

UNIVERSIDADE FEDERAL DE OURO PRETO

Mestrado em Ciência da Computação

**Especificação Sistêmica de Carro Robô Seguidor de Linha
com Sistema Avançado Utilizando Controle Proporcional**

Autor:

Rodolfo Labiapari Mansur Guimarães -
rodolfolabiapari@decom.ufop.br

Professor:

Ricardo Augusto Rabelo Oliveira - rrabelo@gmail.com

Ouro Preto - MG

9 de fevereiro de 2017

Sumário

I	Especificação de Projeto Embarcado	1
1	Introdução	1
2	Especificações	1
3	Referencial Teórico	2
3.1	Malhas em um Sistema de Controle	2
3.1.1	Malha Aberta	2
3.1.2	Malha Fechada	2
3.2	Sistema Controlador Proporcional Usando Equação Diferencial Lineal de Primeira Ordem	3
4	Elementos do Projeto	4
4.1	Visão Geral	4
4.2	Sensores do Robô \mathcal{A}	5
4.2.1	Sensor de Luz - Fototransistor	6
4.2.1.1	Tecnologia	6
4.2.1.2	Propósito no Projeto	6
4.2.2	Sensores de Movimento	7
4.2.2.1	Acelerômetro	7
4.2.2.2	Giroscópio	8
4.2.2.3	Resultado da Combinação desses Sensores	8
4.3	Atuadores	9
4.3.1	Tecnologia	9
4.3.2	Programação	10
4.3.2.1	Comportamentos Sobre Controle Simples	10
4.3.2.2	Comportamentos Sobre Controle Proporcional	10
4.4	Microcontrolador	11
4.5	Alimentação	12
4.6	Nuvem	13
5	Modelamento Matemático	13
5.1	Teoria da Equação Diferencial Linear	13
5.2	Definições	13
5.2.1	Teoria de Controle Proporcional (P)	14
5.3	Funcionamento Teórico-Procedural	14
6	Arquitetura de Comunicação	15
6.1	<i>Clouds</i> Disponíveis	15
6.1.1	Amazon <i>Web Services</i> (AWS)	15
6.1.2	Oracle Cloud Computer	15
6.1.3	Google Cloud Platform	16
6.2	Arquitetura	16
6.2.1	Descrição Geral de Operação	16
6.3	Protocolarização	16

6.3.1	Temporização	16
-------	------------------------	----

Parte I

Especificação de Projeto Embarcado

1 Introdução

Esta Parte I consiste em uma especificação de um trabalho de sistema embarcado complexo a ser construído em uma Parte II.

Atualmente é possível construir Carro Robô Seguidor de Linha utilizando poucos componentes, e com poucas linhas de programação, além da facilidade em encontrar em sites de venda de eletrônicos kits de sensores e atuadores de baixo custo prontos para serem acoplados à placa de prototipagem e assim, a construção de um carro seguidor.

O desenvolvimento do sistema controlador também segue o mesmo princípio de facilitação de uso. Utilizando uma plataforma de prototipagem tal como Arduino, é possível escrever o controle de um carro seguidor com poucas linhas de instruções. Isso pode ser levado como um desafio até para crianças e adolescentes com criatividade e entusiasmo para desenvolver um sistema completo e funcional como incentivo à robótica.

Na Seção 2 será exibido a especificação deste trabalho e na Seção 3 é feita toda a introdução teórica sobre o tema. Em Seção 4 é descrito com detalhes todos os elementos que serão utilizados para fazer este projeto ser concretizado, em 5 é feita uma exibição das definições matemáticas que o projeto levará em consideração para seu funcionamento e por fim, na Seção 6 é descrito como será a arquitetura escolhida para comunicação e processamento das informações.

2 Especificações

Abaixo é escrito a especificação segundo Professor Rabelo Oliveira rrabelo@gmail.com.

Neste trabalho, o robô será controlado por uma instância externa na Cloud, que irá efetuar a leitura dos dados transmitidos pelo robô via NodeMCU para retornar como feedback para os atuadores.

O caminho executado/aprendido pelo robô deverá ser usado para controlar um segundo robô que não possui sensores. Considere o ponto de partida similar para ambos.

A programação do segundo robô será enviada pela Cloud que contém os dados e algoritmo do primeiro robô.

O trabalho consistirá em duas entregas

Primeira Entrega - Especificação do projeto: deverá conter de maneira detalhada as seguintes características:

- a- Referencial teórico
- b- Proposta dos sensores do primeiro robô
- c- Proposta dos atuadores dos robôs, explicando como serão programados
- d- Alimentação do sistema, indicando como será montada a parte da alimentação dos motores e do NodeMCU
- e- Modelo matemático considerando os dados dos sensores discretos.

f- A SOLUÇÃO DA EQUAÇÃO DEVERA SER DISCRETA
SOMENTE DE PRIMEIRA ORDEM

g- Arquitetura para a execução do controle de primeira ordem nas nuvens e o envio dos dados pela internet, Indicar a solução de Cloud para uso (azure, amazon, watson, etc)

Data entrega:08/02/2017

3 Referencial Teórico

3.1 Malhas em um Sistema de Controle

3.1.1 Malha Aberta

A principal característica do sistema em malha aberta é a inexistência de realimentação.

Os valores assumidos pela variável de controle não dependem dos valores da variável de saída, sendo assim, a ação de controle é função apenas do processamento da variável de referência pelo controlador.

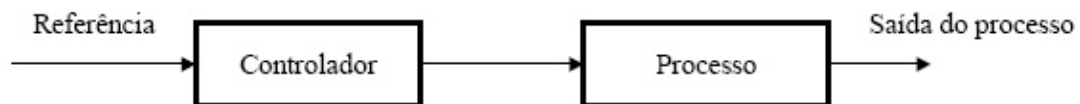


Figura 1: Sistema onde utiliza-se Malha Aberta. Sistemas que não possuem realimentação.

Dessa forma, o processamento de dados é puramente absoluto, sem observar dados de processamento anteriores tendo assim uma visão unicamente momentânea.

3.1.2 Malha Fechada

Já o sistema de controle em malha fechada, também é denotado de sistema de controle por realimentação.

A saída y é medida e comparada com a saída desejada, indicada através da referência r , para processamento através do controlador e a consequente definição da ação de controle u .

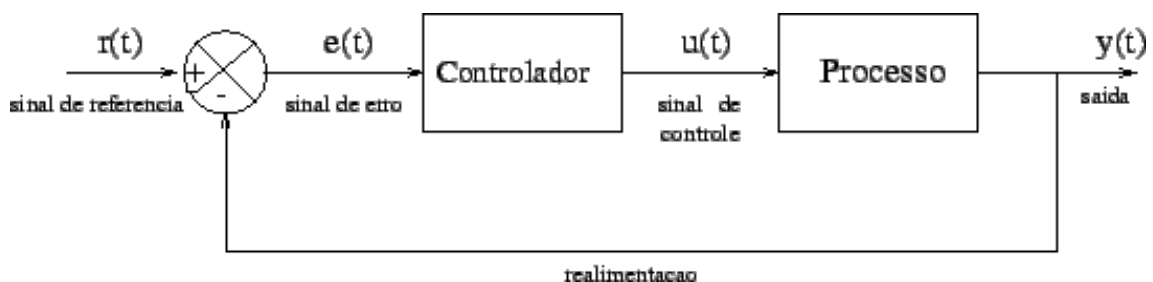


Figura 2: Sistema onde utiliza-se Malha Fechada. Sistemas onde existem um mecanismo de realimentação no controle de determinada ação.

Neste trabalho é utilizado Malha Fechada. O sistema de realimentação utiliza como entrada, dados de processamentos anteriores criando assim uma proporcionalidade mais autêntica á situação a ser trabalhada.

3.2 Sistema Controlador Proporcional Usando Equação Diferencial Lineal de Primeira Ordem

Sistemas de controle é um sistema que possui o propósito de gerenciar o comportamento de outros dispositivos. Pode-se dizer que é uma interconexão de componentes conectados de maneira a comandar, controlar ou ajustar a si mesmo ou outro sistema obtendo uma precisão maior em seus procedimentos.

A teoria de controladores proporcionais se baseiam em sistemas realimentados. Tais podem ser divididos em basicamente três partes sendo elas:

- Sistema a ser controlado;
- Controlador; e
- Realimentação.

O sistema a ser controlado é constituído por atuadores capazes de efetuar as ações necessárias. Os outros dois elementos têm como finalidade fazer com que o desempenho do sistema possua estabilidade e opere com certa precisão e agilidade, seguindo as especificações uma vez estabelecidas. Uma vez estabelecido como o sistema deverá ser desenvolvido, os controladores e sua realimentação serão item essencial para que o processo ocorra de forma estável. Tudo isso pode ser observado no diagrama exibido na Figura 3.

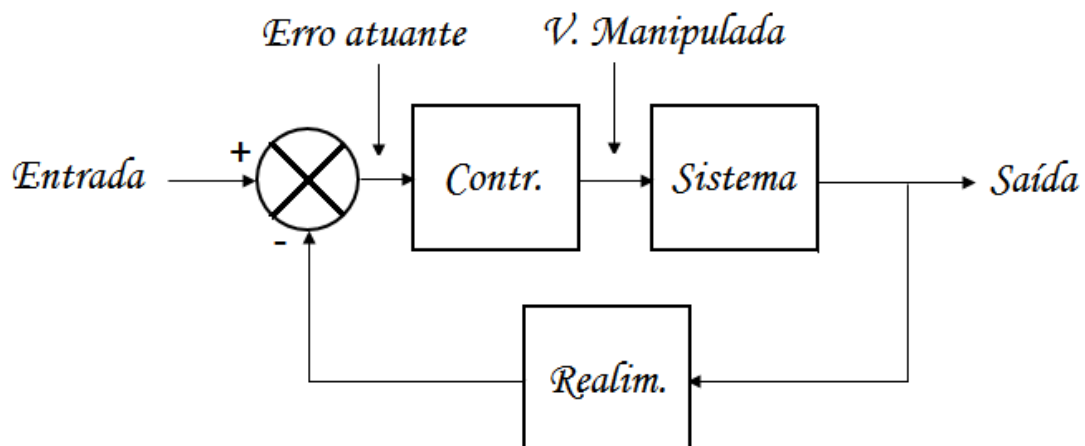


Figura 3: Estrutura de Sistema de Controle Geral Realimentado.

A realimentação é o ponto chave para a diferenciação de sistemas controladores comuns. Um sistema de controle realimentado compara, instantaneamente, o valor de saída anterior, com o valor de referência existente na entrada do sistema como os sensores. O resultado desta comparação é o centro de toda adaptabilidade estudada e é denominado erro atuante. Este é levado ao controlador, que produz

o chamado sinal de controle, cuja função resume-se em reduzir o desvio entre a saída e o sinal desejado. Como o nome sugere, em um controlador proporcional a saída do mesmo, também conhecido como sinal de controle (ou ação de controle), é diretamente proporcional ao sinal de erro, ou seja, ao erro atuante.

Sabendo-se em da proporcionalidade direta entre o sinal de controle e o sinal de erro, é possível afirmar que

$$a(t) \propto e(t) \quad (1)$$

onde o controle é diretamente proporcional à seu erro.

Entretanto, não é possível realizar operações matemáticas exatas com o sinal de proporção da fórmula. Par isso, deve-se admitir então uma constante de proporcionalidade entre as mesmas. Esta possui o nome de ganho proporcional e é representado pela variável f . Sendo assim

$$a(t) = f * e(t) \quad (2)$$

Dessa forma, o sistema matemático mostrado na Equação 2 poderá ser representado pelo diagrama exibido na Figura 4.

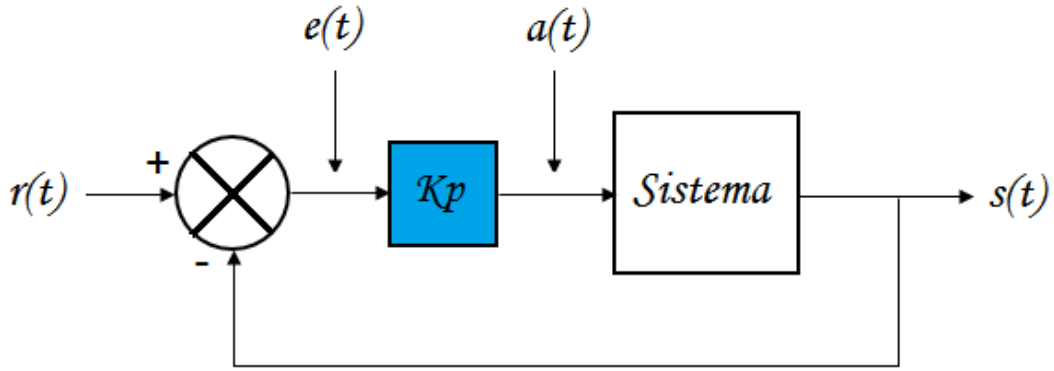


Figura 4: Sistema no Domínio do Tempo Geral.

Sendo assim, como fórmula final, temos

$$a(t) = f * (r(t) - s(t)) \quad (3)$$

onde $a(t)$ é a atuação, $e(t)$ representa o erro, $r(t)$ é o valor de início de execução de controle e $s(t)$ é saída do controle. Todos representando o valor no tempo t .

4 Elementos do Projeto

4.1 Visão Geral

Como já mencionado na Especificação (Seção 2), existirá dois robôes e uma computação em nuvem.

O trabalho consiste em replicar as ações de um Robô \mathcal{A} repleto de sensores em um Robô \mathcal{B} possuindo somente atuadores. Além disso, o processamento de dados e

tomada de decisões será totalmente feita em nuvem, deixando os carros unicamente como plataformas de leitura de sensores e atuadores.

Como é representado na Figura 5, o Robô \mathcal{A} terá sensores e atuadores. Seus dados não serão processados nele em si, mas sim em um computador externo ao sistema atuador. Ele fará o trajeto a ele imposto e deverá completá-lo. Após a execução do trajeto, a *cloud* terá dados suficientes, providos de vários sensores, para fazer o Robô \mathcal{B} realizar o mesmo trajeto sem a utilização de qualquer sensor.

Para que isso possa ser realizado, cada robô terá seus dispositivos e um componente Wi-Fi para envio e recebimento de dados sem fio à *access points* e assim conexão com a *cloud*.

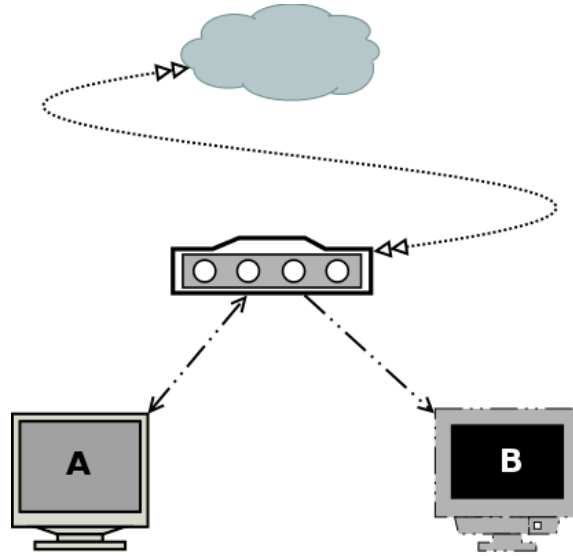


Figura 5: Diagrama Geral Abstrato do Sistema.

Deve-se atentar que os dois robôs possuem características e propósitos diferentes e que todo o processamento é realizado na *cloud* e não nos robôs sendo estes somente para captação de dados e execução de tarefa com seus atuadores.

Abaixo será descrito cada componente do sistema e algumas propriedades importantes.

4.2 Sensores do Robô \mathcal{A}

O projeto do Robô \mathcal{A} terá vários tipos de sensores em várias quantidades diferentes.

Utilizar somente o sensor fototransistor seria suficiente para o projeto de seguido de linha, mas não para este em si. Este sensor é unicamente para direcionar o carro para que ele não saia da linha completando o trajeto. Usando somente ele, é impossível que o carro saiba sua posição e seus movimentos.

Para contornar este problema, necessita-se de mais dois tipos de sensores: acelerômetro; e giroscópio. Eles, junto com o fototransistor, serão mencionados nas Seções 4.2.1 e 4.2.2.

4.2.1 Sensor de Luz - Fototransistor

4.2.1.1 Tecnologia

Para o carro, será utilizado o sensor fototransistor. Em sua superfície, existirá um LED que fará a iluminação da área no qual refletirá sobre a superfície e chegará até o sensor fototransistor. Como o seguidor de linha move sobre dois tipos de superfície (branca e preta), é possível identificar quando ele sairá da direção correta. Essa ideia melhor visualizada com a Figura 6.

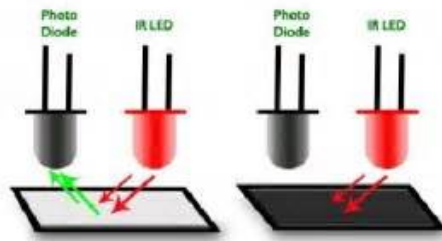


Figura 6: Funcionamento do sensor Fototransistor.

4.2.1.2 Propósito no Projeto

O projeto utilizará não somente dois sensores deste tipo mas uma série deles, posicionados logo à extremidade da faixa de direção como é exibido na Figura 7.

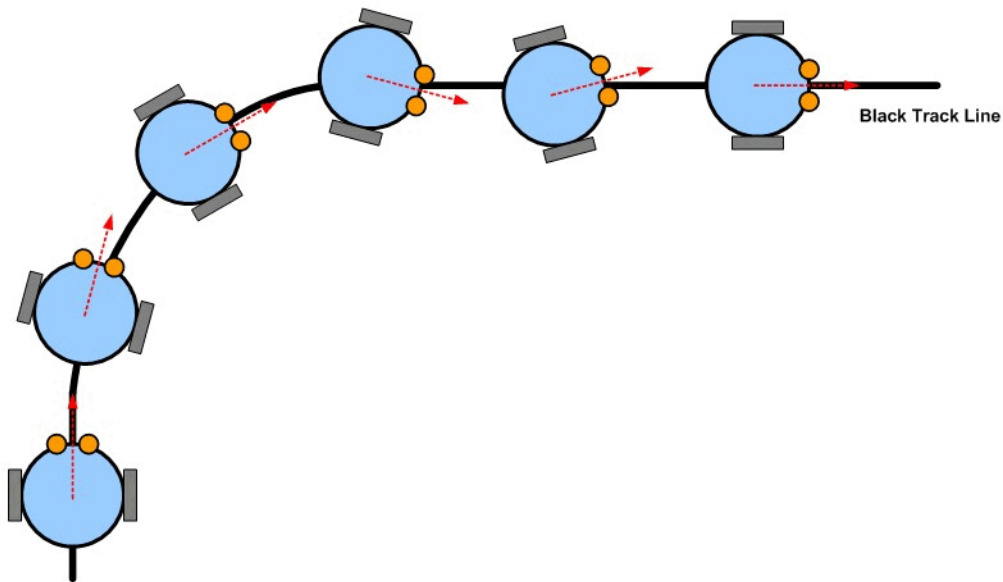


Figura 7: Posição dos Sensores. Exemplo utilizando dois sensores.

Utilizando uma série de sensores, cada sensor será responsável por avaliar se o carro saiu da linha original. Quanto mais à extremidade a faixa estiver em relação à série de sensores, maior será o erro dela e assim, a consequente uma correção mais rápida. Quanto maior a quantidade de arranjo de sensores melhor será a performance

do carro pelo motivo que mais sensores representarão melhor o nível do erro. Por este motivo, será utilizado seis sensores, posicionando três em cada extremidade da faixa. Eles serão dispostos o mais próximo possível de seu adjacente pois a resolução de operação será melhor do que distâncias grandes como três centímetros ou mais. Um exemplo com 8 sensores é exibido na Figura 8.

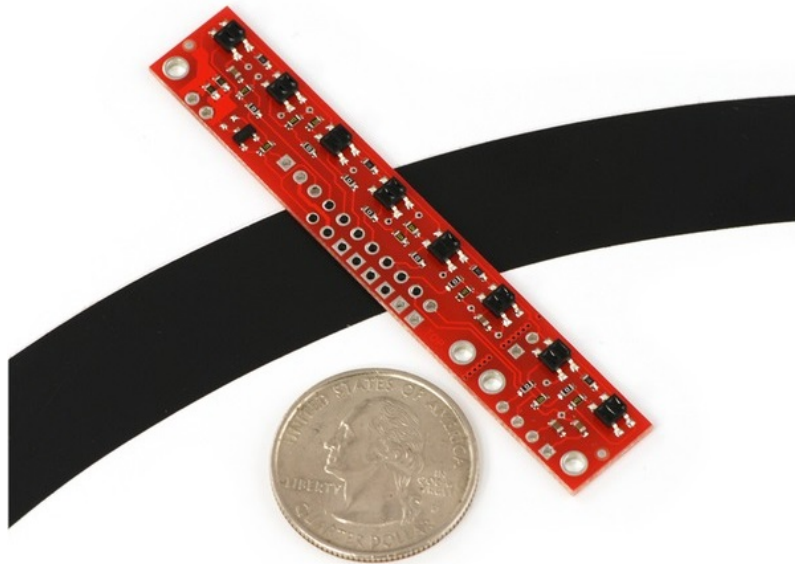


Figura 8: Sensores posicionados em uma *board* em série.

4.2.2 Sensores de Movimento

Como já descrito 4.2.1, sensores fototransistores não conseguem realizar gravações de que detectam movimento. Para suprir essa necessidade utilizar-se-á de dois outro tipos de sensores que detectarão movimento. O acelerômetro e o giroscópio.

4.2.2.1 Acelerômetro

Consegue capturar dados de duas à três dimensões (como mostra a Figura 9) podendo reconhecer qualquer tipo de movimento do ambiente onde está instalado.

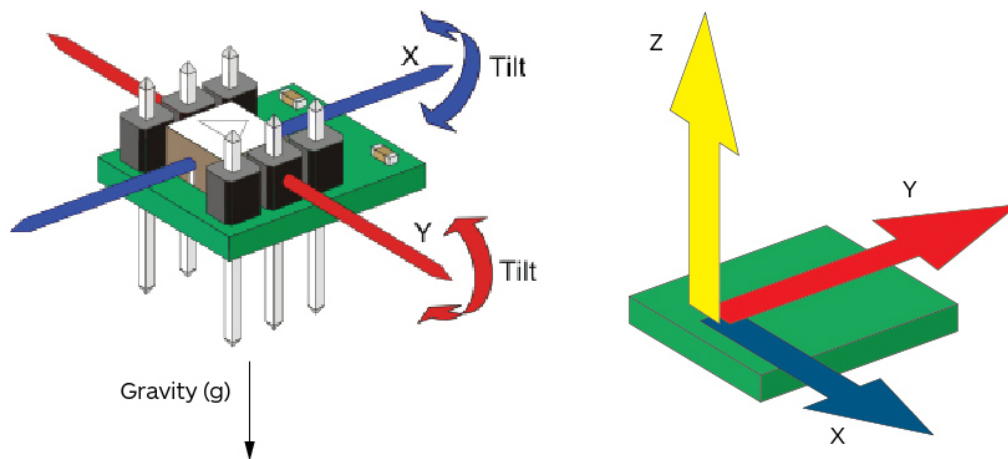


Figura 9: Visão técnica sobre o funcionamento do acelerômetro.

O sensor acelerômetro será utilizado para captar dados de deslocamento do carro como aceleração e distância percorrida.

Entretanto, velocidade e distância também não são suficientes para o projeto do carro como um todo, e por isso será mencionado o uso de giroscópio.

4.2.2.2 Giroscópio

O giroscópio é um sensor que consegue captar a angulação de determinado objeto à ele instalado, mesmo depois de realizar movimentos.

Assim, utilizando o aceleração e angulação, é possível registrar o trajeto que o carro realizou ao percorrer a trilha da faixa. Um exemplo da tecnologia utilizada hoje nos dispositivos embarcados é exibido na Figura 10.

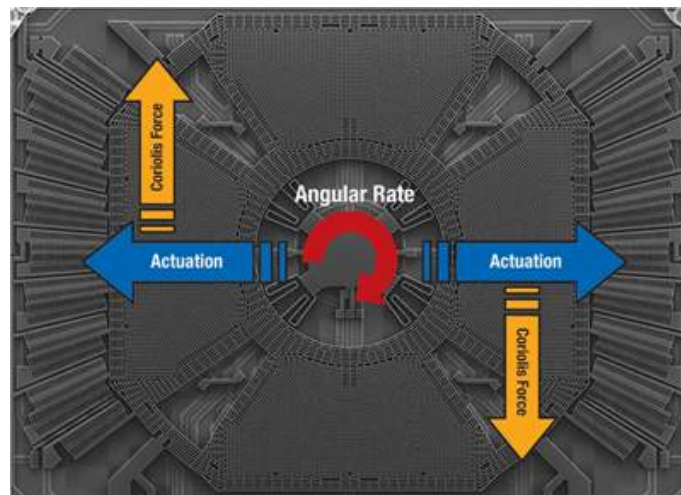


Figura 10: Exemplo de giroscópio situado em telefones digitais modernos.

4.2.2.3 Resultado da Combinação desses Sensores

Assim, o sensor fototransistor realizará a leitura do trajeto a ser feito, o acelerômetro receberá a velocidade atual do carro e o giroscópio registrará a angulação do carro.

Tudo isso processado permite a reprodução exata no Robô \mathcal{B} , mesmo sem utilizar nenhum sensor.

4.3 Atuadores

O sistema possuirá duas unidades de motores atuadores que serão especificados à seguir.

4.3.1 Tecnologia

Os motores serão do tipo de Corrente-Contínua, propriedade física que *não* permite o cálculo exato de distância e velocidade do equipamento sem a utilização de sensores de marcação de passo/angulação. Para isso, será utilizado vários sensores para captação de dados do ambiente, onde seus dados serão enviados para a *cloud*, processados e recebidos novamente para assim realizar a atuação no carro. No caso do Carro \mathcal{B} , não haverá processamento, mas sim a atuação, sendo este somente um receptor escravo.

Serão acoplados à *board* utilizando um *shield* intermediário entre o microcontrolador e os motores. Tal *shield* realizará o processo de interface de controle e energização dos motores criando assim um sistema estável. Seu nome é L293DD e possui suporte total para interface de pinos NodeMCU. Seu sistema de operação utiliza Ponte-H dupla e com isso é possível controlar até dois motores além de conectores para seleção de interface serial UART, SPI, e entrada analógica, além de conexões para habilitar as opções de *Enable* e *Reset* do microcontrolador. Ele pode ser visto por meio da Figura 11.

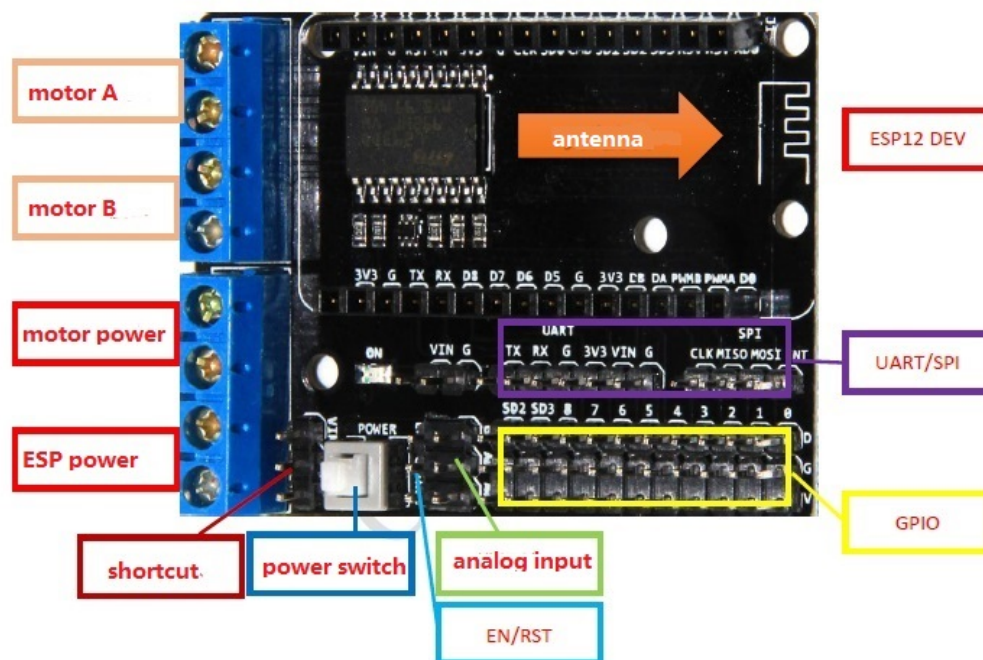


Figura 11: Motor *Shield* a ser Utilizado.

4.3.2 Programação

O carro andará numa velocidade máxima fixa em um determinado valor suportado pelos motores. Quando algum de seus sensores, após processados, perceber que o carro saiu da linha, o procedimento de correção de direção será acionada para avaliar os dados dos sensores e assim verificar os passos a serem realizados nos atuadores para que o carro volte a operar normalmente na direção correta em cima da faixa.

Existem dois tipos de comportamentos em um seguidor de linha comumente conhecido. Eles serão descritos e explicados a seguir.

4.3.2.1 Comportamentos Sobre Controle Simples

Com controle simples, sem realimentação, é possível perceber que seu erro após à primeiro desvio será grande e sua forma de conserto fará com que ele aumente ainda mais sua margem de erro. isso causará grande distúrbio em seu movimento criando assim um padrão geométrico construído por uma sequência de segmentos lineares alternados quanto à direção, formando linhas quebradas com alternância de ângulos salientes e reentrantes como mostrado na Figura 12.

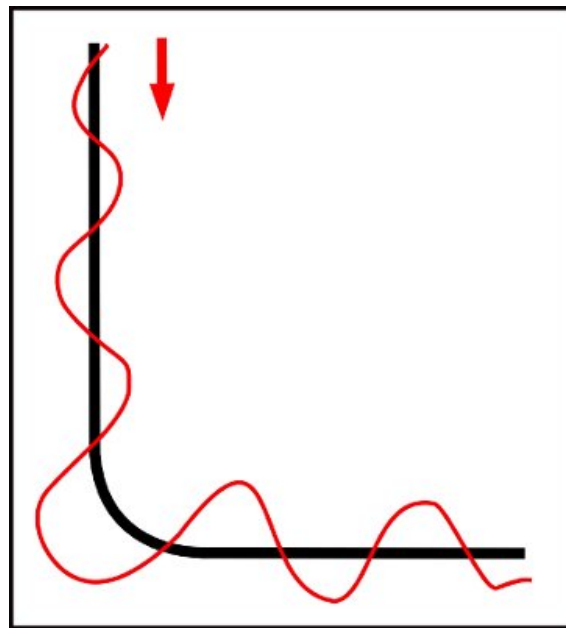


Figura 12: Trajeto esperado de um seguidor de linha comum.

Esse movimento oscilatório faz com que o carro gaste tempo em completar sua tarefa de percorrer a linha e também energia, item essencial para sistemas embarcados.

4.3.2.2 Comportamentos Sobre Controle Proporcional

Utilizando o controle proposto é possível que este padrão seja decrementado ao ponto de tornar o movimento do carro mais suave e portanto seu trajeto mais conciso quanto à linha do trajeto. O resultado esperado é exibido na Figura 13.

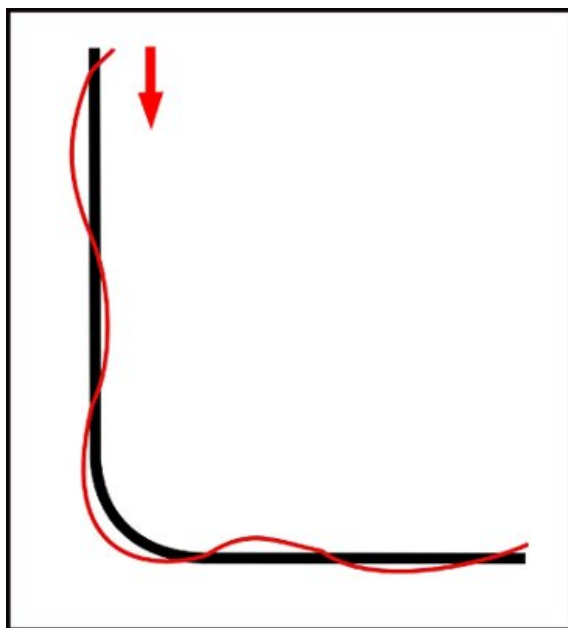


Figura 13: Trajeto esperado de um seguidor de linha utilizando Controle Proporcional.

Um fato importante a se atentar é que, para que o carro funcione como esperado, o algoritmo, seus parâmetros e os parâmetros da fórmula de controle proporcional devem estar devidamente calibrados para que o carro obtenha a melhor performance possível.

Este trajeto faz com que seus movimentos sejam mais suaves, estáveis, ágeis e eficientes em relação ao mostrado na Figura 12, exibida anteriormente.

4.4 Microcontrolador

Microcontrolador utilizado para este trabalho é o NodeMCU. É uma plataforma IoT¹ *open-source*.

Como esperado de um microcontrolador para IoT o sistema possui integrado um componente de comunicação Wi-Fi para troca de dados. Utiliza linguagem de *script* Lua desenvolvida por brasileiros.

Suas especificações são uma CPU ESP8266 possuindo 128 KB de memória, 4 MB de armazenamento e suporta o sistema operacional chamado XTOS. Permite comunicação pelo Wi-Fi e USB onde também é energizada. Possui um total de 10 pinos de entrada e saída de propósito geral (GPIO) onde suportam funções como PWM, comunicação I²C e 1-wire. Além da antena Wi-Fi, possui também um conversor USB-TTL para comunicação serial.

Seus microprocessadores podem ser facilmente ligados a um computador e também utilizado a IDE Arduino. A Figura 14 exibe um esquemático de seu protótipo.

¹*Internet of Things*, ou seja, internet das coisas.

todos os sensores nele contido.

4.6 Nuvem

A nuvem será utilizada para processamento, retirando todo o trabalho dos robôs.

Utilizar este tipo de sistema traz várias vantagens com redução de gasto energia nos robôs e centralização de processamento e *backup* de informações em local seguro além da transparência do sistema de controle.

Entretanto, latências altas em comunicações ou pacotes de informações corrompidas podem tornar o sistema inconsistente ou até mesmo comprometido.

A arquitetura do sistema será melhor detalhada em Arquitetura de Comunicação, Seção 6.

5 Modelamento Matemático

5.1 Teoria da Equação Diferencial Linear

Equações Diferenciais Lineares são equações que suas soluções podem ser somadas a fim de produzir uma nova solução.

É considerada linear quando satisfaz as características de que cada coeficiente a_n e o termo de não-homogeneidade só dependem da variável independente, no caso x ; e a variável dependente, no caso y , e suas derivadas são de primeiro grau. Assim, ela deve representar a seguinte forma

$$f(x) = a_n(x) \frac{d^n y}{dx^n} + a_{n-1}(x) \frac{d^{n-1} y}{dx^{n-1}} + \cdots + a_1(x) \frac{dy}{dx} + a_0(x)y \quad (4)$$

5.2 Definições

Para iniciar a discussão sobre sistema de controle proporcional, primeiramente será definido os termos utilizados.

Alvo: É a posição que desejamos que o carro esteja em relação à faixa, ou seja, centralizado respeitando a devida linha.

Erro: Diferença entre a posição atual e o *Alvo*. Pode ser qualquer valor no conjunto dos \mathbb{R} .

Proporção: Fator que determinará quão distante o carro está da linha. Por exemplo são as proporções: ‘*para a esquerda*’, ‘*para a direita*’, ‘*para a extrema esquerda*’, ‘*pouco para a direita*’, etc. É representado pela constante de variação K_p .

O procedimento de execução é baseado no seguinte pseudocódigo:

1. *Realiza o cálculo inicial da posição atual.*
2. *Calcula o erro baseado na posição atual.*
3. *Ele então comandará os motores para fazer uma girada:*

- Brusca caso o erro for grande; ou
- Leve caso o erro for pequeno.

Assim, basicamente, a magnitude do giro tomado é proporcional ao erro.

4. Repete os passos até completar o objetivo inicial.

Com esses passos, saímos de um controle simples para um Contro Proporcional mais eficaz.

São amplamente utilizados em situações que se baseiam em controladores eletrônicos chamados “*single-loop*”.

Abaixo é descrito a formulação matemática do conceito de controlador proporcional

5.2.1 Teoria de Controle Proporcional (P)

Produz um sinal de saída que é proporcional à parâmetro do erro

$$P_{\text{saida}} = K_p e(t) \quad (5)$$

onde:

$e(t)$: Erro no tempo t .

K_p : Constante relativa à proporção.

Esse método possui a propriedade de eliminar as oscilações do sinal de saída. Para tal, o sistema permanece sempre ligado e o sinal de saída é diferente de zero.

5.3 Funcionamento Teórico-Procedural

Aqui será descrito uma representação de como será a implementação de toda a teoria descrita neste trabalho. Assim, de forma geral, teremos o seguinte algoritmo implementado para tratamento dos valores obtidos dos sensores e ajuste da rotações dos motores.

Algorithm 1 Controle Proporcional de Correção de Angulação

```

1: procedure PROPORTIONAL_CONTROL( $position, set\_point, \Delta, last\_ \Delta, \Phi$ )
2:    $\Delta \leftarrow position - set\_point$ 
3:    $\Phi \leftarrow \Delta * K_p$ 

```

onde:

$position$: Posição atual do carro.

set_point : Ponto central onde encontra-se a faixa.

Δ : Representa o erro proporcional da posição atual contra o ponto central da faixa.

$last_ \Delta$: Representa o erro proporcional da posição da iteração passada.

Φ : Correção a ser realizada nesta iteração.

Como é de se esperar, a correção proporcional Φ será aplicada à velocidade motor para conserto de posição do carro.

6 Arquitetura de Comunicação

6.1 *Clouds* Disponíveis

Nesta seção, será relatado onze serviços de *cloud computing* que podem ser acessados gratuitamente com recursos limitados para teste e avaliação.

- Microsoft Azure;
- Amazon Web Services (AWS);
- IBM Bluemix;
- BMC Software;
- Computer Associates;
- Cisco Systems;
- Hewlett-Packard Cloud Computer;
- Intel Cloud Computer;
- Doit Cloud;
- Oracle Cloud Computer; e
- Google Cloud Plataform.

Abaixo é descrito algumas informações e propriedades de alguns serviços prestados.

6.1.1 Amazon *Web Services* (AWS)

O serviço fornecido pela Amazon possui várias vantagens e facilidades de uso. Um adentro ao tema, o Amazon *Web Services* (AWS) é um subconjunto de operações de computação na nuvem da *amazon.com*.

Ela oferece recursos *on-demand* onde opera em mais de 16 regiões geográficas pelo mundo e uma delas encontra-se em São Paulo, no Brasil. Neste pacote, também é incluso o Amazon *Elastic Compute Cloud* e também o Amazon *Simple Storage Service* como ferramentas adicionais para uso.

Ao final de 2016, AWS já possuía mais de 70 serviços na nuvem incluindo serviços próprios para IoT. Entretanto, o sistema deve executar sistemas operacionais ou linguagens de programação *Node.js*, *Java* ou *Python* sendo esses requisitos não adaptáveis ao dispositivos obtidos no momento.

6.1.2 Oracle Cloud Computer

Necessita compra pra uso do sistema de controle IoT dos serviços prestados. Outra desvantagem de uso acadêmico é que seus serviços são cobrados assim que o prazo de teste acaba sem solicitação prévia.

6.1.3 Google Cloud Platform

Empresas como Coca-Cola, Spotify, Philips, utilizam este serviço para processamento massivo de dados e terceirização de serviços de infraestrutura de projeto como redundância, técnicas de *failover*, backups, monitoramento e armazenamento de grandes volumes de dados.

6.2 Arquitetura

6.2.1 Descrição Geral de Operação

Será estabelecida uma comunicação TCP entre todos os dispositivos, propriedade que utiliza *hand-shake* e verificação de perda de pacotes.

Uma vez a comunicação esteja estabelecida, a *cloud* receberá um comando de início de envio de dados, os dados propriamente ditos e a finalização do envio. Novamente, ao utilizar o protocolo TCP, não é necessário realizar verificação de recebimento e também de *ack*. Como o Robô \mathcal{A} não realiza processamento, os dados recebidos pela *cloud* serão processados para retorno ao Robô \mathcal{A} e assim sua atuação.

Além de processar na *cloud*, as informações coletadas e processadas serão utilizadas para conduzir os atuadores do no Robô \mathcal{B} , repetindo os passos realizados pelo primeiro, sem a utilização de qualquer sensor para auxiliar na obtenção de seu propósito. Deve-se lembrar que toda a explicação descrita no Referencial Teórico (Seção 3) e a fundamentação matemática descrita neste trabalho na Seção 5 serão de fato implementadas na *cloud*.

Dessa forma, a *cloud* terá dois papéis principais neste sistema como um todo, sendo eles:

1. Realizar *todo* o processamento do *Line Tracker* do Robô \mathcal{A} ;
2. Armazenar, computar e enviar os dados obtidos do Robô \mathcal{A} para o Robô \mathcal{B} a fim de que ele consiga executar a mesma tarefa de \mathcal{A} , inúmeras vezes se necessário, sem utilização de qualquer equipamento auxiliar.

6.3 Protocolarização

6.3.1 Temporização

Será definido dentro do Robô \mathcal{A} um intervalo fixo t para captação e envio de dados à *cloud*.

Criando um padrão igual este, é possível reproduzi-lo no Robô \mathcal{B} , com uma precisão maior de reprodução de movimentos capturados pelo Robô \mathcal{A} . Ao final, poderá ser enviado ao Robô \mathcal{B} os dados nos devidos tempos corretos de execução, ou mesmo um vetor com todos os dados nos devidos tempos corretos já preparados para a execução.