

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE MECÂNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

FELIPE DOS SANTOS LAZZARI
HENRIQUE RIBEIRO DE OLIVEIRA
TEILOR SABINO. B. DE Q. PAIXÃO

**DESENVOLVIMENTO DE UM ROBÔ PARALELO TIPO DELTA
ASSOCIADO COM VISÃO COMPUTACIONAL PARA APLICAÇÕES
*PICK AND PLACE***

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2012

FELIPE DOS SANTOS LAZZARI
HENRIQUE RIBEIRO DE OLIVEIRA
TEILOR SABINO. B. DE Q. PAIXÃO

**DESENVOLVIMENTO DE UM ROBÔ PARALELO TIPO DELTA
ASSOCIADO COM VISÃO COMPUTACIONAL PARA APLICAÇÕES
*PICK AND PLACE***

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do curso Superior de Tecnologia em Mecatrônica Industrial dos Departamentos Acadêmicos de Eletrônica – DAELN – e Mecânica – DAMEC – da Universidade Tecnológica Federal do Paraná – UTFPR – como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Rubens Alexandre de Faria.

CURITIBA
2012

FELIPE DOS SANTOS LAZZARI
HENRIQUE RIBEIRO DE OLIVEIRA
TEILOR SABINO. B. DE Q. PAIXÃO

**DESENVOLVIMENTO DE UM ROBÔ PARALELO TIPO DELTA
ASSOCIADO COM VISÃO COMPUTACIONAL PARA APLICAÇÕES
*PICK AND PLACE***

Este trabalho de conclusão de curso foi apresentado no dia 09 de Julho de 2012, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O(s) aluno(s) foi(ram) arguído(s) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Milton Luiz Polli
Coordenador de Curso
Departamento Acadêmico de Mecânica

Prof. Décio Estevão do Nascimento
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Hugo Vieira Neto

Prof. Rubens Alexandre De Faria
Orientador

Prof. Marcelo Victor Wust Zibetti

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Dedico este trabalho às pessoas mais importantes da minha vida: meus pais Ênio e Sueli pelo apoio e suporte durante essa jornada e a minha namorada Bárbara pelo carinho, compreensão, companherismo e principalmente amor, que me levaram a essa conquista.

Felipe dos Santos Lazzari

Dedico este trabalho primeiramente aos meus pais e irmãos que sempre me apoiaram e incentivaram neste desafio, aos amigos pela paciência e compreensão e aos professores pelo auxílio e sabedoria na orientação. Dedico também a minha namorada que nos últimos dias esteve fortalecendo e ajudando nesta vitória.

Henrique Ribeiro de Oliveira

Dedico este trabalho a minha avó, exemplo de vida e quem tornou possível esta conquista. Também a minha esposa pelo apoio, incentivo, carinho e paciência nos momentos de maiores dificuldades. Obrigado, amo vocês!

Teilor Sabino Paixão

AGRADECIMENTOS

Sabendo, desde já, que a memória irá nos trair e certamente não lembraremos de todos que nos ajudaram e apoiaram nesse momento tão importante, registramos nossos pedidos de desculpas e a confissão sincera que, embora seus nomes possam não aparecer nestas linhas, certamente, estão gravados em nossos sentimentos e gratidão.

Agradecemos ao professor Rubens Alexandre de Faria pela orientação recebida no desenvolvimento deste projeto.

Agradecemos ao professor Hugo Vieira pela paciência e orientação quanto a melhorias a serem realizadas no trabalho, suas observações, que com certeza nos ajudaram a executar um melhor projeto.

Agradecemos ao professor Marcelo Victor Wüst Zibetti pela orientação quanto ao sistema de visão de máquina.

Agradecemos ao professor Winderson pela orientação quanto aos cálculos de movimentação do robô e por fornecer materiais que ajudaram no enriquecimento do trabalho.

Agradecemos ao professor Francisco Gódke pelo empréstimo de componentes do sistema pneumático para realização das apresentações do projeto.

Agradecemos ainda aos nossos familiares, amigos e amores: sem vocês nada disso seria possível!

Agradecemos a Deus: quem nos deu a vida e, mais do que isso, deu a Sua própria vida para que pudéssemos viver com Ele eternamente.

Now to him who is able to do immeasurably more than all we ask or imagine, according to his power that is at work within us, to him be glory in the church and in Christ Jesus throughout all generations, forever and ever! Amen. (BIBLE, Ephesians 3.20,21)

Ora, àquele que é poderoso para fazer tudo muito mais abundantemente além daquilo que pedimos ou pensamos, segundo o poder que em nós opera, a esse glória na igreja, por Jesus Cristo, em todas as gerações, para todo o sempre. Amém. (BÍBLIA, Efésios 3.20,21)

RESUMO

LAZZARI, Felipe dos Santos; OLIVEIRA, Henrique Ribeiro de; PAIXÃO, Teilor Sabino B. de Q. **Desenvolvimento de um Robô Paralelo Tipo Delta Associado Com Visão Computacional para Aplicações *Pick and Place***. 2012. 66 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

Este trabalho tem por objetivo o desenvolvimento de um robô paralelo tipo delta que com as suas características se torna mais vantajoso que os robôs de arquitetura serial em aplicações do tipo *pick and place*. Foi desenvolvido um protótipo para apresentação à comunidade acadêmica com a finalidade de incentivar a geração de novos conhecimentos, bem como motivar trabalhos futuros que se proponham a abordar esse tipo de robô e sua arquitetura. O desenvolvimento do robô foi associado à possibilidade do uso da visão computacional para operações de maior precisão em manipulações industriais do tipo “pegar-e-posicionar”, aliado à programação através do *software Labview*.

Palavras-chave: Robô Delta. Arquitetura Paralela. Visão Computacional. *Labview*.

ABSTRACT

LAZZARI, Felipe dos Santos; OLIVEIRA, Henrique Ribeiro de; PAIXÃO, Teilor Sabino B. de Q. **Development of a Parallel Robot Delta Type Associated with Computer Vision for Pick and Place Applications.** 2012. 66 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

This paper presents the development steps of a Delta Parallel Robot whose characteristics make it advantageous on pick and place tasks when compared to serial architecture robots. A prototype was developed to be presented to the academic community in order to gather new findings and improve knowledge, as well as inspiring future endeavors related to this type of device and architecture. To maximize efficiency and precision in “pick-and-place” industrial procedures, this device is associated with the possibility of using computer vision as well as the programming tool LabView.

Keywords: Delta Robot. Parallel Architecture. Computer Vision. Labview

LISTA DE FIGURAS

Figura 1 - O mecanismo de Pollard.....	11
Figura 2 – Desenho CAD do projeto do robô	15
Figura 3 - ISIS/SurgiScope.....	20
Figura 4 - Simulador de condução NADS da Universidade de Iowa.	20
Figura 5 - Okuma 5 eixos e PM-600 em detalhe de seu cabeçote.....	21
Figura 6 - Robôs Delta sendo usados para embalar em indústria alimentícia.	21
Figura 7 - Base do robô.....	22
Figura 8 - Triângulo retângulo formado no plano xz.	23
Figura 9 - Plano yz triângulo retângulo para determinação do θ_1	24
Figura 10 – Componente mecânico para ligação entre a haste superior e inferior. ..	28
Figura 11 – Robô finalizado em CAD.	29
Figura 12 - Foto do robô finalizado.....	30
Figura 13 – Robô realiza a aproximação do objeto em alta velocidade.	32
Figura 14 – Robô se aproxima do objeto com velocidade reduzida e atua o efetuador.	33
Figura 15 – Robô carregado com o objeto se afasta da mesa.	33
Figura 16 – Robô se movimenta em alta velocidade para nova posição.	34
Figura 17 – Robô se aproxima da mesa para posicionar o objeto.	34
Figura 18 – Robô se afasta do objeto com velocidade reduzida e aguarda novo ciclo.	35
Figura 19 – Movimentação do robô.....	36
Figura 20 – Movimentação do robô.....	36
Figura 21 - Esquemático da comunicação serial.....	38
Figura 22 - Diagrama de blocos das conexões físicas (cabos).	46
Figura 23 – Esquemático com Ligações do Sistema de Iluminação.	47
Figura 24 - Esquemático da placa de comunicação RS232.....	48
Figura 25 - Esquemático dos Microcontroladores.	49
Figura 26 – Esquema do sistema pneumático.	50
Figura 27 – Fluxograma Principal.....	51
Figura 28 – Fluxograma do Processo Contínuo	52
Figura 29 – Árvore de Funções do Código em Labview.....	53
Figura 30 – Árvore de Subfunções da Função Main.VI.....	54
Figura 31 - Árvore de sub-funções da função Processo Continuo.VI.....	55
Figura 32 - Árvore de subfunções da função Movimenta Ciclo.VI.....	55
Figura 33 - Árvore de sub-funções da função Envia Dados.VI.....	56
Figura 34 - Fluxograma das funções da câmera.....	60

SUMÁRIO

	DESENVOLVIMENTO DE UM ROBÔ PARALELO TIPO DELTA ASSOCIADO COM VISÃO COMPUTACIONAL PARA APLICAÇÕES <i>PICK AND PLACE</i>	1
1	INTRODUÇÃO	10
1.2	PROBLEMA	12
1.3	JUSTIFICATIVA	13
1.4	OBJETIVOS	14
1.4.1	OBJETIVO GERAL.....	14
1.4.2	OBJETIVOS ESPECÍFICOS.....	14
1.5	MÉTODO DE PESQUISA	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	<i>LABVIEW</i>	17
2.2	CAD (Desenho Assistido por Computador).....	17
2.3	SISTEMA DE VISÃO.....	18
2.4	ROBÔS.....	19
2.5	CINEMÁTICA INVERSA.....	22
3	DESENVOLVIMENTO	25
3.1	MECÂNICA	25
3.2	HARDWARE	30
3.3	SOFTWARE	39
4	TESTES E RESULTADOS	41
4.1	DIFICULDADES ENCONTRADAS.....	42
5	CONCLUSÃO	43
5.1	CONSIDERAÇÕES FINAIS.....	43
5.2	SUGESTÕES PARA TRABALHOS POSTERIORES.....	43
	REFERÊNCIAS.....	45
	APÊNDICE A – Diagrama de Blocos das Conexões Físicas (cabos).....	46
	APÊNDICE B – Esquemático com Ligações do Sistema de Iluminação.....	47
	APÊNDICE C – Esquemático da Placa de Comunicação RS232.....	48
	APÊNDICE D – Esquemático dos Microcontroladores.....	49
	APÊNDICE E – Esquemático do Sistema Pneumático.....	50
	APÊNDICE F - Fluxogramas.....	51
	APÊNDICE G – Configuração dos Microcontroladores.....	62
	APÊNDICE H– Orçamento.....	64

1 INTRODUÇÃO

Os robôs com arquitetura paralela estão atraindo a atenção das indústrias e do meio acadêmico devido às suas características que os tornam mais vantajosos que os robôs com arquitetura serial. Entre essas características, destacam-se a maior leveza, velocidades e acelerações e a capacidade de carga (FINOTTI, 2008).

Os robôs paralelos, em função de suas características, passaram a ser uma ótima alternativa para operações do tipo “pegar-e-posicionar” (*pick-and-place*), necessárias em indústrias de componentes eletrônicos, alimentícias, farmacêuticas e de cosméticos (FINOTTI, 2008).

Neste trabalho é apresentado um robô com arquitetura paralela denominado de robô Delta. Este robô é amplamente utilizado em aplicações que envolvem a manipulação de objetos.

Para a melhor compreensão do trabalho é necessário que alguns termos sejam definidos. Existem diversas definições do que vem a ser um robô. Para este trabalho adotou-se a definição da Associação das Indústrias de Robótica:

Para Groover (1988, p. 23) o robô é “um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais de movimentos variáveis programados para a realização de uma variedade de tarefas”.

Definido o que vem a ser um robô, faz-se necessário especificar o que vem a ser um robô de arquitetura paralela ou simplesmente um robô paralelo. Esse robô, segundo Merlet (2006, p. 5), cuja definição será utilizada neste trabalho, é “um robô constituído de mecanismos de cadeias cinemáticas fechadas no qual o efetuador é ligado a uma base fixa por diversas cadeias cinemáticas independentes.”.

Os primeiros mecanismos paralelos tiveram origem com um sistema de plataforma móvel para cinema patenteado por Gwinnett em 1928 e em um mecanismo para testes de pneus proposto por Gough em 1947 (SICILIANO; KHATIB, 2008). Os mecanismos paralelos foram evoluindo dando origem aos modernos robôs de arquitetura paralela, os quais em função de suas características tendem a substituir a convencional arquitetura serial em uma série de operações. O antecessor do robô paralelo tipo Delta – que é alvo deste trabalho – foi um

mecanismo (figura 1) descrito por Pollard em 1942 e que se destinava à pintura de carros (MERLET, 2006).

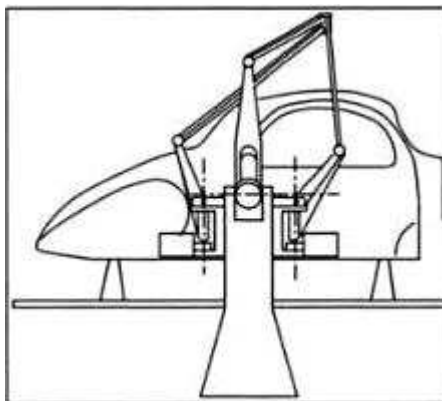


Figura 1 - O mecanismo de Pollard
Fonte: Merlet (2006, p. 27).

O robô Delta, como é conhecido hoje, foi desenvolvido por Raymond Clavel na Escola Politécnica Federal de Lausanne (EPFL), Suíça. Sua invenção tornou-se um dos mais bem sucedidos projetos de robô paralelo do mundo (PARALLEMIC, 2011).

Devido à crescente necessidade de aumentar a produção, as indústrias farmacêuticas e de embalagens em geral adotaram os robôs Delta em suas linhas devido a sua característica de elevada eficiência nas tarefas de “pegar-e-posicionar” (*pick-and-place*). Os robôs Delta podem trabalhar em conjunto nesse tipo de tarefa e chegar a desempenhos de 100 a 2500 manipulações de produtos por minuto (MECHATRONIC-DESIGN, 2010).

É inquestionável a importância que o robô Delta possui hoje no meio industrial em operações do tipo “pegar-e-posicionar”, confirmando-se como uma das mais promissoras soluções para a realização dessas manipulações.

Por outro lado, a visão computacional está intimamente ligada ao sucesso e bom desempenho dos robôs Delta em vários segmentos industriais. Até pouco tempo a visão computacional era, basicamente, utilizada para se realizar a inspeção em sistemas manufaturados e a identificação de peças nesses sistemas (GROOVER et al., 1988, p. 39). No entanto, com o avanço no campo de visão computacional ela não só realiza as funções de identificação e inspeção, como também é utilizada para orientar os movimentos do robô, além de ter sofrido uma melhora significativa em sua capacidade de reconhecimento, localização e inspeção da peça.

Esse sistema de visão permite que o robô movimente o seu objeto-alvo com maior precisão, velocidade e qualidade (colocando o objeto no local apropriado), uma vez que trabalha em conjunto com um banco de dados que possui todas as opções possíveis de manipulação a serem realizadas pelo robô. O sistema identifica um determinado objeto, processa a imagem e a compara com o banco de dados disponível, atuando conforme a informação contida nos arquivos do banco. Tudo isso em frações de segundo (GROOVER et al., 1988).

Tendo-se o conhecimento da criação e avanço dos robôs com arquitetura paralela, especialmente o robô Delta, e a importância que os robôs com essa arquitetura estão adquirindo nas últimas décadas, um trabalho que tenha por objetivo desenvolver este robô, inclusive associado com visão de máquina, mostra-se oportuno e de grande importância para o possível desenvolvimento de novas soluções para problemas industriais, além de possuir significativa importância no meio acadêmico, firmando-se como uma das áreas de pesquisa que mais prometem estudos e desenvolvimentos nos próximos anos (PARALLEMIC, 2011).

1.2 PROBLEMA

Os robôs de arquitetura paralela ou simplesmente robôs paralelos ainda são pouco explorados como soluções para problemas no meio industrial e também são pouco explorados no meio acadêmico.

Embora o uso de robôs paralelos ainda seja pouco explorado quando comparado aos robôs de arquitetura serial, é possível observar um pequeno, mas gradativo crescimento na preferência da arquitetura paralela. Por esta ser constituída de várias cadeias (hastes) atuando ao mesmo tempo sobre a plataforma móvel (órgão terminal) em vez de uma só, como ocorre na arquitetura serial, ela se mostra mais leve, veloz e com uma maior aceleração, além de possuir uma maior capacidade de carga. Estas características tornam a escolha da arquitetura paralela ideal para o desenvolvimento de robôs destinados à manipulação de objetos.

Existem algumas áreas industriais que necessitam de velocidade, precisão e delicadeza na movimentação de seus materiais. O robô Delta é comumente empregado para a realização desse tipo de tarefa. Ele pode movimentar e posicionar componentes eletrônicos com alta velocidade e precisão. Também pode ser usado na indústria alimentícia, embalando os alimentos com rapidez, precisão e higiene,

sendo ainda utilizado nas indústrias farmacêuticas e de cosméticos (BOSCH-LIVE, 2010).

Com o objetivo de garantir a exatidão de seus movimentos (pegar um determinado material e o posicionar no seu exato lugar), este robô geralmente é dotado de visão computacional ou visão de máquina. Este sistema de visão permite que o Delta saiba exatamente onde pegar e colocar cada tipo de componente, material ou alimento.

Conhecendo-se a realidade da exploração dos robôs com arquitetura paralela no meio industrial e acadêmico – que embora pouco explorados, estão em crescimento – este trabalho visa o desenvolvimento de um determinado tipo de robô paralelo, o robô Delta, para assim demonstrar a viabilidade comercial destes robôs bem como a importância do estudo dos mesmos no meio acadêmico.

1.3 JUSTIFICATIVA

O uso de robôs no meio industrial cresceu muito nos últimos anos e ainda está em crescimento. Eles podem realizar uma série de tarefas com maior velocidade, precisão e qualidade que o homem, além de aumentarem de uma maneira significativa a produção.

Além de ser interessante como solução para problemas industriais, o robô Delta também tem atraído o interesse do meio acadêmico. Existindo ainda várias questões a serem abordadas e discutidas e poucos trabalhos acadêmicos que se proponham a lidar com esse tipo de robô ou ainda com esse tipo de arquitetura.

Acredita-se que a realização deste trabalho:

- Possa demonstrar o funcionamento de um robô paralelo tipo Delta, bem como a sua viabilidade comercial e acadêmica;
- Possa incentivar a apresentação de soluções alternativas para problemas industriais com base na arquitetura paralela;
- Possa ser uma solução para as indústrias que necessitam realizar movimentação de materiais diversos com velocidade, precisão e uma capacidade de carga variável;

- Possibilite o desenvolvimento do robô Delta com o objetivo do desenvolvimento acadêmico e científico.

1.4 OBJETIVOS

1.4.1 OBJETIVO GERAL

Desenvolvimento de um robô paralelo tipo delta associado com visão computacional capaz de identificar, manipular e separar diferentes tipos de elementos em ambiente industrial.

1.4.2 OBJETIVOS ESPECÍFICOS

- Gerar os desenhos em CAD dos componentes do robô e construí-lo.
- Programar um *software* de visão computacional (*Vision Assistant*).
- Implementar um sistema de controle para o robô.
- Implementar um sistema para a comunicação entre os *drivers* e o sistema de controle.
- Implementar o sistema de manipulação do robô.
- Demonstrar o funcionamento de um robô com arquitetura paralela.
- Elaborar a documentação técnica.

1.5 MÉTODO DE PESQUISA

Elaborou-se o projeto CAD do robô no *software SolidWorks* para que fosse possível levantar informações sobre as características do mesmo, tais como o tamanho, a movimentação e dimensões em geral.

Com uma ideia das características do robô em mãos fez-se a escolha do material a ser usado nas hastes superiores do robô (que são mais curtas), das hastes inferiores (que são mais longas), bem como de toda a estrutura do mesmo (figura 2).

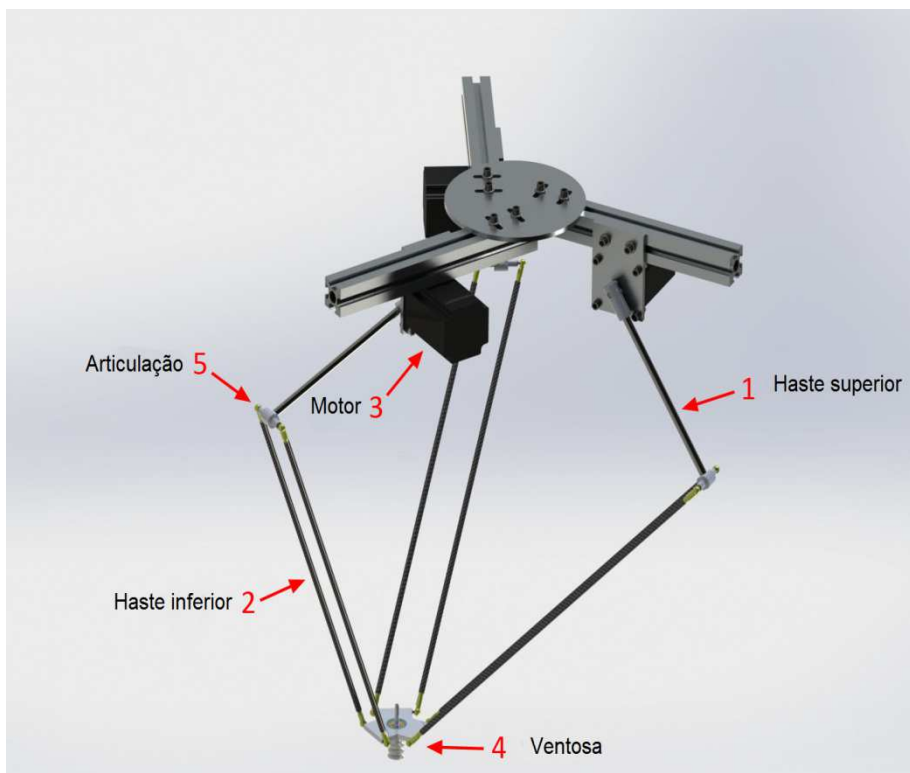


Figura 2 – Desenho CAD do projeto do robô
Fonte: Autoria própria.

Utilizou-se motores-de-passo, uma vez que os mesmos não precisam de realimentação (o que facilitou a programação) e, principalmente, por causa de sua resolução, que é algo de grande importância neste projeto. A fim de se alcançar a precisão necessária para as movimentações do robô, estima-se o uso de *driver* para controle do motor com *microstep* de até 1/40. Este *driver* poderia ser confeccionado pela equipe ou adquirido, uma vez que os custos não variam muito entre as duas opções.

Fez-se o uso de juntas esféricas a fim de aumentar o número de movimentações possíveis. As mesmas podem ser adquiridas via importação com preço estimado em U\$ 10. A comunicação entre o computador e o *driver* de controle foi realizada com um microcontrolador PIC. A comunicação entre o PIC e o computador foi realizada pela entrada serial (apêndices D e G).

O *software labview* foi utilizado para a estruturação do sistema supervisório e também para a parte de cálculos de cinemática inversa do robô, uma vez que o mesmo já dispõe de módulos e bibliotecas que realizam essas funções. Para o sistema de visão computacional foi utilizada uma *webcam*, por causa de seu baixo custo, e também porque o *labview* possui um módulo chamado *Vision Assistant* que

é projetado para a captação de imagens via câmera, processamento e interpretação dos dados obtidos.

Dentro do *labview* foi feito o uso da linguagem C para a implementação de algumas funções e a programação do PIC também foi realizada nessa linguagem, por sua familiaridade pelos integrantes do grupo. Foi utilizado um efetuador para o robô Delta do tipo sucção.

Com o andamento dessas etapas o grupo foi capaz de desenvolver o robô fisicamente, adquirir novos conhecimentos durante esse processo e demonstrar a superioridade desse tipo de robô em manipulações de objetos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 LABVIEW

Segundo o fabricante National Instruments (2011):

“LabVIEW é um software de projeto gráfico de sistemas que oferece a engenheiros e cientistas as ferramentas necessárias para criar e implementar sistemas de medição e controle por meio de uma integração de hardware sem precedentes. O LabVIEW inspira você a resolver problemas, acelera sua produtividade e traz mais segurança aos seus esforços de inovação contínua. (NATIONAL INSTRUMENTS, 2012)”

Os programas em *Labview* são aplicados para:

- Aquisição de dados;
- Controle de instrumentos;
- Automatização de testes;
- Processamento de sinal;
- Controle Industrial;
- Projeto embarcado.

2.2 CAD (Desenho Assistido por Computador)

CAD, ou desenho assistido por computador, é uma tecnologia que consiste em utilizar-se de computadores para realizar certas funções no que se remete a desenhos e fabricação. Essa tecnologia, atualmente, está se aliando à tecnologia CAM (Manufatura Auxiliada por Computador) e dessa forma estão se tornando cada vez mais usadas nas indústrias de fabricação (GROOVER; ZIMMERS, 1984).

É importante salientar que a tecnologia CAD só tornou-se possível devido à evolução da capacidade de armazenamento e velocidade de processamento dos computadores, e também de *softwares* capazes de realizar as funções de desenho requeridas pelo usuário, facilitando todo o trabalho de engenharia envolvido no processo (GROOVER; ZIMMERS, 1984).

O CAD pode ser usado para auxiliar na criação, modificação, análise, ou otimização de desenhos de fabricação. Também pode ser usado para determinar análises de tensão de escoamento de materiais, respostas dinâmicas de

mecanismos e cálculos de transferência de calor, por exemplo. (GROOVER; ZIMMERS, 1984).

2.3 SISTEMA DE VISÃO

Conforme foi visto, a visão computacional é de significativa importância para o sucesso e bom desempenho do robô Delta em operações de “pegar-e-posicionar” (*pick-and-place*) por isso, é importante, conhecer os fundamentos dessa área do conhecimento.

A visão computacional ou visão de máquina pode ser definida, basicamente, como a aquisição e digitalização de dados por um dispositivo de visão (câmeras), no processamento desses dados e em sua interpretação por um computador (GROOVER et al., 1988).

A aquisição das imagens é feita através de uma câmera de vídeo e o sistema de digitalização armazena esses dados para serem analisados posteriormente (GROOVER et al., 1988).

O processamento e a análise das imagens obtidas são o segundo passo de um sistema de visão. Existem várias tecnologias para se realizar a análise das imagens obtidas em um sistema de visão, sendo uma das principais a técnica da segmentação (GROOVER et al., 1988).

A última etapa, a interpretação dos dados obtidos, consiste em se comparar os dados obtidos pelo sistema de visão com modelos ou valores pré-existentes em um banco de dados. O objeto em questão deve apresentar valores que se encontrem dentro das tolerâncias especificadas nos arquivos do computador para que o sistema seja capaz de interpretar a operação que deve realizar sobre o objeto em análise, peça conforme, não conforme, descarte, etc. (GROOVER et al., 1988)

Além disso, existem outras questões a serem consideradas em sistemas de visão como, por exemplo, o tipo de câmera a ser utilizada, a iluminação, a resolução que se pretende obter da imagem e o treinamento do sistema de visão.

2.4 ROBÔS

A robótica teve o seu início na ficção científica. Atribui-se a peça *Os Robôs Universais de Rossum*, de autoria de Karel Capek, a origem do termo robô. A palavra tcheca *robot* quer dizer servidão ou trabalhador forçado e quando foi traduzida para o inglês transformou-se em *robot*. A peça conta a história de um cientista chamado Rossum, e seu filho, que criam uma determinada substância e a usam para a fabricação de robôs. Estes deveriam servir obedientemente à humanidade e realizar todo o trabalho físico. No entanto, os robôs acabam por rebelar-se, não gostando de seu lugar de servidão, e destroem toda a vida humana (GROOVER et al., 1988).

Isaac Asimov merece destaque entre os escritores de ficção científica, tendo escrito inúmeras histórias sobre robôs e lhe é atribuído o crédito de criar o termo “robótica”. O robô criado por Asimov é uma máquina bem projetada, que não falha e que atua de acordo com as três leis da robótica criadas por ele (GROOVER et al., 1988):

1. *Um robô não pode ferir um ser humano ou, por inação, permitir que um humano seja ferido.*
2. *Um robô deve obedecer às ordens dadas por humanos, exceto quando isto conflitar com a Primeira Lei.*
3. *Um robô deve proteger sua própria existência, a menos que isso conflite com a Primeira ou a Segunda Lei.*

Os robôs e os conceitos de robótica foram e ainda são explorados em muitas obras de ficção científica. Ao contrário da ficção, os primeiros robôs desenvolvidos pouco lembravam a figura de um humano e sua capacidade de atuação era muito limitada. Ao longo do tempo, no entanto, com o desenvolvimento de novas tecnologias e o aperfeiçoamento de áreas como engenharia mecânica, elétrica, eletrônica, robótica e outras, os robôs passaram a ser mais eficazes, sendo hoje indispensáveis em vários ramos da indústria e do conhecimento (GROOVER et al., 1988).

Robôs com cadeias cinemáticas fechadas são conhecidos como robôs com arquitetura paralela (MERLET, 2006). Os robôs paralelos são, hoje, utilizados em vários ramos industriais e do conhecimento. Na área médica, entre outras aplicações, destaca-se o ISIS/SurgiScope (figura 3), onde um robô delta sustenta um pesado microscópio (MERLET, 2006).



Figura 3 - ISIS/SurgiScope.
Fonte: Merlet (2006, p. 28)

Um dos setores onde as estruturas paralelas tiveram maior sucesso foi na área de simuladores (figura 4).



Figura 4 - Simulador de condução NADS da Universidade de Iowa.
Fonte: Merlet (2006, p. 31).

As estruturas paralelas também passaram a ser usadas em máquinas-ferramenta (figuras 5, a esquerda e a direita, respectivamente):

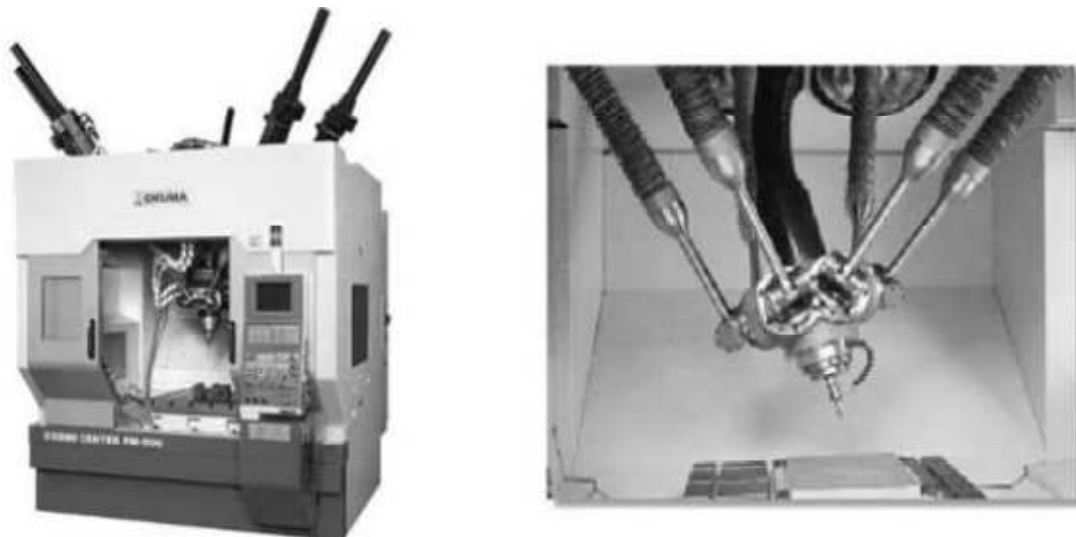


Figura 5 - Okuma 5 eixos e PM-600 em detalhe de seu cabeçote
Fonte: Merlet (2006, p. 34)

Por sua leveza, velocidade e precisão nos movimentos o robô Delta é amplamente utilizado na indústria de embalagens de alimentos (figuras 6).

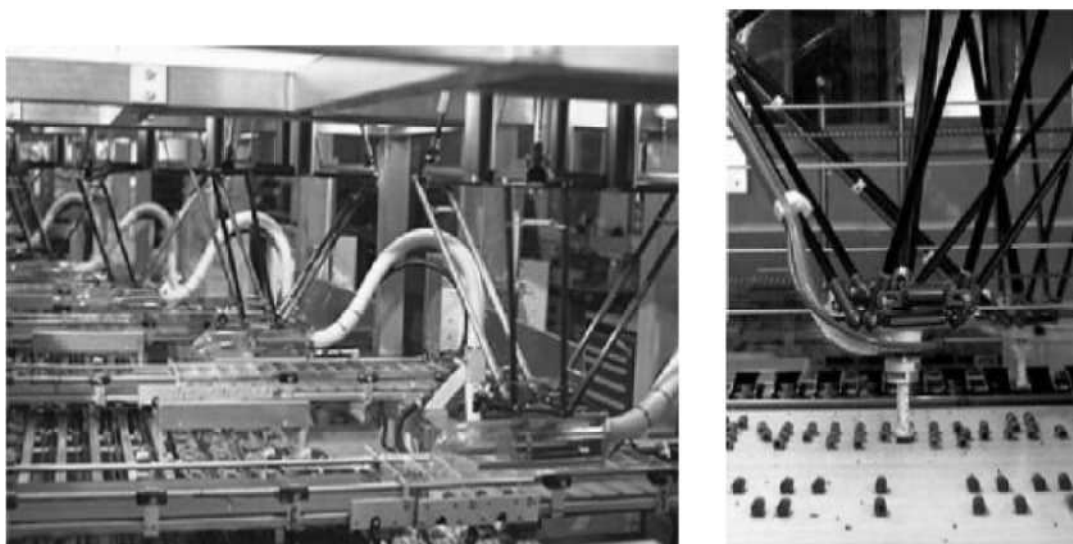


Figura 6 - Robôs Delta sendo usados para embalar em indústria alimentícia.
Fonte: Merlet (2006, p. 37)

Pode ser visto que o uso dos robôs sofreu um grande crescimento desde sua concepção até os dias de hoje. Também é possível observar que os robôs com arquitetura paralela passaram a ganhar espaço nos mais variados ramos da indústria, mostrando-se uma solução viável para diversas aplicações.

2.5 CINEMÁTICA INVERSA

Para mover o robô para uma determinada posição, aqui chamada de posição $C(x_0, y_0, z_0)$, é necessário calcular os ângulos dos três motores de passo em relação à origem do sistema cartesiano tridimensional. Esse tipo de cálculo é chamado de cinemática inversa, que nos permite calcular a posição dos efetadores através dos valores da posição final desejada.

A posição final $C(x_0, y_0, z_0)$, se dá em relação ao efetador, no caso a ventosa, que está disposta no centro do triângulo formado pela base menor do robô, como mostra a figura 7. Como a junta esférica está fixada na lateral desse triângulo (ponto C_1) é necessário levar em conta a distância do centro até a lateral h .

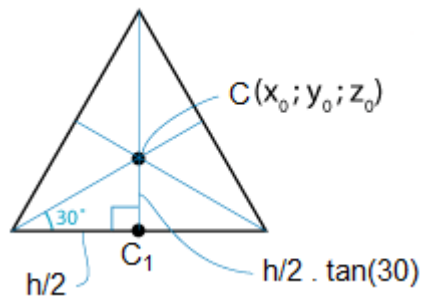


Figura 7 - Base do robô.
Fonte: Autoria própria.

O ponto CC_1 é definido por:

$$CC_1 = \frac{h}{2} \tan 30 = \frac{h}{2\sqrt{3}} \quad (1)$$

Considerando que o segmento de reta CC_1 está no plano alinhado no eixo y , podemos escrever o ponto C_1 como:

$$C_1(x_0, y_0 - \frac{h}{2\sqrt{3}}, z_0) \quad (2)$$

Sendo o ponto C'_1 a projeção do ponto C_1 no plano yz então:

$$C'_1(0, y_0 - \frac{h}{2\sqrt{3}}, z_0) \quad (3)$$

Se o segmento de reta C'_1B_1 a projeção da haste inferior no plano yz, podemos formar um triângulo retângulo no plano xz conforme a figura 8:

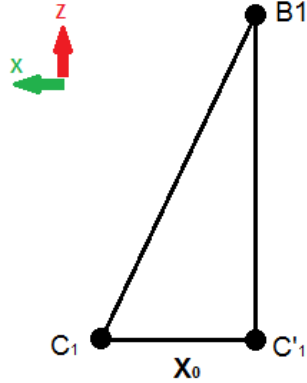


Figura 8 - Triângulo retângulo formado no plano xz.
Fonte: Autoria própria.

Se o segmento de reta C_1B_1 o comprimento da haste inferior (haste b) do robô e C'_1B_1 sendo x_0 , temos que o comprimento da projeção de C_1B_1 no plano yz é:

$$C'_1B_1 = \sqrt{(C_1B_1)^2 - (C_1C'_1)^2} \rightarrow \sqrt{b^2 - x_0^2} \quad (4)$$

Para a base superior fazemos o mesmo cálculo feito para a base inferior já que os motores estão fixados na lateral desse triângulo (ponto A_1) é necessário levar em conta a distância do centro até a lateral do triângulo da base superior r . Por isso o ponto F_1 é

$$F_1\left(0, -\frac{r}{2\sqrt{3}}, 0\right) \quad (5)$$

No plano yz podemos formar dois triângulos retângulos, conforme a figura 9:

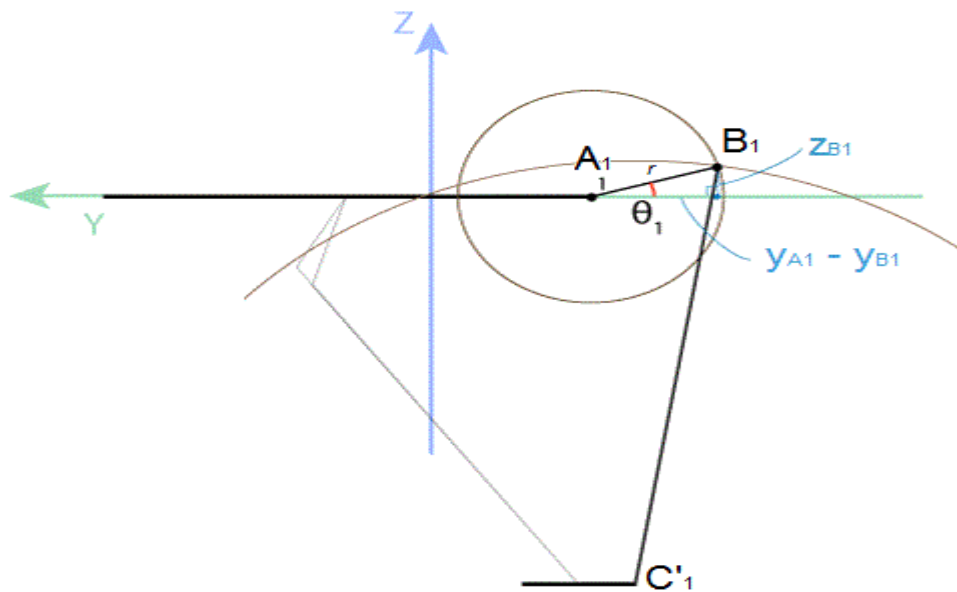


Figura 9 - Plano yz triângulo retângulo para determinação do θ_1
 Fonte: Autoria própria.

Com os dois triângulos retângulos formados no plano yz, é possível obter-se duas equações para a determinação de duas variáveis faltantes z_{B1} e y_{B1} da seguinte forma:

$$\begin{aligned} (y_{B1} - y_{A1})^2 + (z_{B1} - z_{A1})^2 &= r^2 \\ (y_{B1} - y_{C'1})^2 + (z_{B1} - z_{C'1})^2 &= b^2 - x_0^2 \quad (6) \end{aligned}$$

Com z_{B1} e y_{B1} podemos determinar o ângulo θ_1 , que é o ângulo do motor de passo, da seguinte forma:

$$\theta_1 = \arctan\left(\frac{z_{b1}}{y_{A1} - y_{B1}}\right) \quad (7)$$

3 DESENVOLVIMENTO

3.1 MECÂNICA

O desenvolvimento mecânico do robô teve início com o projeto CAD, onde foram observadas características como tamanho, movimentação e dimensões em geral. Com o resultado destas primeiras simulações em mãos, partiu-se para a construção de um primeiro protótipo do robô. Foram necessárias, aproximadamente, duas semanas para a conclusão dessa etapa.

Para o desenvolvimento deste primeiro protótipo foi utilizada uma estrutura em madeira – espetinhos de churrasco – para confecção das hastes do robô. Para simular as juntas esféricas foram utilizados fios de cobre torcidos entre si. Neste primeiro teste prático foram observadas algumas características relacionadas ao movimento do robô, por exemplo, sua área de trabalho, ângulos das juntas, tamanhos e distâncias das hastes do robô. Pôde-se observar durante estes ensaios que a área de trabalho do Delta sofre grande influência da haste superior e, além disso, é limitada em razão do ângulo máximo da junta esférica. Também foi possível visualizar que o robô apresentaria problemas com vibração (em função de sua estrutura mecânica) e que na movimentação dos motores a trajetória realizada poderia facilmente ocasionar o choque do robô com a plataforma onde o mesmo estaria realizando a movimentação dos objetos.

Com o resultado desta primeira simulação passou-se a desenvolver no *software SolidWorks* uma versão CAD mais desenvolvida e atualizada do que seria o robô. Com o projeto CAD finalizado foram realizados novos ensaios e testes a fim de se identificar quais seriam as mudanças necessárias para obterem-se melhorias na área de trabalho e ângulos das juntas do robô. As soluções tomadas para se tentar corrigir as inconsistências apresentadas nos primeiros testes foram a alteração da distância entre os motores, posicionando-os mais próximos do eixo central do Delta, o aumento no comprimento de todas as hastes e a decisão de posteriormente alterar-se as juntas esféricas com o intuito de possibilitar um maior ângulo nas mesmas.

Com a conclusão desta etapa as pesquisas foram intensificadas com a intenção de se identificar quais seriam os materiais que melhor se adequariam ao

desenvolvimento do projeto, ou seja, que possibilitariam melhorias em relação à área de trabalho, ângulos das juntas, etc.

Projetou-se o uso de perfis estruturados de alumínio para serem utilizados nas hastes do robô, uma vez que esse tipo de material poderia ser facilmente adquirido sob a forma de antenas para televisão. No entanto, o diâmetro desse material na forma de antenas era superdimensionado para as necessidades e expectativas do projeto, além disso, foi verificado que os mesmos seriam frágeis a colisões (a fragilidade relaciona-se ao baixo momento inercial do perfil de alumínio), esteticamente o robô ficaria com um aspecto indesejável e para se conseguir o mesmo material com um diâmetro menor seria necessário a realização de um pedido de grande porte. Por estes motivos o uso deste tipo de material foi descartado pela equipe.

Paralelamente deu-se continuidade às pesquisas a fim de se identificar possíveis materiais a serem utilizados. Alguns materiais foram localizados, mas, os mesmos apresentavam elevados preços no mercado nacional. Assim sendo, privilegiou-se a aquisição destes materiais no mercado internacional, mesmo conhecendo-se que os mesmos levariam um maior tempo para estarem à disposição da equipe.

Foram adquiridos alguns materiais (juntas com rolamentos) que possibilitariam o desenvolvimento das juntas do robô de forma a permitir que o mesmo tivesse 3 graus de liberdade. Ao mesmo tempo foi realizada a encomenda de dois tipos de juntas com características semelhantes entre si para se estudar qual delas desempenharia com maior eficiência a função proposta.

Com o desenvolvimento das juntas com rolamento pode-se observar que as mesmas apresentavam características superdimensionadas para o projeto. Uma vez que os materiais importados não possuíam medidas ou cotas explícitas era necessário realizar testes práticos com os mesmos a fim de se levantar características sobre seu funcionamento e desempenho. Os testes realizados mostravam a limitação que as juntas impunham ao ângulo de trabalho, usando-se para isso um transferidor acoplado em cada um dos motores para a realização das medições.

Com o recebimento dos novos materiais encomendados deu-se início ao desenvolvimento de um novo protótipo que simplificaria a mecânica necessária. Neste momento partiu-se para o uso de raios de bicicleta para serem utilizados

como hastes do robô, este material apresentava vantagens como fixação mais simples, maior resistência a colisões e facilidade de aquisição (em torno de R\$, 0,50 o raio). Para a ligação das juntas foram desenvolvidas peças de acrílico que foram anteriormente modeladas em madeira para se realizar o molde em borracha de silicone, sendo que os mesmos seriam usados para replicar as peças posteriormente.

Com a realização dos testes no protótipo algumas falhas puderam ser observadas no projeto, tais como, o ângulo de trabalho (ângulo das juntas) era muito pequeno, o que acabava por diminuir de maneira significativa a área de atuação (ou área de trabalho) do robô e as pernas do robô ainda não apresentavam o desempenho esperado pelo grupo.

Seguindo a pesquisa e consultando outros projetos de robô paralelo tipo delta que obtiveram sucesso, foi encontrado um modelo de junta esférica que permitiria uma área de trabalho muito maior que as anteriores, podendo ainda sofrer ajustes e alterações que melhorassem este ângulo. Também através da pesquisa chegou-se à conclusão de que peças em fibra de carbono seriam ideais para atenderem às expectativas do projeto. Deste modo localizou-se um fornecedor para as mesmas.

Para as hastes do robô, que deveriam ser mais rígidas e oferecerem maior robustez, decidiu-se por usar perfis tubulares de inox. Enquanto a equipe aguardava o recebimento dos materiais e utilizando estimativas de peso, carga de trabalho e velocidade do robô realizou-se o dimensionamento dos motores que atendessem aos requisitos de projeto.

Tendo-se recebido os materiais encomendados, partiu-se para o desenvolvimento do restante da mecânica, sendo que, para a conexão do motor com o braço, foram usinadas buchas de alumínio as quais foram travadas diretamente no tubo de inox, usando-se para isso trabisil – utilizado para vedar e travar peças cilíndricas.

Essas buchas foram usinadas de forma a terem um furo passante para o eixo do motor o qual seria fixado usando-se parafuso do tipo *allen* sem cabeça. Na conexão entre a haste superior e inferior onde são utilizadas duas juntas esféricas foi desenvolvido um componente mecânico que pode ser visto a seguir (figura 10):



Figura 10 – Componente mecânico para ligação entre a haste superior e inferior.
Fonte: Autoria própria.

Este componente tem as juntas fixadas em suas extremidades e o braço engastado na parte central. A base foi modelada de forma a ter seus fixadores formados por duas juntas esféricas, que estão posicionadas a cada 120° para coincidir com a orientação dos motores. Com a chegada da fibra de carbono foram projetadas as buchas de fixação entre a fibra e a junta esférica, sendo que estas foram coladas, uma vez que a fibra de carbono não apresenta acabamento interno o que a tornava muito delicada para outro tipo de fixação, tal como, através de parafusos.

O efetuator do robô, que foi produzido próximo à conclusão do mesmo, é composto por uma ventosa posicionada no centro da base inferior, emprestada do departamento de pneumática da UTFPR, sendo que a mesma possui um rolamento no seu eixo de fixação para possibilitar o giro das peças a fim de propiciar melhorias futuras no projeto (apêndice E). Foi utilizado um tubo de silicone para a conexão da ventosa, sendo que este não influi de maneira significativa na movimentação do robô.

Para a realização dos primeiros testes e ensaios do robô, foi desenvolvido e construído um cavalete de madeira onde seria realizada a fixação da sua parte superior a fim de, assim, observar-se o comportamento dos motores quanto a velocidade e aceleração e também para dar início a programação do projeto. Com o uso do cavalete constatou-se a necessidade de se ter uma estrutura rígida, uma vez que a vibração e a inércia criada pela movimentação do robô eram transferidas para a estrutura de maneira significativa na mesma, o que acabava por gerar problemas relacionados à precisão dos movimentos.

Para se realizar a fixação dos motores, foram desenvolvidas peças com maior rigidez para serem fixadas diretamente em perfis estruturados de alumínio, os

quais possibilitariam maior liberdade para ajustes no posicionamento dos motores. Para se conseguir criar o ângulo de 120 graus correspondente à base inferior foi desenvolvida uma peça em formato circular com rasgos (oblongos) que permitem ajustes no ângulo dos perfis estruturais que sustentam os motores. Os perfis de alumínio também foram utilizados para se fazer a fixação dos sensores que viriam a servir como finais de curso posteriormente. Na base superior também foi feita a fixação da câmera responsável pela aquisição das imagens, as quais seriam posteriormente processadas por *software*. Posteriormente, após vários testes e ensaios, iniciou-se o desenvolvimento da parte de sustentação definitiva do robô. Essa estrutura foi construída em perfis estruturados de alumínio que garantem boa rigidez ao projeto e leveza que facilita transporte do mesmo. Outra vantagem em se utilizar esse tipo de estrutura foi a facilidade em se realizar a fixação de toda a parte de *hardware* de forma a criar uma estrutura limpa, organizada e esteticamente agradável. Nas figuras abaixo é possível visualizar o projeto do robô finalizado em CAD (figura 11) e uma foto do mesmo desenvolvido fisicamente (figura 12).

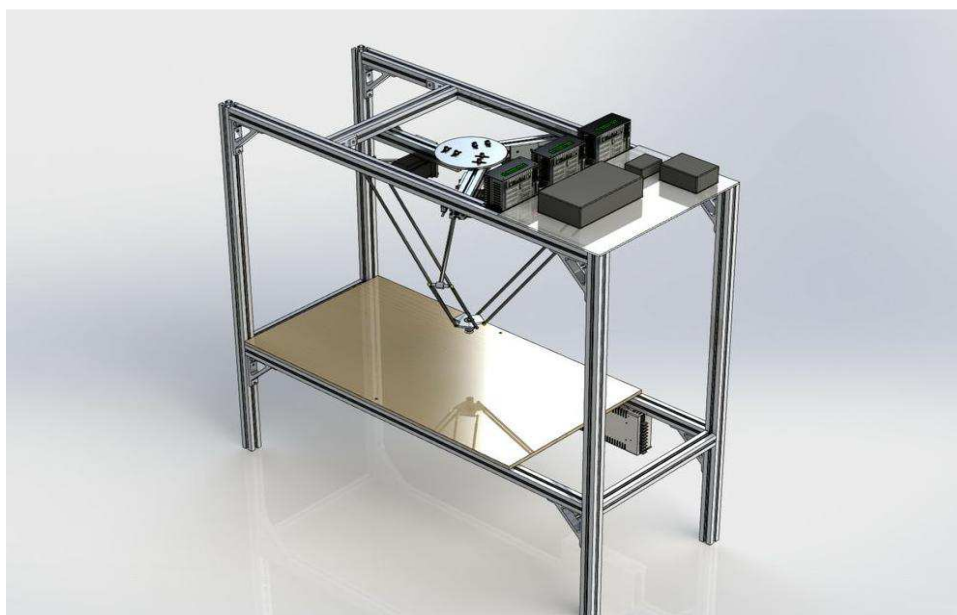


Figura 11 – Robô finalizado em CAD.

Fonte: Autoria própria.

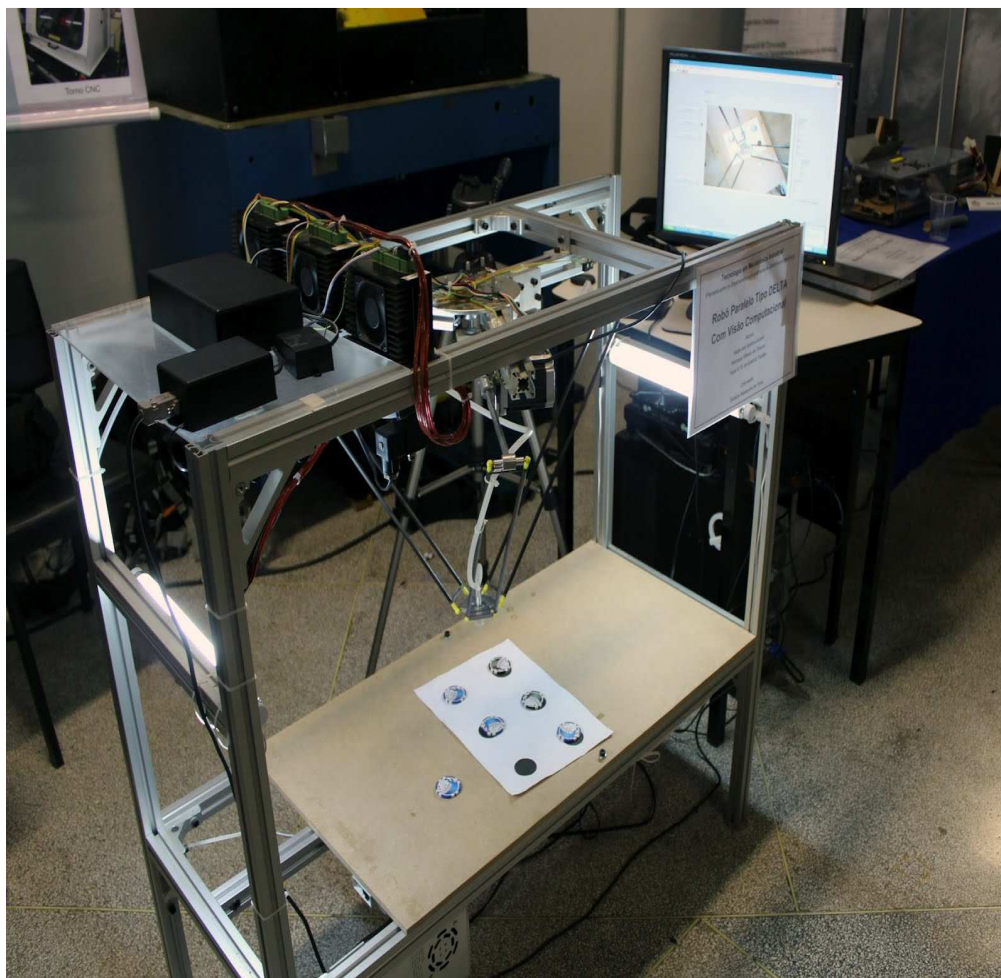


Figura 12 - Foto do robô finalizado.
Fonte: Autoria própria.

3.2 HARDWARE

Para o desenvolvimento do projeto do robô constatou-se que seria necessário fazer uso da eletrônica de potência para realizar o acionamento dos motores. Deste modo, após realizar-se a aquisição dos motores de passo iniciou-se o desenvolvimento do projeto do driver do motor. Em um primeiro momento foram realizadas pesquisas que conferissem ao grupo a capacidade de desenvolver um projeto próprio dos drivers de potência necessários para o acionamento dos motores em micro passos e assegurariam a alimentação de três ampères aos mesmos.

Pretendia-se para o desenvolvimento deste driver o uso de um microcontrolador PIC, que associado ao uso de alguns MOSFETS (Transistor de Efeito de Campo de Semicondutor de Óxido Metálico) de potência realizaria o chaveamento dos pólos do motor.

Foram tentadas algumas alternativas pelo grupo no que se refere à escolha dos *drivers*, sendo que, ao final de alguns ensaios e testes, optou-se pelo uso de *drivers* industriais que proporcionariam um maior número de funções em relação a outros *drivers* testados e também apresentariam maior robustez.

Após a escolha do *driver* foram iniciados testes com o objetivo de se realizar a escolha do microcontrolador a ser utilizado no projeto. Em um primeiro momento cogitou-se o uso do PIC 16F877. Entretanto, após alguns testes relacionados a movimentação do robô foi constatado que o mesmo não seria capaz de realizar parte dos cálculos de aceleração que foram implementados ao longo do projeto devido à limitação de seu *clock* de processamento.

Embora existissem algumas soluções que amenizariam o problema e permitiriam ao grupo continuar utilizando o mesmo PIC, optou-se pela troca do mesmo por um de maior *clock* de processamento com o intuito de aumentar o desempenho dos cálculos realizados. Estimou-se, em um primeiro momento, que para a movimentação do robô delta seriam criadas trajetórias paralelas (retilíneas) ao plano, no entanto, percebeu-se que para esse tipo de movimentação o processamento seria muito pesado, pois, o PIC não teria capacidade de gerar todos os cálculos necessários. Assim, imaginou-se o uso do computador para realizar a geração das posições e posterior transmissão das mesmas para o microcontrolador.

Apesar de projetar-se que a solução apresentada acima fosse a mais simples, percebeu-se que a mesma criaria dificuldades para trabalhar em movimentação em tempo real em conjunto com a aceleração e desaceleração dos motores. Nesta ideia o programa *Labview* seria responsável por gerar a interpolação da trajetória em conjunto com informações referentes à aceleração e desaceleração dos motores e transmissão dessas informações para o PIC, que não possuía capacidade para realização de tais cálculos uma vez que o tempo de transmissão de cada posição interpolada do *labview* para o PIC seria maior que o tempo necessário para gerar as curvas de aceleração, tornando o movimento irregular. O software de interpolação já havia sido implementado, entretanto, dificuldades na programação do microcontrolador para que o mesmo trabalhasse em paralelo com os outros PICs e gerasse as curvas de aceleração e desaceleração fizeram com que o grupo optasse por deixar essa ideia de lado e adotar uma nova estratégia.

Tendo-se em mente a simplificação da programação foi decidido que a movimentação retilínea do robô seria retirada, mantendo-se somente cálculos

referentes à aceleração e desaceleração no PIC e geração de rotinas de intertravamento para que, assim, todos os motores viessem a trabalhar em conjunto. Enquanto isso no *Labview* seriam realizados cálculos da posição final, inicial e de afastamento e estes resultados transmitidos para os microcontroladores de forma a criarem três trajetórias para cada movimento de pegar-e-posicionar.

Para melhor compreensão das três trajetórias envolvidas em cada movimentação executada pelo robô segue abaixo ilustração destes movimentos.

Robô sai de uma posição inicial qualquer para se aproximar do objeto com uma alta velocidade (figura 13)



**Figura 13 – Robô realiza a aproximação do objeto em alta velocidade.
Fonte: Autoria própria.**

Com velocidade reduzida realiza a descida até o objeto e ao final do movimento atua o efetuator (ventosa) sobre o mesmo (figura 14):



Figura 14 – Robô se aproxima do objeto com velocidade reduzida e atua o efetuator.

Fonte: Autoria própria.

Ainda com velocidade reduzida, realiza o movimento de afastamento da mesa (figura 15):



Figura 15 – Robô carregado com o objeto se afasta da mesa.

Fonte: Autoria própria.

Movimenta-se com alta velocidade para realizar a aproximação da nova posição onde será depositado o objeto (figura 16):



**Figura 16 – Robô se movimenta em alta velocidade para nova posição.
Fonte: Autoria própria.**

Com velocidade reduzida, realiza a descida até onde será depositado o objeto e no fim do movimento desliga o efetuador (figura 17):



**Figura 17 – Robô se aproxima da mesa para posicionar o objeto.
Fonte: Autoria própria.**

Com velocidade reduzida se afasta do objeto já depositado e espera início de novo ciclo (figura 18):



Figura 18 – Robô se afasta do objeto com velocidade reduzida e aguarda novo ciclo.

Fonte: A autoria própria.

Para assegurar que o robô saiba a sua posição inicial e não ultrapasse os limites de atuação das juntas, foram utilizados sensores fim de curso que estabelecem a posição inicial do robô ao mesmo tempo em que interrompem o sinal enviado aos motores, sendo necessário um redirecionamento deste sinal em outra porta do microcontrolador para fazer o robô retornar em segurança.

Para que os motores se movimentassem de uma maneira mais eficaz, optou-se pelo uso de curva S de aceleração.

Para uma rampa com aceleração constante o sistema sofre com mudanças de aceleração abruptas no início do movimento e no ponto onde a velocidade é mínima/máxima. Estas variações da aceleração causam ao sistema problemas de vibração e *stress* excessivo dos componentes (figura 19):

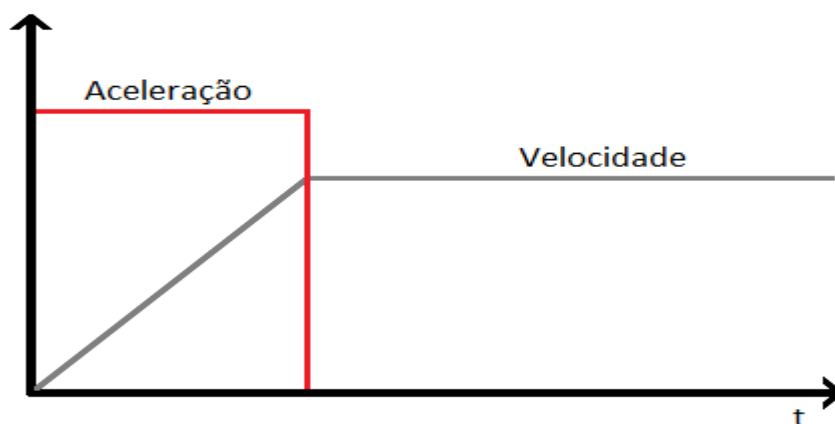


Figura 19 – Movimentação do robô.
Fonte: Autoria própria.

Já em um sistema com curva de aceleração tipo “S” os sistemas mecânicos não sofrem o choque da aceleração abrupta. No momento inicial a aceleração é nula, esta aumenta e diminui formando uma curva suave que pode ser dimensionada de acordo com o motor (figura 20):

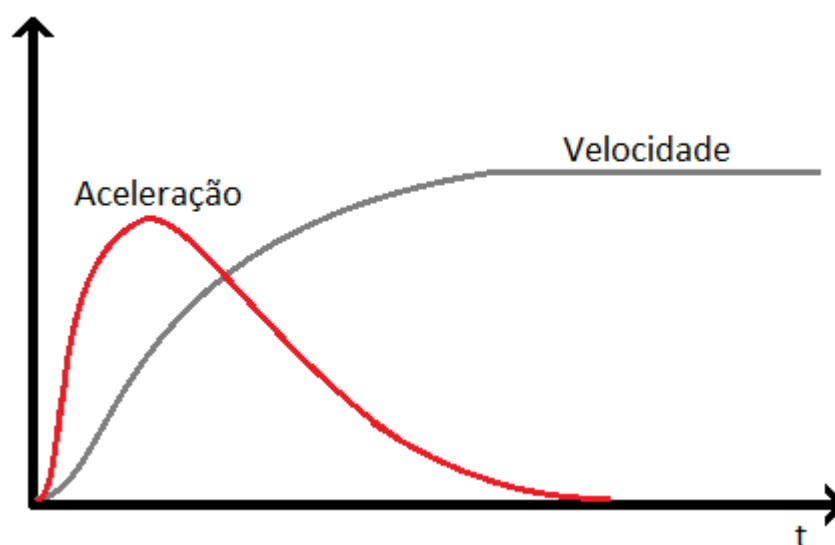


Figura 20 – Movimentação do robô.
Fonte: Autoria própria.

Entretanto, como o microcontrolador apresenta algumas restrições quanto a capacidade de processamento, devido a limitação de seu *clock*, foi necessário simplificar de maneira significativa a programação para que não fossem gerados problemas com a rampa de aceleração do robô.

Foi utilizada a seguinte fórmula para a geração de perfis de aceleração em tempo real:

$$C_i = C_{i-1} - \frac{2 \cdot C_{i-1}}{4 \cdot (i - m) + 1}, i < m \quad (8)$$

Onde:

- “C” é o valor utilizado para gerar as interrupções no microcontrolador;
- “i” é o numero do passo atual; e
- “m” é o numero total de passos.

Os ensaios realizados com os motores de passo mostraram-se satisfatórios com a utilização da fórmula mencionada acima, mas sendo necessário implementar um código que, dentro de cada interrupção, faz os cálculos de aceleração e altera o valor do temporizador responsável pela mesma. Assim não sendo afetado por outros códigos que rodam por fora da interrupção.

Para se trabalhar com os três motores foi desenvolvida e criada uma comunicação serial que transmite informações do *Labview* para os três PICs. No entanto, somente um dos PICs transmite informações para o *Labview*, sendo este PIC denominado de mestre, buscando-se assim evitar colisões, pois na comunicação RS232 não podem haver mais que dois dispositivos se comunicando no mesmo barramento. Com a realização de alguns testes detectou-se problemas na movimentação dos motores, uma vez que não havia garantias de que todos os microcontroladores estariam operando sincronizadamente, pois os mesmos não se comunicavam entre si. Buscando solucionar-se esse problema foi realizada a programação dos PICs para que fizessem as movimentações sincronizadas, para que isso fosse possível, na parte eletrônica dos mesmos foram realizadas conexões entre os microcontroladores escravos e o mestre criando uma forma de comunicação entre eles (sistema de *flag*) gerando um intertravamento. Para um melhor entendimento, as conexões físicas constam no apêndice A.

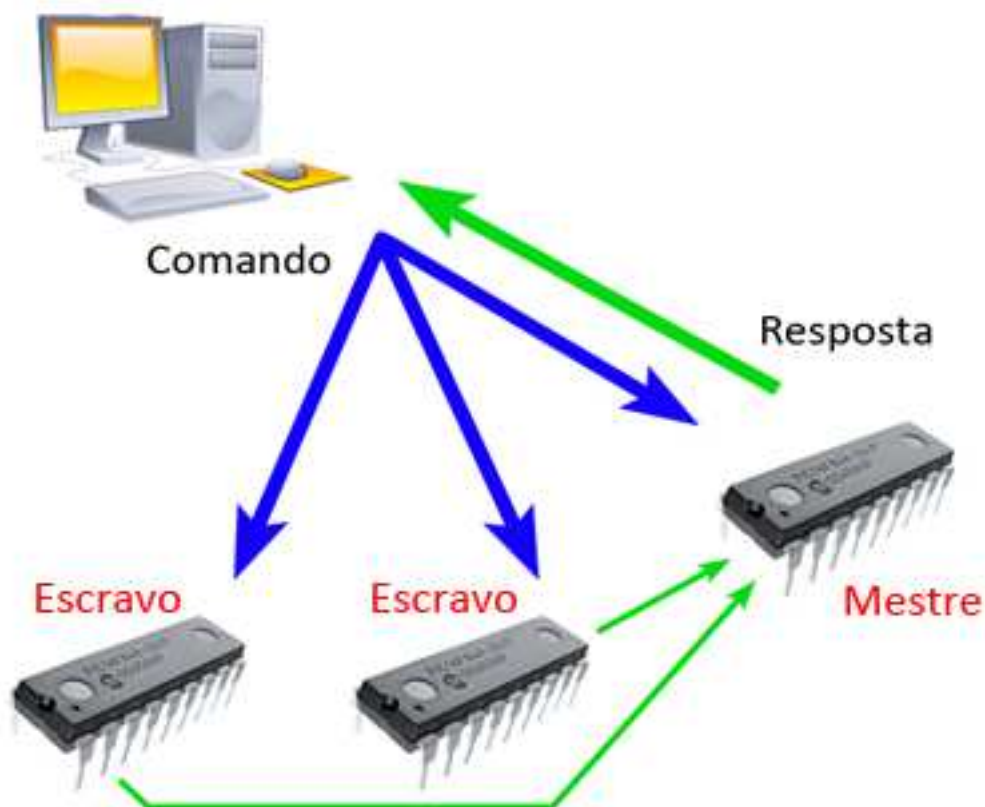


Figura 21 - Esquemático da comunicação serial.
Fonte: Autoria própria.

Primeiramente o *software labview* envia um comando para os microcontroladores, estes por sua vez vão checar os dados recebidos. Caso a informação recebida seja validada, o microcontrolador escravo atua uma saída para que o mestre possa identificar como sinal recebido. Caso tenha algum erro na informação o microcontrolador escravo ignora a mensagem e não atua a *flag* de mensagem recebida. O dispositivo mestre, quando recebe os sinais dos escravos dentro de um período de tempo específico, envia uma resposta para o *labview* para confirmar que os dados foram recebidos e todos foram validados. Em caso de problemas de comunicação (dispositivo mestre não recebendo o sinal dos escravos no período esperado ou recebendo um pacote de dados com erros) o microcontrolador mestre envia uma mensagem avisando o *labview* que houve problemas na comunicação e indica qual dos dispositivos recebeu a mensagem errada (apêndice C).

A área de comunicação diz respeito a parte que é responsável por realizar a conversão de sinais entre a área de processamento e o computador.

3.3 SOFTWARE

O *software Labview* é responsável pela realização de todos os cálculos relacionados ao processamento de imagem e movimentação do robô. Além disso, é capaz de tratar informações externas recebidas dos sensores ou dos microcontroladores. Os fluxogramas referentes às funções principais do robô estão contidas no apêndice F.

Para melhorar a estabilidade de movimento do robô foi implementado um código para o controle de aceleração e desaceleração que ameniza a inércia do robô. Como esse sistema precisa trabalhar em tempo real, para que eventuais atrasos ou rotinas extras não afetem o código da aceleração, o mesmo foi implementado na forma de *firmware* nos microcontroladores externos, através de interrupções de hardware.

O *software* escolhido para a programação do controle foi o *Labview* da *National Instruments*, devido à facilidade em se adquirir dados e em se realizar a modularização de códigos, tornando possível a implementação do recurso para aquisição de imagens de maneira mais simplificada, eficiente e direta, pois esse *software* possui módulos prontos para funções básicas de manipulação de imagem, como detecção de bordas, comparação de imagens com grau específico de acerto, entre outras, além de ser um programa que opera com uma linguagem de programação gráfica que faz uso de ícones para criação de suas aplicações. Diferenciando-se, portanto, das linguagens que realizam suas ações através de linhas de comando (texto).

A programação do *Labview* está baseada em fluxo de dados, onde um determinado fluxo de dados é responsável por determinar qual ação será executada. Embora em um primeiro momento o grupo não tivesse conhecimento prático dessa ferramenta, uma vez que a mesma nunca havia sido utilizada em projetos anteriores, optou-se por usá-la, pois, a mesma é conhecida pela sua facilidade de utilização dentro de variadas áreas de atuação.

Assim, iniciou-se a programação do sistema de visão com o auxílio de tutoriais que acompanhavam a própria ferramenta. Já nos primeiros testes esta ferramenta de programação mostrou-se amigável pela facilidade de aprendizado além dos bons resultados obtidos.

Paralelamente à programação do sistema de visão iniciou-se o estudo e programação da cinemática do robô. Foram necessárias várias pesquisas sobre os cálculos de cinemática inversa, uma vez que a cinemática do delta mostrava-se um tanto quanto complexa em um primeiro momento. Durante as pesquisas foram encontradas várias fontes de informações que continham muitos dados úteis a respeito da cinemática necessária para o robô, inclusive, entre elas, um código do *Labview* utilizado em um robô semelhante que foi apresentado em uma feira de exposições.

4 TESTES E RESULTADOS

O desenvolvimento dos desenhos dos componentes mecânicos em *CAD* foi realizado em mais de um momento até se atingir as características necessárias para o bom desempenho do robô delta.

Para a elaboração dos desenhos foi utilizado o *software Solid Works*. Esta etapa não demandou grandes dificuldades, já que o grupo possuía familiaridade com o programa usado.

O *software labview* foi a ferramenta utilizada para se realizar a programação do sistema de visão do robô e também para a implementação do sistema de controle do mesmo. A implementação do sistema de visão foi facilitada, pois, trabalhou-se com o módulo *Vision Assistant* que é um módulo auxiliar do *labview*, projetado para operar com sistemas de visão. Para a escolha da *webcam* considerou-se os *drivers* USB que eram compatíveis com esse módulo. Com a integração da *webcam* com o módulo de visão foi possível realizar os primeiros testes e, assim, verificar-se a viabilidade de utilizar o *software* para executar o sistema de visão. Para a melhoria da aquisição de imagem foi elaborado um sistema de iluminação fluorescente (apêndice B). Para a implementação do sistema de controle no *labview* foram criadas algumas funções básicas que, posteriormente, seriam incrementadas de maneira a tornar completo o sistema de controle. As funções básicas são: 1) as funções de configuração; 2) estabelecimento da posição inicial; 3) início do processo de controle contínuo; e 4) emergência. Partindo-se dessas funções foram desenvolvidos códigos mais complexos que permitiram a integração e execução dessas funções. O sistema de manipulação adotado no projeto para ser utilizado pelo robô foi um efetuador com ventosa em sua extremidade, sendo que esta opera de maneira pneumática, realizando a sucção dos objetos a serem manipulados. Com relação ao efetuador, decidiu-se pelo uso de uma ventosa de sucção, pois o peso desse conjunto é muito menor do que o de uma garra eletromecânica. O menor peso acaba diminuindo a inércia do robô, possibilitando, deste modo, que os movimentos sejam executados com melhor exatidão.

Foi possível construir o robô paralelo tipo delta fisicamente (embora incompleto) e apresentá-lo na ExpoUT 2011. Nesta feira ele foi programado para considerar como objeto fichas de pôquer e como lugar a depositar estas fichas,

círculos pretos impressos em uma folha de papel A4. Assim, toda vez que uma ficha era localizada pelo sistema de visão o robô atuava sobre a mesma para colocá-la sobre um dos círculos disponíveis.

4.1 DIFICULDADES ENCONTRADAS

A principal dificuldade encontrada refere-se ao fato da estrutura paralela ser algo relativamente recente e, portanto, não existirem muitos trabalhos acadêmicos e científicos em língua portuguesa. Além disso, existiram dificuldades em todas as áreas relativas ao desenvolvimento do robô, tais como, desenvolvimento mecânico, elétrico, eletrônico, sistema de visão e sistema de controle, sendo que as mesmas estão descritas nos capítulos específicos a cada área.

5 CONCLUSÃO

Neste trabalho foi apresentado um robô paralelo tipo delta associado com o uso de visão computacional e concebido com o objetivo de demonstrar suas características, bem como para realização de operações do tipo “pegar-e-posicionar”.

Com relação ao desenvolvimento do robô, construiu-se o mesmo após analisar o desempenho de diferentes tipos de protótipos construídos com diferentes tipos de matéria-prima. Chegando-se, ao final, a escolha do material que parecia mais apropriado para o projeto (os custos estão contidos no apêndice H).

Com relação à implementação do sistema de visão computacional, desenvolveu-se o mesmo com o uso de uma *webcam* simples, com um sistema de luz e com o uso de um módulo do *software Labview* que auxilia nesse tipo de aplicação.

Com relação à implementação do sistema de controle e supervisão, adotou-se o uso do *software Labview*, que tornou possível a realização dessa etapa.

O grupo foi capaz de integrar as etapas do projeto e desenvolver o protótipo do robô delta, demonstrando seu funcionamento e aplicabilidade na Expo UT 2011.

Todos os objetivos propostos na introdução do projeto foram atingidos de forma integral.

5.1 CONSIDERAÇÕES FINAIS

Dentro das expectativas do trabalho de conclusão de curso e tendo-se em mente a dificuldade do tema abordado, considera-se que o grupo teve um desempenho satisfatório, desenvolvendo o robô, complementando-o com a visão computacional e apresentando à comunidade acadêmica e industrial a arquitetura paralela e seus benefícios.

5.2 SUGESTÕES PARA TRABALHOS POSTERIORES

Uso de esteiras para realizar a chegada das peças a serem manipuladas pelo robô, possibilitando-se operar com a câmera fora da área de trabalho, evitando-se, deste modo, que as hastes do robô obstruam a visão da peça.

Implementação de sistema de interpolação de coordenadas a fim de se realizar trajetórias retilíneas com o robô.

Otimização e aprofundamento no sistema de captação, iluminação e processamento de imagens.

Melhorar processo de produção dos componentes mecânicos, minimizando-se, assim, problemas relacionados a folgas e precisão nos movimentos.

Implementar comunicação entre os microcontroladores mais robusta e, portanto, mais confiável e para que também exista padronização com outros sistemas.

O uso de servomotores no lugar dos motores-de-passo com o objetivo de atingir-se maior precisão e velocidade nas movimentações.

Desenvolver estrutura de suporte ao robô capaz de oferecer maior rigidez minimizando-se, desse modo, problemas relacionados à vibração.

REFERÊNCIAS

BOSCH-LIVE. Disponível em:

<http://www.bosch.com.br/content/language1/html/715_2181.htm>. Acesso em: novembro. 2010.

FINOTTI, Gilson. **Cálculo explícito dos torques dos atuadores de um robô paralelo plano empregando o método de Kane**. 2008. 199 f. Dissertação (Mestrado em Engenharia) – Escola Politécnica da Universidade de São Paulo, 2008. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3151/tde-11092008-141036/publico/adissertationr11_3__12_2_08.pdf>. Acesso em: 15 mar. 2010, 16:54.

GROOVER, Mikell et al. **Robótica: tecnologia e programação**. São Paulo: McGraw-Hill, 1988.

GROOVER, Mikell e ZIMMERS, Emory W. Jr. **CAD/CAM Computer-Aided design and Manufacturing**. Department of Industrial Engineering Lehigh University, Bethlehem, Pennsylvania, 1984.

MECHATRONIC-DESIGN. Disponível em:

<http://mechatronic-design.com/features/robot_redesign_pace/>. Acesso em: 01 março. 2010.

MERLET, Jean. P. **Parallel robots**. 2. ed. Holanda: Springer, 2006.allHall

NATIONAL INSTRUMENTS. Disponível em:

<<http://www.ni.com/labview/whatis/>>. Acesso em: 25 abril. 2011.

PARALLEMIC. Disponível em:

<<http://www.parallemic.org/Reviews/Review002p.html>>. Acesso em: 25 abril. 2011.

SICILIANO, Bruno e KHATIB, Oussama. **Springer Handbook of Robotics**. Department of computer Science Stanford University. Stanford, USA, 2008.

TROSSEN ROBOTICS, Disponível em:

<<http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot-kinematics-3276/>>. Acesso em: 22 abril. 2011.

APÊNDICE A – Diagrama de Blocos das Conexões Físicas (cabos).

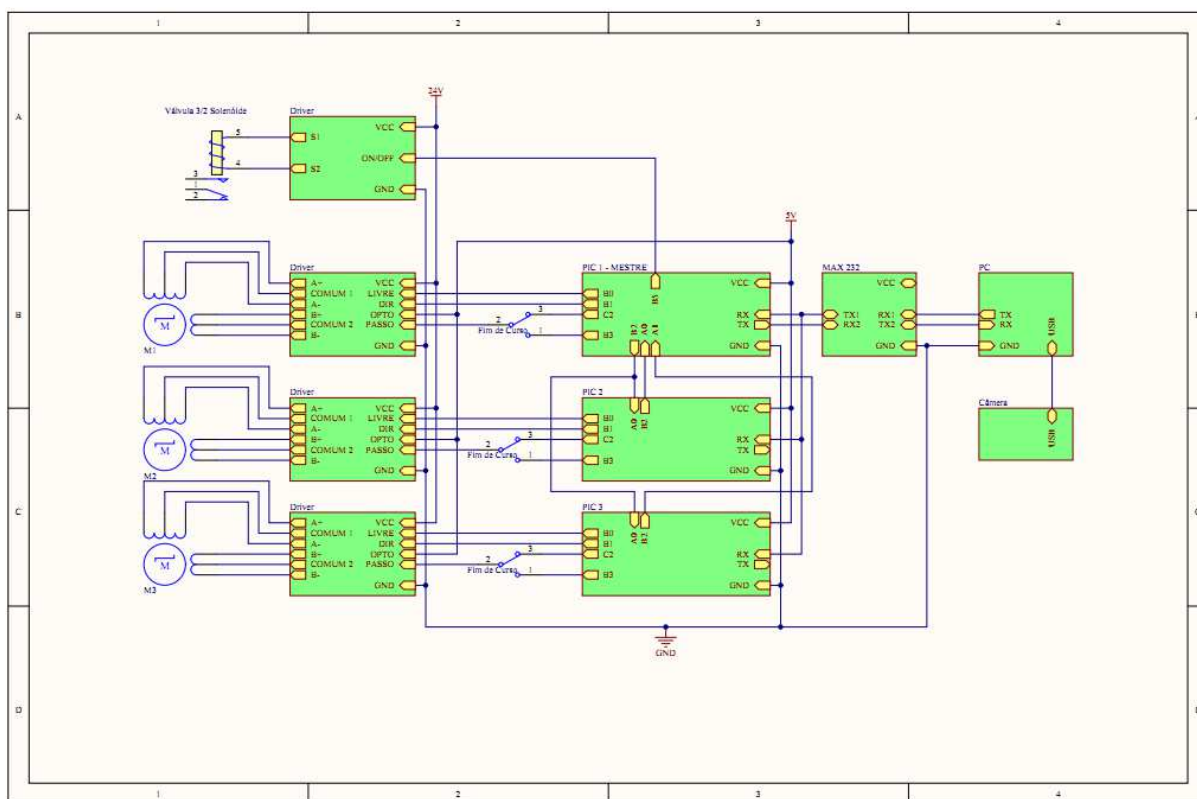


Figura 22 - Diagrama de blocos das conexões físicas (cabos).
Fonte: Autoria própria.

APÊNDICE B – Esquemático com Ligações do Sistema de Iluminação

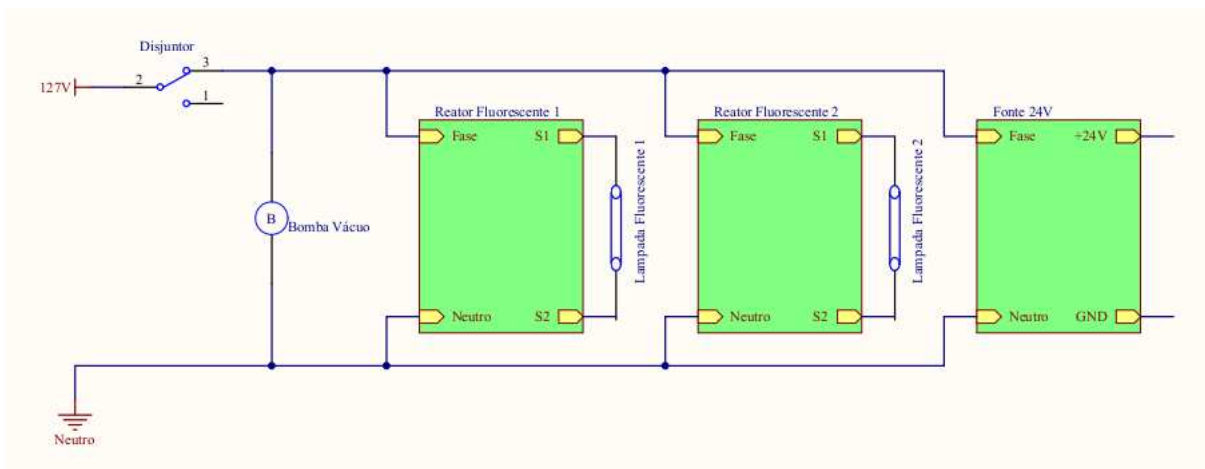


Figura 23 – Esquemático com Ligações do Sistema de Iluminação.
Fonte: Autoria própria.

APÊNDICE C – Esquemático da Placa de Comunicação RS232

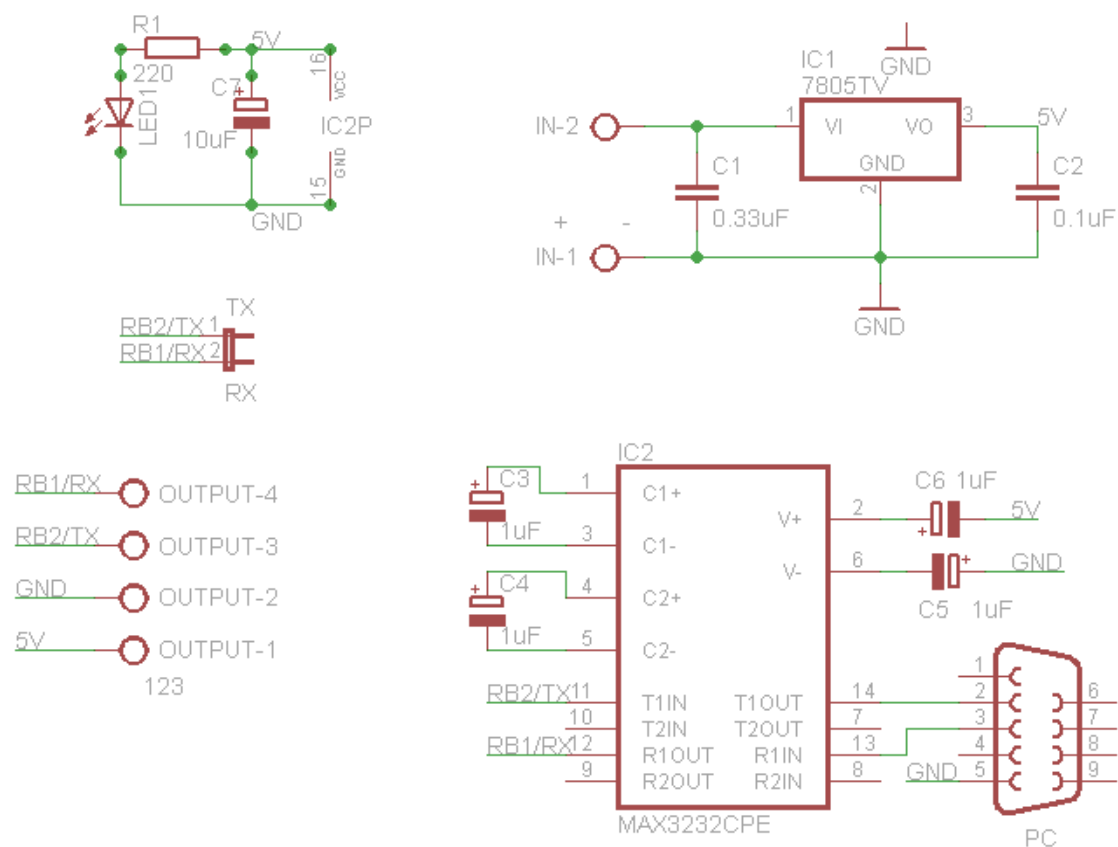


Figura 24 - Esquemático da placa de comunicação RS232.
Fonte: Autoria própria.

APÊNDICE D – Esquemático dos Microcontroladores

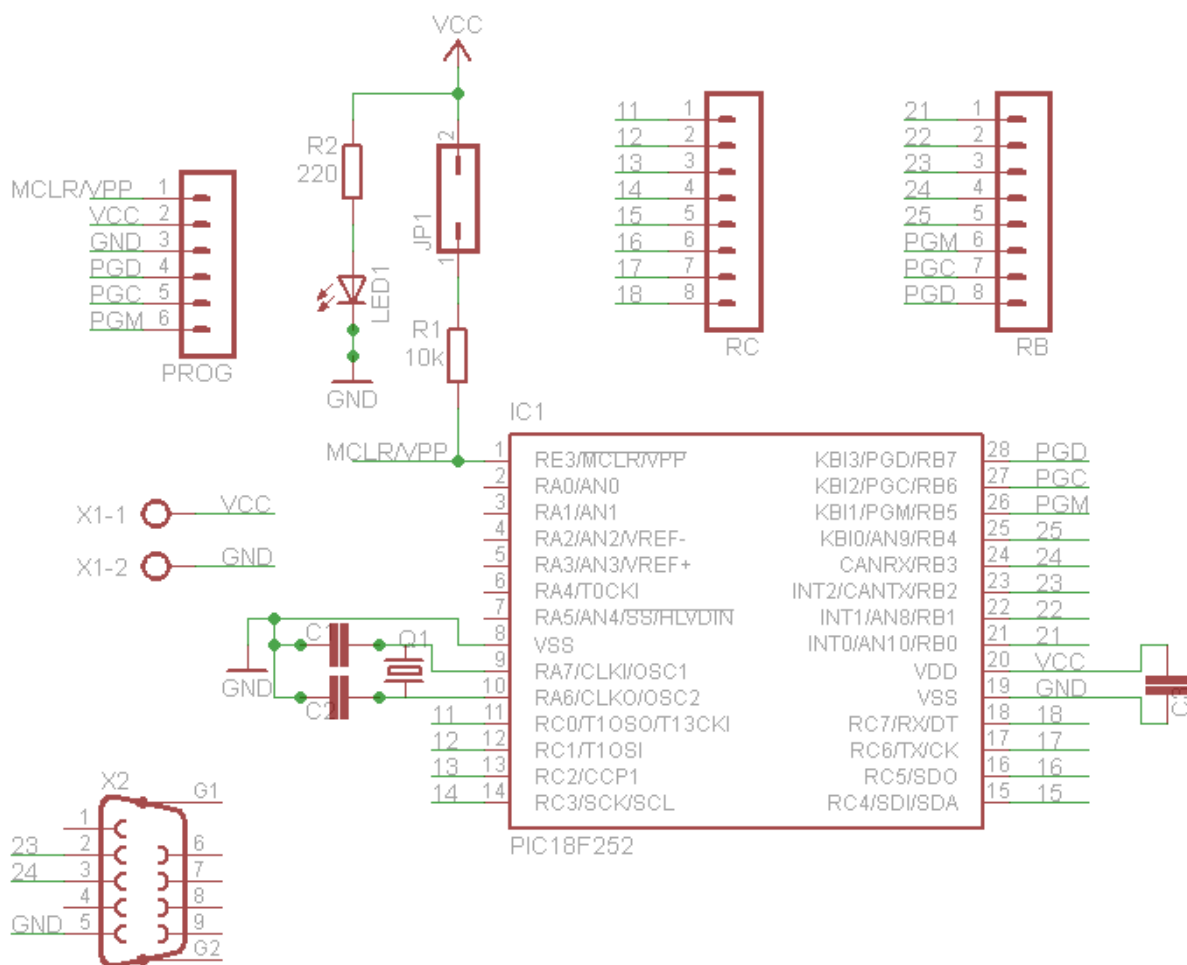


Figura 25 - Esquemático dos Microcontroladores.
Fonte: Autoria própria.

APÊNDICE E – Esquemático do Sistema Pneumático

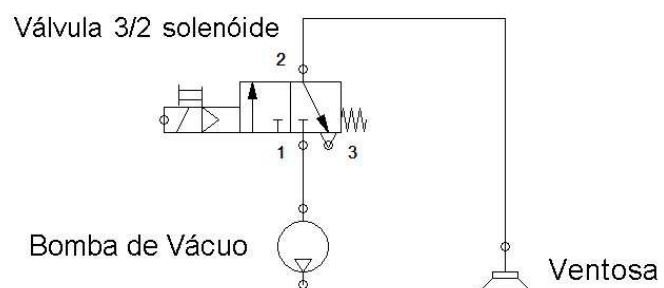


Figura 26 – Esquema do sistema pneumático.
Fonte: Autoria própria.

APÊNDICE F - Fluxogramas

Neste fluxograma está indicado o ciclo básico de funcionamento do robô.

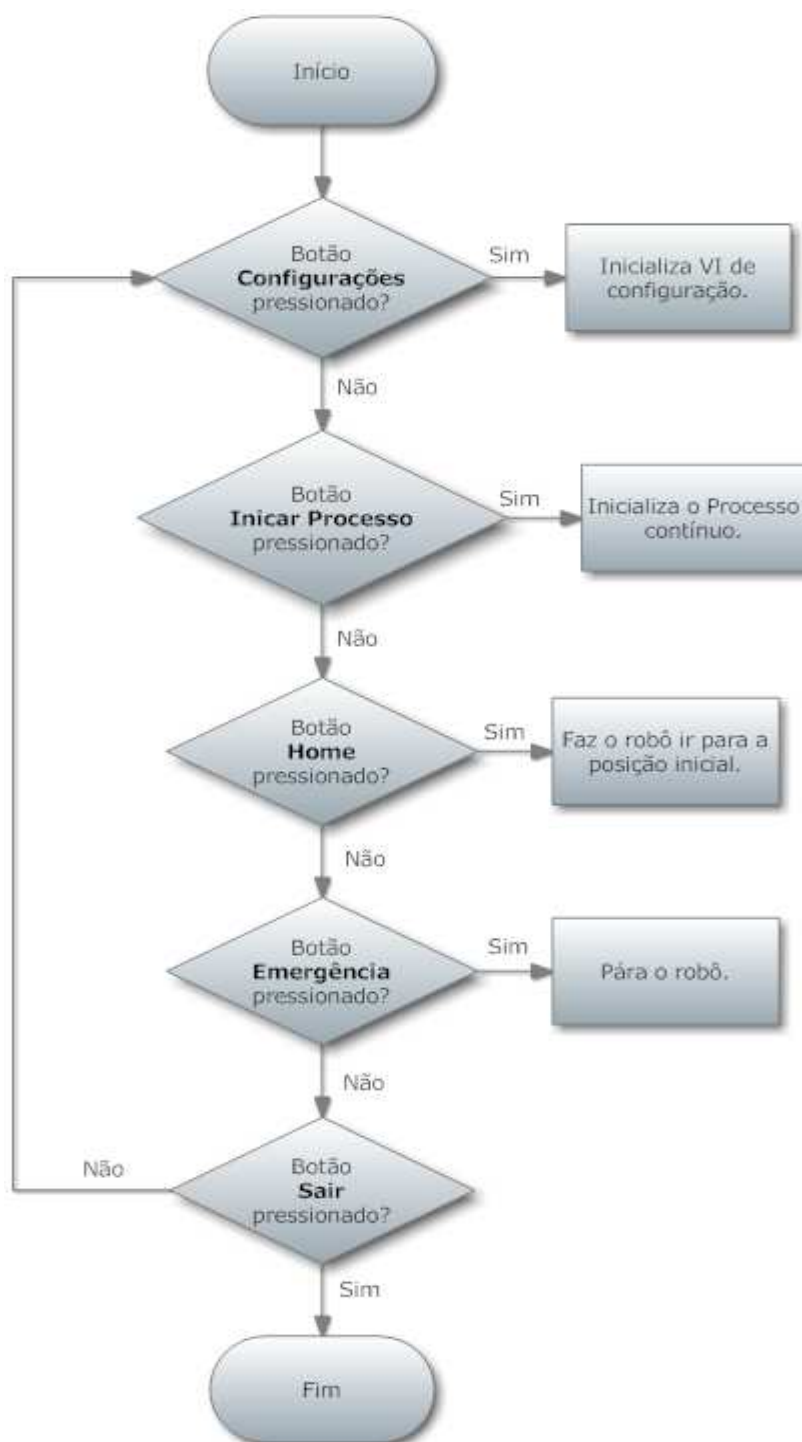


Figura 27 – Fluxograma Principal.
Fonte: Autoria própria.

Neste fluxograma está indicado uma síntese do programa do processo contínuo.

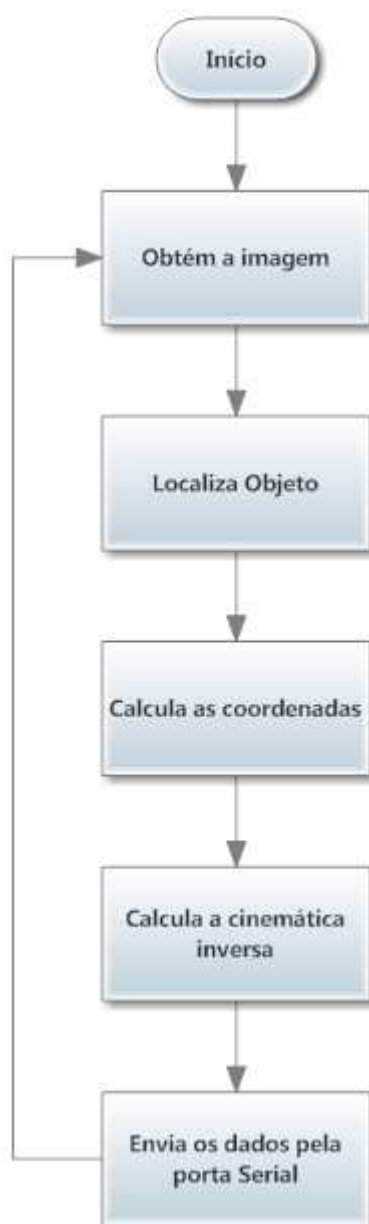


Figura 28 – Fluxograma do Processo Contínuo
Fonte: Autoria própria.

Árvore de Funções do Código em *Labview*



Figura 29 – Árvore de Funções do Código em Labview.
Fonte: Autoria própria.

Árvore de Subfunções da Função MAIN.VI

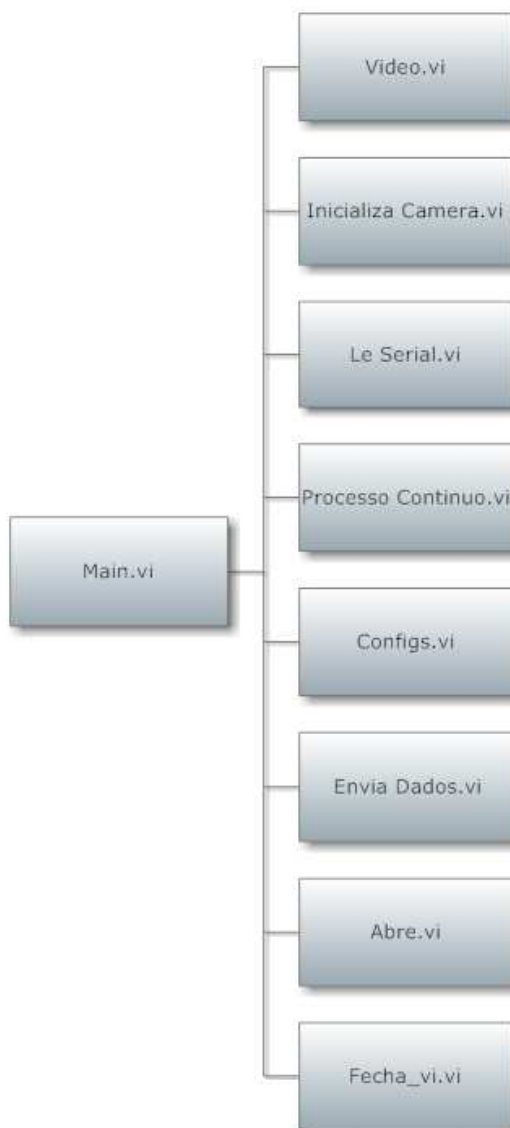


Figura 30 – Árvore de Subfunções da Função Main.VI.
Fonte: Autoria própria.

Main.vi

Esta é a função principal que engloba todas as outras funções do sistema de controle. É nesta função que as variáveis globais são inicializadas. Nela estão contidos menus de acesso que permitem ao usuário configurar os parâmetros de controle do robô, tais como, configurações de imagem e de cinemática. Possui, ainda, os laços (ligações em nível de programação) que permitem ao programa funcionar de maneira contínua e, além disso, funções para interromper a ação do robô em casos de emergência e funções que permitem movimentar o robô delta para a posição inicial.

Processo Continuo.vi

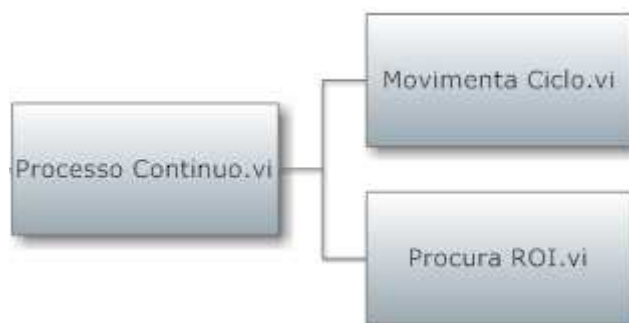


Figura 31 - Árvore de sub-funções da função Processo Continuo.VI.
Fonte: Autoria própria.

Esta função inicializa o processo contínuo de operação do robô. Ela é responsável por efetuar a diferenciação dos casos em que o robô está procurando pela forma e pela peça, agindo da maneira adequada em cada uma das situações. Em cada um dos casos são acionadas as respectivas funções para executar a busca pela imagem e os cálculos para obtenção e envio das coordenadas.

Movimenta Ciclo.vi

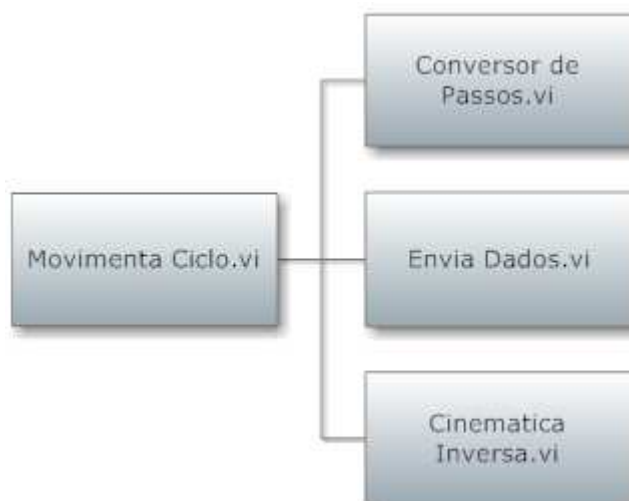


Figura 32 - Árvore de subfunções da função Movimenta Ciclo.VI.
Fonte: Autoria própria.

O principal objetivo desta função é dividir o movimento requisitado para o robô em três etapas. A primeira dessas etapas diz respeito ao próprio movimento, onde é realizada a aproximação do robô em relação ao alvo. A segunda diz respeito

à movimentação efetuada pelo robô quando este desce em direção ao alvo, movimentando-se somente na direção vertical, a fim de apanhar ou soltar a peça. Por fim, temos que a última etapa é o inverso da segunda, ou seja, aqui acontece o movimento de afastamento do robô em relação ao alvo, sendo que este também só é realizado na direção vertical. Essas etapas visam a garantir a integridade física do robô e da peça, para que não haja movimentos bruscos que poderiam causar avarias.

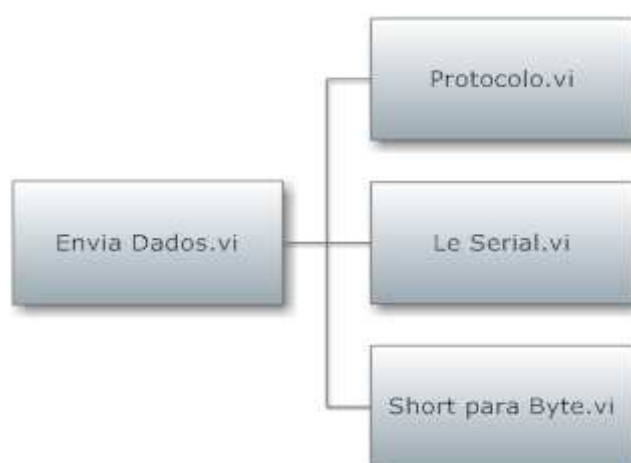
Cinemática Inversa.vi

Após as entradas das coordenadas no sistema cartesiano calcula a cinemática inversa do movimento do robô, obtendo os ângulos absolutos para cada motor de passo. Essa função é alimentada com os dados construtivos da mecânica que se referem ao comprimento das hastes superiores e inferiores, os tamanhos das plataformas e os ângulos relativos dos motores entre si.

Conversor de passos.vi

Calcula a diferença dos ângulos calculados pela cinemática inversa e a origem dos motores e converte o resultado em quantidade de passos.

Envia Dados.vi



**Figura 33 - Árvore de sub-funções da função Envia Dados.VI.
Fonte: Autoria própria.**

É responsável pelo envio das informações pela porta serial. Esta função realiza todo o tratamento necessário para assegurar o envio e recebimento das

informações sem que haja erros. Para isto, sempre que uma mensagem é enviada para os PICs ela espera a resposta de ACK (Acrônimo do Inglês para *Acknowledgment*, que significa reconhecimento). Cada tipo de mensagem enviada contém uma ACK específica para confirmação de recebimento. Quando há um erro de confirmação de dados a função reenvia os dados, garantindo assim a integridade da comunicação. Esta função também gerencia todos os tempos de espera para sincronizar as mensagens da serial, bem como os *timeouts* para cada tipo de transmissão, avisando o usuário ou reenviando dados.

Protocolo.vi

Responsável pela elaboração do protocolo de comunicação serial que segue:

Tabela 3 – Protocolo.vi.

Cabeçalho	Tipo	Aceleração						Posição			Checksum
		V_Max1	V_Min1	V_Max2	V_Min2	V_Max3	V_Min3	Posição 1	Posição 2	Posição 3	
0x7E	1 Byte	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	1 byte

Fonte: Autoria própria.

Cabeçalho

É indicado pelo valor 0x7E, que indica um novo início de pacote.

Tipo

Indica o tipo de comando que se deseja enviar. Os valores são:

- 0x00 - Ir para posição **Home**.
- 0x01 - Configurar a quantidade de passos do próximo movimento
- 0x02 - Calibrar
- 0x09 - Iniciar o movimento

V_Max

Indica o valor máximo para o uso no cálculo de aceleração, são 2 *bytes* para cada microcontrolador PIC.

V_Min

Indica o valor mínimo para o uso no cálculo de aceleração, são 2 *bytes* para cada microcontrolador PIC.

Posição

Indica a quantidade de passos a serem executados.

Checksum

Validação da mensagem. Para o cálculo do *Checksum* temos 0xFF menos o valor da soma de todos os *bytes* dos campos: Tipo, V_Max, V_min e Posição.

A fórmula para o cálculo da quantidade de passos necessários para movimentação dos motores é:

$$Qp = \frac{Pr \cdot Mp \cdot Ad}{360} \quad (9)$$

Onde:

- Qp é a Quantidade de passos.
- Pr é a quantidade de passos por revolução do motor.
- Mp é a relação de micropassos aplicada no driver.
- Ad é o ângulo desejado.

Levando-se em consideração que o motor é de 200 passos por revolução e está sendo controlado por um driver com 1/20 micropassos para realizar o movimento do motor com um deslocamento de 36 graus no sentido horário o cálculo do número de passos a ser enviado dá-se da seguinte maneira:

$$Qp = \frac{200 \cdot 20 \cdot 36}{360} = 400 \text{ passos} \quad (10)$$

Le serial.vi

Responsável por efetuar a leitura dos dados presentes na porta serial. A quantidade de dados a ser lida pode variar e é ajustada conforme o tipo de comando

enviado anteriormente. Esta função contém *timeouts* para leitura dos dados, que assegura que o programa não fique travado em caso de erro de envio dos PICs.

Funções da Câmera

Inicializa Camera.vi

Trata-se da etapa responsável por inicializar e identificar a porta USB em que a câmera está conectada, exportando, deste modo, a sessão da câmera.

Video.vi

Responsável por realizar a captura da imagem fornecida pela *webcam* e disponibilizar a sessão para o restante do código.

Apreende ROI.vi

Responsável por efetuar a extração referente ao ROI (acrônimo do inglês *Region of Interest*, que significa Região de Interesse), usando essa porção de imagem no processo de aprendizado de padrão. A imagem do ROI é salva em arquivo do tipo ROI.BMP.

Procura ROI.vi

Esta função permite que seja realizada uma busca pela região de interesse na imagem da câmera, comparando a imagem alvo (salva em arquivo) com a imagem que a câmera está fornecendo em tempo real. Após encontrar o objeto de interesse, calcula as coordenadas em *pixels* e converte para o sistema métrico tridimensional.

Neste fluxograma está indicado uma síntese das funções da câmera.



Figura 34 - Fluxograma das funções da câmera
Fonte: Autoria própria.

Configurações

Configs.vi

Abre o menu onde estão contidas as opções de configurações disponíveis.

Configs robo.vi

Abre o menu onde estão contidas as opções de configurações disponíveis relativas à mecânica do robô. As propriedades que podem ser configuradas acessando-se esse menu são: dimensões físicas do Robô, ângulo entre os motores, valores para aceleração/desaceleração do movimento do robô, distâncias de afastamento da peça, altura da peça, altura da forma e posição inicial (home).

Configs camera.vi

Abre o menu onde estão contidas as opções de configurações disponíveis relativas à câmera do robô. As propriedades que podem ser configuradas acessando-se esse menu são: os ROIs da peça e da forma; além de permitir ao usuário configurar o centro e a calibração da imagem, assegurando-se, assim, a correção de eventuais erros de perspectiva.

Configs serial.vi

Configura os parâmetros que serão usados para abrir a porta serial. Os parâmetros configurados são: *baud rate*, paridade, controle de fluxo, *timeout*, nome da porta, quantidade de bits de dados e bit de parada.

Abre_vi.vi

Função para abrir os painéis de controles de VIs para a configuração de parâmetros.

Fecha_vi.vi

Função para fechar os painéis de controles de VIs.

Short para Byte.vi

Função que quebra um número de 16 *bits* (*short*) em 2 números de 8 *bits* (*byte*)

APÊNDICE G – Configuração dos Microcontroladores

O microcontrolador escolhido foi o PIC 18f2550 e a programação foi realizada em linguagem C. O objetivo principal do microcontrolador é executar o cálculo em tempo real da aceleração e desaceleração do movimento dos motores de passo, sendo também necessário executar a comunicação dos mesmo com os outros PICs e com o *Labview*. As funções do código em C são:

```
void main()
```

É a primeira função que o microcontrolador executa ao iniciar. Esta função executa as subfunções de inicialização de parâmetros de *hardware*. Também é esta função que interpreta os tipos de comandos que o *Labview* envia e toma as ações necessárias para cada tipo de comando.

```
void initialize(void)
```

É a função que inicializa os parâmetros das funções de interrupção e contadores internos, valores iniciais dos pinos de *output*, bem como zera a *flag* que indica que o microcontrolador está processando um movimento.

```
void motor_run(signed int16 pos_new)
```

É a função que calcula a quantidade de passos que se precisa para alcançar a posição absoluta recebida do *Labview* tendo a posição relativa que é a última posição que o motor parou. Inicializa os valores dos contadores internos do microcontrolador que serão usados para o cálculo da rampa de aceleração/desaceleração e habilita a interrupção que fará esses cálculos em tempo real, dando início ao movimento do robô.

```
void isr_motor_step(void)
```

É a função de interrupção que realiza a máquina de estados do cálculo da rampa de aceleração. Está dividida em quatro estados:

- *ramp_up*
- *ramp_down*
- *ramp_max*
- *default*

O estado *ramp_up* indica que o cálculo deve ser feito de forma a acelerar o movimento do robô.

O estado *ramp_down* indica que o cálculo deve ser feito de forma a desacelerar o movimento do robô.

O estado *ramp_max* indica que o robô executou a quantidade necessária de passos na aceleração e parte para um movimento uniforme.

O estado *default* serve para limpar as *flags* e parar a máquina de estados quando a mesma alcança seu último passo.

Cada vez que essa função é executada um passo é dado e são realizados os cálculos para obter-se o novo valor do contador para gerar a aceleração ou desaceleração que, por modo de **Capture Compare**, ativará novamente a interrupção para o passo seguinte.

void com_serial(void)

É a função que recebe e valida as mensagens do *labview* separando cada tipo de informação em variáveis apropriadas para a interpretação. Para toda mensagem é realizado um cálculo de *Checksum* para verificação da integridade da mensagem, e caso esse valor não corresponda ao valor calculado pelo *software labview* para a mensagem, esta é descartada e o microcontrolador envia uma mensagem de *Nack*, fazendo assim com que o programa envie novamente a mensagem. As mensagens que apresentam erro de *Checksum* são armazenadas e podem ser acessadas pelo *labview* para uma análise do tipo de erro ocorrido.

void wait_pics(void)

O objetivo desta função é fazer com que o microcontrolador mestre tenha intertravamento com os outros dois microcontroladores escravos. Isso é necessário para que o mestre saiba quando os outros microcontroladores estão realizando um movimento ou estão esperando o recebimento de novos dados.

APÊNDICE H– Orçamento

Foi realizado orçamento com o objetivo de verificar-se a viabilidade econômica em desenvolver-se o projeto. Os valores podem ser vistos na tabela abaixo:

Tabela 4 – Orçamento.

		Preço	Quantidade	IPi	Frete	Total em US\$	Total em R\$
MECÂNICA	Perfis de alumínio KANYA 30x30mm B03-1 anodizados. Fornecedor: ABG Curitiba	R\$ 20,76/m	11,48m	-	-	-	R\$ 238,35
	Cantoneiras (com inserto) B30-40 Fornecedor: ABG Curitiba	R\$ 10,70/pc	8 peças	-	-	-	R\$ 85,60
	Cantoneiras simples B30-10 Fornecedor: ABG Curitiba	R\$ 6,38/ pc	10 peças	-	-	-	R\$ 63,80
	Porca B32-60 Fornecedor: ABG Curitiba	R\$ 1,35/ pc	62 unidades	-	-	-	R\$ 83,70
	Porca B32-50 Fornecedor: ABG Curitiba	R\$ 1,35/ pc	10 unidades	-	-	-	R\$ 13,50
	Tubo de Fibra de Carbono 6mm L800mm Fornecedor: eBay, Hong Kong	US\$24,46/ 3 pc	1 unidade	-	Incluso	US\$24,46	-
	Materiais diversos (chapas e tarugos de alumínio, tubo de inox)	-	-	-	-	-	R\$ 40,00
	Acoplamento esférico	-	-	-	-	-	-
	Fixador dupla face 3M	R\$ 9,90/ pc	1 unidade	-	-	-	R\$ 9,90
	Tábua de MDF	R\$ 25,90/ pc	1 unidade	-	-	-	R\$ 25,90

	Abraçadeiras hellerman	R\$ 3,50/ pacote	-	-	-	-	R\$ 3,50
	Parafusos e arruelas	-	-	-	-	-	R\$ 30,00
HARDWARE	Motor de passo NEMA 23mm 15Kgf.cm (AK57H/3-1.8) Fornecedor: Akiyama Curitiba	US\$34,33/ pc	3 peças	5,00%	-	US\$108,15 (Câmbio \$1,90)	R\$ 205,50
	Drive de micro-passo. Corrente máx. 3,5ª. Tensão máx. 40VDC (AKDMP5-3.5A) Fornecedor: Akiyama Curitiba	US\$103,57/ pc	3 peças	5,00%	-	US\$326,25 (Câmbio \$1,90)	R\$ 620,00
	Fonte de alimentação 24V 15A Fornecedor: easysupply (eBay)	US\$59,99/ pc	1 peça	-	US\$14	US\$73,99	-
	Fonte de alimentação 24V 6.5A Fornecedor: carolbrent (eBay)	US\$29,99/ pc	1 peça	-	US\$11	US\$40,99	-
	Mangueira de silicone 6mm	-	-	-	-	-	R\$ 8,00
	Mangueira pneumática 6mm	-	-	-	-	-	R\$ 6,00
	Rolamento	-	-	-	-	-	R\$ 1,50
	CA RÔNI ELET	Microcontrolador PIC 18F252	R\$19,90/ pc	3 peças	-	-	-

	Microcontrolador PIC 18F4550	R\$29,50/ pc	3 peças	-	-	-	R\$ 88,50
	Diversos (Max232, capacitores, resistores, cristais, placa de cobre – fenolite, Verniz	-	-	-	-	-	R\$ 35,00
	Cabos	-	-	-	-	-	R\$ 20,00
	Conectores	-	-	-	-	-	R\$ 8,50
	Chave fim de curso	1,50/ pc	3 peças	-	-	-	R\$ 4,50
	Lâmpadas fluorescentes	R\$ 4,00/ pc	2 unidades	-	-	-	R\$ 8,00
	Reatores	R\$ 9,90/ pc	2 unidades	-	-	-	R\$ 19,80
	Bocais para lâmpada fluorescente	R\$ 0,45/ pc	4 unidades	-	-	-	R\$ 1,80
	Disjuntor 10A	R\$ 6,90/ pc	1 unidade	-	-	-	R\$ 6,90
	Chave liga/desliga para lâmpada	R\$ 2,90/ pc	1 unidade	-	-	-	R\$ 2,90
	Fio paralelo branco	R\$ 0,95/ m	6 metros	-	-	-	R\$ 5,70
	Tomada	R\$ 4,30/ pc	1 unidade	-	-	-	R\$ 4,30

Fonte: Autoria própria.

O gasto aproximado do projeto com materiais foi de R\$ 1965,70. Valor considerado viável pelo grupo. O tempo aproximado de pesquisa e execução do projeto foi de 1800h.