

UNIVERSIDADE BRAZ CUBAS
CURSO DE ESPECIALIZAÇÃO EM
DESENVOLVIMENTO ORIENTADO A OBJETOS JAVA

APLICAÇÃO DE CONTROLES DE TRÊS TERMOS (PID)
PARA ROBÓTICA MÓVEL

Ricardo Rodrigues dos Santos

Mogi das Cruzes, SP

2017

FOLHA DE APROVAÇÃO

Ricardo Rodrigues dos Santos

Aplicação de Controles de Três Termos (PID) para Robótica Móvel

Artigo apresentado ao Curso de Especialização em Desenvolvimento Orientado a Objetos em Java da Universidade Brás Cubas, como requisito parcial à obtenção do título de Especialista em Desenvolvimento Orientado a Objetos em Java.

Prof. Me. André Luiz Osvaldo Herzer

Prof. Me. André Luiz Osvaldo Herzer

Mogi das Cruzes, 27 de Dezembro de 2017

RESUMO

O trabalho desenvolvido apresenta o estudo de desenvolvimento de um software para robôs de competição seguidores de linha para a plataforma Arduino, utilizando técnicas de Controladores de Três Termos (PID - Proporcional Integrador Derivativo). O desenvolvimento do trabalho visa aprimorar os métodos e agilizar o processo de detecção das variáveis dos controladores PID. Nesta experimentação, foi utilizado uma estrutura robótica com sensores de refletância que identificam a cor preta do circuito a ser percorrido pelo robô. A programação usou como base a linguagem Java da qual desenvolveu-se um transmissor e receptor de sinal dos termos de controle em tempo real. Foi utilizado o ambiente Eclipse IDE com plugins SWT (Standard Widget Toolkit) que permitiram o desenvolvimento ágil do software. Foram executados ensaios que comprovam a efetiva utilização das variáveis de controle PID e seus comportamentos gráficos e benéficos diante do percurso percorrido pela estrutura robótica. Os resultados obtidos comprovam a utilização positiva no processo de teoria dos Controles de Três Termos (PID).

Palavras-chave: controle de três termos; pid; robótica; arduino; java

ABSTRACT

This work presents the study of development of a software for robots competition, followers line to the Platform Arduino, using techniques Controllers Three Terms (PID - Proporcional Integrador Derivativo). The development of work aims to improve the methods and streamline the process of detection of the variables of PID controllers. In this experiment, we used a structure robotics with sensors reflectance that identify the black circuit to be covered by the robot. Programming used as the basis the Java language which developed if a transmitter and receiver sign of the terms control in real time. Was used the environment Eclipse IDE with plugins SWT (Standard Widget Toolkit), that allowed the development agile software. Have been executed tests that prove the effective use of control variables PID and their behaviors graphics and beneficial in front of the route covered by the structure robotics. The results prove the use positive in the process of Theory of Controls Three Terms (PID).

Keywords: controllers three terms; pid; robotic; arduino; java

1 INTRODUÇÃO

1.1 Estudo dos Aspectos da Robótica Móvel

As invenções produzidas pela humanidade foi um dos aspectos evolucionistas de maior importância na concepção humana, trazendo consigo um grande impacto social na temática mundial. A simplificação do trabalho humano braçal sempre foi uma busca constante com a criação de modelos, das quais permitia que a humanidade pudesse ter um maior desenvolvimento da arte, do intelecto, do lazer ou mesmo em novas formas de entretenimento. De fato, desde que a roda ganhou forma e função no cenário do homem, entre 3.500 e 3.200 a.C., a otimização do tempo, a partir do controle de processos, vem ganhando mais espaço (SILVEIRA & SANTOS, 1998). Pio, Castro e Júnior (2006) defendem o conceito de que um robô móvel é uma estrutura autônoma capaz de extrair informações do ambiente e utilizar o conhecimento dessas de maneira segura, sendo versáteis inclusive nos aspectos industriais. Suas atuações implicam em uma transposição das barreiras geográficas, movimentando-se a partir de pernas, rodas, esteiras por empuxo ou flutuação.

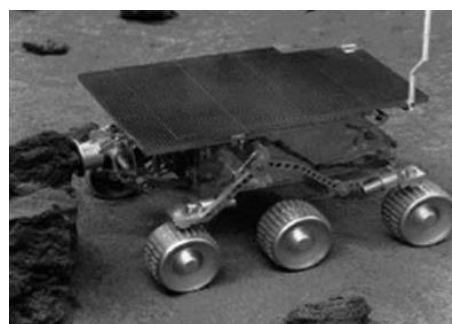
Com os avanços tecnológicos durante os tempos, os robôs móveis ganham mais espaço na sociedade. Suas aplicações permitem entender o seu vasto funcionamento, observado principalmente nos centros de pesquisa, como por exemplo, Sojourner, primeiro robô desenvolvido pela Nasa para explorar Marte; ou até mesmo Carnegie Mellon, utilizado para explorar minas abandonadas.

Figura 1 - Aplicações de robótica móvel



Fonte: JUNIOR & SILVA (2009)

Robô Groundhog desenvolvido na Universidade
Carnegie Mellon



Fonte: JUNIOR & SILVA (2009)

Primeiro robô desenvolvido pela Nasa denominado
Sojourner

Robô, cuja terminologia deriva do inglês Robot, possui uma origem tcheca, significando “servidão ao trabalhador forçado”. Mesmo tendo uma definição impactante, “a robótica se tornou uma das áreas mais fascinantes” (SILVEIRA & SANTOS, 1998, p. 15); atraentes

principalmente no âmbito educacional com estudos que relatam emoções positivas ao desenvolver suas metodologias com estímulo ao uso de plataformas computacionais.

A inserção da robótica nas escolas, desde o ensino fundamental, se tornou nos últimos anos uma ferramenta de auxílio importante no ensino [...] despertando a curiosidade e a vontade de aprender, especialmente quando veem a possibilidade de aplicar conceitos abstratos da física, matemática, entre outras ciências em seus projetos e protótipos. (LIMA, 2017, p. 15)

O livro “Eu Robô” publicado pelo escritor americano Isaac Asimov, em 1950, aplicou o termo robótica a partir de coletâneas de nove livros de ficção ao longo anterior a sua compilação. Na obra, Asimov descreve os robôs como simples ferramentas que executavam trabalhos domésticos, até a evolução no sistema de máquinas que determinam os rumos de um sistema produtivo.

A robótica é um dos princípios que cria uma ligação entre a percepção e a ação. O estudo, projeção e implementação de sistemas ou dispositivos são bastante rudimentares e determinantes no formato das tarefas entre o meio físico e sua interação com os sistemas industriais (BRITO, et al., 2014).

1.2 Aspectos de Competições, Torneios e Aprendizagem na Robótica Móvel

Os torneios de robótica são uma realidade na vida de adeptos às eventuais disputas educacionais e a ganhos de experiência e conhecimento. Dessa forma, é importante salientar os aspectos mais relevantes a fim de introduzir os principais conceitos e vertentes de estudos existentes sobre a filosofia de torneios de robótica. Os desafios são uma forma de instigar o comportamento dos entusiastas a novas descobertas e criar uma forma de visão aprimorada dos conceitos de eletrônica, programação e mecânica, conforme afirma Ferreira (2017).

O reconhecimento mundial destas características são vistas a partir de torneios como FIRST® LEGO® League (FLL) e World Robot Olympiad™ (WRO). Este tipo de valorização conforme Afari e Khine (2017), encoraja os aspectos de desafio dos estudantes no desenvolvimento de situações problemas, tornando-os inovadores, criativos e prontos para solucionar situações-problemas.

A forma de incentivo para o ganhos nos aspectos desse campo apontado por Lima (2015), a partir da concepção dos torneios, são trabalhados por kits do LEGO® Mindstorms® existentes desde 2006 no mercado, que podem ser utilizados no desenvolvimento da aprendizagem dos conceitos envolvendo Engenharia, Física e de outras áreas de estudo.

Figura 2 - Versões dos Kit LEGO Mindstorms NXT



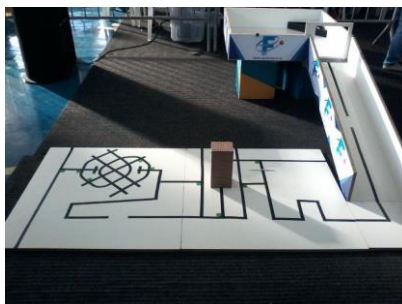
Fonte: BRITO, et al. (2014)

No Brasil, existe uma modalidade de torneio bastante explorada denominada OBR (Olimpíada Brasileira de Robótica). A OBR teve início em 2003 doravante de um torneio denominado CBR – Competição Brasileira de Robótica, englobando a participação apenas de estudantes universitários. A partir de esforços de professores, a OBR passou a compor o quadro dos estudantes do ensino fundamental e médio. Em 2006, o projeto para a OBR foi enviado ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), sendo portando iniciado pelo Prof. Luis Marcos Gonçalves da UFRN, tendo início a partir das Olimpíadas do Conhecimento.

As regras principais que regem o funcionamento da OBR estão disponível no próprio site do evento. Um resumo bastante prático da logística do funcionamento da Modalidade Prática da OBR é citado por Gonçalves, Jubilado Jr e Paiva (2016). No descritivo da obra o robô tem por objetivo completar o desafio da arena para resgate de vítima; onde os mesmos devem seguir a linha preta sem falha de percurso. A arena, por sua vez, possui obstrução que são denominadas por GAP (interrupções de linha), redutores de velocidades (compostos por lápis envoltos por um material branco), obstáculos (podendo ter tamanhos pequenos e grandes) e cruzamentos, havendo a possibilidade de ter ou não quadrados verdes de 20mm x 20mm, que determinam o sentido da linha preta.

O percurso ao qual o robô realizará o resgate da vítima, é composto também de uma rampa com uma sala especial. As vítimas são compostas por bolas de isopor recoberta por alumínio que devem ser inseridos na área de resgate em uma base triangular.

Figura 3 – Percurso de Arena OBR



Fonte: GONÇALVES, et al. (2016)

2 MÉTODOS

1.3 Controladores de três termos (PID)

O controle PID (Proporcional Integrador Derivativo), é um controle de ciclo de malha fechada. Também conhecido por Controle de três tempos, é um sistema bastante utilizado por mais da metade dos sistemas de controle industriais em uso; atualmente, empregam esquemas de controle PID ou PID modificado. (OGATA, 2003). Como o próprio nome sugere, o Algoritmo PID consiste basicamente nos seguintes coeficientes: proporcional, integral e derivativo; com uma variável que permite obter uma otimização em tempo de resposta (DORF & BISHOP, 1998, p. 309). Este controlador tem por função a equação de transferência:

$$C(s) = K_p + \frac{K_i}{s} + sK_D$$

Dessa forma, temos a seguinte equação integro-diferencial:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

A real definição do termo derivativo possui uma definição:

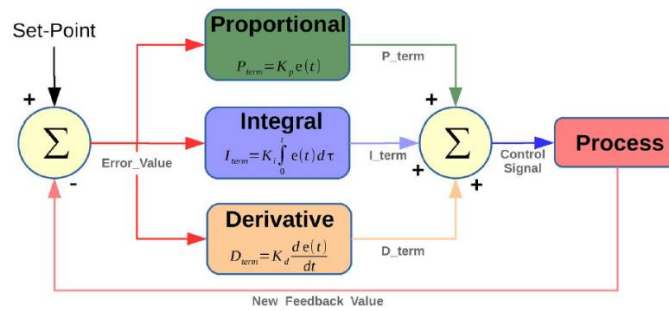
$$G_d(s) = \frac{K_D s}{\tau_d s + 1}$$

No entanto, o termo de τ_D , possui um limite de constante menor que o próprio processo, tornando-o nesses casos desprezível.

A ideia da utilização do PID se baseia na manipulação do erro e seu retrabalho. Existem várias formas de implementação do sistema PID. Entende-se que SP significa Setpoint, sendo o valor que o processo deve atingir. Já a variável PV significa Variável do Processo; representa o valor do processo de saída. Offset significa o deslocamento em regime permanente. MV representa a Variável Manipulada, sendo o elemento no qual o controlador atua sob o processo. O erro (Error) é a diferença da Variável do Processo (PV) e o Setpoint (SP), referência utilizada como base nos processos do erro.

$$Error = PV - SP$$

A proporção corrige instâncias de erro, a integral corrige o acúmulo de erro, e a derivada corrige o erro atual contra o erro na última vez que foi verificado. O efeito da derivada é contrariar o excesso causado por P e I. Quando o erro é grande, o P e o I empurra a saída do controlador. Esta resposta do controlador torna a mudança de erro rapidamente, o que, por sua vez, causa o controle derivativo para contrariar de forma mais agressiva o P e o I.

Figura 4 - Estrutura de um compensador PID

Fonte: MATHWORKS (2017)

Existem 3 modos de combinações que o PID pode se comportar. Não é necessário utilizar todas as combinações de processos. Os ajustes vão depender da maneira como pode ser e de qual forma podem ser trabalhados.

- P – Pouca utilização
- PI – Usado frequentemente
- PID – Pouca utilização
- PD – Raramente utilizado (foco mais em servo motores)

Em termos de expressão:

Se considerar $K_D = 0$, tem-se um controlador proporcional e integral (PI),

$$G_C(s) = K_p + \frac{K_I}{s}$$

Se considerar $K_I = 0$, resulta-se em um controlador proporcional derivativo (PD)

$$G_C(s) = K_p + K_D s$$

1.4 Sintonia de Controles PID

O ajuste de um controle PID depende relativamente das variáveis de ação do objetos externos. Para se obter os parâmetros necessários precisa-se que seus principais controles (ganho, proporcional, integral, taxa de derivação); estejam sintonizados para uma obtenção de resposta da variável de processo. Os parâmetros de sintonia de controlador PID devem estar bem segmentados para a estabilidade de um sistema. Assim devem ser respeitados os seguintes parâmetros:

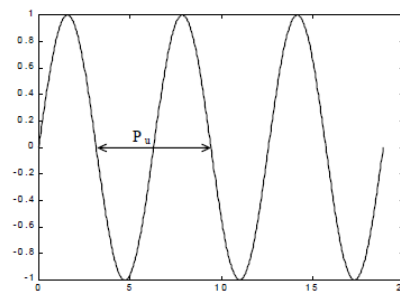
- O K_P Muito alto levará à oscilação em valores e tenderá a gerar um deslocamento
- O K_I irá contrair o deslocamento de um sistema. O valor em uma proporção muito alta, implica que o Setpoint (variável de referência) alcançará a variável de processo (PV) rapidamente.

- Isso dentro da parametrização de um sistema é inviável, visto que o comportamento brusco pode acarretar oscilações no sistema. Portanto a variável de processo é propensa a ser instável.
- K_D mantém as parametrizações de K_I sob controle, impedindo que as oscilações dentro do sistema sejam bruscas.

1.5 Sintonia de Controles PID de Ziegler-Nichols

As sintonias de controle PID recaem implicitamente em KP, KI e KD. Conforme Lourenço (1996), os métodos de controle de sintonia PID, baseando-se nos modelos de Ziegler-Nichols, é realizado a partir do método de malha-fechada, até se obter oscilações na amplitude constante. Os parâmetros de sintonia necessário para os ajustes são dois tipos de variáveis: Ganho (G_u : ganho mínimo do processo o tornando criticamente estável), e o período de oscilação (P_u).

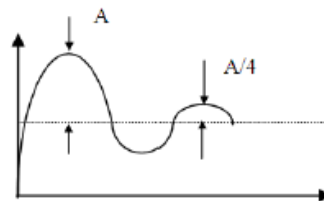
Figura 5 - Método Sensibilidade e Limite



Fonte: SARAIVA (2011)

De maneira geral, as regras de sintonia no modelo de Ziegler-Nichols obtém uma razão de $1/4$ sobre a resposta em um circuito de malha-fechada, representando a taxa de decaimento.

Figura 6 - Representação do controle PID sobre $1/4$ da malha



Fonte: SARAIVA (2011)

Obtendo esses valores, é possível calcular os parâmetro de acordo com a tabela abaixo.

Tabela 1 - Tabela de Parâmetros de Cálculo de PID

Controlador	Fórmulas		
P	$K_p = 0.5 G_u$		
PI	$K_p = 0.45 G_u$	$T_i = P_u/1.2$	
PID	$K_p = 0.6 G_u$	$T_i = P_u/2.0$	$T_d = P_u/8$

Fonte: SARAIVA (2011)

1.6 Sintonia de Controles PID pelo Método Tentativa e Erro

Os métodos descritos até o momento representam a maneira quando se tem a resposta de um sistema em degrau de malha aberta. No entanto quando os métodos não são sustentáveis, obtém-se uma forma simplificada para localização dos ganhos. Brito, Maladosso e Guibes, (2014), descrevem os métodos a partir dos seguintes parâmetros:

- Os valores de K_d e K_i são zerados, enquanto que K_p é sintonizado até o momento em que o sistema encontra a resposta desejada;
- Após obter K_p , K_i é utilizado para diminuir perturbações, aumentando-se o ganho até encontrar os valores que oscilam de maneira positiva no sistema. Esse valor não deve ser alto, pois pode causar instabilidade;
- Com K_p e K_i definidos, varia-se K_d até o momento em que o mesmo responda de maneira satisfatória. No entanto, deve-se considerar certo cuidado ao manipulá-la, pois pode gerar também instabilidades.

1.7 Princípios da plataforma Arduino

O desenvolvimento de softwares que oferecem o controle de sistemas embargados de microcontroladores dá-se pela versatilidade estando presente em praticamente todos os ambientes. No atual cenário mundial, sempre há um dispositivo controlado por eletricidade e um sistema embargado coordenando seu funcionamento.

O Arduino, é uma plataforma gerenciada por um microcontrolador capaz de possuir uma facilidade na construção de sistemas de computação física. A computação física é um conjunto de ferramentas que permitem atuar no mundo convencional, permitindo uma conversa entre o mundo físico e o mundo virtual (SULLIVAN & IGOE, 2004). Os sistemas de comunicação na plataforma Arduino, destaca-se por enviar conjunto de instruções para o microcontrolador, permitindo manipular entradas de processamento (ler um sensor, uma mensagem, ou pressionar um botão), e transformar em uma saída de dados, permitindo manipular um motor ou simplesmente ascender um LED.

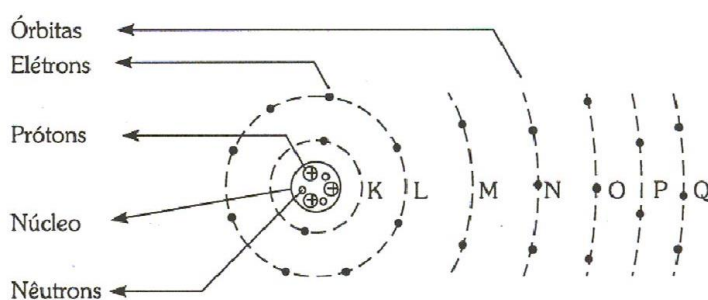
Segundo McRoberts (2011), a maior vantagem de utilizar a plataforma Arduino é sua facilidade de manipulação; pessoas que possuem pouco, ou até mesmo nenhum conhecimento sobre o assunto, podem rapidamente aprender os aspectos básicos da linguagem e do hardware aplicados em uma experimentação elaboradas em um curto espaço de tempo.

1.8 Introdução aos Materiais Semicondutores

Todos os objetos a nossa volta são formados por átomos. Os átomos são compostos pelos elementos básicos de sua estrutura: prótons, elétrons e nêutrons. Seu conceito define-se pelos elétrons que giram em torno de seu núcleo, e o mesmo é composto por prótons e nêutrons.

Conforme Marques (2007), os elétrons giram em órbitas, compostas por K, L, M, N, O, P e Q, e representam-se pelo modelo atômico de Bohr, conforme a figura 7. As órbitas externas situadas ao átomo possui um nível maior de energia. A última órbita do elétron denomina-se por banda de valência, ou camada de valência. A quantidade de elétrons nesse nível de órbita pode-se libertar com maior facilidade a partir de fontes externas como calor, luz ou outro tipo de radiação. Dessa forma é possível com que os elétrons da última camada se tornem livres com maior facilidade, formando assim uma banda de condução. São estes tipo de elétrons livres ao qual denominados por corrente elétrica.

Figura 8 - Descrição e Formação das Camadas dos Elétrons

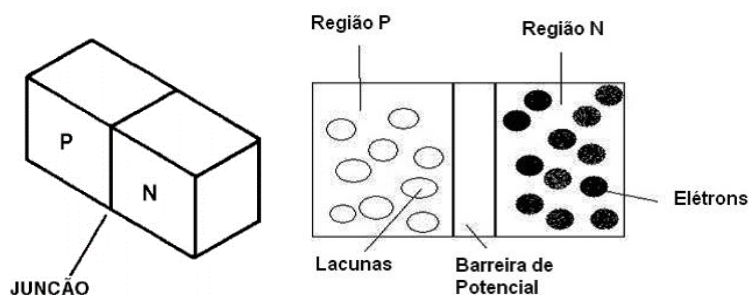


Fonte : MARQUES (2007)

Conforme Braga (1999), os semicondutores, possuem essa classificação por terem tipos diferentes de materiais, P e N, conforme pode ser visto na figura 9. Dessa maneira, na junção PN, os materiais são separados, a partir de uma barreira de potencial.

Os materiais que estão em excesso em N se deslocam para P, ou vice-versa, procurando espaços de alocação denominados por lacunas. A barreira de potencial, é uma limitação que permite que o elétron realize um salto de N para P.

Figura 10 - Características das Junções de materiais P e N



Fonte: BRAGA (2012)

A característica da barreira varia de acordo com o material. Segundo Marques (2007) e Braga (1999), de maneira geral, existem 2 tipos de materiais mais utilizados, como o silício (Si) e o Germânio (Ge). Os valores da barreira de potencial, quando o elétron realiza sua transposição do material, varia de 0,2 V (Germânio), e 0,6 V (Silício). A uma temperatura de 25°C, variam nas proporções de 0,3 V (Germânio) e 0,7 V (Silício).

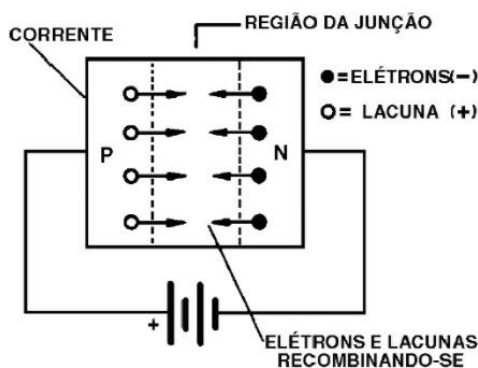
Quando se classifica uma região do tipo P, pode-se afirmar que os elétrons são portadores minoritários, enquanto que as lacunas são portadores majoritários. Todavia, quando se classifica uma região do tipo N, os portadores minoritários passam a ser as lacunas, enquanto que os portadores majoritários, os elétrons (MARQUES, 2007).

1.9 Diodos

Os diodos são estruturas formadas pelos materiais da Junção PN. Dessa forma, a corrente que flui diretamente dentro do componente, permite obtermos 2 sentidos: a sua fluidez a partir de P para N ou vice e versa. Consideremos um circuito fechado, onde possui um componente de alimentação (bateria); e um diodo semicondutor. Uma bateria, conforme afirma Braga (2010), possui seu comportamento baseando-se em geradores de energia, compostos de polaridades negativas e positivas.

Conforme pode ser observado na figura 11, o material P é ligado diretamente no polo positivo do gerador de energia, enquanto que o material do tipo N está conectado ao polo negativo. Este tipo de ligação conforme Braga (2012), mostra que os portadores de carga positiva e negativa, que encontram-se na região da junção, são polarizados diretamente. Quando isso ocorre, as cargas são “empurradas” pelo gerador na região das lacunas onde os mesmos se neutralizam; observando, portanto, o movimento dos elétrons em um fluxo constante.

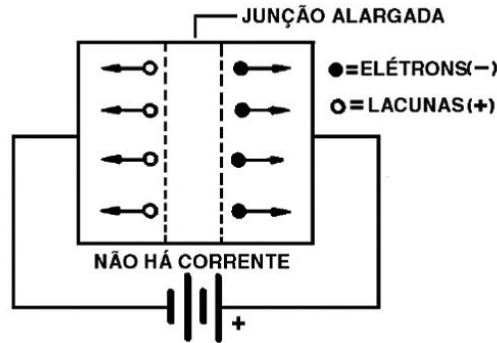
Figura 12 - Circuito com polarização direta do diodo



Fonte: BRAGA (2012)

Na figura 13, observa-se o contrário, os portadores de carga positiva e negativa são polarizados inversamente, ou seja, os portadores da junção N são atraídos pelo lado positivo da do gerador, perfazendo assim um afastamento da junção.

Figura 14 - Circuito com polarização inversa do diodo



Fonte: BRAGA (2012)

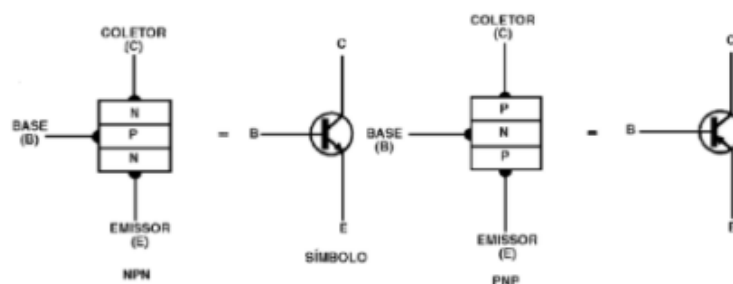
1.10 Transistores Bipolares

Um dos maiores dispositivos de interesse em pesquisa e desenvolvimento foi a válvula. A primeira válvula foi apresentada por J. A. Fleminf. No ano de 1906, Lee de Forest, inseriu um dispositivo chamado grade de controle, que permitiu controlar o fluxo de corrente entre o anodo e o catodo, resultando em um primeiro amplificador chamado tríodo.

O transistor é constituído a partir de 3 camadas derivadas dos portadores do tipo N ou P. Dessa forma, obtém-se 2 tipos de classificação: NPN e PNP. Os transistores do tipo NPN são constituídos a partir de 2 camadas do tipo N e 1 camada do tipo P; já os transistores de características PNP possuem 1 camada do tipo N e 2 camadas do tipo P. (BOYLESTAD, 2013)

A estrutura básica é baseada a partir da classificação das regiões semicondutoras denominadas por: Emissor (E), Base (B) e Coletor (C). Seu funcionamento baseia-se pelo processo de polarização, para que assim o transistor possui uma faixa adequada de funcionamento.

Figura 15 - Metodologia de funcionamento e nomenclatura eletrônica dos transistores NPN e PNP

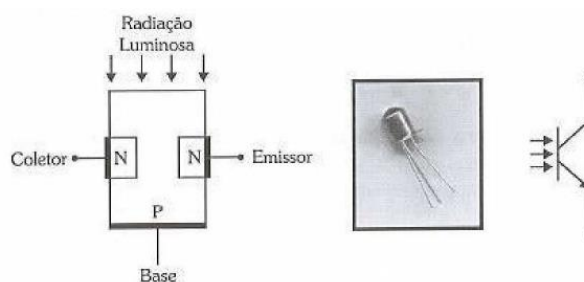


Fonte: BRAGA (2012)

1.11 Características principais de Fototransistores

A partir de sensores, é possível programar o movimento e o conjunto de ações de uma estrutura robótica. Assim, a base inicial ao qual é possível iniciar os critérios de movimentação são a partir de Sensores de Fototransistor. O fototransistor é um dispositivo que possui uma sensibilidade maior, bastante utilizado em controles remotos. Seu esquemático pode ser observado na figura abaixo:

Figura 16 - Funcionamento de um fototransistor, simbologia física e eletrônica



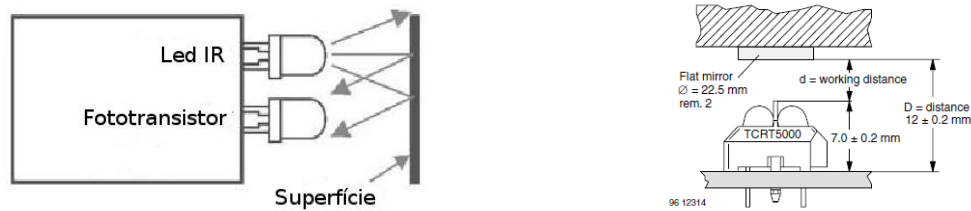
Fonte: MARQUES, et al. (2008)

O fototransistor normalmente é acompanhado por um LED. De acordo com Marques (2008), quando uma certa quantidade de luz atinge a base, uma variação de corrente no coletor além da geração de portadores., ou seja, a incidência de luz é diretamente proporcional a faixa de operação de luz visível e/ou refletida.

1.12 Robôs Seguidor de Linha.

Um robô linha é uma estrutura autônoma programada para seguir uma referência a partir de sensores, compostas por uma trajetória de desenvolvimento, usando linhas para que o robô possa acompanhar o percurso. Conforme aponta Bento (2014), os sensores são uma parte fundamental desse desenvolvimento, sendo os elementos principais que conseguem “enxergar” diferentes tonalidades de cores, e, a partir de sua luminosidade, transformar a frequência da luz em uma informação manipulável através da programação. Dessa forma os sensores que condicionam a forma com que o Robô possa detectar as variações de luz existentes são os fototransistores (detectores de luz) e os diodos infravermelhos (emissores de luz). No momento em que o diodo infravermelho emite um feixe de luz em uma determinada superfície, o mesmo retorna e é mensurado a partir do fototransistor. A figura 17, retrata o funcionamento do sensor fototransistor em contato com a superfície.

Figura 18 - Ilustração do Funcionamento de um Fototransistor em contato com a superfície



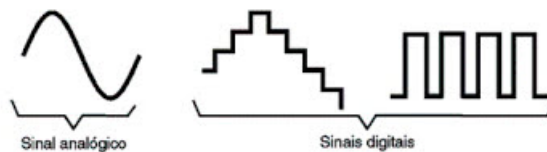
Fonte: LABORATÓRIO DE ELETRÔNICA (2017)

Fonte: VISHAY SEMICONDUCTORS (2018)

1.13 Diferenças Analógicas e Digitais

Os termos eletrônicos Analógicos e Digitais compõem as estruturas do estudo Eletrônica. Entende-se por digital, a representação derivada de dígito (do latim dedo), que está diretamente ligada a quantidade inteiras; já a analógica, os sinais podem variar em uma determinada escala. (BRAGA, 2012). Em suma, a eletrônica digital trabalha com o aspecto dos valores discretos ou inteiros, enquanto que a analógica assume qualquer valor entre dois limites.

Figura 19 - Comportamento dos sinais Analógicos e Digitais

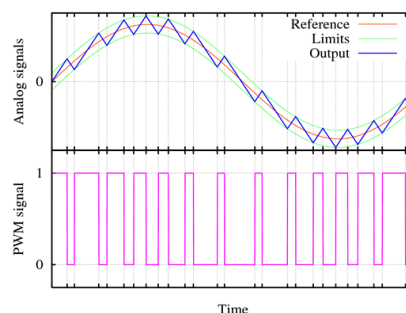


Fonte: BRAGA (2012)

1.14 Conceito de Sinais PWM (Pulse Width Modulation)

O PWM (Pulse Width Modulation ou Lardura por Modulação de Pulso) é uma técnica que obtém os resultados analógicos através dos meios digitais. Dessa forma, Torres (2012) argumenta que o conceito de PWM é empregada para o controle de brilho em lâmpadas e velocidade de motores. O formato do processo é definido a partir de uma onda quadrada com sinais oscilantes nos estados 0 e 1, ou respectivamente, desligado e ligado. Em um determinado ciclo de tempo, o sinal permanece em alta, enquanto no outro estado de tempo, o sinal permanece em baixa. A esta variação, dá-se o nome de Modulação de Pulso.

Figura 20 - Análise dos Comportamentos dos sinais analógicos e digitais no conceito de PWM



Fonte: CITISYSTEMS (2018)

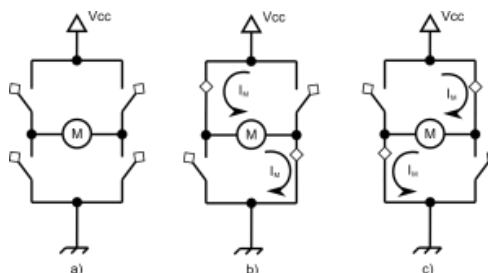
Na prática, isso demonstra que um sinal PWM pode ser representado em um formato de onda retangular. A composição de um ciclo de carga existente em um formato de onda retangular é visto a partir de percentuais não sendo apenas 50%-50% (Torres, 2012).

1.15 Motores de Corrente Contínua

Quando se trata dos aspectos de locomoção em robô, o pensamento centraliza-se na utilização de um componente bastante importante para seu processo de locomoção: o Motor DC, também conhecido como motor de corrente contínua. No entanto, a forma com que o motor deve ser inserido no sistema de alimentação deve ser trabalhado a partir do comportamento de sinais.

Um motor DC tem por característica a capacidade de movimentação a partir do sentido horário e anti-horário. Deve-se portanto, criar um sistema composto por chaveamento capaz de inverter o sentido do motor. Para obter as variáveis de controle de motor em seus dois sentidos mais sua velocidade, torque e outros dados técnicos, é necessário montar um circuito básico demonstrado na figura 31 (a), denominado por Ponte H (H-Bridge). Na figura 31 (b) e figura 31 (c); pôde -se observar que se o motor for movimentado para um dos sentidos, o mesmo o faz girar em um dos sentidos, tanto horário quanto anti-horário, apenas com as movimentações do chaveamento.

Figura 21 - Esquemático dos motores DC em Ponte H.



Fonte: ATHOS ELECTRONICS (2018)

1.16 Comunicação Bluetooth

O Bluetooth destaca-se por ser uma categoria existente nos mais diversos dispositivos no mercado como laptops, palms, fones de ouvido, dentre outros. Criado em 1998 com a união de diversas empresas como Ericsson, Intel, IBM, Toshiba, Nokia, Lucent, Motorola e outras. Sendo assim, desenvolveu-se a proposta da tecnologia bluetooth, tecnologia de baixo custo para conectividade sem fio. Em 1999, foi elaborado o padrão bluetooth cujo funcionamento garantiu o lançamento da versão 1.0 do modelo.

A faixa de frequência de operação no bluetooth varia de 2,4 Ghz a 2,483 Ghz, tecnologia conhecida como ISM (Industrial, Scientific & Medical), da qual garante a comunicação com

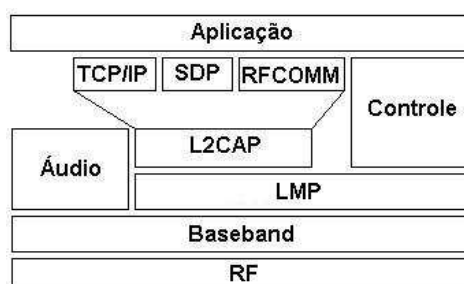
outras faixas de frequência como o Wi-fi. A faixa de operação da tecnologia bluetooth permite que a mesma possa ser utilizada em uma ampla gama de países, não sendo necessário sua aplicação de licença. (COSTA & MENDES, 2016)

O Bluetooth possui uma área de operação com alcance de sinal de 1 m a 100 m com uma taxa de transferência de 1 MB/s na versão 1.2 e 3 MB/s na versão 2.0. Seu modo de operação baseia-se em master-slave (mestre-escravo). Dá-se o nome do dispositivo ao qual realiza a conexão de mestre; o dispositivo responsável por receber os dados de conexão, denomina-se por escravo.

1.17 Protocolo de comunicação Bluetooth

Baseando-se no conceito de redes de computadores, o protocolo bluetooth possui regras predefinidas para a comunicação entre os dispositivos. Essa representatividade define-se a partir da forma em como os dados são recebidos, enviados e tratados, de acordo com o modelo de camada em específico, conforme a figura 33. (COSTA & MENDES, 2016)

Figura 22 - Arquitetura de Protocolo Bluetooth



Fonte: COSTA & MENDES (2016)

RF (Rádio Frequeccy) – Camada que define os aspectos físicos das transmissões de dados, como potência de transmissão, modulação, tolerância da variação de frequência e nível de sensibilidade do receptor;

Baseband – Camada responsável pela transmissão de bits, definindo a forma de salto de frequência os slots de tempo, o formato dos pacotes, o endereço dos dispositivos, os tipos de pacotes e os tipos de conexão (mestre e escravo);

Áudio – Camada que trabalha em conjunto com Baseband. Os dados dessa camada são enviados para Baseband, estabelecendo uma conexão direta entre dois dispositivos no formato áudio;

LMP (Link Manager Protocol) – Camada que trabalha em parceria com a camada de controle, gerenciando o controle de enlaces, consumo de energia, e os aspectos de segurança.

L2CAP (Logical Link Control and Adaptation Protocol) – Camada que gerencia processo de comunicação serial a partir das camadas superiores.

TCP/IP (Transmission Control Protocol / Internet Protocol) – Camada encontrada em processos industriais, permitindo a comunicação dos dispositivos conectados a partir da internet.

SDP (Service Discover Protocol) – Camada que permite os dispositivos obter informações sobre os serviços de outros recursos bluetooth.

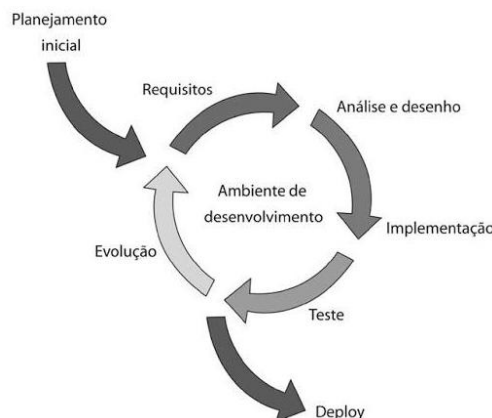
Aplicação – Camada que é desenvolvida a partir dos parâmetros especificados pelo fabricante responsável na elaboração do dispositivo.

1.18 Introdução às Etapas de Processo de Engenharia de Software

Os softwares são os elementos que fazem parte da demanda de todas as operações de negócios (SOMERVILLE, 2011). O desenvolvimento de softwares representa um ponto crítico e importante que compõem todo o seu processo de criação. É notório que uma plataforma de software, tendo por base suas características mais comuns de análise, como uma programação detalhada, documentações elucidativas, e metodologias intuitivas, demonstra um detalhamento de planos bem segmentado.

Nos termos de desenvolvimento de software, é necessário a utilização de quatro etapas fundamentais: Levantamento de requisitos, Análise e Projetos, Implementação e Testes (BEZERRA, 2006).

Figura 23 - Etapas fundamentais do desenvolvimento do software



Fonte: SOMMERVILLE (2011)

1.19 Etapa de “Levantamentos de Requisitos”

A determinação de requisitos funcionais, ocorre a partir de ações que o sistema pode executar, contendo os elementos principais de variáveis de entrada e saída. Os requisitos não funcionais do projeto, determinam as características do ambiente e seus atributos condicionados a partir do sistema, perfazendo seus principais pontos como as propriedades principais e suas características.

1.20 Etapa de “Análise e Projeto”

Nesta etapa, pode-se analisar dois requisitos bastante peculiares: os requisitos funcionais e os não funcionais. Essas espécies de requisitos são essenciais para o aprimoramento e visualização do produto final do processo.

A tabela relaciona os requisitos funcionais e não funcionais.

Tabela 2 - Requisitos Funcionais e Não Funcionais

Requisitos Funcionais	Requisitos Não Funcionais
1. Setpoint (SP) – Referência do Sensor	1. Interferências de Luz
2. Fator Proporcional (P)	2. Inclinação
3. Fator Integral (I)	3. Textura do Material
4. Fator Derivativo (D)	4. Redutores, Encruzilhadas e Verdes
5. Máximos e Mínimos	

Fonte: Elaborada pelo autor

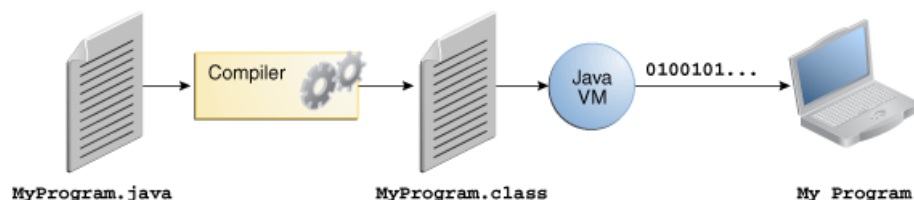
1.21 Etapa de “Implementação”

O software possui suas características das bases de programação. Neste processo pode-se obter as condições necessárias para a utilização de uma linguagem de programação. A base do software foi codificada a partir da linguagem Java. Conforme cita BARRETO (1999, p. 8), “A linguagem Java é diferente, pois seus programas são compilados E interpretados”; característica que a difere das outras linguagens que necessitam de um processo de compilação ou interpretação.

O compilador, na linguagem Java, tem-se seu processo de transformação a partir de uma linguagem intermediária, denominada bytecode. Posteriormente, os bytecodes são interpretados por um interpretador Java, conforme pode ser visto na figura 24.

Java utiliza uma variante dos bytecodes, codificados a partir de instruções de máquina para a Java Virtual Machine (ou JVM); podendo os programas derivados da linguagem java serem executados em computadores ou em browser’s (BARRETO, 1999).

Figura 25 - Ilustração do Processo de Compilação e Interpretação Java



Fonte: ORACLE JAVA DOCUMENTATION (2017)

1.22 Etapa de “Teste”

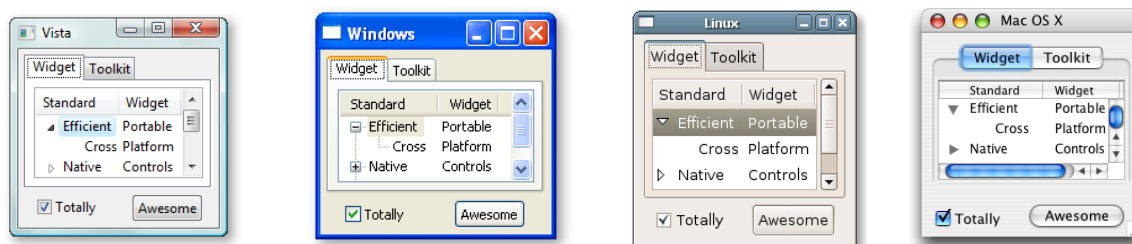
Nesta etapa, foram determinados os testes de caixa preta e teste de caixa branca para validação e análise da aplicação em execução. O teste de caixa preta verifica as funcionalidades do sistema, sem observar o código fonte de modo que sejam verificadas as saídas dos dados. O teste de caixa branca verifica o código fonte de modo a analisar a estrutura do código e se as variáveis declaradas estão relacionadas à ação do sistema e também se erros são gerados.

1.23 Eclipse IDE (Integrated Development Environment)

O Eclipse é uma IDE (Integrated Development Environment) é um ambiente que auxilia no desenvolvimento de softwares, criada pela IBM em parceria com outras empresas adeptas à linguagem Java. Além da IDE Eclipse, é necessário a instalação de pacotes e configurações dos módulos em Java. As ferramentas existentes no eclipse permitem a exploração de conteúdos e materiais de exemplo que podem ser vistos dentro da própria IDE.

No ambiente Eclipse, é possível utilizar kits de ferramentas que agilizam o processo de criação de tarefas e desenvolvimento dos processos de software. Uma dessas ferramentas, que permitem a o desenvolvimento de interface gráfica, é denominada SWT (Standard Widget Toolkit), compostas por widgets destinadas à criação de códigos Java com maior agilidade, fornecendo interfaces de acordo com cada o sistema operacional que o mesmo queira executar.

Figura 26 - Representação de Interfaces desenvolvida em Java SWT adaptada para diversos Sistemas Operacionais



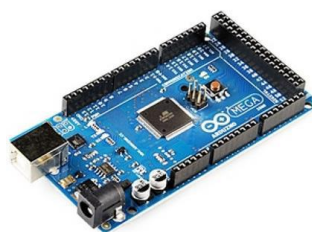
Fonte: ECLIPSE (2018)

3 RESULTADOS E DISCUSSÃO

Com base nos componentes necessários e básicos para o modelamento de uma estrutura robótica, o projeto utiliza dois motores CC para coordenar o avanço do motor. Além disso, é necessário utilizar uma fonte de alimentação que ofereça tensão suficiente para alimentar Shields, Motores e a própria placa Arduino (MCROBERTS, 2011). Conforme as especificações do fabricante, o Arduino Mega possui uma arquitetura baseado em um microcontrolador ATmega versão 2560. Possui 64 pinos de comunicação digital, dos quais 15 podem ser usados no modo PWM, com 16 entradas analógicas, 4 UARTS (portas de hardware), um oscilador de cristal de 16Mhz, uma conexão USB, um conector de tomada de energia, um cabeçalho ICSP,

e um botão reset disponível na placa. O intervalo de tensão de operação recomendado para o sistema de alimentação do Arduino Mega 2560 é de 7 ~ 12 volts. Tensões abaixo, ou acima do nível estipulado pelo fabricante, podem danificar ou gerar comportamentos inesperados. Outras características que envolvem a concepção do Arduino Mega 2560 (figura 27) é o tipo de prototipagem que segundo Oliveira, Santos e Rodrigues (2014), além do desenvolvimento permitir o desenvolvimento a partir da linguagem C/C++, através de uma IDE escrita em Java, a plataforma é uma prototipagem de hardware para o desenvolvimento livre.

Figura 28 - Arduino Mega 2560



Fonte: OLIVEIRA, et al. (2014)

Para o projeto, é necessário a conexão da Shield nos dos 2 motores CC (figura 29) . Cada motor foi conectado através das portas de saídas M3 e M4 . A estrutura do Robô é delimitada pela base compostas de anexos e furos de conexão, para incorporar os circuitos e sensores necessários para a movimentação do Robô segue-linha.

A codificação da programação do Robô foi escrita em Arduino e Java, utilizando as bibliotecas necessárias para a incorporação do motor shield e seus movimentos.

Figura 30 - Motorshield e base de apoio para o desenvolvimento do projeto.



Fonte: FILIPEFLOP (2017)

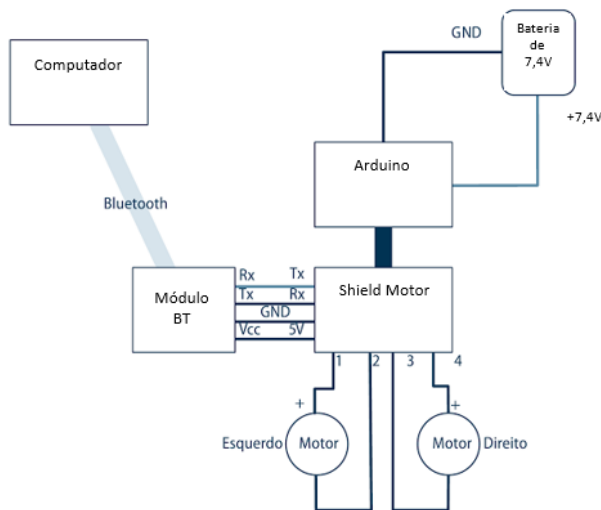
O módulo Bluetooth RS 232 HC-05, fornece uma maneira fácil e barata para incorporação de processos de comunicação sem fio. O alcance do dispositivo é de 10 m.

Figura 31 - Módulo RS 232 HC-05

Fonte: OLIVEIRA, et al. (2014)

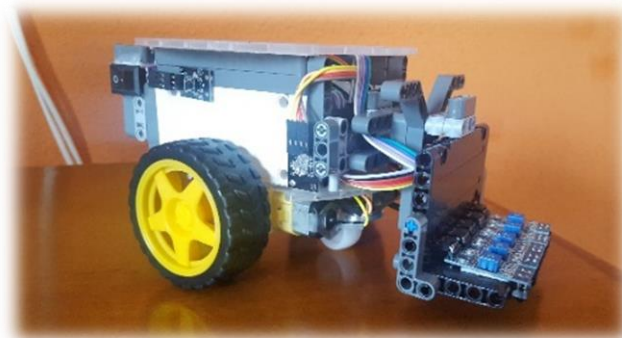
O módulo possui um regulador de tensão que controla as faixas de operações entre 3,3V a 5V. Oliveira, Santos e Rodrigues (2014), descrevem o processo de comunicação do módulo bluetooth RS 232 HC-05, só é possível se o mesmo obtiver um processo de instalação e configuração de software, dependendo de da versão do Sistema Operacional utilizado.

A partir de um escopo de desenvolvimento especificado por Monk (2014), é possível obter os parâmetros necessários que referenciam o desenvolvimento da base estrutural. Além dos motores DC conectados em suas devidas saídas, o sistema de alimentação utilizando bateria corresponde ao diagrama esquemático listado na figura 32. Monk (2014) cita que a necessidade da soldagem de pinos extras no Shield permite que mais conexões possam ser delegadas e unidas em um só ponto, o que torna possível uma maior aplicabilidade.

Figura 33 - Diagrama esquemático do desenvolvimento do robô (Adaptado)

Fonte: MONK (2014)

O desenvolvimento estrutural do robô foi elaborado em conjunto com as peças do *kit* LEGO ®Mindstorms. Foi adquirido um sistema de carenagem baseado em 2 rodas conforme especificado por McRoberts (2011).

Figura 34 - Visão panorâmica do robô

Fonte: Elaborada pelo autor

O módulo de programação Arduino, foi desenvolvido juntamente com a interface Java no Eclipse IDE. Inicialmente a programação foi dividida em sketches para melhor organização. Na prática, os sketches em Arduino são arquivos localizados e postos em grupos de abas para uma melhor organização. Nas abas ativas foram declarados os arquivos contendo as definições de blocos de códigos PID necessários para a execução dos cálculos de ajustes de variáveis. O arquivo denominado na programação por **PID.ino** incorporação técnicas e cálculos de PID, além do recebimento dos dados da mesma. É composta pela função `calculatePID()` onde os dados de K_p , K_i e K_d são utilizados para o cálculo de variáveis.

Figura 35 - Função de Cálculo do PID

```

1 void calculatePID()
2 {
3     P = error;
4     I = I + error;
5     D = error - previousError;
6     PIDvalue = (Kp * P) + (Ki * I) + (Kd * D);
7     previousError = error;
8 }

```

Fonte: LABORATÓRIO DE GARAGEM (2017)

No entanto, além da efetiva função de cálculo do PID, é necessário que essa compensação seja distribuída para os motores. Com isso, desenvolveu-se uma função denominada `motorPIDControl()`. Nas linhas 12 e 13 é possível observar a compensação do motor `M1Speed` e `M2Speed` com a variável `PIDvalue`. A função `constrain()` vai garantir o intervalo de velocidade do motor e que os termos sejam controlados. Logo após, nas linhas 18 e 19 da figura 28, o modelo de compensação é aplicado na velocidade e impulsionando o robô à frente com a função `setMotorFowardd()`.

Figura 36 - Transferência de Função PID para motor

```

10 void motorPIDcontrol()
11 {
12     int M1speed = iniMotorPower - PIDvalue;
13     int M2speed = iniMotorPower + PIDvalue;
14
15     constrain(M1speed, 100, 200);
16     constrain(M2speed, 100, 200);
17
18     motor_M1.setSpeed(M1speed);
19     motor_M2.setSpeed(M2speed);
20
21     setMotorFoward();
22
23 }

```

Fonte: LABORATÓRIO DE GARAGEM (2017)

A necessidade de controle da variável erro é fundamental, visto que o PID necessita o valor do erro para realizar seu processo de calibração. Com isso a função readLFSensors() trabalha com as cláusulas if em combinação com a lógica booleana para controlar os espaços de acordo com a tabela de erros e condições existentes para que o robô percorra a linha. Inicialmente é realizada a leitura dos sensores de refletância e suas atribuições no momento real. Em seguida, essas leitura que tiveram suas atribuições realizadas a partir do vetor, é transportada como leitura serial para a classe arduino, certificando uma rotina de teste da leitura. Após a conclusão deste processo, as cláusulas If atribuem o valor do erro de acordo com o posicionamento do robô.

Figura 37 - Condição de Existência da Variável Erro

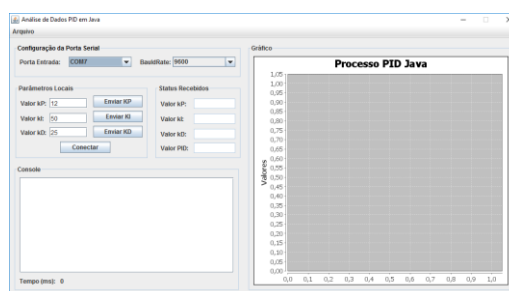
```

25 void readLFSensors()
26 {
27     readDigitalTCRT5000();
28
29     monitorDigitalTCRT5000();
30
31     if((LightValue[0]== 0 )&&(LightValue[1]== 0 )&&(LightValue[2]== 0 )&&(LightValue
32 [3]== 0 )&&(LightValue[4]== 1 )) {mode = FOLLOWING_LINE; error = 4;}
33 else if((LightValue[0]== 0 )&&(LightValue[1]== 0 )&&(LightValue[2]== 0 )&&(LightValue
34 [3]== 1 )&&(LightValue[4]== 1 )) {mode = FOLLOWING_LINE; error = 3;}
35 else if((LightValue[0]== 0 )&&(LightValue[1]== 0 )&&(LightValue[2]== 0 )&&(LightValue
36 [3]== 1 )&&(LightValue[4]== 0 )) {mode = FOLLOWING_LINE; error = 2;}

```

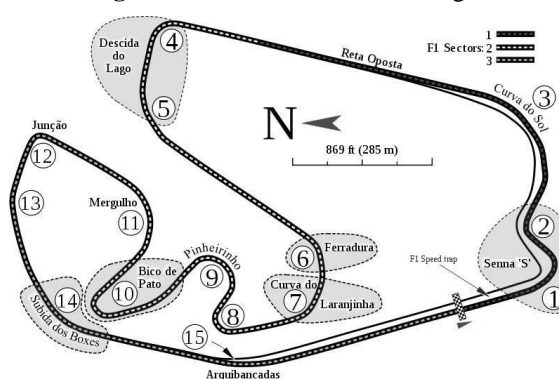
Fonte: LABORATÓRIO DE GARAGEM (2017)

No desenvolvimento da programação Java foi utilizado o modelamento de interface com o recurso SWT (Standard Widget Toolkit). O SWT no Java permite criar as interfaces GUI através de uma maneira produtiva. Como é possível observar na imagem 30, a tela da interface é composta do desenvolvimento de menus, dispostos na parte superior do software. O objetivo principal do programa é coletar as amostras do processamento PID no circuito de Interlagos, impresso em tamanho de 1,20 m x 2,00 m, em formato de lona, para detectar quais os ajustes necessários para o método de sintonia de controle das variáveis Kp, Ki e Kd, suavizando o movimento do robô.

Figura 38 - Interface do Módulo de Recepção Java

Fonte: Elaborada pelo autor

A etapa de análise do controle de três termos (PID – Proporcional Integrador Derivativo), envolveu a experimentação de processos através da configuração das variáveis de controle. Com isso, conforme Brito, Maladosso e Guibes (2014), a estrutura robótica tinha como objetivo, executar o roteiro em determinados pontos de um dado circuito no formato do Autódromo de Interlagos (figura 31). A principal característica do circuito é a composição de curvas íngremes e acentuadas.

Figura 39 - Autódromo de Interlagos

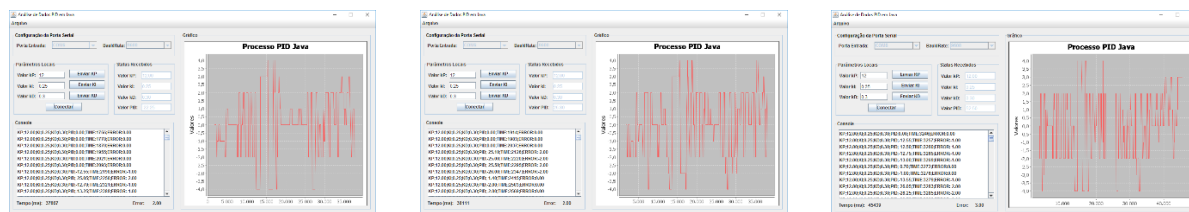
Fonte: BRITO, R. C., MADALOSO, E. & GUIBES, G. A. O. (2014)

As numerações do circuito foram utilizadas como referência para o percurso do robô. Inicialmente foram caracterizados o roteiro do robô, com 3 lançamentos subsequentes, a partir dos seguintes pontos:

- 15 - Arquibancadas
- 1 e 2- Senna 'S'
- 3 - Curva do Sol e Reta Oposta
- 4 e 5 – Descida do Lago
- 6 – Ferradura
- 7 – Curva do Laranjinha

O software realizou as 3 etapas de leitura e acompanhamento das variáveis em tempo real. E com isso gerou um log de dados, mais um processo de leitura que pode ser observado pela figura 32.

Figura 40 - Análise de Amostragem das Condições PID



Fonte: Elaborada pelo autor

Com os dados emitidos pelo gráfico nos percursos citados, observa-se o comportamento dos modelos de variáveis atribuídas pelo Método da Tentativa e Erro. As definições mostram o comportamento que a referencia dada inicialmente pelo Setpoint (com o valor 0) nos modelos pré-programados, tende a procura constante da variável 0. Por mais que o circuito tenha curvas, ou mesmo a velocidade dos motores, sejam fatores que mudam a estrutura da rota na linha preta, os critérios programados a partir dos valores PID, realizam a manobra necessária para manter o robô dentro dos parâmetros do roteiro de pista, de maneira suave e precisa. Os eixos dos gráficos são definidos a partir do erro da variável PID versus o tempo. O robô teve o comportamento necessário para com os retornos de manobra nas curvas e nos pontos citados.

4 CONCLUSÃO

O desenvolvimento e a utilização de testes incorporadas a partir da engenharia de requisitos, utilizou o Circuito de Interlagos como roteiro de percurso. O software desenvolvido foi partir da linguagem de programação Java, no ambiente de desenvolvimento Eclipse IDE, em conjunto com o plugin SWT, juntamente com uma estrutura robótica seguidor de linha desenvolvido na plataforma Arduino IDE. Com a pretensão do desenvolvimento do software para coleta e recepção de dados da análise e processos de variáveis PID, obteve-se um grau de desenvolvimento satisfatório, no intuito de otimizar os processos do Controle de Três Termos a partir do Método da Tentativa e Erro. Os dados extraídos representam o desenvolvimento real no quesito de desenvolvimento da estrutura de um robô seguidor de linha para a robótica móvel.

As dificuldades existentes no processo de elaboração e descoberta das variáveis de controle, foram um dos aspectos que geravam instabilidades no roteiro de padronização do robô. Um dos detalhes de maior rendimento para o PID, foi o processo de força para gerar o movimento do robô. A mudança de velocidade, muitas vezes, gerava instabilidades no robô. Com isso, momentaneamente, a estrutura gerava falha de percurso no seguidor de linha, fazendo com que o mesmo realizava manobras inadequadas, fazendo-o desviar do circuito. O processo de comunicação via Bluetooth, também foi uma das condições que envolveu padrões e

algoritmos de alinhamento para que as variáveis se atribuíssem nos valores especificados no Controle de Três Termos.

Em uma análise em tempo real, observa-se graficamente o comportamento das variáveis PID para o encontro da referência do seguidor de linha. As técnicas de controle permitem de maneira eficaz e prática os ajustes de controle de três termos bastante simplificada. O controle PID demonstrou-se uma técnica prática que pode ser aplicado a processos simples e complexos obtendo resultados bastante eficientes.

5 REFERÊNCIAS

AFARI, E. & KHINE, M. S., 2017. **Robotics as an Educational Tool: Impact of Lego Mindstorms**. Disponível em: <<http://www.ijiet.org/vol7/908-T108.pdf>>.

Acesso em: 20 dez. 2017.

ANON, 2003. **A System for Volumetric Robotic Mapping of Abandoned Mines**.

Disponível em: <https://www.ri.cmu.edu/pub_files/pub4/thrun_sebastian_2003_1/thrun_sebastian_2003_1.pdf>. Acesso em 13 nov. 2017.

AROCA, R. V. et al., 2014. **Olimpíada Brasileira de Robótica: relatos da primeira regional em São Carlos-SP**. Disponível em: <<http://www.producao.usp.br/handle/BDPI/48596>>. Acesso em 12 jan. 2018.

ATHOS ELECTRONICS. **Ponte H: Tutorial com Transistores TIP120**. Disponível em: <<https://athoselectronics.com/ponte-h/>>. Acessado em 18 jan. 2018.

BANZI, Massimo. **Getting Started with Arduino**. 2. nd. Califórnia: Books, 2011. p. 95-106

BARBOSA, J. L. F. B., s.d. **Controle de Malha Aberta e Fechada**. Disponível em: <<http://www.ebah.com.br/content/ABAAABM7MAE/aula-02-controle-malha-aberta-fechada>>. Acesso em 09 nov. 2017.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. 2ª ed. Rio de Janeiro: Campus, 2006. p. 21-30.

BOYD, Stephen & BARRATT, Craig. **Linear Controller Design: Limits of Performance**. 6ª ed. s.l.:Prentice Hall, 1991. cap. 1-2.

BOYLESTAT, R. L. **Dispositivos Eletrônicos e Teoria dos Circuitos**. 11ª ed. São Paulo: Pearson Education do Brasil, 2013. cap. 3-6.

BRAGA, Newton C. **Eletrônica Analógica**. 1ª ed. São Paulo: Instituto NCB, 2012. cap. 2.

BRAGA, Newton C. **Eletrônica Básica**. 1ª ed. São Paulo: Instituto NCB, 2012. cap. 1-3-4.

BRAGA, Newton C. **Eletrônica Digital – Parte 1**. 1ª ed. São Paulo: Instituto NCB, 2012. p. 8.

- BRITO, R. C., MADALOSO, E. & GUIBES, G. A. O., 2014. **Seguidor de linha para LEGO ® Mindstorms utilizando controle PID**. Disponível em: <<https://siaiap32.univali.br/seer/index.php/acotb/article/view/5335>>. Acesso em 25 out. 2017.
- BURNETTE, E., 2009. **Eclipse IDE Pocket Guide, Using the Full-Featured IDE**. Sebastopol: O'Reilly Media. p. 15-16.
- COSTA, R. A. A. & MENDES, L. A. M., 2016. **Evolução das Redes Sem Fio: Um Estudo Comparativo Entre Bluetooth e ZigBee**. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-a010b188f93af4c28ca9af23b9e3c476.pdf>>. Acesso em 22 jan. 2018.
- CITYSYSTEMS. **O controle PID de Forma Simples e Descomplicada**. Disponível em: <<https://www.citisystems.com.br/controle-pid/>>. Acesso em 14 set. 2017.
- CITYSYSTEMS. **O que é PWM e Para que Serve?**. Disponível em: <<https://www.citisystems.com.br/pwm/>>. Acessado em 23 jan. 2018.
- DORF, Richard C. & BISHOP, Robert H.. **Sistemas de Controle Modernos**. Rio de Janeiro: LTC Editora, 1998. p. cap. 1-7-12.
- ECLIPSE. **SWT: The Standard Widget Toolkit**. Disponível em: <<https://www.eclipse.org/swt/>>. Acessado em 17 de Jan. de 2018.
- FERREIRA, R. K., s.d. **Robôs Invadem a Escola: A Robótica Educacional em Busca de Cérebros Flexíveis**. Disponível em: <http://www.sbece.com.br/resources/anais/7/1493088954_ARQUIVO_Robosinvademaescola-aroboticaeducacionalembuscadecerebrosflexiveis.pdf>. Acesso em 14 nov. 2017.
- FILIFELOP. **Como montar um Robô Seguidor de Linha com Arduino Motor Shield**. Disponível em <<https://www.filifeelop.com/blog/projeto-robo-seguidor-de-linha-arduino/>>. Acesso em 19 out. 2017.
- FRANCHI, Clayton Moro. **Acionamentos Elétricos**. 4ª ed. São Paulo: Érica, 2008. cap. 1.
- FURGERI, S. **Java 7 - Ensino Didático**. 2ª ed. São Paulo: Érica, 2012.
- GONÇALVES, Roselito Ferreira et. al., 2016. **Robôs de Resgate com Arduino: Proteus e Toth**. Disponível em: <http://fablearn.org/wp-content/uploads/2016/09/FLBrazil_2016_paper_124.pdf> Acesso em 14 nov. 2017.
- JUNIOR, N. V. & SILVA, A. L. V., 2009. **O Uso de Protótipos Para o Ensino De Robótica E Desenvolvimento Dos Modelos Mentais**. Disponível em: <<http://www.bibliotekevirtual.org/index.php/2013-02-07-03-02-35/2013-02-07-03-03-11/242-abenge/v28n02/2125-v28n02a06.html>>. Acesso em 13 nov. 2017.

LABORATÓRIO DE ELETRÔNICA. **Sensor de Linha com Arduino**. Disponível em:

<<http://labdeeletronica.com.br/sensor-de-linha-arduino/>> Acessado em 19 dez. 2017.

LABORATÓRIO DE GARAGEM. **Tutorial: Robô seguidor de linha com controle PID e ajustes por aplicativo Android**. Disponível em:

<<http://labdegaragem.com/profiles/blogs/tutorial-rob-seguidor-de-linha-com-controle-pid-e-ajustes-por>>. Acessado em 14 de set. de 2017.

LIMA, F. R. M., 2015. **LEGO® ZOOM: Ferramenta para Obtenção de dados**

Experimentais na Física Para o Ensino Fundamental. Disponível em:

<<http://www1.fisica.org.br/mnpef/?q=dissertacao/lego-zoom-ferramenta-para-obten%C3%A7%C3%A3o-de-dados-experimentais-na-f%C3%ADsica-para-o-ensino>>.

Acesso em 27 dez. 2017.

MARQUES, A. E. B., JÚNIOR, S. C. & CRUZ, E. C. A. **Dispositivos Semicondutores:**

Diodos e Transistores. 12ª ed. São Paulo: Érica Ltda, 2008. cap. 2-3-6-15.

MATHWORKS. **Unified Tuning of PID Load Frequency Controller for Multi Area**

Power Systems via IMC. Disponível em:

<<https://www.mathworks.com/matlabcentral/fileexchange/58257-unified-tuning-of-pid-load-frequency-controller-for-multi-area-power-systems-via-imc>>. Acesso em: 19 out. 2017.

MATIAS, J., 2002. **Teoria de Controle PID**. Disponível em:

<http://coral.ufsm.br/beltrame/arquivos/disciplinas/medio_automacao_industrial/Artigo_Teoria_controle_PID.pdf>. Acesso em 01 nov. 2017.

MCROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec, 2011.

MERKOURIS, A. & CHORIANOPOULOS, K., 2015. **Introducing Computer**

Programming to Children through. Disponível em:

<<https://users.ionio.gr/~choko/publications/Introducing%20Computer%20Programming%20to%20Children%20through%20Robotic%20and%20Wearable%20Devices.pdf>>. Acesso em 13 nov. 2017.

MONK, Simon. **Projetos com Arduino e Android - Use seu Smartphone ou Tablet para**

Controlar o Arduino. Porto Alegre - RS: Bookman Companhia, 2014. cap. 1.

NICOLOSI, Denys E. C.. **Microcontrolador 8051 Detalhado**. 6ª ed. São Paulo: Erica, 2000. cap. 1.

OGATA, K.. **Engenharia de Controle Moderno**. 4ª ed. s.l.:Prentice-Hall, 2003.

OLIVEIRA, I. R. H., SANTOS, C. R. B. & RODRIGUES, M. A. L., 2014. **Desenvolvimento de um Aplicativo Android para Monitoramento Microcontrolado do Nível de um**

Reservatório de Água Residencial em Tempo Real. Disponível em:

<http://www.ceel.eletrica.ufu.br/artigos2014/ceel2014_artigo009_r01.pdf>. Acesso em 23 01 2018.

OLIVEIRA, R. A. R., 2003. **Bluetooth e Multimídia**. Disponível em: <<http://laplace.dcc.ufmg.br/npdi/uploads/c8962954-8be0-43ed.pdf>>. Acesso em 22 jan. 2018.

ORACLE JAVA DOCUMENTATION. **About the Java Technology**. Disponível em: <<https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>>. Acesso em: 26 dez. 2017.

PIO, J. L. d. S., CASTRO, T. H. C. d. & JUNIOR, A. N. d. C., 2006. **A Robótica Móvel como Instrumento de Apoio à Aprendizagem de Computação**. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/510/496>>. Acesso em 11 nov. 2017.

REVISTA MUNDO DA ROBÓTICA. **Como Tudo Começou!. Mundo da Robótica**. Disponível em: <www.obr.org.br/revista/>. Acesso em: 14 nov. 2017.

ROVAI, M., 2016. **Robô seguidor de linha com controle PID e ajustes por aplicativo Android**. Disponível em: <<http://labdegaragem.com/profiles/blogs/tutorial-rob-seguidor-de-linha-com-controle-pid-e-ajustes-por>>. Acesso em 20 set. 2017.

SARAIVA, F. d. A., 2011. **Métodos de Sintonia de Controladores PID**. Disponível em: <https://biblioteca.unilasalle.edu.br/docs_online/tcc/graduacao/engenharia_telecomunicacoes/2011/fasaraiva.pdf>. Acesso em 09 nov. 2017.

SILVEIRA, P. R. d. & SANTOS, W. E. d., 1999. **Automação e Controle Discreto**. 1ª ed. São Paulo: Érica.

SOMMERVILLE, I., 2011. **Engenharia de Software**. 9ª ed. São Paulo: Pearson Prentice Hall.

SOUZA, F., 2013. **Arduino UNO**. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em 04 jan. 2018.

STALLINGS, W., 2003. **Arquitetura e Organização de Computadores**. São Paulo: Prentice Hall.

SULLIVAN, D. O. & IGOE, T., 2004. **Physical Computing**. Boston : Thomson Course Technology PTR.

TORRES, Gabriel. **Eletrônica para Autodidatas, Estudantes e Técnicos**. Rio de Janeiro: Novaterra Editora. 2012, cap. 4-27-29-31.

VISHAY SEMICONDUCTORS. **Reflective Optical Sensor with Transistor Output**. Disponível em: <<https://www.vishay.com/docs/83760/tcrt5000.pdf>>. Acesso em 03 jan. 2018.

XUE, D., CHEN, Y. & ATHERTON, D. P., 2007. **PID Controller Design**. s.l.: s.n.

6 AGRADECIMENTOS

À Deus que sempre esteve presente em minha vida, a minha grande família, alicerce de todas as horas, ao meu grande Amor, e ao nosso grande amigo e exímio Ayrton Senna do Brasil.

7 SOBRE O AUTOR

Ricardo Rodrigues dos Santos, graduado em Engenharia de Computação, graduando em Engenharia Elétrica, Analista de Suporte em Informática na empresa SESI São Paulo. Projetista e Analista de ferramentas Controle e Automação CAE/CAD com ênfase em softwares de Geoprocessamento de Dados (AutoCAD, Topcon Tools, TopoEVN e TopoGRAPH). Técnico da Equipe SESI Macgyvers, com título de 2º lugar no Torneio Regional em 2017 pela OBR (Olimpíadas Brasileira de Robótica).

E-mail: ricardoinforodri@gmail.com / ricardo.santos@sesisp.org.br