



Instituto Politécnico do Cávado e do Ave
Escola Superior de Tecnologia

Engenharia Eletrotécnica e de Computadores

Robótica

Manipulador Robótico – LEGO EV3

Trabalho Prático Nº2

Tiago Carvalho nº18601
João Sampaio nº18611
Rui Ribeiro nº16419

Janeiro 2023

Resumo

O trabalho prático tem como objetivo a avaliação da componente prática da unidade curricular de Robótica, do curso de Engenharia Eletrotécnica e de Computadores.

O presente relatório visa demonstrar e explicar o processo de desenvolvimento desse mesmo trabalho prático, que consolida os dois tópicos abordados ao longo da unidade curricular, mais propriamente, processamento de imagem e robótica.

Recorrendo à tecnologia da Lego Mindstorms contruímos um manipulador robótico de 3 graus de liberdade, que tem como objetivo deslocar-se para uma determinada posição e recolher um objeto (dentro do seu espaço de trabalho).

O projeto está dividido em duas partes:

- A primeira parte baseada em processamento de imagem, que permite recolher a informação da posição do objeto, utilizamos uma webcam que captura a imagem, aplicando um threshold referente à cor do objeto.
- A segunda parte, através do centroide desse objeto na imagem, convertemos as medidas para o plano real, que de seguida serão convertidas novamente para os ângulos das juntas, permitindo ao robô chegar a esse ponto.

Índice

Introdução	6
1 - Manipulador Robótico	7
2 - Parâmetros Denavit-Hartenberg	8
3 - Espaço de Trabalho e Limites das Juntas.....	10
3.1 - Espaço de Trabalho	10
3.2 - Limite das Juntas.....	11
4 - Cálculo da Trajetória	12
5 - Movimento das Juntas	13
5.1 – A Implementação.....	13
5.1.1 - Método 1	13
5.1.2 - Método 2.....	14
6 - End-Effector / Garra	15
7 - Processamento de Imagem	16
7.1 - Posição no Mundo.....	18
8 – Fluxograma.....	20
9 - Resultado final.....	23
Conclusão	24

Índice de Figuras

Figura 1 - Manipulador Robótico Industrial	7
Figura 2 - Manipulador robótico H25	7
Figura 3 - Imagem Auxiliar para Extração dos Parâmetros de Denavit-Hartenverg	8
Figura 4 - Matriz Denavit-Hartenberg	9
Figura 5 - Exemplo de uma Matriz Homogênia	9
Figura 6 - Espaço Trabalho Robot Cilindrico	10
Figura 7 - Espaço de Trabalho Robô LEGO 3DOF	10
Figura 8 - Limites Mínimos de Todas as Juntas	11
Figura 9 - Limite Máximo da Junta da Base	11
Figura 10 - Limite Máximo da Junta do Pulso	11
Figura 11 - Limite Máximo da Junta do Cotovelo	11
Figura 12 - Garra - LEGO EndEffector	15
Figura 13 - Imagem Capturada	16
Figura 14 - Filtragem e Preenchimento	16
Figura 15 - Calculo da Posição do Objeto	17

Introdução

No âmbito da unidade curricular de Robótica do curso de Engenharia Eletrotécnica e de Computadores, da Escola Superior de Tecnologia IPCA, foi proposta a realização de um trabalho prático de modo a explorar técnicas de controlo robótico através da utilização de vários sensores/atuadores .

Este projeto tem como objetivo colocar em prática os conhecimentos adquiridos ao longo da unidade curricular, desenvolvendo também a programação em Matlab.

O mesmo consiste na captura de imagem, e através da deteção de objetos presentes na imagem, controlar um braço robótico, para que este pegue no objeto e o separe de acordo com a cor selecionada, este braço deve ser movido automaticamente consoante a posição do objeto no mundo.

A imagem capturada e processada é de uma peça lego de uma determinada cor, e para além da cor será também extraída a posição da peça.

O robô é uma variação da montagem do LEGO MINDSTORM EV3, que foi montado como um manipulador robótico de 3 graus de liberdade, e que será responsável pela separação das peças.

O presente relatório irá clarificar todo o processo de desenvolvimento realizado pelo grupo de trabalho.

Tem como objetivo também, trabalhar sobre alguns pontos importantes lecionados, tais como cinemática direta e inversa, espaço de trabalho, planeamento de trajetórias.

1 - Manipulador Robótico

Um manipulador multifuncional reprogramável projetado para mover materiais, peças, ferramentas ou dispositivos especializados por meio de movimentos variados programados, para o desempenho de uma variedade de tarefas que também adquirem informação do ambiente imóvel de forma inteligente em resposta ao mesmo.

Na nossa terminologia, um robô será composto de um circuito eletrônico computadorizado de controlo e um mecanismo articulado denominado manipulador que é composto pelo braço e pulso.

O braço consiste em elementos denominados elos e unidos por juntas.

O manipulador escolhido, foi o modulo EV3 que através da tecnologia da Lego Mindstorms contruímos o manipulador básico que o kit permitiu (H25).

Após contruída essa base, replicamos eficientemente as instruções da segunda junta, criando assim um robot com 3 juntas(3DOF) revolutas, consequentemente aumentando as capacidades do robot. Esta junta permite um espaço de trabalho maior do que o robô original

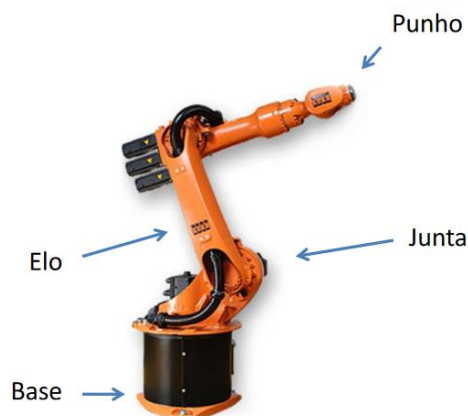


Figura 1 - Manipulador Robótico Industrial



Figura 2 - Manipulador robótico H25

2 - Parâmetros Denavit-Hartenberg

O método de cálculo da cinemática direta ou inversa, pode ser feito através dos parâmetros de Denavit-Hartenberg, que é um método para calcular a posição e orientação do end-effector do manipulador, relativo a base em função das variáveis das juntas. É um método sistemático para atribuir sistemas de eixos de coordenadas a cada elo de uma cadeia articulada. Qualquer robô pode ser descrito cinematicamente através das quatro quantidades dos links, onde dois descrevem o próprio elo e os outros a ligação ao próximo.

Para a elaboração da tabela dos parâmetros de Denavit-Hartenberg, começou-se por marcar os eixos do mesmo, colocando o robô numa posição que seria a posição “base”, depois pelas regras da definição dos eixos, começou-se a definir os eixos para depois se retirar os parâmetros corretos.

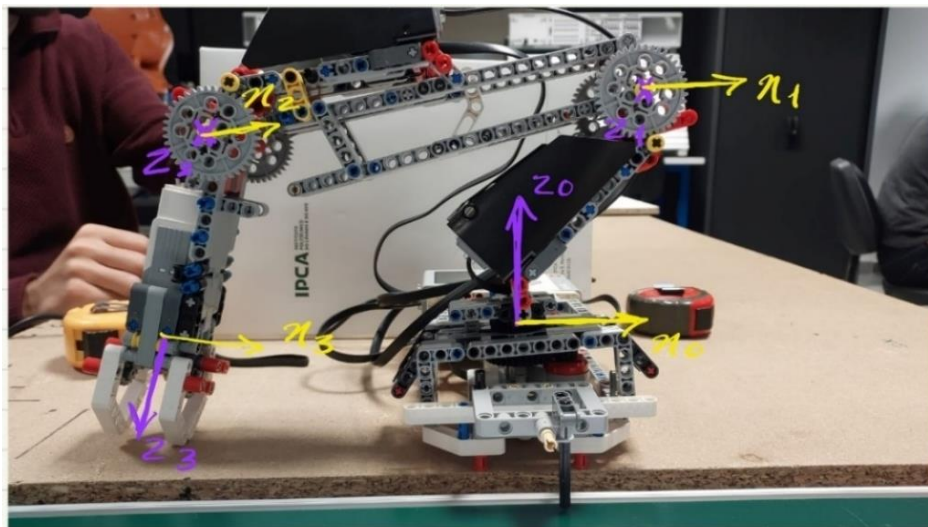


Figura 3 - Imagem Auxiliar para Extração dos Parâmetros de Denavit-Hartenberg

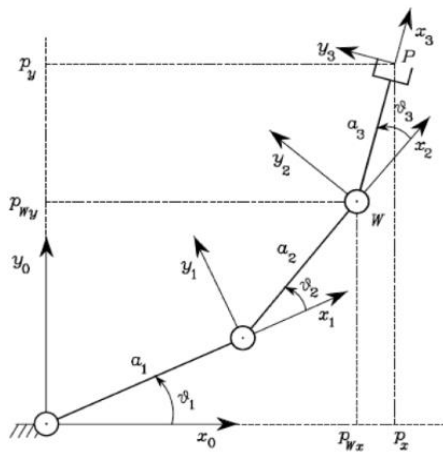
Após a marcação dos eixos, como se pode ver pela imagem 2, colocou-se na seguinte tabela os valores correspondentes aos parâmetros de Denavit-Hartenberg. Respeitando a tabela dos parâmetros, que definem as regras para a dedução dos parâmetros do robô.

Juntas	Alpha	a	d	Theta
1	-90	0.13	0	Θ_1
2	0	-0.19	0	Θ_2
3	-90	0	0	Θ_3

Os parâmetros da tabela acima são usados numa matriz homogênea que que se pode observar pelo exemplo da figura 4, esta matriz permite saber sempre as posições e os ângulos das juntas dos elos ligados.

$${}^{i-1}A_i(q_i) = {}^{i-1}A_{i'}(q_i) {}^{i'}A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4 - Matriz Denavit-Hartenberg



$$T_3^0(q) = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Link	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

Figura 5 - Exemplo de uma Matriz Homogenia

3 - Espaço de Trabalho e Limites das Juntas

3.1 - Espaço de Trabalho

O espaço/volume de trabalho é o termo que se refere ao espaço que um determinado braço consegue posicionar o seu pulso. Este volume, em geral, é estabelecido conforme os limites impostos pelo projeto estrutural do braço, ou seja, a configuração física do braço robótico, os limites dos movimentos das juntas e o tamanho dos componentes do corpo, braço e pulso. Na maior parte deles, o volume é altamente dependente de detalhes construtivos e raramente se aproxima do volume teórico.

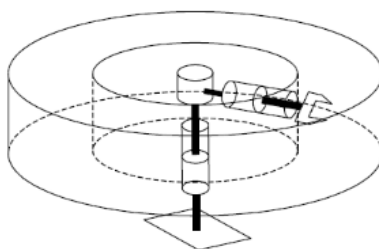


Figura 6 - Espaço Trabalho Robot Cilíndrico

Para delimitar o espaço de trabalho, o robô deve ser colocado em todas as posições onde o end effector consegue ter acesso com apenas um movimento e assim é possível saber o espaço de trabalho total do robô.

Neste caso, o espaço de trabalho pode ser observado pela imagem 7. Para o determinar foi colocado o end effector na posição máxima e registrou-se o espaço primário máximo e mínimo, e o que fica entre os dois é o espaço secundário, onde o robot consegue chegar com qualquer posição das juntas.

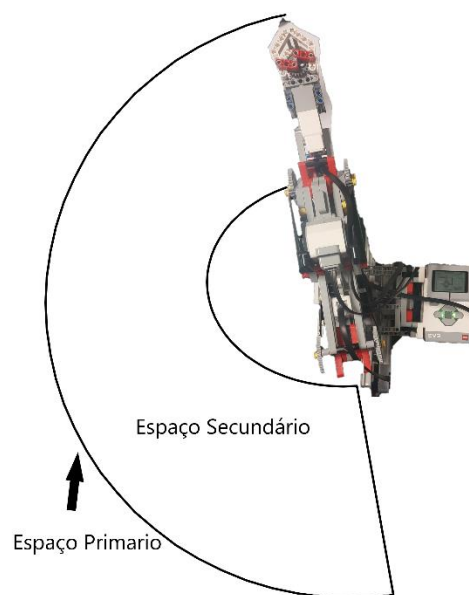


Figura 7 - Espaço de Trabalho Robô LEGO 3DOF

3.2 - Limite das Juntas

Quanto aos limites das juntas, estes são importantes para o cálculo da trajetória, isto pois é necessário saber até onde o robô pode rodar, por exemplo, senão na base por exemplo, o robot para atingir a posição 270° , pode girar entre 0° e 270° , segundo o arco trigonométrico, ou girar -60° , que é a trajetória mais curta, mas pode ser fisicamente impossível para o robô girar -60° , por isso é que é necessário ter em atenção os valores das juntas, mesmo se quisermos limitar o robô no espaço de trabalho.

Nesta situação, os valores das juntas foram retirados manualmente, movendo os motores para as suas posições máximas, desde o ponto inicial 0° , e foram registados os seguintes valores:

Junta	Mínimo	Máximo
1 – “Base”	0°	-180°
2 – “Cotovelo”	0°	-90°
3 – “Pulso”	0°	-120°

NOTA: Para os motores da LEGO, 180° corresponde na base a aproximadamente 600° , no cotovelo 300° e no pulso 250° .

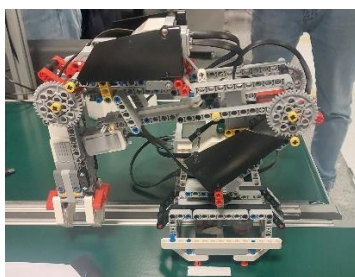


Figura 8 - Limites Mínimos de Todas as Juntas

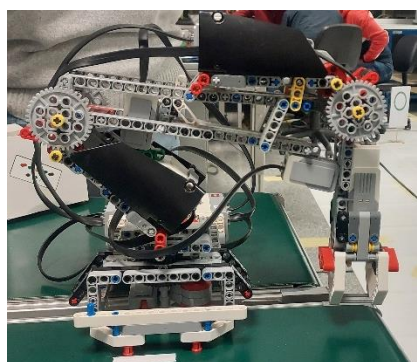


Figura 9 - Limite Máximo da Junta da Base

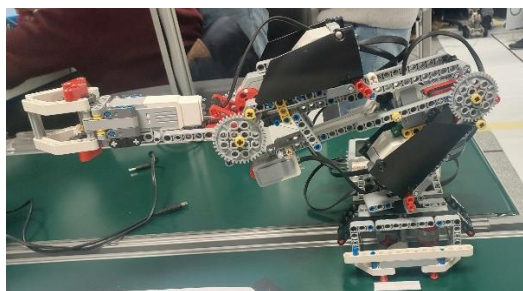


Figura 10 - Limite Máximo da Junta do Pulso

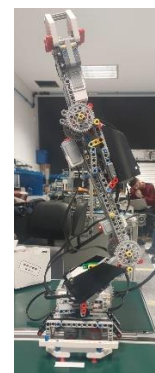


Figura 11 - Limite Máximo da Junta do Cotovelo

4 - Cálculo da Trajetória

Após definidos os parâmetros de Hartenberg corretamente, procedemos ao cálculo das trajetórias, enviando o robô para um ponto no plano, de forma a testá-lo.

Inicialmente o robô é colocado na sua posição inicial que corresponde ao 0 em todas as juntas, sendo de seguida calculada a cinemática direta. Este passo vai permitir resolver a posição cartesiana e a orientação de todas as juntas, dado o conhecimento da estrutura cinemática e das coordenadas dessas mesmas juntas.

Para calcular a trajetória, é necessário ter duas posições e um intervalo de tempo, sendo o intervalo de tempo dez segundos, e de forma a suavizar a trajetória, cada movimento será feito de 50ms em 50ms, num total de 200 intervalos, onde cada um deles terá um valor de velocidade distinto.

Quanto as posições, a primeira posição será a base, onde o valor das juntas é igual a zero, para saber a posição cartesiana, aplica-se a cinemática direta.

O cálculo da cinemática direta baseia-se no cálculo das matrizes rotação e/ou translação, que multiplicadas são expressas pela matriz de transformação homogênea, mencionada anteriormente.

Quanto á segunda posição, já são conhecidos os valores cartesianos, pois é o valor que a função de processamento de imagem retorna.

Em seguida e através da função “ctraj”, pode-se calcular as trajetórias, assim como todas as posições cartesianas do end effector para cada intervalo de tempo.

Para converter da posição cartesiana do end effector para a posição das juntas, utiliza-se a cinemática inversa que através da posição do end effector nos diz os ângulos de cada junta.

5 - Movimento das Juntas

Para se mover as juntas do robô, acede-se à matriz resultante do cálculo da trajetória, e usa-se o valor que foi calculado para cada junta, em ângulos, para calcular a velocidade dos motores, onde a velocidade vai ser igual ao ângulo da junta menos a posição anterior, a dividir pelo intervalo de tempo, no caso 50ms, ou seja:

$$V_{motor} = \frac{Angulo_{Base} - Posição_{Anterior}}{Intervalo_{Tempo}}$$

Posto isto basta passar para o robô a velocidade calculada, para ele se mover durante o intervalo de tempo, que é de 10 segundos, dividido em intervalos de 50ms, o que dá um total de 200 iterações.

Pelo facto de a velocidade ser muito pequena para o robô da LEGO se mexer, devido ao peso, foi necessário encontrar um novo método para a movimentação do robô.

5.1 – A Implementação

Como não foi possível seguir o método anterior devido as limitações do robô, foi inicialmente feita uma abordagem mais simples do controlo do mesmo, e depois numa outra aproximação, foi codificada uma solução mais próxima daquela que é a real.

5.1.1 - Método 1

O primeiro método de movimentação do robô, foi baseado em duas posições fixas, onde já existem duas posições predefinidas. Depois com o processamento de imagem, podemos saber em qual da posição se encontra a peça, para mover o robô. Usamos uma velocidade fixa no motor ao longo do tempo e existe uma comparação da posição final e a atual, ou seja, sabendo a posição em ângulos para a qual o robô se tem de mexer, este move-se a uma velocidade constante,

conseguindo suportar o seu peso e chegar à posição dentro do tempo pretendido. Durante o movimento lê-se a posição atual do robô, em ângulos, se essa posição for menor que a desejada a robô continua a mexer-se, quando esta for atingida, a velocidade do motor passa a ser zero, indicando que chegou ao objetivo.

O robô com ajuda da garra pega no objeto e move-se até a posição de descarga do objeto, no caso, duas posições previamente definidas, quando o robô chegar a posição de descarga apenas abre a garra e deixa cair o objeto, depois retorna à posição inicial.

Para retomar à posição inicial o processo é mais simples, pois com o recurso a três fins de curso, provenientes do kit, o robô vai mexer-se a uma velocidade constante, até que os fins de curso sejam pressionados, quando todos estes fins de curso tiverem o valor verdadeiro, o robô atingiu a posição “home”, e está preparado para realizar mais operações.

5.1.2 - Método 2

Na segunda aproximação à solução, através das coordenadas do centro do objeto, a posição do mesmo é calculada em relação ao tamanho da folha A4, e ao tamanho da imagem, convertendo cada pixel para centímetros.

Em seguida converte-se essa posição para o espaço trigonométrico, de 0° a 360° , e depois converte-se novamente do espaço trigonométrico para ângulos do motor, visto que os ângulos do círculo trigonométrico não correspondem aos mesmos ângulos dos movimentos dos motores.

Na movimentação do robô, a aproximação à posição é feita semelhante ao método anterior, onde se compara a posição atual com a final. Depois de depositar o objeto retorna à posição inicial novamente.

6 - End-Effector / Garra

Na robótica o órgão terminal é usado para descrever a mão ou ferramenta que está acoplada ao pulso, como por exemplo, uma garra pulverizadora de tinta, entre outros.

O órgão terminal é responsável por realizar manipulação de objetos em diferentes tamanhos formas e materiais, porém esta manipulação depende da aplicação ao qual se destina.

É de salientar que o end effector requer cuidados ao ser dimensionado pois é necessário controlar a força que será aplicada no objeto e a forma como este irá pegar no mesmo.

O modelo escolhido para end effector foram os “grippers”, que consiste em uma garra de dois dedos, modelo que proporciona pouca versatilidade na manipulação dos objetos, visto que existe limitação na abertura da garra, como no caso os objetos a serem agarrados são peças LEGO encaixadas numa roda LEGO, os limites da garra nunca são alcançados.

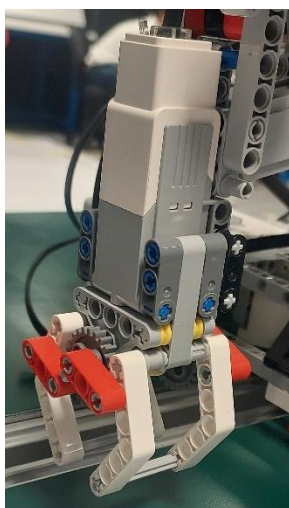


Figura 12 - Garra - LEGO EndEffector

7 - Processamento de Imagem

Neste projeto, a incorporação do processamento de imagem consiste na implementação da identificação das cores das peças, e consoante a cor, o robô coloca a peça na caixa correspondente.

A função de captura de imagem irá possibilitar tirar uma foto ao espaço onde existem as peças, que por sua vez será processada identificando a posição da peça escolhida no mundo.

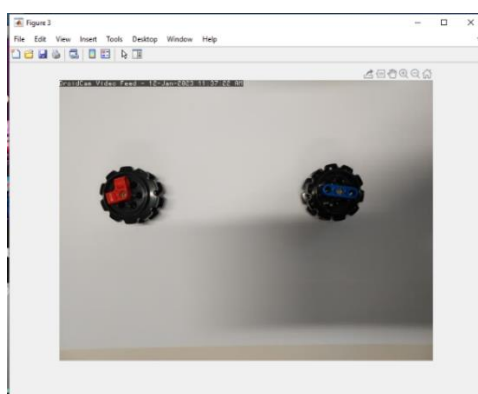


Figura 13 - Imagem Capturada

Inicialmente a imagem é capturada, seguida de uma segmentação por threshold de cor, no exemplo vermelha. Em seguida, através da função `imclose` é feito um fecho morfológico (dilatação e uma erosão) para delinear melhor o objeto. Depois através da função `imfill`, o interior do objeto é preenchido para que possa ser etiquetado.

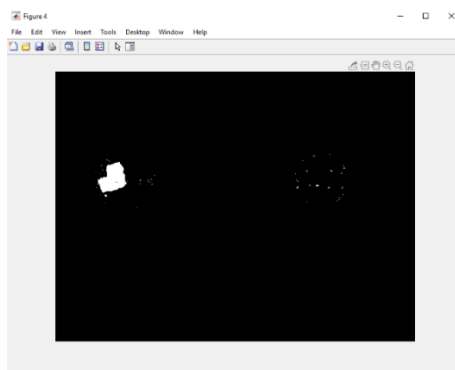


Figura 14 - Filtragem e Preenchimento

O próximo passo é eliminar o ruído existente na imagem de forma a ficar apenas o objeto que queremos, e excluir pequenos pixels que não tenham sido removidos na aplicação do filtro de cor.

Após eliminar todo o ruído, etiquetou-se os objetos detetados, e em seguida, extraiu-se as características necessárias para se poder saber a posição no mundo do nosso objeto. Através da função `regionprops`, foi retirado o centro do objeto(centroid) e daí obtivemos a posição x e y do mesmo.

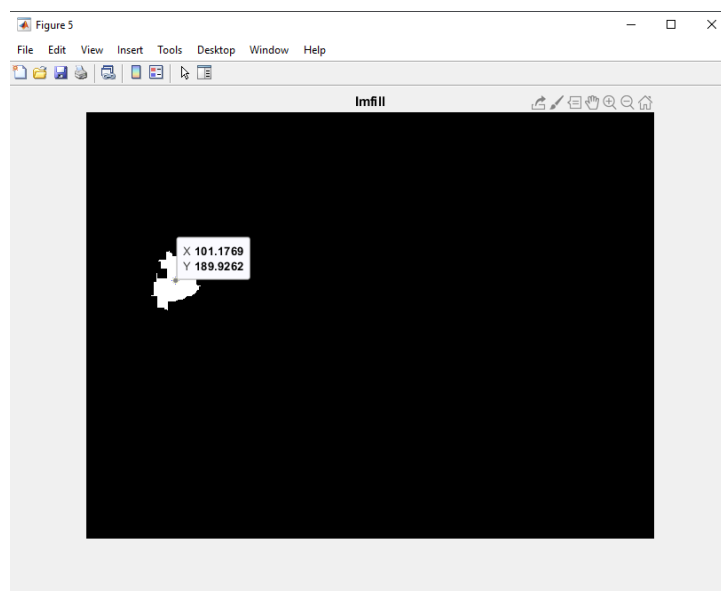


Figura 15 - Calculo da Posição do Objeto

7.1 - Posição no Mundo

Para converter a posição do objeto em pixels para coordenadas cartesianas, foram retiradas as coordenadas x e y correspondentes ao centroide. Essas coordenadas foram multiplicadas pelo tamanho da folha A4, e divididas pelo tamanho da imagem. Depois foi adicionada uma compensação a x e y, de forma a colmatar um ligeiro erro entre os cálculos e a componente real.

No próximo excerto pode-se observar como foi realizada a conversão.

```
"imSize = size(im) )% Tamanho da Imagem em x/y"  
"% Conversão da Posição em Relação a Folha"  
"objPosX = ((centroid(:,1) * 21) / imSize(:,1)) -1 % Posição em X"  
"objPosY = ((centroid(:,2) * 29.7) / imSize(:,2)) -2.5 %Posição em Y"
```

Como referido anteriormente, as coordenadas cartesianas são convertidas para distância, entre os limites da zona de repouso das peças. Em seguida essas distâncias são convertidas para o espaço trigonométrico, onde posteriormente são ajustadas aos ângulos dos motores.

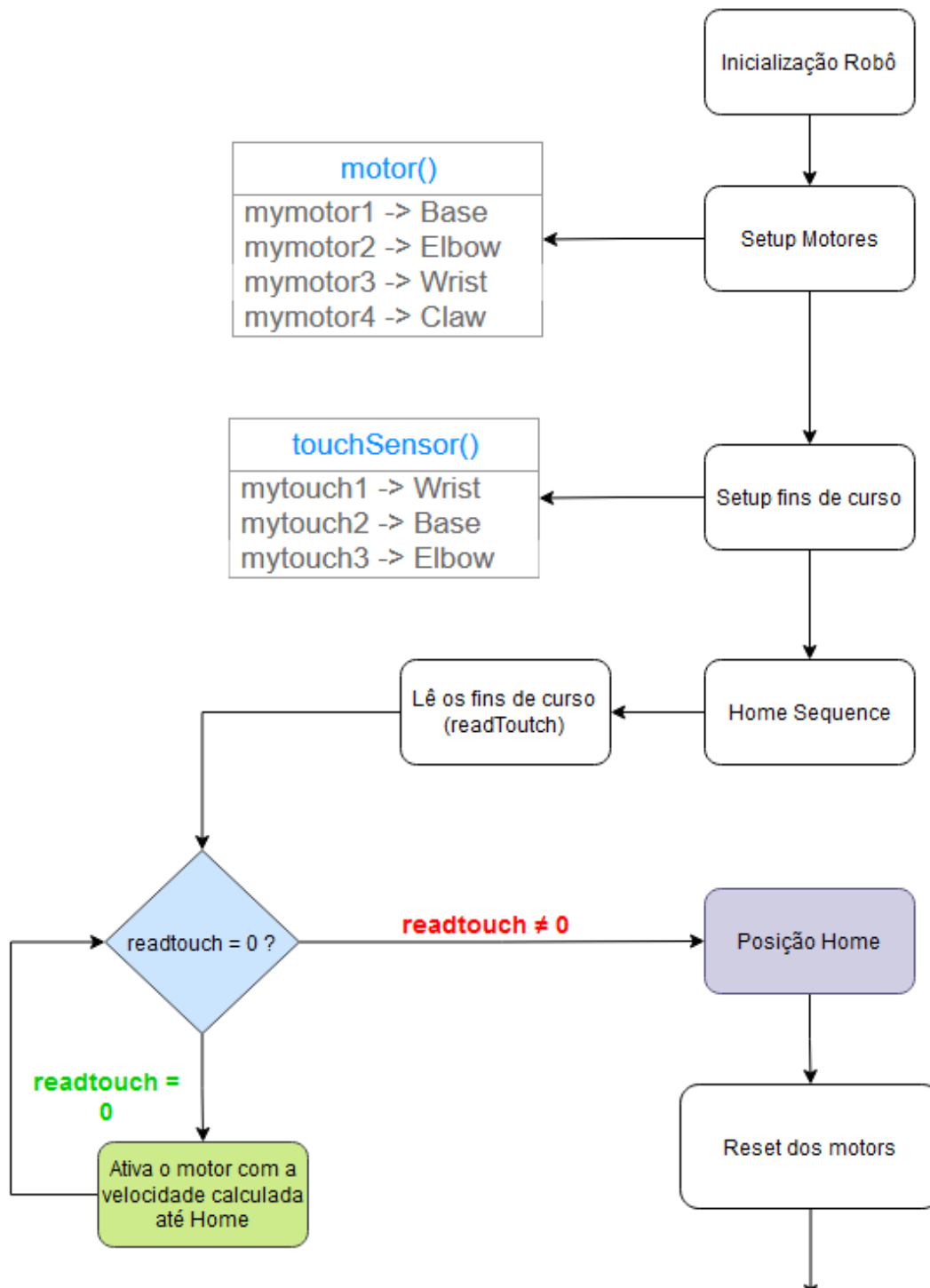
```
"% Conversão de Distancia para Ângulos"  
"valueX = (x * 90)/30; % Em X"  
"valueY = (y * 45)/21; % Em Y"  
  
"% Conversão para Ângulos dos Motores LEGO"  
"angleX = -180 + ((valueX * 350) / -90);"  
"angleY = ((valueY * 350) / 80);"
```

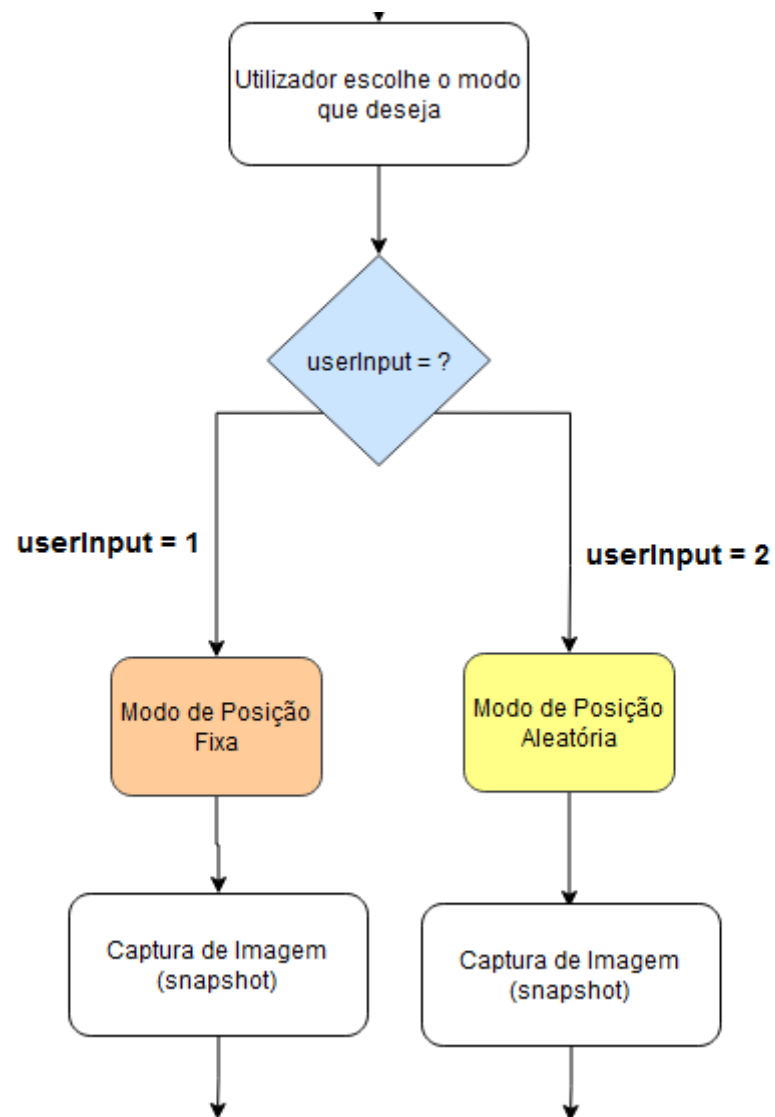
No caso do método 1 estão a ser usadas posições fixas, e depois alternativamente ao processo de extração das posições do objeto(x,y), apenas existe uma divisão da imagem, realizando de seguida uma contagem dos pixels que por sua vez irá indicar se o objeto se encontra à direita ou à esquerda.

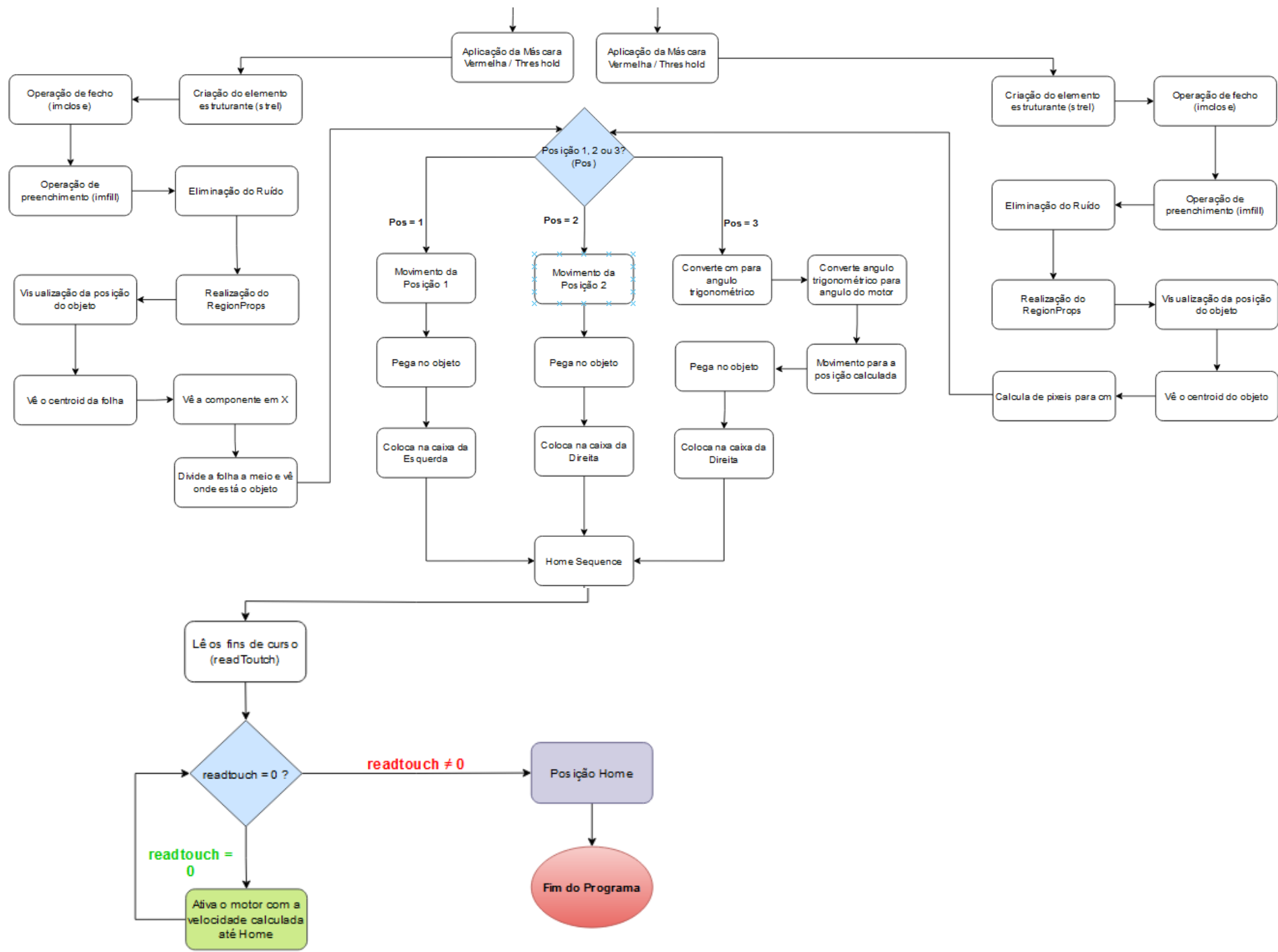
Caso esteja posicionado à esquerda é retornado a posição de colocação do objeto, posição2. Caso seja direita, é a posição 1.

```
% Divisão da Imagem e Contagem de Pixels"
"stats = regionprops(filledIm,'Centroid'); % Image Centroid"
"middleColumn = stats.Centroid(2); % Imaginary vertical line cutting the
""image in 2 pieces"
"leftHalf = floor(nnz(filledIm(:,1:middleColumn))); % Count the pixels on
""the left"
"rightHalf = floor(nnz(filledIm(:,middleColumn+1:end))); % Count the
"pixels on the right"
"if leftHalf < rightHalf"
    " pos = 1; % go to position 1"
    "disp("Right")"
"else"
    " pos = 2; % go to position 1"
    "disp("Left")"
"end"
```

8 – Fluxograma







9 - Resultado final

Comentando os resultados obtido, pode-se observar que o robô pode executar as operações que lhe foram implementadas, mesmo apesar da abordagem ter sido realizada de uma maneira não ideal, visto que essa estava a dar muitos problemas no cálculo das velocidades dos motores, devido ao peso dos elos.

Contudo, foi possível fazer duas aproximações à solução, em que uma delas é mais simples de implementação e a outra já se baseia no cálculo da posição real do objeto no mundo.

Quanto ao cálculo das trajetórias não é possível fazer essa implementação 100% correta pois como a velocidade é demasiado pequena, o robô não consegue suportar o peso e não se move. No entanto, esse método é funcional em simulação e foi validado no matlab fazendo “plot” do nosso robô.

No vídeo em anexo é possível visualizar a implementação do trabalho prático.

Método 1:

PosiçãoFixa.mp4

Método 2:

PosiçãoAleatória.mp4

Conclusão

Com a realização deste trabalho prático, foram aplicados diversos conhecimentos obtidos no decorrer da unidade curricular.

O trabalho contribuiu para o desenvolvimento de novas competências, tais como, aplicação de técnicas de processamento de imagem, estruturação de um manipulador robótico, assim como perceber como este opera, e também foram desenvolvidas competências no software MATLAB.

Ao longo do desenvolvimento tornou-se cada vez mais evidente que o aspeto mais importante para a movimentação do robot era a definição correta dos parâmetros de hartenberg, definir corretamente os limites do robot e perceber como funcionam as cinemáticas direta e inversa.

Como foi feita uma abordagem diferente daquela que é a correta, podemos observar também que tem de existir uma relação entre o peso do robot, a velocidade e a movimentação do robot.

Fora esse aspeto, o grupo foi capaz de implementar uma boa solução, objetivando sempre a simplificação do processo, a clareza e funcionalidade do código criado, completando os objetivos propostos.

Bibliografia

Bruno Oliveira – PowerPoint, “Robôs Manipuladores” – Consultado 2022/2023

Bruno Oliveira – PowerPoint , “Robôs Manipulador Industrial” – Consultado 2022/2023

Bruno Oliveira – PowerPoint , “Posição e Orientação de Corpos Rígidos” – Consultado 2022/2023

Bruno Oliveira – PowerPoint , “Cinemática e Dinâmica de Manipuladores” – Consultado 2022/2023

Bruno Oliveira – PowerPoint , “Cinemática Inversa de Manipuladores” – Consultado 2022/2023

Bruno Oliveira – PowerPoint , “Planeamento de Trajetórias” – Consultado 2022/2023

Michael Thompson, Victoria Serrano, Jordan Noriega, Vanessa Martinez - “Learning Robotic Concepts with a 3R Lego NXT Robotic Arm” - Consultado 2022/2023

Repositório - https://github.com/sampaio00joao/TP2_Robotics