

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JOÃO ALEXANDRE DE CARVALHO SANTANA  
MATHEUS SCHLOSSER BASSO**

**DESENVOLVIMENTO DE FRAMEWORK DE PROTOTIPAGEM  
RÁPIDA PARA TESTE DE ALGORITMOS E ESTRATÉGIAS  
DE CONTROLE EM SISTEMAS VEICULARES DINÂMICOS**

**PONTA GROSSA**

**2022**

**JOÃO ALEXANDRE DE CARVALHO SANTANA  
MATHEUS SCHLOSSER BASSO**

**DESENVOLVIMENTO DE FRAMEWORK DE PROTOTIPAGEM  
RÁPIDA PARA TESTE DE ALGORITMOS E ESTRATÉGIAS  
DE CONTROLE EM SISTEMAS VEICULARES DINÂMICOS**

**DEVELOPMENT OF RAPID PROTOTYPING FRAMEWORK  
FOR TESTING ALGORITHMS AND CONTROL  
STRATEGIES IN DYNAMIC VEHICLE SYSTEMS**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel/Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Max Mauro Dias Santos

**PONTA GROSSA**

**2022**



Este Trabalho de Conclusão de Curso está licenciado sob uma Licença Creative Commons Atribuição–NãoComercial–SemDerivações 4.0 Internacional.

**JOÃO ALEXANDRE DE CARVALHO SANTANA  
MATHEUS SCHLOSSER BASSO**

**DESENVOLVIMENTO DE FRAMEWORK DE PROTOTIPAGEM RÁPIDA PARA  
TESTE DE ALGORITMOS E ESTRATÉGIAS DE CONTROLE EM SISTEMAS  
VEICULARES DINÂMICOS**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título  
de Bacharel/Bacharel em Engenharia Elétrica da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Data de aprovação: 12 de dezembro de 2022

---

Max Mauro Dias Santos  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Carlos Henrique Illa Font  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Cristhiane Gonçalves  
Doutorado  
Universidade Tecnológica Federal do Paraná

**PONTA GROSSA  
2022**

Dedicamos este trabalho às nossas  
famílias e amigos, pelos momentos  
de ausência.

## **AGRADECIMENTOS**

O presente trabalho não poderia ser finalizado sem a ajuda de diversas pessoas e instituições às quais prestamos nossos agradecimentos. Certamente, esses parágrafos não abrangem todas as pessoas que fizeram parte dessa importante fase. Portanto, desde já pedimos desculpas àquelas que não estão presentes entre estas palavras, mas elas podem estar certas que fazem parte do nosso pensamento e tem nossa gratidão.

As nossas respectivas famílias, pelo carinho, incentivo e total apoio em todos os momentos da vida, especialmente durante essa jornada que foi a graduação.

Ao nosso orientador, que nos mostrou os caminhos a serem seguidos e sempre nos incentivou a buscar resultados condizentes com a nossa capacidade.

A todos os professores e colegas do curso, especialmente aos colegas do Grupo de Sistemas Automotivos, que ajudaram de forma direta e indireta na realização deste trabalho.

“O que fazemos agora ecoa pela eternidade” (**Marcus Aurelius**)<sup>1</sup>.

“As dificuldades fortalecem a mente, assim como o trabalho o faz com o corpo”  
(**Lucius Annaeus Seneca**)<sup>2</sup>.

---

<sup>1</sup> “*What we do now echoes in eternity*” (**Marcus Aurelius**).

<sup>2</sup> “*Difficulties strengthen the mind, as labor does the body.*” (**Lucius Annaeus Seneca**).

## RESUMO

A prototipagem rápida e o desenvolvimento baseado em modelo são abordagens de engenharia que muitas vezes parecem distantes da realidade acadêmica, o presente trabalho buscou criar um *framework* a ser seguido com objetivo de facilitar o desenvolvimento de algoritmos de controle. Através de uma revisão bibliográfica para entender/reproduzir as técnicas de Model Based Design, recorreu-se da literatura sobre dinâmica veicular para propor um modelo fidedigno ao sistema real, sendo importante contemplar a movimentação lateral e longitudinal do veículo. Além do embasamento teórico é crucial para o trabalho a devida compreensão dos equipamentos empregados, como é possível com tais dispositivos simular uma planta de controle, também sendo necessário entender como devidamente estimular o modelo. Por fim os resultados obtidos cumpriram com a meta proposta, obter um modelo e estabelecer um guia que permita o teste de algoritmos de controle.

**Palavras-chave:** Prototipagem de controle rápida; Hardware em loop; Design baseado em modelo.

## ABSTRACT

Rapid prototyping and model-based development are engineering approaches that often seem distant from academic reality, the present work sought to create a *framework* to be followed in order to facilitate the development of control algorithms. Through a bibliographic review to understand/reproduce the Model Based Design techniques, the literature on vehicle dynamics was used to propose a reliable model for the real system, being important to contemplate the lateral and longitudinal movement of the vehicle. In addition to the theoretical basis, it is crucial for the work to properly understanding from the equipment used, how it is possible with such devices to simulate a control plant, and it is also necessary to understand how to properly stimulate the model. Finally, the obtained results fulfilled the proposed goal, to obtain a model and establish a guide that allows the test of control algorithms.

**Keywords:** Rapid Control Prototyping; Hardware in the Loop; Model Based Design.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Evolução de componentes puramente mecânicos à eletrônicos nos veículos . . . . .	14
Figura 2 – Etapas de desenvolvimento de produtos automotivos . . . . .	15
Figura 3 – Os diferentes domínios do sistema veicular . . . . .	16
Figura 4 – Exemplo de diagrama de bloco . . . . .	22
Figura 5 – Sistemas de Coordenadas proposto pela SAE . . . . .	23
Figura 6 – Fluxograma da interação do sistema embarcado com o subsistema automotivo . . . . .	26
Figura 7 – Esquema de funcionamento RCP . . . . .	27
Figura 8 – Esquema de funcionamento do modelo em V . . . . .	28
Figura 9 – Exemplo de MIL . . . . .	30
Figura 10 – Exemplo de SIL . . . . .	31
Figura 11 – Exemplo de PIL . . . . .	32
Figura 12 – Exemplo de HIL . . . . .	34
Figura 13 – Esquemático de simulação em tempo real . . . . .	34
Figura 14 – Bloco para modelagem do motor . . . . .	38
Figura 15 – Bloco para modelagem da força longitudinal no pneu . . . . .	38
Figura 16 – Bloco para modelagem da força contrária ao movimento longitudinal . . . . .	39
Figura 17 – Bloco para modelagem da força contrária ao movimento longitudinal . . . . .	39
Figura 18 – Bloco para modelagem da força contrária ao movimento lateral . . . . .	40
Figura 19 – Fluxograma para geração de arquivo .sdf . . . . .	41
Figura 20 – Modelo implementado em ambiente Simulink . . . . .	42
Figura 21 – Tela para configuração do integrador . . . . .	43
Figura 22 – Tela para configuração da linguagem de programação e target . . . . .	45
Figura 23 – Seleção de target . . . . .	46
Figura 24 – Tela do ControlDesk . . . . .	48
Figura 25 – Hardware DS1104 . . . . .	49
Figura 26 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	50
Figura 27 – Mapeamento do conector P1A . . . . .	50
Figura 28 – Mapeamento do conector P1B . . . . .	51
Figura 29 – Mapeamento ddo conector CP17 . . . . .	51
Figura 30 – Mapeamento ddo conector CP18 . . . . .	52
Figura 31 – Acesso aos I/O's através do hardware CLP1104 . . . . .	52
Figura 32 – LED's de stauts presentes no hardware CLP1104 . . . . .	53
Figura 33 – Relação de conectores e sinais . . . . .	53
Figura 34 – Características das entradas analógicas da placa DS1104 . . . . .	54
Figura 35 – Características das saídas analógicas da placa DS1104 . . . . .	54
Figura 36 – Características das entradas/saídas digitais da placa DS1104 . . . . .	55
Figura 37 – Características das entradas para leitura de encoder incremental da placa DS1104 . . . . .	55
Figura 38 – Diagrama de blocos da arquitetura do hardware da placa DS1104 . . . . .	56
Figura 39 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	57
Figura 40 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	57
Figura 41 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	57
Figura 42 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	58
Figura 43 – Mapeamento de entradas e saídas da placa DS1104 . . . . .	58

Figura 44 – Tela inicial da RTI Lib . . . . .	59
Figura 45 – Blocos de acesso via RTI Lib para os principais periféricos da placa DS1104 . . . . .	59
Figura 46 – Bloco para acesso ao ADC presente na DS1104 . . . . .	60
Figura 47 – Características do ADC presente na DS1104 . . . . .	60
Figura 48 – Bloco para acesso ao DAC presente na DS1104 . . . . .	60
Figura 49 – Características do DAC presente na DS1104 . . . . .	60
Figura 50 – Bloco para acesso à DI presente na DS1104 . . . . .	60
Figura 51 – Características do DI presente na DS1104 . . . . .	60
Figura 52 – Bloco para acesso à DO presente na DS1104 . . . . .	60
Figura 53 – Características do DO presente na DS1104 . . . . .	61
Figura 54 – Dispositivo dSpace CP1104/CLP1104 . . . . .	61
Figura 55 – Entradas e saídas modelo lateral . . . . .	62
Figura 56 – Entradas e saídas modelo longitudinal . . . . .	63
Figura 57 – Setup de teste 1 . . . . .	63
Figura 58 – Setup de teste 2 . . . . .	64
Figura 59 – Parcela do esquemático da bancada de teste . . . . .	65
Figura 60 – Setup de teste 3 . . . . .	65
Figura 61 – Resposta do sistema ao degrau unitário aplicado no pedal de aceleração	67
Figura 62 – Resposta do sistema ao degrau unitário aplicado no pedal de frenagem	68
Figura 63 – Desempenho do modelo com rodas livres . . . . .	69
Figura 64 – Desempenho do modelo com rodas livres em declive . . . . .	70
Figura 65 – Desempenho do modelo para caminho arbitrário 1 . . . . .	71
Figura 66 – Desempenho do modelo para caminho arbitrário 2 . . . . .	72
Figura 67 – Desempenho do modelo para caminho arbitrário 3 . . . . .	73

## LISTA DE TABELAS

Tabela 1 – Entradas e saídas do modelo lateral . . . . .	62
Tabela 2 – Entradas e saídas do modelo longitudinal . . . . .	63

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

### SIGLAS

ABS	Antilock Breaking System
ACC	Automatic Cruise Control
ADAS	Sistema Avançado de Assistência ao Motorista, do Inglês <i>Advanced Driver-Assistance System</i>
AP	Estacionamento Automático, do Inglês <i>Automated Parking</i>
BCM	Body Control Module
CAN	Controller Area Network
CAN FD	Controller Area Network FD
DSP	Digital Signal Processing
ECM	Sistema de Distribuição de Energia, do Inglês <i>Electrical Distribution System</i>
ECU	Eletronic Control Unit
EVB	Evaluation Board
FCW	Foward Collision Warning
GPS	Sistema de posicionamento global, do Inglês <i>Global Positioning System</i>
HIL	Hardware em Loop, do Inglês <i>Harware In the Loop</i>
ICE	Motor de Combustão Interna, do Inglês <i>Intern Combustion Engine</i>
IMU	Unidade de Medida Inercial, do Inglês <i>Inertial Measurement Unit</i>
IPC	Instrument Panel Cluster
ISO	International Organization for Standardization
LDW	Lane Departure Warning
LKA	Lane Keep Assist
MAaS	Mobility As a Service
MBD	Model Based Design
MIL	Modelo em Loop, do Inglês <i>Model In the Loop</i>
MIMO	Multiple Input Multiple Output
MISRA	Motor Industry Software Reliability Association
PIL	Processador em Loop, do Inglês <i>Processor In the Loop</i>
RCP	Rapid Control Prototyping
RTI	Real Time Interface
SAE	Sociedade de Engenheiros da Mobilidade, do Inglês <i>Society of Automotive Engineers</i>
SIL	Software em Loop, do Inglês <i>Software In the Loop</i>
SISO	Single Input Single Output
UTFPR	Universidade Tecnológica Federal do Paraná
VA	Veículos Autônomos, do Inglês <i>Autonomous Vehicles</i>
XIL	X em Loop, do Inglês <i>X In the Loop</i>

### ACRÔNIMOS

PCI	Placa de Circuito Impresso
RAM	Random Acess Memory
ROM	Read Only Memory

## SUMÁRIO

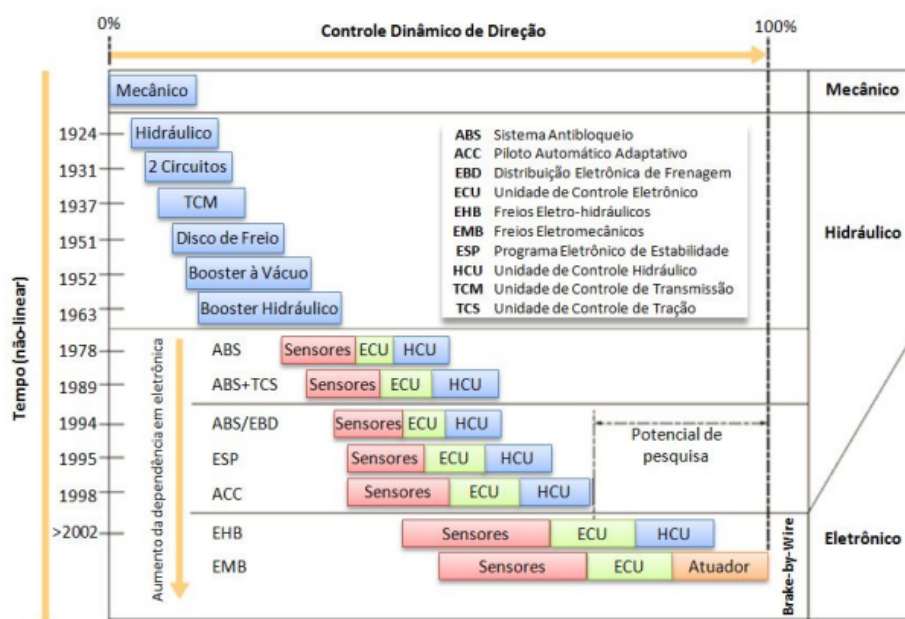
<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	TEMA	17
1.1.1	Delimitação do tema	17
1.2	PROBLEMA	18
1.3	OBJETIVO DO TRABALHO	18
1.3.1	Objetivos Secundários	18
1.4	MÉTODO DE PESQUISA	18
1.5	ORGANIZAÇÃO DO TRABALHO	19
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>20</b>
2.1	MODELAGEM MATEMÁTICA	20
2.1.1	Modelos lineares e não lineares	20
2.1.2	Representações de modelos matemáticos	21
2.1.2.1	Função de transferência	21
2.1.2.2	Espaço de estados	22
2.1.2.3	Diagrama de blocos	22
2.2	DINÂMICA VEICULAR	23
2.2.1	Modelo Longitudinal	24
2.2.2	Modelo Lateral	25
2.3	ENGENHARIA DE SISTEMAS E MODEL BASED DESIGN	25
2.4	PROTOTIPAGEM DE CONTROLE RÁPIDO	27
2.4.1	Modelo V	28
2.5	X IN THE LOOP (XIL)	29
2.5.1	Model In the Loop (MIL)	29
2.5.2	Software In the Loop (SIL)	30
2.5.3	Processor In the Loop (PIL)	32
2.5.4	Hardware In the Loop (HIL)	33
2.5.4.1	Sistema em tempo real	34
2.6	SISTEMAS AVANÇADOS DE ASSISTÊNCIA AO MOTORISTA	35
<b>3</b>	<b>MATERIAL E MÉTODOS</b>	<b>37</b>
3.1	ESTRATÉGIA UTILIZADA	37
3.1.1	Elaboração do modelo - Model in the Loop	37
3.1.1.1	Modelagem motor	38
3.1.1.2	Modelagem força longitudinal do pneu	38
3.1.1.3	Modelagem da carga	39
3.1.1.4	Modelagem da dinâmica longitudinal	39
3.1.1.5	Modelagem cinemática da dinâmica lateral	40
3.1.1.6	Implementação em Simulink	41
3.1.2	Fluxograma para geração de arquivo	41
3.1.2.1	Modelo da planta física para embarcar em hardware dSpace	41
3.1.2.2	Configuração do integrador	43
3.1.2.3	Seleção do target	44
3.2	FERRAMENTAS MATHWORKS	46
3.3	FERRAMENTAS DSPACE	47
3.3.1	Software dSpace ControlDesk	47
3.3.2	Hardware dSpace DS1104	48

3.3.2.1	Mapeamento e acesso aos I/O's . . . . .	49
3.3.2.2	Características das entradas e saídas da placa DS1104 . . . . .	53
3.3.2.3	Periféricos e seus respectivos blocos para acesso . . . . .	56
3.3.2.4	Biblioteca para Interface de Tempo-Real - RTI Lib . . . . .	58
3.3.3	Hardware dSpace CP1104/CLP1104 . . . . .	61
3.4	<b>CENÁRIOS ANALISADOS</b> . . . . .	61
3.4.1	Cenários de atuação do Modelo Lateral . . . . .	62
3.4.2	Cenários de atuação do Modelo Longitudinal . . . . .	62
3.5	<b>SETUP DE TESTES</b> . . . . .	63
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	<b>66</b>
4.1	<b>RESPOSTA AO DEGRAU</b> . . . . .	66
4.1.1	Resposta no pedal do acelerador . . . . .	66
4.1.2	Resposta no pedal de freio . . . . .	68
4.2	<b>ANÁLISE DE RODA LIVRE</b> . . . . .	68
4.3	<b>ANÁLISE DE CAMINHO ARBITRÁRIO</b> . . . . .	70
<b>5</b>	<b>CONCLUSÕES</b> . . . . .	<b>74</b>
5.1	<b>PROBLEMAS ENFRENTADOS</b> . . . . .	74
5.1.1	Entrada da força motriz . . . . .	74
5.1.2	Propor modelo que o hardware processe em tempo real . . . . .	75
5.1.3	Ferramentas computacionais . . . . .	75
5.2	<b>TRABALHOS FUTUROS</b> . . . . .	75
	<b>REFERÊNCIAS</b> . . . . .	<b>76</b>
	<b>APÊNDICE A – MODELAGEM IMPLEMENTADA</b> . . . . .	<b>79</b>
A.1	MODELAGEM DO PNEU . . . . .	80
A.2	MODELO LATERAL . . . . .	81
A.3	MODELO LONGITUDINAL . . . . .	82
	<b>APÊNDICE B – CÓDIGOS IMPLEMENTADOS NA SIMULAÇÃO</b> . . . . .	<b>83</b>
	<b>ÍNDICE REMISSIVO</b> . . . . .	<b>86</b>

## 1 INTRODUÇÃO

Ao longo da história os carros que majoritariamente só tinham elementos mecânicos, sofreram uma mudança apresentando mais componentes eletrônicos, conforme pontuado em (DAL POGGETTO, 2009). Uma vez que utilizar de recursos eletrônicos torna o sistema mais seguro, manutenível surge também a necessidade de validar e devidamente testar o funcionamento dos componentes.

**Figura 1 – Evolução de componentes puramente mecânicos à eletrônicos nos veículos**



Fonte: (DAL POGGETTO, 2009).

Conforme visto na Figura 1, a utilização de componentes eletrônicos como sensores foi popularizada no final da década de 70. No entanto realizar a validação das componentes presentes no veículo, é necessário aplicar uma metodologia que busque contemplar inúmeros cenários, sendo alguns dessas situações comuns ao sistema e outras situações na qual o sistema se encontra em um estado crítico.

Da mesma forma que gradativamente os veículos passaram a contar com mais componentes eletrônicos. Passaram a desenvolver funcionalidade que auxiliam o condutor em diversas situações, essas funções em questão são chamadas de ADAS, sendo o principal objetivo promover uma experiência segura e mais confortável ao condutor. Um dos primeiros casos de implementação dessas funções foi na Alemanha na década de 80. O experimento resultou que uma van percorreu um trajeto de 20 quilômetros

em uma rodovia vazia, sem interferência humana, conforme descrito por (DAVIDSON; SPINOULAS, 2015).

Um vez abordada a inserção de componentes eletrônicos à arquitetura do veículo. É importante entender que diversas áreas são necessárias para confecção do produto, como Engenharia de Produto, Engenharia de Manufatura, entre outras. Sendo isso melhor evidenciado na Figura 2:



**Fonte: autoria própria (2022).**

As técnicas propostas e discutidas no trabalho, possuem um maior enfoque na engenharia de produto, visando otimizar esse processo e reduzir os custos para manufatura. As etapas de marketing e pós-venda se encontram fora do escopo.

Além da implementação de elementos eletrônicos, o ambiente veicular costuma ser dividido em 5 principais domínios, de acordo com (WANG; GANESAN, 2020), sendo eles:

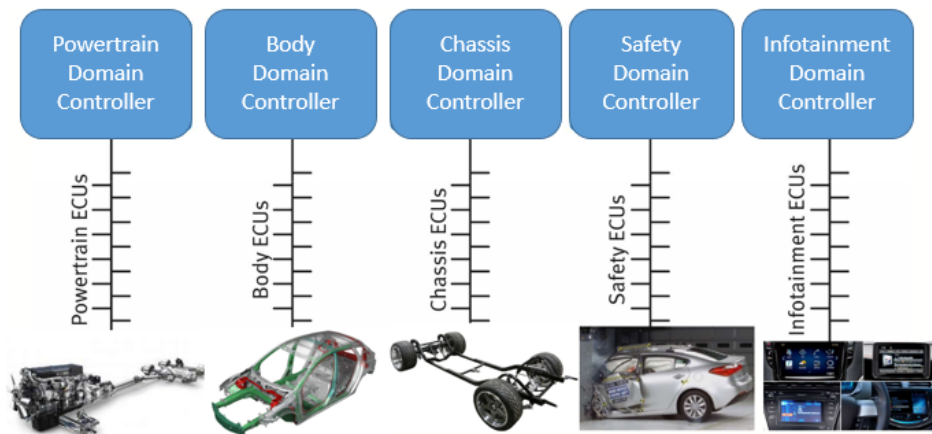
- *Powertrain*;
- *Body*;
- *Chassis*;
- *Safety*;
- *Infotainment*.

Cada respectivo domínio tem sua importância e contém inúmeras ECU's fundamentais para o devido funcionamento do veículo.

A descrição dos diferentes sistemas que compõe a arquitetura utiliza como base o trabalho proposto por Wang e Ganesan, sendo essa divisão evidenciada na Figura 3. O domínio de *Powertrain* possui os módulos relacionados com a geração e transmissão de energia mecânica, sendo o principal responsável pela locomoção do



**Figura 3 – Os diferentes domínios do sistema veicular**



**Fonte: autoria própria (2022).**

veículo, um exemplo de ECU deste domínio é a ECM que atua no controle do motor do carro.

O domínio de *Body* e *Chassis* costumam ser analisados em conjunto. Já *Body* diz respeito aos elementos estruturais do veículo, como as colunas de sustentação, controle de vibração entre outros elementos sendo o principal módulo BCM a qual é responsável por acionamento de diversos *features* do veículo, como por exemplo reconhecer a chave do carro, ligar as luzes internas, entre outras funcionalidades.

*Safety* é o domínio focado em segurança do condutor e dos passageiros do veículo visando a amenização de colisões, sendo alguns dos principais componentes deste sistema são os *airbags* e sistema de frenagem ABS. Por fim o domínio de *Infotainment* que é vinculado ao ambiente de multimídia, sendo o principal módulo o IPC, painel de instrumentos.

Além dos principais sistemas veiculares já discutidos, existem outros sistemas tão importantes quanto os citados, no entanto são considerados secundários como telemática que rege a respeito da comunicação do veículo com o ambiente externo sendo um exemplo desse sistema a utilização de GPS, outro domínio também muito importante é o sistema de distribuição de energia que contém bateria, alternador, fusíveis e outros componentes presentes no veículo.

Todos domínios, sejam eles os principais ou secundários, interagem no veículo graças a tecnologia de redes que é empregada no sistema. A topologia de rede mais comum é a CAN, no entanto alguns modelos já apresentam novas tecnologias como CAN FD , *Flexray*. A principal função da rede é assegurar que as ECU's se comuniquem

entre si e que consigam extrair as informações necessárias dos sensores presentes no veículo. Sendo fundamental que os diversos domínios troquem as devidas informações.

Ao pontuar os diferentes domínios presentes no veículo, é possível notar que o sistema em questão apresenta uma grande complexidade e projetar/implementar novas funcionalidades ou componentes se torna uma tarefa árdua. Afim de resolver esse desafio, ao longo dos anos foram propostas diferentes soluções e maneiras de realizar projetos de ECU's. A maneira mais aceita e difundida para projetar sistemas complexos, os quais integram diversos domínios presente no veículo é utilizar o *Model Based Design*. Consistindo em um desenvolvimento baseado em modelos matemáticos, além de usar de ferramentas matemáticas, para projetar sistemas. Conforme pontuado (GORDON; BIEMAN) a técnica de RCP, se popularizou na década de 90.

## 1.1 TEMA

Com passar dos anos e dos avanços tecnológicos, os produtos desenvolvidos pelos times de engenharia foram deixando de ter um caráter puramente mecânico, ou melhor de *hardware*, e passaram a incorporar elementos de *software*. Devido essa mudança o desenvolvimento de *software* se tornou uma etapa essencial, do processo da engenharia de produto.

É importante lembrar de outra etapa que possui relevância tanto quanto o desenvolvimento de *software*, essa etapa é a validação do produto. Uma vez que na etapa de validação são realizados os devidos testes para assegurar o devido funcionamento do algoritmo embarcado no produto, além de certificar que o *hardware* que contém o código é capaz de performá-lo da forma correta.

### 1.1.1 Delimitação do tema

O desenvolvimento de software e teste de algoritmos de controle ocorrem de diferentes formas a depender do seguimento em questão. O presente trabalho tem como objetivo contemplar as questões presentes na indústria automotiva, como o modelo V que consta na ISO 26262, testes de XIL que são definidos pelo padrão AUTOSAR.

## 1.2 PROBLEMA

O emprego de técnicas como MBD,RCP ainda é algo distante do ambiente acadêmico, por isso este trabalho tem como um de seus objetivos fornecer um guia para os acadêmicos que tenham interesse em aprender sobre prototipagem rápida e emprego de algoritmos de controle.

## 1.3 OBJETIVO DO TRABALHO

O presente trabalho tem como principal objetivo propor um *framework* que irá auxiliar na prototipagem rápida para estratégias de controles, também conhecido como RCP, possibilitando o emprego desses algoritmos em veículos. O sistema de um carro consiste em uma situação na qual haverá a interação de diversos domínios como por exemplo a interação entre as componentes mecânicos, eletrônicos, pneumáticos entre outros.

### 1.3.1 Objetivos Secundários

Além de desenvolver um guia para RCP, o estudo apresenta outros objetivos como:

- Estudar uma sistema de múltiplas entradas e saídas (MIMO);
- Entender como modelar o veículo e qual modelagem empregar a depender da situação problema;
- Aplicar conhecimentos multidisciplinares no desenvolvimento de um projeto.

Para realizar o *framework* recorre-se a conhecimentos que abrangem desde a área de elétrica à modelagem mecânica através de princípios físicos.

## 1.4 MÉTODO DE PESQUISA

Os seguintes capítulos contemplam a teoria necessária para propor a modelagem de um sistema, comentando a respeito das diferentes possíveis representações para o modelo, além de pontuar as ferramentas computacionais que podem auxiliar neste procedimento.

Uma vez conceituada a etapa de modelagem, foi realizado um estudo dos dispositivos disponíveis em laboratório. A finalidade desse estudo foi obter a informação se os equipamentos teriam capacidade de processar, executar os modelos e algoritmos propostos.

Após as etapas anteriores, sugeriu-se um *setup* de testes, para avaliar os modelos obtidos e posteriormente analisar os algoritmos de controle, visto que é o objetivo do trabalho. Ao submeter os modelos à alguns cenários específicos, foi possível observar se o sistema sugerido se comportar de forma similar ao sistema real.

## 1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho se organiza da seguinte forma:

- Capítulo 2, no qual ocorre a revisão da literatura necessária para entender os conceitos abordados no trabalho;
- Capítulo 3, onde são abordados os materiais e métodos utilizados;
- Capítulo 4, discussão dos resultados obtidos através dos experimentos;
- Capítulo 5, conclusão do trabalho e proposição de futuros trabalhos;
- Apêndice A, apêndice o qual detalha a modelagem empregada no trabalho;
- Apêndice B, apêndice que consta os códigos implementados na execução do trabalho.

## 2 REVISÃO DA LITERATURA

A revisão de literatura tem como objetivo abordar a modelagem matemática necessária para devida representação de um sistema automotivo e as diferentes representações desses modelos. Tratar de tópicos como dinâmica veicular, fundamentar as técnicas de prototipagem e desenvolvimento empregadas neste trabalho. Além de abordar a respeito de algumas funções ADAS , de forma a exemplificar os modelos que podem ser utilizados na aplicação de estratégias de controle em determinadas situações.

### 2.1 MODELAGEM MATEMÁTICA

O presente trabalho tem como sua principal ferramenta a representação do comportamento de sistemas dinâmicos utilizando representações matemáticas a partir de equações diferenciais, ou seja, este obtido a partir de modelagem matemática. Segundo (AGUIRRE, 2015), "a modelagem matemática é a área do conhecimento que estuda maneira de desenvolver e implementar modelos matemáticos de sistemas reais", isto é representar matematicamente o comportamento de um sistema físico. Aguirre ainda classifica a modelagem matemática em duas categorias distintas, conforme sua obtenção. A primeira categoria, chamada *modelagem caixa branca*, é obtida a partir das equações que descrevem as leis físicas que regem o sistema em questão, sendo também chamada de *modelagem fenomenológica*. A segunda categoria é chamada *modelagem caixa preta*, uma vez que não é necessário nenhum conhecimento profundo da física que envolve o sistema que deseja-se modelar, apenas estabelecer uma relação.

#### 2.1.1 Modelos lineares e não lineares

Os modelos lineares são modelos simplificados, os quais muitas vezes não contemplam questões como atrito, arrasto, elasticidade do sistema entre outras características como comentado por (LIMA, 2019). Esses modelos são eficazes somente em uma situação idealizada, na qual o sistema não sofrerá uma mudança muito além da sua condição de operação.

Já os modelos não lineares, são modelos mais similares às condições reais, sendo esses gradativamente mais utilizados para estudo/implementação de estratégias de controle. É importante salientar que o fator de tal modelo contemplar inúmeras características do sistema, gerar um código que o atenda será computacionalmente mais custoso em relação ao algoritmo para um modelo linear.

### 2.1.2 Representações de modelos matemáticos

A representação de modelos matemáticos pode acontecer de diversas formas. No presente trabalho, existe um enfoque nas representações obtidas através dos *dados de identificação* do sistema, os quais são obtidos através da medição simultânea da entrada e de sua respectiva resposta segundo (CORRÊA, 1997). Os modelos matemáticos obtidos pelo processo de identificação de sistemas podem ser divididos em duas classes sendo uma *modelos de entrada e saída* e a outra *modelos de espaços de estados*.

#### 2.1.2.1 Função de transferência

Para obter um *modelos de entrada e saída*, costuma-se recorrer às funções de transferência, as quais segundo (COUGHANOWR; KOPPEL, 1978) consistem na relação de duas variáveis no sistema estudado. Sendo as entidades em questão a variável de perturbação, também conhecida como variável de entrada e a outra denominasse variável de controle, ou variável de saída. É válido pontuar que essa representação eficaz para sistemas SISO, não sendo utilizada para sistemas MIMO.

Para o emprego da função de transferência, existem duas abordagens amplamente utilizadas. Uma forma seria modelar considerando o domínio contínuo, o domínio da Transformada de Laplace por se tratar de uma maneira de converter equações diferenciais em funções algébricas (OGATA; SEVERO, 1998).

A outra abordagem seria modelar no domínio discreto, utilizando a transformada Z. Essa estratégia costuma empregada em aplicações embarcadas, uma vez que para os processadores é conveniente trabalhar em um domínio que não seja contínuo devido as limitações do dispositivo.

### 2.1.2.2 Espaço de estados

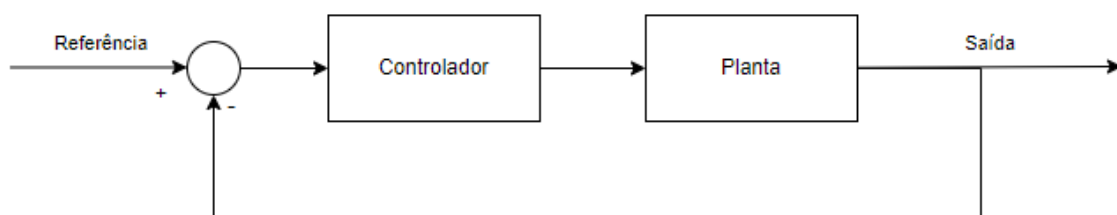
A representação em espaço de estados consiste em uma solução eficaz tanto para sistemas SISO quanto para sistemas MIMO, além disso pode ser utilizada para modelar variações internas do conjunto estudado de acordo com Ogata e Severo. A respeito da modelagem por espaço de estados, existem duas observações que devem ser consideradas, conforme Aguirre:

- O conhecimento dos vetores de estado em qualquer instante  $t_0$  especifica o estado ou condição do sistema para instante em questão;
- A representação em espaço de estados não é única, ou seja, um sistema pode ser representado por mais de um modelo no espaço de estados.

### 2.1.2.3 Diagrama de blocos

O diagrama de blocos consiste em uma representação visual obtida através das equações que regem o sistema em estudo. Sendo possível representar as funções desempenhadas por cada componente e o fluxo de sinais presente no sistema conforme Ogata e Severo.

**Figura 4 – Exemplo de diagrama de bloco**



**Fonte: autoria própria (2022).**

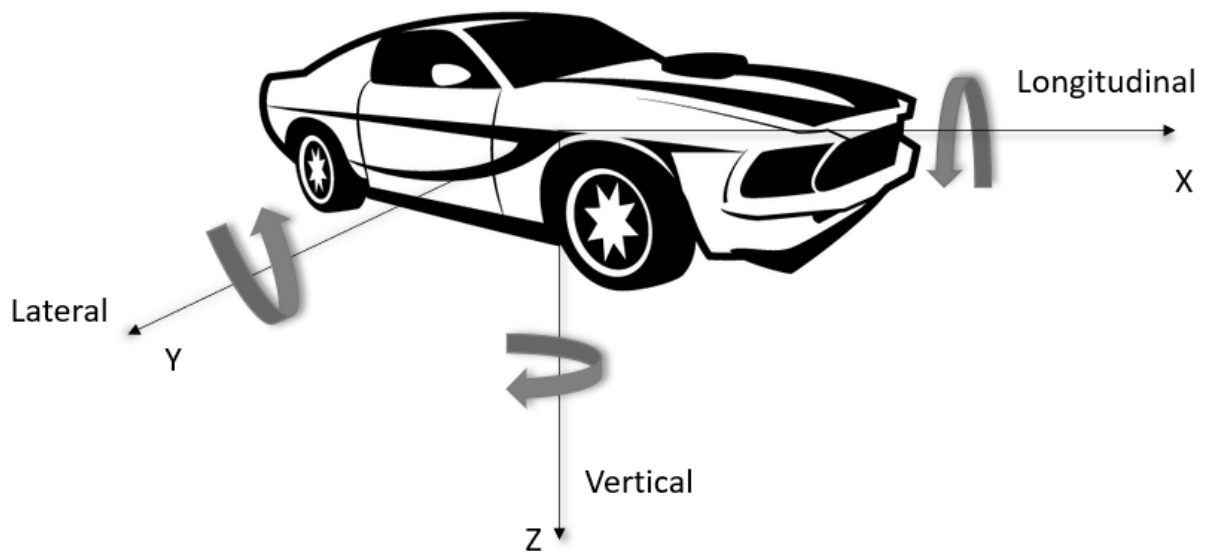
Ao analisar o comportamento das variáveis do sistema, deve-se relacionar a entrada e saída de cada bloco, seguindo o mesmo conceito empregado na seção anterior da Função de Transferência. No entanto em um sistema existem diversas variáveis que são intermediárias em relação a entrada e saída do sistema, devido esse fator estabelece-se relações entre as diversas variáveis deste sistema, através substituição

das variáveis intermediárias, nas equações que descrevem o sistema, resultando assim uma expressão a qual relacione diretamente as variáveis de interesse.

## 2.2 DINÂMICA VEICULAR

A modelagem da dinâmica veicular, é essencial para o desenvolvimento de funções ADAS, uma vez que retratam todo sistema a ser controlado por diferentes estratégias em diferentes situações. Existem diversas maneiras de retratar o sistema do carro, sendo em grande maioria retratada conforme o sistema de orientação apresentado pela Sociedade dos Engenheiros da Mobilidade (SAE).

**Figura 5 – Sistemas de Coordenadas proposto pela SAE**



Fonte: autoria própria(2022).

Como exemplificado acima, o veículo se orienta conforme os 3 eixos cartesianos (X,Y,Z). Essa orientação é mensurada através do sensor *Inertial Measurement Unit* (IMU), o qual consiste na combinação de acelerômetros, giroscópios e em alguns casos magnetômetros. Por questão de convenção admite-se:

- Eixo X terá orientação positiva, conforme dianteira do carro;
- Eixo Y aponta para esquerda, quando o veículo é visto de frente;
- Eixo Z aponta para o chão, seguindo a "regra da mão direita".

O sistema de coordenadas, é baseado na ISO 8855, onde todos os eixos estão no



sentido horário quando observados a partir da orientação positiva de cada eixo (ISO, 2013).

A modelagem de controle para sistemas veiculares costuma ser separada em duas partes, o controle lateral no eixo X e controle longitudinal no eixo Y, qual modelo será utilizado é um requisito determinado pelo tipo de função de assistência ao condutor a ser implementada.

O modelo longitudinal atende questões de propulsão e frenagem do veículo, assim controlando a aceleração e desaceleração do carro através dos pedais de freio e acelerador. O modelo lateral se responsabiliza pelo ângulo do volante.

### 2.2.1 Modelo Longitudinal

O modelo longitudinal é obtido através da segunda lei de Newton conforme a equação abaixo (HALLIDAY, 2009):

$$\sum F_{ext} = ma \quad (1)$$

Resultando que o somatório das forças externas aplicadas a um corpo é igual a massa desse objeto multiplicada pela aceleração que foi imposta ao mesmo.

No caso do modelo, a equação deverá ser reescrita da seguinte forma segundo (PACEJKA, 2012):

$$m_v \dot{v} = F_{ice} + F_{brake} + F_{air} + F_{rr} + F_{gr} + F_{int} \quad (2)$$

Sendo  $m_v$  e  $\dot{v}$  respectivamente a massa e aceleração do veículo. A força  $F_{ice}$  representa a força mecânica gerada pelo motor de combustão interna (ICE),  $F_{brake}$  força retratada pela ação de frenagem,  $F_{air}$  o atrito gerado pelo ar,  $F_{rr}$  força de rolagem,  $F_{gr}$  a interação gravitacional do carro com a terra e por fim  $F_{int}$  as forças internas do veículo.

Por motivos de simplificação pode-se considerar duas forças  $F_x F$  que representa as componentes atuantes na parte dianteira do veículo e  $F_x R$  representa a parcela traseira do carro. Sendo somente consideradas as componentes no eixo X, por que em superfícies planas as forças em Y se anulam com a força normal.

### 2.2.2 Modelo Lateral

O modelo lateral, segue o conhecido modelo bicicleta (bicycle model), que consiste em uma simplificação do modelo veicular de 4 rodas, pois admite os 2 pneus dianteiros em movimento juntos e o mesmo princípio é aplicado à parte traseira. O modelo adotado no artigo segue a simplificação proposta pelo Modelo de Ackerman. As equações para entender esse modelo encontram-se descritas em apêndice sendo elas estabelecidas de acordo com o trabalho de (OSMAN; GHOMMAM; SAAD, 2020).

A Equação (8) e Equação (9) auxiliam na descrição a respeito de cinemática do veículo, as demais equações tratam da dinâmica do sistema. Através dessas equações pode-se prever o movimento lateral do veículo, como em condição de estacionamento ou mudanças de faixa.

As variáveis  $X, Y$  descrevem a posição lateral e o ângulo de yaw do veículo,  $r$  representa a taxa de yaw,  $U_x, U_y$  são respectivamente a taxa de yaw a partir de um ponto fixado na terra, assim descrevendo às forças propulsoras. Os demais parâmetros como  $C_f, C_r$  representam a resistência à curva enquanto  $a, b$  distâncias do centro da gravidade ao eixo dianteiro e traseiro respectivamente, esses parâmetros são utilizados para estabelecer o valor das forças que são contrárias ao sentido de movimento do veículo estudado. Propriedades do veículo como massa e o momento de inércia em relação ao eixo  $Z$  são denotadas por  $m$  e  $I_z$  respectivamente.

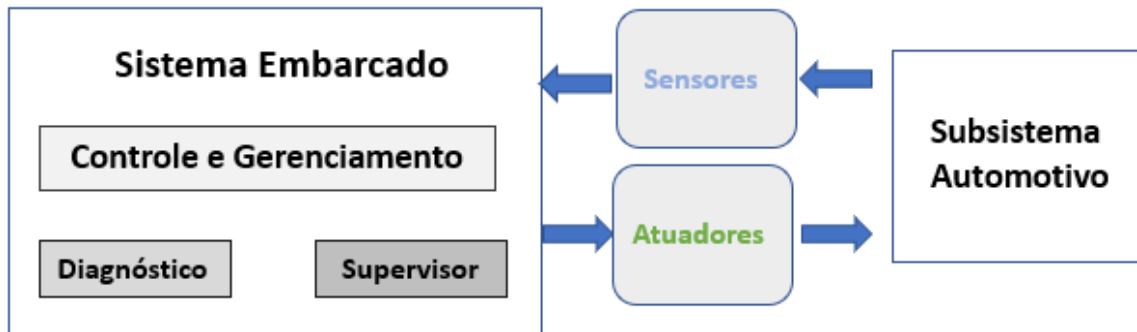
## 2.3 ENGENHARIA DE SISTEMAS E MODEL BASED DESIGN

Particionar o problema, com intuito de solucionar pequenas parcelas é uma técnica empregada em diversas áreas do conhecimento. A engenharia de sistemas e a metodologia de MBD se apresentam como alternativas eficazes para a solução de diversos problemas atuais, pois estes se configuram como interdisciplinares, complexas, envolvendo a integração de software e hardware (RAMOS; FERREIRA; BARCELÓ, 2011).

A fim de compreender o fluxo de informação entre o software embarcado nos módulos do veículo com o restante do sistema.

Conforme visto na Figura 6, o sistema embarcado irá performar funções como controle, diagnóstico, supervisão e gerenciamento. Solicitando dados do subsistema

**Figura 6 – Fluxograma da interação do sistema embarcado com o subsistema automotivo**



Fonte: autoria própria (2022).

automotivo através dos sensores e performando as devidas ações de controle através dos atuadores.

A engenharia de sistemas é a solução para esse problema, uma forma de aplicar essa estratégia é o desenvolvimento baseado em modelo, *Model Based Design* que consiste em utilizar de modelo matemáticos para melhor compreensão dos fenômenos físicos e entendimento das leis que regem os sistemas estudados, tornou-se uma ferramenta essencial para engenharia sendo empregada em diversas áreas segundo (FRANCESCHINI; MACCHIETTO, 2008).

O *Model Based Design* é fundamental no ramo automotivo, por se tratar de uma aplicação que concilia diversos domínios da física como o elétrico, mecânico, hidráulico, entre outros. Uma vez que os veículos modernos segundo (WAN; CANEDO; AL FARUQUE, 2015) utilizam de recursos de eletrônica como as ECU's e os softwares embarcados nessas unidades de controle, o processo de desenvolver esses sistemas se tornou algo muito complexo.

Devido esse fator, optou-se em adotar uma abordagem de desenvolvimento que recorre a simulações que aplicam modelos de forma a reduzir o custo com prototipagem. A ISO 26262 de acordo com (NUNES *et al.*, 2017), a qual normatiza sobre segurança funcional de sistemas elétricos e/ou eletrônicos instalados em veículos rodoviários de produção em série exemplifica como *Model Based Design* é empregado no ramo automotivo.

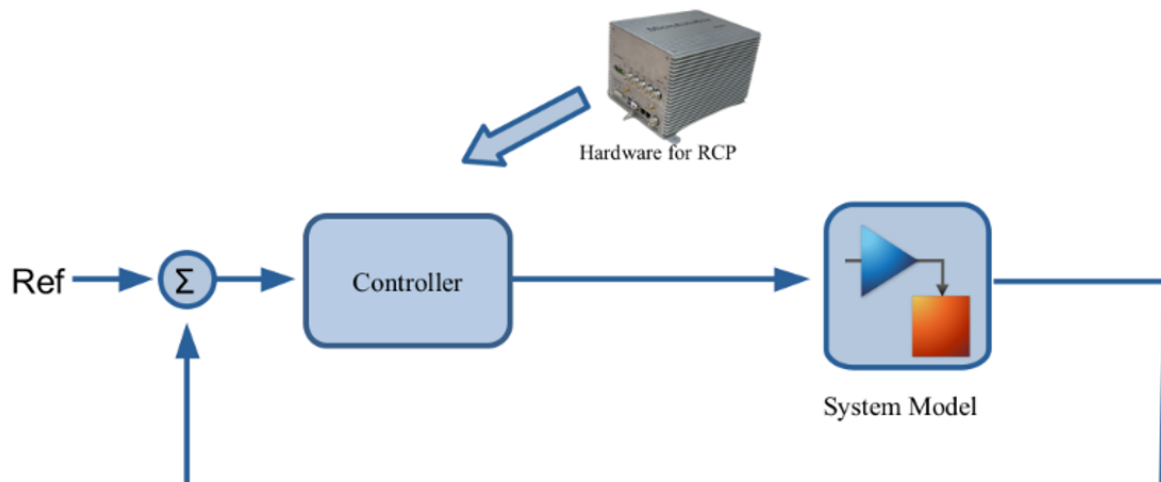
## 2.4 PROTOTIPAGEM DE CONTROLE RÁPIDO

A metodologia de desenvolvimento RCP é uma forma rápida e eficaz de programar novas estratégias de controle, sem necessariamente ter que recorrer a programação de baixo nível. Tal abordagem deixa o processo mais intuitivo para o profissional que estiver responsável pela implementação do algoritmo de controle.

As principais dificuldades enfrentadas para o desenvolvimento de algoritmos de controle segundo (BRUNO; MARCHETTO, 1986) seria simular e atender especificações do sistema. Uma forma rápida e eficaz de desenvolver o algoritmo a ser embarcado, é a utilização de ferramentas computacionais para geração de código como realizado por (DUMA *et al.*, 2007).

Duma *et al.*, utilizou o hardware DSP fabricado pela *Texas Instrument*, chamado TMS320F2812 DSP, no presente trabalho os dispositivos empregados foram outros, todavia a estratégia é a mesma. Consistindo em desenvolver um modelo em diagrama de blocos, contemplando os requisitos do sistema. O diagrama posteriormente será utilizado para gerar um código C que atuará no hardware desejado.

**Figura 7 – Esquema de funcionamento RCP**



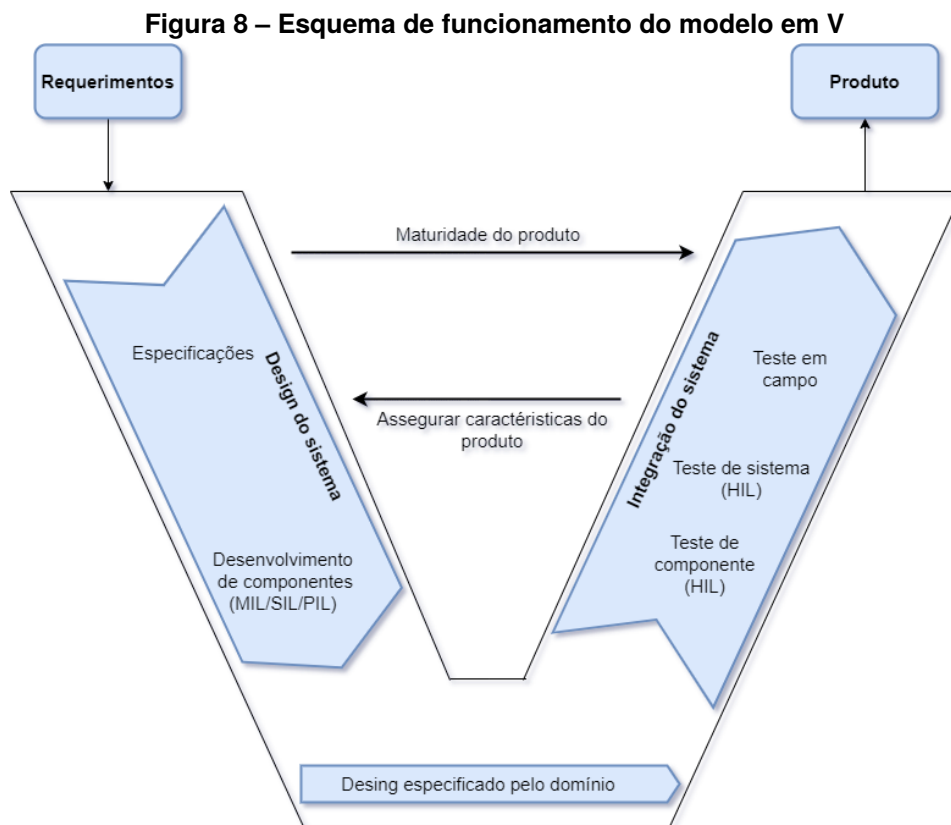
**Fonte: autoria própria (2022).**

Como visto na Figura 7, a técnica de RCP em uma perspectiva do diagrama de bloco. Será colocar o *hardware* de RCP para performar o papel do controlador.

### 2.4.1 Modelo V

O modelo de desenvolvimento em V representa uma das abordagens mais utilizadas em engenharia de requisitos, sendo muitas vezes empregada como uma estratégia de *Model Based Design* e sendo uma forma de RCP. A utilização desse sistema de desenvolvimento é amplamente difundida no meio automotivo, sendo adotada de fornecedores à montadoras.

O modelo consiste em definir o sistema no qual o produto irá atuar, após isso especificar os requisitos de *software/hardware* de funcionamento, implementar o produto, realizar testes com o mesmo, integrar o mesmo ao restante do sistema e por fim testar o sistema. O modelo em V empregado foi baseado na proposição feita por (HENKE; MICHAEL, 2022) sendo as etapas descritas conforme a Figura 8.



**Fonte: autoria própria (2022).**

As distintas etapas da Figura 8 tem como objetivo:

- Especificações, é a primeira etapa do design do sistema. Nessa fase ocorre uma análise dos requisitos que serão implementados, conforme a necessidade do escopo do projeto.

- Desenvolvimento de componentes de *software* através de técnicas como MIL, SIL, PIL. Essas técnicas são essenciais para identificação de falhas nos estágios iniciais do projeto.
- O controle de prototipagem rápida RCP, o design do sistema estará mais alinhado com o domínio do mesmo.
- Teste de componente HIL. Nessa etapa ocorre o teste do controlador através do HIL.
- Teste de sistema HIL. Nesse estágio todo o sistema será validado com técnicas de HIL
- Teste em campo, será o estágio final para confecção do produto. É a etapa que ocorre os ajustes finais ao protótipo.

## 2.5 X IN THE LOOP (XIL)

A aplicação do XIL que consiste em uma série de testes, conforme (SCHREIBER *et al.*, 2018) os quais são realizados em diferentes níveis de maturidade do produto. Sendo abordados nesse documento 4 etapas amplamente difundidas para desenvolvimento:

- Model In the Loop (MIL);
- Software In the Loop (SIL);
- Processor In the Loop (PIL);
- Hardware In the Loop (HIL) .

A técnica recebe o nome de XIL, justamente pela ideia que cada etapa a ser colocada em loop, assim possibilitando uma validação em diversos níveis do produto em questão.

### 2.5.1 Model In the Loop (MIL)

O desenvolvimento em MIL, é o primeiro estágio de um projeto em XIL, sendo caracterizado por completo como a fase de simulação da planta física e do controlador os quais não operam tempo real conforme (NEME *et al.*, 2017). O emprego MIL utiliza de modelos matemáticos os quais representam o veículo estudado, observando a interação do algoritmo com os diversos domínios do modelo, analisando os princípios físicos impostos ao veículo e a partir disso ir realizando a correção de falhas, ou

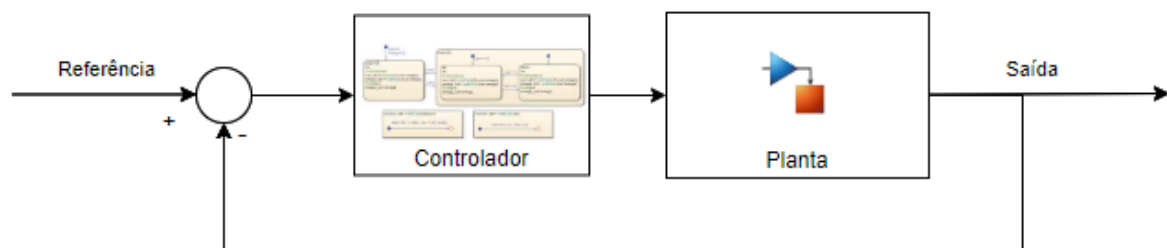
reajuste de requisitos. Esses modelos são confeccionados com auxílio de ferramentas computacionais como *Matlab*, *Simulink*, *Stateflow*. Sendo importante salientar que essa parte dos testes não considera detalhes como a quantidade de bits utilizada por cada variável e outros parâmetros afins (JAIKAMAL, 2009).

O objetivo dessa etapa é especificar o projeto do controlador e planta física. Na etapa de *Model In the Loop* será desenvolvida a estratégia e algoritmo de controle, as quais são realizadas em um mesmo ambiente computacional. Algumas das principais características do MIL são:

- Utilizar planta física e controlador em um mesmo computador, sem simular o sistema em circunstâncias que não são tempo real;
- Verifica o comportamento do controlador;
- Definição e configuração das escalas de variáveis;
- Identificação e detecção de *overflow*.

Empregar MIL é de suma importância, pois o desenvolvimento da mesma acontece em ambiente computacional, assim permitindo ver o desempenho da solução com um baixo custo uma vez que não houve a implementação em hardware.

**Figura 9 – Exemplo de MIL**



**Fonte: autoria própria(2022).**

Na Figura 9 é possível observar o emprego de:

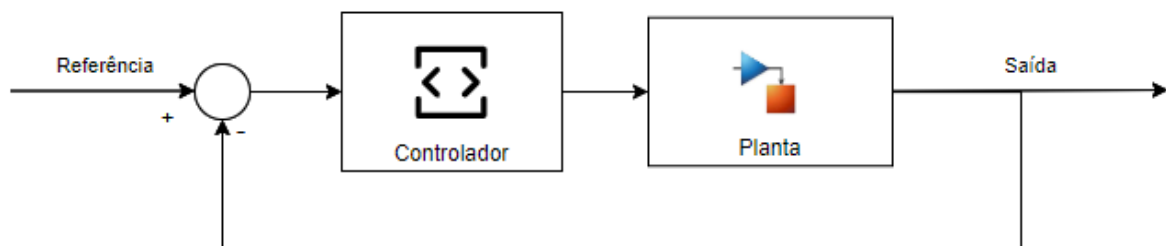
- Controlador representado por um diagrama de blocos ou por uma máquina de estados finitos a ser simulado;
- Planta é um modelo a ser simulado .

### 2.5.2 Software In the Loop (SIL)

Segundo Jaikamal a aplicação de SIL consiste em realizar testes em um software, nos primeiros estágios de desenvolvimento do produto, buscando identificar as

primeiras falhas presentes no código desenvolvido, falhas relacionadas ao *drivers* de comunicação, assim corrigindo alguns requisitos e refinando a qualidade do produto. Testes em SIL possuem extrema importância pois a implementação de software demanda um baixo custo quando comparado ao hardware, dessa forma submeter a solução à testes em SIL assegura uma qualidade e auxilia no processo de amadurecimento do produto.

**Figura 10 – Exemplo de SIL**



**Fonte: autoria própria(2022).**

Ao observar o diagrama na Figura 10, é possível notar que o modelo de SIL consiste em:

- Controlador representado por um linguagem de programação;
- Planta assim como no MIL, será um modelo simulado.

Conforme pontuado por Neme *et al.*, a etapa de SIL consiste num estágio onde a ferramenta de geração automática de código fornece a estratégia de controle previamente obtida na etapa anterior, ou seja MIL, automaticamente em código em linguagem C ou C++ permitindo que o modelo do controlador seja mais próximo do sistema real. É considerado como um estágio essencial para testar o sistema de geração de código (seja feito de maneira automática ou manual). A interação do modelo diminui um pouco com relação ao estágio MIL, todavia é possível identificar falhas no algoritmo desenvolvido.

Em aplicações automotivas, é necessário assegurar que o software esteja de acordo com as especificações da MISRA-C que é uma *guideline* desenvolvida pela organização presente no Reino Unido denominada *Motor Industry Software Reliability Association* (MISRA). Sendo essa diretriz baseada na ISO 9899, a qual normatiza sobre o desenvolvimento de software na linguagem de programação C.

Além de atender as especificações da MISRA-C, a aplicação de SIL busca analisar outras características do produto :



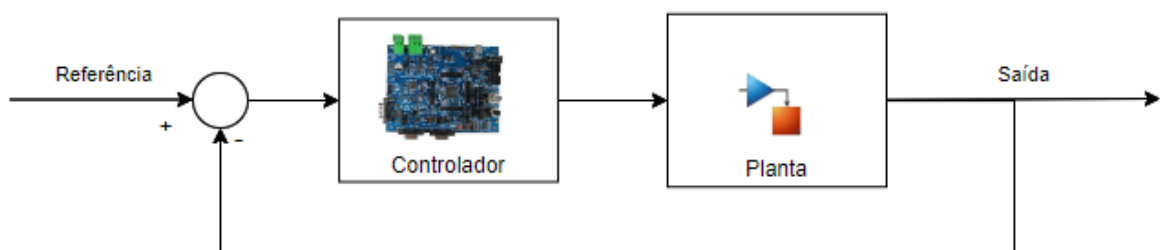
- Análise de ponto fixo;
- Análise de erros de quantização;
- Detectar falhas nos estágios iniciais do projeto, especialmente no Software de controle;
- Permitir a inclusão da funcionalidade de algoritmos de controle para o qual não existe modelo;
- Análise de saturação e *overflow*;
- Acelerar a velocidade de simulação, incluindo código compilado ao invés de modelos interpretativos.

### 2.5.3 Processor In the Loop (PIL)

No estágio de PIL, o software não será executado em ambiente simulado, ao invés disso recorre-se a uma *Evaluation Board* (EVB) a qual consiste em um microcontrolador que será utilizado em uma ECU.

A etapa de PIL, visa avaliar o desempenho do processador que irá executar a estratégia de controle. Uma vez que o código foi submetido ao processo de SIL, o próximo passo é embarcar esse código em um dispositivo conforme demonstrado por (MOTAHHIR *et al.*, 2017). Muitas vezes utiliza-se de ferramentas de geração de código automático, assim agilizando o processo de desenvolvimento do produto, além de assegurar que o processador tem capacidade de executar devidamente o algoritmo conforme as diretrizes estabelecidas pela MISRA-C.

**Figura 11 – Exemplo de PIL**



**Fonte: autoria própria(2022).**

Na Figura 11 tem-se que:

- Controlador será um EVB, a qual irá hospedar e executar o controlador;
- Planta em modelo de simulação.

As características mais importantes a serem analisadas nessa etapa do processo são:

- Análise validação de código executando com a planta em tempo real;
- Analisar o tempo de execução do código;
- Medir a quantidade de memória RAM/ ROM solicitada;
- Análise do gerenciamento de energia.

#### 2.5.4 Hardware In the Loop (HIL)

A técnica de testar com HIL consiste em utilizar de um hardware que irá simular entradas e saídas do sistema, permitindo em bancada um ambiente de teste o mais próximo possível do esperado no veículo. Segundo documentação da (DSPACE, 2011a) aplicação de HIL reduz o tempo de validação do produto, permite testar inúmeros cenários.

A implementação o HIL acontece através da conexão de um controlador com a simulação, assim representando o sistema real no qual o controlador projetado deverá atuar. Empregar HIL é fundamental pelas seguintes razões:

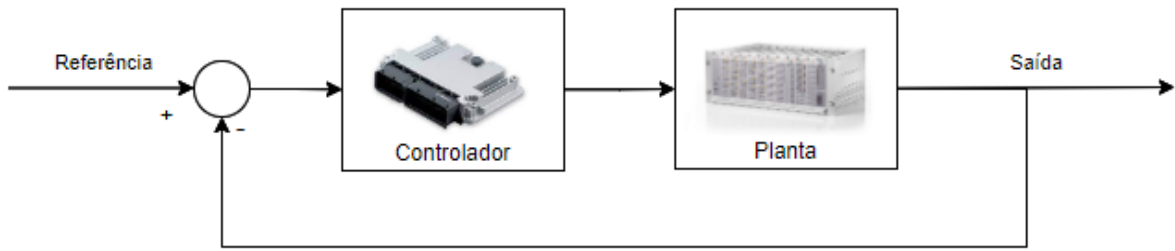
- Segurança, uma vez que planta estará em ambiente controlado assim não havendo risco de danificar o equipamento;
- Maior escopo de testes permitindo a criação de cenários ou até mesmo injeção de falhas no sistema, assim observando o comportamento do produto em situações críticas;
- Redução do tempo de desenvolvimento do produto.

As empresas do setor automotivo costumam utilizar simuladores que reproduzem o comportamento da ECU, assim permitindo ao profissional realizar os testes e verificar se a solução ali proposta irá atender os requisitos especificados na etapa inicial do modelo em V.

Na Figura 12:

- O controlador é embarcado em uma ECU que possui os módulos de entrada e saída com microcontrolador embarcado.
- A planta física será um *hardware* o qual simula as entradas e saídas do sistema real

**Figura 12 – Exemplo de HIL**



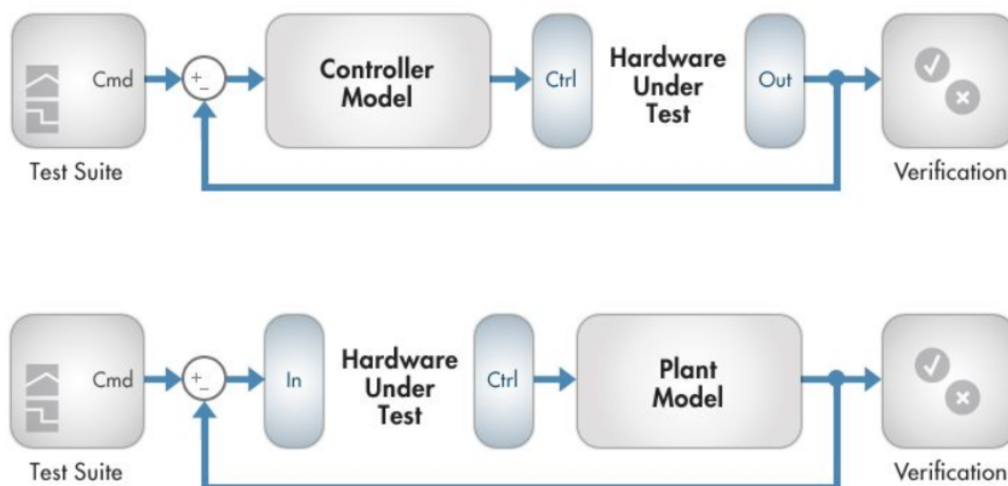
Fonte: autoria própria(2022).

#### 2.5.4.1 Sistema em tempo real

Uma das vantagens da aplicação do HIL, é verificar em tempo real o comportamento do sistema dessa forma permitindo realizar as seguintes verificações:

- Verificar se o modelo de controle será compatível com o hardware disponível para essa atividade;
- Desenvolver uma solução de maneira contínua e flexível explorando novas técnicas;
- Investigar cenários e interações do dispositivos que podem ser complexas, de alto custo ou até mesmo perigosas para expor o hardware de produção.

**Figura 13 – Esquemático de simulação em tempo real**



Fonte: autoria própria(2022).

## 2.6 SISTEMAS AVANÇADOS DE ASSISTÊNCIA AO MOTORISTA

O desenvolvimento de veículos autônomos (VA) é fomentado devido duas das principais necessidades da sociedade moderna, sendo elas sistemas de transporte mais seguros e a mobilidade enquanto serviço. A maioria dos acidentes são causados por falha humana como comentado por (CHIANG *et al.*, 2014), dessa forma minimizando o fator humano a tendência é reduzir o número de acidentes. Já as questões envolvendo MAaS, perpassam a esfera de conectividade dos VA com os diversos dispositivos presentes na rede, sejam eles outro veículos, a internet, o celular de um pedestre entre outros (HIETANEN, 2014). Essa conectividade é vital para criação de um ambiente seguro, além de viabilizar uma melhor logística nas rodovias.

Os VA se apresentam como uma solução para essas duas questões aparentemente distintas, pois as funções de assistência ao condutor (ADAS), que caracterizam o terceiro estágio de automação do veículo, conforme descrito por Davidson e Spinoulas, o qual visa tornar a experiência de dirigir mais segura. Através de ações ativas e passivas que alertaram o condutor em situação de perigo. Além de proporcionar segurança, VA se apresentam como uma solução de mobilidade urbana, uma vez que é dos pilares de *Mobility As a Service* (MAaS) (JITTRAPIROM *et al.*, 2017). Exemplo de funções ADAS que utilizam do modelo longitudinal seriam o *Automatic Cruise Control* (ACC), uma vez que o algoritmo estabelecerá uma velocidade de rodagem e tomará as devidas ações de controle para manter o veículo rodando conforme essa referência. Outro exemplo da utilização do modelo longitudinal é a função de *Foward Collision Warning* (FCW).

Em caso de funções como *Lane Keep Assist* (LKA) será utilizado somente o modelo lateral, visto que o interesse é manter o carro entre duas faixas de rodagem, outro exemplo é o *Lane Departure Warning* (LDW) na qual alerta o condutor a respeito caso o veículo esteja dentro, ou não, das devidas faixas de rodagem.

Um exemplo de função que utiliza ambos modelos, será a manobra de ultrapassagem, na qual o modelo longitudinal é necessário para o veículo de interesse se encontra com uma velocidade superior em relação a outro carro que está a frente. Caso essa condição seja verdadeira, o controle aplicado ao modelo lateral passará a atuar, primeiramente, verificando se é uma faixa que permite a manobra e caso essa condição seja atendida, o veículo se desloca lateralmente performando ultrapassagem. Outra função que contempla tanto a modelagem longitudinal quanto a lateral, é o estacionamento

automático (AP).

### 3 MATERIAL E MÉTODOS

O presente capítulo busca expor a estratégia empregada para desenvolver o *framework*, comentar a respeito das ferramentas utilizadas, desde de *software* a *hardware*. Por fim discorrer sobre os cenários analisados e exemplificar ao leitor o critério de seleção a respeito da utilização do modelo lateral ou longitudinal.

#### 3.1 ESTRATÉGIA UTILIZADA

Para o seguinte trabalho foi utilizado o dispositivos *DS1104* e *CLP1104* da empresa *dSpace*, uma vez que esses equipamentos são vastamente utilizados em aplicações que demandam *Hardware-in-the-Loop* para aplicações sensíveis a custo, já os softwares *Matlab* e *Simulink* da *MathWorks*, e o *ControlDesk* também da *dSpace*. Os softwares citados foram selecionados, pois conseguem ser integrados entre si, com a finalidade de seguir o seguinte diagrama de desenvolvimento:

##### 3.1.1 Elaboração do modelo - Model in the Loop

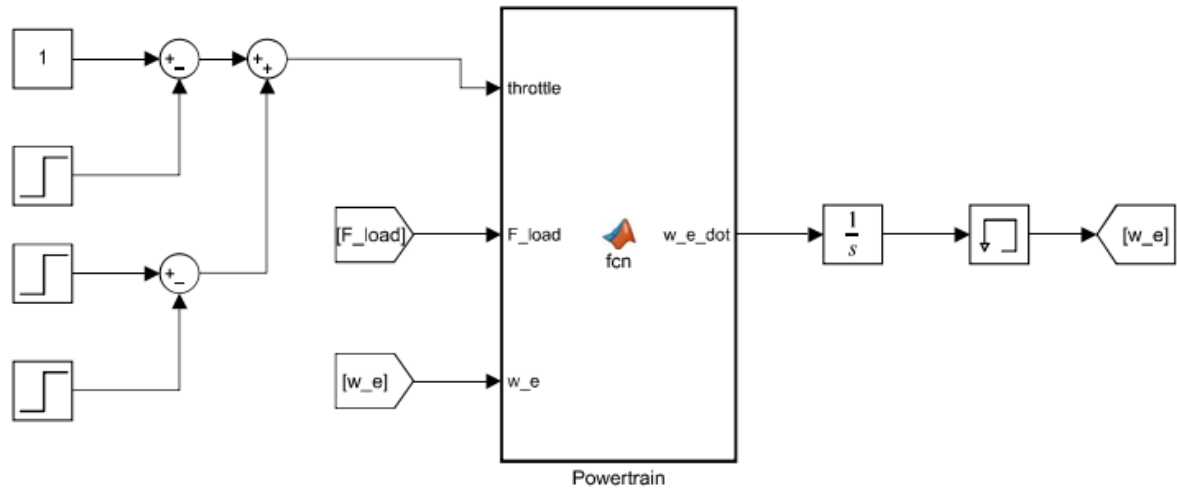
Inicialmente, foi elaborado o modelo, baseando-se nas equações cinemáticas de dinâmica lateral, dinâmica longitudinal e uma modelagem abstraída da dinâmica do motor que relaciona a força gerada nos pneus com a variação da rotação do motor, em função da porcentagem de pressão no acelerador.

Este modelo em questão foi elaborado inteiramente em ambiente *Simulink*, utilizando das equações presentes no Apêndice A (Equações implementadas no HIL), sem a necessidade de pensar na implementação em hardware, uma vez que o principal objetivo desta etapa foi conseguir elaborar um modelo que fosse capaz de integrar motor, força longitudinal do pneu, dinâmica longitudinal e modelo cinemático da dinâmica lateral do veículo.

Abaixo podem ser observados os blocos utilizados para a elaboração do modelo em questão.

### 3.1.1.1 Modelagem motor

Figura 14 – Bloco para modelagem do motor

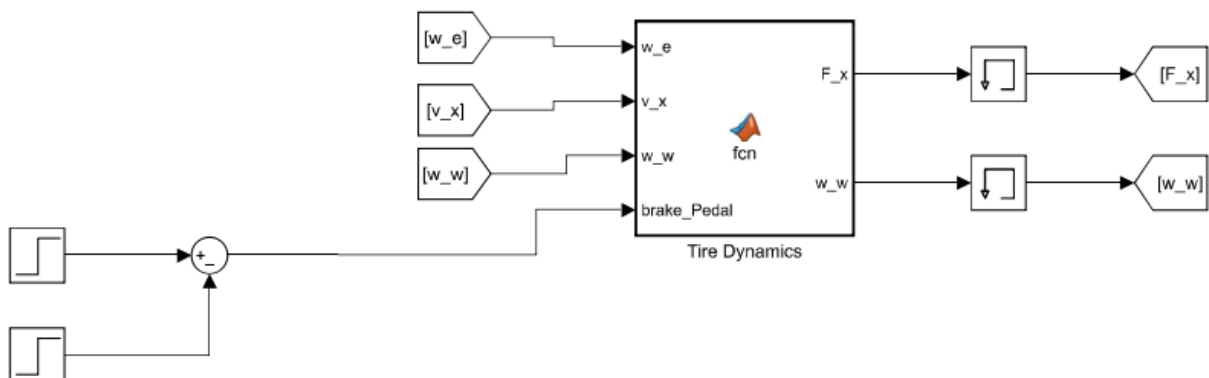


Fonte: autoria própria (2022).

O bloco de *Powertrain*, na Figura 14, recebe como entrada a porcentagem do pedal de acelerador (*throttle*) variando de 0 a 1, a força da carga atuando sobre os pneus (*F\_load*) e o valor atual da rotação do motor (*w\_e*). Sua saída é a variação da rotação do motor, a qual ao passar por um bloco de integrador resolve para um novo valor de rotação, baseando-se nas duas entradas.

### 3.1.1.2 Modelagem força longitudinal do pneu

Figura 15 – Bloco para modelagem da força longitudinal do pneu

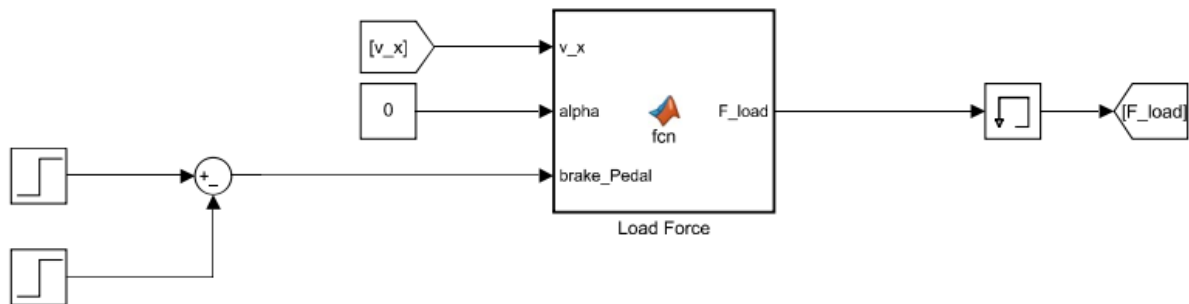


Fonte: autoria própria (2022).

A Figura 15 apresenta o bloco de *Tyre Dynamics*, que é responsável por calcular a força longitudinal do pneu e a rotação no eixo. Assim, recebe como entrada a rotação atual do motor ( $w_e$ ), a velocidade longitudinal do veículo ( $v_x$ ), a velocidade no eixo ( $w_w$ ) e a intensidade de frenagem aplicada no pedal do freio ( $brake\_Pedal$ ). Já como saída o bloco tem a força longitudinal do pneu ( $F_x$ ) e a rotação no eixo ( $w_w$ ).

### 3.1.1.3 Modelagem da carga

**Figura 16 – Bloco para modelagem da força contrária ao movimento longitudinal**

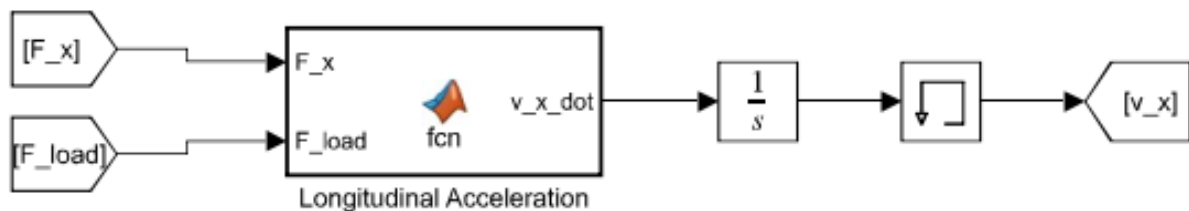


Fonte: autoria própria (2022).

Na Figura 16, o bloco *Load Force* é responsável por calcular a força contrária que irá se opor ao movimento longitudinal do veículo. Suas entradas são compostas pela velocidade longitudinal do veículo ( $v_x$ ), o ângulo de inclinação da estrada ( $\alpha$ ) e a intensidade do pedal de frenagem ( $brake\_Pedal$ ). A sua saída, como esperado, é a força de oposição ao movimento longitudinal ( $F_{load}$ ).

### 3.1.1.4 Modelagem da dinâmica longitudinal

**Figura 17 – Bloco para modelagem da força contrária ao movimento longitudinal**



Fonte: autoria própria (2022).



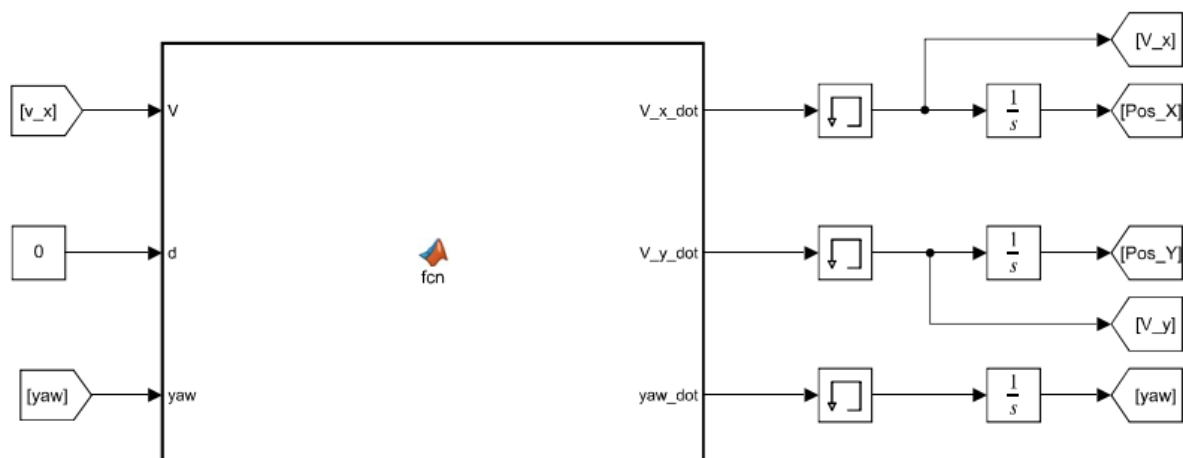
O bloco *Longitudinal Acceleration* presente na Figura 17 é responsável por implementar a dinâmica longitudinal do veículo, isto é, realizar a aceleração e desaceleração do mesmo. Para isto, o bloco recebe como entrada a força longitudinal gerada pelo pneu, bem como a força de oposição ao movimento longitudinal.

A saída deste bloco é a aceleração do veículo, lembrando que segundo a lei de Newton, a força é igual a massa vezes a aceleração. É importante ressaltar que o parâmetro da massa é estabelecido internamente ao bloco.

Como a saída deste é a aceleração, para que seja obtida a velocidade, basta integrar a aceleração, tal qual é realizado pelo bloco integrador.

### 3.1.1.5 Modelagem cinemática da dinâmica lateral

**Figura 18 – Bloco para modelagem da força contrária ao movimento lateral**



**Fonte: autoria própria (2022).**

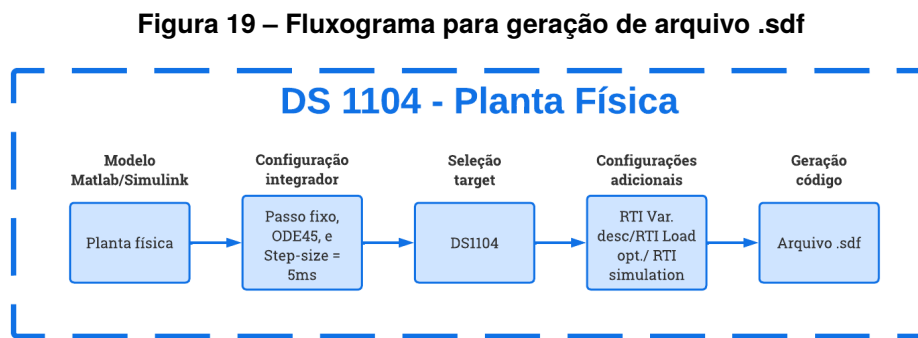
Finalmente, na Figura 18, é possível ver o bloco responsável por realizar a implementação do modelo cinemático do veículo, assim recebendo como entrada a velocidade produzida pelo bloco de dinâmica longitudinal ( $v$ ), o ângulo de esterçamento ( $d$ ) e a guinada atual do veículo ( $yaw$ ), pode-se então calcular como saída a aceleração em  $x$  ( $V_{x\_dot}$ ), em  $y$  ( $V_{y\_dot}$ ) e a variação da guinada ( $yaw\_dot$ ).

Estas saídas, tal qual no bloco de dinâmica longitudinal, ao serem integradas tem como resultado a velocidade em  $x$ ,  $y$  e a guinada para o passo posterior.

### 3.1.1.6 Implementação em Simulink

Ao verificar as imagens da subseção anterior, observa-se que foram utilizados blocos de função para a implementação dos modelos. Isto pode ser facilmente justificado, uma vez que dada a complexidade das equações a serem convertidas em lógica de blocos, optou-se pela utilização da linguagem de *script* do *Matlab*, uma vez que esta é mais fácil de ajustar e modificar em casos em que os procedimentos de elaboração dos modelos demandassem tal.

### 3.1.2 Fluxograma para geração de arquivo



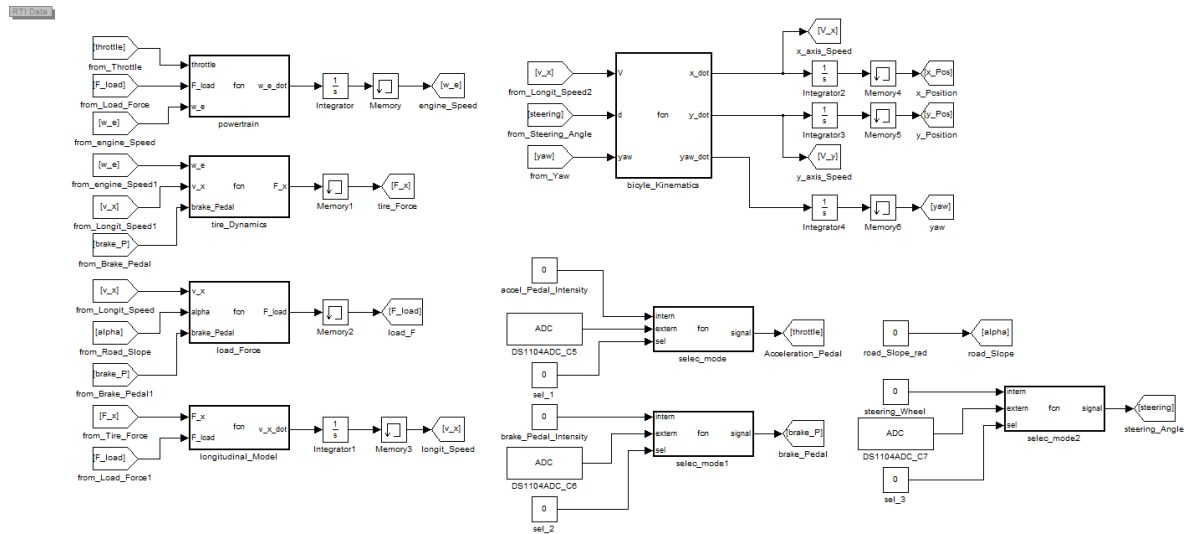
**Fonte: autoria própria (2022).**

O fluxograma presente na Fig. 19 indica as etapas necessária para gerar o código a ser embarcado no *hardware*.

#### 3.1.2.1 Modelo da planta física para embarcar em hardware dSpace

Inicialmente é utilizado o software *Simulink* para implementar o modelo matemático estabelecido no estudo inicial da planta a ser modelada. A imagem abaixo mostra o modelo implementado em *Simulink*. É importante salientar que o modelo foi implementado utilizando como bloco base, os blocos de função *Matlab* disponíveis no ambiente *Simulink*, ou seja, todo o modelo foi escrito utilizando a sintaxe de *script Matlab*, porém segmentado em funções individuais contendo componentes da modelagem veicular.

Figura 20 – Modelo implementado em ambiente Simulink



Fonte: autoria própria (2022).

A modelagem implementada na Fig. 20 seguiu os fundamentos da Lei de Newton como citado no Capítulo 2, no entanto foi necessário implementar todo o equacionamento usando funções matemáticas, devido a limitação da versão do *Simulink* utilizada. A utilização desta versão se fez necessária pois a licença da *dSpace* existente somente suportava versões de *Matlab* 2010.

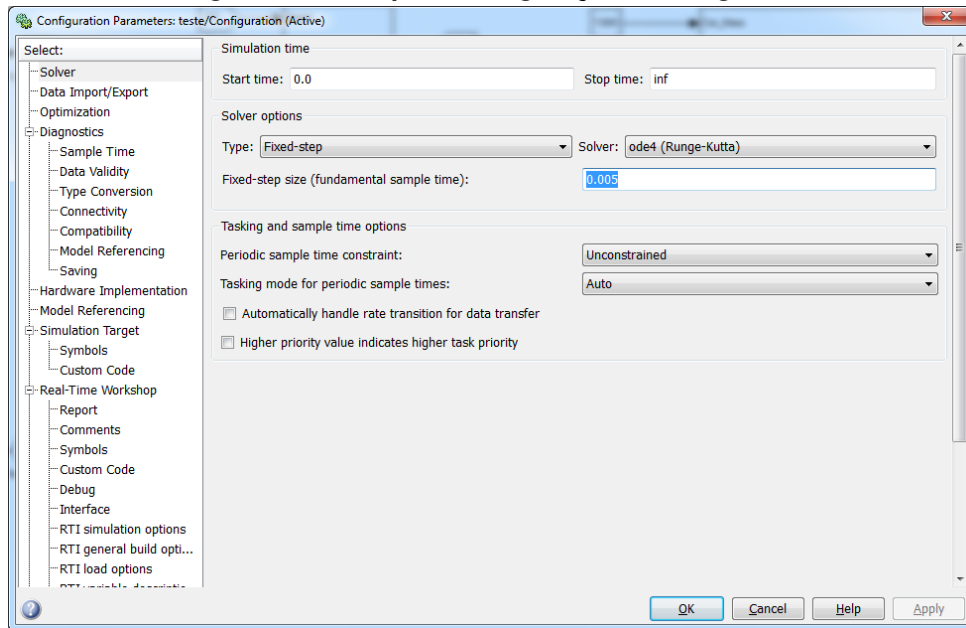
É importante pontuar que foi escolhido utilizar um modelo de dinâmica longitudinal, o qual considera a existência de diferença entre a velocidade do carro e do pneu em alguns casos, chamado *slip*, assim, foi necessário um modelo do pneu para devidamente avaliar o algoritmo. Para isto, foi escolhido o modelo linear, uma vez que para a grande maioria dos casos este é suficiente, conforme fala Pacejka. Poderia ser utilizado a *magic formula* desenvolvida por Pacejka, mas este traria uma complexidade maior à implementação do modelo, a qual não traria uma vantagem para as situações onde este modelo irá trabalhar, como pode ser visto nos resultados.

No entanto como o principal objetivo do trabalho é desenvolver um *framework* que auxiliará na confecção de algoritmos de controle. A modelagem necessária para reproduzir o conteúdo abordado, está descrita no Apêndice A, já os códigos do modelo podem ser obtidos no Apêndice B.

### 3.1.2.2 Configuração do integrador

A imagem abaixo mostra a segunda etapa para a geração do arquivo .sdf necessário para que o modelo possa ser embarcado ao *hardware*.

**Figura 21 – Tela para configuração do integrador**



**Fonte: autoria própria (2022).**

Na primeira tela de configuração é necessário especificar o método de resolução do sistema, esse parâmetro irá determinar o quão rápido e preciso a solução obtida será. É conhecido que um sistema dinâmico é solucionado numericamente através de integradores, que nada mais são que algoritmos com capacidade de resolver equações diferenciais por métodos computáveis. Isto posto, para que o modelo da planta física possa ser simulado, foi necessário a seleção de um integrador, o qual será responsável por fazer com que o sistema modelado convirja para o valor correto. A seleção do integrador é um passo muito importante, pois este influencia em fatores vitais para a confiabilidade da simulação como o erro do modelo discreto em relação ao modelo analítico e, principalmente, na velocidade de processamento. Inicialmente foi considerado utilizar o método de Euler como integrador, uma vez que este é muito simples computacionalmente, porém foi desconsiderado, já que este método de integração depende de um período de amostragem (*step-size*) muito pequeno para ter um erro baixo, e isto faz com que o *hardware* precise realizar muito processamento para chegar à convergência, algo que compromete a simulação em tempo real. Nesse caso optou-se

por um método de quarta ordem (Runge-Kutta) o qual foi o método de maior ordem suportados para os equipamentos empregados no trabalho.

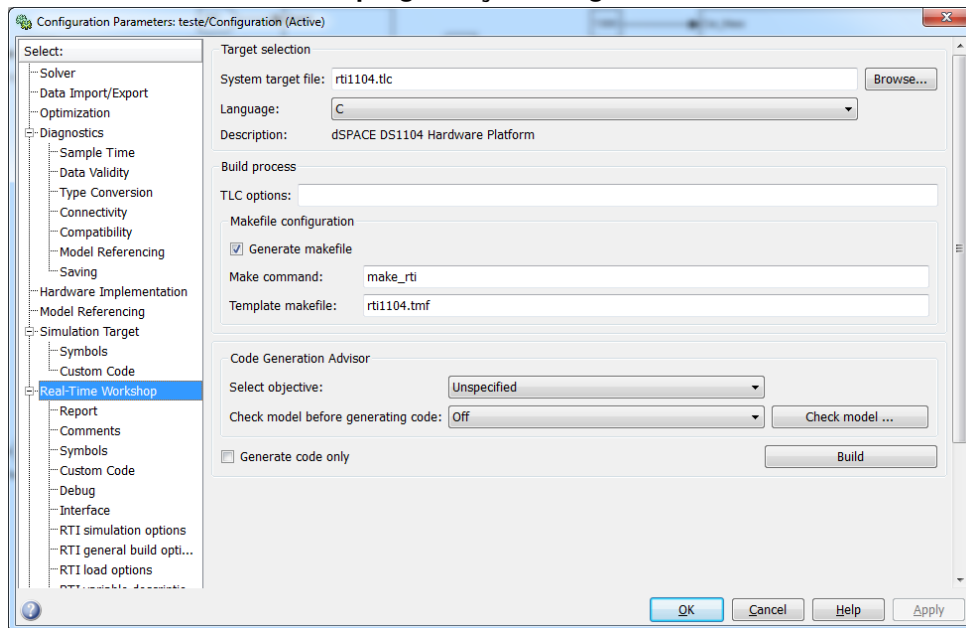
$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3)$$

Conforme (BOYCE; DIPRIMA, 1998),  $h$  é o passo da solução e os demais coeficientes são utilizados para realizar a devida aproximação desejada. A definição de tais componentes é descrita postulado do método numérico. Com a seleção do integrador, outro importante parâmetro a se especificar é o tempo de amostragem da simulação, assim como o método de resolução, o tempo de amostragem será delimitado devido às capacidades dos hardwares utilizados, entretanto como o integrador Runge-Kutta tem capacidade de pequenos erros, utilizando períodos de amostragens relativamente grandes, foi possível utilizar um período de 5 milissegundos. Por fim, nesta tela ainda precisa indicar que a simulação será rodada continuamente. Para este fim, basta colocar no parâmetro *stop time* como "inf", indicando que a simulação será infinita.

### 3.1.2.3 Seleção do target

As duas telas a seguir Fig. 22 e Fig. 23, mostram dois passos importantes na geração do arquivo .sdf, sendo estas a seleção da linguagem de programação que o código gerado será (C ou C++) e qual o *target* destino o código será embarcado.

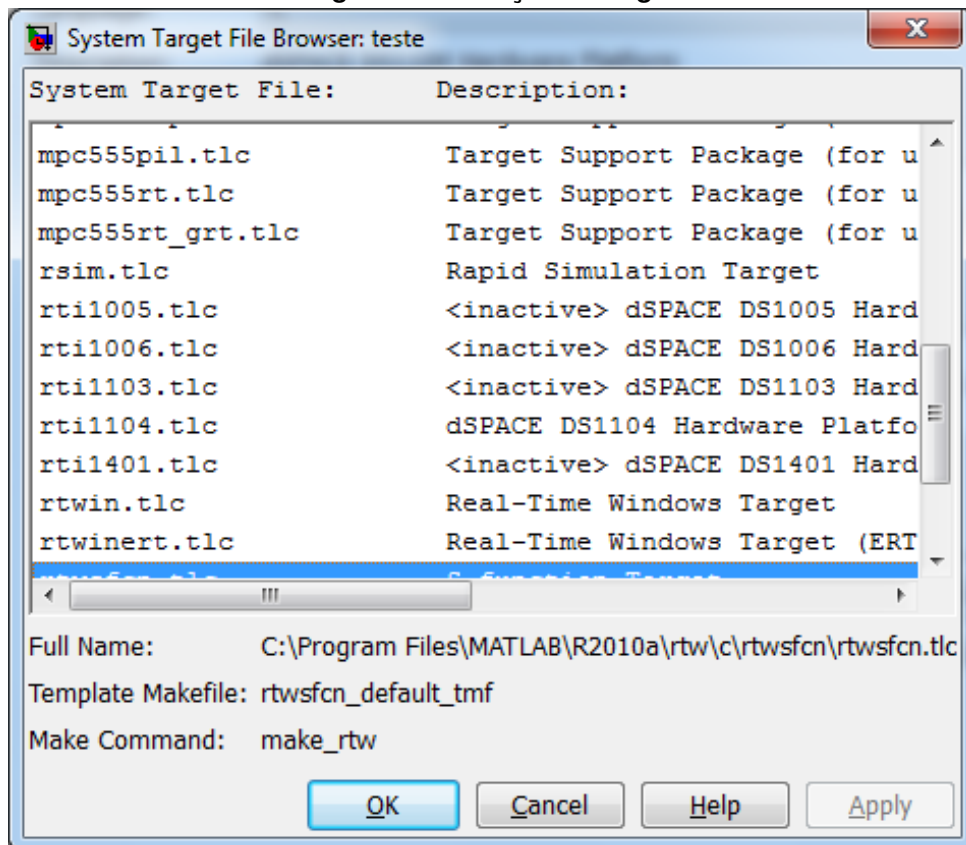
**Figura 22 – Tela para configuração da linguagem de programação e target**



**Fonte: autoria própria (2022).**

Esta etapa é grandemente dependente do *hardware* utilizado, uma vez que a seleção errada destes itens não necessariamente irá gerar erro, porém ao ser embarcado no dispositivo este não irá se comportar como desejado, ou mesmo não irá nem aceitar que o código seja embarcado. Inicialmente, na Figura 22, existe a possibilidade da seleção ou da linguagem C ou da linguagem C++. Para a implementação do modelo em questão deve-se manter selecionado a linguagem C, a qual já é a seleção padrão. O segundo ponto nesta tela é realizar a seleção do *target* correto, e para isto basta selecionar em *System target file*, para que a tela de seleção seja exibida.

Figura 23 – Seleção de target



Fonte: autoria própria (2022).

Nesta tela, Figura 23, é necessário selecionar o arquivo .tlc correspondente ao *hardware* utilizado. Este arquivo não é necessariamente fornecido pela *Mathworks*, mas sim pela desenvolvedora do sistema utilizado, no caso do sistema em questão, a *dSpace*, a qual fornece o arquivo adquirindo a licença necessária.

Para a geração do código para o modelo do presente trabalho deve-se selecionar o arquivo **rti1104.tlc**. Após a seleção do arquivo basta confirmar a ação.

### 3.2 FERRAMENTAS MATHWORKS

A ferramenta utilizada para modelar o veículo foi o *Simulink*, que consiste em um software o qual utiliza de diagrama de blocos e fluxo de sinais para devidamente representar um sistema físico. Um recurso do programa que foi essencial para confecção do modelo foi a extensão RTI provida pela *dSpace*, a qual provém blocos utilizados para simulação em tempo real.

Com a utilização das ferramentas da *MathWorks*, pode-se construir um modelo

com restrições parecidas ao sistema real, dessa forma assegurando que o comportamento visto em ambiente controlado será bem próximo do comportamento em um ambiente real. Além de realizar a geração de um código na linguagem C, o qual acelera o processo de desenvolvimento de software e de um descritor de arquivos que viabilizará a integração das ferramentas.

Para viabilizar a posterior utilização do *ControlDesk*, é necessário realizar uma série de configurações no ambiente *Simulink*, assim viabilizando a integração das duas ferramentas de desenvolvimento.

A terceira tela sinaliza o sucesso do procedimento, mostrando os arquivos gerados e que posteriormente deveram ser utilizados no software da *dSpace*.

### 3.3 FERRAMENTAS DSPACE

No presente trabalho foram utilizados dois diferentes equipamentos para embarcar e testar os algoritmos de controle, ambos fabricados pela empresa *dSpace*. Sendo eles o DS1104 e MicroAutobox, os quais as especificações serão melhor descrita ao longo do texto. O software, que permitirá programar o código a ser embarcado nos hardwares citados, denominasse *ControlDesk*.

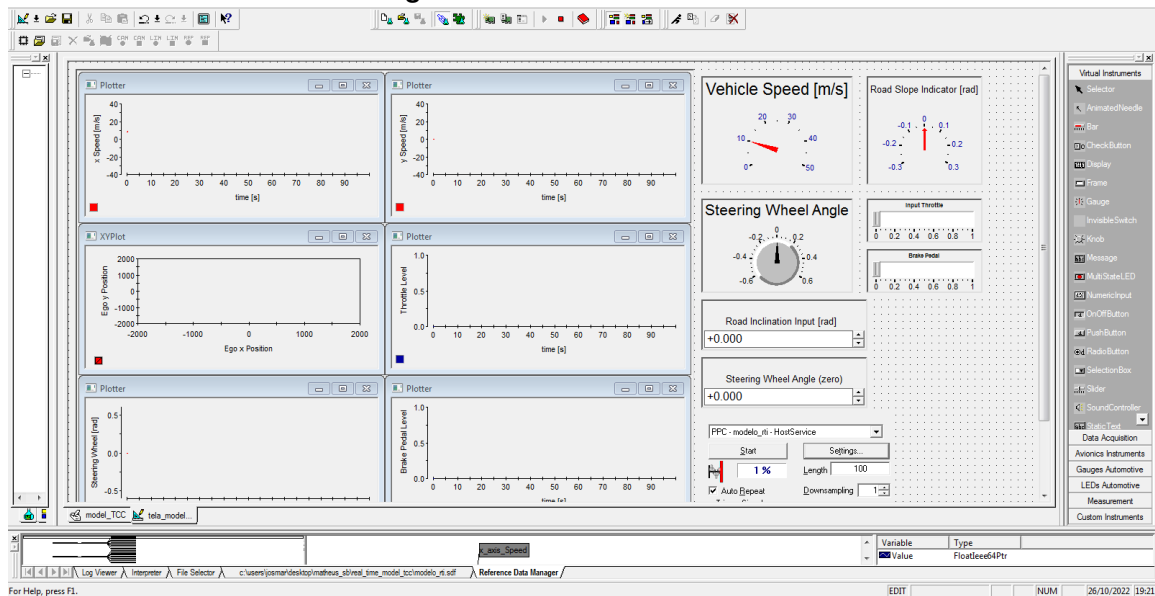
#### 3.3.1 Software dSpace ControlDesk

O *ControlDesk* funciona com uma lógica similar a um sistema supervisorio, oferecendo uma interface gráfica ao usuário. A ferramenta da *dSpace*, utiliza o diagrama e código gerado na ferramenta da *MathWorks* para embarcar o mesmo no hardware desejado. Essa integração de ferramentas de ambas empresas acontece através de arquivo *.sdf* que atua como um descritor de arquivos, assim informando o *ControlDesk* a maneira correta de executar o software gerado pelo *Simulink*.

O supervisorio proposto para o trabalho consta de informações como a velocidade do veículo, o ângulo formado pelo volante, a inclinação da rodovia, entre diversos outros parâmetros que auxiliam na criação de cenários de teste.



Figura 24 – Tela do ControlDesk



Fonte: autoria própria (2022).

### 3.3.2 Hardware dSpace DS1104

Conforme o manual de instalação e configuração (DSPACE, 2011b) informa o *hardware* DS1104 é uma placa controladora, a qual consiste de uma placa padrão, que pode ser conectada a uma placa PCI de um computador. Esta placa controladora é um sistema completo para controle em tempo real, tendo em sua construção eletrônica um processador com unidade de ponto flutuante baseado em arquitetura PowerPC 603, com frequência de *clock* de 250MHz. A placa controladora ainda possui um processador digital de sinais (DSP) *slave*, para aplicações avançadas de entrada/saída (IO).

**Figura 25 – Hardware DS1104**

**Fonte: dSpace (2011a).**

Como pode ser visto no manual supracitado, a placa controladora contém um conector de *I/O* de 100 pinos, um conector para *debug* do DSP *slave* do tipo *JTAG 1149*, um *jumper* para controle do modo de proteção da memória *flash* e do modo de *watchdog*, um *LED* vermelho, usado para fins de *troubleshooting*, e o conector de *debug* do *master PPC*, do tipo *JTAG 1149*. (DSPACE, 2011b, p. 37).

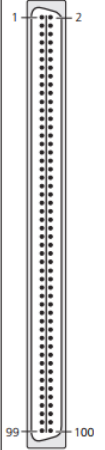
### 3.3.2.1 Mapeamento e acesso aos I/O's

O conector de *I/O* possui 100 pinos, do tipo *KEL* de alta densidade, isto é, disponibilizando os sinais em um conector que ocupa um pequeno espaço em relação à quantidade de pinos. É válido ressaltar que o manual especifica que no caso do uso da fonte de +5V proveniente da placa (Pinos P1 19/20, P1 A 4, e p1B 4), a carga máxima não deve ultrapassar 500mA.

Com relação à proteção de sobrecorrente e curto-circuito, o multi fusível presente na placa controladora irá sobreaquecer, elevando sua resistência ôhmica. Para que a placa volte a operar normalmente, basta realizar seu desligamento por alguns minutos, até que o multi fusível esfrie (DSPACE, 2011b).

Ao longo desse seguimento do trabalho pode ser visto o mapeamento das entradas e saídas presentes no conector P1 da placa controladora DS1104.

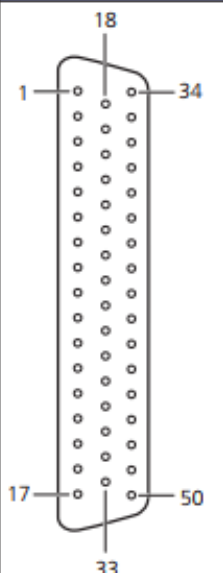
**Figura 26 – Mapeamento de entradas e saídas da placa DS1104**

I/O Connector (P1)	Pin	Sub-D Pin	Signal	Pin	Sub-D Pin	Signal	I/O Connector (P1)	Pin	Sub-D Pin	Signal	Pin	Sub-D Pin	Signal
	P1 1	P1B 1	GND	P1 2	P1A 1	GND	P1 65	P1B 28	IO3	P1 66	P1A 28	IO2	
	P1 3	P1B 34	DCD (CTS)	P1 4	P1A 34	(RTS)	P1 67	P1B 12	IO1	P1 68	P1A 12	IO0	
	P1 5	P1B 18	CTS (CTS)	P1 6	P1A 18	RTS (RTS)	P1 69	P1B 45	GND	P1 70	P1A 45	GND	
	P1 7	P1B 2	DSR (RXD)	P1 8	P1A 2	DTR (TXD)	P1 71	P1B 29	DACH8	P1 72	P1A 29	DACH7	
	P1 9	P1B 35	RXD (RXD)	P1 10	P1A 35	TXD (TXD)	P1 73	P1B 13	GND	P1 74	P1A 13	GND	
	P1 11	P1B 19	SSOMI	P1 12	P1A 19	SCAP4	P1 75	P1B 46	DACH6	P1 76	P1A 46	DACH5	
	P1 13	P1B 3	SSIMO	P1 14	P1A 3	SCAP3	P1 77	P1B 30	GND	P1 78	P1A 30	GND	
	P1 15	P1B 36	SSTE	P1 16	P1A 36	SCAP2	P1 79	P1B 14	DACH4	P1 80	P1A 14	DACH3	
	P1 17	P1B 20	SSCLK	P1 18	P1A 20	SCAP1	P1 81	P1B 47	GND	P1 82	P1A 47	GND	
	P1 19	P1B 4	VCC (+5 V)	P1 20	P1A 4	VCC (+5 V)	P1 83	P1B 31	DACH2	P1 84	P1A 31	DACH1	
	P1 21	P1B 37	ST3PWM	P1 22	P1A 37	SPWM6	P1 85	P1B 15	GND	P1 86	P1A 15	GND	
	P1 23	P1B 21	ST2PWM	P1 24	P1A 21	SPWM5	P1 87	P1B 48	ADCH8	P1 88	P1A 48	ADCH7	
	P1 25	P1B 5	ST1PWM	P1 26	P1A 5	SPWM4	P1 89	P1B 32	GND	P1 90	P1A 32	GND	
	P1 27	P1B 38	SPWM9	P1 28	P1A 38	SPWM3	P1 91	P1B 16	ADCH6	P1 92	P1A 16	ADCH5	
	P1 29	P1B 22	SPWM8	P1 30	P1A 22	SPWM2	P1 93	P1B 49	GND	P1 94	P1A 49	GND	
	P1 31	P1B 6	SPWM7	P1 32	P1A 6	SPWM1	P1 95	P1B 33	ADCH4	P1 96	P1A 33	ADCH3	
	P1 33	P1B 39	GND	P1 34	P1A 39	GND	P1 97	P1B 17	GND	P1 98	P1A 17	GND	
	P1 35	P1B 23	IDX(2)	P1 36	P1A 23	IDX(1)	P1 99	P1B 50	ADCH2	P1 100	P1A 50	ADCH1	
	P1 37	P1B 7	IDX(2)	P1 38	P1A 7	IDX(1)							
	P1 39	P1B 40	PHI90(2)	P1 40	P1A 40	PHI90(1)							
	P1 41	P1B 24	PHI90(2)	P1 42	P1A 24	PHI90(1)							
	P1 43	P1B 8	PHI0(2)	P1 44	P1A 8	PHI0(1)							
	P1 45	P1B 41	PHI0(2)	P1 46	P1A 41	PHI0(1)							
	P1 47	P1B 25	GND	P1 48	P1A 25	GND							
	P1 49	P1B 9	IO19	P1 50	P1A 9	IO18							
	P1 51	P1B 42	IO17	P1 52	P1A 42	IO16							
	P1 53	P1B 26	IO15	P1 54	P1A 26	IO14							
	P1 55	P1B 10	IO13	P1 56	P1A 10	IO12							
	P1 57	P1B 43	IO11	P1 58	P1A 43	IO10							
	P1 59	P1B 27	IO9	P1 60	P1A 27	IO8							
	P1 61	P1B 11	IO7	P1 62	P1A 11	IO6							
	P1 63	P1B 44	IO5	P1 64	P1A 44	IO4							

Fonte: dSpace (2011a).

Abaixo pode ser visto o mapeamento do conector P1A, presente no hardware do CLP1104, o qual é utilizado para que o usuário do sistema possa usar alguns dos I/O's presentes na placa DS1104.

**Figura 27 – Mapeamento do conector P1A**

Connector P1A	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND	18	RTS (RTS)	34	(RTS)
	2	DTR (TXD)	19	SCAP4	35	TXD (TXD)
	3	SCAP3	20	SCAP1	36	SCAP2
	4	VCC (+5 V)	21	SPWM5	37	SPWM6
	5	SPWM4	22	SPWM2	38	SPWM3
	6	SPWM1	23	IDX(1)	39	GND
	7	IDX(1)	24	PHI90(1)	40	PHI90(1)
	8	PHI0(1)	25	GND	41	PHI0(1)
	9	IO18	26	IO14	42	IO16
	10	IO12	27	IO8	43	IO10
	11	IO6	28	IO2	44	IO4
	12	IO0	29	DACH7	45	GND
	13	GND	30	GND	46	DACH5
	14	DACH3	31	DACH1	47	GND
	15	GND	32	GND	48	ADCH7
	16	ADCH5	33	ADCH3	49	GND
	17	GND			50	ADCH1

Fonte: dSpace (2011a).

Tal qual o conector P1A, o conector P1B também faz parte do hardware CLP1104 para dar acesso a outros I/O's da placa DS1104.

Figura 28 – Mapeamento do conector P1B

Connector P1B	Pin	Signal	Pin	Signal	Pin	Signal
	1	GND	18	CTS ( $\overline{CTS}$ )	34	DCD (CTS)
	2	DSR (RXD)	19	SSOMI	35	RXD ( $\overline{RXD}$ )
	3	SSIMO	20	SSCLK	36	SSTE
	4	VCC (+5 V)	21	ST2PWM	37	ST3PWM
	5	ST1PWM	22	SPWM8	38	SPWM9
	6	SPWM7	23	$\overline{IDX(2)}$	39	GND
	7	$\overline{IDX(2)}$	24	PHI90(2)	40	$\overline{PHI90(2)}$
	8	$\overline{PHI0(2)}$	25	GND	41	PHI0(2)
	9	IO19	26	IO15	42	IO17
	10	IO13	27	IO9	43	IO11
	11	IO7	28	IO3	44	IO5
	12	IO1	29	DACH8	45	GND
	13	GND	30	GND	46	DACH6
	14	DACH4	31	DACH2	47	GND
	15	GND	32	GND	48	ADCH8
	16	ADCH6	33	ADCH4	49	GND
	17	GND			50	ADCH2

Fonte: dSpace (2011a).

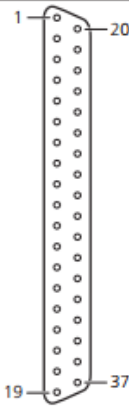
Abaixo nas Fig. 29 e Fig. 30, podem ser observados a relação de pinos de acessos nos conectores CP17 e CP18.

Figura 29 – Mapeamento ddo conector CP17

Connector (CP17)	Pin	Signal	Pin	Signal
	19	GND	37	VCC (+5 V)
	18	GND	36	VCC (+5 V)
	17	GND	35	GND
	16	GND	34	GND
	15	IO19	33	IO18
	14	IO17	32	IO16
	13	GND	31	GND
	12	IO15	30	IO14
	11	IO13	29	IO12
	10	GND	28	GND
	9	IO11	27	IO10
	8	IO9	26	IO8
	7	GND	25	GND
	6	IO7	24	IO6
	5	IO5	23	IO4
	4	GND	22	GND
	3	IO3	21	IO2
	2	IO1	20	IO0
	1	GND		

Fonte: dSpace (2011a).

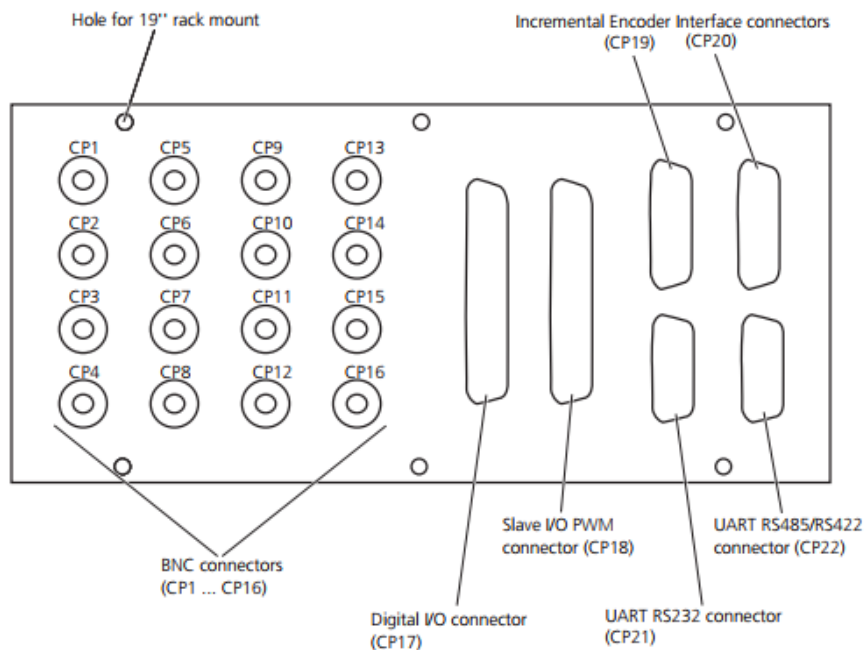
**Figura 30 – Mapeamento ddo conector CP18**

Connector (CP18)	Pin	Signal	Pin	Signal
	1	GND	20	GND
	2	SCAP1	21	SCAP2
	3	SCAP3	22	SCAP4
	4	GND	23	ST1PWM
	5	ST2PWM	24	ST3PWM
	6	GND	25	GND
	7	SPWM1	26	SPWM2
	8	SPWM3	27	SPWM4
	9	SPWM5	28	SPWM6
	10	SPWM7	29	SPWM8
	11	SPWM9	30	GND
	12	GND	31	GND
	13	GND	32	GND
	14	GND	33	GND
	15	GND	34	SSOMI
	16	SSIMO	35	SSTE
	17	SCLK	36	GND
	18	VCC (+5 V)	37	GND
	19	VCC (+5 V)		

Fonte: dSpace (2011a).

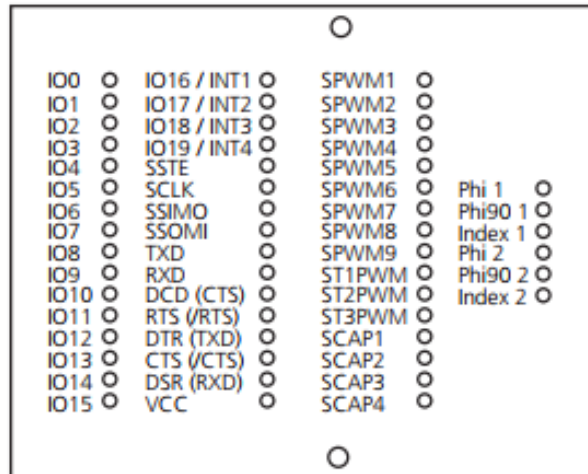
Abaixo na Figura 31 pode ser obtido o esquemático geral da disposição dos conectores do hardware CLP 1104, já na Figura 32 tem se os LED's de status das saídas do DS1104. Este LEDs são visualmente acessíveis pelo CLP1104, no entanto referenciam o estado atual de algumas saídas da placa DS1104, bem como alguns status de funcionamento, como a correta alimentação dos circuitos (VCC).

**Figura 31 – Acesso aos I/O's através do hardware CLP1104**



Fonte: dSpace (2011a).

**Figura 32 – LED's de stauts presentes no hardware CLP1104**



Fonte: dSpace (2011a).

Por fim, a figura abaixo relaciona os sinais, que serão acessados via a biblioteca de interface para tempo-real (RTI), com cada um de seus conectores providos no hardware CLP1104.

**Figura 33 – Relação de conectores e sinais**

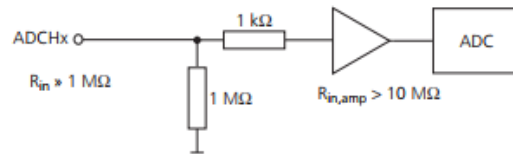
Connector	Signal	Connector	Signal	Connector	Signal	Connector	Signal
CP1	ADCH1	CP5	ADCH5	CP9	DACH1	CP13	DACH5
CP2	ADCH2	CP6	ADCH6	CP10	DACH2	CP14	DACH6
CP3	ADCH3	CP7	ADCH7	CP11	DACH3	CP15	DACH7
CP4	ADCH4	CP8	ADCH8	CP12	DACH4	CP16	DACH8

Fonte: dSpace (2011a).

### 3.3.2.2 Características das entradas e saídas da placa DS1104

As imagens nesta subseção mostram as características eletrônicas das entradas e saídas da DS1104. Estas características devem ser obedecidas ao elaborar um *setup* de teste sob pena do não funcionamento do sistema, bem como para o caso dos limites de tensão e corrente, podem danificar de forma irreversível o funcionamento dos equipamentos.

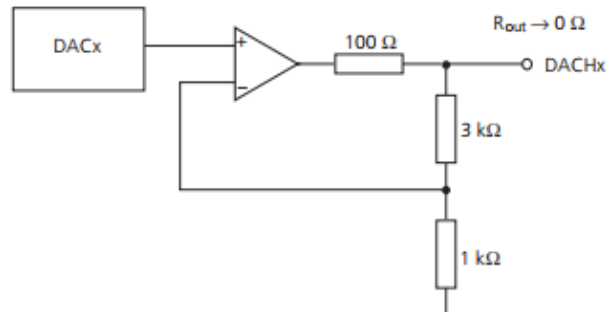
**Figura 34 – Características das entradas analógicas da placa DS1104**



Parameter	Value	
Input voltage	Min. -10 V	Max. +10 V
Input resistance	Approx. 1 MΩ	
SNR (signal-to-noise ratio)	<ul style="list-style-type: none"> <li>■ 16-bit muxed ADCs</li> <li>■ 12-bit parallel ADCs</li> </ul>	<ul style="list-style-type: none"> <li>■ &gt; 80 dB</li> <li>■ &gt; 65 dB</li> </ul>

Fonte: dSpace (2011a).

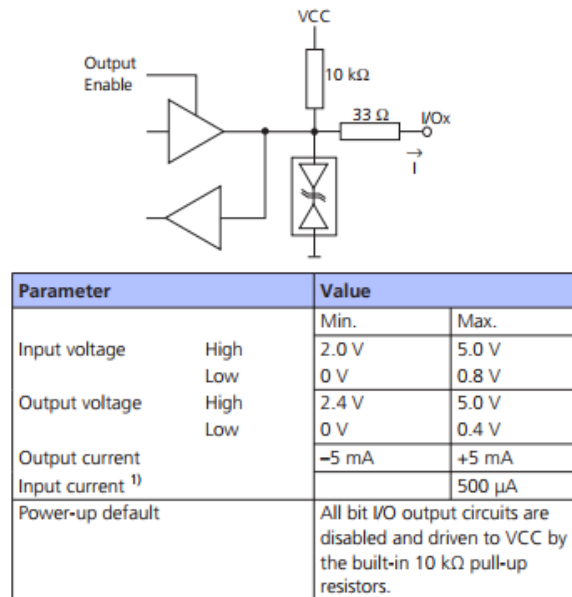
**Figura 35 – Características das saídas analógicas da placa DS1104**



Parameter	Value	
Output voltage	Min. -10 V	Max. +10 V
Output current	-5 mA	+5 mA
Output resistance	→ 0 Ω	
Power-up default	0 V	
SNR (signal-to-noise ratio)	> 80 dB	

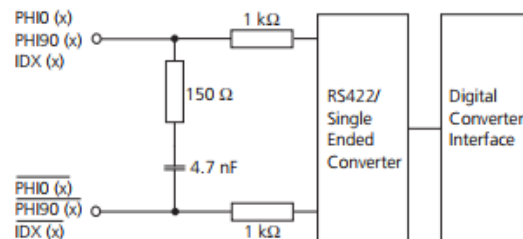
Fonte: dSpace (2011a).

**Figura 36 – Características das entradas/saídas digitais da placa DS1104**



Fonte: dSpace (2011a).

**Figura 37 – Características das entradas para leitura de encoder incremental da placa DS1104**



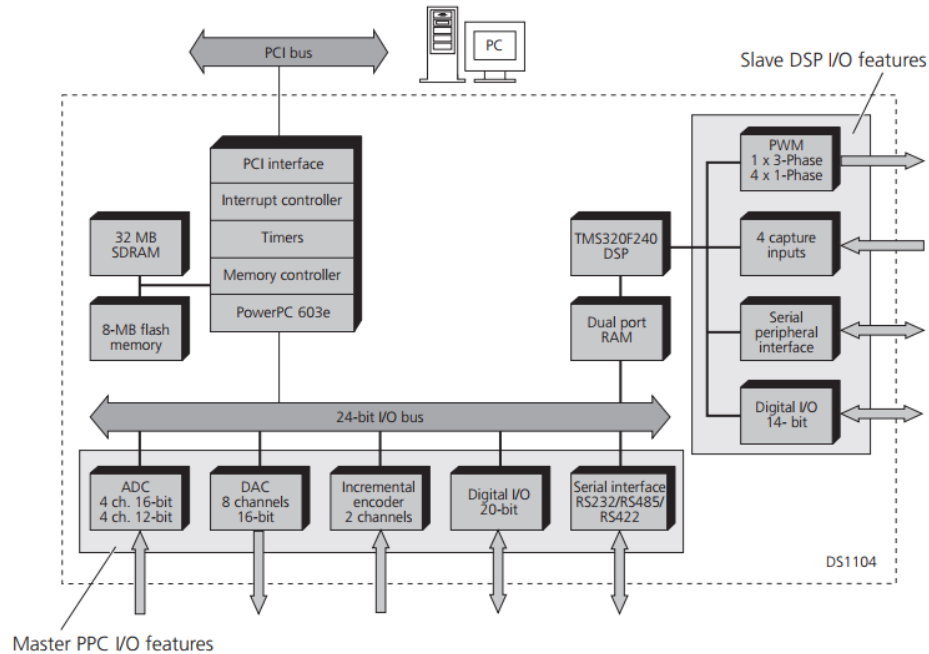
Parameter		Value	
		Min.	Max.
TTL input voltage	High	2.0 V	5.0 V
	Low	0 V	0.8 V
TTL input resistance		12 kΩ	
RS422 input voltage	High	Diff > +0.2 V	
	Low	Diff < -0.2 V	
		<ul style="list-style-type: none"> <li>■ Diff = Voltage difference between non-inverted and inverted signal</li> <li>■ The input signal, together with the corresponding inverted signal, must be in the range 0 ... 5 V.</li> </ul>	
RS422 input resistance		The input resistance gradually drops from 8.5 kΩ at the corner frequency of 28 kHz to 210 Ω at the corner frequency of 225 kHz.	

Fonte: dSpace (2011a).



### 3.3.2.3 Periféricos e seus respectivos blocos para acesso

**Figura 38 – Diagrama de blocos da arquitetura do hardware da placa DS1104**



**Fonte: dSpace (2011a).**

As figuras abaixo, pertencentes a esta subseção tem como objetivo relacionar os conectores de acesso do hardware CLP1104, com seus sinais na placa DS1104 e sua forma de acesso a partir da biblioteca RTI via *Simulink*.

A exemplo, pode-se imaginar que para um dado projeto deseja-se usar uma entrada analógica não multiplexada, assim, seguindo as relações mostradas nas figuras, sabe-se que o conector CP5 irá fornecer a entrada física para o sistema, através do ADCH5. Já para acessá-lo via RTI, deve-se utilizar o bloco DS1104ADC\_Cx, onde o x no bloco será substituído pelo valor 5, que refere-se ao hardware selecionado. A relação dos blocos usados no *Simulink* pode ser vista na próxima subseção.

**Figura 39 – Mapeamento de entradas e saídas da placa DS1104**

Related RTI Blocks	Channel (RTI)	Related RTLib Functions	Channel (RTLib)	Conn. Pin	Sub-D Pin	Pin on CP/CLP	Signal
DS1104MUX_ADC	Ch 1	See ADC Unit (DS1104 RTLib Reference)	Ch 1	P1 100	P1A 50	CP1	ADCH1
	Ch 2		Ch 2	P1 99	P1B 50	CP2	ADCH2
	Ch 3		Ch 3	P1 96	P1A 33	CP3	ADCH3
	Ch 4		Ch 4	P1 95	P1B 33	CP4	ADCH4
DS1104ADC_Cx	Ch 5		Ch 5	P1 92	P1A 16	CP5	ADCH5
	Ch 6		Ch 6	P1 91	P1B 16	CP6	ADCH6
	Ch 7		Ch 7	P1 88	P1A 48	CP7	ADCH7
	Ch 8		Ch 8	P1 87	P1B 48	CP8	ADCH8

Fonte: dSpace (2011a).

**Figura 40 – Mapeamento de entradas e saídas da placa DS1104**

Related RTI Block	Related RTLib Functions	Channel	Conn. Pin	Sub-D Pin	Pin on CP/CLP	Signal
DS1104DAC_Cx	See DAC Unit (DS1104 RTLib Reference)	Ch 1	P1 84	P1A 31	CP9	DACH1
		Ch 2	P1 83	P1B 31	CP10	DACH2
		Ch 3	P1 80	P1A 14	CP11	DACH3
		Ch 4	P1 79	P1B 14	CP12	DACH4
		Ch 5	P1 76	P1A 46	CP13	DACH5
		Ch 6	P1 75	P1B 46	CP14	DACH6
		Ch 7	P1 72	P1A 29	CP15	DACH7
		Ch 8	P1 71	P1B 29	CP16	DACH8



Fonte: dSpace (2011a).

**Figura 41 – Mapeamento de entradas e saídas da placa DS1104**

Related RTI Blocks	Related RTLib Functions	Bit	Conn. Pin	Sub-D Pin	Pin on CP/CLP	Signal
DS1104BIT_IN_Cx/ DS1104BIT_OUT_Cx	See Bit I/O Unit (DS1104 RTLib Reference)	Bit 0	P1 68	P1A 12	CP17 20	IO0
		Bit 1	P1 67	P1B 12	CP17 2	IO1
		Bit 2	P1 66	P1A 28	CP17 21	IO2
		Bit 3	P1 65	P1B 28	CP17 3	IO3
		Bit 4	P1 64	P1A 44	CP17 23	IO4
		Bit 5	P1 63	P1B 44	CP17 5	IO5
		Bit 6	P1 62	P1A 11	CP17 24	IO6
		Bit 7	P1 61	P1B 11	CP17 6	IO7
		Bit 8	P1 60	P1A 27	CP17 26	IO8
		Bit 9	P1 59	P1B 27	CP17 8	IO9
		Bit 10	P1 58	P1A 43	CP17 27	IO10
		Bit 11	P1 57	P1B 43	CP17 9	IO11
		Bit 12	P1 56	P1A 10	CP17 29	IO12
		Bit 13	P1 55	P1B 10	CP17 11	IO13
		Bit 14	P1 54	P1A 26	CP17 30	IO14
		Bit 15	P1 53	P1B 26	CP17 12	IO15
		Bit 16	P1 52	P1A 42	CP17 32	IO16
		Bit 17	P1 51	P1B 42	CP17 14	IO17
		Bit 18	P1 50	P1A 9	CP17 33	IO18
		Bit 19	P1 49	P1B 9	CP17 15	IO19



Fonte: dSpace (2011a).

**Figura 42 – Mapeamento de entradas e saídas da placa DS1104**

Related RTI Blocks	Related RTLib Functions	Channel/Bit	Conn. Pin	Sub-D Pin	Pin on CP/CLP	Signal
DS1104ENC_POS_Cx/ DS1104ENC_SET_POS_Cx	See <i>Incremental Encoder Interface</i> (  <i>DS1104 RTLib Reference</i> )	Ch 1	P1 46	P1A 41	CP19 2	PHIO(1)
			P1 44	P1A 8	CP19 3	/PHIO(1)
			P1 42	P1A 24	CP19 4	PHI90(1)
			P1 40	P1A 40	CP19 5	/PHI90(1)
		Ch 2	P1 45	P1B 41	CP20 2	PHIO(2)
			P1 43	P1B 8	CP20 3	/PHIO(2)
			P1 41	P1B 24	CP20 4	PHI90(2)
			P1 39	P1B 40	CP20 5	/PHI90(2)
DS1104ENC_HW_INDEX_Cx/ DS1104ENC_SW_INDEX_Cx	See <i>Incremental Encoder Interface</i> (  <i>DS1104 RTLib Reference</i> )	Ch 1	P1 38	P1A 7	CP19 6	IDX(1)
			P1 36	P1A 23	CP19 7	/IDX(1)
		Ch 2	P1 37	P1B 7	CP20 6	IDX(2)
			P1 35	P1B 23	CP20 7	/IDX(2)

Fonte: dSpace (2011a).

**Figura 43 – Mapeamento de entradas e saídas da placa DS1104**

Related RTI Blocks	Related RTLib Functions	Conn. Pin	Sub-D Pin	Pin on CP/CLP	Signal
<b>RS232 mode</b>					
DS1104SER_SETUP/ DS1104SER_STAT/ DS1104SER_TX/ DS1104SER_RX/ DS1104SER_INT_jy/ DS1104SER_INT_REC_LEV	See <i>Serial Interface Communication</i> (  <i>DS1104 RTLib Reference</i> )	P1 3	P1B 34	CP21 1	DCD
		P1 5	P1B 18	CP21 8	CTS
		P1 6	P1A 18	CP21 7	RTS
		P1 7	P1B 2	CP21 6	DSR
		P1 8	P1A 2	CP21 4	DTR
		P1 9	P1B 35	CP21 2	RXD
		P1 10	P1A 35	CP21 3	TXD
<b>RS422/RS485 mode</b>					
DS1104SER_SETUP/ DS1104SER_STAT/ DS1104SER_TX/ DS1104SER_RX/ DS1104SER_INT_jy/ DS1104SER_INT_REC_LEV	See <i>Serial Interface Communication</i> (  <i>DS1104 RTLib Reference</i> )	P1 3	P1B 34	CP22 8	CTS
		P1 4	P1A 34	CP22 7	RTS
		P1 5	P1B 18	CP22 9	$\overline{\text{CTS}}$
		P1 6	P1A 18	CP22 6	$\overline{\text{RTS}}$
		P1 7	P1B 2	CP22 3	RXD
		P1 8	P1A 2	CP22 2	TXD
		P1 9	P1B 35	CP22 4	$\overline{\text{RXD}}$
P1 10	P1A 35	CP22 1	$\overline{\text{TXD}}$		

Fonte: dSpace (2011a).

#### 3.3.2.4 Biblioteca para Interface de Tempo-Real - RTI Lib

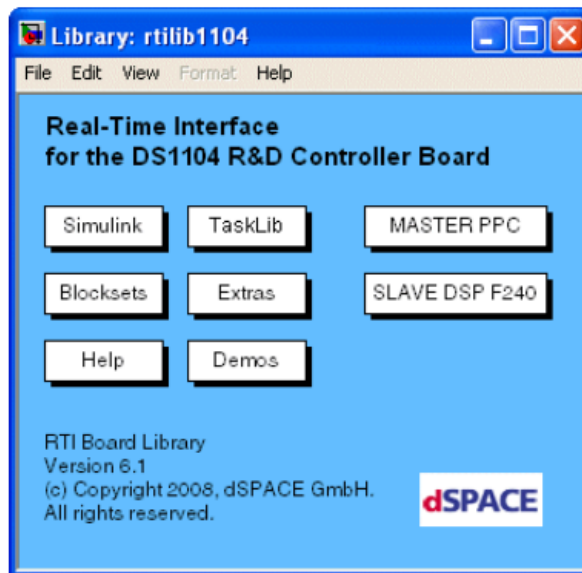
Abaixo podem ser visualizados os principais blocos para acesso dos periféricos da placa DS1104, via *Simulink*.

Para se ter acesso a estes blocos, uma vez que seja feita a correta instalação do suporte *dSpace* para *Matlab/Simulink*, basta ao iniciar o *Matlab*, digitar o comando "**rti1104**"(caso a placa utilizada seja a DS1104, no caso de um hardware de RCP como

a *MicroAutobox*, deve-se utilizar o comando "rti1401").

Ao realizar o comando supracitado, a tela da Figura 44 irá aparecer. Nesta tela inicial podem ser acessados todos os recursos de RTI, desde os mais simples até os mais avançados, como o uso do hardware DSP slave presente no dispositivo.

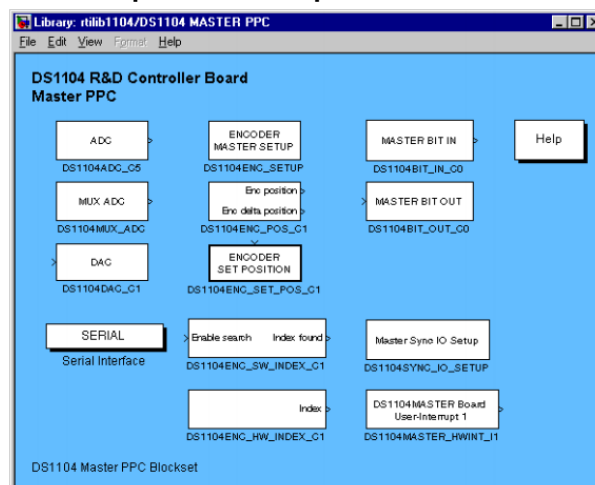
**Figura 44 – Tela inicial da RTI Lib**



Fonte: dSpace (2011a).

Para acessar os periféricos presentes na placa, basta dar um duplo clique no botão com o texto "MASTER PPC". Após este procedimento ser realizado, a tela presente na Figura 45 irá aparecer, contendo os blocos para acesso aos principais periféricos, como os já citados ADC e DAC.

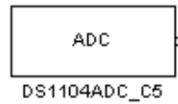
**Figura 45 – Blocos de acesso via RTI Lib para os principais periféricos da placa DS1104**



Fonte: dSpace (2011a).

As figuras abaixo, nesta subseção, mostram os blocos, bem como suas características de tipo de dados e limites de sinais, os quais devem ser respeitados para a correta implementação dos sistemas usando *Simulink* e a *dSpace RTI Lib*.

**Figura 46 – Bloco para acesso ao ADC presente na DS1104**



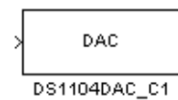
Fonte: dSpace (2011a).

**Figura 47 – Características do ADC presente na DS1104**

Input Voltage Range	Simulink Output
-10 V ... +10 V	-1 ... +1 (double)

Fonte: dSpace (2011a).

**Figura 48 – Bloco para acesso ao DAC presente na DS1104**



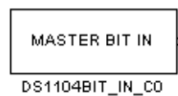
Fonte: dSpace (2011a).

**Figura 49 – Características do DAC presente na DS1104**

Simulink Input	Output Voltage Range
-1 ... +1 (double)	-10 V ... +10 V

Fonte: dSpace (2011a).

**Figura 50 – Bloco para acesso à DI presente na DS1104**



Fonte: dSpace (2011a).

**Figura 51 – Características do DI presente na DS1104**

Digital Input (TTL)	Simulink Output	
	Without Data Typing	With Data Typing
High	1 (double)	1 (boolean)
Low	0 (double)	0 (boolean)

Fonte: dSpace (2011a).

**Figura 52 – Bloco para acesso à DO presente na DS1104**



Fonte: dSpace (2011a).

**Figura 53 – Características do DO presente na DS1104**

Simulink Input		Digital Output (TTL)
<i>Without Data Typing</i>	<i>With Data Typing</i>	
> 0 (double)	1 (boolean)	High
≤ 0 (double)	0 (boolean)	Low

Fonte: dSpace (2011a).

### 3.3.3 Hardware dSpace CP1104/CLP1104

**Figura 54 – Dispositivo dSpace CP1104/CLP1104**



Fonte: dSpace (2011a).

## 3.4 CENÁRIOS ANALISADOS

A modelagem proposta permite a realização de testes que utilizam do modelo m e do modelo longitudinal do veículo. O modelo elaborado é MIMO, uma vez que permite monitorar diversas saídas e entradas do sistema em questão. É importante salientar que o modelo a ser utilizado irá ser definido pela função estudada.

### 3.4.1 Cenários de atuação do Modelo Lateral

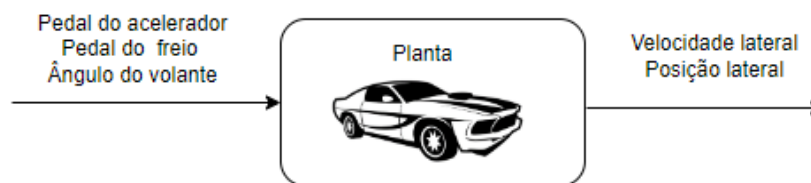
Em situações de mudança que exigem mudança de faixa, ou movimentação no eixo Y, o controle deverá ser feito com base no modelo lateral do veículo. Ao aplicar funções de LKA,LDW os sinais a serem analisados são:

**Tabela 1 – Entradas e saídas do modelo lateral**

Input	Output
Pedal do acelerador	Velocidade lateral
Pedal do freio	
Ângulo do volante	Posição lateral

**Fonte: autoria própria (2022).**

**Figura 55 – Entradas e saídas modelo lateral**



**Fonte: autoria própria (2022).**

Caso o cenário analisado seja uma manobra ultrapassagem sinais como velocidade lateral, velocidade longitudinal deveram ser contemplados. Sendo importante pontuar que uma das condições de execução da manobra é que o veículo em estudo se encontre com uma velocidade longitudinal superior ao objeto a ser ultrapassado. Em situações que existe uma mudança abrupta no ângulo do volante, é necessário modelar o escorregamento lateral do pneu, o modelo proposto no trabalho desconsidera esse fator, logo o sistema desenvolvido irá funcionar devidamente para pequenas taxas de variação no ângulo do volante.

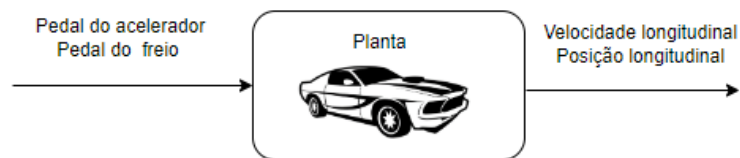
### 3.4.2 Cenários de atuação do Modelo Longitudinal

O modelo longitudinal, o qual contempla a movimentação no eixo X, é requisitado quando funções como ACC, FCW atuam no sistema. O escorregamento do pneu não será contemplado nessa situação, uma vez que a variação no ângulo do volante é desprezada em tal circunstância. Durante esses procedimentos deve-se analisar os sinais de:

**Tabela 2 – Entradas e saídas do modelo longitudinal**

Input	Output
Pedal do acelerador	Velocidade longitudinal
Pedal do freio	Posição longitudinal

Fonte: autoria própria (2022).

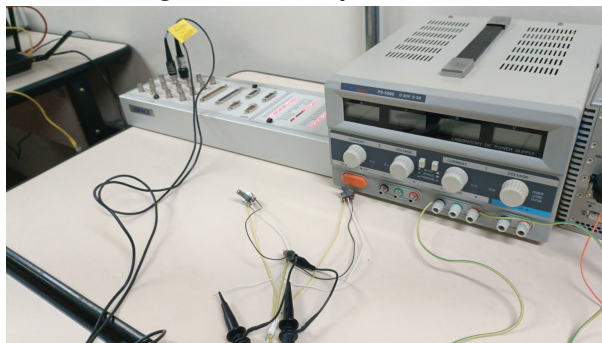
**Figura 56 – Entradas e saídas modelo longitudinal**

Fonte: autoria própria (2022).

Como o modelo longitudinal pode envolver situações de aceleração ou frenagem. Os sinais ora podem ser entrada, ora podem ser saída.

### 3.5 SETUP DE TESTES

O *setup* de testes implementado consistiu na utilização do equipamento DS1104 como simulador da planta física, utilizando deste dispositivo para ler as alterações nos sinais de entrada do modelo.

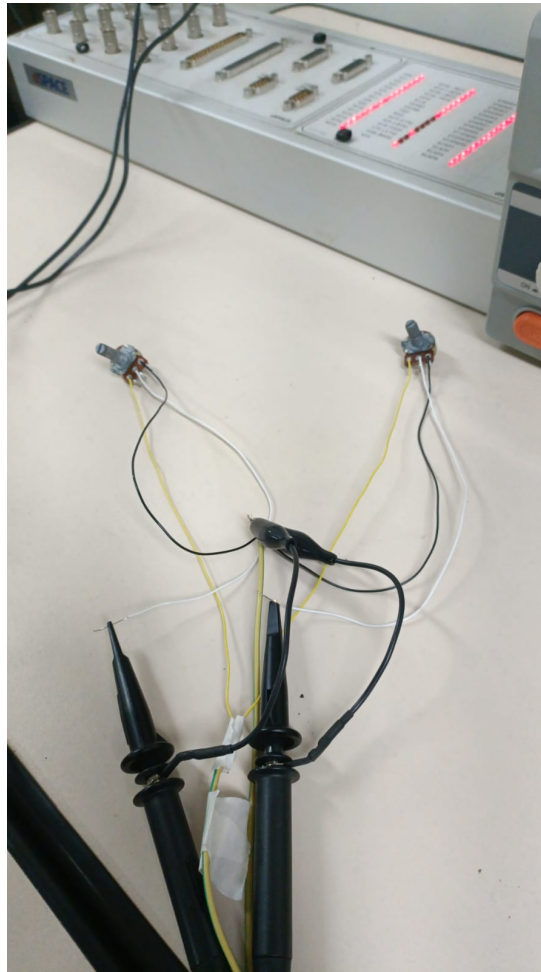
**Figura 57 – Setup de teste 1**

Fonte: autoria própria (2022).

Os sinais de entrada podem ser simulados de maneira manual através da utilização de sistemas baseados em resistências variáveis (potenciômetros) como visto na Figura 58



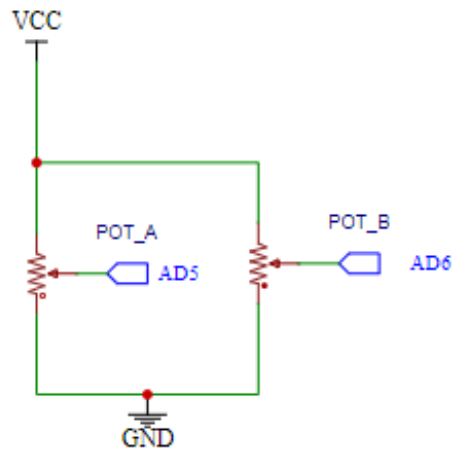
**Figura 58 – Setup de teste 2**



**Fonte: autoria própria (2022).**

Conforme a Figura 59 este trabalho utilizou-se duas entradas analógicas do dispositivo para realizar as devidas ações no modelo.

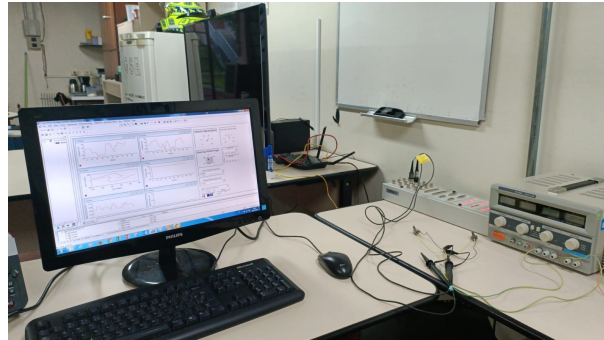
**Figura 59 – Parcela do esquemático da bancada de teste**



**Fonte: autoria própria (2022).**

Dessa forma o *setup* consistiu na utilização da DS1104 juntamente com CLP1104 afim de simular e devidamente representar a planta física. Sendo o gerenciamento do modelo realizado através de um *host*, contendo as ferramentas computacionais utilizadas.

**Figura 60 – Setup de teste 3**



**Fonte: autoria própria (2022).**

Outra forma de simular as entradas no sistema, seria através de um hardware como a *MicroAutoboxII*. É importante salientar que o sistema proposto pode fornecer como saída sinais simulando sensores medindo grandezas físicas da planta física, como velocidade, posição, aceleração entre outros

## 4 RESULTADOS E DISCUSSÃO

Os resultados e discussão foram obtidos após a montagem do *setup* de testes em bancada. Sendo analisados cenários como a resposta ao degrau e o desempenho do veículo ao longo de um trajeto aleatório.

### 4.1 RESPOSTA AO DEGRAU

A análise a resposta ao degrau unitário é uma das formas mais utilizadas para observar o comportamento de um determinado sistema. Para o estudo à resposta ao degrau optou-se em empregar o modelo longitudinal uma vez que os sinais de entrada presentes nele, podem ser devidamente representados pela entrada degrau.

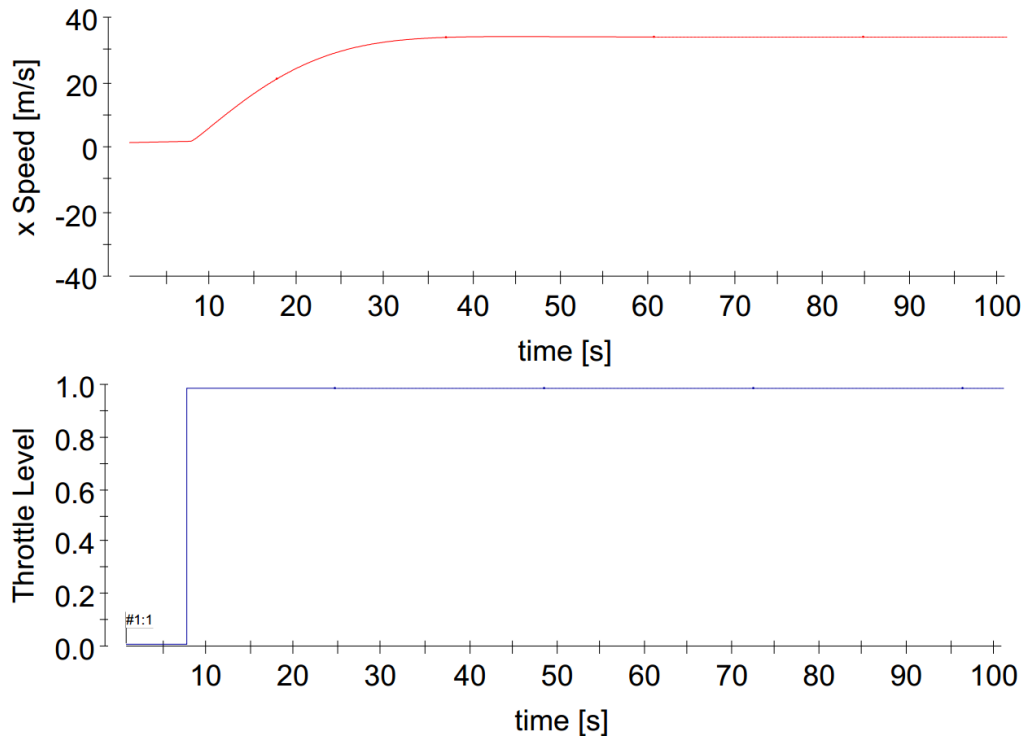
- Pedal do acelerador;
- Pedal do freio.

A modelagem de ambos pedais possuem um *range* que varia entre 0 e 1, onde o valor 1 significa o acionamento total do pedal. No entanto caso seja necessário utilizar um valor diferente de 0 ou 1, é possível empregar uma quantização da entrada.

#### 4.1.1 Resposta no pedal do acelerador

A entrada do modelo em questão foi o sinal lido pelo pedal do acelerador, a saída foi a velocidade longitudinal do veículo. Primeiramente para avaliar a performance do sistema modelado, o modelo foi submetido a uma entrada degrau. Sendo assim possível avaliar características como estabilidade, tempo de resposta, entre outros comportamentos do sistema proposto.

**Figura 61 – Resposta do sistema ao degrau unitário aplicado no pedal de aceleração**



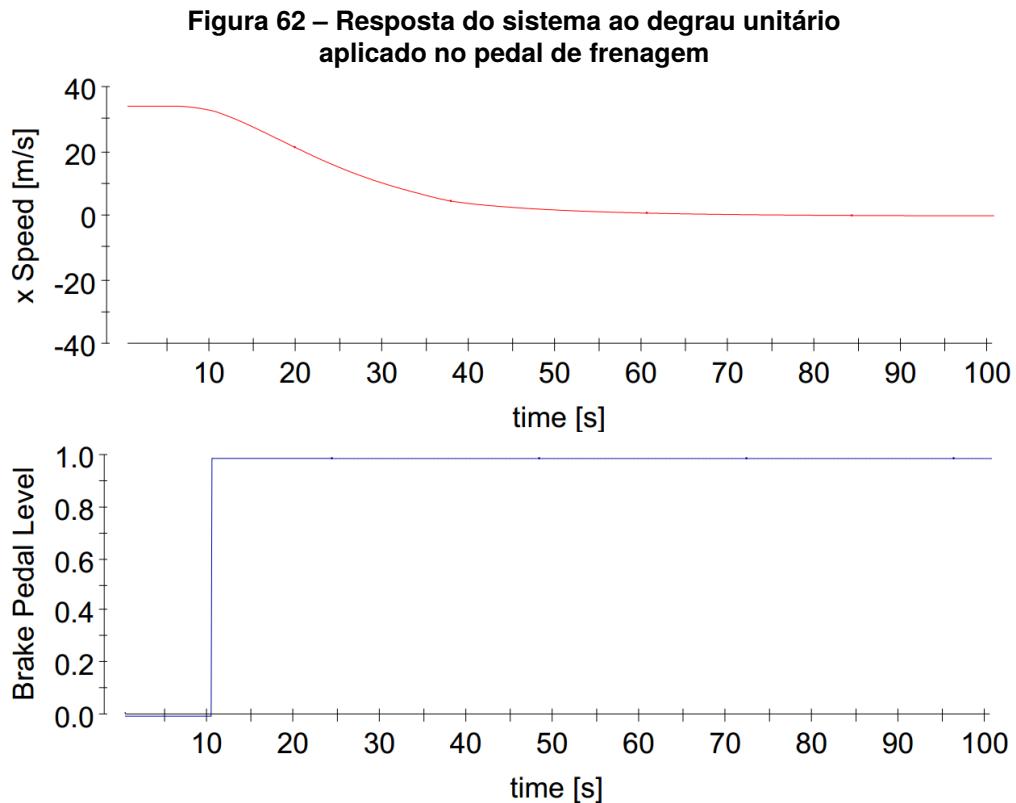
**Fonte: autoria própria (2022).**

Na Figura 61 o gráfico superior como a saída, ou seja, a velocidade longitudinal ou velocidade no eixo X. Já no gráfico inferior está sendo projetada a entrada no pedal do acelerador. É possível ver que o sistema modelado, para a entrada em questão, começa a apresentar uma variação de forma praticamente instantânea. Entrando em regime com uma velocidade de aproximadamente 30 m/s , ou 108 km/h.

O fato do sistema entrar em regime, assegura a estabilidade do modelo desenvolvido. É válido pontuar que o valor dessa velocidade é algo simbólico. Visto que a velocidade depende de inúmeros fatores como a transmissão do veículo, as especificações do motor, entre outras coisas, as quais não são o foco desse trabalho. O mais importante para o estudo em questão, é assegurar que os polos do sistema estejam devidamente alocados e que esse modelo apresente as devidas semelhanças ao sistema real.

#### 4.1.2 Resposta no pedal de freio

Uma vez aplicada a entrada degrau em relação ao acelerador, foi realizado o mesmo procedimento em relação ao pedal de freio.



Fonte: autoria própria (2022).

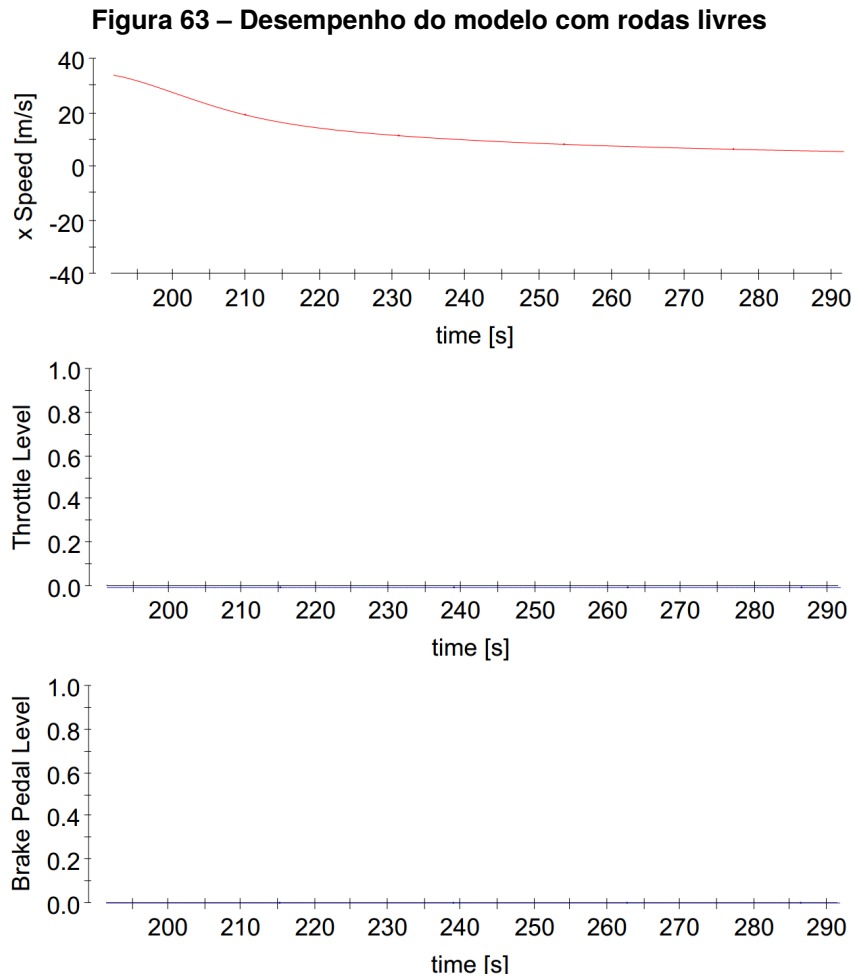
Assim como na Figura 61, a Figura 62 o gráfico superior reflete a saída, no caso a velocidade longitudinal e o gráfico inferior representa o sinal do pedal de freio.

O pedal de freio apresenta uma modelagem igual ao do caso anterior, todavia o seu acionamento gera um resultado que implica na ausência de velocidade longitudinal. A transição de estado, após o acionamento total do acelerador ocorreu de forma suave, e assim como esperado o veículo em questão se encontrou em repouso após o total acionamento do freio.

## 4.2 ANÁLISE DE RODA LIVRE

Após analisar a resposta do modelo ao degrau unitário. A próxima etapa foi validar o modelo em diferentes cenários. Foram impostas ao sistema algumas condições

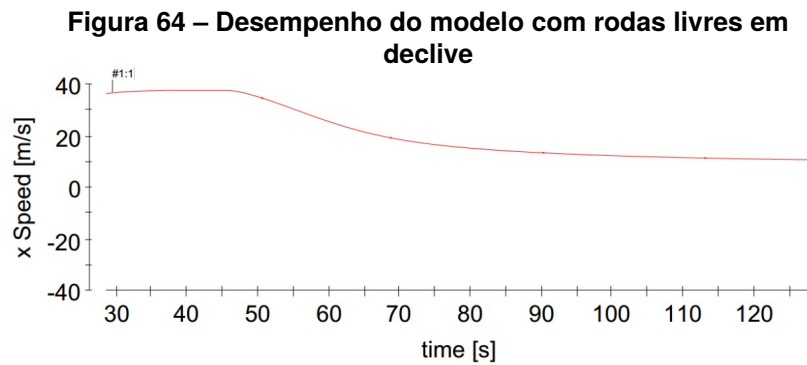
adversas no intuito de verificar a resposta do modelo em situações críticas. O objetivo desse cenário foi analisar em quanto tempo o veículo demoraria para parar, sem nenhum estímulo externo. O resultado da análise do modelo pode ser visto na Figura 63:



**Fonte: autoria própria (2022).**

Esta condição de teste contempla parâmetros como a resistência à rolagem e resistência aerodinâmica do modelo proposto. O teste proposto leva em consideração o relevo da pista, diferente do cenário da Figura 63 que representa uma pista plana.

A Figura 64 admite uma situação na qual o carro percebe um declive na via.



**Fonte: autoria própria (2022).**

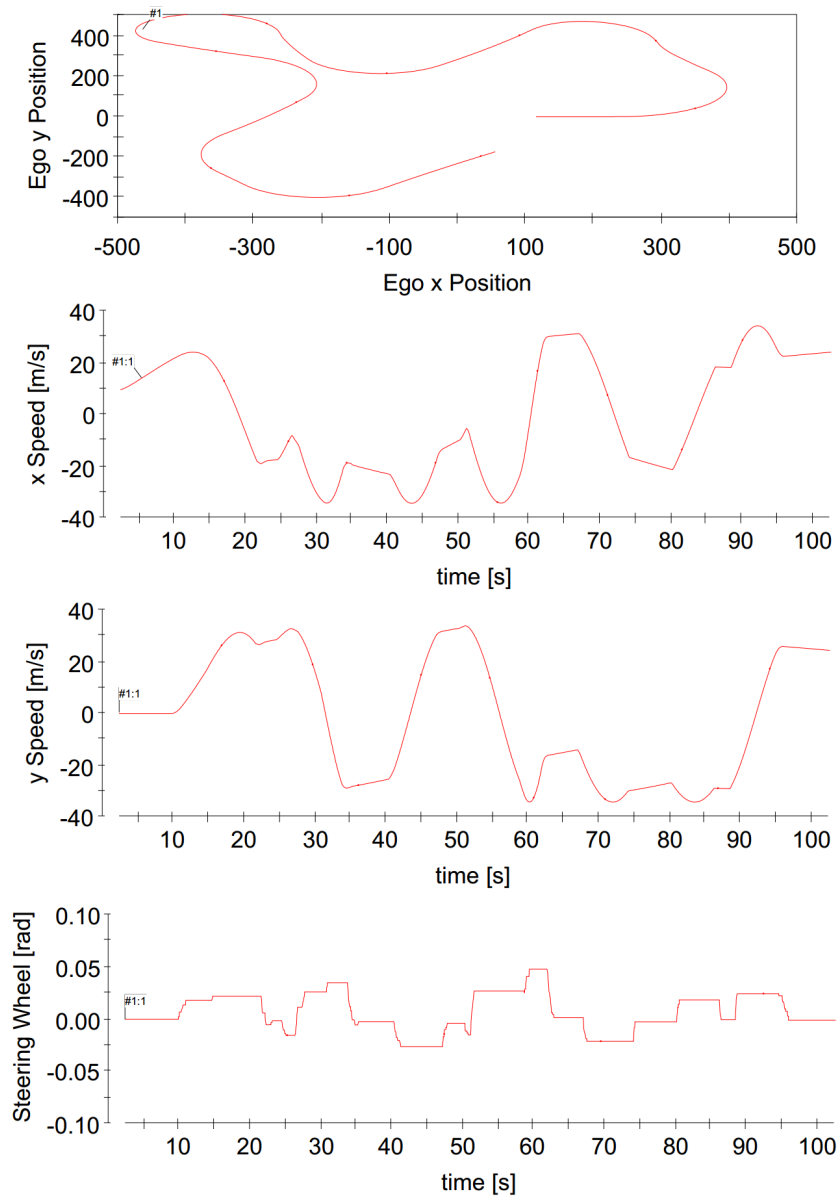
Conforme esperado, a condição de aclive/declive afeta bastante o modelo com rodas livres, uma vez que a gravidade passa a intervir de forma mais intensa na resistência à rolagem já presente no sistema.

### 4.3 ANÁLISE DE CAMINHO ARBITRÁRIO

O teste do caminho arbitrário, foi para analisar a mobilidade do modelo. O intuito desse cenário foi observar a resposta do sistema à diferentes estímulos, como variação no ângulo do volante, variação na posição no pedal do acelerador, entre outros.

A realização desse teste consistiu em instruções dadas ao longo da simulação e da observação da resposta fornecida pelo modelo proposto. Para assegurar a integridade do modelo, realizou-se essa verificação para 3 caminhos distintos. É possível ver nas Fig. 65, Fig. 66, Fig. 67:

**Figura 65 – Desempenho do modelo para caminho arbitrário 1**

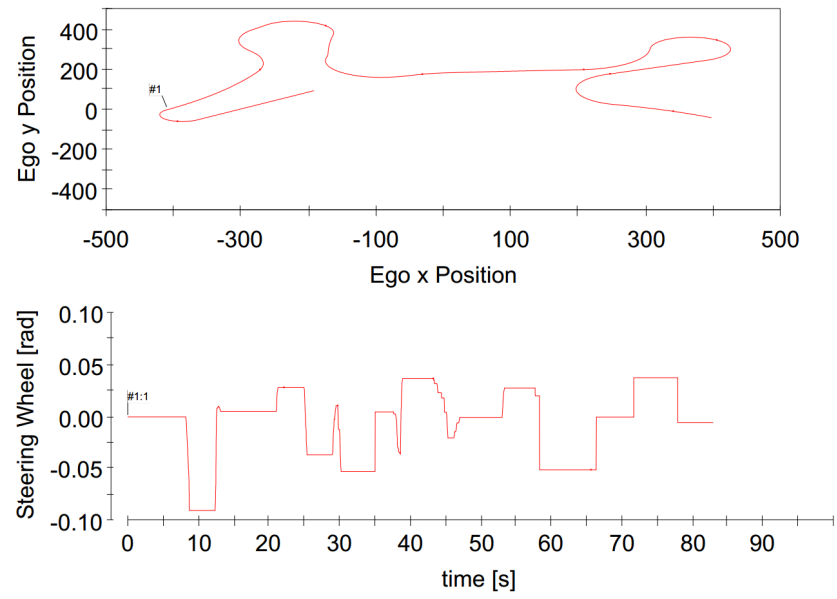


**Fonte: autoria própria (2022).**

Na Figura 65, observa-se que quando o ângulo do volante é diferente de 0, ou seja, quando o veículo está performando uma curva. O módulo velocidade expressa pelo sistema deverá ser decomposto no eixo X e eixo Y, para assim inferir um valor à velocidade longitudinal e lateral, respectivamente.

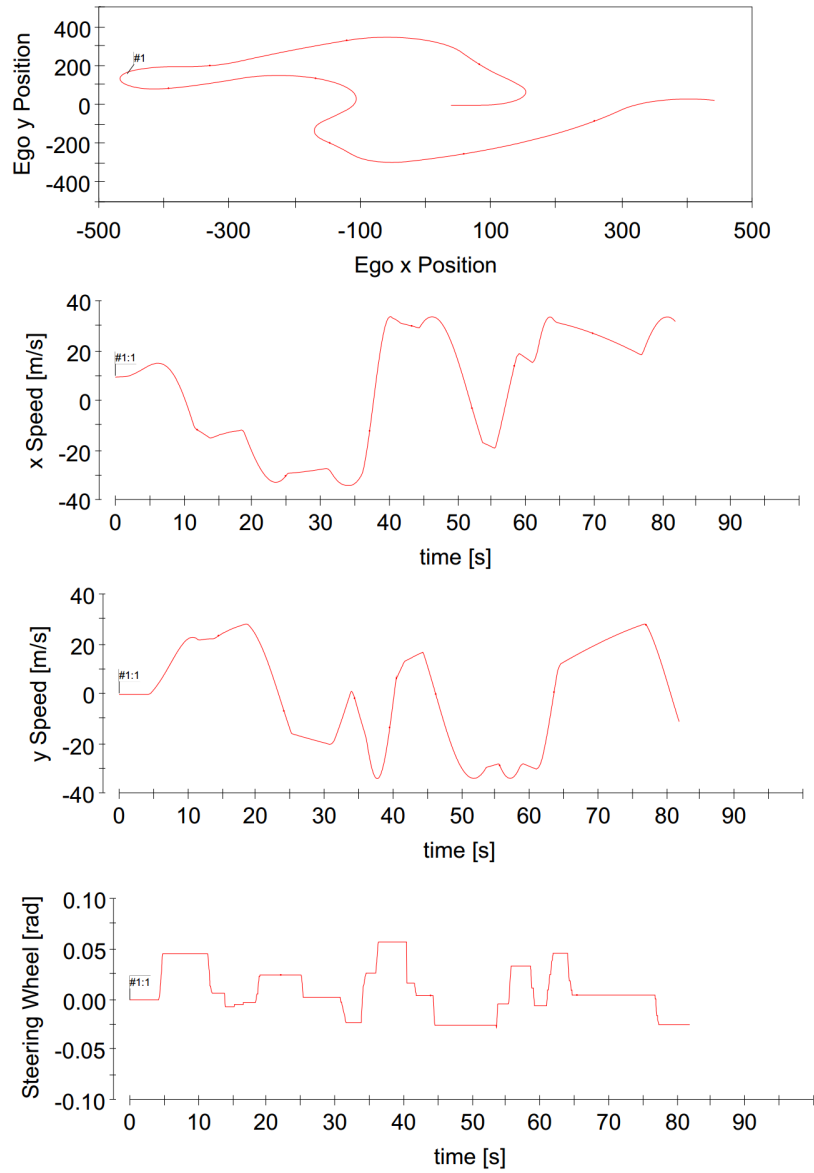


**Figura 66 – Desempenho do modelo para caminho arbitrário 2**



**Fonte: autoria própria (2022).**

**Figura 67 – Desempenho do modelo para caminho arbitrário 3**



**Fonte: autoria própria (2022).**

## 5 CONCLUSÕES

Ao longo do desenvolvimento do trabalho, houveram diversas situações desafiadoras nas quais foram necessárias a tomada de decisão que mantivesse um modelo fiel ao real e ao mesmo tempo fosse condizente com as limitações dos dispositivos utilizados na bancada.

### 5.1 PROBLEMAS ENFRENTADOS

Entre as inúmeros problemas, duas situações foram as mais notórias.

- Realizar a entrada da força motriz no modelo.
- Propor um modelo o qual os dispositivos utilizados, tivessem a capacidade de processar em tempo real.
- Utilização de ferramentas computacionais compatíveis com os equipamentos empregados no projeto.

#### 5.1.1 Entrada da força motriz

Afim de ter um modelo que se movimentasse livremente em todos sentido e direções do sistema de coordenadas proposto no trabalho. Foi necessário aplicar uma modelagem que fizesse a força que entrava no sistema, apresentar uma saída em forma de variação da posição do veículo.

Para resolver esse problema utilizou-se do modelo do pneu, descrito no Apêndice A, sendo este componente responsável pela transferência de força entre o veículo e o mundo externo.

Além de permitir modelar essa interação do veículo, o emprego do modelo do pneu foi essencial para tornar o sistema proposto em ambiente computacional parecido com o real, pois admitiu a condição de escorregamento. Condição essa fundamental para modelagem de dinâmica veicular.

### 5.1.2 Propor modelo que o hardware processe em tempo real

Como pontuado no Capítulo 3 existem inúmeros parâmetros a serem contemplados para realizar a devida exportação do modelo em ambiente computacional, para um ambiente embarcado.

A maneira encontrada para contornar essa situação foi a escolha do integrador, componente responsável pelas aproximações matemáticas que deverão ser realizadas no modelo. Outra solução adotada foi escrever o código descrito na obra de Pacejka ao invés de recorrer aos blocos fornecidos pela ferramenta computacional utilizada.

### 5.1.3 Ferramentas computacionais

Os dispositivos utilizados são suportados por versões obsoletas do *software* da *Mathworks*, logo buscar a documentação necessária para desenvolvimento deste projeto torna-se uma tarefa desafiadora. Sendo preciso implementar diversas funcionalidades manualmente, problema esse o qual a depender do equipamento disponível pode ser facilmente solucionado ou agravado.

## 5.2 TRABALHOS FUTUROS

O presente trabalho propõe um *framework* para desenvolvimento de estratégias de controles. Além de fornecer um modelo MIMO, que pode ser utilizado tanto para situações que solicitam um controle lateral, quanto para situações que precisam de controle longitudinal. Possíveis aplicações que podem ser desenvolvidas a partir dos resultados obtidos nesse projeto são:

- Implementar uma função de ACC, utilizando o modelo longitudinal.
- Desenvolver um algoritmo de LKA, empregando o modelo lateral.
- Propor uma estratégia de ultrapassagem, sendo essa uma situação que requer utilizar o modelo lateral e longitudinal do sistema.

## REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à identificação de sistemas**: Técnicas lineares e não lineares: Teoria e aplicação. Belo Horizonte, Minas Gerais, Brasil: Editora UFMG, 2015. 774 p. ISBN 978-85-423-0079-6.
- BOYCE, W. E.; DIPRIMA, R. C. Equações diferenciais ordinárias e problemas de valores de contorno. **Rio de Janeiro: LTC**, 1998.
- BRUNO, G.; MARCHETTO, G. Process-translatable Petri nets for the rapid prototyping of process control systems. **IEEE Transactions on Software Engineering**, SE-12, n. 2, p. 346–357, 1986. DOI: 10.1109/TSE.1986.6312948.
- CHIANG, H.-H. *et al.* Embedded Driver-Assistance System Using Multiple Sensors for Safe Overtaking Maneuver. **IEEE Systems Journal**, v. 8, n. 3, p. 681–698, 2014. DOI: 10.1109/JSYST.2012.2212636.
- CORRÊA, M. V. Identificação de sistemas dinâmicos não-lineares utilizando modelos narmax racionais: aplicação a sistemas reais. Universidade Federal de Minas Gerais, 1997.
- COUGHANOWR, D. R.; KOPPEL, L. B. **Análise e controle de processos**. [S. l.]: Guanabara Dois, 1978.
- DAL POGGETTO, G. **Critérios para seleção da arquitetura elétrica veicular em mercados emergentes**. 2009. Tese (Doutorado) – Universidade de São Paulo.
- DAVIDSON, P.; SPINOULAS, A. Autonomous vehicles: what could this mean for the future of transport. *In*: AUSTRALIAN Institute of Traffic Planning and Management (AITPM) National Conference, Brisbane, Queensland. [S. l.: s. n.], 2015.
- DSPACE. **ControlDesk**: Experiment Guide. Paderborn, Alemanha, nov. 2011. 134 p.
- DSPACE. **DS1104 RD Controller Board**: Hardware Instalation and Configuration. Paderborn, Alemanha, nov. 2011. 126 p. For DS1104 and CP1104/CLP1104 Connector Panels.
- DUMA, R. *et al.* Rapid prototyping of control systems using embedded target for TI C2000 DSP. *In*: 2007 Mediterranean Conference on Control Automation. [S. l.: s. n.], 2007. P. 1–5. DOI: 10.1109/MED.2007.4433860.
- FRANCESCHINI, G.; MACCHIETTO, S. Model-based design of experiments for parameter precision: State of the art. **Chemical Engineering Science**, Elsevier, v. 63, n. 19, p. 4846–4872, 2008.
- GORDON, V.; BIEMAN, J. Rapid prototyping: lessons learned. **IEEE Software**, v. 12, n. 1, p. 85–95, 1995. DOI: 10.1109/52.363162.

HALLIDAY, R. Walker-Fundamentos de Física vol. 1. **Editora LTC Sears–Física**, v. 1, 2009.

HENKE, C.; MICHAEL, J. **X-in-the-Loop Entwicklungs- und Testumgebungen**. [S. l.: s. n.], jul. 2022. Site da Fraunhofer-Institut für Entwurfstechnik Mechatronik. Disponível em:

<https://www.iem.fraunhofer.de/de/ueber-uns/forschung/leistungsangebot/scientific-automation/X-in-the-LoopEntwicklungs-undTestumgebungen.html>.

Acesso em: 13 out. 2022.

HIETANEN, S. Mobility as a Service. **the new transport model**, v. 12, n. 2, p. 2–4, 2014.

ISO, D. 8855, Road Vehicles-Vehicle Dynamics and Road-Holding Ability-Vocabulary(ISO 8855:1991,modified). **DIN Deutsches Institut Für Normung e**, 2013.

JAIKAMAL, V. **Model-based ecu development–an integrated mil-sil-hil approach**. [S. l.], 2009.

JITTRAPIROM, P. *et al.* Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges, 2017.

LIMA, G. d. S. **Controle de sistemas dinâmicos não lineares com compensação por Processo Gaussiano**. 2019. Diss. (Mestrado) – Brasil.

MOTAHHIR, S. *et al.* MIL and SIL and PIL tests for MPPT algorithm. **Cogent Engineering**, Taylor & Francis, v. 4, n. 1, p. 1378475, 2017.

NEME, J. H. Z. *et al.* **Um estudo da aplicação do padrão AUTOSAR para desenvolvimento de funções em sistemas embarcados automotivos**. 2017. Diss. (Mestrado) – Universidade Tecnológica Federal do Paraná.

NUNES, L. R. *et al.* The Monte Carlo method for non-detected failure analysis on dynamic systems testing. *In*: IEEE. IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society. [S. l.: s. n.], 2017. P. 5084–5089.

OGATA, K.; SEVERO, B. **Engenharia de controle moderno**. [S. l.]: Prentice Hall do Brasil, 1998.

OSMAN, K.; GHOMMAM, J.; SAAD, M. Guidance based lane-changing control in high-speed vehicle for the overtaking maneuver. **Journal of Intelligent & Robotic Systems**, Springer, v. 98, n. 3, p. 643–665, 2020.

PACEJKA, H. B. **Tire and Vehicle Dynamics**. [S. l.]: Butterworth-Heinemann, abr. 2012. 632 p. ISBN 978-0-08-097016-5.

RAMOS, A. L.; FERREIRA, J. V.; BARCELÓ, J. Model-based systems engineering: An emerging approach for modern systems. **IEEE Transactions on Systems, Man, and**

**Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 42, n. 1, p. 101–111, 2011.

SCHREIBER, V. *et al.* Shared and distributed X-in-the-loop tests for automotive systems: Feasibility study. **IEEE Access**, IEEE, v. 6, p. 4017–4026, 2018.

WAN, J.; CANEDO, A.; AL FARUQUE, M. A. Functional model-based design methodology for automotive cyber-physical systems. **IEEE Systems Journal**, IEEE, v. 11, n. 4, p. 2028–2039, 2015.

WANG, D.; GANESAN, S. Automotive domain controller. *In*: IEEE. 2020 International Conference on Computing and Information Technology (ICCI-1441). [S. l.: s. n.], 2020. P. 1–5.

## **APÊNDICE A — MODELAGEM IMPLEMENTADA**



A modelagem proposta por (OSMAN; GHOMMAM; SAAD, 2020), para proposição do modelo lateral é:

$$\dot{X} = U_y \cos(\varphi) - U_x \sin(\varphi) \quad (4)$$

$$\dot{Y} = U_x \cos(\varphi) + U_y \sin(\varphi) \quad (5)$$

$$\dot{\varphi} = r \quad (6)$$

$$\dot{r} = \frac{-aC_f + bC_r}{I_z U_x} U_y - \frac{a^2 C_f + b^2 C_r}{I_z U_x} r + \frac{aC_f}{I_z} \delta \quad (7)$$

$$\dot{U}_y = -\frac{C_f + C_r}{m U_x} + \left( \frac{a^2 C_f + b^2 C_r}{m U_x} - U_x \right) r + \frac{aC_f}{m} \delta \quad (8)$$

$$\dot{U}_x = r U_y + \frac{F_x}{m} \quad (9)$$

Já a modelagem implementada com a finalidade de gerar o código a ser embarcado no modelo de HIL, contemplou 3 diferentes situações as quais são necessárias aplicar técnicas de controle. Sendo elas:

- Modelo do pneu, sendo este responsável pela entrada de energia ao sistema;
- Modelo Lateral, que explica a movimentação do veículo em relação ao eixo X;
- Modelo Longitudinal, movimentação em relação ao eixo Y.

#### A.1 MODELAGEM DO PNEU

A modelagem do pneu representa uma entidade que atua muitas vezes de forma contrária ao movimento. Para obter uma representação adequada desse componente primeiramente é necessário modelar o torque de entrada como visto em Equação (10).

$$T_e = throttle * (a_0 + a_1 * w_e + a_2 * w_e^2) \quad (10)$$

Onde *throttle* é a variação da pressão no pedal de aceleração e os demais coeficientes são estabelecidos no modelo.

O coeficiente  $\dot{w}_e$  representa a velocidade angular no motor, sendo esta grandeza utilizada em Equação (10) e definida pela relação presente em Equação (11):

$$\dot{w}_e = \frac{(T_e - GR * r_e * F_{load})}{J_e} \quad (11)$$

Por fim tem-se  $F_{load}$  a força contrária ao movimento, sendo estada equacionada conforme a seguinte relação:

$$F_{load} = F_{aero} + R_x + F_g + F_b \quad (12)$$

Onde  $F_{aero}$  é a força devido a resistência do ar,  $R_x$  a componente longitudinal de atrito,  $F_g$  resistência gravitacional e  $F_b$  advinda da ação de frenagem.

## A.2 MODELO LATERAL

O modelo lateral aplicado, buscou levar em consideração o angulo formado pelo volante em relação ao sistemas de coordenadas adotado para o modelo

$$yaw = \frac{(V * \cos(\beta) * \tan(d))}{l} \quad (13)$$

A velocidade no eixo X contemplando o modelo lateral é definida por Equação (14):

$$\dot{X} = V * \cos(yaw + \beta) \quad (14)$$

A velocidade no eixo Y contemplando o modelo lateral é definida por Equação (15):

$$\dot{Y} = V * \sin(yaw + \beta) \quad (15)$$

O coeficiente  $\beta$  empregado representa a diferença entre o centro do modelo em relação a roda dianteira do veículo.

### A.3 MODELO LONGITUDINAL

O modelo longitudinal pode ser representado através das Leis de Newton, sendo a aceleração definida conforme:

$$\dot{v}_x = \frac{(F_x - F_{load})}{m} \quad (16)$$

## **APÊNDICE B — CÓDIGOS IMPLEMENTADOS NA SIMULAÇÃO**

```

function w_e_dot = fcn(throttle, F_load, w_e)
a_0 = 400;
a_1 = 0.1;
a_2 = - 0.002;
GR = 0.35;
r_e = 0.3;
J_e = 10;
g = 9.81;
T_e = throttle * (a_0 + a_1 * w_e + a_2 * w_e * w_e);
w_e_dot = (T_e - GR * r_e * F_load)/J_e;
function [F_x, w_w] = fcn(w_e, v_x, w_w, brake_Pedal)
GR = 0.35;
r_e = 0.3;
c = 10000;
F_max = 10000;
w_w = GR * w_e;
if (brake_Pedal == 0)
s = (w_w * r_e - v_x)/(w_w * r_e + 1e-6);
else
s = (w_w * r_e - v_x)/(v_x + 1e-6);
end
if (s > 1), s = 1; end
if s > 1
F_x = F_max;
elseif (s <= 1 s >= 0)
F_x = c * s;
elseif (s < 0 s >= -1)
F_x = c * s;
else
F_x = -F_max;
end
function F_load = fcn(v_x, alpha, brake_Pedal)
c_a = 1.36;

```

```

c_r1 = 0.01;
m = 2000;
g = 9.81;
F_aero = c_a * v_x * v_x;
R_x = c_r1 * v_x;
F_g = m * g * sin(alpha);
if (v_x < 5)
F_b = brake_Pedal * 60 * v_x;
else
F_b = brake_Pedal * 500;
end
F_load = F_aero + R_x + F_g + F_b;
function v_x_dot = fcn(F_x, F_load)
m = 2000;
v_x_dot = (F_x - F_load)/m;
function [X_dot, Y_dot, yaw_dot] = fcn(V, d, yaw)
l_f = 1.4;
l_r = 1.6;
l = l_f + l_r;
Beta = atan((l_r * tan(d))/l);
yaw_dot = (V * cos(Beta) * tan(d))/l;
X_dot = V * cos(yaw + Beta);
Y_dot = V * sin(yaw + Beta);

```

**ÍNDICE REMISSIVO**

ABS, 16  
ACC, 35, 62, 75  
ADAS, 14, 20, 23, 35  
AP, 36

BCM, 16

CAN, 16  
CAN FD, 16

DSP, 27, 48, 49

ECM, 16  
ECU, 15–17, 26, 32, 33  
EVB, 32

FCW, 35, 62

GPS, 16

HIL, viii, 29, 33, 34, 37, 80

ICE, 24  
IMU, 23  
IPC, 16  
ISO, 17, 23, 26, 31

LDW, 35, 62  
LKA, 35, 62, 75

MAaS, 35  
MBD, 18, 25  
MIL, viii, 29–31  
MIMO, 21, 22, 61, 75  
MISRA, 31, 32

PCI, 48  
PIL, viii, 29, 32

RAM, 33  
RCP, 17, 18, 27–29, 58  
ROM, 33  
RTI, 46, 53, 56

SAE, viii, 23  
SIL, viii, 29–32

SISO, 21, 22

UTFPR, i

VA, 35

XIL, 17, 29