

PROJETO DE UM BRAÇO ROBÓTICO PARA FINS DIDÁTICOS

Lucas Kato

UFSC

Florianópolis, SC, Brasil

lucaskkato@gmail.com |

Rodrigo Braga

UFSC

Florianópolis, SC, Brasil

rodrigo.braga.ufsc@gmail.com

Ana Veronica Pazmino

UFSC

Florianópolis, SC, Brasil

ana.veronica@ufsc.br

RESUMO

O artigo apresenta o projeto e a implementação de um braço robótico manipulador para fins didáticos, como uma forma de popularizar a robótica no espaço de ciência da UFSC/Joinville, buscando alcançar o público do ensino médio, técnico e superior. Como resultados, o trabalho apresenta um braço robótico de baixa complexidade com quatro graus de liberdade do tipo antropomórfico materializado via impressora 3D, além de possuir funções similares aos braços utilizados em indústrias, como a função *teach* e uma proposta de implementação de movimentos retilíneos, desenvolvida a partir de conceitos de cinemática de robôs manipuladores.

Palavras-chave: Braço robótico, robótica; ensino.

ABSTRACT

The paper presents the design and implementation of a robotic manipulator arm for teaching purposes, as a way to popularize robotics in space science UFSC / Joinville, seeking to achieve the public high school, technical and higher. As a result, the paper presents a robotic low complexity arm with four degrees of freedom anthropomorphic type materialized via 3D printer, besides having similar functions to the arms used in industries such as teach function and a proposal to implement rectilinear movements developed from handlers robot kinematics concepts.

Key-words: robotic arm, robotic, teach.

1. INTRODUÇÃO

Nos últimos anos, com o desenvolvimento tecnológico, redução de custos de componentes eletrônicos e crescimento da mão de obra especializada [1], o campo da robótica vem se popularizando e se difundindo cada vez mais nos países em desenvolvimento, principalmente nas indústrias automobilísticas. Segundo a IFR (*International Federation of Robotics*) [2], em 2013, cerca de 70.000 unidades de braços robóticos foram vendidas no mundo e, em 2011, 60.000. Como reflexo dessa popularização, muitos grupos de pesquisadores e interessados no assunto começaram a investir nessa área através do uso dos robôs como instrumento de auxílio para suas atividades bem como para o estudo dos componentes mecânicos e eletrônicos, os quais são vistos no ensino técnico e superior.

Nessa perspectiva, o projeto contemplou áreas do Design e Engenharia de Controle e Automação, gerando um protótipo com alternativas no modo que pode ser utilizado no ensino.

2. MANIPULADORES ROBÓTICOS

A implementação da proposta para a movimentação retilínea do efetuator ou órgão terminal do braço robótico necessita de conceitos de cinemática de manipuladores robóticos, tais como movimentos de corpo rígido, cinemática direta e inversa. Estes conceitos são comumente vistos em disciplinas de robótica nos cursos de graduação. Pelo fato de envolverem teorias

matemáticas que não são triviais para uma correlação com a prática, muitos estudantes se sentem desmotivados com o assunto, o que poderia ser contornado com a observação e análise de um modelo real de um braço robótico apresentando como ele se comporta de acordo com certos parâmetros dimensionais do robô.

2.1. MOVIMENTOS DE CORPO RÍGIDO

Para o equacionamento da cinemática de braços robóticos, necessita-se, primeiramente, ter uma referência ou sistemas de coordenadas para representar as posições e orientações de corpos rígidos. Também é de grande relevância saber as transformações de coordenadas entre esses sistemas, para que vetores que representam posições, velocidades e acelerações, dados em um determinado sistema de coordenadas, possam ser representados em outros sistemas de coordenadas.

A seguir será apresentada a teoria básica de operações de rotação e translação entre dois sistemas de mesmo tipo, além da transformação homogênea.

2.1.1. ROTAÇÕES

Dado um ponto P no espaço tridimensional, deseja-se relacionar as coordenadas deste ponto no sistema de coordenadas $O(x_1, y_1, z_1)$ em relação ao sistema fixo inercial $O(x_0, y_0, z_0)$, com representado na Figura 1.

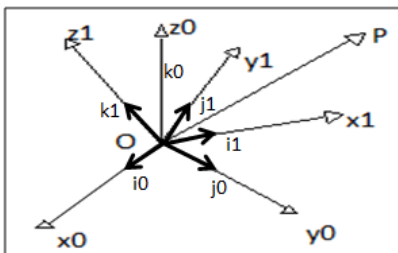


Figura 1 – Sistemas dextrórgos

Considerando os vetores unitários: (i_0, j_0, k_0) e (i_1, j_1, k_1) , pode-se representar o ponto P em relação aos dois sistemas.

$$P_0 = P_{0x}i_0 + P_{0y}j_0 + P_{0z}k_0 \quad (1)$$

$$P_1 = P_{1x}i_1 + P_{1y}j_1 + P_{1z}k_1 \quad (2)$$

Como (1) e (2) representam o mesmo ponto, pode-se escrever:

$$P_{0x} = P_0 \cdot i_0 = P_1 \cdot i_0$$

$$P_{0y} = P_0 \cdot j_0 = P_1 \cdot j_0$$

$$P_{0z} = P_0 \cdot k_0 = P_1 \cdot k_0$$

Substituindo esse sistema por (2):

$$P_{0x} = P_{1x}i_1 \cdot i_0 + P_{1y}j_1 \cdot i_0 + P_{1z}k_1 \cdot i_0$$

$$P_{0y} = P_{1x}i_1 \cdot j_0 + P_{1y}j_1 \cdot j_0 + P_{1z}k_1 \cdot j_0$$

$$P_{0z} = P_{1x}i_1 \cdot k_0 + P_{1y}j_1 \cdot k_0 + P_{1z}k_1 \cdot k_0$$

Ou

$$P_0 = R_0^1 P_1 \quad (3)$$

Sendo que a matriz R_0^1 representa a matriz de rotação das coordenadas do ponto P do sistema $O(x_1, y_1, z_1)$ em relação ao sistema fixo $O(x_0, y_0, z_0)$, dada por:

$$R_0^1 = \begin{bmatrix} i_1 \cdot i_0 & j_1 \cdot i_0 & k_1 \cdot i_0 \\ i_1 \cdot j_0 & j_1 \cdot j_0 & k_1 \cdot j_0 \\ i_1 \cdot k_0 & j_1 \cdot k_0 & k_1 \cdot k_0 \end{bmatrix}$$

As colunas da matriz são os cossenos diretores do sistema 1 em relação ao fixo.

2.1.2. COMPOSIÇÃO DE ROTAÇÕES

Na explicação anterior, apenas se considerou dois sistemas de coordenadas, como se o fixo estivesse na base do robô e o outro, no eixo da junta *shoulder* do braço. Só que um manipulador possui mais juntas, consequentemente, mais sistemas (x_n, y_n, z_n) . Por isso, são necessárias as composições de matrizes de rotação.

Considerando um novo sistema $O(x_2, y_2, z_2)$, o mesmo ponto P pode agora ser representado de três formas:

$$P_0 = R_0^1 P_1$$

$$P_0 = R_0^2 P_2 \quad (4)$$

$$P_1 = R_1^2 P_2 \quad (5)$$

Substituindo (5) em (3):

$$P_0 = R_0^1 R_1^2 P_2$$

Desse modo, temos:

$$R_0^2 = R_0^1 R_1^2$$

Então, para se transformar as coordenadas do ponto P do sistema 2 ao fixo, deve-se primeiro transformá-lo para o sistema 1, através de R_1^2 ,

para então, transformar para o sistema fixo com R_0^1 .

Com vários sistemas, é possível usar uma forma generalizada de composições de rotações, em que:

$$R_0^n = R_0^1 R_1^2 \dots R_{n-1}^n$$

2.1.3. TRANSFORMAÇÕES HOMOGÊNEAS

Primeiramente, considera-se uma transformação homogênea quando se agrega um parâmetro a uma matriz representativa de um vetor tridimensional [16]. Por exemplo: dado o vetor $v = (x, y, z)^T$, com o incremento de um parâmetro "d", $\hat{v} = (x, y, z, d)^T$, \hat{v} é uma representação homogênea de v .

Para completar as informações de um ponto em relação a outros sistemas de coordenadas são necessárias aquelas que dizem respeito à translação.

Considera-se um sistema com apenas uma translação ao sistema fixo de origem, como na figura 2, e distância entre as origens do sistema representada pelo vetor d_0^1 .

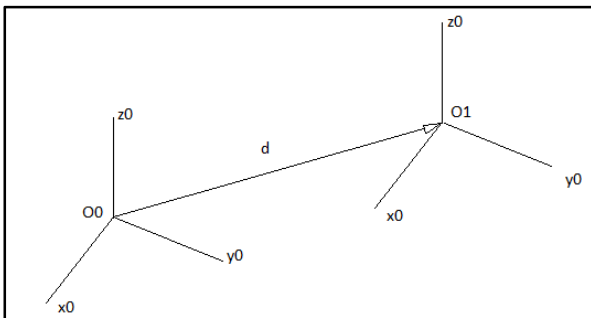


Figura 2 – Translação de coordenadas

Assim, sua representação homogênea será da seguinte forma:

$$P_0 = R_0^1 P_1 + d_0^1$$

Ou, em forma matricial:

$$\begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} P_{1x} \\ P_{1y} \\ P_{1z} \end{bmatrix} + \begin{bmatrix} d_{0x}^1 \\ d_{0y}^1 \\ d_{0z}^1 \end{bmatrix}$$

$$\begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_{0x}^1 \\ r_{21} & r_{22} & r_{23} & d_{0y}^1 \\ r_{31} & r_{32} & r_{33} & d_{0z}^1 \end{bmatrix} \begin{bmatrix} P_{1x} \\ P_{1y} \\ P_{1z} \\ 1 \end{bmatrix}$$

Então se acrescenta mais uma linha para que a matriz seja quadrada.

$$\begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_{0x}^1 \\ r_{21} & r_{22} & r_{23} & d_{0y}^1 \\ r_{31} & r_{32} & r_{33} & d_{0z}^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{1x} \\ P_{1y} \\ P_{1z} \\ 1 \end{bmatrix}$$

Ou, simplificando:

$$\tilde{P}_0 = A_0^1 \tilde{P}_1$$

Sendo que a matriz A (transformação homogênea) possui parâmetros de rotação e translação.

Com esses conceitos, é possível discorrer sobre as cinemáticas de braços robóticos.

Considerando q_i como sendo a coordenada generalizada das posições de juntas (θ_i, d_i) , é fundamental a relação entre esses parâmetros de juntas e o do órgão terminal. Isso é feito através da cinemática direta e inversa, representado na Figura 3.

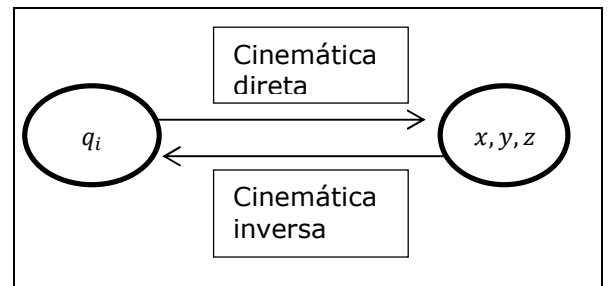


Figura 3 - cinemática direta e inversa

Estas conversões são de suma importância para o controle de movimentos do robô, já que, normalmente, as trajetórias requeridas estão em coordenadas cartesianas e as referências para os motores, em ângulos (graus ou radianos).

2.2. CINEMÁTICA DIRETA

O objetivo deste procedimento é encontrar os valores de posição e orientação do órgão terminal em função das variáveis de juntas. Para isso, utilizam-se as matrizes de transformação homogênea.

Um dos métodos mais conhecidos e utilizados é o de Denavit – Hartenberg (DH) [4], o qual fornece passos para gerar um modelo geométrico simplificado do braço com suas dimensões e

coordenadas de cada junta e do efetuador. Com base nisso, formula-se uma tabela, cujos parâmetros são utilizados para construir as matrizes homogêneas necessárias para encontrar os valores de orientação e posição.

2.3. CINEMÁTICA INVERSA

O processo da cinemática inversa é o contrário da direta, ou seja, a partir dos dados de posição e orientação do efetuador, deseja-se conhecer informações sobre as juntas.

Por exemplo, através da cinemática direta se obtém equações para um determinado tipo de manipulador, indicando parâmetros de rotação e translação. A título de ilustração, algumas dessas equações são apresentadas em seguida.

$$C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) = r_{11}$$

...

$$C_1S_2d_3 - S_1d_2 - d_6(C_1C_2C_4S_5 + C_1C_5S_2 - S_1S_4S_5) = d_x$$

...

As variáveis da direita da equação representam os valores de posição e orientação do órgão terminal. Os valores r_{ij} são elementos da matriz de rotação de uma transformação homogênea e d_x é o elemento relativo à translação do sistema de coordenadas em relação ao sistema inercial (base).

As letras 'c' e 's' representam os cossenos e senos, respectivamente. E os números de 1 a 6 representam a junta em questão.

Enquanto na forma direta sempre se obtém uma única solução, na inversa pode se encontrar uma, várias ou nenhuma solução, além de ser muito difícil de resolver o equacionamento, já que se trata de um sistema não linear.

Porém, existem algumas saídas. É possível solucionar numericamente ou geometricamente, sendo este último, o mais simples e intuitivo, o qual é utilizado neste trabalho.

O fato de poder existir múltiplas soluções implica na existência de várias configurações possíveis do braço para posicionar o terminal em um ponto. Quanto maior o número de graus de liberdade, maiores são essas possibilidades.

3. DESENVOLVIMENTO DO BRAÇO

Atualmente existem muitos produtos e projetos destinados à robótica educacional, porém poucos dão a devida ênfase aos métodos utilizados para mover um robô real (industrial), apresentando poucos detalhes de seu funcionamento ou de como é programado para realizar certos movimentos.

Em vista disso, este projeto busca contribuir nesse ramo com o desenvolvimento de um robô manipulador capaz de popularizar a ciência e a tecnologia, auxiliando no ensino fundamental, técnico e superior. Isto pode ocorrer pela utilização do robô como um meio facilitador para outras atividades educativas ou através da programação, incluindo funções similares aos braços robóticos industriais. Na programação do microcontrolador são explícitos de forma didática os detalhes conceituais de robótica para certos movimentos, como os de tipo retilíneos.

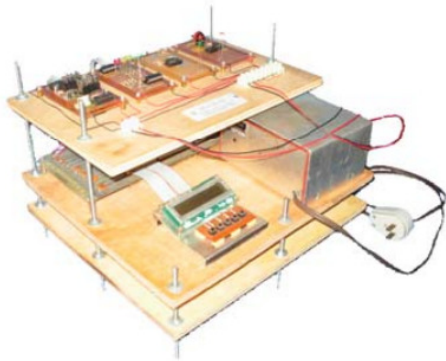
O Trabalho se inicia com uma pesquisa de mercado, detalhada no tópico seguinte.

3.1. ANÁLISE SINCRÔNICA

Esta etapa faz parte da fase de análise da metodologia de Bonsiepe [5], cujo objetivo é levantar informações sobre os produtos similares ao que se pretende projetar, que estão disponíveis no mercado, e outros que são projetos acadêmicos. Desta análise são obtidos dados importantes para o desenvolvimento do produto no que tange à especificação de requisitos, destacando suas principais características, qualidades e defeitos. Para isso foram analisados três kits de robótica, com a possibilidade de se construir diversas estruturas, e três kits com estrutura bem definida – braço robótico.

Os seguintes kits foram analisados: Lego Mindstorms EV3 [6]; Linxmotion AL5D [7]; Modelix kit 411 [8]; RoboFácil [9]; Hajime [10] e EasyArm DS [11].

Para cada um deles foi montado um quadro com informações gerais e técnicas, como no modelo RoboFácil, apresentado no quadro 1.

Quadro 1 – Informações do RoboFácil


Informações gerais
Nome: RoboFácil
Fabricação: Nacional
Material: NI (Não informado)
Preço estimado: R\$ 313,30
Informações técnicas
Arquitetura: Aberta (hardware)
Linguagem de programação: Visual
Microcontrolador: Intel MCS-51
Nº de portas (sinal de saída): 16
Nº de portas (sinal de entrada): 16
Estrutura: NA (Não aplicável)
Observação: Produto não comercializado

Com base nas características de cada modelo, é necessário compará-los através de um quadro contendo alguns atributos com seus respectivos pesos, já que cada um deles tem graus de relevância distintos.

Os seguintes atributos foram comparados:

- **Fabricação:** podendo ser nacional ou internacional. O maior peso vai para os produtos nacionais, já que os mesmos podem ter custo mais baixo e são de grande importância para o desenvolvimento econômico e incentivo à pesquisa no país;

- **Material:** MDF, plástico (PLA¹) ou alumínio. O MDF possui maior importância devido ao seu custo (R\$/m²) ser baixo em relação ao alumínio²

¹ PLA – Polylactic acid

² Segundo [13], o valor do MDF de 3 mm de espessura é de 11,59 R\$/m², enquanto que, segundo [14] e [15], uma chapa de alumínio de mesma espessura é em torno de 32 R\$/m².

e de fácil usinagem. Apesar de o PLA ser o mais caro, possui grande flexibilidade para a criação de peças complexas através da impressão 3D, ficando com o peso intermediário.

- **Preço:** baixo para valores menores que R\$ 500,00 médio para valores entre R\$ 501,00 e R\$ 1500,00 e alto para valores acima de R\$ 1501,00.

- **Arquitetura:** aberta ou fechada [12]. Somente foi levada em consideração a arquitetura de hardware, já que não se obtiveram informações sobre as de software para todos os modelos;

- **Linguagem de programação:** visual – fluxograma, icônica ou convencional. Neste caso considera-se icônica aquela em que não se utiliza fluxogramas e não é convencional – aquelas que necessitam digitar textualmente o comando;

- **Estrutura:** Modular ou não modular. Modular é quando o produto pode assumir várias configurações estruturais – formas [12];

Com esses atributos, montou-se o quadro 2 que indica os pesos para as variações de cada atributo, apresentados em seguida. O valor 3 indica maior importância entre as possibilidades de cada atributo e o 1, o menor. Os produtos que não apresentam alguma informação são representados no quadro com o símbolo “---”.

Com base no quadro anterior, montou-se outro que relaciona os produtos com os atributos e seus respectivos pesos, os quais indicam o grau de relevância de cada atributo.

Por haver kits com características bem distintas, resolveu-se separá-los em kits sem forma definida, que são os da *Lego*, *Modelix* e *RoboFácil*, e kits com forma de braços robóticos do tipo antropomórfico.

Quadro 2 – Quadro de análise de atributos

Pesos		3	2	1
Gerais	Fabricação	Nacional	Não comercializado	Importado
	Material	MDF	Plástico	Alumínio
	Preço	Baixo	Médio	Alto
Técnicos	Arquitetura	Aberta	---	Fechada
	Linguagem	Visual - fluxograma	Icônica	Convencional
	Estrutura	Modular	---	Não modular

Na penúltima linha é calculada a soma dos valores para cada produto, considerando os pesos. Para os produtos que não contêm informação em determinado atributo, este não é levado em consideração na soma.

Por último é feita uma classificação, em que o melhor é aquele cuja pontuação é maior. O quadro 3 apresenta essas relações de produtos, atributos e pesos.

Quadro 3 – Comparação dos produtos

	Peso	Kits sem forma definida				Kits de braços robóticos		
		Nome	Lego	Modelix	Robo Fácil	Lynxm.	Hajime	Easy Arm
Gerais	1	Fabricação	1	3	2	1	2	3
	1	Material	2	1	---	1	2	3
	3	Preço	1	1	3	2	---	3
Técnicos	3	Arquitetura	1	1	3	1	3	---
	3	Linguagem	2	3	2	---	2	---
	2	Estrutura	3	3	---	1	1	1
	-	Soma	21	25	26	13	21	17
Classificação			3º	2º	1º	3º	1º	2º

Nessa análise obtêm-se dois melhores classificados (RoboFácil e Hajime), um entre os kits sem forma definida e outro com a estrutura de um braço robótico antropomórfico, respectivamente.

Para cada um dos melhores colocados é feita uma lista de verificação, cujo objetivo é pontuar os aspectos positivos e negativos, para, posteriormente, levantar os requisitos do projeto.

Por exemplo, no projeto RoboFácil, o maior destaque está na disponibilização da arquitetura de hardware, com descrições de como foi implementado o circuito eletrônico principal, os *displays*, portas paralelas, porta serial, conversores DAC³ e ADC⁴, plug-ins de controle dos motores de passo, *led* e sensores.

Em contrapartida, a estrutura possui grandes dimensões comparadas às outras e não possui um modelo de robô como demonstração que possa ser utilizado no sistema.

No Hajime, por se tratar também de um projeto acadêmico, o usuário deve buscar todo o material para a construção do braço robótico, além de utilizar a porta paralela, que vem

entrando em desuso. Esses pontos são os que mais se destacam entre os negativos. Porém, como vantagem, o Hajime não possui circuito eletrônico e a interface é simples e intuitiva, podendo ser utilizada nos níveis de ensino mais básicos.

Com base na lista de verificação, levanta-se uma lista de necessidades explicitando as características de cada modelo que podem ser mantidas e aquelas que podem ser aperfeiçoadas. Considerando-se, também, o acréscimo de novas funcionalidades para cada um dos melhores modelos de cada categoria – RoboFácil e Hajime.

A partir disso formulam-se, primeiramente, os requisitos de forma mais abrangente, que são:

- Ter arquitetura de hardware/software aberta;
- Ter Baixo custo;
- Possuir estrutura física compacta, leve e resistente;
- Vir com manual de instrução;
- Ter funcionalidades para o aprendizado em programação e robótica;

Para então, formular os demais requisitos, conforme o tópico seguinte.

³ DAC – Conversor digital analógico.

⁴ ADC – Conversor analógico digital.

3.2. REQUISITOS TÉCNICOS E ESTRUTURAIS

Nesta etapa, com base na lista de necessidades, analisam-se como aqueles pontos podem ser colocados em prática. Os resultados disso são apresentados nos requisitos de projeto conforme o quadro 4, através de especificações estruturais e técnicas. Devido à quantidade de informações do quadro, o mesmo é apresentado apenas com alguns atributos.

Neste quadro, a coluna intitulada Objetivo indica o valor, dimensão, quantidade ou material do requisito em questão.

Os requisitos estruturais abrangem os aspectos mecânicos do braço, enquanto que os técnicos, os aspectos eletrônicos e de software.

Quadro 4 – Especificações estruturais e técnicas

	Requisitos	Objetivo	Unidade	Classificação
Estruturais	Juntas	3	nº	Obrigatório
	Material da junta	PLA	---	Desejável
	Base	13x10x4	cm	Desejável
	Material da garra	ABS	---	Desejável
	Acoplamentos motor/elo/junta	5	nº	Obrigatório
Técnicos	Documentação	1	nº	Desejável
	Plataforma de desenvolvimento	1	nº	Obrigatório
	Leitor cartão SD	1	nº	Obrigatório
	Servomotor	5	nº	Obrigatório
	Interface de Programação	1	nº	Obrigatório

3.3. REQUISITOS DE SOFTWARE

Para a especificação do software é utilizada parte da metodologia de Waslawick [16] referente à especificação de requisitos de sistemas de informação.

Nesta especificação os requisitos são classificados em funcionais, os quais expressam as funções principais do sistema, e suplementares, que declaram as restrições tecnológicas ou lógicas do sistema, sendo organizadas em atributos de implementação, interface e empacotamento.

Os requisitos funcionais são:

- Identificar função;
- Executar função;
- Controlar motores;

E os suplementares:

a. Usabilidade:

O sistema virá com um manual de instrução, apresentando como se deve montar a estrutura

física, as conexões eletrônicas e uma breve explicação das funções do software.

b. Implementação:

- Linguagem ACL (*Advanced Control Language*)

O projeto do braço robótico foi inspirado no modelo Scrobot-ER VII [17], cuja linguagem de programação é o ACL que foi desenvolvida pela empresa *Eshed Robotec*.

Baseada na linguagem *Basic*, a qual foi desenvolvida com propósitos educativos, possui poucas funções e é de alto nível, tornando-se propícia para os fins desse projeto.

- Linguagem C

Utilizada para programar a plataforma microcontroladora.

c. Interface:

A interface conterá duas janelas, uma será o *Hyper Terminal*⁵ e a

⁵ Hyper Terminal: antigo aplicativo de comunicações do Sistema Operacional Windows. Ele é utilizado para o envio de

outra, um editor de texto para digitar o código na linguagem ACL.

d. Empacotamento

Os arquivos executáveis e o código fonte do *firmware* devem estar disponíveis na internet.

3.4. ANÁLISE FUNCIONAL

A ferramenta permitiu montar uma árvore de funções do novo braço robótico, conforme a figura 4. Pelo fato de se apresentar como as funções irão funcionar, pode-se considerar essa etapa de criatividade. Já que permite identificar as funções necessárias e posteriormente as estruturas físicas necessárias. Para a função principal global que é mover o braço robótico.

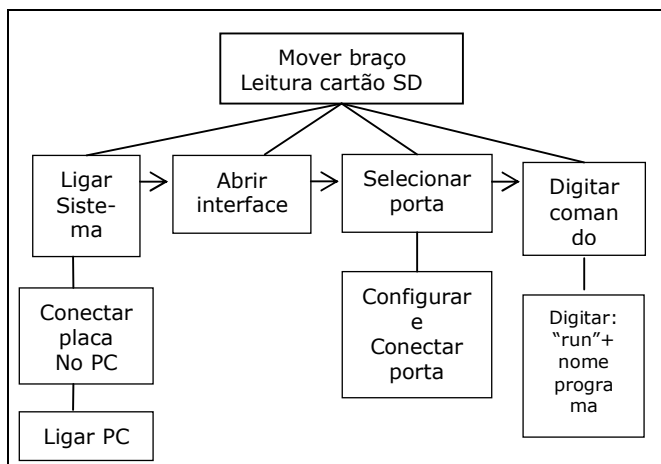


Figura 4 - Fluxo de ações e funções (mover via leitura SD)

Existem duas possibilidades para alcançar o principal objetivo: através de um terminal, em que o usuário move cada motor via teclado ou através da leitura de um código pré-definido em um arquivo de texto do software que é armazenado em um cartão SD, como descrito na figura anterior.

4. GERAÇÃO DE ALTERNATIVAS

Da mesma forma que foram avaliados os diferentes produtos e projetos de robótica educacional (análise sincrônica), torna-se fundamental avaliar as diferentes partes que irão

compor os subsistemas do projeto em questão, levando em conta as características dos mesmos e os requisitos do produto.

Assim, foram analisadas as possibilidades para as quatro principais partes do braço manipulador: Plataforma microcontroladora, que faz parte do robô; motor elétrico; IDE e linguagem de programação.

4.1. PLATAFORMA MICROCONTROLADORA

Trata-se de um componente fundamental para o projeto, já que é ela que faz a integração entre os motores e a interface de programação.

Foram analisadas três alternativas para o projeto, as quais são: Arduino [18], Raspberry Pi [19] e Beaglebone [20].

Os modelos comparados possuem características distintas, porém, o *Arduino* leva uma grande vantagem devido a sua difusão no mercado brasileiro, com vários modelos nacionais, já que sua plataforma de hardware é aberta.

A primeira versão do braço robótico não utilizará *encoders* ópticos, sendo desnecessária a utilização de microcontroladores com alta frequência de *clock*, presentes no Raspberry e no Beaglebone. Com isso, a Plataforma Arduino se torna suficiente para a aplicação.

4.2. ACIONAMENTO

Parte indispensável que compõe os mecanismos robóticos são os motores, podendo ser de acionamento elétrico, hidráulico ou pneumático, cada um deles tem sua particularidade, com vantagens para determinadas aplicações.

Neste projeto, por ter fins didáticos, não se necessita de grande esforço e alta velocidade para movimentar cargas, sendo, portanto, o mais adequado os motores de passo, segundo [21]. Entre os motores elétricos, existem outros que podem se adequar ainda mais ao projeto.

O que se necessita é de um motor que receba como sinal de entrada um ângulo de referência e que o eixo do motor se mova para a referência recebida, ou seja, um servo-motor cumpre com o requisito. Tal dispositivo nada mais é que um motor de corrente contínua com um potenciômetro e um sistema de controle. No mercado existem modelos extremamente simples e baratos, como os chamados *micro servos*.

A desvantagem desse acionamento é o torque máximo, que de acordo com a especificação [22], é de 1.3 kg.cm.

4.3. INTERFACE DE PROGRAMAÇÃO

O intermédio entre o mecanismo robótico e o usuário é feito através da interface de programação, que é a área onde o usuário comandará os motores através de comandos e funções pré-definidos. Essa interface foi construída com a interface de desenvolvimento Delphi, a qual tem como linguagem de programação o *Object Pascal*. Um dos maiores motivos que levou à escolha desse ambiente integrado de desenvolvimento foi a experiência dos membros do laboratório com o uso desta ferramenta, que apresenta grandes vantagens, tais como:

- Linguagem de alto nível orientada a objeto.
- Integração com a API do Windows, o que permite a criação de programas que exploram ao máximo os recursos do sistema operacional.
- Compilador que gera arquivos executáveis nativos, ou seja, em código de máquina, tornando-o extremamente rápido e com proteção do código fonte.

4.4. LINGUAGEM DE PROGRAMAÇÃO

Para o acionamento dos motores, utilizam-se funções em linguagem C que são utilizadas na programação da plataforma *Arduino*. Porém, devido à complexidade desta linguagem, para leigos no assunto da programação, torna-se a menos indicada para fins didáticos, sendo interessante o uso de uma linguagem mais simples e condizente com a realidade dos manipuladores robóticos.

O modelo que serviu de inspiração para o desenvolvimento deste trabalho foi o Scorbob – ER VII. Esse robô utiliza a linguagem ACL (*Basic*) pela qual o usuário realiza as operações com o braço robótico.

Assim, torna-se necessária a implementação de um interpretador ACL para linguagem C.

Na prática, o usuário, através de um terminal, deve inserir os comandos em ACL. Em seguida a interface de programação (contém o terminal) envia os comandos digitados para o

interpretador, presente no microcontrolador do *Arduino*.

5. DESENVOLVIMENTO DE SOFTWARE

Neste item são abordados os detalhes de implementação do projeto, com ênfase na criação da interface de programação, *firmware* e um detalhamento de como foi feita a geração de trajetória do órgão terminal.

Para cada função, levantaram-se os pontos essenciais relativos aos parâmetros de entrada e saída, detalhando-se com um fluxograma as etapas internas da função, como criação de variáveis, invocação de outra função existente, laços de repetição e etc.

5.1. INTERFACE DE PROGRAMAÇÃO

O objetivo da interface é permitir comunicação entre o usuário e o microcontrolador de forma simples. Os principais campos dela são: opção de configuração de porta serial, terminal, editor e execução rápida de programas do cartão SD.

No terminal, conforme se digita um caractere, o mesmo é enviado para o microcontrolador, que fará o tratamento dos dados recebidos.

No editor, a diferença é que os dados são enviados quando o programa em ACL estiver completo e validado (sem erros de sintaxe, por exemplo). Então, os dados são encapsulados em caracteres especiais (colchetes) e enviados para o *firmware*.

A versão final da interface é apresentada na imagem seguinte.

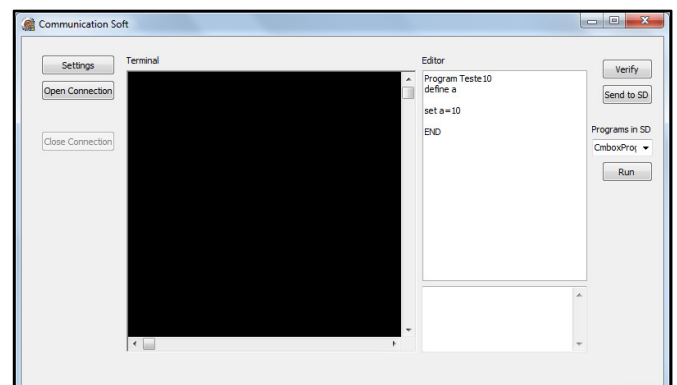


Figura 5 – Interface de programação

O Delphi permitiu a criação da interface em poucas semanas devido aos inúmeros componentes já existentes, como os botões, terminal, editor de texto e etc. O que favoreceu o bom andamento do projeto.

5.2. FIRMWARE INTERPRETADOR

A principal função do interpretador é invocar as funções da linguagem utilizada pelo microcontrolador através de funções em ACL, as quais são utilizadas pelo usuário e que contém outras funções não nativas, sendo um ACL adaptado.

A medida que o *firmware* recebe os dados da interface, as funções são identificadas junto com os seus respectivos argumentos quando for o caso, gerando, também, tratamento de exceção quando algo estiver incorreto.

Existem dois estados do interpretador:

a. Modo echo: Os dados são transferidos a partir do terminal da interface de programação.

b. Modo download: Neste caso o usuário escreve um código no editor da interface e o envia para o *Arduino*. O código não é mostrado no terminal e é gravado em um cartão SD.

As funções em ACL englobam aquelas que são indispensáveis nas linguagens de programação, como as de definição de variáveis e alocação de valores àquelas, laços de repetição (if e for), definição de variáveis de posição do robô, função para mover o robô via terminal e programas do cartão SD, entre outras.

No tópico seguinte é apresentada a proposta de uma função para movimentar o órgão efetuador (garra) do braço em linha reta.

5.3. PROPOSTA: MOVIMENTOS RETILÍNEOS

Um ponto importante a ser observado é que todos os cálculos devem ser feitos com base no sistema de coordenadas cartesianas. Como os campos da variável de posição que indicam os ângulos dos eixos de cada motor não estão neste sistema de coordenadas, os mesmos podem ser convertidos utilizando a Cinemática Direta com o método de Denavit-Hartenberg, o qual obtém as coordenadas cartesianas partindo de valores de juntas, que normalmente são do tipo angular. Este método pode fornecer parâmetros de posição e orientação de cada junta, mas o objetivo é obtê-los somente do efetuador os valores de posição.

O método se baseia na estrutura física do braço, como é mostrado na figura 6, na qual se

cria um modelo simplificado do real, com suas dimensões em mm e coordenadas cartesianas de cada junta.

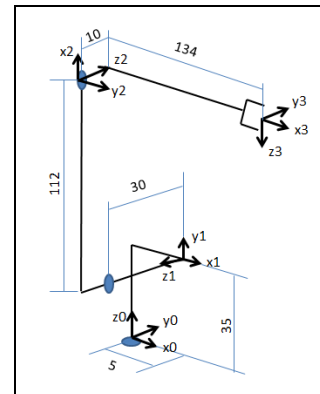


Figura 6 – Modelo do braço robótico

Com esses dados pode-se montar a tabela de Denavit-Hartenberg para a obtenção dos valores de posição do órgão terminal, como apresentado em seguida.

Tabela 1 – Parâmetros DH

Junta	θ_i	a_i	α_i	d_i
1	θ_1	90	5	35
2	θ_2	180	112	40
3	θ_3	-90	134	10

Com esses valores determinam-se as matrizes de transformação homogênea, as quais apresentam os parâmetros de posição e orientação das coordenadas cartesianas de duas juntas consecutivas:

$$A_{i-1}^i = R_{z,\theta_i} T_{z,d_i} T_{x,a_i} R_{x,\alpha_i}$$

A partir disso pode-se encontrar a matriz de transformação homogênea total indireta, que mostra a posição e orientação do órgão terminal em relação ao sistema de coordenadas na base.

$$H_0^n = A_0^1 A_1^2 \dots A_{n-1}^n$$

Também é possível encontrar a mesma matriz através do modo direto, fazendo uma análise de projeção vetorial. Os elementos da matriz

(apresentada em seguida) entre a 1ª e 3ª linha e coluna representam os cossenos diretores dos vetores do efetuador em relação ao sistema fixo inercial e na última coluna estão os valores de posição do efetuador.

$$H_0^n = \begin{bmatrix} \cos(x_n, x_0) & \cos(y_n, x_0) & \cos(z_n, x_0) & x_0 \\ \cos(x_n, y_0) & \cos(y_n, y_0) & \cos(z_n, y_0) & y_0 \\ \cos(x_n, z_0) & \cos(y_n, z_0) & \cos(z_n, z_0) & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim, igualam-se as duas equações para obter os componentes de posição da garra em relação à coordenada da base, os quais são x_0 , y_0 e z_0 .

Com isso, são feitos os seguintes passos:

a. Encontram-se as coordenadas cartesianas de duas variáveis de posição. Em cada variável estão os valores em graus de cada eixo de cada junta do robô. Para isso é utilizado o método DH para fazer essa transformação.

Para encontrar essas coordenadas é necessário utilizar as equações com variáveis simbólicas fornecidas pelo método de DH que foram feitas a partir do Matlab aplicando a função *syms*.

b. Gera-se uma equação tridimensional para a reta que liga esses pontos.

c. Para cada ponto, exceto os dos extremos da linha, realiza-se o procedimento inverso para encontrar os valores em ângulos necessários para cada junta. Isso é feito com a Cinemática Inversa pelo método geométrico.

O principal problema encontrado nesse ponto é a redundância, já que para cada coordenada pode existir pelo menos duas configurações possíveis (*elbow down e up*) para o manipulador, com exceção nas singularidades, em que o braço fica completamente "esticado".

Outro fator limitante que deve ser considerado são as restrições do volume de trabalho do braço, ou seja, é possível ter duas variáveis de posição, mas devido à limitação desse volume, não é possível traçar uma linha reta com o órgão terminal.

Uma forma simples de contornar esse problema é fazer com que o robô atue apenas na

posição *elbow down* ou cotovelo para baixo, dessa forma é excluída a redundância.

A determinação dos ângulos das juntas inicia com o do eixo *elbow* ou junta 3 (J3) denominado θ_3 como pode ser visto na figura 7, que representa a vista geométrica lateral do braço.

Já em posse dos valores das coordenadas P_x , P_y e P_z , calculados no passo anterior, os seguintes cálculos são feitos para encontrar o parâmetro θ_3 :

$$L = \sqrt{(P_x - 5)^2 + (P_z - 35)^2}$$

Os valores 5 e 35 são as dimensões presentes na estrutura do eixo x e z, respectivamente, porém são excluídas para que a origem da coordenada fique na junta 2 ou *shoulder*.

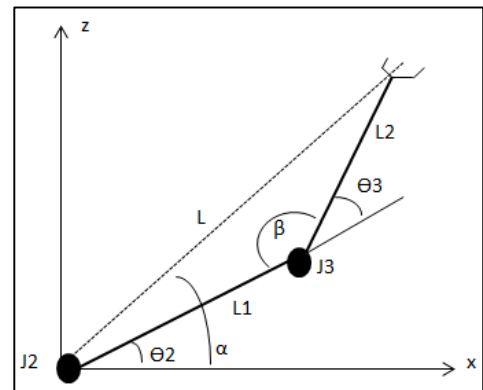


Figura 7 – Vista lateral do braço sem a base

A variável β é calculada com a lei dos cossenos:

$$\beta = \cos^{-1}\left(\frac{-L^2 + L_1^2 + L_2^2}{2L_1L_2}\right)$$

Portanto:

$$\theta_3 = \pi - \beta$$

Observação: o motor do eixo 3 atua de 0 a 180 graus, mas geometricamente, θ_3 pode variar de -90 a 90 graus, sendo necessário fazer as devidas conversões no programa. Exemplo: se θ_3 for -90, o sinal que deve ser enviado ao motor é de 0 graus.

O processo é semelhante para encontrar o ângulo referente à junta 2 (θ_2).

$$\alpha = \tan^{-1}\left(\frac{P_z - 35}{P_x - 5}\right)$$

$$\alpha = \theta_3 + \theta_2, \text{ logo:}$$

$$\theta_2 = \alpha - \theta_3$$

Para o cálculo de θ_1 , referente à base, utiliza-se a vista superior (Figura 8).

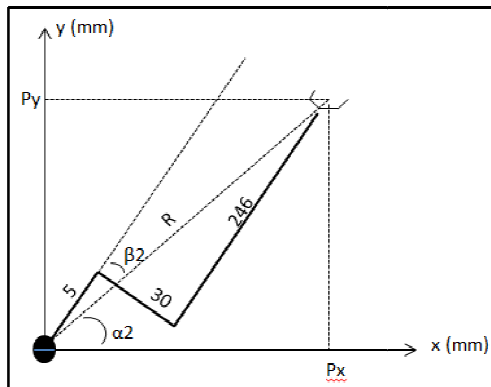


Figura 8 – Vista superior do braço

$$R = \sqrt{P_x^2 + P_y^2}$$

O ângulo α_2 é encontrado com a função “arco tangente 2” que retorna um valor entre $-\pi$ e π .

$$\alpha_2 = \tan^{-1}\left(\frac{P_y}{P_x}\right)$$

$$\beta_2 = \cos^{-1}\left(\frac{251}{R}\right)$$

$$\theta_1 = \alpha_2 + \beta_2$$

Cada ponto encontrado na reta possui seu correspondente em ângulos $(\theta_1, \theta_2, \theta_3)$ radianos, que devem ser convertidos em graus para serem usados como referência para os motores.

No *firmware* esses pontos são adicionados em um vetor de *Structs*, os quais contêm parâmetros de ângulos. O movimento de ponto a ponto é feito com a função *Move*, movendo os motores de grau em grau na ordem junta 1, junta 2 e junta 3, repetindo-se o processo até atingir a referência de posição.

6. RESULTADOS

O primeiro modelo foi feito em cartão duplex, projetado através do recurso de modelamento de chapas (Figura 9).



Figura 9 – Modelo 1

Este modelo, incluindo sua base, não era rígido o suficiente para manter o braço estável.

O segundo modelo foi criado em impressora 3D com material em PLA.

A primeira versão deste modelo (Figura 10, esquerda) ficou com uma estrutura um pouco mais rígida que a anterior, porém insuficiente para garantir que a estrutura ficasse estável durante a movimentação, sendo necessário criar uma nova versão materializada com impressão 3D, com o mesmo material, mas com ressaltos nas bordas dos elos para tornar a estrutura mais rígida e estável (Figura 10, direita).

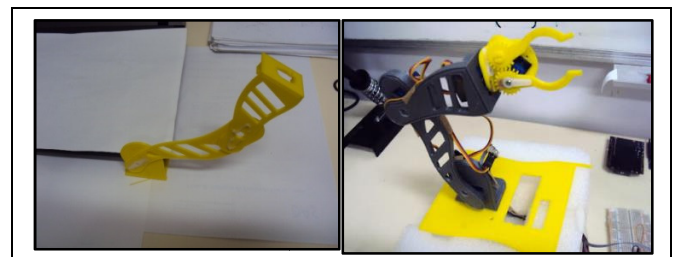
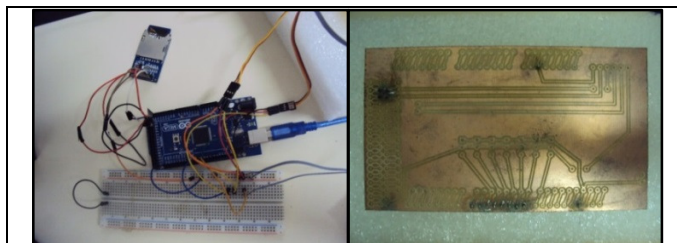


Figura 10 – Modelo 2 versão 1 e 2

O custo das impressões 3D, incluindo o material utilizado e o custo de máquina/hora, foi de aproximadamente R\$ 80,17, de acordo com o software de simulação da própria impressora.

O sistema de hardware (Figura 11, esquerda) possui a placa *Arduino*, leitor de cartão SD e conexões com os servomotores.

Figura 11 - placa Arduino, leitor de cartão SD e conexões com os servomotores.



Para reduzir o número de conexões, tornando o sistema mais compacto, foi criada uma placa de circuito impresso a qual se conecta ao *Arduino* e aos demais componentes sem necessidade de *protoboard*.

A figura 12 mostra o protótipo final para realização de testes, o mesmo que foi apresentado no *Homo Faber Exhibition* do CAAD Futures 2015 em São Paulo de 6-10 de Julho.



Figura 12 Protótipo de alta fidelidade

Com base em lojas especializadas em componentes eletrônicos [23], a placa *Arduino Mega* de produção nacional custa em média R\$ 89,00, o leitor de cartão SD, R\$ 14,90 e os 5 motores utilizados, R\$ 74,50. A placa de circuito impresso foi feita no Laboratório de Montagem Mecatrônica do Departamento de Automação e Sistemas da UFSC sem custo algum, totalizando R\$ 178,40 somente para os componentes de hardware. Somando com a estrutura, o custo final fica em aprox. R\$ 258,17, bem abaixo dos produtos similares.

7. CONCLUSÕES

O projeto desenvolvido buscou contribuir em um campo emergente da grande área da robótica. Trata-se de uma robótica educacional, fundamentada nas teorias de educação provenientes da psicologia, como o construtivismo. Na robótica, buscou-se atuar no campo dos robôs manipuladores, com base nos modelos industriais do tipo antropomórfico, resultando no desenvolvimento de um kit didático de um braço robótico de baixo custo. A meta era desenvolver um produto similar aos modelos industriais, porém simplificado, abaixo de R\$ 500,00.

No aspecto educacional as funções criadas auxiliam no aprendizado da programação, funcionamento de robôs industriais, além de permitir que o braço seja um instrumento secundário para atividades educacionais com outros focos além da robótica propriamente dita, como a elaboração de atividades que desenvolvam a criatividade e curiosidade do aluno. O projeto também mostrou que um trabalho multidisciplinar das áreas de design de produto e engenharia de automação para desenvolver e produzir um protótipo funcional e com estética adequada só foi possível pela troca de saberes e competências.

REFERÊNCIAS

- [1] Malone, T. W; Laubacher, R; Johns, T. The big idea: The age of hyperspecialization. Disponível em: <<https://hbr.org/2011/07/the-big-idea-the-age-of-hyperspecialization/ar/1>>. Acesso em 25 de fevereiro de 2015.
- [2] IFR. International Federation of Robotics. Disponível em: <ifr.org>. Acesso em 25 de fevereiro de 2015.
- [3] Lopes, A. M., 2002, Modelação Cinemática e Dinâmica de Manipuladores de Estrutura em Série. Dissertação de Mestrado. Porto: FEUP – DEMEGI, p. 101.
- [4] Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G., 2009, Robotics: Modelling, Planning and Control. 1st edition. Springer Science & Business Media, pp. 62 – 69.
- [5] Bonsiepe, Guy, 1984. Metodologia Experimental: Desenho industrial. CNPq/Coordenação Editorial, Brasília.



[6] Lego. Disponível em: <<http://www.legobrasil.com.br/mindstorms>>. Acesso em: 25 de agosto de 2014.

[7]Lynxmotion. Disponível em: <<http://www.robotshop.com>>. Acesso em 25 de agosto de 2014.

[8]Modelix. Disponível em: <<http://www.wskits.com.br>>. Acesso em 25 de agosto de 2014.

[9]Miranda, L. C., 2006. RoboFácil: Especificação e Implementação de Artefatos de Hardware e Software de Baixo Custo para um Kit de Robótica Educacional. Dissertação de Mestrado. Rio de Janeiro: IM – NCE – UFRJ, P. 124.

[10]Sasahara, L. R.; Cruz, S. M. S., 2007. Hajime: Uma Nova Abordagem em Robótica Educacional. Artigo. Rio de Janeiro: Anais do XXVII Congresso da SBC.

[11] EasyArm. Disponível em: <<http://www.labdegaragem.org/>>. Acesso em 25 de agosto de 2014.

[12] Oliveira, C. B. M., 1999. Estruturação, Identificação e Classificação de produtos em ambientes integrados de manufatura. Dissertação de Mestrado. São Carlos: EESC, USP.

[13]Leroy Merlin. Disponível em: <http://www.leroymerlin.com.br/?xdtoken=santa_catarina>. Acesso em janeiro de 2015.

[14]Shockmetais. Dados de chapas de alumínio. Disponível em:

<www.shockmetais.com.br/produtos/aluminio/chapa>. Acesso em 28 de janeiro de 2015.

[15]Abal. Associação Brasileira do Alumínio. Disponível em: <<http://www.abal.org.br/sustentabilidade/reciclagem/preco-da-sucata>>. Acesso em 28 de janeiro de 2015.

[16]Waslavick, R. S., 2011. Análise e Projeto de Sistemas de Informação Orientados a Objetos. 2 ed. Rio de Janeiro: Elsevier.

[17] Eshed Robotec, 1998. Scrobot-ER VII Users Manual. 2nd edition. Eshed Robotec.

[18]Arduino. Disponível em: <arduino.cc>. Acesso em 9 de fevereiro de 2015.

[19]Raspberry Pi. Disponível em: <<http://www.raspberrypi.org>>. Acesso em 9 de fevereiro de 2015.

[20]Disponível em: <<http://beagleboard.org/bone>>. Acesso em 9 de fevereiro de 2015.

[21]Negri, V. J., 2001. Sistemas Hidráulicos e Pneumáticos para Automação e Controle: Parte I – Princípios Gerais da Hidráulica e Pneumática. Florianópolis: LASHIP/ EMC/ UFSC.

[22]Brasilrobotics. Disponível em: <brasilrobotics.blogspot.com.br>. Acesso em: 11 de fevereiro de 2015.

[23]FilipeFlop. Disponível em: <<http://www.filipeflop.com/placa-ct-3d60d>>. Acesso em 25 de fevereiro de 2015