DOI: 10.48011/asba.v2i1.1190

Navegação Visual Autônoma de um Veículo Terrestre em Escala Reduzida

Daniel Veloso Ribeiro, **Cairo Lúcio Nascimento Júnior**, **Wagner Chiepa Cunha**

Instituto Tecnológico de Aeronáutica
Divisão de Engenharia Eletrônica
Laboratório de Máquinas Inteligentes
São José dos Campos - SP
E-mails: dribeiro@ita.br, cairo@ita.br, chiepa@ita.br

Abstract: This article proposes and presents simulations of an autonomous visual navigation system for a small scale four-wheel differential drive vehicle. The navigation system is composed of: 1) a lane keeping controller that process the image of an onboard vehicle camera and corrects the vehicle orientation and speed, 2) a vehicle localization algorithm that uses the Kalman Filter algorithm to estimate the vehicle pose (position and orientation) using data from a RTK GNSS receiver, a digital compass and odometry sensors. Three possible solutions for the lane keeping controller are investigated and compared. The first solution uses a sliding window approach to process the onboard camera image and extract the distance from the center of the vehicle to the center of the lane and the lane curvature radius. Then a proportional controller is used to calculate the desired speed for the vehicle wheels. The second solution uses a deep convolutional neural network to imitate the first solution, that is, the neural network input is the camera image and the network is trained to generate the same output as the proportional controller used in the first solution. The third solution also employs a deep convolutional neural network but in this case the network is trained using the data recorded when a human drove the simulated vehicle through the same simulated environment. All three solutions are capable of maintaining the car within acceptable error from the intended trajectory.

Resumo: Este artigo propõe e apresenta simulações de um sistema de navegação visual autônoma para um veículo diferencial com quatro rodas em escala reduzida. O sistema de navegação é composto por: 1) um controlador de seguimento de faixa que processa a imagem de uma câmera embarcada no veículo para corrigir sua orientação e velocidade, 2) um algoritmo de localização do veículo que usa filtro de Kalman para estimar a pose do veículo (posição e orientação) usando dados de um receptor GNSS RTK, uma bússola digital e sensores de odometria. Foram investigadas e comparadas três possíveis soluções para o controlador de manutenção de faixa. A primeira solução usa uma abordagem com algoritmo de janelas deslizantes para processar a imagem da câmera embarcada, extrair a distância do veículo para o centro da faixa e o seu raio de curvatura. Um controlador proporcional é utilizado, então, para calcular a velocidade desejada para as rodas do veículo. A segunda solução usa uma rede neural convolucional profunda para imitar a primeira solução, isso é, a rede neural recebe a imagem da câmera como entrada e é treinada para gerar a mesma saída que o controlador proporcional usado na primeira solução. A terceira solução também aplica uma rede neural convolucional profunda, mas a rede é treinada usando dados gravados quando um motorista humano dirigiu o veículo no mesmo ambiente simulado. Todas as três soluções foram capazes de manter o veículo em uma margem de erro aceitável da trajetória desejada.

Keywords: Lane keeping; Autonomous vehicle; Visual navigation; Neural network; Kalman Filter; GNSS RTK .

Palavras-chaves: Seguimento de faixa; Veículo autônomo; Navegação visual; Rede neural; Filtro de Kalman; GNSS RTK.

1. INTRODUÇÃO

Com a popularização dos veículos autônomos nos últimos anos, solucionar o problema de manter o veículo entre faixas através de soluções automáticas têm se tornado mais relevante. Este problema é conhecido como *lane keeping* e é um tipo de seguimento de trajetória. Existem diversas formas de solucioná-lo, aplicando técnicas variadas como redes neurais ou algoritmos de tratamento de imagens em conjunto com técnicas de controle (Bing et al., 2018; Jhung et al., 2018; Samuel et al., 2018).

Uma das abordagens para *lane keeping* aplica algoritmos de tratamento de imagem para localizar e extrair as características da faixa que serão utilizadas para gerar uma saída de controle adequada. Pode-se identificar as faixas na imagem utilizando algoritmos de detecção de bordas, linhas, curvas, cor, entre outras características que as definem (Yim and Oh, 2003; Lee and Moon, 2018). Essas técnicas podem ser ainda aplicadas em conjunto, com o intuito de tornar a detecção das faixas mais robusta.

Redes Neurais Artificiais (RNAs) são algoritmos que funcionam como sistemas de computação paralelos (Zurada, 1992) e possuem diversas aplicações. Algumas dessas são a classificação e extração de características de imagens, reconhecimento de fala, regressão, entre outras. As RNAs podem ser utilizadas em direção autônoma de pelo menos três formas: percepção mediada, reflexo de comportamento e percepção direta (Chen et al., 2015). Na percepção mediada, a rede é treinada para identificar os parâmetros da pista, como sua posição na imagem, raio de curvatura e sinalização. No reflexo de comportamento, a rede é treinada para gerar um comando direto de uma imagem de entrada. Por fim, na percepção direta, a rede é treinada para prever a oportunidade de executar ações de direção.

Neste artigo serão abordadas três técnicas para solucionar o problema de *lane keeping* de um veículo de quatro rodas em escala reduzida com controle diferencial. Nas três soluções a entrada é a imagem da pista gerada por uma câmera embarcada no veículo e a saída são os comandos para os motores que tracionam as rodas. A primeira solução baseia-se em um algoritmo que extrai as características das faixas contidas na imagem utilizando a técnica de janelas deslizantes. Uma lei de controle usa essas características e gera os comandos para os motores. A segunda solução utiliza uma rede neural convolucional profunda de percepção direta treinada para imitar a primeira solução usando a imagem da câmera sem tratamento. A terceira solução aplica a mesma topologia de rede neural que a anterior treinada com dados gerados observando um ser humano dirigir o veículo simulado.

Foi criado um ambiente de testes no simulador V-REP (M. S. Grewal, 2013) com o intuito de reproduzir um veículo real e as ciclovias localizadas nas proximidades do ITA (Instituto Tecnológico de Aeronáutica), onde pretende-se aplicar as soluções propostas. O veículo real possui uma câmera USB embarcada, *encoders* no eixo dos motores traseiros, um receptor *Global Navigation Satellite System* (GNSS) com capacidade para *Real Time Kinematic* (RTK) e uma bússola digital.

Em diversas aplicações de veículos autônomos é interessante que o algoritmo tenha capacidade de localização. Uma das formas mais consagradas de implementar a solução para esta tarefa é através do Filtro de Kalman (FK), um estimador de estados de um sistema dinâmico a partir de medidas corrompidas por ruídos. Para tal, é importante que sejam usados bons modelos para ruído dos sensores e para o comportamento dinâmico do veículo (Almeida, 2017).

2. FORMULAÇÃO E SOLUÇÃO DO PROBLEMA

2.1 Lane Keeping

Lane Keeping é um tipo de seguimento de caminho que consiste em manter um robô ou veículo em torno de um caminho de referência. Neste trabalho, tal caminho é definido por duas faixas brancas no asfalto sendo a referência a média entre essas faixas. As características do ambiente de simulação, incluindo dados do veículo e das faixas, foram ajustadas para melhor representar a situação que espera-se encontrar nos testes reais. Tais parâmetros podem ser observados na tabela 1.

Tabela 1. Dimensões do veículo e da pista simulados

Parâmetro	Valor	Unidade
Massa do veículo	$12,7$	kg
Largura do veículo incluindo rodas	44	cm
Comprimento do veículo	53	cm
Diâmetro das rodas	$7,5$	cm
Altura total da câmera	45	cm
Torque dos motores	$4,5$	kg cm
Distância média entre as faixas	85	cm
Largura das faixas	5	cm

2.2 Solução Visual Proporcional

A primeira solução proposta utiliza como entrada a imagem de uma câmera embarcada para identificar as faixas e extrair suas características. Essas características serão utilizadas para definir o raio de curvatura da trajetória e o erro de posição do veículo em relação ao centro da trajetória de referência. Este algoritmo foi baseado na solução proposta por Dwivedi (2017).

Para que o algoritmo funcione adequadamente deve-se fazer uma transformação de perspectiva, de forma que as faixas fiquem paralelas quando o trecho for uma reta, simulando uma visão superior da pista. Esse tipo de tratamento é uma modificação geométrica da imagem e é conhecida como *bird's eye view* (Pratt, 2007). As características da trajetória serão calculadas na imagem em perspectiva. Portanto é necessário estabelecer uma relação de transformação de pixels para metros em ambos os eixos de coordenadas da imagem. Essa relação foi calculada através de um padrão quadriculado no ambiente simulado, onde cada elemento possui lados com comprimento de meio metro.

Para identificar a faixa utiliza-se detecção de bordas, linhas e cor. Esses dados são utilizados para gerar uma

imagem binária, contendo apenas os pixels de interesse. O algoritmo de janelas deslizantes (Wei and Tao, 2010) é utilizado para definir um conjunto de pontos pertencentes às faixas, através dos quais será ajustado um polinômio de segundo grau que será usado para calcular o raio de curvatura e a distância do veículo em relação a referência. O resultado desses processamentos é demonstrado na figura 1.

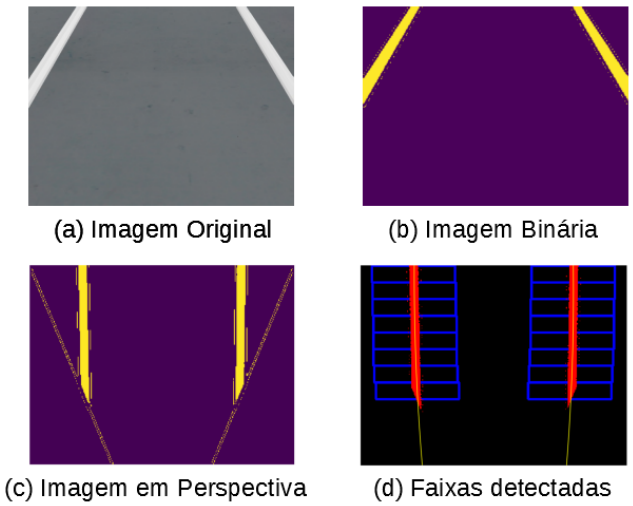

Figura 1. Polinômios ajustados as faixas

A eq. (1) foi utilizada para definir a velocidade base do veículo em função do raio de curvatura das faixas, que aproxima a resposta do controlador ao comportamento humano (Odhams, 2006), e a eq. (2) define a diferença de velocidade entre as rodas em função do erro para o centro da trajetória:

$$V = V_{max} * (1 - e^{-\lambda * R}) \tag{1}$$

$$dV = \frac{2 * V * E_C}{L} \tag{2}$$

onde V = velocidade base do veículo em metros por segundo, V_{max} = velocidade em uma reta em metros por segundo, λ = constante que define a variação da velocidade base, R = raio de curvatura da faixa em metros, dV = diferença de velocidade entre as rodas, E_C = erro em relação ao centro da faixa em metros e L = largura da faixa em metros.

2.3 Solução Neural Proporcional

A segunda solução proposta neste trabalho utiliza uma rede neural convolucional profunda proposta por Bojarski et al. (2016). Trata-se de uma rede de ponta-a-ponta (*end-to-end*) para controlar veículos autônomos, cuja entrada é uma imagem sem tratamento e a saída é o ângulo de volante. Dado que o veículo utilizado nas simulações é diferencial, o controle de guinada deve ser feito através da diferença de velocidade entre as rodas. No entanto, como a rede neural possui apenas uma saída, foi desenvolvido um mapeamento de velocidade para as rodas.

O controlador funciona determinando uma razão entre as velocidades das rodas. Esse valor é uma porcentagem da velocidade base e seu sinal determina a direção de curva, onde valores negativos representam correções para a esquerda e valores positivos para a direita, conforme a eq. (3). Portanto, sempre que a saída da rede é negativa, a velocidade da roda direita é mantida em seu valor base e a roda esquerda tem sua velocidade reduzida até um mínimo de zero. O oposto acontece caso a saída da rede seja positiva. Além disso, a velocidade base do veículo é diretamente proporcional ao raio de curvatura da pista, conforme a eq. (4).

$$r = \begin{cases} (1 - \frac{V_{direita}}{V_{esquerda}}) & \text{se } V_{esquerda} \geq V_{direita} \\ (-1 + \frac{V_{esquerda}}{V_{direita}}) & \text{se } V_{direita} > V_{esquerda} \end{cases} \tag{3}$$

$$V_{Base} = \frac{V_{Max} * (1 - e^{-\lambda * (1 - |r|)})}{(1 - e^{-\lambda})} \tag{4}$$

Onde $V_{direita}$ = velocidade da roda direita em m/s, $V_{esquerda}$ = velocidade da roda esquerda em m/s, λ = constante positiva que define a variação da velocidade base e r = saída esperada da rede.

Para o treinamento da rede neural foi criado um conjunto de dados utilizando como entradas as imagens não tratadas da câmera e como saída desejada o comando gerado pelo algoritmo da primeira solução. Com esse método foram armazenadas 2825 imagens em diversos trechos da pista, compostas por curvas para a esquerda, direita e retas. Para aumentar o conjunto de treinamento, todas as imagens de curva foram duplicadas e invertidas, assim como os comandos de saída associados a elas. Além disso, para tornar a rede neural mais robusta a ruídos, 30% das imagens foram degradadas intencionalmente por adição de ruídos gaussianos e do tipo "sal e pimenta". O resultado dessas modificações é ilustrado na figura 2.

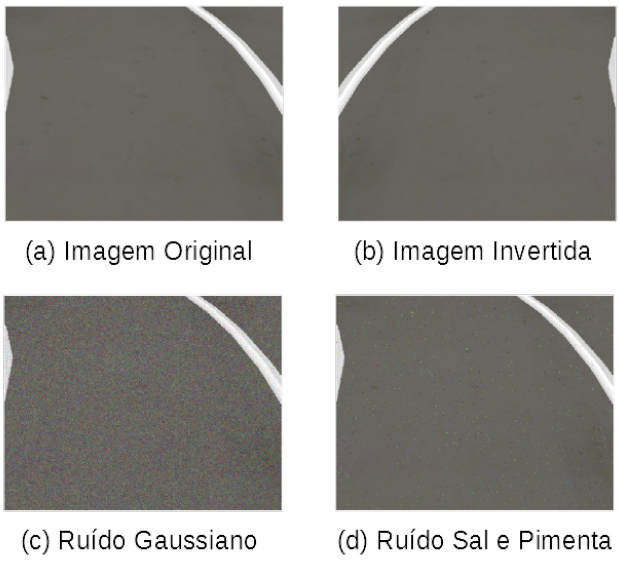

Figura 2. Tratamentos utilizados na base de treinamento

Utilizando essas técnicas e os conjuntos criados foram obtidos os resultados de treinamento mostrados na figura 3. O conjunto de validação foi composto por 20% das

imagens do banco de dados escolhidas aleatoriamente e essas imagens não foram usadas durante o treinamento.

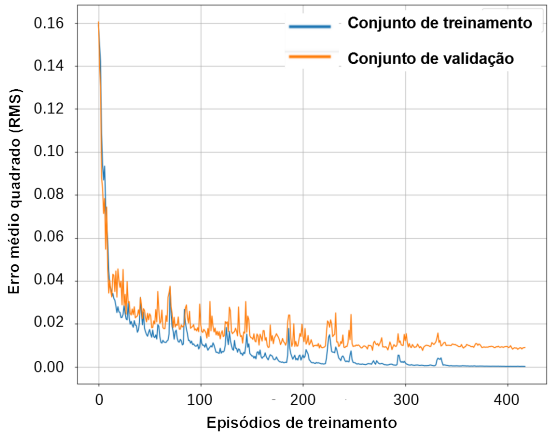

Figura 3. Resultado do treinamento com dados gerados pelo controlador automático

2.4 Solução Neural Manual

A terceira solução proposta neste trabalho utiliza a mesma topologia de rede neural do controlador anterior. No entanto, o objetivo de seu treinamento é diferente. Ao invés de aprender com um algoritmo automático que possui controles consistentes de acordo com a leitura dos sensores, pretendeu-se treiná-la com comandos dados pelo ser humano. Como a pista é relativamente estreita em relação as dimensões do veículo, o controle manual em velocidades altas é difícil, pois qualquer pertubação faz com que o veículo saia da pista. Para evitar esse problema durante a captura das imagens, a velocidade do veículo foi limitada a no máximo 1 km/h.

O conjunto de treinamento e validação foi levantado no mesmo percurso da solução anterior. No entanto, a frequência de captura das imagens foi reduzida para evitar a repetição de informação devido à baixa velocidade do veículo. Ao final da captura dos dados o conjunto contou com 1850 imagens. Foi feito o mesmo tratamento de duplicação e adição de ruídos do conjunto anterior. Com esses dados, obteve-se o resultado de treinamento da figura 4.

2.5 Sistema de Localização

O veículo utilizado para as simulações é terrestre com quatro rodas e controle diferencial. Dessa forma a sua postura é determinada por 3 variáveis: as coordenadas (x, y) no plano local e o ângulo de guinada (θ). O Filtro de Kalman (Nascimento Jr., 1988) foi utilizado para estimar essas 3 variáveis usando as medidas geradas por sensores ruidosos e o modelo dinâmico do veículo. Neste trabalho serão utilizados três tipos de sensores: GNSS RTK, bússola digital e odometria de rodas.

A solução para GNSS usando RTK (Real-time Kinematic) é uma técnica de localização diferencial, o que significa que existem um ou mais sensores coletando sinais para aumentar a sua acurácia e confiabilidade (Parkinson and Spilker, 1997). Isso é alcançado compensando a influência

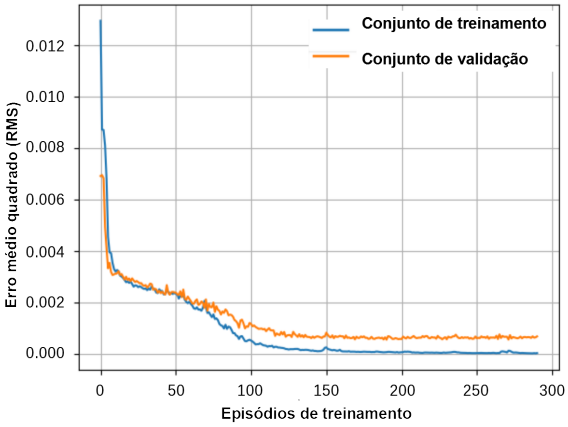

Figura 4. Resultado do treinamento com dados gerados pelo ser humano

dos erros das órbitas dos satélites e da atmosfera. Entre os receptores utilizados deve existir pelo menos um que se mantém fixo por um longo tempo o qual é chamado de base. Dessa maneira, é possível diminuir a incerteza em relação a posição desse receptor e então fornecer dados de correção aos dispositivos móveis. No GNSS RTK, a base também é utilizada para sincronizar a fase da portadora do sinal recebido, fornecendo resolução em níveis centimétricos da medida de posição do receptor móvel (Langley, 1998). Em função das distorções na propagação da onda de alta frequência, o erro relativo da solução RTK depende da distância entre a base e o receptor móvel. Portanto, recomenda-se mantê-los em distâncias menores que 10 km.

Para que o sistema de localização funcione adequadamente é necessário um modelo fiel da dinâmica do veículo. Segundo Corke (2011), a variação dos estados para um veículo diferencial são dadas pelas eqs. de (5) a (7), onde dR é a distância percorrida pela roda direita, dL é a distância percorrida pela roda esquerda e W é a distância entre as duas rodas do veículo.

$$\dot{x} = \frac{1}{2}(dR + dL)cos(\theta) \tag{5}$$

$$\dot{y} = \frac{1}{2}(dR + dL)sen(\theta) \tag{6}$$

$$\dot{\theta} = \frac{(dR - dL)}{W} \tag{7}$$

Essas equações serão utilizadas na etapa de propagação do filtro, através da leitura de odômetros para obter o deslocamento das rodas e uma bússola digital será usada para medir a orientação angular do veículo. No entanto, as medidas dos sensores são corrompidas por erros, portanto ao estimar o comportamento do veículo através de seus sensores obtemos as eqs. de (8) a (10).

$$\dot{x}^{NAV} = \frac{1}{2}(dR + \omega_R + dL + \omega_L)cos(\theta^{NAV}) \tag{8}$$

$$\dot{y}^{NAV} = \frac{1}{2}(dR + \omega_R + dL + \omega_L) sen(\theta^{NAV}) \quad (9)$$

$$\dot{\theta}^{NAV} = \frac{(dR + \omega_R - dL - \omega_L)}{W} \quad (10)$$

Onde $\dot{x}^{NAV}$, $\dot{y}^{NAV}$ e $\dot{\theta}^{NAV}$ são as equações dinâmicas do veículo quando são utilizados sensores contaminados por ruído, ω_R é o ruído de odometria da roda direita e ω_L é o ruído de odometria da roda esquerda.

Neste trabalho, a etapa de atualização fará a estimação do erro entre os estados gerados na propagação e os estados verdadeiros do veículo. Para isso deve-se criar um sistema de erro que expressa a diferença entre a localização estimada e a real demonstrado na eq. (11), onde $\dot{x}^V(t)$, $\dot{y}^V(t)$ e $\dot{\theta}^V(t)$ são as variações nos estados verdadeiros

$$\dot{X}^E(t) = \begin{bmatrix} \dot{x}^{NAV}(t) - \dot{x}^V(t) \\ \dot{y}^{NAV}(t) - \dot{y}^V(t) \\ \dot{\theta}^{NAV}(t) - \dot{\theta}^V(t) \end{bmatrix} \quad (11)$$

Aplicando as expressões do modelo dinâmico do veículo representado pelas eqs.(5), (6) e (7), além das eqs. (8), (9) e (10) na eq. (11) e considerando que a orientação fornecida pela bússola, geralmente possui erros suficientemente pequenos, de tal forma que a medida do sensor pode ser considerada igual à orientação real (dos Santos et al., 2013), obtemos o sistema de erro da eq. (12).

$$\dot{X}^E(t) = \begin{bmatrix} 0.5(\omega_R + \omega_L) cos(\theta^V) \\ 0.5(\omega_R + \omega_L) cos(\theta^V) \\ 0.5(\omega_R - \omega_L) \end{bmatrix} \quad (12)$$

Onde $\dot{X}^E(t)$ é o vetor de erro de variação nos estados.

Assim, a propagação do sistema discretizado é dada pelo seguinte conjunto de equações:

$$\begin{bmatrix} x^E_{k+1} \\ y^E_{k+1} \\ \theta^E_{k+1} \end{bmatrix} = \begin{bmatrix} x^E_k \\ y^E_k \\ \theta^E_k \end{bmatrix} + \begin{bmatrix} cos(\theta_k)^V & cos(\theta_k)^V \\ sen(\theta_k)^V & sin(\theta_k)^V \\ \frac{1}{W} & -\frac{1}{W} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (13)$$

Onde x_K, y_K, θ_K denotam o valor dos estados no tempo $t = kT$.

Utilizando a simplificação mencionada, o sistema de erro formado se torna linear, então é possível aplicar o Filtro de Kalman sem linearização. O algoritmo utilizado neste trabalho é descrito a seguir.

Passo 1: Inicialização do Filtro de Kalman:

Para inicializar o filtro, o usuário deve fornecer os seguintes dados: estados iniciais do veículo, matriz de covariância do erro destes estados e a matriz de ruído dos sensores.

$$X_0^{NAV-} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} \quad (14)$$

$$P_0^- = diag\{\alpha_1, \alpha_2, \alpha_3\} \quad (15)$$

$$R = diag\{\sigma_x^2, \sigma_y^2, \sigma_\theta^2\} \quad (16)$$

Onde X_0^{NAV-} são os estados iniciais do sistema de localização, P_0^- é a matriz de covariância dos erros iniciais σ_x^2, σ_y^2 e σ_θ^2 são as variâncias dos sensores nas três coordenadas.

Inicializa-se a contagem para o tempo do filtro.

Passo 2: Atualização do Filtro de Kalman:

Quando a contagem de tempo para o filtro for um múltiplo da taxa de amostragem de atualização, deve-se realizar a leitura do GPS e bússola. Em seguida, esses dados são usados para calcular o ganho do filtro e atualizar os estados.

$$X_k^E = X_k^{NAV-} - \begin{bmatrix} x_k^{UP} \\ y_k^{UP} \\ \theta_k^{UP} \end{bmatrix} \quad (17)$$

$$G_k = P_k^- H^T [H P_k^- H^T + R]^{-1} \quad (18)$$

$$P_k^+ = [I_{3x3} - G_k H] P_k^- \quad (19)$$

$$X_k^{NAV+} = X_k^{NAV-} - G_k X_k^E \quad (20)$$

Onde X_k^E é o erro nos estados, X_k^{NAV-} são os estados estimados, G_k é o ganho de Kalman estimado no tempo $t = kT$, P_k^- é a incerteza estimada, P_k^+ é a incerteza atualizada, R é a matriz de covariância de ruído do GPS e bussola, x_k^{UP}, y_k^{UP} e θ_k^{UP} são os estados fornecidos pelos sensores utilizados na etapa de atualização.

Passo 3: Propagação do Filtro de Kalman:

Quando o tempo for um múltiplo da taxa de amostragem de propagação, deve ser feita a leitura do odômetro para calcular a posição do veículo, baseando-se no modelo dinâmico e nas leituras da odometria. Nesta etapa, faz-se a propagação da matriz de covariância dos erros do modelo.

$$X_{k+1}^- = X_k + \begin{bmatrix} cos(\theta_k^V) & cos(\theta_k^V) \\ sen(\theta_k^V) & sin(\theta_k^V) \\ \frac{1}{W} & -\frac{1}{W} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (21)$$

$$P_{k+1}^- = A_d P_k^+ (A_d)^T + B_D Q_k (B_D)^T \quad (22)$$

Onde A_D e B_D são as matrizes discretizadas do modelo, θ^V é a orientação verdadeira do veículo e Q_K é a matriz de covariância do erro de odometria.

2.6 Variância dos sensores embarcados

Visando melhorar a modelagem dos ruídos dos sensores simulados, foram levantados os histogramas dos erros dos sensores embarcados no veículo real. No caso do receptor GPS, sua antena foi colocada em uma posição fixa, livre de obstruções, e os dados de posição foram coletados por duas horas. Os resultados obtidos podem ser vistos nas figuras 5, 6 e 7. As medidas do receptor GPS foram convertidas do sistema de coordenadas geodésico para o sistema de

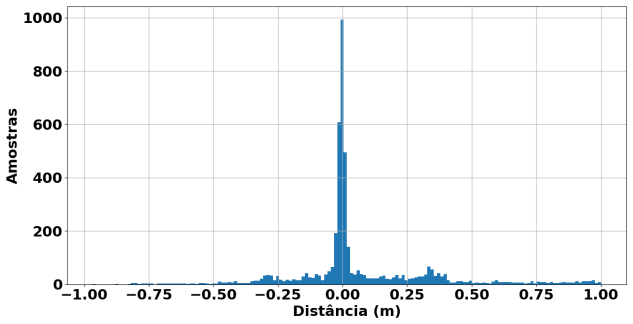

Figura 5. Histograma do erro de posição em latitude

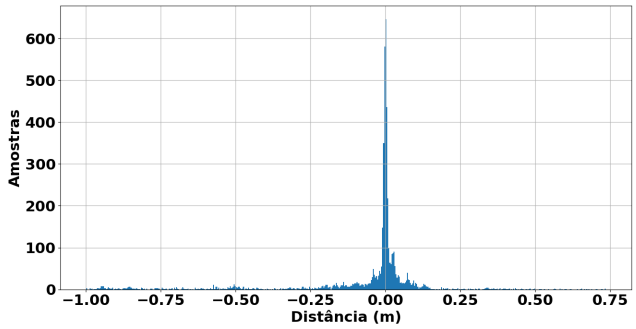

Figura 6. Histograma do erro de posição em longitude.

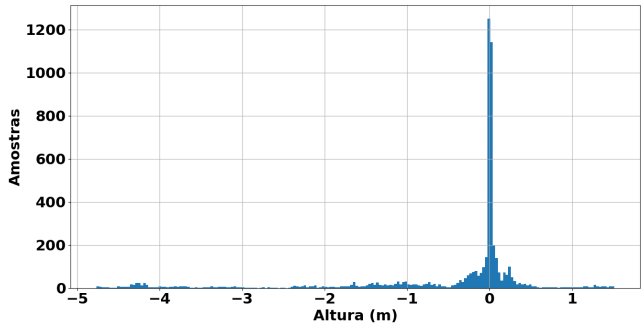

Figura 7. Histograma do erro de posição em altitude.

coordenadas retangular seguindo o procedimento descrito em E. Rohmer (2001).

Para determinar a variância da bússola digital, ela foi calibrada (Almeida, 2017) e em seguida mantida em uma posição fixa por duas horas, armazenando as diversas leituras obtidas. O histograma resultante destes dados é demonstrado na figura 8.

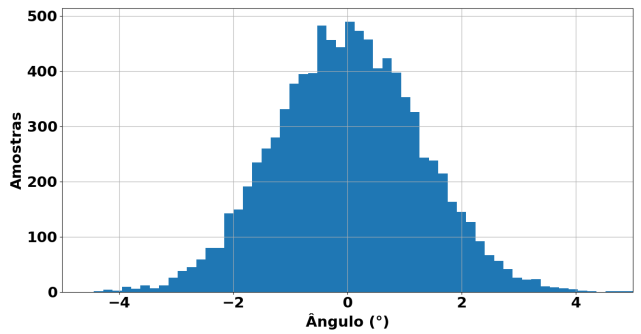

Figura 8. Histograma do erro na orientação.

As variâncias e desvios padrões associados aos histogramas para os erros dos sensores são mostradas na tabela 2. O veículo real ainda não possui os sensores de odometria instalados. Portanto não foi possível obter dados reais sobre os erros de odometria e os parâmetros desses erros foram estimados.

Tabela 2. Variâncias e desvios padrões.

Variável	Variância (σ^2)	Desvio Padrão (σ)
GPS_{lat}	0.0674 m^2	0.2597 m
GPS_{long}	0.0472 m^2	0.2173 m
GPS_{alt}	1.2476 m^2	1.1170 m
Bússola	2.9231 ^{o2}	1.7097 ^{o}
Odometria	0.0025 m^2	0.05 m

3. RESULTADOS

Para validar os algoritmos de localização e controle, foram feitos testes no ambiente de simulação, onde modelou-se um veículo com características similares ao que espera-se encontrar na prática. Para o sistema de sensoriamento, foram simulados os seguintes sensores: uma câmera virtual, um GPS, uma bússola digital e um odômetro em cada eixo dos motores traseiros. Com o intuito de tornar a simulação mais realista, os sensores foram corrompidos com ruído gaussiano com a mesma variância observada nos sensores reais. Além disso, o torque dos motores e as dimensões do veículo tiveram seus valores ajustados para serem iguais aos valores dos dispositivos reais.

3.1 Resultados dos controladores

Para verificar o desempenho dos controladores projetados, dois testes foram realizados. No primeiro, a simulação foi iniciada com o veículo fora do centro da trajetória e foi observada a distância do veículo para o caminho de referência em função do tempo. Tal experimento é ilustrado nas figuras 9, 10 e 11.

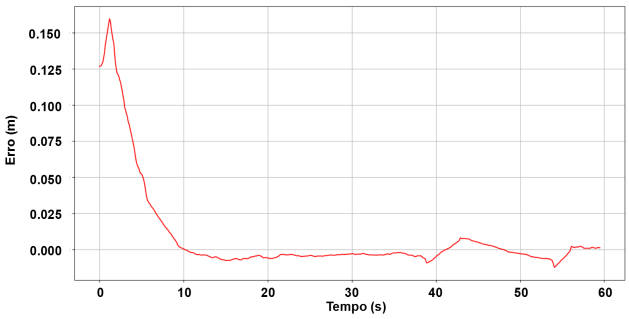

Figura 9. Distância do veículo em relação ao centro da pista com o controlador analítico.

Já no segundo teste, uma trajetória fechada com diversas curvas para ambos lados foi construída. As três soluções foram aplicadas com diversos parâmetros para velocidade máxima, um trecho do percurso é ilustrado na figura 12. Os controladores deram cinco voltas na pista para cada velocidade. Desta forma, foram calculados os erros médios quadráticos em relação à trajetória de referência.

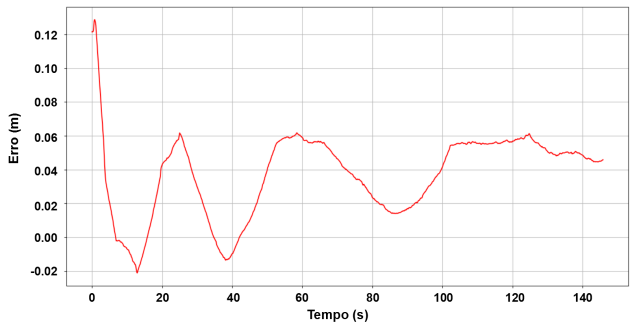

Figura 10. Distância do veículo em relação ao centro da pista da rede, treinada com dados obtidos pelo ser humano.

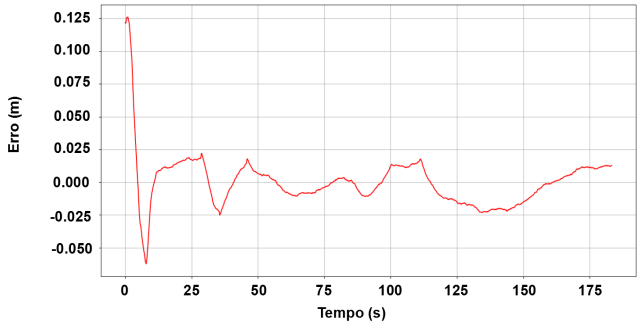

Figura 11. Distância do veículo em relação ao centro da pista da rede treinada com dados obtidos pelo controlador analítico.

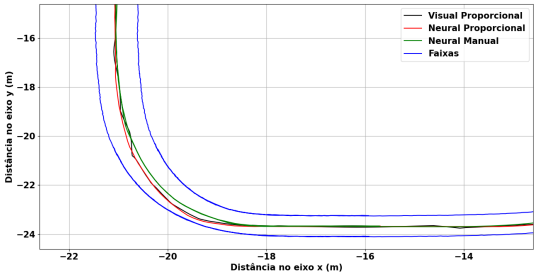

Figura 12. Trajetória do veículo com cada controlador e velocidade média de $2.7km/h$

3.2 Comparação das soluções propostas

Para comparar as soluções propostas, foi feita uma simulação onde o veículo percorreu uma pista, composta por diversas curvas, com cada controlador e diversos ajustes de velocidade máxima, por cinco voltas. Foram medidos o erro médio quadrático em relação a trajetória de referência e o tempo médio de processamento de cada solução.

Conforme observado na figura 13, todas as soluções propostas apresentam erros proporcionais à velocidade média do veículo. O controlador proporcional mantém-se com melhor desempenho em todos os casos. No entanto, os controladores baseados em rede neural possuem melhor tempo de processamento, conforme visto na tabela 3, o que pode ser uma vantagem para sua integração em sistemas complexos que necessitam utilizar a CPU do computador embarcado para realizar outras tarefas. Por outro lado o consumo maior de memória RAM demanda um dispositivo com a capacidade adequada.

É importante ressaltar, ainda, que a construção do banco de dados para o controlador neural manual apresenta custo maior em relação ao neural proporcional, já que depende do piloto para geração dos dados. Portanto, apesar do desempenho e custo computacional similar, o seu treinamento é uma desvantagem.

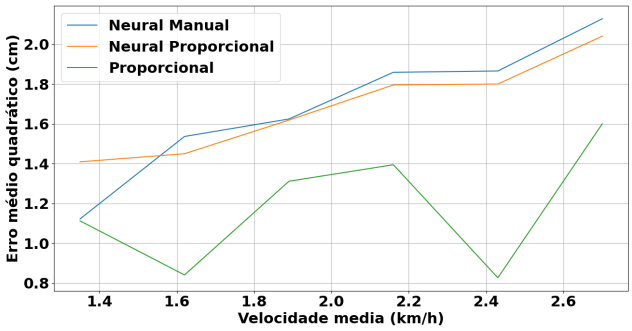

Figura 13. Erro médio quadrático em relação ao centro da pista para os três controladores

Tabela 3. Frequência média de processamento dos controladores.

Controlador	Tempo de processamento médio
Proporcional	106.4 ms
Neural proporcional	12.44 ms
Neural manual	12.53 ms

3.3 Sistema de Localização

Para testar o sistema de localização, o veículo foi simulado realizando um trajeto com uma curva. O teste foi executado com todas as condições ideais, as covariâncias dos sensores informadas para o filtro estavam corretas e o erro na posição inicial estava dentro da faixa de incerteza informada. Desta forma foram obtidos os resultados demonstrados nas figuras 14 e 15. Em seguida, foi feito um teste com o erro na posição inicial maior do que a estimativa da incerteza da posição inicial e na estimativa do ruído dos sensores. No caso em que o filtro foi inicializado com a posição acima da incerteza informada, o filtro convergiu rapidamente após a fase de atualização. Já na situação onde o ruído do sensor foi subestimado, a resposta do algoritmo é degradada e pode chegar a divergir se o erro for muito grande.

4. CONCLUSÃO

Foram apresentados três possíveis soluções para o problema de *lane keeping* de um veículo autônomo diferencial com sistema de localização. Todos os controladores testados foram capazes de realizar a tarefa proposta, se manter e centralizar o veículo entre as duas faixas em um trecho com curvas variadas. Cada solução demonstra certas vantagens e desvantagens na sua aplicação. Em relação à sua capacidade de seguimento de trajetória, o controlador visual proporcional centraliza o veículo mais rapidamente que os outros e apresenta menor erro médio quadrático para o centro da pista até mesmo em velocidades mais altas, no entanto, necessita de mais tempo do

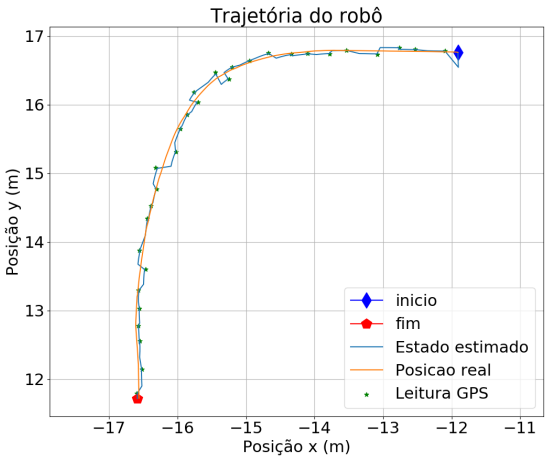

Figura 14. Resultado do sistema de localização

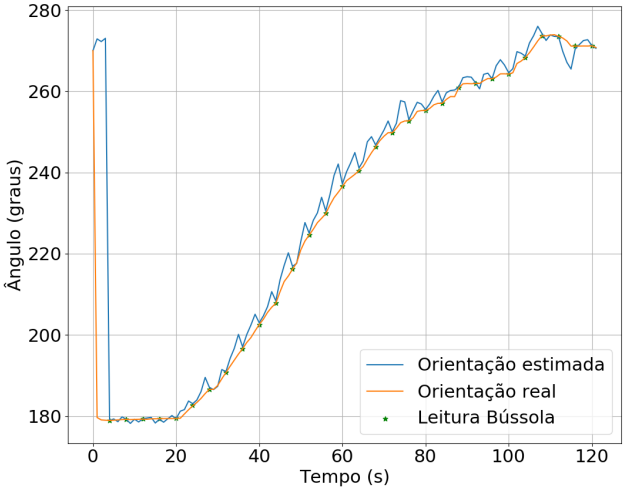

Figura 15. Resultado do sistema de localização com erro na posição inicial maior que a incerteza inicial

processador para gerar a saída. Já os dois controladores baseados em rede neural apresentam erro médio para o centro da pista e tempo de processamento similares, se diferenciando na facilidade de implementação, onde o neural manual apresenta maior grau de dificuldade, já que necessita de um piloto humano para geração do banco de treinamento. No futuro, pretende-se aplicá-los no veículo real para determinar se tais resultados também são válidos fora do ambiente de simulação.

AGRADECIMENTOS

Os autores do artigo agradecem à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro concedido ao primeiro autor na forma de uma bolsa de mestrado.

REFERÊNCIAS

Almeida, H.P. (2017). *Navegação Autônoma de um Veículo Terrestre em Escala Reduzida Usando GPS/INS de Baixo Custo*. Master's thesis, Instituto Tecnológico de Aeronáutica.

Bing, Z., Meschede, C., Huang, K., Chen, G., Rohrbein, F., Akl, M., and Knoll, A. (2018). End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle. *2018 IEEE International Conference on Robotics and Automation (ICRA)*.

Bojarski, M., Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). DeepDriving: Learning affordance for direct perception in autonomous driving. *2015 IEEE International Conference on Computer Vision (ICCV)*, 468–474.

Corke, P. (2011). *Robotics, Vision and Control*. Springer Nature, Berlin.

dos Santos, D.S., Nascimento, C.L., and Cunha, W.C. (2013). Autonomous navigation of a small boat using IMU/GPS/digital compass integration. *2013 IEEE International Systems Conference (SysCon)*, 468–474.

Dwivedi, P. (2017). Automatic lane detection for self driving cars. URL `shorturl.at/fkyB3`. Acessado em 13-9-2020.

E. Rohmer, S. P. N. Singh, M.F. (2001). *Global Positioning Systems, Inertial Navigation, and Integration*. A John Wiley & Sons, Inc. Publication, New York.

Jhung, J., Bae, I., andTaewoo Kim, J.M., Kim, J., and Kim, S. (2018). End-to-end steering controller with cnn-based closed-loop feedback for autonomous vehicles. *2018 IEEE Intelligent Vehicles Symposium (IV)*.

Langley, R.B. (1998). GPS rtk. *GPS World*, 9, 70–72.

Lee, C. and Moon, J.H. (2018). Robust lane detection and tracking for real-time applications. *IEEE Transactions on Intelligent Transportation Systems*.

M. S. Grewal, L. R. Well, A.P.A. (2013). Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*. Www.coppeliarobotics.com.

Nascimento Jr., C.L. (1988). *Estudo Comparativo de Métodos de Combate à Divergência de Filtros de Kalman*. Master's thesis, Instituto Tecnológico de Aeronáutica.

Odhams, A.M.C. (2006). *Identification of Driver Steering and Speed Control*. Ph.D. thesis, Queen's College.

Parkinson, B.W. and Spilker, J.J. (1997). *Progress In Astronautics and Aeronautics: Global Positioning System: Theory and Applications*, volume 1. American Institute of Aeronautics & Astronautics.

Pratt, W.K. (2007). *Digital Image Processing: PIKS Scientific Inside*. Wiley-Interscience, 4th ed., newly updated and rev. ed edition.

Samuel, M., Mohamad, M., Hussein, M., and Saad, S.M. (2018). Development of lane keeping controller using image processing. *International Journal of Computing and Network Technology*.

Wei, Y. and Tao, L. (2010). Efficient histogram-based sliding window. 3003–3010.

Yim, Y.U. and Oh, S.Y. (2003). Three-feature based automatic lane detection algorithm (tfalda) for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.

Zurada, J.M. (1992). *Introduction to Artificial Neural Systems*. West Publishing Company, Minnesota.