

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**DEPARTAMENTO ACADÊMICO DE ELÉTRICA**  
**CURSO DE ENGENHARIA ELÉTRICA**

**ALAN JUNIOR GUADAGNIN**

**CONTROLE HÍBRIDO DE UM ROBÔ SEGUIDOR DE LINHA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PATO BRANCO**

**2014**

ALAN JUNIOR GUADAGNIN

## **CONTROLE HÍBRIDO DE UM ROBÔ SEGUIDOR DE LINHA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia Elétrica da Coordenação de Engenharia Elétrica – COELT – da Universidade Tecnológica Federal do Paraná – UTFPR, Campus Pato Branco, como requisito parcial para obtenção do título de Engenheiro.

Orientador: Prof. Dr. César Rafael Claire Torrico

PATO BRANCO

2014

## **TERMO DE APROVAÇÃO**

O trabalho de Conclusão de Curso intitulado **CONTROLE HÍBRIDO DE UM ROBÔ SEGUIDOR DE LINHA**, do aluno **ALAN JUNIOR GUADAGNIN** foi considerado **APROVADO** de acordo com a ata da banca examinadora N° **61** de 2014.

Fizeram parte da banca os professores:

**César Rafael Claire Torrico**

**Fernando José Avancini Schenatto**

**Emerson Giovani Carati**

## DEDICATÓRIA

Dedico este trabalho ao meus pais, pelo apoio e incentivo durante a graduação.

## **AGRADECIMENTOS**

Agradeço ao professor orientador César Rafael Claire Torrico pelos auxílios prestados para o desenvolvimento deste trabalho.

Gostaria de agradecer ao meu colega Eulino Silveira pela elaboração da parte mecânica no robô, e aos demais colegas acadêmicos por darem força e motivação para finalizar este trabalho.

## EPÍGRAFE

É muito melhor lançar-se em busca de conquistas grandiosas, mesmo expondo-se ao fracasso, do que alinhar-se com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem numa penumbra cinzenta, onde não conhecem nem vitória, nem derrota. (Theodore Roosevelt)

## RESUMO

GUADAGNIN, Alan Junior. Controle Híbrido de um Robô Seguidor de Linha. 2014. 65 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

Este trabalho apresenta o desenvolvimento da modelagem e implementação do controle híbrido (de tempo contínuo e a eventos discretos) em um robô seguidor de linha. Inicialmente são apresentados os métodos formais de controle contínuo (controladores P, I e D) e também a eventos discretos (autômatos de Moore e Mealy) necessários para a realização do trabalho. No desenvolvimento são utilizados os métodos para condicionamento de sinal, fazendo-se a modelagem experimental do processo, é utilizado o software Simulink para simulação do modelo matemático e testes para a implementação de um controlador Proporcional, o qual foi utilizado para controle de posição. Para os eventos discretos é usado a teoria de autômatos de Moore para modelagem e gerado um código através de um software computacional (Deslab). A implementação do controle híbrido é feita no microcontrolador onde são integrados os controladores de variáveis contínuas e a eventos discretos. Uma vez montado, o robô funcionou de forma correta em um percurso de competições da Robocore, percorrendo este percurso sem deixar a faixa central e também identificando os eventos discretos.

**Palavras-chave:** Condicionamento de sinais. Controle híbrido. Sistemas a eventos Discretos; Robô seguidor de linha; Modelagem de variáveis contínuas.

## ABSTRACT

GUADAGNIN, Alan Junior. Hybrid Control of a Following Line Robot. In 2014. 65 f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

This work presents the realization of the modeling and implementation of the hybrid control (continuous-time and discrete events) on a line follower robot. Initially the formal control methods continue controllers (P, I and D) as well as discrete events (Moore and Mealy automata) needed to perform the work are presented. In developing methods for signal conditioning are used, making the experimental process modeling, Simulink software is used for simulation of the mathematical model for testing and implementation of a proportional controller, which was used for position control. For the discrete events is used to automata theory Moore for modeling and code generated by a computer software (Deslab). The implementation of hybrid control is taken where the microcontroller are integrated controllers for continuous variables and discrete events. Once assembled, the robot worked correctly on a course of competitions Robocore, traversing this route without leaving the central belt and also identifying discrete events.

**Keywords:** Signal Conditioning. Hybrid Control. Modeling of discrete events. Following line robot. Modeling continuous variables.



## LISTA DE FIGURAS

FIGURA 1 - EXEMPLO DE MÁQUINA DE MOORE. ....	17
FIGURA 2 - EXEMPLO DE MÁQUINA DE MEALY.....	17
FIGURA 3 - COMPARAÇÃO ENTRE VARIÁVEL QUANTIZADA E ANALÓGICA. ...	18
FIGURA 4 - MODELO IDENTIFICADO DO PROCESSO. ....	19
FIGURA 5 – ELEMENTOS BÁSICOS DE UM CONTROLADOR.....	21
FIGURA 6 - DIAGRAMA DE BLOCOS DE UM CONTROLADOR PROPORCIONAL.....	22
FIGURA 7 - DIAGRAMA DE BLOCOS DE UM CONTROLADOR INTEGRAL.....	23
FIGURA 8 - CONFIGURAÇÕES TÍPICAS DE CONTROLE POR TENSÃO DE ARMADURA.....	25
FIGURA 9 - CHAVEAMENTO PWM: (A) CIRCUITO SIMPLIFICADO; (B) E (C) SÃO DUAS FORMAS DE ONDA DA TENSÃO APLICADA AO MOTOR PARA CICLOS DE CONDUÇÃO DISTINTOS. ....	26
FIGURA 10 – TRANSISTOR FUNCIONANDO COMO CHAVE.....	27
FIGURA 11 - ESQUEMÁTICO DO SENSOR DE REFLETÂNCIA. ....	28
FIGURA 12 - FUNCIONAMENTO DOS SENSORES DE REFLETÂNCIA. ....	28
FIGURA 13 - EXEMPLO DE LINEARIZAÇÃO DE UM SINAL QUALQUER. ....	29
FIGURA 14 - EXEMPLO DE CRUZAMENTO NO PERCURSO.....	33
FIGURA 15 - ÁREA DE PARTIDA-CHEGADA.....	34
FIGURA 16 - DETALHES DA ÁREA DE PARTIDA-CHEGADA.....	35
FIGURA 17 - EXEMPLO DE CURVA NO PERCURSO. ....	36
FIGURA 18 - ESQUEMÁTICO DE LIGAÇÃO DOS SENSORES.....	37
FIGURA 19 - POSICIONAMENTO DOS SENSORES SOBRE A FAIXA CENTRAL.	38
FIGURA 20 - CIRCUITO DE CONDICIONAMENTO DAS VARIÁVEIS CONTÍNUAS.....	39
FIGURA 21 - GRÁFICO “TENSÃO X POSIÇÃO”.....	40

FIGURA 22 - POSIÇÃO DOS SENSORES LATERAIS E CENTRAIS .....	41
FIGURA 23 - CIRCUITO COMPARADOR PARA SENSORES LATERAIS.....	42
FIGURA 24 - ESQUEMA DE LIGAÇÃO DO <i>PUSH BUTTON</i> .....	42
FIGURA 25 - SAÍDA DO CIRCUITO DE CONDICIONAMENTO DAS VARIÁVEIS CONTÍNUAS.....	43
FIGURA 26 - PLANTA DO SISTEMA NO SIMULINK. ....	44
FIGURA 27 - RESPOSTA DO SISTEMA A UM DEGRAU NA ENTRADA.....	44
FIGURA 28 - MALHA DE CONTROLE. ....	45
FIGURA 29 - SAÍDA DA MALHA DE CONTROLE.....	45
FIGURA 30 - DIAGRAMA DE CONTROLE DISCRETO. ....	47
FIGURA 31 - PROCESSO IMPLEMENTADO NO DESLAB. ....	49
FIGURA 32 - ESTRUTURA DO SISTEMA DE CONTROLE HÍBRIDO. ....	50
FIGURA 33 - CIRCUITO DE ACIONAMENTO DE UM MOTOR.....	51
FIGURA 34 - PWM PARA SENSORES NA FAIXA CENTRAL, SINAL EM LARANJA: PWM DO MOTOR DA DIREITA, SINAL EM AZUL: PWM DO MOTOR DA ESQUERDA. ....	53
FIGURA 35 - PWM PARA SENSOR DA ESQUERDA FORA DA FAIXA CENTRAL, SINAL EM LARANJA: PWM DO MOTOR DA DIREITA, SINAL EM AZUL: PWM DO MOTOR DA ESQUERDA.....	54
FIGURA 36 - PWM PARA SENSOR DA DIREITA FORA DA FAIXA CENTRAL, SINAL EM LARANJA: PWM DO MOTOR DA DIREITA, SINAL EM AZUL: PWM DO MOTOR DA ESQUERDA.....	54
FIGURA 37 - ROBÔ SEGUIDOR DE LINHA.....	55

## SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	OBJETIVO GERAL .....	14
1.2	OBJETIVOS ESPECÍFICOS .....	14
1.3	JUSTIFICATIVA .....	15
1.4	ORGANIZAÇÃO DO TRABALHO .....	15
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>16</b>
2.1	MODELOS PARA SISTEMAS A EVENTOS DISCRETOS. ....	16
2.1.1	Autômatos como modelos de Sistemas a Eventos Discretos. ....	16
2.1.1.1	Autômato de Moore.....	16
2.1.1.2	Autômato de Mealy.....	17
2.2	MODELAGEM E CONTROLE DE SISTEMAS DE VARIÁVEIS CONTÍNUAS. ....	18
2.2.1	Modelagem fenomenológica .....	18
2.2.2	Modelagem experimental.....	19
2.2.3	Ações básicas de controle. ....	20
2.3	SISTEMAS DE CONDICIONAMENTO DE SINAIS E ACIONAMENTO DE MOTORES DE CORRENTE CONTÍNUA. ....	24
2.3.1	Motores de corrente contínua e seu acionamento .....	24
2.3.2	Sensores de refletância.....	27
2.3.3	Condicionamento de sinais .....	29
<b>3</b>	<b>METODOLOGIA DE PESQUISA .....</b>	<b>31</b>
<b>4</b>	<b>CONTROLE HÍBRIDO DE UM ROBÔ SEGUIDOR DE LINHA. ....</b>	<b>32</b>
4.1	REGRAS IMPOSTAS POR COMISSÕES ORGANIZADORAS DE COMPETIÇÕES DE ROBÔS SEGUIDORES DE LINHA.....	32
4.2	PROJETO E CONDICIONAMENTO DOS SISTEMAS DE AQUISIÇÃO DE SINAIS CONTÍNUOS. ....	37
4.3	PROJETO E CONDICIONAMENTO DOS SISTEMAS DE AQUISIÇÃO DE SINAIS DISCRETOS.....	40

4.4 LEVANTAMENTO EXPERIMENTAL DO MODELO MATEMÁTICO DO PROCESSO DE VARIÁVEIS CONTÍNUAS E PROJETO DO CONTROLADOR.....	43
4.5 MODELAGEM DO PROCESSO DE VARIÁVEIS DISCRETAS. ....	46
4.6 IMPLEMENTAÇÃO DO CONTROLE HÍBRIDO. ....	50
<b>5 TESTES E RESULTADOS.....</b>	<b>53</b>
<b>6 CONCLUSÕES .....</b>	<b>56</b>
<b>REFERÊNCIAS.....</b>	<b>58</b>
<b>APÊNDICES .....</b>	<b>60</b>

## 1 INTRODUÇÃO

Robôs são dispositivos físicos que, através de sensores que “percebem” o mundo para interagir com ele por meio de ações como manipulação e locomoção, sendo assim classificados em manipuladores e móveis. Os manipuladores são, em sua maioria, de uso industrial, e operam em áreas de trabalho restritas e dedicadas, por exemplo, a pintura, montagem, posicionamento de peças etc. Já os robôs móveis podem mover-se pelo ambiente, realizar supervisão, inspeção, transporte, monitoramento etc. Seu uso pode ser no meio industrial, agrícola, militar, doméstico, medicina, entretenimento, entre outros (MARIN, 2010, p. 21).

Na robótica móvel, há dois tipos de robôs: os veículos guiados automaticamente (AGVs - do acrônimo em inglês: *Automatic Guided Vehicle*) e os robôs móveis autônomos. Os primeiros operam em ambientes projetados (p.e., com caminhos induzidos, faróis de direção, ou outras marcas de balizamento) e executam tarefas de transporte ao longo de rotas fixas. Por esta razão se tornam inflexíveis e frágeis: alterar uma rota implica em aumento de custo e qualquer mudança inesperada (tais como objetos bloqueando o caminho) pode levar a falhas na execução da tarefa. Como alternativa a este tipo de robô, surgiram os robôs móveis autônomos (NEHMZOW, 2000).

A crescente utilização de robôs autônomos em diversas aplicações, principalmente na exploração de locais onde há risco de vida, faz com que muitas pesquisas e competições surjam no intuito de incentivar estudantes de engenharia e áreas afins a desenvolver conhecimento sobre o tema (E. SILVA, *et al.*, 2006, p. 2).

Robôs seguidores de linha podem ser classificados como um tipo de veículo guiado automaticamente. Buscando melhorar o desempenho destes robôs, diversas comissões organizadoras propiciam competições nesta área. Normalmente os objetivos são diminuir o tempo para realizar um percurso sobre uma faixa de orientação e ainda reconhecer marcas (sinais discretos) indicando algumas informações do percurso, por exemplo, marca de início de trajeto, cruzamentos, curvas, retas etc. Esse sistema com variáveis contínuas e eventos discretos interagindo entre si será denotado neste trabalho como sistema híbrido.

O termo sistemas híbridos foi empregado primeiramente por Witsenhausen (1966) para descrever a combinação entre dinâmicas discretas e

continuas de um sistema linear, desde então vem utilizando essas técnicas de controle híbrido para controle de robôs autônomos, por exemplo: Em Heine (2002) é desenvolvido um sistema de controle robusto para robôs moveis autônomos sendo capaz de operar e de se adaptar a diferentes ambientes e condições, utilizando uma arquitetura de controle híbrida (deliberativa e reativa); e também em Leite (2005) são utilizados técnicas de controle híbrido para robôs autônomos neste trabalho, o autor faz o uso de um sensor de força e de uma câmera fixa, o método de controle proposto serve para combinar controle de força direta com o servo visão fazendo com que o robô se adapte na presença de superfícies com geometria desconhecida.

## 1.1 OBJETIVO GERAL

Este trabalho tem como objetivo principal realizar a modelagem e implementação do controle híbrido (de tempo contínuo e a eventos discretos) de um robô seguidor de linha.

## 1.2 OBJETIVOS ESPECÍFICOS

- Definir o modelo de sensor fotoelétrico capaz de perceber diferentes níveis de refletância da superfície em que robô irá percorrer.
- Desenvolver circuitos eletroeletrônicos para condicionamento de sinais e acionamento dos motores.
- Realizar a modelagem e projeto do controlador do sistema de posicionamento do robô (Sistema com variáveis contínuas no tempo).
- Realizar a modelagem do sistema a eventos discretos do robô.
- Implementar o controle híbrido em um microcontrolador.
- Realizar testes do robô em um circuito de competições da Robocore.

### 1.3 JUSTIFICATIVA

Neste trabalho será desenvolvido um robô móvel seguidor de linha explorando o controle de posição (variável contínua) e detecção de marcas laterais (eventos discretos) através uma metodologia formal (métodos de controle já conhecidos). Fazendo-se o uso desta metodologia de controle e aplicando eles em sistemas híbridos é possível realizar simulação, verificação, modularização (divisão do problema em partes) de uma maneira mais simples e prática, além disso pode ser aplicado na geração automática de linguagem em código para controladores (Microcontroladores ou CLP's) facilitando a realização de projetos.

Busca-se neste trabalho fazer a implementação do controle híbrido que atue em um percurso de competições da Robocore de forma precisa, reconhecendo todas as marcas e atuando sobre a faixa central, com este controle é possível fazer com que o robô atue com uma velocidade mais alta nas retas e mais baixa nas curvas, percorrendo de forma mais rápida o percurso, comparado com os robôs já existentes para competições.

### 1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em cinco capítulos:

Capítulo 2 – Faz uma abordagem da teoria necessária para realização deste trabalho.

Capítulo 3 – É feito o desenvolvimento do trabalho, desde a parte de aquisição de sinais, modelagem e implementação no microcontrolador. Apresentado também os testes e resultados.

Capítulo 4 – São apresentadas as discussões finais e conclusões sobre o trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

Neste primeiro capítulo será realizada uma revisão bibliográfica sobre a teoria necessária para realização deste trabalho.

### 2.1 MODELOS PARA SISTEMAS A EVENTOS DISCRETOS.

Neste capítulo são apresentadas as técnicas de modelagem de sistemas a eventos discretos, como por autômatos de Moore e de Mealy.

#### 2.1.1 Autômatos como modelos de Sistemas a Eventos Discretos.

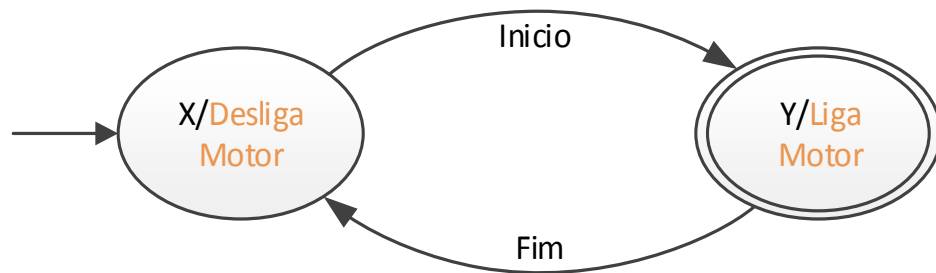
O autômato é, por excelência, a "máquina" responsável pelo comando dos sistemas automatizados (INESC, 2007). Tais sistemas têm em comum a maneira pela qual percebem as ocorrências no ambiente a sua volta, o que se dá pela recepção de estímulos, denominados eventos. São exemplos de eventos o início e o término de uma tarefa e a percepção de uma mudança de estado em um sensor. Estes eventos são, por sua natureza, instantâneos, o que lhes confere um caráter discreto no tempo. Sistemas com estas características são denominados sistemas a eventos discretos (SED) (CURY, 2001, p. 7). Estes sistemas podem ser representados por modelos de linguagem de autômatos de Moore ou Mealy dentre outros (Redes de Petri, Cadeias de Markov etc.).

##### 2.1.1.1 Autômato de Moore.

A Máquina de Moore possui uma função que gera uma palavra de saída (que pode ser vazia) para cada estado da máquina. Esta saída só depende do



estado atual da máquina (DE BRITO; MARTENDAL; DE OLIVEIRA, 2003, p. 2), ou seja, a ação de saída da máquina está nos estados (Figura 1).



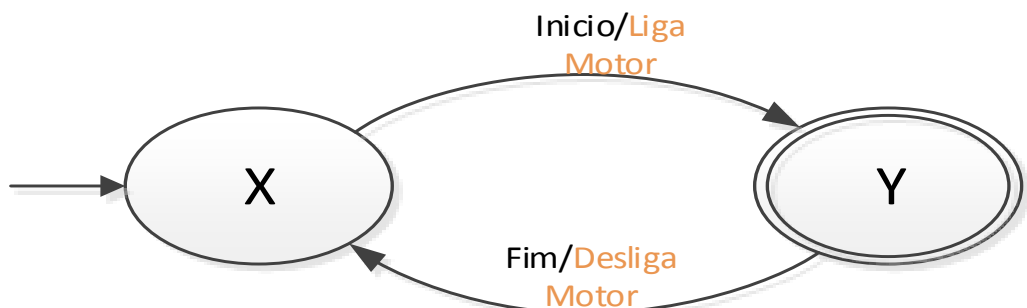
**Figura 1 - Exemplo de máquina de Moore.**

**Fonte: Torrico (2013, p. 28)**

Toda máquina de Moore pode ser simulada por um máquina de Mealy (MENEZES, 2004, p. 23).

#### 2.1.1.2 Autômato de Mealy.

Máquina de Mealy (vide Figura 2) é um Autômato Finito modificado de forma a gerar uma palavra de saída para cada transição entre os estados. Neste tipo de máquina de estado estas palavras de saída dependem do estado atual e do valor das entradas (DE BRITO; MARTENDAL; DE OLIVEIRA, 2003, p. 2), ou seja, a ação de saída está nas transições.



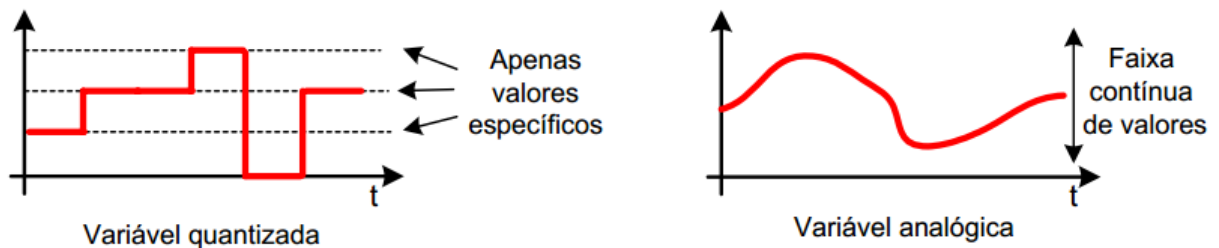
**Figura 2 - Exemplo de máquina de Mealy.**

**Fonte: Torrico (2013, p. 29).**

Toda máquina de Mealy pode ser simulada por uma máquina de Moore (MENEZES, 2004, p. 27).

## 2.2 MODELAGEM E CONTROLE DE SISTEMAS DE VARIÁVEIS CONTÍNUAS.

Sistemas de variáveis contínuas são os sistemas cuja evolução dinâmica no tempo envolve variáveis com um comportamento contínuo no tempo, são normalmente descrito por variáveis analógicas e é expresso ao longo de uma escala contínua de valores (Figura 3) (MICHELS, 2011, p. 36).



**Figura 3 - Comparação entre variável quantizada e analógica.**

Fonte: Michels (2011, p. 36).

Esses sistemas devem ser modelados matematicamente através de técnicas que serão apresentadas a seguir, também são apresentados técnicas de controle para sistemas de variáveis contínuas.

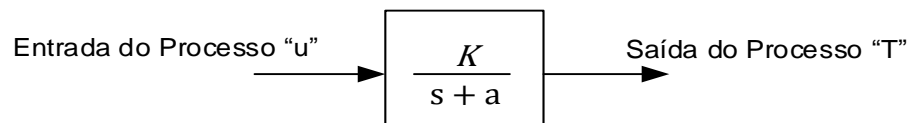
### 2.2.1 Modelagem fenomenológica

Segundo Mellichamp e Seborg (2010) a modelagem fenomenológica usa as leis físicas e correlações para descrever o sistema. Um processo pode ser caracterizado por suas variáveis de estado que descrevem a quantidade de massa, energia e momento linear do sistema. As variáveis do sistema podem ser sistemas mecânicos (posição e velocidade), sistemas elétricos (tensões e correntes) entre outras.

### 2.2.2 Modelagem experimental.

A modelagem experimental se dá pela determinação de um conjunto de variáveis de resposta, pensadas para representar a qualidade do produto, resultantes do tratamento matemático de possibilidades delimitadas de arranjos entre matérias-primas, sob determinadas condições de processo (DA SILVA; DA SILVA, 2009, p. 2).

A Figura 4 mostra um modelo de primeira ordem, que é na maioria das vezes apenas uma aproximação da realidade, pois um processo real raramente é linear e de primeira ordem. Entretanto, esta aproximação da dinâmica da planta industrial é satisfatória para se ajustar e definir muitos controles na prática. (CAMPOS; TEIXEIRA, 2010, p. 17)



**Figura 4 - Modelo identificado do processo.**

**Fonte: Campos e Teixeira (2010, p. 17).**

Sendo  $a = 1/\tau$ ,  $K =$  ganho,  $\tau =$  constante de tempo.

Este modelo é um dos mais utilizados na prática para representar a dinâmica do processo e servir de base para a sintonia dos controladores do tipo PID (ZIEGLER; NICHOLS, 1942).

Para obter estes parâmetros de acordo com Campo e Teixeira (2010, p. 17) que representam a dinâmica do processo  $(K, \tau)$  pode-se utilizar uma metodologia de identificação, que consiste no seguinte:

- Introduzir perturbações iniciais em degrau na variável manipulada (U) de histereses etc.), isto é, efetuar dois ciclos de degraus para cima e para baixo.

- Esperar para que o sistema atinja o regime permanente estável (variável controlada constante) e que não esteja sendo perturbado por alguma outra variável (no caso de sistemas multivariáveis).
- Introduzir um degrau na variável manipulada, e obter a resposta do processo.

Esta resposta do processo é conhecida na prática como “curva de reação” da planta. A partir desta curva pode-se calcular os parâmetros do modelo do processo (Ex. ganho, constante de tempo) (CAMPOS; TEIXEIRA, 2010, p. 18):

- O ganho do processo ( $K$ ) em unidade de engenharia é calculado dividindo-se a variação da variável controlada em regime permanente pela variação da variável manipulada  $K = \frac{\text{Entrada}}{\text{Saída}}$
- A constante de tempo ( $\tau$ ) é o tempo a partir do início da perturbação na variável manipulada, em que a variável controlada já atingiu 63% da variação total até o novo regime permanente.
- Após calculado  $\tau$  é calculado  $a$  ( $a = 1/\tau$ ), que também pode ser calculado pelo tempo de assentamento ( $T_s$ ) pela fórmula  $a = 4/T_s$ .

Em algumas situações se deparamos com um sistema não linear, o qual pode-se linearizado ou aproximado por sistemas lineares ao redor de um ponto de operação.

### 2.2.3 Ações básicas de controle.

Um controlador automático compara o valor de saída real do processo ou planta com o valor de referência, determina o desvio e produz um sinal de controle que reduz o desvio. O modo como o controlador automático produz o sinal de controle é chamado ação de controle (CAMPOS, 1999, p. 1).

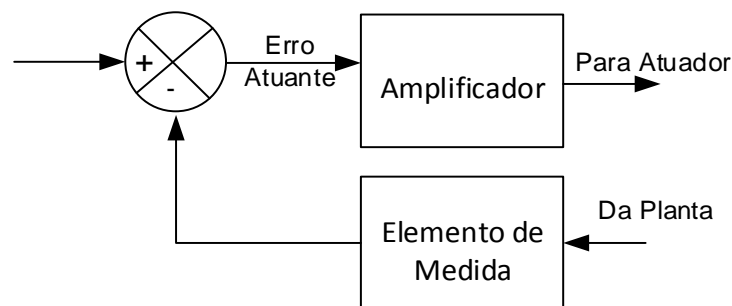
Segundo Campos (1999, p. 1) os controladores automáticos industriais podem ser classificados de acordo com a ação de controle desempenhada:

- Controladores ON-OFF
- Controladores proporcionais;
- Controladores do tipo integral;
- Controladores do tipo derivativo;

- Controladores do tipo proporcional mais integral mais derivativo (PID).

A fonte de potência da maioria dos controladores industriais é a eletricidade ou fluido pressurizado (ar ou óleo), desta maneira os controladores podem ser classificados como hidráulicos, pneumáticos ou eletroeletrônicos.

A Figura 5 mostra o controlador e um elemento de medida. O elemento de medida é um dispositivo que converte a variável de saída em outra variável conveniente que pode ser usada para comparar a saída em relação ao sinal de entrada de referência (CAMPOS, 1999, p. 2).



**Figura 5 – Elementos Básicos de um controlador.**

**Fonte: Campos (1999, p. 2).**

O controlador consiste de um detector de erro e um amplificador. O controlador é ligado a um dispositivo de potência (atuador), tal como um motor ou válvula pneumática

### **Controlador ON-OFF**

De todas as ações de controle, a ação *on-off* é a mais barata e a mais simples, sendo utilizada tanto em sistemas de controle industrial como doméstico. Como o próprio nome indica, ela só permite duas posições para o elemento final de controle, ou seja: totalmente aberto ou totalmente fechado (OLIVEIRA, 1999, p. 23).

Segundo Oliveira (1999, p. 24) o controle através da ação em duas posições é simples e, ainda, econômico, sendo portanto utilizado largamente nos dias atuais. Principalmente, os controles de temperatura nos fornos elétricos pequenos, fornos de secagem, etc., são realizados em sua maioria por este método. No entanto, por outro lado, apresenta certas desvantagens por provocar oscilações e *offset* e, principalmente, quando provoca tempo morto muito grande, os resultados de controle por estes controles simples tornam-se acentuadamente inadequados.

Assim, quando não é possível utilizar este tipo de controle, recorre-se a outros tipos de controle mais complexos, mas que eliminam os inconvenientes deste tipo.

### Controlador Proporcional (P)

De acordo com Oliveira (1999, p. 28) na ação liga-desliga, quando a variável controlada se desvia do valor ajustado, o elemento final de controle realiza um movimento brusco de ON (liga) para Off (desliga), provocando uma oscilação no resultado de controle. Para evitar tal tipo de movimento foi desenvolvido um tipo de ação no qual a ação corretiva produzida por este mecanismo é proporcional ao valor do desvio. Tal ação denominou-se ação proporcional.

Para um controlador com ação de controle proporcional a relação entre a ação de controle  $u(t)$  e o erro de atuação  $e(t)$  é dado por:

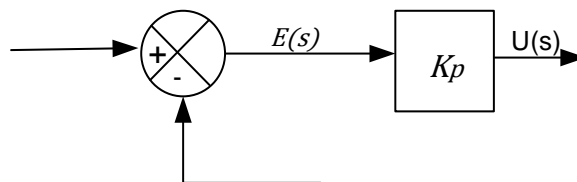
$$u(t) = K_p e(t) \quad 2.1$$

Ou expresso na variável de Laplace como:

$$U(s) = K_p E(s) \quad 2.2$$

Onde  $K_p$  é denominado constante proporcional.

O diagrama de bloco deste tipo de controlador está ilustrado na Figura 6:



**Figura 6 - Diagrama de blocos de um controlador proporcional.**

**Fonte: Campos (1999, p. 4)**

Com a introdução da ação proporcional, se consegue eliminar as inconvenientes oscilações provocadas pelo controle *on-off*. No entanto esta ação não consegue manter o sistema em equilíbrio sem provocar o aparecimento do erro de *offset* caso haja variação na carga (OLIVEIRA, 1999, p. 30).

### Controlador Integral (I)

A ação integral vai atuar no processo ao longo do tempo enquanto existir diferença entre o valor desejado e o valor medido. Assim, o sinal de correção é integrado no tempo e por isto enquanto a ação proporcional atua de forma

instantânea quando acontece um distúrbio em degrau, a ação integral vai atuar de forma lenta até eliminar por completo o erro (OLIVEIRA, 1999, p. 30).

Segundo Oliveira (1999, p. 33) a ação integral foi introduzida principalmente para eliminar o erro de offset deixado pela ação proporcional, atuando então, até que o desvio volte a ser nulo. No entanto, como ela é uma função do tempo, sua resposta é lenta e por isto, desvios grandes em curtos espaços de tempo não são devidamente corrigidos.

Na ação de controle integral, o valor da ação de controle  $u(t)$  varia proporcionalmente ao sinal de erro  $e(t)$ :

$$\frac{du(t)}{dt} = K_i e(t) \quad 2.3$$

Ou,

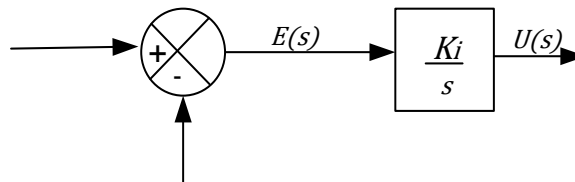
$$u(t) = K_i \int_0^t e(t) dt \quad 2.4$$

A sua função de transferência pode ser representada como:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad 2.5$$

Se o valor de  $e(t)$  dobra, então o valor de  $u(t)$  varia duas vezes mais rápido. Se  $e(t) = 0$  o valor de  $u(t)$  permanece estacionário.

O diagrama de bloco deste tipo de controlador está ilustrado na Figura 7



**Figura 7 - Diagrama de blocos de um controlador integral.**

**Fonte: Campos (1999, p. 4)**

A ação proporcional corrige os erros instantaneamente e a integral se encarrega de eliminar a longo prazo qualquer desvio que permaneça (por exemplo, erro de offset) (OLIVEIRA, 1999, p. 33).

### **Controlador Derivativo (D)**

Esta ação diminuindo o tempo de resposta. Como esta ação de controle depende somente da razão da variação do desvio e não da amplitude deste, não deve ser utilizada sozinha pois tende a produzir movimentos rápidos no elemento

final de controle tornando o sistema instável, pois causará grandes e rápidas variações na saída do controlador levando o processo à instabilidade (OLIVEIRA, 1999, p. 39).

### **Controlador Proporcional-Integral-Derivativo (PID)**

O controle proporcional associado ao integral e derivativo, é o mais utilizado em sistemas de malha fechada.

A proporcional elimina as oscilações, a integral elimina o desvio de offset enquanto a derivativa fornece ao sistema uma ação antecipada evitando previamente que o desvio se torne maior quando o processo se caracteriza por ter uma correção lenta comparada com a velocidade do desvio (por exemplo, alguns controles de temperatura) (OLIVEIRA, 1999, p. 39).

A associação das três ações de controle permite-nos obter um tipo de controle que reúne todas as vantagens individuais de cada um deles e por isto, associação é normalmente utilizada em processo com resposta lenta (constante de tempo grande) e sem muito ruído (OLIVEIRA, 1999, p. 49).

## **2.3 SISTEMAS DE CONDICIONAMENTO DE SINAIS E ACIONAMENTO DE MOTORES DE CORRENTE CONTÍNUA.**

Aqui serão apresentadas as técnicas de condicionamento de sinais, também métodos de acionamento de motores de corrente contínua.

### **2.3.1 Motores de corrente contínua e seu acionamento**

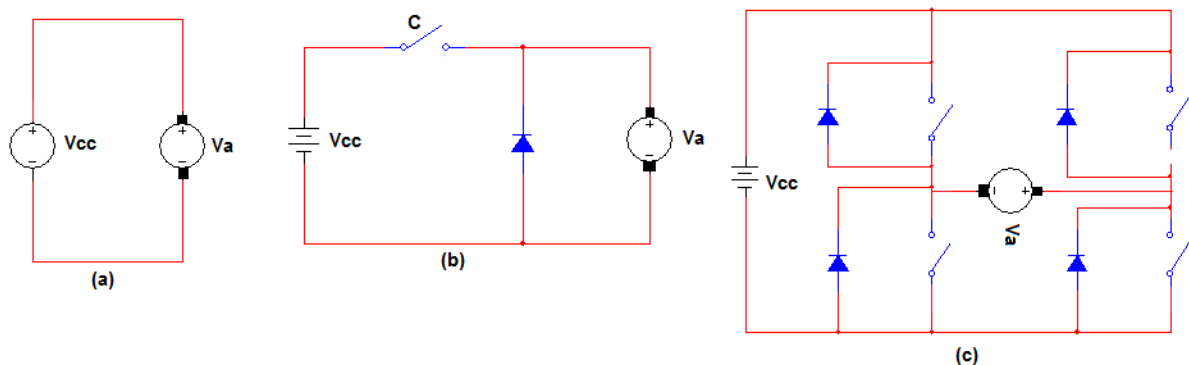
Os motores CC (Corrente contínua) possuem uma ampla versatilidade no seu modo de ser projetado segundo Fitzgerald, Kingsley e Umans (2006), eles podem ser projetadas de modo a apresentar uma ampla variedade de características tensão *versus* corrente ou de velocidade *versus* conjugado, para operações em regime permanente. Devido a essa facilidade de controle os motores CC vem sendo



usados em aplicações que exigem uma ampla faixa de velocidade ou de controle preciso da saída do motor.

Os três métodos mais comuns para controle de velocidade são métodos para ajustar o fluxo, usualmente por meio do controle de corrente de campo, da variação da resistência associada ao circuito de armadura e da variação da tensão dos terminais da armadura (FITZGERALD; KINGSLEY JR.; UMANS, 2006).

O controle por tensão no terminal de armadura segundo Fitzgerald (2006) pode ser feito fazendo o uso de sistemas eletrônicos de potência como mostra a Figura 8.



**Figura 8 - Configurações típicas de controle por tensão de armadura.**

Fonte: (FITZGERALD; KINGSLEY JR.; UMANS, 2006).

Na primeira é aplicado diretamente uma tensão variável nos terminais da armadura, na segunda uma tensão constante é aplicada nos terminais da armadura sendo variada então por um esquema de modulação por largura de pulso, no qual a chave  $C$  é alternadamente aberta e fechada, sendo a tensão média do terminal de armadura igual a

$$V_a = D \cdot V_{cc} \quad 2.6$$

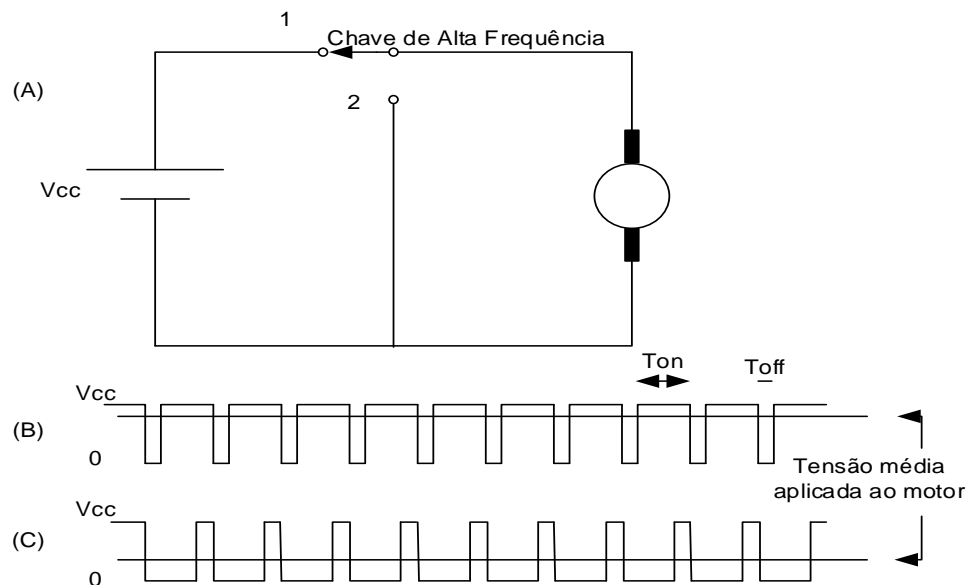
Onde:  $V_a$  é a tensão de armadura,  $D$  é o ciclo de trabalho PWM e  $V_{cc}$  é a tensão constante aplicada.

E na figura 1c é apresentada a configuração de uma ponte H completa, onde a tensão aplicada nos terminais da armadura pode ter valores no intervalo de  $-V_{cc} \leq V_a \leq V_{cc}$ .

O acionamento de motores CC pode ser feito através da variação contínua da tensão aplicada ao motor. Porém usam-se atualmente técnicas de PWM, que são mais modernas e que diminuem as perdas de energia e por

consequente reduzem o aquecimento dos componentes do acionador permitindo a redução de seu custo e tamanho (TORRES, 2004, p. 9)

O PWM consiste num método de modulação de tensão em frequências relativamente altas (usualmente da ordem de kHz), conforme apresenta Torres (TORRES, 2004, p. 10) . Na Figura 9 o tempo  $T_{ON}$  corresponde ao tempo que a chave permanece na posição ligado(1) e  $T_{OFF}$  tempo que a mesma permanece na posição desligado(2).



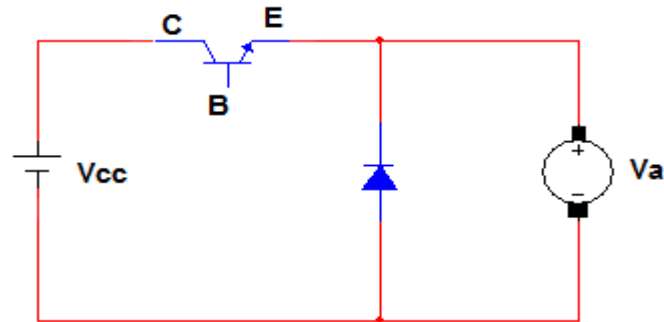
**Figura 9 - Chaveamento PWM: (a) Circuito simplificado; (b) e (c) são duas formas de onda da tensão aplicada ao motor para ciclos de condução distintos.**

Fonte: Torres (2004, p. 10).

A tensão média no motor visto na Figura 9 é calculada através de:

$$V_M = V_{CC} \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad 2.7$$

Normalmente o chaveamento é, de fato, realizado por transistores como no circuito mostrado por Kenjo (1991) na Figura 10. Nesta figura também se utiliza um diodo em paralelo com a armadura para permitir o fluxo de corrente pela mesma quando o transistor está cortado (denominado diodo de roda livre).



**Figura 10 – Transistor funcionando como chave.**

**Fonte: Torres (2004, p. 11).**

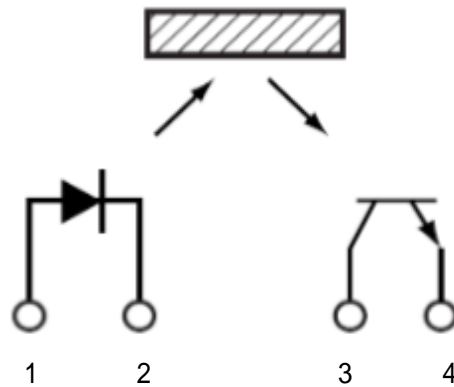
Neste circuito, o sinal PWM é aplicado à base (B) do transistor. Quando uma tensão positiva adequada é aplicada em B com relação ao emissor (E) (TORRES, 2004). Caso o transistor sature ele irá funcionar como uma chave fechada, fazendo com que a tensão ( $V_{CC}$ ) seja aplicada ao motor, caso contrário o transistor funciona como uma chave aberta e nenhuma tensão é aplicada ao motor.

### 2.3.2 Sensores de refletância

Os sensores óticos possuem um princípio de funcionamento baseado num feixe luminoso, em geral infravermelho e que pode ser polarizado (ou não), gerado por um dispositivo emissor e captado por outro dispositivo receptor (Mecatrônica Atual, 2004). Dentro da classe de sensores óticos encontra-se os sensores de refletância.

Um sensor de refletância consiste em um circuito eletrônico que seja capaz de detectar o quão reflexiva é a superfície através de um raio de luz incidente (MENEGUELE; FERREIRA; ARCANJO, 2011, p. 22).

Este tipo de sensor possui um emissor e um receptor de luz (Figura 11), normalmente o emissor emite um raio de luz infravermelho em uma superfície perpendicular a este sendo refletido para o receptor.

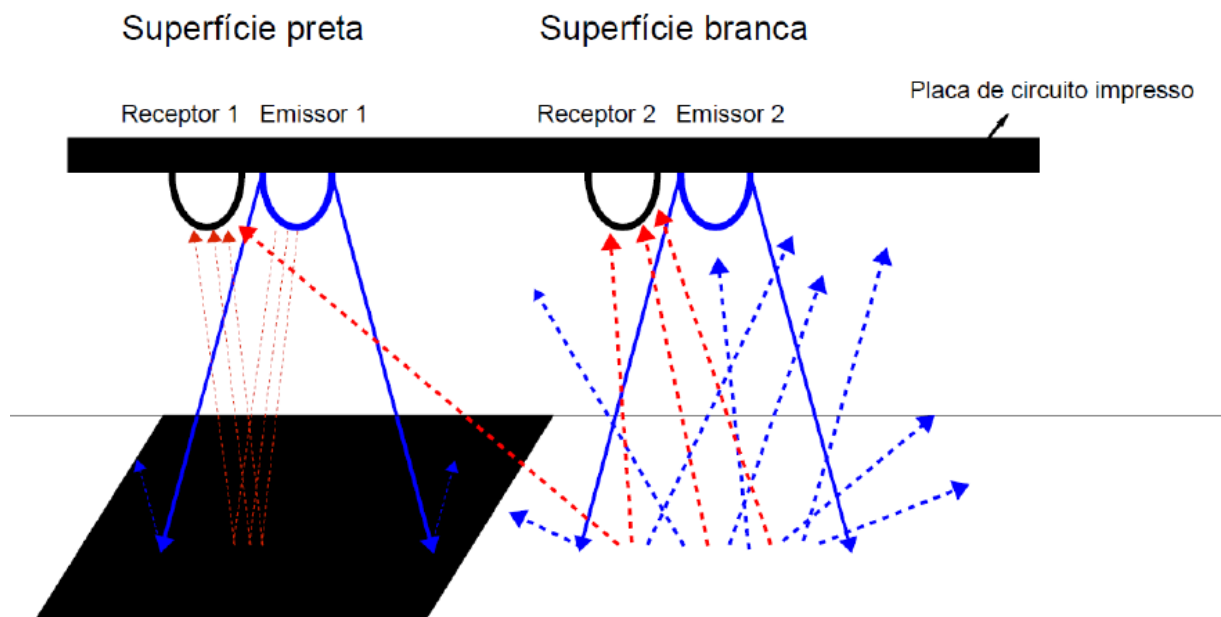


**Figura 11 - Esquemático do sensor de refletância.**

**Fonte: (Datasheet QRE1113GR, 2009)**

Os sensores de refletância possuem fototransistores que são dispositivos fotoelétricos, segundo Souza e Perreira (2003, p. 1) a base do fototransistor é sensível a luz, quando há presença da mesma o transistor conduz, entretanto quando não há presença de luminosidade, o transistor fica cortado, ou seja, com esse sensor é possível diferenciar a faixa do restante da superfície.

Sabe-se que uma superfície preta absorve grande parte da luz incidente enquanto a superfície branca reflete grande parte (vide Figura 12), pode-se usar o fototransistor, devido suas características explicadas, como receptor dos raios de luz refletidos em ambas as superfícies (Mecatrônica Atual, 2004).



**Figura 12 - Funcionamento dos sensores de refletância.**

**Fonte: Meneguele, Ferreira e Arcanjo (2011, p. 23).**

### 2.3.3 Condicionamento de sinais

A existência de vários sensores em um projeto irá resultar em um grande número de sinais. Assim, o condicionamento de sinais é necessário para converter tais sinais e então “interficiar” de forma adequada para o controle de processo (FERREIRA, 2000, p. 61).

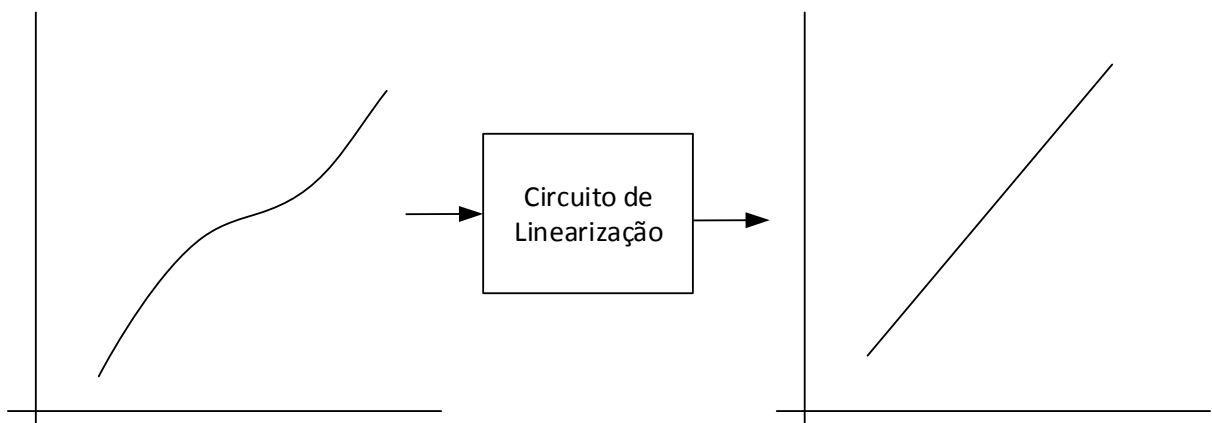
O condicionamento de sinal passa por várias etapas: amplificar, filtrar e equalizar o sinal para que este ganhe níveis de tensão adequados, com boa relação sinal/ruído e distorção harmônica mínima (CASSIOLATO, 2011). Será apresentado a seguir essas técnicas de condicionamento de sinais:

- Mudança no nível

O deslocamento de nível é o método mais simples e mais usado condicionamento de sinais. Um exemplo típico é a necessidade de amplificar ou atenuar um nível de tensão (FERREIRA, 2000, p. 22).

- Linearização

Muitos transdutores, como os termopares, têm uma resposta não-linear às mudanças das ocorrências que vão ser medidas. Deve-se conhecer a natureza do sinal, a configuração de medição e os efeitos do ambiente ao redor do sistema (CASSIOLATO, 2011). Baseado no sinal desses transdutores, é determinado um condicionamento de sinal, que pode ser um circuito de linearização (vide Figura 13) para um sistema de aquisição de dados.



**Figura 13 - Exemplo de linearização de um sinal qualquer.**

**Fonte: Cassiolato (2011).**

- Conversão

Frequentemente, segundo Ferreira (2000, p. 64) o condicionamento de sinais é usado para converter um tipo de variação elétrica em outra. Assim, uma grande quantidade de sensores fornece mudança na resistência quando a variável dinâmica muda. Neste caso, é necessário projetar um circuito que converta variação de resistência em sinal de corrente ou tensão.

- Isolação

Em algumas situações é possível que o sinal do sensor contenha uma tensão muito acima do valor tolerável do circuito de condicionamento de sinal. Nesta situação faz-se o uso de amplificadores isolados para adequar este sinal ao circuito de condicionamento e aquisição de sinal. (FERREIRA, 2000, p. 65).

- Filtragem

O função de um filtro é eliminar ruídos de sinais que estamos medindo. Um filtro de ruídos é usado nos sinais DC para atenuar sinais de alta frequência que podem reduzir a precisão da medição. Sinais de corrente alternada geralmente requerem um tipo diferente de filtro conhecido como filtro *anti-aliasing*. O filtro *anti-aliasing* é um filtro passa baixa que requer uma taxa de corte muito alta, e geralmente remove completamente todas as frequências do sinal que são maiores que a largura de banda de entrada do equipamento. Se esses sinais não forem removidos, eles irão aparecer erroneamente com os sinais da largura de banda de entrada do equipamento (CASSIOLATO, 2011).

### 3 METODOLOGIA DE PESQUISA

Neste capítulo são apresentados os métodos utilizados para o desenvolvimento do robô seguidor de linha.

Neste trabalho realiza-se uma pesquisa experimental, onde faz-se o levantamento das variáveis do processo, em seguida define-se os métodos de controle e por fim realizam-se testes para comprovar o método de controle.

O levantamento do modelo matemático da variável contínua foi escolhido a modelagem experimental, por ser mais simples e prático que a modelagem fenomenológica, utilizando-se o software Simulink para simulação do modelo matemático e também do controlador. Para a modelagem dos eventos discretos escolheu-se a modelagem por autômatos de moore, pela facilidade de implementação em microcontroladores, sendo que já existe geradores de código para isso, como o software Deslab (software desenvolvido pelo professor Dr. César Rafael Claire Torrico).

Para detecção da faixa e das marcas foram utilizados sensores de refletância por ser o tipo de sensor mais adequado para esta aplicação.

Para o controle da variável contínua escolheu-se o controlador P, pois a própria planta já tinha um integrador, esse tipo de controlador também é um dos mais simples de implementação.

A integração dos controladores de variáveis contínuas e a eventos discretos, se realizou com a implementação deste controle híbrido no microcontrolador, com a geração automática de código ficou simples esta implementação.

## 4 CONTROLE HÍBRIDO DE UM ROBÔ SEGUIDOR DE LINHA.

Neste capítulo apresenta-se o desenvolvimento do trabalho, primeiramente apresentam-se regras gerais impostas em competições de robôs seguidores de linha.

Em seguida é apresentado o projeto e condicionamento da parte contínua e discreta, sendo demonstrado os circuitos utilizados para esse processo e também as justificativas para a seu uso.

Por fim é implementado o controle híbrido, ou seja, é projetado o controlador de variáveis contínuas e o de eventos discretos, ambos inseridos no código do microcontrolador.

### 4.1 REGRAS IMPOSTAS POR COMISSÕES ORGANIZADORAS DE COMPETIÇÕES DE ROBÔS SEGUIDORES DE LINHA.

Esta seção estará baseada nas regras fornecidas pela Robocore (2013), o qual é uma das comissões organizadoras mais reconhecidas do Brasil nesta área. As regras são as seguintes:

- **Especificações dos robôs**

Os robôs 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 poderão ser feitas durante a competição. Porém pequenos reparos serão permitidos.

O Robô não pode exceder 250mm de comprimento, 250mm de largura e 200mm de altura.

O Robô não poderá possuir nenhum tipo de mecanismo de sucção para aumentar a força normal em relação ao solo.

- **Percurso**

A superfície da pista será preta, feita com uma ou mais placas de MDF revestidas com fórmica preta, portanto eventuais emendas de placas serão

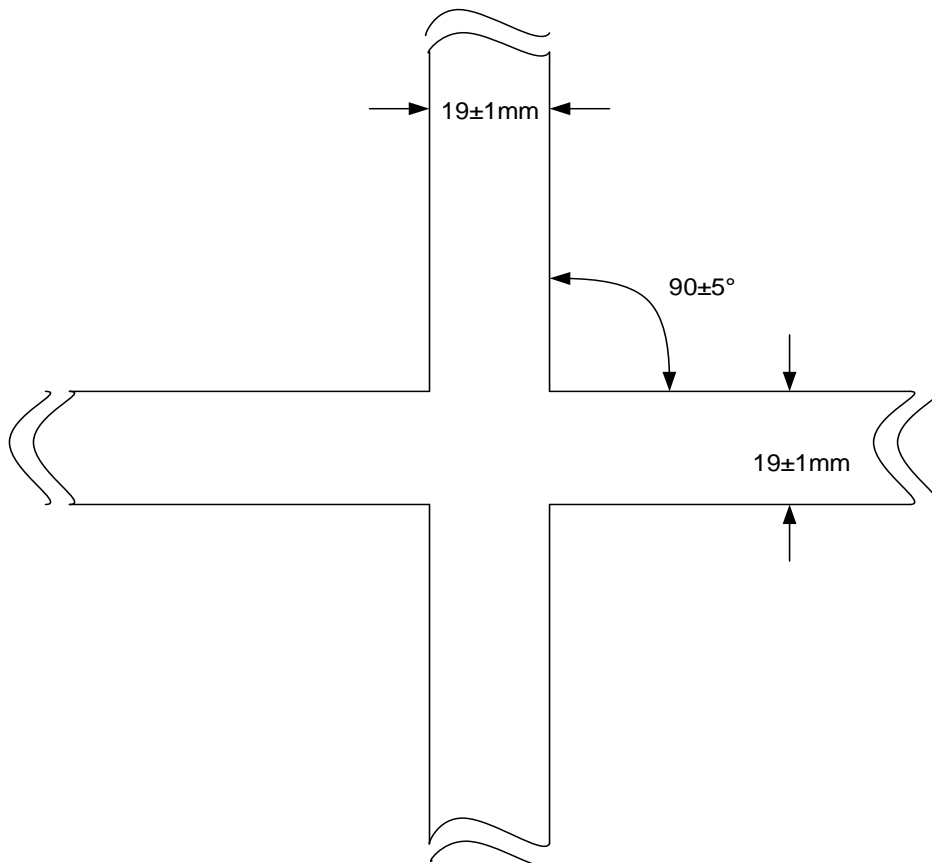


necessárias para compor toda a área do percurso, para tal possíveis desníveis poderão ocorrer, a organização tentará minimizá-los da melhor maneira possível adicionando fita preta em todas as emendas. De qualquer forma os robôs devem ser capazes de superar tais desníveis ( $\pm 1$ mm).

O percurso será indicado por uma linha branca de  $19\pm 1$ mm de largura. O comprimento total da linha será no máximo 60m.

A linha consistirá em combinações de retas e arcos. A linha poderá cruzar sobre ela mesma.

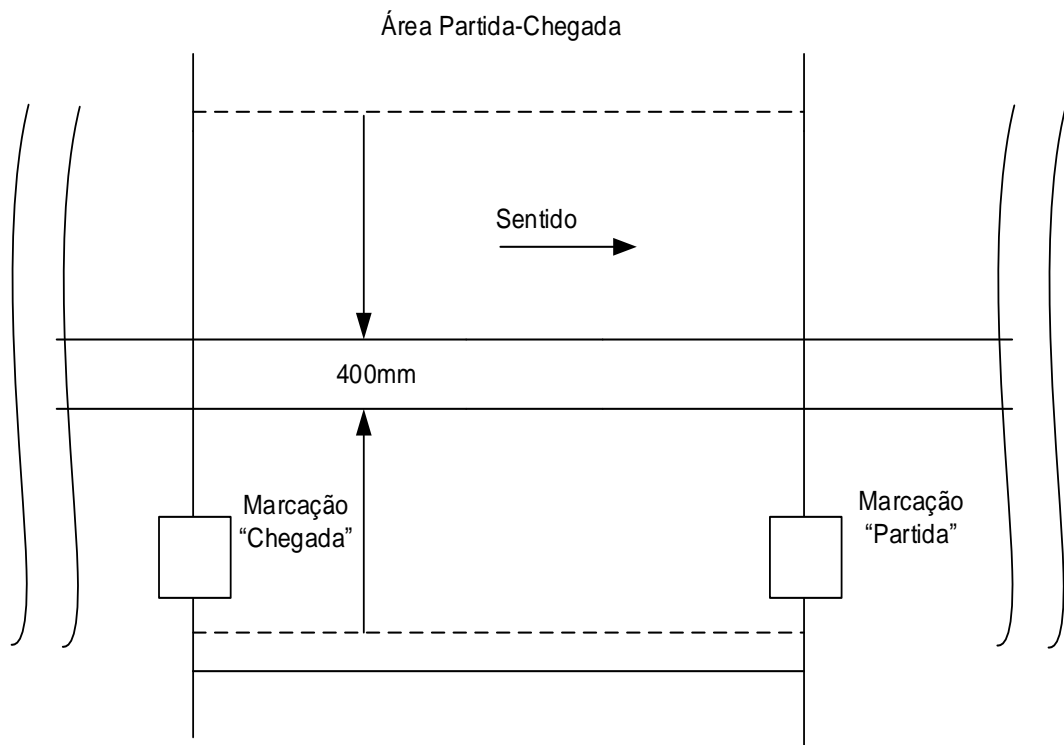
Quando houver um cruzamento, o ângulo de intersecção das linhas será de  $90\pm 5^\circ$  (Figura 14). As partes das linhas 250mm antes e 250mm depois do cruzamento serão retas.



**Figura 14 - Exemplo de cruzamento no percurso.**

**Fonte: (Robocore, 2013).**

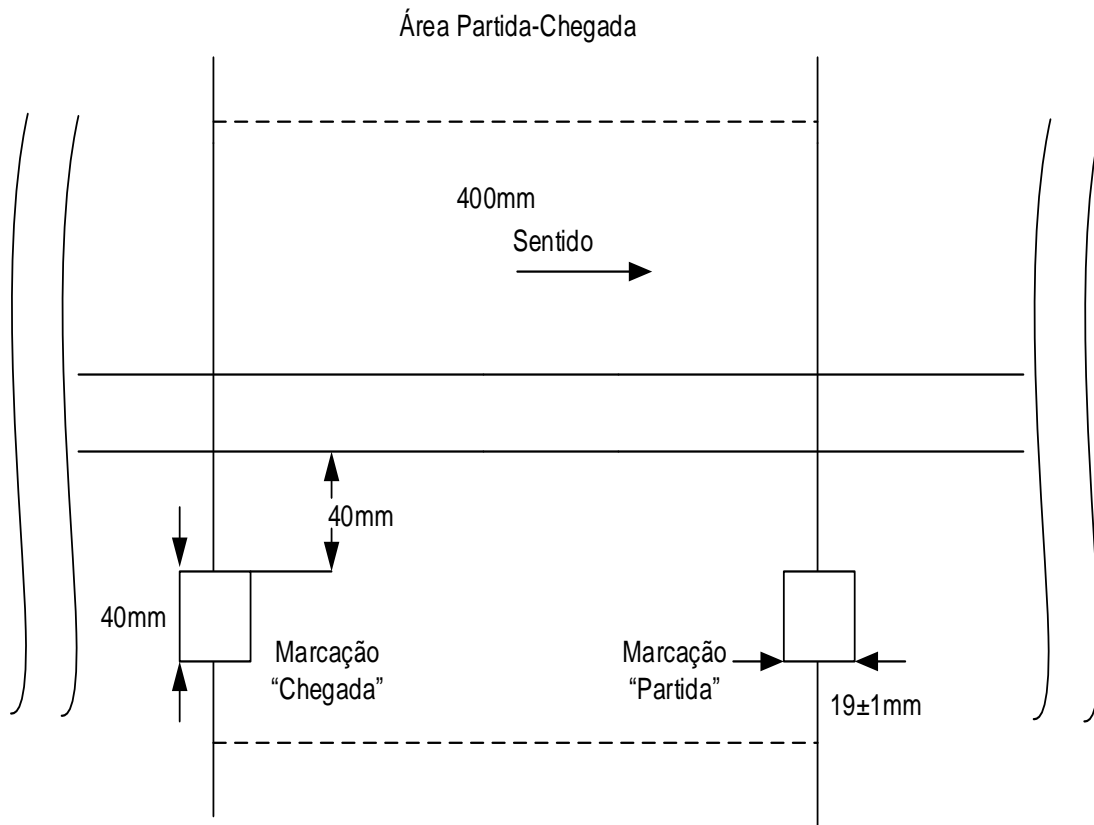
A área a qual se estende entre o ponto de partida e o ponto de chegada, considerando 200mm a direita da linha e 200mm a esquerda da linha é denominada "área de partida-chegada" (Figura 15).



**Figura 15 - Área de partida-chegada.**

**Fonte: (Robocore, 2013).**

A linha de partida e a linha de chegada serão localizadas em uma reta do percurso. A linha de chegada será localizada a um metro para trás da linha de partida. Haverá marcações no lado direito da linha (em relação ao sentido do percurso), indicando o ponto de partida e o ponto de chegada (Figura 15 e Figura 16)

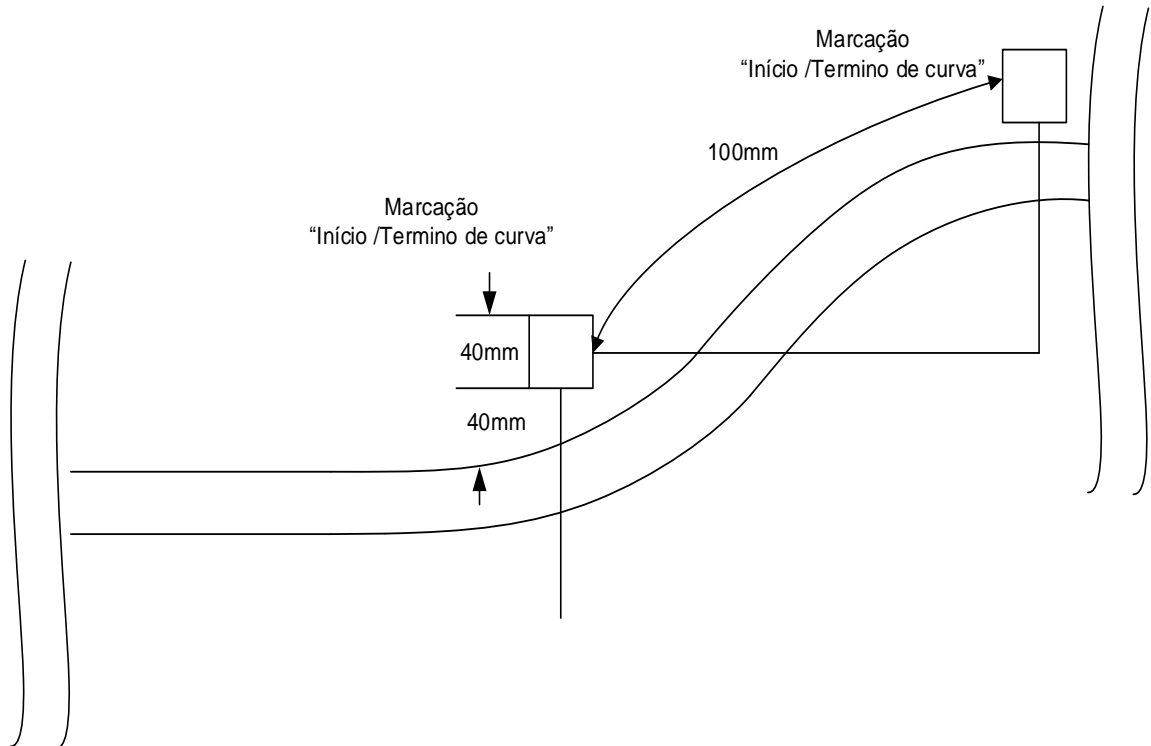


**Figura 16 - Detalhes da área de partida-chegada.**

**Fonte: (Robocore, 2013).**

A linha 250mm antes e 250mm depois da "área de partida-chegada" será reta.

O raio dos arcos será de pelo menos 100mm. Cada arco terá também um comprimento mínimo de 100mm (Figura 17).



**Figura 17 - Exemplo de curva no percurso.**

**Fonte: (Robocore, 2013).**

Haverá uma marcação no lado esquerdo da linha (em relação ao sentido do percurso) no ponto em que houver alteração da curvatura (Figura 17).

O circuito deverá ser totalmente plano. Porém poderá ser incluído inclinações de até no máximo 5°. As cores da linha e da pista estão sujeitas à possíveis variações, dependendo do fornecedor dos materiais utilizados na confecção dos mesmos, porém existe um grande contraste entre o branco da faixa e o preto da pista.

- **Competição**

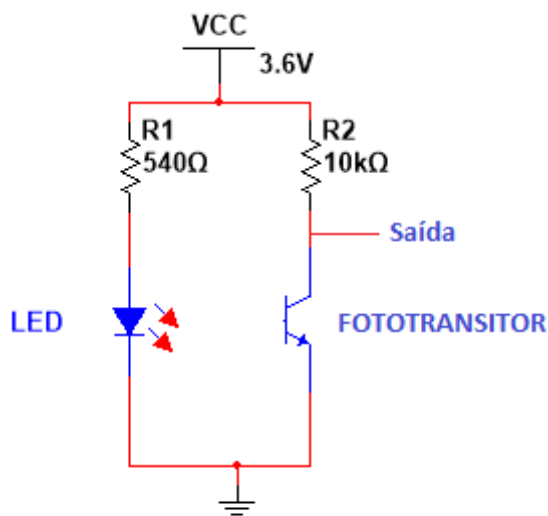
Seguidor de linha é uma competição na qual robôs autônomos devem seguir um trajeto determinado por uma linha. Vence o robô que finalizar o trajeto em menor tempo.

O robô deverá percorrer o circuito tomando como referência a linha branca. O corpo do robô deverá sempre ficar sobre a linha. Caso o robô saia completamente de cima da linha branca, será considerado que o robô saiu do percurso e invalidada a volta.

## 4.2 PROJETO E CONDICIONAMENTO DOS SISTEMAS DE AQUISIÇÃO DE SINAIS CONTÍNUOS.

A variável a ser controlada no processo contínuo é a posição do robô encima da faixa continua central. Para orientação do robô e detecção dessa faixa foi utilizado sensores de refletância do tipo fotoelétricos.

Este sensor possui um LED emissor de luz infravermelho e um fototransistor. O circuito foi ligado conforme o diagrama na Figura 18.

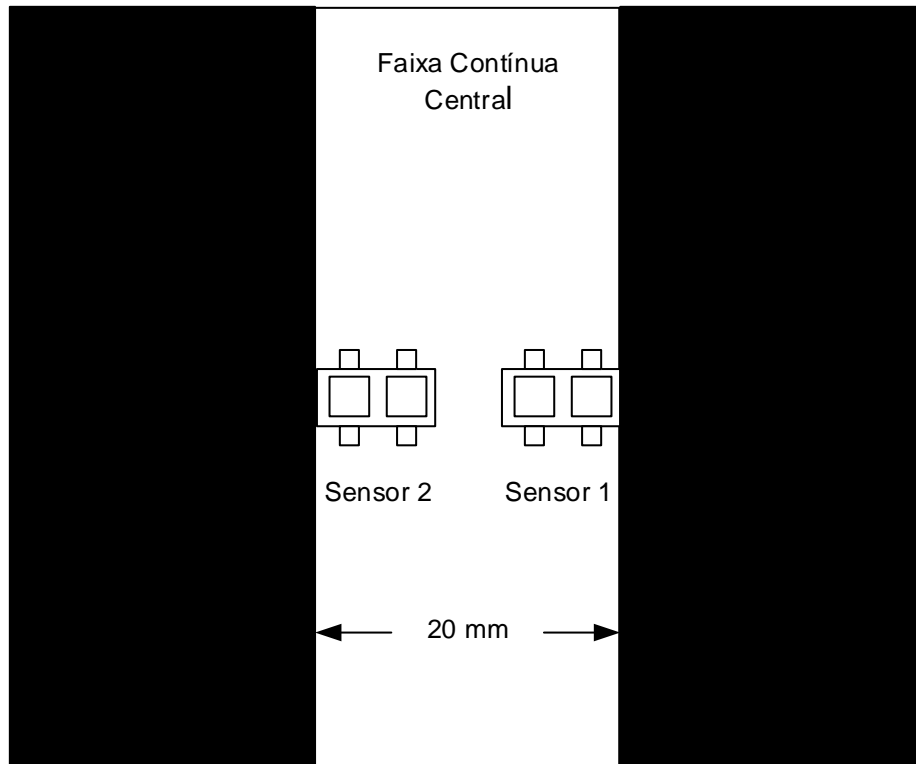


**Figura 18 - Esquemático de ligação dos sensores.**

**Fonte: Autoria Própria.**

A tensão de saída irá variar conforme a refletância (feixe de luz refletido dividido pelo feixe de luz incidente) da superfície, ou seja, quanto maior a refletância menor a tensão de saída, pois o transistor irá ter uma maior condutividade.

O robô possui dois sensores para controle de posição localizados encima da faixa branca central (vide Figura 19), esta posição foi escolhida e adotada como referência para a modelagem do processo, sendo que qualquer deslocamento que ocorrer o controlador entrará em ação para levar o robô de volta a sua posição inicial (referência).



**Figura 19 - Posicionamento dos sensores sobre a faixa central.**

**Fonte: Autoria Própria.**

Para fazer o condicionamento de sinal vindo dos sensores foi unido em uma única forma de onda, de tal forma que o sinal de saída varie proporcionalmente em função da posição do robô encima da faixa. Neste circuito de condicionamento ( Figura 20) é utilizado um circuito somador, e um subtrator, além de um ganho, estes circuitos foram projetados para se adequar às especificações de tensão do microcontrolador, que atua em uma faixa de 0 a 3,6 Volts.

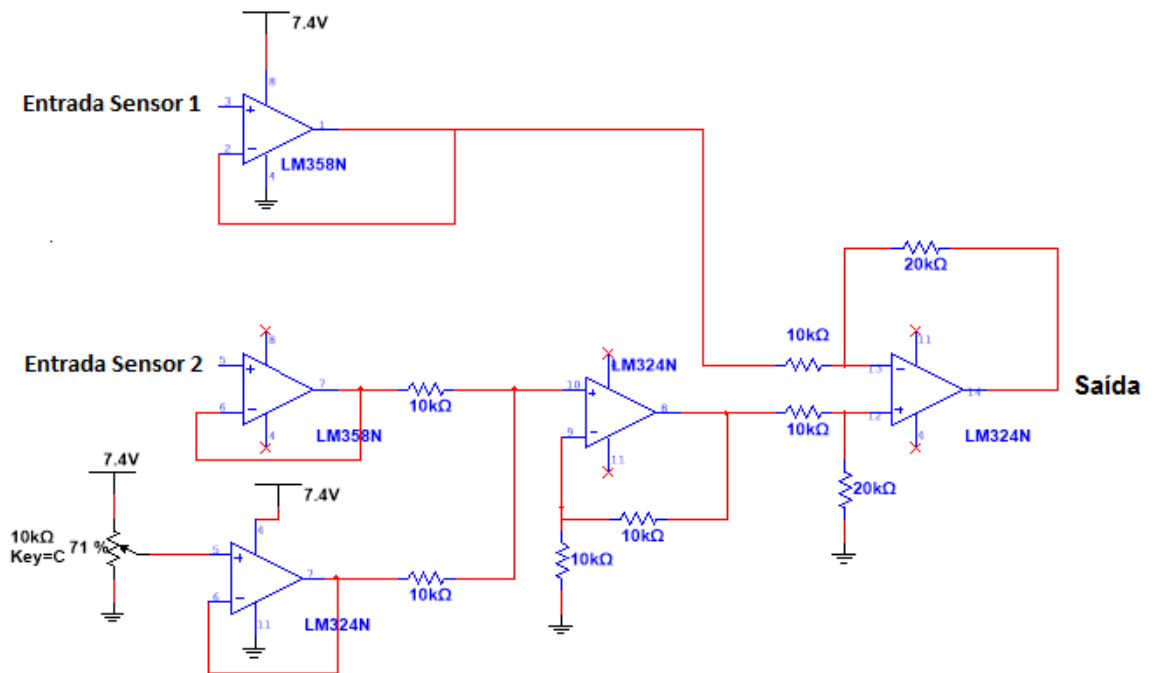
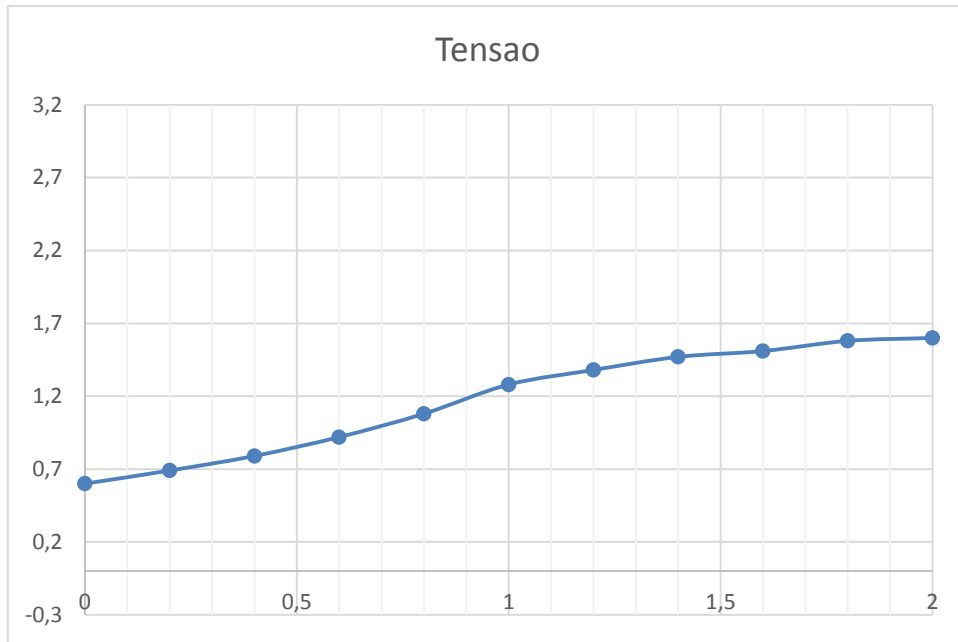


Figura 20 - Circuito de condicionamento das variáveis contínuas.

Fonte: Autoria Própria.

O sinal do sensor da direita é somado junto com um ganho regulado (ajustado conforme o necessário visando a saída com no máximo 3,6 V), este sinal resultante passa por um subtrator que diminui com o sensor da esquerda (vide Figura 20), onde o sensor 1 é o sensor da direita e o sensor 2 é o sensor da esquerda, fazendo a análise do circuito a saída é:  $V_{saida} = 2 \cdot (2 \cdot (V_{sensor\ 2} + V_{ganho}) - V_{sensor\ 1})$ .

Faz-se um gráfico “posição x tensão” para verificar se a tensão está de acordo com o esperado (não mais que 3,6 V) e também para definir a tensão de referência. Este gráfico pode ser visto na Figura 21.



**Figura 21 - Gráfico "Tensão x Posição".**

**Fonte: Autoria Própria.**

Com este gráfico é escolhida a tensão referência, onde os sensores se encontram no meio da faixa (posição 1 cm) que é 1,28 V, também conclui-se que a tensão está de acordo com o adequado para o microcontrolador.

Estando condicionado o sinal analógico ele é ligado a uma das portas de entrada do microcontrolador, o qual utilizará o seu conversor de 10 bits para a conversão analógica-digital, sendo programado a uma frequência de 400 Hz, com um tempo de amostragem de 0,0025 segundos, o suficiente para o robô funcionar adequadamente. O erro de quantização é dado por  $(E_q = V / 2^{n-1})$  (Tensão de operação dividido pelo número de bits do conversor menos um)  $E_q = (3,6/1023) = 0,003$ .

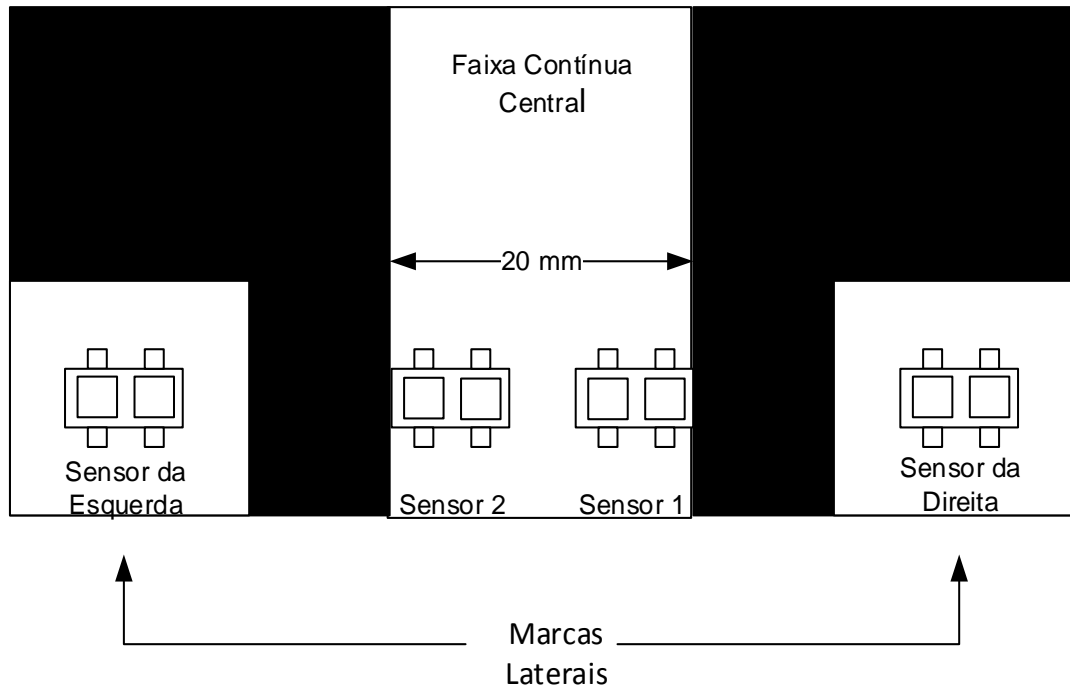
### 4.3 PROJETO E CONDICIONAMENTO DOS SISTEMAS DE AQUISIÇÃO DE SINAIS DISCRETOS.

Os eventos discretos presentes no robô seguidor de linha são provenientes das marcas laterais situadas em ambos os lados, que servem de



orientação para sua locomoção. Como já citadas as marcas da direita são de auxílio para início e fim de trajeto, e as marcas da esquerda para início e fim de curva, nos cruzamentos elas apareceram em ambos os lados.

Para identificação dessas marcas será usado dois sensores de refletância, localizados no meio das marcas laterais conforme a Figura 22.



**Figura 22 - Posição dos sensores laterais e centrais**

**Fonte: Autoria Própria.**

Para condicionamento do sinal, as saídas dos sensores passam por um circuito comparador (um para cada sensor lateral) (Figura 23) que irá transformar a variação da refletância em um sinal discreto alto (1) ou baixo (0) antes de chegar ao microcontrolador, informando a detecção da faixa. Como na presença da faixa a tensão de saída é 2,54 V e fora da faixa (escuro) é 3,46 V é adotado uma tensão de referência de 3,0 V para comparação, podendo ser ajustada através de um potenciômetro caso seja necessário.

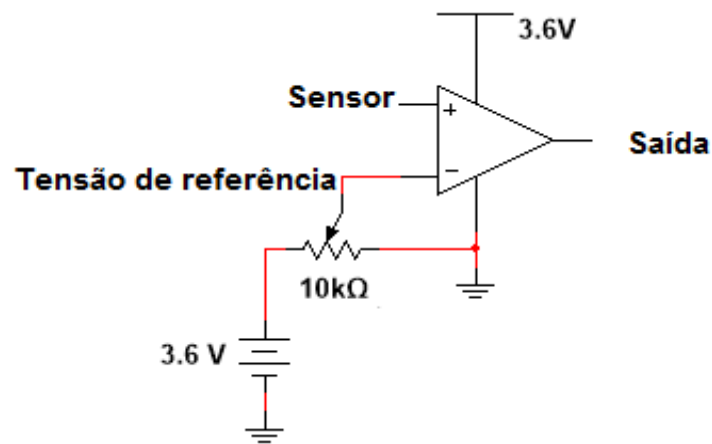


Figura 23 - Circuito comparador para sensores laterais.

Fonte: Autoria Própria.

A saída do comparador irá diretamente para o microcontrolador, este fará a leitura do sinal, interpretando como sinal alto ou baixo através da interrupção do conversor, quando este detectar borda de subida ou descida.

O robô contará também com um chave tátil (*push button*) (vide Figura 24), para início de percurso, este item é utilizado para dar a largada após o robô estar ajustado devidamente no lugar de partida, com os sensores na posição correta sobre a faixa central, somente após esta chave ser pressionada é que o robô começara a andar.

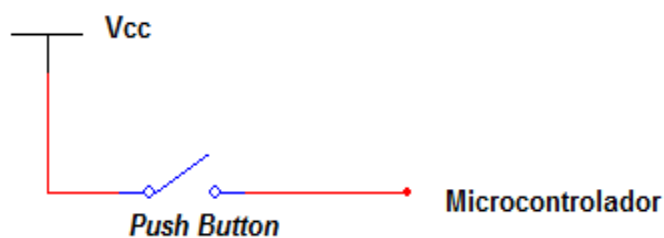


Figura 24 - Esquema de ligação do *push button*.

Fonte: Autoria Própria.

Esta chave como pode se ver na figura é conectada entre o  $V_{cc}$  (3,6 V) e o microcontrolador gerando uma borda de subida ao ser pressionado.

#### 4.4 LEVANTAMENTO EXPERIMENTAL DO MODELO MATEMÁTICO DO PROCESSO DE VARIÁVEIS CONTÍNUAS E PROJETO DO CONTROLADOR.

Fazendo um teste com o sensor do lado direito acima da faixa e o esquerdo fora, é inserido um degrau no motor da esquerda, assim os sensores passam pela linha branca e vão para o outro lado (direito) totalmente fora da faixa. Este procedimento resulta no seguinte sinal:

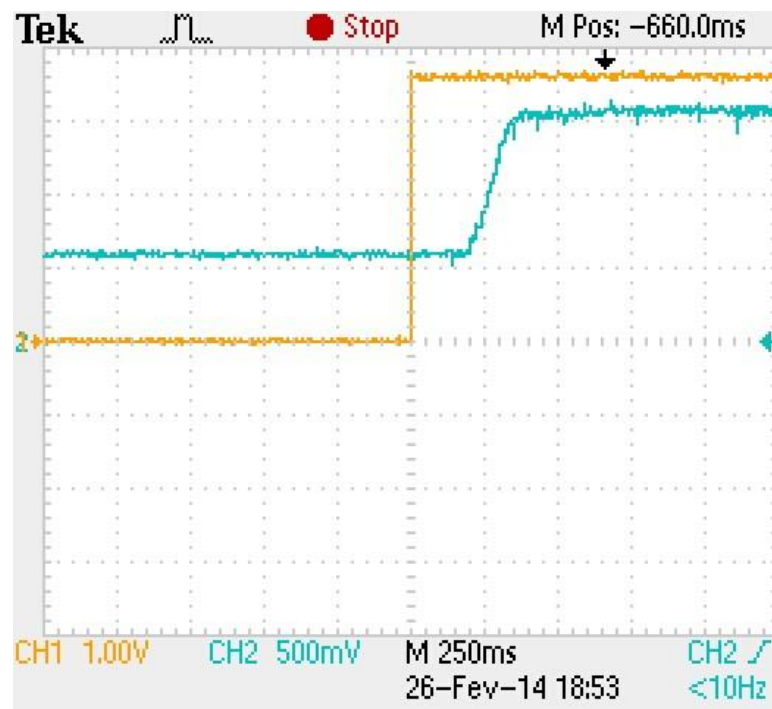


Figura 25 - Saída do circuito de condicionamento das variáveis contínuas.

Fonte: Autoria Própria.

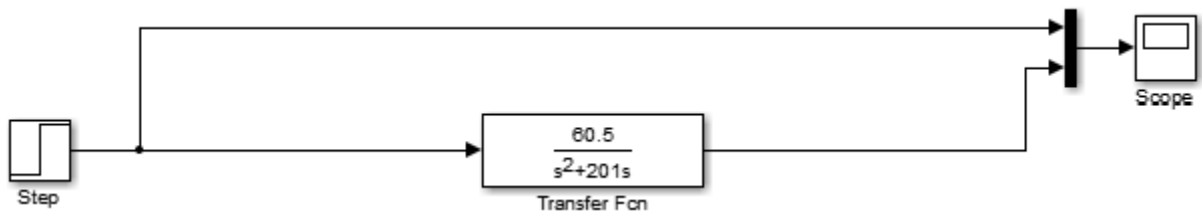
Com esta curva pode-se retirar a função de transferência da planta, tem-se um degrau de entrada que vai de 0 a 90% de PWM, ou seja, de 0 a 3,24 V, e a saída com uma variação de 0,6 a 1,6 Volts em 320 ms.

Sabe-se que a posição (função desejada) é a integral da velocidade no tempo:  $X(t) = \int_0^t v(t)dt$ , aplicando Laplace  $X(s) = V(s)/s$ , então a função de transferência possui o seguinte formato:  $G(s) = K * a/(s(s + a))$  onde  $K = \frac{\text{Entrada}}{\text{Saída}}$ .

Como pode-se observar na Figura 25, o tempo em que o sistema leva 50,25ms para entrar em regime, então  $a = (4/0,05025)$ ,  $a = 201$  e  $K = (1,6 -$

$0,6)/(3,24 - 0) = 0,30$ , a função de transferência do sistema é:  $G(s) = 60,5/(s(s + 201))$ .

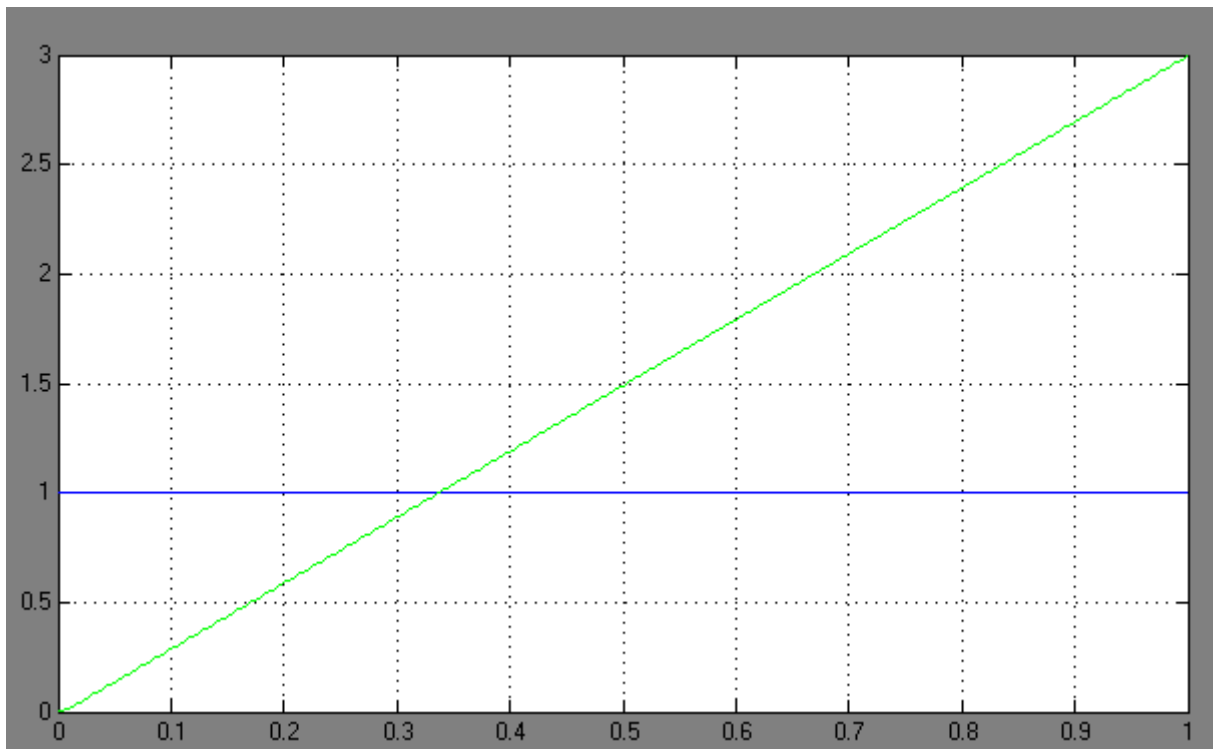
A seguir é verificada a resposta da função de transferência (Figura 26) a um sinal degrau na entrada:



**Figura 26 - Planta do sistema no Simulink.**

**Fonte: Autoria Própria.**

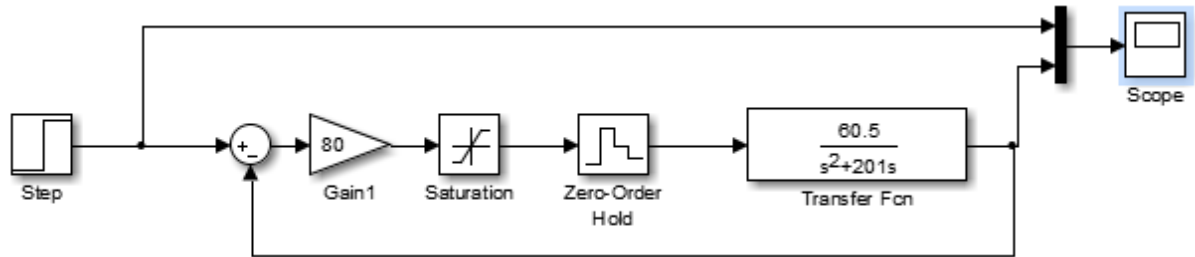
Observa-se que a resposta obtida com o modelo matemático (Figura 25) aproxima-se do obtido experimentalmente (Figura 27)



**Figura 27 - Resposta do sistema a um degrau na entrada.**

**Fonte: Autoria Própria.**

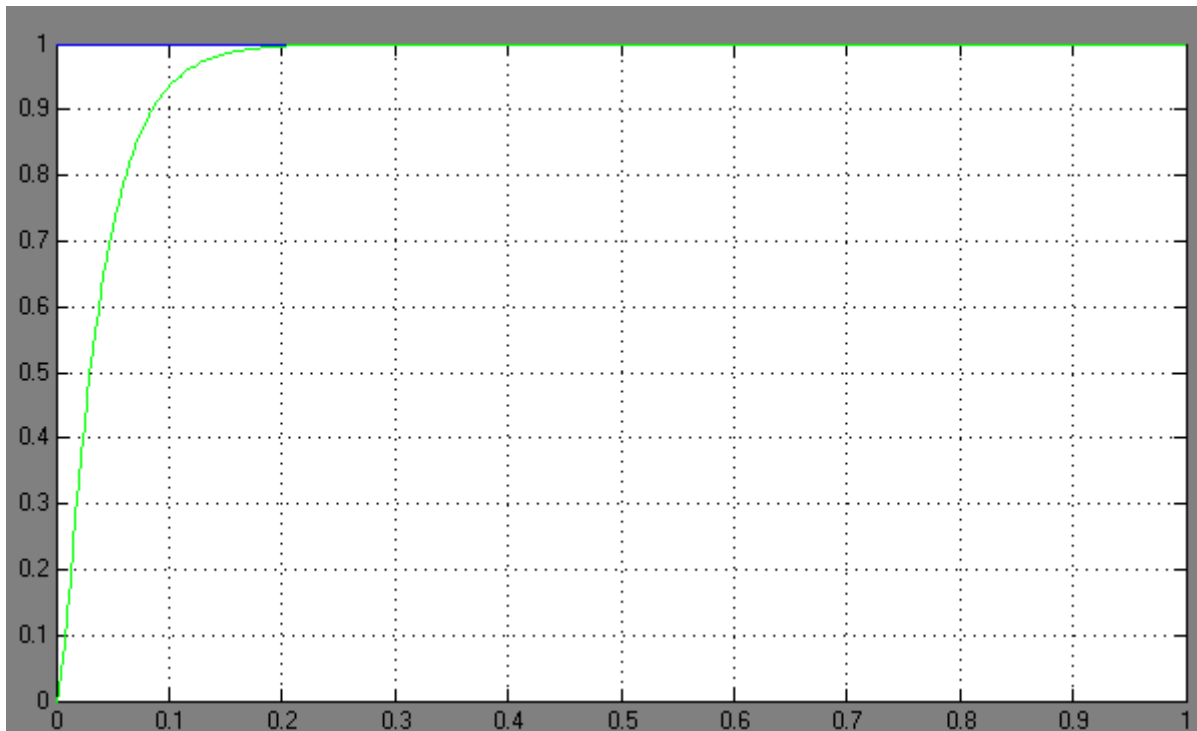
A seguir é implementado um controlador digital, sendo testado qual o melhor tipo para este caso, vale lembrar que a própria planta do sistema já tem um integrador ( $1/s$ ). É simulado no Simulink o controlador digital:



**Figura 28 - Malha de controle.**

**Fonte: Autoria Própria.**

Os ganhos são ajustados através do Simulink para uma saída estável, e também é inserido um saturador a fim de evitar que estoure o valor máximo de saída do PWM. Primeiramente é visto se somente com um ganho P ele aumenta a velocidade de resposta sem ficar instável, foi aumentado o ganho  $K_p$  gradativamente até obter uma resposta satisfatória, como pode ser visto na Figura 29.



**Figura 29 - Saída da malha de controle.**

**Fonte: Autoria Própria.**

Sendo o sinal verde a resposta do sistema e o sinal em azul o degrau inserido.

Com o ganho ajustado ( $K_p = 80$ ), é feita a programação do controlador proporcional no microcontrolador. Como foi escolhido uma frequência de amostragem de 400hz (tempo de amostragem de 2,5 ms), visto no tópico de condicionamento de sinal, este tempo é bem abaixo do tempo de deslocamento dos sensores na faixa central (320 ms), sendo que este tempo é para um degrau em apenas um motor, no caso real isto não acontecerá, estão o tempo de amostragem do conversor está de acordo com o necessário para o robô funcionar corretamente (não “perder” a faixa central).

#### 4.5 MODELAGEM DO PROCESSO DE VARIÁVEIS DISCRETAS.

Para esta modelagem foi feito o uso do software Deslab. Este software é uma ferramenta que permite realizar operações com autômatos (redução, produto síncrono entre outros) e tem a função de exportar código fonte para implementação do modelo em algumas famílias de microcontroladores (PIC, MSP, Stellaris e outros).

As variáveis a serem controladas são os dois sensores laterais e botão de partida, para isso é feito um levantamento do processo para depois fazer a sua implementação no software Deslab.

Este modelo do processo discreto foi projetado visando atender as seguintes especificações:

- Uma chave táctil é pressionada para dar início à corrida (robô encontra-se parado no seu estado inicial)
- O Robô precisa se desligar sozinho após detectar a marca de chegada.
- Nas retas o robô anda com 100% de velocidade.
- Nas curvas o robô reduz a velocidade para 75%.
- O número de curvas do trajeto varia de uma competição para outra, ou seja, o número de curvas não é conhecido.

- O número de cruzamentos do trajeto também varia de uma competição para outra.

Para implementação do modelo foi adotado a máquina de Moore, onde as ações de saída (valor de operação do PWM) estará nos estados. Os eventos adotados para esse processo são definidos como:

AD: Acha marca da direita.

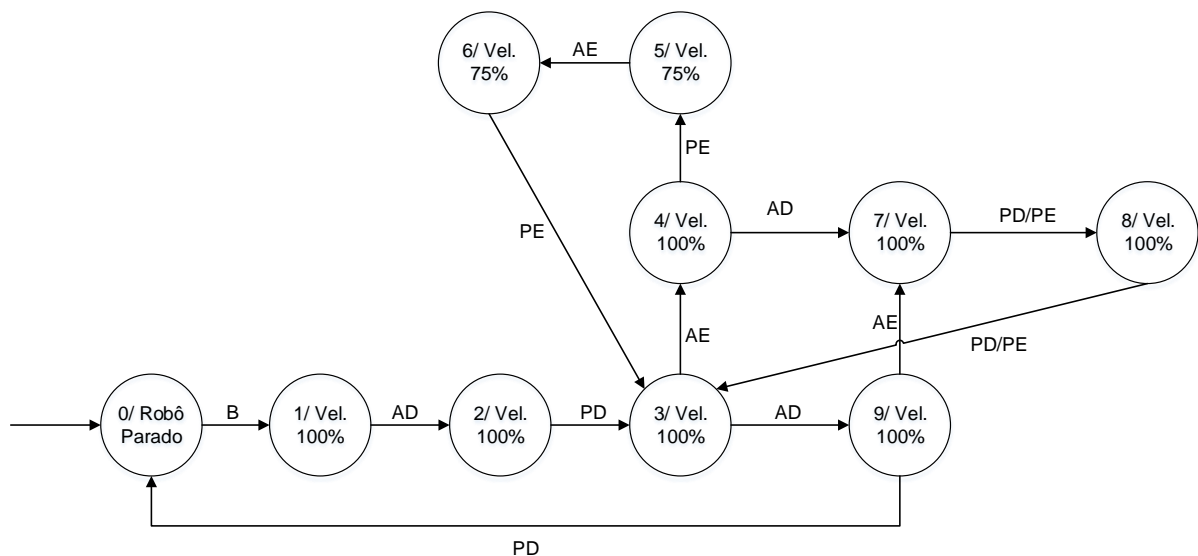
PD: Perde marca da direita.

AE: Acha marca da esquerda.

PE: Perde marca da esquerda.

B: Botão de início.

Os valores dentro dos estados correspondem a velocidade do robô, sendo nas retas (100%), curvas (75%) e robô parado (estado inicial) (0%). O modelo projetado é visto na Figura 30. Este valor de velocidade foi ajustado para que o robô percorra o mais rápido possível sem sair do trajeto.



**Figura 30 - Diagrama de controle discreto.**

Fonte: Autoria Própria.

Na sequência é realizada uma análise de cada estado do robô:

Estado 0: Robô desligado, esperando o *push button* ser pressionado para iniciar o percurso.

Estado 1: Robô na área de chegada e partida com os motores no ponto de operação para retas (PO=100), próximo evento será achar a marca da direita (AD).

Estado 2: Achou a linha branca de início de percurso, agora ele irá “perder” a linha esta linha (PD).

Estado 3: Continua na reta e espera o próximo evento que pode ser achar a marca da esquerda, como também achar a marca da direita.

Estado 4: Acha 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 5: Perde a marca da esquerda, então ele entrou na curva, muda o ponto de operação (PO=75%). Espera o próximo evento que é achar a marca da esquerda, que diz que vai terminar a curva.

Estado 6: Achou a marca da esquerda, está se alinhando para entrar novamente na reta, por isso seu ponto de operação não muda ainda. Só irá mudar para o ponto de operação depois que perder a marca da esquerda e retornar para o estado 3 (PO=100%).

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: Perdeu a marca da direita ou esquerda, está deixando o cruzamento, irá perder novamente uma das duas marcas e irá retornar ao estado 3, não muda o ponto de operação.

Observação: As transições entre o estado 7 e 8 e depois o 3 ocorrem da seguinte maneira, quando o robô irá 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 3, ou seja, saiu do cruzamento).

Estado 9: 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 0 (motores desligam).

Após levantado o diagrama do processo discreto é feito a implementação no software Deslab, este tem a capacidade de transformar esse diagrama em um código para Microcontroladores.

Para implementação foi adotado uma regra, sendo que eventos externos ao microcontrolador recebem um número par:

B: Botão de início (2).



AD: Acha marca da direita (4).

AE: Acha marca da esquerda (6).

PD: Perde marca da direita (8).

PE: Perde marca da esquerda (10).

Este processo não possui eventos discretos internos (Eventos gerados pelo controlador), caso tivesse teria de receber números ímpares. Todo essa metodologia é para auxiliar na implementação e ajuste do código. Na Figura 31 o diagrama transcrito no software Deslab.

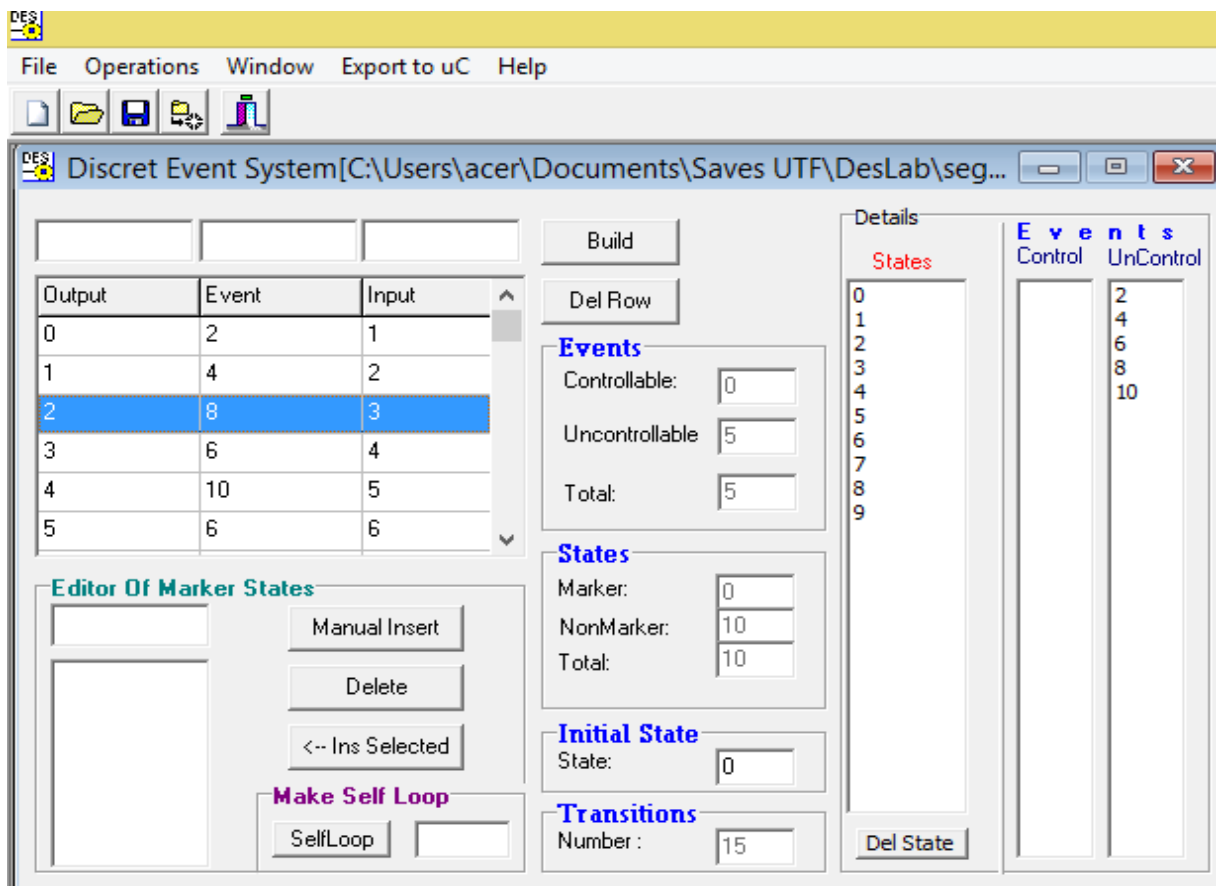


Figura 31 - Processo implementado no Deslab.

Fonte: Autoria Própria.

O robô possui dez estados e quinze transições, os cinco eventos presentes são ditos não controláveis, ou seja, podem acontecer a qualquer momento e o controlador não tem capacidade de inibi-los, são eventos já ditos como externos, todas essas informações também podem ser visualizadas na Figura 31.

Após a implementação, é exportado (através de um função do software) esse diagrama para uma linguagem de microcontrolador MSP (sendo utilizado um

microcontrolador MSP430g2553) este código já contém toda a parte de gerador de eventos e variáveis do autômato,

Foi definida quais as portas de entrada, as ações dos estados, e configurado as interrupções nas portas por borda de subida ou descida.

#### 4.6 IMPLEMENTAÇÃO DO CONTROLE HÍBRIDO.

O controle híbrido é a junção do controle das variáveis contínuas com o controle discreto, ambos atuam simultaneamente no controle do robô, o primeiro terá a função de manter o robô encima da faixa central, e o segundo atuará quando surgir as marcas laterais. Na Figura 32 apresenta-se um esquemático de como funciona o controle hibrido proposto neste trabalho.

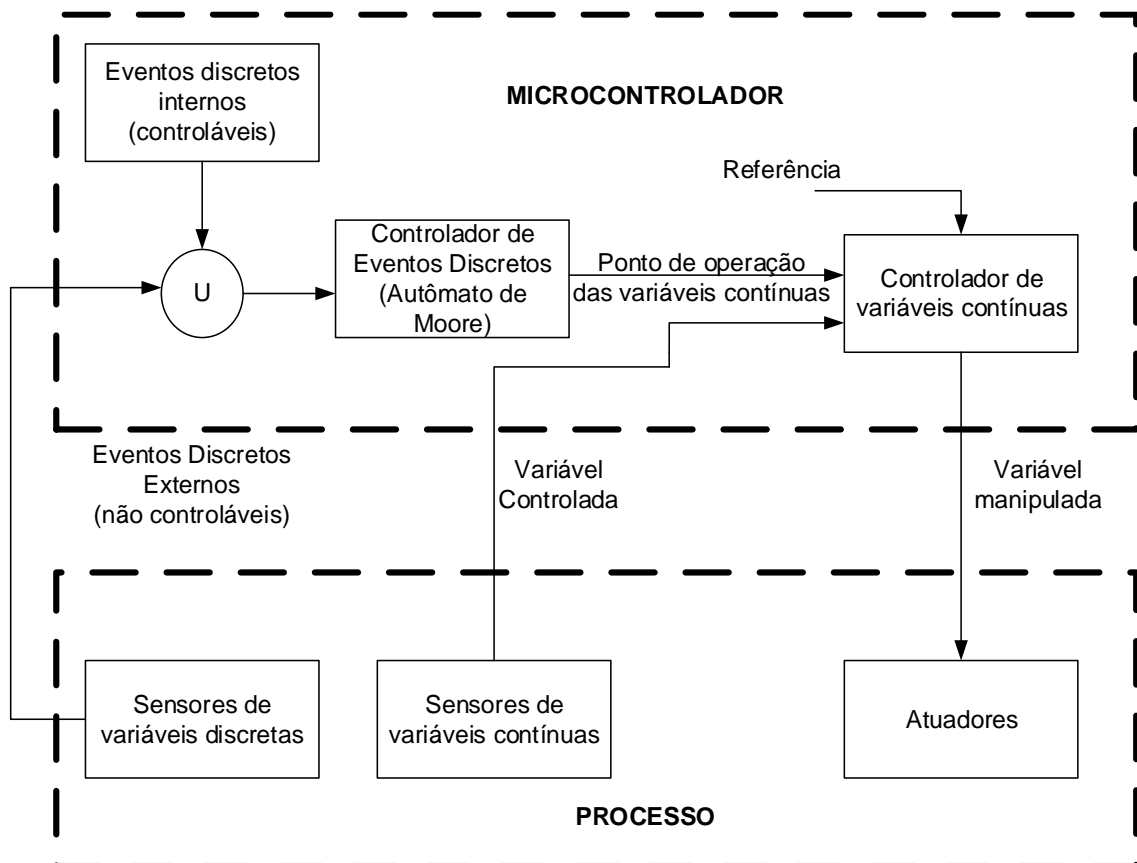
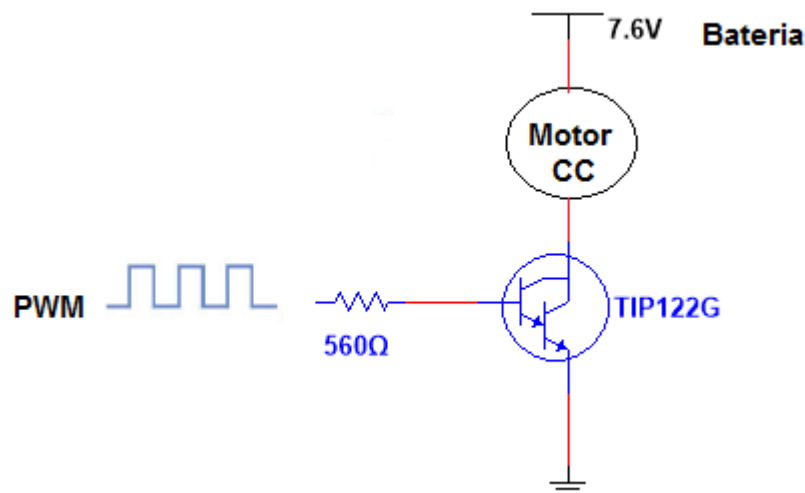


Figura 32 - Estrutura do sistema de controle híbrido.

Fonte: Autoria Própria.

No processo encontram-se as variáveis do sistema, seja elas contínuas ou discretas (faixa central e marcas laterais), elas são condicionadas de forma a serem lidas ou detectadas pelo microcontrolador. No processo está também os atuadores que são dois motores controlados individualmente por sinal PWM vindo do microcontrolador.

No circuito de acionamento (um para cada motor) foi utilizado como chave um transistor TIP 122, este é um transistor NPN do tipo darlington (ou seja: são dois transistores dentro do mesmo invólucro, um excitando o outro, de maneira a se aumentar muito o ganho total). Ele foi desenvolvido para aplicações de chaveamento (tanto que já possui um diodo interno para a proteção da junção coletor/emissor do transistor), foi usado este tipo de transistor devido à baixa tensão que vem do microcontrolador, sendo que transistor consegue dar um alto ganho a este sinal para o chaveamento do circuito (Figura 33)



**Figura 33 - Circuito de acionamento de um motor.**

**Fonte: Autoria Própria.**

Esta ação de controle PWM é constituída de um ponto de operação, definido pelo controlador discreto, sendo ele mais alto nas retas e mais baixo nas curvas (visto na modelagem discreta), e este ponto de operação é somado ou diminuído com a ação de controle P (definida pelo controlador contínuo):

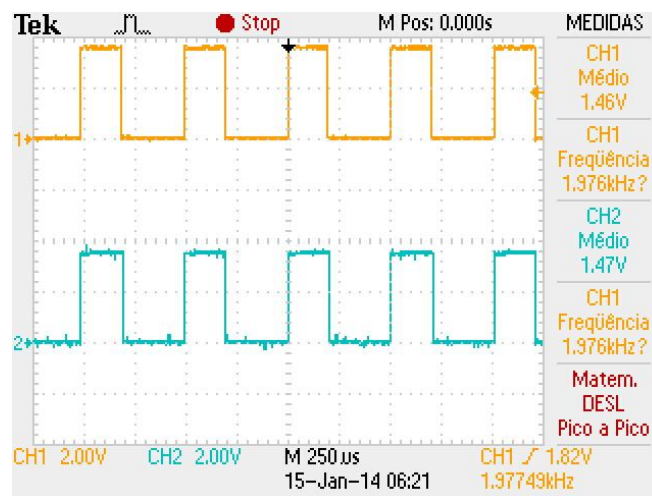
$$PWM = PO \pm P$$

Para o motor da direita irá somar a ação P e para o motor da esquerda irá diminuir, isto conforme as simulações e testes realizados anteriormente.

É implementado o controlador P no código para o microcontrolador, adotado como referência o valor de 1,28 Volts, este valor indica os dois sensores acima da faixa branca central.

## 5 TESTES E RESULTADOS

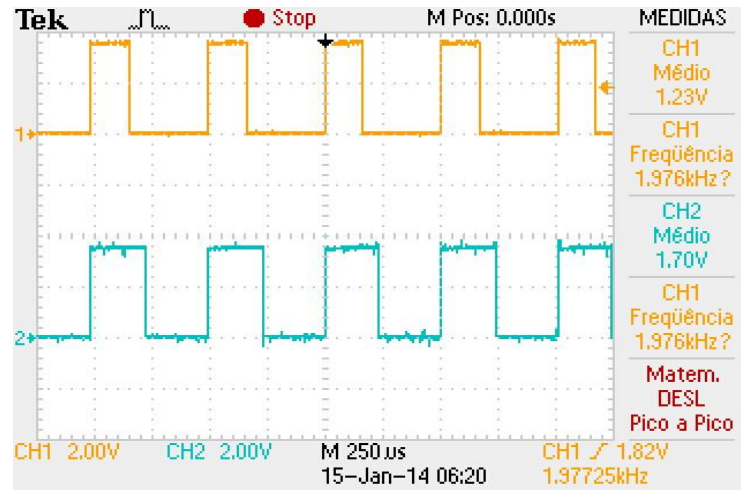
É testado o funcionamento do controlador proporcional através das saídas dos PWMs. As figuras a seguir mostram a variação dos PWMs conforme a posição dos sensores. Na Figura 34 o robô encontra-se na referência (com os dois sensores em cima da faixa central) e pode-se observar os seguintes detalhes: o sinal PWM enviado a cada motor tem o mesmo ciclo de trabalho, mantendo a mesma velocidade para ambos os motores.



**Figura 34 - PWM para sensores na faixa central, sinal em laranja: PWM do motor da direita, sinal em azul: PWM do motor da esquerda.**

**Fonte: Autoria Própria.**

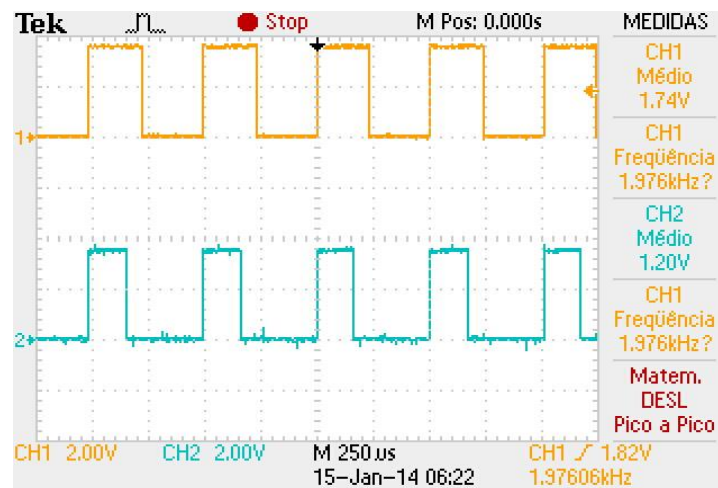
Na Figura 35 pode se ver que os sinais PWM são diferentes para cada motor, neste caso o robô está saindo da faixa central e indo para o lado esquerdo, então o motor da direita diminui sua velocidade enquanto o da esquerda aumenta para voltar ao centro da faixa.



**Figura 35 - PWM para sensor da esquerda fora da faixa central, sinal em laranja: PWM do motor da direita, sinal em azul: PWM do motor da esquerda.**

Fonte: Autoria Própria.

Já a Figura 36 o robô está saindo para o lado direito da faixa central, então o motor da direita aumenta sua velocidade enquanto o da esquerda diminui.



**Figura 36 - PWM para sensor da direita fora da faixa central, sinal em laranja: PWM do motor da direita, sinal em azul: PWM do motor da esquerda.**

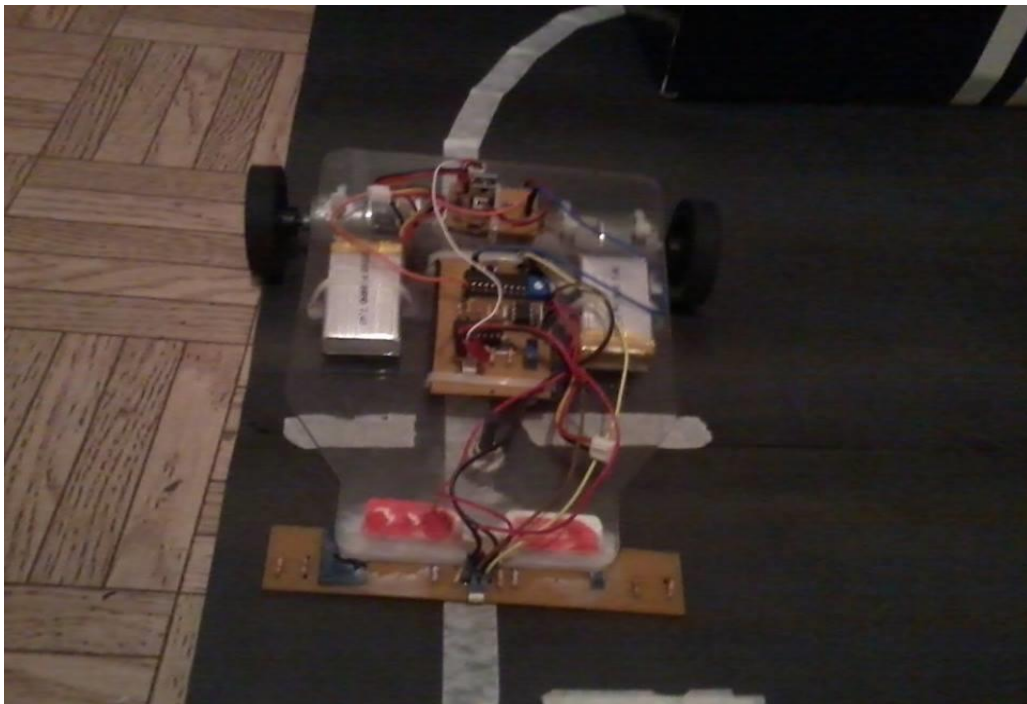
Fonte: Autoria Própria.

Após testado essas variações de tensão para os motores, foi construído uma pista de acordo com as normas da organizadora de competições Robocore, e testado o funcionamento do robô neste percurso.

Testou-se várias vezes o robô no percurso que contava com quatro curvas e dois cruzamentos, as faixas e marcas foram feitas com uma fita crepe em cima de folhas dupla face pretas, inicialmente foram feitos cinco testes, em três destes testes o robô não reconheceu as marcas laterais devido que a superfície não estava conforme a inclinação recomendada (máximo 5°).

Foram feitos novos testes com a superfície devidamente plana, e a cada oito testes somente um não reconheceu as faixas laterais. Também foi aumentando o valor do ponto de operação para 82% nas curvas, e testado o robô no percurso, sendo que neste teste o robô não conseguiu realizar as curvas.

O robô desenvolvido apresenta o aspecto mostrado na Figura 37, feito sobre uma placa de acrílico, com um suporte na parte frontal onde é possível ajustar a altura dos sensores.



**Figura 37 - Robô seguidor de linha.**

**Fonte: Autoria Própria.**

Foram utilizados dois motores de corrente contínua com caixa de redução para aumentar o torque na partida, com isso obteve-se uma boa velocidade de partida comparado com outros motores sem essa caixa de redução. A discussão dos resultados obtidos será apresentado no próximo capítulo.

## 6 CONCLUSÕES

Este trabalho apresentou técnicas de condicionamento de sinais de tempo contínuo e de tempo discreto. Para o sistema de tempo contínuo conseguiu-se uma relação proporcional entre a refletância e posição dentro da faixa desejada. Desde que as variações de posição do robô em cima da faixa é pequena, a relação refletância *versus* posição pôde ser considerada linear para efeitos de controle. O condicionamento do sinal para os eventos discretos foi bastante simples, utilizando apenas um comparador, mas com uma enorme importância para a implementação no microcontrolador.

O levantamento do modelo matemático em sistemas contínuos, feito através da modelagem experimental, a partir da qual foi possível realizar o projeto do controlador, utilizando métodos formais e softwares já conhecidos como o Simulink. Com o uso desses métodos formais tornou o desenvolvimento do controlador mais simples e prático, pois com estes métodos, pode-se realizar simulação, verificação deste projeto de controlador.

A modelagem de variáveis discretas, feita através da teoria de autômatos, conseguiu representar conforme esperado os eventos discretos presentes no percurso do robô. Com este método formal de modelagem foi possível atingir os objetivos sem nenhuma falha. Vale destacar que métodos formais, devido a sua sistemática, permitem desenvolver ferramentas (softwares) como o Deslab para geração automática de código para microcontroladores. Apresentou-se uma estrutura de implementação de controle híbrido, o qual poderá ser utilizada em outras aplicações que demandem a integração de sistemas a eventos discretos e de variáveis contínuas. Com a implementação do controle híbrido (integração dos controladores) em um robô seguidor de linha utilizando microcontrolador conseguiu-se controlar o posicionamento do robô e também a detecção dos eventos discretos simultaneamente.

Pode se dizer que o robô funcionou de acordo com o esperado para um percurso de competições da Robocore, reconhecendo todos os eventos (marcas laterais), e também atuando sobre a faixa central, ou seja, o controle híbrido foi executado de forma desejada. Durante os testes o robô encontrou um pouco de dificuldade de reconhecer as marcas laterais, quando a pista construída não estava



totalmente com o recomendado (inclinações de no máximo 5°), foi montado uma nova superfície onde não ocorresse mais este tipo de problema, então a dificuldade foi eliminada, visto com isso que a altura e o ângulo dos sensores com a superfície é de muita importância para que o robô funcione corretamente.

Trabalhos futuros podem ser desenvolvidos a partir desse. Pode-se acrescentar mais algum sensor ao sistema, como por exemplo, um sensor que identifique obstáculos e faça o robô parar, com a metodologia formal utilizada fica simples para acrescentar esse novo evento.

## REFERÊNCIAS

- CAMPOS, Marcos C. **Dinâmica e Controle de Sistemas**. 1999. 7p.Apostila - UFPR, Curitiba, 1999.
- CAMPOS, Mario C. M. M. D.; TEIXEIRA, Herbert C. G. **Controles Típicos de Equipamentos e Processos Industriais**. 2ª. ed. São Paulo : Edgard Blucher, 2010.
- CASSIOLATO, César. **Condicionamento de Sinais Analógicos & Sensores**. Disponível em: <<http://www.smar.com/brasil/artigostecnicos/artigo.asp?id=16>>. Acesso em: 10 Novembro 2013.
- CURY, José E. R. **Teoria de Controle Supervisório**. 2001. 82p.Apostila - Universidade Federal de Santa Catarina, Canela, 2001.
- DA SILVA, Nilson S.; DA SILVA, Henrique Cislagui. **Modelagem Experimental por Delineamento de Misturas**. 2009. 10p.Artigo - T-cota Engenharia e Minerais Industriais Ltda , Tijucas, 2009.
- DATASHEET QRE1113GR. Disponível em: <[https://www.sparkfun.com/datasheets/Robotics/QR\\_QRE1113.GR.pdf](https://www.sparkfun.com/datasheets/Robotics/QR_QRE1113.GR.pdf)>. Acesso em: 20 Outubro 2013.
- DE BRITO, Roberta C.; MARTENDAL, Diogo M. ; DE OLIVEIRA, Henrique E. M. **Máquinas de estados finitos de Mealy e Moore**. 2003. 4p.Artigo - Universidade Federal de Santa Catarina, Florianópolis, 2003.
- E. SILVA, Lazaro et al. **Utilização do estudo de robôs seguidores de linha como estratégia de avaliação na disciplina controle de processos no curso de engenharia de computação**. 2006. 6p.Artigo, Passo fundo, 2006.
- FERREIRA, Elnatan C. **Sensores e condicionamento de Sinais**. 2000. 111p.Curso IE-763 Departamento de Eletrônica e Microeletrônica - Unicamp, Campinas, 2000.
- FITZGERALD, Arthur E.; KINGSLEY JR., Charles; UMANS, Stephen D. **Máquinas Elétricas**. 6ª. ed. São Paulo: Bookman, 2006.
- HEINEN, Farlei Jose. **Sistema de controle híbrido para robôs moveis autônomos**. 2002. 130p.Dissertação de Mestrado, São Leopoldo, 2002.
- INESC. Promoção das Tecnologias de Electrónica e de Telecomunicações. **Protet**, 2007. Disponível em: <<http://protet.inescporto.pt/o-que-e/automacao-industrial/o-que-e-um-automato.html>>. Acesso em: 07 Janeiro 2014.
- KENJO, Tak. **Electric Motors and Their Controls: An Introduction**. Oxford: Oxford Science Publications, 1991.
- LEITE, Antônio C. **Controle Híbrido de Força e Visão de um Manipulador Robótico sobre Superfícies Desconhecidas**. 2005. 143p.Tese Pós graduação, Rio de Janeiro, 2005.

MARIN, Luciene D. O. **Arquitetura neural cognitiva para controle inteligente de robôs móveis em labirintos dinâmicos**. 2010. 239p. Tese de doutorado em Engenharia Elétrica pela Universidade Federal de Santa Catarina, Florianópolis, 2010.

MECATRÔNICA Atual. Disponível em:

<<http://www.mecatronicaatual.com.br/educacao/1179-sensores-ticos>>. Acesso em: 06 Janeiro 2014.

MELlichamp; SEBORG. **Process Dynamics and Control**. 3. ed. Danvers: Wiley, 2010.

MENEGUELE, Bruno E. D. O.; FERREIRA, Fernando P.; ARCANJO, Vinicius D. S. **Robô explorador de labirintos 2D**. 2011. 52p. Monografia UTFPR, Curitiba, 2011.

MENEZES, Blauth. **Linguagens Formais e Autômatos**. 2004. 58p. Departamento de Informática Teórica, Instituto de Informática / UFRGS, Porto Alegre, 2004.

MICHELS, Leandro. **Modelagem de Sistemas Físicos**. 2011. 30p. Apostila - UFSM, Santa Maria, 2011.

NEHMZOW, Ulrich. **Mobile Robotics: A Practical Introduction**. 2. ed. Essex: Springer, 2000.

OLIVEIRA, Adalberto L. D. L. **Fundamentos de controle de processo**. 1999. Trabalho realizado em parceria SENAI / CST (Companhia Siderúrgica de Tubarão), Vitória, 1999.

ROBOCORE , 2013. Disponível em:

<[http://www.robocore.net/upload/attachments/robocore\\_\\_regras\\_seguidor\\_de\\_linha\\_205.pdf](http://www.robocore.net/upload/attachments/robocore__regras_seguidor_de_linha_205.pdf)>. Acesso em: 23 Julho 2013.

SOUZA, Uilian L. D.; PEREIRA, Thiago R. **Fotodiodos e Fototransistores**. 2003. Artigo - UFPR, Curitiba, 2003.

TORRES, André Euler. **Introdução ao funcionamento e ao acionamento de motores DC**. 2004. 14p. Relatório técnico de iniciação científica - UFRJ, Rio de Janeiro, 2004.

TORRICO, César R. C. **Controle Discreto\_parte1**. Disponível em:

<<https://sites.google.com/site/controldiscreto9/slides-disciplina>>. Acesso em: 20 Agosto 2013.

WITSENHAUSEN, Hans. **A class of hybrid-state continuous-time dynamic systems**. 1966. 167p. IEEE, Cambridge, 1966.

ZIEGLER; NICHOLS. **Optimum Settings for Automatic Controllers**. Rochester: Transactions ASME, 1942.

## APÊNDICES

### Código para o Microcontrolador MSP430G2553

```
// Código desenvolvido por Alan Guadagnin
// Controle de um robô seguidor de linha por P

#include "msp430g2553.h"

//Dados do autômato (Não pode ser declarado dentro da função main por ser const)
#define NTRANS 15 //Número de Transições
#define NESTADOS 10 //Número de Estados
const unsigned int event[NTRANS]={2,4,8,6,10,6,10,4,8,4,6,6,10,6,10};
const unsigned int in_state[NTRANS]={1,2,3,4,5,6,3,7,0,8,8,9,9,3,3};
const unsigned int rfirst[NESTADOS] = {1,2,3,8,10,6,7,11,13,15};
const unsigned int rnext[NTRANS] = {0,0,0,0,0,0,4,0,5,9,0,12,0,14};

//mapeamento de eventos nao controlaveis como entradas
#define B 2 //Entrada 0
#define AD 4 //Entrada 1
#define AE 6 //Entrada 2
#define PD 8 //Entrada 3
#define PE 10 //Entrada 4

//Definição de funções de inicialização
void config_clk(void); //Configura Clock
void config_timer(void); //Configura Timer
void config_io(void); //Configura entradas e saídas
void tempo_ms(unsigned int); // Função de temporização do timer A em milisegundos

//Declaração de variáveis globais
unsigned char buffer[5]; //Buffer para armazenar a fila de eventos externos
unsigned char n_buffer=0; //Número de eventos no Buffer
int ref = 335;
int kp=80;
int uc,se,e,up,ui = 0;
unsigned int PO = 0;
unsigned int PWM = 0;

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD; //Desabilita o WDT
    config_clk(); //configura o clk
    config_timer(); //configura o timer
    config_io(); //configura entradas e saídas

    unsigned int k;
    char occur_event; //Evento ocorrido
    unsigned char current_state = 0; //Estado atual inicializado com estado inicial
    char g=0; //Flag para gerador aleatório de eventos
    char gerar_evento=1; //Flag para habilitar a temporização de eventos controláveis
    char moore_output = 0; //Inicializa saída periférica
```

```

_enable_interrupt(); //Habilita interrupção geral

while(1)
{
    if(n_buffer == 0) //se não existir evento no buffer então gerar um
evento interno(evento controlável)
    {
        if(TACTL&TAIFG) //Se o timer estourar, habilita a geração
de eventos
        {
            gerar_evento=1;
        }
        if(gerar_evento==1)
        {
            switch(g) //Aqui é implementado um gerador automático
de eventos controláveis
            {
                {
            }
        }
    }
    else //se existir evento não controlável pegar do buffer
    {
        occur_event = buffer[0];
        n_buffer--;
        k = 0;
        while(k<n_buffer)
        {
            buffer[k] = buffer[k+1];
            k++;
        }
    }

    //Jogador de autômato
    k = rfirst[current_state];
    if(k==0)
    {
        return; //Dead Lock!!!
    }
    else
    {
        while(k>0)
        {
            k--;
            if(event[k] == occur_event)
            {
                current_state = in_state[k];
                moore_output = 1;
                break;
            }
            k = rnext[k];
        }
    }

    if(moore_output) //Se o evento ocorrido for válido, então imprimir
saída física
    {
        switch(current_state)
        {

```

```

        PO = 0;
        uc = 0;
        case(0):
            break;
        case(1):
            PO = 600;
            break;
        case(2):
            PO = 600;
            break;
        case(3):
            PO = 600;
            break;
        case(4):
            PO = 600;
            break;
        case(5):
            PO = 450;
            break;
        case(6):
            PO = 450;
            break;
        case(7):
            PO = 600;
            break;
        case(8):
            PO = 600;
            break;
        case(9):
            PO = 0;
            uc = 0;
            break;
    } //fim switch
    moore_output = 0;
    occur_event = -1;
} //fim if(mealy_output)
} //fim while(1)
} //fim main

//Tratamento da Interrupção da porta P1
//-----
//-----
#pragma vector=PORT1_VECTOR
__interrupt void RTI_PORT1(void)
{
    {
        if((P1IFG&BIT3)&&(P1IN&BIT3)) //verifica se P1.3 foi presionado
        {
            P1IFG&=~BIT3; //apaga o flag
            buffer[n_buffer]=B; //Atribuir evento a P1.3
            n_buffer++;
        }
        if((P1IFG&BIT1)&&!(P1IN&BIT1)) //verifica se P1.1 foi presionado
        {
            P1IFG&=~BIT1; //apaga o flag
            buffer[n_buffer]=AD; //Atribuir evento a P1.1
            n_buffer++;
            P1IES &=~ BIT1;
        }
    }
}

```

```

    }
    if((P1IFG&BIT1)&&(P1IN&BIT1)) //verifica se P1.1 foi presionado
    {
        P1IFG&=~BIT1; //apaga o flag
        buffer[n_buffer]=PD; //Atribuir evento a P1.1
        n_buffer++;
        P1IES |= BIT1;
    }

    if((P1IFG&BIT2)&&!(P1IN&BIT2)) //verifica se P1.2 foi presionado
    {
        P1IFG&=~BIT2; //apaga o flag
        buffer[n_buffer]=AE; //Atribuir evento a P1.2
        n_buffer++;
        P1IES &=~ BIT2;
    }
    if((P1IFG&BIT2)&&(P1IN&BIT2)) //verifica se P1.2 foi
presionado
    {
        P1IFG&=~BIT2; //apaga o flag
        buffer[n_buffer]=PE; //Atribuir evento a P1.2
        n_buffer++;
        P1IES |= BIT2;
    }

    P1IFG=0;
}

#pragma vector=ADC10_VECTOR
__interrupt void ADC10ISR(void){

    PWM = ADC10MEM;

    e=(PWM - ref);
    up=kp*e;
    uc=up;

    //saturadores
    if(uc > 190) { uc=190;
    }
    if(uc < (-190)) { uc= (-190);
    }

    if(PO ==0)uc=0;
    TA1CCR1 = PO + uc;
    TA1CCR2 = PO - (uc);

}

//Configuração de perifericos
//-----
//-----
void config_clk(void) //configura o clock
{

```

```

    WDTCTL = WDTPW + WDTHOLD;           // Desativando o WatchDog
Timer
    DCOCTL = CALDCO_16MHZ;             // Load 1MHz constants
MCLK=SMCLK.
    BCSCTL1 = CALBC1_16MHZ;
    BCSCTL1 |= XT2OFF;                 //desliga oscilador XTE
    BCSCTL2 |=DIVS_3;                  //divide SMCLK por 8 ==2MHz
    BCSCTL3 |= XCAP0 +XCAP1;           //configura os capacitores para o oscilador
externo ACLK = 32768 Hz
    __enable_interrupt(); ////Seta o bit GIE - "chave geral" das
interrupções
}

//-----
//-----
void config_timer(void) //configura o timer
{

    TA1CTL = TASSEL_2 + MC_1; //utiliza clock secundario 2Mhz
    TA1CCTL1 =OUTMOD_7;        // Modo 7: Reset/Set (comparacao)
    TA1CCTL2 =OUTMOD_7;        //
    TA1CCR0 = 1000;            //frequencia do timer 2khz
    TA1CCR1 = 0;
    TA1CCR2= 0;

    //configurado para gerar gatilho para a conversao (AD) do timer

    TA0CTL = TASSEL_2 + MC_1;
    TA0CCTL1 = OUTMOD0 + OUTMOD1 + OUTMOD2; // Modo 7: Reset/Set
    TA0CCR0 = 5000; // tempo de amostragem aproximadamente 2.5ms
(frequencia de amostragem)
    TA0CCR1 = 2500; // onda quadrada
}

//-----
//-----
void config_io(void) //configura entradas e saídas
{
    //PUSH_BOTTON
    P1DIR &=~BIT1+BIT2+BIT3; //toda a porta P1 como entrada
    P1REN |= BIT1+BIT2+BIT3; //habilita resistores de pull(up/down)
    P1OUT |= BIT1+BIT2; //pull UP
    P1OUT &=~BIT3; //pull down
    P1IES &=~BIT1+BIT2+BIT3; //a borda de descida que provoca
interrupção (descida = 1 - subida = 0)
    P1IES |= BIT1+BIT2;
    P1IFG &=~BIT1+BIT2+BIT3;
    P1IE |= BIT1+BIT2+BIT3; //habilita interrupção das entradas P1

    P2DIR |= BIT1+BIT4; //saida do timer
    P2SEL |= BIT1+BIT4; //saida dos pwms

    P1DIR &=~ (BIT0);///// ENTRADA ADC10

    //CONFIGURA O ADC10

```



```
////////canal 0 AD
ADC10CTL0 =SREF_0 + ADC10ON + ADC10IE + ADC10SHT1 ;
//vcc e vss como referencia, 16 ciclos de clock

ADC10CTL1 = SHS0 + INCH_0 + ADC10SSEL_2 + ADC10DIV_3 +
CONSEQ_0 ;

ADC10AE0 = BIT0; //entrada analogica habilitada

ADC10CTL0 |= ENC; //habilita conversao

}
```