Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Determinação da Direção de Chegada de Sinais de Áudio com Sistema Kinect

Autor:

Raul Lopez Pozuelo

Orientador:

Prof. Sergio Lima Netto, Ph.D.

Orientador:

Prof. Thiago de Moura Prego, D.Sc.

Examinador:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Examinador:

Prof. Amaro Azevedo de Lima, Ph.D.

DEL

Julho de 2012

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica - Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro - RJ CEP 21949-900

DEDICATÓRIA

Este proyecto es para mi familia y amigos, los que están y los que faltan.

AGRADECIMENTO

Eu tive muita sorte na realização deste projeto, por que tive a ajuda de muita gente para poder fazer um bom trabalho.

- Meus dois coordenadores, Sergio Lima e Thiago de M. Prego. Graças a eles eu compreendi melhor os métodos de trabalho brasileiros e como funciona uma universidade tão diferente da minha em Madrid. E também eles foram de muita ajuda quando eu não sabia continuar bem.

- Meu companheiro e amigo José María Fernández. Graças a você pudemos dominar Latex e Matlab sem sofrer demais.

- Minhas colegas de apartamento, as duas Filipas e Venezia, que sempre ajudavam com meu português e sempre com um sorriso.

- Minha amiga Maria Celeste, que sem saber quem era eu antes de chegar ao Brasil, me ajudou sempre sem dúvidas.

- Paloma. Sem suas ajudas de motivação eu teria terminado o projeto um ano depois.

- E muitos agradecimentos a todos aqueles que compartilharam estes incríveis meses no Rio comigo.

RESUMO

Nos últimos anos, vimos um aumento no número de dispositivos que podem reconhecer a identidade e localização do usuário com áudio e vídeo. O propósito deste trabalho é estudar a eficácia dos diferentes algoritmos usados para a determinação da localização da fonte do áudio, e tentar fazer variações para melhorar estes resultados.

O dispositivo Kinect da Microsoft, com seus microfones, foi usado para analisar a eficácia de vários algoritmos. O método de correlação cruzada (CCM) é estudado com profundidade neste trabalho, usando diversos tipos de sinais em diferentes cenários. Depois de estudar este algoritmo, diversas tentativas de aprimoramento foram executadas fazendo variações na estimativa do atraso entre os sinais de diferentes microfones a partir do cálculo da função de correlação cruzada.

Também estudamos outros métodos como os prefiltros *smoothed coherence transforms* (SCOT) e *phase transformation* (PHAT).

Sendo o método mais simples, observamos que o CCM só funciona bem se temos a fonte no *far-field* e na posição frontal dos sensores. Com a fonte se movendo, é quase obrigatório o uso das variações do método CCM ou de prefiltros para obtermos resultados aceitáveis.

Palavras-Chave: Direção de chegada, DoA, Kinect, *far-field*, *near-field*, correlação cruzada, SCOT, PHAT.

ABSTRACT

In the last few years, one has seen an increase of the number of devices that are able to recognize the user's identity and location via video and audio. The purpose of this work is to study the performance of different algorithms used for the determination of the audio-source position and try to make variations to them to improve the results.

In this case, Microsoft's device Kinect with its microphone array has been used to analyze the performance of various algorithms. The cross-correlation method (CCM) is featured heavily in this work, using for its analysis several kinds of signals coming from different positions in the near-field and the far-field. Once this basic algorithm is deeply characterized, we try to improve the results by making variations to the way we pick the number of delay samples between signals based on the cross-correlation function.

Also, we study other methods such as the smoothed coherence transform (SCOT) and the phase transformation (PHAT) pre-filters.

Being the simplest method, it is observed that the CCM only works well when the source is placed in the far-field right in front of the audio sensors. For other situations, the use of CCM variations and pre-filters is almost mandatory to achieve reasonable results.

Key-words: Direction of arrival , DoA, Kinect, far-field, near-field, cross-correlation, SCOT, PHAT.

SIGLAS

CCM - *Cross-Correlation Method* DoA - *Direction of Arrival* FCC - Função de correlação cruzada GCC - *Generalized Cross-Correlation* IDTFT - Transformada de Fourier Inversa LPS - Laboratório de Processamento de Sinais PHAT - *Phase Transformation* SCOT - *Smoothed Coherence Transform* SNR - *Signal-to-Noise Ratio* UFRJ - Universidade Federal do Rio de Janeiro

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Tema

Nos últimos anos, com os grandes desenvolvimentos da tecnologia de reconhecimento de áudio e vídeo, as técnicas de determinação da localização da fonte sonora estão tendo mais e mais importância. Atualmente estas técnicas são empregadas, por exemplo, em câmeras de vigilância e outros sistemas de segurança, em dispositivos de ajuda para pessoas com deficiências ou em sistemas de videojogos.

Neste projeto, precisamente, vamos usar um sistema de videojogos - o Kinect da Microsoft - para executar nossos experimentos de estimativa de direção de chegada a partir de sinais de áudio. Estes experimentos incluem o uso de diferentes algoritmos já existentes na literatura com a plataforma Kinect para ver sua eficácia em diferentes cenários práticos: com ou sem ruído significativo; com ou sem movimento da fonte sonora em relação ao Kinect; com diferentes posições da fonte sonora para o Kinect; e usando diferentes tipos de sinais.

Mais especificamente os objetivos deste projeto de graduação incluem:

- Estudar o problema de estimativa de direção de chegada (DoA, do inglês *direction of arrival*), com foco nos seus aspectos práticos.
- Estudar a plataforma Kinect da Microsoft no contexto do problema de DoA.
- Desenvolver uma base de dados para o problema de DoA usando a plataforma Kinect para aquisição dos dados e considerando diferentes situações práticas.

- Estudar o algoritmo de correlação cruzada (nas suas versões original e generalizada) no problema de estimação de DoA usando a base de dados desenvolvida no âmbito deste trabalho.

Para atingir estas metas, iniciamos nossos estudos com o método de correlação cruzada na sua forma mais simples de todas. Depois de estudar em profundidade este método e os erros que tem e as razões destes erros, tentamos usar esse conhecimento para fazer pequenas variações no jeito que o algoritmo funciona para ter melhores resultados.

Posteriormente, consideramos uma generalização do método original, introduzindo a operação de prefiltragem nos sinais adquiridos. Dois tipos de prefiltro são considerados e seus desempenhos em diferentes situações práticas aqui consideradas são exaustivamente comparados. De modo geral, é possível concluir que o uso destes prefiltros melhoram significativamente a estimativa de DoA obtida.

1.2 Organização do Projeto

No Capítulo 2 fazemos a apresentação do problema de estimação de DoA. Neste sentido, explicamos matematicamente os diferentes cenários que vamos ter, os instrumentos que vamos usar, os sinais que serão considerados em nossas análises e os diferentes modos de gravação.

O Capítulo 3 apresenta o estudo do algoritmo clássico de correlação cruzada para todos os sinais e cenários, permitindo o entendimento em profundidade do funcionamento deste algoritmo.

Outros algoritmos de estimação de DoA são apresentados no Capítulo 4. Neste sentido, inicialmente são apresentadas três variantes do algoritmo clássico. Em seguida, consideramos a família de métodos de correlação cruzada generalizada, utilizando diferentes prefiltros dos sinais de entrada: em particular são considerados os algoritmos SCOT (do inglês *smoothed coherence transform*) e PHAT (também do inglês *phase transformation*).

Finalmente, no Capítulo 5 apresentamos as principais contribuições deste trabalho e apontamos para possíveis extensões do mesmo.

Capítulo 2

Apresentação do Problema

2.1 Introdução

Antes de começar os experimentos para provar a eficácia dos algoritmos, vamos descrever neste Capítulo 2 os diferentes problemas que vamos enfrentar e também os meios com os quais vamos trabalhar durante a realização do projeto.

Primeiro, na Seção 2.2, vamos descrever o problema do "DoA" (do inglês *direction of arrival*), com as equações e os conceitos que fazem parte dele. Depois, na Seção 2.3, vamos descrever o sistema Kinect da Microsoft, a sua configuração e suas características. Finalmente, na Seção 2.4, vamos estabelecer os sinais que vamos gravar e as diferentes modalidades de gravação.

2.2 Problema DoA

Dependendo da distância entre a fonte e o conjunto de microfones, considerando ainda o tamanho deste conjunto de sensores, há duas classes de problemas associados: a estimação da direção de chegada (problema do *far-field*) e a localização da fonte (problema do *near-field*)[2].

Quando as direções de propagação são aproximadamente as mesmas para todos os sensores, é dito que a fonte está no *far-field* do conjunto do microfones. Isto acontece quando a distância entre a fonte e os sensores é muito maior - uma ordem de magnitude acima, por exemplo - do que a distância entre os microfones - parâmetro este também chamado de tamanho de abertura. Nas condições de *far-*

field, as ondas que chegam aos microfones parecem planas porque a curvatura da propagação esférica da onda é pequena em relação ao tamanho de abertura. Quando a distância da fonte ao conjunto de sensores é comparável ao tamanho de abertura, então estamos no *near-field*, onde a curvatura das frentes de onda é significativa em relação ao tamanho de abertura.

Na configuração de *far-field*, ilustrada na Figura 2.1 com dois microfones apenas, podemos apenas estimar a direção de chegada da fonte sonora [1]. Nesta situação, consideramos que a fonte irradia uma onda plana que tem a forma de onda $s(k)$ que se propaga através de um meio não dispersivo. A reta normal à frente de onda tem um ângulo θ com a linha de conexão dos microfones, e o sinal recebido em cada microfone é então uma versão atrasada da versão que chega ao sensor de referência.

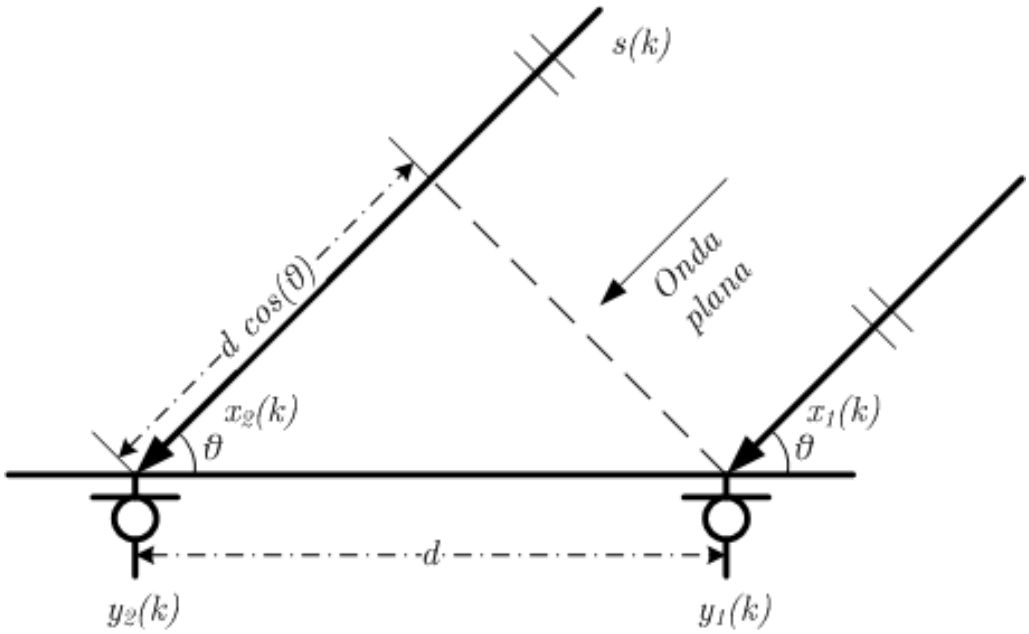

Figura 2.1: Ilustração do problema de estimação de DoA num espaço bidimensional com dois microfones idênticos: a fonte $s(k)$ está localizada no *far-field*, o ângulo de incidência é θ e a distância entre os dois sensores é d.

Para compreender isso, vamos escolher o sensor mais à direita na Figura 2.1 como ponto de referência e vamos denotar por d o espaço que há entre os dois microfones. Neste caso, o sinal no segundo sensor é atrasado do intervalo de tempo necessário para a onda plana em viajar o intervalo de espaço

$$\Delta s = d \cos \theta, \tag{2.1}$$

de modo que a diferença de tempo (atraso temporal) nos dois sensores é dada por

$$\tau_{12} = \frac{d \cos \theta}{c}, \tag{2.2}$$

com c representando a velocidade do som no ar.

Se o ângulo está situado entre 0° e 180° e se τ_{12} é conhecida, então θ é determinado sem erro e vice-versa. Portanto, estimar o ângulo de incidência θ é essencialmente equivalente a estimar a diferença temporal τ_{12}. Em outras palavras, no caso de *far-field*, o problema da estimação de DoA é o mesmo que o problema da estimação da diferença de tempo de chegada (*time-difference-of-arrival*, TDOA).

Embora o ângulo de incidência possa ser estimado com o uso de dois ou mais sensores, a distância entre a fonte sonora e o conjunto de microfones é difícil (se não impossível) de determinar se a fonte sonora está no *far-field* dos sensores. Contudo, se a fonte está localizada no *near-field*, como é ilustrado na Figura 2.2, é possível estimar não só o ângulo no qual a onda chega a cada sensor, mas também a distância entre a fonte e cada microfone.

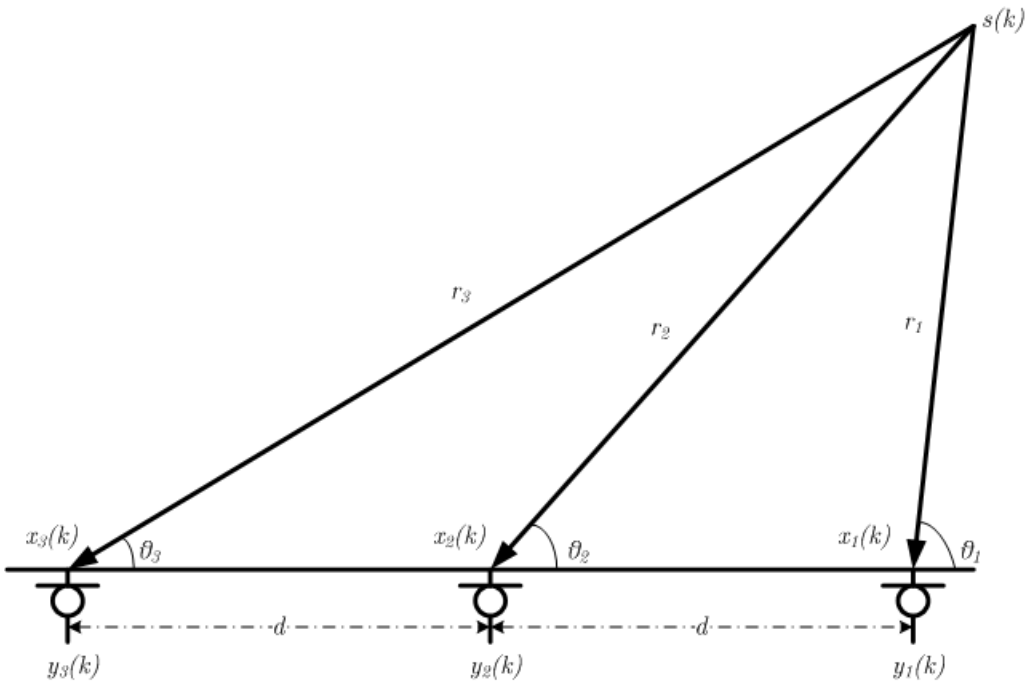

Figura 2.2: Ilustração do problema de localização da fonte sonora com um arranjo linear de microfones: a fonte $s(k)$ está localizada no *near-field* e a distância entre cada microfone adjacente é d.

Para observar isso, vamos considerar um exemplo simples usando três sensores, como vemos na Figura 2.2. Novamente, escolhemos o microfone mais à direita

como sensor de referência. Denotando os ângulos de incidência e as distâncias entre a fonte sonora e os n microfones, respectivamente, como θ_n e r_n, sendo $n = 1, 2, 3$, o TDOA entre o segundo e o primeiro sensor é dado por

$$\tau_{12} = \frac{r_2 - r_1}{c} \tag{2.3}$$

e o TDOA entre o terceiro e o primeiro sensor é

$$\tau_{13} = \frac{r_3 - r_1}{c}. \tag{2.4}$$

Pela Lei dos Cossenos, obtemos

$$r_2^2 = r_1^2 + d^2 + 2r_1 d\cos\theta_1, \tag{2.5}$$

$$r_3^2 = r_1^2 + 4d^2 + 4r_1 d\cos\theta_1. \tag{2.6}$$

Para um sistema prático com um arranjo linear, se d, τ_{12} e τ_{13} são conhecidos, então podemos calcular todos os parâmetros desconhecidos θ_1, r_1, r_2 e r_3 resolvendo as equações (2.3)–(2.6). Se aplicamos também a Lei dos Senos, podemos obter ainda uma estimativa de θ_2 e θ_3, de modo que toda a informação sobre a posição da fonte relativa ao arranjo de microfones pode ser determinada.

Pelos desenvolvimentos algébricos descritos acima, independentemente se a fonte está localizada no *far-field* ou no *near-field*, a etapa fundamental para obter a informação do origem do som é a de estimativa do TDOA entre os diferentes microfones. Este problema de estimativa temporal seria um problema fácil se os sinais recebidos fossem simplesmente uma versão atrasada e escalada uns dos outros. Em realidade, contudo, o sinal de origem está geralmente dentro de ruido ambiente, porque existe um entorno natural no qual a presença de ruido é inevitável. Além disso, cada sinal observado pode ter muitas réplicas dele mesmo devidas às reflexões em objetos, anteparos e paredes. Este efeito de propagação de múltiplos percursos (*multipath*) introduz cópias atrasadas e distorções espectrais nos sinais recebidos - compondo a chamada reverberação - e deteriora fortemente o sinal de origem. Além de tudo isto, a fonte sonora também pode ter movimento, resultando numa mudança no atraso temporal de chegada ao longo do tempo. Todos esses fatores podem tornar o problema de estimativa de TDOA uma tarefa complicada e desafiante.

Contando, com os problemas que mencionados no parágrafo anterior e a quantidade de fontes de som, podemos diferenciar quatro cenários básicos de propagação, dependendo da fonte e seu entorno:

- Uma fonte sem reverberação (*single-source free-field model*).
- Várias fontes sem reverberação (*multiple-source free-field model*).
- Uma fonte com reverberação (*single-source reverberant model*).
- Várias fontes com reverberação (*multiple-source reverberant model*).

Neste projeto vamos desenvolver o mais simples de todos: o cenário com apenas uma fonte e, na medida do possível, sem reverberação, como detalhado a seguir.

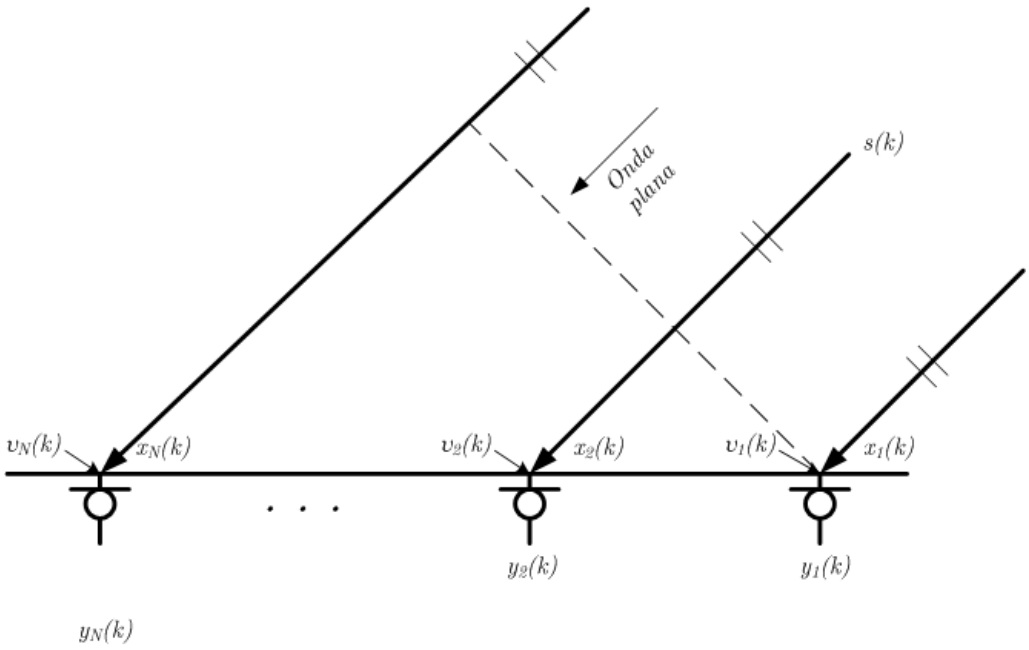

Figura 2.3: Ilustração do modelo ideal do problema de apenas uma fonte sonora sem reverberação.

Vamos supor que temos apenas uma fonte de som no campo sonoro e que vamos usar um array de N microfones - no cenário do sistema Kinect, como mencionado mais adiante na Seção 2.3, temos $N = 4$. Num espaço aberto anecoico[1] como indicado na Figura 2.3, a sinal de som $s(k)$ se propaga radialmente e o nível de som cai em função da distância à fonte. Se escolhemos o microfone mais à direita como

[1]Uma sala anecoica é uma sala especialmente projetada para absorver o som que atinge todas as paredes da sala, incluindo o chão e o teto da sala, cancelando, idealmente, todos os efeitos do eco e reverberação do som.

o ponto de referência, o sinal capturado pelo n-ésimo microfone no momento k pode ser descrito como:

$$\begin{aligned} y_n(k) &= \alpha_n s(k - t - \tau_{n1}) + \upsilon_n(k) \\ &= \alpha_n s(k - t - \mathcal{F}_n(\tau)) + \upsilon_n(k) \\ &= x_n(k) + \upsilon_n(k), \end{aligned} \tag{2.7}$$

para $n = 1, 2, \ldots, N$, onde os coeficientes α_n, que em geral pertencem ao intervalo $0 \leq \alpha_n \leq 1$, são os fatores de atenuação devidos aos efeitos de propagação; t é o tempo de propagação entre a fonte de som desconhecida ao sensor 1; $\upsilon_n(k)$ é a parcela de ruído aditivo no n-ésimo sensor, que assumimos descorrelacionada com $s(k)$ e com os ruídos observados nos outros sensores; τ é o TDOA - também chamado de atraso relativo - entre os sensores 1 e 2; e $\tau_{n1} = \mathcal{F}_n(\tau)$ é o TDOA entre os sensores 1 e n, sendo $\mathcal{F}_1(\tau) = 0$ e $\mathcal{F}_2(\tau) = \tau$. Para $n = 3, ..., N$ a função $\mathcal{F}_n$ depende não só de τ mas também da geometria do conjunto de microfones. Por exemplo, numa situação de *far-field* (propagação de onda plana), com um arranjo linear e igualmente espaçado, temos

$$\mathcal{F}_n(\tau) = (n-1)\tau; \quad n = 2, ..., N, \tag{2.8}$$

e para um arranjo linear não igualmente espaçado, temos

$$\mathcal{F}_n(\tau) = \frac{\sum_{i=1}^{n-1} d_i}{d_1}\tau; \quad n = 2, ..., N, \tag{2.9}$$

onde d_1 é a distância entre os microfones de números i e $i+1$ ($i = 1, ..., N-1$).

No caso de *near-field*, $\mathcal{F}_n$ depende também da posição da fonte de som. É importante dizer que $\mathcal{F}_n(\tau)$ pode ser uma função não linear de τ para uma geometria não linear do arranjo de microfones, mesmo no cenário de *far-field*. Em geral τ não é conhecida, mas a geometria do conjunto de microfones sim é conhecida. Portanto a formulação matemática de $\mathcal{F}_n(\tau)$ está bem definida ou dada. Para este modelo, o problema da TDE (*time-delay-estimation*) é formulado como o problema de determinar uma estimativa $\hat{\tau}$ do atraso temporal correto τ usando um conjunto finito de amostras observadas.

2.3 O Sistema Kinect

Para resolver o problema de determinar a direção de chegada do som precisamos um arranjo de microfones com pelo menos dois sensores independentes para capturar sons. No presente projeto, escolhemos o sistema Kinect da Microsoft, um periférico auxiliar compatível com a plataforma Xbox 360 de videojogos, conforme visto na Figura 2.4.

Figura 2.4: Sistemas Kinect e Xbox 360 de Microsoft.

O sistema Kinect foi desenvolvido usando uma tecnologia inventada em 2005 por Zeev Zalevsky, Alexander Shpunt, Aviad Maizels e Javier García. O nome original do dispositivo era "Project Natal", em homenagem a Natal, a cidade brasileira onde o diretor da Microsoft Alex Kipman, que foi quem primeiro vislumbrou a projeto, nasceu. O sistema Kinect foi anunciado como periférico para o console Xbox 360 em 2009, e depois de diferentes exposições em feiras do setor foi finalmente colocado à venda no ano 2010. Poucos meses depois, em junho de 2011, foi anunciado o lançamento da plataforma de desenvolvimento oficial para uso não comercial.

Em termos práticos, o Kinect é um sistema de captura de movimento. Fisicamente é uma caixa preta que é colocada horizontalmente acima ou debaixo da tela

de jogo de modo a registrar os movimentos do(s) jogador(es) e controlar suas ações no jogo sem a necessidade de um controlador tradicional nas mãos. Este registro visual do jogador é feito usando dois canais de captura: áudio e vídeo.

Figura 2.5: Posicionamento das três câmaras do Kinect.

Para a captura do vídeo, o sistema Kinect tem três câmaras com *autofocus*, localizadas como podemos ver na Figura 2.5. Essas câmaras são dos seguintes tipos:

- Um projetor de luz infravermelha.
- Um sensor de imagem de profundidade.
- Uma câmara do espectro visual normal.

O projetor infravermelho funciona junto com o CMOS monocromático para poder "ver" a habitação em 3D independentemente das condições de luz do lugar. A câmara convencional RGB ajuda no reconhecimento facial e de outras características de detecção graças à sua captura das três cores vermelho, verde e azul. Todas as câmaras funcionam com uma resolução de 640x480 e uma taxa de atualização de 30 Hz. Segundo as especificações do fabricante, o campo de visão do sistema Kinect é de 57º.

Dado que o objetivo deste projeto é a detecção de áudio, o importante é conhecer bem como funciona esta detecção de som no sistema Kinect[3]. O periférico da Xbox 360 tem um arranjo de microfones que é capaz de separar sinais de fala

de outros sons ambiente, para otimizar o desempenho da interface controlada por comandos de voz. Estes microfones processam áudio a 16 bits com uma taxa de amostragem de 16 kHz.

Figura 2.6: Localização dos microfones no dispositivo Kinect.

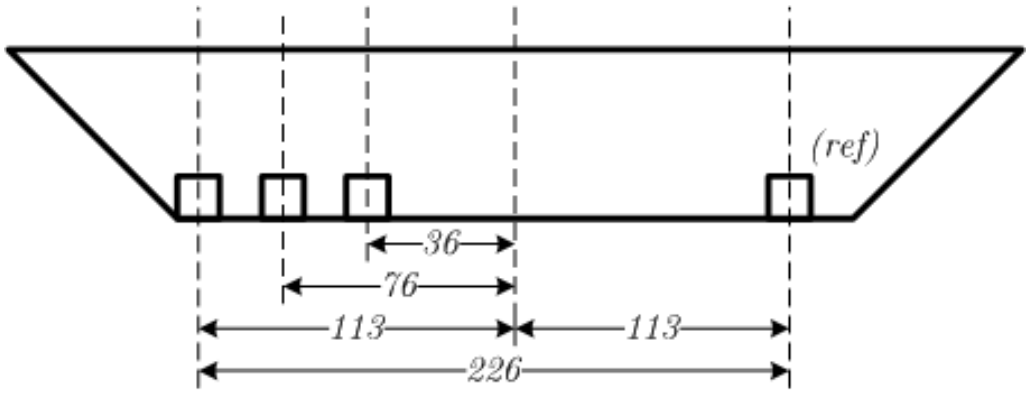

Figura 2.7: Vista em planta do Kinect, com a posição exata dos microfones, em milímetros.

No caso do Kinect, contamos com quatro microfones em linha, três deles no lado esquerdo e o outro no lado direito. Todos eles ficam na parte de baixo do dispositivo, conforme indicado nas Figuras 2.6 e 2.7. Esta geometria do Kinect permite calcular a posição aproximada de uma fonte de som, levando-se em conta a diferença temporal entre os sinais que chegam aos diferentes microfones. Quando a posição do som é calculada, um complexo algoritmo mistura os sinais de todos

os microfones para obter uma melhor estimativa do sinal originalmente gerado pela fonte sonora. Além disso, um filtro exclui todos os componentes espectrais do sinal fora do intervalo de 80 Hz a 1100 Hz, diminuindo o efeito do ruído ambiente sobre um possível sinal de voz. Finalmente, um algoritmo reduz o efeito de reverberação a partir de uma calibração inicial do ambiente, que é continuamente refeita para uma melhor modelagem do ambiente e do sistema de reconhecimento de voz. Para implementar todos estes algoritmos, o Kinect tem vários processadores digitais de sinais (DSPs) próprios.

2.4 Base de Dados: Sinais de Análise

Para viabilizar as análises dos algoritmos de estimação de DoA, fizemos uma serie de gravações com diferentes configurações para as fontes e para os sensores. Mas não só as condições de gravação são importantes, pois também são importantes os tipos de sinais de áudio escolhidos para a fonte, que no nosso caso continham as seguintes características: (i) Devem representar diferentes tipos de fonte sonora; (ii) Devem ter a menor quantidade possível de ruido não deliberado; (iii) Devem ser curtos e controlados.

Idealmente, o melhor seria ter um numero grande de sinais para assim poder ter a maior quantidade de amostras possíveis de todos os tipos de imagináveis de fenômenos acústicos. Por limitações de tempo e espaço, porém, utilizamos as seguintes três classes de sinais:

- Sinais de Banda Estreita: Um dos tipos de sinal típicos que temos que estudar são os sinais de banda estreita, ou seja, com um espectro limitado de frequências. Na base proposta, usamos três sinais que ficam dentro das frequências que o sistema Kinect suporta, isto é, dentro da banda de frequências da fala humana. Estes sinais têm uma duração média de cerca de 10 segundos, e são:

  s1: Um tom de frequência $f1 = 500$ Hz.

  s2: Um tom de frequência $f2 = 1000$ Hz.

  s3: Um tom de frequência $f3 = 3000$ Hz.

- Sinais de Fala: Foram usados ainda dois sinais de fala em português gravados numa câmara anecoica de modo a minimizar o efeito de reverberação. Estes sinais têm entre 5 e 8 segundos de duração e são referenciados neste projeto como:

  s4: Sinal de fala anecoica 1 (mulher).

  s5: Sinal de fala anecoica 2 (homem).

- "Chaleira": Finalmente, foi gravado o som do tipo "assobio" gerado por uma chaleira, com duração aproximada de 5 segundos:

  s6: Assobio de uma chaleira.

  Este sinal é interessante por ser um sinal tonal, com frequência fundamental variando ao logo do tempo.

Os seis sinais descritos acima foram usados na gravação da base de dados aqui proposta. A gravação em si foi realizada na Sala Paulo S. R. Diniz, dentro do Laboratório de Processamento de Sinais (LPS-II, Sala I-146 do Centro de Tecnologia da UFRJ). Esta sala possui boa resposta acústica e para a gravação dos sinais contamos com o seguinte equipamento:

- Sistema Kinect: Já descrito na Seção 2.3.
- Alto-falante: Um alto-falante mono de PC.
- Laptop #1: Ligado ao alto-falante para reproduzir os sinais de áudio.
- Laptop #2: Ligado ao Kinect para capturar os sinais recebidos.
- Gerador de ruído: Para executar a gravação com níveis altos de ruído, neste caso, uma máquina de ar condicionado.

Observando a planta-baixa da sala utilizada, definimos o canto noroeste como o ponto $(0, 0, 0)$ cm, conforme indicado na Figura 2.8.

Nestas condições, o alto-falante foi apoiado numa mesa, paralela e a uma distância de 110 cm da parede norte, com altura de 85 cm, de modo que o alto-falante foi situado numa posição descrita por $(x, 110, 85)$ cm, com x variando para

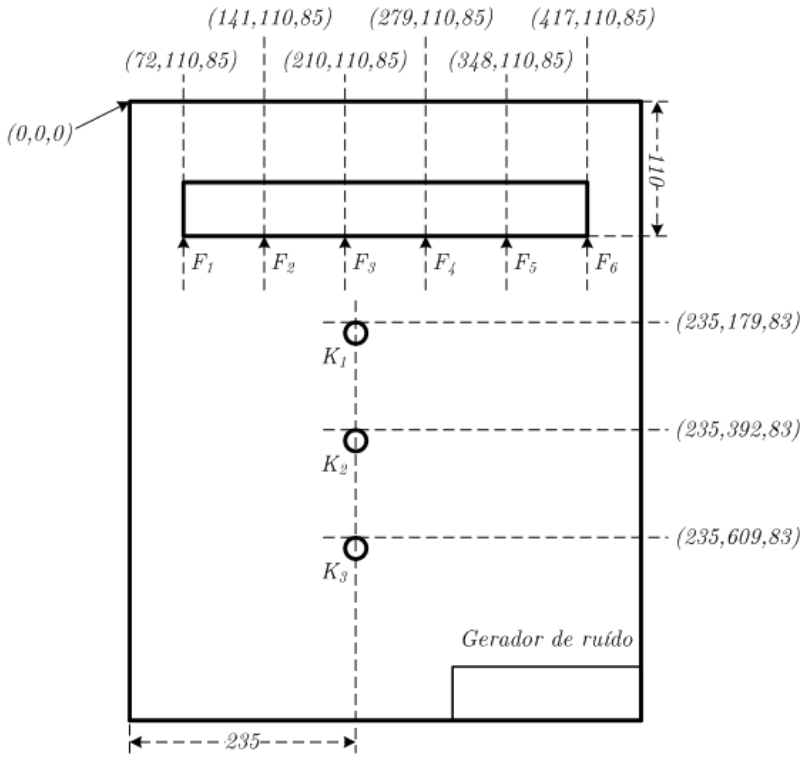

Figura 2.8: Vista em planta-baixa da sala de gravação com indicação esquemática das diferentes posições do sistema Kinect $K\beta$ e da fonte de sons $F\alpha$ (todas as medidas são em centímetros).

diferentes configurações de gravação. Dividindo a mesa em seis posições equidistantes, utilizamos as seguintes posições do alto-falante na sala:

- $F1 = (72, 110, 85)$ cm;
- $F2 = (141, 110, 85)$ cm;
- $F3 = (210, 110, 85)$ cm;
- $F4 = (279, 110, 85)$ cm;
- $F5 = (348, 110, 85)$ cm;
- $F6 = (417, 110, 85)$ cm.

Para o sistema Kinect, foram escolhidas três localizações distintas, todas elas a uma distância de 235 cm da parede oeste - temos que ter em conta que esta é a distância até um dos microfones de Kinect, o microfone de referência, que neste caso é o sensor que fica mais à direita do dispositivo. A altura dos sensores, já incluindo a

altura do próprio Kinect e da mesa, foi de 83 cm, bastante próxima da altura do alto-falante propriamente dito, tornando o problema em questão bidimensional. Para decidir a distância entre as diferentes posições do sistema Kinect e os alto-falantes, é preciso ter em conta os conceitos de *far-field* e *near-field* colocados na Seção 2.2. Para este projeto, consideramos *far-field* a todos os pontos que fiquem mais longe do dispositivo que dez vezes a medida da abertura do conjunto de microfones, que neste caso é de 22,6 cm. Portanto, neste caso, o *far-field* inclui todos os pontos com distância acima de 226 cm do Kinect, e o *near-field* inclui os demais pontos da sala. As posições finais escolhidas para o Kinect quando da montagem da base de dados foram:

- $K1 = (235, 179, 83)$ cm (*near-field*);
- $K2 = (235, 392, 83)$ cm (*far-field*);
- $K3 = (235, 609, 83)$ cm (*far-field*),

conforme ilustrado na Figura 2.8.

A título de exemplo, mostramos na Figura 2.9 duas fotografias com diferentes configurações de gravação: Na figura da esquerda, a fonte sonora está na posição $F1$ e o sistema Kinect na posição $K1$ (*near-field*). Na figura da direita, a fonte está na posição $F6$ e o receptor na posição $K2$ (*far-field*).

Figura 2.9: Exemplos de configuração de gravação: Esquerda - Kinect na posição $K1$ e fonte na posição $F1$; Direita - Kinect na posição $K2$ e fonte na posição $F6$.

Como colocado anteriormente, e como podemos ver no esquema da Figura 2.8, a sala de gravação tem uma fonte de ruído que, junto com uma diminuição apropriada

no volume do alto-falante, pode ser usada para atenuar a clareza do sinal de chegada no Kinect. Neste sentido, portanto, consideramos as gravações sem ruído ($r1$) e com ruído e devida diminuição de volume ($r2$).

Agora que definimos as diferentes localizações $F\alpha$ da fonte sonora e $K\beta$ do sistema de captura Kinect, as duas configurações $r\gamma$ de ruído e os seis tipos diferentes de sinais, consideramos dois modos principais de gravação, descritos a seguir.

2.4.1 Modo de Gravação 1

Neste modo, o alto-falante e o sistema Kinect foram posicionados de forma fixa. Inicialmente foi feita uma gravação sem ruído para todas as posições $K\beta$ do Kinect, todas as posições $F\alpha$ da fonte, e para os seis tipos de sinais. Quando concluídas as gravações para todas as posições de alto-falante e do Kinect, seguimos com as gravações na presença significativa do ruído. De acordo com este esquema, a ordem de gravação, escrita de um jeito esquemático foi da seguinte forma:

- $r1.K1.F1.s1, r1.K1.F1.s2, \ldots, r1.K1.F1.s6, r1.K1.F2.s1, \ldots, r1.K1.F6.s6,$ $r1.K2.F1.s1, \ldots, r1.K3.F6.s6, r2.K1.F1.s1, \ldots, r2.K3.F6.s6$

Assim, ao final do primeiro modo de gravação temos um total de 216 arquivos de áudio gravados por cada microfone, a metade dos quais com e a outra metade sem ruído.

2.4.2 Modo de Gravação 2

Para o segundo modo de gravação, a fonte sonora foi feita móvel, percorrendo o percurso entre as posições $F2$ e $F4$ acima descritas. Nesta situação, escolhemos um sinal de cada tipo: $s1$, $s4$ e $s6$. Assim temos um sinal de banda estreita, um de fala e a chaleira. Além disto, o sistema Kinect foi colocado na posição intermediária das três, ou seja, no *far-field* em $K2$, conforme ilustrado na Figura 2.10.

Neste modo de gravação, temos também uma nova variável que é a velocidade de movimento do alto-falante, para a qual foram definidos duas situações distintas:

- v_1: Velocidade devagar - o alto-falante vai fazer um caminho de ida de $F4$ a $F2$

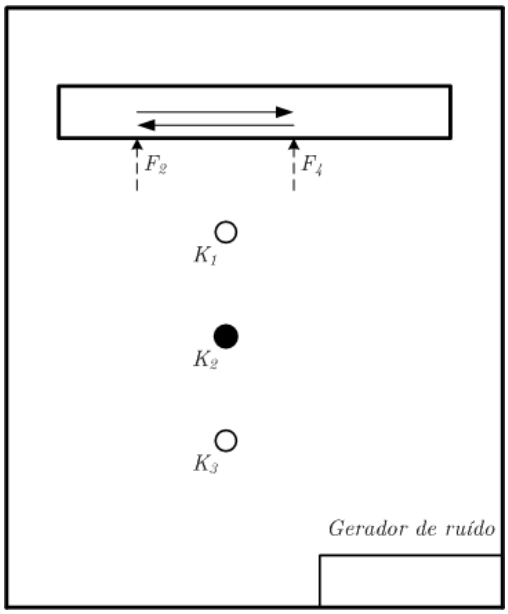

Figura 2.10: Vista esquemática da sala de gravação no modo de gravação 2. Nesse caso, o Kinect está fixo na posição $K2$ e o alto-falante é movido entre as posições $F4$ e $F2$.

- v_2: Velocidade rápida - neste caso o alto-falante perfaz o caminho ida-e-volta entre as posições $F2$ e $F4$ no mesmo tempo, aproximadamente, que dura o sinal.

  Neste caso, o ordem de gravação, com $K2$ sempre fixo, foi da forma:

- $r1.v1.s1, r1.v2.s1, r1.v1.s4, r1.v2.s4, r1.v1.s6, r1.v2.s6,$
  $r2.v1.s1, r2.v2.s1, r2.v1.s4, r2.v2.s4, r2.v1.s6, r2.v2.s6$

Desta forma, no modo 2 de gravação, obtemos 12 conjuntos de sinais para cada microfone com a fonte em movimento.

Capítulo 3

Método da Correlação Cruzada

3.1 Introdução

No anterior Capítulo 2 vimos os diferentes elementos que vamos usar neste projeto para a realização dos experimentos: os dispositivos, os sinais e as diferentes configurações de gravação. Agora, neste Capítulo vamos começar o estudos dos algoritmos.

O primeiro algoritmo que vamos descrever neste projeto para a estimativa do TDOA (*time-difference-of-arrival*) é o método mais simples e o mais direito: o método de correlação cruzada (CCM, do inglês *cross-correlation method*). Este método utiliza a função de correlação cruzada entre os sinais provenientes da mesma fonte que chegam a dois microfones separados por uma distância conhecida para estimar o atraso (TDOA) entre elas.

Na Seção 3.2, inicialmente veremos as bases teóricas do método de correlação cruzada, os conceitos e equações. Na Seção 3.3 vamos explicar a implementação do método, as variáveis que vamos usar e o problema da precisão finita da solução.

Em seguida, apresentamos uma série de resultados experimentais. Nas Seções 3.4 e 3.5, vemos primeiro a eficácia do algoritmo com a fonte no *far-field* dos microfones, nos casos de ausência e presença de ruído. Já na Seção 3.6, fazemos a análise para a fonte no *near-field* do Kinect. Na Seção 3.7, vemos como funciona o CCM se a fonte está se movendo, para diferentes posições dos microfones.

3.2 Teoria

No presente estudo, vamos considerar o modelo de uma única fonte, sem reverberação, e, apenas para facilitar nossa exposição, com apenas $N = 2$ sensores. A função da correlação cruzada (FCC) entre os dois sinais a e b quaisquer é definida como[1]:

$$r_{a.b}^{CC}(p) = E[a(k).b(k+p)]. \tag{3.1}$$

Substituindo a equação (2.7) do modelo de uma única fonte no *free-field*, que foi explicado na equação (3.1), obtemos

$$r_{y_1.y_2}^{CC}(p) = \alpha_1\alpha_2 r_{s.s}^{CC}(p-\tau) + \alpha_1 r_{s.v_2}^{CC}(p+k) + \alpha_2 r_{s.v_1}(p-k-\tau) + r_{v_1.v_2}(p). \tag{3.2}$$

Se assumimos que $v_n(k)$, para $n = 1, 2$ não está correlacionado com o sinal $s(k)$ e com o ruído observado no outro sensor, podemos concluir que $r_{y_1.y_2}^{CC}(p)$ tem seu máximo em $p = \tau$. Então, se temos a CCF, podemos obter uma estimativa do TDOA entre $y_1(k)$ e $y_2(k)$ da forma

$$\hat{\tau}^{CC} = \arg\max_p r_{y_1.y_2}^{CC}(p). \tag{3.3}$$

onde $p \in [-\tau_{max}, \tau_{max}]$ e τ_{max} é o máximo atraso possível.

Em poucas palavras, o que podemos dizer deste algoritmo é que para achar o atraso de um sinal em relação a outro temos que executar uma correlação cruzada entre eles e procurar pelo pico mais alto desta função. A diferença de amostras entre o pico e o centro da FCC é a estimativa de atraso obtida.

3.2.1 Exemplo

Para facilitar o entendimento do funcionamento do algoritmo CCM, apresentamos a seguir um exemplo prático simples.

Neste caso, vamos usar dois sinais cossenoidais de $L = 1001$ amostras cada um, denotados por $x1$ e $y1$ e mostrados na Figura 3.1. Estes sinais estão atrasados de exatamente 2 amostras entre si, simulando assim o atraso de um único sinal que chega a dois microfones diferentes mas muito próximos. Determinando a CCF entre estes dois sinais, obtemos o resultado da Figura 3.2, onde o ponto central corresponde ao atraso $p = 0$. Neste exemplo, o valor máximo da CCF, calculada

com o comando `xcorr` do MATLAB, está na amostra $k^* = 1003$, que corresponde a um atraso $p^* = k^* - L = 2$, como esperado.

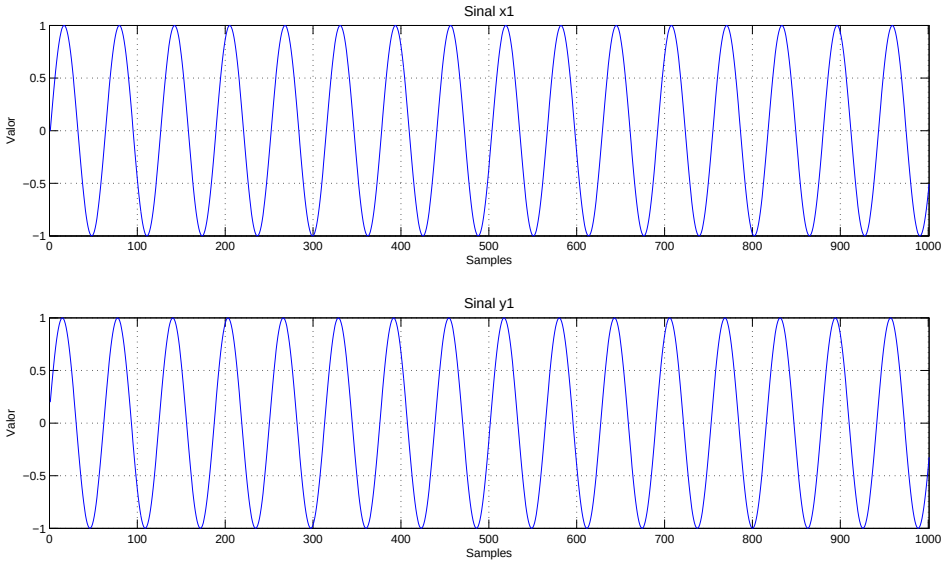

Figura 3.1: Sinais $x1$ e $y1$.

Na prática, o funcionamento do algoritmo CCM é afetado pela presença de ruído ou reverberação nos sinais observados ou mesmo por características intrínsecas do tipo de sinal emitido pela fonte ou ainda pela posição e/ou geometria do arranjo de microfones, como será visto adiante neste capítulo.

3.3 Implementação

Como ilustrado acima, o cálculo da FCC foi realizado no MATLAB usando o comando `xcorr`. Para o algoritmo CCM, como um todo, fornecemos as seguintes constantes e variáveis:

- Sinal: Temos que especificar os sinais que vamos analisar, dentre aqueles que compõem o banco de dados descrito na Seção 2.4.

- Localizações dos sensores: Temos que escolher a posição dos microfones dentre as possíveis combinações descritas na Seção 2.3. Como os algoritmos de correlação cruzada trabalham com pares de sinais, precisamos fazer cada análise escolhendo dois sensores dos quatro disponíveis. Neste caso, sempre usamos

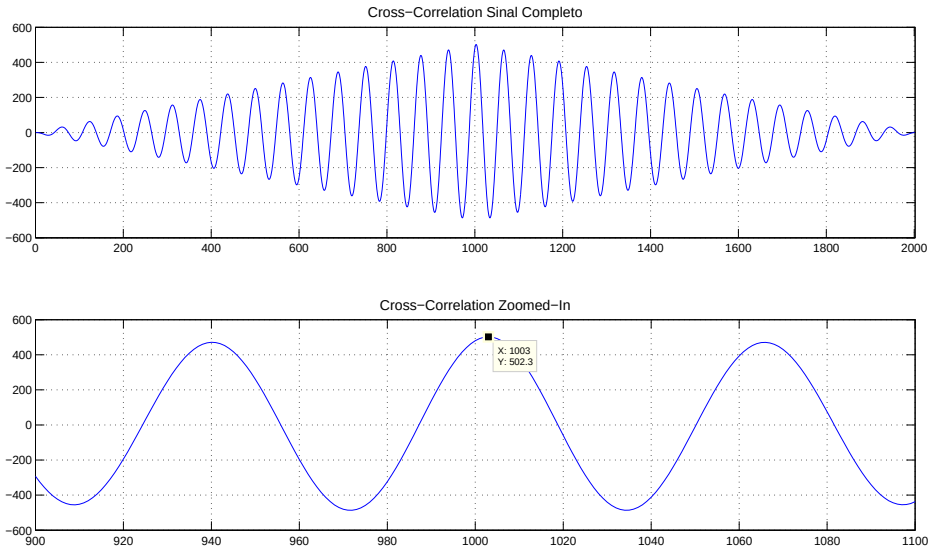

Figura 3.2: Resultado do correlação cruzada entre $x1$ e $y1$ usando o comando `xcorr` do MATLAB.

como sensor de referência o sensor que está no canto direito e definimos por $\delta_{1,n}$ a distância deste microfone para o n-ésimo microfone, com $n = 1, 2, 3$.

- Localização da fonte: A posição correta da fonte funciona como referência para efeito de comparação do resultado estimado pelo algoritmo.

- Velocidade do som (c): Esta informação mapeia o atraso estimado pela FCC na estimativa da DoA.

- Frequência de amostragem (f_s): Assim como a velocidade do som, a frequência de amostragem do sinal de som gravado permite o mapeamento da informação do atraso estimado pela FCC na estimativa desejada de DoA.

- Atraso máximo: Para cada par de microfones, tem-se um atraso máximo possível dado por $p_{max} = \delta_{1,n} * f_s/c$. Se o valor de atraso estimado é maior que o atraso máximo, o algoritmo retorna alguma mensagem de erro.

Dentre as informações determinadas pelo algoritmo CCM, têm-se:

- Atraso: O número de amostras p estimado entre os dois sinais.

- TDOA: A diferença temporal $\Delta t = p/f_s$ que há entre a chegada do sinal ao primeiro microfone e a chegada do sinal ao segundo microfone.

- Distância adicional: A distância extra percorrida $\Delta s = c.\Delta_t$ pelo sinal até o microfone mais distante.

- Ângulo estimado: O ângulo estimado da DoA dado por

$$\hat{\theta} = \arccos \frac{\Delta s}{\delta_{1,n}}. \tag{3.4}$$

3.3.1 O Problema da Resolução Finita

A estimativa da DoA obtida anteriormente possui uma resolução geométrica associada à resolução temporal definida pela frequência de amostragem usada na aquisição dos sinais de entrada. Para entendermos este efeito, devemos lembrar que o atraso obtido a partir da FCC entre os dois sinais é determinado em um número inteiro de amostras. Esta resolução numérica no número de amostras entre dois sinais vai definir a resolução numérica da estimativa de DoA resultante.

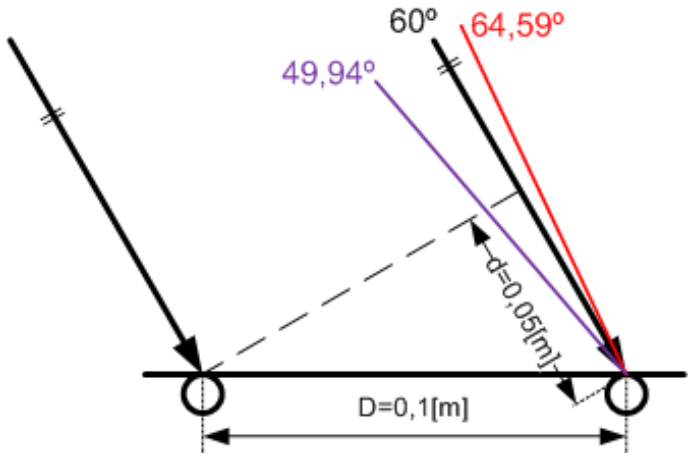

Figura 3.3: Configuração exemplo para ver o problema de resolução. Com cores vemos as linhas que o algoritmo pode dar como corretas, e a preta é a linha com o ângulo correto.

Em termos numéricos, por exemplo, vamos supor que temos dois microfones separados de $D = 10$ cm que recebem um sinal de áudio, com frequência de amostragem de $f_s = 16000$ amostras/s, desde o *far-field* com um ângulo de $\theta = 60°$ com

a horizontal. Usando as relações dadas na Seção 3.3, temos os seguintes resultados:

$$\begin{aligned} \theta = 60^\circ &= \arccos \frac{\Delta s}{\delta} \\ \Rightarrow \Delta s &= 5 \text{ cm} \\ \Rightarrow \Delta t &= \frac{\Delta s}{c} = \frac{0{,}05}{343{,}2} = 1{,}456 \times 10^{-4} \text{ s} \\ \Rightarrow p &= \Delta t \times f_s = 1{,}456 \times 10^{-4} \times 16000 = 2{,}331 \text{ amostras.} \end{aligned} \tag{3.5}$$

Sendo assim, para obter o resultado exato, o pico da FCC deveria estar na posição relativa ao atraso de 2,331 amostras. Porém, como mencionamos anteriormente, o algoritmo de busca do pico da FCC retorna, a princípio, apenas valores inteiros. No caso, o resultado deste algoritmo seria $p = 2$ ou $p = 3$ amostras, que correspondem, respectivamente, conforme ilustrado na Figura 3.3, às seguintes estimativas de DoA:

- Para 2 amostras: $\hat{\theta} = 64{,}59^\circ$;
- Para 3 amostras: $\hat{\theta} = 49{,}94^\circ$.

Assim, para esta configuração do problema, o menor erro possível é de 4,59°, o que pode ser crítico ou não dependendo da aplicação prática de interesse.

Uma outra forma de ver este problema é estudando a influência da distância entre os microfones. Vamos explicar outro cenário, neste caso com 3 microfones. As distâncias do sensor 1 (tomado como referência) para os sensores 2 e 3 são, neste exemplo, $\delta_{1,2} = 10$ cm e $\delta_{1,3} = 1$ m, respectivamente. A frequência de amostragem é de 16000 amostras/s - a mesma que é utilizada pelo Kinect. Nestes casos, os atrasos entre os sinais obtidos pelos microfones 2 e 3, sempre em relação ao microfone 1 de referência, para um DoA de $\theta = 60^\circ$ seriam de $p_{1,2} = 2{,}33$ e $p_{1,3} = 23{,}33$ amostras, respectivamente. Truncando estes valores para números inteiros, as estimativas de DoA obtidas em cada caso seriam:

- Amostras de atraso estimadas para sensores 1-2:
  - $p_{1,2} = 2 \Rightarrow \hat{\theta} = 64{,}59^\circ$.
  - $p_{1,2} = 3 \Rightarrow \hat{\theta} = 49{,}49^\circ$.
- Amostras de atraso estimadas para sensores 1-3:
  - $p_{1,3} = 23 \Rightarrow \hat{\theta} = 60{,}43^\circ$.

– $p_{1,3} = 24 \Rightarrow \hat{\theta} = 59{,}01°$.

Note, agora, pela Figura 3.4, que o erro mínimo para o par 1-3 de microfones é bem menor do que o erro mínimo para o par 1-2, devido à maior distância entre os microfones 1-3. De fato, é possível observar que a resolução geométrica obtida diminui (melhora) com o aumento da distância entre os microfones.

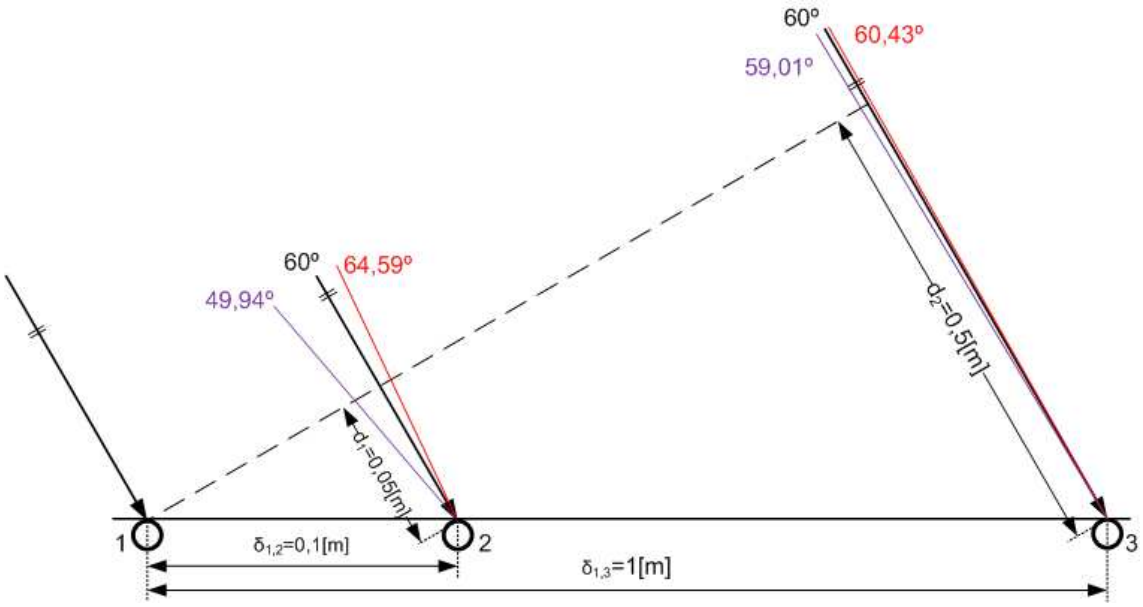

Figura 3.4: Análise da influência da distância entre os microfones na resolução geométrica resultante para a estimativa de DoA.

De modo geral, então, no sistema Kinect, a melhor resolução geométrica será obtida pelo par de microfones 1-4, que como vimos na Seção 2.3, são os que ficam mais separados, com uma distância entre eles de $0,226$ m, enquanto que as distâncias 1-2 e 1-3 são $0,149$ e $0,189$ metros respectivamente.

Isto não significa, porém, que devemos descartar as estimativas obtidas pelos outros pares de microfones, como será visto mais adiante.

3.4 Experimento: Sinais no *Far-Field* Sem Ruído

Como colocado na Seção 2.4, temos uma quantidade grande de sinais e cenários de gravação. Numa primeira análise, vamos considerar apenas a posição $K3$ da fonte no *far-field* e não há componentes ruidosos significativos. Neste caso,

vamos considerar os três tipos de sinais (senoidal, de fala e da chaleira) nas subseções a seguir.

3.4.1 Sinais Senoidais

3.4.1.1 Posição Central da Fonte

Para uma primeira medida, vamos usar a fonte na posição $F3$, que fica quase na vertical do Kinect ($\theta = 88,42°$). Neste caso, o resultado do algoritmo CCM (tanto do atraso quanto da DoA estimada) com o par 1-4 de microfones é mostrado na Figura 3.5, para os três sinais senoidais ($s1$, $s2$ e $s3$), em função do comprimento (em número de amostras) da janela utilizada.

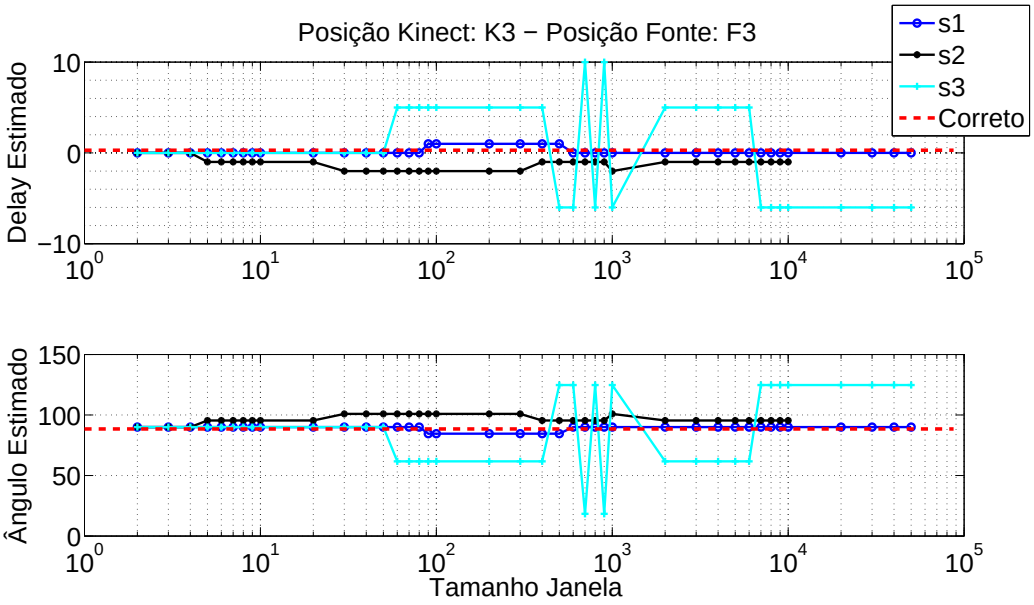

Figura 3.5: Resultados das medidas para sinais de banda estreita, para a fonte na posição $F3$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

Como pode ser visto na Figura 3.5 os resultados não são muito robustos e o comportamento do algoritmo é distinto para cada sinal senoidal. Além disto, o problema da resolução geométrica é facilmente observável. De fato, para a fonte $F3$, o atraso real teórico é de 0,2892 amostra para o ângulo de $88,42°$, o que forçaria um truncamento para 0 ou 1 amostra. Considerando este aspecto, podemos observar um funcionamento adequado do algoritmo nas seguintes situações:

- $s1$: correto para todos os tamanhos de janela fora do intervalo de 90 a 500

amostras.

- $s2$: correto só para tamanhos de janela de 3, 4 ou 5 amostras.
- $s3$: correto para tamanho de janela até 50 amostras.

Em particular, para o sinal $s1$, que é o sinal que gera o melhor comportamento, para um tamanho de janela pequeno, de 4 amostras, por exemplo, que também funciona para os demais sinais senoidais, a FCC é vista na Figura 3.6.

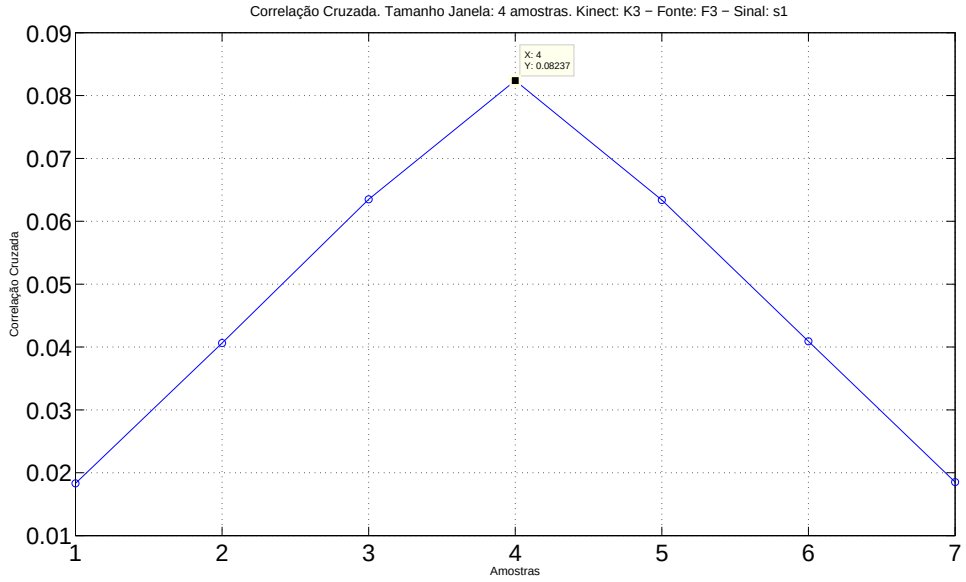

Figura 3.6: FCC para o sinal $s1$ e fonte na posição $F3$ e um tamanho de janela de 4 amostras.

Neste caso, a FCC possui apenas um pico bem no centro da função, gerando um atraso estimado de 0 amostra, como era de se esperar. Este resultado, para o caso do sinal $s1$, é qualitativamente estável até um tamanho da janela de 90 amostras. A FCC neste caso, para tamanhos de janela entre 70 e 100 amostras, é dada na Figura 3.7, na qual observamos um leve deslocamento do pico $\tau = 1$ amostra para a direita, o que causa o (pequeno) erro de estimativa na DoA. Ainda pela Figura 3.5 para o sinal $s1$, para tamanhos da janela acima de 500 amostras, o pico da FCC volta à posição $p = 0$ e a DoA estimada volta a ser a correta.

Agora vamos analisar o que acontece com o sinal $s3$. Observando a FCC do sinal $s3$ para tamanhos de janela iguais a 40, 50, 60 e 70 na Figura 3.8, notamos que esta função possui muitos mais picos do que a do sinal $s1$ devido à maior

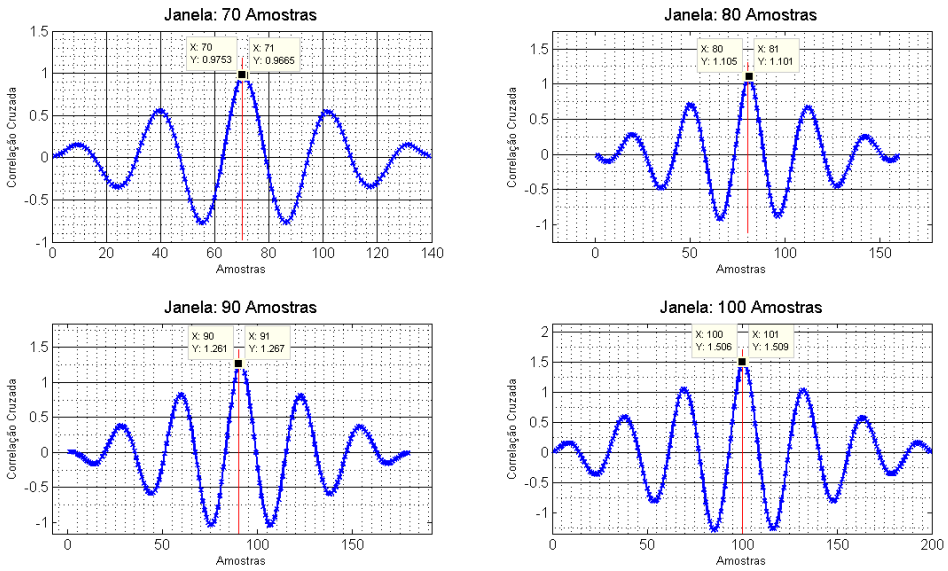

Figura 3.7: FCC para o sinal $s1$, fonte na posição $F3$ e diferentes tamanhos de janela: (a) 70; (b) 80; (c) 90; (d) 100 amostras.

frequência de $s3$. Isto faz com que o máximo da FCC do sinal $s3$ possa ocorrer, com maior probabilidade e frequência, para valores bastante distantes do ponto central, diferentemente do que ocorria para o sinal $s1$. Estes erros significativos, de várias amostras, no atraso estimado para o sinal $s3$ é a causa dos grandes erros de estimativa da DoA observados na Figura 3.5 para os sinais $s2$ e $s3$.

3.4.1.2 Posição Lateral da Fonte

Vamos considerar agora as estimativas de DoA com a fonte na posição lateral $F1$, à esquerda do Kinect. Neste caso, os resultados para os três sinais senoidais e para diferentes tamanhos de janela é visto na Figura 3.9.

Neste experimento, podemos ver que de modo geral as estimativas de DoA para os três sinais senoidais foram inadequadas. Exemplos da FCC para o sinal $s1$ e diferentes tamanhos de janela são vistos na Figura 3.10.

Para piorar, em diversos casos o atraso estimado foi acima do atraso máximo especificado, o que resultou na ausência de diversas estimativas na Figura 3.9. Para esta configuração do problema, o atraso máximo é igual a

$$p_{\max} = \frac{\delta_{1,4}.f_s}{c} = \frac{0{,}226 \times 16000}{343,2} = 10{,}53. \tag{3.6}$$

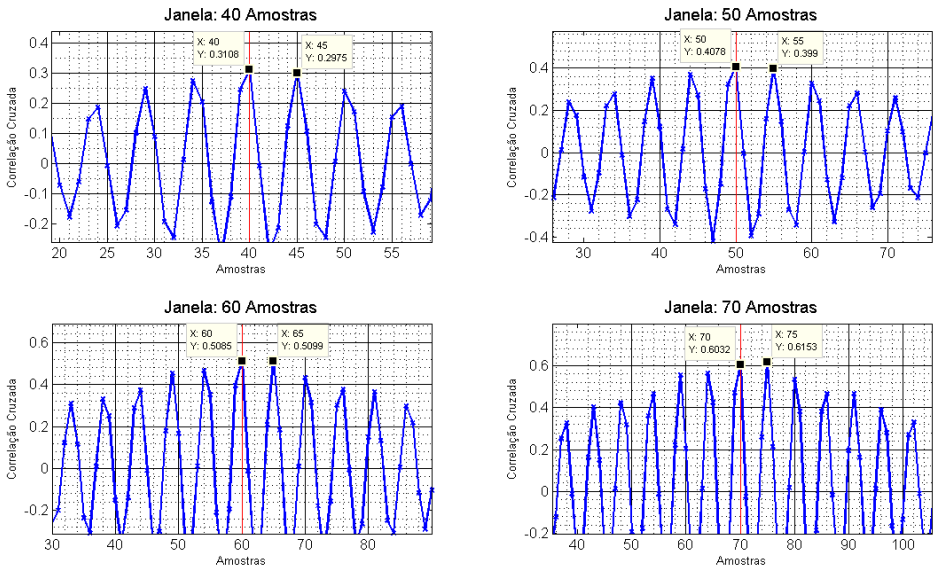

Figura 3.8: FCC para o sinal $s3$, fonte na posição $F3$ e diferentes tamanhos de janela: (a) 40; (b) 50; (c) 60; (d) 70 amostras.

Como vemos, o atraso calculado para 300 amostras é maior que o atraso máximo. De fato, para um tamanho de janela acima de 300 amostras o atraso estimado para o sinal $s1$ é grande demais, inviabilizando uma estimativa adequada da DoA.

3.4.2 Sinais de Fala

Como vimos na Seção 2.4, os sinais de fala de nossa base de dados s'ao pronunciados por locutores feminino ($s4$) e masculino ($s5$). Vale lembrar que o caso de sinais de fala no *far field* é justamente aquela situação para a qual o Kinect foi originalmente desenvolvido. Vamos apresentar os resultados para diferentes posições da fonte nas subseções a seguir.

3.4.2.1 Posição Central da Fonte

Os resultados do CCM para os dois sinais de fala de nossa base de dados quando a fonte está na posição $F1$ central são mostrados na Figura 3.11.

Efetivamente, por esta figura, podemos concluir que resultados para ambos os sinais de fala são bastante consistentes, a menos do efeito de resolução, que neste caso é de $90° - 88{,}43° = 1{,}57°$. O único intervalo para o qual não temos resultado mais exato é nas janelas de comprimento variando de 400 a 800 amostras para o

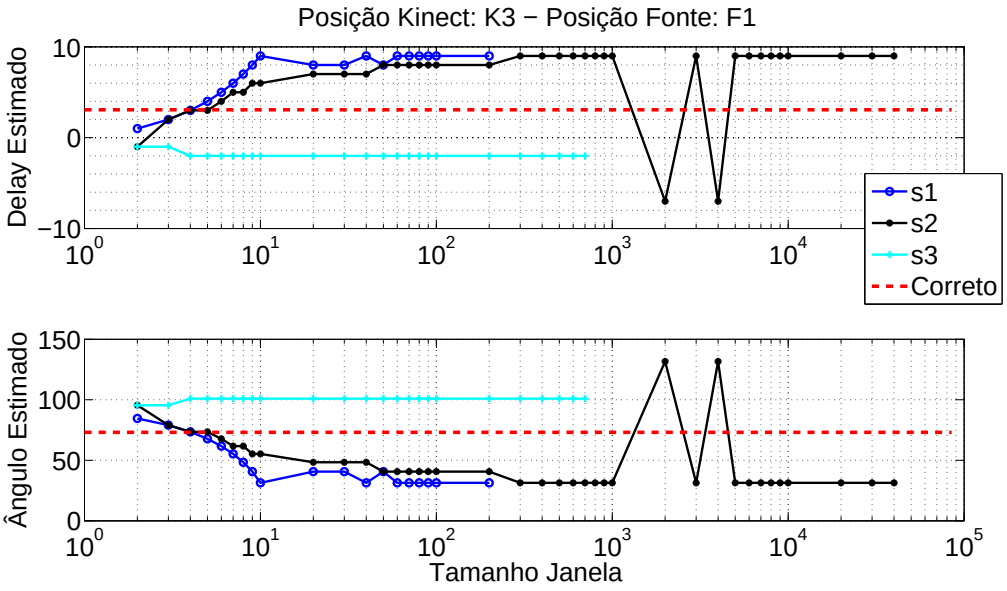

Figura 3.9: Resultados das medidas para sinais de banda estreita, para a fonte na posição $F1$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

sinal $s5$, onde temos um pequeno deslocamento do pico do sinal. Mesmo nestes casos, porém, o erro de estimativa é bastante razoável.

3.4.2.2 Posição Lateral da Fonte

Colocando a fonte de sinal de fala na posição lateral $F1$, os resultados do CCM são mostrados na Figura 3.12.

Para o sinal $s4$, primeiro temos erros devidos ao pequeno tamanho da janela. Depois, para comprimentos da janela acima de 1300 amostras, temos erros por deslocamentos do pico. De modo geral, porém, temos resultados corretos para uma ampla gama de comprimentos. Para o sinal $s5$ todos os erros que temos são de deslocamento de várias amostras do pico correto, o que leva a erros significativos de estimativa.

3.4.3 Sinais de Chaleira

3.4.3.1 Posição Central da Fonte

O último tipo de sinal que temos, como colocado na Seção 2.4, é o do som gerado por uma chaleira durante a fervura da água. Começando novamente pela

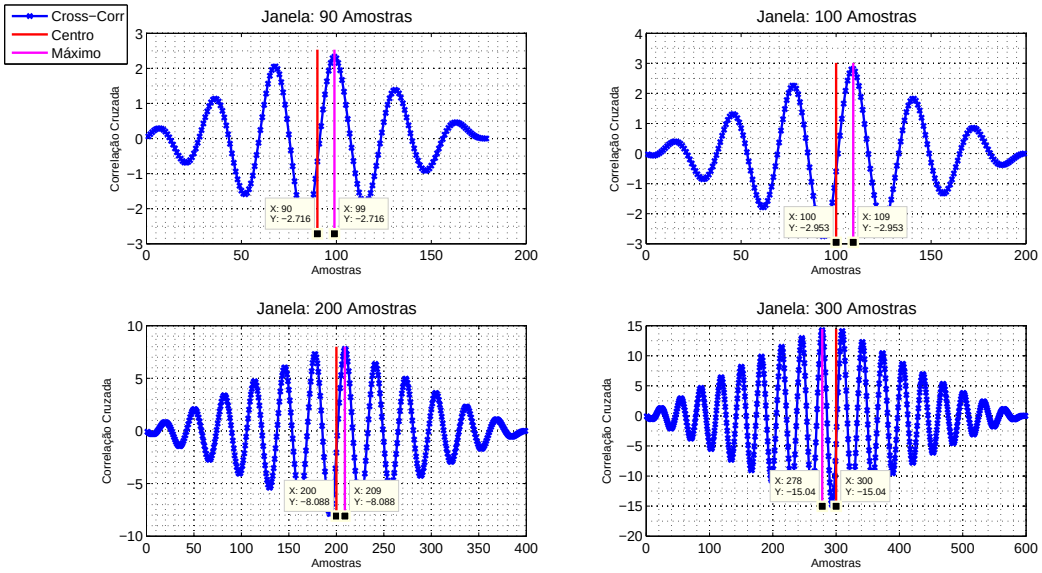

Figura 3.10: FCC para o sinal $s1$, fonte na posição $F1$ e diferentes tamanhos de janela: (a) 90; (b) 100; (c) 200; (d) 300 amostras.

posição central $F3$ da fonte, obtemos as estimativas indicadas na Figura 3.13.

Por esta figura, podemos facilmente observar que neste caso temos resultados bastante consistentes, obtendo erros de estimativa desprezíveis.

3.4.3.2 Posição Lateral da Fonte

Se movemos a fonte para a posição lateral $F1$ acontece o mesmo resultado dos casos anteriores: há uma piora dos resultados estimados. Na grande maioria das vezes, porém, a piora equivale a um erro de 1 amostra na estimativa do atraso. Em alguns casos específicos, porém, para janelas de 200, 10000 e 20000 amostras, a estimativa de atraso é bastante inconsistente.

3.5 Experimento: Sinais no *Far-Field* Com Ruído

Como mencionado na Seção 2.4, para cada um dos sinais temos gravado uma versão sem ruído e outra com ruído. Na Seção 3.4 realizamos o experimento com os sinais sem ruído. Na presente seção, consideramos a mesma análise para os sinais com o ruído gerado por uma máquina de ar-condicionado.

Nas Figuras 3.14 e 3.15 podemos ver os resultados para a posição central $F3$

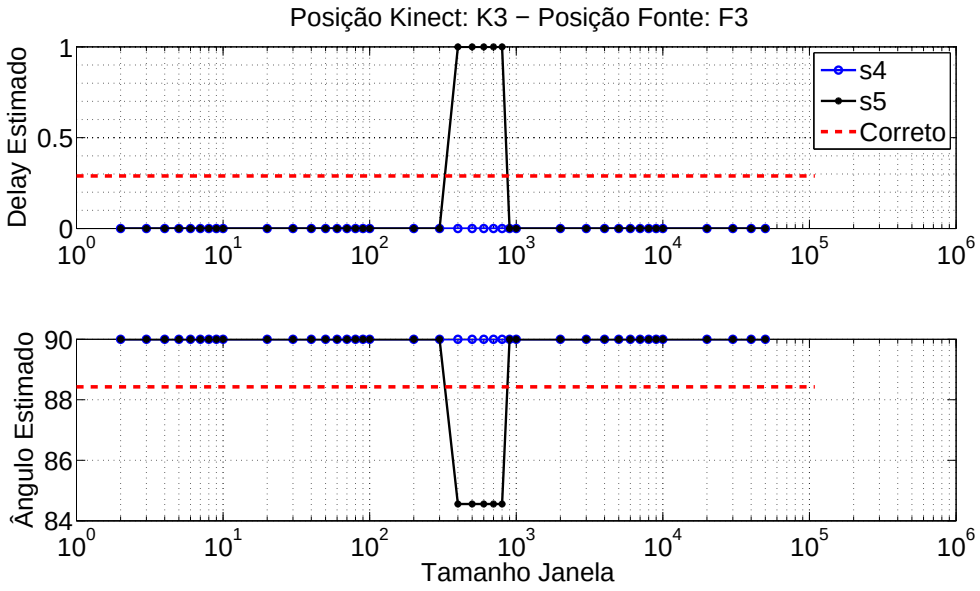

Figura 3.11: Resultados das medidas para sinais de fala, para a fonte na posição $F3$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

da fonte e para os diferentes tipos de sinais com ruído. À primeira vista, é fácil ver que os resultados são piores que os resultados sem ruído, como era de se esperar. De modo geral, o único sinal que se mostrou suficientemente robusto, mesmo na presença do ruído, foi o sinal gerado pela chaleira.

3.6 Experimento: Sinais no *Near-Field*

Nesta seção, vamos considerar a posição $K1$ do Kinect, que corresponde a um problema de DoA no *near field*. Nessas análises, vamos considerar apenas os sinais de fala e da chaleira que, por possuírem maior riqueza espectral, geram melhores resultados da estimação de DoA, como visto na seção anterior. Sendo assim, os resultados de estimação para os sinais $s4$, $s5$ e $s6$ quando a fonte se encontra nas posições $F1$ e $F3$ podem ser vistos nas Figuras 3.16 e 3.17, respectivamente.

Nesta situação de *near-field*, qualitativamente, temos os mesmos resultados obtidos para o *far-field*, isto é: a posição central gerou resultados mais consistentes, para diferentes tamanhos da janela de segmentação, particularmente para o sinal da chaleira. De modo geral, porém, os resultados de *near-field* são ligeiramente mais erráticos que os de *far-field*, como era de se esperar.

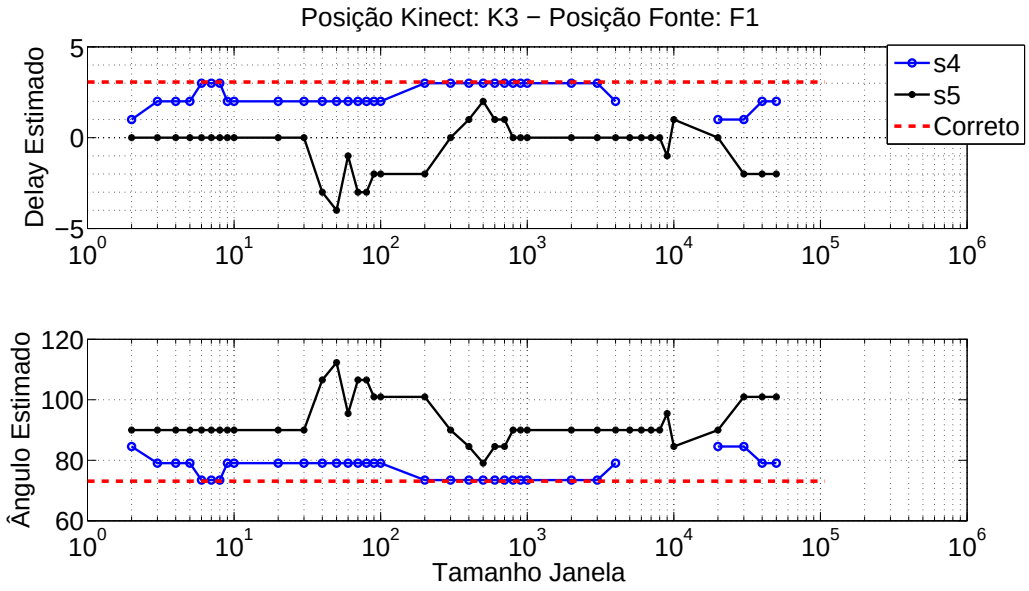

Figura 3.12: Resultados das medidas para sinais de fala, para a fonte na posição $F1$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

3.7 Experimento: Fonte em Movimento sem Ruído

Até agora, os experimentos que estamos fazendo são com a fonte de som estática, sem movimento. Mas na vida real normalmente isto não acontece, e a fonte - geralmente, no caso do sistema Kinect da Microsoft, uma pessoa - tem movimento ao mesmo tempo que faz sons. Portanto, nesta seção, analisamos o comportamento do algoritmo de correlação cruzada para uma fonte que não fica estática. Nestas análises, consideramos que a velocidade da fonte é suficiente baixa para ignorarmos o efeito Doppler.

Conforme descrito na Seção 2.4, nossa base de dados inclui uma situação em que a fonte se move entre as posições $F2$ e $F4$ com o Kinect posicionado na posição $K3$ do *far-field*. Para este caso, foram feitas as gravações dos sinais $s1$ para banda estreita, $s4$ de fala e $s6$ da chaleira. Além disto, nas análises aqui realizadas, realizamos as seguintes considerações de ordem prática:

- Primeiro temos que escolher um tamanho de janela concreto para a análise.
- Dividimos o sinal que vamos analisar em pedaços do tamanho de janela escolhido e executamos o algoritmo para cada um dos pedaços.
- Comparamos cada resultado com a posição real da fonte, ao longo do tempo.

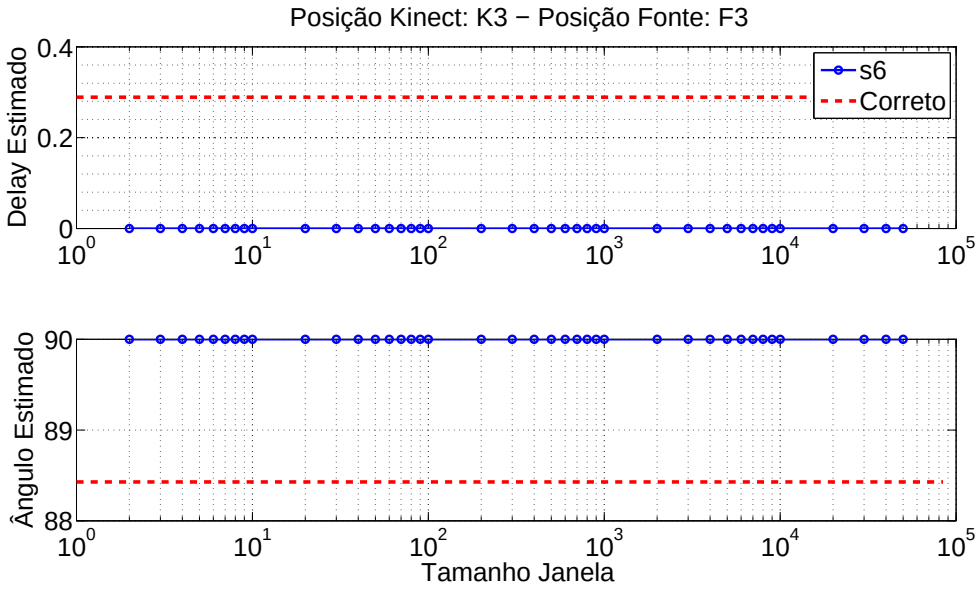

Figura 3.13: Resultados das medidas para o sinal da chaleira, para a fonte na posição $F3$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

Os resultados para os diferentes tipos de sinais são apresentados nas subseções a seguir.

3.7.1 Sinal de Banda Estreita

O resultado da estimativa de DoA para o sinal de banda estreita se movendo "lentamente" em relação ao Kinect é visto na Figura 3.18 para diferentes tamanhos de janela de segmentação. De modo geral, pelos resultados desta figura, conclui-se que uma janela de 50 amostras representa um bom compromisso de erro de estimação e complexidade computacional (janelas menores geram mais segmentos que, por sua vez, correspondem a um maior número de estimativas ao longo do tempo).

Os resultados de estimação para o caso da fonte se movendo mais rapidamente, num percurso de ida-e-volta entre as posições $F2$ a $F4$, no mesmo espaço de tempo total que o caso mais lento, são vistos na Figura 3.19.

Esta nova situação apresentou maiores dificuldades para o algoritmo, principalmente no retorno do sistema. Qualitativamente, porém, as menores janelas (com 50 ou 11 amostras) apresentaram os melhores resultados.

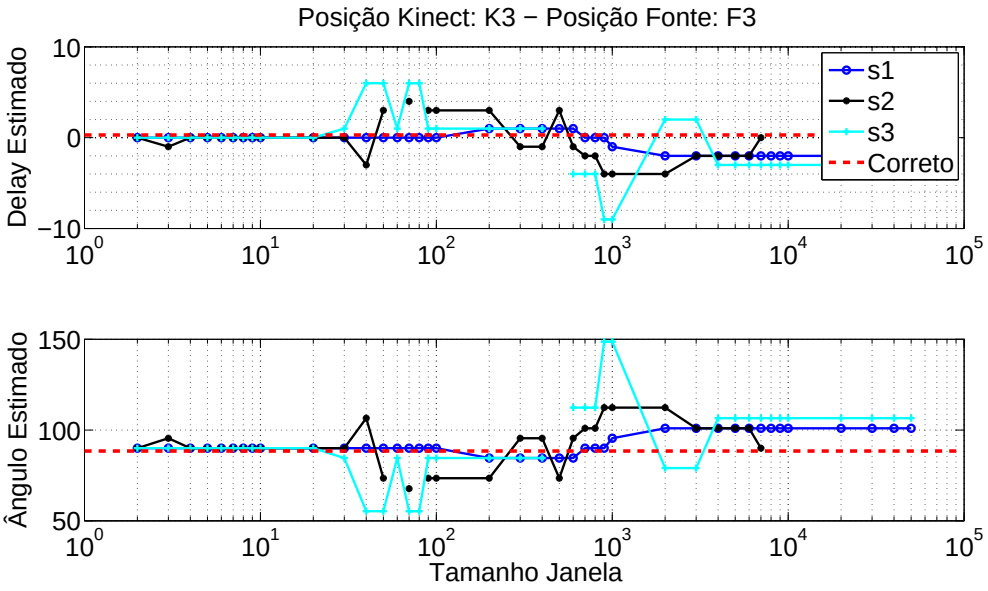

Figura 3.14: Resultados das medidas para os sinais senoidais com ruído, para a fonte na posição $F3$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

3.7.2 Sinal de Fala

Quando se usa o sinal de fala se movendo, os resultados das Figuras 3.20 (velocidade mais baixa) e 3.21 (velocidade maior) indicam um melhor resultado para janelas mais longas, com cerca de 500 amostras, similar ao que que foi observado na Figura 3.11 para a fonte estática.

Efetivamente, aqui não precisamos ter uma janela muito pequena para ter resultados parecidos aos resultados corretos. Para todos os tamanhos de janela os resultados da linha de tendência comparada com a linha do ângulo real são parecidos, menos para o maior tamanho, que não consegue acompanhar direito a fonte em movimento.

3.7.3 Sinal da Chaleira

Os resultados de estimação de DoA quando a fonte se locomovendo reproduz o sinal da chaleira, vistos nas Figuras 3.22 e 3.23, indicam, novamente, um comportamento muito mais eficaz do algoritmo CCM, certamente devido à maior riqueza espectral do sinal em questão.

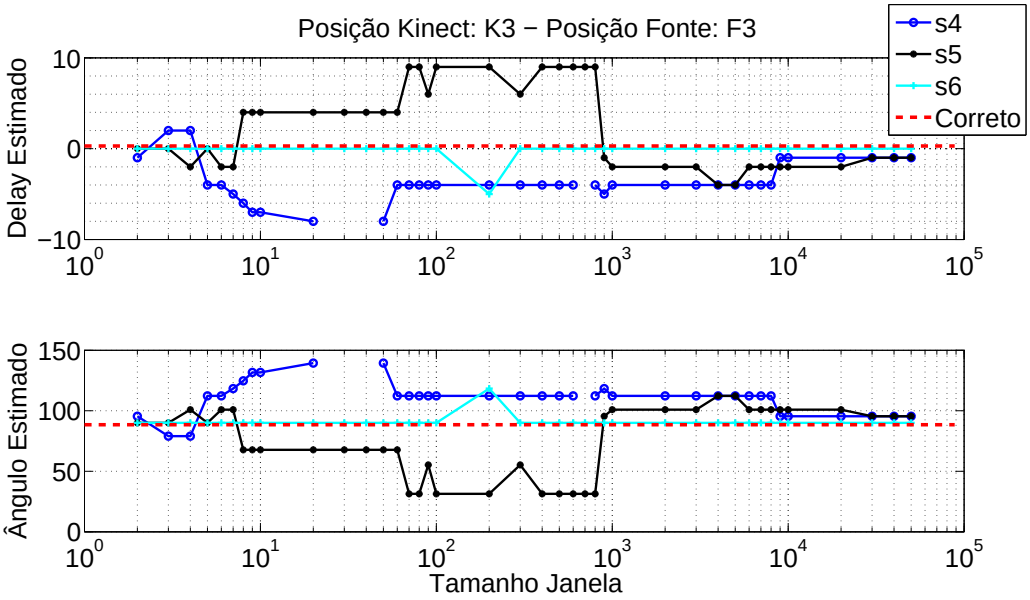

Figura 3.15: Resultados das medidas para os sinais de fala e da chaleira com ruído, para a fonte na posição $F3$, o Kinect na posição $K3$ do *far-field*: (a) atraso; (b) ângulo de chegada.

De fato, na presente situação, o uso de janelas maiores gera resultados suficientemente confiáveis para uma grande gama de situações práticas.

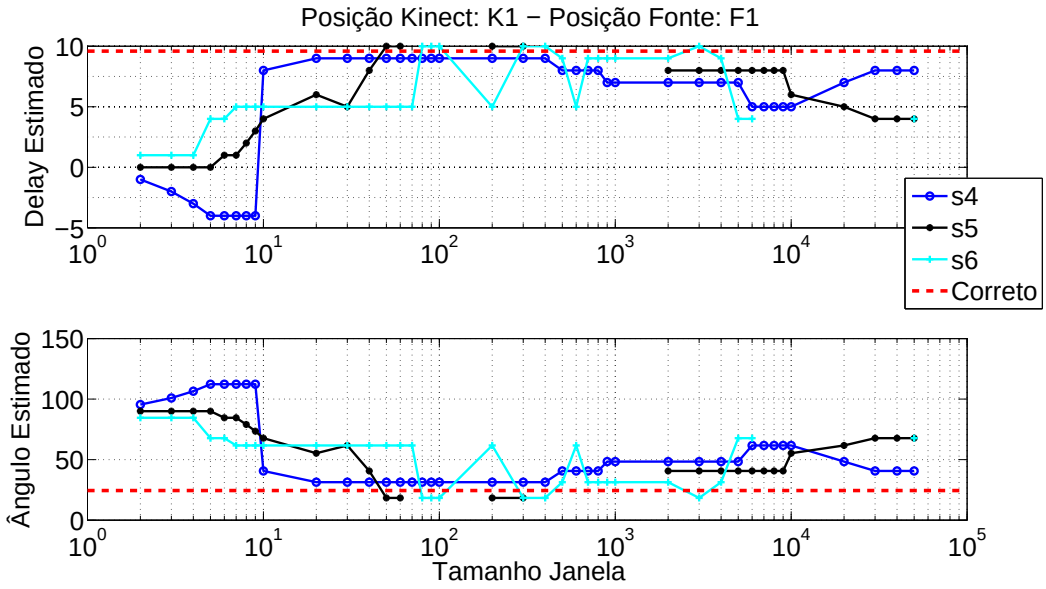

Figura 3.16: Resultados das medidas para os sinais de fala e da chaleira, para a fonte na posição lateral $F1$, o Kinect na posição $K1$ do *near-field*: (a) atraso; (b) ângulo de chegada.

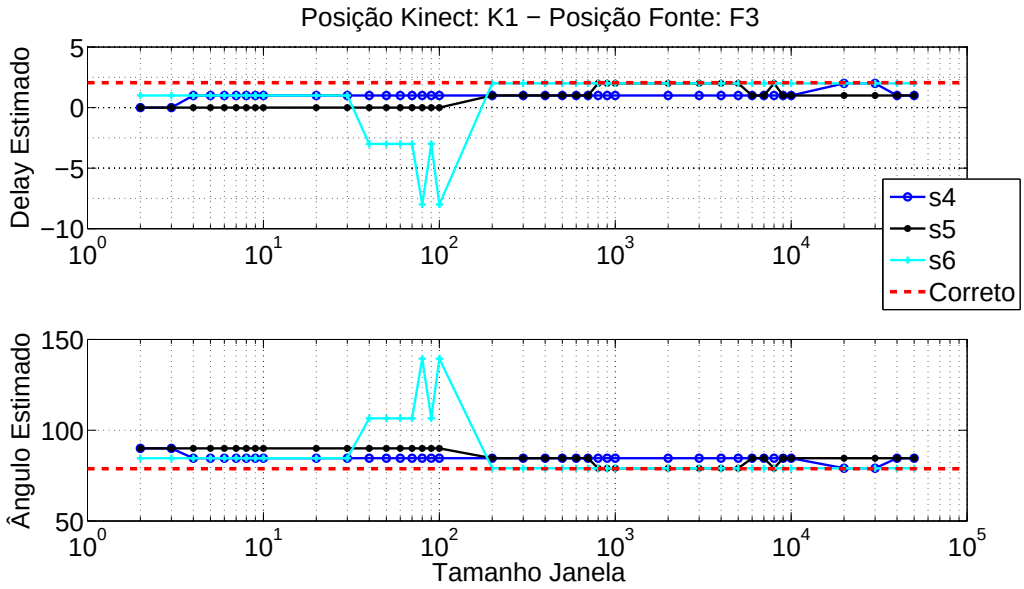

Figura 3.17: Resultados das medidas para os sinais de fala e da chaleira, para a fonte na posição central $F3$, o Kinect na posição $K1$ do *near-field*: (a) atraso; (b) ângulo de chegada.

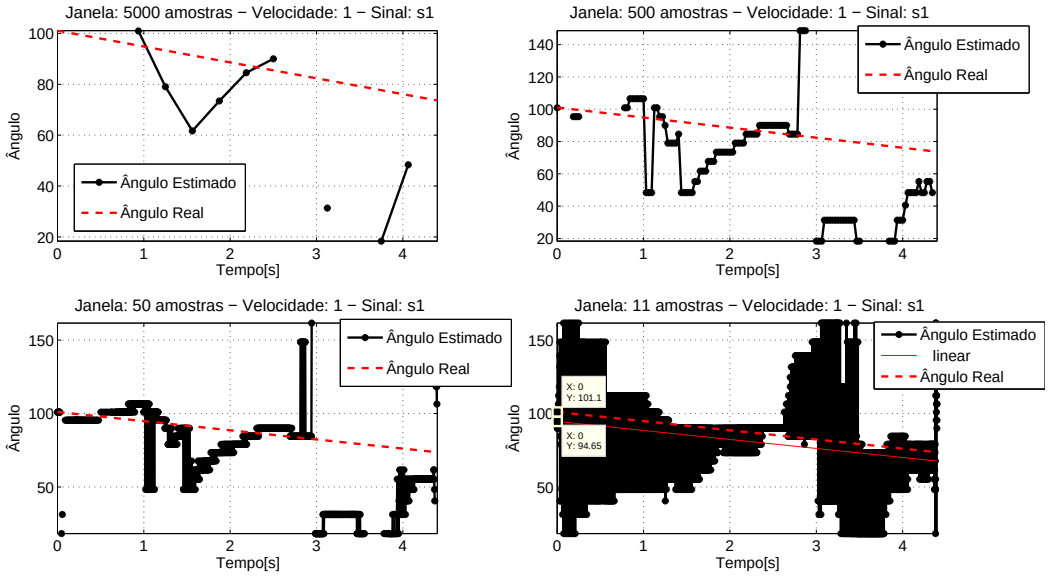

Figura 3.18: Resultado do ângulo estimado para o sinal $s1$ e a fonte em movimento com velocidade baixa.

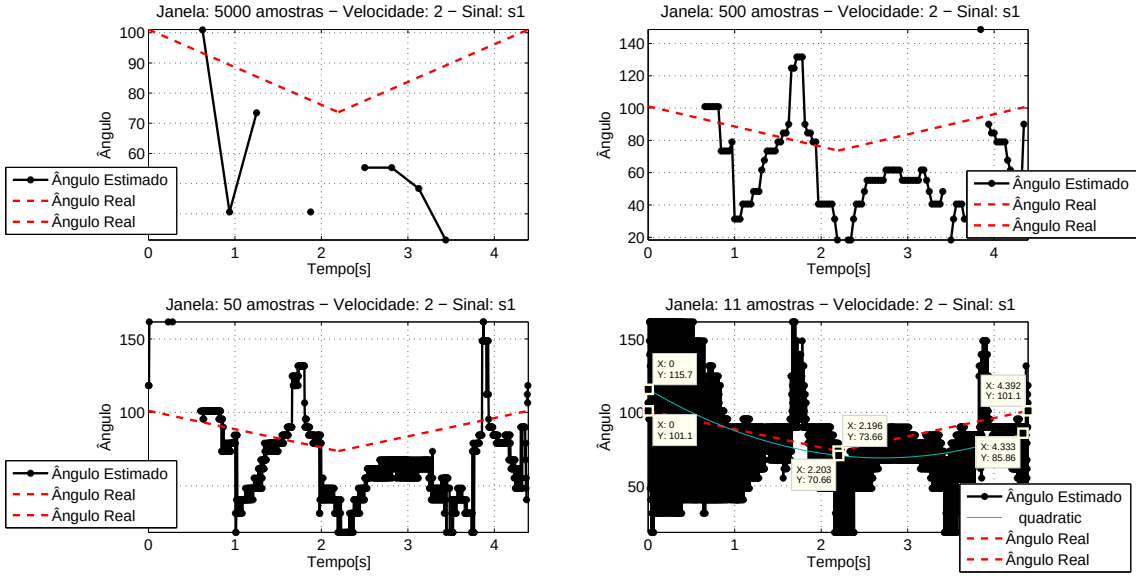

Figura 3.19: Resultado do ângulo estimado para o sinal $s1$ e a fonte em movimento com velocidade alta.

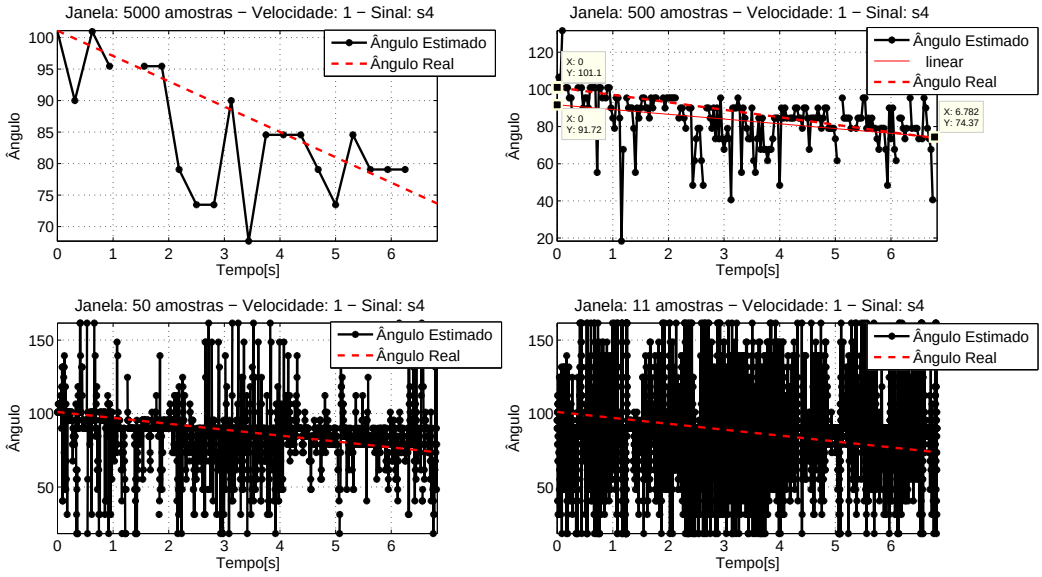

Figura 3.20: Resultado do ângulo estimado para o sinal $s4$ e a fonte em movimento com velocidade baixa.

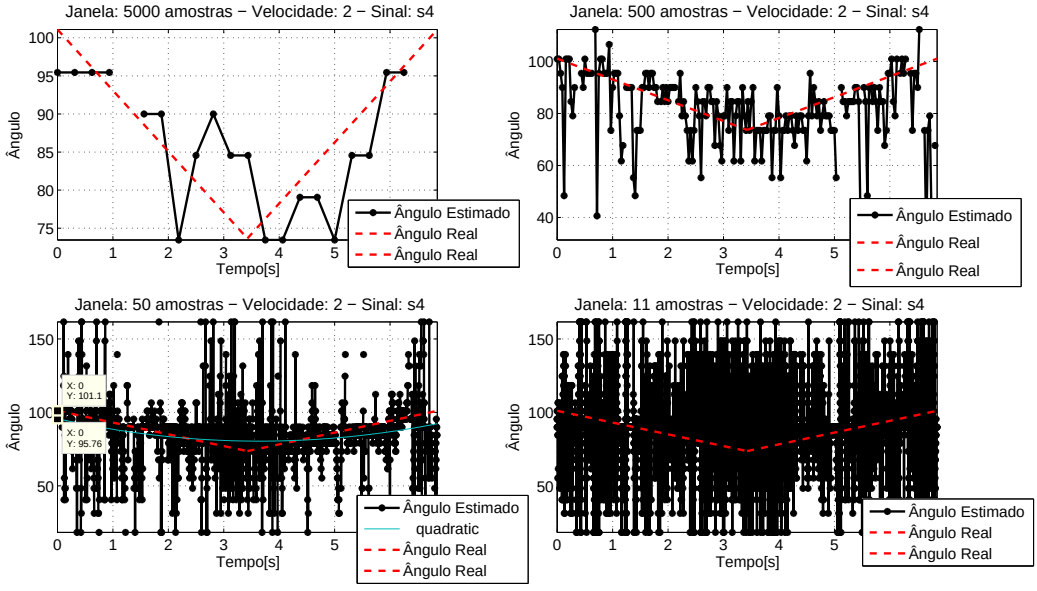

Figura 3.21: Resultado do ângulo estimado para o sinal $s4$ e a fonte em movimento com velocidade alta.

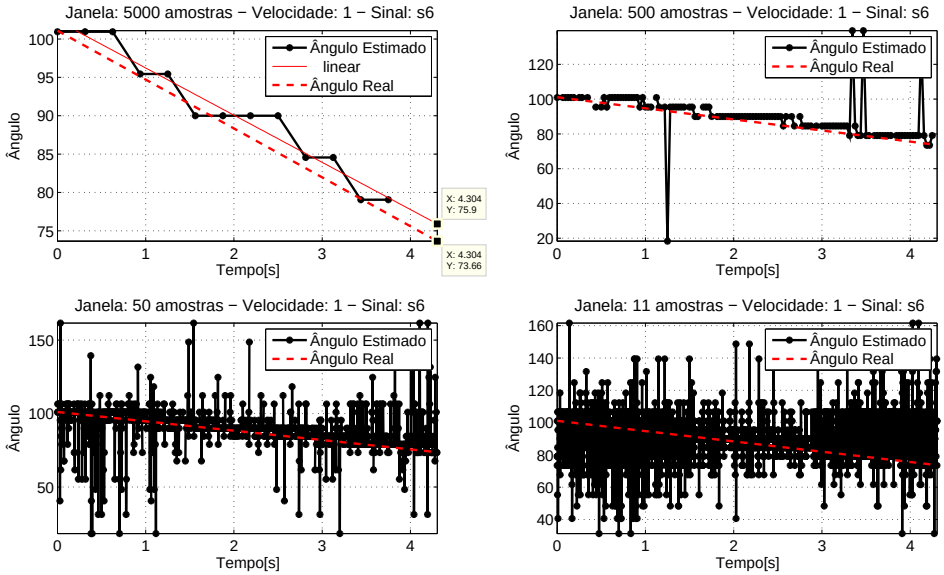

Figura 3.22: Resultado do ângulo estimado para o sinal $s6$ a fonte em movimento com velocidade baixa.

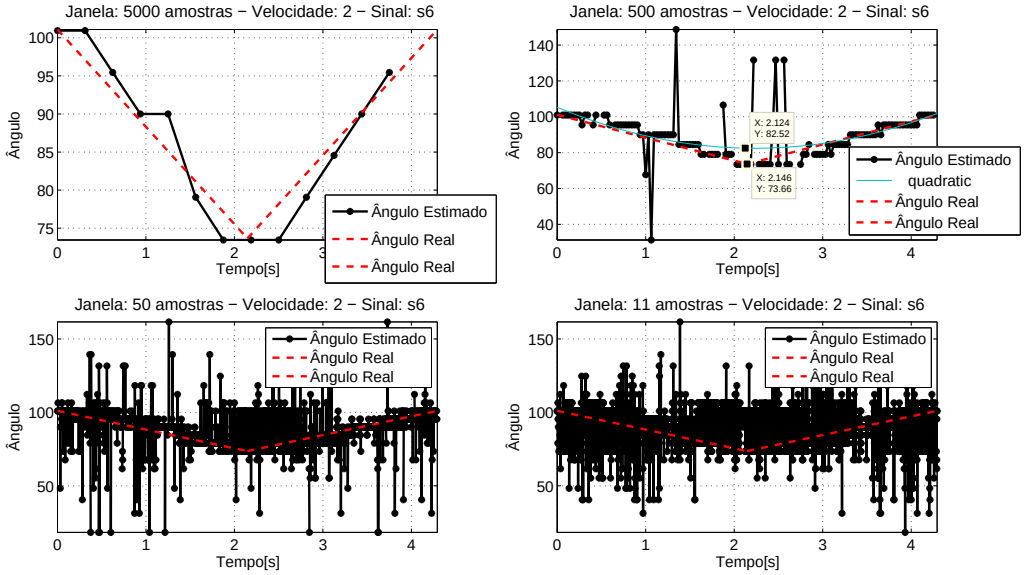

Figura 3.23: Resultado do ângulo estimado para o sinal $s6$ e a fonte em movimento com velocidade alta.

3.8 Conclusão

Neste Capítulo 3 estudamos a eficácia do CCM para a estimativa do TDOA e a determinação do ângulo da fonte para os diferentes cenários descritos no Capítulo 2.

Nossa primeira análise foi para o *far-field*. Para este cenário vimos que temos, além do erro da resolução finita, dois erros possíveis mais: o erro de deslocamento do pico, que acontece para tamanhos pequenos de janela, e o erro de ter outro pico maior que o pico correto, que acontece com janelas grandes e que é maior que o erro anterior.

Vimos que a estimativa piora se a fonte fica na posição lateral, e é melhor se fica perto da posição frontal em relação ao arranjo de microfones. Além disto, dependendo do tipo de sinal, os resultados são piores para sinais de banda estreita, um pouco melhores para os sinais de fala, e muito melhores para o sinal da chaleira, provavelmente devido à sua riqueza espectral. Também vimos que, se temos ruído, a estimativa é muito ruim, e este algoritmo não funciona nesse caso.

Para o *near-field* os resultados tem as mesmas características que os resultados para o *far-field*, mas em geral são piores, provavelmente devido à frente de onda não ser suficientemente plana.

Finalmente, para a análise com movimento, vimos que a qualidade dos resultados depende do tamanho da janela ao longo do tempo. Para os sinais que tinham piores resultados sem movimento - banda estreita - vamos precisar de uma janela menor - que requer mais tempo e recursos de computação - que para a chaleira. Também vemos que quanto maior é a velocidade do movimento e o ruído, piores são os resultados.

No próximo Capítulo 4 vamos ver como podemos melhorar os resultados obtidos, fazendo pequenas melhoras no CCM e tentando implementar alguns prefiltros dentro da família dos métodos da correlação cruzada generalizada.

Capítulo 4

Outros Algoritmos

4.1 Introdução

No Capítulo 3, usamos o método de correlação cruzada (CCM) clássico para determinar a estimativa de DoA, observando uma série de possíveis erros - de resolução, de deslocamento e outros - nos resultados gerados por tal algoritmo.

No presente capítulo, para tentar melhorar o método clássico, discutimos algumas variações do algoritmo CCM e ampliamos o nosso conhecimento acerca do problema de localização de fonte com outros métodos da família da correlação cruzada generalizada.

Sendo assim, na Seção 4.2 discutimos três variações do CCM e analisamos as respectivas melhorias. Posteriormente, na Seção 4.3, apresentamos dois prefiltros para o cálculo da correlação cruzada: o prefiltro *smoothed coherence transform* (SCOT) - cujo desempenho é apresentado na Seção 4.3.1 - e o prefiltro de *phase transformation* (PHAT) - cujo desempenho é analisado na Seção 4.3.2. Finalmente, na Seção 4.4 sintetizamos os principais resultados obtidos e tiramos as conclusões gerais do capítulo.

4.2 Variações do CCM

Nesta Seção 4.2, consideramos algumas modificações no CCM numa tentativa de melhorar os resultados descritos no Capítulo 3.

4.2.1 Variação 1: Busca Limitada

Como colocado na Seção 3.3, existe um valor de atraso máximo possível para o arranjo de microfones utilizado. No Capítulo 3, quando tínhamos um atraso maior (determinado pelo valor de pico da FCC) que o atraso máximo permitido, por convenção nossa, o algoritmo CCM retornava um valor "NaN" (não numérico) para o tamanho de janela em questão. Esta situação gerava ausência de estimativas nos gráficos de atraso e ângulo.

Nesta seção, buscamos o valor máximo da FCC forçosamente dentro do intervalo determinado pelo valor máximo permitido, que no nosso sistema é dado por

$$\frac{\delta_{1,4} f_s}{c} = 10,536 \text{ amostras} \tag{4.1}$$

onde $\delta_{1,4}$ é a distância entre os sensores 1 e 4, f_s é a frequência de amostragem e c é a velocidade do som. Truncando este valor para o número inteiro mais próximo, o novo algoritmo CCM considera o valor máximo da FCC apenas dentro do intervalo definido por [tamanho janela $-$ 10, tamanho janela + 10].

Desta forma, as estimativas de atraso e DoA obtidas para as posições $F1$ (lateral) da fonte e $K3$ (*far field*) do Kinect, considerando ou não, para efeito de comparação, esta simples modificação, são mostradas na Figura 4.1.

Desta figura, porém, observamos que todos os tamanhos de janela geram algum tipo de estimativa, ao contrário do que ocorria anteriormente. Em geral, porém, as novas estimativas apresentam desvios significativos em relação ao valor teórico, dentro do que já ocorria anteriormente.

A FCC para uma janela de 300 amostras do sinal cossenoidal $s1$ é vista na Figura 4.2, onde os limites de busca do valor máximo são denotados pelas linhas verticais tracejadas. Neste caso, com a nova modificação o algoritmo determina o atraso estimado na posição 309, que corresponde a um atraso efetivo, após subtrair o comprimento original da janela, de 9 amostras. Este valor não é o valor esperado, mas é melhor que não ter nenhuma estimativa.

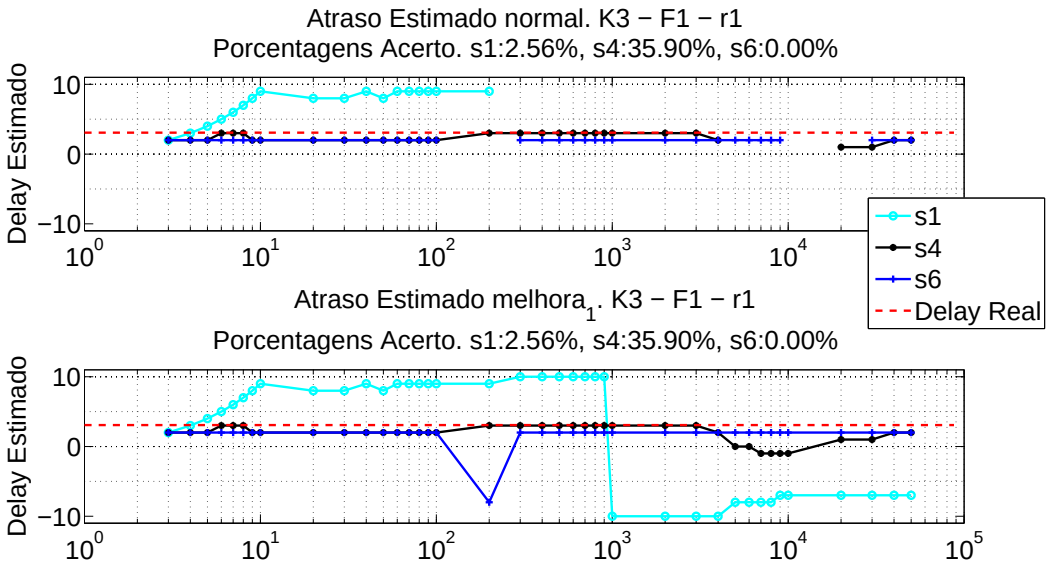

Figura 4.1: Estimativas de atraso e de DoA para fonte lateral $F1$ e Kinect distante $K3$, incorporando (abaixo) ou não (acima) a primeira modificação do algoritmo CCM.

4.2.2 Variação 2: Correção do Pico

Até agora, estimamos o atraso entre os dois sinais pelo valor de pico da FCC dentro de um certo intervalo previamente delimitado. Observando, porém, as diferentes FCCs mostradas ao longo de todo o Capítulo 3, observamos que na maioria das vezes o atraso correto é dado pelo primeiro pico da FCC, o qual nem sempre assume o valor máximo efetivo desta função. A partir desta observação, numa segunda modificação do CCM, optamos tomar como estimativa de atraso a posição do primeiro pico da FCC a partir do atraso nulo. Numa primeira aproximação, vamos definir um pico como um ponto da FCC com valor maior que os pontos imediatamente anterior e posterior. Neste sentido, desenvolvemos a seguinte metodologia de busca do atraso:

1. Averiguamos onde ficam todos os picos dentro do intervalo de valores possíveis que definimos na seção anterior.

2. Vemos se no centro temos pico. Se temos, escolhemos esse pico e temos atraso nulo.

3. Se não temos pico no centro, escolhemos o pico que mais próximo deste centro.

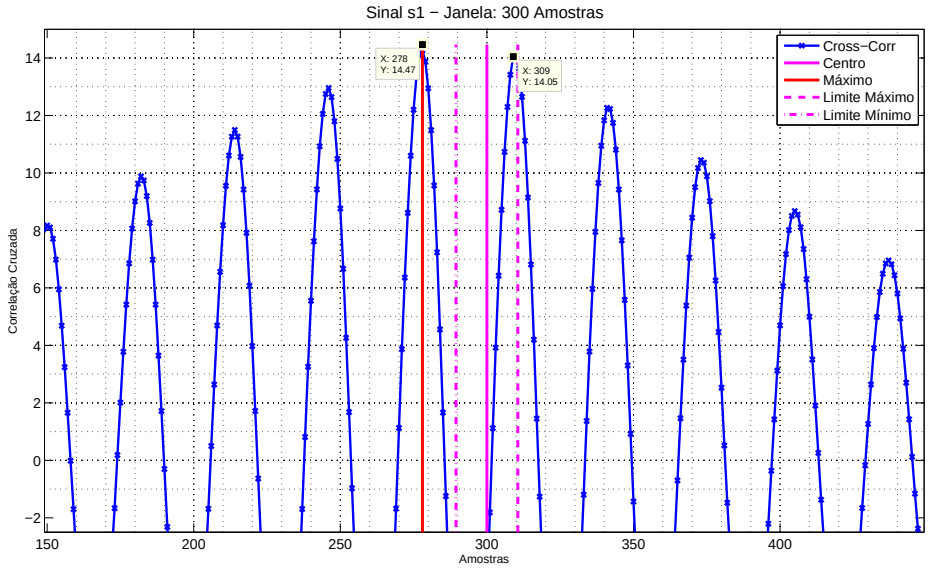

Figura 4.2: FCC para o sinal $s1$ no *far field* e posição $F1$ de fonte lateral e um tamanho de janela de 300 amostras.

Se temos dois picos com a mesma distancia para o centro (um à esquerda e outro à direita), escolhemos o que tem valor mais alto.

4. Se não temos pico no intervalo previamente determinado, pegamos o valor mais alto do intervalo (o mesmo que na variação 1).

Os resultados desta nova versão do CCM são apresentados nos itens a seguir para as diferentes situações contempladas em nossa base de dados.

4.2.2.1 *Far Field*

Neste item, consideramos a posição de *far field* do Kinect e os sinais com ruído, que haviam apresentado maiores dificuldades de estimação anteriormente. Para esta configuração as estimativas do CCM com e sem esta segunda modificação são mostradas na Figura 4.3 para os sinais $s1$ (senoidal), $s4$ (fala) e $s6$ (chaleira).

Desta figura, podemos perceber um novo erro médio para o sinal $s4$, metade do anterior, e uma correção na estimativa obtida para o sinal $s6$ com a janela de 200 amostras.

Para melhor entendermos o funcionamento desta segunda modificação, considere a estimativa de atraso do sinal da chaleira $s6$ com uma janela de 200 amostras,

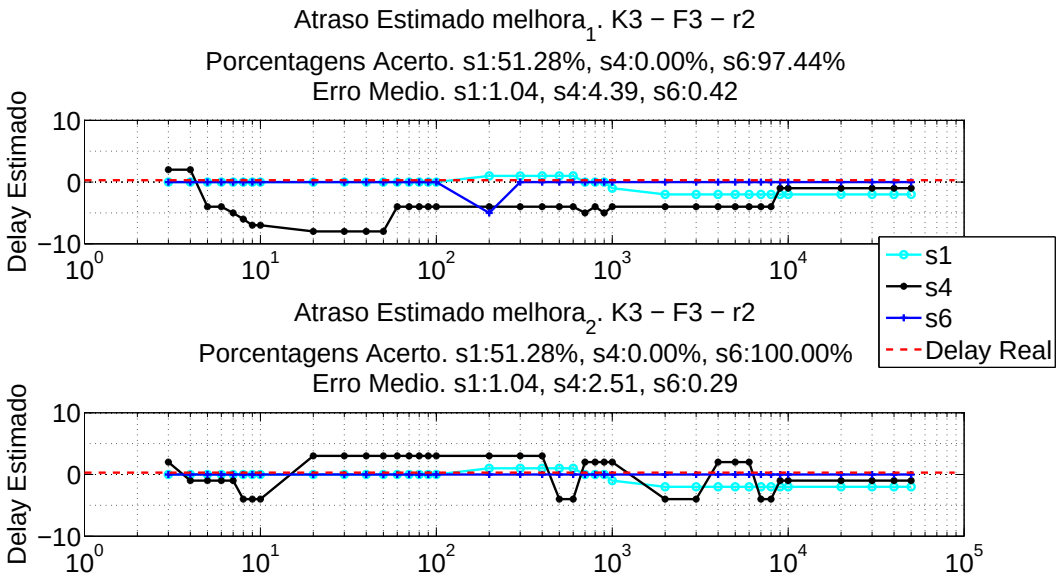

Figura 4.3: Estimativas de atraso e de DoA para fonte central $F3$ e Kinect próximo $K1$, incorporando (abaixo) ou não (acima) a segunda modificação do algoritmo CCM.

conforme indicada na Figura 4.3. Nesta figura, observamos que o máximo da função fica dentro do intervalo de valores possíveis, porém deslocado da posição central, como ilustrado na Figura 4.4, possivelmente devido ao ruído presente nos sinais. Com a segunda variação, porém, ignoramos este valor máximo e obtemos diretamente o atraso correto, mais próximo à posição central da janela.

Para o sinal $s4$ de fala com ruído, porém, a segunda modificação não foi tão efetiva assim, apesar de apresentar melhoras no comportamento geral da nova versão do algoritmo. Nesse caso, a FCC para uma janela de 50 amostras é mostrada na Figura 4.5, na qual observamos que a nova modificação é capaz de reduzir o erro de estimativa sem, porém, zerá-lo como seria desejado.

Uma síntese do desempenho do algoritmo CCM com as duas modificações, para as posições central e lateral da fonte, com o Kinect no *far field* é dada na Tabela 4.1, onde a média é calculada para todos os tamanhos de janela considerados na Figura 4.3.

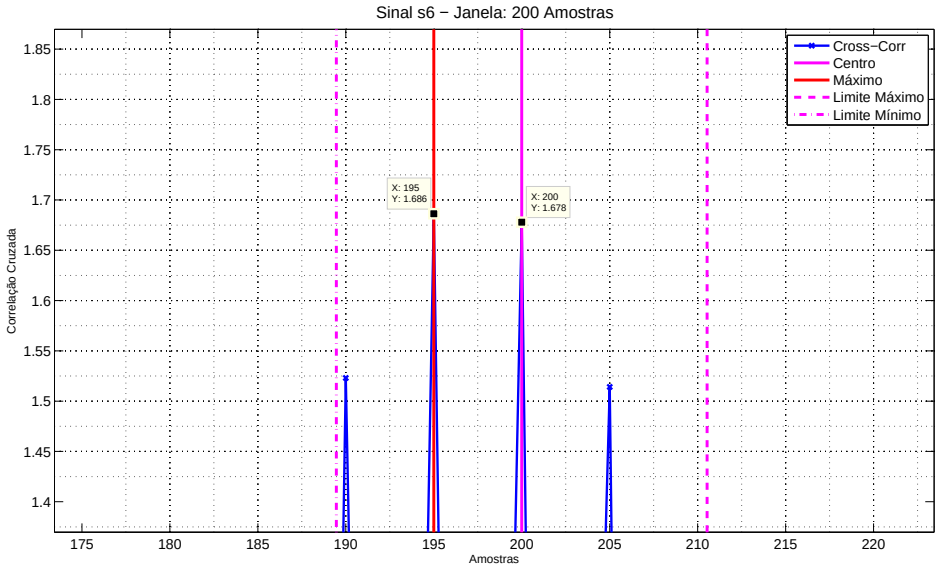

Figura 4.4: FCC para o sinal $s6$ com ruído no *far field* e posição $F3$ de fonte central e um tamanho de janela de 300 amostras.

4.2.2.2 *Near-Field*

Uma síntese do desempenho do algoritmo CCM com as duas modificações, para as posições central e lateral da fonte, com o Kinect no *near field* é dada na Tabela 4.2. Neste caso, porém, de *near field*, a segunda modificação causa uma piora sistemática de desempenho do algoritmo CCM para a posição lateral da fonte.

Concluímos desta tabela que para o *near field* é mais comum que o pico certo não seja o primeiro. Por exemplo, para o sinal $s4$, com um tamanho de janela de 80 amostras e a fonte na lateral, podemos ver a sua FCC na Figura 4.6, cujos valores de atraso para as diferentes versões do CCM são:

- Atraso real: 9,59 amostras.
- Resultado Variação 1: 9 amostras.
- Resultado Variação 2: -6 amostras.

4.2.3 Variação 3: Nova Correção de Pico

Depois de observarmos na Seção 4.2.2 o mal funcionamento da Variação 2 no *near field*, vamos tentar corrigir este erro de outro jeito. Pela Figura 4.6, temos que

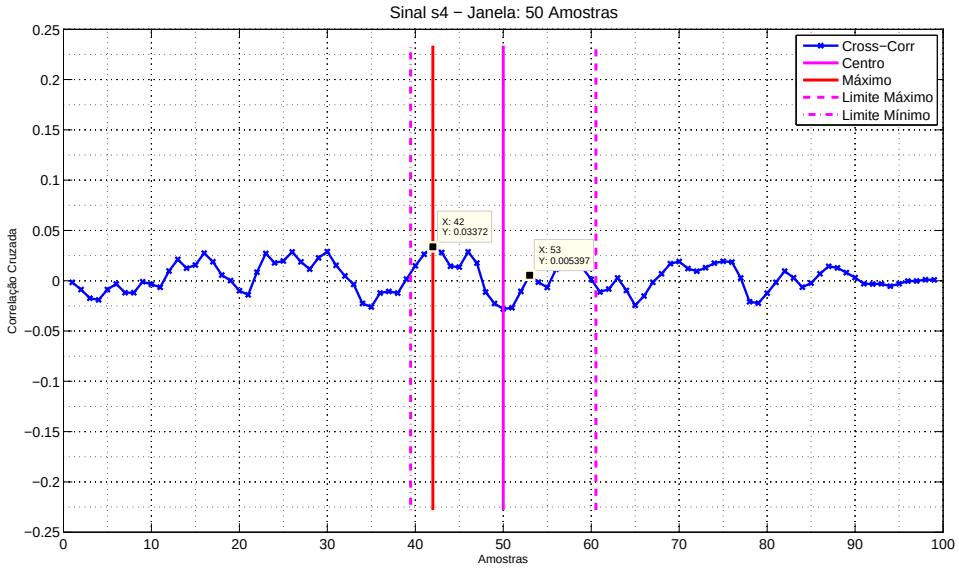

Figura 4.5: FCC para o sinal $s4$ com ruído no *far field* e posição $F3$ de fonte central e um tamanho de janela de 50 amostras.

o problema é que há um pico da FCC mais próximo do centro que o pico principal, o que nos leva à estimativa incorreta do pico. Para tentar solucionar este problema, vamos mudar o método da Variação 2 com estes passos:

1. Averiguamos onde ficam todos os picos dentro do intervalo de valores possíveis.
2. Vemos o pico mas perto do centro que fica na esquerda e o pico mais perto do centro que fica na direita.
3. Escolhemos o mais alto dos dois picos obtidos no passo anterior.

Esta metodologia seria capaz de corrigir o erro ilustrado na Figura 4.6. Se fazemos o análise para todas as janelas, os resultados que temos são que para o *near field* com esta terceira modificação são sumarizados na Tabela 4.3, que indica uma pequena melhora para a posição central da fonte no caso do sinal $s4$.

4.2.4 Análise com Movimento

Neste item, consideramos os sinais da nossa base em que a fonte sonora está em movimento em relação ao Kinect. Para esta análise, vamos considerar dois tamanhos de janela: 50 e 500 amostras, que geram os melhores resultados para os três tipos de sinal. Serão ainda considerados os seguintes casos:

Tabela 4.1: Valores do erro médio, em graus, da Variação 1 e a Variação 2 no *far field*, para todos os tamanhos de janela.

		Sem Ruído		**Com Ruído**	
		Centro	***Lateral***	***Centro***	***Lateral***
Sinal s1	***Variação 1***	1,93	48,3	5,65	43,60
	Variação 2	1,93	41,80	5,65	44,37
Sinal s4	***Variação 1***	1,57	6,5	25,1	30,83
	Variação 2	1,57	6,50	13,84	29,01
Sinal s6	***Variação 1***	1,57	7,51	2,30	10,26
	Variação 2	1,57	5,97	1,57	5,97

Tabela 4.2: Valores do erro médio, em graus, da Variação 1 e a Variação 2 no *near field*, para todos os tamanhos de janela.

		Sem Ruído		**Com Ruído**	
		Centro	***Lateral***	***Centro***	***Lateral***
Sinal s1	***Variação 1***	4,93	13,22	4,50	11,62
	Variação 2	4,93	36,12	4,50	14,52
Sinal s4	***Variação 1***	5,64	29,82	1,84	61,87
	Variação 2	5,64	50,68	1,84	61,87
Sinal s6	***Variação 1***	8,32	29,70	0,29	50,44
	Variação 2	1,84	66,52	0,29	65,54

- Tipo de sinal: sinal de banda estreita $s1$, sinal de fala $s4$ e o sinal da chaleira $s6$.
- Velocidade do movimento: devagar e rápida.
- Com ruído e sem ruído.

Os resultados para todos estes casos são sumarizados, para as janelas de 500 e 50 amostras, respectivamente, nas Tabelas 4.4 e 4.5.

Lendo os dados das tabelas, podemos tirar as seguintes conclusões:

- Igualmente ao que já acontecia no caso de fonte estática, o sinal de banda estreita é o que tem pior comportamento, seguida do sinal de fala e a chaleira,

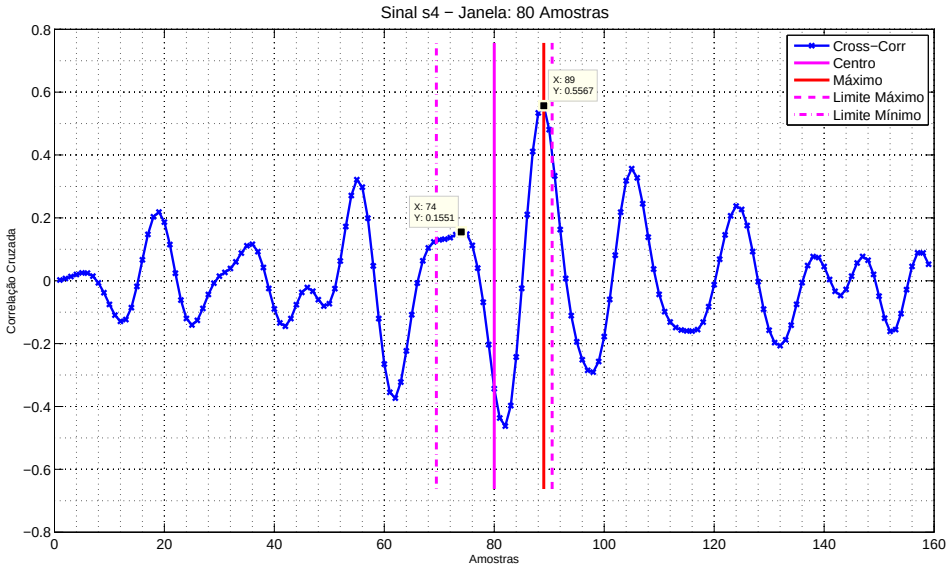

Figura 4.6: FCC para o sinal $s4$ sem ruído no *near field* e posição $F1$ de fonte lateral e um tamanho de janela de 80 amostras.

que é o caso que gera melhores resultados.

- Os resultados são melhores para a janela de 50 amostras, que corresponde, porém, a um maior tempo de execução.
- A primeira modificação é a que tem pior comportamento em todos os casos.
- A terceira modificação é a melhor em alguns poucos casos, particularmente quando usamos o sinal da chaleira.
- A modificação que melhor funciona em geral é a segunda.

Para olhar mais claramente os resultados, vamos considerar alguns casos específicos. Inicialmente, vamos comparar o resultado para o sinal $s1$, sem ruído e velocidade lenta, considerando o CCM sem e com a primeira variação, cujas estimativas ao longo do tempo são mostradas na Figura 4.7 para o caso da janela de 500 amostras. Nesse caso, podemos ver que a primeira modificação permitiu a estimativa (em geral, incorreta) em 34 blocos que não geravam estimativa na versão original do algoritmo.

O resultado para a mesma situação anterior, considerando agora apenas as primeira e segunda modificações é mostrado na Figura 4.8. Nesta figura, podemos

Tabela 4.3: Valores do erro médio, em graus, da Variação 2 e a Variação 3 no *near field*, para todos os tamanhos de janela.

		Sem Ruído		**Com Ruído**	
		Centro	***Lateral***	***Centro***	***Lateral***
Sinal s1	***Variação 2***	4,93	36,12	4,50	14,52
	Variação 3	4,93	36,12	4,50	14,52
Sinal s4	***Variação 2***	5,64	50,68	1,84	61,87
	Variação 3	5,64	29,82	1,84	61,87
Sinal s6	***Variação 2***	1,84	66,52	0,29	65,54
	Variação 3	6,63	59,40	0,29	65,54

observar a redução significativa do erro em alguns instantes, graças à seleção do primeiro e não mais o maior pico da .

Finalmente, podemos ver uma comparação entre os desempenhos das segunda e terceira modificações, por exemplo, para o sinal $s6$ com 500 amostras, velocidade alta e sem ruído, conforme ilustrado na Figura 4.9, na qual a terceira modificação representou uma vantagem.

Tabela 4.4: Valores do erro médio, em graus, com uma janela de 500 amostras para as três variações desenvolvidas.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s1	***Variação 1***	25,05	35,85	43,70	32,80
	Variação 2	14,60	22,16	43,70	32,80
	Variação 3	21,16	22,55	43,70	32,80
Sinal s4	***Variação 1***	11,11	10,72	10,77	10,30
	Variação 2	9,34	9,54	8,26	9,13
	Variação 3	9,26	9,61	8,92	9,21
Sinal s6	***Variação 1***	5,04	6,16	4,67	7,57
	Variação 2	3,34	6,71	4,98	7,27
	Variação 3	3,55	4,40	4,10	6,55

Tabela 4.5: Valores do erro médio, em graus, com uma janela de 50 amostras para as três variações desenvolvidas.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s1	***Variação 1***	22,42	31,96	40,67	29,62
	Variação 2	13,33	19,27	40,59	29,57
	Variação 3	17,61	19,44	40,59	29,57
Sinal s4	***Variação 1***	15,91	15,49	17,12	16,71
	Variação 2	10,85	11,01	8,75	8,77
	Variação 3	11,81	11,68	9,99	9,77
Sinal s6	***Variação 1***	8,11	8,48	8,70	10,64
	Variação 2	5,39	7,07	5,80	7,60
	Variação 3	5,60	6,59	5,90	7,96

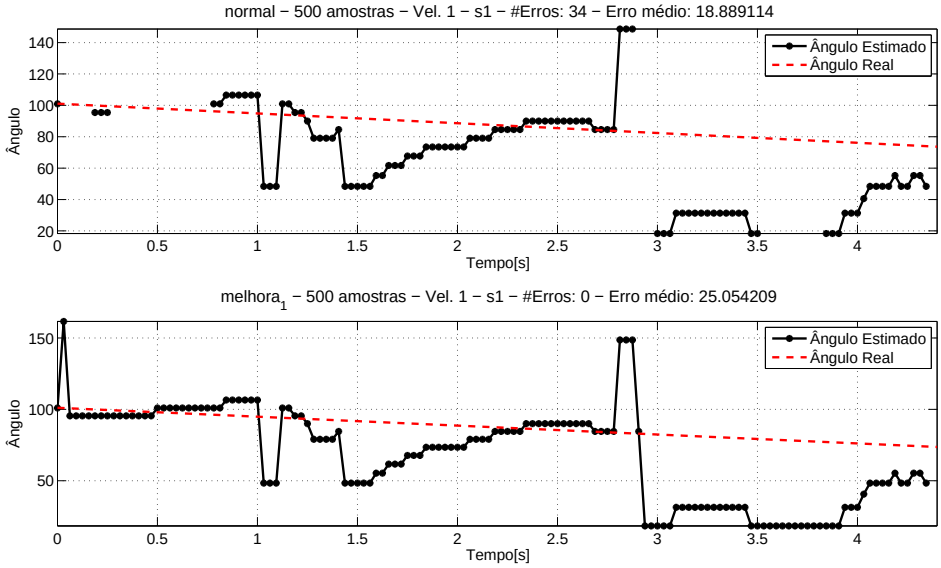

Figura 4.7: Estimativa de DoA ao longo do tempo para as versões original (acima) e com a primeira modificação (abaixo) do algoritmo CCM, sinal $s1$ e janela de 500 amostras.

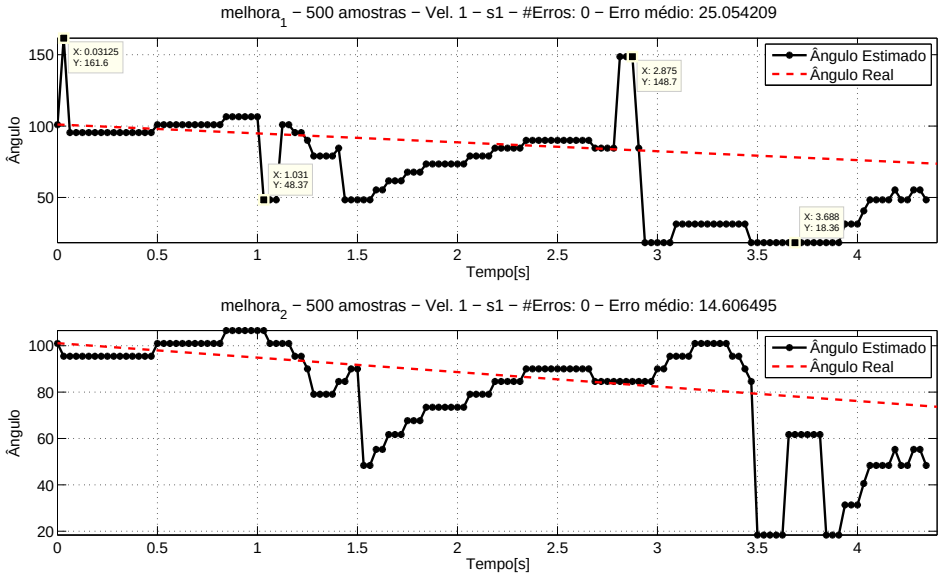

Figura 4.8: Estimativa de DoA ao longo do tempo para o algoritmo CCM com a primeira (acima) e segunda (abaixo) modificações, sinal $s1$ e janela de 500 amostras.

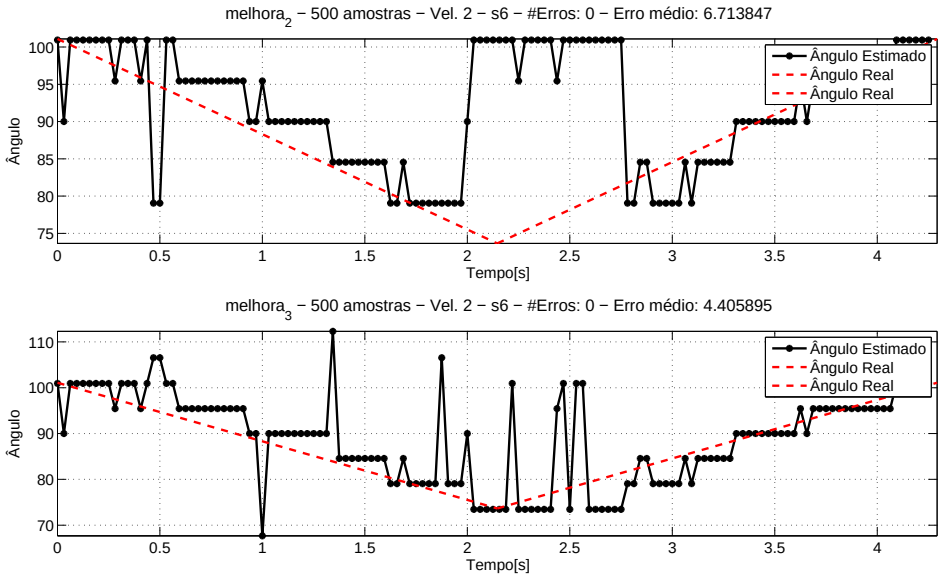

Figura 4.9: Estimativa de DoA ao longo do tempo para o algoritmo CCM com a segunda (acima) e terceira (abaixo) modificações, sinal $s6$ e janela de 500 amostras.

4.3 Métodos da Família da Correlação Cruzada Generalizada

O algoritmo de correlação cruzada generalizado ("GCC", do inglês *generalized cross-correlation*) é o método mais usado em geral para a estimativa do atraso entre dois sinais. Este método também usa a aproximação do modelo *free-field* e só dois microfones. No caso, a função GCC entre dois sinais y_1 e y_2 é calculada usando o domínio da frequência, de modo que

$$r_{y_1,y_2}^{GCC}(p) = F^{-1}[\Psi_{y_1,y_2}(f)] = \int_{-\infty}^{\infty} \Psi_{y_1y_2}(f) e^{j2\pi fp} df, \tag{4.2}$$

onde $F^{-1}[\cdot]$ denota a transformada de Fourier inversa. No caso,

$$\Psi_{y_1,y_2}(f) = \vartheta(f)\phi_{y_1,y_2}(f) \tag{4.3}$$

é o espectro cruzado generalizado, onde

$$\phi_{y_1,y_2}(f) = E[F[y_1].F^*[y_2]] \tag{4.4}$$

e $\vartheta(f)$ é uma função de importância relativa de cada componente espectral. Diferentes funções $\vartheta(f)$ geram diferentes versões do algoritmo GCC. Em particular, fazendo $\vartheta(f) = 1, \forall f$, o GCC se degenera no algoritmo CCM.

4.3.1 *Smoothed Coherence Transforms*

A variante SCOT (do inglês *smoothed coherence transform*) do algoritmo GCC define a função $\vartheta(f)$ de modo a minimizar o impacto da flutuação dos níveis do som na estimativa do TDOA. Para isto, realizamos um prebranqueamento dos sinais dos microfones antes de executar o espectro cruzado através da definição

$$\vartheta(f) = \frac{1}{\sqrt{E[|F[y_1]|^2]E[|F[y_2]|^2]}}. \tag{4.5}$$

Desta forma, o espectro cruzado resultante é tal que

$$\begin{aligned}
\Psi_{y_1y_2}^{SCOT}(f) &= \frac{\alpha_1\alpha_2 e^{-j2\pi f\tau} E[|S(f)|^2]}{\sqrt{E[|Y_1(f)|^2]E[|Y_2(f)|^2]}} \\
&= \frac{\alpha_1\alpha_2 e^{-j2\pi f\tau} E[|S(f)|^2]}{\sqrt{\alpha_1^2 E[|S(f)|^2] + \sigma_{v1}^2(f)} \cdot \sqrt{\alpha_2^2 E[|S(f)|^2] + \sigma_{v2}^2(f)}} \\
&= \frac{e^{-j2\pi f\tau}}{\sqrt{1 + \frac{1}{SNR_1(f)}} \cdot \sqrt{1 + \frac{1}{SNR_2(f)}}},
\end{aligned} \tag{4.6}$$

onde

$$\sigma_{vn}^2(f) = E[|V_n(f)|^2] \tag{4.7}$$

$$SNR_n(f) = \frac{\alpha_n^2 E[|S(f)|^2]}{E[|V_n(f)|^2]}, \quad n = 1, 2. \tag{4.8}$$

Se os níveis de SNR são os mesmos nos dois microfones, temos

$$\Psi_{x_1x_2}^{SCOT}(f) = \left(\frac{SNR(f)}{1 + SNR(f)}\right) \cdot e^{-j2\pi f\tau}. \tag{4.9}$$

Portanto, a eficácia do algoritmo SCOT para a estimativa do atraso varia com a SNR. Para um SNR suficientemente grande, porém, temos

$$\Psi_{x_1x_2}^{SCOT}(f) \approx e^{-j2\pi f\tau}, \tag{4.10}$$

e a estimativa de atraso se torna independente da potência do sinal da fonte. Na prática, por esta análise, o método SCOT é teoricamente superior ao algoritmo CCM original quando o nível de ruído é baixo.

4.3.1.1 Resultados Experimentais

Agora vamos analisar a eficácia do prefiltro SCOT para a estimativa do ângulo, e vamos comparar os resultados com as variações que fizemos na Seção 4.2. Para executar esta análise, vamos utilizar os sinais em movimento e no *far field*, que vai ser o cenário mais habitual, e vamos comparar os resultados de erro médio com o algoritmo CCM incorporando a segunda modificação, que, conforme visto na Seção 4.2.2, apresentou o melhor desempenho geral. Para uma janela de 500 amostras, os erros médios para estes dois algoritmos são dados na Tabela 4.6. Desta tabela, podemos ver claramente um melhor desempenho do algoritmo SCOT em quase todas as situações, principalmente na ausência de ruído.

Este resultado médio, porém, esconde um pouco o desempenho dos algoritmos ao longo do tempo, o qual é indicado na Figura 4.10 para o caso do sinal $s4$ com velocidade lenta e sem ruído. Nesta figura, no geral, percebemos um desempenho menos ruidoso do algoritmo SCOT do que o CCM original. Em diversos instantes, porém, o algoritmo SCOT se mostrou inferior ao algoritmo CCM.

Exemplos da GCC para este caso são mostrados nas Figuras 4.11 e 4.12. No primeiro caso, obtemos uma boa estimativa do atraso usando o algoritmo SCOT

Tabela 4.6: Valores do erro médio, em graus, com uma janela de 500 amostras para o CCM com a Variação 2 e com prefiltro SCOT.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s1	***Variação 2***	14,60	22,16	43,70	32,80
	SCOT	6,84	7,61	15,62	13,65
Sinal s4	***Variação 2***	9,34	9,54	8,26	9,13
	SCOT	6,57	6,82	11,13	10,56
Sinal s6	***Variação 2***	3,34	6,71	4,98	7,27
	SCOT	4,45	4,16	4,54	4,97

num instante em que o sinal de voz está ativo. Já o segundo caso é feito num instante de silêncio, o que provavelmente acarretou no fraco desempenho do algoritmo SCOT neste caso.

Usando uma janela de 50 amostras, os resultados gerados pelos algoritmos CCM com a segunda modificação e o SCOT são resumidos na Tabela 4.7. Neste caso, em geral, ambos os algoritmos melhoram de desempenho, com o SCOT mantendo sua superioridade em relação ao CCM modificado.

Tabela 4.7: Valores do erro médio, em graus, com uma janela de 50 amostras para o CCM com a Variação 2 e com prefiltro SCOT.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s1	***Variação 2***	13,33	19,27	40,59	29,57
	SCOT	5,64	6,04	7,73	7,14
Sinal s4	***Variação 2***	10,85	11,01	8,75	8,77
	SCOT	6,89	6,81	7,84	7,39
Sinal s6	***Variação 2***	5,39	7,07	5,80	7,60
	SCOT	6,08	6,49	6,17	6,85

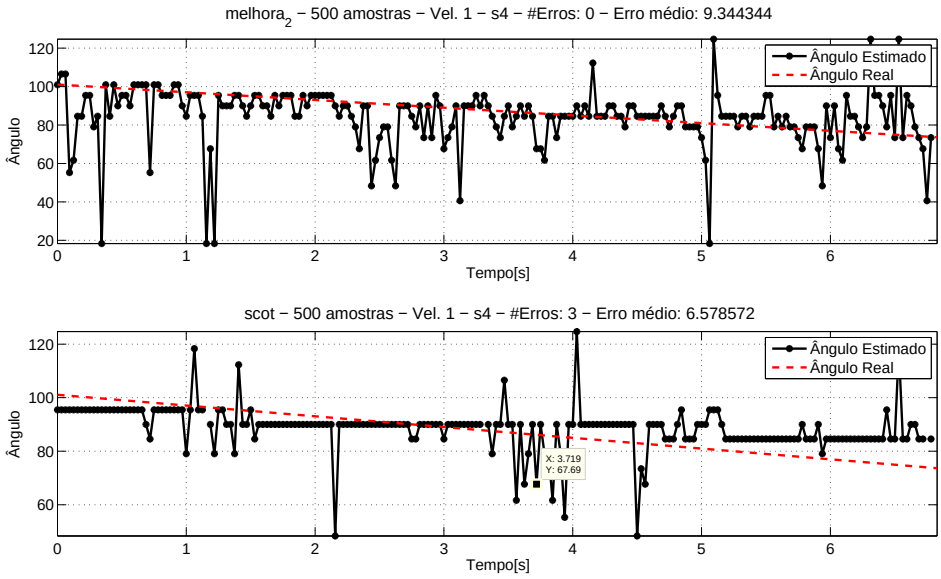

Figura 4.10: Estimativa de DoA dos algoritmos CCM com segunda modificação (acima) e SCOT (abaixo) para sinal $s4$, sem ruído, velocidade lenta da fonte e janela de 500 amostras.

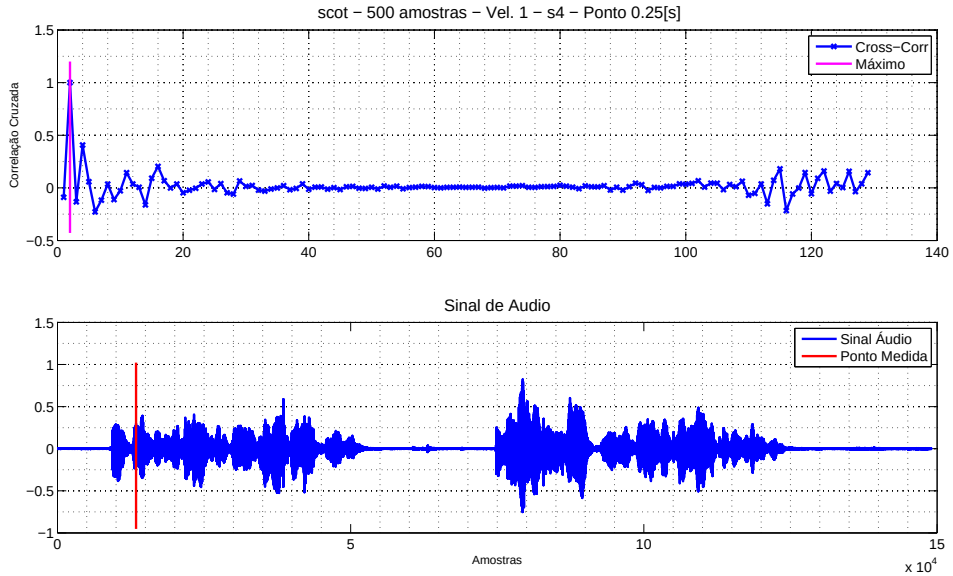

Figura 4.11: Exemplo de GCC do algoritmo SCOT para sinal $s4$, sem ruído, velocidade lenta da fonte e janela de 500 amostras medidas no ponto $x = 0,25$ s.

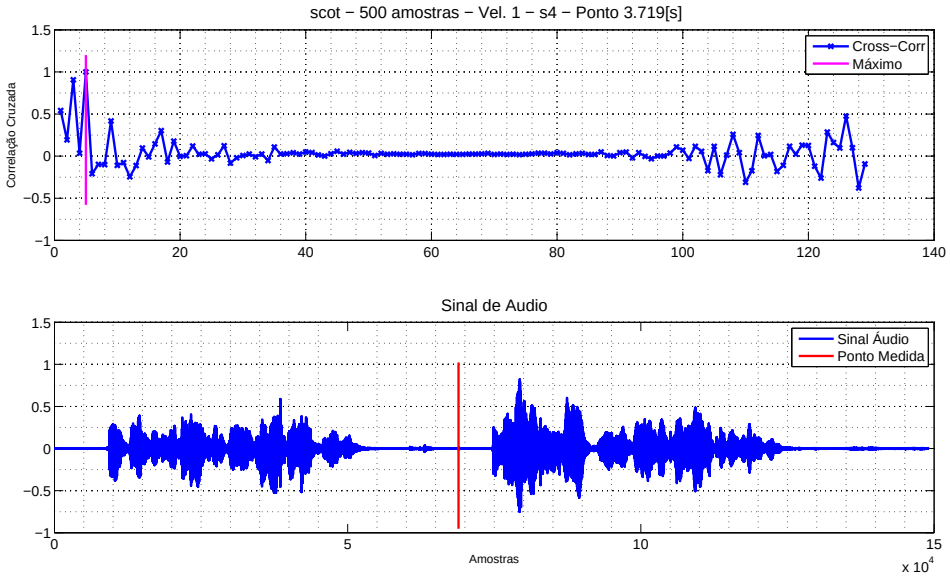

Figura 4.12: Exemplo de GCC do algoritmo SCOT para sinal $s4$, sem ruído, velocidade lenta da fonte e janela de 500 amostras medidas no ponto $x = 3,719$ s.

4.3.2 *Phase Transform*

Uma forma alternativa de impor o conceito de importância espectral relativa é eliminar a importância da amplitude do espectro da correlação cruzada já a que a informação desejada de atraso está contida na fase desta função. Sendo assim, definindo

$$\vartheta(f) = \frac{1}{|\phi_{y1y2}(f)|}, \tag{4.11}$$

temos o método da transformada de fase (PHAT), para o qual o espectro cruzado generalizado é tal que

$$\phi_{y1y2}^{PHAT}(f) = e^{-j2\pi f\tau}. \tag{4.12}$$

Substituindo (4.12) em (4.2), a função GCC ideal é dada por

$$r_{y1,y2}^{PHAT} = \int_{-\infty}^{\infty} e^{j2\pi f(p-\tau)} df = \begin{cases} \infty, & p = \tau \\ 0, & \text{resto} \end{cases} \tag{4.13}$$

4.3.2.1 Resultados Experimentais

O desempenho do algoritmo PHAT é avaliado inicialmente considerando uma janela de 500 amostras, gerando os resultados indicados na Tabela 4.8. Desta tabela, notamos um desempenho do algoritmo PHAT superior ao CCM clássico porém inferior, em geral, ao do algoritmo SCOT.

A estimativa ao longo do tempo do algoritmo PHAT é vista na Figura 4.13 no caso da fonte emitindo o sinal $s4$ sem ruído, com velocidade baixa em relação ao Kinect e janela de 500 amostras. Nesta figura, observamos um desempenho do PHAT bastante similar ao do SCOT: baixa variância e um comportamento ligeiramente quantizado ao longo do tempo.

Considerando uma janela de 50 amostras, para o sinal $s4$, temos os resultados indicados na Tabela 4.9, a qual, como anteriormente constatado, indica uma melhoria do desempenho médio do algoritmo, porém ainda ligeiramente inferior ao SCOT.

Tabela 4.8: Valores do erro médio, em graus, com uma janela de 500 amostras para a GCC sem prefiltro e a Variação 2 e com prefiltro SCOT e PHAT.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s1	***Variação 2***	14,60	22,16	43,70	32,80
	SCOT	6,84	7,61	15,62	13,65
	PHAT	7,22	7,87	17,92	12,56
Sinal s4	***Variação 2***	9,34	9,54	8,26	9,13
	SCOT	6,57	6,82	11,13	10,56
	PHAT	8,09	8,25	14,51	13,32
Sinal s6	***Variação 2***	3,34	6,71	4,98	7,27
	SCOT	4,45	4,16	4,54	4,97
	PHAT	4,58	4,21	5,60	5,94

Tabela 4.9: Valores do erro médio, em graus, com uma janela de 50 amostras para a GCC sem prefiltro e a Variação 2 e com prefiltro SCOT e PHAT, para o sinal s4.

		Sem Ruído		**Com Ruído**	
		Devagar	***Rápido***	***Devagar***	***Rápido***
Sinal s4	***Variação 2***	10,85	11,01	8,75	8,77
	SCOT	6,89	6,81	7,84	7,39
	PHAT	7,04	7,00	7,93	7,84

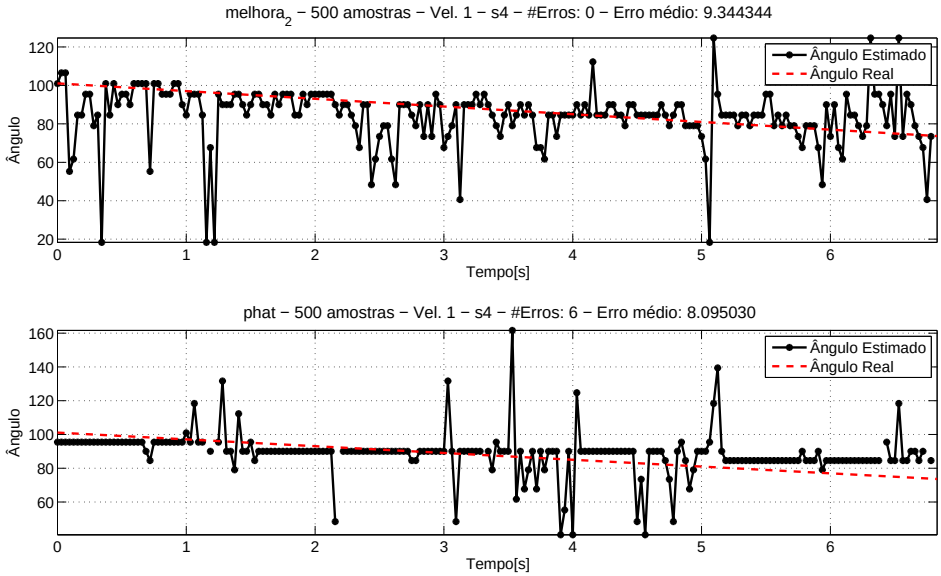

Figura 4.13: Estimativa de DoA dos algoritmos CCM com segunda modificação e PHAT para sinal $s4$, sem ruído, velocidade lenta da fonte e janela de 500 amostras.

4.4 Conclusão

Neste Capítulo 4 analisamos diferentes algoritmos para estimação da DoA.

Em particular, na Seçao 4.2 consideramos três modificações específicas no na maneira que escolhemos a estimativa de atraso temporal na função de correlação entre dois sinais. Os resultados aí incluídos indicam uma melhora de desempenho, em particular considerando as duas primeiras propostas.

Na Seção 4.3 apresentamos a família de métodos baseados na correlação cruzada generalizada, que utilizam de um processo de prefiltragem dos sinais. Em particular consideramos os métodos de *smoothed coherence transform* (SCOT) e *phase transformation* (PHAT). Ambos os métodos apresentaram desempenho bastante superior ao CCM original, particularmente para valores altos de SNR. Na presença significativa de ruído, os resultados variam conforme o tipo de sinal considerado em nossos experimentos.

De modo geral, nossos experimentos indicaram um desempenho superior do algoritmo SCOT em relação aos demais algoritmos.

Capítulo 5

Conclusão

Neste projeto analisamos diferentes algoritmos para o problema de estimação de direção de chegada (DoA) baseados na função de correlação cruzada entre os sinais obtidos por um arranjo de microfones. Em particular, focamos nossos estudos na solução do problema de DoA usando a plataforma Kinect da Microsoft.

Os sinais e os cenários que utilizamos foram descritos no Capítulo 2. Em particular, realizamos análises de desempenho de diferentes algoritmos com três tipos de sinal: de banda estreita, de fala e o assobio de uma chaleira. Também temos duas posições diferenciadas da fonte no *far-field* e no *near-field* do Kinect. Foram testadas variantes dos sinais com ou sem ruído significativo. Todo este conjunto de situações deu origem a uma ampla base de dados desenvolvida no âmbito do presente projeto de graduação.

No Capítulo 3 estudamos a eficácia do algoritmo CCM para a estimativa do atraso entre os sinais adquiridos por dois microfones distintos. A primeira análise considerou a situação de *far-field* da fonte. De modo geral, observamos três tipos principais de erro de estimativa: erro de resolução, erro de deslocamento do pico, e erro de ruído no cálculo da FCC, que provoca picos maiores do que o correto.

No geral, o algoritmo CCM básico apresentou bom desempenho para a posição central da fonte e mau desempenho para a posição lateral, particularmente para sinais senoidais ou de fala na presença de ruído.

Para o *near-field* os resultados são qualitativamente os mesmos do que os para o *far-field*, com uma piora no sentido quantitativo, devido ao fato da frente de onda não mais ser bem modelada por um plano.

Finalmente, para a análise com movimento, vimos que a qualidade dos resultados depende do tamanho da janela ao longo do tempo. Para os sinais que tinham piores resultados sem movimento - banda estreita - precisamos de uma janela menor - que requer mais tempo e recursos de computação - do que quando usamos o sinal da chaleira. Também vemos que uma maior velocidade do movimento da fonte provoca piores resultados de estimativa, como era de se esperar.

No Capítulo 4 analisamos diferentes algoritmos de estimação de DoA. Inicialmente, foram consideradas algumas modificações do CCM padrão, particularmente na escolha do atraso na função de correlação. Posteriormente, apresentamos a família de métodos da correlação cruzada generalizada, com dois diferentes tipos de pre-filtro: *smoothed coherence transform* (SCOT) e a *phase transformation* (PHAT). Vimos que para todos os tipos de sinal, se não temos muito ruído, os dois métodos funcionam melhor que a correlação cruzada clássica mesmo com variações. Inclusive para sinais de banda estreita a melhora é muito grande. Se temos ruído, os resultados variam dependendo do tipo de sinal. Em geral, os algoritmos com pre-filtro são melhores, mas a diferença de eficácia em relação à correlação clássica é menor. Entre os métodos com pre-filtros, de modo geral, o método SCOT funciona melhor que o método PHAT.

Baseados nos resultados mostrados ao longo deste projeto, podemos dizer que o problema de estimação de DoA usando a plataforma Kinect e os algoritmos baseados na função de correlação cruzada tem vários problemas. Possíveis melhorias no processo podem incluir:

- Teste de diferentes pre-filtros.
- Uso da informação temporal (de estimativas anteriores) para o cálculo da estimativa atual.
- Uso de diferentes modificações (como as testadas aqui para o CCM ou mesmo novas variantes) em combinação com os pre-filtros SCOT, PHAT ou mesmo outros existentes na literatura.
- Combinação das estimativas geradas por todos os pares de microfones do Kinect para gerar uma única estimativa. Esta combinação deve levar em conta o

grau de confiabilidade de cada par, o qual depende da distância entre os seus microfones.

Referências Bibliográficas

[1] J. Benesty, J. Chen e Y. Huang, "Microphone Array Signal Processing (Springer Topics in Signal Processing)", Springer-Verlag Berlin Heidelberg, 2008.

[2] J. H. DiBiase, "A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays", MSc Thesis, Trinity College, UK, 1991.

[3] "Kinect: How works its multiarray microphone", http://www.computableminds.com/post/Kinect/multiarray/microphone/how-works/xbox-360 (acessada em julho de 2012).