

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA ELETRICIDADE

EWALDO ÉDER CARVALHO SANTANA JÚNIOR

**EXTRAÇÃO CEGA DE SINAIS COM ESTRUTURAS TEMPORAIS UTILIZANDO
ESPAÇOS DE HILBERT REPRODUZIDOS POR KERNEIS**

São Luís

2012

EWALDO ÉDER CARVALHO SANTANA JÚNIOR

**EXTRAÇÃO CEGA DE SINAIS COM ESTRUTURAS TEMPORAIS UTILIZANDO
ESPAÇOS DE HILBERT REPRODUZIDOS POR KERNEIS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Maranhão, como requisito parcial para a obtenção do grau de Mestre em Engenharia da Eletricidade.

Orientador: Allan Kardec Duailibe Barros Filho – Ph.D.

São Luís

2012

Santana Júnior, Ewalo Éder Carvalho.

Extração cega de sinais com estruturas temporais utilizando espaços de Hilbert reproduzidos por kernels / Ewalo Éder Carvalho Santana Júnior. – São Luís, 2012.

79 f.

Impresso por computador (fotocópia).

Orientador: Allan Kardec Duailibe Barros Filho.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia da Eletricidade, 2012.

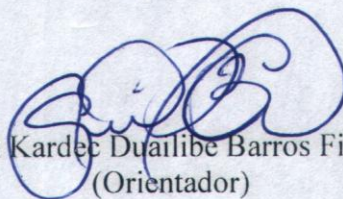
1. Extração cega de sinais. 2. Espaço de Hilbert reproduzido por Kernel. 3. Aprendizagem de máquina utilizando teoria da informação. I. Título.

CDU 621.391

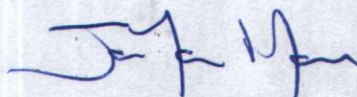
**EXTRAÇÃO CEGA DE SINAIS COM ESTRUTURA
TEMPORAL UTILIZANDO ESPAÇOS DE HILBERT
REPRODUZIDOS POR KERNEL**

Ewaldo Éder Carvalho Santana Júnior

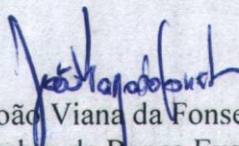
Dissertação aprovada em 10 de fevereiro de 2012.



Prof. Allan Kardec Duaribe Barros Filho, Ph. D.
(Orientador)



Prof. João Marcos Travassos Romano, Dr.
(Membro da Banca Examinadora)



Prof. João Viana da Fonseca Neto, Dr.
(Membro da Banca Examinadora)

*Dedico este trabalho a todas as
consciências em processo de
aperfeiçoamento e evolução.
FORÇA!*

AGRADECIMENTOS

Primeiramente gostaria de referenciar aos Agradecimentos que realizei em minha monografia [1]. Permanecem válidos todos os reconhecimentos por lá realizados. Desta forma também evito me repetir tanto ☺ .

Então lá vai a única parte que você, caro leitor, vai ler por completo desta dissertação.

Muito obrigado ao *Fator que guia todo este Universo* em direções, ao menos por mim sempre vivenciadas, tranqüilas, seguras, felizes, sábias e que nos fazem caminhar mesmo nos *vales das sombras da morte* sem temer mal algum. Deus, essa é pra Você!

Permito-me relembrar aos meus pais Ewaldo e Gorete que geraram este soma que agora uso para vos escrever. Pai, mãe, se vocês não tivessem me dado vida, talvez outro casal o fizesse, mas como foram vocês que aceitaram esta árdua missão de gerar e sustentar uma prole com tanta motivação para os estudos, mas tão pouca energia para as coisas práticas da vida, é para vocês dois o meu mais honesto MUITO OBRIGADO!

Não vou agradecer aos meus irmãos Arthur, Tiago e Ciro porque eles se comportaram muito mal em 2011. É brincadeira! Sem esses três para conversar minha vida certamente teria outro sentido e eu não poderia garantir que eu seria tão feliz quanto sou hoje. Talvez eu fosse, mas ainda bem que eu tenho vocês para compartilhar minha juventude, pensamentos, diversões, dúvidas, frustrações e vida digital.

Depois dos meus pais biológicos, agradeço da mesma forma aos meus mantenedores intelectuais. Primeiramente ao meu orientador o professor Allan Kardec Barros, por confiar na minha capacidade, me aceitar como seu aluno e servir de exemplo intelectual e motivacional. Agradeço também aos professores João Viana Fonseca, Maria da Guia da Silva, Sebastian Yuri, Ginalber Serra e demais professores do programa de pós-graduação em Engenharia Elétrica por manterem o programa em que agora me formo Mestre.

A pessoa em quem eu mais queria chegar nesta seção de Agradecimentos é aquela que simplesmente *aconteceu na minha vida*. Eu agradeço à minha namorada Inez pelo carinho e amor. É muito bom poder cuidar, contar, se divertir e viver contigo, Inez. Não agradeço à paciência durante este tempo de mestrado porque... enfim...

Inez também me deu a oportunidade de conhecer pessoas das quais gostei muito. Seus filhos Lucas, que é parceirão e com que eu adoro conversar e dividir experiências, Lívia, que é a loira mais linda do mundo e Zion... bom, Zion é chorão, mas segundo minha mãe eu também era, então garoto, segue meu exemplo e você vai ter o nobre prazer de ser pesquisador científico e dedicar sua mente a uma causa muito valiosa! Conheci também

Eliene e Rita, minhas cunhadas, que são muito divertidas, falam bem alto e com quem é ótimo almoçar e conversar. Ao marido de Rita, Richardson, sou grato por me ensinar a lidar com uma namorada Boguea. Minha outra cunhada, Julia, seu marido, Carlito e seus filhos San, Jéssica e Juliana também me receberam muito bem na família. Fui também otimamente tratado pela última cunhada que conheci, Karla, sua mãe “Biola” e seu marido Andervilson, com quem adorei tomar cerveja com energético e conversar. Todo este bem estar afetivo sem dúvida é importante para poder acordar cedo e pesquisar, por isto também sou grato.

Meus amigos de engenharia sabem que eu nunca os esqueço. Obrigado por também sempre lembrarem de mim: Emanuel, Polyelton, André Lobato, Leal, João Vítor, Cláudio, Luizão, Igor, Yuri, Nara, Hugo e todos os outros eng.s. Espero que a gente continue sempre se encontrando por tanto tempo quanto as condições físicas (trabalho) e as restrições de projetos (namoradas) permitirem.

Meus amigos do PIB são fundamentais para minha pesquisa e vida laboratorial. Daí meus agradecimentos a Marcus Vinícius, Enio, Denner, André, Daniel, Cristiane, Luís Cláudio, Fausto, Áurea e Simone. Eu torço pelo sucesso de toda esta equipe.

Agradeço aos meus amigos do Centro Holístico Aumbandhan Cabocla Jacira pelo apoio espiritual. Especialmente a Rafael, Kátia, Fátima, Eliane, Maria Légua, Oscar Légua, Maria Padilha, 7 Saias, Mariana, Boiadeiro e Tupinambá. Agradeço mais ainda pelo apoio consciencial de todos os amparadores anônimos que me acompanham.

Este não seria um texto vindo das minhas mãos se eu não agradecesse ao um homem que admiro e tenho como um pai. Agradeço a Leonardo da Vinci, cuja imagem sempre me vem à mente me estimulando à evolução intelectual e artística

Mais uma vez repito a prática mais socialmente correta que já pude pensar. Peço desculpas aos que não foram citados, mas, lembro que são todos parte da Natureza e, portanto, também são eternos credores de minha gratidão. Se mesmo assim alguém se sentir penalizado, por favor, adicione seu nome na lista abaixo e sinta-se agradecido:

“Escolha alguma área de interesse. Estude-a até o ponto de se tornar iniciado no assunto. Pare os estudos formais e passe a aplicar suas novas habilidades na prática. Guie a continuação de seus estudos pelas necessidades geradas pela SUA experiência, jamais pelo que o interesse dos outros definiu como prioritário.”

Eder Santana

“Leonardo da Vinci, Einstein, Aristóteles, Sócrates e todos os outros grandes pensadores que amamos e adoramos citar, eram humanos como eu e você.”

Eder Santana

RESUMO

Esta dissertação deriva e avalia um novo método não-linear para Extração Cega de Sinais através de operações algébricas em um Espaço de Hilbert Reproduzido por Kernel (RKHS, do inglês *Reproducing Kernel Hilbert Space*). O processo de extração de sinais desejados de misturas é realizado utilizando-se informação sobre a estrutura temporal deste sinal desejado. No presente trabalho, esta informação temporal será utilizada para realizar uma transformação linear na matriz de Gram das misturas transformadas para o espaço de Hilbert. Aqui, mostrar-se-á também que o método proposto é mais robusto, com relação a ambigüidades sobre a informação temporal do sinal desejado, que aqueles previamente apresentados na literatura para realizar a mesma operação de extração. A abordagem estudada a seguir pode ser vista como uma generalização da Análise de Componentes Principais utilizando Kerneis para analisar matriz de autocorrelação dos dados para um atraso específico. Sendo também uma *kernelização* da Análise de Componentes Dependentes, o método aqui desenvolvido é denominado *Análise de Componentes Dependentes utilizando Kerneis* (KDCA, do inglês *Kernel Dependent Component Analysis*). Também será abordada nesta dissertação, a perspectiva da Aprendizagem de Máquina utilizando Teoria da Informação do novo método apresentado, mostrando assim, que transformações são realizadas na função densidade de probabilidade do sinal extraído enquanto que operação lineares são calculadas no RKHS.

Palavras-chave: Extração Cega de Fontes, Espaço de Hilbert Reproduzido por Kernel, Aprendizagem de Máquina utilizando Teoria da Informação.

ABSTRACT

This work derives and evaluates a nonlinear method for Blind Source Extraction (BSE) in a Reproducing Kernel Hilbert Space (RKHS) framework. For extracting the desired signal from a mixture *a priori* information about the autocorrelation function of that signal translated in a linear transformation of the Gram matrix of the nonlinearly transformed data to the Hilbert space. Our method proved to be more robust than methods presented in the literature of BSE with respect to ambiguities in the available *a priori* information of the signal to be extracted. The approach here introduced can also be seen as a generalization of Kernel Principal Component Analysis to analyze autocorrelation matrices at specific time lags. Henceforth, the method here presented is a *kernelization* of Dependent Component Analysis, it will be called *Kernel Dependent Component Analysis* (KDCA). Also in this dissertation it will be show a Information-Theoretic Learning perspective of the analysis, this will study the transformations in the extracted signals probability density functions while linear operations calculated in the RKHS.

Keywords: Blind Signal Extraction, Reproducing Kernel Hilbert Spaces, Information-Theoretic Learning.

LISTA DE FIGURAS

Figura 4.1. RMSE da extração do sinal de ECG utilizando KDCA e DCA. a) RMSE do KDCA para vários atrasos τ , de 140 a 180, e tamanhos do kernel gaussiano σ , de 5 a 15. b) RMSE do KDCA para $\sigma = 15$. c) RMSE do DCA.....	29
Figura 5.1. RMSE entre onda triangular extraída e original em função do atraso τ	34
Figura 5.2. Valores esperados da diagonal da matriz $\mathbf{K}\tau$ em função do atraso τ para os sinais de ECG de uma mulher grávida. $\mathbf{K}\tau$ foi calculada com um kernel gaussiano de tamanho $\sigma = 5$	37
Figura 5.3. Sinais de ECG fetais extraídos das gravações de uma mulher grávida [54] utilizando atraso $\tau = \mathbf{110, 112, 114s}$ e os algoritmos KDCA, e os propostos em [11] (Barros) e [14] (Zhang).....	38

LISTA DE TABELAS

Tabela 1.1 – Como <i>kernelizar</i> um algoritmo.....	3
Tabela 2.1 – Etapas básicas de um método para extração cega de fontes.....	8
Tabela 2.2– Propriedades para funções contraste para extração cega de fontes	10
Tabela 4.1 – Algoritmo Análise de Componentes Dependentes com Kernel (KDCA)	27
Tabela 5.1. Resultados do Experimento 1.....	36
Tabela 5.2 Performance do algoritmo KDCA utilizando DIC comparada com calculo direto da matriz de Gram	39
Tabela 5.3 RMSE para extração de sinais senoidais de mistura não-linear utilizando KDCA e KTDSEP	40

LISTA DE SIGLAS

BSE	Extração Cega de Sinais, do inglês <i>Blind Signal Extraction</i>
ICA	Análise de Componentes Independentes, do inglês <i>Independent Component Analysis</i>
RNA	Redes Neurais Artificiais
EEG	Eletroencefalografia
MEG	Magnetoencefalografia
DCA	Análise de Componentes Dependentes, do inglês <i>Dependent Component Analysis</i>
MGG	Magnetogastograma
PCA	Análise de Componentes Principais, do inglês <i>Principal Component Analysis</i>
RKHS	Espaços de Hilbert Reproduzidos por Kernel, do inglês <i>Reproducing Kernel Hilbert Spaces</i>
SVM	Máquina de Vetor de Suporte, do inglês <i>Support Vector Machine</i>
KPCA	Análise de Componentes Principais com Kernel, do inglês <i>Kernel Principal Component Analysis</i>
KTDSEP	Separação de Sinais utilizando Descorrelação Temporal com Kernel, do inglês <i>Kernel Temporal Decorrelation Signal Separation</i>
LMS	Mínimos Quadrados, do inglês <i>Least Mean Square</i>
KLMS	Mínimos Quadrados com Kernel, inglês <i>Kernel Least Mean Square</i>
ITL	Aprendizagem utilizando Teoria da Informação, do inglês <i>Information Theoretic Learning</i>
FDP	Função Densidade de Probabilidade
KDCA	Análise de Componentes Dependentes com Kernel, do inglês <i>Kernel Dependent Component Analysis</i>
BSS	Separação Cega de Sinais, do inglês <i>Blind Signal Separation</i>
RMSE	Raiz Quadrada do Erro Médio Quadrático, do inglês <i>Root-Mean-Squared Error</i>
CIP	Potencial de Informação Cruzada, do inglês <i>Cross Information Potential</i>
DIC	Decomposição Incompleta de Cholesky

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE TABELAS	x
LISTA DE SIGLAS	xi
CAPÍTULO 1	1
INTRODUÇÃO.....	1
CAPÍTULO 2	6
EXTRAÇÃO CEGA DE FONTES	6
2.1 Introdução.....	6
2.2 DCA – Análise de Componentes Dependentes.....	11
CAPÍTULO 3	15
RKHS – ESPAÇOS DE HILBERT REPRODUZIDOS POR KERNEL.....	15
3.1 Introdução.....	15
3.2 KPCA – Análise de Componentes Principais com Kerneis.....	17
3.3 Relações entre métodos kernelizados e aprendizagem baseada em Teoria da Informação.....	20
CAPÍTULO 4	24
KDCA – ANÁLISE DE COMPONENTES DEPENDENTES UTILIZANDO KERNEIS	24
4.1 Introdução.....	24
4.2 Comparação com KPCA	27
4.3 Simulação ilustrativa.....	28
4.4 Perspectiva de KDCA segundo ITL.....	29
CAPÍTULO 5	32
EXPERIMENTOS COMPUTACIONAIS PARA EXTRAÇÃO CEGA DE SINAIS	32
5.1 Introdução.....	32
5.2 Detalhes de implementação dos experimentos.....	32

5.3	Experimento 1: Extração de sinais periódicos	34
5.4	Experimento 2: Extração de sinais de FECG reais	36
5.5	Experimento 3: Redução de custo computacional utilizando <i>Decomposição Incompleta de Cholesky</i>	38
5.6	Experimento 4: Mistura não-linear.....	40
5.7	Discussão dos resultados.....	41
CAPÍTULO 6		43
CONCLUSÕES		43
REFERÊNCIAS		45
APÊNDICE		50

CAPÍTULO 1

INTRODUÇÃO

Em Processamento Digital de Sinais, subárea da Engenharia Elétrica, a filtragem é uma das operações mais básicas e fundamentais [2]. A filtragem consiste na modificação ou remoção de componentes ou características de um determinado sinal. Um exemplo de filtragens é a remoção de frequências indesejadas da Transformada de Fourier de um sinal, como a componente de 60 hertz da rede elétrica que é adquirida espuriamente nas medições de Eletrocardiograma (ECG). Outro exemplo é a eliminação do ruído de alta frequência de uma série temporal gerado em sinais de comunicação por processos eletrônicos aleatórios. Uma limitação deste tipo de filtragem baseada na análise da Transformada de Fourier é a necessidade de conhecimentos prévios do sinal desejado, que são inviáveis para os casos em que se observam apenas misturas de sinais estatisticamente independentes. É para a abordagem destes problemas que foi proposta a Extração Cega de Sinais.

A Extração Cega de Sinais (BSE, do inglês *Blind Signal Extractionion*) refere-se à operação de recuperar sinais tendo-se apenas as observações de misturas de diversas fontes [3]. Algumas das técnicas a partir das quais se tenta realizar BSE são a Análise de Componentes Independentes (ICA – do inglês *Independent Component Analysis*) [4], Redes Neurais Artificiais (RNA) [5], etc. Tal atenção se deve ao fato de BSE ser uma ferramenta potencialmente útil para diversas aplicações, como por exemplo, sistemas para aprimoramento da qualidade de sinais de fala, de telecomunicações e de processamento de sinais biomédicos [6].

Pode-se formular BSE da seguinte forma. Supondo-se um conjunto de m sinais originais $\mathbf{s}[k] = [s_1[k], s_2[k], \dots, s_m[k]]^T$ e uma transformação \mathbf{A} , tem-se na saída os seguintes sinais

$$\mathbf{x}[k] = \mathbf{A}\{\mathbf{s}[k]\}, \quad (1)$$

sendo $\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_m[k]]^T$. Em problemas práticos, tem-se apenas as misturas $\mathbf{x}[k]$ e deseja-se estimar $1 \leq l \leq m$ dos m sinais originais. Para este fim, deve-se estimar uma transformação \mathbf{W} que inverta (parte de) \mathbf{A} , fornecendo

$$\mathbf{y}[k] = \mathbf{W}\{\mathbf{x}[k]\}, \quad (2)$$

sendo $\mathbf{y}[k] = [y_1[k], y_2[k], \dots, y_l[k]]$. Na formulação original de ICA [7] as transformações \mathbf{A} e \mathbf{W} são lineares. Nesta situação podem-se estimar as formas de onda dos sinais $\mathbf{s}[k]$, porém, com uma indeterminação na variância (escalonamento) e na ordem em que eles são organizados (permutação).

Os diferentes métodos para estimar a transformação \mathbf{W} consistem em tornar o processo de mistura o menos cego possível [1], ou seja, fazer suposições sobre a estatística dos sinais originais de forma a explorá-las no processo de separação. As principais suposições feitas com este fim são independência estatística [8], não-gaussianidade [9] e/ou decorrelação temporal [10], este último que será considerado neste trabalho.

Porém, além de separar sinais explorando sua estrutura temporal, deseja-se que o sinal filtrado da mistura seja o desejado, ou seja, deseja-se superar a problemática da permutação de ICA. Barros e Cichocki em [11] mostraram como explorar uma estrutura temporal específica para extrair apenas o sinal desejado de um mistura de m sinais. Este tipo de solução é útil para os problemas em que vários sinais são misturados, porém poucos destes são de algum interesse. Por exemplo, em dispositivos de Eletroencefalografia (EEG) e Magnetoencefalografia (MEG) observa-se tipicamente mais que 64 sensores, porém poucas fontes originais são de alguma importância médica [3].

Diversos trabalhos relevantes foram desenvolvidos utilizando-se a solução linear de Barros e Cichocki [11] que ficou conhecida como Análise de Componentes Dependentes (DCA – do inglês *Dependent Component Analysis*) devido ao foco que se dá para os sinais auto-dependentes temporalmente. Barros e Cichocki [11] propuseram um método para extração do sinal de ECG fetal de gravações de uma mulher grávida, Estombelo et. al. propuseram um método para extração de sinais de Magnetograma (MGG) [12], além disso, vários novos métodos [13][14] para BSE foram desenvolvidos baseados em extensões deste método original.

Porém, a limitação do modelo linear impede a utilização de DCA, bem como de ICA e Análise de Componentes Principais (PCA – do inglês *Principal Component Analysis*) [15], da forma como originalmente proposto em diversas problemáticas. Objetivando torná-las mais abrangentes, sugeriram-se algumas extensões não-lineares de tais análises, tais como Redes Hebbianas [16], Perceptrons de múltiplas camadas auto-associativos [15], Curvas Principais [17], etc. Uma das abordagens não-lineares, aquela que será dada atenção no presente trabalho,

é baseada em Espaços de Hilbert Reproduzidos por Kernel (RKHS – do inglês *Reproducing Kernel Hilbert Spaces*) [18].

RKHS são espaços lineares de funcionais cujos produtos internos podem ser reproduzidos por kernels [19]. Ou seja, para uma função positiva semidefinida $\kappa(x, y)$, chamada kernel, definida no domínio dos dados x e y , tem-se que

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{H_\kappa}, \quad (3)$$

sendo ϕ a transformação que leva do espaço original dos dados x, y para o espaço de Hilbert H_κ dotado do produto interno $\langle \cdot, \cdot \rangle_{H_\kappa}$. Chama-se (3) de truque do kernel (tradução nossa do inglês *kernel trick*). Tendo-se este *truque*, podem-se estender algoritmos e análises que possam ser escritos em função de produtos internos para uma versão não-linear no espaço original dos dados, porém utilizando-se álgebra linear no espaço de Hilbert reproduzido.

A primeira aplicação famosa de tal estratégia foi o aperfeiçoamento da Máquina de Vetor de Suporte (SVM – do inglês *Support Vector Machine*) [20]. No contexto de separação de sinais, PCA ganhou uma versão *kernelizada* em [21] chamada Análise de Componentes Principais com Kernel (KPCA, do inglês *Kernel Principal Component Analysis*). KPCA também foi utilizado como etapa de pré-processamento para separação de sinais utilizando estruturas temporais no método Separação de Sinais utilizando Descorrelação Temporal com Kernel (KTDSEP, do inglês *Kernel Temporal Decorrelation Signal Separation*)[22]. KTDSEP será um dos métodos a serem comparados com o método que será proposto nesta dissertação. Além disso, tem-se também uma versão do algoritmo adaptativo de Mínimos Quadrados (LMS, do inglês *Least Mean Square*) chamado algoritmo de Mínimos Quadrados com Kernel (KLMS, do inglês *Kernel Least Mean Square*) [23].

Todos estes métodos baseados em RKHS seguem um mesmo padrão de derivação [24] mostrado na Tabela 1.1:

Tabela 1.1 – Como *kernelizar* um algoritmo

- | |
|---|
| <p>1° Escreve-se a versão linear do algoritmo em função de produtos internos dos dados, ou seja, em função da matriz de Gram dos dados</p> <p>2° Escolhe-se um kernel não-negativo</p> <p>3° Substitui-se os produtos internos lineares por produtos internos no espaço de Hilbert utilizando o kernel definido na 2° etapa, ou seja, utiliza-se como entrada a versão <i>kernelizada</i> da matriz de Gram dos dados</p> |
|---|

As vantagens de utilizar a versão *kernelizada* do algoritmo, ao invés da versão linear, são as possibilidades de explorar características não-lineares no conjunto de dados e operar classificações e agrupamentos que não podem ser realizadas linearmente [24]. Além disso, métodos de kernels são processadores práticos de informação sob a ótica da Aprendizagem utilizando Teoria da Informação (ITL, do inglês Information Theoretic Learning) [25], isto porque estimadores de Funções Densidade de Probabilidade (FDP) utilizando janelas de Parzen [26] podem ser interpretados como operações vetoriais em um espaço de kernels de Mercer [27].

Nesta dissertação serão apresentadas a *kernelização* e aplicação prática do algoritmo DCA [11], desta forma, ter-se-á a apresentação formal do método Análise de Componentes Dependentes com Kernel (KDCA, do inglês *Kernel Dependent Component Analysis*). Uma versão resumida deste trabalho foi previamente publicada em [28].

O restante desta monografia se organiza da seguinte forma:

- **Extração Cega de Sinais:** No Capítulo 2 será revisado o conceito de BSE. Esta revisão focará na ótica proposta pelo presente autor em [1], que apresenta uma extensão do trabalho de Cruces-Alvarez et. al. [29]. Neste capítulo, também, será apresentado a derivação e validação de DCA, para que seja *kernelizada* no Capítulo 4.
- **RKHS – Espaços de Hilbert Reproduzidos por Kernel:** No Capítulo 3 RKHS será apresentado formalmente, bem como o *truque do kernel* e como utilizá-lo em problemas práticos. Neste mesmo capítulo será mostrada, em caráter ilustrativo, a *kernelização* de PCA em KPCA. No Capítulo 4, ver-se-á que KDCA é uma extensão de KPCA para análise em diferentes atrasos de tempo. Em seguida, mostrar-se-á a relação entre RKHS e ITL.
- **Extração de Sinais com Estrutura Temporal Específica utilizando Kernels:** Este capítulo contém a principal contribuição desta dissertação: a apresentação da versão *kernelizada* do método de BSE baseado em análises temporais DCA. Conforme o anúncio, no Capítulo 4 será apresentado o algoritmo KDCA. A derivação será semelhante àquela realizada para DCA no Capítulo 2. Em seguida, o algoritmo para KDCA será interpretado sob a ótica de ITL.
- **Experimentos Computacionais para Extração Cega de Sinais:** Por fim, neste capítulo, realizam-se experiências comparativas entre KDCA e DCA para extração de sinais em misturas lineares, além de comparações entre KDCA e KTDSEP para

extração de sinais em misturas não-lineares. Na apresentação dos experimentos computacionais ficarão claras as vantagens e desvantagens de KDCA sobre DCA. Por exemplo, para estes experimentos, foram necessárias abordagens alternativas para estimação da matriz de Gram do kernel utilizando Decomposição Incompleta de Cholesky (DIC) [30] para reduzir custo computacional e acelerar a simulação.

- **Conclusões.** Neste Capítulo será mostrada uma sinopse desta dissertação, assim como, direções futuras para a pesquisa apresentada neste texto.

CAPÍTULO 2

SEPARAÇÃO CEGA DE FONTES

2.1 Introdução

Neste capítulo a mistura de m sinais discretos será considerada como um sistema linear quadrado, ou seja, o número de sinais misturados é igual ao número de misturas gravadas. Casos em que tem-se mais sinais originais que misturas (sobrecompleto) ou mais misturas (subcompleto) que sinais originais não serão considerados neste texto, e ao leitor interessado recomenda-se as referências [3][4] para o estudo dos casos subcompleto e sobrecompleto, recomenda-se [31][32][33] para a separação de sinais no caso em que se tem apenas um canal de gravação. Misturas não-lineares serão consideradas no Capítulo 4.

Considerando-se m sinais originais discretos, independentes, ergódicos e estacionários como $\mathbf{s}[k] = [s_1[k], s_2[k], \dots, s_m[k]]^T$, tal que $k = \{1, \dots, n\}$, e uma *matriz de mistura* quadrada e invertível $\mathbf{A}_{m \times m}$, tem-se o seguinte processo de mistura definido como

$$\mathbf{x}[k] = \mathbf{A}\mathbf{s}[k], \quad (4)$$

sendo $\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_m[k]]^T$, tal que $k = \{1, \dots, n\}$, com $x_i[k] = \sum_{j=1}^m a_{ij}s_j[k]$, sendo $\{a_{ij}\}_{i,j=1}^m$ os coeficientes da matriz \mathbf{A} .

A Separação Cega de Sinais (BSS, do inglês *Blind Signal Separation*) [3] consiste em recuperar todos os m sinais $s_i[k]$ tendo-se apenas $\mathbf{x}[k]$. BSE consiste em extrair l ($0 < l \leq m$) sinais $s_i[k]$ a partir de $\mathbf{x}[k]$, apenas. Quando $l = m$ BSE torna-se BSS. Tal *desmistura* é realizada como um processo linear, que consiste em definir uma matriz $\mathbf{W}_{m \times l}$ cujas colunas sejam estimações de parte das colunas da inversa de \mathbf{A} de tal forma que

$$\mathbf{y}[k] = \mathbf{W}^T \mathbf{x}[k], \quad (5)$$

sendo $\mathbf{y}[k] = [y_1[k], y_2[k], \dots, y_l[k]]^T$, tal que $k = \{1, \dots, n\}$, aproximação de l sinais $s_i[k]$ distintos, com $y_i[k] = \sum_{j=1}^m w_{ji} x_j[k]$, sendo $\{w_{ji}\}_{j=\{1, \dots, l\}}^{i=\{1, \dots, m\}}$ os coeficientes de \mathbf{W} . Desta forma quando $l = m$ tem-se $\mathbf{W}^T \mathbf{A} = \mathbf{I}_m$, sendo \mathbf{I}_m a matriz identidade de dimensão m . Porém, como mencionado no capítulo anterior, tem-se na prática uma indeterminação quanto ao escalonamento e a ordem dos sinais estimados [7]. Matematicamente, esta indeterminação é definida como $\mathbf{W}^T \mathbf{A} = \mathbf{D} \mathbf{P}$, sendo \mathbf{D} uma matriz diagonal de escalonamento e \mathbf{P} uma matriz de permutação de linhas. Para superar a indeterminação do escalonamento, assume-se que os sinais originais $s_i[k]$ tenham variância unitária e, desta forma, o conhecimento de \mathbf{D} torna-se desnecessários, pois os sinais extraídos $y_i[k]$ são normalizados para variância unitária durante o processo de *desmistura*.

Resolver o problema da permutação talvez seja uma das principais motivações para o desenvolvimento de BSE, uma vez que já se possuía BSS. Pois, quando diversos sinais estão misturados e apenas alguns são desejados, saber-se como extrair um sinal específico é fundamental. Na presente dissertação, detalhar-se-á a extração de um sinal por vez, processo este que pode ser estendido para a extração de l fontes por métodos deflacionários [34] ou paralelos [9].

O calculo de \mathbf{W} para a extração dos sinais desejados é o principal problema em BSE. E, uma vez que se tem apenas as misturas $\mathbf{x}[k]$, o calculo da matriz de *desmistura* exige algumas suposições quanto à estatística dos sinais misturados $\mathbf{s}[k]$ [1],[3],[4],[29]. Por exemplo, supondo-se que os sinais originais sejam processos estocásticos gaussianos e independentes, para *desmiturar* os sinais, basta descorrelacioná-los. O descorrelacionamento de sinais gaussianos pode ser realizado facilmente por PCA [15] da seguinte forma:

Da ergodicidade e estacionariedade dos processos, define-se a matriz de correlação dos sinais misturados como

$$\mathbf{R}_x = \frac{1}{n-1} \sum_{k=1}^n \mathbf{x}[k] \mathbf{x}[k]^T. \quad (6)$$

Como os sinais observados são correlacionados, \mathbf{R}_x deve ser não diagonal, porém simétrica e invertível. Deste modo, definindo-se $\mathbf{W}_{\text{PCA}} = \mathbf{R}_x^{1/2}$ descorrelaciona as misturas, pois, calculando-se a matriz de correlação de (5) tem-se

$$\begin{aligned}
\mathbf{R}_y &= \frac{1}{n} \sum_{k=1}^n \mathbf{W}_{\text{PCA}}^T \mathbf{x}[k] \mathbf{x}[k]^T \mathbf{W}_{\text{PCA}} \\
&= \mathbf{W}_{\text{PCA}}^T \left\{ \frac{1}{n} \sum_{k=1}^n \mathbf{x}[k] \mathbf{x}[k]^T \right\} \mathbf{W}_{\text{PCA}} \\
&= \mathbf{W}_{\text{PCA}}^T \mathbf{R}_x \mathbf{W}_{\text{PCA}} \\
&= \left(\mathbf{R}_x^{\frac{1}{2}} \right)^T \mathbf{R}_x \mathbf{R}_x^{\frac{1}{2}} \\
&= \mathbf{I}.
\end{aligned} \tag{7}$$

Pode-se ver em (7) porque a equação $\mathbf{y}[k] = \mathbf{W}_{\text{PCA}}^T \mathbf{x}[k]$ também é chamada de matriz branqueamento dos dados [4].

Outras suposições relevantes quanto à estatística dos sinais originais são não-gaussianidade [9] e decorrelação temporal [10]. A suposição de não-gaussianidade leva aos algoritmos de ICA de estatísticas de alta ordem chamados *Infomax* [35], *FastICA* [9] e *Jade* [36], dentre muitos outros [7] [8] [29][37]. A suposição de decorrelação temporal leva aos algoritmos de ICA de estatísticas de segunda ordem chamados *SOBI* [10], *AMUSE* [38] e *DCA* [11].

De maneira geral, um algoritmo para BSE segue as etapas básicas apresentadas na Tabela 2.1, apresentada originalmente pelo presente autor em [1].

Tabela 2.1 – Etapas básicas de um método para extração cega de fontes.

1. *Branquear* as observações $\mathbf{x}[k]$ utilizando PCA [15], ou seja, $\mathbf{x}[k] \leftarrow \mathbf{W}_{\text{PCA}}^T \mathbf{x}[k]$.
2. Estimar $\mathbf{s}[k]$ através de $\mathbf{y}[k] = \mathbf{W}^T \mathbf{x}[k]$.
3. Definir uma função contraste $J(\mathbf{y}[k])$ que quantize quão boa é a aproximação $\mathbf{y}[k] \approx \mathbf{s}[k]$, mesmo sem conhecimento de $\mathbf{s}[k]$.
4. Adaptar \mathbf{W} para melhorar a quantização da função contraste $J(\mathbf{y}[k])$.
5. Repetir o processo a partir de 2. até que $J(\mathbf{y}[k])$ assuma uma quantidade desejada.

Uma maneira de quantizar a não-gaussianidade de $\mathbf{y}[k]$ é utilizar a Negentropia [4]

$$J_{\text{Neg}}(y_i[k]) = \frac{1}{2} \log(2\pi e) - H(y_i), \tag{8}$$

sendo $H(y_i) = E[-\log p(y_i[k])]$ a entropia de Shannon [39]. Deste modo, (8) assume o valor 0 quando $y_i[k]$ é gaussiano. A maximização da não-gaussianidade através da maximização da Negentropia pode ser realizada através da regra de adaptação de \mathbf{W} do algoritmo *FastICA*:

$$\begin{aligned} \mathbf{w} &\leftarrow \sum_{k=1}^n \mathbf{x}[k]g(\mathbf{w}^T \mathbf{x}[k]) - \mathbf{w} \sum_{k=1}^n g'(\mathbf{w}^T \mathbf{x}[k]), \\ \mathbf{w} &\leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}, \end{aligned} \quad (9)$$

sendo $g(y_i)$ uma função que quantiza a não gaussianidade do sinal y_i , por exemplo $\tanh(y_i)$, e $g'(\cdot)$ é a derivada desta função.

A extração cega de fontes explorando estruturas temporais é baseada em decorrelacionar $\mathbf{x}[k]$ para diversos atrasos de tempo. Definindo-se a matriz de autocorrelação para diferentes atrasos como

$$\mathbf{R}_x(\tau) = \frac{1}{n-1-\tau} \sum_{k=1+\tau}^n \mathbf{x}[k]\mathbf{x}[k-\tau]^T, \quad (10)$$

pode-se quantizar o nível de correlação temporal dos sinais misturados como

$$J_{temporal}(y_i[k]) = \sum_{\tau} off(\mathbf{R}_x(\tau)), \quad (11)$$

sendo $off(\cdot)$ a função que retorna o somatório dos quadrados dos elementos de fora da diagonal principal da matriz passada como argumento, que é dada matematicamente como

$$off(\mathbf{R}) = \sum_{i,j|i \neq j} \mathbf{R}(i,j)^2. \quad (12)$$

Ou seja, quando uma matriz de correlação é passada como argumento, esta função quantiza o nível de correlação cruzada entre os sinais. Um método de adaptação de \mathbf{W} para tornar

$\mathbf{R}_x(\tau) = \mathbf{I}$ para vários atrasos τ , ou seja, minimizar (11) é a regra de adaptação do algoritmo *SOBI* [10]

$$\begin{aligned} \mathbf{W} &\leftarrow \mathbf{W} + \eta \sum_{\tau} \mathbf{B}_{\tau} \mathbf{W} \bar{\mathbf{R}}_x(\tau), \\ \mathbf{W} &\leftarrow GSort(\mathbf{W}), \end{aligned} \quad (13)$$

sendo, $\mathbf{B}_{\tau} = \text{diag}(\mathbf{W} \bar{\mathbf{R}}_x(\tau) \mathbf{W}^T)^{-1}$, $\bar{\mathbf{R}}_x(\tau) = \frac{1}{2}(\mathbf{R}_x(\tau) + \mathbf{R}_x(\tau)^T)$ e $GSort(\cdot)$ a função que executa a ortonormalização de Gram-Schmidt [40]. A vantagem da metodologia proposta em (13) com relação àquela proposta em (9) é o fato de ela utilizar apenas estatísticas de baixa ordem (médias e correlações). Porém, a desvantagem é a de exigir que os sinais envolvidos tenham estruturas temporais diferentes.

De maneira geral, para que uma função contraste, $J(\mathbf{y}[k])$, sirva para BSE de sinais não-gaussianos, ela deve satisfazer certas condições analisadas por Cruces-Alvarez et. al. [29]. Para que a função sirva para BSE tanto para sinais não-gaussianos quanto para sinais com estrutura temporal, a função deve satisfazer as condições analisadas em [1] e que são mostradas na Tabela 2.2 a seguir.

Tabela 2.2– Propriedades para funções contraste para extração cega de fontes

Propriedade 1: A função $J(\cdot)$ satisfaz uma *forma fraca de convexidade estrita*, que é definida com relação apenas à combinações lineares de componentes independentes.

Neste sentido, se $y_i[k] = \sum_{j=1}^m G_{ij} s_j[k]$ e $\sum_{j=1}^m |G_{ij}|^2 = 1$, então

$$J(\delta y_i[k]) \leq \sum_{j=1}^m |G_{ij}|^2 J(\delta s_j[k]) \quad (14)$$

sendo que $\delta y_i[k] = y_i[k] - f(y_i[k-1], y_i[k-2], \dots, y_i[k-\tau])$, $f(\cdot)$ é uma função que estima valores de $y_i[k]$ a partir de realizações anteriores e que para $J(\delta y_i[k]) > 0$ a igualdade se faz em (14) se somente se $y_i[k]$ for uma das componentes independentes.

Propriedade 2: A função $J(\cdot)$ é sempre não negativa e, por convenção, assume o valor zero quando seu argumento for uma variável aleatória com distribuição Gaussiana, ou seja, quando $y_i[k]$ for Gaussiano sem estrutura temporal.

Porém, nesta dissertação, o interesse está focado na extração de sinais com estrutura temporal. Na subseção seguinte, será apresentado o método DCA de maneira reformulada para que ele possa ser *kernelizado*, levando ao desenvolvimento do KDCA no Capítulo 4.

2.2 DCA – Análise de Componentes Dependentes

A abordagem de Barros e Cichocki para BSE assume que os sinais originais, ou fontes, como $\{\mathbf{s}[k] = [s_1[k], s_2[k], \dots, s_m[k]]^T, k \in \mathbf{N}\}$, sendo que T denota a transposição matricial e \mathbf{N} é um conjunto de índices. Neste modelo, as gravações, ou seja, os sinais misturados observados são representados como $\{\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_m[k]]^T, k \in \mathbf{N}\}$, sendo que a mistura é dada pela expressão linear $\mathbf{x}[k] = \mathbf{A}\mathbf{s}[k]$ como em (4), sendo que \mathbf{A} é quadrada de ordem $m \times m$ e invertível, porém desconhecida. Além disso, os sinais originais devem satisfazer as seguintes condições para um atraso conhecido τ e um sinal desejado $s_i[k]$

$$\begin{aligned} E[s_i[k]s_i[k - \tau]] &\neq 0 \\ E[s_i[k]s_j[k - \tau]] &= 0 \quad \forall j \neq i \\ E[s_h[k]s_j[k - \tau]] &= 0 \quad \forall h \neq j \end{aligned} \tag{15}$$

Para extrair-se linearmente o sinal $s_i[k]$, assumido como desejado, das misturas $\mathbf{x}[k]$, deve-se estimar um filtro \mathbf{w} tal que $y[k] = \mathbf{w}^T \mathbf{x}[k]$, sendo que $y[k]$ é uma estimaco de $s_i[k]$ normalizado para varincia unitria, conforme comentado previamente sobre a indeterminaco do escalonamento [7]. Este problema do escalonamento é resolvido em DCA, assim como ICA, impondo-se uma restrio na estimaco de \mathbf{w} . Restrio, esta que pode ser expressa matematicamente como

$$E[y^2[k]] = 1, \tag{16}$$

que pode ser reescrita como

$$\mathbf{w}^T \mathbf{R}_x \mathbf{w} = 1, \tag{17}$$

sendo \mathbf{R}_x a matriz de correlação das misturas dada por (6).

Sem perda de generalidade, assume-se que o conjunto de índices \mathbf{N} é finito e com os seguinte elementos $\{1, 2, \dots, n\}$ e que \mathbf{w} pode ser escrito como uma combinação linear dos vetores $\mathbf{x}[k]$, ou seja, assume-se que o filtro pertença ao mesmo espaço gerado pelos vetores de gravação por combinações lineares. Deste modo, existe um vetor de coeficientes $\mathbf{c} = [c_1, c_2, \dots, c_n]^T$, tal que

$$\mathbf{w} = \sum_{i=1}^n c_i \mathbf{x}[i] \quad (18)$$

e

$$\begin{aligned} y[k] &= \mathbf{w}^T \mathbf{x}[k] = \sum_{i=1}^n c_i \mathbf{x}[i]^T \mathbf{x}[k], \\ &= \mathbf{K}(k, :) \mathbf{c}, \end{aligned} \quad (19)$$

sendo que $\mathbf{K}(k, i) = \mathbf{x}(k)\mathbf{x}(i)$ é a matriz $n \times n$ de Gram dos sinais misturados e $\mathbf{K}(k, :)$ é a k -ésima linha de \mathbf{K} . Além disso, a restrição (17) pode ser reescrita como

$$\mathbf{c}^T \mathbf{K} \mathbf{c} = 1. \quad (20)$$

Utilizando-se as definições (18) e (19) e a informação sobre a função de autocorrelação de $s_i[k]$ para o atraso τ em (15), define-se a seguinte função contraste a ser maximizada

$$\begin{aligned} J(\mathbf{c}) &= E[y(t)y(t - \tau)], \\ &= \frac{1}{n} \sum_{i, k \in \mathbf{N}} c_i c_k \mathbf{K}(k, i) \mathbf{K}(k - \tau, i), \\ &= \frac{\mathbf{c}^T \mathbf{K}^T \mathbf{K}_\tau \mathbf{c}}{n}, \end{aligned} \quad (21)$$

sendo \mathbf{K}_τ definida como a versão circularmente transladada das linhas de \mathbf{K} , τ vezes, ou seja,

$$\mathbf{K}_\tau(k, i) = \begin{cases} \mathbf{K}(k - \tau, i), & \text{se } k > \tau \\ \mathbf{K}(n + k - \tau, i), & \text{se } k \leq \tau \end{cases} \quad (22)$$

Prova-se que maximizar $J(\mathbf{c})$ com a restrição (20) é suficiente para extrair $s_i[k]$ da seguinte forma: assumindo-se, por simplicidade, que os dados $\mathbf{x}[k]$ foram branqueados com PCA e renormalização, (21) pode ser escrita como

$$\mathbf{E}[y[k]y[k - \tau]] = \mathbf{w}^T \mathbf{A} \mathbf{E}[\mathbf{s}[k]\mathbf{s}[k - \tau]^T] \mathbf{A}^T \mathbf{w}. \quad (23)$$

Utilizando (15), sabe-se que $\mathbf{E}[\mathbf{s}[k]\mathbf{s}[k - \tau]^T]$ é uma matriz diagonal com todos os elementos iguais a zero, exceto o i -ésimo que, pela suposição de renormalização, é igual a 1. Portanto,

$$\mathbf{w}^T \mathbf{A} \mathbf{E}[\mathbf{s}[k]\mathbf{s}[k - \tau]^T] \mathbf{A}^T \mathbf{w} = \mathbf{w}^T \mathbf{A}(:, i) \mathbf{A}^T(:, i) \mathbf{w}, \quad (24)$$

sendo $\mathbf{A}(:, i)$ a i -ésima coluna de \mathbf{A} . Pelo branqueamento de $\mathbf{x}[k]$, tem-se

$$\mathbf{E}[\mathbf{x}[k]\mathbf{x}[k - \tau]^T] = \mathbf{A} \mathbf{E}[\mathbf{s}[k]\mathbf{s}[k - \tau]^T] \mathbf{A}^T = \mathbf{A} \mathbf{A}^T = \mathbf{I}, \quad (25)$$

sendo \mathbf{I} a matriz identidade. Logo, sendo \mathbf{A} ortogonal, extremizar (24) é o mesmo que tornar \mathbf{w} correlacionado com a i -ésima linha da inversa de \mathbf{A} , ou seja, a i -ésima linha de \mathbf{A}^T , o que suficiente para extrair $s_i[k]$, ou seja, é suficiente para tornar $y[k] \approx s_i[k]$.

A maximização de (21) pode ser realizada facilmente derivando-se $J(\mathbf{c})$ com relação a \mathbf{c} e utilizando multiplicadores de Lagrange para definir um sistema de equações diferenciais. De fato, o gradiente de $J(\mathbf{c})$ será

$$\frac{\partial J(\mathbf{c})}{\partial \mathbf{c}} = \frac{(\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}) \mathbf{c}}{n}, \quad (26)$$

portanto, tem-se o sistema de equações diferenciais da seguinte forma

$$\begin{cases} (\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}) \mathbf{c} = 2\lambda n \mathbf{K}^2 \mathbf{c} \\ \mathbf{c}^T \mathbf{K} \mathbf{c} - 1 = 0 \end{cases}, \quad (27)$$

sendo λ um multiplicador de Lagrange.

Hyvärinen e Oja mostraram em [9] como solucionar numericamente sistemas como aquele mostrado em (27). Tal solução consiste em uma regra de aprendizado de ponto-fixo em dois passos. O primeiro passo adapta a variável para que ela seja igual ao gradiente da função contraste e o segundo passo renormaliza a variável adaptada. Aplicando-se esta estratégia, soluciona-se o sistema numericamente com a seguinte regra

$$\begin{cases} \mathbf{c} \leftarrow (\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}) \mathbf{c} \\ \mathbf{c} \leftarrow \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{K}^2 \mathbf{c}}} \end{cases}, \quad (28)$$

sendo que omitiu-se $2\lambda n$ em (28) devido à renormalização tornar tal escalonamento irrelevante.

O autovetor correspondente ao maior autovalor de $\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}$ é um ponto atrator da regra adaptativa (28), como mostrado em [41]. Portanto, outra solução para (27) é dada por

$$\mathbf{c} \leftarrow \text{eig}(\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}), \quad (29)$$

sendo que a função $\text{eig}(\cdot)$ retorna o autovetor correspondente ao maior autovalor da matriz passada como argumento de entrada. Pode-se, além disso, estender este algoritmo para torná-lo mais robusto, assim como realizado por Zhang e Yi [14], da seguinte forma

$$\mathbf{c} = \text{eig}\left((\mathbf{K}^T \mathbf{K}_\tau + \mathbf{K}_\tau^T \mathbf{K}) + (\mathbf{K}^T \mathbf{K}_{2\tau} + \mathbf{K}_{2\tau}^T \mathbf{K}) + \dots + (\mathbf{K}^T \mathbf{K}_{p\tau} + \mathbf{K}_{p\tau}^T \mathbf{K})\right), \quad (30)$$

sendo p uma constante escolhida previamente, normalmente 3 ou 4.

Os métodos propostos para DCA neste capítulo, mais especificamente, as soluções numéricas (28) e (30), podem ser estendidas para uma versão não-linear utilizando kernels. Para realizar este avanço, primeiramente precisa-se analisar com mais detalhes o que são RKHS e como utilizá-los na prática, o que será realizado no Capítulo 3 seguinte. Logo, a definição de DCA não linear utilizando kernels será adiada para o Capítulo 4.

CAPÍTULO 3

RKHS – ESPAÇOS DE HILBERT REPRODUZIDOS POR KERNEL

3.1 Introdução

Um RKHS é um espaço de Hilbert [42] no qual todos os funcionais lineares são limitados. Assumindo-se, por exemplo, que \mathcal{H} seja um espaço funcional de Hilbert em dado domínio \mathfrak{S} , dotado do produto interno $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, tem-se que para todo $t \in \mathfrak{S}$, pode-se calcular $f(t)$ como uma operação linear em \mathcal{H} . Ou seja, existe um elemento $g_t \in \mathcal{H}$, tal que

$$f(t) = \langle g_t, f \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}. \quad (31)$$

O Teorema de Moore-Aronszajn [43] estabelece que para toda função positivo-definida $\kappa(\cdot, \cdot)$, em $\mathfrak{S} \otimes \mathfrak{S}$, existe um RKHS unicamente associado e virse-versa. Desta forma, tem-se que, para $s, t \in \mathfrak{S}$ e representantes $g_s, g_t \in \mathcal{H}$

$$\langle g_s, g_t \rangle = \kappa(s, t), \quad (32)$$

logo, pode-se definir o representante de operação como $g_t \stackrel{\text{def}}{=} \kappa(t, \cdot)$ e $\langle \kappa(t, \cdot), \kappa(s, \cdot) \rangle \stackrel{\text{def}}{=} \kappa(s, t)$, sendo (32) a equação que originou o termo “reproduzido por kernel”. Também, (32) é conhecida na literatura de aprendizagem de máquina como *truque do kernel*. Assim sendo, pode-se construir o espaço \mathcal{H} como o conjunto de todas as combinações lineares finitas de $\sum a_j \kappa(t_j, \cdot)$, e seus limites sob a norma induzida pelo produto interno $\langle \kappa(t, \cdot), \kappa(s, \cdot) \rangle = \kappa(s, t)$, ou seja

$$\mathcal{H} = \{ \sum a_j \kappa(t_j, \cdot) \mid a_j \in \mathbb{R}, t_j \in \mathfrak{S} \}, \quad (33)$$

A exigência de que o kernel, κ , que reproduz o espaço \mathcal{H} seja positivo definido vem das definições de *norma* [42]. Pois a *norma* $\|\cdot\|_{\mathcal{H}}$ satisfaz as seguintes condições $\forall f, g \in \mathcal{H}$ e $a \in \mathbb{R}$

$$\begin{aligned}
\|f\|_{\mathcal{H}} &\geq 0 \text{ e } \|f\|_{\mathcal{H}} = 0 \text{ sse } f = 0; \\
\|f + g\|_{\mathcal{H}} &\leq \|f\|_{\mathcal{H}} + \|g\|_{\mathcal{H}}; \\
\|af\|_{\mathcal{H}} &= |a|\|f\|_{\mathcal{H}};
\end{aligned} \tag{34}$$

além disso, a *norma* $\|\cdot\|_{\mathcal{H}}$ pode ser definida via produto interno como

$$\|g_t\|_{\mathcal{H}} = \sqrt{\langle g_t, g_t \rangle_{\mathcal{H}}}, \tag{35}$$

portanto, sendo que o produto interno em \mathcal{H} é definido como (32), segue-se que κ deve ser positivo definido, pois, lembrando-se que κ será positivo definido se para finitos $a_i, a_j \in \mathbb{R}$ e $t_i, t_j \in \mathfrak{X}$ se

$$\sum_{i,j} a_i a_j \kappa(t_i, t_j) \geq 0, \tag{36}$$

seguinto-se que

$$\sum_{i,j} a_i a_j \kappa(t_i, t_j) = \sum_{i,j} a_i a_j \langle \kappa(t_i, \cdot), \kappa(t_j, \cdot) \rangle_{\mathcal{H}} = \left\| \sum_i a_i \kappa(t_i, \cdot) \right\|_{\mathcal{H}}^2 \geq 0, \tag{37}$$

que justifica a exigência mencionada.

Diz-se também que um RKHS tem as funcionais limitadas pois para todo funcional $F_t: \mathcal{H} \rightarrow \mathbb{R}$, existe M , tal que

$$|F_t[f]| = |f(t)| \leq M \|f\|_{\mathcal{H}} \quad \forall f \in \mathcal{H}. \tag{38}$$

Nota-se que o conhecido espaço \mathcal{L}_2 de funções de quadrado integrável não é de funcionais limitadas, pois, modificando-se uma função em um número contável de pontos a integral $\int |f(t)|^2 dt$ matem-se a mesma. Portanto, o espaço em que se trabalha a Transformada de Fourier é um espaço de Hilbert, mas não um RKHS.

Alguns exemplos de kernels simétricos utilizados em problemas de aprendizagem de máquina, tais como, classificação, agrupamento e PCA são

o kernel linear

$$\kappa(t_i, t_j) = t_i \cdot t_j, \quad (39)$$

o kernel polinomial

$$\kappa_d(t_i, t_j) = (t_i \cdot t_j + 1)^d, \quad d \in \mathbb{N} \quad (40)$$

e o kernel Gaussiano, que será utilizado neste trabalho,

$$\kappa_\sigma(t_i, t_j) = (2\pi\sigma^2)^{-1/2} e^{-\frac{\|t_i - t_j\|^2}{\sigma^2}}, \quad \sigma > 0. \quad (41)$$

A dimensão do RKHS depende do kernel. O espaço reproduzido pelo kernel Gaussiano (41) tem dimensão infinita [44], que oferece uma grande vantagem para classificação graças ao Teorema de Cover [45].

A vantagem de utilizar-se o kernel gaussiano no presente contexto, como será comentado mais adiante, é o fato de o método assim *kernelizado* poder ser visto como um processador de informação de Renyi [46], sob a ótica do Janelamento de Parzen [26] e ITL [44].

A seguir, mostra-se o método KPCA para ilustrar a utilização de RKHS na prática.

3.2 KPCA – Análise de Componentes Principais com Kerneis

Para definir KPCA, Schölkopf et. al. [21] primeiro reescreveram PCA em termos de produtos internos e então substituíram os produtos internos lineares por produtos internos em um RKHS utilizando o *truque do kernel*, da seguinte forma. Seja um conjunto de n observações $\mathbf{x}[k]$, $k = 1, \dots, n$, $\mathbf{x}[k] \in \mathbb{R}^m$, $\sum_{k=1}^n \mathbf{x}[k] = 0$, o objetivo de PCA é tornar a matriz de covariância destas observações uma matriz diagonal, como comentado no Capítulo 2. Sendo a matriz de correlação $\mathbf{R}_x = \frac{1}{n} \sum_{j=1}^n \mathbf{x}[j] \mathbf{x}[j]^T$, para diagonalizá-la, precisa-se primeiro resolver a seguinte equação de autovalores desta matriz

$$d\mathbf{v} = \mathbf{R}_x \mathbf{v}, \quad (42)$$

sendo $d \geq 0$ os autovalores e $\mathbf{v} \in \mathbb{R}^m \setminus \{0\}$ os autovetores. Uma vez que $\mathbf{R}_x \mathbf{v} = \frac{1}{n} \sum_{j=1}^n (\mathbf{x}[j]^T \mathbf{v}) \mathbf{x}[j]$, todas os autovetores \mathbf{v} devem pertencer ao espaço expandido por todas as combinações lineares finitas das observações $\mathbf{x}[1], \dots, \mathbf{x}[n]$, isto é equivalente a dizer que existe um vetor de coeficientes $\mathbf{a} = [a_1, \dots, a_n] \in \mathbb{R}^n$ tal que

$$\mathbf{v} = \sum_{i=1}^n a_i \mathbf{x}[i]. \quad (43)$$

Portanto, multiplicando-se (42) à esquerda por $\mathbf{x}[k]$ e aplicando-se (44), chega-se à seguinte equação $\mathbf{R}_x \mathbf{v} = \frac{1}{n} \sum_{j=1}^n (\mathbf{x}[j]^T \mathbf{v}) \mathbf{x}[j]$

$$\begin{aligned} d \mathbf{x}[k]^T \mathbf{v} &= \mathbf{x}[k]^T \mathbf{R}_x \mathbf{v} \\ d \mathbf{x}[k]^T \sum_{i=1}^n a_i \mathbf{x}[i] &= \mathbf{x}[k]^T \mathbf{R}_x \sum_{i=1}^n a_i \mathbf{x}[i] \\ d \sum_{i=1}^n a_i \mathbf{x}[k]^T \mathbf{x}[i] &= \frac{1}{n} \mathbf{x}[k]^T \sum_{j=1}^n \sum_{i=1}^n (a_i \mathbf{x}[j]^T \mathbf{x}[i]) \mathbf{x}[j], \end{aligned} \quad (44)$$

Definindo-se matriz de Gram $\mathbf{K}(i, j) = \mathbf{x}[j]^T \mathbf{x}[i]$ e aplicando-se em (44), chega-se a

$$\begin{aligned} dn \mathbf{K} \mathbf{a} &= \mathbf{K}^2 \mathbf{a} \\ dn \mathbf{K}^{-1} \mathbf{K} \mathbf{a} &= \mathbf{K}^{-1} \mathbf{K} \mathbf{a} \\ dna &= \mathbf{K} \mathbf{a}, \end{aligned} \quad (45)$$

sendo que a solução da equação de autovalores (45) para d e \mathbf{a} é equivalente à equação (42), porém, com a vantagem daquela ser escrita toda em termos de produtos internos, ou seja, em termos da matriz de Gram \mathbf{K} e não da matriz de correlação \mathbf{R}_x que exige o calculo de produtos internos.

Após resolver este problema de autovalores, calculam-se as componentes principais como as projeções dos dados nos autovetores \mathbf{v} , ou seja, as componentes principais serão

$$\begin{aligned} y[k] &= \mathbf{v}^T \mathbf{x}[k], \\ &= \sum_{j=1}^n a_j \mathbf{x}[j]^T \mathbf{x}[k] \\ &= \mathbf{K}(k, :) \mathbf{a} \end{aligned} \quad (46)$$

sendo $\mathbf{K}(k, :)$ a k -ésima linha da matriz \mathbf{K} . Portanto, vê-se que as componentes principais nada mais são que os coeficientes \mathbf{a} que definem os autovetores multiplicados pelo matriz \mathbf{K} , sendo que ambos podem ser calculados apenas em função de produtos internos dos dados observados.

A desvantagem de trabalhar-se com produtos externos, é que estes exigem acesso direto ao espaço de vetores em que se trabalha. Utilizando-se apenas produtos internos como em (45), pode-se definir uma transformação não-linear $\Psi: \mathbb{R}^m \rightarrow \mathcal{H}_\kappa$ para um RKHS \mathcal{H}_κ reproduzido pelo kernel κ , ou seja,

$$\mathbf{x}_k \mapsto \Psi(\mathbf{x}[k]), \quad (47)$$

que devido à propriedade da reprodutibilidade, permite-se que se escreva (47) como

$$\mathbf{x}[k] \mapsto \kappa(\mathbf{x}[k], \cdot). \quad (48)$$

Do mesmo modo, o produto interno em \mathcal{H}_κ pode ser escrito como

$$\bar{\mathbf{K}}(i, j) = \langle \Psi(\mathbf{x}[j]), \Psi(\mathbf{x}[i]) \rangle_{\mathcal{H}_\kappa} = \kappa(\mathbf{x}[j], \mathbf{x}[i]), \quad (49)$$

sendo $\bar{\mathbf{K}}$ a matriz de Gram das observações no RKHS. É fácil perceber que tendo-se (49) pode-se facilmente calcular os autovalores e portanto, a análise de componentes principais, no espaço de Hilbert expandido por todas as combinações lineares das observações transformadas. Ou seja, definindo-se as autofunções, nome dado a um autovetor em um espaço de funções, do tensor de autocorrelação no RKHS como

$$\boldsymbol{\omega} = \sum_{i=1}^n \alpha_i \Psi(\mathbf{x}[i]) = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}[i], \cdot), \quad (50)$$

tem-se que as componentes principais das observações $\Psi(\mathbf{x}[1]), \dots, \Psi(\mathbf{x}[n])$ serão os vetores $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]$, que são solução da equação de autovalores em RKHS

$$\begin{aligned}
d\langle \Psi(\mathbf{x}[k]), \boldsymbol{\omega} \rangle_{\mathcal{H}_k} &= \langle \Psi(\mathbf{x}[k]) | \mathbf{R}_{\Psi(\mathbf{x}[k])} | \boldsymbol{\omega} \rangle_{\mathcal{H}_k} \\
d \sum_{i=1}^n \alpha_i \langle \Psi(\mathbf{x}[k]), \Psi(\mathbf{x}[i]) \rangle_{\mathcal{H}_k} &= \sum_{i=1}^n \alpha_i \langle \Psi(\mathbf{x}[k]) | \mathbf{R}_{\Psi(\mathbf{x}[k])} | \Psi(\mathbf{x}[i]) \rangle_{\mathcal{H}_k} \\
dn\bar{\mathbf{K}}\boldsymbol{\alpha} &= \bar{\mathbf{K}}^2\boldsymbol{\alpha}, \\
dn\boldsymbol{\alpha} &= \bar{\mathbf{K}}\boldsymbol{\alpha}
\end{aligned} \tag{51}$$

Por fim, as componentes principais não-lineares, com relação ao espaço original dos dados, mas lineares em RKHS, serão dadas por

$$\begin{aligned}
\bar{y}[k] &= \langle \Psi(\mathbf{x}[k]), \boldsymbol{\omega} \rangle_{\mathcal{H}_k} \\
&= \sum_{j=1}^n \alpha_j \langle \Psi(\mathbf{x}[k]), \Psi(\mathbf{x}[j]) \rangle_{\mathcal{H}_k} \\
&= \bar{\mathbf{K}}(k, :) \boldsymbol{\alpha},
\end{aligned} \tag{52}$$

sendo $\bar{\mathbf{K}}(k, :)$ a k -ésima linha da matriz $\bar{\mathbf{K}}$. Portanto, vê-se que pode-se calcular componentes principais não-lineares utilizando álgebra linear em RKHS. As vantagens e desvantagens de utilizar-se (52) para classificação, separação de fontes, agrupamento, etc. são comentadas em detalhes em [21].

Tal desenvolvimento de KPCA é realizado considerando-se que os dados observados preservam-se com média zero depois que são transformados não-linearmente para o RKHS. Porém, tal suposição não é normalmente válida, entanto, pod-se centralizar os dados transformados com a seguinte operação na matriz de Gram $\bar{\mathbf{K}}$ [21]

$$\bar{\mathbf{K}} \leftarrow \bar{\mathbf{K}} - \mathbf{1}_{n \times n} \cdot \bar{\mathbf{K}}/n - \bar{\mathbf{K}} \cdot \mathbf{1}_{n \times n}/n + \mathbf{1}_{n \times n} \cdot \bar{\mathbf{K}} \cdot \mathbf{1}_{n \times n}/n, \tag{53}$$

Na seção seguinte, comenta-se as características dos métodos baseados em kernels segundo o ponto de vista de ITL.

3.3 Relações entre métodos kernelizados e aprendizagem baseada em Teoria da Informação

A principal motivação para o desenvolvimento de métodos de aprendizagem de máquina, ou seja, algoritmos adaptativos, baseados em teoria da informação foi analisar estatísticas que vão além das de primeira, segunda e mesmo algumas estatísticas de altas

ordens. Por exemplo, ITL busca lidar diretamente com a entropia do erro, ao invés de um número finito de momentos para adaptar algoritmos de filtragem, classificação, etc. [47]. E, como se sabe, a entropia é uma quantidade associada a uma FDP que caracteriza quão aleatória é uma variável aleatória, ou seja, quanto de surpresa tem-se ao observar amostras de uma variável aleatória [48].

A primeira definição de informação e entropia, no contexto de processamento de sinais, foi proposta por Claude Shannon [39]. Porém a estimação da entropia de Shannon a partir de um número finito de observações exige sérias parametrizações e limitações no conjunto de dados observados, por este motivo, ITL foi desenvolvida a partir de definições mais genéricas de entropia, mais explicitamente, a entropia de Renyi [46], que é dada por

$$H_\alpha(x) = \frac{1}{1-\alpha} \log \int f^\alpha(x) dx, \quad \alpha > 0, \quad (54)$$

sendo $f(x)$ a FDP da variável aleatória contínua X e α a ordem da entropia. Além disso, tem-se que H_α tende à entropia de Shannon quando $\alpha \rightarrow 1$. A vantagem de tal definição de entropia, é que a entropia de segunda ordem pode ser diretamente estimada a partir de um conjunto finito de amostras da variável aleatória X utilizando-se o estimador de FDPs não-paramétrico de Parzen[26]. Ou seja, dada uma função simétrica positivo definida $\kappa(\cdot, \cdot)$, um estimador não-paramétrico, assintoticamente não-enviesado e consistente de f é dado por

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \kappa(x_i, x) \quad (55)$$

sendo x_1, \dots, x_n , $n \in \mathbb{N}$ um conjunto finito de amostras da variável aleatório X e κ escolhido de tal forma que $\int \hat{f}(x) dx = 1$. Aplicando-se (55) em (54) para $\alpha = 2$, tem-se a estimação da entropia diretamente do conjunto de dados

$$\hat{H}_2(\mathbf{x}) = -\log \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \kappa(x_i, x_j). \quad (56)$$

Porém, para finalidades adaptativas, devido à monotonicidade da função logaritmo, importa apenas o argumento de (56), ou seja,

$$\hat{V}(f) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \kappa(x_i, x_j), \quad (57)$$

sendo $\hat{V}(f)$ conhecida como a estimação do potencial de informação de f , devido à semelhança aos campos ponteciais da física [44].

Para análises multivariáveis, definiu-se o Potencial de Informação Cruzado (CIP, do inglês *Cross Information Potential*), dado por

$$V(f_1, f_2) = \int f_1(x)f_2(x)dx, \quad (58)$$

que pode ser estimado a partir de um conjunto finito de dados como x_{1i} e x_{2i} de f_1 e f_2 , respectivamente

$$\hat{V}(f_1, f_2) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \kappa(x_{1i}, x_{2j}), \quad (59)$$

que é a base fundamental das medidas de divergência entre FDPs dadas pela distância Euclidiana, definida como

$$\begin{aligned} D_{ED}(f_1, f_2) &= \int (f_1(x) - f_2(x))^2 dx \\ &= \int f_1^2(x)dx - 2 \int f_1(x)f_2(x)dx + \int f_2^2(x)dx, \end{aligned} \quad (60)$$

e a divergência entre FDPs baseada na inequação de Cauchy-Schwarz, que pode ser escrita como

$$D_{CS}(f_1, f_2) = -\log \frac{\int f_1(x)f_2(x)dx}{\sqrt{(\int f_1^2(x)dx)(\int f_2^2(x)dx)}}. \quad (61)$$

Nota-se que ambas, D_{ED} e D_{CS} são maiores que ou iguais a zero, e a igualdade só se faz se e somente se $f_1(x) = f_2(x)$. Dada a participação fundamental do CIP $V(f_1, f_2)$, ITL é definida baseada no RKHS, \mathcal{H}_V , reproduzido por ele.

A relação entre os RKHS \mathcal{H}_κ sobre os quais os métodos baseados em kernels são definidos e o RKHS de ITL, está no fato de que os dados transformados $\mathbf{x}[k] \mapsto \kappa(\mathbf{x}[k], \cdot)$ são estocásticos enquanto que as funcionais de FDPs transformadas são determinísticas. Deste modo no RKHS de ITL apenas álgebra é necessária para se realizarem inferências estatísticas [49]. Deste modo, através do cálculo de esperanças em \mathcal{H}_κ , define-se um novo kernel neste espaço, que leva a \mathcal{H}_V .

Portanto, podem-se analisar os métodos de kernel quando estes são janelas de Parzen e observar que operações algébricas são realizadas entre as FDPs dos dados observados. PCA, por exemplo, é motivada pela busca por uma transformação linear que extraia as componentes ortogonais com maiores variâncias possíveis. KPCA, portanto, busca pelas direções de maiores variâncias no RKHS \mathcal{H}_κ . Enquanto isso, KPCA analisada do ponto de vista de \mathcal{H}_V , busca pelas direções de maior entropia de segunda ordem de Renyi, pois se maximiza $\int f^2(\mathbf{x})d\mathbf{x}$, utilizando a estimação $\sum \alpha_i \kappa(\mathbf{x}[i], \mathbf{x}[k]) = \bar{\mathbf{K}}(k, :)\boldsymbol{\alpha}$, sendo $\hat{f}(\mathbf{x}) = \sum_i \alpha_i \kappa(\mathbf{x}, \mathbf{x}[i])$ a estimação de $f(\mathbf{x})$ por Janelamento Ponderado de Parzen [27].

Deste modo, considerando-se o janelamento ponderado de Parzen e o RKHS \mathcal{H}_V reproduzido pelo CIP, traça-se o paralelo entre métodos baseados em RKHS e ITL, de onde se vê tais métodos são processadores práticos de informação.

De posse de tais informações, no capítulo seguinte desenvolve-se a versão *kernelizada* de DCA, chamada KDCA, que se espera seja, igualmente, um processador prático de informação segundo a teoria de ITL e forneça melhores resultados que outros métodos propostos na literatura para BSE.

CAPÍTULO 4

KDCA – ANÁLISE DE COMPONENTES DEPENDENTES UTILIZANDO KERNEIS

4.1 Introdução

Para *kernelizar* DCA, assume-se que os sinais originais são conjuntos discretos de números reais e os sinais misturados sejam definidos em um espaço de Hilbert \mathcal{H}_κ que pode ser reproduzido pelo kernel κ , conforme comentado no capítulo anterior. Desta forma, os sinais originais são mapeados linearmente para \mathcal{H}_κ , e, então, misturados linearmente. Portanto, assumindo-se este processo de mistura linear e que os sinais resultantes sejam um conjunto de funções de dimensões (possivelmente) infinitas, denotados como $\{\boldsymbol{\zeta}(t) = [\zeta_1[k], \zeta_2[k], \dots, \zeta_m[k]]^T, k \in \mathbf{N}\}$. Os sinais originais são escritos como no modelo linear $\{\mathbf{s}[k] = [s_1[k], s_2[k], \dots, s_m[k]]^T, k \in \mathbf{N}\}$, sendo n o número de elementos no conjunto de índices \mathbf{N} . Precisa-se, tendo as definições acima, estimar o sinal original $s_i[k]$ específico através de

$$y[k] = \langle \boldsymbol{\omega}, \boldsymbol{\zeta}[k] \rangle_{\mathcal{H}_\kappa} \quad (62)$$

, sendo $\langle \cdot, \cdot \rangle_{\mathcal{H}_\kappa}$ o produto interno do qual \mathcal{H}_κ é dotado e $\boldsymbol{\omega} \in \mathcal{H}_\kappa$ uma função de desmistura.

Conhecendo-se um atraso τ que satisfaça

$$\begin{aligned} E[s_i[k]s_i[k - \tau]] &\neq 0 \\ E[s_i[k]s_j[k]] &= 0 \quad \forall i \neq j \\ E[s_i[k]s_j[k - \tau]] &= 0 \quad \forall i \neq j \end{aligned} \quad (63)$$

e assumindo-se que $\boldsymbol{\omega}$ pertença a \mathcal{H}_ζ , um subespaço de \mathcal{H}_κ expandido por todas as combinações lineares finitas dos tensores $\boldsymbol{\zeta}[k]$. Logo, existe um vetor $\mathbf{b} = [b_1, b_2, \dots, b_n]^T$ satisfazendo

$$\boldsymbol{\omega} = \sum_{i=1}^n b_i \boldsymbol{\zeta}[i] \quad (64)$$

e, portanto, (41) assume a forma

$$\begin{aligned} y[k] &= \left\langle \sum_{i=1}^n b_i \zeta[i], \zeta[k] \right\rangle_{\mathcal{H}_\kappa} \\ &= \bar{\mathbf{K}}(k, :) \mathbf{b} \end{aligned} \quad (65)$$

sendo $\bar{\mathbf{K}}[k, i] = \langle \zeta[i], \zeta[k] \rangle$ a matriz de Gram no RKHS.

Em analogia à análise realizada no Capítulo 2, restringe-se a variância do sinal extraído como $E[y^2[k]] = 1$, que resulta em

$$E[y^2[k]] = 1 \quad (66)$$

ou seja,

$$\langle \boldsymbol{\omega} | \bar{\mathbf{R}}_\zeta | \boldsymbol{\omega} \rangle = 1 \quad (67)$$

sendo, a norma $\bar{\mathbf{R}}_\zeta$ dada como a função de correlação dos tensores $\boldsymbol{\zeta}[k]$, dada por $\bar{\mathbf{R}}_\zeta(i, j) = E[\langle \zeta_i[k], \zeta_j[k] \rangle]$. Seguindo a análise, pode-se reescrever (67) como

$$\mathbf{b}^T \bar{\mathbf{K}}^2 \mathbf{b} = 1. \quad (68)$$

Nesta abordagem em RKHS, deve-se maximizar o valor da função contraste $J(y[k]) = E[y[k]y[k - \tau]]$, que, seguindo o raciocínio proposto no Capítulo 2, é o mesmo que maximizar $J(\mathbf{b}) = \mathbf{b}^T \bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau \mathbf{b}$. Sendo $\bar{\mathbf{K}}_\tau$ a versão com as linhas transladadas τ vezes como

$$\bar{\mathbf{K}}_\tau(k, i) = \begin{cases} \bar{\mathbf{K}}(k - \tau, i), & \text{se } k > \tau \\ \bar{\mathbf{K}}(n + k - \tau, i), & \text{se } k \leq \tau \end{cases} \quad (69)$$

Na superfície definida por (68), pode-se maximizar $J(\mathbf{b})$ repetindo-se

$$\begin{cases} \mathbf{b} \leftarrow (\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}) \mathbf{b} \\ \mathbf{b} \leftarrow \frac{\mathbf{b}}{\sqrt{\mathbf{b}^T \bar{\mathbf{K}}^2 \mathbf{b}}} \end{cases}, \quad (70)$$

até a convergência.

Assim como analisado no Capítulo 2, o ponto fixo do procedimento (70) é o autovetor correspondente ao maior autovalor de $\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}$. Deste modo, uma formula para calcular o máximo de $J(\mathbf{b}) = \mathbf{b}^T \bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau \mathbf{b}$ com a restrição (68) é

$$\mathbf{b} = \text{eig}(\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}), \quad (71)$$

sendo $\text{eig}(\cdot)$ a função que retorna o autovetor correspondente ao maior autovalor da matriz passada como argumento.

Agora, supõe-se que os sinais observados $\{\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_m[k]]^T, k \in \mathbf{N}\}$ no espaço original dos dados são gerados por uma transformação linear de \mathcal{H}_κ para \mathbb{R}^m . Desde que se lida com RKHS, a transformação não-linear $\Psi: \{\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_m[k]]^T, k \in \mathbf{N}\} \rightarrow \mathcal{H}_\zeta$ pode ser formulada como

$$\mathbf{x}[k] \mapsto \Psi(\mathbf{x}[k]) = \boldsymbol{\zeta}[k], \quad (72)$$

ou seja,

$$\mathbf{x}[k] \mapsto \kappa(\mathbf{x}[k], \cdot) = \boldsymbol{\zeta}[k], \quad (73)$$

De fato, a suposição implica em um RKHS \mathcal{H}_ζ com kernel κ , de forma que se pode calcular a matriz de Gram de $\boldsymbol{\zeta}[k]$ como

$$\bar{\mathbf{K}}(k, i) = \kappa(\mathbf{x}[i], \mathbf{x}[k]), \quad (74)$$

e a estimação $y[k]$ utilizando (62) e (70) ou (71). Lembrando que a metodologia adotada de implicar uma penalidade na norma de $\boldsymbol{\omega}$ e, conseqüentemente, de \mathbf{b} , é uma forma de evitar parte do *problema do escalonamento*[7] inerente aos métodos de BSE.

Concluindo-se, propõe-se o algoritmo no Tabela 4.1 chamado KDCA baseado em (62) e (71). Nota-se, porém, que se derivou o algoritmo assumindo-se que as funções $\zeta_i[k]$ tivessem média zero, porém, nem sempre pode-se garantir isto [21]. Por isto, é muito provável que o primeiro autovetor de $\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}$ esta componente dc, fazendo com que o vetor desejado passe a ser o referente ao segundo maior autovalor [50]. É por isto que calcula-se o segundo autovetor de $\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}$ no algoritmo da Tabela 4.1. Alternativamente, a formula proposta em [21] e comentada em (53) pode ser utilizada para centralizar a matriz de Gram $\bar{\mathbf{K}}$ e definir

$$\bar{\mathbf{K}}_c = \bar{\mathbf{K}} - \mathbf{1}_{n \times n} \cdot \bar{\mathbf{K}}/n - \bar{\mathbf{K}} \cdot \mathbf{1}_{n \times n}/n + \mathbf{1}_{n \times n} \cdot \bar{\mathbf{K}} \cdot \mathbf{1}_{n \times n}/n, \quad (75)$$

Tabela 4.1 – Algoritmo Análise de Componentes Dependentes com Kernel (KDCA)

1 – Entrar	dados branqueados $\mathbf{x}[k] \mid k \in \mathbf{N}$ e atraso τ
2 – Definir	um kernel positivo definido $\kappa(\cdot, \cdot)$
3 – Calcular	$\bar{\mathbf{K}}$ utilizando (74)
4 – Se $\bar{\mathbf{K}}$ é centralizado	\mathbf{b} recebe o primeiro autovetor de $\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}$
5 – Senão	\mathbf{b} recebe o segundo autovetor de $\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}}$
6 - Retornar	$y[k] = \bar{\mathbf{K}}(k, :)\mathbf{b}$,

Esta análise de autovalores torna KDCA relacionado com KPCA, na seção seguinte, esta relação serão detalhadas.

4.2 Comparação com KPCA

Pode-se notar que KDCA é semelhante à extração da primeira componente principal quando o atraso τ é igual a zero. Esta observação segue do fato de que a matriz de Gram é simétrica, logo,

$$\bar{\mathbf{K}}^T \bar{\mathbf{K}}_\tau + \bar{\mathbf{K}}_\tau^T \bar{\mathbf{K}} = \bar{\mathbf{K}}^T \bar{\mathbf{K}} + \bar{\mathbf{K}}^T \bar{\mathbf{K}} = 2\bar{\mathbf{K}}^2, \quad (76)$$

portanto, substituindo (76) e (71) em (65), tem-se

$$y[k] = \bar{\mathbf{K}} \text{eig}(2\bar{\mathbf{K}}^2). \quad (77)$$

Além disso, denotando a decomposição espectral de $\bar{\mathbf{K}}$ como $\bar{\mathbf{K}} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, sendo \mathbf{D} uma matriz diagonal cujos valores são os autovalores de $\bar{\mathbf{K}}$ em ordem decrescente, as colunas de \mathbf{E} são os correspondentes autovetores. Desde que $\bar{\mathbf{K}}^2 = \mathbf{E}\mathbf{D}^2\mathbf{E}^T$, pode-se escrever (77) como

$$y[k] = d_1 \mathbf{e}_1[k], \quad (78)$$

sendo que \mathbf{e}_1 denota a primeira coluna de \mathbf{E} e d_1 é o autovalor correspondente a este autovetor, que ilustra a relação entre KDCA e KPCA.

Na seção seguinte será mostrada uma simulação ilustrativa para comparar DCA e KDCA.

4.3 Simulação ilustrativa

Uma simulação ilustrativa para comparar DCA e KDCA pode ser realizada com um sinal que tenha estrutura temporal, como por exemplo, o sinal de ECG.

Neste exemplo, KDCA foi implementado com kernel gaussiano $\kappa_\sigma(\mathbf{v}, \mathbf{u}) = \exp\left[-\frac{(\mathbf{v}-\mathbf{u})^T(\mathbf{v}-\mathbf{u})}{2\sigma^2}\right]$, apresentado no capítulo anterior. Os sinais orginais $\mathbf{s}[k] = [s_1[k], s_2[k], s_3[k], s_4[k]]$, com $k = 1, \dots, 1000$, assumindo-se que s_1 seja um sinal de ECG amostrado a aproximadamente 0.0059 batidas por amostra, que corresponde a um sinal de 1Hz amostrado a 250Hz. Os outros 3 sinais são ruídos gaussianos.

Este quatro sinais são misturados linearmente e a Raiz Quadrada do Erro-Médio-Quadrático (RMSE, do inglês *Root-Mean-Squared Erro*) entre o sinal original e o extraído para quantizar o desempenho de cada algoritmo. Na Figura 4.1 mostra-se o RMSE para diferentes valores de atraso τ e tamanho do kernel σ para verificar que existe uma quantidade relativamente grande de valores que resultam em uma extração bem sucedida. Nata-se que para $\sigma = 15$ KDCA forneceu um RMSE médio de 0,1987, enquanto que DCA teve uma RMSE médio 0,6771.

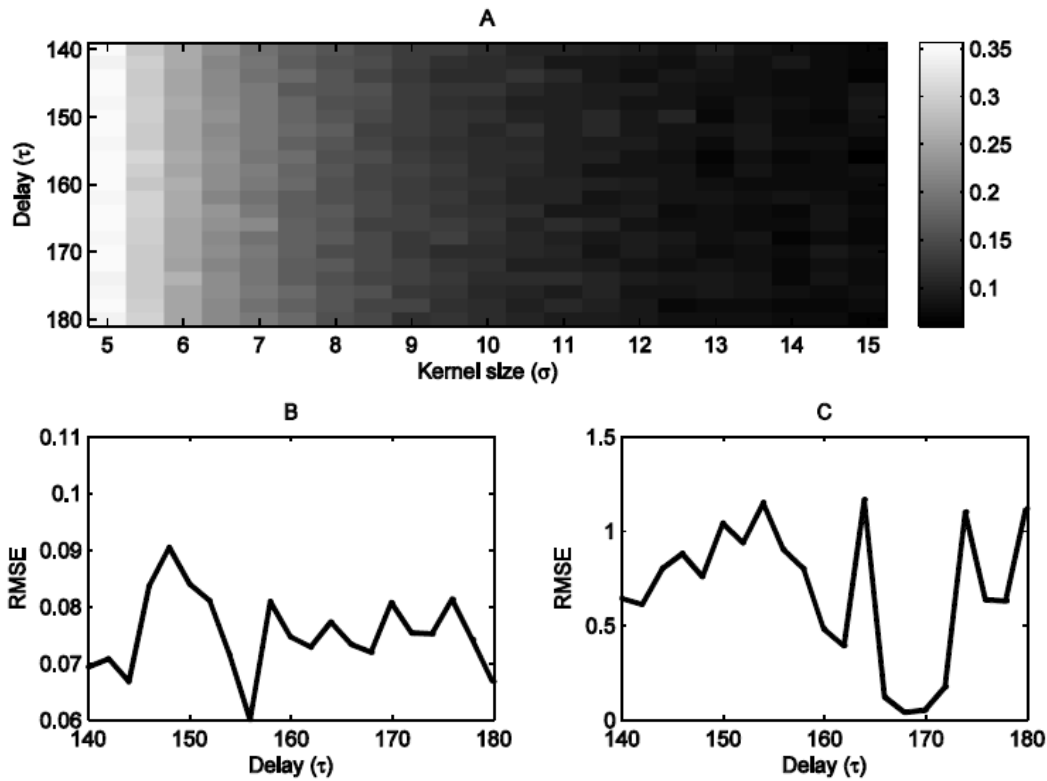


Figura 4.1. RMSE da extração do sinal de ECG utilizando KDCA e DCA. a) RMSE do KDCA para vários atrasos τ , de 140 a 180, e tamanhos do kernel gaussiano σ , de 5 a 15. b) RMSE do KDCA para $\sigma = 15$. c) RMSE do DCA.

Na próxima seção, serão analisadas as transformações equivalentes realizadas no espaço original dos dados enquanto que se realizam operações lineares em RKHS. Esta análise será realizada referentemente a ITL.

4.4 Perspectiva de KDCA segundo ITL

Existem diversas semelhanças entre métodos baseados em kernels e estimação de densidade de Parzen [27]. Por este motivo, procura-se aqui uma interpretação de KDCA a partir do ponto de vista ITL [51].

Dadas duas FDPs $f_1(x)$ e $f_2(x)$, a medida de divergência baseada na inequação de Cauchy-Schwarz entre elas é [44]

$$D_{CS}(f_1, f_2) = -\log \frac{\int f_1(x)f_2(x)dx}{\sqrt{\int f_1^2(x) f_2^2(x)dx}} \quad (79)$$

D_{CS} em (79) é maior ou igual a zero e igualdade é satisfeita se somente se $f_1(x) = f_2(x)$. Devido à normalização imposta em (68) é necessário utilizar o janelamento de Parzen ponderado definido em [27] para estimar as FDPs analisadas como

$$\hat{f}(x_i) = \sum_{j=1}^n a_j \kappa(x_i, x_j) = \bar{\mathbf{K}}(i, :)\mathbf{a}, \quad (80)$$

sendo, a_j escalares positivos tais que satisfaz-se a condição $\int \hat{f}(x)dx = 1$, \mathbf{a} é um vetor cujas entradas são a_j . A estimação do argumento do logaritmo em (79), utilizando (80), é calculado utilizando-se observações independentes \mathbf{x}_1 e \mathbf{x}_2 , realizações das FDPs $f_1(x)$ e $f_2(x)$. A fórmula da estimação de $\frac{\int f_1(x)f_2(x)dx}{\sqrt{\int f_1^2(x) f_2^2(x)dx}}$ é como se segue

$$I_{CS} = \frac{\mathbf{a}_1^T \bar{\mathbf{K}}_1 \bar{\mathbf{K}}_2 \mathbf{a}_2}{\sqrt{\mathbf{L}(\mathbf{1}^T \bar{\mathbf{K}}_1 \mathbf{a}_1)(\mathbf{1}^T \bar{\mathbf{K}}_2 \mathbf{a}_2)}}, \quad (81)$$

sendo que \mathbf{a}_1 e \mathbf{a}_2 são vetores cujas entradas são os pesos da estimação de Parzen das funções $f_1(x)$ e $f_2(x)$, respectivamente, e $\mathbf{L} = \sum_{i,j}^n a_1^i \kappa(\mathbf{x}_1(i), \mathbf{x}_1(j)) \kappa(\mathbf{x}_2(i), \mathbf{x}_2(j)) a_2^j$. Em (81), I_{CS} é chamada de informação quadrática mútua de Cauchy-Schwarz.

Na análise realizada em Jian-Wu Xu et. al. [49] observou-se que o valor no numerador do argumento do logaritmo em (79) caracteriza a dependência entre as duas FDPs. Esta quantidade foi nomeada CIP no capítulo anterior. O CIP quantifica a interação entre os potenciais de informação marginais quando a Entropia Quadrática de Renyi [46] é utilizada como medida de informação [44]. Portanto, pode-se verificar que maximizar $\mathbf{a}_1^T \bar{\mathbf{K}}_1 \bar{\mathbf{K}}_2 \mathbf{a}_2 \approx \int f_1(\mathbf{x})f_2(\mathbf{x})d\mathbf{x}$, mesmo que com outras normalizações, contanto que estas mantenham $\bar{\mathbf{K}}_1$, $\bar{\mathbf{K}}_2$ e as direções do vetores \mathbf{a}_1 e \mathbf{a}_2 fixas, também maximiza a informação quadrática mútua de Cauchy-Schwarz entre as duas FDPs. Relembrando que a função contraste analisada aqui é $J(\mathbf{b}) = \mathbf{b}_1^T \bar{\mathbf{K}}_1 \bar{\mathbf{K}}_2 \mathbf{b}_2$, sob a restrição $\mathbf{b}^T \bar{\mathbf{K}}^2 \mathbf{b} = 1$, ou outra restrição de norma semelhante, pode-se ver que, de fato, realiza-se a maximização da informação quadrática mútua de Cauchy-Schwarz entre as FDPs

$$g(\mathbf{x}[k]) = \sum_{i=1}^n b_i \kappa(\mathbf{x}[k], \mathbf{x}[i]) = \bar{\mathbf{K}}(k, :)\mathbf{b}, \quad (82)$$

$$g_\tau(\mathbf{x}[k]) = \sum_{i=1}^n b_i \kappa(\mathbf{x}[k - \tau], \mathbf{x}[i]) = \bar{\mathbf{K}}_\tau(k, :) \mathbf{b}, \quad (83)$$

Desde que as FDPs caracterizam toda a informação disponível sobre uma variável aleatória, espera-se que o presente método baseado em kernel para extração de sinais forneça melhores resultados que aqueles baseados meramente em correlações.

É interessante notar que a máxima informação quadrática de Cauchy-Schwarz no presente problema é obtida quando $g(\mathbf{x}[k])$ é uniforme, o que torna $g_\tau(\mathbf{x}[k])$ igualmente uniforme e $g(\mathbf{x}[k]) = g_\tau(\mathbf{x}[k])$. O interesse nesta observação deve-se ao fato de que finitas amostras de uma onda triangular, quando tratadas como uma variável aleatória, apresentam distribuição aproximadamente uniforme e satisfazem as condições em (63) para quase todo atraso τ . Esta propriedade será explorada no Capítulo 5 para testar a robustez dos algoritmos para BSE com relação a erros no cálculo de τ .

Interpretar o kernel como uma janela de Parzen, também sugerem alguns métodos para a escolha do tamanho do kernel em função dos dados, uma vez que existem diversas regras para tal escolha no caso de estimação de FDPs [52].

No próximo capítulo, realizar-se-ão experimentos computacionais com dados artificiais e reais para comparar KDCA com métodos lineares para BSE, bem como outros métodos não-lineares.

CAPÍTULO 5

EXPERIMENTOS COMPUTACIONAIS PARA EXTRAÇÃO CEGA DE SINAIS

5.1 Introdução

Neste capítulo serão realizados experimentos computacionais para ilustrar a validade de KDCA. Observa-se que para um atraso $\tau = 0$, KDCA iguala-se a KPCA e, portanto, sua validade já foi analisada em [21]. Para verificar nos casos em que $\tau \neq 0$, primeiramente simula-se a extração de uma onda triangular para diferentes atrasos, conforme a motivação discutida na Seção 4.4 sobre ITL. O segundo experimento é baseado na extração de sinais de ECG Fetal (FECG) de gravações reais do tórax e abdômen de uma mulher grávida. Em seguida, no terceiro experimento, mostra-se que a Decomposição Incompleta de Cholesky (DIC) [30] pode ser utilizada para reduzir o custo computacional de KDCA. Por fim, no quarto experimento, realiza-se a extração de senoides de misturas não-lineares e comparação com o método KTDSEP [22], que é basicamente o algoritmo SOBI [10] *kernelizado*. A quantificação dos resultados fornecidos por cada método (sejam os lineares de DCA Barros [11] e Zhang [14] ou não-lineares como KDCA e KTDSEP [22]) nos experimentos seguintes é dada como o RMSE entre o sinal extraído pelo algoritmo analisado e o sinal desejado original, desta forma, quanto menor o RMSE melhor a qualidade da extração.

5.2 Detalhes de implementação dos experimentos

Os experimentos computacionais foram realizados em um computador *Intel® Core 2 Duo*, 2.53 GHz de CPU com 3 GB de RAM, utilizando *Matlab* em um Windows XP. Com o objetivo de facilitar a reprodução dos resultados que serão apresentados, todos os códigos utilizados nos experimentos seguintes serão disponibilizados no Apêndice desta dissertação.

Na prática, antes de utilizar-se KDCA, deve-se primeiro passar-se pelas seguintes etapas.

1 – Definir o atraso τ baseando-se nas estatísticas do sinal desejado. Primeiramente, deve-se escolher um valor para τ tal que os sinais originais satisfaçam (63). Desde que

normalmente não se possui conhecimento de tal valor na prática, recomenda-se aqui utilizar-se aqueles valores de $\tau \neq 0$ que forneçam os maiores picos na função de autocorrelação dos dados branqueados. Tal escolha tem fornecido bons resultados práticos [11][28]. Nota-se que estes valores podem ser calculados a partir da diagonal principal de $\bar{\mathbf{K}}_\tau$ como, $\frac{1}{n} \sum_{i=1}^n \bar{\mathbf{K}}_\tau(i, i)$, em função de τ . Vale ressaltar que o valor esperado da diagonal principal de $\bar{\mathbf{K}}_\tau$, quando esta é definida utilizando-se o kernel gaussiano, é igual à função de correntropia [50] dos dados, sendo que esta é uma medida generalizada de correlação que também pode ser utilizada para BSE [53].

2 – Escolher os kernel κ que será utilizado pelo algoritmo KDCA para calcular $\bar{\mathbf{K}}$. Neste capítulo, os experimentos serão realizados utilizando-se o kernel gaussiano (41), denotado como κ_σ , que também é uma janela de Parzen válida para estimação de FDPs [25]. Porém, espera-se que os resultados seguintes sejam válidos para quaisquer kernels positivo-definido simétricos de Mercer [43]. Além disso, a escolha do kernel κ deve ser casada com a escolha de τ para preservar as estruturas temporais mesmo no RKHS. Entretanto, a definição da família de todos os kernels válidos para BSE ainda não foi definida e é um problema em aberto [28].

Outro fator que foi verificado experimentalmente, é a melhora dos resultados de extração de misturas não-lineares quando um pré-processamento de branqueamento em RKHS é realizado. Em tais misturas, muitas ambiguidades podem ser adicionadas à estrutura temporal dos dados, e branqueamento resolveu as dificuldades impostas por estas ambiguidades. Para misturas lineares, nenhuma ambiguidade extra é adicionada e pré-processamento em RKHS torna-se desnecessário. O branqueamento em RKHS é realizado utilizando-se KPCA de forma que

$$\bar{\mathbf{K}} \leftarrow \mathbf{D}_d^{-1/2} \mathbf{E}_d^T \bar{\mathbf{K}}, \quad (84)$$

sendo que \mathbf{D}_d é a matriz diagonal cujas entradas são d maiores autovalores de $\bar{\mathbf{K}}$ e as colunas de \mathbf{E}_d são os correspondentes autovetores. A escolha da quantidade d foi realizada de forma heurística nos seguintes experimentos e definida como o menor valor para o qual acréscimo não representa diminuição significativa no RMSE entre o sinal extraído e desejado.

5.3 Experimento 1: Extração de sinais periódicos

Para analisar a robustez dos métodos de BSE, misturou-se linearmente uma onda triangular com período de 125 amostras com dois sinais de ruídos gaussianos e um de ruído com distribuição uniforme. Para realizar a extração, variou-se o valor do atraso no intervalo $0 \leq \tau \leq 50$. As melhores extrações, ou seja, aquelas com menor RMSE são obtidas para $\tau = 125$ ou múltiplos, uma vez que este o valor do período do sinal deseja, a onda triangular.

Um algoritmo robusto com relação a erros na escolha do valor de τ apropriado é aquele que extrai a onda triangular com menor RMSE possível para qualquer valor de atraso τ . Repetiu-se o experimento de extração da onda triangular 50 vezes para cada valor de τ , a cada repetição, amostrou-se aleatoriamente a matriz de mistura \mathbf{A} e os ruídos. Calculou-se a média do RMSE ao longo das 50 repetições. Os RMSE médios são mostrados em função do atraso τ na Figura 5.1.

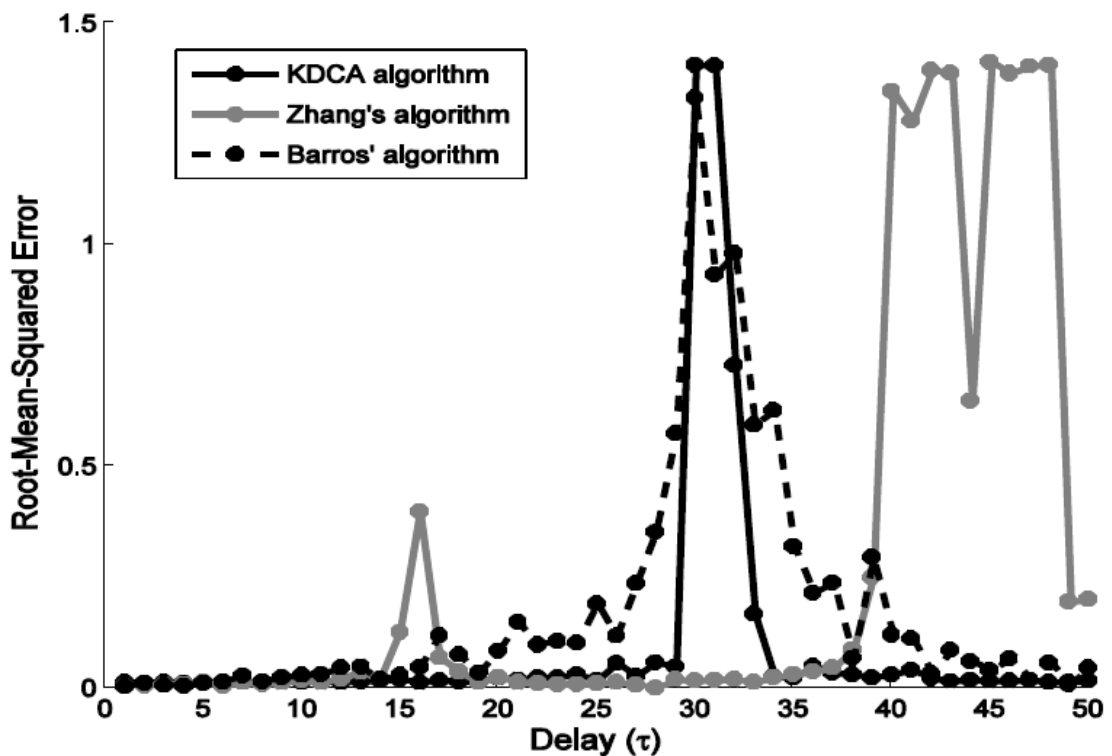


Figura 5.1. RMSE entre onda triangular extraída e original em função do atraso τ .

O algoritmo proposto, ou seja, KDCA apresentou o menor valor de RMSE para quase todos os valores de τ . O RMSE só cresce substancialmente para o valor do atraso em que a fase da onda triangular é descontínua, ou seja, para $\tau = 31$. O algoritmo de Zhang [14]

forneceu os melhores resultados ao redor deste atraso, porém, falhou em extrair o sinal desejado para muitos outros atrasos.

A Tabela 5.1 apresenta-se a média e o desvio padrão do RMSE de cada algoritmo neste experimento. Tal resultado nos permite concluir que KDCA é o mais robusto entre os algoritmos comparados para variações na escolha do atraso τ , que é problema comum em BSE [14]. Pode-se atribuir tal robustez de KDCA às propriedades de processamento de informação do algoritmo discutidas na Seção 4.4 sob a ótica de ITL.

Para verificar a extração de sinais que não sigam uma distribuição uniforme, propõe-se uma modificação na onda triangular que consiste em modulá-la com um sinal aleatório binário (0 ou 1) com diferentes probabilidades de assumir o valor 0. Realizou-se simulações para 0 com probabilidades de 0,25 ou 0,5 ou 0,75. Os resultados para este caso também são apresentados na Tabela 5.1, de onde se pode observar que quanto maior a probabilidade de observarem-se zeros e maior a distância da distribuição do sinal resultante da distribuição uniforme, maior também se torna o RMSE entre sinais extraídos e originais. Além disso, a escolha do atraso τ também se torna mais crucial quanto maior o número de zeros nesta situação, fato este que também pode ser predito pela teoria da Seção 4.4.

Para simulações com sinais com estruturas temporais mais complexas, apresentar-se-ão simulações com sinais de ECG na próxima Seção.

Tabela 5.1. Resultados do Experimento 1.

RMSE	KDCA	Zhang	Barros
RMSE da extração de onda triangular utilizando diferentes atrasos			
Média	0.1032	0.2698	0.1860
Desvio padrão	0.3302	0.4976	0.2935
RMSE da extração de onda triangular multiplicada por sinal binário aleatório com 25% dos valores iguais a zero			
Média	0.1606	0.3601	0.2272
Desvio padrão	0.3075	0.5006	0.2564
RMSE da extração de onda triangular multiplicada por sinal binário aleatório com 50% dos valores iguais a zero			
Média	0.2716	0.3661	0.3364
Desvio padrão	0.3774	0.4358	0.3650
RMSE da extração de onda triangular multiplicada por sinal binário aleatório com 75% dos valores iguais a zero			
Média	0.4111	0.4801	0.5020
Desvio padrão	0.4252	0.4775	0.4364

5.4 Experimento 2: Extração de sinais de FECG reais

Para a análise de sinais de ECG reais, utiliza-se o banco de dados disponibilizado em [54]. Barros e Cichocki [11] e Zhang e Yi [14] validaram seus algoritmos para DCA com este banco de dados. Estes dados consistem em três canais de gravações no tórax e quatro no abdômen de uma mulher grávida. Os sinais são amostrados a uma frequência de 250Hz resultando em 2500 amostras gravadas em 10 segundos.

Objetivando a extração do sinal da atividade cardíaca do feto, Barros e Cichocki propuseram a utilização do valor de atraso de $\tau = 112$ baseando-se nos picos da função de autocorrelação de um dos canais de gravação, aquele com maior influência visível dos sinais de FECG.

Aqui, utiliza-se a estratégia proposta na Seção Detalhes de implementação dos experimentos baseada em calcular a função de correntropia, ou seja, o valor esperado da diagonal $\bar{\mathbf{K}}_{\tau}$ em função de τ . Como mostrado na Figura , correntropia também sugere a

escolha de $\tau = 112$. Os outros picos na Figura sugerem atrasos para separação de outros sinais presentes na misturas, tais como o sinal de ECG materno e o artefato da respiração.

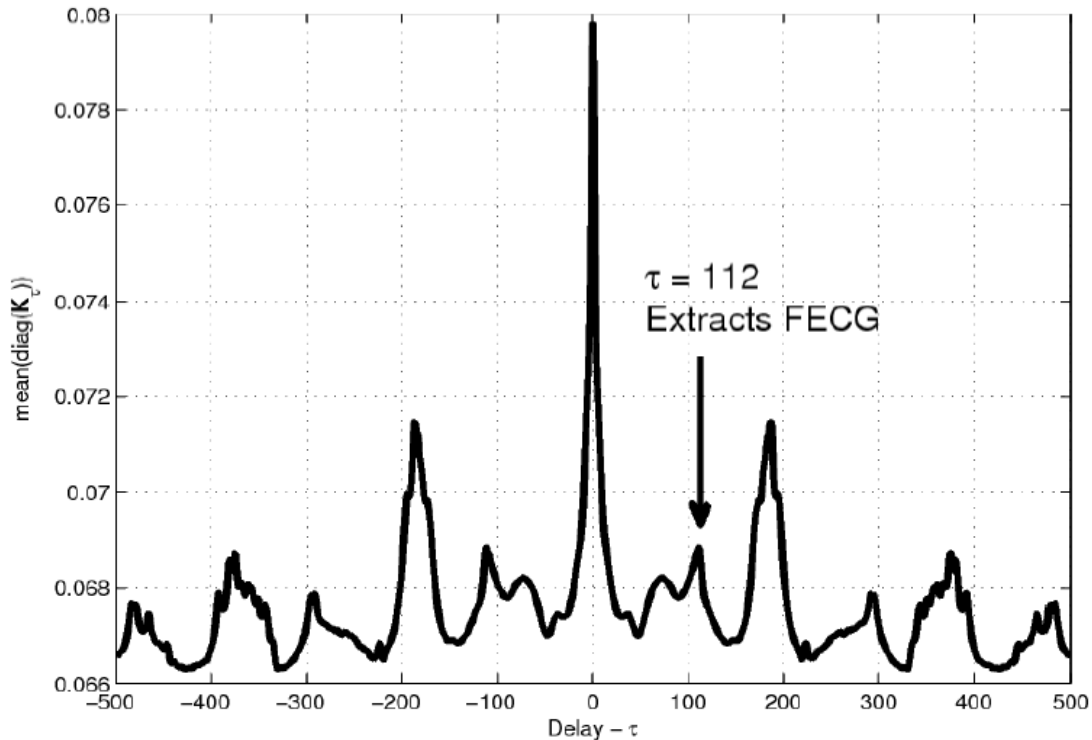


Figura 5.2. Valores esperados da diagonal da matriz $\bar{\mathbf{K}}_\tau$ em função do atraso τ para os sinais de ECG de uma mulher grávida. $\bar{\mathbf{K}}_\tau$ foi calculada com um kernel gaussiano de tamanho $\sigma = 5$.

Este problema de BSE também foi analisado em [14], que mostrou que utilizar atrasos específicos da função de autocorrelação para a extração resulta em melhores extrações que aquelas utilizando atrasos arbitrários, tal trabalhado pelo algoritmo SOBI [10].

Nesta seção, comparam-se os resultados de KDCA com os métodos de Barros [11] e Zhang [14] para extração utilizando atraso $\tau = 112$ e atrasos ligeiramente “incorretos” de $\tau = 110$ e $\tau = 114$. Além disso, devido a possíveis não-linearidades no processo real de mistura dos sinais de ECG, utiliza-se o branqueamento em RKHS (84) como pré-processamento para KDCA com redução de dimensão para $d = 20$.

Na Figura 5.3 mostram-se os sinais extraídos por KDCA e pelos métodos de Barros [11] e Zhang [14]. Observa-se que para $\tau \neq 112$ o método de Barros e Cichocki extraiu basicamente o artefato de respiração. KDCA e o método de Zhang e Yi extraem o sinal de FECG, porém este é corrompido pelo artefato da respiração para $\tau \neq 112$. Porém, vê-se que KDCA foi o algoritmo menos sensível a pequenas modificações no valor do atraso τ .

Os parâmetros utilizados em KDCA neste experimento foram kernel gaussiano com tamanho $\sigma = 5$ e $\bar{\mathbf{K}}$ não centralizado.

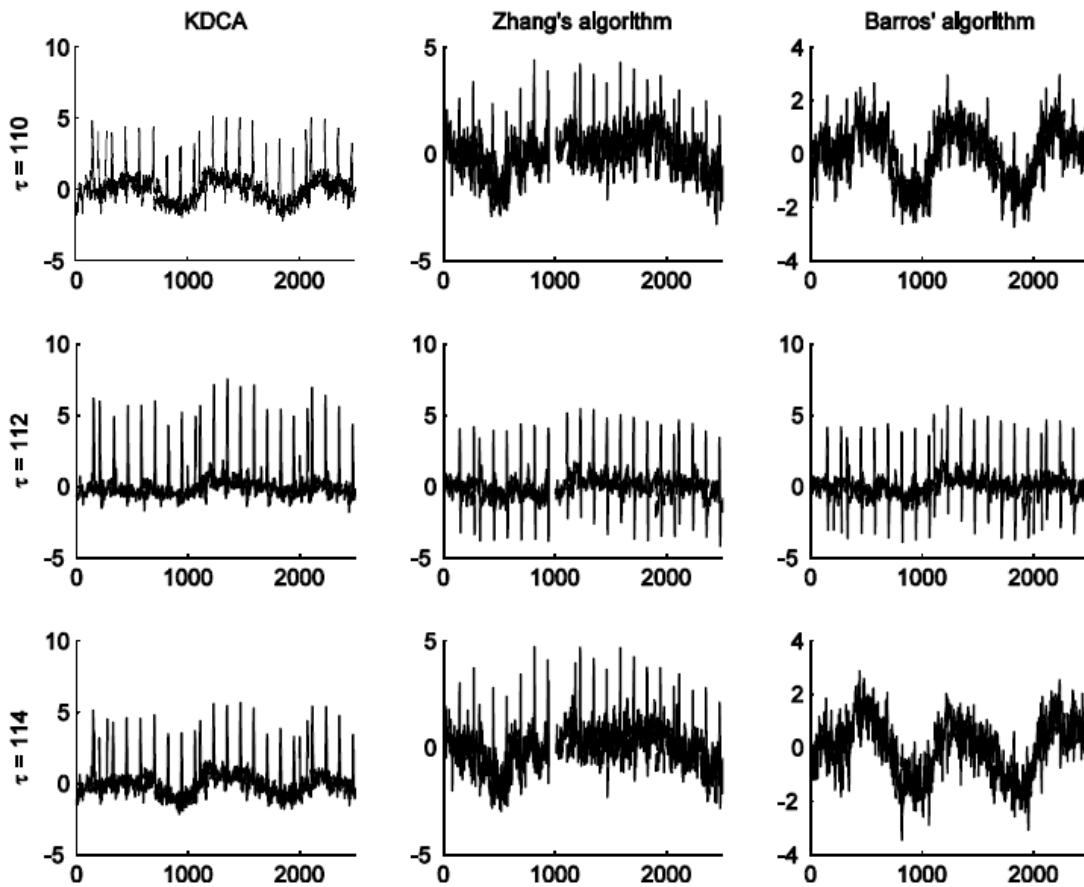


Figura 5.3. Sinais de ECG fetais extraídos das gravações de uma mulher grávida [54] utilizando atraso $\tau = 110, 112, 114$ s e os algoritmos KDCA, e os propostos em [11] (Barros) e [14] (Zhang).

5.5 Experimento 3: Redução de custo computacional utilizando *Decomposição*

Incompleta de Cholesky

Sendo que normalmente tem-se mais amostras temporais que sinais observados, matriz $\mathbf{X} = [\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[n]]_{m \times n}$ tem normalmente mais colunas n que linhas m . Deste modo o calculo da matriz de Gram $n \times n$ exige maior custo computacional que o calculo da matriz de correlação $m \times m$. Assim também, KDCA tem normalmente maior custo computacional que métodos lineares como os apresentados em [11] e [14].

Para reduzir o custo computacional de KDCA, pode-se utilizar a DIC para estimar a matriz de Gram $\bar{\mathbf{K}}$. A DIC aplicada a $\bar{\mathbf{K}}$ consiste em uma estimação de *rank* reduzido da forma $\bar{\mathbf{K}}^* = \mathbf{G}\mathbf{G}^T$, sendo $\mathbf{G} \in \mathbb{R}^{n \times d}$, com $d < n$. Para ilustrar a redução de custo computacional,

repete-se a experiências das seções anteriores comparando-se o calculo direto de $\bar{\mathbf{K}}$ e a estimação $\bar{\mathbf{K}}^*$ por DIC utilizando o algoritmo proposto em [30], com código em Matlab disponibilizado em [55].

O custo computacional do calculo do de autovalores e autovetores necessário para KDCA é de ordem $O(n^3)$ para $\bar{\mathbf{K}}$ e ordem $O(d^3)$ para $\bar{\mathbf{K}}^*$. Para o Experimento 1: Extração de sinais periódicos de extração de onda triangular o algoritmo em [55] definiu $d = 96$ e $d = 374$ para o Experimento 2: Extração de sinais de FECG reais, sendo que originalmente as dimensões eram $n = 1000$ e $n = 2500$, respectivamente. Aqui se utiliza os $\tau = 125$ para a extração da onda triangular e $\tau = 112$ para extração do sinal de ECG. Na Tabela xxx mostra-se a complexidade computacional, o RMSE entre o sinal extraído e original, além do tempo necessário para realizar-se a simulação utilizando o calculo direto e a estimação por DIC da matriz de Gram. Por motivos de comparação, no experimento com FECG considerou-se o sinal original como sendo aquele extraído pelo método de Barros e Cichocki [11] para $\tau = 112$.

Tabela 5.2 Performance do algoritmo KDCA utilizando DIC comparada com calculo direto da matriz de Gram

DIC	Direto	Complexidade computacional		RMSE		Tempo* (s)	
Experimento 1		$O(10^3)$	$O(10^9)$	0,014	0,014	0,225	11,385
Experimento 2		$O(10^7)$	$O(10^9)$	0,556	0,556	54,579	148,12

* Realizou-se esta simulação em um computador com Windows, Intel Core 2 Duo, CPU 2,53GHz e 3 Gb RAM utilizando o Matlab.

Pode-se concluir dos resultados apresentados na Tabela que DIC proporciona economias computacionais consideráveis para KDCA, enquanto que matem aproximadamente a mesma precisão na extração.

Existem outros métodos além da DIC para redução do custo computacional de métodos baseados em kerneis que poderiam ser aplicados em KDCA. Por exemplo, o método de Nyström que é baseado em aproximar a matriz de Gram por combinações lineares apropriadas de colunas aleatoriamente selecionadas desta matriz [56]. Além disso, poderia se

utilizar o método proposto por Smola e Schölkopf baseado em aproximações esparsas da matriz de Gram [57].

5.6 Experimento 4: Mistura não-linear

Este experimento compara o desempenho de KDCA em comparação com KTDSEP [22], sendo que este método foi proposto como uma extensão não-linear do algoritmo SOBI [10], tal como KDCA é proposto como uma extensão não-linear de DCA [11][14], que não podem separar as misturas não lineares aqui apresentadas.

Nesta simulação, realiza-se a mistura de dois sinais senoidais, $s_1[k]$ e $s_2[k]$, de 1000 amostras com frequências 0,05Hz e 0,025Hz, respectivamente. A mistura é definida como

$$\begin{aligned} x_1[k] &= \exp(s_1[k]) - \exp(s_2[k]) \\ x_2[k] &= \exp(-s_1[k]) + \exp(-s_2[k]). \end{aligned} \quad (85)$$

Devido à não-linearidade do processo de mistura em (85), precisa-se branquear os dados em RKHS utilizando (84) com redução de dimensão pra $d = 70$ como pré-processamento para KDCA e KTDSEP [22]. Neste, experimento ambos os métodos utilizam kernels gaussianos com tamanho $\sigma = 0,5$. Nota-se que o tamanho do kernel utilizando neste experimento é menor que aquela utilizada no experimento com sinais de FECG, pois os sinais senoidais tem distribuição mais esparça que os sinais de FECG, sendo que este fenômeno é previsto pela estimação de FDPs de Parzen [26].

Os resultados dos RMSE entre os sinais extraídos por KDCA e KTDSEP e os sinais originais são apresentados na Tabela 5.3.

Tabela 5.3 RMSE para extração de sinais senoidais de mistura não-linear utilizando KDCA e KTDSEP

	Mistura		KDCA		KTDSEP	
	x_1	x_2	y_1	y_2	y_1	y_2
s_1	0,791	0,791	0,022	1,413	0,010	1,413
s_2	0,791	0,791	1,413	0,008	1,413	0,006

Como se pode observar na Tabela 5.3 ambos os métodos obtiveram resultados aproximados na extração dos sinais senoidais, com KTDSEP ligeiramente melhor. Porém, observa-se que KDCA retorna apenas um sinal extraído, o desejado e especificado pelo atraso τ , enquanto que para KTDSEP, depois do processo de separação de sinais, ainda precisa-se procurar pelas fontes desejadas entre os 70 sinais de saída, uma vez KTDSEP é um método de BSS, ou seja, de separação mútua de sinais, o que adiciona mais custo ao processo.

5.7 Discussão dos resultados

Como previamente comentado, KDCA pode ser visto como uma extensão de KPCA para a análise da autocorrelação para diversos atrasos. Por este motivo, dentre os métodos baseados em RKHS, aquele que mais se aproxima de KDCA é o KTDSEP [22]. Porém, enquanto que em KDCA escolhem-se atrasos específicos para diagonalização da matriz de autocorrelação, KTDSEP utiliza todos os atrasos até certo valor, desconsiderando, portanto, a estrutura temporal de sinais específicos ou desejados. Além disso, o problema de separação mútua de fontes em KTDSEP se complica pelo fato de além dos sinais originais, serem outros separados sinais gerados pelo processo de mistura ou que compõem parte da estrutura dos sinais originais e, deste modo, tornam-se necessário técnicas heurísticas para a busca dos sinais originais dentre os separados [22]. KDCA, por outro lado, não sofre deste problema se as fontes satisfizerem as condições (63).

Uma desvantagem que o algoritmo aqui apresentado sofre com relação a métodos lineares é o acréscimo no custo computacional, sendo que tal acréscimo é também presente em outros métodos baseados em kernels [20][19]. Além disso, em BSE o número de canais de observação m é normalmente menor que o número de amostras n , o que torna a matriz de autocorrelação menor que a matriz de Gram. Uma vez que KDCA é baseado na decomposição em autovalores e autovetores da matriz de Gram, irá demandar uma maior quantidade de recursos computacionais que um método baseado na decomposição da matriz de autocorrelação, como é o caso do algoritmo de Zhang e Yi [14]. Por exemplo, no Experimento 1: Extração de sinais periódicos onde se trabalhou com sinais armazenados em uma matriz de dimensões 4×1000 (quatro sinais originais, com mil amostras cada), ordem computacional cresce de $O(10)$ para decomposição necessária para o método de Zhang e Yi [14], para a ordem de $O(10^9)$ na decomposição necessária para KDCA. Para estimações de *rank* reduzido da matriz de Gram e redução de custo computacional, utilizou-se aqui a DIC,

que reduziu o custo computacional para valores entre $O(10^3)$ e $O(10^6)$ dependendo da complexidade dos dados envolvidos na mistura.

Comparando-se KDCA com métodos não-lineares para BSS [4][22], podem-se esperar piores resultados com KDCA, uma vez que métodos de BSS utilizam mais estatísticas na separação, ao custo do acréscimo de ambiguidades quanto aos sinais desejados entre os separados. Enquanto que KDCA evita os excessos computacionais e de memória necessários para separar todos os sinais envolvidos na mistura, quando se desejam apenas alguns sinais específicos [58].

CAPÍTULO 6

CONCLUSÕES

Apresentou-se nesta dissertação a Análise de Componentes Dependentes utilizando Kerneis, chamada KDCA, que é uma técnica para extração cega de fontes utilizando espaços de Hilbert reproduzidos por kernel. Aqui, reformulou-se a Análise de Componentes Dependentes, originalmente proposta por Barros e Cichociki em 2001, e estendeu-se o a análise para um espaço de Hilbert de várias (possivelmente infinitas) dimensões, sendo que um kernel foi utilizado para calcular produtos internos entre funções deste espaço sem que fosse necessário definir explicitamente a transformação que leva para tal espaço.

Desta forma, KDCA foi derivado matematicamente, definiu-se como realizar extração de fontes específicas utilizando operações algébricas em um espaço de Hilbert. Estudou-se também a relação destas operações algébricas com a maximização da informação mútua de Cauchy-Schwarz, segundo a teoria da Aprendizagem de Máquina utilizando Teoria da Informação.

Para validar experimentalmente a teoria apresentada, realizaram-se simulações computacionais que comprovam que o presente método é mais robusto que os métodos lineares, com relação a possíveis enganos no conhecimento a priori da estrutura temporal dos sinais desejados.

O lado negativo desta nova abordagem foi o acréscimo no custo computacional com relação aos métodos lineares. Sendo que se mostrou como amenizar este problema em parte utilizando estimações de *rank* reduzido da matriz de Gram.

Possíveis linhas futuras de pesquisa incluem o desenvolvimento de uma abordagem unificada para extração cega de fontes utilizando kerneis, bem como foi realizada para métodos lineares [1][59]. Outras pesquisas podem incluir a análise do significado físico da função densidade de probabilidade “translada” que se observou em (83) e seu valor esperado com relação à função densidade de probabilidade original, ou seja, o produto interno entre estas duas densidades que é parte de KDCA. Já o estudo de KTDSEP segunda a ótica da Aprendizagem de Máquina utilizando Teoria da Informação envolveria o estudo da convolução de uma função densidade de probabilidade com ela mesma. Por fim, ainda é problema em aberto a definição de todos os kerneis que reproduzem espaços de Hilbert que

preservam a estrutura temporal dos dados originais, bem como definir que tipos de misturas não-lineares podem ser separadas utilizando KDCA.

REFERÊNCIAS

- [1] Eder Santana, "Extração Cega de Fontes Utilizando Correntropia," Universidade Federal do Maranhão, São Luís, Monografia 2010.
- [2] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed.: Prentice-Hall Signal Processing Series, 1999.
- [3] Andrzej Cichocki and Shun I. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York, NY, USA: John Wiley & Sons, Inc., 2002. [Online]. <http://portal.acm.org/citation.cfm?id=863120>
- [4] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 1st ed.: Wiley-Interscience, May 2001.
- [5] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed.: Prentice Hall, 1998.
- [6] Te-Won Lee, *Independent Component Analysis: Theory and Applications*, Te-Won Lee, Ed.: Kluwer Academic Publishers, Setembro, 1998.
- [7] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287-314, April 1994.
- [8] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1-10, July 1991.
- [9] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483-1492, 1997.
- [10] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434-444, August 2002.
- [11] Allan Kardec Barros and A. Cichocki, "Extraction of Specific Signal with Temporal Structure," *Neural Computation*, vol. 13, no. 1, September 2001.
- [12] C. A. Estombelo-Montesco et al., "Dependent Component Analysis for the Magnetogastrographic Detection of Human Electrical Response Activity," *Physiological Measurement*, vol. 28, pp. 1029-1044, 2007.

- [13] Zhenwei Shi and Changshui Zhang, "Blind Source Extraction Using Generalized Autocorrelations," *IEEE Trans. on Neural Networks*, vol. 18, no. 5, pp. 1516-1524, September 2007.
- [14] Z.-L. Zhang and Z. Yi, "Robust extraction of specific signals with temporal structure," *Neurocomputing*, vol. 69, no. 7-9, pp. 888-893, March 2006.
- [15] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: John Wiley, 1996.
- [16] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biology*, vol. 15, pp. 267-273, 1982.
- [17] T. Hastie and W. Stuetzle, "Principal Curves," *JASA*, vol. 84, pp. 502-516, 1989.
- [18] N. Aronszjn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, 1950.
- [19] Bernhard Schölkopf and Alexander J. Smola, *Learning with kernels : support vector machines, regularization, optimization, and beyond*, #dec# 2002.
- [20] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, September 1995.
- [21] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.
- [22] Stefan Harmeling, Andreas Ziehe, Motoaki Kawanabe, and Klaus-Robert Müller, "Kernel-based nonlinear blind source separation," *Neural Comput.*, vol. 15, pp. 1089-1124, May 2003. [Online]. <http://dl.acm.org/citation.cfm?id=795506.795512>
- [23] Weifeng Liu, P. Pokharel, and J. C. Principe, "The Kernel Least-Mean-Square Algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, 2008.
- [24] Wenan Chen and Hongbin Zhang, "The Condition of Kernelizing an Algorithm and an Equivalence Between Kernel Methods," , Berlin, Heidelberg, 2007, pp. 338-345. [Online]. http://dx.doi.org/10.1007/978-3-540-72847-4_44
- [25] Jose C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, 1st ed.: Springer Publishing Company, Incorporated, 2010.
- [26] E. Parzen, "On the estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065-1076, 1962.

- [27] Robert Jenssen, E. Torbjorn, E. Deniz, and J. C. Principe, "Some equivalences between kernel methods and information theoretic methods," vol. 45, no. 1-2, pp. 49-65, November 2006.
- [28] Eder Santana, Jose C. Principe, Ewaldo Santana, R. C. S. Freire, and Allan Kardec Barros, "Extraction of Signals with Specific Temporal Structure using Kernel Methods," *IEEE Transactions on Signal Processing*, 2010.
- [29] S. A. Cruces-Alvarez, A. Cichocki, and S.-I. Amari, "From Blind Signal Extraction to Blind Instantaneous Signal Separation: Criteria, Algorithms and Stability," *IEEE Trans. on Neural Networks*, vol. 15, no. 4, pp. 859-873, July 2004.
- [30] Sohan Seth and Jose C. Principe, "On speeding up computation in information theoretic learning," in *International Joint Conference of Neural Networks IJCNN*, 2009.
- [31] Allan Kardec Barros and Noboru Ohnishi, "Single channel speech enhancement by efficient coding," *Signal Process.*, vol. 85, pp. 1805-1812, September 2005. [Online]. <http://dl.acm.org/citation.cfm?id=1103527.1103536>
- [32] M. E. Davies and C. J. James, "Source separation using single channel ICA," *Signal Process.*, vol. 87, pp. 1819-1832, August 2007. [Online]. <http://dl.acm.org/citation.cfm?id=1240335.1240508>
- [33] A.M. Tome, A.R. Teixeira, E.W. Lang, and A.M. da Silva, "Greedy Kernel PCA Applied to Single-Channel EEG Recordings," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Lyon, 2007, pp. 5441 - 5444.
- [34] Nathalie Delfosse and Philippe Loubaton, "Adaptive blind separation of independent sources: a deflation approach," *Signal Process.*, vol. 45, pp. 59-83, July 1995. [Online]. <http://dl.acm.org/citation.cfm?id=209244.209256>
- [35] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129-1159, November 1995.
- [36] J. F. Cardoso and A. Soloumiac, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157-192, January 1999.
- [37] D. Erdogmus, K. E. Hild II, and J. C. Principe, "Blind source separation using Renyi's α -marginal entropies," *Neurocomputing*, vol. 49, no. 1-4, pp. 25-38, December

- 2002.
- [38] L. Tong, V. C. Soon, Y. F. Huang, and R. Liu, "AMUSE: a new blind identification algorithm," in *IEEE International Symposium on Circuits and Systems*, vol. 3, 1990, pp. 1784-1787.
- [39] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 24, 1948.
- [40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed.: The Johns Hopkins University Press, 1996.
- [41] L. Molgedey and G. H. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review*, vol. 72, no. 23, pp. 3634-3637, 1994.
- [42] N. Akhiezer and I. Glaszman, *Theory of Linear Operators in Hilbert Space (em inglês)*. Nova York, EUA: Ungar, 1963.
- [43] N. Aronszajn, "Theory of reproducing kernels," *Trans. Am. Math. Soc.*, vol. 68, pp. 337-404, 1950.
- [44] J. C. Principe, Xu Dongxin, and J. W. Fisher III, "Information Theoretic Learning, in Unsupervised Adaptive Filtering," in *Unsupervised Adaptive Filtering, Volume 1: Blind Source Separation*, Simon Haykin, Ed.: Wiley-Interscience, 2000, vol. 1.
- [45] Jose C. Principe, Neil R. Euliano, and W. Curt Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations with CD-ROM*, 1st ed. New York, USA: John Wiley & Sons, Inc., 1999.
- [46] Alfred Renyi, "On Measures of Entropy and Information," in *Selected Papers of Alfred Renyi*. Budapest: Akademia Kiado, 1976, vol. 2, pp. 565-580.
- [47] D. Erdogmus and J. C. Principe, "From linear adaptive filtering to nonlinear information processing," *IEEE Signal Processing Magazine*, vol. 23, no. 6, pp. 14-33, November 2006.
- [48] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 99th ed.: Wiley-Interscience, 1991.
- [49] Jian-Wu Xu, A.R.C. Paiva, Il Park, and Jose C. Principe, "A Reproducing Kernel Hilbert Space Framework for Information-Theoretic Learning," *IEEE Transaction on Signal Processing*, vol. 56, no. 12, pp. 5891-5902, Dec 2008.
- [50] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in

- non-Gaussian signal processing," *IEEE Trans. on Signal Processing*, vol. 55, no. 11, pp. 5286-5298, November 2007.
- [51] Jose C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, 1st ed.: Springer Publishing Company, Incorporated, 2010.
- [52] B. W. Silverman, *Density Estimation for Statistics and Data Analysis (em inglês)*. Nova York, EUA: Chapman and Hall, 1986.
- [53] Eder Santana, Jose C. Principe, Ewaldo Santana, and Allan Kardec Barros, "Extraction of signals with higher order temporal structure using Correntropy," *Signal Processing (aceito para publicação)*.
- [54] D. De Moor. Daisy: Database for identification of systems. [Online]. <http://www.esat.kuleuven.ac.be/sista/daisy>
- [55] Sohan Seth. (2009, Março) [Online]. <http://sites.google.com/site/sohanseth/files-/itl.zip?attredirects=0>
- [56] C. K. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *NIPS, T. K. Leen, T. G. Dietterich, V. Tresp, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, Vancouver, Canada, 2000*, pp. 682-688.
- [57] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. 17th International Conf. on Machine Learning*, San Francisco, CA, 2000, pp. 911-918.
- [58] Wai Yie Leong, Wei Liu, and D. Mandic, "Blind source extraction: Standard approaches and extensions to noisy and post-nonlinear mixing," *Neurocomputing*, vol. 71, no. 10-12, Junho 2008.
- [59] Aapo Hyvärinen, "Complexity pursuit: separating interesting components from time-series.," *Neural Computation*, vol. 13, no. 4, April 2001.
- [60] Eder Santana. (2012, Janeiro) KDCA_MatlabPackage. [Online]. http://www.4shared.com/rar/6PNcZAWK/KDCA_MatlabPackage.htm

APÊNDICE

APÊNDICE A – Códigos em Matlab para reprodução de resultados

Nesta seção são apresentados os códigos em Matlab desenvolvidos durante o mestrado e que são utilizados para reproduzir os resultados apresentados nesta dissertação. O pacote com todos os códigos apresentados a seguir e aqueles desenvolvidos por outros autores, mas que são necessários para as simulações, são disponibilizados online em [60]. Os códigos e suas utilidades são as seguintes:

Tabela A.1 – Códigos de Matlab

Função	Utilidade
cholkdca.m	KDCA utilizando DIC
dca.m	Método de Barros e Cichociki, DCA
eigBSE.m	Método de Zhang e Yi, Robust DCA
experiment_1.m	Reproduz experimento da Seção <i>Experimento 1: Extração de sinais periódicos</i>
experiment_2.m	Reproduz experimento da Seção <i>Experimento 2: Extração de sinais de FECCG reais</i>
experiment_3.m	Reproduz experimento da Seção <i>Experimento 3: Redução de custo computacional utilizando Decomposição Incompleta de Cholesky</i>
experiment_4.m	Reproduz experimento da Seção <i>Experimento 4: Mistura não-linear</i>
gera_data.m	Carrega os dados necessários para as simulações
kcentering.m	Calcula a matriz de Gram do dados centralizados em RKHS
kdca.m	Método aqui proposto KDCA
kprod.m	Calcula a matriz de Gram dos dados de entrada
plotee.m	Plota sinais
redux_dimm.m	Reduz a dimensão dos dados utilizando KPCA
rmse.m	Calcula RMSE entre sinais

toy_example.m

Reproduz o experimento da Seção *Simulação
ilustrativa***cholkdca.m**

```

function [y,I,D] =
cholkdca(x,dl,sig)
% This is the kernel dca
algorithm using Cholesky
Decomposition.
% Input
% x          : observed
signal
% dl        : delay of
the specific signal
% sig       : variace of
the kernel
% op.adaptation :
if(op.adaptation = 1) (fixed
point adaptation of w
%           eq. 14),
else (eig adaptation of w using
eq. 23)
% op.center   :
if(op.center = 1) centerate
kernel matrix
%
% Output
% y          : extracted
signal
%
% By         : Eder
Santana, PIB
% Date       : 05-08-2009
% Reference  : Eder
Santana et. al, Extraction of
Specific Signal
%           with
Temporal Strucuture Using Kernel
Methods

[e,d] = eig(x*x'/size(x,2));
X      = d^(-.5)*e'*x; % whiten x
K =
incompleteCholeskyMulti(X',sig);
% call Sohan Seth's routine for
Cholesky Decomposition
KK     = [K(1+dl:end,:);
K(1:dl,:)]; % calculate delayed
kernel
K2     = K'*KK; % calculate
correlation between matrices
K2     = K2+K2';
[E,D] = eig(K2); % calculate
eigvectors of the new kernel
[t,I] =
sort(abs(diag(D)), 'descend');
E      = E(:,I);
y      = K*E(:,2);
y      = zscore(y)';

```

dca.m

```

function [Y,W] = dca(x,L,op)

% [y,W]=DCA(X,L) finds the components of a mixture
% using Barros and Cichocki's routine.
% X is the input matrix.
% L is the length autocorrelation lag.

% By Allan Kardec Barros - RIKEN/UFMA.
% Date:      21-Jan-2000

[nn,M]=size(x);
if M<nn, x=x'; [nn,M]=size(x);end
x=x-mean(x')'*ones([1,M]);      % remove the mean

%-----%

if exist('op')==1,fprintf('Calculating PCA... | ');end

[E,D]=eig(cov(x'));
nr=rank(D);
if nr<nn,
    fprintf('DIMENSION REDUCED FROM %d TO %d \n',nn,nr);
    [I,J]=min(diag(D));
    T=1:nn; T(J)=0; T=find(T>0);
    D=D(T,T); E=E(:,T);
end
V=inv(sqrt(D))*E';
nn=length(D);

%-----%

v=V*x;
if exist('op')==1,fprintf('DCA - Calculating Components...\n');end
w=randn(nn,1); w(1)=1;w=w./norm(w);
limit=200;t=limit;
y=v'*w;
Rxxp=v(:,1:M-L)*v(:,L+1:M)';
for i = 1:100;
    %%% ALGORITHM %%%
    w=Rxxp*w; w=w./norm(w);
    y=v'*w;
end
Y=y';
W=w;
W=W'*V;

```

experiment_1.m

```

function experiment_1
%This routine tests the robustness of kdca to error in the
%calculation of the delay. It also reproduces Fig. 2 in [1]
%
% By      : Eder Santana, PIB
% Date    : 05-07-2009

```

```

% Reference          : [1] Eder Santana et. al, Extraction of Specific
Signal
%                   with Temporal Structure Using Kernel Methods

%% Running parameters
n  = 1e3; % samples number
sig = 10; % define the kernel size
N  = 50; % number of different delays
M  = 50; % number of random repetitions of the experiment
op.center = 0; % do not centerate de kernel
op.adaptation = 0; % allow eigencomputaion
op.eigenumber = 2; % extract the signal at second eigenvector [1]
r = zeros(3,N);
r1 = zeros(N,M);
r2 = r1; r3 = r1;

%% Extraction procedure
h = waitbar(0, 'KDCA : Running Experiment 1');
for i = 1:N
    %i
    dl = 0 + i;
    s  = gera_data(4,n); % load the senoide and the noises
    A  = randn(4); % generate mixing matrix
    x  = A*s; % mix the sources
    for j = 1:M
        j
        %y(1,:) = kdca(x,dl,sig,op);
        y(1,:) = cholkdca(x,dl,sig); %using incomplet Cholesky
        decomposition %for faster computations. You can take
                        %the 'chol' off and use kdca algorithm
                        %'y = kdca(x,dl,sig,op)', but, it will
                        %take a really longer time.

        if (y(1,:)*s(1,:))<0 % correct the signal
            y(1,:) = -y(1,:);
        end
        z = whiten(x);
        y(2,:) = eigBSE(z,dl*[1:3],1,0);
        y(2,:) = zscore(y(2,:))';
        if (y(2,:)*s(1,:))<0 % correct the signal
            y(2,:) = -y(2,:);
        end
        y(3,:) = dca(x,dl);
        y(3,:) = zscore(y(3,:))';
        if (y(3,:)*s(1,:))<0 % correct the signal
            y(3,:) = -y(3,:);
        end

        %         r1(i,j) = rmse(y(1,:),s(1,:));
        %         r2(i,j) = rmse(y(2,:),s(1,:));
        %         r3(i,j) = rmse(y(3,:),s(1,:));

        r(1,i) = (r(1,i)*(j-1)+rmse(y(1,:),s(1,:)))/j;
        r(2,i) = (r(2,i)*(j-1)+rmse(y(2,:),s(1,:)))/j;
        r(3,i) = (r(3,i)*(j-1)+rmse(y(3,:),s(1,:)))/j;
        %         r(1,i) = (r(1,i)*(j-1)+corr(y(1,:) ',s(1,:)'))/j;
        %         r(2,i) = (r(2,i)*(j-1)+corr(y(2,:) ',s(1,:)'))/j;
        %         r(3,i) = (r(3,i)*(j-1)+corr(y(3,:) ',s(1,:)'))/j;
    end
end

```

Apêndice

```

        waitbar(i/N,h)
    end
    close(h)

    %% Plot RMSE in function of time delay
    %plot(r','.-')
    %save rms.mat r
    % plot rmse
    plot(1:N,r(1,:), 'k*-','LineWidth',2), hold on % kdca at black
    plot(1:N,r(2,:), '*-','LineWidth',2,'Color',[.5 .5 .5]) % Zhang's at gray
    plot(1:N,r(3,:), '--*k','LineWidth',2) % Barros' at uncontinuous
    set(gca, 'Box', 'off' ); % here gca means get current axis
    legend('KDCA algorithm','Zhang's algorithm','Barros' algorithm')
    fh = figure(gcf); % returns the handle to the figure object
    set(fh, 'color', 'white'); % sets the color to white
    ylabel('Root-Mean-Squared Error','FontSize',12)
    xlabel('Delay (\tau)','FontSize',12)
    saveas(fh, 'fig2', 'jpeg'); % save figure

```

experiment_2.m

```

function experiment_2
%This routine tests kdca for extracting foetal ECG signal.
%It also reproduces Fig. 4 in [1]
%
% By           : Eder Santana, PIB
% Date        : 05-08-2009
% Reference   : [1] Eder Santana et. al, Extraction of Specific
Signal
%             with Temporal Structure Using Kernel Methods
%             [2] Zhi-Lin Zhang and Zhang Yi, Robust extraction of
%             specific signals with temporal structure

%% Running parameters
n = 2.5e3; % samples number
sig = 10; % define the kernel size
dl = 157; % delay of the autocorrelation
sig = 10; % kernel size
op.center = 0; % do not centerate de kernel
op.adaptation = 0; % allow eigencomputation
op.eigenumber = 2; % extract the signal at third eigenvector [1]

%% Extraction procedure
x = gera_data(3,n);
fecg = eigBSE(whiten(x),112*[1:3],1,0); % extract the signal if Fig. 3.c at
[2]

y(1,:) = kdca(x,dl,sig,op); % extract ecg using kdca
if (y(1,:)*fecg(1,:)')<0 % correct the signal
    y(1,:) = -y(1,:);
end
y(2,:) = eigBSE(whiten(x),dl*[1:3],1,0); % extract ecg using Zhang's
algorithm
if (y(2,:)*fecg(1,:)')<0 % correct the signal
    y(2,:) = -y(2,:);
end
y(3,:) = dca(x,dl); % extract ecg using Barros' algorithm
if (y(3,:)*fecg(1,:)')<0 % correct the signal
    y(3,:) = -y(3,:);
end
end

```

```

%% Calculate and plot extraction error and extracted signals
subplot(4,1,1)
plot(fecg,'k')
legend('Desired Foetal ECG')
title('Foetal ECG')
%title('Foetal ECG signals and extraction RMSE','FontSize',14)
titles = {'KDCA algorithm';'Zhang's algorithm';'Barros' algorithm'};
set(gca, 'Box', 'off' ); % here gca means get current axis
for i = 1:3
    subplot(4,1,i+1)
    r(i) = rmse(y(i,:),fecg);
    plot(y(i,:), 'k')
    legend(['RMSE = ' num2str(r(i))]);
    title(titles(i))
    set(gca, 'Box', 'off' ); % here gca means get current axis
end
fh = figure(gcf);
set(fh,'color', 'white')
saveas(fh, 'fig4', 'jpeg'); % save figure

```

experiment_3.m

```

function experiment_3
%This routine tests kdca with Cholesky Decomposition.
%
% By : Eder Santana, PIB
% Date : 05-08-2009
% Reference : [1] Eder Santana et. al, Extraction of Specific
Signal
% with Temporal Structure Using Kernel Methods
% [2] Sohan Seth, Jose C. Principe, On speeding up
% computation in information theoretic learning

%% Running parameters
M = 5; % number of independent runs
sig = 10; % kernel size
op.center = 0; % do not centerate de kernel
op.adaptation = 0; % allow eigencomputation
op.eigennumber = 2; % extract the signal at third eigenvector [1]

%% Comparation for senoide extraction
dl = 13;% delay of the autocorrelation
for i = 1:M
    s = gera_data(4,1e3); % load senoide and noise
    A = randn(4); % generate mixxing matrix
    x = A*s; % mix the sources
    tic
    y1 = kdca(x,dl,sig,op);
    elapsed1(1,i) = toc;
    tic
    y2 = cholkdca(x,dl,sig);
    elapsed1(2,i) = toc;
end

%% Comparation for ECG extraction
dl = 110;% delay of the autocorrelation
for i = 1:M
    s = gera_data(2,1e3); % load senoide and noise
    A = randn(4); % generate mixxing matrix

```

```

    x = A*s; % mix the sources
    tic
    y1 = kdca(x,d1,sig,op);
    elapsed2(1,i) = toc;
    tic
    y2 = cholkdca(x,d1,sig);
    elapsed2(2,i) = toc;
end
%% Comparison for foetal ECG extraction
d1 = 110;% delay of the autocorrelation
h = waitbar(0,'KDCA : Running Experiment 4');
for i = 1:M
    x = gera_data(2,2.5e3); % load senoide and noise
    tic
    y1 = kdca(x,d1,sig,op);
    elapsed3(1,i) = toc;
    tic
    y2 = cholkdca(x,d1,sig);
    elapsed3(2,i) = toc;
    waitbar(i/M,h)
end
close(h)

%% Plot elapsed times
elapsed_mean_kdca      = [mean(elapsed1(1,:)) mean(elapsed2(1,:))...
                        mean(elapsed3(1,:))]
elapsed_mean_cholkdca = [mean(elapsed1(2,:)) mean(elapsed2(2,:))...
                        mean(elapsed3(2,:))];
elapsed_std_kdca      = [std(elapsed1(1,:)) std(elapsed2(1,:))...
                        std(elapsed3(1,:))]
elapsed_std_cholkdca = [std(elapsed1(2,:)) std(elapsed2(2,:))...
                        std(elapsed3(2,:))];

errorbar(1:3,elapsed_mean_kdca,elapsed_std_kdca,'o-',...
         'color',[0 0 0],'LineWidth',2) % plot kdca time
hold on
errorbar(1:3,elapsed_mean_cholkdca,elapsed_std_cholkdca,'o-',...
         'color',[.5 .5 .5],'LineWidth',2) % plot cholkdca time
axis([0 4 -5 160])
ylabel('Elapsed time (s)','FontSize',12);
xlabel('Experiment number','FontSize',12)
set(gca,'XTick',[1:3])
set(gcf,'color','white')
set(gca,'Box','off'); % here gca means get current axis
legend('KDCA','CholKDCA')
saveas(gcf,'extrafig1','jpeg'); % save figure

```

experiment_4.m

```

function experiment_1
%This routine tests kdca under nonlinear mixtures
%
% By           : Eder Santana, PIB
% Date        : 13-11-2009
% Reference   : [1] Eder Santana et. al, Extraction of Specific
Signal
%
%              with Temporal Structure Using Kernel Methods

t = linspace(1,100,1e3);
%s(1,:) = sin(t);

```

```

s(2,:) = sawtooth(.9*t);

T = 1e3;

s = [sin(linspace(0,100*pi,T)); (sin(linspace(0,42*pi,T)))]';

x(1,:) = exp(s(1,:)) - exp(s(2,:));
x(2,:) = exp(-s(1,:)) + exp(-s(2,:));

op.center = 1;
op.eigenumber = 1;
op.adaptation = 1;

y(1,:) = edca(x,20,.5,0); %run kdca with reduction of the dimension of the
feature space
y(2,:) = edca(x,48,.5,0); %run kdca with reduction of the dimension of the
feature space

root_mean_squared_erros = [prd(s(1,:),y(1,:)) prd(s(2,:),y(2,:))]

figure, plot(y(1,:),y(2,:),'.')

```

gera_data.m

```

function s = gera_data(op,m)
% Generate data for Blind Source Separation
% op = 1 -> ecg, senoides e onda triangular
% op = 2 -> sinais aleatórios
% op = 3 -> fetal ecg
% op = 4 -> senoide e ruído
% op = 5 -> ruído modulado
%
% By : Eder Santana, PIB
% Reference : [1] D. De Moor, "Daisy: Database for Identification of
Systems".
% Available online at: http://www.esat.kuleuven.ac.be/sista/daisy

switch(op)
case 1
dir = cd;
load ecgdata; % Load ECG data
cd(dir)
s(1,:) = svdb800(1,1:m);
s(2,:) = randn(1,m);
s(3,:) = sawtooth([1:m] *2*pi*.003);
s(4,:) = sin([1:m] * 2*pi*.006);
case 2
dir = cd;
load ecgdata; % Load ECG data
cd(dir)
s(1,:) = svdb800(1,1:m);
s(2:3,:) = randn(2,m);
s(4,:) = rand(1,m);
case 3
dir = cd;
load foETAL_ECG.dat; % Load Daisy's fECG data [1]
cd(dir)
s = foETAL_ECG(1:m,2:end)';

```



```

case 4
    s(1,:) = modulate(sin(.005*[1:m]),.1,1);
    s(1,:) = sin(.5*[1:m]);
    s(2:3,:) = randn(2,m);
    s(4,:) = rand(1,m);
case 5
    s(1,:) = sin(.02*[1:m]) .* randn(1,m);
    %s(2:3,:) = randn(2,m);
    s(2:4,:) = rand(3,m);
    s = zscore(s')';
otherwise
    error('Option not available')
end
s = zscore(s')';

```

kcentering.m

```

function Kc = kcentering(K)
% kernel centering fucntion, a staf routine of KDCA algorithm
% By : Eder Santana, PIB
[m,n] = size(K);
Kc = K - ones(m,n)/m - K*ones(n,n)/n + ones(m,m)*K*ones(n,n)/(m*n);

```

kdca.m

```

function [y] = kdca(x,dl,sig,op)
% This is the kernel dca algorithm.
% Input
% x          : observed signal
% dl        : delay of the specific signal
% sig       : variace of the kernel
% op.adaptation : if(op.adaptation = 1) (fixed point adaptation of w,
%                eq. 22), else (eig adaptation of w using eq. 23)
% op.center  : if(op.center = 1) centerate kernel matrix
% op.eigenumber : number of the eigenvector in descendent order
%                (normally 2)
%
% Output
% y          : extracted signal
%
% By         : Eder Santana, PIB
% Date      : 05-07-2009
% Reference  : Eder Santana et. al, Extraction of Specific Signal
%                with Temporal Structure Using Kernel Methods

x = zscore(x')'; % extract mean and renormalize to unit std
[e,d] = eig(x*x'/size(x,2)); % calculate eigenstuffs of the autocorrelation
of x
N      = size(x,2);
X      = d^(-.5)*e'*x; % whitening of x
K      = kprod(X',X,sig); % calculate the correlation kernel
%% extra
%K      = (X'*X + 1).^4;
% [A,B] = eig(K1);
% K      = A(:,1:5);
%%

if op.center == 1

```

Apêndice

```

    K      = kcentering(K); % Centerate data usign eq. 26
end
KK      = [K(1+dl:end,:) ; K(1:dl,:)]; % calculate delayed kernel
K2      = K'*KK; % calculate the action between distributions
K2      = K2+K2';
w = randn(size(K2,2),1); w = w./norm(w); % initialize w

for i=2:1 % put the last number greater than 1 using eq. 16
    KK = [K(1+i*dl:end,:) ; K(1:i*dl,:)]; % calculate delayed kernel
    K2 = K2 + (K'*KK + KK'*K);
end
if op.adaptation == 1
    for i=1:1e2;
        w = K2*w;
        w = w./norm(w);
    end
    y = K*w;
else
    %K2      = K2+K2' + diag( .5*N*c*ones(N,1) );
    [E,D] = eig(K2); % calculate eigenvectors of the new kernel
    [t,I] = sort(diag(D),1,'descend');
    y      = (K*E(:,I(op.eigenumber))); % extract signals (needing a more
efficient
    % algorithm for calculating the second eigenvector)
end
y = zscore(y)';

```

kprod.m

```

function K = kprod(X,Y,sig)
K = zeros([size(X,1) size(Y,2)]);
if size(X,2)==size(Y,1)
K = exp(-0.5*(repmat(diag(X*X'),[1 size(Y,2)]) ...
    + repmat(diag(Y'*Y)',[size(X,1) 1]) ...
    - 2*X*Y)/sig);
    for i = 1:size(X,1)
        for k = 1:i
            x = X(i,:)'; y = Y(:,k); e = x-y;
            K(i,k) = exp(-e'*e/(2*sig^2));% / sqrt(2*pi*sig^2);
            K(k,i) = K(i,k); % due to symmetry of the kernel matrix
        end
    end
else
    error('Matrix dimension must agree')
end

```

plotee.m

```

function h = plotee(x,fs,scale)

% H = PLOTTEE(X,SCALE)
% Plots the elements in the rows of matrix X, each at a line.
% H      : is the handle of the object of each plot
% FS     : sampling frequency
% SCALE  : distance between lines
%
% By     : Eder Santana and Enio Aguiar, PIB (Brazil)
% First version: 01-May-2009.

```

Apêndice

```

if nargin == 1
    fs = 1;
end
if nargin < 3
    scale = max(max(abs(x)));
end

t = (1:size(x,2))/fs;
tantos = size(x,1):-1:1;
y = x + scale*tantos'*ones(1,size(x,2));
h = plot(t,y', 'b'); axis tight;
m = mean(y,2);
set(gca, 'YTick', m(end:-1:1), 'YTickLabel', tantos);

```

redux_dimm.m

```

function psi = redux_dimm(K,d);
% Reduces the dimension to a value 'd' the feature space defined by
% vectors in K

dimmax = size(K,2);
Kc = K - repmat(sum(K), [dimmax 1])/dimmax;
opts.disp = 0;
[V,L] = eigs(Kc,d, 'lm',opts); % calculate the first d eigenvectors
psi = (inv(L)*V'*(Kc))'; % project feature space in the first d
eigenvectors

```

toy_example.m

```

% load ECG signals and noises
s = gera_data(2,1e3);
A = randn(4);
x = A*s;

dls = 140:2:180; %varies time delay from 140 to 180
quantos = length(dls);
tam = length(5:.5:15);
rmse1 = zeros(1,quantos);
rmse2 = zeros(quantos,tam);
op.center = 1; % kdca parameter
op.adaptation = 1; % kdca parameter
op.eigenumber = 1; % kdca parameter

%% DCA
for i = 1 : quantos
    for count = 1:10
        s = gera_data(2,1e3);
        x = randn(4)*s;
        y = dcaewaldo2(x,dls(i));
        if (y*s(1,:)') < 0
            y = -y;
        end
        rmse1(i) = rmse1(i) + prd(s(1,:),y);
        %rmse(i) = rmse(i) + y*s(1,:)'/size(s,2);
    end
    rmse1(i) = rmse1(i)/10;
end

%% KDCA

```

```

for i = 1 : quantos %varies through delays
    i
    c = 1;
    for k = 5:.5:15 %varies through kernel sizes
        for count = 1:10
            count
            s = gera_data(2,1e3);
            x = randn(4)*s;
            y = kdca(x,dls(i),k,op);
            if (y*s(1,:))<0
                y = -y;
            end
            rmse2(i,c) = rmse2(i,c) + prd(y,s(1,:));
            %rmse2(i,c) = rmse2(i,c) + s(1,:)*y/size(y,1);
        end
        rmse2(i,c) = rmse2(i,c)/10;
        c = c + 1;
    end
end
end
%% show errorgram
subplot(211)
imagesc([5:.5:15],dls,rmse2); colormap gray, colorbar
title('A'), ylabel('Delay (\tau)'), xlabel('Kernel size (\sigma)')

% plot kdca's rmse for kernel size = 15
subplot(223)
plot(dls,rmse2(:,end),'-black','LineWidth',2);
%set(gca,'XTickLabel',[163:4:176])
title('B'), ylabel('RMSE'), xlabel('Delay (\tau)')

% plot dca's rmse
subplot(224)
plot(dls,rmse1,'-black','LineWidth',2);
title('C'), ylabel('RMSE'), xlabel('Delay (\tau)')

```