

Redução de Ruído para Sistemas de Reconhecimento de Voz Utilizando Subespaços Vetoriais

Gutemberg Gonçalves dos Santos Júnior

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich, Ph.D.

Orientador

Antonio Marcus Nogueira Lima, Dr.

Orientador

Campina Grande, Paraíba, Brasil

©Gutemberg Gonçalves dos Santos Júnior, Maio de 2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Redução de Ruído para Sistemas de Reconhecimento de Voz Utilizando Subespaços Vetoriais

Gutemberg Gonçalves dos Santos Júnior

Dissertação de Mestrado apresentada em Maio de 2009

Angelo Perkusich, Ph.D.

Orientador

Antonio Marcus Nogueira Lima, Dr.

Orientador

José Ewerton Pombo de Farias, D.Sc.

Componente da Banca

José Sérgio da Rocha Neto, D.Sc.

Componente da Banca

Marcos Ricardo Alcântara Moraes, Ph.D.

Componente da Banca

Campina Grande, Paraíba, Brasil, Maio de 2009

Resumo

O estabelecimento de uma interface de comunicação através da voz entre seres humanos e computadores vem sendo perseguido desde o início da era da computação. Nesta direção, diversos avanços foram realizados nas últimas seis décadas, permitindo o uso comercial de aplicações com reconhecimento de voz nos dias atuais. Entretanto, fatores como ruídos, reverberações, distorções entre outros, comprometem o desempenho desses sistemas ao reduzir a taxa de acerto quando submetidos a ambientes adversos. Assim, o estudo de técnicas que diminuam os efeitos desses problemas é de grande valia e vem ganhando destaque nas últimas décadas.

O trabalho apresentado nesta dissertação tem como objetivo a redução dos problemas referentes aos ruídos característicos de ambientes automotivos, tornando os sistemas de reconhecimento de voz utilizados nesses ambientes mais robustos. Dessa forma, o controle de funcionalidades não-críticas de um automóvel, ou seja, funcionalidades que não coloquem em risco a vida do usuário como tocadores de música e ar condicionado, pode ser realizado através de comandos de voz.

O sistema proposto é baseado numa etapa de pré-processamento do sinal de voz através do método de subespaços vetoriais. O desempenho deste método está diretamente relacionado com as dimensões (linhas \times colunas) das matrizes representativas do sinal de entrada. Levando isso em consideração, a decomposição ULLV, apesar de se tratar de uma aproximação do método de subespaços vetoriais, foi utilizada por oferecer uma menor complexidade computacional quando comparada a métodos tradicionais baseados na decomposição SVD. O sistema de reconhecimento de voz Julius foi o escolhido para o estudo de caso por se tratar de um sistema desenvolvido em código livre que oferece um alto desempenho. Um banco de dados de voz com 44800 amostras foi gerado como modelo de um ambiente automotivo. Por fim, a robustez do sistema foi avaliada e comparada com um método tradicional de redução de ruído chamado subtração espectral.

Abstract

The establishment of a speech-based communication interface between humans and computers has been pursued since the beginning of the computer era. Several studies have been made over the last six decades in order to accomplish this interface, making possible commercial use of speech recognition applications. However, factors such as noise, reverberation, distortion among others degrades the performance of these systems. Thus, reducing their success rate when operating in adverse environments. With this in mind, the study of techniques to reduce the impact of these problems is of a great value and has gained prominence in recent decades.

The work presented in this dissertation aims to reduce problems related to noise encountered in an automotive environment, improving the speech recognition system robustness. Thus, control of non-critical features of a car, such as CD player and air conditioning, can be performed through voice commands.

The proposed system is based on a speech signal preprocessing step using the signal subspace method. Its performance is related to the size (lines \times columns) of the matrices that represents the input signal. Therefore, the ULLV decomposition was used because it offers a lower computational complexity compared to traditional methods based on SVD decomposition. The speech recognizer Julius is an open source software that offers high performance and was the chosen one for the case study. A noisy speech database with 44800 samples was generated to model the automotive environment. Finally, the robustness of the system was evaluated and compared with a traditional method of noise reduction called spectral subtraction.

Sumário

1	Introdução	1
1.1	Revisão bibliográfica	3
1.1.1	Métodos no domínio da frequência	4
1.1.2	Métodos no domínio do tempo	6
1.2	Motivação	7
1.3	Estrutura da dissertação	10
1.4	Definições e notações utilizadas	11
2	Conceitos fundamentais	12
2.1	Produção da fala	12
2.1.1	Fonemas e fones	13
2.1.2	Espectro da fala humana	14
2.2	Sistemas de reconhecimento de voz	16
2.2.1	Julius - sistema de reconhecimento contínuo de palavras	18
2.3	Resumo	21
3	Métodos de subespaços de sinais	22
3.1	Modelo linear	24
3.1.1	Representação dos dados	25
3.1.2	Matrizes de covariância	26
3.2	Decomposição do sinal	26
3.3	Decomposição em valores singulares	27
3.3.1	Método de subespaços de sinais e o SVD	28
3.4	Método de <i>prewhitening</i>	30
3.5	Decomposição em valores singulares generalizada	31
3.6	Decomposição ULLV	31
3.7	O Algoritmo da decomposição ULLV	33
3.7.1	Algoritmo inicial	34
3.7.2	Atualização da decomposição ULLV	35

3.7.3	Complexidade computacional do algoritmo	36
3.7.4	Mecanismo de atualização	37
3.8	Resumo	38
4	Implementação do sistema	39
4.1	Abordagem utilizada	39
4.2	Metodologia utilizada	41
4.3	Geração do banco de dados	43
4.4	Método de subespaços de sinais	44
4.4.1	Implementação	44
4.4.2	Custo Computacional da Implementação	46
4.4.3	Escolha dos parâmetros	47
4.5	Configuração do Julius	48
4.5.1	Modelo acústico e gramática	48
4.5.2	Opções de configuração	51
5	Resultados obtidos	52
5.1	Avaliações com ruído branco	52
5.2	Avaliações com ruído colorido	53
5.3	Avaliações com ruído de um carro parado	53
5.4	Avaliações com ruído de um carro em movimento	55
5.5	Resumo	55
6	Conclusões e trabalhos futuros	57
	Referências bibliográficas	59
A	Fundamentos de álgebra linear	65
A.1	Espaços vetoriais	65
A.1.1	Subespaços vetoriais	66
A.2	Projeções	66
A.2.1	Projeções oblíquas	66
A.2.2	Projeções ortogonais	67
A.3	Consistência de um sistema linear	67
A.4	Decomposição em valores singulares - SVD	69
A.4.1	Cálculo do SVD utilizando EVD	69
A.4.2	Aplicações para os valores singulares	71

Lista de Abreviaturas

EVD	Decomposição em Autovalores
FPGA	Field-Programmable Gate Array
GSVD	Decomposição Generalizada em Valores Singulares
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
MAP	Maximum a Posteriori
MLLR	Maximum Likelihood Linear Regression
SNR	Relação Sinal Ruído
SVD	Decomposição em Valores Singulares
ULLVD	Decomposição ULLV
ULVD	Decomposição ULV
URVD	Decomposição URV
WSS	Processo estocástico em sentido amplo

Lista de Tabelas

1.1	Desempenho no reconhecimento de palavras do software <i>Aurora Digital Speech Recognition</i> utilizando um modelo acústico sem ruídos [4]	4
4.1	Funções Básicas	47
4.2	Decomposição ULLV	47
5.1	Resultados para as amostras contaminadas com ruído branco gaussiano . .	53
5.2	Resultados para as amostras contaminadas com ruído colorido artificial . .	54
5.3	Resultados para as amostras contaminadas com ruído real de um carro parado com o motor ligado	54
5.4	Resultados para as amostras contaminadas com ruído real de um carro trafegando em uma rua movimentada com as janelas abertas	55

Lista de Figuras

1.1	Sinal de voz incidindo na linha de microfones	5
1.2	Resposta do filtro com relação ao ângulo de incidência da onda	6
1.3	Cenário escolhido	8
2.1	Ilustração do trato vocal de um homem [62]	13
2.2	Espectrograma da sentença “ <i>Vou tomar um café</i> ”	15
2.3	Espectrograma das vogais ‘ <i>a</i> ’, ‘ <i>e</i> ’, ‘ <i>i</i> ’, ‘ <i>o</i> ’ e ‘ <i>u</i> ’	15
2.4	Arquivos utilizados na definição de uma gramática para o Julius	19
2.5	Estrutura do sistema de reconhecimento de voz Julius	20
3.1	Representação do sinal desejado X e do sinal com ruído Y	23
3.2	Representação do sinal resultante do método de subespaços vectoriais	24
3.3	Rotação de plano	35
4.1	Cenário escolhido	40
4.2	Abordagem utilizada	40
4.3	Espectro do ruído colorido gerado artificialmente	44
4.4	Fluxograma do método de subespaços de sinais utilizando SVD	45
4.5	Fluxograma do método de subespaços de sinais utilizando ULLVD	46
4.6	Diagrama de comandos	49
4.7	Comportamento frequencial das primeiras palavras dos comandos definidos na gramática	50
A.1	Exemplo de projeção oblíqua	66
A.2	Exemplo de projeção ortogonal	67

Capítulo 1

Introdução

A busca por uma interface ideal para a comunicação entre homens e computadores existe desde o início da era da computação. Visionários como Arthur C. Clarke no seu livro *2001: A Space Odyssey* [9] já idealizavam computadores com capacidade de comunicação vocal com os seres humanos. Apesar de tais previsões serem bastante otimistas, diversos avanços foram realizados na direção do estabelecimento desta interface.

No campo da síntese de voz, as técnicas utilizadas atualmente são capazes de gerar uma fala artificial com uma qualidade tão alta que pode ser considerada quase indistinguível da fala natural [62]. No campo do reconhecimento de voz, apesar de todos os avanços realizados nas últimas seis décadas, existe ainda um longo caminho a ser percorrido. Entretanto, as taxas de acerto atuais já possibilitam a utilização de sistemas de reconhecimento de voz tanto em computadores pessoais quanto em dispositivos embarcados. Este fato vem sendo impulsionado principalmente pelo crescimento na capacidade computacional das máquinas atuais e pelos avanços no campo de processamento de sinais.

Sistemas de reconhecimento de voz podem ser encontrados em diversos cenários e aplicações:

- muitas empresas utilizam sistemas de reconhecimento de voz no atendimento ao cliente através do telefone, como é o caso da Telefônica, da Oi e da CTBC. Nesses sistemas, através da utilização de comandos de voz, um determinado serviço pode ser acessado ou dados como o número de um cartão de crédito podem ser inseridos;
- diversos sistemas de conversão fala-texto estão disponíveis comercialmente para aumentar a produtividade do usuário na digitação de textos. Neste ramo destacam-se os aplicativos *Dragon Naturally Speaking* [57] e o *IBM ViaVoice* [66], ambos disponibilizados pela *Nuance*.
- inúmeras aplicações permitem a utilização de comandos de voz como interface para executar uma determinada ação. Sistemas operacionais como o *Mac OS X*, tocadores

de música como o *Ipod Shuffle*, celulares como o *Nokia N95* são apenas alguns exemplos de aplicações em que algumas funcionalidades podem ser controladas pela voz. Esses sistemas de reconhecimento de comandos de voz estão presentes até em automóveis, como no caso do *Blue&Me* da Fiat [16] e de alguns sistemas em carros conceito como no *Ford Model U Concept Vehicle* [51]. Nesses sistemas uma série de comandos de voz podem, por exemplo, trocar a estação do rádio ou realizar uma ligação em um automóvel, mantendo a visão do motorista na estrada.

Entretanto, diversas dificuldades estão presentes no desenvolvimento de sistemas de reconhecimentos de voz comerciais, dentre as quais pode-se citar:

- humanos utilizam como informações estrutura gramatical, redundância e conhecimentos sobre o assunto sendo discutido para realizar uma previsão da próxima palavra a ser falada;
- não apenas a voz é utilizada na comunicação entre humanos, mas também uma linguagem corporal expressa principalmente por movimentos realizados com as mãos e com os olhos;
- ruídos presentes no sinal de fala são tratados com eficiência pelo sistema auditivo humano. O desempenho deste sistema chega a um nível tão alto que permite a escolha de uma determinada fonte sonora dentre várias presentes no ambiente. Entretanto, em sistemas de reconhecimento de voz este problema ainda não possui solução definitiva [4]. Existe, portanto, um longo caminho a ser percorrido para que esses sistemas estejam no mesmo patamar de qualidade alcançado pelo sistema auditivo humano ao longo da evolução.
- a voz de um determinado locutor possui um conjunto de características únicas, chamado impressão vocal, que é bastante utilizado em sistemas biométricos. Através deste conjunto, um ser humano pode facilmente identificar quem está falando com ele. Entretanto, esta diversidade de características da voz torna bastante difícil o estabelecimento de um modelo eficiente independente do locutor para o reconhecimento de voz.

A presença de ruídos devido ao ambiente, reverberação ou distorção no sinal de fala degrada bastante o desempenho de sistemas de reconhecimento de voz. A dificuldade de tratar este tipo de degradação chega a um nível tão alto que continua a ser um problema não resolvido [4]. Uma explicação mais detalhada sobre os problemas envolvidos no desenvolvimento de sistemas de reconhecimento de voz pode ser encontrada em [21]. O escopo desta dissertação é estudar o desempenho do método de subespaços para a redução de

ruído em sinais de voz com o intuito de aumentar a robustez em sistemas de reconhecimento de voz, ou seja, manter uma alta taxa de acerto mesmo quando estes sistemas são submetidos a ambientes ruidosos.

1.1 Revisão bibliográfica

O desempenho de sistemas de reconhecimento de voz é afetado na presença de ruídos aditivos e multiplicativos provenientes principalmente do canal de comunicação pelo qual o sinal é transmitido. O estudo de técnicas para a redução de ruídos em sinais de voz é, portanto, de grande interesse e diversos esforços vêm sido dirigidos nesta direção. Ao longo dos anos estas pesquisas levaram ao surgimento de diversas técnicas para cancelar ou reduzir estes ruídos. Comparações entre alguns destes métodos que visam uma melhora na relação sinal/ruído em sinais de voz podem ser encontradas em [40], [23], [15], [25].

A grande maioria dos métodos existentes para a redução de ruídos em sinais de voz introduzem alguma distorção no sinal. Como as distorções toleradas pelos humanos e pelos computadores são bastante diferentes [12], alguns métodos que melhoram a qualidade de voz para um ser humano não necessariamente tornará um sistema de reconhecimento de voz mais robusto. Um sistema de reconhecimento de voz pode ser considerado robusto quando seu desempenho é pouco afetado quando utilizado sob condições adversas, ou seja, em ambientes ruidosos que degradem a qualidade do sinal de fala. Uma boa coleção de métodos que visam o melhoramento da qualidade de voz através da redução de ruídos pode ser encontrada no livro *Springer Handbook of Speech Processing* [4].

Os sistemas de reconhecimento de voz desenvolvidos atualmente para identificar comandos através da voz em condições ideais, ou seja, situações com ausência de ruído, conseguem atingir excelentes taxas de reconhecimento. Entretanto, quando utilizados em ambientes com ruído, suas taxas de acerto caem drasticamente [25], [20], [1]. A Tabela 1.1, disponível no livro *Springer Handbook of Speech Processing* [4], exibe o desempenho de um sistema de reconhecimento de voz submetido a diversas condições de relação sinal/ruído. A sequência de tarefas realizadas para a avaliação do sistema seguiu os padrões impostos pelo grupo AURORA. Este grupo faz parte do comitê STQ (*Speech, Transmission Planning, and Quality of Service*) do instituto ETSI (*European Telecommunications Standards Institute*) [63] e tem como objetivo padronizar algoritmos para sistemas de reconhecimento de voz robustos. O método de avaliação desenvolvido pelo grupo para sistemas de reconhecimento de voz é bastante utilizado em todo o mundo. As ferramentas desenvolvidas pelo AURORA, disponíveis gratuitamente em [22], contêm todas as informações e dados necessários para a formulação de modelos acústicos, assim como *scripts* para simular diversos tipos de ambientes ruidosos.

Tabela 1.1: Desempenho no reconhecimento de palavras do software *Aurora Digital Speech Recognition* utilizando um modelo acústico sem ruídos [4]

SNR (dB)	Desempenho (%)
Sem ruído	99,62
20	94,58
15	85,21
10	66,01
5	38,39
0	15,40
-5	7,14

Através da análise da Tabela 1.1, verifica-se que o desempenho de sistemas de reconhecimento de voz é bastante sensível à presença de ruídos. Este problema pode ser encarado como um dos maiores empecilhos para o uso comercial destes sistemas. Assim, o desenvolvimento de técnicas que possam tornar um sistema de reconhecimento de voz robusto é de grande interesse. Estas técnicas são baseadas, geralmente, em um mecanismo de pré-processamento do sinal de voz. O objetivo deste mecanismo é reduzir o efeito dos ruídos do ambiente, mantendo assim uma alta taxa de reconhecimento de voz independentemente do cenário de uso.

Diante da grande quantidade de métodos existentes para o tratamento do sinal de voz, podemos dividi-los em métodos no domínio da frequência e métodos no domínio do tempo.

1.1.1 Métodos no domínio da frequência

Diversas técnicas de redução de ruídos em sinais de voz são realizadas no domínio da frequência. Nesta linha, destaca-se a técnica de subtração espectral proposta por *Boll* em [6]. Este método é bastante importante por sua simplicidade matemática, sendo uma das primeiras técnicas a oferecer capacidade real de implementação [19]. A subtração espectral é baseada na estimação da magnitude média do espectro do ruído do sinal. Essa estimação é subtraída do sinal com ruído, resultando em um sinal com uma maior relação sinal/ruído. O grande problema deste método é a distorção causada frequentemente por esta subtração, resultando em uma grande quantidade de ruídos musicais presentes no sinal filtrado [65].

Um método bastante importante e que se tornou uma das fundações em sistemas de redução de ruído é o filtro de Wiener. Este filtro, proposto por *Norbert Wiener* em 1949

no seu clássico livro *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications* [73], tem como princípio a minimização do erro quadrático médio da saída do filtro em relação ao sinal desejado. Um problema desta técnica é a necessidade de uma boa estimação dos espectros de potências do sinal sem ruído e do ruído contaminante, o que não é uma tarefa fácil de se obter em ambientes com ruídos não-estacionários.

Nas décadas de 80 e 90, o método de cancelamento de ruído adaptativo criado em 1975 por *B. Widrow et al* [72] foi frequentemente proposto para a redução de ruídos em sinais de voz [10], [26], [68]. Este método, que utiliza geralmente dois microfones para captar o sinal, fornece bons resultados quando existe uma forte dependência entre os ruídos presente nas entradas. Enquanto isso, o sinal de voz deve estar bem representado em apenas um microfone. Estas condições são dificilmente encontradas em ambientes reais, resultando em uma diminuição no seu desempenho. Entretanto, o trabalho apresentado por *Henrik Sahlín et al* em [55] possibilita a separação dos dois sinais através de técnicas de filtragem inversa.

Uma técnica que utiliza vários microfones e que vem ganhando bastante destaque no campo de redução de ruído é o *beamforming*. Esta técnica é composta por um conjunto de microfones normalmente arranjados em forma de linha e operando como uma espécie de filtro espacial. Assim, sinais vindos da direção desejada são amplificados enquanto os vindo de outras direções são atenuados. A Figura 1.1 exibe uma onda do tipo plana incidindo em um conjunto de microfones em um determinado ângulo θ . A resposta do filtro espacial em relação ao ângulo de incidência da frente de onda é ilustrado na Figura 1.2. Apesar do processo envolvido nesta técnica de filtragem poder ser executado tanto no domínio do tempo quanto na frequência, em ambientes com ruído de banda larga é preferível que seja utilizado o domínio da frequência.

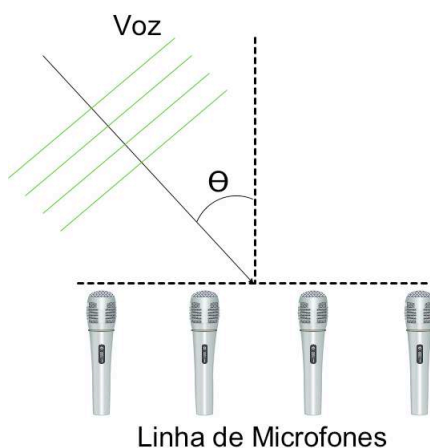


Figura 1.1: Sinal de voz incidindo na linha de microfones

Para alcançar bons resultados utilizando esta técnica os sinais de ruído presentes nos microfones utilizados devem possuir uma correlação alta. Entretanto, em um ambiente automotivo os ruídos característicos são não-estacionários, diminuindo bastante o desempenho deste método [28]. Mais informações sobre este método podem ser encontradas em [8], [4].

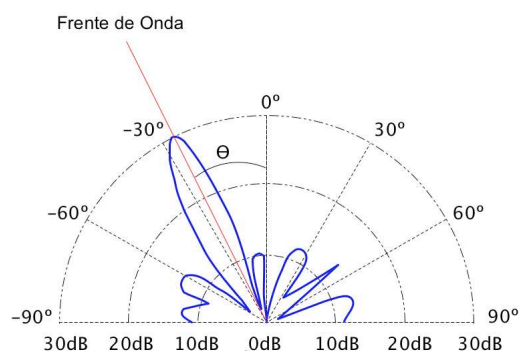


Figura 1.2: Resposta do filtro com relação ao ângulo de incidência da onda

1.1.2 Métodos no domínio do tempo

As técnicas baseadas no domínio da frequência sofrem de um problema em comum. Não possuem informações concretas sobre a fase do sinal desejado e portanto utilizam estimatórias que nem sempre são precisas. Uma técnica que pode ser considerada como uma versão do método de subtração espectral no domínio do tempo é a técnica de subespaços de sinais [28]. Devido a sua excelente capacidade de redução de ruído, a técnica baseada em subespaços de sinais vem recebendo bastante atenção nos últimos anos [31]. Como se trata de uma técnica que possui uma notável complexidade computacional, os avanços realizados na área de eficiência de cálculos matriciais [60], [24], [61], especialmente em decomposições matriciais que possuam a característica de revelar o posto da matriz, têm servido como impulso para a escolha desta técnica em aplicações que necessitam de uma melhora na relação SNR.

Os primeiros estudos baseados em subespaços de sinais datam da década de 1980, com a publicação do trabalho realizado por *Tufts e Kumaresan* em [64]. Neste artigo, foi considerado a redução de ruídos através de métodos de predição linear com sinais compostos por uma soma de senóides amortecidas.

Em [11], um algoritmo de redução de ruído baseado na decomposição em valores singulares foi proposto. Este método oferece ótimos resultados para sinais infectados com ruído branco, tendo grandes avanços na melhoria da robustez do sistema. Entretanto,

devido ao grande peso computacional exigido pelo algoritmo, estes sistemas se tornaram de difícil aplicação para a época.

Com o intuito de obter bons resultados também para ruídos coloridos, um novo método foi proposto baseado no QSVD [36]. Nesta técnica, duas transformações são realizadas no sinal de ruído de forma implícita no algoritmo. A primeira tem o intuito de tornar o espectro do sinal de ruído o mais similar possível com um ruído branco [56] (técnica de *prewhitening*). Ao final do processo, uma transformação inversa é necessária para restaurar as características originais do espectro do sinal (técnica de *dewhitening*).

Embora os métodos [11] e [36] forneçam bons resultados, a capacidade computacional necessária para sua execução é bastante elevada. Isto se deve ao fato de não existirem mecanismos de atualização eficientes para as matrizes resultantes da decomposição em valores singulares. Assim, surgiram novas técnicas baseadas em aproximações do SVD [49] e do QSVD através de decomposições matriciais triangulares, como a URVD [58], a ULVD [59], e a ULLVD [45], [46], [5], [27]. A grande vantagem das decomposições triangulares reside na existência de mecanismos de atualização, ao contrário das decomposições SVD e QSVD. Assim, com a adição ou remoção de uma linha de dados nas matrizes de entrada, este mecanismo é utilizado, evitando que todo o processo de decomposição seja refeito. O mecanismo de atualização pode ainda tirar proveito das estruturas Hankel e Toeplitz utilizadas geralmente nas matrizes de entrada, reduzindo bastante a complexidade computacional necessária. Por exemplo, enquanto na decomposição em valores singulares o grau de complexidade do algoritmo para uma matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ é da ordem de $\mathcal{O}(mn \log(n))$ [74], no caso das decomposições triangulares é possível alcançar $\mathcal{O}(nm)$ quando estas estruturas são utilizadas [27], [35], [17]. Os resultados obtidos com estas técnicas são bastante promissores e muitas pesquisas vêm sendo direcionadas para esta área.

1.2 Motivação

O desenvolvimento de sistemas de reconhecimento de voz comerciais em ambientes adversos continua sendo um problema não resolvido. Ambientes como automóveis possuem uma quantidade ampla de ruídos não-estacionários, diminuindo bastante a taxa de reconhecimento de voz. Uma investigação detalhada sobre a influência de ruídos característicos em um ambiente automotivo no desempenho de sistemas de reconhecimento de voz pode ser encontrado em [20].

O desenvolvimento de sistemas de redução de ruído para este tipo de ambiente é vital para que altas taxas no reconhecimento de palavras sejam alcançadas. Apesar das dificuldades, é esperado que a próxima geração de interface de comunicação entre humanos

e automóveis seja através de comandos de voz e comunicação sem fio. O objetivo desta interface é prover mais segurança para o motorista [1], permitindo que o acesso a algumas funcionalidades seja realizado através da voz. Desta forma, a visão do motorista pode ser mantida na estrada por mais tempo.

O cenário escolhido para a utilização do sistema de redução de ruído é composto por um dispositivo móvel que pode ser acoplado a um automóvel. O dispositivo móvel possui um sistema de reconhecimento de voz que, quando conectado ao veículo, possibilita o acesso a diversas funções através da fala, tanto do dispositivo móvel quanto do carro. Uma análise detalhada dos problemas enfrentados na concepção de um sistema de reconhecimento de voz para este ambiente pode ser visto em [71]. A Figura 1.3 ilustra o cenário descrito.

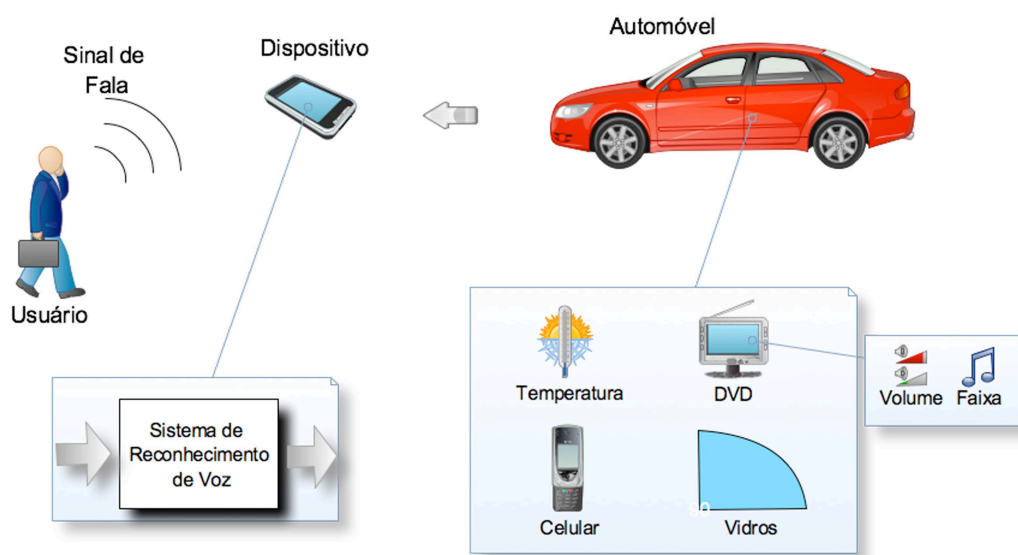


Figura 1.3: **Cenário escolhido**

Aplicativos semelhantes ao ilustrado na Figura 1.3 já existem comercialmente, como o caso do *Blue&Me* da *Fiat* [16]. Apesar destes aplicativos exibirem um excelente funcionamento em situações com poucas perturbações no sinal de voz, seu desempenho continua sendo bastante influenciado pelo ruído do ambiente, como os ruídos do motor, do vento e das conversas paralelas dentro do automóvel [41]. Diante disso, essa dissertação tratará apenas do problema do ruído de ambiente que afeta o desempenho desses aplicativos, levando em consideração que o sistema de reconhecimento de voz utilizado nestes aplicativos já possuem um desempenho satisfatório em ambientes sem ruído.

O trabalho realizado por *Andrew Kun et al* em [41] estuda o desempenho de um motorista utilizando um sistema de reconhecimento de voz enquanto dirige. Como resultado, constatou-se que este desempenho está diretamente relacionado com a precisão do sistema de reconhecimento de voz. Assim, o desenvolvimento de técnicas que permitam manter

uma alta taxa de acerto no reconhecimento de palavras alta independentemente do cenário de uso é de grande interesse. Em sistemas como o apresentado na Figura 1.3, deve ser levado em consideração não apenas a grande quantidade de ruídos presentes, mas também a capacidade computacional disponível no dispositivo. Dessa forma, o processamento dos dados pode ser realizado de forma robusta e em um tempo razoável, ou seja, a resposta do sistema pode ser dada em um tempo compatível com a aplicação.

A ampla gama de ruídos presentes em um automóvel torna necessário que os sistemas de reconhecimento de voz utilizados ofereçam bons resultados para as mais diversas situações. Em outras palavras, que forneçam uma saída confiável mesmo na presença de todos estes ruídos. Entretanto, tornar um sistema de reconhecimento de voz robusto diante de tal situação não é uma tarefa fácil. Basicamente, duas soluções se sobressaem:

1. a utilização de um banco de dados contendo amostras de voz com ruído cujo objetivo é o de introduzir as características deste na parametrização da voz. Com essa abordagem, o sistema torna-se bastante robusto para ambientes onde os ruídos presentes são semelhantes aos utilizados na etapa de treinamento do sistema de reconhecimento de voz.
2. a utilização de uma etapa de pré-processamento do sinal de voz com o objetivo de reduzir a quantidade de ruído presente no sinal de fala antes do início do processo de reconhecimento de voz em si.

Cada uma destas soluções possuem seus pontos fortes e fracos. A primeira, apesar de fornecer resultados melhores que a segunda, é uma técnica bastante atrelada ao aplicativo de reconhecimento de voz, necessitando que alterações na parametrização da voz sejam realizadas para a introdução de informações sobre o ruído. Com isso, torna-se difícil reaproveitar uma solução desenvolvida para um determinado sistema de reconhecimento de voz em outro. A segunda técnica fornece resultados bastante satisfatórios na melhora da robustez do sistema, além de poder ser facilmente reutilizada em outros sistemas de reconhecimento de voz. Ainda mais, por se tratar de uma técnica mais independente do sistema de reconhecimento de voz utilizado, torna-se possível o acoplamento de módulos dedicados em *hardware* para o pré-processamento da voz. Isto possibilita que sistemas com baixos recursos computacionais não sejam sobrecarregados.

Para o caso de um ambiente automotivo, a primeira solução é uma escolha bastante complicada tendo em vista todos os ruídos não-estacionários que o sistema está submetido. A criação de um banco de dados para ser utilizado em um ambiente tão imprevisível torna-se impraticável. Além disso, por se tratar de uma implementação atrelada ao sistema, o processamento deverá ser realizado internamente ao dispositivo. Toda essa quantidade de

dados a serem processados exigiria sistemas com uma grande capacidade computacional, inviabilizando o uso em dispositivos móveis.

A abordagem através do pré-processamento do sinal de voz com um módulo separado é a mais atrativa e será escolhida. Este módulo será localizado anteriormente ao sistema de reconhecimento de voz para que este tenha na entrada um sinal com menos ruído. Esta abordagem permite não apenas bons resultados, mas também soluções dedicadas em *hardware* quando necessárias.

Para o desenvolvimento do sistema de redução de ruído proposto nesta dissertação, a técnica de subespaços de sinais é bastante aconselhável por dispensar um modelo explícito da voz e do ruído presente no ambiente. Assim, o método se apresenta como uma alternativa bastante robusta para variações nas características dos ruídos presentes nos automóveis. Contudo, as limitações na capacidade computacional dos dispositivos embarcados torna necessário o desenvolvimento de otimizações algorítmicas para diminuir a complexidade computacional do método. Isto leva a implementações utilizando decomposições matriciais com custo computacional menor, como o caso da decomposição ULLV.

1.3 Estrutura da dissertação

A organização desta dissertação pode ser descrita da seguinte forma. No Capítulo 2 uma introdução sobre o mecanismo de produção de fala humano é abordado. São explicadas também algumas características em frequência e formas de análise específicas para esse tipo de sinal. Além disso, são abordados os principais conceitos utilizados nos sistemas de reconhecimento de voz atuais.

No Capítulo 3 uma explicação do método de subespaços de sinais é abordada. Detalhes com relação a representação dos sinais e operações matemáticas necessárias também são apresentados. Posteriormente, o algoritmo utilizado para a decomposição ULLV é explicado, assim como alguns conceitos necessários para sua implementação a exemplos das rotações de plano.

No Capítulo 4 detalhes sobre a implementação do sistema são explicitados. Encontra-se ainda neste capítulo explicações sobre a metodologia utilizada e sobre configurações necessárias para a reprodução do cenário de uso.

Os resultados obtidos para o sistema de redução de ruídos são apresentados no Capítulo 5.

Uma breve revisão de conceitos importantes de álgebra linear para o entendimento do trabalho realizado está disponível no Anexo A.

1.4 Definições e notações utilizadas

Esta seção tem como objetivo apresentar ao leitor a notação utilizada ao longo desta dissertação.

O conjunto dos números reais e complexos serão representados por \mathbb{R} e \mathbb{C} , respectivamente. O espaço vetorial de matrizes com m linhas e n colunas cujos elementos sejam números reais será representado por $\mathbb{R}^{m \times n}$. O espaço vetorial de vetores colunas com m elementos será representado por \mathbb{R}^m .

Matrizes serão representadas por letras maiúsculas em negrito. Vetores por letras minúsculas em negrito. O elemento de uma matriz \mathbf{A} localizado na linha i e na coluna j será representado por a_{ij} . O elemento de um vetor \mathbf{x} localizado na posição i será representado por x_i .

Matrizes identidade de ordem m serão representadas por \mathbf{I}_m . Vetores e matrizes nulas por $\mathbf{0}$. A inversa de uma matriz será representada por \mathbf{A}^{-1} . A operação transposta de uma matriz e de um vetor será representada por \mathbf{A}^T e \mathbf{x}^T , respectivamente.

A norma quadrática de uma matriz \mathbf{A} será representada por $\|\mathbf{A}\|$ enquanto que a norma de Frobenius da mesma por $\|\mathbf{A}\|_F$. O valor absoluto de uma variável a é dado por $|a|$.

O valor esperado de uma variável aleatória qualquer será representado por $E(\cdot)$, a variância por $var(\cdot)$ e a covariância por $cov(\cdot)$. Suas definições estão descritas na equação (1.1).

$$\begin{aligned}
 E(\mathbf{x}) &= \sum_{i=1}^{\infty} x_i p(x_i) \\
 var(\mathbf{x}) &= E((\mathbf{x} - \mu_{\mathbf{x}})^2) \\
 cov(\mathbf{x}, \mathbf{y}) &= E((\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}}))
 \end{aligned} \tag{1.1}$$

Uma matriz \mathbf{A} é dita singular quando seu determinante é nulo, ortogonal quando $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ e ortonormal quando $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ e a norma de cada coluna da matriz \mathbf{A} é unitária.

Capítulo 2

Conceitos fundamentais

O conhecimento do mecanismo de produção de fala dos humanos assim como o funcionamento dos sistemas de reconhecimento de voz atuais são de grande importância para o desenvolvimento e implementação de métodos de redução de ruído voltados para sinais de voz. Uma melhora no desempenho desses métodos pode ser conseguida através da utilização de características próprias da voz humana e do conhecimento do mecanismo envolvido no reconhecimento de palavras.

2.1 Produção da fala

Como qualquer fonte geradora de som, o sistema de produção da voz é composto por elementos produtores de sons (corpos que vibram) e por elementos que modificam ou redirecionam o som (ressonadores). Os principais órgãos envolvidos no processo de produção da fala são o pulmão, a laringe, a faringe, o nariz e várias partes da boca [32], conforme ilustrado na Figura 2.1.

A fonte de energia do sistema vem da força muscular responsável por expelir o ar contido nos pulmões. O fluxo de ar gerado passa então pelo trato vocal, responsável por modificar o espectro para uma faixa audível pelo sistema auditivo.

Todo o processo de produção da fala é baseado na ressonância. O trato vocal é a principal estrutura ressonante do sistema, sendo responsável principalmente pelo sons de vogais. Este elemento é composto por diversos órgãos que vão desde a laringe até os lábios, incluindo a cavidade nasal. Os principais elementos do trato vocal são as pregas vocais, compostas por duas membranas localizadas na laringe. Estas membranas são controladas pelo cérebro, podendo ser tensionadas ou relaxadas através de músculos específicos. Estes movimentos, juntamente com o fluxo de ar proveniente dos pulmões, fazem com que surja uma vibração em uma determinada frequência. As frequências de vibração destas membranas para o caso de homens geralmente variam de 80Hz a 500Hz, enquanto que

para mulheres de 200Hz a 1100Hz [62].

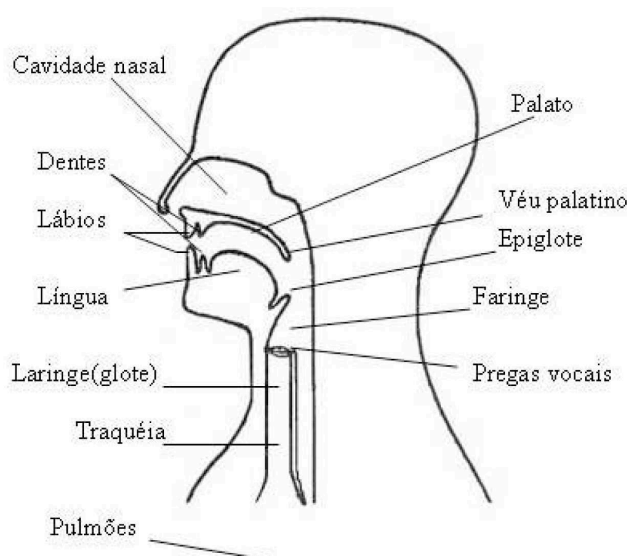


Figura 2.1: Ilustração do trato vocal de um homem [62]

Os sons produzidos pelos humanos podem ser divididos em sonoros ou não sonoros. No primeiro caso, existe uma vibração nas pregas vocais durante a passagem do fluxo de ar originado do pulmão enquanto que no segundo estas membranas estão relaxadas. No caso da língua portuguesa, os sons classificados como sonoros são todas as vogais mais as consoantes 'b', 'v', 'd', 'g', 'z', 'j', 'm', 'n', 'nh', 'l', 'lh', 'r' e 'rr'. Enquanto isso, os sons classificados como não sonoros são as consoantes 'p', 'f', 't', 'k', 's' e 'ch'.

Essa variabilidade de sons é o resultado direto de um processo denominado articulação, onde os outros órgãos do trato vocal entram em ação para modificar o fluxo de ar gerado pelos pulmões. Movimentos de órgãos como a língua, o véu palatino, a mandíbula e os lábios são os responsáveis pela existência de uma quantidade tão grande de sons produzidos. Por exemplo, para a produção do som 's', o ar é forçado a passar por uma estreita abertura entre a língua e os alvéolos dentais (parte saliente no céu da boca logo atrás dos dentes) [62].

2.1.1 Fonemas e fones

As palavras existentes em um determinado idioma são formadas pela combinação dos sons produzidos durante o processo de articulação. Cada palavra é formada por uma sequência definida destes sons, denominados fones. Entretanto, estes elementos variam bastante entre regiões de um país e até mesmo entre pessoas. Essa grande variação torna difícil a formação de um conjunto que represente todos os fones existentes em um

determinado idioma. Essa dificuldade levou a criação de conjuntos de fones concatenados que soem de forma equivalente para os falantes de uma língua. Este conceito mais abstrato para a representação dos sons existentes em um determinado idioma recebeu o nome de fonema.

Cada idioma possui um determinado conjunto de fonemas pelo qual as palavras podem ser representadas, assim como as letras do alfabeto. Entretanto, nem sempre os fonemas correspondem exatamente às letras do alfabeto. Por exemplo, em palavras como *casa* e *lata*, o primeiro 'a' corresponde a um fonema de vogal plena, enquanto o segundo a de vogal reduzida.

O conceito de fonema, por permitir que todas as palavras de um idioma sejam modeladas por um pequeno grupo de elementos, é amplamente utilizado para a síntese e o reconhecimento da fala humana. Estes sistemas lidam com a concatenação ou identificação de um determinado grupo de fonemas para sintetizar ou reconhecer uma determinada palavra. [32].

2.1.2 Espectro da fala humana

A identificação dos fonemas em uma sentença não é uma tarefa trivial pois nem sempre correspondem às letras do alfabeto de um determinado idioma. Apesar dos fonemas possuírem características detalhadas na frequência, a informação de cada fonema depende de variações no tempo dos órgãos articuladores. Assim, em um diagrama espectral de uma determinada palavra, a identificação de que frequências correspondem a um determinado fonema é uma tarefa bastante complicada, dado que informações temporais não são representadas nesse tipo de diagrama.

A solução proposta foi a de representar estes sinais em diagramas que expressem tanto características na frequência quanto no tempo. Estes diagramas, chamados espectrogramas, normalmente utilizam a dimensão horizontal para representar o tempo e a vertical para representar a frequência. A intensidade de cada componente na frequência é representada visualmente através de cores ou áreas mais escuras. O espectrograma da sentença "*Vou tomar um café*", enunciada por um locutor masculino na faixa etária de 25 anos, pode ser visto na Figura 2.2.

Os sons produzidos pelos seres humanos possuem um conjunto de frequências onde boa parte da energia do sinal está concentrada. Estas frequências, denominadas formantes, são as principais características analisadas para diferir um fonema de outro. Analisando as Figuras 2.2 e 2.3 é possível identificar as componentes do espectro referentes às formantes dos sons sonoros contidos na sentença "*Vou tomar um café*". A partir destas informações, as vogais dessa sentença podem ser identificadas visualmente.

Diversos sistemas de reconhecimento de voz, como o Julius [38] e o Sphinx [53], utilizam

os conceitos de fonemas e de formantes para realizar o processo de identificação de uma palavra. Assim, modelos são criados para a identificação de cada fonema e algoritmos detectam o início e o final de uma determinada palavra através de um dicionário contendo quais fonemas constituem determinada palavra. O funcionamento desses sistemas será abordado com mais detalhes na seção 2.2.

O conhecimento do mecanismo de produção da fala, das características frequenciais da voz e de como os algoritmos de reconhecimento de voz tratam essas informações são importantes para a concepção de filtros para redução de ruído. A resposta em frequência destes filtros devem garantir que as formantes dos fonemas não sejam afetadas. Desta forma, consegue-se que grande parte da energia presente no sinal de entrada esteja presente também na saída e o sistema de reconhecimento de voz tenha seu desempenho melhorado com a redução de ruído.

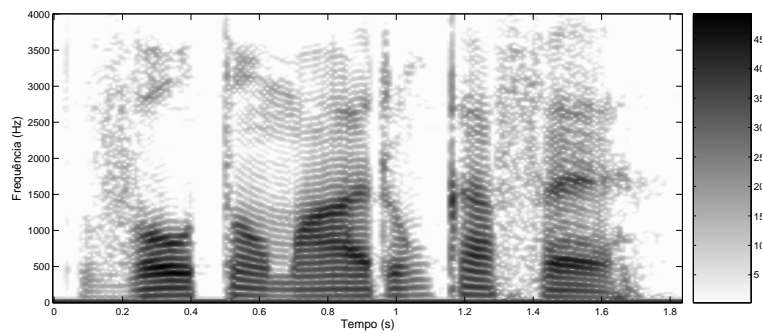


Figura 2.2: Espectrograma da sentença “*Vou tomar um café*”

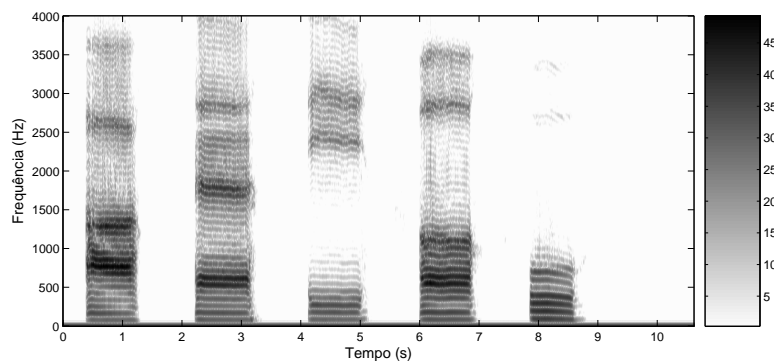


Figura 2.3: Espectrograma das vogais ‘a’, ‘e’, ‘i’, ‘o’ e ‘u’

2.2 Sistemas de reconhecimento de voz

O reconhecimento de voz é um campo da ciência no qual um computador reconhece palavras faladas por um locutor através de um mapeamento do sinal de voz em informações textuais. Observe que o entendimento sobre o significado do que está sendo falado faz parte de um outro campo chamado Processamento de Linguagem Natural.

Os sistemas de reconhecimento de voz estão a cada dia mais presentes em nossas vidas. Frequentemente utilizamos estes sistemas em aplicações como a inserção de textos em computadores, em sistemas de atendimento ao cliente por telefone, no controle de diversas funções em um automóvel entre outras.

O desempenho desses sistemas é avaliado geralmente a partir da quantidade de acertos no reconhecimento de palavras e da velocidade na qual a resposta correta é encontrada. A primeira medida leva à definição da *Taxa de Acerto* de um sistema de reconhecimento de voz através da equação (2.1).

$$\textit{Taxa de Acerto} = \frac{\textit{Quantidade de Palavras Reconhecidas Corretamente}}{\textit{Quantidade de Palavras Faladas}} \times 100\% \quad (2.1)$$

Os sistemas de reconhecimento de voz atuais podem alcançar taxas de acerto de palavras de até 99% quando utilizados em condições ótimas, como o caso do *Dragon Naturally Speaking* [57]. Estas condições podem ser alcançadas quando o aplicativo é utilizado na presença de pouco ruído de ambiente e quando as características da voz do usuário são parecidas com as características do banco de dados de voz utilizado no treinamento do sistema. Entretanto, como discutido na Tabela 1.1, o desempenho desses sistemas caem drasticamente quando utilizados fora dessas condições.

Os sistemas que oferecem maiores taxas de acerto são dependentes do locutor, ou seja, necessitam de uma pequena etapa de treinamento realizado pelo usuário do sistema. Nesta etapa, o usuário pronuncia algumas frases pré-determinadas para realizar uma calibração no sistema de reconhecimento de voz. Entretanto, os sistemas independentes do locutor, os quais não necessitam de treinamento, estão próximos de alcançar esse mesmo patamar de qualidade. O Julius, por exemplo, é um sistema de reconhecimento de voz gratuito independente de locutor que oferece taxas de acerto acima de 95% para condições ótimas [43].

Apesar do crescente uso de sistemas de reconhecimento de voz comercialmente, a utilização destes sistemas em uma grande quantidade de cenários e aplicações leva à escolha de características específicas na etapa de desenvolvimento [4]:

- *Tamanho do Vocabulário*: Esta escolha afeta diretamente na dificuldade de um sistema de reconhecimento de voz identificar corretamente a palavra falada. Quanto

maior o vocabulário e maior a quantidade de palavras parecidas, maior será a dificuldade no reconhecimento das palavras. De acordo com [4], pode-se dividir os sistemas em relação ao tamanho do vocabulário em pequeno (2 a 100 palavras), médio (100 a 1000 palavras) ou grande (acima de 1000 palavras).

- *Sintaxe ou Gramática*: A utilização deste mecanismo tem como intuito restringir quais outras palavras podem vir após uma determinada palavra. Por exemplo, pode-se definir que apenas as palavras "direita" e "esquerda" poderão vir após a sentença "Vire a". Desta forma, tanto a taxa de acerto quanto o tempo de execução do sistema são melhorados.
- *Modo de Fala*: O modo de entrada do sistema pode ser através de fala contínua ou de palavras isoladas. A dificuldade de sistemas de reconhecimento contínuo de palavras é maior pelo fato de que os humanos tendem a juntar o final de uma palavra com o começo de outra em uma sentença, tornando complicado a identificação do término de uma palavra e do começo de outra.
- *Locutor*: Os sistemas de reconhecimento de voz podem ser classificados em dependentes ou independentes do locutor. Os sistemas dependentes do locutor necessitam de uma etapa de treinamento realizado pelo usuário do sistema, podendo oferecer uma maior taxa de reconhecimento. Entretanto, a necessidade do treinamento pelo usuário torna esses sistemas bastante utilizados apenas em aplicações de inserção de textos. Nas demais aplicações a utilização de sistemas independentes do locutor prevalece.
- *Unidade Fundamentais*: Na grande maioria dos sistemas de reconhecimento de voz atuais a identificação de uma determinada palavra se dá pela sequência de fonemas reconhecidos em uma sentença. Esta unidade fundamental de reconhecimento vem sendo bastante utilizada por ser uma unidade mais independente do sotaque do usuário e por permitir que uma grande quantidade de palavras seja representada por um pequeno conjunto de elementos [32]. Entretanto, outras unidades de reconhecimento também podem ser utilizadas, como o caso das sílabas ou das próprias palavras.

Um aplicativo de reconhecimento de voz é baseado em um modelo acústico e em um modelo linguístico ou uma gramática. O modelo acústico é composto por representações estatísticas de fonemas retiradas de um banco de dados de voz e de suas transcrições. Estas representações são então utilizadas para a identificação das palavras. O modelo linguístico é utilizado em aplicações de inserção de texto onde não existe uma sequência

determinada de palavras em uma sentença. Assim, probabilidades são atribuídas para a próxima palavra a ser identificada baseada nas palavras anteriores já reconhecidas.

A técnica mais comum nos sistemas de reconhecimento de voz atuais é a de modelos ocultos de Markov (Hidden Markov Model - HMM). Esta técnica nada mais é que um conjunto de dois processos estocásticos de Markov. O primeiro é uma cadeia de Markov oculta, referente aos dados no domínio do tempo, e o outro um processo observável, relacionado aos dados no domínio da frequência. Os parâmetros de um HMM são otimizados durante a etapa de treinamento do sistema de reconhecimento de voz. Assim, quando as características acústicas das amostras de teste não casam com as das amostras de treinamento, o desempenho do sistema é afetado [14]. Isso acontece principalmente pelos ruídos de ambiente presentes nas amostras de teste e por distorções devido ao canal de transmissão.

A robustez de um sistema de reconhecimento de voz, ou seja, a permanência de uma alta taxa de acerto de palavras mesmo quando utilizado em condições adversas, pode ser melhorada através da busca por um melhor casamento entre o banco de dados de treinamento e as amostras de voz do usuário. Este objetivo pode ser alcançado ou pela modificação do banco de dados de treinamento para conter informações sobre ruídos de ambiente e distorções do canal de transmissão, ou por adaptações realizadas nas amostras de voz do usuário para que os parâmetros acústicos correspondam aos do banco de dados de treinamento.

A modificação do banco de dados de treinamento é realizada geralmente por técnicas de adaptação, como o *Maximum Likelihood Linear Regression* e o *Maximum a Posteriori* [14]. Assim, baseado em informações referentes ao ambiente, os parâmetros do HMM são atualizados.

As adaptações das amostras da voz do usuário podem ser entendidas como processos de filtragem e transformadas com o objetivo de reduzir a quantidade de ruídos e distorções presentes no sinal de fala. Sua principal vantagem é conseguir melhorar a robustez de sistemas de reconhecimento de voz independentemente da tecnologia utilizada no reconhecimento, tornando sua reutilização mais frequente.

Este trabalho utilizará a segunda alternativa na tentativa de obtenção de uma melhora na robustez de um sistema de reconhecimento de voz. O objetivo será o de reduzir o impacto causado por um dos maiores empecilhos para a obtenção de altas taxas de acerto de palavras em ambientes reais, o ruído.

2.2.1 Julius - sistema de reconhecimento contínuo de palavras

O Julius [38] é um sistema de reconhecimento contínuo de palavras em código aberto de alto desempenho baseado em uma estratégia de dois passos. No primeiro passo, uma

busca aproximada pelos fonemas é realizada com o objetivo de gerar um conjunto dos fonemas mais prováveis. Em seguida, o segundo passo realiza uma busca com um maior rigor dentro do conjunto gerado no primeiro passo. Este sistema oferece uma taxa de acerto acima de 95% quando não operando em tempo real e acima de 90% para situações em tempo real para um vocabulário de 20.000 palavras [43].

Além do modelo baseado em fonemas, o Julius também utiliza o modelo trifone que utiliza informações de contexto referentes à fonemas a esquerda e a direita. Isto se dá devido ao fato que um fonema pode soar um pouco diferente quando precedido por fonemas específicos. Assim, mesmo que dois determinados fonemas sejam o mesmo, informações de contexto diferentes à esquerda e à direita podem levar à formação de trifones diferentes [76].

O sistema de reconhecimento Julius foi desenvolvido com um banco de dados de voz japonês que também é oferecido livremente. Entretanto, o algoritmo de identificação é independente do idioma utilizado. Assim, o Julius pode ser utilizado com outros idiomas desde que utiliza modelos acústicos e linguísticos retirados através de um banco de dados de voz do idioma alvo. Estes modelos devem ser gerados com o formato HTK (Hidden Markov Model Toolkit) [34].

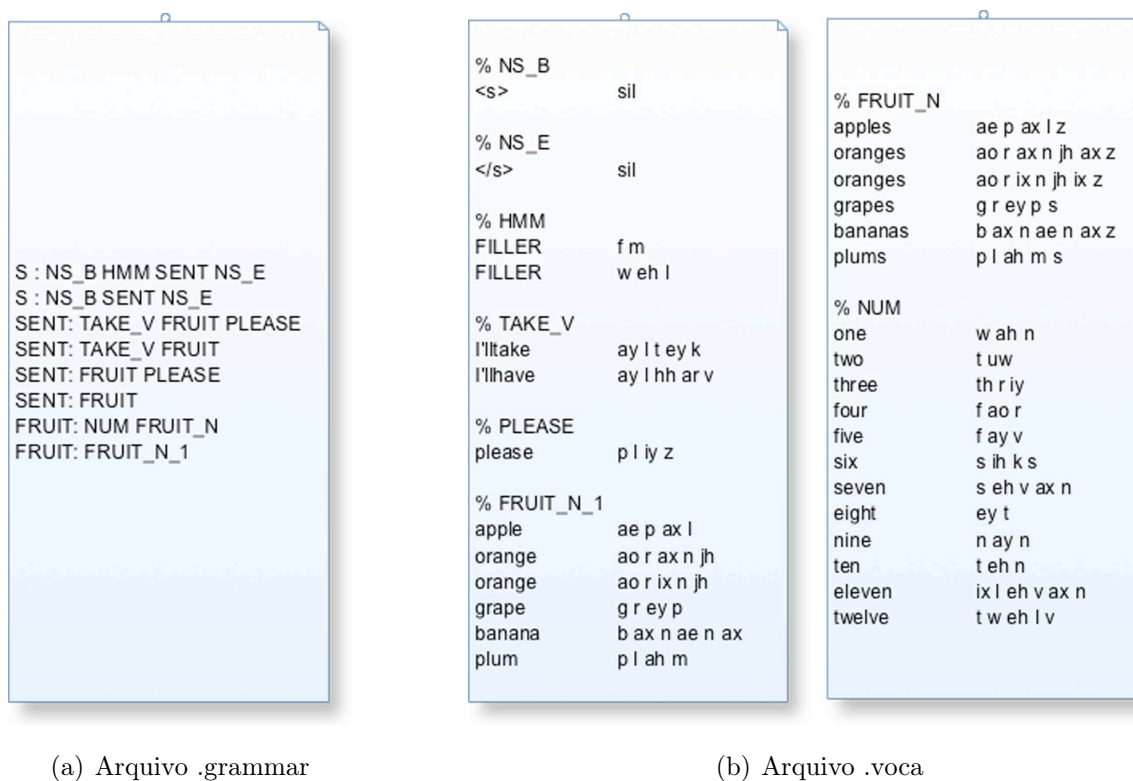


Figura 2.4: Arquivos utilizados na definição de uma gramática para o Julius

A gramática utilizada no Julius segue um padrão próprio disponível em [38]. Este padrão utiliza dois arquivos para a definição da gramática chamados ".grammar" e ".voca".

O primeiro arquivo define como diferentes categorias de palavras podem ser interligadas. O segundo arquivo define quais as categorias existentes e que palavras estão disponíveis em cada categoria, assim como suas divisões fonéticas. A Figura 2.4 ilustra os arquivos ".grammar" e ".voca" retirados do documento de como construir uma gramática para o Julius [38].

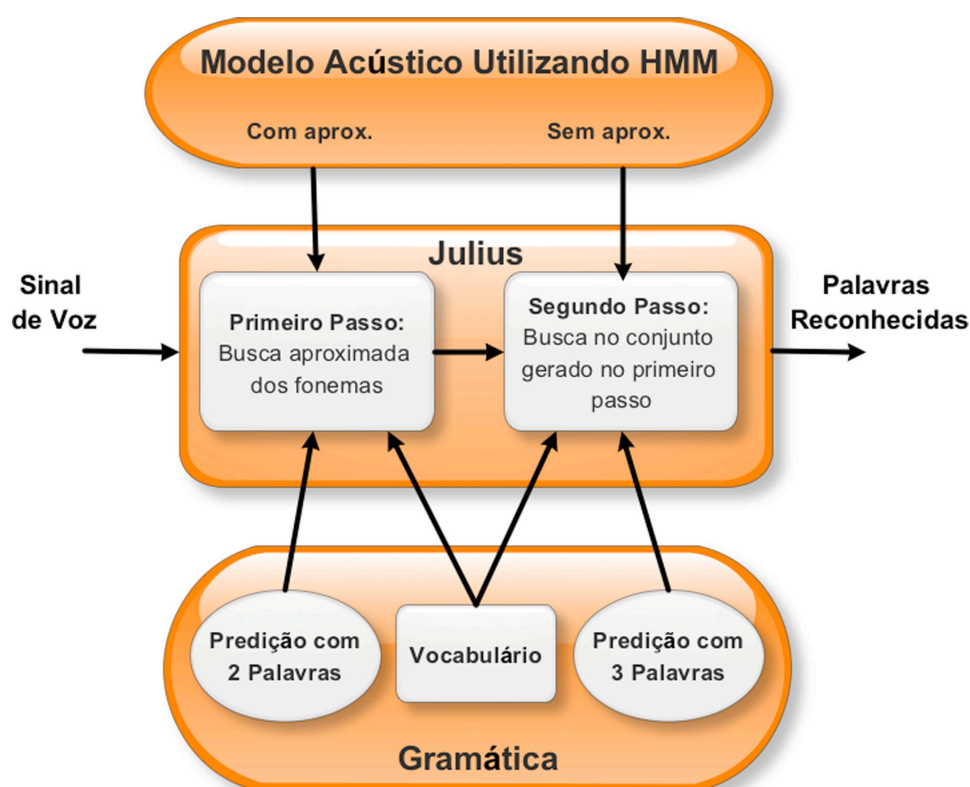


Figura 2.5: Estrutura do sistema de reconhecimento de voz Julius

Nesta definição de gramática os símbolos NS_B e NS_E indicam um silêncio no início e no final da sentença. As categorias de palavras definidas no arquivo ".voca" são HMM , $TAKE$, $PLEASE$, $FRUIT_N_1$, $FRUIT_N$ e NUM , além das categorias correspondentes ao silêncio listadas anteriormente. As possibilidades de combinação dessas categorias são definidas no arquivo ".grammar". Após a definição desses arquivos, deve-se executar um script, disponível no próprio aplicativo, para a geração dos arquivos necessários para a utilização da gramática no Julius. Um diagrama ilustrando a estrutura do Julius informando em que parte do algoritmo são utilizados os modelos acústico e a gramática pode ser observado na Figura 2.5.

O sistema de reconhecimento de voz Julius possui um sistema de redução de ruído baseado no método de subtração espectral. Este método pode estimar o espectro do sinal de ruído considerando que o primeiro quadro contém apenas amostras de ruído ou

utilizando uma média espectral de uma entrada que contenha apenas ruído e carregando-a no Julius [43].

O método de subtração espectral utilizado, apesar de utilizado para melhorar a taxa de acerto quando o sistema é submetido a ambientes com ruído, causa deformação na entrada e não consegue fornecer bons resultados em situação onde a relação sinal-ruído é baixa. Assim, o emprego de outras técnicas podem ser utilizadas para melhorar a robustez desse sistema em conjunto com a técnica já contida no aplicativo.

2.3 Resumo

Os fundamentos do mecanismo responsável pela produção da fala humana foi abordado neste capítulo. As funções dos órgãos envolvidos nesse processo assim como características frequenciais da fala foram explicadas. O conhecimento dessas características são importantes para o entendimento do funcionamento dos sistema de reconhecimento de voz. Ainda, o conceito de fonema foi introduzido e os principais motivos de sua importância para os sistemas de reconhecimento de palavras explicitados.

Em seguida, os fundamentos envolvidos no processo de reconhecimento de voz também foram abordados. As principais características desses sistemas assim como o funcionamento dos mesmos foram detalhados. Por fim, o sistema de reconhecimento de voz Julius, utilizado neste trabalho, foi apresentado e suas principais características especificadas. Foi definido também os conceitos de gramática e de modelo acústico, fundamentais para a execução de um sistema de reconhecimento de voz.

No próximo capítulo será apresentado o método de redução de ruído escolhido neste trabalho utilizado para melhorar a robustez dos sistemas de reconhecimento de palavras.

Capítulo 3

Métodos de subespaços de sinais

Nos últimos anos, diversas abordagens baseadas em subespaços vetoriais vêm ganhando destaque em aplicações de processamento de sinais, em especial na redução de ruído em sinais de voz [28], [29], [31], [27], [75]. Nesta linha, uma estimação linear não paramétrica da voz desconhecida é realizada. Esta estimação leva em consideração a decomposição do sinal com ruído em dois subespaços vetoriais ortogonais, um de sinal e outro de ruído. Esta técnica denomina-se método de subespaços de sinais e, como a grande maioria das técnicas de redução de ruído, necessita de algumas suposições sobre as características do sinal de entrada e do ruído contaminante. Dentre estas, pode-se citar:

- A matriz correlação dos vetores de dados \mathbf{S} do sinal de voz sem ruído é semidefinida positiva, ou seja, alguns autovalores da matriz são nulos. Tal suposição implica que a energia deste sinal está confinada em um subespaço vetorial definido por um subconjunto das colunas dessa matriz.
- A matriz correlação do ruído \mathbf{R} é assumida positiva definida. Assim, a energia do ruído está distribuída em todos os vetores de \mathbf{R} , ou seja, suas colunas expandem todo o espaço vetorial.

A primeira suposição indica que o sinal de voz sem ruído $\mathbf{s} = [s(1)s(2) \cdots s(n)]$ pode ser escrito como uma combinação linear de $p < n$ funções linearmente independentes h_i , conforme pode ser visto no exemplo da seção 3.1. Tanto a forma quanto a base desta combinação linear são dependentes da janela analisada, dado as características dinâmicas da voz.

A segunda suposição resulta em ruído aditivo branco com média zero e totalmente decorrelacionado com o sinal de voz. A suposição de se tratar de um ruído branco é bastante razoável dado que qualquer ruído colorido pode passar por um processo denominado *prewhitening*. Neste processo, uma transformação no espectro do sinal de ruído é realizada com o intuito de torná-lo semelhante a um ruído branco, conforme pode ser

visto na seção 3.4. Uma transformação inversa, denominada *dewhitening*, é necessária para a reconstrução correta do sinal.

Em outras palavras, essas suposições indicam que o sinal de interesse pode ser representado em um subespaço vetorial do espaço do ruído. Por exemplo, imagine um sinal X representado em \mathbb{R}^3 cujas informações estão disponíveis em um subespaço vetorial \mathbb{R}^2 . Este sinal é então contaminado por um ruído que está presente em todas as dimensões do espaço vetorial, resultando no sinal Y . As ilustrações destes sinais podem ser vistas nas Figuras 3.1(a) e 3.1(b).

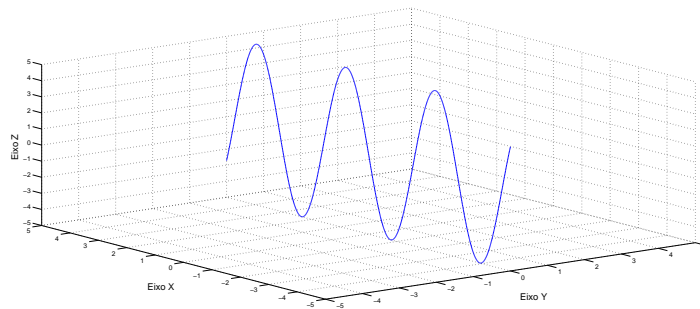
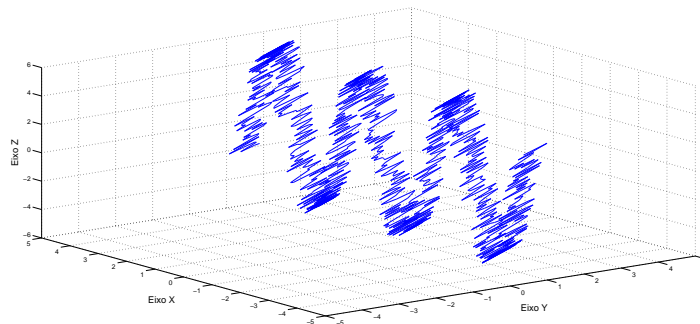
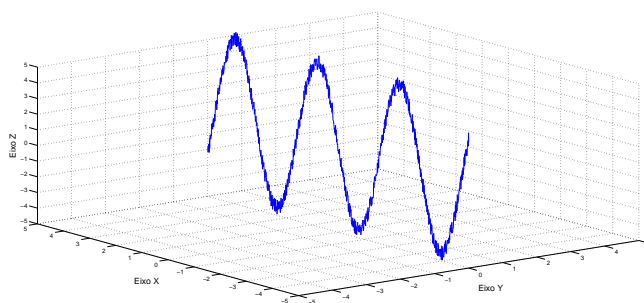
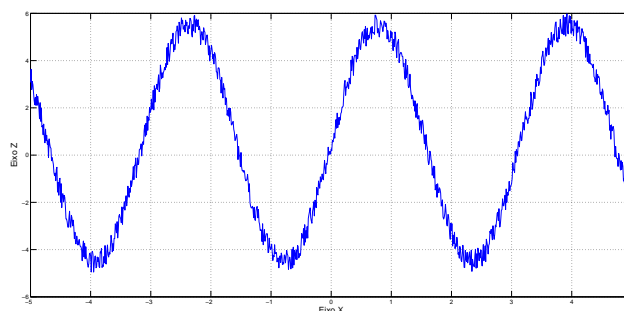
(a) Representação do sinal X (b) Representação do sinal Y

Figura 3.1: **Representação do sinal desejado X e do sinal com ruído Y**

O princípio do método de subespaços vetoriais é encontrar a quantidade de dimensões na qual o sinal de interesse pode ser totalmente representado e em seguida anular o subespaço de ruído. A representação deste passo pode ser visto na Figura 3.2(a).

Com esses passos, a redução de ruído não ocasiona deformação no sinal e outros métodos podem ser utilizados para remover o ruído restante presente no subespaço do sinal. A representação do sinal desejado após a remoção do subespaço de ruído realizada pelo método de subespaços vetoriais pode ser visto na Figura 3.2(b). Os fundamentos envolvidos na decomposição vetorial do espaço euclidiano do sinal e algumas ferramentas básicas de álgebra linear são abordados com mais detalhes no Apêndice A.

(a) Representação do sinal Y após anulação do subespaço do ruído(b) Representação do sinal Y após remoção do subespaço do ruídoFigura 3.2: **Representação do sinal resultante do método de subespaços vectoriais**

3.1 Modelo linear

O sinal de fala apresenta diversas características que o torna não estacionário, dificultando sua modelagem. Entretanto, para janelas curtas, geralmente entre $10ms$ e $30ms$, este sinal pode ser considerado como um processo WSS. Isto torna possível sua representação através de um modelo estocástico linear descrito pela equação (3.1).

$$\mathbf{S} = \mathbf{H}\Theta = \sum_{i=1}^n h_i \Theta_i \quad (3.1)$$

em que $\mathbf{S} \in \mathbb{R}^m$ representa uma sequência de amostras aleatórias do sinal, $\mathbf{H} \in \mathbb{R}^{m \times p}$ é uma matriz ortonormal cujas colunas representam uma base para o subespaço do sinal e $\Theta \in \mathbb{R}^p$ trata-se de um vetor de coeficientes aleatórios de média zero retirados de uma distribuição multivariável.

Conforme pode ser visto em (3.1), qualquer vetor \mathbf{s} pode ser representado como uma combinação linear das p colunas de \mathbf{H} . Isto implica que os vetores coluna de \mathbf{H} expandem o espaço vetorial do sinal, ou seja, o sinal está contido em um espaço vetorial de ordem p . Sendo $p < m$, existem dois subespaços vectoriais de ordem p e $m - p$ com características bastante peculiares. O primeiro, denominado subespaço do sinal, contém todas as informações relacionadas ao sinal e algumas informações remanescentes sobre o ruído. Já

o segundo, denominado subespaço do ruído, contém apenas informações sobre o ruído.

Por se tratarem de subespaços complementares, o subespaço do ruído é ortogonal ao do sinal. Este fato, juntamente com a ausência de informações referentes ao sinal desejado no subespaço do ruído, possibilita a remoção deste subespaço sem causar deformação no sinal. Como resultado, um sinal com uma quantidade menor de ruído pode ser reconstruído através deste modelo linear.

3.1.1 Representação dos dados

Considerando que o ruído presente no sinal seja aditivo, a representação do sinal com ruído pode ser dada por (3.2).

$$\mathbf{X} = \mathbf{S} + \mathbf{R}, \quad \mathbf{S} = \mathbf{H}\Theta \quad (3.2)$$

em que \mathbf{X} representa a matriz de dados do sinal com ruído, \mathbf{S} representa o sinal sem ruído e \mathbf{R} representa o ruído presente no sinal.

Nesta representação, cada linha da matriz \mathbf{X} é considerada como uma realização do experimento deslocada no tempo. Para que as características dos subespaços sejam reveladas, a quantidade de realização de experimentos deve ser muito maior que a dimensão dos vetores que representam esta realização. Estes dados são então organizados em uma matriz na forma Hankel [27], conforme pode ser visto em (3.3).

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_3 & \cdots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \cdots & x_N \end{pmatrix} \quad (3.3)$$

em que x_i representa uma amostra do sinal com ruído no tempo.

Com a utilização dessa estrutura matricial, a estimação das matrizes de covariância de entrada podem ser realizadas de forma bastante simples. Levando em consideração a simetria existente, essa estimação pode ser feita conforme ilustrada na equação (3.4).

$$\mathbf{C}_s \approx \frac{1}{m} \mathbf{X}^T \mathbf{X} \quad (3.4)$$

em que \mathbf{C}_s é uma estimação da matriz de covariância do sinal de entrada e \mathbf{X} representa os dados de entrada organizados em uma matriz com estrutura Hankel.

É bastante comum utilizar-se também matrizes com estruturas Toeplitz, não havendo diferenças significativas entre ambas [28].

3.1.2 Matrizes de covariância

Tendo em vista que a representação do sinal é feita através de realizações de experimentos de um processo estocástico, todo o método de subespaços vetoriais para a remoção de ruído da voz é baseado em matrizes de covariância [28].

A covariância da matriz \mathbf{X} de entrada é dado por (3.5).

$$\mathbf{R}_x = E(\mathbf{X}\mathbf{X}^T) \quad (3.5)$$

Considerando o ruído aditivo, a matriz \mathbf{X} pode ser representada por (3.2). Sendo o ruído totalmente decorrelacionado do sinal de voz, chega-se na relação (3.6).

$$\begin{aligned} \mathbf{R}_x &= E((\mathbf{S} + \mathbf{R})(\mathbf{S} + \mathbf{R})^T) \\ \mathbf{R}_x &= E(\mathbf{S}\mathbf{S}^T) + E(\mathbf{S}\mathbf{R}^T) + E(\mathbf{R}\mathbf{S}^T) + E(\mathbf{R}\mathbf{R}^T) \\ \mathbf{R}_x &= E(\mathbf{S}\mathbf{S}^T) + E(\mathbf{R}\mathbf{R}^T) \\ \mathbf{R}_x &= \mathbf{R}_s + \mathbf{R}_r \end{aligned} \quad (3.6)$$

Considerando o modelo linear para o sinal sem ruído dado pela equação (3.1) e que \mathbf{H} é uma matriz ortonormal, a matriz de covariância do sinal puro pode ser encontrada através de (3.7).

$$\begin{aligned} \mathbf{R}_s &= E(\mathbf{S}\mathbf{S}^T) \\ \mathbf{R}_s &= E((\mathbf{H}\Theta)(\mathbf{H}\Theta)^T) \\ \mathbf{R}_s &= E(\mathbf{H}\Theta\Theta^T\mathbf{H}^T) \\ \mathbf{R}_s &= \mathbf{H}E(\Theta\Theta^T)\mathbf{H}^T \\ \mathbf{R}_s &= \mathbf{H}\mathbf{R}_\Theta\mathbf{H}^T \end{aligned} \quad (3.7)$$

Sendo o ruído branco com variância $\nu_{ruído}^2$, a matriz de covariância do ruído pode ser dada pela expressão (3.8).

$$\mathbf{R}_r = \nu_{ruído}^2 \mathbf{I}_m \quad (3.8)$$

3.2 Decomposição do sinal

Conforme descrito anteriormente, o método de subespaços de sinais utiliza as diferentes características presentes no sinal de fala e no ruído para reconstruir o sinal desejado. Para isso, é necessário a separação do sinal contaminado em dois subespaços vetoriais

ortogonais. O primeiro, de dimensão p , se refere ao sinal enquanto o segundo, de dimensão n , se refere ao ruído. Neste último, nenhuma informação sobre o sinal de fala deve estar presente. Embora o primeiro subespaço seja denominado subespaço do sinal, algumas informações sobre o ruído também estão presentes.

No método de subespaços de sinais, a remoção do ruído presente no sinal de entrada pode ser realizada com os seguintes passos [31]:

- separar o subespaço do sinal do subespaço do ruído através de uma decomposição ortogonal;
- remover o subespaço do ruído;
- remover o ruído remanescente no subespaço do sinal puro com ruído (passo opcional).

O terceiro passo muitas vezes é utilizado com o objetivo de melhorar a performance do método, reduzindo ainda mais a quantidade de ruído remanescente. Entretanto, pelo fato dos sinais de ruído e de fala estarem sobrepostos no subespaço do sinal, a redução do ruído ocasiona distorção no sinal de fala. Portanto, busca-se, neste passo adicional, encontrar o melhor custo-benefício para a aplicação desejada em termos de ruído remanescente e distorção do sinal de fala.

A separação do sinal em dois subespaços se dá por meio de transformações ortogonais, das quais serão citadas neste trabalho a decomposição SVD, a GSVD e a ULLVD.

3.3 Decomposição em valores singulares

A decomposição em valores singulares (SVD - Singular Value Decomposition), introduzida por *E. Beltrami* no seu trabalho *Sulle funzioni bilineari* [2] e independentemente também por *C. Jordan* no seu trabalho *Mémoire sur les formes bilinéaires* [37], é tida como uma das mais poderosas decomposições matriciais existentes, tanto para propósitos teóricos quanto computacionais [70]. Esta ferramenta pode ser utilizada para tratar com matrizes singulares ou muito próximas da singularidade, onde técnicas como eliminação Gaussiana e decomposição LU podem falhar. Esta técnica não apenas indica qual o problema na resolução do sistema linear, como também pode resolvê-lo no sentido de fornecer uma resposta numérica útil, diferente da resposta esperada mas sendo a mais próxima possível. A decomposição em valores singulares é baseada no Teorema 1, cuja demonstração pode ser encontrada no livro *SVD and Signal Processing III: Algorithms, Architectures and Applications* [48].

Teorema 1 *Seja $\mathbf{A} \in \mathbb{R}^{n \times m}$ uma matriz não-nula com posto igual a $r \leq \min(n, m)$, onde m, n e r são inteiros positivos. A matriz \mathbf{A} pode ser expressa como o produto (3.9).*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.9)$$

sendo $\mathbf{U} \in \mathbb{R}^{n \times n}$ e $\mathbf{V} \in \mathbb{R}^{m \times m}$ matrizes ortogonais. A matriz $\mathbf{\Sigma}$ é definida em (3.10).

$$\mathbf{\Sigma} = \begin{pmatrix} \hat{\mathbf{\Sigma}} & 0 \\ 0 & 0 \end{pmatrix} \quad (3.10)$$

em que,

$$\hat{\mathbf{\Sigma}} = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_r \end{pmatrix} \quad (3.11)$$

Esta decomposição é denominada decomposição em valores singulares (SVD). Levando em consideração que o posto da matriz \mathbf{A} é igual a r , pode-se destacar as seguintes propriedades desta decomposição:

- os elementos da diagonal principal da matriz $\mathbf{\Sigma}$ são os valores singulares da matriz \mathbf{A} ;
- as colunas da matriz \mathbf{U} são os vetores singulares à esquerda;
- as colunas da matriz \mathbf{V} são os vetores singulares à direita;
- o posto da matriz \mathbf{A} é dado pela quantidade de valores singulares não nulos;
- o espaço vetorial da matriz \mathbf{A} é dado pela expansão dos vetores singulares da matriz $\mathbf{U} \rightarrow (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$;
- o espaço nulo da matriz \mathbf{A} é dado pela expansão dos vetores singulares da matriz $\mathbf{V} \rightarrow (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$

3.3.1 Método de subespaços de sinais e o SVD

Considerando que o ruído presente no sinal de fala a ser processado é do tipo branco e aditivo, a correlação do sinal ruidoso pode ser calculada por (3.12).

$$\begin{aligned} \mathbf{C}_x &= \mathbf{C}_s + \mathbf{C}_r \\ \mathbf{C}_x &= \mathbf{C}_s + \nu_{\text{ruído}}^2 \mathbf{I} \end{aligned} \quad (3.12)$$

Supondo que as matrizes do sinal \mathbf{S} e do ruído \mathbf{R} satisfazem as expressões (3.13), ou seja, que as colunas da matriz de ruído \mathbf{R} sejam ortogonais e que o sinal de fala seja totalmente decorrelacionado do ruído, pode-se escrever as relações (3.14) e (3.15).

$$\begin{aligned} \frac{1}{m} \mathbf{R}^T \mathbf{R} &= \nu_{\text{ruído}}^2 \mathbf{I}_n \\ \mathbf{S}^T \mathbf{R} &= \mathbf{0} \end{aligned} \quad (3.13)$$

$$\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = (\mathbf{U}_1 \mathbf{U}_2) \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{pmatrix} (\mathbf{V}_1 \mathbf{V}_2)^T \quad (3.14)$$

$$\bar{\mathbf{S}} = \bar{\mathbf{U}} \bar{\mathbf{\Sigma}} \bar{\mathbf{V}}^T = (\bar{\mathbf{U}}_1 \bar{\mathbf{U}}_2) \begin{pmatrix} \bar{\mathbf{\Sigma}}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} (\bar{\mathbf{V}}_1 \bar{\mathbf{V}}_2)^T \quad (3.15)$$

Tendo em vista que a representação do sinal da fala sem ruído pode ser realizada através do modelo exibido em (3.15) e do sinal com ruído através do modelo em (3.14), pode-se observar que o sinal de fala é modelado com uma dimensão superior ao necessário. Assim, a matriz $\mathbf{\Sigma}$ contém alguns valores singulares iguais a zero, mostrando que as componentes do sinal desejado não estão presentes ao longo de todo o espaço vetorial. Ainda, o sinal de ruído está espalhado por todo o espaço vetorial representado pelo modelo e, portanto, sobrepõem algumas componentes do sinal de voz. Entretanto, parte do ruído está isolado em um subespaço vetorial ortogonal ao do sinal de tal forma que pode ser removido sem causar distorção no sinal desejado. Esta remoção se dá por meio de uma redução de ordem k do modelo, obtendo assim a representação (3.16) para o sinal aproximado.

$$\bar{\mathbf{S}} = \bar{\mathbf{U}}_1 \bar{\mathbf{\Sigma}}_1 \bar{\mathbf{V}}_1^T \quad (3.16)$$

Esta simples redução de ordem do sistema pode ser entendido como uma estimação do sinal desejado utilizando o critério dos mínimos quadrados, conforme pode ser visto em [27]. Para a escolha da dimensão k do modelo que resultará em uma remoção apenas de componentes referentes ao ruído no sinal, escolhe-se o valor no qual verifica-se a ocorrência de uma redução brusca nos valores singulares presentes em $\mathbf{\Sigma}$. Esta queda se deve ao fato de haver uma mudança do subespaço do sinal, onde há uma sobreposição de sinal e ruído, para o do ruído. Ainda, este ponto pode ser entendido como um limiar de ruído e permite o cálculo da variância deste, conforme mostrado em (3.17). Esta informação que pode ainda ser utilizada na remoção adicional do ruído sobreposto ao subespaço do sinal.

$$\nu_{\text{ruído}}^2 = \frac{1}{n-p} \sum_{i=p+1}^n \sigma_i^2 \quad (3.17)$$

3.4 Método de *prewhitening*

Toda a discussão realizada na seção 3.3 é feita com base em um sinal infectado com ruído aditivo do tipo branco. Para a consideração de um ruído colorido, a matriz de correlação dada em (3.8) não é mais válida. Assim, surge a necessidade de um método que trate com essa classe de ruídos. Uma forma bastante interessante para estender o método exibido na seção 3.3 é utilizar a técnica de *prewhitening* do sinal.

O método de *prewhitening* é baseado na idéia de tornar o espectro do sinal de interferência presente no sinal o mais similar possível de um ruído branco. Assim, uma transformação é feita no sinal de entrada com o intuito de diagonalizar a matriz de correlação do ruído, ou seja, $\mathbf{R}_r^T \mathbf{R}_r = \mathbf{I}_n$. Esta transformação é feita através de um processo de filtragem que não altera a natureza do sinal de fala [28]. Com a aplicação desta técnica, a matriz de correlação do ruído pode novamente ser expressa por (3.8) e, assim, o método descrito na seção anterior (3.3) pode ser aplicado, necessitando apenas que uma transformação inversa seja realizada no final.

Supondo que a matriz de correlação do ruído é conhecida, a decomposição QR [60] correspondente é dada por (3.18) e a decomposição de Cholesky [60] por (3.19), onde a matriz $\mathbf{R} \in \mathbb{R}^{n \times n}$ é uma matriz triangular superior referente ao fator de Cholesky e $\mathbf{Q} \in \mathbb{R}^{m \times n}$ possui todas suas colunas ortonormais, ou seja, $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_n$.

$$\mathbf{R}_r = \mathbf{Q}\mathbf{R} \quad (3.18)$$

$$\mathbf{R}_r = \mathbf{R}^T \mathbf{R} \quad (3.19)$$

Assim, fazendo uma suposição bastante razoável de que o ruído está espalhado por todo o espaço vetorial do modelo, o posto de \mathbf{R}_r é completo. Isto permite a relação exibida em (3.20).

$$\begin{aligned} \mathbf{X} &= \mathbf{S} + \mathbf{R}_r \\ \mathbf{X}\mathbf{R}^{-1} &= \mathbf{S}\mathbf{R}^{-1} + \mathbf{R}_r\mathbf{R}^{-1} \\ \mathbf{X}\mathbf{R}^{-1} &= \mathbf{S}\mathbf{R}^{-1} + \mathbf{Q}\mathbf{R}\mathbf{R}^{-1} \\ \mathbf{X}\mathbf{R}^{-1} &= \mathbf{S}\mathbf{R}^{-1} + \mathbf{Q} \end{aligned} \quad (3.20)$$

Com esse procedimento, a matriz de correlação do ruído é diagonalizada e o método exibido na seção anterior passa a ser válido. Ao final do procedimento, entretanto, uma multiplicação do sinal estimado pela matriz de *prewhitening* \mathbf{R} deve ser realizada para recuperar as características originais do sinal de fala. Esta operação por muitas vezes recebe o nome de *dewhitening*.

3.5 Decomposição em valores singulares generalizada

A técnica apresentada na seção 3.4 torna possível o tratamento de qualquer tipo de ruído presente em um sinal. Entretanto, existe uma decomposição bem mais elegante que agrega os passos de *prewhitening* e *dewhitening* ao próprio algoritmo. Esta técnica de decomposição se denomina GSVD [69] e foi apresentada pela primeira vez por Van Loan em [44]. A GSVD permite ainda sua aplicação mesmo em casos onde a matriz de correlação do ruído não possui posto completo [27] e é definida pelo Teorema 2, cuja prova pode ser encontrada no livro *Matrix Computations* [24].

Teorema 2 *Seja duas matrizes $\mathbf{A} \in \mathbb{R}^{m \times n}$ e $\mathbf{B} \in \mathbb{R}^{m \times n}$ e seja $r = \text{posto}([\mathbf{A}^T, \mathbf{B}^T]^T) \leq n$, a decomposição em valores singulares generalizada (GSVD) conjunta de \mathbf{A} e de \mathbf{B} é dada por (3.21),*

$$\begin{aligned}\mathbf{A} &= \mathbf{U}_\mathbf{A} \mathbf{\Gamma} \mathbf{X}^T \\ \mathbf{B} &= \mathbf{U}_\mathbf{B} \mathbf{\Delta} \mathbf{X}^T\end{aligned}\tag{3.21}$$

em que $\mathbf{U}_\mathbf{A}, \mathbf{U}_\mathbf{B} \in \mathbb{R}^{m \times n}$ e $\mathbf{X} \in \mathbb{R}^{n \times n}$ possuem colunas ortonormais e $\mathbf{\Gamma}, \mathbf{\Delta} \in \mathbb{R}^{n \times n}$ são matrizes diagonais que obedecem à relação (3.22).

$$\mathbf{\Gamma}^2 + \mathbf{\Delta}^2 = \mathbf{I}\tag{3.22}$$

Os valores singulares da decomposição generalizadas são dados pela relação (3.23),

$$\sigma_i = \frac{\gamma_{ii}}{\delta_{ii}}\tag{3.23}$$

em que γ_{ii} e δ_{ii} são elementos das matrizes diagonais $\mathbf{\Gamma}$ e $\mathbf{\Delta}$ respectivamente.

3.6 Decomposição ULLV

A decomposição em valores singulares generalizada permite o tratamento de sinais contaminados com ruídos de qualquer natureza sem a necessidade de uma aplicação explícita

do procedimento de *prewhitening* e *dewhitening*. Entretanto, o custo computacional desta abordagem ainda é bastante alto. Este problema motivou o surgimento de uma aproximação da GSVD denominada ULLVD. Esta decomposição vem sendo bastante aplicada em sistemas que necessitam de uma execução em tempo real mas o hardware disponível não tem capacidade computacional para prover. A ULLVD, introduzida por *Luk et Qiao* em [45], é definida pelo teorema 3:

Teorema 3 *Seja \mathbf{X} e $\mathbf{N} \in \mathbb{R}^{m \times n}$ matrizes de posto completo com $m \geq n$, existem matrizes com colunas ortogonais $\mathbf{U}_{\mathbf{X}} \in \mathbb{R}^{m \times m}$ e $\mathbf{U}_{\mathbf{N}} \in \mathbb{R}^{m \times m}$ e $\mathbf{V} \in \mathbb{R}^{n \times n}$ que satisfazem a relação (3.24),*

$$\begin{aligned}\mathbf{X} &= \mathbf{U}_{\mathbf{X}}\mathbf{L}_{\mathbf{X}}\mathbf{L}\mathbf{V}^T \\ \mathbf{N} &= \mathbf{U}_{\mathbf{N}}\mathbf{L}\mathbf{V}^T\end{aligned}\tag{3.24}$$

em que $\mathbf{L}_{\mathbf{X}}$ e $\mathbf{L} \in \mathbb{R}^{n \times n}$ são matrizes triangulares inferiores.

A decomposição do espaço vetorial em dois subespaços ortogonais leva em consideração a existência de uma diminuição brusca nos valores singulares da matriz $\mathbf{L}_{\mathbf{X}}$, dividindo-a segundo a expressão (3.25). Assim, sendo os valores singulares da decomposição generalizada da matriz $\mathbf{L}_{\mathbf{X}}$ representados por δ_i e os valores singulares de \mathbf{X} por σ_i , as aproximações exibidas em (3.26) são válidas [28].

$$\begin{aligned}\mathbf{X} &= \mathbf{U}_{\mathbf{X}}\mathbf{L}_{\mathbf{X}}\mathbf{L}\mathbf{V}^T \\ \mathbf{X} &= (\mathbf{U}_{\mathbf{X}1}\mathbf{U}_{\mathbf{X}2}) \begin{pmatrix} \mathbf{L}_{\mathbf{X}1} & \mathbf{0} \\ \mathbf{F}_{\mathbf{X}} & \mathbf{G}_{\mathbf{X}} \end{pmatrix} \mathbf{L} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix}\end{aligned}\tag{3.25}$$

$$\begin{aligned}\sigma_i(\mathbf{L}_{\mathbf{X}1}) &\approx \delta_p \\ \|\mathbf{F}_{\mathbf{X}}\|_F^2 + \|\mathbf{G}_{\mathbf{X}}\|_F^2 &\approx \sigma_{p+1}^2 + \dots + \sigma_n^2\end{aligned}\tag{3.26}$$

Esse método, apesar de se tratar de uma aproximação do GSVD, pode fornecer resultados bastante similares aos obtidos através do método original, com um esforço computacional extremamente menor [28]. Uma implementação utilizando a decomposição ULLV que tire vantagens das estruturas Hankel ou Toeplitz das matrizes de entrada pode reduzir o custo computacional de uma ordem $\mathcal{O}(mn \log(n))$ [74] para $\mathcal{O}(mn)$ [28], conforme pode ser visto na seção 3.7.3. Esta redução permite que a escolha da dimensão da matriz de entrada $m \times n$, essencial para o desempenho do algoritmo, possa ser realizada diante de uma menor penalidade no tempo de execução.

3.7 O Algoritmo da decomposição ULLV

O método de subespaços de sinais necessita de uma decomposição ortogonal que possa revelar o posto da matriz. A decomposição ULLV, além de possuir essa característica peculiar, é numericamente estável e possui um ótimo mecanismo de atualização das matrizes resultantes.

Esta decomposição é bastante utilizada em problemas com algoritmos recursivos, tirando vantagem do seu mecanismo de atualização. Primeiramente a decomposição é realizada com um conjunto de dados disponíveis na entrada do sistema, resultando nas matrizes que compõem a equação (3.27).

$$\begin{aligned} \mathbf{X} &= (\mathbf{U}_{\mathbf{x}1}\mathbf{U}_{\mathbf{x}2}) \begin{pmatrix} \mathbf{L}_{\mathbf{x}1} & \mathbf{0} \\ \mathbf{F}_{\mathbf{x}} & \mathbf{G}_{\mathbf{x}} \end{pmatrix} \mathbf{L} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \\ \mathbf{R} &= \mathbf{U}_r \mathbf{L} \mathbf{V}^T \end{aligned} \quad (3.27)$$

em que as matrizes \mathbf{X} e \mathbf{R} representam os dados de entrada e a estimativa da matriz covariância do ruído respectivamente.

Na medida que novos dados são disponibilizados, estes são arranjados nos vetores \mathbf{s}^T e \mathbf{r}^T que representam os dados do sinal ruidoso e da estimativa do ruído respectivamente. O algoritmo apresentado no trabalho de *Peter S. K. Hansen* em [28] necessita, para o processamento de cada linha, que alguns ou todos os passos descritos a seguir sejam executados:

- Adicionando uma Linha (*Updating*): Nesta etapa, uma nova linha é adicionada na matriz de dados \mathbf{X} ou na matriz de ruído \mathbf{R} .
- Removendo uma Linha (*Downdating*): A linha de dados mais antiga da matriz é removida da decomposição.
- Estabilização (*Deflation*): Estabiliza a decomposição de forma a manter a característica de revelar o posto da matriz.
- Refinamento (*Refinement*): Trata-se de um passo opcional para a melhora da qualidade da decomposição. O refino é obtido através da minimização da norma do bloco $\mathbf{F}_{\mathbf{x}}$ da equação (3.27).

A representação dos passos em que uma linha é adicionada ou removida pode ser realizada através das equações (3.28) e (3.29) respectivamente:

$$\begin{aligned}\hat{\mathbf{X}} &= \begin{pmatrix} \mathbf{X} \\ \mathbf{x}^T \end{pmatrix} \\ \hat{\mathbf{X}}^T \hat{\mathbf{X}} &= \mathbf{X}^T \mathbf{X} + \mathbf{x} \mathbf{x}^T\end{aligned}\quad (3.28)$$

$$\begin{aligned}\mathbf{X} &= \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} \\ \hat{\mathbf{X}}^T \hat{\mathbf{X}} &= \mathbf{X}^T \mathbf{X} - \mathbf{x} \mathbf{x}^T\end{aligned}\quad (3.29)$$

em que $\hat{\mathbf{X}}$ representa a matriz após a operação de adição ou remoção de uma linha e \mathbf{X} representa a matriz de dados antes deste processo.

3.7.1 Algoritmo inicial

A parte inicial do algoritmo trata-se da obtenção das matrizes $\mathbf{U}_\mathbf{X}$, $\mathbf{U}_\mathbf{N}$, $\mathbf{L}_\mathbf{X}$, \mathbf{L} e \mathbf{V} necessárias para a decomposição ULLV. Esta decomposição é rerepresentada na equação (3.30) apenas por conveniência.

$$\begin{aligned}\mathbf{X} &= \mathbf{U}_\mathbf{X} \mathbf{L}_\mathbf{X} \mathbf{L} \mathbf{V}^T \\ \mathbf{N} &= \mathbf{U}_\mathbf{N} \mathbf{L} \mathbf{V}^T\end{aligned}\quad (3.30)$$

Primeiramente realiza-se uma decomposição do tipo QR, a qual pode ser vista com mais detalhes em [60] e [52], na matriz \mathbf{N} . Considerando a matriz $\mathbf{V} = \mathbf{I}_n$, através dessa decomposição são obtidas as matrizes $\mathbf{U}_\mathbf{N}$ e \mathbf{L} . Esse passo está representado na equação (3.31).

$$\mathbf{N} = \mathbf{U}_\mathbf{N} \mathbf{L} = \mathbf{U}_\mathbf{N} \mathbf{L} \mathbf{V}^T, \quad \mathbf{V} = \mathbf{I}_n \quad (3.31)$$

Em seguida, uma nova decomposição QR é realizada na matriz \mathbf{Z} . Esta matriz pode ser obtida após algumas manipulações matemáticas na matriz \mathbf{X} que podem ser visualizadas na equação (3.32). Como resultado, as matrizes $\mathbf{U}_\mathbf{X}$ e $\mathbf{L}_\mathbf{X}$ são obtidas.

$$\begin{aligned}\mathbf{X} &= \mathbf{U}_\mathbf{X} \mathbf{L}_\mathbf{X} \mathbf{L} \mathbf{V}^T \\ \mathbf{Z} &= \mathbf{X} (\mathbf{L} \mathbf{V}^T)^{-1} \\ \mathbf{Z} &= \mathbf{U}_\mathbf{X} \mathbf{L}_\mathbf{X}\end{aligned}\quad (3.32)$$

Através destas operações matriciais apresentadas, consegue-se realizar a decomposição inicial. Entretanto, a cada nova linha de dado inserido, deve-se realizar o procedimento de atualização desta decomposição que será explicado a seguir.

Computando o ângulo θ

Como a operação de rotação de planos aplica-se sobre um conjunto de duas linhas, é possível a anulação apenas do elemento q representado nas relações (3.34) e (3.35).

Caso 1:

$$\begin{pmatrix} \cos(\theta) & \text{sen}(\theta) \\ -\text{sen}(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ q \rightarrow 0 \end{pmatrix} \quad (3.34)$$

Caso 2:

$$\begin{pmatrix} \cos(\theta) & \text{sen}(\theta) \\ -\text{sen}(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} q \rightarrow 0 \\ r \end{pmatrix} \quad (3.35)$$

As soluções para o primeiro e segundo casos são ilustrados nas equações (3.36) e (3.37).

$$\begin{aligned} -\text{sen}(\theta)a + \cos(\theta)b &= 0 \\ \tan(\theta) &= \frac{b}{a} \end{aligned} \quad (3.36)$$

$$\begin{aligned} \cos(\theta)a + \text{sen}(\theta)b &= 0 \\ \tan(\theta) &= -\frac{a}{b} \end{aligned} \quad (3.37)$$

3.7.3 Complexidade computacional do algoritmo

Uma das grandes vantagens em se utilizar a decomposição ULLV para revelar o posto da matriz do sinal em contraste da SVD é o menor custo computacional do algoritmo. O custo computacional dos mais conhecidos algoritmos utilizados para decomposição SVD, mesmo quando apenas uma linha é adicionada ou removida da matriz, é da ordem de $\mathcal{O}(mn^2)$, em que m e n são as dimensões da matriz de entrada $\mathbf{A}^{m \times n}$. Recentemente, um algoritmo formulado por *Wei Xu et al* no seu trabalho *A fast symmetric SVD algorithm for square Hankel matrices* [74] reduziu o custo computacional envolvido na decomposição SVD para $\mathcal{O}(mn \log(n))$. Entretanto, a inexistência de mecanismos de atualização das matrizes resultantes da decomposição torna necessário a realização de uma nova decomposição por menor que seja a alteração na matriz de entrada.

O custo computacional envolvido na atualização da decomposição ULLV pode ser facilmente derivado das operações de rotação de plano. Para o cálculo de cada valor do ângulo θ , são necessárias 2 multiplicações, 2 divisões, 1 adição e 1 raiz quadrada. Isto leva o custo computacional envolvido no cálculo de cada novo elemento em $\mathcal{O}(1)$. Assim, ao

se aplicar uma rotação de planos em uma determinada linha de tamanho n , a quantidade de operações realizadas é de ordem $\mathcal{O}(n)$. Realizando este procedimento para toda a matriz $\mathbf{A}^{m \times n}$, leva ao custo computacional necessário para a realização da decomposição de $\mathcal{O}(nm)$. Além disso, essa decomposição possui um mecanismo de atualização das matrizes resultantes, reduzindo o custo computacional envolvido quando uma linha é adicionada ou removida da matriz. Mais detalhes podem ser encontrados no trabalho desenvolvido por *Peter S. K. Hansen* em [28].

3.7.4 Mecanismo de atualização

O mecanismo de atualização da decomposição ULLV é baseado principalmente nas rotações de plano. Como esta rotação trata-se de uma transformação ortogonal, tem-se que $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. Assim, o caso de adicionar uma linha na matrix \mathbf{X} de dados pode ser expressa através da equação (3.38). Esta mesma relação pode ser expandida para o caso de remover uma linha da matriz \mathbf{X} , resultando na expressão (3.39).

$$\hat{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{x}^T \end{pmatrix} = \begin{pmatrix} \mathbf{U}_X & \mathbf{0} \\ \mathbf{0}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T \mathbf{V} \end{pmatrix} \mathbf{V}^T = \hat{\mathbf{U}}_X \hat{\mathbf{L}}_X \hat{\mathbf{L}} \mathbf{V}^T \quad (3.38)$$

$$\mathbf{N} = \begin{pmatrix} \mathbf{U}_N & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{x}^T \mathbf{V} \end{pmatrix} \mathbf{V}^T = \hat{\mathbf{U}}_N \hat{\mathbf{L}} \mathbf{V}^T$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^T \\ \hat{\mathbf{X}} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_X & \mathbf{e}_1 \end{pmatrix} \begin{pmatrix} \mathbf{L}_X & \mathbf{0} \\ \mathbf{0}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \hat{\mathbf{U}}_X \hat{\mathbf{L}}_X \hat{\mathbf{L}} \mathbf{V}^T \quad (3.39)$$

$$\mathbf{N} = \mathbf{U}_N \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{0}^T \end{pmatrix} \mathbf{V}^T = \hat{\mathbf{U}}_N \hat{\mathbf{L}} \mathbf{V}^T$$

em que a linha adicionada e removida em (3.38) e (3.39), respectivamente, é \mathbf{x}^T .

Sendo \mathbf{P} , \mathbf{Q} e \mathbf{J} matrizes ortogonais que aplicam uma rotação de plano nas matrizes \mathbf{V} , \mathbf{L} e \mathbf{L}_X , a relação exposta em (3.40) é válida. Através desta relação pode-se atualizar as matrizes resultantes da decomposição ULLV sem a necessidade de refazer todo o processo de decomposição inicial. Mais detalhes sobre o funcionamento do algoritmo em questão podem ser vistos no trabalho de *Peter S. K. Hansen* em [28]

$$\hat{\mathbf{X}} = (\hat{\mathbf{U}}_X \mathbf{P})(\mathbf{P}^T \hat{\mathbf{L}}_X \mathbf{Q})(\mathbf{Q}^T \hat{\mathbf{L}} \mathbf{J})(\mathbf{J}^T \mathbf{V}^T) \quad (3.40)$$

$$\mathbf{N} = (\hat{\mathbf{U}}_N \mathbf{Q})(\mathbf{Q}^T \hat{\mathbf{L}} \mathbf{J})(\mathbf{J}^T \mathbf{V}^T)$$

3.8 Resumo

O método de subespaços de sinais foi abordado neste capítulo. Primeiramente o funcionamento deste método foi explicado para que o leitor tenha uma visualização do procedimento envolvido para o processo de redução de ruído. Em seguida, o modelo matemático envolvido para o sinal de voz assim como a representação dos dados deste sinal foram detalhados.

Algumas ferramentas matemáticas utilizadas neste método, como o caso do SVD, GSVD e ULLVD foram explicadas. Um procedimento para a utilização deste método com qualquer tipo de ruído, denominado *prewhitening*, foi explicado.

Por fim, o algoritmo envolvido na decomposição ULLV, escolhida para ser utilizada neste trabalho, foi abordado assim como seu mecanismo de atualização. Os conceitos matemáticos fundamentais envolvidos neste algoritmo, como as rotações de plano, assim como a complexidade computacional do algoritmo foram explicados.

A concepção e a implementação do sistema proposto no cenário apresentado na Figura 1.3 serão abordados no próximo capítulo.

Capítulo 4

Implementação do sistema

Os sistemas de reconhecimento de voz sofrem um grande impacto negativo em sua performance quando utilizados em ambientes ruidosos. Assim, torna-se extremamente necessário o estudo de técnicas para a redução de ruído nos sinais de fala com o objetivo de melhorar a taxa de acerto no reconhecimento de palavras.

Neste trabalho foi realizada a implementação de um método de redução de ruídos baseado em subespaços vetoriais através de decomposições matriciais triangulares (ULLVD), o qual vem ganhando bastante atenção nos últimos anos [27]. Foi utilizado como cenário uma simulação de um sistema de reconhecimento de voz acoplado a um automóvel, permitindo assim o controle de diversas funcionalidades através da voz. O aplicativo Julius [38] foi selecionado como sistema de reconhecimento de voz por ser uma alternativa em código aberto e que oferece um ótimo desempenho. A abordagem escolhida para a concepção do sistema é explicada a seguir.

4.1 Abordagem utilizada

No cenário apresentado na Figura 4.1, o usuário, através de um sinal de voz enviado ao dispositivo móvel, pode controlar diversas funcionalidades no automóvel. Dentre estas funções, pode-se citar o controle de temperatura do veículo, o controle do DVD, o levantamento dos vidros entre outros. A identificação dos comandos é realizada pelo sistema de reconhecimento de voz localizado no dispositivo móvel. Quando o sinal de fala na entrada deste sistema possui bastante ruído, sua performance, ou seja, a taxa de identificação correta do comando dado, diminui drasticamente.

Tendo em vista esse problema, um sistema de redução de ruídos foi localizado na entrada do aplicativo de reconhecimento de voz, conforme pode ser visto na Figura 4.2. A entrada desse sistema recebe dados brutos de voz provenientes da conversão A/D realizada no sinal captado pelo microfone. A saída desse sistema também apresenta dados digitais

de voz, embora tenham sido submetidos ao processo de redução de ruído. Devido a esta equivalência entrada/saída, a utilização de um módulo separado localizado na entrada do aplicativo de reconhecimento de voz permite que o sistema de redução de ruídos seja acoplado a qualquer aplicativo de reconhecimento de voz, independente da tecnologia utilizada. Ainda, essa abordagem permite sua posterior implementação em *hardware*, como no caso de um FPGA, ou ainda a utilização de um DSP (Digital Signal Processor) acoplado ao dispositivo móvel. Desta forma, possíveis problemas relacionados ao poder de processamento disponível do dispositivo móvel podem ser eliminados parcialmente ou até em sua totalidade.

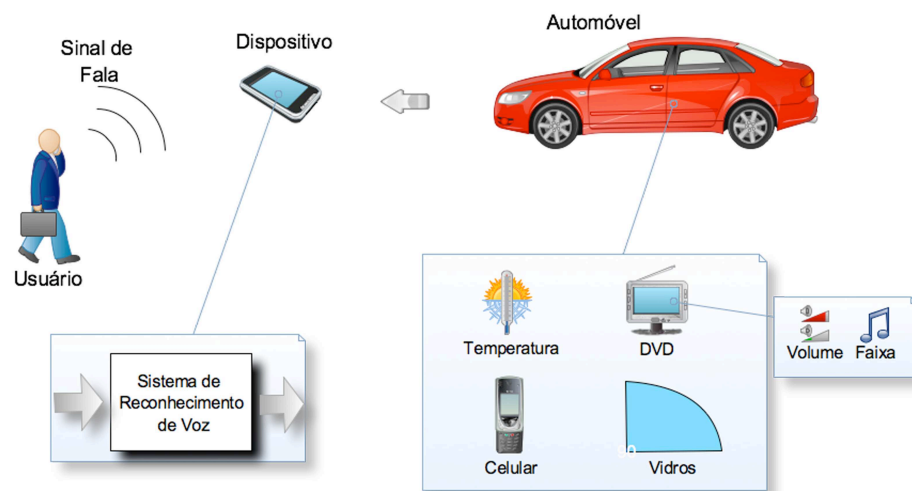


Figura 4.1: Cenário escolhido

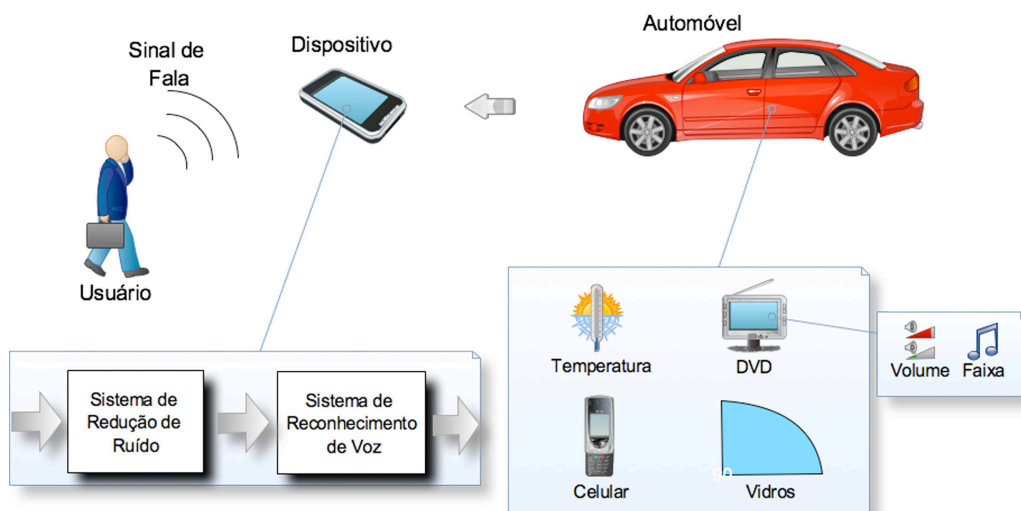


Figura 4.2: Abordagem utilizada

4.2 Metodologia utilizada

A metodologia utilizada para a implementação do sistema foi baseada primeiramente em um modelo de verificação feito em Matlab seguido por uma implementação desse modelo em linguagem C. Seguindo esta linha, as atividades realizadas podem ser vistas a seguir:

1. Concepção do método de subespaços de sinais para redução de ruído utilizando a decomposição SVD:
 - Implementação de um modelo de verificação em Matlab com a finalidade de validar o método implementado em C;
 - Implementação do método de subespaços de sinais utilizando a linguagem C.
2. Concepção do método de subespaços de sinais para redução de ruído utilizando a decomposição ULLV:
 - Implementação de um modelo de verificação em Matlab com a finalidade de validar o método implementado em C;
 - Implementação do método de subespaços de sinais utilizando a linguagem C.
3. Configuração do aplicativo de reconhecimento de voz Julius para a utilização no cenário proposto:
 - Geração de um modelo acústico para a utilização no Julius;
 - Elaboração e implementação de uma gramática para a aplicação pretendida.
4. Elaboração do banco de dados de voz para a avaliação do método de redução de ruído:
 - Gravação dos comandos descritos na gramática criada;
 - Adição de ruídos característicos de um ambiente automotivo às amostras sob diferentes relações sinal/ruído.
5. Avaliação do método de subespaços de sinais:

A primeira etapa do sistema envolveu a utilização da decomposição em valores singulares (SVD) no método de subespaços de sinais. Para a implementação do modelo de verificação em Matlab, a biblioteca de cálculo com matrizes interna do Matlab assim como a função *svd* foram utilizadas. Diante deste fato, foi necessário o desenvolvimento de uma biblioteca própria para cálculos com matrizes na implementação em Linguagem

C. Nesta biblioteca estão disponíveis métodos que vão desde operações elementares com matrizes como transposta e soma, até operações mais complexas como multiplicações e inversas. A rotina utilizada para o cálculo da decomposição *SVD* foi adaptada de uma versão disponível no livro *Numerical Recipes - The Art of Scientific Computing* [52], na qual o algoritmo foi modificado para trabalhar com matrizes não quadradas e reescrito na linguagem C.

Na segunda etapa, a decomposição em valores singulares foi substituída pela decomposição matricial triangular ULLV. O modelo de verificação em Matlab foi implementado utilizando a ferramenta para decomposições matriciais *UTV Tools* [18] e o seu pacote de expansão, disponibilizados gratuitamente no site do autor [30]. Em seguida, o método de subespaços de sinais utilizando a decomposição ULLV foi implementado em C e posteriormente validado pelo modelo de verificação produzido no Matlab. Com o término dessa implementação, algumas otimizações algorítmicas foram feitas com o intuito de diminuir o custo computacional necessário para a execução do método. Estas otimizações levaram em consideração características das matrizes para realizar algumas operações básicas como o cálculo da inversa de uma matriz triangular. Além disso, uma normalização dos dados de entrada foi realizada com o intuito de facilitar uma implementação em ponto-fixado.

O sistema de reconhecimento de voz escolhido foi o Julius [38] por se tratar de um aplicativo em código aberto e oferecer um ótimo desempenho. Apesar deste aplicativo de reconhecimento de voz ter sido concebido utilizando um banco de dados de voz no idioma japonês, a tecnologia utilizada para a identificação das palavras é independente do idioma. Tendo isso em mente, foi realizada uma busca por um banco de dados de voz com uma boa qualidade em português mas infelizmente não foi encontrada nenhuma solução gratuita. O idioma inglês, entretanto, fornecia bancos de dados gratuitos com uma boa qualidade. Além disso, por existir uma maior facilidade em encontrar falantes deste idioma para a gravação do banco de dados de voz para a realização dos testes, esse idioma foi o escolhido (vale salientar que não foram utilizados falantes nativos do idioma inglês, penalizando em parte o desempenho do sistema). Assim, foi utilizado um modelo acústico em inglês disponível gratuitamente no VoxForge [67]. O Voxforge é um programa que coleta dados de voz com suas respectivas transcrições em diversos idiomas. Esses dados servem como uma alternativa gratuita para a criação de modelos acústicos para a utilização em sistemas de reconhecimento de voz. Após a definição do modelo acústico para o Julius, uma gramática foi criada levando em consideração possíveis comandos para o cenário escolhido.

Para a realização dos testes, diversas amostras de voz foram gravadas em formato *wav* com sentenças correspondendo aos comandos definidos na gramática. Essas gravações seguiram um padrão de qualidade no qual buscou-se a obtenção de amostras com ruídos

desprezíveis. Essas amostras ainda teriam que possuir uma qualidade suficientemente boa para que o Julius identificasse o resultado correto de cada sentença no primeiro passo de seu algoritmo. Em seguida, a ferramenta FaNT, desenvolvida pelo grupo AURORA [22] do Instituto ETSI-STQ [63], foi utilizada para adicionar ruídos característicos de um ambiente automotivo em diversas intensidades. Este grupo é responsável pelo desenvolvimento e padronização de algoritmos que tornem sistemas de reconhecimento de voz mais robustos a ruídos. Os ruídos utilizados foram retirados do Projeto Freesound [54], cujo objetivo é o de criar um banco de dados colaborativo de sons disponíveis sob a licença Creative Commons [7].

As amostras geradas infectadas com ruídos em diferentes intensidades foram colocadas novamente como entrada do sistema de reconhecimento de voz. O desempenho foi avaliado ativando e desativando o sistema de redução de ruídos contido no Julius. Desta forma, o impacto do ruído de ambiente sob diferentes condições de SNR assim como a eficácia do método interno de redução de ruído do Julius foram avaliados. Em seguida, estas amostras com ruído foram submetidas ao método de redução de ruídos implementado e novamente analisadas no Julius, ativando e desativando o sistema de redução de ruído interno. Diante disso, não apenas foi avaliado o método de subespaços vetoriais em questão, mas também o comportamento deste quando utilizado em conjunto com o método de subtração espectral implementado no Julius.

4.3 Geração do banco de dados

O processo de gravação dos comandos foi realizado em um ambiente silencioso utilizando um microfone USB *Microsoft Lifechat LX-3000* [47] que oferece um sistema de cancelamento de ruídos no microfone. Tendo em vista este cenário, as amostras gravadas possuem um nível de ruído bastante baixo e são consideradas como amostras sem ruído nesse trabalho, muitas vezes chamadas de amostras limpas. O aplicativo utilizado para a gravação do banco de dados de voz foi o Audacity, disponível gratuitamente em [13]. Durante o processo de gravação foi utilizada uma taxa de amostragem de 8000Hz, suficiente para a representação da voz humana.

Foi utilizado um conjunto de 70 comandos de voz, definidos na Figura 4.6, pronunciados por um locutor do sexo masculino e por um do sexo feminino com idade de 25 anos, totalizando 140 amostras sem ruído. Essas amostras foram contaminadas com ruídos artificiais e reais. Primeiramente, foi gerado artificialmente um ruído com característica branca com a utilização da função *rand*, disponível no Matlab. Em seguida, um ruído colorido foi obtido por um processo de filtragem do ruído branco, no qual buscou-se caracterizar os ruídos ambientes de um automóvel. A função de transferência do filtro utilizado

assim como o comportamento espectral do ruído colorido gerado podem ser vistos na equação (4.1) e na Figura 4.3, respectivamente. Os ruídos reais foram coletados do Projeto FreeSound [54].

$$H(z) = \frac{1}{1 - 0.7z^{-1}} \quad (4.1)$$

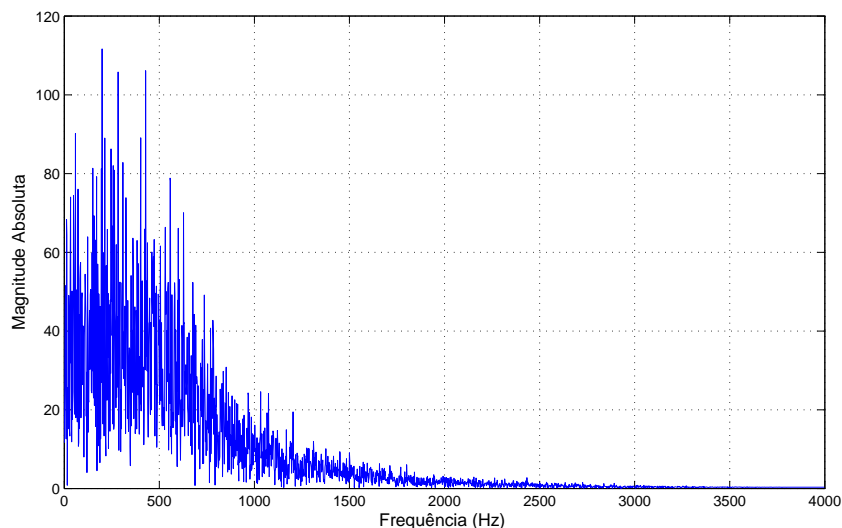


Figura 4.3: **Espectro do ruído colorido gerado artificialmente**

As amostras de ruído foram adicionadas às sentenças sem ruído utilizando a ferramenta FaNT, desenvolvida pelo grupo AURORA [22] do Instituto ETSI-STQ [63]. Foram escolhidos para os testes ruídos relacionados a automóveis parados (apenas o motor ligado) e a automóveis em movimento (ruídos relacionados ao trânsito, motor, pneus e vento). Cada amostra gravada foi contaminada 10 vezes com cada um dos tipos de ruídos selecionados com as seguintes relações sinal/ruído: 30dB, 25dB, 20dB, 15dB, 10dB, 5dB, 0dB e -5dB. Assim, o banco de dados gerado para os testes de desempenho do método de subespaços de sinais possui um conjunto de 44800 amostras.

4.4 Método de subespaços de sinais

4.4.1 Implementação

A implementação do método de subespaços de sinais foi iniciada com a criação de ferramentas básicas para manipulações matriciais. Estas ferramentas, além de fornecer suporte para a realização de cálculos simples, oferece a possibilidade de realizar operações

de decomposição matricial, como o caso da SVD e da ULLVD. Tais decomposições são fundamentais para a implementação do método de subespaço de sinais.

A primeira implementação utilizou a decomposição em valores singulares para revelar a dimensão do subespaço do sinal. Para isso, os valores singulares da matriz de entrada foram ordenados em forma decrescente. A escolha da dimensão do subespaço de interesse é obtida ao se verificar um decaimento brusco entre valores singulares adjacentes. Um fluxograma do algoritmo utilizado nessa implementação pode ser visto na Figura 4.4.

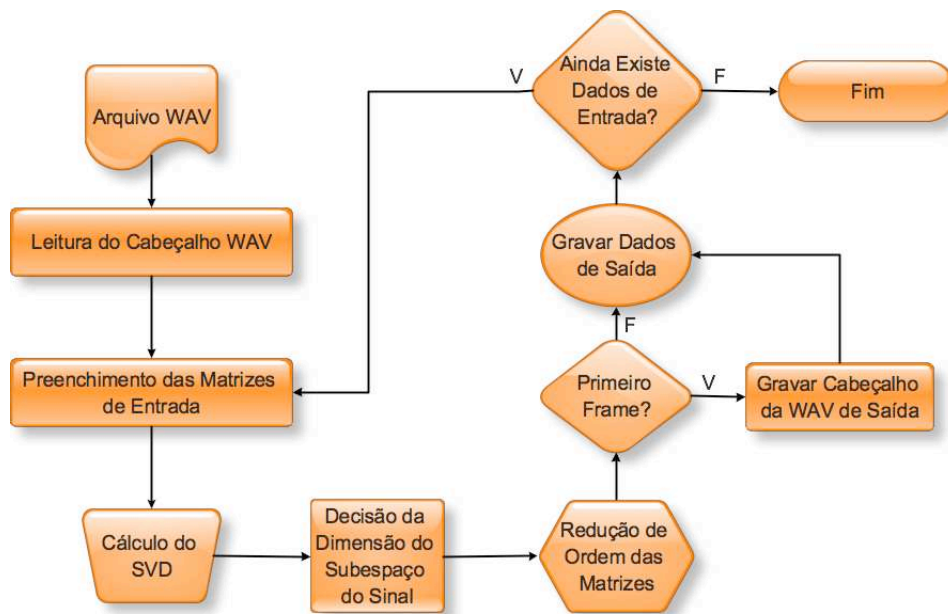


Figura 4.4: Fluxograma do método de subespaços de sinais utilizando SVD

A segunda implementação do método de subespaços de sinais envolveu, no lugar da SVD, a decomposição matricial triangular ULLVD. Esta decomposição também tem como principal função revelar a dimensão do subespaço do sinal. Sua grande vantagem em relação ao SVD é que o custo computacional envolvido para sua execução é da ordem de $O(mn)$, enquanto na utilização dos valores singulares este custo cresce para $O(mn^2)$. Isto se deve principalmente à existência de um mecanismo de atualização da decomposição ULLVD, evitando que a cada janela de dados todo o procedimento envolvido na decomposição seja refeito. Ainda, esta decomposição trata-se de uma aproximação da decomposição GSVD, possibilitando assim o tratamento de sinais contaminados não apenas com ruído branco. Para a identificação da dimensão do subespaço de sinal, estima-se os menores valores singulares da matriz \mathbf{L}_A , resultante da ULLVD, e em seguida compara-se com um determinado limiar. Uma das grandes dificuldades encontradas na implementação deste método foi a escolha correta desse limiar. A solução encontrada para este

problema foi a análise de janelas com presença apenas de ruído, retirando assim informações importantes sobre a energia do ruído. Estas informações são então analisadas e utilizadas para a atribuição desse limiar em tempo de execução. A Figura 4.5 ilustra o fluxograma do algoritmo envolvido nessa implementação.

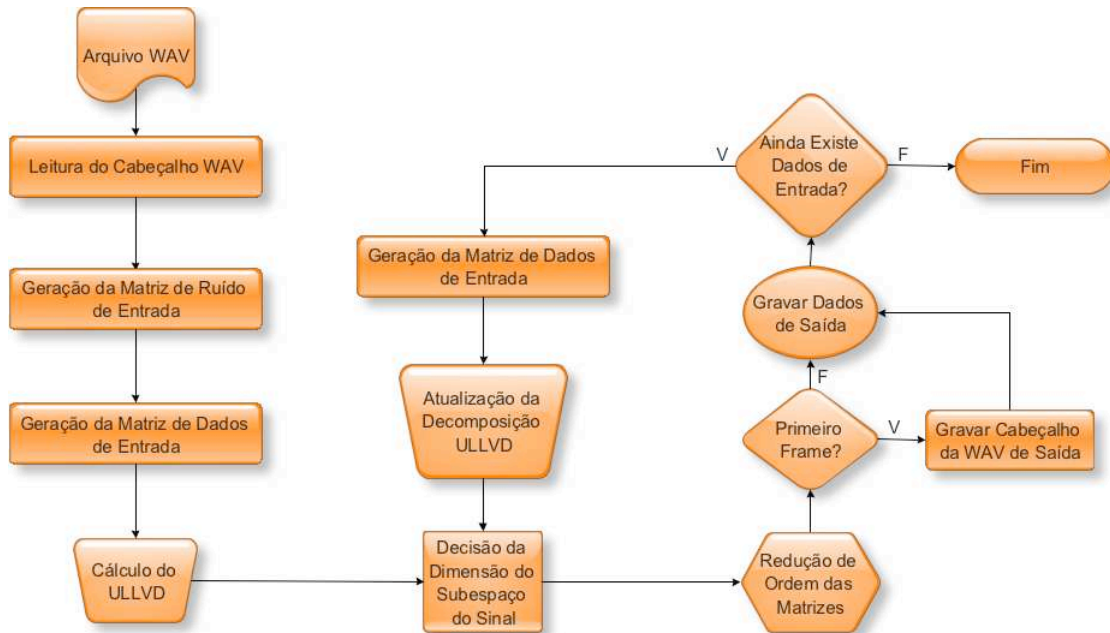


Figura 4.5: Fluxograma do método de subespaços de sinais utilizando ULLVD

4.4.2 Custo Computacional da Implementação

A implementação final do sistema de redução de ruídos utilizou o algoritmo descrito na Figura 4.5. Entretanto, para obter uma melhor análise dessa implementação, é importante avaliar a quantidade de operações em ponto flutuante realizadas por ciclo do algoritmo, ou seja, até a geração de dados de saída. Assim, passos como a atualização das matrizes resultantes da decomposição ULLV assim como o cálculo inicial da mesma devem ser analisados minuciosamente para a avaliação do custo computacional da implementação.

Com base nessa necessidade, o cálculo da quantidade de operações em ponto flutuante realizadas nos principais métodos do algoritmo foi avaliado. O resultado pode ser visto nas Tabelas 4.1 e 4.2, em que:

- m → Número de colunas das matrizes de entrada;
- n → Número de linhas das matrizes de entrada;
- r → Posto da matriz de saída calculado em tempo de execução.

Tabela 4.1: Funções Básicas

Função	Quantidade de operações em ponto flutuante
Norma de Frobenius	$m^3 + m + 1$
Matriz Inversa	$\frac{1}{2}(n^2m + 2nm^2 + nm)$
CCVL	$\frac{1}{2}(3m^4 - 11m^2 + 38m - 60)$
Decomposição QR	$\frac{1}{2}(-2m^2 + 12mn + 8mn^2 - 4nm^2 + 10m + 2 + 8 \sum_{k=0}^{m-2} (n-k)(m-k-1))$

Tabela 4.2: Decomposição ULLV

Função	Quantidade de operações em ponto flutuante
Decomposição Inicial	$\frac{1}{2}(6m^4 + 2m^3 - 26m^2 + 62m + 25nm + 17n^2m - 4nm^2 - 24n + 12r^2 + 12r + 24rn + 36rm - 92 + 16 \sum_{k=0}^{m-2} (n-k)(m-k-1))$
Adicionar Linha	$\frac{1}{2}(3r^4 + 2r^3 + 77r^2 + 11r + 48rn - 46n - 76)$
Remover Linha	$\frac{1}{2}(3r^4 + 2r^3 + 73r^2 + 34r + 80nr - 22n - 74)$
Refinar Decomposição	$6r^2 + 6r + 12rn - 12n + 18rm - 18m + 9$

4.4.3 Escolha dos parâmetros

O sistema implementado realiza o processamento do sinal de voz através do mecanismo de janelamento do sinal de entrada. A escolha do tamanho da janela utilizada para a representação desse sinal é crucial para o funcionamento do método. Quanto maior este tamanho, melhor serão as estimativas das matrizes de covariância dos dados de entrada [3]. Entretanto, um sinal de fala só pode ser considerado um processo estacionário em sentido amplo (WSS) quando a duração da janela está entre $10ms$ e $30ms$. Ainda, quanto maior o tamanho da janela utilizada, maior será o custo computacional necessário para a execução do método. Assim, considerando a taxa de amostragem de $8kHz$ do sinal de entrada, foi definido que cada janela de processamento contém 240 amostras, totalizando um tamanho de janela de $30ms$.

O processamento dos dados no método de subespaços de sinais é realizado através de matrizes de dimensão $m \times n$, em que m indica o número de linhas e n o número de colunas. Os algoritmos normalmente utilizados neste método consideram cada linha dessas matrizes uma realização do experimento (neste caso a fala). Para que seja possível que as

informações necessárias de cada subespaço seja revelada, é necessário que a quantidade de experimentos seja muito maior que suas dimensões, ou seja, $m \gg n$ [28]. Entretanto, quanto maior for a dimensão de cada linha, a energia do ruído poderá se espalhar por mais dimensões enquanto que a do sinal continuará confinada ao subespaço do sinal, possibilitando uma maior redução de ruído sem demasiada distorção do sinal. Deve-se levar em consideração ainda que quanto maior for a dimensão de cada linha, maior será o custo computacional envolvido no processo de redução de ruído. Uma boa relação custo/benefício encontrada de forma eurística foi a utilização de matrizes de dimensões 210×31 para a representação de cada janela de dados.

A estimação da dimensão do subespaço do sinal é um outro problema que deve ser resolvido para a implementação do método de subespaços de sinais. Uma estimação mal realizada deste nível resulta em uma maior distorção do sinal de voz ou uma maior quantidade de ruído presente na saída. A solução encontrada foi a consideração, que é bastante razoável, de que a primeira janela de dados contém apenas informações referente ao ruído. Assim, realiza-se uma decomposição SVD deste sinal e, a partir dos seus valores singulares, define-se o limiar do subespaço do sinal. Os resultados obtidos são bastante promissores pois fornecem uma saída com uma melhor qualidade em troca de um pequeno acréscimo no custo computacional necessário para sua execução.

4.5 Configuração do Julius

O aplicativo de reconhecimento de voz Julius possui uma grande quantidade de opções de configuração que afetam diretamente no desempenho do sistema. Assim, diversos aspectos característicos do sistema foram analisados para a configuração desse aplicativo.

4.5.1 Modelo acústico e gramática

Para a utilização do Julius no cenário proposto, primeiramente foram definidos um modelo acústico e uma gramática para o mesmo. Conforme dito anteriormente, o primeiro representa estatisticamente os fonemas e o segundo estabelece uma série de comandos para a aplicação.

A definição do modelo acústico utilizou dados retirados do VoxForge [67], que trata-se de uma organização que recolhe dados de voz em vários idiomas e suas transcrições para a criação de modelos acústicos gratuitos. Entretanto, o único banco de dados com uma qualidade aceitável até o momento na sua página web está no idioma inglês. Assim, foi utilizado um modelo acústico para o inglês criado através de amostras de voz com taxa de amostragem de 8kHz. Esta taxa de amostragem foi escolhida para garantir uma

boa representação dos fonemas seja obtida, uma vez que as sentenças de entrada foram gravadas com essa mesma taxa.

A primeira etapa envolvida na criação da gramática para o cenário escolhido foi a elaboração de uma pequena lista de funcionalidades que pudessem ser controladas através da voz (Vidros, Som, CD, Ar Condicionado). Em seguida, foi definido, de forma arbitrária, que qualquer comando dado ao sistema deveria ser composto por apenas 3 palavras. Como o modelo acústico disponível é para o idioma inglês, todos os comandos estabelecidos estão neste idioma. Esse conjunto de comandos pode ser visto na Figura 4.6. A escolha das palavras utilizadas nos comandos foi realizada com o objetivo de evitar ambiguidades e de manter a simplicidade para que seja de fácil memorização para o usuário final. Além disso, pode-se observar que o comportamento frequencial de cada palavra localizada na primeira posição de cada comando é bastante distinto. Este comportamento pode ser visto na Figura 4.7.





 AC	Plus	One	Six	Eleven	Sixteen
	Minus	Two	Seven	Twelve	Seventeen
		Three	Eight	Thirteen	Eighteen
Turn	Four	Nine	Fourteen	Nineteen	
	Five	Ten	Fifteen	Twenty	
 CD	Track	One	Six	Eleven	Sixteen
		Two	Seven	Twelve	Seventeen
		Three	Eight	Thirteen	Eighteen
		Four	Nine	Fourteen	Nineteen
		Five	Ten	Fifteen	Twenty
Turn	On	Off			
 Radio	Turn	On	Off		
 Window	Left	Open	Close		
	Right				

Figura 4.6: Diagrama de comandos

Para a definição da gramática utilizada no Julius, foi necessária a implementação da especificação para a criação de uma gramática para este aplicativo. Este documento, disponível gratuitamente em [38], especifica a criação dos arquivos ".voca" e ".grammar" necessários para a sua especificação. Uma definição formal da gramática utilizada pode ser vista na Definição 4. Mais informações sobre definições de gramáticas em linguagem formal podem ser vistas no livro *Introduction to Automata Theory, Languages, and Computation* [33].

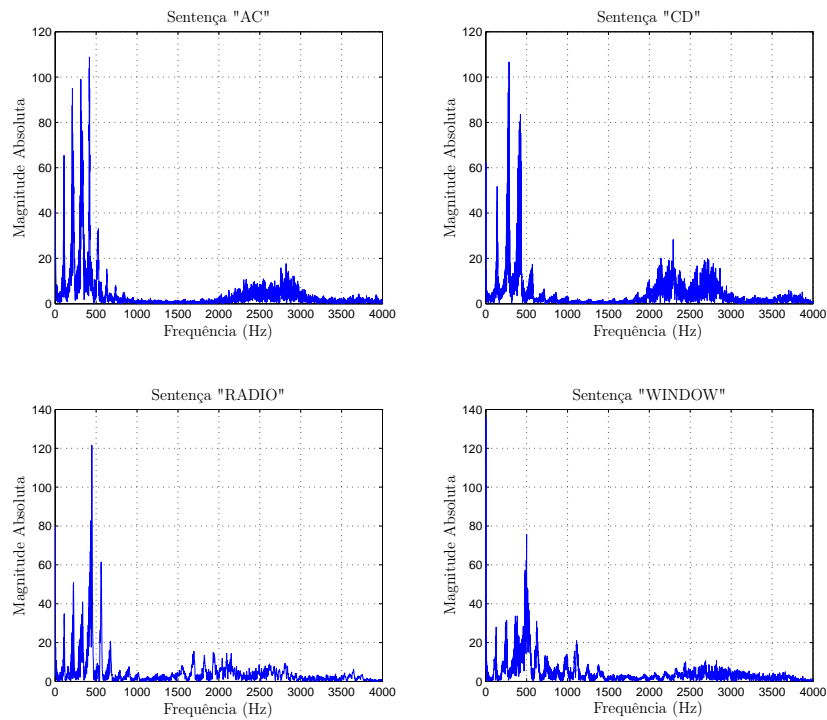


Figura 4.7: Comportamento frequencial das primeiras palavras dos comandos definidos na gramática

Definição 4 Seja V_n o conjunto dos símbolos não terminais de uma gramática G , V_t os símbolos terminais, P o conjunto de regras de produção e S o símbolo inicial, a gramática das sentenças da Figura 4.6 pode ser definida por $G = \{V_n, V_t, P, S\}$, em que:

$$V_n = \{A, B, C, D, E, S\}$$

$V_t = \{ac, cd, radio, window, plus, minus, left, right, open, close, turn, on, off, track, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty\}$

$$P = \{$$

$$1. S \rightarrow ac A \mid ac E \mid cd track C \mid cd E \mid radio E \mid window B D$$

$$2. A \rightarrow plus C \mid minus C$$

$$3. B \rightarrow left \mid right$$

$$4. C \rightarrow one \mid two \mid three \mid four \mid five \mid six \mid seven \mid eight \mid nine \mid ten \mid eleven \mid twelve \mid thirteen \mid fourteen \mid fifteen \mid sixteen \mid seventeen \mid eighteen \mid nineteen \mid twenty$$

$$5. D \rightarrow open \mid close$$

$$6. E \rightarrow turn on \mid turn off \quad \}$$

4.5.2 Opções de configuração

O Julius oferece suporte a utilização de um arquivo de configuração de entrada que carrega todas as opções necessárias para sua execução. A explicação sobre cada opção disponível pode ser encontrada em [43]. Algumas opções utilizadas no cenário proposto podem ser vistas a seguir:

- *-input rawfile* → utiliza como entrada padrão arquivos de áudio sem compressão (wav);
- *-zmean* → remove o nível DC da entrada padrão;
- *-nostrip* → desabilita a opção com a qual o Julius remove sequências de zero contidas na entrada;
- *-gram* → define uma gramática para utilização durante o etapa de reconhecimento de voz;
- *-h* e *-hlist* → estes comandos definem o modelo acústico utilizado;
- *-spmodel "sp"* → define o nome do modelo para o silêncio;
- *-iwcd1 max* → utiliza um método de aproximação de trifones com critério de máxima verossimilhança;
- *-smpFreq 8000* → estabelece a taxa de amostragem como sendo 8kHz.
- *-sscalc* → habilita o método de subtração espectral para a remoção de ruídos contidos no sinal de entrada;
- *-sscalclen 50* → define que os primeiros 50ms são considerados como apenas ruído. Esta informação é utilizada na estimação do espectro do ruído necessário para o procedimento de subtração espectral.

Capítulo 5

Resultados obtidos

Neste capítulo serão apresentados os resultados obtidos com o método de redução de ruído implementado. O objetivo deste método é tornar o sistema de reconhecimento de voz mais robusto, ou seja, que mantenha sua taxa de acerto no reconhecimento de palavras mesmo quando submetido a ambientes ruidosos. Assim, as avaliações do método não levaram em consideração o fator inteligibilidade das amostras de voz para o ser humano, apenas para o sistema de reconhecimento de palavras.

O banco de dados de voz utilizado para os testes de reconhecimento de voz com o aplicativo Julius foi duplicado devido à adição das amostras resultantes do método de subespaços de sinais, totalizando 89600 amostras. Em seguida, o sistema de redução de ruídos contido nesse aplicativo (subtração espectral) foi testado para ser obtido uma comparação com o método de subespaços de sinais. Também foi avaliado o comportamento do método implementado em conjunto com o método de subtração espectral. Os resultados obtidos para cada tipo de ruído contaminante podem ser vistos a seguir.

5.1 Avaliações com ruído branco

Os resultados obtidos para as amostras contaminadas com ruído branco podem ser vistos na Tabela 5.1. Conforme indicado pelos resultados apresentados na Tabela 1.1, o desempenho do sistema de reconhecimento de voz é bastante degradado com a presença de ruídos. Ainda, analisando a Tabela 5.1, é possível verificar uma superioridade na maior parte dos casos nos resultados obtidos com o método de subespaços de sinais comparado com o método de subtração espectral. A utilização dos dois métodos analisados forneceram os melhores resultados e se colocam como uma importante combinação para a redução de ruído em sinais de voz. Entretanto, para algumas relações sinal/ruído, essa combinação apresenta resultados inferiores aos obtidos com a utilização do método de subespaços de sinais isolado. O motivo para isto é a distorção inserida durante os processos

de redução de ruído, impedindo assim a correta identificação da palavra pelo sistema de reconhecimento de voz.

Tabela 5.1: Resultados para as amostras contaminadas com ruído branco gaussiano

Relação sinal/ruído	Sem redução de ruído	Com subespaços de sinais	Com subtração espectral	Com subespaços vetoriais e subtração espectral
-5dB	11,00%	16,00%	13,97%	27,28%
0dB	25,08%	28,43%	33,00%	56,00%
5dB	39,05%	56,00%	42,43%	69,00%
10dB	52,06%	67,57%	55,43%	81,00%
15dB	56,98%	80,72%	63,28%	87,28%
20dB	66,98%	89,72%	72,43%	86,71%
25dB	72,22%	92,57%	77,29%	93,00%
30dB	74,29%	96,28%	80,57%	96,43%

5.2 Avaliações com ruído colorido

Os resultados obtidos para as amostras contaminadas com ruído colorido, obtido através de um processo de filtragem de um ruído gaussiano através do filtro definido na equação (4.1), podem ser vistos na Tabela 5.2. Este tipo de ruído ocasionou uma degradação mais intensa no desempenho do sistema de reconhecimento de voz quando comparado com o ruído branco. Isto se deve ao fato do processo de *prewhitening* não garantir o espalhamento uniforme da energia do sinal de ruído por todas as dimensões do espaço vetorial. O método de subespaços de sinais e a combinação deste com o método de subtração espectral mais uma vez foram os que ofereceram os melhores resultados.

5.3 Avaliações com ruído de um carro parado

Os primeiros testes realizados com ruídos reais utilizaram ruídos de um carro parado com o motor ligado. Os resultados obtidos para as amostras contaminadas com este tipo de ruído podem ser encontrados na Tabela 5.3. Analisando esta tabela, verifica-se um declínio mais intenso que nos casos anteriores no desempenho do sistema de reconhecimento de voz. O método de subespaços de sinais, apesar de não utilizar qualquer modelo de ruído, mostrou-se bastante robusto para este tipo de sistema. A combinação dos métodos de subespaços de sinais e de subtração espectral apresentou desempenho superior ao obtido com o primeiro método para todas as relações sinal/ruído.

Tabela 5.2: Resultados para as amostras contaminadas com ruído colorido artificial

Relação sinal/ruído	Sem redução de ruído	Com subespaços de sinais	Com subtração espectral	Com subespaços vetoriais e subtração espectral
-5dB	8,71%	14,92%	15,29%	18,73%
0dB	19,71%	26,67%	25,24%	35,57%
5dB	37,14%	44,71%	41,27%	56,86%
10dB	43,18%	56,00%	46,03%	68,86%
15dB	52,22%	67,86%	59,52%	76,14%
20dB	60,79%	76,57%	67,46%	80,43%
25dB	67,78%	83,72%	74,13%	86,28%
30dB	70,48%	89,96%	76,83%	91,00%

Tabela 5.3: Resultados para as amostras contaminadas com ruído real de um carro parado com o motor ligado

Relação sinal/ruído	Sem redução de ruído	Com subespaços de sinais	Com subtração espectral	Com subespaços vetoriais e subtração espectral
-5dB	6,51%	33,43%	31,14%	34,00%
0dB	17,86%	42,86%	39,52%	44,00%
5dB	32,06%	54,86%	53,43%	61,71%
10dB	48,89%	69,71%	61,57%	75,57%
15dB	56,67%	82,57%	69,86%	87,71%
20dB	69,51%	91,14%	71,57%	94,43%
25dB	71,59%	96,71%	75,00%	97,28%
30dB	74,76%	97,43%	78,14%	99,29%

5.4 Avaliações com ruído de um carro em movimento

As últimas simulações utilizaram um ruído de um carro em movimento com as janelas abertas e com tráfego nas ruas. Este tipo de ruído foi utilizado para caracterizar os ruídos mais comuns encontrados em ambientes automotivos e comprovar a eficácia do método escolhido para este cenário. Os resultados obtidos para as amostras contaminadas com ruído de carro em movimento podem ser vistos na Tabela 5.4. Analisando esta tabela, pode-se constatar que o método de subespaços de sinais teve um desempenho menor que o obtido nos outros casos. Isso se deve à natureza não estacionária deste tipo de ruído. O ganho obtido com a combinação dos dois métodos para este caso não obteve melhoras significativas para casos com uma alta relação sinal/ruído. Entretanto, houve uma grande melhora para a maioria dos casos no qual o sinal de voz estava bastante degradado.

Tabela 5.4: Resultados para as amostras contaminadas com ruído real de um carro trafegando em uma rua movimentada com as janelas abertas

Relação sinal/ruído	Sem redução de ruído	Com subespaços de sinais	Com subtração espectral	Com subespaços vetoriais e subtração espectral
-5dB	1,57%	6,82%	4,11%	5,08%
0dB	6,00%	10,18%	12,38%	20,48%
5dB	18,75%	26,43%	21,59%	44,92%
10dB	24,64%	47,94%	47,86%	66,19%
15dB	30,18%	66,14%	54,92%	73,65%
20dB	40,54%	72,86%	66,67%	76,35%
25dB	56,79%	76,86%	72,54%	82,06%
30dB	63,75%	84,72%	78,68%	85,08%

5.5 Resumo

Neste capítulo os resultados obtidos durante as simulações envolvendo o método de subespaços de sinais e o sistema de reconhecimento Julius foram abordados. Foram avaliados o desempenho do Julius quando submetido a ruídos característicos de um ambiente automotivo assim como quando submetido a um ruído branco gaussiano. Para esses testes, foram utilizados ruídos criados artificialmente, casos do ruído branco e do ruído colorido, e ruídos gravados de um ambiente real, ruído de um carro parado com o motor ligado e trafegando em uma rodovia movimentada com as janelas abertas.

No próximo capítulo as conclusões referentes aos resultados obtidos com o método de subespaços de sinais assim como com a combinação deste método com o de subtração espectral serão detalhados.

Capítulo 6

Conclusões e trabalhos futuros

Neste trabalho foi realizado um estudo do método de subespaços de sinais para redução de ruído em sinais de voz. O cenário levado em consideração foi o de um automóvel no qual diversas funcionalidades são controladas por comandos de voz.

Os resultados obtidos demonstram uma melhora significativa no desempenho do sistema de reconhecimento de voz analisado através da utilização de uma etapa de pré-processamento do sinal para a redução de ruído. O método de subespaços de sinais se demonstrou superior à subtração espectral independente da relação sinal/ruído das amostras. O principal motivo para este resultado é a característica deste método, apesar de causar deformações no sinal de voz como todos os outros métodos, manter uma grande parte da energia das formantes do sinal de fala na saída. Com isso, a relação sinal/ruído é melhorada assim como o desempenho do sistema de reconhecimento de voz.

O desempenho do sistema de redução de ruído implementado se mostrou bastante robusto para diversos tipos de ruído. Isto se deve ao fato do método de subespaços de sinais não utilizar um modelo predeterminado de ruído, mas apenas suposições coerentes sobre a natureza deste ruído.

O conceito utilizado no método de subespaços de sinais no qual a voz contaminada com ruído é projetada no subespaço vetorial de sinal é bastante utilizado para a remoção de parte da influência negativa causada pelos ruídos. Entretanto, métodos clássicos como a subtração espectral, avaliada também neste trabalho, complementam este método oferecendo uma maior remoção de ruído no sinal de saída. Desta forma, combinações como esta oferecem uma melhora significativa nos resultados obtidos com o método de subespaços de sinais e são também uma importante área de estudo para o aumento da robustez de um aplicativo de reconhecimento de voz.

Um dos grandes desafios enfrentados para a implementação do método de subespaços de sinais para a remoção de ruídos em aplicações que exigem uma resposta em tempo real é a complexidade algorítmica deste método. A utilização da decomposição ULLV,

em conjunto com seu mecanismo de atualização, reduz esta complexidade de uma ordem $\mathcal{O}(mn^2)$ para $\mathcal{O}(mn\log(n))$ para matrizes de entrada de ordem $m \times n$, conforme apresentado na seção 3.7.3. Com a utilização dessa decomposição, o tempo de execução da etapa de redução de ruído ficou da mesma ordem do tempo gasto na etapa de reconhecimento de voz.

Como trabalho futuro é sugerido uma implementação do método de subespaços em um sistema dedicado em FPGA com o objetivo de se obter uma melhora no tempo de resposta do sistema. Desta forma, o método de subespaços de sinais se torna um importante candidato, dentre as técnicas existentes para a redução de ruído, em aplicações que necessitem de respostas em tempo real, como é o caso do cenário escolhido neste trabalho.

Referências bibliográficas

- [1] H. Abut, J. H. L. Hansen, and K. Takeda. *DSP for In-vehicle and Mobile Systems*, volume 1. Springer, 2005.
- [2] E. Beltrami. Sulle Funzioni Bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11:98–106, 1873.
- [3] J. Benesty, S. Makino, and J. Chen. *Speech Enhancement*, volume 1 of *Springer Series on Signals and Communication Technology*. Springer, 2005.
- [4] J. Benesty, M. M. Sondhi, and Y. Huang. *Springer Handbook of Speech Processing*. Springer, 1 edition, 2007.
- [5] A. W. Bojanczyk and J. M. Lebak. DOWNDATING A ULLV DECOMPOSITION OF TWO MATRICES. In *Conf. Rec of LA'94*, pages 261–265, Junho 1994.
- [6] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113 – 120, 1979.
- [7] Creative Commons Brasil. <http://www.creativecommons.org.br>, Acessado em 03 de Março de 2009.
- [8] J. J. Christensen and J. Hald. Beamforming. Technical Report 1, Brüel & Kjær Sound & Vibration Measurements A/S, 2004. Disponível em: <http://www.bksv.com/Library/Technical%20Reviews.aspx>, Acessado em 06 de Abril de 2009.
- [9] A. C. Clarke. *2001: A Space Odyssey*. New American Library, 1968.
- [10] N. Dal Degan and C. Prati. Acoustic noise analysis and speech enhancement techniques for mobile radio applications. *Signal Processing*, 15(1):43 – 56, 1988.
- [11] M. Dendrinos, S. Bakamidis, and G. Carayannis. Speech enhancement from noise: A regenerative approach. *Speech Communication*, 10(1):45–57, 1991.
- [12] P. Divenyi. *Speech Separation by Humans and Machines*, volume 1. Kluwer Academic Publishers, 2005.

- [13] Audacity: Free Audio Editor and Recorder. <http://audacity.sourceforge.net>, Acessado em 04 de Fevereiro de 2009.
- [14] C. Englund. Speech recognition in the jas 39 gripen aircraft - adaptation to speech at different g-loads. Master's thesis, KTH Speech, Music and Hearing, Março 2004.
- [15] N. Fan, R. Balan, and J. Rosca. Comparison of wavelet and FFT based single channel speech signal noise reduction techniques. In *Conf. Rec. of SPIE*, 2004.
- [16] FIAT. <http://www.fiat.com>, Acessado em 29 de Janeiro de 2009.
- [17] R. D. Fierro and P. C. Hansen. Low-rank revealing UTV decompositions. *Numerical Algorithms*, 15(1):37–55, 1997.
- [18] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen. UTV Tools: Matlab templates for rank-revealing UTV decompositions. *Numerical Algorithms*, 20:165–194, 1999.
- [19] J. V. Filho. *Redução de Ruído em Sinais de Voz nos Sistemas de Rádio Móveis Veiculares*. PhD thesis, UNICAMP, Agosto 1996.
- [20] D. G. Fish. The influence of in-vehicle noise on speech recognition for automotive voice-activated control systems. *SAE Transactions*, 110(6):1863 – 1867, 2001.
- [21] M. Forsberg. Why is speech recognition difficult? Technical report, Chalmers University of Technology, Fevereiro 1993. Disponível em: http://www.speech.kth.se/~rolf/gslt_papers/MarkusForsberg.pdf, Acessado em 06 de Abril de 2009.
- [22] AURORA Speech Recognition Experimental Framework. <http://aurora.hsnr.de>, Acessado em 28 de Janeiro de 2009.
- [23] S. Goetze, V. Mildner, and K. D. Kammeyer. Comparison of speech enhancement systems for noise fields in a car environment. *FORTSCHRITTE DER AKUSTIK*, 2006.
- [24] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition edition, 1996.
- [25] Y. Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 1995.
- [26] M. M. Gouling and J. S. Bird. Speech enhancement for mobile telephony. *IEEE Transactions on Vehicular Technology*, 39(4):316 – 326, 1990.

- [27] P. C. Hansen and S. H. Jensen. Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis. *EURASIP Journal on Advances in Signal Processing*, 2007.
- [28] P. S. K. Hansen. *Signal subspace methods for speech enhancement*. PhD thesis, Technical University of Denmark, Setembro 1997.
- [29] P. S. K. Hansen, P. C. Hansen, S. D. Hansen, and J. A. Sørensen. Experimental comparison of signal subspace based noise reduction methods. In *Conf. Rec. of ICASSP'99*, volume 1, pages 101–104, Março 1999.
- [30] Per Christian Hansen. <http://www2.imm.dtu.dk/pch>, Acessado em 11 de Dezembro de 2008.
- [31] K. Hermus, P. Wambacq, and H. Van Hamme. A review of signal subspace speech enhancement and its application to noise robust speech recognition. *EURASIP Journal on Advances in Signal Processing*, 2007.
- [32] J. Holmes and W. Holmes. *Speech Synthesis and Recognition*, volume 2. Taylor & Francis, 2001.
- [33] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 2nd edition edition, 2001.
- [34] Hidden Markov Model Toolkit (HTK). <http://htk.eng.cam.ac.uk>, Acessado em 26 de Fevereiro de 2009.
- [35] S. Van Huffel and H. Zha. An efficient total least squares algorithm based on a rank-revealing two-sided orthogonal decomposition. *Numerical Algorithms*, 4(1-2):101–133, 1993.
- [36] S. H. Jensen, P. C. Hansen, S. D. Hansen, and J. A. Sørensen. Reduction of broadband noise in speech by truncated QSVD. *IEEE Transactions on Speech and Audio Processing*, 3(6):439 – 448, 1995.
- [37] C. Jordan. Mémoire sur les Formes Bilinéaires. *Journal de Mathématiques Pures et Appliquées*, 19:35–54, 1874.
- [38] Open-Source Large Vocabulary CSR Engine Julius. http://julius.sourceforge.jp/en_index.php, Acessado em 26 de Fevereiro de 2009.
- [39] T. Katayama. *Subspace Methods for System Identification*. Communications and Control Engineering. Springer, 1st edition edition, 2005.

- [40] J. Kim, C. W. Lee, J. K. Lee, J. S. Hwang, and J. H. Park. Comparison of adaptive systems for noise reduction in speech performance. In *Conf. Rec. of SSST'06*, 2006.
- [41] A. Kun, T. Paek, and Z. Medenica. The effect of speech interface accuracy on driving performance. In *Conf. Rec. of INTERSPEECH'07*, pages 1326–1329, Agosto 2007.
- [42] D. C. Lay. *Linear Algebra and Its Applications*. Addison-Wesley Publishing Company, 1rd edition edition, 1994.
- [43] A. Lee. The Julius book. Disponível em: <http://keihanna.dl.sourceforge.jp/julius/37581/Juliusbook-part-4.1.2-en.pdf>, Fevereiro 2009. Acessado em 06 de Abril de 2009.
- [44] C. F. Van Loan. Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13(1):76–83, 1976.
- [45] F. T. Luk and S. Qiao. A new matrix decomposition for signal processing. *Automatica*, 30(1):39–43, 1994.
- [46] F. T. Luk and S. Qiao. Adaptive algorithm for interference canceling in array processing. *Advanced Signal Processing Algorithms, Architectures and Implementations VI*, 2846:151–161, 1996.
- [47] Microsoft LifeChat LX-3000. <http://www.microsoft.com/hardware/digitalcommunication/productdetails.aspx?pid=006>, Acessado em 18 de Março de 2009.
- [48] M. Moonen and B. de Moor, editors. *SVD and Signal Processing III: Algorithms, Architectures and Applications*. Elsevier, 1st edition edition, 1995.
- [49] M. Moonen, P. Van Dooren, and J. Vandewalle. A singular value decomposition updating algorithm for subspace tracking. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1015–1038, 1992.
- [50] P. Van Overschee and B. de Moor. *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Kluwer Academic Publishers, 1st edition edition, 1996.
- [51] R. Pieraccini, K. Dayanidhi, J. Bloom, J. G. Dahan, and M. Phillips. A multimodal conversational interface for a concept vehicle. In *Conf. Rec. of EUROSPEECH'03*, volume 1, pages 2233–2236, Setembro 2003.

- [52] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes - The Art of Scientific Computing*, volume 3. Cambridge University Press, 2007.
- [53] CMU Sphinx Project. <http://cmusphinx.sourceforge.net/html/cmusphinx.php>, Acessado em 26 de Março de 2009.
- [54] The Freesound Project. <http://www.freesound.org>, Acessado em 03 de Março de 2009.
- [55] H. Sahlin and H. Broman. Signal separation applied to real world signals. In *Conf. Rec. of IWAENC'97*, pages 11–12, Setembro 1997.
- [56] L. L. Scharf. *Statistical Signal Processing*, volume 1. Prentice Hall, 1991.
- [57] Nuance Dragon Naturally Speaking. <http://www.nuance.com/naturallyspeaking>, Acessado em 24 de Março de 2009.
- [58] G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 40(6):1535–1541, 1992.
- [59] G. W. Stewart. Updating a rank-revealing ULV decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, 1993.
- [60] G. W. Stewart. *Matrix Algorithms. Volume I: Basic Decompositions*. SIAM Press, 1998.
- [61] G. W. Stewart. The decompositional approach to matrix computation. *Computer in Science and Engineering*, 2(1):50–59, 2000.
- [62] R. O. Stolfi. Síntese e reconhecimento da fala humana. Master's thesis, UNICAMP, Outubro 2006.
- [63] European Telecommunications Standards Institute's - Speech, Transmission Planning, and Quality of Service (ETSI - STQ). <http://portal.etsi.org/stq/Summary.asp>, Acessado em 28 de Janeiro de 2009.
- [64] D. W. Tufts and R. Kumaresan. Singular value decomposition and improved frequency estimation using linear prediction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(4):671–675, 1982.
- [65] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*, volume 3. John Wiley & Sons Ltd, 2006.

- [66] IBM ViaVoice. <http://www.nuance.com/viavoice>, Acessado em 24 de Março de 2009.
- [67] VoxForge. <http://www.voxforge.org>, Acessado em 03 de Março de 2009.
- [68] R. B. Wallace and R. A. Goubran. Noise cancellation using parallel adaptive filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(4):239 – 243, 1992.
- [69] J. Wang. Computing the CSD and GSVD. Technical report, UC Davis - University of California, Setembro 2004. Disponível em: http://wwwcsif.cs.ucdavis.edu/~wangjj/gsvd/gsvdj_jw3.pdf, Acessado em 06 de Abril de 2009.
- [70] D. S. Watkins. *Fundamentals of Matrix Computations*. Wiley-Interscience, second edition edition, 2002.
- [71] K. White, H. Ruback, J. Sedivy, K. Kojima, and K. Kondo. Advanced development of speech enabled voice recognition enabled embedded navigation systems. *SAE Transactions*, 115:134 – 140, 2006.
- [72] B. Widrow, J. R. Grover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. S. Zeidler, E. Dong, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692 – 1716, Dezembro 1975.
- [73] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. The MIT Press, 1949.
- [74] W. Xu and S. Qiao. A fast symmetric SVD algorithm for square hankel matrices. *Linear Algebra and its Applications*, 428(2-3):550–563, Janeiro 2008.
- [75] C. H. Yang, J. C. Wang, J. F. Wang, C. H. Wu, and K. H. Chang. Design and implementation of subspace-based speech enhancement under in-car noisy environments. *IEEE Transactions on Vehicular Technology*, 57(3), 2008.
- [76] C. A. Ynoguti. *Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov*. PhD thesis, UNICAMP, Maio 1999.

Apêndice A

Fundamentos de álgebra linear

Neste apêndice alguns conceitos básicos de álgebra linear importantes para o método de subespaços de sinais serão revisados. Dentre estes, destacam-se a decomposição SVD e as projeções ortogonais e oblíquas, conceitos nos quais diversos algoritmos de subespaços residem.

A.1 Espaços vetoriais

Um espaço vetorial pode ser definido como um conjunto \mathbb{V} de vetores considerados fechados para as operações de adição e multiplicação por escalares. Isso implica que sejam \mathbf{u} , \mathbf{v} e \mathbf{w} vetores pertencentes ao espaço vetorial \mathbb{V} , os seguintes axiomas devem ser verdadeiros [42]:

1. se $\mathbf{u} \in \mathbb{V}$ e $\mathbf{w} \in \mathbb{V}$, então $(\mathbf{u} + \mathbf{w}) \in \mathbb{V}$
2. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
3. $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
4. Existe um vetor nulo $\mathbf{0}$ tal que $\mathbf{u} + \mathbf{0} = \mathbf{u}$
5. Para cada vetor $\mathbf{u} \in \mathbb{V}$ existe um vetor $-\mathbf{u} \in \mathbb{V}$ tal que $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$
6. Se $\mathbf{u} \in \mathbb{V}$ e z é um escalar, então $(z\mathbf{u}) \in \mathbb{V}$
7. $z(\mathbf{u} + \mathbf{v}) = z\mathbf{u} + z\mathbf{v}$
8. $(z + d)\mathbf{u} = z\mathbf{u} + d\mathbf{u}$
9. $z(d\mathbf{u}) = (zd)\mathbf{u}$
10. $1\mathbf{u} = \mathbf{u}$

A.1.1 Subespaços vetoriais

Um subespaço vetorial \mathbb{S} pode ser definido como um subconjunto dos vetores pertencentes a \mathbb{V} que seguem os axiomas definidos acima, ou seja, são considerados fechados para as operações de adição e multiplicação por escalares. Por exemplo, o conjunto $\mathbb{Z} = \{x, 2x\}$ é um subespaço vetorial de \mathbb{R}^2 uma vez que $\mathbb{Z} \subset \mathbb{R}^2$ e os axiomas definidos anteriormente são verdadeiros para todo vetor $\mathbf{z} \in \mathbb{Z}$.

A.2 Projeções

Os métodos que envolvem subespaços vetoriais são muitas vezes baseados em conceitos geométricos. Dessa forma, operações básicas com vetores como projeções ortogonais e projeções oblíquas são importantes para a obtenção de determinados parâmetros nesses sistemas [50].

A.2.1 Projeções oblíquas

As projeções oblíquas são utilizadas para realizar a decomposição de um determinado espaço vetorial euclidiano \mathbb{A} em dois subespaços vetoriais não-ortogonais \mathbb{B} e \mathbb{D} [42] [39], conforme ilustrado na Figura A.2.1. A equação (A.1) define esta relação.

$$\mathbb{A} = \mathbb{B} + \mathbb{D}, \quad \mathbb{B} \cap \mathbb{D} = \{0\} \tag{A.1}$$

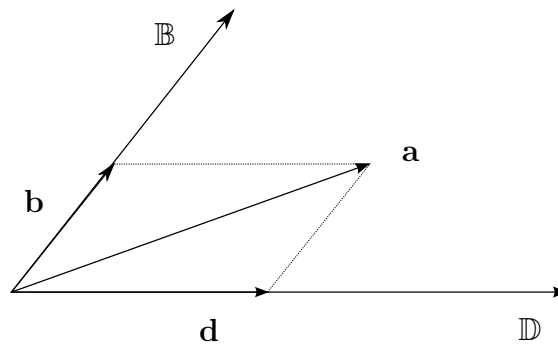


Figura A.1: Exemplo de projeção oblíqua

Dessa forma, um determinado vetor $\mathbf{a} \in \mathbb{A}$ possui uma decomposição única expressa pela relação (A.2). A projeção do vetor \mathbf{a} em \mathbb{D} é representada por $\mathbf{d} = P_{\parallel \mathbb{D}}^{\mathbb{D}}(\mathbf{a})$ e a projeção desse mesmo vetor em \mathbb{B} é representada por $\mathbf{b} = P_{\parallel \mathbb{B}}^{\mathbb{B}}(\mathbf{a})$.

$$\mathbf{a} = \mathbf{b} + \mathbf{d}, \quad \mathbf{b} \in \mathbb{B}, \quad \mathbf{d} \in \mathbb{D} \tag{A.2}$$

A.2.2 Projeções ortogonais

As projeções ortogonais se diferenciam das projeções oblíquas pelo fato dos dois subespaços vetoriais nos quais é decomposto um determinado espaço euclidiano \mathbb{A} serem ortogonais. Um exemplo dessa decomposição está representada na Figura A.2.2.

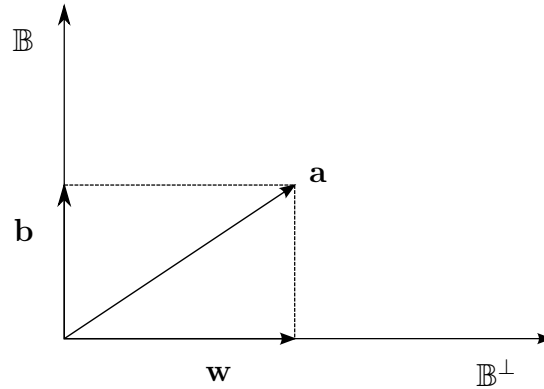


Figura A.2: Exemplo de projeção ortogonal

Este tipo de decomposição é bastante utilizada no método de subespaços vetoriais uma vez que os subespaços referentes ao sinal e ao ruído são ortogonais. Assim, a separação do sinal nesses subespaços é realizada através de projeções ortogonais. A expressão (A.3) representa a decomposição de um vetor $\mathbf{a} \in \mathbb{A}$ em dois subespaços \mathbb{V} e \mathbb{V}^\perp , em que \mathbb{V}^\perp representa um subespaço ortogonal a \mathbb{V} .

$$\mathbf{a} = \mathbf{v} + \mathbf{w}, \quad \mathbf{v} \in \mathbb{V}, \quad \mathbf{w} \in \mathbb{V}^\perp \quad (\text{A.3})$$

A.3 Consistência de um sistema linear

Todo sistema de equações pode ser representado em forma matricial. Por exemplo, o sistema (A.4) pode ser representado em função de uma matriz de coeficientes \mathbf{A} , uma matriz de variáveis \mathbf{x} e uma matriz de termos independentes \mathbf{b} , conforme pode ser observado na equação (A.5).

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \quad (\text{A.4})$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (\text{A.5})$$

Com esta representação, o sistema (A.4) tem como solução a expressão (A.6). Verifique-se que a solução de um sistema linear depende da inversa da matriz dos coeficientes. Portanto, o sistema pode ter ou não solução.

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (\text{A.6})$$

Um sistema de equações é denominado consistente quando existe alguma solução para suas variáveis e inconsistente caso contrário. Em geral, existem três possibilidades a respeito das soluções para um sistema linear:

- Uma única solução \rightarrow o sistema é dito possível e determinado;
- Nenhuma solução \rightarrow o sistema é dito impossível;
- Infinitas soluções \rightarrow o sistema é dito possível e indeterminado.

Para determinar a consistência de um sistema linear, a forma matricial é bastante utilizada. Assim, sendo $\mathbf{A}^{M \times N}$ (matriz dos coeficientes) e $\mathbf{A}_b^{M \times (N+1)}$ (matriz dos coeficientes acrescentada da matriz coluna dos termos independentes), em que $M < N$, determina-se a consistência de um sistema linear da seguinte forma:

- $\text{posto}(\mathbf{A}_b) = \text{posto}(\mathbf{A}) = N \rightarrow$ existe uma única solução;
- $\text{posto}(\mathbf{A}_b) = \text{posto}(\mathbf{A}) = M < N \rightarrow$ existem infinitas soluções;
- $\text{posto}(\mathbf{A}_b) > \text{posto}(\mathbf{A}) \rightarrow$ não existem soluções.

Por exemplo, a consistência do sistema linear representado em (A.7) pode ser determinada da seguinte forma:

$$\begin{pmatrix} 2 & -1 & 7 \\ 4 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix} \quad (\text{A.7})$$

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 7 \\ 4 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix}$$

$$\mathbf{A}_{\mathbf{b}} = \begin{pmatrix} 2 & -1 & 7 & 2 \\ 4 & 2 & 2 & 5 \\ 3 & 1 & 3 & 1 \end{pmatrix}$$

$$\text{posto}(\mathbf{A}) = 2$$

$$\text{posto}(\mathbf{A}_{\mathbf{b}}) = 3 \tag{A.8}$$

$$\text{posto}(\mathbf{A}_{\mathbf{b}}) > \text{posto}(\mathbf{A})$$

Observando o resultado obtido na equação (A.8), pode-se determinar a inexistência de soluções para o sistema linear (A.7). Portanto, este sistema é inconsistente.

A.4 Decomposição em valores singulares - SVD

A decomposição em valores singulares (SVD - Singular Value Decomposition) é uma das mais poderosas ferramentas matemáticas para cálculos matriciais. Essa decomposição é apresentada no Teorema 1 e é utilizada para diversos propósitos, dentre os quais podemos citar:

- Determinação do posto e do posto numérico na presença de erros de arredondamento;
- Resolução de problemas de mínimos quadrados, mesmo quando a matriz de coeficientes não possui posto completo;
- Cálculo da matriz pseudoinversa bastante utilizada quando precisa-se da inversa de uma matriz não-quadrada.

Outras informações e aplicações sobre a decomposição matricial SVD podem ser encontradas no livro *Fundamentals of Matrix Computations* [70].

A.4.1 Cálculo do SVD utilizando EVD

A decomposição em valores singulares (SVD) está relacionada com a decomposição em autovalores (EVD). Primeiramente, pode-se definir a decomposição em autovalores através do Teorema 5. A prova desse Teorema pode ser encontrada no livro *Fundamentals of Matrix Computations* [70].

Teorema 5 *Seja \mathbf{A} uma matriz com autovalores dados por $\lambda_1, \lambda_2, \dots, \lambda_k$ e autovetores linearmente independentes $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$, a relação de similaridade descrita na equação (A.9) é verdadeira,*

$$\begin{aligned}\mathbf{A} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \\ \mathbf{U} &= (\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_k)\end{aligned}\tag{A.9}$$

em que \mathbf{U} é uma matriz ortogonal que possui os autovetores de \mathbf{A} e $\mathbf{\Lambda}$ é uma matriz diagonal que contém os autovalores de \mathbf{A} .

Uma característica bastante interessante da decomposição EVD é que o cálculo desta decomposição para potências da matriz \mathbf{A} , como no caso de \mathbf{A}^2 , pode ser facilmente encontrado como é ilustrado na equação (A.10).

$$\begin{aligned}\mathbf{A} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \\ \mathbf{A}^2 &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \\ \mathbf{A}^2 &= \mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^{-1}\end{aligned}\tag{A.10}$$

Assim, ao calcular o SVD de uma determinada matriz $\mathbf{A}\mathbf{A}^T$ obtém-se o resultado expresso na equação (A.11).

$$\begin{aligned}\text{SVD}(\mathbf{A}\mathbf{A}^T) &= \text{SVD}(\mathbf{A})\text{SVD}(\mathbf{A}^T) \\ \text{SVD}(\mathbf{A}\mathbf{A}^T) &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)(\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T)\end{aligned}\tag{A.11}$$

Como as matrizes \mathbf{U} e \mathbf{V} são ortonormais, a relação expressa na equação (A.12) é válida.

$$\begin{aligned}\mathbf{U}\mathbf{U}^T &= \mathbf{I} \\ \mathbf{V}\mathbf{V}^T &= \mathbf{I} \\ \text{SVD}(\mathbf{A}\mathbf{A}^T) &= (\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T) \\ \text{SVD}(\mathbf{A}\mathbf{A}^T) &= (\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^{-1}) \\ \text{SVD}(\mathbf{A}\mathbf{A}^T) &= \text{EVD}(\mathbf{A}\mathbf{A}^T)\end{aligned}\tag{A.12}$$

Com base na relação (A.12) é possível encontrar os valores singulares de uma matriz \mathbf{A} através da relação expressa em (A.13).

$$\sigma_{\mathbf{A}} = \sqrt{\lambda_{\mathbf{A}\mathbf{A}^T}}\tag{A.13}$$

SVD e matrizes ortonormais

Uma matriz \mathbf{A} é dita ortonormal quando $\mathbf{A}\mathbf{A}^T = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ e as normas de cada coluna de \mathbf{A} é unitária. Para este tipo de matriz, a decomposição SVD possui uma interessante característica. Todos os valores singulares de uma matriz ortonormal são iguais e unitários, conforme pode ser visto na expressão (A.14).

$$\begin{aligned} \text{EVD}(\mathbf{A}\mathbf{A}^T) &= \mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^{-1} \\ \mathbf{A}\mathbf{A}^T &= \mathbf{I} \\ \text{EVD}(\mathbf{I}) &= \mathbf{U}\mathbf{I}\mathbf{U}^{-1} \\ \sigma_{\mathbf{A}} &= \sqrt{\lambda_{\mathbf{A}\mathbf{A}^T}} = 1 \end{aligned} \tag{A.14}$$

A.4.2 Aplicações para os valores singulares

Norma euclidiana de uma matriz

A norma euclidiana de uma matriz, muitas vezes chamada de norma espectral, é definida pela equação (A.15). Esta norma representa o maior aumento possível que um determinado vetor $\mathbf{x} \in \mathbb{R}^m$ pode sofrer quando multiplicado por uma matriz \mathbf{A} .

$$\|\mathbf{A}\|_2 = \max \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}, \quad p/\mathbf{x} \neq \mathbf{0} \tag{A.15}$$

A norma euclidiana de uma certa matriz $\mathbf{A} \in \mathbb{R}^{n \times m}$ é dada pelo maior valor singular da mesma. A prova pode ser encontrada através de uma análise realizada na expressão (A.16) que representa a decomposição em valores singulares de uma determinada matriz \mathbf{A} .

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \\ \mathbf{A}\mathbf{V} &= \mathbf{U}\mathbf{\Sigma} \end{aligned} \tag{A.16}$$

Assim, tem-se que $\mathbf{A}\mathbf{v}_1 = \sigma_1 \mathbf{u}_1$, $\mathbf{A}\mathbf{v}_2 = \sigma_2 \mathbf{u}_2$, \dots , $\mathbf{A}\mathbf{v}_m = \sigma_m \mathbf{u}_m$. Com isso, dentre os vetores da matriz \mathbf{V} o que fornece o maior valor para a expressão (A.15) é \mathbf{v}_1 , conforme descrito na equação (A.17).

$$\begin{aligned} \frac{\|\mathbf{A}\mathbf{v}_1\|_2}{\|\mathbf{v}_1\|_2} &= \frac{\|\mathbf{u}_1\sigma_1\|_2}{\|\mathbf{v}_1\|_2} \\ \frac{\|\mathbf{A}\mathbf{v}_1\|_2}{\|\mathbf{v}_1\|_2} &= \sigma_1 \frac{\|\mathbf{u}_1\|_2}{\|\mathbf{v}_1\|_2} \\ \frac{\|\mathbf{A}\mathbf{v}_1\|_2}{\|\mathbf{v}_1\|_2} &= \sigma_1 \end{aligned} \tag{A.17}$$

Assim,

$$\|\mathbf{A}\|_2 \geq \sigma_1 \quad (\text{A.18})$$

Seja $\mathbf{x} \in \mathbb{R}^m$, este pode ser representado através de uma combinação linear dos vetores singulares à direita conforme (A.19). Como os vetores singulares de \mathbf{V} ($\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$) são ortonormais, temos que a norma euclidiana deste vetor é dada por (A.20).

$$\mathbf{x} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \dots + c_m \mathbf{v}_m \quad (\text{A.19})$$

$$\|\mathbf{x}\|_2^2 = |c_1|^2 + \dots + |c_m|^2 \quad (\text{A.20})$$

Temos ainda que \mathbf{Ax} é dado pela equação (A.21), onde o posto de \mathbf{A} é igual a $r \leq \min(m, n)$. Como os vetores singulares à esquerda $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r$ são ortonormais, a norma de \mathbf{Ax} é dada por (A.22).

$$\begin{aligned} \mathbf{Ax} &= c_1 \mathbf{A}\mathbf{v}_1 + c_2 \mathbf{A}\mathbf{v}_2 + \dots + c_r \mathbf{A}\mathbf{v}_r \\ \mathbf{Ax} &= \sigma_1 c_1 \mathbf{u}_1 + \sigma_2 c_2 \mathbf{u}_2 + \dots + \sigma_r c_r \mathbf{u}_r \end{aligned} \quad (\text{A.21})$$

$$\begin{aligned} \|\mathbf{Ax}\|_2^2 &= |\sigma_1 c_1|^2 + |\sigma_2 c_2|^2 + \dots + |\sigma_r c_r|^2 \\ \|\mathbf{Ax}\|_2^2 &\leq \sigma_1^2 (|c_1|^2 + |c_2|^2 + \dots + |c_r|^2) \\ \|\mathbf{Ax}\|_2^2 &\leq \sigma_1^2 \|\mathbf{x}\|_2^2 \end{aligned} \quad (\text{A.22})$$

A norma euclidiana de \mathbf{A} é dado pela expressão (A.23). Assim, a única solução comum para as expressões (A.18) e (A.23) é representada na equação (A.24).

$$\begin{aligned} \|\mathbf{A}\|_2^2 &= \frac{\|\mathbf{Ax}\|_2^2}{\|\mathbf{x}\|_2^2} \\ \|\mathbf{A}\|_2^2 &\leq \frac{\sigma_1^2 \|\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \\ \|\mathbf{A}\|_2^2 &\leq \sigma_1^2 \end{aligned} \quad (\text{A.23})$$

$$\|\mathbf{A}\|_2 = \sigma_1 \quad (\text{A.24})$$

Número de condição de uma matriz

Como foi dito anteriormente, o posto de uma determinada matriz \mathbf{A} é dado pela quantidade de valores singulares não nulos. No caso de uma matriz possuir posto completo,

a inversa desta matriz \mathbf{A}^{-1} pode ser diretamente calculada. Já no caso desta matriz não possuir posto completo, ou seja, caso possua algum valor singular igual a zero, esta matriz é singular.

Quando algum valor singular não é zero mais é bastante próximo, cria-se um problema. A matriz não é singular mas está bastante próxima da singularidade e, com isso, sua inversa tende a infinito. Isto ocasiona erros de precisão na representação desta inversa. Por conta disto, precisa-se de um parâmetro que possa indicar o quão próximo da singularidade uma matriz se encontra. Este parâmetro é denominado número de condição.

O número de condição de uma matriz é um parâmetro bastante importante para a resolução de um sistema linear. Este número é diretamente relacionado à precisão do resultado obtido para um determinado sistema e é definido pela expressão (A.25).

$$k_2(\mathbf{A}) = \frac{\max_{mag}(\mathbf{A})}{\min_{mag}(\mathbf{A})} \quad (\text{A.25})$$

em que,

$$\begin{aligned} \max_{mag} &= \max \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2}, \quad p/\mathbf{x} \neq \mathbf{0} \\ \min_{mag} &= \min \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2}, \quad p/\mathbf{x} \neq \mathbf{0} \end{aligned}$$

O resultado para a expressão $\max_{mag}(\mathbf{A})$ foi fornecido na demonstração da norma euclidiana de uma matriz e é igual ao maior valor singular de \mathbf{A} . Similarmente constata-se que o valor para o $\min_{mag}(\mathbf{A})$ é dado pelo menor valor singular de \mathbf{A} , provendo o resultado exibido em (A.26) para o cálculo do número de condição.

$$k_2(\mathbf{A}) = \frac{\sigma_1}{\sigma_n} \quad (\text{A.26})$$

Assim, quando o número de condição de uma matriz \mathbf{A} é muito grande, diz-se que esta matriz é mal condicionada e que, portanto, a sua inversa conterà valores muito altos que poderão ocasionar erros de precisão.

Posto numérico de uma matriz

Muitas vezes uma matriz \mathbf{A} que não possui posto completo pode apresentar posto completo devido a alguma perturbação. Por exemplo, a matriz A ilustrada em (A.27) claramente não possui posto completo pois a terceira coluna é a soma das duas primeiras. Quando reparamos os elementos dessa matriz com uma precisão simples, poderá ocorrer erros de arredondamento, perturbando a matriz. Isto faz com que a terceira coluna deixe de ser entendida como a soma das duas primeiras, ocasionando no aumento do posto desta matriz.

$$\mathbf{A} = \begin{pmatrix} 1/3 & 1/3 & 2/3 \\ 2/3 & 2/3 & 4/3 \\ 1/3 & 2/3 & 3/3 \\ 2/5 & 2/5 & 4/5 \\ 3/5 & 1/5 & 4/5 \end{pmatrix} \quad (\text{A.27})$$

De fato, calculando o SVD desta matriz com uma precisão simples nos fornece como valores singulares $\sigma_1 = 2.5987$, $\sigma_2 = 0.3682$ e $\sigma_3 = 8.6614 \times 10^{-17}$. Como pode-se observar, o último valor singular desta matriz é muito menor que os outros valores e poderia ser considerado igual a zero.

Por conta de casos como este o conceito de posto numérico foi criado. Trata-se apenas de um determinado limite imposto pela dimensão dos valores singulares de forma que qualquer valor abaixo deste limite é considerado igual a zero. Desta forma, pode-se dizer que uma matriz tem posto numérico igual a k quando k valores singulares desta matriz são maiores que o limite imposto.

SVD e o problema dos mínimos quadrados

Seja um sistema linear representado pela equação (A.28), em que $\mathbf{A} \in \mathbb{R}^{n \times m}$ possui posto igual a r , $\mathbf{b} \in \mathbb{R}^n$ e $\mathbf{x} \in \mathbb{R}^m$. Quando $n > m$ o sistema não possui uma solução exata, sendo superdeterminado. Portanto, a solução buscada não é mais a exata, mas sim uma aproximação de forma que $\|\mathbf{b} - \mathbf{Ax}\|_2$ seja minimizado. Esta é a formulação do problema dos mínimos quadrados.

$$\mathbf{Ax} = \mathbf{b} \quad (\text{A.28})$$

Como foi mostrado anteriormente, no caso de $n \geq m$ e $\text{posto}(\mathbf{A}) = m$, o sistema possui apenas uma solução. No caso de $n \geq m$ e $\text{posto}(\mathbf{A}) < m$, o sistema possui infinitas soluções e deve ser escolhida a solução que minimize $\|\mathbf{b} - \mathbf{Ax}\|_2$. Apesar desta consideração, o sistema pode possuir mais de uma solução que minimize $\|\mathbf{b} - \mathbf{Ax}\|_2$, devendo ser escolhida para este caso a solução que minimizar $\|\mathbf{x}\|_2$. Assim sendo, o problema dos mínimos quadrados passa a sempre possuir uma solução única.

Para a formulação do problema dos mínimos quadrados, será considerado que as matrizes \mathbf{A} e \mathbf{b} são conhecidas, que $\mathbf{A} \in \mathbb{R}^{n \times m}$, em que $n < m$, e que $\text{posto}(\mathbf{A}) = r < m$. Portanto, a decomposição em valores singulares da matriz \mathbf{A} pode ser calculada pela expressão (A.29), em que $\mathbf{U} \in \mathbb{R}^{n \times n}$ e $\mathbf{V} \in \mathbb{R}^{m \times m}$ são matrizes ortogonais e a matriz $\mathbf{\Sigma}$ é dada pela expressão (A.30).

$$\text{SVD}(\mathbf{A}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{A.29})$$

$$\Sigma = \begin{pmatrix} \hat{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (\text{A.30})$$

em que,

$$\hat{\Sigma} = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sigma_r \end{pmatrix}, \quad \text{com } \sigma_1 \geq \cdots \geq \sigma_r > 0$$

Como a matriz \mathbf{U} é ortogonal, a expressão (A.31) é válida. Substituindo $\mathbf{c} = \mathbf{U}^T \mathbf{b}$ e $\mathbf{y} = \mathbf{V}^T \mathbf{x}$, obtém-se a equação (A.32).

$$\begin{aligned} \|\mathbf{b} - \mathbf{Ax}\|_2 &= \|\mathbf{U}^T(\mathbf{b} - \mathbf{Ax})\|_2 \\ \|\mathbf{b} - \mathbf{Ax}\|_2 &= \|\mathbf{U}^T \mathbf{b} - \Sigma(\mathbf{V}^T \mathbf{x})\|_2 \end{aligned} \quad (\text{A.31})$$

$$\begin{aligned} \|\mathbf{b} - \mathbf{Ax}\|_2^2 &= \|\mathbf{c} - \Sigma \mathbf{y}\|_2^2 \\ \|\mathbf{b} - \mathbf{Ax}\|_2^2 &= \sum_{i=1}^n |c_i - \sigma_i y_i|^2 \end{aligned} \quad (\text{A.32})$$

Como o posto da matriz \mathbf{A} é igual a $r < n$, os valores singulares σ_i para $i > r$ são todos iguais a zero. Assim, pode-se simplificar a expressão (A.32) resultando na equação final (A.33).

$$\|\mathbf{b} - \mathbf{Ax}\|_2^2 = \sum_{i=1}^r |c_i - \sigma_i y_i|^2 + \sum_{i=r+1}^n |c_i|^2 \quad (\text{A.33})$$

Com base na expressão (A.33), é simples verificar que para minimizá-la deve-se fazer $y_i = \frac{c_i}{\sigma_i}$ para $i = 1, \dots, r$. A equação (A.33) indica ainda que os valores para y_{r+1}, \dots, y_m podem ser escolhidos de forma arbitrária uma vez que estes valores não influenciam no resultado desta expressão.

Diante disto, uma solução interessante a ser tomada é a escolha da solução de tal forma que $\|\mathbf{y}\|_2$ também seja minimizado, resultado obtido quando se faz $y_{r+1} = \dots = y_m = 0$. Desta forma, o problema dos mínimos quadrados é resolvido em função da norma mínima.

A inversa generalizada Moore-Penrose

A inversa generalizada Moore-Penrose é uma interessante generalização da inversa de uma matriz. Esta inversa também é conhecida como pseudo-inversa e apesar de apenas matrizes quadradas admitirem inversão, matrizes não-quadradas admitem pseudo-inversão. Assim, para um determinado sistema linear dado por $\mathbf{Ax} = \mathbf{b}$, onde $\mathbf{A} \in \mathbb{R}^{n \times m}$,

a solução para o problema dos mínimos quadrados em função da norma mínima pode ser encontrado através da pseudo-inversa \mathbf{A}^\dagger , conforme demonstrado em (A.34).

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} \quad (\text{A.34})$$

No caso da matriz \mathbf{A} possuir posto completo, pode-se calcular a inversa da matriz através da equação (A.35). Já a pseudo-inversa da matriz \mathbf{A}^T é fornecida pela expressão (A.36).

$$\begin{aligned} \mathbf{A}^\dagger &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \\ &= \mathbf{A}^{-1} (\mathbf{A}^T)^{-1} \mathbf{A}^T \\ &= \mathbf{A}^{-1} \end{aligned} \quad (\text{A.35})$$

$$\begin{aligned} (\mathbf{A}^T)^\dagger &= (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{A} \\ &= (\mathbf{A}^T)^{-1} \mathbf{A}^{-1} \mathbf{A} \\ &= (\mathbf{A}^T)^{-1} \end{aligned} \quad (\text{A.36})$$

Supondo que o posto da matriz \mathbf{A} seja $r < \min(n, m)$, para calcular a pseudo-inversa dessa matriz pode-se utilizar a decomposição em valores singulares conforme será descrito a seguir.

Primeiramente, a ação da matriz \mathbf{A} pode ser completamente descrita pelo diagrama (A.37).

$$\mathbf{A} \mathbf{v}_i = \begin{cases} \sigma_i \mathbf{u}_i & , \text{ para } 0 < i \leq r \\ \mathbf{0} & , \text{ caso contrário} \end{cases} \quad (\text{A.37})$$

Assim, deve-se definir uma matriz pseudo-inversa $\mathbf{A}^\dagger \in \mathbb{R}^{m \times n}$ que obedeça ao diagrama (A.38). Como o posto da matriz \mathbf{A} é igual a $r < \min(n, m)$, deve-se escolher um mapeamento para os termos $\mathbf{A}^\dagger \mathbf{u}_{r+1}, \dots, \mathbf{A}^\dagger \mathbf{u}_n$. Uma escolha bastante razoável é a anulação desses termos.

$$\mathbf{A}^\dagger \mathbf{u}_i = \begin{cases} \sigma_i^{-1} \mathbf{v}_i & , \text{ para } 0 < i \leq r \\ \mathbf{0} & , \text{ caso contrário} \end{cases} \quad (\text{A.38})$$

A decomposição em valores singulares da matriz \mathbf{A} é dada por (A.39). Conforme pode-se observar em (A.40), é necessário o cálculo da inversa da matriz dos valores singulares $\mathbf{\Sigma}$. O problema enfrentado é que esta matriz é singular e portanto não podemos calcular a inversa diretamente. A solução está exatamente na atribuição realizada em (A.38) sobre os termos com $i > r$. Define-se então uma matriz $\mathbf{\Sigma}^\dagger$ como a inversa da matriz $\mathbf{\Sigma}$ obedecendo a forma exibida em (A.41).

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{A.39})$$

$$\begin{aligned} \mathbf{A}^{-1} &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1} \\ &= (\mathbf{V}^T)^{-1}\mathbf{\Sigma}^{-1}(\mathbf{U})^{-1} \\ &= \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T \end{aligned} \quad (\text{A.40})$$

$$\mathbf{\Sigma}^\dagger = \begin{pmatrix} \hat{\mathbf{\Sigma}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (\text{A.41})$$

em que,

$$\hat{\mathbf{\Sigma}}^{-1} = \begin{pmatrix} \sigma_1^{-1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sigma_r^{-1} \end{pmatrix}$$

Com isso, através da definição (A.42) pode-se calcular a pseudo inversa da matriz \mathbf{A} , cujo posto é dado pelo número de valores singulares não nulos contidos na matriz $\mathbf{\Sigma}^\dagger$.

$$\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T \quad (\text{A.42})$$

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)