UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO - CECAU

JÚLIA GHERARDI MAIA DE SOUZA

# PLANEJAMENTO DE ROTAS PARA ROBÔS DIFERENCIAIS DA CATEGORIA VSSS UTILIZANDO ALGORITMOS GENÉTICOS ALIADOS À CURVA DE BÉZIER

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2022

**JÚLIA GHERARDI MAIA DE SOUZA**

# PLANEJAMENTO DE ROTAS PARA ROBÔS DIFERENCIAIS DA CATEGORIA VSSS UTILIZANDO ALGORITMOS GENÉTICOS ALIADOS À CURVA DE BÉZIER

**Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.**

Orientador: Prof. Alan Kardek Rêgo Segundo, Dr.Sc.

**Ouro Preto**
**Escola de Minas – UFOP**
**2022**

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S729p Souza, Julia Gherardi Maia De.
Planejamento de rotas para robôs diferenciais da categoria VSSS utilizando algoritmos genéticos aliados à curva de Bézier. [manuscrito] / Julia Gherardi Maia De Souza. - 2022.
52 f.: il.: color., gráf.. + Algoritmos.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo.
Coorientador: Prof. Dr. Agnaldo José da Rocha Reis.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas. Graduação em Engenharia de Controle e Automação .

1. Robôs. 2. Algoritmos genéticos. 3. Bézier, Curva de. 4. Robôs - Futebol. 5. Robôs - Planejamento de rotas. I. Reis, Agnaldo José da Rocha. II. Segundo, Alan Kardek Rêgo. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

**FOLHA DE APROVAÇÃO**

**Júlia Gherardi Maia de Souza**

**Planejamento de Rotas para Robôs Diferenciais da Categoria VSSS Utilizando Algoritmos Genéticos Aliados à Curva de Bézier**

Monografia apresentada ao Curso de Engenharia de controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheira de Controle e Automação

Aprovada em 29 de novembro de 2022

Membros da banca

Prof. Dr. Alan Kardek Rêgo Segundo - Orientador (Universidade Federal de Ouro Preto)
Prof. Dr. Agnaldo José da Rocha Reis - Coorientador (Universidade Federal de Ouro Preto)
Profa. Dra. Karla Boaventura Pimenta Palmieri - Convidada (Universidade Federal de Ouro Preto)

Alan Kardek Rêgo Segundo, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 06/12/2022

Documento assinado eletronicamente por **Alan Kardek Rego Segundo**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 06/12/2022, às 14:04, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0434249** e o código CRC **835D3B19**.



R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

# AGRADECIMENTOS

Agradeço aos meus pais, minha avó e amigos que sempre estiveram ao meu lado me dando forças para não desistir. Agradeço em especial ao meus amigo Samuel que contribuiu na realização deste trabalho e que sempre esteve ao meu lado apesar das dificuldades. Por fim, agradeço aos integrantes da equipe Rodetas, ao professor Alan Kardek por todas as oportunidades de crescimento profissional e pessoal que me proporcionou durante o curso e aos demais professores que contribuíram diretamente em minha formação acadêmica.

*“O começo de todas as ciências é o espanto de as coisas serem o que são“. (Aristóteles)*

# RESUMO

A robótica móvel vem se mostrando cada vez mais presente no cotidiano em diversos cenários, como na indústria, na medicina e até mesmo na agricultura. Diante dessa variedade de aplicações, escolheu-se uma delas para ser adotada no presente trabalho: o cenário de competições de futebol de robôs na categoria IEEE VSSS (*Very Small Size Soccer*. Tendo em vista que esse cenário é altamente dinâmico e veloz, este trabalho traz o desenvolvimento de um planejador de caminhos utilizando algoritmos genéticos. Através de escolhas tanto arbitrárias quanto baseadas em métodos de seleção de melhores indivíduos, que são por sua vez uma sequência de números correspondentes a trechos do campo, mutações são realizadas a fim de encontrar o melhor e mais otimizado caminho entre os pontos inicial e final pré-definidos. A intenção deste trabalho é, de assim, solucionar os problemas identificados na Equipe Rodetas Robô Clube, da Universidade Federal de Ouro Preto, referentes à precisão, velocidade e eficiência dos robôs quando atuando em partidas competitivas. Uma vez escolhido o caminho para que o robô o siga de maneira precisa e suave, curvas de Bézier são implementadas para deixar o movimento menos robótico e as curvas mais brandas. Os resultados obtidos apontam que o algoritmo genético implementado foi capaz de gerar propostas de rotas factíveis e ainda apontar a mais otimizada entre elas, fazendo com que o robô atinja o alvo da maneira esperada. Já a implementação da curva de Bézier foi fundamental para que as curvas da rota escolhida fossem suavizadas sem perder o objetivo do traçado.

**Palavras-chaves**: Robôs Autônomos; Algoritmos Genéticos; Curvas de Bézier; Futebol de robôs; Planejamento de rotas.

# ABSTRACT

Mobile robotics are increasingly present in everyday life multiple scenarios, such as industry, medicine and even in agriculture. Considering this range of applications, this paper presents the field of robots soccer competition in the IEEE VSSS category. Considering the dinamicity and speed of this scenario, the paper brings foward the development of a path planning using genetic algorithms that through both arbitrary choices and based on methods of selecting the best individuals, which are a sequence of numbers corresponding to branches of the field, mutations are carried out in order to find the best and most optimized path between the initial and final points pre-defined with the intention to solve the problems identified until the current moment at Rodetas Robô Clube from Federal University of Ouro Preto regarding to precision, quickness and efficiency of the robots while playing competitive matches. Once the path is chosen for the robot to follow in an accurately and smoothly way, Bézier curves are implemented to make the movement less robotic and the curves lenient. The results indicate that the implemented genetic algorithm wad capable of genertate options of fisible paths besides choose de best path among all, leading the robot to the target as expected. The Bezier Curve implementation was fundamental for the smoothing movement of the robot in the chosen path curves without losing the target.



**Key-words**: Autonomous Robots; Genetic Algorithms; Bezier Curves; Robot Soccer; Path Planning.

# LISTA DE ILUSTRAÇÕES


Figura 1 – Fluxograma linear representado os segmentos a serem tratados pelos códigos das equipes. . . . . 13
Figura 2 – Visão geral do ambiente VSSS (IEEE, 2008). . . . . 14
Figura 3 – Robô diferencial e suas variáveis (SIMBA; UCHIYAMA; SANO, 2016). . . 17
Figura 4 – Robô da Equipe Rodetas Robô Clube. . . . . 18
Figura 5 – Campo da categoria VSSS e suas medidas. . . . . 19
Figura 6 – Simulação de uma partida no VSS-Viewer. . . . . 20
Figura 7 – Fluxograma de um Algoritmo Genético. . . . . 22
Figura 8 – Representação *crossover*. . . . . 22
Figura 9 – Representação de *Crossover de Dois Pontos* (PACHECO, 1999). . . . . 23
Figura 10 – Representação de *Crossover Uniforme* (PACHECO, 1999). . . . . 23
Figura 11 – Representação de seleção pelo método da roleta. . . . . 25
Figura 12 – Construção de uma curva de Bézier de grau 3. . . . . 26
Figura 13 – *Grid 15 x 13*. . . . . 31
Figura 14 – Demarcação do *grid* sobre o campo. . . . . 32
Figura 15 – Campo representado como *grid* com células demarcadas. . . . . 32
Figura 16 – Representação de uma rota a ser percorrida. . . . . 33
Figura 17 – Representação da menor distância entre o ponto A e B. . . . . 35
Figura 18 – Avaliação do tamanho população inicial x tempo de execução do AG. . . . . 42
Figura 19 – Avaliação do tamanho população inicial x tempo de execução do AG. . . . . 43
Figura 20 – Avaliação de taxa de mutação x tempo de execução do AG. . . . . 43
Figura 21 – Taxa de mutação do AG x *fitness* do melhor indivíduo. . . . . 44
Figura 22 – Avaliação de taxa de *crossover* x tempo de execução do AG. . . . . 44
Figura 23 – Taxa de *crossover* do AG x *fitnes* do melhor indivíduo. . . . . 45
Figura 24 – Caminho gerado pelo AG. . . . . 46
Figura 25 – Caminho gerado pelo AG antes e após aplicação da Curva de Bézier. . . . . 46
Figura 26 – Tempo de execução do AG. . . . . 47

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AG | Algoritmo genético |
| CBR | Competição Brasileira de Robótica |
| IEEE | Instituto de Engenheiros Eletricistas e Eletrônicos |
| LARC | *Latin American Robotics Competition* |
| PWM | Modulação por largura de pulso |
| VSSS | Very Small Size Soccer |

# LIST OF ALGORITHMS


1 Estrutura geral do algoritmo genético . . . . . 34
2 Inicialização de indivíduos . . . . . 37
3 Cálculo da *fitness* . . . . . 38
4 *Crossover* . . . . . 39
5 Mutação . . . . . 40
6 Definição da curva de Bézier . . . . . 41

# SUMÁRIO


**1 INTRODUÇÃO . . . . . 13**
1.1 Objetivos gerais . . . . . 15
1.2 Objetivos específicos . . . . . 15
1.3 Contribuições do trabalho . . . . . 16
1.4 Estrutura do trabalho . . . . . 16

**2 REFERENCIAL TEÓRICO E REVISÃO DA LITERATURA . . . . . 17**
2.1 Robô diferencial da categoria VSSS . . . . . 17
2.2 Regras e definições da categoria VSSS . . . . . 18
*2.2.1 Robôs jogadores . . . . . 18*
*2.2.2 O campo, a bola e suas dimensões . . . . . 19*
*2.2.3 A partida . . . . . 20*
2.3 Algoritmos genéticos . . . . . 20
*2.3.1 Operadores genéticos . . . . . 22*
*2.3.2 Métodos de seleção . . . . . 24*
2.4 Curvas de Bézier e suas aplicações . . . . . 25

**3 DESENVOLVIMENTO . . . . . 28**
3.1 Ferramentas . . . . . 28
3.2 IEEE Very Small Size Soccer (VSSS) . . . . . 28
*3.2.1 Sistema de processamento externo* - VSS-SDK . . . . . *28*
*3.2.2 Módulo* VSS-Simulator . . . . . *28*
*3.2.3 Módulo* VSS-Viewer . . . . . *29*
*3.2.4 Módulo* VSS-Vision . . . . . *29*
3.3 Arquitetura de *software* da equipe Rodetas . . . . . 29
*3.3.1 Estratégia de jogo . . . . . 30*
*3.3.2 Movimentação . . . . . 30*
3.4 Algoritmos genéticos no planejamento de rotas . . . . . 30
*3.4.1 Representação do ambiente . . . . . 31*
*3.4.2 Definição dos indivíduos . . . . . 32*
*3.4.3 Definição do algoritmo genético . . . . . 34*
3.4.3.1 Definição de parâmetros do algoritmo genético . . . . . 35
3.4.3.2 Inicialização da população . . . . . 36
3.4.3.3 Cálculo e definição de regras da *fitness* . . . . . 36
3.4.3.4 *Crossover* . . . . . 38
3.4.3.5 Mutação . . . . . 38

3.4.3.6 Elitismo . . . . . 38
3.5 Curva de Bézier na suavização de rotas curvilíneas . . . . . 41

**4 RESULTADOS . . . . . 42**
4.1 Avaliação de parâmetros . . . . . 42
4.2 Execução do algoritmo genético . . . . . 45

**5 CONCLUSÃO E TRABALHOS FUTUROS . . . . . 48**

**REFERÊNCIAS . . . . . 49**

# 1 INTRODUÇÃO

A robótica móvel é um tópico especial no ramo da automação. Novas tecnologias vem sendo desenvolvidas a fim de contemplar uma gama de problemas encontrados no cotidiano. Soluções utilizando robôs autônomos aplicadas à agricultura podem ser encontradas no trabalho de Sanchez-Hermosilla, Paez e Rincon V. (2012) e soluções no campo de análises de solo com Parizoto (2010). Dentre esses ramos, pode-se destacar o desenvolvimento de veículos autônomos que são capazes de, a partir de um objetivo e sua localização atual, traçar a melhor rota entre esses dois pontos, considerando obstáculos e outros impedimentos no caminho. O trabalho de Raza (2018) apresenta um exemplo de veículo autônomo aplicado à um carro tripulado de passeio.

Tendo essa alta variedade de áreas onde a robótica móvel e, mais especificamente, os robôs autônomos podem atuar, para o presente trabalho escolheu-se o desenvolvimento de tecnologias voltadas para competições de futebol de robôs na categoria IEEE Very Small Size Soccer (VSSS), na qual a equipe Rodetas Robô Clube, da UFOP, atualmente participa.

Nessa modalidade de competição, duas equipes se enfrentam, com três robôs cada, que podem medir até $7,5\ cm\ x\ 7,5\ cm\ x\ 7,5\ cm$, sendo completamente autônomos, ou seja, sem controle humano. Os jogos acontecem em um campo com dimensão de $150\ cm\ x\ 130\ cm$, onde a bola possui $46\ g$ e $4,27\ cm$ de diâmetro. As equipes se enfrentam em partidas de $10$ minutos divididas em dois tempos de $5$ minutos, onde a equipe que tiver o maior número de gols marcados ao final do tempo vence, ou quem abrir dez gols de diferença em relação ao adversário.

Para cumprir com os objetivos, as equipes devem ser capazes de transformar as informações coletadas em campo em comandos para os robôs, seguindo o fluxo da Figura 1.

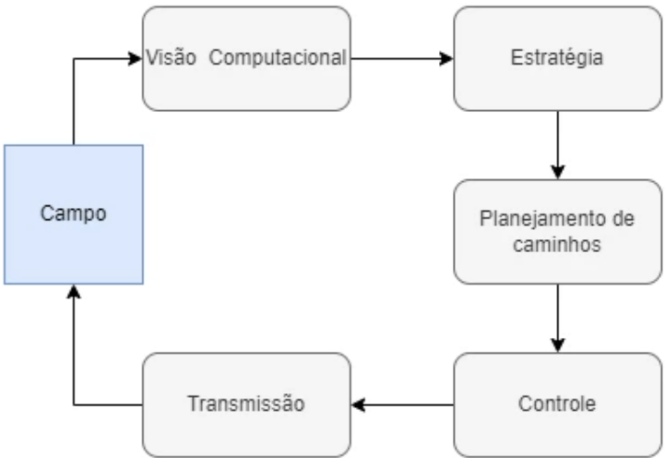


Figura 1 – Fluxograma linear representado os segmentos a serem tratados pelos códigos das equipes.

Começando pelo segmento da visão computacional, tem-se uma câmera posicionada

em cima do campo que capta as imagens do jogo em tempo real. Um *software* trata as imagens recebidas, fazendo a identificação das cores presentes em cima de cada robô. A cor primária representa a equipe a qual o robô pertence, sendo ela azul ou amarelo. Já a cor secundária dá uma identificação individual ao robô, que pode ser de qualquer cor, exceto azul, amarelo, branco, cinza e laranja. Identificados os robôs, o *software* é capaz de calcular as velocidades angulares e lineares e a posição e a orientação de cada um dos componentes em campo. A Figura 2 mostra estrutura do ambiente durante uma partida da categoria VSSS.

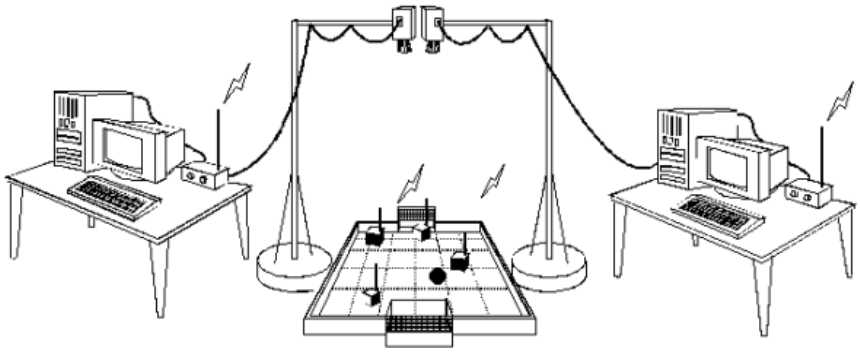

Figura 2 – Visão geral do ambiente VSSS (IEEE, 2008).

A partir dessas informações disponibilizadas pelo *software*, passa-se para a definição da estratégia a ser seguida por cada robô, dado o cenário de jogo do momento. As diferenças no comportamento de cada robô pode ser notada dada sua função na equipe. Por exemplo, enquanto a função do goleiro é permanecer próximo ao gol defendendo os ataques, a de um defensor é de retirar a bola do campo de defesa e levá-la para o campo de ataque. Por último, a função de um atacante é a de levar a bola em direção ao gol adversário a fim de pontuar. A estratégia define qual robô executará cada função em determinado momento, podendo haver uma troca de funções entre eles durante a partida, como por exemplo, um robô que é defensor se tornar goleiro em determinado ponto. Além disso, a estratégia é responsável por definir a posição e a orientação a serem alcançadas pelo robô em seu objetivo final.

O próximo seguimento é o de planejamento de caminho, que tem como finalidade gerar a melhor rota para que o robô deixe a posição inicial e atinja a posição desejada. Cada robô deve atuar de forma a, na medida do possível, evitar colisão com adversários e até mesmo com robôs do mesmo time. Para isso, o caminho escolhido deve levar em consideração o ambiente dinâmico e os obstáculos para levar o robô ao ponto final desejado. O caminho gerado, por sua vez, deve, de preferência, ser o mais curto possível, uma vez que é necessário ser mais rápido que os adversários.

Na fase seguinte, o robô deve executar o caminho gerado, usando o segmento do controle, responsável por fazer o robô seguir com precisão e agilidade o caminho escolhido. Neste momento, as informações geradas pelo planejamento de caminho são convertidas em valores PWM (*Pulse Width Modularization*), que serão enviados para as rodas de cada robô. Neste

módulo, as decisões tomadas até então pelos módulos anteriores são transformadas em ações que devem ou não serem executadas em campo.

O último seguimento é o de transmissão, que recebe os valores gerados pelo controle, encapsula-os em um pacote de envio de dados e os transmite via rede sem fio aos robôs. Cada robô é munido de um Arduino que decodifica esse pacote enviando finalmente aos motores as informações de velocidade a serem aplicadas em cada roda, já que este trabalho trata de robôs diferenciais.

Dado este contexto, pode-se notar que as áreas tecnológicas desenvolvidas são diversas, abrindo espaço para o uso de novos métodos e abordagens. A partir disso, neste trabalho, adota-se o uso de algoritmos genéticos no segmento do planejamento de caminho do robô. Segundo Katoch, Chauhan e Kumar (2020) um algoritmo genético é um algoritmo de otimização inspirado na seleção natural. É um algoritmo de busca baseado em populações que utiliza do conceito de sobrevivência do mais apto ao ambiente adotado. Esta metodologia é implementada por estar-se trabalhando em um ambiente dinâmico, onde utilizar apenas de funções lineares para avaliação pode não ser muito prático. Portanto, o algoritmo genético surge como opção por comportar-se bem nesse tipo de cenário.

Além disso, a utilização do conceito de curvas de Bézier é um fundamental complemento para a abordagem adotada, a fim de suavizar as curvas feitas pelo robô durante o trajeto. Com isso, uma vez que obtida a rota, o controlador usa a saída fornecida anteriormente como entrada para calcular os valores PWM, e guia o robô pelo melhor caminho com suavidade, precisão e velocidade.

## 1.1 Objetivos gerais

Como objetivo geral, este trabalho visa trazer novas soluções para o problema de planejamento de caminho encontrado atualmente pela Equipe Rodetas Robô Clube. Com isso, busca-se desenvolver, utilizando algoritmos genéticos, um sistema capaz de traçar rotas e fazer com que o robô atinja seu objetivo de forma precisa e veloz, evitando obstáculos no caminho, de forma suave, levando em conta a implementação de curvas de Bézier.

## 1.2 Objetivos específicos

Como objetivos específicos tem-se o estudo de aplicações de algoritmos genéticos em problemas de diversos cenários similares ao proposto neste trabalho, assim como a análise da usabilidade de curvas de Bézier com o intuito de dar suavidade aos movimentos robóticos em um ambiente dinâmico e altamente mutável, a fim de operar robôs diferenciais da Equipe Rodetas na categoria VSSS.

## 1.3 Contribuições do trabalho

Neste trabalho utilizou-se uma abordagem envolvendo algoritmos genéticos e curvas de Bézier para o planejamento de rotas. Essa abordagem envolve duas técnicas propostas, que combinadas, permitem que o robô consiga alcançar seu objetivo desviando de obstáculos e criando caminhos mais curtos, e por consequência, mais rápidos se comparados com a estratégia atualmente utilizada pela equipe Rodetas Robô Clube.

No trabalho de Song, Wang e Sheng (2016) são abordados os tópicos relacionados ao planejamento de caminho utilizando algoritmos genéticos, assim como a suavidade das curvas feitas pelo robô ao seguir o caminho proposto utilizando Bézier, porém em um cenário estático e muito menos complexo. Já no trabalho de Choi, Curry e Elkaim (2008) as curvas de Bézier são utilizadas para suavizar rotas curvilíneas, mas ainda assim em um cenário inerte. Com isso, o presente trabalho propõe resolver o problema da dinamicidade de cenários ao aplicar algoritmos genéticos ao planejamento de caminhos e também curvas de Bézier no mesmo cenário.

Outra contribuição importante deste trabalho são os avanços trazidos para a equipe Rodetas, que poderá utilizar os métodos aplicados aqui para fins competitivos na categoria VSSS tendo dentro de seus códigos técnicas e algoritmos computacionais bem revisados bibliograficamente, além de deixar espaço para futuras melhorias e estudos pelos integrantes da equipe.

## 1.4 Estrutura do trabalho

Os próximos capítulos deste trabalho são organizados como segue. No Capítulo 2 é apresentada a revisão da literatura. O Capítulo 3 discorre sobre o desenvolvimento. Os experimentos e resultados são apresentados e discutidos no Capítulo 4. Por fim, o Capítulo 5 apresenta as conclusões a respeito do trabalho produzido e aponta futuras melhorias para a área de pesquisa escolhida.

# 2 REFERENCIAL TEÓRICO E REVISÃO DA LITERATURA

No problema de planejamento de caminho, o robô é considerado como um ponto no espaço de configurações e todas as dinâmicas e incertezas são desconsideradas. Dessa forma, o objetivo do planejamento é produzir uma ordem de configurações que leve o robô de uma configuração inicial a uma configuração de destino (MOLINA, 2014). Para tornar isso possível, é necessário que o mapa de ocupação do campo seja gerado, assim como é necessário que as posições inicial e final sejam conhecidas.

Neste capítulo tem-se as referências teóricas em que o trabalho se baseia assim como a revisão da literatura para a criação de um algoritmo genético capaz de resolver o problema acima aliado à curvas de Bézier.

## 2.1 Robô diferencial da categoria VSSS

O robô utilizado neste trabalho é definido como diferencial, já que a velocidade linear é variada através da mudança de velocidade aplicada em cada uma de suas duas rodas. Esse tipo de robô é comum nas competições da categoria VSSS pelas equipes e em muitas outras aplicações. Pode-se considerar um robô diferencial típico de duas rodas em um quadro de coordenada global como mostrado na Figura 3.

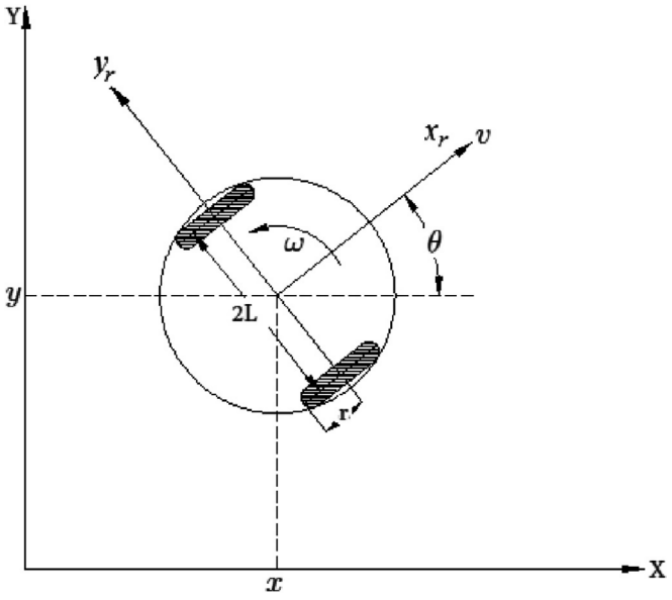


Figura 3 – Robô diferencial e suas variáveis (SIMBA; UCHIYAMA; SANO, 2016).

As velocidades linear e angular são dadas pela Equação 2.1:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \text{sen}\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{v} \\ \omega \end{bmatrix} \tag{2.1}$$

em que $x$ e $y$ são as duas variáveis de coordenadas dimensionais, $\theta$ representa o ângulo de orientação, e por fim, $v$ e $\omega$ são respectivamente as velocidades linear e angular do robô.

## 2.2 Regras e definições da categoria VSSS

Nesse segmento o trabalho trata de apresentar algumas regras das partidas da categoria de competição VSSS, além de mostrar algumas dimensões dos robôs, bola e campo.

### 2.2.1 Robôs jogadores

Dois times competem, cada um com seus máximos três robôs, geralmente sendo divididos entre atacante, aquele que se dirige na maior parte do tempo na parte ofensiva na tentativa de marcar gols para seu time, defensor, que permanece na maioria do tempo no campo da sua equipe evitando que ela sofra ataque do oponente e goleiro, que é responsável por permanecer de frente ao gol com o objetivo de impedir que a equipe adversária marque o gol.

Segundo Pinto (2021), cada robô possui seu tamanho limitado em até $7,5\ cm\ x\ 7,5\ cm\ x\ 7,5\ cm$, desconsiderando a altura da antena usada para comunicação via rádio. No topo de cada robô deve haver duas cores sendo elas diferentes de laranja, branco, cinza ou preto. Um adesivo amarelo ou azul deve ser designado pelos organizadores definindo a cor de cada time e usado no topo de cada robô a fim de identificá-lo. Na Figura 4 é apresentado o robô da equipe Rodetas Robô Clube.

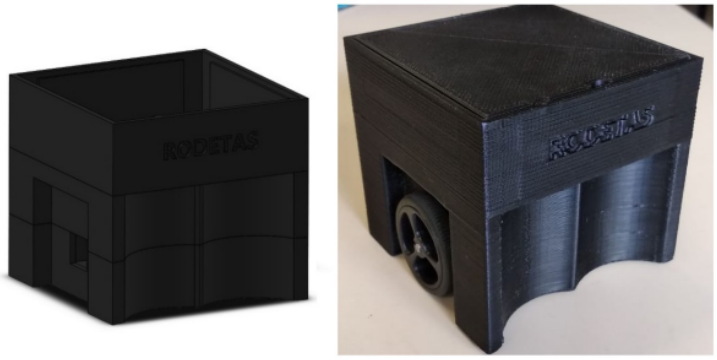


Figura 4 – Robô da Equipe Rodetas Robô Clube.

É permitido a cada robô possuir ou ser equipado com braços ou pernas que não devem sair das restrições de tamanho já apresentadas, mesmo depois desses membros já expandidos.

O robô que estiver na considerada área do gol é definido como goleiro e as demais regras serão aplicadas para esse tipo de jogador. O goleiro, e somente ele, pode segurar a bola somente dentro da área. Nenhum dos demais jogadores pode pegar e segurar a bola se mais de $30\%$ da bola estiver fora do campo de visão superior ou lateral.

Durante as competições cada robô deve conter seu próprio mecanismo de alimentação e conjunto de motores, sendo totalmente independente. Dessa maneira, apenas comunicações sem fio são aceitas para que o robô converse com o computador.

### 2.2.2 O campo, a bola e suas dimensões

O campo utilizado é de cor preta, não reflexiva, feito de madeira de medida $150\ cm\ x\ 130\ cm$ e com textura de uma mesa de ping pong. Além disso paredes com uma altura de 5cm e espessura de $2,5\ cm$. Para evitar que a bola fique presa nos cantos do campo, é adicionado um triângulo de madeira maciça em cada um, com dimensões de $7\ cm\ x\ 7\ cm$.

O gol de cada lado do campo tem uma dimensão de $40\ cm$ de comprimento e $10\ cm$ de profundidade.

As marcações do campo são feitas de acordo com a Figura 5, que mostra as dimensões das áreas demarcadas para cada área do campo.

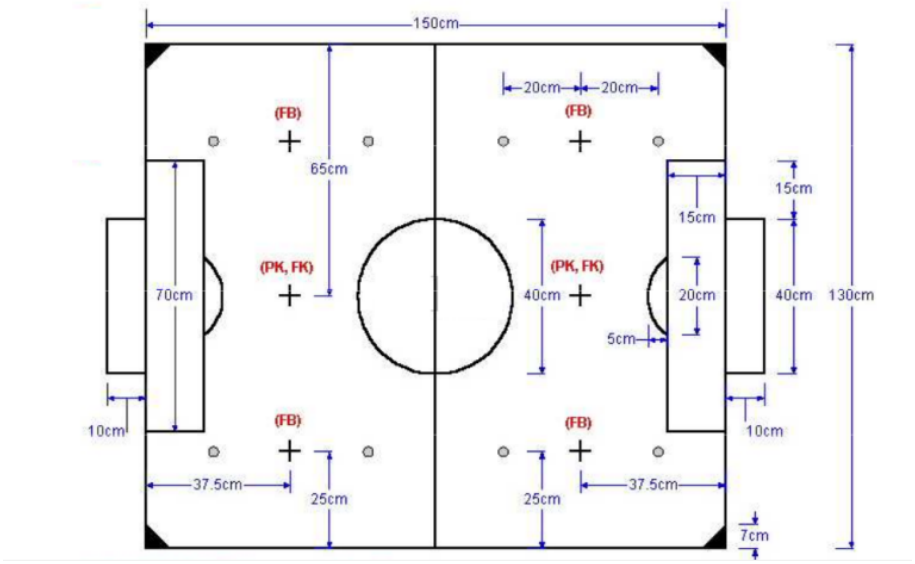


Figura 5 – Campo da categoria VSSS e suas medidas.

Uma bola de golf de cor laranja é usada, com $42,7\ mm$ de diâmetro e aproximadamente $46\ g$ de peso.

### 2.2.3 A partida

A duração de uma partida é de 10 minutos dividida em dois tempos de 5 minutos cada, com um intervalo entre eles de 10 minutos, o que pode ser alterado quando discutido em comum acordo entre os participantes das duas equipes e a organização dependendo das condições e regulamento da competição em questão. Durante substituições, para que um robô seja removido do campo e outro posto em seu lugar, infrações, para que os robôs e a bola sejam reposicionados e intervalos o tempo é pausado.

Antes do início de um jogo é de suma importância que haja a conferência de alguns tópicos para garantir que as regras sejam seguidas por ambas as equipes.

Antes do início da partida uma moeda é lançada ao ar e a partir de seu resultado é definida a cor de cada time ou quem irá começar o jogo com posse da bola. Cada time deve posicionar seus jogadores robôs na posição adequada do campo sendo ele de defesa ou ataque.

A partir de um sinal do árbitro a partida é iniciada e os robôs poderão se movimentar livremente pelo campo seguindo a estratégia definida pela equipe. A Figura 6 representa a simulação de uma partida entre dois times no *software* VSS-Viewer.

Figura 6 – Simulação de uma partida no VSS-Viewer.

## 2.3 Algoritmos genéticos

Para Frenzel (1993), um algoritmo genético é um procedimento exploratório que muitas das vezes é capaz de localizar soluções ótimas para problemas complexos e para isso são mantidas uma série de soluções, chamadas de indivíduos, e essas são forçadas a evoluírem até que seja encontrada uma solução aceitável para o problema proposto.

Ao longo dos tempo e ao passar das gerações, as populações evoluíram no meio natural de acordo com os postulados de Darwin(1859), onde os mais fortes sobreviviam em detrimento

dos mais fracos de acordo com seu ambiente e suas expertises. Os AGs imitam esse processo natural criando soluções para o mundo real. A obtenção de respostas a partir da evolução das soluções e de valores considerados ótimos para cada problema levam em conta uma codificação especial para cada uma.

Os AGs segundo Fernandes (2003), na natureza a combinação de boas características provenientes de diferentes indivíduos ancestrais pode, às vezes, produzir descendentes "super-indivíduos", cuja adaptação é muito maior que a de seus ancestrais. Sendo assim, as espécies evoluem, deixando para seus descendentes características cada vez melhores para que esses consigam garantir a sobrevivência no ambiente em que vivem.

Cada indivíduo possui um grau de aptidão que define o quão apto e bom ele é levando tópicos de análise em conta. Quanto maior esse grau, melhor adaptado e mais propenso a transmitir boas características para seus descendentes ele é, assim como maior é sua chance de ser selecionado para reprodução, cruzando seu material genético com outro indivíduo semelhante em grau de aptidão. A partir desse cruzamento, novos indivíduos são gerados, esses com material genético de ambos os pais e com uma grande chance de se tornarem melhores que os indivíduos da geração anterior a suas.

Outra operação genética comumente feita é a mutação, onde, de acordo com uma taxa pré determinada, indivíduos são escolhidos para terem, outra vez, de acordo com uma taxa, seus cromossomos alterados, buscando maior grau de aptidão, ou *fitness*.

Os indivíduos com menor ou quase nenhum grau de aptidão avaliado serão descartados ou não serão escolhidos para se reproduzirem. Dessa forma, ao longo das gerações somente as melhores características serão repassadas para os indivíduos melhorando o grau de aptidão e gerando cada vez mais "superindivíduos". Se bem avaliado, pensado e projetado um AG aplicado a um problema real tem altas chances de resultar em um ótimo local e consequentemente gerar soluções para o tal problema.

O sucesso na utilização de AGs vem da facilidade em manipular os parâmetros e regras para obtenção de ótimos assim como a possibilidade de aplicação em diversas áreas, incluindo aquelas que outros métodos não conseguem ou tem dificuldades para resolver. No caso de técnicas ou algoritmos já especializados na resolução de algum tipo de problema, o AG provavelmente será superado na questão de tempo de execução e eficácia, sendo essa uma de suas desvantagens.

Em resumo, os AGs tratam problemas de otimização como um processo em busca da melhor solução dentro do possível conjunto de respostas desses mesmos problemas. Inicia-se então uma espaço de indivíduos que consistem na população inicial. Combinando os melhores e mais aptos indivíduos dessa população obtém-se a próxima geração, com indivíduos ainda melhores que substitui a anterior. Esse ciclo é repetido, sendo que a cada repetição a população é refinada gerando novos e cada vez melhores indivíduos, e, consequentemente, novas e melhores soluções para o problema em questão, culminando com a sua convergência.

A Figura 7 apresenta um fluxograma exemplificando esse processo de repetição citado acima.

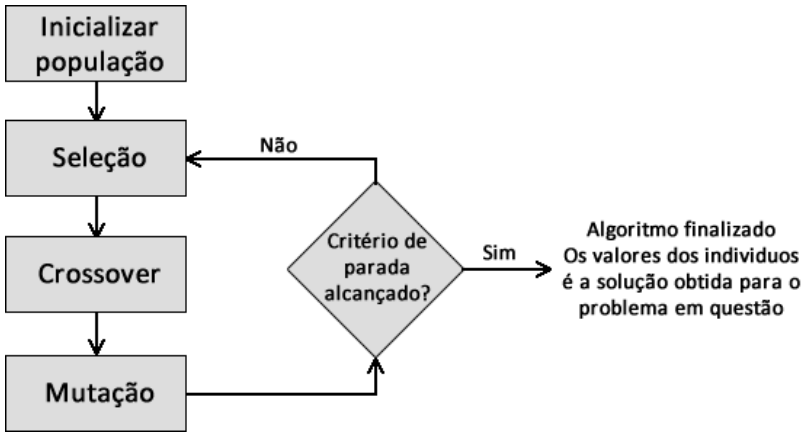


Figura 7 – Fluxograma de um Algoritmo Genético.

### 2.3.1 Operadores genéticos

Alguns operadores genéticos são mais conhecidos, como o de mutação, o de reprodução e o de cruzamento (*crossover*). Para este trabalho foram usados os operadores de mutação de *crossover*, pois se aplicavam melhor ao problema apresentado, gerando soluções satisfatórias.

Para Medeiros (2005), cruzamento, ou *crossover*, é um operador baseado na troca de partes dos cromossomos dos indivíduos uma vez selecionados para tal operação, pais, formando-se duas novas soluções, filhos. Este processo pode ser observado no exemplo a seguir na Figura 8, onde a solução está codificada com alfabeto binário.

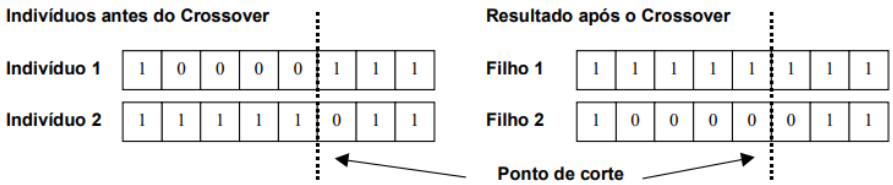


Figura 8 – Representação *crossover*.

Para realizar o cruzamento, primeiro é necessária a escolha, por sorteio, dos cromossomos $c$. Em seguida ocorre a realização ou não do cruzamento segundo um parâmetro, denominado taxa de cruzamento. Deste modo, de acordo com a taxa de cruzamento, pode ocorrer que os cromossomos $c$ sejam repassados sem modificação para a geração seguinte, criando filhos $f$ idênticos a eles. A intenção maior do operador de *crossover* é explorar o material genético presente na população.

O operador de *crossover* apresenta algumas variações que merecem destaque, já que é de fundamental importância para os AGs. Em Pacheco (1999) são descritas as variações: a primeira pode ser chamada de *Crossover de Dois Pontos*, que executa a recombinação de dois indivíduos a partir de dois pontos escolhidos aleatoriamente. Este operador é capaz de combinar schemata com posições fixas nas extremidades, como no exemplo da Figura 9.

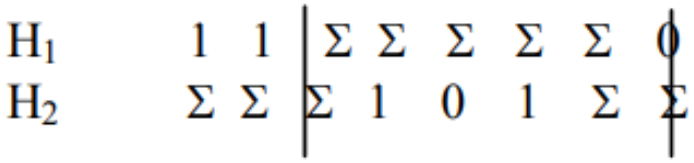


Figura 9 – Representação de *Crossover de Dois Pontos* (PACHECO, 1999).

A segunda variação é chamada de *Crossover Uniforme* e é capaz de recombinar quaisquer posições entre dois genitores. Este operador utiliza um padrão (palavra binária) escolhido aleatoriamente para designar os bits selecionados em cada indivíduo $G$ na criação dos filhos $D$. A Figura 10 representa essa seleção e resultado da combinação.

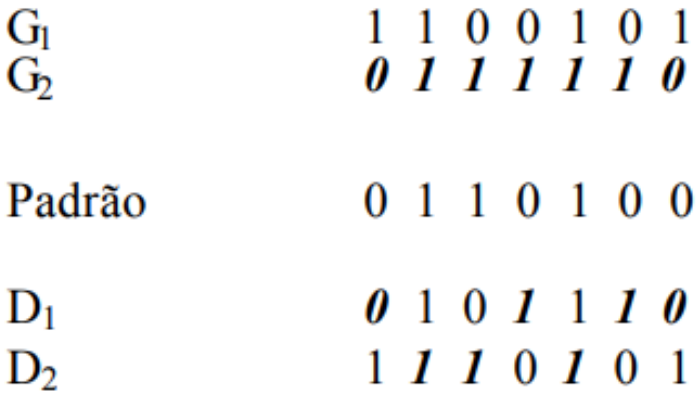


Figura 10 – Representação de *Crossover Uniforme* (PACHECO, 1999).

Outro operador genético é o de mutação, que é responsável pela inserção de pequenas mudanças aleatórias nos cromossomos dos filhos. Como no caso dos operadores de cruzamento, existem vários tipos de operadores de mutação, como Troca de valores, Aritmética e Troca de posições. Porém neste trabalho explora-se o funcionamento do operador de mutação do tipo troca de valores.

No exemplo do trabalho de Burdelis (2009), o operador de mutação por "Troca de Valores", uma ou mais posições do indivíduo são escolhidas aleatoriamente, e os seus valores são trocados por outros valores pertencentes ao universo de discurso do problema. Por exemplo: Seja o universo $U$ dado por: $U = \{x\ E\ N\}$, e um indivíduo $i$ dado pelo seguinte vetor: $i$ = <1, (2),

3, 4, 5, 6>, (onde o número entre parênteses representa a posição selecionada para a mutação), então após a mutação, o novo indivíduo $i'$ será dado, por exemplo, por: $i'$ = <1, (27), 3, 4, 5, 6>.

Por fim, o operador genético de reprodução, para Medeiros (2005), refere-se ao processo de selecionar e copiar um determinado cromossomo para a população seguinte de acordo com sua aptidão. Isto significa que os cromossomos mais aptos têm maior probabilidade de contribuir para a formação de um ou mais indivíduos da população seguinte. Existem basicamente os seguintes métodos: troca de toda população, troca de toda população com elitismo, onde todos os cromossomos são substituídos, sendo o cromossomo mais apto da população corrente copiado para população seguinte, e troca parcial da população (*steady state*), onde os $M$ melhores indivíduos da população corrente são copiados para população seguinte.

### 2.3.2 Métodos de seleção

Uma das etapas do AG é o processo seleção. Esse processo depende previamente do cálculo da *fitness*, uma vez em posse dessa, é possível determinar quais indivíduos produzirão a próxima geração. A seleção é um processo onde etapas são determinadas em um processo iterativo (BLICKLE; THIELE, 1996).

O processo de seleção é de fundamental importância para o AG, segundo Huang e Wang (2011), pois esse será usado como referência na determinação da qualidade dos cromossomos. Ainda para Shanahan (2000), a melhor geração é a geração que passará pra o próximo método de seleção.

Várias técnicas podem ser usadas no processo de seleção e a melhor será escolhida dependendo do problema abordado. Alguns dos métodos de seleção parental citados no trabalho de Mohebi, Bouyer e Karimi (2009) são:

- Método de seleção da roleta,
- Método de seleção local,
- Método de seleção de corte,
- Método de Seleção por torneio,
- Método de Seleção por escalonamento baseado em *ranking*,
- Método de Seleção por amostragem universal estocástica (SUS).

Para este trabalho o método de seleção escolhido foi o da roleta, já que o problema discutido apresenta valores de *fitness* estáveis e baixos. Nesse método primeiramente escolhe-se uma solução que seja plausível de ser recombinada. A função *fitness* entrega um valor para cada indivíduo proporcional ao seu índice de aptidão, dessa forma, cada indivíduo da população é

representado na roleta por uma fatia proporcional ao seu índice de aptidão com mostra a Figura 11. Nela é apresentado o seguinte exemplo retirado de Herman et al. (2019): 5 indivíduos fazem parte da população, cada um tendo seu valor de *fitness* respectivamente de $f(1) = 0.4, f(2) = 0.1, f(3) = 0.16, f(4) = 0.6$ e $f(5) = 0.5$, tendo assim um valor total de *fitness* de 1.76. Assim, as probabilidades individuais obtidas são $p[1] = 0.23, p[2] = 0.057, p[3] = 0.091, p[4] = 0.34$ e $p[5] = 0.28$.

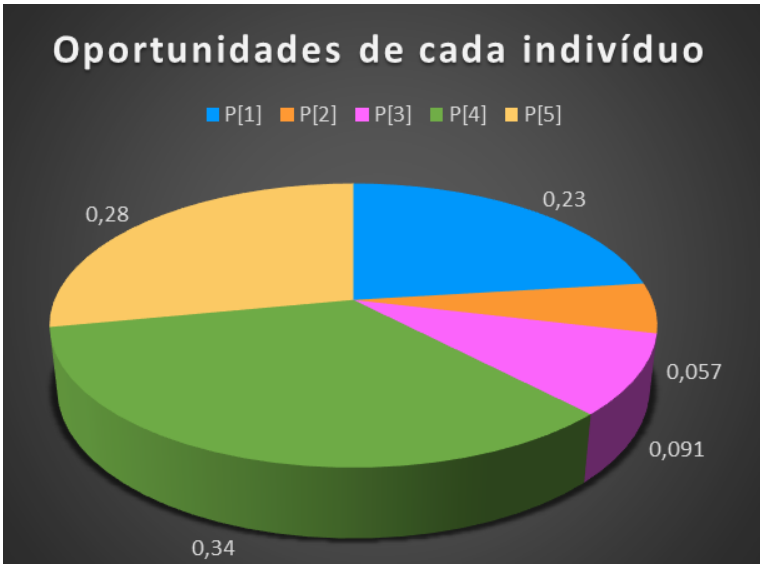


Figura 11 – Representação de seleção pelo método da roleta.

O próximo passo é a geração aleatória de números $[0, 1]$ para escolher então os pais. Indivíduos com fatias maiores tem maior chance de serem escolhidos.

## 2.4 Curvas de Bézier e suas aplicações

As curvas de Bézier foram introduzidas em 1962 pelo engenheiro Pierre Bezier com a intenção de auxiliar no *design* e estrutura de veículos. Atualmente, as curvas, são amplamente usadas em projetos de computação gráfica e animação além de serem comumente utilizadas em aplicações focadas em execução de caminhos. Segundo Farin (1983), a curva de Bézier de grau n pode ser representada como

$$P_{[t_0,t_1]}(t) = \sum_{i=0}^{n} B_i^n(t) P_i \tag{2.2}$$

em que $P_i$ representa os pontos de controle assim como $P(t_0) = P_0$ e $P(t_1) = P_n, B_i^n(t)$ são um polinômio de Bernstein dado por

$$B_i^n(t) = \binom{n}{i} \left(\frac{t_1 - t}{t_1 - t_0}\right)^{n-i} \left(\frac{t - t_0}{t_1 - t_0}\right)^i, i \epsilon \{0, 1, ..., n\} \tag{2.3}$$

As curvas de Bézier possuem propriedades úteis e interessantes quando se trata de planejamento de caminhos. São elas:

- Elas sempre passam pelos pontos $P_0$ e $P_n$;
- Elas sempre permanecem dentro do escopo convexo em que consistem seus pontos de controle;
- Elas sempre tangenciam as linhas que conectam $P_0 \longrightarrow P_1$ e $P_n \longrightarrow P_{n-1}$ em $P_0$ e $P_n$ respectivamente.

A construção de uma curva de Bézier cúbica resultante do polinômio de terceiro grau é ilustrada na Figura 12, tendo pontos auxiliares que variam entre os pontos de controle, existindo pontos $R_0$ e $R_1$ que se movem entre os pontos $Q_0$, $Q_1$, e $Q_2$ e finalmente o ponto $B$ da curva resultante varia entre $R_0$ e $R_1$. Desta forma é possível construir uma curva cúbica a partir dos pontos $P_i$, $Q_i$, e $R_i$ todos variando linearmente em função das posições dos pontos de referência e do parâmetro $t$.

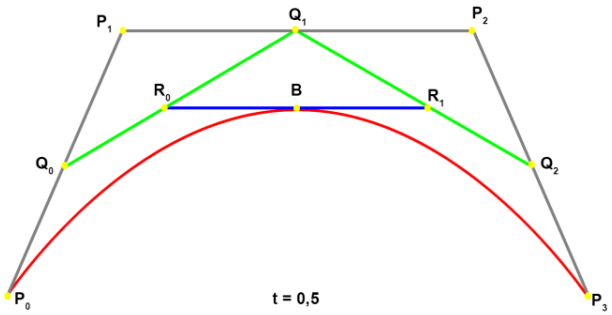


Figura 12 – Construção de uma curva de Bézier de grau 3.

No trabalho de Mohamed, Abdulaziz e Xiaohui (2017) é abordado um tema parecido com o proposto neste trabalho. Um planejamento de caminho baseado em um algoritmo genético é sugerido para trabalhos em cenários dinâmicos chamado de GADPP. Esse método utiliza a curva de Bézier para refinamento do caminho final gerado de acordo com controle de pontos identificados previamente. Nesse caso, para a atualização do caminho durante o movimento o robô recebe um sinal da estação base (BS) baseado em alertas que são periodicamente acionados por sensores. Se comparados ao estado da arte de outros métodos, o GADPP melhora a performance dos robôs testados em termos de tamanho e suavidade do caminho, além de obter a rota mais otimizada. Os resultados obtidos são satisfatórios devido, em grande parte, à aplicação da curva de Bézier, já que essa é a grande responsável pelo processo de suavização obtido no trabalho citado.

Um outro exemplo de aplicação de curva de Bézier encontrado na literatura no ramo da robótica móvel é o apresentado no trabalho de Simba, Uchiyama e Sano (2016), no qual a proposta é a geração de trajetórias suaves para robôs móveis usando curva de Bézier com propriedades específicas para o problema abordado. Semelhante ao proposto neste trabalho, a literatura sugere a adição de pontos de controle, ou auxiliares, para a curva de Bézier a cada mudança de direção, de modo a permitir que os robôs realizem curvas mais suaves.

# 3 DESENVOLVIMENTO

Neste capítulo são mostradas todas as partes do ambiente simulado aplicado no futebol de robô da categoria VSSS assim como ferramentas e o sistema de *software* utilizado pra o desenvolvimento deste trabalho. Encontra-se também nesta seção o desenvolvimento do AG utilizado para a escolha de caminho e a utilização de curvas de Bézier para suavização de rotas geradas.

## 3.1 Ferramentas

Para o desenvolvimento do proposto englobando execução de algoritmos e simulações computacionais foi utilizada uma máquina com as seguintes configurações:

- Processador: Intel(R) Core(TM) i7-7500U CPU 2,70GHz (4 CPUS), 2,9GHz;
- Memória RAM: 8GB;
- GPU: NVIDIA GeForce 940MX 2GB de memória de vídeo.

## 3.2 IEEE Very Small Size Soccer (VSSS)

Como dito anteriormente no primeiro capítulo, o desenvolvimento do trabalho apresentado se baseia no ambiente VSSS. Com alta dinamicidade e velocidades altas, o VSSS é muito desafiador e complexo, o que habilita as diversas possibilidades de aplicação de técnicas e abordagens para as áreas do projeto Rodetas que são: sistema embarcado que envolve *hardware* e firmware e o sistema de processamento externo, que é aprofundado a seguir.

### 3.2.1 Sistema de processamento externo - *VSS-SDK*

O *VSS-SDK* é um projeto *open source* que possui todos os componentes e aplicações necessárias para simular uma partida de futebol de Robôs focado na categoria *IEEE Very Small Size Soccer* presente na Competição Brasileira de Robótica (CBR) e na Competição Latino-Americana de Robótica (LARC). Criado e desenvolvido em C++, tem o intuito principal de auxiliar equipes iniciantes para que possam participar das competições na categoria. É dividido em três módulos, que são descritos nas subseções abaixo, a fim de facilitar a utilização e instalação do projeto.

### 3.2.2 Módulo *VSS-Simulator*

O *VSS-Simulator* é o simulador de partidas de futebol para a categoria *IEEE Very Small Size Soccer*. Utiliza da biblioteca *Bullet Physics* apra projetar um universo $3D$, onde simula de

forma realista colisões fazendo com que os objetos tenham restrições cinemáticas e dinâmicas. O *VSS-Simulator* e o *VSS-Vision* possuem o mesmo papel de enviar pacotes de estados em tempo real.

### 3.2.3 Módulo *VSS-Viewer*

O módulo *VSS-Viewer* módulo é um visualizador de estados de jogo, que também mostra as informações de *debug*. O projeto permite que o usuário inicie e pause uma partida no *VSS-Simulator* com a possibilidade de mudar a posição dos robôs e da bola. Fortemente ligado ao *VSS-Simulator*, o *VSS-Viewer* é responsável por mandar um pacote de controle e ao *VSS-Simulator* e somente após isso é iniciada a partida. Para o envio do pacote basta que a tecla de espaço seja pressionada no *VSS-Viewer*. O *VSS-Viewer* recebe os pacotes de estado de jogo, ouvindo as mensagens do *VSS-Simulator* e do *VSS-Vision*.

### 3.2.4 Módulo *VSS-Vision*

O Modulo *VSS-Vision* é o responsável por capturar coordenadas gráficas baseadas em um plano cartesiano dos robôs a partir da imagem da câmera. Utiliza a biblioteca de processamento de imagem *open source OpenCV*. O *VSS-Vision* está dividido em dois módulos, sendo um de calibragem, onde é possível calibrar cores, cortar e rotacionar a imagem e um outro, o módulo de jogo, onde seta-se o padrão de cada robô, assim como a cor do time. A definição das posições dos robôs são realizadas a partir do rastreamento das cores calibradas. Ambos os módulos possuem a opção de trabalhar com imagens ou com uma câmera.

A maior função do *VSS-Vision* é verificar se a leitura e captura de cores estão sendo feitas de maneira correta, para que as estratégias funcionem com os robôs corretos e suas devidas posições.

Todos os módulos citados anteriormente poodem ser encontrados no *VSS-SDK* que é um projeto *open source* e a equipe Rodetas teve grandes contribuições no desenvolvimento do mesmo. O projeto pode ser acessado no *Github* através do link https://vss-sdk.github.io/book/general.html.

## 3.3 Arquitetura de *software* da equipe Rodetas

A equipe Rodetas faz uso do *Suite Development Kit* denominado *VSS-SDK*, citado na Seção 3.2.1, que vem sendo desenvolvido, com a ajuda da própria equipe, especificamente para a categoria *IEEE Very Small Size Soccer*. Esse *Kit* fornece um módulo de visão computacional (Módulo *VSS-Vision*), um módulo para simulação dos robôs (Módulo *VSS-Simulator*) e um módulo para visualização de posições e *debug* (Módulo *VSS-Viewer*).

A utilização do *SDK* traz grandes facilidades para a equipe em relação à área de visão computacional, evitando o desenvolvimento de algoritmos elaborados, portanto cabe à equipe

apenas construir os algoritmos de controle, movimentação dos robôs e estratégia de jogo, sendo assim possível focar no desenvolvimento.

### 3.3.1 Estratégia de jogo

A estratégia de jogo da equipe Rodetas é bastante precisa como pode-se ver no trabalho de Santos (2019):

> "O código da estratégia é implementado em C++ com os paradigmas de orientação a objeto. O desenvolvimento do algoritmo permite em tempo real a troca de funções entre os jogadores, dando desta forma maior dinâmica ao jogo. As estratégias criadas estão diretamente ligadas as posições dos jogadores e da bola no campo, criando-se condições e estados que o programa identifica e gerencia de acordo com o pré estabelecido no código. Deste modo gera-se uma certa inteligência do computador na tomada de decisões durante a partida.
>
> Neste trabalho, a estratégia de jogo tem o princípio de dar a cada robô um comportamento de: atacante, ou defensor ou goleiro. O atacante sempre buscará a bola quando essa estiver do meio do campo para frente, a ideia é que o atacante chegue sempre na bola com sua orientação voltada para o centro do gol, objetivando levá-la até o gol e marcando o ponto. O defensor tem o papel de se posicionar na linha do meio do campo, acompanhando a bola verticalmente, e maneira que quando a bola chega até ele, o mesmo faz um giro em torno do próprio eixo expelindo a bola para frente. Quando a bola está atrás do meio do campo, o defensor busca a bola, assim como o atacante, com uma orientação de chegada mirando o gol adversário. Nesse caso, quando o defensor chega até a bola ele se torna um atacante e o atacante anterior se torna um defensor. Por fim, o goleiro apenas fica no gol, protegendo e expelindo a bola da área de gol, procurando sempre barrar o ataque dos adversários".

### 3.3.2 Movimentação

A movimentação dos robôs no campo é crucial para o bom desempenho do time. Um ponto importante da movimentação é o planejamento da trajetória que o robô deve executar nas diferentes situações de jogo. Por ser o tópico principal abordado neste trabalho, o planejamento de rotas, assim como a movimentação do robô pelo campo, serão discutidos nas seções que se seguem, 3.4 e 3.5.

## 3.4 Algoritmos genéticos no planejamento de rotas

Para o planejamento de caminhos dos robôs diferenciais da equipe Rodetas em campo este trabalho utiliza de uma combinação de técnicas envolvendo algoritmos genéticos e curvas de Bézier. Nesta seção será discorrido sobre as etapas de implementação do Algoritmo genético aplicado ao traçado de caminho.

Definidos previamente pela estratégia, os pontos iniciais e finais de uma rota são enviados para o módulo de movimentação. Este, por sua vez, tem a responsabilidade de determinar o melhor percurso entre esses dois pontos e a maneira como o robô o fará. Para a execução dessa tarefa o seguinte algoritmo genético foi definido com as seguintes características.

### 3.4.1 Representação do ambiente

O ambiente é o espaço onde tanto o robô quanto os obstáculos coexistem. Para este trabalho foi utilizada a abordagem de Song, Wang e Sheng (2016), que propõe a divisão do campo em células de $10\ cm\ x\ 10\ cm$ que compõem um *grid* de tamanho *M x N*, sendo *M* e *N* dependentes do tamanho total do ambiente. Esta abordagem de representação simplifica a codificação dos indivíduos do problema, uma vez que qualquer posição do campo pode ser representada por um único número. Tomando como exemplo um *grid* de tamanho $10\ x\ 10$, tem-se as células representadas por números de $0$ a $99$. Um *grid* de tamanho $15\ x\ 13$ foi definido, já que o campo da modalidade VSSS, na qual a equipe Rodetas compete, possui dimensões de $150\ cm\ x\ 130\ cm$. Com isso, qualquer posição do campo pode ser representada por um número entre $0$ e $194$ ($195$ células), como pode ser visto na Figura 13.

| Grid de um campo 15x13 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 |
| 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 |
| 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 |
| 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 |
| 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 |

Figura 13 – *Grid 15 x 13*.

Além disso, a Figura 14, mostra a demarcação do *grid* sobre campo. A Figura 15 traz o *grid* que representa a Figura 14 com os robôs como obstáculos e a bola como objetivo. O conteúdo de cada célula representa se ela está ocupada ou não, sendo $0$ a representação da célula vazia e $1$ a representação da célula ocupada, ou seja, há um obstáculo naquela posição. Ainda, a célula definida como objetivo possui o valor $2$ em seu conteúdo.

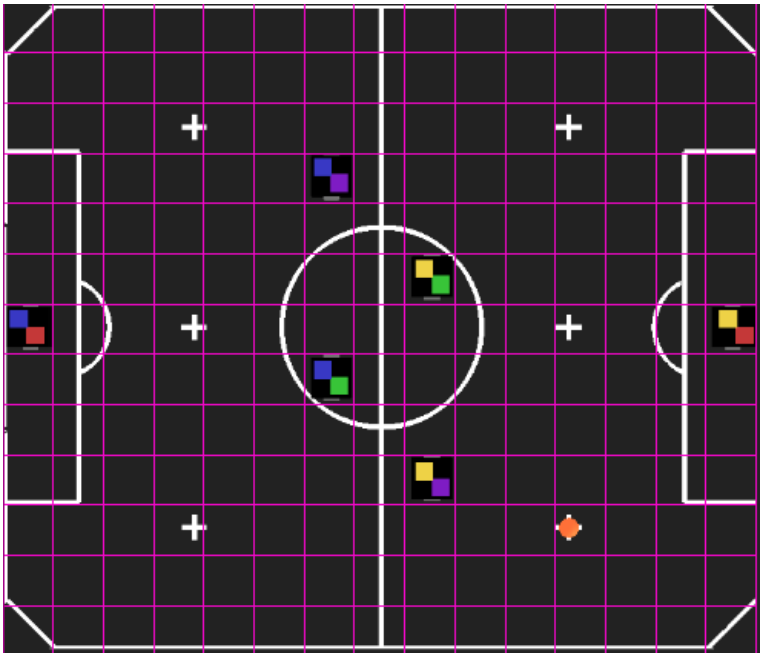

Figura 14 – Demarcação do *grid* sobre o campo.


Figura 15 – Campo representado como *grid* com células demarcadas.

### 3.4.2 Definição dos indivíduos

Tendo como base o *grid* definido na Seção 3.4.1, pode-se estabelecer o indivíduo do problema a ser resolvido. Cada indivíduo é representado por um caminho partindo do ponto inicial, que é a posição inicial do robô, e finalizando no objetivo recebido pelo módulo de estratégia. Dessa forma, cada um deles é composto por uma sequência de números do *grid*

demarcando cada célula a ser percorrida pelo robô durante seu trajeto. Por sua vez, cada célula dessa sequência representa um cromossomo do indivíduo. Na Figura 16 pode ser visto um caminho que se inicia na célula de número 48 e termina na célula de número 70, passando pelas células 49, 50, 51, 52, 53, 54 e 55. Assim, este indivíduo é representado como:

$$i = [48, 49, 50, 51, 52, 53, 54, 55, 70] \tag{3.1}$$

| Grid de um campo 15x13 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 |
| 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 |
| 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 |
| 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 |
| 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 |

Figura 16 – Representação de uma rota a ser percorrida.

Como o *VSS-Viewer* entrega para o código as coordenadas do robô e da bola no formato cartesiano é necessário converter essas coordenadas para o formato de representação adotado pelo *grid*. Esta conversão é denotada por:

$$celula = \lfloor \frac{P_x}{10} \rfloor + (\lfloor \frac{P_y}{10} \rfloor \times 15) - 1 \tag{3.2}$$

em que $P_x$ e $P_y$, em $cm$, representam as coordenadas cartesianas X e Y do ponto a ser convertido.

Em contrapartida, o módulo de controle necessita que o caminho traçado esteja em coordenadas cartesianas para que seja possível conduzir de maneira adequada o robô. Para tal, é preciso que o número de uma célula do *grid* seja convertido de volta para a forma cartesiana. Esta conversão é denotada por:

$$P_x = (celula\%15) \times 10 + 5 \tag{3.3}$$

$$P_y = \lfloor \frac{2}{3} \times (1 + celula - \frac{P_x}{10}) \rfloor \tag{3.4}$$

em que $P_x$ e $P_y$, em $cm$, representam as coordenadas cartesianas $X$ e $Y$ do ponto já convertido. Vale ressaltar que ao converter um número de uma célula para coordenadas cartesianas obtém-se o ponto central dessa célula, podendo assim haver truncamentos que não impactam negativamente o desenvolvimento deste trabalho, uma vez que o caminho traçado é suavizado corrigindo quaisquer desvios.

### 3.4.3 Definição do algoritmo genético

Este trabalho aborda um algoritmo genético envolvendo as etapas de:

- Definição de parâmetros do algoritmo;
- Inicialização da população;
- Classificação e ordenação da população inicial (cálculo de *fitness*);
- Seleção de indivíduos a serem mutados e seleção de pais para cruzamento (*wheel selection*);
- Operação genética de mutação;
- Operação genética de cruzamento;
- Classificação por elitismo.

O pseudocódigo das etapas apresentadas é exibido no algoritmo 1, que é detalhado no decorrer do trabalho.

**Algorithm 1** Estrutura geral do algoritmo genético

**Entrada:** Posição do robô, objetivo, representação do campo em forma de *grid*
**Saída:** Caminho representado por um vetor de células do *grid*

```
Definição de parâmetros
Inicialização da população
i ← 0
enquanto i ≤ Número de gerações & !Critério de parada faça
    Calcula fitness da população atual e ordena ascendentemente
    Realiza crossover
    Realiza mutação
    Calcula fitness dos novos indivíduos gerados
    Remove piores indivíduos
fim enquanto
Retorna melhor indivíduo da população final
```

#### 3.4.3.1 Definição de parâmetros do algoritmo genético

Como primeira etapa de um algoritmo genético é importante definir os parâmetros que serão utilizados de maneira precisa, de forma que esses colaborem para a convergência da solução da forma mais otimizada.

Para definição dos parâmetros de taxa de *crossover*, mutação e tamanho da população este trabalho utiliza uma abordagem que foca em executar múltiplas vezes o mesmo algoritmo variando os parâmetros e verificando qual deles fornece melhor desempenho. A apresentação detalhada dessa etapa se encontra no capítulo 4 na seção 4.1.

Visando um melhor tempo de execução do AG, ou seja, menor, foram utilizados alguns critérios de parada para o algoritmo. O primeiro deles, determina um número máximo de gerações. Caso o algoritmo chegue a $100$ gerações o melhor caminho atual é retornado e ele não prossegue com a execução. O segundo critério, que é o critério de convergência, considera que caso o melhor caminho de uma geração seja menor ou igual a $120\%$ da distância em linha reta da posição inicial até o objetivo o algoritmo cessa sua execução retornando esse indivíduo. A equação 3.5 apresenta como o cálculo da menor distância entre dois pontos, $A$ e $B$, dentro do campo é efetuado e a Figura 17 mostra a representação gráfica do cálculo utilizado, sendo a linha na cor verde a menor distância e as linhas na cor vermelha, as distâncias nos eixos $x$ e $y$ entre os mesmos pontos.

$$dist = \sqrt{D_x^2 + D_y^2} \tag{3.5}$$

Na Equação 3.5, $D_x$ representa, em $cm$, a distância de A para B no eixo x, $D_y$ a distância, em $cm$, de A para B no eixo y e $dist$, também em $cm$, a menor distância entre esses pontos.

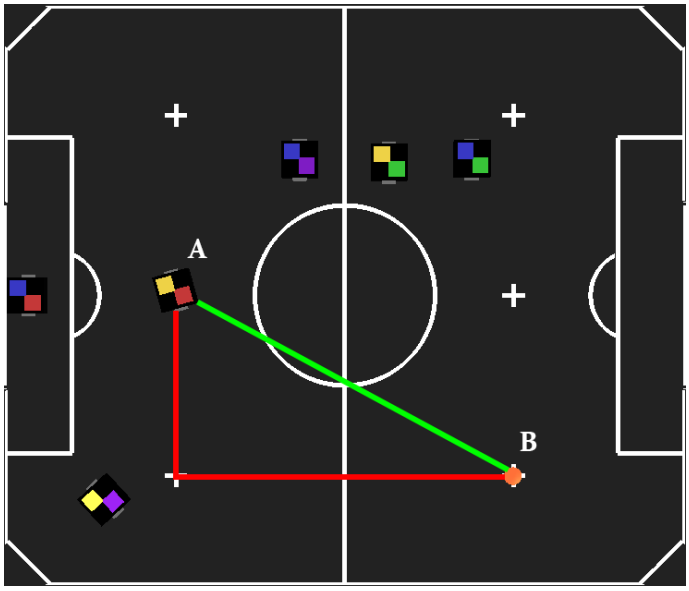


Figura 17 – Representação da menor distância entre o ponto A e B.

#### 3.4.3.2 Inicialização da população

Em posse da posição inicial e do objetivo a ser atingido são gerados aleatoriamente caminhos válidos, ou seja, indivíduos. Para garantir a validade desses caminhos parte-se da célula inicial, que representa a posição atual do robô, confere-se todas as células vizinhas desta que não estejam ocupadas e sorteia-se uma das opções. A opção sorteada é adicionada ao vetor que representa o caminho. Faz-se isso até que o objetivo seja alcançado, gerando assim, um caminho válido e contínuo.

Alguns critérios foram adotados para garantir que *loops* infinitos não aconteçam, que o objetivo seja sempre alcançado e que alguns indivíduos sejam descartados previamente por invalidez:

- Para assegurar que nenhum indivíduo demore muito a atingir seu objetivo, qualquer um que ultrapasse o tamanho de 100 células é descartado;
- Células duplicadas no vetor que representa o caminho não são aceitas;
- Caso uma célula esteja rodeada de células já visitadas o caminho torna-se inviável. Para contornar essa condição exclui-se a célula atual e sorteia-se outra respeitando o critério apresentado anteriormente de vizinhos válidos;
- Se durante a construção de um caminho ocorrerem mais de 200 movimentos, sejam esses de inserção ou remoção de células, o indivíduo é descartado.

O pseudocódigo da geração da população inicial é apresentado no algoritmo 2 a seguir.

#### 3.4.3.3 Cálculo e definição de regras da *fitness*

Para avaliar cada indivíduo esse trabalho propõe a criação de uma função capaz de mensurar seu desempenho diante do objetivo estabelecido. Cada indivíduo é iniciado com 1000 pontos de *fitness* e esse valor é decrementado de acordo com as seguintes condições:

- Para cada célula visitada pelo caminho 4 pontos são descontados, para que assim, caminhos menores, ou seja, mais rápidos, sejam valorizados em detrimento de caminhos maiores e mais longos;
- Caso a célula visitada esteja ocupada, 6 pontos são perdidos. Dessa forma, a penalidade é aplicada a caminhos que contenham obstáculos, evitando colisões.

Após a avaliação de todos os indivíduos a população é ordenada de forma ascendente. Com isso, o algoritmo 3 descreve a forma adotada para cálculo da *fitness*.

**Algorithm 2** Inicialização de indivíduos

**Entrada:** Posição do robô, objetivo
**Saída:** Grupo de indivíduos

Inicializa vetor de indivíduos vazio como $popInicial$
$i \leftarrow 0$
**enquanto** $i \leq$ Tamanho da população **faça**
    $j \leftarrow 0$
    $fimDoCaminho \leftarrow falso$
    $deletaCaminho \leftarrow falso$
    $numeroDeMovimentos \leftarrow 0$
    Inicializa vetor $V$ vazio para representar indivíduo atual
    Adiciona posição inicial à $V$
    **enquanto** $fimDoCaminho = falso$ **faça**
        Obtém vizinhos da última posição de $V$
        **se** Encontrou vizinhos **então**
            Escolhe aleatoriamente um dos vizinhos encontrados
            Adiciona a célula sorteada à $V$
        **senão**
            Remove a última posição de $V$
        **fim se**
        **se** Última posição de $V$ = Objetivo **então**
            $fimDoCaminho \leftarrow verdadeiro$
        **fim se**
        **se** Tamanho de $V \geq 100$ || $numeroDeMovimentos \geq 200$ **então**
            $fimDoCaminho \leftarrow verdadeiro$
            $deletaCaminho \leftarrow verdadeiro$
            $i \leftarrow i - 1$
        **fim se**
        **se** $deletaCaminho = falso$ **então**
            Adiciona $v$ à $popInicial$
        **fim se**
    **fim enquanto**
**fim enquanto**
Retorna $popInicial$

**Algorithm 3** Cálculo da *fitness*

**Entrada:** Representação do campo em forma de *grid*
**Saída:** Grupo de indivíduos ordenados ascendentemente pela *fitness*

1: **para** Cada indivíduo $i$ da população **faça**
2: $totalFitness \leftarrow 1000$
3: **para** Cada célula $c$ de $i$ **faça**
4: $totalFitness \leftarrow totalFitness - 4$
5: **se** $grid[c] = 1$ **então**
6: $totalFitness \leftarrow totalFitness - 6$
7: **fim se**
8: **fim para**
9: $fitness[i] = totalFitness$
10: **fim para**
11: Ordena população ascendentemente baseada na *fitness*
12: Retorna população ordenada

#### 3.4.3.4 *Crossover*

Visando uma maior diversidade da população e a garantia de que os melhores indivíduos possam se tornar ainda melhores o operador genético de *crossover* foi proposto como no algoritmo 4. Dois indivíduos da população, que tenham no mínimo uma célula em comum, são selecionados através do método da roleta, citado na seção Algoritmos genéticos. De acordo com os parâmetros previamente definidos na seção Definição de parâmetros do algoritmo genético tem-se a quantidade de novos indivíduos que serão gerados nesse método, e para cada novo indivíduo dois novos pais são selecionados.

#### 3.4.3.5 Mutação

Outra maneira de diversificar a população é através da mutação. Por ela, inserimos novas informações aos cromossomos de um indivíduo também de acordo com parâmetros pré determinados. Para este trabalho a operação de mutação realizada é específica frente ao problema apresentado, uma vez que a partir de um cromossomo sorteado do indivíduo substitui-se todos os outros cromossomos subsequentes. O algoritmo 5 explicita a mutação proposta.

#### 3.4.3.6 Elitismo

Após as operações genéticas citadas anteriormente, a população é novamente avaliada. Como novos indivíduos foram adicionados, o tamanho da população excede o parâmetro definido na seção Definição de parâmetros do algoritmo genético. Para que a população retome o seu tamanho correto os piores indivíduos da geração atual, de acordo com a *fitness*, são removidos, dando início assim à próxima geração.

**Algorithm 4** *Crossover*

**Saída:** Adição de novos indivíduos na população

1: $totalCrossover \leftarrow tamanhoDaPopulacao * taxaDeCrossover$
2: $i \leftarrow 0$
3: **enquanto** $i \leq totalCrossover$ **faça**
4: Seleciona primeiro pai pelo *método da roleta*
5: Seleciona um cromossomo $m$ do primeiro pai de forma aleatória
6: $encontrouSegundoPai \leftarrow falso$
7: $j \leftarrow 0$
8: **enquanto** $encontrouSegundoPai = falso$ **faça**
9: Seleciona segundo pai pelo *método da roleta*
10: **se** Segundo pai possuir o cromossomo $m$ **então**
11: $primeiraMetade \leftarrow$ células anteriores ao cromossomo $m$ do primeiro pai
12: $segundaMetade \leftarrow$ células posteriores ao cromossomo $m$ do segundo pai
13: $novoIndividuo \leftarrow primeiraMetade + segundaMetade$
14: Adiciona $novoIndividuo$ à população
15: $encontrouSegundoPai \leftarrow verdadeiro$
16: **senão**
17: Descarta segundo pai
18: $j \leftarrow j + 1$
19: **se** $j \geq 4$ **então**
20: $encontrouSegundoPai \leftarrow verdadeiro$
21: $i \leftarrow i - 1$
22: **fim se**
23: **fim se**
24: **fim enquanto**
25: **fim enquanto**

**Algorithm 5** Mutação

**Entrada:** Objetivo
**Saída:** Adição de novos indivíduos na população

1: $totalMutacoes \leftarrow tamanhoDaPopulacao * taxaDeMutacao$
2: $i \leftarrow 0$
3: **enquanto** $i \leq totalMutacoes$ **faça**
4: Seleciona individuo $V$ pelo *método da roleta*
5: Seleciona um cromossomo $m$ de $V$ de forma aleatória
6: Exclui todos os cromossomos a partir de $m$ até o final de $V$
7: $j \leftarrow 0$
8: $fimDoCaminho \leftarrow falso$
9: $deletaCaminho \leftarrow falso$
10: $numeroDeMovimentos \leftarrow 0$
11: **enquanto** $fimDoCaminho = falso$ **faça**
12: Obtém vizinhos da última posição de $V$
13: **se** Encontrou vizinhos **então**
14: Escolhe aleatoriamente um dos vizinhos encontrados
15: Adiciona a célula sorteada à $V$
16: **senão**
17: Remove a última posição de $V$
18: **fim se**
19: **se** Última posição de $V$ = Objetivo **então**
20: $fimDoCaminho \leftarrow verdadeiro$
21: **fim se**
22: **se** Tamanho de $V \geq 100$ || $numeroDeMovimentos \geq 200$ **então**
23: $fimDoCaminho \leftarrow verdadeiro$
24: $deletaCaminho \leftarrow verdadeiro$
25: $i \leftarrow i - 1$
26: **fim se**
27: **se** $deletaCaminho = falso$ **então**
28: Adiciona $v$ modificado à população
29: **fim se**
30: **fim enquanto**
31: **fim enquanto**

## 3.5 Curva de Bézier na suavização de rotas curvilíneas

A fim de suavizar o caminho gerado pelo algoritmo genético proposto neste trabalho, o cálculo da curva de Bézier aplicada é apresentada pelo algoritmo 6. Inicialmente verifica-se todos os pontos onde ocorre mudança de direção no caminho recebido. Cada ponto em que essa mudança acontece é considerado um ponto auxiliar e é adicionado a um vetor. Após detectar todas as mudanças a curva de Bézier de grau $n$ igual ao número de pontos auxiliares é calculada e retornada.

**Algorithm 6** Definição da curva de Bézier

**Entrada:** Caminho gerado pelo algoritmo genético
**Saída:** Curva de Bézier do caminho recebido

1: $pontosAuxiliares \leftarrow []$
2: **para** Para cada célula do caminho **faça**
3: **se** Verifica se houve mudança de direção **então**
4: Adiciona célula à $pontosAuxiliares$
5: **fim se**
6: **fim para**
7: $n \leftarrow$ tamanho de $pontosAuxiliares$
8: Gera Curva de Bézier de grau $n$ a partir de $pontosAuxiliares$
9: Retorna a curva gerada

# 4 RESULTADOS

Para avaliar o desempenho do planejador e das técnicas aplicadas foram feitos vários experimentos com diversas situações simuladas.

## 4.1 Avaliação de parâmetros

Inicialmente, a fim de identificar o melhor tamanho da população inicial, tendo como parâmetro de avaliação o tempo de execução e *fitness*, foi realizado um teste da seguinte forma: as taxas de mutação e *crossover* foram fixadas em respectivamente, $0,15$ e $0,8$, e o algoritmo genético foi executado um total de $18$ vezes, variando em cada execução o tamanho da população inicial, partindo de uma população com $10$ indivíduos e incrementando este valor em $5$. As Figuras 18 e 19 apresentam os resultados obtidos.

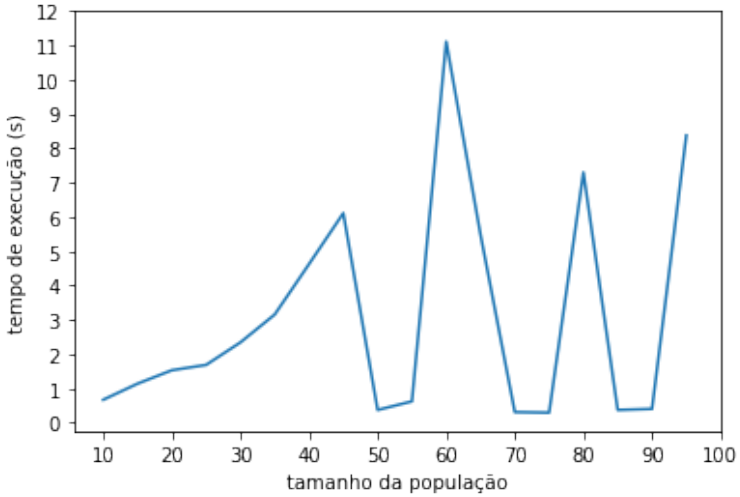


Figura 18 – Avaliação do tamanho população inicial x tempo de execução do AG.

Na Figura 18 tem-se o gráfico resultante da comparação entre tamanho da população e tempo de execução do algoritmo. A irregularidade do gráfico pode ser explicada devido ao fato de que nem sempre grandes números de indivíduos na população levam a um grande tempo de execução, já que uma vez atingida a configuração ideal, a convergência do AG é mais rápida. Na Figura 19 tem-se, por sua vez, o gráfico resultante da comparação entre tamanho da população e *fitness* de cada indivíduo.

Com isso, após a análise dos gráficos gerados, pode-se extrair o valor ideal para o parâmetro *tamanho da população*: $50$. Pela Figura 18 percebe-se que existem alguns valores que fazem o AG executar com baixo tempo como por exemplo os valores $50$, $70$ e $90$. Porém, ao

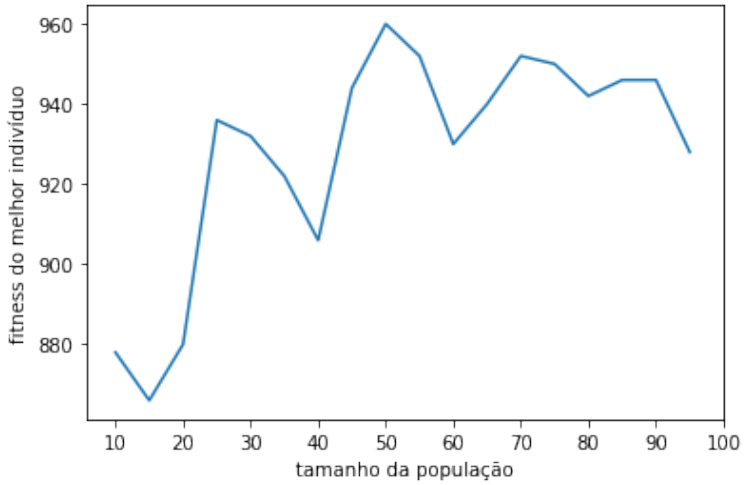


Figura 19 – Avaliação do tamanho população inicial x tempo de execução do AG.

analisar-se a Figura 19 vê-se que a performance do AG com o valor $50$ representando o tamanho da população foi a melhor, ou seja, obteve o maior valor de *fitness*.

Outro parâmetro que precisou ser definido foi a taxa de mutação. Para isso, fixou-se o tamanho da população em $50$, por ser o melhor valor após avaliados os gráficos das figuras 18 e 19 apresentados anteriormente e a taxa de *crossover* em $0,8$ e variou-se a taxa de mutação de $0,05$ a $0,9$, incrementando $0,05$ em cada iteração. Foram então gerados outros gráficos de *taxa de mutação x tempo de execução* e *taxa de mutação x fitness do melhor indivíduo* mostrados respectivamente nas Figuras 20 e 21.

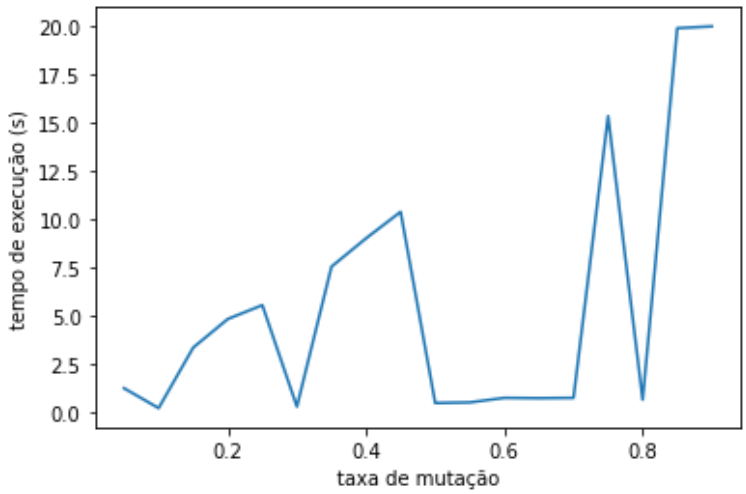


Figura 20 – Avaliação de taxa de mutação x tempo de execução do AG.

Na Figura 20 nota-se que um alto valor de taxa de mutação culmina em um tempo de execução extremamente elevado, por ser uma operação de grande custo computacional. Em contra partida, com a configuração ideal do parâmetro o tempo de convergência é baixo.

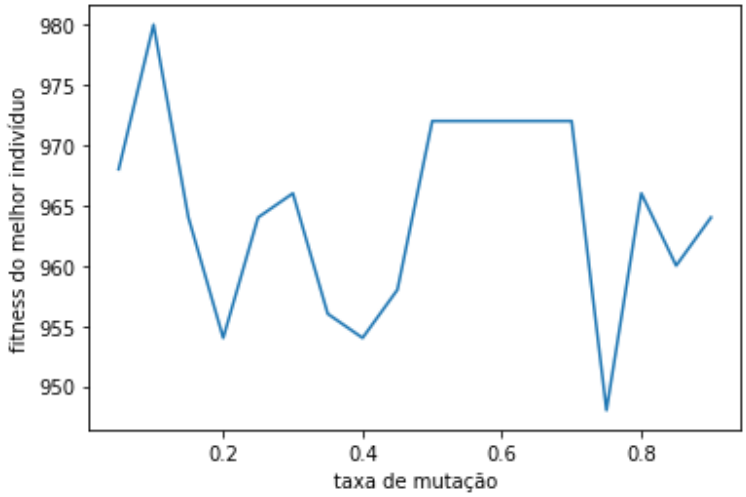


Figura 21 – Taxa de mutação do AG x *fitness* do melhor indivíduo.

A Figura 21 apresenta a variação da *fitness* do melhor indivíduo em função da variação da taxa de mutação. Percebe-se que uma maior taxa de mutação nem sempre significa um melhor desempenho do AG, uma vez que essa operação genética pode acabar manipulando melhores indivíduos da população e levando-os para mais longe da convergência.

Após a análise, chegou-se a conclusão que a melhor taxa de mutação para o problema abordado nesse trabalho foi a de $0,1$, pois apesar de diversas taxas apresentarem baixo tempo de execução, essa foi a que apresentou a melhor *fitness* comparada às demais candidatas.

As Figuras 22 e 23 representam os gráficos que foram utilizados para a avaliação da melhor taxa de *crossover*. Nelas pode-se ver respectivamente a comparação de taxa de *crossover* em relação ao tempo de execução do AG e taxa de *crossover* em relação à *fitness*.

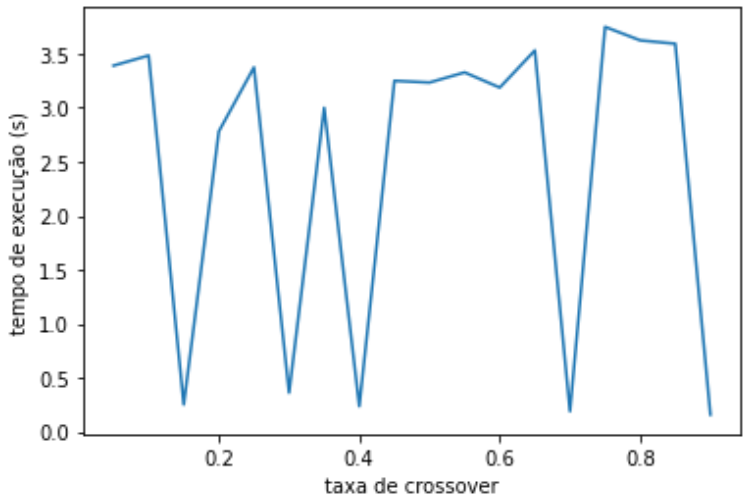


Figura 22 – Avaliação de taxa de *crossover* x tempo de execução do AG.

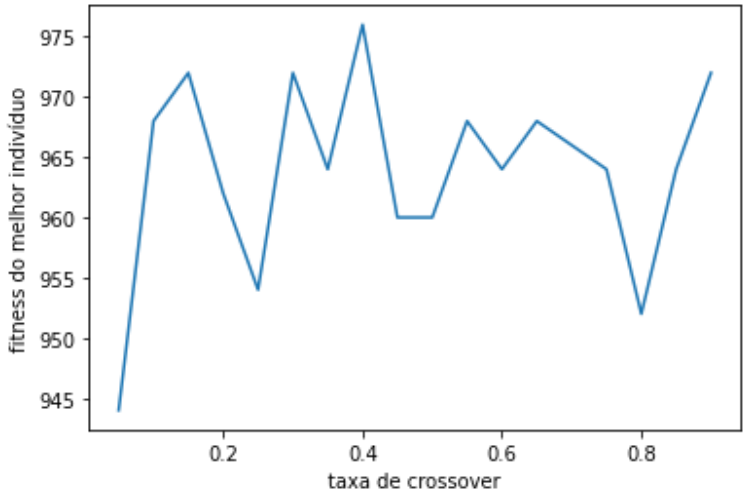


Figura 23 – Taxa de *crossover* do AG x *fitnes* do melhor indivíduo.

Para a avaliação e geração dos gráficos apresentados acima fixou-se o tamanho da população em $50$ e a taxa de mutação em $0,1$. Como resultado obteve-se o valor de $0,4$ para melhor taxa de *crossover* por ser a taxa com o menor tempo de execução e ao mesmo tempo maior *fitness*.

Vale ressaltar que a definição inicial dos parâmetros foi feita de acordo com a literatura, baseando-se na abordagem de Carvalho (2017), que propõe a fixação de parâmetros a fim de definir o tamanho da população inicial e posteriormente, utilizando o valor encontrado, a definição dos demais.

## 4.2 Execução do algoritmo genético

Tendo definidos os parâmetros na seção Avaliação de parâmetros foi possível executar o algoritmo genético proposto. Diante do cenário onde deseja-se levar o robô até a bola, a Figura 24 mostra a saída do AG, ou seja, o caminho gerado.

O caminho apresentado em verde possui curvas muito bruscas, de $45$ ou $90$, além de dar algumas voltas um pouco mais longas que o necessário. Isso acontece devido à mudança de direção durante o traçado do caminho através do *grid*. Para resolver esse problema, foi proposta a utilização da curva de Bézier, que busca suavizar o caminho gerado. A Figura 25 mostra em vermelho o caminho obtido após a aplicação do algoritmo 6 descrito na seção 3.5.

Para calcular o tempo de execução do algoritmo foi considerada a média dos tempos de $50$ execuções, com robô e bola em posições variadas. O valor obtido foi de $1,59$ segundos, o que dado o cenário altamente dinâmico pode representar um possível ponto de melhora, já que idealmente esse tempo deveria ser de no máximo $0,5$ segundos, pois é o tempo em que ocorre a troca de pacotes entre o simulador e o código da equipe. O resultado apresentado no gráfico

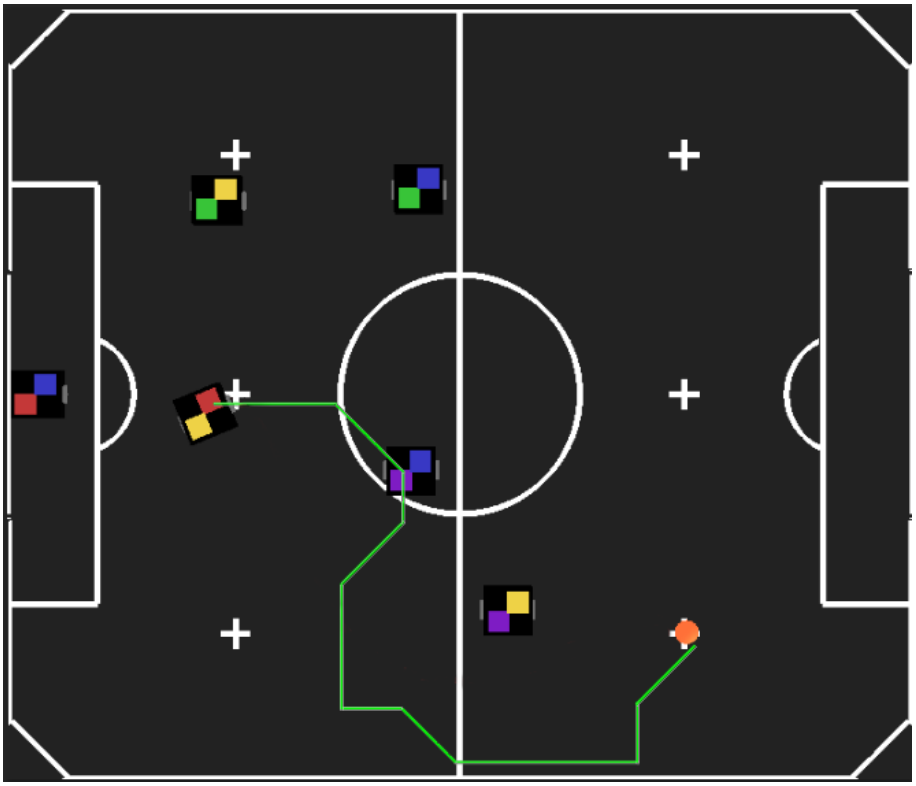

Figura 24 – Caminho gerado pelo AG.

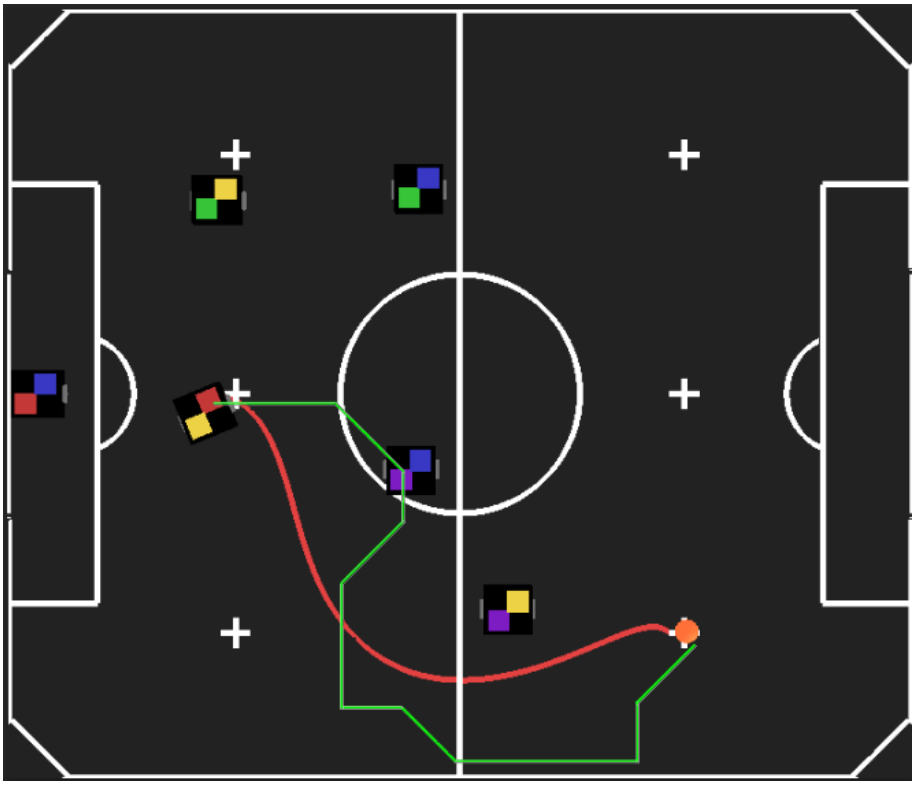

Figura 25 – Caminho gerado pelo AG antes e após aplicação da Curva de Bézier.

da Figura 26 é visivelmente reflexo da aleatoriedade, uma vez que para solucionar o mesmo problema tem-se uma grande variação de tempos de convergência.

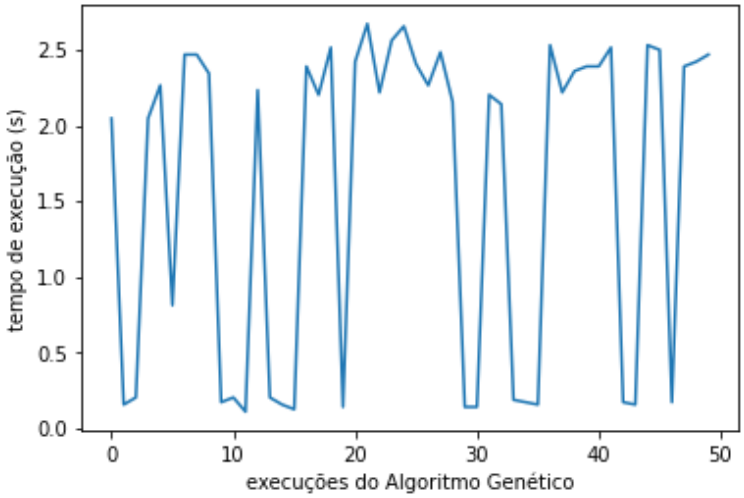


Figura 26 – Tempo de execução do AG.

# 5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi apresentado de maneira geral o desenvolvimento de robôs diferenciais futebolistas para a categoria IEEE Very Small Size Soccer. O foco do trabalho foi o desenvolvimento e aplicação de técnicas de planejamento de caminho para o seguimento. Uma vez definidos os pontos inicial e final pelo módulo de estratégia, o algoritmo genético apresentado neste trabalho acompanha a proposta de Song, Wang e Sheng (2016) e é responsável por responder com a melhor rota entre esses pontos, deixando o robô em melhor condição para alcançar seu objetivo. Posteriormente, ainda faz-se a utilização de curvas de Bézier para suavizar o caminho, deixando o robô com mais dinamicidade e movimentos menos robóticos.

Dada a importância do planejamento de caminhos no ramo da robótica móvel, esse trabalho atendeu aos objetivos propostos, criando um planejador de caminhos eficiente que calcula a melhor rota para o robô, dando a ele a capacidade de planejar os seus movimentos de maneira mais otimizada e exata.

Este trabalho indica a possibilidade de utilizar o algoritmo genético desenvolvido aliado às curvas de Bézier, como um planejador para robôs futebolistas, já que o mesmo tende a encontrar o caminho ótimo até o objetivo, principalmente em situações em que a região livre para se locomover é pequena.

Como sugestões de trabalhos futuros propõe-se a implementação de um novo controle de posição para o robô, de modo que o mesmo possa seguir o caminho planejado previamente e um controle embarcado de velocidade. Também é interessante notar que o trabalho abre espaço para novos testes envolvendo modificações nos parâmetros, o que pode influenciar no tempo de execução do AG, já que em cenários realistas e competitivos as tomadas de decisão do robô precisam ser extremamente rápidas para que esse possa alcançar seu objetivo antes que seu adversário.

# REFERÊNCIAS


BLICKLE, T.; THIELE, L. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 1996. v. 4, n. 4, p. 361–394, 1996. Citado na página 24.

BURDELIS, M. A. P. *Ajuste de Taxas de Mutação e de Cruzamento de Algoritmos Genéticos Utilizando-se Inferências Nebulosas*. 2009. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3141/tde-14082009-180444/en.php. Acesso em: 24 de outubro 2022. Citado na página 23.

CARVALHO, W. L. d. O. *Estudo de Parâmetros Ótimos em Algoritmos Genéticos Elitistas*. 2017. Disponível em: https://repositorio.ufrn.br/bitstream/123456789/24067/1/WandersonLaerteDeOliveiraCarvalho_DISSERT.pdf. Acesso em: 21 de outubro 2022. Citado na página 45.

CHOI, J.-w.; CURRY, R.; ELKAIM, G. Path planning based on bézier curve for autonomous ground vehicles. In: . [S.l.: s.n.], 2008. p. 158 – 166. Citado na página 16.

FARIN, G. Algorithms for rational bézier curves. *Computer-Aided Design*, 1983. v. 15, n. 2, p. 73–77, 1983. ISSN 0010-4485. Disponível em: <https://www.sciencedirect.com/science/article-/pii/0010448583901719>. Citado na página 25.

FERNANDES, L. C. *Análise e implementação de algoritmos para localização de robôs móveis*. 2003. Disponível em: https://www.teses.usp.br/teses/disponiveis/55/55134/tde-01022016-105642/en.php. Acesso em: 21 de outubro 2022. Citado na página 21.

FRENZEL, J. Genetic algorithms. *IEEE Potentials*, 1993. v. 12, n. 3, p. 21–24, 1993. Citado na página 20.

HERMAN, H. et al. Scheduling using genetic algorithm and roulette wheel selection method considering lecturer time. *Journal of Information Technology and Its Utilization*, 2019. v. 2, p. 24, 08 2019. Citado na página 25.

HUANG, G.; WANG, L. Hybrid neural network models for hydrologic time series forecasting based on genetic algorithm. In: *2011 Fourth International Joint Conference on Computational Sciences and Optimization*. [S.l.: s.n.], 2011. p. 1347–1350. Citado na página 24.

IEEE. *Rules for the IEEE Very Small Competition*. 2008. Disponível em: https://ore.ucsp.edu.pe/wp-content/uploads/2019/05/ReglasORE2019_VerySmall.pdf. Acesso em: 28 de outubro 2022. Citado 2 vezes nas páginas 8 e 14.

KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 2020. 2020. Citado na página 15.

MEDEIROS, S. C. D. *Inversão de Parâmetros em Dados Sísmicos por Algoritmos Genéticos*. 2005. Disponível em: https://www.maxwell.vrac.puc-rio.br/8622/8622_1.PDF. Acesso em: 24 de outubro 2022. Citado 2 vezes nas páginas 22 e 24.

MOHAMED, E.; ABDULAZIZ, S.; XIAOHUI, Y. *Optimizing robot path in dynamic environments using Genetic Algorithm and Bezier Curve*. 2017. Disponível em: https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs17348. Acesso em: 24 de outubro 2022. Citado na página 26.

MOHEBI, E.; BOUYER, A.; KARIMI, M. An efficient clustering of the som using rough set and genetic algorithm. In: *2009 5th IEEE GCC Conference Exhibition*. [S.l.: s.n.], 2009. p. 1–5. Citado na página 24.

MOLINA, L. *Planejamento de movimento para robôs móveis baseado na condição de horizonte continuado (CHC)*. 2014. Disponível em: http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/8318. Acesso em: 10 de outubro 2022. Citado na página 17.

PACHECO, M. A. C. *ALGORITMOS GENÉTICOS: PRINCÍPIOS E APLICAÇÕES*. 1999. Disponível em: http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/MQ-intro_apost.pdf. Acesso em: 24 de outubro 2022. Citado 2 vezes nas páginas 8 e 23.

PARIZOTO, F. C. *Especificação de sonda robótica móvel para coleta de amostras de solo para análise química de parâmetros de fertilidade do solo*. 2010. Disponível em: http://hdl.handle.net/1884/25906. Acesso em: 21 de outubro 2022. Citado na página 13.

PINTO, A. H. M. *Regras IEEE Very Small Size Soccer (VSSS) - Série B*. 2021. Disponível em: https://www.cbrobotica.org/wp-content/uploads/2021/05/vssRules3x321.pdf. Acesso em: 21 de outubro 2022. Citado na página 18.

RAZA, M. Autonomous vehicles: Levels, technologies, impacts and concerns. *International Journal of Applied Engineering Research*, 2018. 2018. Citado na página 13.

SANCHEZ-HERMOSILLA, J.; PAEZ, F.; RINCON V., D. J. G. *Mechanical Design and Development of an Electric Mobile Robot for Agricultural Tasks in Greenhouses*. 2012. Disponível em: https://www.researchgate.net/publication/268268830_Mechanical_Design_and_Development_of_an_Electric_Mobile_Robot_for_Agricultural_Tasks_in_Greenhouses. Acesso em: 21 de outubro 2022. Citado na página 13.

SANTOS, P. H. dos. *Planejamento de movimento utilizando campos vetoriais para robôs diferenciais futebolistas*. 2019. Disponível em: http://www.monografias.ufop.br/handle/35400000/2355. Acesso em: 10 de outubro 2022. Citado na página 30.

SHANAHAN, J. G. Machine learning. In: ____. *Soft Computing for Knowledge Discovery: Introducing Cartesian Granule Features*. Boston, MA: Springer US, 2000. p. 143–175. ISBN 978-1-4615-4335-0. Disponível em: <https://doi.org/10.1007/978-1-4615-4335-0_7>. Citado na página 24.

SIMBA, K. R.; UCHIYAMA, N.; SANO, S. Real-time smooth trajectory generation for nonholonomic mobile robots using bézier curves. *Robotics and Computer-Integrated Manufacturing*, 2016. v. 41, p. 31–42, 2016. ISSN 0736-5845. Disponível em: <https://www-.sciencedirect.com/science/article/pii/S0736584516300552>. Citado 3 vezes nas páginas 8, 17 e 27.

SONG, B.; WANG, Z.; SHENG, L. A new genetic algorithm approach to smooth path planning for mobile robots. *Assembly Automation*, 2016. 2016. Citado 3 vezes nas páginas 16, 31 e 48.