

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

JOÃO VITOR CAMPANER ZAGO

**ROBÔ PARALELO ATUADO POR CABOS: UM ESTUDO DE CASO DE UM ROBÔ
COM 3 GRAUS DE LIBERDADE**

São Carlos
2020

JOÃO VITOR CAMPANER ZAGO

ROBÔ PARALELO ATUADO POR CABOS: UM ESTUDO DE CASO DE UM ROBÔ
COM 3 GRAUS DE LIBERDADE

Monografia apresentada ao Curso de Engenharia Mecatrônica da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientadora: Prof^a. Dr^a. Maira Martins da Silva

São Carlos

2020

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

Z18r Zago, João Vitor Campaner
 Robô paralelo atuado por cabos: um estudo de caso
 de um robô com 3 graus de liberdade / João Vitor
 Campaner Zago; orientadora Maira Martins da Silva. São
 Carlos, 2020.

Monografia (Graduação em Engenharia Mecatrônica)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2020.

1. Robôs paralelos atuados por cabos. 2. Modelagem
matemática. 3. Prototipagem. I. Título.

FOLHA DE AVALIAÇÃO

Candidato: João Vitor Campaner Zago

Título: ROBÔ PARALELO ATUADO POR CABOS: UM ESTUDO DE CASO DE
UM ROBÔ COM 3 GRAUS DE LIBERDADE

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos da
Universidade de São Paulo
Curso de Engenharia Mecatrônica.

BANCA EXAMINADORA

Profª Dr.^a Maira Martins da Silva (Orientadora)

Nota atribuída: 9,0 (NOVE, ZERO)

Maria M. de Silve

(assinatura)

Prof. Dr. Mário Luiz Tronco

Nota atribuída: 9,0 (NOVE, ZERO)

Maria M. de Silve

(assinatura)

Prof. Dr. João Vitor de Carvalho Fontes

Nota atribuída: 9,0 (NOVE, ZERO)

Maria M. de Silve

(assinatura)

Média: 9,0 (NOVE, ZERO)

Resultado: APROVADO

Data: 08/12/2020.

Este trabalho tem condições de ser hospedado no Portal Digital da Biblioteca da EESC

SIM NÃO Visto do orientador Maria M. de Silve

AGRADECIMENTOS

À Carolina, minha companheira, que esteve ao meu lado durante todos os dias desse projeto, me acompanhando, auxiliando, aconselhando e que sabe melhor do que ninguém o que significa para mim.

Aos meus pais, Auro e Rosangela, que sempre zelaram pela minha educação, e garantiram que eu conseguisse minha formação da melhor maneira possível.

À Prof.^a Dr.^a Maira, excelente professora e melhor pessoa ainda, que me orientou ao longo do trabalho, com clareza, atenção e pontualidade, mesmo com a distância e todos os empecilhos que o período apresentou.

RESUMO

ZAGO, J. V. C. Robô paralelo atuado por cabos: um estudo de caso de um robô com 3 graus de liberdade. 2020. Trabalho de conclusão de curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020

Robôs paralelos atuados por cabos fazem parte de um conjunto de robôs chamados paralelos, que se destacam pela sua rapidez de atuação garantida pela conexão em formato paralelo de seus elos, assim seu principal destaque é o fato de ser controlado por cabos, em face à usual disposição de uniões rígidas dos robôs. Sua mobilização é realizada de maneira segura e eficaz através de cabos que são tracionados por motores, dando movimento à uma plataforma móvel denominada efetuador. Esta monografia apresenta as características gerais, fundamentais para o entendimento básico de funcionamento dos robôs paralelos atuados por cabos, além de construir e validar um modelo matemático de um robô dessa classe, utilizando ferramentas de simulação computacional e a construção de um protótipo físico. Assim, o trabalho apresenta uma visão pouco usual na concepção de robôs expandindo a ideia de robótica em níveis acadêmicos com a finalidade de que essa expansão tenha impactos no desenvolvimento em maior escala desse tipo de robôs.

Palavras-chave: Robôs paralelos atuados por cabos. Modelagem matemática. Prototipagem.

ABSTRACT

ZAGO, J. V. C. Cable-driven parallel robot: a 3 degree of freedom robot case study. 2020. Trabalho de conclusão de curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020

Cable-driven parallel robots are part of a set of robots called parallels, which stand out for their speed of action guaranteed by the parallel connection of their links, so its main highlight is the fact that it is controlled by cables, in view of the usual provision of rigid unions of robots. Its mobilization is carried out safely and effectively through cables that are pulled by engines, giving movement to a mobile platform called end-effector. This monograph presents the general characteristics, fundamental for the basic understanding of the functioning of cable-driven parallel robots, besides constructing and validating a mathematical model of a robot of this class, using computational simulation tools and the construction of a physical prototype. Thus, the work presents an unusual view in the design of robots expanding the idea of robotics at academic levels in order that this expansion has impacts on the larger-scale development of this type of robots.

Keywords: Cable-driven parallel robots. Mathematical modeling. Prototyping.

SUMÁRIO

RESUMO.....	7
ABSTRACT.....	9
1. INTRODUÇÃO.....	13
1.1. CONCEITUAÇÃO.....	13
1.2. ÁREAS DE APLICAÇÃO.....	15
1.2.1. MANUFATURA.....	15
1.2.2. LOGÍSTICA.....	16
1.2.3. CONSTRUÇÃO.....	17
1.2.4. ENTRETENIMENTO.....	17
1.2.5. MEDICINA E REABILITAÇÃO.....	17
1.3. OBJETIVOS.....	18
2. MODELAGEM.....	19
2.1. INTRODUÇÃO DO MODELO.....	19
2.2. MODELAGEM CINEMÁTICA.....	21
2.2.1. CINEMÁTICA INVERSA.....	21
2.2.2. CINEMÁTICA DIRETA.....	23
2.3. MODELAGEM ESTÁTICA.....	24
2.4. MODELAGEM DOS ATUADORES.....	25
3. SIMULAÇÃO COMPUTACIONAL.....	26
3.1. ALGORITMO.....	26
3.2. TRAJETÓRIA EM LINHA RETA.....	27
3.3. TRAJETÓRIA EM ESPIRAL.....	30
3.4. CINEMÁTICA DIRETA.....	33
4. PROTÓTIPO.....	34
4.1. ELETRÔNICA.....	34
4.2. PROGRAMAÇÃO.....	39
4.3. FUNCIONAMENTO DO PROTÓTIPO.....	40
5. CONCLUSÃO.....	44
REFERÊNCIAS.....	45
APÊNDICES.....	46

1. INTRODUÇÃO

1.1. CONCEITUAÇÃO

Tradicionalmente, seguindo o senso comum, quando se tratando de robótica, o usual é pensar em ou retratar um robô humanoide, robôs movidos por rodas ou braço robótico, visto que estes são amplamente difundidos na sociedade, nas artes, na indústria e no âmbito acadêmico. No entanto o conceito de robô se estende muito além, é possível definir, com simplicidade, um robô como sendo uma máquina ou um mecanismo capaz de realizar ações automáticas pré-determinadas.

Desde tempos remotos, a utilização de cabos pela humanidade é muito ampla. Pensar na movimentação de cargas fazendo uso de cabos remete a tempos antigos, incluindo a construção das pirâmides e os primeiros guindastes na Grécia Antiga. Na atualidade, é possível encontrar em diversas áreas do conhecimento, principalmente na construção civil e na produção industrial, a implementação de cabos para movimentar cargas.

Partindo deste ponto, é possível observar como cabos, amplamente utilizados para a transmissão de força e momento, em um primeiro instante, partindo do trabalho manual e, evoluindo ao longo do tempo, juntamente com outras tecnologias, ganham cada vez mais visibilidade e importância no meio da robótica. Assim, com motores e computadores é possível controlar cabos de tal modo que se obtenha um robô paralelo atuado por cabos, em inglês, *Cable-driven Parallel Robots* (CDPRs).

Os CDPRs pertencem à classe dos robôs paralelos, ou seja, robôs que possuem uma base móvel, chamada de *end-effector* (EE), suspensa por mecanismos que atuam independentemente, gerando o movimento, e são classificados de acordo com o número de graus de liberdade (GDL) controláveis. Neste caso, o EE é suspenso por cabos diferentemente da concepção habitual de *links* rígidos. Quando comparados um CDPR e um robô serial, que pode ser exemplificado por um braço robótico amplamente difundido, é possível encontrar diversas vantagens, que incluem a redução de massa e alta velocidade de atuação.

Neste ponto é interessante trazer um pequeno compilado, de fácil entendimento, de termos que estão sendo e serão utilizados ao longo de todo o texto, facilitando, assim, o entendimento para os que não estão familiarizados com o assunto, ou mesmo com os termos do modo em que estão aqui dispostos

- a) atuador:
 - pode-se compreender o atuador como aquele que gera movimento, é o responsável por atuar no sistema do robô como fornecedor de energia mecânica.
- b) robô paralelo:
 - um robô paralelo é chamado desse modo pelos seus elos se ligarem diretamente à uma só junta, de modo que são paralelos um ao outro.
- c) robô serial:
 - no caso do robô serial, este possui elos e juntas construídos em sequência.
- d) elos ou *links*:
 - elos são a parte entre as juntas do robô, podem ser barras, cabos, braços e são movidos conforme o movimento das juntas.
- e) juntas ou *joints*:
 - juntas são a parte responsável pela mobilidade do robô. Nestes pontos são posicionados os atuadores que criam a cinemática para o movimento de fato ocorrer.
- f) efetuador ou *end-effector* (EE):
 - é o ponto final, que se deseja controlar, do robô, leia-se, também, como ponto onde deve ser fixada qualquer ferramenta a ser utilizada.
- g) grau de liberdade (GDL):
 - o número de graus de liberdade de um robô é o número de eixos de coordenadas, translacionais ou rotacionais, nas quais o efetuador é controlável.
- h) *workspace*:
 - a área de trabalho de um robô, se trata de todas as posições que podem ser ocupadas pelo efetuador, existem condições mais aprofundadas mecânicas e de controle a fim de saber a área útil para o controle do efetuador, no entanto, a ideia a ser considerada é apenas sobre a possibilidade do EE ocupar a posição ou não.

Estando disposta essa breve conceituação, é possível prosseguir com a expansão do conceito de CDPRs.

Um exemplo amplamente conhecido de CDPR é o de câmeras que atuam em eventos esportivos e shows, grande inspiração para este projeto. Neste meio têm-se a *SkyCam* (figura 1) e a *spidercam* que são câmeras controláveis por computador e que se deslocam em 3 dimensões por cabos em um espaço para cobrir uma grande área com alta velocidade.

Figura 1 - Câmera SkyCam



Fonte: Disponível em <<http://skycam.tv/>>

Pode-se de tal modo observar um robô atuado por cabos e estudar seus diversos componentes fundamentais. Assim, é necessário compreender que a base móvel (EE) que está conectada por cabos à uma estrutura de sustentação, seja diretamente em um motor ou passando por polias, é dessa forma que se dá a alteração do comprimento dos cabos, ou seja, ativando os motores os cabos serão alongados ou encurtados o que resulta no deslocamento do EE de acordo com o acionamento realizado.

1.2. ÁREAS DE APLICAÇÃO

Um benefício dos CDPRs é a sua alta adaptabilidade, o que resulta em uma gama bem diversificada de áreas de atuação, a capacidade de transportar cargas elevadas, velocidade de atuação, facilidade de transporte e instalação e precisão demonstram como robôs atuados por cabos ganham espaço e se apresentam como boas alternativas em diversas áreas.

1.2.1. MANUFATURA

No ramo da indústria, CDPRs apresentam alta aplicabilidade, sua viabilidade econômica é notável quando se trata de *Workspaces* mais amplos e manufatura aditiva e pintura. Sua capacidade de realizar movimentos rapidamente e com precisão se destacam nessa área.

Um exemplo é do robô GoGiRo (figura 2) sendo utilizado para que sua plataforma móvel seja utilizada como suporte de um dispositivo de pintura.

Figura 2 - GoGiRo sendo utilizado para pintura

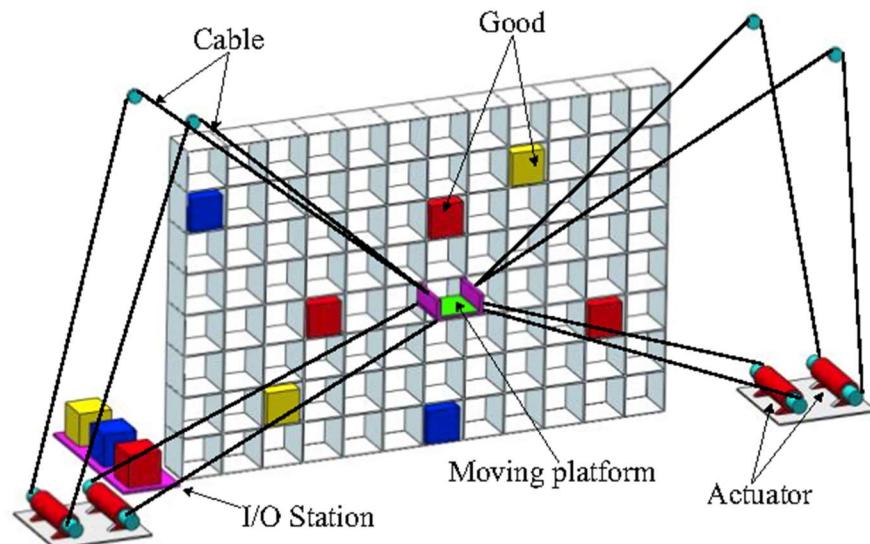


Fonte: Disponível em <<https://www.youtube.com/watch?v=Q05F7Kb96wI>>

1.2.2. LOGÍSTICA

Sistemas de armazenamento utilizando robôs atuados por cabos como o da figura 3 são exemplos claros de aplicação no ramo da logística. Estes robôs conseguem trabalhar em grandes armazéns e auxiliados por um sistema robusto de localização e organização conseguem substituir sistemas mais antigos, lentos e de difícil organização como o trabalho manual e com empilhadeiras.

Figura 3 - esquemático de CDPR sendo utilizado para armazenamento



Fonte: Disponível em
<<https://d3i71xaburhd42.cloudfront.net/1b29d73fd830f385778f0d0bc8c44be08b1b1a93/2-Figure2-1.png>>

1.2.3. CONSTRUÇÃO

Devido à capacidade de adaptação de CDPRs, estes podem ser instalados por meio de guindastes controlados simultaneamente de modo que atuem no ramo da construção civil, geralmente transportando cargas elevadas com precisão ao ponto desejado, conforme exemplo na figura 4.

Figura 4 - CDPR sendo utilizado em uma planta solar



Fonte: Disponível em <<https://bit.ly/2TIYh2K>>

1.2.4. ENTRETENIMENTO

Já citadas anteriormente, câmeras de transmissão que cobrem grandes eventos controladas por cabos são exemplos famosos, são capazes de percorrer grandes distância em alta velocidade e com alta precisão para acompanhar, por exemplo, uma bola de futebol, trabalhando com um *workspace* bem amplo.

Um exemplo já citado anteriormente é o da *spidercam*, um CDPR que é utilizado na transmissão de diversos eventos esportivos.

1.2.5. MEDICINA E REABILITAÇÃO

Na área da medicina, CDPRs são muito utilizados em pesquisas sobre reabilitação neural em diversas situações, em especial reduzindo as cargas exigidas pelo corpo para a

realização de determinados movimentos, auxiliando a reabilitação de pacientes em substituição à utilização de *links* rígidos muitas vezes utilizados.

Têm-se, na figura 6, o exemplo de um exoesqueleto movido a cabos para auxiliar na reabilitação neural de um paciente.

Figura 5 - Exoesqueleto movido à cabos (1)



1.3. OBJETIVOS

Este estudo busca avaliar o conceito de robôs paralelos atuados por cabos, introduzir termos e aplicações para esse tipo de máquinas compreendendo a alternativa presente nesse conceito.

Também busca desenvolver matematicamente um modelo de CDPR com 3 graus de liberdade, capaz de se deslocar pelas coordenadas translacionais cartesianas, o qual será submetido a simulações computacionais.

Por fim o estudo trata do desenvolvimento de um protótipo físico, a partir de um protótipo previamente idealizado para um CDPR com 2 GDLs. (2)

O protótipo busca validar o modelo matemático previamente concluído.

2. MODELAGEM

Aqui serão apresentados os modelos matemáticos do robô, primeiro se dará a introdução pela descrição do modelo e, em sequência, o equacionamento da cinemática direta e inversa bem como o equacionamento estático e dos atuadores.

2.1. INTRODUÇÃO DO MODELO

Dentro da categoria dos *Cable-Driven Parallel Robots* é possível encontrar diversas formas de construir um robô, que se diferenciam em vários pontos, o que inclui o número de cabos, forma de fixação na estrutura fixa e no EE, tamanho, flexibilidade dos cabos, entre outros. O desenho do robô deve ser proposto para atingir sua finalidade.

Neste projeto, a finalidade do robô é se deslocar, a partir de instruções pré-determinadas pelo espaço cartesiano, ou seja, se deslocar com 3 graus de liberdade translacionais. A fim de atingir esse objetivo, o sistema proposto inclui a fixação de 4 cabos à uma estrutura rígida e ao *end-effector*. Esta ligação é feita de tal modo que seja possível controlar a posição do EE no espaço cartesiano. No entanto, não são controláveis os graus de liberdade ligados à rotação nos eixos.

Assim, de acordo com uma característica muito utilizada na classificação de robôs atuados por cabos, diferenciam-se robôs de acordo com a quantidade de cabos utilizados com relação ao número de graus de liberdade. Seguindo o proposto por MING e HIGUCHI (1994) e VERHOEVEN et al. (1998), dá-se o seguinte método de classificação:

a) manipulador **incompletamente restrito**:

- neste caso, o manipulador tem o número de cabos (m) inferior ou igual ao número de graus de liberdade do EE (n), ou seja, $m \leq n$.

b) manipulador **completamente restrito**:

- aqui é possível determinar completamente a posição do EE, nesta configuração, o número de cabos (m) é exatamente igual ao número de graus de liberdade (n) mais um, ou seja, $m = n + 1$.

c) manipulador **completamente restrito e redundante**:

- esta configuração permite ampliar e aprimorar o controle de posição do *end-effector*, já que o número adicional de cabos restringe melhor o movimento, atingindo um controle mais exato e um *workspace* mais amplo. Para esse manipulador, o número de cabos (m) é superior ao número de graus de liberdade (n) mais um, ou seja, $m > n + 1$.

O manipulador proposto nesta tese é classificado como completamente restrito, já que o número cabos (4) é igual ao número de graus de liberdade mais um (4) que o EE é controlável.

Outro parâmetro interessante e de análise relevante para o caso proposto é a forma como os cabos são fixados ao EE. É possível diferenciar as conexões basicamente como sendo com os cabos sem cruzamento, com cruzamento ou ainda uma terceira opção, que é a adotada no projeto, onde os cabos se ligam à um ponto central no EE. Nesta configuração não é possível controlar os graus de liberdade de rotação do robô, entretanto ela permite uma simplificação extremamente útil para o controle do manipulador nos 3 graus de liberdade translacionais do espaço.

Em resumo, neste modelo temos um manipulador atuado por cabos fixos ao EE em um ponto central por quatro cabos, permitindo o controle de três dos seis graus de liberdade do espaço, no caso os translacionais, assim o manipulador é classificado como incompletamente restrito.

Neste ponto é fundamental que sejam definidos alguns parâmetros do sistema. A posição do EE será definida como $p = \{x \ y \ z\}^T$, já que está fixo por um ponto central e tem dimensões pequenas, o que permite trabalhar sem os graus de liberdade rotacionais da parte móvel do robô sem perdas em sua modelagem. Também estarão definidos os comprimentos de cada cabo como sendo: $l_i \{i = 1, \dots, 4\}$; o ângulo entre o cabo i e o eixo x : $\theta_i \{i = 1, \dots, 4\}$; e o ângulo entre o cabo i e o eixo z : $\phi_i \{i = 1, \dots, 4\}$.

Também são definidos aqui, alguns parâmetros do atuador, o que inclui o torque do motor i : $\tau_i \{i = 1, \dots, 4\}$; o ângulo de rotação do motor i : $\beta_i \{i = 1, \dots, 4\}$; e, também, o raio da polia fixada ao motor i : $r_i \{i = 1, \dots, 4\}$.

2.2. MODELAGEM CINEMÁTICA

O robô projetado possui 4 cabos conectados à plataforma móvel de tal modo que podem ser considerados conectados à um ponto central, com isto é possível excluir da modelagem as distâncias entre o ponto de conexão do cabo no EE e seu centro, bem como excluir da modelagem as variáveis rotacionais, simplificando a modelagem matemática.

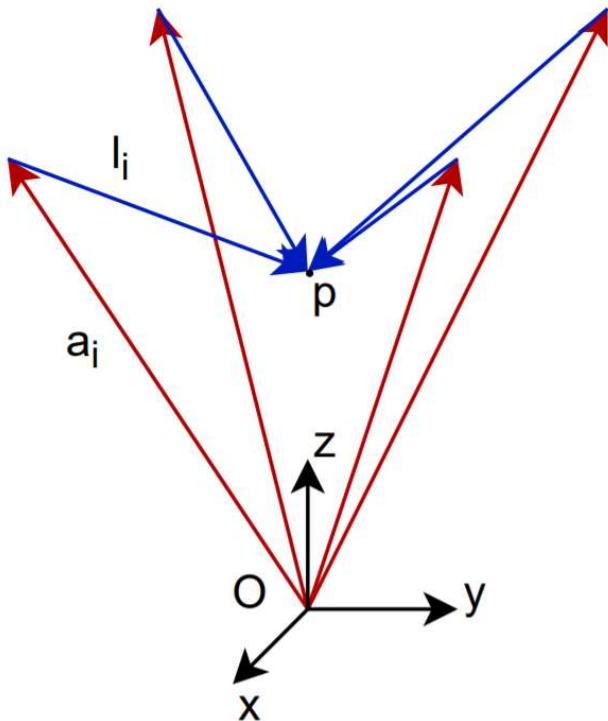
Feitas as devidas observações, a seguir está disposta a modelagem cinemática do CDPR.

2.2.1. CINEMÁTICA INVERSA

A cinemática inversa do CDPR consiste na obtenção dos comprimentos dos cabos à partir da posição do EE, em outras palavras, com a cinemática inversa é possível obter o vetor correspondente ao comprimento dos cabos com base nos dados da *pose* da plataforma móvel do atuador.

O modelo vetorial básico é apresentado na figura 6 a seguir:

Figura 6 – Modelo vetorial do CDPR



Fonte: Elaboração do Autor

A partir do modelo apresentado anteriormente, é possível obter uma primeira equação que relaciona o vetor entre a origem do sistema e o ponto de fixação do cabo i, o comprimento do cabo e a posição do EE.

$$a_i + l_i = p \quad (3.1)$$

Reescrevendo a equação (1) de modo, a fim de isolar o vetor comprimento do cabo, a relação encontrada é:

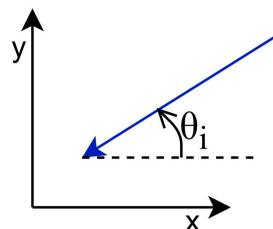
$$l_i = p - a_i \quad (3.2)$$

De tal modo, como as variáveis rotacionais do EE estão sendo descartadas, o módulo do vetor do cabo será dado por:

$$|l_i| = \sqrt{(x - a_{ix})^2 + (y - a_{iy})^2 + (z - a_{iz})^2} \quad (3.3)$$

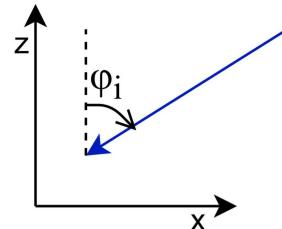
Seguindo este raciocínio é possível determinar os ângulos de rotação dos cabos, representados nas figuras 7 e 8, θ_i em relação ao eixo x e ϕ_i em relação ao eixo z:

Figura 7 – Projeção de l_i no plano XY



Fonte: Elaboração do Autor

Figura 8 – Projeção de l_i no plano XZ



Fonte: Elaboração do Autor

$$\theta_i = \arctan\left(\frac{y - a_{iy}}{x - a_{ix}}\right) \quad (3.4)$$

$$\varphi_i = \arctan\left(\frac{x - a_{ix}}{z - a_{iz}}\right) \quad (3.5)$$

2.2.2. CINEMÁTICA DIRETA

Ao inverso da cinemática inversa, que é consideravelmente mais simples em CDPRs, a cinemática direta consiste em determinar a *pose* do EE com base nos comprimentos dos cabos. No robô do projeto, graças a ligação por ponto central basta o conhecimento de apenas um vetor cabo para a determinar com facilidade a posição do EE.

No entanto, como apenas o módulo do comprimento do cabo é uma variável mais fácil de se obter do que os ângulos do cabo, para equacionar a cinemática direta, serão necessários os comprimentos de ao menos 3 cabos.

Para o caso estudado, o equacionamento cinemático será feito, portanto, tomando como base o comprimento dos cabos. Pensando de maneira simplória e, ignorando o *workspace* do manipulador, um cabo com comprimento l_i pode ter sua ponta em qualquer ponto de uma esfera de raio l_i . Quando se tratando de 4 cabos, os quais as pontas devem estar conectadas à um ponto em comum, as esferas têm apenas 1 ou 2 pontos em comum. Nesse caso, desses pontos, apenas um estará contido no *workspace* do robô. É dessa forma que será feito o equacionamento da cinemática direta neste projeto.

Para encontrar a interseção de n , será utilizado o primeiro método de Coope (3), este método utiliza a fatoração LU, ou Eliminação de Gauss, e requer que os centros das esferas sejam linearmente independentes. Aqui não serão apresentados todos os passos algébricos utilizados para obtenção da interseção das esferas.

É interessante destacar que o método utilizado tem resultado baseado em uma equação quadrática, então os possíveis resultados são: nenhum ponto em comum, um ponto em comum (o que equivale às duas raízes iguais) e 2 pontos em comum. Conforme dito anteriormente, no manipulador projetado as esferas encontradas utilizando o comprimento dos cabos retornaram sempre 1 ou 2 pontos em comum, em geral ocorrerá a segunda hipótese, já que a primeira se dá apenas quando o EE se encontra no plano que contém os pontos de fixação dos cabos.

O método utilizado consiste em encontrar os valores de p que satisfazem a relação:

$$\|p - a_i\|_2^2 = l_i^2 \quad (3.6)$$

Aqui, a próxima equação trará o resultado para p , no entanto é fundamental que alguns parâmetros sejam definidos previamente a fim de compreender a equação encontrada. Assim, têm-se os seguintes parâmetros requeridos: $b_i = \|a_i\|_2^2 - l_i^2$, $A = [a_1, \dots, a_n]$, $e = [1, \dots, 1]^T$, $B = [b_1, \dots, b_n]^T$, $u = A^{-T}e$ e $v = A^{-T}B$.

Por fim, a equação que resolve o sistema a fim de encontrar a posição do EE, é:

$$p = \left(\frac{(2 - u^T v) \pm \sqrt{(2 - u^T v)^2 - (u^T u)(v^T v)}}{2u^T u} \right) u + \frac{1}{2} v \quad (3.7)$$

2.3. MODELAGEM ESTÁTICA

Na modelagem estática serão analisadas as forças aplicadas ao EE pelos cabos bem como a ação da gravidade, obtendo, assim, a força resultante sobre o EE, ou seja, a modelagem servirá para a partir do tracionamento dos cabos, definir qual a força resultante que será responsável por movimentar a parte móvel do robô.

Desse modo, o vetor da força resultante sobre o EE é dado de tal modo que equivale à soma das forças de tração T_i ($i = 1, \dots, 4$) mais a força F_g causada pela gravidade, representado na equação a seguir.

$$F_R = \sum_{i=1}^4 T_i + F_g \quad (3.8)$$

Nesse ponto, é importante destacar que o vetor T_i é obtido a partir da multiplicação do escalar t_i , que representa o módulo da tração à qual está sendo aplicada ao cabo i , e o vetor $-L_i$, o inverso do vetor L_i que representa a direção do cabo i , dado por $L_i = [\cos \theta_i, \sin \theta_i, \cos \varphi_i]^T$.

De tal modo, sabendo que o vetor \mathbf{F}_R é dado da forma $F_R = [F_{RX}, F_{RY}, F_{RZ}]^T$, a força resultante sobre o *end-effector* é dada como sendo:

$$\begin{bmatrix} F_{RX} \\ F_{RY} \\ F_{RZ} \end{bmatrix} = \begin{bmatrix} -\cos \theta_1 & -\cos \theta_2 & -\cos \theta_3 & -\cos \theta_4 \\ -\sin \theta_1 & -\sin \theta_2 & -\sin \theta_3 & -\sin \theta_4 \\ -\cos \varphi_1 & -\cos \varphi_1 & -\cos \varphi_1 & -\cos \varphi_1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -Mg \\ 0 \end{bmatrix} \quad (3.9)$$

2.4. MODELAGEM DOS ATUADORES

Por fim, é necessário modelar os atuadores do sistema, isso significa modelar os motores que, quando acionados, serão responsáveis por tracionar os cabos que movimentam o EE.

Neste trecho o fundamental é que o atuador seja capaz de controlar a posição do *end-effector*, isso coloca o controle dos torques nos motores e trações nos cabos em segundo plano, considerando que a massa da plataforma móvel é significativamente pequena.

Portanto, a modelagem dos atuadores terá foco em controlar o comprimento l_i de cada cabo. Pensando o motor como acionador de uma bobina de raio r_i onde cada cabo é fixado, o controle do comprimento do cabo é feito a partir do controle do ângulo de rotação β_i de cada motor. De tal modo, é possível encontrar a variação Δl_i de cada cabo, que corresponde ao comprimento do cabo l_i subtraído do comprimento l_{0i} do cabo na posição “zero”, como função do ângulo de rotação do motor, obtendo, portanto:

$$\Delta l_i = -r_i \beta_i \quad (3.10)$$

Neste caso, para que o EE se encontre na posição desejada, é necessário controlar o ângulo de rotação de cada motor.

Dessa maneira, adotando valores iguais de raio para todos os atuadores, é possível obter β_i para cada atuador sabendo que $|l_i| = \sqrt{(x - a_{ix})^2 + (y - a_{iy})^2 + (z - a_{iz})^2}$ e de modo semelhante, $|l_{0i}| = \sqrt{(x_0 - a_{ix})^2 + (y_0 - a_{iy})^2 + (z_0 - a_{iz})^2}$. Portanto os valores de β_i são dados por:

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = -\frac{1}{r} \begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \\ \Delta l_4 \end{bmatrix} \quad (3.11)$$

3. SIMULAÇÃO COMPUTACIONAL

Após realizada a modelagem do robô, é possível simular computacionalmente o funcionamento e a geometria do robô. Utilizando o MATLAB®, um *software* voltado para operações numéricas e algébricas, é possível representar o CDPR do projeto geometricamente, seja plotando suas coordenadas de posição, valores de comprimento e ângulo dos cabos, ou representando o atuador visualmente.

A simulação computacional tem por finalidade principal, validar o modelo construído anteriormente além de servir como base para construção do protótipo, bem como implementação de códigos nos controladores e construir um resultado esperado de como agirá o CDPR projetado.

3.1. ALGORITMO

Para a implementação do algoritmo em MATLAB, foram observados uma série de pontos a fim de construir um modelo computacional simples, robusto e de boa representatividade do projeto esperado, além de servir como base para um projeto mais elaborado que venha a envolver um maior número de graus de liberdade de atuação do robô.

O primeiro passo no algoritmo foi a definição de diversas variáveis, conhecidas ou não, que são necessárias para a simulação ou que serão obtidas e analisadas durante a simulação. Nessas variáveis estão inclusas as condições iniciais do robô.

Nesta declaração inicial, as variáveis do *workspace* nos 3 eixos foi definida como sendo 300 mm nas direções dos eixos x e y (de -150 até 150) e 150 mm na direção do eixo z (de 0 a 150), o raio da polia do motor foi determinado como sendo 10 mm, essas dimensões foram escolhidas de modo à ficarem com valores de fácil observação, mas não distantes dos valores do protótipo construído posteriormente, sendo assim uma comparação para a validação do modelo completamente possível.

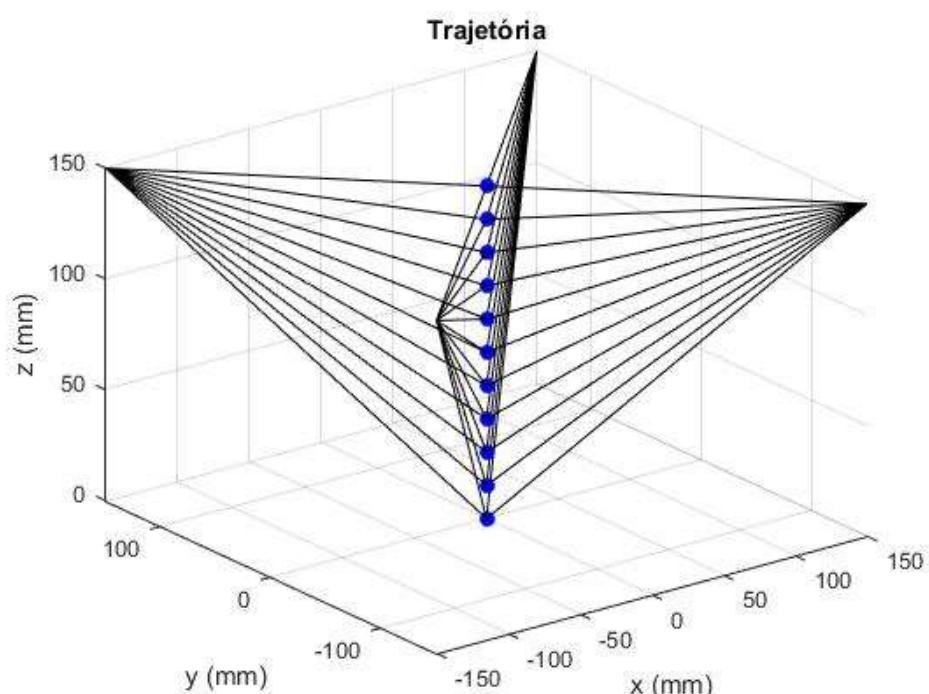
Todos os códigos dos programas em MATLAB estão dispostos nos Apêndices.

3.2. TRAJETÓRIA EM LINHA RETA

Após a definição de variáveis, uma simulação na qual o robô descreve uma descida em linha reta (ao longo do eixo z) foi construída, desse modo foi possível obter com simplicidade diversas informações do modelo projetado, incluindo o comprimento e ângulos dos cabos e ângulos de rotação dos motores.

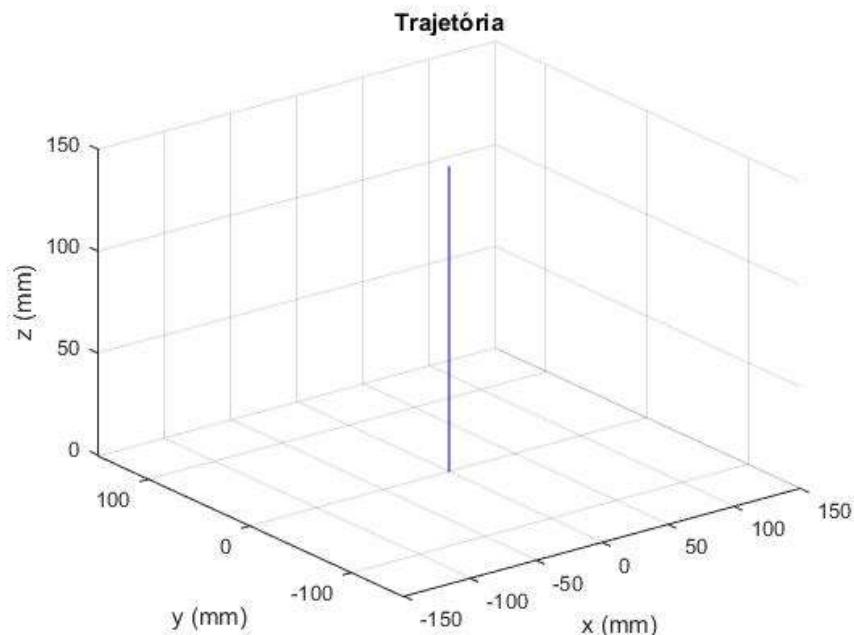
Primeiro está definida a trajetória realizada pelo CDPR (figuras 9 e 10).

Figura 9 – Trajetória em linha reta com cabos



Fonte: Elaboração do Autor

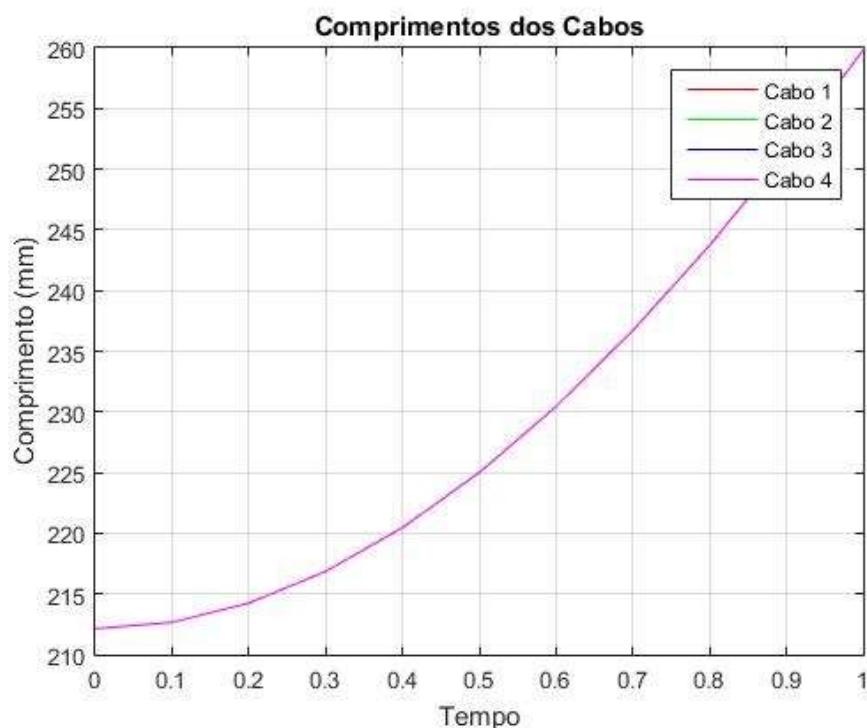
Figura 10 – Linha da trajetória em linha reta



Fonte: Elaboração do Autor

Em seguida a simulação traz os valores de l_i para os 4 cabos, nota-se que os 4 cabos têm o mesmo comprimento ao longo de todo o trajeto (figura 11).

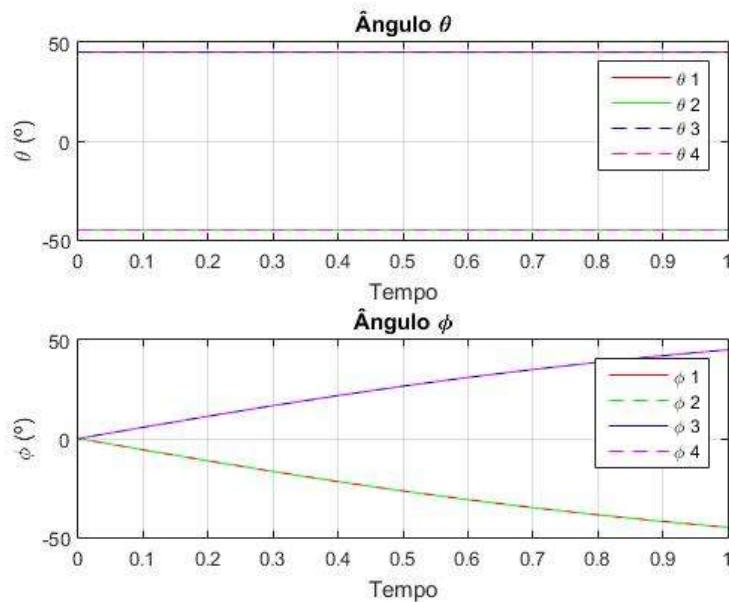
Figura 11 – Comprimentos dos cabos na trajetória em linha reta



Fonte: Elaboração do Autor

Depois são apresentados os ângulos θ e ϕ dos cabos em relação, respectivamente, aos eixos x e z (figura 12).

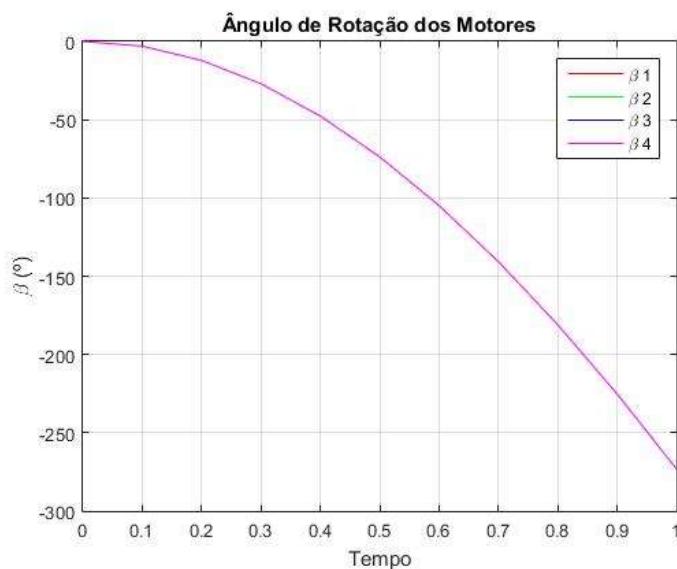
Figura 12 – Ângulos dos cabos na trajetória em linha reta



Fonte: Elaboração do Autor

Por fim, um último gráfico (figura 13) apresenta os ângulos de rotação dos motores para cada motor, aqui, de modo similar aos comprimentos dos cabos, os ângulos dos 4 motores são iguais ao longo de toda a trajetória.

Figura 13 – Ângulos dos motores na trajetória em linha reta

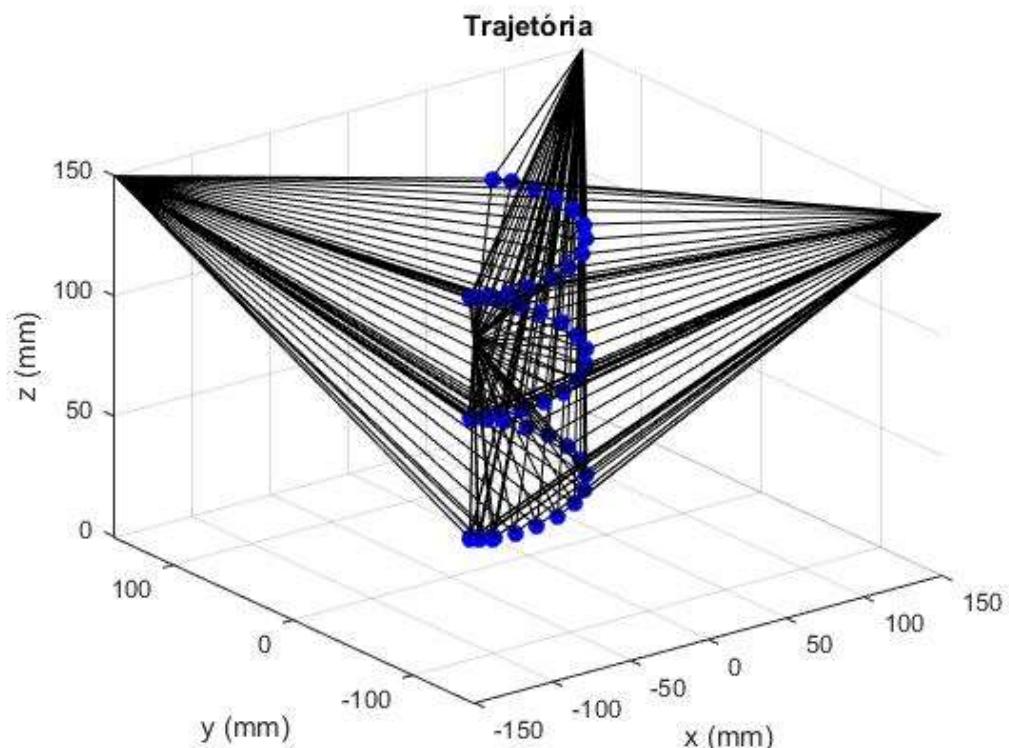


Fonte: Elaboração do Autor

3.3. TRAJETÓRIA EM ESPIRAL

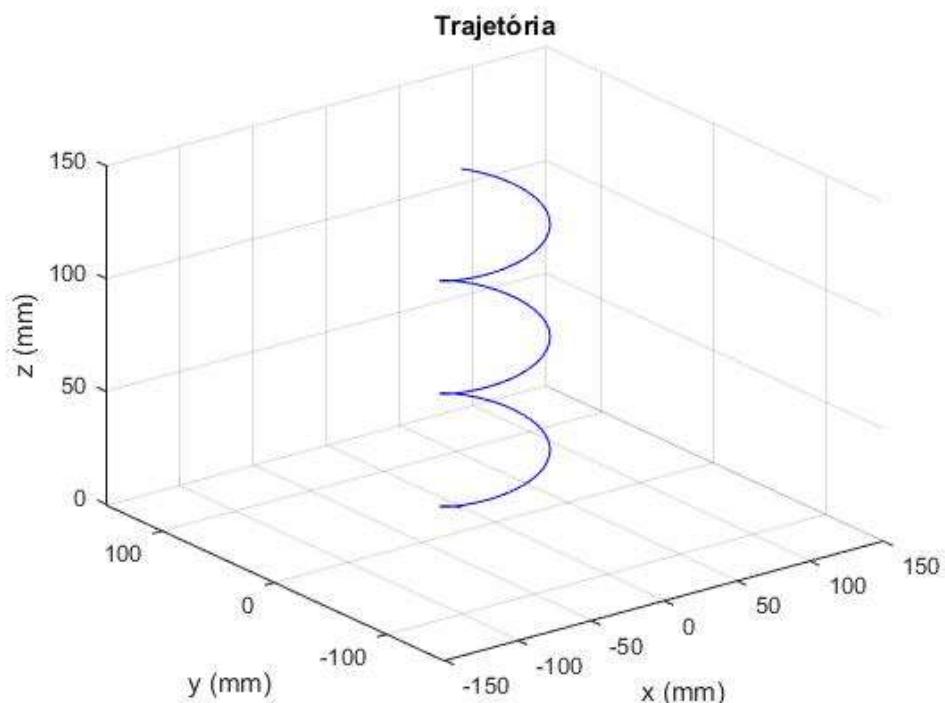
Posteriormente, a trajetória da simulação foi alterada para a descrição de uma espiral pelo EE (figuras 14 à 18). Os gráficos obtidos estão apresentados na mesma sequência que os gráficos da trajetória em linha reta, porém, desta vez, é possível perceber um pequeno aumento na complexidade das variáveis.

Figura 14 – Trajetória em espiral com cabos



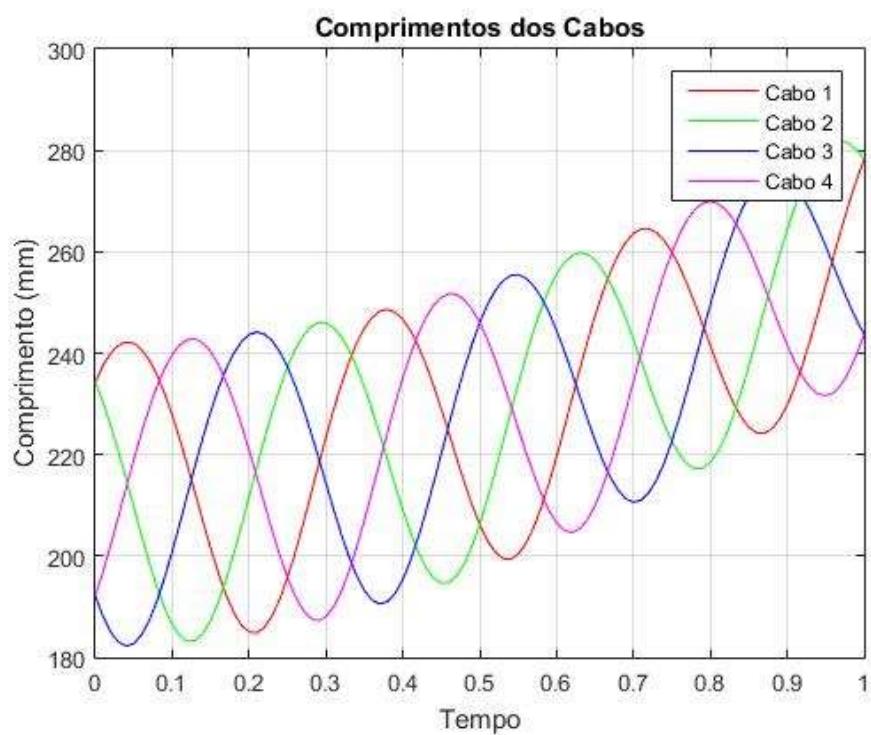
Fonte: Elaboração do Autor

Figura 15 – Linha da trajetória em espiral



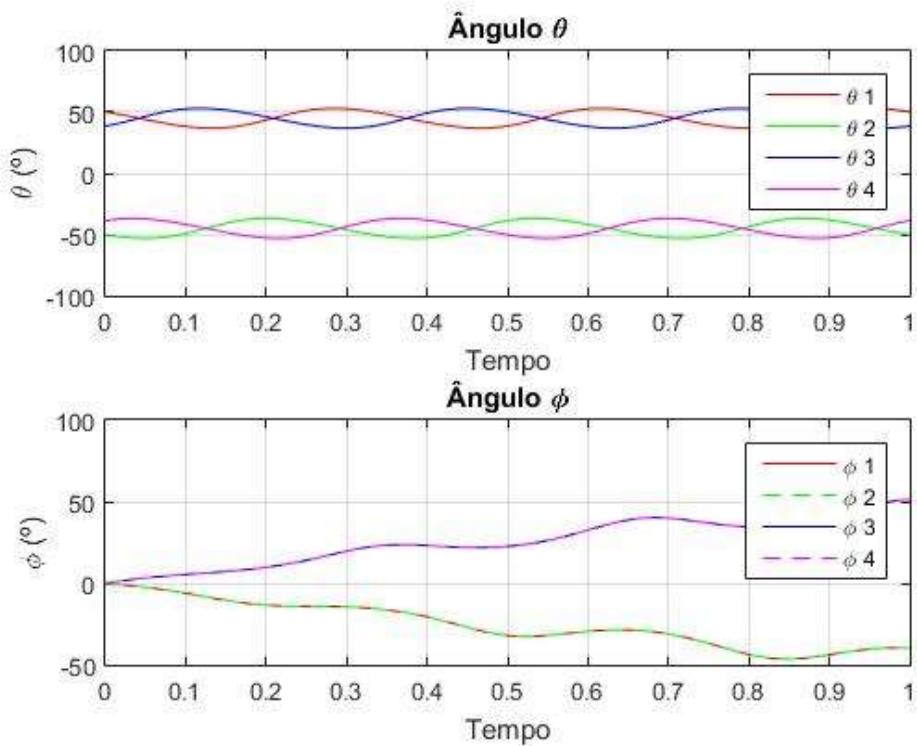
Fonte: Elaboração do Autor

Figura 16 – Comprimento dos cabos na trajetória em espiral



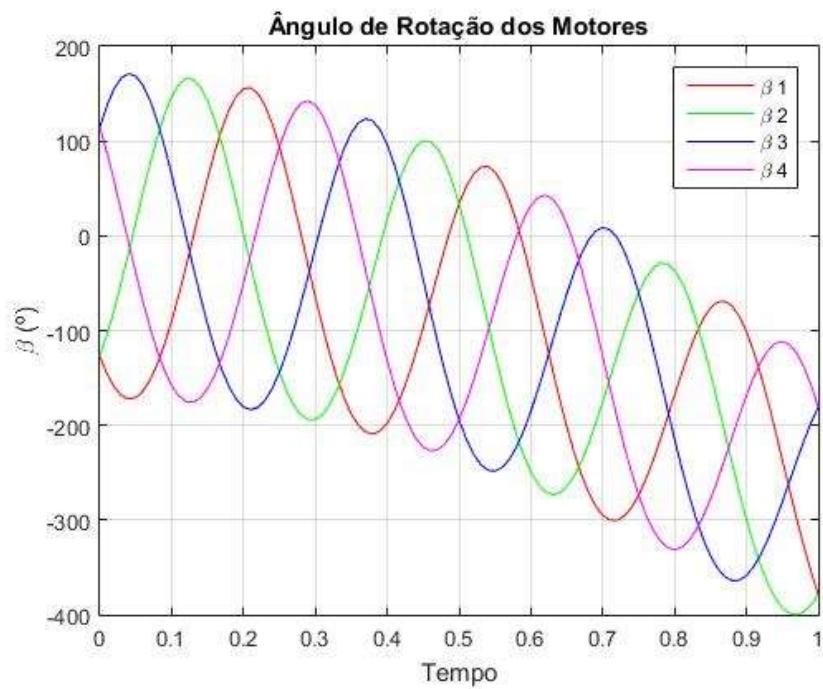
Fonte: Elaboração do Autor

Figura 17 – Ângulos dos cabos na trajetória em espiral



Fonte: Elaboração do Autor

Figura 18 – Ângulos dos motores na trajetória em espiral



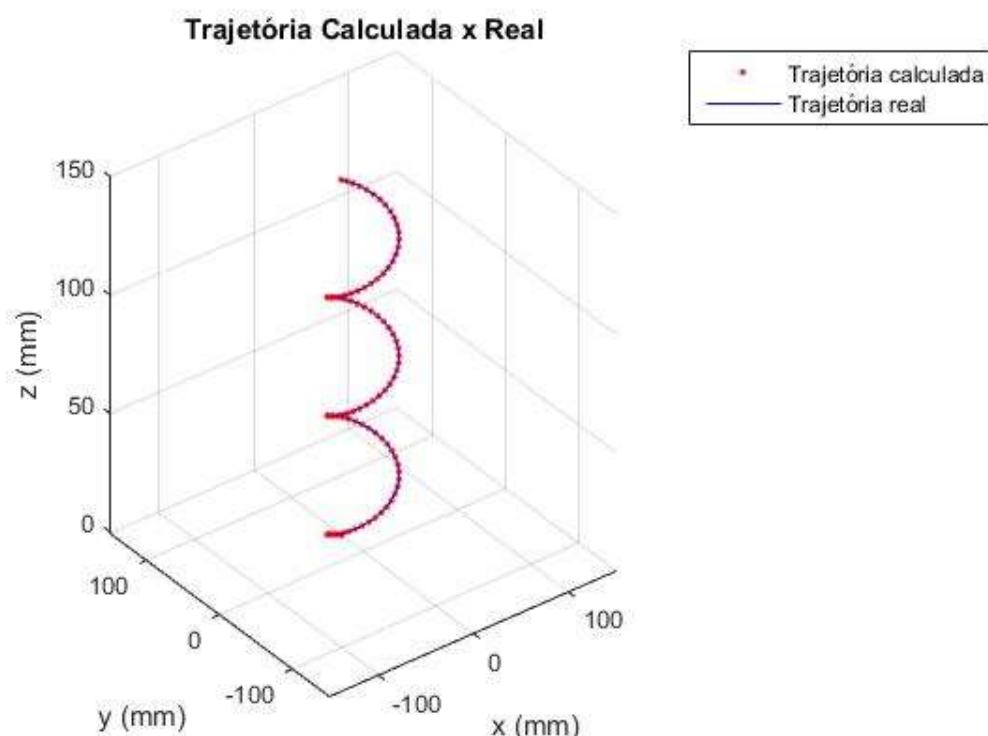
Fonte: Elaboração do Autor

3.4. CINEMÁTICA DIRETA

Por fim, foi possível verificar o modelo para a cinemática direta do CDPR pela simulação computacional. Assim, com os comprimentos de 3 cabos, facilmente obtidos com os ângulos de rotação dos motores que são fatores passíveis de obtenção e controle, é possível determinar a posição do EE.

Desse modo, foi criado um algoritmo que representa a posição do EE a partir do comprimento de 3 cabos, utilizando os comprimentos de cabos obtidos na simulação da trajetória em espiral, foi desenvolvido para comparar a trajetória real com a calculada utilizando o modelo de cinemática direta (figura 19).

Figura 19 – Comparação entre as trajetórias



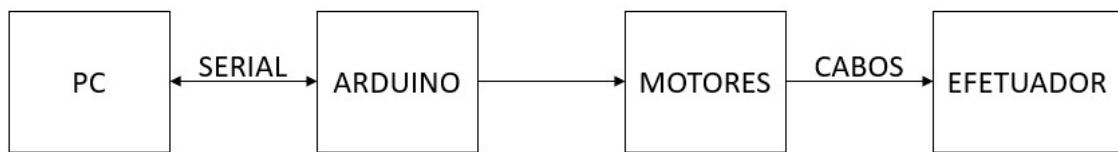
Fonte: Elaboração do Autor

4. PROTÓTIPO

Para a validação do modelo obtido, foi desenvolvido um protótipo simples, porém, capaz de realizar as ações mecânicas esperadas de um CDPR com 3 graus de liberdade. Para isso, um sistema eletrônico controlado por um Arduino foi instalado de modo que correspondesse ao modelo construído matematicamente.

No protótipo, o controlador fica conectado à um computador, de onde recebe as instruções. O computador também é responsável pela alimentação do protótipo. O diagrama de blocos do protótipo está disposto da figura 20 a seguir.

Figura 20 – Diagrama de blocos do funcionamento do protótipo



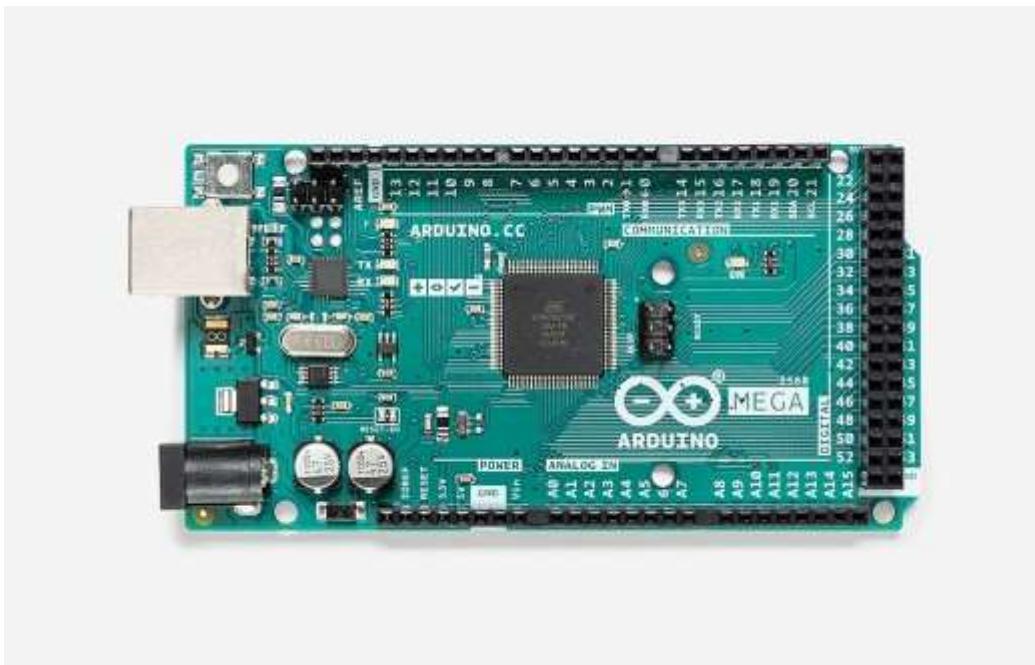
Fonte: Elaboração do Autor.

4.1. ELETRÔNICA

Nesta seção estão descritos os componentes eletrônicos utilizados na construção do protótipo e, também, o circuito eletrônico do CDPR.

Para o controle do protótipo, o microcontrolador utilizado foi um Arduino Mega 2560 (figura 21), que é responsável por conversar com o computador, via comunicação serial, e controlar os motores de passo instalados, a fim de controlar a posição do EE.

Figura 21 – Arduino Mega 2560



Fonte: Disponível em <<https://store.arduino.cc/usa/mega-2560-r3>>

Assim, os quatro motores utilizados são do modelo 28BYJ-48 e para sua utilização, estão presentes quatro drivers ULN2003 (figura 22), um para cada motor. Esses motores são facilmente alimentados com 5V pela própria porta de saída de tensão do do Arduino Mega, e estão melhor representados e descritos nos esquemáticos abaixo (figura 23).

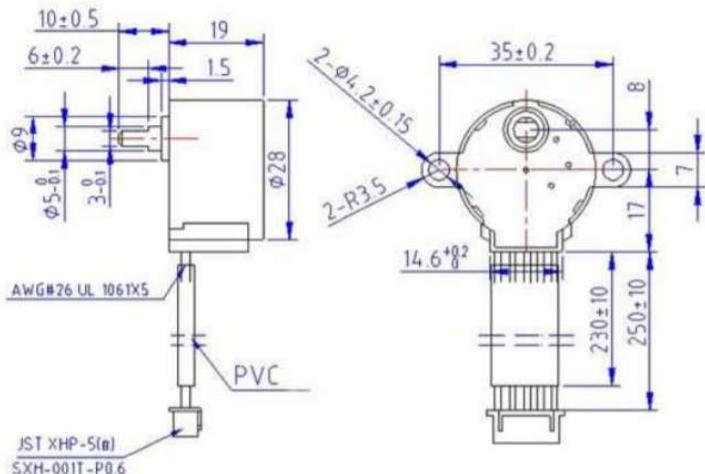
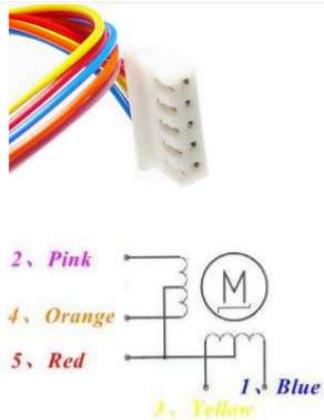
Figura 22 – Motor 28BYJ-48 com driver ULN2003



Fonte: Disponível em <<https://www.4hobby.com.br/Motor-De-Passo-28byj-48-Com-Driver-Uln2003-Arduino-100225>>

Figura 23 – Datasheet 28BYJ-48

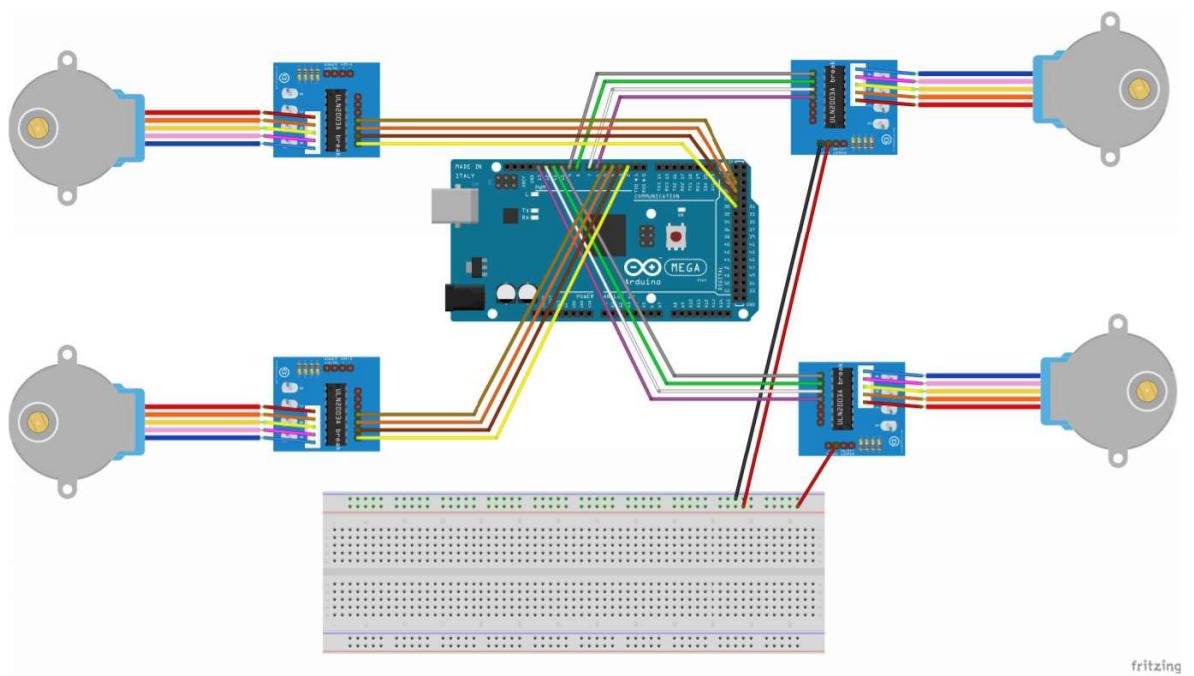
Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz, No load, 10cm)
Model	28BYJ-48 – 5V



Fonte: Disponível em <<https://datasheetspdf.com/pdf/1006817/Kiatronics/28BYJ-48/1>>

Por fim, com a intenção de demonstrar melhor as ligações realizadas no circuito, uma versão foi criada no Fritzing, um software de design de Hardware eletrônico, seu caráter é *open source* e acaba mais voltado para circuitos amadores, mas cumpre bem a função de representar de maneira clara e objetiva a construção do circuito eletrônico do protótipo (figuras 24).

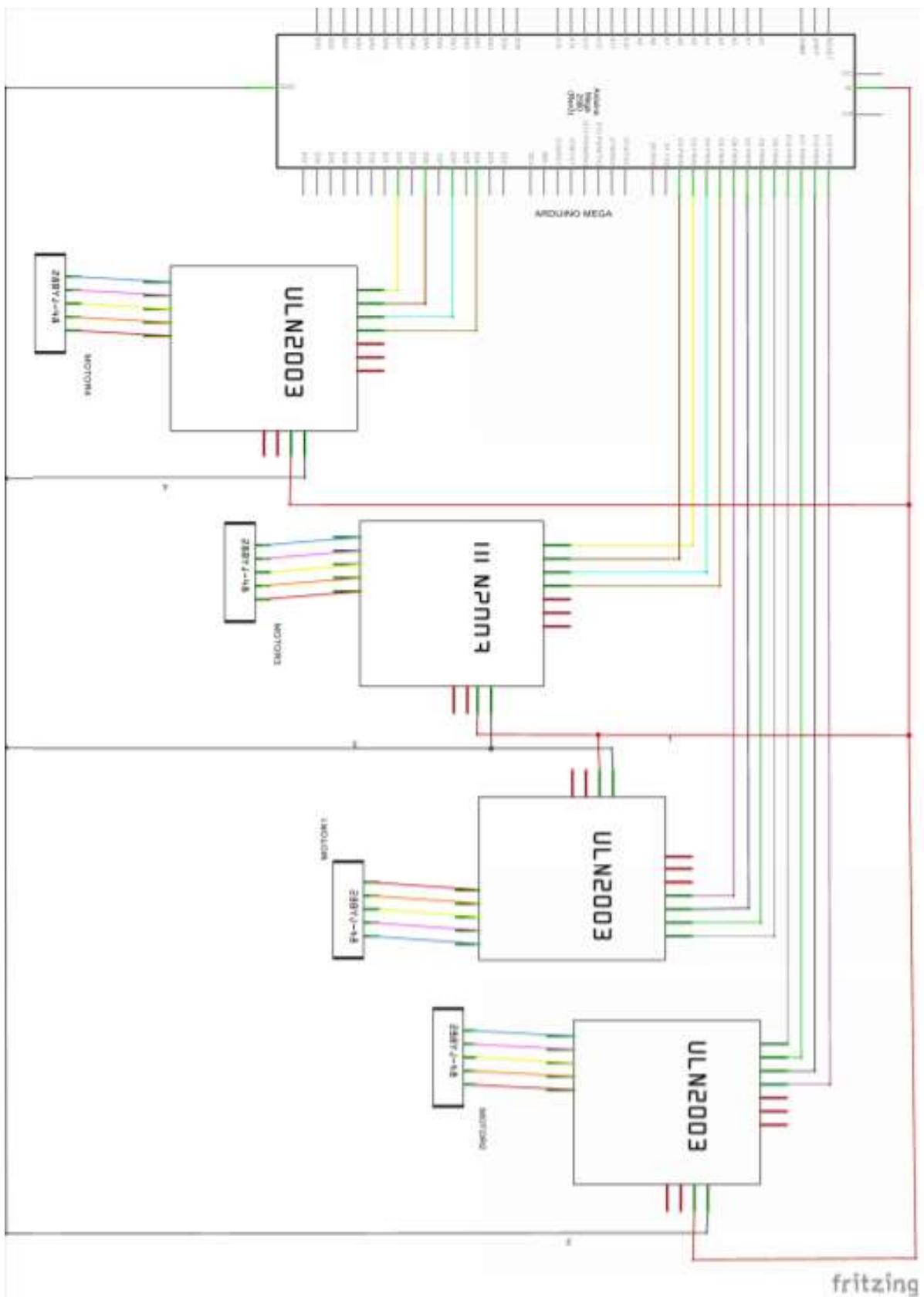
Figura 24 – Representação do circuito eletrônico



Fonte: Elaboração do Autor

Assim, também no Fritzing, além da representação do circuito, foi criado um esquemático das ligações do circuito, que se encontra na figura 25.

Figura 25 – Esquemático do circuito eletrônico



Fonte: Elaboração do Autor

4.2. PROGRAMAÇÃO

Para o controle do protótipo via Arduino, foi desenvolvido um programa em C++ na IDE do Arduino. O programa recebe comandos de posição via comunicação serial e aciona os motores para que o EE atinja a posição estipulada. Para isso, o programa conta com rotinas para recebimento de dados de posição, cálculo do giro que os motores realizarão e de acionamento dos motores. O programa completo está disposto nos Apêndices.

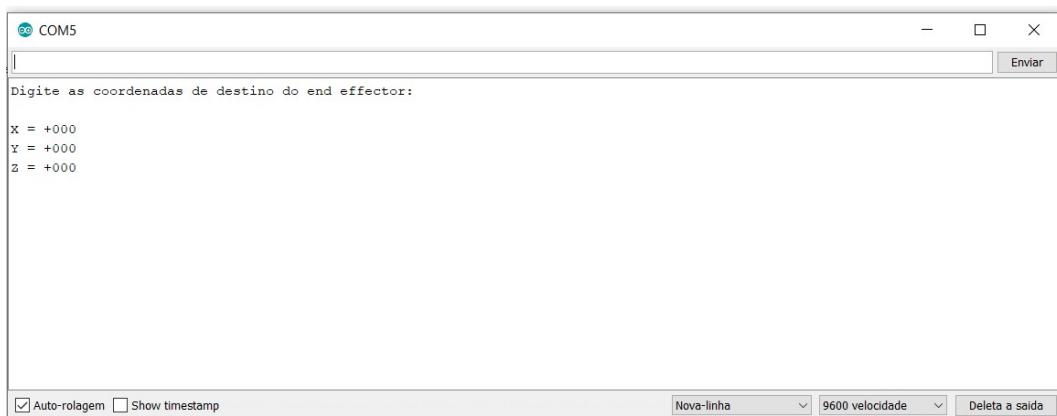
Na primeira parte do programa são incluídas as bibliotecas math.h e AccelStepper.h, a primeira dispõe de ferramentas matemáticas que posteriormente serão utilizadas para os cálculos necessários durante o funcionamento do protótipo, enquanto a segunda biblioteca é utilizada para o controle dos motores de passo.

Apesar do Arduino dispor da biblioteca Stepper.h, que é de entendimento mais simples, a biblioteca adicionada funciona melhor no caso proposto. Isso se deve ao fato de a biblioteca escolhida apresentar ferramentas que melhor satisfizeram a necessidade de controle simultâneo dos motores.

Em um próximo momento o programa conta com a inclusão dos motores de passo em seus respectivos pinos de saída digitais, além da declaração de variáveis dimensionais do protótipo e variáveis necessárias para a entrada de dados via comunicação serial (figura 26).

No Setup do Arduino fica determinada a entrada do tipo serial que o programa utiliza, e, também são definidos os parâmetros de velocidade e aceleração dos motores, seguindo os padrões da biblioteca AccelStepper.h.

Figura 26 – Monitor Serial do Arduino



Fonte: Elaboração do Autor.

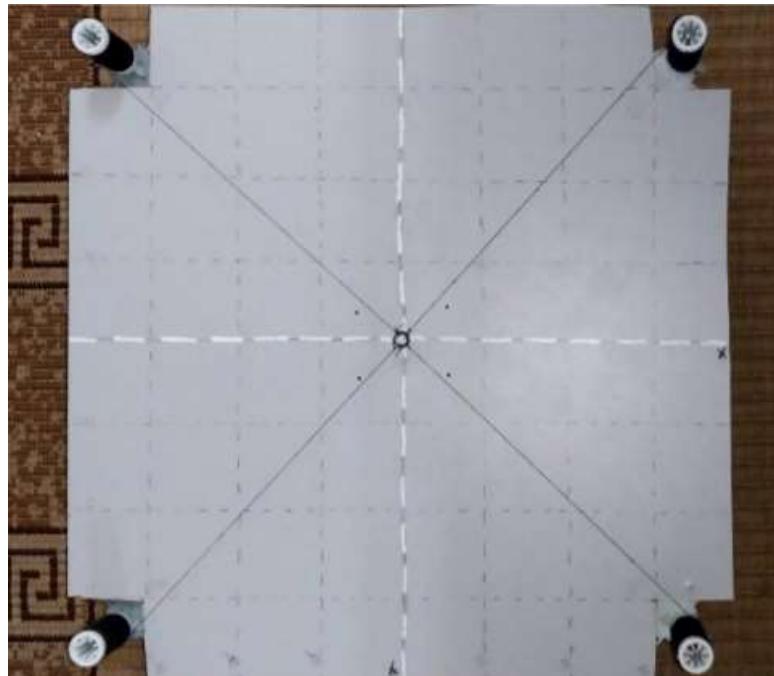
Já no Loop, o programa apresenta a rotina de recebimento de dados via comunicação serial. Então os dados são processados de forma que, após recebidos, são calculados os comprimentos de cabo para os quatro cabos e, portanto, o giro necessário a ser realizado por cada motor para que o EE atinja a posição desejada. Por fim, é chamada a rotina de funcionamento dos motores.

Para o acionamento dos motores é definida uma função própria que conta com a transformação do ângulo de giro dos motores em passos e, por fim, o acionamento dos quatro motores utilizando a biblioteca para a manipulação de motores de passo AccelStepper.h.

4.3. FUNCIONAMENTO DO PROTÓTIPO

O ponto inicial para o desenvolvimento do protótipo foi um protótipo previamente desenvolvido para um CDPR de 2 GDLs (figura 27). ([2](#))

Figura 27 – Protótipo inicial ([2](#))



Deste ponto o protótipo foi modificado a fim de ser capaz de atuar nos 3 graus de liberdade do espaço cartesiano, para isso foram adicionadas estruturas, feitas por impressão 3D, por onde passam os cabos mantendo o EE suspenso (figura 28).

Figura 28 – Estrutura de suspensão dos cabos



Fonte: Elaboração do Autor.

A fixação dos cabos no motor também foi modificada (figura 29), assim, os carreteis de costura fixados aos motores do protótipo anterior foram substituídos por polias a fim de diminuir a imprecisão de posicionamento do EE que o comprimento do carretel gerava. Os motores também foram fixados de modo que seu eixo de atuação se encontre na posição paralela ao plano do protótipo para que os ângulos de saída dos cabos com a polia permaneçam os mesmos em qualquer posição do EE.

Figura 29 – Fixação dos motores



Fonte: Elaboração do Autor.

Por fim, a fixação do EE também foi repensada, desse modo, um corpo de maior massa foi fixado à ponta dos cabos para garantir que se mantenham tracionados durante todo o funcionamento do protótipo. O EE está na figura 30.

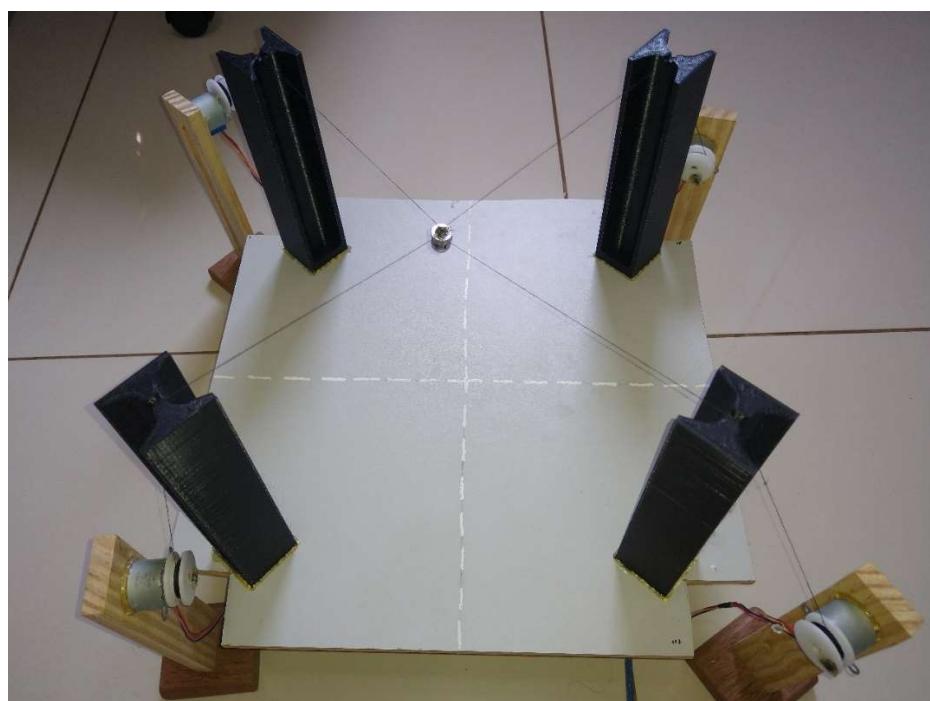
Figura 30 – *End-effector*



Fonte: Elaboração do Autor.

O protótipo (figura 31), portanto, foi colocado sob testes para seu funcionamento adequado e o intento de validar o modelo matemático construído anteriormente, no entanto o funcionamento real do protótipo não correspondeu às expectativas para a formulação de uma base de dados suficientemente exata para a validação do modelo. Esse fato é observado visualmente quando após a realização dos mesmos movimentos, o efetuador se encontra em posições variáveis, devido à perda de tração dos cabos.

Figura 31 – Protótipo final



Fonte: Elaboração do Autor.

Isso ocorre, principalmente, devido às limitações dos motores utilizados no protótipo bem como sua programação em Arduino, o controle simultâneo das velocidades dos motores se mostrou inviável com a estrutura de programação utilizada, revelando a necessidade da reformulação da eletrônica do protótipo e utilização de hardwares capazes do controle em paralelo dos 4 motores.

De fato, a imprecisão por fatores mecânicos era esperada devido ao caráter menos elaborado da montagem do protótipo, no entanto a reformulação do protótipo, se mostrou insuficiente para a validação do modelo, apesar disso, traz consigo o interesse, e uma direção para trabalhos posteriores, na criação de um novo protótipo, que se baseie no modelo matematicamente construído, planejado especialmente para esta aplicação.

5. CONCLUSÃO

Ao longo deste estudo foi possível trazer o conceito de CDPRs, bem como demonstrar sua aplicabilidade em diversas áreas e, desenvolver, deste modo, um modelo matemático, funcional e aplicável, demonstrado através da simulação computacional desse modelo. Desse modo, a discussão sobre essa classe de robôs, que ainda não tem seu devido destaque, esteve enriquecida no âmbito acadêmico.

O estudo também levantou importantes pontos sobre o desenvolvimento de protótipos, fruto do caráter experimental e inovativo do desenvolvimento do protótipo de CDPR, além da incitação à futuros trabalhos seguindo a mesma linha, que tragam consigo o desenvolvimento de protótipos mais robustos e a aplicação de técnicas de controle em tempo real.

REFERÊNCIAS

- 1 QIAN, S., ZI, B., SHANG, W. *et al.* **A Review on Cable-driven Parallel Robots.** *Chin. J. Mech. Eng.* 31, 66. 2018.
- 2 DURAN, L. D R. **Robô Paralelo atuado por Cabos: Estudo de caso de um robô com 2 GDL.** 2018. 048f. Tese de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.
- 3 MAIOLI, D S. **Interseção de esferas no R^n e aplicações.** Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, Campinas, 2017.
- 4 POTT, A. & BRUCKMANN, T. **Cable-driven Parallel Robots - Proceedings of the Fourth International Conference on Cable-driven Parallel Robots.** 2019. 10.1007/978-3-030-20751-9.
- 5 SHAHMIRI, F. & GENTRY, R. **A Survey of Cable-Suspended Parallel Robots and their Applications in Architecture and Construction.** 2016. 914-920. 10.5151/despro-sigradi2016-484.
- 6 MERSI, R. & VALI, S. & HAGHIGHI, M. & ABBASNEJAD, G. & MASOULEH, M. **Design and Control of a Suspended Cable-Driven Parallel Robot with Four Cables.** 2018. 10.1109/ICRoM.2018.8657534.
- 7 POTT, A. & SCHMIDT, V. **On the Forward Kinematics of Cable-Driven Parallel Robots.** 2015. 10.1109/IROS.2015.7353818.
- 8 PARK, J. & JIN, X. & JUN, D. & POTT, A. & PARK, S. & KO, S. **Four-cable-driven parallel robot.** 2013.
- 9 WANG, W. et al. **Study on Energy Consumption and Cable Force Optimization of Cable-Driven Parallel Mechanism in Automated Storage/Retrieval System.** 2015. *Second International Conference on Soft Computing and Machine Intelligence (ISCFI)* (2015): 144-150.

APÊNDICES

APÊNDICE A – Programa para a simulação em MATLAB: Dados

```
% Dimensão da Base
Q = 300;

% Altura das polias
H = 150;

% Coordenadas dos pontos de ancoragem
A1 = [-Q/2 -Q/2 H];
A2 = [Q/2 -Q/2 H];
A3 = [Q/2 Q/2 H];
A4 = [-Q/2 Q/2 H];

% Dimensão da polia
r = 10;

% Dimensões Iniciais dos cabos na posição X0 Y0
x = 0;
y = 0;
z = 150;

% Comprimento dos cabos na posição inicial
L01 = sqrt((x - A1(1))^2 + (y - A1(2))^2 + (z - A1(3))^2);
L02 = sqrt((x - A2(1))^2 + (y - A2(2))^2 + (z - A2(3))^2);
L03 = sqrt((x - A3(1))^2 + (y - A3(2))^2 + (z - A3(3))^2);
L04 = sqrt((x - A4(1))^2 + (y - A4(2))^2 + (z - A4(3))^2);

% Ângulos dos cabos
Theta01 = radtodeg(atan((y - A1(2))/(x - A1(1))));
Theta02 = radtodeg(atan((y - A2(2))/(x - A2(1))));
Theta03 = radtodeg(atan((y - A3(2))/(x - A3(1))));
Theta04 = radtodeg(atan((y - A4(2))/(x - A4(1))));
Phi01 = radtodeg(atan((z - A1(3))/(y - A1(2))));
Phi02 = radtodeg(atan((z - A2(3))/(y - A2(2))));
Phi03 = radtodeg(atan((z - A3(3))/(y - A3(2))));
Phi04 = radtodeg(atan((z - A4(3))/(y - A4(2))));

% Ângulo de rotação dos motores
L1 = L01;
L2 = L02;
L3 = L03;
L4 = L04;
B01 = radtodeg((-1/r)*(L1-L01));
B02 = radtodeg((-1/r)*(L1-L01));
B03 = radtodeg((-1/r)*(L1-L01));
B04 = radtodeg((-1/r)*(L1-L01));
```

APÊNDICE B – Programa para a simulação em MATLAB: Simulação de linha reta

```

T = 0:0.1:1;

%Determinando a trajetória
Z = 150*(1 - T);
X = zeros(1,length(T));
Y = zeros(1,length(T));

%Comprimento dos cabos ao longo da trajetória
L1 = sqrt((X - A1(1)).^2 + (Y - A1(2)).^2 + (Z - A1(3)).^2);
L2 = sqrt((X - A2(1)).^2 + (Y - A2(2)).^2 + (Z - A2(3)).^2);
L3 = sqrt((X - A3(1)).^2 + (Y - A3(2)).^2 + (Z - A3(3)).^2);
L4 = sqrt((X - A4(1)).^2 + (Y - A4(2)).^2 + (Z - A4(3)).^2);

%Ângulos dos cabos ao longo da trajetória
Tetha1 = radtodeg(atan((Y(:) - A1(2))./(X(:) - A1(1)))); %ângulo entre o eixo Z e o segmento 1-2
Tetha2 = radtodeg(atan((Y(:) - A2(2))./(X(:) - A2(1)))); %ângulo entre o eixo Z e o segmento 2-3
Tetha3 = radtodeg(atan((Y(:) - A3(2))./(X(:) - A3(1)))); %ângulo entre o eixo Z e o segmento 3-4
Tetha4 = radtodeg(atan((Y(:) - A4(2))./(X(:) - A4(1)))); %ângulo entre o eixo Z e o segmento 4-1
Phi1 = radtodeg(atan((Z(:) - A1(3))./(Y(:) - A1(2)))); %ângulo entre o eixo X e o segmento 1-2
Phi2 = radtodeg(atan((Z(:) - A2(3))./(Y(:) - A2(2)))); %ângulo entre o eixo X e o segmento 2-3
Phi3 = radtodeg(atan((Z(:) - A3(3))./(Y(:) - A3(2)))); %ângulo entre o eixo X e o segmento 3-4
Phi4 = radtodeg(atan((Z(:) - A4(3))./(Y(:) - A4(2)))); %ângulo entre o eixo X e o segmento 4-1

%Ângulo de rotação dos motores ao longo da trajetória
B1 = radtodeg((-1/r)*(L1-L01));
B2 = radtodeg((-1/r)*(L2-L01));
B3 = radtodeg((-1/r)*(L3-L01));
B4 = radtodeg((-1/r)*(L4-L01));

%Simulação do movimento descrito
figure
for t=1:length(T);
    cable1x = [-Q/2 X(t)];
    cable1y = [-Q/2 Y(t)];
    cable1z = [H Z(t)];
    cable2x = [Q/2 X(t)];
    cable2y = [-Q/2 Y(t)];
    cable2z = [H Z(t)];
    cable3x = [Q/2 X(t)];
    cable3y = [Q/2 Y(t)];
    cable3z = [H Z(t)];
    cable4x = [-Q/2 X(t)];
    cable4y = [Q/2 Y(t)];
    cable4z = [H Z(t)];
    EEx = X(t);
    EEy = Y(t);
    EEz = Z(t);
    eex(t) = EEx;
    eey(t) = EEy;
    eez(t) = EEz;
    plot3 (cable1x, cable1y, cable1z, 'black', cable2x, cable2y, cable2z,
    'black', cable3x, cable3y, cable3z, 'black', cable4x, cable4y, cable4z,
    'black', EEx, EEy, EEz, 'o','MarkerEdgeColor', 'blue', 'MarkerFaceColor',
    'blue');
    grid on;
    title('Trajetória');
    xlabel('x (mm)');
    ylabel('y (mm)');
end

```

```

zlabel('z (mm)');
if t==1
    hold;
else
end
axis([-150 150 -150 150 0 150]);
axis normal
%M(:,t)=getframe;
end

%Gráfico dos comprimentos dos cabos
figure
cc1 = plot(T, L1);
cc1.Color = 'red';
hold;
cc2 = plot(T, L2);
cc2.Color = 'green';
cc3 = plot(T, L3);
cc3.Color = 'blue';
cc4 = plot(T, L4);
cc4.Color = 'magenta';
title('Comprimentos dos Cabos');
ylabel('Comprimento (mm)');
xlabel('Tempo');
grid on;
legend('Cabo 1', 'Cabo 2', 'Cabo 3', 'Cabo 4', 'L1', 'L2', 'L3', 'L4');

%Gráficos dos ângulos dos Cabos
figure
subplot(2,1,1);
th1 = plot(T, Tetha1);
th1.Color = 'red';
hold;
th2 = plot(T, Tetha2);
th2.Color = 'green';
th3 = plot(T, Tetha3, '--');
th3.Color = 'blue';
th4 = plot(T, Tetha4, '--');
th4.Color = 'magenta';
title('Ângulo \theta');
ylabel('\theta (°)');
xlabel('Tempo');
grid on;
legend('\theta 1', '\theta 2', '\theta 3', '\theta 4', 'Tetha1', 'Tetha2', 'Tetha3', 'Tetha4');
subplot(2,1,2);
ph1 = plot(T, Phil);
ph1.Color = 'red';
hold;
ph2 = plot(T, Phi2, '--');
ph2.Color = 'green';
ph3 = plot(T, Phi3);
ph3.Color = 'blue';
ph4 = plot(T, Phi4, '--');
ph4.Color = 'magenta';
title('Ângulo \phi');
ylabel('\phi (°)');
xlabel('Tempo');
grid on;
legend('\phi 1', '\phi 2', '\phi 3', '\phi 4', 'Phil', 'Phi2', 'Phi3', 'Phi4');

```

```
%Gráfico dos ângulos de rotação dos motores
figure
beta1 = plot (T, B1);
beta1.Color = 'red';
hold;
beta2 = plot (T, B2);
beta2.Color = 'green';
beta3 = plot (T, B3);
beta3.Color = 'blue';
beta4 = plot (T, B4);
beta4.Color = 'magenta';
title('Ângulo de Rotação dos Motores');
ylabel('\beta (\circ)');
xlabel('Tempo');
grid on;
legend ('\beta 1', '\beta 2', '\beta 3', '\beta 4', 'B1', 'B2', 'B3', 'B4');

%Representação gráfica da trajetória do EE
figure
tr = plot3 (X,Y,Z);
tr.Color = 'blue';
title('Trajetória');
xlabel('x (mm)');
ylabel('y (mm)');
zlabel('z (mm)');
axis ([-150 150 -150 150 0 150]);
axis normal
grid on;
```

APÊNDICE C – Programa para a simulação em MATLAB: Simulação de espiral

```

T = 0:0.01:1;

%Determinando a trajetória
Z = 150*(1-T);
X = 30*sin(6*pi*T);
Y = 30*cos(6*pi*T);

%Comprimento dos cabos ao longo da trajetória
L1 = sqrt((X - A1(1)).^2 + (Y - A1(2)).^2 + (Z - A1(3)).^2);
L2 = sqrt((X - A2(1)).^2 + (Y - A2(2)).^2 + (Z - A2(3)).^2);
L3 = sqrt((X - A3(1)).^2 + (Y - A3(2)).^2 + (Z - A3(3)).^2);
L4 = sqrt((X - A4(1)).^2 + (Y - A4(2)).^2 + (Z - A4(3)).^2);

%Ângulos dos cabos ao longo da trajetória
Tetha1 = radtodeg(atan((Y(:) - A1(2))./(X(:) - A1(1)))); %ângulo entre o eixo Z e o cabo 1
Tetha2 = radtodeg(atan((Y(:) - A2(2))./(X(:) - A2(1)))); %ângulo entre o eixo Z e o cabo 2
Tetha3 = radtodeg(atan((Y(:) - A3(2))./(X(:) - A3(1)))); %ângulo entre o eixo Z e o cabo 3
Tetha4 = radtodeg(atan((Y(:) - A4(2))./(X(:) - A4(1)))); %ângulo entre o eixo Z e o cabo 4
Phi1 = radtodeg(atan((Z(:) - A1(3))./(Y(:) - A1(2)))); %ângulo entre o eixo X e o cabo 1
Phi2 = radtodeg(atan((Z(:) - A2(3))./(Y(:) - A2(2)))); %ângulo entre o eixo X e o cabo 2
Phi3 = radtodeg(atan((Z(:) - A3(3))./(Y(:) - A3(2)))); %ângulo entre o eixo X e o cabo 3
Phi4 = radtodeg(atan((Z(:) - A4(3))./(Y(:) - A4(2)))); %ângulo entre o eixo X e o cabo 4

%Ângulo de rotação dos motores ao longo da trajetória
B1 = radtodeg((-1/r)*(L1-L01));
B2 = radtodeg((-1/r)*(L2-L01));
B3 = radtodeg((-1/r)*(L3-L01));
B4 = radtodeg((-1/r)*(L4-L01));

%Simulação do movimento descrito
figure
for t=1:length(T);
    cable1x = [-Q/2 X(t)];
    cable1y = [-Q/2 Y(t)];
    cable1z = [H Z(t)];
    cable2x = [Q/2 X(t)];
    cable2y = [-Q/2 Y(t)];
    cable2z = [H Z(t)];
    cable3x = [Q/2 X(t)];
    cable3y = [Q/2 Y(t)];
    cable3z = [H Z(t)];
    cable4x = [-Q/2 X(t)];
    cable4y = [Q/2 Y(t)];
    cable4z = [H Z(t)];
    EEx = X(t);
    EEy = Y(t);
    EEz = Z(t);
    eex(t) = EEx;
    eey(t) = EEy;
    eez(t) = EEz;
    plot3 (cable1x, cable1y, cable1z, 'black', cable2x, cable2y, cable2z,
    'black', cable3x, cable3y, cable3z, 'black', cable4x, cable4y, cable4z,
    'black', EEx, EEy, EEz, 'o','MarkerEdgeColor', 'blue', 'MarkerFaceColor',
    'blue');
    grid on;
    title('Trajetória');
    xlabel('x (mm)');
    ylabel('y (mm)');
end

```

```

zlabel('z (mm)');
axis([-150 150 -150 150 0 150]);
axis normal
M(:,t)=getframe;
end

%Gráfico dos comprimentos dos cabos
figure
cc1 = plot(T, L1);
cc1.Color = 'red';
hold;
cc2 = plot(T, L2);
cc2.Color = 'green';
cc3 = plot(T, L3);
cc3.Color = 'blue';
cc4 = plot(T, L4);
cc4.Color = 'magenta';
title('Comprimentos dos Cabos');
ylabel('Comprimento (mm)');
xlabel('Tempo');
grid on;
legend('Cabo 1','Cabo 2','Cabo 3','Cabo 4','L1','L2','L3','L4');

%Gráficos dos ângulos dos Cabos
figure
subplot(2,1,1);
th1 = plot(T, Tetha1);
th1.Color = 'red';
hold;
th2 = plot(T, Tetha2);
th2.Color = 'green';
th3 = plot(T, Tetha3);
th3.Color = 'blue';
th4 = plot(T, Tetha4);
th4.Color = 'magenta';
title('Ângulo \theta');
ylabel('\theta (°)');
xlabel('Tempo');
grid on;
legend('\theta 1','\theta 2','\theta 3','\theta 4','Tetha1','Tetha2','Tetha3','Tetha4');
subplot(2,1,2);
ph1 = plot(T, Phil);
ph1.Color = 'red';
hold;
ph2 = plot(T, Phi2, '--');
ph2.Color = 'green';
ph3 = plot(T, Phi3);
ph3.Color = 'blue';
ph4 = plot(T, Phi4, '--');
ph4.Color = 'magenta';
title('Ângulo \phi');
ylabel('\phi (°)');
xlabel('Tempo');
grid on;
legend('\phi 1','\phi 2','\phi 3','\phi 4','Phil','Phi2','Phi3','Phi4');

%Gráfico dos ângulos de rotação dos motores
figure
beta1 = plot(T, B1);
beta1.Color = 'red';

```

```

hold;
beta2 = plot (T, B2);
beta2.Color = 'green';
beta3 = plot (T, B3);
beta3.Color = 'blue';
beta4 = plot (T, B4);
beta4.Color = 'magenta';
title('Ângulo de Rotação dos Motores');
ylabel('\beta (\circ)');
xlabel('Tempo');
grid on;
legend ('\beta 1', '\beta 2', '\beta 3', '\beta 4', 'B1', 'B2', 'B3', 'B4');

%Representação gráfica da trajetória do EE
figure
tr = plot3 (X,Y,Z);
tr.Color = 'blue';
title('Trajetória');
xlabel('x (mm)');
ylabel('y (mm)');
zlabel('z (mm)');
axis ([-150 150 -150 150 0 150]);
axis normal
grid on;

```

APÊNDICE D – Programa para a simulação em MATLAB: Cinemática direta

```
for i=1:length(L1);
    l1=L1(i);
    l2=L2(i);
    l3=L3(i);
    A = [A1; A2; A3].';
    At = A';
    a1 = norm(A1);
    a2 = norm(A2);
    a3 = norm(A3);
    e = [1 1 1].';
    B = [a1.^2 - l1.^2; a2.^2 - l2.^2; a3.^2 - l3.^2];
    u = inv(At)*e;
    v = inv(At)*B;
    p1 = (((2-u'*v) + ((2 - u'*v)^3 - (u'*u)*(v'*v))^(1/2))/(2*u'*u))*u +
    v./2;
    p2 = (((2-u'*v) - ((2 - u'*v)^2 - (u'*u)*(v'*v))^(1/2))/(2*u'*u))*u +
    v./2;
    p2x(i) = p1;
    p2y(i) = p2;
    p2z(i) = p2;
end

P2 = [p2x; p2y; p2z];
ee = [eex; eey; eez];
dif = P2-ee;

figure
cd = plot3 (p2x, p2y, p2z, '.');
cd.Color = 'red';
xlabel('x (mm)');
ylabel('y (mm)');
zlabel('z (mm)');
hold;
tr = plot3 (X,Y,Z);
tr.Color = 'blue';
xlabel('x (mm)');
ylabel('y (mm)');
zlabel('z (mm)');
axis ([-150 150 -150 150 0 150]);
axis normal
grid on;
legend ('Trajetória calculada', 'Trajetória real', 'cd', 'tr');
title ('Trajetória Calculada x Real');
```

APÊNDICE E – Código em Arduino para o funcionamento do protótipo

```
#include <AccelStepper.h>
#include <math.h>

//Nº de passos para 360° no eixo do motor
const int stepsPerRevolution = 4100;

// SENTIDO ANTI HORÁRIO = + POSITIVO O VALOR DE GIRO
// SENTIDO HORÁRIO = - NEGATIVO O VALOR DE GIRO

//Inicialização dos motores de passo nas portas digitais passadas como parâmetro
AccelStepper Motor_A1(8,6,8,7,9);
AccelStepper Motor_A2(8,10,12,11,13);
AccelStepper Motor_A3(8,2,4,3,5);
AccelStepper Motor_A4(8,24,28,26,30);

//Dados Gerais do CDPR
//Dimensão da base
double Q = 400;
//Altura
double H = 200;
//Dimensão das polias
double r = 10;

//Coordenadas dos pontos de ancoragem
double A1x = -Q/2;
double A1y = -Q/2;
double A1z = H;
double A2x = Q/2;
double A2y = -Q/2;
double A2z = H;
double A3x = Q/2;
double A3y = Q/2;
double A3z = H;
double A4x = -Q/2;
double A4y = Q/2;
double A4z = H;

//Variáveis dinâmicas de posição do end effector
double x = 0;
double y = 0;
double z = H;

// Comprimento dos cabos na posição inicial
double L01 = sqrt(pow((x - A1x),2) + pow((y - A1y),2) + pow((z - A1z),2));
double L02 = sqrt(pow((x - A2x),2) + pow((y - A2y),2) + pow((z - A2z),2));
double L03 = sqrt(pow((x - A3x),2) + pow((y - A3y),2) + pow((z - A3z),2));
double L04 = sqrt(pow((x - A4x),2) + pow((y - A4y),2) + pow((z - A4z),2));
```

```

//Variáveis do tipo string para tratamento e captura das coordenadas de forma separada
String ax;
String ay;
String az;

void setup()
{
    //Define entrada para visualização no plotter serial
    Serial.begin(9600);
    Serial.println("Digite as coordenadas de destino do end effector: ");

    //Determina a Velocidade inicial dos motores
    Motor_A1.setMaxSpeed(1000.0);
    Motor_A1.setAcceleration(1000.0);
    Motor_A1.setSpeed(1000);
    Motor_A1.moveTo(4100); // 2048 = 1 revolution

    Motor_A2.setMaxSpeed(1000.0);
    Motor_A2.setAcceleration(1000.0);
    Motor_A2.setSpeed(1000);
    Motor_A2.moveTo(4100); // 2048 = 1 revolution

    Motor_A3.setMaxSpeed(1000.0);
    Motor_A3.setAcceleration(1000.0);
    Motor_A3.setSpeed(1000);
    Motor_A3.moveTo(4100); // 2048 = 1 revolution

    Motor_A4.setMaxSpeed(1000.0);
    Motor_A4.setAcceleration(1000.0);
    Motor_A4.setSpeed(1000);
    Motor_A4.moveTo(4100); // 2048 = 1 revolution
}

void loop()
{
    //Processo Iterativo de Captura de Coordenadas
    while(Serial.available())
    {
        //Variável de captura das coordenadas - Formato [sinal]XXX[sinal]YYY[sinal]ZZZ
        String inputString = Serial.readString();
        ax = inputString.substring(0,4);
        Serial.println("\nX = " + ax);
        ay = inputString.substring(4,8);
        Serial.println("Y = " + ay);
        az = inputString.substring(8,12);
        Serial.println("Z = " + az);
        x = ax.toDouble();
        y = ay.toDouble();
        z = az.toDouble();
    }
}

```

```

//Processo Iterativo de Cálculo

// Comprimento dos cabos na posição especificada
double L1 = sqrt(pow((x - A1x),2) + pow((y - A1y),2) + pow((z - A1z),2));
double L2 = sqrt(pow((x - A2x),2) + pow((y - A2y),2) + pow((z - A2z),2));
double L3 = sqrt(pow((x - A3x),2) + pow((y - A3y),2) + pow((z - A3z),2));
double L4 = sqrt(pow((x - A4x),2) + pow((y - A4y),2) + pow((z - A4z),2));

//ângulo de giro dos motores para posicionamento do end effector
double Beta1 = ((1/r)*(L01 - sqrt(pow((x - A1x),2) + pow((y - A1y),2) + pow((z - A1z),2))))*(180/PI);
double Beta2 = ((1/r)*(L02 - sqrt(pow((x - A2x),2) + pow((y - A2y),2) + pow((z - A2z),2))))*(180/PI);
double Beta3 = ((1/r)*(L03 - sqrt(pow((x - A3x),2) + pow((y - A3y),2) + pow((z - A3z),2))))*(180/PI);
double Beta4 = ((1/r)*(L04 - sqrt(pow((x - A4x),2) + pow((y - A4y),2) + pow((z - A4z),2))))*(180/PI);

aciona_motores(Beta1,Beta2,Beta3,Beta4);
}

void aciona_motores(double Beta1, double Beta2, double Beta3, double Beta4)
{
    double Pas1, Pas2, Pas3, Pas4;

    //Conversão entre os ângulos de polia calculados para a quantidade de passos
    Pas1 = ang2pas(Beta1);
    Pas2 = ang2pas(Beta2);
    Pas3 = ang2pas(Beta3);
    Pas4 = ang2pas(Beta4);

    // Gira os motores com base no valor dos ângulos calculados para o ponto destino do end effector

    Motor_A1.moveTo(Pas1);
    Motor_A1.run();
    Motor_A2.moveTo(Pas2);
    Motor_A2.run();
    Motor_A3.moveTo(Pas3);
    Motor_A3.run();
    Motor_A4.moveTo(Pas4);
    Motor_A4.run();
}

double ang2pas(double Beta)
{
    return ((Beta*4100)/360);
}

```