

CONTROLE DE TRAJETORIA DE UM ROBÔ MÓVEL BASEADO NA ARQUITETURA ACKERMANN

ARTUR F. MOREIRA¹, PEDRO H. A. MIRANDA², ANDRESSA DA S. FERNANDES³, ALAN V. DE A. BATISTA¹, JOSÉ A. B. DA SILVA¹

¹Instituto Federal de Ciências, Educação e Tecnologia do Ceará (IFCE) – Alameda José Quintino, s/n, Prado, CEP 63400-000, Tel.: +55 (88) 3564-1000 – Cedro – CE

²UNIVERSIDADE FEDERAL DO CEARÁ (UFC) – AV. DA UNIVERSIDADE, 2853, BENFICA, CEP 60020-181, TEL.: +55 (85) 3366 7300 – FORTALEZA – CE

³Universidade Federal do Ceará (UFC) – Rua Coronel Estanislau Frota, 563, Centro, CEP 62010-560, Tel.: +55 (88) 3695 4630 – Sobral – CE

E-mails: artur31415926@gmail.com, pedrohenriqbg@gmail.com, andressafernandes06.af@gmail.com, alan.ifce@gmail.com, ailton.ifce@gmail.com

Abstract— The purpose of this article is an implementation of a control system using PID control in a servomotor, applied to a mobile robot. The application of the control technique is aimed at the use of TFC-KIT: the Freescale Cup Intelligent Car Development System, which is the production of the prototype of an autonomous robot. The built prototype is a standalone robot capable of following a path delimited by two black lines on a safety surface from the application of optical sensor (linear camera). The goal is slightly better than any of the external control types. An implementation of the proposed control brings an alternative to improve the performance response time of the mobile robot before the on-off control previously implemented.

Keywords— Autonomous, Competition, Robot, Line Follower.

Resumo— O objetivo deste artigo é a implementação de um sistema de controle de trajetória usando controle PID em um servomotor, aplicado a um robô móvel. A aplicação da técnica de controle visa à utilização e aplicação do TFC-KIT: The Freescale Cup Intelligent Car Development System, consistente na produção do protótipo de um robô autônomo. O protótipo construído é um robô autônomo capaz de seguir um trajeto delimitado por duas linhas pretas sobre uma superfície de cor branca a partir da aplicação de sensor óptico (câmera linear). A meta a ser atingida é obter um carro para seguir os trajetos não definidos da competição no menor tempo possível e sem nenhum tipo de controle externo. A implementação do controle proposto traz uma alternativa de melhorar o tempo de resposta de atuação do robô móvel perante ao controle on-off anteriormente implementado.

Palavras-chave— Autônomo, Competição, Robô, Seguidor De Linha.

1 Introdução

Atualmente a robótica é uma área de pesquisa que tem ganhado espaço em virtude do avanço tecnológico bastante disseminado nos últimos anos. Os robôs têm auxiliado o homem em diversas tarefas, tais como, a exploração espacial, auxílio à pessoas com deficiências físicas, realização de trabalhos industriais ou simplesmente como entretenimento (Pscheidt, 2007).

Robôs autônomos têm a capacidade de obter informações sobre seus ambientes, e trabalhar por um longo período de tempo, sem intervenção humana. Esse tipo de robô é comumente visto, seja em helicópteros autônomos ou robôs para fins domésticos. Estes robôs autossuficientes podem mover-se durante toda a operação sem assistência humana, e são capazes de evitar situações que são prejudiciais para si, para as pessoas à sua volta. Robôs autônomos também são susceptíveis a adaptarem-se aos novos ambientes. Os robôs autônomos mais simples utilizam sensores infravermelhos ou ultrassom para identificar obstáculos, permitindo-lhes navegar em torno dos obstáculos sem controle humano. Robôs mais avançados usam a visão estereó para ver ambientes, ou seja, câmeras lhes dão a percepção de profundidade, e a implementação de algoritmos lhes permitem localizar e classificar os objetos em tempo real (Robot Worx, 2016).

O projeto de um sistema robótico envolve a especificação e seleção de diferentes componentes, sensores e atuadores e a combinação destes em um sistema autônomo. Este deve ser projetado de modo a ser dotado de dispositivos capazes de prover dados relevantes (obtidos através dos seus sensores), para que o sistema de controle robótico possa planejar e acionar os atuadores de modo a executar a ação desejada. (Wolf et al., 2009).

Os sensores fornecem apenas uma informação parcial, incompleta e sujeita a erros, sendo papel do sistema de controle realizar a fusão dos sensores e tratar estas informações de forma robusta. Os comandos de atuação também não são precisos, pois estão sujeitos a erros de posicionamento do robô e da atuação dos motores, cabe ao sistema de controle prover formas de corrigir/compensar esses erros para que as ações sejam executadas de forma correta (Thrun, 2004).

Na utilização de sistemas robóticos, é necessário o uso de sistemas de simulação computacional, para que sejam feitos os testes e as correções do projeto antes da aplicação real. Isto gera maior segurança, pois com a análise prévia em ambientes simulados pode-se observar elementos que não se adequariam ao projeto antes mesmo de se fazer testes reais. Para adaptação de um modelo a simulação, é necessário criar um modelo simplificado da realidade usando um ambiente

computacional. Na área de robótica isto significa modelar: sensores, atuadores e comportamento físico do robô (Osório, 2006).

Reconhecimento de trajetória é a premissa para direção autônoma dos chamados veículos inteligentes (*Intelligent Vehicle*) (Liu et al., 2008; Afonso et al., 2006; Kolski, 2006). A área de pesquisa da Inteligência Artificial (IA) propõe técnicas diversas para navegação inteligente de robôs móveis e outros artefatos, as quais integram resultados de pesquisas de vários campos da IA, como por exemplo, visão de máquina (*Machine Vision*). Muitos sistemas desenvolvidos com direção autônoma utilizam câmeras para auxiliar navegação inteligente (Liu et al., 2008).

Na literatura existem diversas propostas para o controle de trajetória de robôs móveis não-holonômicos com rodas (WMR - *Wheeled Mobile Robot*), sendo que a grande maioria das propostas usam o modelo para o robô baseado nas equações da cinemática, sendo denominado modelo cinemático, tanto pela simplicidade quanto pelo fato da maioria dos robôs terem como parâmetro de entrada as velocidades linear e angular e não valores de aceleração ou torque (Klancar et al., 2005). Porém, existem outras estruturas a serem utilizadas: Kanayama et al (2009) utiliza uma linearização da equação diferencial do sistema. Em Kuhne et al (2004) o modelo é linearizado levando em consideração o erro do robô de referência e o robô o real.

Entre os controladores propostos na atualidade existem vários ramos de estudos, incluindo a utilização de inteligência artificial, com propostas de utilização de lógica fuzzy e redes neurais, como apresentado em Fierro e Lewis (1995), enquanto que em Jang e Chung (2009) é proposto um controlador que combina lógica fuzzy e redes neurais. Há propostas de controladores PI e PID tanto com estruturas lineares como não linear, em Al-Araji et al (2011) é proposto um controlador PID adaptativo, já em Klancar et al (2005) o controlador proposto possui características de um controlador PI com termos não-lineares.

No cerne destas tecnologias está o desenvolvimento de um controlador PID para esta aplicação, de modo que o robô autônomo trabalhe de forma mais segura e que os erros apresentados no controle de trajetória deste robô móvel seja minimizados ao máximo, gerando assim mais eficiência e confiabilidade.

A estrutura do trabalho divide-se da seguinte forma: a seção 2 relata uma breve apresentação do TFC-KIT. Na seção 3, são descritos todos os componentes que podem ser utilizados para a construção de robô seguidor de linha, como, sensores, controladores centrais e componentes para controle de velocidade do motor. Os resultados são apresentados na seção 4, e as conclusões são expostas na seção 5.

2 Metodologia

2.1 TFC-KIT

A estrutura mecânica do carro consiste em uma plataforma de desenvolvimento, disponibilizada pelo fabricante, que é utilizada como material base para todos os participantes, não podendo sofrer quaisquer modificações em sua estrutura original.

A placa de desenvolvimento aplicada ao projeto será a FRDM-KL25Z, ilustrado na Figura 1, consistindo em uma plataforma de desenvolvimento de baixo custo da família Kinetis L. Tais microcontroladores são baseados na tecnologia de processamento de dados ARM® Cortex™ M0, cujas características, incluem: fácil acesso a entradas e saídas digitais bem como entradas analógicas, ambas possuem baixo consumo de energia, e apresentam uma interface de depuração para a programação *flash*.

O projeto eletrônico do carro foi dividido em duas partes: detecção e controle, cada um com suas funções e circuitos específicos. O sensoriamento é feito a partir de sensores ópticos que realizam o reconhecimento da mudança de estado (cor), o que ocasiona a mudança do estado binário do sinal, identificando essa mudança e atuando conforme a programação pré-designada. O sinal do sensor é definido por níveis de tensão, sendo enviado para o microcontrolador que executará a função da interrupção, processando o código fonte que fará o sincronismo entre os motores de propulsão e o leme do carro.

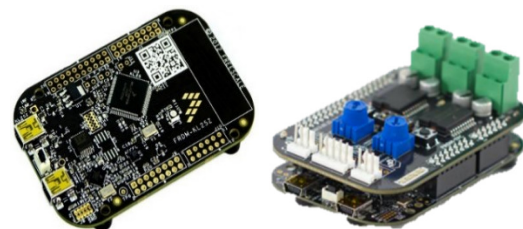


Figura 1 – Placa KL25Z e shield FRDM-TFC.

A câmera é uma combinação de um sensor de imagem e uma lente. A luz que se alterna no ambiente é captada através da lente que, em seguida, é desviada para o sensor. O dispositivo sensor é constituído internamente por uma matriz de 128 micro sensores ópticos, que serão responsáveis por identificar a intensidade de luz e, através de um meio analógico, o circuito de leitura, transmite as informações coletadas ao sistema de processamento de dados, onde será tratado por meio de algoritmos. O sensor de visão é o modelo TSL1401R-LF, projetado para uso em aplicações como: escaneamento de imagem, reconhecimento de caracteres (OCR), detecção de borda e posicionamento, entre outras (Farnell, 2017).

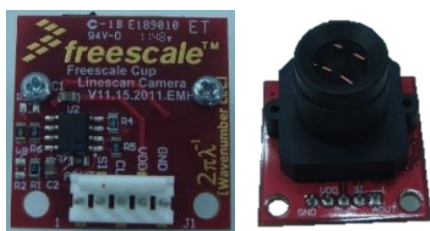


Figura 2 – Câmera linear TSL1401R-LF.

Na estrutura do carro há 3 (três) motores, responsáveis pela locomoção do veículo autômato onde dois motores desenvolvem o papel de propulsores e um motor fica responsável pelo controle de direção (leme). No sistema de propulsão, os motores são controlados por PWM (Pulse Width Modulation – modulação por largura de pulso), compostos por um circuito de controle de corrente que evita a reinicialização total do sistema, caso haja um pico de altas correntes, ilustrado na figura 1. Na figura 3 é apresentado a estrutura do robô móvel, onde será aplicado o estudo em tela.

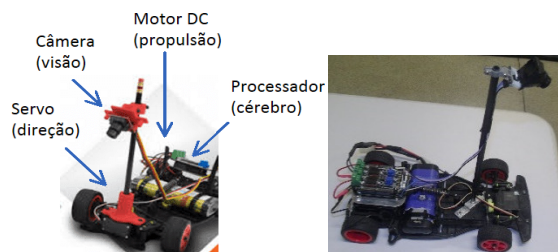


Figura 3 – Robô móvel guiado por câmera linear.

2.2 Sensores

O robô seguidor de linha é guiado através de navegação por referência óptica, ou seja, necessitando de uma boa luminosidade para uma melhor operação. A seguir serão demonstrados alguns dos sensores utilizados para construção de um robô seguidor de linha.

2.2.1 Câmera comum

O sensoriamento do robô pode ser feito a partir de câmeras de vídeo, que requer a utilização de processamento de imagem para que a imagem recebida seja analisada, e assim, possa se tomar uma decisão para controle.

O processamento de imagens é um conjunto de rotinas computacionais que incluem a gravação da imagem, a formatação e correção dos dados digitais, uso de técnicas para melhor visualização e, finalmente, a classificação dos alvos (UFRRJ, 2016). Para realizar o processamento são seguidas as rotinas demonstradas na Figura 4.

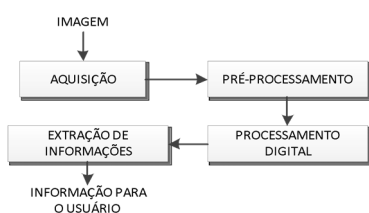


Figura 4 - Rotina para processamento da imagem. Adaptado de UFRRJ, 2016.

A imagem é recebida e armazenada, em seguida é realizada a parte de pré-processamento da imagem que inclui a formatação, remoção de ruídos, correções radiométricas e geométricas e a reamostragem. No processamento digital é feito o registro, a classificação e o realce da imagem. Após estes passos é feita a extração de informações em que é feita a interpretação visual, mapeamento temático, reconhecimento de padrões e a seleção dos atributos, onde as informações serão repassadas ao usuário ou controlador na finalidade de se obter uma melhor decisão de controle.

2.2.2 Câmera comum

Câmeras *Line Scan*, ou câmeras de linha, é uma rede de foto-transistor que utiliza um sistema de captura onde a entrada de dados é apenas uma linha de pixels da imagem real. Sistemas de varredura de linha são melhor empregadas em processamentos de alta velocidade ou aplicações de linha de transporte rápido. Com uma única linha de pixels, a linha de varredura pode construir imagens contínuas não limitadas a uma resolução vertical específica. Isto permite resoluções mais elevadas do que câmeras de varredura de área (Newton, 2016).

Uma desvantagem desse tipo de câmera é que possui uma saída totalmente analógica, o que significa que o usuário tem que ser criativo para processar este sinal, a fim de torná-lo compreensível (Fuentes et al., 2011). A Figura 5 ilustra a forma de onda da câmera de linha.

Como sensor, o robô móvel usa uma câmera *Line Scan*, que possui o campo de visão em formato de um triângulo não perpendicular com o plano a ser navegado. Analisando a vista lateral do robô móvel, a interseção do plano de visão da câmera com o plano a ser navegado, juntamente com uma linha perpendicular ao plano de navegação com origem no centro de visão da câmera, gera um triângulo retângulo. Tais relações geométricas da montagem da câmera pode ser entendidas conforme o diagrama da Figura 6. Onde CAM é o ponto considerado o centro geométrico da câmera (Moreira et al., 2016).



Figura 5 – Representação da visão de uma câmera de linha. Adaptada de (Fuentes et al., 2011).

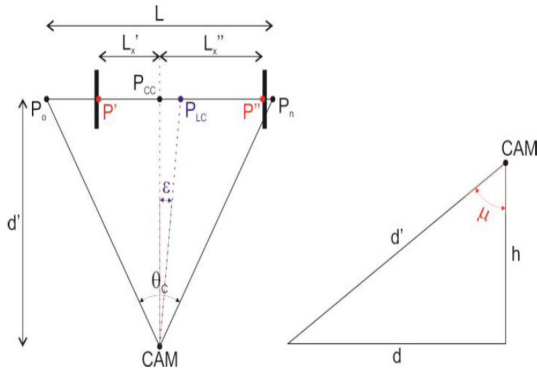


Figura 6 – Diagrama esquemático do campo de visão da câmera. Na esquerda a visão superior e na direita vista de perfil.

Onde d é a distância paralela ao plano da superfície entre a câmera e o ponto de observação da câmera, h é altura da câmera, d' é a distância da câmera até o ponto de observação, no plano da câmera, μ é a angulação do plano de visão da câmera com o plano da superfície, θ_C é o ângulo de abertura da câmera, L é o comprimento que a câmera vê no plano da superfície, P_{CC} é o pixel do centro da câmera, P_{LC} é o pixel do centro da linha detectada, ε é o erro de orientação do robô móvel, P_o e P_n são o pixel inicial e final do campo de visão da câmera, respectivamente, P' e P'' são o primeiro pixel da borda esquerda e direita, respectivamente, considerando a orientação a partir do pixel do centro (P_{CC}). Da Figura 6 tem-se:

$$d = h \tan \mu \quad (01)$$

$$d' = \frac{h}{\cos \mu} \quad (02)$$

$$L = 2d' \tan(\theta_C/2) \quad (03)$$

$$L = \frac{2h \tan(\theta_C/2)}{\cos \mu} \quad (04)$$

Adotando N' e N'' como a posição do *pixels* P' e P'' , respectivamente, e definindo L_{MIN} como o menor comprimento que a câmera pode detectar, como a razão entre o comprimento que a câmera observa e o número de *pixels* da imagem, N_p , e sendo N_C o número do P_{CC} , pode-se derivar o valor de L'_x e L''_x , como segue:

$$\begin{aligned} L'_x &= L \left[1 - \frac{(N_p - N_C)}{N_p} \right] - \sum_{i=0}^{N'} L_{MIN} \Rightarrow \\ &\Rightarrow L'_x = L \frac{N_C}{N_p} - L \frac{(N' - N_o)}{N_p} \Rightarrow \\ &\Rightarrow L'_x = \frac{L}{N_p} (N_C - N') \end{aligned} \quad (05)$$

$$\begin{aligned} L''_x &= L \frac{(N_p - N_C)}{N_p} - \sum_{i=N''}^{N_p} L_{MIN} \Rightarrow \\ &\Rightarrow L''_x = L \frac{(N_p - N_C)}{N_p} - L \frac{(N_p - N'')}{N_p} \Rightarrow \\ &\Rightarrow L''_x = \frac{L}{N_p} (N'' - N_C) \end{aligned} \quad (06)$$

Logo, o erro de orientação pode ser computado de L'_x e L''_x , como segue:

$$L_\varepsilon = L'_x - L''_x = \frac{L}{N_p} (2N_C - N' - N'') \quad (07)$$

$$\begin{aligned} \varepsilon &= \tan^{-1} \frac{L_\varepsilon}{d'} \Rightarrow \\ &\Rightarrow \varepsilon = \tan^{-1} \left[\frac{L \cos \mu}{h N_p} (2N_C - N' - N'') \right] \end{aligned} \quad (08)$$

O valor de ε é então usado como variável de controle usando um controle PID (*Proportional Integral and Derivative*) discreto. A equação geral do PID contínuo pode ser vista na equação (09) e sua discretização nas equações (10), (11) e (12), sendo ε_{Ck} o k -ésimo erro aplicado na correção da orientação do robô e k o atual índice da interação do algoritmo.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (09)$$

$$\varepsilon_{\sigma_k} = \varepsilon_{\sigma_{k-1}} + \varepsilon_k \quad (10)$$

$$\varepsilon_{\delta_k} = \varepsilon_k - \varepsilon_{k-1} \quad (11)$$

$$\varepsilon_{Ck} = K_p \varepsilon_k + K_i \varepsilon_{\sigma_k} + K_d \varepsilon_{\delta_k} \quad (12)$$

O erro de orientação ε_{Ck} foi limitado para estar no intervalo de $[-45^\circ, 45^\circ]$, pois a saída do PID pode estar em um intervalo em que o mecanismo Arckermann não seja capaz de efetuar a correção da orientação, podendo, possivelmente, danificar o referido mecanismo. Considerando que o servo motor é controlado por um PWM de 8 bits, ε_{Ck} deve ter seu valor mapeado por meio de interpolação para o intervalo $[0, 255]$. Considerando a equação da interpolação como:

$$\begin{aligned} \Delta Lim_y &= Lim_{s_y} - Lim_{i_y} \\ \Delta Lim_x &= Lim_{s_x} - Lim_{i_x} \end{aligned} \quad (13)$$

$$Val_y = Lim_{i_y} + (Val_x - Lim_{i_x}) \left(\frac{\Delta Lim_y}{\Delta Lim_x} \right)$$

Onde Val_y e Val_x são os valores dentro da escala y e x , respectivamente, e Lim_{s_y} , Lim_{i_y} , Lim_{s_x} e Lim_{i_x} são os limites superior e inferior da escala y e x , respectivamente e ΔLim_y e ΔLim_x são as variações entre os valores dos limites superior e inferior da escala x e y , respectivamente. Usando $[-45^\circ, 45^\circ]$ como a escala x e $[0, 255]$ como a escala y e ε_{Ck} como o valor a ser interpolado, tem-se que o k -ésimo controle do servo motor u_{Ck} é:

$$u_{Ck} = \text{floor} \left(\left(\frac{127.5}{\pi} \right) (4\varepsilon_{Ck} + \pi) \right) \quad (14)$$

Como u_{Ck} tem que ser um valor inteiro para ser usado como *PWM*, *floor* foi introduzida na equação (14), onde a mesma é responsável por arredondar u_{Ck} para baixo.

2.2.3 Estimativa dos parâmetros do PID para o controle do servo

Os parâmetros K_p , K_i e K_d foram estimados usando o método de Ziegler-Nichols para controle de malha fechada. Este método é o mais usado para determinar os parâmetros do PID. Esta metodologia é caracterizada por dois parâmetros, que são K_{cr} e P_{cr}

(Ogata, 2013). Os mesmos são obtidos da seguinte maneira:

K_i e K_d são inicializados com o valor de 0.

O valor de K_p é incrementado de 0 até um valor crítico ($K_p = K_{cr}$) onde as oscilações começam a ocorrer.

Quando o valor de K_{cr} for estabelecido, a oscilação mantida terá um período de P_{cr} .

Logo usando as equações (08), (10), (11), (12) e (14), o algoritmo do controle de orientação, pode ser descrito abaixo:

1. $\text{ContrServo}(k, L, h, N_c, N_p, \mu, N'_k, N''_k, \varepsilon_{\sigma_{k-1}}, \varepsilon_{\sigma_{k-1}}, K_p, K_i, K_d)$:
2. $\varepsilon_k = \tan^{-1} \left[\frac{L \cos \mu}{h N_p} (2N_c - N'_k - N''_k) \right]$
3. $\varepsilon_{\sigma_k} = \varepsilon_{\sigma_{k-1}} + \varepsilon_k$
4. $\varepsilon_{\delta_k} = \varepsilon_k - \varepsilon_{k-1}$
5. $\varepsilon_{C_k} = K_p \varepsilon_k + K_i \varepsilon_{\sigma_k} + K_d \varepsilon_{\delta_k}$
6. $\varepsilon_{C_k} = \text{restringir}(\varepsilon_{C_k}, -\pi/4, \pi/4)$
7. atribuir ε_{k-1} o valor de ε_k
8. $u_{C_k} = \text{floor} \left(\left(\frac{127.5}{\pi} \right) (4\varepsilon_{C_k} + \pi) \right)$
9. retorna u_{C_k}

2.3 Acionamento e Controle dos motores

Para se obter eficiência no desafio proposto é necessário realizar um controle eficaz dos motores, levando em consideração suas características de funcionamento e atuação. Sendo assim, é importante conhecer as características pertinentes de cada tipo de motor, pois o conhecimento destas características fazem com que se possa implementar um controle adequado, garantindo assim que quando forem acionados, os motores possam responder de maneira satisfatória a cada ação.

2.3.1 Acionamento e Controle dos motores

Para a realização do controle de velocidade dos motores DC (*Direct Current* – corrente contínua), faz-se necessário o conhecimento sobre controle de PWM.

A técnica de PWM ilustrada na Figura 7, é empregada em diversas áreas da eletrônica. Tendo em vista, que o mais comum é para utilização em fontes chaveadas, mas também pode ser utilizada para controle de velocidade de motores, controle de luminosidade, controle de servo-motores e diversas outras aplicações. A frequência do PWM permanece fixa, normalmente em um valor alto suficiente para não afetar o funcionamento do equipamento a ser controlado desse modo a forma de onda resultante não terá variações (Meca Web, 2016).

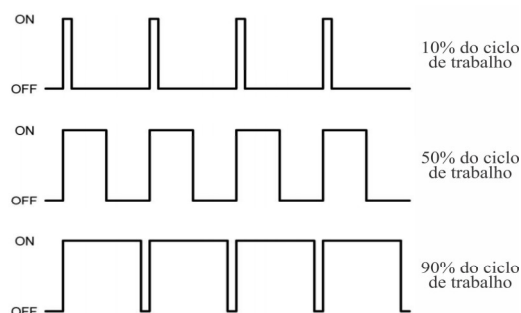


Figura 7 - Exemplos de PWM. (Meca Web, 2017).

2.3.2 Ponte H

As pontes H possuem este nome devido ao formato que é montado no circuito semelhantes à letra H. O circuito utiliza quatro chaves (S1, S2, S3 e S4) que são acionadas de forma alternada, ou seja, (S1-S3) ou (S2-S4). Dependendo da configuração entre as chaves tem-se a corrente percorrendo o motor, hora por um sentido, hora por outro (Cardoso, 2015). Na Figura 8, pode-se visualizar essa afirmação.

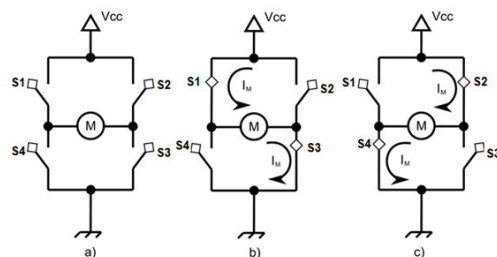


Figura 8 – Circuito ponte H. (CARDOSO, 2015).

Quando nenhum par de chaves está acionado, o motor encontra-se desligado (a). Quando o par S1-S3 é acionando a corrente percorre fazendo com que o motor gire em um sentido (b). Já quando o par S2-S4 é acionado, a corrente percorre por outro caminho fazendo com que o motor gire no sentido contrário (c) (Cardoso, 2015).

O CI L298 presente na *shield* FRDM-TFC na Figura 1, é muito utilizado para o propósito de controle de motores, nada mais é que uma ponte H em um componente integrado. Possuem respostas a sinais PWM, sendo possível regular a tensão de saída, e dessa forma regular a velocidade dos motores (Cardoso, 2015). Esse também é utilizado para alta tensão e alta corrente, projetado para aceitar níveis lógicos TTL padrão e conduzir cargas indutivas tais como relés, solenóides, motores DC e motores de passo. Permite duas entradas para ativar ou desativar o dispositivo, independentemente dos sinais de entrada (St, 2016).

3 Resultados e discussão

Para validação do algoritmo, o mesmo foi implementado no simulador de robôs V-REP, da Coppelia Robotics, na versão EDU (Coppelia Robotics, 2015). Um modelo CAD de robô móvel com a geometria Arckeman (Wu; Xu; Wang, 2013) foi usado com a mesma

estrutura, ou seja, igual ao drive do robô móvel do KIT TFC.

Na simulação, três trajetos foram gerados e o robô móvel foi programado para completá-los no sentido anti-horário. Qualquer um dos sentidos poderia ser utilizado fazendo uso do algoritmo derivado supramencionado, junto com o pré e pós processamento de imagem, já citados. Dos três trajetos, dois são de geometria oval, com a diferença de comprimento entre eles. Já o terceiro, é um trajeto gerado de forma aleatória. Os resultados obtidos das duas primeiras simulações podem ser observados na Figura 9 e os resultados do terceiro trajeto, esse para validação do algoritmo na Figura 10 e 11. Cada segmento do trajeto, ou seja, cada bloco que contém os trechos do trajeto, como podem ser vistos nas Figuras 9, 10 e 11, seguem um padrão de dimensão, seguindo os mesmos padrões quanto à geometria do trajeto estipulado na competição da (Nxp Cup, 2017).

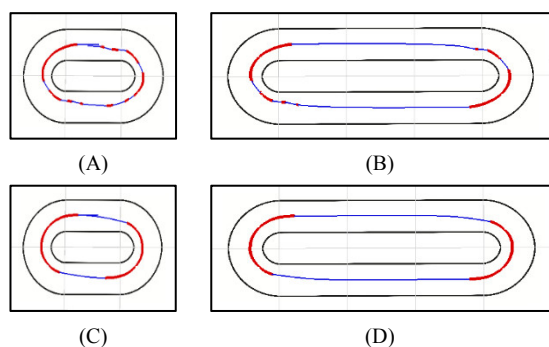


Figura 9: Os dois trajetos simulados para validação do algoritmo, controle *on-off*: (A) e (B); controle PID: (C) e (D).

Na Figura 9 pode-se observar a posição do eixo geométrico do robô simulado sobre a pista pelas linhas azul e vermelha, quando o robô simulado “acredita” estar em uma reta e quando o mesmo “acredita” estar em uma curva, respectivamente. Destas duas simulações, pode-se concluir que o algoritmo respondeu de forma satisfatória. Além disso, observa-se que, às vezes, o sistema não corrige a orientação, como na saída da curva I para a reta que segue na trajetória, ocorrido devido à uma heurística aplicada. A referida heurística consiste em limitar a correção de orientação com o uso de um *threshold*, ou seja, a correção só é feita se o erro de orientação for superior ao valor do *threshold*. A heurística foi usada para evitar que o sistema tente corrigir pequenos erros, evitando-se que a saída do sistema tenha uma oscilação considerável a ponto de tornar-se a navegação autônoma impraticável e otimizando a passagem do robô móvel pela pista de (Nxp Cup, 2017).

Para análise dos dados obtidos das simulações, tais dados foram plotados em gráficos pelo *software* MATLAB® da MathWorks, como pode ser visto na Figura 12. Destes gráficos, vê-se que a resposta nas regiões sombreadas I e II, que é a região em que o sistema está corrigindo o erro de orientação nas respectivas curvas, possui uma resposta rápida, entretanto

com uma pequena oscilação. Entretanto, mesmo com essa pequena oscilação, pode-se observar que a resposta do algoritmo é eficiente, uma vez que a oscilação não vem do algoritmo. Outro detalhe interessante observado na Figura 12 é que no intervalo entre as curvas, a resposta do algoritmo é rápida e precisa, tão logo o erro tenha valor considerável o suficiente para ser corrigido, ou seja, acima do *threshold*, o robô móvel corrige a sua orientação em um pequeno tempo, como pode ser visto entre as regiões I e II.

O erro médio quadrático (EMQ) é a soma das diferenças entre o valor estimado e o valor real dos dados, ponderados pelo número de termos, esse dado foi calculado para mensurar o erro de trajetória para comparação dos dois controles utilizados. Assim os valores do EMQ dos três trajetos são recalculados de forma que os erros com PID sejam frações dos erros com PID no mesmo trajeto, ou seja, o EMQ de um determinado com PID é dividido pelo EMQ sem PID e o EMQ, deste mesmo trajeto, sem PID é definido como 100%. Isto é mestrado na Figura 13. Onde desta pode-se observar que o erro com a utilização do PID proposto foi inferior quando comparado com o controle *on-off*, satisfazendo objetivo deste trabalho.

4 Conclusão

Nesse trabalho foi apresentado à modelagem e implementação de um sistema de visão 1-D para controle de trajetória, utilizando um controlador do tipo PID, para um robô móvel autônomo desenvolvido para a Freescale Cup Intelligent Car Racing mediante simulações computadorizadas. Dos dados obtidos dos testes simulados da modelagem demonstrada nas seções anteriores, pode-se observar que a resposta do controlador PID sintonizado pelo método Ziegler–Nichols é satisfatória. Evidencia-se que o centro geométrico do robô móvel, em um determinado instante, é quase sempre congruente ao centro do trajeto, quando não o é, ou o centro geométrico do robô móvel situa-se perto da borda interna da curva, ou o mesmo se encontra em uma posição próxima ao centro do trajeto.

Futuramente, pretende-se concluir um algoritmo para a estimação dos parâmetros do PID, para o controle dos motores de propulsão, trabalho este já em andamento, porém ainda não finalizado. Além deste algoritmo, há a possibilidade de assistir ao sistema de visão computacional do robô móvel com o uso de Dead-reckoning (Cho et al., 2011), tornando a estimativa de posicionamento do sistema mais precisa, trabalho este também em andamento.

Agradecimentos

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia (IFCE) campus Cedro pela disponibilidade de seus laboratórios onde sem os mesmos, esta pesquisa não seria possível.

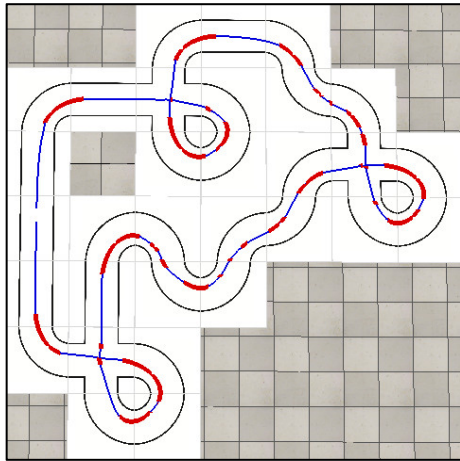


Figura 10: Trajeto do robô móvel utilizando o controle *on-off*.

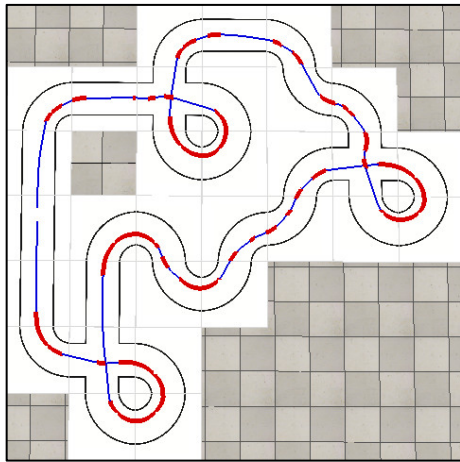
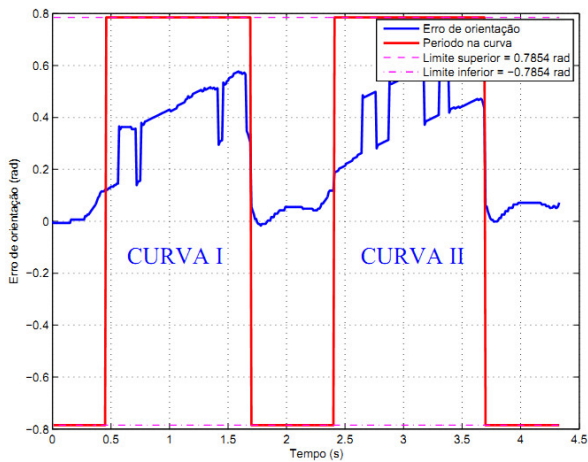
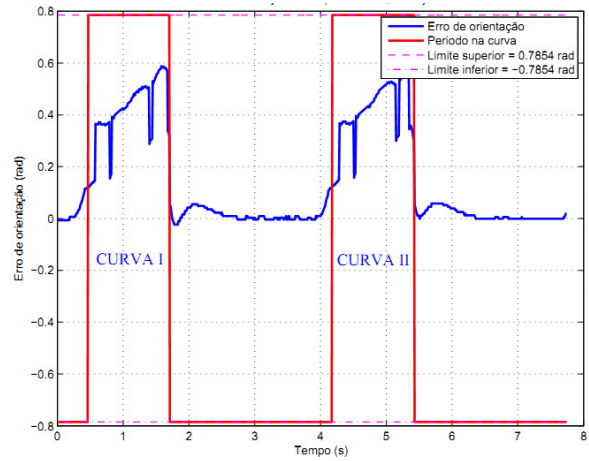


Figura 11: Trajeto simulado para validação do algoritmo utilizando o controle PID projetado.



(A)



(B)

Figura 12: Dois gráficos, de Erro de orientação x tempo, dos dados obtidos das simulações nos trajetos da Figura 9, sendo (A) e (B) o gráfico do trajeto (C) e (D), respectivamente.

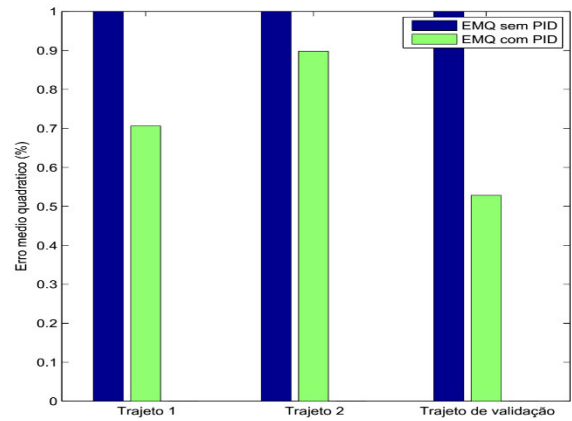


Figura 13: Erro médio quadrático, em porcentagem, dos diferentes controles usados.

Agradecimentos

Agradecemos ao Instituto Federal do Ceará (IFCE) - Campus Cedro e Campus Caucaia, a Universidade Federal do Ceará (UFC) e ao CNPQ pelo apoio e suporte financeiro.

Referências Bibliográficas

- Afonso, P.; Azevedo, J.; Carneira, C.; Cunha, B.; Lima, P.; Santos, V. Challenges and Solutions in an Autonomous Driving Mobile Robot Competition Intelligent Systems. Proc. Control2006, 7th Portuguese Conference in Automatic Control, Lisboa, Portugal, 2006.
- Al-araji, A.; Abbod, M.; Al-raweshidy, H. Design of an adaptive nonlinear pid controller for nonholonomic mobile robot based on posture identifier. In: Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on. [S.l.: s.n.], 2011. p. 337-342.

- Cardoso, Daniel, 2015. Módulo Ponte H L298N – O primeiro passo para montar seu robô com Arduino. Disponível em: <<http://blog.vidadesilicio.com.br/arduino/modulo-ponte-h-l298n-arduino/>>. Acesso em: 17 de jul. de 2016.
- Cho, B.-S.; Moon, W.-S.; Seo, W.-J.; Baek, K.-R. (2011). A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. *Journal of mechanical science and technology*. v. 25, n. 11, pp. 2907-2917.
- Copellia Robotics. Virtual Robot Experimentation Platform – V-REP user manual. Disponível em: <http://coppeliarobotics.com/helpFiles/index.html>. Acesso em: 4 de agosto de 2015.
- Farnell. Disponível em: <<http://www.farnell.com/datasheets/315815.pdf>>. Acesso em: 07 de jul. de 2017.
- Fierro, R.; Lewis, F. Control of a nonholonomic mobile robot using neural networks. In: *Intelligent Control, 1995., Proceedings of the 1995 IEEE International Symposium on*. [S.l.: s.n.], 1995. p. 415–421. ISSN 2158-9860.
- Fuentes, F.R., Trujillo, M., Padilha, C., Mendonza, R., Using Parallax TSL1401-DB Linescan Camera Module for line detection. Freescale Semiconductor, 2011.
- Jang, J. O.; Chung, H. T. Neuro-fuzzy network control for a mobile robot. In: *American Control Conference, 2009. ACC '09*. [S.l.: s.n.], 2009. p. 2928–2933. ISSN 0743-1619.
- Kanayama, Y. et al. A stable tracking control method for an autonomous mobile robot. In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. [S.l.: s.n.], 1990. p. 384–389 vol.1.
- Klancar, G.; Matko, D.; Blazic, S. Mobile robot control on a reference path. In: *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*. [S.l.: s.n.], 2005. p. 1343–1348. ISSN 2158-9860.
- Kolski, S.; Ferguson, D.; Bellino, M.; Siegwart, R. Autonomous Driving in Structured and Unstructured Environments. *Intelligent Vehicles Symposium, 2006 IEEE, Switzerland*, pp. 558-563, June 2006.
- Kuhne, F.; JR, J. M. G. d. S.; LAGES, W. F. Model predictive control of a mobile robot using linearization. In: *4th Mechatronics and Robotics, 2004, Aachen*. [S.l.: s.n.], 2004.
- Liu, C., Chen, J., Xu, Y., Luo, F. Intelligent Vehicle Road Recognition Based on the CMOS Câmera. *IEEE Vehicle Power and Propulsion Conference (VPPC)*, September 3-5, 2008, Harbin, China.
- Meca Web, PWM - Modulação Por Largura de Pulso. Disponível em: <http://www.meca-web.com.br/electronica/content/e_pwm>. Acesso em: 17 de jul. de 2016.
- Moreira, A. F., Xavier, A. R., Silva, J. A. B., Desenvolvimento de um Sistema De Visão 1-D para Controle de um Robô Móvel com Driver Arc-keman para a Freescale Cup Intelligent Car Racing. Instituto Federal de Educação, Ciência e Tecnologia do Ceará, 2016.
- Newton Labs, Comparing Line Scan and Area Scan Technologies. Disponível em: <http://www.newtonlabs.com/line_systems.htm>. Acesso em: 19 de Jul. de 2016.
- NXP CUP. Nxp Community. Disponível em: <https://community.nxp.com/docs/DOC-1284>. Acesso em: 10 de julho de 2017.
- Ogata, K. Engenharia de Controle Moderno – 4ª Edição, 2003, Prentice-Hall.
- Osório, F. S.; Musse, S. R.; Vieira, R; Heinen, M. R. and Paiva, D. C., (2006). “Increasing Reality in Virtual Reality Applications through Physical and Behavioral Simulation. *Proceedings of the Virtual Concept International Conference, Berlin, Springer Verlag*, v. 1, pp. 1-45.
- Pscheidt, E.R., Robô autônomo – Modelo chão de fábrica. UNICENP/NCET, Engenharia da Computação, Curitiba, 2007.
- Robot Worx, What are autonomous robots? Disponível em: <<https://www.robots.com/articles/viewing/what-are-autonomous-robots>>. Acesso em: 14 de jul. de 2016.
- ST Microelectronics, Dual Full-Bridge Driver. Disponível em: <http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/8/L298N.shtml>. Acesso em: 17 de jul. de 2016.
- Thrun, S.; Burgard, W. and Fox, D., (2004). “Probabilistic Robotics”. The MIT Press: Cambridge, London, pp. 667.
- UFRRJ, Processamento de Imagens. Disponível em: <<http://www.ufrj.br/institutos/it/de/acidentes/sr1.htm>>. Acesso em: 15 de jul. de 2016.
- Wolf, D. F.; Simões, E. V.; Osório, F, S. and Trindade Jr., O., (2009). *Robótica Inteligente: Da Simulação às Aplicações no Mundo Real. XXVIII Jornada de Atualização em Informática*, pp 1-51.
- Wu, X.; Xu, M.; Wang, L. (2013). Differential Speed Steering Control for Four-Wheel Independent Driving Electric Vehicle. *2013 IEEE International Symposium on Industrial Electronics (ISIE)*.