

Desenvolvimento de um Sistema de Controle em Malha Aberta para Protótipo de SCARA

Eduardo Lazzari

Instituto Federal de Educação, Ciência
e Tecnologia do Rio Grande do Sul - IFRS
Av. São Vicente, 785,
Farroupilha, RS, Brasil
eduardolazzari.el@gmail.com

Gustavo Künzel

Instituto Federal de Educação, Ciência
e Tecnologia do Rio Grande do Sul - IFRS
Av. São Vicente, 785,
Farroupilha, RS, Brasil
gustavo.kunzel@farroupila.ifrs.edu.br

Resumo—É inegável que o aumento do uso de robôs contribuiu para o crescimento substancial da produtividade. Baseado nisto, e na importância dos manipuladores robóticos para a indústria, foi proposto o desenvolvimento de um sistema de acionamento e posicionamento para um protótipo de SCARA, utilizando motores de passo para movimentação dos ângulos variáveis e um servo motor para ativação da ferramenta. Neste trabalho de Conclusão de Curso II foi desenvolvido o equacionamento da cinemática direta e inversa do manipulador. A partir destes dados, foi proposto um código base para encontrar o espaço de trabalho do robô, bem como sua região operacional. Com estas etapas concluídas iniciou-se a fabricação do protótipo realizada com auxílio de impressora 3D. Também é relatado de forma detalhada como funciona o protocolo de comunicação entre os componentes e como o operador pode manipular o protótipo. Por fim, são efetuados testes e ensaios de precisão utilizando o protótipo e realizada uma análise de desempenho geral.

Palavras-chave—SCARA, cinemática, robótica

I. INTRODUÇÃO

Em 1978 surgiu a terceira geração dos robôs industriais, caracterizados pela maior capacidade de interação com o operador, pela habilidade de se reprogramar, ainda que de forma limitada, para executar diferentes tarefas, e trazendo consigo um novo atributo, os controladores dedicados, também conhecidos como computadores. Estas novas particularidades fizeram com que bilhões de dólares fossem investidos por empresas em todo mundo para automatizar tarefas básicas em suas linhas de produção, como pintura, solda, movimentação de componentes e montagem, resultando em um aumento de 80 % nas vendas de robôs industriais em comparação com os anos anteriores [1].

Neste mesmo ano, uma nova estrutura cinemática é proposta pelo cientista japonês Hiroshi Makino. Tal estrutura era composta por três juntas de revolução com eixos paralelos e uma junta prismática situada no final da cadeia cinemática. O manipulador originado desta arquitetura foi denominado SCARA, acrônimo para *Selective Compliance Assembly Robot Arm*, devido a sua montagem de conformidade seletiva, onde apresenta alta rigidez na direção vertical e conformidade na direção horizontal [2].

Esta configuração com quatro graus de liberdade, representada genericamente pela Fig. 1, possibilita realizar posicionamentos planos devido ao movimento vertical da ferramenta.

Em razão desta característica, o SCARA era adequado para ser empregado em tarefas específicas, como na produção de bens de consumo, principalmente no mercado eletrônico, e em linhas de montagem, onde aproximadamente 80 % das tarefas de montagem eram realizadas por um movimento unidirecional, concedido por este manipulador robótico [3] [4].

O aumento do uso de robôs industriais está diretamente associado a aumentos na produtividade do trabalho e a reduções nos preços de produção. À medida que novos recursos são desenvolvidos e implantados em robôs industriais, mais substancial é a contribuição para o crescimento econômico da unidade que está usufruindo de tais tecnologias [5]. Desta forma, tanto o estudo quanto o desenvolvimento de manipuladores robóticos são importantes para a indústria.

Este trabalho tem como objetivo geral desenvolver um sistema de posicionamento e acionamento em malha aberta aplicado a um protótipo de SCARA. Para isso, será desenvolvida toda a cinemática do manipulador, tanto direta quanto inversa, onde posteriormente os cálculos resultantes serão testados a fim de comprovar a funcionalidade matemática do sistema. A partir da cinemática direta será desenvolvido um algoritmo para obtenção do espaço de trabalho do robô, e por fim será confeccionado o protótipo do robô e desenvolvido um ambiente gráfico no Matlab para poder operar o manipulador.

Este projeto está dividido da seguinte forma: Na seção II é realizada uma revisão bibliográfica abrangendo assuntos relacionados à robótica. Já na seção III é desenvolvida a cinemática do protótipo de manipulador SCARA a partir de sua estrutura mecânica. Na seção IV é abordada a proposta do trabalho, apresentando o protótipo do robô e o seu espaço de trabalho. Também são exibidos os componentes do sistema e como eles estão interconectados. Na seção V é demonstrado como o projeto foi executado, detalhando através de fluxogramas como os sistemas de comunicação operam. Por fim são apresentados testes e ensaios realizados para validar o total funcionamento do sistema.

II. FUNDAMENTAÇÃO TEÓRICA

A. Fundamentos de Robótica

A cinemática é responsável por descrever os movimentos de um manipulador sem considerar as forças e torques



Figura 1. Robô SCARA All-In-One Epson Synthix T3 [6].

responsáveis por este movimento. A análise cinemática de um manipulador robótico é dividida em cinemática direta e inversa. A cinemática direta mostra como determinar a posição e orientação do efetor final em termos das variáveis articulares, enquanto a cinemática inversa tem a função de encontrar as variáveis articulares em termos da posição e orientação do efetor final [7].

Em [8] é introduzido o conceito das matrizes de Denavit-Hartenberg. Esta definição é amplamente usada em robótica para facilitar os cálculos cinemáticos referentes a determinado manipulador, de acordo com sua estrutura cinemática. Do ponto de vista cinemático, robôs estão conectados através de cadeias cinemáticas. As juntas que compõem esta cadeia cinemática podem ser rotacionais ou prismáticas. O grande problema destas estruturas, é obter as coordenadas cartesianas do final de toda cadeia cinemática em relação a base do robô. Este problema pode ser resolvido com a aplicação de conceitos geométricos elementares. Tais conceitos são descritos em [9], onde são apresentadas as principais transformações, isto é, movimentos no espaço, de vetores ou de objetos. Uma transformação pode ser uma translação pura, uma rotação pura sobre um eixo ou uma combinação de translação e rotação. Se um vetor se move no espaço sem qualquer mudança em sua orientação, então esta é uma transformação de translação pura. Neste caso, a única coisa que muda é o local da origem do vetor. Por outro lado, temos a rotação, que tem como objetivo alterar a direção do vetor ou do objeto, sem mudar sua magnitude. Já as transformações que envolvem ambos os casos tem como objetivo redefinir a direção e local de origem do vetor.

O método clássico de atribuição dos parâmetros de Denavit-Hartenberg é atribuir sistematicamente um quadro de coordenadas para cada elo, levando em consideração as restrições impostas por cada um dos mesmos [10]. A definição dos parâmetros de cada ponto de referência relacionado a um determinado elo pode ser encontrado seguindo as regras descritas na Tabela I, onde basicamente, i representa o elo atual, a_i

significa o comprimento do elo, θ_i simboliza o ângulo de junta do elo, d_i retrata o deslocamento do elo em relação a outro elo, e por fim, α_i descreve a torção entre um elo e outro. Os parâmetros z e x são parte do sistema de coordenadas, e a combinação destes dois parâmetros dá origem ao parâmetro j , que representa todo o sistema de coordenadas [11].

Tabela I
CONVENÇÃO DENAVIT-HARTENBERG

Parâmetros	Descrição
a_i	transladar o eixo de coordenadas j_{i+1} ao longo de x_{i+1} até a intersecção entre z_i e x_{i+1} .
α_i	rotacionar o eixo z_i em torno de x_{i+1} até ficar paralelo ao eixo z_{i+1} .
d_i	transladar o eixo de coordenadas j_i ao longo de z_i até a intersecção entre z_i e x_{i+1} .
θ_i	rotacionar o eixo x_i em torno de z_i até ficar paralelo ao eixo x_{i+1} .

O espaço de trabalho de um manipulador pode ser definido como o volume do espaço que o efetor final do manipulador pode alcançar [12]. O tamanho e a forma resultante do espaço de trabalho depende da estrutura do robô, tipos de juntas, e também da quantidade de graus de liberdade. Alguns espaços de trabalho mais simples podem ser limitados quase inteiramente a um plano horizontal. Outros podem ser cilíndricos e/ou esféricos. Algumas configurações de manipuladores apresentam um espaço de trabalho com formas complexas. Ao escolher um braço robótico para determinado propósito industrial, é importante que o espaço de trabalho seja grande o suficiente para abranger todos os pontos necessários, por outro lado, um braço robótico muito maior que o necessário acaba sendo subutilizado.

B. Trabalhos Relacionados

Manipuladores robóticos de vários tipos, tamanhos e formatos tornaram-se parte fundamental da indústria devido a sua ampla gama de aplicações. Em [13] é demonstrado que é possível desenvolver um manipulador robótico de baixo custo com características semelhantes as dos robôs industriais, atingindo satisfatória precisão no controle da posição e velocidade dos atuadores.

O trabalho [14] apresenta o desenvolvimento de um robô planar criado com auxílio de uma impressora 3D. Ressalta que, com o uso desta nova tecnologia, o design se torna de fácil fabricação, podendo alterar tamanhos e formas dos elos até atingir um resultado considerado bom, sem que o custo aumente de maneira excessiva. Entretanto, construir protótipos utilizando polímeros podem resultar em alguns problemas. São citadas as principais desvantagens mecânicas do modelo impresso em 3D, como a baixa resistência da construção, baixa capacidade de carga e, dependendo da impressora, folgas significativas que afetam a precisão do manipulador.

O trabalho de [15] tem como objetivo projetar e construir um braço robótico com quatro graus de liberdade com o intuito de servir como uma ferramenta educacional. Para isso, o processo de manufatura do robô inicia com o desenvolvimento do projeto em ambiente CAD para posteriormente ser

fabricado com auxílio de uma impressora 3D. A montagem resultante, composta pela parte mecânica e também pela parte elétrica, possibilitou a realização de testes que envolviam a movimentação dos atuadores do manipulador, e o resultado mostrou-se satisfatório dentro do propósito.

III. CINEMÁTICA DE UM ROBÔ SCARA

Nesta seção serão apresentados os cálculos envolvendo a cinemática de um manipulador SCARA. Será abordado tanto a cinemática direta quanto a cinemática inversa, utilizando como princípio a notação de Denavit-Hartenberg.

A. Cinemática Direta

Determinar os parâmetros de Denavit-Hartenberg é a primeira etapa para encontrar a cinemática do robô. Para isso, sistemas de coordenadas são fixados no manipulador, como mostrado na Fig. 2.

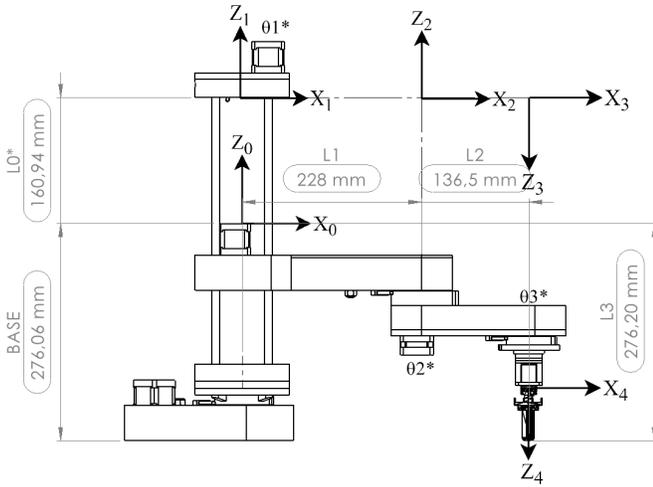


Figura 2. Sistemas de coordenadas anexados ao robô SCARA [16].

Aplicando as regras descritas na Tabela I ao protótipo do manipulador demonstrado na Fig. 2 podemos obter os parâmetros demonstrados na Tabela II.

Tabela II
CONVENÇÃO DENAVIT-HARTENBERG PARA MANIPULADOR SCARA

Elo	a_i	α_i	d_i	θ_i
0	0	0	L_0^*	0
1	L_1	0	0	θ_1^*
2	L_2	180	0	θ_2^*
3	0	0	L_3	θ_3^*

Então é necessário aplicar as matrizes de transformação homogênea [9] para cada um dos elos dispostos na Tabela II. Usamos as matrizes de rotação em x e em z , demonstradas respectivamente pelas Eq. 1 e 2.

$$Rot(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$Rot(z, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Para transladar um vetor no espaço, sem que o mesmo tenha alteração em sua orientação, utiliza-se a matriz de translação representada na Eq. 3, onde x , y e z são os três componentes de um vetor de translação puro.

$$T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Então, com base nas Eq. 1, 2 e 3 é possível calcular a matriz homogênea para cada elo, resultando nas matrizes apresentadas nas Eq. 4, 5, 6 e 7.

$$A_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_1^2 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1 * \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1 * \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A_2^3 = \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & a_2 * \cos(\theta_2) \\ \sin(\theta_2) & -\cos(\theta_2) & 0 & a_2 * \sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$A_3^4 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

A_0^1 representa a posição do elo 1 em relação ao elo 0, que neste caso é a própria base do manipulador, A_1^2 mostra a posição relativa do elo 2 em relação ao elo 1, A_2^3 demonstra a posição relativa do elo 3 em relação ao elo 2, e por fim, A_3^4 mostra a posição relativa do elo 4 em relação ao elo 3.

A transformação total entre a base do robô e o efetor final, A_0^4 , pode ser encontrada multiplicando-se as matrizes homogêneas representadas nas Eq. 4, 5, 6 e 7. Para fins de simplificação, utiliza-se $\theta_{12} = \theta_1 + \theta_2$, $\theta_{12-3} = \theta_1 + \theta_2 - \theta_3$, $c = \cos$ e $s = \sin$. Desta forma, as equações que descrevem a cinemática direta do manipulador podem ser descritas da seguinte maneira:

$$T = A_0^1 * A_1^2 * A_2^3 * A_3^4 \quad (8)$$

$$A_0^4 = \begin{bmatrix} c(\theta_{12-3}) & s(\theta_{12-3}) & 0 & a_2c(\theta_{12}) + a_1c(\theta_1) \\ s(\theta_{12-3}) & -c(\theta_{12-3}) & 0 & a_2s(\theta_{12}) + a_1s(\theta_1) \\ 0 & 0 & -1 & d_0 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

B. Cinemática Inversa

Para o cálculo da cinemática inversa, pode-se utilizar diferentes métodos para obtenção dos ângulos variáveis do manipulador. Neste caso, a matriz representada pela Eq. 9 é igualada aos ângulos de Euler, utilizando uma versão simplificada, uma vez que a ferramenta deste robô apresenta apenas rotação sobre o eixo z , e não há nenhum tipo de posicionamento da ferramenta em relação aos eixos x e y . Tal rotação é representada pelo ângulo ϕ [9].

$$A_0^4 = \begin{bmatrix} \cos(\phi) & \text{sen}(\phi) & 0 & x \\ \text{sen}(\phi) & -\cos(\phi) & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Igualando os elementos (1,4), (2,4) e (3,4) da Eq. 10 obtemos as seguintes equações:

$$x = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) \quad (11)$$

$$y = a_1 \text{sen}(\theta_1) + a_2 \text{sen}(\theta_1 + \theta_2) \quad (12)$$

$$z = d_0 - d_3 + \text{Base} \quad (13)$$

É importante notar que na Eq. 13 há a adição da constante Base que não havia sido referenciada na Tabela II por motivos de simplificação dos cálculos. Esta constante interfere apenas no resultado da posição no eixo z , então pode ser inserida neste momento. A partir das Eq. 11, 12 e 13, é possível isolar as variáveis do manipulador, representadas por d_0 , θ_1 , θ_2 e θ_3 , e encontrar as equações responsáveis pela movimentação de cada um destes ângulos.

$$\theta_1 = \text{atan2} \left[\begin{array}{l} a_1 y + a_2 \cos(\theta_2) y - a_2 x \text{sen}(\theta_2), \\ a_1 x + a_2 \cos(\theta_2) x + a_2 y \text{sen}(\theta_2) \end{array} \right] \quad (14)$$

$$\theta_2 = \pm \arccos \left[\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2} \right] \quad (15)$$

$$\theta_3 = \theta_1 + \theta_2 - \phi \quad (16)$$

$$d_0 = z + d_3 - \text{Base} \quad (17)$$

Na Eq. 14 é utilizada a função atan2 , usada em linguagens de programação, que consegue distinguir pontos opostos no círculo unitário. Esta função é uma implementação da função atan , que leva em consideração os quadrantes e os sinais das entradas x e y [17]. Já na Eq. 15 é possível verificar a possibilidade deste ângulo assumir valores positivos e negativos. Há pontos no espaço de trabalho do manipulador que só será possível alcançar utilizando o ângulo positivo. Da mesma forma, haverá situações em que apenas a utilização do ângulo negativo atingirá o ponto no espaço requerido.

O ângulo ϕ representado na Eq. 16 é determinado levando em consideração o ângulo do efetor final em relação à sua

base. Neste projeto optou-se por deixar a ferramenta perpendicular à base, fazendo com que ϕ tenha um valor igual a 180° .

IV. PROPOSTA

Nesta seção é apresentada a proposta do TCC, iniciando pelas especificações do protótipo do manipulador, para a partir dele, derivar a área de trabalho do robô, com auxílio visual de gráficos, bem como o seu espaço operacional. Além disso será descrito, através de um diagrama de blocos, como os componentes se associam um com os outros a fim de garantir o funcionamento completo do robô e do sistema proposto. Neste trabalho não será desenvolvida a geração de trajetória, uma vez que o enfoque será delimitado ao acionamento dos atuadores juntamente com uma interface gráfica.

A. Protótipo

Para o presente trabalho é utilizado um protótipo desenvolvido por [16]. O manipulador apresenta quatro graus de liberdade seguindo a topologia SCARA de construção. Cada elo, assim como sua base, podem ser fabricados utilizando uma impressora 3D. Para os outros componentes mecânicos do sistema, como rolamentos, guias lineares, parafusos e polias, foram usados modelos comerciais a fim de reduzir o custo do projeto. Em seu efetor final, apresenta uma garra, também impressa em polímero, representando a ferramenta do robô. Na Fig. 3 é retratado o projeto do manipulador.

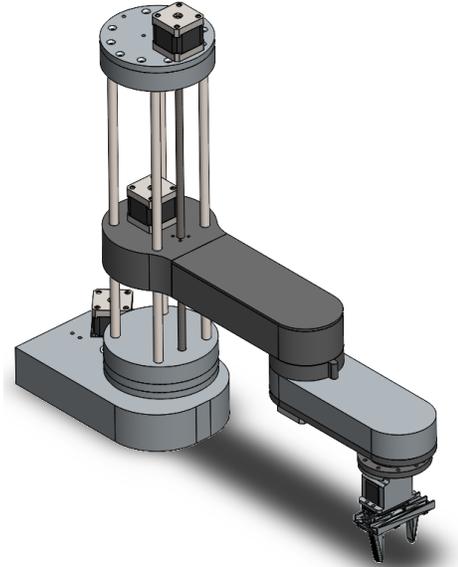


Figura 3. Projeto do protótipo do manipulador SCARA. Vista isométrica [16].

B. Espaço de Trabalho

Conhecendo as dimensões do manipulador e os ângulos mínimos e máximos que as variáveis do robô são capazes de operar, pode-se determinar o espaço de trabalho do mesmo. Na Fig. 4 é demonstrada a região de operação do robô, ou seja, o espaço que o protótipo ocupa em seu local de atuação.

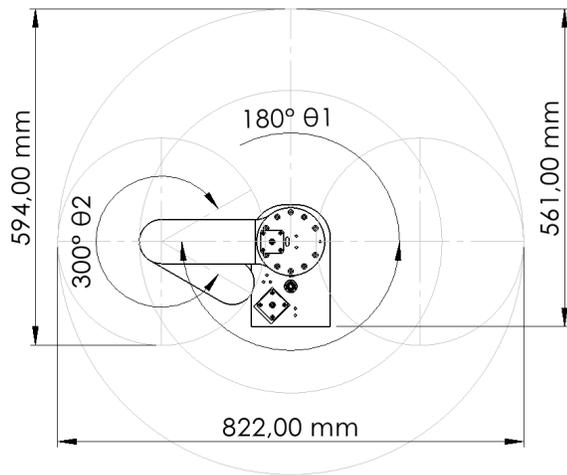


Figura 4. Ângulos θ_1 e θ_2 . Vista superior [16].

A verificação do espaço de trabalho é de fundamental importância não apenas para análise e avaliação da área em que a ferramenta pode alcançar, mas também para deduzir critérios racionais de design. A área de trabalho do robô pode ser descrita como a região que pode ser alcançada por um ponto de referência [18]. Neste caso, o ponto de referência usado é o efector final do protótipo.

Conforme apresentado na Fig. 4, é possível verificar que o ângulo θ_1 varia de 0 a 180° enquanto o ângulo θ_2 varia de 0 a 300° . A variação do ângulo θ_3 , que vai de 0 a 324° , responsável pela rotação do efector final, não altera o espaço de trabalho, então pode ser desconsiderado. Sabendo os ângulos mínimos e máximos das variáveis do sistema, bem como as medidas de cada elo, dispostos na Fig. 2, é possível desenvolver um algoritmo para obtenção do espaço de trabalho.

Há diversos métodos para determinar a área de trabalho de um determinado manipulador robótico. O modo utilizado neste trabalho envolve a cinemática direta do robô, representado na Eq. 9. O processo consiste em variar os ângulos θ_1 , θ_2 , θ_3 e d_0 de seus valores mínimos até seus valores máximos, plotando em um gráfico uma sequência de pontos em que é possível a ferramenta alcançar. Dessa forma, o resultado da área de trabalho gerada é demonstrada na Fig. 5. A área de trabalho 3D do manipulador segue o mesmo formato do gráfico representado na Fig. 5, com adição de uma profundidade de 160.9 mm no eixo z .

C. Componentes do Sistema

Todo o processo envolvendo o controle do robô é iniciado no computador. Este irá se comunicar com o microcontrolador, que por sua vez é responsável pela manipulação dos motores do braço robótico. Por fim, sensores de fim de curso anexados ao SCARA enviam sinais para o microcontrolador. Na Fig. 6 é demonstrado como os componentes se interligam.

1) *Computador*: O computador possui uma interface gráfica para o operador poder manipular o robô. Esta

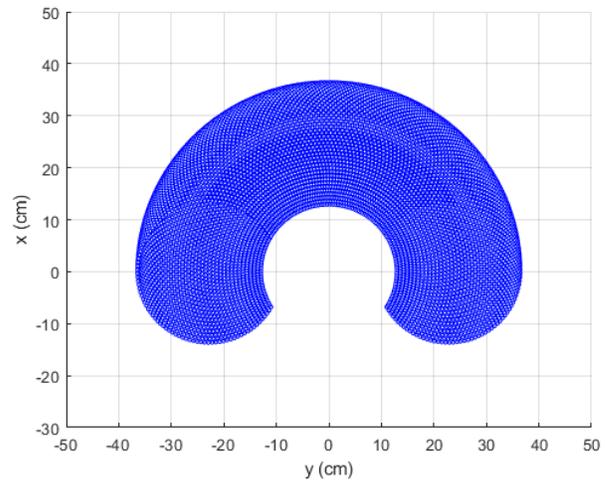


Figura 5. Espaço de trabalho. Vista superior 2D.

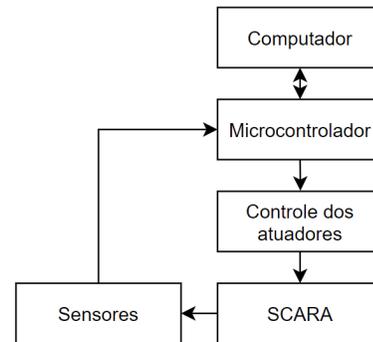


Figura 6. Diagrama de blocos do sistema.

interface é implementada no *software* Matlab. Neste programa são inseridas as equações matemáticas referentes à cinemática direta e inversa do robô, descritas previamente nas Eq. 9, 11, 12 e 13. O computador é responsável por captar a ação desejada pelo usuário, realizar os cálculos cinemáticos, e a partir deste resultado, realizar uma nova operação para computar a quantidade de passos necessária para que cada um dos atuadores atinja o ângulo requerido pelo usuário.

2) *Microcontrolador*: A plataforma de prototipagem utilizada no projeto é o *Arduino Uno* com o *chipset ATmega328*. O *Arduino* recebe um pacote de dados do computador, com a quantidade de passos que cada um dos quatro atuadores devem realizar. A partir desta informação, executa o algoritmo responsável pela movimentação dos atuadores. O microcontrolador também é responsável por receber os sinais dos sensores de fim de curso presentes no SCARA. A comunicação entre o *Arduino* e os motores do robô é intermediada por uma placa de controle. Esta placa, por sua vez, fornece ao microcontrolador a potência necessária para acionar os motores de passo, auxiliando também no controle da velocidade de cada atuador. Os motores possuem velocidade fixa definidas durante a fase de desenvolvimento do protótipo. De acordo com o

comportamento mecânico foram determinadas as velocidades para cada um dos ângulos variáveis.

3) *Atuadores*: Os atuadores responsáveis por realizar o movimento das articulações do robô, ângulos θ_1 , θ_2 , θ_3 e d_0 , são motores de passo do modelo NEMA 17. A fim de aumentar o torque exercido em cada um destes ângulos, é utilizado um sistema de redução com polias, conforme Fig. 7. No primeiro ângulo, θ_1 , responsável pela movimentação da base e que faz o maior esforço mecânico, há uma redução de 20:1 acoplada ao motor. Já no segundo ângulo rotacional, θ_2 , existe uma redução na proporção de 16:1 atado ao motor. Quanto ao último ângulo rotacional, θ_3 , encarregado de manipular a orientação da ferramenta, há uma pequena redução de 4:1. O motor responsável pelo movimento vertical, d_0 , não tem nenhum tipo de redução acoplada. Para o acionamento da ferramenta é utilizado um servo motor.

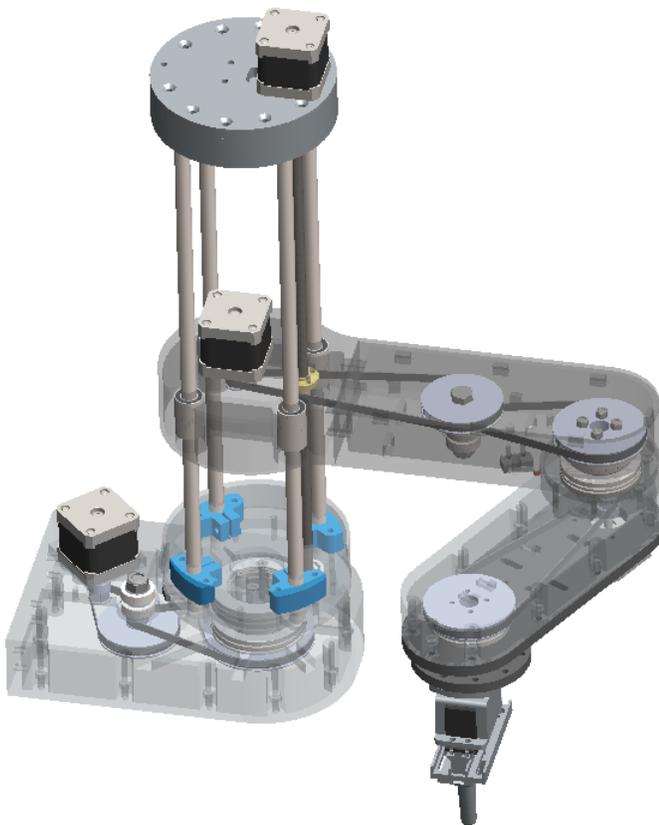


Figura 7. Representação interna do sistema de redução [16].

4) *Comunicação*: Foi desenvolvido um protocolo de comunicação serial para o envio e recebimento de dados entre o microcontrolador e o computador. Tanto o Matlab quanto o Arduino trocam informações entre si. Esta troca de informações deve ser confiável, sem erros no envio e recebimento dos pacotes de dados. Para obter um nível de confiabilidade elevado, é adicionado um quadro no pacote de dados, responsável por garantir a integridade das informações trocadas.

V. IMPLEMENTAÇÃO E RESULTADOS

Nesta seção são apresentados detalhes sobre a execução do projeto, detalhando como os componentes estão conectados entre si e especificando como a comunicação é realizada entre os elementos através de fluxogramas. Também é abordada a construção do protótipo bem como os testes e ensaios realizados para validar o funcionamento do sistema.

A. Eletrônica

O projeto faz uso de quatro motores de passo com resolução de 1.8° por passo. Para auxílio no controle dos motores são utilizados *drivers A4988*, juntamente com uma placa *CNC Shield*, conectada no Arduino. Para acionamento da garra é utilizado um servo motor modelo *MG 996r*. Os componentes encontram-se conectados conforme a Fig. 8.

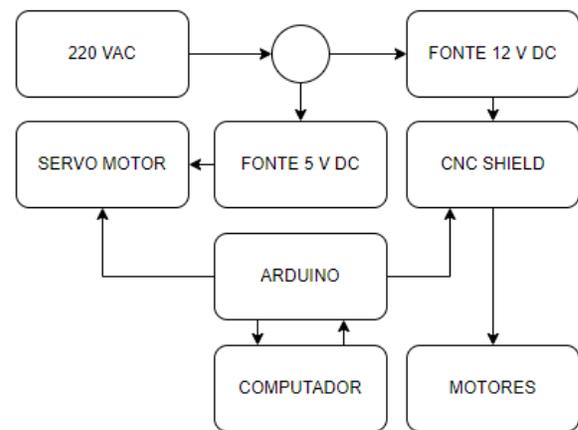


Figura 8. Diagrama de blocos das conexões dos componentes.

B. Protótipo

O protótipo foi fabricado em uma impressora 3D utilizando filamento PLA de 1.75 mm de espessura. O *software* responsável por preparar o modelo da peça para impressão foi configurado com resolução de 0.2 mm por camada e preenchimento de 30 %. O protótipo construído é mostrado na Fig. 9.

Como o protótipo construído tem reduções em suas articulações, optou-se por utilizar os *drivers A4988* na resolução de 1/4. A utilização de resoluções maiores, como 1/16 por exemplo, foi descartada devido ao baixo valor resultante da relação grau por passo, quando utilizado junto com as reduções presentes no manipulador. Ao utilizar esta resolução, um passo no motor não conseguia mover a engrenagem final, ficava apenas tensionando a correia dentada. Assim, através de testes práticos, o que se julgou melhor foi a utilização da resolução de 1/4. Desta forma, os motores ficam com uma resolução de 0.45° por passo. Já quando atrelados as reduções o ângulo θ_1 assume resolução de 0.0225° por passo, o θ_2 0.0281° por passo, o θ_3 0.1125° por passo e o ângulo linear d_0 fica com resolução de 0.01 mm por passo, proveniente do modelo da barra roscada utilizada pra realizar o movimento

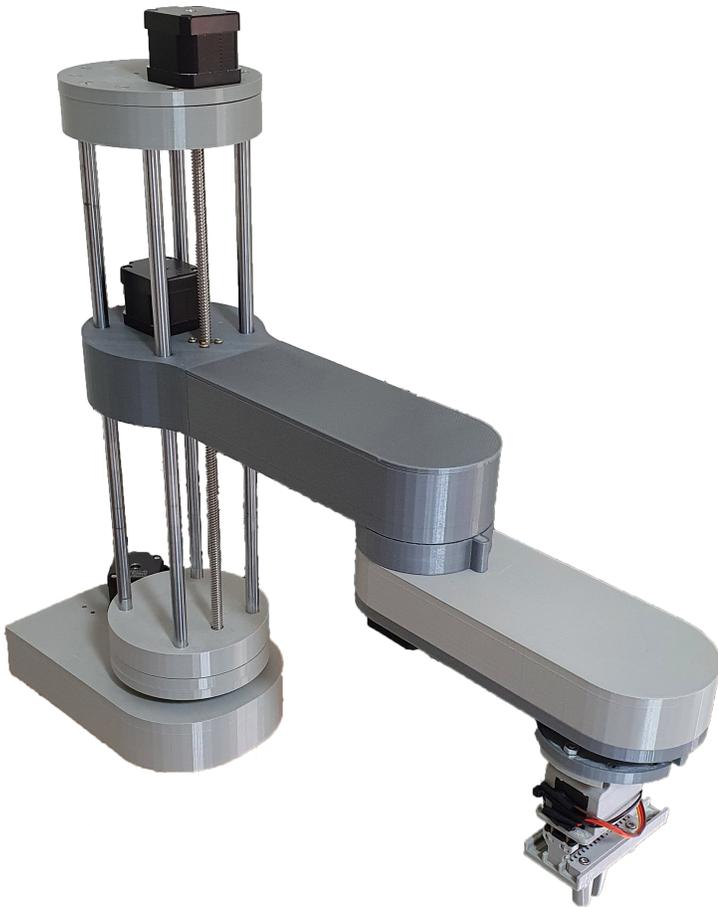


Figura 9. Protótipo de SCARA construído.

de subida e descida. A partir destes dados são calculados quantos passos devem ser dados para um determinado motor movimentar determinado ângulo.

C. Funcionamento da interface e sistema de comunicação

Após ser energizado, o sistema segue o fluxograma descrito na Fig. 10. O usuário só terá acesso à movimentação do protótipo depois do mesmo ter realizado a calibração dos quatro motores. Caso ocorra algum erro de *timeout* no recebimento da confirmação do fim da função *auto homing* por parte do Matlab, a interface não será inicializada. Prováveis causas para este erro são a utilização da porta COM errada para realizar a comunicação serial, ou até mesmo problemas de energização do circuito, impossibilitando que os motores acionem as chaves fim de curso.

A função *auto homing* desenvolvida no Arduino faz com que todos os motores acionem seus respectivos *switches*, um de cada vez. Concluída esta etapa é então enviada a informação que o movimento foi finalizado através da porta serial, habilitando assim o programa.

Ao habilitar o programa o usuário pode optar por manipular o protótipo através de duas formas diferentes: utilizando a

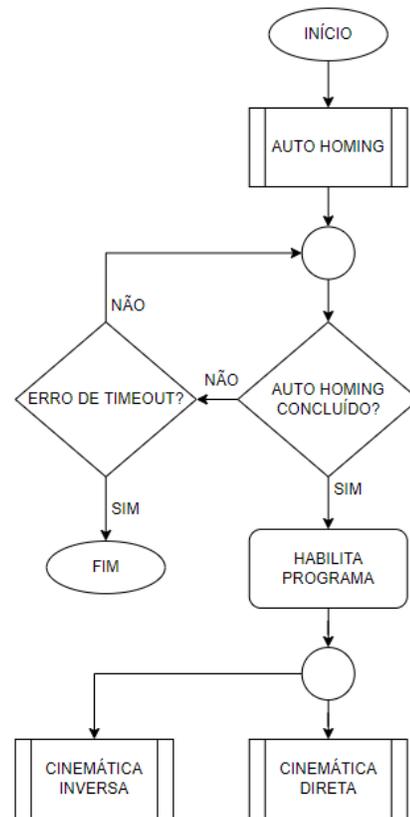


Figura 10. Fluxograma geral.

cinemática direta ou utilizando a cinemática inversa. O operador sempre poderá escolher mudar o método de manipulação ao terminar cada movimento. Ou seja, é possível intercalar movimentos realizados através da cinemática direta e inversa.

Para o movimento utilizando a cinemática direta o operador deve deslizar as barras mostradas na Fig. 11. As barras já servem como delimitação dos ângulos máximos e mínimos de cada uma das variáveis do sistema. As barras foram postas de forma que ao soltar o cursor em algum ponto, elas sempre assumam um valor múltiplo da resolução do ângulo que está sendo variado. Por isso, por exemplo, que o θ_2 mostrado na Fig. 11 apresenta um valor inicial de -149.9978° e não de -150° . Isto foi feito a fim de tornar o sistema mais confiável, fazendo com que a quantidade de passos que é enviado ao Arduino assumam sempre valores inteiros. Caso o usuário queria inserir manualmente o valor então a quantidade de passos será arredondada para o número mais próximo, porém tal ação pode afetar o desempenho do robô.

Se o usuário optar por movimentar o protótipo através da cinemática direta, o sistema funcionará seguindo o fluxograma demonstrado na Fig. 12.

A segunda forma de movimentar o robô é fazendo uso da cinemática inversa. Para tal, o usuário deverá inserir uma posição espacial x, y, z nos seus respectivos campos de acordo com a Fig. 13. Neste caso, após a inserção dos valores,

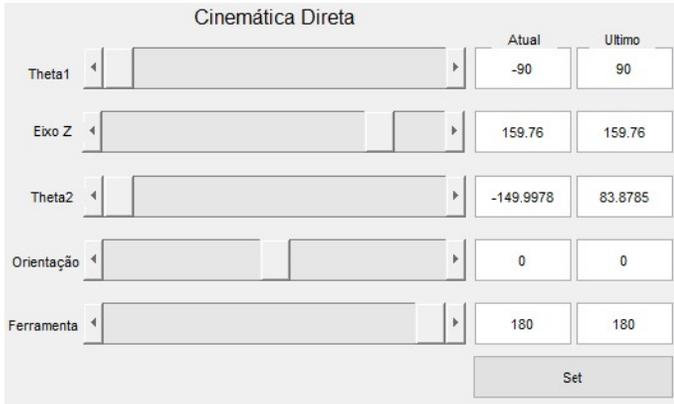


Figura 11. Interface cinemática direta.

é feita uma análise e o valor digitado é substituído pelo valor múltiplo da combinação da resolução dos motores mais próximo possível, a fim de que todas as variáveis a serem movimentadas possam assumir uma quantidade de passos que tenha um valor inteiro. Para o usuário saber quais pontos de coordenadas são válidos, a área de trabalho é disposta na interface, seguindo a Fig. 5.



Figura 13. Interface cinemática inversa.

Quando o usuário opta por movimentar o manipulador através da cinemática inversa, o sistema funciona seguindo o fluxograma representado na Fig. 14.

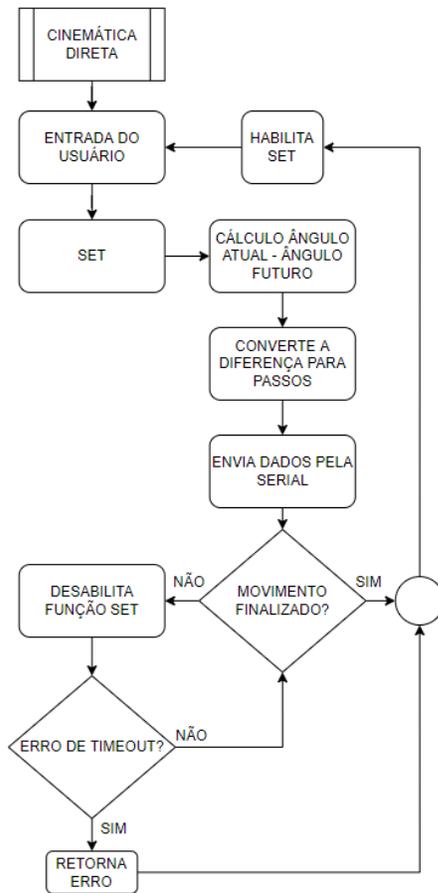


Figura 12. Fluxograma cinemática direta.

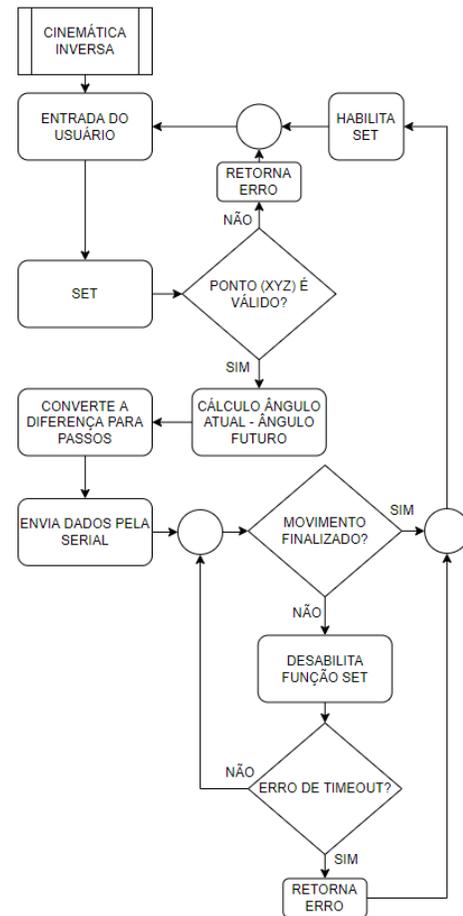


Figura 14. Fluxograma cinemática inversa.

Outra função da interface é a possibilidade do usuário salvar/ posições e fazer com que o manipulador execute o movimento na sequência em que foram salvas e em formato de *looping*, ou seja, quando chega na última posição retorna à primeira e inicia todo o ciclo novamente. Esta particularidade é demonstrada na Fig. 15.

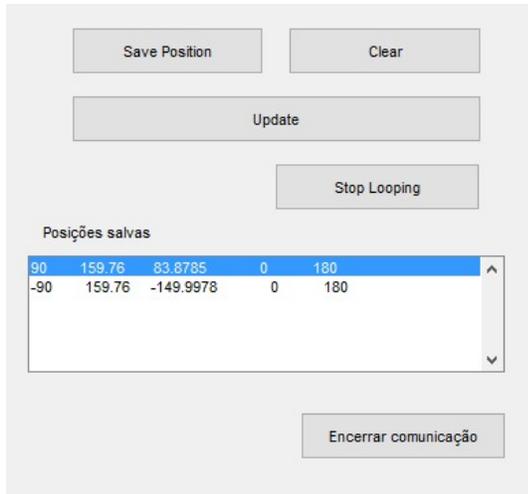


Figura 15. Interface para gerar uma sequência de movimentos.

Nesta interface, representada pela Fig. 15 é possível salvar a posição usando a cinemática direta ou inversa. Também é possível limpar as posições já salvas usando o botão "Clear". Esta função limpa toda a caixa de sequências. Ao terminar a sequência é necessário clicar em "Update" para o programa começar a rodar. É possível também utilizar o botão de "Stop Looping" a fim de parar a sequência. Porém, ao utilizar este último botão a sequência para apenas quando chegar no primeiro movimento da lista. Além disso, uma linha azul progride juntamente com os movimentos para o usuário saber qual ação está sendo executada. Por fim, o botão "Encerrar Comunicação" desabilita a possível troca de informações entre Matlab e Arduino. Os números dispostos como exemplo nas Fig. 11, Fig. 13 e Fig. 15 são os valores assumidos para a realização dos testes de precisão que serão abordados adiante.

Para que o Arduino receba corretamente as informações é montado um quadro contendo a quantidade de passos que cada motor deve executar. Este quadro está demonstrado na Fig. 16. Para cada motor são destinados 6 bytes, sendo o primeiro responsável pelo sentido de rotação do motor e os demais são responsáveis por armazenar a quantidade de passos. Para controle do acionamento do servo motor são destinados 3 bytes. Para garantir o correto envio das informações também é enviado um *Checksum*, sendo eles os últimos 3 bytes do quadro. Ao todo são enviados 30 bytes.

Quando este pacote de dados é enviado ao Arduino, o mesmo deve estar disponível para fazer a leitura e interpretação dos dados. Para isso, o Arduino segue o fluxograma representado na Fig. 17.

DIREÇÃO 1	MOTOR 1				
DIREÇÃO 2	MOTOR 2				
DIREÇÃO 3	MOTOR 3				
DIREÇÃO 4	MOTOR 4				
SERVO	SERVO	SERVO	CRC	CRC	CRC

Figura 16. Quadro enviado para o Arduino.

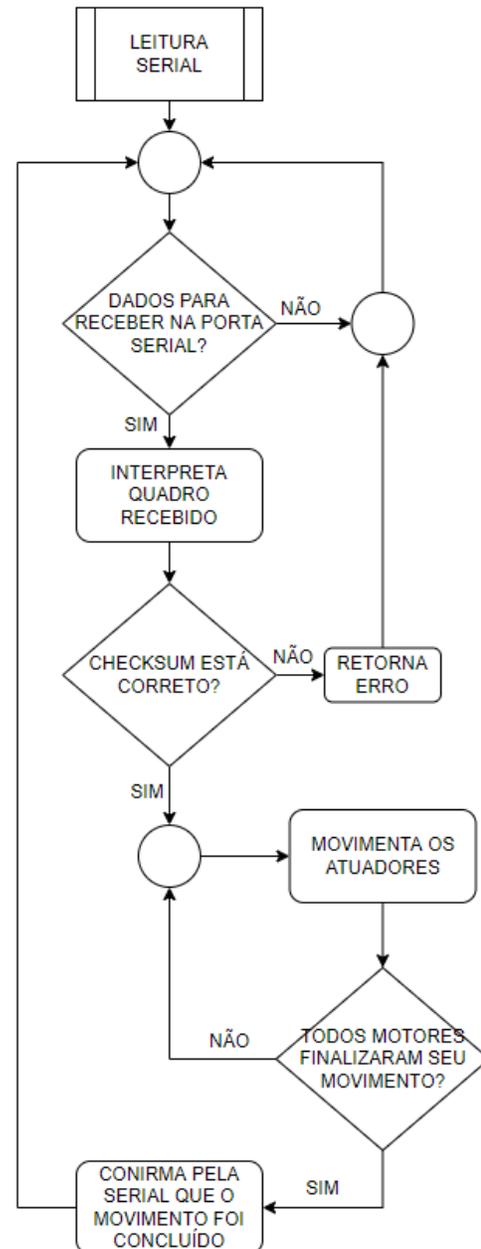


Figura 17. Fluxograma da interpretação de dados no Arduino.

D. Testes práticos

Para validar o funcionamento do sistema foram desenvolvidos testes de precisão, tanto das coordenadas x e y , quanto da coordenada z . O teste constitui em realizar o mesmo movimento por diversas vezes e analisar a divergência entre as posições resultantes. No final do movimento foi adicionado um relógio comparador digital da marca *Insize* com resolução de 0.005 mm. Inicialmente, com o manipulador na posição final, o relógio é setado para zero. Então são iniciados os testes de repetição. Para o caso das coordenadas x e y são variados os ângulos θ_1 e θ_2 conforme demonstrado na Fig. 15. Desta forma o ângulo θ_1 varia 180° enquanto o θ_2 varia 233.8763°. Em nenhum momento do teste os motores passam pelos seus respectivos fim de curso para calibrar. Assim, os resultados obtidos, que são a diferença entre a primeira medição e a medição atual, são demonstrados na Fig. 18.

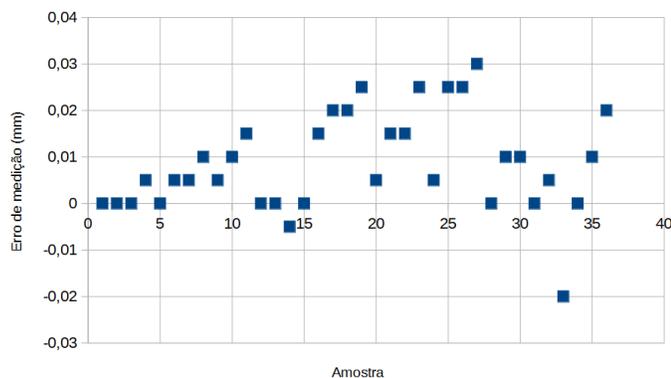


Figura 18. Teste de precisão das coordenadas x e y (mm).

Já para a medição da divergência na coordenada z é movimentado apenas o ângulo d_0 . O teste segue o mesmo princípio de medição citado acima. Assim, na Fig. 19 são demonstrados os resultados obtidos.

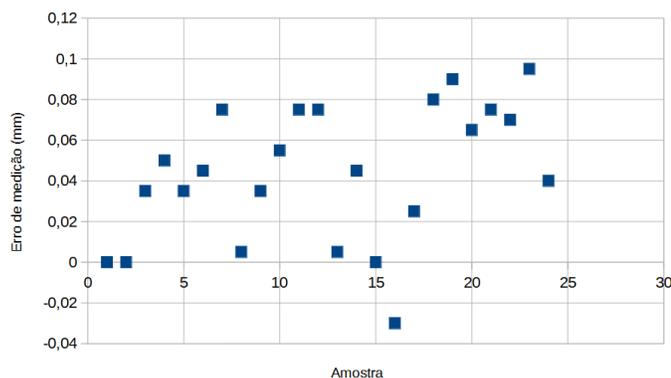


Figura 19. Teste de precisão da coordenada z (mm).

Além destes dois testes, foram executados movimentos básicos de "Pick and Place", onde o protótipo realizou os movimentos corretamente. Por fim, foi realizado um último teste básico que consistia em movimentar a ferramenta do robô

em uma distância linear de 100 mm. A medição final apontou que o protótipo movimentou uma distância de 99.3 mm. Este e todos os testes citados acima podem ser conferidos acessando [19].

VI. CONCLUSÃO

O desenvolvimento de um sistema de controle para um protótipo de SCARA compreende diversas áreas relacionadas a Engenharia de Controle e Automação, desde toda álgebra necessária para derivar a cinemática do manipulador até a construção do protótipo final.

Apesar de utilizar um protótipo *Open Source* ainda há dificuldades na fabricação do projeto, uma vez que há diversos tipos e marcas de filamentos para impressão e cada um age de maneira distinta após processados pela impressora 3D. Foi tomado todo cuidado para que não houvesse qualquer tipo de folga entre os componentes. Mesmo sem ter folgas consideráveis, o projeto mecânico se mostrou pouco rígido no eixo z , o que afeta diretamente o desempenho no movimento do ângulo d_0 . Por esse motivo, o protótipo foi programado para realizar separadamente os movimentos das coordenadas x , y e da coordenada z . Quando as três coordenadas eram movidas simultaneamente o protótipo apresentava elevadas trepidações, mesmo com aceleração e velocidade reduzida. Tal imperfeição poderia ser resolvida diminuindo a altura do protótipo, fator capaz de enrijecer a componente z do sistema. Os movimentos relacionados as coordenadas x e y se mostraram muito satisfatórios, conforme resultado dos testes de precisão.

A proposta do trabalho de desenvolver um sistema de controle baseado em malha aberta que tivesse um nível de confiabilidade alto, utilizando materiais de baixo custo, e escolhendo metodologias de desenvolvimento que melhor se aplicassem ao sistema foi efetuada com sucesso. Ao longo deste trabalho foram desenvolvidos códigos base no Matlab a partir do equacionamento do sistema. Tais códigos foram utilizados com sucesso na criação da interface gráfica, responsável por fazer a interação entre o operador e a máquina. A interface desenvolvida para controlar os movimentos do protótipo atendeu às expectativas da proposta, uma vez que o usuário pode movimentar corretamente o manipulador através da cinemática direta e inversa, podendo também executar seqüências de movimentos pré definidos. A comunicação entre o Matlab e o Arduino mostrou-se eficiente e funcional.

REFERÊNCIAS

- [1] I. Zamalloa, R. Kojcev, A. Hernández, I. Muguruza, L. Usategui, A. Bilbao, and V. Mayoral, "Dissecting Robotics - historical overview and future perspectives," *Acutronic Robotics*, pp. 1–9, apr 2017. [Online]. Available: <http://arxiv.org/abs/1704.08617>
- [2] H. Makino and N. Furuya, "Selective compliance assembly robot arm," *Proc. 1st Intern. Conf. on Assembly Automation (ICAA)*, pp. 77–86, 1980.
- [3] A. Gasparetto and L. Scalera, "A Brief History of Industrial Robotics in the 20th Century," *Advances in Historical Studies*, vol. 08, no. 01, pp. 24–35, 2019.
- [4] H. Makino, "Development of the SCARA," *Journal of Robotics and Mechatronics*, vol. 26, no. 1, pp. 5–8, 2014.

- [5] G. Graetz and G. Michaels, "ROBOTS AT WORK," *Review of Economics and Statistics*, vol. C, no. 5, pp. 754–768, 2018.
- [6] EPSON, "Robôs scara all-in-one Epson Synthis t3," Epson, site. [Online]. Available: <https://epson.com.br/Para-empresas/Rob%C3%B3tica/Rob%C3%B4s-SCARA/Rob%C3%B4s-SCARA-All-In-One-Epson-Synthis-T3/p/RT3-401SS>
- [7] J. d. J. R. Victor Hernandez, Gustavo Bravo and J. Pacheco, "KINEMATICS FOR THE SCARA AND THE CYLINDRICAL MANIPULATORS," *ICIC Express Letters*, vol. 2, no. 2, pp. 421–425, 2011.
- [8] F. X. Steinparz, "Co-ordinate transformation and robot control with Denavit-Hartenberg matrices," *Journal of Microcomputer Applications*, vol. 8, no. 4, pp. 303–316, 1985.
- [9] S. B. Niku, *Introduction To Robotics: Analysis, Control, Applications*, 2nd ed. River Street: JOHN WILEY SONS, INC, 2011, vol. 53, no. 9.
- [10] P. I. Corke, "A Simple and Systematic Approach to Assigning Denavit-Hartenberg Parameters," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 590–594, 2007.
- [11] F. Ding and C. Liu, "Applying coordinate fixed Denavit-Hartenberg method to solve the workspace of drilling robot arm," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, pp. 1–8, 2018.
- [12] S. Kucuk and Z. Bingul, "Robot workspace optimization based on a novel local and global performance indices," *IEEE International Symposium on Industrial Electronics*, vol. IV, no. April 2016, pp. 1593–1598, 2005.
- [13] M. Shariatee, A. Akbarzadeh, A. Mousavi, and S. Alimardani, "Design of an economical SCARA robot for industrial applications," *2014 2nd RSI/ISM International Conference on Robotics and Mechatronics, ICRoM 2014*, pp. 534–539, 2014.
- [14] I. Chavdarov, V. Nikolov, B. Naydenov, and G. Boiadjiev, "Design and control of an educational redundant 3D printed robot," *2019 27th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2019*, pp. 1–6, 2019.
- [15] A.-E. I. Florin COVACIU, "DESIGNING AND BUILDING A SERIAL ROBOTIC ARM WITH FOUR DEGREES OF FREEDOM," *Acta Technica Napocensis*, vol. 62, no. II, pp. 317–322, 2019.
- [16] Dejan, "Scara robot — how to build your own arduino based robot," How To Mechatronics, blog, Out. 2020. [Online]. Available: <https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/>
- [17] F. De Dinechin and M. Istoan, "Hardware Implementations of Fixed-Point Atan2," *Proceedings - Symposium on Computer Arithmetic*, vol. 2015-August, pp. 34–41, 2015.
- [18] G. Castelli and E. Ottaviano, "A Fairly General Algorithm to Evaluate Workspace Characteristics of Serial and Parallel Manipulators," *Mechanics Based Design of Structures and Machines*, no. April 2013, pp. 14–33, 2008.
- [19] Lazzari, "Demonstração do funcionamento e testes realizados com protótipo de scara," Google Drive, Dez. 2021. [Online]. Available: https://drive.google.com/file/d/1hhvJD07F_GC7IFd4iL0ptROe0iZzG1Mm/view?usp=sharing