Revisitando o Uso da Transformada de Fourier no Reconhecimento de Voz para Robótica Móvel

Igor R. Sousa
Pós-Grad. Eng. de Teleinformática(PPGETI)
Universidade Federal do Ceará (UFC)
Campus do Pici, Centro de Tecnologia
igor.sousa@alu.ufc.br

Jefferson C. Figueiredo
Departamento de Ensino
Instituto Federal do Ceará (IFCE)
Campus Tauá
jefferson.figueiredo@ifce.edu.br

Guilherme A. Barreto
Pós-Grad. Eng. de Teleinformática(PPGETI)
Universidade Federal do Ceará (UFC)
Campus do Pici, Centro de Tecnologia
gbarreto@ufc.br

***Abstract*—Este trabalho trata do reconhecimento de comandos de voz para o acionamento de um robô móvel. Comandos básicos formados pela elocução das palavras *avançar*, *direita*, *esquerda*, *parar* e *recuar* são gravados por um usuário e utilizados para criação de um banco de arquivos de áudio. A partir dos áudios gravados, técnicas de extração de atributos de sinais de voz são usadas para gerar dois bancos de dados que servirão para treinamento e teste do módulo de reconhecimento de voz do *software* de controle do robô. Para isso, são usadas as clássicas técnicas de codificação linear preditiva (*linear predictive coding*, LPC) e transformada rápida de Fourier (*fast Fourier transform*, FFT). Em particular, a FFT é usada de um modo não usual, porém melhor adequada ao uso em sistemas embarcados. Além disso, diversas técnicas de pré-processamento são testadas sobre os atributos extraídos para avaliar possíveis impactos na acurácia dos classificadores avaliados (discriminante linear de mínimos quadrados e a rede perceptron multicamadas). A contribuição principal do estudo reside na obtenção de uma solução adequada para ser embarcada em um sistema robótico real.**

***Index Terms*—Comando de voz, coeficientes LPC, transformada de Fourier, rede MLP, robô móvel.**

I. Introdução

Sistemas de comando por voz constituem uma importante área de pesquisa em processamento de sinais e aprendizado de máquinas devido ao escopo quase ilimitado de suas possíveis aplicações, de eletrodomésticos a sistemas robóticos para fins especiais, tais como calculadoras acionadas por voz [1] ou cadeiras de rodas robotizadas [2]. Além disso, interfaces com o usuário via comandos de voz estão se tornando cada vez mais presentes em atividades cotidianas de todos que fazem uso de tecnologias da informação e da comunicação, haja vista a popularização de assistentes virtuais, como Alexa e Siri.

Fundamentais para um sistema comandado por voz são os módulos de *processamento* e *reconhecimento*, que podem estar disponíveis para processamento embarcado no próprio *hardware* do robô ou remotamente via rede local ou de telefonia celular. Atualmente, há diversas bibliotecas de *software* para a implementação de sistemas para o reconhecimento de comados de voz, tal como a CMU SPHINX[1] usada em [3]. A maioria delas, porém, foi desenvolvida para aplicações em língua inglesa. Em razão da carência deste tipo de biblioteca para o português brasileiro, o presente artigo apresenta os resultados da implementação de um sistema de processamento/extração de atributos de comandos de voz e subsequente classificação da elocução.

O desenvolvimento do sistema envolveu desde a criação do banco de comandos de voz, ou seja, da gravação das elocuções relativas a cinco comandos de movimentação de um robô móvel, até a implementação de todas as rotinas matemáticas de processamento de sinais e aprendizado de máquinas para fins de extração de atributos dos sinais de voz e classificação dos comandos. O sistema não depende de bibliotecas externas, pois todas as rotinas foram implementadas *from scratch*.

As elocuções gravadas foram inicialmente usadas para treinar e testar *offline* os módulos de extração de atributos e reconhecimento dos sinais de voz. Para extração de atributos foram usadas duas técnicas clássicas de processamento de sinais, a saber, a codificação linear preditiva (*linear predictive coding*, LPC) e a transformada rápida de Fourier (*fast Fourier transform*, FFT). O discriminante linear de mínimos quadrados (LMQ) e a rede perceptron multicamadas (MLP) foram usadas para fins de classificação dos comandos de voz.

Contudo, diferentemente do modo usual com que a FFT é aplicada em reconhecimento de voz, ou seja, em segmentos curtos do sinal de voz, neste estudo mostramos que a aplicação da FFT sobre o sinal inteiro é viável e conduz a elevadas acurácias na classificação, mesmo sendo o sinal de voz um sinal não estacionário. Esta abordagem é contrastada com a técnica de extração LPC, que para ser aplicada necessariamente exige o particionamento do sinal de voz completo em segmentos menores. Este achado da pesquisa permitiu reduzir sobremaneira o custo de implementação do sistema de extração/reconhecimento de voz, sendo esta a principal contribuição do trabalho. É importante ressaltar, contudo, que muito embora o presente trabalho trate especificamente dos módulos de extração e classificação de comandos por voz, estes foram desenvolvidos com o intuito de serem embarcados no robô móvel mostrado na Figura 1.

O robô utilizado consiste num kit de robótica RC-FLS, fabricado pela Robocore®, constituído de um chassi de metal acoplado a dois servomotores de rotação contínua, cada um conectado a uma roda para realizar o movimento. Esse sistema se liga diretamente a uma placa de circuito para interface de energia, de modo a isolar o controlador embarcado da

[1] `https://cmusphinx.github.io/`

Figura 1: Robô móvel usado nesta pesquisa.

corrente necessária para mover o motor. A alimentação dos motores e do microcontrolador pode ser feita através de baterias (armazenadas num espaço abaixo do chassi) ou um cabo de energia ligado a uma fonte ou computador de longo comprimento para permitir o translado do robô móvel.

A organização deste trabalho segue a seguinte ordem. A Seção II contém informações a respeito do método LPC e como sua modelagem pode ser realizada. A Seção III explica de maneira breve o funcionamento dos classificadores LMQ e MLP. A Seção IV traz explicações de métodos de transformação da matriz de dados, bastante utilizada em processos de classificação. Por fim, as Seções V, VI e VII trazem respectivamente a metodologia empregada neste trabalho, os resultados obtidos e as conclusões do trabalho.

II. Extração de Atributos via Coeficientes LPC

A técnica LPC é uma das abordagens mais populares para parametrizar sinais de voz, produzindo bons resultados em ambientes livres de ruído, mas com desempenho comprometido em ambientes ruidosos [4]. Esta técnica consiste essencialmente em parametrizar um segmento curto (de $10-30\,ms$ de duração) da fala pelos coeficientes de um modelo autorregressivo (AR) que aproxima as características do trato vocal [5]. O processo AR de ordem p é descrito como

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + \cdots + a_p x[n-p] + \nu[n]\,, \quad (1)$$

em que $\{a_1, \ldots, a_p\}$ são os coeficientes do processo e $\nu[n]$ simboliza o ruído AWGN, de média nula e variância σ_ν^2. Dentre os métodos de estimação dos parâmetros $\mathbf{a}$ para processos AR, pode-se destacar o método de Yule-Walker.

A. Estimação via método de Yule-Waker

A equação de Yule-Walker (YW) discreta é definida como

$$\rho[\tau] = a_1\rho[\tau-1] + a_2\rho[\tau-2] + \cdots + a_p\rho[\tau-p], \quad (2)$$

em que $\rho[\tau]$ denota a função de autocorrelação (FAC) no *lag* τ ($\tau \geq 0$), com $\rho[-\tau] = \rho[\tau]$ e $\rho[0] = 1$, enquanto que $\{a_j\}_{j=1}^{p}$ são os mesmos coeficientes do modelo AR da Eq. (1). Estimativas de $\rho[\tau]$, denotadas doravante por $r[\tau]$, podem ser computadas pela seguinte expressão:

$$r[\tau] = \frac{\sum_{k=1}^{N-\tau} x[k]x[k+\tau]}{\mathbf{x}^T\mathbf{x}}, \quad (3)$$

em que $\mathbf{x} \in \mathbb{R}^{N\times 1}$ é o segmento de voz de tempo discreto representado na forma de um vetor coluna de N amostras.

Assim, a partir da Eq. (2), monta-se o seguinte sistema de equações lineares:

$$\mathbf{R}\mathbf{a} = \mathbf{r}\,, \quad (4)$$

$$\begin{bmatrix} 1 & r[1] & \ldots & r[p-1] \\ r[1] & 1 & \ldots & r[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[p-1] & r[p-2] & \ldots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[p] \end{bmatrix}, \quad (5)$$

em que $\mathbf{R} \in \mathbb{R}^{p\times p}$ é uma matriz quadrada simétrica do tipo Toeplitz. A solução deste sistema provê as estimativas dos parâmetros do modelo AR(p); ou seja, $\hat{\mathbf{a}} = \mathbf{R}^{-1}\mathbf{r}$.

Para estimar a ordem dos modelos AR(p) destacam-se o critério de informação de Akaike (*Akaike's information criterion* - AIC)[6], o critério de informação bayesiano (*Bayesian information criterion* - BIC)[7], o critério do erro final de predição (*final prediction error* - FPE)[8] e o critério do comprimento mínimo de descrição (*minimum description length* - MDL)[9]. As versões normalizadas destes critérios são definidas como $AIC(p) = \ln\sigma_e^2(p) + 2p/N_e$, $BIC(p) = \ln\sigma_e^2(p) + p\ln(N_e)/N_e$, $FPE(p) = \ln\sigma_e^2(p) + \ln((N_e + p)/(N_e - p))$, e $MDL(p) = \ln\sigma_e^2(p) + p\ln(N_e)/2N_e$, em que σ_e^2 é a variância dos resíduos $e[n] = x[n] - \hat{x}[n]$. Estes critérios avaliam a qualidade do preditor de acordo com σ_e^2, penalizando o excesso de parâmetros através dos termos que dependem de p [10].

III. Classificadores Avaliados

Nesta seção, são descritos brevemente os dois classificadores implementados neste estudo.

A. Classificador LMQ

Este classificador formula o problema de classificação como uma transformação linear $\mathbf{d}_n = \mathbf{W}\mathbf{x}_n$, em que o rótulo $\mathbf{d}_n$ é um vetor binário, de dimensão M e de norma unitária que identifica unicamente a classe de $\mathbf{x}_n$ entre as M existentes. Já $\mathbf{W}$ é a matriz de pesos responsável pela transformação linear do vetor de atributos. O rótulo ${}^i\mathbf{d}$ da classe i é definido pela convenção *one-hot-encoding* como

$${}^1\mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad {}^2\mathbf{d} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad {}^M\mathbf{d} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad (6)$$

de forma que as classes são mutualmente exclusivas. Agregando todos os N vetores ${}^i\mathbf{d}$ em uma matriz $\mathbf{D}$ de dimensão $M \times N$ e todas as N amostras em uma matriz $\mathbf{X}$ de dimensão $P \times N$, em que P é o número de atributos, forma-se a seguinte versão matricial de $\mathbf{d}_n = \mathbf{W}\mathbf{x}_n$:

$$\mathbf{D} = \mathbf{W}\mathbf{X}\,. \quad (7)$$

O critério dos mínimos quadrados consiste em minimizar o quadrado do erro de estimação $J_{LMQ}(\hat{\mathbf{W}})$ cometido pela estimativa $\hat{\mathbf{W}}$ da matriz $\mathbf{W}$:

$$J_{LMQ}(\hat{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{\left(\mathbf{D}-\hat{\mathbf{W}}\mathbf{X}\right)^T\left(\mathbf{D}-\hat{\mathbf{W}}\mathbf{X}\right)\right\}, \quad (8)$$

em que Tr denota o operador *traço* de uma matriz. A estimativa de $\mathbf{W}$ que minimiza $J_{LMQ}(\hat{\mathbf{W}})$ de forma ótima é

$$\hat{\mathbf{W}} = \mathbf{D}\mathbf{X}^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}. \quad (9)$$

Finalmente, a classe predita para a amostra $\mathbf{x}_n$ é obtida pelo índice da maior componente do vetor $\hat{\mathbf{d}}_n = \hat{\mathbf{W}}\mathbf{x}_n$.

B. Classificador MLP

A seguir, é descrito o funcionamento básico de uma rede MLP totalmente conectada e com uma camada oculta, treinada via algoritmo de retropropagação do erro. Na iteração t, a ativação do i-ésimo neurônio oculto, i=1, ..., Q, é dada por

$$u_i^{(h)}(t) = \sum_{j=1}^{P} w_{ij}(t)x_j(t) - \theta_i(t) = \sum_{j=0}^{P} w_{ij}(t)x_j(t), \quad (10)$$

em que w_{ij} é o peso sináptico que conecta a entrada j ao neurônio oculto i, $\theta_i(t)$ é o limiar do neurônio oculto i, Q $(2 \leq Q < \infty)$ é o número de neurônios ocultos e P é a dimensão do vetor de entrada (excluindo o limiar). Para simplificar, definimos $x_0(t) = -1$ e $w_{i0} = \theta_i^{(h)}(t)$. A saída do neurônio i é então definida por

$$y_i^{(h)}(t) = \varphi\left[u_i^{(h)}(t)\right] = \varphi\left[\sum_{j=0}^{P} w_{ij}(t)x_j(t)\right],$$

em que $\varphi(\cdot)$ é a função tangente hiperbólica. Os valores de saída dos neurônios de saída são dados por

$$y_k^{(o)}(t) = \varphi\left[u_k^{(o)}(t)\right] = \varphi\left[\sum_{i=0}^{Q} m_{ki}(t)y_i^{(h)}(t)\right],$$

em que m_{ki} é o peso sináptico que conecta o neurônio oculto i ao neurônio de saída k ($k = 1, \ldots, M$), e $M \geq 1$ é o número de neurônios da camada de saída. Definimos $y_0^{(h)}(t) = -1$ e $m_{k0} = \theta_k^{(o)}(t)$, onde $\theta_k^{(o)}(t)$ é o *bias* do neurônio k.

A retropropagação começa na camada de saída propagando os sinais de erro, $e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t)$, onde $d_k(t)$ é o rótulo do neurônio k, em direção à camada oculta. O chamado *gradiente local* do neurônio k é dado por

$$\delta_k^{(o)}(t) = \varphi'\left[u_k^{(o)}(t)\right] e_k^{(o)}(t),$$

em que $\varphi'\left[u_k^{(o)}(t)\right] = \partial\varphi/\partial u_k^{(o)}$. Da mesma forma, o *gradiente local* $\delta_i^{(h)}(t)$ do neurônio oculto i é calculado como

$$\delta_i^{(h)}(t) = \varphi'\left[u_i^{(h)}(t)\right]\sum_{k=1}^{M} m_{ki}(t)\delta_k^{(o)}(t) = \varphi'\left[u_i^{(h)}(t)\right] e_i^{(h)}(t), \quad (11)$$

em que o termo $e_i^{(h)}(t)$ desempenha o papel de um sinal de erro retropropagado ou projetado para o neurônio oculto i, uma vez que tais sinais de erro "ocultos" são combinações dos sinais de erro "verdadeiros" calculados para os neurônios de saída.

Finalmente, os pesos sinápticos dos neurônios de saída são atualizados de acordo com

$$m_{ki}(t+1) = m_{ki}(t) + \eta(t)\delta_k^{(o)}(t)y_i^{(h)}(t), \quad i = 0, \ldots, Q,$$

em que $0 < \eta(t) < 1$ é a taxa de aprendizado. Os pesos dos neurônios ocultos são, por sua vez, ajustados por meio de uma regra de aprendizagem semelhante,

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\delta_i^{(h)}(t)x_j(t), \quad j = 0, \ldots, P.$$

Uma apresentação completa de todo o conjunto de treinamento durante o processo de aprendizagem é chamada de época. Muitas épocas podem ser necessárias até que a convergência do algoritmo de retropropagação aconteça. Uma maneira simples de avaliar a convergência é por meio do erro quadrático médio,

$$\varepsilon_{train} = \frac{1}{2N}\sum_{t=1}^{N}\sum_{k=1}^{M}\left[d_k(t) - y_k^{(o)}(t)\right]^2, \quad (12)$$

calculado no final de uma execução de treinamento usando os vetores de dados de treinamento.

IV. Métodos de transformação nos dados

Nesta seção, são descritas duas transformações de dados bastante utilizadas no âmbito de classificação de padrões: a transformação de Box-Cox e a análise de componentes principais *(principal component analysis*, PCA).

A. Box-Cox

Normalidade é uma importante suposição para muitas métodos estatísticos. Porém, muitos atributos não seguem uma distribuição normal. Nestes casos, é possível realizar uma operação não linear, conhecida como transformação de Box-Cox (BC) [11] sobre a variável x,

$$x^* = \begin{cases} \dfrac{x^\gamma - 1}{\gamma}, & \text{se } 0 < \gamma < 1 \\ \ln x, & \text{se } \gamma = 0 \end{cases}, \quad (13)$$

em que γ é um hiperparâmetro a ser especificado. Por meio da Eq. (13), uma variável não gaussiana pode assumir uma distribuição mais simétrica, mais similar à normal.

B. Análise de componentes principais

PCA é uma transformação linear que atua nos vetores de atributos originais $\mathbf{x}$ para gerar um novo conjunto com atributos descorrelacionados $\mathbf{z}$, ou seja, em que a matriz de covariância dos novos atributos é diagonal [12]. A transformação dos dados é realizada de acordo com

$$\mathbf{z} = \mathbf{V}_\beta\mathbf{x}, \quad (14)$$

em que $\mathbf{V}_\beta$ é uma matriz de ordem $\beta \times P$ formada pelos autovetores $\mathbf{v}$ da matriz de covariância das amostras,

$$\mathbf{V}_\beta = \left[\mathbf{v}_1 \mid \mathbf{v}_2 \mid \ldots \mid \mathbf{v}_\beta\right]^T, \quad 1 \leq \beta \leq P. \quad (15)$$

Os autovetores $\{\mathbf{v}_k\}_{k=1}^{\beta}$ estão ordenados em associação com a ordem decrescente dos autovalores da matriz de covariância do conjunto original. Para $\beta = P$, a nova matriz $\mathbf{z}$ possui mesma dimensão que a matriz original $\mathbf{x}$. Porém, para $\beta < P$, ocorre uma redução de dimensionalidade. Desta forma, a matriz de covariância dos dados transformados possui dimensão β. Por fim, com a aplicação do PCA e a escolha adequada de β, pode-se descorrelacionar os dados e reduzir sua dimensão ao mesmo tempo que se preserva a informação relevante contida nos dados originais [13].

V. Aquisição e Parametrização do Sinal de Voz

A sequência de passos utilizada na metodologia deste trabalho é apresentada na Figura 2. O primeiro passo é o da aquisição de sinais de voz, em que a voz é capturada por um microfone. Em seguida, o sinal passa por uma reamostragem para que se obtenha mais dados a partir dos sinais de voz gravados originalmente. O terceiro passo consiste em realizar extração de características para treinamento e classificação, usando LPC e FFT. Os dois conjuntos de dados gerados, com atributos distintos, foram usados para treinamento e teste dos classificadores LMQ e MLP.

A. Aquisição de Sinais de Voz

Os sinais de voz utilizados neste trabalho são sinais *WAVE-form stereo* amostrados em $F_s = 44.100\,Hz$, através do programa Audacity[2], com dois canais. A aquisição foi realizada usando um microfone de computador comum a cabo (P2, $3{,}5\,mm$), com resposta em frequência de 100 a $12.000\,Hz$.

Os arquivos de áudio são divididos em 5 comandos (ou classes). Cada comando possui 10 repetições, totalizando 50 arquivos. Esses áudios possuem variações de distância do microfone e duração da fala. A voz de cada unidade é a de uma mesma pessoa. O ambiente de gravação é controlado, mas não livre de ruídos. O objetivo disso é tornar o treinamento compatível com ambientes do cotidiano, como em casa ou em laboratório.

[2]https://www.audacityteam.org/

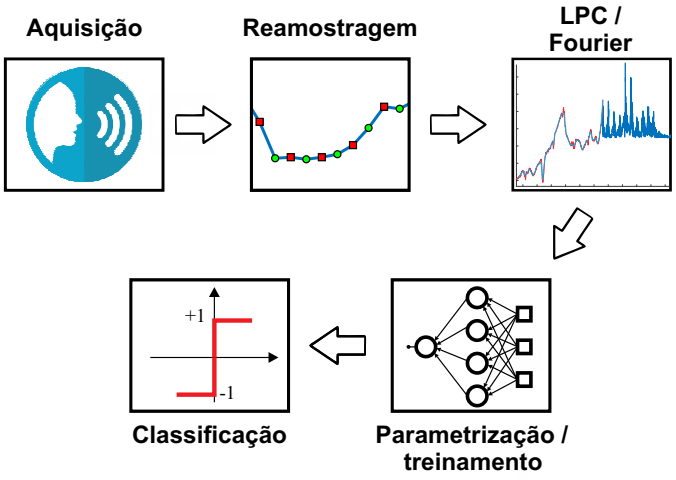


Figura 2: Passos utilizados para a classificação.

As cinco classes são constituídas de cinco palavras, definidas considerando o contexto de controle de um sistema robótico móvel por voz. Os comandos e suas respectivas funções são listados a seguir:

- "avançar": o robô segue em frente em linha reta até que um novo comando seja dado ou até que encontre um obstáculo no caminho;
- "direita": o robô para e gira $90°$ à direita;
- "esquerda": o robô para e gira $90°$ à esquerda;
- "parar": o robô aborta imediatamente seu movimento, caso esteja avançando ou recuando;
- "recuar": o robô segue para trás em linha reta até que um novo comando seja dado ou até que ele encontre um obstáculo no caminho.

Devido à pequena quantidade de áudios por classe, optou-se por reamostrar os sinais para $F_s = 22.050\,Hz$, conforme ilustrado na Figura 3. Cada unidade de áudio é, portanto, dividida em dois áudios, cada um com metade das amostras do sinal original, amostradas de forma intercalada. Assim, dobra-se a quantidade de unidades de áudios, totalizando 20 sinais. Como os arquivos de áudio utilizados neste trabalho são sinais estéreo, tem-se 40 sinais por classe.

B. Parametrização do sinal de voz via LPC

É importante lembrar que a parametrização do sinal via coeficientes LPC é aplicada na análise de curta duração, ou seja, em segmentos do sinal de voz de $10-30\,ms$ de duração. Assim, o sinal de voz deve ser janelado, formando quadros, com ou sem superposição entre eles. Os coeficientes LPC são estimados para cada quadro do sinal de voz. Na prática, utilizam-se modelos de predição que variam, em geral, de AR(10) a AR(20) [1]. Os coeficientes de todos os quadros são então concatenados, formando assim os atributos para serem utilizados em um classificador.

Um problema existente na utilização de LPC para classificação de voz é a quantidade de quadros utilizados, pois cada palavra a ser reconhecida possui tamanho diferente, o que acarreta em tempos de execução de fala diferentes para cada tipo de comando. Este problema ocorre nos dados de voz utilizados neste trabalho, como por exemplo, entre as palavras "avançar" e "parar". Se o tamanho dos quadros for fixado em $10\,ms$, a palavra "avançar" teria cerca de 87 quadros, enquanto a palavra "parar" teria 65 quadros. Na prática, isto resultaria em amostras com diferentes número de atributos, o que impediria utilizar métodos de classificação tradicionais.

Uma solução para este problema é a normalização da elocução, necessário para fixar o número de atributos que será apresentado ao classificador. Para padronizar as elocuções neste trabalho, são selecionados quadros sem sobreposição. Baseado no sinal de áudio de menor duração, todos os sinais são divididos em 31 quadros, como ilustrado na Figura 4. Para

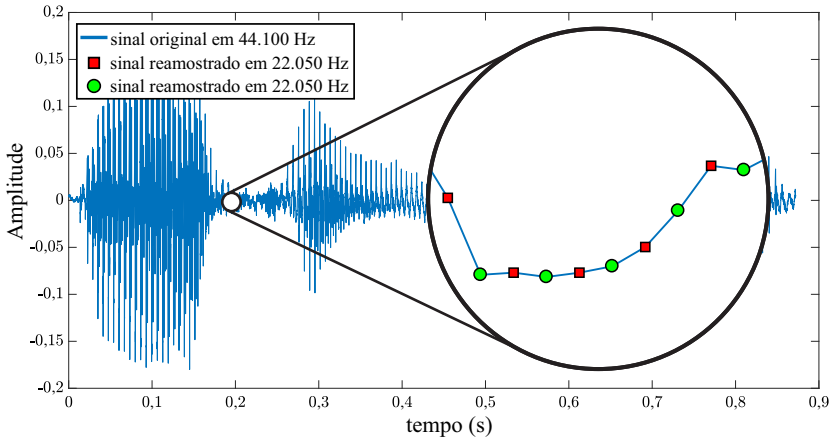


Figura 3: Reamostragem de um sinal de voz.

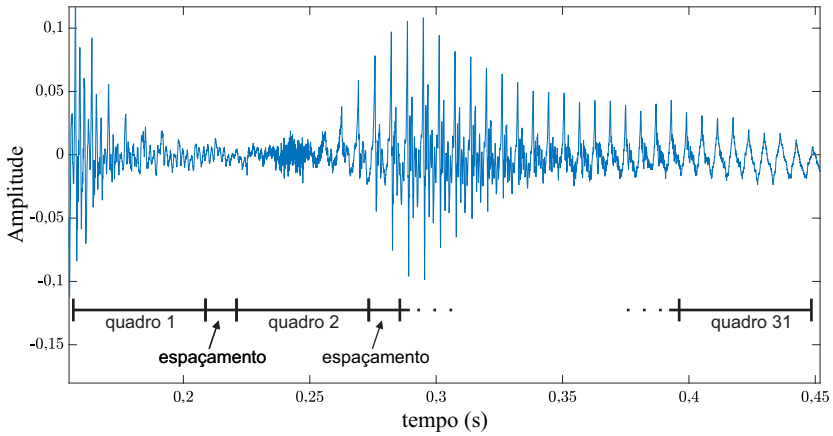


Figura 4: Divisão de quadros dos sinais de áudio.

cada sinal, o tamanho dos quadros T_q e do espaçamento T_{esp} em número de amostras é calculado como

$$T_q = \frac{T_{sinal} T_{q_{min}} F_s}{T_{minimo}}, \quad (16)$$

$$T_{esp} = \frac{T_{sinal} - 31 T_q}{30}, \quad (17)$$

em que T_{sinal}, T_{minimo} e $T_{q_{min}}$ são, respectivamente, o tamanho do sinal, tamanho do menor sinal do banco de dados e duração mínima do quadro em segundos. Neste trabalho, é utilizado $T_{q_{min}} = 15\,ms$. Cada um dos 31 quadros é submetido à modelagem AR(p) pela técnica de Yule-Walker.

Conforme Souza (2009) [1], em geral se utiliza $10 \leq p \leq 20$. Para escolha da ordem p dos filtros AR neste trabalho, são utilizados os critérios de informação descritos na Seção II para $p = 10$, $p = 15$ e $p = 20$, apresentados na Tabela I.

Tabela I: Critérios de informação para um quadro.

ordem	AIC	BIC	FPE	MDL
$\mathbf{p = 10}$	$\mathbf{-13,1665}$	$\mathbf{-13,0928}$	$\mathbf{-13,1664}$	$\mathbf{-13,1464}$
$p = 15$	$-13,2240$	$-13,1128$	$-13,2240$	$-13,1938$
$p = 20$	$-13,1092$	$-13,9599$	$-13,1092$	$-13,0687$

Em todos os critérios, os três modelos possuem índices de valores bastante próximos. O modelo de ordem 10 é escolhido pelo princípio da parcimônia, um vez que para resultados bastante similares, o modelo de ordem 10 possui menos parâmetros. A comparação entre um quadro original e a estimativa obtida pelo modelo AR(10) com parâmetros estimados pela técnica de Yule-Walker é ilustrado na Figura 5.

Os parâmetros dos 31 filtros AR(10) são concatenados de forma a obter um vetor de atributos de dimensão $P = 310$ elementos para os classificadores a serem aplicados na Seção VI. Assim, o banco de dados via LPC possui $N = 196$ amostras de $P = 310$ atributos.

C. *Parametrização do sinal de voz via FFT*

Para a criação de um outro banco de dados, é utilizada a FFT. Neste artigo, não se aplica a FFT no esquema de análise de curta duração, como na técnica STFT (*short-time Fourier transform*) [14]. Em vez disso, a FFT é aplicada sobre o sinal reamostrado inteiro. Ao assim proceder, a suposição de estacionariedade do sinal, necessária para a aplicação da FFT, não é satisfeita. Esta abordagem leva a uma perda na resolução e, consequentemente, na capacidade de reconstrução adequada do espectro do sinal de voz. Contudo, os autores apostam neste

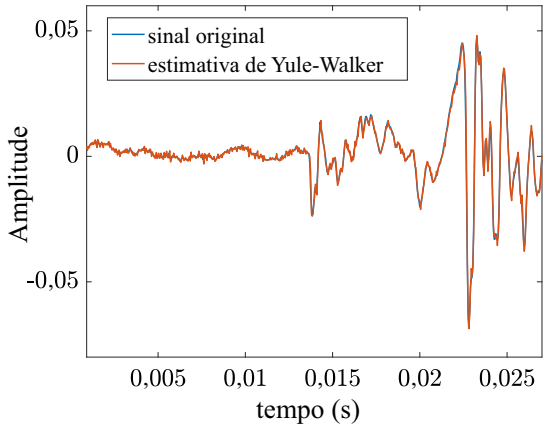


Figura 5: Modelagem de um quadro por modelo AR(10).

Tabela II: Extensão de frequência das faixas utilizadas.

faixa	**extensão** (Hz)	**faixa**	**extensão** (Hz)
1	$0 - 100$	8	$1.000 - 1.200$
2	$100 - 200$	9	$1.200 - 1.320$
3	$200 - 360$	10	$1.320 - 1.500$
4	$360 - 500$	11	$1.500 - 2.200$
5	$500 - 700$	12	$2.200 - 2.700$
6	$700 - 850$	13	$2.700 - 3.700$
7	$850 - 1.000$		

esquema alternativo sob o argumento de que o objetivo final da tarefa é a classificação de padrões, e não a reconstrução exata do espectro do sinal. Os resultados a serem mostrados adiante no artigo corroboram o bom desempenho desta abordagem.

Os sinais de áudio reamostrados em $22.050\,Hz$ são submetidos à FFT e seus espectros de frequência são analisados em busca de se estabelecer quais componentes de frequência devem ser utilizadas para compor o vetor de atributos. Após análise de diversas amostras, são definidas 13 faixas de frequência de interesse, conforme ilustrado na Figura 6 e na Tabela II. É escolhida então a maior amplitude de cada uma das 13 faixas. O fato de se definirem faixas de frequência em vez de frequências específicas torna o sistema robusto a pequenas variações e erros de frequência que possam ocorrer no processo de elocução, aquisição, reamostragem e aplicação da FFT. Assim, as 13 amplitudes são concatenadas em um vetor, que é utilizado como vetor de atributos para os classificadores a serem utilizados na Seção VI. Por fim, o banco de dados via Fourier possui $N = 196$ amostras de $P = 13$ atributos.

VI. Simulações e Resultados

São feitas 100 realizações independentes de cada classificador (LMQ e MLP), em que são analisados os valores máximo, mínimo, médio, mediano e desvio-padrão da taxa de acerto (TA) do conjunto de teste, além do posto e do número de condicionamento recíproco (rcond) da matriz de regressores do LMQ. Para parametrizar os classificadores, são utilizados 50% dos dados. O restante é utilizado para validar o classificador em um conjunto de teste. O resultado das aplicações dos classificadores nos dois bancos de dados formados por LPC e FFT são apresentados na Tabela III e na Figura 7.

As MLPs utilizadas neste trabalho obtiveram seus resultados com cinco neurônios na camada oculta, decaimento linear de η ($\eta_0 = 0{,}1$) e 150 épocas de treinamento para os dados obtidos via LPC e 300 para os dados obtidos via FFT. Os rótulos utilizados para o treinamento da MLP, dados pela Eq. (6), não são unitários e sim $\pm 0{,}98$, para evitar o fenômeno da paralisia da rede e pesos sinápticos de valor muito alto ou muito baixo. As amostras são normalizadas entre $\pm 0{,}98$.

Todos os classificadores, com exceção do classificador LMQ + FFT, obtiveram 100% de mediana na taxa de acerto. Pode-se observar que o classificador LMQ + LPC obteve a maior média e menor desvio-padrão ($99{,}96 \pm 0{,}41\%$), com apenas um *outlier* em $95{,}92\%$.

Porém, este classificador possui uma matriz $\boldsymbol{X}^T\boldsymbol{X}$ de 310×310 e posto $= 94$, o que justifica o baixo valor de $2{,}4 \cdot 10^{-22}$ do número de condicionamento recíproco. É importante mencionar que a estimativa da inversa de $\boldsymbol{X}^T\boldsymbol{X}$ foi obtida por decomposição em valores singulares (SVD). Sem este método, o desempenho do classificador seria prejudicado.

Visando melhorar o condicionamento e o posto desta matriz, aplica-se PCA para diminuir as dimensões dos atributos. Foi escolhida $\beta = 60$, que concentra $99{,}80\%$ da informação do conjunto de dados. Observa-se que a matriz $\boldsymbol{X}^T\boldsymbol{X}$ possui posto completo e seu número de condicionamento melhorou consideravelmente, mantendo a mediana de 100% de taxa de acerto, porém com uma pequena redução da média (de $99{,}96\%$ para $99{,}56\%$). É possível notar que há uma grande redução no número de parâmetros, de 1550 para 300.

É importante mencionar que os números de parâmetros na Tabela III se referem apenas aos classificadores, não contendo parâmetros de normalização dos dados ou parâmetros da matriz V_β do PCA, necessária para a transformação dos dados.

A aplicação do classificador MLP ao conjunto de dados por LPC obteve resultados similares ao LMQ, tanto em mediana, média e desvio-padrão, quanto em número de parâmetros. Diferenças entre o classificador MLP e LMQ podem ser visualizadas na Figura 7 apenas no número de *outliers*. A aplicação de PCA com $\beta = 60$ obteve resultados similares.

As classificações utilizando o banco de dados construído a partir da aplicação de Fourier apresentam excelentes resultados, visto que com apenas 13 atributos, o classificador MLP obteve média de $99{,}38\% \pm 1{,}58\%$. O classificador LMQ + FFT que obteve melhor resultado possui transformação Box-Cox com $\gamma = 0{,}1$, atingindo mediana de $96{,}94\%$, média de $96{,}90\% \pm 2{,}74\%$. Além disso, a matriz $\boldsymbol{X}^T\boldsymbol{X}$ possui posto completo e rcond de $8{,}6 \cdot 10^{-6}$. Este classificador possui o menor número de parâmetros entre os utilizados.

O gráfico da saída de dois neurônios da camada oculta da MLP + FFT é apresentado na Figura 8. Observa-se que as classes são bem separadas já na camada oculta. Embora as palavras "avançar" e "direita" estejam próximas no gráfico, no espaço de atributos da camada oculta (*hidden feature space*) estas elocuções estão bem separadas.

As matrizes de confusão da melhor e pior realização presentes na Figura 7 são apresentadas na Figura 9. Uma vez que cinco das seis diferentes classificações possuem mediana de

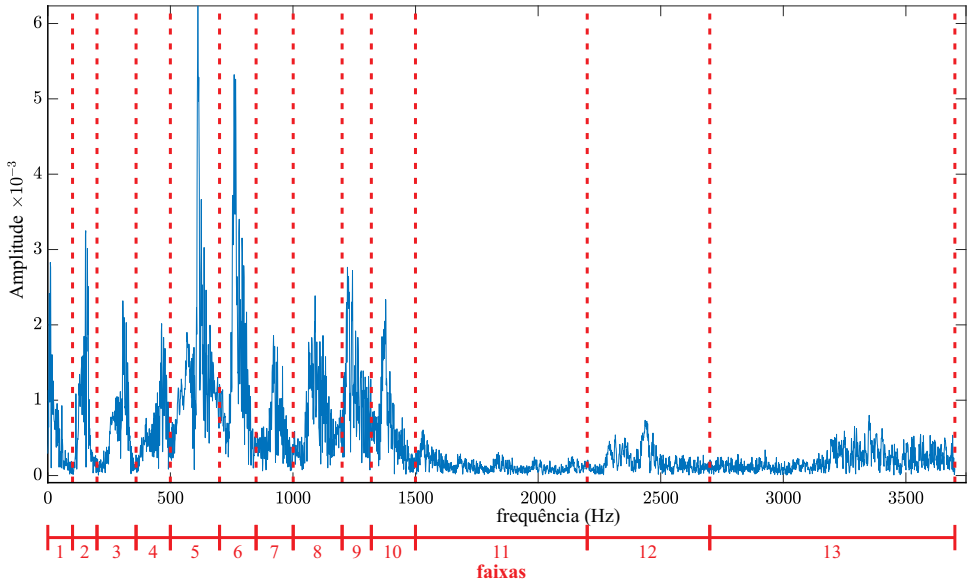


Figura 6: Faixas de frequências utilizadas para montar o banco de dados baseado em FFT.

Tabela III: Resultados dos classificadores.

classificador	**media DP** (%)	**mediana** (%)	**máximo mínimo** (%)	**número de parâmetros**	**atributos**	**posto**	**rcond**
LMQ + LPC	99,96 0,41	100	100 95,92	1.550	310	94	$2{,}4 \cdot 10^{-22}$
LMQ + PCA + LPC	99,56 1,35	100	100 92,86	300	60	60	$1{,}4 \cdot 10^{-5}$
LMQ + BC + FFT	96,90 2,74	96,94	100 89,80	65	13	13	$8{,}6 \cdot 10^{-6}$
MLP + LPC	99,70 1,05	100	100 95,70	1.585	310	—	—
MLP + PCA + LPC	99,16 1,71	100	100 92,47	335	60	—	—
MLP + FFT	99,38 1,58	100	100 91,40	100	13	—	—

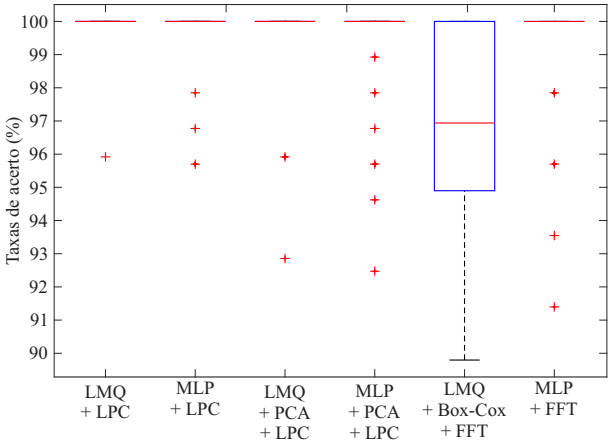


Figura 7: Boxplot dos classificadores.

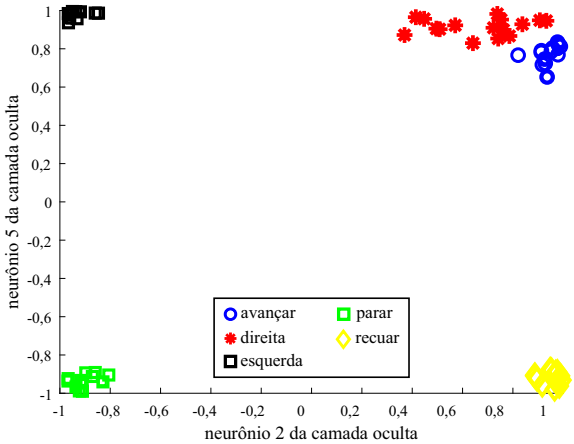
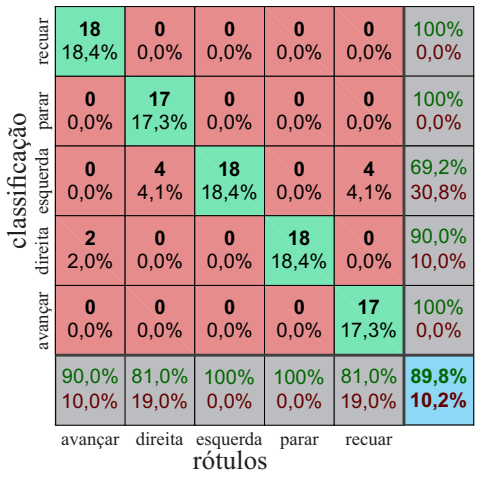


Figura 8: Saídas da camada oculta da para teste.

100%, a escolha do melhor classificador é feita a partir dos outros critérios presentes na Tabela III. O classificador MLP + FFT possui TA média de 99,38%, e apenas 100 parâmetros. Este é um fator importante a ser considerado para inclusão no *software* embarcado de controle do robô móvel.

Observa-se na Figura 9 que o classificador LMQ + BC + FFT erra apenas 10 das 98 amostras de teste, sendo duas amostras "avançar" confundidas com "direita", qua-



(a) LMQ + BC + Fourier.

(b) MLP + Fourier.

Figura 9: Matrizes de confusão da melhor e pior realização.

tro amostras "direita" confundidas com "esquerda" e quatro amostras "recuar" confundidas com esquerda. Já o classificador MLP + FFT acerta todas as amostras de testes, atingindo 100% de acerto em todas as classes.

VII. Conclusões

Este trabalho tratou do reconhecimento de comandos de voz para acionamento de um robô móvel. Dois bancos de dados foram construídos a partir das técnicas LPC e FFT aplicadas na parametrização dos sinais de áudio. São utilizados cinco comandos: "avançar", "direita", "esquerda", "parar" e "recuar". O banco de dados via LPC resultou em $P = 310$ atributos, em que cada áudio de voz é dividido em 31 quadros os quais são modelados por um filtro AR de ordem $p = 10$. Já o banco de dados via FFT ficou com $P = 13$ atributos constituídos das maiores amplitudes presentes em determinadas faixas de frequência (vide Figura 6).

Os classificadores LMQ e MLP foram aplicados nos bancos de dados criados. Os dois classificadores obtiveram excelentes resultados, com o classificador LMQ atingindo medianas da taxa de acerto de 100% para o banco de dados LPC e $96{,}94\%$ para o banco de dados FFT. Já o classificador MLP obteve mediana de 100% em ambos os conjuntos de dados. Visando reduzir o número de atributos do conjunto de dados LPC, foi aplicada a técnica PCA com redução de dimensionalidade. Assim, com $\beta = 60$ (que representam $99{,}80\%$ da informação contida nos dados originais), ambos os classificadores obtiveram 100% na mediana da taxa de acerto, porém com redução de 80% do número de parâmetros.

É importante mencionar que o processo de reamostragem duplicou o número de amostras, o que pode ter melhorado o processo de parametrização dos classificadores. Além disso, a hipótese levantada de que a FFT aplicada sobre o sinal inteiro (diferente do esquema mais usual de aplicá-la sobre segmentos curtos do sinal) levaria a bons desempenhos se confirmou.

Por fim, é importante notar que a palavra "parar" é reconhecida em 100% das amostras de testes, mesmo na pior realização. Isto é importante pois "parar" também é um comando de segurança para o robô em situações que podem envolver acidentes. Testes preliminares no sistema embarcado foram realizados com sucesso.

VIII. Agradecimentos

O presente trabalho foi realizado com apoio da CAPES (Código de Financiamento 001) e do CNPq(N° 309379/2019-9).

Referências

[1] A. H. Souza Júnior, "Avaliação de redes neurais auto-organizáveis para reconhecimento de voz em sistemas embarcados," Master's thesis, Universidade Federal do Ceará, 2009.

[2] G. K. Berdibaeva, O. N. Bodin, V. V. Kozlov, D. I. Nefed'ev, K. A. Ozhikenov, and Y. A. Pizhonkov, "Pre-processing voice signals for voice recognition systems," in *2017 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*. IEEE, 2017, pp. 242–245.

[3] M. V. Veloso, J. T. Costa Filho, and G. A. Barreto, "SOM4R: a middleware for robotic applications based on the resource-oriented architecture," *Journal of Intelligent & Robotic Systems*, vol. 87, pp. 487–506, 2017.

[4] H. Gupta and D. Gupta, "LPC and LPCC method of feature extraction in speech recognition system," in *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*. IEEE, 2016, pp. 498–502.

[5] B.-H. Juang, D. Wong, and A. Gray, "Distortion performance of vector quantization for LPC voice coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 2, pp. 294–304, 1982.

[6] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[7] G. Schwarz *et al.*, "Estimating the dimension of a model," *Annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[8] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[9] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[10] L. A. Aguirre, *Introdução à identificação de sistemas–Técnicas lineares e não lineares: Teoria e aplicação*, 4th ed., 2015.

[11] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[12] G. H. Dunteman, *Principal components analysis*. Sage, 1989, no. 69.

[13] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[14] C. Mateo and J. A. Talavera, "Short-time fourier transform with the window size fixed in the frequency domain (stft-fd): Implementation," *SoftwareX*, vol. 8, pp. 5–8, 2018.