

# Anti-balance Load Control System Applied to an Overhead Crane Prototype Activated by Voice Commands

C. G. Bernardo and L. E. M. de Lima

**Abstract**— This paper presents a proposal of anti-balance control applied to a didactic overhead crane, with the possibility of its activation by voice commands through a Graphical User Interface. It was opted for the application of fuzzy intelligent control for the load anti-balance and Proportional-Derivative (PD) classic control with Ziegler-Nichols (ZN) tuning for the car sidelong movement. The whole control is embedded in an ATmega 2560 microcontroller, responsible for commuting between the two forms of control, as well as accomplish the communication with the interface. Moreover, it was used the Mel-Frequency Cepstral Coefficients for the characterization of the voice signals and a Multilayer Perceptron artificial neural network for the classification of the extracted coefficients. After the accomplishment of the car movement tests and load balance analysis, it was verified that the fuzzy controller for the sidelong movement allied to the ZN tuning for the anti-balance PD control and the speech commands recognition by artificial neural network were effective for the studied application.

**Index Terms**— Artificial neural networks, Fuzzy control, Mel-frequency cepstral coefficients, Voice commands.

## I. INTRODUÇÃO

Indústrias de médio a grande porte geralmente possuem a necessidade de realizar o movimento de cargas e materiais entre os setores fabris. Em muitos casos, essas cargas são volumosas e pesadas, chegando a pesar dezenas ou até centenas de toneladas. Com isso, para que o transporte da carga seja possível, é comum o uso de pontes rolantes.

Nas atividades fabris, cargas de diversos tipos são movimentadas com o auxílio da ponte rolante. Muitas dessas cargas são valiosíssimas, e é imprescindível que o seu traslado seja feito de forma precisa e com o mínimo de oscilação para que a carga não sofra nenhum dano, nem ofereça risco de acidentes em função de uma eventual queda. Todavia, essa não é uma tarefa simples: na maioria dos casos, o controle da movimentação da carga por meio de uma ponte rolante é realizado de forma não-automática com a ação de um operador presente no chão de fábrica, e de acordo com sua experiência. Caso o operador não seja suficientemente experiente ou cometa erros [1], provavelmente a carga oscilará, podendo ter um aumento na amplitude no decorrer da movimentação. Isso proporciona risco de danos a serem causados à carga, bem como o risco de acidentes envolvendo o operador e demais trabalhadores. Além disso, outros fatores imprevisíveis podem

umentar a oscilação da carga, como a presença de ventos laterais na ponte (no caso de um sistema operando em ambiente externo) e deficiências de *design* e construção das pontes [2]. Essas oscilações impróprias podem ser contornadas por meio de um sistema de controle automático antibalanco da carga, atuando como parte do sistema de controle da ponte rolante, e permitindo que esta seja comandada remotamente pelo operador, afastando-o dos riscos de acidentes no chão de fábrica.

Algumas contribuições para solucionar o problema em estudo já foram desenvolvidas, como é o caso do estudo mostrado em [3], no qual foi construído um protótipo de ponte rolante (o mesmo utilizado neste trabalho), sendo aplicado um controle antibalanco, sem movimentos laterais, com uso de controle clássico e algoritmo PID. Entretanto, para sistemas muito complexos, uma abordagem puramente clássica pode inserir grande dificuldade no desenvolvimento do projeto de controle por conta da necessidade de uma modelagem matemática precisa. Uma alternativa é a utilização do controle *fuzzy*, que evita modelagens matemáticas complexas e se apoia na experiência do projetista [4]. Isso é indicado pelas contribuições dadas por [5][6], com o desenvolvimento de um sistema inteiramente baseado em lógica *fuzzy*, com o objetivo de controlar os movimentos de ponte rolante. Podemos ainda ver outras abordagens de controle apresentadas em [7][8], integrando a funcionalidade do algoritmo PID à compensação neural adaptativa em pontes rolantes e quadricópteros.

Portanto, para o trabalho apresentado nesse artigo, foi considerada a simplicidade de aplicação do controle *fuzzy* a sistemas complexos, sem desprezar a robustez do controle clássico, principalmente em sistemas de dinâmica veloz [9]. Com isso, uma solução apresentada nesse projeto para a eliminação automática do balanço de uma carga transportada por uma ponte rolante é o emprego de controle híbrido integrando controle *fuzzy*, para o movimento de translação, e o controle clássico, através do algoritmo Proporcional-Derivativo (PD) para eliminação do balanço, com uso de uma Interface Gráfica para o Usuário (GUI).

Uma das formas de acionamento propostas nesse projeto é a utilização de comandos de voz para indicar ao sistema qual a posição final desejada para a carga. Essa abordagem tem importante função como tecnologia assistiva [10][11], simplificando o controle da posição da carga e, conseqüentemente, melhorando as condições de trabalho de pessoas com deficiência visual e motora.

Para o desenvolvimento do projeto, foi utilizado um protótipo de ponte rolante, sendo composto de um pequeno carro que translada na direção do eixo longitudinal de uma estrutura construída em alumínio, além de uma carga içada

C. G. Bernardo, Instituto Federal do Espírito Santo (Ifes), Vitória, Espírito Santo, Brasil, christofergaldino2@gmail.com

L. E. M. de Lima, Instituto Federal do Espírito Santo (Ifes), Vitória, Espírito Santo, Brasil, luisedu@ifes.edu.br

com uso de uma haste conectada ao carro. Na Fig. 1, pode-se ver uma foto do protótipo que será utilizado, com seus respectivos componentes indicados.

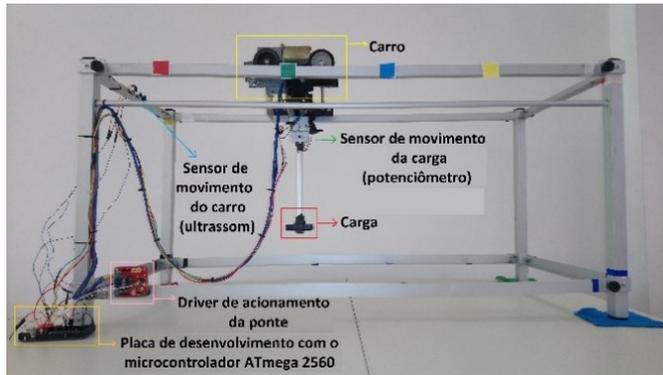


Fig. 1. Protótipo de ponte rolante utilizado no trabalho.

## II. DESENVOLVIMENTO

O desenvolvimento do projeto começa pela criação de uma GUI, com o objetivo de possibilitar o acionamento do carro da ponte rolante com as devidas configurações para a operação.

### A. Interface Gráfica do Usuário (GUI)

Para o desenvolvimento da GUI, foi utilizado o *software* MatLab® R2012b, em função da versatilidade e robustez dessa ferramenta em tarefas de prototipação.

A GUI desenvolvida pode ser visualizada na Fig. 2.

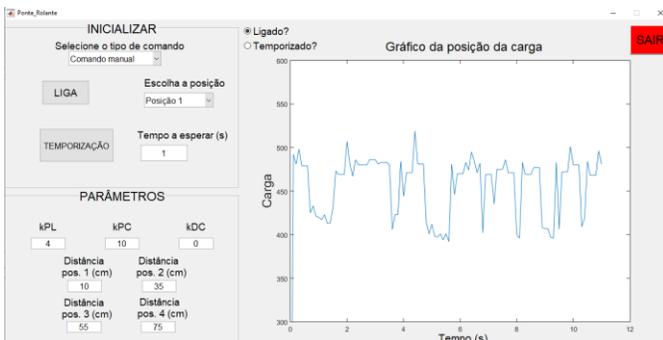


Fig. 2. Estrutura da GUI.

A interface possibilita dois modos de comando dos movimentos da ponte: comando manual e comando por voz. No modo de comando manual, tanto a definição dos valores dos parâmetros (posições e ganhos: kPL – ganho proporcional para o movimento de translação; kPC e kDC – ganhos proporcional e derivativo para antibalço da carga) quanto o acionamento do carro da ponte (através do botão “LIGA”) são feitos de forma manual na GUI. Ressalta-se que os valores referentes a cada posição da ponte têm como referência o sensor ultrassom responsável pela realimentação do movimento lateral do carro. No modo de comando por voz, o acionamento é feito a partir do comando de voz do usuário, que são nomes de cores de posições de interesse, cujos valores de posição são definidos na GUI e podem assumir qualquer valor ao longo da estrutura da ponte. Contudo, por fins de demonstração e visualização do sistema em funcionamento,

definiu-se indicadores coloridos em posições fixas na estrutura da ponte, conforme ilustrado na Fig. 3.



Fig. 3. Indicadores de cores na ponte para demonstração do comando por voz.

Quanto à operação, a ponte pode funcionar em dois modos, sendo um deles o modo de acionamento direto do carro, e o segundo modo o acionamento temporizado. No modo temporizado uma sequência de movimentos é definida e o carro moverá ininterruptamente entre as posições escolhidas para o movimento, respeitando um intervalo de espera entre uma posição e a seguinte. Esse intervalo é definido pelo usuário na seção “INICIALIZAR”, disponível na interface. A Fig. 4 apresenta o fluxograma do acionamento em modo temporizado.

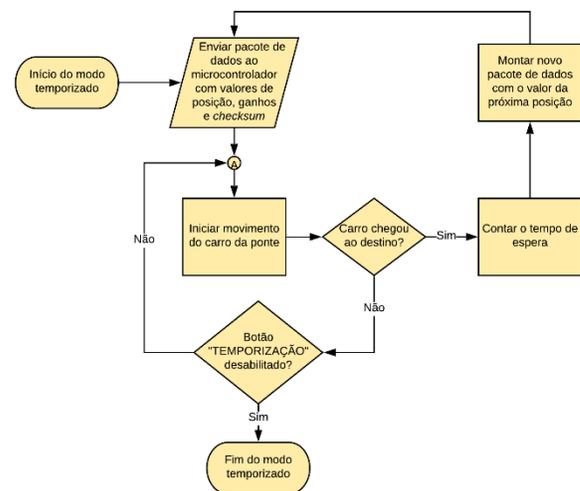


Fig. 4. Fluxograma do acionamento em modo temporizado.

### B. Projeto de Controle Híbrido

O controle híbrido para esse trabalho consiste em um controlador PD clássico para o controle antibalço da carga e controle *fuzzy* para translação do carro da ponte rolante.

#### 1) Controle *fuzzy* de translação da carga

O controle de translação é utilizado para execução do correto deslocamento da carga entre posição inicial e de destino, minimizando o tempo de deslocamento e auxiliando a correção da oscilação da carga. A escolha do paradigma *fuzzy* para o controle de translação reduziu o custo da modelagem necessária e permitiu resultados com precisão adequada.

Em função do conhecimento adquirido sobre a operação da ponte rolante, oriundo de outros projetos, escolheu-se o método de inferência Mandani para o controlador *fuzzy*. Com essa escolha, e considerando o uso da arquitetura de *hardware* ATmega 2560, é utilizada uma biblioteca de *software* livre que facilita a implementação do controlador *fuzzy*. A biblioteca aplicada nesse trabalho é a eFLL (*embedded Fuzzy Logic Library*) [12], que processa o método de inferência Mandani, a defuzzificação por centro de área, a formatação dos conjuntos *fuzzy* de entrada e saída e o conjunto de regras.

Para se definir qual o número ideal de funções de pertinência (FP) do controlador *fuzzy*, decidiu-se fazer testes com três diferentes configurações: controlador com três, cinco e sete FPs. Além da variação na quantidade de FPs, também foi testada uma variação no ganho do controlador proporcional. Os resultados obtidos serão apresentados nas seções seguintes. Para exemplificar, a Fig. 5 ilustra uma comparação dos conjuntos *fuzzy* de entrada e saída de um controlador com cinco FPs para diferentes ganhos proporcionais.

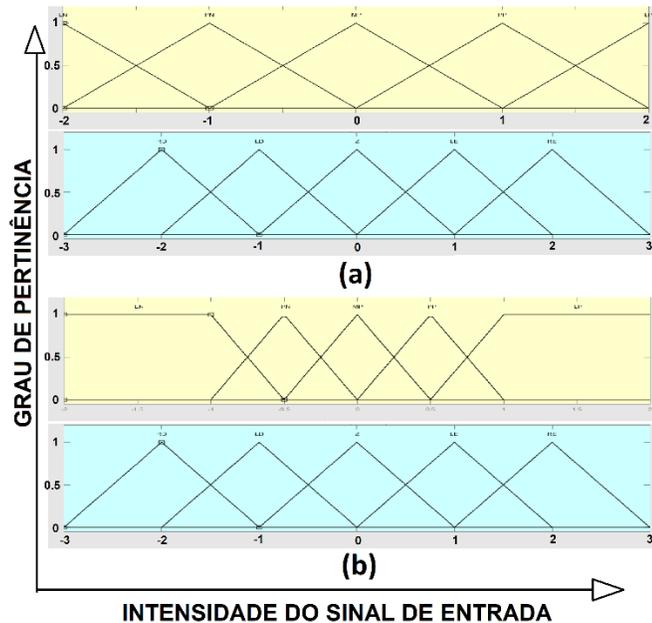


Fig. 5. Comparação dos conjuntos *fuzzy* de entrada e saída para controladores com cinco FPs e ganho proporcional (a) unitário e (b) valor 2.

## 2) Controle PD Antibalço da Carga

O projeto do controlador PD antibalço da carga teve como base o segundo método de sintonia Ziegler-Nichols (ZN) [13]. A escolha por esse método se deu com o objetivo de que a sintonia do controlador fosse definida por respostas dinâmicas reais do sistema através de ensaios práticos, minimizando a influência da estimativa de parâmetros na descrição de modelos analíticos.

A dinâmica veloz de oscilação da carga, sob influência dos torques gerados pela translação e a ação da gravidade, definiu o uso das ações de controle proporcional e derivativa. Além disso, como a variável a ser controlada é o ângulo de oscilação da carga, e sendo o sistema do tipo pêndulo simples, seu *setpoint* é a sua posição de repouso na vertical, assim a eliminação do erro em regime permanente é facilitada, pela própria ação da gravidade sobre o pêndulo.

Após a aplicação do segundo método de sintonia ZN foram definidos, de forma empírica, os valores dos ganhos proporcional e derivativo, sendo  $K_p = 22,2$  e  $K_d = 0,561$  respectivamente.

Considerando esses valores, realizou-se uma comparação com alguns resultados encontrados por [3], que utiliza o mesmo protótipo utilizado neste trabalho. Nesse outro projeto, por meio de uma abordagem puramente matemática, em que o autor buscou construir um modelo matemático do sistema com

o máximo de precisão, foram encontrados valores de ganhos proporcional e derivativo  $K_p = 0,4838$  e  $K_d = 0,0538$ . Contudo, como afirmado pelo próprio autor, alterações foram necessárias nesses valores, principalmente no ganho proporcional, já que o tempo de acomodação do sistema na simulação estava muito grande. Com isso, o  $K_p$  foi alterado para 40, valor bem diferente do encontrado inicialmente de forma teórica e mais próximo do encontrado neste trabalho, que é de 22,2.

Por fins de demonstração, simulou-se a resposta ao degrau do sistema para os ganhos encontrados neste trabalho, utilizando o modelo matemático e comparando o resultado fornecidos em [3], como visto na Fig. 6.

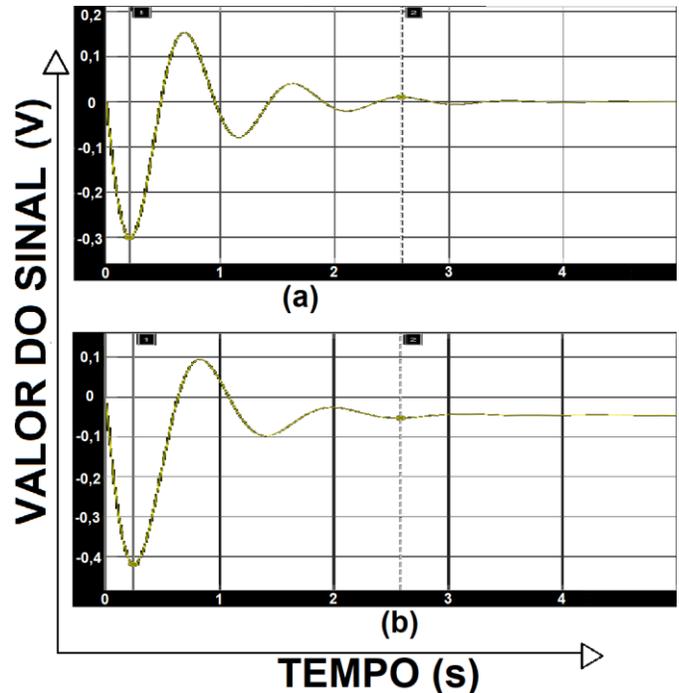


Fig. 6. Comparação entre as simulações da resposta ao degrau do sistema para (a) ganhos encontrados neste trabalho e (b) ganhos ajustados encontrados em [3].

Analisando a figura anterior, é possível observar que os resultados estão próximos, sendo a resposta obtida por [3] levemente menos oscilatória. Isso indica que algumas mudanças podem ser realizadas nas constantes encontradas anteriormente para este projeto, a fim de reduzir as oscilações e o tempo de acomodação. Portanto, a partir dos valores iniciais, e com a realização de testes no uso da ponte rolante, foram obtidos novos valores de constantes para o melhor desempenho, a serem apresentados na seção de resultados obtidos.

## 3) Integração dos controles no microcontrolador

Os algoritmos de controle PD e *fuzzy* estão integrados em um único processamento lógico embarcado. A comutação entre esses algoritmos é processada da seguinte forma: controle *fuzzy* para o movimento lateral, caso a carga esteja em posição estável de acordo com o seu eixo longitudinal, e controle PD para antibalço da carga sempre que há oscilação da mesma. Ou seja, se a carga balançar, o controle se ocupará por remover o balanço, e com a carga de volta à condição de

estabilidade, ele pode também voltar a controlar o carro da ponte rolante no seu movimento lateral (de translação).

O sensor responsável por indicar o balanço da carga é um potenciômetro, e o sensor responsável por indicar a posição do carro em seu percurso é um sensor de ultrassom HC-SR04. Os sinais obtidos desses sensores são processados sendo submetidos a uma filtragem do tipo média móvel, necessária para a eliminação de leituras espúrias por influência de ruído e variações oriundas da margem de erro dos sensores. Além do controle, o programa embarcado também é composto de rotinas de comunicação de dados que permitem a recepção dos comandos de movimentação da ponte definidos pelo usuário na interface MatLab®. A Fig. 7 mostra um fluxograma que resume o processo de controle embarcado no microcontrolador.

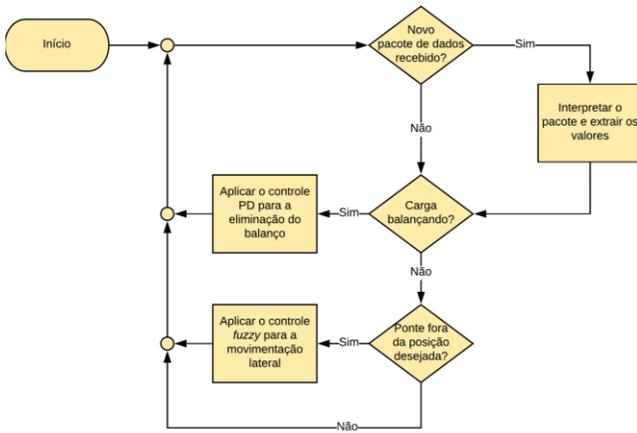


Fig. 7. Fluxograma do processo de controle embarcado.

### C. Reconhecimento de Comandos de Voz

Nesse trabalho é utilizado o método de obtenção dos coeficientes mel-cepstrais (MFCC's) do sinal de voz, para reconhecimento de comandos de movimentação da ponte rolante. A escolha pelo uso de MFCC's é devido à sua validação já comprovada em projetos de sistemas de reconhecimento automático de voz (RAV) [14][15]. É importante ressaltar que o sistema RAV implementado neste trabalho é do tipo dependente do locutor, ou seja, o funcionamento é garantido para o locutor que forneceu amostras de sua própria voz em testes realizados.

A sequência de processamento do sinal de voz, executada pelo sistema RAV implementado, foi estabelecida a partir de referências em projetos executados [16]. Essa sequência começa pela aquisição de dados, sendo seguida pelo pré-processamento, extração de características e, por fim, classificação. Na Fig. 8 está ilustrado o fluxo de execução do processo RAV, e nas próximas seções, cada uma dessas etapas será detalhada.

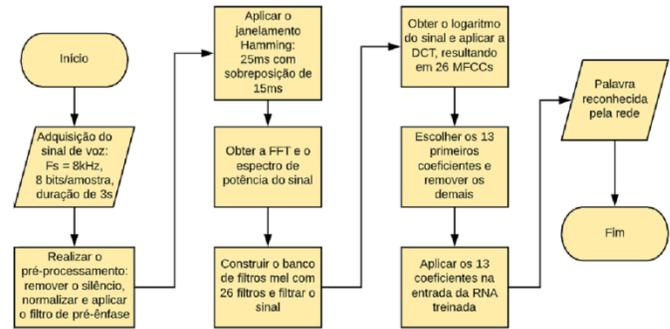


Fig. 8. Fluxograma do processo de reconhecimento de voz utilizado.

#### 1) Aquisição do Sinal de Voz e Pré-Processamento

A aquisição do sinal de voz foi feita com uso do software MatLab®, sendo gravados como comandos os nomes de cores de posições de referência, conforme ilustrado na Fig. 3, isto é, “vermelho”, “verde”, “azul” e “amarelo”. A quantidade de gravações foi igualmente dividida em áudios com diferentes velocidades de fala de modo a aumentar a robustez do sistema, e assim torna-lo capaz de realizar o reconhecimento independentemente da velocidade da fala do locutor, sendo ela rápida, normal ou lenta. Os áudios foram gravados com um microfone e convertidos a áudios “.wav” mono com 8 bits por amostra e uma taxa de amostragem de 8kHz.

O próximo passo foi a remoção de alguns blocos de silêncio. Para essa etapa, dividiu-se o sinal de voz em blocos de tamanhos iguais e removeu-se os blocos com energia menor que 1% da máxima. Isso reduz o tamanho do sinal de voz, bem como remove dados que não são úteis no reconhecimento, tornando o processamento mais rápido. Por exemplo, para uma gravação da palavra “amarelo”, o sinal de voz original, que continha 24.000 amostras, foi reduzido para um sinal de voz com 3.400 amostras, uma redução de, aproximadamente, 85%. Após essa redução, o sinal de voz é normalizado para valores de -1 a 1 e, em seguida, o filtro de pré-ênfase é aplicado a fim de equalizar o espectro. O coeficiente de pré-ênfase utilizado foi de 0,97.

#### 2) Análise espectral

Essa etapa inicia-se pelo janelamento do sinal e, para isso, o sinal é dividido em janelas de 25 milissegundos cada com sobreposição de 15 milissegundos, valores esses dentro do recomendado por Rabiner e Juang [17], sendo aplicada a janela de Hamming a fim de suavizar os efeitos das transições entre janelas adjacentes. Então, é realizada a Transformada Rápida de Fourier (FFT) e obtido o espectro de magnitude de cada janela. Com esse valor, é possível obter o espectro de potência para cada janela, isto é:

$$P(i) = \frac{M(i)^2}{n_{FFT}} \quad (1)$$

em que  $M(i)$  é o espectro de magnitude de uma dada janela  $i$ ,  $P(i)$  é o espectro de potência dessa mesma janela e  $n_{FFT}$  é o número de pontos da FFT, definido como 512 por ser um valor comum nesse tipo de aplicação.

#### 3) Extração de características

Com o sinal no domínio da frequência, o próximo passo é a obtenção dos MFCC's. Inicialmente é necessário que as frequências do sinal de voz em hertz sejam convertidas para a

escala mel, cujo propósito é representar de forma mais fiel a percepção auditiva humana [14][15]. Com isso, um banco de filtros triangulares mel composto por 26 filtros foi construído e, em seguida, foi realizada a filtragem de cada janela do sinal de voz, resultando em uma matriz com o número de linhas igual ao número de filtros, isto é, 26, e o número de colunas igual ao número de janelas. Os valores nessa matriz variam para cada sinal de voz devido às suas características, tendo em vista ainda que o número de blocos de silêncio removidos na etapa de pré-processamento pode variar para cada caso.

Sendo o cepstro definido em [18] como “o espectro de potência do logaritmo do espectro de potência” de um dado sinal, foi obtido o logaritmo do sinal filtrado para, então, a aplicação da transformada discreta do cosseno, DCT (*Discrete Cosine Transform*). A DCT transforma o sinal de voz de volta ao domínio do tempo, agora com suas componentes desconvoluídas e separadas, permitindo, assim, a obtenção dos MFCC's. Com a aplicação da DCT, foram entregues 26 MFCC's para cada janela; contudo, descarta-se os coeficientes de ordem superior, utilizando-se apenas com os 13 primeiros. Isso porque a DCT acumula a maior parte da informação do sinal nos coeficientes de ordem inferior, reduzindo custo computacional [16], além do fato de que os coeficientes de ordem inferior concentram as informações do trato vocal, desejado para realizar o reconhecimento de um comando de voz [19].

Portanto, tem-se um sinal representado agora por uma matriz cujo número de linhas é igual ao número de MFCC's, isto é, 13, e o número de colunas é igual ao número de janelas. Contudo, não é interessante, principalmente no momento de treinar a rede neural responsável pelo reconhecimento, ter o número de colunas variando para cada sinal. Então, para facilitar a análise e, posteriormente, o treinamento da rede neural, obteve-se a média de cada coeficiente para todas as janelas. Com isso, o resultado final seria um vetor com 13 coeficientes que seriam capazes de caracterizar um sinal de voz como um todo.

Na Fig. 9 é ilustrada uma comparação entre um exemplo de um sinal de voz correspondente à palavra “amarelo”, caracterizado pelos seus 13 MFCC's, e os MFCC's correspondentes a 100 diferentes gravações relativas à mesma palavra, sendo diferenciados no gráfico por cores de linhas diferentes. É interessante observar nessa figura que, embora alguns conjuntos de coeficientes apresentem diferentes magnitudes, eles seguem uma mesma forma, o que serve para caracterizar e possibilitar a classificação do sinal de voz. Na Fig. 10 é apresentada uma comparação dos MFCC's para sinais de voz correspondentes a duas palavras distintas, “vermelho” e “azul”. Percebe-se que os dois conjuntos de coeficientes facilmente se distinguem entre si nos seus formatos, confirmando que seriam possíveis o treinamento e a utilização de uma Rede Neural Artificial (RNA) a fim de realizar a classificação de um determinado sinal de voz.

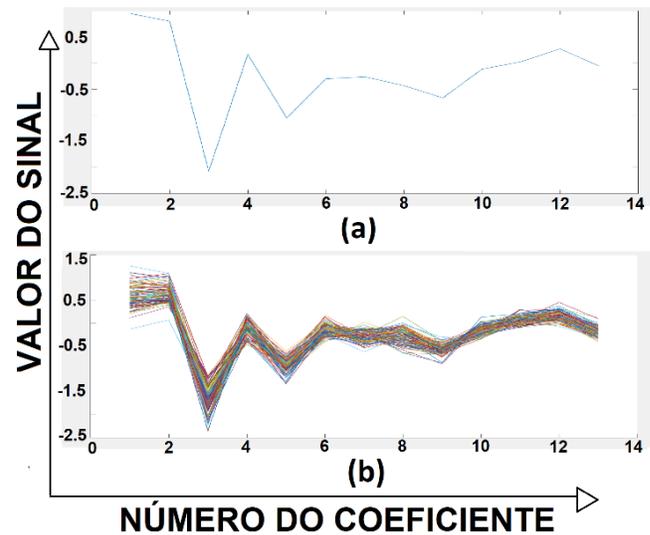


Fig. 9. Comparação de MFCC's correspondentes à palavra "amarelo" para (a) uma única gravação e (b) 100 diferentes gravações.

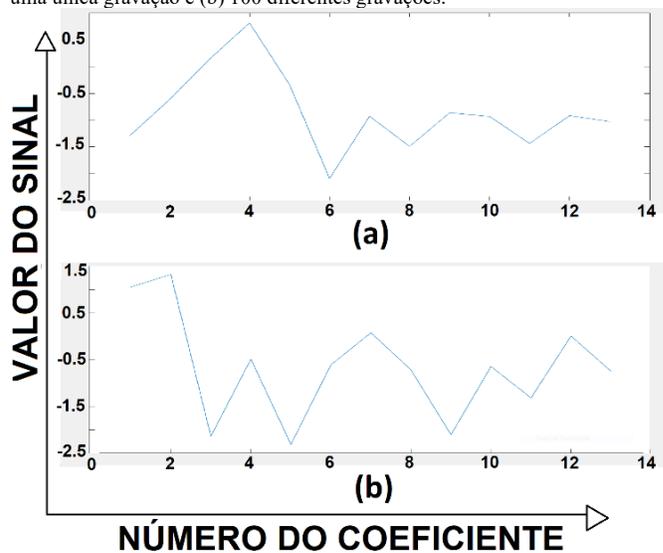


Fig. 10. Comparação de MFCC's para sinais de voz correspondentes às palavras (a) "vermelho" e (b) "azul".

#### 4) Classificação dos comandos por Rede Neural Artificial

Para a criação e treinamento da RNA empregada como classificador dos comandos de voz neste trabalho, fez-se uso da seção de reconhecimento de padrões e classificação da *toolbox* do MatLab® para redes neurais, *Neural Network Start*, ou *nnstart*.

A rede a ser usada é de arquitetura *perceptron* multicamadas com uma camada oculta, função de ativação sigmoide na camada oculta e função de ativação *softmax* para a camada de saída. Além disso, são aplicados no treinamento da rede o algoritmo *backpropagation*, o otimizador SCG (*Scaled Conjugate Gradient*) e o erro quadrático médio (MSE) como métrica de avaliação. A rede possui 13 entradas referentes aos 13 MFCC's característicos de cada sinal, e 4 neurônios na camada de saída, referentes às quatro cores possíveis de serem identificadas. A classificação do sinal se dá pela cor representada pelo neurônio com maior valor na saída. Para a definição do número de neurônios na camada oculta, foram testadas quatro topologias com diferentes números de neurônios nessa camada, sendo definidos por diferentes

critérios. O primeiro é a Regra Oja, que define o número de neurônios na camada oculta como:

$$H = T/5(N + M) \quad (2)$$

sendo H o número de neurônios na camada oculta, T o número de amostras de treinamento, N o número de neurônios na camada de entrada e M o número de neurônios na camada de saída [20]. O segundo é a Regra Hecht-Nielsen, que defende que H pode ser definido como  $2N + 1$  [21]. Os dois últimos são critérios empíricos, definindo H como  $N/2$  e  $2N/3 + M$  [14]. Os resultados dos testes para a definição do número ideal de neurônios na camada oculta serão vistos na próxima seção.

### III. RESULTADOS

Nas próximas seções serão apresentados os resultados referentes aos testes realizados a fim de se definir as configurações de controle, arquitetura e topologia da rede neural, além de permitir uma análise de desempenho do funcionamento do sistema.

#### A. Sintonia do Controlador Fuzzy

Os testes realizados para a definição da sintonia do controlador *fuzzy* tiveram como objetivo otimizar o desempenho do sistema com relação à velocidade de resposta, amplitude das oscilações da carga e erro de posição final do carro da ponte rolante em relação ao movimento de translação para controladores com diferentes ganhos e quantidade de FPs. Para a realização desses testes, controladores *fuzzy* com 3, 5 e 7 conjuntos *fuzzy* de entrada e saída foram programados e embarcados no microcontrolador. Então, realizou-se o controle do movimento lateral do carro em um trajeto de 60 centímetros, com distintos valores de ganho e análise de desempenho (velocidade e precisão) para cada situação. Destaca-se que apenas o movimento lateral é controlado nesses testes, isto é, a carga não está submetida a controle antibalanco. Além disso, por meio de ensaios prévios, confirmou-se que há significativa diferença de comportamento dinâmico do sistema em relação aos traslados de sentidos opostos, o que determinou a realização de testes e análise em ambos os sentidos de deslocamento.

Após a análise dos dados de ensaio, chegou-se à conclusão que a configuração 3 conjuntos *fuzzy* e ganho 10 é a adequada considerando-se o desempenho em ambos os sentidos de deslocamento, apesar da oscilação da carga estabelecida na partida, sendo compensada pelo reduzido balanço na parada. A TABELA I e a TABELA II apresentam os dados comparativos dos ensaios para os percursos de ida e volta, respectivamente. Ainda, será visto na próxima seção que a ação do controle antibalanco reduz consideravelmente o tempo de acomodação do sistema.

TABELA I  
DADOS COMPARATIVOS DE CONTROLADORES FUZZY COM 3, 5 E 7  
CONJUNTOS PARA EXCURSÃO DE IDA

Nº conjuntos fuzzy	Kp	Tempo de excursão (s)	Sobressinal máximo (%)	Subsinal máximo (%)	Tempo de acomodação
3	1	6,36	2,51	-4,63	1,255
	2	4,67	3,42	-6,69	1,715
	3	4,11	3,19	-6,77	1,395
	5	3,64	3,4	-6,6	1,37
	10	3,25	3,28	-6,65	1,4
	20	3,01	4,32	-6,68	1,395
5	1	6,87	2,81	-4,72	1,305
	2	4,94	3,97	-5,95	1,35
	3	4,33	4,36	-5,62	1,34
	5	3,7	4,07	-5,93	1,395
	10	3,28	4,49	-5,54	1,32
	20	2,98	4,8	-6,3	1,4
7	1	7,09	3,5	-4,98	1,35
	2	4,96	4,06	-5,82	1,325
	3	4,33	3,96	-5,98	1,38
	5	3,69	4,02	-5,92	1,325
	10	3,34	3,41	-6,52	1,395
	20	3,01	4,83	-6,27	1,375

TABELA II  
DADOS COMPARATIVOS DE CONTROLADORES FUZZY COM 3, 5 E 7  
CONJUNTOS PARA EXCURSÃO DE VOLTA

Nº conjuntos fuzzy	Kp	Tempo de excursão (s)	Sobressinal máximo (%)	Subsinal máximo (%)	Tempo de acomodação
3	1	5,86	4,39	-4,21	1,68
	2	4,37	5,86	-4,92	1,795
	3	3,8	5,78	-5,12	1,915
	5	3,19	5,36	-5,17	2,595
	10	2,9	5,23	-4,97	3,005
	20	2,65	5,22	-6,64	3,93
5	1	6,15	4,55	-4,05	1,42
	2	4,56	5,39	-4,9	1,78
	3	3,86	6,02	-5,11	1,865
	5	3,36	5,89	-5,32	2,725
	10	3	6,3	-5,05	2,93
	20	2,68	6,15	-5,68	4,405
7	1	6,16	4,92	-4,33	1,765
	2	4,48	5,67	-5,36	1,86
	3	3,89	5,32	-5,56	1,91
	5	3,37	5,62	-5,41	2,675
	10	3,06	5,52	-5,22	2,995
	20	2,66	5,34	-6,47	4,7

#### B. Sintonia do Controlador Antibalanco

Definida a sintonia do controlador *fuzzy*, a próxima etapa consiste na definição dos ganhos ideais para o controlador antibalanco. Para isso, considerando que o carro se movimenta ao longo da extensão da ponte e a carga pode se mover apenas no eixo longitudinal do carro, foram realizados testes semelhantes aos anteriores, com a inclusão do controle antibalanco na execução dos movimentos da carga. Com isso, os testes foram realizados inicialmente sem as ações de controle antibalanco e, em seguida, com uso de alguns valores de ganhos. Por exemplo, o  $K_p$  assumiu os valores 1, 2, 5, 10 e 22,2 (valor estabelecido pelo método ZN), enquanto que o  $K_d$  assumiu os valores 0, 0,25, 0,56 (valor estabelecido pelo método ZN), 1 e 2.

Nesses testes, constatou-se que a resposta é bem semelhante para todos os pares de ganhos avaliados. Isso acontece porque o controlador antibalço atua basicamente nas oscilações de parada e partida, sendo esta última a mais significativa. Todavia, essas oscilações são de pequena magnitude, e o caso mais crítico sem controle antibalço apresentou oscilações que não chegaram nem a  $0.1745\text{rad}$  ( $10^\circ$ ) de amplitude. Devido a isso, o comportamento da carga submetida a um controlador antibalço não varia com a mesma intensidade da variação dos ganhos. Para que fosse possível observar grandes distinções entre os valores, seria necessária uma análise para oscilações muito maiores que  $0.1745\text{rad}$  ( $10^\circ$ ), o que foge não apenas da realidade de um controlador antibalço aplicado a pontes rolantes como também dos modelos e aproximações assumidos para movimento pendular [22]. Portanto, optou-se por utilizar os valores de ganho sugeridos pelo método de sintonia ZN, isto é,  $K_p = 22,2$  e  $K_d = 0,56$ .

O desempenho obtido para uso do sistema com controle antibalço está ilustrado na Fig. 11 e na TABELA III, sendo apresentada a comparação entre as respostas sem e com controle antibalço. É perceptível que, desde a partida, o movimento da carga é caracterizado por uma oscilação de amplitude e tempo de acomodação maiores sem controle, enquanto que para o caso com controle há uma significativa redução desses parâmetros no movimento da carga e, conseqüentemente, no tempo de acomodação.

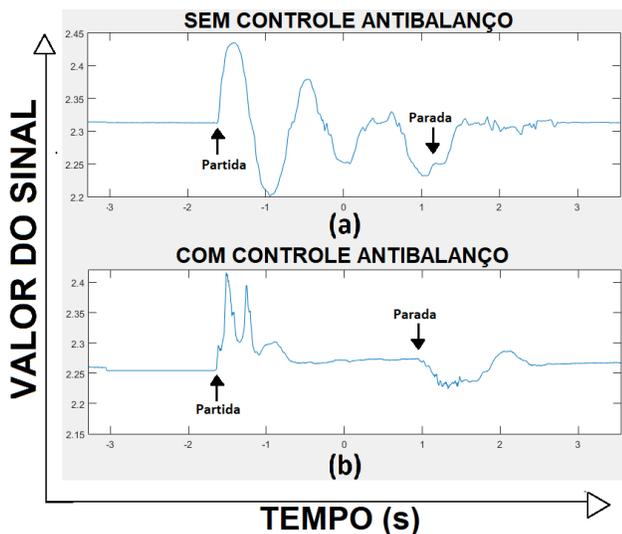


Fig. 11. Comparação das respostas da carga para movimento lateral do carro (a) sem controle antibalço e (b) com controle antibalço da carga.

TABELA III  
COMPARAÇÃO DOS TEMPOS  
DE ACOMODAÇÃO (EM SEGUNDOS)  
DO SISTEMA SEM E COM CONTROLE  
ANTIBALANÇO

Sem antibalço		Com antibalço	
Ida	Volta	Ida	Volta
1,4	3,0	0,48	0,47

Na Fig. 12 são ilustradas curvas obtidas para uma movimentação do carro em um percurso de volta e apresenta a atuação do controlador, mostrada na curva verde, de acordo

com a movimentação da carga, mostrada pela curva amarela. Os picos positivos do controlador representam a movimentação do carro para a esquerda, enquanto que os picos negativos representam a movimentação para a direita. Já para a curva da carga, uma oscilação positiva representa uma oscilação para a direita por parte da carga. Com base nisto, algumas análises podem ser feitas. Primeiramente, o pico inicial de atuação do controlador é positivo, confirmando que o carro está no sentido de volta, isto é, se movimentando para a esquerda. Então, pouco tempo após a partida do carro, devido à inércia, a carga apresenta uma oscilação para a direita. Para corrigir isto, é necessário que o carro interrompa o movimento anterior (para a esquerda) e se mova no sentido da oscilação da carga (para a direita) a fim de eliminar o balanço. Isso é exemplificado pelo primeiro pico negativo de atuação do controlador. Então, a oscilação da carga é mitigada, permitindo que o carro volte a se movimentar no sentido da excursão de volta. Contudo, assim que o carro volta a se movimentar, a carga apresenta um novo balanço, agora de amplitude reduzida. Então, o processo anterior de eliminação do balanço se repete, como pode ser visto pelos picos em sucessão, até que a carga se estabilize e o carro possa se movimentar para o seu destino final, sem interrupções, como visto a partir do último pico.

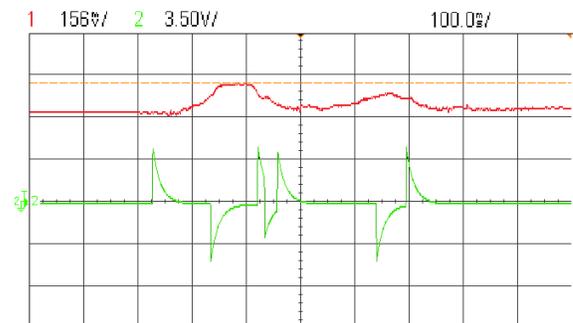


Fig. 12. Gráfico da atuação do controlador (em verde) de acordo com a movimentação da carga (em amarelo).

### C. Reconhecimento dos Comandos de Voz com uso de RNA's

Os testes foram realizados com diversas topologias de RNA's *perceptron* multicamada, buscando-se alcançar o melhor desempenho para a definição do sistema RAV utilizado neste trabalho. Os testes consistiram em, para cada topologia, submeter amostras de teste às redes criadas e treinadas com uso da *toolbox nntstart*, do MatLab® e, utilizando a acurácia como métrica de avaliação, verificar o percentual de acertos para cada palavra. Além disso, durante o processo de treinamento, foram registradas as quantidades de épocas que foram necessárias para cada topologia. Três testes desse tipo foram feitos, obtendo-se ao final as médias dos acertos e desvios padrões dos resultados, sendo esses mostrados na TABELA IV destaca-se que 100 amostras de áudio por cor foram utilizadas para o treinamento (400 no total) e 15 amostras de áudio por cor foram utilizadas para validação (60 no total).

Analisando as tabelas, é interessante observar que os erros de reconhecimento aconteceram exclusivamente para as palavras “vermelho” e “verde”. Além disso, os erros de classificação da palavra “vermelho” aconteceram porque esta

palavra foi classificada como “verde”, e vice-versa; todavia, essas duas palavras nunca foram confundidas com as palavras “azul” e “amarelo”. Isso pode ser explicado pelo fato de que, enquanto as pronúncias das duas últimas palavras são bem distintas, facilitando o reconhecimento pela RNA, as pronúncias das primeiras duas palavras são semelhantes, principalmente nas primeiras sílabas, ocasionando eventuais erros de reconhecimento.

TABELA IV  
MÉDIA E DESVIO PADRÃO POR COR DOS TRÊS TESTES DE DESEMPENHO PARA AS DIFERENTES TOPOLOGIAS DE REDES NEURAIS, CONSIDERANDO 100 AMOSTRAS DE TREINAMENTO E 60 AMOSTRAS DE TESTE.

Topol.	H	Vermelho		Verde		Azul		Amarelo	
		Média (%)	$\sigma$						
Oja Rule	4	77,78	16,63	88,89	6,28	100	0	100	0
Hecht-Nielsen	27	93,33	5,44	93,33	5,44	100	0	100	0
N/2	7	82,22	11,33	93,33	0,00	100	0	100	0
$2N/3 + M$	13	91,11	8,31	95,55	3,14	100	0	100	0

TABELA V  
RESULTADOS GERAIS DOS TRÊS TESTES DE DESEMPENHO PARA AS DIFERENTES TOPOLOGIAS DE REDES NEURAIS TREINADAS, CONSIDERANDO 100 AMOSTRAS DE TREINAMENTO E 60 AMOSTRAS DE TESTE.

Topol.	Média de acertos (%)	$\sigma$ geral	Número de épocas para treinamento
Oja Rule	91,67	9,21	32
Hecht-Nielsen	96,67	3,33	31
N/2	93,89	7,26	27
$2N/3 + M$	96,67	3,69	32

Na Fig. 13 é ilustrada a semelhança entre os MFCC's que caracterizam as palavras: “vermelho” e “verde”.

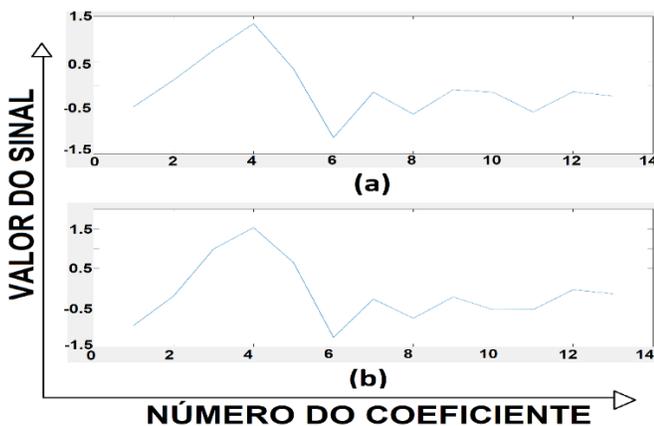


Fig. 13. Comparação de MFCC's para sinais de voz correspondentes às palavras (a) "vermelho" e (b) "verde".

Apesar dos erros, as redes possuem um desempenho satisfatório, já que a rede com o pior desempenho apresenta uma média de acertos superior a 90% (91,67%). Isso pode ser explicado pelo método de reconhecimento aplicada, isto é,

dependente de locutor. A quantidade de épocas para treinamento das redes não é muito variável; além disso, essas quantidades são pequenas em função da relativa simplicidade das topologias empregadas, com poucos neurônios em uma única camada oculta. Todas as redes criadas tiveram um treinamento bem rápido, com tempos que chegaram a, no máximo, 1 segundo.

Analisando a média de acertos das redes, pode-se destacar duas RNA's com melhor desempenho, sendo essas as redes com  $H = 27$  (Hecht-Nielsen) e  $H = 13$  (empírica com fórmula  $2N/3 + M$ ), ambas com 96,67% de média de acertos. O fator preponderante para a escolha de uma delas é a quantidade de neurônios na camada oculta. A rede  $2N/3 + M$  possui menos que a metade da quantidade de neurônios na camada oculta da rede Hecht-Nielsen. Logo, optou-se por aplicar a rede de topologia  $2N/3 + M$  ao sistema RAV, pois apresenta desempenho satisfatório à aplicação em estudo e menor complexidade topológica.

Foram realizados treinamentos com 100 amostras de áudio por cor para proporcionar robustez ao treinamento da RNA e, conseqüentemente, satisfatório desempenho de classificação dos comandos de voz. Contudo, para ampliar a validação do sistema RAV, também foram utilizadas quantidades menores de amostras, de forma que a RNA também foi treinada com 50 e 20 amostras. Os resultados são vistos na

TABELA VI.

TABELA VI  
COMPARAÇÃO DOS RESULTADOS DOS TESTES DE DESEMPENHO DA REDE DE TOPOLOGIA  $2N/3 + M$  QUANDO TREINADA COM 100, 50 E 20 AMOSTRAS DE TREINAMENTO

Nº de amost. de trein.	Acertos (%)				Média de acertos (%)	$\sigma$ geral	Nº de époc. de trein.
	Vermelho	Verde	Azul	Amarelo			
100	91,11	95,55	100	100	96,67	3,69	32
50	91,11	93,33	100	100	96,11	3,97	27
20	86,67	88,89	93,33	100	92,22	5,07	14

É possível verificar uma pequena queda no desempenho geral da rede conforme o número de amostras de treinamento cai. Essa queda de desempenho é fruto de uma redução nas classificações corretas das palavras “azul” e, principalmente, “vermelho” e “verde”, casos mais críticos como já fora relatado. Apesar da queda de desempenho, a rede ainda mantém uma boa classificação geral, com médias de acertos acima de 90%. Entretanto, para uma outra aplicação que venha requerer uma classificação de mais cores, é válido considerar realizar o treinamento com menos amostras.

#### D. Testes com o Sistema Integrado

A partir dos procedimentos e conclusões já apresentadas, tornou-se possível a realização de testes com todo o sistema integrado. Para isso, todos os comandos e definições de valores de ganhos e posições foram realizados com uso da GUI, sendo realizada uma análise com relação ao tempo de resposta do sistema em função dos comandos utilizados, para os dois tipos de acionamento: por comando de voz e por comando manual. Para isso, mediu-se o tempo entre o acionamento da ponte e o início efetivo do seu movimento,

sendo que, no caso do acionamento por comando de voz, o tempo é medido a partir do momento em que a obtenção do sinal de voz se encerra. Cinco testes foram realizados para cada caso, obtendo-se ao final a média e o desvio padrão dos resultados. Os testes para acionamento por comando de voz apresentaram um resultado para tempo de resposta médio de 2,91 segundos, com desvio padrão de 0,1. Já os testes para acionamento por comando manual apresentaram um resultado para tempo de resposta médio de 0,2 segundo, com desvio padrão de 0,014. O acionamento por comando manual permite uma movimentação quase instantânea do carro, enquanto o acionamento por comando de voz causa um atraso no início do movimento do carro de aproximadamente 15 vezes maior. Isso pode ser explicado pela relativa simplicidade do processo computacional do acionamento manual em relação à complexidade do processo de acionamento por comandos de voz, conforme detalhado no decorrer desse trabalho.

O funcionamento do sistema integrado é apresentado através de um vídeo, podendo ser acessado pelo link: <https://www.youtube.com/watch?v=Fo2r0BO4p7E&t=1s>.

Além disso, todos os códigos fonte utilizados para o funcionamento do sistema foram disponibilizados livremente através do GitHub®, podendo ser acessados pelo link: <https://github.com/ChristoferGB/Controle-de-ponte-rolante>.

#### IV. CONCLUSÕES

Este trabalho agregou aos estudos e aplicações de sistemas automatizados já existentes, cumprindo com a proposta de desenvolver um controle antibalanco aplicado a uma ponte rolante didática, apresentando um diferencial que é a implementação de um controle híbrido, em conjunto com um sistema para reconhecimento de voz. O modelo utilizado é de pequena escala se comparado a uma ponte rolante real; contudo, o sucesso da aplicação didática abre o caminho para uma implementação em modelo real.

Comprovou-se que é possível integrar o controle dos movimentos laterais do carro de uma ponte rolante e o controle antibalanco de sua carga sem que se perca desempenho, isto é, mantendo um bom tempo de excursão e acomodação e uma diminuta oscilação na carga.

Ademais, constatou-se a eficácia da lógica e controle fuzzy na implementação do controle dos movimentos laterais do carro, mostrando-se uma boa alternativa à lógica booleana e ao controle clássico, principalmente quando o sistema é complexo de se modelar matematicamente e o projetista possui conhecimento prático no mesmo sistema.

Além disso, foi apresentado o comando por voz como uma boa alternativa no acionamento de sistemas, principalmente considerando a sua atuação como tecnologia assistiva. Conquanto que proporcione uma boa base dados, uma caracterização fidedigna do sinal de voz e um classificador adequado para a dada aplicação, um sistema RAV é uma excelente opção para automatização, podendo oferecer um desempenho altíssimo de reconhecimento, como visto nesse trabalho.

#### AGRADECIMENTOS

Os autores agradecem ao Instituto Federal do Espírito Santo por todo o suporte dado para essa pesquisa através da disponibilização de laboratórios, materiais e *softwares*.

#### REFERÊNCIAS

- [1] S. Mandal, K. Singh, R. K. Behera, S. K. Sahu, N. Raj, and J. Maiti, "Human error identification and risk prioritization in overhead crane operations using HTA, SHERPA and fuzzy VIKOR method," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7195–7206, 2015.
- [2] A. A. Marquez, P. Venturino, and J. L. Otegui, "Common root causes in recent failures of cranes," *Eng. Fail. Anal.*, vol. 39, pp. 55–64, 2014.
- [3] R. L. Baióco, "Sistema anti-balanço para transporte de cargas em pontes rolantes," B.S. final paper, Dept. Elec. Eng., Inst. Fed. do Espírito Santo, Vitória, ES, Brazil, 2012.
- [4] Y. Zhai, M. C. Li, J. Luo, Y. Zhou, and L. Liu, "Research of the motion balance of spherical mobile robot based on fuzzy control," *J. Vibroengineering*, vol. 17, no. 1, pp. 13–23, 2015.
- [5] A. A. Assis, "Desenvolvimento de um sistema de controle difuso antibalanco para um modelo didático de ponte rolante," B.S. final paper, Dept. Elec. Eng., Inst. Fed. do Espírito Santo, Vitória, ES, Brazil, 2015.
- [6] Y. Zhao and H. Gao, "Fuzzy-Model-Based Control of an Overhead Crane With Input Delay and Actuator Saturation," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 181–186, 2012.
- [7] W. Yu, M. A. Moreno-Armendariz, and F. O. Rodriguez, "Stable adaptive compensation with fuzzy CMAC for an overhead crane," *Inf. Sci. (Nj)*, vol. 181, no. 21, pp. 4895–4907, 2011.
- [8] C. Rosales, S. Tosetti, C. Soria, and F. Rossomando, "Neural Adaptive PID Control of a Quadrotor using EFK," *IEEE Lat. Am. Trans.*, vol. 16, no. 11, pp. 2722–2730, 2018.
- [9] K. Ogata, *Modern Control Engineering*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [10] D. Hernandez de la Iglesia, G. Villarubia Gonzalez, J. F. De Paz Santana, and J. Bajo Perez, "Design and implementation of a low-cost Universal Control for intelligent electric wheelchairs," *IEEE Lat. Am. Trans.*, vol. 16, no. 5, pp. 1328–1336, 2018.
- [11] B. Villeda Félix, J. H. Arroyo Nuñez, and R. S. Nuñez Cruz, "Displacement aid system for people with visual impairment using magnetic coils," *IEEE Lat. Am. Trans.*, vol. 18, no. 02, pp. 400–406, 2020.
- [12] D. Kridi, A. J. Alves, M. Lemos, and R. de A. L. Rabêlo, "Desenvolvimento de Uma Biblioteca Fuzzy para o Controle Autônomo de Um Robô Móvel em Ambiente Desconhecido," in *Iª Mostra Nacional de Robótica, At Universidade Federal de São João Del-Rei (UFSJ) - MG*, 2011, pp. 155–161.
- [13] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *J. Dyn. Syst. Meas. Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.
- [14] E. R. de S. Thiago, "Reconhecimento de Voz utilizando extração de Coeficientes Mel-Cepstrais e Redes Neurais Artificiais," B.S. final paper, Dept. Telecom. Eng., Inst. Fed. de Santa Catarina, São José, SC, 2017.
- [15] C. J. Siddhant and A. N. Cheeran, "MATLAB Based Feature Extraction Using Mel Frequency Cepstrum Coefficients for Automatic Speech Recognition," *Int. J. Sci. Eng. Technol. Res.*, vol. 3, no. 6, pp. 1820–1823, 2014.
- [16] M. Yuan, "Speech Recognition on DSP: Algorithm Optimization and Performance Analysis," M.S. thesis, Dept. Electron. Eng., Chinese Univ. of Honk Kong, Honk Kong, China, 2004.
- [17] L. R. Rabiner and B. W. Juang, *Fundamentals of Speech Recognition*, 1st ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.
- [18] B. P. Bogert, J. R. Healy, and J. W. Tukey, "The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking," in *Proceedings of the Symposium on Time Series Analysis*, 1963, pp. 209–243.
- [19] P. Taylor, *Text-to-Speech Synthesis*, 1st ed. NY, USA: Cambridge University Press, 2009.
- [20] W. Gevaert, G. Tsenov, V. Mladenov, and S. Member, "Neural networks used for speech recognition," *J. Autom. Control*, no. 20, pp. 1–7, 2010.
- [21] R. H. Hecht-Nielsen, "Kolmogorov's Mapping Neural Network

Existence Theorem,” in *Proceedings of the IEEE First International Conference on Neural Networks*, 1987, vol. 3, pp. 11–13.

- [22] C. Medina, S. Velazco, and J. Salinas, “Experimental control of simple pendulum model,” *Sci. Educ.*, vol. I, no. 1992, pp. 67–76, 2005.



**Christofer Galdino Bernardo** possui graduação em Engenharia Elétrica pelo Instituto Federal do Espírito Santo (2019), tendo trabalhado em projetos nas áreas de Sistemas Inteligentes, Controle em Sistemas Embarcados, Redes Neurais e *Smart Grid*.



**Luis Eduardo Martins de Lima** possui formação em Engenharia Elétrica 1992 pela Universidade Federal do ES - Brasil, Mestrado e Doutorado em Eng. Elétrica pela UFES-Brasil, Professor Titular do Instituto Federal do ES - Brasil desde 1996, atuando nas áreas de Sistemas Digitais, Modelagem e Controle de Sistemas Dinâmicos, Controle Fuzzy e Aplicações de Redes Neurais Artificiais.