



TIC-TAC-ROBOT

*Nelson Hugo de Carvalho – onelson.hugo.nelson@gmail.com
FIAP-Faculdade de Informática e Administração Paulista
Rua Vinicius de Moraes,254
06341-330 – Carapicuíba – São Paulo*

*Daniel Shibata Oliveira – smart.leinad@gmail.com
FIAP-Faculdade de Informática e Administração Paulista
Rua Doutor Achilles Lisboa 196, apto. 132
02937-090 – São Paulo – SP*

*Harley Ferreira Silva – harley201009@hotmail.com
FIAP-Faculdade de Informática e Administração Paulista
Rua Cônego Ladeira. 438, casa 1
02309-080 - São Paulo - SP*

*Raphael Nakamura de Oliveira – raphalol17@hotmail.com
FIAP-Faculdade de Informática e Administração Paulista
Rua Osiris Magalhães de Almeida 331
05634-020 – São Paulo – SP*

*Erick Toshio Yamamoto – proferick.yamamoto@fiap.com.br
FIAP-Faculdade de Informática e Administração Paulista
Av. Lins de Vasconcelos, 1222
01538-001 – São Paulo - SP*

Resumo: Neste projeto desenvolveu-se um braço robótico com dois graus de liberdade, controlado por uma inteligência artificial, com o objetivo de jogar jogo da velha, a partir de capturas de imagens feitas de um tabuleiro e suas peças em tempo real. O tabuleiro foi planejado com peças físicas. Na extremidade do braço foi adaptado uma válvula de vácuo para que fosse possível pegar e movimentar as peças do tabuleiro. A estrutura física foi feita com material leve e em suas juntas foram adicionados servos motores para movimentar o braço no plano XY e a válvula para cima e para baixo que está acoplada na ponta da estrutura. Foi também desenvolvido um programa de controle contendo as equações de cinemática inversa, de escala, e rotação com o intuito de orientar o braço robótico. Para fazer as capturas de imagens e tratá-las, utilizou-se, as bibliotecas OpenCV e a Numpy, com isso é gerado uma matriz, que é enviada para a inteligência artificial para que seja executada uma jogada. Para validar a matriz do tabuleiro foi feito um algoritmo de uma "Inteligência Artificial", I.A., onde a mesma faz uma simulação virtual do jogo, analisando os espaços que ainda não contém uma peça e procurar o melhor espaço a ser preenchido para alcançar a vitória.

Palavras-chave: Inteligência artificial. Reconhecimento de Imagem. Braço robótico.

1 INTRODUÇÃO

O objetivo deste trabalho é desenvolver um robô autônomo com a capacidade de executar jogadas, com base na fundamentação teórica dos conceitos de inteligência artificial (I.A.), eletrônica de interface, reconhecimento de objetos, *computer-aided design* (CAD), cinemática inversa e pneumática.

A ideia se constrói a partir da foto do tabuleiro físico e com uma base de tratamentos na imagem, para melhor o reconhecimento das peças. Aliada com a tomada de decisões da I.A. e a realização da jogada a partir de um braço robótico no tabuleiro físico, similar aos trabalhos desenvolvidos por RIBEIRO et al. (2000), NETO (2007) e ALBUQUERQUE (2016).

Este projeto procura, assim, oferecer aos estudantes da área de tecnologia um material amplo de estudo, para que possam avaliar a importância da robótica e ao mesmo tempo incentivar a aplicação de novas tecnologias, como a visão computacional, circuitos eletrônicos, modelagem, realidade virtual e comparações de técnicas alternativas.

Será apresentada inicialmente na seção 2, os principais temas que serão utilizados, de forma sucinta, no decorrer do projeto além do que foi discorrido nesta introdução e justificando o uso das tecnologias. A seção 3 apresenta a lógica usada para desenvolvimento da inteligência artificial. Logo em seguida, na seção 4 é abordada a projeção das capturas de imagens utilizando a OpenCV, na seção 5, as metodologias aplicadas no projeto. Nas seções 6 e 7 são apresentados o desenvolvimento do projeto, o qual constitui-se do desenho e modelagem da estrutura física do braço, do desenvolvimento do circuito de interface digital que é utilizado para o controle e comunicação do braço robótico, e do fluxograma do sistema de controle do processo de escrita. Na seção 8 são demonstrados os resultados obtidos e, por fim, são apresentadas as principais conclusões do projeto.

2 TEMAS ABORDADOS

A teoria apresentada neste artigo desenvolve a base de assuntos que começaram a ser apresentados atualmente como: visão computacional e inteligência artificial (I.A.). Onde já eram discutidos e utilizados em pequena escala na indústria e em pesquisas. Nos últimos anos a população tem tido maior contato com eles através de alguns produtos, apesar do preço atual deles ser alto, por exemplo, placas de vídeo dedicados para I.A. (GARCIA, 2020) e os carros autônomos (PEREIRA, 2020). Mas essas são aplicações de grande complexidade então não servem como porta de entrada e mesmo que sejam explicadas de forma detalhada, poucos entenderiam completamente.

O objeto que será estudado com a tecnologia proposta é um jogo da velha, devido a interação de uma pessoa contra uma máquina, o jogo se encontra no meio físico e no digital, assim apontando para o desenvolvimento no ambiente de visão computacional.

3 INTELIGÊNCIA ARTIFICIAL

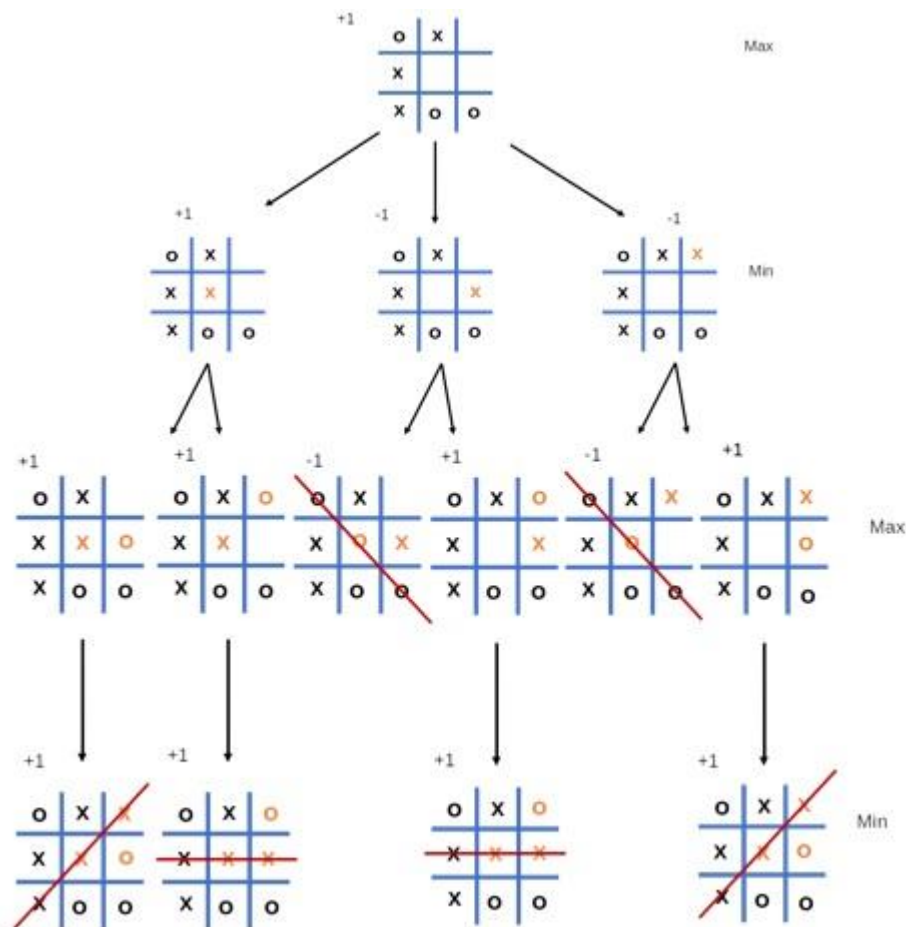
Para o funcionamento do projeto foi utilizado o uso de um algoritmo desenvolvido na linguagem de programação Python. O algoritmo do controle do foi baseado em conceitos de Machine Learning, inspirado em projetos de dois autores, AGRAWAL (2020) e BORCAN (2020).

Foi utilizado um método de desenvolvimento baseado no conceito de Game Tree no qual é imaginado o jogo a partir de uma ramificação tendo seu ponto inicial no top e camadas de divisões com possíveis movimentos a seres feitos, e cada movimento é chamado de “nó”.

O programa é capaz de fazer uma previsão de movimento com bases nos espaços vazios no tabuleiro. Sendo assim, prevendo jogadas futuras do jogador e fazendo a melhor tomada de decisão. Abaixo segue a ilustração do jogo como um *Game Tree* (Árvore de jogo), que é a base utilizado para criar todo o algoritmo de controle do jogo.

A Figura 1 representa as jogadas possíveis, e cada possível jogada é identificada como um nó, e os resultados são apresentados como: -1 (Perder), +1 (Ganhar) e 0 (Empatar). Assim basta a I.A. seguir o caminho que leva a +1.

Figura 1 - GAME TREE



Fonte: (Autoria Própria)

Seu funcionamento se aplica desta forma:

- O algoritmo vai descendo os nós da *Game Tree* até chegar no final, identificando onde o jogador perdeu, ganhou ou empatou;
- Logo após, o algoritmo sobe um nó e identifica de quem é o turno (jogador ou oponente);
- Caso for o turno do oponente, o algoritmo guarda o menor (Min) resultado de suas respectivas ramificações;
- Caso for o turno do jogador, o algoritmo guarda o maior (Max) resultado de suas respectivas ramificações;
- Este processo será repetido até chegar no primeiro nó da *Game Tree*.

3.1 Jogo da Velha

O jogo da velha possui uma lógica de funcionamento de fácil compreensão, constituído por nove casas onde os dois jogadores são representados como X ou O, e ganha o primeiro jogador a completar uma sequência de três casas, seja vertical, horizontal ou diagonal.

Dentro do código é simulado um tabuleiro virtual a partir de uma matriz, uma lista para a criação das peças, funções lógicas para realizar a verificação de uma jogada e seus respectivos resultados.

4 CAPTURA DE IMAGENS

Neste projeto foi utilizado várias abordagens para reconhecer e analisar objetos de diferentes formatos. Com o objetivo de encontrar a posição que está localizado cada objeto. (ANTONELLO, 2017; HASSAN, 2020)

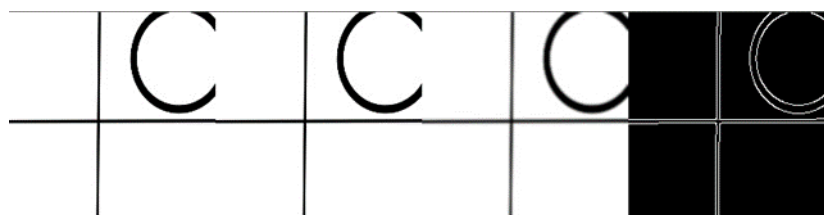
O programa inicia com o uso de uma webcam ou câmera que possa tirar as fotos do tabuleiro de forma programada. Assim é definido a resolução delas, suas identificações e classificação das peças. As estratégias para reconhece-las no tabuleiro são divididas em valor da área e a cor de cada objeto.

4.1 Valor da área de cada objeto

Ao capturar a imagem do tabuleiro é feito o tratamento de *Crop* na imagem e mudança de perspectiva, essa função tem o objetivo de centralizar o tabuleiro e deixar a perspectiva da foto a um ângulo reto ao tabuleiro. Isso facilita a normalizar o formato das peças e dos quadrantes. Em seguida começa a divisão do tabuleiro em nove quadrantes, após isso vem um tratamento de cor colocando-os em uma escala de cinza para cada pedaço. Assim melhorando os contornos e aumentando o contraste que o tabuleiro faz com as peças.

O seguinte tratamento de imagem, conforme a Figura 2, é o *Blur* que irá borrar as imagens de tal forma que deixe o fundo desfocado e a peça destacada. Em seguida foi feito o tratamento de imagem *Canny* para aumentar os contornos das linhas e diminuir o ruído da imagem.

Figura 2 - Utilização do *Blur* na imagem adquirida pela câmera.



Fonte: (Autoria Própria)

Por fim é feito uma função que irá reconhecer a área do objeto. Deste modo, sobram três possibilidades: um valor para a área de círculo, um para o X ou nenhuma área. Esses dados são colocados em duas matrizes:

- Uma delas a matriz do jogador (O);
- A outra matriz do Computador (X).

Com ambas as matrizes geradas *a priori* e elas são encaminhadas para a I.A..

4.2 Cor de cada Objeto

Essa estratégia baseia-se na diferenciação de cores. Nela, o tabuleiro e os 2 conjuntos de peças terão cores diferentes. A câmera capturará a imagem geral do tabuleiro. Equivalente a primeira estratégia, é utilizada a função *Crop* e mudança de perspectiva para reduzir erros de captura.

Em seguida o programa dividirá em nove quadrantes o tabuleiro, com isso começa o tratamento da imagem. Primeiro, ocorrerá a mudança da escala de cor da imagem para escala de saturação. Posteriormente, é usado o *Blur* para borrar a imagem e melhorar o reconhecimento.

Depois foi empregado uma função que irá analisar qual cor é predominante em cada quadrante.

Com isto teremos três possibilidades, sendo elas cor azul que representa Círculo (O) e verde que representa o (X) ou por fim branco que representa um espaço vazio e não utilizado no quadrante. Mandamos esses dados a duas matrizes:

- Uma delas a matriz do jogador(O);
- A outra matriz do Computador(X).

Com ambas as matrizes geradas, como a estratégia anterior, elas são também encaminhadas para a I.A..

O intuito dessa parte do programa foi estudar os princípios básicos de reconhecimento de imagem operando o Python na versão 3.7, e as bibliotecas OpenCV e Numpy.

5 METODOLOGIAS

Para o desenvolvimento do projeto utilizamos uma metodologia aplicada em projetos com foco na gestão, PMBOK (*Project Management Body of Knowledge*), dispo de um gestor de projetos, além de datas para os entregáveis garantindo um avanço eficiente para a próxima etapa, tendo como base um escopo definido e então criado um protótipo virtual. Dentro de cada etapa de desenvolvimento do projeto são usados métodos baseado em conceitos tecnológicos de aplicação tanto robótica (Scara - *Selective Compliance Assembly Robot Arm*) quanto de programação para IA (*Game Tree*) e programação para *Front-End* (Open CV – Python), além de um método para visão computacional (*Image Recognition*) - focado em contornos de áreas e diferenciação de cores.

6 BRAÇO ROBÓTICO

O braço robótico é baseado no modelo SCARA (*Selective Compliance Assembly Robot Arm*) onde ele pode se mover livremente nos eixos XY, permanecendo fixo no eixo Z. O movimento é controlado pela rotação de dois servos acoplados às engrenagens na base de cada elo do braço. No final do braço está outro servo que move uma viga para cima e para baixo. Na parte inferior da viga, há uma válvula de vácuo que será ativada dinamicamente para levantar as peças do jogo.

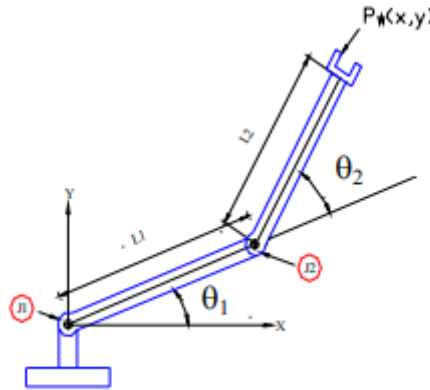
Foi utilizado a placa Arduino, através da comunicação serial possibilitando conectar-se com a linguagem Python, que tem a biblioteca de suporte ao serial. Assim, toda vez que há a captura de imagem, conforme visto na seção 3 e 4, a qual é processada e enviado a matriz para I.A..

A inteligência artificial abrirá uma comunicação com a serial do Arduino, e enviará os ângulos em forma de vetor de *byte*, por meio deste, receberá esse vetor e o transformará em um número flutuante. Logo, converterá para o servomotor para que possa executar um movimento.

6.1 Cinemática Inversa

Para a movimentação dos dois servos motores é usado o conceito de cinemática inversa, visto a Figura 3. (KIEU, 2017; FDXLABS, 2019)

Figura 3 - Cinemática do Braço Robótico.



Fonte: (MOREIRA,2006)

Onde braço recebe duas coordenadas (x, y) e através do cálculo da cinemática inversa apresentadas pelas Equações (1), (2), (3) e (4), são gerados dois ângulos um para cada servo de um elo. (MOREIRA, 2006)

$$\theta_2 = \pm \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right) \quad (1)$$

$$\tan \alpha = \frac{y}{x} \quad (2)$$

$$\beta = \arccos\left(\frac{l_2^2 - l_1^2 - r^2}{2 l_1 r}\right) \quad (3)$$

$$\theta_2 = (\alpha - \beta) \quad (4)$$

6.2 Válvula de vácuo

Utilizou-se uma válvula de vácuo para captura da peça. Após o braço executar o movimento no plano, é enviado um sinal para o servo que está acoplado na ponta do braço, logo ele é acionado, fazendo um movimento para que a válvula se aproxime da peça e pegue-a. Portanto, o servo faz o movimento de recuo. Sendo assim, feita em seguida uma verificação através da captura de imagem para confirmar se o robô moveu a peça corretamente.

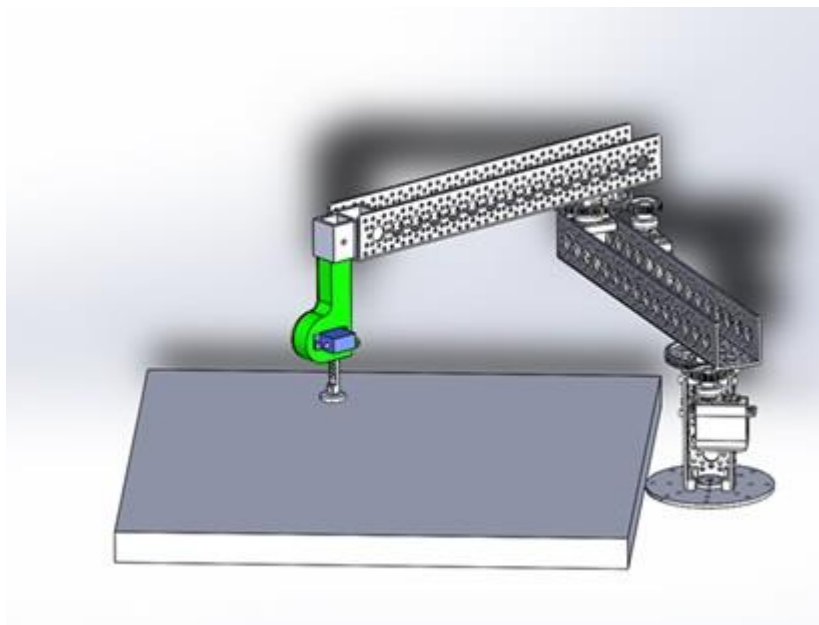
6.3 Circuito

Para fazer o controle do braço foi utilizada a placa eletrônica Arduino, na qual controla dois servomotores modelos 996R, acoplados em cada elo e um servo SG90 que movimenta uma haste no eixo z. Também foi usada, uma válvula de vácuo, que é acionada pelo Arduino e está conectada a um dispositivo relé.

7 DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto baseia-se na construção da estrutura do braço robótico, da válvula de vácuo, do circuito digital, da comunicação de dados e do desenvolvimento do código computacional em linguagem Python e Linguagem C++. O comprimento das hastes fora definido em 200 mm, para que fosse possível cobrir todo o tabuleiro.

Figura 4 - Braço Robótico no Simulador.



Fonte: (Autoria Própria)

8 CONSIDERAÇÕES FINAIS

No presente trabalho investigamos o uso do processamento de sinais de imagem, da inteligência artificial e do desenvolvimento do braço robótico. Foram apresentadas as principais soluções para a filtragem das imagens obtidas via webcam/câmera e o tratamento delas com métodos de *Crop*, *Blur* e *Canny*. *A posteriori*, exploramos o uso da inteligência artificial na tomada de decisão em relação aos dados do jogador. Finalmente a integração de todos os sistemas com o braço robótico via comunicação serial, gerando a interação com o público.

Portanto, este projeto procura mesclar a visão computacional, inteligência artificial e meio físico, com objetivo de disseminar que essas categorias, pois apesar de parecerem complicadas e avançadas, não são coisas fora da realidade. Com isso, chegamos à conclusão que, o jogo da velha é algo que facilita a aplicação desses conceitos e também é um ótimo exemplo, para apresentar ao público, uma vez que, é de fácil entendimento.

Os resultados vistos nesse artigo apresentam uma aplicação simples para a solução e métodos de tratamento na visão computacional, que são abordados regularmente no dia a dia

da população, tais como: o cálculo da área, e diferenciação do contraste da cor para o reconhecimento das peças, utilizadas no jogo. Em relação ao desenvolvimento físico (braço robótico) buscamos, instigar o estudo a automação industrial e, trazer para o meio do entretenimento algo que é visto com mais frequentemente no ramo tecnológico industrial.

Em virtude de tudo que foi mencionado, a conclusão desse projeto tem como objetivo, apresentar a comunidade FIAP, um protótipo funcional, em um evento interno da instituição (NEXT FIAP), transparecendo todos resultados obtidos desse estudo. Logo, pretendemos expandi-lo à outras instituições, compartilhando nossos conhecimentos e resultados deste projeto.

Agradecimentos

Agradecemos ao Professor Erick Toshio Yamamoto, por ser nosso mentor nesse projeto, e nos ensinar o caminho de entrar no mundo Científico. Agradecemos também ao Centro Universitário FIAP por disponibilizar espaço e a oportunidade de entrar no mundo de Iniciação Científica.

REFERÊNCIAS

ALBUQUERQUE, Eduardo Prates. **Inteligência artificial: aplicação do algoritmo Minimax em um jogo de xadrez**. 2016. 50 f. Monografia (Bacharel em Sistemas de Informação) - Universidade Estadual de Mato Grosso do Sul, Dourados, 2016.

AGRAWAL, Rohit. **Playing Tic Tac Toe using Reinforcement Learning**. Codementor.io. Disponível em: <https://www.codementor.io/@rohitagrawalofficialmail/playing-tic-tac-toe-using-reinforcement-learning-x5rf9xvey> . Acesso em: 1 de maio de 2020.

ANTONELLO, Ricardo. **Introdução a Visão Computacional com Python e OpenCV: Versão 0.8**. Rio Grande do Sul: IFC. 2017. p.68

BORCAN, Marius. **Machine Learning Project: Neural Network Learns To Play Tic-Tac-Toe**. Programmerbackpac. Disponível em: <https://programmerbackpack.com/python-neural-network-tutorial/> . Acesso em: 1 de maio de 2020.

FDXLABS. **Convertendo coordenadas Y, Z em ângulos para um braço robótico de dois eixos**. Fdx LABS. Disponível em: <https://www.fdxlabs.com/converting-y-z-coordinates-into-angles-for-a-two-axis-robot-arm/> . Acesso em: 10 de mai. de 2020.

GARCIA, Álvaro Molina. **Inteligencia Artificial para Videojuegos con Deep Reinforcement Learning**. Universidad de Alicante. Departamento de Tecnología Informática y Computación, 2020

HASSAN, Murtaza. **LEARN OPENCV in 3 HOURS with Python | Including 3x Example Projects (2020)**. Disponível em: <https://www.youtube.com/watch?v=WQeoO7MI0Bs&t=6792s> . Acesso em: 2 de mar. 2020.

KIEU, Aary. **Projeto Python | Cinemática Avançada e Inversa com Robô Planar 2 DOF**. YouTube. Disponível em: <https://www.youtube.com/watch?v=3rFaZMvgNe8> . Acesso em: 10 de mai. de 2020.

MEYER, Joey. **The Arm. Raspberry Turk, c2017**. Disponível em:
<http://www.raspberryturk.com/details/arm.html> . Acesso em: 21 de mar. de 2020

MOREIRA, Flávia. TERUHIKO, Émerson. CARRARA, Valdemir. **ESTUDO DA CINEMÁTICA INVERSA APLICADA NUM BRAÇO ROBÓTICO**. Anais do 12º Encontro de Iniciação Científica e Pós-Graduação do ITA – XII ENCITA. Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil, Outubro, 16 a 19, 2006.

NETO, Henrique de Castro. **LS-DRAUGHTS um sistema de aprendizagem de jogos de Damas baseado em algoritmos genéticos, redes neurais e diferenças temporais**. 2007. 145 f. Dissertação (Mestrado em Ciências Exatas e da Terra) - Universidade Federal de Uberlândia, Uberlândia, 2007.

PEREIRA, Caio Faganello. **Inteligência artificial em veículos autônomos: um estudo sobre o campo de aplicação**. 2020. 68 f. Monografia (Bacharel em Sistemas de Informação) - Universidade do Sul de Santa Catarina, Florianópolis, 2020.

QUAGLIO, Vinicius Gomes. **Técnicas de inteligência artificial aplicada do jogo Othelo: um estudo comparativo**. 2013. 33 f. Monografia (Bacharel em Ciência da Computação) - Universidade Estadual de Londrina, Londrina, 2013.

RIBEIRO, Sérgio Faustino; FRANÇA, Juracy Emanuel M. da; BARROS, Marcelo Alves de; CAVALCANTI, José Homero Feitosa. **UM SISTEMA DE VISÃO PARA RECONHECIMENTO DE PEÇAS EM UM TABULEIRO DE XADREZ**. Congresso Nacional de Engenharia Mecânica (CONEM). Universidade Federal da Paraíba – CCT/COPIN/NEUROLAB-CT/DTM, 2000.

TIC-TAC-ROBOT

Abstract: *In this paper we developed a robot arm with 2 degrees of freedom, controlled by Artificial Intelligence, with the objective to play Tic Tac Toe, starting with image capture of the board and its pieces in real time. The board was planned in physical pieces. At the end of the robotic arm it was adapted a vacuum valve to make it possible whenever in need to take or move the pieces of the board. The physical structure was made with a light material and in your joints were added servo motors for the arm movement in the plane XY and the valve for up and down that is accompanied at end of the structure. It was too developed a program of control containing the equations of inverse kinematics, of scale, and rotation with the purpose to orient the robotic arm. To take images and treat them, it was used the libraries OpenCV and a NumPy, thereby it generates two matrices, that is sent to the artificial intelligence so that is executed the next move. To validate the matrix of the board, an “Artificial Intelligence” algorithm was made, where the same one does the virtual simulation of the game, analyzing the spaces that still have pieces or a blank space and make the best decision to achieve victory.*

Keywords: *Artificial Intelligence. Image Recognition. Robotic Arm.*