

PosAutomação

Programa de Pós-Graduação em Engenharia de Automação e Sistemas

Inteligência Artificial



Equipe

Comissão editorial

- Felipe Cabral
- Julio Normey-Rico
- Livia Scheffer

Histórico de coordenações

- Eugênio B. Castelan / Edson R. De Pieri (2007 - 2010);
- José E. R. Cury / Eugênio B. Castelan (2010 - 2012);
- Jomi Fred Hübner / José E. R. Cury (2012 - 2014);
- Rômulo S. de Oliveira / Jomi Fred Hübner (2014 - 2016);
- Daniel F. Coutinho / Rômulo S. de Oliveira (2016 - 2018);
- Werner Kraus Junior / Jomi Fred Hübner (2018 - 2020);
- Werner Kraus Junior / Jomi Fred Hübner (2020 - 2022);
- Julio Normey-Rico / Max Hering de Queiroz (2022-2024).

Conteúdo

<u>Editorial</u>	<u>4</u>
<u>Uma breve introdução à inteligência artificial no PosAutomação</u>	<u>6</u>
<u>Redes Neurais Consistentes com Leis Físicas para Controle de Sistemas Dinâmicos</u>	<u>8</u>
<u>Estimativa do Estado de Carga de Baterias em Veículos Elétricos Utilizando Inteligência Computacional</u>	<u>12</u>
<u>Qualidade de software no desenvolvimento de agentes cognitivos</u>	<u>18</u>
<u>Aprendizado por imitação generativo adversarial para navegação autônoma em ambientes urbanos</u>	<u>22</u>
<u>Otimização de política com espaço de ações contínuo e limitado: superando o estado da arte no CarRacing</u>	<u>27</u>
<u>Inteligência artificial e justiça: predição de decisões para incentivar acordos em audiências de conciliação</u>	<u>32</u>
<u>Utilização de agentes BDI na operação de veículos aéreos não tripulados ...</u>	<u>39</u>
<u>Softbots inteligentes na Indústria 4.0</u>	<u>42</u>
<u>Aplicações de aprendizado de máquina em ensaios de compressores de refrigeração</u>	<u>50</u>

59 PosAutomação no
CBA2022



61 Time PosAutomação
Conheça a trajetória dos Professores

Ricardo Rabelo & Rômulo de Oliveira



Editorial

Nesta quinta edição da nossa revista, temos uma novidade: trocamos o nome! Agora nosso programa será chamado carinhosamente de PosAutomação, e este será também o nome da nossa revista a partir de agora!

Neste número, a Revista do Programa de Pós-Graduação em Automação e Sistemas (PosAutomação) da Universidade Federal de Santa Catarina (UFSC) traz como tema central as pesquisas realizadas no programa em “Inteligência Artificial”, assunto bastante tratado na mídia e que vem impactando diversos setores da economia e da vida das pessoas. Apesar de bem popular, trata-se de um tema complexo, mas crucial para o desenvolvimento científico e econômico do país. O Brasil não tem ficado atrás do resto do mundo e muitas empresas têm adotado as tecnologias associadas ao conceito de Inteligência Artificial (IA) para desenvolvimento de produtos e serviços.

Todas as empresas e entidades que utilizam tecnologias digitais têm formado equipes ou contratado empresas especializadas para o desenvolvimento de novos sistemas de gestão, comercialização e atendimento baseadas em IA. Em paralelo, várias universidades do Brasil realizam pesquisas em IA, para diversas aplicações. No nosso programa, vários pesquisadores par-

ticipam de projetos na área, e nesta edição da revista trazemos vários artigos com os resultados gerados nos estudos e pesquisas realizadas.

Ainda, como é tradição da nossa revista, vamos conhecer a trajetória acadêmica, profissional e as experiências de vida de dois docentes do programa, que estão na nossa universidade há mais de 20 anos. E atenção, a partir desta edição a nossa série passa a se chamar “Conhecendo o time do PosAutomação”.

Finalmente, você também confere a participação dos discentes e docentes no último Congresso Brasileiro de Automática, o CBA 2022.

Inteligência

Artificial

A Inteligência Artificial conquista cada vez mais espaço em diversas aplicações de engenharia. Diversas aplicações atuais contam com algum tipo de algoritmo de aprendizado de máquina para funcionar. Nesta edição, você confere o que o Pos-Automação tem desenvolvido na área.

Uma breve introdução à inteligência artificial no PosAutomação

Autor: Prof. Jomi Hübner

O tema desta edição da revista é Inteligência Artificial (IA). A IA é uma área fascinante, até pelo que imaginamos a partir deste nome. Para muitos ela tem um certo ar de magia nos resultados que apresenta. Como todo mágico sabe, na verdade são truques que fazem brilhar os olhos do público, enquanto ingênuos espectadores.

Definir IA é muito complexo, e existem muitas visões. Eu gosto muito da visão do Guilherme Bitencourt, primeiro professor de IA do DAS. Para ele, brevemente, usamos IA para tentar resolver problemas para os quais não conhecemos bons algoritmos, problemas que eventualmente nós mesmos não sabemos resolver. A IA vem como uma muleta para ajudar nossas incapacidades (e como muleta, é ferramental/instrumental). Por exemplo, eu não sei implementar um programa que joga xadrez, até pelo fato de nem jogar bem; mas consigo usar técnicas de IA para me ajudar nessa tarefa. Brevemente, entrego o problema para uma "IA" e ela me ajuda a encontrar uma solução. Nesse caso, o produto final tem comportamento dito inteligente, mas a implementação vinda da "IA" é simples e só é possível porque eu especifiquei cuidadosamente o problema. Tenho ressalvas em dizer que "a máquina ganhou do Gasparov" (um dos melhores jogadores humano de xadrez), na verdade uma equipe gigante, com anos de pesquisa e recursos computacionais expressivos ganhou. Não deixa, claro, de ser um grande feito. O mérito, entretanto, está mais para a equipe do que para a máquina. A máquina foi uma ferramenta para essa equipe que não sabia jogar xadrez tão bem quanto o Gasparov.

Penso que visão similar se aplica para a "mágica" da tão famosa "machine learning". A inteligência desses sistemas vem de exemplos, de especialistas, de projetistas, que, com a ajuda da máquina, conseguem produzir soluções impressionantes. Mas, novamente, a fonte fundamental da solução é muito mais a equipe humana sendo a máquina e as técnicas de IA ferramentas. Certamente, como ferramenta, a IA tem ajudado a produzir resultados impressionantes.

As pesquisas em IA têm abordagens bem variadas, porém podem ser agrupadas em dois grandes grupos principais: IA simbólica e IA conexionista. De forma muito breve, o primeiro grupo parte da hipótese de que inteligência é resultado de uma mente, enquanto no segundo, que é resultado de um cérebro. No primeiro, a IA busca especificar um agente que tem uma base de conhecimento simbolicamente representada. Os objetos do mundo tem um símbolo que os representam e o agente pode "raciocinar" sobre eles. Essa abordagem tem resultados em aplicações como sistemas especialistas, IA explicável, e planejamento automático. No segundo grupo, a IA busca modelar a rede de neurônios do cérebro, com foco especial em processos de aprendizado/adaptação dessas redes. Os resultados têm, recentemente, sido bem expressivos, especialmente com o surgimento de novos modelos baseados em redes profundas e sua integração com aprendizado por reforço. Parte do sucesso dessa abordagem deve-se à disponibilização de ferramentas de aprendizado de máquina de fácil uso.

Obviamente, existem trabalhos que não se enquadram bem nestes dois gru-

pos. Por exemplo, temos a técnica de Algoritmos Genéticos (AG), onde a inteligência é resultado de um processo evolutivo similar ao que aconteceu com a espécie humana. Existe também um esforço de unir os dois grupos numa abordagem unificada. Por exemplo, nas iniciativas de IA neuro-simbólica.

No contexto da PosAutomação, desde sua fundação, a IA tem sido amplamente considerada. Por um lado, temos pesquisas que usam ferramentas de IA (tanto teóricas quanto práticas) na resolução de problemas ligados à automação, como, por exemplo, aplicações na indústria de petróleo e gás, drones, sistemas de refrigeração, processamento de imagens, indústria 4.0, etc. Por outro lado, temos pesquisas que desenvolvem tais ferramentas, como, por exemplo, linguagens de programação para agentes inteligentes, modelos de Redes Neurais para controle considerando a física do processo, modelos de softbots, etc. Nesta edição da revista, temos artigos reportando resultados nos dois casos.

No corpo docente da PosAutomação, temos vários professores ligados à temática de IA, seja como usuários das ferramentas ou desenvolvedores delas. Podemos nominar os seguintes professores: Eduardo Camponogara, Marcelo Stemmer, Rodolfo Flesch, Ricardo Rabelo, Werner Kraus, Leandro Becker, Eric Antonelo e Jomi Hubner; esses dois últimos concursados para a área. Esses professores orientaram várias dissertações e teses em um amplo espectro de temas de IA. Alguns dos artigos publicados nesta edição da revista são resultados recentes destes trabalhos.

Segue também uma lista de alguns projetos desenvolvidos ou em andamento relacionados com a IA que são conduzidos

por professores e alunos da PosAutomação.

- Distributed Optimization for Cooperative Machine Learning in Complex Networks (coordenador Prof. Eduardo Camponogara, financiado pela FAPESC)
- Planejamento, gestão e operação de sistemas de mobilidade urbana baseados em dados (coordenador Prof. Werner Kraus Junior, financiado pelo CNPq)
- Arquiteturas neurais para Modelagem e Controle de Sistemas Dinâmicos (coordenador Prof. Eric Aislan Antonelo)
- Sistemas e algoritmos de controle multiagentes (coordenador Prof. Jomi Hubner, financiado pela ANP/Petrobras).

Redes Neurais Consistentes com Leis Físicas para Controle de Sistemas Dinâmicos

Autores: Eric Aislan Antonelo e Eduardo Camponogara

Introdução

A competição no mundo globalizado de hoje é uma força motriz para a digitalização de processos industriais e de empresas, uma iniciativa conhecida como indústria 4.0. Nesse contexto, tanto a simulação quanto o controle de sistemas complexos do mundo real devem ser feitos de forma inteligente e eficiente. Para atingir esse objetivo, uma das ferramentas disponíveis advém do aprendizado profundo de redes neurais, que pode ser usado tanto na modelagem quanto no controle inteligente de plantas ou sistemas dinâmicos. Em particular, podemos usar redes neurais para construir modelos para serem empregados em Controle Preditivo ou *Model Predictive Control* (MPC).

MPC é uma técnica que se tornou padrão para controle multivariável na indústria e na academia. Desde a sua criação na década de 1970, o MPC tem sido aplicado com sucesso nas indústrias de petróleo e gás, aeroespacial e de processos, bem como em robótica. A ideia principal de MPC é controlar um sistema empregando um modelo preditivo: a cada iteração da malha de controle, um problema de otimização é resolvido usando um modelo da planta em uma abordagem de horizonte deslizante.

Existem dois casos principais que consideramos para os quais a aplicação prática do MPC ou mesmo apenas a simulação eficiente de um sistema dinâmico são um desafio: (a) dados históricos escassos ou insuficientes da planta real para construir um modelo de aprendizado de máquina suficientemente preciso; (b) quando a simulação numérica de um modelo preciso dado por Equações Diferenciais Ordinárias (EDOs) ou Equações Diferenciais Parciais (EDPs) é muito onerosa para ser considerada em uma aplicação em tempo real.

No entanto, uma abordagem recentemente introduzida para treinar redes neurais profundas usando leis da física pode ser apli-

cada em ambos os desafios acima mencionados. Chamamo-las por Redes Neurais Consistentes com Leis Físicas (*Physics Informed Neural Networks - PINN*). Para o primeiro desafio (a), assumimos que o conhecimento a priori construído previamente por especialistas ou empregando leis da natureza está disponível. Para (b), as PINNs podem ser usadas no lugar de soluções numéricas de equações diferenciais, inclusive com ganho de tempo de execução computacional, pois apresentam um tempo de inferência inferior ao da solução numérica de EDOs ou EDPs. Isto permite a extensão da aplicação de MPC para cenários de tempo real.

Redes Neurais consistentes com leis físicas

Uma PINN tradicional possui uma variável de tempo contínuo t como entrada e as variáveis de estado do sistema como saída y (Figura 1). O principal resultado dessa abordagem de PINN é que a necessidade de coleta de dados reais é reduzida, uma vez que o comportamento da rede neural é especificado por leis físicas dadas em termos de EDPs ou EDOs. Essas equações diferenciais, da forma:

$$\partial_t \mathbf{y} + \mathcal{N}[\mathbf{y}] = 0, \quad t \in [0, T]$$

são incluídas na função de custo do problema de aprendizado como operadores diferenciais não lineares da saída y da rede

$$\mathcal{F}(\mathbf{y}) := \partial_t \mathbf{y} + \mathcal{N}[\mathbf{y}]$$

constituindo um segundo termo da função de custo que regulariza o processo de aprendizado (Figura 2). Assim, a minimização do erro físico resulta no treinamento da rede neural de forma que suas saídas y respeitem as leis definidas na função $\mathcal{F}(\mathbf{y})$.

Efetivamente, essa abordagem permite resolver EDPs ou EDOs complexas usando apren-

dizado profundo, pois a saída y da rede representa a solução dessas equações.

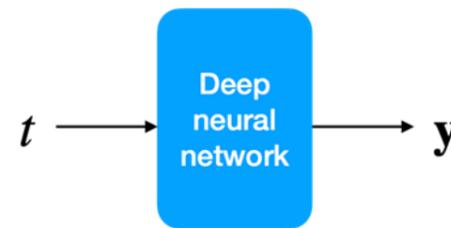


Fig. 1: PINN tradicional.

$$\begin{aligned} \text{MSE} &= \text{MSE}_y + \text{MSE}_{\mathcal{F}}, \\ \text{MSE}_y &= \frac{1}{N_y} \sum_{i=1}^{N_y} \frac{1}{N_t} \sum_{j=1}^{N_t} |y_i(t^j) - \hat{y}_i^j|^2, & \text{Data points} \\ \text{MSE}_{\mathcal{F}} &= \frac{1}{N_y} \sum_{i=1}^{N_y} \frac{1}{N_{\mathcal{F}}} \sum_{k=1}^{N_{\mathcal{F}}} |\mathcal{F}(y_i(t^k))|^2, & \text{Collocation points} \end{aligned}$$

Fig. 2: Função de custo MSE (erro quadrático médio) para PINNs composta do erro dos dados MSE_y e do erro físico $\text{MSE}_{\mathcal{F}}$. N_y é o número de estados do sistema dinâmico, enquanto N_t e $N_{\mathcal{F}}$ são o número de exemplos (dados) e o número de pontos de colocação em que o erro físico é avaliado, respectivamente.

Observe que a derivada de y com relação ao tempo t é calculada por diferenciação automática, um recurso disponível no *tensorflow*, ou *pytorch*.

$$\mathcal{F}(\mathbf{y}) := \partial_t \mathbf{y} + \mathcal{N}[\mathbf{y}]$$

Automatic differentiation
(*tensorflow*, *pytorch*)

```
with tf.GradientTape() as tape:
    y = network_output(t)
    dy_dt = tape.gradient(y, t)
```

Fig. 3: Exemplo de diferenciação automática.

No entanto, não há arquiteturas PINN em tempo contínuo na literatura que permitem que técnicas de controle ótimo como *Multiple Shooting* (MS) e *Model-based Predictive Control* (MPC) sejam diretamente aplicadas. O desafio é tornar PINNs compatíveis com aplicações de controle para que possam ser utilizadas como modelo preditivo de uma planta ou

processo. Em sua forma original, PINNs não permitem entradas de controle, nem podem simular por intervalos de tempo variáveis, ou seja, suas previsões se deterioram quando simuladas após o intervalo de tempo para o qual foram treinadas.

Redes Neurais consistentes com leis físicas para controle

Com essas limitações em mente, este trabalho apresenta um novo *framework* chamado *Physics-Informed Neural Nets for Control* (PINN), que propõe uma nova arquitetura baseada em PINN que é apta para aplicações de controle. Em particular: (i) nossa arquitetura baseada em PINN, daqui em diante chamada de rede PINN, é estendida com entradas extras, como o estado inicial do sistema e entrada de controle, além do tempo contínuo t .

Em nossa proposta, uma única rede aprende a solução de uma EDO condicionada ao estado inicial e à entrada de controle fornecida no intervalo mais curto.

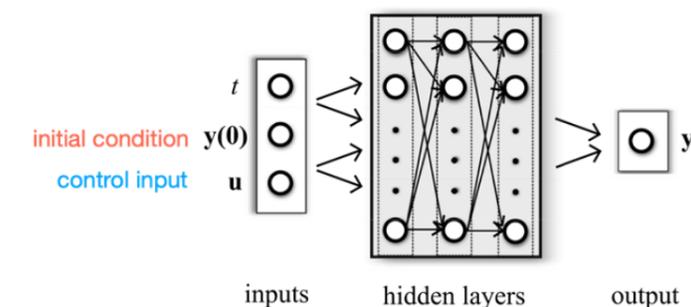


Fig. 4: Arquitetura PINN.

(ii) essa inovação permite aprimorar as capacidades de simulação de PINNs convencionais, que não podem sustentar corretamente uma simulação além do intervalo de tempo que foi fixado durante o treinamento da rede. Essa degradação da previsão da rede está relacionada ao valor máximo T permitido para a entrada t , que é fixado no momento do treinamento. Por outro lado, nossa rede PINN proposta pode simular uma EDO por um horizonte de tempo arbitrário, sem deterioração significativa das previsões. Por exemplo, para

simular o próximo intervalo $[T, 2T]$, o estado terminal do intervalo $[0, T]$ se torna o estado inicial do intervalo $[T, 2T]$, ou seja, a saída da rede no instante T é dada como entrada no próximo intervalo (Figura 4).

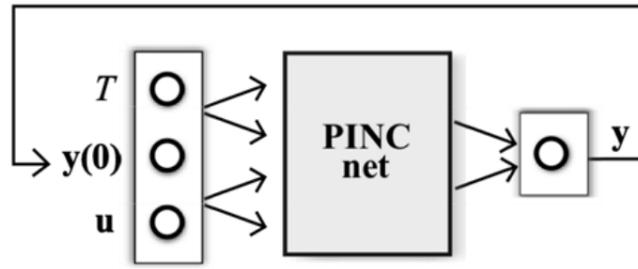


Fig. 4: Modo de operação realimentação: Estado terminal (saída y da rede) é atribuído ao estado inicial (entrada $y(0)$) no próximo intervalo de tempo na simulação.

(iii) a estrutura particular de PINC torna as redes consistentes com leis físicas em tempo contínuo aptas para aplicações de MPC.

(iv) finalmente, os requisitos de tempo real para simulação de equações diferenciais, em particular para aplicações de MPC, são melhor satisfeitos com PINC do que os métodos tradicionais de simulação numérica, uma vez que a inferência de uma rede PINC já treinada pode substituir um método de solução numérica em cada passo de tempo no horizonte de predição do problema de MPC.

Aplicação no Oscilador Van Der Pol

O oscilador de Van der Pol é uma EDO inici-

almente descoberta por Balthazar Van der Pol que tinha o propósito original de modelar oscilações de triodo em circuitos elétricos. Desde então, sua EDO tem sido usada para prova de conceito em aplicações de controle ótimo.

As equações que governam o oscilador de Van der Pol são as seguintes:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - x_1 + u \end{aligned}$$

onde $\mu = 1$ é referido como o parâmetro de amortecimento, que afeta o quanto o sistema irá oscilar, $x = (x_1, x_2)$ é o estado do sistema e u é um sinal de controle.

No gráfico abaixo, podemos ver como a nova proposta PINC pode simular por um intervalo de tempo arbitrário sem fixar esse tempo no treinamento da rede neural. Já a PINN tradicional precisa definir esse valor do intervalo de antemão no seu treinamento. As bolinhas nas trajetórias PINC, nas cores azul e rosa, marcam os momentos em que os estados finais preditos em $T=0,5s$ são realimentados como novas condições iniciais de entrada para a rede PINC. Embora a rede PINC seja treinada com um $T=0,5s$ fixo, sua predição encadeada (usando o modo realimentação) pode ser usada para realizar simulação de longo alcance para um tempo total de simulação arbitrário sem fixá-lo de antemão como ocorre

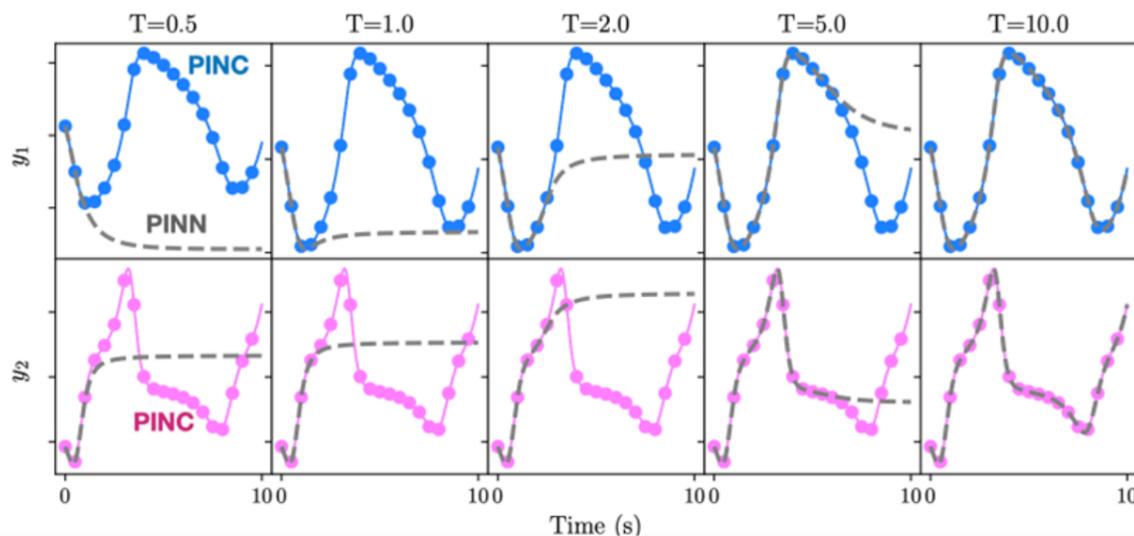


Fig. 5: Simulação por intervalo de tempo arbitrário: PINN adicional (cinza) e PINC (azul e rosa). Da esquerda para a direita, as redes PINN são treinadas com T fixo em $\{0.5, 1, 2, 5, 10\}$, enquanto a rede PINC é treinada apenas uma vez com $T=0.5s$, embora possa ser executada por tempos de simulação mais longos arbitrários não fixos de antemão.

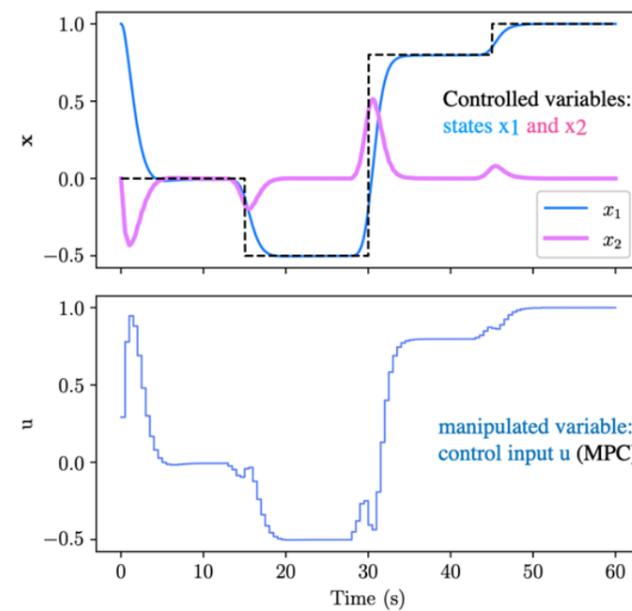


Fig. 6: Controle do oscilador Van Der Pol por 60s usando a rede PINC e NMPC. A trajetória de referência para x_1 (x_2) é dada por um sinal de passo preto tracejado, enquanto as variáveis controladas são os estados x_1 e x_2 dados pelas linhas azul e rosa. A entrada de controle u é a variável manipulada no gráfico inferior, definida por MPC.

nas PINNs tradicionais, cuja trajetória é mostrada pelas linhas tracejadas em cinza no mesmo gráfico. Observe que a trajetória real alvo do sistema dinâmico, desenhada em linha sólida preta, é completamente sobreposta pela trajetória PINC. Assim, a PINN tradicional não consegue extrapolar predições para $t > T$. Após o treinamento da rede segundo a metodologia proposta, o controle pode ser visto na Figura 6, onde a rede PINC treinada é usada em NMPC, servindo como modelo preditivo no problema de otimização.

Conclusão

Atualmente, temos pesquisadores atuando com a proposta PINC em problemas de controle industrial, como o controle de poços na indústria de petróleo e gás, dentro de uma cooperação bilateral entre NTNU (Noruega) e DAS-UFSC (Brasil).

Há perspectivas e interesses de uso e expansão da PINC dentro do CENPES-Petrobras em problemas a serem ainda definidos, especialmente aqueles em que conhecimentos prévios de equações diferenciais são conhecidos em conjunto com dados coletados de uma planta. Isso permitirá não somente

uma simulação em um intervalo de tempo variável, constituindo uma alternativa ao método numérico, mas também o controle desses sistemas dinâmicos. Havendo dados históricos disponíveis, a abordagem apresentada tem o potencial de integrá-los com leis físicas conhecidas, ao invés de somente simular as equações diferenciais já conhecidas como em métodos numéricos.

Para mais informações, consulte o artigo que descreve a proposta mais detalhadamente em:

<https://arxiv.org/abs/2104.02556>

Esta pesquisa foi desenvolvida inicialmente por:

Prof. Dr. Eric Aislan Antonelo
 Prof. Dr. Eduardo Camponogara
 Prof. Dr. Laio Oriel Seman
 Eduardo Rehbein de Souza
 MSc. Jean P. Jordanou
 Prof. Dr. Jomi Hubner

Estimativa do Estado de Carga de Baterias em Veículos Elétricos Utilizando Inteligência Computacional

Autores: Caio César Branco Nunes, Lucas Simoes Borges, Marco Antônio Silva Pereira e Roberto Willoughby.

Resumo

Este trabalho desenvolveu métodos para estimar o estado de carga em baterias de íons-lítio presentes em veículos elétricos, utilizando inteligência computacional. Apesar de existirem várias formas de aferição da quantidade de energia presente nestas baterias, muitas são imprecisas e não apresentam os resultados em tempo real. Diante disso, com um amplo banco de dados, BATTERY AND HEATING DATA IN REAL DRIVING CYCLES, dos autores Matthias Steinstraeter, Johannes Buberger, Dimitar Trifonov, que foram coletados por sensores instalados no carro BMW I3 (60hA) em viagens de teste, foi possível implementar modelos para previsão do estado de carga associado ao uso da bateria do veículo em estudo. Estes modelos foram desenvolvidos por técnicas de inteligência computacional no software Jupyter, tais como, Regressão Linear Múltipla, Árvore de Decisão e Redes Neurais Artificiais.

Introdução

A frota de veículos elétricos está em ascensão a cada ano que passa desde a proposta apresentada por uma comissão europeia, de acordo com AQUINO (2021), que visa a retirada de carros a combustão em um prazo de 15 anos. Tal proposta prevê a proibição da venda de veículos que emitem gás carbônico a partir de 2035, entrando em vigor com o Acordo de Paris (BRASIL,2017).

A utilização dos veículos elétricos apresenta inúmeras vantagens, destacando-se: a baixa emissão de poluentes e gases de efeito estufa em seu funcionamento, reduzida emis-

são de ruídos e elevado rendimento energético. As baterias presentes em veículos elétricos têm como um dos principais parâmetros o estado de carga, traduzido do inglês state of charge (SOC), que representa o nível de carga da bateria. Quando o estado de carga da bateria é 100% quer dizer que a bateria está totalmente carregada e quando o estado de carga é 0% a bateria está totalmente descarregada. Esse parâmetro específico tem impactos na operação das baterias e estimar seu estado pode evitar acidentes, que podem ser desde a pequenas chamas até explosões e incêndios de grandes proporções. Além disso, uma estimativa adequada do estado de carga de uma bateria é fundamental no gerenciamento da autonomia do veículo.

Portanto, estimar o estado de carga das baterias, que pode ser entendido como a quantidade de energia remanescente, é crucial para que se viabilize de maneira eficiente e segura a utilização das mesmas. Implementar um método de estimação preciso ainda é um desafio, sendo possível utilizar diversas metodologias baseadas, principalmente, em aprendizado de máquina (RAHIMI-EICHI et al., 2013).

Diante desse cenário,este trabalho buscou estimar o estado de carga para uma bateria de íon de lítio, considerando diferentes variáveis medidas em um veículo elétrico. Para alcançar esse resultado, é proposto o desenvolvimento de modelos baseados em técnicas de inteligência computacional.

Desenvolvimento

A metodologia utilizada neste trabalho é

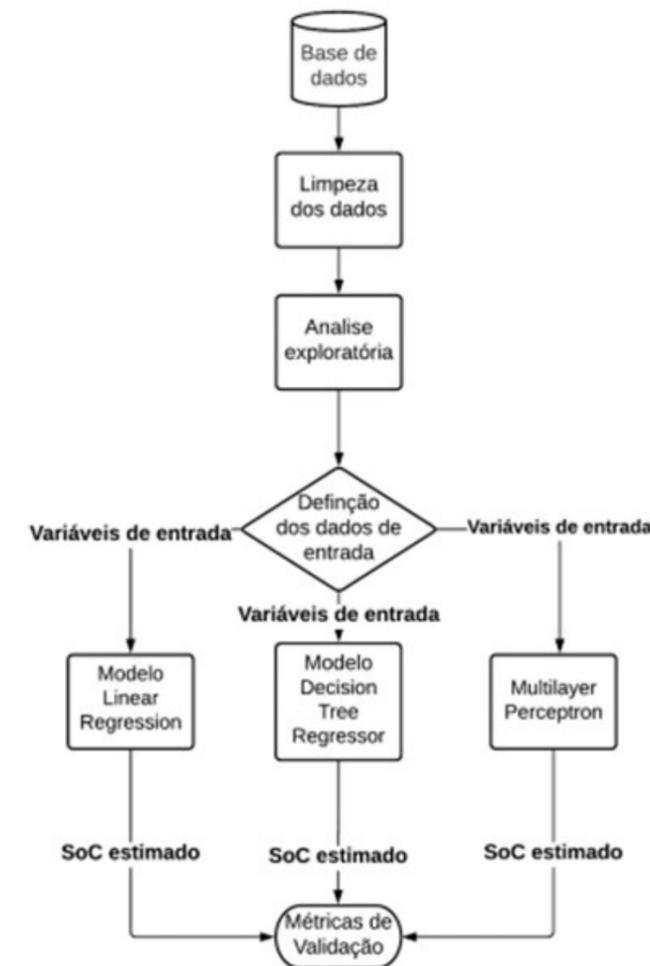


Fig. 1: Fluxograma referente à metodologia utilizada. Fonte: autor.

apresentada na Figura 1. É exposto um fluxograma que descreve a sequência de instruções definida para a rotina computacional desenvolvida neste trabalho.

A primeira etapa consiste na limpeza de dados contidos no banco em análise. Este tratamento é necessário para possibilitar a manipulação adequada das informações. Basicamente, as informações contidas no banco de dados foram cuidadosamente avaliadas e, as grandezas que apresentavam medições sem valores válidos, ou que manifestavam escalas discrepantes entre valores mínimos e máximos, foram excluídas das análises.

Dessa forma, a próxima etapa descrita

como análise exploratória, se torna mais eficiente para identificar correlações, padrões e hipóteses. Nas etapas finais, os modelos de Regressão Linear, Árvore de decisão e RNA foram desenvolvidos e submetidos às métricas de avaliação, de modo a analisar a performance dos resultados. O treinamento dos modelos foi realizado considerando-se 70% dos dados, sendo que os 30% restantes foram utilizados para os testes de validação.

Base de dados

Os dados utilizados neste trabalho foram disponibilizados pela IEEE DataPort™, que se trata de um repositório de dados online desenvolvido pelo IEEE (Institute of Electrical and Electronics Engineers), permitindo aos usuários armazenar, pesquisar, acessar e gerenciar dados diversos. Especificamente, foi utilizado, como objeto de estudo, o conjunto de dados BATTERY AND HEATING DATA IN REAL DRIVING CYCLES, dos autores Matthias Steinstraeter, Johannes Buberger, Dimitar Trifonov do Technical University of Munich, Institute of Automotive Technology.

Para obter este banco de dados, foram realizados registros de viagens reais de condução do automóvel BMW I3 (60hA) com variações de terreno, horário, condutor, etc. Em cada viagem, diversas grandezas foram medidas a cada 100ms a partir de sensores instalados no veículo citado, conforme ilustrado pela Figura 2.

Dentre as grandezas medidas, podem se destacar: dados ambientais (temperatura, altitude, etc.); dados do veículo (velocidade,

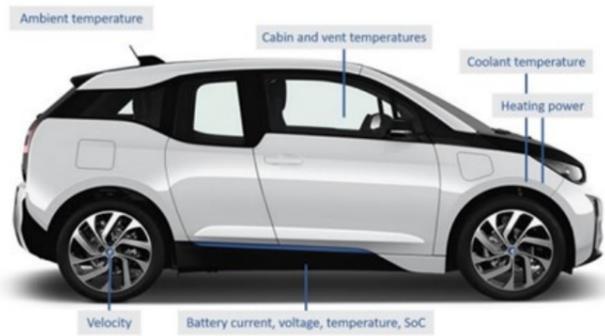


Fig. 2: Veículo BMW I3 (60hA). Fonte: (MATHIAS et al., 2020).

aceleração, etc.); dados da bateria (tensão, corrente, temperatura, SoC); dados do circuito de aquecimento (temperatura interna, energia de aquecimento, etc.). Especificamente, trata-se de 48 variáveis, as quais estão listadas no Quadro 1, que foram amostradas em intervalos de 1ms, para viagens denominadas pelos autores das medições como Trip B. A Trip B se refere à registros de 38 viagens que, como mencionado anteriormente, variam as condições do trajeto e, como objeto de estudo, foram agrupadas informações de 4 destas viagens, totalizando 61458 registros para análise. É importante ressaltar que o banco de dados completo também apresenta outro agrupamento de viagens, denominado como Trip A, que não foi utilizado no presente estudo por se tratar de dados simulados.

Análise exploratória do banco de dados

Após realizada a limpeza dos dados, iniciou-se a análise exploratória, onde é feito um estudo aprofundado do conjunto de dados com o intuito de colher as suas principais características. Assim, permite-se compreender padrões, anomalias, gerar hipóteses e suposições.

Considerando-se o banco de dados em estudo, o primeiro passo desta análise consistiu em gerar gráficos para verificar as varia-

Quadro 1: Lista de variáveis do banco de dados.

Nº	Grandeza	Nº	Grandeza
1	Tempo[s]	25	Núcleo do aquecedor de temperatura do refrigerante [°C]
2	Velocidade [km/h]	26	Temperatura Solicitada do Refrigerante [°C]
3	Elevação [m]	27	Entrada de temperatura do refrigerante [°C]
4	Aceleração [%]	28	Fluxo de volume do refrigerante +500[l/h]
5	Torque do Motor [Nm]	29	Temperatura do trocador de calor [°C]
6	Aceleração Longitudinal [m/s ²]	30	Sensor de temperatura da cabine [°C]
7	Sinal de frenagem regenerativa	31	Entrada do aquecedor de refrigerante de temperatura [°C]
8	Tensão da Bateria [V]	32	Saída do aquecedor do refrigerante de temperatura [°C]
9	Corrente da Bateria [A]	33	Saída do trocador de calor de temperatura[°C]
10	Temperatura da Bateria [°C]	34	Temperatura Descongelamento lateral esquerdo[°C]
11	Max. Temperatura da Bateria [°C]	35	Temperatura Descongelção lateral direita[°C]
12	SoC [%]	36	Temperatura de descongelção central [°C]
13	Display SoC [%]	37	Temperatura descongelção central esquerda [°C]
14	Min. SoC [%]	38	Temperatura de descongelção central direita [°C]
15	Max. SoC [%]	39	Temperatura Footweel Driver [°C]
16	Potência de Aquecimento CAN [kW]	40	Temperatura Footweel Co-Driver [°C]
17	Potência de Aquecimento LIN [W]	41	Temperatura Feetvent Co-Driver [°C]
18	Potência de aquecimento solicitada [W]	42	Temperatura Feetvent Driver [°C]
19	Potência do Ar Condicionado [kW]	43	Temperatura Head Co-Driver [°C]
20	Sinal Aquecedor	44	Temperatura Head Driver [°C]
21	Tensão do aquecedor [V]	45	Temperatura Ventilação direita [°C]
22	Corrente do aquecedor[A]	46	Temperatura Ventilação central direita [°C]
23	Temperatura ambiente [°C]	47	Temperatura Ventilação central esquerda[°C]
24	Sensor de temperatura ambiente [°C]	48	Temperatura Ventilação esquerda [°C]

ções de cada grandeza. Estes gráficos permitem investigar a distribuição dos dados, para identificação daqueles que se diferenciam drasticamente dos outros, os denominados outliers. Estes pontos considerados fora da curva devem ser eliminados das análises para não influenciar negativamente nos resultados da implementação.

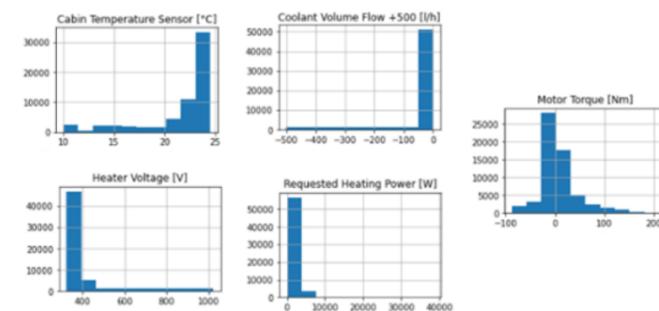


Fig. 3: Gráficos da distribuição de dados das variáveis com outliers. Fonte: autor.

Foi possível observar que algumas grandezas são constantes e não influenciam, portanto, na estimação do estado de carga. Na Figura 3 está ilustrada a distribuição de dados para as grandezas constantes, as quais foram removidas.

Em terceiro plano, foi implementado uma função para relacionar a variável a ser estimada (SoC) com as demais variáveis do banco de dados. Neste contexto, duas análises foram realizadas: correlação linear e não-

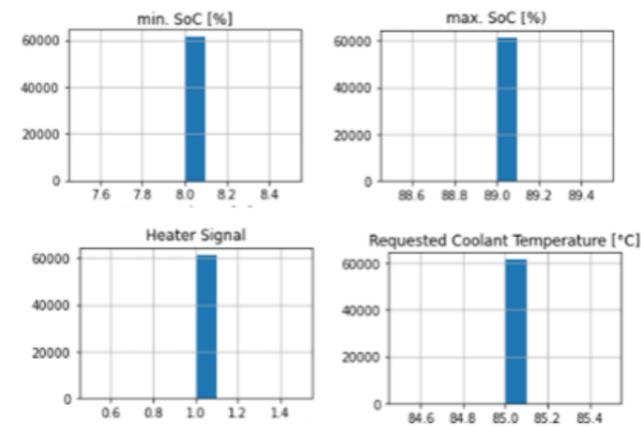


Fig. 4: Gráficos da distribuição de dados constantes.

linear entre as variáveis e o SoC (estado de carga). As Figuras 5 e 6 apresentam estas correlações, em que as variáveis com maiores valores positivos (cores claras), possuem maior correlação positiva com o estado de carga da bateria. Em contrapartida, as variáveis mais escuras, possuem valores negativos, pois, são as variáveis que possuem correlação inversa.

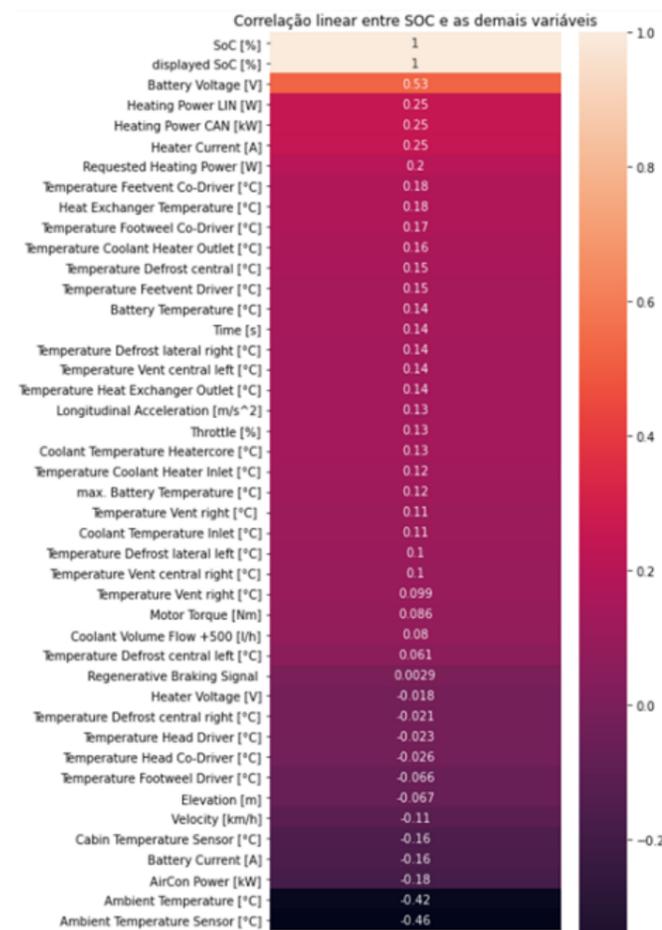


Fig. 5: Correlação linear entre SoC e as demais variáveis. Fonte: autor.

A partir destas análises, foi possível definir as possíveis variáveis de entradas para aplicar os modelos de Regressão Linear, Árvore de Decisão e Redes Neurais Artificiais.

Resultados

Com base na metodologia proposta, são apresentados os resultados obtidos com as implementações dos modelos para a estimação de carga, SoC. Os testes foram realizados através dos dados obtidos do veículo elétrico BMW I3 (60hA) e que foram devidamente avaliados por uma análise exploratória.

A seguir, são apresentados na Tabela 1 e Figura 7 os resultados obtidos através do modelo de Regressão Linear Múltipla, que utiliza como entrada, duas ou mais variáveis preditoras. Desta forma, foram realizados tes-

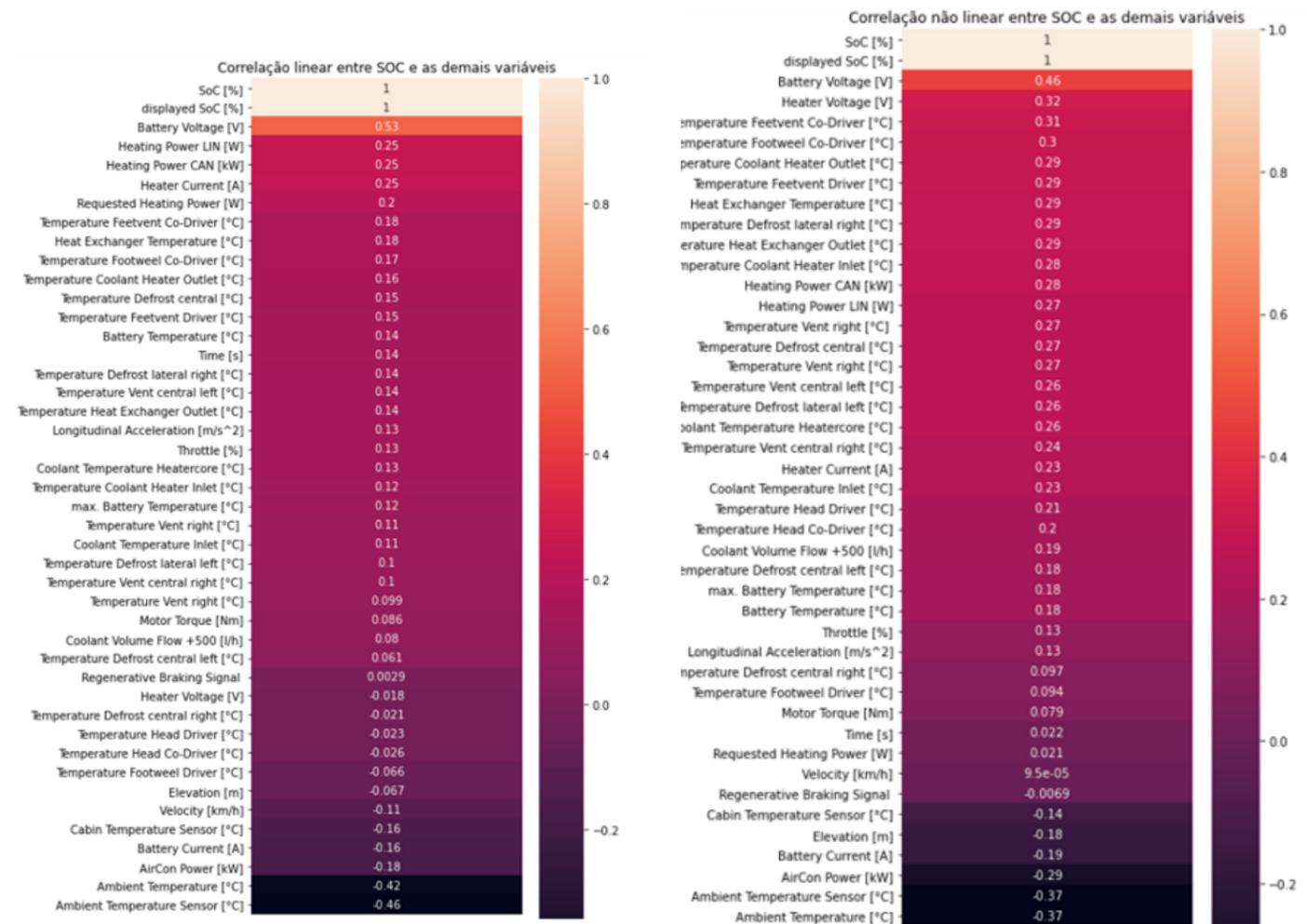


Fig. 6: Correlação não linear entre SoC e as demais variáveis. Fonte: autor.

Modelo	MAE	MSE	RMSE	R ²
Modelo RL-I	4,3581	29,2492	5,4082	0,8441
Modelo RL-II	1,9547	7,2164	2,6863	0,9615
Modelo RL-III	2,0323	6,6645	2,5815	0,9644

Tabela 1: Métricas de avaliação para Modelos de Regressão Linear Múltipla

tes alterando-se a quantidade de variáveis de entradas, para que fosse encontrado um melhor modelo de previsão.

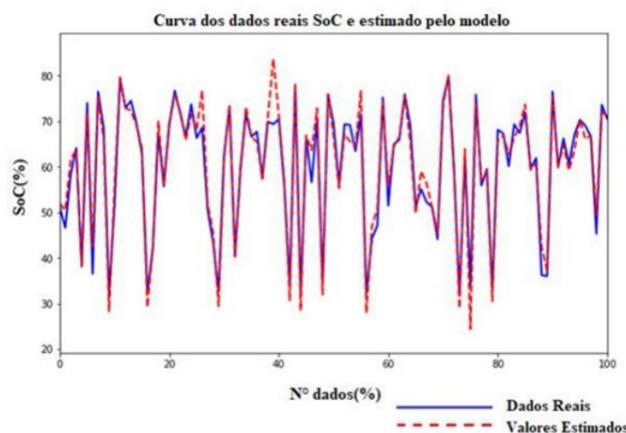


Fig. 7: Curva de dados reais e valores previstos do modelo de Regressão Linear Múltipla (7 variáveis). Fonte: autor.

Em um segundo momento, foi implementado o algoritmo de Árvore de Decisão, que é um modelo mais complexo e que exige um maior processamento computacional. Entretanto, este modelo apresentou resultados mais eficientes. Por apresentar uma complexidade maior, nestes modelos foram feitos testes alterando-se apenas o parâmetro definido como profundidade da árvore. Os resultados obtidos são apresentados na Tabela 2 e Figura 8.

Foi implementado um modelo de Perceptron multicamadas com testes na mesma linha de raciocínio do modelo de Árvore de Decisão, mantendo-se as sete variáveis mais impactantes como dados de entrada e varian-

Modelo	MAE	MSE	RMSE	R ²
Modelo AD-I	4,3584	35,3780	5,9479	0,8114
Modelo AD-II	1,4112	5,5173	2,3488	0,9705
Modelo AD-III	0,1573	0,1044	0,3231	0,9994

Tabela 2: Resultados métricas de avaliação Modelo Árvore de Decisão

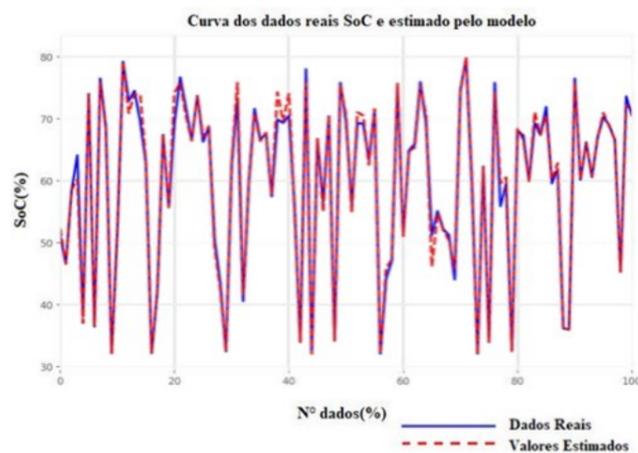


Fig. 8: Curva de dados reais e valores previstos do modelo de Regressão Linear Múltipla (6 variáveis). Fonte: autor.

do-se apenas a arquitetura da RNA. Os resultados são apresentados na Tabela 3 e Figura 9.

Modelo	MAE	MSE	RMSE	R ²
Modelo RNA-I	1,8198	5,2971	2,3015	0,9717
Modelo RNA-II	1,4181	3,2728	1,8091	0,9825
Modelo RNA-III	1,1544	2,2725	1,5075	0,9878
Modelo RNA-IV	0,8106	1,4156	1,1897	0,9924

Tabela 3: Resultados métricas de avaliação Modelo Redes Neurais Artificiais

O modelo de Regressão Linear apresentou bons resultados, com um R² de 0,9615 e um MAE de 1,9547 no modelo com melhor performance. Já o modelo de Árvore de Decisão obteve resultados melhores, sendo que o cálculo das métricas de avaliação de um dos modelos obteve um R² igual a 0,9705 e MAE de 1,412.

Entretanto, durante a realização dos

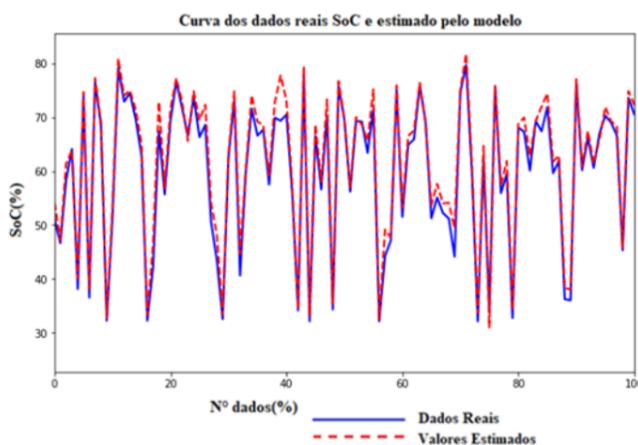


Fig. 9: Curva de dados reais e valores previstos do modelo de Regressão Linear Múltipla (5 variáveis). Fonte: autor.

testes, o algoritmo apresentou sinais de overfitting, sendo necessário avaliar o limite para o parâmetro de profundidade. A implementação de um algoritmo utilizando redes neurais também se mostrou eficaz, apresentando um R² de 0,9717 e um MAE de 1,8198 no modelo mais simplificado. Por ser um algoritmo robusto e complexo, mesmo com poucos neurônios, a rede foi capaz de prever a variável SoC com uma boa performance.

Conclusão

O presente trabalho apresentou a proposta de implementação de modelos, baseados em inteligência computacional, para estimar o estado de carga (SoC) em baterias de íons de lítio em veículos elétricos.

A implementação dos modelos propostos foi precedida por uma análise exploratória, visto que, o tratamento prévio dos dados auxilia a busca eficiente para o problema. Foi possível descartar variáveis constantes, dados faltantes e a remoção de outliers, que prejudicam a performance de qualquer algoritmo de previsão. Outro fator agregado pelo estudo dos dados, foi compreender a multicolinearidade entre variáveis e destacar as mais impactantes, sendo elas a tensão da bateria (battery voltage), corrente da bateria (battery current), temperatura ambiente (ambient temperature), elevação (elevation) e velocidade (velocity).

De acordo com os resultados apresentados, é evidente o fato de que o desenvolvimento de técnicas de inteligência computacional, associado à análise de dados, é de extrema utilidade na engenharia. Isto se justifica pela aplicabilidade no reconhecimento de padrões, previsões ocultas e modelagem de dados altamente voláteis, a qual pode contribuir para encontrar soluções inovadoras para di-

versos tipos de problemas.

No caso específico do presente trabalho, tais técnicas se mostraram eficientes na previsão do estado de carga de baterias de veículos elétricos, podendo caracterizar, por exemplo, a definição de parâmetros como autonomia de um veículo elétrico, assim como alertar sobre o estado de saúde da bateria.

Referências

- AQUINO, Vagner. Carros elétricos serão obrigatórios na Europa em 2035. Disponível em: < <https://jornaldocarro.estadao.com.br/carros/carros-eletricos-serao-obrigatorios-na-europa-em-2035/> >
- BRASIL. Decreto-lei nº 9.073, de 5 de junho de 2017. Promulga o Acordo de Paris sob a Convenção-Quadro das Nações Unidas sobre Mudança do Clima, celebrado em Paris, em 12 de dezembro de 2015, e firmado em Nova Iorque, em 22 de abril de 2016. Disponível em: < http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/decreto/d9073.html >
- MATTHIAS, Steinstraeter, JOHANNES, Buberger, DIMITAR, Trifonov. Battery and Heating Data in Real Driving Cycles, 2020. Disponível em: < <https://iee-dataport.org/openaccess/battery-and-heating-data-real-driving-cycles> >
- RAHIMI-EICHI, H. et al. Battery management system: An overview of its application in the smart grid and electric vehicles. Disponível em: < <https://ieeexplore.ieee.org/document/6532486> >

Autores: Cleber Amaral e Jomi Hübner.

O progresso no campo de pesquisa em inteligência artificial (IA) pode ser constatado em diversas áreas (para além do notório aprendizado de máquina). Os avanços em Programação Orientada a Agentes (POA) e em Sistemas Multiagentes (SMA) são bons exemplos. Essas áreas têm na sua base a noção de agente: entidades autônomas orientadas a objetivos. Suas aplicações são cada vez mais comuns, já que estamos envolvidos por entidades de software que interagem conosco e com certo grau de autonomia nos ajudam em muitas tarefas, tais como a revisão de um texto enquanto estamos digitando, controlando eletrodomésticos e até mesmo dirigindo veículos.

Entre os focos de atenção na área de agentes, temos teorias, arquiteturas, linguagens e algoritmos para ajudar um agente a escolher as ações que atingem seu objetivo. Esse problema, chamado de raciocínio prático (caracterizado como resultando em ações), complementa os estudos mais clássicos da IA, onde se dá mais atenção ao raciocínio teórico (onde a saída é conhecimento). Escolher adequadamente as ações não é uma tarefa fácil. Por exemplo, vamos considerar um veículo autodirigido que tem como objetivo ir do ponto A para o ponto B de uma cidade. O ambiente neste exemplo é dinâmico e não determinístico, ou seja, para alcançar seu objetivo o agente precisa dispor de diversos planos e tomar várias decisões para resolver situações imprevisíveis como a passagem de um pedestre pela faixa, o recente bloqueio de uma estrada, entre outras. Além disso, os agentes são entidades sociais, o que significa que algumas vezes precisam cooperar com outros agentes para alcançar objetivos comuns.

Dentre as várias linguagens de programação de agentes existentes, no DAS/UFSC, temos investido no Jason. Essa linguagem tem sido aplicada, desde 2005, na competição Multi-Agent Programming Contest (MAPC)¹. O MAPC propõe um cenário o qual times podem desenvolver suas soluções e avaliar seu desempenho com relação às estratégias e implementações de outros times. Na edição de 2020/21, estudantes e egressos da PosAutomação se juntaram em um time para participar desta competição [1]. Uma das principais conquistas de nosso time foi a implementação de ferramentas para verificar a qualidade de software do SMA desenvolvido para a competição.

Sobre o MAPC 2020/21

Na edição de 2020/21, em cada rodada, há dois times que jogam um contra o outro. Cada time tem um certo número de agentes que pode controlar. Esses agentes podem se mover ortogonalmente e têm uma visão limitada do ambiente. Para descobrir o que tem a sua volta, os agentes podem se deslocar e, conforme a implementação, mapear os recursos que o ambiente dispõe. Aleatoriamente, o simulador envia aos agentes tarefas que requerem que os agentes conectem a eles certos tipos de blocos e submetam estas estruturas em áreas específicas de entrega. Algumas estruturas só podem ser formadas com a ajuda de outros agentes. Quanto mais complexa e em menor tempo uma estrutura é entregue, maior a pontuação conferida ao time. Vence a competição o time que somar mais pontos ao final da si-

mulação.

Como cada agente tem uma visão limitada do ambiente, um dos desafios é o de encontrar e mapear os recursos existentes para cumprir as tarefas. Outro desafio é o de fazer com que os agentes identifiquem-se no ambiente, uma vez que o simulador não informa qual membro está no campo de visão de outro agente. Apenas com a identificação dos agentes é possível que eles se comuniquem e cooperem para realizar tarefas mais complexas. Determinar se uma tarefa é viável, escolher entre as tarefas com melhor risco/retorno, gerenciar situações imprevisíveis do ambiente como ações do oponente e de eventos aleatórios estão também entre os desafios que este cenário traz.

A biblioteca de testes desenvolvida por nosso time

Um agente em um cenário não determinístico está sujeito a diversas situações que colocam suas funcionalidades em teste. Algumas situações revelam falhas em funcionalidades existentes e outras revelam a necessidade de implementação de novos planos para lidar com situações não previstas. Na época em que ocorreu o MAPC, a linguagem de programação orientada a objetivos utilizada por nossa equipe, Jason, não possuía uma biblioteca para testar os agentes. Sem este recurso, os testes são normalmente feitos por mensagens de debug e recorrentes simulações. A biblioteca de testes desenvolvida otimiza o processo de desenvolvimento, remove dúvidas sobre funcionalidades existentes e dá maior garantia de que as funcionalidades continuarão se com-

portando conforme esperado mesmo após sucessivas implementações futuras. Essa perspectiva vai na direção da metodologia Goal-Oriented Test-Driven Development (GOTDD), proposta por membros de nosso grupo [2]. Nestas abordagens, as funcionalidades dos agentes são desenvolvidas assim como os testes para estas funções. Por exemplo, em um nível mais elementar de testes (unitário), vamos considerar a regra `distance/5` escrita em Jason para calcular a distância Manhattan entre dois pontos, conforme segue.

```
distance(X1,Y1,X2,Y2,D) :- D = math.abs(X2-X1) + math.abs(Y2-Y1).
```

Para testar este tipo de funcionalidade, nossa biblioteca provê um mecanismo que instancia testes automaticamente a cada compilação, indicando eventuais falhas no código que está sendo testado. Este mecanismo é compatível com ferramentas de Integração Contínua (IC), como o GitHub Actions, o que significa que cada versão que é gravada no repositório está sujeita a uma bateria de testes pré-configurados. O GitHub indica no histórico, para cada atualização do projeto, se esta passou em todos os testes ou não. Em caso de falha, os desenvolvedores do projeto podem ser notificados por e-mail. A biblioteca possui planos para assegurar que um determinado resultado esperado está sendo produzido por uma funcionalidade ou não. Por exemplo, a distância Manhattan entre os pontos (0,0) e (3,3) é conhecida, sendo o resultado igual a 6. Portanto, nesta abordagem um plano de testes invoca a regra `distance/5` dando como argumentos as referidas coordenadas e verificando se a regra realmente está retornando o resultado 6. O trecho de código a seguir apresenta alguns testes para assegurar que a regra

1. <http://multiagentcontest.org>

distance/5 está funcionando adequadamente.

```
@[test]
+!testDistance : distance(0,0,3,3,D0) & distance(-30,-20,4,4,D1) &
distance(-0,10,-9,8.9,D2) & distance(0.7,-17,4,-19,D3)
<- !assert_equals(D0,6);
!assert_equals(D1,58);
!assert_equals(D2,10.1);
!assert_equals(D3,5.3).
```

A biblioteca também suporta testes orientados a objetivos. Ela permite verificar se o agente executa um curso de ações cujo resultado é esperado. Por exemplo, para fazer os agentes se deslocarem da posição atual até uma determinada coordenada, nós criamos o plano goto/3, o qual requer a coordenada desejada e retorna a distância efetiva que o agente deve percorrer até este ponto. Este plano utiliza o conhecido algoritmo de busca no espaço A*, o qual é ótimo e completo. O algoritmo contorna obstáculos caso existam retornando sempre o caminho mais curto até o destino. Para testar um plano deste tipo, o agente de teste recebeu como crenças iniciais uma cópia da situação do cenário num dado momento da simulação, conforme ilustração a seguir, onde o símbolo # representa obstáculos, o símbolo O representa o ponto de origem do agente, o símbolo @ representa o ponto de destino e os símbolos <, >, v e ^ representam os movimentos nas 4 direções que devem ser realizados para alcançar o objetivo.

```
0 o>>v ##
1 >v#####
2 ### >>>v# # #####
3 ### >v# ### #
4 >>>v#####
5 >>>>>>v#@<<
6 >v##^
7 @>>^
01234567890123456789012
```

Foi estabelecido que o agente inicia na

coordenada (0,0). No primeiro teste, o agente deve se deslocar até a coordenada (19,7). Neste caso, há obstáculos próximos, porém há ao menos um caminho ótimo que permite que o agente alcance o seu objetivo pela menor distância possível (neste caso, realizando 26 movimentos que é a distância Manhattan entre as dadas coordenadas). O plano de testes para este cenário é ilustrado a seguir.

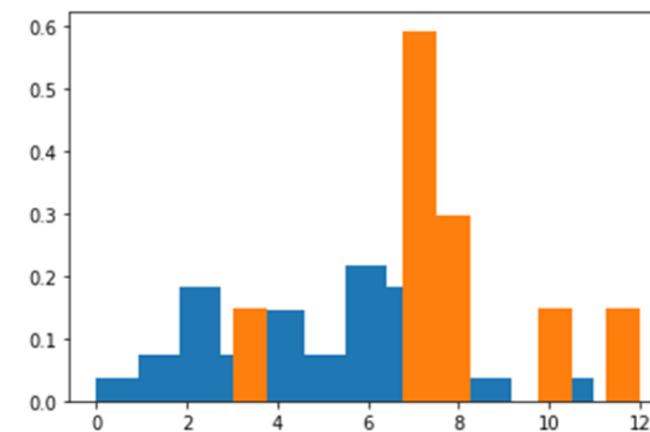
```
@[test]
+!test_goto
<-
+myposition(0,0);
// Teste 1
!goto(19,7,R1);
!assert_equals(26,R1).
// Teste 2
!goto(20,5,R2);
!assert_equals(6,R2).
```

Este tipo de teste assegura que o algoritmo esteja retornando uma solução ótima que é conhecida. Seguindo esta mesma lógica, um segundo teste assegura que o algoritmo contorna corretamente os obstáculos. Neste caso a distância de Manhattan entre os pontos (19,7) e (20,5) é de 3 unidades (movimentos). No entanto, dada existência de obstáculos, o caminho mais curto é de 6 unidades.

Nossa equipe também aplicou testes A/B para verificação de conformidade fim-a-fim, checando se uma dada versão do time é “melhor” que outras. Neste caso os testes não são estáticos, mas sim dinâmicos, realizados em tempo de execução no ambiente final onde os agentes agem. Realmente, se o desempenho dos agentes em um cenário estocástico pode ser medida e a média de ocorrência de certos eventos exibe uma distribuição estatística normal, uma dada versão do SMA pode ser comparada com outras.

Para aplicar esse conceito, implementamos uma função para registrar certos eventos

de interesse. Nós registramos eventos toda vez que em uma simulação os agentes submetiam tarefas com sucesso ou sofriam algum tipo de falha que estava sendo monitorada. As entregas de tarefas deveriam ser maximizadas, enquanto os eventos de falhas deveriam ser minimizados. Cada nova versão que era mesclada com o ramo principal do repositório, manualmente recebia um rótulo. Esta nova versão era colocada para rodar repetidamente, por ao menos trinta vezes, em um computador que utilizamos especificamente para estes testes fim-a-fim. A análise estatística comparava a média de ocorrência dos eventos de interesse entre a nova versão e cada versão anterior a esta. Por exemplo, como ilustrado no gráfico a seguir, ambas as versões 0.17 e 0.12 exibem distribuição normal da média de submissão de tarefas e há evidências estatísticas de que a versão 0.17 (barras de cor laranja) difere da versão 0.12 (barras azuis) e comparando a média de submissões, a versão 0.17 é melhor, pois foram submetidas em média 7,7 tarefas contra 4,8 entregas em média da versão anterior².



Conclusões

Testar o comportamento de entidades autônomas traz novos desafios, pois o curso de ações que um agente executa pode não ser previsível.

2. Mais informações podem ser obtidas em: <https://cleberjamaral.github.io/fastpages/comparison/jupyter/2020/11/06/agents-goal-benchmark-test.html>

Para lidar com isso, nós implementamos uma biblioteca que nos permitiu definir testes unitários, orientados a objetivos e fim-a-fim, os quais ajudam a assegurar que as funcionalidades do agente estão respondendo conforme esperado, que não serão afetadas de maneira desavisada por mudanças futuras do projeto e que mudanças futuras no projeto de fato representam melhorias. Estas implementações ajudam a aproximar uma linguagem de programação de agentes cognitivos como o Jason aos padrões de qualidade de linguagens de programação utilizadas pela indústria de software.

Nossa equipe

Cleber Amaral (UFSC-DAS), Vitor Furio (UFSC-Blumenau), Robson Zagre Junior (UFSC-INE), Timotheus Kampik (Umeå), Maiquel de Brito (UFSC-Blumenau), Maicon R Zатели (UFSC-INE), Tiago L Schmitz (UDESC), Jomi F Hübner (UFSC-DAS), Mauri Ferrandin (UFSC-Blumenau).

Referências

- [1] Cleber J Amaral, Vitor Luis Babireski Furio, Robson Zagre Junior, Timotheus Kampik, Maiquel de Brito, Maicon R Zатели, Tiago L Schmitz, Jomi F Hübner, Mauri Ferrandin (2021), “JaCaMo Builders: Team Description for the Multi-agent Programming Contest 2020/21”, DOI: https://doi.org/10.1007/978-3-030-88549-6_6
- [2] Timotheus Kampik, Cleber Jorge Amaral, and Jomi Fred Hubner (2021), “Developer operations and “ engineering multi-agent systems”, DOI: [10.1007/978-3-030-97457-2_10](https://doi.org/10.1007/978-3-030-97457-2_10)

Aprendizado por imitação generativo adversarial para navegação autônoma em ambientes urbanos

Autores: Eric Aislan Antonelo e Gustavo Claudio Karl Couto

Introdução

Aprendizado por imitação é uma abordagem pela qual um modelo caixa-preta (rede neural) é treinado para imitar um especialista usando um conjunto fixo de amostras de pares observação-ação (ou trajetórias) obtidas daquele especialista. Isso acontece sem a possibilidade de consultar o especialista durante o treinamento. A clonagem comportamental (CC) é um tipo de aprendizagem por imitação baseada em um processo de treinamento supervisionado de um modelo usando um grande conjunto de dados. A CC tem sido utilizada para a obtenção de políticas de condução autônoma para veículos desde 1989, quando a rede neural ALVINN havia sido treinada para dirigir um carro autônomo que dependia de imagens de uma câmera como entrada. As amostras de treinamento são geradas por um conjunto de motoristas humanos.

No entanto, o método de CC gera agentes enviesados, que tenderão a possuir certos comportamentos determinados pelo tipo de motorista empregado na geração das amostras. A CC é passível de complexidade da amostra, pois requer muitos dados de treinamento (amostras de observação-ação) gerados por especialistas (por exemplo, motoristas humanos) para funcionar bem na prática. Mesmo com uma grande quantidade de dados, a CC sempre sofrerá com erros em cascata (erro que é retroalimentado) já que seus modelos são treinados apenas em um subconjunto das amostras necessárias (pares de observação-ação) para uma condução segura e robusta: assim que o veículo encontra uma nova estrada e começa a deslocar-se ligeiramente para o

lado esquerdo ou direito da pista, ele irá realimentar seu erro através de novas observações alimentadas para a rede neural, que por sua vez irá deslocar ainda mais o veículo até que mais nenhuma ação válida possa ser tomada.

Por outro lado, as políticas obtidas por Aprendizado por Reforço (AR) resolvem o problema do erro em cascata, uma vez que aprendem com informações de trajetórias inteiras e não apenas amostras isoladas de ação-observação como em CC, mas exigem a definição de uma função de recompensa para encontrar a política ótima. No AR, o treinamento é um método evolutivo onde um agente aprende por tentativa e erro, ou seja, interagindo com o ambiente e recebendo um sinal de recompensa indicando a qualidade da solução encontrada.

No contexto da aprendizagem por imitação, o AR pode ser usado para aprender a imitar trajetórias geradas por um motorista humano em um processo chamado aprendizagem por reforço inversa (ARI).

A ARI pode ser usada para encontrar políticas de condução autônoma: primeiro encontrando uma função de custo sob a qual o especialista, ou seja, o conjunto de trajetórias de treinamento, é excepcionalmente ótimo; e, em seguida, usando algoritmos de AR que otimizam a função de custo aprendida no passo anterior.

A ARI costuma ser dispendiosa, já que requer AR em um loop interno e, portanto, apresenta dificuldades em funcionar para ambientes grandes. Trabalhos recentes em ARI procuram lidar com essas questões. Ainda assim, aprender uma função de custo na ARI

torna o problema computacionalmente mais caro do que apenas aprender uma política diretamente das trajetórias de treinamento. Uma alternativa recente e poderosa para este problema é dada por Generative Adversarial Imitation Learning (GAIL) que possibilita a aprendizagem de políticas sem a necessidade da etapa de aprendizagem de uma função de custo como em ARI.

Aprendizado por Imitação Generativo Adversarial

Uma nova abordagem chamada Generative Adversarial Imitation Learning (GAIL) pode construir políticas de agentes a partir de um conjunto de trajetórias como em CC, mas que são mais robustas do que as obtidas por CC, por permitir que o agente aprenda a partir da interação com o ambiente como em aprendizagem por reforço. Isso faz com que o agente experimente trajetórias inexistentes no conjunto de trajetórias original, o que lhe permite aprender políticas de controle robustas para situações não previstas nos dados iniciais.

O GAIL permite:

- obter políticas de agente robustas como em aprendizagem por reforço sem a necessidade da definição de uma função de recompensa;
- ser usado com redes neurais profundas para processar entradas de alta dimensionalidade (imagens capturadas por câmera);
- usar um conjunto de treinamento muito menor do que em CC, ou seja, é eficiente em termos do número de amostras (*sample efficient*).

Basicamente, existem dois componentes do GAIL que são treinados iterativamente em um jogo min-max: um classificador discriminativo D é treinado para distinguir entre amostras geradas pela política de aprendizado π e amostras geradas pela política π_E do especialista ou motorista humano (representada pelo conjunto de treinamento rotulado); e a política de aprendizagem π é otimizada para imitar a política π_E do especialista. Matematicamente, GAIL encontra um ponto de sela (π, D) da expressão:

$$\text{onde } D: S \times A \rightarrow (0,1), S \text{ é o espaço de estados,}$$
$$\mathbb{E}_{\pi}[\log(D(s, a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s, a))] - \lambda H(\pi)$$

A é o espaço de ações; π_E é a política do especialista; $H(\pi)$ é um regularizador de política controlado por $\lambda \geq 0$. Tanto a política π , quanto o discriminador D podem ser representados por redes neurais profundas.

Aprendizado por imitação generativo adversarial para navegação autônoma

No entanto, o treinamento em GAIL pode ser instável, gerando dificuldades para alcançar um resultado satisfatório dependendo da tarefa. Embora seja eficiente em termos do número necessário de trajetórias de especialistas, não é tão eficiente no número de interações do ambiente necessárias para a convergência.

Por outro lado, CC converge em algumas épocas, mas assume que seu conjunto de dados é composto por amostras independentes e identicamente distribuídas. Uma abordagem recente na literatura combina as funções de custo de CC e de GAIL em uma função de



Fig. 1: A visão de cima para baixo da simulação com o carro no centro e fazendo uma curva durante um percurso. Cada imagem mostra uma das quatro curvas possíveis, da esquerda para a direita: curvas à esquerda, à esquerda, à direita e à direita.

custo integrada para um aprendizado por imitação estável e eficiente. Essa abordagem foi avaliada em tarefas de controle de baixa dimensionalidade e também na tarefa de alta dimensionalidade do CarRacing do OpenAI Gym.

Em nosso trabalho, propomos uma arquitetura baseada em GAIL de ponta a ponta para navegação autônoma urbana, que é avaliada em trajetórias fixas no simulador realista de condução autônoma CARLA (Figura 1).

O agente recebe uma entrada de alta dimensionalidade oriunda de três câmeras frontais, bem como outras entradas contínuas, como velocidade e o próximo ponto de uma trajetória GPS esparsa no quadro de referência local do veículo (Figuras 2 e 3).

A arquitetura do agente, mostrada na Figura 3, possui duas redes neurais, uma para o ator-crítico e outra para o discriminador. Este último possui uma entrada adicional para a ação, na cor laranja, e uma saída sigmoideal $D(s,a)$ indicando a classificação do par (s,a) . A

camada de saída da rede ator-crítico consiste na direção e aceleração do veículo como ações para o ator (política) e o valor do estado atual $V(s)$ para o crítico. A arquitetura comum, embora não compartilhada (em azul) é composta por um bloco convolucional que processa as imagens das três câmeras frontais, cujas características de saída são concatenadas com outras nove entradas contínuas para a velocidade, o próximo ponto alvo na trajetória GPS esparsa e um comando de direção de alto nível.

Nossos experimentos mostraram que, embora a arquitetura GAIL possa aprender a imitar bem o especialista, o GAIL combinado com CC tem uma convergência muito mais rápida para a trajetória de navegação desejada.

Resultados de Navegação Autônoma no simulador CARLA

O experimento consiste em navegar de forma



Fig. 2: Imagens das três câmeras frontais localizadas na parte esquerda, central e direita do veículo, respectivamente. Elas foram tiradas no ambiente de simulação CARLA. Cada câmera produz uma imagem RGB com 144 pixels de altura e 256 pixels de largura.

Actor Critic / Discriminator Network

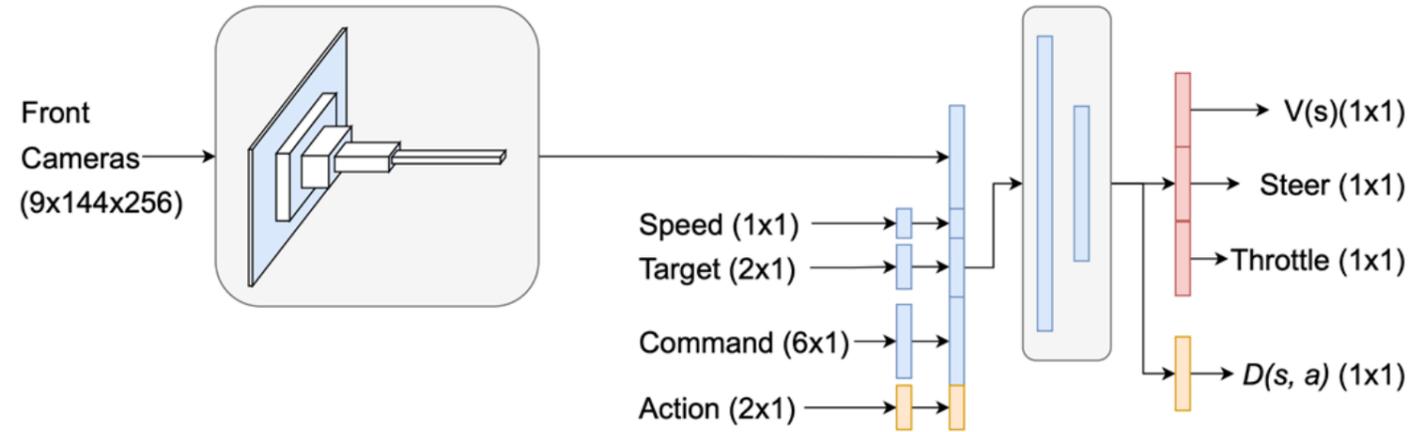


Fig. 3: Arquitetura do agente: rede ator-crítico e rede do discriminador.

autônoma em uma rota de 2.500 metros contendo quatro curvas, a qual foi obtida do CARLA Leaderboard. Para avaliar o desempenho dos agentes, a métrica de pontuação é definida como o número de pontos da trajetória GPS alcançados, representando o quanto da trajetória o agente completou sem nenhum erro. Assim, uma recompensa máxima é equivalente ao número total de pontos GPS na trajetória densa.

O desempenho de aprendizado de GAIL e GAIL combinado com CC (BC_GAIL) pode ser visto na Figura 4. Em média, o agente BC_GAIL conseguiu completar a rota sem nenhum erro muito antes do agente GAIL, mostrando também um avanço rápido e precoce da política. Isso é possível devido à forte influência do termo de CC na função de custo na parte inicial do processo de treinamento. Observe que um agente treinado apenas por CC não é capaz de resolver essa tarefa (alcançou apenas uma recompensa de 173,8 indicado pela linha verde horizontal BC) treinando apenas no mesmo conjunto de dados que o GAIL foi treinado.

A trajetória do agente pode ser visualizada na Figura 5. Ele mostra os erros iniciais na cor vermelha cometidos por um agente GAIL com CC no gráfico superior. À medida

que o treinamento avança, cada vez menos erros são cometidos, como pode ser notado nos gráficos restantes.

Conclusão

Neste trabalho, uma arquitetura de ponta a ponta baseada em GAIL para navegação autônoma em ambientes urbanos foi proposta. Apesar das dificuldades e instabilidades do aprendizado GAIL, tanto GAIL quanto GAIL com CC foram capazes de convergir e gerar agentes capazes de completar toda a trajetória sem erros, sendo que o último conseguiu encontrar rapidamente uma política adequada quando comparado ao primeiro. Ambos superaram a Clonagem Comportamental em desempenho, a qual não foi capaz de gerar um agente sequer capaz de fazer mais de uma volta em média no percurso em questão.

Embora as trajetórias tenham sido previamente fixadas, a arquitetura é geral o suficiente para permitir rotas variáveis, ou seja, um agente que pode mudar de rumo em tempo real em uma rota dinâmica, o que está sendo abordado atualmente por um mestrando. Também planejamos investigar cenários de risco e quais funções de custo podem ser usadas para lidar com essas configurações impor-

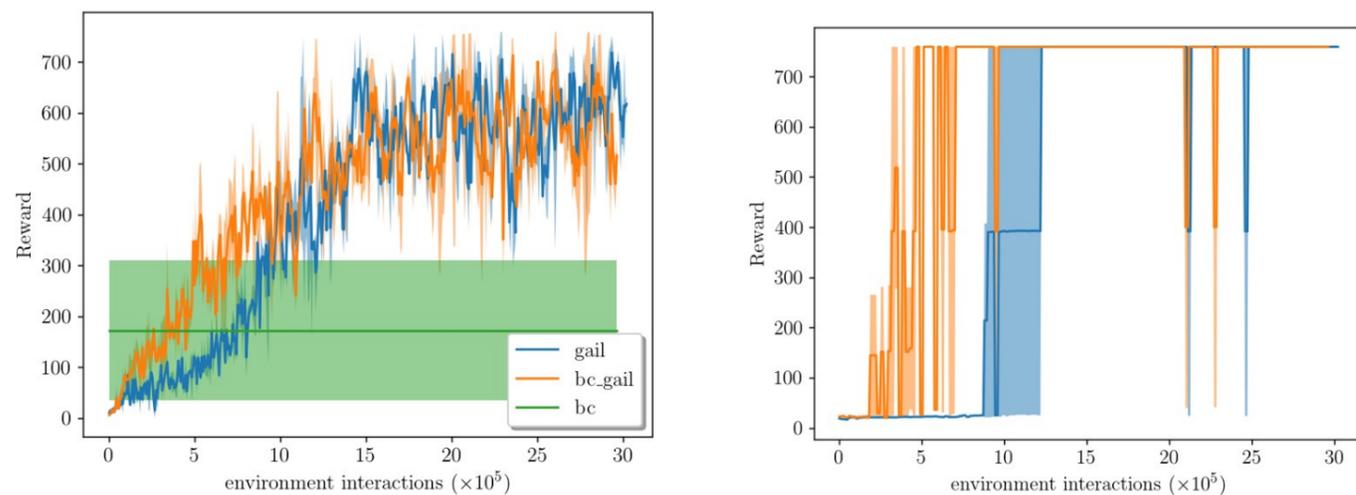


Fig. 4: Recompensas médias versus interações com o ambiente durante o treinamento para navegação de uma rota preestabelecida. Para cada método (GAIL e GAIL com CC/BC), o desempenho médio de duas execuções (ou seja, dois agentes treinados do zero) é mostrado com uma política estocástica (gráfico esquerdo) e uma política determinística (gráfico direito). A área sombreada representa o desvio padrão. A clonagem comportamental (CC) atinge uma recompensa média de 173,6 por dez episódios, enquanto o máximo é de 760, alcançado por GAIL e GAIL combinado com CC.

tantes. É aqui que as abordagens de aprendizado por reforço podem fazer a diferença, pois a CC depende de dados estáticos coletados que podem não ser representativos de todos os cenários possíveis do mundo real.

Para mais informações, consulte o artigo que descreve a proposta mais detalhadamente em: <https://arxiv.org/abs/2110.08586>

O grupo envolvido nesta pesquisa e temas correlatos é composto por:

Prof. Dr. Eric Aislan Antonelo

Eng. Gustavo Claudio Karl Couto

Eng. Irving Giovanni Bronzatti Petrazzini

Fernando Kendy Marciniak Arake

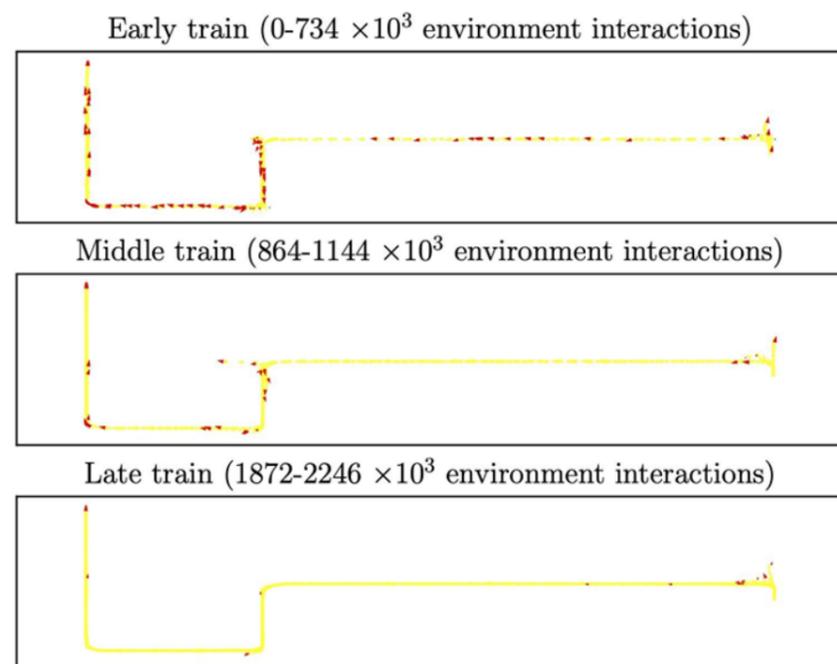


Fig. 5: Trajetória do veículo, em amarelo, para o percurso preestabelecido em diferentes momentos do processo de treinamento. Nas primeiras iterações de treinamento, erros, marcados em vermelho, são comuns. À medida que o treinamento avança, cada vez menos erros acontecem. A trajetória começa no lado direito, em direção ao norte, e termina no lado esquerdo, também em direção ao norte.

Otimização de política com espaço de ações contínuo e limitado: superando o estado da arte no CarRacing

Autores: Eric Aislan Antonelo e Irving Giovanni Bronzatti Petrazzini

Introdução

O Aprendizado por Reforço Profundo (ARP) alcançou resultados sem precedentes em problemas desafiadores com espaço de estado contínuo de alta dimensionalidade, superando o desempenho humano em 29 dos 49 jogos do Atari 2600 em 2015, por exemplo. Em seguida, o AlphaGO, um agente que combina aprendizado por reforço e algoritmos Monte Carlo de árvore de busca balanceada com self play, foi capaz de vencer Lee Sedol, um campeão mundial de 9º dan. Nesse contexto, as Redes Neurais Convolucionais (CNNs) servem como aproximadores de função para a função valor Q, uma vez que podem processar eficientemente entradas do tipo imagem e aprender representações de características úteis desses domínios de espaço de estados contínuos de alta dimensionalidade.

A manipulação de espaços de ação discretos em uma tarefa de aprendizado por reforço profundo geralmente resume-se à definição de uma camada de saída de uma rede neural que tem a mesma dimensão do espaço de ação. Se este espaço for pequeno, uma ação pode ser facilmente extraída da distribuição produzida pela ativação da camada. Caso contrário, encontrar a melhor ação para espaços de ação de alta dimensionalidade ou contínuos constitui um processo de otimização caro por se, que precisa ser executado dentro de outro loop, o ciclo agente-ambiente.

Muitos problemas interessantes do

mundo real, como o controle de manipuladores robóticos e carros autônomos, exigem um espaço de ação contínuo. Ao invés de modelar a função valor Q, o controle contínuo desprovido de modelo é obtido através da otimização direta de uma política que mapeia estados para distribuições de probabilidade definidas sobre espaços de ação contínuos. Essa família de métodos baseados em gradiente de política tem se desenvolvido de forma expressiva, viabilizando a sua aplicação em problemas com espaço de estados contínuos de alta dimensionalidade (como imagens) e espaços de ação também contínuos.

Para modelar uma política estocástica, esses métodos escolhem a distribuição gaussiana, cujos parâmetros média e desvio padrão são preditos pelas saídas de uma rede neural profunda. No entanto, muitas aplicações do mundo real têm espaços de ação limitados a um certo intervalo, geralmente devido a restrições físicas, por exemplo, pelas articulações de um robô humanóide ou manipulador robótico e pelo acelerador e direção de um veículo. Assim, nestes casos, esta política gaussiana, que tem suporte infinito, introduz um viés de estimação, pois ela pode retornar uma probabilidade diferente de zero para ações fora do espaço de ação válido.

Controle Contínuo com a Distribuição

Beta

Em 2017, Chou propôs modelar a política es-

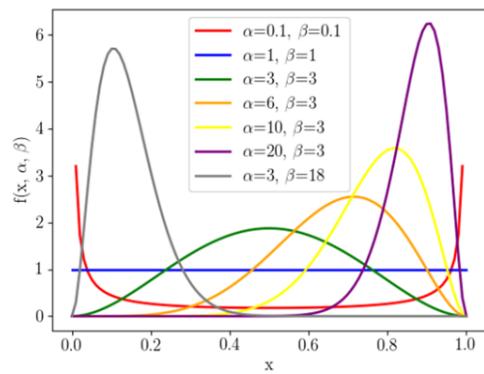


Fig. 1: Função densidade da Distribuição Beta para diferentes valores de α e β . Veja que o suporte desta distribuição é finito, diferentemente da Distribuição Gaussiana.

ocástica com a distribuição Beta (Figura 1), com os parâmetros α e β , de forma que a política resultante tenha um espaço de ação adequadamente delimitado, que não apresente viés como a distribuição gaussiana. Ao invés da média e da variância, agora as saídas da rede neural representam os parâmetros α e β da distribuição Beta. Esta distribuição pode ser usada com qualquer método on- ou off-policy, como Otimização de Política com Região de Confiança (TRPO) de Schulman e Replay de Experiência Ator-Crítico (ACER) de Wang.

Até o momento presente, a distribuição Beta foi avaliada apenas para TRPO e ACER em uma variedade de problemas. A Otimização de Política Proximal (PPO), que evoluiu da TRPO, mas tem uma implementação muito mais simples e um desempenho semelhante ao ACER, ainda carece de experimentação com a distribuição Beta. Este é o primeiro trabalho a relatar experimentos em PPO com a distribuição Beta em aplicações de aprendizado por reforço com espaços de observação de alta dimensionalidade. Além disso, nossa investigação se concentra em duas aplicações de controle contínuo da OpenAI Gym, o Lunar Lander e o Car Racing, ambos não considerados ainda na literatura.

Os benefícios da abordagem são melhor estabilidade e convergência mais rápida do

processo de treinamento. Além disso, como o viés de estimativa está ausente, a política beta aprendida final é superior à política gaussiana final. Também relatamos resultados melhores do que o estado da arte no problema do *Car Racing*.

Aplicação no Controle do *Car Racing*

O ambiente *CarRacing-v0* simula um ambiente de direção autônoma em 2D. Se o agente realizar a condução perfeitamente, por exemplo, em 732 quadros (iterações), a recompensa será de 926,8 pontos, representando o valor máximo para qualquer agente nesse ambiente nessas condições. A tarefa é considerada resolvida se o agente conseguir uma recompensa média de pelo menos 900 pontos em 100 episódios consecutivos.

O espaço de observação consiste em imagens de cima para baixo (Figura 2) de 96x96 pixels e três canais de cores (RGB). Os quatro quadros de imagem mais recentes foram empilhados e fornecidos como entrada para a rede do agente após redimensioná-los e pré-processá-los em escala de cinza (totalizando 84x84x4 dimensões de entrada). O espaço de ação tem três dimensões: uma codifica o ângulo de direção e é delimitada no intervalo $[-1, +1]$. As outras duas dimensões codificam acelerador e freio, ambos limitados a $[0, 1]$.

Para nossa implementação, acelerador e freio foram mesclados em uma única dimensão para que o agente não acelere e freie simultaneamente. Acreditamos que esta seja uma estrutura mais representativa dos sistemas do mundo real: entradas de controle separadas (acelerador/freio), mas mecanismo de ativação único (pé direito). Na prática, um

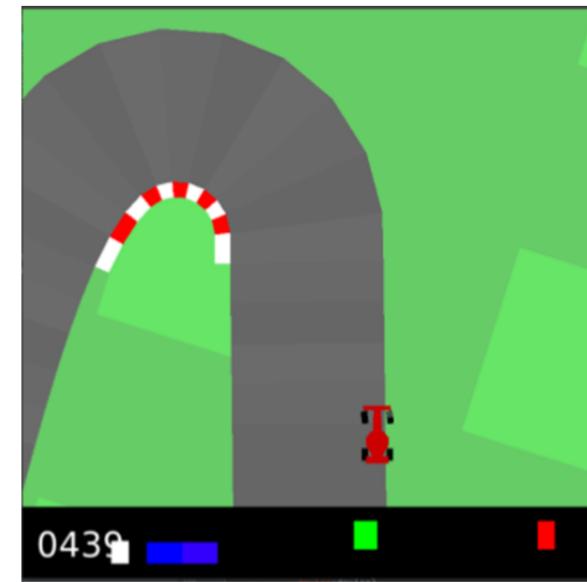


Fig. 2: Ambiente do *CarRacing-v0*. O agente recebe esta imagem de cima como entrada e deve controlar os três atuadores do carro: direção, aceleração e frenagem.

neurônio de saída é responsável por ambas as ações, fazendo com que a saída do agente seja um vetor bidimensional. Com esta abordagem, conseguimos fazer com que o agente aprendesse de forma eficaz, principalmente porque não entra em estado de deadlock resultante da aceleração e frenagem ao mesmo tempo. Se não seguissemos esta abordagem,

aprender a controlar o veículo se tornaria inviável. Até o presente momento, não conseguimos encontrar outros trabalhos na literatura usando a abordagem proposta acima. Além disso, observe que não alteramos o sinal de recompensa original como alguns outros trabalhos geralmente fazem.

O número de interações agente-ambiente foi fixado em 5 milhões de passos de tempo durante o treinamento. Em seguida, uma avaliação do desempenho do agente é realizada pelo cálculo da recompensa média recebida nos últimos 100 episódios consecutivos. A tarefa é resolvida se a recompensa média for pelo menos 900 pontos. Neste problema, observamos que agentes treinados com distribuições Beta e Gaussianas apresentam uma taxa de convergência semelhante durante o treinamento. Na Figura 3, mostramos a recompensa média em uma janela móvel de 10 episódios ao longo do processo de treinamento. Cada passo da otimização de política considera 500 iterações do ambiente em 8 am-

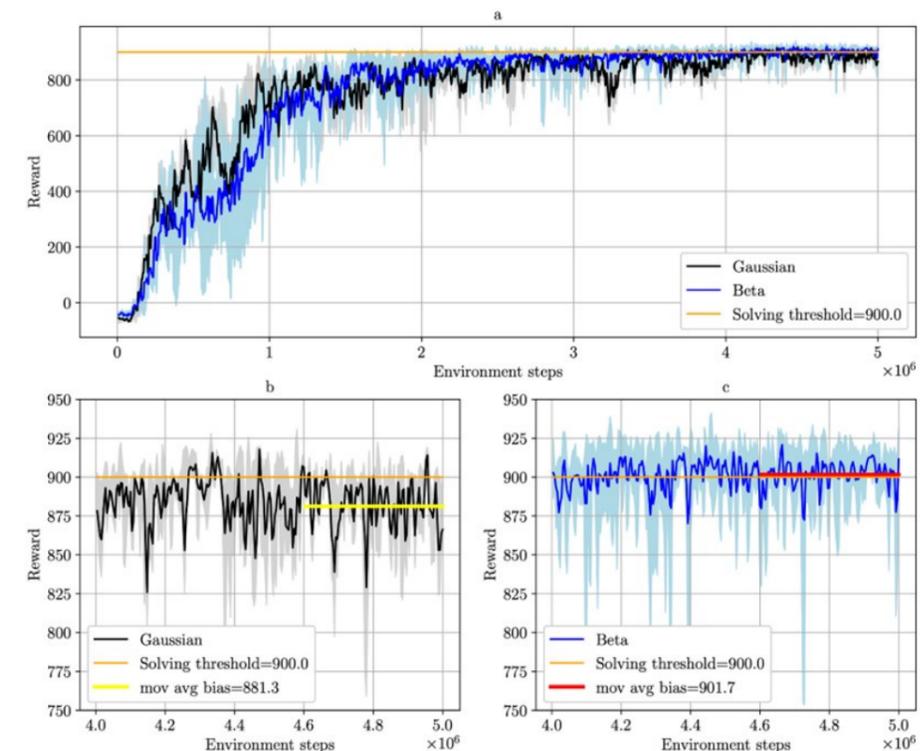


Fig. 3: Recompensas médias para 5 agentes com sementes diferentes para o ambiente *CarRacing*. O desempenho final é mostrado nos gráficos inferiores em uma escala maior, para agentes com política Gaussiana (canto inferior esquerdo) e política Beta (canto inferior direito).

bientes simulados paralelamente.

Usando a medida de desempenho para os últimos 100 episódios consecutivos durante o treinamento, na Figura 4, mostramos que a política estocástica apresentou desempenho médio melhor que a política determinística para ambas as distribuições. Para a distribuição gaussiana, observamos que os cinco agentes com política determinística apresentam uma pontuação média de 370,4 pontos, muito inferior aos 900 pontos necessários para resolver a tarefa. No modo estocástico, a política apresenta um desempenho aprimorado com pontuação média de 890,5 pontos, embora em 38% dos 100 episódios os agentes não tenham conseguido ultrapassar o patamar de 900 pontos.

Para a distribuição Beta, o desempenho dos agentes de política determinística melhora em relação à política gaussiana em 320%, com pontuação média de 816,1 pontos. Esses agentes ultrapassam o patamar de 900 pontos em 26% dos episódios de avaliação. No modo estocástico, todos os agentes conseguiram pontuar acima do limiar de 900 pontos em pelo menos 60% dos 100 episódios jogados por cada agente.

Todos os cinco agentes com a política Beta conseguiram resolver o jogo com sucesso, pois cada um deles atingiu um desempenho superior a 900 pontos. Este não foi o caso da política estocástica gaussiana, onde cada agente obteve menos do que o limiar de 900 pontos. O agente com melhor desempenho, B2, alcançou consistentemente pontuações acima dos outros cinco agentes, e é o agente escolhido para comparar nossa abordagem com outros trabalhos da literatura.

A Figura 5 mostra as políticas Gaussiana e Beta resultantes, após o treinamento,

quando o carro estava prestes a virar à esquerda, como pode ser visto na imagem fornecida à rede neural. A amostragem de ações para essa imagem fixa mostra que a distribuição Gaussiana, com seu suporte infinito, está fora do espaço de ação válido, tornando-a uma política enviesada. Por outro lado, a distribuição Beta se encaixa bem dentro do espaço de ação limitado.

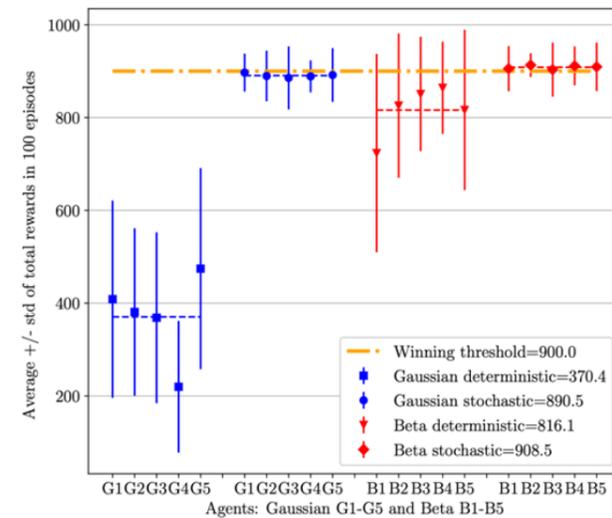


Fig. 4: Comparação da política Gaussiana com a política Beta em termos das recompensas médias obtidas por agentes por 100 episódios consecutivos após o treinamento. Em azul (vermelho), a recompensa média e o desvio padrão para cada um dos 5 agentes usando a política gaussiana (Beta). Tanto políticas determinísticas quanto estocásticas foram empregadas para avaliação. O limiar vencedor dado pela linha amarela horizontal representa o limite mínimo para a conclusão bem-sucedida da tarefa. Os agentes alimentados pela distribuição Beta obtiveram desempenho superior e menos variação. Para esta tarefa, a política estocástica claramente rendeu melhor desempenho do que a determinística.

Superando o estado da arte

O [OpenAI CarRacing-v0 Leaderboard](https://openai.com/research/car-racing-v0-leaderboard) hospeda uma série de pontuações auto-relatadas no problema do CarRacing. Comparamos nossos resultados apenas com aqueles encontrados em artigos revisados por pares (Tabela 1), já que eles fornecem uma base para comparação e discussão. Nessa tabela, podemos observar que nosso agente, com pontuação média de

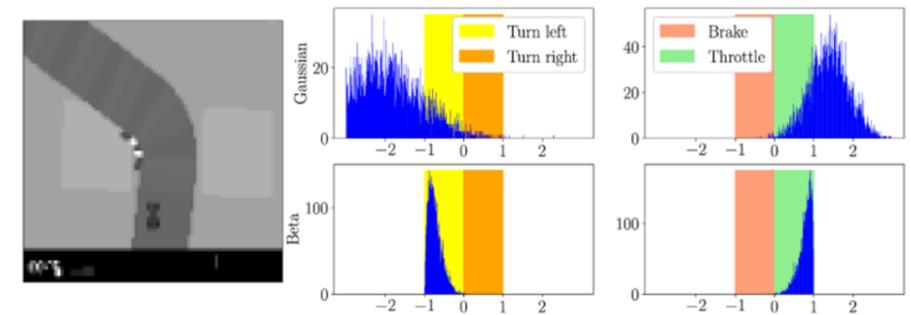


Fig. 5: Ilustração das distribuições de política estocástica Gaussiana e Beta em relação ao espaço de ação do ambiente CarRacing. Para uma observação fixa (imagem pré-processada no gráfico à esquerda), amostramos 5.000 ações usando as políticas Gaussiana e Beta. Para a distribuição gaussiana, uma parte significativa das ações está fora do intervalo válido de valores de direção e freio/acelerador (ambos [-1, 1]), enquanto que para as distribuições Beta, todas as ações amostradas estão dentro dos limites.

913, supera todos os outros agentes encontrados na literatura.

Conclusão

Neste estudo, observamos que os agentes treinados com PPO usando uma distribuição Beta para a política estocástica apresentaram desempenho final significativamente superior aos treinados com uma Distribuição gaussiana. Assim, a distribuição Beta satisfaz melhor os requisitos de aplicações do mundo real com espaços de ação delimitados, eliminando o viés de estimação da política gaussiana.

Nossos resultados também mostram que o controle contínuo com espaço de ação delimitado para o CarRacing, que é um pro-

CARRACING-V0 LEADERBOARD

Method	Average Evaluation Score
PPO with Beta (Ours)	913 +/- 26
World models [17]	906 +/- 21
Adapted DQN [18]	905 +/- 24
Genetic Algorithms [19]	903 +/- 72
PPO with Gaussian (Ours)	897 +/- 41
Weight Agnostic NN [20]	893 +/- 74
PPO [21]	740 +/- 86
Random agent	-32 +/- 6

Tabela 1.

blema desafiador com pistas aleatórias e uma alta dimensionalidade do espaço de observação (baseada em imagens) é facilitado quando

a distribuição Beta é empregada. Finalmente, os resultados sugerem que a distribuição Beta deve ser uma escolha padrão para esse tipo de tarefa.

Planejamos estender esses experimentos para outros tipos de algoritmos de aprendizado por reforço que são mais eficientes em números de amostras, na tentativa de verificar se a distribuição Beta pode ser usada em outras configurações. Além disso, experimentos mais complexos de navegação autônoma em cenários urbanos podem se beneficiar da convergência mais rápida e estável, já que o treinamento de modelos de ponta a ponta não é uma tarefa trivial.

Para mais informações, consulte o artigo que descreve a proposta mais detalhadamente em:

<https://arxiv.org/abs/2111.02202>

O grupo envolvido nesta pesquisa e temas correlatos é composto por:

Prof. Dr. Eric Aislan Antonelo

Eng. Irving Giovanni Bronzatti Petrazzini

Eng. Gustavo Claudio Karl Couto

Fernando Kendy Marciniak Arake

Inteligência artificial e justiça: predição de decisões para incentivar acordos em audiências de conciliação

Autores: Cristina Sabo (PPGD), Thiago Raulino Dal Pont (PosAutomação), Jomi Fred Hübner (DAS/UFSC) e Aires José Rover (DIR/UFSC)

Introdução

O Poder Judiciário brasileiro enfrenta atualmente altos índices de litigiosidade e de congestionamento processual. O último relatório “Justiça em Números”, publicação anual do Conselho Nacional de Justiça (CNJ), aponta números crescentes relativos ao ajuizamento de novas ações, das quais significativa parte ocorre nos Juizados Especiais Cíveis (JECs) (CNJ, 2021). Estes órgãos são responsáveis por julgar causas de menor complexidade e por facilitar o acesso do cidadão à Justiça, através da isenção de custas processuais e dispensa de advogado. Para minimizar essa crise e aprimorar a gestão processual, o CNJ tem elaborado políticas envolvendo duas agendas: (1) métodos de soluções alternativas de conflito, e.g., incorporando a conciliação e a mediação como fase obrigatória nas ações judiciais; (2) uso intensivo de tecnologias da informação (e-Justiça), e.g., criando aberturas e incentivos para promover soluções automatizadas baseadas em Inteligência Artificial (CNJ, 2010; 2020).

Dado esse contexto, reunimos pesquisadores das duas áreas do conhecimento – Automação e Direito – para projetar uma solução que alinhasse as duas grandes metas: o consenso entre as partes litigantes e a modernização da Justiça brasileira. No atual estado da arte, o emprego de técnicas robustas de Mineração de Texto, Processamento de Linguagem Natural e Aprendizado de Máquina, possibilitam a predição de resultados de decisões judiciais a partir da representação vetori-

al do texto jurídico processual (ASHLEY, 2022). Logo, se hoje conseguimos estimar por meio de modelos baseados em aprendizado de máquina o quanto um consumidor lesado irá receber a título de indenização por danos morais, porque não fornecer essa informação às partes envolvidas, a fim de que elas iniciem uma negociação e evitem o prolongamento da ação judicial? Essa indagação motivou a pesquisa desenvolvida que, como qualquer outra, enfrentou (e ainda enfrenta) as seguintes questões:

- **Qualidade dos dados (entrada):** embora o Poder Judiciário seja um grande armazenador de dados, estes encontram-se descentralizados em diferentes sistemas de processo eletrônico com pouca (ou nenhuma) informação quantitativa e qualitativa estruturada. Isso dificulta não só a preparação dos dados, como também a própria coleta.
- **Qualidade das predições (saída):** mesmo que a literatura científica forneça condições para a predição de valores, estes precisam ser precisos para que possam servir de apoio à tomada de decisão (acordo celebrado pelas partes litigantes). Isso demanda a experimentação e combinação das mais variadas técnicas no processo de aprendizado, o que, conseqüentemente, exige tempo e custo computacional.
- **Explicação das predições:** ainda que obtenhamos resultados precisos, não significa que eles serão aceitos pelas partes

litigantes. Por exemplo, nem sempre o valor estimado da indenização atenderá as expectativas do consumidor lesado, o que poderá levá-lo a preferir aguardar pelo valor real, ou seja, pela sentença. Para que o consumidor aceite a predição, uma explicação que a justifique é fundamental.

O ambiente escolhido para a realização desta pesquisa foi o JEC da própria UFSC, dada a proximidade da unidade e por ele já servir de apoio às pesquisas universitárias. Após realizar visitas ao ambiente e em contato com a Juíza de Direito responsável, Vânia Petermann, escolhemos um assunto jurídico bastante recorrente nas audiências de conciliação ali realizadas: ações de indenização por danos morais ajuizadas por consumidores contra companhias aéreas devido a falhas na prestação do serviço.

Técnicas, experimentos e resultados

Coletadas as sentenças específicas sobre o tema (hoje contamos com aproximadamente 1.200 decisões), iniciamos a experimentação considerando cinco tarefas/técnicas de Aprendizado de Máquina:

Clusterização: uma vez que os dados coletados não são estruturados (são arquivos de texto simples), o emprego de aprendizado não supervisionado permite agrupar as decisões que são semelhantes entre si, permitindo ao especialista identificar um padrão. Com isso, é possível extrair informações dos dados

minimizando vieses. Essas informações serão posteriormente estruturadas em (i) atributos/variáveis independentes, que correspondem aos fatores que podem influenciar a decisão do juiz; e (ii) rótulos de classe/variáveis dependentes, que consistem nos resultados práticos da decisão (o veredito do juiz e o valor por ele fixado como indenização por dano moral). Utilizando abordagens de *soft* e *hard clustering* e modelagem de tópicos com LSI (*Latent Semantic Indexing*), conseguimos identificar atributos como extravio temporário de bagagem, assim como o intervalo em dias entre a data do extravio e a devolução da bagagem ao consumidor; ainda, atraso de voo e o intervalo em horas entre o horário da reserva efetuada e da efetiva decolagem (SABO et al., 2021). A Figura 1 demonstra uma das clusterizações experimentadas (Hierárquica), ao passo que a Figura 2 apresenta os principais tópicos identificados em um dos clusters (referente ao atributo ‘extravio de bagagem’).

Associação: uma vez estruturados os dados, outra técnica de aprendizado não supervisionado consiste na geração de regras de implicação lógica, o que possibilita descobrir relacionamentos entre os atributos (fatores da decisão) e os rótulos de classe (resultados práticos da decisão). Empregando o algoritmo clássico de regras de associação Frequent Pattern (FP) Growth, conseguimos obter regras tendo como antecedente uma determinada faixa de intervalo de atraso de voo, e como conseqüente, uma determinada faixa de indenização por dano moral. Esses resultados posteriormente auxiliarão na explicação das predi-

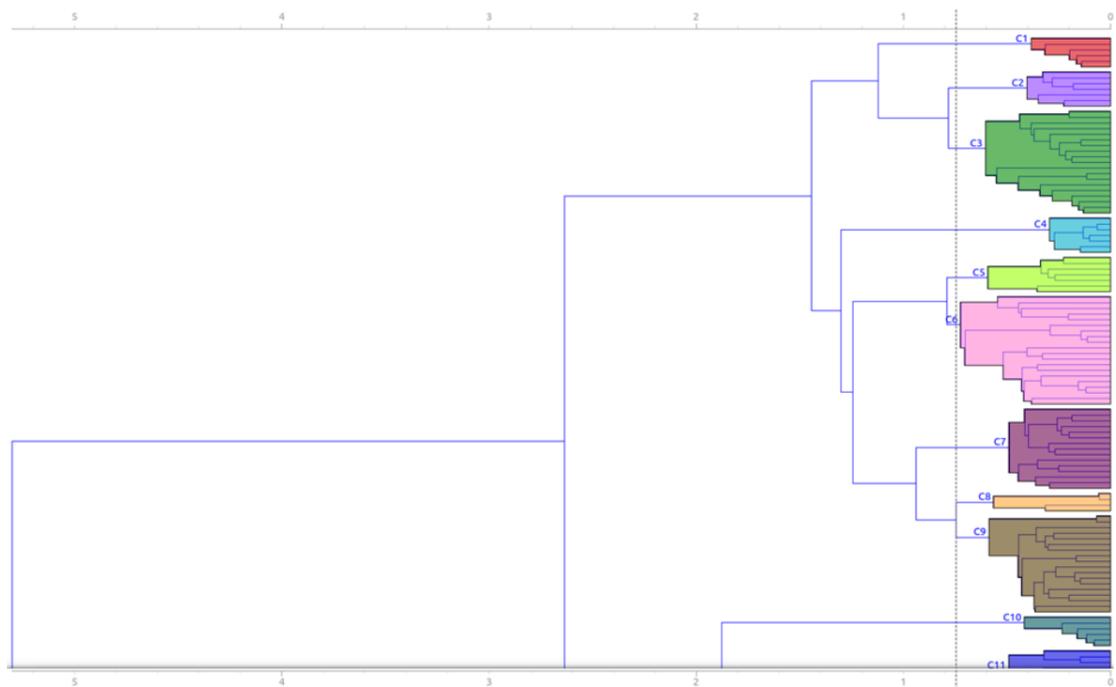


Fig. 1: Dendrograma parcial (Clusterização Hierárquica).

ções, uma vez que permite fornecer às partes quais fatores levam à decisão predita.

Representação: considerando que os documentos jurídicos são muito particulares em termos de vocabulário, aprimorar a representação textual para melhor captação da sua semântica também foi uma importante etapa que nos dedicamos a aprimorar no processo de aprendizado. Nesse sentido, treinamos um modelo robusto de representação aplicando o algoritmo GloVe Word Embeddings e utili-



Fig. 2: Nuvem de palavras referente a um determinado cluster (atributo 'extravio de bagagem').

zando uma quantidade maior de decisões judiciais (aproximadamente 2 milhões) coletadas de instâncias superiores àquelas do JEC/UFSC, quais sejam, do Tribunal de Justiça do Estado de Santa Catarina (TJSC), do Superior Tribunal de Justiça (STJ) e do Supremo Tribunal Federal (STF) (DAL PONT et al., 2020). A Figura 3 demonstra a pipeline referente a esse experimento.

Classificação: cumpridas as etapas anteriores (extração dos rótulos de classe/resultados da sentença e aprimoramento da representação textual), a classificação possibilita predizermos o resultado categórico, que aqui corresponde ao veredito do juiz: (i) totalmente procedente: significa que o consumidor teve todos os seus pedidos acolhidos; (ii) parcialmente procedente: significa que o consumidor teve parte dos seus pedidos acolhidos (quando o consumidor pleiteia, por exemplo, R\$ 10.000,00 de indenização, porém lhe é fixado R\$ 5.000,00); (iii) improcedente: significa que o consumidor teve todos os seus pedidos rejeitados; (iv) extinção sem julgamento do mérito: significa que o consumidor não teve o

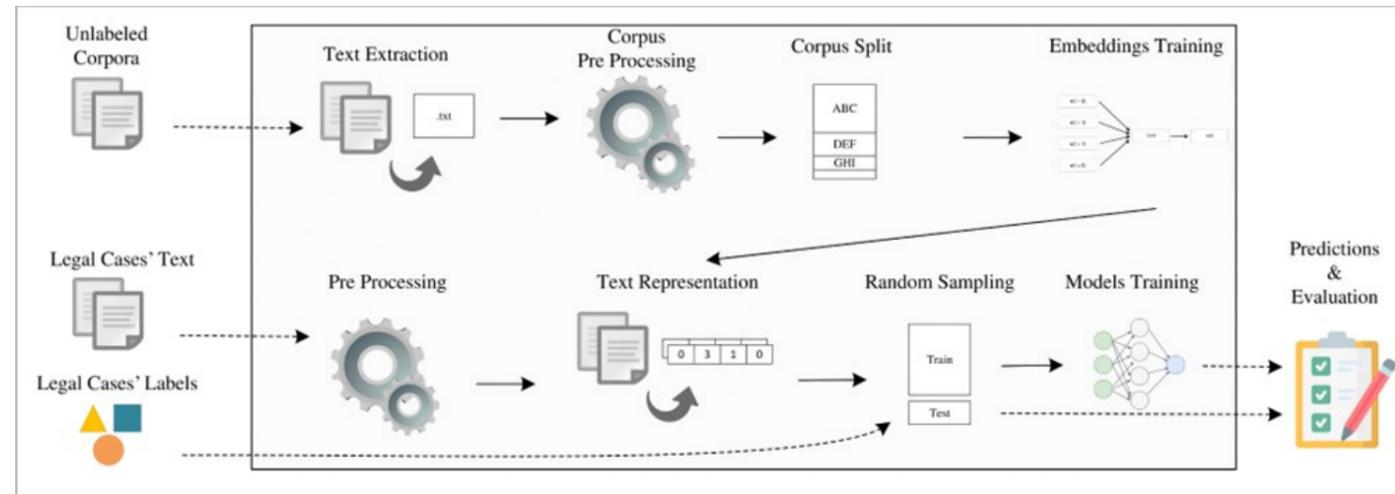


Fig. 3.

seu pedido analisado em razão de alguma incongruência processual (quando o consumidor ajuizou a ação contra a companhia aérea equivocada, por exemplo) (SABO et al., 2019). Dado o problema multiclasse e a partir do modelo de representação treinado especificamente com decisões judiciais, recorreremos ao aprendizado profundo, com o uso de CNN (Convolutional Neural Network) e LSTM (Long Short Term Memory), por meio das quais atingimos uma acurácia total de 95,5% (DAL PONT, 2021). A Figura 4 demonstra a pipeline referente a esse experimento.

Regressão: uma vez que conseguirmos prever se o consumidor será atendido em sua demanda, o próximo passo é prever o valor da indenização. Dado que o Código de Processo Civil não prevê parâmetros de fixação – o juiz é livre para decidir a quantia de

acordo com o seu convencimento da gravidade dos fatos –, a estimativa do valor pecuniário é uma informação de interesse às partes envolvidas, já que o escopo do processo é fazer com que o consumidor lesado seja reparado pela companhia aérea que violou o seu direito. Após experimentar diferentes técnicas de regressão, alcançamos, em um primeiro momento, um erro médio de R\$ 2.006,00, baseado no método Ensemble Voting, que agrega o melhor desempenho de diferentes regressores para obter um valor final. Por fim, após adicionar à entrada os atributos (fatores da decisão) extraídos e ajustar alguns detalhes do pré-processamento do texto, bem como alguns parâmetros de treinamento, conseguimos um erro médio de R\$ 915,00 (mais detalhes em DAL PONT et al., 2022). A Figura 5 demonstra a pipeline referente a esse experimento.

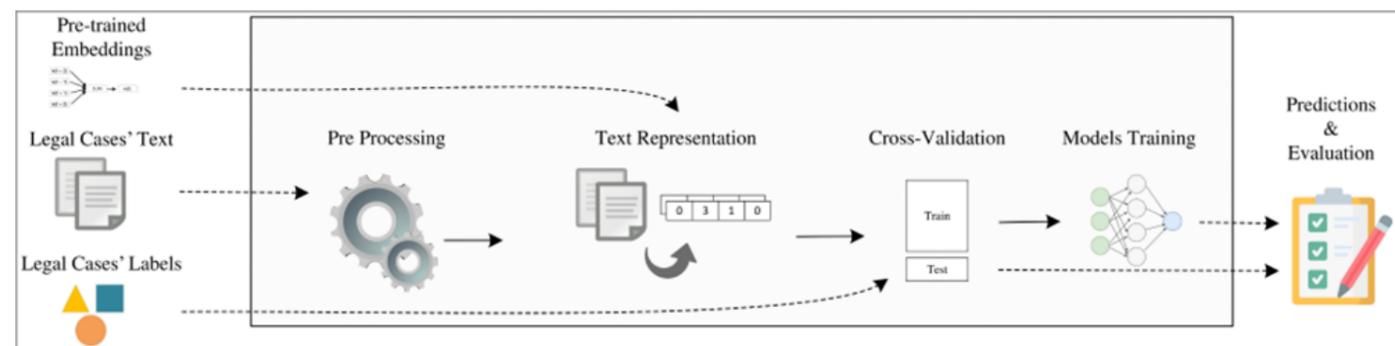


Fig. 4.

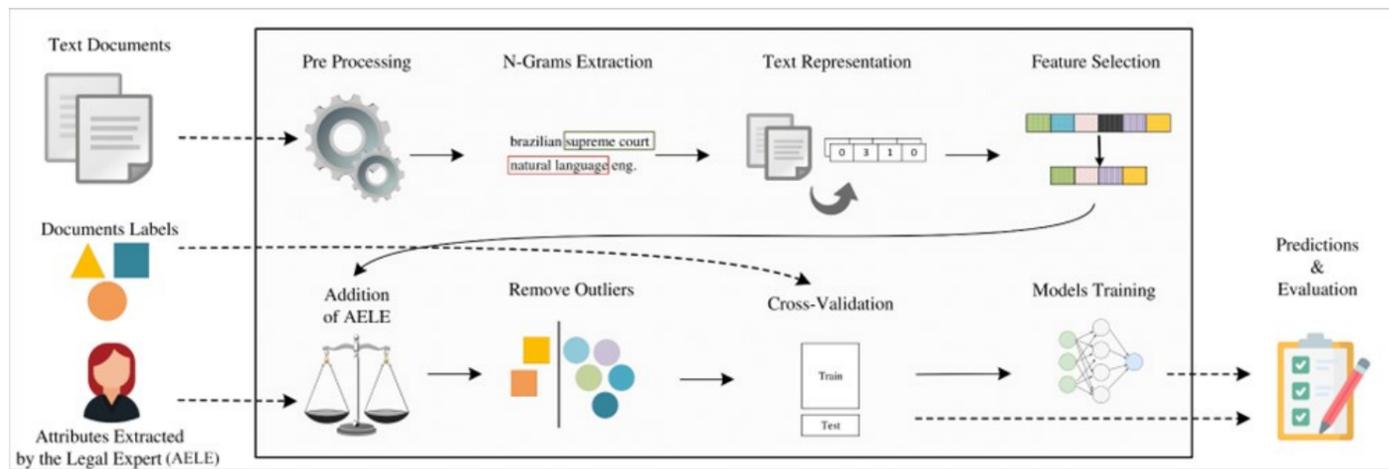


Fig. 5.

Como resultado da aplicação da proposta temos um caso em uma negociação em que a companhia aérea ofereceu ao consumidor lesado uma indenização no valor de R\$ 5.000,00, baseado em um resultado predito em nossa melhor pipeline. Ciente de que a possibilidade de erro gira em torno de R\$ 1.000,00, entendemos que o consumidor lesado tenderá a aceitar a proposta, finalizando o processo com um acordo. Portanto, do ponto de vista jurídico, trata-se de um erro razoável, permitindo o uso das predições em ações judiciais recém-ajuizadas e que são submetidas às audiências de conciliação.

Aplicação no JEC/UFSC

São diversas as causas que impedem a realização de um acordo em uma audiência de conciliação: o não comparecimento das partes litigantes durante o ato, o não oferecimento de propostas iniciais de acordo (ou propostas de valores ínfimos); e ainda, a ausência de orientações por parte do conciliador, que não possui acesso a uma base de dados locais que lhe possibilite, por exemplo, informar às partes alguns fatores que influenciam a decisão e as faixas de indenização para o caso em específico. Neste projeto, nós nos concentramos nesta

última situação.

Para possibilitar uma aplicação da melhor pipeline, a Juíza de Direito nos designou como conciliadores em 14 novos casos submetidos à audiência no JEC/UFSC. Destes, extraímos os mesmos atributos e escolhemos um texto similar em nossa base de dados, além de executarmos várias vezes a pipeline para obter uma média mínima e máxima dos valores resultados. Com isso, apresentamos às partes litigantes uma faixa, e não um valor fixo. Para melhor interação com elas, preparamos um material contendo: (i) propósito do projeto (fornecer informação para auxiliar na negociação baseada em dados de casos anteriores); (ii) vantagens do acordo em relação à sentença (como obter a indenização naquele momento, sem necessitar aguardar o tempo médio de conclusão do processo); (iii) atributos/fatores que influenciam a decisão (como extravio de bagagem, atraso de voo, intervalo do atraso, etc.); (iv) visão geral das indenizações fixadas no JEC/UFSC (quais valores são mais recorrentes, a fim de demonstrar que indenizações altas são exceções); (v) em que circunstâncias essas largas indenizações foram arbitradas (em um desses casos, o atraso do voo fez com que o consumidor não conseguisse comparecer no velório de um parente), a

fim de que as partes pudessem compará-los com o seu caso e refletir sobre o nível de gravidade dele; por fim, (iv) quais as médias previstas para aquele caso em específico, seus respectivos atributos/fatores, bem como a relação entre eles e o valor predito. Também enviamos um questionário anônimo e voluntário às partes e advogados que participaram das audiências, com o intuito de avaliar sua opinião a respeito das informações fornecidas e dessa nova dinâmica de conciliação.

Dos 14 casos, conseguimos realizar acordo em 1 deles. Embora tenha sido um alcance mínimo, entendemos que foi um passo inicial relevante. Neste caso em específico, percebemos que o valor predito alinhou as expectativas de ambas as partes (a companhia aérea aumentou sua oferta e, do outro lado, o consumidor não majorou seu pedido). Uma causa comum que prejudicou a nossa atuação em alguns casos foi a ausência de uma oferta inicial por parte da companhia aérea (há casos em que a empresa contrata um advogado somente para o ato, sem conceder-lhe poderes para transigir). Isto dificultou iniciarmos uma negociação, mesmo com as predições disponibilizadas. Ao final, analisando as respostas obtidas no questionário, verificamos que, de forma geral, as informações e predições apresentadas (tanto sobre os casos anteriores, como para o caso específico) foram bem recebidas pelas partes.

Considerações finais e próximos passos

São várias as vantagens de se realizar uma pesquisa interdisciplinar. Neste projeto, conseguimos avançar no estado da arte em relação ao Aprendizado de Máquina aplicado a

textos jurídicos, avaliando o desempenho de diferentes abordagens e técnicas; no Direito, conseguimos propor uma nova dinâmica para as audiências de conciliação em termos de Justiça. Além disso, a pesquisa realizada é uma forma de aproximar a universidade dos órgãos públicos e, principalmente, do cidadão que busca o Poder Judiciário para solucionar causas menos complexas (mas não menos importantes).

Como introduzido, enfrentamos três questões durante a pesquisa: (i) qualidade dos dados; (ii) qualidade das predições; e (iii) explicação das predições. Conseguimos avançar nos dois primeiros, entretanto, há ainda o que se avançar em termos de XAI (Explainable Artificial Intelligence), por exemplo, utilizando técnicas como SHAP (SHapley Additive exPlanations) para verificar a contribuição de cada atributo/fator na predição, e LIME (Local Interpretable Model-agnostic Explanations) para destacar partes textuais preditivas. Nesse sentido, visualizamos possíveis contribuições nessa linha unindo a abordagem simbólica (extração de atributos/fatores e de regras de associação) à conexãoista.

Ainda, visando dar autonomia ao projeto desenvolvido, a pesquisa poderá evoluir para a construção de um sistema de ODR (Online Dispute Resolution) baseado nas melhores pipelines obtidas, através do qual as partes, conciliadores e juízes, respondendo alguns questionamentos prévios relacionados ao caso em específico, tenham fácil e aberto acesso às predições e demais estatísticas do JEC/UFSC, facilitando e informando a negociação.

Produtos

Essa colaboração resultou na publicação de artigos de revista e de evento (referenciados ao longo do item 3, relativos a cada etapa da pesquisa), bem como na dissertação de mestrado do autor Thiago Raulino Dal Pont, e da tese de doutorado da autora Isabela Cristina Sabo.

Referências

ASHLEY, Kevin D. Prospects for legal analytics: some approaches to extracting more meaning from legal texts. *University of Cincinnati Law Review*, v. 90, n. 4, p. 5, 2022.

CONSELHO NACIONAL DE JUSTIÇA (CNJ). *Justiça em Números 2021: ano-base 2020*. Brasília, DF, 2021.

CONSELHO NACIONAL DE JUSTIÇA (CNJ). Resolução nº 125 de 29 de novembro de 2010. Dispõe sobre a Política Judiciária Nacional de tratamento adequado dos conflitos de interesses no âmbito do Poder Judiciário e dá outras providências. *Diário da Justiça eletrônico*, Brasília, DF, 29 Nov. 2010.

CONSELHO NACIONAL DE JUSTIÇA (CNJ). Resolução nº 332 de 21 de agosto de 2020. Dispõe sobre a ética, a transparência e a governança na produção e no uso de Inteligência Artificial no Poder Judiciário e dá outras providências. *Diário da Justiça eletrônico*, Brasília, DF, 21 Aug. 2020.

DAL PONT, Thiago R.; SABO, Isabela C.; HUBNER, Jomi F.; ROVER, Aires J. Impact of text specificity and size on word embeddings performance: an empirical evaluation in Brazilian legal domain. In: Cerri R.; Prati R.C. (Org.). *Lecture Notes in Computer Science*. Cham: Springer International Publishing,

2020, v. 12319, p. 521-535.

DAL PONT, Thiago R. Representation, classification and regression techniques applied to legal judgments about immaterial damage due to failures in air transport services. 2021. 120p: Dissertação (Mestrado em Engenharia de Automação e Sistemas) - Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2021.

DAL PONT, Thiago R.; SABO, Isabela C.; HUBNER, Jomi F.; ROVER, Aires J. Regression applied to legal judgments to predict compensation for immaterial damage. *PeerJ Computer Science* (In press), 2022.

SABO, Isabela C.; DAL PONT, Thiago R.; ROVER, Aires J.; HÜBNER, Jomi F. Classificação de sentenças de Juizado Especial Cível utilizando aprendizado de máquina. *Revista Democracia Digital e Governo Eletrônico*, v. 1, n. 18, p. 94-106, 2019.

SABO, Isabela C.; DAL PONT, Thiago R.; WILTON, Pablo E. V.; ROVER, Aires J.; HÜBNER, Jomi F. Clustering of Brazilian legal judgments about failures in air transport service: an evaluation of different approaches. *Artificial Intelligence and Law*, Springer, v. 30, p. 21-57, 2022. ■

Utilização de agentes BDI na operação de veículos aéreos não tripulados

Autores: Jomi Fred Hübner, Leandro Buss Becker

Introdução

Nos últimos anos, as aplicações envolvendo os chamados Veículos Aéreos Não Tripulados (VANTs) passaram a extrapolar a área militar e também se tornaram populares na área civil~\cite{fahstrom2012}. Atentos a esse contexto em 2012 foi criado no Departamento de Automação e Sistemas (DAS) da UFSC o chamado Projeto de Veículos Aéreos Não Tripulados - PROVANT, cujo objetivo inicial é explorar técnicas de controle e o projeto de sistemas embarcados nesse domínio aplicação. Estiveram envolvidos os professores Jomi Fred Hübner, Julio Elias Normey Rico e Leandro Buss Becker.

Rapidamente o PROVANT foi se ampliando em termos de escopo e de participação. A Universidade Federal de Minas Gerais (UFMG), na figura do professor Guilherme Raffo (co-fundador do projeto), foi a primeira a se juntar. Também se destaca a participação da Universidade de Sevilla (Espanha), principalmente através do professor Sérgio Esteban. Atualmente estão em fase final de construção dois VANTs do tipo VTOL (Vertical Take Off and Landing), um no Brasil e outro na Espanha.

Em 2013, ano seguinte à criação do PROVANT, iniciou-se a investigação do uso de Sistemas Multi Agentes (SMA) na operação dos VANTs, através de dissertações de mestrado na PosAutomação. Tal interesse se deu pelo fato de que nós, co-autores desse artigo, vislumbramos SMA como sendo um paradigma adequado para modelagem e enge-

nharia de tais sistemas. Um SMA é um conjunto organizado de entidades autônomas orientadas a objetivos chamadas *agentes*, as quais se comunicam e interagem dentro de um *ambiente*. Individualmente, cada agente costuma ter seus próprios objetivos e tarefas para realizar, de forma autônoma, decidindo quais ações fazer. Como um conjunto, os agentes precisam se coordenar e cooperar para atingir o objetivo global do SMA como um todo, como uma *organização*. No contexto do PROVANT, a pesquisa em SMA se concentra em sistemas com as seguintes características (Boissier et al. 2020).

- autonomia: os VANTs devem ser capazes de decidir as melhores ações sem o controle direto de humanos (ou mesmo de outros agentes).
- descentralização: dada a comunicação tipicamente limitada, o sistema não deve ter um ponto central de controle.
- distribuição: cada VANT deve ser capaz de executar com seus próprios recursos de hardware.
- cooperação: o eventual conjunto de todos os VANTs tem que funcionar como uma equipe coerente, dividindo tarefas e colaborando para os objetivos da aplicação.

Em relação ao desenvolvimento dos agentes, a parte de especificação dos processos de decisão é programada em Jason (Bordini et al. 2007) (<http://jason.sf.net>). Jason é uma linguagem de programação orientada a agentes baseada na arquitetura BDI. Nesta arquitetura, o programa é definido em ter-

mos de Crenças, Desejos e Intenções (Beliefs, Desires, Intentions). As crenças representam informações sobre o ambiente (dadas por sensores) e outros agentes (dadas pela comunicação). Os desejos são objetivos potenciais do agente (dados pelo desenvolvedor ou outros agentes, no nosso caso). Intenção são desejos que o agente está realmente perseguindo e tem um plano de ação selecionado para isso.

Inicialmente Fernando Santos (Santos 2015, Santos et al. 2015) mostrou que é possível usar programas em Jason em um sistema embarcado de um VANT (placa *BeagleBone Black*). O resultado deste trabalho mostrou que há vantagens em usar uma linguagem de programação orientada a agentes para desenvolver aplicações de VANTs quando comparado, por exemplo, ao paradigma imperativo (usando C), em uma abordagem mais tradicional. Para tal foi definida uma aplicação simples, a qual define pontos (waypoints) por onde o VANT deve passar. As principais vantagens estavam no tamanho do programa e na facilidade de se imbuir pró-atividade. Também a extensibilidade do programa em Jason foi considerada superior. Entre as desvantagens, foi destacado o aumento do uso de CPU para a abordagem orientada a agentes, quando comparada a uma abordagem tradicional.

Já Marcelo Menegol (Menegol et al. 2018) buscou avaliar o desempenho de SMA quando envolvendo a necessidade de *coordenação*. A solução proposta se baseia no uso do *Moise+* (Hübner et al. 2007), que é um modelo organizacional que considera as relações estruturais, funcionais e deonticas entre os agentes. Com o *Moise+* uma organização pode ser especificada independentemente dos comportamentos individuais dos agentes. A implementação da proposta foi realizada com o uso do framework JaCaMo (Boissier et al.

2020). Para avaliar a mesma, foi utilizado um cenário de aplicação complexo, do tipo busca e resgate (S&R - *search-and-rescue*). A proposta foi testada em simulação e também embarcada em um VANT real, utilizando a mesma placa *BeagleBone Black*. Um vídeo do voo de demonstração está disponível em <https://youtu.be/FcS4QDtrBCI>.

Por sua vez, Gustavo Rezende (Silva 2020, Silva et al. 2021) propõe um modelo de *percepção ativa* integrado com agentes em uma arquitetura BDI. Na percepção ativa, o agente planeja e atua para conseguir percepções necessárias para seus objetivos. Além disso, também foi proposta uma arquitetura para programação de robôs autônomos utilizando o ROS (*Robotic Operating System*). Foi desenvolvido um conjunto de simulações realistas baseadas no software Gazebo para testar a abordagem proposta. Um vídeo da simulação realizada está disponível em <https://www.youtube.com/watch?v=vqcli8xVwEM>.

Mais recentemente, no ano de 2022, a UFSC/Blumenau, por meio do Prof. Maiquel de Brito, se juntou ao grupo na criação de uma equipe para participar da *2nd SARC-BARINet Aerospace Competition*, uma competição de VANTs. Nesta edição, um enxame de VANTs deveria ser usado para fazer o combate a incêndio em uma área florestal. Além da classificação na primeira etapa da competição, essa iniciativa resultou no aceite de trabalho científico na seção de demonstrações do *16th Workshop-School on Agents, Environments and Applications (WESAAC 2022)* (de Lima et al. 2022). Um vídeo com a simulação produzida está disponível em <https://youtu.be/-bLGymzZwzI>.

O grupo de pesquisa do PROVANT vislumbra várias novas iniciativas para continu-

ar a investigação da integração de VANTS com agentes inteligentes. Um tema que nos interessa é estudo e proposição de *patterns* para programação de agentes embarcados em situações onde há necessidade de acompanhamento dos planos em execução considerando questões temporais. Das questões as serem investigadas, temos, por exemplo: o plano de atuação atual terminará a tempo, continua viável, novas oportunidades surgiram que apontem para outros planos de atuação? Esse novo projeto está sendo lançado como possível tema de mestrado em 2022/2, e também vai contar com a colaboração com o professor da UFSC/Blumenau Maiquel de Brito. O professor Maiquel, inclusive, está colaborando intensamente no desenvolvimento da infraestrutura de interação entre os agentes Jason e o ROS.

Por fim destaca-se a saída sabática do professor Leandro para a Universidade de Manchester (UK), onde será trabalhada a temática de agentes e sistemas autônomos. O supervisor será o professor Michael Fisher, o qual coordena a rede de pesquisa intitulada *Autonomy and Verification Network* <https://autonomy-and-verification.github.io/>.

Referências

- Boissier, O., Bordini, R. H., Hübner, J., and Ricci, A. (2020). *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo*. MIT Press.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. John Wiley & Sons.
- de Lima, B., Silvestre, I., Dias, P. H., Becker, L.

B., Hübner, J. F., and de Brito, M. (2022). *Distributed UAV-swarm control using BDI agents*. In *16th Workshop-School on Agents, Environments and Applications (WESAAC 2022)*, Blumenau, Brasil.

Fahlstrom, P. G. and Gleason, T. J. (2012). *Introduction to UAV Systems*. John Wiley & Sons.

Hübner, J. F., Sichman, J. S., and Boissier, O. (2007). *Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels*. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395.

Menegol, M. S., Hübner, J. F., and Becker, L. B. (2018). *Coordinated UAV search and rescue application with JaCaMo*. In *Proc. of 16th International Conference International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2018)*, volume 10978 of LNCS, pages 335–338.

Santos, F. (2015). *Avaliação do uso de agentes no desenvolvimento de aplicações com veículos aéreos não-tripulados*. Master's thesis, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil.

Santos, F. R., Hübner, J. F., and Becker, L. B. (2015). *Concepção e análise de um modelo de agente BDI voltado para o planejamento de rota em um VANT*. In *IX Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações (WESAAC 2015)*, pages 66–77. ■

Autor: Ricardo J. Rabelo

Introdução

As indústrias têm gradualmente adotado o modelo da Indústria 4.0 para melhorar sua eficiência geral com sustentabilidade, ao mesmo tempo em que lidam com ciclos de vida mais curtos de produtos e serviços, e em vários setores cada vez mais customizados ou personalizados. Beneficiando-se dos avanços em automação industrial, tecnologias de informação e comunicação (TIC), sistemas de chão de fábrica e equipamentos vêm também gradualmente se transformando em entidades mais ativas, autônomas, colaborativas e inteligentes.

A concepção de sistemas de automação voltados para suprir as necessidades da Indústria 4.0 envolve uma série de “princípios de projeto” (*design principles*), tais como:

- Controle distribuído, descentralizado e colaborativo.
- Autonomia de sistemas e pessoas.
- Supervisão inteligente e resiliência dos sistemas computacionais e equipamentos.
- Arquiteturas de integração *plug-and-play* e orientadas a serviços.
- Conformidade de TI, governança, segurança cibernética e privacidade de dados.
- Comportamento emergente e auto-organização de sistemas de manufatura.
- Gestão *data-driven* baseada na coleta de dados em tempo real.
- Virtualização/digitalização de processos e recursos.

- Interação simbiótica entre sistemas, máquinas e pessoas.

Uma das estratégias mais adotadas nesse processo de transformação digital da Indústria 4.0 é o desenvolvimento de sistemas computacionais nos quais os gestores são providos de um ambiente mais integrado, mais intuitivo, mais sistemático e mais preciso para mais confiáveis e ágeis tomadas de decisão.

Apesar dos benefícios trazidos por tais ambientes *data-driven*, estes têm vindo a trazer maior complexidade e estresse aos gestores. A prática vem mostrando que eles têm sido cada vez mais expostos a enormes quantidades de informações sobre suas empresas e processos, onde muitas ações de verificação, análise, supervisão e tomada de decisões críticas precisam ser realizadas com maior frequência e rapidez.

Vários trabalhos na literatura vêm ressaltado a subestimação do impacto disso sobre os trabalhadores e, ao mesmo tempo, sobre os novos requisitos de interações entre humanos, máquinas e sistemas. Tais trabalhos apontam para a necessidade de uma mudança na forma como trabalhadores e máquinas interagem visando aumentar a colaboração entre eles e a eficiência operacional bem como uma maior satisfação dos trabalhadores, na forma de uma maior simbiose e inteligência na interação.

Há diferentes abordagens para tal. Uma delas são os “robôs de software”, ou simplesmente “softbots”. Um softbot pode ser definido como um sistema virtual instalado em um determinado ambiente virtual, que automatiza e ajuda as pessoas na execução de tarefas,

combinando recursos de conversação, inteligência, autonomia, proatividade e automação de processos.

No contexto da Indústria 4.0, um “softbot 4.0” pode ser definido como um agente social inteligente capaz de auxiliar os operadores na gestão e automação de processos de negócios, sistemas computacionais e ativos de produção, em um ambiente industrial em rede imerso na Internet das Coisas, Serviços e Pessoas. Usando tecnologias de processamento de linguagem natural e interfaces inteligentes de voz, texto, gestos, etc., um softbot pode entender e aprender com os interesses e comportamentos de seus usuários, executar tarefas e serviços em seu nome, e responder considerando a intenção, o contexto e as restrições dos processos.

Os softbots não visam substituir as interfaces nativas humano-máquina. Eles representam um tipo de interação humana de mais alto nível, incluindo trazer maior flexibilidade e usabilidade frente aos usuais menus fixos e predefinidos acessados via voz, teclado e/ou mouse. A interação dos softbots com humanos pode ser feita por diferentes meios, como navegadores Web e computadores desktop, dispositivos móveis, holografia, realidade aumentada, linguagem natural, *smart devices*, etc.

Softbots não é um assunto novo. O primeiro artigo sobre um softbot que simplesmente respondia a perguntas (*chatbots*) foi publicado há quase 60 anos, de um software desenvolvido no MIT que permitia aos usuários interagir com computadores via linguagem natural. Com o advento das redes de computadores, um maior desenvolvimento da enge-

nharia de software e da inteligência artificial, dentre outros, atualmente já é bastante popular o uso de chatbots pelas empresas, nas mais diversas áreas gerais de aplicação, como turismo, governo, bancos, lojas, saúde, etc. “Assistentes pessoais” surgiram mais recentemente para ajudar os usuários em diferentes atividades cotidianas, podendo-se citar a *Cortana*, *Siri* e *Alexa*. Por outro lado, a sua adoção na manufatura e em chão de fábrica é bem recente e ainda bastante limitada, basicamente funcionando para perguntas sobre questões bem simples sobre uma máquina, um produto, etc. Como se poderá observar mais a frente neste texto, a questão é que a Indústria 4.0 abrange muitos cenários bem mais complexos e acrescenta vários outros requisitos para interação humano-sistemas-máquinas que vão muito mais além do que os suportados por clássicos chatbots.

Propriedades de um Softbot

Para ser considerado um softbot, ele deve ter as seguintes propriedades:

- (Algum nível de) *Conhecimento* sobre os assuntos que têm que lidar, considerando também que esse conhecimento pode ser continuamente enriquecido ao longo do tempo, tanto pelos projetistas do softbot, quanto pelo próprio softbot graças à sua autonomia e inteligência.
- (Algum grau de) *Autonomia* para raciocinar, evoluir, planejar e deliberar sobre por que, quando, onde e como fazer, com quem se comunicar, sobre quanto as ações escolhidas custarão em termos de tempo, desempenho, recursos, etc.

- (Alguns níveis de) *Inteligência* para aprender com suas interações (com humanos, sistemas, máquinas e sensores) e evoluir de acordo com; fazer inferências sobre informações imprecisas ou incompletas presentes nas conversas; e filtrar e selecionar fontes de informação corretas, confiáveis e menos custosas financeiramente e computacionalmente.
- *Coordenação* de ações dados os diferentes fluxos de execução (contextos) e sequências de invocações (síncronas e assíncronas) para diferentes sistemas e usuários.
- (Alguns graus de) *Flexibilidade* para atuar e interagir em diferentes cenários em diferentes contextos.
- (Alguns graus de) *Adaptabilidade* para agir e interagir consistentemente com diferentes atores em relação às suas preferências, emoções e o conhecimento que tem sobre eles.
- *Sociabilidade* de interagir com diferentes atores usando protocolos de comunicação de baixo nível e alto nível e semântica.
- *Integrabilidade* com outros sistemas (incluindo com outros softbots), com ambientes computacionais e dispositivos móveis distribuídos e heterogêneos;
- *Interoperabilidade* para garantir a troca de informações (controle e dados) sem problemas.
- (Alguns níveis de) *Segurança* para se proteger de ataques maliciosos externos e de ser acessado por atores não autorizados.
- (Alguns níveis de) *Autogestão* para se autoavaliar em relação às suas metas e métricas de desempenho; supervisionar a execução de suas ações; para monitorar o

ambiente de computação em que está executando e seu status de execução; e ser resiliente de modo a tomar as medidas adequadas para permanecer 'vivo'.

A implementação dessas propriedades pode variar em complexidade. A maioria dos softbots são implementados como sistemas baseados em regras, programados para reconhecer certos termos e padrões pré-definidos dos quais eles podem responder com respostas pré-definidas. Softbots mais sofisticados são sistemas baseados em IA e podem fazer várias ações automaticamente ou autonomamente. Eles usam recursos de processamento de linguagem natural para entender o contexto, as intenções e as emoções das solicitações dos usuários, e isso evolui à medida que interagem e aprendem com as conversas. Sistemas baseados em IA podem adotar diferentes representações de conhecimento, incluindo regras em *backward* ou *forward*.

Diferenças de um Softbot para outros conceitos equivalentes

Vários conceitos têm algumas profundas intersecções com o que é um softbot, e não há uma clara fronteira conceitual separando-os quando se olha para sua "equivalência funcional" ou "objetivo principal". Dependendo do propósito, arquitetura e modelagem do softbot, isso pode levar até mesmo a incorretas interpretações.

Adotando como premissa que um softbot é uma "interface inteligente" que conecta humanos, sistemas e máquinas com outros para executar certas ações, em termos gerais pode-se afirmar que:

- Um softbot não é um *agente* (de software), mas pode ser se for implementado

como tal, ou seja, se ele suportar (pelo menos) as propriedades obrigatórias dos agentes: (algum) conhecimento para resolver algum problema(s), sociabilidade e autonomia. Isso acontece independentemente de sua arquitetura interna (por exemplo, BDI) ou se ele pode se mover (agentes móveis) através da rede para executar suas tarefas dentro do ambiente digital no mesmo ou outro domínio de segurança diferente.

- Um softbot não é um *sistema inteligente* (ou baseado em conhecimento ou em IA), mas pode ser se ele puder perceber seu ambiente, sensores e humanos, raciocinar e planejar de acordo com ou se adaptar a, e executar ações apropriadas considerando suas metas, requisitos do plano e as condições do ambiente computacionais onde atuará. Isso significa que mesmo ações ou respostas muito simples, rápidas e diretas aos usuários podem ser resultado de um processo de profundo raciocínio ou ser a ação necessária para um problema grave em um sistema crítico.
- Um softbot não é um *assistente pessoal* (virtual ou inteligente), mas pode ser se ele agir como um softbot mais dedicado a lidar com as necessidades particulares de um ou de alguns usuários com objetivos semelhantes.
- Um softbot não é um *chatbot*, mas pode ser se ele basicamente suportar o processamento de linguagem (natural ou na forma de termos pré-definidos) relacionado a conversas sobre um determinado assunto.
- Um softbot não é um simples *robô* (de software), mas pode ser se ele for apenas

programado para fazer automaticamente ações ou missões muito específicas (e quase sempre sem interação humana).

- Um softbot não é um *immobot*, mas pode ser se ele for especificamente projetado para monitorar a si mesmo (por exemplo, quando ligado a um equipamento) para mantê-lo funcionando.
- Um softbot não é um *holon* (dos sistemas holônicos de manufatura), mas pode ser se ele for projetado para representar ativamente entidades físicas de uma fábrica, como peças, paletes, máquinas, sensores, objetos industriais inteligentes.
- Um softbot não é um *avatar*, mas pode ser se ele for projetado para representar, agir e interagir com outros usuários, sistemas, gêmeos digitais ou entidades físicas de uma fábrica em nome do usuário.
- Um softbot não é um *sistema especialista* ou tutor inteligente, mas pode ser se ele for projetado para orientar e ajudar os humanos a resolver algum problema específico dentro de um determinado domínio de conhecimento; ou ensinar humanos adaptativamente, de acordo com seus conhecimentos prévios, perfil atual, nível de respostas (profundidade, número de acessos, etc.), entre outros aspectos.
- Um softbot não é um *serviço* (de software), mas pode ser se ele estiver imerso em container local, ou numa nuvem ou arquitetura tipo SaaS (*Software-as-a-Service*).
- Um softbot não é um sistema de *workflow* (ou de RPA [*Robotic Process Automation*] ou de BPM [*Business Process Management*]), mas pode agir como tal se for projetado para automatizar a execução de tarefas (geralmente) pré-definidas em

com o auxílio de realidade aumentada. No Exemplo 2 (figura 2) é mostrado como um softbot pode fazer coisas automaticamente pelo usuário; no caso, ao final de cada turno do seu trabalho gerar relatório de desempenho. No Exemplo 3 (figura 3) é mostrado uma interface mais textual, onde um operador vai perguntando inúmeras coisas para o softbot e este autonomamente interage com outros sistemas e fontes de dados para buscar informações e satisfazer as necessidades daquele.

Conclusões

Considerando os problemas que hoje os gestores e operadores têm que lidar em suas atividades diárias, verifica-se que softbots são meio muito poderoso para ajudar na gestão inteligente da produção. Ajudam a esconder a complexidade no acesso às necessárias informações, auxiliando os gestores na busca em enormes silos de dados (às vezes duplicados e armazenados em diferentes fontes), gerados por sistemas diferentes e distribuídos.

Esse acesso também pode ser mais rápido e confiável, pois o softbot sabe onde obter as informações necessárias e faz isso já lidando com os problemas de interoperabilidade.

Além disso, pode ajudar a que apenas pessoas autorizadas possam solicitar ou ter acesso a algum tipo de informação, auxiliando assim as empresas na conformidade com a LGPD e na governança.

Os softbots podem trazer informações de uma forma mais sintetizada, às vezes compondo um determinado conjunto de dados ou fazendo alguns cálculos anteriores antes de enviá-los/mostrá-los aos usuários. Em outras palavras, os softbots podem fornecer respostas mais próximas do que e como os usuários

pediram.

A forma como os usuários interagem com o softbot e o softbot interage com eles é semelhante a uma linguagem natural, ajudando a criar um ambiente de trabalho mais simbiótico, "amigável". Isso também pode acelerar a execução de processos de negócios.

Os softbots ajudam os usuários não apenas em suas solicitações sob demanda, mas também a si mesmos e à própria empresa na execução de tarefas programadas, complexas, repetitivas e/ou para execução de tarefas automaticamente ou em nome de usuários. No entanto, eles também podem fazer ações de forma proativa, por exemplo, para monitorar permanentemente alguns aspectos de produção e alertar os usuários quando necessário. Como resultado, os usuários podem dedicar mais tempo a atividades que agregam mais valor à empresa, e que realmente requerem maior inteligência, criatividade e importantes tomadas de decisão. Os softbots podem raciocinar sobre dados e ajudar os usuários em análises de negócios/produção. Eles também podem ajudar os usuários a entender o que os sistemas sugerem, por exemplo, quando de ações preditivas ou prescritivas.

Apesar de todos esses potenciais, muitos projetos de softbots não têm atingido as expectativas. De acordo com uma pesquisa sobre chatbots em 2019 considerando 529 gestores da América do Norte e Europa de várias empresas de diferentes indústrias: para quase 50% dos usuários, o chatbot deu respostas inúteis; quase 40% disseram que o chatbot muitas vezes simplesmente redirecionava os usuários para FAQs dada a sua incapacidade de fornecer alguma resposta; 39% afirmaram que os chatbots forneceram sugestões de má qualidade; 59% afirmaram que os chatbots en-

tendiam mal as solicitações dos usuários bem como não conseguiam lidar com nuances do diálogo humano; e para 30% o chatbot executou comandos imprecisos. A pesquisa também apontou duas principais razões para esses problemas. O primeiro refere-se à falta de um envolvimento mais profundo dos gestores na formulação do projeto e suas interações. O segundo refere-se à falta de conhecimento dos desenvolvedores de softbot, pois a implementação de interfaces interativas - conforme exigido por esse tipo de sistemas - é muito diferente do desenvolvimento de interfaces web ou de aplicativos para celulares.

A área geral dos softbots é relativamente antiga, mas agora está ficando cada vez mais madura para ser usada em casos reais. No entanto, ainda existem várias e complexas questões intrínsecas para se lidar e que representam pontos abertos para novas pesquisas. Por exemplo, a necessidades de um uso mais intenso do aprendizado de máquina na compreensão, processamento e raciocínio de mensagens; interoperabilidade semântica, inclusive lidando com diferentes expressões e contextos humanos; controle e compartilhamento de dados de privacidade sobre as informações dos usuários; cibersegurança, garantindo que os softbots não sejam hackeados ou façam ações maliciosas; suporte para operadores com deficiência física ou mental; integração com diversos tipos de *smart devices* (óculos, luvas, vestíveis, etc.); e geração/dedução automática de "workflows" sobre o que fazer e quais sistemas acessar, etc., a partir da interpretação de pedidos de usuários.

Referências

RABELO, R. J.; ROMERO, D.; ZAMBIAZI, S. P.. Softbots Supporting the Operator 4.0 at

Smart Factory Environments. In: Proceedings APMS'18 - IFIP Advanced Production Management Systems, New York: Springer, 2018. p. 456-464.

RABELO, R. J.; ZAMBIAZI, S. P.; ROMERO, D.. Collaborative Softbots: Enhancing Operational Excellence in Systems of Cyber-Physical Systems. In: Proceedings PRO-VE'2019 - 20th IFIP Working Conference on Virtual Enterprises, New York: Springer, 2019. p. 55-68.

MACHADO, B. A.; RABELO, R. J.; ZAMBIAZI, S. P.; ROMERO, D.. Production Management as-a-Service: A Softbot Approach. In: Proceedings APMS'2020 - International Conference on Advanced Production Management Systems. New York: Springer, 2020. p. 19-30.

RABELO, R. J.; ROMERO, D.; ZAMBIAZI, S. P.; MAGALHAES, L. C.. When Softbots Meet Digital Twins: Towards Supporting the Cognitive Operator 4.0. In: Proceedings APMS'21 - IFIP International Conference on Advances in Production Management Systems. New York: Springer, 2021. p. 37-47.

ZAMBIAZI, L. P.; RABELO, R. J.; ZAMBIAZI, S. P.; LIZOT, R.. Supporting Resilient Operator 5.0: an Augmented Softbot Approach. In: Proceedings APMS'22 - IFIP International Conference on Advances in Production Management Systems. New York: Springer, 2022. *To be published.*

Aplicações de aprendizado de máquina em ensaios de compressores de refrigeração

Autores: Ahryman S. B. Nascimento, Bernardo B. Schwedersky, Gabriel Thaler, Rodolfo C. C. Flesch

Introdução

Com o aumento da competitividade e alterações nos padrões de desempenho nos últimos anos, empresas fabricantes de compressores de refrigeração têm investido na aplicação de estratégias para aumento de eficiência e inteligência de seus produtos (Embraco, 2022; SE-COP, 2022). A utilização de técnicas de aprendizado de máquina pode contribuir na redução de custos de ensaios e no desenvolvimento de novas estratégias de avaliação de seus produtos.

O compressor de refrigeração é um dos quatro componentes principais de um sistema de refrigeração, sendo os outros três o evaporador, o condensador e o dispositivo de expansão. Compressores de refrigeração têm como objetivo elevar a pressão do fluido refrigerante de forma a permitir os processos de troca de calor e mudança de fase nos outros componentes. Sua estrutura típica consiste em uma câmara de compressão de fluido cujo volume é continuamente variado por meio de um mecanismo biela-manivela acoplado a um motor elétrico. Tal conjunto é alojado dentro de um corpo de aço selado hermeticamente. Na Fig. 1 é possível observar uma amostra de compressor de refrigeração com o corpo em acrílico, o que permite visualizar os principais componentes internos.

No contexto da parceria entre o Laboratório de Instrumentação e Automatização de Ensaios, grupo de pesquisa com forte atuação de professores e alunos do PosAutomação, e a empresa fabricante de compressores Nidec Global Appliance, diversos projetos de aplicação de técnicas de aprendizado de máquina

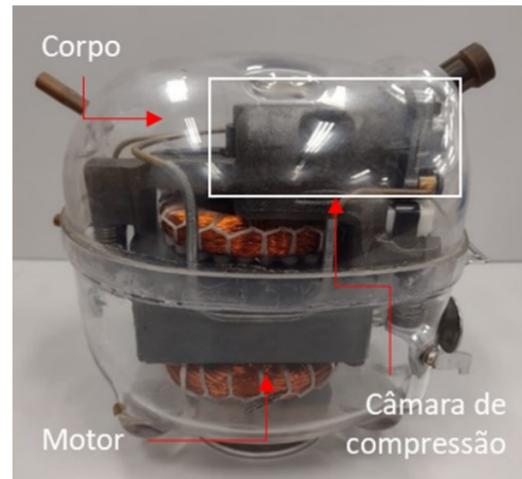


Fig. 1: Compressor hermético de refrigeração.

em ensaios de compressores de refrigeração foram propostos. Este artigo tem como objetivo apresentar os principais trabalhos desenvolvidos nessa linha e seus impactos para os processos da empresa. Na seção 2 são apresentadas ferramentas desenvolvidas para aceleração de ensaios de desempenho energético de compressores de refrigeração. Na seção 3 são apresentadas ferramentas desenvolvidas para monitoramento do processo de amaciamento de compressores. Na seção 4 é apresentado um método de sensoriamento virtual para estimar condições de operação de sistemas de refrigeração por meio de grandezas externamente mensuráveis. Considerações finais são apresentadas na seção 5.

Identificação de regime permanente e prognóstico de grandezas em ensaios de desempenho energético

Um dos ensaios realizados rotineiramente pela indústria de refrigeração é o ensaio de desempenho energético. Ele consiste na medição de algumas grandezas associadas ao funcionamento do compressor de refrigeração em condições específicas de operação, com desta-

que para o consumo energético (medido em watt) e a capacidade de refrigeração (também expressa em watt), a qual pode ser interpretada como a capacidade do compressor, em um dado circuito de refrigeração, de permitir a troca de calor do meio a ser refrigerado para o meio externo. Esse tipo de ensaio é realizado em várias etapas do ciclo de vida do compressor, com destaque para as etapas de projeto, certificação por órgãos reguladores e verificação da qualidade de produção.

Na indústria, tal ensaio é realizado em bancadas automatizadas, as quais são capazes de emular o funcionamento de um circuito de refrigeração, controlando as condições de operação do compressor enquanto as grandezas de interesse são medidas. Essas medições são realizadas, tipicamente, de acordo com normas internacionais, as quais regulamentam vários aspectos da realização do ensaio, que vão desde métodos para a medição das grandezas até tempos de assentamento requeridos para as variáveis de processo e limites de variação das mesmas. As condições requeridas pelas normas, bem como as dinâmicas associadas aos sistemas de refrigeração, as quais são tipicamente lentas, fazem com que a duração média de tais ensaios seja longa. Até em bancadas com um nível alto de automação, os tempos médios superam duas horas.

Abordagens para aceleração do ensaio de desempenho por meio de técnicas de aprendizado de máquina são temas de investigação em diversos trabalhos. Essas estratégias abordam dois desafios distintos: a identificação da condição atual do ensaio, de forma a garantir que o ensaio se encontre em uma

condição favorável ao prognóstico do valor final; e o prognóstico do valor final das variáveis de interesse, em especial capacidade de refrigeração e consumo energético. A seguir, são detalhados ambos os problemas, sendo apresentadas as abordagens baseadas em aprendizado de máquina para solução desses problemas.

Identificação de regime permanente do ensaio

O ensaio de avaliação de desempenho energético envolve o controle de diversas condições de operação do compressor sob ensaio. Destaca-se o controle das pressões associadas às linhas de sucção e de descarga do fluido refrigerante, o qual é necessário para garantir que o compressor opere sob condições padronizadas. Devido ao controle dessas pressões, e também ao transitório associado ao aquecimento natural do compressor, existe uma região do ensaio de avaliação de desempenho em que o prognóstico dos valores finais pode conter muito erro, já que não são garantidas condições mínimas de que o ensaio se encontra próximo das condições desejadas para medição das grandezas de interesse. Dessa forma, é necessária uma ferramenta capaz de identificar o estado atual do ensaio, classificando-o em regime transitório, no qual ainda não são garantidas as condições mínimas para realização de prognóstico dos valores finais das variáveis de interesse, ou regime permanente, no qual as pressões já estão controladas próximas dos valores de referência e as temperaturas do compressor e do ambiente de medição encontram-se com taxas de variação pequenas.

Um ensaio de avaliação típico é apresentado na Fig. 2. Nesse ensaio, os valores de capacidade de refrigeração sofrem grandes variações durante a porção inicial do ensaio, devido ao assentamento das pressões de sucção e de descarga. Após o assentamento dessas variáveis, as grandezas de interesse convergem aos valores finais. A condição de regime permanente do ensaio é apresentada na porção inferior da Fig. 2, sendo que o ensaio atinge uma condição de assentamento suficiente para realização de prognósticos dos valores finais após o primeiro quarto do ensaio.

Esse problema foi modelado como uma tarefa de classificação binária, na qual a ferramenta inteligente deve ser capaz de classificar se o ensaio encontra-se em regime permanente ou não. Essa ferramenta considerou como variáveis de entrada os valores instantâneos e passados das grandezas de interesse (capacidade de refrigeração e consumo), valores atuais e passados da temperatura do corpo do compressor e variáveis binárias que indicam se as pressões de sucção e de descarga encontram-se dentro de um limite de tolerân-

cia ao redor dos valores de referência.

A ferramenta implementada, baseada em aprendizado de máquina, consistiu em um comitê (*ensemble*) de redes neurais artificiais (RNAs). Já que nesse tipo de problema a ocorrência de falsos positivos é indesejada, ou seja, a identificação do regime permanente antes do assentamento completo do ensaio é pior do que a identificação atrasada da entrada em regime permanente, optou-se por empregar um treinamento que minimize a ocorrência de falsos positivos.

Prognóstico das variáveis de interesse

Por meio da utilização do modelo de identificação de entrada em regime permanente descrito na seção 2.1, é possível identificar quando o ensaio alcança uma condição adequada para a realização do prognóstico do valor final das variáveis de interesse. Esse prognóstico consiste na predição do valor final de cada uma das variáveis com base em valores atuais e passados das medições realizadas nos ensaios.

Esse problema foi modelado como um problema de regressão, ou seja, a partir de valores atuais e passados das medições é feita uma predição do valor final das variáveis de interesse em um dado ensaio. A ferramenta de aprendizado de máquina desenvolvida consistiu em um comitê (*ensemble*) de redes perceptron de múltiplas camadas com atrasos no tempo em sua entrada. A utilização desse banco de atrasos no tempo confere ao modelo a capacidade de aprender o comportamento dinâmico das variáveis de interesse.

O problema do prognóstico é exemplificado na Fig. 3. A variável de interesse, a qual neste exemplo é a capacidade de refrigeração, passa por um pré-processamento, sendo fil-

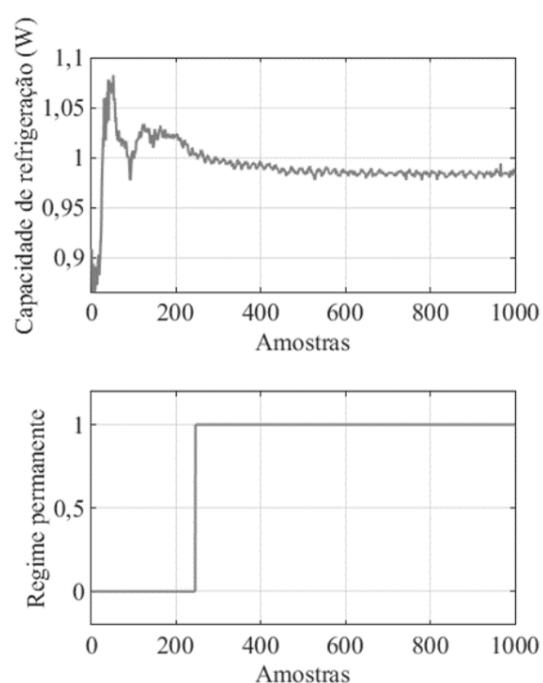


Fig. 2: Ensaio de desempenho energético e sua estabilidade.

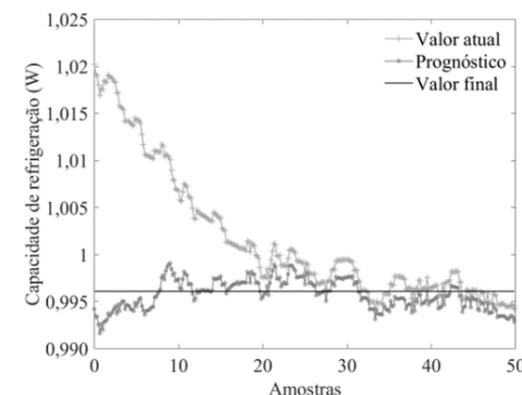


Fig. 3: Prognóstico da capacidade de refrigeração.

trada e reamostrada. Após o pré-processamento, são treinados modelos utilizando subconjuntos do dataset completo, para prever o valor final que seria obtido se o ensaio completo fosse realizado. Dessa forma, é possível obter um prognóstico do valor final, a cada instante do ensaio.

Uma ferramenta completa, considerando módulos para predição do regime permanente de cada variável de interesse e módulos para prognóstico do valor final do ensaio, foi desenvolvida e testada em bancadas reais de avaliação de compressores. A avaliação real das ferramentas indicou a capacidade da ferramenta de produzir prognósticos com erros menores que 1% em mais de 90% dos ensaios realizados. A utilização dessa ferramenta resultou em reduções de mais de 55% no tempo médio dos ensaios, o que, para o conjunto testado, representa mais de uma hora de redução em cada ensaio.

Aprendizado de máquina para identificação não-destrutiva de amaciamento

Assim como outros mecanismos que envolvem contato deslizante, as primeiras horas de operação de um compressor são marcadas por um período transitório denominado amaciamento, que é sucedido por um regime permanente tribológico (RPT), durante o qual suas características de superfície e contato perma-

necem relativamente constantes. O amaciamento, que ocorre apenas uma vez na vida útil do dispositivo, envolve fatores como desgaste abrasivo e adesivo das superfícies, alterações na lubrificação do contato, assentamento de peças e até modificações na têmpera de metais (BLAU, 2005).

Quando comparada à duração total da vida útil de um compressor, projetado para operar normalmente até por décadas, a duração do amaciamento pode ser desprezível, e em poucos casos afeta o consumidor final. Já para o campo de pesquisa e desenvolvimento em compressores, o estudo desse período pode trazer ganhos significativos, dado que o amaciamento adequado pode ser benéfico à sua operação. Ainda, segundo as principais normas para ensaios de desempenho energético, a avaliação de um compressor deve ser realizada apenas após o amaciamento total do dispositivo, de modo a evitar caracterizações que não reflitam seu desempenho em RPT.

Apesar de sua importância, são escassos os estudos de caracterização do amaciamento em máquinas complexas, com boa parte dos trabalhos avaliando ou as características de RPT dos materiais ou os efeitos do amaciamento em cenários simplificados, já que sua influência em sistemas de múltiplas superfícies de contato é de difícil caracterização. Ainda assim, é sabido que o fenômeno afeta a magnitude e as flutuações no coeficiente de atrito nos contatos, que normalmente é utilizado como parâmetro nos estudos.

Em compressores herméticos, além das dificuldades impostas pela existência de múltiplos sistemas móveis, não é possível acessar diretamente as peças durante sua operação, e mesmo a avaliação pós-operação das superfícies requer a destruição do corpo metálico do

compressor. Assim, o processo atual de amaciamento na indústria consiste na operação do dispositivo por um período predeterminado de horas, após as quais se assume que o dispositivo já se encontra amaciado. Nesse contexto, propostas de avaliação e identificação indiretas do amaciamento, além de garantir o término desse período para posterior avaliação da unidade, permitem reduzir a duração dos ensaios de amaciamento para o tempo mínimo necessário, poupando tempo e recursos financeiros.

A partir dos conceitos sobre os efeitos do fenômeno em grandezas físicas mensuráveis, foi projetada uma bancada experimental de ensaios que permite a avaliação e o monitoramento do compressor em condições controladas. Os ensaios realizados nessa bancada permitiram, então, a avaliação das grandezas possivelmente relacionadas ao fenômeno e, posteriormente, o desenvolvimento de métodos de aprendizado de máquina capazes de realizar detecção indireta do término do amaciamento.

Avaliação Experimental

A literatura sobre o fenômeno do amaciamento e sobre motores elétricos sugere que grandezas como a corrente elétrica consumida pelo motor e a vibração gerada pela operação de um dispositivo podem ser indicativos do coeficiente de atrito e do tipo de desgaste que ocorre nas superfícies (BLAU, 1992). Tais grandezas, juntamente com a vazão mássica, que é um indicativo da capacidade de refrigeração de um compressor, foram selecionadas para monitoramento na bancada experimental desenvolvida.

A bancada foi projetada de modo a possibilitar controle das pressões e temperaturas de sucção e de descarga, com capacidade de

manter as condições de operação desejadas em ensaios de longa duração. Para avaliação do amaciamento, foram realizados ensaios em alguns compressores de um mesmo modelo, sendo que para cada unidade foram realizados um ensaio de amaciamento, com duração suficiente para garantir o término do fenômeno, e, posteriormente, dois ensaios de referência, com o dispositivo já amaciado.

Para análise da corrente elétrica e da vibração nos compressores, foram extraídos o valor eficaz, curtose e variância, selecionados como features que possam representar a magnitude e variações no atrito e no desgaste durante a operação. Tais grandezas foram posteriormente avaliadas em análises comparativas, nas quais verificou-se que o assentamento do valor eficaz, curtose e variância da corrente elétrica é mais lento em ensaios de amaciamento do que em ensaios de referência. A Fig. 4 mostra o valor eficaz da corrente elétrica para uma das unidades, na qual é possível observar que o regime permanente no ensaio de amaciamento é atingido apenas em torno de 7 h, enquanto nos ensaios de referência é atingido em apenas 2 h.

A diferença entre tempos de assentamento da corrente elétrica não foi observada nas outras grandezas medidas nos ensaios, o que permite inferir que ao menos um bom palpite inicial de término do amaciamento pode ser obtido a partir da observação da corrente elétrica. Tal informação foi utilizada como critério de avaliação dos métodos de detecção não supervisionada desenvolvidos, descritos a seguir.

Detecção não supervisionada

Dada a escassez de métodos preexistentes que permitam detecção do término do período de amaciamento, foram buscadas propostas que

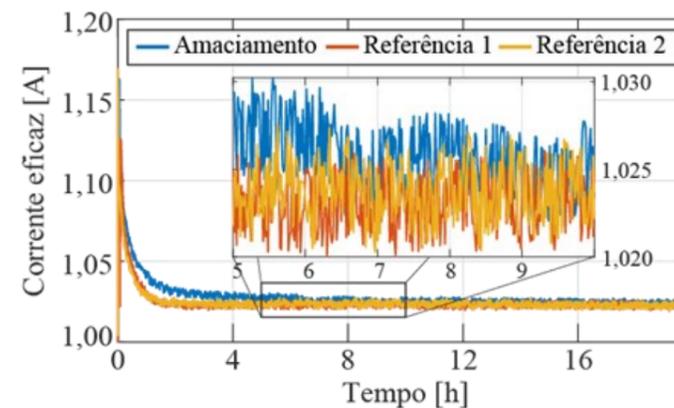


Fig. 4: Diferença no valor RMS da corrente elétrica entre ensaios de amaciamento e de referência.

permitam avaliação não supervisionada dos dados. Após análise inicial, optou-se pelo algoritmo de agrupamento *k-means*, que busca minimizar a soma do erro quadrático entre elementos de um mesmo grupo.

Para aplicação na ferramenta não supervisionada, as séries temporais obtidas nos ensaios experimentais foram filtradas com filtro de médias móveis e processadas utilizando janelas deslizantes em espaço de atraso. Tal método, que consiste em dividir a série original em múltiplos vetores de amostras com espaçamento temporal, permitiu formação dos dados em uma matriz de observações adequada para o agrupamento.

Os parâmetros de pré-processamento e as grandezas extraídas dos ensaios foram avaliados em busca em grade. Em alguns dos agrupamentos resultantes, foram observados padrões possivelmente indicativos do processo de amaciamento, i.e., a classificação de um grupo que ocorre apenas no início de ensaios e que perdura por mais tempo em ensaios de amaciamento do que em ensaios de referência. A Fig. 5 apresenta um exemplo de agrupamento considerado adequado para a aplicação, obtido por meio da curtose da corrente elétrica do compressor.

A partir da análise dos agrupamentos, foram

desenvolvidos métodos automáticos para detecção do término do amaciamento, baseados na proporção de grupos em uma dada janela de ensaio. Tais métodos foram, então, utilizados para análise de grandezas possivelmente relacionadas ao fenômeno.

Dentre as grandezas analisadas, a vazão mássica, a curtose da vibração e a curtose, variância e valor eficaz da corrente elétrica geraram resultados válidos para detecção do amaciamento. A partir da frequência de detecção de cada uma das grandezas, utilizou-se o método de estimativa de densidade por kernel para a obtenção de funções de densidade cumulativa que representam a probabilidade de o compressor ter atingido RPT. A Fig. 6 apresenta as curvas de probabilidade nos ensaios de amaciamento para cada uma das grandezas selecionadas. A partir das curvas de probabilidade pôde-se observar que o valor eficaz e a variância da corrente elétrica foram os mais condizentes com as definições do amaciamento e com as observações empíricas das análises comparativas.

Os resultados obtidos na aplicação de aprendizado não supervisionado no contexto do amaciamento, além de aumentarem a confiabilidade do estado de regime permanente de compressores para ensaios de desempenho, também indicam a possível aplicabilidade

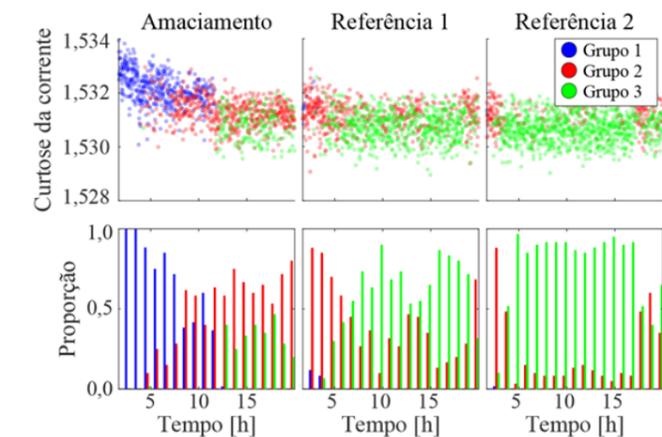


Fig. 5: Exemplo de agrupamento e proporção de grupos ao longo de ensaios de amaciamento e de referência.

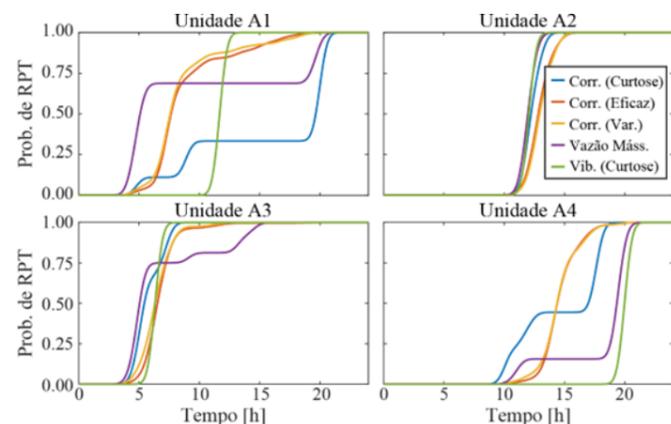


Fig. 6: Probabilidade de RPT em ensaios de amaciamento para grandezas selecionadas.

de dos métodos desenvolvidos para outros campos de análise, como a detecção de falhas e outros problemas de detecção de anomalias. Nesses contextos, os métodos desenvolvidos podem ser utilizados para indicar comportamentos diferentes da operação normal do compressor, aumentando, assim, sua confiabilidade e vida útil.

Sensoriamento virtual de condições de operação de sistemas de refrigeração por meio de grandezas externamente mensuráveis

Sensoriamento virtual consiste na utilização de medições de grandezas físicas sabidamente relacionadas com a grandeza de interesse e do uso de modelos de regressão para estimar a grandeza de interesse por meios dessas medições (LIU et al., 2009). Tal modelo pode ser fenomenológico ou baseado em observações obtidas experimentalmente. Diversos trabalhos da literatura abordam a utilização de sensores virtuais na indústria para tarefas de controle de qualidade de processos, diagnóstico e detecção de falhas, redução de custos por eliminação de sensores, estimação indireta de grandezas, dentre outras aplicações.

Uma medição tipicamente realizada para avaliação de desempenho de compressores

de refrigeração é a medição das temperaturas de operação. Um sistema de refrigeração funciona basicamente para gerar um diferencial de temperatura entre o evaporador (local a ser refrigerado) e o condensador. Dessa forma, a condição de operação de tal sistema é definida pelas temperaturas de evaporação e de condensação. Durante ensaios de compressores, tais temperaturas são avaliadas por meio da medição das pressões na entrada (sucção) e na saída (descarga) do compressor. Medindo as pressões de sucção e de descarga, é possível obter as temperaturas de evaporação e de condensação, respectivamente, por meio das propriedades termodinâmicas do fluido refrigerante. A medição tipicamente não é feita em termos de temperaturas porque as condições de operação podem mudar de forma rápida e a medição de temperatura tem uma dinâmica associada que é muito lenta, dado que a temperatura do ponto de medição precisa equalizar com a de mudança de fase do fluido para que a medição seja representativa.

Para avaliação de tais pressões quando o compressor já está instalado em um sistema de refrigeração, é necessário abrir o sistema de refrigeração, inserir os transdutores e fazer uma nova carga de fluido refrigerante. Dessa forma, não há um meio não intrusivo de avaliar a condição de operação imposta a um compressor quando o mesmo já se encontra instalado em um sistema de refrigeração. Tal possibilidade seria interessante, por exemplo, para avaliar se o sistema está operando na condição definida em projeto, para predição de falhas e identificação de defeitos.

Para contornar tal limitação, foi proposto um sensor virtual capaz de estimar a temperatura de evaporação de forma não invasiva, sem a necessidade de violar o sistema. Uma visão geral do método é apresentada nas

subseções a seguir.

Aquisição de grandezas correlacionadas às temperaturas de operação

Tendo em vista a relação termodinâmica entre as temperaturas de evaporação e a pressão de sucção, é coerente assumir que uma grandeza relacionada à pressão também será relacionada à temperatura.

Uma grandeza diretamente relacionada às pressões de operação do compressor consiste na vibração medida no corpo do mesmo, uma vez que os caminhos de transmissão de vibração do kit mecânico para o corpo metálico são impactados pelo valor de tais pressões. Trabalhos encontrados na literatura confirmam o potencial da vibração como entrada de modelos para classificar a condição de operação e identificar defeitos em compressores de refrigeração. Dessa forma, tal grandeza foi selecionada como entrada do modelo para o sensor virtual.

Para adquirir dados de vibração em diferentes condições de operação, amostras de compressores foram instaladas em uma bancada capaz de controlar as pressões de sucção e de descarga. Para medir a velocidade vibratória no corpo do compressor, foram empregados um acelerômetro e uma placa de aquisição de dados com taxa de aquisição de 50 kHz.

Ao todo, cinco unidades do mesmo modelo de compressor foram ensaiadas na bancada em 461 condições de operação diferentes. Foram realizadas 6363 medições de vibração.

Tratamento dos dados e treinamento dos modelos

Os dados brutos de vibração foram processa-

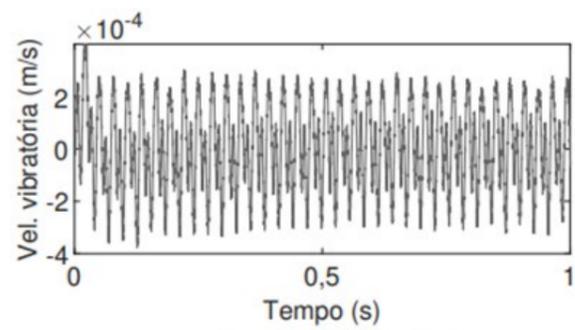
dos de forma a serem utilizados como entrada dos modelos para estimativa da temperatura de evaporação. Foi feita a análise do espectro de frequência por meio de transformada rápida de Fourier, seguido por um filtro passa-banda com banda passante de 200 Hz e 10% de sobreposição, totalizando 139 bandas analisadas. A Fig. 7 apresenta as etapas de processamento.

Com o espectro filtrado como entrada e a temperatura de evaporação como saída, foi realizado o treinamento de diferentes modelos de aprendizado de máquina. Para o treinamento, 80% dos dados disponíveis foram utilizados, sendo os 20% restantes reservados como conjunto de teste. Foram treinados modelos de regressão linear múltipla (RL) e RNA. Os hiperparâmetros de cada modelo foram ajustados por meio de um subconjunto do conjunto de treinamento.

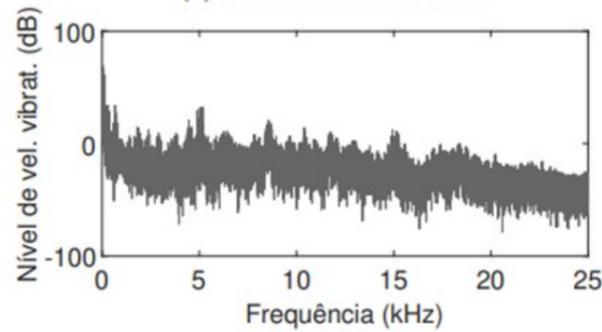
Resultados e discussões

A Tabela 1 apresenta os resultados dos modelos treinados para estimar a temperatura de evaporação. Os resultados dos modelos indicam a capacidade de estimar a temperatura de evaporação com uma raiz do erro médio quadrático (RMSE) abaixo de 2 °C utilizando apenas dados de vibração. Adicionalmente, o modelo RNA apresentou um RMSE 0,45 °C menor do que o modelo RL, indicando uma certa não linearidade na relação entre o espectro filtrado e a temperatura de evaporação. Dessa forma, o modelo RNA mostra-se mais adequado para a tarefa.

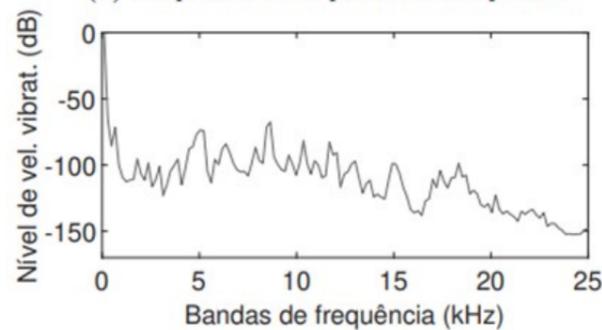
Os resultados indicam a capacidade de estimar a temperatura de evaporação por meio dos dados de vibração. A ferramenta proposta abre a possibilidade de avaliação da temperatura de evaporação de forma indireta e não invasiva em sistemas de refrigeração,



(a) Velocidade vibratória



(b) Amplitude do espectro de frequência



(c) Amplitude do espectro de frequência filtrada

Fig. 7: Etapas de processamento dos dados de vibração.

podendo ser utilizada, por exemplo, próxima à linha de montagem como um método de avaliação da conformidade de produtos. Outra possibilidade seria avaliação de sistemas de refrigeração em campo, sem a necessidade de levá-los para laboratórios.

Considerações finais

As aplicações de técnicas de aprendizado de máquina em compressores de refrigeração discutidas neste trabalho contribuíram para redução de tempo e custo de ensaios de desempenho, aumento da confiabilidade dos resultados dos ensaios e forneceram novas possibilidades de avaliação das condições de operação de sistemas de refrigeração. Além dos

Técnica	RMSE (°C)	R^2
RL	1,79	0,92
RNA	1,34	0,95

Tabela 1: Resultados dos modelos treinados no conjunto de teste — Temperatura de evaporação.

resultados científicos inovadores resultantes desses estudos, que foram publicados em congressos e periódicos, é importante reforçar a preocupação do PosAutomação em desenvolver trabalhos com aplicação direta ao setor produtivo, por meio de trabalhos em parceria e convênios institucionais.

Referências

- Embraco. 2022. Nidec Global Appliance responds to new energy labels with smarter compressors and motors - Embraco. [online] Disponível em: <https://www.embraco.com/en/nidec-global-appliance-responds-to-new-energy-labels-with-smarter-compressors-and-motors/> [Acessado em 25 de agosto de 2022].
- SECOP, 2022. New Secop R&D Center in Spišská Nová Ves, Slovakia | Secop – Sustainable Cooling Solutions. [online] Secop.com. Disponível em: <https://www.secop.com/updates/news/news-show/new-secop-rd-center-in-spisska-nova-ves-slovakia> [Acessado em 25 de agosto de 2022].
- BLAU, P. J. On the nature of running-in, *Tribology International*, v. 38, n. 11-12, 2005, p. 1007-1012. doi: 10.1016/j.triboint.2005.07.020
- BLAU, P. J. (ed.). *ASM Handbook Volume 18: Friction, Lubrication, and Wear Technology*. Materials Park: ASM International, 1992.
- LIU, L.; KUO, S. M.; ZHOU, M. Virtual sensing techniques and their applications. In: *IEEE Sensing and Control, International Conference on Networking*. Okayama, Japão, 2009. p. 31-36.



O Congresso Brasileiro de Automática (CBA) é o maior evento científico na área de controle e automação do Brasil e acontece a cada dois anos, com a organização da **Sociedade Brasileira de Automática (SBA)**. O evento reúne participantes da comunidade científica e também da indústria, possibilitando a troca de informações entre os diversos profissionais da área de engenharia de controle e automação e de engenharia elétrica. No evento realizam-se minicursos, apresentações plenárias e de trabalhos técnicos, e mesas redondas.

O CBA 2022 aconteceu nos dias 16 a 19 de Outubro de 2022 em Fortaleza-CE, com organização local do Departamento de Engenharia Elétrica do Centro Tecnologia da Universidade Federal do Ceará (UFC) e com apoio do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE).

Neste ano a SBA e os organizadores locais promoveram um evento satélite ao CBA

2022, o Workshop “SBA na Terra da Luz” que aconteceu nos dias 20 e 21 de Outubro de 2022 em Canoa Quebrada-CE. Neste segundo evento teve a apresentação de 06 palestras em temas especialmente selecionados para a comunidade da SBA.

O PosAutomação teve expressiva participação no CBA 2022 e no Workshop “SBA na Terra da Luz”. Destes dois eventos participaram os discentes João Pedro Brunoni, Gabriel Thaler, João Paulo Zomer Machado, Diogo Ortiz Machado, Letícia Capistrano Favero, Rafael Sartori e Patrícia Mônica Campos Mayer Vicente, e os docentes Públio Macedo Lima, Eugênio de Bona Castelan Neto e Julio Elias Normey-Rico. A figura 1 retrata parte da equipe do PosAutomação que esteve presente ao CBA 2022.

Os seguintes trabalhos se apresentaram no CBA, em ordem alfabética por apresentador:



Fig. 1: Parte da equipe do PosAutomação no CBA 2022.

- Diogo Machado - *Fresnel Solar Collector: distributed parameters model identification*
- Eugênio de Bona Castelan Neto - *Event-triggered Control for Discrete-time Saturated LPV Systems using a Partially-dependent Dynamic Output Controller*
- Gabriel Thaler - *Efeito da seleção de métricas de desempenho em modelos de aprendizado supervisionado para detecção de amaciamento em compressores*
- Jaqueline Vargas - *Invariância positiva no controle sob restrições de sistemas lineares de segunda ordem*
- João Paulo Zomer Machado - *Aplicação de sistema de aquecimento de rolamentos para redução de variabilidade de ensaios em motores de corrente contínua sem escovas*
- João Pedro Brunoni - *Implementação de GPC de cômputo rápido com restrições baseada em microcontrolador*
- Julio Elias Normey-Rico - *Introdução ao Controle de Processos: Ensinando controle com matemática básica*
- Leticia Capistrano Favero - *Modelagem Dinâmica e Controle de Sistemas de Geração de Vapor Baseado em Energia Solar Térmica para Recuperação de Campos de Petróleo*
- Patrícia Mônica Campos Mayer Vicente - *Diagnóstico Síncrono Descentralizado com Coordenação*
- Públio Macedo Lima - *Uso da cifra de fluxo ChaCha20 em redes de automação*
- Rafael Sartori - *Estudo comparativo: EMPC e LP+DMC - aplicado em uma Unidade de Processamento de Gás Natural*

Além dos trabalhos apresentados no congresso, o Prof. Julio Elias Normey-Rico ministrou o mini curso: *Ensinando controle com matemática básica: Disciplina de Introdução ao*

Controle de Processos e apresentou a palestra: *Controle Preditivo com Modelos Lineares a Parâmetros Variáveis: Teoria e Aplicações*, como convidado para o Workshop “SBA na Terra da Luz”.

Mas o grande destaque da participação do PosAutomação no evento foi o prêmio recebido pela discente Patrícia Mônica Campos Mayer Vicente, que obteve a menção honrosa na categoria mestrado pelo paper apresentado, de título: *Diagnóstico Síncrono Descentralizado com Coordenação* (figura 2). Parabéns a Patrícia e ao seu orientador, o Prof. Felipe Gomes Cabral, pelo reconhecimento!

Outros artigos com co-autoria de membros do PosAutomação também foram apresentados pelos co-autores dos trabalhos. A lista completa pode ser acessada em: <https://easychair.org/smart-program/CBA2022/index.html>



Fig. 2: Patrícia Mayer Vicente recebendo a menção honrosa por seu paper na categoria de mestrado.

Time PosAutomação

O PosAutomação conta com um time de profissionais de excelência com conhecimento em inúmeras áreas que colaboram com um programa de referência internacional

Nesta edição, conheça os Professores **Ricardo Rabelo e Rômulo de Oliveira.**

Período antes da UFSC

Nascido em São Paulo (SP) em setembro de 1963, é filho de Valmor Rabelo e Rosa Lima Rabelo, ambos nascidos no sul de SC (Meleiro e Criciúma). Por parte de pai, a ascendência é portuguesa e italiana; de mãe, italiana e polonesa. Meu pai, a partir de cursos do Senai feitos em Siderópolis (SC) ao longo da década de 50, viu uma oportunidade de emprego na Varig num anúncio de jornal como mecânico de motor de aviões para a oficina geral em Porto Alegre. Conseguiu a vaga, casaram-se em 1959 e mudaram-se para Porto Alegre, onde tiveram o primeiro filho. Posteriormente, com o crescimento da empresa, meu pai foi transferido em 1962 para São Paulo (aeroporto de Congonhas, que era o único na época em São Paulo) para ser o chefe do setor de manutenção de motores. Lá me tiveram, como o segundo e último filho. Minha mãe, com apenas o ensino fundamental completo, sempre foi Do Lar e tinha como missão a usual na época, de criar os filhos enquanto o esposo trabalhava e sustentava a família. Paralelamente ao trabalho de oito horas diárias e com gosto por construção, meu pai fazia faculdade de engenharia civil à noite, na Fundação Casper Líbero, e lá se formou. Como “tinha poucas coisas para fazer”, meu pai ainda tinha uma oficina de automóveis ao lado de casa ... e possivelmente daí veio parte também pelo meu gosto por automóveis e automobilismo. Dentre fatos marcantes da época nacional, ainda tenho lembranças do pouso na Lua em 69, de irmos ver o desfile dos jogadores da Seleção do Brasil da Copa de 70 em carros de bombeiros, e de viaturas do “DOPS” pelas ruas, que na época não compreendia por que havia tantas.

Após décadas na empresa, meu pai aposentou-se e fomos morar em Criciúma em 1975, onde tínhamos muitos familiares. Morávamos em São Paulo exatamente onde posteriormente veio a ser construído o Shopping Center Ibirapuera, no bairro de Moema. Tratava-se de uma enorme área residencial, uma vila de casas basicamente de trabalhadores de algumas empresas da região. Também havia lá a “sede social” da Varig, que era composta por barracos de madeira, muitas árvores e um campo de terra de futebol. Boa parte da minha infância foi passada nesse espaço. Até hoje ainda mantenho contato com alguns dos amigos daquela época.

Em Criciúma, estudei no Colégio Marista até 1979; portanto, durante toda a adolescência (e ainda preservo várias amizades desta época). Neste período participei de várias Feiras de Ciências, sendo o feito maior a conquista do prêmio nacional e depois sul-americano de ciências, em 1978, com um protótipo de uma usina de aproveitamento de lixo (com mais quatro colegas). Na época, as feiras de ciências (de cada escola) e os campe-



Fig. 1: Prêmio nacional de ciências com o protótipo de uma usina de aproveitamento de lixo. 1978.

onatos de esporte interescolas eram “os” eventos da cidade (além dos muitos bailes de Carnaval de salão e “discotecas”, que eram muito comuns e “instituições”!). Tornei-me torcedor do time de futebol Comerciarío, que muitos anos depois tornou-se o time atual Criciúma. Apaixonado por esportes e também torcedor do Palmeiras (do tempo que morei em São Paulo), participou de competições na cidade e depois em SC, de xadrez, natação (a salientar que não existiam piscinas térmicas na época ...) e futebol de salão. Fui inclusive campeão estadual infantil duas vezes de futebol de salão.

Período de graduação na UFSC

Com meu irmão já fazendo engenharia elétrica na UFSC (foi inclusive aluno, por exemplo, dos professores Augusto Bruciapaglia e Jean-Marie Farines, quando ainda não havia o DAS), mudamos para Florianópolis em 1980 para eu fazer o 3. ano do Ensino Médio (Colégio Barddal, que ficava bem no centro da cidade), fazendo o vestibular em dezembro de

1980. Morávamos na Trindade, bem perto da UFSC. Aprovado, entrei na UFSC em março de 1981, no curso de Ciências da Computação, com 17 anos. Até hoje não sei bem ao certo o porquê desta escolha, dado que “ninguém” sequer sabia bem ao certo o que era essa tal coisa de “computador”. Talvez porque sempre adorei coisas de ficção científica e, certamente por influência do meu pai, sou até hoje também apaixonado por aviação. Na realidade, dado que meu pai construía prédios e casas (pela formação de engenharia civil) depois também em Florianópolis, eu inicialmente pretendia também fazer civil (até porque ajudava meu pai nos projetos de plantas baixas e fachadas ... sempre tive jeito para desenho). Mas, por conta de um estágio (aos 16 anos) que fiz num outro escritório, vi que não era isso que eu queria. E só mesmo no dia da matrícula para o vestibular decidi optar mesmo por Computação, pois estava em dúvidas em relação ao curso de Arquitetura (que me inscrevi como segunda opção).

Ainda sob os resquícios finais da dita-



Fig. 2: Campeão estadual infantil de futebol de salão.

dura militar (governos Geisel e Figueiredo), era leitor assíduo do Pasquim e tinha até carteirinha do cinema do CIC, onde só passavam filmes “alternativos”. Complementarmente, havia muitas atividades culturais na UFSC, de vários tipos ... numa época que não havia celulares, Internet e TV à cabo. Sem dúvida alguma que essa exposição à cultura foi fundamental para meu grande gosto pelas Artes e Música em geral.

Na terceira fase do curso de Computação já comecei a procurar estágios, e foi onde entrei no Grucon, um grupo de pesquisa de automação industrial da Engenharia Mecânica (onde é o atual Bloco B). Na época, o Grucon e o Labmetro eram os berços da automação industrial do Brasil, semeados via projetos de cooperação com a Alemanha (Bremen e Aachen) estabelecidos pelo “visionário, crucial e majestoso” Prof. Caspar Erich Stemmer (pai do Prof. Marcelo Stemmer). Fui orientado pelo saudoso e já falecido Prof. Áureo Ferreira, que coordenava o Grucon. Do lado do Labmetro, com quem dividíamos paredes, o coordenador era o Prof. Carlos Schneider (que posteriormente viria a criar a CERTI), tendo como braço direito o Prof. Carlos Flesch (pai do Prof. Rodolfo Flesch).

Pelo fato de fazer computação mas não entender de automação, fiz inúmeras disciplinas da engenharia mecânica, sendo uma delas ministrada pelo próprio Prof. Caspar Stemmer. Ele dava tão bem as aulas e com tanta didática que até hoje (!) lembro de algumas aulas dele. Com o detalhe que, na época, os retroprojetores estavam apenas começando a chegar. A maioria das aulas era na base do quadro & giz. Nesta época (1982-85) chegavam dos seus doutorados no exterior vários professores que depois viriam a ministrar aulas para o curso de Eng. Controle e Automa-

ção, como Abelardo Queiroz, Lourival Boehs, Walter Weingaertner, João Carlos Espíndola e, um pouco mais tarde, o Carlos Ahrens.

As programações de computador eram quase todas em cima do mainframe IBM 4341 (que tinha recém-chegado, substituindo o IBM 360 com seus “gigantes” 256K de RAM), com “enormes” 4GB de RAM e inúmeras unidades de disco rígido (“panelas” de 20MB) e unidades de fita (de 600 a 800MB). Inicialmente programados ainda com cartões perfurados, gradualmente a UFSC foi recebendo “terminais” para digitação de programas (principalmente na linguagem Fortran). A microinformática só viria a chegar vários anos depois, sob a chamada “reserva de mercado de informática”. Ainda lembro como se fosse hoje a emoção de ver pela primeira vez um programa meu aparecendo numa “tela” de computador, na 3ª. fase do curso, às 3 da manhã, no ‘NPD’ (atualmente Setic), pois os terminais eram liberados apenas para os alunos de computação e da 3ª. fase em diante (as horas eram agendadas por prioridade de avanço de fase do curso), e entre as 22h e 6h da manhã.

Porém, o meu primeiro uso de computador foi em cima de um computador analógico na engenharia mecânica, onde as instruções (para fins de automação) eram na base de instrução por instrução criadas baseado em Mapas de Karnaugh. Dado ao conhecimento adquirido, foi um passo para eu me tornar monitor da disciplina de programação Assembly (da computação) para o microprocessador 8080, por um ano, sob a supervisão do Prof. João Bosco (à época, o hoje INE – Departamento de Informática e Estatística – se chamada CEC: Departamento de Ciências Estatísticas e da Computação).

No Grucon, desenvolvíamos dezenas de softwares de automação para empresas, principalmente do Estado de São Paulo, como Mercedes-Benz, Ford, Volkswagen, Mecânica Pesada, Nuclebrás, Mannesmann, entre outras. Isso implicava em várias viagens, muitas vezes de ônibus, durante a noite. Fiquei no Grucon de março de 1982 à meados de 1986. E todo final do ano fazíamos uma grande festa de confraternização na ilha do Prof. Caspar Stemmer, na região da Tapera (bem perto do aeroporto de Florianópolis). O Grucon tinha umas 50 pessoas, distribuídas em alguns subgrupos, compostos por alunos estagiários de graduação e pós-graduação de engenharia mecânica e computação bem como de alguns alunos formados.

Nesta época ainda, o Grucon organizava anualmente o então maior e mais importante congresso nacional de automação industrial do Brasil, sempre realizado em São Paulo, no Hotel Hilton: o Congresso da Sobracon

(Sociedade Brasileira de Comando Numérico). Todo ano eu participava do evento. E em 1989 escrevi e apresentei meu primeiro no congresso, intitulado “Inteligência Artificial e suas aplicações na Manufatura”, onde o principal foco eram os sistemas especialistas, fruto de um sistema que desenvolvi para a Embraer para projeto de engrenagens mecânicas. Nesse período, o CIM (*Computer Integrated Manufacturing*) era o grande paradigma de automação empresarial/industrial, representando de certa forma o que posteriormente se convencionaria chamar de “Indústria 3.0”. No Grucon, já começávamos a receber as primeiras “estações CAD/CAM”, um salto enorme tecnológico (à despeito do rudimentar equipamento e do preço absurdo que era o software e principalmente o hardware), que permitia com que se desenhasse peças e já se gerasse o programa de comando numérico através de um complexo programa tradutor chamado de “pós-processador”.



Fig. 3: Festa do Grucon na ilha do Prof. Caspar Stemmer.



Fig. 4: Sistema especialista desenvolvido para a Embraer para projeto de engrenagens mecânicas.

Um dos trabalhos que mais me trouxe orgulho foi o desenvolvimento de um software especial para a indústria Mecânica Pesada, uma empresa francesa no Brasil que tinha por responsabilidade a fabricação das pás das turbinas de Itaipu (que ainda estava sendo construída). Eram máquinas gigantescas de usinagem, sendo uma delas um torno vertical de 20 metros (!) de altura, que era o principal recurso para o tipo de usinagem requerido

(movimento em spline das ferramentas de usinagem). Devido à reserva de mercado de informática, tive de desenvolver “do zero” alguns softwares especiais para esse torno e um pós-processador CAD/CAM, que então permitiram que a máquina efetivamente pudesse realizar tais complexas usinagens.

Mesmo estagiando e depois já trabalhando como consultor, me formei em final de 1984, com 21 anos de idade, com 6 meses de

antecedência. Uma loucura ... :-). Na época, todas as formaturas eram feitas no Clube Doze de Agosto, no centro da cidade, em cerimônias imensas (e o posterior e hiper tradicional baile no mesmo Clube), que reuniam todas as turmas do CTC do semestre. Depois de formado, até comecei a fazer o mestrado na Engenharia Mecânica. Mas, não gostando, desisti depois de ter cursado algumas disciplinas.

Período pós-UFSC

Em março de 1986 recebi proposta da Mercedes-Benz, de São Bernardo do Campo, para trabalhar lá e ajudar a consolidar o time de automação industrial que estavam criando por conta das demandas de automação da matriz alemã para com as fábricas de caminhões e ônibus (esta, em Campinas). Aceitei, e em julho de 86 comecei lá. Ajudava o time em inúmeros tipos de desenvolvimentos e integrações de sistemas e máquinas, ainda sob o “fardo” da reserva de mercado, o que nos abrigava até a nós mesmos construirmos cabos de comunicação para interfaceamentos

com CLPs antigos, entre outros. Fui duas vezes no período para Stuttgart, na Alemanha, para pegar sistemas de lá e trazer e adaptar para as duas fábricas daqui. Apenas sabia o “basicão” de alemão (quando era graduando na UFSC eu fiz dois anos de extracurricular), e era “um parto” entender todas as documentações dos sistemas, todas em alemão.

Nessa época, já começava a existir os princípios de rede de computadores e do que mais tarde seria a Internet. Como a empresa usava tudo da IBM, a própria IBM tinha criado a *Bitnet*, permitindo que as pessoas das diferentes fábricas do mundo pudessem trocar informações, de uma forma que mais tarde passou-se a chamar de *e-mail*.

Um ano depois, com 24 anos de idade, fui promovido a “Chefe de Divisão” de automação, coordenando uma das grandes equipes de automação industrial da empresa.

Tudo corria normalmente bem até que, em meados de 1988, todavia, recebi um telefonema do meu ex-chefe do Grucon (Prof. Áu-



Fig. 5: Placa com o nome da empresa Mecânica Pesada S.A. na Itaipu Binacional.



Fig. 6: Colação de grau da graduação. 1984.

reio Ferreira) me perguntando se eu não queria passar 1 ano em Portugal. Seria algo no âmbito de um programa ibero-latinoamericano financiado pela Espanha (CYTED-D), num grupo de pesquisa de referência europeu em Inteligência Artificial para Manufatura (um dos criadores da linguagem *Prolog* trabalhava lá, por exemplo), que trabalhava fortemente em projetos europeus (na época, o Programa ESPRIT). Mas isso implicava em eu ter que sair da Mercedes...

Demorei 6 meses (!) para tomar a decisão, também considerando que eu tinha uma namorada super firme ... Mas depois de muito refletir, resolvi pedir demissão. O problema é que a Mercedes não quis me liberar e me propôs várias coisas. Mesmo assim, não aceitei. A empresa então me liberou, mas desde que eu mantivesse o meu vínculo empregatício (embora sem remuneração), de forma que depois de um ano em Portugal eu poderia voltar e o meu lugar estaria garantido na volta. Assim eu aceitei ... :-).

Período pós-Mercedes-Benz

Chegando em Portugal em janeiro de 1989, completamente sozinho e sem conhecer uma sequer pessoa, comecei meus estudos e trabalhos na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Um verdadeiro choque! Não apenas cultural, mas também tecnológico ... afinal, Portugal não tinha vivido sob o atraso gerado pela reserva de mercado de informática. Eram microcomputadores em rede para todos os lados (PCs e Apples), linguagens de programação que não tinha experiência alguma (como C e Pascal), várias ferramentas para desenvolvimento de sistemas para IA (além das linguagens Lisp e Prolog), etc. Me sentia totalmente despreparado, apesar de toda minha experiência anterior. Foi um período inicial extremamente difícil. Mas aos poucos fui engrenando, vários colegas me ajudaram, e a coisa depois fluiu bastante bem.

Comecei a participar de várias reuniões de projetos de pesquisa, principalmente de

um sobre programação de células adaptativas de montagem com geração automática de *dashboards* gráficos de informação em tempo real (se me contassem isso hoje, eu chamaria isso de Indústria 4.0 !). Os estudos envolvidos que tive que fazer e relatórios associados acabaram por me habilitar a requerer a Especialização em Informática com ênfase em Inteligência Artificial.

Neste período, no verão europeu, fiz o famoso “mochilão de trem” pela Europa, onde também pude conhecer muitos países, cidades e muitas pessoas. Toda essa atmosfera cosmopolita, multicultural e histórico-arquitetônica me fascinava mais e mais a cada dia. E o namoro que tinha ficado “pendente” em São Paulo naturalmente “foi-se”. :-).

Eu gostava muito do meu trabalho na Mercedes-Benz e nunca tinha passado pela cabeça em continuar os estudos com mestrado, doutorado, etc. Até hoje eu adoro um chão de fábrica !! Porém, comecei a gostar muitíssimo daquele tipo de trabalho, de pesquisa de ponta mas aplicada, e minhas ideias mudaram. Em janeiro do ano seguinte, ao voltar para o Brasil, pedi demissão formal da Mercedes. Ainda retornei depois à Portugal por

mais 4 meses para terminar as coisas do projeto, e então retornei ao Grucon. O objetivo já era então fazer um mestrado no exterior, no mesmo grupo. Enquanto retomava as consultorias e repassava o conhecimento adquirido a um grupo de estagiários, já ia preparando o plano de trabalho e “papitada” para sair. Tinha tido já a oportunidade de conhecer outros dois polos de IA europeus (um consolidado, em Edimburgo/Escócia, e um emergente, em Kaiserslautern/Alemanha), mas o clima nessas cidades era de “afugentar”. Então optei por tentar ir para o mesmo grupo de Portugal.

Nesta época, já com várias publicações e uma boa experiência industrial, meu chefe no Grucon sugeriu-me que eu me candidatasse direto para um doutorado. Embora pouco usual na época, segui o conselho e, para minha surpresa, fui contemplado com bolsa para doutorado pleno, tanto pela Capes, como pelo CNPq.

Período do Doutorado

Seis longos anos ... :-). Diferentemente do que mais comumente ocorre no Brasil, na Europa os doutorados em grandes grupos são desenvolvidos a partir de resultados de projetos de



Fig. 7: Divisão de automação da Mercedes-Benz.



Fig. 8: Mochilão de trem pela Europa.

pesquisa, envolvendo várias outras instituições europeias. Trabalha-se para vários projetos em simultâneo (e inúmeras viagens, laboratórios técnicos, implementações, escrita de novos projetos e artigos, etc.) e gradualmente vai-se definindo o que será a tese a ser tirada disso. Isso faz com que o doutorado normalmente se alongue bastante. A minha bolsa do CNPq terminou (pois o limite era de 48 meses) e o período restante fui pago com recursos dos projetos e algumas aulas de programação e IA em alguns cursos.

Nessa época eu já tinha noivado formalmente com uma portuguesa e os planos eram de vivermos no Brasil após meu doutorado. Mas terminamos o relacionamento quatro anos depois.

Em novembro de 1997 defendi o doutorado em Engenharia Elétrica com especialidade em *Robótica e Sistemas Integrados de Manufatura*. Envolveu o desenvolvimento de um framework e um sistema computacional para *scheduling* dinâmico baseado em negociação multiagente para células virtuais de manufatura composta por máquinas autônomas. A

implementação foi “real”, no laboratório de robótica do grupo, que tinha uma célula (com robôs industriais ABB) como que o dobro do tamanho do laboratório LAI do DAS. Interessante perceber que, passados 25 anos, isso é ainda “estado da arte” da Indústria 4.0, e um dos artigos meus sobre isso ainda continua sendo bastante citado.

Apesar de ter sido um período muito estressante (como é o de qualquer doutorado), fiz uma enorme quantidade de amigos dentro e fora da universidade, vários dos quais mantenho contato até hoje.

Professor da UFSC

Voltando ao Brasil, trouxe “debaixo do braço” dois projetos europeus de duração de 3 anos cada, parcialmente financiados pelo CNPq, com uma bolsa DTI-A, na qualidade de coordenador técnico. Isso implicava em muitas viagens por ano para a Europa para as reuniões dos projetos.

Como forma de melhor organizar o trabalho até então voltado para projetos europeus e algumas assessorias a empresas e organizações

nacionais, criei (junto com dois outros recém-doutores colegas em situação equivalente a minha) o grupo de pesquisas GSIGMA (Grupo de Pesquisas em Sistemas Inteligentes de Manufatura e Redes Colaborativas) dentro da Engenharia Mecânica da UFSC, ligado ao Grucon.

Com vistas a entrar como professor da UFSC, surgiu um concurso para o Departamento de Automação e Sistemas (DAS) em 1999, na área de Sistemas Integrados de Manufatura, extremamente alinhado à minha área de atuação. Fiz e fui aprovado em primeiro lugar, que tinha como membros da banca os professores Jean-Marie, José Cury e Marcelo Stemmer. Porém, o governo FHC congelou as contratações e só assumi em janeiro de 2000, com 37 anos de idade, juntamente com os professores Werner e Rômulo, que estavam nos dois outros concursos. No início, ainda recebi algumas propostas para retornar à iniciativa privada, mas já tinha decidido ser mesmo professor e ao mesmo tempo poder “atender” a iniciativa privada.

Na vida pessoal, comecei a namorar a Maria Inês Meurer em 1998, com quem me casei em 2001. Ela tornou-se também profes-

sora da UFSC (Depto Patologia) em 2004, depois de ter concluído seu doutorado em 2002 na PUC/RS. Divorciada e com dois filhos bem pequenos na época (Lucas e Luisa, hoje músico profissional e professor, e jornalista, com 31 e 29 anos, respectivamente), tivemos o Artur em 2003 e que atualmente é aluno da 2ª. fase do curso de engenharia de controle e automação da UFSC de Florianópolis. “Em breve” terei que dar aula para ele. :-)

Como professor, atuei desde sempre no ensino, pesquisa, extensão e administração. Progridi regularmente na carreira a cada dois anos, e em 2016 tornei-me professor titular.

Dei várias disciplinas na graduação do curso de Engenharia de Controle e Automação (ECA), sendo duas delas desde há muitos anos: a de Avaliação de Desempenho de Sistemas e a de Integração de Sistemas Corporativos. Com a aposentadoria do Prof. Augusto Bruciapaglia em 2012, assumi as disciplinas de Estágio, PFC e Intercâmbio até 2018. Com o assumir a coordenação do projeto institucional Capes-PrInt no final de 2018, passei a ser subcoordenador dessas três disciplinas.

Desde minha entrada na UFSC, fui ho-



Fig. 9: Defesa do doutorado. 1997.



Fig. 10: Grupo de pesquisa GSIGMA.

menageado 25 vezes, como professor homenageado, paraninfo, patrono ou nome de turma. Orientei dezenas de estágios e PFCs.

Na pós-graduação (iniciando ainda quando a PosAutomação não existia e pertencíamos ao Programa de Pós-Graduação em Engenharia Elétrica), dentre outras disciplinas pontuais, ministrei por vários anos a disciplina de Sistemas Multiagente; inicialmente dividindo-a com o (falecido) Prof. Guilherme Bitencourt, e posteriormente com o Prof. Jomi Hübner (que o substituiu). Deixei de dar esta disciplina em 2015. Atualmente ministro duas disciplinas de Indústria 4.0 (sendo que uma delas é mais voltada às questões de inovação e empreendedorismo) e a de Engenharia de Software Orientada a Serviços. Orientei até então 24 dissertações de mestrado, 11 de doutorado e 3 de pós-doutorado, vários deles em estreita relação com empresas e indústrias. Atualmente oriento 3 alunos de mestrado e 3 de doutorado. Publiquei mais de 200 artigos científicos e estou entre os cem professores com trabalhos mais citados da UFSC.

Ainda na pesquisa, atuei como coordenador local de vários projetos europeus e com empresas, que movimentaram alguns milhões de reais ao longo dos dez anos que estive à frente desse tipo de iniciativa.

Por duas vezes estive à frente da organização da conferência internacional PRO-VE (IFIP Working Conference on Virtual Enterprises) no Brasil, a mais relevante na área do mundo: PRO-VE'2000, realizada em 2000, em Florianópolis, e PRO-VE'2011, realizada na cidade de São Paulo.

Sou membro formal de sociedades e grupos de trabalho científicos, a saber: IFIP 5.5 (Collaborative Networks and Virtual Enterprises), IFIP 5.7 (Advanced Production Ma-

nagement Systems), IFIP 5.12 (Architectures for Enterprise Integration); IFAC 5.3 (Enterprise Architecture), IEEE Industrial Agents, e Socolnet (Society of Collaborative Networks). Fui e ainda sou membro de comitês científicos de várias conferências nacionais e internacionais assim como revisor de uma série de periódicos.

Fui bolsista PQ do CNPq por 9 anos, deixando de tê-la após o pós-doutorado. Este, realizado na Griffith University, Brisbane, Austrália, ocorreu entre agosto de 2014 e julho de 2015, com pesquisas na área de Inovação Colaborativa em Redes de Empresas. Tive como supervisor o Prof. Peter Bernus, uma das maiores autoridades mundiais na área de Enterprise Architecture, e cientista-chefe que liderou o processo de criação da respectiva norma ISO. Neste período, pude estudar os principais ecossistemas de inovação do mundo e da Austrália, e propus um modelo de inovação colaborativa, tanto em redes de empresas, como em ambientes de startups.

Através de projetos que escrevi, o DAS recebeu - via editais de fomento - a planta didático-industrial da FESTO que hoje equipa o LAI (em 2012) e em 2018 (mas com recursos recebidos apenas neste ano) uma extensão dessa planta, com o que existe de mais moderno no mundo.

Coordeno atualmente o projeto institucional PrInt Automação 4.0, envolvendo mais de 30 professores do DAS, EEL e INE. É o representante do CTC no Comitê de Inovação da UFSC bem como da ACATE, dentro da vertical 'Manufatura 4.0', composta por mais de cem empresas provedoras de soluções de automação, IA, bigdata e IoT. Ainda com relação à Indústria 4.0, foi selecionado nacionalmente pelo Ministério da Economia e MCTI

em 2019 para elaborar o texto-base da Estratégia Nacional da Indústria 4.0, cujo objetivo final era o de servir de referência para o estabelecimento de políticas públicas para a área. Neste momento, articula com a Fiesc, governo do SC e ACATE a elaboração de uma estratégia estadual de Indústria 4.0, paradigma este que já caminha rapidamente para o da Indústria 5.0.

Fui representante do CTC na Câmara de Extensão da UFSC e por quatro anos representante no Conselho Universitário. Participei de dezenas de comissões, colegiados e bancas do curso de ECA, do DAS, da PosAutomação, do CTC e da UFSC para os mais diversos assuntos no contexto de ensino, pesquisa, extensão e administração. Fui supervisor de laboratórios do DAS por vários anos e Chefe de Departamento no período 2010-2012. Dei inúmeros cursos de formação e palestras para empresas. E, com o objetivo de contribuir para o CTC em termos de mudanças que considerava bastante necessárias, candidatei-me numa chapa à vice direção do CTC em 2021. Porém, perdi a eleição para a chapa da situação.

Atualmente estou envolvido na submissão de duas grandes propostas de projetos que poderão trazer excelentes benefícios e recursos para o DAS/PosAutomação/ECA. Um deles é uma rede multi-universidades (UFSC, PUC-PR, UTFPR, USP e UFRGS) sobre "Manufatura do Futuro", um projeto de um edital do CNPq de 5 anos e com orçamento de 7 milhões de reais. O outro é a candidatura a sermos um Centro de Competência EMBRAPPII na área de Indústria 4.0, IoT e Segurança Cibernética, junto com os departamentos INE, EEL, EMC, EPS e EGC. Também sendo um projeto inicial de 5 anos, o projeto prevê um aporte de 60 milhões de reais no Centro, incluindo em infraestruturas e bolsas, com forte

conexão com as indústrias e o ecossistema de inovação.

Apesar de amar a profissão e o que faço, adoro também outras coisas e assuntos. A música sempre esteve presente no meu meio e família. Toco baixo (tenho até um baixolão acústico na minha sala do DAS, mas que infelizmente não consigo arranjar nem cinco minutos por dia para exercitar) e até já tive uma "banda de garagem", onde era puro rock'n roll (cresci escutando Led Zepellin, Genesis, Pink Floyd, Jethro Tull, dentre inúmeras outras bandas), embora adore jazz também. Ainda hoje tenho mais de mil CDs ... que ainda não sei o que farei com eles. :-). Sou um apreciador de vários tipos de bebidas e gastronomia (me "viro" na cozinha), de leitura (política, economia, tecnologia, história, filosofia, religiões, etc.), de artes, de design e arquitetura, natureza, de esportes, de amigos e de estar em família, e viajar. Pretendo intensificar fazer essas coisas após minha aposentadoria, que não deve demorar mais do que cinco anos a partir de agora.

Hoje sou muito grato pelo excelente ambiente de trabalho no DAS/UFSC e pelas excelentes amizades e parcerias construídas ao longo do tempo, e por tudo o que a UFSC me ofereceu. Tenho um gigantesco orgulho de ser da UFSC, do DAS, e professor do curso de Engenharia de Controle e Automação e da PosAutomação! ■

Rômulo Silva de Oliveira

Nasci em Porto Alegre, Rio Grande do Sul, em 1962, onde passei os primeiros anos da infância com meus pais Nauro e Zina e uma irmã mais nova. Meu pai trabalhava na Companhia Estadual de Silos e Armazéns (CESA), criada na década de 1960 para fornecer uma infraestrutura de armazenagem de grãos no estado. Várias unidades foram criadas em diferentes cidades. Cada unidade ocupava uma área na periferia da cidade, cercada, com segurança armada, e contava com silos, armazéns, escritório, balança para caminhões, terminal ferroviário e a residência do chefe e do subchefe da unidade.

Em 1965 fomos morar em Santa Bárbara do Sul, onde meu pai foi trabalhar como chefe da unidade. Era uma cidade do Planalto Gaúcho com cerca de 10 mil habitantes na época, região agrícola, acesso por estrada de chão. Morávamos dentro do pátio da unidade (Figura 1). O silo graneleiro de concreto tinha cerca de 9 andares de altura e o único elevador da cidade, o que o tornava uma atração turística. Era o prédio mais alto em um raio de 60Km. Na verdade aqui foi o meu primeiro contato com automação, pois o silo contava com um painel sinótico eletromecânico que ocupava uma parede inteira em uma sala do



Fig. 1: Vida em um armazém graneleiro (1968).

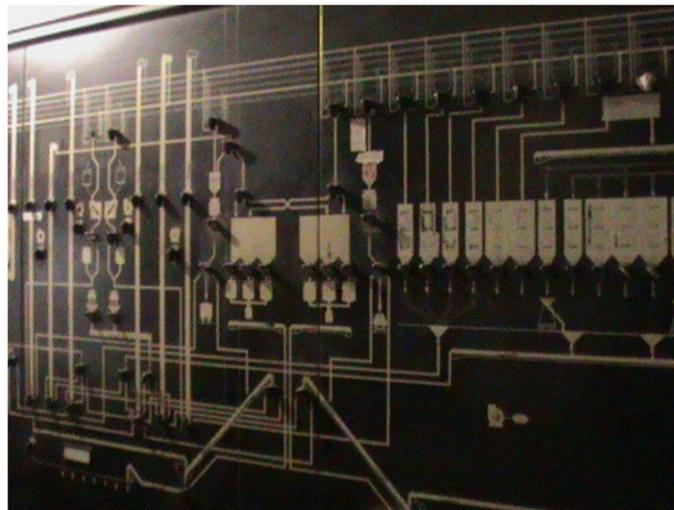


Fig. 2: Painel sinótico eletromecânico do silo graneleiro (1960s).

segundo andar (Figuras 2 e 3).

Fiz o primário (4 anos) em escola pública municipal e o ginásio (4 anos) em escola pública estadual. Embora houvessem tiroteios eventuais no único hotel da cidade e nas "zonas" das cercanias, como criança tinha-se enorme liberdade. Morávamos nos limites da cidade e, atrás do pátio da CESA, já era campo com gado solto.

Em 1975 nos mudamos para Carazinho, cidade também do Planalto Gaúcho, pois meu pai foi transferido para aquela unidade da CESA. Lá conheci a Simone que viria a ser minha esposa no futuro. Em 1978 meu pai foi novamente transferido, desta vez para a sede administrativa da empresa em Porto Alegre. Ali concluí o ensino de segundo grau e prestei vestibular. Na época a UFRGS aceitava 4 opções de curso na inscrição. A pontuação obtida pelo candidato era considerada por inteiro na primeira opção. Caso o candidato não fosse aprovado, ele disputava a segunda opção com a pontuação reduzida por um certo fator, e assim por diante. Minha primeira opção era Engenharia Eletrônica e quase fui aprovado. A segunda opção era Tecnólogo em Processa-

mento de Dados (não havia bacharelado na época) e quase fui aprovado. Fui aprovado na terceira opção Economia. A quarta opção era Física. Como fui aprovado em Engenharia Eletrônica na PUCRS, e minha família conseguia pagar o curso, fiz a graduação na PUCRS e acabei não cursando Economia na UFRGS.

Iniciei a Engenharia Eletrônica na PUCRS em março de 1979. Neste ano fiz meu primeiro programa de computador, em Fortran, usando cartões de papel. O deck de cartões perfurados com o programa (cada linha é um cartão perfurado) era entregue e 24 horas depois recebíamos uma listagem com o resultado da execução. A ênfase do curso foi em telecomunicações. Durante a graduação trabalhei como monitor em disciplinas de Teoria Eletromagnética e Laboratório de Física. A partir de 1982 tive oportunidade de fazer estágio na antiga RFFSA, na Fundação de Ciência e Tecnologia (CIENTEC, extinta em 2017) e na empresa Digitel (fabricante de modems fundada em

1978).

Na RFFSA tive oportunidade de fazer uma viagem para manutenção das linhas de comunicação que margeavam os trilhos da linha entre Porto Alegre e Santa Maria. A equipe da qual eu fazia parte viajava em um vagão dormitório que pegava carona nos comboios da linha, estacionava em algum desvio no meio do percurso, e trabalhava nos fios e retransmissores da região. O interessante é que o vagão dormitório usado na verdade era um vagão que Getúlio Vargas usou em viagens pelo interior do estado enquanto governador. É uma lástima pensar que aquele veículo não estivesse preservado em um museu.

Terminei o curso de Engenharia em dezembro de 1983 e, como acontece quase sempre, veio a dúvida sobre o que fazer a seguir. Desde criança eu tinha o sonho de ser cientista, algo que em uma pequena cidade agrícola nas décadas de 1960 e 1970 parecia maluco. Eu tinha uma proposta de emprego como

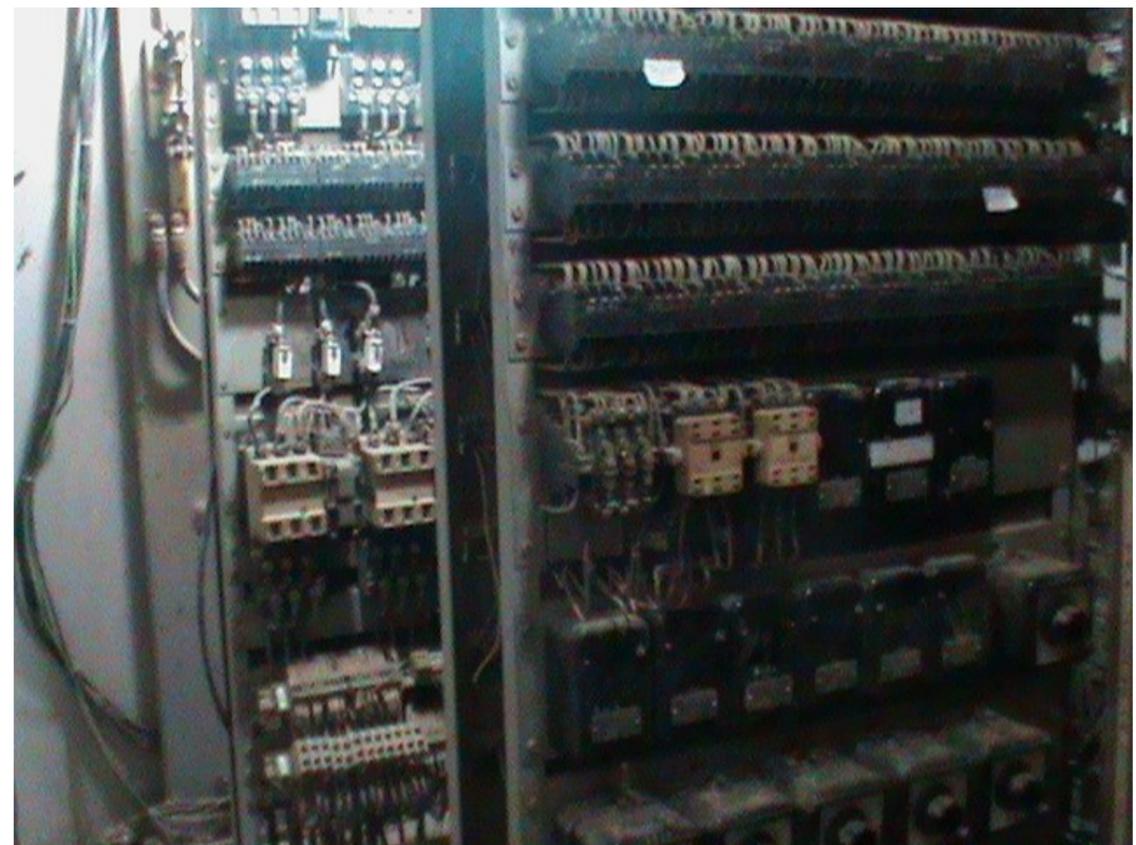


Fig. 3: Parte traseira do painel sinótico (1960s).



Fig. 4: LABO-B034 (Nixdorf 8870/1) (1986).

engenheiro de desenvolvimento na Digitel, mas fiz a inscrição no mestrado em Ciência da Computação da UFRGS e fui aceito. Também fiz a inscrição no mestrado em Engenharia Elétrica da UFSC, no LCMI, principalmente pelo critério de proximidade, e fui aceito também. É incrível pensar que poderia ter feito mestrado na UFSC orientado pelas pessoas que seriam meus colegas de departamento 16 anos depois. Seguindo o sonho antigo, entrei para o mestrado na UFRGS.

Levei 3 anos para concluir o mestrado, o que era normal na computação da UFRGS naquela época. Fui orientado pelo Prof. Thadeu Botteri Corso na área de Sistemas Operacionais. A dissertação, cujo título foi "SOMBRA: Um Sistema Operacional Multiusuário", descrevia a especificação e implementação de um sistema operacional multiprogramado, multiusuário, construído usando Pascal Concorrente em um minicomputador LABO-B034 (Nixdorf 8870/1, Figura 4). Durante o mestrado também trabalhei como assistente de pesquisa da Profa. Liane Tarouco na área de redes de computadores.

Durante o período do mestrado muitas coisas aconteceram. Em 1985 me casei com a Simone, estamos completando 37 anos de casamento. Também nasceu nosso filho Augusto, que no futuro viria a fazer mestrado no PosAutomação sob a orientação do Prof. Eduardo Camponogara. Ainda durante o mestrado

iniciei a carreira de professor universitário no Instituto de Informática da PUCRS. Naquela época a PUCRS aceitava como professor horista quem já tivesse concluído os créditos no mestrado, sem tê-lo ainda terminado, o que era o meu caso.

Tenho muito carinho pelas memórias da minha passagem como professor dos cursos de Informática e de Análise de Sistemas da PUCRS. Havia muito companheirismo e o ambiente de trabalho era muito bom. Quando comecei tinha 23 anos de idade, vários alunos eram mais velhos do que eu, mas consegui ter um bom desempenho. Acabei sendo 7 vezes professor homenageado e uma vez paraninfo naqueles 5 anos como professor entre 1985 e 1989.

Com o final do mestrado veio novamente um momento de decisão. Entre 1987 e 1989 trabalhei como professor horista na PU-



Fig. 5: Buscando ideias para o doutorado na praia (1992).



Fig. 6: Filho tirando neve do carro em Urbana-IL (1995).

CRS enquanto também atuava como desenvolvedor "freelancer" de software para equipamentos (embedded systems) para algumas empresas da grande Porto Alegre. Também nesta época estava crescendo a demanda por programadores na linguagem C e poucas pessoas conheciam a linguagem, de maneira que em 1988 a minha principal fonte de renda foi ministrar cursos de C. Após este curto período de experiência trabalhando com a iniciativa privada decidi que queria mesmo era ser professor universitário.

Em 1989 surgiu a oportunidade. O Instituto de Informática da UFRGS (criado naquele mesmo ano) abriu 11 vagas para professores, e aceitava pessoas apenas com o mestrado. É preciso lembrar que nos anos 1980 a computação era ainda uma área nova, e quase nenhum professor da computação tinha real-

mente formação em computação. Prestei concurso para duas áreas e fiquei em primeiro lugar nas duas, optei pela vaga na área de Sistemas Operacionais, minha área favorita. Entre dezembro de 1989 e agosto de 1992 ministrei disciplinas de sistemas operacionais e



Fig. 7: Natal em Urbana-IL (1995).

afins no curso de Bacharelado em Ciência da Computação, criado em 1983 no lugar do Tecnólogo em Processamento de Dados.

Em outubro de 1992 iniciei o doutorado na Engenharia Elétrica da UFSC, no antigo Laboratório de Controle e Microinformática (LCMI), sob orientação do Prof. Joni da Silva Fraga. Sou muito grato ao Prof. Joni pela orientação e paciência ao longo dos quatro anos e meio do curso. Quando cheguei no LCMI eu tinha uma forte bagagem de desenvolvedor. Minhas ideias para o doutorado foram prontamente escoraçadas no estilo Joni, o qual me conduziu para uma abordagem mais científica, mais perene e mais profunda, nos trabalhos do doutorado. Foi escolhido o tema Computação Imprecisa dentro da área de Sistemas de Tempo Real. Lembro do primeiro texto que escrevi e passei para o Joni fazer uma revisão, o qual ele cobriu completamente com os seus comentários. Fiquei indignado e preparei uma réplica para cada comentário dele. Porém, com o tempo, percebi que ele estava certo e

passamos a nos entender bem. Tenho enorme respeito pela seu conhecimento técnico e muito apreço pela suas qualidades como pessoa.

O doutorado é um momento muito rico para todos. Fiquei muito feliz em sair de Porto Alegre e mudar para Florianópolis, onde a qualidade de vida era muito melhor. Sempre que o doutorado travava em algum ponto, eu tinha a opção de caminhadas na praia para encontrar as soluções (figura 5), e jogos de videogame com o filho. Também fiz uma missão de estudos (doutorado sanduíche) na University of Illinois at Urbana-Champaign durante 6 meses, onde trabalhei com a Profa. Jane Liu. Ficamos lá durante o inverno de Illinois e vi muita neve, mas foi muito bom (figuras 6 e 7).

Com o final do doutorado em 1997, mais uma bifurcação. A qualidade de vida para mim e minha família era tão boa em Florianópolis que decidimos ficar. Porém, eu era professor concursado na UFRGS, e havia me



Fig. 9: Confraternização com orientados (2013).

afastado para doutoramento com vencimentos, uma demissão somente seria possível depois de permanência por igual período do afastamento, o que é correto. Uma solução seria a transferência para a UFSC. O Departamento de Informática e Estatística (INE) da UFSC concordava em trocar a minha vinda da UFRGS por uma vaga de professor, porém o Instituto de Informática da UFRGS não concordou. Sou muito grato ao Prof. Jovelino Falqueto, chefe do INE na época, que colocou todo o empenho possível, mas a troca não aconteceu.

Em 1999 o DAS abriu concurso para professor em várias áreas, inclusive sistemas de tempo real. Particpei e fiquei em primeiro lugar. A posse foi marcada para janeiro de 2000. Naquele momento eu já havia trabalha-

do de volta na UFRGS por 3,5 anos e, como a transferência seria para outra Instituição Federal, obtive a concordância de todos. Os anos de 1997, 1998 e 1999 foram muito difíceis para mim. Minha família ficou em Florianópolis e eu passava as semanas em Porto Alegre, indo e vindo de ônibus todo domingo e toda sexta-feira. Não tenho saudades desta época.

Com o ingresso no DAS tive oportunidade de ministrar disciplinas no curso de Engenharia de Controle e Automação e também na Pós-Graduação em Engenharia Elétrica (PPGEEL) da UFSC. Comecei a orientar mestrado no PPGEEL já em 2000 e a orientar doutorado em 2003. Foi um período muito produtivo. Fui coautor do livro Sistemas de Tempo Real, juntamente com os professores Jean-Marie Farines e Joni da Silva Fraga, que conti-



Fig. 8: Coordenando sessão do WTR (2008).



Fig. 10: Pós-doutorado em Portugal, Prof. Luis Almeida é o primeiro à esquerda (2020).

nua a ser referenciado até hoje em muitos cursos. Também fui coautor do livro *Sistemas Operacionais* pela editora Bookman, que vendeu cerca de 10 mil cópias ao longo de suas 4 edições. Tive participação importante no Workshop de Sistemas de Tempo Real (figura 8) e participei da criação do Workshop de Sistemas Operacionais (WSO) em 2004. A partir de 2003 passei a receber a bolsa de produtividade em pesquisa do CNPq.

Os anos seguintes representaram uma atuação mais intensa na gestão acadêmica. Atuei como sub-chefe do DAS entre 2006 e 2008 e como chefe do departamento entre 2008 e 2010. Durante meu período como chefe de departamento a reitoria da UFSC iniciou a discussão de uma proposta de transformar o curso de graduação em Engenharia de Con-

trole e Automação em uma especialidade (2 anos) do curso de Engenharia Elétrica. Todo o corpo docente do DAS trabalhou unido e fomos capazes de mostrar para a reitoria que o curso de Engenharia de Controle e Automação era substancialmente diferente do curso de Engenharia Elétrica e tal fusão não seria possível.

Em 2006 fiz parte da comissão que elaborou a proposta de criação do Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS, atual PosAutomação) na UFSC, o qual iniciou suas atividades em 2007, com o conceito 5 da Capes. Participei da comissão de implantação do PosAutomação entre 2007 e 2008, e depois do seu colegiado delegado entre 2009 e 2014. Fui coordenador do Programa de Pós-Graduação entre 2014 e 2016

e depois subcoordenador entre 2016 e 2018. No meu período como coordenador do PosAutomação consolidamos a descrição do Programa na plataforma Sucupira recém criada pela Capes. Também foram criadas várias comissões formadas por docentes e discentes, com o propósito de democratizar a gestão. O esforço e dedicação do pessoal docente, técnico-administrativo e discente ligados ao PosAutomação culminaram com a elevação do Programa para o conceito 6 da Capes em 2017.

Em minha carreira fui 25 vezes professor homenageado, paraninfo ou patrono das turmas de graduação para as quais lecionei. Ao longo dos anos tive a felicidade de concluir a orientação de 26 alunos de mestrado e 11 alunos de doutorado. Trabalhei com alunos brilhantes que muito me ensinaram (figura 9). Entre os doutores formados, seis deles são professores em Universidades Federais ou Institutos Federais. Entre dezembro de 2019 e junho de 2020 desfrutei de um período sabático na Faculdade de Engenharia da Universidade do Porto, Portugal, em colaboração com o Prof. Luis Almeida (figura 10). Infelizmente a pandemia de Covid-19 afetou os planos de participação em congressos e visitas a outras instituições, mas foi um período muito produtivo em termos científicos, dedicado a estudar a aplicação de métodos estatísticos aos sistemas de tempo real.

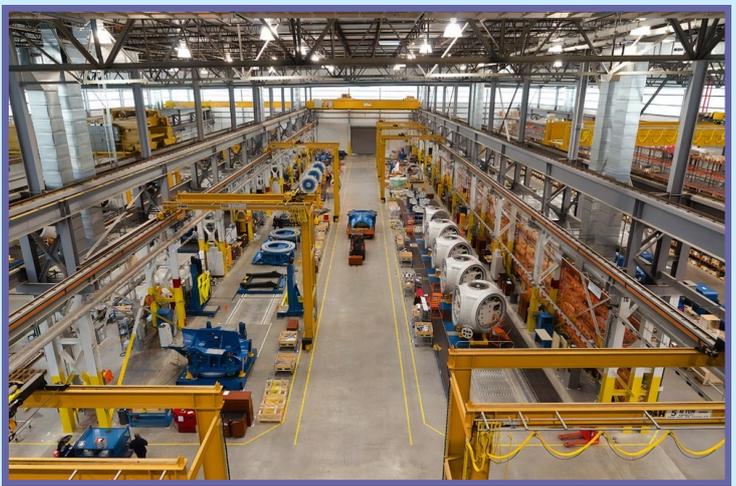
Minha atuação na área de Sistemas de Tempo Real, no contexto do PosAutomação, gerou cerca de 150 artigos em periódicos e congressos. Publiquei artigos nos dois principais veículos da área: o IEEE Real-Time Systems Symposium e o Real-Time Systems Journal. Sou atualmente editor associado do Real-Time Systems Journal. Recebo bolsa de produtividade em pesquisa do CNPq continua-



Fig. 11: Foto atual (2022).

mente desde 2003. Em 2018 publiquei o livro *Sistemas de Tempo Real*, com o material que tipicamente abordo nas disciplinas que ministro na UFSC. Aproveitei o período de ensino remoto, durante a pandemia, para atualizar o livro e publicar uma segunda edição. E também para gravar vídeos sobre as matérias que leciono na UFSC. Foram 125 vídeos totalizando cerca de 40 horas de vídeos publicados no youtube que permanecerão como um material auxiliar disponibilizado para todos.

Vir para Florianópolis, para a UFSC e para o DAS representou algo muito bom na minha vida. Passei aqui os melhores momentos da minha vida adulta. Agradeço aos professores Joni da Silva Fraga e Jean-Marie Farines pelo apoio e oportunidades que me deram nestes 30 anos, desde que comecei como doutorando no segundo semestre de 1992. Com a aposentadoria à vista chego a mais um ponto de bifurcação, e está na hora de iniciar novos projetos (figura 11). ■



Venha estudar **CONTROLE, AUTOMAÇÃO E SISTEMAS COMPUTACIONAIS** em um dos mais prestigiados programas de pós-graduação do Brasil! Nota 6 (de 7) na CAPES.

Cursos de mestrado e doutorado acadêmicos na melhor cidade do país:
Florianópolis!

Abertura de processos seletivos
2 vezes por ano.

<https://posautomacao.ufsc.br/>

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS (POSAUTOMAÇÃO) - UFSC

