Uma Rede Regularizada Esparsa e Robusta para Identificação Recursiva de Sistemas Dinâmicos

Michael S. Duarte and Guilherme A. Barreto
Graduate Program in Teleinformatics Engineering, Federal University of Ceará
Center of Technology, Campus of Pici, Fortaleza, Ceará, Brasil
E-mails: michael.santos@ifce.edu.br, gbarreto@ufc.br

***Resumo*—Redes regularizadas em espaços de Hilbert gerados por kernel (RKHS) constituem um arcabouço poderoso para estimação de funções não-lineares, com aplicações bem-sucedidas em áreas como identificação de sistemas dinâmicos, previsão de séries temporais e filtragem adaptativa. Tal técnica, porém, possui aplicação limitada a problemas que envolvem o processamento de sinais de larga escala, contaminados com ruído não-gaussiano e variantes no tempo. Isto posto, neste artigo introduzimos uma nova proposta de redes regularizadas no RKHS com as seguintes características. (i) O modelo preditor é atualizado para cada nova amostra de dados via aprendizado recursivo. (ii) O critério de otimalidade baseado no erro médio quadrático (MSE) é substituído pela *correntropia* a fim de conferir robustez a ruído não-gaussiano. (3) O critério de esparsificação por *Novidade* é usado para adicionar amostras a um dicionário de vetores-suporte. (4) Um critério de poda usando *divergência de Kullback-Leibler* é aplicado para excluir amostras do dicionário de vetores-suporte tornando-o capaz de rastrear um sistema variante no tempo. A proposta é avaliada em três conjuntos de dados, sendo um destes de larga escala, para diferentes níveis de contaminação por *outliers* na tarefa de identificação de sistemas dinâmicos. Os resultados obtidos pela rede regularizada proposta tem reduzido custo e complexidade computacional, atingindo um alto poder preditivo, com excelente robustez a *outliers* e reduzido uso de memória pela matriz de kernel.**

***Keywords*—Correntropia; esparsidade; identificação de sistemas; métodos de kernel; redes regularizadas.**

I. Introdução

A exploração da teoria de regularização de Tikhonov [1] e de espaços de Hilbert gerados por kernel (*reproducing kernel Hilbert space* - RKHS) [2] tem permitido que esquemas como redes regularizadas (*regularization networks* - RN) [3], regressão por processo gaussiano (*Gaussian process regression* - GPR) [4] e máquinas de vetores-suporte (*support vector machines* - SVM) [5] realizem a tarefa de aprendizagem por amostras de dados através da estimação de funções de dados ruidosos e esparsos [6]. Estas técnicas conhecidas como métodos de kernel possuem uma variedade de aplicações, como cancelamento de ruído, identificação de sistemas e equalização de canais, e tem ganhando destaque devido a capacidade de lidar com processamento de sinais não-lineares.

Entretanto, as técnicas baseadas em métodos de kernel perdem precisão quando lidam com problemas de larga escala, onde a quantidade de dados é elevada, ou quando é necessária uma estimação recursiva (*online*), como em problemas de controle adaptativo ou detecção de falta. Isso ocorre devido a expansão dos kernels cujo número de termos é igual ao número de dados de entrada. Para contornar esta questão diversos procedimentos de esparsificação vem sendo propostos [7], [8], [9], [10]. Contudo, estes procedimentos não são adequados em um cenário variante no tempo onde é necessário que a técnica seja capaz de esquecer informações passadas pouco relevantes e rastreie alterações na saída [11], [12].

Além da problemática relacionada a expansão dos kernels, a técnica de estimação precisa lidar com processamento de sinais não-gaussianos presentes em dados reais. Uma alternativa para esta questão vem sendo substituir as funções custo baseadas em estatísticas de segunda ordem, como o erro médio quadrático (*mean square error* - MSE), por descritores de teoria da informação (*information theoretic learning* - ITL) [13]. Um que se destaca por sua simplicidade e robustez, é conhecido como correntropia [14], [15].

A. Descrição do Problema

No presente trabalho, estamos interessados em estender uma rede regularizada *batch* para uma aprendizagem recursiva (*online*) através da construção e atualização sequencial de matrizes e suas inversas. Assim nosso desafio é adaptar a RN para lidar com processamento de dados de larga escala, variantes no tempo e não-gaussianos. Esta adaptação é baseada em quatro paradigmas:

1) Filtragem adaptativa, de forma que o modelo seja construído continuamente para cada nova amostra de dados tornando a rede apta para o aprendizado recursivo (*online*);
2) ITL, para permitir extrair mais informações dos dados do que as estatísticas de segunda ordem e prover robustez a *outliers* (ruído não-gaussiano);
3) Critério de adição de vetores-suporte, para tornar o modelo parcimonioso, onde aplicamos o critério da novidade [16] para selecionar os vetores-suporte do kernel;
4) Critério de remoção de vetores-suporte [11], para tornar a rede adaptativa e capaz de lidar com dados variantes no tempo.

Nossa proposta, aqui chamada rede regularizada baseada em correntropia recursiva e esparsa (*Correntropy-based Regularization Network Online and Sparse* - CRoNOS) melhora a RN pela possibilidade de lidar com dados de larga escala, variantes no tempo e contaminados com ruído não-gaussiano através de um procedimento de esparsificação híbrido (*growing and shrinking*) que seleciona, adiciona e remove vetores-suporte.

Para avaliar o desempenho da nossa proposta (CRoNOS) realizamos experimentos computacionais como prova de conceito utilizando três conjuntos de dados na tarefa de identificação de sistemas não-lineares (dois de pequena escala e um de larga escala). Diferentes níveis de contaminação por *outliers* são considerados para avaliar a robustez do modelo proposto e um regime de simulação livre (i.e. com realimentação da saída predita) é estabelecido para avaliar a capacidade de generalização.

B. Estrutura do trabalho

O restante deste artigo esta organizado da seguinte forma. Na Seção 2 revisamos os métodos empregados para lidar com os desafios explorados neste trabalho. Na Seção 3 apresentamos nossa proposta (CRoNOS) e discutimos como realizar a adição e remoção dos vetores-suporte com baixo custo computacional. Na Seção 4 realizamos a análise dos experimentos computacionais e discutimos os resultados alcançados. As conclusões e propostas de trabalhos futuros são apresentados na Seção 5.

II. Fundamentos Teóricos

A. Redes regularizadas no RKHS

As redes regularizadas (RN) são interpretadas como uma rede neural com uma camada oculta, onde supõe-se que o conjunto de dados de entrada e saída $g = \{(\boldsymbol{x}_i, y_i) \in \mathcal{R}^d \times \mathcal{R}\}$ é obtido por uma amostragem aleatória da função $f(\cdot)$, que pertence ao mesmo espaço da função $\mathcal{X}$ definida em $\mathcal{R}^d$, na presença de ruído. Para recuperar a função $f(\cdot)$, ou uma aproximação, uma escolha comum para solução deste problema é minimizar a seguinte função [3]:

$$H[f] = \sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda||f||^2_{\mathcal{H}}, \tag{1}$$

onde λ é um número positivo chamado parâmetro de regularização e $||\cdot||_{\mathcal{H}}$ denota a norma de uma função de suavidade no espaço de Hilbert $\mathcal{H}$. O primeiro termo da Equação 1 impõe proximidade entre os dados, e o segundo suavidade, enquanto o parâmetro de regularização controla o ajuste entre estes dois termos.

O acoplamento desta teoria de regularização com a de RKHS permite a definição da função de suavização no espaço de Hilbert $\mathcal{H}$ na forma do kernel $\boldsymbol{K}$, como uma função simétrica e definida positiva [6]. Neste caso a função de suavização pode ser definida, por exemplo, como $||f||^2_{\mathcal{H}} = \int d\boldsymbol{x} f^2(\boldsymbol{x})/\boldsymbol{K}(\boldsymbol{x})$. Assim a função que minimiza a Equação 1 tem a seguinte forma:

$$f(x) = \sum_{i=1}^{N}\alpha_i \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}_i) + \sum_{j=1}^{k} d_j \psi_j(\boldsymbol{x}), \tag{2}$$

onde $\{\psi_j\}_{j=1}^{k}$ é uma base de dimensão k no espaço nulo $\mathcal{N}$ da função de suavidade, que em muitos casos é um termo polinomial. Os coeficientes d_j e α_i dependem dos dados de tal forma que satisfaz o seguinte sistema linear:

$$(\boldsymbol{K} + \lambda \boldsymbol{I})\boldsymbol{\alpha} + \Psi^\top \boldsymbol{d} = \boldsymbol{y}, \tag{3}$$

onde I é a matriz identidade e o coeficiente $\boldsymbol{\alpha}$ pode ser determinado (Negligenciando o termo polinomial por simplificação) como:

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}. \tag{4}$$

Observação 1. A função de estimação da RN no RKHS definida nesta seção é baseada no MSE que não possui robustez a *outliers*. Além disso nota-se na Equação 4 um processo *batch* de estimação onde para estimar o coeficiente $\boldsymbol{\alpha}$ é necessário utilizar todos os dados de entrada para montar a matriz de kernel (Matriz de Gram) e uma operação de inversão de matrizes com custo computacional $\mathcal{O}(N)^3$.

B. Correntropia

A correntropia é uma função de correlação generalizada [14] onde do ponto de vista de ITL está diretamente relacionada à estimativa da entropia quadrática de Renyi's. Assim, a correntropia é definida em termos de produtos internos de vetores no RKHS definida como $V(X,Y) = E[k(X,Y)]$ onde $E[\cdot]$ denota o valor esperado e $k(\cdot,\cdot)$ é um kernel de Mercer. Como os produtos internos são uma medida de similaridade, essa função efetivamente mede a diferença entre pares dos vetores de características, separados por um certo atraso de tempo no espaço de entrada. Este descritor vem sendo efetivamente aplicado no processamento de sinais não-gaussianos [7], [17] através de um função custo para treinamento de sistemas adaptativos definida como critério da máxima correntropia (*maximum correntropy criterion* - MCC) [15]:

$$\hat{\boldsymbol{\theta}}_{mcc} = \arg\ \max_{\boldsymbol{\theta}} \sum_{i=1}^{N} k_{\sigma_1}(e_t|\boldsymbol{\theta}), \tag{5}$$

onde $e_i|\boldsymbol{\theta} = y_i - f(\boldsymbol{x}_i;\boldsymbol{\theta})$, $i = 1,\ldots,N$, é o erro de predição (ou residual) produzido por um modelo aproximado $f(\boldsymbol{x}_i;\boldsymbol{\theta})$ parametrizado pelo vetor $\boldsymbol{\theta}$.

O kernel Gaussiano $k_{\sigma_1}(e_i|\boldsymbol{\theta})$, adotado neste trabalho, é o mais utilizado nas aplicações e é definido como:

$$k_{\sigma_1}(x-y) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left\{-\frac{(x-y)^2}{2\sigma_1^2}\right\}, \tag{6}$$

onde o tamanho (*bandwith*) do kernel $\sigma_1 > 0$ controla a largura da Gaussiana.

A estimativa da Equação 6 é equivalente a um problema dos mínimos quadrados ponderados e tem uma forte relação com o Estimador M de Huber [18]:

$$\hat{\boldsymbol{\theta}}_{mcc} = \arg\ \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \rho(e_i) e_i^2, \tag{7}$$

onde o termo ponderador $\rho(e_i)$ penaliza os erros grandes, usualmente causados por *outliers* e para o kernel Gaussiano:

$$\rho(e_i) = \frac{1}{\sqrt{2\pi}\sigma_1^3} \exp\left\{-\frac{e_i^2}{2\sigma_1^2}\right\}, \quad i = 1, \ldots, m_t, \quad (8)$$

pode ser visto como uma função custo robusta capaz de manipular amostras discrepantes porque não amplifica seus efeitos.

Observação 2. Em comparação com os critérios tradicionais adotados para a modelagem de processos orientada aos dados, como o bem conhecido MSE, o MCC tem várias vantagens: (1) sempre é limitado a qualquer distribuição; (2) contém todos os momentos de ordem par, e é útil para processamento de sinal não-linear e não gaussiano; (3) é uma medida de similaridade local, portanto, é robusta para amostras discrepantes [17].

C. Critério de adição de vetores-suporte

Durante o processo de aprendizagem das técnicas baseadas em métodos de kernel as entradas $\boldsymbol{x}$ são colecionadas na forma de vetores-suporte para cômputo da matriz de kernel $\boldsymbol{K}$ o que faz com que esta cresça exponencialmente e tenha dimensão $N \times N$ onde N é a quantidade total de amostras de entrada disponíveis. Entretanto, em aplicações recursivas (*online*) a quantidade de dados de entrada cresce para o infinito o que faz que para em um sentido de filtragem adaptativa a matriz de kernels $\boldsymbol{K}$ não possa crescer a cada novo par $(\boldsymbol{x}_t, y_t)$. Muitos critérios na forma de heurísticas, em sua maioria, vem sendo utilizados para selecionar os vetores-suporte que irão formar um chamado dicionário $\mathcal{D}$ de vetores-suporte [16], [19], [10], [20]. Através da utilização destes critérios a matriz de kernels $\boldsymbol{K}$ cresce de tempos em tempos pela adição seletiva de vetores-suporte.

A ideia geral é assumir que a cada instante de tempo t, depois de $t-1$ observações de amostras de treinamento $\{\boldsymbol{x}_i\}_{i=1}^{t-1}$, iremos montar um dicionário de m_t $(m_t \ll t)$ vetores-suporte comprimidos em um subconjunto de amostras de entradas relevantes $\mathcal{D}_{t-1}^{sv} = \{\tilde{\boldsymbol{x}}_j\}_{j=1}^{m_t-1}$.

Neste trabalho adotamos como critério de seleção para adição de vetores-suporte o de novidade [16] por sua capacidade de formar novas representações, memória e simplicidade de implementação, características desejáveis em problemas que lidam com dados sequencias e variantes no tempo. O critério de novidade basicamente pode ser definido como uma medida de distância entre a amostra de entrada atual e amostras passadas ou dicionário, onde sem perda de generalidade, simplesmente usamos a distância Euclidiana [20]. Para cada amostra de entrada $\boldsymbol{x}_t$ e comparamos a norma quadrática com seus vizinhos mais próximos ($\tilde{\boldsymbol{x}}$ no $\mathcal{D}_i$) com um limiar de quantização vetorial $\delta \geq 0$. A nova amostra somente é nova se $\min ||\boldsymbol{x}_t - \tilde{\boldsymbol{x}}||^2 > \delta$. O aumento deste limiar reduz o tamanho do dicionário final. Na aplicação prática deste critério é adicionado também um critério do erro controlado pelo limiar ϵ e computado por $||\hat{y}_t - y_t|| > \epsilon$ onde $\hat{y}_t$ é a saída predita pelo modelo no treinamento. Para a amostra de entrada $\boldsymbol{x}_t$ ser adicionada no dicionário os dois critérios (Novidade e erro) devem ser atendidos.

D. Critério de remoção de vetores-suporte

Em um regime de aprendizado sequencial e variante no tempo as amostras de entrada $\boldsymbol{x}$ podem ser tornar pouco relevantes para o modelo no decorrer do tempo. Sendo assim remover os vetores-suporte menos relevantes depois de adicionar um novo no dicionário é importante para indução de uma esparsidade eficiente no modelo.

Um critério utilizado para selecionar os vetores-suporte que deverão ser removidos é selecionar o vetor de suporte i que minimiza a divergência de Kullback-Leibler (KL) entre a predição exata e a aproximação a posteriori considerando a informação perdida com a remoção de vetor de suporte [11]:

$$KL(p(f_{t+1}|\mathcal{D}_{t+1})||p([f_{t+1}]_i|[f_{t+1}]_{-i})p([f_{t+1}]_{-i}|\mathcal{D}_{t+1}) \quad (9)$$

Para diminuir o custo computacional ao invés de computar a divergência de KL analiticamente é considerado minimizar o erro quadrático que a aproximação a posteriori induz o que produz resultados igualmente efetivos na poda dos vetores-suporte menos relevantes.

Para computar o erro quadrático que será introduzido pela remoção de cada vetor de suporte e assim indicar qual vetor de suporte removida implica no menor erro utilizamos o seguinte critério por poda [21], [22]:

$$\arg\min_i \left(\frac{[\boldsymbol{P}_t \boldsymbol{y}_t]_i}{[\boldsymbol{P}]_{i,i}}\right)^2 > \gamma, \quad (10)$$

onde $\boldsymbol{P}_t = (\boldsymbol{K}_t + \lambda I)^{-1}$ e γ é o limiar que determina se o vetor de suporte que produz o menor erro será removido. Observe que aqui a matriz $\boldsymbol{P}_t$ deverá ser construída sequencialmente para o instante de tempo t como discutiremos na próxima seção.

III. Abordagem proposta

Nosso principal objetivo neste trabalho é apresentar uma proposta variante da RN recursiva, robusta a *outliers*, esparsa e adaptativa capaz de realizar a modelagem de sistemas dinâmicos a partir de dados disponibilizados sequencialmente no tempo. A abordagem proposta, nomeada CRoNOS, é então baseada na fusão dos métodos discutidos na Seção 2: (i) Rede regularizada [3], (ii) correntropia [14], (iii) critério de novidade [16] e (iv) critério do erro quadrático [11].

A primeira etapa é substituir a função custo baseada no MSE e definida na Equação 1 pelo MCC:

$$H[f] = \sum_{i=1}^{t} k_{\sigma_1}(f(\boldsymbol{x}_i) - y_i) + \lambda ||f||_{\mathcal{H}}^2, \quad (11)$$

onde ao invés de encontrar a função $f(\boldsymbol{x}_i)$ que minimiza a Equação 1 desejamos agora agora encontrar a função $f(\boldsymbol{x}_i)$ que maximiza a Equação 11.

Calculando o gradiente com relação à $||f||$ seguindo a mesma metodologia descrito na Seção 2A [3] a função que maximiza a Equação 11 terá o coeficiente $\tilde{\boldsymbol{\alpha}}$ dado por:

$$\tilde{\boldsymbol{\alpha}}_t = (\tilde{\boldsymbol{K}}_t \tilde{\boldsymbol{B}}_t + \lambda \sigma_1^2 \tilde{\boldsymbol{I}})^{-1} \tilde{\boldsymbol{B}}_t \tilde{\boldsymbol{y}}_t, \quad (12)$$

onde usando o lema de inversão de matriz esta pode ser reescrita como:

$$\tilde{\boldsymbol{\alpha}}_t = (\tilde{\boldsymbol{K}}_t + \lambda\sigma_1^2 \tilde{\boldsymbol{B}}_t^{-1})^{-1}\tilde{\boldsymbol{y}}_t, \tag{13}$$

Agora, usando nosso procedimento de esparsificação, a dimensão dos elementos na Equação 13 é m_t e elas calculadas com o subconjunto de amostra do dicionário, onde $\tilde{\boldsymbol{K}} \in \mathbb{R}^{m_t \times m_t}$ é uma matriz de kernels cujo entradas são dados por $\tilde{K}_{ij} = \langle \phi(\tilde{\boldsymbol{x}}_i), \phi(\tilde{\boldsymbol{x}}_j)\rangle, \forall\ i,j = 1,\ldots,m_t$. $\tilde{\boldsymbol{B}}_t$ é uma matriz diagonal cujo elementos diagonais são computados como $\tilde{\boldsymbol{B}}_t = \rho(e_i), \quad i = 1,\ldots,m_t$.

Para tornar esta proposta apta para a aprendizagem recursiva (note a presença do subscrito 't' na Equação 12) e evitar o custo computacional elevado $\mathcal{O}(m_t^3)$ da inversão de matrizes iremos computar recursivamente os elementos da Equação 12 onde a atualização das matrizes irá depender dos critérios de esparsificação aplicados.

A. Adicionando vetores-suporte

Caso o critério de seleção de vetores-suporte para adição no dicionário definido na Seção 2C seja atendido, a amostra do dados de entrada $\boldsymbol{x}_t$ deverá ser adicionada no dicionário $\mathcal{D}_t^{sv} = \mathcal{D}_{t-1}^{sv} \bigcup \{\boldsymbol{x}_t\}$ e $m_t = m_{t-1}+1$. Como uma consequência dessa inclusão, a matriz de kernel $\boldsymbol{K}$ deverá ser atualizada de acordo.

Fazendo $\tilde{\boldsymbol{P}}_t = \boldsymbol{H}_t^{-1} = (\tilde{\boldsymbol{K}}_t + \lambda\sigma_1^2\tilde{\boldsymbol{B}}_t^{-1})^{-1}$, temos

$$\boldsymbol{P}_t = \begin{bmatrix} \tilde{\boldsymbol{K}}_{t-1} + \lambda\sigma_1^2\tilde{\boldsymbol{B}}_{t-1}^{-1} & \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t) \\ \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)^\top & \tilde{k}_{tt} + \lambda\sigma_1^2\tilde{b}_t \end{bmatrix}^{-1}, \tag{14}$$

onde $\tilde{b}_t = 1/\rho(e_t)$, $e_t = \hat{y}_t - y_t$, $\hat{y}_t = \sum_{i=1}^{m_t}\tilde{\alpha}_i k(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_i) = \tilde{\boldsymbol{\alpha}}^\top\tilde{\boldsymbol{k}}_t$, $\tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t) = [k(\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_1) \ \cdots \ k(\boldsymbol{x}_t,\tilde{\boldsymbol{x}}_{m_t})]^\top$ e $\tilde{k}_{tt} = k(\tilde{\boldsymbol{x}}_t,\tilde{\boldsymbol{x}}_t)$. Onde é observável que

$$\tilde{\boldsymbol{P}}_t^{-1} = \begin{bmatrix} \tilde{\boldsymbol{P}}_{t-1}^{-1} & \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t) \\ \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)^\top & \tilde{k}_{tt} + \lambda\sigma_1^2\tilde{b}_t \end{bmatrix}. \tag{15}$$

Usando a identidade de inversão de matrizes em bloco obtemos a seguinte expressão:

$$\tilde{\boldsymbol{P}}_t = r_t^{-1}\begin{bmatrix} \tilde{\boldsymbol{P}}_{t-1}r_t + \boldsymbol{z}_t\boldsymbol{z}_t^\top & -\boldsymbol{z}_t \\ -\boldsymbol{z}_t^\top & 1 \end{bmatrix}, \tag{16}$$

onde $\boldsymbol{z}_t = \tilde{\boldsymbol{P}}_{t-1}\tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)$ e

$$r_t = \lambda\sigma_1^2\tilde{b}_t + \tilde{k}_{tt} - \boldsymbol{z}_i^\top\tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t). \tag{17}$$

B. Removendo vetores-suporte

Toda vez que um novo vetor de suporte é adicionado ao dicionário $\mathcal{D}$ aplicamos o critério definido na Seção 2D para determinar se e quais vetores-suporte se tornaram pouco relevantes e podem ser removidos do dicionário $\mathcal{D}_t^{sv} = \mathcal{D}_t^{sv} - \{\boldsymbol{x}_i\}$ e $m_t = m_{t-1} - 1$.

Assim primeiramente computamos o erro quadrático para cada vetor de suporte candidato a ser removido usando a Equação 10 e em seguida removemos o vetor de suporte que produz o menor erro atualizando a matriz $\tilde{\boldsymbol{P}}_t$ e eliminando o par de entrada e saída correspondentes ($\boldsymbol{x}_i$,y_i).

Para atualizar a matriz $\tilde{\boldsymbol{P}}_t$ com baixo custo computacional utilizamos uma metodologia baseada na relação entre a inversa de uma matriz e sua matriz [23].

Sendo assim, iremos ter uma submatriz $\tilde{\boldsymbol{H}}_{\bar{\boldsymbol{p}};\bar{\boldsymbol{q}}}$ obtida de $\tilde{\boldsymbol{H}}_t$ pela eliminação da linha p e da coluna q. Este problema é diretamente relacionado com o cálculo de uma matriz inversa perturbada $(\tilde{\boldsymbol{H}}_t + Z)^{-1}$, onde Z é a perturbação de uma matriz de $\tilde{\boldsymbol{H}}_t$. Essa matriz inversa pode ser calculada utilizando a fórmula de Sherman-Morrison-Woodbury:

$$\tilde{\boldsymbol{P}}_t = (\tilde{\boldsymbol{H}}_t - vu^\top)^{-1} = \tilde{\boldsymbol{H}}_t^{-1} + \frac{(\tilde{\boldsymbol{H}}_t^{-1}v)(u^\top\tilde{\boldsymbol{H}}_t^{-1})}{1 - u^\top\tilde{\boldsymbol{H}}_t^{-1}v}, \tag{18}$$

onde $v, u \in \mathbb{R}^{m_t}$ são vetores coluna da fórmula de Sherman-Morrison-Woodbury. Assim, $\tilde{\boldsymbol{H}}_t - vu^\top$ é definido por $v = \tilde{\boldsymbol{H}}_q - c_p$, onde $\tilde{\boldsymbol{H}}_q$ é o vetor da coluna q de $\tilde{\boldsymbol{H}}_t$, $c_p \in \mathbb{R}^{m_t}$ é o vetor canônico da coluna p, e $u = c_q$ é o vetor canônico da coluna q. Com estas definições, $\tilde{\boldsymbol{H}}_t - vu^\top$ é igual a $\tilde{\boldsymbol{H}}_t$ exceto na coluna q, que é igual a c_p. Pela aplicação da fórmula de Sherman-Morrison-Woodbury (18) para calcular $(\tilde{\boldsymbol{H}}_t - vu^\top)^{-1}$, $(\tilde{\boldsymbol{H}}_{\bar{\boldsymbol{p}};\bar{\boldsymbol{q}}})^{-1}$ é obtido pela eliminação da linha q e da coluna p de $(\tilde{\boldsymbol{H}}_{\bar{\boldsymbol{p}};\bar{\boldsymbol{q}}})^{-1}$ obtendo assim a atualização de $\tilde{\boldsymbol{P}}_t$.

Observe que a base canônica da matriz $\tilde{\boldsymbol{H}}$ é formada por vetores que possuem 1 em uma coordenada e 0 na outra, além disso, a linha q e a coluna p são iguais, então é suficiente determinar a partir de uma linha de $\tilde{\boldsymbol{H}}$ sua forma canônica.

Como nossa proposta utiliza no procedimento de esparsificação um critério para selecionar vetores-suporte que serão removidos podemos retirar no critério de seleção para adicionar vetores-suporte o critério do erro exposto na Seção 2C, permanecendo apenas o critério de novidade puro.

Na tabela I fornecemos o pseudocódigo do algoritmo proposto neste trabalho.

IV. Experimentos computacionais

Nesta seção, realizamos a análise dos experimentos computacionais na tarefa de identificação de sistemas não-lineares.

Para esta tarefa assumimos uma estrutura não linear ARX (NARX). que pode ser representa por uma classe de sistemas não-lineares discretos [24]:

$$\begin{aligned} y_t &= f(y_{t-1},\ldots,y_{t-L_y},u_{t-1},\ldots,u_{t-L_u}) + e_t, \\ &= f(\boldsymbol{x}_t) + e_t, \quad t = 1,\ldots,N, \end{aligned} \tag{19}$$

onde L_u e L_y denota a ordem dos vetores de entrada e saída, respectivamente. $f(\cdot)$ é a função alvo e y_t, u_t e e_t são, respectivamente, a saída do sistema, a entrada do sistema e o ruído aleatório [1] amostrado no instante de tempo t. Neste caso, o vetor de entrada $\boldsymbol{x}_t$ é obtido pela concatenação L_y saída passadas observáveis $y_t \in \mathbb{R}$ e L_u entradas passadas controladas $u_t \in \mathbb{R}$ dentro de um único vetor de regressão:

$$\boldsymbol{x}_t = [y_{t-1},\ldots,y_{t-L_y},u_{t-1},\ldots,u_{t-L_u}]^\top. \tag{20}$$

[1] Usualmente assumido como ruído aditivo branco gaussiano (AWGN).

Tabela I
PSEUDOCÓDIGO DO MODELO CRONOS.

Algoritmo	Complexidade
Hiperparâmetros: $\lambda, \sigma_1, \sigma_2, \delta, \gamma$;	-
Inicialização: $\tilde{\boldsymbol{P}}_1 = 1/\tilde{k}_{11}; \quad b_1 = 1; \quad m_1 = 1;$	-
Computa $\tilde{\boldsymbol{\alpha}}_1$ da Eq. (13);	$\mathcal{O}(1)$
Computa $\tilde{b}_1 = 1/\rho(e_1)$ da Eq. (8);	$\mathcal{O}(1)$
for $t = 2 : N$,	-
Pega nova amostra: $(\boldsymbol{x}_t, y_t)$	-
Teste para adicionar vetor-suporte:	-
if $\min \|\|\boldsymbol{x}_t - \tilde{\boldsymbol{x}}\|\|^2 > \delta$	$\mathcal{O}(m_t)$
$\mathcal{D}_t^{sv} = \mathcal{D}_{t-1}^{sv} \bigcup \{\boldsymbol{x}_t\}$;	$\mathcal{O}(1)$
$\boldsymbol{z}_t = \tilde{\boldsymbol{P}}_{t-1}\tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)$;	$\mathcal{O}(m_t^2)$
$\tilde{b}_t = 1/\rho(e_t)$ da Eq. (8);	$\mathcal{O}(1)$
$r_t = \lambda\sigma_1^2\tilde{b}_t + \tilde{k}_{tt} - \boldsymbol{z}_i^\top \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)$;	$\mathcal{O}(m_t)$
Computa $\tilde{\boldsymbol{P}}_t$ da Eq. (16);	$\mathcal{O}(m_t^2)$
Computa $\tilde{\boldsymbol{\alpha}}_t$ da Eq. (13);	$\mathcal{O}(m_t)$
$m_t = m_{t-1} + 1$;	$\mathcal{O}(1)$
Teste para remover vetor-suporte:	-
if $\min([\tilde{\boldsymbol{P}}_t\tilde{\boldsymbol{y}}_t]_i/[\tilde{\boldsymbol{P}}]_{i,i})^2 > \gamma$	$\mathcal{O}(m_t^2)$
$\mathcal{D}_t^{sv} = \mathcal{D}_t^{sv} - \{\boldsymbol{x}_i\}$;	$\mathcal{O}(1)$
Atualiza $\tilde{\boldsymbol{P}}_t$ da Eq. (18);	$\mathcal{O}(m_t^2)$
Atualiza $\tilde{\boldsymbol{\alpha}}_t$ da Eq. (13);	$\mathcal{O}(m_t)$
$m_t = m_{t-1} - 1$;	$\mathcal{O}(1)$
end if	-
else	-
$\mathcal{D}_t^{sv} = \mathcal{D}_{t-1}^{sv}$;	-
end if	-
end for	-
Saída: $\tilde{\boldsymbol{\alpha}}_t, \mathcal{D}_t^{sv}$.	-

Assim, assumimos que as amostras de treinamento são sequencialmente disponibilizadas:

$$\mathcal{D}_t = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_t, y_t)\}, \tag{21}$$

onde ($\boldsymbol{x}_t$,y_t) $\in \mathbb{R}^{L_u+L_y} \times \mathbb{R}$ denota o par de entrada e saída do sistema.

Os experimentos envolvem dois conjuntos de dados artificiais de pequena escala e um de grande escala, onde são considerados dois cenários, sem e com contaminação por *outliers*. A contaminação por *outliers* permite avaliar o desempenho dos modelos na presença de ruído não-gaussiano. As figuras de mérito exploradas neste trabalho são a raiz do erro médio quadrático (*root mean square error* - RMSE) e o grau de esparsidade (tamanho do dicionário de vetores-suporte). Todas as avaliações do modelo durante a fase de teste correspondem ao regime de simulação livre (*com realimentação da saída predita*), em que os valores da saída predita são retornados para o vetor de regressão. Este regime de avaliação é muito mais exigente do que a usual predição um passo a frente (*one-step ahead*). Nestes experimentos todos os dados são reescalados no intervalo $[-1, 1]$.

Analisamos o desempenho de nossa proposta CRoNOS em comparação com uma rede regularizada clássica [3] e com um método baseado em filtragem adaptativa kernelizada que também usa a correntropia, conhecido como *kernel recursive maximum correntropy* (KRMC) [7]. No método KRMC usamos o critério de esparsidade de novidade (KRMC-NC) para verificarmos a diferença de performance com nossa proposta híbrida de esparsificação.

A. Conjuntos de dados de pequena escala

Neste primeiro experimento são utilizados dois conjuntos de dados artificiais de pequena escala [25]. O conjunto de dados *Artificial 1* é gerado de acordo com o seguinte sistema dinâmico de tempo discreto:

$$\begin{cases} y_{k+1} = \frac{y_k y_{k-1} y_{k-2}(y_{k-2}-1)u_{k-1}+u_k}{1+y_{k-2}^2+y_{k-1}^2} \\ u_k = \begin{cases} \sin(\pi k/125), k \leq 203 \\ 0.8\sin(\pi k/125) + 0.2\sin(2\pi k/25), k \geq 203 \end{cases} \end{cases} \tag{22}$$

O conjunto de dados *Artificial 2* é gerado conforme descrito abaixo:

$$\begin{cases} y_k = \frac{y_{k-1}}{1+y_{k-1}^2} + u_k^3 \\ u_k = \begin{cases} U(-2,2), \text{dados de treino} \\ \sin(2\pi k/25) + \sin(2\pi k/10), \text{dados de teste} \end{cases} \end{cases} \tag{23}$$

onde $U(-2, 2)$ denota uma distribuição uniformemente distribuída de números aleatórios em uma faixa específica.

Neste experimento avaliamos os resultados para o cenário livre de contaminação de *outliers* e com contaminação de *outliers*. No cenário de contaminação as amostras de saída y do treinamento são contaminadas aleatoriamente com as porcentagens de 10%, 20% e 30%. Os *outliers* são gerados como um ruído impulsivo com *pdf* de uma mistura gaussiana dada por $p_z(z) = (1-\epsilon) \times \mathcal{N}(0, 0.065) + \epsilon \times \mathcal{N}(0.1, 2\sigma_1(y))$, onde ϵ é determinado de acordo com a porcentagem de *outliers* no cenário e $\sigma_1(y)$ é o desvio padrão dos dados de treinamento originais. Além disso, são realizadas 10 rodadas independentes, onde cada rodada possui uma diferente distribuição de contaminação e os hiperparâmetros são ajustados por validação cruzada. A Tabela II contém o número de amostras de treinamento e teste, a ordem NARX do modelo e o tipo e largura de kernel ajustados conforme o estado da arte [25].

Relatamos nas Figuras 1(a) e 1(c) o RMSE, onde mostramos os valores médios e o desvio padrão para cada conjunto de dados nos cenários propostos. Nestas figuras verificamos que a RN baseada na função custo do MSE não é capaz de manter uma boa capacidade de predição do modelo a medida que a contaminação por *outliers* aumenta. Por outro lado, comparando a performance da nossa proposta CRoNOS com o KRMC-NC verificamos resultados numéricos similares, mas no cenário de máxima contaminação (30%) nossa proposta começa a apresentar uma significativa vantagem.

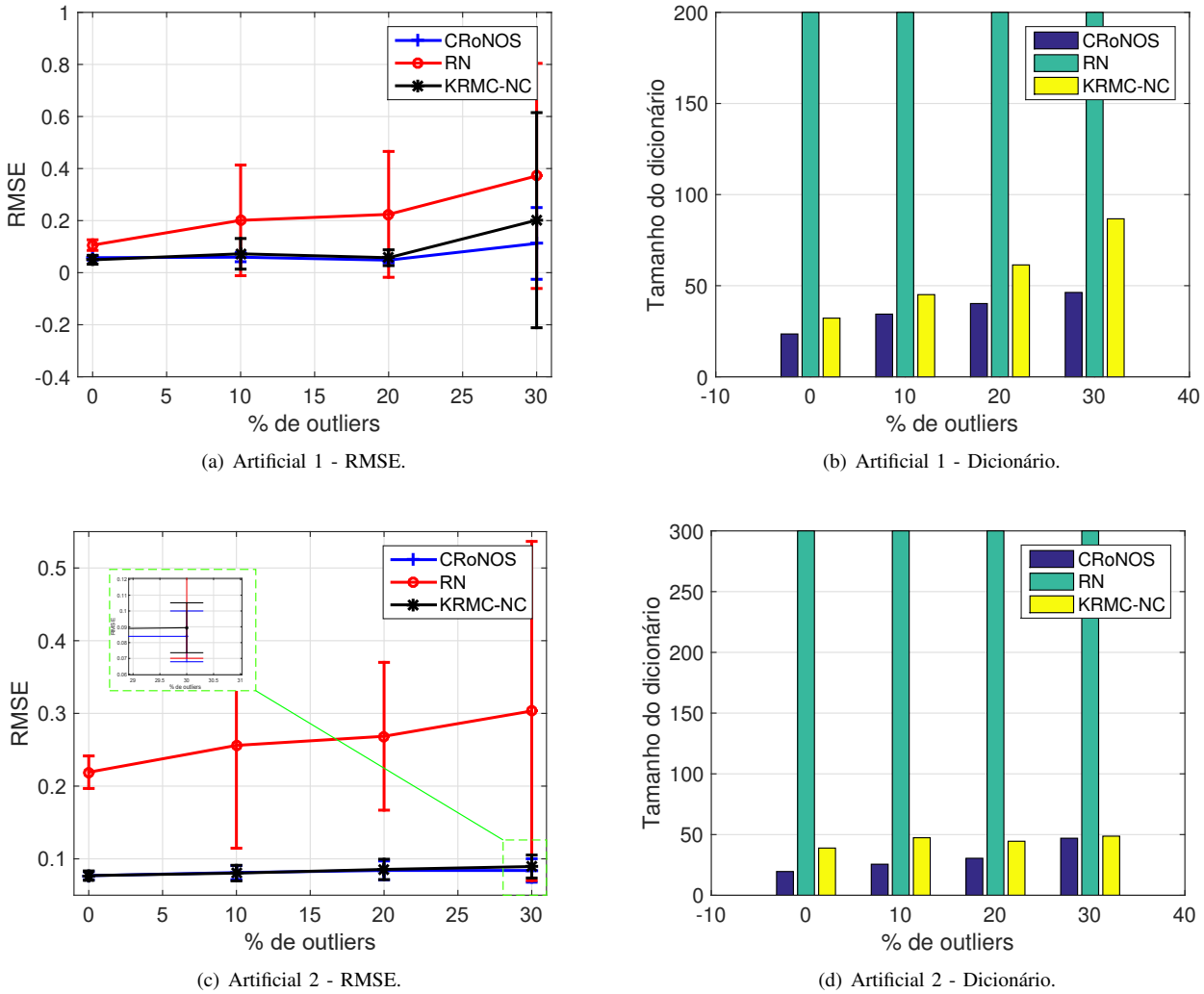
Os melhores resultados do CRoNOS frente ao KRMC-NC estão na melhor capacidade de selecionar vetores-suporte relevantes e em menor quantidade, reduzindo a norma dos coeficientes α estimados durante o treinamento, conforme pode ser verificado nas Figuras 1(b) e 1(d) onde é apresentado a quantidade média de vetores-suporte selecionada por cada modelo.

B. Conjunto de dados de larga escala

Neste experimento utilizamos um conjunto de dados de larga escala (*Large-scale*) conhecido como Silverbox [26].

Tabela II
RESUMO DOS CONJUNTOS DE DADOS.

Conjunto de dados	Número de amostras		Ordem NARX		Kernel	
	Treinamento	Teste	L_u	L_y	Tipo	Largura (σ_2)
Artificial 1	200	200	3	2	Gaussiano	0.1
Artificial 2	300	100	1	1	Gaussiano	0.1
Silverbox	45.000	40.000	10	10	Polinominal	1



(a) Artificial 1 - RMSE.

(b) Artificial 1 - Dicionário.

(c) Artificial 2 - RMSE.

(d) Artificial 2 - Dicionário.

Figura 1. (a) e (c) são os gráficos dos valores de RMSE relacionados ao teste em simulação livre com os dados contendo diferentes níveis de contaminação por outliers. (b) e (d) são os gráficos do tamanho médio do dicionário de vetores-suporte.

Este conjunto de dados representa um circuito elétrico simulando um sistema massa-mola amortecedor e contém um total de 131.072 amostras. Nós utilizamos as primeiras 40.000 amostras para avaliação do modelo e 45.000 das restantes para treinamento do modelo. Dois cenários são avaliados, um sem contaminação por *outliers* e outro com uma contaminação por *outliers* de 5% utilizando um ruído não-Gaussiano pela amostragem de uma distribuição t de Student com média zero e 2 graus de liberdade. Para este experimento são realizadas 5 rodadas independentes, onde cada rodada possui uma diferente distribuição de contaminação e os hiperparâmetros são ajustados também por validação cruzada. A Tabela II contém o número de amostras de treinamento e teste, a ordem NARX do modelo e o tipo e largura de kernel ajustados conforme o estado da arte [26].

Os resultados deste experimento são apresentados nas Figuras 2(a) e 2(b), onde apresentamos, respectivamente, através de um gráfico de caixa o RMSE, e do gráfico em barras o tamanho médio do dicionário de vetores-suporte para predição em regime de simulação livre das 5 rodadas independentes.

Neste experimento comparamos o desempenho da nossa proposta CRoNOS apenas com o modelo KRMC-NC pois para problemas com grandes quantidades de dados (*Large-scale*) o uso da RN se torna inviável, já que o custo computacional (Operação de inversão de matrizes) cresce exageradamente devido o uso de todas os pares de amostras de entrada e saída. Comparando o desempenho do CRoNOS e do KRMC-NC verificamos uma maior capacidade de predição da nossa proposta com um menor valor de RMSE e com uma menor quantidade de vetores-suporte nos dois cenários avaliados. Isto implica diretamente na norma dos coeficientes α estimados na construção do modelo, onde obtemos uma norma média de 1.6 no CRoNOS e de 4.8 no KRMC-NC. A menor norma do CRoNOS faz com que o modelo propague menos erros no regime de simulação livre, ou seja, tenha maior capacidade de generalização, conforme pode ser verificado nas Figuras 3(a) e 3(b), onde é possível confirmar o melhor ajuste da curva de predição produzida pelo CRoNOS.

V. Conclusões e trabalhos futuros

Neste artigo apresentamos uma nova proposta para uma técnica de redes neurais, conhecida como redes regularizadas. Nossa proposta, denominada CRoNOS, possui como características ser recursiva, esparsa, robusta a *outliers* e adaptativa.

O modelo de aprendizado CRoNOS integra a arquitetura de redes regularizadas no RKHS, de uma perspectiva de métodos de filtragem adaptativa kernelizados, com o critério da máxima correntropia (MCC). Esta proposta melhora as clássicas RN em importantes dimensões, uma vez que pode ser utilizada eficientemente para aprendizado recursivo (*online*) e que sua complexidade computacional é consideravelmente reduzida pela introdução de um procedimento de esparsificação que além selecionar os vetores-suporte que irão fazer parte do dicionário pode também descartar funções de kernel pouco relevantes para o modelo.

Nós avaliamos a proposta atráves da tarefa de identificação de sistemas não-lineares usando dois conjuntos de dados artificiais de pequena escala e uma conjunto de dados de larga escala (Silverbox), em regime de simulação livre (*Infinite Step-ahead*) e incluindo uma contaminação por *outliers*. Nossos experimentos como prova de conceito mostraram claramente os seguintes benefícios:

1) Redução do custo e complexidade computacional através da estratégia recursiva de construção e atualização de matrizes;
2) Aumento da robustez a *outliers* do modelo pela substituição do função custo MSE pelo MCC;
3) Redução do uso de memória através da utilização de um procedimento de esparsificação híbrido (*Growing and Shrinking*);

Além disso, os resultados demonstraram que a proposta é capaz de realizar a aprendizagem recursiva e manter um alto poder de predição em regime de simulação livre com um reduzido tamanho do dicionário.

Para trabalhos futuros esperamos realizar experimentos com o CRoNOS em um cenário não-estacionário e avaliar outros critérios de esparsificação para selecionar e descartar vetores-suporte.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores também agradecem ao IFCE (Campus de Canindé) e ao CNPq (Concessão nº 309451/2015-9) por apoiarem esta pesquisa.

Referências

[1] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed problems*. W.H. Winston, 1977.

[2] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950. [Online]. Available: http://www.jstor.org/stable/1990404

[3] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[4] C. Lincoln C. Mattos, A. Damianou, G. Barreto, and N. D. Lawrence, "Latent autoregressive gaussian processes models for robust system identification," *IFAC-PapersOnLine*, vol. 49, pp. 1121–1126, 12 2016.

[5] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[6] A. Chiuso and G. Pillonetto, "System identification: A machine learning perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 281–304, 2019.

[7] Z. Wu, J. Shi, X. Zhang, W. Ma, and B. Chen, "Kernel recursive maximum correntropy," *Signal Processing*, vol. 117, pp. 11 – 16, 2015.

[8] Y. Engel, S. Mannor, and R. Meir, "Sparse online greedy support vector regression," *Machine Learning: ECML 2002*, 2002.

[9] C. Saide, R. Lengelle, P. Honeine, and R. Achkar, "Online kernel adaptive algorithms with dictionary adaptation for mimo models," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 535–538, 2013.

[10] J. Zhao, H. Zhang, and G. Wang, "Projected kernel recursive maximum correntropy," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 7, pp. 963–967, July 2018.

[11] S. Van Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, Aug 2012.

[12] W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, "Kernel lms algorithm with forward-backward splitting for dictionary learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5735–5739.

[13] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[14] I. Santamaria, P. P. Pokharel, and J. C. Principe, "Generalized correlation function: definition, properties, and application to blind equalization," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2187–2197, 2006.

[15] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

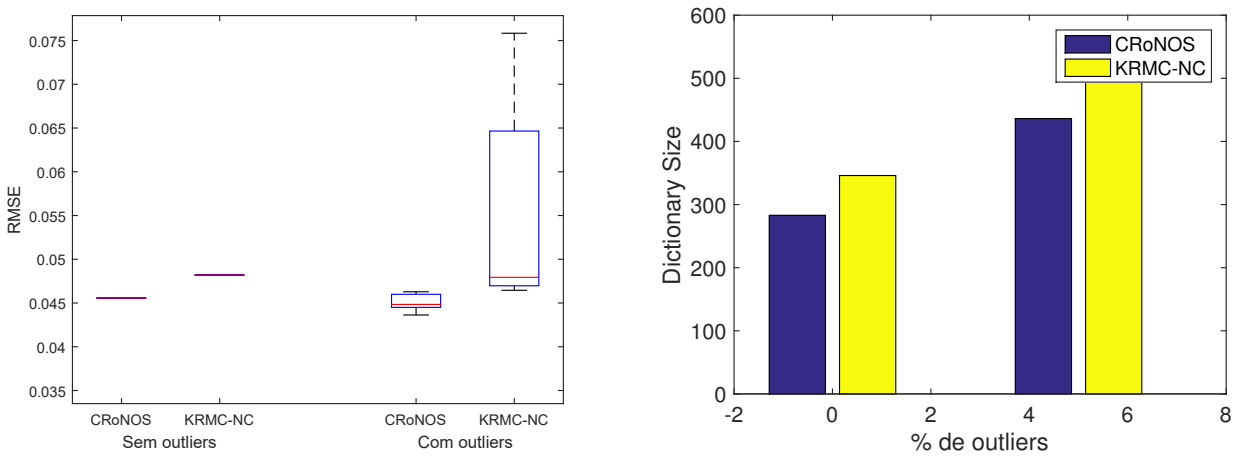
[16] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, June 1991.

[17] Y. Liu and J. Chen, "Correntropy-based kernel learning for nonlinear system identification with unknown noise: an industrial case study," *IFAC Proceedings Volumes*, vol. 46, no. 32, pp. 361 – 366, 2013, 10th IFAC International Symposium on Dynamics and Control of Process Systems.

[18] P. Huber, "Robust statistics," *Advances in Neural Information Processing Systems 9*, 1981, new York: Wiley.

[19] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

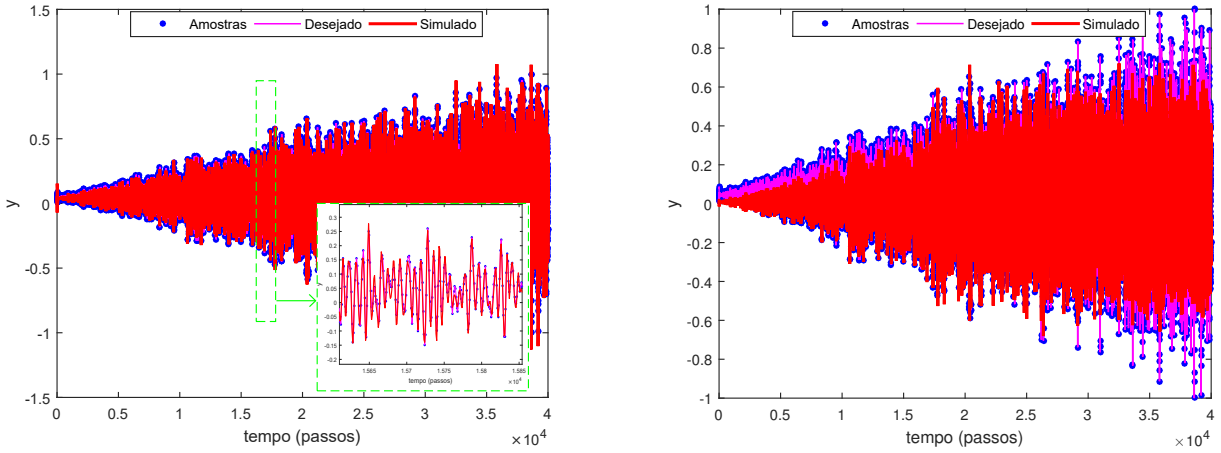
[20] K. Li and J. C. Príncipe, "Surprise-novelty information processing for gaussian online active learning (snip-goal)," *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2018.



(a) Silverbox - RMSE.

(b) Silverbox - Dicionário.

Figura 2. (a) é o gráfico dos valores de RMSE relacionados ao teste em simulação livre com os dados contendo diferentes níveis de contaminação por outliers. (b) é o gráfico do tamanho médio do dicionário de vetores-suporte.



(a) CRoNOS.

(b) KRMC-NC.

Figura 3. Saída predita para o teste em simulação livre com outliers usando o CRoNOS (a) e o KRNC-NC (b).

[21] L. Csató and M. Opper, "Sparse representation for gaussian process models," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00. Cambridge, MA, USA: MIT Press, 2000.

[22] B. de Kruif and T. de Vries, "Pruning error minimization in least squares support vector machines," *IEEE transactions on neural networks and learning systems*, vol. 14, no. 3, pp. 696–702, 2003.

[23] E. Juarez-Ruiz, R. Cortes-Maldonado, and F. Perez-Rodriguez, "Relationship between the inverses of a matrix and a submatrix," *Computacion y Sistemas*, vol. 20, pp. 251 – 262, 06 2016.

[24] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, N.J. : Prentice Hall, 1999, prentice Hall, Englewood Cliffs.

[25] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[26] T. Wigren and J. Schoukens, "Three free data sets for development and benchmarking in nonlinear system identification," *European Control Conference (ECC)*, pp. 2933–2938, July 2013.