

Desenvolvimento de uma Bancada Integrada de Robótica, Inteligência Artificial e Visão Computacional

Herculano De Biasi¹, Nilton Kazuo Gomes Suzuki², Rudinei Santos³

^{1,2,3}Departamento de Engenharia de Controle e Automação
Universidade do Contestado (UnC) – Curitiba, SC – Brasil

{¹herculano.debiasi, ²nkazuo}@unc.br, ³engenheiro.rudy@hotmail.com

Abstract. *This article describes the design and construction of an integrated workbench of robotics, artificial intelligence and computer vision. After explaining the project and assembly of the workbench, some experiments developed with it are described, including: Integration with PLC, gesture control with Kinect, remote control by radio frequency via a control glove, detection of dominoes on a treadmill with orientation alignment. The workbench has been used to assist the lecturers during classes of several subjects of the undergraduate course in Automation and Control Engineering at the UnC – University of Contestado on the city of Curitiba-SC. Arduino and Raspberry Pi microcontroller boards were used, as well as a RaspiCam module.*

Resumo. *O presente artigo descreve o projeto e construção de uma bancada integrada de robótica, inteligência artificial e visão computacional, detalhando aspectos de projeto e montagem. Experimentos realizados com ela são descritos, incluindo: Integração com CLP, controle gestual com Kinect, controle remoto por radiofrequência através de luva de controle, e detecção de peças de dominó em uma esteira com correção de posicionamento. A bancada foi utilizada em diversos componentes curriculares do curso superior de Engenharia de Controle e Automação da UnC – Universidade do Contestado, da cidade de Curitiba-SC. Foram utilizadas as plataformas Arduino e Raspberry Pi bem como a câmera RaspiCam.*

1. Introdução

A utilização de bancadas didáticas é um recurso valioso para o processo de ensino-aprendizagem em cursos de engenharia. Infelizmente para instituições de ensino como as universidades comunitárias, o custo de aquisição de tais equipamentos é proibitivo. A bancada desenvolvida e descrita neste artigo, em especial o braço robótico, é de baixo custo e de projeto totalmente aberto, o que contrasta com soluções comerciais como as bancadas de automação industrial da empresa Festo Didactics, que são tecnologias fechadas. O fato de ele precisar de toda uma programação para funcionar faz com que os acadêmicos trabalhem com as mais variadas situações e criem diversas soluções para os problemas, além de possibilitar a integração com equipamentos como CLPs (Controladores Lógicos Programáveis), inversores de frequência e motores, sem distinção de marcas.

Outra vantagem que a bancada desenvolvida oferece é a possibilidade de realizar experimentos detalhados de cinemática (estudo dos movimentos) visando a determinação de posições, orientações e trajetórias que são possíveis de serem executadas pelo manipulador, aplicando para isso cálculos matemáticos.

A bancada detalhada neste artigo tem por objetivo integrar as áreas de robótica, inteligência artificial e visão computacional, mas também pode ser usada para projetos de automação envolvendo sensores, atuadores e CLPs.

2. Materiais e Métodos

Esta seção descreve o projeto e construção do braço robótico, bem como o de uma simulação interativa realizada com o *engine* de jogos UDK (*Unreal Development Kit*). A bancada completa é então apresentada. Um exemplo de integração com CLP é discutido. Duas formas de controle são abordadas, por gestos através do dispositivo Kinect [Bradski 2008] e por controle remoto da garra através de RF (radiofrequência) por meio de módulos XBee [Faludi 2011]. A seção encerra com uma aplicação de visão computacional e inteligência artificial para reconhecimento de peças de dominó.

2.1. Projeto e Construção do Braço Robótico

O manipulador foi projetado no software SolidWorks (Figura 1). Cerca de 20 peças separadas foram projetadas e desenhadas nele. Após isso foram inseridas as restrições (*constraints*) de movimentação. Essas restrições fazem com que o objeto virtual se comporte como o equipamento real, incluindo aí limitações nos graus de liberdade de movimentação e orientação, tipo de material e peso.

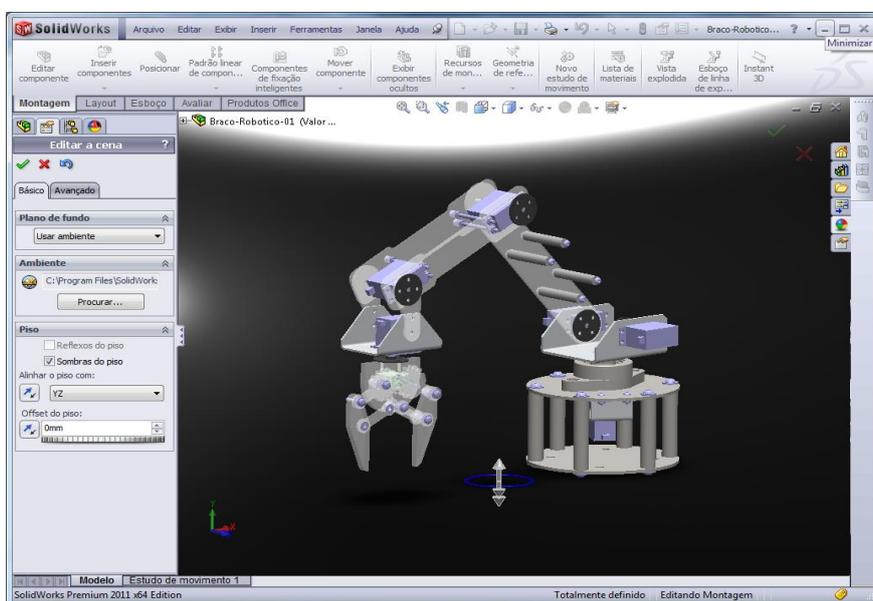


Figura 1. Projeto em SolidWorks

O braço robótico é controlado por uma placa Arduino. Arduino é uma plataforma *open source* de computação física e prototipação criada na cidade de Ivrea, Itália, em 2005. A plataforma é composta por uma placa microcontrolada baseada nos microcontroladores ATmega da empresa Atmel, uma IDE de desenvolvimento e bibliotecas que facilitam a prototipagem de sistemas embarcados [McRoberts 2011].

A Figura 2 mostra o braço já montado e a placa microcontrolada Arduino.



Figura 2. Braço montado e placa Arduino

2.2. Bancada Robótica

A Figura 3 mostra a bancada robótica em sua montagem final. Ela é composta pelos seguintes componentes: um sistema de câmera para a visão computacional, sensores de ultrassom que identificam se há objetos na esteira, esteira para movimentação de objetos, mesa de triagem de objetos e dois braços robóticos. A integração destes componentes permitiu aos acadêmicos realizarem vários experimentos voltados às disciplinas de automação, programação, robótica e visão computacional.



Figura 3. Bancada completa

2.2.1 Acionamento

Um circuito foi projetado e confeccionado para controlar a esteira. Este consiste basicamente de uma ponte-H capaz de acionar o motor da esteira para frente e parar trás. Os pinos quatro e cinco da placa Arduino são responsáveis por essa movimentação seguindo a tabela-verdade da Tabela 1.

Tabela 1: Tabela-verdade de controle do motor da esteira

Pino 4	Pino 5	Rotação do Motor
0	0	Parado
0	1	Anti-horário
1	0	Horário
1	1	Parado

A fonte de alimentação externa precisa fornecer 12 VCC necessários para atingir a corrente elétrica adequada para que o motor tenha torque suficiente para movimentar a esteira. A Figura 4 mostra o circuito elétrico desenvolvido com o *software* Yenka.

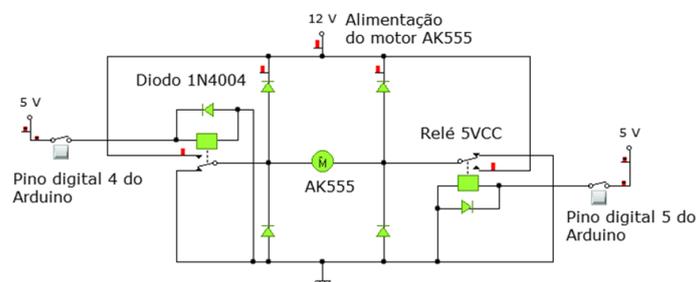


Figura 4. Esquema elétrico para movimentação da esteira

2.3. Integração com CLP

Neste experimento (Figura 5) foi utilizado um CLP (Controlador Lógico Programável) Siemens S7-200 programado em linguagem *Ladder* para realizar uma aplicação industrial em escala reduzida.

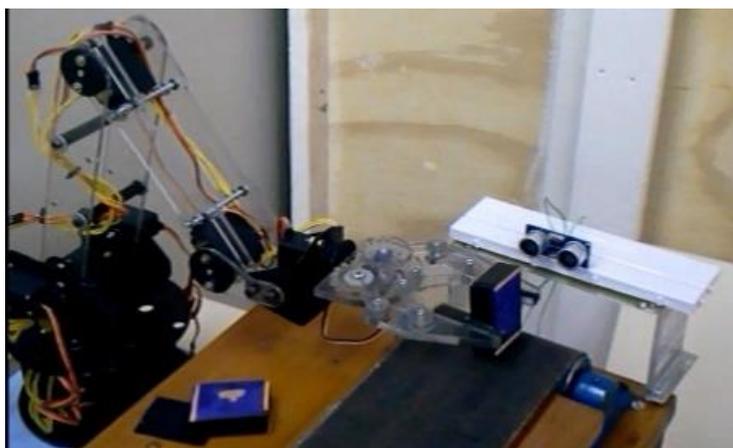


Figura 5. Experimento de integração com CLP

O experimento funciona da seguinte forma:

1. O robô coloca uma caixa sobre uma esteira.
2. Uma placa Arduino, em conjunto com um sensor ultrassônico, detecta a presença na caixa sobre a esteira. O sensor ultrassônico, HC-SR04, consegue realizar medições entre 2 e 400 cm, com uma precisão de 3 mm.
3. Após a caixa permanecer por mais de 3 segundos na frente da esteira, o Arduino chaveia um relé, que por sua vez fecha um circuito de 24V conectado a uma entrada digital do CLP.
4. Esta entrada do CLP, quando ativada, controla um programa em *Ladder* que tem por objetivo controlar um inversor de frequência.
5. Este inversor de frequência é ligado a um motor, o qual movimentar a esteira.
6. A caixa vai até o final da esteira e volta e, nesse intervalo, o manipulador robótico aguarda o seu retorno sem executar qualquer outro movimento.
7. Após o retorno da caixa, esta é retirada da esteira pelo braço robótico e o programa encerrado.

O processo somente inicia quando é dado o comando de início no CLP. Durante o processo são apresentadas mensagens como: Caixa avançando, caixa retrocedendo, processo iniciado e processo finalizado. A comunicação entre placa Arduino (controladora do braço robótico) e o CLP se deu através de suas entradas/saídas digitais, cujo esquema pode ser visto na Figura 6. Nesse processo foram utilizados relés, pois o Arduino trabalha em suas portas digitais com 5V e o CLP com 24V, fato que impossibilita a conexão direta das duas portas.

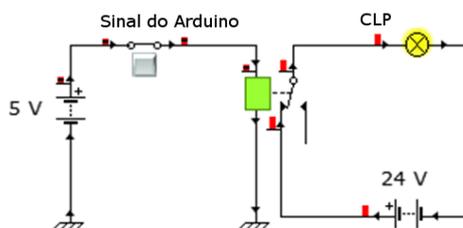


Figura 6. Experimento de integração com CLP

2.4. Simulações

Após a modelagem no SolidWorks, o projeto foi importado no software Maya (Figura 7), que auxilia o desenhista a criar modelos 3D realistas e as animações com eles, levando em conta fatores como peso, distorção devida a movimentos, cinemática inversa para articulações, etc. Maya torna rápido o desenvolvimento de animações, gerando cenas intermediárias entre uma cena inicial e outra final.

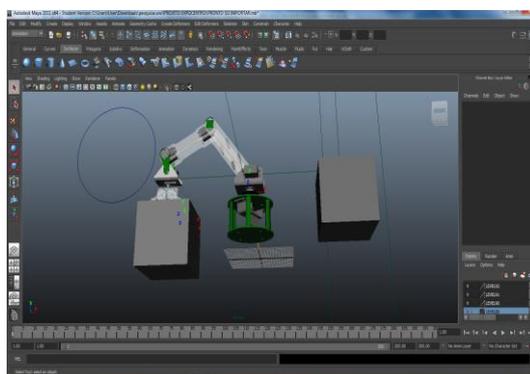


Figura 7. Modelagem em Maya

Esse modelo foi então empregado no *engine* UDK (Figura 8) para possibilitar a navegação pelo cenário, como se o usuário estivesse em um ambiente de jogo em 1ª pessoa. O *engine* UDK é gratuito para uso não comercial [UDK 2016].

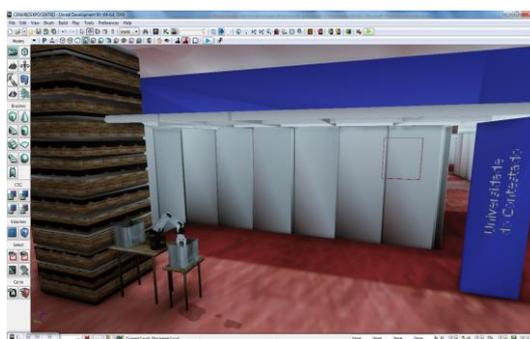


Figura 8. Cenário com animação do robô feito no engine UDK

2.5. Controle Gestual

Dois tipos de experimentos foram realizados: O 1º realiza o controle cinemático direto, o 2º faz o controle cinemático inverso. No primeiro experimento, o braço robótico procura reproduzir o posicionamento do braço, antebraço e mão do usuário que se encontra na frente do Kinect. No segundo, o atuador final do braço robótico tenta seguir a posição da cabeça do usuário em frente ao Kinect. Os experimentos foram criados com o dispositivo Kinect, da Microsoft, e com o *framework* OpenNI.

2.5.1. OpenNI

A organização denominada OpenNI (*Open Natural Interaction*) foi fundada em 2010 e, em 2014, adquirida pela Apple Inc.. Esta organização teve por objetivo melhorar a interoperabilidade dispositivos de interfaces de usuário chamados de ‘naturais’.

Uma versão da biblioteca chamada SimpleOpenNI, que trabalha em conjunto com a linguagem Processing¹ foi utilizada. OpenNI facilita muito o trabalho com a técnica de esqueletização, pois fornece os pontos de articulação e vetores correspondentes que podem ser utilizados para verificar os ângulos entre as juntas do esqueleto. Para isso primeiramente é necessário realizar um processo de calibração, que inicia com a detecção da pessoa em uma determinada posição (‘pose’). Se a calibração foi concluída com sucesso, pode-se iniciar o rastreamento (*tracking*) do esqueleto.

2.5.2. Primeiro Experimento: Cinemática Direta

O primeiro experimento realiza a leitura das articulações como ombro, cotovelo e mãos (Figura 9), e calcula os ângulos entre elas para poder então controlar o braço robótico de forma a simular nele o movimento feito pela pessoa.

O ângulo do ombro é formado entre a parte superior do braço e o torso, já o cotovelo é o ângulo criado entre a parte superior do braço e antebraço. As posições das juntas do esqueleto são obtidas através do OpenNI, que fornece um vetor indicando as posições x , y e z da junta. A distância entre as juntas pode ser obtida através da subtração entre dois vetores, e o ângulo entre as juntas pode ser obtido através do produto escalar entre os dois vetores.

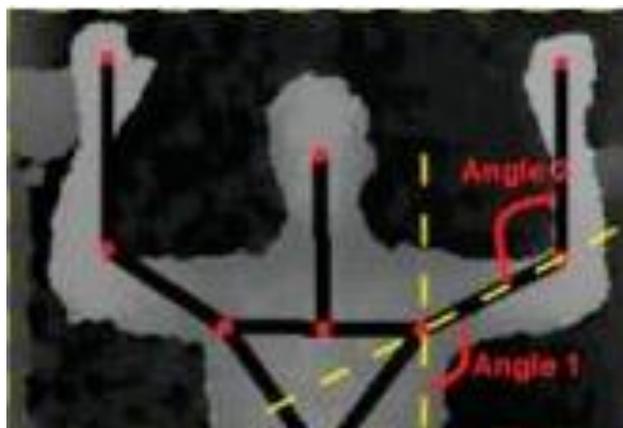


Figura 9. Processo de esqueletização da biblioteca OpenNI [Borenstein 2012]

¹ *Framework* para a linguagem Java e que se integra muito bem com a plataforma Arduino.

Os dados são então enviados através da interface USB (transmissão serial) a uma placa Arduino que ligada ao computador. Esta placa Arduino está conectada a um módulo de comunicação XBee e, através deste, envia por radiofrequência as informações relativas aos ângulos para o manipulador, tornando possível seu controle de forma remota. O XBee instalado no manipulador repassa os comandos recebidos ao Arduino que está conectado ao braço robótico. Este Arduino controla os ângulos dos servos-motores instalados nas juntas robóticas.

2.5.3. Segundo Experimento: Cinemática Inversa

O 2º experimento realiza o controle cinemático inverso, ou seja, a partir de um ponto final são calculados os ângulos necessários para o manipulador atingir a configuração necessária. Com esta técnica torna-se possível movimentar o robô somente movimentando o corpo e a cabeça, já que, através da cinemática inversa, o robô procura assumir a postura que faça com que o efetuador final alcance o ponto indicado.

A Figura 10 exhibe o rastreamento (*tracking*) da cabeça do usuário feita pelo programa (linha destacada em vermelho). O movimento realizado pela cabeça movimenta o braço robótico.



Figura 10. Controle via Kinect

2.6. Controle Remoto da Garra Robótica

Este experimento controla a garra do braço robótico através de uma luva de controle. Somente a abertura e fechamento da garra podem ser realizados.

Na Figura 11 podem ser vistos o sensor *flex* [Spectrasymbol 2016], no dedo anular, o *shield* LilyPad [Buechley 2006] XBee com um XBee sobre ele, um acelerômetro ao centro, o Arduino Lilypad (na base do dedo indicador), um módulo LilyPad que fornece a alimentação de energia, e uma bateria de polímero de lítio.



Figura 11. Luva de Controle

O sensor *flex*, ao ser curvado, tem seu valor de tensão lido pelo microcontrolador Arduino LilyPad. Este, por sua vez, envia o valor ao módulo XBee transmissor, que transmite a informação via RF. A informação enviada pela luva é recebida pelo módulo XBee conectado ao manipulador robótico, que a repassa ao Arduino ligado a ele, acionando então o servo que controla a garra robótica.

2.7. Reconhecimento de Dominós

O experimento desenvolvido consiste em identificar a posição, orientação e os valores das peças de dominó colocadas sobre a esteira. As peças podem ser colocadas na vertical, horizontal ou diagonal e o braço robótico as alinhará sempre na mesma orientação, perpendicular ao eixo de movimentação da esteira. A placa Arduino é responsável por controlar a esteira e o robô. A placa Raspberry [Richardson 2013] realiza as tarefas de visão computacional e identificação do valor da peça através de redes neurais. As peças são colocadas sobre uma esteira, esta se move até que a peça fique sob a câmera RaspiCam, que tira uma foto e a envia à Raspberry para processamento. Após isso uma rede neural identifica o valor da peça do dominó.

A comunicação entre a Raspberry Pi e o Arduino Uno foi realizada através da conexão I²C (Circuito Inter-Integrado, do inglês *Inter-Integrated Circuit*). O empecilho de realizar esta conexão é o fato do Arduino trabalhar com 5 volts nas portas de conexão, enquanto que o Raspberry Pi opera em 3,3 volts. A solução foi utilizar uma pequena placa FTDI² que realiza a conversão entre tensões, possibilitando assim que as duas placas fossem interligadas. O experimento final pode ser visto na Figura 12:

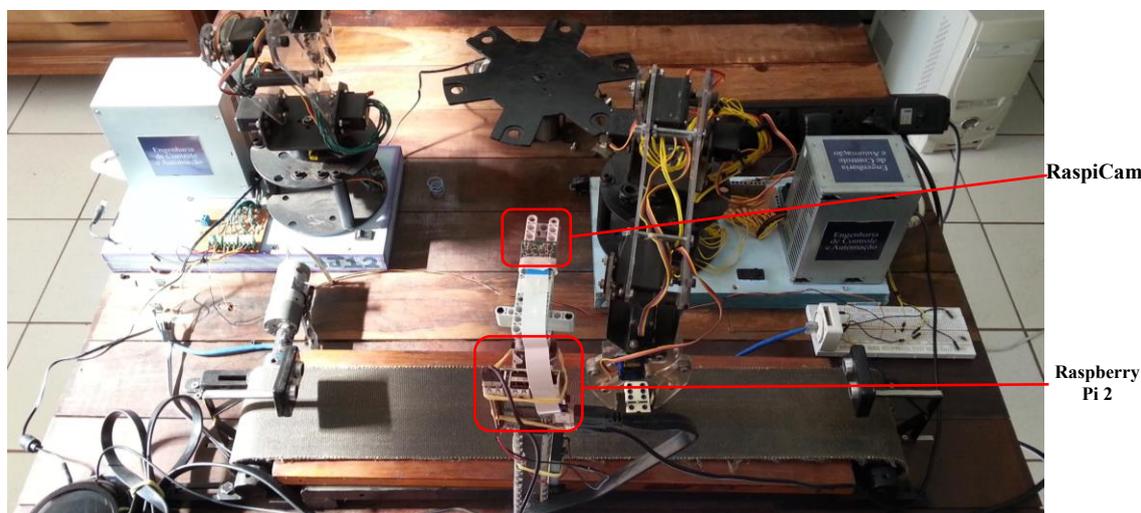


Figura 12. Experimento com Dominós

2.7.1. Análise de Imagens

As peças de dominó são posicionadas ao longo do eixo de movimentação da esteira (eixo x) com um espaçamento uniforme entre elas. Assim é possível realizar um

² Chip FTDI: Circuito integrado da empresa Future Technology Devices International capaz de converter sinais, tornando possível a comunicação entre duas tecnologias diferentes.

controle de movimentação da esteira baseado em tempo de modo a fazer que as peças de dominó fiquem posicionadas exatamente sob a câmera quando a esteira for parada.

Após tirada a foto, é realizada a segmentação da imagem, com dois resultados podendo ser vistos nas Figuras 13a e 13b. O primeiro passo da análise é selecionar uma região da imagem e fazer sua segmentação. Este corte é importante para reduzir o tempo de processamento. Os segmentos (chamados de *blobs*), estão marcados em rosa. Para selecionar o segmento correspondente a todo o dominó basta escolher o maior deles.

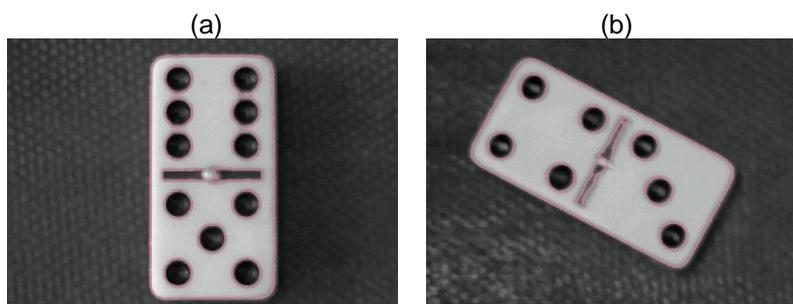


Figura 13. Imagem original do dominó sobre a esteira

O ângulo do segmento é obtido a partir do método `angle()` aplicado ao blob que representa o dominó. Este ângulo é utilizado tanto para rotacionar o segmento em software quanto para definir qual será o ângulo de rotação do servo-motor que controla o giro da garra do robô.

O método `coordinates()` da classe *blob* é usado para se obter as coordenadas x e y do segmento. Essa informação é utilizada para posicionar a garra do robô ao longo do eixo perpendicular ao de movimentação da esteira. Atualmente a garra não realiza correção de posicionamento ao longo do eixo x , logo é necessário que as peças de dominó sejam colocadas aproximadamente na mesma coordenada x sobre a esteira. Visto que o posicionamento no eixo z da garra está sempre na mesma altura, os dominós têm todos a mesma espessura e as fotos são tiradas sempre da mesma distância, basta trabalhar com os servo-motores dos dois eixos do robô para efetuar o correto posicionamento no eixo y . Os ângulos dos servos são calculados apenas com base em uma regra de três entre a largura da imagem da esteira captada pela câmera, e o quanto a garra do robô se desloca em relação a 1 grau de giro do servo.

2.7.2. Reconhecimento dos Padrões

Para realizar o reconhecimento dos padrões correspondentes aos números foram utilizadas as bibliotecas PyBrain, NumPy e matplotlib.

A tarefa de reconhecimento de imagens é sempre feita com o dominó alinhado perpendicularmente ao eixo de movimentação da esteira. Como a peça do dominó pode estar de ponta-cabeça, a rede neural foi treinada sempre com as duas versões de cada peça, totalizando 56 padrões de treinamento. Para a entrada da rede neural foi utilizado o segmento de imagem correspondente ao dominó obtido na fase de processamento de imagens. A saída da rede neural fornece um valor entre 1 e 28, codificado em binário, visto que este é o número de peças de um conjunto tradicional de dominós.

A rede foi construída manualmente com 3 camadas, do tipo supervisionada e método de treinamento *backpropagation*. A rede foi treinada até que o erro apresentado por ela fosse menor do que 0,001. Um banco de dados de treinamento de imagens foi criado, sendo que os padrões de treinamento eram formados por tuplas <imagem, padrão desejado de saída da rede>. A rede foi executada e testada. Sob condições controladas de iluminação, a rede neural conseguiu uma taxa de acerto de 100%.

3. Resultados e Discussões

A bancada tem sido utilizada regularmente no curso de Engenharia de Controle e Automação da UnC-Universidade do Contestado, campus de Curitiba-SC, nas disciplinas de Programação, Circuitos Digitais, Robótica, CLP e Projetos para Automação e o retorno dos alunos tem sido positivo. A bancada também tem funcionado como uma plataforma de testes em projetos de pesquisa envolvendo tecnologias estado-da-arte que ainda são incipientes no mercado, como o controle de equipamentos por gestos. Também torna possível a realização de testes de integração com Android, Internet das Coisas, tecnologias de transmissão sem fio como *bluetooth*, *wifi*, etc.

Vários melhoramentos estão previstos para serem realizados em trabalhos futuros. O braço robótico precisa de um novo projeto, mais robusto e capaz de realizar trabalhos com cargas maiores. É preciso também implementar um sistema de controle realimentado no braço, que hoje é inexistente, assim como rampas de aceleração e desaceleração dos motores. Atualmente o controle do braço é feito eixo a eixo, o que dificulta a programação dos movimentos. Um sistema de coordenadas de mundo deve ser implementado para facilitar o posicionamento do efetuador final no espaço 3D.

O experimento de visão computacional com dominós é hoje muito suscetível às condições de iluminação na sala, e funciona a contento somente se estiver sendo executado em ambiente controlado.

Referências

- Borenstein, G. (2012), “Making Things See”. Editora O’ Reilly Media.
- Bradski, G. and Kaehler, A. (2008) “Learning OpenCV. Computer Vision with OpenCV Library”. Sebastopol, California, Estados Unidos. O’ Reilly Media Inc.
- Buechley, L. (2006) “A construction kit for electronic textiles”. In Proceedings of the IEEE International Symposium on Wearable Computers (ISWC).
- Faludi, R. (2011) “Building Wireless Sensor Networks”. O’Reilly.
- McRoberts, M. (2011) Arduino Básico. Novatec, São Paulo.
- Richardson, M. and Wallace, S. (2013) “Getting Started with Raspberry Pi”. Sebastopol, California. O’ Reilly Media Inc.
- Spectra Symbol. “Sensor Flex”. Disponível em: <<http://lilypadarduino.org/?p=1494>> Acesso em 23 de dez. de 2016.
- UDK. “UDK Unreal Engine”. Disponível em: <<http://www.unrealengine.com/en/-udk>>. Acesso em 23 de dez. de 2016.