

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

IGOR DE CAMPOS
RAFAEL SANTANA FARIA

**ELABORAÇÃO DE UM SISTEMA DE CONTROLE DE
VELOCIDADE LONGITUDINAL UTILIZANDO CONTROLADORES
PID E MÉTODOS HEURÍSTICOS DE SINTONIA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2017

IGOR DE CAMPOS
RAFAEL SANTANA FARIA

**ELABORAÇÃO DE UM SISTEMA DE CONTROLE DE
VELOCIDADE LONGITUDINAL UTILIZANDO CONTROLADORES
PID E MÉTODOS HEURÍSTICOS DE SINTONIA**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR) como requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Elder Oroski

Coorientador: Prof. Dr. Alexandre José Tuoto Silveira Mello

CURITIBA
2017

Igor de Campos
Rafael Santana Faria

Elaboração de um sistema de controle de velocidade utilizando controladores PID e métodos heurísticos de sintonia

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro de Controle e Automação, do curso de Engenharia de Controle e Automação do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 13 de Junho de 2017.

Prof. Paulo Sérgio Walenia, Esp.
Coordenador de Curso
Engenharia de Controle e Automação

Prof. Marcelo de Oliveira Rosa, Dr.
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia de Controle e Automação do DAELT

ORIENTAÇÃO

Elder Oroski, Dr.
Universidade Tecnológica Federal do Paraná
Orientador

Alexandre José Tuoto Silveira Mello, Dr.
Universidade Tecnológica Federal do Paraná
Co-Orientador

BANCA EXAMINADORA

Rafael Fontes Souto, Dr.
Universidade Tecnológica Federal do Paraná

Ismael Chiamenti, Dr.
Universidade Tecnológica Federal do Paraná

Alexandre José Tuoto Silveira Mello, Dr.
Universidade Tecnológica Federal do Paraná

RESUMO

CAMPOS, Igor; FARIA, Rafael. **Elaboração de um sistema de controle de velocidade longitudinal utilizando controladores PID e métodos heurísticos de sintonia.** 2017. 97 f. Trabalho de Conclusão de Curso - Curso de Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná, Curitiba, 2017

Este trabalho de conclusão de curso tem como objetivo obter o controle em malha fechada da velocidade de um automodelo com motor de combustão interna. Os desafios são explicados a partir de revisões bibliográficas e também da aplicação da teoria em um sistema real, incluindo a construção de uma estrutura de testes. A velocidade é medida com um sensor digital conectado a um microcontrolador, que realiza os cálculos necessários e comanda o atuador que controla a velocidade do automodelo. A partir dos modelos de Hammerstein-Wiener, testes determinam um modelo não-linear para o sistema. Estes modelos são validados e depois servem para a sintonia dos controladores PID por software. Esta sintonia é realizada pelo Algoritmo Genético. Os controladores PID são testados no automodelo e seus desempenhos são comparados. Assim, trabalho envolve conhecimentos em identificação de sistemas, controle e sintonia utilizando métodos heurísticos, eletrônica e mecânica.

Palavras-chave: Controle. Malha fechada. Automodelo. Motor de combustão interna. Controladores PID. Sensor digital. Microcontrolador. Hammerstein. Wiener. Não linear. Sintonia. Algoritmo Genético. Identificação de sistemas. Eletrônica.

ABSTRACT

CAMPOS, Igor; FARIA, Rafael. **Elaboration of a longitudinal speed control system using PID controllers and heuristic tuning methods**. 2017. 97 f. Trabalho de Conclusão de Curso - Curso de Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná, Curitiba, 2017

It has a main purpose of obtaining a closed loop speed control of an internal combustion engine scale car. The challenges are explained based on bibliography review and also by applying theory in a real system, including the construction of a test base. The speed is measured by a digital sensor connected to a microcontroller, which performs the necessary calculations and drives the actuator which controls the scale model speed. From Hammerstein-Wiener models, tests have provided a non-linear model for the system. This model is then validated and used as base for the PID tuning via software. The tuning is performed by the Genetic Algorithm. The PID controllers are tested in the scale model and their performances are compared against each other. Therefore, this project enwraps knowledge in system identification, control and tuning using heuristic methods, electronics and mechanics.

Key words: Control. Closed loop. Scale model. Internal combustion engine. PID Controller. Digital sensor. Microcontroller. Hammerstein. Wiener. Non-linear. Tuning. Genetic Algorithm. System identification. Electronics.

LISTA DE FIGURAS

FIGURA 1	– Operação do acelerador - Fonte: (TEETOR, 1950)	12
FIGURA 2	– Seção transversal de um carburador - Adaptado de (YOURDICTIONARY, 2016)	12
FIGURA 3	– Esquemáticos dos circuitos digitais que implementam o <i>Cruise Control</i> - Fonte: (OTHERS, 1970) e (WISNER, 1971)	13
FIGURA 4	– Força de arrasto aerodinâmico - Fonte: (KO, 2016)	15
FIGURA 5	– Modelo em escala de veículo de combustão interna	18
FIGURA 6	– Fluxograma do Trabalho - Fonte: Autoria própria	18
FIGURA 7	– Corte lateral de um motor de combustão interna - Adaptado de (GANESAN, 1995)	21
FIGURA 8	– Ponto morto superior e inferior - Adaptado de (GANESAN, 1995)	21
FIGURA 9	– Curso de admissão - Adaptado de (GANESAN, 1995)	24
FIGURA 10	– Curso de compressão - Adaptado de (GANESAN, 1995)	24
FIGURA 11	– Curso de expansão - Adaptado de (GANESAN, 1995)	25
FIGURA 12	– Curso de escape - Adaptado de (GANESAN, 1995)	25
FIGURA 13	– Ciclos de trabalho de motor de compressão - Adaptado de (GANESAN, 1995)	26
FIGURA 14	– Ciclos de motor dois tempos - Adaptado de (GANESAN, 1995)	27
FIGURA 15	– Sistema de controle e seus componentes - Autoria própria	30
FIGURA 16	– Comparação de respostas ao degrau unitário de Controlador P - Autoria própria	31
FIGURA 17	– Comparação de respostas ao degrau unitário de Controlador PI - Autoria própria	32
FIGURA 18	– Comparação de respostas ao degrau unitário de Controlador PD - Autoria própria	33
FIGURA 19	– Resposta ao degrau para sistema com controlador PID - Autoria própria ..	34
FIGURA 20	– Conversão de sinal analógico para digital - Adaptado de (PHILLIPS; HABOR, 1995)	34
FIGURA 21	– Estrutura de um sistema de controle digital - Adaptado de (PALM, 1988) ..	35
FIGURA 22	– Pseudo-algoritmo genético. Autoria própria.	37
FIGURA 23	– Seleção. Autoria própria.	39
FIGURA 24	– Cruzamento. Autoria própria.	39
FIGURA 25	– Mutação. Autoria própria.	40
FIGURA 26	– Acoplamento mecânico entre servomotor e carburador	42
FIGURA 27	– Agulhas de Alta e Baixa	42
FIGURA 28	– Estrutura modelo. Retirado de http://www.dynomitedynamometer.com/ .	43
FIGURA 29	– Primeira estrutura. Autoria própria.	43
FIGURA 30	– Rolos de treino para bicicleta. Retirado de goo.gl/2df2SF	45

FIGURA 31	– Segunda Estrutura. Autoria própria.	45
FIGURA 32	– Estrutura final. Autoria própria.	46
FIGURA 33	– Encaixe frontal. Autoria própria.	47
FIGURA 34	– Encaixe do aerofólio. Autoria própria.	47
FIGURA 35	– Placa de madeira para fixação do sensor. Autoria própria.	47
FIGURA 36	– Placa de madeira que bloqueia o combustível expelido. Autoria própria. .	48
FIGURA 37	– Faixas brancas/pretas. Autoria própria.	49
FIGURA 38	– Sensor óptico TCRT5000. Retirado do <i>datasheet</i> do fabricante.	49
FIGURA 39	– Esquemático do sensor. Retirado do <i>datasheet</i> do fabricante.	50
FIGURA 40	– Teste de detecção para superfícies planas. d é a distância do cartão refletor, I_{Crel} é a corrente relativa no coletor. Adaptado do <i>datasheet</i> do fabricante.	50
FIGURA 41	– Configuração básica do sensor óptico. Autoria própria.	51
FIGURA 42	– Esquemático do circuito utilizado. Autoria própria.	51
FIGURA 43	– Simulação do circuito no <i>LTSPice</i> . Autoria própria.	53
FIGURA 44	– Curvas de sinais do sensor. Autoria própria.	53
FIGURA 45	– Exemplo de comparador inversor com histerese. Retirado do <i>datasheet</i> do fabricante.	54
FIGURA 46	– Saída do comparador. Autoria própria.	55
FIGURA 47	– Saída do comparador capturada no osciloscópio. Autoria própria.	55
FIGURA 48	– Circuito confeccionado artesanalmente. Autoria própria.	56
FIGURA 49	– (a) modelo de Hammerstein, (b) modelo de Wiener. Em aplicações práticas $v(k)$ é desconhecido (AGUIRRE, 2004).	57
FIGURA 50	– Curva de entrada para identificação do automodelo. Autoria própria.	58
FIGURA 51	– Parâmetros do modelo de Hammerstein e Wiener. Autoria própria.	58
FIGURA 52	– Vetor de validação. Autoria própria.	59
FIGURA 53	– Vetor de <i>setpoints</i> . Autoria própria.	60
FIGURA 54	– Resposta do automodelo para identificação. Autoria própria.	62
FIGURA 55	– Resultado do modelo Hammerstein Wiener. Autoria própria.	63
FIGURA 56	– Resposta do modelo digital para validação. Autoria própria.	63
FIGURA 57	– Resposta do automodelo para validação. Autoria própria.	64
FIGURA 58	– Resposta do modelo digital com ganhos via AG. Autoria própria.	65
FIGURA 59	– Resposta do automodelo com ganhos via AG. Autoria própria.	66
FIGURA 60	– Ação do controlador. Autoria própria.	66
FIGURA 61	– Ação individual dos termos do PID. Autoria própria.	67
FIGURA 62	– Resposta do modelo digital ao controlador PI. Autoria própria.	68
FIGURA 63	– Resposta do automodelo ao controlador PI. Autoria própria.	68
FIGURA 64	– Ação do controlador PI. Autoria própria.	69
FIGURA 65	– Ação individual dos termos PI. Autoria própria.	69

LISTA DE TABELAS

TABELA 1	– Características Automodelo de Combustão Interna	41
TABELA 2	– Resultados do Algoritmo Genético	64
TABELA 3	– Resultados do Algoritmo Genético para controlador PI	67

LISTA DE ACRÔNIMOS

A/D - Analógico para Digital

AG - Algoritmo Genético

D/A - Digital para Analógico

IDE - *Integrated Development Environment*

LED - *Light Emitting Diode*

PI - Proporcional Integral

PID - Proporcional Integral Derivativo

PMI - Ponto Morto Inferior

PMS - Ponto Morto Superior

PSO - *Particle Swarm Optimization*

PVC - *Polyvinyl Chloride*

UTFPR - Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO	11
1.1 TEMA	11
1.1.1 Delimitação do Tema	11
1.2 PROBLEMAS E PREMISSAS	14
1.3 OBJETIVOS	16
1.3.1 Objetivo Geral	16
1.3.2 Objetivos Específicos	16
1.4 JUSTIFICATIVA	17
1.5 PROCEDIMENTOS METODOLÓGICOS	17
1.6 ESTRUTURA DO TRABALHO	19
2 EMBASAMENTO TEÓRICO	20
2.1 MOTORES DE COMBUSTÃO	20
2.1.1 Motor de Combustão Interna	20
2.1.2 Componentes Básicos e Nomenclatura	21
2.1.3 Princípio de Funcionamento	23
2.1.3.1 Motor Otto Quatro Tempos	23
2.1.3.2 Motor Diesel Quatro Tempos	25
2.1.3.3 Motor Dois Tempos	26
2.2 SISTEMAS DE CONTROLE	28
2.2.1 Definições	29
2.2.2 Controle PID	30
2.2.3 Controlador Proporcional	31
2.2.4 Controlador Proporcional Integrativo	32
2.2.5 Controlador Proporcional Derivativo	32
2.2.6 Controlador Proporcional Integral Derivativo	33
2.2.7 Controle Discreto	34
2.3 ALGORITMOS GENÉTICOS	36
2.3.1 Funcionamento Simplificado	36
2.3.2 Avaliação (<i>fitness</i>)	37
2.3.3 Operadores Genéticos	38
2.3.3.1 Seleção	38
2.3.3.2 Cruzamento (<i>crossover</i>)	38
2.3.3.3 Mutação	40
3 PROCEDIMENTOS PRÁTICOS	41
3.1 MODELO DE VEÍCULO DE COMBUSTÃO INTERNA	41
3.2 PLATAFORMA DE TESTES	43
3.2.1 Primeira estrutura	43
3.2.1.1 Problemas da primeira estrutura	44
3.2.2 Segunda estrutura	44
3.2.2.1 Problemas da estrutura de madeira com esteira	46
3.2.3 Estrutura final	46

3.3	<i>ENCODER</i>	49
3.3.1	Escolha do sensor	49
3.3.1.1	Circuito e simulações	51
3.4	IDENTIFICAÇÃO DO SISTEMA VIA MODELO DE HAMMERSTEIN E WIENER	56
3.4.1	Modelos de Hammerstein-Wiener	56
3.4.2	Processo de identificação	57
3.4.3	Validação do modelo obtido	59
3.5	ALGORITMO GENÉTICO	59
3.6	CONTROLE PID	60
3.6.1	Rotina de Inicialização	60
3.6.2	Rotina PID	60
4	RESULTADOS E DISCUSSÕES	62
4.1	IDENTIFICAÇÃO DO SISTEMA	62
4.1.1	Processo de Identificação	62
4.1.2	Processo de Validação do Modelo	63
4.2	ALGORITMO GENÉTICO	64
4.3	CONTROLE	65
4.3.1	Controlador PID	65
4.3.2	Controlador PI	67
5	CONCLUSÕES	70
5.1	CONSIDERAÇÕES FINAIS E SUGESTÕES DE TRABALHO FUTURO	70
	REFERÊNCIAS	71

1 INTRODUÇÃO

1.1 TEMA

Sistemas de *Cruise Control* realizam controle longitudinal ao regular a velocidade do veículo através de uma referência dada pelo motorista ou um sistema mais complexo adiante (MORAND et al., 2015). O controle de velocidade automático do veículo é um dos tópicos mais pesquisados na indústria automobilística (KILIÇ et al., 2015).

Neste trabalho, o termo *Cruise Control* se refere apenas ao controle de velocidade longitudinal de um veículo. O objetivo da automação de velocidade é melhorar a segurança nas rodovias, reduzir as emissões, melhorar a *performance* dinâmica e conforto da jornada, utilizando um controle para substituir o motorista (KIM; YI, 2013), (QIU, 2014).

1.1.1 Delimitação do Tema

Na indústria automobilística, o controle de velocidade é um dos aspectos mais críticos que necessitam de um controlador muito bem projetado que consiga responder à não-linearidade do sistema e atender às exigências do mercado (ASERE et al., 2015). Apesar de ainda estar pouco presente no mercado brasileiro, o *Cruise Control* é utilizado há muito tempo em outros países como Japão e Estados Unidos.

O primeiro controle de velocidade moderno na indústria automobilística foi desenvolvido pelo engenheiro mecânico americano Ralph Teetor (TEETOR, 1950). No sistema desenvolvido por ele, a aceleração é controlada por um motor elétrico bidirecional acoplado no pedal do acelerador, exposto na Figura 1. Quando o motor elétrico gira em um sentido, maior será a aceleração, pois a válvula que regula a entrada da mistura ar/combustível no carburador abre. Se girar para o sentido contrário, a válvula fecha. A Figura 2 apresenta a seção transversal de um carburador, onde a mistura mencionada anteriormente acontece. A tensão que controla a direção do motor elétrico depende da leitura das rotações de um velocímetro rotacional que é conectado ao eixo de transmissão do veículo. Este sistema complexo é uma combinação eletromecânica que realiza o controle dos contatos que acionam o motor elétrico para uma determinada direção.

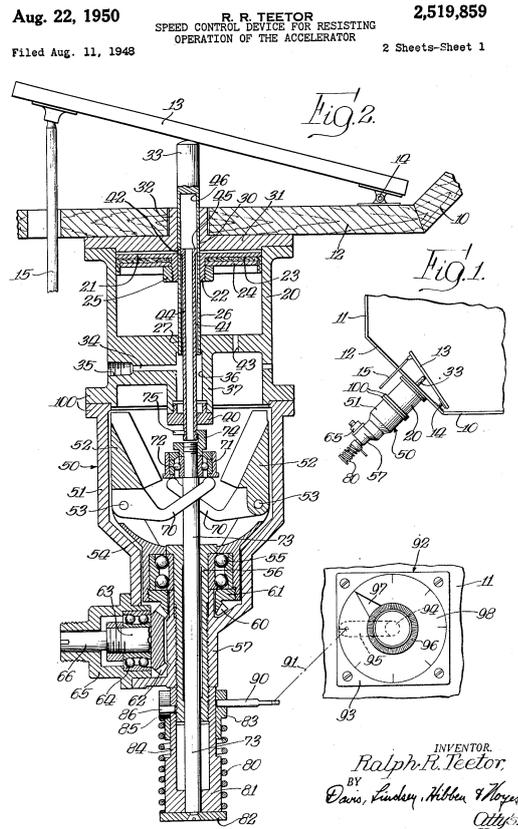


Figura 1: Operação do acelerador - Fonte: (TEETOR, 1950)

Alguns anos depois, em 1968, Daniel A. Wisner inventou o controle automotivo eletrônico que é descrito por duas patentes: US 3570622 e US 3511329. Neste sistema, a tensão correspondente a velocidade desejada é armazenada e comparada com a tensão gerada pelo velocímetro. O resultado da comparação aciona uma solenoide que controla a posição da válvula. Os esquemáticos dos circuitos digitais são apresentados na Figura 3.

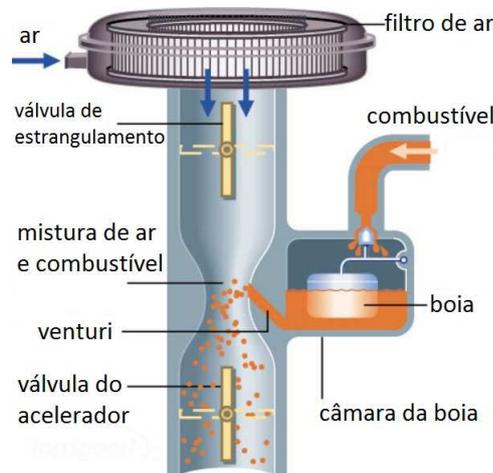
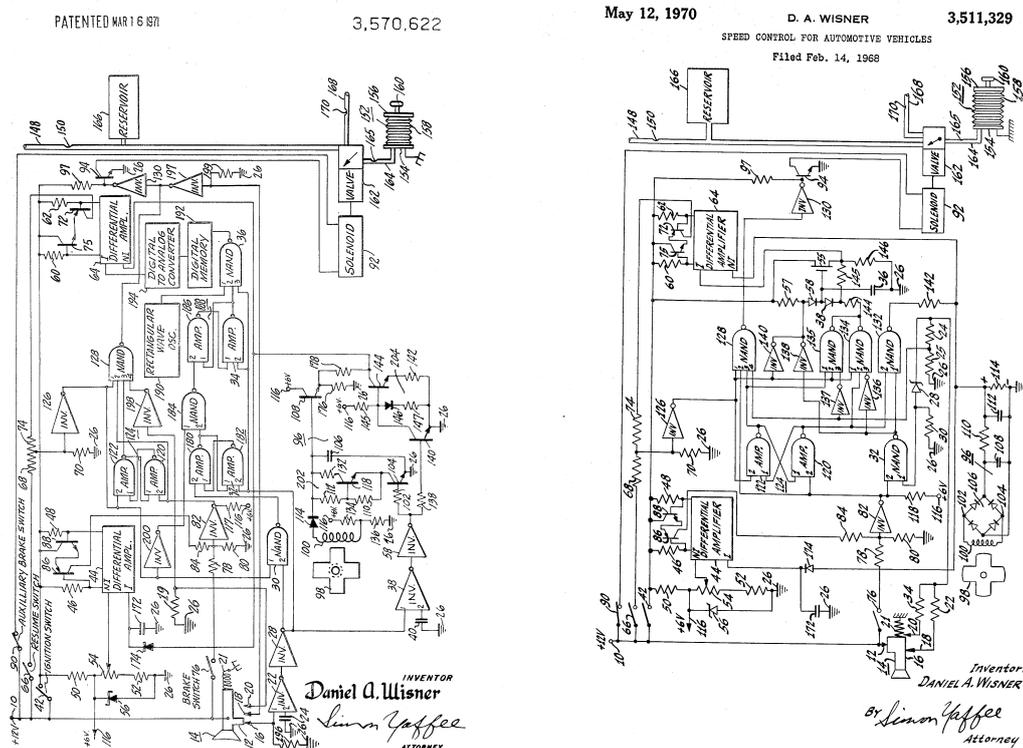


Figura 2: Seção transversal de um carburador - Adaptado de (YOURDICTIONARY, 2016)

As principais vantagens do controle eletrônico de velocidade em relação ao seu predecessor eletromecânico são: ser mais barato e possuir uma quantidade menor de partes mecânicas móveis que precisam ser constantemente lubrificadas e reparadas (WISNER, 1971).



(a) Modelo Eletrônico 1

(b) Modelo Eletrônico 2

Figura 3: Esquemáticos dos circuitos digitais que implementam o Cruise Control - Fonte: (OTHERS, 1970) e (WISNER, 1971)

A eletrônica evoluiu muito e passou a fazer a integração de informações coletadas pelos sensores em tempo real que ajudam na autonomia do automóvel e na economia de combustível. Por isso, hoje é possível aplicar técnicas mais avançadas para o controle de velocidade programando as unidades de controle eletrônico (PANANURAK et al., 2009).

Assim como a eletrônica, a área de engenharia de controle evoluiu também. Do ponto de vista de controle, o tema é estudado a partir de várias abordagens diferentes de difícil implementação. É possível abordar este assunto considerando a distância e a velocidade relativa do carro a frente, conhecido como ACC - *Adaptive Cruise Control* (WARD et al., 1999), ou, até mesmo, considerando uma comunicação constante entre os carros trafegando a mesma rodovia em exemplos como o AHS - *Automated Highway Systems* (GOTO et al., 1999) e o CACC - *Cooperative Adaptive Cruise Control* (MACNEILL; MILLER, 2003). Outros sistemas

procuram evitar colisão e manter o veículo dentro de uma faixa na pista usando câmeras e radares, além de controlar a velocidade (MORIZANE et al., 1999).

Outro ponto a ser observado é a modelagem do sistema. Em uma abordagem, o carro controlado é modelado a partir de relações da cinemática em relação ao carro da frente (ARARAT et al., 2006). Em outra, são usadas analogias com hidrodinâmica para modelar o automóvel como se fosse um fluido passando por obstáculos (GEHRIG; STEIN, 2007). Partindo de outro princípio, o peso do carro e a inclinação são levados em conta na modelagem do sistema (KIM; YI, 2013). E, por fim, pode ser aplicado o *gain scheduling*, para encontrar controladores para certos intervalos de velocidade, dada a resposta não-linear do sistema (HUO; LI, 1999).

Ao analisar os exemplo anteriores, verifica-se que a autonomia dos meios de transporte ganha cada vez mais importância na indústria automotiva. Observa-se também que quanto maior a autonomia desses, maior a dificuldade em controlá-los. O *Cruise Control* é a base de controle de automóveis e é nesse tema em que este trabalho se concentra.

1.2 PROBLEMAS E PREMISSAS

Quando um sistema físico ou processo deve ser precisamente controlado, costuma-se utilizar uma operação de malha fechada, ou *feedback*. Uma variável de saída, chamada de resposta, é ajustada conforme a necessidade pelo sinal de erro. Este é a diferença entre a resposta do sistema, tal como medida pelo sensor, e o sinal de referência, o qual representa a resposta desejada do sistema (PHILLIPS; HAVOR, 1995).

O desafio de um sistema de *Cruise Control* é manter a velocidade de saída do sistema de acordo com o sinal de entrada, determinado pelo motorista. Forças de distúrbio oriundas de sistemas externos podem gerar problemas no sistema (OSMAN et al., 2009). Dentre estas forças, destacam-se (ASERE et al., 2015):

- Força de arrasto aerodinâmico: força que será produzida pelo ar e age contra o movimento do carro. Uma representação é mostrada na Figura 4;
- Força de atrito dinâmico: força gerada devida à fricção entre as rodas e a estrada;
- Componentes gravitacionais em planos inclinados.

O motorista ativa o sistema de *Cruise Control* a uma certa velocidade, chamada de velocidade alvo, e o veículo deve seguir esta velocidade, mesmo quando confrontado por uma

rajada de vento e/ou uma estrada sinuosa (PHILLIPS; HAVOR, 1995).



Figura 4: Força de arrasto aerodinâmico - Fonte: (KO, 2016)

A resolução de problemas de controle geralmente envolve :

1. Escolher sensores para medir a saída da planta;
2. Escolher atuadores para conduzir a planta;
3. Desenvolver as equações (modelos) da planta, dos atuadores e dos sensores;
4. Projetar o controlador baseado nos modelos desenvolvidos e nos critérios de controle;
5. Avaliar o projeto analiticamente, através de simulação, e finalmente através de testes do sistema físico;
6. Se os testes físicos forem insatisfatórios, refazer os passos anteriores.

É uma tarefa muito árdua modelar um sistema físico de forma exata. Qualquer incerteza paramétrica no modelo resulta não apenas em mudanças substanciais a seu comportamento, mas também à sua estabilidade. Os métodos de *design* baseados em modelos convencionais requerem informações completas da planta e geralmente produzem um único controlador ótimo. Normalmente, ao aumentar a ordem de um sistema linear, a precisão do modelo também aumenta. Entretanto, um ponto é alcançado em que aumentar a ordem do modelo não o melhorará significativamente. Para estes casos, é necessário adicionar não-linearidades.

Alguns sistemas não-lineares podem ser estáveis para algumas entradas e então se tornarem instáveis se entradas diferentes forem aplicadas. Outros sistemas não lineares podem ser estáveis para um conjunto de condições iniciais, mas podem ser instáveis para outros. Esta é a natureza do mundo físico, já que grande parte dos sistemas físicos são não lineares (PHILLIPS; HAVOR, 1995).

Não existe uma técnica de análise de estabilidade não linear geral que sempre determinará a estabilidade deste tipo de sistema. Assim, engenheiros de controle devem tentar encontrar uma técnica que seja aplicável ao sistema sob consideração. Para vários sistemas de ordem elevada contendo várias não linearidades, a estabilidade pode ser determinada apenas através de simulações e/ou operações do sistema físico. Um sistema físico é linear apenas entre certos limites de operação e pode ser levado à não-linearidade. Além do mais, alguns sistemas físicos aparentam não ter regiões de operação linear (PHILLIPS; HABOR, 1995).

Perante ao que foi exposto nos parágrafos anteriores, espera-se que este trabalho resulte em um controlador de velocidade longitudinal (*Cruise Control*) que forneça uma resposta satisfatória independentemente da maneira com a qual as não linearidades do sistema mecânico e distúrbios do meio possam influenciar o protótipo de veículo (PHILLIPS; HABOR, 1995).

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver um sistema de controle de velocidade longitudinal (*Cruise Control*) para um protótipo de veículo automotivo, em escala, de combustão interna, visando a aplicação de conceitos de sistemas de controle de plantas e processos, contornando as dificuldades que surgem devido à folgas e não-linearidades do modelo em escala do veículo.

1.3.2 Objetivos Específicos

- Construir uma plataforma de testes para que o automodelo possa rodar sem representar perigo aos usuários e atinja velocidades de até 10 m/s.
- Determinar e validar um modelo não-linear que possa ser utilizado para determinar os parâmetros do controlador.
- Utilizar Algoritmos Genéticos para calcular os termos de um controlador proporcional, integral e derivativo de um (PID), para que este possa ser utilizado para controlar o sistema.
- Validar o controlador PID obtido e comparar os resultados práticos com os resultados simulados.

1.4 JUSTIFICATIVA

O sistema de *Cruise Control* se baseia em um sistema de controle difundido em diversas companhias automobilísticas e possui diversas formas de resolução. Este projeto aborda conceitos que passam por um projeto real de controlador, desde suas especificações até a elaboração de um protótipo funcional. Trata-se também de um sistema mecânico que possui diversas não-linearidades, tornando-o um grande desafio a ser superado. Além do mais, a elaboração de uma lógica de controle de velocidade longitudinal é um tema que pode gerar interesse de alunos de engenharia às matérias relacionadas a controle e suas aplicações práticas.

1.5 PROCEDIMENTOS METODOLÓGICOS

Após o problema que guiará este trabalho ter sido definido, e baseado nas motivações expostas, determina-se o procedimento de pesquisa a ser utilizado. Propõe-se uma pesquisa bibliográfica sobre o tema para que os principais conceitos ligados a este possam ser definidos e esclarecidos. A pesquisa abordará os seguintes tópicos:

- *Cruise Control*;
- Algoritmos Genéticos;
- Controle PID;
- Motores de combustão interna;

A busca por material ocorrerá através dos portais de artigos científicos como o Portal de Periódicos da Capes, Biblioteca Digital IEEE Xplore, Ebrary, Google Acadêmico e biblioteca central de Curitiba da Universidade Tecnológica Federal do Paraná.

Após o embasamento teórico ter sido finalizado, testes práticos com um automodelo em escala de um veículo de combustão interna, ilustrado na Figura 5, serão realizados em um dinamômetro automotivo em escala confeccionado pelos autores deste trabalho. Neste, todos os testes de aquisição de dados e avaliação de desempenho serão feitos. O modelo em escala já foi adquirido e controladores PID serão elaborados para desempenhar a função de *Cruise Control*.

Durante os testes, serão projetados controladores PID que deverão controlar a planta para que esta atinja um comportamento desejável. A Figura 6 resume os procedimentos e sua ordem.



Figura 5: Modelo em escala de veículo de combustão interna

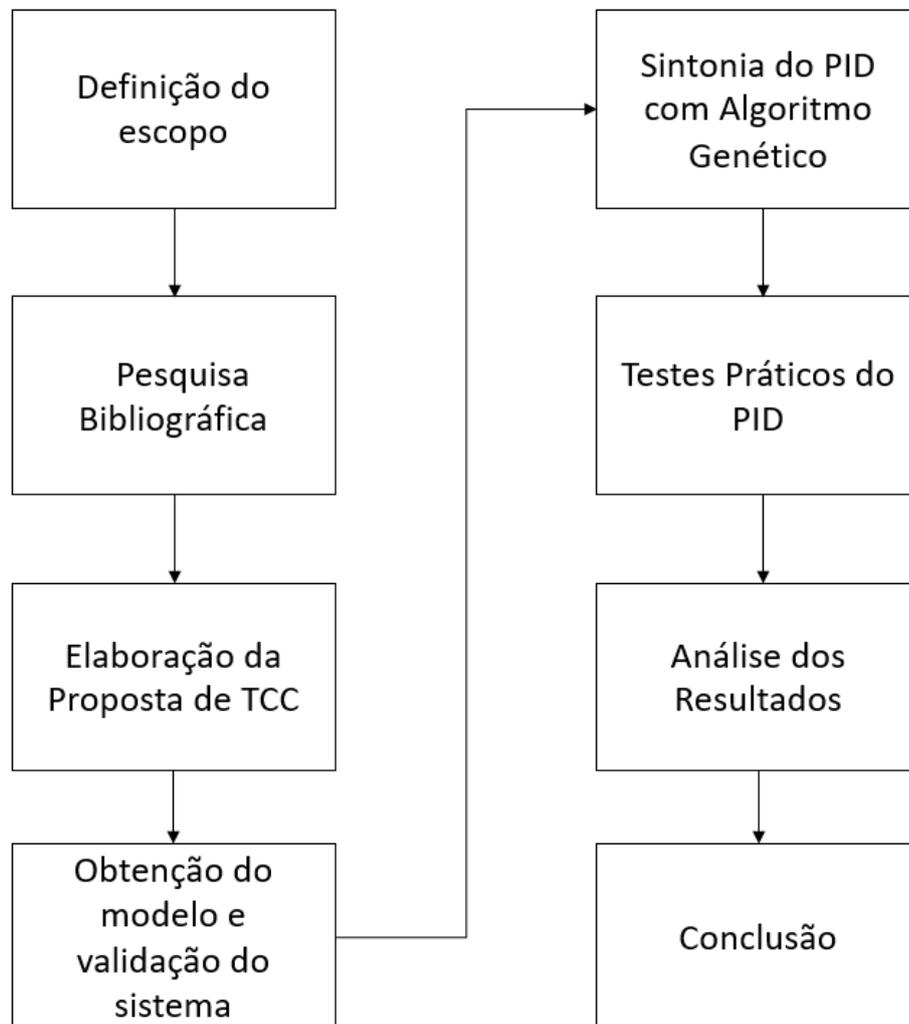


Figura 6: Fluxograma do Trabalho - Fonte:Autoria própria

1.6 ESTRUTURA DO TRABALHO

O capítulo 1, que remete à introdução deste trabalho, cumpre as metas de metodologia de pesquisa, esclarece o tema principal, Sistema de *Cruise Control*, define o problema a ser resolvido, os objetivos a serem alcançados e o princípio de pesquisa a ser seguido.

A descrição do funcionamento de motores de combustão interna será apresentada no capítulo 2. Ainda serão detalhados seus principais componentes, ciclos de trabalho, aplicações e tipos de motor, assim como suas diferenças. Neste mesmo capítulo, serão apresentados os conceitos teóricos envolvidos em sistemas de controle de malha fechada, com um foco principal em controlador PID, desde suas definições até métodos de implementação. Ainda neste capítulo, a teoria relevante à Algoritmos Genéticos será caracterizada. Também será exposto como esta teoria auxiliará o desenvolvimento dos controladores utilizados ao longo deste trabalho.

No terceiro capítulo, testes práticos serão descritos e realizados com o automodelo para desenvolver um controlador PID através de um Algoritmo Genético.

No capítulo 4, serão apresentados os resultados obtidos com o modelo em escala e também será realizada a comparação dos métodos de controle.

No capítulo final, encontram-se as conclusões sobre os resultados e abre-se espaço para dissertar sobre possíveis trabalhos futuros.

Nos complementos, encontram-se as referências bibliográficas utilizadas, apêndices elaborados e anexos necessários para o profundo entendimento deste trabalho.

2 EMBASAMENTO TEÓRICO

2.1 MOTORES DE COMBUSTÃO

Um motor pode ser definido como um dispositivo que converte um tipo de energia em outro. Muitos dos motores convertem energia térmica em trabalho mecânico, sendo assim chamados de motores térmicos. Um motor térmico transforma a energia química de um combustível em energia térmica e utiliza esta para realizar trabalho útil. Motores térmicos podem ser amplamente classificados em duas categorias: motores de combustão interna e motores de combustão externa. Estas categorias ainda podem ser subdivididas em motores rotativos e motores alternativos. Este capítulo tem como objetivo descrever e listar os componentes, princípio de funcionamento e aplicações de motores de combustão interna (GANESAN, 1995).

2.1.1 Motor de Combustão Interna

Neste tipo de motor a combustão ocorre dentro do motor, por exemplo, em motores à gasolina ou diesel, os produtos gerados pela queima de combustível e ar dentro do cilindro são responsáveis pela geração de trabalho (GANESAN, 1995).

Uma das vantagens de motores de combustão interna é o fato de todos seus componentes trabalharem em uma temperatura média bem menor do que a temperatura máxima do fluido de trabalho (mistura de combustível e ar). Isto ocorre pois a alta temperatura do fluido se mantém por apenas uma pequena fração do ciclo de trabalho. Então é possível empregar altas temperaturas no fluido de trabalho, resultando em maior eficiência térmica. Além disso, melhor eficiência térmica pode ser obtida com pressões máximas de trabalho moderadas e, portanto, a relação peso/potência é menor do que o de uma turbina à vapor. Também é mais viável a utilização deste tipo de motor para aplicações de baixa potência (GANESAN, 1995).

Algumas das desvantagens deste tipo de motor são as vibrações causadas pelo movimento de seus componentes e a restrição do tipo de combustível que pode ser empregado, no caso, apenas combustíveis líquidos ou gasosos de especificações pré-determinadas, que costumam ser mais caros. Com as considerações feitas acima, este tipo de motor pode

ser utilizado em automóveis, motocicletas, barcos, navios, aeronaves de baixa velocidade, locomotivas e geradores de energia de baixa potência (GANESAN, 1995).

2.1.2 Componentes Básicos e Nomenclatura

Embora motores de combustão interna pareçam simples, são máquinas altamente complexas, pois existem centenas de componentes que devem exercer sua função satisfatoriamente para gerar trabalho. Antes de abordar seu princípio de funcionamento, uma breve explicação de alguns componentes e medidas será apresentada. As Figuras 7 e 8 servem como auxílio para o melhor entendimento da localização destes componentes.

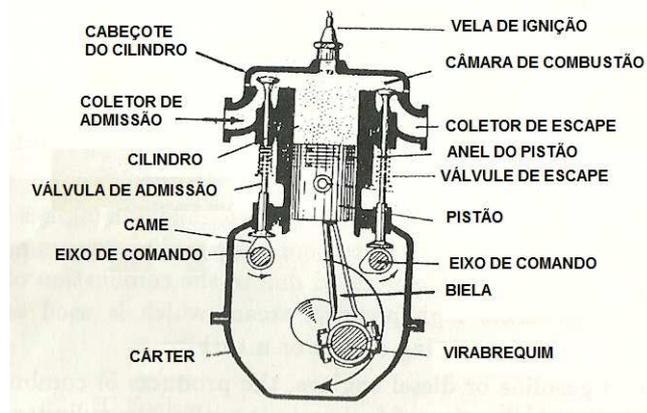


Figura 7: Corte lateral de um motor de combustão interna - Adaptado de (GANESAN, 1995)

O cilindro é um espaço cilíndrico, sustentado pelo bloco do cilindro, no qual o pistão realiza seu movimento. O volume dinâmico criado no cilindro durante o funcionamento do motor é preenchido com o fluido de trabalho. Sua medida útil é conhecida como diâmetro do cilindro, que corresponde a seu diâmetro interno (GANESAN, 1995).

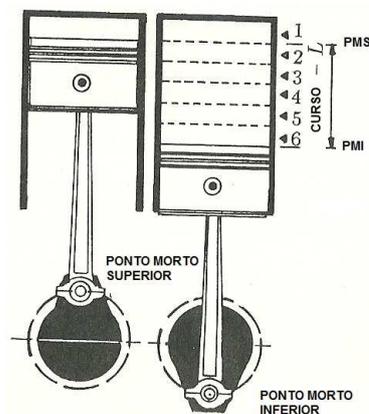


Figura 8: Ponto morto superior e inferior - Adaptado de (GANESAN, 1995)

O pistão é o primeiro componente que transmite a força da combustão para o eixo de saída. É um componente cilíndrico que se encaixa com a menor folga possível no cilindro, criando a barreira móvel do sistema de combustão. Para criar esta barreira, conta com os anéis do pistão e o lubrificante. Esta barreira possui área circular, com diâmetro igual ao do cilindro. O curso de um pistão é a distância nominal entre os dois pontos em que o pistão muda de direção. Estes pontos são conhecidos como pontos mortos, sendo eles: ponto morto superior e ponto morto inferior. O primeiro, abreviado como PMS, é o ponto em que o pistão está mais longe do virabrequim. Já o segundo, PMI, é aquele em que o pistão se encontra a menor distância do virabrequim (GANESAN, 1995).

O espaço superior compreendido entre o cabeçote do cilindro e o topo do pistão é chamado de câmara de combustão. É nesta parte que a queima do combustível e o aumento de pressão no cilindro ocorrem. Antes de ser direcionado à câmara de combustão e entrar em ignição, o combustível líquido é vaporizado com proporções devidas de ar no carburador. Alguns tipos de motores adicionam o combustível sujeito a alta pressão na câmara através de injetores (GOODSELL, 1995).

O coletor de admissão é a tubulação que conecta o sistema de admissão com a válvula de admissão do motor e por onde passa o fluido de trabalho que vai ao cilindro. Analogamente, o coletor de escape é a tubulação que conecta o sistema de escape à válvula de escape e que permite que os gases da combustão sejam retirados do motor. As válvulas de escape e admissão, como citado acima, são responsáveis pela admissão e escape de fluidos do cilindro. Sua localização varia de motor para motor, mas geralmente estão localizadas no cabeçote do cilindro ou em sua lateral (GANESAN, 1995).

Em motores de ignição por centelha como motores à gasolina ou etanol, há um componente que inicia o processo de combustão, chamado de vela de ignição, localizado no cabeçote do cilindro e constituído por eletrodos pelos quais passa alta tensão (GOODSELL, 1995).

A conexão entre o pistão e o virabrequim é feita através da biela, que é responsável por transmitir o movimento do primeiro para o segundo. A biela, por sua vez, é conectada ao pistão através do pino do pistão (GANESAN, 1995).

O virabrequim, por sua vez, converte o movimento oscilatório em um sentido do pistão em movimento circular útil para o eixo do motor. No virabrequim de motores monocilíndricos há um par de braços para a conexão entre este e a biela e pesos de balanço para aliviar as vibrações do motor. O virabrequim se encontra dentro do cárter do motor (GANESAN, 1995).

O controle da abertura e fechamento das válvulas é realizado pelo eixo de comando. Também é responsável pelo comando do sistema de ignição. O posicionamento e acionamento do eixo de comando são feitos pelo virabrequim e através de engrenagens (GOODSELL, 1995).

Os cames são partes integrais do eixo de comando, tendo sido elaborados para realizar movimentos lineares ou angulares de peças conectadas à eles. A maioria dos cames utilizados em aplicações automotivas são não-circulares, de tal maneira que realizem a abertura das válvulas no momento e duração corretos (GOODSELL, 1995).

Durante o ciclo de operação do motor, o torque líquido oriundo do virabrequim varia, causando variações de velocidade angular no eixo. Para se obter um torque líquido uniforme, adiciona-se uma massa de inércia ao eixo do motor. Esta massa é conhecida como volante do motor. Quanto maior a variação de torque, maior deverá ser o volante. A variação de torque pode ser reduzida ao aumentar o número de cilindros do motor, com isso um motor monocilíndrico possui um volante maior do que um motor com múltiplos cilindros (GANESAN, 1995).

O volume nominal percorrido pelo pistão dentro do cilindro, compreendido entre PMS e PMI é chamado de volume de deslocamento. É calculado a partir da área circular do pistão e seu curso. Outro volume cujo valor é muito importante para um motor é o volume de folga, este é o volume da câmara de combustão quando o pistão se encontra no PMS. A soma destes dois volumes é o volume total do cilindro. Dividindo este valor pelo volume de folga, tem-se a taxa de compressão (GANESAN, 1995).

2.1.3 Princípio de Funcionamento

Para um motor funcionar satisfatoriamente, este deve seguir um ciclo de operações em sequência, que não pode ser alterado. O motor de centelha (gasolina e etanol) foi inventado por Nicolaus A. Otto em 1876 e o motor de compressão (diesel) por Rudolf Diesel em 1892, sendo assim conhecidos como motor Otto e Diesel, respectivamente (GANESAN, 1995). Embora sejam muito parecidos, serão abordados em tópicos separados. Ao fim desta seção, será abordado também o princípio de funcionamento de motores de dois tempos.

2.1.3.1 Motor Otto Quatro Tempos

Neste tipo de motor, o ciclo de operação é finalizado após quatro cursos do pistão, o que é equivalente à duas voltas do virabrequim. Durante o ciclo, cinco funções são realizadas,

sendo elas: admissão, compressão, combustão, expansão e escape (GANESAN, 1995).

Quando o pistão se encontra no PMS, na iminência de se movimentar para baixo, tem início o curso de admissão. Ainda neste momento, a válvula de admissão se abre e a válvula de escape está fechada, como ilustrado na Figura 9. Devido ao movimento de descida do pistão e à abertura da válvula, o fluido de trabalho, neste caso uma mistura de ar e combustível, é sugado para dentro da câmara de combustão. Quando o cilindro chega ao PMI, a válvula de admissão se fecha e o curso de admissão é finalizado (GANESAN, 1995).

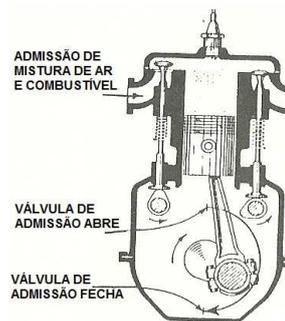


Figura 9: Curso de admissão - Adaptado de (GANESAN, 1995)

Ao começar o movimento de subida, representado na Figura 10, inicia-se o curso de compressão. Aqui as duas válvulas estão fechadas e, conforme o pistão continua a subir, a pressão interna dentro da câmara de combustão aumenta. Ao atingir novamente o PMS a mistura de ar e combustível está comprimida no volume de folga, a taxa de compressão para este tipo de motor varia entre seis e dez. Neste momento, o eixo de comando faz com que a vela de ignição solte uma faísca que inicia a ignição do fluido de trabalho. Devido à alta pressão, a queima é quase instantânea (GANESAN, 1995).

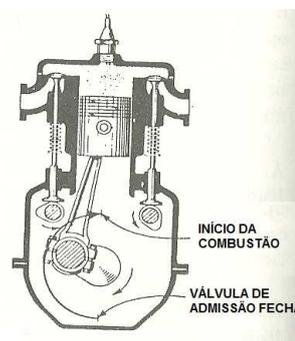


Figura 10: Curso de compressão - Adaptado de (GANESAN, 1995)

A alta pressão da queima dos gases empurra o pistão novamente para baixo, o que caracteriza o curso de expansão, ilustrado na Figura 11. Este curso também pode ser chamado de curso de trabalho, já que, como poderá ser concluído após o entendimento dos quatro cursos, é o único que efetivamente transforma a energia da ignição em movimento (GANESAN, 1995).

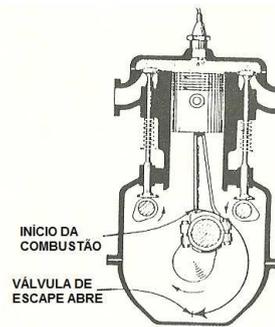


Figura 11: Curso de expansão - Adaptado de (GANESAN, 1995)

Ao final do curso de expansão, o pistão se encontra no PMI e a válvula de escape é aberta, enquanto a de admissão continua fechada. Assim se inicia o curso de escape, ilustrado na Figura 12. Nele, o movimento do pistão para cima empurra os gases residuais da queima da mistura de ar e combustível para fora da câmara de combustão através da válvula de escape. Ao atingir novamente o PMS, a válvula de escape se fecha, com isso é possível iniciar novamente o ciclo de trabalho (GANESAN, 1995).

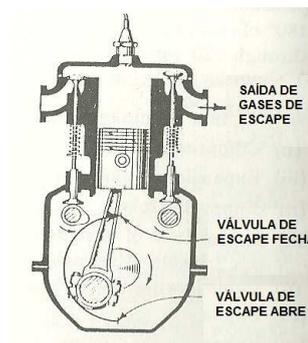


Figura 12: Curso de escape - Adaptado de (GANESAN, 1995)

2.1.3.2 Motor Diesel Quatro Tempos

Motores Diesel, também conhecidos como motores de compressão, são muito similares aos motores Otto, mas operam em taxas de compressão muito maiores, variando entre 16 e 20. Outra diferença reside na entrada de fluido de trabalho, aqui apenas ar entra pela válvula de admissão e o combustível é adicionado à câmara de combustão através de uma bomba de alta pressão e um injetor. Devido à taxa de compressão ser relativamente maior, a pressão interna na câmara também é mais elevada e, com isso, o combustível inicia sua combustão sem a necessidade de uma vela de ignição (GANESAN, 1995).

Durante o curso de admissão, somente ar é induzido à câmara de combustão através da válvula de admissão, enquanto que a válvula de escape se mantém fechada. O ar presente

na câmara é comprimido no volume de folga enquanto as duas válvulas se encontram fechadas, caracterizando o curso de compressão. Momentos antes do final deste curso, combustível é injetado na câmara. Esta mistura entra em combustão espontaneamente devido à alta pressão e à variação de temperatura resultante da redução de volume. A combustão empurra o pistão do PMS para o PMI, realizando trabalho e finalizando o curso de expansão. Ao chegar no PMI, a válvula de escape se abre. Com esta válvula aberta e o pistão se movendo para cima, os gases resultantes da combustão são retirados do motor (GANESAN, 1995). A Figura 13 representa cada um destes cursos.

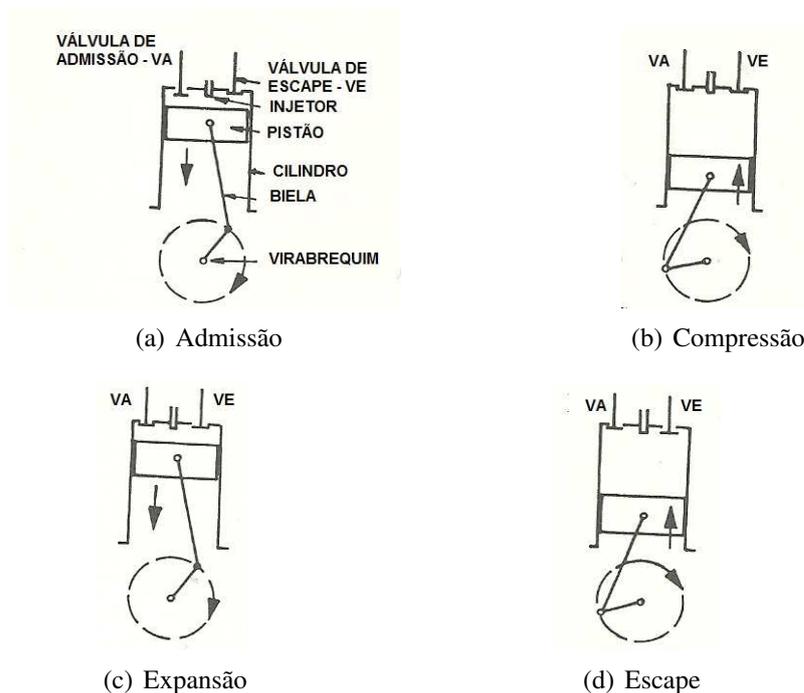


Figura 13: Ciclos de trabalho de motor de compressão - Adaptado de (GANESAN, 1995)

Como a pressão interna durante o ciclo de combustão deste tipo de motor é muito maior do que a de motores Otto, os componentes que o compõem devem ser mais resistentes. Com isso, para uma mesma potência, motores de compressão são mais pesados do que suas versões à centelha, resultando também em operações com menores velocidades (GANESAN, 1995).

2.1.3.3 Motor Dois Tempos

Como já citado anteriormente, motores quatro tempos possuem apenas um curso de trabalho para cada duas voltas do virabrequim. Caso os cursos de admissão e escape pudessem ser realizados de forma independente do movimento do pistão, haveria um curso de trabalho para cada volta do virabrequim. Com esse arranjo, teoricamente, a potência de saída seria o

dobro para uma mesma velocidade. Assim, Dugald Clark inventou o motor dois tempos em 1878 (GANESAN, 1995).

A maior diferença entre este motor e os de quatro tempos está na forma em que fluido de trabalho é admitido à câmara e os gases de escape são retirados. O fluido de trabalho é inserido no cárter através da porta de admissão quando a pressão do cárter se reduz devido ao movimento de subida do pistão. Assim, compressão, ignição e expansão ocorrem da mesma forma que os modelos anteriores. Durante a expansão, o fluido no cárter é comprimido e, próximo ao fim deste curso, o pistão libera a porta de escape e a pressão no cilindro reduz conforme os gases de escape são liberados. Ao continuar seu movimento de descida, o pistão libera o fluxo das portas de transferência, permitindo que o fluido do cárter sejam direcionados à câmara de combustão. No topo do pistão geralmente se encontra um defletor que direciona o fluido que vem do cárter ao topo da câmara, isso garante que este fluido possa ser utilizado para o próximo ciclo e não fuja diretamente pela porta de escape (GANESAN, 1995). Todos estes componentes e características são apresentados na Figura 14.

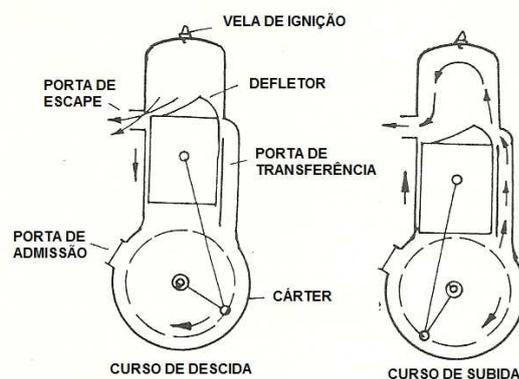


Figura 14: Ciclos de motor dois tempos - Adaptado de (GANESAN, 1995)

Além de obter maior potência de saída que motores quatro tempos, o mecanismo do motor garante maior simplicidade mecânica ao eliminar o arranjo de válvulas. A grande maioria não possui válvulas e quando possuem, trata-se somente da válvula de escape. Isto torna este motor mais barato, com manutenção mais simples e com menor tamanho para uma mesma potência (GANESAN, 1995).

Seu funcionamento simples, porém, traz algumas desvantagens. Durante pequenos intervalos de tempo, ambas as portas de admissão e escape estão simultaneamente abertas, o que pode fazer com que o fluido recém armazenado no cárter saia diretamente, levando a um maior consumo de combustível. Quando a aceleração não se dá em toda sua magnitude, parte dos gases de escape não saem da câmara de combustão, fazendo com que o motor funcione de maneira irregular. E, finalmente, é necessário maior resfriamento e lubrificação do sistema

(GANESAN, 1995).

Um tipo específico de motores dois tempo é o motor *Glow* ou Nitro, muito utilizado em modelos de rádio controle de combustão interna, assim como o modelo usado neste trabalho, que será detalhado na explanação das atividades práticas. Seu princípio de funcionamento é muito similar aos motores de centelha, mas neste caso não é utilizada uma vela de ignição, e sim um aquecedor elétrico. Para dar partida é necessário energizar o aquecedor, que aumenta a temperatura da câmara de combustão até o ponto em que o combustível entra em ignição. A partir deste ponto, o próprio calor gerado pela explosão é suficiente para manter a ponta do aquecedor na temperatura de funcionamento, podendo assim retirar a fonte de energia elétrica. O combustível utilizado é uma mistura de metanol, lubrificantes e nitrometanol, originando assim o nome Nitro.

2.2 SISTEMAS DE CONTROLE

Um sistema de controle consiste de componentes interconectados e é projetado para realizar um determinado objetivo, por exemplo, o automóvel pode ser considerado um sistema de locomoção, no qual o motor é um de seus componentes. Já que é muito difícil isolar completamente algo da natureza de todo o resto, a seleção dos limites de um sistema está atrelada ao propósito e limitações do sistema (PALM, 1988).

A operação bem sucedida de um sistema sob condições em constante mudança geralmente requer um sistema de controle. Por exemplo, o sistema de aquecimento de um prédio necessita de um termostato que ligará ou desligará os aquecedores conforme a temperatura aumenta ou diminui. A mudança de temperatura externa é a principal responsável pela variação da temperatura interna. Se as condições climáticas (temperatura, insolação, vento) pudessem ser equacionadas com precisão, seria possível desenvolver um aquecedor que trabalhasse em uma rotina que compensasse qualquer variação. Como não se consegue equacionar a dinâmica climática, a saída do aquecedor deve ser ajustada de acordo com a temperatura atual do ambiente. Ainda neste exemplo, se a temperatura desejada for alterada para um outro valor, o termostato deve agir de tal forma que este valor seja atingido e se mantenha até que outro seja escolhido (PALM, 1988).

O termo controle refere-se ao processo de influenciar deliberadamente o comportamento de um objeto para produzir um resultado desejado e o dispositivo responsável por isso é o controlador. O piloto automático de aeronaves para manter altitude, orientação e velocidade, o sistema de *Cruise Control* de automóveis e o regulador de pressão de sistemas de

fornecimento de água são exemplos de sistemas de controle (PALM, 1988).

O controlador deve responder satisfatoriamente a mudanças da entrada e manter a *performance* do sistema, mesmo com distúrbios. Em sistemas dinâmicos complexos, um ou mais controladores são normalmente necessários para que os elementos ajam em conjunto para alcançar o objetivo desejado (PALM, 1988).

Estratégias de projeto de controle são necessárias para melhorar processos de manufatura, uso energético, controle de automóveis, dentre outros. Geralmente, a intuição, gerada durante experimentos com o sistema físico, representa uma importante parte do processo de elaboração. A relação entre conhecimento e controle é de suma importância, visto que o controle efetivo de sistemas requer a compreensão e modelagem dos sistemas. À interconexão de componentes que forma um sistema com resposta desejada é dado o nome de sistema de controle (DORF RICHARD C E BISHOP, 1998).

2.2.1 Definições

Um sistema de controle de malha fechada é definido como um em que as entradas do conjunto controlador mais planta são determinadas, ao menos de forma parcial, pelas saídas. A planta de um sistema de controle é o sistema físico ou processo a ser controlado (PHILLIPS; HABOR, 1995).

Para controlar qualquer variável física é necessário saber seu valor. O sensor é a instrumentação física que mede o sinal de saída e converte esta medida em um sinal útil para o controle (PHILLIPS; HABOR, 1995).

Em muitos sistemas de malha fechada, é necessário conectar um terceiro sistema à malha para obter características satisfatórias para o sistema como um todo. Esse sistema adicional é chamado de compensador ou controlador (PHILLIPS; HABOR, 1995).

A entrada do sistema, ou *setpoint*, é o sinal de referência. A saída do sistema é medida pelo sensor e comparada com a entrada. O sinal da diferença é chamado de erro. Se a saída for igual à entrada, o erro é nulo e a saída da planta permanece em seu valor atual. Se o erro não for nulo, em um sistema devidamente projetado, o sinal de erro causará uma resposta da planta tal que a magnitude do erro será reduzida (PHILLIPS; HABOR, 1995). O sistema como um todo pode ser visto na Figura 15.

Um sistema estável responderá de uma maneira apropriada à uma entrada aplicada. Em um sistema instável existe pouca relação aparente entre a entrada e a saída do sistema. Enquanto

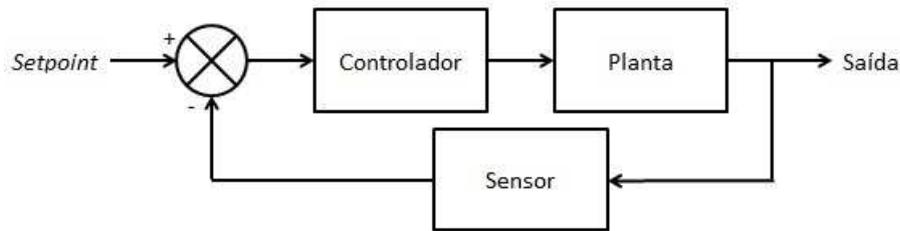


Figura 15: Sistema de controle e seus componentes - Autoria própria

há uma dificuldade considerável em determinar a estabilidade de um sistema físico para todas as condições, a questão da estabilidade não é difícil para sistemas lineares e invariantes no tempo. O tópico geral da mudança de características do sistema com variações de parâmetros do sistema é chamado de sensibilidade (PHILLIPS; HAVOR, 1995).

Geralmente, um controlador é requisitado para filtrar o sinal de erro para que certos critérios, ou especificações, de controle sejam satisfeitos. Esses critérios podem envolver, mas não se limitar a:

- Rejeição de distúrbios;
- Erros de estado estacionário;
- Características de resposta transitória;
- Sensibilidade a mudanças de parâmetros na planta.

Quando a modelagem ou estudo de um sistema se torna muito onerosa, costuma-se tratá-lo como uma caixa preta. Nessa hipótese, a análise do sistema é focada em como as conexões entre seus elementos influencia o comportamento total do sistema. A relação entrada-saída expressa a função causa-efeito da caixa preta (PALM, 1988). Neste trabalho a caixa preta em questão é o automodelo.

2.2.2 Controle PID

Mais da metade dos controladores industriais em uso atualmente emprega sistemas de controle PID. O uso deste tipo de controlador se deve ao fato de poder ser aplicado à maioria dos sistemas de controle, em particular nos quais o modelo matemático da planta não é conhecido (como o automodelo em questão), já que métodos de projeto analítico não podem ser utilizados (OGATA, 2013). Para implementar este controlador, deve-se determinar três parâmetros: ganho proporcional, ganho integral e ganho derivativo (DORF RICHARD C E BISHOP, 1998).

Frequentemente não é necessário implementar os três termos para se adequar às especificações de um sistema de controle em particular. A seguir serão abordadas as características de cada um destes termos e seu conjunto, supondo uma entrada $e(t)$ e saída $u(t)$. A planta utilizada para a elaboração das figuras que serão expostas é dada pela equação (1), no domínio da frequência.

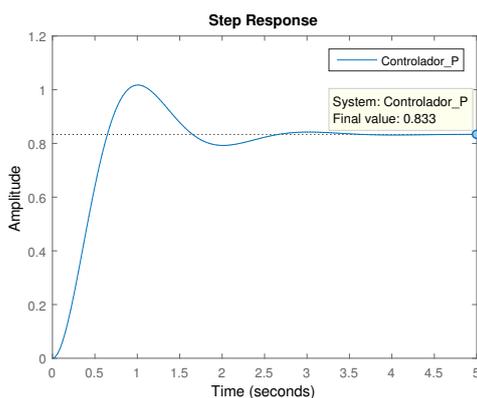
$$G(s) = \frac{1}{(s+1)(s+2)}. \quad (1)$$

2.2.3 Controlador Proporcional

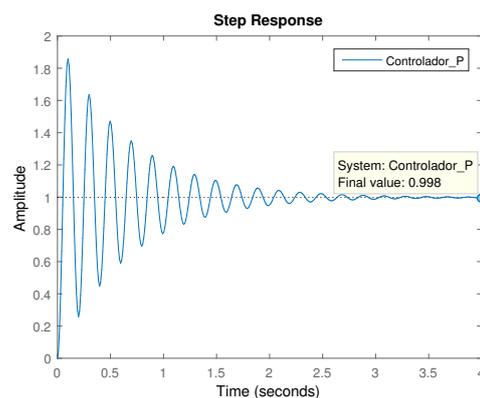
É utilizado em situações em que respostas satisfatórias de estados transitório e estacionário podem ser obtidas apenas ao ajustar o ganho do sistema (PHILLIPS; HAVOR, 1995). Assim, um controlador que atue proporcionalmente à diferença entre o valor atual da saída e o valor desejado é chamado de controlador proporcional e a equação (2) representa seu comportamento no domínio do tempo, com K_P sendo o ganho proporcional (PALM, 1988)

$$u(t) = K_P e(t). \quad (2)$$

Em alguns sistemas, ao se aumentar o ganho em excesso, a planta poderá oscilar e se tornar instável (PALM, 1988). A Figura 16 ilustra a oscilação ao se elevar o ganho.



(a) Ganho proporcional de valor 10



(b) Ganho proporcional de valor 500

Figura 16: Comparação de respostas ao degrau unitário de Controlador P - Autoria própria

2.2.4 Controlador Proporcional Integrativo

Sua equação no domínio do tempo é apresentada abaixo, com K_P representando o ganho proporcional e K_I o ganho integral (PALM, 1988)

$$u(t) = K_P e(t) + K_I \int e(t) dt. \quad (3)$$

Sua principal função é fazer com que processos sigam um sinal de referência com erro nulo no estado estacionário. Porém, ganhos elevados do termo integral podem levar o sistema à instabilidade (PHILLIPS; HAVOR, 1995), como mostrado na figura a seguir:

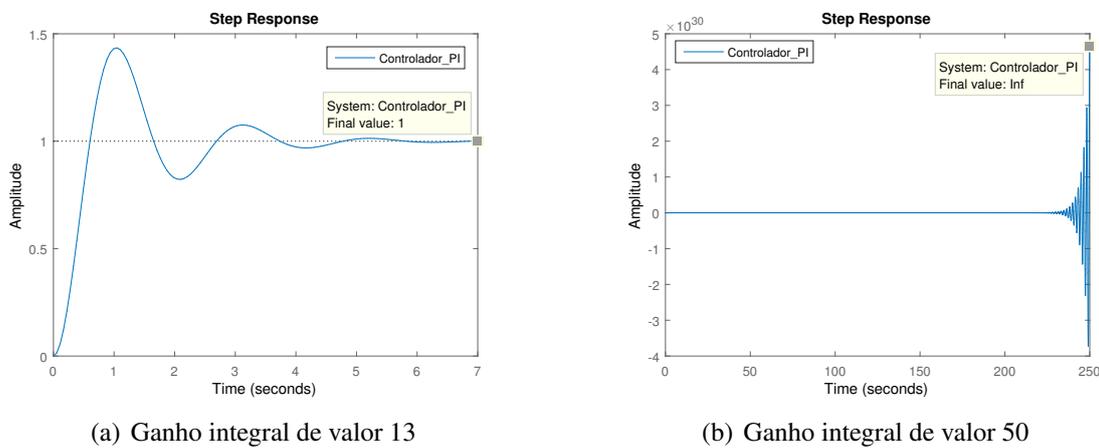


Figura 17: Comparação de respostas ao degrau unitário de Controlador PI - Autoria própria

2.2.5 Controlador Proporcional Derivativo

A equação (4) representa o comportamento no domínio do tempo deste controlador, com K_P representando o ganho proporcional e K_D o ganho derivativo (PALM, 1988)

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt}. \quad (4)$$

A inércia inerente da saída de um processo em relação às alterações na entrada faz com que mudanças sejam observadas apenas após um intervalo de tempo. A ação do termo derivativo, em conjunto com a ação proporcional, melhora a resposta transitória do sistema (PHILLIPS; HAVOR, 1995). Isto pode ser observado ao se comparar as Figuras 16(a) e 18(a). Além disso, ao se aumentar demasiadamente o ganho derivativo, a resposta da planta se torna mais lenta, como ilustra a Figura 18(b).

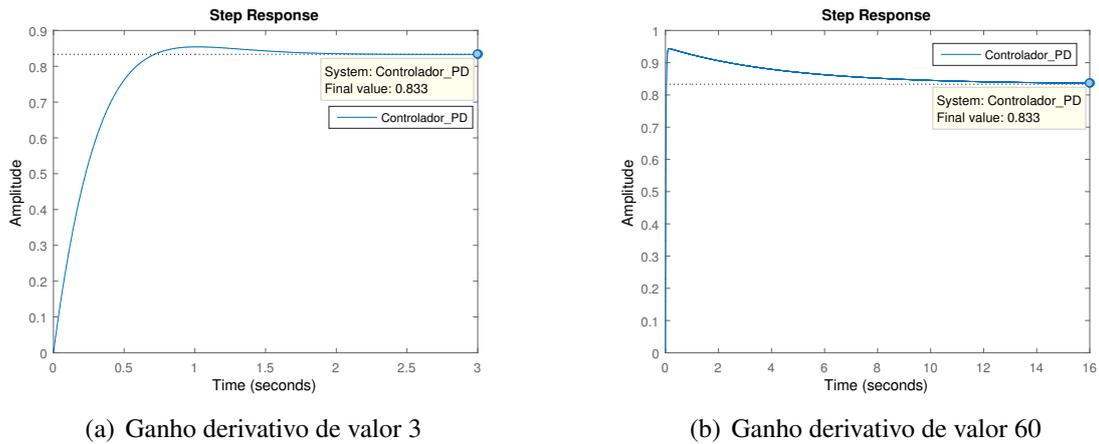


Figura 18: Comparação de respostas ao degrau unitário de Controlador PD - Autoria própria

2.2.6 Controlador Proporcional Integral Derivativo

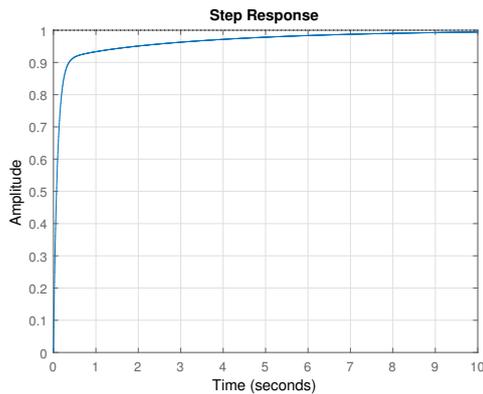
O controlador PID é provavelmente o mais comumente utilizado em sistemas de controle via *feedback*. É empregado em sistemas de controle em que melhorias em ambas as respostas transitórias e de estado estacionário são necessárias (PHILLIPS; HABOR, 1995). Assim, este controlador possui erro nulo em regime permanente oriundo do controle PI e uma resposta transitória mais rápida devido à componente derivativa. O controlador PID é definido pela equação (PHILLIPS; HABOR, 1995)

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}, \quad (5)$$

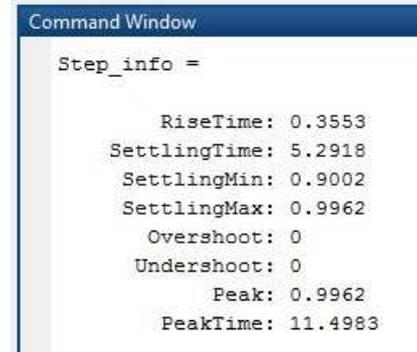
e sua transformada de Laplace gera a função de transferência, representada pela equação (6), com K_P representando o ganho proporcional, K_I o integral e K_D o derivativo (PHILLIPS; HABOR, 1995).

$$C(s) = \frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s. \quad (6)$$

Suponha que para a planta expressa pela equação (1), deseja-se erro de estado estacionário nulo, tempo de subida menor que 1 segundo, tempo de estabilização menor do que 3 segundos e percentual de *overshoot* menor do que 10%. Ao se ajustar os ganhos proporcional, integral e derivativo, respectivamente, para os valores $K_P = 20$, $K_I = 5$, $K_D = 10$, o sistema pode ser controlado satisfatoriamente, como ilustra a Figura 19.



(a) Resposta ao degrau para sistema com controlador PID



(b) Informações sobre resposta ao degrau

Figura 19: Resposta ao degrau para sistema com controlador PID - Autoria própria

2.2.7 Controle Discreto

O uso amplo de computadores teve um profundo impacto na prática de engenharia e educação. O fardo atrelado à cálculos foi reduzido drasticamente e, com isso, é possível utilizar diversos métodos numéricos. Dispositivos digitais lidam com relações e operações matemáticas apenas quando estas são expressas em conjuntos finitos de números, então qualquer medição contínua deve ser convertida em uma série de pulsos através de amostragem (PALM, 1988).

Este processo mede variáveis contínuas no tempo em instantes de tempo distintos e separados. Os valores medidos são então arredondados a um valor determinado em um intervalo finito de níveis que depende das características do dispositivo. Este processo é conhecido como quantização (PALM, 1988). A Figura 20 ilustra um exemplo de conversão de sinal analógico para digital.

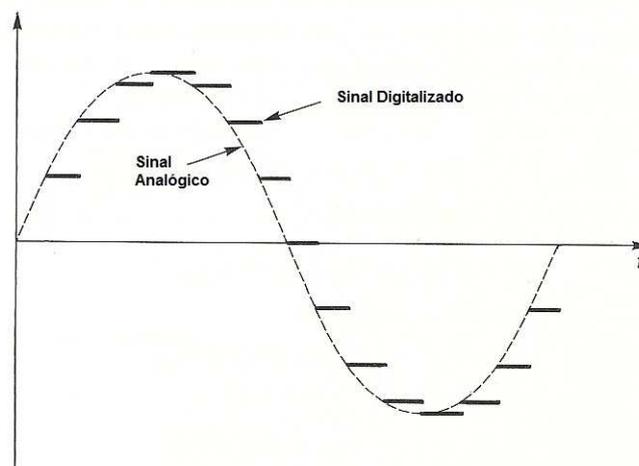


Figura 20: Conversão de sinal analógico para digital - Adaptado de (PHILLIPS; HAVOR, 1995)

Existem diversas vantagens em se utilizar computadores digitais em sistemas de controle. Algoritmos complexos podem ser implementados facilmente, por serem feitos em *software*. Assim, a dificuldade que existe na implementação de leis de controle através de componentes eletrônicos, pneumáticos e hidráulicos deixa de ocorrer. Como o *software* pode ser facilmente criado, também pode ser modificado para alterar o algoritmo ou os valores dos ganhos. Os computadores também são úteis para armazenar registros de consumo de energia, tempo ocioso, entre outros, para questões administrativas (PALM, 1988).

A estrutura básica de um controlador digital de malha fechada é mostrada na Figura 21. O computador com seu temporizador interno comanda os conversores A/D e D/A, também compara o sinal de comando com o sinal realimentado e gera o sinal de controle que será enviado aos elementos. O cálculo deste sinal é feito no algoritmo de controle armazenado na memória (PALM, 1988).

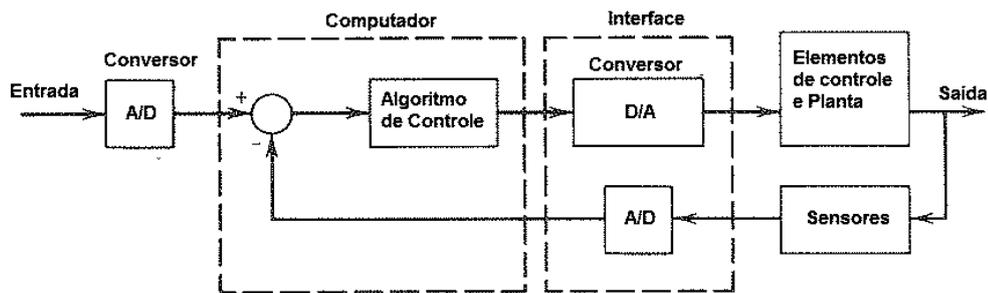


Figura 21: Estrutura de um sistema de controle digital - Adaptado de (PALM, 1988)

Controladores PID digitais também satisfazem a equação (5), exceto que a multiplicação, integração e diferenciação são realizadas numericamente pelo computador. Se forem calculadas com precisão, haverá pouca diferença na resposta do sistema com controle analógico e com controle digital (PHILLIPS; HAVOR, 1995).

Supondo um período de amostragem T , sinal de saída y , sinal de erro e e utilizando a regra de Euler com atraso para integração e diferenciação, a equação de diferenças de um controlador PID digital pode ser expressa por (PHILLIPS; HAVOR, 1995):

$$y[k] = K_p e[k] + K_I (y[k-1] + T e[k]) + \frac{K_D (e[k] - e[k-1])}{T}. \quad (7)$$

2.3 ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AG's) se baseiam na teoria Darwiana de seleção natural. São métodos de busca e otimização generalizados que simulam processos naturais da evolução, para resolução de problemas reais (ALVES et al., 2002).

A busca por soluções de um Algoritmo Genético (AG) se diferencia de métodos exatos, principalmente porque trabalha a partir de uma população de soluções, e não apenas com uma solução, utiliza regras de transição probabilísticas e informações de custo, mas não de derivadas (ALVES et al., 2002).

Alguns critérios básicos precisam ser levados em consideração ao aplicar este algoritmo para a solução de um problema. Os critérios abrangem aspectos como codificação, função de avaliação (*fitness*), operadores genéticos (*crossover*, seleção, mutação), taxas probabilísticas de realização dos operadores genéticos, critérios de parada e convergência (ALVES et al., 2002).

Na seção 2.3.1, há uma breve explicação de como o algoritmo funciona. Na seção 2.3.2 em diante, alguns aspectos teóricos dos operadores genéticos utilizados neste trabalho são explicados.

2.3.1 Funcionamento Simplificado

O algoritmo é inicializado com um conjunto de possíveis soluções. Este conjunto é identificado como população. Cada indivíduo da população é conhecido como cromossomo e representa uma possível solução para o problema (SILVA et al., 2011).

Cada indivíduo pode sofrer mutações de acordo com o princípio da seleção natural. No caso do AG, as mutações podem ser feitas com um operador que insere mudanças aleatórias nos genes dos cromossomos.

A importância da mutação está em recuperar um gene perdido durante a seleção ou simplesmente proporcionar o aparecimento de um gene que não foi gerado na população inicial. Dentro deste contexto, a taxa de mutação deve ser limitada em um percentual do número total de genes de uma determinada população para controlar a taxa com que novos genes são introduzidos na população (GEN; CHENG, 2000).

O valor limite para a taxa mutação não é definido, sendo necessário testar para cada problema. Uma elevada taxa de mutação pode interferir na herança genética, fazendo com que

os indivíduos descendentes percam a semelhança com seus ascendentes. Desta forma, o AG não teria capacidade de evolução a partir do histórico das gerações passadas. Porém, com uma taxa de mutação muito baixa, alguns genes importantes não seriam testados (SILVA et al., 2011).

Do ponto de vista de convergência do processo de otimização, a operação de mutação visa impedir que o algoritmo possa convergir a soluções que representam mínimos locais, que não são as melhores soluções para o sistema. Um processo típico de AG pode ser representado pela Figura 22 (SILVA et al., 2011). Na Figura 22, g é o número da geração em teste e $P(g)$ é a população de cromossomos da geração g .

1 - $g \leftarrow 0$	▷ Primeira geração.
2 - Inicializar $P(g)$	▷ Cromossomos recebem genes aleatórios.
3 - Enquanto critério de parada não encontrado, fazer:	
4 - $P'(g) \leftarrow$ Seleção	
5 - $P'(g) \leftarrow$ Cruzamento	
6 - $P'(g) \leftarrow$ Mutação	
7 - Avaliar $P'(g)$	▷ Atualiza o <i>fitness</i> dos cromossomos.
8 - $P'(g + 1) \leftarrow$ Sobreviventes de $P(g), P'(g)$	

Figura 22: Pseudo-algoritmo genético. Autoria própria.

A primeira geração é inicializada com genes aleatórios. Neste trabalho, cada gene representa um ganho do controlador PID. Logo, o cromossomo possui três genes que representam seus ganhos e um parâmetro que representa seu *fitness*.

O algoritmo inicializa um laço, a cada iteração (geração) uma condição de parada é verificada. Esta, envolve o *fitness*, explicado em 2.3.2, e o número máximo de gerações. Se o *fitness* desejado não é obtido, o algoritmo para quando chega ao número máximo de gerações.

Após a verificação do critério, os operadores genéticos (seleção, cruzamento e mutação) são aplicados em sequência. Estes operadores são detalhados em 2.3.3.

Depois da mutação, o *fitness* dos cromossomos é atualizado, o laço recomeça e o critério de parada é verificado com a população sobrevivente.

2.3.2 Avaliação (*fitness*)

A função de avaliação tem o objetivo de medir a capacidade do indivíduo de se adaptar ou resolver o problema (SILVA et al., 2011). Neste trabalho, a avaliação é representada pelo erro quadrático médio entre a saída da planta (velocidade do protótipo) e o *setpoint* desejado.

O erro quadrático médio de um cromossomo pode ser visto na equação (8), Sendo N o número máximo de pontos medidos na saída y da planta.

$$Fitness_{cromossomo} = \frac{1}{N} \sum_{i=1}^N (y_i - set\ point)^2. \quad (8)$$

A avaliação da equação (8) é importante para comparar cromossomos ao longo do algoritmo, como na seleção, ou na verificação do critério de parada. Ou seja, quanto menor o erro, melhor o cromossomo. Neste caso, o *fitness* é inversamente proporcional ao erro quadrático médio. No capítulo 3, a implementação da avaliação na prática é explicada com mais detalhes.

2.3.3 Operadores Genéticos

Como mencionado na seção 2.3.1, os operadores genéticos utilizados neste trabalho foram: seleção, cruzamento e mutação. Esta seção será focada nos operadores genéticos, destacados nas linhas quatro, cinco e seis da Figura 22 .

2.3.3.1 Seleção

Após a população ser inicializada e avaliada, ocorre um processo denominado de seleção. Existem vários métodos de seleção (JEBARI; MADIIFI, 2013). Neste trabalho, o objetivo da seleção é escolher um cromossomo (pai) que realizará o cruzamento com um outro cromossomo (mãe). A operação de cruzamento é explicada em 2.3.3.2.

O cromossomo pai é selecionado por torneio que basicamente escolhe cromossomos aleatoriamente e os compara para selecionar o que possui o menor *fitness*. Já o cromossomo mãe pode ser qualquer cromossomo da população. Por exemplo, se existe uma população de dez indivíduos, pode-se escolher aleatoriamente três para o torneio, como demonstrado na Figura 23.

2.3.3.2 Cruzamento (*crossover*)

Após a seleção por torneio, o cruzamento é realizado para inserir variações nos genes da população misturando os genes entre dois cromossomos. Existem várias formas de realizar o cruzamento (MAGALHAES-MENDES, 2013). Neste trabalho, o cruzamento é realizado gene por gene a partir de uma taxa que varia ao longo das gerações.

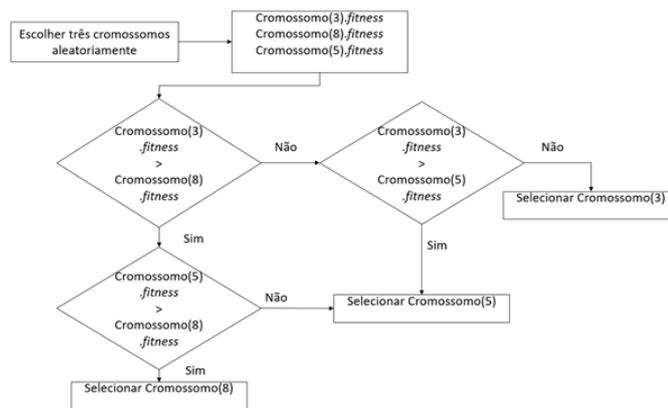


Figura 23: Seleção. Autoria própria.

Um número aleatório (τ_1) é gerado e comparado a esta taxa. Se a taxa for menor que este número, o cruzamento não ocorre. Porém, se for maior ou igual, outro número aleatório (τ_2) é gerado.

Compara-se este número, τ_2 , à média de sua distribuição normal. Se ele for menor que a média, o gene do cromossomo da mãe é atribuído ao gene do cromossomo descendente. Do contrário, o gene do cromossomo do pai é selecionado. Na Figura 24, pode-se observar um exemplo de cruzamento.

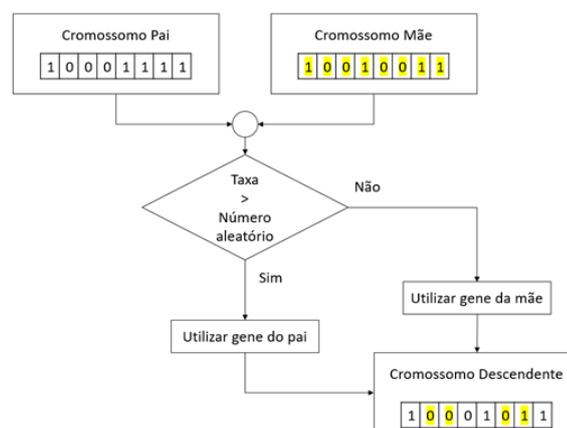


Figura 24: Cruzamento. Autoria própria.

Cada gene do cromossomo descendente vem de um dos cromossomos ascendentes. O processo se repete para todos os cromossomos da população. Os descendentes são salvos para a operação de mutação.

2.3.3.3 Mutaç o

O operador de muta o aumenta a diversidade de cromossomos alterando seus genes aleatoriamente. Existem m todos diferentes para realizar a muta o (SHARAPOV, 2007). Neste trabalho, similar ao cruzamento, compara-se um n mero gerado aleatoriamente   uma taxa de muta o da gera o em teste. Se a taxa de muta o for maior que o n mero gerado, ocorre uma muta o no gene. Esse opera o   repetida para todos os cromossomos da popula o.

No exemplo da Figura 25, os genes representam tr s ganhos de um controlador PID. Ao ocorrer a muta o, o gene (um dos ganhos do controlador) recebe um valor real aleat rio. Como pode ser observado, os genes destacados na Figura 25 sofreram muta o.

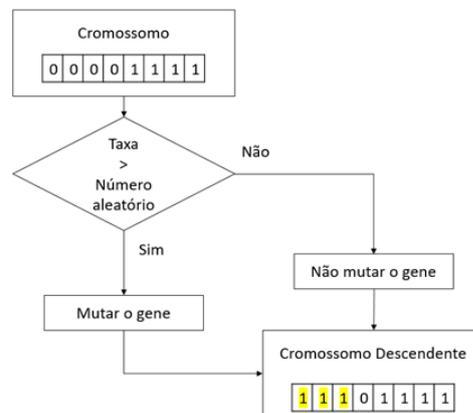


Figura 25: Muta o. Autoria pr pria.

Ao realizar esse procedimento para toda a popula o, todos os cromossomos s o avaliados novamente. O melhor cromossomo da popula o   comparado com o resultado desejado, que varia de acordo com o problema a ser resolvido. Os crit rios de parada utilizados para este trabalho s o discutidos no cap tulo 3.

3 PROCEDIMENTOS PRÁTICOS

3.1 MODELO DE VEÍCULO DE COMBUSTÃO INTERNA

A Tabela 1 fornece as principais características do automodelo de combustão interna utilizado neste trabalho.

Tabela 1: Características Automodelo de Combustão Interna

Marca	EXCEED-RC
Modelo	Forza .18 Nitro
Motor	Tipo 18 / 3 ml
Potência	1,1 hp (820 W)
Tração	4x4
Comprimento	400 mm
Largura	250 mm
Altura	160 mm
Distância entre eixos	275 mm
Altura do solo	26 mm
Dimensões pneus dianteiros	86 mm (diâmetro) x 32 mm (largura)
Dimensões pneus traseiros	86 mm (diâmetro) x 40 mm (largura)
Capacidade do tanque de combustível	75 ml
Combustível	Nitro 18%
Transmissão	2 marchas
Peso seco	1,78 kg
Escala	1/10
Frequência de transmissão de rádio	2.4 GHz

O princípio de funcionamento do motor deste modelo foi explicado anteriormente, na seção 2.1.3.3. A abertura do carburador é controlada por um servomotor que inicialmente era comandado pelo rádio controle. Como será visto posteriormente, este será removido da malha de controle, sendo substituído por um microcontrolador. O servomotor é acoplado mecanicamente ao carburador e, com isso, folgas estão presentes a todo momento. A Figura 26 mostra estes componentes e sua conexão.

A mistura de combustível e ar atmosférico em conjunto com a umidade relativa afeta o comportamento do automodelo. Como a quantidade de combustível que é inserida no cilindro por ciclo de combustão é pequena, a menor variação de umidade faz com que o combustível esteja mais ou menos diluído. Para contornar este problema, o motor deste modelo possui duas

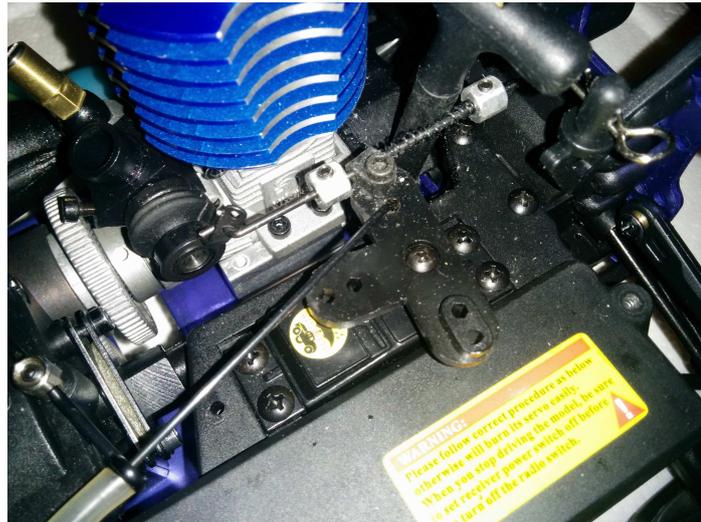
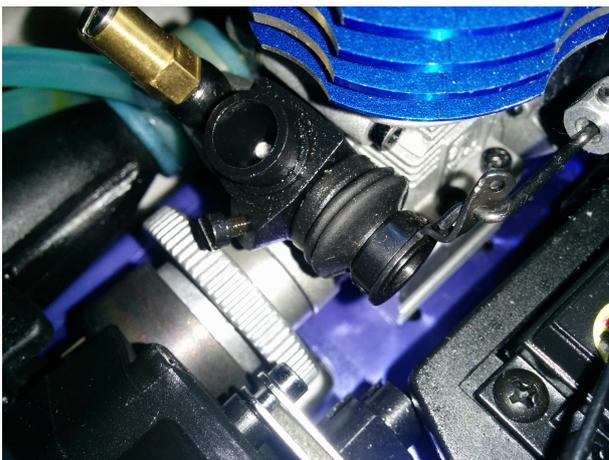


Figura 26: Acoplamento mecânico entre servomotor e carburador

agulhas que regulam a proporção da mistura combustível/ar, mostradas na Figura 27. A primeira agulha, conhecida como agulha de Baixa, regula a proporção para as baixas velocidades. Já a segunda, agulha de Alta, para as altas velocidades. Ambas funcionam da seguinte forma: ao serem giradas no sentido horário, diminuem a passagem de combustível fazendo com que a mistura fique mais “limpa”, com maior quantidade de ar. Ao girar no sentido contrário, aumenta a proporção de combustível e a mistura torna-se mais “rica”. Caso testes sejam feitos em diversos dias, deve-se ficar atento às condições climáticas e ajustar as agulhas para que o automodelo tenha comportamento regular e não desvie muito do modelo obtido.



(a) Detalhe da agulha de Baixa (parafuso preto)



(b) Detalhe da agulha de Alta (parafuso dourado)

Figura 27: Agulhas de Alta e Baixa

3.2 PLATAFORMA DE TESTES

3.2.1 Primeira estrutura

Para que não fosse preciso criar um sistema totalmente automatizado, em que o automodelo ficasse confinado em uma pista fechada, foi decidido que o sistema seria testado de forma estacionária. A primeira ideia foi de construir algo parecido com dinamômetro automotivo, como o da Figura 28.



Figura 28: Estrutura modelo. Retirado de <http://www.dynomitedynamometer.com/>.

Foi realizada uma tentativa de construir algo semelhante, utilizando tubos de PVC (*Polyvinyl chloride*) de meia, uma e duas polegadas e alguns conectores do mesmo material. No caso, a base era formada de tubos conectados por joelhos. Medidas foram tomadas e os tubos foram cortados para que o automodelo ficasse suspenso com as rodas bem encaixadas no suporte construído. Também havia uma inclinação entre rodas traseiras e o chão, controlada por um encaixe mecânico. O resultado está representado na Figura 29.



Figura 29: Primeira estrutura. Autoria própria.

3.2.1.1 Problemas da primeira estrutura

Os tubos onde as rodas se encaixavam possuíam maior diâmetro que o tubo interno e rodavam em falso quando o automodelo acelerava (conforme previsto). Porém, este sistema apresentou dois problemas: o atrito estático do tubo PVC e a instabilidade da estrutura em alta velocidade.

Como o atrito estático entre os tubos era grande, muitas vezes o automodelo não conseguia girar as rodas partindo do repouso. Isso gera um grande problema, pois desgasta todas as peças do automodelo, em especial a transmissão e o motor, onde a temperatura fica muito elevada.

Já a instabilidade em alta velocidade também era problemática, pois a estrutura inteira balançava, o que tornava mais difícil controlar a direção e medir a velocidade com um sensor. O suporte de PVC era forte, porém, isso era um problema recorrente.

Quando esta estrutura foi construída, algumas condições para ligar e aquecer o motor, assim como a regulagem da mistura de ar e combustível não eram conhecidas. Essas informações são muito importantes e não estavam no manual do automodelo. Isso também atrapalhou nos testes.

Colocar rolamentos na estrutura provavelmente ajudaria com estes problemas, porém, depois de pesquisas, foi decidido que construir uma base diferente poderia facilitar esta parte do trabalho. Adaptar este sistema com rolamentos seria bem difícil, por falta de ferramentas e experiência com trabalhos técnicos que envolvem mecânica.

Como esta estrutura foi utilizada por pouco tempo, não são dados detalhes sobre a construção da mesma neste trabalho.

3.2.2 Segunda estrutura

A segunda ideia foi fazer algo parecido com rolos de treinamento para bicicleta como na Figura 30.

Há algumas diferenças entre a Figura 30 e o que foi feito. Na Figura 31, notam-se algumas diferenças e as semelhanças. Uma esteira foi colocada para evitar o contato direto da roda com os rolos. O automodelo fica aproximadamente no meio desta estrutura.

Todas as placas e os sarrafos de madeira da estrutura da Figura 31 possuem aproximadamente 3 cm de espessura. Os sarrafos de madeira são vendidos a partir de 1,5

m, ou seja, foi necessário cortar as outras partes de 50 cm de comprimento. Uma serra circular de bancada da UTFPR foi utilizada para realizar o corte.



Figura 30: Rolos de treino para bicicleta. Retirado de goo.gl/2df2SF.

Cantoneiras de 4x4 cm foram utilizadas para conectar os sarrafos. Cada cantoneira é fixada com quatro parafusos de madeira. Para reforçar a estrutura, mais dois pregos foram fixados em cada canto.

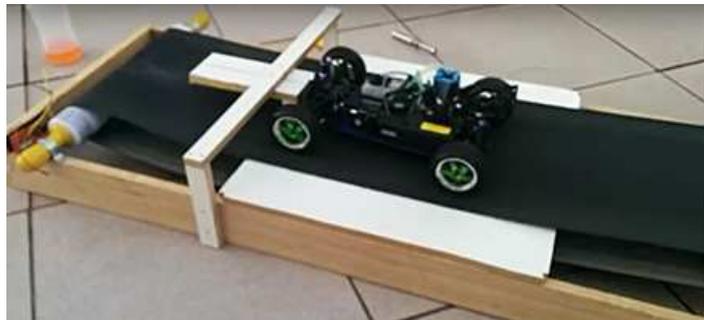


Figura 31: Segunda Estrutura. Autoria própria.

Outra cantoneira de 4x4 cm serve para fixar o sensor na estrutura. Para isso, mais quatro parafusos foram utilizados. A placa do sensor estava colada na placa de 6x3 cm com fita dupla face.

Para colocar a esteira de 1,1 m de comprimento, um dos rolos foi colocado com duas braçadeiras de metal, fixadas com dois parafusos cada. O outro rolo foi colocado dentro da esteira e ela foi esticada a cerca de 1,1 m de distância. Para segurar o segundo rolo em posição, duas cantoneiras de 3x3 cm foram colocadas, cada uma fixada com um parafuso.

O automodelo deveria ficar próximo ao meio da estrutura, então foi decidido colocar a placa de 50x50 cm por onde a esteira deslizaria e sobre ela o automodelo ficaria apoiado.

Por fim, para que o automodelo não escapasse para fora da estrutura, foi necessário

colocar uma barreira física por cima da esteira. As duas placas de 17x3 cm foram fixadas nos lados da estrutura, conectando a base com a placa de 56x3 cm, com quatro pregos em cada lado. Na placa de 56x3 cm outras duas placas foram fixadas para realizar o contato físico com o automodelo, uma de 21x9 cm, fixada com dois pregos, e outra de 9x3 cm, fixada com fita dupla face.

3.2.2.1 Problemas da estrutura de madeira com esteira

Foram feitos vários testes, já utilizando o sensor para medir a velocidade. O principal problema foi a esteira. Se a esteira está muito tensionada, o atrito estático é muito alto para realizar testes. Porém, se está um pouco tensionada, a esteira corre para os lados após aproximadamente um minuto de teste. Apesar de aparentar ser um problema simples, infelizmente esta ideia foi descartada após várias tentativas de alinhar a esteira. Todavia, boa parte desta estrutura foi reutilizada posteriormente.

3.2.3 Estrutura final

Foi decidido voltar à ideia inicial utilizando os rolos e descartando a esteira e os tubos de PVC. Algumas mudanças foram realizadas na estrutura anterior, mas a maioria dos materiais foi reaproveitada (Figura 32).

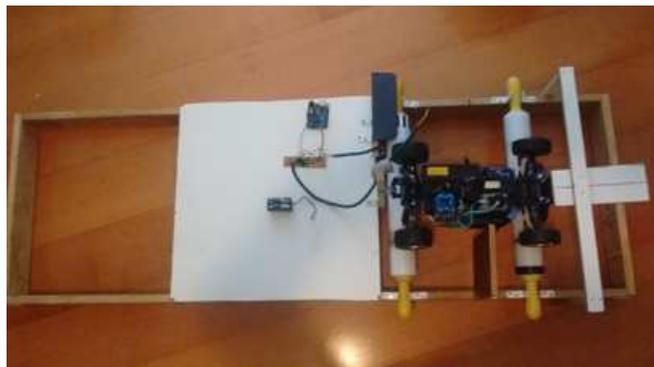


Figura 32: Estrutura final. Autoria própria.

Cada braçadeira de metal foi substituída por duas cantoneiras de 4x4 cm para fixação dos rolos. Isso foi feito porque os rolos ainda apresentavam folga com as braçadeiras. Para que o automodelo ficasse estável em cima dos rolos, sem deslizar para frente ou para trás, um encaixe com dois parafusos de 3x5 mm foi realizado na placa de madeira à frente do automodelo (Figura 33).



Figura 33: Encaixe frontal. Autoria própria.

Por segurança, o automodelo também foi preso na parte traseira por uma braçadeira parafusada a um pedaço de madeira de 2,5x41,5 cm, fixada na estrutura com duas cantoneiras de 3x3 cm. A parte de encaixe do aerofólio foi colocada entre a braçadeira de metal e a placa de madeira. Depois, um parafuso de 4x30 mm foi colocado para fixação (Figura 34).



Figura 34: Encaixe do aerofólio. Autoria própria.

Para a fixação do sensor, a mesma placa de madeira foi aproveitada. Porém, foi colocada mais uma cantoneira para melhorar a fixação da placa. Também foi colocado um parafuso de 2x40 mm para ajustar a posição do sensor em relação ao rolo. Uma placa de plástico preta foi colada com fita dupla face em cima da placa de madeira do sensor para evitar interferência de luz por cima da estrutura (Figura 35).



Figura 35: Placa de madeira para fixação do sensor. Autoria própria.

O escapamento do automodelo expela muito combustível. Para contornar este problema, foi fixada uma outra placa de madeira de 15x19 cm entre a estrutura e o escapamento com uma cantoneira de 4x4 cm. Assim, este combustível sempre escorre para baixo da estrutura, evitando o problema de molhar o rolo da parte traseira (Figura 36).



Figura 36: Placa de madeira que bloqueia o combustível expelido. Autoria própria.

O material utilizado para a construção da estrutura final foi:

1. Ripas de madeira pinus de 1.5 m (x2), 50 cm (x2);
2. Rolos 40x6 cm de poliéster (x2);
3. Cantoneiras de metal de 4x4 cm (x14);
4. Cantoneiras de metal de 3x3 cm (x3);
5. Braçadeira de metal de 5 cm de diâmetro (x1);
6. Parafusos cabeça chata 4x30 mm (x48);
7. Pregos 15x21 mm (x16);
8. Fita dupla face;
9. Joelho PVC 2" (x1);
10. Placas de madeira: 50x50 cm, 56x3 cm, 17x3 cm (x2), 21x9 cm, 9x3 cm, 6x3 cm, 15x19 cm, 2,5x41,5 cm;
11. Ferramentas: parafusadeira, lixas de madeira, chave de fenda, tesoura, alicate de bico, martelo.

3.3 ENCODER

3.3.1 Escolha do sensor

Para fechar a malha de controle é preciso medir a velocidade do automodelo. O rolo é puxado pelas rodas do automodelo. Em um dos rolos foi colado um pedaço de papel, com faixas brancas e pretas. A ideia foi de medir a velocidade do automodelo a partir da contagem de mudanças entre as faixas (Figura 37).



Figura 37: Faixas brancas/pretas. Autoria própria.

O sensor óptico TCRT5000 (Figura 38) foi escolhido para realizar esta medição. De acordo com o *datasheet* do fabricante, este sensor é constituído de um LED (*light emitting diode*) e um fototransistor (receptor) isolado opticamente.

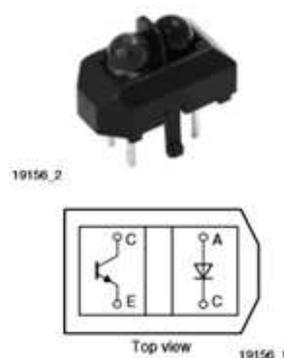


Figura 38: Sensor óptico TCRT5000. Retirado do *datasheet* do fabricante.

A onda emitida pelo LED pode saturar o fototransistor (Figura 39), similar à corrente de base de um transistor comum. De acordo com o *datasheet*, esta onda tem o comprimento de 950 nm, o sensor possui filtro para ondas de outros comprimentos e a melhor distância para detecção é de 2,5 mm (Figura 40). Esta distância é definida para superfícies planas.

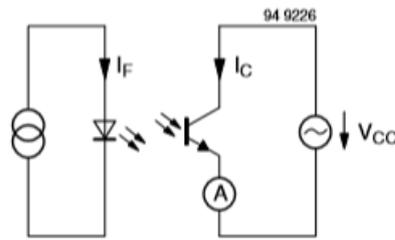


Figura 39: Esquemático do sensor. Retirado do *datasheet* do fabricante.

Alguns testes foram feitos para verificar o funcionamento do sensor, como a leitura de uma das portas analógicas do microcontrolador ATmega328p, utilizando a IDE (*Integrated Development Environment*) Arduino. O circuito está representado na Figura 41, utilizando o software EAGLE.

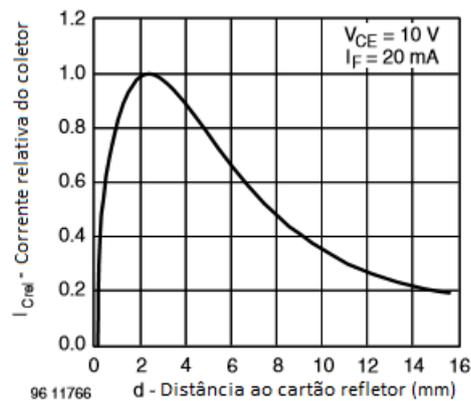


Figura 40: Teste de detecção para superfícies planas. d é a distância do cartão refletor, I_{Crel} é a corrente relativa no coletor. Adaptado do *datasheet* do fabricante.

O sinal medido no coletor do fototransistor é conectado à porta analógica A0 do microcontrolador. Na Figura 41, o diodo e o fototransistor não estão isolados opticamente, mas na realidade estão. O resistor R2 tem a função de manter o pino do coletor em nível lógico alto (5V). O resistor R1 limita a corrente no LED em aproximadamente 37,5 mA, pois a queda de tensão no LED é aproximadamente 1,25 V e a alimentação é de 5V.

Ao se aproximar um objeto refletor do sensor, a tensão medida pelo conversor AD do microcontrolador é próxima a zero, pois a luz é refletida e chega com intensidade suficiente ao transistor. Quando aproxima-se um objeto opaco perto do sensor, a intensidade da reflexão é muito baixa. No entanto, se o objeto aproximado está entre o claro e o escuro, a medição oscila bastante. Isso ocorre pois o fototransistor responde rapidamente a qualquer variação da luz refletida.

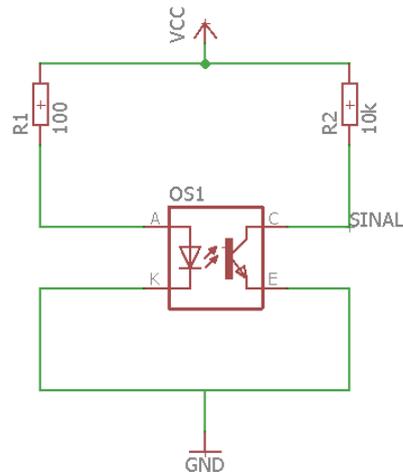


Figura 41: Configuração básica do sensor óptico. Autoria própria.

3.3.1.1 Circuito e simulações

A ideia para medir a velocidade é de contar quantas mudanças ocorrem em um período de tempo entre as faixas brancas e pretas da Figura 37, porém apenas o sensor é muito impreciso para a contagem. Quando o sensor mede entre uma faixa branca e outra preta, a leitura varia muito, contando mudanças que não ocorreram. Para diminuir este efeito, decidiu-se fazer um circuito auxiliar para que apenas a mudança de cores fosse contabilizada. Ou seja, o objetivo é obter uma onda quadrada do sensor para facilitar a leitura da velocidade. O esquemático do circuito está representado na Figura 42 e foi obtido com o software *EAGLE*.

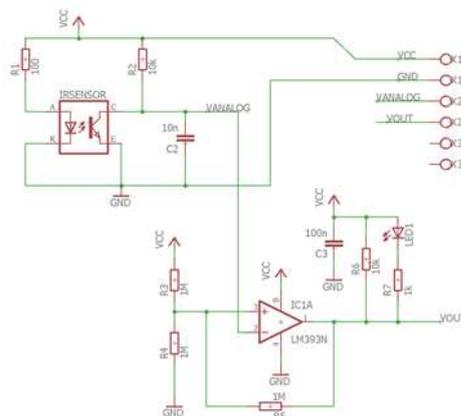


Figura 42: Esquemático do circuito utilizado. Autoria própria.

Os resistores R1 e R2 são equivalentes ao da Figura 41. Na Figura 42, observa-se que há um capacitor de 10 nF na saída do coletor do fototransistor, colocado para atenuar sinais em alta frequência.

Quando o fototransistor está saturado, a corrente passa por R2 em direção ao terra (GND). Já quando não está saturado, o caminho de menor impedância é pelo capacitor C2, visto que a entrada do amplificador possui alta impedância. Para simplificar, considera-se que não há corrente fluindo pela entrada não-inversora do amplificador.

Pelo teoria de filtro passa-baixas passivo (ALEXANDER; SADIKU, 2013), a frequência de corte para o filtro RC é de

$$\omega_c = \frac{1}{RC} = 2 \times \pi \times f, \quad (9)$$

e, portanto,

$$C_2 = \frac{1}{\omega_c R_2} = \frac{1}{2 \times \pi \times f \times R_2}. \quad (10)$$

Como existem onze faixas brancas próximas ao sensor, há onze pulsos em uma volta. Na velocidade de 10 m/s (máxima para este projeto), sabendo que o diâmetro do rolo é de 6 cm, ele dá uma volta completa, aproximadamente, 53 vezes a cada segundo (n):

$$n = \frac{v}{d \times \pi} = \frac{10}{\pi \times 0,06} = 53. \quad (11)$$

Como há onze pulsos por volta, o número total de pulsos é:

$$pulsos = 53 \times 11 = 583. \quad (12)$$

De acordo com estas considerações, a frequência de corte é de 583 Hz. Para frequências maiores, o sinal é atenuado. Logo, a partir da equação (10), pode-se determinar a capacitância de C2:

$$C_2 = \frac{1}{2 \times \pi \times f \times R_2} = \frac{1}{2 \times \pi \times 583 \times 10^4} = 27 \times 10^{-9} [F]. \quad (13)$$

A constante de tempo para o circuito com 27 nF de capacitância é:

$$\tau = R_2 \times C_2 = 10000 \times 27 \times 10^{-9} = 0,27 \times 10^{-3} [s]. \quad (14)$$

Considerando que cada faixa branca ou preta tem a mesma largura, metade do período do sinal em 583 Hz dura aproximadamente 0,85 ms. O capacitor de 27 nF demora apenas 0,27 ms para atingir 63,2% (CAVALCANTE et al., 2011) da carga máxima (3,16 V). Este já é um valor suficiente para detecção em menos de 0,85 ms. Na Figura 43, está o esquemático da

simulação com o circuito descrito anteriormente. Porém, o valor utilizado para o capacitor foi de 25 nF, por ser um valor comercial.

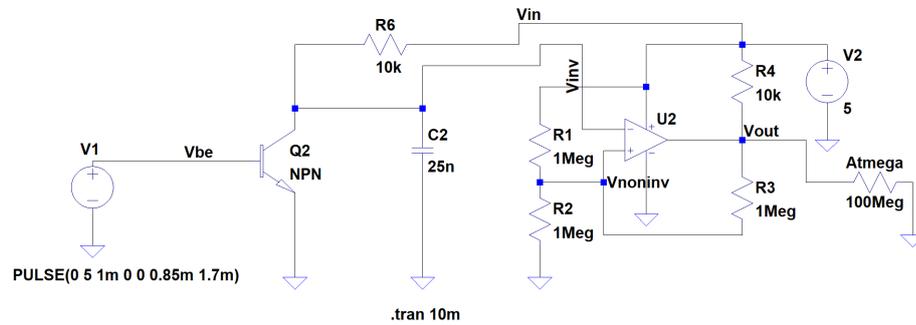


Figura 43: Simulação do circuito no LTSpice. Autoria própria.

Utilizando o software *LTSpice*, o emissor foi substituído por uma fonte de tensão de onda quadrada, simulando o que o emissor faria, e todos os componentes são genéricos. Quando o fototransistor satura, a maior parte da corrente passa por ele, pois possui menor impedância que o capacitor em paralelo. Logo, a tensão no coletor é de 1,6 V. Já quando está na região de corte, a corrente passa pelo capacitor e este acumula a tensão. As curvas do sinal entre base e emissor (azul) e do sinal na saída do coletor (preto) estão na Figura 44.

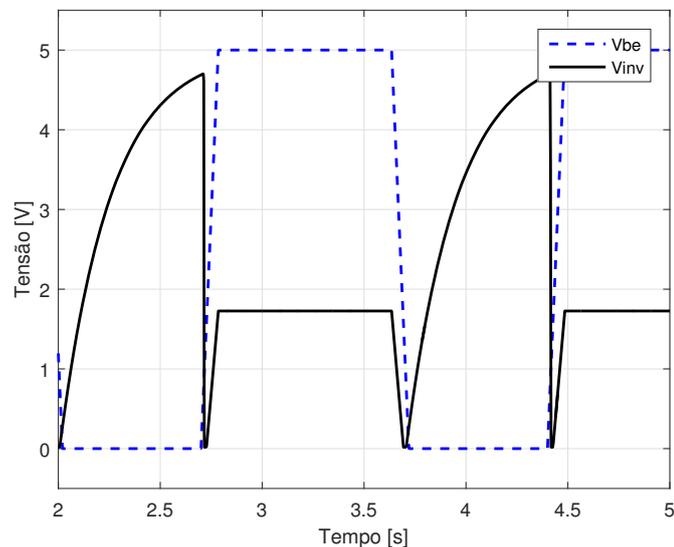


Figura 44: Curvas de sinais do sensor. Autoria própria.

Na prática, o valor definido para C2 foi de 10 nF. Na realidade, os pulsos não têm exatamente o mesmo tamanho, já que as faixas foram colocadas artesanalmente e o sensor é posicionado manualmente. Além disso, a superfície do rolo não é plana.

Com uma capacitância um pouco menor, pode-se ter mais certeza de que as mudanças de estado acontecem na interrupção externa de um microcontrolador, porque o transiente é

mais rápido. Infelizmente, perde-se a atenuação a partir da frequência de 583 Hz. Refazendo o cálculo com as equações 11 e 12, a nova frequência de corte é de 1591 Hz. Apesar de admitir mais sinais ruidosos, o capacitor de 10 nF não apresentou grande diferença para o de 25 nF nos testes.

Em seguida, este sinal do coletor do fototransistor passa por um amplificador operacional do circuito integrado LM393. Ele possui dois amplificadores operacionais, porém apenas um é utilizado. De acordo com o *datasheet*, a alimentação não precisa ser simétrica.

A configuração utilizada é o comparador inversor com histerese. O *datasheet* do fabricante dá um exemplo desta configuração, como pode-se ver na Figura 45. Essa configuração permite definir dois limites, um para saída digital positiva (5 V) e outro para saída digital negativa (0 V). Assim, é possível prevenir alguma flutuação de tensão entre estes limites.

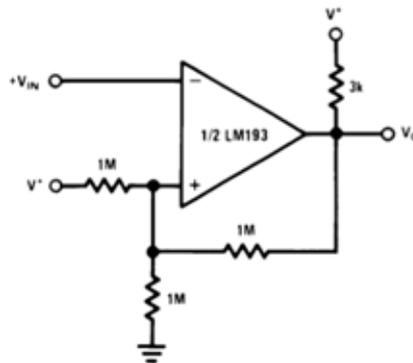


Figura 45: Exemplo de comparador inversor com histerese. Retirado do *datasheet* do fabricante.

Tomando a Figura 42 como referência, quando a saída do comparador está em nível lógico baixo, a tensão na porta não-inversora é:

$$V_{ninv} = (R5 || R4) \times \frac{V_{cc}}{R3 + R4 || R5} = 1,67[V]. \quad (15)$$

Ou seja, para que o estado lógico na saída do comparador seja 0 V, é necessário que a tensão seja menor do que 1,67 V na entrada inversora. Porém, se a saída do comparador está em nível lógico baixo, a tensão da porta não-inversora é:

$$V_{ninv} = R4 \times \frac{V_{cc}}{R4 + R3 || R5} = 3,33[V]. \quad (16)$$

Então, para que a saída esteja em 5 V, deve-se aplicar uma tensão acima de 3,3 V na

porta inversora. Os limiares de tensão apenas mudam quando o comparador muda de estado, como pode-se notar pela Figura 46. A curva em vermelho é a tensão de saída do comparador, a marrom é a entrada não inversora e a preta é a entrada inversora.

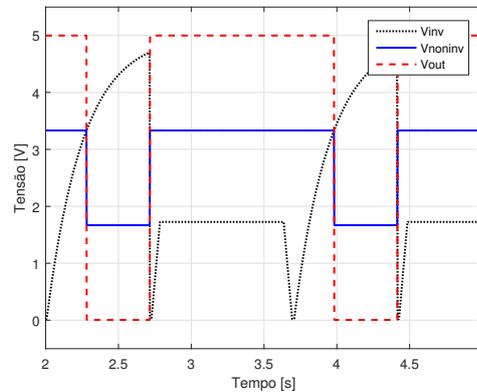


Figura 46: Saída do comparador. Autoria própria.

No esquemático da Figura 42, há um resistor de *pull-up* na saída do comparador (recomendação do *datasheet*), mantendo o pino do microcontrolador em 5 V ao ligar o sistema. O *datasheet* do fabricante não especifica a impedância de saída do amplificador, porém ela é baixa por definição.

O LED1 da Figura 42 tem apenas a função de emitir luz no momento em que a saída do amplificador está em nível lógico alto ou não emitir quando o amplificador está em nível lógico baixo. Isso ajuda a verificar se existe um problema no posicionamento do sensor em relação ao rolo em movimento, apenas alimentando o circuito. Na Figura 47, pode-se ver uma imagem da saída do amplificador quando o rolo está em movimento e o sensor bem posicionado.

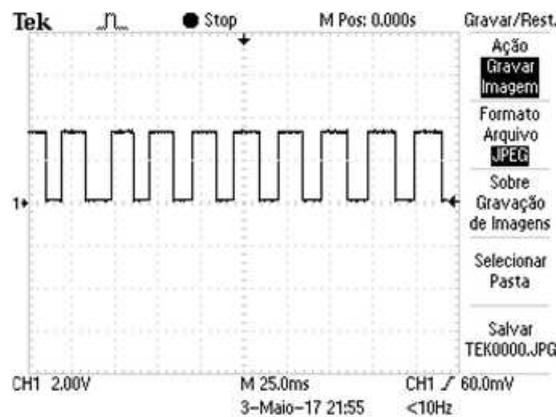


Figura 47: Saída do comparador capturada no osciloscópio. Autoria própria.

A placa deste circuito foi confeccionada artesanalmente utilizando o método de transferência de calor. O roteamento pode ser visto na Figura 48.

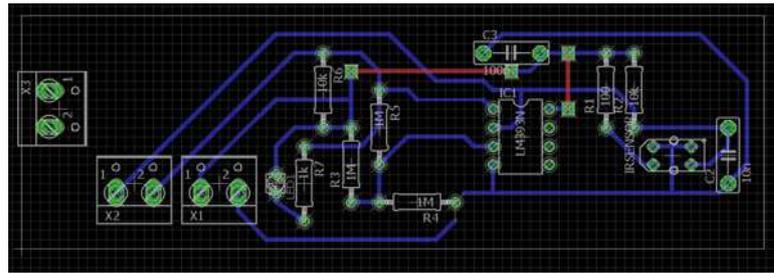


Figura 48: Circuito confeccionado artesanalmente. Autoria própria.

3.4 IDENTIFICAÇÃO DO SISTEMA VIA MODELO DE HAMMERSTEIN E WIENER

Para que seja possível sintonizar o controlador por AG, é mais vantajoso modelar o sistema. Desta forma, é possível sintonizar o controlador sem realizar múltiplos testes no automodelo e verificar quais são os melhores ganhos para o controle de velocidade.

O automodelo apresenta diversas características que o torna um sistema variante no tempo. Seu comportamento muda por várias razões, tais como: temperatura ambiente, regulagem das entradas de ar e de combustível, temperatura do motor, alinhamento e quantidade de combustível no tanque.

Neste trabalho, a modelagem utilizada se baseia nas representações de Hammerstein e de Wiener (AGUIRRE, 2004), que utilizam modelos de blocos interconectados para representarem sistemas dinâmicos não-lineares. Nesta seção, há uma explicação breve sobre estas representações.

3.4.1 Modelos de Hammerstein-Wiener

Estes modelos surgiram na década de 70. São uma composição de um modelo dinâmico linear $H(s)$ em cascata com uma função estática não-linear $f(\cdot)$. O modelo de Hammerstein apresenta a não-linearidade estática antes do modelo linear. Já no modelo de Wiener, o modelo dinâmico linear precede a não-linearidade estática (AGUIRRE, 2004).

Ambos modelos podem ser definidos em tempo discreto. O modelo de Hammerstein é descrito pela equação (17).

$$\begin{aligned} y(k) &= a_1 y(k-1) + \dots + a_{n_y} y(k-n_y) + b_1 v(k-1) + \dots + b_{n_u} v(k-n_u) \\ &= a_1 y(k-1) + \dots + a_{n_y} y(k-n_y) + f[u(k-1) + \dots + u(k-n_u)]. \end{aligned} \quad (17)$$

O modelo de Wiener é descrito pela equação (18):

$$y(k) = f[a_1 v(k-1) + \dots + a_{n_y} v(k-n_y) + \dots + b_{n_u} u(k-\tau_d)] \quad (18)$$

$$= f[a_1 f^{-1}(y(k-1)) + \dots + a_{n_y} f^{-1}(y(k-n_y)) + b_0 u(k-\tau_d) + \dots + b_{n_u} u(k-n_u)].$$

Na Figura 49, assume-se que f é invertível e τ_d é o atraso puro de tempo. Em problemas de identificação, apenas $y(k)$ e $u(k)$ são conhecidos, a função $f(\cdot)$ é desconhecida (AGUIRRE, 2004).

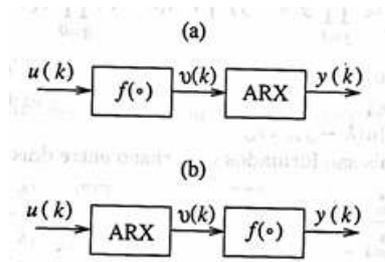


Figura 49: (a) modelo de Hammerstein, (b) modelo de Wiener. Em aplicações práticas $v(k)$ é desconhecido (AGUIRRE, 2004).

Ao observar a Figura 49, não é evidente quais são as diferenças entre os modelos. Porém, é possível mostrar analiticamente que os autovalores da matriz jacobiana dos modelos de Hammerstein são constantes. Nos modelos de Wiener, tais autovalores dependem do ponto de operação (COELHO et al., 2002) e (AGUIRRE et al., 2005). Ao utilizar ambos os modelos para um mesmo sistema, tem-se o modelo de Hammerstein e Wiener, o qual foi utilizado neste trabalho.

3.4.2 Processo de identificação

Uma entrada pseudo-aleatória, gerada a partir do *script* que se encontra no Anexo A, foi utilizada para obter a resposta do automodelo à diferentes valores de abertura do carburador. Este *script* gera um vetor de entrada com valores limitados no intervalo fechado entre 90° e 105° , com metade de seus valores em baixa frequência (maior tempo de permanência em determinado ângulo) e a outra metade em alta (menor tempo de permanência). A entrada pode ser observada na Figura 50.

Com o perfil de velocidade obtido com o *encoder* a partir desta entrada, foi possível determinar o modelo não-linear de Hammerstein e de Wiener com o comando *ident* do MATLAB. Este comando permite gerar o modelo ao se alterar os seguintes parâmetros, que também estão representados na Figura 51:

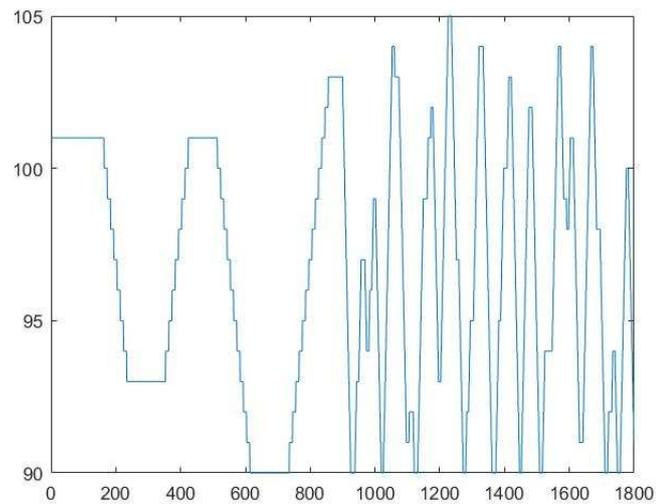


Figura 50: Curva de entrada para identificação do automodelo. Autoria própria.

- Número de polos;
- Número de zeros;
- Número de atrasos;
- Número de unidades do canal de entrada;
- Número de unidades do canal de saída.

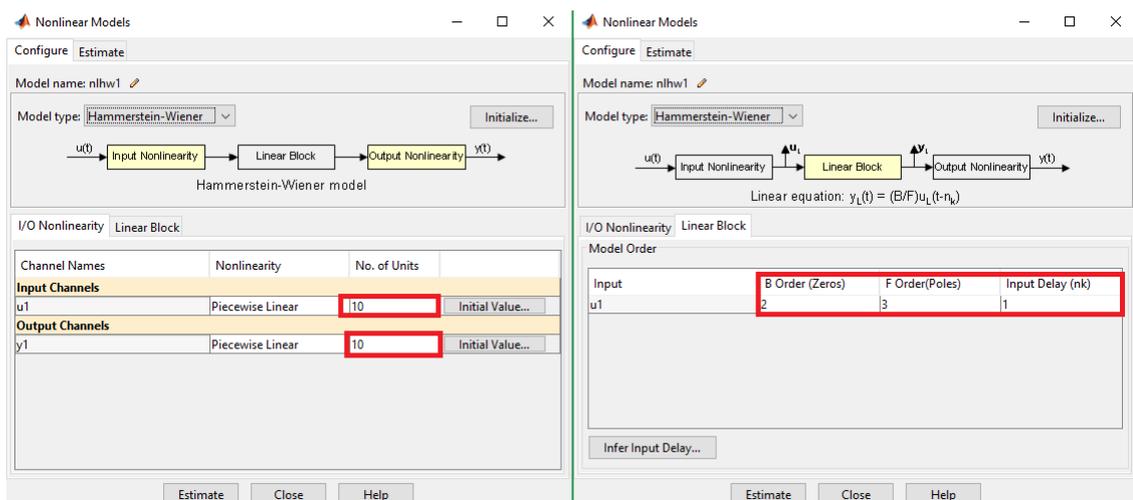


Figura 51: Parâmetros do modelo de Hammerstein e Wiener. Autoria própria.

O processo de escolha dos parâmetros é empírico. Os parâmetros são ajustados para cima ou para baixo de acordo com o *fitness* obtido.

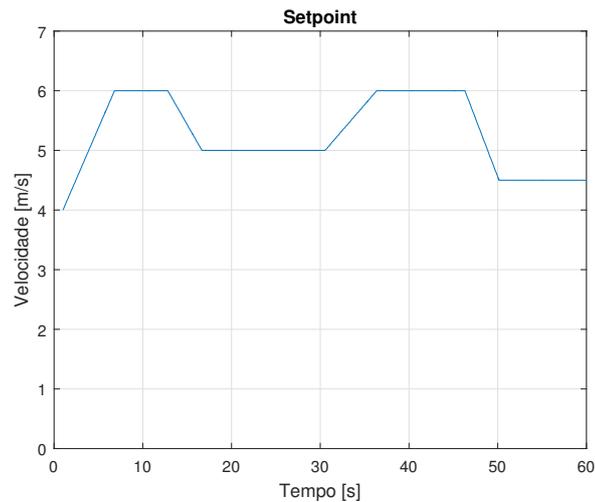


Figura 53: Vetor de *setpoints*. Autoria própria.

que a média dos valores obtidos esteja ao redor de cinco, o que contribui para que o contador interno do microcontrolador não ultrapassasse seu limite durante a soma dos valores do PID.

3.6 CONTROLE PID

3.6.1 Rotina de Inicialização

Antes de se iniciar a rotina PID com o vetor de *setpoints* mostrado na Figura 53, é necessário tirar o automodelo do estado de repouso pois caso o controlador demande uma aceleração brusca, isto pode danificar as engrenagens da transmissão do automodelo. Para isso, uma rotina é utilizada para que a abertura do carburador seja incrementada gradualmente até que o automodelo atinja a velocidade de 3 m/s.

3.6.2 Rotina PID

Para se manter o mais próximo possível ao modelo obtido via MATLAB, a mesma equação do PID utilizada no SIMULINK, apresentada na equação (19), foi utilizada no microcontrolador. Nesta equação, K_P representa o ganho proporcional, K_I o integral, K_D o derivativo, N o filtro da parcela derivativa e T o tempo de amostragem.

$$\frac{Y(z)}{E(z)} = K_P + K_I \times T \times \frac{1}{z-1} + K_D \times \frac{N}{1 + N \times T + \frac{1}{z-1}}. \quad (19)$$

Para que pudesse ser implementada, a equação (19) teve que ser transformada em uma

equação de diferenças. Cada um dos termos da soma foi transformado separadamente, com a equação (20) representando o termo proporcional, a equação (21) o termo integrativo e a equação (22) o termo derivativo.

$$P = K_p \times e[x]. \quad (20)$$

$$I = K_I \times T \times e[x - 1] + I[x - 1]. \quad (21)$$

$$D = K_D \times N \times (e[x] - e[x - 1]) + D[x - 1] \times (1 - N \times T). \quad (22)$$

Os termos que contém $[x]$ e $[x - 1]$ denotam, respectivamente, os valores atuais e os valores da iteração passada. Ao somar as três equações acima, tem-se a equação do controlador PID utilizada pelo microcontrolador, representada pela equação (23).

$$PID = P + I + D. \quad (23)$$

Os valores de T e N utilizados neste projeto são 0,1 e 5. O código completo e comentado é encontrado no Anexo C.

4 RESULTADOS E DISCUSSÕES

4.1 IDENTIFICAÇÃO DO SISTEMA

4.1.1 Processo de Identificação

Como explicado na seção 3.4.2, o automodelo foi submetido ao vetor de *setpoints* apresentado na Figura 50. No momento do teste, realizado a céu aberto, às 18:25 horário de Brasília, a temperatura ambiente era de 20°C, umidade de 63% e velocidade do vento de 9 km/h. Assim, o perfil de velocidade do automodelo neste teste é apresentado na Figura 54.

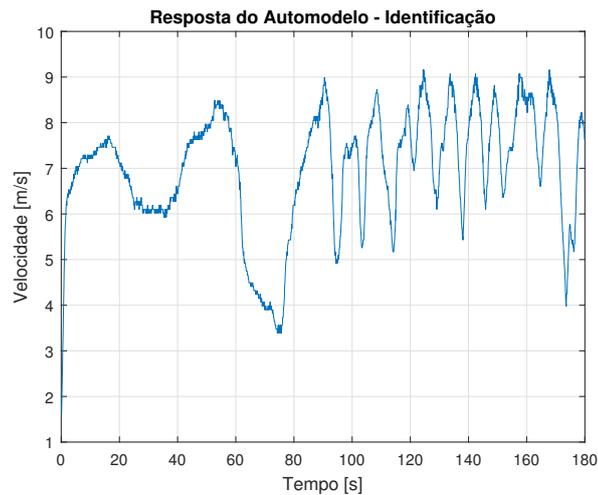


Figura 54: Resposta do automodelo para identificação. Autoria própria.

Com este resultado, foi possível determinar o modelo não-linear de Hammerstein e Wiener. Os valores dos parâmetros utilizados são mostrados na Figura 55 e resumidos como segue:

- Número de polos = 5;
- Número de zeros = 3;
- Número de atrasos = 1;
- Número de linearizações “parte a parte” do canal de entrada = 12;

- Número de linearizações “parte a parte” do canal de saída = 11;

Apesar de a figura mostrar um valor de zeros, nb , igual a 4, o valor real é 3, já que os arquivos de ajuda do MATLAB para esta função especificam que o número de zeros é o valor de nb decrescido de uma unidade.

```

model =
Hammerstein-Wiener model with 1 output and 1 input
  Linear transfer function corresponding to the orders  nb = 4, nf = 5, nk = 1
  Input nonlinearity: pwnlinear with 12 units
  Output nonlinearity: pwnlinear with 11 units
Name: nlhw2
Sample time: 0.1 seconds

Status:
Estimated using PEM on time domain data "Model".
Fit to estimation data: 72.51%
FPE: 0.1425, MSE: 0.1336
~~

```

Figura 55: Resultado do modelo Hammerstein Wiener. Autoria própria.

O que resultou em um *fitness* de 0,1336 ou 72,51%.

4.1.2 Processo de Validação do Modelo

Utilizando o vetor de *setpoints* da Figura 52, simulou-se o modelo obtido com o comando *sim* do MATLAB, resultando na Figura 56. Esta figura vem em inglês pois é o resultado do comando. Nela, y_1 representa a velocidade do automodelo e o eixo horizontal é o tempo em segundos.

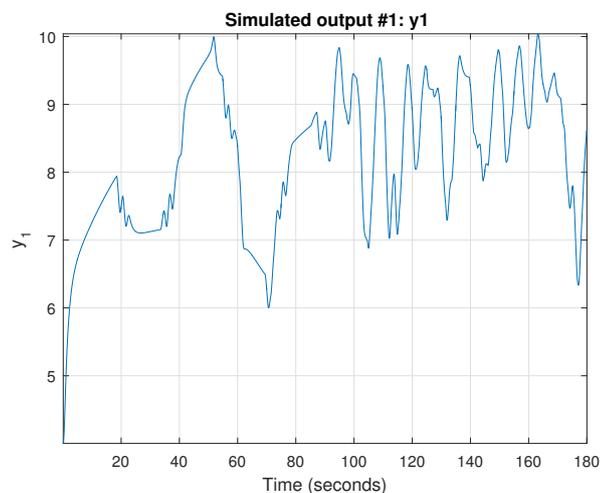


Figura 56: Resposta do modelo digital para validação. Autoria própria.

Já o resultado do automodelo para este mesmo vetor de *setpoints* é apresentado na Figura 57. Nota-se que os os perfis de velocidade são parecidos, mas o resultados do modelo digital têm magnitude maior do que os reais. Isto pode ser justificado pela adequação do modelo à realidade.

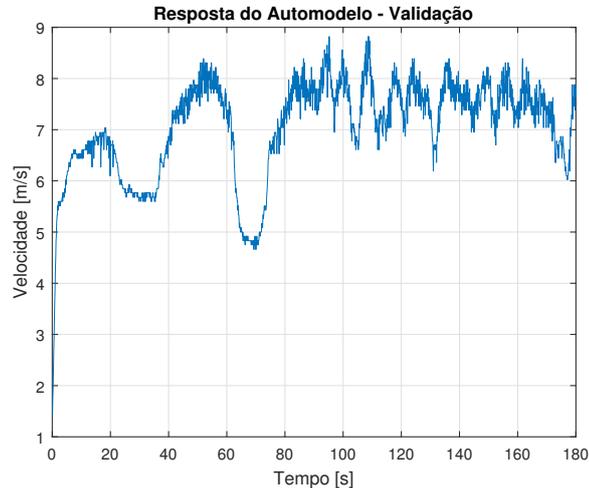


Figura 57: Resposta do automodelo para validação. Autoria própria.

4.2 ALGORITMO GENÉTICO

Com o modelo gerado e validado, utilizou-se o *script* de Algoritmo Genético para encontrar os melhores ganhos K_P , K_I e K_D que fizessem com que o automodelo seguisse a curva de *setpoints* apresentada na Figura 53.

Após 45 gerações, o resultado final é apresentado na tabela a seguir:

Tabela 2: Resultados do Algoritmo Genético

K_P	7,3896
K_I	8,8941
K_D	7,6883
<i>Fitness</i>	0,5842

Estes parâmetros, juntamente com o modelo obtido anteriormente resultam na resposta ao vetor de entrada apresentada na Figura 58. É possível notar que a velocidade do automodelo inicia em 4 m/s, isto se deve ao fato de que os testes de identificação e controle possuem uma rotina de inicialização que faz com que o automodelo acelere até uma velocidade próxima a esta e então inicie a rotina de identificação/controla.

O valor de *fitness* obtido para o modelo digital é considerado relativamente alto, mas pode ser atribuído totalmente à queda brusca de velocidade nos segundos iniciais de simulação.

Desconsiderando esta queda abrupta e recalculando o *fitness* à partir de 10 segundos, o resultado é de 0,0035.

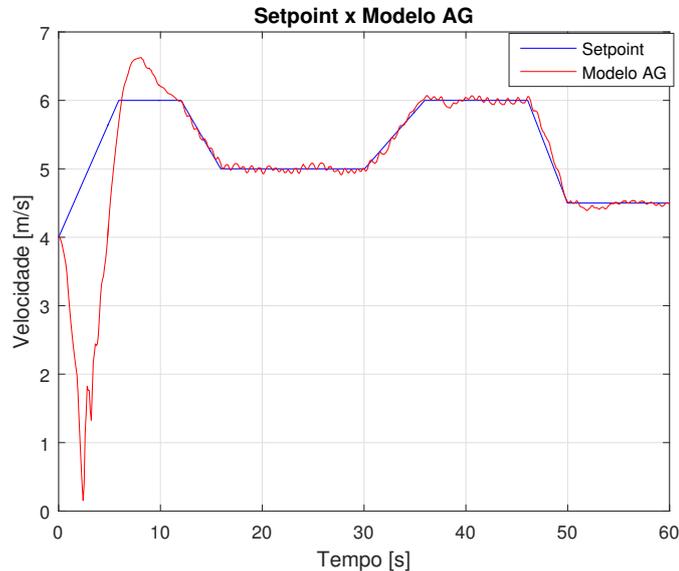


Figura 58: Resposta do modelo digital com ganhos via AG. Autoria própria.

4.3 CONTROLE

4.3.1 Controlador PID

O teste prático do controlador PID ocorreu a céu aberto, em uma noite com 93% de umidade, 14 °C e velocidade de vento de 9 km/h. Estas condições, principalmente temperatura e umidade, são bem diferentes daquelas em que a identificação foi realizada, o que pode afetar o comportamento do automodelo. Utilizando os mesmos ganhos oriundos do Algoritmo Genético, o resultado do teste é apresentado na Figura 59.

O *fitness* deste teste é melhor do que o do modelo digital, atingindo a marca de 0,4817. Isto se deve ao fato de a queda inicial de velocidade ser menor do que a simulada. Fazendo a análise do *fitness* para os 50 segundos finais, atinge-se o resultado de 0,0283. Este aumento comparado à simulação pode ser atribuído à oscilação em torno do *setpoint* que é muito mais assídua no teste prático.

Calculando o *fitness* entre a curva do modelo digital e a do automodelo, obteve-se o valor de 0,313. Novamente, este valor relativamente alto pode ser atribuído ao comportamento dos sistemas durante os 10 segundos iniciais. Ao refazer o cálculo, excluindo este período, o resultado é de 0,027, o que indica grande correlação entre os sinais.

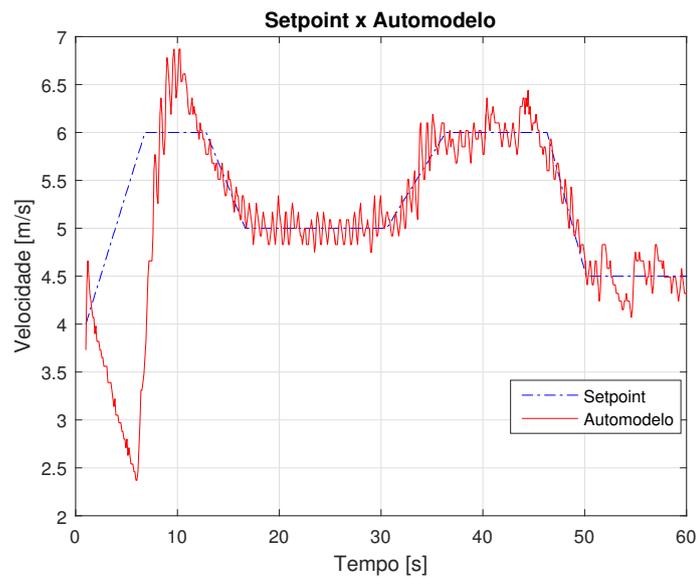


Figura 59: Resposta do automodelo com ganhos via AG. Autoria própria.

Para avaliar a ação do controlador, foi elaborada a Figura 60. A ação real é representada pela linha em azul. É possível notar que o controlador se encontra saturado em 90° durante aproximadamente o mesmo período de tempo em que a velocidade do automodelo cai. Esta abertura do carburador representa o “ponto morto” ou a marcha neutra de veículos comerciais, sendo assim, o motor não faz com que as rodas girem e dependam apenas da inércia. Como há atrito entre os pneus e os rolos, o veículo reduz gradativamente sua velocidade.

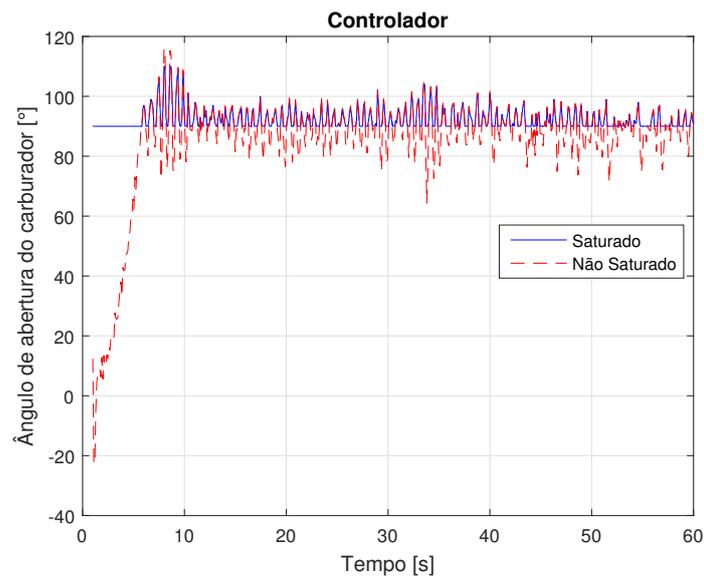


Figura 60: Ação do controlador. Autoria própria.

Uma investigação mais detalhada produziu a Figura 61. Nela a ação individual de cada um dos termos do controlador PID pode ser observada. Pode-se inferir que o principal

responsável pelo controle do automodelo é a parcela integral, que se assemelha à ação do controlador demonstrada na Figura 60. Além disso, a parcela proporcional oscila próximo a zero. Outra observação pertinente é o efeito da parcela derivativa, que parece ser a responsável pela grande oscilação da velocidade do automodelo em torno do *setpoint*. Para averiguar a veracidade desta afirmação, um controlador PI foi elaborado.

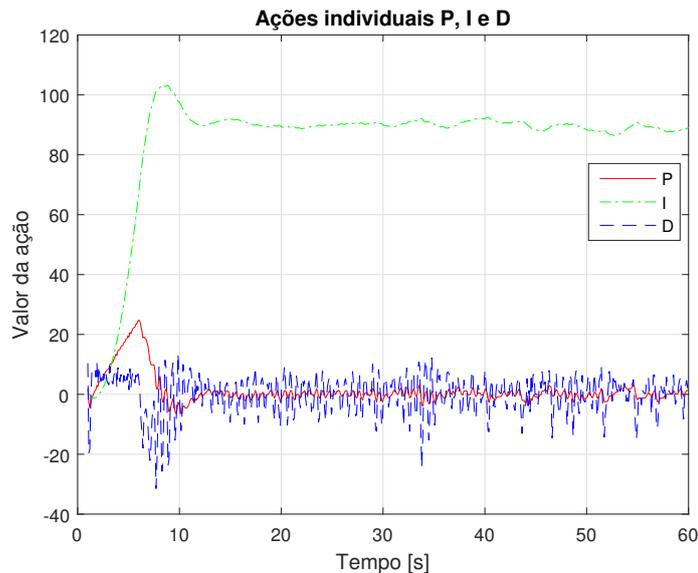


Figura 61: Ação individual dos termos do PID. Autoria própria.

4.3.2 Controlador PI

Para elaborar um controlador PI, novamente foi utilizado o *script* de Algoritmo Genético, mas com o terceiro termo nulo em todas as gerações. Após 45 gerações de 55 indivíduos, atingiu-se o resultado apresentado abaixo.

Tabela 3: Resultados do Algoritmo Genético para controlador PI

K_P	8,0903
K_I	7,9966
<i>Fitness</i>	0,6593

Com estes resultados, a resposta do modelo digital é a apresentada na Figura 62. Novamente, o valor do *fitness* pode ser justificado pela queda abrupta de velocidade. Ao se aplicar este controlador ao automodelo, a resposta é a apresentada na Figura 63. O *fitness* nesse caso é de 0,4852. É possível notar os picos de ação do controlador nos momentos em que a curva de *setpoints* sai de um valor constante para um valor mais baixo. Isso pode ser atribuído à falta da parcela derivativa.

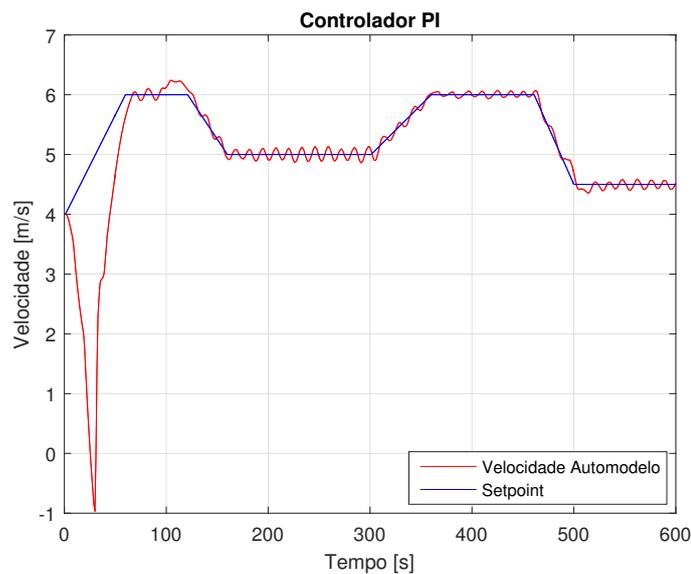


Figura 62: Resposta do modelo digital ao controlador PI. Autoria própria.

O automodelo apresentou significativa redução de oscilação de velocidade, o que pode ser visto ao comparar as Figuras 59 e 63. A ação do controlador também oscila menos, o que pode ser constatado na Figura 64.

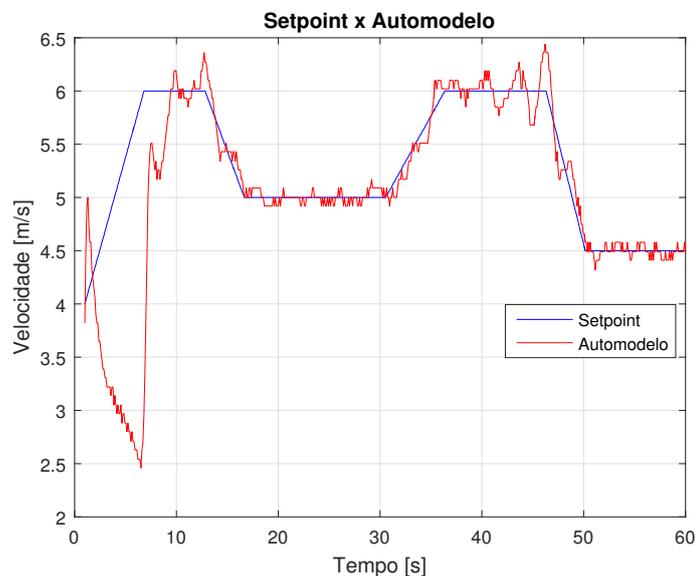


Figura 63: Resposta do automodelo ao controlador PI. Autoria própria.

Uma investigação detalhada das ações Proporcional e Integral deste controlador, presente na Figura 65, comprovam a menor oscilação do sistema. Assim como no caso do controlador PID, a parcela integral é a maior responsável pela atuação do sistema e sua demora em atingir valores próximos aos de atuação do automodelo pode ser responsável pela queda inicial de velocidade. É possível também que o controlador, juntamente com a planta e a

realimentação formem um sistema de fase não mínima, que tem como característica principal sua resposta indo em direção contrária à da entrada durante o início da ação.

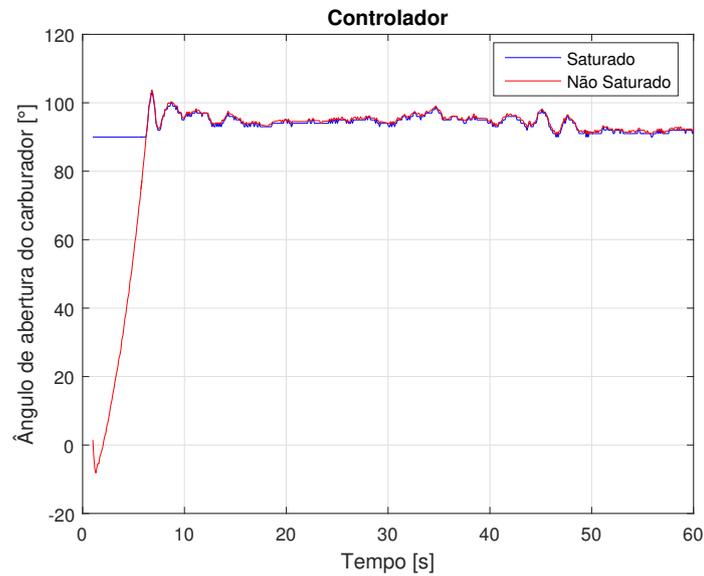


Figura 64: Ação do controlador PI. Autoria própria.

Ao comparar o *fitness* dos 50 segundos finais de teste do controlador PID e PI com valores de, respectivamente, 0,0283 e 0,0139, pode-se dizer que o controlador PI realmente fez com que o automodelo mantivesse velocidades mais próximas dos valores desejados.

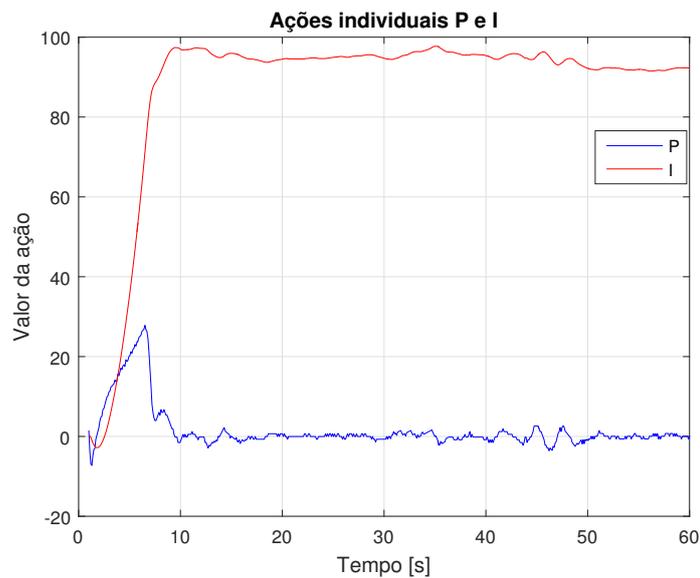


Figura 65: Ação individual dos termos PI. Autoria própria.

5 CONCLUSÕES

5.1 CONSIDERAÇÕES FINAIS E SUGESTÕES DE TRABALHO FUTURO

Apesar de ambos sistemas de controle oscilarem em torno da velocidade desejada, é justo dizer que ambos cumpriram o objetivo deste trabalho. As diversas dificuldades atreladas ao sistema mecânico do veículo, como variação de massa ao longo do tempo (combustível sendo consumido), umidade relativa do ar afetando a proporção de ar e combustível que é injetada no carburador, temperatura ambiente que afeta a combustão e a vibração da estrutura quanto maior a velocidade do automodelo, são responsáveis diretas por essa oscilação.

O *encoder* artesanal utilizado também pode afetar o sistema de controle, já que uma leitura errônea faria com que o erro pudesse aumentar abruptamente e o sistema tentaria compensar de alguma forma. Assim, um *encoder* produzido em escala comercial ou um *encoder* de precisão poderia reduzir este risco.

Já a plataforma de testes poderia ser reforçada, com mais cantoneiras ao se colar as partes estruturais. Além disso, rolamentos poderiam ser confeccionados ao invés de se utilizar rolos de poliéster.

Apesar de o controlador utilizado atingir seus objetivos, um outro sistema de controle (controle preditivo, *feedforward*, *gain scheduling*) poderia apresentar resultados melhores. Outra alternativa seria confirmar se o sistema realmente é de fase não mínima e adicionar um polo no mesmo local do zero que está fora da área de estabilidade, anulando-o.

O Algoritmo Genético é uma das formas mais simples de se convergir ao resultado esperado, outras opções mais robustas e complexas poderiam gerar resultados melhores. Uma alternativa viável seria utilizar um algoritmo de Otimização por Enxame de Partículas, PSO (*Particle Swarm Optimization*), para determinar os parâmetros do controlador.

Com todos os desafios encontrados ao longo deste trabalho, a interação e aprendizado de sistemas que vão além da simulação puderam proporcionar um olhar crítico sob situações mundanas aparentemente simples, mas que podem resultar em dificuldades que influenciam diretamente o resultado final de um projeto.

REFERÊNCIAS

- AGUIRRE, L.; COELHO, M.; CORREA, M. On the interpretation and practice of dynamical differences between hammerstein and wiener models. **IEE Proceedings-Control Theory and Applications**, IET, v. 152, n. 4, p. 349–356, 2005.
- AGUIRRE, L. A. **Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais**. Editora UFMG, 2004.
- ALEXANDER, C. K.; SADIKU, M. N. **Fundamentos de circuitos eléctricos (5a**. McGraw Hill Mexico, 2013.
- ALVES, A. C.; STEINER, M. T. A.; MUSSI, N. H.; ZANELATTO, G. Algoritmos genéticos aplicados ao planejamento da distribuição de energia elétrica em curitiba e região metropolitana. **UFPR. Tese de mestrado defendida em novembro**, 2002.
- ARARAT, Ö.; KURAL, E.; GÜVENÇ, B. A. Development of a collision warning system for adaptive cruise control vehicles using a comparison analysis of recent algorithms. In: **IEEE. Intelligent Vehicles Symposium, 2006 IEEE**. 2006. p. 194–199.
- ASERE, H.; LEI, C.; JIA, R. Cruise control design using fuzzy logic controller. In: **IEEE. Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on**. 2015. p. 2210–2215.
- CAVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. Physics with arduino for beginners. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 33, n. 4, p. 4503–4503, 2011.
- COELHO, M.; AGUIRRE, L.; CORREA, M. Metodologia para representação de modelos narx polinomiais na forma de hammerstein e wiener. **Trends in Applied and Computational Mathematics**, v. 3, n. 1, p. 71–80, 2002.
- DORF RICHARD C E BISHOP, R. H. **Sistemas de controle Moderno**. ADDISON WESLEY LONGMAN, a Pearson Education company, 1998.
- GANESAN, V. **Internal combustion engines**. McGraw Hill Education (India) Pvt Ltd, 1995.
- GEHRIG, S. K.; STEIN, F. J. Collision avoidance for vehicle-following systems. **Intelligent Transportation Systems, IEEE Transactions on**, IEEE, v. 8, n. 2, p. 233–244, 2007.
- GEN, M.; CHENG, R. **Genetic algorithms and engineering optimization**. John Wiley & Sons, 2000.
- GOODSELL, D. **DICTIONARY OF AUTOMOTIVE ENGINEERING**. 1995.
- GOTO, Y.; FURUSAWA, H.; ARAKI, M.; FUKUDA, T. A safe traffic speed control in ahs. In: **IEEE. Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on**. 1999. p. 459–464.

- HUO, H.; LI, J. Cruise control based on gain scheduling pi diesel engine speed controller. In: **IEEE International Vehicle Electronics Conference (1999: Changchun China). Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'99)**. 1999.
- JEBARI, K.; MADIIFI, M. Selection methods for genetic algorithms. **International Journal of Emerging Sciences**, Springfield Publishing Corporation, v. 3, n. 4, p. 333–344, 2013.
- KILIÇ, İ.; YAZICI, A.; YILDIZ, Ö.; ÖZÇELİKORS, M.; ONDOĞAN, A. Intelligent adaptive cruise control system design and implementation. In: **IEEE. System of Systems Engineering Conference (SoSE), 2015 10th**. 2015. p. 232–237.
- KIM, H.; YI, K. Combined throttle and brake control for vehicle cruise control: A model free approach. In: **IEEE. Intelligent Vehicles Symposium (IV), 2013 IEEE**. 2013. p. 859–864.
- KO, R. **What affects the aerodynamics of a car?** 2016. Disponível em <<https://goo.gl/GezB0Q>>. Acesso em: 01 mai. 2016.
- MACNEILL, P.; MILLER, R. A new technology for a cruise control system. In: **IEEE. Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th**. 2003. v. 5, p. 3341–3345.
- MAGALHAES-MENDES, J. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. **WSEAS transactions on computers**, v. 12, n. 4, p. 164–173, 2013.
- MORAND, A.; MOREAU, X.; MELCHIOR, P.; MOZE, M.; GUILLEMARD, F. Crone cruise control system. **IEEE**, 2015.
- MORIZANE, H.; TAKENAGA, H.; KOBAYASHI, Y.; NAKAMURA, K. Cut-in vehicle recognition system. In: **IEEE. Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on**. 1999. p. 976–980.
- OGATA, K. **Engenharia de Controle Moderno**. Pearson Prentice Hall, 2013.
- OSMAN, K.; RAHMAT, M. F.; AHMAD, M. A. Modelling and controller design for a cruise control system. In: **IEEE. Signal Processing & Its Applications, 2009. CSPA 2009. 5th International Colloquium on**. 2009. p. 254–258.
- OTHERS and. **Speed control for automotive vehicles**. Google Patents, maio 12 1970. US Patent 3,511,329.
- PALM, W. J. **Control systems engineering**. John Wiley & Sons, Inc., 1988.
- PANANURAK, W.; THANOK, S.; PARNICHKUN, M. Adaptive cruise control for an intelligent vehicle. In: **IEEE. Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on**. 2009. p. 1794–1799.
- PHILLIPS, C. L.; HAVOR, R. D. **Feedback control systems**. Simon & Schuster, 1995.
- QIU, C. A design of automobile cruise control system based on fuzzy pid. In: **IEEE. Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on**. 2014. v. 1, p. 450–452.

SHARAPOV, R. **Genetic algorithms: basic ideas, variants and analysis**. INTECH Open Access Publisher, 2007.

SILVA, W. G. da; ALVARENGA, B.; ACARNLEY, P. P.; FINCH, J. W. Aplicação on-line de algoritmo genético na otimização de um regulador de velocidade fuzzy para acionamentos elétricos. 2011.

TEETOR, R. R. **Speed control device for resisting operation of the accelerator**. Google Patents, ago. 22 1950. US Patent 2,519,859.

WARD, D.; BERTRAM, T.; HILLER, M. Vehicle dynamics simulation for the development of an extended adaptive cruise control. In: IEEE. **Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on**. 1999. p. 730–735.

WISNER, D. A. **Speed control for motor vehicles**. Google Patents, mar. 16 1971. US Patent 3,570,622.

YOURDICTIONARY. **Carburetor**. 2016. Disponível em <<http://goo.gl/6qnNxG>>. Acesso em: 29 abr. 2016.

ANEXO A

Contents

- Função que gera o sinal de identificação
- Função sinal_identificação
- Exemplo

Função que gera o sinal de identificação

Função para criar sinal de identificação ou de validação. Não é um sinal puramente aleatório e é específico para este trabalho.

```
function sinal = sinal_identificacao(min,max,grau_dif,tempo_min,...
tempo_max,amostras,ciclos_repetidos)
```

Função sinal_identificação

Parâmetros:

- min: grau de abertura mínimo
- max: grau de abertura máximo
- grau_dif: diferencial máximo
- tempo_min: mínimo número de ciclos repetidos
- tempo_max: máximo número de ciclos repetidos
- amostras: número de amostras
- ciclos_repetidos: ciclos repetidos exclusivamente para rampas

Cria o vetor do sinal de entrada

```
sinal = zeros(amostras,1);
% contadores auxiliares
i = 1; % contador de iterações
j = 1; % contador da variável 'diferencial'
contador = 0; % contador de amostras
```

```

% divide código em frequência baixa ou frequência alta
frequencia_alta = false;
% intervalos de tempo para frequência alta
fa_vetor_tempo = (randi([tempo_min/10 tempo_max/10],amostras,1));
fa_intervalo = [min max];
% intervalos entre velocidade baixa e alta, para frequência baixa
intervalo_baixa = [min round((max + min)/2)];
intervalo_alta = [(1 + round((max + min)/2)) max];
% diminui em 20 (2s) ciclos o tempo máximo em alta velocidade
vetor_tempo_alta = (randi([tempo_min tempo_max-20],amostras,1));
vetor_tempo_baixa = (randi([tempo_min tempo_max],amostras,1));
% número gerado para a primeira iteração
aux1 = randi(intervalo_alta,1,1);
% começa com abertura acima da média no carburador
for j=1:vetor_tempo_alta(i)
sinal(contador + j) = aux1;
end
% atualiza contadores do vetor de entrada
contador = contador + j + 1;
i = i+1;
while contador < amostras
% a partir daqui faz as rampas, metade do teste com mudanças
% lentas, metade com mudanças mais rápidas
% se ainda está na parte lenta (antes da metade de amostras), define
% variáveis de abertura do carburador e de tempo
if frequencia_alta ~= true
% se i é par, define rampa ascendente
if rem(i,2) == 0
min = intervalo_alta(1);
max = intervalo_alta(2);
vetor_tempo = vetor_tempo_alta;
else
% do contrário, define rampa descendente
min = intervalo_baixa(1);
max = intervalo_baixa(2);

```

```

vetor_tempo = vetor_tempo_baixa;
end
% se o contador passou da metade do número de amostras, as mudanças
% devem ocorrer com maior frequência
if contador > amostras/2
frequencia_alta = true;
ciclos_repetidos = 2;
min = fa_intervalo(1);
max = fa_intervalo(2);
vetor_tempo = fa_vetor_tempo;
contador = amostras/2 + 1;
end
end
%      cria duas variáveis aleatórias para comparação
aux2 = randi([min max],1,1);
aux = [aux1 aux2];
%      faz a diferença das variáveis
diferencial = diff(aux);
%      faz a rampa positiva
if(diferencial > grau_dif)
for j = 1:diferencial
for m = 1:ciclos_repetidos
sinal(contador + j + m - 2) = aux1 + j;
end
contador = contador + m - 1;
end
contador = contador + diferencial - 1;
%      faz a rampa negativa
elseif((diferencial < grau_dif)&&(diferencial < 0))
for j = 1:abs(diferencial)
for m = 1:ciclos_repetidos
sinal(contador + j + m - 2) = aux1 - j;
end
contador = contador + m - 1;
end

```

```

contador = contador + abs(diferencial) - 1;
else
sinal(contador) = aux2;
end
%     repete o sinal
if aux2 < 110 % se o sinal é menor que 110°
for j=1:vetor_tempo(i) % repete o sinal com o vetor de tempo
sinal(contador + j) = aux2;
end
else % do contrário repete o vetor com o tempo mínimo
for j=1:tempo_min
sinal(contador + j) = aux2;
end
end
% atualiza contador de amostras
contador = contador + j + 1;
% atualiza contador de iterações
i = i+1;
% salva a variável auxiliar gerada para gerar a rampa nesta iteração
aux1 = aux2;
end
% conversão para 8 bits
sinal = uint8(sinal);
% salva o sinal até o número de amostras desejado
sinal = sinal(1:amostras);
end

```

Exemplo

Exemplo de aplicação. O vetor pode ser gerado em C com ajuda do MATLAB coder.

```

% clc
% close all
% clear all
%
% min = 90; % angulo maximo

```

```
% max = 105; % angulo minimo
% grau_dif = 1; % diferencial
% tempo_min = 50; %ciclos de 100 ms
% tempo_max = 120; %ciclos de 100 ms
% amostras = 1800; % metade baixa, metade alta frequencia
% ciclos_repetidos = 10; % baixa frequencia
%
% u = sinal_identificacao(min,max,grau_dif,tempo_min,tempo_max,amostras)
% plot(u) % faz o gráfico do sinal
% grid
```

ANEXO B

- AG.m

Script de chamada do AG acelerado por NM

Adaptado do script do professor Elder Oroski, UTFPR-Curitiba

Contents

- Início do script
- Início do AG
- Definição dos setpoints de velocidade
- Definição dos períodos de tempo
- Vetor de entrada (tempo,setpoints)
- Geração da população Inicial
- Avalia a população inicial de soluções:
- Loop Principal
- Operadores genéticos
- Mostra a geração e o melhor fitness
- Simulação do melhor indivíduo no simulink

Início do script

Comandos para inicialização do script

```
clc; % limpa janela de comando
clear; % limpa variáveis
close all; % fecha janelas do MATLAB
warning off; % desliga avisos
load('workspace.mat') % carrega espaço de trabalho
```

Início do AG

Define parâmetros do AG

```

gmax      = 45; % Número máximo de gerações;
VinfMut   = 5; % Valor inicial da taxa de mutação;
VsupMut   = 25; % Valor final da taxa de mutação
Tmut      = linspace(VinfMut,VsupMut,gmax); % Taxa de mutação;
VinfCru   = 80; % Valor inicial da taxa de cruzamento;
VsupCru   = 60; % Valor final da taxa de cruzamento;
Tcru      = linspace(VinfCru,VsupCru,gmax); % Taxa de cruzamento;
Erromax   = 1E-5;

```

Definição dos setpoints de velocidade

```

v0 = 4;
v1 = 6;
v2 = 5;
v3 = 6;
v4 = 4.5;

```

Definição dos períodos de tempo

```

t_subida = 6;
t_descida = 4;
t_v1 = 6;
t_v2 = 14;
t_v3 = 10;
t_v4 = 10;

tt = t_subida + t_v1 + t_descida + t_v2 + t_subida + t_v3 + t_descida .
+ t_v4; % tempo total
periodo = 0.1; % define período de amostragem do sensor

```

Vetor de entrada (tempo,setpoints)

```

u1 = linspace(v0,v1,t_subida/periodo); % rampa ascendente
u2 = linspace(v1,v1,t_v1/periodo); % constante
u3 = linspace(v1,v2,t_descida/periodo); % rampa descendente
u4 = linspace(v2,v2,t_v2/periodo); % constante

```

```

u5 = linspace(v2,v3,t_subida/periodo); % rampa ascendente
u6 = linspace(v3,v3,t_v3/periodo); % constante
u7 = linspace(v3,v4,t_descida/periodo); % rampa descendente
u8 = linspace(v4,v4,t_v4/periodo); % constante

t = linspace(0,tt,tt/periodo); % cria vetor de tempo total
t = t';
u = [u1 u2 u3 u4 u5 u6 u7 u8]';
entrada = [t, u]; % vetor de entrada

```

Geração da população Inicial

```

Pop_size = 55; % Número de individuos na populacao do AG;
Cromossomo = struct('parametro', {}, 'fitness', {}); % Cria estrutura do
... cromossomo

for m = 1:Pop_size % Gera a população inicial

Cromossomo(m).parametro = randn(1,3) + 5; % Determina valores
... aleatórios de Kp, Ki e Kd para cada membro da população
Cromossomo(m).fitness = inf; % Fitness: Valor Inicial-> +oo;
end

```

Avalia a população inicial de soluções

```

disp('inicial')
tic % inicia o relógio
Cromossomo_fit = Fitness(Cromossomo,entrada,model); % Calcula o fitness
[best_one,worst_one,fit_mean] = Best(Cromossomo_fit); % Salva o melhor
... indivíduo

```

Loop Principal

Sequência: seleção (torneio), cruzamento e mutação.

```

Cont_Ger = 0; %Contador de gerações;
disp('loop')
while Cont_Ger < gmax && best_one.fitness > Erromax % condições de para
... contador > n° máximo de gerações, melhor fitness < erro máximo

Cont_Ger = Cont_Ger + 1; % acrescenta contador
fitness_min(Cont_Ger) = best_one.fitness; % salva o melhor fitness
fitness_med(Cont_Ger) = fit_mean; % salva o fitness médio
Tc = Tcru(1:Cont_Ger); % atualiza a taxa de cruzamento
Tm = Tmut(1:Cont_Ger); % atualiza a taxa de mutação

```

Operadores genéticos

```

% Seleção de progenitores baseada em Torneio
% Cruzamento está dentro da função selection, após o torneio
Cromossomo_sel = selection(Cromossomo_fit,Tcru,Cont_Ger);

%Operador de mutação
Cromossomo_mut = mutacao(Cont_Ger,Cromossomo_sel,Tmut);

% Atualiza o fitness dos cromossomos
Cromossomo_fit = Fitness(Cromossomo_mut,entrada,model);

% Elitismo, salva o melhor cromossomo da geração na primeira posição
Cromossomo_fit(1) = best_one;

% Salva o melhor, o pior e a média do fitness desta geração
[best_one,worst_one,fit_mean] = Best(Cromossomo_fit);

```

Mostra a geração e o melhor fitness

```

clc;
disp('Geração: ')
disp(Cont_Ger);
disp('Fitness: ');

```

```

disp(best_one.fitness);
%   graficos(fitness_min,fitness_med,best_one,Tc,Tm); % faz gráficos
%   pause(0.000001); % espera a próximo iteração

end

toc % inicia o relógio
Kp2 = best_one.parametro(1); % Kp
Ki2 = best_one.parametro(2); % Ki
Kd2 = best_one.parametro(3); % Kd

```

Simulação do melhor indivíduo no SIMULINK

```

options = simset('SrcWorkspace','current'); % opções de simulação
sim('plot_best',[],options); % abre arquivo plot_best
plot(yout_best.Data) % gráfico da resposta
hold
plot(entrada(:,2)) % gráfico da entrada
figure
plot(controller_signal_best) % sinal do controlador PID

save best_one % salva melhor indivíduo na área de trabalho

```

●Fitness.m

Contents

- Função Fitness
- Simulação pelo simulink e fitness
- Parâmetros do PID
- Simulação do sistema no simulink

Função Fitness

Simula o sistema e realiza o cálculo do erro médio quadrático

```
function Cromossomo_out = Fitness(Cromossomo_in,input,modelo)
```

Simulação pelo SIMULINK e *Fitness*

Parâmetros: * Cromossomo_in: Cromossomos da geração * input: entrada do sistema * modelo: modelo de Hammerstein-Wiener

```
Cromossomo_out = Cromossomo_in; % Cria cromossomos de saída
NumCrom = size(Cromossomo_in(:,1)); % Determina o número de cromossomos
model_sim = modelo; % atribui o modelo a model_sim para simulação

for i = 1:NumCrom
```

Parâmetros do PID

```
Kp = Cromossomo_in(i).parametro(1); % ganho proporcional
Ki = Cromossomo_in(i).parametro(2); % ganho integral
Kd = Cromossomo_in(i).parametro(3); % ganho derivativo
```

Simulação do sistema no simulink

```
options = simset('SrcWorkspace','current'); % opções de simulação
sim('simul',[],options); % abre o simulink e simula o sistema
erro = (yout.Data - input(:,2)); % calcula o erro entre setpoint e
... saída da simulação
SQR = sum(erro.*erro);
N = size(yout.Data,1);
MSE = SQR/N; % cálculo do erro médio quadrático (avaliação)
Cromossomo_out(i).fitness = MSE; % atribui o valor ao cromossomo

end
```

•Best.m

- Função Best
- Elitismo
- Cria estruturas para salvar os parâmetros retornados

Função Best

salva os melhores parâmetros do controlador, a média e os piores parâmetros

```
function [best_one, worst_one, fit_mean] = Best(Cromossomo_fit)
```

Elitismo

Parâmetros: * Cromossomo_fit: cromossomos desta geração

Cria estruturas para salvar os parâmetros retornados

```
best_one = struct('parametro', {}, 'fitness', {});
worst_one = struct('parametro', {}, 'fitness', {});
best_one(1).fitness = inf; % atribui infinito para o melhor fitness
worst_one(1).fitness = 0; % atribui 0 ao pior fitness

for i = 1:size(Cromossomo_fit(:,1),1) % laço de 1 até o tamanho da geração
%     Se o fitness do cromossomo for melhor do que o atual melhor
if Cromossomo_fit(i).fitness < best_one(1).fitness
%         atualiza o melhor cromossomo
best_one(1) = Cromossomo_fit(i);
end
%     Se o fitness do cromossomo for pior do que o atual pior
if Cromossomo_fit(i).fitness > worst_one(1).fitness
%         atualiza o pior cromosso
worst_one(1) = Cromossomo_fit(i);
end
%     Vetor utilizado para calcular a média dos fitness
aux_mean(i) = Cromossomo_fit(i).fitness;
end
% Cálculo da média dos fitness
fit_mean = mean(aux_mean);
```

- Selection.m

Contents

- Função de seleção
- Seleção e cruzamento
- Torneio
- Cruzamento

Função de seleção

realiza seleção por torneio e logo em seguida cruzamento entre genes do progenitor selecionado e o um cromossomo da população

```
function Cromossomo_sel = selection(Cromossomo_fit,Tcru,Cont_Ger)
```

Seleção e cruzamento

Parâmetros: * Cromossomo_fit: cromossomos * Tcru: taxa de cruzamento (varia ao longo das gerações) * Cont_Ger: contador das gerações

```
dim_pop = size(Cromossomo_fit,2); % Dimensão da População.
dim_gen = size(Cromossomo_fit(1).parametro,2); % Quantidade de parâmetros
```

```
for i = 1:dim_pop
```

Torneio

```
pai = torneio(Cromossomo_fit); % pai selecionado por torneio
mae = Cromossomo_fit(i); % cromossomo da iteração
```

Cruzamento

```
for j = 1:dim_gen % laço para cruzamento dos parâmetros
tau1 = (100)*rand(1); % "dado" para realização ou não da
```

```

... recombinação dos genes (apenas valores positivos)
if tau1 <= Tcru(Cont_Ger) % Se tau <= Tcru ocorre o cruzamento.
% Se o rand retornar um valor maior que 0.5 usa-se o gene
... do cromossomo pai
if rand(1,1)>0.5
Crom(j) = pai.parametro(j); % Se menor que 0.5
... usa-se o gene do cromossomo pai
else
Crom(j) = mae.parametro(j); % Do contrário usa-se o
... gene da mãe
end
else % se não ocorrer cruzamento, usa-se o gene da mãe
Crom(j) = mae.parametro(j);
end
end
Cromossomo_sel(i).parametro = Crom; % salva os valores do cromossomo

end

```

- Mutacao.m

Contents

- Função mutação
- Mutaçã

Função mutação

realiza mutação nos genes da população

```
function Cromossomo_mut = mutacao(ContGer,Cromossomo_sel,Tmut)
```

Mutaçã

Parâmetros: * ContGet: contador da geração * Cromossomo_sel: Cromossomos que vieram de seleção * Tmut: taxa de mutação da geração

```

Cromossomo_mut = Cromossomo_sel; % Cria cromossomos com mesma estrutura
... dos cromossomos de entrada
dim_pop = size(Cromossomo_sel,2); % Dimensão da população
dim_gen = size(Cromossomo_sel(1).parametro,2); % Dimensão dos genes

for i = 2:dim_pop % laço até o fim da população
for j = 1:dim_gen % laço para os genes do cromossomo
tau = (100)*rand(1); % número aleatório uniformemente distribuído
if tau <= Tmut % se a taxa for maior que o número aleatório
Cromossomo_mut(i).parametro(j) = randn(1,1) + 5; % atribui
... um novo número aleatório ao gene
else
% não há mutação
Cromossomo_mut(i).parametro(j) = Cromossomo_sel(i).parametro(j);
end
end
end
end

```

•Torneio.m

```

function winner = torneio(pop_in)
% Método de seleção entre os indivíduos.
%-----
cromossomo = pop_in;
pop_size = length(pop_in); % Tamanho da população

%-----
competidor = randi([1,pop_size],1,3); % Cria um vetor 1x3 com valores a
% Compara o fitness dos três competidores entre si
if cromossomo(competidor(1)).fitness < cromossomo(competidor(2)).fitnes
if cromossomo(competidor(1)).fitness < cromossomo(competidor(3)).fitnes
winner = cromossomo(competidor(1));
else
winner = cromossomo(competidor(3));
end
else

```

```
if cromossomo(competidor(2)).fitness < cromossomo(competidor(3)).fitnes
winner = cromossomo(competidor(2));
else
winner = cromossomo(competidor(3));
end
end
%-----
```

ANEXO C

```

#include <Servo.h> //biblioteca de Servomotores
#include <avr/pgmspace.h> // biblioteca de microcontroladores de
    arquitetura AVR, programação na memória flash
Servo myservo; // cria objeto do servo

float Kp = 7.3896, Ki = 8.8942, Kd = 7.6883; // Ganhos das ações P, I
    e D
int N = 5; // fator de filtragem de D
float T_ms = 0.1; // Período de Amostragem
float ek1 = 0, Ik1 = 0, Dk1 = 0; //Auxiliares de etapas anteriores
float ek; // variável de erro atual
float x; // variável resposta da função PID
float P,D; //ações P, I, D
float I; // integrador
float PID; //ação PID = P + I + D
float Vell = 0; // Velocidade anterior
int val; // variable to read the value from the analog pin
char a = 0; // variavel de estado
float vel = 0; // velocidade do carro
volatile int contador = 0; // contador da interrupção
const float pi = 3.14; // pi
int c = 0; // variavel auxiliar
int repetir = 0; //variavel auxiliar
float setpoint = 0; //variavel auxiliar

//vetor de setpoints, salvos na memória flash do microcontrolador,
    vetor gerado pelo MATLAB
static const float uv1[600] PROGMEM = { 4.0F, 4.0339F, 4.0678F,
    4.1017F, 4.1356F,
4.1695F, 4.2034F, 4.2373F, 4.2712F, 4.3051F, 4.339F, 4.3729F, 4.4068F
    ,
4.4407F, 4.4746F, 4.5085F, 4.5424F, 4.5763F, 4.6102F, 4.6441F, 4.678F
    ,
4.7119F, 4.7458F, 4.7797F, 4.8136F, 4.8475F, 4.8814F, 4.9153F, 4.9492
    F,

```



```

(D2)
myservo.attach(9); // habilita interrupção no pino D9. Utiliza
    interrupção do TIMER1
pinMode(A0,INPUT); // habilita leitura do pino digital A0 (apenas
    para testes)
Serial.println("Rotina de arranque");
}

int angulo = 86; // ângulo inicial e final para arranque

void loop() { // estados: 0 -> arranque, 1 -> controle
    switch (a) {
    case 0:

        detachInterrupt(0); // desabilita interrupção
        myservo.write(angulo); // atualiza a interrupção que escreve no
            servo
        vel = pi*.027*contador; // calcula a velocidade
        contador = 0; // atribui 0 ao contador
        Serial.println(angulo); // mostra o ângulo
        attachInterrupt(0, isr, CHANGE); // habilita interrupção na
            mudança

        if(vel > 3) // se a velocidade é maior que 3
            {
                a = 1; // muda o estado para 1
                Serial.println("Rotina de controle");
            }
        else
            angulo = angulo + 1; // incrementa o ângulo
            delay(100); // delay para contagem de pulsos

            break;

    case 1:
        for(repetir = 1; repetir > 0; repetir --) { // repetir a rotina de
            controle
            for(c = 0; c < 600; c++) // ~ 60 s

```

```

{
  detachInterrupt(0); // desabilita interrupção
  setpoint = pgm_read_float_near(uv1 + c); // lê valor do
    setpoint da memória flash
  vel = pi*.027*contador; // calcula a velocidade
  contador = 0; // atribui 0 ao contador
  Serial.print(vel); // mostra a velocidade
  Serial.print('\t');
  //PID
  ek = (setpoint - vel); // determinação do erro atual
  P = Kp*ek; // cálculo da ação proporcional
  I = Ki*T_ms*ek1 + Ik1; // cálculo da ação integral
  D = Kd*N*(ek-ek1)+Dk1*(1-N*T_ms); // cálculo da ação derivativa
  Ik1 = I; //somatoria do integrativo
  ek1 = ek; //armazena o erro atual para a próxima iteração
  Dk1 = D; // armazena a ação derivativa para a próxima iteração
  PID = P + I + D; // calcula o valor da ação do PID

  x = int(PID); //Ajusta o valor da ação do PID na escala
  // Evita ações maiores que 110 ou menores do que 90,
  // mas não satura o integrador
  if (x>=110){
    x = 110;
  }
  else if (x <= 90){
    x = 90;
  }
  Serial.print(x); // mostra saída do controlador
  Serial.print('\t');
  Serial.println(PID); // mostra soma do PID
  myservo.write(x); // atualiza ângulo
  attachInterrupt(0, isr, CHANGE); // habilita interrupção
  delay(100);
}
}

a = 2; // rotina final

```

```
    break;
case 2:
    Serial.println("FIM"); // mostra que a rotina de controle acabou
    delay(1000);
    myservo.write(angulo); // mantém o ângulo em 86 graus
    detachInterrupt(0); // desabilita interrupção
    break;
}
}
```