

ESTIMAÇÃO DE MODELO DE REGRESSÃO NÃO LINEAR COM PARÂMETROS DEPENDENTES DE REGRESSORES DE MAPEAMENTO

ELVIS J. ALEGRIA*, CELSO P. BOTTURA*

**Av. Albert Einstein 400, Campinas, SP, Brasil,
Departamento de Semicondutores Instrumentos e Fotônica,
Universidade Estadual de Campinas*

Emails: ejara@dsif.fee.unicamp.br, cpbottura@fee.unicamp.br

Abstract— This work proposes a new estimation method for autoregressive models with parameters that depend on mapping regressors. The behavior of this type of model is highly non-linear and even chaotic and cannot be estimated using conventional least squares based methods. The proposal consists of a special reordering of data followed by a process of data selection in subgroups, so that soft local models are obtained. To obtain these local models it is proposed to use windows that move along the mapping regressors. This method is quite simple and efficient compared to other methods in the literature. Finally, a simulated example of a system with 3 unknown parameters that vary rapidly is shown.

Keywords— Parameter estimation, nonlinear regression model, mapping regressor.

Resumo— Este trabalho propõe um novo método de estimação para modelos autorregressivos com parâmetros que dependem de regressores de mapeamento. O comportamento deste tipo de modelo é altamente não linear e inclusive caótico e não pode ser estimado utilizando métodos convencionais baseados em mínimos quadrados. A proposta consiste de um reordenamento especial de dados seguido de um processo de seleção de dados em subgrupos afins, de modo que modelos locais suaves são obtidos. Para obter estes modelos locais se propõe utilizar janelas que se movimentam ao longo dos regressores de mapeamento. Este método resulta bastante simples e eficiente comparado com outros métodos da literatura. Finalmente se mostra um exemplo simulado de estimação de um sistema com 3 parâmetros desconhecidos que variam rapidamente.

Palavras-chave— Estimação de parâmetros, modelo de regressão não linear, regressores de mapeamento.

1 Introdução

A particularidade do modelo autorregressivo com parâmetros que dependem de regressores de mapeamento (do inglês AutoRegressive with eXogenous input and mapping-Regressor Dependent Parameters ARX-RDP) (Priestley, 1988), é que os parâmetros do modelo são funções de mapeamento desconhecidas que dependem de sinais conhecidos, os quais neste artigo são chamados de regressores de mapeamento, que podem coincidir ou não com os regressores do modelo. Este tipo de modelo é altamente não linear e inclusive caótico e não pode ser estimado utilizando métodos convencionais baseados em mínimos quadrados, devido à rápida variação dos parâmetros. Neste trabalho vamos nos focar na estrutura autorregressiva e não vamos abordar o caso de estrutura em espaço de estado, embora a nossa proposta possa ser facilmente convertida a uma estrutura no espaço de estados utilizando uma transformação não mínima, como feito em (E. Shaban and Chotai, 2004).

Para estimar modelos ARX-RDP, atualmente existe o método computacional baseado na técnica de backfitting, proposta em (J. Sadeshi and Young, 2010) e muito utilizado para resolver problemas reais em distintas áreas como hidrologia e robótica, ver por exemplo (C. Taylor and Tych, 2007; Ochieng and Otieno, 2009) e (P. Smith and Young, 2014). A desvantagem desse método é o seu alto custo computacional e sua complexidade para implementar; estas limitações motivaram a

atual proposta, que resulta sendo um algoritmo bem mais rápido e que ademais facilitaria grandemente as versões on-line e MIMO do algoritmo; ver propostas anteriores que utilizam a técnica do backfitting em (Alegria and Bottura, 2015; Alegria and Bottura, 2016; Alegria and Bottura, 2017).

Basicamente, o método proposto consiste de um reordenamento especial dos dados, seguido de um processo de seleção de dados em subgrupos afins que geram modelos locais com parâmetros que variam lentamente. Para isto são utilizadas janelas que se movem ao longo dos regressores de mapeamento. Como etapa final é necessária uma parametrização de baixa complexidade que permitirá representar o modelo todo de uma forma analítica.

1.1 O modelo ARX-RDP

O modelo (ARX-RDP), na sua forma SISO, apresenta a seguinte forma:

$$y(k) = \mathbf{z}^T(k) \boldsymbol{\rho} \{\mathbf{x}(k)\} + e(k); \quad e(k) = \mathcal{N}(0, \sigma^2), \quad (1)$$

onde,

$$\begin{aligned} \mathbf{z}^T(k) &= \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) \\ u(k-\delta) & \cdots & u(k-\delta-m) \end{bmatrix} \\ \boldsymbol{\rho} \{\mathbf{x}(k)\} &= \begin{bmatrix} \rho_1 \{x_1(k)\} & \rho_2 \{x_2(k)\} & \cdots \\ \rho_{n+m+1} \{x_{n+m+1}(k)\} \end{bmatrix}^T. \end{aligned}$$

Este modelo se caracteriza por considerar dependências entre os parâmetros $\rho_i(k)$ e os regressores

de mapeamento $x_i(k)$, $k = 1, 2, \dots, N$. Este é denotado como $\rho_i \{x_i(k)\}$, $i = 1, \dots, n+m+1$. Consideramos que o vector de regressores de mapeamento $\mathbf{x}(k) = [x_1(k) \ x_2(k) \ \dots \ x_{n+m+1}(k)]$ é conhecido. O vector de regressão do modelo $\mathbf{z}(k)$ contém dados passados das saídas $y(k)$ e das entradas exógenas $u(k)$; δ é um retardo da entrada e $e(k)$ é um ruído branco de média zero. Note que este modelo sempre é não linear devido ao produto entre o regressor $z(k)$ e o parâmetro $\rho \{x(k)\}$.

1.2 Algoritmo recursivo de estimação baseado em mínimos quadrados

Aqui apresentamos um algoritmo recursivo convencional para estimação de parâmetros que variam lentamente em modelos ARX. Este algoritmo é baseado nos mínimos quadrados e recebe muitas outras denominações dependendo da área de aplicação; aqui vamos considerá-lo como sendo um caso particular do filtro de Kalman para modelos de regressão, como é feito em (Young, 2011).

$$\hat{\boldsymbol{\rho}}(k) = \hat{\boldsymbol{\rho}}(k-1) + \mathbf{g}(k) \{y(k) - \mathbf{z}^T(k) \hat{\boldsymbol{\rho}}_f(k-1)\} \quad (\text{KF1})$$

$$\mathbf{g}(k) = \mathbf{P}(k-1) \mathbf{z}(k) [\alpha + \mathbf{z}^T(k) \mathbf{P}(k-1) \mathbf{z}(k)]^{-1} \quad (\text{KF2})$$

$$\mathbf{P}(k) = \frac{1}{\alpha} \{ \mathbf{P}(k-1) - \mathbf{g}(k) \mathbf{z}^T(k) \mathbf{P}(k-1) \} \quad (\text{KF3})$$

Onde $\hat{\boldsymbol{\rho}}(k)$ é a estimação do parâmetro $\boldsymbol{\rho}(k)$, que depende da estimação num instante anterior $\hat{\boldsymbol{\rho}}(k-1)$, do ganho do filtro $\mathbf{g}(k)$ e da inovação $y(k) - \mathbf{z}^T(k) \hat{\boldsymbol{\rho}}_f(k-1)$. A matriz $\mathbf{P}(k)$ é a covariância do erro de estimação. Este algoritmo funciona sempre que os parâmetros variem lentamente, então ele não serve para estimar o modelo (1) de parâmetros rápidos. Mas, como mostraremos na seção 2, a nossa proposta consegue transformar o modelo global em modelos locais onde cada um deles têm parâmetros que variam lentamente, e onde o algoritmo (KF1)-(KF3) funciona.

2 Algoritmo de estimação do modelo ARX-SDP

Para simplificar a explicação do algoritmo, vamos considerar o seguinte caso particular de modelo ARX-RDP com dois parâmetros desconhecidos:

$$y = a(x_1)z_1 + b(x_2)z_2 \quad (2)$$

onde os parâmetros $a(x_1)$ e $b(x_2)$ são funções de mapeamento desconhecidas dependentes dos regressores de mapeamento x_1 e x_2 . A seguir mostramos os 4 passos que conformam a proposta deste trabalho.

2.1 Passo 1: Ordenamento dos dados

O processo de ordenamento dos sinais conhecidos a priori $y(k)$, $z(k)$ e $x(k)$ é realizado de acordo com a ordem ascendente do sinal $x(k)$. Esta transformação reduz a rápida variação do parâmetro do qual este depende, como é sugerido em (Young, 2011). Para representar o reordenamento de dados usaremos o símbolo $(^r)$. Como exemplo didático, consideremos o modelo ordenado de (1), considerando unicamente um parâmetro:

$$y^r(k) = z^r(k) \rho^r \{x^r(k)\} \quad (3)$$

Agora consideremos o seguinte vector de regressores de mapeamento $x(k) = [40 \ 10 \ 60 \ 20 \ 30]$ e a saída $y(k) = [y_1 \ y_2 \ y_3 \ y_4 \ y_5]$. Utilizando o comando do Matlab `sort` no regressor de mapeamento $x(k)$ como argumento obtemos o vector ascendente $x^r(k) = [10 \ 20 \ 30 \ 40 \ 60]$ o seu respectivo vector de índices $i_x = [2 \ 4 \ 5 \ 1 \ 3]$. Este vector de índices i_x é utilizado para mudar também a ordem de $y(k)$. Então a saída ordenada acorde ao vector de índices i_x é $z^r(k) = [y_2 \ y_4 \ y_5 \ y_1 \ y_3]$. O mesmo procedimento tem que se fazer para o vector de regressores do modelo $z(k)$.

Agora, utilizando este processo de ordenamento em (2) obtemos uma suavização similar à mostrada na Fig. 1 (esquerda). Devido à dependência paramétrica, a suavização obtida em x_1 implica na suavização do parâmetro $a(x_1)$, mas o parâmetro $b(x_1)$ ainda varia rapidamente mesmo que também reordenado em base aos índices i_x .

2.2 Passo 2: Separação em subgrupos para gerar os modelos locais suavizados

Aqui são selecionados os subgrupos de dados convenientemente usando uma janela de dimensão r e um tamanho de passo s o qual se move a través do regressor de mapeamento x_2 . Aqui r é o suficientemente pequeno para que x_2 seja considerado sinal estacionário, Dessa forma o parâmetro $b(x_2)$ seria também estacionário. Em outras palavras, selecionamos as amostras de x_2 de modo que $x_{2s} - \delta < x_2(t^*) < x_{2s} + \delta$, onde $\delta > 0$ e x_{2s} é fixo em cada posição da janela, ver Fig. 1 (direita). Com isto, pode-se calcular facilmente o numero total de subgrupos s_{max} pode se calcular facilmente assim:

$$s_{max} = \frac{\max(x_2) - \min(x_2)}{2\delta} \quad (4)$$

2.3 Passo 3: Filtro de Kalman para modelos de regressão

O algoritmo processa s_{max} iterações, onde em cada iteração se obtém uma estimação do parâmetro em $a\{x_1\}$, $\min(x_1) < x_1 < \max(x_1)$ e uma

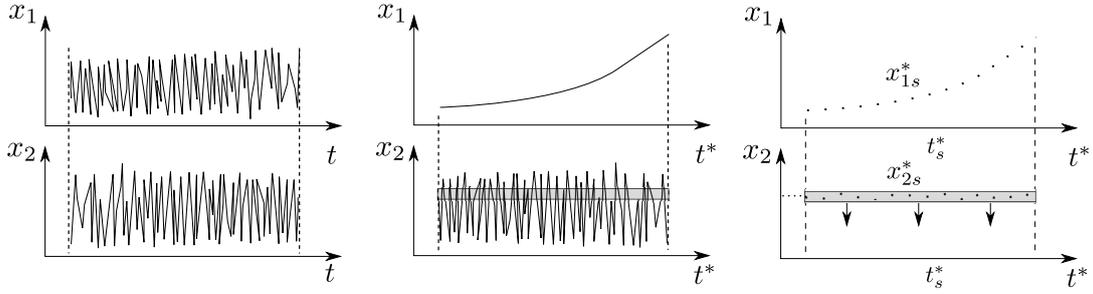


Figura 1: Suavização do estado x_1 e reordenamento do x_2 acorde aos índices de reordenamento de x_1 (esquerda). Subgrupo de dados onde $a(x_1)$ é suavizado e $b(x_2)$ é quase constante (direita).

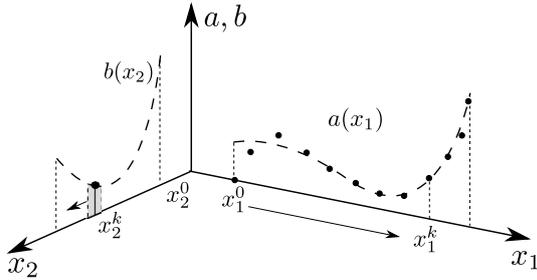


Figura 2: Representação gráfica do resultado de uma iteração k

estimação local do parâmetro $b\{x_2\}, x_2 = x_{2s}$. Então o parâmetro $b(x_2)$ fica evidente por partes. Então, são obtidos s_{max} estimações, uma para cada modelo local suavizado onde $b(x_2)$ é estacionário para cada um. A Fig. 2 mostra a estimação do resultado de uma iteração e também o resultado de estimação quando a janela esta se movimentando de x_2^{min} até x_2^{max} .

Inicialização do Filtro de Kalman

Devido a que cada subgrupo contém um pequeno grupo de dados, uma correta inicialização do filtro de Kalman é muito importante para garantir a convergência da estimação. Neste trabalho recomendamos inicializar os parâmetros desconhecidos $a(x_1)$ e $b(x_2)$ com o resultado da estimação de uma iteração prévia, ou seja o resultado de uma posição anterior da janela. Obviamente tem que se fazer o mesmo com a matriz de covariância do erro $P(k)$.

2.4 Passo 4: Parametrização

Até o passo 3 obtivemos as estimações dos parâmetros como funções de mapeamentos, ou pares coordenados, que dependem dos seus correspondentes regressores de mapeamento, mas estes ainda não estão representados de forma analítica; para isto precisamos da etapa final de parametrização. Existem muitos métodos para isto na literatura, por exemplo: *support vector regression*, *radial basis function* ou *simple interpolation*

(Haykin, 2009). A forma geral de parametrização pode se expressar pela seguinte expressão:

$$a(x_1) = \omega_1^T \phi_1, \quad (5)$$

$$b(x_2) = \omega_2^T \phi_2, \quad (6)$$

onde ϕ é um vetor que contém funções candidatas fixas e ω é um vector de pesos que precisam ser calculado neste passo. Uma diferença interessante com o método de estimação de modelos ARX-RDP proposto em (Young, 2011), onde a parametrização é necessariamente a etapa final, é que o método aqui proposto permite um ajuste dos parâmetros ω em modo on-line, ou seja podem ser ajustados em cada iteração.

2.5 Custo computacional desta proposta

Uma das principais vantagens desta proposta é o baixo custo computacional. Diferentemente da proposta apresentada em (Young, 2011) e (Alegria and Bottura, 2015), a nossa proposta executa o filtro de Kalman s_{max} vezes para subgrupos reduzidos de dados, ou seja um pequeno grupo de dados. Então o custo computacional depende da quantidade de passos da janela e do tamanho dela. Esta vantagem computacional é importante para implementar uma futura versão OnLine desta proposta, que seria bem mais rápida da que apresentamos recentemente em (Alegria and Bottura, 2017), assim também para identificar a estrutura de um modelo ARX-RDP no caso de caixa-preta, como fizemos em (Alegria and Bottura, 2016).

3 Exemplos

Considere o seguinte modelo ARX-RDP:

$$y(k) = \rho_1 \{x_1(k)\} u_1(k-1) + \rho_2 \{x_2(k)\} u_2(k-1) + \rho_3 \{x_3(k)\} u_3(k-1) + e(k), \quad (7)$$

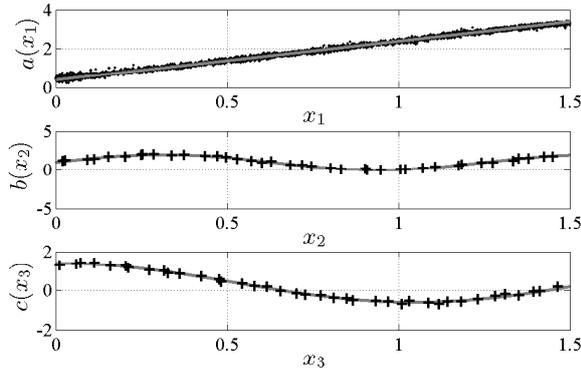


Figura 3: Resultado da estimação dos parâmetros $a(x_1)$, $b(x_2)$ y $c(x_3)$ (marcas pretas) e referências do benchmark (linhas cinzas)

onde:

$$\rho_1 \{x_1(k)\} = 0.8x_1(k-1) + 3 \quad (8)$$

$$\rho_2 \{x_2(k)\} = x_2^2(k-1) + 1 \quad (9)$$

$$\rho_3 \{x_3(k)\} = \sin\left(\frac{\pi}{2}x_3(k-1)\right) + 2 \quad (10)$$

os sinais x_i ; $i = 1, 2, 3$, são simulados como ruídos com distribuição homogênea e média zero, e as entradas u_i ; $i = 1, 2, 3$, são simuladas como ruídos brancos com médias em 1, -1 y 2, respectivamente. Também um ruído branco de média zero $e(k)$; $k = 1, 2, \dots, N$; $N = 1500$ amostras é considerado.

As funções de dependência entre os parâmetros e os sinais (8-10) são consideradas desconhecidas e somente vão ser utilizadas para gerar o benchmark. A Fig. 3 mostra a dependência entre o parâmetro e o estado estimado utilizando o algoritmo apresentado na seção 2.

3.1 Teste com poucos dados

Devido a que cada subgrupo contém um número reduzido de dados, consideramos importante fazer um experimento reduzindo o numero de amostras neste exemplo para ver o efeito nos resultados. Experimentalmente concluímos que se o filtro de Kalman for bem inicializado, utilizando como condição inicial o resultado de uma iteração anterior, o método acaba precisando de poucos dados para convergir.

Até onde sabemos, como alternativa de solução somente existe o método baseado em Backfitting (Young, 2011), que estima os parâmetros dos modelos ARX-RDP e encontra-se implementado no Toolbox CAPTAIN do Matlab (C. Taylor and Tych, 2007). Por isso utilizamos este algoritmo para comparar os nossos resultados. A principal vantagem da nossa proposta, demonstrada experimentalmente é o baixo custo computacional.

A Fig. 4 mostra o resultado da estimação utilizando unicamente 600 amostras e também as re-

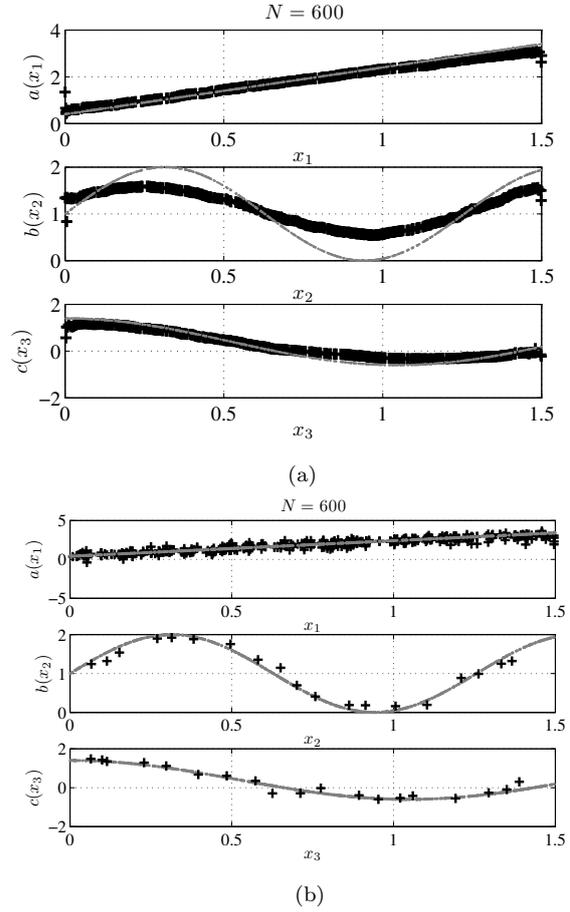


Figura 4: Estimações com poucos dados baseadas em backfitting (a) e usando a proposta deste trabalho (b)

ferências cinza. Podemos notar que a nossa proposta estima satisfatoriamente apesar da redução no número de dados, mas quando utilizamos o método baseado no backfitting ocorre uma distorção grave nas estimações.

A implementação dos nossos algoritmos para estimação de modelos ARX-RDP, utilizando o método de backfitting e incluindo as melhoras que propusemos recentemente em (Alegria and Bottura, 2015; Alegria and Bottura, 2016; Alegria and Bottura, 2017), e utilizando a proposta atual deste trabalho, estão implementados no Toolbox INCA (Identificação de sistemas Não-lineares para Controle Automático) que desenvolvemos para Matlab. Todos os resultados e gráficos mostrados neste trabalho foram obtidos utilizando o Toolbox INCA.

Deste experimento concluímos que o primeiro parâmetro falha mais do que o segundo; isto pode ser devido a que a função de dependência entre o parâmetro e o regressor de mapeamento é menos suavizada. Podemos então dizer que o método aqui proposto é mais flexível à funções de dependência desconhecidas.

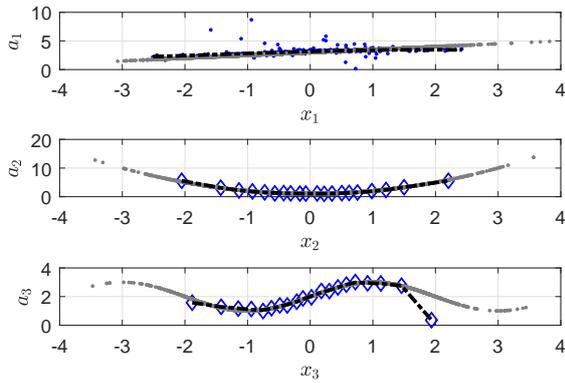


Figura 5: Resultado das dependências entre os parâmetro e os regressores de mapeamento $a(x_1)$, $b(x_2)$ e $c(x_3)$ com distribuição gaussiana (marcas azuis) e referências do benchmark (marcas cinzas)

3.2 Regressores de mapeamento com distribuição Gaussiana

No exemplo anterior os regressores de mapeamento tinham distribuição homogênea, isto facilitou muito a definição das janelas retangulares pois todas elas podiam ser da mesma dimensão de modo a conter a mesma quantidade de dados. Já no caso de uma distribuição Gaussiana é necessário algum método para definir os limites das janelas de modo que cada uma contenha a mesma quantidade de dados. Neste trabalho vamos aproveitar que os regressores de mapeamento são conhecidos a priori, assim como a quantidade total de amostras. Com isto simplesmente utilizaremos um contador de amostras desejadas para cada janela como: $N_s = N/s_{max}$, onde N é a quantidade de amostras totais e s_{max} é a quantidade de sub-grupos desejados.

Então, considerando $s_{max} = 20$ amostras por janela, obtemos as estimações mostradas na Fig. 5. Para validar estes resultados foram simulados outros 50 dados gaussianos; os resultados de validação para os parâmetros estimados e a saída estimada se mostram nas figuras 6 e 7, respectivamente. Destes resultados podemos concluir que o algoritmo proposto consegue resolver o problema de estimar parâmetros em modelos ARX-RDP bem melhor do que o método convencional resumido nas equações (KF1)-(KF3). O tempo de execução deste exemplo foi de 1 segundo aproximadamente, utilizando um computador padrão.

4 Conclusão

Neste trabalho propusemos um método original para estimar modelos com estrutura ARX-RDP. Esta proposta resolve o problema de estimar os parâmetros do modelo, altamente não linear, de uma forma simples e com baixo custo computacional. Os passos que conformam este método são: Ordenamento de dados, aplicação de janelamento

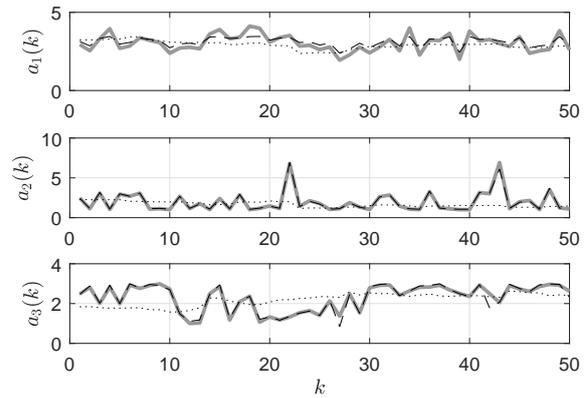


Figura 6: Resultados de validação: Parâmetros reais (linha cinza), parâmetros estimados (linha tracejada) e estimação utilizando o método convencional do KF1-KF3 (linha pontilhada)

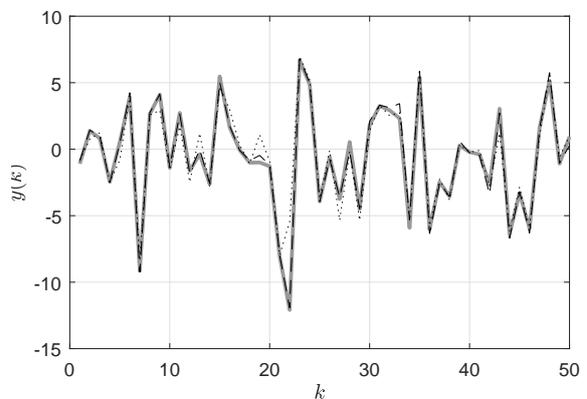


Figura 7: Resultados de validação: Saída real (linha cinza), saída estimada (linha tracejada) e saída estimada utilizando o método convencional do KF1-KF3 (linha pontilhada)

ao longo dos regressores de mapeamento, filtragem do tipo Kalman recursivo e parametrização. Por simplicidade foi explicado detalhadamente o algoritmo utilizando um modelo de somente 2 parâmetros desconhecidos mas a proposta pode se aplicar a estruturas com mais parâmetros, como foi mostrado no exemplo. Experimentalmente mostramos que, além da nossa proposta conseguir estimar satisfatoriamente e com um custo computacional baixo, ela precisa de menos dados do que outros métodos baseados em backfitting (Young, 2011; Alegria and Bottura, 2015).

Referências

- Alegria, E. and Bottura, C. (2015). Off-line state-dependent parameter models identification using simple fixed interval smoothing, *12th International Conference on Informatics in Control, Automation and Robotics ICINCO*.
- Alegria, E. and Bottura, C. (2016). MIMO ARX-SDP model estimation proposal and applica-

- tion to a hydraulic system., *IEEE Conference on Control Applications CCA* .
- Alegria, E. and Bottura, C. (2017). Real-time identification of ARX-SDP model using a rectangular moving window, *23rd International Conference on Automation and Computing (ICAC)* .
- C. Taylor, D. Pedregal, P. Y. and Tych, W. (2007). Environmental time series analysis and forecasting with the CAPTAIN toolbox., *Environmental Modelling & Software*, 22(6), pp. 797–814 .
- E. Shaban, C. T. and Chotai, A. (2004). State dependent parameter proportional-integral-plus (SDP-PIP) control of a nonlinear robot digger arm, *Control 2004, University of Bath, UK* .
- Haykin, S. (2009). *Neural networks and learning machines*, Upper Saddle River, NJ.
- J. Sadeshi, W. Tych, A. C. and Young, P. (2010). Multi-state dependent parameter model identification and estimation for nonlinear dynamic systems, *IEEE Electronics Letters*, Vol 46, number 18, pp. 1265-1266 .
- Ochieng, G. and Otieno, F. (2009). Data-based mechanistic modelling of stochastic rainfall-flow processes by state dependent parameter estimation, *Environmental Modelling & Software* 24 .
- P. Smith, K. Beven, D. L. A. W. and Young, P. (2014). Testing probabilistic adaptive on-line flood forecasting models, *Journal of flood risk management*, 7, pp. 265–279 .
- Priestley, M. (1988). *Non-linear and non-stationary time series analysis.*, Academic Press, London.
- Young, P. (2011). *Recursive Estimation and Time-Series Analysis*, Springer-Verlag, 2 edition.