

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS**  
**UNIDADE ACADÊMICA DE GRADUAÇÃO**  
**CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

**BRUNO NUNES TOSO**

**SISTEMA DE OTIMIZAÇÃO DO PERFIL DE MOVIMENTO DE UM ROBÔ  
OMNIDIRECIONAL AUTÔNOMO COM BASE EM PONTOS DE INTERESSE**

**São Leopoldo**

**2022**

BRUNO NUNES TOSO

**SISTEMA DE OTIMIZAÇÃO DO PERFIL DE MOVIMENTO DE UM ROBÔ  
OMNIDIRECIONAL AUTÔNOMO COM BASE EM PONTOS DE INTERESSE**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação, pelo Curso de Engenharia de Controle e Automação da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. Rodrigo Marques de Figueiredo

São Leopoldo

2022

*Dedico este trabalho á meus pais, que me apoiam e incentivam. Sem eles, esta jornada não seria possível.*

## AGRADECIMENTOS

Agradeço primeiramente á meus alunos e amigos da equipe de FRC Under Control, do colégio Marista Pio XII, onde sou inspirado diariamente a buscar crescimento e inovação todos os dias. Em especial aos colegas Henrique Schimitz, pelo auxílio no desenvolvimento e construção do módulo contido neste trabalho, Rafaela Werle, pelo auxílio com a modelagem 3D e Luca Carvalho pelos *insights* de software. Agradeço também á meus amigos e sócios, Filipe Ghesla e Daniel Hanauer, que nos momentos de ausência sempre compreenderam que o futuro é feito a partir da constante dedicação ao presente.

Aos meus professores do curso de Engenharia de Controle e Automação da Unisinos, em especial a meu orientador Prof. Dr. Rodrigo Marques de Figueiredo, pela paciência e compreensão ao longo desta caminhada.

Por fim, reitero meu agradecimento a meus pais, Andersom Toso e Elisabete Nunes Toso, que, além de me proporcionarem com a motivação e apoio incondicional para esta conquista, me inspiram a ser a minha melhor versão todos os dias.

## RESUMO

A implementação de competições de robótica educacional fomenta o interesse do jovem por áreas de STEM (*Science, Technology, Engineering and Mathematics*), além de motivar o desenvolvimento de novas tecnologias dentro deste campo, o impacto que estes eventos possuem no desenvolvimento de robôs autônomos para a indústria já é visível. Este trabalho visa o desenvolvimento e implementação de um sistema de movimentação para robôs omnidirecionais utilizando o formato de movimentação do tipo *Swerve*, já comum nestas competições, porém utilizando sistema gráfico baseado em pontos de interesse. Este sistema permite que um usuário rapidamente possa construir ou modificar a trajetória estipulada para o robô. Para isto, é realizado o desenvolvimento e construção de um protótipo de um módulo de movimentação, que utiliza sensores de encoders e uma malha de controle PID (Proporcional, Integral e Derivativo) para seguir a trajetória estipulada pelo usuário. Os testes e implementações do sistema mostram que ele é capaz de ampliar significativamente a performance de uma equipe de FRC (*FIRST Robotics Competition*), como também demonstram que este mesmo projeto pode ser aplicado em uma escala industrial.

**Palavras-chave:** Robô autônomo omnidirecional. Gerador de trajetória. Interpolação de caminho. Seguidor de caminho.

## LISTA DE FIGURAS

Figura 1 – AMRs sendo utilizados para o transporte de produtos em um pavilhão.....	14
Figura 2 – Exemplo de um módulo de tração <i>Swerve</i> . .....	15
Figura 3 – Exemplo de um perfil de movimento trapezoidal. ....	16
Figura 4 – Diferenças entre um caminho e uma trajetória. ....	17
Figura 5 – Caminho gerado pelo algoritmo de <i>Discrete point path planning</i> . ....	19
Figura 6 – Interface gráfica para definição das coordenadas de interesse.....	20
Figura 7 – Resultados do sistema de planejamento de caminho com função <i>Spline</i> .....	21
Figura 8 – Estrutura do robô modelado. ....	22
Figura 9 – Representação da manobra Ackerman dupla. ....	23
Figura 10 – Trajetória com movimentos de rotação e translação simultâneos.....	24
Figura 11 – Exemplo de um perfil de velocidade resultante do algoritmo proposto.....	25
Figura 12 – Diagrama da malha de controle utilizada pelo autor.....	27
Figura 13 – Quadra de jogo para o desafio da FRC de 2020, <i>Infinite Recharge</i> . ....	30
Figura 14 – Robô de FRC para o desafio de 2015(esquerda) e 2016(direita). ....	30
Figura 15 – Diagrama descrevendo a sequência de desenvolvimento do projeto. ....	31
Figura 16 – Diagrama do módulo <i>Swerve</i> . ....	33
Figura 17 – Diagrama funcional do sistema de controle .....	34
Figura 18 – Diagrama detalhado do gerador de trajetória. ....	36
Figura 19 – Diagrama de blocos simplificado para a malha de controle. ....	37
Figura 20 – Primeiro modelo do módulo <i>swerve</i> para prototipagem. ....	39
Figura 21 - Projeto final do módulo construído. ....	40
Figura 22 - Módulo Swerve para testes construído. ....	41
Figura 23 - Módulo fixado ao suporte para realização de testes. ....	42
Figura 24 - Montagem completa do sistema de prototipagem. ....	42
Figura 25 - Diagrama de funcionamento do gerador de trajetória. ....	44
Figura 26 - Interface utilizada para definir os pontos de interesse pelo usuário. ....	45
Figura 27 - Exemplo de trajetória gerada pelo sistema. Em verde, velocidade da roda e em preto, orientação do módulo. ....	46
Figura 28 - Pontos de interesse utilizados na análise do sistema, da esquerda para direita T1, T2 e T3.....	49
Figura 29 - Caminhos obtidos utilizando os pontos de interesse T1(a), T2(b) e T3(c).....	49
Figura 30 - Caminho T1 ideal(vermelho) <i>versus</i> caminho gerado (branco pontilhado). ....	51

Figura 31 - Gráfico de velocidade e orientação do módulo gerado para os caminhos T1(a), T2(b) e T3(c). ..... 51

Figura 32 - Resultado de execução das trajetórias T1(a), T2(b) e T3(c). Acima, a velocidade do módulo e abaixo a sua orientação..... 53

## LISTA DE TABELAS

Tabela 1 – Artigos utilizados para o embasamento do capítulo Estado da Arte. ....	27
Tabela 2 – Componentes eletrônicos necessários para a construção do protótipo.....	33
Tabela 3 - Características físicas do protótipo.....	41
Tabela 4 - Tabela de trajetória gerada pelo sistema. ....	46
Tabela 5 - Exatidão medida para cada sistema.....	47
Tabela 6 - Ganhos de PID encontrados para o módulo protótipo. ....	48
Tabela 7 - Comparação entre o tempo calculado e executado. ....	55

## LISTA DE SIGLAS

FRC	<i>FIRST Robotics Competition</i>
FIRST	<i>For Inspiration and Recognition of Science and Technology</i>
CLP	Controlador lógico programável
CAD	<i>Computer Aided Design</i>
IDE	<i>Integrated Development Enviroment</i>
NHTSA	<i>National Highway Traffic Safety Administration</i>
EUA	Estados Unidos da América
STEM	<i>Science Technology Engineering and Mathematics</i>
PID	Proporcional Integral Derivativo

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
1.1. OBJETIVOS GERAIS E ESPECÍFICOS .....	12
1.2. JUSTIFICATIVA .....	13
1.3. ESTRUTURA DO TEXTO .....	13
<b>2. REFERENCIAL TEÓRICO</b> .....	<b>14</b>
2.1. CONCEITOS BÁSICOS .....	14
<b>2.1.1. AMR – <i>Autonomous mobile robot</i></b> .....	<b>14</b>
<b>2.1.2. Movimento omnidirecional através de um sistema de tração <i>Swerve</i></b> .....	<b>15</b>
<b>2.1.3. Perfil de movimento</b> .....	<b>16</b>
2.2. TRABALHOS CORRELATOS .....	18
<b>2.2.1. Planejamento de Caminho</b> .....	<b>18</b>
<b>2.2.2. Análise Cinemática</b> .....	<b>21</b>
<b>2.2.3. Construção de Trajetória</b> .....	<b>24</b>
<b>2.2.4. Controle de trajetória</b> .....	<b>26</b>
<b>2.2.5. Tabela de Artigos</b> .....	<b>27</b>
<b>3. METODOLOGIA</b> .....	<b>29</b>
3.1. FUNCIONAMENTO E APLICAÇÃO DO SISTEMA .....	29
3.2. SISTEMA PROPOSTO .....	31
<b>3.2.1. Desenvolvimento do protótipo</b> .....	<b>32</b>
<b>3.2.2. Sistema de Controle</b> .....	<b>34</b>
3.2.2.1. Gerador de trajetória .....	35
3.2.2.2. Malha de controle .....	36
3.3. MÉTODO DE VALIDAÇÃO .....	37
<b>4. PROJETO</b> .....	<b>39</b>
4.1. DESIGN E PRODUÇÃO DE UM MÓDULO PROTÓTIPO .....	39
4.2. CONSTRUÇÃO DO GERADOR DE TRAJETÓRIA .....	43
<b>4.2.1. Construção do caminho utilizando pontos de interesse</b> .....	<b>44</b>
<b>4.2.2. Conversão do caminho para a melhor trajetória de tração</b> .....	<b>45</b>
<b>4.2.3. Conversão do caminho para melhor trajetória de rotação</b> .....	<b>46</b>
4.3. CONSTRUÇÃO DO SISTEMA DE CONTROLE.....	47
<b>5. ANÁLISE DE RESULTADOS</b> .....	<b>49</b>
5.1. ANÁLISE DO GERADOR DE TRAJETÓRIA .....	49

	10
5.2. ANÁLISE DO SISTEMA DE CONTROLE.....	53
<b>6. CONCLUSÕES.....</b>	<b>56</b>
<b>REFERÊNCIAS.....</b>	<b>58</b>

## 1. INTRODUÇÃO

O mundo está passando por uma nova transformação tecnológica, onde cada vez mais tarefas mundanas e repetitivas estão sendo desenvolvidas por sistemas robóticos automatizados. Um grande exemplo disso pode ser visto atualmente na indústria automotiva, onde a gradual redução nos preços das tecnologias associadas permitiu que funcionalidades como direção assistida por computador se tornassem relativamente comuns em modelos recentes. Além disso, agências reguladoras, como a NHTSA (*National Highway Traffic Safety Administration*) dos EUA, já estão trabalhando em estruturas legais para regulamentar o uso de veículos completamente autônomos, tecnologia que já é realidade em cidades como San Francisco.

Porém, algumas das aplicações mais importantes destes sistemas autônomos estão na indústria, onde sistemas de transportes automatizados são comuns desde que Henry Ford implementou a primeira esteira em uma linha de montagem em 1914. Onde hoje os AMRs, robôs móveis autônomos, já são facilmente encontrados em pavilhões de empresas logísticas e armazenamento. Diferentemente de um AGV (*Autonomous Guided Vehicle*), um AMR (*Autonomous Mobile Robot*) não necessita de indicadores de navegação externa, ele utiliza sensores e processamento *on-board* para realizar rotinas autônomas. Essa característica tem favorecido a sua implementação em ambientes onde a flexibilidade e adaptabilidade do veículo é importante. (BANKER, 2020).

Muitos destes AMRs são robôs com capacidade de movimentação omnidirecional, isso significa que eles podem se deslocar instantaneamente para qualquer sentido, o que permite a execução de manobras complexas em espaços apertados. Porém, embora valiosa, essa habilidade usualmente requer sistemas de controle mais complexos que os métodos de locomoção comuns.

Uma outra aplicação inusitada desta tecnologia é na educação, onde a robótica educacional utiliza estes sistemas para ensinar conceitos de STEM para estudantes de diversas faixas etárias. Com isso, competições de robótica como a *FIRST Robotics Competition* prometem desenvolver as habilidades para a indústria através do rigor da competição. O impacto destes programas já está sendo observado no mercado de trabalho, o estudo realizado por Melchior *et al.* (2018) mostra que alunos engajados com programas da *FIRST* tem 30% mais probabilidade de estudar em áreas de STEM.

Este trabalho busca desenvolver um sistema capaz de gerar o perfil de movimento para um robô autônomo omnidirecional, com base em uma série de pontos de interesse definidos por um usuário. A intenção é que ele seja posteriormente utilizado em competições da FRC,

onde a aplicação de técnicas como essa permitem que estudantes de ensino médio entrem em contato com tecnologias de ponta da indústria, com o precedente de adquirir uma vantagem competitiva.

Isso está atrelado ao desenvolvimento de dois *softwares* e um protótipo. O primeiro programa é o gerador de trajetória, responsável por registrar os pontos de interesse determinados pelo usuário, realizar uma interpolação polinomial para determinar o caminho e calcular todos os pontos da trajetória através de um modelo cinemático do protótipo. O segundo é a malha de controle que irá realizar a leitura da trajetória desenvolvida pelo gerador e aplicá-la nos controladores de cada atuador no sistema, garantindo que o robô siga a trajetória estabelecida.

Um protótipo foi projetado e construído com um sistema de tração omnidirecional do tipo *Swerve*. A construção mecânica foi feita principalmente de peças usinadas e impressão 3D, enquanto na parte eletrônica foram utilizados os mesmos motores, controladores e sensores comumente utilizados na FRC.

O processo de validação do sistema foi dividido em duas etapas, a primeira visa avaliar qualitativamente o resultado do gerador de trajetória. Enquanto a segunda foi realizada através de uma bateria de testes, onde as trajetórias geradas foram executadas no protótipo através da malha de controle e os resultados foram medidos e averiguados.

## 1.1. OBJETIVOS GERAIS E ESPECÍFICOS

Desenvolver um sistema onde um usuário possa demarcar pontos de interesse em um mapa pré-estabelecido e ele irá gerar uma trajetória otimizada para um robô omnidirecional autônomo do tipo *Swerve* que passe por todos esses pontos. Assim como o desenvolvimento do sistema de controle e protótipos necessários para seguir essa trajetória.

De forma mais específica, os objetivos deste projeto são:

- a) avaliar o melhor método de interpolação para a construção de uma rota dada as condições estabelecidas pelo projeto;
- b) desenvolver uma interface gráfica para utilização do sistema;
- c) desenvolver uma aplicação para gerar trajetórias;
- d) modelar o sistema de controle necessário para que o robô possa seguir a trajetória
- e) projetar e simular um robô omnidirecional do tipo *Swerve*;
- f) avaliar desempenho e precisão do sistema, considerando sua aplicação na FRC;

## 1.2. JUSTIFICATIVA

O estudo realizado por Melchior *et al.* (2018) demonstra que o engajamento em competições de robótica, como a FRC, aumenta significativamente a probabilidade de um estudante do ensino médio se interessar por áreas de STEM, além de melhorar o seu desempenho em outras áreas do conhecimento. Um dos principais motivos para isso é a capacidade do programa de colocar o estudante em contato com ferramentas de engenharia sendo utilizadas na prática, esse contato permite uma visualização concreta e prática dos conceitos que são ensinados em sala de aula. A ferramenta desenvolvida neste projeto também foi utilizada para esse mesmo propósito, demonstrando como conceitos físicos e matemáticos podem resolver problemas complexos do mundo real.

O resultado deste projeto também é relevante para a indústria, uma vez que a implementação de AMRs em linhas de operação não requer a instalação de trilhos ou guias visuais, como nos AGVs. Além disso, eles são muito mais flexíveis e conseguem se adaptar melhor em ambientes que estão em constante mudança. A principal vantagem do AMR está justamente em sua capacidade de processamento e sensoriamento do local por onde está percorrendo, porém, esta vantagem só se concretiza quando o AMR possui uma implementação de desenvolvimento de trajetória eficiente.

## 1.3. ESTRUTURA DO TEXTO

Após esta introdução é apresentado no capítulo 2 um referencial teórico, contendo os conceitos básicos para compreensão do trabalho e uma análise de trabalhos correlatos a este. No capítulo 3 é apresentada a metodologia utilizada para concluir os objetivos propostos, nela é descrita a aplicação e funcionamento do sistema a ser desenvolvido, bem como o método utilizado para sua validação. Em seguida, no capítulo 4, é apresentado o produto elaborado, desde o processo de design e produção do protótipo até o desenvolvimento dos softwares de construção de trajetória e controle.

O capítulo 5 traz a análise detalhada das diferentes etapas do trabalho, a partir dos testes descritos na metodologia. O capítulo 6 finaliza este trabalho com as conclusões do autor e sugestões de trabalhos futuros e melhorias.

## 2. REFERENCIAL TEÓRICO

A seguir serão apresentadas as ferramentas e conceitos utilizados na confecção deste projeto. Elas estão divididas entre os conceitos básicos necessários para o entendimento deste trabalho e o estado da arte das tecnologias que serão apresentadas no mesmo.

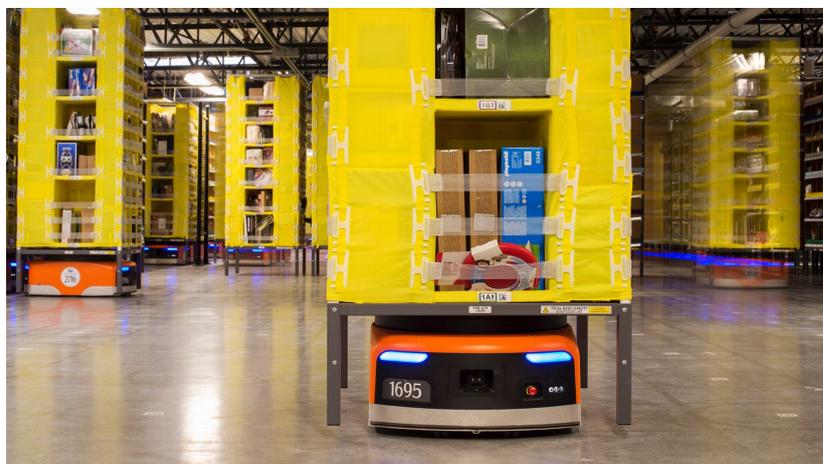
### 2.1. CONCEITOS BÁSICOS

Os tópicos apresentados a seguir buscam contextualizar a relação destes conceitos com este trabalho, expondo uma breve descrição deles. O tema central são as práticas e tecnologias utilizadas para o desenvolvimento de um AMR com capacidade de movimento omnidirecional. Também serão abordados conceitos importantes relacionados a criação e controle de um perfil de movimento para este tipo de robô.

#### 2.1.1. AMR – *Autonomous mobile robot*

O AMR é um sistema robótico automatizado, utilizado principalmente para transporte de cargas e produtos, conforme o exemplo apresentado na Figura 1. Ele possui sensores e processamento embarcado, isso permite que o robô possa realizar estes movimentos de forma autônoma, sem depender de métodos externos de localização, como uma linha no chão. Por isso AMRs são bastante flexíveis, permitindo mudanças de trajetória apenas com comandos via *software*, sem necessitar alterações ao espaço físico onde o robô opera.

Figura 1 – AMRs sendo utilizados para o transporte de produtos em um pavilhão.



Fonte: Bhattacharya (2016).

A Figura 1 mostra um exemplo deste tipo de robô sendo utilizado para transporte dentro de um pavilhão da Amazon. O sistema desenvolvido neste projeto permite o controle de trajetória de robôs como esses, porém com capacidade de movimento omnidirecional.

### 2.1.2. Movimento omnidirecional através de um sistema de tração *Swerve*

Existem diversas topologias diferentes para desenvolver um sistema de locomoção omnidirecional, porém todas possuem algumas características em comum, as principais sendo que um robô equipado com este sistema consegue se deslocar instantaneamente em qualquer direção e consegue realizar movimentos de translação e rotação simultaneamente. A topologia explorada e utilizada no protótipo deste trabalho é comumente conhecida como *Swerve*, ela é composta por 4 módulos de tração, onde cada módulo possui um motor para rotação e outro motor para tracionar a roda. A Figura 2 traz um exemplo deste tipo de módulo.

Figura 2 – Exemplo de um módulo de tração *Swerve*.



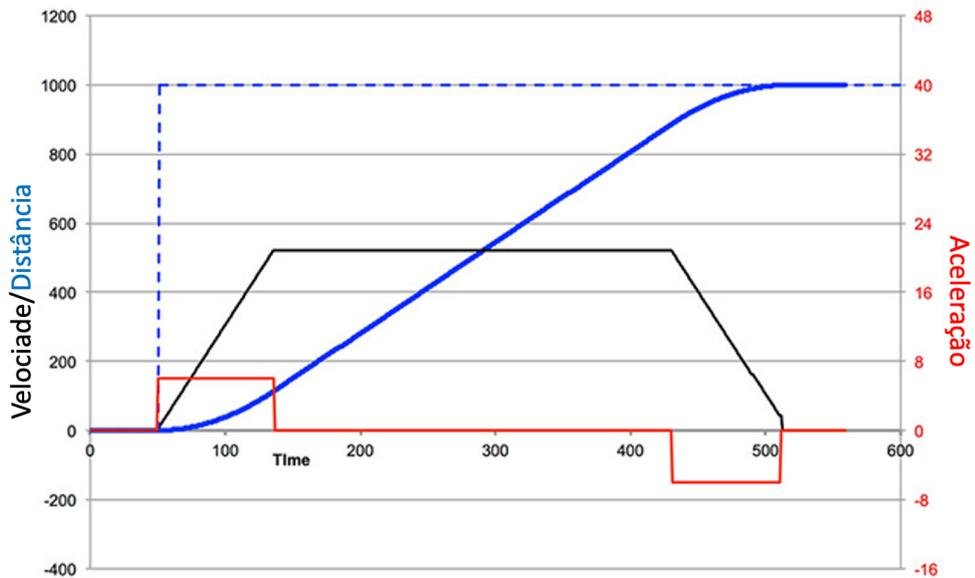
Fonte: ANDYMARK (2020).

Como cada um dos motores presentes nos módulos são controlados de forma independente, eles permitem que o robô execute manobras complexas em espaços apertados. Como durante o movimento a roda está sempre apontada para a direção em que o robô está se deslocando, o arrasto lateral é mínimo e o escorregamento mecânico é reduzido significativamente. Essa característica em conjunto com um sistema de controle otimizado permite que robôs atinjam níveis de precisão e repetibilidade superiores a outros tipos de sistema de deslocamento omnidirecional.

### 2.1.3. Perfil de movimento

O Perfil de movimento é o nome dado ao conjunto de informações que descreve como um motor deve se comportar durante um certo movimento, normalmente em termos de posição, aceleração e velocidade. Esse controle granular do motor permite que o estresse sobre o sistema mecânico seja reduzido significativamente, melhorando a precisão e repetibilidade do sistema. (COLLINS, 2019).

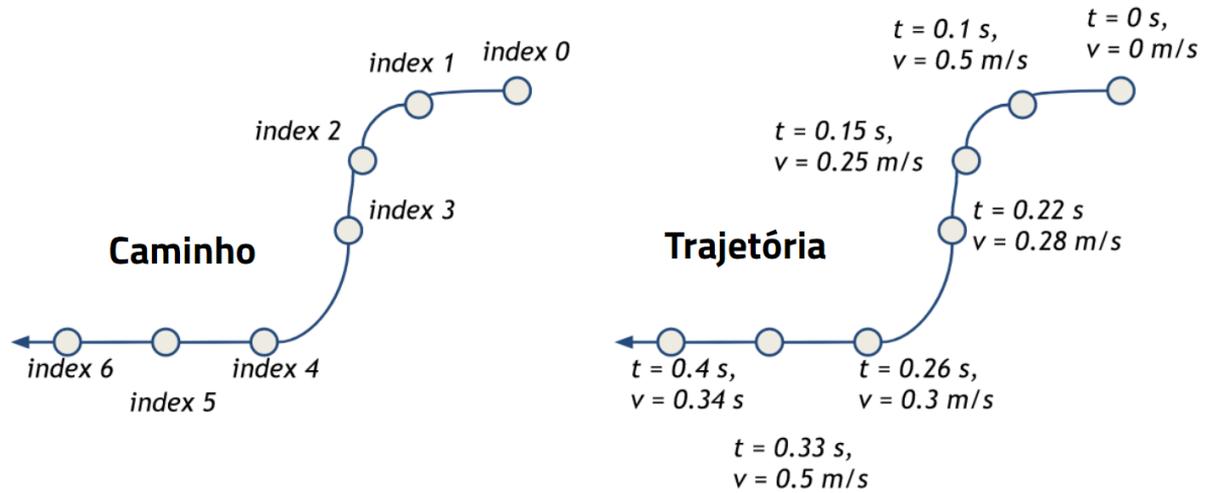
Figura 3 – Exemplo de um perfil de movimento trapezoidal.



Fonte: Collins (2019).

O formato deste perfil pode variar de acordo com a aplicação e nível de precisão necessários, a Figura 3 mostra um exemplo de um perfil trapezoidal para um 1D. Este mesmo conceito pode ser aplicado para um movimento 2D como é o caso deste trabalho, porém neste caso a análise precisa ser separada em duas etapas distintas: Encontrar um caminho e gerar uma trajetória. O caminho é a rota física, ou sequência de coordenadas que, no caso deste projeto, um robô precisa percorrer para atingir o seu objetivo. Enquanto a trajetória coloca o caminho no domínio do tempo, adicionando os valores de velocidade, aceleração e posição. A Figura 4 exemplifica a diferença entre um caminho e uma trajetória.

Figura 4 – Diferenças entre um caminho e uma trajetória.



Fonte: Elaborado pelo autor.

O perfil de movimento para 2D já requer que análises mais complexas sejam feitas, estabelecer um caminho não é trivial, uma vez que é necessário levar em consideração os limites físicos do robô que está sendo modelado. Enquanto a trajetória requer um equacionamento através de uma análise cinemática do robô para que possa ser construída. Ambos estes temas serão abordados ao longo deste projeto, particularmente na próxima seção, onde será apresentado o estado da arte desta aplicação.

## 2.2. TRABALHOS CORRELATOS

Neste capítulo é realizada uma análise e discussão a respeito de outros artigos ou monografias recentemente publicados nas áreas de interesse para este trabalho, com o objetivo de identificar o atual estado da arte das técnicas e tecnologias que foram utilizadas ao transcorrer deste projeto. Através destas análises foi possível identificar quais abordagens utilizadas pelos autores dos artigos seriam mais efetivas durante a elaboração deste projeto, assim como justificar a utilização, ou não, das técnicas propostas por eles. Os artigos analisados foram separados em capítulos conforme a relação deles com este trabalho.

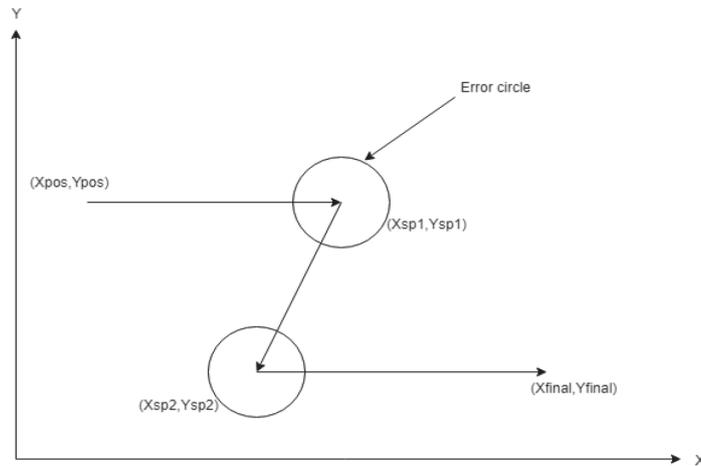
### 2.2.1. Planejamento de Caminho

Problemas relacionados a planejamento de caminho sempre foram relevantes dentro do campo da robótica móvel, por isso já existem diversas técnicas que visam realizar este processo de forma otimizada dadas certas condições. A função do planejamento de caminho é determinar uma rota física otimizada que transcorra por determinados pontos, levando em consideração os limites físicos do robô e do ambiente, mas sem necessariamente determinar o seu perfil de velocidade e aceleração durante o movimento. Ou seja, o caminho resultante não contém informações no domínio do tempo, apenas coordenadas espaciais.

Nesta seção serão analisadas diferentes metodologias propostas para planejamento de caminho, com o intuito de encontrar aquela que melhor permitirá traduzir a informação de caminho resultante para um sistema mecânico, realizando o trajeto mais curto possível. É considerado que as coordenadas de início, fim e pontos de interesse ao longo da trajetória serão definidas previamente. Assim como não serão considerados obstáculos no caminho do robô.

O método de planejamento de caminho proposto por Chaudhari *et al.* (2019) busca determinar o menor caminho que um robô omnidirecional com 3 rodas possa percorrer, enquanto mantém sua orientação constante e transcorre uma série de coordenadas pré-estabelecidas. Este método foi chamado de *Discrete point path planning* e consiste em conectar os pontos pré-estabelecidos com segmentos de retas, onde cada segmento do caminho é tratado separadamente. Com isso o caminho resultante acaba não contendo curvas contínuas, e sim diversos vértices onde a mudança de direção é instantânea. A Figura 5 representa um exemplo de caminho resultante deste método, com base em 3 pontos de interesse.

Figura 5 – Caminho gerado pelo algoritmo de *Discrete point path planning*.



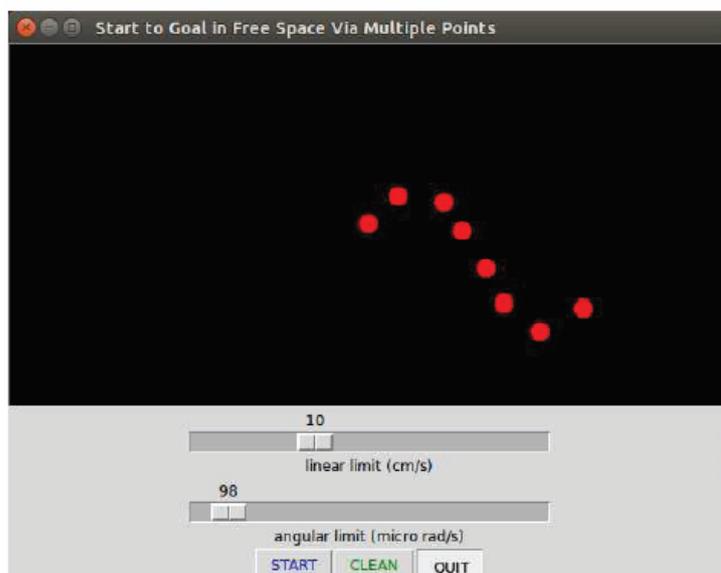
Fonte: Chaudhari *et al.* (2019).

Como também pode ser observado na Figura 5, foi adicionado um círculo de erro ao redor da coordenada objetivo. De forma que o raio dele representa a tolerância do sistema, sendo que quando o robô atinge este círculo ele imediatamente começa a se deslocar para a próxima coordenada. (CHAUDHARI *et al.* 2019).

Conforme observado pelos autores, a utilização deste método implica que o robô precisa reduzir significativamente sua velocidade, ou parar por completo, em cada vértice do caminho. Isso acontece porque o caminho gerado possui mudanças de direção instantâneas, algo que não é possível traduzir para um movimento contínuo em sistemas mecânicos, principalmente devido a inércia e limitações mecânicas do próprio robô.

Outra perspectiva utilizada para planejamento de caminho é explorada por Yin *et al.* (2015), onde os autores buscam definir o perfil de movimento necessário para o controle de um robô omnidirecional. Previamente já estão definidas as coordenadas cartesianas de início  $(x_i, y_i, \theta_i)$  e fim  $(x_f, y_f, \theta_f)$ , assim como quaisquer outros pontos de interesse que se deseja transcorrer ao longo do caminho realizado pelo robô. Porém considera-se que devem ocorrer apenas variações suaves de velocidade e aceleração durante este movimento. Assim, para evitar bruscas alterações de velocidade e aceleração no robô, é necessário gerar um caminho que não possua mudanças de direção instantâneas. O primeiro passo é a especificação pelo usuário das coordenadas de interesse em uma interface gráfica, conforme apresentado na Figura 6. Em seguida utiliza-se uma função Spline com interpolação polinomial cúbica segmentada entre as coordenadas definidas, assim é possível obter um caminho com curvas suaves e contínuas que passe por todas elas.

Figura 6 – Interface gráfica para definição das coordenadas de interesse.

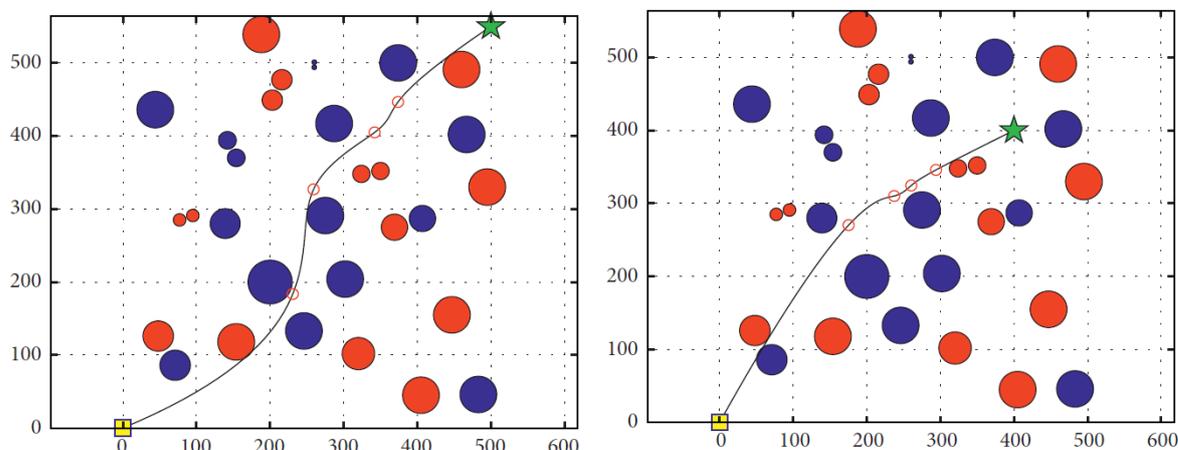


Fonte: Yin *et al.* (2015).

Esta mesma função é utilizada no artigo de Lian *et al.* (2020), onde é avaliado um algoritmo para identificação da rota mais curta que um robô autônomo pode percorrer através de um campo com obstáculos. As principais coordenadas do caminho são definidas através de um algoritmo de aprendizagem de máquina, e em seguida é aplicada a função *Spline* com interpolação polinomial cúbica segmentada para realizar a conexão destes pontos de forma suave e formular o caminho completo que o robô deve percorrer.

A Figura 7 demonstra dois resultados da aplicação deste sistema de planejamento de caminho em um determinado ambiente. Nesta imagem os círculos vermelhos vazios ao longo do trajeto representam as coordenadas identificadas como ótimas pelo algoritmo de aprendizagem de máquina, os círculos pintados de vermelho e azul são obstáculos e a linha preta é a trajetória interpolada utilizando a função *Spline*.

Figura 7 – Resultados do sistema de planejamento de caminho com função *Spline*.



Fonte: Lian *et al.* (2020).

É possível observar que com a aplicação desta técnica o perfil de movimento do robô passa a ser suave e contínuo, o que permite que ele seja traduzido para um sistema físico de forma mais efetiva, reduzindo o desgaste mecânico. Ela também permite, através de seu equacionamento, que a orientação do robô seja definida no início, fim e durante os pontos de interesse ao longo do percurso.

Um dos percalços do método apresentado por Lian *et al.* (2020) é que a interpolação por função polinomial cúbica não resulta em curvas de arcos constantes, isso significa que o raio de uma curva pode variar durante o movimento do robô. Esse efeito pode ser prejudicial, pois implica em mudanças bruscas na taxa de aceleração do sistema (arranque), o que pode causar instabilidades no sistema mecânico.

Zhang *et al.* (2015) propõe resolver este problema utilizando um gerador de caminhos baseado em uma curva de Bezier de quinta ordem, aplicada em um robô com sistema de tração diferencial independente. Além de resolver o problema apresentado acima, esta técnica também permite que o limite físico de aceleração do robô seja incluso no equacionamento do caminho, evitando que ele seja excedido durante a movimentação do robô.

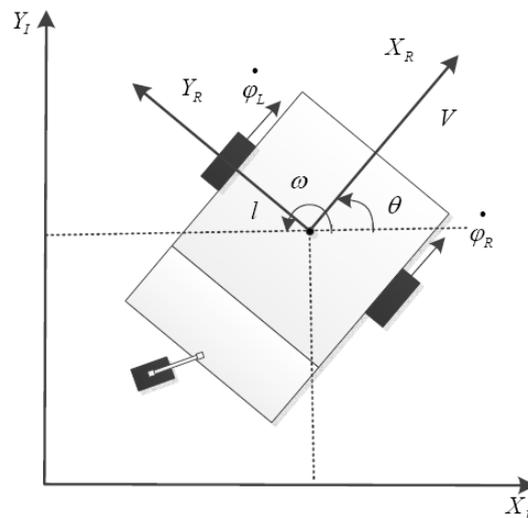
### 2.2.2. Análise Cinemática

Uma vez que o caminho a ser realizado pelo robô é estabelecido, é necessário a construção de um modelo cinemático que possa descrever o sistema mecânico a ser controlado. Esse modelo irá descrever a dinâmica do robô, permitindo que as restrições físicas do sistema sejam identificadas, como sua velocidade e aceleração máximas em função do tipo de

movimento a ser realizado. O resultado é utilizado em conjunto com o caminho a ser percorrido, de forma a estabelecer a trajetória completa do sistema. Em seguida, serão apresentadas diferentes abordagens utilizadas para a modelagem cinemática de sistemas similares ao proposto neste trabalho.

Com o objetivo de construir o perfil de velocidade ao longo do caminho encontrado por eles Zhang *et al.* (2015) descrevem a construção de um modelo cinemático para um robô diferencial independente com duas rodas tracionadas, conforme a Figura 8, que estabelece uma relação entre a velocidade de translação ( $V$ ) e rotação ( $\omega$ ) do robô com a velocidade de cada roda ( $\varphi_L$  e  $\varphi_R$ ).

Figura 8 – Estrutura do robô modelado.



Fonte: Zhang *et al.* (2015).

Através da análise cinemática, os autores também equacionam restrições que precisam ser seguidas durante a construção da trajetória de velocidade, são elas:

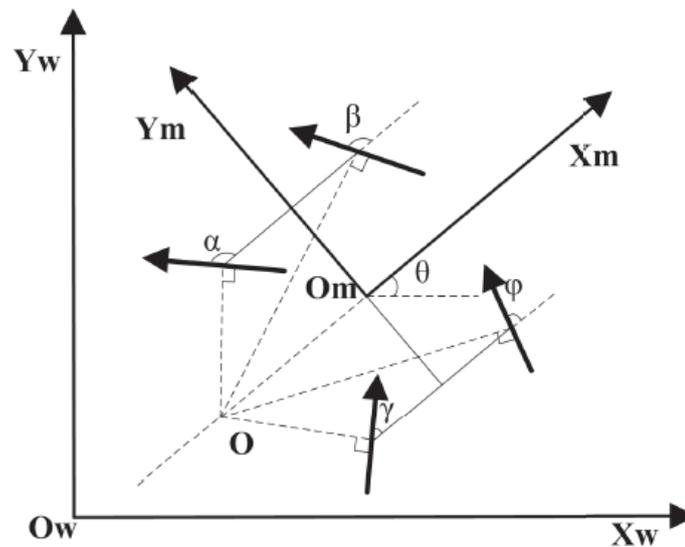
- a) restrições de velocidade de translação;
- b) restrições de velocidade de rotação;
- c) restrições de velocidade das rodas;
- d) restrições de aceleração.

Embora o sistema apresentado por Zhang *et al.* (2015) possua características mecânicas distintas daquelas do sistema proposto neste trabalho, ele é um ótimo exemplo de como o processo de análise cinemática deve ser realizado. Delimitando as restrições físicas impostas

sobre o sistema, de forma a estabelecer um equacionamento que permita a criação de uma trajetória que respeite estas restrições.

Um sistema similar ao proposto neste trabalho é analisado por Sun *et al.* (2018), que explora, de forma bastante extensa, a cinemática de diferentes manobras que podem ser realizadas por um robô omnidirecional do tipo *Swerve*. Um resultado interessante obtido pelos autores é que, ao percorrer um caminho onde o raio de curvatura é constante, o robô pode utilizar uma manobra de Ackerman dupla, conforme a Figura 9 onde  $\beta$ ,  $\alpha$ ,  $\varphi$  e  $\gamma$  representam a orientação de cada roda do sistema. Essa manobra mantém o sistema de direção em um ângulo fixo para todas as rodas, o que reduz a complexidade do controle do sistema.

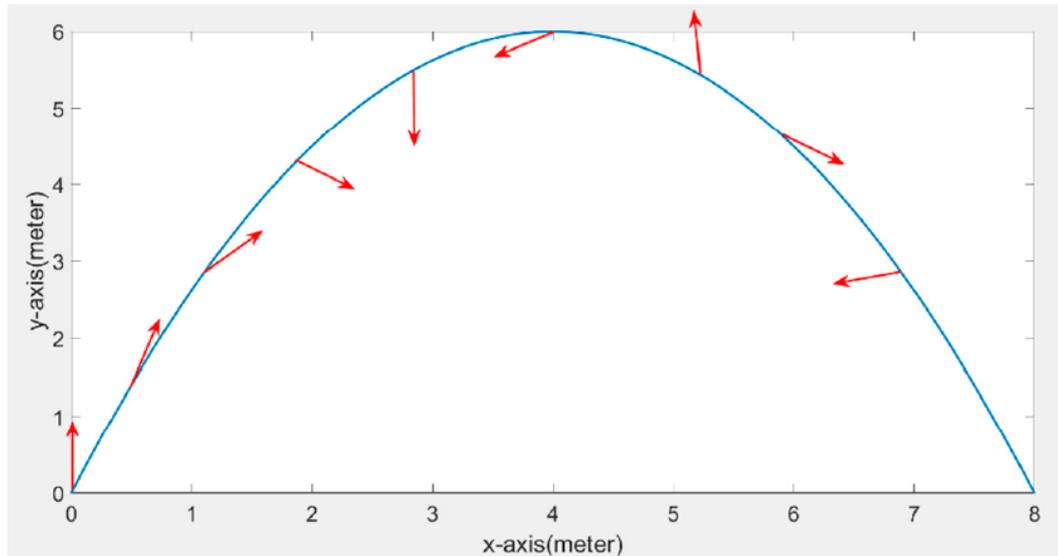
Figura 9 – Representação da manobra Ackerman dupla.



Fonte: Sun *et al.* (2018).

Conforme a análise realizada por Sun *et al.* (2018) esta manobra se torna mais complexa em caminhos onde o robô precisa percorrer curvas de raio não constante, como senóides ou elipses. Nesse caso o robô precisaria realizar ajustes constantes no ângulo de direção das rodas para que elas se mantenham tangentes ao raio da curva sendo percorrida.

Figura 10 – Trajetória com movimentos de rotação e translação simultâneos.



Fonte: Jia *et al.* (2019).

Jia *et al.* (2019) também apresenta uma extensa análise cinemática em um robô com movimentação omnidirecional do tipo *Swerve*, porém com um foco em otimizar a utilização de torque e energia durante movimentações. Esta análise permite que os autores desenvolvam um sistema capaz de utilizar todo o potencial omnidirecional do robô, conforme apresentado na trajetória presente na Figura 10 onde são realizados movimentos de translação e rotação simultaneamente.

### 2.2.3. Construção de Trajetória

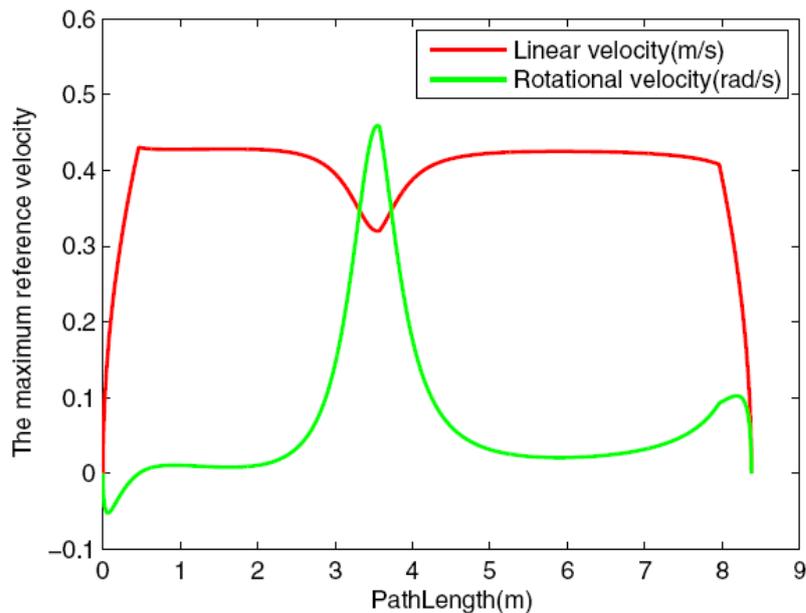
Para criar uma trajetória, é necessário adicionar informações de velocidade, tempo e aceleração em um caminho. Assim, se faz necessário a aplicação ponto-a-ponto de um algoritmo que leve em consideração as restrições estabelecidas através de uma análise cinemática. Essa técnica também é conhecida como a aplicação de um perfil de movimento, pois o resultado é o histórico completo do movimento que deve ser realizado pelo robô em um intervalo de tempo. Conforme Zhang *et al.* (2015), planejamento de trajetória é uma habilidade básica para um sistema robótico autônomo, que determina possíveis movimentos que podem levar o veículo de um estado inicial para um objetivo, dado seus modelos cinemáticos associados.

O método para a criação de um perfil de movimento aplicado por Zhang *et al.* (2015) parte do princípio de que o robô deve executar todos os movimentos da forma mais rápida

possível, por isso estas restrições são calculadas antes deste processo. O autor divide o algoritmo de construção de trajetória em três passos distintos:

- a) a velocidade de translação máxima é calculada para cada segmento do trajeto, levando em consideração como os raios das curvas afetam estes limites ao longo do trajeto;
- b) o algoritmo itera do início para o fim do trajeto, aplicando as restrições de aceleração e desaceleração máximas, além de considerar os limites de velocidade de translação entre pontos vizinhos. Evitando bruscas mudanças de velocidade;
- c) o processo realizado na etapa b) ocorre novamente, porém desta vez o algoritmo itera do fim para o início do trajeto.

Figura 11 – Exemplo de um perfil de velocidade resultante do algoritmo proposto.



Fonte: Zhang *et al.* (2015).

Depois da execução destes passos, o perfil de velocidade já deve atender a todas as restrições estabelecidas sobre o sistema, a Figura 11 demonstra um exemplo deste resultado, onde estão explicitas as velocidades linear e rotacional ao longo do trajeto. É importante ressaltar que as restrições de velocidade de translação indicadas no item a) são características de um robô com sistema de tração diferencial independente, como o proposto pelos autores do artigo.

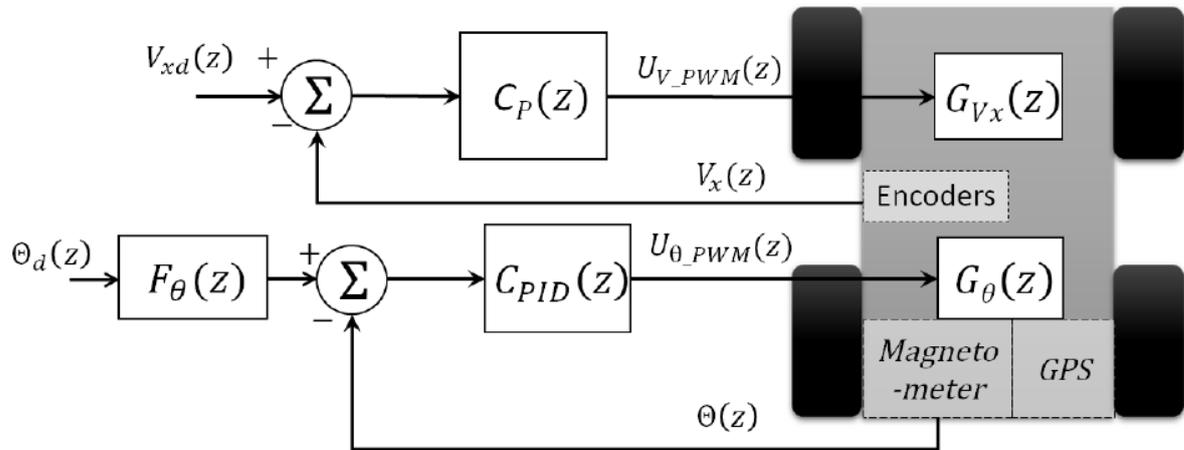
#### 2.2.4. Controle de trajetória

Com o estabelecimento de uma trajetória adequada, o último passo para a execução de um movimento autônomo é modelar a malha de controle que irá garantir que o robô percorra essa trajetória da forma mais adequada possível. Algoritmos para controle de velocidade e aceleração em malha fechada já são constantemente utilizados em aplicações industriais, além de serem facilmente encontrados em sistemas robotizados. Este é um campo de pesquisa bastante explorado e em constante evolução, uma vez que sistemas automatizados estão cada vez mais presentes na indústria e no mercado. Neste capítulo serão exploradas aplicações recentes desta técnica para o controle de trajetória em robôs autônomos.

O artigo publicado por Zhu *et al.* (2018) apresenta o desenvolvimento de um sistema robótico autônomo, capaz de detectar obstáculos em seu caminho e recalculá-la sua trajetória em tempo real. O robô utilizado nesta análise possui é similar a um carro de passeio, com atuações separadas para as rodas de tração e direção. A malha de controle proposta utiliza 3 termos, *feedforward*( $\delta f$ ), Proporcional( $\delta p$ ) e integral( $\delta i$ ) para regular o ângulo de curvatura das rodas de acordo com a trajetória. Os termos  $\delta p$  e  $\delta i$  representam a intensidade da correção que precisa ser aplicada aos motores a partir do feedback de sensores acoplados ao robô. Enquanto o termo  $\delta f$  é determinado apenas através da cinemática do robô e da informação de trajetória já calculada previamente, praticamente operando em malha aberta.

Essa mesma fusão entre malhas de *feedforward* e *feedback* é apresentada por Barrero *et al.* (2018) para controlar um robô com tração diferencial independente ao longo de uma trajetória calculada previamente. Como pode ser observado na Figura 12 a proposta dos autores utiliza malhas de controle distintas para o controle de velocidade ( $V_x$ ) e direção( $\Theta$ ). A malha de velocidade é bastante simples, utilizando apenas o feedback de sensores do robô junto a um controlador proporcional. Enquanto a malha para controle de direção utiliza um conjunto entre um controlador PID operando com o feedback de sensores, e um compensador *feedforward* que trabalha em malha aberta.

Figura 12 – Diagrama da malha de controle utilizada pelo autor.



Fonte: Barrero *et al.* (2018).

De acordo com os autores, a utilização deste compensador em *feedforward* junto a malha de controle PID tradicional, é importante para reduzir o erro em regime permanente do sistema. Isso ocorre porque o compensador permite que o controlador PID possa trabalhar apenas como um ajuste fino, uma vez que com o cálculo de trajetória e a análise da cinemática do sistema realizados previamente, a determinação do ganho do compensador se torna trivial.

### 2.2.5. Tabela de Artigos

Os artigos utilizados para realizar a análise do estado da arte das tecnologias relacionadas a este trabalho são apresentados na Tabela 1, eles estão classificados conforme o tema que abordam e a ordem que aparecem no texto.

Tabela 1 – Artigos utilizados para o embasamento do capítulo Estado da Arte.

Artigo	Autores	Ano de publicação	Relação com o trabalho
<i>Trajectory planning for an indoor mobile robot using quintic Bezier curves</i>	Zhang <i>et al.</i>	2015	Planejamento de caminho Análise cinemática Construção de trajetória
<i>Motion Planning Implemented in ROS for Omni-directional Wheeled Mobile Robot</i>	Yin <i>et al.</i>	2015	Planejamento de caminho
<i>Path Planning and Controlling of Omni-Directional Robot Using Cartesian Odometry and PID Algorithm</i>	Chaudhari <i>et al.</i>	2019	Planejamento de caminho

<i>Cubic Spline Interpolation-Based Robot Path Planning Using a Chaotic Adaptive Particle Swarm Optimization Algorithm</i>	Lian <i>et al.</i>	2020	Planejamento de caminho
<i>Analysis on Kinematics of Omnidirectional Full Drive Mobile Robot</i>	Sun <i>et al.</i>	2018	Análise cinemática
<i>Energy-Efficient Torque Distribution Optimization for an Omnidirectional Mobile Robot with Powered Caster Wheels</i>	Jia <i>et al.</i>	2019	Análise cinemática
<i>Online Quintic Path Planning of Minimum Curvature Variation with Application in Collision Avoidance</i>	Zhu <i>et al.</i>	2018	Controle de trajetória
<i>Outdoors Trajectory Tracking Control for a Four-Wheel Skid-Steering Vehicle</i>	Barrero <i>et al.</i>	2018	Controle de trajetória

Fonte: Elaborado pelo autor.

A interface gráfica apresentada por Yin *et al.*(2015) que permite o usuário definir uma série de pontos de interesse será utilizada como inspiração para a definição da interface do trabalho a ser desenvolvido. O estudo de algoritmos de planejamento de caminho realizado no capítulo 2.2.1 demonstra que a utilização de um algoritmo de interpolação polinomial de terceira ordem satisfaz os requisitos para a execução deste trabalho.

A metodologia para construção de uma trajetória trapezoidal de velocidade, conforme proposta por Zhang *et al.* (2015), será utilizada como base para o equacionamento do gerador de trajetória deste projeto. Assim como a aplicação de uma malha de controle com ganhos de *feedforward* e *feedback* proposta por Barrero *et al.* (2018) será utilizada na malha de controle deste projeto, caso seja necessário.

### 3. METODOLOGIA

Os conceitos apresentados até agora contextualizam as diferentes abordagens que já foram utilizadas para desenvolver sistemas similares ao proposto neste projeto, onde foi possível averiguar as vantagens e desvantagens de cada aplicação. Este estudo servirá como guia para determinar a estrutura metodológica deste trabalho, e será utilizado como base para decidir entre a utilização de diferentes algoritmos a partir das observações e resultados dos autores citados nos capítulos anteriores.

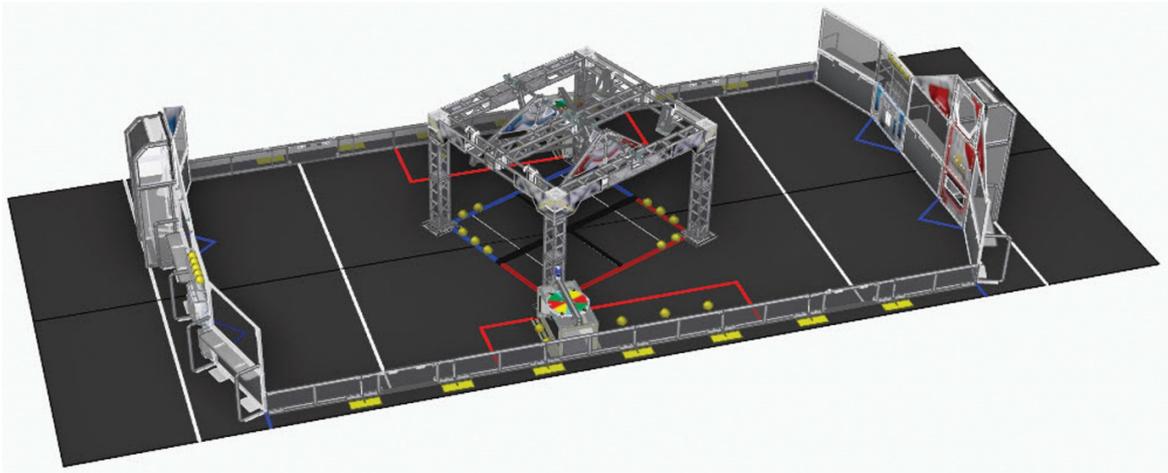
No seguinte capítulo será primeiramente apresentado como e onde o resultado deste projeto será aplicado. Seguido de uma descrição do protótipo mecânico necessário para o seu desenvolvimento, assim como a estrutura de algoritmos e ferramentas de *Software* que esperam-se ser utilizados. Por último será apresentado o processo a ser utilizado na validação do sistema proposto e a sequencias de testes que serão realizados para explorar a funcionalidade do sistema.

#### 3.1. FUNCIONAMENTO E APLICAÇÃO DO SISTEMA

O resultado esperado para este projeto é um sistema que permita um usuário a facilmente programar um robô autônomo omnidirecional para realizar trajetórias complexas. Essa programação seria realizada apenas marcando pontos de interesse sobre um mapa previamente conhecido, o sistema então seria responsável por gerar e controlar uma trajetória otimizada que atravessasse todos esses pontos, respeitando os limites físicos do robô em questão.

O propósito para a criação deste sistema é a sua aplicação em competições de FRC, particularmente durante o período de movimentação autônoma. Durante este curto período é esperado que os robôs se movimentem em uma quadra de 16 m x 8 m e executem diversas tarefas pré-determinadas com velocidade e exatidão. Além de que, a repetibilidade destes movimentos precisa ser excelente, pois é esperado que cada robô execute estas tarefas em até 30 partidas por campeonato. A Figura 13 mostra um exemplo de quadra na FRC, como estas quadras e as regras de jogo alteram anualmente, o foco deste sistema será apenas em realizar a movimentação do robô pela quadra, e não em controlar os mecanismos específicos para interagir com ela.

Figura 13 – Quadra de jogo para o desafio da FRC de 2020, *Infinite Recharge*.



Fonte: FIRST (2020).

Como o tempo do período autônomo é significativamente limitado, normalmente sendo apenas 15 segundos, os movimentos realizados pelo robô durante este período precisam ser otimizados para velocidade, permitindo que mais tarefas possam ser realizadas. Como uma partida de FRC é jogada em uma configuração de 3 robôs contra 3 robôs, é comum que cada um assuma um papel estratégico específico durante o jogo. Porém, como a composição destas equipes é aleatória, é essencial que a trajetória percorrida pelo robô possa ser rapidamente alterada ou modificada, dependendo da estratégia adotada em cada partida. Por isso, uma das características do sistema a ser desenvolvido neste projeto é que ele deve permitir rápido ajuste e reprogramação das trajetórias pelo usuário.

Figura 14 – Robô de FRC para o desafio de 2015(esquerda) e 2016(direita).



Fonte: Elaborado pelo autor.

Conforme pode ser observado na Figura 14 a construção mecânica de robôs de FRC altera significativamente entre diferentes desafios. Isso significa que, para garantir longevidade ao sistema proposto, é preciso que ele seja facilmente reconfigurado a partir das características físicas e cinemáticas do robô que será programado.

### 3.2. SISTEMA PROPOSTO

Considerando a aplicação deste sistema, descrita na seção anterior, e buscando simplificar o desenvolvimento do algoritmo, algumas características da trajetória foram assumidas de antemão, são elas:

- a) o usuário definiu pelo menos 2 pontos de interesse por trajetória, assim como a orientação desejada do robô em cada ponto;
- b) o robô inicia o percurso em repouso;
- c) o robô deve parar no último ponto da trajetória especificada;
- d) não há obstáculos entre os pontos de interesse definidos pelo usuário;
- e) todas as medidas da quadra ou ambiente onde se encontra o robô, são conhecidas;

A partir destas considerações, serão desenvolvidos dois *softwares*, uma descrição mais detalhada deles está na seção 3.2.2. O primeiro consistirá em uma interface gráfica, onde o usuário irá mapear os pontos de interesse e realizar a configuração das características físicas do robô, com estas informações a aplicação deverá gerar uma tabela as informações da trajetória completa já otimizada. Essa aplicação deve ser executada em um computador comum. Enquanto a segunda aplicação será executada no CLP do robô, essa deve executar a sua malha de controle utilizando a tabela gerada na primeira aplicação. Para o teste e validação deste sistema, será simulado um protótipo de um robô omnidirecional do tipo *Swerve*.

Figura 15 – Diagrama descrevendo a sequência de desenvolvimento do projeto.



Fonte: Elaborado pelo autor.

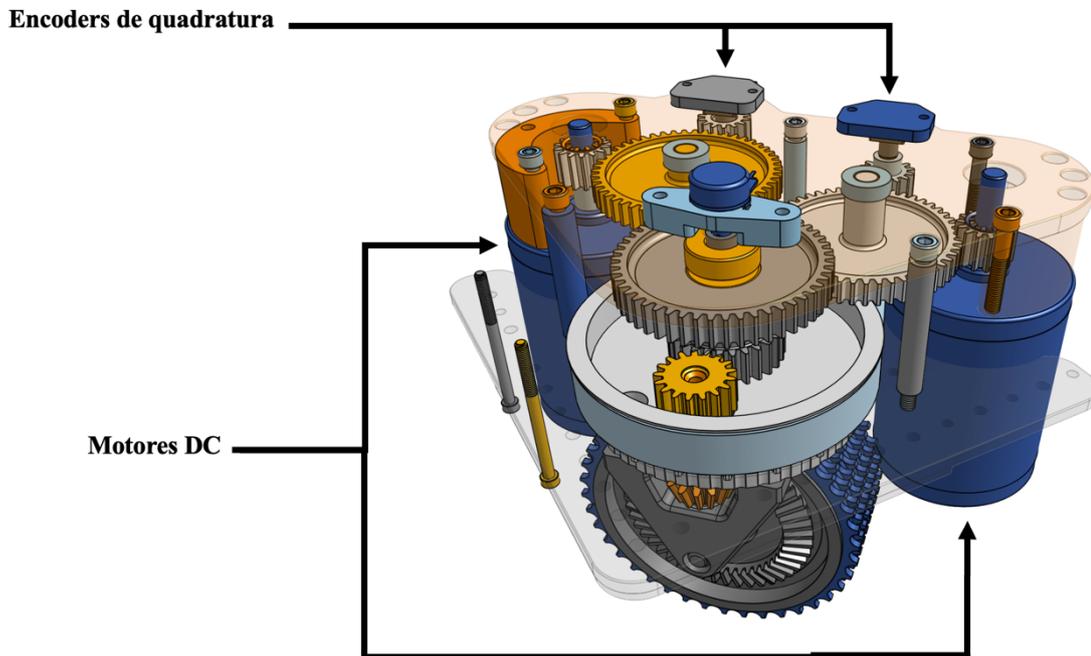
O diagrama na Figura 15 descreve o processo de desenvolvimento esperado para este projeto, junto com uma linha do tempo simplificada. Esse processo foi separado em seis etapas distintas, iniciando pelo estudo do estado da arte das tecnologias que serão utilizadas no projeto, o que permite que possa ser construída uma estrutura com de projeto a partir das ferramentas e algoritmos que precisam ser utilizados. Posteriormente será realizado a modelagem e fabricação do protótipo para realização de testes do sistema. As últimas três etapas consistem na escrita dos dois *softwares* necessários, assim como o processo de testes e validação destes sistemas. Ambos os *softwares* podem ser alterados e modificados do seu estado original dependendo do resultado dos testes executados.

### 3.2.1. Desenvolvimento do protótipo

É necessária a construção de um protótipo físico para a validação deste sistema porque uma das premissas adotadas para este projeto é verificar se os algoritmos utilizados conseguem minimizar o impacto de imperfeições mecânicas do sistema sobre a precisão da movimentação do robô. O protótipo a ser desenvolvido será um módulo de movimentação para um robô omnidirecional do tipo *Swerve* que possui:

- a) um motor DC escovado para tracionar a roda;
- b) um motor DC escovado para o controle de direção da roda;
- c) uma roda de 4 polegadas de diâmetro;
- d) um encoder de quadratura acoplado ao eixo de rotação da roda, para efetuar medidas de velocidade;
- e) um encoder de quadratura acoplado ao eixo de rotação do módulo, para efetuar medidas de direção do movimento.

O diagrama na Figura 16 ilustra a utilização destes componentes no módulo. Essa configuração, embora complexa de controlar, habilita o robô a realizar uma enorme variedade de manobras e movimentos sem sacrificar aderência ao chão, o que é um problema recorrente com outras formas de locomoção omnidirecional. O protótipo será completamente impresso em 3D, de forma apenas a servir como uma referência para modelagem e testes do sistema, não como uma aplicação final. Todo o projeto mecânico será modelado utilizando a ferramenta de CAD *Onshape*.

Figura 16 – Diagrama do módulo *Swerve*.

Fonte: Elaborado pelo autor.

Além dos sensores e atuadores, também será necessário utilizar outros componentes para garantir o controle eficiente do sistema. Entre eles está o RoboRIO, que atuará como o CLP do robô, os Victors SPX, periféricos responsáveis pelo controle individual dos motores DC e a PDP (*Power Distribution Panel*), que irá regular e distribuir a tensão da bateria entre todos os componentes do sistema. A Tabela 2 resume a lista de componentes eletrônicos que serão necessários para a construção do protótipo.

Tabela 2 – Componentes eletrônicos necessários para a construção do protótipo.

<b>Equipamento</b>	<b>Fabricante</b>	<b>qtd</b>
RoboRIO	National Instruments	1
Power Distribution Panel	CTR Eletronics	1
Victor SPX	CTR Eletronics	2
Mini CIM Motor	VEX Robotics	2
Encoder KY-040	Keyes	2

Fonte: Elaborado pelo autor.

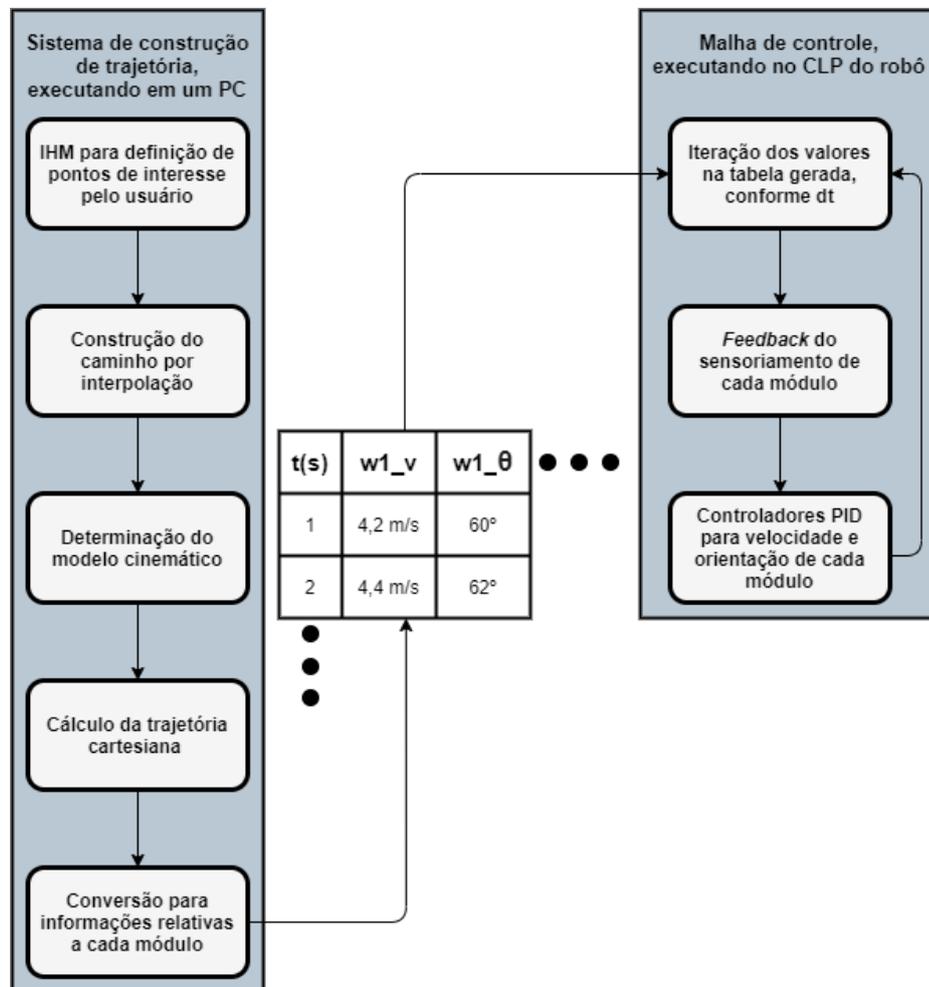
Durante os testes do protótipo deve ser analisado também a posição do acoplamento dos encoders de quadratura dentro da caixa de engrenagens, uma vez que folgas entre os dentes das

engrenagens utilizadas podem causar grandes imprecisões nas medidas realizadas. Caso isso ocorra, existe a possibilidade de modificar a posição dos sensores conforme necessário.

### 3.2.2. Sistema de Controle

O foco deste projeto está no desenvolvimento do *software* de modelagem e controle de trajetória para o protótipo desenvolvido. Ele foi dividido em dois programas diferentes, que devem operar de forma sequencial conforme o diagrama de blocos funcional apresentado na Figura 17.

Figura 17 – Diagrama funcional do sistema de controle



Fonte: Elaborado pelo autor.

Ambos foram desenvolvidos considerando a cinemática do protótipo apresentado anteriormente, porém os programas precisam permitir que o usuário ajuste características físicas do robô para que o modelo gerado seja compatível com ele. Isso permite que este mesmo

programa possa ser utilizado em diferentes plataformas de movimento robotizado, sem a necessidade de reescrever grandes parcelas da aplicação. Algumas dessas características que precisam ser indicadas pelo usuário são:

- a) máxima velocidade de translação;
- b) máxima aceleração;
- c) distância entre módulos;
- d) máxima velocidade de rotação do módulo.

### 3.2.2.1. Gerador de trajetória

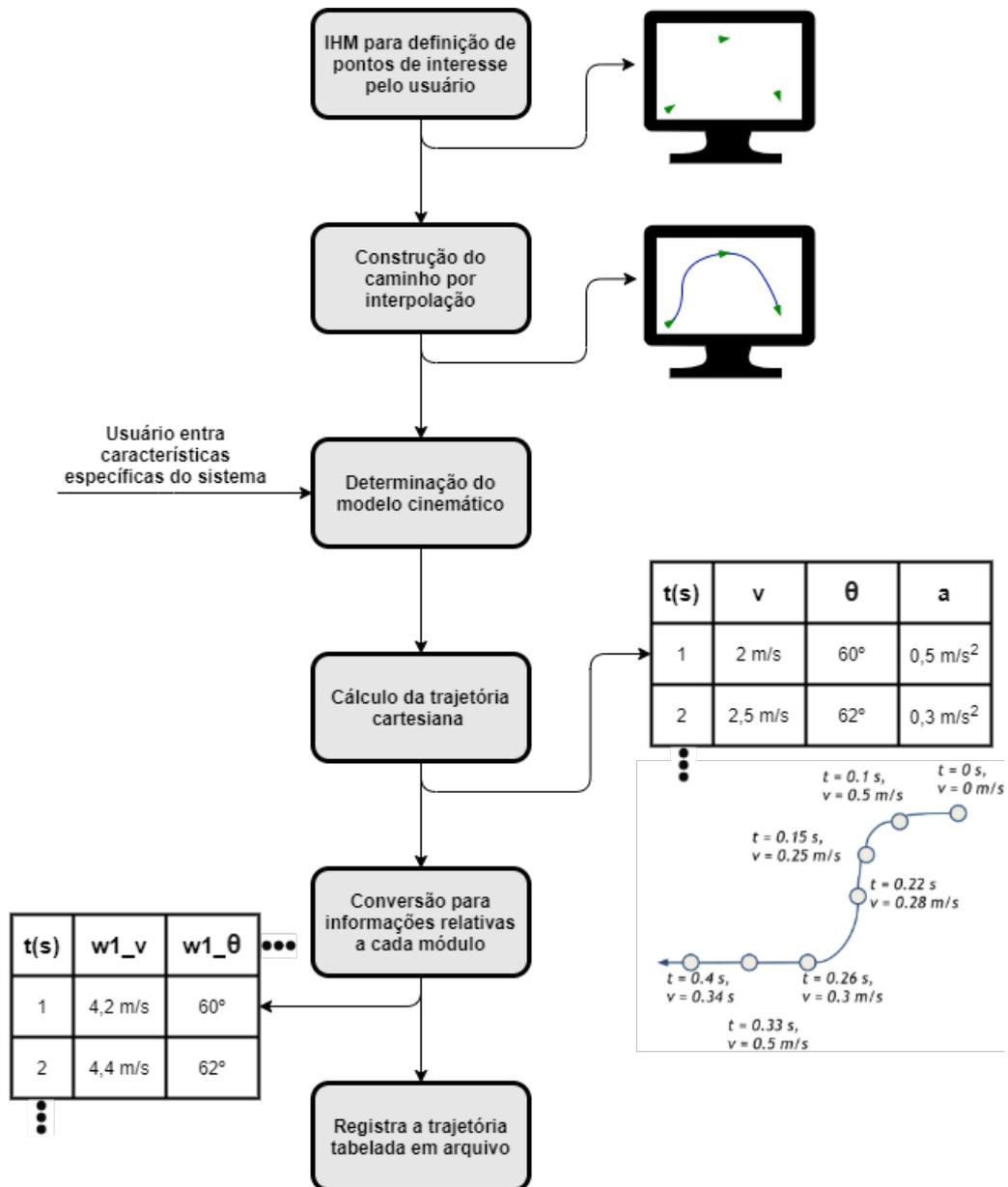
Este programa foi escrito em LabVIEW. A escolha desta linguagem em ambiente de desenvolvimento se dá pela disponibilidade de bibliotecas de interpolação polinomial e de recursos relacionados a construção de trajetória neste ambiente. Essa aplicação é responsável por transformar as entradas do usuário em uma trajetória completa que o protótipo em questão consegue percorrer, a Figura 18 possui um diagrama detalhado de como os dados devem ser tratados a cada etapa do programa.

Para propriamente desenvolver o código necessário no gerador de trajetória é necessário a implementação de uma função de interpolação polinomial, que é utilizada para definir o caminho percorrido pelo robô. Neste caso foi utilizado uma Curva de Bezier de quinta ordem, porque a interpolação resultante desta função é suave e contínua o suficiente para minimizar o estresse sobre o sistema mecânico. Além disso, essa função também vai garantir que a curvatura da trajetória seja constante, situação que é ideal para o sistema omnidirecional tipo *Swerve* porque minimiza a quantidade de movimentos de rotação necessários sobre o módulo e simplifica significativamente o controle dele.

Outra necessidade é a implementação de um equacionamento para a cinemática de um sistema de tração do tipo *Swerve*, esse deve permitir que o algoritmo utilize todo o potencial omnidirecional do protótipo. Alguns exemplos de manobras a serem implementadas são:

- a) movimentos de translação e rotação simultâneos;
- b) movimento de translação enquanto mantem a orientação do robô constante;
- c) manobra de Ackerman dupla para navegar curvas fechadas.

Figura 18 – Diagrama detalhado do gerador de trajetória.



Fonte: Elaborado pelo autor.

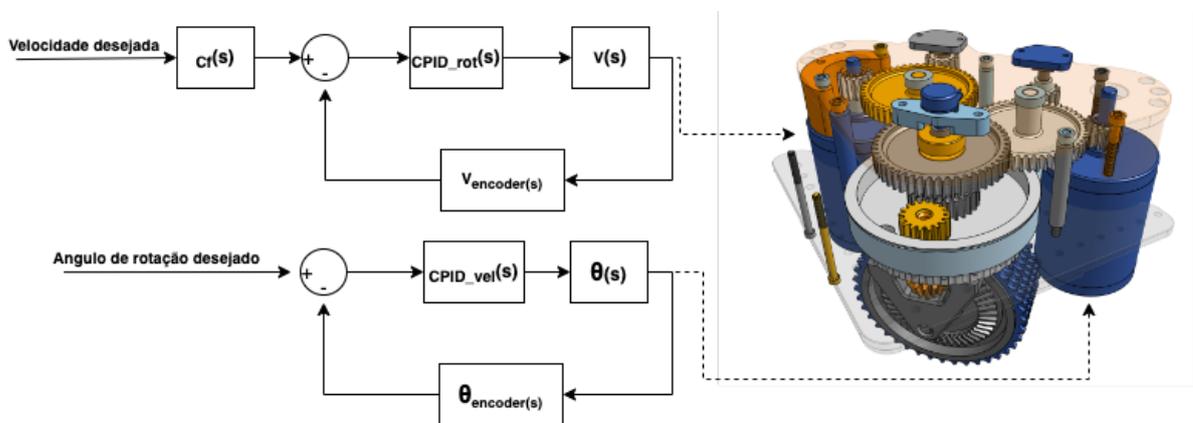
### 3.2.2.2. Malha de controle

Essa aplicação executará diretamente no CLP do robô, o RoboRIO, e será escrita em LabVIEW. Sua principal responsabilidade é transcorrer a tabela criada pelo Gerador de Trajetória e alimentá-la aos controladores individuais de cada módulo. Dessa forma o robô deve percorrer a trajetória gerada da forma mais fiel possível. Como todos os módulos podem ser controlados de forma independente se faz necessária a implementação de controladores

individuais para cada módulo, sendo que cada módulo deve conter um controlador para o motor de rotação( $\theta(s)$ ) e outro para o motor de tração( $V(s)$ ).

Para realizar o controle do movimento de rotação, foi utilizado um controle PID ( $C_{PID\_rot}(s)$ ). Enquanto para o motor de tração foi utilizado um controle PID ( $C_{PID\_vel}(s)$ ) em conjunto com um compensador *feedforward* ( $C_f(s)$ ), para realizar o controle de velocidade de cada roda. A Figura 19 apresenta o diagrama de blocos simplificado para um módulo do protótipo.

Figura 19 – Diagrama de blocos simplificado para a malha de controle.



Fonte: Elaborado pelo autor.

Os ganhos do controlador PID e *feedforward* para o controle de velocidade e posição precisam ser ajustados para o módulo em questão, isto foi feito por base em testes individuais de cada sistema após a finalização do protótipo.

### 3.3. MÉTODO DE VALIDAÇÃO

Nessa seção será descrita a metodologia de avaliação que foi utilizada para determinar se o sistema proposto atinge, ou não, as expectativas do projeto. Os dois principais aspectos a serem avaliados no projeto são relacionados a qualidade da trajetória gerada e eficiência do sistema de controle de realizá-la.

Para avaliar a trajetória gerada, serão propostos 3 diferentes cenários a partir do sistema de pontos de interesse, e para cada cenário será realizada uma avaliação qualitativa do resultado entregue pelo sistema. As questões chaves desta avaliação estão descritas a seguir:

- a) O caminho gerado é o menor possível dado as restrições estabelecidas?
- b) O caminho gerado é suave e contínuo?

- c) A trajetória faz o melhor proveito possível da potência disponível nos motores?
- d) O resultado apresenta algum comportamento não esperado?

A efetividade do sistema de controle foi avaliada através da execução das três trajetórias geradas anteriormente. Antes do início dos testes foram inspecionados a carga da bateria e o sistema mecânico do protótipo, para minimizar o impacto de fatores externos. As trajetórias a serem realizadas serão definidas posteriormente e, a fim de simplificar o projeto, foi construído apenas um módulo de movimentação, que foi avaliado individualmente. Durante estes testes, as seguintes métricas foram registradas:

- a) tempo necessário para executar a trajetória do início ao fim;
- b) histórico de leitura dos sensores do sistema;
- c) posição inicial e final;
- d) vídeo da execução do movimento.

Após a bateria de testes, os dados foram comparados com os valores esperados para cada trajetória e os seguintes indicadores calculados:

- a) erro de deslocamento entre o caminho esperado e o real;
- b) diferença entre o tempo de execução esperado e o real;

A partir destas avaliações é possível determinar se os objetivos do projeto foram alcançados, ou não. Em seguida, será apresentado o desenvolvimento do protótipo e dos dois *softwares* necessários para a realização destas análises.

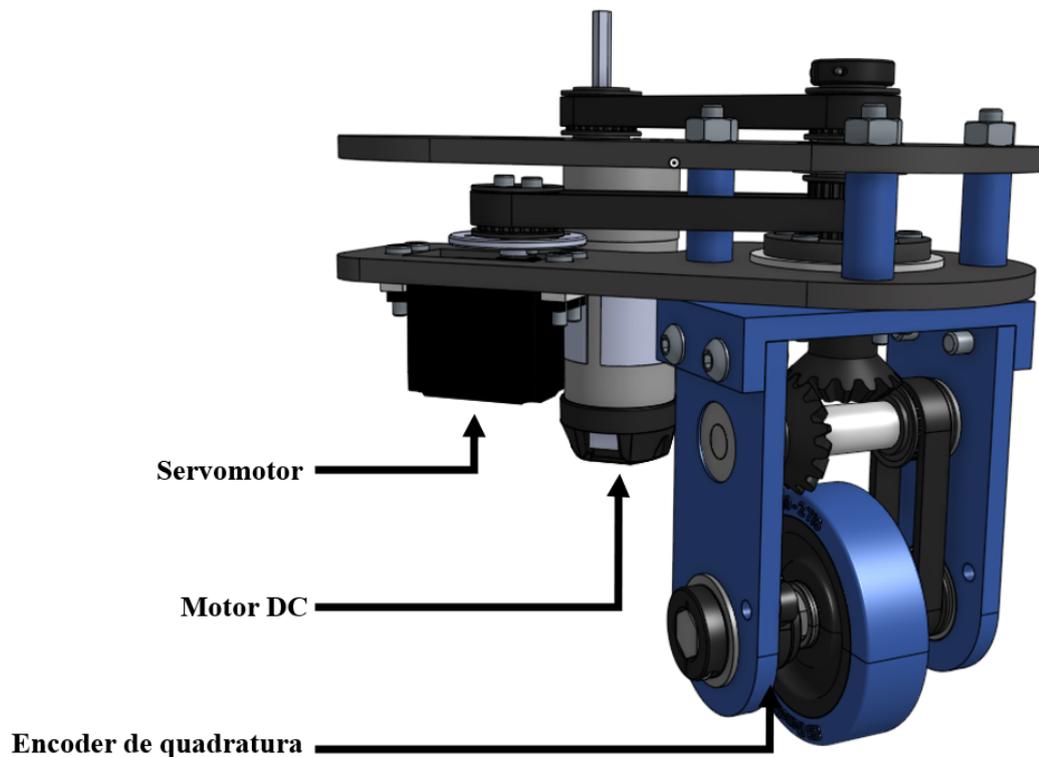
## 4. PROJETO

Em seguida será apresentado o produto resultante do projeto, assim como suas peculiaridades durante seu desenvolvimento.

### 4.1. DESIGN E PRODUÇÃO DE UM MÓDULO PROTÓTIPO

Para a validação apropriada do sistema, foi necessária a elaboração de um projeto em CAD de um módulo *Swerve* contendo as características necessárias para a validação do sistema de trajetórias. Este módulo atuará apenas como um modelo representativo do sistema e não necessariamente precisa atender os rigores de uso contínuo. Por isso, foi projetado para que pudesse ser fabricado apenas utilizando peças de prateleira e impressão 3D. A Figura 20 – Primeiro modelo do módulo *swerve* para prototipagem apresenta o primeiro modelo projetado para implementação neste projeto.

Figura 20 – Primeiro modelo do módulo *swerve* para prototipagem.



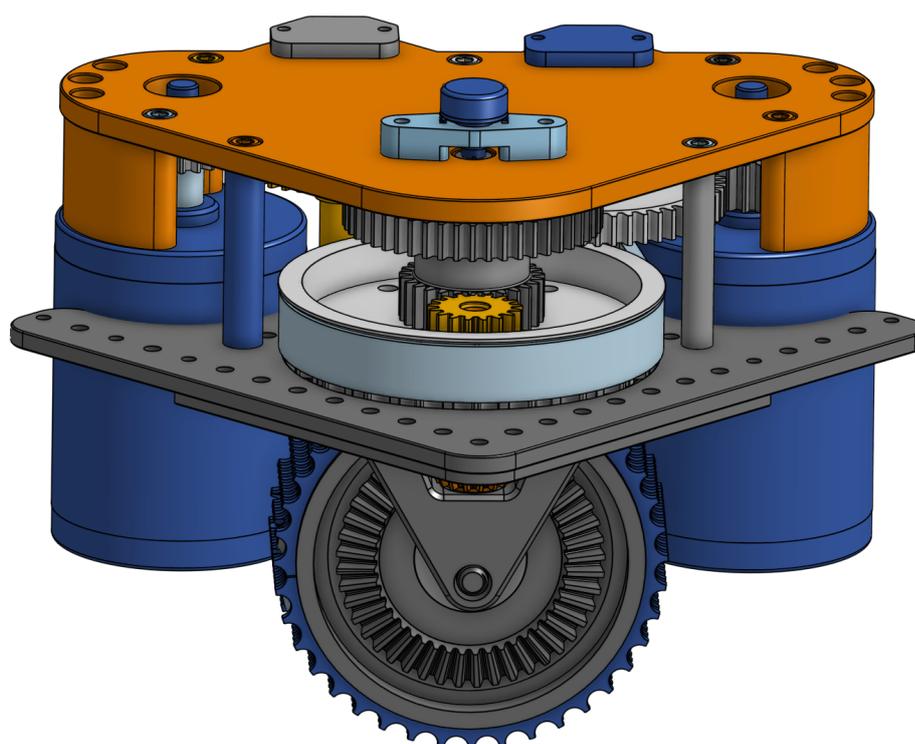
Fonte: Elaborado pelo autor.

O primeiro módulo projetado permite a avaliação do sistema utilizando componentes mais acessíveis, como um servomotor e um motor DC de baixa potência. Isto permite que o

sistema possa ser replicado sem acesso a componentes eletrônicos específicos da FRC, o que viabilizaria a realização deste projeto caso a atual situação da logística internacional não permitisse acesso aos componentes eletrônicos específicos da FRC. Porém, como foi possível adquirir os equipamentos, não houve a necessidade de construí-lo.

Com o intuito de desenvolver um protótipo que fosse o mais próximo possível do equipamento utilizado na FRC, foi projetada uma segunda versão utilizando os atuadores e sensores listados na Tabela 2 – Componentes eletrônicos necessários para a construção do protótipo. Esta versão é apresentada na Figura 21 - Projeto final do módulo construído.. Este novo projeto permite que quase todas suas partes mecânicas sejam impressas em 3D, o que viabiliza sua confecção.

Figura 21 - Projeto final do módulo construído.



Fonte: Elaborado pelo autor.

Este projeto possui algumas características físicas que são importantes para a modelagem do sistema, elas são apresentadas na Tabela 3 - Características físicas do protótipo.. As características de redução serão utilizadas durante o processo de construção do sistema de

controle para converter a leitura do encoder de quadratura para as unidades adequadas, enquanto as características de velocidade limite serão utilizadas na construção do gerador de trajetória.

Tabela 3 - Características físicas do protótipo.

<b>Relação de redução para rotação do módulo</b>	21,47:1
<b>Relação de redução para tração do módulo</b>	6,75:1
<b>Velocidade teórica máxima de tração</b>	174 pol/s
<b>Velocidade teórica máxima de rotação</b>	267 RPM

Fonte: Elaborado pelo autor.

As partes mecânicas do projeto finalizado foram fabricadas utilizando uma impressora 3D *Crealty Ender 3 V2* com o material PLA. O sistema foi posteriormente montado em conjunto com os motores DC e encoders e foi fixado em um suporte que permitia a realização de testes sem que o sistema entrasse em contato com o chão, minimizando os impactos sobre as peças impressas em 3D. A Figura 22 - Módulo Swerve para testes construído. mostra o módulo finalizado, e a Figura 23 - Módulo fixado ao suporte para realização de testes. mostra o módulo fixado no suporte para testes.

Figura 22 - Módulo Swerve para testes construído.



Fonte: Elaborado pelo autor.

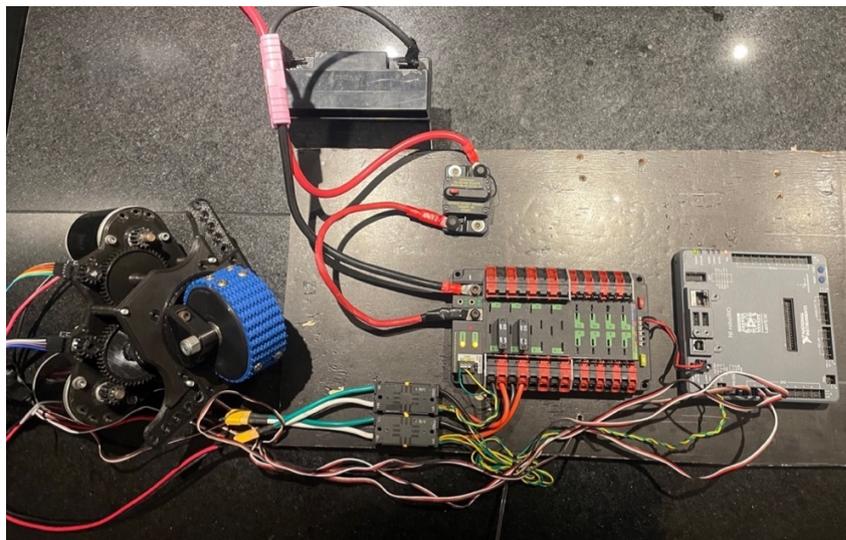
Figura 23 - Módulo fixado ao suporte para realização de testes.



Fonte: Elaborado pelo autor.

Junto ao protótipo foi montada uma tabua com os eletrônicos necessários para controlar o sistema. Estes eletrônicos são os mesmos utilizados na *FIRST Robotics Competition* e estão listados na Tabela 2 – Componentes eletrônicos necessários para a construção do protótipo. a montagem final é apresentada na Figura 24.

Figura 24 - Montagem completa do sistema de prototipagem.



Fonte: Elaborado pelo autor.

O sistema é energizado por uma bateria de 12 V e 18 Ah, isso permite que sejam recriadas as mesmas condições elétricas que em um robô de FRC.

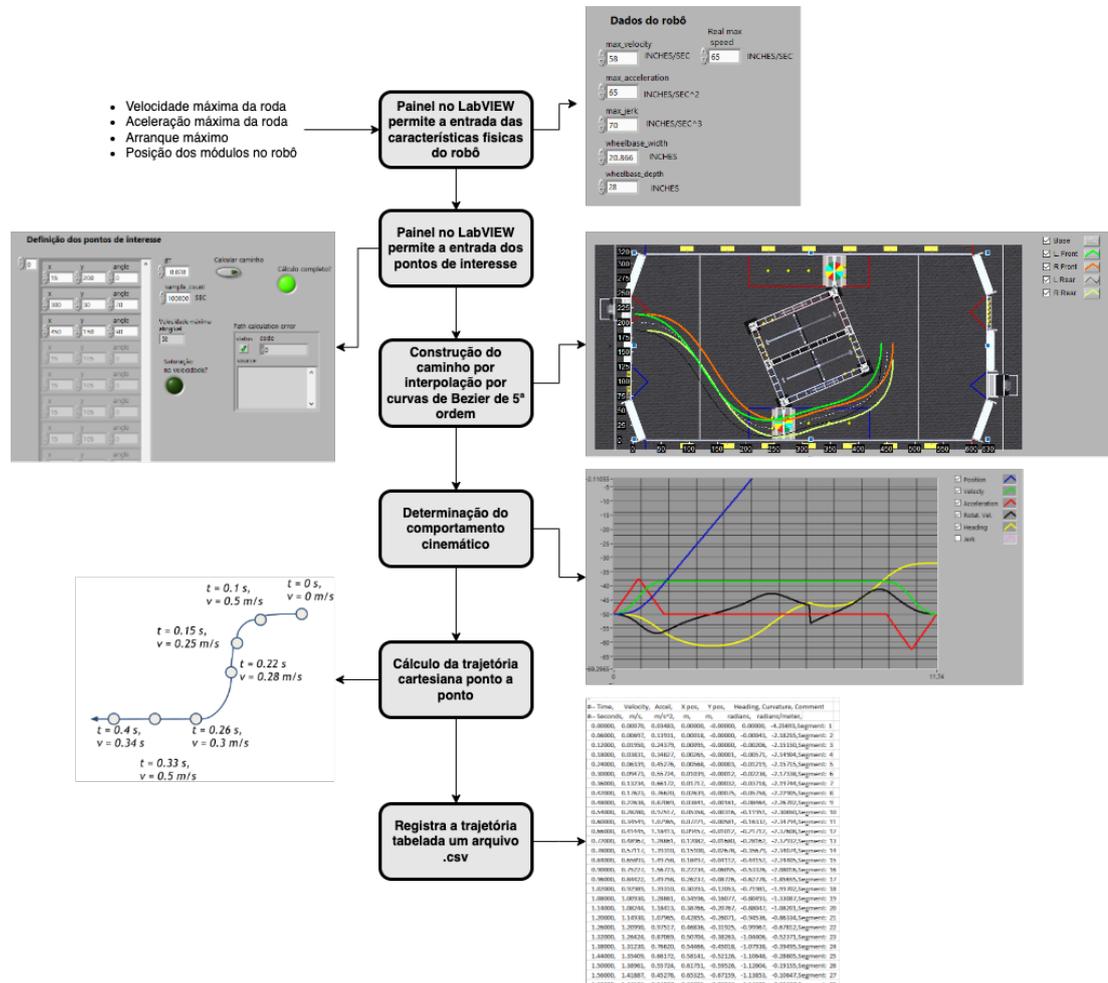
#### 4.2. CONSTRUÇÃO DO GERADOR DE TRAJETÓRIA

A demanda por sistemas capazes de gerar trajetórias dentro do ecossistema da FRC é significativa ao ponto que, tanto o processo de interpolação por pontos de interesse, quanto os cálculos cinemáticos, já possuem diversas bibliotecas validadas para implementação. Em função dos estudos apresentados no capítulo 2.2.3 foi escolhida uma biblioteca que utilizasse interpolação polinomial cúbica para determinar o caminho entre os pontos de interesse.

Este tipo de interpolação não gera trajetórias de raio constante, porém isto não causa problemas em um sistema de movimentação do tipo *Swerve*, uma vez que para realizar uma curva é necessário apenas variar o ângulo de rotação do módulo, e não a velocidade de movimentação da roda, como em uma tração diferencial. Assim, a biblioteca *Pathfinder* foi escolhida por possuir uma documentação extensa e compatibilidade com a linguagem LabVIEW, escolhida para realizar este projeto.

A implementação desta biblioteca em LabVIEW possui uma interface onde é possível inserir os pontos de interesse desejados através de coordenadas cartesianas. Estas coordenadas são então utilizadas para realizar uma interpolação polinomial de terceira ordem, gerando o caminho a ser percorrido pelo sistema. Em seguida, são utilizados os parâmetros físicos do robô para realizar o cálculo de trajetória ponto-a-ponto que é registrada em formato de tabela, para ser interpretada pelo software de controle.

Figura 25 - Diagrama de funcionamento do gerador de trajetória.



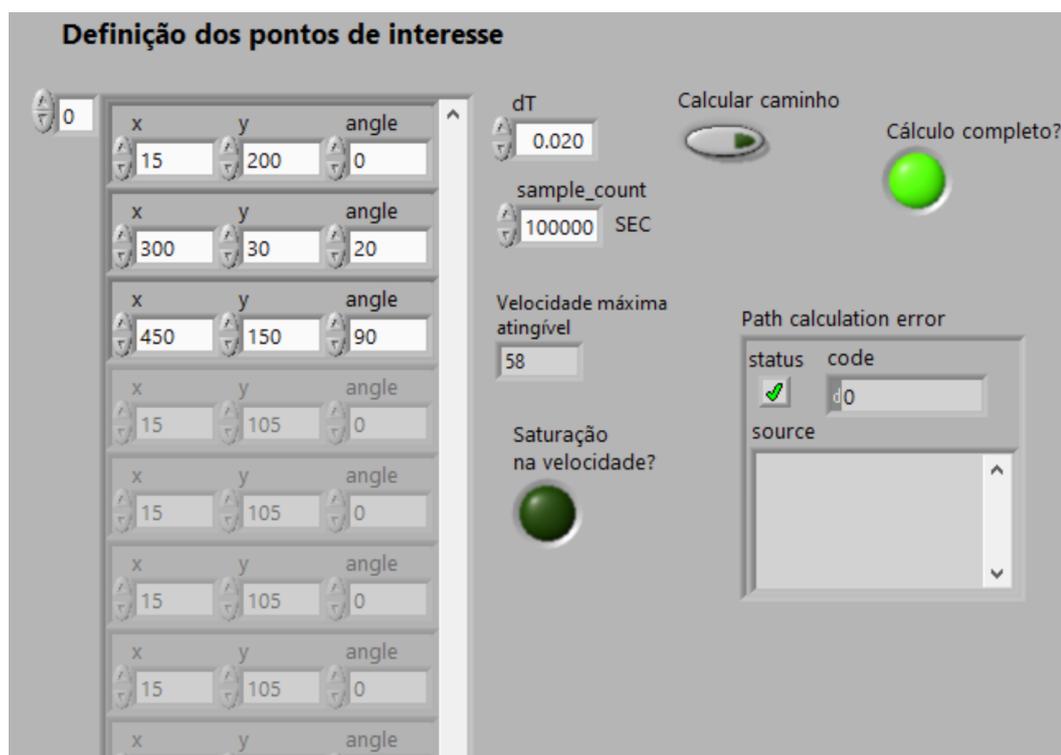
Fonte: Elaborado pelo autor.

A Figura 25 é um diagrama que resume a operação do gerador de trajetória e demonstra sua utilização. Em seguida serão discutidas as diferentes etapas utilizadas pela biblioteca para construir a trajetória estabelecida pelo usuário, assim como as limitações de cada uma.

#### 4.2.1. Construção do caminho utilizando pontos de interesse

O primeiro passo do software durante a construção da trajetória é o cálculo de um caminho utilizando pontos de interesse com base em uma função *spline* interpolação polinomial de terceira ordem, estes pontos de interesse são definidos pelo usuário através de uma tabela de coordenadas cartesianas. Conforme consta na Figura 26, o usuário define uma série de coordenadas, contendo os valores de distância em X e Y, e o ângulo de orientação do robô quando ele estiver atravessando este ponto.

Figura 26 - Interface utilizada para definir os pontos de interesse pelo usuário.



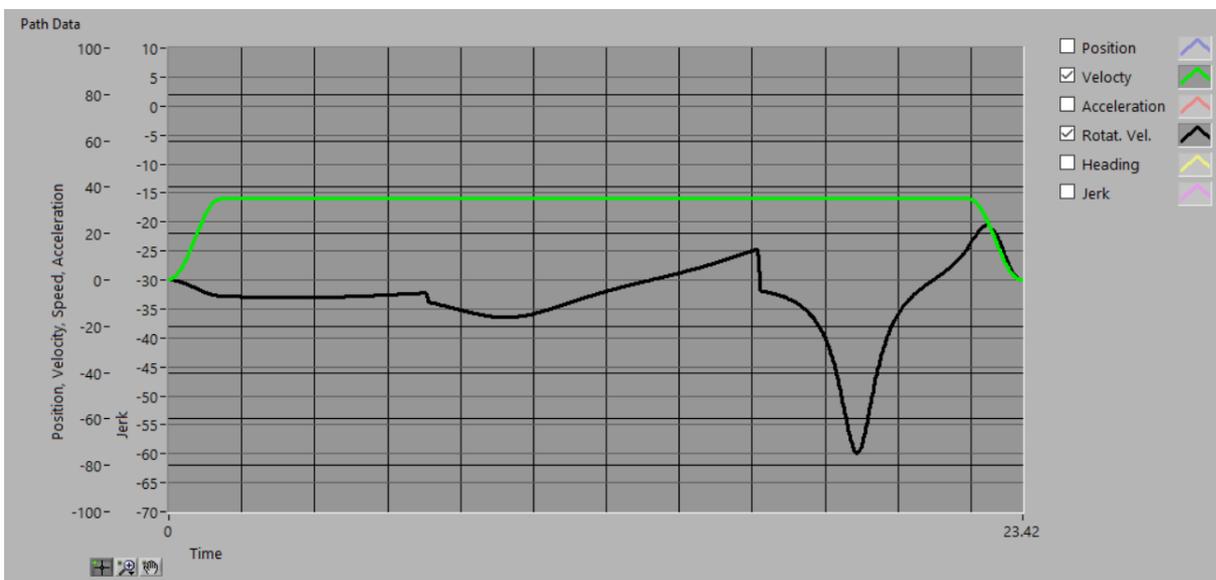
Fonte: Elaborado pelo autor.

O caminho é interpolado e apresentado ao usuário no próximo menu. O *software* exige que, pelo menos, 2 pontos de interesse sejam informados pelo usuário, porém não possui um limite teórico na quantidade de pontos a serem definidos. É possível que o usuário informe uma sequência de pontos de interesse que resultem em um caminho que não é executável, dada as restrições físicas do robô. Neste caso, o software indica esta situação ao usuário. Com o caminho definido, o próximo passo é a construção da trajetória passo-a-passo do módulo *Swerve*, com informações de velocidade e direção de movimento.

#### 4.2.2. Conversão do caminho para a melhor trajetória de tração

O processo de construção de caminho é separado em dois processos, o primeiro é o cálculo de cinemática para a roda de tração do módulo, este cálculo utiliza os parâmetros físicos do sistema, que foram informados pelo usuário, para determinar a velocidade da roda de tração ao longo de toda a trajetória. Para definir a velocidade ao longo do caminho utiliza-se um perfil de movimento trapezoidal, este perfil é calculado utilizando as informações de velocidade e aceleração máxima do sistema.

Figura 27 - Exemplo de trajetória gerada pelo sistema. Em verde, velocidade da roda e em preto, orientação do módulo.



Fonte: Elaborado pelo autor.

A trajetória de velocidade é calculada ponto-a-ponto utilizando-se um passo de tempo determinado pelo usuário, neste projeto será utilizado um passo de 20 ms para os testes realizados. Esta informação é posteriormente tabelada e salva para uso com o sistema de controle do módulo. A Figura 27 demonstra o resultado calculado e apresentado pelo sistema em formato gráfico.

#### 4.2.3. Conversão do caminho para melhor trajetória de rotação

O segundo processo utiliza as coordenadas do caminho gerado para determinar a orientação, em graus, que o robô deve estar em cada ponto da trajetória para segui-la adequadamente. Esta informação é salva juntamente com a informação de velocidade de tração. A Figura 27 também demonstra o resultado da cinemática de rotação do módulo ao longo da trajetória calculada.

Tabela 4 - Tabela de trajetória gerada pelo sistema.

Tempo	Velocidade	Aceleração	X pos	Y pos	Direção
Segundos	m/s	m/s <sup>2</sup>	m	m	radianos
0.00000	0.00070	0.03503	0.00000	0.00000	0.00000
0.06000	0.00701	0.14011	0.00015	-0.00000	-0.00003
0.12000	0.01962	0.24519	0.00092	-0.00000	-0.00012

0.18000	0.03853	0.35028	0.00264	-0.00000	-0.00034
0.24000	0.06375	0.45536	0.00568	-0.00000	-0.00072
0.30000	0.09528	0.56044	0.01042	-0.00001	-0.00132
0.36000	0.13310	0.66552	0.01724	-0.00002	-0.00218
0.42000	0.17724	0.77061	0.02653	-0.00004	-0.00336
0.48000	0.22768	0.87569	0.03865	-0.00009	-0.00489
0.82000	0.54853	0.94574	0.17048	-0.00185	-0.02176
0.98000	0.69565	0.84066	0.27021	-0.00466	-0.03471

Fonte: Elaborado pelo autor.

Em seguida a informação calculada é compilada e salva em formato de tabela, como a apresentada na Tabela 4, que posteriormente é utilizada pelo sistema de controle para que o robô siga a trajetória estabelecida. Uma vez que o sistema utilizado para prototipagem possui apenas um módulo de movimentação, não foram consideradas a realização de manobras que exijam mais de um módulo neste gerador. Assim, os dados tabelados já estão prontos para a utilização com o sistema de controle, sem precisar de tratamento prévio.

#### 4.3. CONSTRUÇÃO DO SISTEMA DE CONTROLE

Com a trajetória finalizada, o próximo passo é o desenvolvimento de um software que executará no CLP RoboRIO e é responsável por realizar a leitura da tabela criada previamente, utilizando-a para efetivar o controle do sistema ao longo da trajetória. Para a criação deste software utilizou-se novamente o LabVIEW. Onde primeiramente foi construído um programa de teste para validar o funcionamento do módulo e calibrar a leitura dos sensores, para os testes sejam realizados nas unidades de medida adequadas.

Para esta calibração, foram-se acionados os motores de tração e movimento até que ambos os sistemas visualmente completassem 10 voltas. O valor obtido foi então utilizado para determinar a exatidão de cada um dos sistemas. A Tabela 5 expressa os resultados obtidos durante os testes realizados.

Tabela 5 - Exatidão medida para cada sistema.

<b>Sistema</b>	<b>Precisão em termos da unidade de medida utilizada</b>
Tração	0,12 pol
Rotação	1,125 graus

Fonte: Elaborado pelo autor.

Como o módulo de teste foi totalmente fabricado utilizando impressão 3D, não é seguro rodar o sistema de tração em sua velocidade máxima de 175 pol/s. Por isso, também foram realizados testes para identificar um valor de velocidade seguro para a operação do sistema. Para isso, a velocidade do sistema de tração foi aumentada gradativamente, até que a vibração do sistema se tornasse significativa. Desta forma foi identificado que o limite seguro para operação do módulo era de 38 pol/s, aproximadamente 22% da velocidade teórica máxima dele.

O sistema de controle foi codificado conforme proposto na Figura 19, utilizando dois controladores de PID independentes para cada sistema. Estes controladores foram calibrados individualmente, a Tabela 6 traz os ganhos obtidos para cada um.

Tabela 6 - Ganhos de PID encontrados para o módulo protótipo.

<b>Controlador</b>	<b>Kp</b>	<b>Ki</b>	<b>Kd</b>
C_PIDrot	0,007	0,01	0
C_PIDvel	0,03	0	0

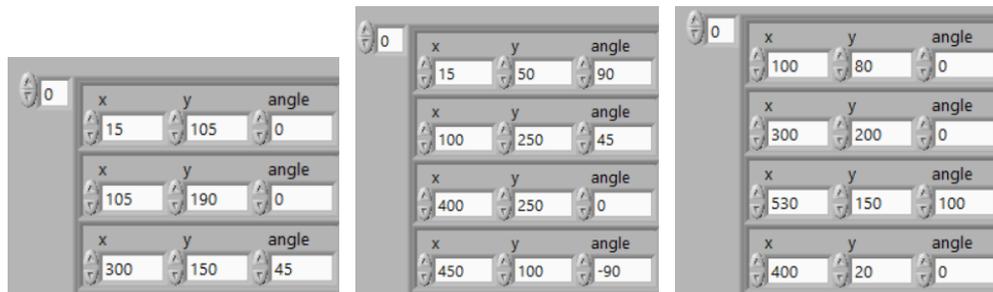
Fonte: Elaborado pelo autor.

Para seguir a trajetória calculada, a tabela gerada no capítulo anterior é primeiramente carregada na memória do RoboRIO. Em seguida é realizada uma leitura, linha a linha, desta tabela a cada 20ms atualizando os valores de velocidade e direção nos controladores PID. Desta forma o sistema percorre a trajetória completa.

## 5. ANÁLISE DE RESULTADOS

Conforme discutido no capítulo 3.3 o sistema desenvolvido foi avaliado em 2 áreas distintas, em sua capacidade de gerar trajetórias e de segui-las. Para isso, foram definidos 3 conjuntos de pontos de interesse diferentes para que sejam avaliados. Estes pontos são apresentados na Figura 28.

Figura 28 - Pontos de interesse utilizados na análise do sistema, da esquerda para direita T1, T2 e T3.

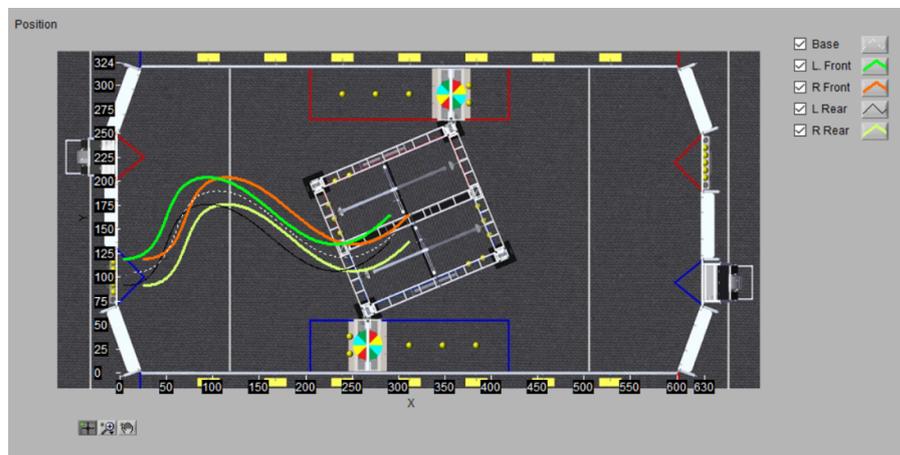


Fonte: Elaborado pelo autor.

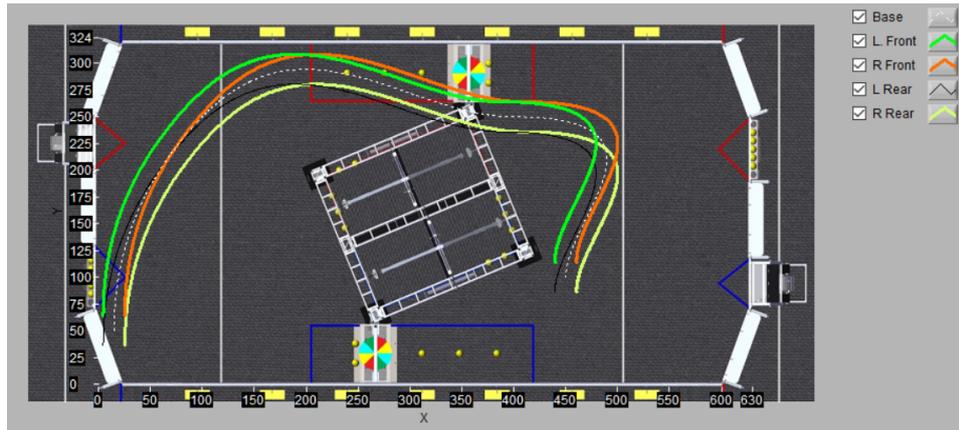
### 5.1. ANÁLISE DO GERADOR DE TRAJETÓRIA

Todos os conjuntos foram executados no sistema, utilizando o parâmetro de velocidade máxima estabelecido previamente. Os caminhos resultantes estão apresentados na Figura 29. Como o sistema foi executado para um robô que possua 4 módulos, o gráfico gerado pelo programa apresenta um caminho para cada módulo.

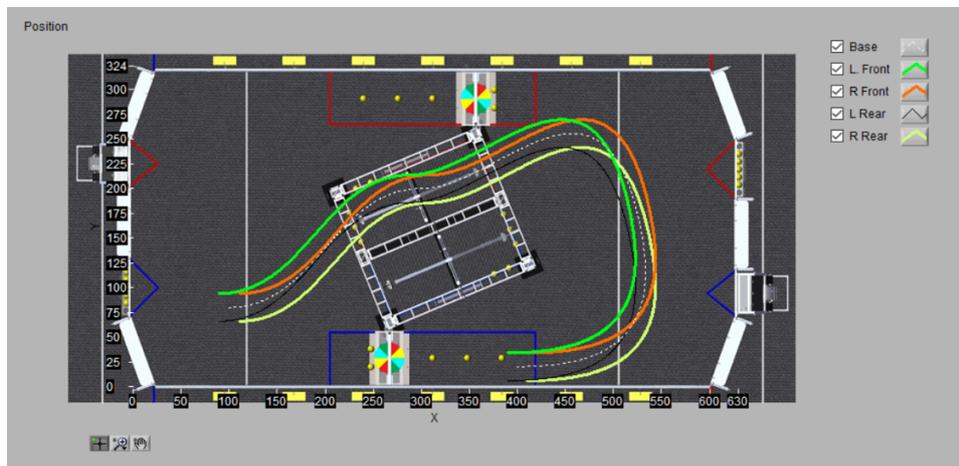
Figura 29 - Caminhos obtidos utilizando os pontos de interesse T1(a), T2(b) e T3(c).



(a)



(b)

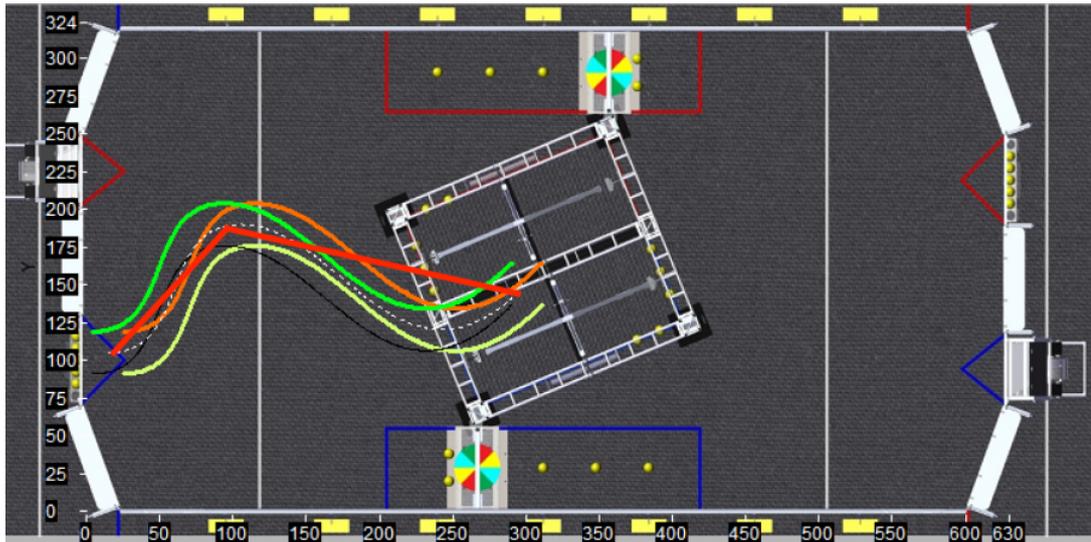


(c)

Fonte: Elaborado pelo autor.

Os pontos de interesse escolhidos geram caminhos bastante distintos, mas que são comuns dentro da aplicação que considerada neste projeto. Na Figura 30 é apresentada uma comparação entre o caminho gerado para os pontos de interesses T1 com o caminho mais curto possível. Dada as restrições de orientação, é possível observar que o caminho gerado está muito próximo do ótimo, sendo a única forma de torná-lo mais curto seria modificando o sistema para que ele pudesse permitir movimentos de translação e rotação simultâneos.

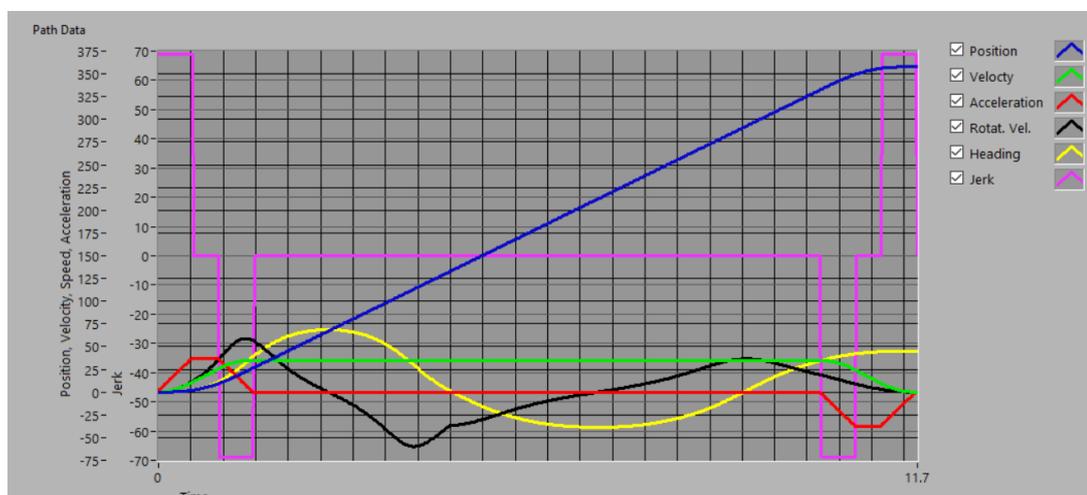
Figura 30 - Caminho T1 ideal(vermelho) *versus* caminho gerado (branco pontilhado).



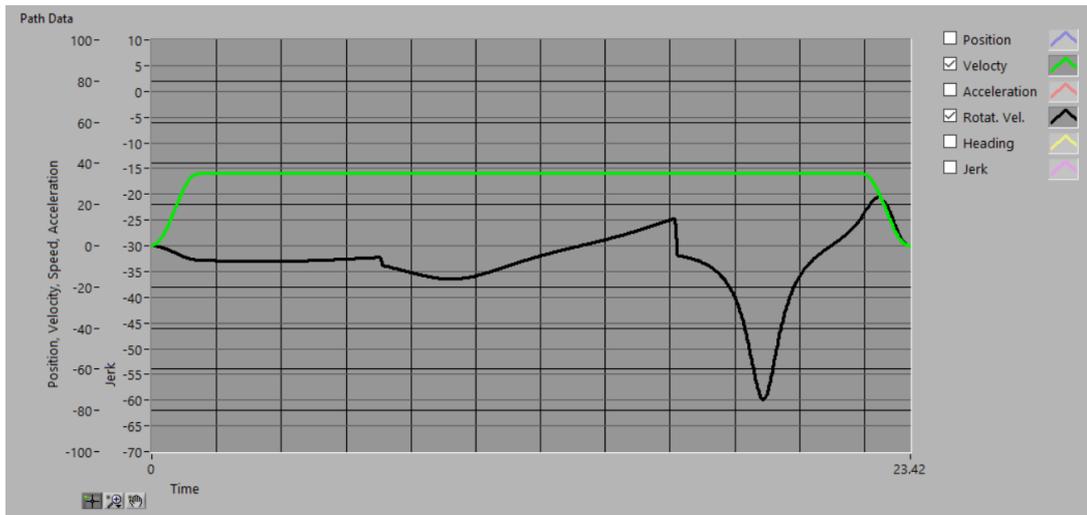
Fonte: Elaborado pelo autor.

É visível que em praticamente todos os cenários os caminhos possuem uma trajetória suave e contínua, ou seja, não há picos de variação instantâneas no movimento de rotação do módulo. A Figura 31 apresenta os resultados da trajetória com valores de velocidade e orientação, neles pode ser observado que há apenas 1 pico significativo de velocidade de orientação do módulo na trajetória T3. Este pico demonstra que se faz necessário a implementação de uma lógica que permita limitar a aceleração da orientação do módulo ao longo da trajetória.

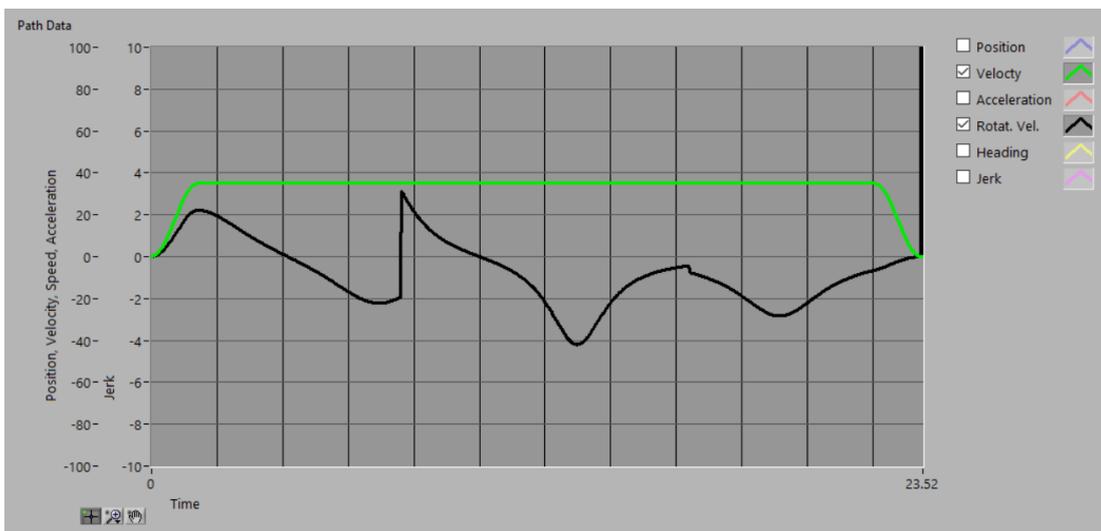
Figura 31 - Gráfico de velocidade e orientação do módulo gerado para os caminhos T1(a), T2(b) e T3(c).



(a)



(b)



(b)

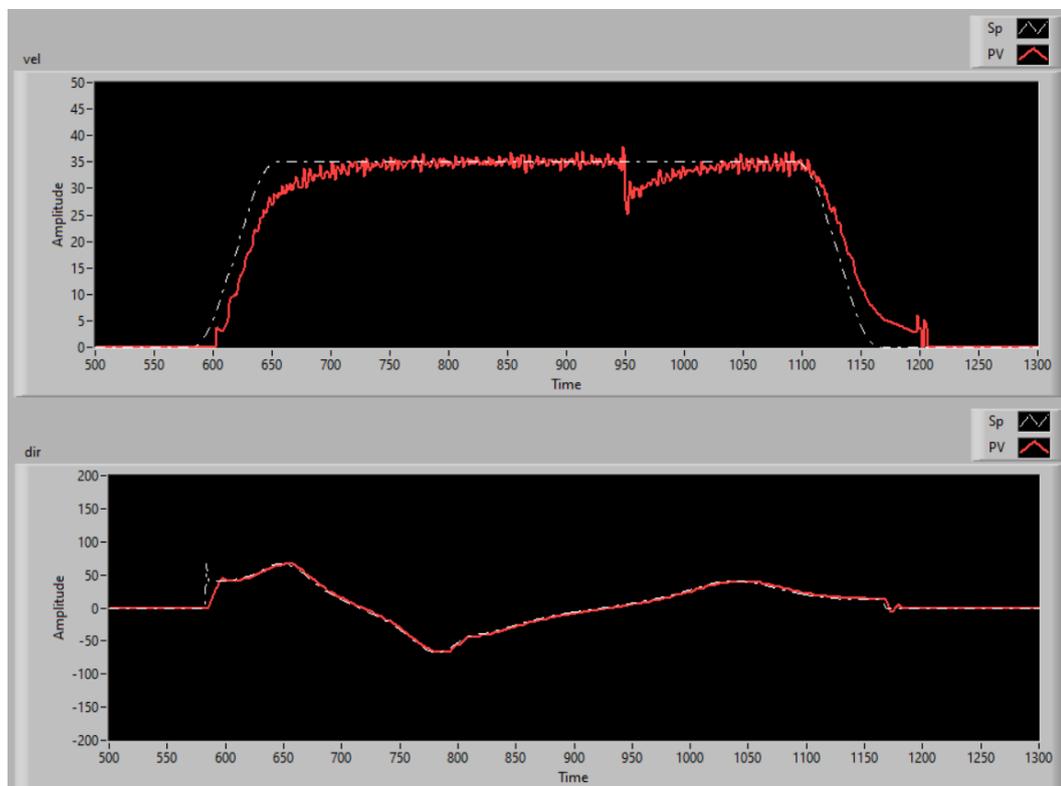
Fonte: Elaborado pelo autor.

Com base nos gráficos de velocidade de tração do módulo é possível afirmar que o perfil de movimento trapezoidal permite um ótimo aproveitamento da potência disponível no motor, uma vez que em todos os casos o motor de tração está utilizando a velocidade máxima permitida pelo sistema em todos os momentos em que isto é possível. Com exceção do tempo de aceleração e desaceleração do módulo, que totaliza 2,68s por trajetória.

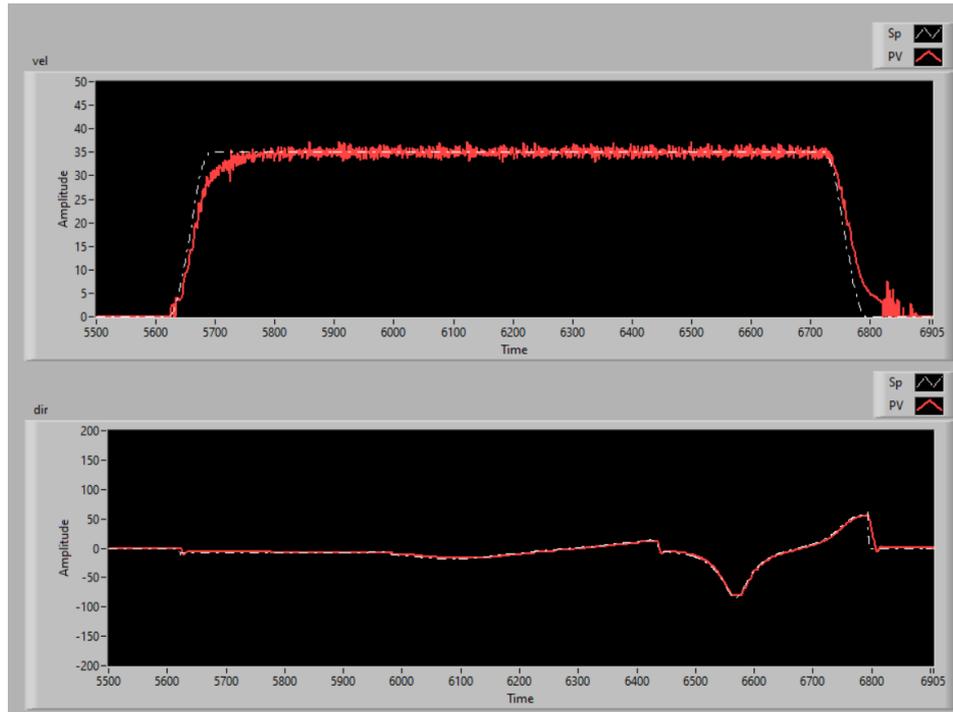
## 5.2. ANÁLISE DO SISTEMA DE CONTROLE

As trajetórias geradas foram carregadas e executadas utilizando o *software* de controle desenvolvido no capítulo 4.3. Tanto a leitura dos encoders quanto o sinal de posição e velocidade sendo escrito pela trajetória foram registrados ao longo dos testes, e são apresentados na Figura 32 onde a posição e velocidade desejadas estão representadas por uma linha branca pontilhada e a leitura dos sensores está em uma linha vermelha contínua.

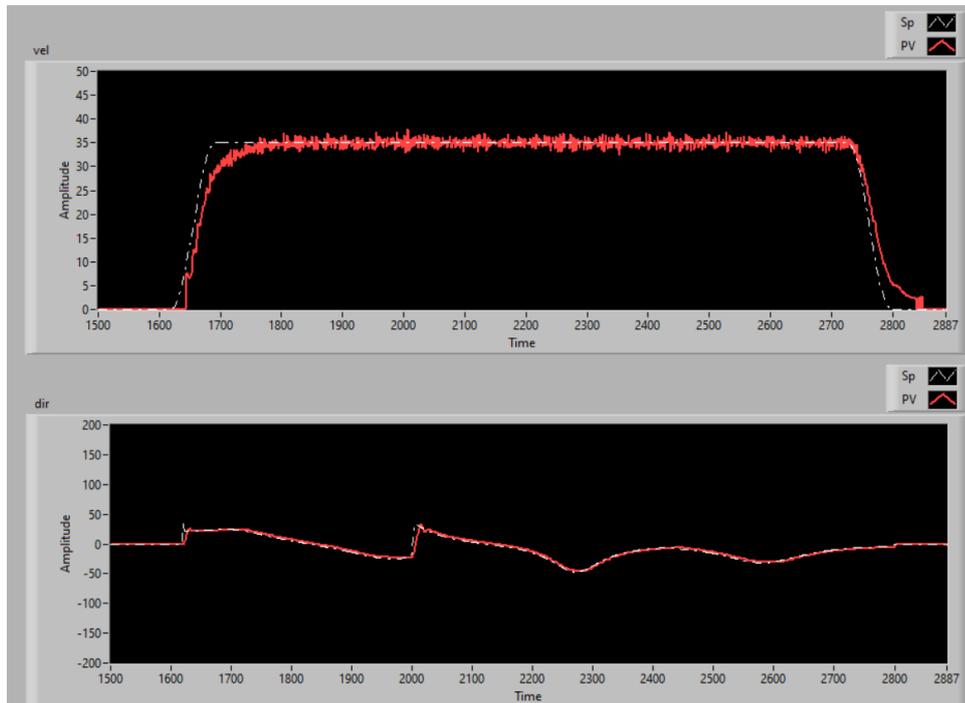
Figura 32 - Resultado de execução das trajetórias T1(a), T2(b) e T3(c). Acima, a velocidade do módulo e abaixo a sua orientação.



(a)



(b)



(c)

Fonte: Elaborado pelo autor.

Nos cenários apresentados, o sistema foi capaz de seguir quase perfeitamente a trajetória de orientação do módulo, isso se deve principalmente por ser um sistema mais lento e de menor inercia, o que torna o seu controle relativamente simples. Já o controle da roda de tração se

mostrou mais complexo, é visível que em todos os casos o sistema está atrasado em relação a trajetória desejada. A Tabela 7 apresenta o tempo estimado para a execução de cada trajetória, assim como o tempo medido durante os testes.

Tabela 7 - Comparação entre o tempo calculado e executado.

<b>Trajectoria</b>	<b>Tempo calculado (s)</b>	<b>Tempo medido (s)</b>	<b>Erro</b>
<b>T1</b>	11,70	12,46	+6,1%
<b>T2</b>	23,42	24,70	+5,1%
<b>T3</b>	23,52	24,58	+4,3%

Fonte: Elaborado pelo autor.

É visível que ainda existe margem para melhoria na malha de controle de velocidade do módulo, uma vez que o erro no tempo de execução é significativo. Isto pode ser corrigido com a utilização de um sistema mecânico mais robusto e sensores de maior qualidade, uma vez que estes fatores introduzem muito ruído no sistema e dificultam seu controle de forma apropriada.

## 6. CONCLUSÕES

Dentro dos objetivos previstos no capítulo 1.1, é possível afirmar que os itens (a) a (f) foram atingidos em sua totalidade. Onde o item (a) foi realizado através de uma análise de trabalhos anteriores.

Para os itens (b) e (c), o *software* para gerar trajetórias, embora tenha partido de uma biblioteca, foi adaptado e executado com êxito, de forma a permitir a construção de caminhos e trajetórias para qualquer robô de movimento omnidirecional do tipo *Swerve* a partir de um sistema de pontos de interesse. Embora algumas adaptações tenham sido necessárias e nem todos os recursos desejados tenham sido implementados.

O algoritmo implementado para realizar o controle do sistema funcionou conforme esperado, inclusive superando as expectativas quanto ao controle de posição do módulo. Embora, conforme comentado durante a análise de resultados, a malha de controle de velocidade ainda possui margem para melhoria. Desta forma, concluindo o objetivo proposto no item (d).

Em relação ao item (e), o design e construção do protótipo para validação do sistema se mostrou um desafio maior que o esperado inicialmente, havendo a necessidade de criar diferentes iterações dele. Assim como o acesso ao equipamento necessário para validá-lo e o tempo necessário para realizar a impressão 3D de todos os componentes acabou por atrasar o cronograma do projeto, limitando a implementação de alguns recursos. As análises realizadas também mostram que a utilização de sensores de maior qualidade, assim como a implementação de um protótipo que possua menores folgas mecânicas podem melhorar significativamente o resultado do sistema.

Com os resultados obtidos, foi possível validar a aplicação de um sistema de controle de trajetórias avançadas utilizando equipamentos da competição de robótica FRC. Isto permitirá que o mesmo possa ser utilizado futuramente na competição, entregando um ganho de performance significativo para a equipe que o utilizará e concluindo o objetivo proposto no item (f).

Como sugestão para continuidade deste projeto, a implementação de recursos como a manobra de Ackerman dupla e movimentos de translação e rotação simultâneos no software gerador de trajetória são opções interessantes para tornar o movimento do robô mais rápido. Uma vez que permitem que o robô possa executar as mesmas trajetórias, mas realizando caminhos mais curtos e eficientes.

Para aplicação em um projeto futuro, também pode ser realizado um protótipo de maior robustez, com 4 módulos *Swerve* para operação do sistema. Desta forma o mesmo *software* realizado neste trabalho pode ser implementado, porém será possível realizar análises mais precisas da resposta e eficiência do sistema de controle.

## REFERÊNCIAS

ANDYMARK: Swerve and Steer. [S. l.], 1 nov. 2019. Disponível em: <https://www.andymark.com/products/swerve-and-steer>. Acesso em: 28 nov. 2020.

BANKER, Steve. **Robots And The Autonomous Supply Chain**. *In Forbes*. 2 abr. 2020. Disponível em: <https://www.forbes.com/sites/stevebanker/2020/04/02/robots-and-the-autonomous-supply-chain/?sh=5dfe9e7787a3>. Acesso em 15 nov. 2020.

BHATTACHARYA, Ananya. **Amazon is just beginning to use robots in its warehouses and they're already making a huge difference**. QUARTZ, [S. l.], 17 jun. 2016. Disponível em: <https://qz.com/709541/amazon-is-just-beginning-to-use-robots-in-its-warehouses-and-theyre-already-making-a-huge-difference/>. Acesso em: 20 nov. 2020.

BARRERO, O.; TILAGUY, S.; NOVA, Y. M. **Outdoors Trajectory Tracking Control for a Four Wheel Skid-Steering Vehicle**. 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), Barranquilla, 2018, pp. 1-6.

CHAUDHARI, T.; JHA, M.; BANGAL, R.; CHINCHOLKAR, G. **Path Planning and Controlling of Omni-Directional Robot Using Cartesian Odometry and PID Algorithm**. 2019 International Conference on Computing, Power and Communication Technologies (GUCON), NCR New Delhi, India, 2019, pp. 63-68.

FIRST: Game and Season Manual. [S. l.], 5 jan. 2020. Disponível em: <https://www.firstinspires.org/resource-library/frc/competition-manual-qa-system>. Acesso em: 25 nov. 2020.

JIA, W.; YANG, G.; WANG, C.; ZHANG, C.; CHEN, C.; FANG, Z. **Energy-Efficient Torque Distribution Optimization for an Omnidirectional Mobile Robot with Powered Caster Wheels**. *Energies* 2019, 12, 4417.

LIAN, Jianfang; YU, Wentao; XIAO, Kui; LIU, Weirong. **Cubic Spline Interpolation-Based Robot Path Planning Using a Chaotic Adaptive Particle Swarm Optimization Algorithm**. *Mathematical Problems in Engineering*, vol. 2020, Article ID 1849240, 20 pages, 2020.

MELCHIOR, Alan L.; BURACK, Cathy; HOOVER, Matthew. **Impacts of After-School Robotics Programming on STEM Interests and Attitudes**. AERA Online Paper Repository, New York, NY, 13 abr. 2018. Disponível em: [https://www.firstinspires.org/sites/default/files/uploads/resource\\_library/impact/impacts-of-after-school-robotics-programming-on-stem-interests-and-attitudes.pdf](https://www.firstinspires.org/sites/default/files/uploads/resource_library/impact/impacts-of-after-school-robotics-programming-on-stem-interests-and-attitudes.pdf). Acesso em: 12 nov. 2021.

SIMPSON, Jim. **FRC LabVIEW Trajectory Library**. Github. Disponível em: [https://github.com/jsimpso81/FRC\\_LV\\_TrajLib](https://github.com/jsimpso81/FRC_LV_TrajLib). Acesso em 10 mai. 2022

SUN, Q; LI, S.; KANG, X. **Analysis on Kinematics of Omnidirectional Full Drive Mobile Robot**. 2018 IEEE International Conference on Information and Automation (ICIA), Wuyishan, China, 2018, pp. 615-620.

YIN, J.; YANG, G.; ZHAO, F.; QIU, H. **Motion planning implemented in ROS for Omni-directional Wheeled Mobile Robot.** 2015 IEEE International Conference on Information and Automation, Lijiang, 2015, pp. 2695-2700.

ZHANG, L.; SUN, L.; ZHANG, S; LIU, J. **Trajectory planning for an indoor mobile robot using quintic Bezier curves.** 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 757-762.

ZHU, S.; GELBAL, S. Y.; AKSUN-GUVENC, B. **Online Quintic Path Planning of Minimum Curvature Variation with Application in Collision Avoidance.** 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 669-676.

