

Universidade Federal de Minas Gerais  
Escola de Engenharia  
Curso de Graduação em Engenharia de Controle e Automação

## **Controle Ótimo Aplicado em um Veiculo Elétrico de Competição**

Michael Feliphe da Silva Barbosa

Orientador: Prof. Dimas Abreu Archanjo Dutra, Dr.  
Supervisor: Prof. Víctor Costa da Silva Campos, Dr.

Belo Horizonte, Julho de 2020



## **Monografia**

### **Controle Ótimo Aplicado em um Veiculo Elétrico de Competição**

Monografia submetida à banca examinadora designada pelo Colegiado Didático do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Minas Gerais, como parte dos requisitos para aprovação na disciplina Projeto Final de Curso II.

Belo Horizonte, Julho de 2014

# Resumo

No presente trabalho estudamos

Por fim, determinamos um modelo matemático para o veículo elétrico e aplicamos a teoria de Controle Ótimo com o objetivo de minimizar a quantidade de energia elétrica consumida durante uma prova da competição.



# Abstract

Write a version of your “resumo” in English. Beware of literal translations and double check the translation of technical terms.



# Agradecimentos

Aqui vai o texto dos agradecimentos.



# Sumário

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Agradecimentos</b>	<b>v</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Justificativa . . . . .	1
1.2 Objetivos do Projeto . . . . .	2
1.3 Local de Realização . . . . .	2
1.4 Estrutura da Monografia . . . . .	3
<b>2 Revisão Bibliográfica</b>	<b>5</b>
2.1 Modelo do veículo . . . . .	5
2.1.1 Inércia rotacional . . . . .	6
2.1.2 Arrasto aerodinâmico . . . . .	6
2.1.3 Relevo da pista . . . . .	7
2.1.4 Resistência ao rolamento . . . . .	7
2.1.5 Sistema de propulsão . . . . .	8
2.2 Controle Ótimo . . . . .	12
2.2.1 Introdução teórica . . . . .	13
2.2.2 O <i>software</i> FALCON.m . . . . .	15
<b>3 Metodologia</b>	<b>17</b>
3.1 Modelo do veículo DT1 . . . . .	17
3.2 Modelo da pista . . . . .	19
3.3 Problema de controle ótimo (OPC) . . . . .	20
3.4 Código para solução do OPC . . . . .	20
<b>4 Resultados</b>	<b>23</b>

<b>5 Conclusões</b>	<b>25</b>
5.1 Considerações Finais . . . . .	25
5.2 Propostas de Continuidade . . . . .	25
<b>Referências Bibliográficas</b>	<b>26</b>
<b>A Códigos implementados</b>	<b>29</b>
<b>B Dados da pista</b>	<b>33</b>

# Listas de Figuras

1.1	Veículo elétrico protótipo DT1 . . . . .	1
2.1	Diagrama de forças de um veículo em movimento . . . . .	6
2.2	Legenda diagrama pneu . . . . .	8
2.3	Diagrama de blocos do sistema de propulsão de um veículo elétrico . .	9
2.4	Classificação de motores elétricos . . . . .	9
2.5	Círculo equivalente de um motor BLDC . . . . .	10
2.6	Motor Dunkermotoren BG75x75 40V . . . . .	11
2.7	Representação da transmissão do DT1 . . . . .	12
2.8	Exemplo do comportamento desejado das variáveis em um OPC . . . .	14
2.9	Visão geral dos métodos numéricos para controle ótimo . . . . .	15
2.10	Resumo com principais objetos da biblioteca FALCON.m . . . . .	16
3.1	Perfil de altitude da pista da Shell Eco-marathon Americas de 2019 . .	19
3.2	Arquivos MATLAB® do projeto . . . . .	21



# **Lista de Tabelas**

2.1	Comparação do $c_d$ de diferentes tipos veículos . . . . .	7
2.2	Comparação do coeficiente $c_r$ de diferentes pneus . . . . .	8
2.3	Dados do motor BG75x75 40V . . . . .	11
3.1	Constantes utilizadas no modelo do veiculo . . . . .	18

# Capítulo 1

## Introdução

Este capítulo explica a motivação e objetivos deste projeto de otimização do desempenho em competições de um veículo elétrico protótipo de alta eficiência. Apresenta a equipe de competição estudantil Milhagem UFMG onde este PFC foi realizado. E, também, descreve a estrutura desta monografia.

### 1.1 Motivação e Justificativa

A equipe Milhagem UFMG constrói protótipos de veículos para participar de competições de eficiência energética. Atualmente a equipe possui dois veículos: o M84 equipado com um motor a combustão interna a gasolina e o DT1, apresentado na Figura 1.1, que possui motor elétrico e bateria. As competições em que a equipe participa são a Shell Eco-Marathon Brasil (nacional) e a Shell EcoMarathon Americas (internacional). Nestas competições os veículos devem consumir a menor quantidade de energia para percorrer trajeto, ou seja, devem ter a maior eficiência energética.

Figura 1.1: Veículo elétrico protótipo DT1



Fonte: Equipe Milhagem UFMG

Os principais fatores que influenciam no consumo de energia do veículo são a aerodinâmica, o peso total, a resistência ao rolamento, o relevo do trajeto e a estratégia de pista. Essa estratégia consiste na forma e nos momentos em que o motor deve ser

acionado. Um exemplo de uma estratégia de pista muito utilizada nesses protótipos é a estratégia start-stop, na qual o motor é desligado quando a velocidade é maior que 30 km/h e religado apenas quando é menor que 20 km/h, semelhante a um controle on-off com histerese.

Durante a avaliação da eficiência energética de um protótipo ele deve seguir as seguintes restrições: posições inicial e final fixas, velocidade inicial nula, velocidade instantânea máxima e velocidade média mínima. Uma vez que essas restrições permitem inúmeras estratégias de pista tem-se a necessidade encontrar a estratégia que maximize a eficiência energética do veículo durante a avaliação de consumo.

## 1.2 Objetivos do Projeto

Tendo em vista o exposto acima, este projeto tem por objetivos direcionados ao protótipo DT1:

- (a) Definir o modelo matemático para a dinâmica do protótipo;
- (b) Formular o problema de controle ótimo (OCP) pra obter a estratégia ótima;
- (c) Implementar o algoritmo para solução desse OCP;
- (d) Definir a estratégia ótima para a pista da Shell EcoMarathon Américas de 2019.

## 1.3 Local de Realização

Esse projeto foi desenvolvido na equipe de competição Milhagem UFMG, na qual o autor foi integrante de 2013 à 2015 e em 2018. A equipe é composta por alunos de graduação em engenharia de diversos períodos e sua sede é no Departamento de Engenharia Mecânica da UFM. Foi fundada, sobre orientação do professor Paulo Iscold, em 2005 no Centro de Estudos Aeronáuticos (CEA) do qual fez parte até 2006. De 2006 a 2011 o projeto da equipe ficou suspenso retornando as atividades, sobre orientação do professor Fabrício Pujatti, no Centro de Tecnologia de Mobilidade (CTM). A equipe já desenvolveu 6 veículos, sendo DT1 o primeiro elétrico, e participou de 10 competições com os seguintes resultados:

- Maratona Universitária de Eficiência Energética
  - Categoria gasolina
    - \* 2005: 2º Lugar, com a marca de 227,6 [km/L]
    - \* 2006: 1º Lugar, com a marca de 598,9 [km/L]
    - \* 2011: 5º Lugar, com a marca de 199,0 [km/L]
    - \* 2013: 4º Lugar, com a marca de 234,9 [km/L]

- Shell Eco-marathon Brasil
  - Categoria gasolina
    - \* 2016: 2º Lugar, com a marca de 196,0 [km/L]
  - Categoria elétrico
    - \* 2017: 3º Lugar, com a marca de 315,6 [km/kWh]
    - \* 2018: 1º Lugar, com a marca de 266,4 [km/kWh]
- Shell Eco-marathon Americas
  - Categoria elétrico
    - \* 2018: 6º Lugar, com a marca de 266,5 [km/kWh]
    - \* 2019: 2º Lugar, com a marca de 226,9 [km/kWh]

## 1.4 Estrutura da Monografia

O relatório está dividido em cinco capítulos. Este capítulo apresentou uma introdução ao projeto a ser descrito nos capítulos seguintes. O Capítulo 2 é uma revisão bibliográfica que dos princípios básicos de modelagem veicular e controle ótimo de forma que abrange todos os conceitos necessários para um melhor entendimento do projeto. O Capítulo 3 aborda a metodologia de desenvolvimento do modelo matemático do DT1 e de implementação implementação do software para otimização da estratégia de pista. Os resultados obtidos no projeto são apresentados no Capítulo 4 e no Capítulo 5 tem-se a conclusão da monografia com algumas sugestões para trabalhos futuros e dificuldades encontradas durante a realização do projeto.



# Capítulo 2

## Revisão Bibliográfica

Este capítulo é dividido em duas seções que visam apresentar os conceitos necessários para a compreensão deste PFC. Na primeira seção são apresentados conceitos para uma modelagem matemática da dinâmica de um veículo em baixas velocidades. São apresentados na segunda seção uma introdução à teoria de controle ótimo e a utilização de *software* na solução de problemas de controle ótimo.

### 2.1 Modelo do veículo

Um modelo matemático, ou apenas modelo, é um conjunto de equações que descreve de forma adequada o comportamento de um sistema que deseja-se estudar. Uma forma usual de classificação dos métodos de modelagem é separá-los nas categorias: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza. A modelagem caixa branca, também conhecida como modelagem conceitual, consiste na aplicação princípios fundamentais e por isso exige um conhecimento da natureza do sistema. A modelagem caixa preta, ou modelagem empírica, é baseada na aplicação de técnicas de identificação de sistemas que exigem pouco ou nenhum conhecimento do sistema. Já na modelagem caixa cinza são utilizadas técnicas que estão entre a modelagem caixa branca e a modelagem caixa preta<sup>[1]</sup>.

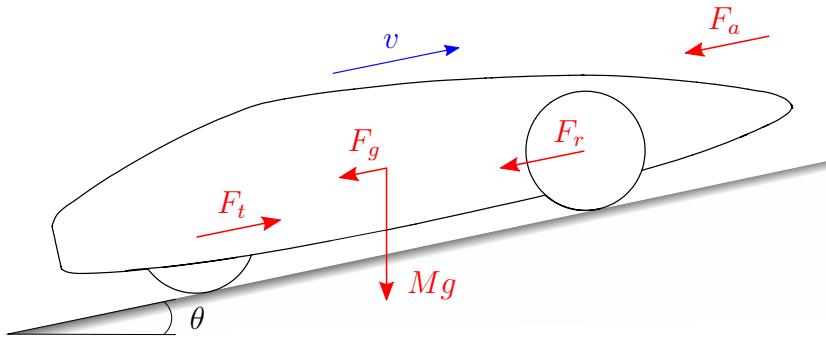
Nessa seção o método de modelagem aplicado é de modelagem caixa branca. A partir da aplicação da segunda lei de Newton no veículo representado no diagrama da Figura 2.1, obtém-se a equação que descreve a dinâmica longitudinal do mesmo

$$(m_v + m_p + m_r)\dot{v} = F_t - (F_a + F_g + F_r) , \quad (2.1)$$

em que  $m_v$  é a massa do veículo,  $m_p$  é a massa do piloto,  $m_r$  é massa equivalente ao momento de inércia das partes rotativas (rodas e eixo do motor),  $x$  é a posição,  $F_t$  é a propulsão feita pelo motor,  $F_a$  é o arrasto aerodinâmico,  $F_g$  é a componente do

peso que esta direção da velocidade e  $F_r$  é a resistência ao rolamentos dos pneus no pista. Os modelos que descrevem a massa equivalente  $m_r$  e forças  $F_a$ ,  $F_g$ ,  $F_r$  e  $F_t$  estão apresentados nas subseções a seguir.

Figura 2.1: Diagrama de forças de um veículo em movimento



Fonte: Elaborado pelo autor.

### 2.1.1 Inércia rotacional

A inércia de todas as peças giratórias dentro do veículo causa forças fictícias. Esta inércia pode ser representada nos modelos dinâmicos do motor e da transmissão, ou ser representada com uma massa equivalente  $m_r$ . Para essa representação é considerado que não há deslizamento do pneu no asfalto e não a escorregamento na transmissão. O cálculo de  $m_r$  é

$$m_r = m_{r,r} + m_{r,m} , \quad (2.2a)$$

$$m_{r,r} = \sum_{n=1}^{Nr} J_r \cdot \frac{1}{r_r^2} , \quad (2.2b)$$

$$m_{r,m} = J_m \cdot \frac{\varphi^2}{r_r^2} , \quad (2.2c)$$

em que  $m_{r,r}$  é a massa equivalente a inércia das rodas e  $m_{r,m}$  é a massa equivalente a inércia do motor,  $Nr$  é a quantidade de rodas,  $J_r$  e  $r_r$  são o momento de inércia e o raio da roda,  $J_m$  é o momento de inércia do motor e  $\varphi$  é a relação de transmissão.<sup>[2]</sup>

### 2.1.2 Arrasto aerodinâmico

O movimento de um objeto imerso em um fluido sofre uma resistência causada por esse fluido. No caso de veículos que se deslocando no ar, essa resistência é chamada de arrasto aerodinâmico. Pode-se aproximar o cálculo dessa força  $F_a$  com a equação

$$F_a(v) = \frac{\rho \cdot a_f \cdot c_d \cdot v^2}{2} , \quad (2.3)$$

em que  $v$  é a velocidade do veículo em relação ao vento,  $\rho$  a densidade do ar,  $a_f$  a área frontal do veículo e  $c_d$  o coeficiente de arrasto aerodinâmico. O coeficiente  $c_d$  é um número adimensional e depende da geometria veículo, é determinado por meio de simulações em *software* de fluido dinâmica computacional (CFD, do inglês *computational fluid dynamics*) e/ou experimentos em túnel de vento<sup>[2]</sup>. Alguns valores típicos de  $C_d$  para diferentes tipos de veículos são apresentados na Tabela 2.1.

Tabela 2.1: Comparação do  $c_d$  de diferentes tipos veículos

Veículo	$c_d$
Carro	0,3 - 0,4
Ônibus	0,6 - 0,7
Caminhão	0,6 - 1,0
Moto	0,5 - 1,0

Fonte: Adaptado de Popp e Schiehlen<sup>[3]</sup>.

### 2.1.3 Relevo da pista

A componente do peso,  $F_g$ , afeta consideravelmente a dinâmica do veículo e atua sempre que a pista não é plana. Seu modelo é a equação

$$F_g(\theta) = (m_v + m_p) \cdot g \cdot \sin(\theta) , \quad (2.4)$$

que, pra pequenas inclinações, pode ser aproximado pela equação

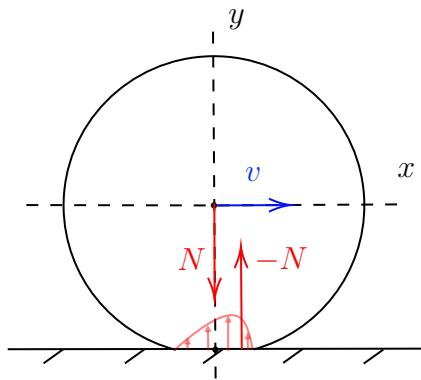
$$F_g(\theta) \approx (m_v + m_p) \cdot g \cdot \theta ,$$

em que  $m$  é a massa total do veículo,  $g$  é a aceleração da gravidade e  $\theta$  é a inclinação da pista expressa em radianos<sup>[2]</sup>.

### 2.1.4 Resistência ao rolamento

A norma ISO 4223-1:2017 define a resistência ao rolamento de um pneu, como a energia consumida pelo pneu por unidade de distância percorrida. Esse consumo de energia se deve principalmente as propriedades viscoelásticas dos compostos de borracha presente no pneu. Durante a rolagem o pneu é deformado na zona de contato entre o pneu e o pavimento, nessa zona de contato a resultante da força de reação à força normal não está no mesmo eixo que a força normal, Figura 2.2, de forma a gerar um força,  $F_r$ , contraria a movimentação do pneu.

Figura 2.2: Legenda diagrama pneu



Fonte: Elaborado pelo autor.

A força de resistência ao rolamento,  $F_r$ , depende da construção do pneu e do tipo de pavimento. Essa força também depende da velocidade do veículo e da pressão do ar no pneu, embora nesse trabalho não considera-se essa dependência. Para calcula-la usa-se a relação:

$$F_r = c_r \cdot N , \quad (2.5)$$

em que  $c_r$  é o coeficiente de resistência ao rolamento,  $N$  é a força normal sobre o pneu e  $F_r$  é a força gerada pela resistência ao rolamento. Estão apresentados na Tabela 2.2 o valor do coeficiente  $c_r$  para pneus de uso típico em carros de passeio, bicicletas e de dois pneus específicos para a competição SEM, Michelin® 45-406 Prototype (Figura ??) e Michelin® Radial 45-75 R16.

Tabela 2.2: Comparação do coeficiente  $c_r$  de diferentes pneus

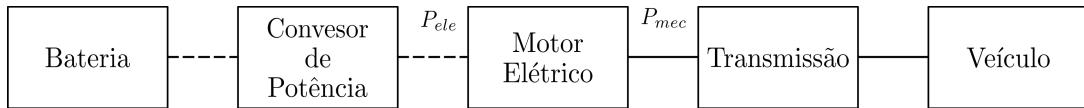
Pneu	$c_r$
Usado em carro	0,013
Usado em bicicleta	0,006
Michelin® 45-406 Prototype	0,0024
Michelin® Radial 45-75 R16	0,00081

Fonte: Adaptado de Santin et al.<sup>[4]</sup>.

### 2.1.5 Sistema de propulsão

De forma genérica, o sistema de propulsão de um veículo elétrico, representado no diagrama da Figura 2.3, é composto por bateria, conversor de potencia, motor elétrico e transmissão.

Figura 2.3: Diagrama de blocos do sistema de propulsão de um veículo elétrico



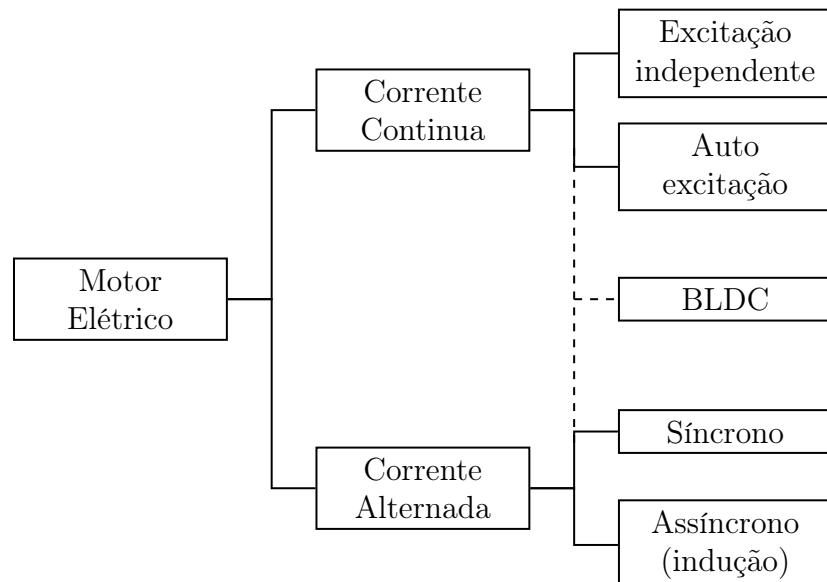
Fonte: Elaborado pelo autor.

Bateria eletroquímica, ou apenas bateria, é um dispositivo em que durante a descarga ocorre a conversão de energia potencial química em energia elétrica e na carga ocorre a conversão inversa. Ou seja, uma bateria armazena energia elétrica na forma de energia potencial química. Uma bateria é composta por vários células ligadas entre se. Uma célula de bateria é basicamente composta por dois eletrodos – positivo e negativo – imersos em um eletrólito<sup>[5]</sup>.

Conversor eletrônico de potência, ou conversor de potência, é o circuito cujo a finalidade é extrair energia elétrica de um sistema de energia e transformá-la em uma forma adequada e necessária para um motor<sup>[6]</sup>.

O motor elétrico converte a potencia elétrica – tensão e corrente – em potencia mecânica rotativa – torque e rotação – para impulsionar o veículo.<sup>[5]</sup> Podem ser classificados em relação a sua alimentação: corrente continua (CA) ou corrente alternada (CC), conforme apresentado na Figura 2.4. No entanto o motor de corrente continua sem escovas, ou BLDC do inglês *brushless direct current*, é difícil de ser classificado desse forma pois sua configuração é semelhante à de um motor CA, enquanto suas características elétricas são semelhantes às de um motor CC.<sup>[6]</sup> Nesse trabalho estudara-se o modelo dos motores BLDC pois é o tipo de motor utilizado no protótipo DT1.

Figura 2.4: Classificação de motores elétricos

Fonte: Adaptado de Kim<sup>[6]</sup>

O motor BLDC foi desenvolvido em 1962 e possui um sistema de comutação eletrônica ao invés de comutação mecânica como os motores CC. Para não utilizar as escovas da comutação mecânica, os enrolamentos da armadura são colocados no estator (parte estacionária) e os ímãs são colocados no rotor (parte rotativa). A comutação eletrônica baseia-se em sensores para identificar a posição do rotor e acionar o enrolamento necessário para manter o movimento de rotação.<sup>[6]</sup> O conjunto de equações que descreve a corrente de armadura  $i_a$  e o torque gerado  $T$  para um motor BLDC de enrolamentos ligados em Y com neutro acessível e operando em meia onda (a tensão é aplicada entre o terminal de um enrolamento e o neutro) é

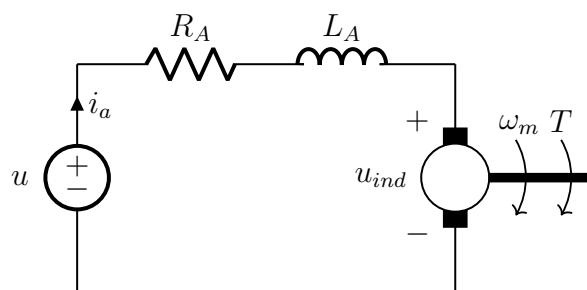
$$u - u_{ind} = L_a \cdot \dot{i}_a + R_a \cdot i_a , \quad (2.6a)$$

$$u_{ind} = K_v \cdot \omega_m , \quad (2.6b)$$

$$T_m = K_t \cdot i_a , \quad (2.6c)$$

em que  $u$  é a tensão aplicada,  $u_{ind}$  é a tensão induzida,  $L_a$  e  $R_a$  são, respectivamente a indutância e a resistência da armadura,  $K_v$  é a constante de tensão induzida,  $\omega_m$  é a rotação do motor e  $K_t$  é a constante de torque. A Figura 2.5 apresenta o modelo o circuito de um enrolamento do motor BLDC.<sup>[7]</sup>

Figura 2.5: Circuito equivalente de um motor BLDC



Fonte: Elaborado pelo autor.

O protótipo DT1 é equipado com um motor Dunkermotoren BG75x75 40V (Figura 2.6). O motor que possui 3 enrolamentos, ímã de neodímio com 8 polos e sensor hall integrado para medir a posição do rotor (24 pulsos por volta), suas características elétricas e mecânicas estão apresentadas na Tabela 2.3.

Figura 2.6: Motor Dunkermotoren BG75x75 40V



Fonte: Adaptado de

Tabela 2.3: Dados do motor BG75x75 40V

Tensão nominal	[V]	40
Corrente nominal	[A]	15,6
Torque nominal	[N.m]	1,50
Velocidade nominal	[rpm]	3370
Troque de atrito	[N.m]	0,13
Torque de parada	[N.m]	12
Velocidade sem carga	[rpm]	4100
Potencia de saída nominal	[W]	530
Potencia de saída máxima	[W]	1150
Constante de torque	[N.m/A]	0,119
Resistência	[Ω]	0,07
Indutância	[mH]	0,45
Corrente de pico	[A]	63
Inercia do rotor	[g.m²]	0,0625
Massa	[kg]	2,8

Fonte: Adaptado de

A transmissão do veículo regula a transferência de potência do motor para as rodas. É basicamente composta por mecanismo de redução – ex. caixa de câmbio – e por um mecanismo de interrupção – ex. embreagem<sup>[5]</sup>. No DT1, o mecanismo de redução utilizado é do tipo roda de atrito e o mecanismo de interrupção é o pivotamento entre os componentes da roda de atrito, apresentado na Figura 2.7.

Rodas de atrito são uma das maneiras mais simples para se transmitir potência mecânica entre eixos e sua eficiência é tipicamente de 95 a 98 %. São composta por dois ou mais cilindros em contato direto. Para não ocorrer deslizamento em uma roda de atrito o torque transmitido não deve exceder a força de atrito entre os cilindros. Os torque e rotações são descritos pelas equações

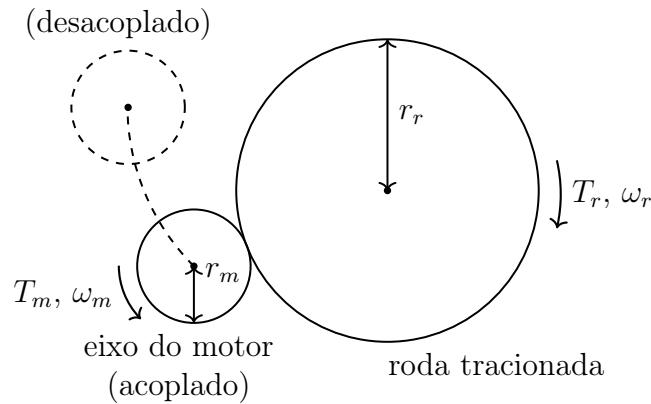
$$\varphi = \frac{r_r}{r_m} , \quad (2.7a)$$

$$\omega_r = -\frac{\omega_m}{\varphi} , \quad (2.7b)$$

$$T_r = \varphi \cdot T_m \cdot \eta , \quad (2.7c)$$

onde  $\varphi$  é a relação de transmissão,  $R_r$  e  $\omega_r$  são o diâmetro e a velocidade angular da roda,  $R_m$  e  $\omega_m$  são o diâmetro e a velocidade angular do cilindro motor e  $\eta$  é a eficiência da roda de atrito.<sup>[8, 9]</sup>

Figura 2.7: Representação da transmissão do DT1



Fonte: Elaborada pelo autor.

A partir das Equações (2.6) e (2.7), desconsiderando as perdas por atrito visco nos rolamentos, as perdas causadas pela resistência interna na bateria e a perda no conversor de potência, o conjunto que descreve a força de propulsão  $F_t$  de um veículo elétrico é

$$\dot{i}_a = \frac{u - K_v \cdot \omega_m - R_a \cdot i_a}{L_a} , \quad (2.8a)$$

$$F_t = \frac{K_t \cdot i_a \cdot \eta}{r_m} . \quad (2.8b)$$

## 2.2 Controle Ótimo

"O objetivo da teoria de controle ótimo é determinar os sinais de controle que farão com que um processo satisfaça as restrições físicas e ao mesmo tempo minimize (ou maximize) alguns critérios de desempenho."<sup>[10]</sup>

### 2.2.1 Introdução teórica

O conjunto de Equações (3.3) é uma formulação genérica e comum para um problema de controle ótimo (OCP)

$$\min_{x(\cdot), u(\cdot), p, T} \int_0^T L(x(t), u(t), p) dt + E(x(T), p) \quad (2.9a)$$

$$\text{s.t.} \quad x(0) - x_0 = 0, \quad (2.9b)$$

$$\dot{x}(t) - f(x(t), u(t)) = 0, \quad t \in [0, T], \quad (2.9c)$$

$$h(x(t), u(t)) \leq 0, \quad t \in [0, T], \quad (2.9d)$$

$$r(x(T)) \leq 0, \quad (2.9e)$$

$$T \leq T_{max}, \quad (2.9f)$$

$$p_{min} \leq p \leq p_{max} \quad (2.9g)$$

em que a Equação (3.3a) é o função objetivo, Equação (3.3c) é uma restrição de estados inciais fixos, Equação (3.3e) é a restrição que representa a dinâmica do sistema, Equação (3.3f) são restrições de caminho sobre os estados do sistema e sobre as variáveis de controle, Equação (3.3i) é uma restrição de espaço para os estados finais, Equação (3.3j) é a restrição para o tempo final e Equação (3.3k) são restrições de intervalo para parâmetros otimizáveis. O função objetivo, também conhecido como objetivo de Bolza, é composto por uma integral de  $L(x, u)$  conhecida como termo de Lagrange e uma função  $E(x)$  conhecida como termo de Meyer<sup>[11]</sup>.

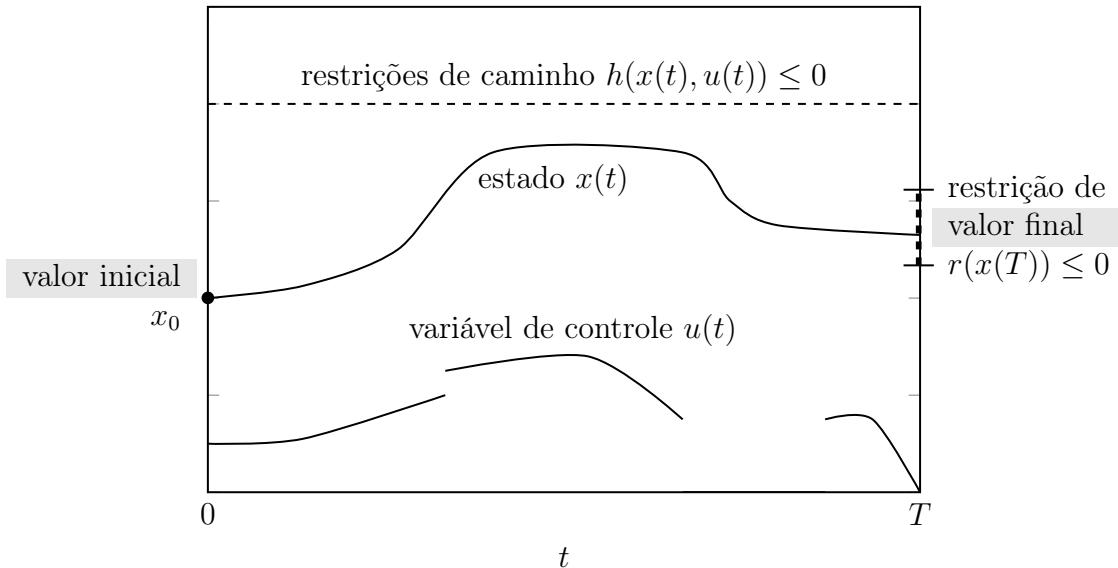
A solução de um problema de controle ótimo é uma sequencia de sinais de controle em malha aberta e um conjunto de parâmetros, o comportamento esperado é exemplificado na Figura 2.8.

De forma geral, há três abordagens básicas para solução computacional de um OPC, (a) programação dinâmica, (b) métodos indiretos e (c) métodos diretos, conforme apresentado na Figura 2.9.

- (a) Programação dinâmica: utiliza o princípio de otimalidade de Bellman (em um caminho ótimo A-B-C os caminhos A-B e B-C também são ótimos) para calcular recursivamente a sequencia de controle ótimo.
- (b) Métodos indiretos: utiliza a filosofia "primeiro otimizar e então discretizar", que é escrever as condições de otimalidade contínuas primeiro, resultando em um problemas de valor de contorno (sistema de equações diferenciais acrecido de um conjunto adicional de restrições chamadas condições de contorno) e depois discretizá-lo para resolver numericamente.
- (c) Métodos diretos: utiliza a filosofia "primeiro discretizar e então otimizar", isto é primeiro discretize as equações do OPC e, em seguida, aplique um algoritmo de

otimização para resolver o problema de programação não linear resultante (NPL do inglês *Nonlinear Programming Problem*) resultante.

Figura 2.8: Exemplo do comportamento desejado das variáveis em um OPC



Fonte: Adaptado de

Para a solução de OPC em aplicações no mundo real, os métodos diretos são atualmente as técnicas mais difundidas e usadas com sucesso. [12, 13]

Métodos diretos reformulam o OPC da Equação (3.3) em um problema de programação não linear de dimensão finita (NLP) da forma:

$$\min_w a(w) \quad (2.10a)$$

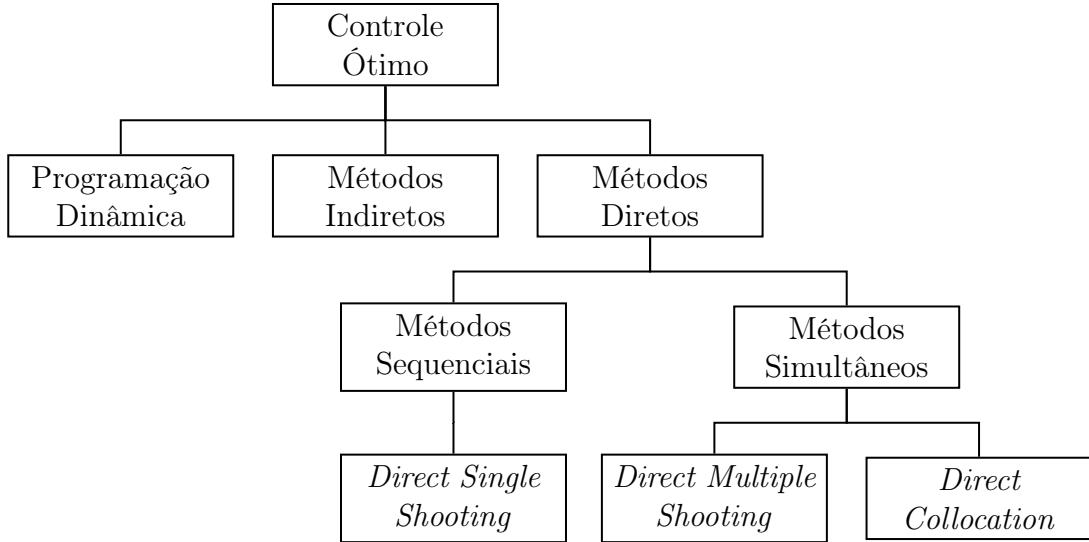
$$\text{s.t. } b(w) = 0, \quad (2.10b)$$

$$c(w) \geq 0, \quad (2.10c)$$

com um vetor de dimensão finita  $w$  representando os graus de liberdade de otimização e com funções diferenciáveis  $a$  (escalar),  $b$  e  $c$  (vetores). Todos os métodos diretos começam por uma parametrização da variável de controle, mas diferem na forma como os estados são tratados. Podem ser classificados em duas abordagens, seqüenciais e simultâneos.

Nas abordagens seqüenciais, os estados  $x(t)$  são uma função implícita dos controles  $u(t)$  e do valor inicial  $x_0$ . Dessa forma, a simulação e as iterações de otimização procedem seqüencialmente e o NLP possui apenas o controle discretizado como graus de liberdade de otimização. O método mais usado de abordagem seqüencial é o *Direct Single Shooting*.

Figura 2.9: Visão geral dos métodos numéricos para controle ótimo



Fonte: Adaptado de Diehl et al.<sup>[12]</sup>

As abordagens simultâneas fazem uma parametrização dos estados como variáveis de otimização dentro do NLP e adicionam restrições de igualdade que representam a dinâmica do sistema. Com isso, a simulação e a otimização ocorrem simultaneamente, e somente na solução do NLP os estados representam uma solução da equação da dinâmica do sistema válida e correspondente à variável de controle. As duas variações mais comuns da abordagem simultânea são *Direct Multiple Shooting* e *Direct Collocation*.

### 2.2.2 O software FALCON.m

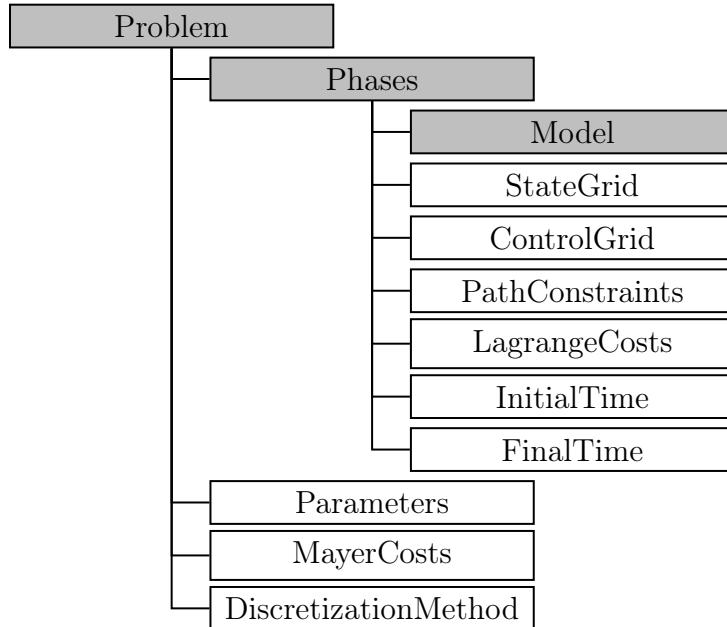
FALCON.m é um é uma biblioteca de *software*, proprietária de uso gratuito, para MATLAB®, desenvolvida no *Institute of Flight System Dynamics* da *Technische Universität München* para resolução e análise de problemas de controle ótimo e estimativa de parâmetros utilizando o método *Direct Collocation*. Capaz de resolver grandes problemas de controle ótimo com de até 600 mil variáveis de otimização.<sup>[14, 15]</sup>

Ela segue o paradigma de programação orientada a objetos, um resumo de suas principais classes de objetos é apresentado na Figura 2.10. A classe **Problem** é a principal, nela são configurados as fases (**Phases**) do OPC, os parâmetros otimizáveis, o custo de Mayer e o método de discretização a ser utilizado. Os métodos de discretização disponíveis na biblioteca são: trapezoidal (padrão) ou *backward Euler*. Na classe **Phases**, são adicionados o modelo da dinâmica do sistema, os estados, as variáveis de controle, as restrições de caminho, o custo de Lagrange e os instantes de tempos inicial e final.<sup>[16]</sup>

A biblioteca utiliza otimizadores para NLP externos, Como cada otimizador tem uma interface única, no momento, é possível utilizar interfaces para três otimizadores: IPOPT, SNOPT e WORHP. O otimizador utilizado neste trabalho é o IPOPT, configuração

pré-definida da biblioteca FALCON.m. Mais informações sobre a utilização do FALCON.m pode ser constuldas em seu manual Rieck et al.<sup>[14]</sup>.

Figura 2.10: Resumo com principais objetos da biblioteca FALCON.m



Fonte: Adaptado de Bittner<sup>[16]</sup>

## IPOPT

O projeto COIN-OR, do inglês *COmputational INfrastructure for Operations Research*, é uma iniciativa que visa promover *software* de código aberto para a comunidade de pesquisa operacional<sup>[17]</sup>. Dentre as bibliotecas disponibilizadas no projeto COIN-OR está a IPOPT, do inglês *Interior Point OPTimizer*. Escrita em C++ e Fortran para solução de problemas de otimização não linear (NLP) utilizando o método de pontos interiores com filtro apresentado por Wächter e Biegler<sup>[18]</sup>. Sua licença é a EPL que permite uso gratuito e acesso ao código. Mais detalhes sobre esta biblioteca podem ser obtidos na pagina do seu repositório no GitHub® <<https://github.com/coin-or/Ipopt>>.

# Capítulo 3

## Metodologia

É apresentado neste capítulo o desenvolvimento realizado neste projeto, os problemas encontrados e as soluções propostas. Na primeira seção tem-se a definição do modelo matemático do veículo protótipo DT1. A segunda seção traz a formulação proposta para o problema de controle ótimo que encontrar a estratégia de pista ótima. Por fim, na última seção é apresentado os passos realizados no desenvolvimento do código que soluciona o OPC proposto.

### 3.1 Modelo do veículo DT1

O modelo da dinâmica do veículo protótipo DT1 foi definido com base na revisão bibliográfica apresentada na Seção 2.1. Para isto as seguintes simplificações e considerações foram feita:

- **Dinâmica lateral:** não foi levando em conta a dinâmica lateral do veículo considerando que ele se move apenas em linha reta;
- **Vento:** a velocidade do vento foi desconsiderada na Equação (2.3);
- **Pneus:** a variação do coeficiente de rolamento em função da temperatura, da pressão e a velocidade não foi considerada na Equação (2.5);
- **Rolamentos:** o atrito viscos dos rolamentos foi considerado nulo;
- **Perdas elétricas:** a bateria e o conversor eletrônico de potência foram tratados como componentes ideias não apresentando perdas de energia ou modificações na dinâmica das grandezas elétricas;
- **Indutância do motor:** como o dinâmica da velocidade é pelo menos dez vez mais lenta que a dinâmica da corrente elétrica no motor a contribuição do indutor foi removida da Equação (2.8a).

A partir das Equações (2.1) à (2.5) e (2.8) e destas simplificações e considerações foi definido a seguinte equação para descrever o comportamento do veículo

$$(m_v + m_p + m_r)\dot{v} = F_t - (F_a + F_g + F_r) , \quad (3.1a)$$

$$m_r = N_r \cdot \frac{J_r}{r_r^2} + \frac{J_m}{r_m^2} , \quad (3.1b)$$

$$F_t = \frac{K_t \cdot i_a \cdot \eta}{r_m} , \quad (3.1c)$$

$$R_a \cdot i_a = u - \frac{K_v \cdot v}{r_m} , \quad (3.1d)$$

$$F_a = \frac{\rho \cdot a_f \cdot c_d \cdot v^2}{2} , \quad (3.1e)$$

$$F_g = (m_v + m_p) \cdot g \cdot \sin(\theta(x)) , \quad (3.1f)$$

$$F_r = c_r \cdot (m_v + m_p) \cdot g \cdot \cos(\theta(x)) , \quad (3.1g)$$

em que as constantes estão apresentadas na Tabela 3.1 e as variáveis são  $\dot{v}$  a aceleração do carro em  $[m/s^2]$ ,  $v$  a velocidade do carro em  $[m/s^2]$ ,  $x$  a distância percorrida pelo carro na pista em  $[m]$ ,  $u$  a tensão aplicada no motor em  $[V]$ ,  $r_m$  o raio do cilindro de transmissão no eixo do motor elétrico em  $[m]$  e  $\theta$  é a inclinação da pista em  $[rad]$ .

Tabela 3.1: Constantes utilizadas no modelo do veículo

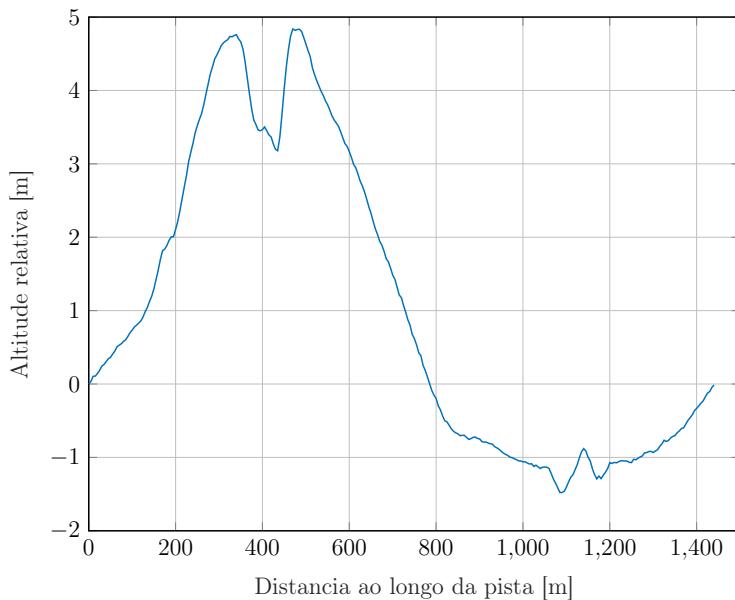
Constante	Símbolo	Unidade	Valor
Massa veículo	$m_v$	[kg]	36
Massa do piloto	$m_p$	[kg]	50
Resistência de armadura	$R_a$	[ $\Omega$ ]	0,07
Const. de tensão induzida	$K_v$	[ $V/(rad/s)$ ]	0,119
Quantidade de rodas	$N_r$	[ ]	3
Momento de inércia da roda	$J_r$	[ $kg.m^2$ ]	0,015
Momento de inércia do motor	$J_m$	[ $kg.m^2$ ]	$0,0625 \times 10^{-3}$
Raio da roda	$r_r$	[m]	0,254
Const. de torque	$K_t$	[ $Nm/A$ ]	0,119
Eficiência da roda de atrito	$\eta$	[ ]	0,95
Densidade do ar	$\rho$	[ $kg/m^3$ ]	1,22
Área frontal do veículo	$a_f$	[ $m^2$ ]	0,26
Coeficiente de arrasto aerodinâmico	$c_d$	[ ]	0,164
Aceleração da gravidade	$g$	[ $m/s^2$ ]	9,81
Coeficiente de resistência ao rolamento	$c_r$	[ ]	0,0024

Fonte: Elaborada pelo autor.

## 3.2 Modelo da pista

Para obter os dados de altitude da pista da Shell Eco-marathon Americas de 2019 no Kartódromo de Sonoma foi feito uma requisição a equipe *Duke Electric Vehicles* da *Duke University* que disponibilizou os dados coletados e tratados por ela. A Figura apresentada 3.1 o perfil de altitude da pista com estes dados.

Figura 3.1: Perfil de altitude da pista da Shell Eco-marathon Americas de 2019



Fonte: Dados da equipe de competição *Duke Electric Vehicles*.

Uma forma de representar esses dados no código para solução o OPC é por meio de uma *lookup table* porém ao implementar os dados dessa forma a solução não convergiu. Uma possível explicação para essa não convergência é que os dados eram derivados duas vezes, uma para determinar a inclinação e outra pelo realizado algorítimo de otimização e ao analisar esses dados derivados usando o comando `diff` do MATLAB® foram observadas curvas com grande variação de inclinação e ruido. Além disso seria preciso repetir sete vezes os dois vetores de dados para emular as repetidas voltas realizadas pelo carro durante a tentativa. A solução para esses problemas foi aproximar a curva de altitude da pista por uma soma de senos com o comando comando `fit(x, h, 'sin8')` do MATLAB® e então realizar a derivada para obter a seguinte equação inclinação:

$$\begin{aligned} \theta(x) = & \arctg(0.012293\cos(0.004363x - 0.228758) + 0.003041\cos(0.000236x + 0.399758) \\ & + 0.005818\cos(0.008726x - 2.193381) + 0.002926\cos(0.000246x + 3.490690) \\ & + 0.003958\cos(0.017449x - 3.058604) + 0.005365\cos(0.026184x + 0.579933) \\ & + 0.004361\cos(0.021814x + 1.627015) + 0.005539\cos(0.030552x - 1.400496)). \quad (3.2) \end{aligned}$$

### 3.3 Problema de controle ótimo (OPC)

$$\min_{i(t), r_m, T} \int_0^T i(t) \cdot [i(t) \cdot Ra + Kv \cdot (v(t)/r_m)] dt \quad (3.3a)$$

$$\text{s.t.} \quad x(0) = 0, \quad (3.3b)$$

$$v(0) = 0, \quad (3.3c)$$

$$\dot{x}(t) - v(t) = 0, \quad t \in [0, T], \quad (3.3d)$$

$$\dot{v}(t) - \frac{F_t - (F_a + F_g + F_r)}{M} = 0, \quad t \in [0, T], \quad (3.3e)$$

$$i(t) - \frac{42 - Kv \cdot (v/r_m)}{R_a} \leq 0, \quad t \in [0, T], \quad (3.3f)$$

$$i(t) \leq 30, \quad t \in [0, T], \quad (3.3g)$$

$$v(t) \leq 12.5, \quad t \in [0, T], \quad (3.3h)$$

$$x(T) = 10080, \quad (3.3i)$$

$$T \leq 1400, \quad (3.3j)$$

$$0.020 \leq r_m \leq 0.050 \quad (3.3k)$$

### 3.4 Código para solução do OPC

Nesta etapa do trabalho a versão do MATLAB® utilizada foi a R2019b Update 7 e a versão do FALCON.m utilizada foi a v1.26. O compilador MinGW-w64 C/C++, pré-requisito da biblioteca FALCON, foi instalado gratuitamente na versão 19.2.0 pela loja de aplicativos do MATLAB®. A licença do MATLAB® utilizada foi a licença acadêmica fornecida pela UFMG à seus alunos. Já a licença do FALCON.m, de uso individual e intransferível, foi obtida por meio de solicitação feita ao *Institute of Flight System Dynamics* da *Technische Universität München* pelo site <<https://www.fsd.lrg.tum.de/software/falcon-m/license-and-download/>>.

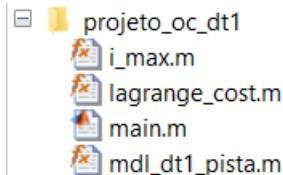
A implementação do código para solução do problema de controle ótimo proposto segui os seguintes passos:

1. **Definir as variáveis de estado e de controle e os parâmetros:** no arquivo principal, do tipo *script*, instanciar as variáveis de estado e de controle e os parâmetros otimizáveis;
2. **Implementar os modelos dinâmicos:** em arquivos do tipo *function*, implementar o modelo dinâmico do sistema;
3. **Definir os modelos dinâmicos dos subsistemas:** no arquivo principal, utilizar o *Model Builder* da *FALCON.m* para vincular os arquivos do passo 2 no programa principal;

4. **Derivar as funções necessária:** o *Model Builder* irá criar, a partir da chamado do método `.Build()`, as derivadas analíticas de todos os subsistemas e combiná-los ao gradiente geral dos modelos usando a regra da cadeia;
5. **Implementar funções de custo e de restrição de caminho:** repetir os passos de 1 à 4 para todas as restrições de caminho e funções de custo;
6. **Definir o problema de controle ótimo:** no arquivo principal, gerados no passo 4 com o problema e definir as condições finais e iniciais para os estados do sistema;
7. **Resolver o problema:** chamar o método `.Solve()`, no arquivo, principal para a resolução do problema.

O código de solução desenvolvido a partir destes passos é composto por quatro arquivos, apresentados na Figura 3.2, três *functions* e um *script*. Os arquivos *function*, `i_max.m`, `lagrange_cost.m` e `mdl_dt1_pista.m` são, respectivamente, a restrição de corrente máxima da Equação(??), o custo de Lagrange da Equação (??) e o modelo da dinâmica do veículo da Equação (??) acrescido do modelo de inclinação da pista.

Figura 3.2: Arquivos MATLAB® do projeto



Fonte: Elaborada pelo autor.

Durante a escrita do arquivo `main.m` uma dificuldade encontrada foi a determinação da variável `scale`, ultima variável nos construtores `falcon.State()`, `falcon.Control()` e `falcon.Parameter()`. Inicialmente os valores definidos foram a ordem de grandeza da cada variáveis o que gerou matrizes mau condicionadas no NLP e consequente um alto tempo de solução o problema. Os valores corretos, conforme sugerido no manual do FALCON.m<sup>[14]</sup>, dever ser definidos de tal forma que a ordem de grandeza da variável seja um. Por exemplo: uma varivel que representa a correte elétrica de um circuito em [mA] deve ter o parâmetro `scale` igual a  $10^{-3}$ . A definição dos estados, variável de controle e parâmetros com o parâmetro `scale` adequado é apresentado no Código 3.1.

Além disso, verificou-se que, apesar do indicado no manual da biblioteca FALCON.m<sup>[14]</sup>, ela, na versão utilizada com IPOPT não executa em um MATLAB com a versão recente que a utilizada pois a MathWorks não garante su

Código 3.1: Trecho do arquivo `main.m` com definição dos estados, variável de controle e parâmetros do OPC proposto

```
4 % vetor de estados
5 x_vec = [ falcon.State('x', 0, 11000, 1e-4);...
6         falcon.State('v', 0, 12.5, 1e-1) ];
7 % variavel de controle
8 u_vec = falcon.Control('i', 0, 63, 1);
9 % tempo final
10 tf = falcon.Parameter('FinalTime', 1400, 0, 1400, 1e-3);
11 % raio do cilindro no motor
12 rm = falcon.Parameter('raioM', 0.026, 0.020, 0.050, 1e2);
```

Fonte: Elaborado pelo autor.

# Capítulo 4

## Resultados

Para a execução do projeto, algumas etapas de desenvolvimento tiveram de ser seguidas: familiarização com o sistema, estudo dos módulos envolvidos, leitura dos requisitos, elaboração de documento descrevendo todo o processo de implementação e relacionamento com os diversos módulos, implementação e testes.



# Capítulo 5

## Conclusões

Novamente, este será um dos trechos que o leitor experiente lerá antes de decidir se vale a pena ler o texto integral. Seja convincente.

### 5.1 Considerações Finais

Reitere o que de mais importante foi feito, qual era o objetivo inicial e qual o resultado obtido. Se houve requisitos ou especificações de projeto, discuta se foram atingidos. Se os resultados não foram conclusivos ou contrariam o que se esperava, seja honesto e diga-o explicitamente. Busque explicar os insucessos com argumentos sólidos e plausíveis.

### 5.2 Propostas de Continuidade

Se houve questões ainda não respondidas ou resultados insatisfatórios, aponte direções de continuação.



# Referências Bibliográficas

- 1 AGUIRRE, L. A. **Introdução à Identificação de Sistemas:** Técnicas lineares e não-lineares aplicadas a sistemas reais. 3. ed. Belo Horizonte: Editora UFMG, 2007.
- 2 GUZZELLA, L.; SCIARRETTA, A. **Vehicle Propulsion Systems:** Introduction to modeling and optimization. Heidelberg: Springer, 2013.
- 3 POPP, K.; SCHIEHLEN, W. **Ground Vehicle Dynamics.** Heidelberg: Springer, 2010.
- 4 SANTIN, J. J. et al. **The World's Most Fuel Efficient Vehicle:** Design and development of Pac Car II. Zürich: vdf Hochschulverlag AG an der ETH Zürich, 2007.
- 5 EHSANI, M. et al. **Modern Electric, Hybrid Electric, and Fuel Cell Vehicles.** 3. ed. Boca Raton: CRC Press, 2018.
- 6 KIM, S. H. **Electric Motor Control:** DC, AC, and BLDC motors. 1. ed. [S.l.]: Elsevier Science, 2017.
- 7 GIERAS, J. F. **Permanent Magnet Motor Technology: Design and Applications.** 3. ed. Boca Raton: CRC Press, 2009.
- 8 NIEMANN, G. **Elementos de máquinas.** 1. ed. [S.l.]: BLUCHER, 1971. v. 3. ISBN 9788521214274.
- 9 NORTON, R. **Cinemática e dinâmica dos mecanismos.** [S.l.]: McGraw Hill Brasil, 2010.
- 10 KIRK, D. E. **Optimal Control Theory:** An introduction. [S.l.]: Dover Publications, 2004.
- 11 GROS, S.; DIEHL, M. **Numerical Optimal Control.** [s.n.], 2020. Em fase de elaboração. Disponível em: <<https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf>>.
- 12 DIEHL, M. et al. Fast direct multiple shooting algorithms for optimal robot control. In: **Lecture Notes in Control and Information Sciences.** Heidelberg: Springer, 2006. p. 65–93.
- 13 BETTS, J. **Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.** 2. ed. Philadelphia: Society for Industrial and Applied Mathematics, 2010. (Advances in Design and Control).
- 14 RIECK, M. et al. **FALCON.m User Guide.** München, 2020. Disponível em: <<https://www.fsd.lrg.tum.de/software/wp-content/uploads/UserGuideMain.pdf>>.

- 15 RIECK, R. M. **Discrete Controls and Constraints in Optimal Control Problems.** Tese (Doutorado) — Technische Universität München, München, 2017. Disponível em: <<https://mediatum.ub.tum.de/doc/1316413/1316413.pdf>>.
- 16 BITTNER, M. **Utilization of Problem and Dynamic Characteristics for Solving Large Scale Optimal Control Problems.** Tese (Doutorado) — Technische Universität München, München, 2017. Disponível em: <<https://mediatum.ub.tum.de/doc/1343164/1343164.pdf>>.
- 17 SALTZMAN, M. J. Coin-or: An open-source library for optimization. In: **Advances in Computational Economics.** [S.l.]: Springer, 2002. p. 3–32.
- 18 WÄCHTER, A.; BIEGLER, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical Programming**, Springer, v. 106, 03 2006.

# Apêndice A

## Códigos implementados

Este apêndice contem os códigos implementados pelo autor apresentados na Seção 3.4.

Código A.1: Arquivo i\_max.m

```
1 function [Iup] = i_max(v, i, rm)
2 Kv = 0.119;      % Constante de tensão induzida no motor [V/(rad/s)]
3 Ra = 0.07;       % Resistência elétrica do motor [V/A]
4 Vbat = 42;       % Tensão da bateria [V]
5
6 I_MAX = (Vbat-Kv*(v/rm))/Ra;
7
8 Iup = i - I_MAX;
9 end
```

Código A.2: Arquivo lagrange\_cost.m

```
1 function [Cost] = lagrange_cost(v, i, rm)
2 Kv = 0.119;      % Constante de tensão induzida no motor [V/(rad/s)]
3 Ra = 0.07;       % Resistência elétrica do motor [V/A]
4
5 u = i*Ra + Kv*(v/rm);
6
7 Cost = u*i;
8 end
```

Código A.3: Arquivo mdl\_dt1\_pista.m

```
1 function [mdl_out] = mdl_dt1_pista(x, v, i, rm)
2 % constantes
3 Kt = 0.119;      % Constante de torque do motor [N/A]
4 rr = 0.254;       % Raio da roda [m]
```

```

5 ro = 1.22;           % Densidade do ar [kg/m³]
6 Af = 0.26;           % Área frontal [m²]
7 Cd = 0.164;          % Coef. de arrasto aerodinâmico []
8 G = 9.81;             % Aceleração da gravidade [m/s²]
9 mi = 0.0024;          % Coef. de resistência ao rolamento dos pneus []
10 Jr = 0.015;           % Momento de inercia da roda[kg.m²]
11 Jm = 0.0625e-3;       % Momento do inercia do motor [kg.m²]
12 mv = 18;              % Massa do veiculo [kg]
13 mp = 50;              % Massa do piloto [kg]
14 ef = 0.95;            % Eficiênciam da roda de atrito (transmissão) []
15
16 % aproximação com 8 cos da inclinação da pista
17 theta = atan(0.012293*cos(0.004363*x-0.228758)+...
18     0.003041*cos(0.000236*x+0.399758)+...
19     0.005818*cos(0.008726*x-2.193381)+...
20     0.002926*cos(0.000246*x+3.490690)+...
21     0.003958*cos(0.017449*x-3.058604)+...
22     0.005365*cos(0.026184*x+0.579933)+...
23     0.004361*cos(0.021814*x+1.627015)+...
24     0.005539*cos(0.030552*x-1.400496));
25
26 eta = rr/rm;           % relação de transmissão
27 mr = 3*(Jr/rr^2) + Jm*(eta^2/rr^2); % massa eq. rotacional
28 M = mv + mp + mr;
29
30 Ft = ((Kt*i)*ef)/rm;      % força de tração
31 Fa = (ro*Af*Cd*v^2)/2;    % arrasto aerodinâmico
32 Fr = (mv + mp)*G*mi*cos(theta); % resistência ao rolamento
33 Fg = (mv + mp)*G*sin(theta); % componente do peso
34
35 % derivada dos estados
36 x_dot = v;
37 v_dot = (Ft - Fa - Fr - Fg)/M;
38
39 mdl_out = [x_dot; v_dot; Ft; Fa; Fr; Fg; theta];
40 end

```

Código A.4: Arquivo main.m

```

1 clc; clearvars; close all;
2
3 %% Definição do estados, variáveis de controle e de saída
4 % vetor de estados

```

```

5 x_vec = [ falcon.State('x', 0, 11000, 1e-4); ...
6         falcon.State('v', 0, 12.5, 1e-1) ];
7 % variavel de controle
8 u_vec = falcon.Control('i', 0, 63, 1);
9 % tempo final
10 tf = falcon.Parameter('FinalTime', 1400, 0, 1400, 1e-3);
11 % raio do cilindro no motor
12 rm = falcon.Parameter('raioM', 0.026, 0.020, 0.050, 1e2);
13 % variaveis de saida
14 modeloutput = [...
15     falcon.Output('Ft');...
16     falcon.Output('Fa');...
17     falcon.Output('Fr');...
18     falcon.Output('Fg');...
19     falcon.Output('theta')];
20
21 %% Construção do modelo, da restrição de caminho e o custo de Lagrange
22 % modelo do veiculo e pista
23 mdl = falcon.SimulationModelBuilder('DT1', x_vec, u_vec, rm);
24 mdl.addSubsystem(@mdl_dt1_pista, {'x', 'v', 'i', 'raioM'}, {'mdl_out'});
25 mdl.SplitVariable('mdl_out', {'x_dot','v_dot', 'Ft', 'Fa', 'Fr', 'Fg', ...
    'theta'});
26 mdl.setStateDerivativeNames('x_dot','v_dot');
27 mdl.setOutputs(modeloutput);
28 mdl.Build;
29 % restricao de caminho
30 lgc = falcon.PathConstraintBuilder('LCost', [], x_vec(2), u_vec(1), rm,
    @lagrange_cost);
31 lgc.Build;
32 % custo de lagrange
33 pti = falcon.PathConstraintBuilder('LimCor', [], x_vec(2), u_vec(1), rm,
    @i_max);
34 pti.Build;
35
36 %% Definição do problema de controle ótimo
37 % instancia do problema
38 problem = falcon.Problem('DT1_B1');
39 % discretização
40 tau = linspace(0,1,1001);
41 % fase 1 de 1
42 phase = problem.addNewPhase(@DT1, x_vec, tau, 0, tf);
43 phase.Model.setModelParameters(rm);

```

```
44 % malha do vetor de controle
45 phase.addNewControlGrid(u_vec, tau);
46 % variaveis de saida do modelo
47 phase.Model.setModelOutputs(modeloutput);
48 % condições inciais e finais dos estados
49 phase.setInitialBoundaries([0;0]);
50 phase.setFinalBoundaries([10080;0],[10080;12.5]);
51 % restrição de corrente
52 pathc = phase.addNewPathConstraint(@LimCor,falcon.Constraint('Iup', -inf
      , 0, 1),tau);
53 pathc.setParameters(rm);
54 % funcional de custo
55 lagc = phase.addNewLagrangeCost(@LCost,falcon.Cost('Cost',1e-4));
56 % parametro otimizavel
57 lagc.setParameters(rm);
58
59 %% Calculo da solução
60 problem.setMajorIterLimit(10000)
61 problem.Solve();
62
63 %% Plots
64 mostar_resultados(problem,rm)
```

## Apêndice B

### Dados da pista

É apresentado na Tabela ?? o dados coletados e processados da pista da Shell Eco-marathon Americas de 2019 no Kartódromo de Sonoma pela equipe *Duke Electric Vehicles* da *Duke University* concedidos a equipe Milhagem UFMG por meio de solicitação no site <<http://www.duke-ev.org/>>.