Controle Híbrido de um Robô Autônomo Seguidor de Linha: Projeto e Implementação

Fábio Favarim[1], Willian A. Lopes[1], Marcio Luis Petry[1], César R. Claure Torrico[2]

[1]Departamento Acadêmico de Informática (DAINF)
[2]Departamento Acadêmico de Engenharia Elétrica (DAELE)

Universidade Tecnológica Federal do Paraná (UTFPR)
Via do Conhecimento, KM-1, 85503-390, Pato Branco – PR

{favarim,cesartorrico}@utfpr.edu.br, {walopes23,petrymarcio@gmail.com}

Abstract. *This work describes the development of an autonomous line follower robot. It is presented the modeling and the implementation of a hybrid control, which is composed by continuous and discrete dynamics, that will be, respectively, the Proportional-Integral-Derivative and the Discrete Event Systems controllers. The discrete event control was modeled with Moore automata. Line followers robots were developed to test the developed hybrid control and practical tests were carried out on a test track, as well as in a competition at the national level.*

Resumo. *Este trabalho descreve o desenvolvimento de um robô seguidor de linha autônomo. É apresentada a modelagem e a implementação de um controle híbrido, composto por dinâmicas contínuas e discretas, que são, respectivamente, os controladores Proporcional-Integral-Derivativo e Sistemas a Eventos Discretos. O controle a eventos discretos foi modelado com os autômatos de Moore. Robôs seguidores de linha foram desenvolvidos para testar o controle híbrido desenvolvido e testes práticos foram realizados em uma pista de testes e em uma competição em nível nacional.*

1. Introdução

A robótica é uma das áreas mais promissoras da engenharia, tendo aplicabilidade em várias áreas: de domésticas a aeroespaciais. Um dos ramos da robótica, são os robôs móveis os quais vem se destacando na atualidade no âmbito industrial, automotivo, doméstico, exploração espacial, lazer, dentre outros. Como exemplos, carros autônomos são projetados para circularem em circuitos urbanos, nas aplicações residenciais destacam-se os aspiradores de pó autônomos.

Vários centros de pesquisa têm proporcionado à robótica móvel grande avanço no desenvolvimento de robôs cada vez mais eficientes. A ênfase dessas pesquisas está relacionada com problemas como a operação (movimentação) em ambientes que se modificam dinamicamente, compostos por obstáculos móveis e estáticos. Para operar neste tipo de ambiente é necessário que o robô tenha autonomia/inteligência e ser capaz de adquirir e utilizar conhecimento sobre o ambiente, estimar uma posição, reconhecer obstáculos e responder em tempo real. A percepção do ambiente de um robô móvel se dá a partir das informações colhidas de um conjunto de sensores presentes no robô, e um dos principais desafios da robótica móvel é implementar a lógica de locomoção autônoma baseada nessas informações [Romero, 2015].

Com o intuito de promover e divulgar o desenvolvimento e avanço tecnológico em robótica móvel e incentivar estudantes de engenharia e áreas afins a desenvolver

conhecimento por esse tipo de aplicações surgiram várias competições, como a Winter Challenge e a RoboCup que propiciam competições e desafios em diferentes categorias. A Winter Challenge é uma das maiores competições de robótica móvel da América Latina, contando na Edição de 2018 com a participação de vários países e tendo cerca de 1300 pessoas e 540 robôs inscritos.

Em meados de 2015, foi criada uma equipe de robótica em uma universidade federal, composta inicialmente por alunos do curso de Engenharia de Computação, com o objetivo de participar dessas competições. Um dos robôs que tem sido desenvolvido é o seguidor de linha (*line follower*), que é foco deste trabalho. Essa categoria é composta por robôs completamente autônomos que devem realizar um percurso, indicado por uma linha, no menor tempo possível. A superfície da pista é plana, feita com uma ou mais placas de MDF revestidas com manta de borracha preta ou branca, e a linha tem a cor branca ou preta. Pode ser empregado qualquer método de controle, desde que esteja completamente contido no robô e que não haja interação com um sistema de controle externo (humano ou máquina). Em geral, a técnica de controle usada é a Proporcional-Integral-Derivativo, mais conhecida por PID [Ogata 2010].

Um robô seguidor de linha, além de continuamente ter que manter-se sobre uma faixa de orientação, deve reconhecer marcas (eventos discretos) indicando algumas informações do percurso, por exemplo, marca de início de trajeto, cruzamentos, curvas. Esse sistema com variáveis contínuas e eventos discretos interagindo entre si consiste de um sistema híbrido. Neste sentido, os robôs projetados e implementados sempre levaram em consideração um sistema de controle híbrido, o qual consiste de um controlador PID, de variável contínua, e outro de Sistemas a Eventos Discretos (SED). Os SEDs são sistemas descritos por um espaço de estados discreto com transições de dados discretas orientadas a eventos [Cassandras e Lafortune 2008]. Assim, a principal contribuição deste trabalho é a modelagem de um SED em conjunto com um controlador PID aplicado a robôs seguidores de linha.

2. Sistemas de Controle de Robôs Móveis

Para que um robô seja autônomo, é necessário que este apresente uma resposta desejada para as mais diversas situações. Para tanto se utiliza de um sistema de controle, que consiste em subsistemas e processos, conhecidos como plantas, dos quais se obtém uma saída com desempenho desejado para uma dada entrada.

Essa seção descreve as duas abordagens de controle utilizadas neste trabalho. Inicialmente é apresentado o controle PID por ser amplamente utilizado no controle de sistemas contínuos, seguido por uma descrição sobre os Sistemas a Eventos Discretos (SED).

2.1. Controle Proporcional, Integral e Derivativo

O Sistema de Controle PID (Proporcional Integral Derivativo) é uma das técnicas mais empregadas quando se deseja realizar o controle de variáveis contínuas, sendo amplamente usado em sistemas de controle industrial devido ao seu desempenho robusto e simplicidade funcional. O controle PID consiste em um algoritmo matemático composto por três coeficientes [Ogata, 2010]: proporcional (*P*), integral (*I*) e derivativo (*D*), que são variados para obter a resposta ideal conforme Equação 1.

$$u(t) = VM(t) = Kp.e(t) + Ki.\int_0^t e(\tau).\mathrm{d}\tau + Kd.\frac{de(t)}{dt} \qquad \text{Eq. (1)}$$

Sendo que $u(t)$ é a saída do controlador em relação ao tempo, conhecida também como variável manipulada ($VM(t)$). A parte proporcional do algoritmo é calculada por uma

multiplicação entre o erro $e(t)$ e o ganho proporcional Kp. O erro $e(t)$ é gerado pela diferença entre os valores de saída e entrada. A parte integradora consiste no cálculo de uma integral do erro afetado por uma constante, ou seja, numericamente ela realiza uma soma do valor anterior da integral com o erro e tudo isso multiplicado pelo ganho integral Ki. A parte derivativa consiste por uma derivada, que numericamente é a subtração entre o valor do erro anterior e o erro atual, este valor é multiplicado ganho derivativo Kd. Na Figura 1 é representado um sistema a Malha Fechada com controle PID com uma entrada de referência $ref(t)$ e uma saída $y(t)$.

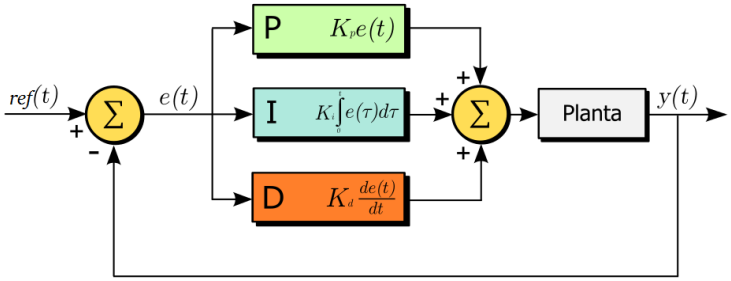


Figura 1 – Diagrama de Blocos de um sistema de controle PID.

2.2. Sistemas a Eventos Discretos

Segundo Cury (2011), Sistemas a Eventos Discretos (SED) podem ser definidos como um sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempo em geral irregulares e desconhecidos. Eventos como o início e o término de uma tarefa e a percepção de uma mudança de estado de um sensor são por natureza instantâneos e tem um caráter discreto no tempo. Para modelar tais sistemas vários modelos foram desenvolvidos como: Redes de Petri, Cadeias de Markov, Teoria das Filas, Teoria de Linguagem e autômatos, etc.

Cassandras e Lafortune (2008) definem um autômato como um dispositivo apropriado para representar uma linguagem de acordo com regras bem definidas. Caso o conjunto de estados do autômato seja finito e caso este seja determinístico, ou seja, que este não possua duas ou mais transições com o mesmo nome saindo de um estado, o autômato é chamado de Autômato Finito Determinístico (AFD). Um SED modelado por um AFD pode ser representado formalmente por $G = (X, \Sigma, f, x_0, X_m)$ em que X é o conjunto de estados do autômato, Σ é o conjunto de símbolos que formam um alfabeto de entrada, $f: X \times \Sigma \rightarrow X$ é a função de mapeamento (ou função de transição), x_0 é o estado inicial e X_m é o conjunto de estados marcados ou finais, $X_m \subseteq X$. Um autômato pode ser representado graficamente como um grafo dirigido, em que os nós representam os estados e os arcos etiquetados representam as transições entre os estados. O estado inicial é identificado através de uma seta apontando para ele e os estados finais são representados com círculos duplos [Cury, 2011].

Segundo [Ramadge and Wonham 1989], para associar a um SED estruturas de controle, particiona-se o alfabeto de eventos Σ em $\Sigma_c \cup \Sigma_u$, em que Σ_c é o alfabeto de eventos controláveis e Σ_u é o alfabeto de eventos não controláveis. Pode-se caracterizar os eventos não controláveis como sendo os eventos em que não se pode evitar sua ocorrência, enquanto que os eventos controláveis são possíveis de serem evitados pelo agente de controle.

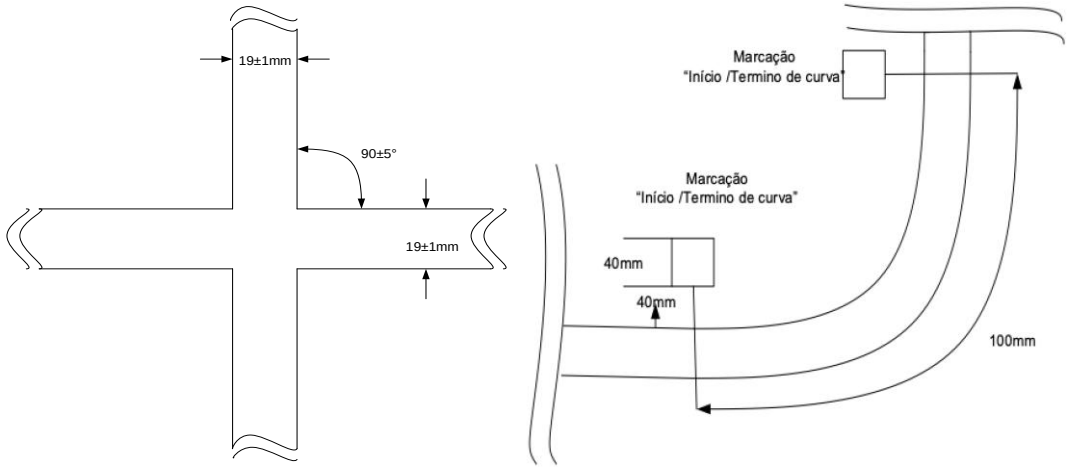
3. Modelagem e Controle do Robô Seguidor de Linha

Nesta seção é apresentada a modelagem do comportamento do robô e as suas especificações gerais. Antes disso, é detalhado o percurso o qual o robô deve percorrer.

3.1. Especificação do Percurso

O robô autônomo deve percorrer um circuito feito com uma ou mais placas de MDF revestidas com borracha preta, tomando como referência uma linha branca de 19±1 mm de largura. O comprimento total da linha é de no máximo 60 metros. Vence o robô que finalizar o trajeto em menor tempo. O corpo do robô deve sempre ficar sobre a linha. Caso o robô saia completamente de cima da linha branca, será considerado que o robô saiu do percurso, assim sendo desclassificado. A linha consiste em combinações de retas e arcos, podendo cruzar sobre ela mesma. Quando houver um cruzamento, o ângulo de intersecção das linhas será de 90±5°, conforme ilustrado na Figura 2(a). As partes das linhas que formam a intersecção serão retas com comprimento mínimo de 250 mm antes e depois do cruzamento. O raio dos arcos é de pelo menos 100 mm. Cada arco possui um comprimento mínimo de 100 mm, conforme ilustrado na Figura 2(b). Haverá uma marcação no lado esquerdo da linha (em relação ao sentido do percurso) no ponto em que houver alteração da curvatura.

Há também marcações do lado direito da linha do percurso, de modo a identificar o início/fim. A marcação que indica o fim de pista está localizada a um metro antes da marcação de início.



(a) **(b)**

Figura 2 – Exemplo do percurso: (a) Cruzamento; (b) Curva.

3.2. Especificação do Robô Seguidor de Linha

Os robôs seguidores de linha devem ser totalmente autônomos e com todos os componentes embarcados. Não pode ser controlado externamente por fio ou por rádio, com exceção para ser iniciado. Nenhuma adição, remoção ou alteração de hardware ou software pode ser feitas durante a competição. O robô não pode exceder 250 mm de comprimento, 250 mm de largura e 200 mm de altura. O robô não pode possuir nenhum tipo de mecanismo de sucção para aumentar a força normal em relação ao solo.

Os robôs seguidores de linha desenvolvidos são compostos por uma barra composta por sensores de refletância dispostos na frente do robô para detecção do trajeto e dos cruzamentos e por sensores de refletância situados na frente das rodas para detecção das marcações de início/fim de curva e início/fim de pista. A locomoção do robô é efetuada por 2 motores de corrente continua. Toda a lógica de funcionamento, isto é, o controle em si é embarcado em um microcontrolador.

3.3. Modelagem do SED

O software utilizado para modelagem o SED foi o “Supremica” [Malik et al. 2017]. Os eventos que ocorrem no sistema foram classificados a partir do ponto de vista do

microcontrolador. Assim, os eventos gerados a partir dos sensores foram definidos como não controláveis e os eventos de comandos para acionamento dos motores foram considerados controláveis, já que é o microcontrolador que dispara esses eventos. As variáveis para desenvolvimento da lógica de controle a eventos discretos vêm dos dois sensores laterais (marcações do lado direito da pista são início/fim da pista, marcas do lado esquerdo são início/fim de curvas) e um botão de partida e calibração dos sensores.

A modelagem do SED do seguidor de linha foi projetada visando atender as seguintes especificações:

- Um botão é pressionado para calibrar os sensores e para iniciar a corrida;
- Nas retas o robô anda a 100% da velocidade máxima;
- Nas curvas o robô reduz para 80% da velocidade máxima;
- O número de curvas e de cruzamentos do trajeto não é conhecido.

Os eventos identificados para modelagem foram definidos como:

- ED: Encontra marca da direita;
- PD: Perde marca da direita;
- EE: Encontra marca da esquerda;
- PE: Perde marca da esquerda;
- Aperta_bot: Botão de calibragem sensores e iniciar corrida.

Com base nessas especificações e eventos, obteve-se modelo apresentado na Figura 3. Vale ressaltar que o modelo criado é genérico e modela a maioria das pistas de competição para seguidores de linha que apresentem marcações laterais. Os valores dentro dos estados correspondem à velocidade do robô, sendo nas retas (100%), curvas (80%) e robô parado (estado inicial) (0%). Este valor de velocidade deverá ser ajustado para que o robô percorra a linha o mais rápido possível sem sair do trajeto.

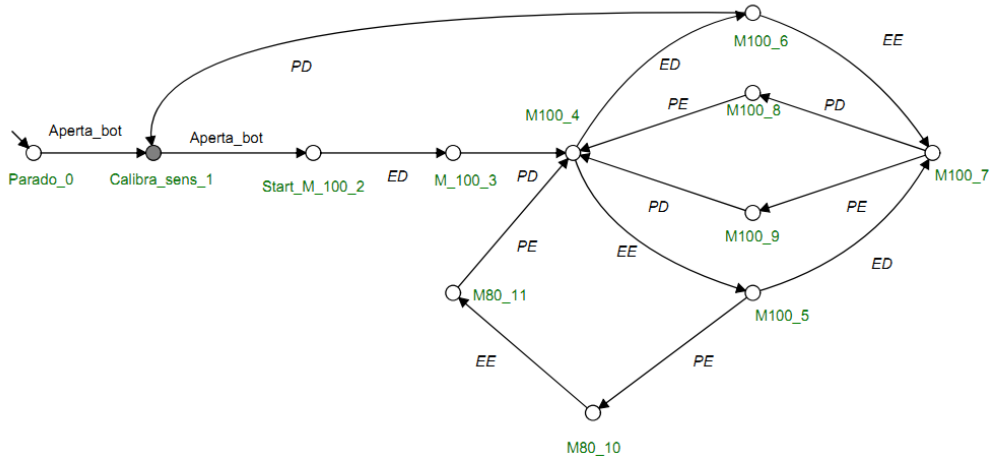


Figura 3 – Modelagem do Sistema a Eventos Discretos do Robô Seguidor de Linha

Uma breve descrição de cada estado do robô é apresentada a seguir:

- Estado 0: Robô ligado, esperando o botão ser pressionado para iniciar a calibragem e permanecer parado;
- Estado 1: Robô ligado e parado, esperando o botão ser pressionado para iniciar o percurso;

- Estado 2: Robô está percorrendo a área de chegada e partida com os motores no ponto de operação para retas (PWM=100%), próximo evento será achar a marca da direita (ED);
- Estado 3: Encontrou a linha branca de início de percurso, agora ele irá "perder" a linha (PD);
- Estado 4: Continua na reta e espera o próximo evento que pode ser encontrar a marca da esquerda, como também achar a marca da direita;
- Estado 5: Encontrou a marca da esquerda, e espera o próximo evento, que pode ser perder a esquerda, que significa que vai entrar na curva, ou achar a direita, que significa que é um cruzamento;
- Estado 6: Achou a marca da direita, e em seguida achar a marca da esquerda (vai para estado 7), que diz que entrou em um cruzamento, ou perder a direita, que significa que chegou a faixa de fim de percurso, retornando para o estado 1 (espera de novo evento do Botão).
- Estado 7: Achou a marca da direita após ter achado a marca da esquerda, está em um cruzamento, ponto de operação não muda, espera perder a marca da direita ou da esquerda para mudar de estado;
- Estado 8 e Estado 9: Perdeu a marca da direita ou esquerda, está deixando o cruzamento, irá perder novamente uma das duas marcas e ira retornar ao estado 4, não muda o ponto de operação.

  Observação: As transições entre o estado 7- 8 ou 7- 9 e depois o 4 ocorrem da seguinte maneira, quando o robô ira deixar o cruzamento ele irá perder primeiro uma faixa e depois a outra (Exemplo: do estado 7 ele perde a direita vai para o 8, em seguida perde o da esquerda, vai para o estado 4, ou seja, saiu do cruzamento).
- Estado 10: Perde a marca da esquerda, então ele entrou na curva, muda o ponto de operação (PWM=80%). Espera o próximo evento que é achar a marca da esquerda, que diz que vai terminar a curva;
- Estado 11: Achou a marca da esquerda, está se alinhando para entrar novamente na reta, por isso o valor do PWM não muda ainda. Só irá mudar para o ponto de operação depois que perder a marca da esquerda e retornar para o estado 4 (PWM=100%);

3.2. Modelagem do Controlador PID

Foi utilizado o sistema de malha fechada, em que o valor da saída é dependente do sinal de entrada e do sinal de referência. No projeto do controlador PID, a variável a ser controlada é *posição*, a qual é referente a posição do robô sobre a linha branca que define a trajetória do percurso. O sinal de referência é o valor da posição do sensor do meio da barra frontal de sensores, quando posicionado sobre a linha branca, enquanto que os outros estão fora da linha (sobre a parte preta).

A *posição* é obtida ao se multiplicar o valor de cada sensor por um número, múltiplo de 1000 e que corresponde a sua posição na barra e dividir esse valor obtido pela soma dos valores de cada elemento. Esse procedimento pode ser melhor compreendido pelas Equações 2 e 3, considerando uma barra com cinco sensores. Nessas Equações, n corresponde à posição de cada sensor (quanto maior o n, mais a direita está localizado o componente) e V_n ao valor do respectivo sensor (com valores possíveis de 0 a 4095, pois a resolução do conversor A/D é de 12 bits). Um valor de retorno 0 indica que a linha está diretamente abaixo do sensor 0, um valor de retorno

1000 indica que a linha está diretamente abaixo do sensor 1, etc. Valores intermediários indicam que a linha está entre dois sensores. Aproximando-se os valores lidos como 0 e 4095 para os sensores em cima da linha branca e na parte preta, respectivamente, e com base na Equação 2, tem-se que a posição de referência para o controlador é 2000, a qual representa a barra de sensores no meio da linha branca.

$$posicao = \frac{\sum_{n=1}^{5} 1000(n-1)V_n}{\sum_{n=1}^{5} V_n} \quad \text{Eq.(2)}$$

$$posicao = \frac{0*V_1+1000*V_2+2000*V_3+3000*V_4+4000*V_5}{V_1+V_2+V_3+V_4+V_5} \quad \text{Eq. (3)}$$

O erro é computado pela posição estimada com a Eq. (2) subtraindo a referência (2000). A derivada do erro é obtida fazendo uma subtração do valor atual de erro subtraído do erro da amostra anterior. A integral do erro é calculado pelo mesmo princípio numérico através da área embaixo da curva a cada período de amostragem. Esse cálculo é periódico, visto que o valor dos sensores é lido somente a cada interrupção do *timer* do microcontrolador. A partir do valor da posição, é possível saber o erro do protótipo: se for maior do que 0, o carro está mais para a direita da curva; se for menor do que 0, o carro está mais para a esquerda; se for 0, então o veículo está exatamente no meio da linha.

A ação de controle é um sinal *PWM (Pulse Wide Modulation)*, o qual multiplica as respectivas constantes, de forma a obter a saída do PWM, conforme apresentado na Equação 4.

$$PWM = (proporcional * Kp) + (derivativo * Kd) + (integral * Ki) \quad \text{Eq.(4)}$$

A cada novo cálculo do controlador PID um novo *duty cycle* é aplicado aos motores. Caso a saída do PWM seja negativa, o *duty cycle* aplicado ao motor da direita é aumentado e o da esquerda se mantém; caso PWM seja positivo, então o motor da direita se mantém e o da esquerda é reduzido.

4. Experimentos e Resultados

Durante o desenvolvimento deste trabalho foram confeccionados três protótipos para testar os sensores e sistemas de controle desenvolvidos. O desenvolvimento do hardware dos protótipos foi dividido em duas partes: o chassi do robô, no qual estão embarcados os principais componentes do sistema, como o microcontrolador, motores e o driver de acionamento destes, os sensores de marcação laterais, situados na frente das rodas; e a barra frontal de sensores, a qual contém os sensores de refletância para a detecção da linha. Os protótipos desenvolvidos eram testados em pista de testes e também participaram de competições de nível nacional e regional. A pista de testes utilizada segue os padrões apresentados na Seção 3.1. A pista de testes desenvolvida é apresentada na Figura 4.

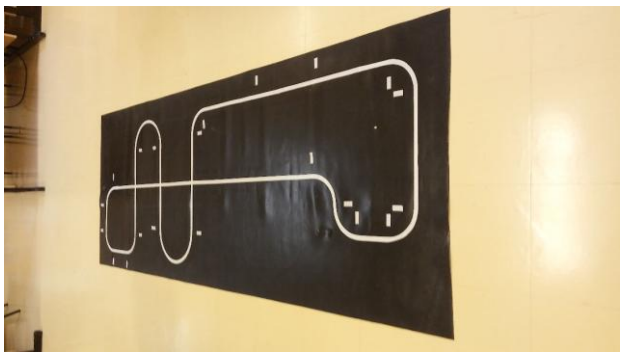
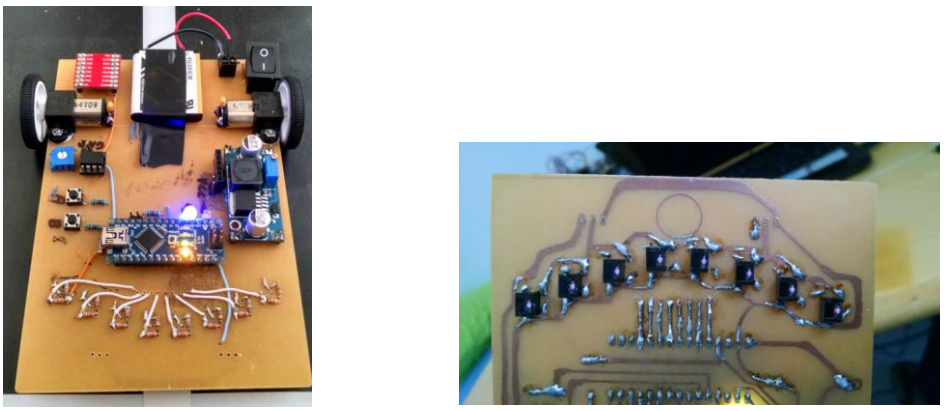


Figura 4 – Pista de Testes

4.1 Protótipo 1

Em meados de 2015 a primeira versão de um robô seguidor de linha foi desenvolvida, tendo como base uma placa de desenvolvimento Arduino Nano com microcontrolador Atmega328p e oscilador de 16MHz. Foi optado pelo Arduino devido a equipe ser nova e pela facilidade de implementação e uso de bibliotecas prontas para testes dos sensores. A Figura 5 (a) apresenta uma imagem do protótipo desenvolvido. Nessa versão apenas foi implementado o controle PID. Para guiar o robô autônomo sobre a linha é necessária uma leitura de posição do protótipo sobre a linha. Para isto foram testados diferentes tipos de estruturas e encapsulados de sensores de refletância. Na primeira versão do protótipo foram utilizados 8 sensores de refletância QRE1114 conforme a Figura 5 (b). Essa configuração teve problemas ao percorrer as curvas do circuito, pois os sensores mais externos eram afetados pelas marcações de início e final de curva, causando instabilidade e a perda da linha. Além disso um número maior de sensores afeta o tempo de processamento (leitura, calibração, cálculo de posição e atuação nos motores).



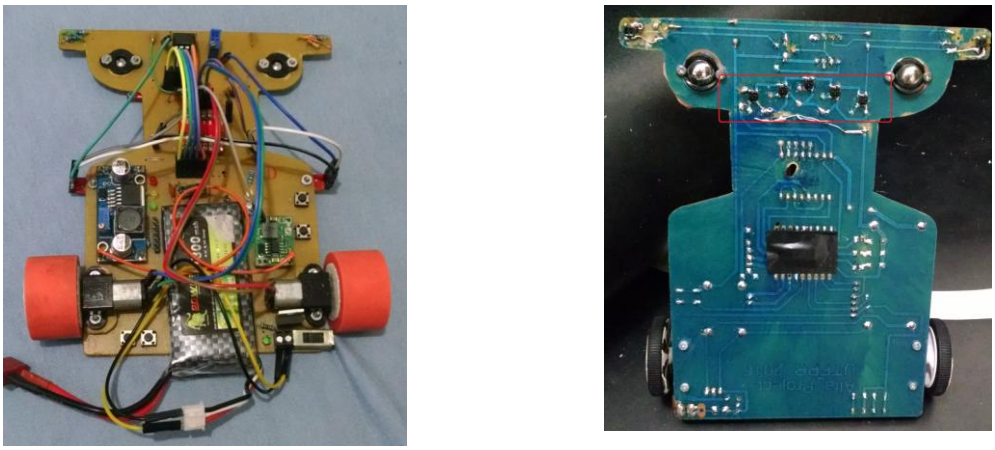
(a) **(b)**

Figura 5 - Versão 1 do protótipo desenvolvido: (a) Rôbo; (b) Detalhe dos Sensores

Essa versão do protótipo foi testada com o controle PID em na competição WinterChallenge e SummerChallenge de 2015, percorrendo a linha numa velocidade média de 0,85 cm/s.

4.2 Protótipo 2

Na segunda versão (Figura 6 (a)), desenvolvida em 2016, foi utilizado o microcontrolador MSP430G2553 da Texas com uma barra de sensores frontal contendo 5 sensores de refletância SMD GP2S700HCP conforme apresentado na Figura 6(b).



(a) **(b)**

Figura 6 - Versão 2 do protótipo desenvolvido: (a) Robô; (b) Detalhe dos Sensores

Apesar do microcontrolador da Texas também trabalhar na frequência de 16MHz sua arquitetura permite a leitura e conversão de todos os sensores em paralelo com poucos ciclos de *clock*, obtendo a posição do robô sobre a linha 4600 vezes por segundo. Nessa versão foi implementado o controle do sistema a eventos discretos. Conforme mencionado na Seção 3.1 as variáveis discretas são provenientes de um botão e dois sensores laterais. Foram utilizados sensores de refletância QTR-1A para detecção das marcações laterais. O botão (*push button*) foi ligado entre o Vcc (3,6V) e microcontrolador, ao ser pressionado gera uma borda de subida (nível lógico alto) dando início ao processo de calibração dos sensores, pressionado mais uma vez o robô inicia o percurso da linha. Ao detectar as marcas de sinalização na pista os sensores laterais apresentam o sinal em azul (3,68V no preto e 1,92V ao passar pela faixa branca) da Figura 7. Esse sinal foi tratado por um circuito comparador para gerar os eventos discretos (nível lógico alto (2,48V) e baixo 0,4V) em amarelo. O circuito comparador apresentado na Figura 7(b) foi implementado com um amplificador operacional LM358N e um potenciômetro para ajuste da tensão de referência em 3V. A saída é ligada ao microcontrolador que gera uma interrupção toda vez que uma borda (subida ou descida) é detectada.

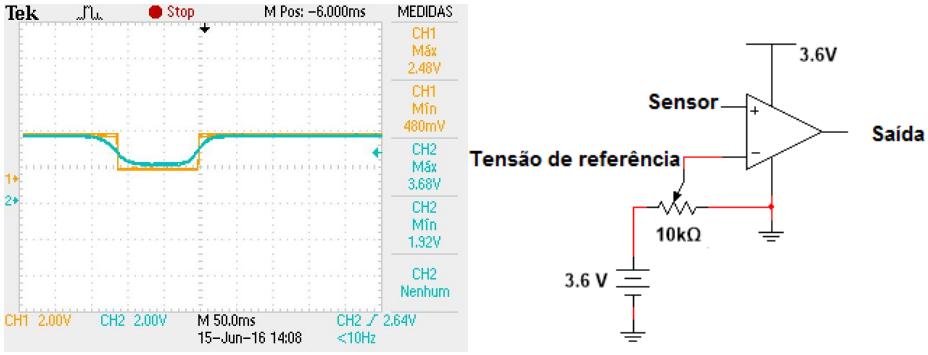


Figura 7 – Condicionamento do sinal do sensor lateral: (a) medição osciloscópio; (b) circuito comparador

Nos testes realizados na pista de testes o robô conseguiu identificar todas as marcações de curvas, cruzamentos e sinal de início e fim de percurso comprovando o perfeito funcionamento do controle discreto. Com o controle híbrido esse protótipo conseguiu seguir a linha em velocidade média 1,05m/s podendo acelerar na potência máxima dos motores nas retas. Vale ressaltar que foram usados os mesmos motores que a versão anterior. Essa versão competiu na edição de 2016 da Winter Challenge. Uma limitação observada nessa versão foi o desempenho variável do protótipo a medida que a carga da bateria era reduzida. Isto é, conforme a bateria se descarrega, a tensão que com que os motores são alimentados também reduzia.

4.3 Protótipo 3

As principais mudanças ocorridas em relação à versão anterior foram a habilitação da inversão de giro do motores, a adição de um circuito regulador de tensão conhecido como *step-up* e de dois transistores MOSFET, assim como a substituição do microcontrolador. A inversão de giro dos motores tem a sua principal vantagem nas curvas. Quando está em uma curva, o controlador pode transmitir um PWM negativo a um dos motores. Caso isso ocorra, o motor tem o seu sentido de rotação invertido, o que não necessariamente faz com que ele comece a girar no sentido oposto, mas que possa funcionar como um freio, permitindo assim que o veículo saia das curvas mais rapidamente. O circuito de *step-up* foi adicionado à placa de modo a manter constante a tensão aplicada sobre os motores, permitindo manter o desempenho. Os transistores MOSFET foram adicionados para garantir a segurança do circuito: o primeiro foi

utilizado de forma a evitar que seja aplicada uma tensão invertida na placa, o que pode ocorrer caso os fios de alimentação sejam trocados; o segundo MOSFET localizado entre a alimentação positiva da bateria e a alimentação do *driver* de acionamento dos motores, impede que alguma corrente reversa dos motores ou do próprio *driver* possa retornar para a placa, o que poderia ocasionar danos nos componentes do protótipo. Para essa versão foi utilizado *kit* de desenvolvimento NUCLEO-F303K8, da ST Microelectronics, com microcontrolador STM32F303K8 e processador ARM Cortex-M4 de 32 *bits* e 72MHz de *clock*, permitindo realizar o processamento em maior frequência, assim permitindo melhor controle sobre o robô. A Figura 8 mostra o terceiro protótipo desenvolvido.

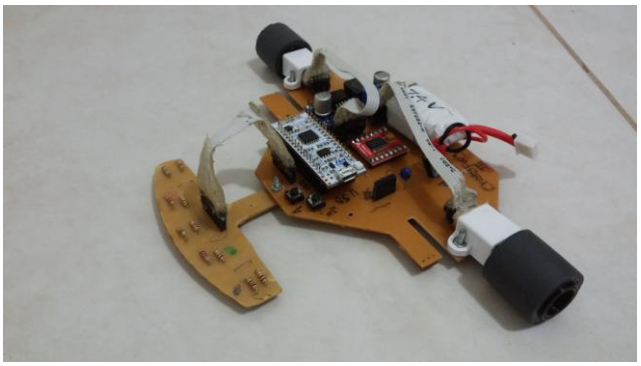


Figura 8 - Versão 3 do protótipo desenvolvido

5. Conclusões

O objetivo deste trabalho foi projetar e modelar um sistema de controle híbrido, composto por um controlador de tempo contínuo e um controlador de tempo discreto, aplicado a um robô autônomo seguidor de linha. A implementação do controle híbrido realizado a partir da modelagem dos eventos discretos representados por autômatos apresentou o resultado esperado, detectando todas as marcas na pista facilitando o controle da posição do robô sobre a linha. Três protótipos foram desenvolvidos visando melhorar os aspectos de componentes utilizados e a partir de problemas encontrados, novas versões foram criadas. Os resultados obtidos nos testes em pista de testes e em eventos de competição mostraram a efetividade do controle híbrido. Como trabalhos futuros, espera-se explorar a terceira versão do protótipo desenvolvido, utilizando o potencial do maior poder computacional do microcontrolador com ponto flutuante, assim como realizar mapeamento da pista, permitindo melhor desempenho do robô para percorrer a pista.

Referências

Cassandras, C. G. e Lafortune, S. (2008) "*Introduction to Discrete Event Systems"*, Springer US, New York, 2nd edition.

Cury, J. E. R. (2001) "Teoria de controle supervisório de sistemas a eventos discretos". In: V Simpóstio Brasileiro de Automação Inteligente (SBAI).

Malik,R., Akesson,K., Flordal, H., e Fabian, M. (2017) "Supremica, an efficient tool for large-scale discrete event systems". *IFAC-PapersOnLine*, 50(1):5794–5799.

Ogata, K. (2010) "Engenharia de Controle Moderno". 5. ed. São Paulo: Pearson Prentice Hall, 2010.

Ramadge, P. J. e Wonham, W. M. (1982). Supervision of discrete event processes. In *1982 21st IEEE Conference on Decision and Control*, pages 1228–1229. IEEE.

Romero, R. (2014) "Robótica Móvel". 1. ed. São Paulo: LTC, 2014.