')
a   .or. (MLnome(3).ne.'O') .or. (MLok.ne.0) ) then
        write( 6, 7103 )
7103   format( 1x, 'O escalar não é "dtO"!, ou houve erro )
            stop 'Chame o LuísHCQ'
        endif
        do 7113 i = 1, MLnlin
            dto = vetr( k )
            k = k + 1
7113   continue

```

```

write( leunit, 199 )
write( leunit, 7150 )
7150 format( 16x, 'Valor de dtO (em minutos)' )
write( leunit, 201 ) dto*60.

c leitura da matriz "Fimod"
call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
k = 1
if ( (MLnome(1).ne.'F').or.(MLnome(2).ne.'i')
a      .or. (MLnome(3).ne.'m').or.(MLnome(4).ne.'o')
b      .or. (MLnome(5).ne.'d').or.(MLok.ne.0) ) then
write( 6, 1500 )
1500 format( 1x, 'A matriz não é "Fimod"!, ou houve erro )
stop 'Chame o LuísHCQ'
endif
do 1505 i = 1, MLncol
do 2505 j = 1, MLnlin
fimod( j, i ) = vetr( k )
k = k + 1
2505 continue
1505 continue
write( leunit, 199 )
write( leunit, 3215 )
3215 format( 16x, 'Matriz Fimod' )
do 2135 i = 1, MLnlin
write( leunit, 2510 ) ( fimod( i, j ), j = 1, MLncol )
2510 format( 8( 3x, e15.9 ) )
2135 continue

c leitura da matriz "Gamamod"
call loadmat( MLtipo, MLnlin, MLncol, MLimag, MLncar, MLnome,
1      vetr, veti, 9, MLok )
k = 1
if ( (MLnome(1).ne.'G').or.(MLnome(2).ne.'a')
a      .or. (MLnome(3).ne.'m').or.(MLnome(4).ne.'a')
b      .or. (MLnome(5).ne.'m').or.(MLnome(6).ne.'o')
c      .or. (MLnome(7).ne.'d').or.(MLok.ne.0) ) then
write( 6, 4100 )
4100 format( 1x, 'A matriz não é "Gamamod"!, ou houve erro )
stop 'Chame o LuísHCQ'
endif
do 410 i = 1, MLncol
do 420 j = 1, MLnlin
gamamod( j, i ) = vetr( k )
k = k + 1
420 continue
410 continue
write( leunit, 199 )
write( leunit, 4200 )
4200 format( 16x, 'Matriz Gamamod' )
do 4230 i = 1, MLnlin
write( leunit, 4201 ) ( Gamamod( i, j ), j = 1, MLncol )
4201 format( 10( 3x, e15.9 ) )
4230 continue

```

```

write ( leunit, 199 )
write ( leunit, 199 )

close( 9 )
return
end

c *****
c *****
      subroutine loadmat( type, mrows, ncols, imagf, namlen, name,
$          rpart, ipart, lunit, rdflg )
c
c LOADMAT - A Fortran subroutine that reads a double-precision matrix
c   from a data file in the .MAT-file format used by MATLAB.
c
c See also:
c   - TESTLS1.FOR and TESTLS2.FOR, for examples of using LOADMAT
c   - The SAVEMAT.FOR file
c   - TESTLS.C, SAVEMAT.C, and LOADMAT.C, for C-language programs
c   - LOAD in the reference section of the MATLAB User's Guide, for
c     detailed information on the file format of M-files.
c   - Note that the LOAD/SAVE commands in MATLAB can now directly
c     read and write ASCII files.
c
c These subroutines are known to work with Microsoft Fortran
c versions 3.0 and higher. They are not thought to work with
c PRO-FORT, RM, or Layhey Fortran compilers. A compiler
c specific open option allows arbitrary byte streams to be
c written.
c
c Description of inputs:
c
c   lunit   - logical unit number of input file
c
c
c Description of outputs:
c
c   type    - matrix type flag; considering the type flag as a
c             decimal integer the ones decimal place is used to
c             indicate numeric or textual interpretation of the
c             matrix data (0 for numeric or 1 for textual);
c             the 1000's decimal place is used to indicate the
c             machine format for the matrix data (0 for Intel 8086
c             based machines, 1 for Motorola 68000 based machines,
c             and 2 for Vax d format). A flag of 1000 indicates
c             numeric data in a 68000 machine format and a flag of
c             1001 indicates textual interpretation of the 68000
c             machine format data.
c   mrows   - number of rows in matrix
c   ncols   - number of columns in matrix
c   imagf    - imaginary flag; 0 for no imaginary part or 1 if the
c             data has an imaginary part
c   namlen   - number of characters in matrix name plus 1 (for zero
c             byte string terminator)

```

Listagem do programa de simulação CALVIC.

```

c      name      - character array holding the matrix name. (Be sure
c                  character array has room for the zero byte terminator)
c      rpart     - real part of matrix (mrows x ncols double precision
c                  elements stored column wise)
c      ipart     - imaginary part of matrix (only used if imagf = 1)
c      lunit     - logical fortran unit
c      rdflg     - read status flag; = 0 good read, = 1 end of file,
c                  = -1 error reading file
c
c  NOTES :
c
c      o THE OUTPUT FILE MUST BE OPENED WITH THE FOLLOWING STATEMENT:
c        open(unit=lunit, file='file_name', form='binary')
c
c      o Each call to this routine will read another MATLAB
c        matrix until an error occurs or an end-of-file is reached.
c
c      o Be sure that the 4 byte integer option is used when compiling
c        this routine and that LOADMAT's integer arguments are 4 byte.
c
c  Author: S.N. Bangert 5-31-85
c  Revised 10-27-88 SNB
c
c  20 byte header
c      integer type, mrows, ncols, imagf, namlen, i, mn
c
c  character string for name (length of name plus one)
c      character name(*)*1
c
c  double precision data arrays for example
c      double precision rpart(*), ipart(*)
c
c  output file logical unit number
c      integer lunit
c
c
c  read flag
c      integer rdflg
c
c  zero byte that follows name
c      integer*1 zerobyte
c
c  read header
c      read(lunit,err=998,end=999) type,mrows,ncols,imagf,namlen
c
c  read variable name
c      read(lunit,err=998,end=999) (name(i),i=1,namlen-1),zerobyte
c      mn = mrows*ncols
c
c  read real part of variable data
c      read(lunit,err=998,end=999) (rpart(i),i=1,mn)
c
c  read complex part of variable data if present
c      if ( imagf .eq. 1 ) then
c        read(lunit,err=998,end=999) (ipart(i),i=1,mn)
c      end if

```

```

c
c set read flag to ok and return
      rdflg = 0
      return

c
c error during read
998      rdflg = -1
      return

c
c end of file
999      rdflg = 1
      return
      end

c *****
c *****
      subroutine perturb1( t, unit )

c   Lê na unidade unit (na verdade com o valor 50) o arquivo de perturbações
c   dos parâmetros do modelo não-linear. O arquivo consiste numa tabela com o
c   tempo na primeira coluna, e depois as variações percentuais nos parâmetros,
c   a primeira linha deve corresponder ao tempo 0.0, e a última a um tempo
c   *maior* que o tempo final da simulação. O arquivo é aberto em CALPRIN.

c                                     versão 8/11/93  8h54m

      double precision dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1          dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2          dkp, dyxo, dypo, t, ta, td, dcga, dmimaxa, dks1a,
3          dks2a, dwa, dalfaa, dbetama, dkbetas1a, dkbetaoa,
4          dkoa, dyxs1a, dyps1a, dyxs2a, dkpa, dyxoa, dypoa,
5          dcgd, dmimaxd, dks1d, dks2d, dwd, dalfad,
6          dbetamd, dkbetas1d, dkbetaod, dkod, dyxs1d,
7          dyps1d, dyxs2d, dkpd, dyxod, dypod, deltat

      logical fl_ini /.true./
      integer unit

      common /cperturb1/ dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1          dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2          dkp, dyxo, dypo

c   dcga - "delta cg" antes
c   dcgd - "delta cg" depois
c   DEMAIS VARIÁVEIS - ANALOGAMENTE

100  format( f10.6, 16( f5.1 ) )

      if ( fl_ini ) then
          fl_ini = .false.
          dcgd = 0.
          dmimaxd = 0.
          dks1d = 0.
          dks2d = 0.
          dwd = 0.

```

Listagem do programa de simulação CALVIC.


```

dalfad = 0.
dbetamd = 0.
dkbetas1d = 0.
dkbetaod = 0.
dkod = 0.
dyxs1d = 0.
dyps1d = 0.
dyxs2d = 0.
dkpd = 0.
dyxod = 0.
dypod = 0.
read( unit, 100 ) td, dcgd, dmimaxd, dks1d, dks2d, dwd,
1      dalfad, dbetamd, dkbetas1d, dkbetaod, dkod,
2      dyxs1d, dyps1d, dyxs2d, dkpd, dyxod, dypo
endif

if ( t .gt. td ) then
10    ta = td
      dcga = dcgd
      dmimaxa = dmimaxd
      dks1a = dks1d
      dks2a = dks2d
      dwa = dwd
      dalfaa = dalfad
      dbetama = dbetamd
      dkbetas1a = dkbetas1d
      dkbetaoa = dkbetaod
      dkoa = dkod
      dyxs1a = dyxs1d
      dyps1a = dyps1d
      dyxs2a = dyxs2d
      dkpa = dkpd
      dyxoa = dyxod
      dypoa = dypod
      read( unit, 100 ) td, dcgd, dmimaxd, dks1d, dks2d, dwd,
1      dalfad, dbetamd, dkbetas1d, dkbetaod, dkod,
2      dyxs1d, dyps1d, dyxs2d, dkpd, dyxod, dypo
      if ( t .gt. td ) go to 10
      deltat = t - ta
      dcg = dcga + deltat * ( dcgd - dcga )
      dmimax = dmimaxa + deltat * ( dmimaxd - dmimaxa )
      dks1 = dks1a + deltat * ( dks1d - dks1a )
      dks2 = dks2a + deltat * ( dks2d - dks2a )
      dw = dwa + deltat * ( dwd - dwa )
      dalfa = dalfaa + deltat * ( dalfad - dalfaa )
      dbetam = dbetama + deltat * ( dbetamd - dbetama )
      dkbetas1 = dkbetas1a + deltat * ( dkbetas1d - dkbetas1a )
      dkbetao = dkbetaoa + deltat * ( dkbetaod - dkbetaoa )
      dko = dkoa + deltat * ( dkod - dkoa )
      dyxs1 = dyxs1a + deltat * ( dyxs1d - dyxs1a )
      dyps1 = dyps1a + deltat * ( dyps1d - dyps1a )
      dyxs2 = dyxs2a + deltat * ( dyxs2d - dyxs2a )
      dkp = dkpa + deltat * ( dkpd - dkpa )
      dyxo = dyxoa + deltat * ( dyxod - dyxoa )
      dypo = dypoa + deltat * ( dypod - dypoa )

```

Listagem do programa de simulação CALVIC.

endif

return
end

Listagem do programa de simulação CALVIC.

```

c *****
c *****
c *****
c
c
c
c      Projeto Vitamina C
c      IPT-DME/SS-DQ/AB
c
c      Programa CALVIC.EXE
c
c
c      Módulo CALMODEL.FOR
c
c  Neste módulo temos as rotinas DERIVS, SONIDAD, CONTRS, FILTRU e MEMOREX,
c  relativas ao modelo físico simulado.
c  A sub-rotina DERIVS calcula derivadas dos estados do sistema (dinâmica
c  do sistema), sendo chamada passo a passo na solução das equações pelo mó-
c  dulo CALEDO. Chama CONTRS, que calcula, com base nos estados (estimados ou
c  não) do sistema, a ação de controle. Como há um observador de estado, DERIVS
c  engloba a dinâmica do observador, de modo a permitir sua solução simultanea-
c  mente com as equações do sistema; a estimação dos estados requer algumas
c  medidas do sistema, realizadas por SONIDAD, uma "sonda discreta" que faz as
c  medidas e, como estas têm atrasos, propaga ("atualiza") os seus valores de
c  forma a ter um valor "mais provável" do que seria a medida em tempo real.
c  FILTRU simula um filtro para os controles "u", de modo que a resposta
c  dos controles não apresente variações muito rápidas, capazes de desestabi-
c  lizar o sistema.
c  MEMOREX armazena um pequeno histórico dos controles e estados medidos,
c  sendo tais valores utilizados na propagação das medidas por SONIDAD.
c
c *****
c *****
$declare
$debug
  subroutine derivs( X, Y, DYDX )
c  fornece as derivadas do sistema linearizado (ou não-linear) e do
c  observador de estado

c
c      versão 5/7/95 10h29m

c 19-20/7/94: aperfeiçoamento do observador discreto ("estagio3")
c 5/8: chamada de perturb2 em CALEDO (rotina RK4)

double precision x, y( 1:10 ), dydx( 1:10 ),
1      u ( 1:4 ), yl, beta, mi,
2      fimod, gamamod
double precision a, b, c, l
double precision cg, mimax, ks1, ks2, w, alfa, betam, kbetas1,
1      kbetao, ko, yxs1, yps1, yxs2, kp, yxo, ypo,
2      ad( 1:5 ), bd( 1:5 ), dd( 1:5 ), xe, ue, deadband
double precision dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1      dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2      dkp, dyxo, dypo, pesomed, pesoprop, delay
c , ruido,
c 3      ruido2

```

Listagem do programa de simulação CALVIC.

```

integer i
real trand, xant

common /matri1/ a(5,5), b(5,4), c(3,5), l(5,3)
common /discretmod/ fimod(5,5), gamamod(5,9)
common /co/ u
common /meds/ yl( 5 )
common /deriva/ cg, mimax, ks1, ks2, w, alfa, betam, kbetas1,
1      kbetao, ko, yxs1, yps1, yxs2, kp, yxo, ypo,
2      trand
common /valeq0/ xe( 5 )
common /valeq2/ ue( 4 ), deadband( 4 )
common /cperturb1/ dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1      dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2      dkp, dyxo, dypo
common /cadj/ pesoprop( 5 ), pesomed( 5 )
common /cso7nda/ delay(5)
c , ruido, ruido2

data xant/0./

c x - variável independente (tempo) - ---- ATENÇ ÃO ----
c y - vetor das variáveis dependentes (estados do sistema, em nosso caso)
c yl - medidas vindas da sondad
c med - vetor de medidas do sistema
c ymod - "medidas modificadas" para observador de estados discretizado
c ad - constantes auxiliares
c bd - constantes auxiliares
c dd - constantes auxiliares
c dcg - variação no valor do parâmetro Cg ("DeltaCG")
c dmimax - idem, mimax
c dks1 - " ks1
c dks2 - " ks2
c dw - " w
c dalfa - " alfa
c dbetam - " betam
c dkbetas1 - " dkbetas1
c dkbetao - " dkbetao
c dko - " dko
c dyxs1 - " dyxs1
c dyps1 - " dyps1
c dyxs2 - " dyxs2
c dkp - " dkp
c dyxo - " dyxo
c dypo - " dypo

c simulação do sistema com modelo não-linear
if ( trand.gt.0. ) then
  if ( (x-xant) .ge. trand ) then
    xant = x
c calcular incrementos nos parâmetros aleatoriamente, para testar robustez
    call perturb2
  endif
else
  call perturb1( x, 50 )

```

Listagem do programa de simulação CALVIC.

c (observação: a unidade 50 é aberta e fechada no módulo CALPRIN)
 endif

```
mi = ( mimax*( 1.+dmimax )) * ( y( 3 ) + xe( 3 ) ) /
1  ( ks1 * ( 1.+dks1 ) + y( 3 ) + xe( 3 ) +
2  ( ( y( 3 ) + xe( 3 ) )**2 ) /
3  ( w*( 1.+dw )) * ( y( 4 ) + xe( 4 ) ) / ( ks2 * ( 1.+dks2 )
4  + y( 4 ) + xe( 4 ) ) * ( y( 5 ) + xe( 5 ) ) / ( ko * ( 1.+dko )
5  + y( 5 ) + xe( 5 ) ) * exp( -( Kp * ( 1.+dKp ) ) *
6  ( y( 2 ) + xe( 2 ) ) )
```

```
beta = ( betam * ( 1.+dbetam )) * ( y( 3 ) + xe( 3 ) ) /
1  ( kbetas1 * ( 1.+dkbetas1 ) + y( 3 ) + xe( 3 ) ) *
2  ( y( 5 ) + xe( 5 ) ) / ( kbetao * ( 1.+dkbetao ) +
3  y( 5 ) + xe( 5 ) )
```

```
dd( 1 ) = - ( y( 1 ) + xe( 1 ) ) * ( u( 1 ) + ue( 1 ) )
dd( 2 ) = - ( y( 2 ) + xe( 2 ) ) * ( u( 1 ) + ue( 1 ) )
dd( 3 ) = ( u( 2 ) + ue( 2 ) - y( 3 ) - xe( 3 ) ) *
1  ( u( 1 ) + ue( 1 ) )
```

```
dd( 4 ) = ( u( 3 ) + ue( 3 ) - y( 4 ) - xe( 4 ) ) *
1  ( u( 1 ) + ue( 1 ) )
```

```
dd( 5 ) = ( cg * ( 1.+dcg ) - y( 5 ) - xe( 5 ) ) *
1  ( u( 1 ) + ue( 1 ) + u( 4 ) + ue( 4 ) )
```

```
ad( 1 ) = 1.
ad( 2 ) = alfa * ( 1. + dalfa )
ad( 3 ) = -( 1/( Yxs1*(1.+dyxs1) ) + (alfa*(1.+dalfa))/
1  ( Yps1*(1.+dyps1) ) )
ad( 4 ) = -( 1/( Yxs2*(1.+dyxs2) ) )
ad( 5 ) = -( 1/( Yxo * (1.+dyxo ) ) + (alfa*(1.+dalfa) ) /
1  ( Ypo * (1.+dypo ) ) )
bd( 1 ) = 0.
bd( 2 ) = 1.
bd( 3 ) = -1/( Yps1 * (1.+dyps1) )
bd( 4 ) = 0.
bd( 5 ) = -1/( Ypo * (1.+dypo ) )
```

```
do i = 1, 5
  dydx( i ) = ( ad( i ) * mi + bd( i ) * beta ) *
1  ( y( 1 ) + xe( 1 ) )
  dydx( i ) = dydx( i ) + dd( i )
end do
```

```
return
end
```

```
c *****
c *****
  subroutine obsdisc( y,dydx )
c
c      versão 4/7/95 *** 10h30m
c  calcula o observador de estados discretizado no tempo
c 1/6: variáveis MED e X eliminadas (não eram usadas)
```

Listagem do programa de simulação CALVIC.

```

double precision y( 1:10 ), dydx( 1:10 ),
1      u ( 1:4 ), yl, delay,
c  ruido, ruido2,
2      fimod, gamamod, ymod( 1:5 )
integer i, j, despro

```

```

common /discretmod/ fimod(5,5), gamamod(5,9)
common /co/ u
common /meds/ yl( 5 )
common /csonda/ delay(5)
c  ruido, ruido2
common /estimad/ despro

```

C Escolha do metodo para estimativa dos estados do sistema

c 1. Colocaremos os estados reais no lugar dos respectivos estados estimados

```

if ( despro .eq. 1 ) then
do 15 i = 1, 5
y( i + 5 ) = y( i )
dydx( i + 5 ) = 0.
15 continue

```

c 2. Observador discreto "Gambiberger"

```

else if ( despro .eq. 2 ) then
do 23 i = 1, 5
ymod( i ) = yl( i )
if ( delay( i ) .eq. 0. ) then
ymod( i ) = 0.
do 22 j = 1, 4
ymod( i ) = fimod(i,j) * y( j+5 )
1      + gamamod(i,j) * u( j )
2      + gamamod(i,j+4) * yl( j )
3      + ymod( i )
22 continue
ymod( i ) = fimod(i,5) * y(10)
1      + gamamod(i,9) * yl(5)
2      + ymod( i )
endif
y( i+5 ) = ymod( i )
dydx( i+5 ) = 0.0
23 continue

```

c 3. Calculo nao possivel de realizar, valor incorreto de despro

```

else
write( 6, 101 )
101 format( 5x, 'Valor incorreto no parâmetro "despro"! )
stop ' Final do programa!'
endif

return
end

```

c *****

c *****

subroutine sondad (y, yl, x)

c determina os valores de medição levando-se em conta o atraso

c dos mesmos. SONDA DISCRETA

c versão 4/7/95 10h50m

double precision ymemo(1:2, 1:5), y(1:10), yl(1:5),

1 xmemo(1:2, 1:5), x, xant(1:5), xult(1:5),

2 dto, delay, umem, ymem, fi, gama, ydant(1:5),

3 pesoprop, pesomed, x_aux, y_aux, xe

c 4 ruido2, ruido

integer i, j, contmem, qqpos, kao, k, kao1, sub, intd, maxvet

logical fl_ini

c real rano

common /sonda/ delay(5)

c , ruido, ruido2

common /cmemorex/ umem(1:4, 1:600), ymem(1:5, 1:600), contmem

common /cmaxvet/ maxvet

common /discret/ fi(5, 5), gama(5, 4)

common /cdto/ dto

common /cadj/ pesoprop(5), pesomed(5)

common /valeq0/ xc(5)

data fl_ini /.true./

if (fl_ini) then

fl_ini = .false.

do 8 i = 1, 5

if(pesoprop(i).gt.1.) pesoprop(i) = 1.

if(pesoprop(i).lt.0.) pesoprop(i) = 0.

pesoprop(i) = 1.

pesomed(i) = 1. - pesoprop(i)

xmemo(1, i) = 0.

ymemo(1, i) = 0.

xult(i) = 0.

xant(i) = 0.

yl(i) = 0.

x_aux = 0.

y_aux = 0.

ydant(i) = y(i)

c ymem(i, 1) = y(i)

8 continue

return

endif

do 11 i = 1, 5

c 0. medidas não propagadas (modelo LHCQ)

if (delay(i).lt.0.) then

c 01. grande não propagação de medida no intervalo delay/dto

if ((x-xant(i)) .ge. (-delay(i))) then

xant(i) = x

intd = int(-delay(i) /dto)

Listagem do programa de simulação CALVIC.

```

      qqpos = contmem - intd
      if ( qqpos.le.0 ) qqpos = qqpos + maxvet
      ymem( i, qqpos ) = ydant( i )
      ydant( i ) = y( i )
      + xe( i ) * ruido * rano( i )
c
      do 101 k = 1, 2 * intd
      sub = 0
      if ( (qqpos+k) .gt. maxvet ) sub = maxvet
      kao = qqpos + k - sub
      kao1 = kao - 1
      if ( kao1.eq. 0 ) kao1 = maxvet
      ymem( i, kao ) = ymem( i, kao1 )
101      continue
      yl( i ) = ymem( i, kao )
      xult( i ) = x
c      03. medida é a última medição
      else
      yl( i ) = ymemo( 1, i )
      xult( i ) = xmemo( 1, i )
      endif
c      1. medidas inexistentes
      elseif ( delay( i ).eq.0. ) then
      yl( i ) = 0.0
      xult( i ) = x
c      3. medida é a última medição
      elseif ( ( x-xult( i ) ).lt.dto ) then
      yl( i ) = ymemo( 1, i )
      xult( i ) = xmemo( 1, i )
c      4. propagação de uma medida no intervalo dto
      elseif ( ( x-xant( i ) ).lt.delay( i ) ) then
      qqpos = contmem
      sub = 0
      if ( qqpos.eq.maxvet ) sub = maxvet
      kao = qqpos + 1 - sub
      kao1 = kao - 1
      if ( kao1.eq. 0 ) kao1 = maxvet
      ymem( i, kao ) = 0.
      do 103 j = 1, 4
      ymem( i, kao ) = ymem( i, kao )
1      + fi( i, j ) * ymem( j, kao1 )
2      + gama( i, j ) * umem( j, kao1 )
103      continue
      ymem( i, kao ) = ymem( i, kao )
1      + fi( i, 5 ) * ymem( 5, kao1 )
      yl( i ) = ymem( i, kao )
      xult( i ) = x
c      5. grande propagação de medida no intervalo delay/dto
      else
      xant( i ) = x
      intd = int( delay( i )/dto )
      qqpos = contmem - intd
      if ( qqpos.le.0 ) qqpos = qqpos + maxvet
      ymem( i, qqpos ) = ydant( i )
      ydant( i ) = y( i )
      + xe( i ) * ruido * rano(i)
c

```

Listagem do programa de simulação CALVIC.

```

do 104 k = 1, intd + 1
  sub = 0
  if ( (qqpos+k) .gt. maxvet ) sub = maxvet
  kao = qqpos + k - sub
  kao1 = kao - 1
  if ( kao1.eq. 0 )    kao1 = maxvet
  yl( i ) = ymem( i, kao )
  ymem( i, kao ) = 0.
  do 105 j = 1, 4
    ymem( i, kao ) =      ymem( i, kao )
1      + fi( i, j ) * ymem( j, kao1 )
2      + gama( i, j ) * umem( j, kao1 )
105    continue
    ymem( i, kao ) =      ymem( i, kao )
1      + fi( i, 5 ) * ymem( 5, kao1 )
    ymem( i, kao ) = ymem( i, kao ) * pesoprop( i )
1      + yl( i )      * pesomed( i )
104    continue
    yl( i ) = ymem( i, kao )
    xult( i ) = x
  endif
  xmemo( 1, i ) = xult( i )
  ymemo( 1, i ) = yl( i )
11 continue
return
end

c *****
c *****
  subroutine contrs( x, y, u )
c   fornece a ação de controle ao sistema (realimentando os estados
c   estimados, se observ=1, ou os reais)
c                                     versão 05/07/94 13h51m
c 14/10/93: chamada de MEMOREX feita em DERIVS

  double precision x, y( 10 ), u( 4 ), gs, gf, atraso,
1      umax, umin, yh, xe( 5 )
  integer observ, i, j

  common /matri2/ gs( 4, 4 ), gf( 4 )
  common /control/ observ, atraso, umax( 4 ), umin( 4 )
  common /valeq0/ xe

c 1. Sistema sem controle ou antes do atrasador
  if ( ( x.le.atraso ).or.( observ.eq.1 ) ) then
    do 11 i = 1, 4
      u( i ) = 0.
11    continue

c 2. Realiza o controle realimentando estados estimados e razoaveis
  else if ( observ.eq.2 ) then
    do 13 i = 1, 4
      u( i ) = 0.
    do 12 j = 1, 4
      yh = y(j+5)

```

```

        if ( yh .gt. xe(j) ) yh = xe(j)
        if ( yh .lt. -xe(j) ) yh = -xe(j)

        u( i ) = gs( i, j ) * yh + u( i )
12      continue
        yh = y(5+5)
        if ( yh .gt. xe(5) ) yh = xe(5)
        if ( yh .lt. -xe(5) ) yh = -xe(5)

        u( i ) = gff( i ) * yh + u( i )
13      continue

c 3. Controle por realimentação de estados reais
      else if ( observ.eq.3 ) then
        do 15 i = 1, 4
          u( i ) = 0.
          do 14 j = 1,4
            yh = y(j)
            u( i ) = gs( i, j ) * yh + u( i )
14          continue
            yh = y(5)
            u( i ) = gff( i ) * yh + u( i )
15          continue

        else
c 4. Saida por erro de dados na entrada
          write( 6, 100 )
100      format( 15x, 'Arquivo de dados para simulação errado!' )
          stop ' Final anormal do programa! '
        endif

        call filtru( u, x )
        return
        end

c *****
c *****
      subroutine filtru( u, x )
c Filtra os valores de controle de maneira a atenuar as altas frequências,
c tendo como entrada os valores instantâneos do controlador teórico. Filtro
c digital "Zero-Order Holder".

c                                     versão 4/7/95 11h00m

c 4/7: eliminada a "volta no tempo"

      double precision u( 1:4 ), x, xant, ultx, dt, dtant, tac, atraso,
1          integ( 1:4 ), intant( 1:4 ), conmed( 1:4, 1:2 ),
2          tacant, delaym, deadzone, ue( 4 ), deadband( 4 ),
3          umax, umin, dto, ndto, delay
c , ruido, ruido2

      integer i, observ, shift
      logical fl_ini, atuei

```

Listagem do programa de simulação CALVIC.

```

c   real rano

c   tac - tempo acumulado
c   conmed( i, j ) - controle médio para variável i; j=1 para valor atual
c                       e j=2 para valor anterior (usado em restaurações)

```

```

common /cdto/ dto
common /cndto/ ndto, shift
common /csonda2/ delaym
common /valeq2/ ue, deadband
common /control/ observ, atraso, umax( 4 ), umin( 4 )
common /csonda/ delay( 5 )
c   , ruido, ruido2

data fl_ini /.true./

if( fl_ini ) then
  fl_ini = .false.
  xant = x
  dtant = 0.
  tac = 0.
  do 10 i = 1, 4
    integ( i ) = 0.
    conmed( i, 1 ) = 0.
    conmed( i, 2 ) = 0.
10  continue
endif
dt = x - ultx

atuei = .false.
c segue o processamento convencional
do 30 i = 1, 4
  intant( i ) = integ( i )
  integ( i ) = integ( i ) + u( i ) * dt
30  continue
tacant = tac
tac = tac + dt
xant = ultx
ultx = x
c cálculo dos valores de controle
if( tac .ge. ndto ) then
  atuei = .TRUE.
  do 40 i = 1, 4
    conmed( i, 2 ) = conmed( i, 1 )
    conmed( i, 1 ) = integ( i )/tac
    integ( i ) = 0.
    deadzone = abs( conmed( i, 1 ) )/ ue( i ) * 100.
    if( conmed( i, 1 ).gt.umax( i ) ) conmed( i, 1 ) = umax( i )
    if( conmed( i, 1 ).lt.umin( i ) ) conmed( i, 1 ) = umin( i )
    if( deadzone.lt.deadband( i ) ) conmed( i, 1 ) = 0.
    conmed( i, 1 ) = conmed( i, 1 )
c + ue( i ) * ruido2 * rano(i)
40  continue
  tac = 0.
endif

```

Listagem do programa de simulação CALVIC.

```

do 50 i = 1, 4
  u( i ) = conmed( i, 1 )
50 continue

return
end

c *****
c *****
  subroutine memorex( u, x, y )
c  Armazena valores de controles e estados estimados a cada dto
c                                versão 01/01/95 13h35m
c 2/12: introdução da variável atraso2
  double precision umem( 1:4, 1:600 ), ymem( 1:5, 1:600 ), x, tac,
  1      u( 1:4 ), y( 1:10 ), dto, delay, ndto, xant, xult,
  2      sum(5), ymemo(5), atraso2, ue, deadband
c  3      ruido2, ruido
  integer contmem, j, maxvet, atrasado, shift
  logical fl_ini /.true./

  common /cmemorex/ umem, ymem, contmem
  common /cmaxvet/ maxvet
  common /cdto/ dto
  common /cndto/ ndto, shift
  common /csonda/ delay( 5 )
c  , ruido, ruido2
  common /catraso2/ atraso2
  common /valeq2/ ue( 4 ), deadband( 4 )

  if ( fl_ini ) then
    contmem = 1
    xant = x
    xult = x
    tac = 0.0
    fl_ini = .false.
  endif

  if ( (x-xant).ge.dto ) then
    xant = x
    contmem = contmem + 1
    atrasado = contmem - shift
    if ( contmem .gt. maxvet ) contmem = 1
    if ( atrasado .gt. maxvet ) atrasado = atrasado - maxvet
    if ( atrasado .lt. 1 ) atrasado = atrasado + maxvet
    do 100 j = 1, 4
      umem( j, atrasado ) = u( j )
100 continue

c  Tentativa de melhoria das propagacoes com insercao de valores
c  mais provaveis das variaveis nao medidas no memorex {ymem} !
    do 200 j = 1, 5
      if ( x .gt. atraso2 ) then
        ymemo(j) = sum(j) / tac
      else

```

Listagem do programa de simulação CALVIC.


```

        ymemo(j) = 0.
    endif
    if (delay(j) .eq. 0.) ymem( j, contmem ) = ymemo(j)
    sum(j) = 0.
200    continue
    tac = 0.
    xult = x

else
    do 300 j = 1, 5
        sum(j) = sum(j) + (x-xult) * y( j+5 )
300    continue
    tac = tac + (x-xult)
    xult = x

endif

return
end

c *****
c *****
c      subroutine perturb2
c                      versão 9/8/97 14h00m
c 9/8/94: amplitudes de variação individuais

real rano

double precision dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1      dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2      dkp, dyxo, dypo
real a_mimax, a_ks1, a_ks2, a_w, a_alfa, a_betam, a_kbetas1,
1      a_kbetao, a_ko, a_yxs1, a_yxs2, a_kp, a_yxo

common /cperturb1/ dcg, dmimax, dks1, dks2, dw, dalfa, dbetam,
1      dkbetas1, dkbetao, dko, dyxs1, dyps1, dyxs2,
2      dkp, dyxo, dypo
common /ampli/ a_mimax, a_ks1, a_ks2, a_w, a_alfa, a_betam,
1      a_kbetas1, a_kbetao, a_ko, a_yxs1, a_yxs2, a_kp,
2      a_yxo

dcg    = 0.    ! oxigênio dissolvido máximo é constante
dmimax = a_mimax * rano( 1 )
dks1   = a_ks1 * rano( 2 )
dks2   = a_ks2 * rano( 3 )
dw     = a_w * rano( 4 )
dalfa  = a_alfa * rano( 5 )
dbetam = a_betam * rano( 6 )
dkbetas1 = a_kbetas1 * rano( 7 )
dkbetao = a_kbetao * rano( 8 )
dko    = a_ko * rano( 9 )
dyxs1  = a_yxs1 * rano( 10 )
dyps1  = 0.    ! fator estequiométrico
dyxs2  = a_yxs2 * rano( 11 )
dkp    = a_kp * rano( 12 )

```

Listagem do programa de simulação CALVIC.

```

dyxo  = a_yxo * rano( 13 )
dypo  = 0.  ! fator estequiométrico

return
end

c *****
c *****
function rano(idum)
c          versão 4/8/97  14h45m
c retorna um numero aleatorio uniforme nao inviesado entre zero e um,
c usando a funcao random. idum negativo inicializa, ou reinicializa a sequencia.
integer idum, iff, iseed, j
real v(97), randval, y
data iff /0/
if ( (idum.lt.0) .or. (iff.eq.0) ) then
  iff = 1
  iseed = abs(idum)
  call seed(iseed)
  idum = 1
  do 11 j = 1, 97
    call random(randval)
11  continue
  do 12 j = 1, 97
    call random(randval)
    v(j) = randval
12  continue
  call random(randval)
  y = randval
endif
j = 1 + int(97. * y) + idum
if( j.lt.1 ) pause 'algo de errado no mundo do caos'
if( j.gt.97) j = 98 - idum
y = v(j)
call random(randval)
v(j) = randval
rano = 2. * ( y - 0.5 )

return
end

c *****
c *****
c *****

```

Arquivo VIT_C.DAT:

```

-20.0  -20.0  -20.0  -20.0  -20.0  estados iniciais
0.00   0.00   0.00   0.00   0.00   observador inicial
0.00   tempo inicial (hora)      hoje 27/11/95 14h50m
100.00 tempo final  (hora)
3.00   duração do filtro do atuador (min)
20.00  atraso para início do controle (horas)
400.00 atraso para melhoria da medição (horas)
1.e-6  precisão
1.0e-5 passo inicial (chute)
1.e-8  menor passo admissível (hora)
-1.0   tempo entre resultados armazenados (segundos)
400000 número máximo de passos
0      =1 simula-se até o máximo de passos; =0 não há limite
2      =1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)
2      =1 observador é o estado real; =2 obs discreto (despro)
0      numero inteiro de dto's para atraso dos contr na memorex (shift)
0.0    erro máximo aleatório para observadores iniciais (%)
360000.0 tempo de aleatorização dos parâmetros (min);
10.0   10.0   10.0   10.0   valores máximos de controle (%)
-10.0  -10.0  -10.0  -10.0  valores mínimos de controle (%)
0.0    30.0   30.0   0.0    -0.25 atrasos das medidas (min)
1.0    1.0    1.0    1.0    1.0    fatores de ajuste da propagação
0.5    0.5    0.5    0.0    0.5    deadband (%) p/ atuadores(4) e estados(1)
0.0    0.0    0.0    0.0    0.0    _μ0, _Ks1, _Ks2, _W, __ (%)
0.0    0.0    0.0    0.0    0.0    _β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0    0.0    0.0    12.0   7.0    _Yxs2, _Kp, _YxO, ruído, ruído2 (%)
5.0                                ruído1 (%)

```

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor Igual ao módulo do passo inicial

Resultados

Diversas simulações foram realizadas no sentido de se obter resultados satisfatórios. Os resultados iniciais se mostraram longe do esperado. É bastante claro que a planta estava estabilizando em outro ponto de equilíbrio. A presença do ruído é facilmente notada pela oscilação de todas as variáveis.

As primeiras simulações foram baseadas nos seguintes valores de ruído:

- 12% de ruído em todas as variáveis de estado
- 5% de ruído nos valores medidos de Sorbitol e Sorbose
- 7% de ruído nos valores medidos de Oxigênio

Esses valores são baseados na observação da realidade. Os 12% impostos às variáveis de estado devem-se ao fato de que o modelo é extremamente impreciso. Os outros valores de ruído são causados pela precariedade da instrumentação disponível.

Diante do fato de que os resultados eram inaceitáveis, a primeira tentativa no sentido de fazer o sistema convergir, foi se reduzir os erros impostos às variáveis. A tentativa nada mais era do que a modificação do modelo no sentido do melhorá-lo.

A observação dos resultados é bastante clara ao demonstrar que o desvio em regime em relação ao ponto de equilíbrio aumentou com a diminuição do nível de ruído. Todas as variáveis de estado se distanciaram do ponto de equilíbrio.

Outra tentativa se baseou no fato que a estratégia adota considerava o erro percentual durante a evolução do sistema, constante. Considera-se que esta hipótese não é verdadeira e a partir disto tentou-se mudar os ganhos do filtro de Kalman de forma a ele fazer as estimativas baseadas em um nível real de ruído.

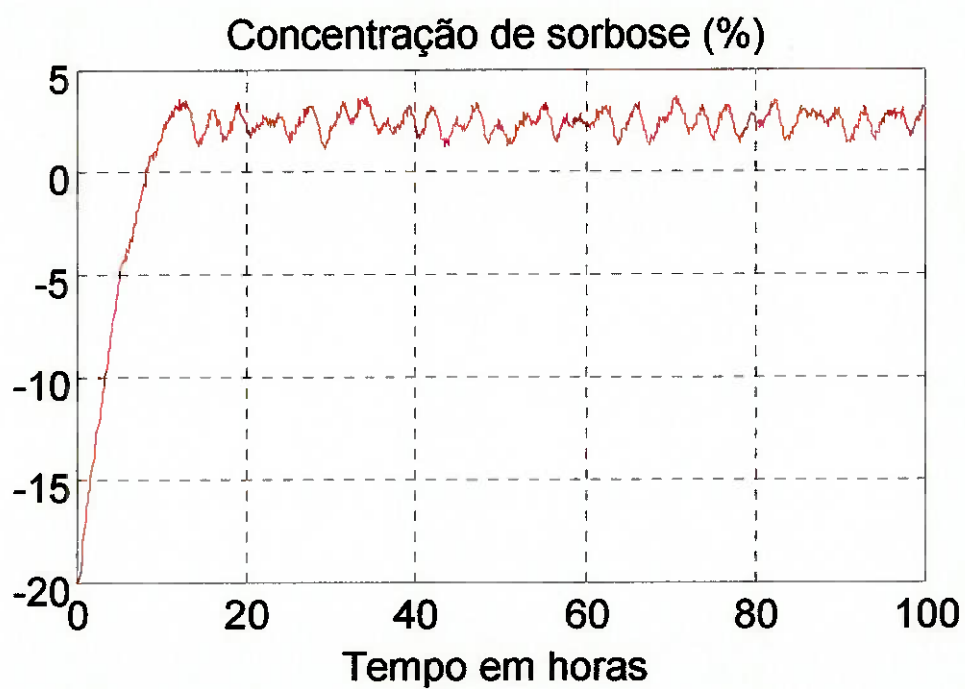
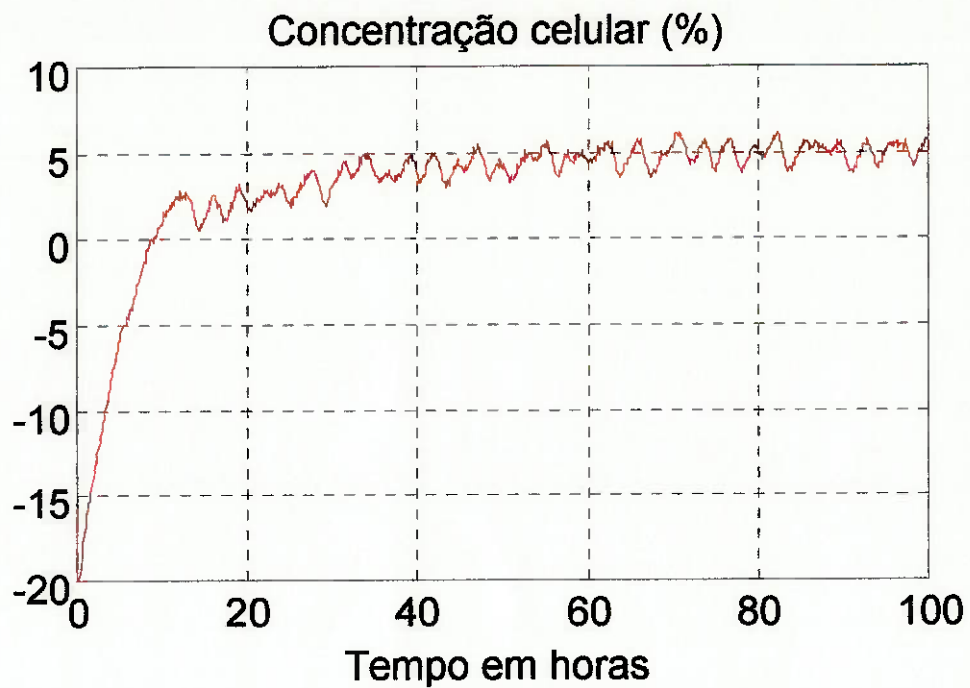
Ao mesmo tempo que se fez esta modificação, se colocou também, um atraso no início o controle de 20 horas. Isso é possível porque o sistema é estável. Esta última solução se mostrou satisfatória e foi mantida como uma das soluções definitivas. Em relação à alteração do ganho do filtro de Kalman durante a simulação, pode-se dizer que trata-se de uma alternativa que pode apresentar resultados aproveitáveis. No nosso caso, como demonstra os resultados das simulações, não se conseguiu chegar a um valor que contribuísse para a convergência do sistema. Mas a pesquisa sobre essa alternativa não foi explorada de maneira completa e não se pode afirmar que tal estratégia não traz resultados.

As últimas tentativas de melhora dos resultados foram baseados na observação de que as entradas de controle atingiam os pontos de saturação com muita frequência durante a evolução da planta. Assim a alternativa era mudar empiricamente os valores limites para essas variáveis. A idéia inicial era retirar o limite de forma a deixar o sistema livre. Simulações foram feitas adotando esta idéia, como os resultados só se distanciavam do ideal, foi natural tentar-se restringir a ação das variáveis de controle, o que apresentou resultados surpreendentes. Todas as simulações anteriores estavam baseados no limite de 50% dos valores de equilíbrio como valor de saturação. Nos resultados finais este limite foi reduzido para 10%.

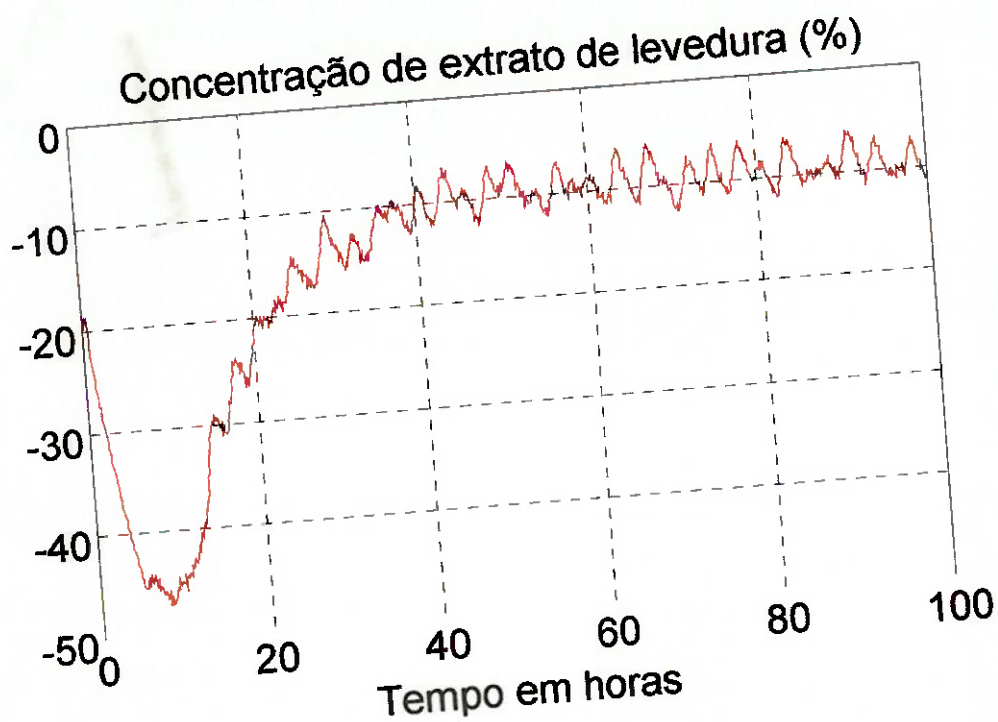
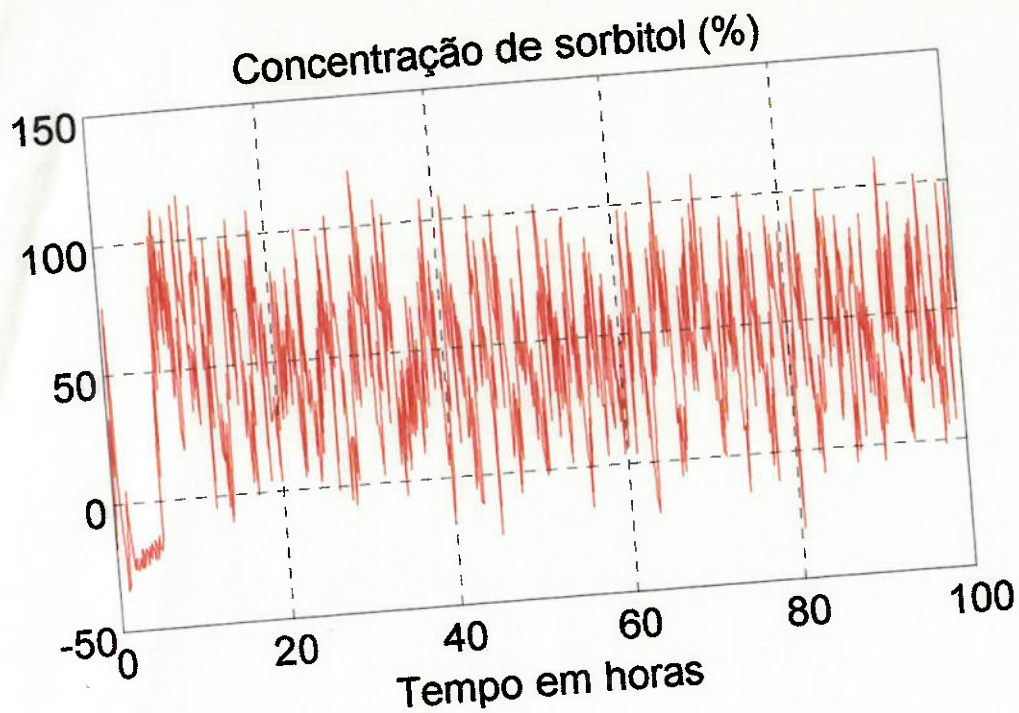
Todos os resultados descritos acima estão apresentados nas páginas seguintes.

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

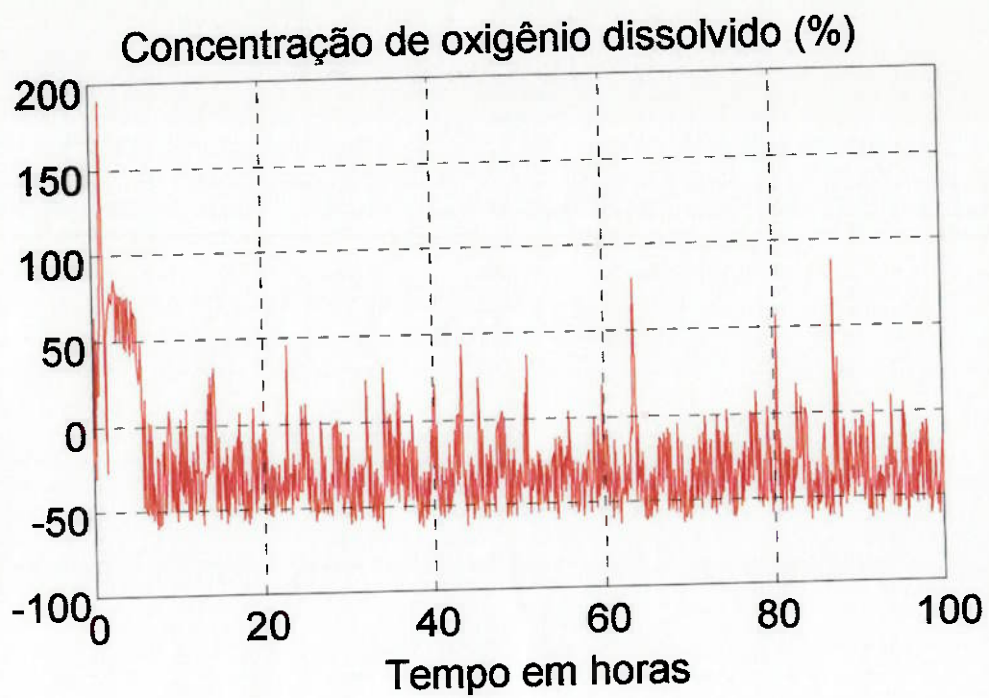
Variáveis de Estado:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

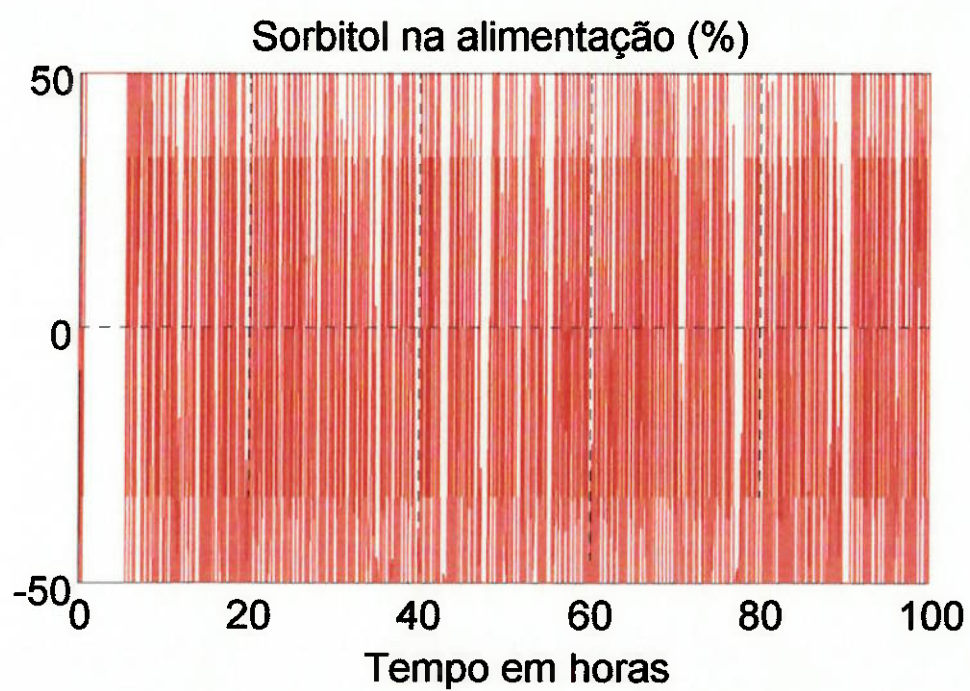
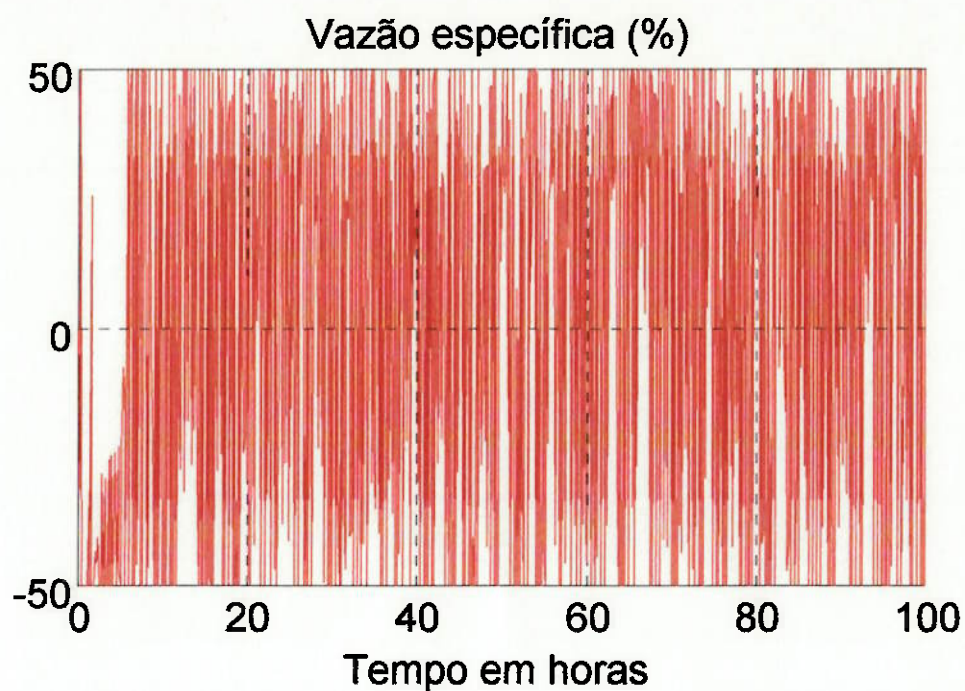


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

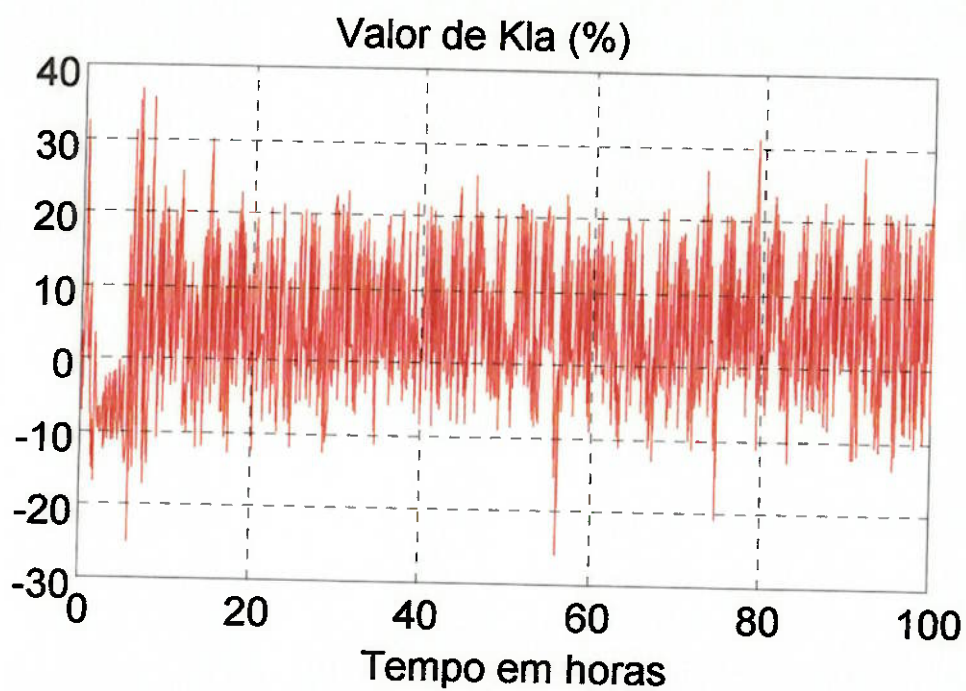


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Variáveis de Controle:

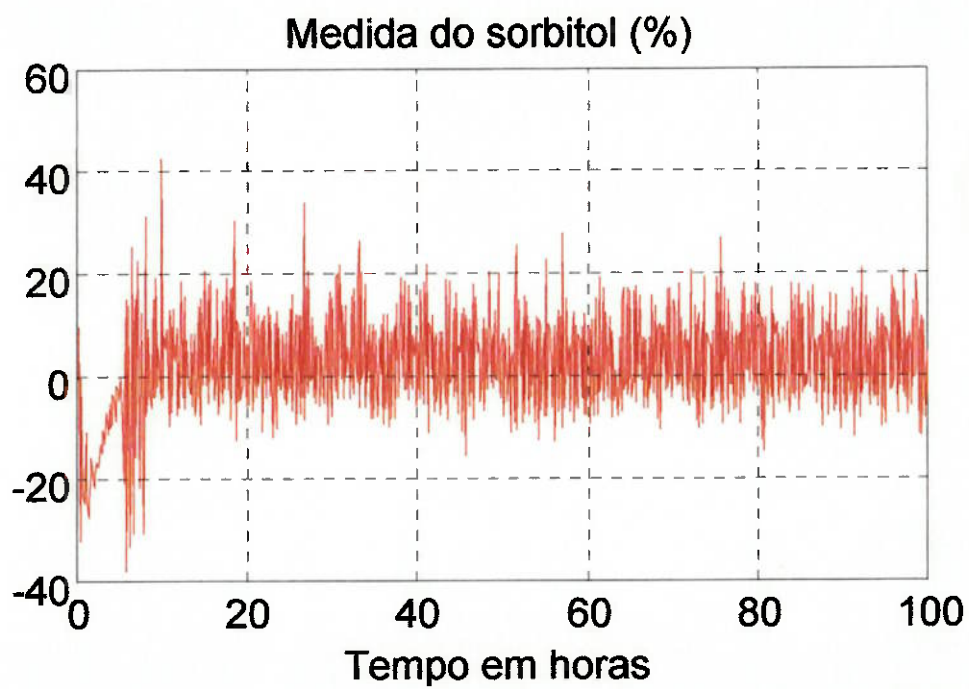
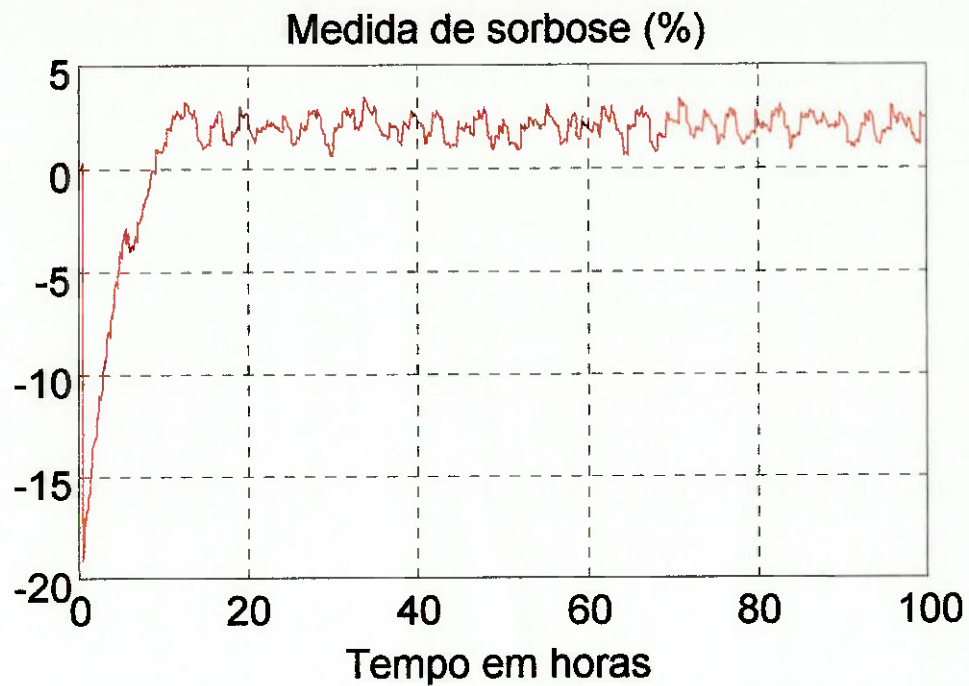


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

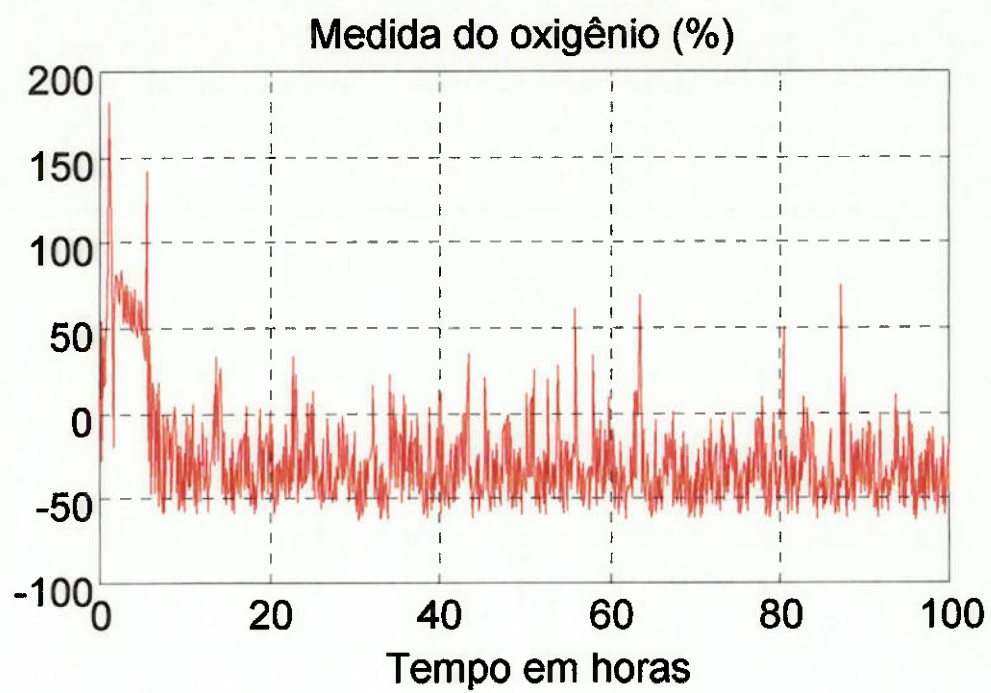


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Variáveis Medidas:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xe - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00	-.123794160E-02	.707919687E-01	.176010163E-01	.812277235E-01
.393306572E+01	-.115528924E+00	.359136491E+01	.505149169E+00	.558033286E+01
-.431723616E+01	.411919255E-01	-.401255173E+01	-.585665554E+00	-.598804914E+01
-.847457627E-01	.131137881E-02	-.749914922E-01	-.986451444E-01	-.860463173E-01
-.955666447E+03	.125364720E+02	-.855535230E+03	-.178243181E+03	-.143892113E+04

Matriz Fi

.999981172E+00	-.784903394E-06	.421348740E-04	.111597328E-04	.393510318E-04
.198542078E-02	.999920663E+00	.183598939E-02	.180172673E-03	.270530063E-02
-.221106331E-02	.161531388E-04	.997897628E+00	-.229664841E-03	-.290284112E-02
-.506743625E-04	.831465460E-06	-.446344004E-04	.999921514E+00	-.416854150E-04
-.463296555E+00	.608081778E-02	-.414739698E+00	-.864568839E-01	.302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01	.000000000E+00	.000000000E+00	.000000000E+00
-.116870000E+03	.000000000E+00	.000000000E+00	.000000000E+00
.128284000E+03	.800000000E-01	.000000000E+00	.000000000E+00
.251000000E+01	.000000000E+00	.800000000E-01	.000000000E+00
.693600000E+01	.000000000E+00	.000000000E+00	.693600000E+01

Matriz Gama

-.197813202E-02	.155751693E-08	.403827121E-09	.136024765E-06
-.972653142E-01	.716993845E-07	.819115086E-08	.934885760E-05
.106764081E+00	.665853199E-04	-.100006689E-07	-.100316676E-04
.208880533E-02	-.164991200E-08	.666640167E-04	-.144094031E-06
-.230530652E-01	-.165312790E-04	-.344534729E-05	.336908405E-02

Matriz C - Matriz de Medições de estados do modelo linearizado

.000000000E+00	.684521263E-01	.000000000E+00	.000000000E+00	.000000000E+00
.000000000E+00	.000000000E+00	.466200466E+01	.000000000E+00	.000000000E+00
.000000000E+00	.000000000E+00	.000000000E+00	.000000000E+00	.709219858E+00

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Matriz L - Ganho do Filtro de Kalman

.465760675E+00 -.224123860E+02 .114531090E+00
.975836222E+00 -.349756170E+02 .786826933E+01
-.513546323E+00 .251201092E+02 -.844314929E+01
.379678399E-01 -.287699813E+01 -.121325307E+00
.000000000E+00 .000000000E+00 .160744419E+01

Matriz G1 - Ganho do Controlador para o subsistema lento

.158744793E-01 .993981208E-02 -.330529012E+00 .136667582E-02
-.965998529E+01 -.179734813E+02 -.273850235E+04 -.163525113E+01
-.134234327E+01 .169365076E-01 -.476839228E-03 -.290876994E+00
.435123781E+02 -.578435716E+01 .207306491E+03 .904427383E+01

Matriz G2 - Ganho do Controlador para o subsistema rápido

-.310214934E-04
.000000000E+00
.000000000E+00
-.103944787E+03

Valor de magO - Velocidade do Observador

.300000000E+01

Valor de magR - Magnitude do Controlador

.500000000E-01

Vetor PesoR - Controle

.400000000E+01
.800000000E+00
.200000000E+01
.200000000E+02

Vetor PesoC - Medidas

.800000000E+01
.800000000E+01
.400000000E+00

Valor de dtO (em minutos)

.500000000E-01

Matriz Fimod

.999848371E+00 -.249162074E-04 .828721152E-01 -.590258823E-05 -.200629427E-13
.303553040E-02 .999852304E+00 .132217891E+00 .388135339E-03 .468637453E-12
-.342179254E-02 .605560643E-04 .903846351E+00 -.464200999E-03 -.518055570E-12
-.899526998E-04 -.729859306E-06 .105680944E-01 .999915176E+00 -.114153485E-13
-.462850818E+00 .607497374E-02 -.414343325E+00 -.863737242E-01 .301178861E+00

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Matriz Gamamod

.252483276E-02	.280896018E-05	.279785817E-10	-.325459060E-16	.000000000E+00
.253322152E-04	-.829540523E-01	.000000000E+00	-.142626085E-03	
-.902031488E-01	.448001361E-05	.132965596E-07	.167764397E-14	.000000000E+00
.537341222E-04	-.129449934E+00	.000000000E+00	.431469009E-02	
.101679895E+00	.634092732E-04	-.157341122E-07	-.183233274E-14	.000000000E+00
.279081151E-04	.929732408E-01	.000000000E+00	-.474618075E-02	
.266616533E-02	.358244295E-06	.666638678E-04	-.347485402E-16	.000000000E+00
.200717244E-05	-.106485569E-01	.000000000E+00	-.985218481E-04	
-.230423954E-01	-.165225768E-04	-.344352103E-05	.336584553E-02	.000000000E+00
.486931705E-08	.265398677E-05	.000000000E+00	.179180810E-02	

Dados iniciais para simulação (arquivo VIT_C.DAT):

-20.0	-20.0	-20.0	-20.0	-20.0	estados iniciais
0.00	0.00	0.00	0.00	0.00	observador inicial
0.00	tempo inicial (hora)				hoje 27/11/95 14h50m
100.00	tempo final (hora)				
3.00	duração do filtro do atuador (min)				
0.00	atraso para início do controle (horas)				
400.00	atraso para melhoria da medição (horas)				
1.e-6	precisão				
1.0e-5	passo inicial (chute)				
1.e-8	menor passo admissível (hora)				
-1.0	tempo entre resultados armazenados (segundos)				
400000	número máximo de passos				
0	=1 simula-se até o máximo de passos; =0 não há limite				
2	=1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)				
2	=1 observador é o estado real; =2 obs discreto (despro)				
0	numero inteiro de dto's para atraso dos contr na memorex (shift)				
0.0	erro máximo aleatório para observadores iniciais (%)				
360000.0	tempo de aleatorização dos parâmetros (min);				
50.0	50.0	50.0	50.0	valores máximos de controle (%)	
-50.0	-50.0	-50.0	-50.0	valores mínimos de controle (%)	
0.0	30.0	30.0	0.0	-0.25	atrasos das medidas (min)
1.0	1.0	1.0	1.0	1.0	fatores de ajuste da propagação
0.5	0.5	0.5	0.0	0.5	deadband (%) p/ atuadores(4) e
0.0	0.0	0.0	0.0	0.0	μ_0 , K_{s1} , K_{s2} , W , γ (%)
0.0	0.0	0.0	0.0	0.0	β_0 , β_{s1} , β_O , K_O , Y_{xs1} (%)
0.0	0.0	0.0	12.0	7.0	Y_{xs2} , K_p , Y_{xO} , ruído, ruído
5.0					ruído1 (%)

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

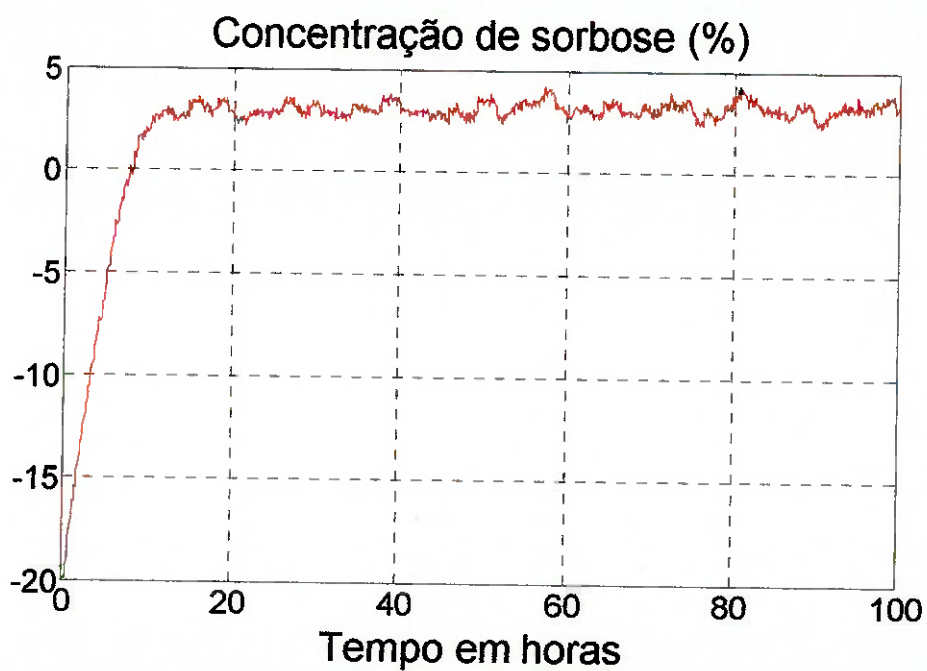
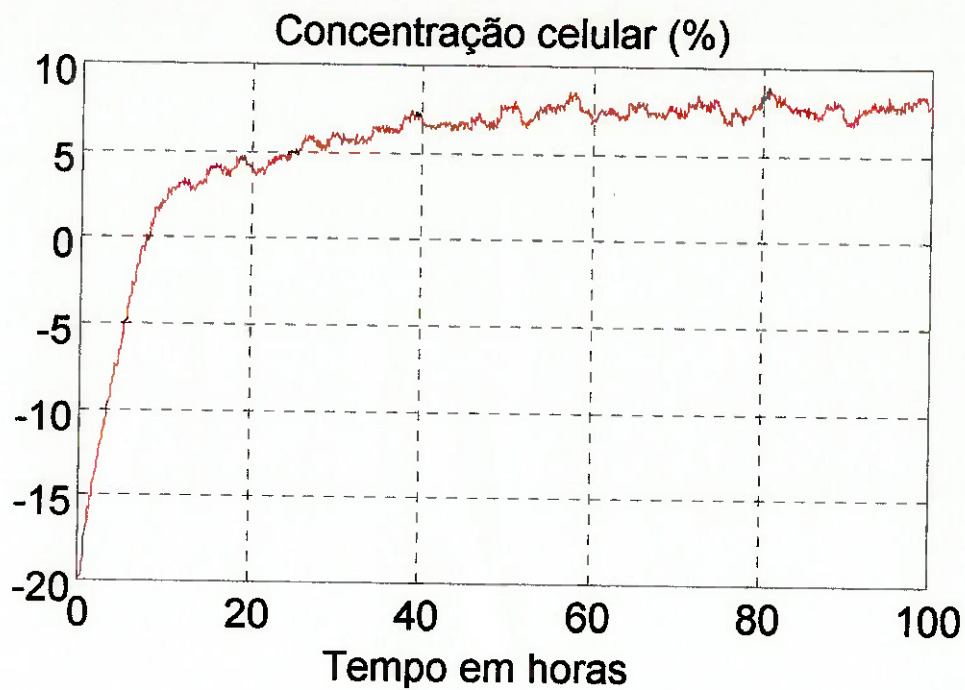
Simulação realizada em 0h11m19.37s.

Final normal da simulação

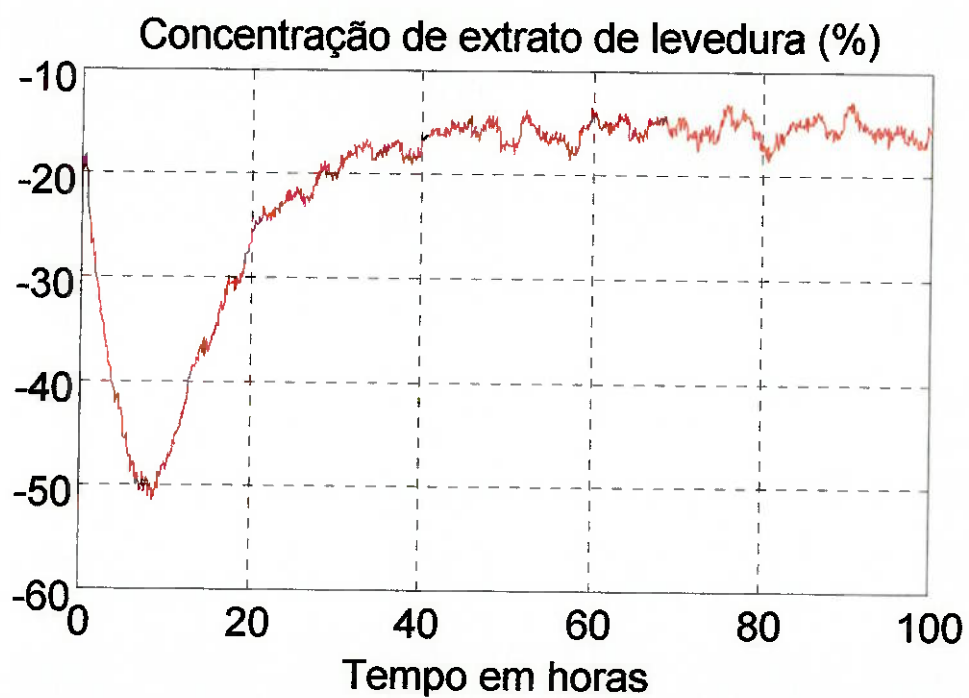
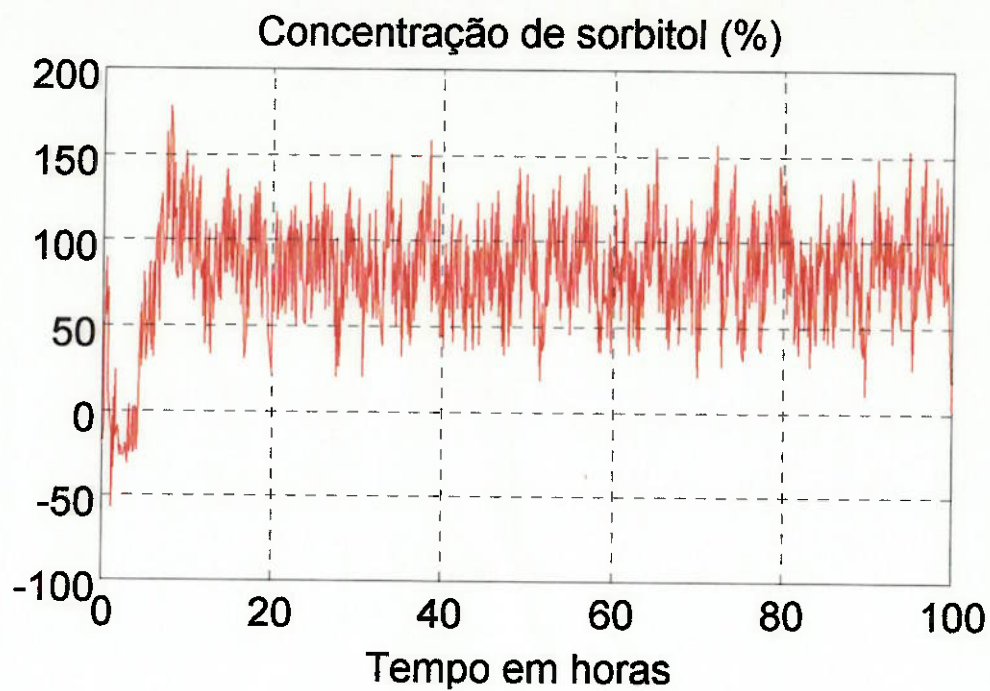
Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio.

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas

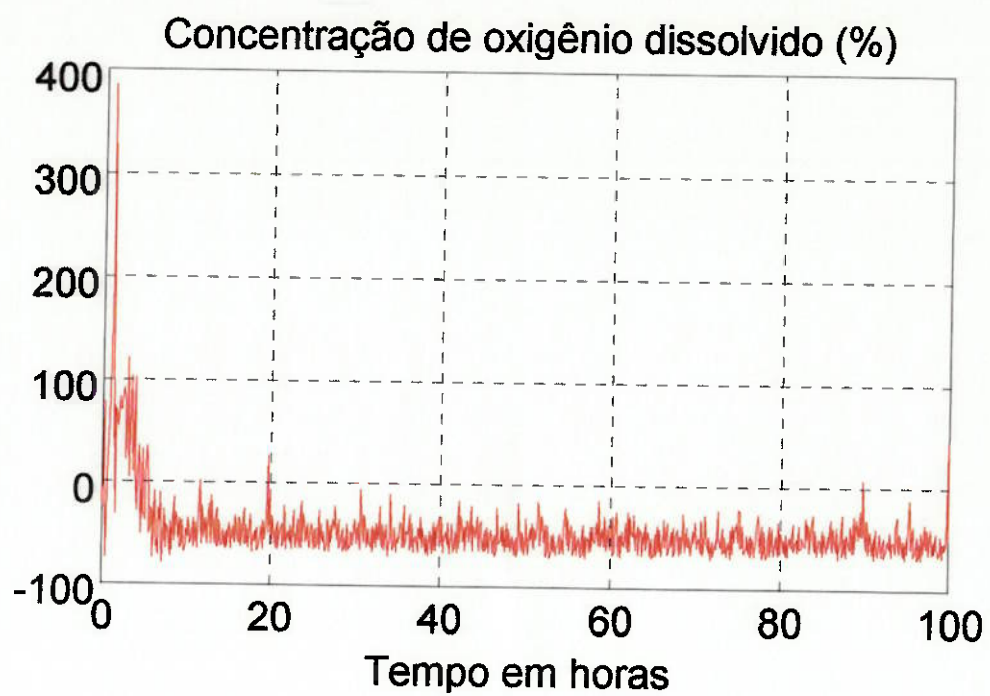
Variáveis de Estado:



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

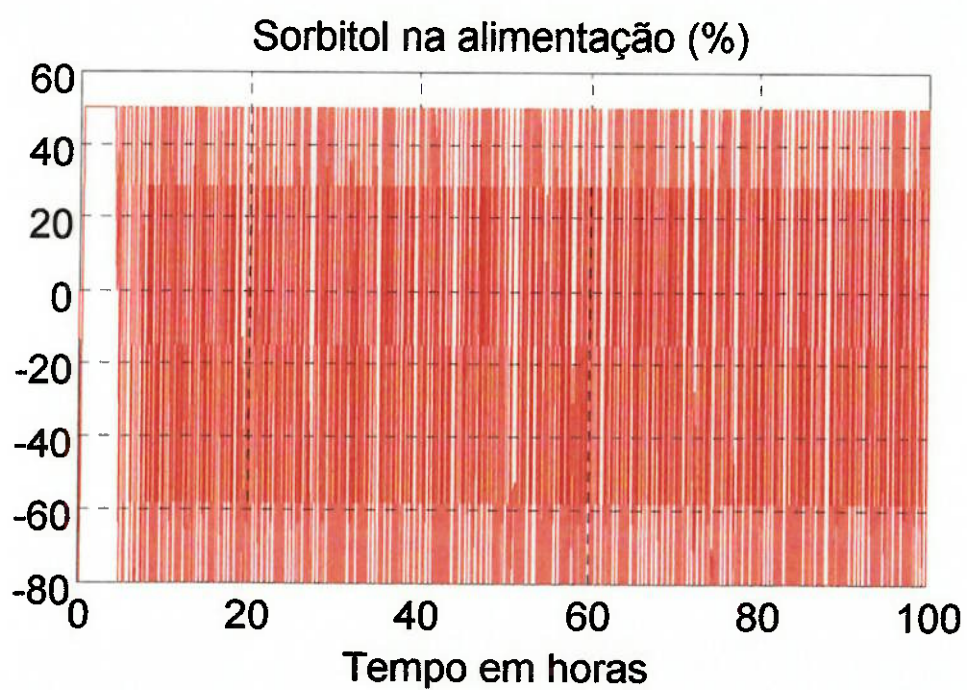
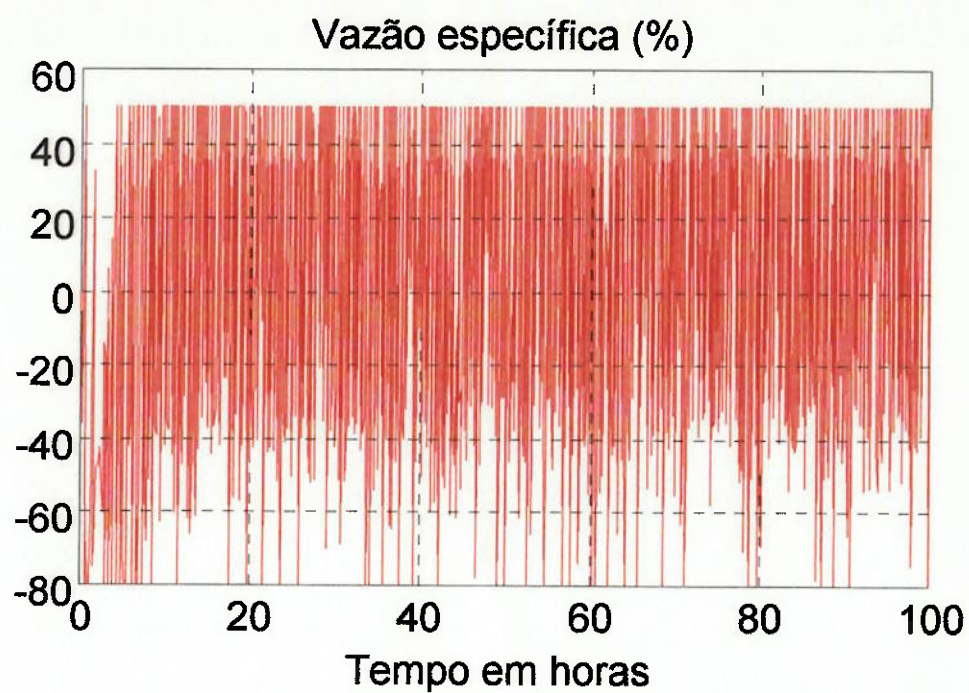


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

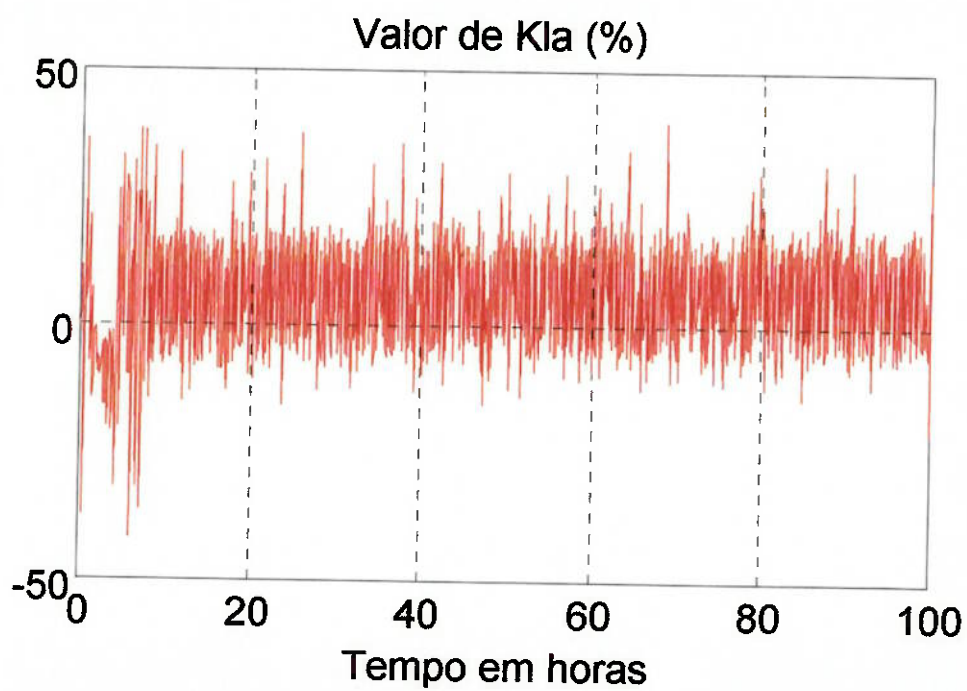
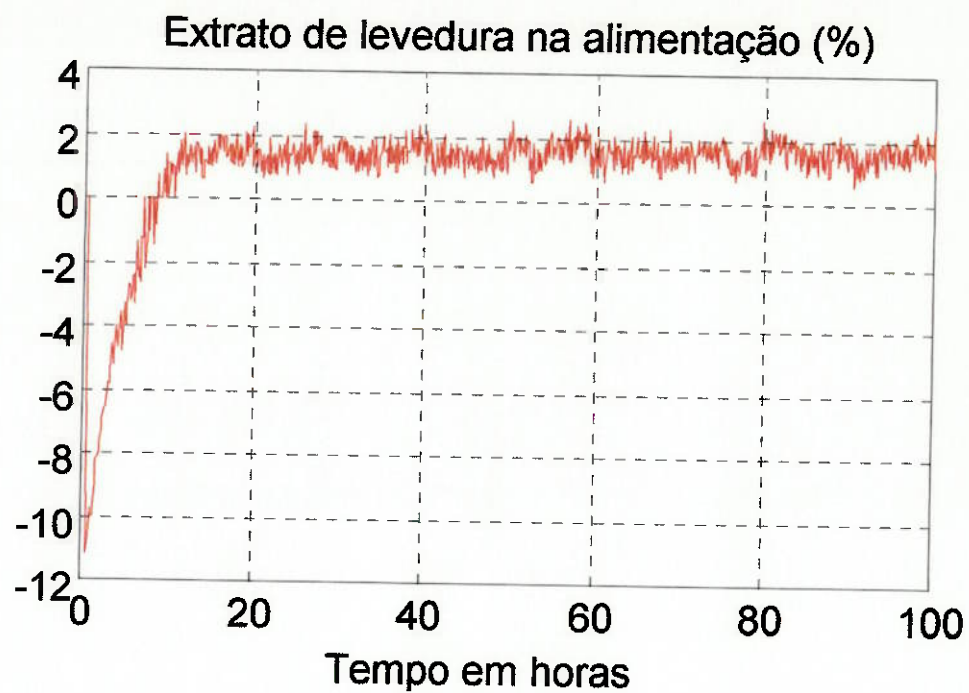


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Variáveis de Controle:

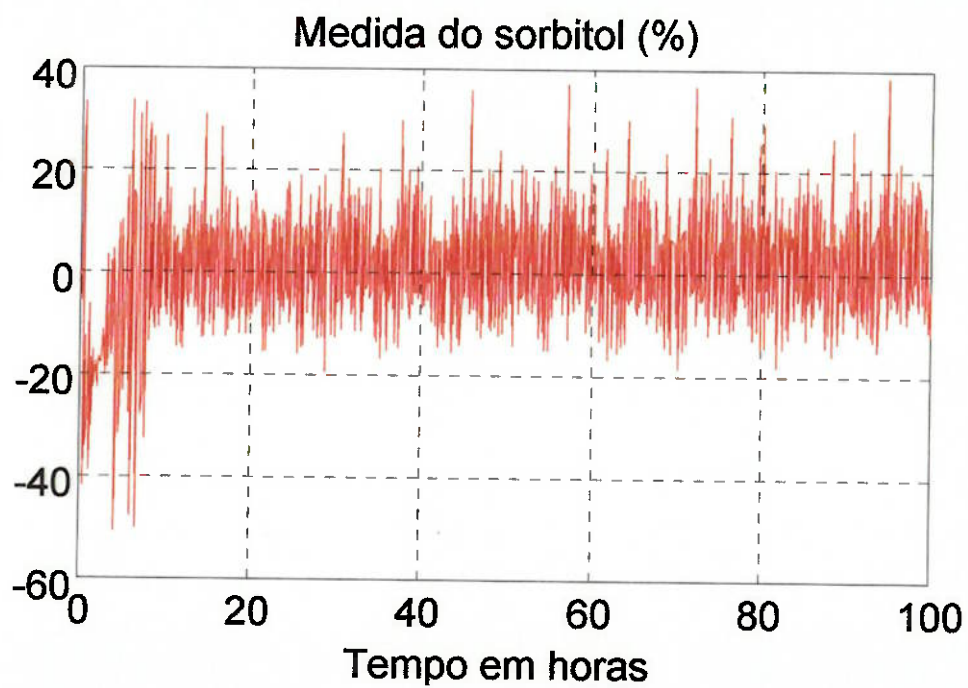
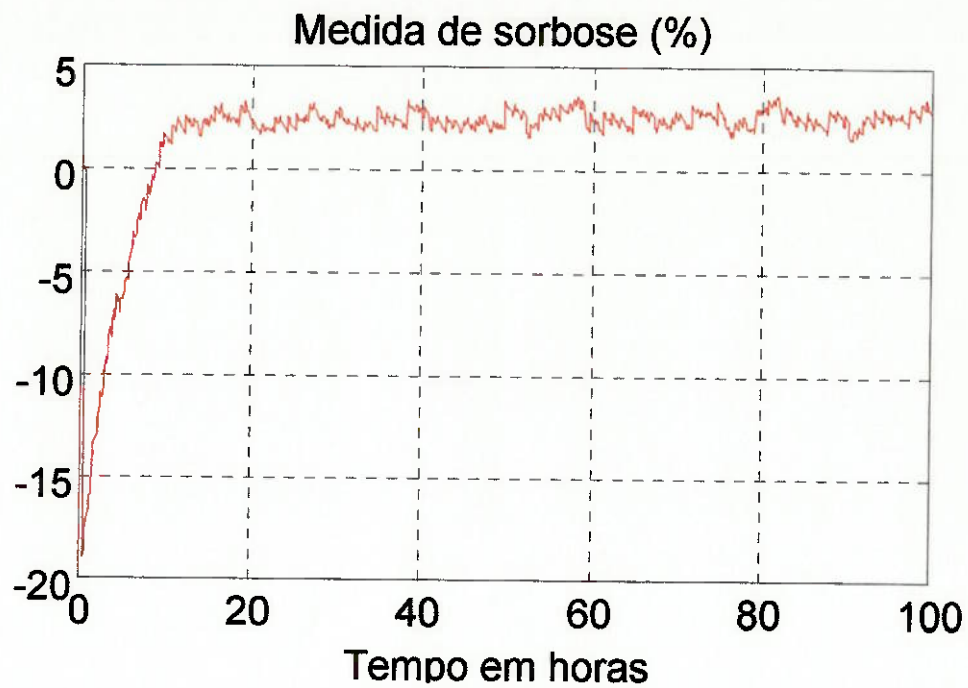


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

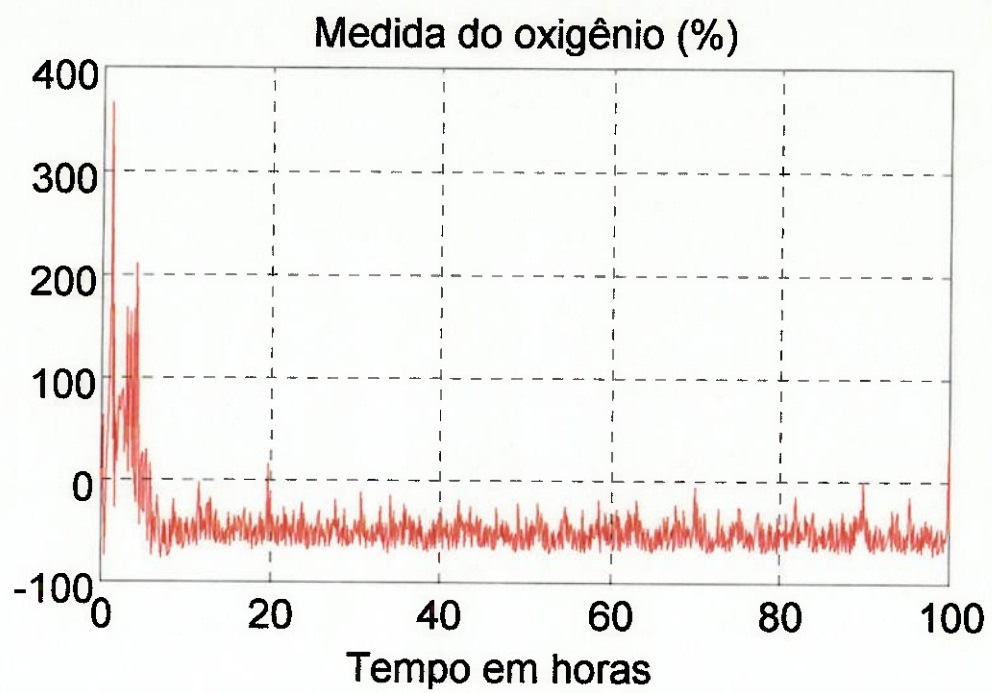


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Variáveis Medidas:



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xc - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00 -.123794160E-02 .707919687E-01 .176010163E-01 .812277235E-01
.393306572E+01 -.115528924E+00 .359136491E+01 .505149169E+00 .558033286E+01
-.431723616E+01 .411919255E-01 -.401255173E+01 -.585665554E+00 -.598804914E+01
-.847457627E-01 .131137881E-02 -.749914922E-01 -.986451444E-01 -.860463173E-01
-.955666447E+03 .125364720E+02 -.855535230E+03 -.178243181E+03 -.143892113E+04

Matriz Fi

.999981172E+00 -.784903394E-06 .421348740E-04 .111597328E-04 .393510318E-04
.198542078E-02 .999920663E+00 .183598939E-02 .180172673E-03 .270530063E-02
-.221106331E-02 .161531388E-04 .997897628E+00 -.229664841E-03 -.290284112E-02
-.506743625E-04 .831465460E-06 -.446344004E-04 .999921514E+00 -.416854150E-04
-.463296555E+00 .608081778E-02 -.414739698E+00 -.864568839E-01 .302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01 .000000000E+00 .000000000E+00 .000000000E+00
-.116870000E+03 .000000000E+00 .000000000E+00 .000000000E+00
.128284000E+03 .800000000E-01 .000000000E+00 .000000000E+00
.251000000E+01 .000000000E+00 .800000000E-01 .000000000E+00
.693600000E+01 .000000000E+00 .000000000E+00 .693600000E+01

Matriz Gama

-.197813202E-02 .155751693E-08 .403827121E-09 .136024765E-06
-.972653142E-01 .716993845E-07 .819115086E-08 .934885760E-05
.106764081E+00 .665853199E-04 -.100006689E-07 -.100316676E-04
.208880533E-02 -.164991200E-08 .666640167E-04 -.144094031E-06
-.230530652E-01 -.165312790E-04 -.344534729E-05 .336908405E-02

Matriz C - Matriz de Medições de estado do modelo linearizado

.000000000E+00 .684521263E-01 .000000000E+00 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .466200466E+01 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .000000000E+00 .000000000E+00 .709219858E+00

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Matriz L - Ganho do Filtro de Kalman

.354029393E+00 -.149809542E+02 .114531090E+00
.673189339E+00 -.266734164E+02 .786826933E+01
-.391645269E+00 .166718501E+02 -.844314928E+01
.336460176E-01 -.161046309E+01 -.121325307E+00
.000000000E+00 .000000000E+00 .281302727E+01

Matriz G1 - Ganho do controlador para o subsistema lento

.158744793E-01 .993981208E-02 -.330529012E+00 .136667582E-02
-.965998529E+01 -.179734813E+02 -.273850235E+04 -.163525113E+01
-.134234327E+01 .169365076E-01 -.476839228E-03 -.290876994E+00
.435123781E+02 -.578435716E+01 .207306491E+03 .904427383E+01

Matriz G2 - Ganho do controlador para o subsistema rápido

-.310214934E-04
.000000000E+00
.000000000E+00
-.103944787E+03

Valor de magO - Velocidade do observador

.100000000E+01

Valor de magR - Magnitude do controlador

.500000000E-01

Vetor PesoR - Controle

.400000000E+01
.800000000E+00
.200000000E+01
.200000000E+02

Vetor PesoC - Medidas

.800000000E+01
.800000000E+01
.400000000E+00

Valor de dtO (em minutos)

.500000000E-01

Matriz Fimod

.999897508E+00 -.196107628E-04 .563171873E-01 .763287219E-06 -.556097617E-13
.308971035E-02 .999868255E+00 .103153106E+00 .395485151E-03 .246150429E-11
-.347767784E-02 .548124670E-04 .934055264E+00 -.471782277E-03 -.271592254E-11
-.815103575E-04 -.654020565E-06 .598540775E-02 .999916321E+00 -.506062515E-13
-.462718077E+00 .607324063E-02 -.414228864E+00 -.863489575E-01 .300964342E+00

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Matriz Gamamod

```
.106471143E-02 .189859388E-05 .178185405E-09 -.635364098E-16 .000000000E+00
.195577371E-04 -.563534539E-01 .000000000E+00 -.744705022E-04
-.918131090E-01 .347653285E-05 .134627460E-07 .873997689E-14 .000000000E+00
.372656704E-04 -.100334792E+00 .000000000E+00 .438984038E-02
.103340529E+00 .644446656E-04 -.159049271E-07 -.953387153E-14 .000000000E+00 -
.216310831E-04 .627123854E-01 .000000000E+00 -.482369587E-02
.241530071E-02 .201808024E-06 .666638936E-04 -.161177996E-15 .000000000E+00
.185077596E-05 -.605802349E-02 .000000000E+00 -.868120263E-04
-.230384075E-01 -.165194776E-04 -.344285268E-05 .336488088E-02 .000000000E+00 -
.136842425E-07 .702352608E-05 .000000000E+00 .220621712E-02
```

Dados iniciais para simulação (arquivo VIT_C.DAT)

```
-20.0 -20.0 -20.0 -20.0 -20.0 estados iniciais
0.00 0.00 0.00 0.00 0.00 observador inicial
0.00 tempo inicial (hora) hoje 27/11/95 14h50m
100.00 tempo final (hora)
3.00 duração do filtro do atuador (min)
0.00 atraso para início do controle (horas)
400.00 atraso para melhoria da medição (horas)
1.e-6 precisão
1.0e-5 passo inicial (chute)
1.e-8 menor passo admissível (hora)
-1.0 tempo entre resultados armazenados (segundos)
400000 número máximo de passos
0 =1 simula-se até o máximo de passos; =0 não há limite
2 =1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)
2 =1 observador é o estado real; =2 obs discreto (despro)
0 numero inteiro de dto's para atraso dos contr na memorex (shift)
0.0 erro máximo aleatório para observadores iniciais (%)
360000.0 tempo de aleatorização dos parâmetros (min);
50.0 50.0 50.0 50.0 valores máximos de controle (%)
-80.0 -80.0 -80.0 -50.0 valores mínimos de controle (%)
0.0 30.0 30.0 0.0 -0.25 atrasos das medidas (min)
1.0 1.0 1.0 1.0 1.0 fatores de ajuste da propagação
0.5 0.5 0.5 0.0 0.5 deadband (%) p/ atuadores(4) e
0.0 0.0 0.0 0.0 0.0 _μ0, _Ks1, _Ks2, _W, __ (%)
0.0 0.0 0.0 0.0 0.0 _β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0 0.0 0.0 4.0 4.0 _Yxs2, _Kp, _YxO, ruído, ruído
4.0 ruído1 (%)
```

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

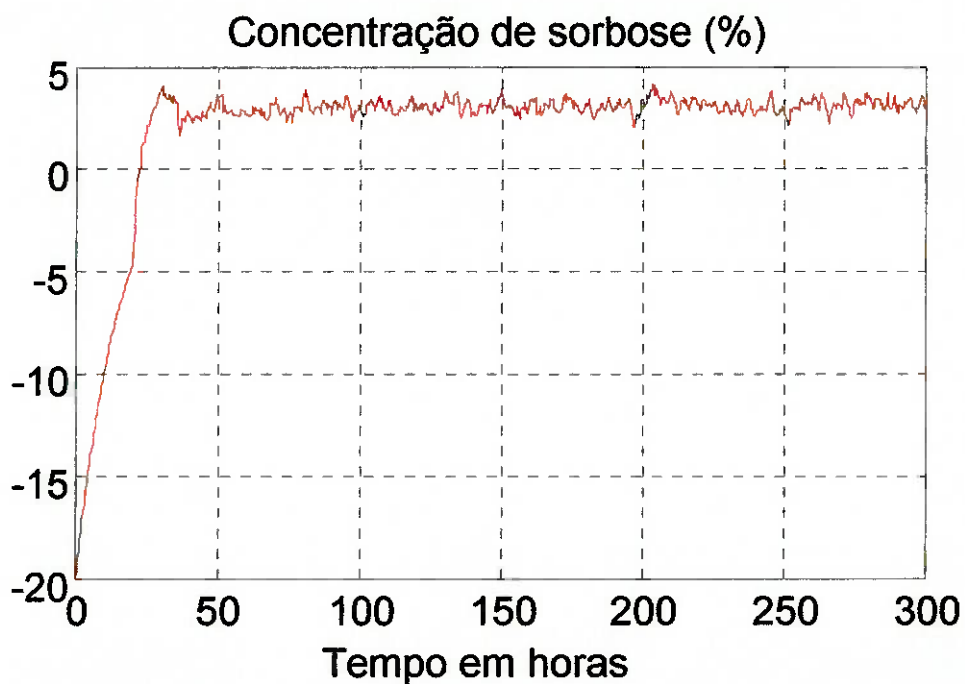
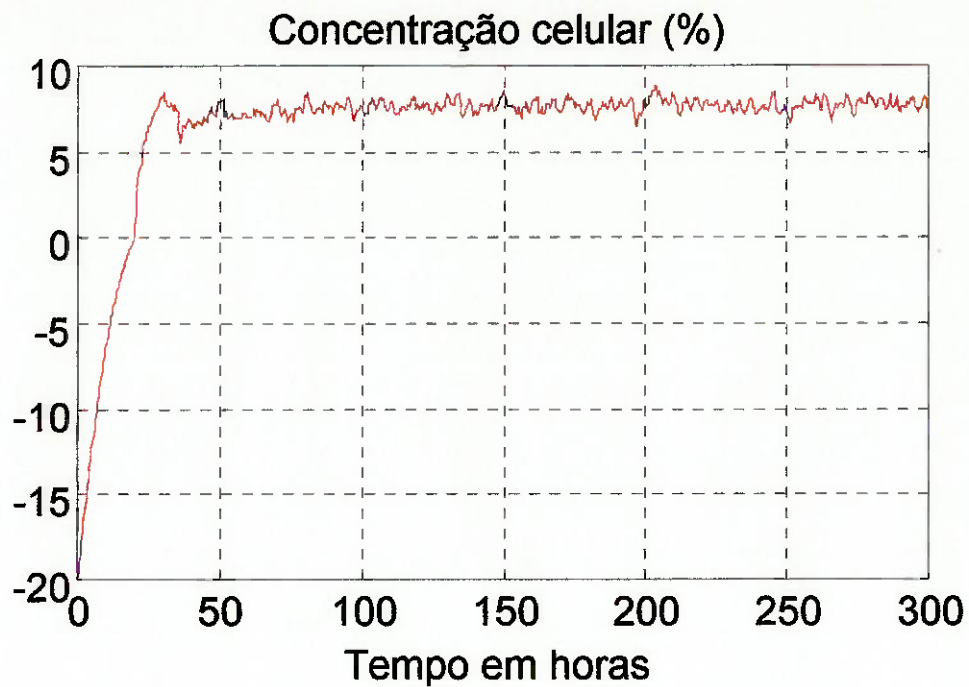
Simulação realizada em 0h11m21.74s.

Final normal da simulação

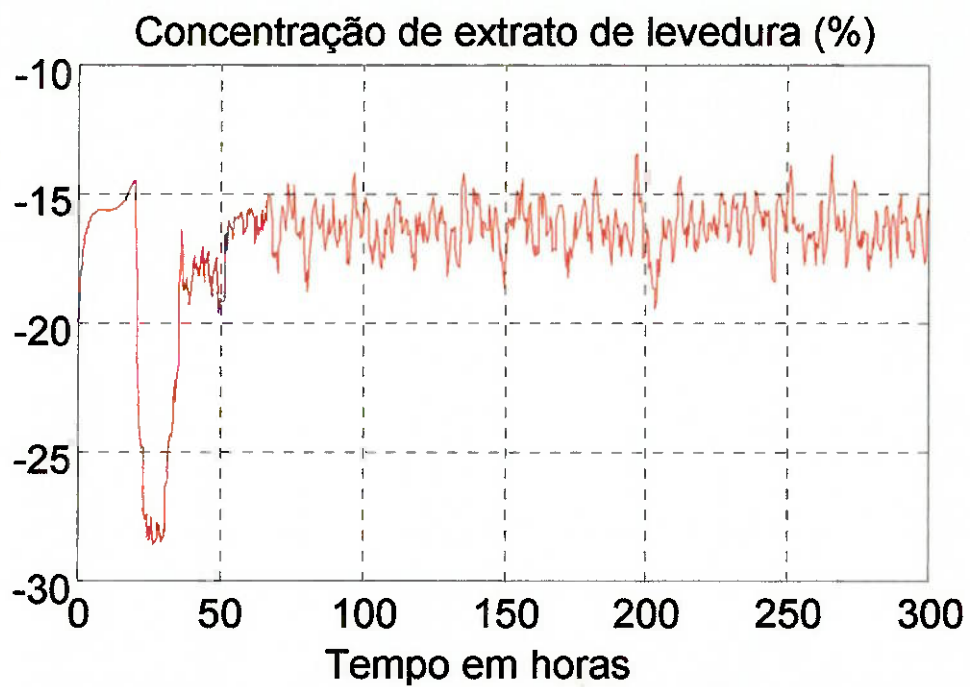
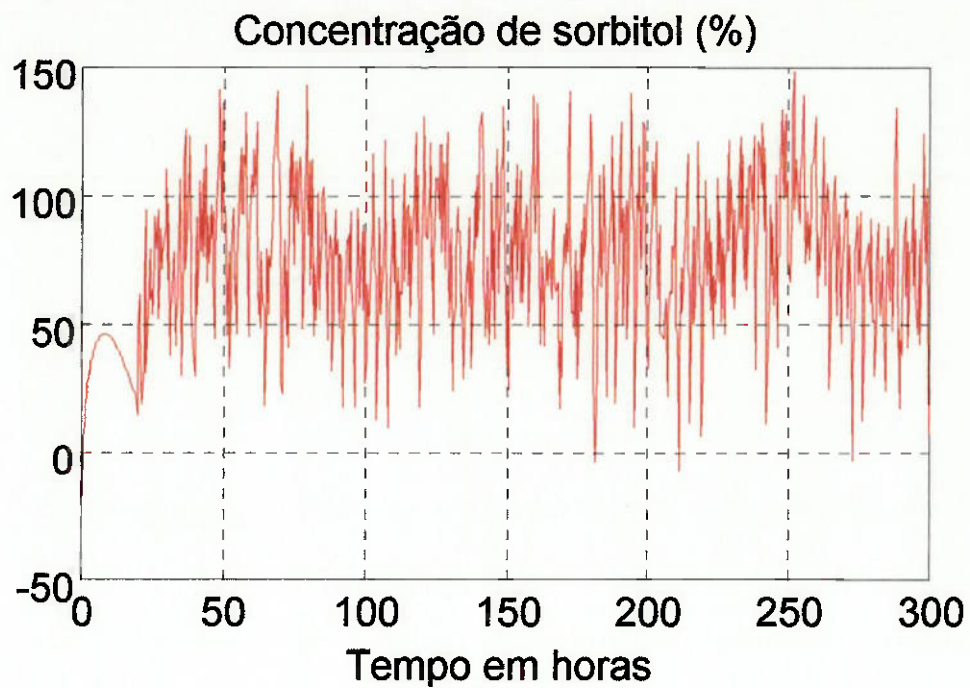
Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas.

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio , 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

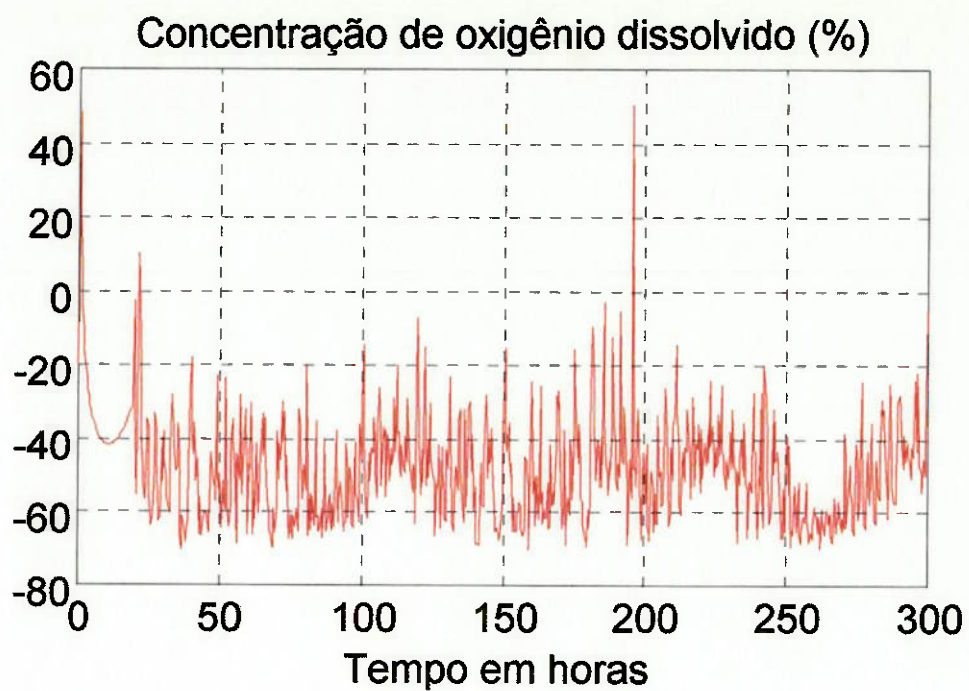
Variáveis de Estado:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

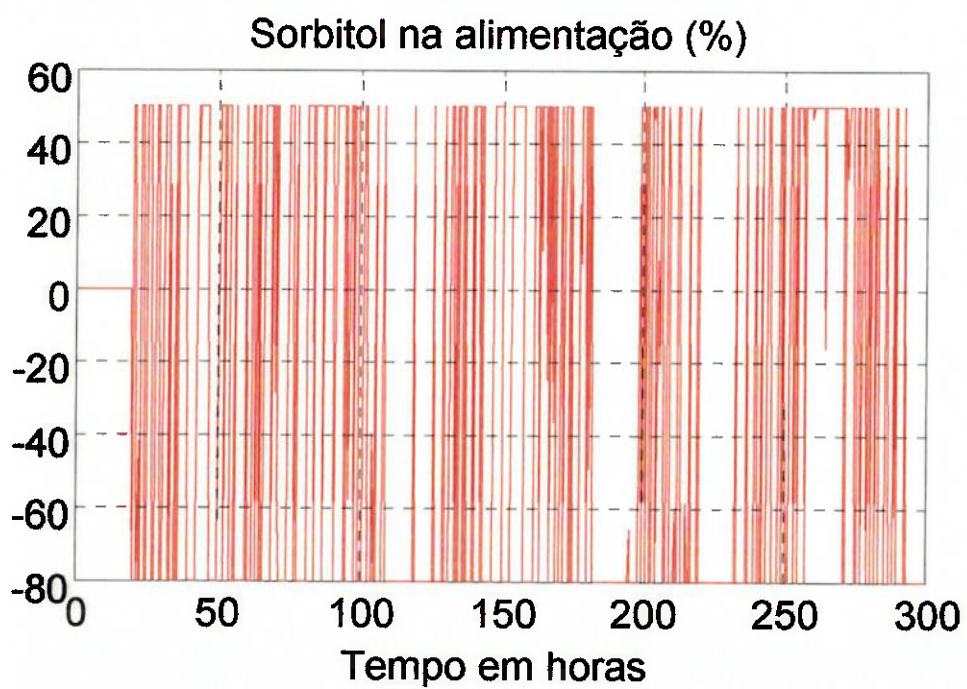
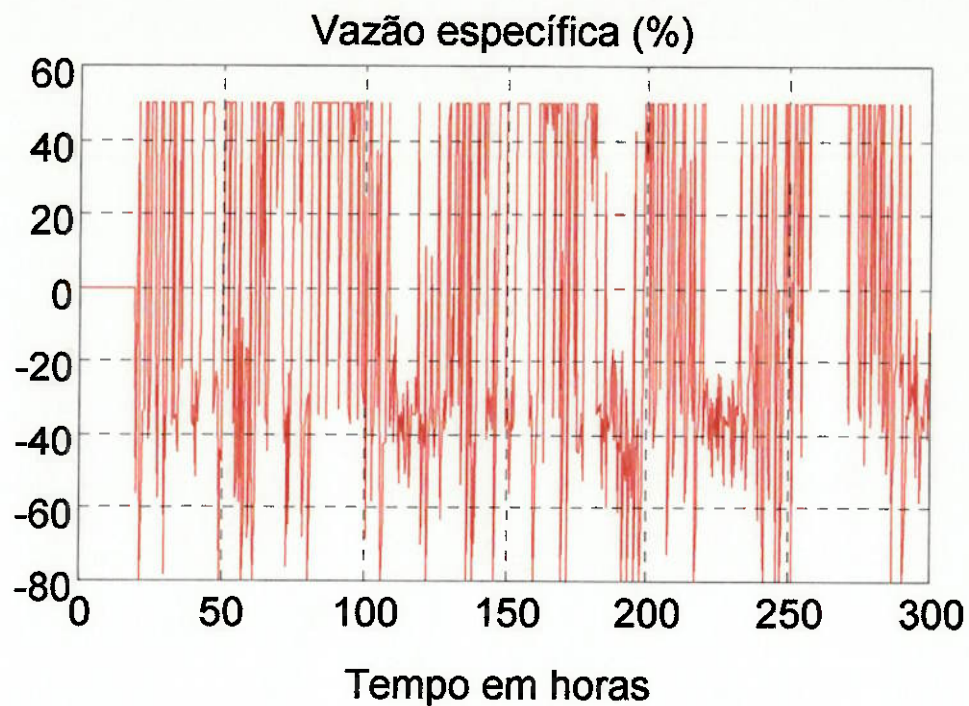


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

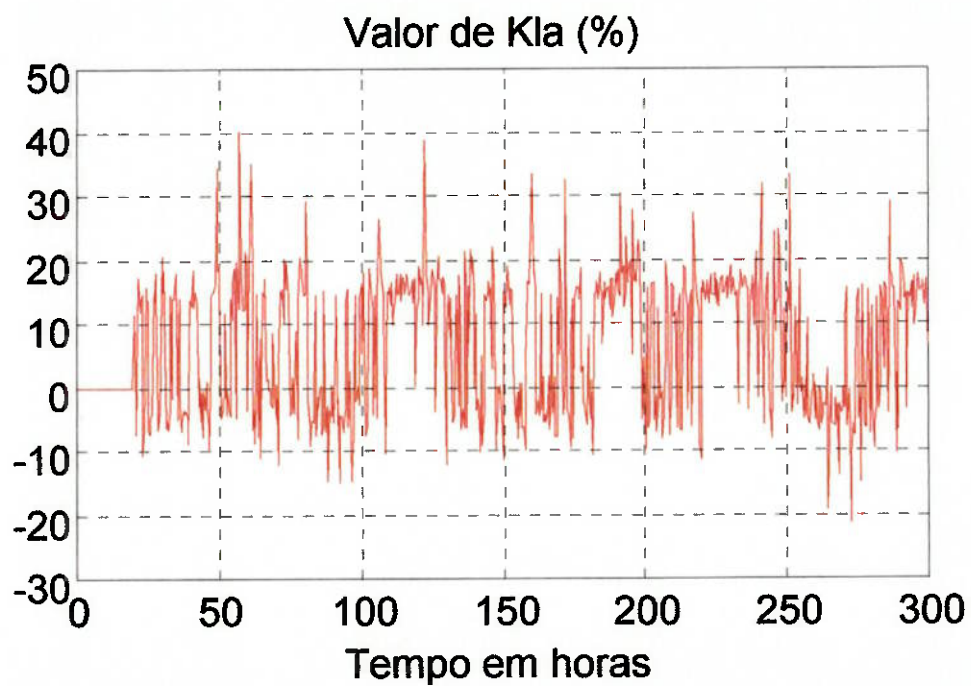
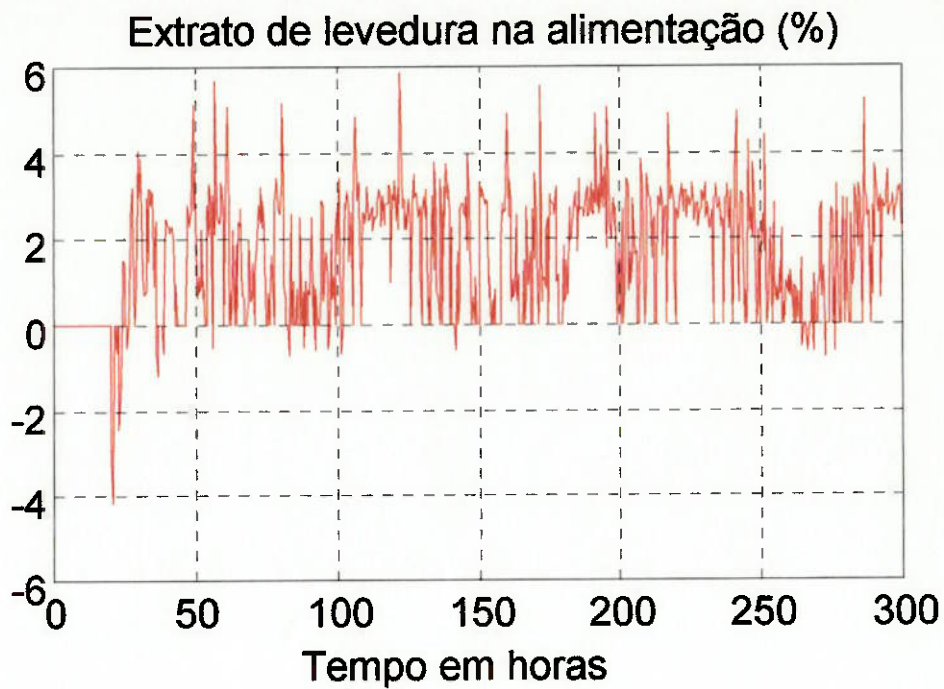


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Variáveis de Controle:

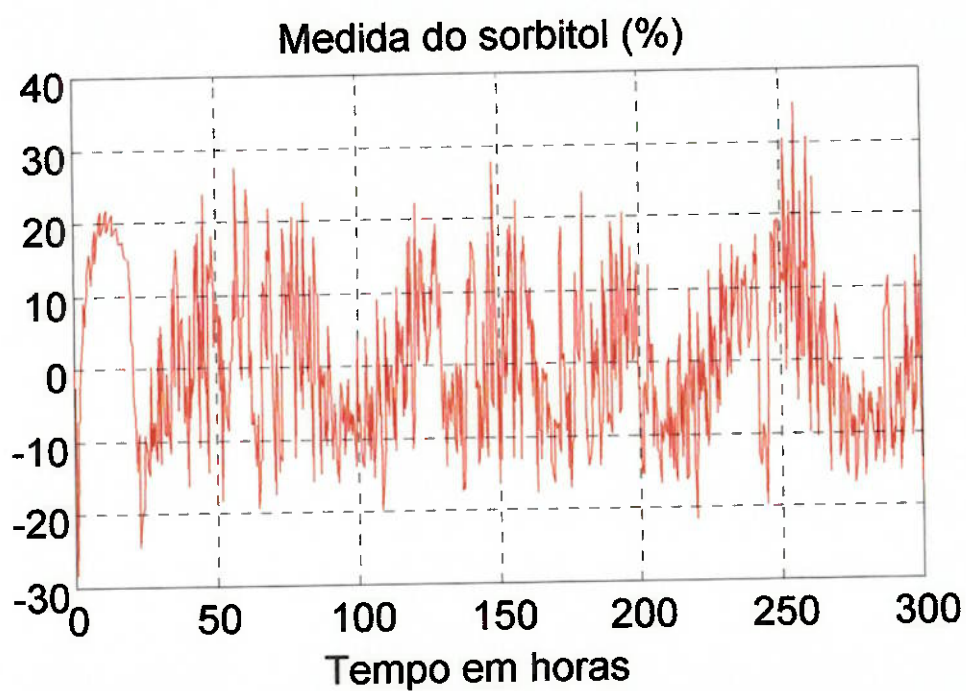
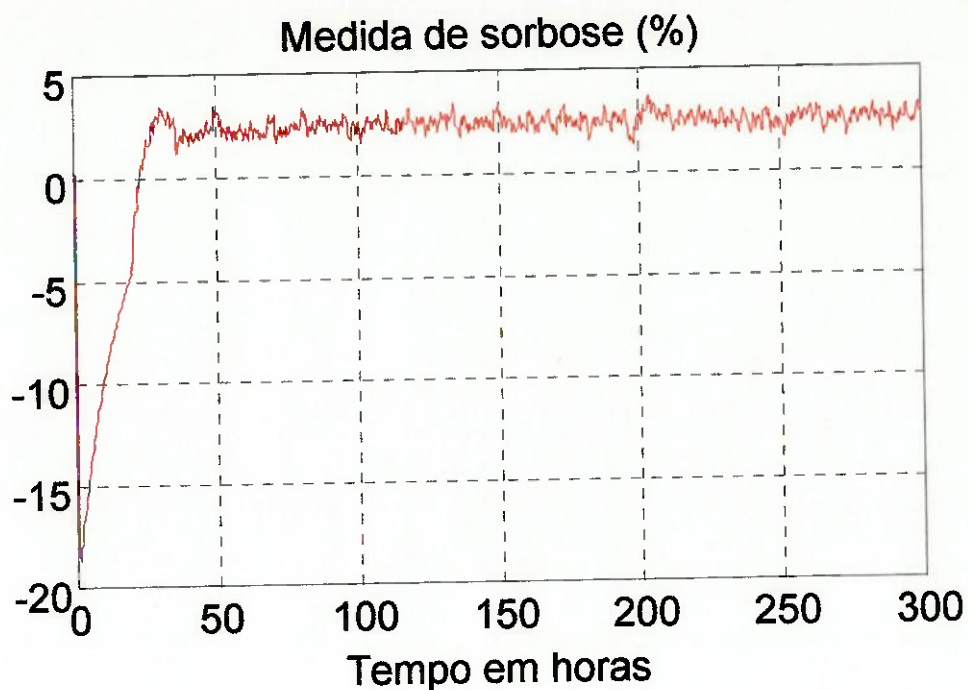


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

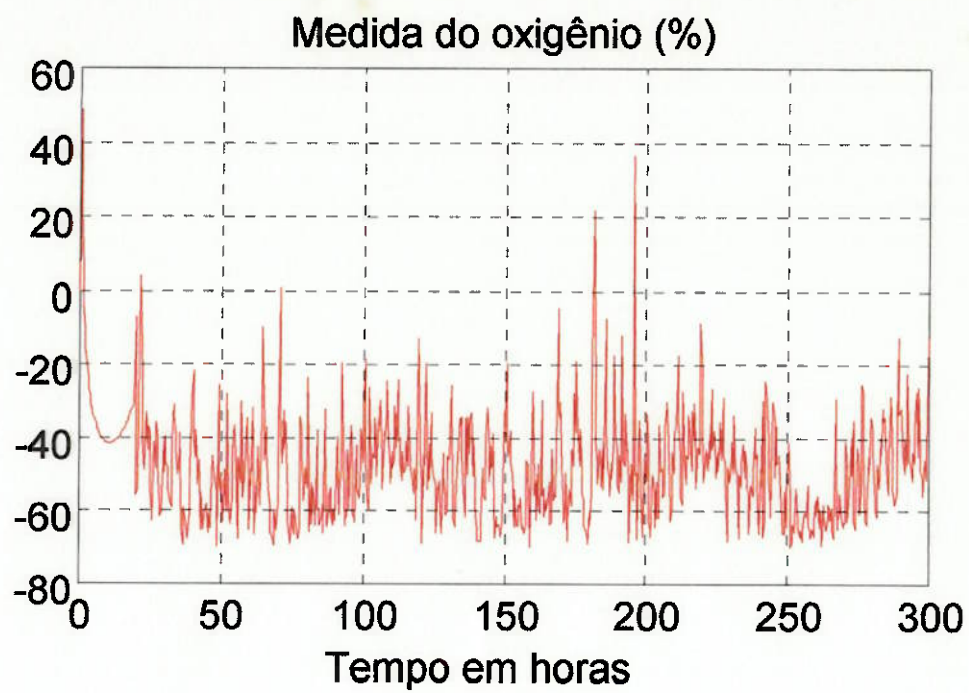


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Variáveis Medidas:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Protocolo de Simulação

Vetor U_c - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz X_e - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00	-.123794160E-02	.707919687E-01	.176010163E-01	.812277235E-01
.393306572E+01	-.115528924E+00	.359136491E+01	.505149169E+00	.558033286E+01
-.431723616E+01	.411919255E-01	-.401255173E+01	-.585665554E+00	-.598804914E+01
-.847457627E-01	.131137881E-02	-.749914922E-01	-.986451444E-01	-.860463173E-01
-.955666447E+03	.125364720E+02	-.855535230E+03	-.178243181E+03	-.143892113E+04

Matriz F_i

.999981172E+00	-.784903394E-06	.421348740E-04	.111597328E-04	.393510318E-04
.198542078E-02	.999920663E+00	.183598939E-02	.180172673E-03	.270530063E-02
-.221106331E-02	.161531388E-04	.997897628E+00	-.229664841E-03	-.290284112E-02
-.506743625E-04	.831465460E-06	-.446344004E-04	.999921514E+00	-.416854150E-04
-.463296555E+00	.608081778E-02	-.414739698E+00	-.864568839E-01	.302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01	.000000000E+00	.000000000E+00	.000000000E+00
-.116870000E+03	.000000000E+00	.000000000E+00	.000000000E+00
.128284000E+03	.800000000E-01	.000000000E+00	.000000000E+00
.251000000E+01	.000000000E+00	.800000000E-01	.000000000E+00
.693600000E+01	.000000000E+00	.000000000E+00	.693600000E+01

Matriz G_{ama}

-.197813202E-02	.155751693E-08	.403827121E-09	.136024765E-06
-.972653142E-01	.716993845E-07	.819115086E-08	.934885760E-05
.106764081E+00	.665853199E-04	-.100006689E-07	-.100316676E-04
.208880533E-02	-.164991200E-08	.666640167E-04	-.144094031E-06
-.230530652E-01	-.165312790E-04	-.344534729E-05	.336908405E-02

Matriz C - Matriz de Medições de estado do modelo linearizado

.000000000E+00	.684521263E-01	.000000000E+00	.000000000E+00	.000000000E+00
.000000000E+00	.000000000E+00	.466200466E+01	.000000000E+00	.000000000E+00
.000000000E+00	.000000000E+00	.000000000E+00	.000000000E+00	.709219858E+00

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Matriz L - Ganho do Filtro de Kalman

```
.926552996E+00 -.666472053E+02 .458124361E+00
.203649087E+01 -.971229455E+02 .314730773E+02
-.101861027E+01 .749402080E+02 -.337725972E+02
.685233677E-01 -.923410991E+01 -.485301229E+00
.000000000E+00 .000000000E+00 .900168745E+01
```

Matriz G1 - Ganho do controlador para o subsistema lento

```
.158744793E-01 .993981208E-02 -.330529012E+00 .136667582E-02
-.965998529E+01 -.179734813E+02 -.273850235E+04 -.163525113E+01
-.134234327E+01 .169365076E-01 -.476839228E-03 -.290876994E+00
.435123781E+02 -.578435716E+01 .207306491E+03 .904427383E+01
```

Matriz G2 - Ganho do controlador para o subsistema rápido -.310214934E-04

```
.000000000E+00
.000000000E+00
-.103944787E+03
```

Valor de magO - Velocidade do observador

```
.300000000E+01
```

Valor de magR - Magnitude do controlador

```
.500000000E-01
```

Vetor PesoR - Controle

```
.400000000E+01
.800000000E+00
.200000000E+01
.200000000E+02
```

Vetor PesoC - Medidas

```
.800000000E+01
.800000000E+01
.400000000E+00
```

Valor de dtO (em minutos)

```
.500000000E-01
```

Matriz Fimod

```
.999271159E+00 -.389895354E-04 .224016496E+00 -.997541989E-04 .109513424E-02
.597830442E-02 .999759875E+00 .332712364E+00 .956552239E-03 -.631246656E-02
-.684040707E-02 .129009595E-03 .741202954E+00 -.111761291E-02 .730623692E-02
-.238563430E-03 .126601528E-06 .309078115E-01 .999889445E+00 .292501428E-03
-.461234037E+00 .605376498E-02 -.412901807E+00 -.860720824E-01 .297373565E+00
```

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Matriz Gamamod

```
.105940093E-01 .783890558E-05 -.176893455E-08 .221898561E-05 .000000000E+00
.460072062E-04 -.224624461E+00 .000000000E+00 -.207465167E-02
-.788048438E-01 .116017557E-04 .277975383E-07 -.241484456E-04 .000000000E+00
.106186686E-03 -.327323047E+00 .000000000E+00 .151037137E-01
.924788084E-01 .576549683E-04 -.321427269E-07 .270661007E-04 .000000000E+00 -
.504056958E-04 .252566760E+00 .000000000E+00 -.172366863E-01
.382809282E-02 .108229570E-05 .666633101E-04 .794997274E-06 .000000000E+00
.296048483E-05 -.311226530E-01 .000000000E+00 -.610149391E-03
-.230014935E-01 -.164896356E-04 -.343662728E-05 .335409819E-02 .000000000E+00 -
.116334094E-07 .848762000E-05 .000000000E+00 .803170234E-02
```

Dados iniciais para simulação (arquivo VIT_C.DAT)

```
-20.0 -20.0 -20.0 -20.0 -20.0 estados iniciais
0.00 0.00 0.00 0.00 0.00 observador inicial
0.00 tempo inicial (hora) hoje 27/11/95 14h50m
300.00 tempo final (hora)
3.00 duração do filtro do atuador (min)
20.00 atraso para início do controle (horas)
400.00 atraso para melhoria da medição (horas)
1.e-6 precisão
1.0e-5 passo inicial (chute)
1.e-8 menor passo admissível (hora)
-1.0 tempo entre resultados armazenados (segundos)
400000 número máximo de passos
0 =1 simula-se até o máximo de passos; =0 não há limite
2 =1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)
2 =1 observador é o estado real; =2 obs discreto (despro)
0 número inteiro de dtos para atraso dos contr na memorex (shift)
0.0 erro máximo aleatório para observadores iniciais (%)
360000.0 tempo de aleatorização dos parâmetros (min);
50.0 50.0 50.0 50.0 valores máximos de controle (%)
-80.0 -80.0 -80.0 -50.0 valores mínimos de controle (%)
0.0 30.0 30.0 0.0 -0.25 atrasos das medidas (min)
1.0 1.0 1.0 1.0 1.0 fatores de ajuste da propagação
0.5 0.5 0.5 0.0 0.5 deadband (%) p/ atuadores(4) e
0.0 0.0 0.0 0.0 0.0 _μ0, _Ks1, _Ks2, _W, __ (%)
0.0 0.0 0.0 0.0 0.0 _β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0 0.0 0.0 12.0 7.0 _Yxs2, _Kp, _YxO, ruído, ruído
5.0 ruído1 (%)
```

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, ser o erro inicial para observadores de estado; se positivo, ser erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

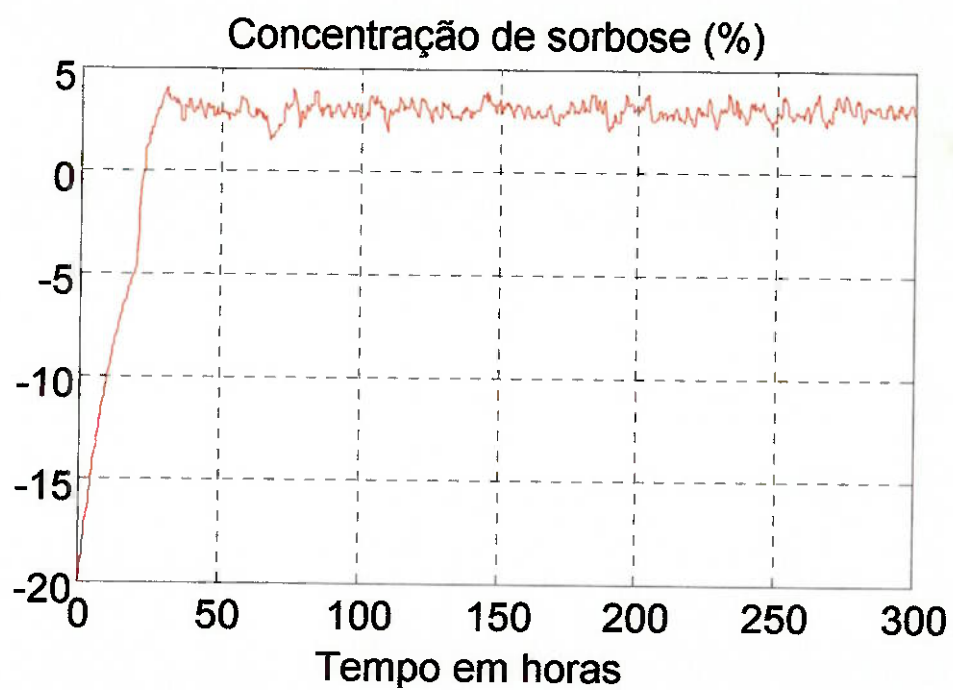
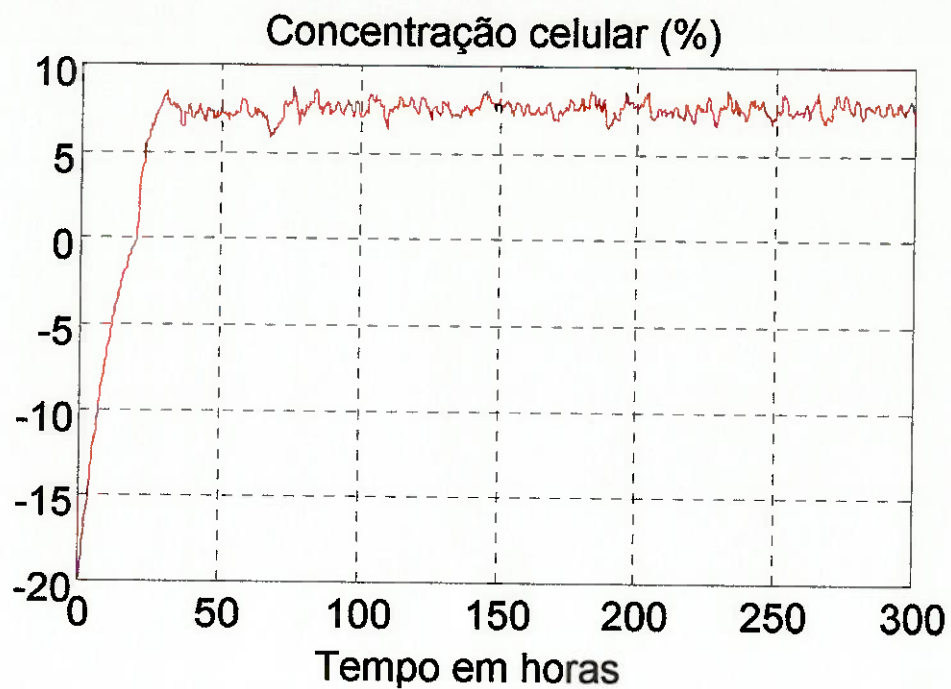
Simulação realizada em 0h33m21.92s.

Final normal da simulação

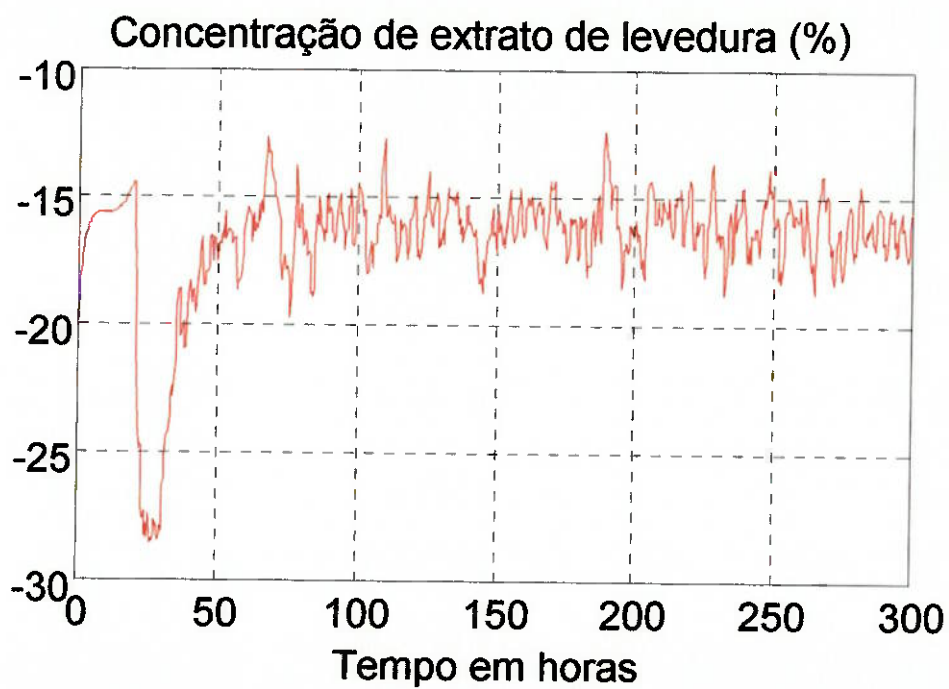
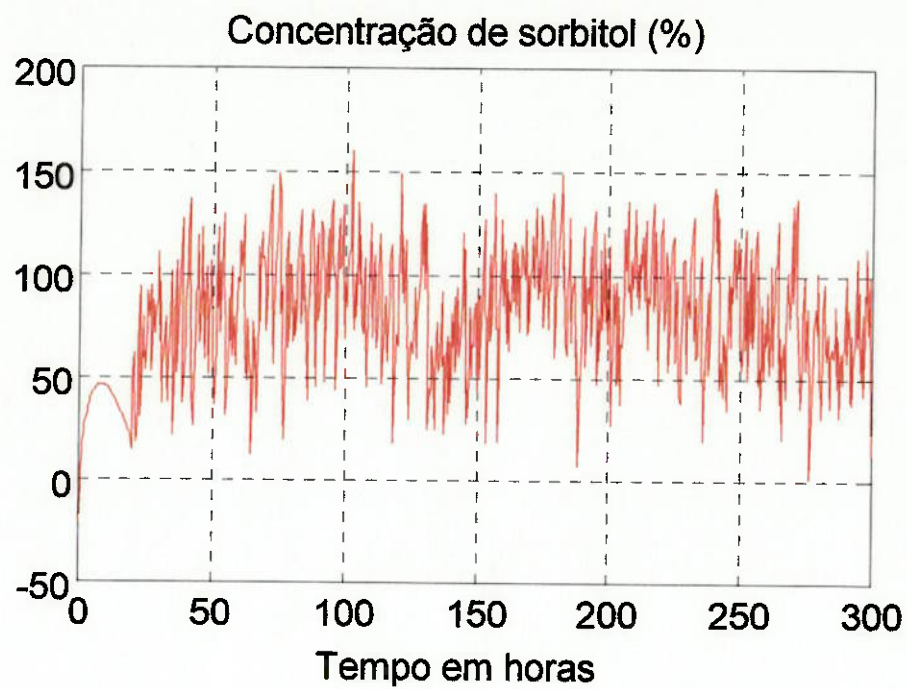
Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

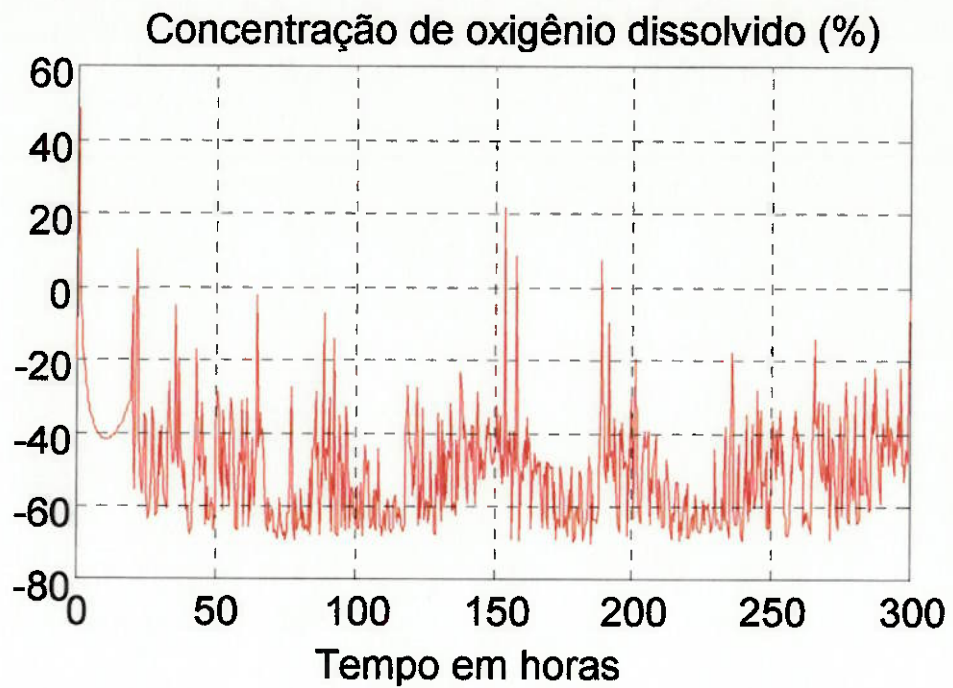
Variáveis de Estado:



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

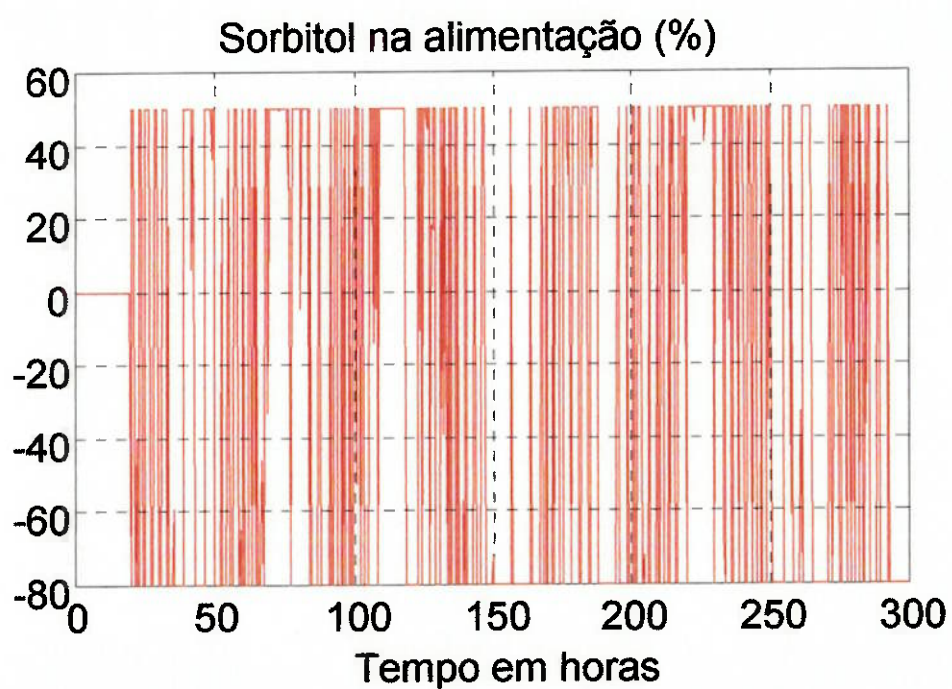
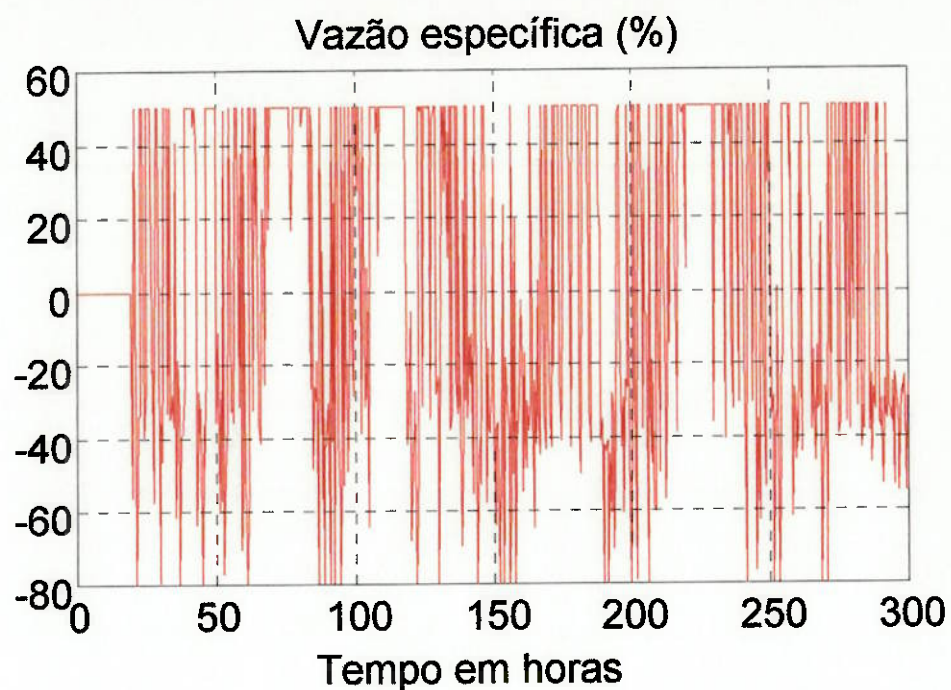


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

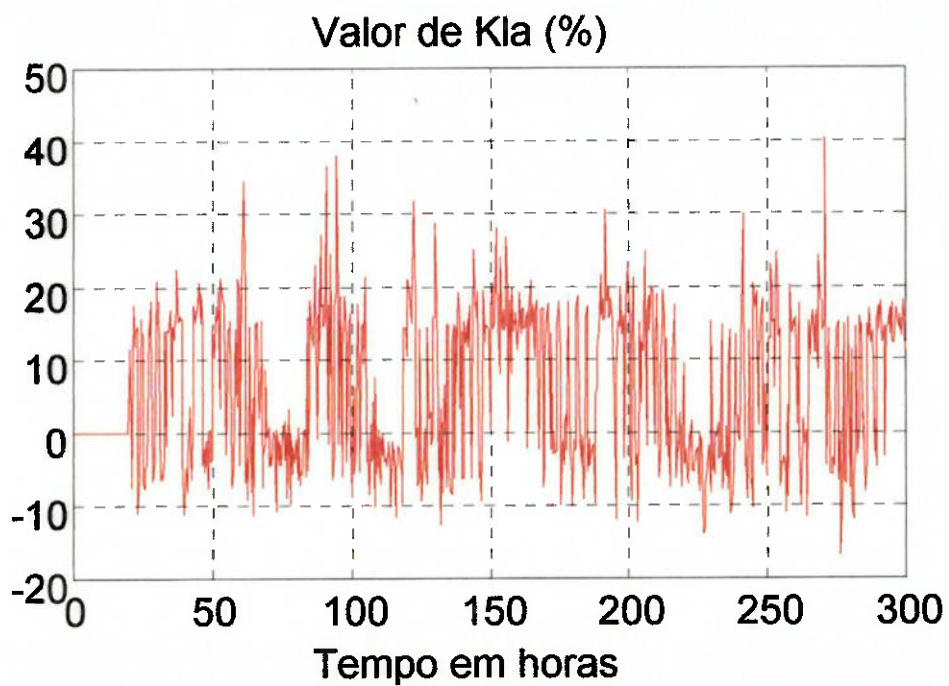
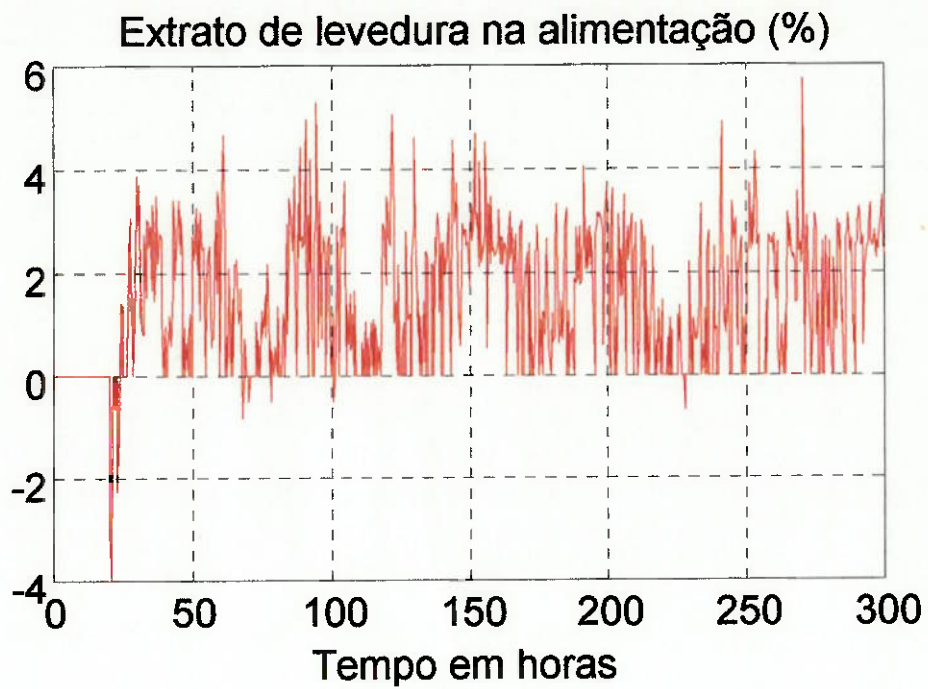


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Variáveis de Controle:

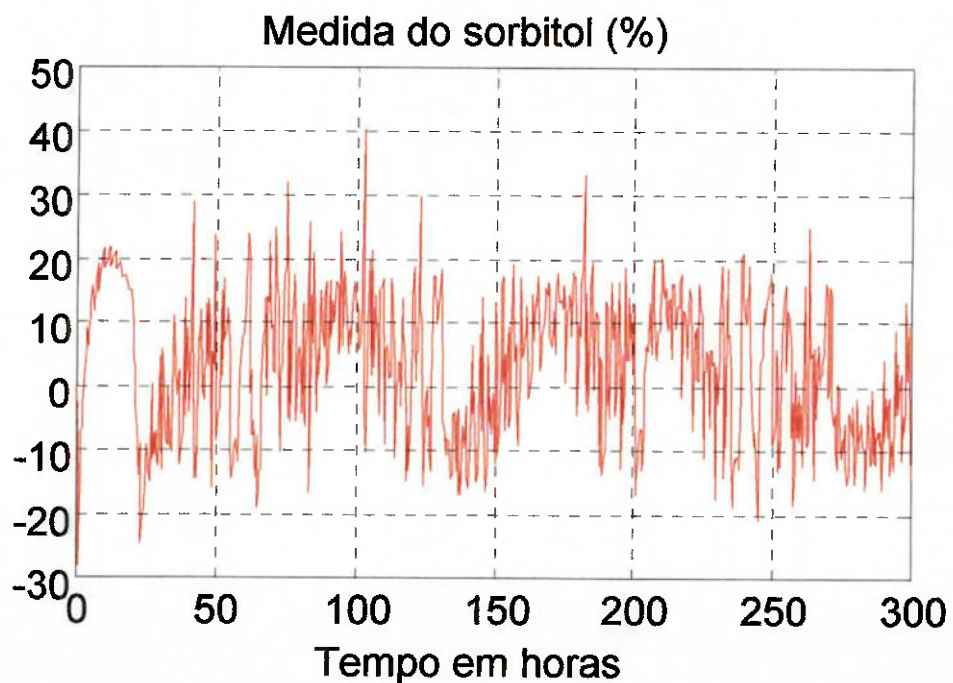
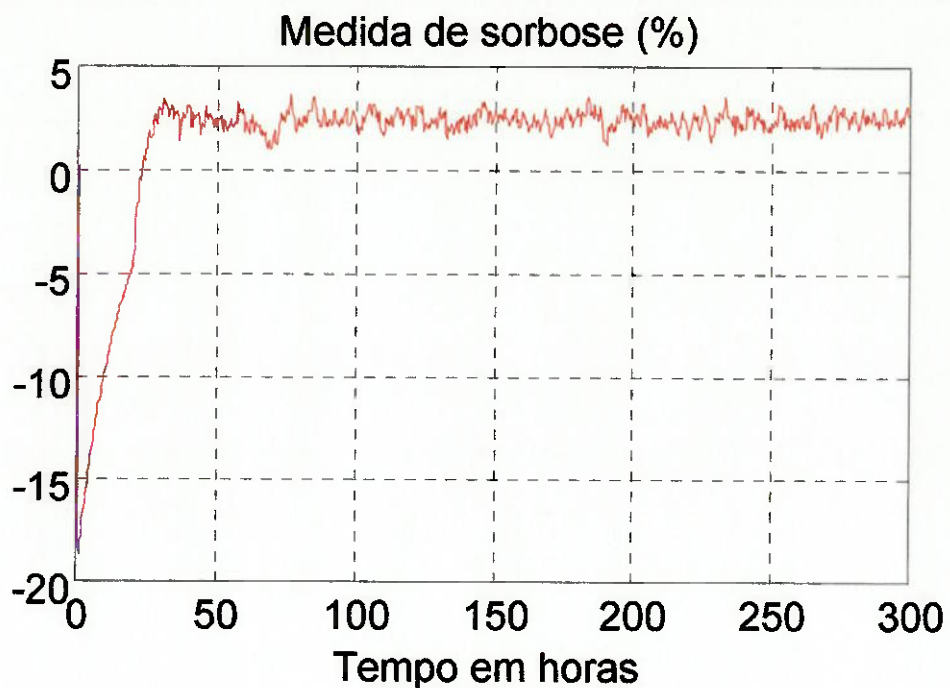


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

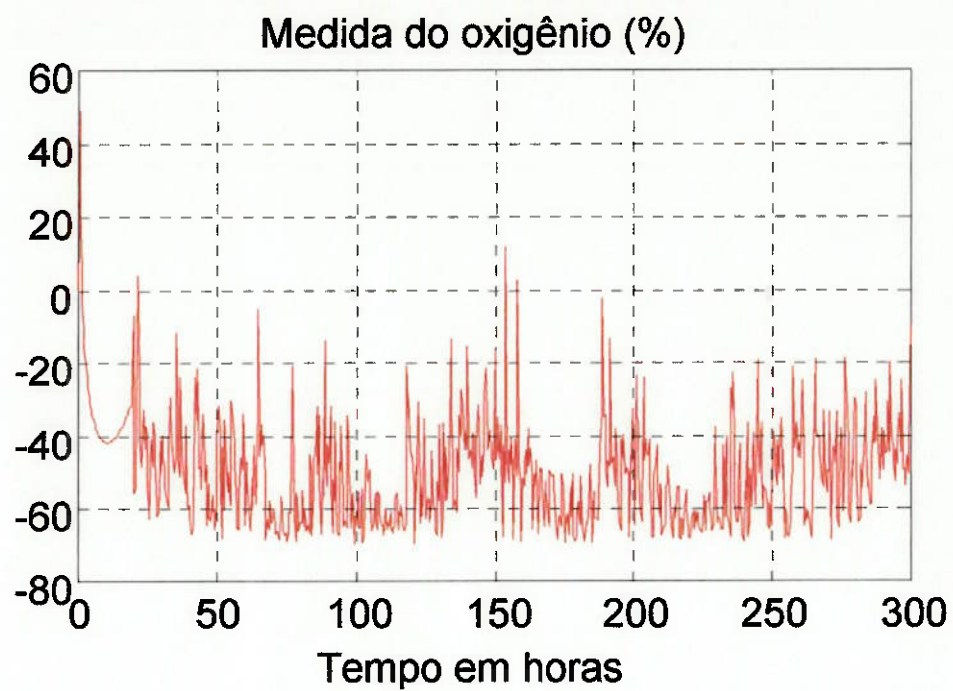


Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Variáveis Medidas:



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.



Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xe - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00 -.123794160E-02 .707919687E-01 .176010163E-01 .812277235E-01
.393306572E+01 -.115528924E+00 .359136491E+01 .505149169E+00 .558033286E+01
-.431723616E+01 .411919255E-01 -.401255173E+01 -.585665554E+00 -.598804914E+01
-.847457627E-01 .131137881E-02 -.749914922E-01 -.986451444E-01 -.860463173E-01
-.955666447E+03 .125364720E+02 -.855535230E+03 -.178243181E+03 -.143892113E+04

Matriz Fi

.999981172E+00 -.784903394E-06 .421348740E-04 .111597328E-04 .393510318E-04
.198542078E-02 .999920663E+00 .183598939E-02 .180172673E-03 .270530063E-02
-.221106331E-02 .161531388E-04 .997897628E+00 -.229664841E-03 -.290284112E-02
-.506743625E-04 .831465460E-06 -.446344004E-04 .999921514E+00 -.416854150E-04
-.463296555E+00 .608081778E-02 -.414739698E+00 -.864568839E-01 .302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01 .000000000E+00 .000000000E+00 .000000000E+00
-.116870000E+03 .000000000E+00 .000000000E+00 .000000000E+00
.128284000E+03 .800000000E-01 .000000000E+00 .000000000E+00
.251000000E+01 .000000000E+00 .800000000E-01 .000000000E+00
.693600000E+01 .000000000E+00 .000000000E+00 .693600000E+01

Matriz Gama

-.197813202E-02 .155751693E-08 .403827121E-09 .136024765E-06
-.972653142E-01 .716993845E-07 .819115086E-08 .934885760E-05
.106764081E+00 .665853199E-04 -.100006689E-07 -.100316676E-04
.208880533E-02 -.164991200E-08 .666640167E-04 -.144094031E-06
-.230530652E-01 -.165312790E-04 -.344534729E-05 .336908405E-02

Matriz C - Matriz de Medições de estado do modelo linearizado

.000000000E+00 .684521263E-01 .000000000E+00 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .466200466E+01 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .000000000E+00 .000000000E+00 .709219858E+00

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Matriz L - Ganho do Filtro de Kalman

```
.141611757E+01 -.599238166E+02 .458124360E+00
.269275736E+01 -.106693666E+03 .314730773E+02
-.156658108E+01 .666874004E+02 -.337725971E+02
.134584070E+00 -.644185235E+01 -.485301229E+00
.000000000E+00 .000000000E+00 .112521091E+02
```

Matriz G1 - Ganho do controlador para o subsistema lento

```
.158744793E-01 .993981208E-02 -.330529012E+00 .136667582E-02
-.965998529E+01 -.179734813E+02 -.273850235E+04 -.163525113E+01
-.134234327E+01 .169365076E-01 -.476839228E-03 -.290876994E+00
.435123781E+02 -.578435716E+01 .207306491E+03 .904427383E+01
```

Matriz G2 - Ganho do controlador para o subsistema rápido

```
-.310214934E-04
.000000000E+00
.000000000E+00
-.103944787E+03
```

Valor de magO - Velocidade do observador

```
.300000000E+01
```

Valor de magR - Magnitude do controlador

```
.500000000E-01
```

Vetor PesoR - Controle

```
.400000000E+01
.800000000E+00
.200000000E+01
.200000000E+02
```

Vetor PesoC - Medidas

```
.800000000E+01
.800000000E+01
.400000000E+00
```

Valor de dtO (em minutos)

```
.500000000E-01
```

Matriz Fimod

```
.999343765E+00 -.650092312E-04 .204529040E+00 -.881810227E-04 .984811072E-03
.585264996E-02 .999730900E+00 .370392156E+00 .936560103E-03 -.611284745E-02
-.692880157E-02 .158306520E-03 .765478029E+00 -.113166010E-02 .743845544E-02
-.206373582E-03 -.392202021E-05 .218599048E-01 .999894570E+00 .242923670E-03
-.460987383E+00 .605057060E-02 -.412698696E+00 -.860260629E-01 .296976896E+00
```

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Matriz Gamamod

.944317282E-02	.712059463E-05	-.153170265E-08	.197305052E-05	.000000000E+00
.712137893E-04	-.205070680E+00	.000000000E+00	-.185960208E-02	
-.767922037E-01	.128556035E-04	.274037288E-07	-.237279434E-04	.000000000E+00
.136568617E-03	-.365117645E+00	.000000000E+00	.147229311E-01	
.939047883E-01	.585449282E-04	-.324306229E-07	.273594353E-04	.000000000E+00 -
.787156082E-04	.228210909E+00	.000000000E+00	-.174963053E-01	
.331416243E-02	.761758542E-06	.666634137E-04	.686867706E-06	.000000000E+00
.664880432E-05	-.220453372E-01	.000000000E+00	-.514155588E-03	
-.229944577E-01	-.164841098E-04	-.343538349E-05	.335230582E-02	.000000000E+00 -
.539800373E-07	.261834820E-04	.000000000E+00	.879979519E-02	

Dados iniciais para simulação (arquivo VIT_C.DAT)

-20.0	-20.0	-20.0	-20.0	-20.0	estados iniciais
0.00	0.00	0.00	0.00	0.00	observador inicial
0.00	tempo inicial (hora)		hoje 27/11/95 14h50m		
300.00	tempo final (hora)				
3.00	duração do filtro do atuador (min)				
20.00	atraso para início do controle (horas)				
400.00	atraso para melhoria da medição (horas)				
1.e-6	precisão				
1.0e-5	passo inicial (chute)				
1.e-8	menor passo admissível (hora)				
-1.0	tempo entre resultados armazenados (segundos)				
400000	número máximo de passos				
0	=1 simula-se até o máximo de passos; =0 não há limite				
2	=1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)				
2	=1 observador é o estado real; =2 obs discreto (despro)				
0	numero inteiro de dto's para atraso dos contr na memorex (shift)				
0.0	erro máximo aleatório para observadores iniciais (%)				
360000.0	tempo de aleatorização dos parâmetros (min);				
50.0	50.0	50.0	50.0	valores máximos de controle (%)	
-80.0	-80.0	-80.0	-50.0	valores mínimos de controle (%)	
0.0	30.0	30.0	0.0	-0.25	atrasos das medidas (min)
1.0	1.0	1.0	1.0	1.0	fatores de ajuste da propagaçã
0.5	0.5	0.5	0.0	0.5	deadband (%) p/ atuadores(4) e
0.0	0.0	0.0	0.0	0.0	_μ0, _Ks1, _Ks2, _W, __ (%)
0.0	0.0	0.0	0.0	0.0	_β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0	0.0	0.0	4.0	4.0	_Yxs2, _Kp, _YxO, ruído, ruído
4.0	ruído1 (%)				

Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Observações:

- 1 - tempo entre resultados: se -1. o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0. usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

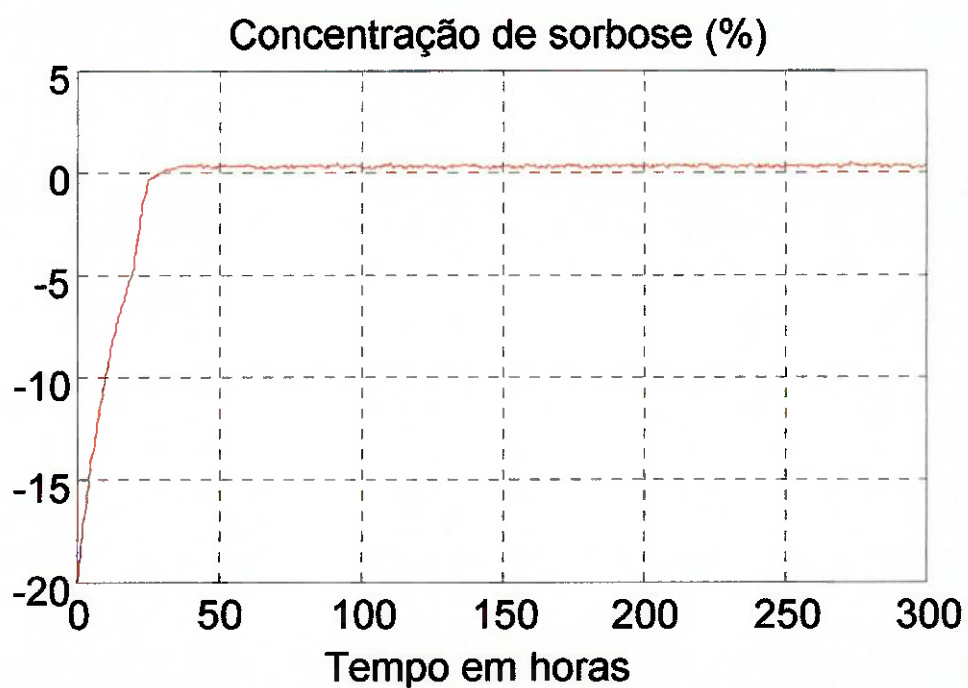
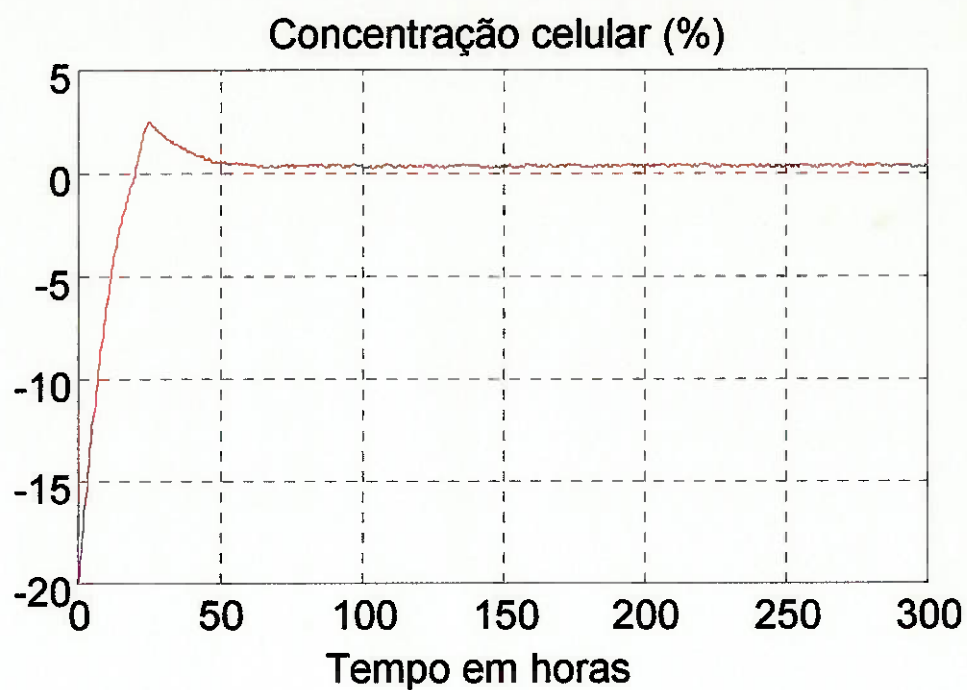
Simulação realizada em 0h32m43.34s.

Final normal da simulação

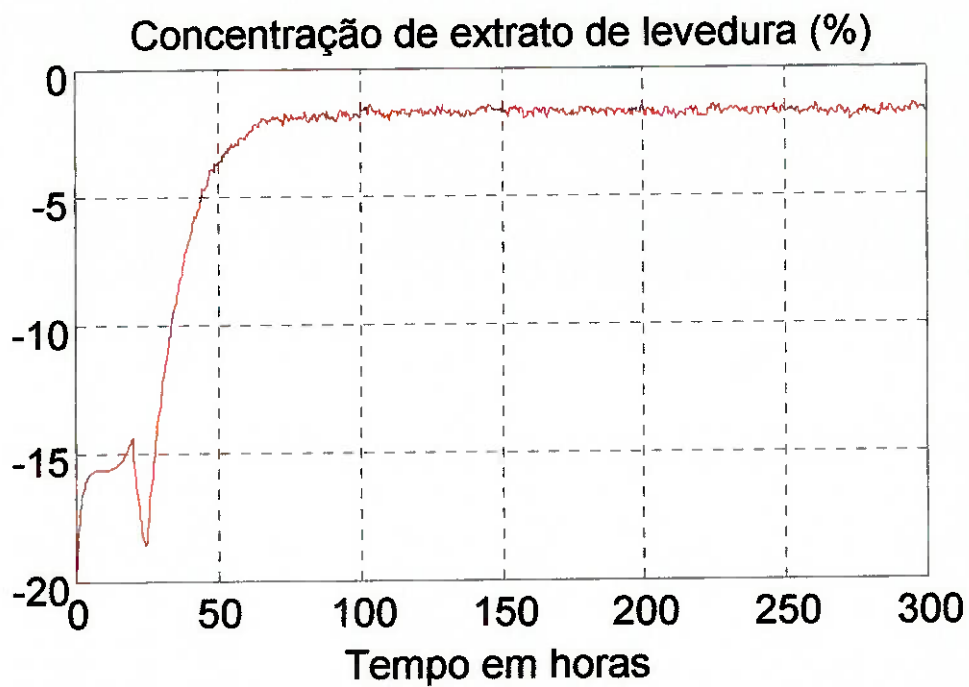
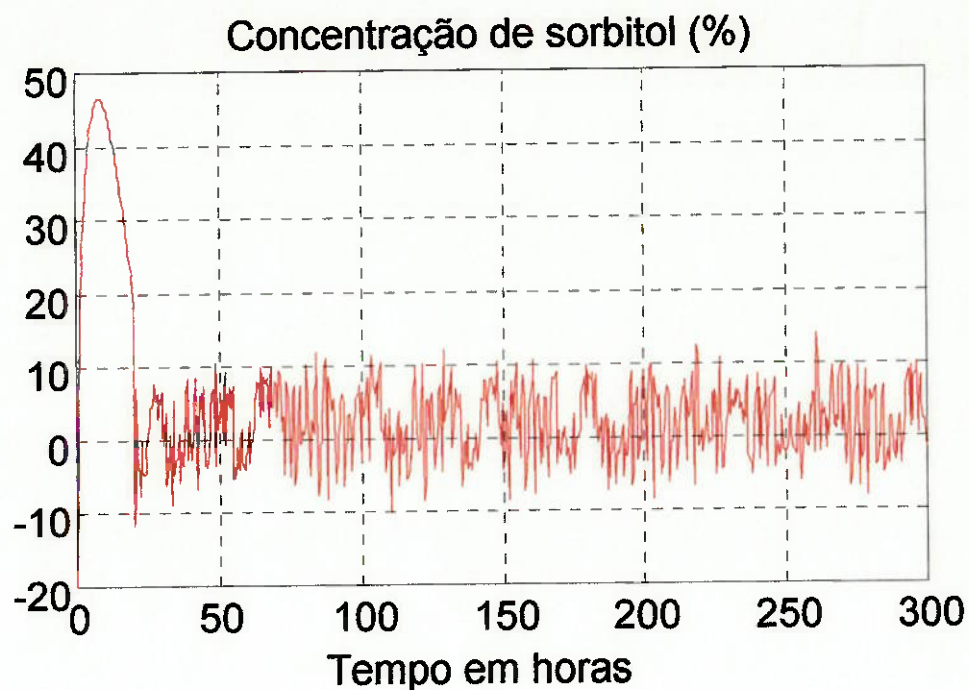
Simulação com 4% de ruído nas variáveis de estado e nas variáveis medidas, 20 horas de atraso para o início do controle e ganho do Filtro de Kalman multiplicado por 4.

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

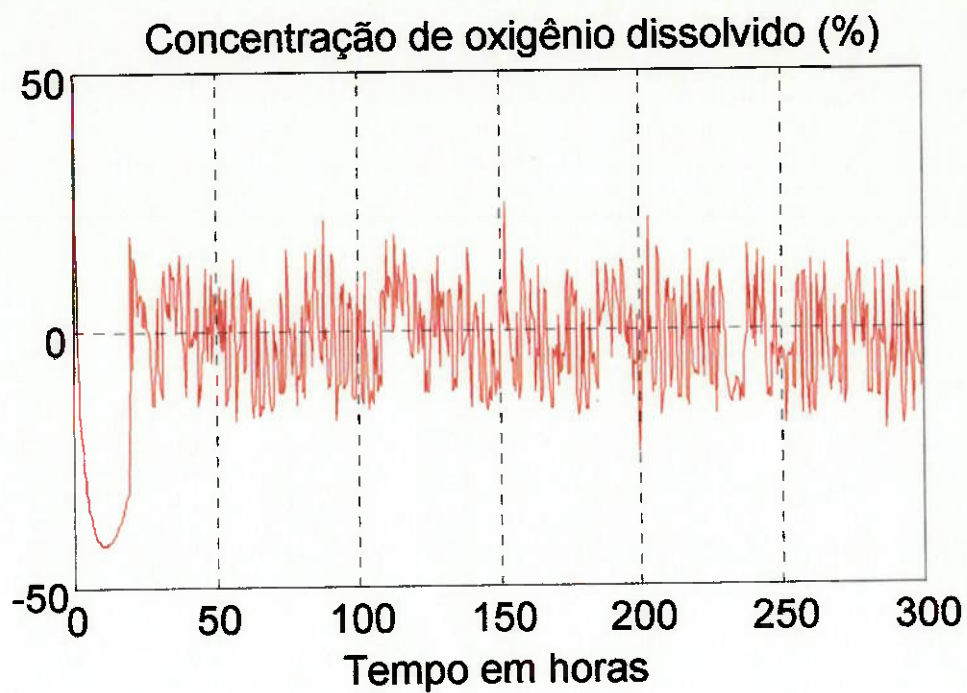
Variáveis de Estado:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

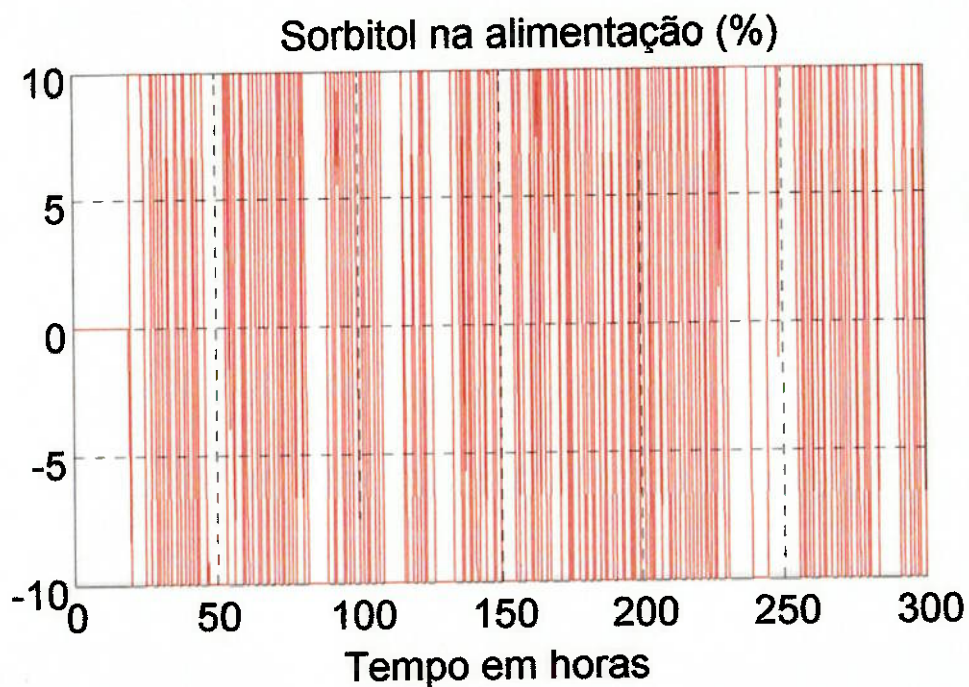
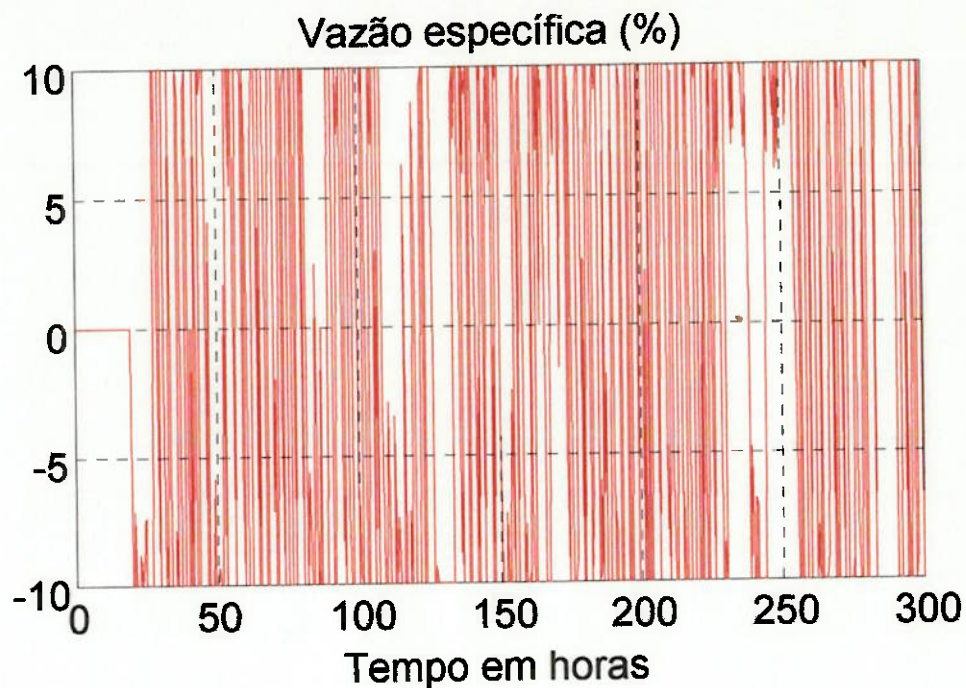


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

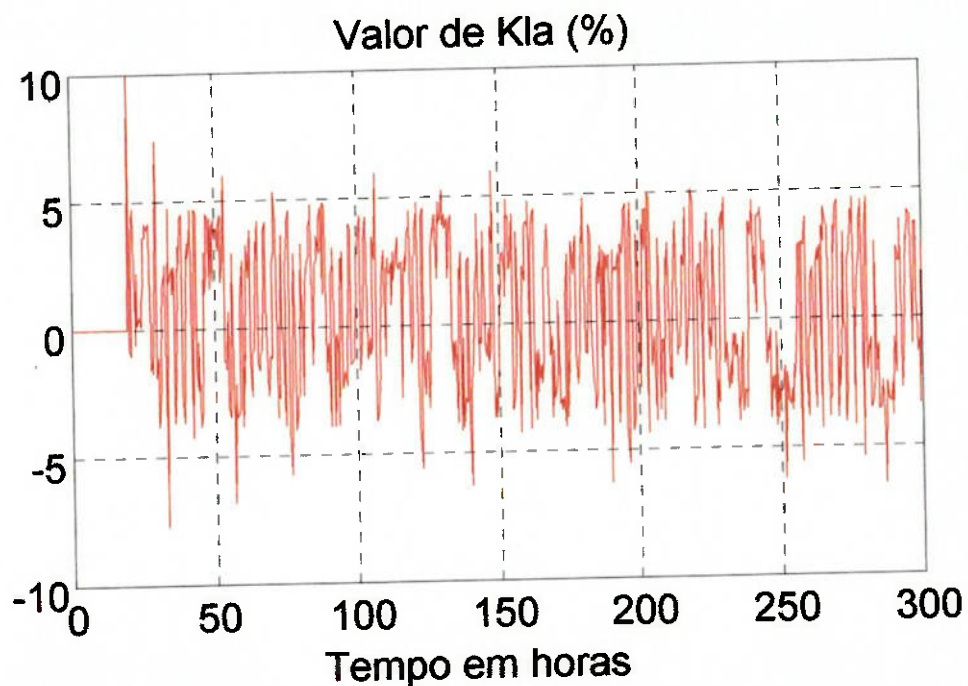
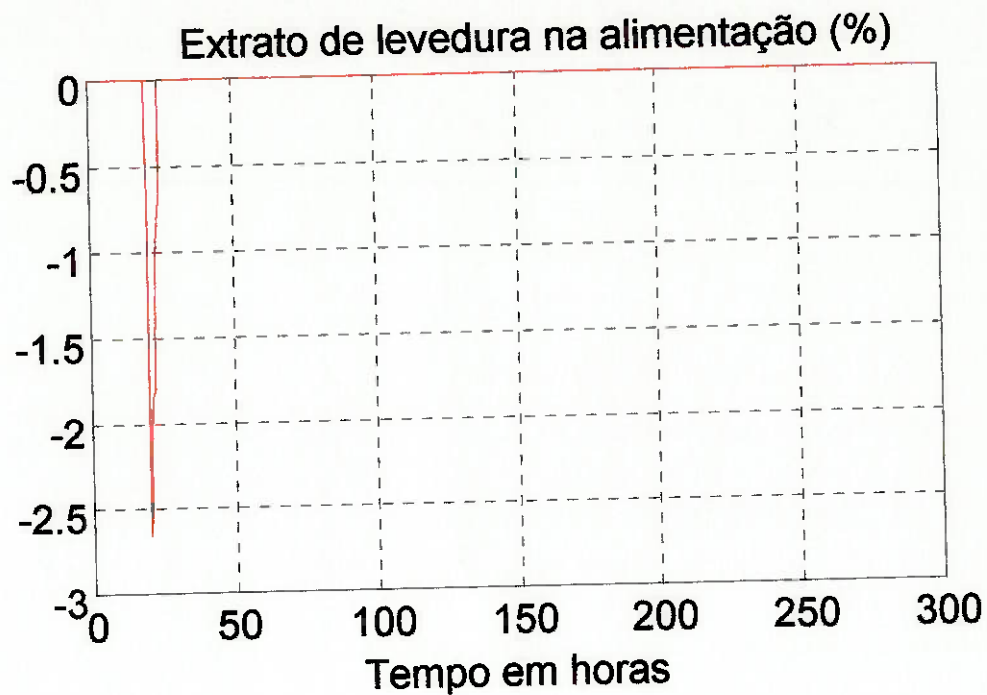


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Variáveis de Controle:

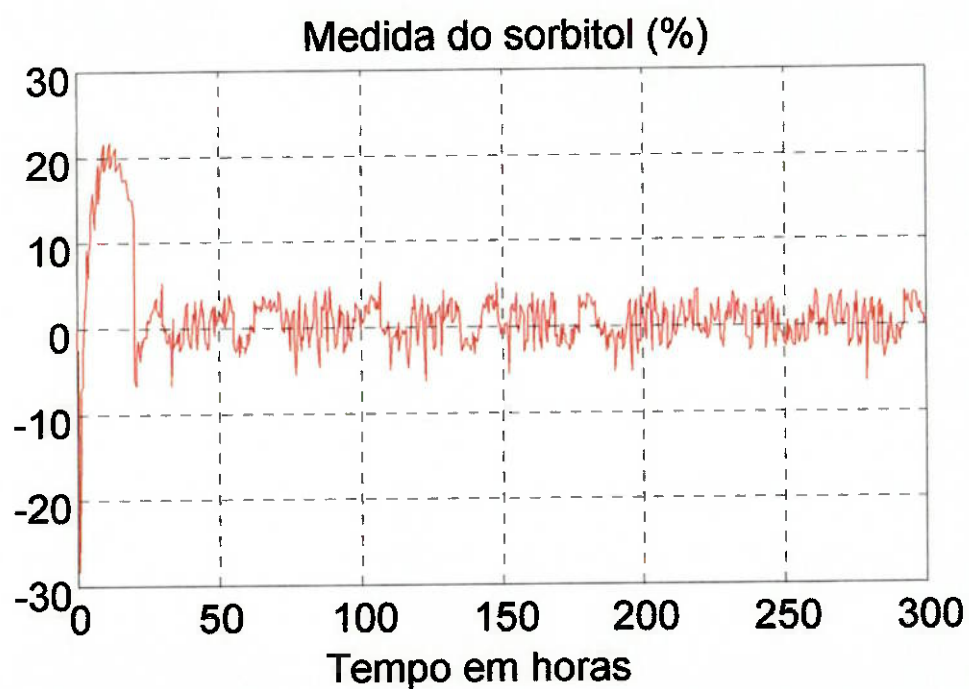
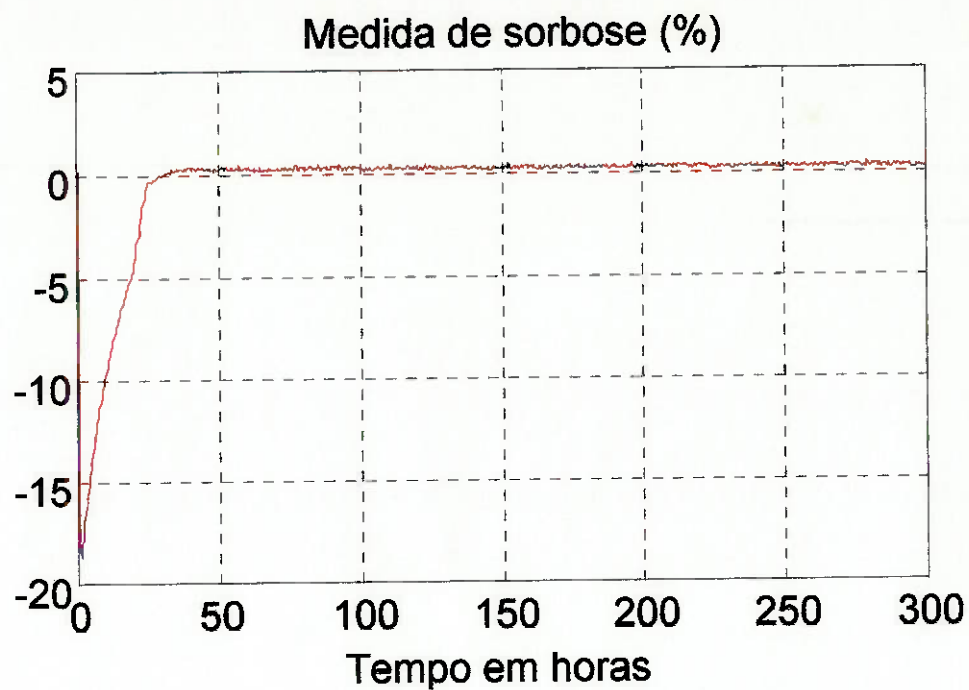


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

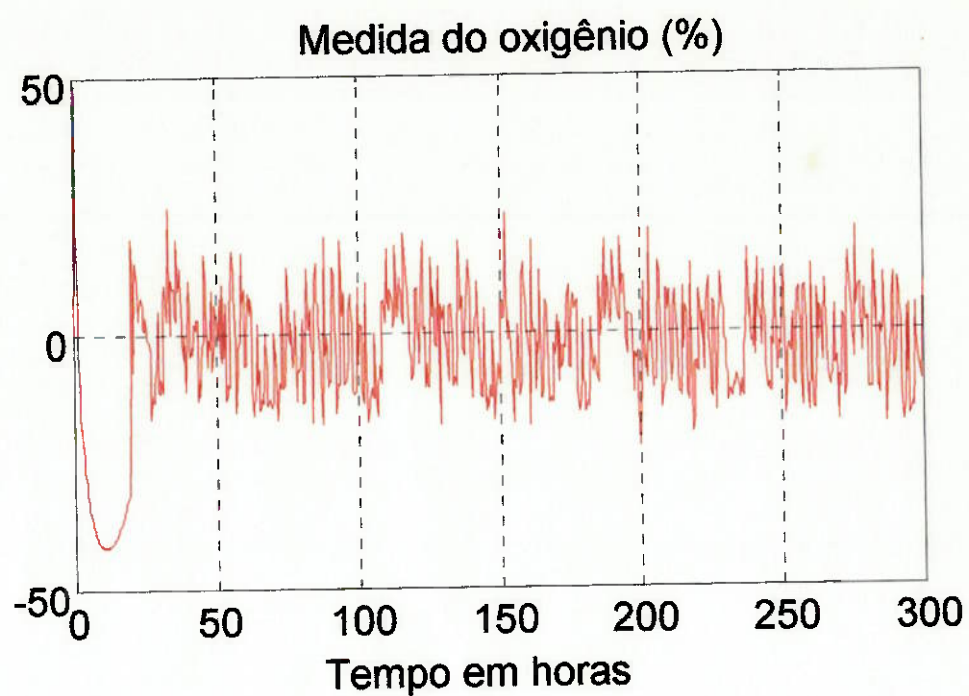


Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Variáveis Medidas:



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.



Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Protocolo de Simulação

Vetor Ue - Entradas para o ponto ótimo de controle

.800000000E-01
.130000000E+03
.351000000E+01
.327450000E+03

Matriz Xe - Ponto de Equilíbrio Ótimo adotado

.237700000E+01
.116870000E+03
.171600000E+01
.100000000E+01
.564000000E+00

Matriz A - Matriz de Parâmetros de estados do modelo linearizado

.000000000E+00 -.123794160E-02 .707919687E-01 .176010163E-01 .812277235E-01
.393306572E+01 -.115528924E+00 .359136491E+01 .505149169E+00 .558033286E+01
-.431723616E+01 .411919255E-01 -.401255173E+01 -.585665554E+00 -.598804914E+01
-.847457627E-01 .131137881E-02 -.749914922E-01 -.986451444E-01 -.860463173E-01
-.955666447E+03 .125364720E+02 -.855535230E+03 -.178243181E+03 -.143892113E+04

Matriz Fi

.999981172E+00 -.784903394E-06 .421348740E-04 .111597328E-04 .393510318E-04
.198542078E-02 .999920663E+00 .183598939E-02 .180172673E-03 .270530063E-02
-.221106331E-02 .161531388E-04 .997897628E+00 -.229664841E-03 -.290284112E-02
-.506743625E-04 .831465460E-06 -.446344004E-04 .999921514E+00 -.416854150E-04
-.463296555E+00 .608081778E-02 -.414739698E+00 -.864568839E-01 .302299469E+00

Matriz B - Matriz de Parâmetros dos sinais de controle do modelo linearizado

-.237700000E+01 .000000000E+00 .000000000E+00 .000000000E+00
-.116870000E+03 .000000000E+00 .000000000E+00 .000000000E+00
.128284000E+03 .800000000E-01 .000000000E+00 .000000000E+00
.251000000E+01 .000000000E+00 .800000000E-01 .000000000E+00
.693600000E+01 .000000000E+00 .000000000E+00 .693600000E+01

Matriz Gama

-.197813202E-02 .155751693E-08 .403827121E-09 .136024765E-06
-.972653142E-01 .716993845E-07 .819115086E-08 .934885760E-05
.106764081E+00 .665853199E-04 -.100006689E-07 -.100316676E-04
.208880533E-02 -.164991200E-08 .666640167E-04 -.144094031E-06
-.230530652E-01 -.165312790E-04 -.344534729E-05 .336908405E-02

Matriz C - Matriz de Medições de estado do modelo linearizado

.000000000E+00 .684521263E-01 .000000000E+00 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .466200466E+01 .000000000E+00 .000000000E+00
.000000000E+00 .000000000E+00 .000000000E+00 .000000000E+00 .709219858E+00

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Matriz L - Ganho do Filtro de Kalman

```
.354029393E+00 -.149809542E+02 .114531090E+00
.673189339E+00 -.266734164E+02 .786826933E+01
-.391645269E+00 .166718501E+02 -.844314928E+01
.336460176E-01 -.161046309E+01 -.121325307E+00
.000000000E+00 .000000000E+00 .281302727E+01
```

Matriz G1 - Ganho do controlador para o subsistema lento

```
.158744793E-01 .993981208E-02 -.330529012E+00 .136667582E-02
-.965998529E+01 -.179734813E+02 -.273850235E+04 -.163525113E+01
-.134234327E+01 .169365076E-01 -.476839228E-03 -.290876994E+00
.435123781E+02 -.578435716E+01 .207306491E+03 .904427383E+01
```

Matriz G2 - Ganho do controlador para o subsistema rápido

```
-.310214934E-04
.000000000E+00
.000000000E+00
-.103944787E+03
```

Valor de magO - Velocidade do observador

```
.300000000E+01
```

Valor de magR - Magnitude do controlador

```
.500000000E-01
```

Vetor PesoR - Controle

```
.400000000E+01
.800000000E+00
.200000000E+01
.200000000E+02
```

Vetor PesoC - Medidas

```
.800000000E+01
.800000000E+01
.400000000E+00
```

Valor de dtO (em minutos)

```
.500000000E-01
```

Matriz Fimod

```
.999897508E+00 -.196107628E-04 .563171873E-01 .763287219E-06 -.556097617E-13
.308971035E-02 .999868255E+00 .103153106E+00 .395485151E-03 .246150429E-11
-.347767784E-02 .548124670E-04 .934055264E+00 -.471782277E-03 -.271592254E-11
-.815103575E-04 -.654020565E-06 .598540775E-02 .999916321E+00 -.506062515E-13
-.462718077E+00 .607324063E-02 -.414228864E+00 -.863489575E-01 .300964342E+00
```

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Matriz Gamamod

.106471143E-02	.189859388E-05	.178185405E-09	-.635364098E-16	.000000000E+00
.195577371E-04	-.563534539E-01	.000000000E+00	-.744705022E-04	
-.918131090E-01	.347653285E-05	.134627460E-07	.873997689E-14	.000000000E+00
.372656704E-04	-.100334792E+00	.000000000E+00	.438984038E-02	
.103340529E+00	.644446656E-04	-.159049271E-07	-.953387153E-14	.000000000E+00 -
.216310831E-04	.627123854E-01	.000000000E+00	-.482369587E-02	
.241530071E-02	.201808024E-06	.666638936E-04	-.161177996E-15	.000000000E+00
.185077596E-05	-.605802349E-02	.000000000E+00	-.868120263E-04	
-.230384075E-01	-.165194776E-04	-.344285268E-05	.336488088E-02	.000000000E+00 -
.136842425E-07	.702352608E-05	.000000000E+00	.220621712E-02	

Dados iniciais para simulação (arquivo VIT_C.DAT)

-20.0	-20.0	-20.0	-20.0	-20.0	estados iniciais
0.00	0.00	0.00	0.00	0.00	observador inicial
0.00	tempo inicial (hora)				hoje 27/11/95 14h50m
300.00	tempo final (hora)				
3.00	duração do filtro do atuador (min)				
20.00	atraso para início do controle (horas)				
400.00	atraso para melhoria da medição (horas)				
1.e-6	precisão				
1.0e-5	passo inicial (chute)				
1.e-8	menor passo admissível (hora)				
-1.0	tempo entre resultados armazenados (segundos)				
400000	número máximo de passos				
0	=1 simula-se até o máximo de passos; =0 não há limite				
2	=1 contr nulo; =2 realimenta est esti; =3 real est reais (observ)				
2	=1 observador é o estado real; =2 obs discreto (despro)				
0	numero inteiro de dto's para atraso dos contr na memorex (shift)				
0.0	erro máximo aleatório para observadores iniciais (%)				
360000.0	tempo de aleatorização dos parâmetros (min);				
10.0	10.0	10.0	10.0	valores máximos de controle (%)	
-10.0	-10.0	-10.0	-10.0	valores mínimos de controle (%)	
0.0	30.0	30.0	0.0	-0.25	atrasos das medidas (min)
1.0	1.0	1.0	1.0	1.0	fatores de ajuste da propagação
0.5	0.5	0.5	0.0	0.5	deadband (%) p/ atuadores(4) e
0.0	0.0	0.0	0.0	0.0	_μ0, _Ks1, _Ks2, _W, __ (%)
0.0	0.0	0.0	0.0	0.0	_β0, _βs1, _βO, _KO, _Yxs1 (%)
0.0	0.0	0.0	12.0	7.0	_Yxs2, _Kp, _YxO, ruído, ruído
5.0					ruído1 (%)

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Observações:

- 1 - tempo entre resultados: se -1, o programa calcula (automático)
- 2 - cálculo do observador: se 0 temos cálculo normal, se 1 não calcularemos o observador, e se 2 teremos as medidas colocadas no lugar dos correspondentes estados estimados
- 3 - estados iniciais: valores em porcentagem
- 4 - observador: idem (se erro percentual máximo = 0.) (ver observação 6)
- 5 - atrasos das medidas em minutos; atrasos negativos eliminam propagação
- 6 - erro percentual máximo aleatório: se negativo, será o erro inicial para observadores de estado; se positivo, será erro (ruído) nos parâmetros do modelo não-linear (neste caso devemos ter observador discreto!)
- 7 - tempo de aleatorização: se 0, usaremos rotina "perturb1"
- 8 - passo inicial: se negativo, usaremos passo de integração fixo de valor igual ao módulo do passo inicial

Simulação realizada em 0h30m33.47s.

Final normal da simulação

Simulação com 12% de ruído nas variáveis de estado, 5% de ruído nas medidas de sorbitol e sorbose, 7% de ruído nas medidas de oxigênio, 20 horas de atraso para o início do controle e 10% de saturação em todas as variáveis de controle.

Discussão

Os resultados apresentados no item anterior demonstram claramente que os objetivos estão bem próximos de serem alcançados, deixando claro que os esforços não foram em vão. Frisa-se, entretanto, que tais resultados não atingiram o ponto ótimo, pois existem ainda parâmetros que podem ser otimizados.

Estão relacionadas abaixo, os principais problemas e as principais estratégias utilizadas na tentativa de se melhorar o desempenho do sistema.

- Modelagem do ruído

Para se projetar e simular o controle Linear Quadrático Gaussiano deve-se incluir no modelo um ruído de média zero. Para tanto deve-se conhecer a amplitude deste ruído. Durante as diversas simulações realizadas percebeu-se que independente do nível de ruído que era imposto na simulação e no projeto, as características gerais do resultado não se alteravam muito, apesar de ser claro que havia um ponto onde se melhor representaria este ruído. Frisa-se que a alteração do nível de ruído da planta não altera muito seu desempenho porque altera-se juntamente o projeto do observador.

Os valores encontrados como mais próximos do ideal durante as simulações estão dispostos abaixo: (Tais valores são bastante próximos com o que se constata na realidade).

- 12% de ruído em todas as variáveis de estado
- 5% de ruído nos valores medidos de Sorbitol e Sorbose
- 7% de ruído nos valores medidos de Oxigênio

- Utilização do Modelo Linearizado

Uma das características do Filtro de Kalman é que as suas estimativas se baseiam no modelo linearizado, ou seja a concepção do observador para do princípio de que será utilizado em modelos lineares, o que não é nosso caso. Assim a inclusão do ruído e do filtro de Kalman na simulação faz com que sempre tenhamos um fonte de erro devido a linearização o modelo,

- Utilização uma estratégia de controle estático.

A estratégia adotada, considera que o ruído é sempre o mesmo durante a evolução da planta, e que o erro é sempre o mesmo. Verifica-se que durante a simulação que essa hipótese não corresponde à realidade. Em resposta a isso deve-se mudar os ganhos do observador no sentido de se fazer estimativas das variáveis de estado baseados nos erros reais.

Algumas tentativas foram feitas no sentido de se mudar os ganhos do filtro de Kalman durante a simulação. Deve-se citar que conseguir resultados a partir dessa estratégia é bastante difícil devido ao grande leque de opções de alterações que pode-se simular. A tentativa marcante que foi feita, está demonstrada no item anterior e se refere a multiplicação do ganho do observador por um fator, no início da simulação e ir diminuindo esse fator ao longo desta. Como esta demonstrado pelos resultados, não se obteve indicação de que esforços neste sentido seriam vantajosos.

- Utilização de um valor limite para as variáveis de entrada do sistema

Os resultados das simulações mostram claramente que os valores de saturação das variáveis de controle são atingidos com frequência. Tal característica é bastante relevante, pois valores muito altos para as variáveis de controle fazem com que o sistema fuja do valor onde se fez a linearização deste. É bastante claro que valores altos para as variáveis de controle fazia a planta estabilizar em outro ponto. Assim para se resolver este problema, deve se encontrar um ponto ótimo de limite para as variáveis de controle, que impeça que o sistema divirja.

Para se chegar ao resultado final apresentado nesta dissertação, além da estratégia acima citada, utilizou-se também o recurso de se atrasar em 20 horas o início do controle. Tal recurso faz com que a planta atinja níveis menores de desvio em relação ao ponto de equilíbrio antes de se iniciar o controle, assim os valores de erro são menores e os sinais de controle também.

Uma característica marcante dos resultados se refere ao fato de todos apresentarem uma oscilação muito grande, o que já era esperado pela presença do ruído.

É importante frisar que o estudo sobre a aplicação de controle LQG em processos fermentativos ainda está longe de ser concluído, deu-se apenas um passo na direção de um melhor entendimento destes e de seus principais problemas.