

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E MECÂNICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

**Rodrigo Perobeli Silva Costa**

**Desenvolvimento de Software de Otimização da Escolha de Motores Elétricos  
de Corrente Contínua para Robôs de Competição**

Juiz de Fora

2019

**Rodrigo Perobeli Silva Costa**

**Desenvolvimento de Software de Otimização da Escolha de Motores Elétricos  
de Corrente Contínua para Robôs de Competição**

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica.

Orientador: Alexandre da Silva Scari

Coorientador: Daniel de Almeida Fernandes

Juiz de Fora

2019

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Costa, Rodrigo Perobeli Silva.

Desenvolvimento de software de otimização da escolha de motores elétricos de corrente contínua para robôs de competição / Rodrigo Perobeli Silva Costa. -- 2019.

62 p. : il.

Orientador: Alexandre da Silva Scari

Coorientador: Daniel de Almeida Fernandes

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia, 2019.

1. Motor CC. 2. Minissumô. 3. Robótica. I. Scari, Alexandre da Silva, orient. II. Fernandes, Daniel de Almeida, coorient. III. Título.

**Rodrigo Perobeli Silva Costa**

**Desenvolvimento de Software de Otimização da Escolha de Motores Elétricos  
de Corrente Contínua para Robôs de Competição**

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica.

Aprovada em 05 de Novembro de 2019.

**BANCA EXAMINADORA**

---

Alexandre da Silva Scari - Orientador  
Universidade Federal de Juiz de Fora

---

Dr. Daniel de Almeida Fernandes - Coorientador  
Universidade Federal de Juiz de Fora

---

Dr. Moisés Luiz Lagares Júnior  
Universidade Federal de Juiz de Fora

## AGRADECIMENTOS

Agradeço primeiramente a meus pais, Walter Floriano da Costa e Sílvia Helena Perobeli da Silva Floriano Costa que tornaram possível minha graduação, bem como a moral fornecida pelo meu considerado irmão João Paulo Bridi e o suporte afetivo fornecido também por minha namorada Letícia Bartels Stephani.

Aos meus amigos, que estavam comigo nos problemas encontrados no decorrer do trabalho, em especial aos companheiros da equipe Rinobot da UFJF e Jimmy Botezine Lopes, que me forneceram um incrível apoio.

Aos meus professores que durante a minha graduação me forneceram a base do meu conhecimento de forma a tornar esse dia possível, em especial ao meu orientador Alexandre da Silva Scari, e coorientador Daniel de Almeida Fernandes por todo o suporte durante esse trabalho.

Também agradeço a Deus, pois sem ele nada é possível.

Por fim, agradeço a todos aqueles que de alguma forma contribuíram com minha formação em Engenharia Mecânica e, principalmente, com este projeto.

“Não se é reconhecido por se tornar um líder, torna-se líder por ser reconhecido.”

Masashi Kishimoto.

## RESUMO

Em uma equipe de competição de robótica, muitos elementos robóticos têm movimentos controlados via motores elétricos ou servomotores; logo, a boa escolha deles é de grande importância para o bom desempenho dos robôs. Atualmente, no ambiente de competições de robótica, os projetistas escolhem os motores via testes experimentais, via planilhas simples, ou simplesmente através da troca de informações com outras equipes participantes durante as competições. O presente trabalho vem disponibilizar um *software* em C++ que utiliza uma forma de inteligência computacional baseada em simulações para ajudar o projetista a definir o melhor motor para uma aplicação desejada.

Palavras-chave: Motor CC. Otimização. Robótica.

## ABSTRACT

On a competition team of robotics, many robotic elements in the robots have their movement controlled by electric motors or servomotors, consequently, making the right choice of these motors is of great importance to the proper functioning of the robots. Currently, in robotics competition environment, the designers choose the motors based on experimental tests, based on simple spreadsheet or even by simply experience exchange inside the competitions, therefore, this paper comes to provide a C++ *software* that uses a computational intelligence based on, giving assistance to the designer to take the best motor for the desired application.

Key-words: DC motor. Optimization. Robotics.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Robô sumoBOY de 2 rodas alinhadas. . . . .	17
Figura 2 – Exemplo de robô com 4 rodas alinhadas. . . . .	17
Figura 3 – Exemplo de robô de esteira. . . . .	18
Figura 4 – Ilustração de um motor elétrico CC. . . . .	21
Figura 5 – Subsistemas de um motor elétrico. . . . .	21
Figura 6 – Centro de gravidade. . . . .	24
Figura 7 – Viga biapoiada. . . . .	25
Figura 8 – Plano inclinado. . . . .	26
Figura 9 – Fluxograma do trabalho. . . . .	30
Figura 10 – <i>Dohyō</i> de minissumô. . . . .	32
Figura 11 – Aplicação <i>widget</i> desenvolvida para este projeto, em fase intermediária, no QtCreator. . . . .	33
Figura 12 – Esquemático das informações sobre o robô. . . . .	38
Figura 13 – Tela de informação de propriedades do <i>SolidWorks</i> . . . . .	40
Figura 14 – Robô Destiny da Rinobot. . . . .	41
Figura 15 – Balança usada no experimento. . . . .	41
Figura 16 – Foto do ensaio sendo executado. . . . .	42
Figura 17 – Roda StickyMAX S30 32 mm. . . . .	42
Figura 18 – Tela principal do <i>software</i> . . . . .	46
Figura 19 – Tela onde são apresentados os motores do banco de dados. . . . .	47
Figura 20 – Janela de adição de novos motores ao banco de dados. . . . .	48
Figura 21 – Tela de alteração de motor. . . . .	49
Figura 22 – Tela de inserção de dados externos. . . . .	49
Figura 23 – Tela de resultados do <i>software</i> . . . . .	51
Figura 24 – Gráfico polar com as notas individuais. . . . .	52
Figura 25 – Gráficos polares dos 2 melhores motores indicados pelo <i>software</i> . . . . .	53
Figura 26 – Detalhes dos motores emitidos pelo <i>software</i> para cada robô. . . . .	54
Figura 27 – Desempenho do motor <i>Fingertech GoldSpark 35:1 22,2 V</i> na simulação de luta. . . . .	55
Figura 28 – Motores <i>Fingertech GoldSpark</i> , de Fingertech Robotics, 2019. . . . .	61
Figura 29 – Motores <i>Fingertech SilverSpark</i> , de Fingertech Robotics, 2019. . . . .	62

## LISTA DE TABELAS

Tabela 1 – Matriz de triagem. . . . .	28
Tabela 2 – Matriz de pontuação. . . . .	29
Tabela 3 – Dimensões máximas de robôs de sumô. . . . .	31
Tabela 4 – Tabela de pesos para cada teste de acordo com o foco do motor. . . . .	37
Tabela 5 – Tabela com as informações dos robôs da Rinobot. . . . .	43
Tabela 6 – Tabela com os valores obtidos pelo ensaio de tração. . . . .	44
Tabela 7 – Tabela com os valores exibidos pelo modelo e os valores reais. . . . .	45
Tabela 8 – Notas obtidas pelo melhor motor indicado pelo <i>software</i> para os robôs da Rinobot. . . . .	52
Tabela 9 – Notas obtidas pelo segundo melhor motor. . . . .	53

## LISTA DE ABREVIATURAS E SIGLAS

CAD	<i>Computer Aided Design</i> (Desenho Assistido por Computador)
CC	Corrente Contínua
CG	Centro de Gravidade
ID	Identificador de cada motor no banco de dados
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
HPCB	<i>High Power Carbon Brushes</i> (Escovas de carbono de alta potência)
MDF	<i>Medium-Density Fiberboard</i> (Placas de fibras de madeira de média densidade)
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
UFJF	Universidade Federal de Juiz de Fora
VSSS	<i>Very Small Size Soccer</i> (Futebol mini)

## LISTA DE SÍMBOLOS

$\int$	Integração
$a$	Ponto de aplicação da expansão de Taylor
$B$	Constante de amortecimento viscoso
$\frac{d}{dt}$	Derivada temporal de primeira ordem
$\frac{d^2}{dt^2}$	Derivada temporal de segunda ordem
$dW$	Peso de uma partícula infinitesimal
$E1$	Distância entre o eixo traseiro do robô e a referência do <i>software</i>
$E2$	Distância entre o eixo dianteiro do robô e a referência do <i>software</i>
$e_{percent}$	Erro percentual entre a velocidade modelada e velocidade dada pela fabricante
$F_b$	Força de atrito exercida pelo atrito viscoso do motor
$F_{motor}$	Força exercida pelos motores do robô
$F_{res}$	Força resistente que simula um robô adversário
$F_{x,y}$	Força na direção $x$ ou na direção $y$
$G$	Localização do CG
$g_r$	Relação de transmissão de um dado motor
$h$	Passo de integração
$i$	Corrente elétrica
$J$	Momento de inércia polar do eixo do rotor
$K_t$	Constante de relação linear entre $T_m$ e $i$
$K_v$	Constante de relação linear entre $V_{cem}$ e $\omega$
$L$	Indutância das bobinas do motor
$L_{robo}$	Comprimento total do robô
$M_z$	Momentos em torno do eixo $z$
$m$	Massa do robô

$N$	Força normal na superfície do <i>dohyō</i>
$n$	Número de critérios
$O(h)$	Erro de truncamento de ordem $h$
$O(h^4)$	Erro de truncamento de ordem $h^4$
$P_e$	Potência elétrica
$P_m$	Potência mecânica
$R$	Resistência inerente ao motor
$R_{traseiro}$	Reação no apoio traseiro do robô
$R_{dianteiro}$	Reação no apoio dianteiro do robô
$r$	Raio da roda do robô
$r_{ij}$	Avaliação da $j$ -ésima alternativa para o $i$ -ésimo critério
$S_j$	Pontuação final para a $j$ -ésima alternativa
$T$	Torque resultante no eixo
$T_b$	Torque de atrito viscoso
$T_c$	Torque de carga gerado por forças externas
$T_d$	Torque causado pela força de atrito estático
$T_j$	Torque inercial
$T_m$	Torque magnético
$t_1$	Tempo inicial
$t_2$	Valor que representa $t_1 + h$
$t_n$	Valor de tempo na $n$ -ésima iteração
$t_{n+1}$	Valor de tempo na iteração consecutiva à $n$ -ésima iteração
$u(t)$	Função degrau no domínio do tempo
$V_{bat}$	Tensão da fonte de alimentação elétrica (bateria)
$V_{cem}$	Força contraeletromotriz gerada pela rotação do rotor
$V_L$	Queda de tensão devida à indutância do motor

$V_R$	Queda de tensão devida à resistência elétrica do motor
$v$	Velocidade linear do robô
$W$	Peso
$w_i$	Peso para o $i$ -ésimo critério
$X$	Posição do CG para ser informado no programa
$x$	Posição do robô em um espaço unidimensional
$\bar{x}$	Posição do CG no eixo $x$
$y(a)$	Valor da função $y(t)$ no ponto $a$
$\dot{y}(a)$	Derivada temporal da função $y(t)$ avaliada no ponto $a$
$y_1$	Valor da função $y(t)$ em $t_1$
$\dot{y}_1$	Derivada temporal da função $y(t)$ avaliada em $t_1$
$y_2$	Valor da função $y(t)$ em $t_1 + h/2$ baseando-se em $\dot{y}_1$
$\dot{y}_2$	Derivada temporal da função $y(t)$ avaliada em $t_1 + h/2$
$y_3$	Valor da função $y(t)$ em $t_1 + h/2$ baseando-se em $\dot{y}_2$
$\dot{y}_3$	Derivada temporal da função $y(t)$ avaliada em $t_1 + h/2$
$y_4$	Valor da função $y(t)$ em $t_1 + h$ baseando-se em $\dot{y}_3$
$\dot{y}_4$	Derivada temporal da função $y(t)$ avaliada em $t_1 + h$
$y_n$	Valor da função $y(t)$ na $n$ -ésima iteração
$y_{n+1}$	Valor da função $y(t)$ na iteração consecutiva à $n$ -ésima iteração
$\eta$	Eficiência do motor
$\mu$	Coefficiente de atrito estático
$\theta$	Ângulo de inclinação da superfície
$\omega$	Velocidade angular
$\omega_{modelo}$	Velocidade angular máxima gerada pelo modelo
$\omega_{fabricante}$	Velocidade angular máxima dada pela fabricante

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	Considerações iniciais	16
1.2	Tipos de robôs	16
1.3	Motivação	17
1.4	Objetivos	18
1.4.1	Objetivo geral	18
1.4.2	Objetivos específicos	19
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
2.1	Banco de dados	20
2.2	Modelo de um motor elétrico de corrente contínua	20
2.3	Resolução de sistemas de Equações Diferenciais Ordinárias	23
2.4	Centro de gravidade	24
2.5	Forças de apoio	25
2.6	Dinâmica em um plano inclinado	26
2.7	Matriz de decisão	26
<b>3</b>	<b>METODOLOGIA</b>	<b>30</b>
3.1	Regras de Sumô	31
3.1.1	Especificações dos robôs	31
3.1.2	O <i>dohyō</i> , o ringue de Sumô	31
3.1.3	A luta de Sumô	32
3.2	QtCreator	32
3.3	Atribuição de notas	33
3.3.1	Teste de aplicação em luta	34
3.3.2	Testes de arrancada e reversão	34
3.3.3	Eficiência	34
3.4	Modelagem matemática aplicada ao trabalho	35
3.5	Ponderação das notas	37
3.6	Torque máximo de deslizamento	37
3.6.1	Cálculo do CG do robô	39
3.7	Ensaio de força de tração	40
3.8	Ensaio do coeficiente de atrito	40
3.9	Banco de dados	42
3.10	Aplicação na Rinobot	43

<b>4</b>	<b>RESULTADOS E ANÁLISES</b>	<b>44</b>
4.1	Resultado do ensaio de tração	44
4.2	Resultado do ensaio do coeficiente de atrito	44
4.3	Validação do modelo	44
4.4	Funcionamento do <i>software</i>	45
4.4.1	Tela principal	45
4.4.2	Tela de motores	46
4.4.3	Tela de dados externos	48
4.4.4	Tela de resultados	50
4.5	Aplicação na Rinobot	51
<b>5</b>	<b>CONCLUSÕES</b>	<b>56</b>
5.1	Conclusões	56
5.2	Sugestões para trabalhos futuros	56
	<b>REFERÊNCIAS</b>	<b>57</b>
	<b>APÊNDICE A – CÓDIGO MATLAB® DE DESEMPENHO DE 1 MOTOR</b>	<b>58</b>
	<b>ANEXO A – MOTORES USADOS NO BANCO DE DADOS</b>	<b>61</b>
	<b>ANEXO B – TERMO DE AUTENTICIDADE</b>	<b>63</b>

# 1 INTRODUÇÃO

## 1.1 Considerações iniciais

Competições de robótica normalmente são compostas por várias categorias de aplicação, entre elas: sumô, futebol, combate, *followline* (seguidor de linha) e *trekking* (caminhada), principalmente. No Brasil, as maiores competições de robótica são organizadas pela Robocore (ROBOCORE, 2017) que é uma empresa de grande influência na área. A equipe de competição da Universidade Federal de Juiz de Fora (UFJF), intitulada Rinobot, que foi fundada em 2016, participa de diversas competições durante o ano e atualmente dispõe de categorias que abrangem o futebol, o sumô e o *followline*.

Como o presente trabalho tem o foco na categoria minissumô, é necessário discutir sobre o funcionamento dessa categoria. Antes da luta, os robôs competidores devem passar por uma avaliação para garantir que estejam dentro das especificações da categoria (Subseção 3.1.1), que para o caso do minissumô é  $100 \times 100$  mm de comprimento e largura e 500 g de massa máxima. A luta de sumô acontece em um *dohyō*, isto é, em um ringue, que é apresentado na Subseção 3.1.2. A luta é comandada por um juiz e o vencedor é aquele que conseguir garantir dois pontos de yukô, que podem ser obtidos através de descumprimento de regras ou através da vitória de um *round* (rodada).

## 1.2 Tipos de robôs

Após conhecer sobre como funciona uma partida de sumô e seus requisitos para vencê-la, é necessário discutir sobre como normalmente os robôs da categoria minissumô são fabricados. Um robô de minissumô está sujeito ao peso próprio. A estrutura desse tipo de modelo normalmente é construída sobre 2 ou 4 rodas alinhadas, havendo também modelos com 1 esteira e 2 rodas desalinhadas. O modelo mais comum é o de 2 rodas alinhadas (Figura 1). Esse tipo de modelo é mais simples, exige menos componentes eletrônicos e favorece a implementação de um controle mais robusto, que não exige o esforço de sincronizar 2 rodas para garantir que não haja escorregamento ou deformação da esteira. O peso distribui-se de forma que parte do apoio está sobre a lâmina frontal, que se arrasta no chão ao movimento do robô e a outra parte está sobre os motores alinhados.

O robô também pode ter 4 rodas alinhadas (Figura 2). A vantagem apresentada por esse modelo é a distribuição igual do peso em cima apenas das rodas, bem como a potência que é dobrada. No entanto, ao ser levantado pelo robô adversário, normalmente as rodas dianteiras perdem contato com o solo e portanto, se tornam inúteis.

Modelos mais raros de serem encontrados são os de esteira (Figura 3), que se valem de uma superfície de contato com o solo maior, perdendo em velocidade, e os de 2 rodas desalinhadas, que tornam o robô instável, mas possibilitam o uso de motores fisicamente



Figura 1 – Robô sumoBOY de 2 rodas alinhadas.  
Fonte: (ROBOT-NEST, 2019).

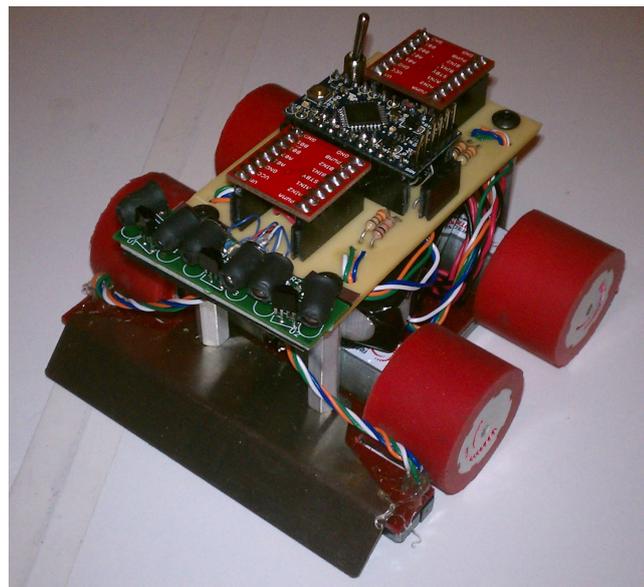


Figura 2 – Exemplo de robô com 4 rodas alinhadas.  
Fonte: (ROBOT-DIALOGS, 2019).

maiores e, conseqüentemente, mais potentes.

### 1.3 Motivação

A definição do motor para uma equipe de competição é de extrema importância, pois uma pequena diferença entre os motores usados nos robôs pode garantir a vitória, seja numa corrida ou numa luta de sumô.

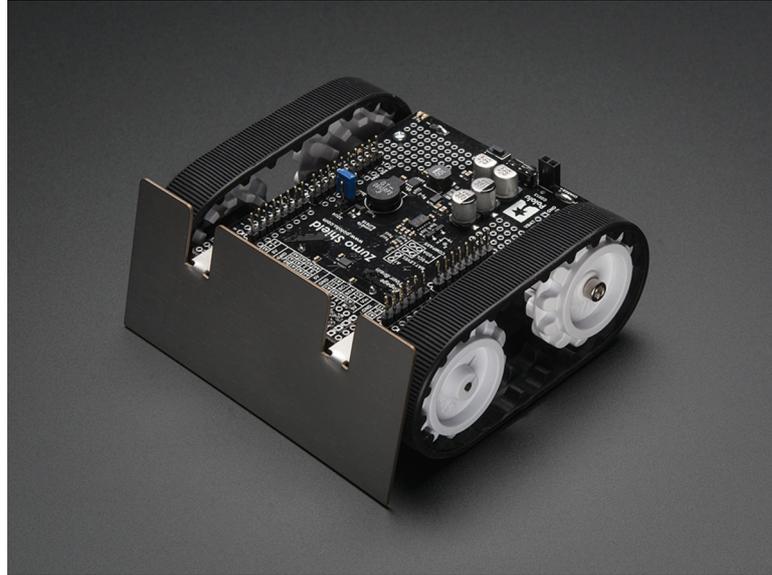


Figura 3 – Exemplo de robô de esteira.  
 Fonte: (ROBOT-DOMESTICI, 2019).

Atualmente na Rinobot, o cálculo e a especificação do motor são feitos diante da observação do que é usado nas competições, porém podem ser feitos calculando manualmente observando os gráficos fornecidos pelos *datasheets* (folha de dados) das fabricantes. Também pode ser feito o uso de planilhas para facilitar o cálculo manual, de forma que a comparação visual via planilha seja mais rápida e eficiente. Tal método é o mais utilizado por outras equipes. Há também um *website* desenvolvido pela equipe Tentacle chamado Team Tentacle Torque/Amp-hour calculator (TENTACLE, 2018) onde, através de algumas entradas, alguns parâmetros de saída são calculados. Tal *software* possui uma vasta quantidade de motores cadastrados em seu banco de dados; no entanto, essa quantidade de motores não é variável, não sendo permitido o cadastro de novos motores. O *software* também não informa qual o melhor motor, apenas calcula os parâmetros de saída. Para o cálculo de parâmetros, a referência (MEGGIOLARO et al., 2016) contém várias instruções e sugestões para a criação de um robô de combate, cuja teoria envolvida se estende às aplicações de sumô, tema deste trabalho.

Visando melhorar esse aspecto, o presente trabalho propõe um programa de computador que possibilite o usuário informar o modelo do robô, além de algumas restrições, para que ele indique automaticamente o melhor modelo de motor que pode ser usado.

## 1.4 Objetivos

### 1.4.1 Objetivo geral

Desenvolver um *software* que consiga determinar o melhor modelo de motor de Corrente Contínua (CC) para uma aplicação de equipe de competição de robótica que se

utiliza da categoria minissumô.

#### 1.4.2 Objetivos específicos

- O *software* deverá ser capaz de informar o melhor motor para o uso do minissumô através do uso de um banco de dados.
- Desenvolver avaliações para os motores de forma a poder atribuir notas e assim classificá-los, sendo baseadas no modelo matemático de motores elétricos CC;
- Para validação, um corpo de teste será usado para aquisição de dados relevantes para as avaliações prescritas.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Banco de dados

Como o objetivo do *software* é determinar o desempenho comparativo de vários motores, é necessário que haja uma quantidade deles para serem testados. É inviável que a cada simulação, seja preciso inserir todos os motores a serem analisados. Para tal, requer o uso de um banco de dados, que armazene as informações de cada motor.

O banco de dados é uma forma na qual é possível armazenar informações, de forma que permita inserir, alterar, excluir e consultar dados requisitados pelo usuário. Os dados ficam armazenados no servidor do banco, que possuem diversas plataformas, sendo a mais usadas as que trabalham com linguagem SQL (Linguagem de Consulta Estruturada), uma linguagem própria para esse tipo de aplicação. O *software* auxiliar escolhido, que gera o banco de dados, foi o SQLite. Esse *software* propicia a criação fácil de um banco de dados bem como uma comunicação adequada ao QtCreator, tornando a programação mais direta.

### 2.2 Modelo de um motor elétrico de corrente contínua

O motor elétrico é o elemento principal deste projeto e deve ser modelado de forma consistente a aproximar a simulação da realidade. O princípio básico do motor é a conversão eletromagnética, ocasionada pela Lei de Indução de Faraday. A conversão eletromagnética ocorre quando a mudança no fluxo é associada com um movimento mecânico. Em máquinas rotativas, as tensões são geradas nas bobinas por uma das seguintes formas ([FLITZGERALD A. E. KINGSLEY, 2002](#)):

- Girando essas bobinas mecanicamente através de um campo magnético;
- Girando mecanicamente um campo magnético que atravessa as bobinas;
- Projetando um circuito magnético de forma que a relutância magnética varie com a rotação do motor.

O efeito de variação do fluxo magnético ocasiona o aparecimento de uma tensão induzida (lei de Faraday), é o princípio básico de um gerador, alternadamente, ao inserir uma tensão na bobinas do rotor e estator, um torque magnético é gerado no rotor, tornando essa máquina elétrica um motor. O motor elétrico de corrente contínua (Figura 4) é composto basicamente de um estator, onde ficam as bobinas de campo, que geram um campo magnético dentro do motor, e um conjunto rotor, comutador e escovas, por onde passa a corrente contínua que gera a força magnética no eixo acoplado ao rotor.

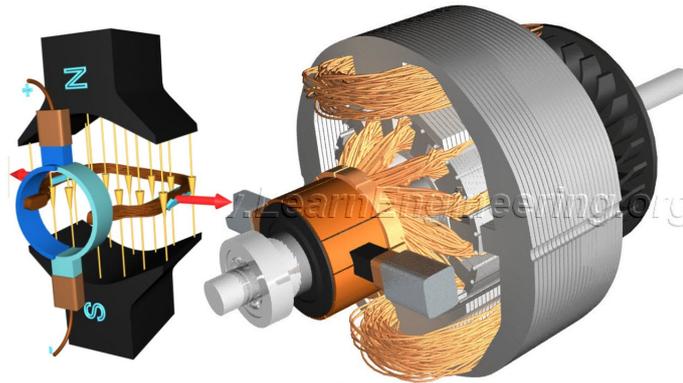


Figura 4 – Ilustração de um motor elétrico CC.  
Fonte: Electric Page, 2019.

O sistema motor elétrico pode ser subdividido em dois subsistemas: o elétrico e o mecânico. Esses subsistemas são necessários para definir as equações diferenciais do motor de forma a obter as equações de corrente e velocidade angular deste.

O subsistema elétrico trata do circuito elétrico representado pelo motor, e pode ser resumido em uma fonte de tensão  $V_{bat}$ , uma resistência inerente do motor  $R$ , uma indutância das bobinas  $L$  e a uma tensão contra eletromotriz  $V_{cem}$  gerada pela rotação do campo magnético do rotor ( $\omega$ ), que se relaciona com aquele através de uma constante  $K_v$ . Esse circuito é mostrado na Figura 5.

Já o subsistema mecânico é composto pelo eixo e os torques que agem nele. Há o torque magnético  $T_m$  gerado pela corrente elétrica no rotor ( $i$ ) multiplicado pela constante  $K_t$ , o torque inercial  $T_j$  inerente ao eixo, dependente de um momento de inércia  $J$ , o torque de atrito  $T_b$  modelado como um atrito viscoso do lubrificante usado (com uma constante de amortecimento  $B$ ), e o torque de carga  $T_c$  que é gerado por forças externas.

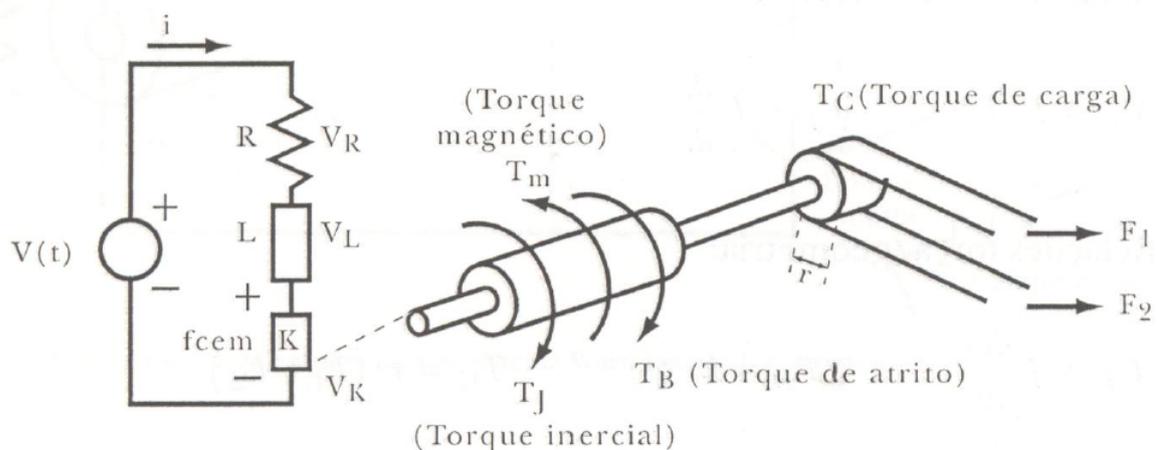


Figura 5 – Subsistemas de um motor elétrico.  
Fonte: (GARCIA, 2016).

As equações que regem esse tipo de sistema são mostradas por (GARCIA, 2016),

e mostram a relação linear entre o torque gerado e a corrente no rotor, bem como a linearidade da tensão contra-eletromotriz em função da rotação do rotor.

$$T_m = K_t \cdot i \quad (2.1)$$

$$V_{cem} = K_v \cdot \omega \quad (2.2)$$

Dos subsistemas apresentados, outras relações são usadas, como a Lei de Ohm (2.3), segunda lei de Newton para corpos rotativos (2.4), a lei de indução de Faraday (2.5) e o modelo de atrito viscoso para corpos em rotação (2.6).

$$V_{bat} = R \cdot i \quad (2.3)$$

$$T_j = J \cdot \frac{d\omega}{dt} \quad (2.4)$$

$$V_L = L \cdot \frac{di}{dt} \quad (2.5)$$

$$T_b = B \cdot \omega \quad (2.6)$$

Usando as leis de Kirchhoff de malhas é possível determinar a equação (2.8), e através da 2ª lei de Newton para somatório de momentos obtém-se a equação (2.10)

$$V_{bat} = V_R + V_L + V_{cem} \quad (2.7)$$

$$V_{bat} = R \cdot i + L \cdot \frac{di}{dt} + K_v \cdot \omega \quad (2.8)$$

$$T_m = T_c + T_b + T_j \quad (2.9)$$

$$K_t \cdot i = T_c + B \cdot \omega + J \cdot \frac{d\omega}{dt} \quad (2.10)$$

Ajustando as equações (2.8) e (2.10) é possível obter um sistema de equações diferenciais ordinárias de primeira ordem com 2 equações. No espaço de estados, o sistema é dado pela equação (2.11), onde  $u(t)$  é a função degrau unitário.

$$\frac{d}{dt} \begin{bmatrix} i \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{K_v}{L} & \frac{R}{L} \\ -\frac{B}{J} & \frac{K_t}{J} \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{V_{bat}}{L} \\ -\frac{T_c}{J} \end{bmatrix} \cdot u(t) \quad (2.11)$$

### 2.3 Resolução de sistemas de Equações Diferenciais Ordinárias

O sistema (2.11) pode ser resolvido numericamente de forma a ter a corrente e velocidade angular em função do tempo. Para resolver numericamente uma equação diferencial, há métodos de passo simples e de passo múltiplo. Ambos os métodos estimam a solução como sendo uma série de valores a intervalos específicos, que geram uma função que satisfaz a equação diferencial (GARCIA, 2016). No presente trabalho, serão abordados os métodos de passo simples, sendo eles o método de Euler e o Método Runge-Kutta de 4ª ordem, que são apresentados também em (GARCIA, 2016).

O método de Euler trabalha com a derivada de primeira ordem, já conhecida, e usa um passo de integração  $h$ , onde termos de ordem superior a 2 são truncados, seguindo uma expansão de Taylor (SANTOS, 2017):

$$y(a + h) = y(a) + h \cdot \dot{y}(a) \quad (2.12)$$

Equivalentemente:

$$u_1 = h \cdot \dot{y}_n(a) \quad (2.13)$$

$$y_{n+1} = y_n + u_1 \quad (2.14)$$

Para determinar o ponto  $y_{n+1}$  no instante  $t_{n+1} = t_n + h$ , o método de Euler necessita do ponto anterior ( $y_n$ ), interpolando a distância real entre esses dois pontos como uma reta, assumindo um erro. Tal erro, chamado erro de truncamento ( $O(h)$ ) é proporcional ao tamanho  $h$  do passo, portanto, para se atingir erros menores, é necessário um menor valor de passo, o que implica em um maior número de iterações do método para um dado intervalo.

Posteriormente ao método de Euler, surgiu o método de Runge-Kutta. Tendo o método de Euler como base, o Runge-Kutta consegue aproximar as soluções de uma equação diferencial de forma mais eficiente sem, no entanto, derivar a equação original, procedimento necessário caso fosse baseado no polinômio de Taylor. Segundo (GARCIA, 2016), o método Runge-Kutta de 4ª ordem é o método mais usado e consiste em avaliar as derivadas no início, no meio e no fim do intervalo de integração de tamanho  $h$ , realizando uma soma ponderada dessas derivadas ao final.

Os passos para realização do método Runge-Kutta são os seguintes (GARCIA, 2016):

- A derivada  $\dot{y}_1$  é avaliada em  $t_1$  e, aplicando-se o método de Euler, o valor da função é calculado em  $(t_1 + h/2)$ , resultando em  $y_2$ ;
- A derivada é avaliada em  $(t_1 + h/2)$  resultando em  $\dot{y}_2$ ;

- Iniciando-se em  $(y_1, t_1)$ , a função em  $(t_1 + h/2)$  é recalculada, usando-se a derivada  $\dot{y}_2$  para gerar  $y_3$ ;
- A derivada  $\dot{y}_3$  é avaliada em  $(t_1 + h/2)$ ;
- Iniciando-se em  $(y_1, t_1)$  a função é calculada em  $t_2 = t_1 + h$  usando-se a derivada  $\dot{y}_3$  para gerar  $y_4$ ;
- Usam-se as derivadas  $\dot{y}_1$  a  $\dot{y}_4$ , o valor da função  $y(t_2)$  é calculado por:

$$y(t_2) = y(t_1) + \frac{h}{6} \cdot (\dot{y}_1 + 2 \cdot \dot{y}_2 + 2 \cdot \dot{y}_3 + \dot{y}_4) \quad (2.15)$$

O erro de truncamento do método de Runge-Kutta de 4ª ordem é  $O(h^4)$ , que é proporcional ao passo  $h$  elevado à quarta potência. Uma vez que  $h < 1$ , o método Runge-Kutta garante um erro muito menor que o método de Euler para um mesmo  $h$ .

## 2.4 Centro de gravidade

Segundo (HIBBELER, 2011), um corpo é composto de uma série infinita de partículas de tamanho diferenciado, e assim, se o corpo estiver localizado em um campo gravitacional, então cada partícula terá um peso  $dW$ . Esses pesos formarão um sistema de forças aproximadamente paralelas, e a resultante desse sistema é o peso total do corpo, que passa por um único ponto chamado centro de gravidade (CG),  $G$ , conforme a Figura 6.

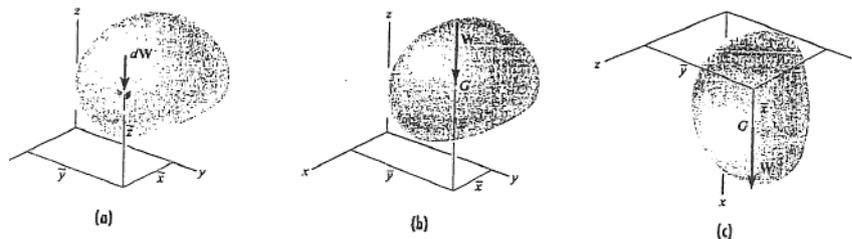


Figura 6 – Centro de gravidade.  
Fonte: (HIBBELER, 2011).

O peso total do corpo pode ser medido somando-se os pesos das partículas infinitesimais que o compõem, conforme (2.16):

$$W = \int dW \quad (2.16)$$

Como, por definição, o centro de gravidade é um ponto por onde é colocada a força peso do corpo, em um corpo em equilíbrio, o somatório dos momentos gerados pelas infinitas partículas que compõem o corpo em relação a um referencial, deve ser igual ao

momento gerado pela força peso aplicada no centro de gravidade em relação ao mesmo referencial, de forma que:

$$\bar{x} \cdot W = \int x \cdot dW \quad (2.17)$$

onde  $\bar{x}$  representa a posição do centro de gravidade em relação a um dado referencial.

## 2.5 Forças de apoio

Determinar reações de apoio em estruturas estáticas é uma tarefa importante principalmente para o dimensionamento de estruturas. Para o presente trabalho, se faz necessário tal cálculo para determinar o torque máximo de deslizamento (Seção 3.6). O cálculo de reações de apoio pode ser facilmente obtido em estruturas planas em equilíbrio.

A condição para um corpo plano estar em equilíbrio é que o somatório das forças nos eixos do plano sejam iguais a zero, conforme indica a equação (2.18), e que o somatório dos momentos em torno do eixo que sai do plano também o seja, conforme indica a equação (2.19), tendo então 3 equações possíveis de serem trabalhadas. Uma viga submetida ao peso próprio pode ser modelada como mostra a Figura 7. Nela há um apoio de primeiro gênero à direita, que executa forças apenas na vertical, e na esquerda apresenta um apoio de segundo gênero, que executa forças de reação na horizontal e vertical. Com isso, a estrutura de uma viga pode ser modelada como uma estrutura isostática.

$$\sum F_{x,y} = 0 \quad (2.18)$$

$$\sum M_z = 0 \quad (2.19)$$

Em particular, uma viga sujeita ao peso próprio não apresenta reações na direção horizontal no apoio de segundo gênero, a estrutura fica instável e suscetível ao movimento nessa direção, mas é possível ainda determinar as reações na direção vertical em ambos os apoios. Nota-se que esse é um modelo que pode representar um robô móvel sob rodas simétrico, e ele que será utilizado nos cálculos da Seção 3.6.

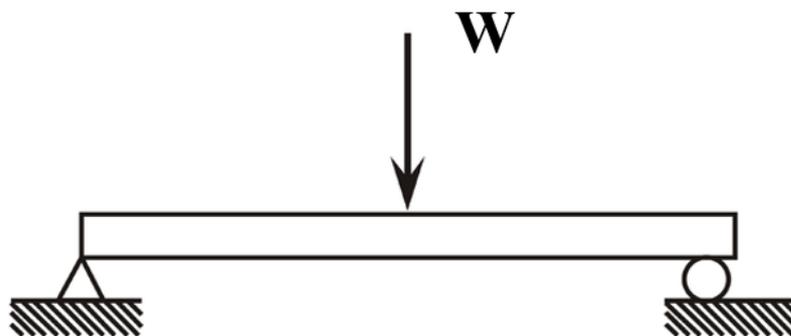


Figura 7 – Viga biapoiada.  
Fonte: Adaptada de Research Gate, 2019.

## 2.6 Dinâmica em um plano inclinado

A dinâmica de um corpo quando em um plano inclinado é apresentada por (NUSSENZVEIG, 2002). Em um plano inclinado, para o corpo permanecer em equilíbrio, é necessário que a força de atrito do corpo com o solo seja igual a componente do peso que age sobre o corpo na direção do plano. A Figura 8 representa tal situação:

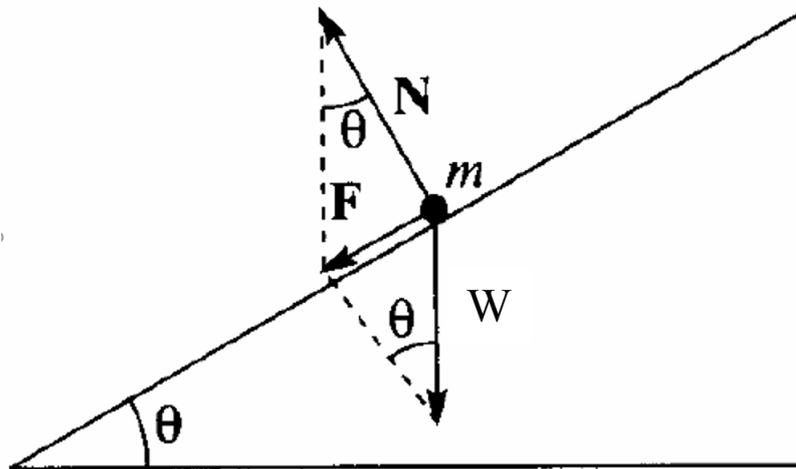


Figura 8 – Plano inclinado.  
Fonte: (NUSSENZVEIG, 2002).

A equação que rege essa situação é apresentada por (2.20):

$$W \cdot \text{sen } \theta = W \cdot \text{cos } \theta \cdot \mu \quad (2.20)$$

onde:

$W$  = Peso da partícula de massa  $m$ ;

$\theta$  = Ângulo de inclinação da superfície;

$\mu$  = Coeficiente de atrito estático da superfície com o corpo.

Rearranjando (2.20) é possível determinar qual o coeficiente de atrito estático entre o corpo e a superfície:

$$\mu = \tan \theta \quad (2.21)$$

## 2.7 Matriz de decisão

A matriz de decisão é um método de auxílio em projetos, discutido por (ULRICH KARL T. EPPINGER, 2011). Aplica-se à seleção de conceitos em projetos dentre os quais, várias opções estão a disposição do projetista, buscando o projeto que apresenta maior chance de sucesso. Visando a agilidade do processo de decisão, evitando que o projetista tenha que escolher entre muitas alternativas, o método é dividido em duas etapas: A triagem e a pontuação. A triagem é o estágio em que é feita uma rápida

avaliação aproximada que visa eliminar as alternativas inviáveis. A pontuação é uma etapa mais cuidadosa em que é feita uma análise nas alternativas reduzidas pela triagem.

Ambos os estágios seguem uma lista de passos que devem ser efetuados que levam à decisão final do projeto:

- Preparar as matrizes de seleção;
- Avaliar as alternativas;
- Classificar as alternativas;
- Combinar e aprimorar as alternativas;
- Selecionar uma ou mais alternativas;
- Refletir sobre os resultados do processo.

Apesar de (ULRICH KARL T. EPPINGER, 2011) apresentar a definição do processo, é deixado claro que os projetistas ou a equipe de projetistas são quem devidamente escolhem os métodos de avaliação e efetuam as decisões de forma subjetiva de acordo com o ponto de vista de cada colaborador.

O processo de criação da matriz de triagem visa a diminuição de alternativas de projeto com base nas premissas e nas necessidades do projeto sem, no entanto, aplicar pesos aos critérios. É necessário então comparar as possibilidades com um certo padrão. Normalmente o padrão utilizado é um produto atual e as alternativas são a melhoria do produto. É definida então uma métrica de comparação em que “+” representa “maior que”, 0 denota “igual ou semelhante a” e “-” atribui “pior que”. O somatório dos critérios é efetuado e os resultados são ordenados em um *ranking* (classificação). A Tabela 1 mostra o exemplo de uma matriz de triagem para a produção de uma seringa.

O processo de pontuação segue o princípio da triagem, no entanto, após o processo de triagem, um número menor de alternativas estarão disponíveis, como complementações ou combinações das ideias anteriores. Como posteriormente há uma ponderação associada, se faz necessário criar uma associação numérica de valores (recomendado por (ULRICH KARL T. EPPINGER, 2011)) de 1 a 5 em que cada valor equivale a uma avaliação relativa:

1. = Muito pior que
2. = Pior que
3. = Igual ou semelhante a
4. = Melhor que

5. = Muito melhor que

Dada a métrica de avaliação, organizam-se as informações na forma de uma matriz de pontuação, apresentada pela Tabela 2. Cada alternativa é avaliada em cada critério, com cada critério tendo o seu peso. O resultado das alternativas é obtido usando a equação (2.22):

$$S_j = \sum_{i=1}^n r_{ij} \cdot w_i \quad (2.22)$$

onde:

$r_{ij}$  = Avaliação da  $j$ -ésima alternativa para o  $i$ -ésimo critério;

$w_i$  = Peso para o  $i$ -ésimo critério;

$n$  = Número de critérios;

$S_j$  = Pontuação final para a  $j$ -ésima alternativa.

Os resultados são novamente analisados, incrementados e combinados, feita novamente a classificação via matriz de pontuação até resultar em uma única alternativa restante, que é o melhor projeto, que será encaminhado para execução.

ALTERNATIVAS							
Critério	A	B	C	D	E	F	G
Fácil manipulação	0	0	-	0	0	-	-
Fácil utilização	0	-	-	0	0	+	0
Legibilidade	0	0	+	0	+	0	+
Precisão de dosagem	0	0	0	0	-	0	0
Durabilidade	0	0	0	0	0	+	0
Fácil manufatura	+	-	-	0	0	-	0
Portabilidade	+	+	0	0	+	0	0
Total de +	2	1	1	0	2	2	1
Total de 0	5	4	3	7	4	3	5
Total de -	0	2	3	0	1	2	1
Resultado líquido	2	-1	-2	0	1	0	0
Rank	1	6	7	3	2	3	3
Continuar?	Sim	Não	Não	Combinar	Sim	Combinar	Revisar

Tabela 1 – Matriz de triagem.

Fonte: Adaptada de (ULRICH KARL T. EPPINGER, 2011).

ALTERNATIVAS									
		A		DF		E		G+	
Critério	Peso	Nota	Final	Nota	Final	Nota	Final	Nota	Final
Fácil manipulação	5%	3	0,15	3	0,15	4	0,2	4	0,2
Fácil utilização	15%	3	0,45	4	0,6	4	0,6	3	0,45
Legibilidade	10%	2	0,2	3	0,3	5	0,5	5	0,5
Precisão de dosagem	25%	3	0,75	3	0,75	2	0,5	3	0,75
Durabilidade	15%	2	0,3	5	0,75	4	0,6	3	0,45
Fácil manufatura	20%	3	0,6	3	0,6	2	0,4	2	0,4
Portabilidade	10%	3	0,3	3	0,3	3	0,3	3	0,3
	Total	2,75		3,45		3,10		3,05	
	Rank	4		1		2		3	
	Continua?	Não		Desenvolver		Não		Não	

Tabela 2 – Matriz de pontuação.

Fonte: Adaptada de (ULRICH KARL T. EPPINGER, 2011).

### 3 METODOLOGIA

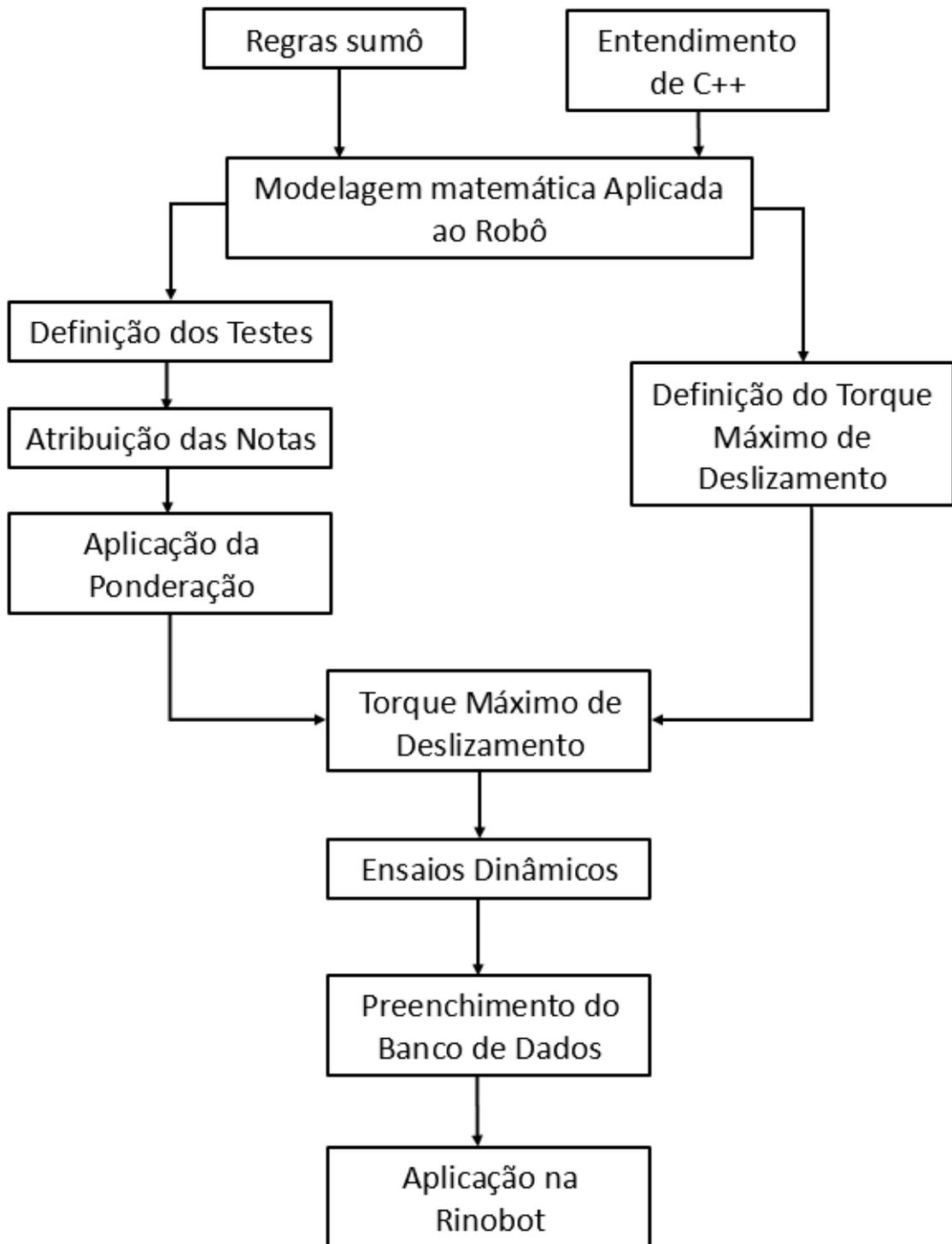


Figura 9 – Fluxograma do trabalho.

Fonte: Autor, 2019.

### 3.1 Regras de Sumô

Para determinar motores para uma dada aplicação, é necessário entendê-la de forma a desenvolver uma métrica de avaliação que atenda aos requisitos da aplicação. Para isso, é necessário entender as regras de uma competição de minissumô.

No Brasil, as regras de minissumô são regidas pela organização Robocore ([ROBOCORE, 2017](#)), que além da venda de produtos relacionados a robôs de combate, *followline* e sumô, também é a maior organizadora brasileira de eventos de robótica, que torna possível a realização de eventos como a Winter Challenge, a IronCup, o ArduinoDay (na cidade de São Paulo), a RoboGames, entre outros. Como organizadora, a Robocore proporciona as regras que são usadas nas mesmas. Em ([ROBOCORE, 2017](#)) são apresentadas todas as regras relacionadas à categoria de sumô, das quais, para o presente trabalho, apenas algumas partes são interessantes e serão abordadas abaixo.

#### 3.1.1 Especificações dos robôs

Os robôs de sumô devem ser capazes de caber em um quadrado ou cubo com dimensões referentes à sua classe bem como a massa do robô no início da luta. Para o caso da equipe Rinobot, que participa das categorias mini e LEGO<sup>®</sup>, apenas a categoria mini é relevante para o presente estudo, uma vez que somente motores LEGO<sup>®</sup> podem ser usados na categoria sumô LEGO<sup>®</sup>. Portanto, os valores de dimensão máxima a serem estudados são  $100 \times 100$  mm e a massa máxima é de 500 g (os demais valores são apresentados na Tabela 3). Outro ponto importante é que dispositivos como ímãs e bombas de vácuo, que visam o aumento da força normal, só são permitidos na categoria de sumô 3 kg, logo, não é o foco do presente trabalho.

Classe	Altura	Largura	Comprimento	Massa
3kg sumô	Ilimitada	20 cm	20 cm	3000 g
Minissumô	Ilimitada	10 cm	10 cm	500 g
Microsumô	5 cm	5 cm	5 cm	100 g
Nanosumô	2,5 cm	2,5 cm	2,5 cm	25 g
Sumô LEGO <sup>®</sup>	Ilimitada	15,2 cm	15,2 cm	1000 g
Humanóide (RoboOne)	50 cm	20 cm	20 cm	3000 g

Tabela 3 – Dimensões máximas de robôs de sumô.

Fonte: Adaptada de ([ROBOCORE, 2017](#)).

#### 3.1.2 O *dohyō*, o ringue de Sumô

O *dohyō* tem formato circular e tem material e tamanho apropriados para cada classe de sumô (Figura 10). Para o caso do minissumô, o *dohyō* apresenta espessura de 2.5 cm e diâmetro de 77 cm sendo composto de uma placa de MDF. Tudo que não se encontra

na área interna do *dohyō* é considerado área externa. Outras especificações do *dohyō* são apresentadas em (ROBOCORE, 2017) porém não são pertinentes para o trabalho.



Figura 10 – *Dohyō* de minissumô.  
Fonte: Adaptada de (ROBOCORE, 2017).

### 3.1.3 A luta de Sumô

A luta de minissumô é cheia de detalhes, conforme em (ROBOCORE, 2017), porém pode ser resumida para facilitar o estudo. Cada equipe compete com 1 robô construído de acordo com as especificações da Subseção 3.1.1. A partida é iniciada ao comando do juiz principal e continua até um dos competidores conquistar 2 pontos de yukô. O ponto de yukô pode ser obtido através do descumprimento de alguma regra prevista, ou como ocorre na maioria dos casos, quando um dos competidores consegue levar seu adversário para fora dos limites do *dohyō*, fazendo com que uma parte do adversário toque a área externa, vencendo o *round*.

## 3.2 QtCreator

Como a aplicação do estudo visa a interação e análise de dados obtidos através do usuário, faz-se necessária a criação de um *software* com interface para a boa utilização do mesmo. Dessa forma, foi escolhida a linguagem de programação C++, devido sua alta aplicação no mercado atual, bem como por ser utilizada no ambiente de graduação da UFJF, tornando o aprendizado mais eficiente.

O ambiente de desenvolvimento integrado (IDE) usado é o QtCreator, uma IDE de código aberto que aceita as linguagens C e C++, com uma extensão para Python. Nela é possível desenvolver aplicações de console, em que as informações são apresentadas no terminal do sistema operacional, bem como aplicações *widget* (ferramenta), que utilizam classes de interação com usuário para criar uma interface visualmente agradável e melhor para a interação com o usuário (Figura 11).

Para a aplicação *widget* são utilizados conceitos de programação orientada a objeto. Cada janela do programa é definida como uma classe, que pode possuir funções membro, conhecidas também como métodos, e propriedades, que são referenciadas à classe criada.

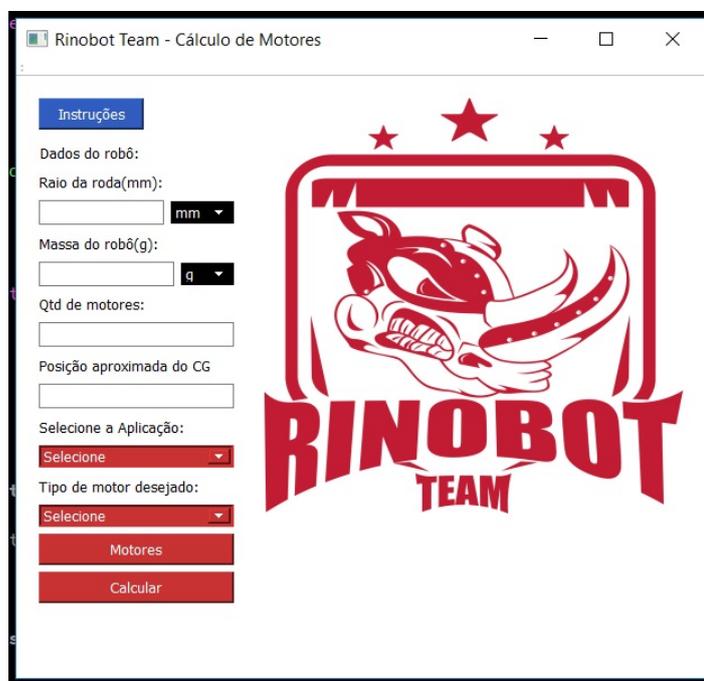


Figura 11 – Aplicação *widget* desenvolvida para este projeto, em fase intermediária, no QtCreator.

Fonte: Autor, 2018.

No caso do QtCreator, cada elemento adicionado na tela é um objeto, instanciado por uma classe, contendo diferentes propriedades, desde cor e tamanho até disponibilidade para uso do usuário. Apresentando uma janela de *Design*, o QtCreator torna fácil a intuitiva a criação de janelas de interação com o usuário, proporcionando um ambiente de rápido aprendizado e com diversas opções disponíveis ao desenvolvedor.

### 3.3 Atribuição de notas

Com o objetivo de definir qual o melhor motor, há a necessidade de criar uma métrica para classificar os motores. Cada métrica é variável com a aplicação, para o presente trabalho, a aplicação escolhida foi para um robô de minissumô. Dito isso, foram criados 6 parâmetros para definir essa métrica:

- Teste de arrancada;
- Teste de reversão;
- Teste de aplicação em luta;
- Eficiência;

- Preço;
- Velocidade máxima de rotação.

Dos quais os parâmetros de preço é informado pelo usuário e rotação máxima é calculada pelo programa. Cada um desses testes apresentará um resultado que será ponderado para obter uma classificação final dos motores, que será apresentado ao usuário.

### 3.3.1 Teste de aplicação em luta

Para a categoria minissumô, este é o teste mais importante para a definição dos motores, pois é nele que é definido se o motor a ser testado consegue ou não empurrar uma força opositora a seu movimento. O teste consiste em modelar matematicamente uma luta, onde o robô com os motores inseridos pelo usuário (Seção 4.4) é posto contra uma força oposta obtida experimentalmente (Seção 3.7) bem no centro do *dohyō*, com velocidade inicial nula. Caso o motor testado consiga empurrar a força opositora pela distância de metade do diâmetro do *dohyō*, é então adquirido o tempo que o motor levou para realizar tal ação. Ordenando de forma crescente, o melhor motor é aquele que realiza a ação em menor tempo. Caso não consiga, o motor é automaticamente descartado recebendo uma nota mínima.

### 3.3.2 Testes de arrancada e reversão

O teste de arrancada consiste em determinar o tempo que o motor usado leva para atingir a velocidade máxima, bem como se consegue atingi-la dentro dos limites do *dohyō*. Tal teste consiste em iniciar com o robô no centro do *dohyō* e ligar o motor. Caso o motor atinja a rotação máxima dada pela fabricante antes de ultrapassar o limite do campo (38.5 cm), o tempo necessário para tal ação é gravado e, novamente, os valores são ordenados e o que apresenta menor valor é considerado o melhor.

De forma semelhante, o teste de reversão consiste em adquirir o tempo necessário pelo motor ir da velocidade máxima negativa para a velocidade máxima positiva, o robô segue os padrões do teste de arrancada, no entanto, com velocidade inicial igual a velocidade máxima negativa, ou seja, com o robô em ré. O tempo total para realizar a reversão é obtido e ordenado, de forma que o menor tempo é considerado o melhor.

### 3.3.3 Eficiência

A eficiência do motor pode ser determinada ao relacionar os parâmetros de velocidade angular, torque resultante e corrente de trabalho obtidos pela simulação durante a aplicação em luta, usando as equações (3.1) e (3.2) onde  $P_m$  é a potência mecânica resultante, com  $T$  sendo o torque resultante no eixo, e  $P_e$  é a potência elétrica consumida pelo motor. O torque resultante, quando o motor não tem capacidade para empurrar a

$F_{res}$  se torna negativo, com isso para que não haja eficiências negativas, atribui a essa eficiência o valor 0. De forma semelhante, quando a corrente elétrica calculada pelo modelo matemático ultrapassa a corrente máxima informada pela fabricante, é atribuído ao valor da corrente, tal valor de corrente máxima.

$$P_m = T \cdot \omega \quad (3.1)$$

$$P_e = R \cdot i^2 \quad (3.2)$$

Em uma divisão para obter a eficiência, como mostra a equação (3.3), onde  $\eta$  representa a eficiência do motor no ponto de operação, onde a velocidade e a corrente se tornam constantes.

$$\eta = \frac{P_m}{P_e} \quad (3.3)$$

### 3.4 Modelagem matemática aplicada ao trabalho

Como visto nas Subseções 3.3.1 e 3.3.2, para determinar qual motor é melhor faz-se necessário modelar o motor elétrico na sua aplicação desejada. Como visto na Seção 2.2, o motor elétrico pode ser modelado de forma a obter um sistema de equações diferenciais, que obtém a resposta do sistema quando resolvido. Para a aplicação desejada, não é possível aplicar a equação (2.10), uma vez que o parâmetro momento de inércia não pode ser facilmente mensurado. A solução foi desenvolver uma outra equação diferencial para substituir a equação (2.10).

Uma vez que trata-se de um robô, que pode ser modelado como um corpo que se desloca em linha reta, a segunda equação diferencial que será usada é a segunda lei de Newton (3.4) aplicada no robô como um todo, não somente ao motor.

$$m \cdot \frac{d^2x}{dt^2} = F_{motor} - F_b - F_{res} \quad (3.4)$$

em que  $m$  é a massa do robô usado e  $F_{res}$  é a soma das forças externas agindo no robô. Para a simulação,  $F_{res}$  é uma força constante medida experimentalmente, que simula a força de um robô adversário.

Uma vez que uma série de motores de um fabricante normalmente possui o mesmo motor elétrico, mudando somente a caixa de transmissão, o torque e a velocidade angular do motor passam pela relação de transmissão  $g_r$ , e as equações (3.5) e (3.6) são usadas para adequar o novo sistema.

$$F_{motor} = g_r \cdot \frac{T_m}{r} \quad (3.5)$$

$$\omega = g_r \cdot \frac{v}{r} \quad (3.6)$$

onde  $T_m$  é representado por (2.1),  $g_r$  é a relação de transmissão do motor, ou *gear ratio*,  $r$  representa o raio da roda do robô e  $v$  representa a velocidade linear do robô. Com isso é possível denotar as equações (3.7) e (3.8) representadas para um único motor.

$$F_{motor} = \frac{K_t \cdot g_r \cdot i}{r} \quad (3.7)$$

$$F_b = \frac{B \cdot v \cdot g_r}{r^2} \quad (3.8)$$

Dessa forma, pode-se isolar o termo  $\frac{d^2x}{dt^2}$  e substituir as equações (3.7) e (3.8) em (3.4) obtendo:

$$\frac{d^2x}{dt^2} = \frac{K_t \cdot g_r \cdot i}{m \cdot r} - \frac{B \cdot v \cdot g_r}{m \cdot r^2} - \frac{F_{res}}{m} \quad (3.9)$$

Com a equação (3.9) o sistema de equações (2.11) pode ser alterado. Por se tratar de uma equação diferencial de segunda ordem, há a necessidade de deixá-la de primeira ordem, portanto, na formulação de espaço de estados, essa equação de segunda ordem tornar-se-á duas equações de primeira ordem, como mostra o sistema (3.10).

$$\begin{cases} \frac{di}{dt} = \frac{V_{bat}}{L} - \frac{R \cdot i}{L} - \frac{K_v \cdot v \cdot g_r}{L \cdot r} \\ \frac{dv}{dt} = \frac{K_t \cdot g_r \cdot i}{m \cdot r} - \frac{B \cdot v \cdot g_r}{m \cdot r^2} - \frac{F_{res}}{m} \\ \frac{dx}{dt} = v \end{cases} \quad (3.10)$$

Valores referentes a viscosidade e lubrificação usados no motor normalmente não são apresentados pelas fabricantes e são de difícil aquisição pelo usuário, portanto, como mostrado em (COSTA, 2016), onde  $B$  assume valores da ordem de  $10^{-5}$ , optou-se por simplificar o modelo tratando como constante  $B = 10^{-5}$ . De forma similar, as resistências e indutâncias inerentes ao motor também foram consideradas constantes, uma vez que como o atraso do subsistema mecânico influencia muito mais do que o atraso decorrente do motor elétrico, portanto, fez-se  $R = 7 \Omega$  e  $L = 10 \text{ mH}$ , valores adquiridos experimentalmente nos motores atuais da equipe Rinobot.

É possível, portanto, transformar esse sistema em uma formulação de espaço de estados, um sistema de equações diferenciais possível e numericamente simples de resolver.

$$\frac{d}{dt} \begin{bmatrix} i \\ v \\ x \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_v \cdot g_r}{L \cdot r} & 0 \\ \frac{K_t \cdot g_r}{m \cdot r} & -\frac{B \cdot g_r}{m \cdot r^2} & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ v \\ x \end{bmatrix} + \begin{bmatrix} \frac{V_{bat}}{L} \\ -\frac{F_{res}}{m} \\ 0 \end{bmatrix} \cdot u(t) \quad (3.11)$$

De acordo com qual teste é efetuado pelo *software* (Seção 3.3), as condições iniciais são alteradas e solução do sistema é feita utilizando o método de Runge-Kutta de 4ª ordem apresentado na Seção 2.3.

### 3.5 Ponderação das notas

A nota final de cada motor é resultado de uma ponderação dos resultados dos diferentes parâmetros. Usando como base parte do método apresentado na Seção 2.7, cada motor apresenta uma nota, obtida através dos testes da Seção 3.3. Essas notas passam por uma interpolação linear para que os valores apresentem métrica semelhante com valores entre 0 e 5. A interpolação varia para cada teste, tendo em vista que há parâmetros que são diretamente proporcionais às notas e outros que são inversamente proporcionais às notas. Com uma métrica semelhante para as notas de cada teste, é possível executar a ponderação, como visto na Tabela 2, no entanto, retirando-se a parte de combinação de alternativas, pois os motores são comprados prontos das fabricantes. O resultado é então um *ranking* com as notas finais de todos os motores inseridos no banco de dados.

Os pesos da ponderação são variáveis de acordo com a escolha de foco do motor, apresentada na Seção 4.4, Cada foco e aplicação escolhidos alteram a ponderação usada de forma que o melhor motor resultante atenda àquele foco. Os valores de pesos são decididos subjetivamente, como discutido em (ULRICH KARL T. EPPINGER, 2011), para isso, um grupo de pessoas experientes em competições de sumô se reuniu de modo a opinar e decidir quais dos parâmetros são mais relevantes em um motor para a aplicação deles, e os resultados estão apresentados na Tabela 4.

Foco	Aplic. em luta	Arrancada	Reversão	Eficiência	Preço	Vel. máx.
Minissumô						
Forte	30%	20%	20%	5%	10%	15%
Rápido	10%	25%	25%	5%	5%	30%
Eficiente	15%	15%	15%	30%	10%	15%
Barato	15%	15%	15%	10%	30%	15%

Tabela 4 – Tabela de pesos para cada teste de acordo com o foco do motor.

Fonte: Autor, 2019.

### 3.6 Torque máximo de deslizamento

Como qualquer objeto que se move sobre rodas, o robô está sujeito a situações onde há deslizamento do pneu com o solo, havendo velocidade relativa entre o eles. Quando há deslizamento, fica evidente que o desempenho do robô independe da força produzida pelo motor, uma vez que o torque máximo que será transferido é o torque de atrito dinâmico, que é menor do que o representado por (3.16), e então o fator determinante da vitória passa a ser a força normal presente ( $N$ ), e não o motor em si.

A situação de deslizamento ocorre quando o torque produzido pelo motor é suficientemente alto para gerar uma força que supere a força de atrito estático. Para determinar tal força de atrito estático é necessário calcular a força normal atuante em cada roda para então calcular o torque de atrito. Para isso modelar-se-á o robô como uma viga biapoiada com 2 apoios de primeiro gênero, conforme mostrado na Seção 2.5.

Conforme a Seção 1.2, os robôs previstos pelo *software* são robôs de 2 ou 4 rodas alinhadas, tendo 1 ou 2 eixos de motores, respectivamente. Sua principal característica é a simetria em relação ao comprimento do robô. Tal simetria que torna possível utilizar o modelo de estrutura plana para o robô. Com isso, os valores calculados onde há rodas devem ser divididos por 2. A Figura 12 apresenta como o *software* trabalha com relação a referenciais para cálculo onde a origem é a parte traseira do robô:

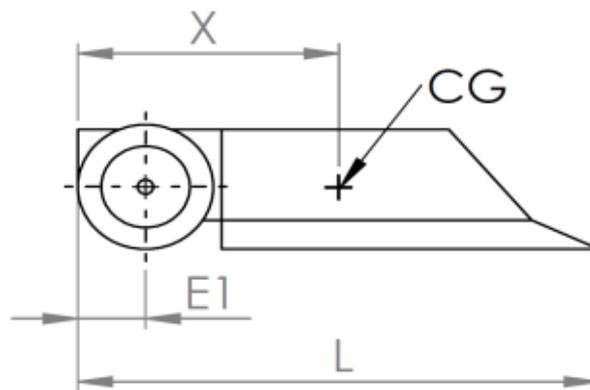


Figura 12 – Esquemático das informações sobre o robô.

Fonte: Autor, 2019.

$X$  = representa a posição do CG em relação ao referencial, onde será inserida a força peso;

$L_{robo}$  = representa o comprimento total do robô, importante para o caso onde o robô tenha apenas 1 eixo com motores, sendo um dos apoios;

$E1$  = representa a distância do eixo de motores até o referencial, sendo um dos apoios;

$E2$  = Não representado pela figura, porém quando o robô tem 4 rodas, e portanto, 2 eixos, representa a posição do segundo eixo em relação ao referencial, sendo esse o segundo apoio.

Para o caso de robôs com 1 eixo, usando as equações (2.18) e (2.19), é possível chegar em:

$$R_{traseiro} = \frac{W}{2} \cdot \left(1 - \frac{1}{L_{robo}} \cdot (X - E1)\right) \quad (3.12)$$

$$R_{dianteiro} = \frac{W}{2 \cdot L_{robo}} \cdot (X - E1) \quad (3.13)$$

onde se considera apenas  $R_{traseiro}$  pois é onde se localizam as rodas do robô, e portanto, é onde não deve ocorrer deslizamento.

De forma semelhante, para o caso de robôs com 2 eixos, as equações são levemente alteradas para:

$$R_{traseiro} = \frac{W}{2} \cdot \left(1 - \frac{1}{E2} \cdot (X - E1)\right) \quad (3.14)$$

$$R_{dianteiro} = \frac{W}{2 \cdot E2} \cdot (X - E1) \quad (3.15)$$

Nesse último caso, para verificar se há deslizamento, considerar-se-á o menor valor de reação, pois esse produzirá um menor torque máximo de atrito.

O torque máximo de atrito então é obtido através da equação (3.16):

$$T_d = \mu \cdot N \cdot r \quad (3.16)$$

onde:

$\mu$  = Coeficiente de atrito estático determinado experimentalmente (Seção 3.8);

$N$  = Valor da força normal ao solo, podendo ser  $R_{traseiro}$  ou  $R_{dianteiro}$ ;

$r$  = Raio da roda.

### 3.6.1 Cálculo do CG do robô

A determinação do parâmetro  $X$  das equações da Seção 3.6 na maioria das vezes não é trivial. A Seção 2.4 apresenta como é feito o cálculo para um corpo unidimensional. Um robô de competição é um projeto espacial de 3 dimensões. Para projetar um robô, uma das partes mais cruciais é a sua modelagem 3D em um *software* de CAD (desenho assistido por computador). Um programa amplamente utilizado no ambiente de modelagem 3D é o *SolidWorks* da empresa Dassaut Systèmes®. Esse *software* utiliza as equações da Seção 2.4 para as 3 dimensões e consegue informar a posição do CG no espaço tridimensional de modelagem. Ele proporciona um ambiente de montagem onde é possível observar todo o modelo do robô no computador e também fornece ferramentas tais quais proporcionam medição de distâncias, verificação de conflitos e conexões, alinhamento de furos, e dentre elas, uma que dispõe na tela as propriedades de massa do robô. Essa ferramenta informa ao usuário a posição do CG em relação a origem da montagem, conforme mostra a Figura 13. Durante a montagem, o projetista é capaz de medir a distância da origem até a parte traseira do robô, determinando então a posição  $X$  do CG para informar no *software* do trabalho.

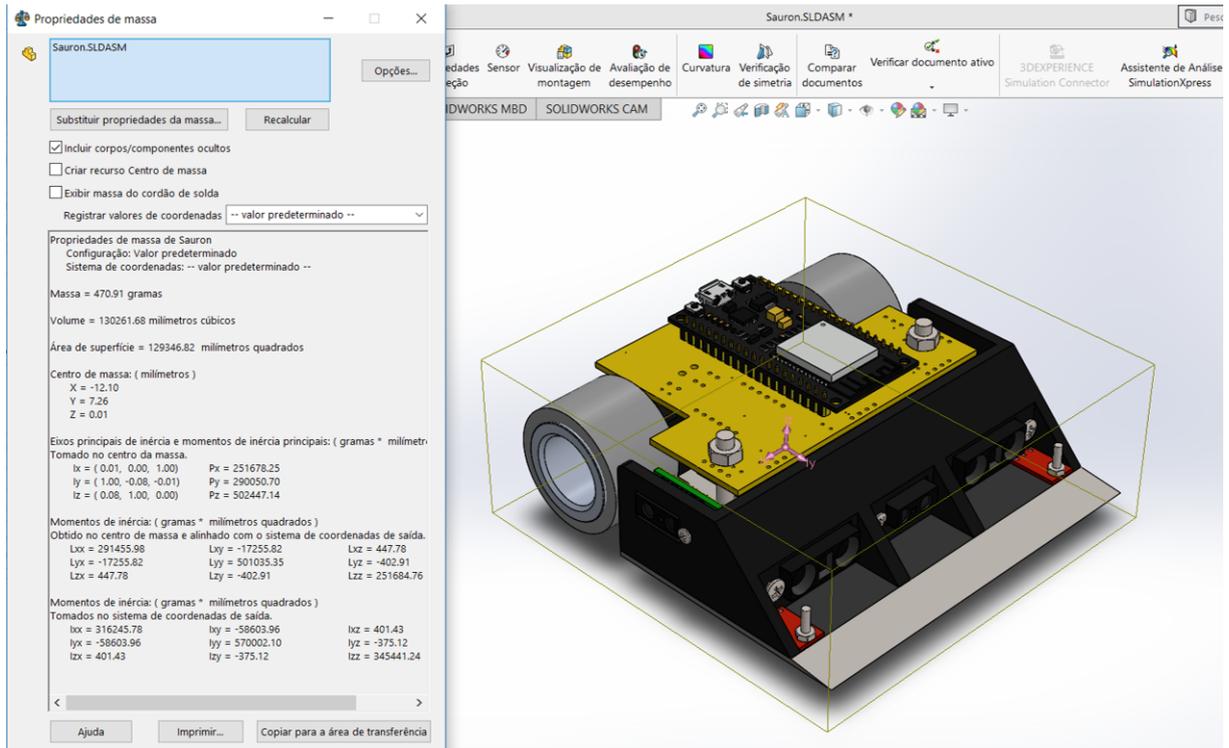


Figura 13 – Tela de informação de propriedades do *SolidWorks*.  
 Fonte: Autor, 2019.

### 3.7 Ensaio de força de tração

O intuito de usar este ensaio é para determinar a força resistente que um robô de minissumô é capaz de realizar em seu adversário. Isso pode ser feito ao simular uma luta de minissumô. O robô que será tratado como corpo de teste deve ser colocado no *dohyō* e preso a uma balança digital (devido aos recursos disponíveis), deverá exercer a maior força possível. Através do valor capturado pela balança é possível obter diretamente a força exercida e substituir seu valor em  $F_{res}$  na equação (3.9) apresentada na Seção 3.4.

O corpo de teste será o robô atual da categoria minissumô da equipe Rinobot, chamado Destiny (Figura 14) que usa motores Pololu (POLOLU, 2019) *High Power Carbon Brushes* (HPCB), com 0,45 kgf.cm de torque máximo, de redução 30:1. Usando a equação (3.5) é possível determinar que a força teórica exercida pelo robô é de 3,53 N. A balança utilizada é uma balança de capacidade máxima de 10 kg conforme mostra a Figura 15. O teste consistiu em fixar a balança de forma que ficasse imóvel durante o teste e prendê-la no Destiny usando barbante, de forma a obter a força exercida pelo robô, conforme Figura 16.

### 3.8 Ensaio do coeficiente de atrito

Para determinar um valor padrão para o coeficiente de atrito, foi utilizado o *dohyō* da equipe Rinobot e o robô Destiny com os pneus StickyMAX S30 (Figura 17), caso o

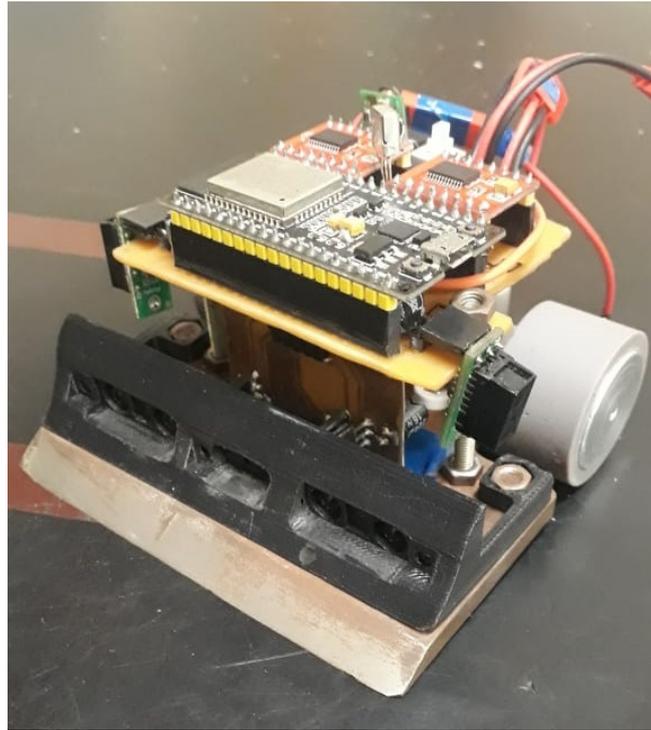


Figura 14 – Robô Destiny da Rinobot.  
Fonte: Autor, 2019.



Figura 15 – Balança usada no experimento.  
Fonte: (HOPSTATION, 2019).

usuário desejar, é possível executar este ensaio com outro pneu de forma a substituir tal valor no programa. O robô foi mantido em cima do *dohyō* com as rodas travadas para não haver rolamento, enquanto o *dohyō* era levantado lentamente até perceber-se um deslizamento do robô. Nesse momento, adquiriu-se o ângulo do *dohyō* com o chão, apresentado na Seção 4.2.

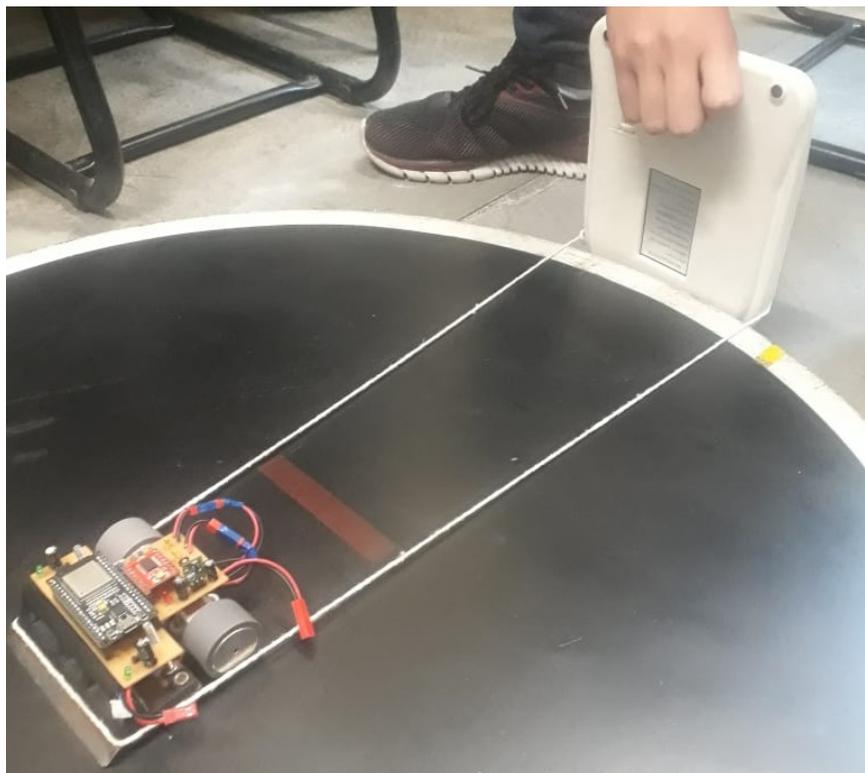


Figura 16 – Foto do ensaio sendo executado.  
Fonte: Autor, 2019.

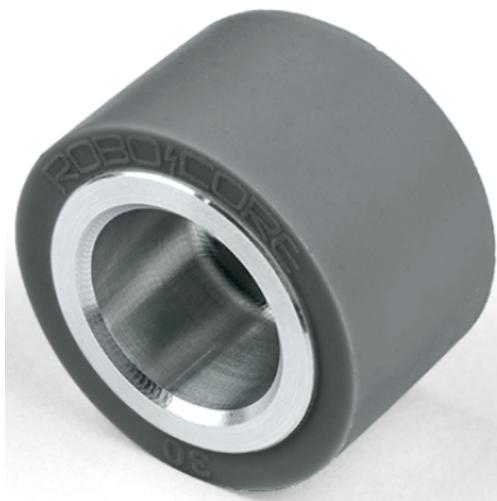


Figura 17 – Roda StickyMAX S30 32 mm.  
Fonte: (ROBOCORE, 2017).

### 3.9 Banco de dados

O banco de dados se refere a todos os motores os quais serão testados e comparados entre si. O usuário tem total acesso para adicionar novos motores, alterar motores já existentes e excluir motores considerados insatisfatórios.

Os motores usados para testar o *software* foram os motores da empresa *Fingertech*,

disponíveis em (FINGERTECH, 2019). As tabelas com os motores da empresa estão disponíveis no Anexo A.

### 3.10 Aplicação na Rinobot

O propósito do *software* desenvolvido foi motivado pela equipe Rinobot, e portanto, esta seção vem apresentar os melhores motores para os robôs atuais da categoria minissumô: Sauron, Destiny, e Blackout. Para isso, é necessário fornecer as informações sobre os robôs, conforme Seção 4.4.3.

As informações são apresentadas na Tabela 5:

Informação	Sauron	Destiny	Blackout
$m$ (g)	448	457	406
$r$ (mm)	12,7	12,7	18,25
$L_{robo}$ (mm)	98	87,5	97,5
$X$ (mm)	36,89	47,69	42,14
Qtd. de motores	2 alinhados	2 alinhados	2 alinhados
$E1$ (mm)	14,02	16,1	18,49
$E2$ (mm)	-	-	-

Tabela 5 – Tabela com as informações dos robôs da Rinobot.

Fonte: Autor, 2019.

Essas informações foram inseridas no *software* e os resultados são apresentados na Seção ??.

## 4 RESULTADOS E ANÁLISES

### 4.1 Resultado do ensaio de tração

Através do ensaio, observou-se que não houve patinação das rodas, mostrando que o torque exercido pelos motores do Destiny não excederam o valor máximo de atrito. O ensaio foi executado 5 vezes de forma que os resultados são apresentados na Tabela 6. Considerando os algarismos significativos de incerteza da balança, após multiplicar pela gravidade, o valor de força exercido pela balança é de  $3,5 \pm 0,3$  N, que é próximo do valor teórico apresentado pela fabricante.

Número do teste	Valor mostrado pela balança (g)
1	333
2	378
3	377
4	368
5	315
Média	Desvio padrão
354,2	28,56

Tabela 6 – Tabela com os valores obtidos pelo ensaio de tração.  
Fonte: Autor, 2019.

### 4.2 Resultado do ensaio do coeficiente de atrito

Conforme a Seção 3.8, com um transferidor, mediu-se o ângulo entre o *dohyō* e o chão ( $\theta$ ). O valor obtido foi de  $\theta = 29,5 \pm 0,5^\circ$ . Com esse ângulo, usou-se (2.21) e determinou-se o valor de coeficiente de atrito  $\mu = 0,56$ .

### 4.3 Validação do modelo

A validação do modelo foi feita através de um código no MATLAB<sup>®</sup>. O código em questão recebe as informações de motores reais, e calcula, através do modelo apresentado na Seção 3.4, as velocidades máximas de saída. Foram inseridos motores aleatórios dos usados no banco de dados e as informações foram confrontadas com os fornecidas pela fabricante de motores *Fingertech*. Os resultados são apresentados na Tabela 7.

O erro apresentado foi calculado usando uma diferença percentual:

$$e_{percent} = \frac{|\omega_{modelo} - \omega_{fabricante}|}{\omega_{fabricante}} \quad (4.1)$$

Disso, nota-se que o maior erro apresentado foi de 2,15% e que, portanto, é válida a utilização do modelo apresentado.

Motor	Valor modelado (rpm)	Valor fabricante (rpm)	Erro (%)
GoldSpark (35:1 / 6 V)	268,938	270	0,39
GoldSpark (86:1 / 22,2 V)	405,8737	406	0,03
GoldSpark (150:1 / 11,1 V)	116,3851	116	0,33
GoldSpark (20:1 / 18,5 V)	1423,7	1455	2,15
SilverSpark (22.2:1 / 7,4 V)	520,869	524	0,6
SilverSpark (100:1 / 6 V)	94,3491	94	0,37
SilverSpark (50:1 / 14,8 V)	464,9987	466	0,21
SilverSpark (33.3:1 / 22,2 V)	1045,6	1049	0,32

Tabela 7 – Tabela com os valores exibidos pelo modelo e os valores reais.

Fonte: Autor, 2019.

#### 4.4 Funcionamento do *software*

A interface, após ser testada por diversos usuários se provou ser intuitiva e apresentar informações para que se faça um uso correto do programa. Esta seção visa explicar tudo o que o programa tem disponível para efetuar o objetivo final de mostrar ao usuário o melhor motor elétrico CC disponível.

##### 4.4.1 Tela principal

A tela principal do *software* é representada pela Figura 18, e é a tela inicial do programa, nela, o usuário é solicitado a informar qual a aplicação do robô, elas são aplicações voltadas para a equipe Rinobot, sendo elas:

- Minissumô;
- Seguidor de linha;
- VSSS (*Very Small Size Soccer*);
- Outros.

Dessas opções, a aplicação relevante é a “Minissumô”, sendo as outras desenvolvidas em trabalhos futuros (5.2). A opção “Outros” é para determinar um motor qualquer para uma aplicação não prevista pela equipe Rinobot, onde será necessária maior atenção do usuário ao preencher os dados necessários. Também é demandado qual o foco desejado pelo usuário no motor que ele deseja. A base de dados não será alterada, no entanto, os pesos são alterados, alterando a ponderação, como mostra a Seção 3.5. Os focos disponíveis de motor são:

- Rápido;
- Forte;



Figura 18 – Tela principal do *software*.  
Fonte: Autor, 2019.

- Eficiente;
- Barato.

Definidas a aplicação e o foco, o usuário pode adequar os motores do banco de dados, acessando a janela “Motores” e adequar as informações referentes ao robô na janela “Dados Externos ao Motor”. É possível utilizar o botão “Calcular” e o programa efetuará a simulação e exibirá uma tela com os resultados.

#### 4.4.2 Tela de motores

A tela de motores, representada pela Figura 19 possibilita que o usuário adicione, exclua, altere e pesquise por motores no banco de dados.

Para adicionar um novo motor ao banco, o usuário, ao clicar no botão “Adicionar Motor”, é apresentado a tela de adição (Figura 20), dispondo de 8 informações referentes aos motores, que são essenciais para o funcionamento do *software*, são elas:

- Fabricante/Modelo, que é como o usuário deverá reconhecer o motor;
- Tensão, que é a tensão elétrica prevista aplicada ao motor;
- Corrente máxima, que é a corrente que o motor utiliza quando tem seu rotor bloqueado;
- Corrente mínima, que é a corrente que o motor utiliza quando sem carga;
- Rotação máxima, que é a rotação máxima quando sem carga;

ID	Fabricante	Redução	Kt(Kg.mm/A)	Kv(V/rpm)	Tensão(v)	Corrente Máx(A)	Corrente Mín(A)	Preço
47	Fingertech/Silver	50	0.352	0.000635728	22.2	1.3	0.09	69€
48	Fingertech/Gold	20	0.352	0.000635728	5	1.3	0.09	39€
49	Fingertech/Gold	35	0.352	0.000635728	5	1.3	0.09	22€
50	Fingertech/Gold	50	0.352	0.000635728	5	1.3	0.09	15€
51	Fingertech/Gold	63	0.352	0.000635728	5	1.3	0.09	12€
52	Fingertech/Gold	86	0.352	0.000635728	5	1.3	0.09	9€
53	Fingertech/Gold	115	0.352	0.000635728	5	1.3	0.09	6€
54	Fingertech/Gold	150	0.352	0.000635728	5	1.3	0.09	5€
55	Fingertech/Gold	20	0.352	0.000635728	7.4	1.3	0.09	58€
56	Fingertech/Gold	35	0.352	0.000635728	7.4	1.3	0.09	33€
57	Fingertech/Gold	50	0.352	0.000635728	7.4	1.3	0.09	23€
58	Fingertech/Gold	63	0.352	0.000635728	7.4	1.3	0.09	18€

Figura 19 – Tela onde são apresentados os motores do banco de dados.  
Fonte Autor, 2019.

- Torque máximo, que é o torque efetuado quando com rotor bloqueado;
- Preço, que é o valor de mercado do referido motor;
- Redução, que é o quando a transmissão consegue reduzir de velocidade e multiplicar de torque para o eixo de saída;
- $K_t$ , que é a constante que relaciona corrente elétrica com o torque efetuado pelo motor;
- $K_v$ , que é a constante que relaciona tensão contra-eletromotriz com a velocidade angular do rotor.

Se não houver um motor selecionado, não é possível apertar o botão de exclusão de motor. Ao selecionar um motor, tal botão exclui o motor desejado do banco de dados totalmente, não havendo retorno. Além disso, o *software* impede a exclusão de vários motores em série, desabilitando o botão excluir após o processo de exclusão, sendo necessária uma nova seleção de motor para repetir o processo.

A pesquisa é efetuada mediante comparação exata sendo *Case Sensitive* (diferencia letras maiúsculas e minúsculas). O usuário seleciona na lista qual aspecto do motor vai

Figura 20 – Janela de adição de novos motores ao banco de dados.

Fonte: Autor, 2019.

utilizar para filtrar, e então insere na caixa de texto as palavras ou valores que deseja buscar no banco de dados e posteriormente, clicando em “Consultar Motor”, a tabela do *software* atualiza para os motores que correspondem à pesquisa, mostrando uma mensagem quando nenhum motor correspondente é encontrado. Para retirar o filtro, e retornar à tabela inicial com todos os motores disponíveis, basta apertar o botão “Atualizar Tabela” e então, novamente, todos os motores do banco de dados estarão disponíveis para visualização.

Para alterar um motor, a janela de alteração é utilizada (Figura 21). Nela, é necessário inserir o identificador (ID) do motor, que está presente na primeira coluna da tabela de motores. O ID é um número gerado automaticamente pelo banco de dados, e é único para cada motor, por isso através dele é possível executar uma alteração. Após inserir o ID do motor o qual se deseja modificar, o *software* já insere nas outras caixas de texto os valores atuais do banco, e o usuário pode modificar e salvar as alterações clicando em “OK”.

#### 4.4.3 Tela de dados externos

Após acertado o banco de dados com os motores os quais se deseja testar, é necessário inserir os dados das condições de contorno, isto é, dados sobre o robô, e sobre o local de aplicação. É necessário antes definir a aplicação desejada, pois para cada aplicação há algumas especificidades. Para o presente trabalho, que está focado na categoria minissumô, a janela apresentada é representada pela Figura 22.

Nessa tela são pedidos os seguintes dados:

Dialog ? X

### Alterar motor:

ID:

Fabricante	Tensão(V)
<input type="text"/>	<input type="text"/>
Corrente Máx(A)	Corrente Mín(A)
<input type="text"/>	<input type="text"/>
Rotação Máx(rpm)	Torque Máx (kgf.mm)
<input type="text"/>	<input type="text"/>
Preço(R\$)	Reducao
<input type="text"/>	<input type="text"/>
Kt (kg.mm/A)	Kv (V/rpm)
<input type="text"/>	<input type="text"/>

Figura 21 – Tela de alteração de motor.  
Fonte: Autor, 2019.

Dialog ? X

### Dados Externos ao Motor

Avisos

Massa do Robô (g):	Gravidade (m/s <sup>2</sup> ):	Quantidade de motores:
<input type="text"/>	<input type="text" value="9.81"/>	<input style="background-color: red; color: white; border: 1px solid red;" type="button" value="Selecione"/>
Raio da Roda (mm):	Coeficiente de Atrito (dinâmico) :	Posição das rodas (mm):
<input type="text"/>	<input type="text" value="0.6"/>	Posição Eixo traseiro:
Comprimento do robô (mm):	Força Resistente (N) :	E1: <input type="text"/>
L: <input type="text"/>	<input type="text"/>	Posição eixo dianteiro:
Centro de Gravidade (mm):		E2: <input type="text"/>
X: <input type="text"/>		

Figura 22 – Tela de inserção de dados externos.  
Fonte: Autor, 2019.

- Massa do robô;
- Raio da roda do robô;
- Comprimento total do robô, como mostrado pelo desenho apresentado na tela;
- Centro de gravidade, com referência como mostrada no desenho;
- Gravidade atuante no local (valor padrão de  $9,81 \text{ m/s}^2$ );
- Coeficiente de atrito estático com o *dohyō* (caso do minissumô), sendo um valor padrão obtido experimentalmente (Seção 3.8);
- Força resistente que os motores deverão superar, podendo esta ser determinada experimentalmente (Seção 4.1);
- A quantidade de motores, como abordado na Seção 1.2, que atua diretamente no esforço dos motores;
- A posição das rodas, que contempla robôs de 2 rodas alinhadas, que apresenta 1 eixo apenas, ou robôs de 4 rodas, que apresenta 2 eixos, como mostrado pelo desenho na tela.

Ao clicar no botão “OK”, os dados inseridos são conferidos de acordo com as regras de minissumô (Subseção 3.1.1) e caso os valores sejam válidos, são salvos para serem usados nos cálculos.

#### 4.4.4 Tela de resultados

A tela de resultados é mostrada pela Figura 23. Nela, os motores já foram testados, já foram feitas as ponderações e é mostrada na tela a classificação final, com os motores em ordem decrescente de nota final (variando de 0 a 5). Também é informado ao usuário se há ou não deslizamento. Caso o usuário queira saber as especificações de um motor, deve selecionar o motor desejado e clicar em “Detalhes Motor” de forma que uma nova janela aparece com os dados daquele motor no banco de dados, conforme mostra também na tela de motores (Subseção 4.4.2).

Caso o usuário também queira saber o valor de cada nota em separado, é possível obtê-las através de um gráfico polar. Ao clicar em “Gráfico Polar” o gráfico aparece (Figura 24) onde cada eixo representa uma nota, de forma que, em sentido horário, os valores representam:

- Teste de arrancada;
- Teste de reversão;

ID	Fabricante/Modelo	Redução	Nota Final	Deslizamento
74	Fingertech/Gold	35	3.8071	Não derrapa
68	Fingertech/Gold	50	3.70221	Não derrapa
71	Fingertech/Gold	35	3.69502	Não derrapa
46	Fingertech/Silver	33.3	3.68224	Não derrapa
36	Fingertech/Silver	50	3.60293	Não derrapa
64	Fingertech/Gold	63	3.58617	Não derrapa
67	Fingertech/Gold	35	3.57549	Não derrapa
41	Fingertech/Silver	33.3	3.57135	Não derrapa
72	Fingertech/Gold	50	3.5525	Não derrapa
63	Fingertech/Gold	50	3.53965	Não derrapa
31	Fingertech/Silver	83.3	3.53043	Não derrapa

Figura 23 – Tela de resultados do *software*.

Fonte: Autor, 2019.

- Teste de aplicação em luta;
- Eficiência;
- Preço;
- Velocidade final.

#### 4.5 Aplicação na Rinobot

Os valores foram inseridos no *software* conforme Seção 3.10, bem como o foco desejado no motor (Seção 4.4.1). Por se tratar de uma luta de sumô, o foco escolhido foi o de força. Os melhores motores indicados pelo *software* para cada robô foram:

- Sauron: *Fingertech* GoldSpark 35:1, com  $V_{bat} = 22,2$  V;
- Destiny: *Fingertech* GoldSpark 35:1, com  $V_{bat} = 22,2$  V;
- Blackout: *Fingertech* SilverSpark 50:1 com  $V_{bat} = 22,2$  V;

Esses motores tiveram notas individuais que estão apresentadas na Tabela 8 e os detalhes dos motores emitidos pelo *software* estão apresentados na Figura 26.

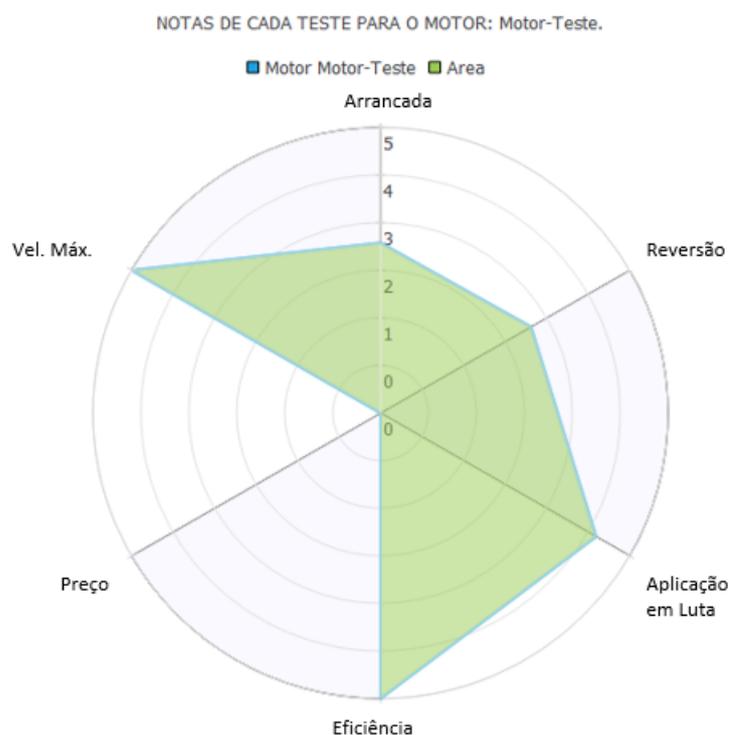


Figura 24 – Gráfico polar com as notas individuais.

Fonte: Autor, 2019.

Teste	Sauron	Destiny	Blackout
	<i>Fingertech/Gold</i> 35:1 22,2 V	<i>Fingertech/Gold</i> 35:1 22,2 V	<i>Fingertech/Silver</i> 50:1 22,2 V
Nota Final	3.8071	3.8066	3.7308
Arrancada	4,36626	4,36481	4,67004
Reversão	4,375	4,37402	4,67621
Luta	4,7844	4,78442	4,69487
Eficiência	4,8338	4,8338	5
Preço	0,992794	0,992794	0
Vel. máxima	1,88371	1,88372	1,35419

Tabela 8 – Notas obtidas pelo melhor motor indicado pelo *software* para os robôs da Rinobot.

Fonte: Autor, 2019.

Os motores indicados como melhores, as vezes, podem ser inviáveis para o usuário comprar, por quaisquer motivos, portanto, na Tabela 9 há os segundos colocados para comparação. Pode ser observado que os segundos colocados, apesar de obterem notas maiores para o teste de arrancada e reversão (por ter maior redução, o torque maior ajuda a chegar mais rápido na velocidade máxima, que é menor) esses motores ganham nota mais baixa na simulação da luta (que é o parâmetro com maior peso), na eficiência e na velocidade máxima. Essas diferenças estão mostradas graficamente para o robô Destiny na Figura 25.

Teste	Sauron	Destiny	Blackout
	<i>Fingertech</i> /Gold 50:1 14,8 V	<i>Fingertech</i> /Gold 50:1 14,8 V	<i>Fingertech</i> /Gold 50:1 18,5 V
Nota Final	3.7022	3.7017	3.7152
Arrancada	4,7203	4,7167	4,6773
Reversão	4,7236	4,7244	4,6828
Luta	4,5631	4,5631	4,6272
Eficiência	4,1968	4,1968	3,7302
Preço	0,9928	0,9928	0,9928
Vel. máxima	0,9027	0,9027	1,1285

Tabela 9 – Notas obtidas pelo segundo melhor motor.  
Fonte: Autor, 2019.

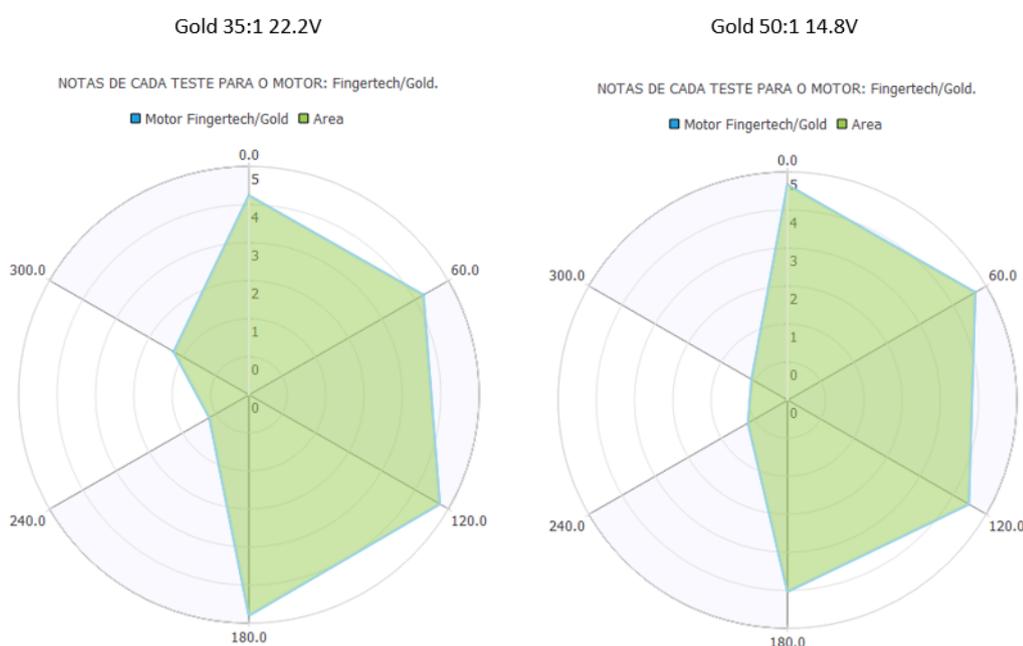


Figura 25 – Gráficos polares dos 2 melhores motores indicados pelo *software*.  
Fonte: Autor, 2019.

Também é interessante notar que os robôs de minissumô costumam ter características parecidas (robôs com aproximadamente 500 g, normalmente utilizando todo o espaço de 100 × 100 mm permitido pelas regras), portanto, há uma tendência de motores sugeridos. Na fabricante *Fingertech*, motores com tensões altas, sejam SilverSpark ou GoldSpark, com reduções entre 30:1 e 50:1 são considerados os melhores pelos cálculos do *software*.

Dos motores considerados melhores para a aplicação na Rinobot, notou-se que a tensão de 22.2 V é comum para todos os robôs. Isso se deve ao fato de que a maior tensão proporciona maior potência elétrica aos motores, gerando também mais potência mecânica. Outra constante foram as reduções baixas, motores de redução alta apesar de fortes, não apresentam velocidade suficientemente grandes para ganhar notas altas no

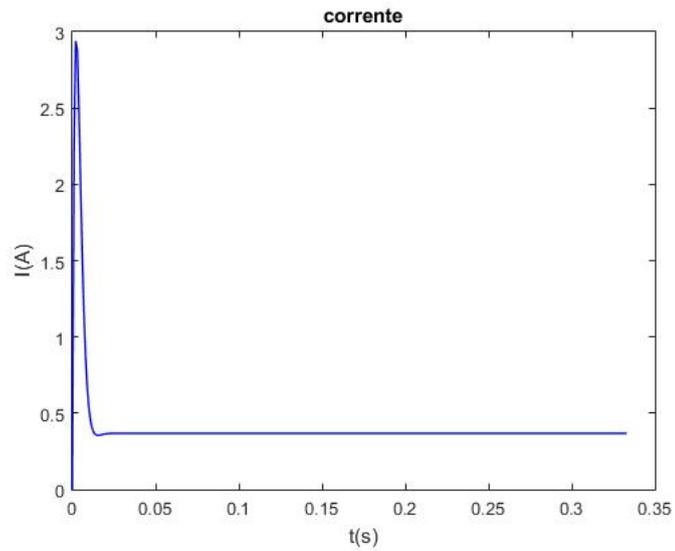
teste de aplicação em luta (um robô muito lento tem maiores chances de ser flanqueado).

Teste	Sauron	Destiny	Blackout
	<i>Fingertech</i> /Gold 35:1 22,2 V	<i>Fingertech</i> /Gold 35:1 22,2 V	<i>Fingertech</i> /Silver 50:1 22,2 V
Nota Final	3.8071	3.8066	3.7308
Arrancada	4,36626	4,36481	4,67004
Reversão	4,375	4,37402	4,67621
Luta	4,7844	4,78442	4,69487
Eficiência	4,8338	4,8338	5
Preço	0,992794	0,992794	0
Vel. máxima	1,88371	1,88372	1,35419

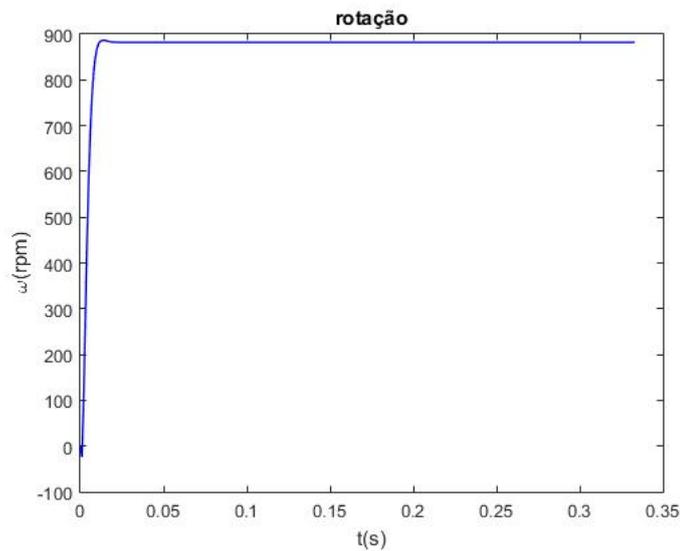
Figura 26 – Detalhes dos motores emitidos pelo *software* para cada robô.

Fonte: Autor, 2019.

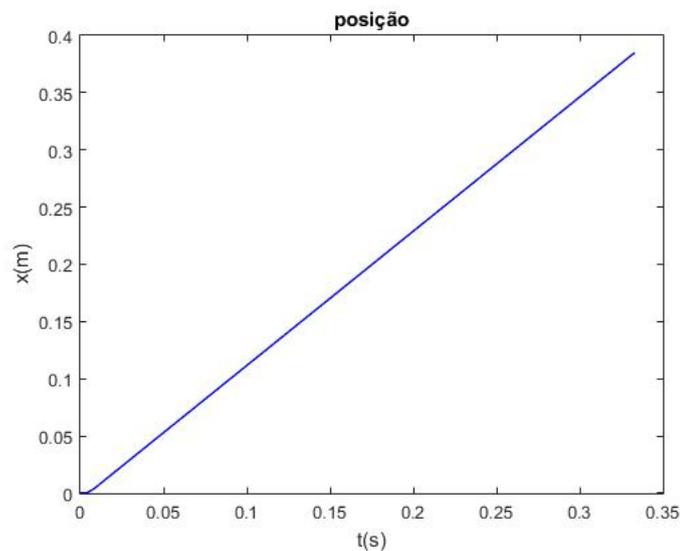
As lutas simuladas pelo *software* também podem ser feitas no MATLAB<sup>®</sup>, onde é possível realizar uma plotagem das respostas de corrente, velocidade linear, posição e rotação do motor, para cada motor (1 por vez), tornando possível visualizar o comportamento do robô com tal motor durante a luta. O código do MATLAB<sup>®</sup> está disponível no Apêndice A. Para o motor *Fingertech* GoldSpark 35:1 de tensão  $V_{bat} = 22,2$  V, as respostas da simulação são apresentadas pelas Figuras 27a, 27b e 27c.



(a) Corrente durante a simulação.



(b) Rotação durante a simulação.



(c) Posição durante a simulação.

Figura 27 – Desempenho do motor *Fingertech* GoldSpark 35:1 22,2 V na simulação de luta.

Fonte: Autor, 2019.

## 5 CONCLUSÕES

### 5.1 Conclusões

O presente trabalho foi capaz de apresentar uma solução para a escolha de motores elétricos de corrente contínua utilizando uma abordagem automatizada e com base físico-matemática para garantir, mesmo com algumas idealizações no modelo, que o motor indicado pelo *software* seja o mais adequado para o robô exato inserido. Dessa forma, a equipe Rinobot é capaz de projetar mais eficientemente robôs de minissumô e testar quaisquer motores existentes desde que tenha as informações necessárias do fabricante.

### 5.2 Sugestões para trabalhos futuros

A equipe Rinobot apresenta 4 outras categorias de robótica, dentre as quais 2 também desenvolvem seus robôs do zero e necessitam de motores para suas aplicações: são elas as categorias *followline* e VSSS. Futuramente, elas podem se aproveitar deste *software*, pois o atual programa já apresenta espaço para a ampliação das aplicações, no entanto será necessário um estudo aprofundado de cada categoria de forma que os testes e simulações sejam adequados a elas.

Também é possível ampliar o *software* para aplicações mais abrangentes, tratada como “Outros”, nessa aplicação os motores estariam sujeitos a um teste mais genérico e o usuário deverá ser mais cauteloso com a informação de entrada. Além disso, otimizações podem ser realizadas no código do programa, de forma a melhorar o gasto de memória e torná-lo mais eficiente bem como adicionar ao *software* a janela de plotagem gráfica linear com a *desempenho* do motor durante a simulação, como mostrado na Seção ?? no MATLAB<sup>®</sup> (Figura 27).

Os motores indicados por esse trabalho já estão na lista para futuras compras para a categoria minissumô da Rinobot. Uma averiguação experimental dos resultados apresentados pelo *software* atual é capaz de confirmar os dados apresentados por tal.

## REFERÊNCIAS

- COSTA, C. *Notas de Aula: Modelo matemático de motores de corrente contínua*. 2016. Disponível em <<http://professorcesarcosta.com.br>>. Acesso em 25 set. 2019.
- FINGERTECH. *Motores usados no banco de dados*. 2019. Disponível em <<https://www.fingertechrobotics.com/products.php?cat=8>>. Acesso em 25 set. 2019.
- FLITZGERALD A. E. KINGSLEY, C. U. S. D. *Electric Machinery*. [S.l.]: New York, McGraw Hill, 2002. 6<sup>a</sup> ed.
- GARCIA, C. *Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos*. [S.l.]: São Paulo, EdUsp, 2016. 2<sup>a</sup> ed.
- HIBBELER, R. C. *Estática. Mecânica para Engenharia*. [S.l.]: São Paulo: Pearson Prentice Hall, 2011. 12<sup>a</sup> ed.
- HOPSTATION. *Balança Digital 10 kg*. 2019. Disponível em <<https://www.hopstation.com.br/balanca-digital-10kg>>. Acesso em 25 set. 2019.
- MEGGIOLARO, M. A. et al. *Tutorial em Robôs de Combate*. 2016. Disponível em <<https://www.riobotz.com/tutorial>>. Acesso em 25 set. 2019.
- NUSSENZVEIG, H. M. *Curso de Física Básica*. [S.l.]: São Paulo: Edgard Blucher, 2002. v. 1. 4<sup>a</sup> ed.
- POLOLU. *Pololu micro metal gearmotors*. 2019. Disponível em <<https://www.pololu.com/category/60/micro-metal-garmotors>>. Acesso em 25 set. 2019.
- ROBOCORE. *Regras Sumô*. 2017. Disponível em <<https://www.robocore.net>>. Acesso em 25 set. 2019.
- ROBOT-DIALOGS. *The Translator... Mini-sumo Bot*. 2019. Disponível em <<http://www.robotdialogs.com/2012/06/translatormini-sumo-bot.html>>. Acesso em 25 set. 2019.
- ROBOT-DOMESTICI. *Zumo Robot for Arduino*. 2019. Disponível em <<https://www.robot-domestici.it/joomla/component/virtuemart>>. Acesso em 25 set. 2019.
- ROBOT-NEST. *Mini Sumo Robot Sumoboy 2.0*. 2019. Disponível em <<http://www.robot-nest.com>>. Acesso em 25 set. 2019.
- SANTOS, E. I. *O Polinômio e Série de Taylor: Um Estudo com Aplicações*. 2017. Programa de Pós-graduação em Matemática, Universidade Federal da Paraíba.
- TENTACLE, T. *Team Tentacle Torque/Amp-hour calculator*. 2018. Disponível em <[http://runamok.tech/squid/newtorquecalc\\_m.htm](http://runamok.tech/squid/newtorquecalc_m.htm)>. Acesso em 25 set. 2019.
- ULRICH KARL T. EPPINGER, S. D. *Product Design and Development*. [S.l.]: McGraw-Hill Education, 2011.

## APÊNDICE A – CÓDIGO MATLAB<sup>®</sup> DE DESEMPENHO DE 1 MOTOR

```

1 close all
2 clear all
3 clc
4
5 %codigo teste para integracao Runge-Kutta de 4 ordem
6 %sistema de equacoes do minisumo
7
8 %variaveis (SI)
9 R = 7; %7 ohm
10 L = 0.01; %10mH
11 k_w = 1/(1573*pi()/30); %fingertech
12 k_t = 0.352/100; %fingertech
13 r = 0.0127;
14 m = 0.110;
15 V = 6;
16 GearRatio = 10;
17 F_res = 0; %forca resistente (adquirida experimentalmente)
18 B = 10^-5;
19
20 h = 0.001; %passo da integracao — 1 ms
21
22
23 %condicoes iniciais
24 i_0 = 0;
25 v_0 = 0;
26 w_0 = 0;
27 x_0 = 0;
28 t_0 = 0;
29 X_0 = [i_0;v_0;x_0];
30
31 %criterio de parada — raio do dohyo
32 x_limite = 0.385;
33
34 %declaracao das veriaveis operantes na integracao
35 X = X_0;
36 t = t_0;

```

```

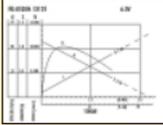
37
38 %declaracao dos vetores de armazenamento
39 cont = 1;
40 MatrizResultados = [];
41 MatrizResultados(1,cont) = X(1);%corrente
42 MatrizResultados(2,cont) = X(2);%velocidade
43 MatrizResultados(3,cont) = X(3);%posicao
44 MatrizResultados(4,cont) = t_0;%tempo
45
46 %%inicio do metodo de integracao
47 while(X(3)<x_limite && X(3)>-x_limite)
48     %armazenamento das variaveis
49     MatrizResultados(1,cont) = X(1);%corrente
50     MatrizResultados(2,cont) = X(2);%velocidade
51     MatrizResultados(3,cont) = X(3);%posicao
52     MatrizResultados(4,cont) = t;
53
54     %equacoes do sistema
55     eq1 = V/L -((R/L)*X(1)) -(((k_w*GearRatio)/(L*r))*X(2));
56     eq2 = ((k_t*GearRatio)/(m*r))*X(1)-(F_res/m)- (B)/(m*r^2)*X
57         (2));
58     eq3 = X(2);
59
60     F1 = [eq1; eq2; eq3];
61     F2 = F1 + (h/2)*F1;
62     F3 = F1 + (h/2)*F2;
63     F4 = F1 + h*F3;
64
65     X_n = X + (h/6)*(F1+ 2*F2 + 2*F3 + F4);
66     %X_n = X+h*F1;
67
68     t = t+h;
69     X = X_n;
70     cont=cont+1;
71
72 end
73
74 omega_rpm = (60/(2*pi*r))*MatrizResultados(2,:);
75 %apresentacao dos resultados
76 figure(1);

```

```
75 plot(MatrizResultados(4,:),MatrizResultados(3,:));
76 title('posicao');
77
78 figure(2);
79 plot(MatrizResultados(4,:),MatrizResultados(2,:));
80 title('velocidade linear');
81
82 figure(3);
83 plot(MatrizResultados(4,:),MatrizResultados(1,:));
84 title('corrente');
85
86 figure(4);
87 plot(MatrizResultados(4,:),omega_rpm);
88 title('rotacao');
```

## ANEXO A – MOTORES USADOS NO BANCO DE DADOS

<p><b>Motor (Mabuchi FK-050SH-13125) Specifications</b></p> <p>Operating Voltage: 3V - 18.5V (&gt;10V decreases motor life)          Nominal Voltage: 6Vdc          No Load RPM: 11530rpm          No Load Current: 0.09A          Stall Current: 1.3A          Stall Torque: 0.0461kg-cm (0.64oz-in)          Kt: 0.0352kg-cm/A (0.49oz-in/A)          Kv: 1573rpm/V          Peak Efficiency: 52%          RPM @ Peak Eff: 9100          Current @ Peak Eff: 0.34A          Click <a href="#">here</a> for the motor torque, current, rpm and efficiency curve.</p>	<p><b>Physical Specifications</b></p> <p>Gearbox Length: 11.6mm (0.455in)          Motor Length: 28.7mm (1.13in)          Total Gearmotor Length: 40.3mm (1.59in)          Gearbox Diameter: 15.5mm (0.61in)          Shaft Diameter: 3mm (0.12in) with flat along length          Shaft Length: 38mm (1.5in)          Mounting Holes (2): #2-56 spaced 11mm (0.433in) apart -do not use screws that protrude more than 4mm (0.156in) into gearbox          Weight (all ratios): 28grams (0.99oz)</p>
---	---



Some ratio gearboxes can be interchanged because they use the same pinion gears on the motor.  
 20, 35, 50, 63: 18T pinions  
 86, 115, 150, 250: 16T pinions  
 360: 14T pinion  
 So (for example) you can remove the 50:1 ratio gearbox from its motor and replace it with a 35:1 ratio gearbox for more speed - without needing to desolder the motor! (Might want to swap the stickers so you know what's in the robot!)

No Load Speed @ Gear Ratio/Voltage							
Speed (rpm)	Voltage						
Gear ratio	5	6	7.4	11.1	14.8	18.5	22.2
20 :1	393	472	582	873	1164	1455	1746
35 :1	225	270	333	499	665	831	998
50 :1	157	189	233	349	466	582	698
63 :1	125	150	185	277	370	462	554
86 :1	91	110	135	203	271	338	406
115 :1	68	82	101	152	202	253	304
150 :1	52	63	78	116	155	194	233
250 :1	31	38	47	70	93	116	140
360 :1	22	26	32	49	65	81	97

(Click to Enlarge)

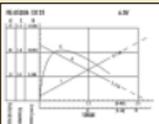
Motors have been tested to work up to 22.2V (6 lipoly cells), but don't expect the motor to run that high forever!

Maximum Torque @ Ratio/Voltage (kg-cm) (oz-in)							
Torque (kg-cm)	Voltage						
Gear ratio	5	6	7.4	11.1	14.8	18.5	22.2
20 :1	0.63	0.75	0.93	1.40	1.86	2.33	2.79
35 :1	1.10	1.32	1.63	2.44	3.26	4.07	4.88
50 :1	1.57	1.89	2.33	3.49	4.65	5.82	6.98
63 :1	1.98	2.38	2.93	4.40	5.86	7.33	8.79
86 :1	2.70	3.24	4.00	6.00	8.00	10.00	12.00
115 :1	3.61	4.34	5.35	8.02	10.70	13.37	16.05
150 :1	4.71	5.66	6.98	10.47	13.96	17.45	20.93
250 :1	7.86	9.43	11.63	17.45	23.26	29.08	34.89
360 :1	11.32	13.58	16.75	25.12	33.49	41.87	50.24

(Click to Enlarge)

Exceeding 6.84kg-cm (95oz-in) will damage the output gear stage. Do not stall the motors highlighted in red at the indicated voltage.

Figura 28 – Motores *Fingertech* GoldSpark, de Fingertech Robotics, 2019.

<p><b>Motor (Mabuchi FK-050SH-13125) Specifications</b>                  Operating Voltage: 3V - 18.5V (&gt;10V decreases motor life)                  Nominal Voltage: 6Vdc                  No Load RPM: 11530rpm                  No Load Current: 0.09A                  Stall Current: 1.3A                  Stall Torque: 0.0461kg-cm (0.64oz-in)                  Kt: 0.0352kg-cm/A (0.49oz-in/A)                  Kv: 1573rpm/V                  Peak Efficiency: 52%                  RPM @ Peak Eff: 9100                  Current @ Peak Eff: 0.34A                  Click <a href="#">here</a> for the motor torque, current, rpm and efficiency curve.</p> 	<p><b>Physical Specifications</b>                  Gearbox Lengths:                  ratios 11: 11.5mm (0.45in)                  ratios 22, 33, 50: 12.7mm (0.50in)                  ratios 83, 100: 14.3mm (0.56in)                  ratios 200, 300: 16.4mm (0.65in)                  ratios 600: 17.9mm (0.70in)                  Motor Length: 28.7mm (1.13in)                  Motor weight (no gearbox): 17g (0.60oz)                  Gearbox Diameter: 16mm (0.63in)                  Shaft Diameter: 3mm (0.12in) with flat along length                  Shaft Length: 38mm (1.5in)                  Mounting Holes (2): #2-56 spaced 11mm (0.433in) apart - do not use screws that protrude more than 2mm (0.078in) into gearbox                  Weight:                  ratios 11: 28g (0.99oz)                  ratios 22, 33, 50, 83, 100: 29g (1.02oz)                  ratios 200, 300: 31g (1.09oz)                  ratios 600: 32g (1.13oz)</p>
--	--

All ratio gearboxes can be interchanged because they use the same 14T pinion gears on the motor. So (for example) you can remove the 50:1 ratio gearbox from its motor and replace it with a 33:1 ratio gearbox for more speed - without needing to desolder the motor! (Might want to swap the stickers so you know what's in the robot!)

No Load Speed @ Gear Ratio/Voltage								Maximum Torque @ Ratio/Voltage (kg-cm) (oz-in)							
<b>Speed (rpm)</b>	Voltage							<b>Torque (kg-cm)</b>	Voltage						
<b>Gear ratio</b>	5	6	7.4	11.1	14.8	18.5	22.2	5	6	7.4	11.1	14.8	18.5	22.2	
11.1 :1	709	850	1049	1573	2097	2622	3146	0.35	0.42	0.52	0.77	1.03	1.29	1.55	
22.2 :1	354	425	524	786	1049	1311	1573	0.70	0.84	1.03	1.55	2.07	2.58	3.10	
33.3 :1	236	283	350	524	699	874	1049	1.05	1.26	1.55	2.32	3.10	3.87	4.65	
50 :1	157	189	233	349	466	582	698	1.57	1.89	2.33	3.49	4.65	5.82	6.98	
83.3 :1	94	113	140	210	279	349	419	2.62	3.14	3.88	5.81	7.75	9.69	11.63	
100 :1	79	94	116	175	233	291	349	3.14	3.77	4.65	6.98	9.30	11.63	13.96	
200 :1	39	47	58	87	116	146	175	6.29	7.54	9.30	13.96	18.61	23.26	27.91	
300 :1	26	31	39	58	78	97	116	9.43	11.32	13.96	20.93	27.91	34.89	41.87	
600 :1	13	16	19	29	39	49	58	18.86	22.63	27.91	41.87	55.82	69.78	83.74	
(Click to Enlarge)								(Click to Enlarge)							
Motors have been tested to work up to 22.2V (6 lipoly cells), but don't expect the motor to run that high forever!								Exceeding 9.72kg-cm (135oz-in) will damage the output gear stage. Do not stall the motors highlighted in red at the indicated voltage.							

Figura 29 – Motores *Fingertech* SilverSpark, de Fingertech Robotics, 2019.

**ANEXO B – TERMO DE AUTENTICIDADE****UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA****Termo de declaração de autenticidade de autoria**

Declaro, sob as penas da lei e para os devidos fins, junto à Universidade Federal de Juiz de Fora, que meu Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia Mecânica é original, de minha única e exclusiva autoria. E não se trata de cópia integral ou parcial de textos e trabalhos de autoria de outrem, seja em formato de papel, eletrônico, digital, áudio-visual ou qualquer outro meio.

Declaro ainda ter total conhecimento e compreensão do que é considerado plágio, não apenas a cópia integral do trabalho, mas também de parte dele, inclusive de artigos e/ou parágrafos, sem citação do autor ou de sua fonte.

Declaro, por fim, ter total conhecimento e compreensão das punições decorrentes da prática de plágio, através das sanções civis previstas na lei do direito autoral<sup>1</sup> e criminais previstas no Código Penal<sup>2</sup>, além das cominações administrativas e acadêmicas que poderão resultar em reprovação no Trabalho de Conclusão de Curso.

Juiz de Fora, 05 de Novembro de 2019.

---

Nome completo – Rodrigo Perobeli Silva Costa  
Matrícula: 201571023 – CPF: 088571686-89

<sup>1</sup> LEI N° 9.610, DE 19 DE FEVEREIRO DE 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

<sup>2</sup> Art. 184. Violar direitos de autor e os que lhe são conexos: Pena – detenção, de 3 (três) meses a 1 (um) ano, ou multa.