

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTOS ACADÊMICOS DE MECÂNICA E ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

**MECANISMO PARALELO ESFÉRICO ORIENTADOR DE CÂMERA
PARA MONITORAMENTO DE IMAGENS**

CARLOS ALBERTO GLISTAU

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2018

CARLOS ALBERTO GLISTAU

MECANISMO PARALELO ESFÉRICO ORIENTADOR DE CÂMERA PARA MONITORAMENTO DE IMAGENS

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

Orientador: Prof. Dr. Rubens Alexandre de Faria.
Coorientador: Prof. Ubiradir Mendes Pinto.

CURITIBA
2018

TERMO DE APROVAÇÃO

CARLOS ALBERTO GLISTAU

MECANISMO PARALELO ESFÉRICO ORIENTADOR DE CÂMERA PARA MONITORAMENTO DE IMAGENS

Este Trabalho de Conclusão de Curso foi apresentado no dia 03 de julho de 2018, como requisito parcial para a obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo-assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

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

Prof. MSc. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Miraldo Matuichuk
UTFPR

Prof. Dr. Celso Salamon
UTFPR

Prof. Dr. Rubens Alexandre de Faria
Orientador

Prof. Ubiradir Mendes Pinto
Coorientador

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

À memória de minha mãe, Eva Kreitler (Evinha).
Por toda ajuda, apoio e incentivo que tive da pessoa
mais importante da minha vida.

AGRADECIMENTOS

À Deus pela minha vida.

Ao meu orientador Rubens Alexandre de Faria, pela amizade e apoio.

Aos amigos e colegas: Augusto César Fantinelli de Carvalho, Anderson Chiquito e Miraldo Matuichuk, pela inestimável ajuda na fabricação das peças usadas no projeto.

A todos os professores e colegas, que de uma forma ou outra contribuíram para que eu chegasse até aqui.

“Science without religion is lame, religion without science is blind.” (Albert Einstein, NY, 1941).

“A ciência sem a religião é manca, a religião sem a ciência é cega.” (Albert Einstein, NY, 1941).

RESUMO

GLISTAU, Carlos Alberto. **Mecanismo Paralelo Esférico Orientador de Câmera para Monitoramento de Imagens**. 2018. 46 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Este trabalho tem por objetivo produzir um mecanismo paralelo esférico orientador de câmera, com um sistema de controle que pode orientá-la para qualquer direção, em um ângulo de até 140°. O resultado é mostrado através da implementação e testes em um protótipo funcional, confeccionado com materiais facilmente adquiridos no comércio local, acoplado a uma câmera CCD (*charge-coupled device*) para visualização periférica. Também foi gerada uma documentação técnica, com a finalidade de motivar trabalhos futuros que se proponham a abordar esse tipo de mecanismo.

Palavras chave: *Agile Eye*. Mecanismo Paralelo Esférico. Arduino. Servomotores.

ABSTRACT

GLISTAU, Carlos Alberto. **Spherical Parallel Camera Guidance Engine for Image Monitoring**. 2018. 46 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial) Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This work aims to produce a parallel spherical camera guiding mechanism, with a control system that can guide it to any direction, at an angle of up to 140 °. The result is shown through the implementation and testing in a functional prototype, made with materials easily acquired in the local trade, coupled to a CCD (charge-coupled device) camera for peripheral visualization. A technical documentation was also generated, with the purpose of motivating future works that propose to approach this type of mechanism.

Keywords: *Agile Eye*. Mecanismo Paralelo Esférico. Arduino. Servomotores.

LISTA DE FIGURAS

Figura 1 - Mecanismo paralelo esférico	10
Figura 2 - (a) Modelo do Agile Eye; (b) Segmento proximal de cada perna do mecanismo.....	14
Figura 3- Modelo cinemático de um manipulador esférico paralelo: (a) convenção de simbologia e (b) espaço de trabalho descrito.....	15
Figura 4 - Segmentos do mecanismo paralelo esférico.....	18
Figura 5 - Mecanismo Paralelo Esférico . (a) posição inicial; (b) centro de rotação do mecanismo.....	19
Figura 6 - Placa Arduino Uno R3.....	21
Figura 7 - Componentes de um servo motor.....	22
Figura 8 - Sinais monitorados pelo circuito de controle.....	23
Figura 9 - Sinais PWM para o controle de um Servo Motor	24
Figura 10 - Câmera fixada no apontador.....	25
Figura 11 - Servomotor Tower Pro SG-5010.....	26
Figura 12 - Acoplamento conectando os eixos do servomotor e do atuador.....	27
Figura 13 - Vista do circuito de controle dos servomotores.....	28
Figura 14 - Imagens da tela do monitor serial do Arduino Software (IDE)	34
Figura 15 - Manipulador Paralelo Esférico montado com atuadores de dois segmentos	39
Figura 16 - Base.....	40
Figura 17- Suporte dos eixos	41
Figura 18 - Segmento externo do atuador.....	42
Figura 19 - Segmento interno do atuador.....	43
Figura 20 - Apontador	44
Figura 21 - Esquemático do circuito de controle dos servosmotores	45

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA	11
1.2	OBJETIVOS	11
1.2.1	Objetivo Geral	11
1.2.2	Objetivos Específicos	11
1.3	PROCEDIMENTOS METODOLÓGICOS	12
1.3.1	Projeto Mecânico	12
1.3.2	Desenvolvimento do Firmware	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	MANIPULADOR PARALELO ESFÉRICO E O AGILE EYE	13
2.2	MODELO CINEMÁTICO DO MANIPULADOR ESFÉRICO COM ATUADORES DE DOIS SEGMENTOS	14
2.3	AGILE EYE	18
2.4	CAD (COMPUTER AIDED DESIGN)	19
2.5	CNC (CONTROLE NUMÉRICO COMPUTADORIZADO)	20
2.6	A PLATAFORMA ARDUINO	20
2.7	SERVO MOTORES	21
2.8	CONTROLE DO SERVOMOTOR	23
2.8.1	PWM (<i>Pulse Width Modulation</i>)	23
3	DESENVOLVIMENTO	25
3.1	PARTES MECÂNICAS	25
3.2	PARTE ELETRÔNICA	27
3.2.1	Descrição da placa do Arduino UNO R3 utilizada	29
4	TESTES E RESULTADOS	32
4.1	DESEMPENHO DO MECANISMO PARALELO ESFÉRICO	32
4.1.1	Velocidade angular	32
4.1.2	Alcance rotacional	32
4.1.3	Precisão de posicionamento	33
4.1.4	Servomotor TOWER PRO SG-5010	33
4.2	FUNCIONAMENTO DO SOFTWARE	33
5	CONCLUSÃO	35
5.1	SUGESTÕES PARA TRABALHOS POSTERIORES	36
	REFERÊNCIAS	37
	APÊNDICE A - Manipulador Paralelo Esférico montado com atuadores de dois segmentos	39
	APÊNDICE B – Base	40
	APÊNDICE C – Suporte dos eixos	41
	APÊNDICE D – Segmento externo do atuador	42
	APÊNDICE E – Segmento interno do atuador	43
	APÊNDICE F- Apontador	44
	APÊNDICE G – Esquemático do circuito de controle dos servomotores	45
	APÊNDICE H – Programa do controle dos servomotores	46

1 INTRODUÇÃO

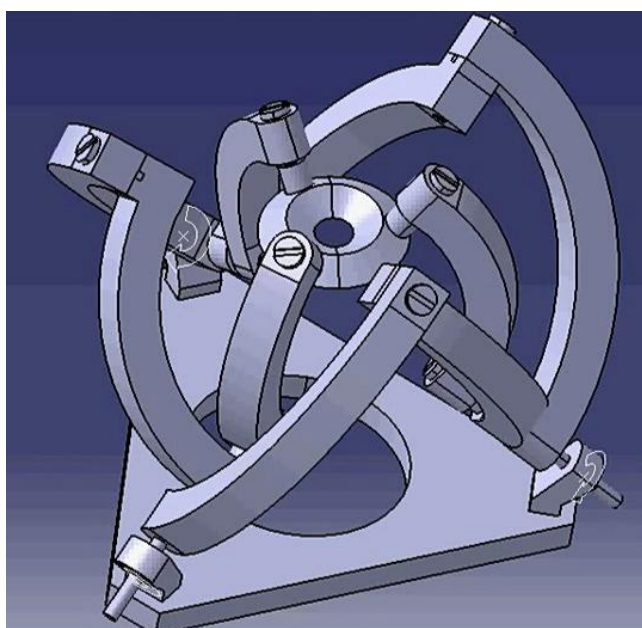
Seja na indústria ou em sistemas de vigilância, existe uma crescente necessidade por mecanismos de monitoramento de imagens que garantam controle e segurança. Para suprir esta demanda, diferentes tipos de produtos podem ser encontrados.

Na maioria das vezes, as câmeras dos sistemas de visão são limitadas quanto à velocidade de movimento, direção e alcance do ângulo de visão. A velocidade é limitada, pois uma movimentação rápida de mecanismos com atuadores independentes, produz oscilações provocadas pela inércia em suas articulações.

No caso de mecanismos paralelos, estes são caracterizados pelo fato de que as extremidades dos atuadores estão ligadas à base através de múltiplas cadeias cinemáticas. Isto leva a uma alta rigidez e propriedades dinâmicas muito boas com a inércia das partes móveis consideravelmente reduzida. (AGILE EYE, 2017).

A Figura 1, ilustra o mecanismo paralelo esférico do projeto original.

Figura 1 - Mecanismo paralelo esférico



Fonte: AGILE EYE (2017).

1.1 JUSTIFICATIVA

O mecanismo paralelo esférico constitui uma grande vantagem em relação aos demais usados para orientação de câmeras nos sistemas de monitoramento de imagens. O desempenho em termos de velocidade, aceleração e mudanças de direção, não é encontrado em mecanismos de equipamentos normalmente oferecidos no mercado. São características importantes quando, por exemplo, se deseja rastrear objetos em movimento e com trajetória inconstante.

1.2 OBJETIVOS

Os objetivos deste trabalho estão descritos a seguir.

1.2.1 Objetivo Geral

Desenvolver e implementar um protótipo de mecanismo paralelo esférico, com três graus de liberdade, orientador de direção em um ângulo de até 140° e um sistema de controle para ser utilizado com câmeras CCD (*charge-coupled device*) no monitoramento de imagens.

1.2.2 Objetivos Específicos

- 1 - Conhecer os princípios que explicam a baixa inércia na dinâmica de atuadores ligados através de múltiplas cadeias cinemáticas.
- 2 - Modelar as peças e simular o funcionamento no CAD (*Computer Aided Design*), mediante modelo (*Agile Eye*), do mecanismo paralelo esférico.
- 3 - Definir os materiais a serem utilizados.

- 4 - Usinar as peças do mecanismo paralelo esférico.
- 5 - Realizar a montagem do protótipo do mecanismo paralelo esférico.
- 6 – Desenvolver o *hardware* e o *firmware* de controle do protótipo.
- 7 - Realizar a montagem eletroeletrônica.
- 8 - Executar testes de funcionamento e ajustes.

1.3 PROCEDIMENTOS METODOLÓGICOS

A execução do trabalho foi dividida nas seguintes etapas:

1.3.1 Projeto Mecânico

A modelagem do mecanismo paralelo esférico foi executada no programa Solid Works 2016 (*Dassault Systemes SolidWorks Corporation*). Nele também foi feita a verificação de possíveis colisões, interferências e a resistência das peças em relação ao material usado na sua fabricação.

1.3.2 Desenvolvimento do Firmware

Para o desenvolvimento do *firmware* foi escolhida a plataforma Arduino para atuar no controle dos servos motores do protótipo.

Arduino é uma plataforma eletrônica de *hardware* e *software* livre de fácil utilização. Os programas para o Arduino são escritos em Linguagem C e C++.

O Arduino IDE vem com uma biblioteca composta por um conjunto de funcionalidades que facilitam o trabalho de programação (ARDUINO, 2017).

2 FUNDAMENTAÇÃO TEÓRICA

A seguir serão explanados os principais temas que auxiliaram no desenvolvimento deste trabalho.

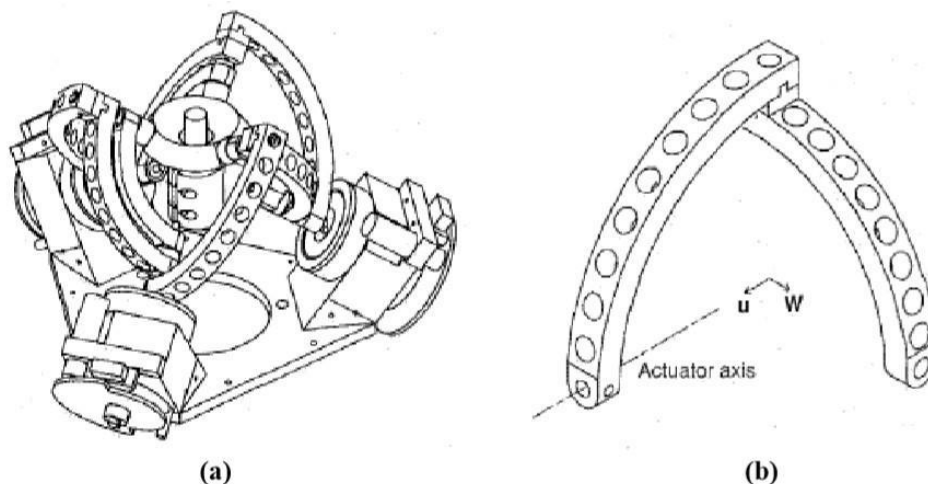
2.1 MANIPULADOR PARALELO ESFÉRICO E O AGILE EYE

Um manipulador paralelo esférico é geralmente composto por duas plataformas em forma de pirâmide. As plataformas, base e móvel, são conectadas por três atuadores igualmente espaçados, contendo somente articulações de rotação. Os eixos de todas as articulações se interceptam num ponto comum, ou seja, o centro de rotação da estrutura.

O movimento do apontador fica confinado à superfície de uma esfera centrada no centro de rotação, resultando em três graus de liberdade compostos apenas de rotações (BAI, 2010).

A maioria das aplicações de mecanismos esféricos paralelos pode ser encontrada em dispositivos de orientação, como articulações de rotação, orientação de ferramentas médicas (LI; PAYANDEH, 2002) e dispositivos de orientação de câmera, como por exemplo: o *Agile Eye*, Figuras 2a e b (GOSSELIN; HAMEL, 1994).

Figura 2 - (a) Modelo do Agile Eye; (b) Segmento proximal de cada perna do mecanismo.



Fonte: adaptado de GOSSELIN; HAMEL (1994).

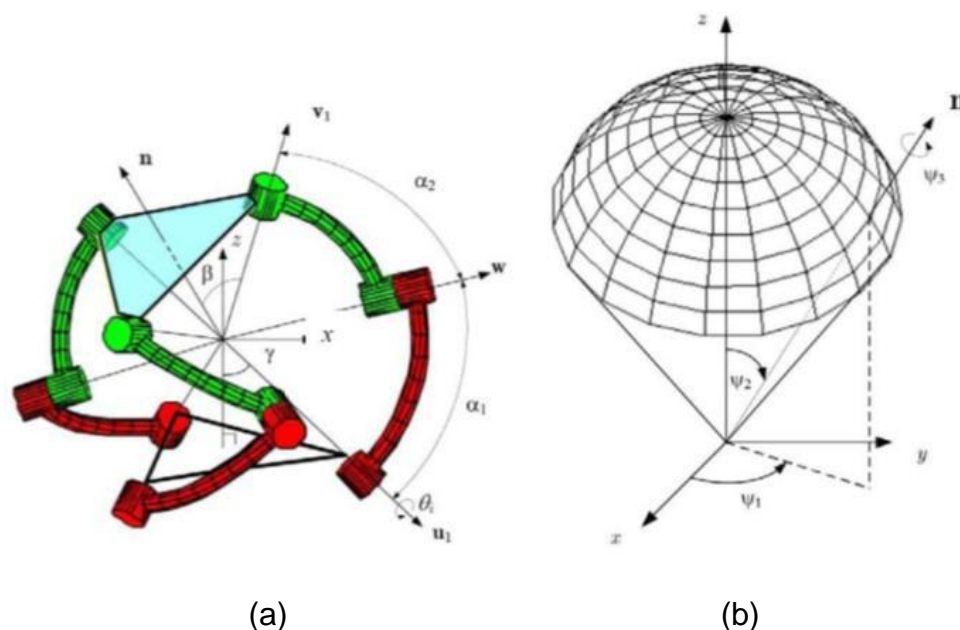
2.2 MODELO CINEMÁTICO DO MANIPULADOR ESFÉRICO COM ATUADORES DE DOIS SEGMENTOS.

A Figura 3(a) apresenta um manipulador esférico paralelo genérico.

O manipulador consiste somente de articulações de rotação, cujos eixos são denotados pelos vetores unitários u_i , v_i , e w_i . Os três segmentos (vermelhos) conectados à plataforma base tem dimensões idênticas α_1 , enquanto os três segmentos (verdes) que são conectados à plataforma móvel tem dimensões idênticas α_2 . Além disto, β e γ definem a geometria de duas pirâmides regulares da base e da plataforma móvel. A articulação esférica ativa é um caso especial de manipulador esférico paralelo para o qual $\gamma = 0$ (BAI, 2010).

Para a estrutura *Agile Eye*, tem-se um caso específico onde $\beta = \gamma$ (GOSSELIN; HAMEL, 1994), e os parâmetros de construção são: $\alpha_1 = 90^\circ$; $\alpha_2 = 90^\circ$; $\beta = \gamma = 54.74^\circ$ e $n = 45.34$ mm. Sendo n o módulo do vetor n , Figura 3(a).

Figura 3- Modelo cinemático de um manipulador esférico paralelo: (a) convenção de simbologia e (b) espaço de trabalho descrito.



Fonte: adaptado de BAI (2010).

Um Sistema de coordenadas é escolhido para o manipulador paralelo esférico, com origem localizada no centro de rotação do mecanismo, de forma que o eixo z forme o mesmo ângulo com todos os vetores u_i , $i=1,2,3$, e de forma que o vetor u_1 esteja localizado no plano X Z conforme Figura 3.

A partir da Figura 3(b), a orientação do mecanismo esférico paralelo é descrita por um vetor de ângulos $\psi = [\psi_1, \psi_2, \psi_3]^T$ para o qual a matriz de rotação é definida por duas rotações sequenciais (BAI, 2010), Eq. (1).

$$Q = \text{rot}(e_2, \psi_2) * \text{rot}(e_3, \psi_3) \quad (1)$$

Onde $e_2 = [-\sin\psi_1, \cos\psi_1, 0]^T$, $e_3 = [0, 0, 1]^T$ e $\text{rot}(e, \psi)$ é uma matriz de rotação seguindo a representação ângulo eixo. Assim, as matrizes de rotação podem ser escritas conforme Eqs.(2) e (3) (BAI, 2010).

$$rot(\mathbf{e}_2, \psi_2) = \begin{bmatrix} \cos\psi_2 - \text{sen}^2\psi_1 + (\cos\psi_2 - 1) & \cos\psi_1 + \text{sen}\psi_1 + (\cos\psi_2 - 1) & \cos\psi_1 + \text{sen}\psi_2 \\ \cos\psi_1 + \text{sen}\psi_1 + (\cos\psi_2 - 1) & \cos\psi_2 - \cos^2\psi_1 + (\cos\psi_2 - 1) & \text{sen}\psi_1 + \text{sen}\psi_2 \\ -\cos\psi_1 + \text{sen}\psi_2 & -\text{sen}\psi_1 + \text{sen}\psi_2 & \cos\psi_2 \end{bmatrix} \quad (2)$$

$$rot(\mathbf{e}_3, \psi_3) = \begin{bmatrix} \cos(\psi_3) & -\text{sen}(\psi_3) & 0 \\ \text{sen}(\psi_3) & \cos(\psi_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

No Sistema de coordenadas escolhido, o vetor unitário \mathbf{u}_i , Fig. 3(a), é dado pela Eq. (4).

$$\mathbf{u}_i = \begin{bmatrix} \cos \eta_i \text{sen } \gamma \\ \text{sen } \eta_i \text{sen } \gamma \\ -\cos \gamma \end{bmatrix} \quad (4)$$

Onde $\eta_i = 2(i-1)\pi/3$. O vetor unitário \mathbf{w}_i , $i = 1, 2, 3$ do eixo da articulação de rotação intermediária do i -ésimo segmento é obtido em termos do ângulo de entrada θ_i , $i = 1, 2, 3$ e dos parâmetros de construção do mecanismo como:

$$\mathbf{w}_i = \begin{bmatrix} s\alpha_1 * (s\eta_i * s\theta_i + c\gamma * c\eta_i * c\theta_i) + c\alpha_1 * c\eta_i * s\gamma \\ c\alpha_1 * s\gamma * s\eta_i - s\alpha_1 * (c\eta_i * s\theta_i - c\gamma * c\theta_i * s\eta_i) \\ s\alpha_1 * c\theta_i * s\gamma - c\alpha_1 * c\gamma \end{bmatrix} \quad (5)$$

Onde na Eq. (5) denota-se s para seno, e c para cosseno. O vetor unitário \mathbf{v}_i , Eq. (6), paralelo ao eixo da articulação superior de rotação do i -ésimo segmento é uma função da orientação da plataforma móvel. Sendo esta orientação descrita pela matriz de rotação Q .

$$\mathbf{v}_i = Q\mathbf{v}_i^* \quad (6)$$

Onde \mathbf{v}_i^* é o vetor unitário \mathbf{v}_i , Figura 3(a), para o eixo da articulação superior de rotação do i -ésimo segmento quando a plataforma móvel está em sua orientação de referência, que pode ser escrito conforme Eq. (7).

$$\mathbf{v}_i^* = \begin{bmatrix} \cos \eta_i \sin \gamma \\ \sin \eta_i \sin \gamma \\ \cos \gamma \end{bmatrix} \quad (7)$$

Para a cadeia fechada do manipulador esférico paralelo, Figura 3(a), a seguinte equação 8 é obtida:

$$w_i \cdot v_i = \cos \alpha_2, \quad i = 1, 2, 3 \quad (8)$$

Substituindo as Eqs. (6) e (5) na Eq. (8), obtém-se:

$$A_i t_i^2 + 2B_i t_i + C_i = 0, \quad i = 1, 2, 3 \quad (9)$$

Onde os coeficientes A_i , B_i , e C_i , são funções das coordenadas cartesianas e dos parâmetros geométricos do manipulador paralelo esférico conforme Eq. (10).

$$\begin{aligned} A_i &= \cos \eta_i \cos \alpha_1 \sin \gamma v_{i1} - \cos \gamma \cos \alpha_1 v_{i3} - \sin \gamma \sin \alpha_1 v_{i3} - \cos \gamma \cos \eta_i \sin \alpha_1 v_{i1} - \\ &\quad \cos \alpha_2 - \cos \gamma \sin \eta_i \sin \alpha_1 v_{i2} + \cos \alpha_1 \sin \gamma \sin \eta_i v_{i2} \\ B_i &= \sin \eta_i \sin \alpha_1 v_{i1} - \cos \eta_i \sin \alpha_1 v_{i2} \\ C_i &= \sin \gamma \sin \alpha_1 v_{i3} - \cos \gamma \cos \alpha_1 v_{i3} + \cos \gamma \cos \eta_i \sin \alpha_1 v_{i1} + \cos \eta_i \cos \alpha_1 \sin \gamma v_{i1} - \\ &\quad \cos \alpha_2 + \cos \gamma \sin \eta_i \sin \alpha_1 v_{i2} + \cos \alpha_1 \sin \gamma \sin \eta_i v_{i2} \end{aligned} \quad (10)$$

Sendo v_{ij} , $j = 1, 2, 3$, o j -ésimo, termo do vetor v_i . A solução para o problema da cinemática inversa do manipulador paralelo esférico é obtida pela resolução da equação de segundo grau, Eq. (10), para cada segmento do manipulador, Eq. (11).

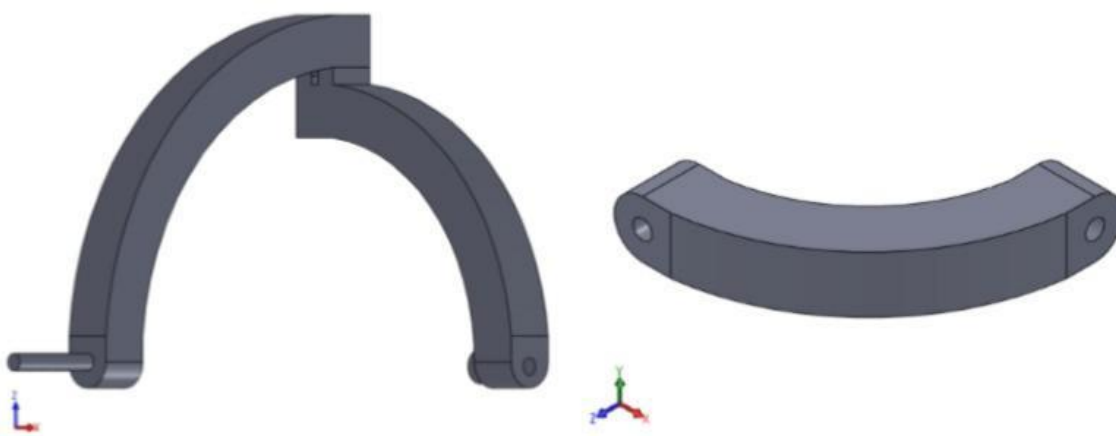
$$t_i = \frac{-B_i \pm \sqrt{B_i^2 - A_i C_i}}{A_i}, \quad i = 1, 2, 3 \quad (11)$$

Fornecendo assim dois valores para cada t_i , $i = 1, 2, 3$, conseqüentemente dois valores para cada θ_i , o que resulta em oito possíveis soluções para a orientação do mecanismo esférico paralelo.

2.3 AGILE EYE

O *Agile Eye*, escolhido como modelo ao mecanismo proposto neste trabalho, é um manipulador paralelo esférico com formato simétrico e sua configuração favorece o espaço de trabalho, com segmentos como os da Figura 4, que visam melhorar o desempenho e minimizar as interferências do mecanismo.

Figura 4 - Segmentos do mecanismo paralelo esférico.

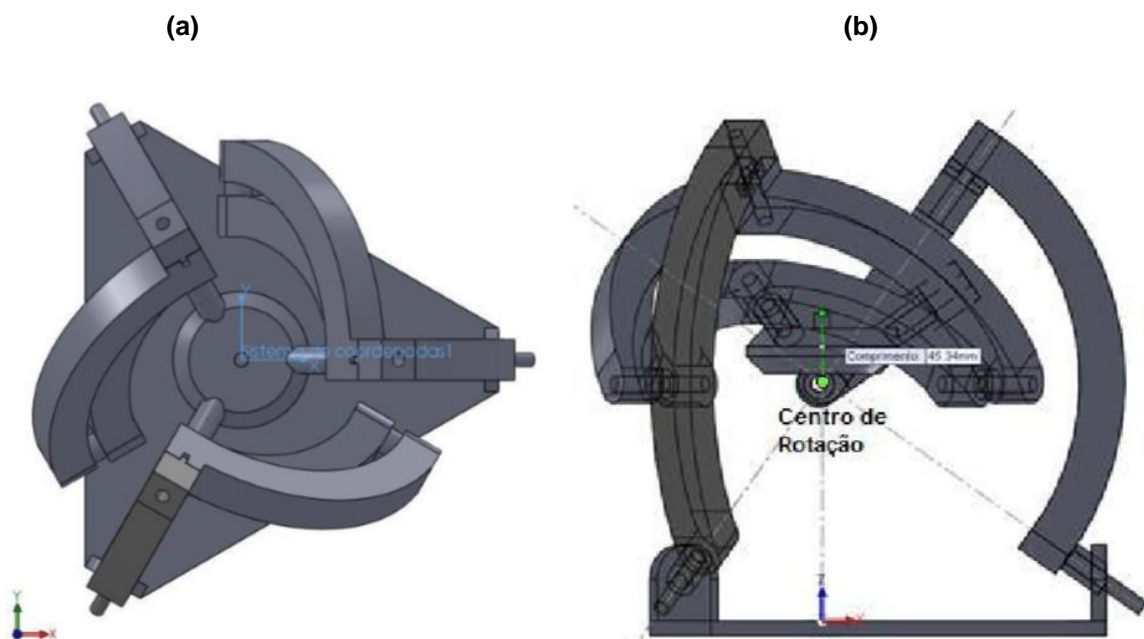


Fonte: adaptado de PÊGO (2014).

Tanto os segmentos, quanto a base e o atuador foram construídos seguindo as especificações expostas por (GOSSELIN et al. 1994), para o manipulador: o ângulo entre cada articulação da base deve ser 90° , assim como o ângulo entre a articulação da base e a articulação subsequente entre os “*links*” proximal e distal de cada segmento também deve ser 90° (Figura 5) e todos os vetores das articulações devem apontar para o centro de rotação do mecanismo, resultando assim em um mecanismo simétrico.

A estrutura paralela esférica é apresentada na Figura 5, onde se pode ver a estrutura em sua posição inicial, Figura 5(a), e o centro de rotação da estrutura, Figura 5(b) (PÊGO, 2014).

Figura 5 - Mecanismo Paralelo Esférico . (a) posição inicial; (b) centro de rotação do mecanismo



Fonte: adaptado de PÊGO (2014).

2.4 CAD (COMPUTER AIDED DESIGN)

CAD, ou em português, desenho assistido por computador, é uma tecnologia que consiste em se utilizar de computadores para realizar certas funções que se remete a desenhos e fabricação. Aliada a tecnologia CAM (*Computer Aided Manufacturing*), ou em português Fabricação Assistida por Computador, estão se tornando cada vez mais usadas nas indústrias de fabricação (GROOVER; ZIMMERS, 1984).

É importante salientar que a tecnologia CAD só se tornou 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).

2.5 CNC (CONTROLE NUMÉRICO COMPUTADORIZADO)

O Comando Numérico Computadorizado representa a automatização do processo de fabricação executado por uma máquina-ferramenta no sentido de que todas as informações, tanto geométricas quanto tecnológicas, necessárias para a fabricação de uma peça, sejam codificadas em um programa. A partir do programa o processo é inteiramente automático, sendo as informações lidas pelo comando e executadas pela máquina (VOLPATO, 2006).

Uma definição simplificada de Comando Numérico Computadorizado seria: um sistema de controle capaz de receber informações numéricas por meio de entrada própria, compilar estas informações e transmiti-las em forma de comando à máquina operatriz, de modo que esta, “sem” a intervenção do operador, realize as operações na sequência programada (VOLPATO, 2006).

2.6 A PLATAFORMA ARDUINO

Arduino é uma plataforma eletrônica de código aberto baseada em *hardware* e *software* de fácil utilização. As placas Arduino são capazes, por exemplo, de ler entradas através de um sensor ou ativar uma saída, ligar um motor, acender um LED ou qualquer outro dispositivo. Pode-se fazer isso, enviando um conjunto de instruções para o microcontrolador da placa. Para isso usa-se a Linguagem de Programação Arduino e o Arduino *Software* (IDE) (ARDUINO, 2017).

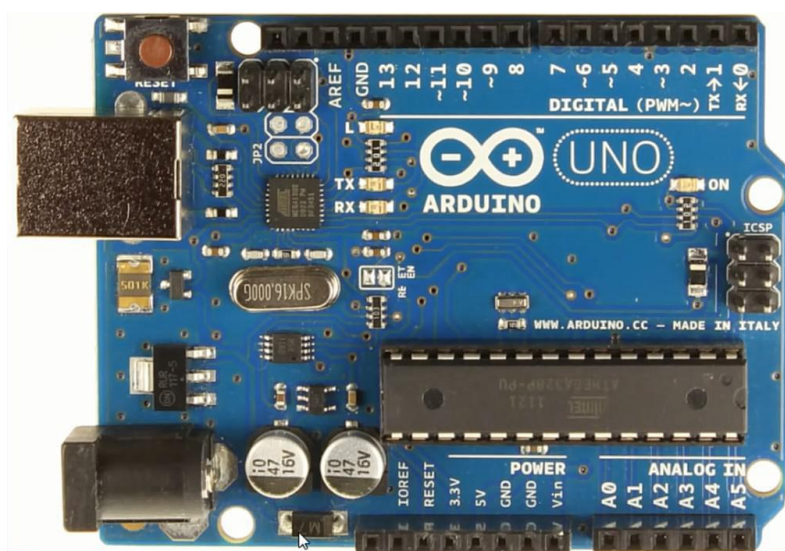
Ao longo dos anos o Arduino tem sido base para milhares de projetos, desde os mais simples até os mais complexos.

O Arduino nasceu no *Ivrea Interaction Design Institute* (ARDUINO, 2017), como uma ferramenta fácil para prototipagem rápida voltada para estudantes sem

formação em eletrônica e programação. Chegando a uma comunidade mais ampla, a placa Arduino começou a mudar para se adaptar as novas necessidades, aperfeiçoando sua placa de 8 bits para várias outras aplicações como, por exemplo, impressão 3D e sistemas embarcados complexos. Todas as placas Arduino são código aberto, capacitando os usuários a desenvolver seus projetos de forma independente. Seu *software* também é aberto, e está crescendo através das contribuições dos usuários em todo o mundo (ARDUINO, 2017).

Na Figura 6 é mostrada a placa Arduino UNO R3.

Figura 6 - Placa Arduino Uno R3



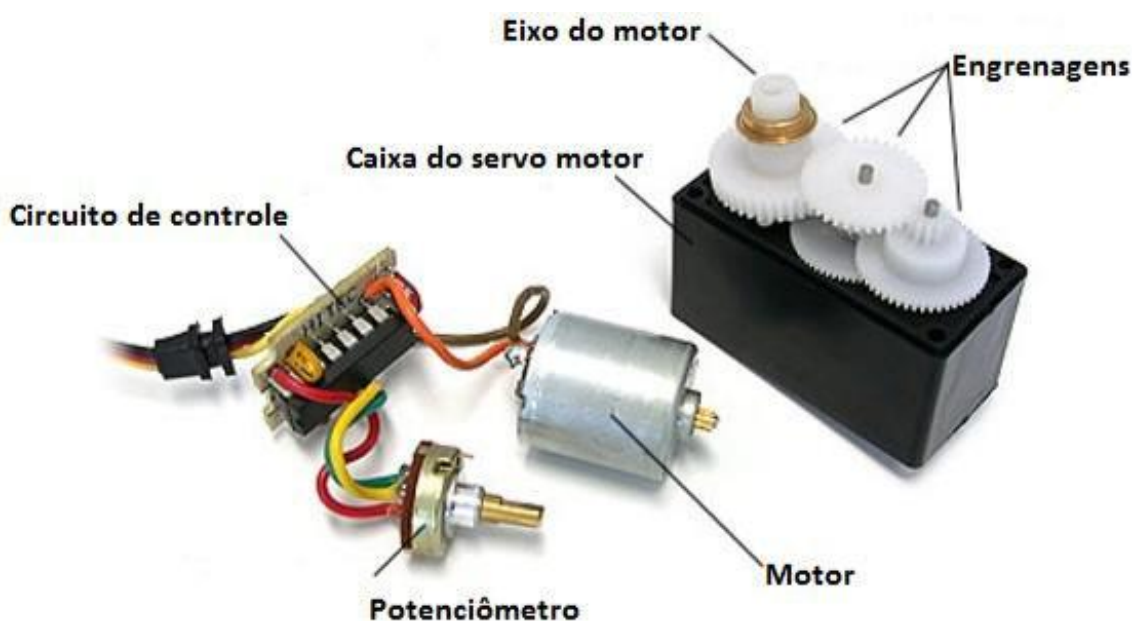
Fonte: autoria própria.

2.7 SERVO MOTORES

O servo motor é um dispositivo eletromecânico que, a partir de um sinal elétrico na sua entrada, pode ter seu eixo posicionado em uma determinada posição angular. Por permitir um posicionamento preciso do seu eixo é usado, por exemplo, em máquinas ferramentas de controle numérico (CNC); sistemas de posicionamento; robôs industriais; sistemas flexíveis de manufatura, entre outros (SANTOS, 2007).

O servo motor utilizado, possui um circuito eletrônico de controle e um potenciômetro acoplado ao eixo do motor. O circuito de controle é o responsável por receber os sinais do receptor, monitora a posição do potenciômetro e controla o motor de acordo com o sinal do receptor. O potenciômetro possibilita ao circuito de controle monitorar o ângulo em que está posicionado o eixo do motor. Se o eixo estiver no ângulo certo, o motor para. Se o circuito detecta que o ângulo está incorreto, o motor gira o eixo até a posição correta. Nas engrenagens de um servo motor existe um limitador que atua no ângulo de giro do eixo, fazendo com que este varie de 0° a 180° (SANTOS, 2007). A seguir, na Figura 7, os componentes de um servo motor.

Figura 7 - Componentes de um servo motor.



Fonte: adaptado de SANTOS (2007).

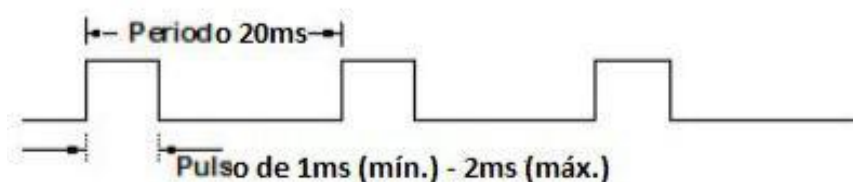
2.8 CONTROLE DO SERVOMOTOR

2.8.1 PWM (*Pulse Width Modulation*)

O servo motor é acionado quando recebe um sinal PWM (*Pulse Width Modulation*), em português: Modulação por Largura de Pulso. Nele têm-se o controle sobre a corrente eficaz aplicada no servo motor.

Os servomotores possuem um circuito de controle que monitora o sinal de entrada a cada 20ms e, se dentro deste intervalo ele recebe uma alteração do sinal de 0V para 5V com uma duração entre 1ms e 2ms, envia um comando para o motor, de maneira a obedecer a ordem do sinal (SANTOS, 2007). A seguir, na Figura 8, o gráfico do sinal monitorado pelo circuito de controle do servo motor.

Figura 8 - Sinais monitorados pelo circuito de controle

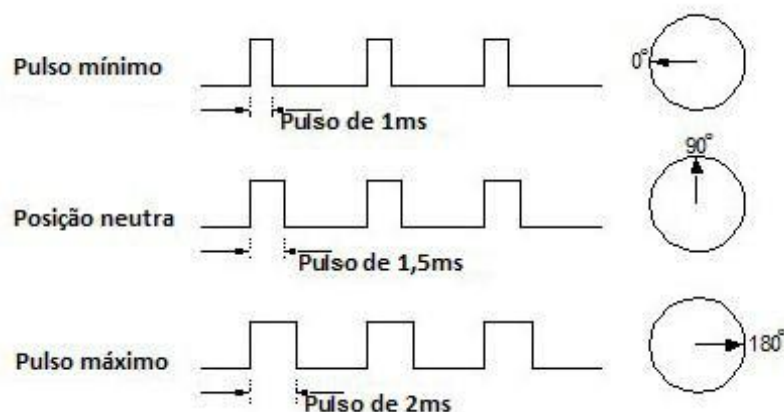


Fonte: adaptado de SANTOS (2007).

Um sinal de 1ms corresponde a uma posição do braço do servo todo à esquerda ou 0°. Um sinal de 1,5 ms é o que chamamos de posição central do servo ou 90°.

Um sinal de 2 ms corresponde a uma posição do braço do servo todo a direita ou 180°. A seguir, na Figura 9, os gráficos mostrando os sinais PWM para o controle de um servo motor.

Figura 9 - Sinais PWM para o controle de um Servo Motor



Fonte: adaptado de SANTOS (2007).

Uma vez que o servo motor recebe um sinal de 1,5ms, por exemplo, ele verifica se o potenciômetro encontra-se na posição correspondente (90°).

Caso se encontre na posição desejada, não acontecerá nada. Se o potenciômetro não estiver na posição correspondente ao sinal recebido, o circuito de controle aciona o motor até o potenciômetro estar na posição certa (SANTOS, 2007).

A direção de rotação do servo motor vai depender também da posição do potenciômetro, sendo que se ele estiver entre 0° e 90° o motor vai girar em uma direção, e entre 90° e 180°, irá à direção contrária (SANTOS, 2007).

Outro detalhe em relação à velocidade: quando um sinal PWM é enviado ao servo motor, quanto mais “afastado” ele estiver da posição de referência (90°), maior será a sua velocidade de rotação. Por exemplo: se for enviado o sinal PWM correspondente a 80°, a velocidade de rotação em direção à posição referente 0°, será muito menor que a de rotação quando o sinal enviado corresponde a 10°.

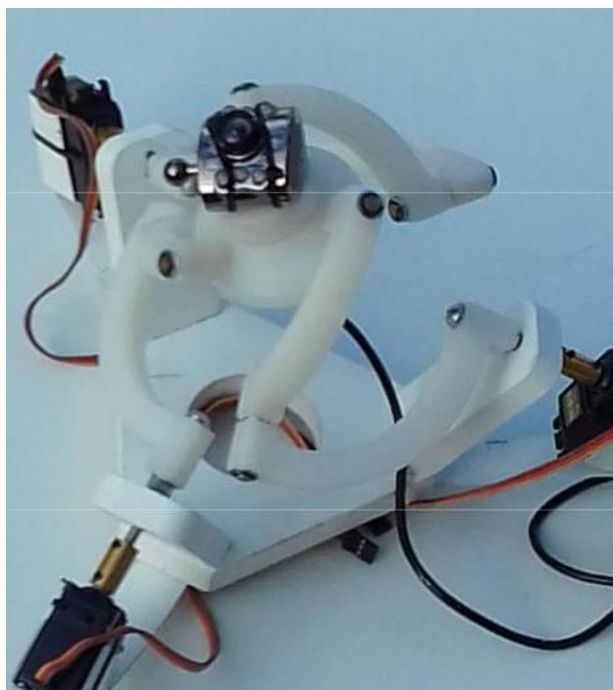
Da mesma maneira, o sinal referente a 100° resulta em uma velocidade muito menor que a resultante de um sinal de 170°, em direção à posição de 180° (SANTOS, 2007).

3 DESENVOLVIMENTO

3.1 PARTES MECÂNICAS

As dimensões do Mecanismo Paralelo Esférico foram determinadas, considerando o tamanho (suficiente para demonstração e portátil) e desempenho (quanto menor a massa e tamanho, menores serão os momentos de inércia, permitindo velocidade e aceleração mais altas, dado o torque do servo motor disponível). Foram levadas em conta também as dimensões exigidas pelo apontador (centro), onde é fixada a câmera. A câmera escolhida foi a Quantum QHM500LM, por ter forma, tamanho e massa que permitem a sua fixação no apontador. Para a fixação da câmera, ela foi desmontada do seu suporte e presa no apontador com duas abraçadeiras de nylon, como mostra a Figura 10.

Figura 10 - Câmera fixada no apontador



Fonte: autoria própria.

Foram feitos no Solid Works 2016, os desenhos das partes mecânicas dos segmentos dos atuadores, da base, do apontador e a montagem (APÊNCICES A, B, C, D, E e F), seguindo as especificações expostas por (GOSSELIN et al. 1994). Também foi feita a verificação de possíveis colisões, interferências e a resistência das peças em relação ao material usado na sua fabricação.

O material inicialmente usado na fabricação da base e dos atuadores foi o compensado MDF (*Medium Density Fiberboard*), por ter uma resistência que parecia ser adequada, de fácil usinagem e custo menor em relação a outros materiais. As peças foram usinadas em uma fresadora CNC, devido a precisão dimensional exigida.

Na construção original do *Agile Eye*, as articulações foram feitas com rolamentos de esferas. Isso implicaria em uma complexidade maior na fabricação dos segmentos dos atuadores. Por se tratar de um protótipo para demonstração, optou-se então por um tipo de articulação mais simples, mas que também deveria satisfazer as exigências de baixo atrito na movimentação dos atuadores.

As articulações foram fabricadas com pinos usinados em latão, fixados com anéis trava (anel elástico). O apontador foi usinado em nylon.

O servo motor usado no projeto, mostrado na Figura 11, foi o Tower Pro SG-5010. É um motor com velocidade controlada e alto torque, exigidos no projeto.

Figura 11 - Servomotor Tower Pro SG-5010.

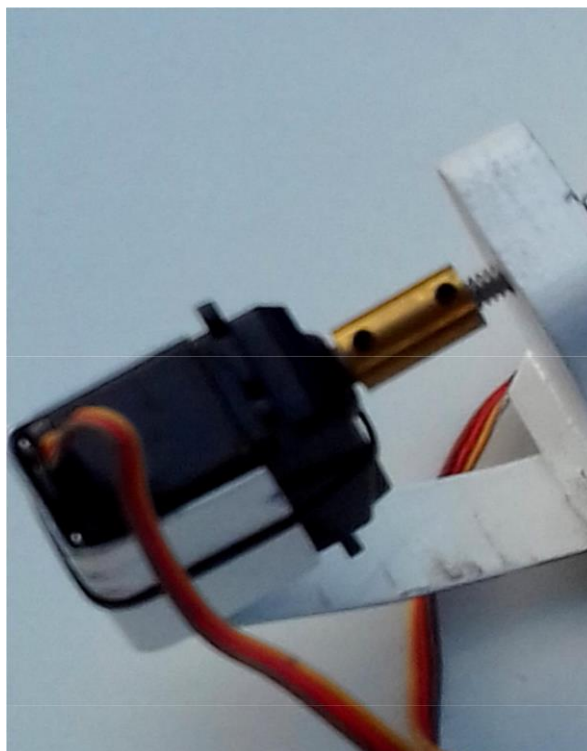


Fonte: adaptado de ptrobotics.com (2017).

Os servomotores foram alinhados e fixados com parafusos e porcas. Seus suportes foram cortados e dobrados em chapa de aço com espessura de 1 mm.

Os eixos foram conectados aos dos atuadores com acoplamentos usinados em latão e fixados com parafusos Allen M3 sem cabeça, como mostra a Figura 12.

Figura 12 - Acoplamento conectando os eixos do servomotor e do atuador.



Fonte: autoria própria.

3.2 PARTE ELETRÔNICA

O controle dos atuadores do mecanismo paralelo esférico, foi feito usando uma placa Arduino Uno R3. Ela realiza o processamento que transforma o movimento das alavancas de dois *joysticks* (um para as posições x, y e outro para a posição z) em rotação dos servomotores. O Arduino é um *hardware* livre, que segue os mesmos conceitos dos *softwares* livres, facilitando e barateando a montagem.

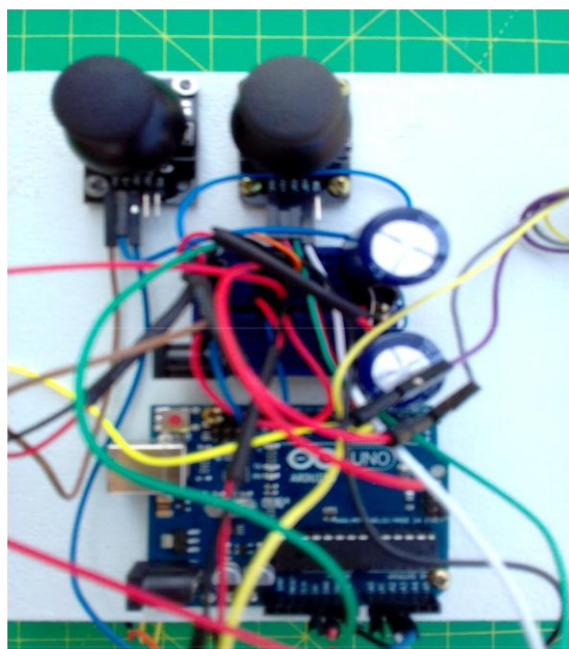
O Arduino Uno R3 foi programado usando a plataforma de *software* Arduino IDE (*download* no *website* oficial Arduino). Os programas para o Arduino são chamados de “*sketch*” (rascunho, esboço) e são escritos em Linguagem C e C++. O Arduino IDE vem com uma biblioteca composta por um conjunto de funcionalidades que facilitam o trabalho de programação (ARDUINO, 2017). Genericamente deve-se definir o que os métodos “*setup*” e “*loop*” vão fazer.

O método “*setup*” é executado uma vez, assim que o Arduino é ligado, e normalmente usado para definir configurações iniciais como, por exemplo, quais serão os pinos de entrada ou saída.

O método “*loop*” é executado após o “*setup*”. Como o próprio nome já sugere ele é executado em “*loop*” enquanto o Arduino estiver ligado. Dentro do método “*loop*” é que normalmente se escreve as funcionalidades do programa, como por exemplo: fazer um LED piscar de dois em dois segundos. (ARDUINO, 2017).

Na alimentação do circuito de controle (APÊNDICE G) e dos três servomotores, foi usada uma fonte chaveada de 5V/1A. Na Figura 13, a seguir, uma vista do circuito de controle dos servo motores.

Figura 13 - Vista do circuito de controle dos servomotores



Fonte: autoria própria.

3.2.1 Descrição da placa do Arduino UNO R3 utilizada.

A placa Arduino Uno R3 é baseada no microcontrolador ATmega328. Ela tem 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP (*In Circuit Serial Programming*) e um botão de *reset* (ARDUINO, 2017).

Dados técnicos:

Microcontrolador:	ATmega328
Tensão de operação:	5V
Tensão de alimentação (recomendada):	7-12V
Tensão de alimentação (limite):	6-20V
Entradas e saídas digitais:	14 das quais 6 podem ser PWM (<i>Pulse Width Modulation</i>)
Entradas analógicas:	6
Corrente contínua por pino de I/O:	40 mA
Corrente contínua para o pino 3.3V:	50 mA
Memória <i>Flash</i> :	32 kB (ATmega328) dos quais 0.5 kB são usados pelo <i>bootloader</i>
Memória RAM:	2 kB (ATmega328)
EPR0M:	1 kB (ATmega328)
Velocidade do <i>Clock</i> :	16 MHz
Dimensões:	68,58 mm x 53,34 mm

Alimentação da placa

O Arduino Uno R3 pode ser alimentado pela conexão USB ou com uma fonte de alimentação externa. A alimentação é selecionada automaticamente.

A alimentação externa (não USB) pode ser tanto de um adaptador CA para CC ou bateria. Cabos vindos de uma bateria podem ser inseridos diretamente nos pinos GND e VIN do conector de alimentação.

Pode funcionar com uma fonte de alimentação externa de 6 a 20 volts.

No entanto, se a alimentação for inferior a 7V, o pino 5V pode fornecer uma tensão menor e a placa se mostrar instável. Se a alimentação for maior do que 12V o regulador de tensão pode superaquecer e danificar a placa. A faixa recomendada é de 7 a 12 volts.

Os pinos de alimentação são os seguintes:

VIN- Entrada de alimentação para a placa Arduino quando se está utilizando uma fonte de alimentação externa.

5V - A fonte de alimentação regulada usada para o microcontrolador e para outros componentes na placa, pode vir tanto do VIN através do regulador embarcado, da conexão USB ou outra fonte regulada em 5V.

3V3 - Na fonte de 3,3V gerada pelo regulador embarcado. A corrente máxima suportada é de 50mA.

GND - Pinos terra.

Entrada e Saída

Cada um dos 14 pinos digitais do Arduino Uno R3 podem ser utilizados como entrada ou saída utilizando-se as funções *pinMode*, *digitalWrite* e *digitalRead*. Eles operam à 5V. Cada pino pode fornecer ou receber uma corrente máxima de 40 mA e tem um resistor *pull-up* interno (desconectado por padrão) de 20 – 50 kΩ.

Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do chip serial USB para TL ATmega8U2.

Interruptores Externos: 2 e 3. Estes pinos podem ser configurados para disparar uma interrupção de acordo com alguma variação sensível pelo circuito.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos dão suporte à comunicação SPI utilizando a biblioteca SPI.

LED: 13. LED integrado ao pino digital 13. Quando este pino está no valor HIGH, o LED está aceso, quando está em LOW, apagado.

I2C: 4 (SDA) e 5 (SCL). Fornecem suporte a comunicação I2C (TWI) utilizando a biblioteca *Wire*.

PWM - As portas digitais 3, 5, 6, 10 e 11 fazem o controle PWM. São as que têm um caractere ~ ao lado do número,

AREF - Tensão de referência para as entradas analógicas. Utilizado com a função *AnalogReference*.

Reset - Envia o valor *LOW* para esta linha, resetando o microcontrolador.

O Arduino Uno R3 possui uma série de facilidades para se comunicar com um computador, outro Arduino, ou outros microcontroladores.

O microcontrolador ATmega328 fornece comunicação serial UART TTL (5V) que está disponível nos pinos digitais 0 (RX) e 1 (TX).

O software do Arduino inclui um monitor serial que permite dados textuais serem enviados e recebidos da placa. LEDs conectados ao RX e TX piscarão enquanto dados estiverem sido transmitidos pelo chip USB-para-serial e pela conexão USB (não para comunicação serial nos pinos 0 e 1).

Proteção contra sobre corrente na USB

O Arduino Uno R3 possui um *polyfuse* resetável que protege a porta USB do computador contra sobrecorrente e curtos-circuitos. Se mais de 500mA forem aplicados a porta USB ele automaticamente interrompe a conexão até que o curto-circuito ou a sobrecarga seja removido.

Dimensões

O comprimento e a largura máximos da placa do Arduino Uno R3 são de 68,58 mm e 53,34 mm, respectivamente. Os conectores USB e de alimentação se estendem além destas dimensões. Quatro orifícios para parafusos permitem que a placa seja fixada a uma superfície ou encapsulamento.

4 TESTES E RESULTADOS

Durante os primeiros testes de funcionamento, apareceram trincas nas extremidades dos segmentos internos dos atuadores. Também foi observado que se os atuadores, com três segmentos, forem montados com dois (desenho do APÊNDICE A), se tem uma melhor rigidez e maior amplitude dos movimentos.

Optou-se, então, por esta solução que se mostrou melhor. Os atuadores foram refeitos, tendo sido escolhido então o polipropileno que se mostrou apropriado para suportar as forças de torção.

4.1 DESEMPENHO DO MECANISMO PARALELO ESFÉRICO

4.1.1 Velocidade angular

A velocidade angular, diretamente proporcional à massa, tamanho dos segmentos dos atuadores e torque dos servomotores 0,54 N.m à tensão de 5V, está satisfatória para efeito de demonstração.

A velocidade angular especificada para o servo motor Tower Pro SG-5010, é de 0,2s/60° à tensão de 4,8V.

4.1.2 Alcance rotacional

A partir da posição inicial de referência (90° nos servomotores), todas as posições angulares necessárias, de 60° à 140°, podem ser alcançadas por cada atuador. Foi observado que o mecanismo exibe uma singularidade nas posições de fim de curso dos atuadores. Os atuadores vão ao limite do alcance rotacional, mas continuam suportando a força de torque do servomotor que procura ir até as posições 0° ou 180°. Para escapar dessa situação de trajetória e diminuir a “faixa de

impacto”, foi implementado no “*sketch*” este limite angular para evitar a aproximação dessas posições.

4.1.3 Precisão de posicionamento

Para este teste, foi usado um apontador *laser* encaixado no centro do mecanismo paralelo esférico, que foi repetidamente acionado (um atuador de cada vez) para apontar na mesma posição de destino em uma parede, sendo percebida uma pequena diferença. Isto parece ser devido a pequenas imprecisões no posicionamento e usinagem dos furos das articulações.

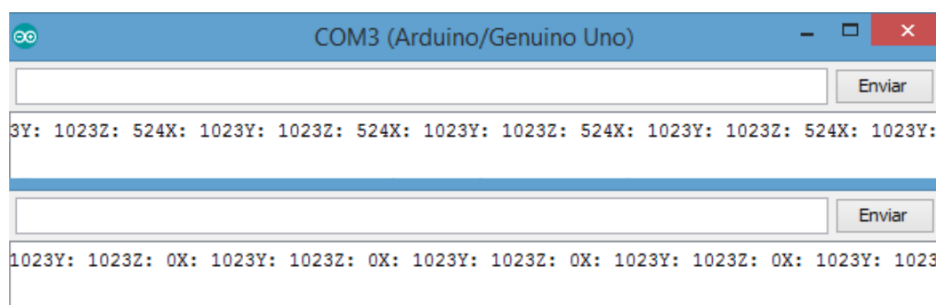
4.1.4 Servomotor TOWER PRO SG-5010

Durante os testes de funcionamento do mecanismo paralelo esférico, foi medida a solicitação de corrente do servomotor, que ficou variando em mais ou menos 250mA, saindo da posição inicial e 50mA durante a operação. Demonstrando assim que a fonte de alimentação escolhida (5V/1A) satisfaz as exigências de consumo.

4.2 FUNCIONAMENTO DO SOFTWARE

A captura de imagem da tela do monitor serial do Arduino *Software* (IDE) com a rotina rodando, na Figura 14, mostra a variação angular dos eixos dos servo motores, conforme a movimentação dos *joysticks*. Os números 1023 e 0 correspondem, respectivamente, as posições 180° e 0° no eixo dos servomotores. Entre 1023 e 0, correspondem aos ângulos intermediários.

Figura 14 - Imagem da tela do monitor serial do Arduino Software (IDE)



Fonte: autoria própria.

Observa-se que, nas posições angulares intermediárias dos eixos dos servomotores, o monitor serial mostra variações nos números mostrados, mesmo com o *joystick* parado. Isto se deve a ruídos (interferências). Preciseria ser feito um cálculo matemático e implementado no “*sketch*” para corrigir este problema.

5 CONCLUSÃO

Neste trabalho foi apresentado um mecanismo paralelo esférico para orientação de câmeras em um sistema de monitoramento de imagens. Com bom desempenho em termos de velocidade, aceleração e mudanças de direção.

Foi desenvolvido e implementado um protótipo com três graus de liberdade, que pode ser direcionado para qualquer lado em um ângulo de até 140° e um sistema de controle para ser utilizado com câmeras CCD (*charge-coupled device*) no monitoramento de imagens.

O desempenho do mecanismo paralelo esférico é caracterizado pelo fato de que as extremidades dos atuadores estão ligadas à base através de múltiplas cadeias cinemáticas. Isto leva a uma alta rigidez e propriedades dinâmicas muito boas com a inércia das partes móveis consideravelmente reduzida.

A modelagem das peças e a montagem, foram feitas no programa Solid Works 2016 (*Dassault Systemes SolidWorks Corporation*), sendo possível observar previamente a possibilidade de colisões, interferências e testar a sua resistência em relação ao material escolhido para a sua fabricação.

O material inicialmente usado na fabricação da base e dos atuadores foi o compensado MDF (*Medium Density Fiberboard*), por ter uma resistência que parecia ser adequada. Durante os testes de funcionamento apareceram trincas na extremidade dos segmentos internos dos atuadores, que foram refeitos. Foi escolhido então o polipropileno que se mostrou apropriado para suportar as forças de torção.

As peças foram usinadas em uma fresadora CNC, com precisão dimensional satisfatória.

Para o desenvolvimento do *firmware* foi escolhida a plataforma Arduino para atuar no controle dos servo motores do protótipo.

O Arduino Uno R3 foi programado usando a plataforma de *software* Arduino IDE (*download no website oficial Arduino*).

Foi possível integrar as etapas do projeto e desenvolver o protótipo, demonstrando o funcionamento, desempenho e aplicabilidade de seu mecanismo.

5.1 SUGESTÕES PARA TRABALHOS POSTERIORES

O manejo do mecanismo paralelo esférico, feito através de dois *joysticks* (um para as posições x, y e outro para a posição z), requer habilidade. Um pequeno deslocamento da alavanca provoca proporcionalmente um grande movimento do atuador. Isso pode ser melhorado com a elaboração e implementação de um *software* para controle de movimentação dos atuadores mais efetivo e preciso.

A precisão do sistema não foi atendida, devido às pequenas diferenças de posição e usinagem nos furos das articulações dos atuadores e suportes dos eixos. Isso pode ser melhorado se forem usinados com maior precisão e as articulações montadas com rolamentos.

Projetar e construir as peças explorando as vantagens da impressão 3D de alta resolução. Possibilitando assim, peças com melhor precisão dimensional.

Pesquisar a implementação de um material mais leve e com resistência mecânica suficiente, para reduzir ao máximo os momentos de inércia, otimizando as propriedades dinâmicas do sistema.

Como aplicação pode ser usado, por exemplo, no rastreamento de objetos baseado em visão ou manipulação de laser.

REFERÊNCIAS

AGILE EYE. **Versão original do Mecanismo Paralelo Esférico**. Disponível em: < <https://robot.gmc.ulaval.ca/en/research/research-thrusts/parallel-mechanisms/the-agile-eye/>>. Acesso em : 15 jun. 2017.

ARDUINO. **Site oficial**. Disponível em:< <https://www.arduino.cc/> >. Acesso em: 15 jun. 2017.

Bai, Anim S. **Optimum design of spherical parallel manipulators for a prescribed workspace**, *Mechanism and Machine Theory*. 45 p. 200–211, 2010. Disponível em: < <http://homes.m-tech.aau.dk/shb/papers/mmt10-wsoptimization.pdf> >. Acesso em: 15 jun. 2017.

GROOVER, MIKELL e ZIMMERS, Emory W. Jr. **CAD/CAM Computer-Aided design and Manufacturing**. Department of Industrial Engineering Lehigh University, Bethlehem, Pennsylvania, 1984. Disponível em: <<https://dl.acm.org/citation.cfm?id=548893>>. Acesso em: 15 jun. 2017.

Gosselin, C.M., Hamel, J.-F. **The agile eye: a high-performance three-degree-of-freedom camera-orienting device**. IEEE International Conference on Robotics and Automation, 1994. Disponível em: <<http://ieeexplore.ieee.org/document/351393/>>. Acesso em: 15 jun. 2017.

Li, T., Payandeh, S. **Design of spherical parallel mechanisms for application to laparoscopic surgery**. *Robotica* 20, p.133–138, 2002. Disponível em <<https://link.springer.com/article/10.1007/s11465-012-0325-4>>. Acesso em: 5 jun. 2017.

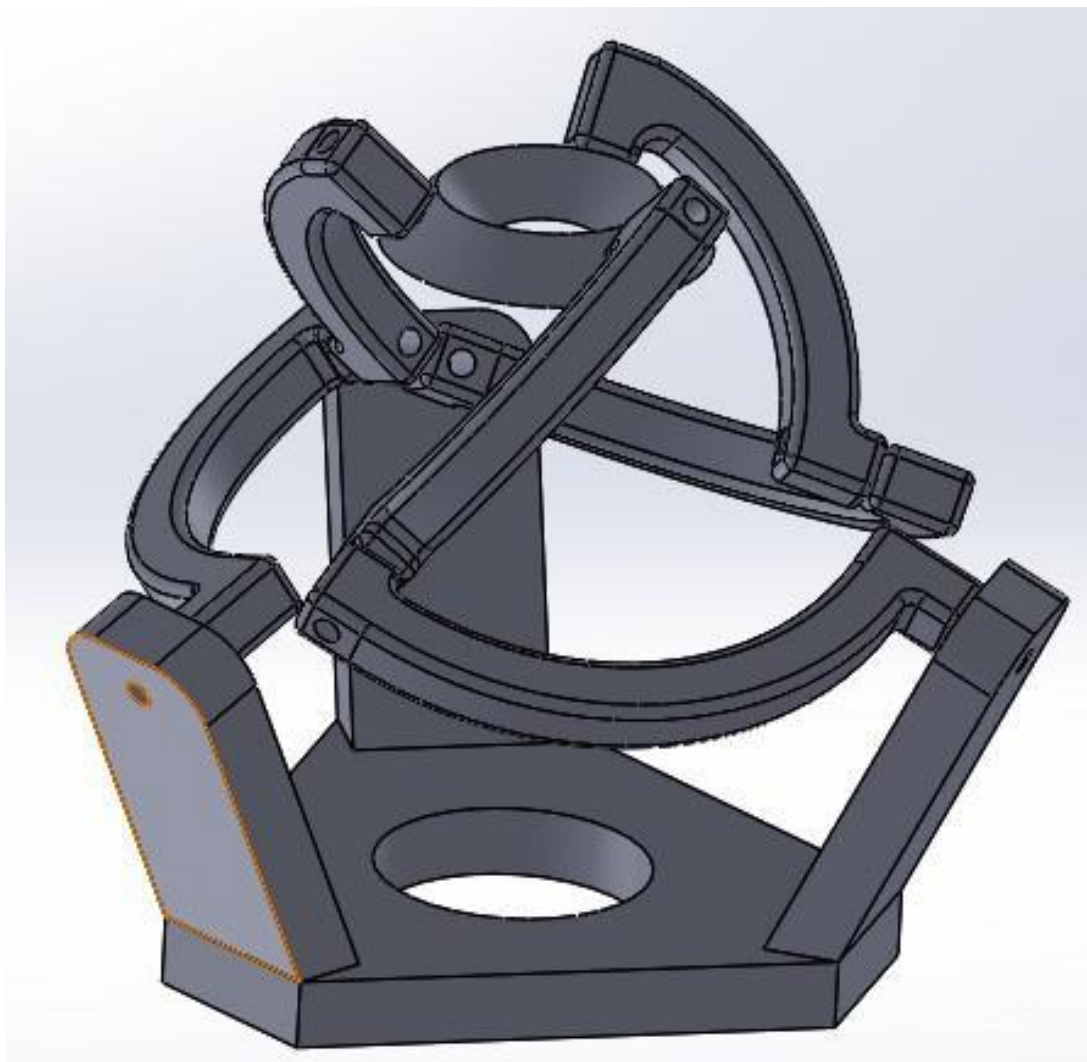
PÊGO, João Marcus Teixeira. **Estudos de uma nova estrutura robótica paralela combinada**. 2014. Disponível em: <<http://www.proceedings.blucher.com.br/article-details/estudo-de-uma-nova-estrutura-robótica-paralela-combinada-11897>>. Acesso em: 15 jun. 2017.

VOLPATO, Neri. **Introdução à Tecnologia CNC e à Programação Manual de Torno e Fresadora**. Universidade Tecnológica Federal do Paraná, Departamento Acadêmico de Mecânica, Núcleo de Prototipagem e Ferramental, 2006.

SANTOS, André. **Servomotores**, 2007. Disponível em:
<<http://www.pictronics.com.br/downloads/apostilas/servomotores.pdf>>. Acesso
em: 15 jun. 2017.

APÊNDICE A - Manipulador Paralelo Esférico montado com atuadores de dois segmentos.

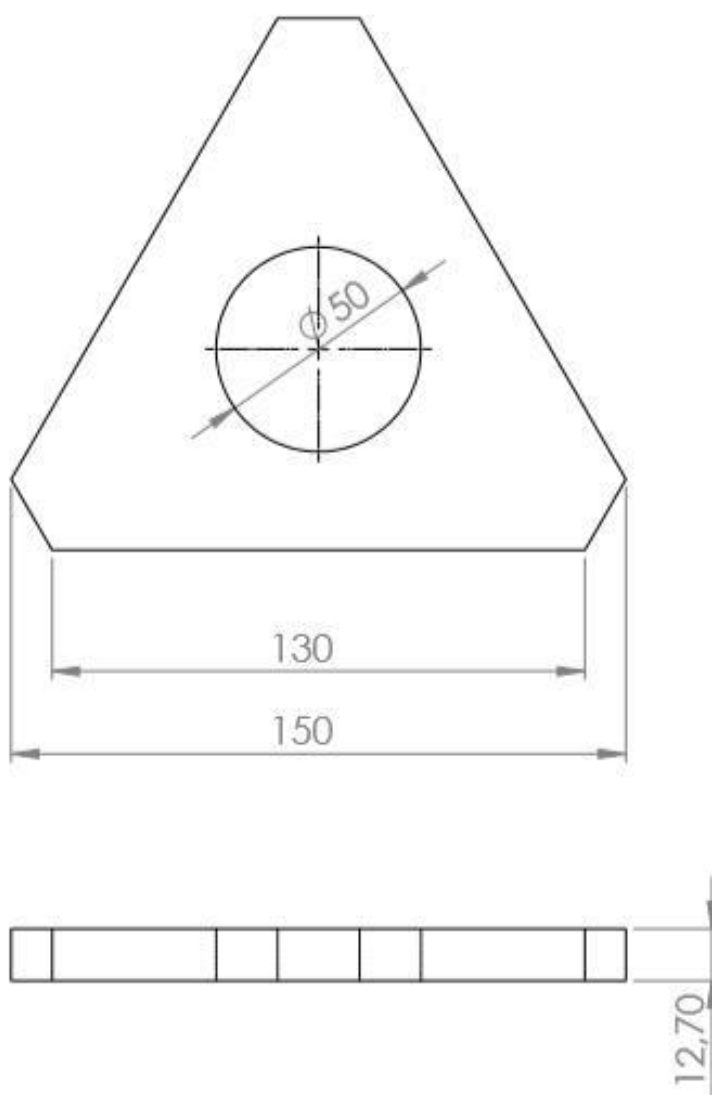
Figura 15 - Manipulador Paralelo Esférico montado com atuadores de dois segmentos



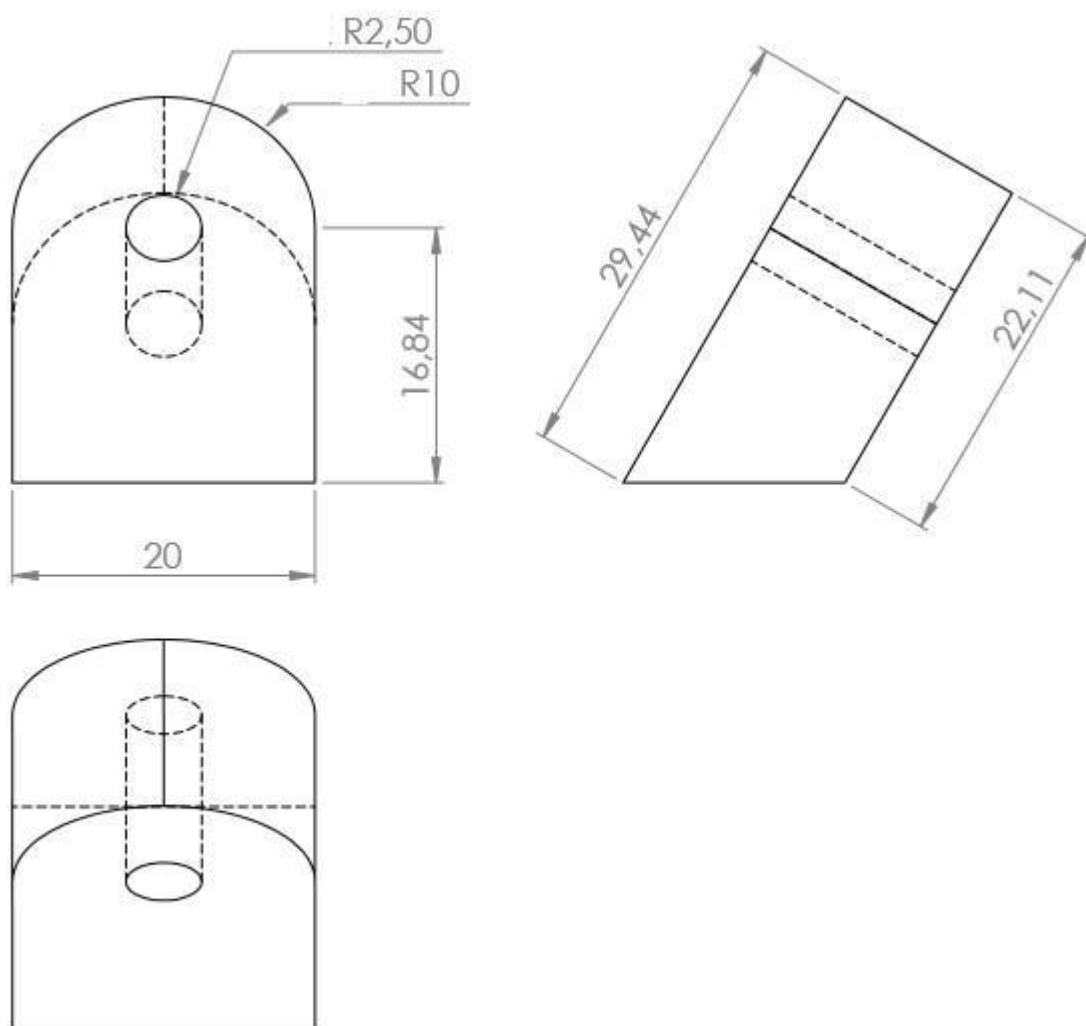
Fonte: autoria própria.

APÊNDICE B – Base

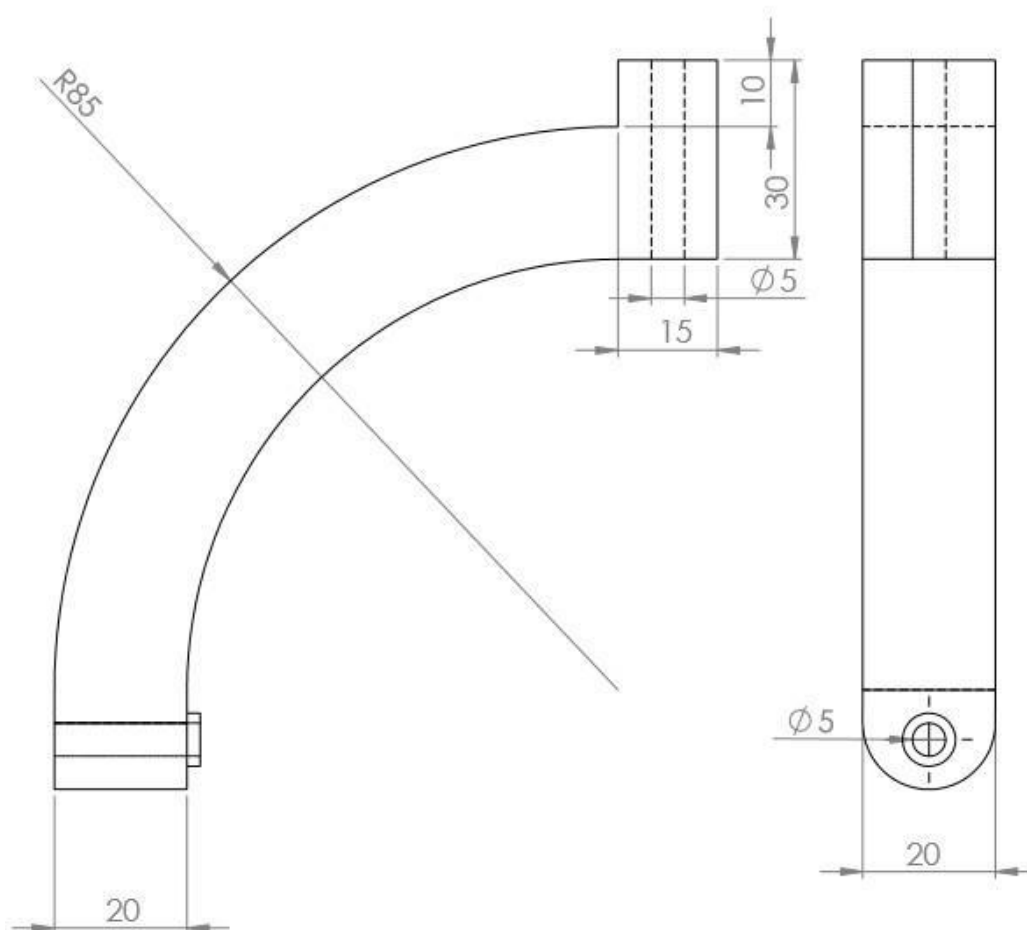
Figura 16 - Base



Fonte: autoria própria.

APÊNDICE C – Suporte dos eixos**Figura 17- Suporte dos eixos**

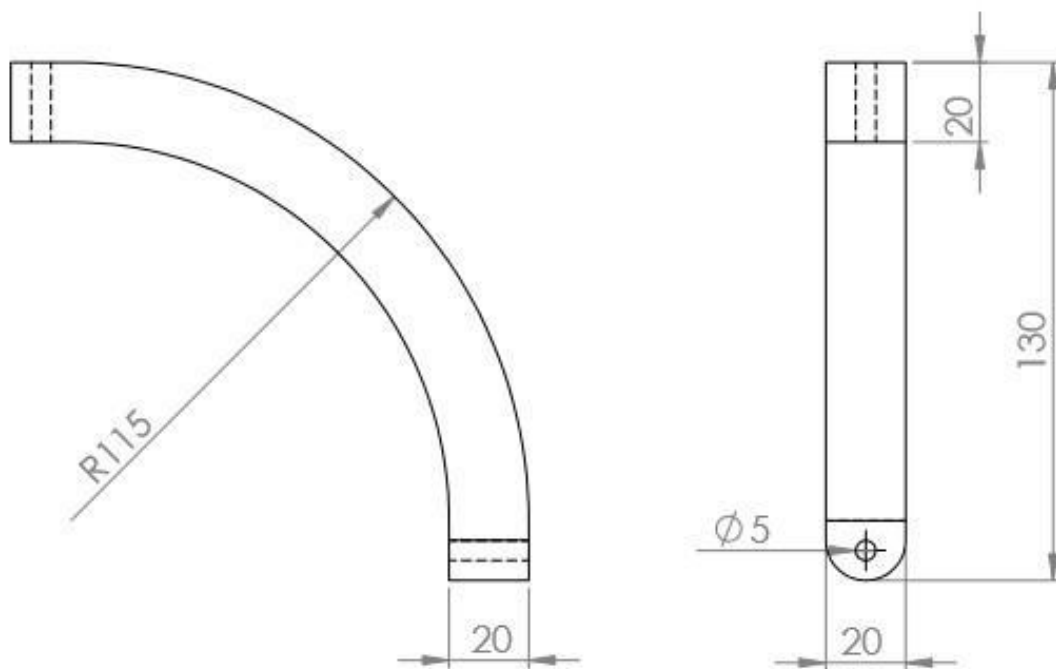
Fonte: autoria própria.

APÊNDICE D – Segmento externo do atuador**Figura 18 - Segmento externo do atuador**

Fonte: autoria própria.

APÊNDICE E – Segmento interno do atuador

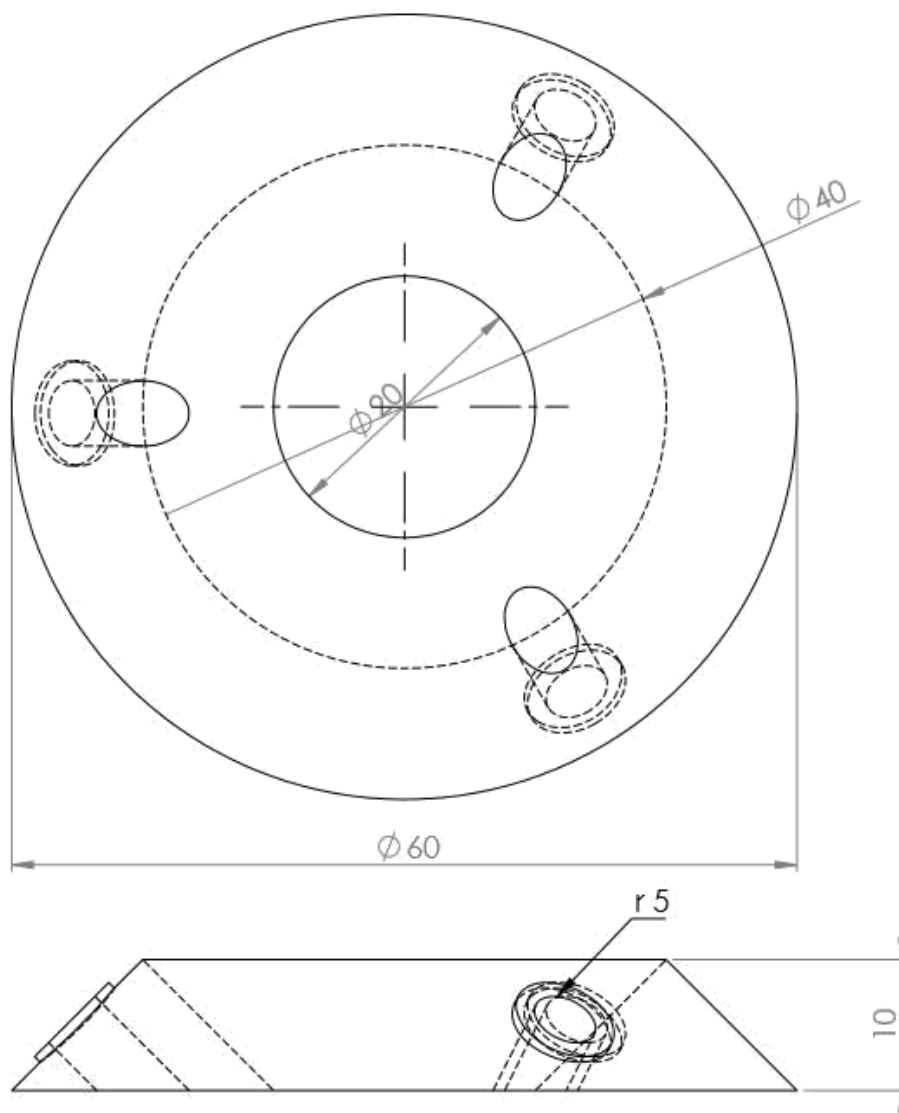
Figura 19 - Segmento interno do atuador



Fonte: autoria própria.

APÊNDICE F- Apontador

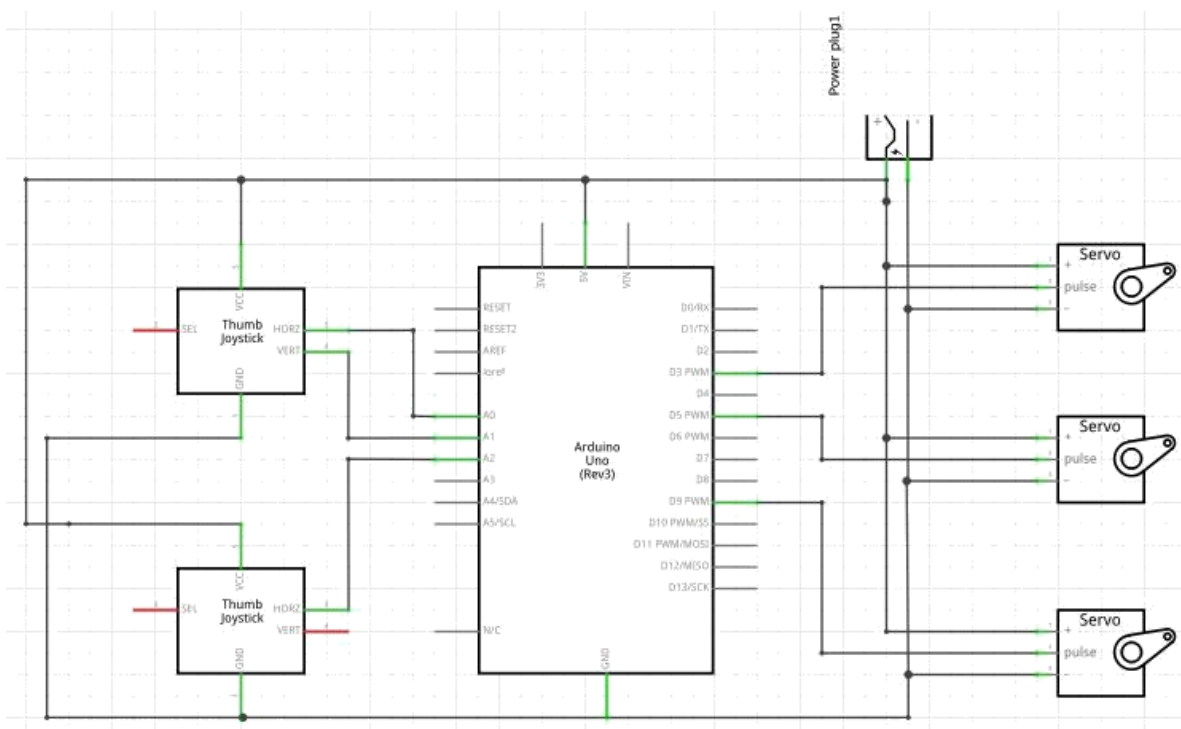
Figura 20 - Apontador



Fonte: autoria própria.

APÊNDICE G – Esquemático do circuito de controle dos servomotores

Figura 21 - Esquemático do circuito de controle dos servos motores



Fonte: autoria própria.

APÊNDICE H – Programa do controle dos servomotores

```

#include <Servo.h>

Servo servoX, servoY, servoZ; //objetos referentes a classe servo

#define pinoServoX 3 //pinos PWM
#define pinoServoY 5
#define pinoServoZ 8

#define pinoJoystickX A0 // pinos que lerão a posição dos eixos do joystick
#define pinoJoystickY A1
#define pinoJoystickZ A2

int joystickX=0;
int joystickY=0;
int joystickZ=0;

void setup() {
  servoX.attach(pinoServoX); // associa os objetos a classe
  servoY.attach(pinoServoY);
  servoZ.attach(pinoServoZ);

  servoX.write(90); // coloca os servos na posição central
  servoY.write(90);
  servoZ.write(90);
  Serial.begin(9600);
}

Serial.print(joystickX);
Serial.print(" - Y: ");
Serial.print(joystickY);
Serial.print(" - Z: ");
Serial.print(joystickZ);

//mapeando o valor lido no joystick para a escala entre 0 e 180
joystickX = map(joystickX, 0, 123, 0, 180);
joystickY = map(joystickY, 0, 123, 0, 180);
joystickZ = map(joystickZ, 0, 123, 0, 180);

servoX.write(joystickX); // posiciona eixos de acordo valores lidos no joystick
servoY.write(joystickY);
servoZ.write(joystickZ);

// espera 20ms, eixos se posicionarem
delay(20);
}

```