



**UNIFOR**

**UNIVERSIDADE DE FORTALEZA**

**CENTRO DE CIÊNCIAS TECNOLÓGICAS**

**CURSO DE GRADUAÇÃO EM ENG. CONTROLE E AUTOMAÇÃO**

**EVANGELISTA, D.M. E SILVA, R.F. E PINTO, I.R.S E AZEVEDO, E.A.**

**RECUPERAÇÃO DE FUNCIONAMENTO DO DR. ROBOT JAGUAR**

**FORTALEZA – CEARÁ**

**2018**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>2</b>
1.1	OBJETIVOS . . . . .	2
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>3</b>
2.1	JAGUAR . . . . .	3
2.2	ROS . . . . .	3
2.3	CINEMÁTICA . . . . .	4
<b>3</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>5</b>
3.1	INSTALAÇÃO ROS E TESTE COM LAZER HOKUYO-30LX . . . . .	5
3.2	CINEMÁTICA DIRETA DO BRAÇO . . . . .	10
3.3	CINEMÁTICA INVERSA DO BRAÇO . . . . .	12
	<b>REFERÊNCIAS . . . . .</b>	<b>15</b>
	<b>GLOSSÁRIO . . . . .</b>	<b>15</b>

# 1 INTRODUÇÃO

Jaguar é um robô da Fabricante Dr. Robot que a UNIFOR possui. Quando iniciado o contato com o Jaguar no meio da disciplina, foi encontrado um robô de custo elevado sem funcionar por motivo desconhecido.

## 1.1 OBJETIVOS

Como objetivo da Disciplina de Sistemas Autônomos, foi proposto primeiramente o estudo do framework de desenvolvimento em robótica ROS, o estudo da cinemática do braço robótico acoplado ao modelo do Robô Jaguar presente no patrimônio da UNIFOR e a partir desses estudos controlar o braço do Jaguar a partir do método da cinemática inversa utilizando ROS.

Inicialmente ao projeto tinha se a ideia que o Jaguar era um robô construído com base em ROS tornado possível a conclusão dos objetivos propostos. Por causa da realidade a qual o Jaguar se encontrava e outros imprevistos o objetivo de controle do braço com ROS teve que ser alterado para recuperação do funcionamento do Jaguar.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 JAGUAR

O robô Jaguar com braço manipulador, o modelo que a UNIFOR, é projetado pela Dr. Robot para aplicações internas e externas que requer robustez de manobrabilidade, exploração em terrenos com rochosos e manipulação de objetos. Ele vem com quatro braços articulado e é totalmente sem fio. Integra também um GPS e um IMU de 9 graus de liberdade (giroscópio, acelerômetro e bússola) para navegação autônoma. Ele é projetado para terrenos extremos e capaz de subir escadas ou verticais até 300 mm com facilidade. Os 4 braços articulados poderiam converter o robô em várias configurações de navegação ideais para superar desafios diferentes do terreno. Possui também câmeras com vídeo e áudio de alta resolução integrado e um scanner laser que oferece informação detalhada do entorno do robô.

Integrado ao Jaguar há um robusto braço manipulador robótico 3 + 1 DOF (graus de liberdade), projetado para aplicações moveis, com baixo peso, tamanho e consumo de energia. A garra do braço possui um alcance máximo de mais de 707mm e a carga máxima de 4Kg (no alcance máximo). Há uma câmera de alta resolução montada ao pulso da garra ideal para a inspeção de objetos e manipulação. Apesar de o braço por si só ser limitado a 3 graus de liberdade (excluindo a garra) o braço alcança 6 graus de liberdade completo ao trabalhar em conjunto com a base móvel do robô e alcançar praticamente qualquer posição dentro de seu espaço permitido.

Por causa de tudo que consiste o Jaguar e também pelo fato que o Jaguar já está pronto, ele se torna uma excelente ferramenta para aplicar robótica autônoma.

### 2.2 ROS

O ROS é um sistema meta-operacional de código aberto para o seu robô. Ele fornece os serviços que você esperaria de um sistema operacional, incluindo abstração de hardware, controle de dispositivo de baixo nível, implementação de funcionalidade comumente usada, passagem de mensagens entre processos e gerenciamento de pacotes. Ele também fornece ferramentas e bibliotecas para obter, criar, gravar e executar código em vários computadores.

O principal recurso do ROS é a forma como o software é executado e a maneira como ele se comunica, permitindo que você projete software complexo sem saber como determinado hardware funciona. O ROS fornece uma maneira de conectar uma rede de processos (nodes) a

um hub central. Os nós podem ser executados em vários dispositivos e se conectam a esse hub de várias maneiras.

As principais formas de criar a rede são fornecer serviços solicitáveis ou definir conexões de publisher / subscriber (publicador / assinante) com outros nós. Ambos os métodos se comunicam por meio de tipos de mensagem especificados. Alguns tipos são fornecidos pelos pacotes principais, mas os tipos de mensagens podem ser definidos por pacotes individuais.

Os desenvolvedores podem montar um sistema complexo conectando soluções existentes para pequenos problemas.

## 2.3 CINEMÁTICA

Em um ambiente industrial, na ponta de um manipulador robótico encontra-se a ferramenta necessária para execução da tarefa em questão, esta pode ser programada para executar uma série de movimentações que a leve para um conjunto de posições e orientações de um plano cartesiano base (COCOTA *et al.*, 2013).

A cinemática é estudo do movimento de um corpo ou de um conjunto de corpos ou de um sistema desconsiderando forças e massas atuantes. Atraves do estudo da cinemática o movimento de uma cadeia de articulações pode ser previsto a partir das relações entre uma articulação e seus vizinhos. Uma aplicação comum para o estudo da cinemática de um robô é estudo dos caminhos para pose deste (CORKE, 2011).

A cinemática direta permite determinar a posição e orientação do efetuador, por exemplo, uma garra, em função das variáveis das juntas do manipulador em relação a sua base. Ou seja, é possível obter as coordenadas (x, y, z) da garra, utilizando as angulações de cada junta e o comprimento de cada elo. Enquanto a cinemática inversa faz exatamente o contrário da cinemática direta, ou seja, a partir da posição desejada em coordenadas (x, y, z) obtêm os ângulos necessários para o robô alcançar essas coordenadas, consistindo em descobrir as variáveis de junta (COCOTA *et al.*, 2013).

### 3 DESENVOLVIMENTO

Quando iniciado o contato com o Jaguar no meio da disciplina, foi encontrado um robô de custo elevado sem funcionar por motivo desconhecido, daí então foi estabelecido como objetivo a recuperação do funcionamento do Jaguar. Afim de se maximizar o aprendizado sobre robótica foi decidido desmontar ele completamente para melhor estudá-lo.

Após a desmontagem do Jaguar cada componente foi individualmente identificado. Todos os componentes do jaguar com exceção o controlador e a placa de distribuição de energia eram produtos de mercado, então poderiam ser estudados pelo datasheet fornecido pela fabricante. Apenas um componente não foi possível de identificar e encontrar o datasheet.

Com o datasheet de cada componente em mão, foi realizado um estudo rápido para testar o funcionamento de cada componente. Todos os equipamentos testados funcionaram de acordo com o esperado menos os encoders de velocidade dos motores que não geraram as ondas de saída de acordo com o datasheet. O componente não identificado foi alimentado e realizado alguns testes pequenos, aparentemente este também está funcionando com êxito. Após a conclusão dos testes de todos os componentes, começou a fase de remontagem do Jaguar.

#### 3.1 INSTALAÇÃO ROS E TESTE COM LAZER HOKUYO-30LX

O laser hokuyo-30LX é um laser infravermelho de longa distância, com 270° graus radianos de detecção e possui um range distância de 100m á 30.000mm (30 metros). Este equipamento necessita de uma alimentação de 12V. A conexão é mostrada da foto a seguir.

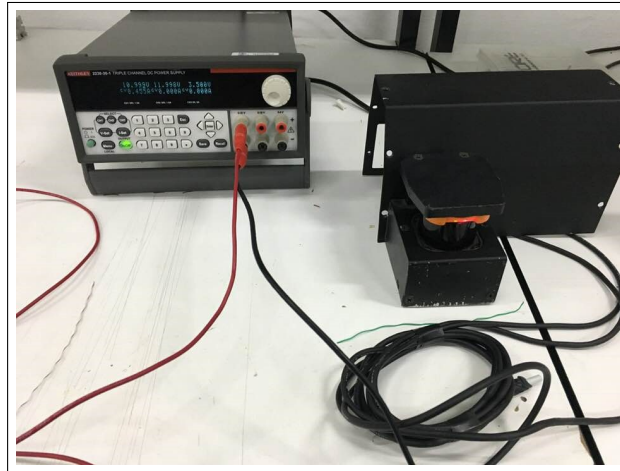
O laser HOKUYO possui uma conexão via interface USB 2.0, onde facilitada a conexão com qualquer PC ou Raspberry. A conexão do laser foi feita com um notebook para teste do mesmo. No notebook esta sendo utilizado o sistema operacional Ubuntu 16.04 e o ROS (Robot Operating System) versão Kinetic. Para ler os dados dos variados sensores através do ROS é necessário a instalação do ROS e dos drivers para os sensores. A versão do Ubuntu utilizada é indicada pois já possui o repositório nativo para se instalar a versão Kinetic do ROS, versão estável. Segue os passos para instalação e configuração desse teste:

1. Instalação do Ubuntu 16.04

Faça o download do Ubuntu no site <https://www.ubuntu.com/>. É um open source, livre para ser baixado mundialmente.

2. Instalação do ROS Kinetic

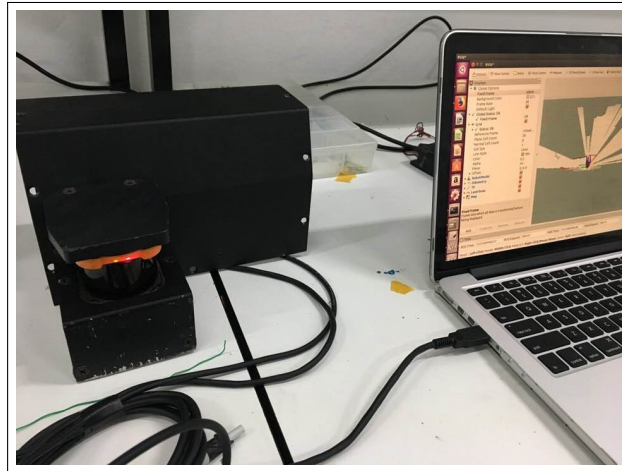
Após iniciar o sistema Ubuntu, abra o “Terminal”. E execute os seguintes comandos para

**Figura 1**

Fonte – Elaborado pelo autor

instalação e configuração do ROS.

- Configurar seu computador para aceitar o software de [packages.ros.org](http://packages.ros.org)  
`sudo sh -c 'echo "deb http://package.ros.org/ros/ubuntu $(lsb_release -sc) main»  
/etc/apt/sources.list.d/ros-latest.list'`
- Configurar as Keys (chaves)  
`sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F71`
- Atualizar o repositório do SO com as configurações setadas  
`sudo apt-get update`
- Instalar o ROS Kinetic versão completa (ROS, rqt, rviz, bibliotecas robóticas genéricas, simuladores 2D / 3D, navegação e percepção 2D / 3D)  
`sudo apt-get install ros-kinetic-desktop-full`
- Inicializar ROSDEP - Antes de poder usar o ROS, você precisará inicializar o rosdep. O rosdep permite que você instale facilmente as dependências do sistema para a origem que você deseja compilar e é necessário para executar alguns componentes principais no ROS.  
`Sudo rosdep init`  
`Rosdep update`
- Configurar Ambiente - É conveniente que as variáveis de ambiente do ROS sejam automaticamente adicionadas à sua sessão bash toda vez que um novo shell for lançado.  
`echo "source /opt/ros/kinetic/setup.bash»> ~/.bashrc`  
`source ~/.bashrc`

**Figura 2**

Fonte – Elaborado pelo autor

- Buildando pacotes - Até agora você instalou o que precisa para executar os pacotes principais do ROS. Para criar e gerenciar seus próprios espaços de trabalho do ROS, existem várias ferramentas e requisitos que são distribuídos separadamente. Por exemplo, o `roscpp` é uma ferramenta de linha de comando usada com frequência que permite baixar facilmente muitas árvores de origem para pacotes ROS com um comando. Para instalar esta ferramenta e outras dependências para construir pacotes ROS, execute:

```
sudo apt-get install python-roscpp python-roscpp-generator python-wstool build-essential
```

A foto a seguir mostra o laser HOKUYO conectado ao notebook via USB.

Para realizar o teste, o cabo USB do laser foi conectado a uma USB do notebook. Foi escolhida uma porta qualquer, não existe restrições. Após realizar a conexão é necessário seguir os seguintes passos:

- Instalar o driver do ROS para Laser Hokuyo - Driver Hokuyo\_Node

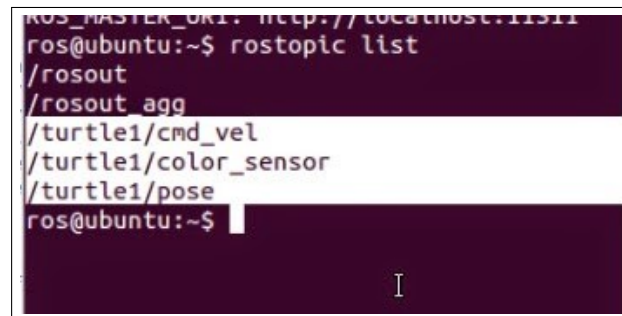
No link [https://github.com/ros-drivers/hokuyo\\_node](https://github.com/ros-drivers/hokuyo_node)

- Configurar no Workspace

É necessário que seja criado um workspace próprio para seus projetos do ROS. Quando o ROS é instalado, um workspace com os pacotes nativos do ROS é instalado nas pastas de libs do sistema operacional. Por isso é necessário a criação de um workspace paralelo para o melhor manuseio de pacotes que serão utilizados por você, em futuros trabalhos. Para criar o seu novo workspace siga os passos dispostos nessa página do site do ROS: [http://wiki.ros.org/catkin/Tutorials/create\\_a\\_workspace](http://wiki.ros.org/catkin/Tutorials/create_a_workspace). São



Figura 3



```

ROS_MASTER_URI: http://localhost:11311
ros@ubuntu:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
ros@ubuntu:~$

```

Fonte – Elaborado pelo autor

poucos passos e que geralmente são executados sem problemas.

Após o workspace ter sido criado, copie a pasta do Hokuyo\_Node e cole dentro da pasta “src” do seu novo workspace. Abra o terminal e entre na do diretório raiz do seu workspace e configura o driver executando os comandos:

- a) `cd ~/home/user/WORKSPACE` (exemplo)

Configurar driver e instalar dependências do mesmo

- b) `Catkin_make`

Se a máquina não tiver processamento e memória suficientes, essa instalação pode travar. Então execute:

- c) `Catkin_make -j3`

### 3. EXECUÇÃO DO ROS E LASER

Após ser realizado esses passos o driver do laser HOKUYO estará instalado corretamente no ROS. Para executar o ROS execute os seguintes comandos no TERMINAL:

- Rodar o ROS MASTER

`Roscore`

- Executar driver LASER - Abra outro terminal e execute o comando:

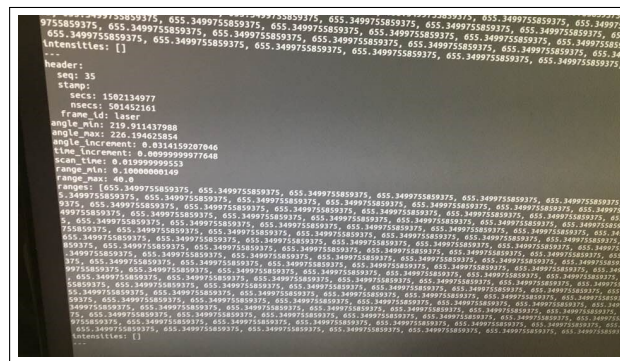
`Rosrun hokuyo_node hokuyo_node`

- Lembre que o laser deve estar conectado via usb com o computador. Para conferir se o driver executou corretamente é necessário abrir outro terminal e executar o comando do ROS para listar todos os tópicos que estão sendo publicados, que correspondem a todos dados de sensores que estão sendo executados em tempo real. Comando:

`Rostopic list`

- No terminal deverá aparecer os dados semelhantes ao da imagem abaixo:
- Procure um tópico com `/scan` ou um nome semelhante a “scan”, que corresponderá

Figura 4



Fonte – Elaborado pelo autor

aos dados do laser. Em seguida abra outro terminal, e execute o seguinte comando:  
Rostopic echo “../scan”

- Os dados do laser deverão aparecer na tela de acordo como mostrado na imagem abaixo.

Estes dados estão dispostos em vetores e correspondem as distancias calculadas por cada ponto de infravermelho emitidos no ambiente pelo laser HOKUYO.

Com o laser funcionando iremos visualizar os dados recebidos utilizando um software que já é instalado nativamente com o ROS. É um software de visualização, compatível com vários tipos de dados referentes a aplicações de robótica e muito utilizado no desenvolvimento dessas aplicações por possibilitar a visualização de forma abstraída de toda aplicação robótica em funcionamento, como também um mapeamento sendo realizado e a rota do robô pelo ambiente.

- Para executar o Rviz execute o seguinte comando:  
Rosrun rviz rviz
- Uma interface gráfica irá ser executada. Para estudar essa interface e aprender de forma completa o manuseio da mesma. Acesse o link:  
<http://wiki.ros.org/rviz/UserGuide>
- Na side bar do lado esquerdo, no canto inferior clique no botão “ADD” e procure o tópico /scan na lista, na janela que ira se abrir. Após adicionar o tópico /scan marque a check box na side bar. E você deverá visualizar os dados do laser.

### 3.2 CINEMÁTICA DIRETA DO BRAÇO

Através de matrizes, transformações complexas podem ser expressadas de forma simples, porém para ser capaz de representar todas as transformações possíveis deve-se converter o universo do problema de 2D para 3D (ou de 3D para 4D) utilizando coordenadas homogêneas (FOLEY; DAM, 1982)

Para dar prosseguimento aos cálculos da modelagem é importante saber as matrizes de rotação para cada eixo (x, y ou z). Cada uma dessas matrizes representa o movimento de rotação entre o eixo **i-1** e **i**. De acordo com Foley e Van Dam (1982):

Para o eixo x:

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

Para o eixo y:

$$R_y = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

Para o eixo z:

$$R_z = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

No caso das matrizes de translação, essas mostram o deslocamento feito pelo elo

sobre o eixo x, y ou z. Sua representação de acordo com Foley e Van Dam (1982):

$$\text{Translação : } \begin{pmatrix} 1 & 0 & 0 & \Delta_x \\ 0 & 1 & 0 & \Delta_y \\ 0 & 0 & 1 & \Delta_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

A Figura 5 mostra um esquemático do braço visto lateralmente. Para a realização dos cálculos da cinemática direta as distâncias entre as juntas do braço robótico são as seguintes:  $a_1 = 99,50\text{mm}$  para a base, e para os demais eixos:  $a_2 = 313\text{mm}$ ;  $a_3 + a_4 = 394\text{mm}$  e  $a_4 = 120\text{mm}$ .

A Figura 6 também é uma vista lateral do braço, só que nesta, é dado ênfase nos graus de liberdade do mesmo, sendo demonstrados os ângulos de rotação da base e das demais articulações. Bem como os elos de ligações e os eixos de rotação e de translação das juntas.

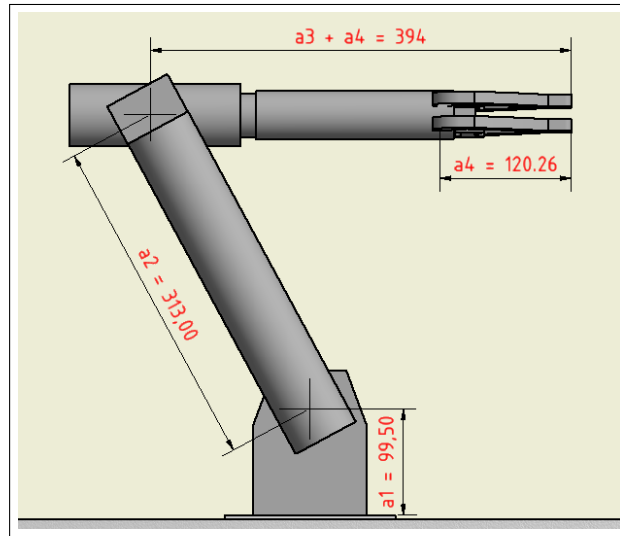
Dessa forma, para cada elo haverá uma matriz equivalente que será formada pela multiplicação das respectivas matrizes de rotação e translação. Neste projeto, as matrizes que representam cada um desses elos são dadas pelas seguintes equações:

$$M_1^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

$$M_2^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(90 + \theta_2) & 0 & \sin(90 + \theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(90 + \theta_2) & 0 & \cos(90 + \theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

$$M_3^2 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & (a_3 + a_4) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

No caso da matriz  $M_1^0$ , esta representa o elo que liga as juntas 0 e 1, sendo o resultado de uma matriz transladada no eixo z de  $a_1$  e rotacionada  $\theta_1$  em y, respectivamente. Já a matriz

**Figura 5**

Fonte – Elaborado pelo autor

$M_2^1$  representa o elo de ligação das juntas 1 e 2, é o resultado de uma matriz transladada  $a_2$  em z em conjunto com uma matriz rotacionada  $90^\circ + \theta_2$  no eixo y, nesse caso isso se faz necessário devido a diferença em relação a posição de repouso do braço conforme é mostrado na Figura 1.

Já a matriz  $M_3^2$  representa o elo de ligação das juntas 2 e 3, sendo o resultado de uma translação no eixo z de  $a_3 + a_4$  acompanhada de uma rotação também em z de  $\theta_3$ , ângulo este que é definido pelo próprio usuário.

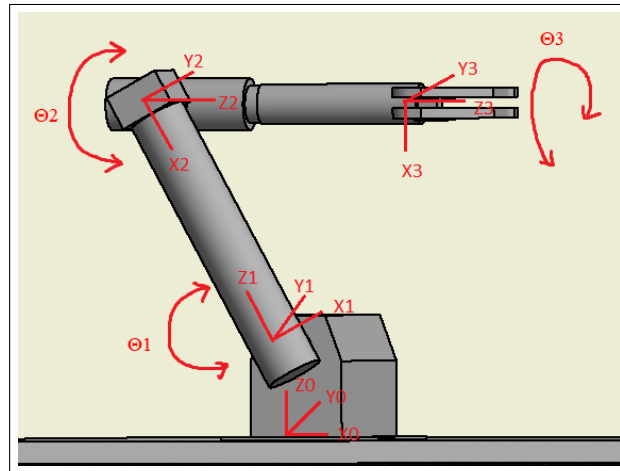
Ao final, será calculada a matriz equivalente formada pela multiplicação de todas as anteriores da seguinte forma:

$$M_3^0 = M_1^0 * M_2^1 * M_3^2 \quad (3.8)$$

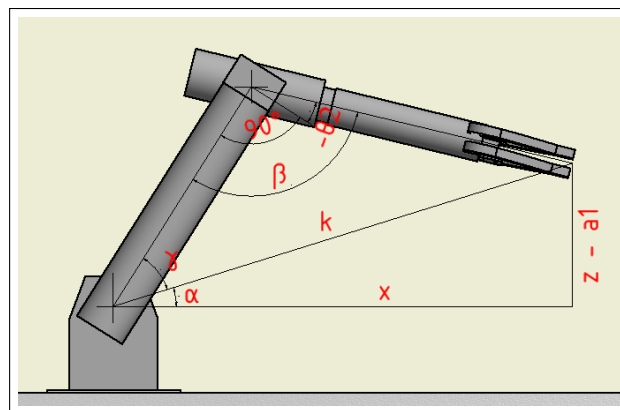
Assim, a partir da matriz representada pela Equação 3.4 será possível mostrar, com base nas rotações e translações ocorridas ao longo da estrutura do braço, na primeira, segunda e terceira linha da última coluna as coordenadas x, y e z, respectivamente. Determinando-se assim a sua posição e orientação.

### 3.3 CINEMÁTICA INVERSA DO BRAÇO

A cinemática inversa faz exatamente o contrário da cinemática direta, ou seja, a partir da posição desejada em coordenadas (x, y, z) obtêm os ângulos necessários para o robô alcançar essas coordenadas, consistindo em descobrir as variáveis de junta (OLIVEIRA I.F.R.,

**Figura 6**

Fonte – Elaborado pelo autor

**Figura 7**

Fonte – Elaborado pelo autor

2017).

A cinemática inversa torna-se complexa pois envolve equações cinemáticas não lineares, referindo-se ao cálculo de todos os possíveis conjuntos de ângulos das juntas para atingir determinada posição, podendo possuir múltiplas soluções (OLIVEIRA, 2017).

Este método tem como características seu alto custo de interação, poucos pontos de interpolação e garantia de chegada no final da trajetória, mesmo passando por uma região não alcançável (ROCHA, 2017).

Analisando a vista lateral, presente na Figura 7, poderemos calcular as variáveis abaixo:

- Para calcularmos  $\theta_1$  primeiro calcularemos através da lei dos cossenos os ângulos  $\alpha$  e  $\gamma$ :

$$\alpha = \text{atan}\left(\frac{z - a_1}{x}\right) \quad (3.9)$$

$$k = \sqrt{x^2 + (z - \alpha_1)^2} \quad (3.10)$$

$$\gamma = \arccos\left(\frac{a_2^2 + k^2 - (a_3 + a_4)^2}{2 * a_2 * k}\right) \quad (3.11)$$

$$\theta_1 = 90 - \alpha - \gamma \quad (3.12)$$

- Para calcularmos  $\theta_2$  também utilizaremos a lei dos cossenos para calcular o ângulo  $\beta$  :

$$\beta = \arccos\left(\frac{a_2^2 + (a_3 + a_4)^2 - k^2}{2 * a_2 * (a_3 + a_4)}\right) \quad (3.13)$$

$$\theta_2 = 90 - \beta \quad (3.14)$$

Para os cálculos da Cinemática inversa não consideraremos  $\theta_4$ .

No processo de validação das cinemáticas diretas e inversas trabalhamos com ângulos na entrada da cinemática direta, e através das multiplicações de matrizes chegamos a valores de x, y e z, que correspondem as coordenadas reais de posicionamento da garra do braço robótico. Com estes valores de x, y e z alimentamos a cinemática inversa para obtermos ângulos no final dos cálculos. Como já abordamos anteriormente, a cinemática inversa pode gerar mais de uma resposta e para validarmos este ciclo de cálculos realimentamos novamente a cinemática direta com estes novos ângulos obtidos. Ao final deste novo ciclo de cálculos da cinemática direta obtivemos os mesmos valores de x, y e z onde validamos nosso processo de teste de modelagem deste braço em conjunto com estas cinemáticas.

## REFERÊNCIAS

- COCOTA, J.; BARBOSA, R.; MOREIRA, A.; LAGE, V. Desenvolvimento de um robô antropomórfico com punho esférico para práticas de robótica com alunos de graduação. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 1, 2013.
- CORKE, P. **Robotics, Vision and Control**. [S.l.]: Springer, 2011. v. 1.
- FOLEY, J.; DAM, A. V. **Fundamentals of Interactive Computer Graphics**. [S.l.]: Addison Wesley, 1982. v. 2.
- OLIVEIRA I.F.R., T. Otimização do controle de um robô quadrúpede usando ferramentas do matlab simulink. 2017.
- OLIVEIRA, J. Projeto e construção do protótipo de um robô articulado com três graus de liberdade acionado por motores elétricos. 2017.
- ROCHA, E. Análise de desempenho e controle de um robô scara. 2017.