

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE TELEMÁTICA

Este exemplar corresponde à redação final da tese
defendida por LUIZA YOKO TANEGUTI
e aprovada pela Comissão
Julgadora em 11/02/93.

Basílio E. A. Milani
Orientador

**“Identificação Adaptativa de Sistemas através de Mínimos
Quadrados Lineares utilizando Fatoração QR e
Janela Móvel de Dados”**

Luiza Yoko Taneguti n.º 155

Dissertação submetida à Faculdade de Engenharia Elétrica da
Universidade Estadual de Campinas, como parte dos requisitos
para obtenção do título de Mestre em Engenharia Elétrica.

Orientador : Prof. Dr. Basílio E. A. Milani

Campinas - SP - Fevereiro de 1993

Ao meu “Pedro Y.”,
com amor.

Ao meu pai Mituzi, com saudades.
À minha mãe Helena.

AGRADECIMENTOS

Ao prof. Dr. Basílio E. A. Milani, por sua amizade, paciência, experiência e por todas as oportunidades que me proporcionou. Espero ter adquirido um pouco da sua postura profissional.

Aos meus irmãos e irmãs : Iti, Wal, Nita e Chila pelo apoio que sempre transmitiram.

À baiana Isa, ao boto Gugu e ao ibitinguense Fábio pela grande amizade.

Aos fãs Menotti, Luís, Robertão, Celso e Adilson pelo carinho.

A todos os colegas e professores do IMECC e da FEE(principalmente do DT).

À CAPES pelo apoio financeiro.

À Faculdade de Engenharia Elétrica e à Universidade de Campinas pela aceitação.

RESUMO

Este trabalho trata da solução numérica de problemas de mínimos quadrados lineares para identificação adaptativa de sistemas variantes no tempo. Algoritmos recursivos tradicionais utilizando as equações normais e ponderação exponencial dos dados são analisados e suas deficiências numéricas, na solução de problemas de mínimos quadrados mal condicionados, são explicitadas. É proposto um novo algoritmo recursivo utilizando fatorações ortogonais QR e janela móvel de dados, numericamente estável, particularmente recomendado para solução computacional de problemas inerentemente mal condicionados. Resultados de simulação significativos são apresentados, ilustrando o desempenho dos seguintes algoritmos na identificação de um modelo ARMA : equações normais com ponderação exponencial de dados, fatoração QR com ponderação exponencial de dados, equações normais com janela móvel de dados e o algoritmo proposto utilizando fatoração QR com janela móvel de dados.

ABSTRACT

This work deals with the numerical solution of linear least-squares problems for adaptive identification of time-varying systems. Traditional recursive algorithms using normal equations and exponential data weighting are analysed and their numerical drawbacks for solution of ill-conditioned least-squares problems are pointed out. It is proposed a new recursive algorithm using orthogonal QR factorizations with sliding window on the data, numerically stable, being particularly recommended for solution of ill-conditioned problems. Significant simulation results are presented, illustrating the performance of the following algorithms identifying an ARMA model : normal equations with exponential data weighting, QR factorization with exponential data weighting, normal equations with sliding data window and the proposed algorithm using QR factorizations with sliding data window.

Conteúdo

1	Introdução	2
2	Estimação por Mínimos Quadrados Lineares	4
2.1	Definição do problema	5
2.2	Método utilizando Equações Normais	7
2.3	Método baseado em Transformações Ortogonais	9
3	Identificação Adaptativa de sistemas variantes no tempo	12
3.1	Métodos de Mínimos Quadrados com Ponderação Exponencial dos dados	14
3.1.1	Definição do Problema	14
3.1.2	Algoritmo Recursivo utilizando as Equações Normais	16
3.1.3	Algoritmo Recursivo baseado em Fatorações Ortogonais	23
3.1.4	Conclusões	27
3.2	Métodos de Mínimos Quadrados com Janela Móvel de dados	28
3.2.1	Definição do Problema	28
3.2.2	Algoritmo Recursivo utilizando as Equações Normais	31
3.2.3	Algoritmo Recursivo Proposto baseado em Fatorações Ortogonais	35
3.2.4	Filtragem dos Parâmetros	43
3.2.5	Conclusões	45
4	Exemplos Numéricos	46
4.1	Exemplo1	47
4.2	Exemplo2	74
5	Conclusões Gerais	80
	Apêndice	82
	Referências	92

Capítulo 1

Introdução

Estimação de parâmetros por mínimos quadrados de sistemas lineares, é de grande importância em muitas aplicações, tais como : Análise de séries temporais, comunicações, análise espectral, controle, etc. [2], [3], [6], [8].

A identificação adaptativa por mínimos quadrados de sistemas lineares com parâmetros variantes no tempo, é geralmente obtida através de algoritmos recursivos utilizando as “Equações Normais de mínimos quadrados” com Ponderação Exponencial dos dados, para levar em conta a maior importância das informações atuais sobre as do passado [3] [6].

A utilização conjunta de equações normais e ponderação exponencial leva a algoritmos recursivos simples, com baixo custo computacional, mas com qualidade numérica precária, quando utilizados em problemas de mínimos quadrados mal condicionados, porque tanto a montagem das equações normais, quanto a utilização da ponderação exponencial, pioram sobremaneira o condicionamento original, inerente ao problema a ser resolvido [5], [7].

Formas alternativas de contornar este problema estão propostos neste trabalho. Uma é a utilização de Fatorações Ortogonais QR ao invés de “Equações Normais” e outra é usar janela móvel no lugar de ponderação exponencial dos dados. Ambas contribuem significativamente para a melhoria da estabilidade numérica dos algoritmos e portanto da precisão final dos resultados obtidos. Com essa mudança, ou seja usando Fatorações Ortogonais com janela móvel de dados não há alteração no condicionamento original do problema de mínimos quadrados, sendo especialmente indicado para problemas inerentemente mal condicionados [12].

No capítulo 2 além da apresentação do problema de mínimos quadrados, as duas formas de resolvê-lo também serão apresentadas, ou seja, o método utilizando “Equações Normais” e o método baseado em Transformações Ortogonais.

No capítulo 3, o problema de identificação adaptativa de sistemas variantes no tempo utilizando o método de mínimos quadrados recursivo é tratado. Para a sua solução são apresentados métodos combinando os dois enfoques, ponderação exponencial de dados e janela móvel de dados, com as duas formas de solução de problemas de mínimos quadrados, ou seja, equações normais e fatorações ortogonais.

No capítulo 4, exemplos numéricos ilustram a eficiência computacional e a qualidade dos resultados fornecidos pelos algoritmos descritos no capítulo anterior.

Para finalizar, no capítulo 5 são apresentadas as conclusões gerais sobre os resultados apresentados e perspectivas de futuros desenvolvimentos.

Capítulo 2

Estimação por mínimos quadrados lineares

Neste capítulo, o problema de estimação de parâmetros utilizando mínimos quadrados lineares é definido. Para sua solução são apresentadas duas abordagens conhecidas, “equações normais” e fatorações ortogonais.

2.1 Definição do problema

Consideremos o problema de estimar um vetor de parâmetros h de dimensões $(n, 1)$, solução da equação de regressão linear

$$y(i) = a(i)h(i) + \omega(i) \quad (2.1)$$

$$a(i) = [a_{i1} \ a_{i2} \ a_{i3} \ \dots \ a_{in}]$$

onde : $a(i)$ é um vetor de entrada de dimensões $(1, n)$, $\omega(i)$ é um ruído branco com média zero e $y(i)$ é um sinal de referência no instante i .

Assumindo h estacionário no intervalo $0 < k < i$, a estimação por mínimos quadrados de h no instante i , corresponde à solução do seguinte problema de otimização :

$$\min_{h(i)} J(i) = [h(i) - h(0)]^T A_0 [h(i) - h(0)] + \sum_{k=1}^i [y(k) - a(k)h(i)]^2 \quad (2.2)$$

onde : A_0 é uma matriz definida positiva representando o peso da estimativa a priori $h(0)$ relativamente aos dados no intervalo $0 < k < i$.

As propriedades de ruído branco com média zero assumidas para $\omega(i)$, garantem a não polarização da estimativa $h(i)$ [6],[8]

Podemos formular o problema de mínimos quadrados(PMQ) da seguinte forma matricial :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} \right\|_2 \quad (2.3)$$

onde :

$$A(i) = \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(i) \end{bmatrix} \text{ matriz de dados de entrada de dimensões } (i,n),$$

$$Y(i) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(i) \end{bmatrix}, \text{ vetor dos sinais de referência de dimensões } (i,1),$$

$A_0 = R(0)^T R(0)$, $R(0)$ matriz triangular superior de dimensões (n,n) obtida da fatoração de Cholesky de A_0 .

Podemos obter a solução do PMQ (2.3) de duas formas principais : método das equações normais e método de fatoração ortogonal. Veremos nas próximas seções como cada método resolve o PMQ (2.3) juntamente com uma análise de suas vantagens e desvantagens computacionais.

2.2 Método utilizando Equações Normais

Para obter a solução do PMQ(2.3) podemos impor a condição de ponto estacionário :

$$\nabla J(i) = 0 \quad (2.4)$$

onde $\nabla J(i)$ é o vetor gradiente de $J(i)$.

Isto corresponde a [5], [6], [8],

$$\nabla J(i) = \begin{bmatrix} R(0) \\ A(i) \end{bmatrix}^T \left\{ \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} - \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} [h(i) - h(0)] \right\} = 0 \quad (2.5)$$

Rearranjando os termos chegamos à “equação normal”

$$\begin{bmatrix} R(0) \\ A(i) \end{bmatrix}^T \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} [h(i) - h(0)] = \begin{bmatrix} R(0) \\ A(i) \end{bmatrix}^T \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} \quad (2.6)$$

que podemos escrever na forma condensada :

$$X^T X [h(i) - h(0)] = X^T b \quad (2.7)$$

onde :

$$X = \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} \quad (2.8)$$

$$b = \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} \quad (2.9)$$

A solução $h(i)$ do PMQ (2.3) é obtida da solução do sistemas de equações lineares (2.7) de ordem n .

Podemos ver que o esforço computacional para obtenção de $h(i)$ é devido principalmente a :

- Produto de matrizes $X^T X$: $n^2(n + i)$ FLOPS
- Produto de matrizes $A(i)h(0)$: ni FLOPS
- Produto de matrizes $X^T b$: ni FLOPS
- Solução do sistema de equações (2.6) pelo método de Cholesky : $\frac{n^3}{6} + \frac{n^2}{2}$ FLOPS

onde FLOPS é o número de operações de mult. e div. de ponto flutuante.

Totalizando, temos o seguinte esforço computacional

$$FLOPS = n^2(n + i) + \frac{n^3}{6} + 2ni + \frac{n^2}{2} \quad (2.10)$$

A precisão da solução do sistema de equações (2.7) depende do número condição da matriz do sistema $X^T X$.

É fácil verificar que [5] :

$$cond(X^T X) = [cond(X)]^2 = \left\{ cond \left[\begin{array}{c} R(0) \\ A(i) \end{array} \right] \right\}^2 \quad (2.11)$$

onde $cond(.)$ é o número condição de $(.)$.

Portanto vemos que o número condição original do PMQ (2.3), $cond\left(\begin{bmatrix} R(0) \\ A(i) \end{bmatrix}\right)$ foi elevado ao quadrado na obtenção das “equações normais”.

2.3 Método baseado em Transformações Ortogonais

O método de solução de problemas de mínimos quadrados lineares utilizando fatorações ortogonais se baseiam nas seguintes propriedades das transformações ortogonais [13].

$$\|Qz\|_2 = \|z\|_2 \quad (2.12)$$

$$\text{cond}(QG) = \text{cond}(G) \quad (2.13)$$

onde :

z é um vetor de dimensões $(n,1)$,

Q é uma matriz ortogonal de dimensões (n,n) , $(Q^T Q = I_n)$,

G é uma matriz com n linhas,

$\|\cdot\|_2$ é o quadrado da norma euclidiana e

$\text{cond}(\cdot)$ é o número condição de (\cdot) .

A propriedade (2.12) permite transformar o índice de desempenho quadrático do PMQ (2.3) da seguinte forma :

$$\begin{aligned} \min_{h(i)} J(i) &= \left\| \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} \right\|_2 = \\ &= \left\| Q \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} [h(i) - h(0)] - Q \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} \right\|_2 \end{aligned}$$

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right\|_2 \quad (2.14)$$

sendo

$$Q \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} ; Q \begin{bmatrix} 0 \\ Y(i) - A(i)h(0) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2.15)$$

onde :

Q matriz ortogonal de dimensões $(n+i, n+i)$,

R matriz triangular superior de dimensões (n, n) ,

b_1 vetor de dimensões $(n, 1)$ e

b_2 vetor de dimensões $(i, 1)$.

A matriz Q pode ser sempre obtida de forma eficiente e numericamente estável através do produto de uma sequência de matrizes de Householder ou Givens [13]. Para a obtenção da matriz $\begin{bmatrix} R \\ 0 \end{bmatrix}$ e do vetor $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, não é necessário construir Q explicitamente [13].

Devido à estrutura de $\begin{bmatrix} R \\ 0 \end{bmatrix}$, é fácil verificar que a solução do PMQ (2.3) é obtida da solução do seguinte sistemas de equações :

$$R[h(i) - h(0)] = b_1 \quad (2.16)$$

Baseado em (2.13) e (2.15) temos :

$$\text{cond} \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} = \text{cond} \begin{bmatrix} R \\ 0 \end{bmatrix} = \text{cond}(R) \quad (2.17)$$

mostrando que a solução do PMQ (2.3) através de (2.16) é feita sem nenhum agravamento do condicionamento original do problema.

O esforço computacional para obtenção de $h(i)$ solução do PMQ (2.3) é devido à obtenção de R e b_1 (2.15) e solução do sistema triangular de equações (2.16). Utilizando matrizes de Householder para obtenção de R e b_1 e substituição regressiva para solução do sistema (2.16), o esforço computacional para obtenção de $h(i)$ é estimado em :

- obtenção de $R(0)$ (fatoração de Cholesky) : $\frac{n^3}{6}$ FLOPS
- obtenção de R e b_1 : $n^2(n + i - \frac{n}{3}) + 2n^2$ FLOPS
- solução do sistema triangular (2.16) : $\frac{n^2}{2}$ FLOPS

onde FLOPS é o número de operações de multiplicações de ponto flutuante.

Totalizando, temos o seguinte esforço computacional

$$FLOPS = n^2(n + i - \frac{n}{6} + \frac{5}{2}) \quad (2.18)$$

Comparando os métodos das equações normais e fatorações ortogonais na solução do PMQ (2.3), conclue-se que :

— a diferença do esforço computacional requerido pelos métodos das “equações normais” e fatoração ortogonal é pequena veja (2.10) e (2.18), podendo ser considerados como equivalentes. A principal desvantagem das “equações normais” é o agravamento do condicionamento original do PMQ (2.3), que é elevado ao quadrado, veja (2.11).

— a principal vantagem do método das fatorações ortogonais é a qualidade numérica de seus resultados, devido à manutenção do condicionamento original do PMQ (2.3) veja (2.13), o que torna este método particularmente recomendado para problemas inerentemente mal condicionados ($cond \begin{bmatrix} R(0) \\ A(i) \end{bmatrix}$ alto).

Capítulo 3

Identificação Adaptativa de Sistemas Lineares variantes no tempo

A identificação adaptativa por mínimos quadrados lineares de sistemas variantes no tempo, é geralmente obtida através de algoritmos recursivos utilizando as equações normais de mínimos quadrados com ponderação exponencial dos dados, para levar em conta a maior importância das informações atuais sobre as do passado [3], [6].

O uso das equações normais, conforme mostrado na seção 2.2 agrava o condicionamento original do problema. O uso de ponderação exponencial também contribui para o agravamento do seu condicionamento, fazendo com que o uso conjunto de “equações normais” e ponderação exponencial de dados não seja recomendado para problemas inerentemente mal condicionados.

Formas alternativas de contornar este problema, a utilização de fatoração ortogonal QR no lugar das “equações normais” e janela móvel de dados no lugar de ponderação exponencial de dados, são apresentadas neste capítulo.

Na seção 3.1, o problema de mínimos quadrados com ponderação exponencial dos dados é definido e para sua solução recursiva são apresentados e discutidos um algoritmo baseado nas “equações normais” e um algoritmo utilizando fatorações ortogonais QR.

Na seção 3.2, o problema de mínimos quadrados com janela móvel de dados será também definido e para sua solução são apresentados e discutidos um algoritmo

baseado nas “equações normais” e um novo algoritmo completamente baseado em transformações ortogonais QR [12].

3.1 Métodos de mínimos quadrados com Ponderação Exponencial dos dados

3.1.1 Definição do Problema

Os algoritmos que iremos apresentar se baseiam nos métodos apresentados no capítulo anterior.

Com a finalidade de seguir as não estacionaridades de um processo, um fator de esquecimento exponencial W é introduzido ao PMQ (2.2). Da mesma forma, considere o problema de estimar um vetor de parâmetros h de dimensões $(n,1)$ com variação temporal lenta, solução da equação de regressão linear dados em (2.1), então, assumindo h constante em $0 < k < i$, a estimação por mínimos quadrados de h no instante i , corresponde à solução do seguinte problema :

$$\min_{h(i)} J(i) = [h(i) - h(0)]^T A_0 [h(i) - h(0)] + \sum_{k=1}^i W^{i-k} [y(k) - a(k)h(i)]^2 \quad (3.1)$$

onde : $A_0 = R(0)^T W^{2i} R(0)$ é uma matriz definida positiva representando o peso da estimativa a priori $h(0)$ relativamente aos dados no intervalo $0 < k < i$ e W é o fator de esquecimento $0 < W < 1$.

Sendo assim podemos formular o nosso problema de mínimos quadrados com ponderação exponencial dos dados (PMQPE) na seguinte forma matricial

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} W^i R(0) \\ \bar{W}(i) A(i) \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} 0 \\ \bar{W}(i) (Y(i) - A(i)h(0)) \end{bmatrix} \right\|_2 \quad (3.2)$$

onde

$$\bar{W}(i) = \begin{bmatrix} W^{i-1} & & & & \\ & \ddots & & & \\ & & W^2 & & \\ & & & W & \\ & & & & 1 \end{bmatrix} \quad \text{de dimensões}(i, i) \quad (3.3)$$

$$A(i) = \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(i) \end{bmatrix} \text{ matriz de dados de entrada de dimensões } (i,n),$$

$$Y(i) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(i) \end{bmatrix}, \text{ vetor dos sinais de referência de dimensões } (i,1),$$

fazendo

$$X(i) = \begin{bmatrix} W^i R(0) \\ \bar{W}(i) A(i) \end{bmatrix} \quad (3.4)$$

$$b(i) = \begin{bmatrix} 0 \\ \bar{W}(i)(Y(i) - A(i)h(0)) \end{bmatrix} \quad (3.5)$$

Podemos escrever (3.2) recursivamente da seguinte forma

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} W X(i-1) \\ a(i) \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} W b(i-1) \\ y(i) - a(i)h(0) \end{bmatrix} \right\|_2 \quad (3.6)$$

3.1.2 Algoritmo Recursivo utilizando as “Equações Normais”

Para o PMQPE(3.6) a equação normal no instante i é dada por

$$[W^2 X^T(i-1)X(i-1) + a^T(i)a(i)][h(i) - h(0)] = W^2 X^T(i-1)b(i-1) + a^T(i)[y(i) - a(i)h(0)] \quad (3.7)$$

No instante $i-1$, a equação normal é :

$$X^T(i-1)X(i-1)[h(i-1) - h(0)] = X^T(i-1)b(i-1) \quad (3.8)$$

Multiplicando (3.8) por W^2 e incorporando adequadamente $a^T(i)a(i)$, temos:

$$\begin{aligned} [a^T(i)a(i) + W^2 X^T(i-1)X(i-1)][h(i-1) - h(0)] = \\ W^2 X^T(i-1)b(i-1) + a^T(i)a(i)[h(i-1) - h(0)] \end{aligned} \quad (3.9)$$

Subtraindo (3.9) de (3.7), temos :

$$[W^2 X^T(i-1)X(i-1) + a^T(i)a(i)][h(i) - h(i-1)] = a^T(i)[y(i) - a(i)h(i-1)] \quad (3.10)$$

onde a solução é dada por

$$h(i) - h(i-1) = [W^2 X^T(i-1)X(i-1) + a^T(i)a(i)]^{-1} a^T(i)[y(i) - a(i)h(i-1)] \quad (3.11)$$

definindo

$$S(i-1) = [W^2 X^T(i-1)X(i-1)]^{-1} \quad (3.12)$$

$$S(i) = [W^2 X^T(i-1)X(i-1) + a^T(i)a(i)]^{-1} \quad (3.13)$$

Pelo Lema de Inversão de Matrizes na seção A.4 temos

$$S(i) = \frac{1}{W^2} \left[S(i-1) - \frac{S(i-1)a^T(i)a(i)S(i-1)}{W^2 + a(i)S(i-1)a^T(i)} \right] \quad (3.14)$$

definindo

$$g(i) = \frac{S(i-1)a^T(i)}{W^2 + a(i)S(i-1)a^T(i)} \quad (3.15)$$

temos então :

$$S(i) = \frac{1}{W^2} [S(i-1) - g(i)a(i)S(i-1)] \quad (3.16)$$

Através da equação (3.16), fazendo alguns acertos algébricos :

$$\begin{aligned} g(i) &= \frac{S(i-1)a^T(i)}{W^2 + a(i)S(i-1)a^T(i)} \cdot \frac{W^2 + a(i)S(i-1)a^T(i) - a(i)S(i-1)a^T(i)}{W^2} = \\ &= \frac{W^2 + a(i)S(i-1)a^T(i)S(i-1)a^T(i) - S(i-1)a^T(i)a(i)S(i-1)a^T(i)}{W^2[W^2 + a(i)S(i-1)a^T(i)]} = \end{aligned}$$

$$= \frac{1}{W^2} S(i-1)a^T(i) - \frac{1}{W^2} g(i)a(i)S(i-1)a^T(i)$$

finalmente

$$g(i) = \frac{1}{W^2} [S(i-1) - g(i)a(i)S(i-1)] \cdot a^T(i)$$

Substituindo a equação (3.16) na última equação temos :

$$g(i) = S(i)a^T(i) \quad (3.17)$$

Para o cálculo dos parâmetros $h(i)$, temos que, para um determinado instante $i-1$:

$$h(i-1) - h(0) = S(i-1) \cdot p(i-1) \quad (3.18)$$

para o instante i , temos :

$$h(i) - h(0) = S(i)p(i) \quad (3.19)$$

onde

$$S(i) = [W^2 X^T(i-1)X(i-1) + a^T(i)a(i)]^{-1},$$

$$p(i) = W^2 p(i-1) + a^T(i) \bar{b}(i)$$

$$\text{sendo } p(i-1) = X^T(i-1)b(i-1), \bar{b}(i) = y(i) - a(i)h(0)$$

Substituindo a equação (3.16) temos :

$$\begin{aligned} h(i) - h(0) &= \frac{1}{W^2} [S(i-1) - g(i)a(i)S(i-1)] [W^2 p(i-1) + a^T(i) \bar{b}(i)] = \\ &= S(i-1)p(i-1) + \frac{1}{W^2} S(i-1) a^T(i) \bar{b}(i) - g(i)a(i)S(i-1)p(i-1) - \\ &\quad - \frac{1}{W^2} g(i)a(i)S(i-1) a^T(i) \bar{b}(i) \end{aligned}$$

Usando (3.16) e (3.18)

$$\begin{aligned} h(i) - h(0) &= h(i-1) - h(0) - g(i)a(i)[h(i-1) - h(0)] + \\ &\quad + \frac{1}{W^2} [S(i-1) - g(i)a(i)S(i-1)] a^T(i) \bar{b}(i) = \\ &= h(i-1) - h(0) - g(i)a(i)[h(i-1) - h(0)] + S(i)a^T(i) \bar{b}(i) = \\ &= [h(i-1) - h(0)] - g(i)a(i)[h(i-1) - h(0)] + g(i) = \bar{b}(i) \\ &= [h(i-1) - h(0)] - g(i)[y(i) - a(i)h(0)] - a(i)[h(i-1) - h(0)] \end{aligned}$$

Assim sendo, temos

$$h(i) - h(0) = h(i - 1) - h(0) + g(i)[y(i) - a(i)h(i - 1)]$$

o que corresponde à recorrência

$$h(i) = h(i - 1) + g(i)e(i) \tag{3.20}$$

onde $e(i) = y(i) - a(i)h(i-1)$ é a estimativa do erro da regressão a priori no instante i .

Na tabela 1 a seguir, estão resumidos os passos do algoritmo recursivo utilizando equações normais LSNEPE para solução do PMQPE (3.6).

TABELA1 Algoritmo de Mínimos Quadrados utilizando Equações Normais com Ponderação Exponencial dos dados LSNEPE

Inicialização

$$S(0) = A_0^{-1} \quad (3.21)$$

Atualizações

— ganho de adaptação

$$v(i) = S(i-1)a^T(i) \quad (3.22)$$

$$g(i) = \frac{v(i)}{W^2 + a(i)v(i)} \quad (3.23)$$

— erro de estimação

$$e(i) = y(i) - a(i)h(i-1) \quad (3.24)$$

— parâmetros do sistema

$$h(i) = h(i-1) + g(i)e(i) \quad (3.25)$$

— matriz inversa de autocorrelação

$$S(i) = \frac{1}{W^2} [S(i-1) - g(i)v^T(i)] \quad (3.26)$$

Conforme pode ser visto na tabela 1, o esforço computacional requerido por iteração do algoritmo LSNEPE é devido majoritariamente às operações correspondentes às equações (3.22) a (3.26). Para estas equações o número de operações de multiplicação/divisão de ponto flutuante(FLOPS) é dado na tabela 2.

TABELA 2 Esforço computacional LSNEPE

Equação	FLOPS
(3.22)	n^2
(3.23)	$2n + 1$
(3.24)	n
(3.25)	n
(3.26)	$2n^2$
total	$3n^2 + 4n + 1$

Das equações (3.4) e (3.13) é fácil ver que :

$$\text{cond}(S(i)) = [\text{cond}(X(i))]^2 \quad (3.27)$$

o que mostra que o condicionamento original do PMQPE é bastante prejudicado no caso de $\text{cond}(X(i))$ ser alto.

Além disso, o lema de inversão de matrizes utilizado para a obtenção de $S(i)$ (3.14) faz com que os erros devido a um mal condicionamento inicial de $S(i)$ não se recuperem posteriormente, mesmo que o condicionamento de $X(i)$ (condicionamento inerente ao PMQPE (3.2) venha a melhorar ao longo da aquisição dos dados [5].

A multiplicação a cada iteração de $X(i)$ pelo fator W utilizada pelo PMQPE leva quase sempre a $X(i)$ mal condicionados.

3.1.3 Algoritmo Recursivo baseado em Fatorações ortogonais

Para o PMQPE (3.2), num dado instante $i-1$, utilizando uma transformação ortogonal $Q(i-1)$, ele se torna :

$$\min_{h(i)} J(i) = \|Q(i-1)X(i-1)[h(i-1) - h(0)] - Q(i-1)b(i-1)\|_2 \quad (3.28)$$

onde :

$$Q(i-1)X(i-1) = \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} \quad Q(i-1)b(i-1) = \begin{bmatrix} b_1(i-1) \\ b_2(i-1) \end{bmatrix}$$

Devido à estrutura de $\begin{bmatrix} R(i-1) \\ 0 \end{bmatrix}$, é fácil verificar que a solução do PMQPE (3.6) é obtida da solução do sistema de equações :

$$R(i-1)[h(i-1) - h(0)] = b_1(i-1) \quad (3.29)$$

No instante i o PMQPE (3.6), utilizando a transformação $Q(i-1)$ fica sendo :

$$\min_{h(i)} J(i) = \left\| Q(i-1) \begin{bmatrix} WX(i-1) \\ a(i) \end{bmatrix} [h(i-1) - h(0)] - Q(i-1) \begin{bmatrix} Wb(i-1) \\ y(i) - a(i)h(0) \end{bmatrix} \right\|_2 \quad (3.30)$$

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} b_1(i-1) \\ b_2(i-1) \\ y(i) - a(i)h(0) \end{bmatrix} \right\|_2 \quad (3.31)$$

com alguns acertos algébricos, para introdução de $h(i)-h(i-1)$, temos :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i) - h(0) + h(i-1) - h(i-1)] - \begin{bmatrix} Wb_1(i-1) \\ b_2(i-1) \\ y(i) - a(i)h(0) \end{bmatrix} \right\|_2 \quad (3.32)$$

$$\begin{aligned} \min_{h(i)} J(i) = & \left\| \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i) - h(i-1)] - \begin{bmatrix} Wb_1(i-1) \\ b_2(i-1) \\ y(i) - a(i)h(0) \end{bmatrix} - \right. \\ & \left. - \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i-1) - h(0)] \right\|_2 \quad (3.33) \end{aligned}$$

$$\begin{aligned} \min_{h(i)} J(i) = & \left\| \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i) - h(i-1)] - \right. \\ & \left. \begin{bmatrix} W[b_1(i-1) - R(i-1)(h(i-1) - h(0))] \\ b_2(i-1) \\ y(i) - a(i)h(0) - a(i)(h(i-1) - h(0)) \end{bmatrix} \right\|_2 \quad (3.34) \end{aligned}$$

de (3.29) temos : $R(i-1)[h(i-1) - h(0)] - b_1(i-1) = 0$, então

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} WR(i-1) \\ 0 \\ a(i) \end{bmatrix} [h(i) - h(i-1)] - \begin{bmatrix} 0 \\ b_2(i-1) \\ y(i) - a(i)h(i-1) \end{bmatrix} \right\|_2 \quad (3.35)$$

Devido à estrutura das matrizes envolvidas podemos reescrever (3.35), como sendo :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} WR(i-1) \\ a(i) \end{bmatrix} [h(i) - h(i-1)] - \begin{bmatrix} 0 \\ y(i) - a(i)h(i-1) \end{bmatrix} \right\|_2 \quad (3.36)$$

Para resolver o problema, devemos obter uma matriz ortogonal $Q(i)$ tal que

$$Q(i) \begin{bmatrix} WR(i-1) \\ a(i) \end{bmatrix} = \begin{bmatrix} R(i) \\ 0 \end{bmatrix} \quad (3.37)$$

$$Q(i) \begin{bmatrix} 0 \\ y(i) - a(i)h(i-1) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i) \end{bmatrix} \quad (3.38)$$

A matriz ortogonal $Q(i)$ simplesmente atualiza a matriz triangular $WR(i-1)$ para a inserção de uma nova linha $a(i)$. Esta matriz pode ser sempre obtida de forma eficiente e numericamente estável através de uma sequência de matrizes de rotações de Givens aplicada a $\begin{bmatrix} WR(i-1) \\ a(i) \end{bmatrix}$ e $\begin{bmatrix} 0 \\ y(i) - a(i)h(i-1) \end{bmatrix}$ ver seção A.2.

Devido à estrutura da matriz $\begin{bmatrix} R(i) \\ 0 \end{bmatrix}$ a solução do PMQPE (3.6) no instante i é dada por :

$$R(i)[h(i) - h(i-1)] = b_1(i) \quad (3.39)$$

É importante notar que para a obtenção de $b_1(i)$ e $R(i)$ não é necessário construir $Q(i)$ explicitamente, bastando aplicar a sequência de matrizes de rotação de Givens diretamente em relação a $\begin{bmatrix} WR(i-1) \\ a(i) \end{bmatrix}$ e $\begin{bmatrix} 0 \\ y(i) - a(i)h(i-1) \end{bmatrix}$.

Na tabela 3 estão resumidos os passos do algoritmo recursivo utilizando fatorações ortogonais LSQRPE para solução do PMQPE (3.6).

TABELA3 Algoritmo de Mínimos Quadrados Recursivo utilizando Fatorações Ortogonais com Ponderação exponencial dos dados LSQRPE

—Inicialização

$R(0), h(0)$

—Atualização

- Determinar $R(i), b_1(i)$ e $b_2(i)$

$$Q(i) \begin{bmatrix} WR(i-1) \\ a(i) \end{bmatrix} = \begin{bmatrix} R(i) \\ 0 \end{bmatrix} \quad (3.40)$$

$$Q(i) \begin{bmatrix} 0 \\ y(i) - a(i)h(i-1) \end{bmatrix} = \begin{bmatrix} b_1(i) \\ b_2(i) \end{bmatrix} \quad (3.41)$$

- Obter $h(i)$ da solução do sistema

$$R(i)[h(i) - h(i-1)] = b_1(i) \quad (3.42)$$

Obs : $Q(i)$ corresponde ao produto de uma sequência de matrizes de rotação de Givens (ver seção A.2). Sua construção explícita e armazenamento não é necessário.

Da tabela 3 podemos verificar que o esforço computacional em FLOPS por iteração do algoritmo LSQRPE, corresponde às operações envolvidas nas equações de (3.40) a (3.42), que são resumidos na tabela 4 :

TABELA 4 Esforço computacional LSQRPE

Equação	FLOPS
(3.40)	$\frac{5}{2}n^2 + \frac{15}{2}n$
(3.41)	n
(3.42)	$\frac{n^2}{2}$
Total	$3n^2 + \frac{17}{2}n$

Das equações (3.4) e (3.37) devido à ortogonalidade da matriz $Q(i)$:

$$\text{cond}(X(i)) = \text{cond}(R(i)) \quad (3.43)$$

o que mostra que o condicionamento original do PMQPE não é alterado ao longo das iterações.

3.1.4 Conclusões

Comparando os algoritmos LSNEPE e LSQRPE para solução do PMQPE (3.6), concluímos que :

— no aspecto de esforço computacional por iteração, ver tabelas 2 e 4, o LSNEPE requer um esforço um pouco menor que o LSQRPE para n pequenos. Para n grandes, o esforço requerido pelos dois algoritmos pode ser considerados equivalentes.

— no aspecto condicionamento numérico, ao contrário do LSNEPE, o LSQRPE mantém o condicionamento original do PMQPE, conduzindo portanto a resultados numéricos melhores. A utilização do LSQRPE é especialmente recomendada quando o PMQPE é mal condicionado.

3.2 Métodos de Mínimos Quadrados com Janela Móvel de Dados

3.2.1 Definição do Problema

Para facilitar a identificação adaptativa de sistemas com parâmetros variantes no tempo e evitar o mal condicionamento introduzido pela ponderação exponencial dos dados, podemos utilizar uma janela móvel de dados. Analogamente, definiremos o novo problema, considerando o vetor $h(i)$ com variação lenta e assumindo estacionaridade local de comprimento ld . A estimação de parâmetros $h(i)$ corresponde à solução do seguinte problema de otimização:

$$\min_{h(i)} J(i) = \left[h(i) - h(0) \right]^T A_0 \left[h(i) - h(0) \right] + \sum_{k=i-ld+1}^i [y(k) - a(i)h(i)]^2 \quad (3.44)$$

onde A_0 é uma matriz definida positiva representando o peso da estimativa a priori $h(0)$ relativamente aos dados da janela corrente.

O problema de mínimos quadrados com janela móvel de dados (PMQJM) fica da seguinte forma matricial :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R(0) \\ A(i) \end{bmatrix} (h(i) - h(0)) - \begin{bmatrix} 0 \\ \bar{b}(i) \end{bmatrix} \right\|_2 \quad (3.45)$$

onde :

$$R^T(0)R(0) = A_0$$

$$A(i) = \begin{bmatrix} a(i-k+1) \\ \vdots \\ a(i-1) \\ a(i) \\ a(i-ld+1) \\ \vdots \\ a(i-k) \end{bmatrix} \quad (3.46)$$

matriz dos dados de entrada de dimensões (ld,n)

$$\bar{b}(i) = \begin{bmatrix} y(i-k+1) - a(i-k+1)h(0) \\ \vdots \\ y(i-1) - a(i-1)h(0) \\ y(i) - a(i)h(0) \\ \vdots \\ y(i-k) - a(i-k)h(0) \end{bmatrix} \quad (3.47)$$

vetor do sinal de referência de dimensões (ld,1), sendo $k = i - \text{pint}[(i-ld)/ld]ld$ a posição de entrada/saída de dados da janela, onde $\text{pint}(\cdot)$ é a parte inteira de (\cdot) .

Da mesma forma que fizemos para o PMQPE (3.2), vamos descrever o PMQJM (3.45) recursivamente.

Das equações (3.46) e (3.47) podemos ver que $A(i)$ e $\bar{b}(i)$ podem ser escritos recursivamente da seguinte forma :

$$A(i) = A(i-1) + z(k)(a(i) - a(i-ld+1)) \quad (3.48)$$

$$\bar{b}(i) = \bar{b}(i-1) + z(k)(e(i) - e(i-ld+1)) \quad (3.49)$$

$$e(i) = y(i) - a(i)h(0) \quad (3.50)$$

$$e(i - ld + 1) = y(i - ld + 1) - a(i - ld + 1)h(0) \quad (3.51)$$

$$z(k) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.52)$$

$$k = i - \text{pint}((i - ld)/ld)ld \quad (3.53)$$

onde $z(k)$ é um vetor de dimensões $(ld,1)$ com todos seus elementos nulos exceto o da posição k que é igual a 1.

Substituindo (3.48) a (3.53) em (3.45), podemos definir o PMQJM da seguinte forma :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R(0) & 0 \\ A(i-1)+ & z(k)v(i) \end{bmatrix} (h(i) - h(0)) - \begin{bmatrix} 0 \\ \bar{b}(i-1)+ & z(k)(e(i) - e(i - ld + 1)) \end{bmatrix} \right\|_2 \quad (3.54)$$

onde $v(i) = a(i) - a(i - ld + 1)$.

3.2.2 Algoritmo Recursivo utilizando as Equações Normais

Para o PMQJM (3.54) a equação normal no instante i é dada por :

$$\left\{ \left[R^T(0) \ A^T(i-1) \right] + v^T(i) \left[0 \ z^T(k) \right] \right\} \left\{ \left[\begin{array}{c} R(0) \\ A(i-1) \end{array} \right] + \left[\begin{array}{c} 0 \\ z(k)v(i) \end{array} \right] \right\} (h(i) - h(0)) =$$

$$\left\{ \left[R^T(0) \ A^T(i-1) \right] + v^T(i) \left[0 \ z^T(k) \right] \right\} \left\{ \left[\begin{array}{c} 0 \\ \bar{b}(i-1) \end{array} \right] + \left[\begin{array}{c} 0 \\ z(k)(e(i) - e(i-ld+1)) \end{array} \right] \right\} \quad (3.55)$$

Fazendo as devidas operações em (3.55), chegamos facilmente na seguinte expressão :

$$\begin{aligned} & [X^T(i-1)X(i-1) + a^T(i)a(i) - a^T(i-ld+1)a(i-ld+1)](h(i) - h(0)) = \\ & = A^T(i-1)\bar{b}(i-1) + a^T(i)e(i) - a(i-ld+1)e(i-ld+1) \quad (3.56) \end{aligned}$$

onde $X(i-1) = \left[\begin{array}{c} R(0) \\ A(i-1) \end{array} \right]$

Para resolver recursivamente a equação (3.56), podemos dividir o problema em duas etapas :

$$[X^T(i-1)X(i-1) + a^T(i)a(i)](h'(i) - h(i-1)) = a^T(i)[y(i) - a(i)h(i-1)] \quad (3.57)$$

$$[X'^T(i)X'(i) - a^T(i-ld+1)a(i-ld+1)](h(i) - h'(i)) =$$

$$-a^T(i - ld + 1)[y(i - ld + 1) - a(i - ld + 1)h'(i)] \quad (3.58)$$

onde $X'(i) = \begin{bmatrix} X(i - 1) \\ a(i) \end{bmatrix}$

Na etapa (3.57) acima, foi utilizado o mesmo desenvolvimento do LSNEPE para adicionar os dados do instante i aos dados do instante $i-1$, gerando uma solução provisória $h'(i)$. Na etapa (3.58) foi também utilizado o desenvolvimento do LSNEPE, com as devidas mudanças de sinal para retirar de $X'(i)$ os dados do instante $i - ld + 1$, obtendo a solução final $h(i)$.

Assumindo para $i \leq 0$, $a(i) = 0$ e $y(i) = 0$. Na tabela 5, é apresentado o algoritmo LSNEJM para solução recursiva do PMQJM utilizando as equações normais (3.57) e (3.58).

TABELA 5 Algoritmo de Mínimos Quadrados utilizando Equações Normais com Janela Móvel de Dados (LSNEJM)

Inicialização

$$S(0) = A_0^{-1}$$

Atualizações

Para $i = 1$ até Final dos dados Faça :

— ganho de adaptação

$$v(i) = S(i-1)a^T(i) \quad (3.59)$$

$$g(i) = \frac{v(i)}{1 + a(i)v(i)} \quad (3.60)$$

— erro de estimação

$$e(i) = y(i) - a(i)h(i-1) \quad (3.61)$$

— parâmetros do sistema

$$h(i) = h(i-1) + g(i)e(i) \quad (3.62)$$

— matriz inversa de autocorrelação

$$S(i) = S(i-1) - g(i)v^T(i) \quad (3.63)$$

Caso $i > ld$

— ganho de adaptação

$$v(i) = S(i)a^T(i-ld) \quad (3.64)$$

$$g(i) = \frac{v(i)}{1 - a(i - ld)v(i)} \quad (3.65)$$

— erro de estimação

$$e(i) = y(i - ld) - a(i - d)h(i - 1) \quad (3.66)$$

— parâmetros do sistema

$$h(i) = h(i) - g(i)e(i) \quad (3.67)$$

— matriz inversa de autocorrelação

$$S(i) = S(i) + g(i)v^T(i) \quad (3.68)$$

O esforço computacional por iteração requerido pelo LSNEJM é devido às equações de (3.59) a (3.68), estando resumido na tabela 6.

TABELA 6 Esforço computacional LSNEJM

Equação	FLOPS
(3.59) e (3.64)	$2n^2$
(3.60) e (3.65)	$4n + 2$
(3.61) e (3.66)	$2n$
(3.62) e (3.67)	$2n$
(3.63) e (3.68)	$2n^2$
TOTAL	$4n^2 + 8n + 2$

Analogamente ao LSNEPE, podemos ver nas equações (3.57),(3.58) que o condicionamento original do problema é elevado ao quadrado. O uso do Lema de Inversão de Matrizes faz com que os erros devidos ao mal condicionamento inicial do problema não sejam recuperados ao longo das iterações [5].

3.2.3 Algoritmo Recursivo Proposto baseado em Fatorações Ortogonais

Para o PMQJM (3.45), num dado instante $i-1$, assumindo a aplicação de uma transformação ortogonal $Q(i-1)$, temos :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} [h(i-1) - h(0)] - \begin{bmatrix} b_1(i-1) \\ b_2(i-1) \end{bmatrix} \right\|_2 \quad (3.69)$$

onde :

$$Q(i-1) \begin{bmatrix} R(0) \\ A(i-1) \end{bmatrix} = \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} \quad (3.70)$$

$$Q(i-1) \begin{bmatrix} X(i-1) \\ \bar{b}(i-1) \end{bmatrix} = \begin{bmatrix} b_1(i-1) \\ b_2(i-1) \end{bmatrix} \quad (3.71)$$

Nessas condições, a solução $h(i-1)$ é dada pela solução do sistema de equações

:

$$R(i-1) - [h(i-1) - h(0)] = b_1(i-1) \quad (3.72)$$

No instante i , temos :

$$\begin{aligned} \min_{h(i)} J(i) = & \left\| \left\{ \begin{bmatrix} R(0) \\ A(i-1) \end{bmatrix} + \begin{bmatrix} 0 \\ z(k)v(i) \end{bmatrix} \right\} [h(i) - h(0)] - \right. \\ & \left. - \begin{bmatrix} 0 \\ \bar{b}(i-1) \end{bmatrix} + \begin{bmatrix} X(i-1) \\ z(k)[e(i) - e(i-ld+1)] \end{bmatrix} \right\| \quad (3.73) \end{aligned}$$

onde :

$$v(i) = a(i) - a(i-ld+1)$$

Aplicando em (3.73) a transformação ortogonal $Q(i-1)$, temos :

$$\begin{aligned} \min_{h(i)} J(i) = & \left\| \left\{ Q(i-1) \begin{bmatrix} R(0) \\ A(i-1) \end{bmatrix} + Q(i-1) \begin{bmatrix} 0 \\ z(k)v(i) \end{bmatrix} \right\} [h(i) - h(0)] - \right. \\ & \left. - Q(i-1) \begin{bmatrix} 0 \\ \bar{b}(i-1) \end{bmatrix} + Q(i-1) \begin{bmatrix} X(i-1) \\ z(k)[e(i) - e(i-ld+1)] \end{bmatrix} \right\|_2 \end{aligned}$$

Substituindo (3.70) e (3.71) na equação acima, temos :

$$\min_{h(i)} J(i) = \left\| \left\{ \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} + q_k(i-1)v(i) \right\} [h(i) - h(0)] - \begin{bmatrix} b_1(i-1) \\ b_2(i-1) \end{bmatrix} + q_k(i-1)[e(i) - e(i-ld+1)] \right\|_2 \quad (3.74)$$

Assumindo

$$q_k(i-1) = \begin{bmatrix} ql_k \\ qs_k \end{bmatrix}, Q' = \begin{bmatrix} I_n & \\ & L \end{bmatrix} \text{ e } Lqs_k = \begin{bmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

onde ql_k corresponde às n primeiras linhas de $q_k(i-1)$ e qs_k às ld linhas restantes de $q_k(i-1)$ e L é uma matriz ortogonal de Householder que aplicada a qs_k anula todas as suas linhas exceto a primeira (ver seção A1).

Aplicando Q' em (3.74), temos :

$$\min_{h(i)} J(i) = \left\| \left\{ \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} + \begin{bmatrix} ql_k(i-1) \\ \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} v(i) \right\} [h(i) - h(0)] - \begin{bmatrix} b_1(i-1) \\ \tilde{b}_2(i-1) \end{bmatrix} + \begin{bmatrix} ql_k(i-1) \\ \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} [e(i) - e(i-ld+1)] \right\|_2 \quad (3.75)$$

onde $\tilde{b}_2(i-1) = Lb_2(i-1)$

Assumindo M uma matriz ortogonal de dimensões $(n+1, n+1)$ obtida de uma seqüência de rotações de Givens (ver seção A.5) tal que :

$$M \left\{ \begin{bmatrix} R(i-1) \\ 0 \end{bmatrix} + \begin{bmatrix} ql_k(i-1) \\ \beta \end{bmatrix} v(i) \right\} = \begin{bmatrix} R(i) \\ 0 \end{bmatrix} \quad (3.76)$$

$$Q'' = \begin{bmatrix} M & \\ & I_{ld-1} \end{bmatrix} \quad (3.77)$$

Aplicando Q'' em (3.75), temos :

$$\min_{h(i)} J(i) = \left\| \begin{bmatrix} R(i) \\ 0 \end{bmatrix} [h(i) - h(0)] - \begin{bmatrix} b_1(i) \\ b_2(i) \end{bmatrix} \right\|_2 \quad (3.78)$$

onde

$$\begin{bmatrix} b_1(i) \\ b_2(i) \end{bmatrix} = Q'' \left\{ \begin{bmatrix} b_1(i-1) \\ \tilde{b}_2(i-1) \end{bmatrix} + \begin{bmatrix} ql_k(i-1) \\ \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} [e(i) - e(i-ld+1)] \right\}$$

Desta forma, a solução $h(i)$ é dada pela solução do sistema de equações

$$R(i)[h(i) - h(0)] = b_1(i) \quad (3.79)$$

Dos passos acima, é fácil verificar que a transformação resultante $Q(i)$, aplicada a $J(i)$ para colocação de $\begin{bmatrix} R(0) \\ A(i) \end{bmatrix}$ na forma triangular superior $\begin{bmatrix} R(i) \\ 0 \end{bmatrix}$ é dada

por :

$$Q(i) = Q''Q'Q(i-1) \quad (3.80)$$

A ortogonalidade de $Q(i-1)$, Q' e Q'' garantem a ortogonalidade de $Q(i)$. Baseado nos resultados acima e racionalizando os cálculos matemáticos propomos o LSQRJM, um algoritmo recursivo utilizando fatorações ortogonais com janela móvel de dados, para solução do PMQJM (3.54) descrito na tabela (7) a seguir.

TABELA 7 Algoritmo proposto de Mínimos Quadrados Recursivo utilizando Fatorações Ortogonais com Janela Móvel de Dados (LSQRJM)

— Inicialização

- Determinar $R(0)$ através da decomposição de Cholesky

$$R^T(0)R(0) = A_0$$

- Fazer $Q(0) = \begin{bmatrix} 0 \\ I_{ld} \end{bmatrix}$

$$ya(0) = 0, \quad a(i) = 0 \text{ e } y(i) = 0 \text{ para } i \leq 0$$

onde : $Q(0)$ é uma matriz de dimensões $(n+ld, ld)$ e $ya(0)$ é um vetor de dimensões $(ld, 1)$.

— Atualizações

Para $i = 1$ até o final dos dados, faça :

- Assumir a partição

$$\begin{bmatrix} Q_s \\ Q_l \end{bmatrix} = Q(i-1) \quad (3.81)$$

onde : Q_s corresponde às n primeiras linhas de $Q(i-1)$ e Q_l às ld últimas.

- Determinar

$$k = i - \text{pint}((i - ld)/ld)ld \quad (3.82)$$

$$\begin{bmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} = Lq_l k \quad (3.83)$$

$$\tilde{Q} = \begin{bmatrix} \tilde{Q}_s \\ \tilde{Q}_l \end{bmatrix} = \begin{bmatrix} I_n & \\ & L \end{bmatrix} \begin{bmatrix} Q_s \\ Q_l \end{bmatrix} \quad (3.84)$$

onde : \tilde{q}_{lk} é a k -ésima coluna da matriz Q_l , β é um escalar e L é uma matriz ortogonal de Householder de dimensões (ld,ld) .

- Fazer

$$v = a(i) - a(i - ld) \quad (3.85)$$

$$t = \begin{bmatrix} \tilde{q}_{s_k} \\ \beta \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \tilde{q}_{l_1 y a(i-1)} \end{bmatrix} \quad (3.86)$$

$$\hat{R} = \begin{bmatrix} R^{(i-1)} \\ 0 \end{bmatrix} + tv \quad (3.87)$$

$$dy = y(i) - y(i - ld) - vh(0) \quad (3.88)$$

$$\tilde{y}(k) = z(k)dy, \quad ya(i) = ya(i-1) + \tilde{y}(k) \quad (3.89)$$

onde : \tilde{q}_{sk} é a k-ésima coluna da matriz \tilde{Q}_s , b é um vetor de dimensões $(n+1, 1)$, \tilde{q}_{l_1} é a primeira linha da matriz \tilde{Q}_l

- Determinar

$$\begin{bmatrix} R(i) \\ 0 \end{bmatrix} = M\tilde{R} \quad (3.90)$$

$$Q(i) = \begin{bmatrix} Q_s \\ Q_l \end{bmatrix} = \begin{bmatrix} M & \\ & I_{ld-1} \end{bmatrix} \tilde{Q} \quad (3.91)$$

$$\tilde{b} = \begin{bmatrix} \tilde{b}_s \\ \tilde{b}_l \end{bmatrix} = Mb \quad (3.92)$$

onde M é uma matriz ortogonal de dimensões $(n+1, n+1)$ correspondente ao produto de uma sequência de matrizes de rotação de Givens que aplicadas à matriz \tilde{R} a colocam na forma triangular superior indicada em (3.90) (ver seção A5).

- Determinar parâmetros

$$R(i)\tilde{h} = \tilde{b}_s + dyq_{sk} \quad (3.93)$$

$$h(i) = h(0) + \tilde{h}$$

onde q_{sk} é a k-ésima coluna da matriz Q_s .

Da tabela 7, podemos ver que a complexidade computacional por iteração do algoritmo é devida majoritariamente às operações feitas em (3.83),(3.84),(3.90),(3.91),(3.92)

e (3.93), que corresponderá à seguinte estimativa de operações de ponto flutuante FLOPS, dada na tabela 8.

TABELA 8 Esforço computacional LSNEJM

Equações	FLOPS
(3.83) e (3.84)	$ld(2ld + 1)$
(3.90), (3.91) e (3.92)	$10n + 2n(4ld + 4n + 8)$
(3.93)	$\frac{n^2}{2} + n$
Total	$2ld^2 + \frac{17}{2}n^2 + 8nld + 17n + ld$

Devido à ortogonalidade de todas as transformações aplicadas a vetores e matrizes do LSQRJM, fica então garantido que o condicionamento original do PMQJM não é alterado.

Outro fator importante a destacar é que para a realização das transformações e solução de sistemas triangulares de equações, foram utilizados algoritmos de reconhecida estabilidade numérica, o que garante a estabilidade numérica global do algoritmo.

3.2.4 Filtragem dos Parâmetros

Em problemas mal condicionados, as pequenas variações de dados entre janelas consecutivas causam variações bruscas nos parâmetros ótimos.

Para contornar esse problema propomos a aplicação do seguinte filtro "passa baixa" aos parâmetros ótimos calculados

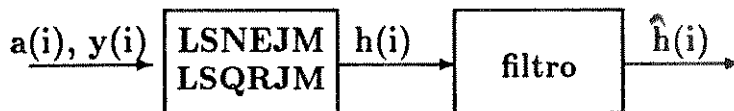
$$\hat{h}(0) = h(0)$$

$$\hat{h}(i) = (1 - \alpha)\hat{h}(i - 1) + \alpha h(i) \tag{3.94}$$

onde $\hat{h}(\cdot)$ são os parâmetros filtrados e α é um escalar $0 \leq \alpha \leq 1$.

O parâmetro α deve ser especificado em função da disparidade do espectro de frequências das variações dos $h(i)$ calculados e a lentidão dos parâmetros do sistema identificado, procurando atenuar ao máximo a amplitude das variações de parâmetros, introduzindo um mínimo de defasagem dos mesmos.

Em diagrama de blocos, o processo de estimação de parâmetros com janela móvel de dados e filtragem adaptativa corresponde ao seguinte :



3.2.5 Conclusões

Comparando os algoritmos de janela móvel LSNEJM e o proposto LSQRJM, podemos ver que :

— a principal vantagem do algoritmo baseado nas equações normais LSNEJM é o seu baixo custo computacional por iteração.

— a principal vantagem do algoritmo baseado em transformações ortogonais LSQRJM e a razão mesmo da sua existência é o seu condicionamento numérico que é o mesmo do problema de mínimos quadrados original. Seu custo computacional é bastante alto quando comparado com o LSNEJM (ver tabela 6 e tabela 8), fazendo que o LSQRJM seja especialmente recomendado para problemas inerentemente mal condicionados.

Capítulo 4

Exemplos Numéricos

Para ilustrar o desempenho dos algoritmos LSNEPE, LSNEJM, LSQRPE e LSQRJM proposto, os quatro algoritmos são utilizados para identificar os parâmetros h do seguinte modelo ARMA :

$$x(k+4) = h_1x(k+3) + h_2x(k+2) + h_3x(k+1) + h_4x(k) + h_5u(k+1) + h_6u(k) \quad (4.1)$$

Os dados para a identificação são obtidos da simulação de um sistema compatível, com a seguinte função de transferência simbólica :

$$\frac{x(k)}{u(k)} = \frac{(q - p_5)}{(q - p_1)(q - p_2)(q - p_3)(q - p_4)} \quad (4.2)$$

onde q é o operador deslocamento unitário ($qx(k) = x(k+1)$).

Na função de transferência (4.2), é importante notar que a medida que o zero p_5 se aproxime de qualquer um dos polos p_1 a p_4 , a observabilidade do sistema é prejudicada, fazendo com que o mesmo, do ponto de vista entrada/saída aparente ser de ordem menor, fazendo com que a matriz de informações A de (4.1) se torne quase singular, o que naturalmente leva a problemas de mínimos quadrados muito mal condicionados.

Considerando (4.1) e (4.2), a seguir serão apresentados e discutidos dois exemplos numéricos envolvendo os algoritmos LSNEPE, LSNEJM, LSQRPE e LSQRJM proposto.

4.1 Exemplo numérico 1

Neste exemplo são utilizados dados gerados por (4.2) com p_i variantes no tempo conforme indicado na figura 1. Através dela pode ser verificado que p_5 ao longo do tempo se aproxima de p_4 e p_1 , fazendo que o problema de mínimos quadrados que inicialmente é bem condicionado, se torne progressivamente muito mal condicionado.

Nas figuras seguintes desta seção, são apresentados os valores dos parâmetros h estimados pelos algoritmos LSNEPE, LSQRPE, LSNEJM e LSQRJM, juntamente com os valores exatos desses parâmetros utilizados pelo simulador.

Os resultados correspondentes ao LSNEPE estão apresentados nas figuras de 2a até 4b. O melhor resultado obtido com esse algoritmo está apresentado na figura 2b correspondente a $W = 0.9$ e $A_0 = 10-4I_6$. Este resultado não é bom porque o valor relativamente alto de A_0 impediu que os parâmetros estimados acompanhassem a variação dos parâmetros do sistema. Além disso, a partir do instante $k = 480$ o algoritmo perdeu completamente a estabilidade numérica, devido ao agravamento progressivo do condicionamento do problema de mínimos quadrados. Para valores menores de A_0 , a perda de estabilidade ocorre mais rapidamente.

Os resultados correspondentes ao LSQRPE estão apresentados nas figuras de 5a até 7b. O melhor resultado obtido com esse algoritmo está apresentado na figura 7a correspondente a $W = 0.75$ e $A_0 = 10-8I_6$. O valor de $W = 0.75$ permitiu que os parâmetros estimados seguissem melhor a variação dos parâmetros do sistema em relação aos obtidos com $W = 0.9$. Devido à utilização da fatoração ortogonal QR não ocorreu perda de estabilidade numérica, mesmo considerando valores de $A_0 = 10-8I_6$. Os resultados no entanto ainda não são satisfatórios devido à sua grande variação em torno dos valores dos parâmetros do sistema.

Os resultados correspondentes ao LSNEJM estão apresentados nas figuras de 8a até 10b. O melhor resultado obtido com esse algoritmo está apresentado na figura

9b correspondente a $ld = 9$ e $A_0 = 10-6I_6$ e $\alpha = 0.05$. O valor de $ld = 9$ permitiu que os parâmetros estimados acompanhassem melhor a variação dos parâmetros do sistema em relação aos obtidos com $ld = 15$. Podemos ver nas figuras 2b e 7a que os resultados obtidos com o LSNEJM apresentados na figura 9b são bem superiores aos LSNEPE e LSQRPE.

Estes resultados não são ainda satisfatórios devido à polarização dos mesmos relacionados com o peso de $A_0 = 10-6I_6$ relativamente à pouca influência dos dados da janela corrente, em função do mal condicionamento do problema. Podemos ver ainda na figura 10b, que não é possível diminuir o valor de A_0 sem perder a estabilidade numérica dos resultados.

Os resultados correspondentes ao LSQRJM estão apresentados nas figuras de 11a até 13b. O melhor resultado obtido com esse algoritmo está apresentado na figura 13b correspondente a $ld = 9$ e $A_0 = 10-8I_6$ e $\alpha = 0.05$. O valor de $ld = 9$ permitiu que os parâmetros estimados acompanhassem melhor a variação dos parâmetros do sistema em relação aos obtidos com $ld = 15$. Este algoritmo foi o que apresentou os melhores resultados, seguindo bem a variação dos parâmetros ao longo de todo o intervalo, sem perda de estabilidade numérica e sem polarização dos resultados. Esse bom desempenho foi devido ao uso conjunto da fatoração ortogonal QR e janela móvel de dados, que conforme mostrado no capítulo 3 fazem com que o LSQRJM mantenha o condicionamento original do problema.

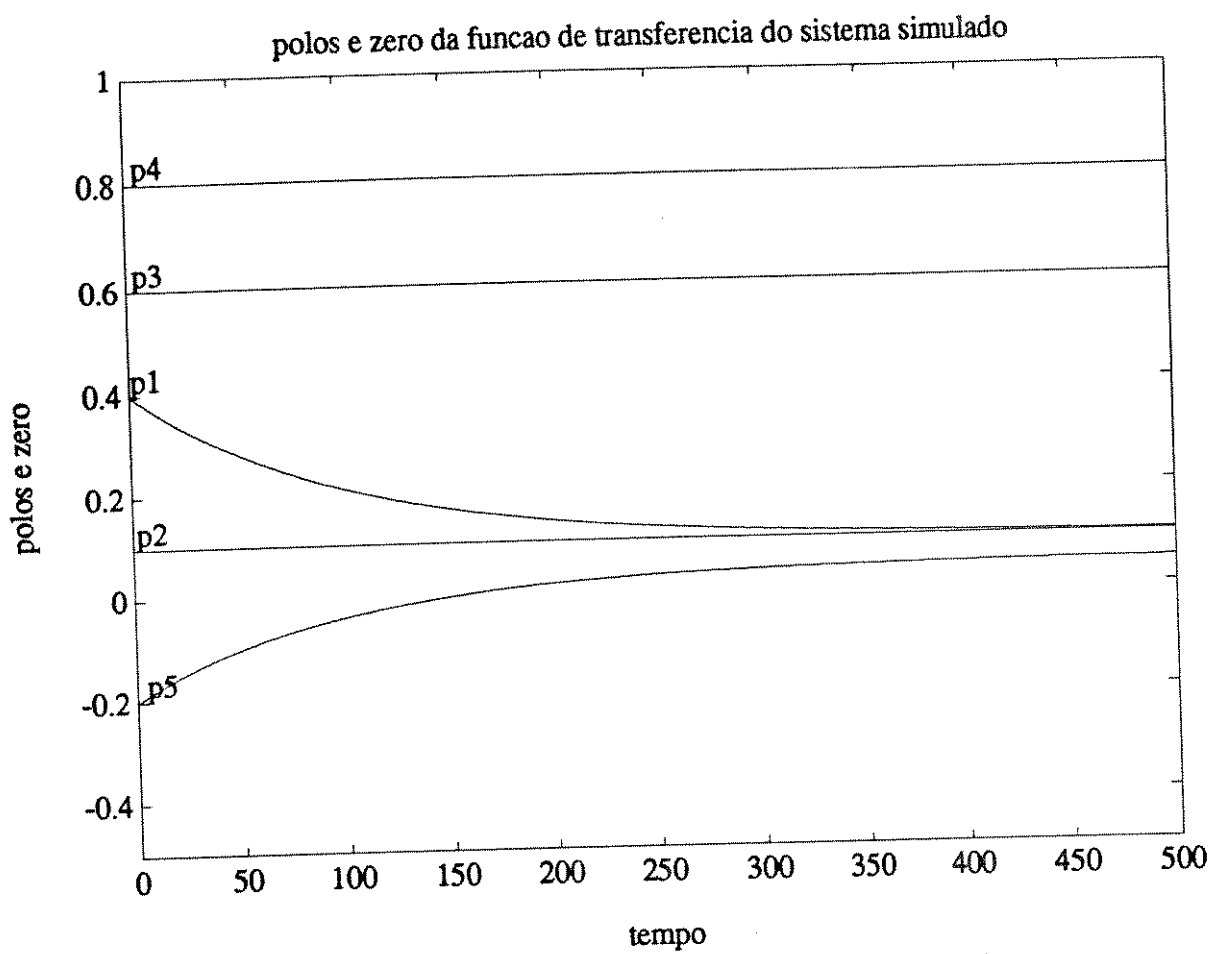
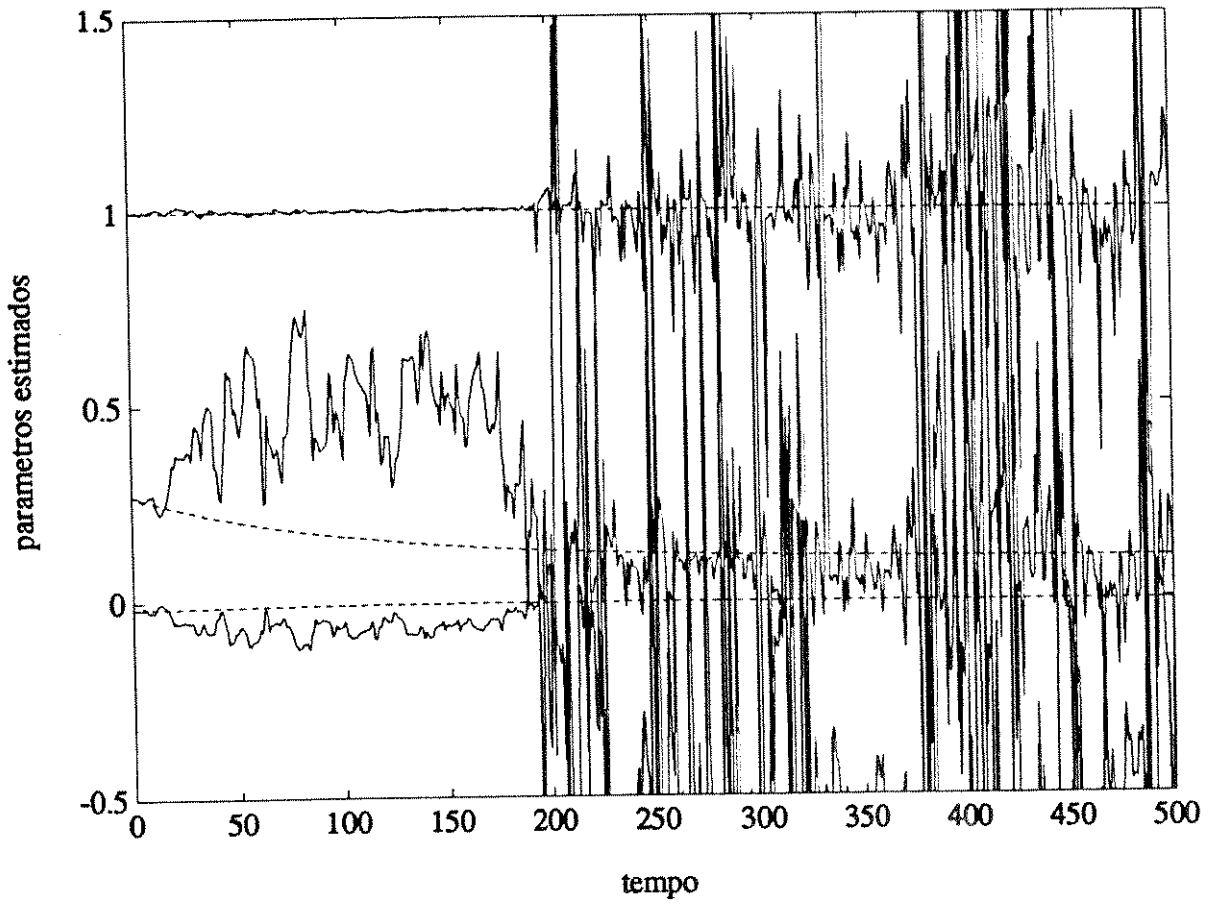


Figura 1 : Exemplo Numérico 1 - polos e zero da função de transferência do sistema simulado

LSNEPE para $A_0 = 10^{-4} \cdot I_6$ e $W = 0.75$



LSNEPE para $A_0 = 10^{-4} \cdot I_6$ e $W = 0.75$

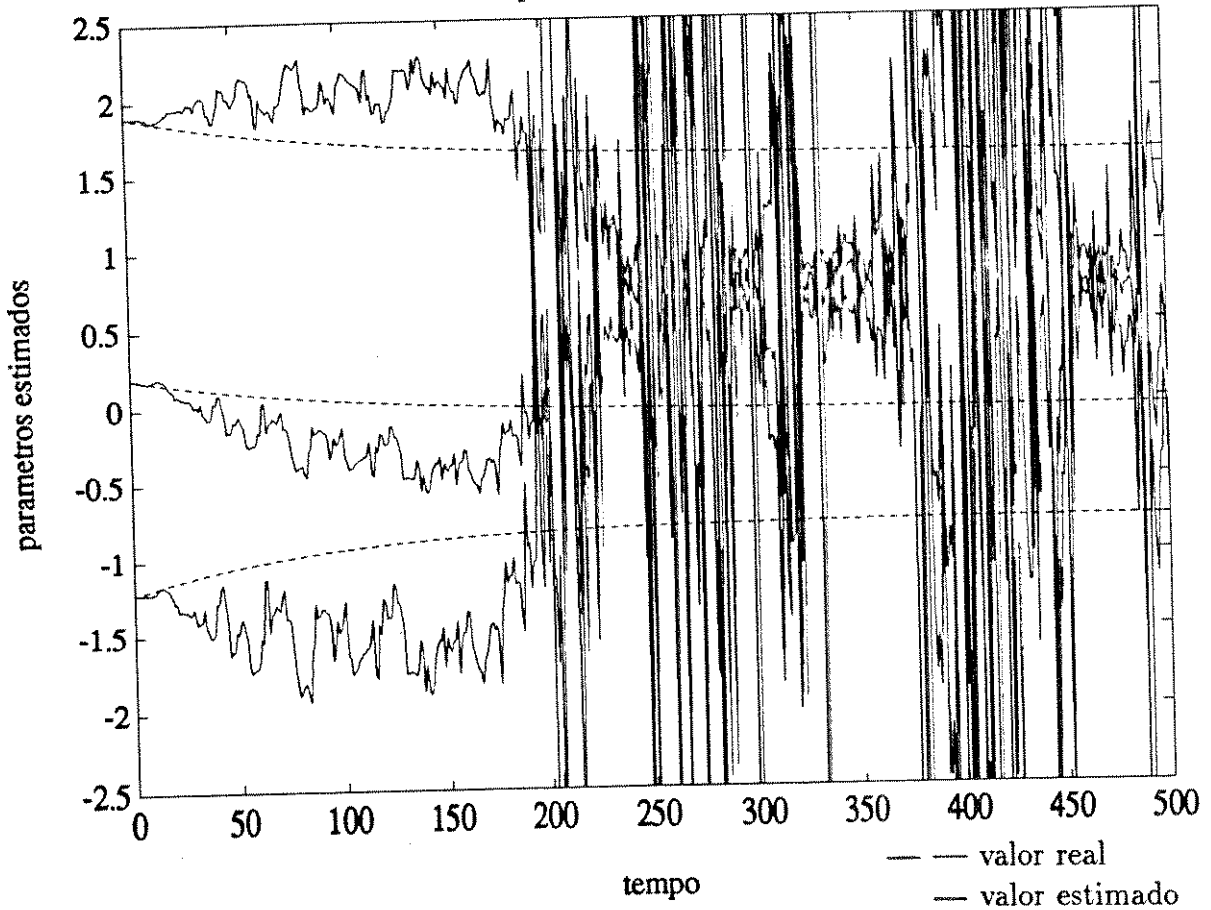
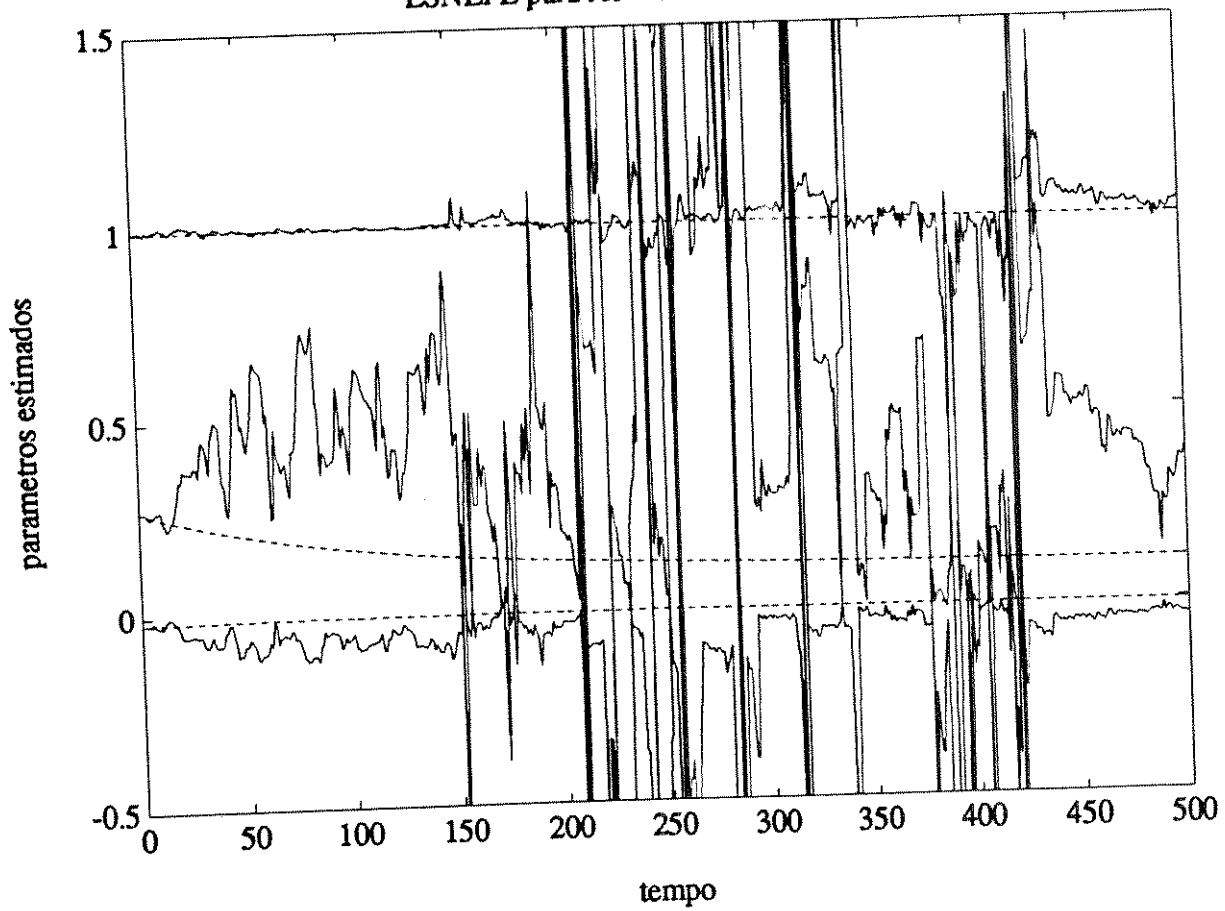


Figura 2a : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.75$



LSNEPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.75$

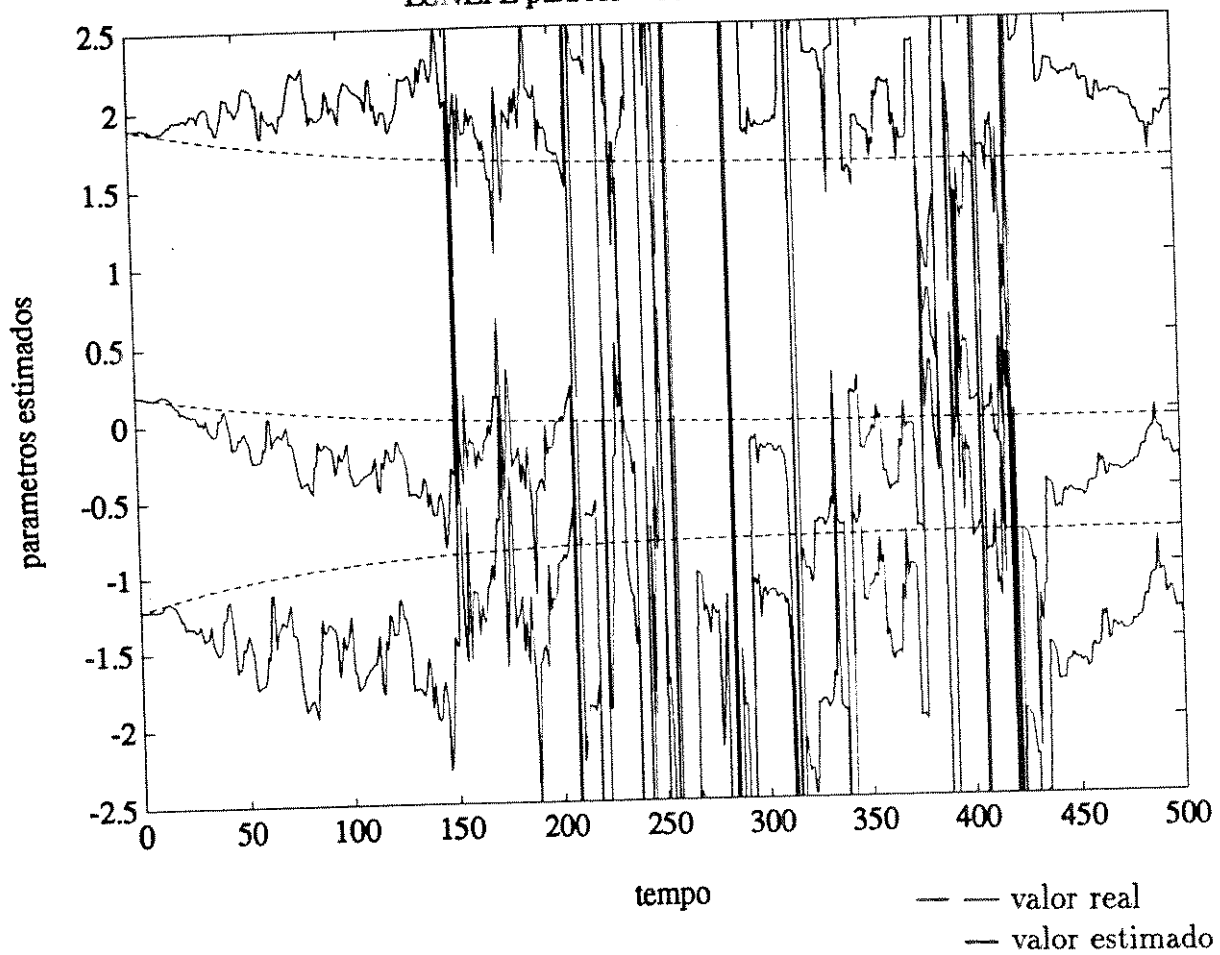
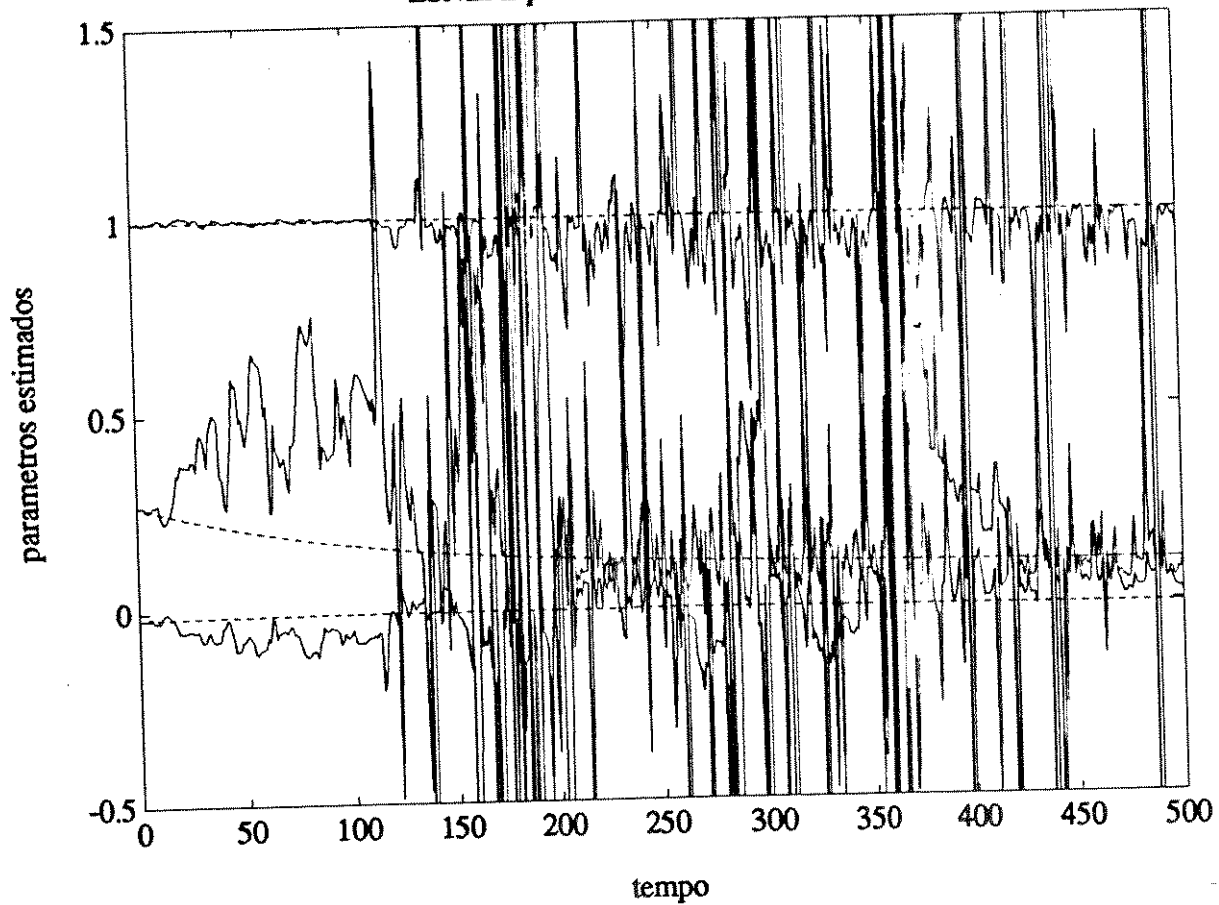


Figura 3a : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEPE para $A_0 = 10-8.16$ e $W = 0.75$



LSNEPE para $A_0 = 10-8.16$ e $W = 0.75$

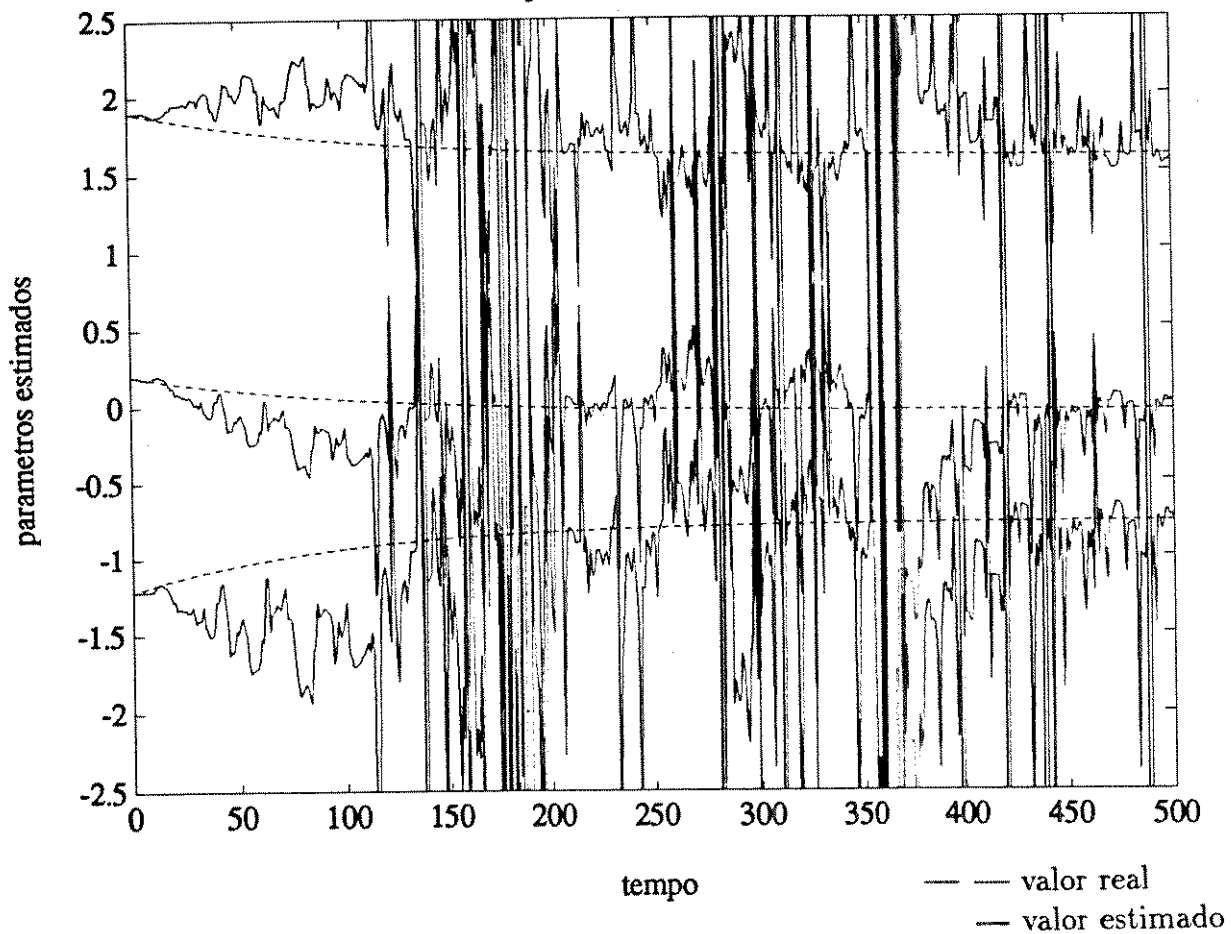
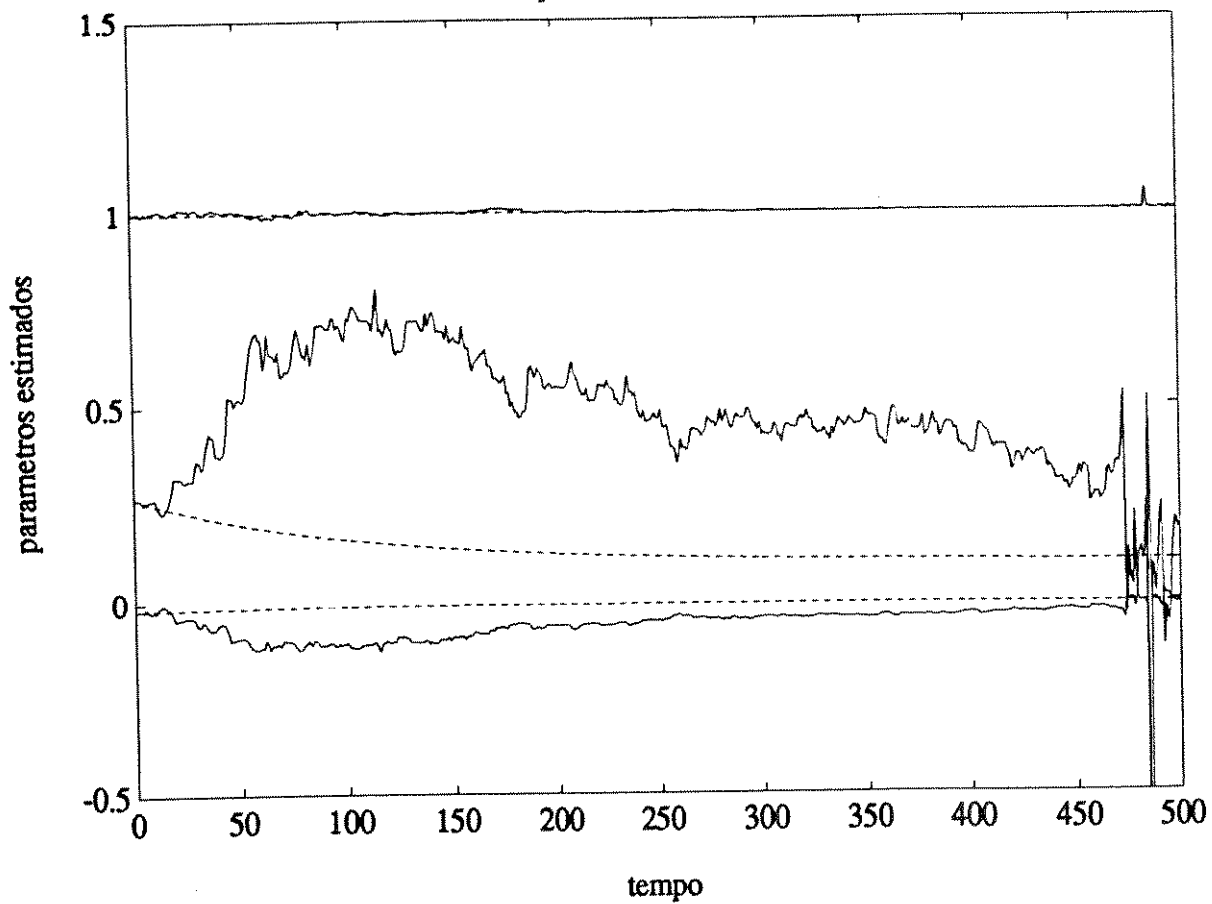


Figura 4a : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEPE para $Ao = 10^{-4}I6$ e $W = 0.9$



LSNEPE para $Ao = 10^{-4}I6$ e $W = 0.9$

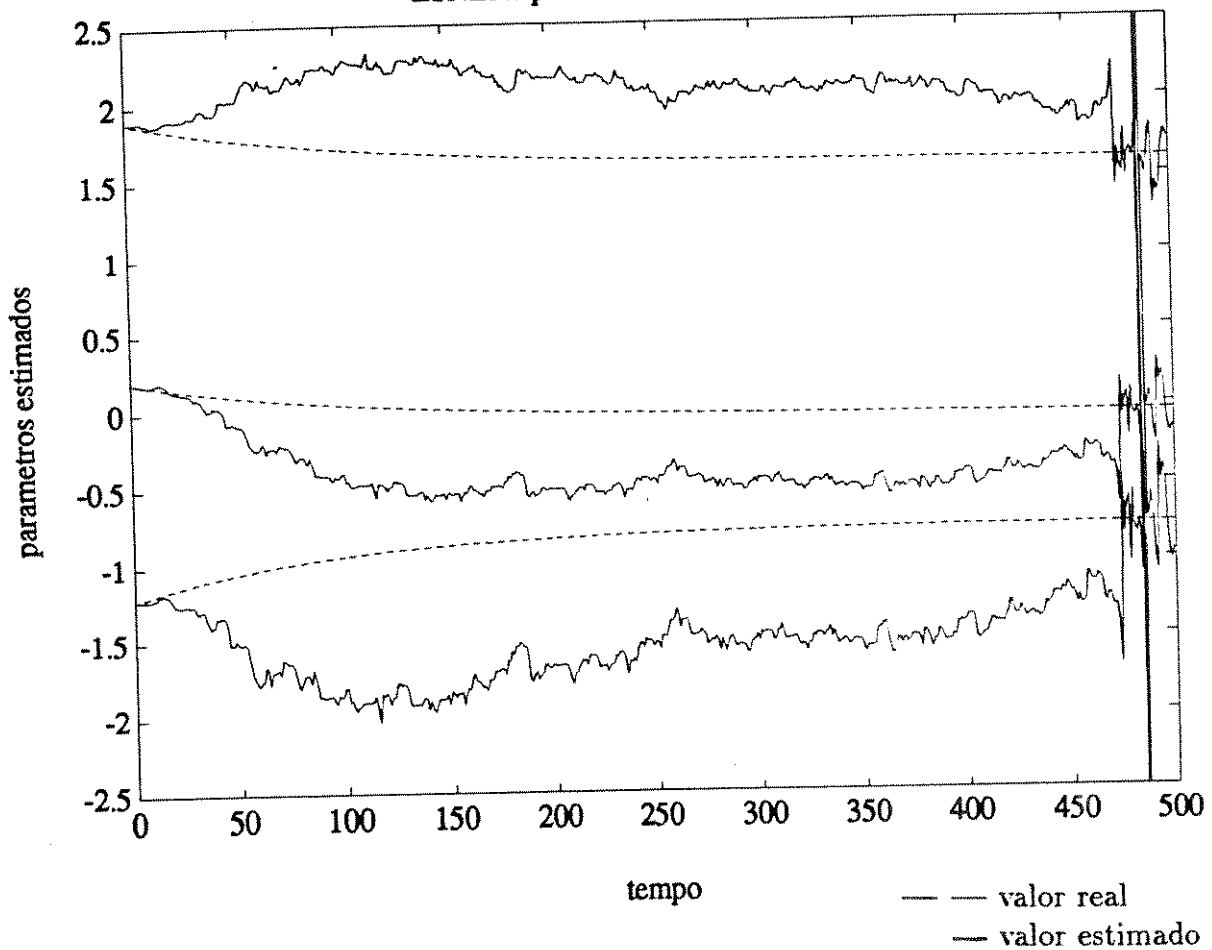
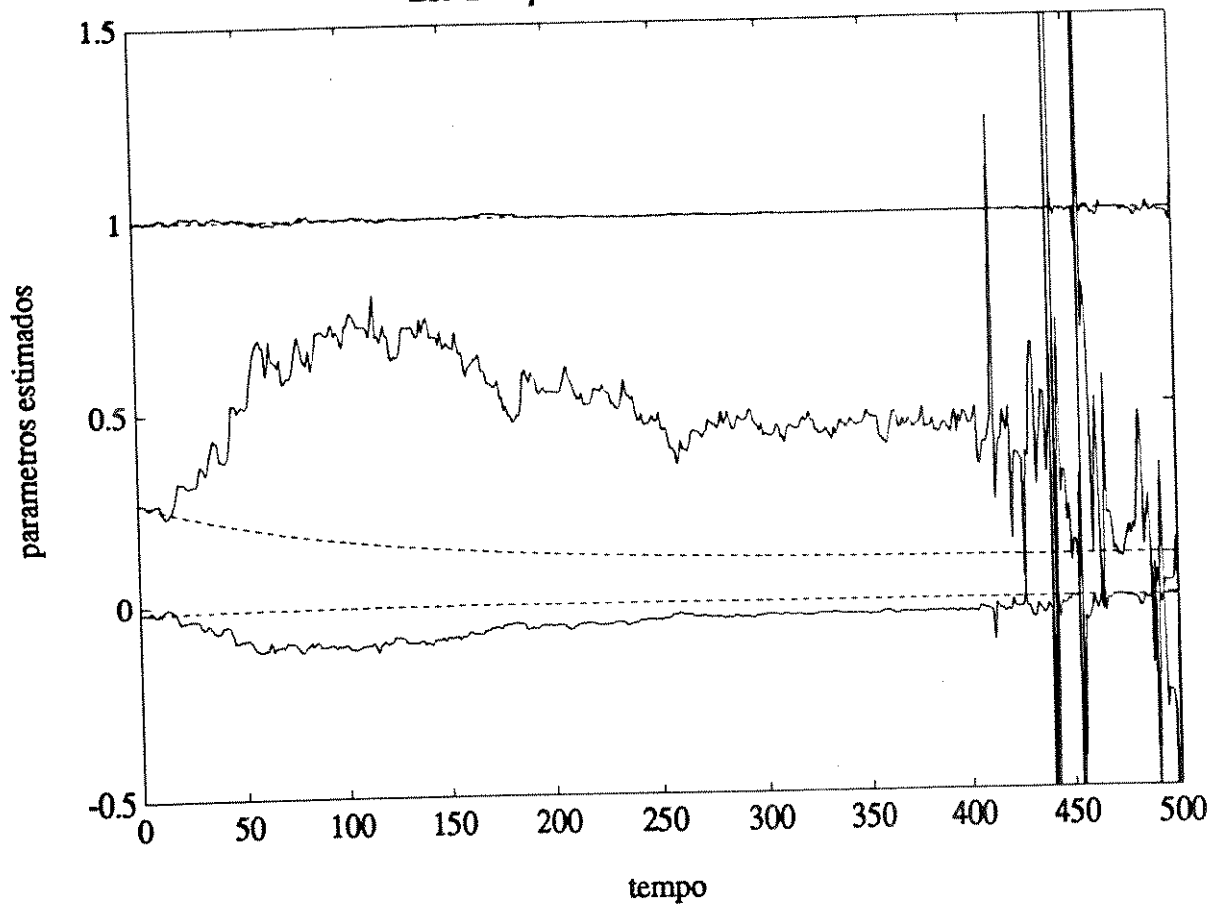


Figura 2b : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEPE para $Ao = 10^{-6}$ e $W = 0.9$



LSNEPE para $Ao = 10^{-6}$ e $W = 0.9$

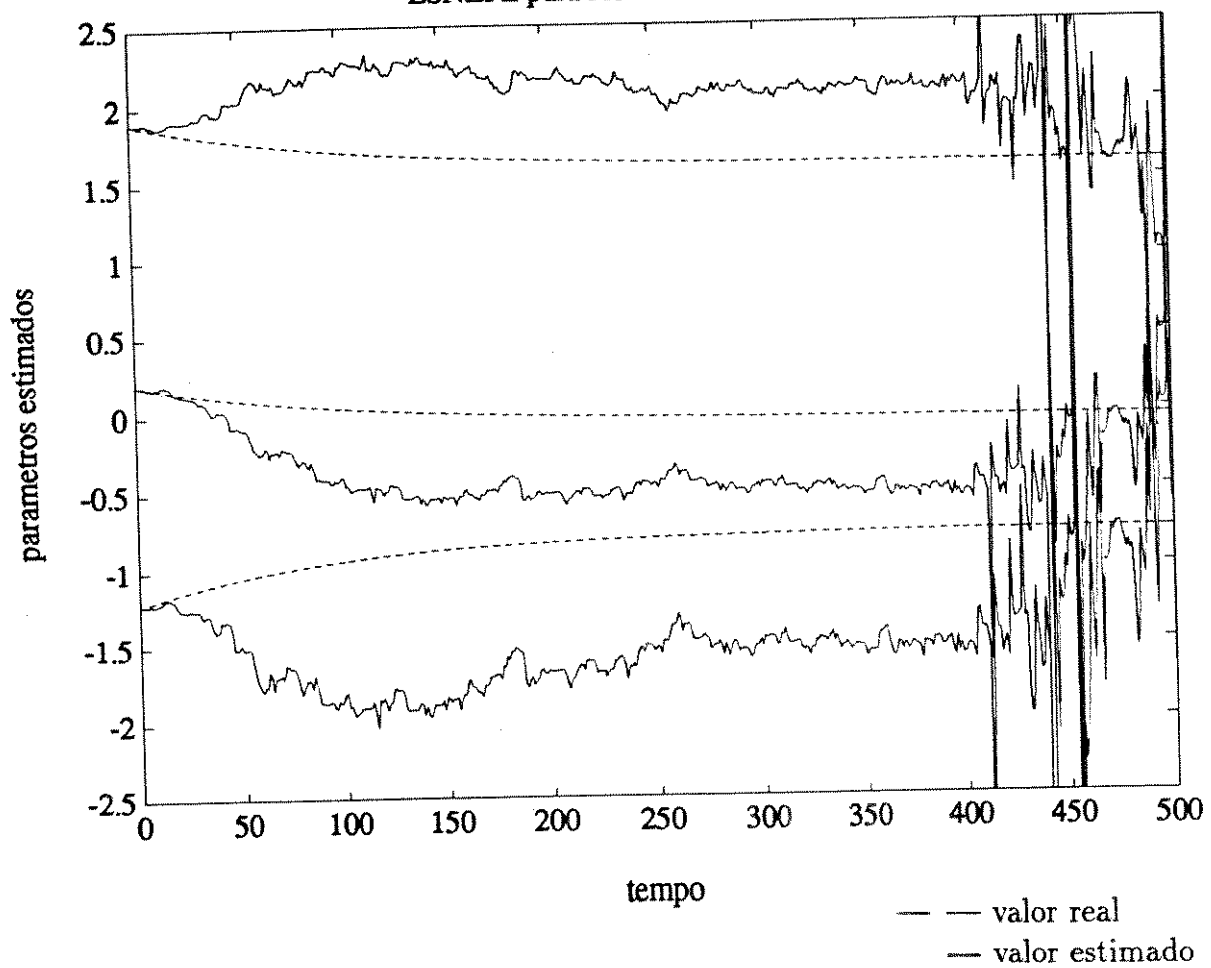
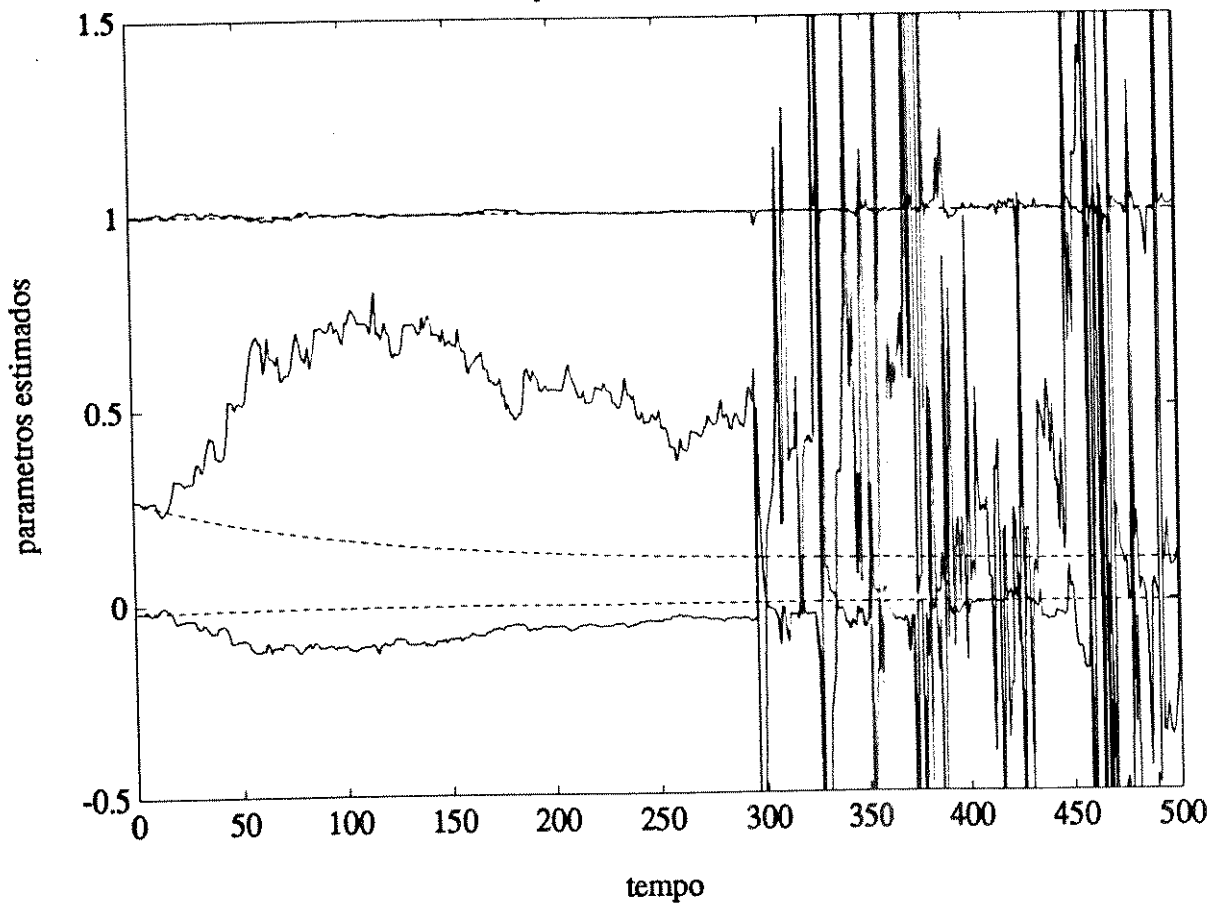


Figura 3b : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEPE para $Ao = 10^{-8.16}$ e $W = 0.9$



LSNEPE para $Ao = 10^{-8.16}$ e $W = 0.9$

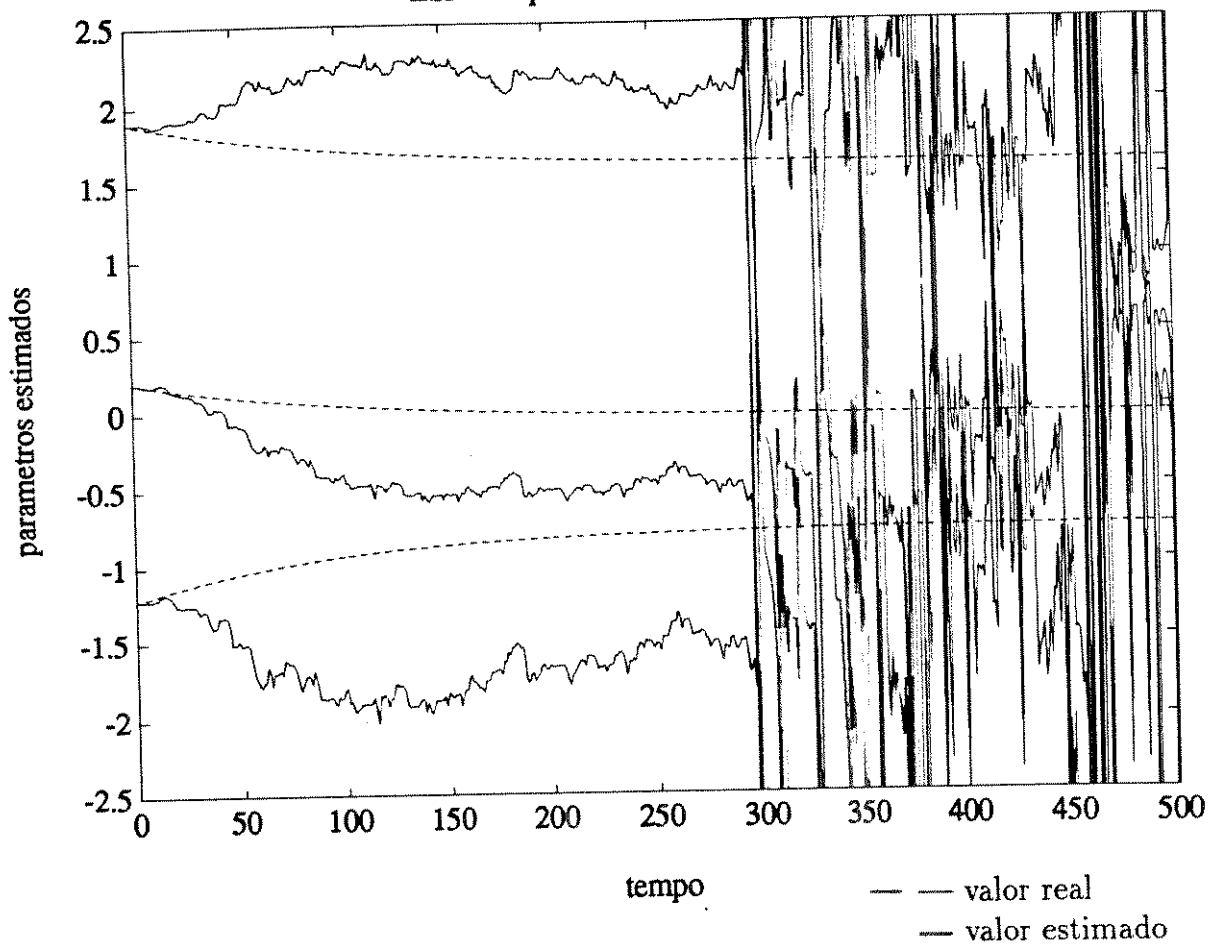
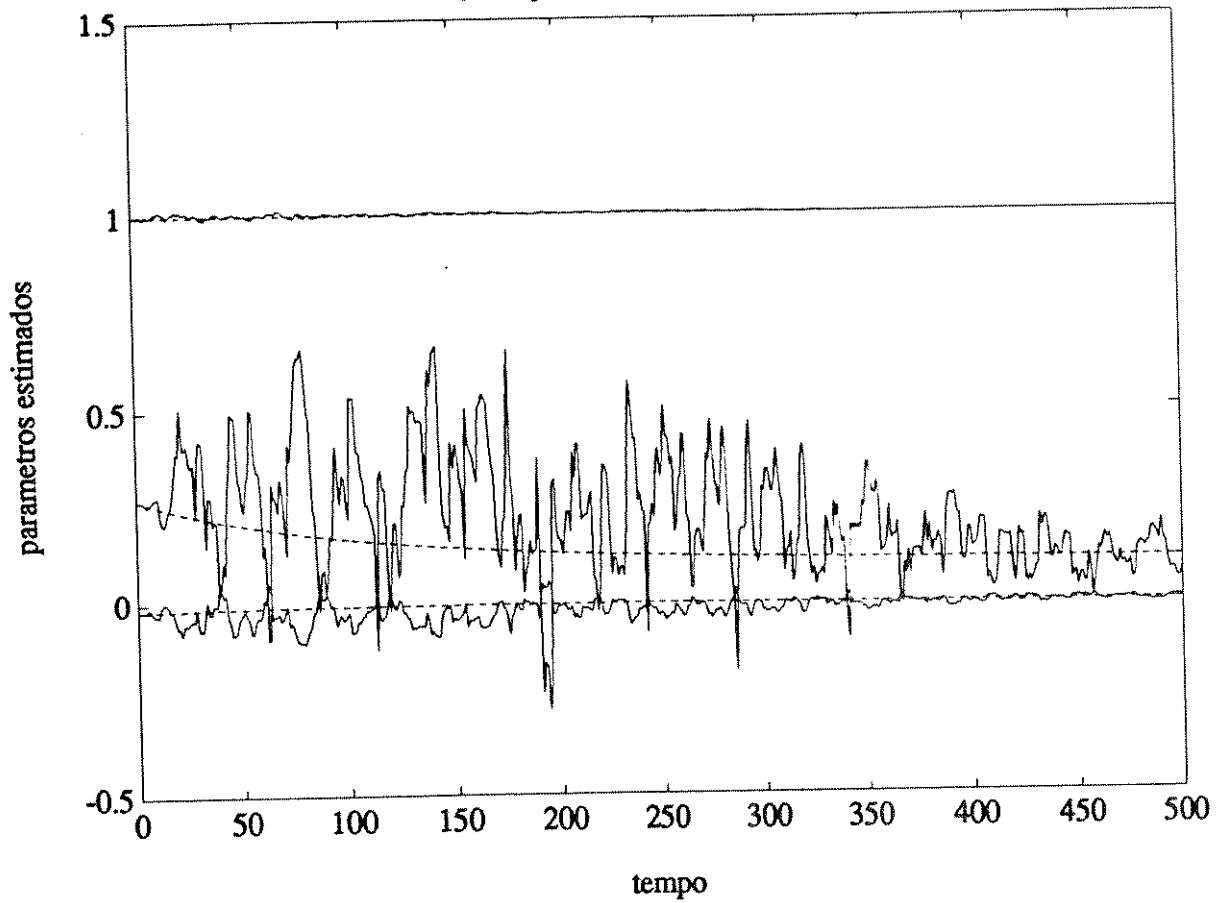


Figura 4b : Exemplo Numérico 1 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-4} \cdot I_6$ e $W = 0.75$



LSQRPE para $A_0 = 10^{-4} \cdot I_6$ e $W = 0.75$

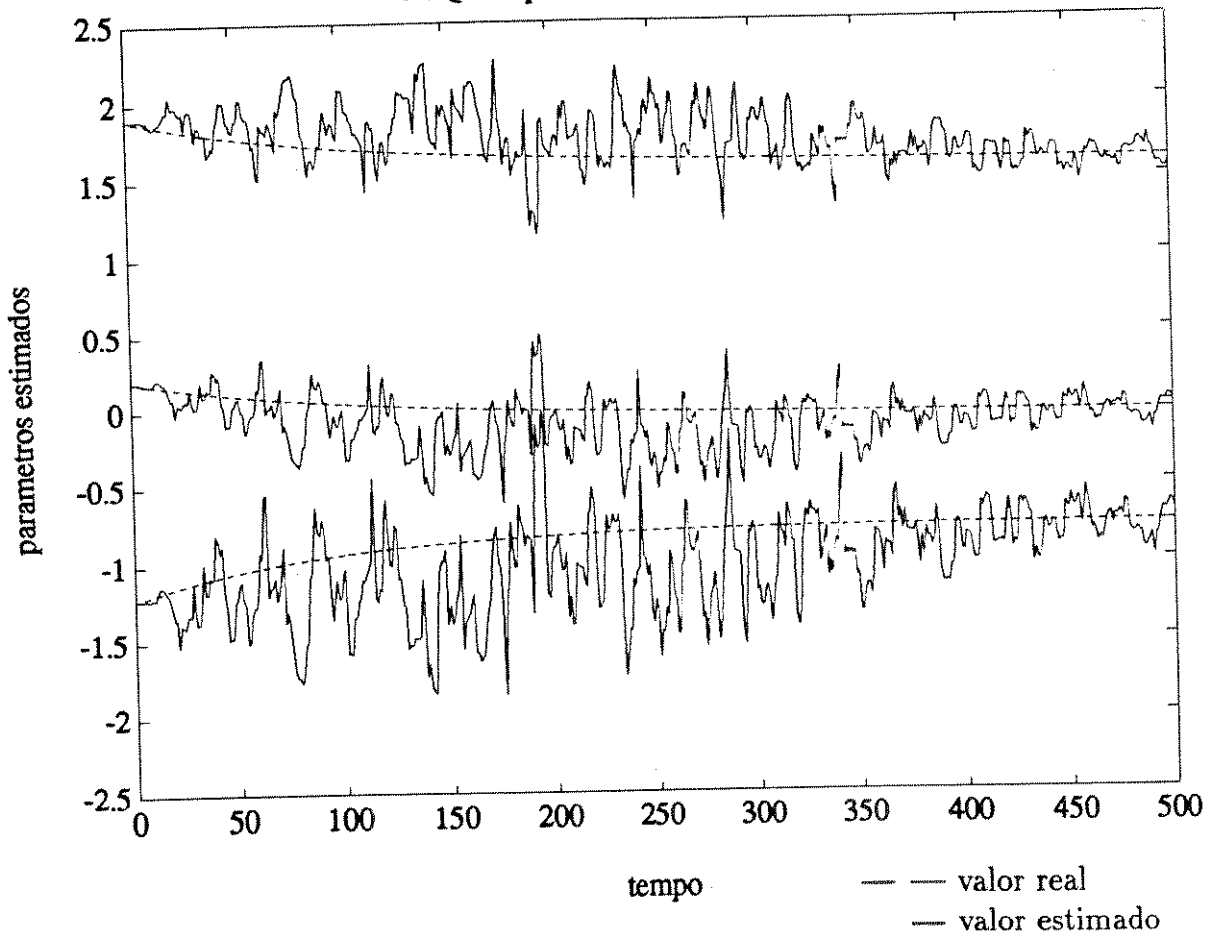
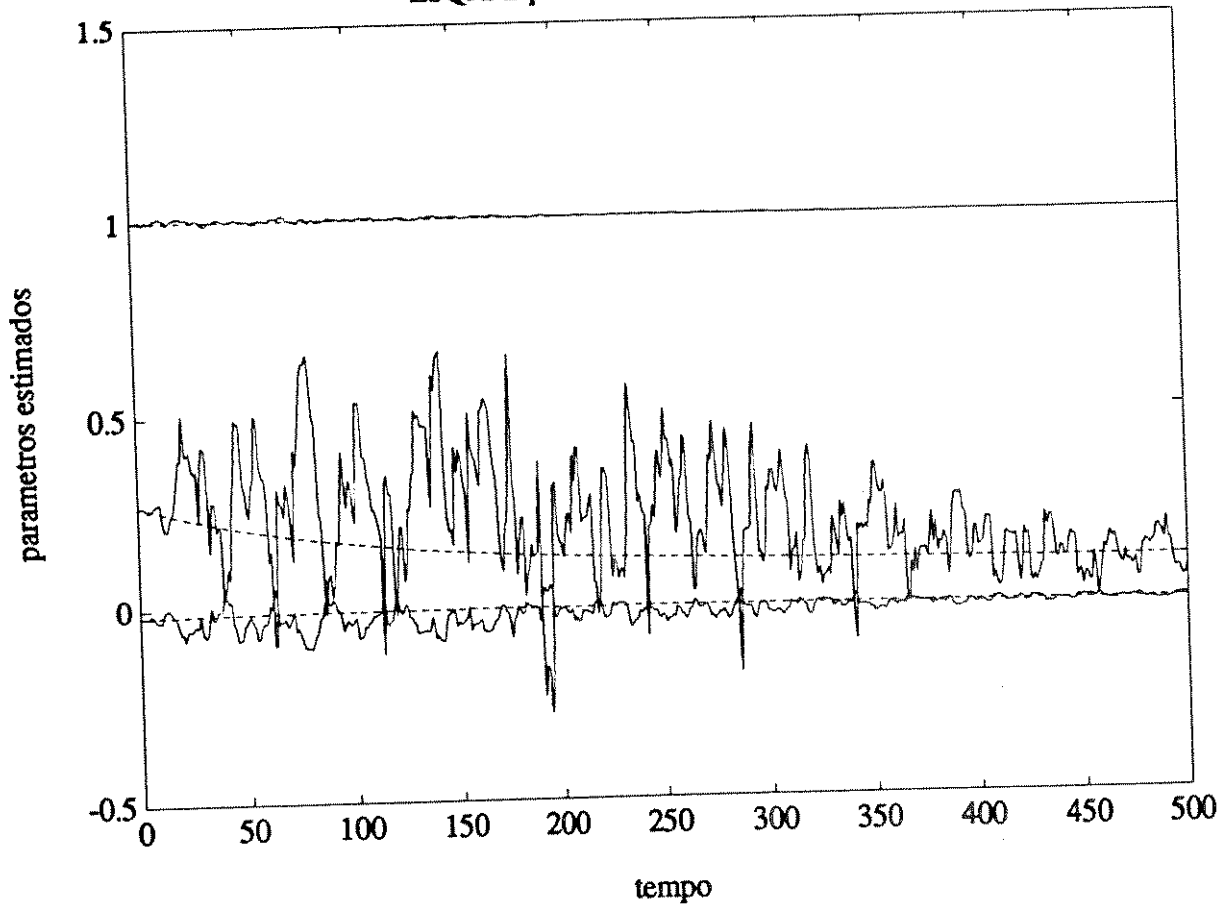


Figura 5a : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.75$



LSQRPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.75$

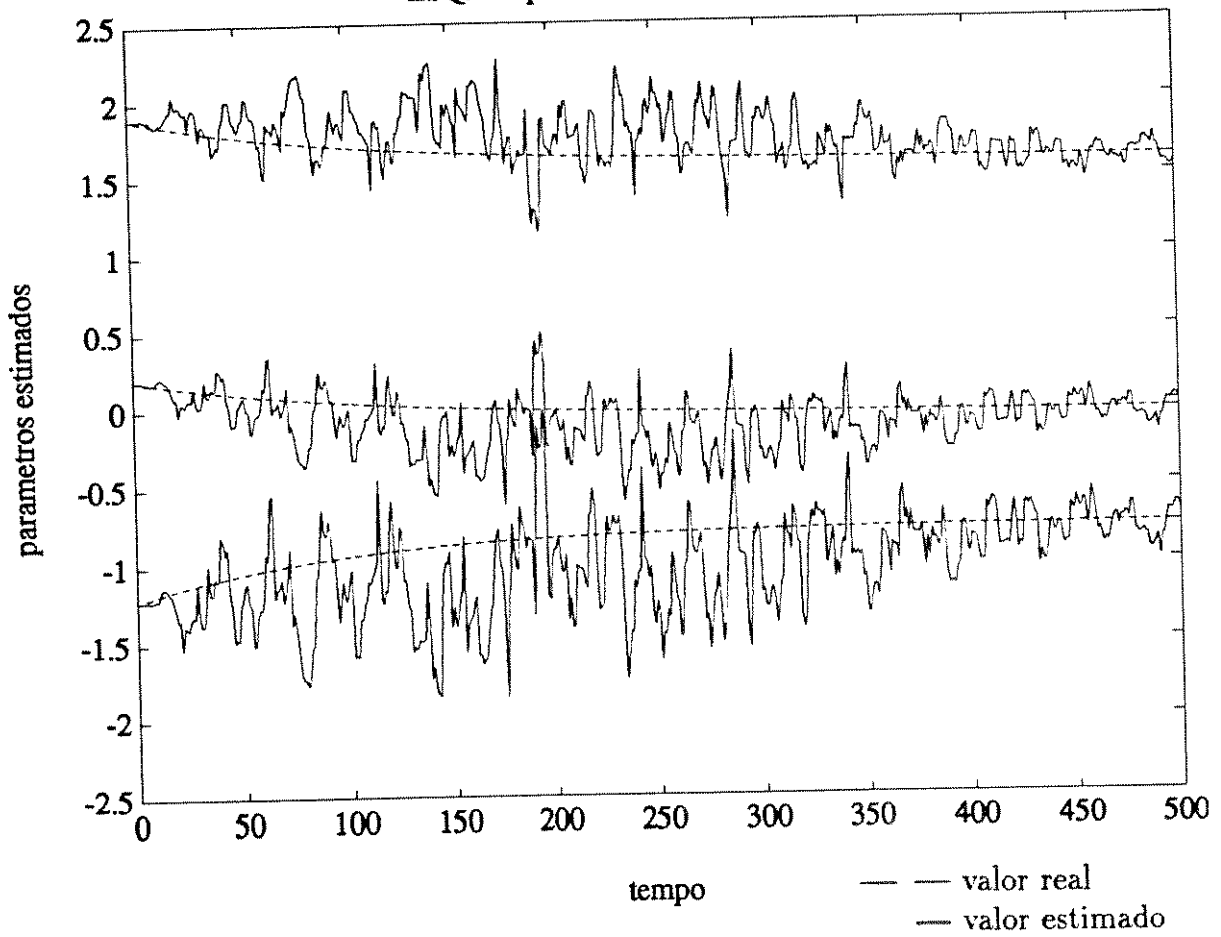
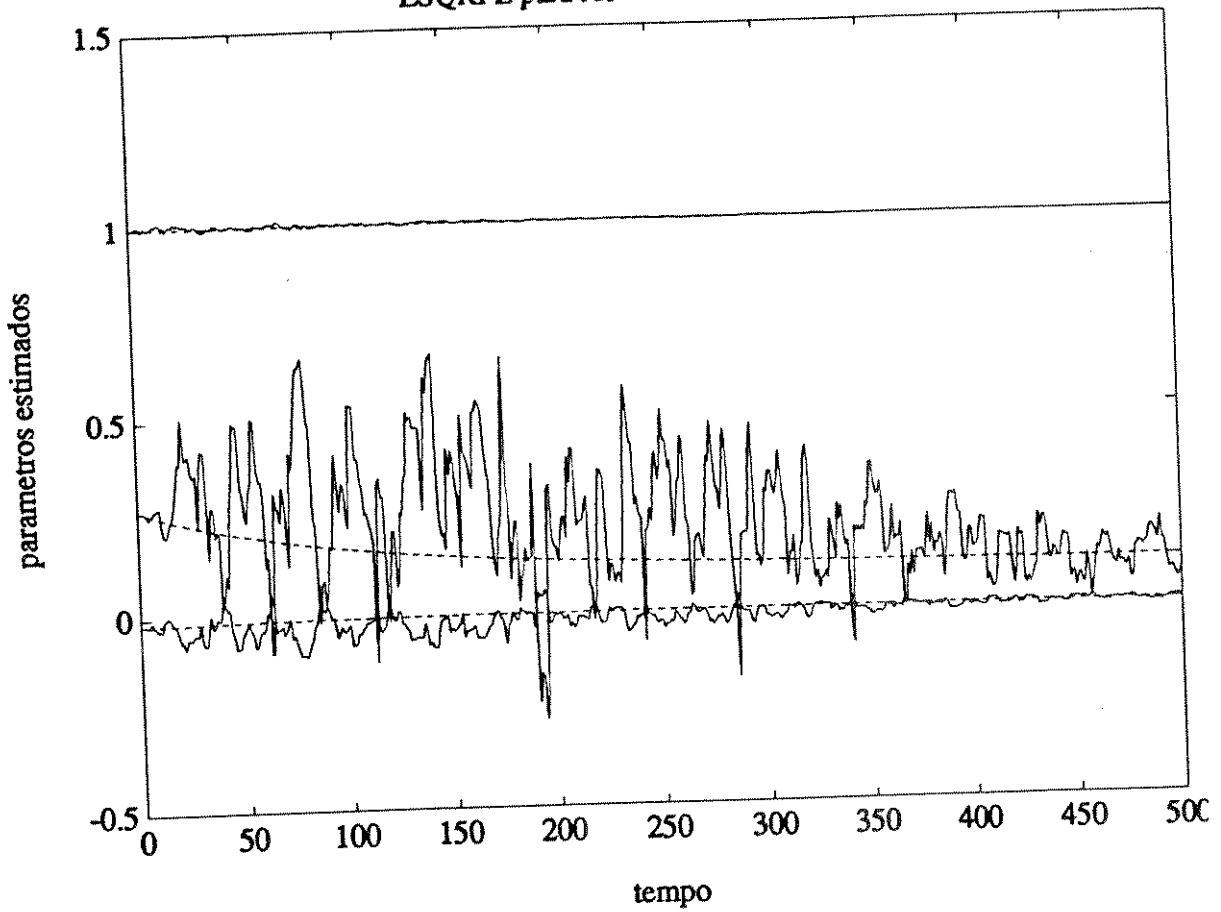


Figura 6a : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.75$



LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.75$

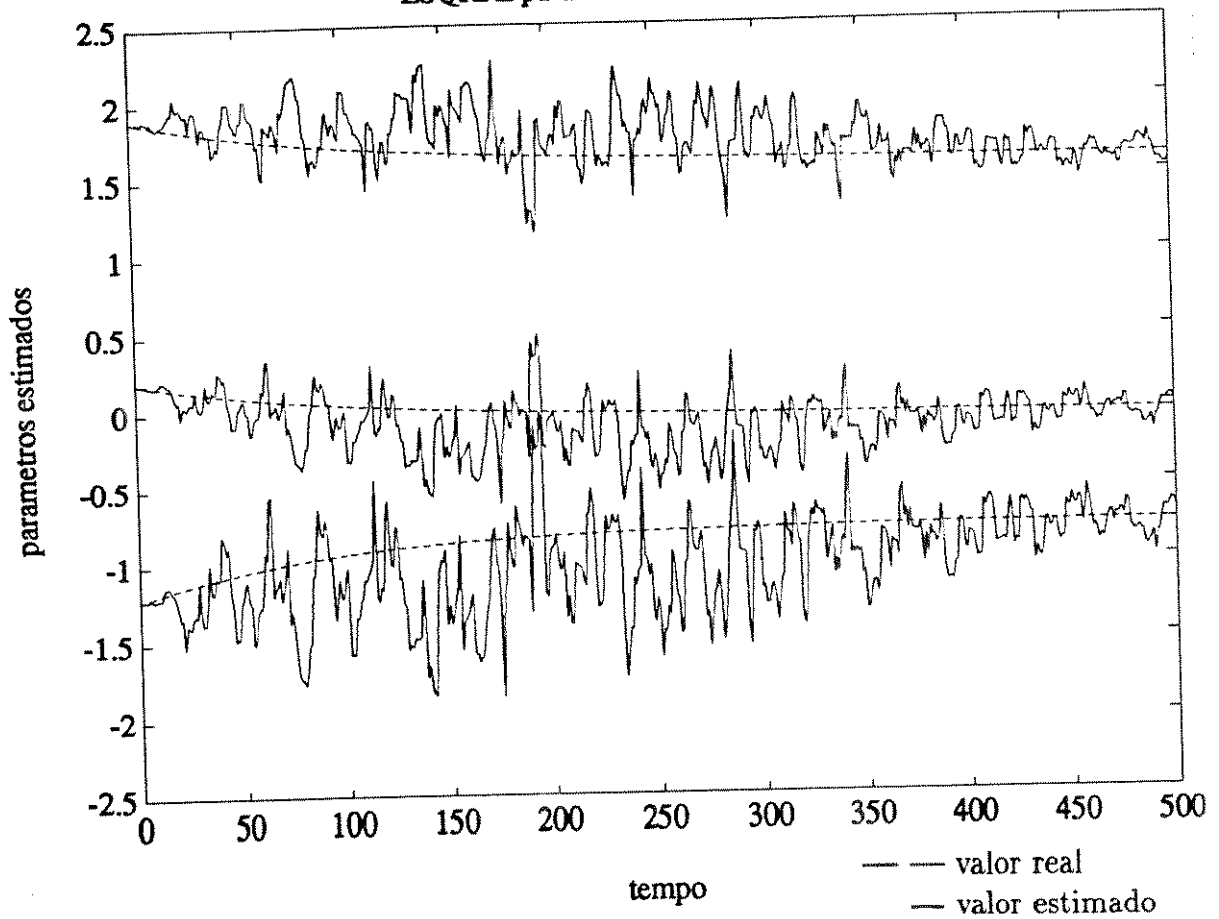


Figura 7a : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-4} \cdot I_6$ e $W = 0.9$

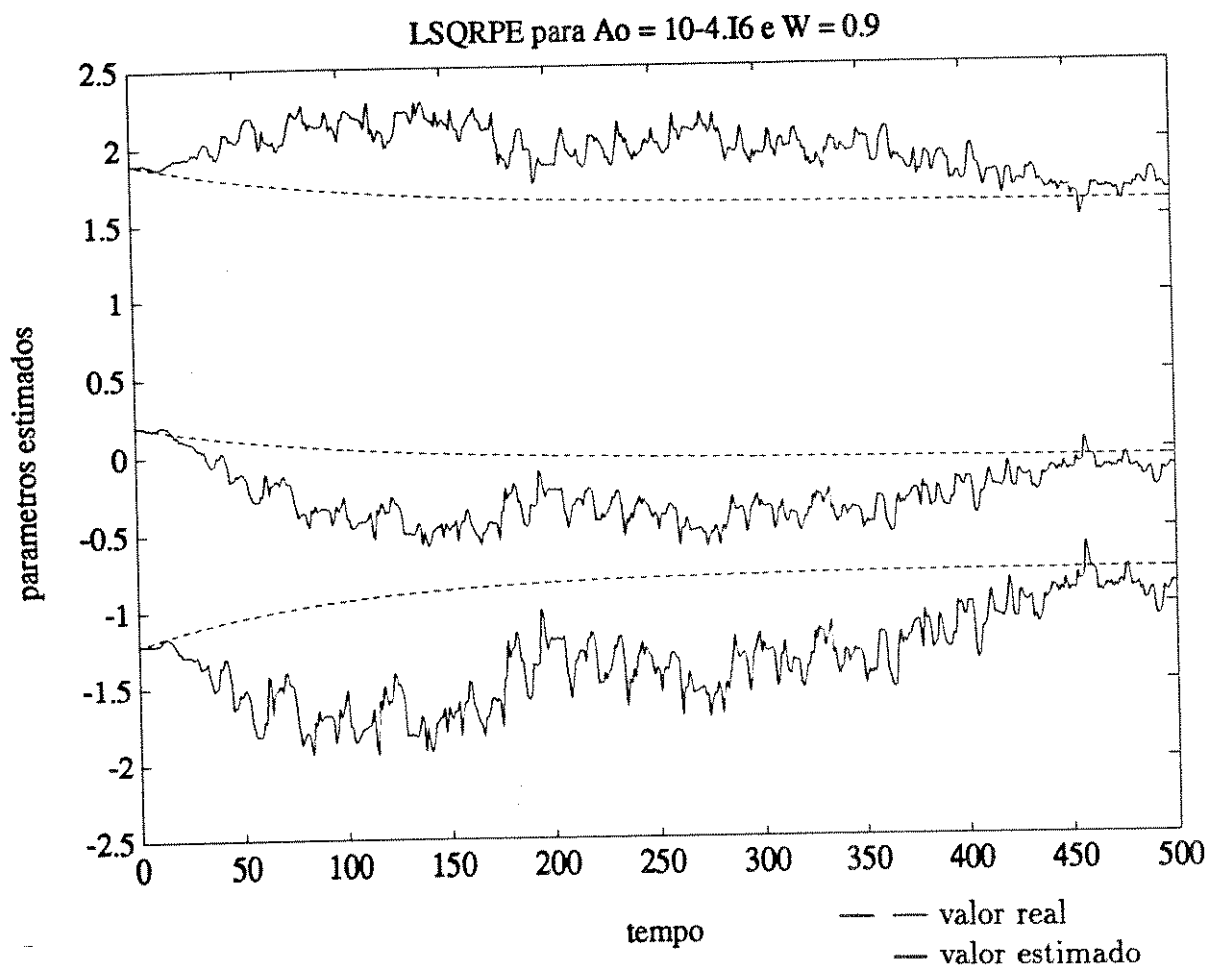
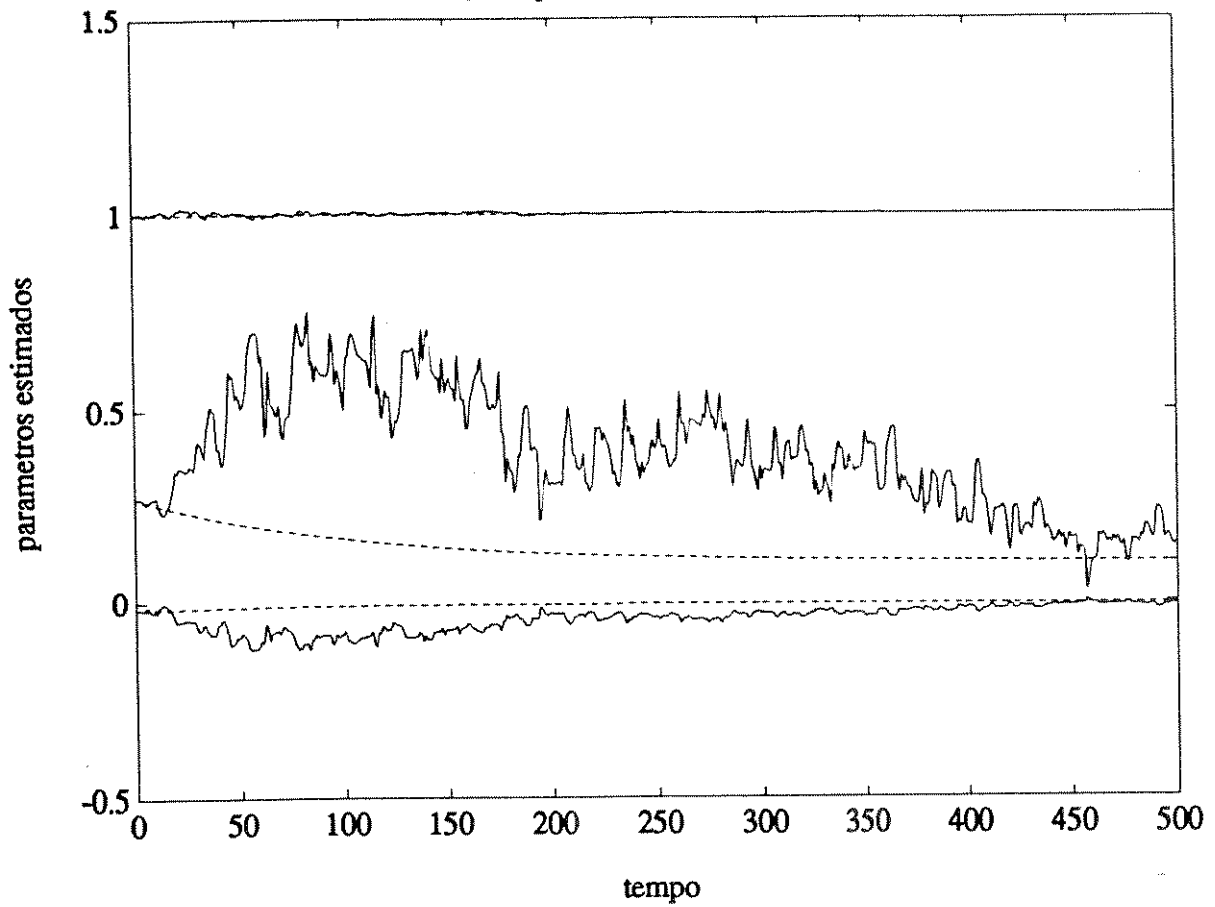
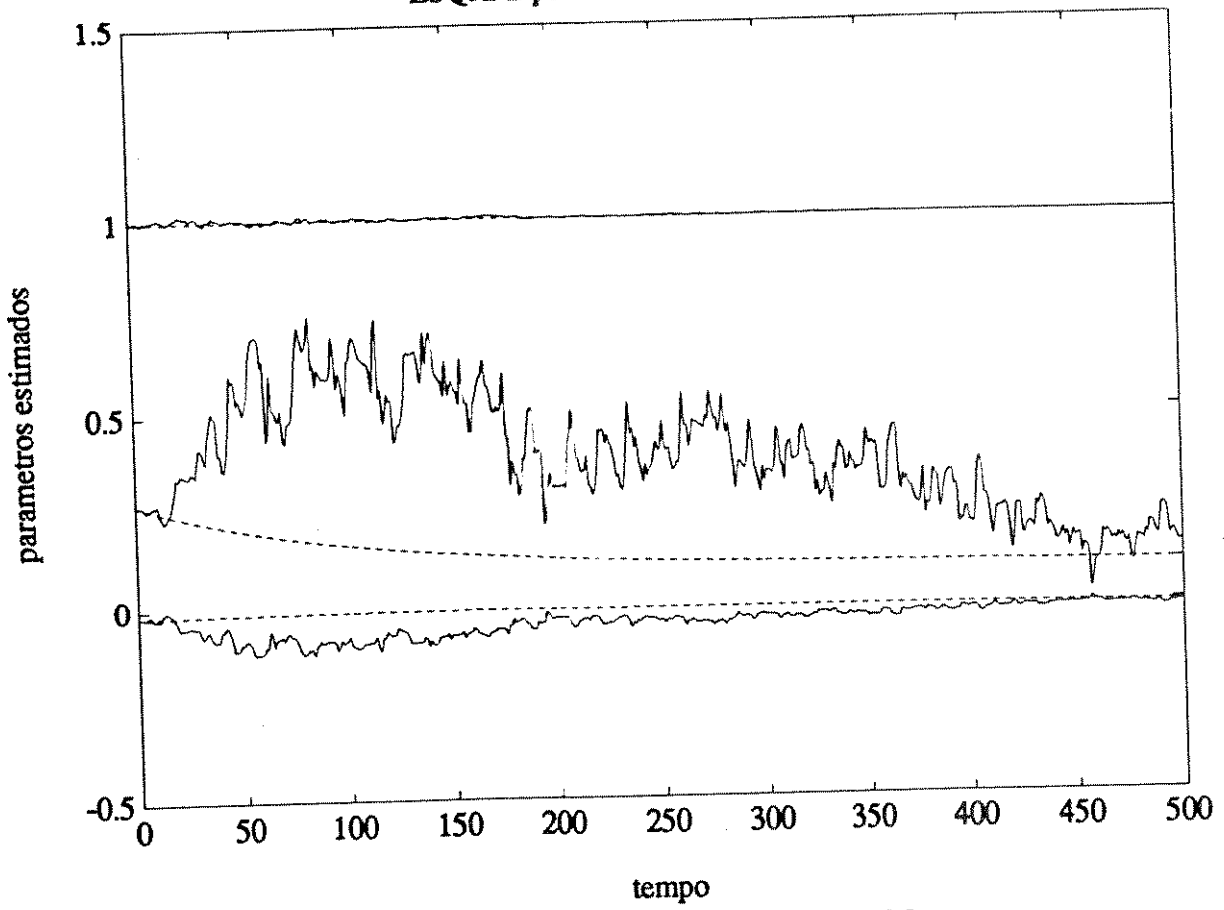


Figura 5b : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.9$



LSQRPE para $A_0 = 10^{-6} \cdot I_6$ e $W = 0.9$

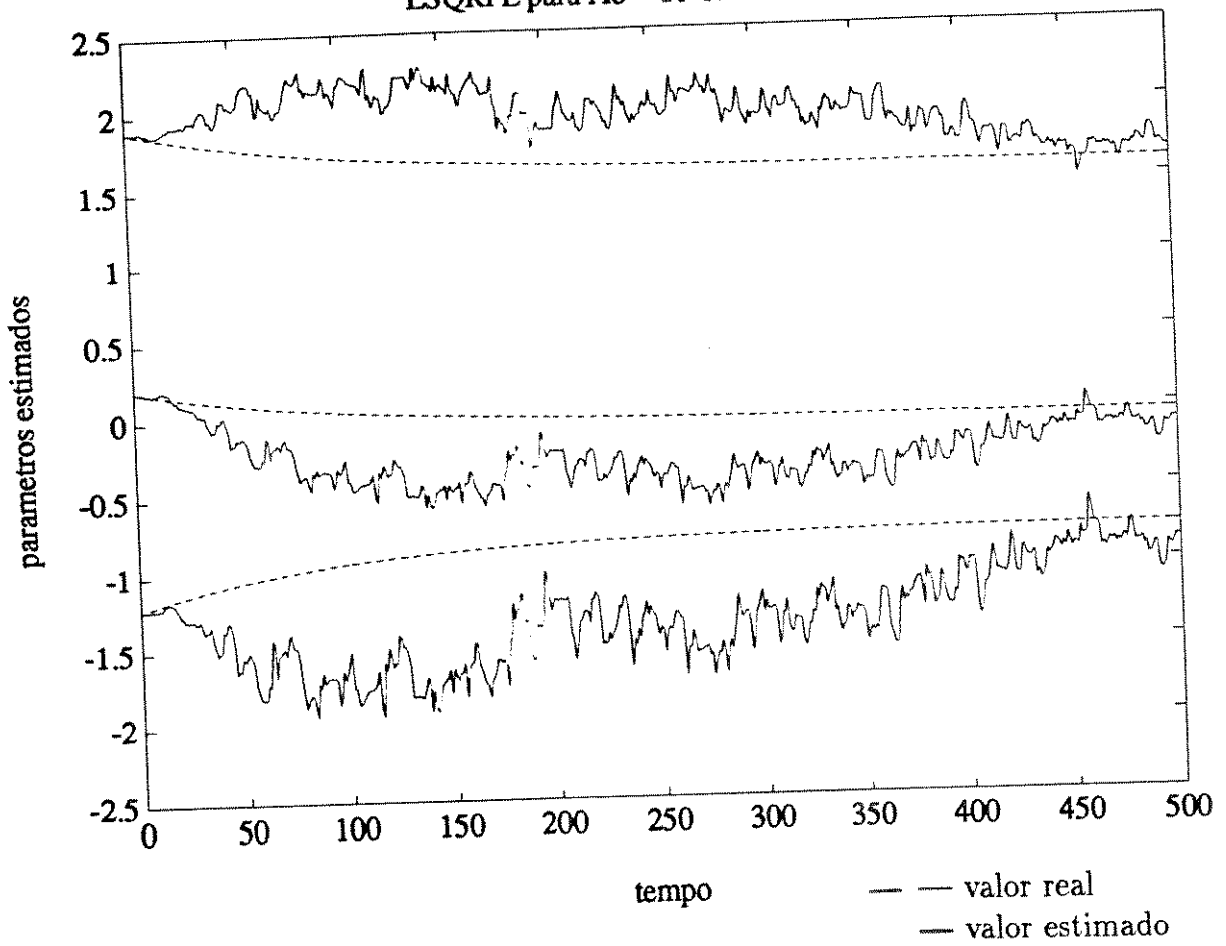
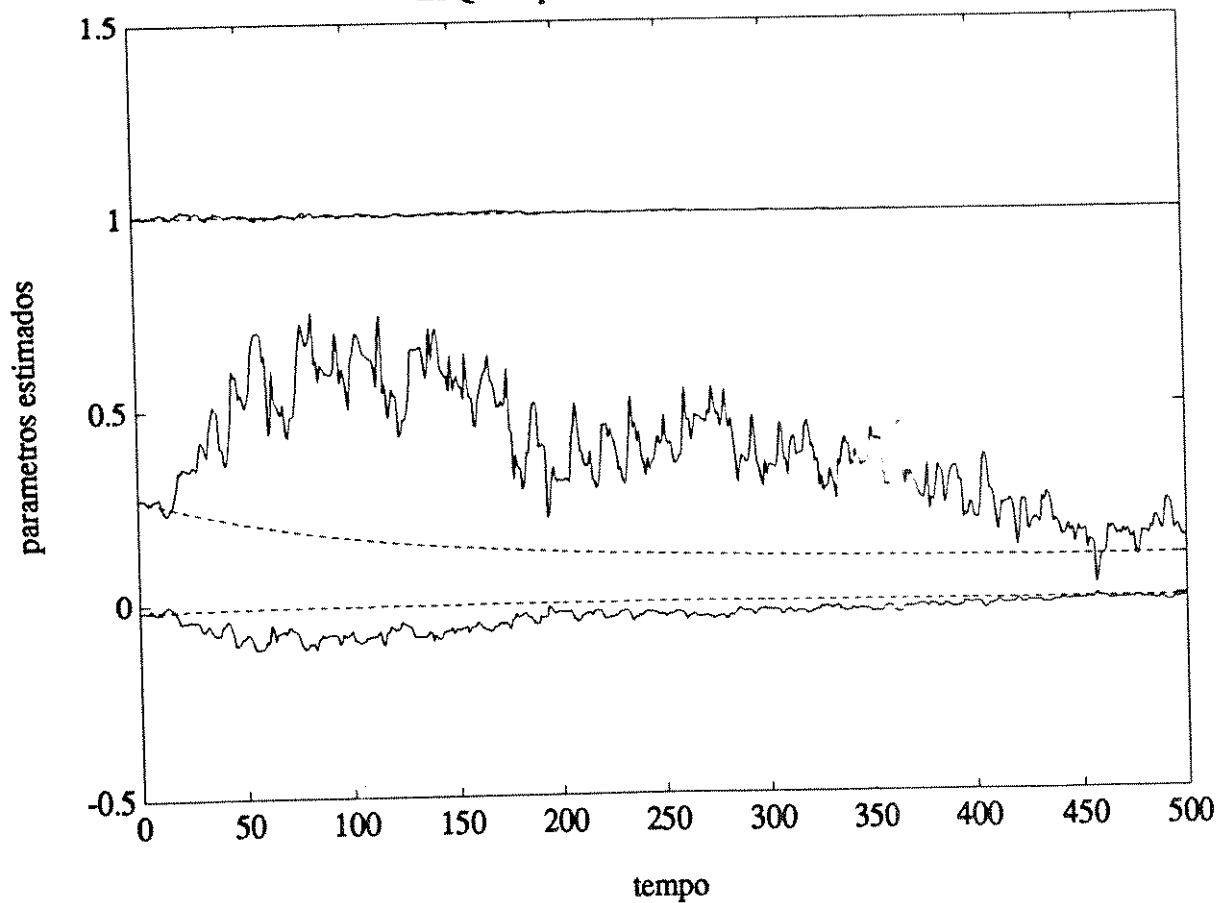


Figura 6b : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.9$



LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.9$

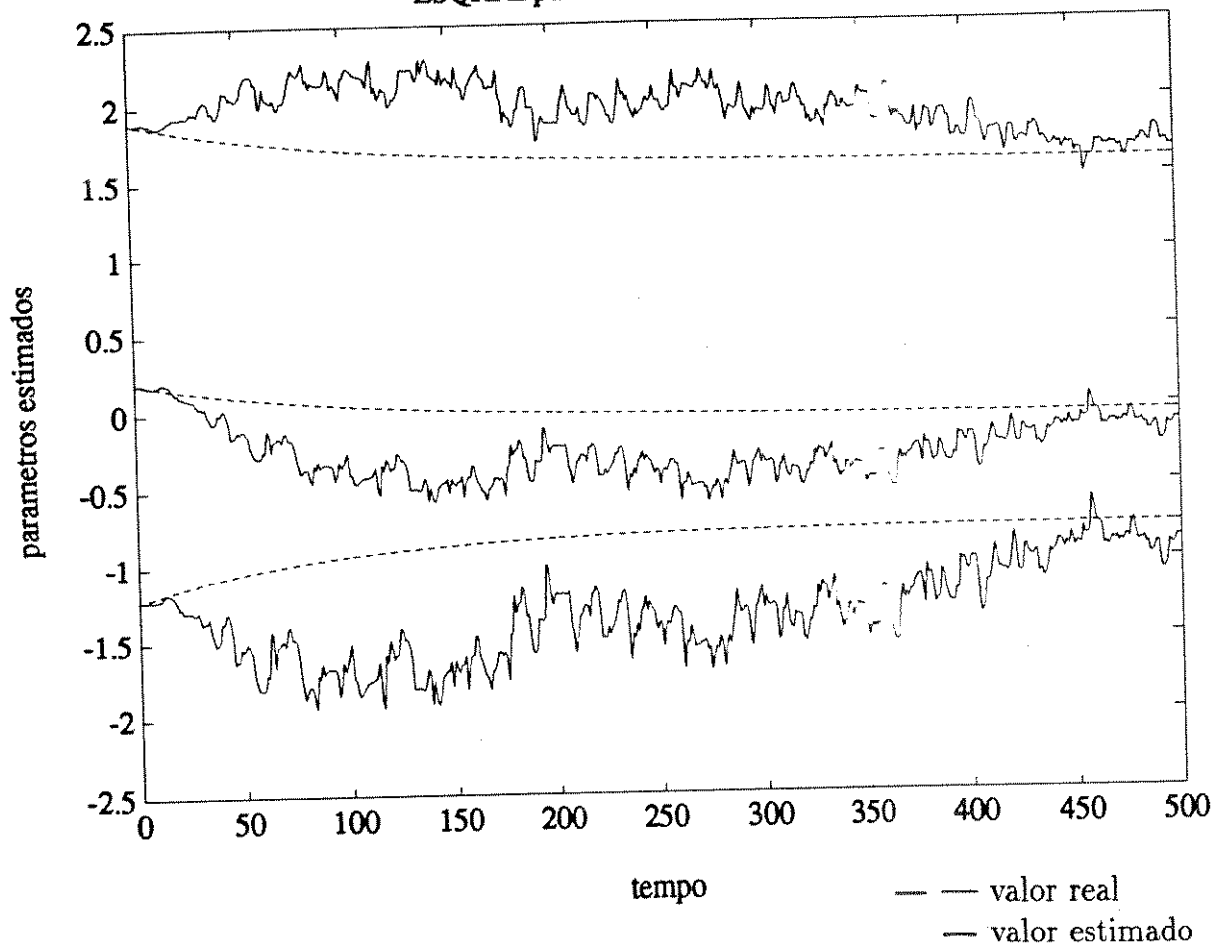
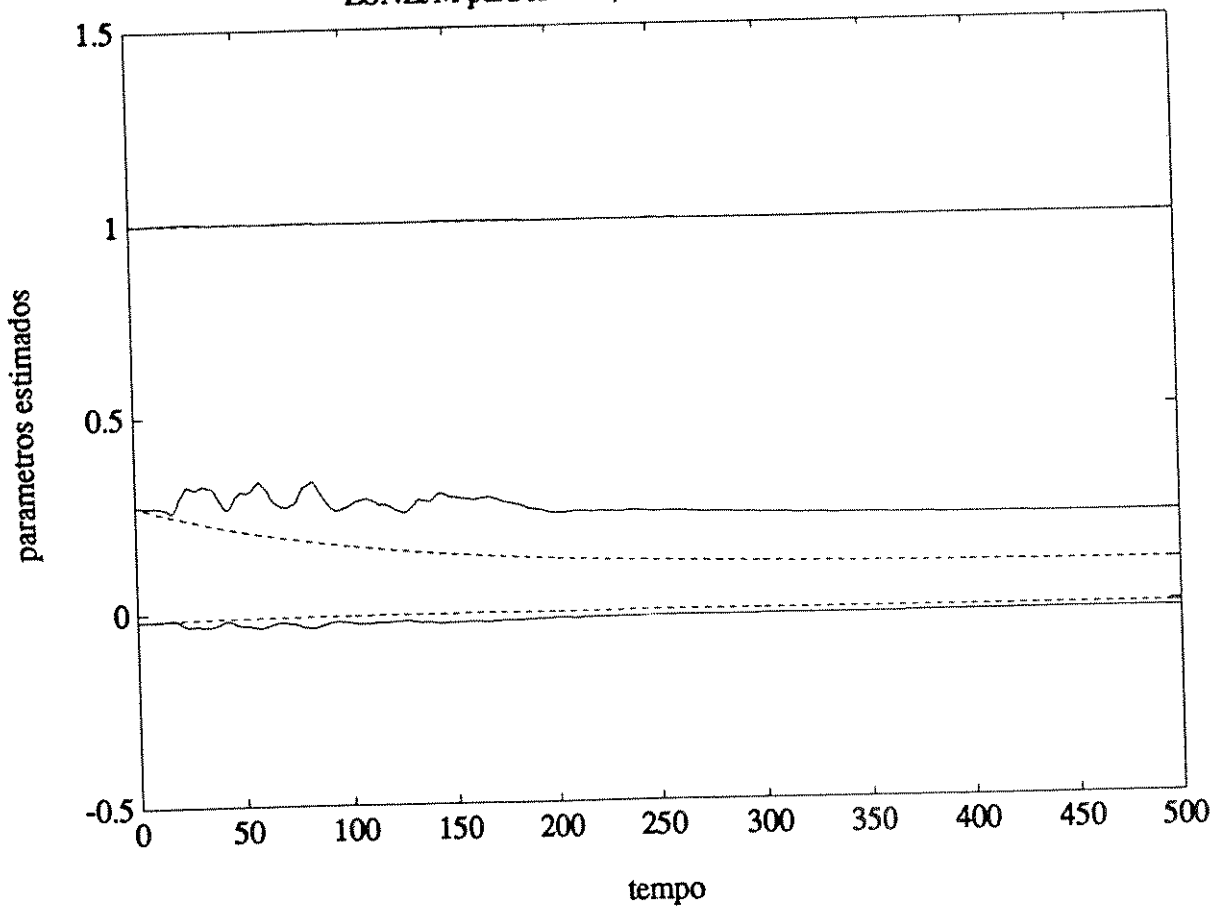


Figura 7b : Exemplo Numérico 1 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $l_d = 15$, $A_0 = 10^{-4} I_6$ e $\alpha = 0.05$



LSNEJM para $l_d = 15$, $A_0 = 10^{-4} I_6$ e $\alpha = 0.05$

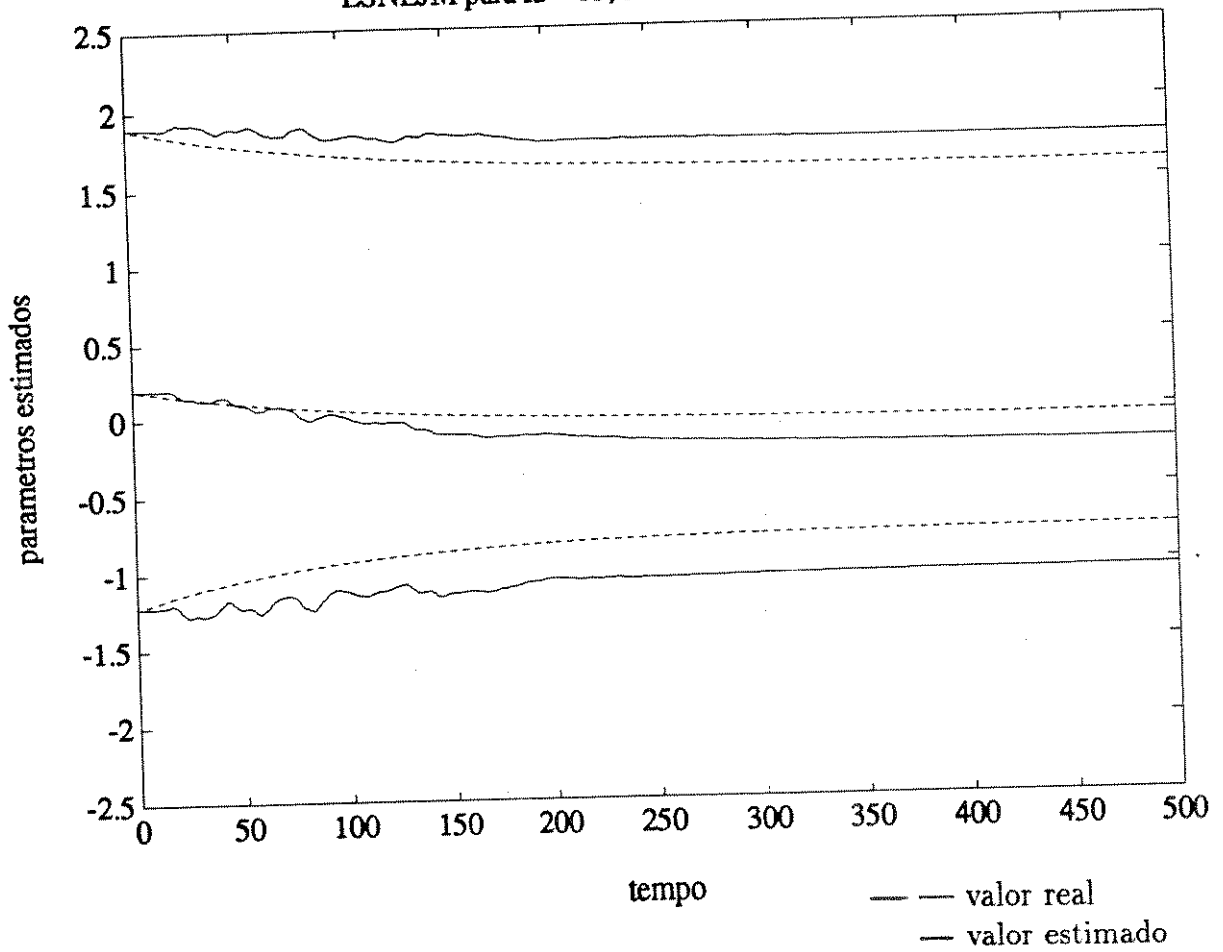
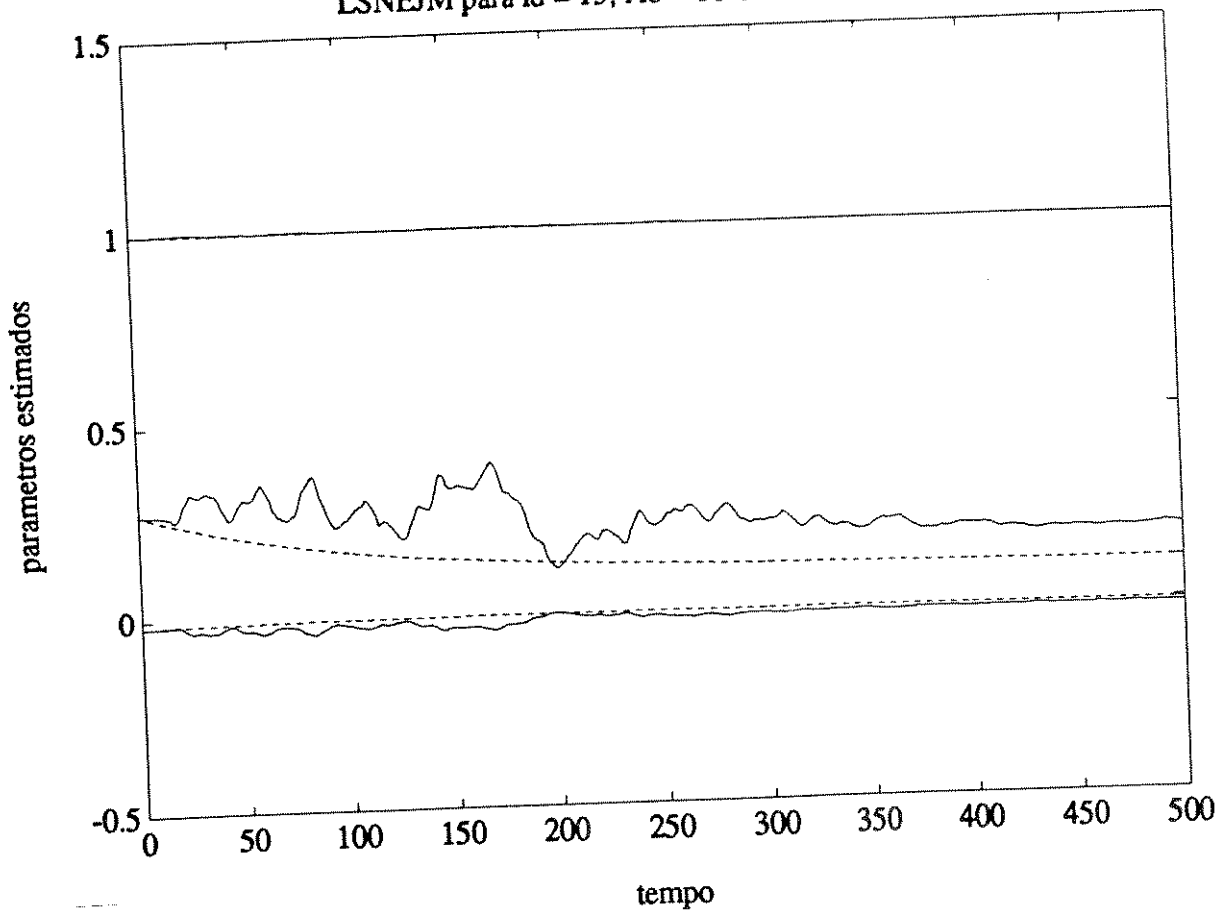


Figura 8a : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $l_d = 15$, $A_o = 10^{-6.16}$ e $\alpha = 0.05$



LSNEJM para $l_d = 15$, $A_o = 10^{-6.16}$ e $\alpha = 0.05$

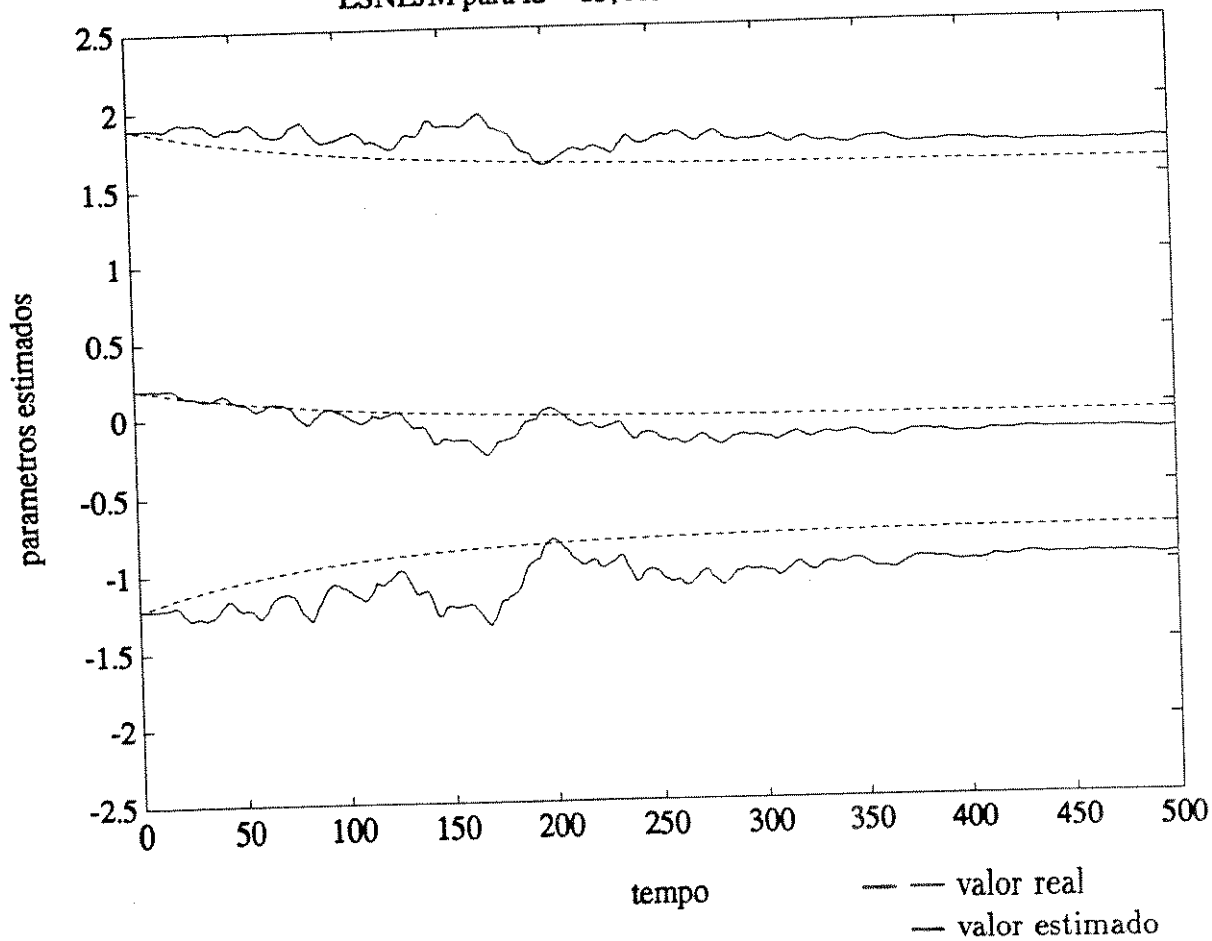
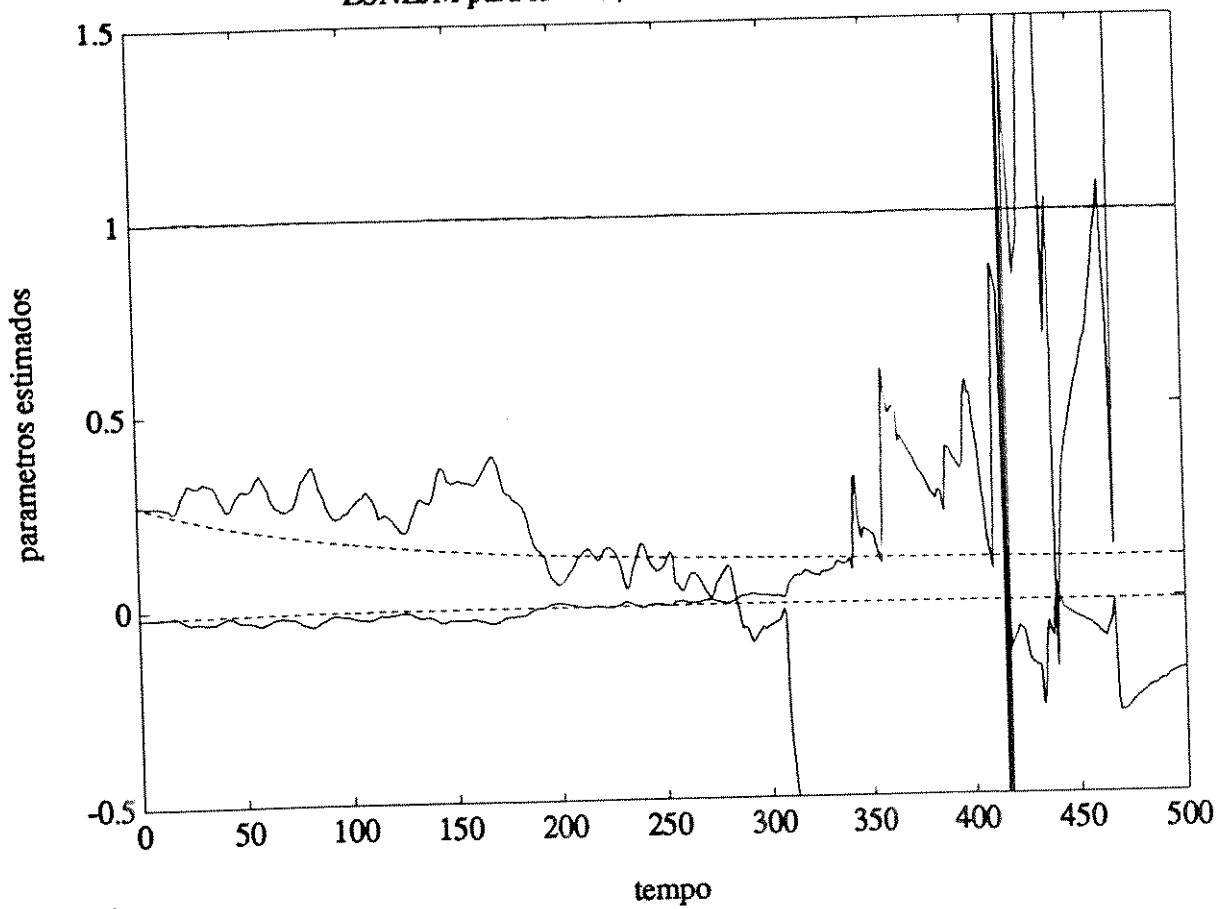


Figura 9a : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $l_d = 15$, $A_0 = 10^{-8.16}$ e $\alpha = 0.05$



LSNEJM para $l_d = 15$, $A_0 = 10^{-8.16}$ e $\alpha = 0.05$

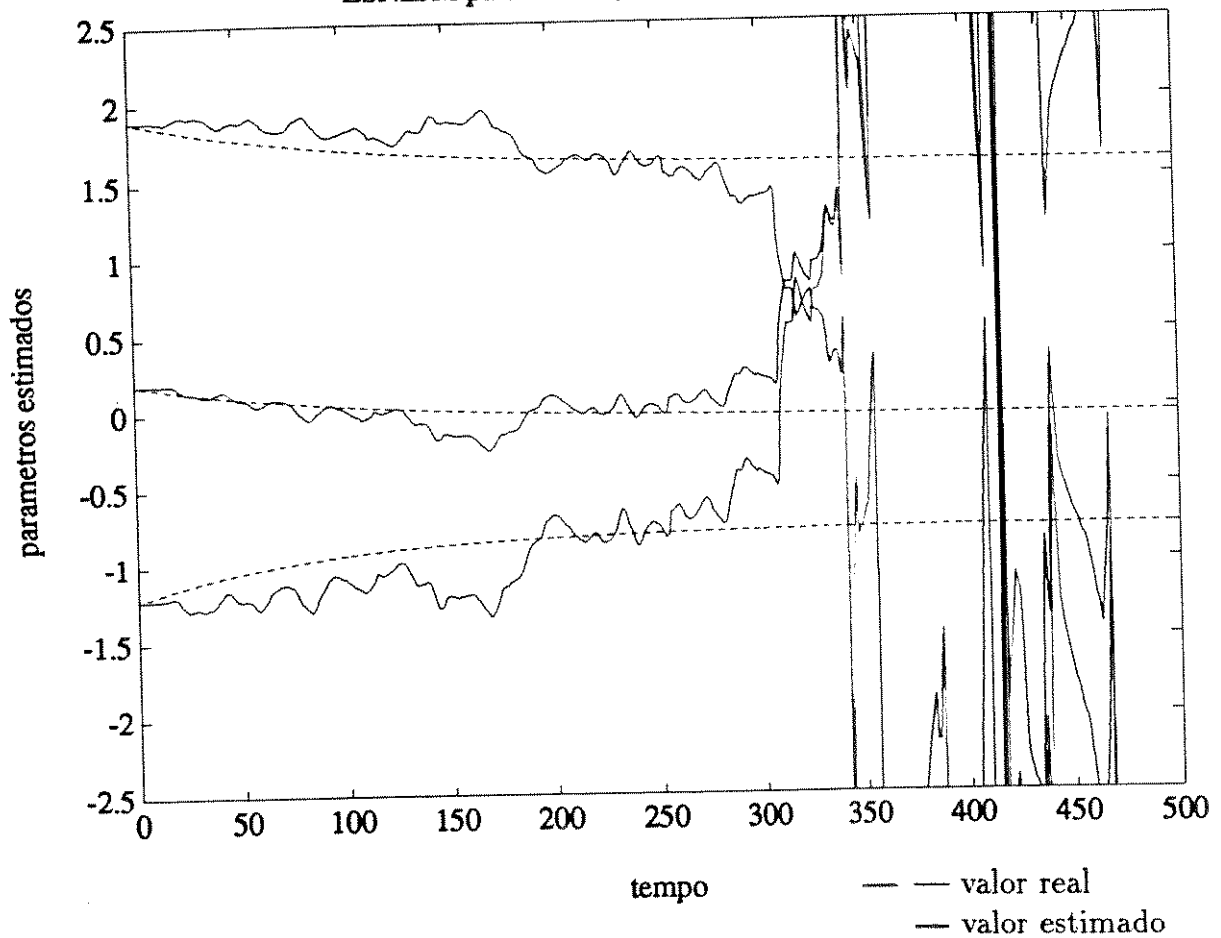
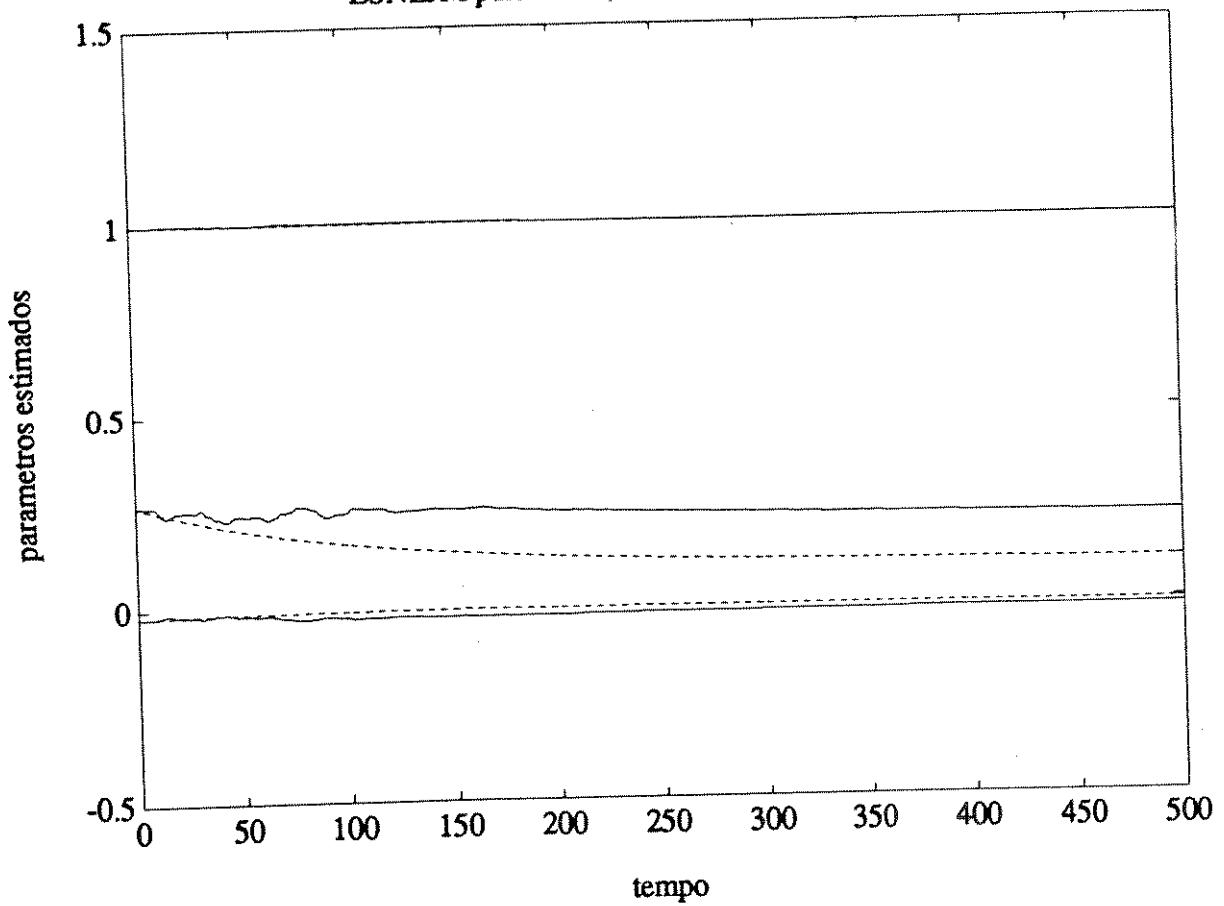


Figura 10a : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $ld = 9$, $Ao = 10^{-4}.16$ e $\alpha = 0.05$



LSNEJM para $ld = 9$, $Ao = 10^{-4}.16$ e $\alpha = 0.05$

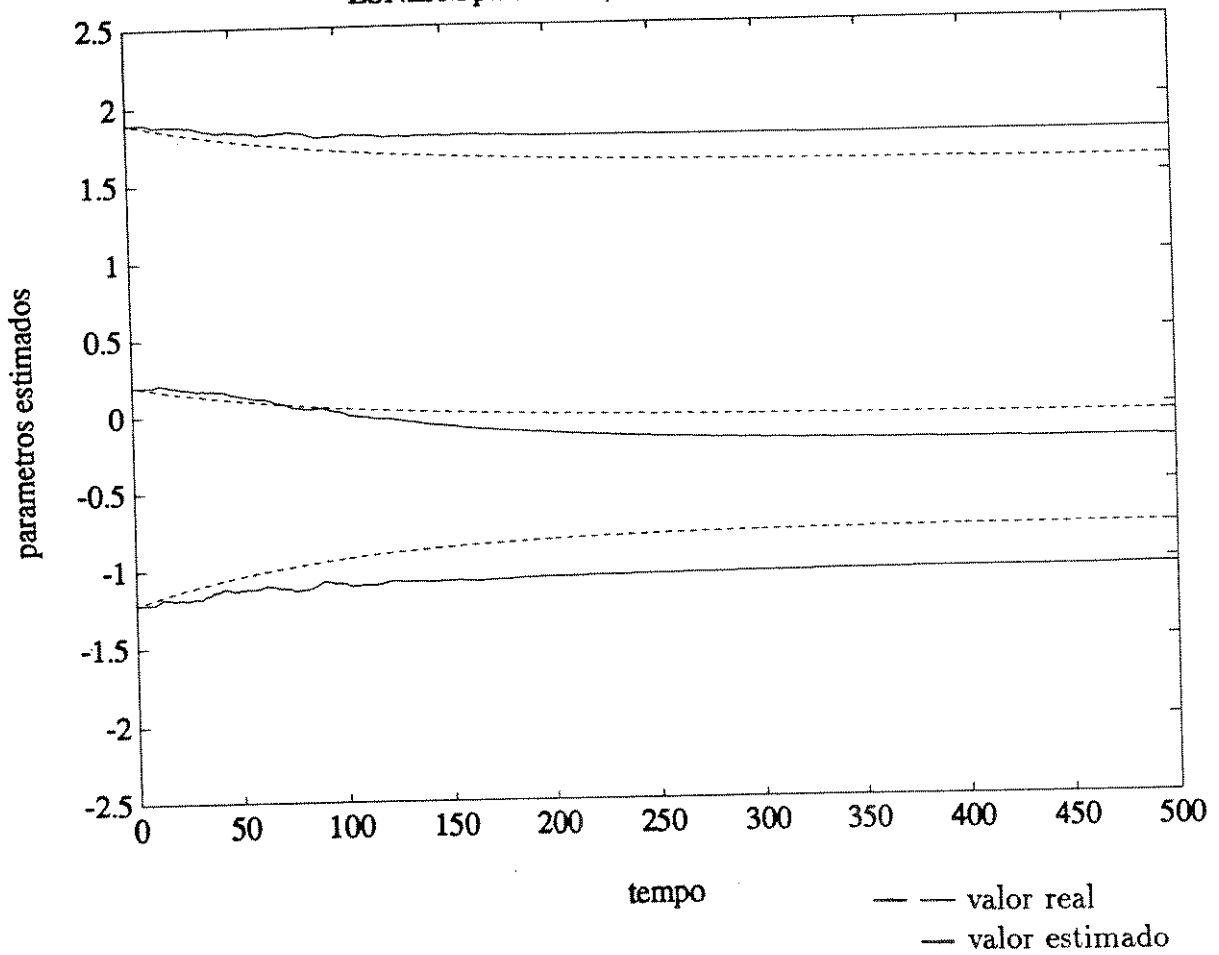
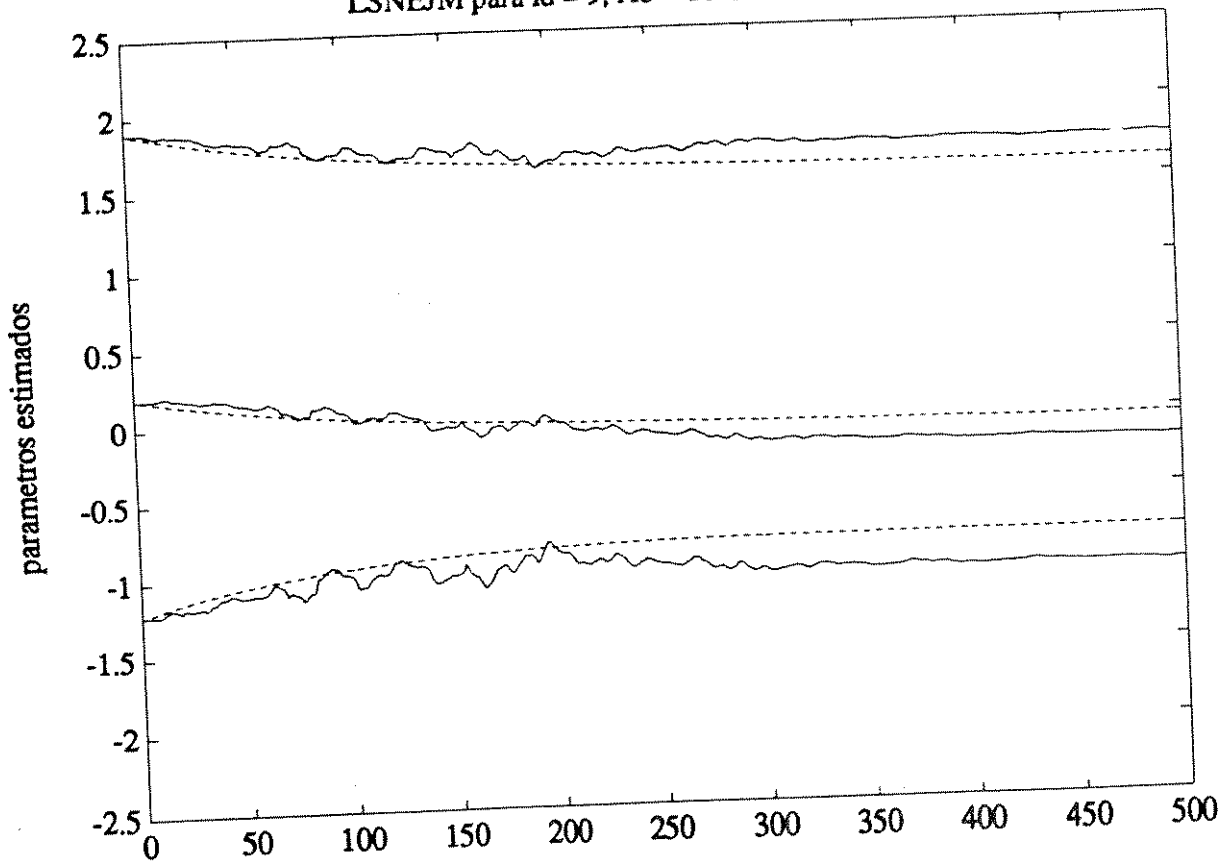


Figura 8b : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $ld = 9$, $Ao = 10^{-6.16}$ e $\alpha = 0.05$



LSNEJM para $ld = 9$, $Ao = 10^{-6.16}$ e $\alpha = 0.05$

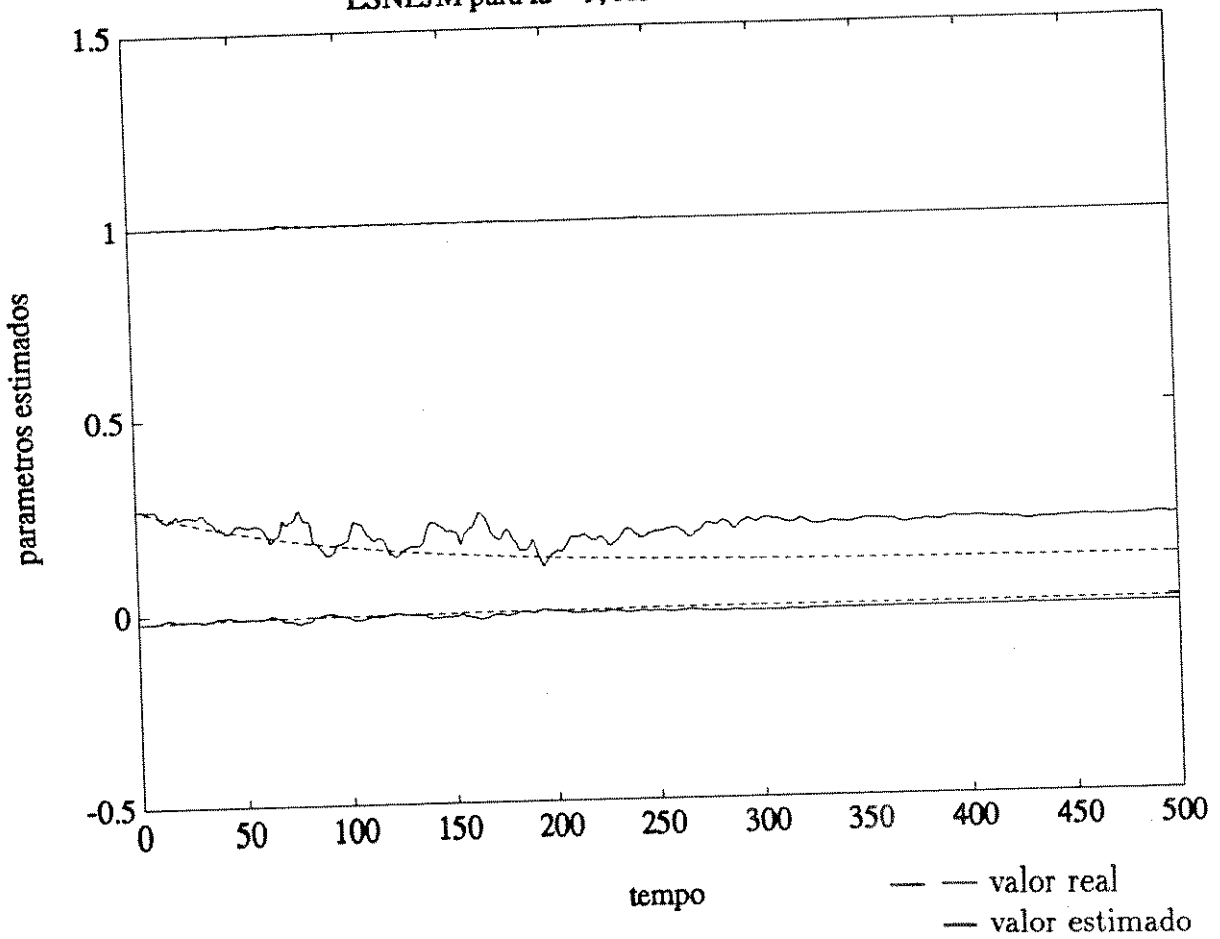
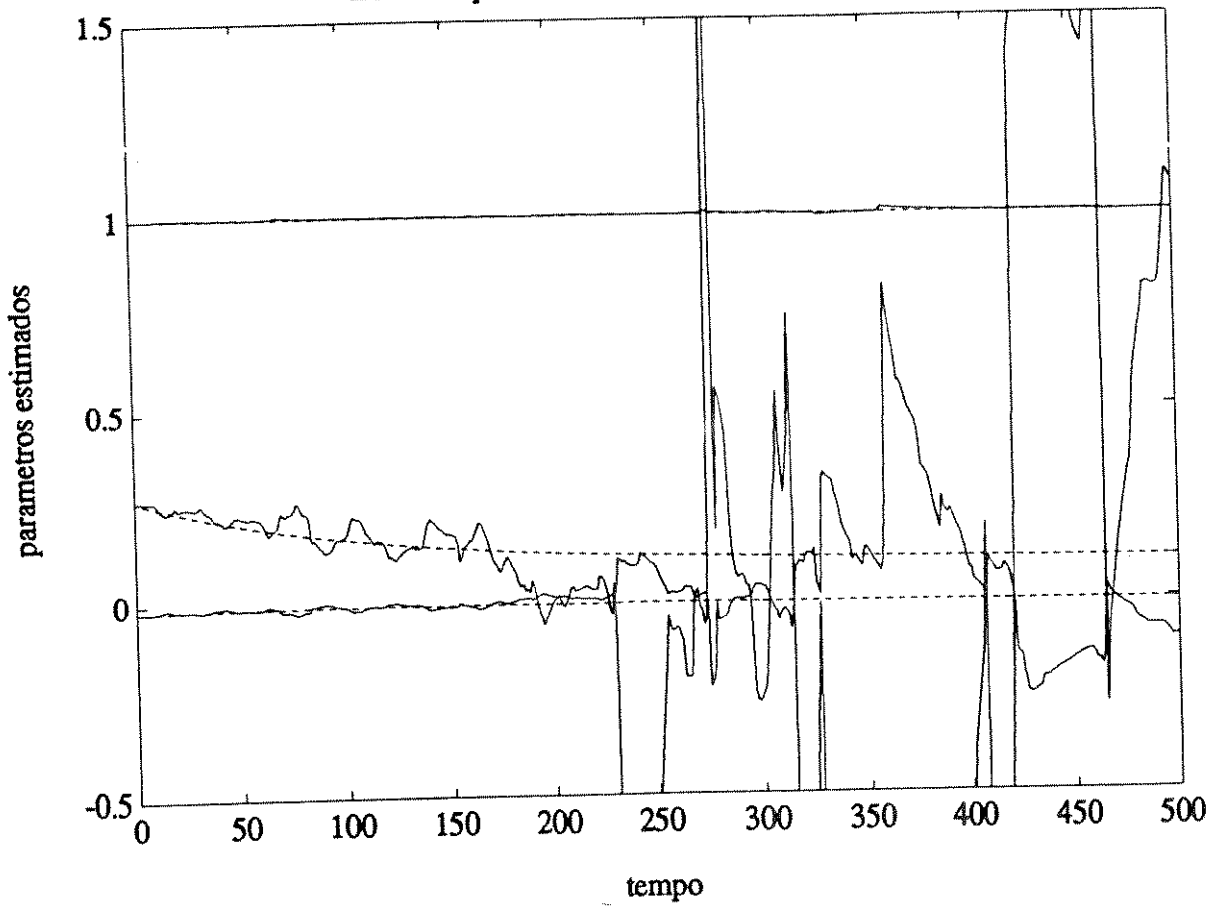


Figura 9b : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $l_d = 9$, $A_0 = 10^{-8.16}$ e $\alpha = 0.05$



LSNEJM para $l_d = 9$, $A_0 = 10^{-8.16}$ e $\alpha = 0.05$

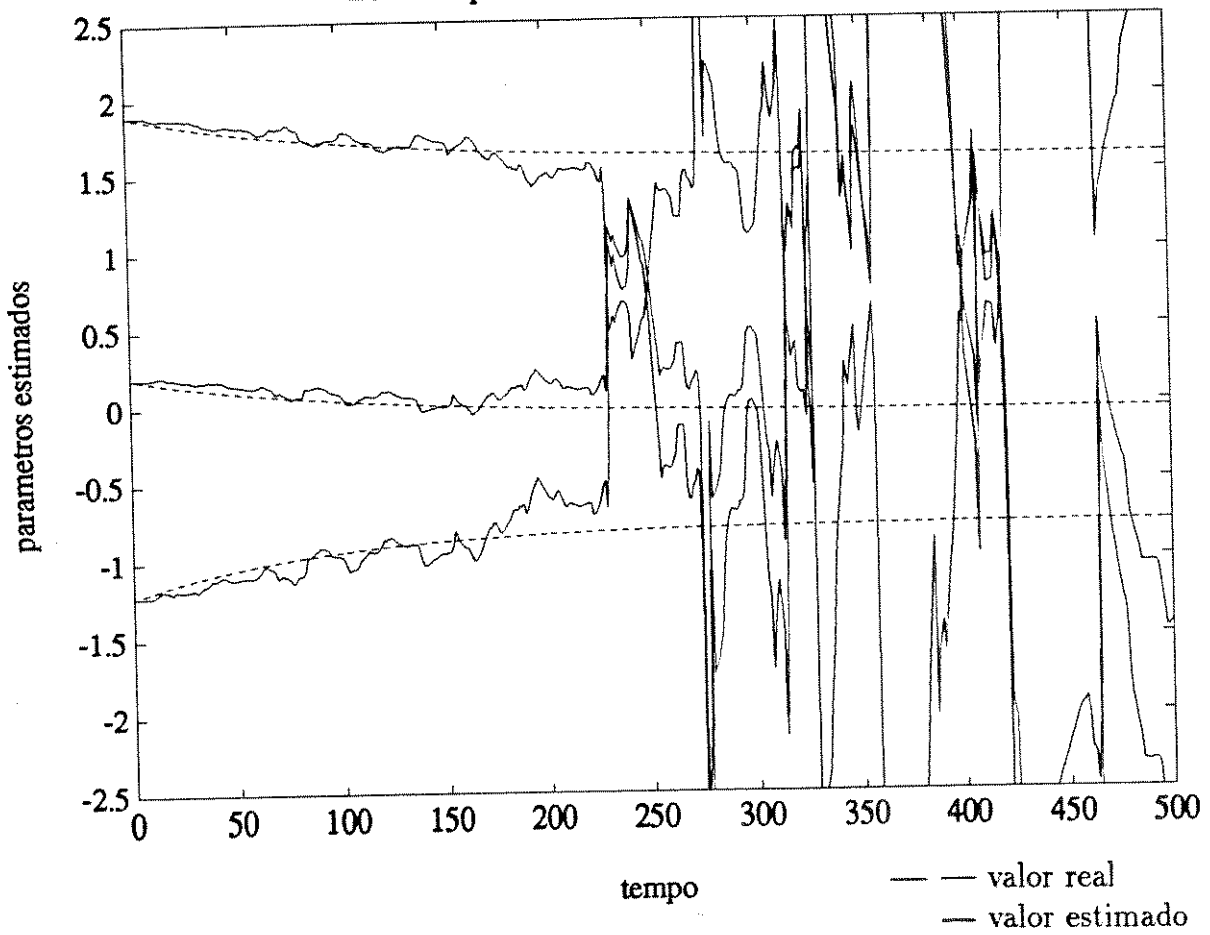
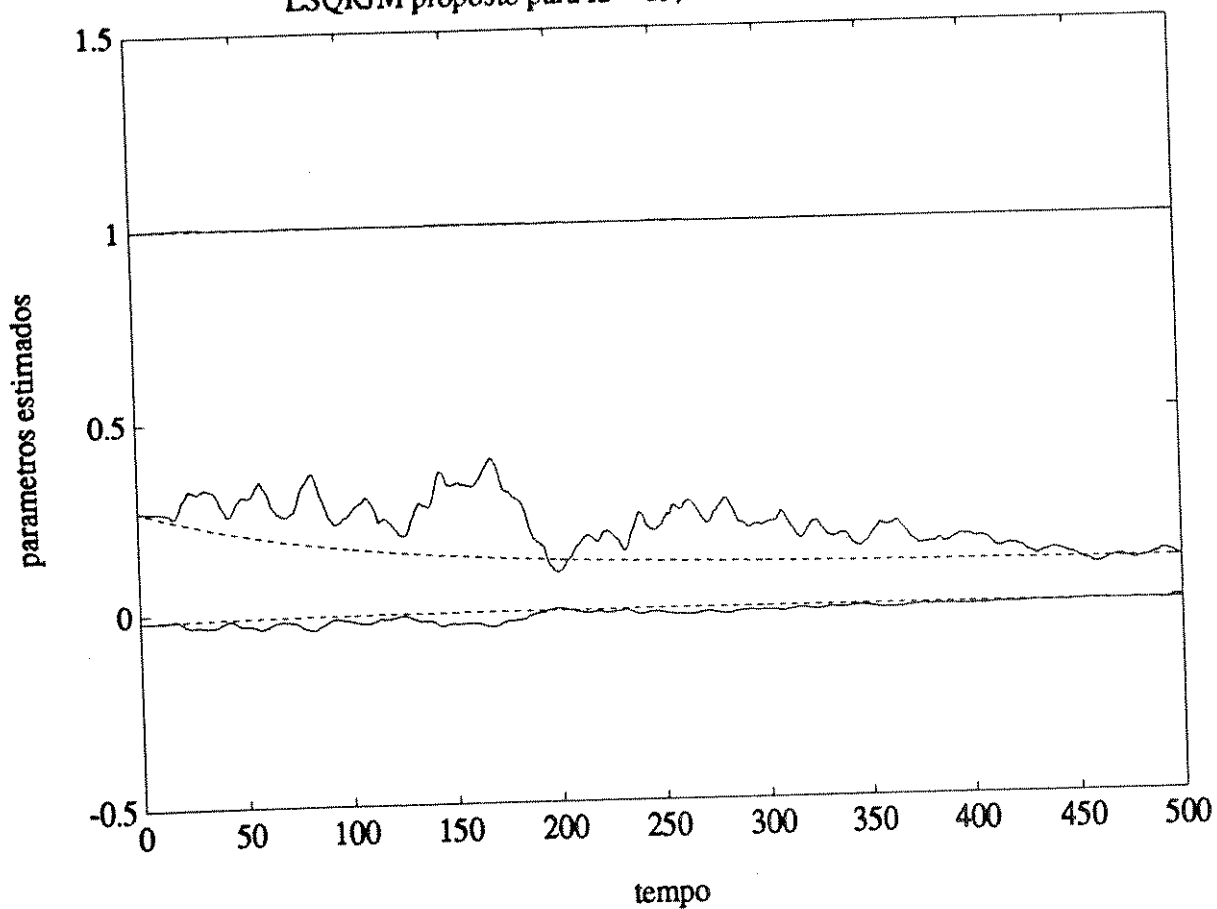


Figura 10b : Exemplo Numérico 1 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $l_d = 15, A_o = 10^{-4}.I_6$ e $\alpha = 0.05$



LSQRJM proposto para $l_d = 15, A_o = 10^{-4}.I_6$ e $\alpha = 0.05$

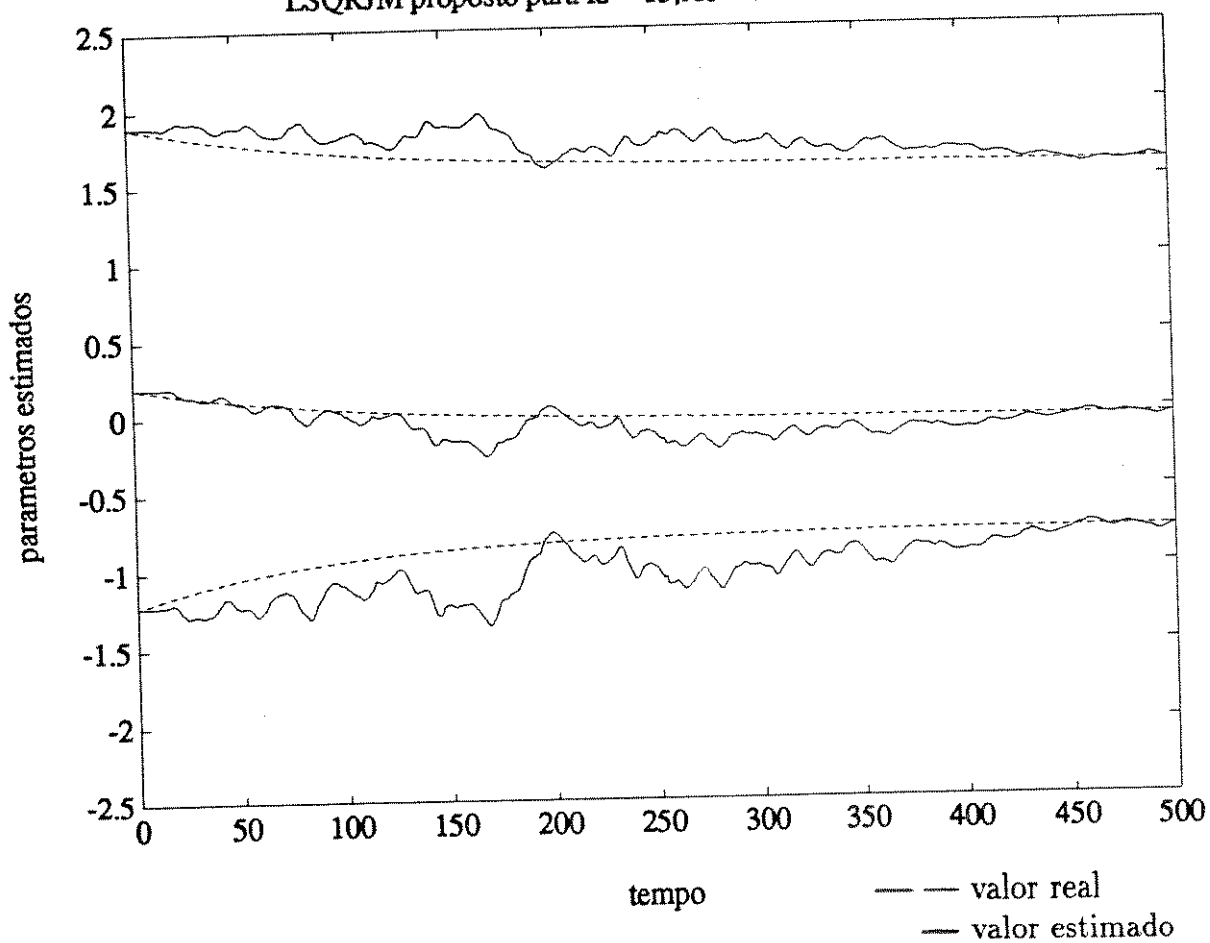
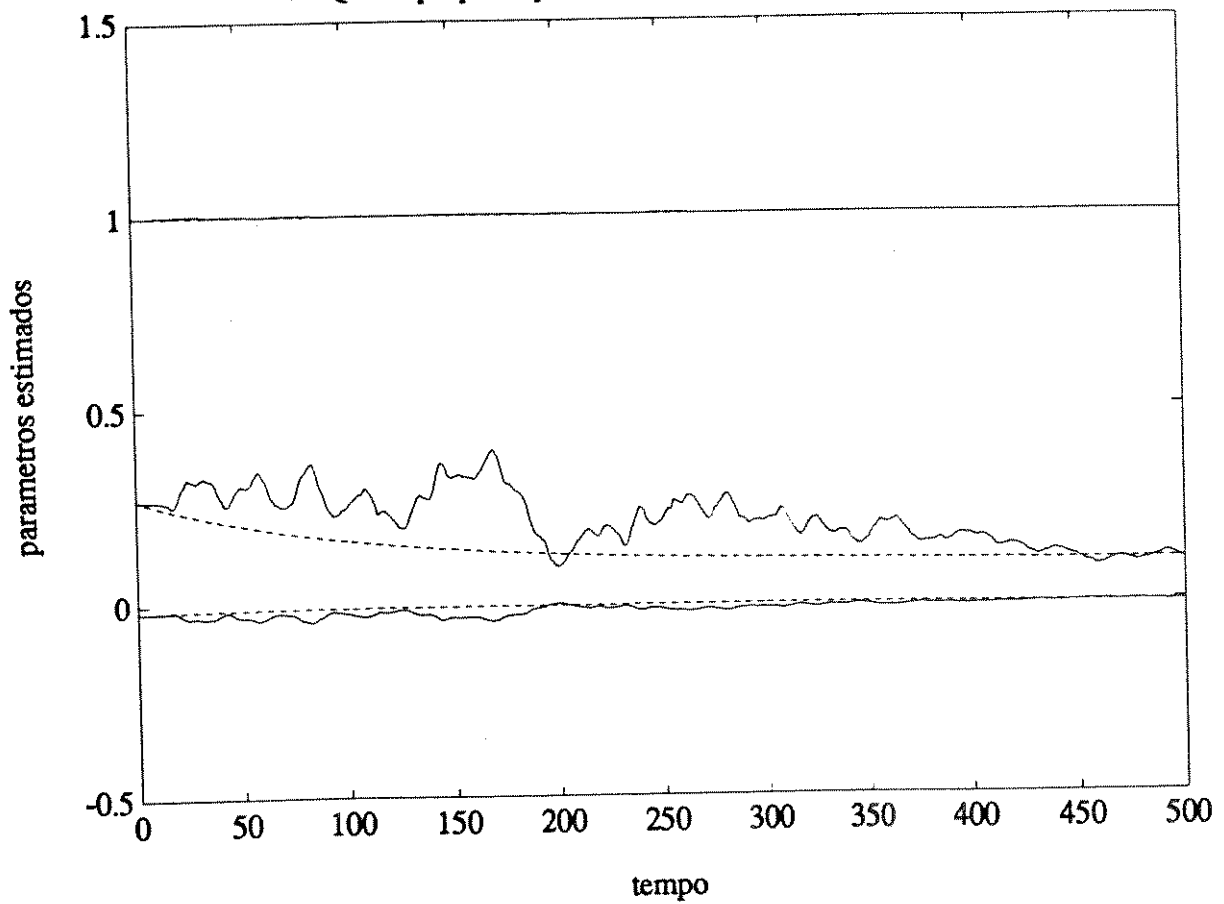


Figura 11a : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $l_d = 15, A_o = 10^{-6} \cdot I_6$ e $\alpha = 0.05$



LSQRJM proposto para $l_d = 15, A_o = 10^{-6} \cdot I_6$ e $\alpha = 0.05$

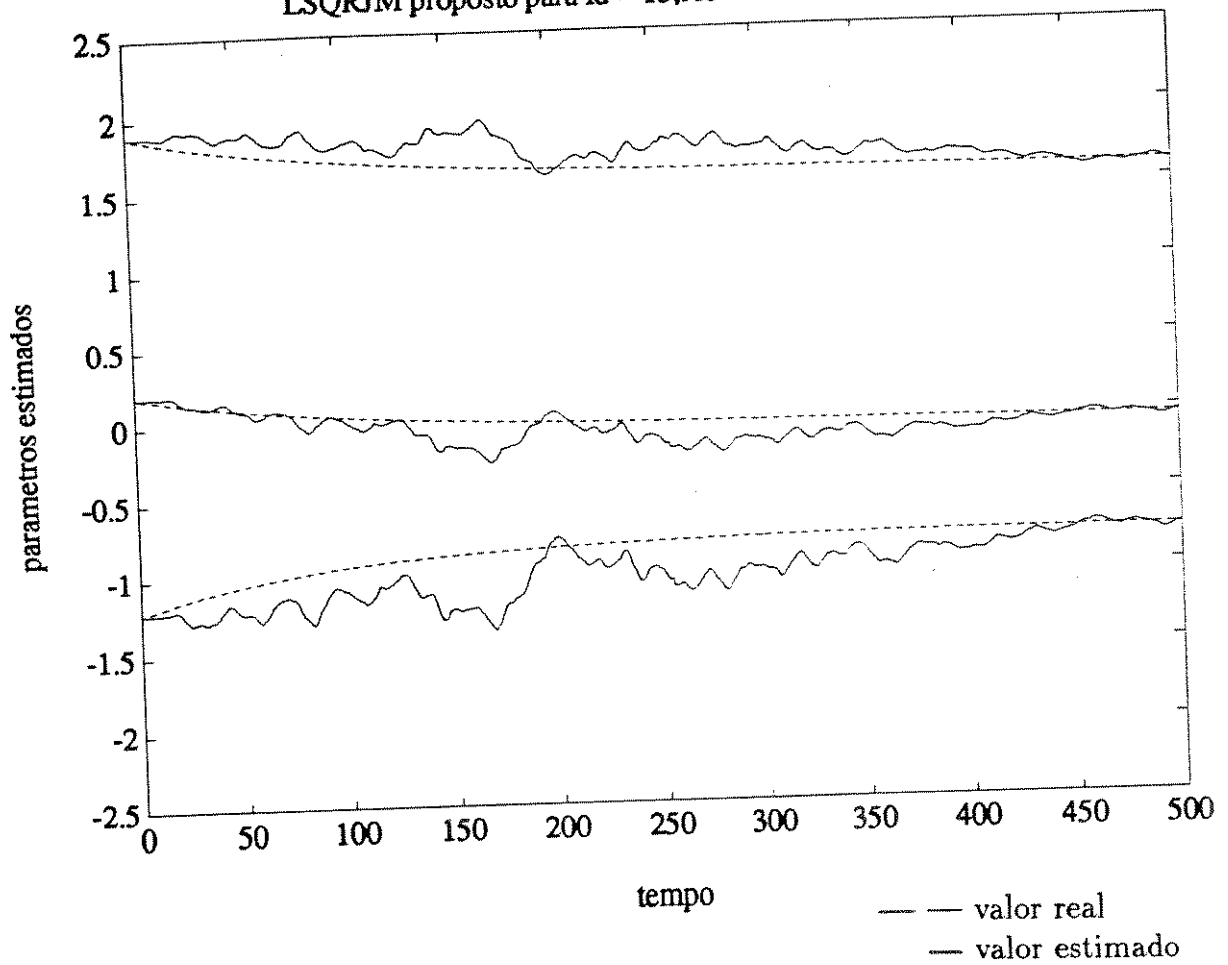
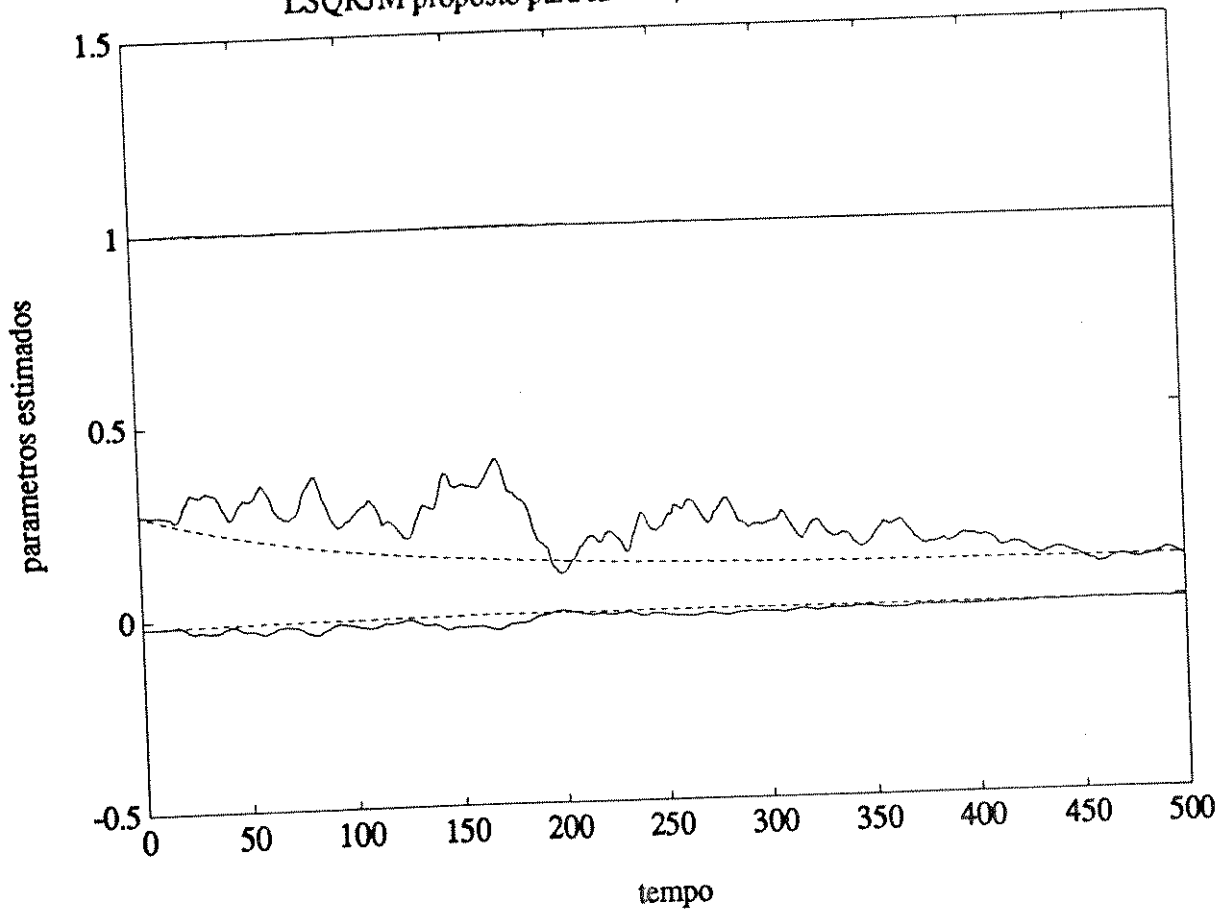


Figura 12a : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $l_d = 15, A_o = 10^{-8.16}$ e $\alpha = 0.05$



LSQRJM proposto para $l_d = 15, A_o = 10^{-8.16}$ e $\alpha = 0.05$

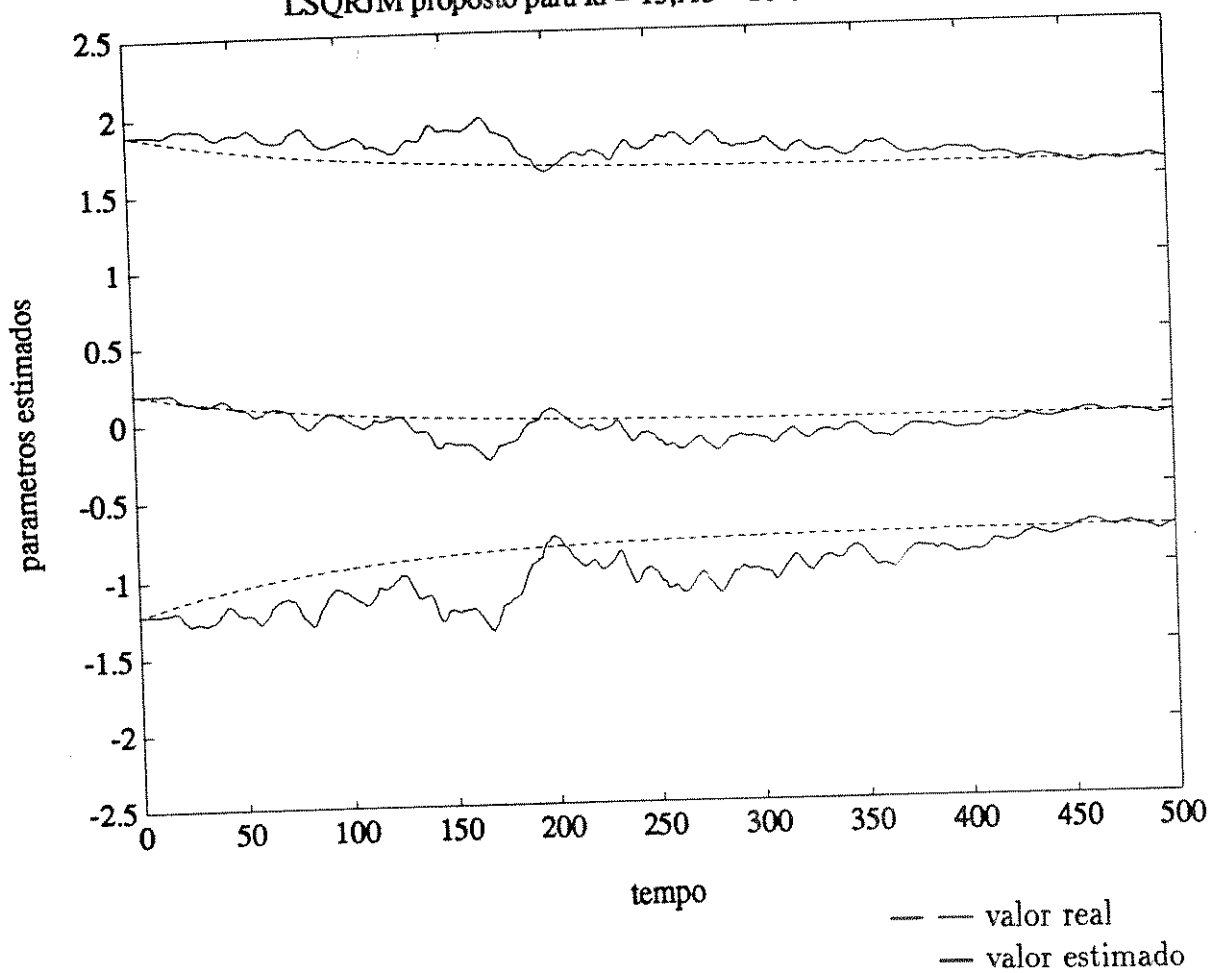
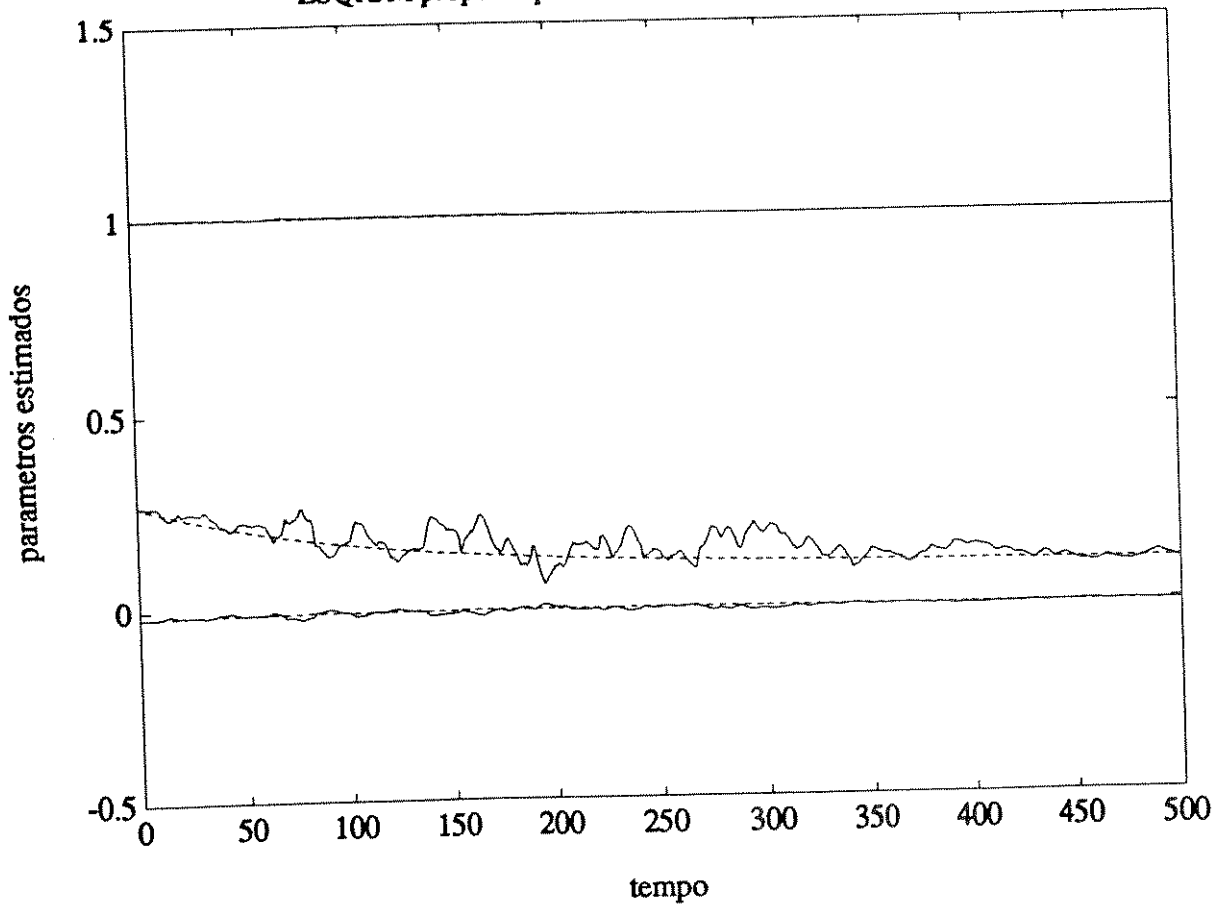


Figura 13a : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $l_d = 9, A_o = 10^{-4}.I_6$ e $\alpha = 0.05$



LSQRJM proposto para $l_d = 9, A_o = 10^{-4}.I_6$ e $\alpha = 0.05$

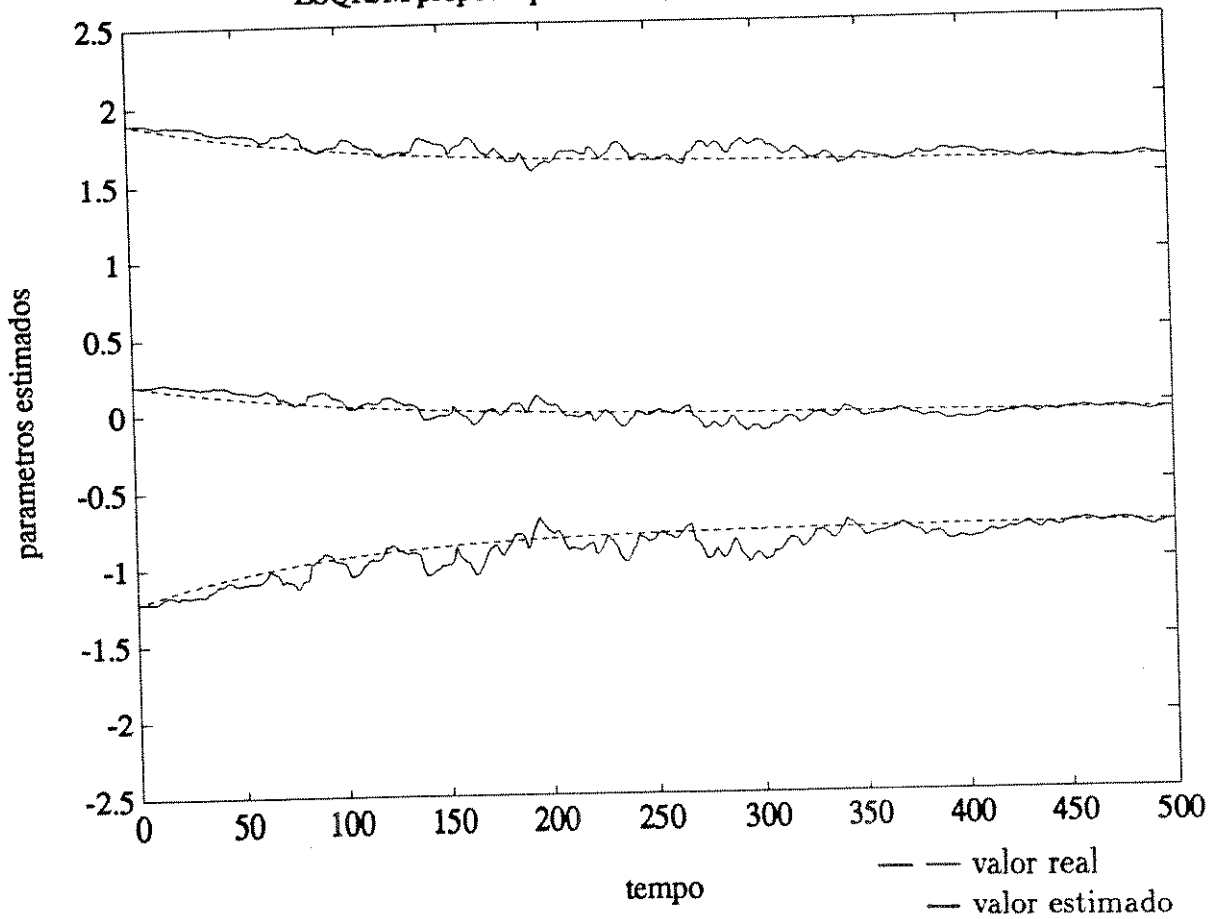
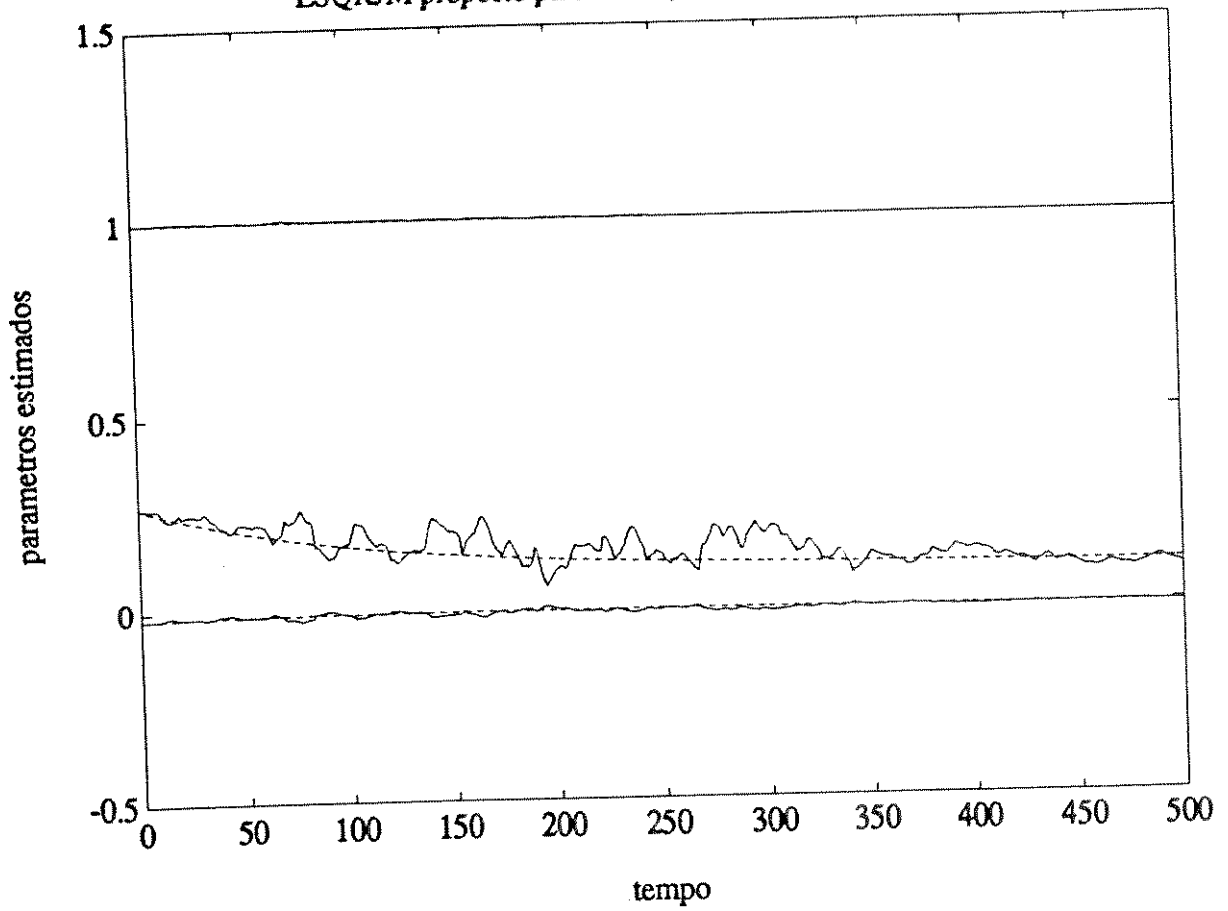


Figura 11b : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $l_d = 9, A_0 = 10^{-6} \cdot I_6$ e $\alpha = 0.05$



LSQRJM proposto para $l_d = 9, A_0 = 10^{-6} \cdot I_6$ e $\alpha = 0.05$

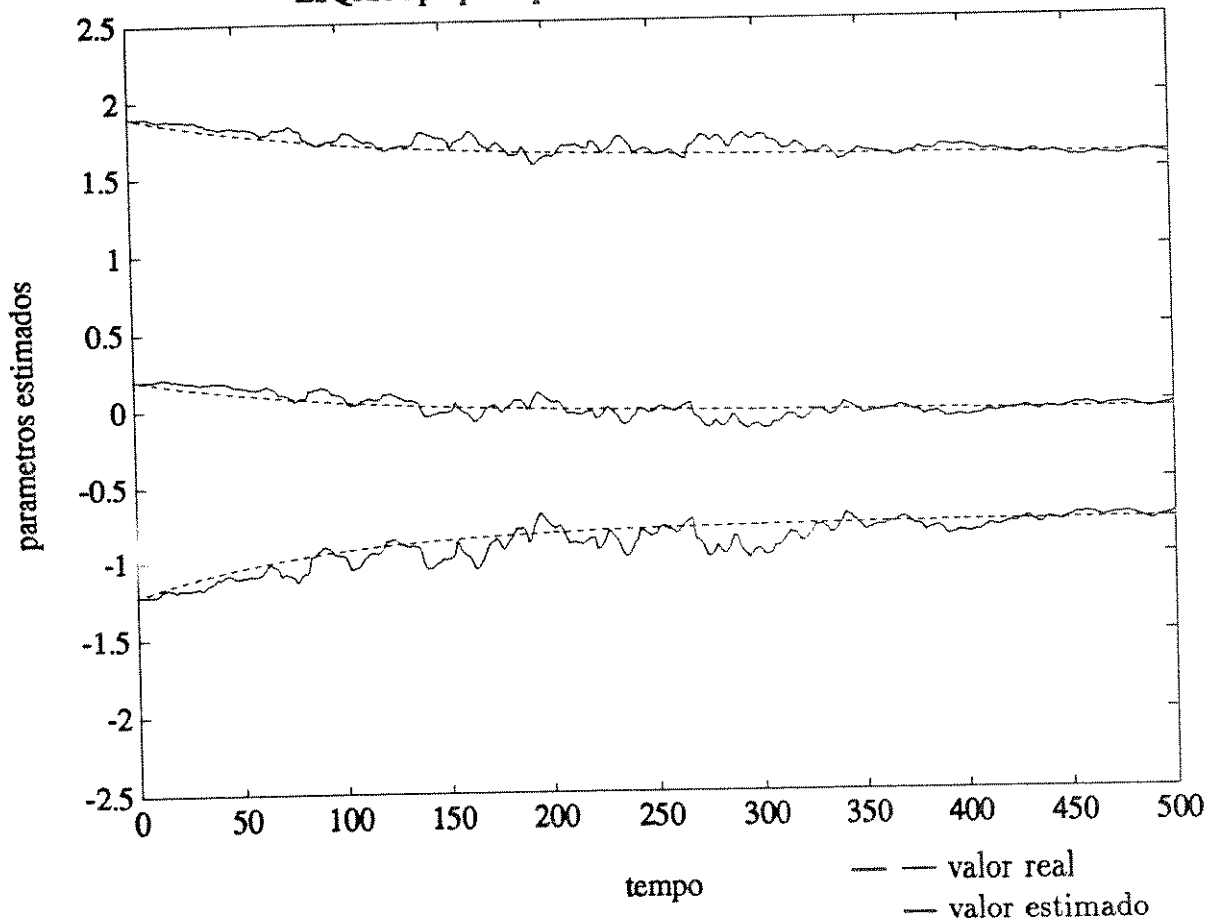
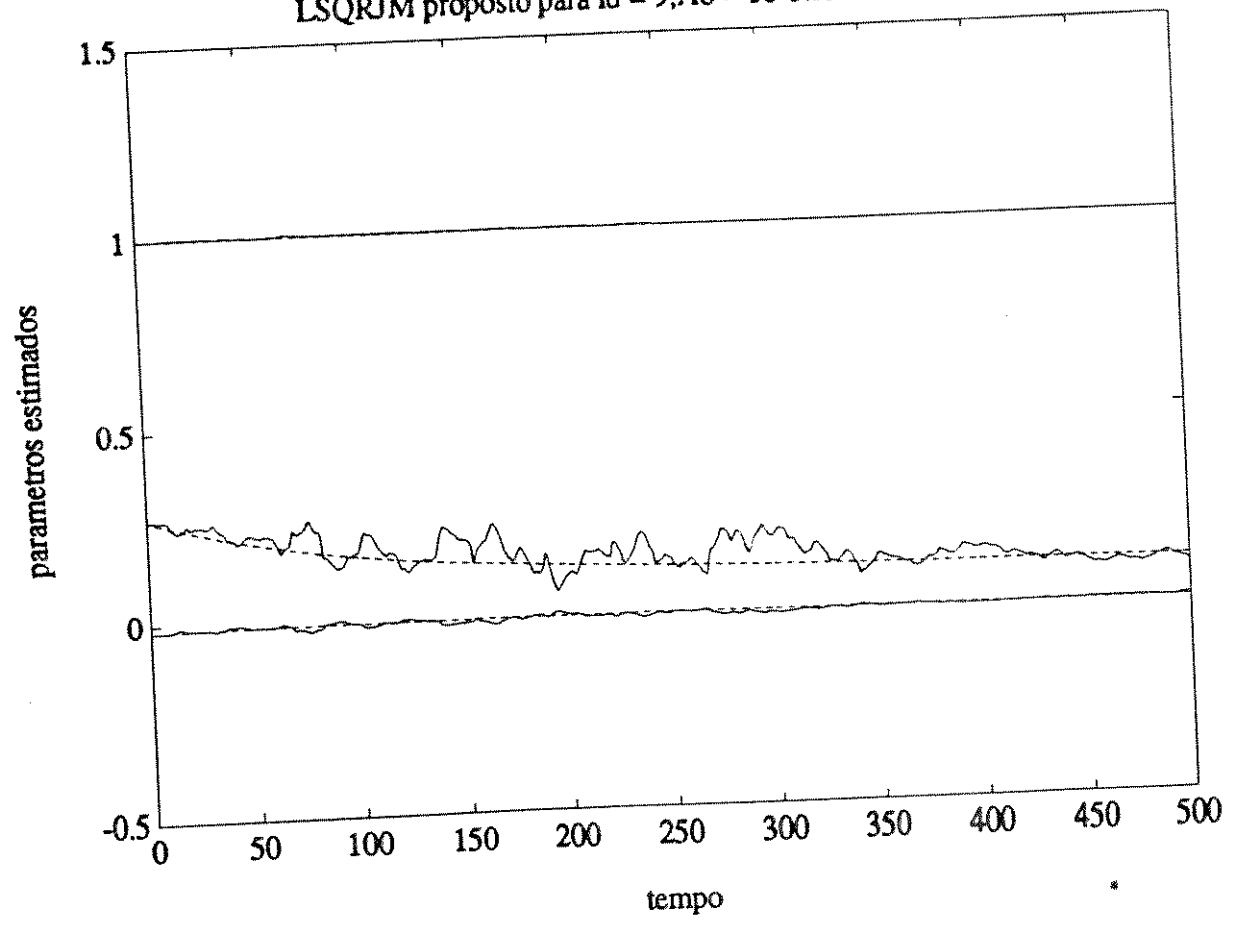


Figura 12b : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

LSQRJM proposto para $ld = 9, A_0 = 10^{-8} \cdot I_6$ e $\alpha = 0.05$



LSQRJM proposto para $ld = 9, A_0 = 10^{-8} \cdot I_6$ e $\alpha = 0.05$

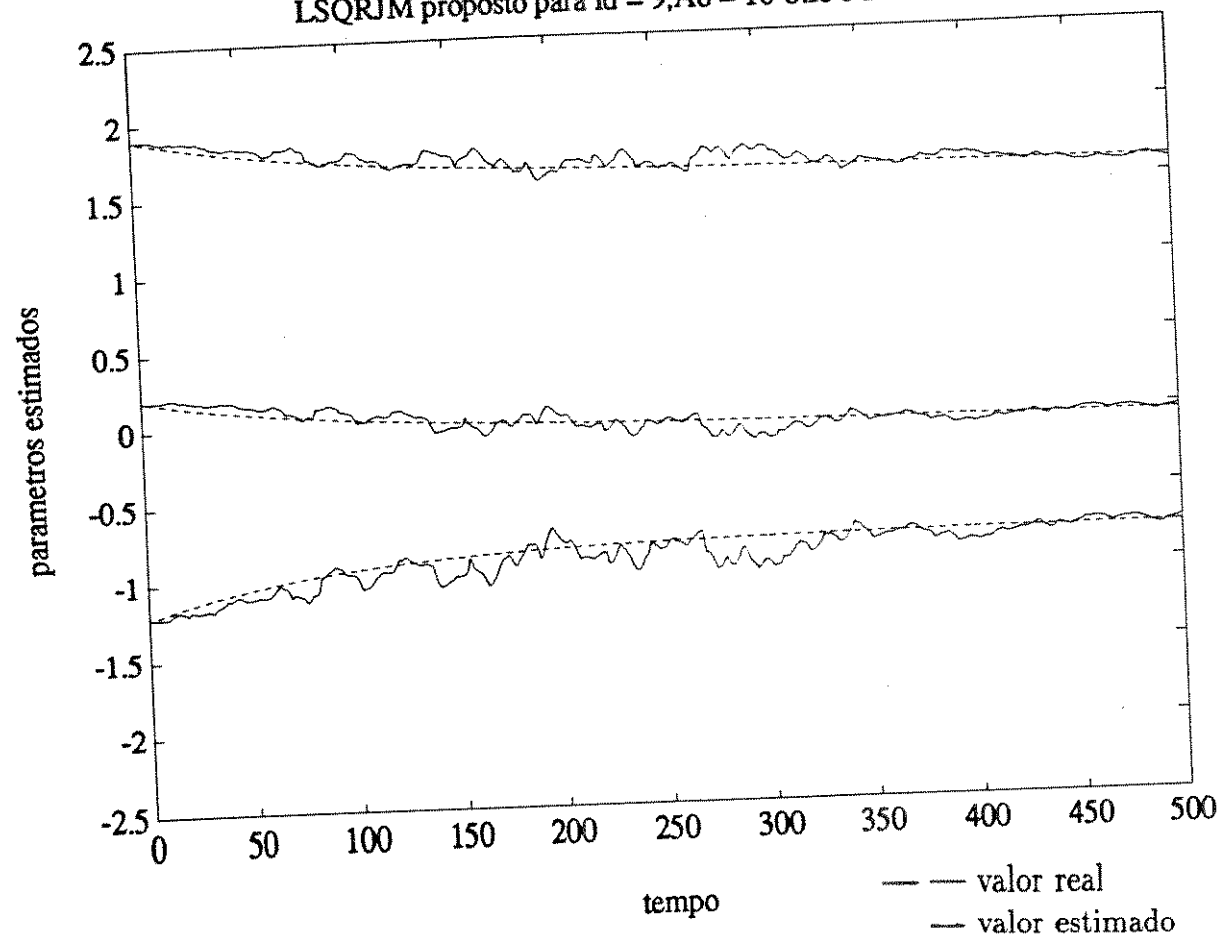


Figura 13b : Exemplo Numérico 1 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

4.2 exemplo numérico 2

Neste exemplo são utilizados dados gerados por (4.2) com p_i variantes no tempo conforme indicado na figura 14. Pode ser verificado nessa figura que p_5 inicialmente é próximo de p_2 , distanciando do mesmo ao longo do tempo, fazendo com que o problema de mínimos quadrados seja inicialmente mal condicionado, melhorando progressivamente ao longo do tempo, o seu condicionamento.

Nas figuras seguintes desta seção, são apresentados os melhores valores dos parâmetros h estimados pelos algoritmos LSNEPE, LSQRPE, LSNEJM e LSQRJM, juntamente com os valores exatos desses parâmetros utilizados pelo simulador.

Os resultados correspondentes ao LSNEPE estão apresentados na figura 15, onde pode ser visto que o algoritmo perde completamente a estabilidade numérica. O comportamento do LSNEPE neste exemplo é pior que no exemplo 1. Isto é devido à dificuldade de recuperação de erros anteriores, dificuldade esta dada à utilização do lema de inversão de matrizes, ver seção A4, na sua concepção

Os resultados correspondentes ao LSQRPE estão apresentados na figura 16, onde pode ser visto que são bem melhores que os resultados obtidos com LSNEPE, não apresentando nenhum problema de recuperação de erros. Isto é devido à utilização da fatoração ortogonal QR. Os resultados não são satisfatórios, devido à sua grande variação em torno dos valores dos parâmetros do sistema.

Os resultados correspondentes ao LSNEJM estão apresentados na figura 17. Esses resultados não são satisfatórios devido à polarização dos mesmos, consequência da dificuldade de recuperação de erros do passado pela utilização do lema de inversão de matrizes.

Os resultados correspondentes ao LSQRJM são apresentados na figura 18. Este algoritmo foi o que apresentou os melhores resultados, sem nenhum problema de estabilidade numérica, polarização ou recuperação de erros do passado.

Dos resultados obtidos no exemplo 1 e 2 com o algoritmo LSQRJM, vemos que o previsto nas considerações teóricas no capítulo 3, realmente são confirmadas pelos resultados numéricos apresentados neste capítulo.

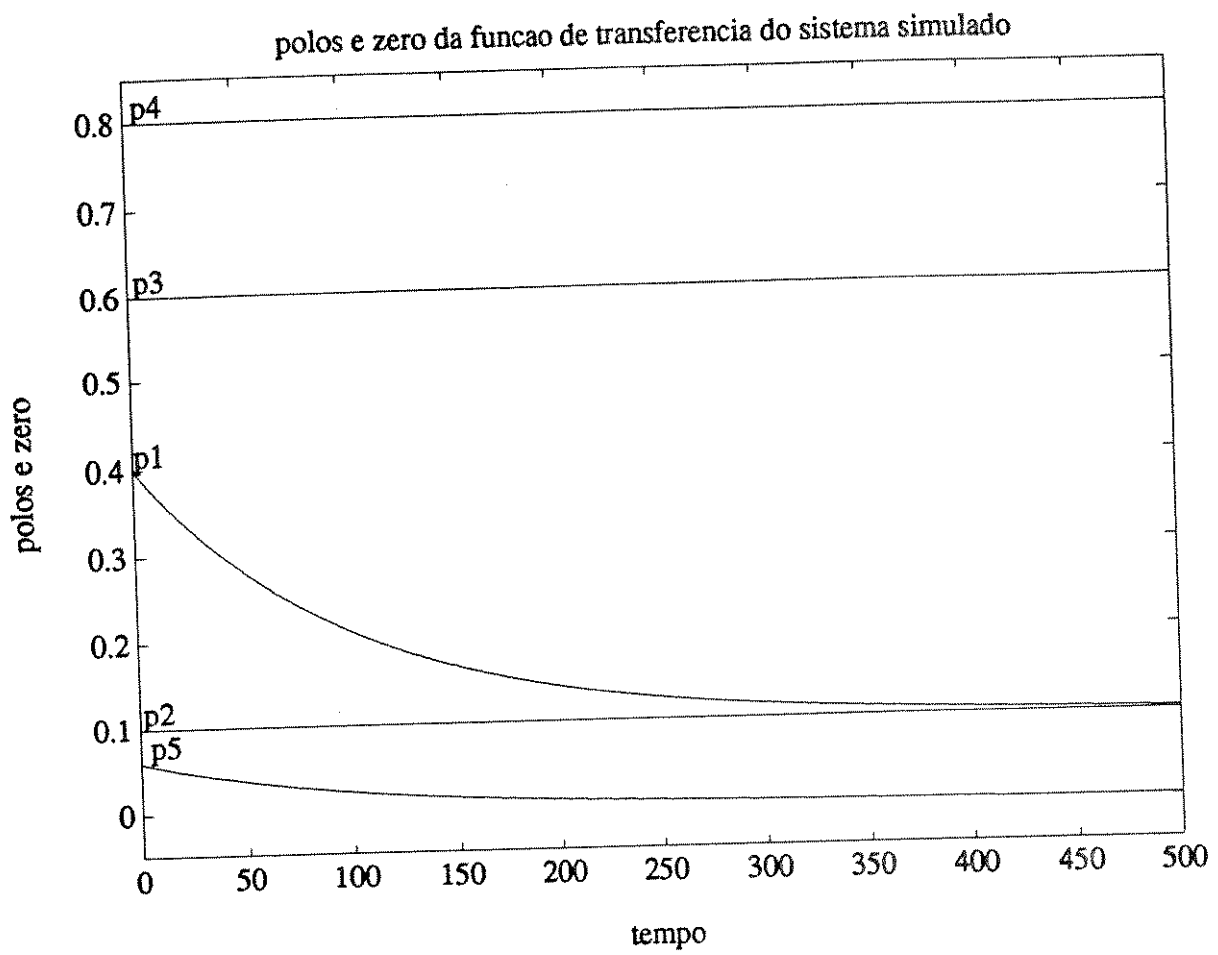
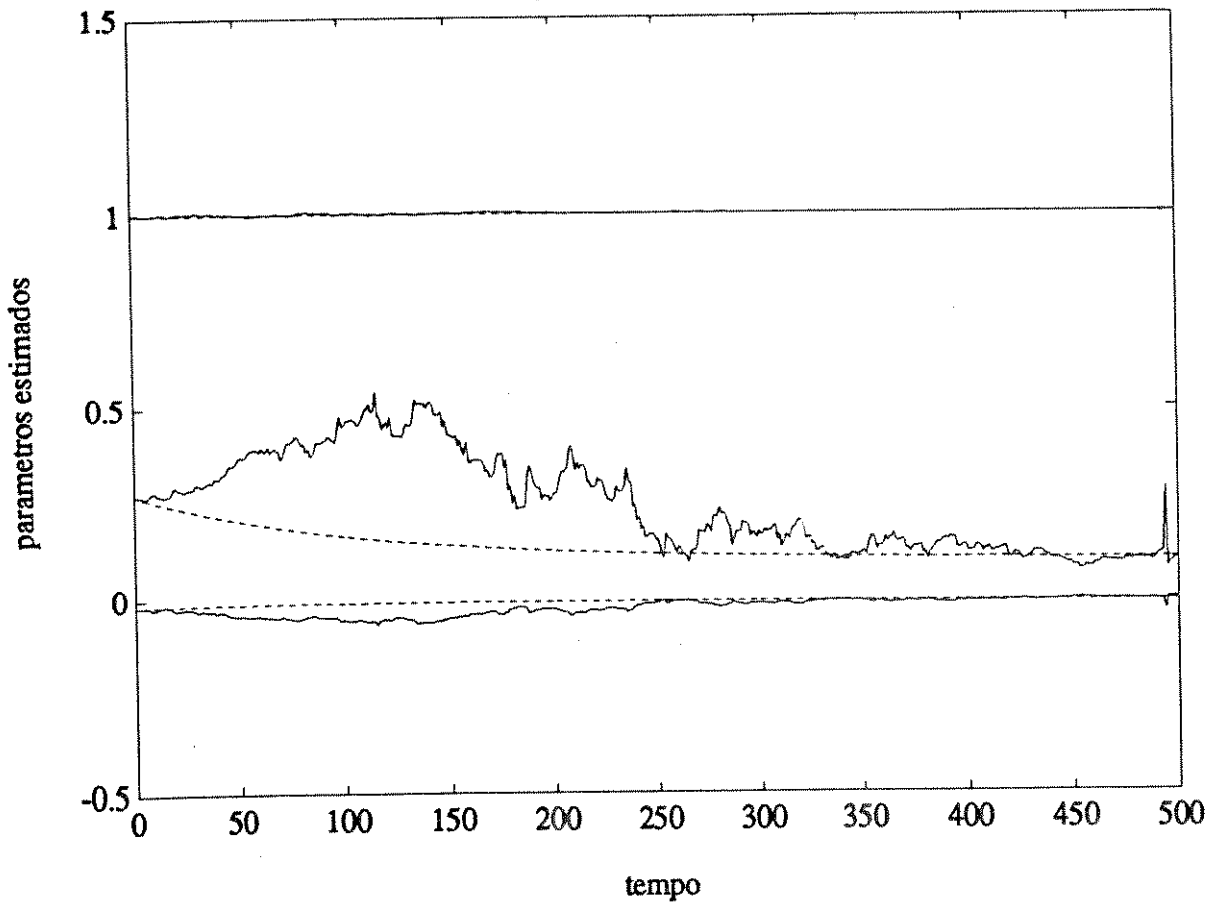


Figura 14 : Exemplo Numérico 2 - polos e zero da função de transferência do sistema simulado

LSNEPE para $Ao = 10^{-4}I6$ e $W = 0.9$



LSNEPE para $Ao = 10^{-4}I6$ e $W = 0.9$

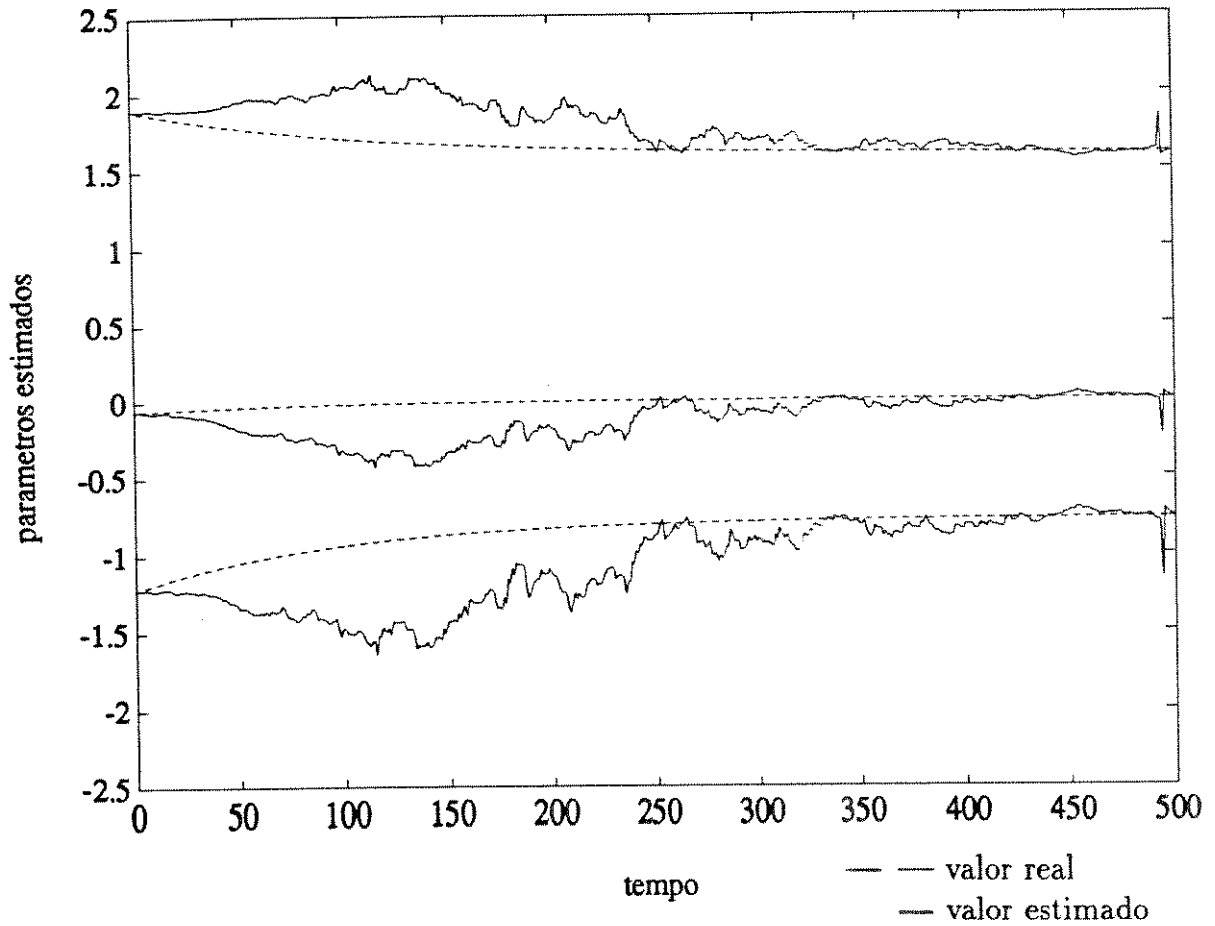
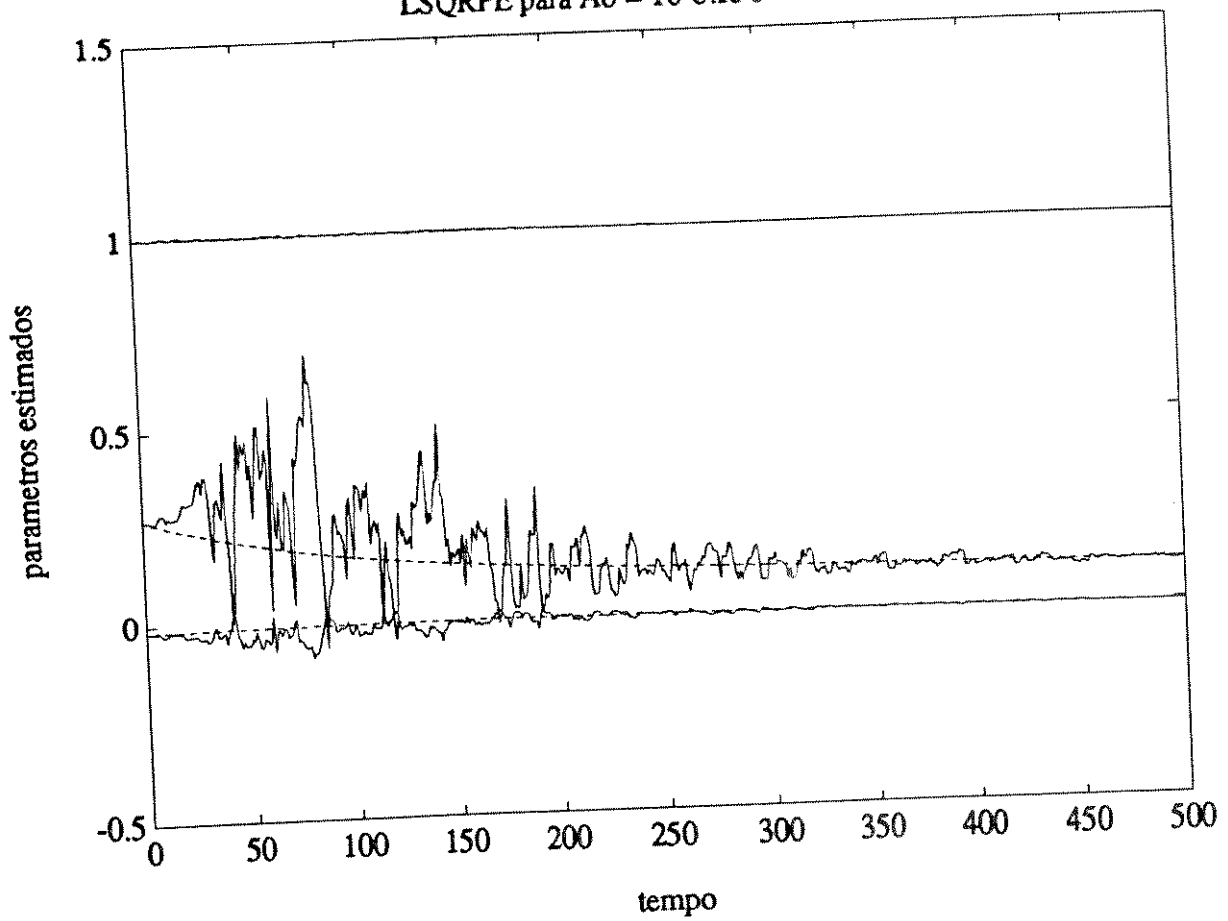


Figura 15 : Exemplo Numérico 2 - LSNEPE - parâmetros estimados e parâmetros do modelo de simulação

LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.75$



LSQRPE para $A_0 = 10^{-8.16}$ e $W = 0.75$

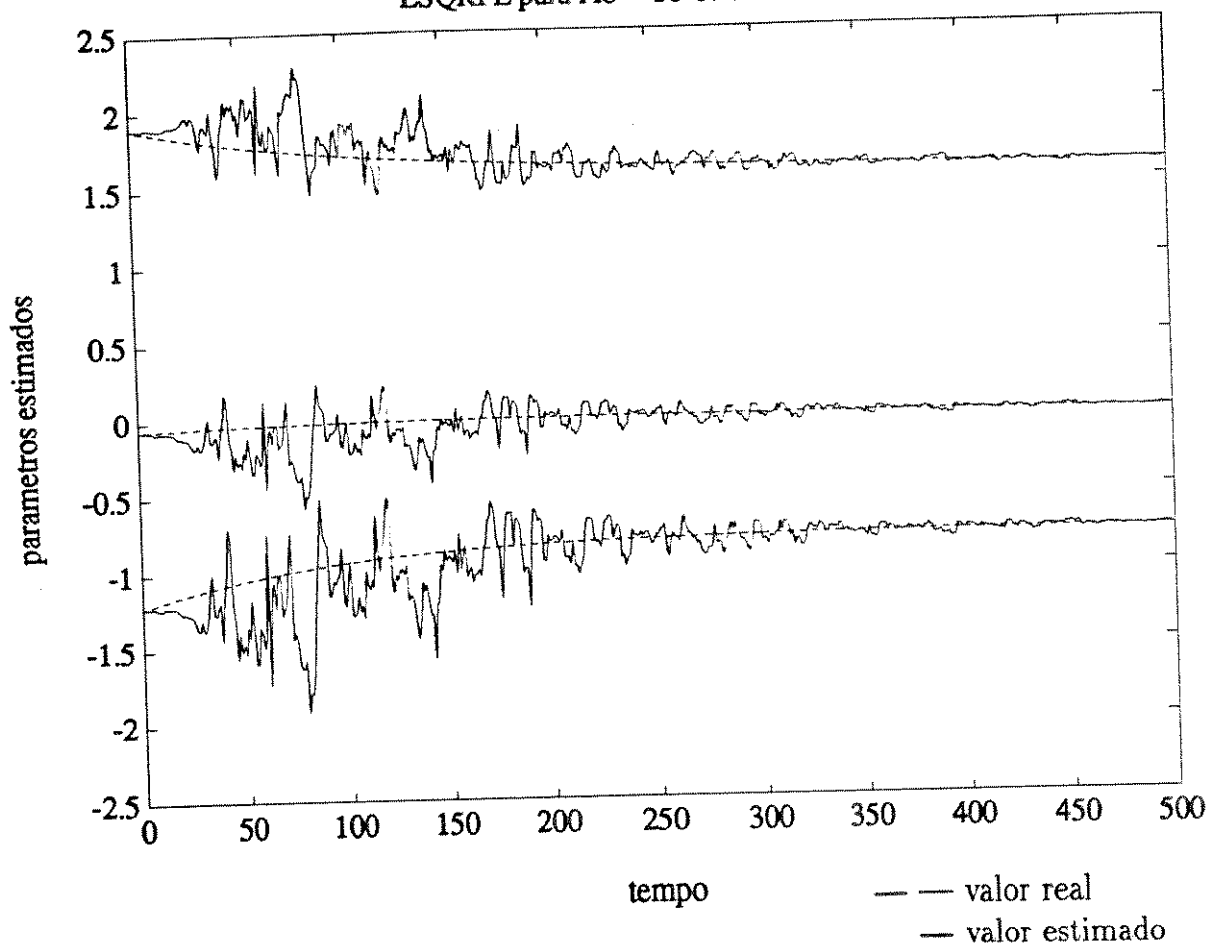
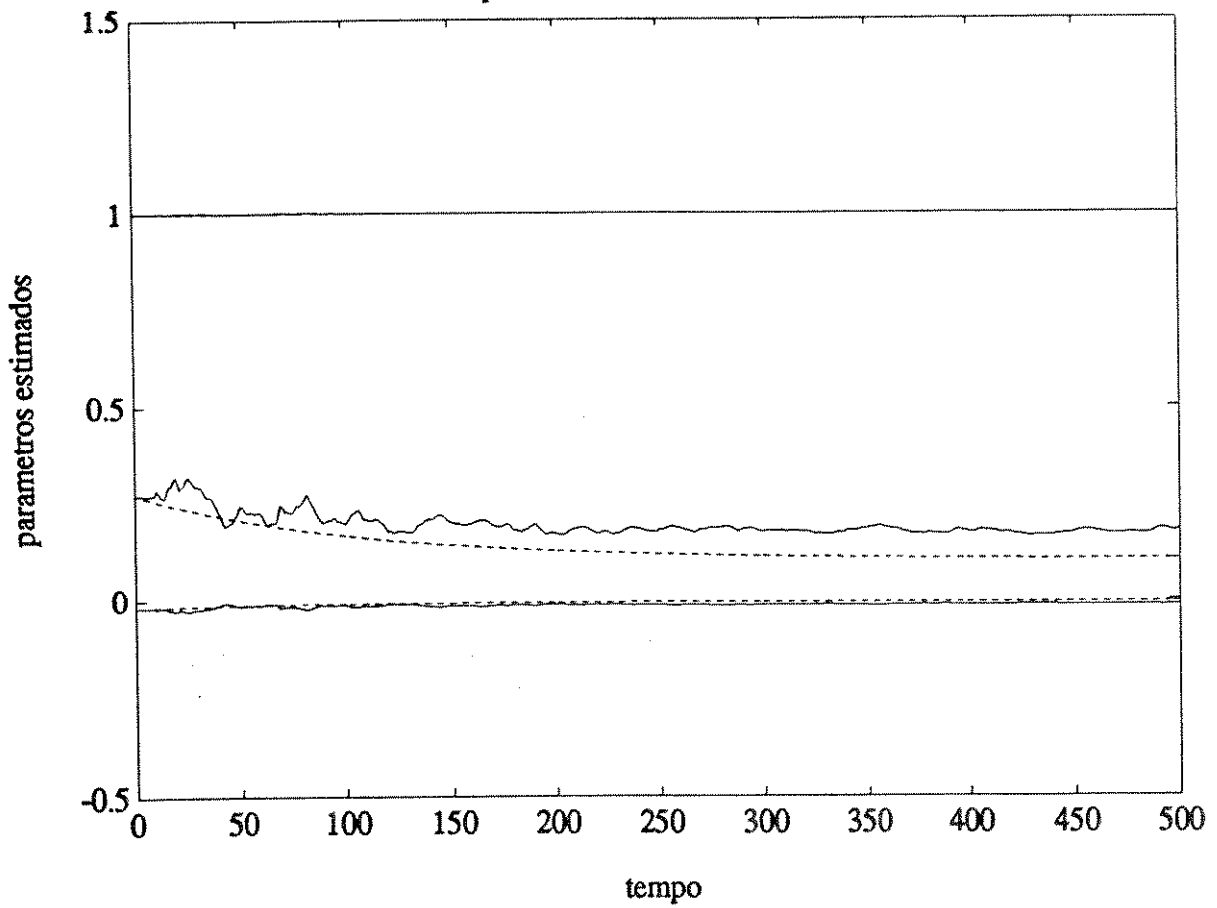


Figura 16 : Exemplo Numérico 2 - LSQRPE - parâmetros estimados e parâmetros do modelo de simulação

LSNEJM para $ld = 9$, $Ao = 10^{-6.16}$ e $\alpha = 0.05$



LSNEJM para $ld = 9$, $Ao = 10^{-6.16}$ e $\alpha = 0.05$

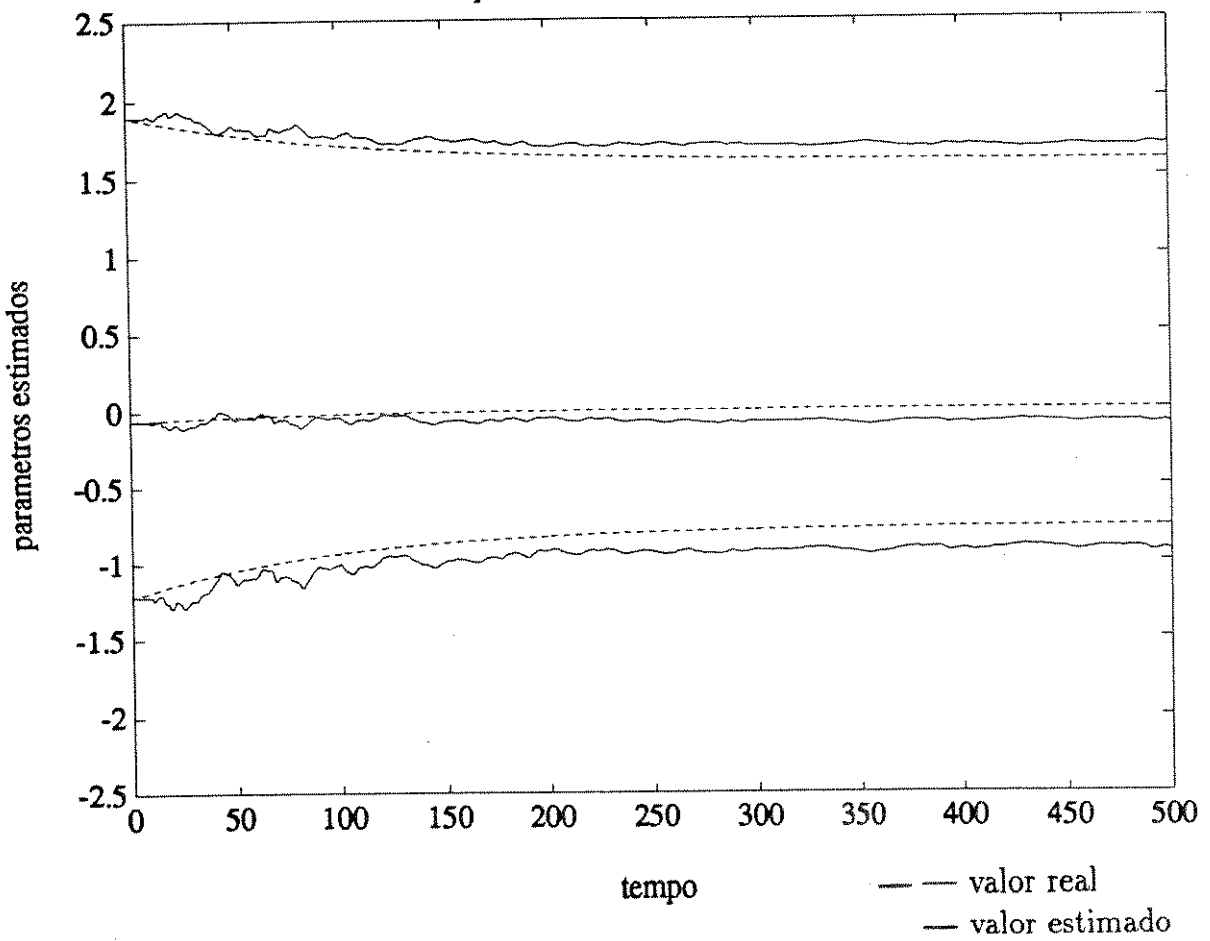


Figura 17 : Exemplo Numérico 2 - LSNEJM - parâmetros estimados e parâmetros do modelo de simulação

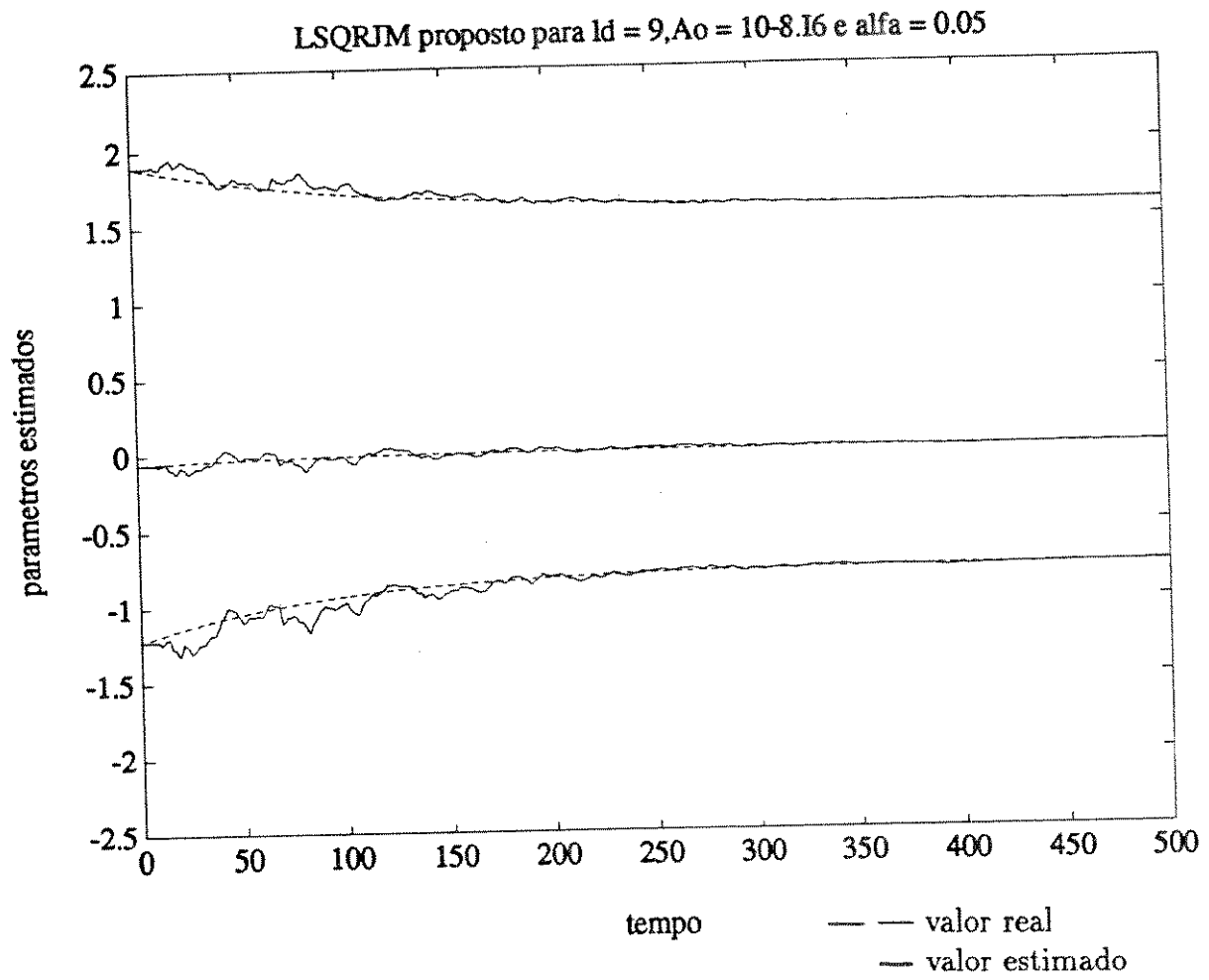
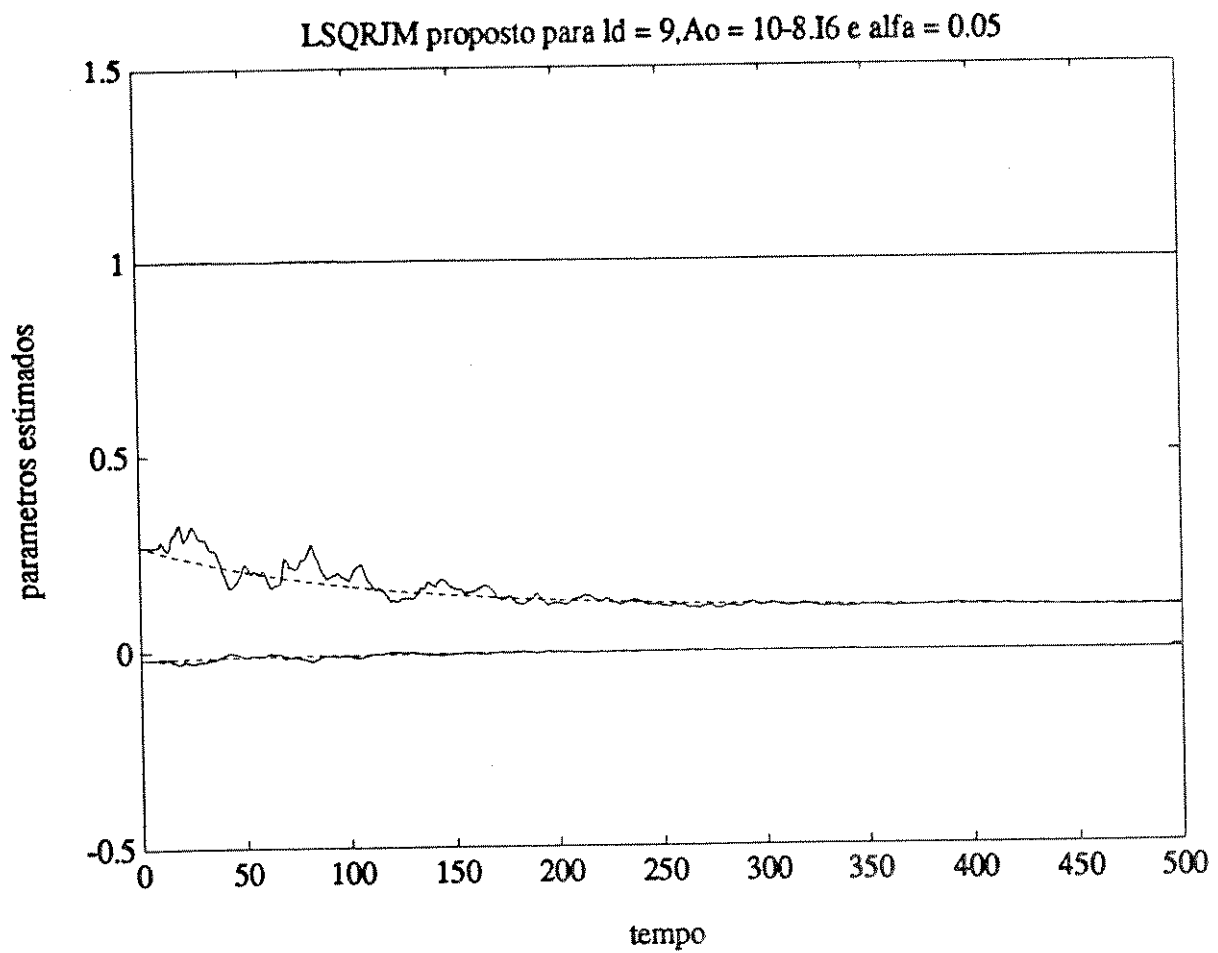


Figura 18 : Exemplo Numérico 2 - LSQRJM - parâmetros estimados e parâmetros do modelo de simulação

Capítulo 5

Conclusões Gerais

Neste trabalho foram apresentados e analisados algoritmos recursivos para identificação adaptativa de sistemas variantes no tempo através de mínimos quadrados lineares. Foram considerados algoritmos envolvendo as seguintes combinações : equações normais com ponderação exponencial de dados, fatoração QR com ponderação exponencial de dados, equações normais com janela móvel de dados e fatoração QR com janela móvel de dados.

Comparando os métodos utilizando Equações Normais e Transformações Ortogonais percebemos que :

— a principal vantagem utilizando equações normais são o seu baixo custo computacional por iteração e a simplicidade dos seus algoritmos.

— a principal vantagem da utilização das transformações ortogonais é o seu condicionamento numérico que é o mesmo do problema de mínimos quadrados original, proporcionando uma qualidade numérica excelente. Mas devido ao seu alto custo computacional é recomendado para problemas bem mal condicionados.

Quanto à utilização da ponderação exponencial e da janela móvel de dados percebemos que :

— os métodos que utilizaram ponderação exponencial de dados ficaram mais prejudicados, pois o uso dela piora o condicionamento do problema.

— os métodos que utilizaram janela móvel de dados obtiveram melhores resultados numéricos, pois o uso deste não faz nenhum efeito sobre o condicionamento do problema.

Os algoritmos com ponderação exponencial de dados e equações normais com

janela móvel de dados são facilmente encontrados na literatura especializada. O algoritmo combinando fatoração QR com janela móvel de dados, não é, do nosso conhecimento, disponível na literatura, tendo sido desenvolvido neste trabalho. Os resultados numéricos apresentados atestam bem a qualidade de seus resultados quando utilizado em problemas inerentemente mal condicionados.

O tratamento de problema de mínimos quadrados com restrições nos parâmetros e problemas de grande porte com matrizes esparsas são extensões futuras possíveis dos resultados aqui apresentados e discutidos.

Apêndice A

A.1 Transformações de Householder

Para um dado vetor $u \neq 0$ de dimensões $(n,1)$ a transformação de Householder ou a matriz de Householder é uma matriz da forma :

$$H = I - \frac{2uu^T}{\|u\|_2^2} = I - \frac{uu^T}{\beta} ; \beta = \frac{1}{2} \|u\|_2^2 \quad (\text{A.1})$$

onde o vetor u é conhecido como vetor de Householder.

As principais propriedades da matriz de Householder H [13] :

- i) são simétricas : $H = H^T$
- ii) são ortogonais : $H^T H = H H^T = I$
- iii) são involutórias : $H^2 = I$

Nos métodos de resolução de problemas de mínimos quadrados dados nos capítulos anteriores houve a necessidade da construção da matriz de transformação de Householder para transformar um dado vetor a de dimensões $(m,1)$ em um múltiplo de e_1 , a primeira coordenada do vetor. Desejamos portanto anular os

componentes abaixo da primeira coordenada, ou seja

$$H_1 a = I - \frac{u_1 u_1^T}{\beta_1} a = \|a\|_2 e_1 = \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{A.2})$$

onde $r_1 = \|a\|_2$.

Sabemos que u_1 é um múltiplo de $\|a\| e_1 - a$ e pelo fato de H_1 depender somente da direção de u_1 , u_1 pode ser dado como o seguinte vetor, que difere de a apenas no primeiro componente :

$$u_1 = a + \text{sign}(a_1) \|a\|_2 e_1$$

onde a_1 é a primeira componente do vetor a .

A.2 Transformações de Givens

Uma matriz J de Givens [13] de dimensões (n,n) tem a seguinte forma :

$$J_n(i, j, c, s) = \begin{bmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & c & & s & \\ & & & 1 & & \\ & & -s & & c & \\ & & & & & 1 \\ 0 & & & & & & 1 \end{bmatrix} \quad (\text{A.3})$$

sendo $c^2 + s^2 = 1$

Em termos trigonométricos, a matriz J_n se torna

$$J_n(i, j, \theta) = \begin{bmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & \cos\theta & & \text{sen}\theta & \\ & & & 1 & & \\ & & -\text{sen}\theta & & \cos\theta & \\ & & & & & 1 \\ 0 & & & & & & 1 \end{bmatrix} \quad (\text{A.4})$$

Geometricamente a matriz $J(i, j, \theta)$ corresponde a uma rotação θ dos eixos coordenados i, j .

Propriedades :

i) $J_n^T(i, j, \theta) = J(i, j, -\theta)$

ii) $J_n^T(i, j, \theta) \cdot J_n(i, j, \theta) = I$ (Ortogonal)

Para um vetor $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ multiplicado por uma matriz de Givens $J_n(i, j, \theta)$, apenas os componentes x_i e x_j de x são afetados.
Computacionalmente fazemos :

$$\begin{bmatrix} x_i^* \\ x_j^* \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix}$$

Em determinados casos, onde faz-se necessário a criação de zero em uma posição específica k de um vetor x , podemos utilizar uma matriz de Givens, com os seguintes parâmetros :

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x_i \\ x_k \end{bmatrix} = \begin{bmatrix} x^* \\ 0 \end{bmatrix}$$

onde $-sx_i + cx_k = 0$ $s^2 + c^2 = 1$
e portanto :

$$s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}} \quad c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}$$

A.3 Decomposição QR

Dada uma matriz A de dimensões (m,n) , $m \geq n$, $\text{posto}(A) = n$, aplicando uma sequência finita de matrizes ortogonais Q_i , capazes de zerar os elementos abaixo da diagonal de A , temos :

$$Q_r \dots Q_2 Q_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

onde R é uma matriz triangular superior de dimensões (n,n) .

As matrizes Q_i podem ser matrizes de Householder ou Givens, especificadas conforme as seções A1 e A2 respectivamente.

Assumindo

$$Q^T = Q_r \dots Q_2 Q_1$$

temos

$$Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$

onde Q^T , Q são ortogonais devido à ortogonalidade das matrizes Q_i .

Utilizando matrizes de Householder, o cálculo da matriz $\begin{bmatrix} R \\ 0 \end{bmatrix}$ requer $n^2(m - \frac{n}{3})$ FLOPS [13].

A.4 Lema de Inversão de matrizes

Sejam Z e S matrizes não singulares de dimensões (m,m) , Q uma matriz não singular de dimensões (i,i) , U uma matriz de dimensões (m,i) e V uma matriz de dimensões (i,m) tais que satisfaçam a equação

$$Z = S + UQV$$

Então a inversa da matriz Z é dada por :

$$Z^{-1} = (S + UQV)^{-1} = S^{-1} - S^{-1}U(VS^{-1}U + Q^{-1})^{-1}VS^{-1}$$

Se $U = V^T$, substituindo na última expressão :

$$Z^{-1} = S^{-1} - S^{-1}U(U^T S^{-1}U + Q^{-1})^{-1}U^T S^{-1}$$

Para o caso particular onde Q é um escalar, U um vetor de dimensões $(m,1)$ denotado por u e V um vetor de dimensões $(1,m)$ denotado por v^T , temos :

$$Z = S + uv^T$$

e

$$Z^{-1} = S^{-1} - \frac{(S^{-1}u)(v^T S^{-1})}{1 + v^T S^{-1}u}$$

Ainda, neste caso, se $u = v$

$$z = S + uu^T$$

e

$$Z^{-1} = S^{-1} - \frac{(S^{-1}u)(u^T S^{-1})}{1 + u^T S^{-1}u}$$

A.5 Atualização da Fatoração QR

Nesta seção é descrito um procedimento para atualização da fatoração QR de uma matriz A de dimensões $(n+1, n)$ e posto n para uma modificação de posto 1 dessa matriz.

Sejam

$$A = QR \text{ ou } Q^T A = R \quad (\text{A.5})$$

$$\bar{A} = A + uv \quad (\text{A.6})$$

onde : Q é uma matriz ortogonal de dimensões $(n+1, n+1)$, R é uma matriz triangular superior de dimensões $(n+1, n)$, $u \neq 0$ e $v \neq 0$ são vetores de dimensões $(n+1, 1)$ e $(1, n)$ respectivamente.

multiplicando (A.6) por Q^T , temos :

$$Q^T \bar{A} = Q^T A + Q^T uv = R + wv \quad (\text{A.7})$$

$$w = Q^T u \quad (\text{A.8})$$

$$\|w\|_2 = \|u\|_2 \quad (\text{A.9})$$

Seja S_1 uma matriz ortogonal de dimensões $(n+1, n+1)$, obtida do produto de uma sequência de n matrizes de Givens nos planos de rotação $(n+1, n)$, $(n+1, n-1)$..., $(n+1, 1)$, tal que

$$S_1 w = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \|w\|_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \|u\|_2 \end{bmatrix} \quad (\text{A.10})$$

Multiplicando (??) por S_1 , temos :

$$S_1 Q^T \bar{A} = S_1 R + S_1 wv \quad (\text{A.11})$$

$$R' = S_1 R \quad (\text{A.12})$$

$$M = S_1 wv = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \|u\|_2 \end{bmatrix} v \quad (\text{A.13})$$

$$Z = S_1 Q^T \bar{A} = R' + M \quad (\text{A.14})$$

Lembrando que as transformações de Givens afetam apenas os pares de linhas envolvidas, ver seção A2, pode ser verificado por inspeção que as estruturas das matrizes R' , M , Z de dimensões $(n+1, n)$ são as seguintes :

$$R' = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ x & x & x & x \end{bmatrix} \quad (\text{A.15})$$

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ x & x & x & x \end{bmatrix} \quad (\text{A.16})$$

$$Z = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ x & x & x & x \end{bmatrix} \quad (\text{A.17})$$

Seja S_2 uma matriz ortogonal de dimensões $(n+1, n+1)$, obtida do produto de uma sequência de n matrizes de Givens nos planos $(1, n+1)$, $(2, n+1)$, ..., $(n, n+1)$ tal que

$$S_2 Z = \bar{R} \quad (\text{A.18})$$

onde \bar{R} é uma matriz triangular superior de dimensões $(n+1, n)$.

De (A.11), (A.14) e (A.18), temos :

$$S_2 S_1 Q^T \bar{A} = \bar{R} \quad (\text{A.19})$$

Devido à ortogonalidade de S_2 , S_1 e Q , temos :

$$\bar{Q}^T \bar{A} = \bar{R} \text{ ou } \bar{A} = \bar{Q} \bar{R} \quad (\text{A.20})$$

$$\bar{Q}^T = S_2 S_1 Q^T \quad (\text{A.21})$$

onde \bar{Q} é uma matriz ortogonal de dimensões $(n+1, n+1)$ e \bar{R} uma matriz triangular superior de dimensões $(n+1, n)$, sendo portanto os fatores da fatoração QR de \bar{A} .

Referências Bibliográficas

- [1] Bierman, G. J. (1977), Factorization Methods for discrete Sequential Equations, Academic Press, New York.
- [2] Box, G. F. P. ; Jenkins, G. M. (1976) "Time Series Analysis Forecasting and Control".
- [3] Cioffi J. M.; T. Kailath (1984) "Fast Recursive Least Squares Transversal Filters for Adaptive Filtering". IEEE Transaction ASSP, vol. ASSP. 32 304-337.
- [4] George, A. and Heath, M. T. (1980) "Solution of Sparse Linear Least Squares using Givens rotations", Linear Algebra and its Applications, 34, pp. 69-83.
- [5] Gill, P. E. ; W. Murray e M. H. Wright (1991), " Numerical Linear Algebra and Optimization". Adison Wesley Publishing Co., Redwood City.
- [6] Goodwin, G. C. ; K. S. Sin (1984) "Adaptive Filtering Prediction and Control". Prentice Hall Inc., Englewood Cliffs, New Jersey, USA.
- [7] Houacine, A. (1990) "Regularized Fast Recursive Least Squares Algorithms" , Proceedings IEEE ICASSP 90 Conference, Albuquerque, New Mexico, USA, vol.3, 1587-1590.
- [8] Lawson, C. L. and Hanson, R. J. (1974) Solving Least Squares Problems, Prentice Hall, Englewood Cliffs, New Jersey.
- [9] Reichel, L. ; Gragg, W. B. (1990) "Algorithm 686, FORTRAN Subroutines for Updating the QR Decomposition" ACM Transactions on Mathematical Software, vol. 16, no. 4, 369-377.
- [10] Resende, L. S. (1991) "Filtragem Adaptativa com restrições lineares por mínimos quadrados", Dissertação de mestrado em engenharia elétrica, UNICAMP.
- [11] Sorenson, H. W. (1980) Parameter Estimation , principles and problems, Marcel DekkerInc., La Jolla, San Diego, California.

- [12] Taneguti, L. Y.; Milani, B. E. A. "Um algoritmo Utilizando Fatoração QR para Identificação Adaptativa de Sistemas através de Mínimos Quadrados com Janela Móvel de Dados" (1992) Anais do XXIV SBPO, SOBRAPO, Salvador, Bahia(a ser publicado).
- [13] Wilkinson, J. H. (1965) The Algebraic Eigenvalue Problem , Oxford University Press.