Apêndice C

Experimento 3: Visualização dos dados dos estados das juntas, sinais de entrada e gravação dos dados

C.1 Visualização dos estados das juntas do UR3

Este experimento objetiva fornecer um primeiro contato do usuário ao sistema robótico do manipulador UR3. Ele permitirá que o usuário seja capaz de

- Visualização usando rqt_plot [32];
- Gravação dos dados em um arquivo rosbag [23];
- Exportação dos dados gravados para o matlab [30];

Assumindo que o leitor tenha feito o processo de setup do ambiente de experimento, como é demonstrado no Apêndice A da seção A.1.1 até a seção A.1.2, vamos direto para o experimento em si.

Como explicado na seção A.1.2, Interface de Comunicação está em funcionamento, ou seja, ela já está capturando os dados dos estados das juntas do robô UR3, publicando no tópico **/ur3/arm** e esperando uma referência de velocidade no tópico **/ur3/ref_vel**.

C.2 Visualização usando rqt_plot

Abra uma nova seção no **Terminator** e rode o comando abaixo para ir para o diretório do **workspce**.

Use o comando abaixo para ir para **workspce**:

cd ~/catkin_ur3_ws

Use o comando abaixo para fazer o diretório do **workspce** um diretório ROS.

source devel/setup.zsh.

Em seguida, vamos rodar o comando abaixo para abrir **rqt_plot**. Uma janela idêntica a Figura C.1 será aberta.

rqt_plot.

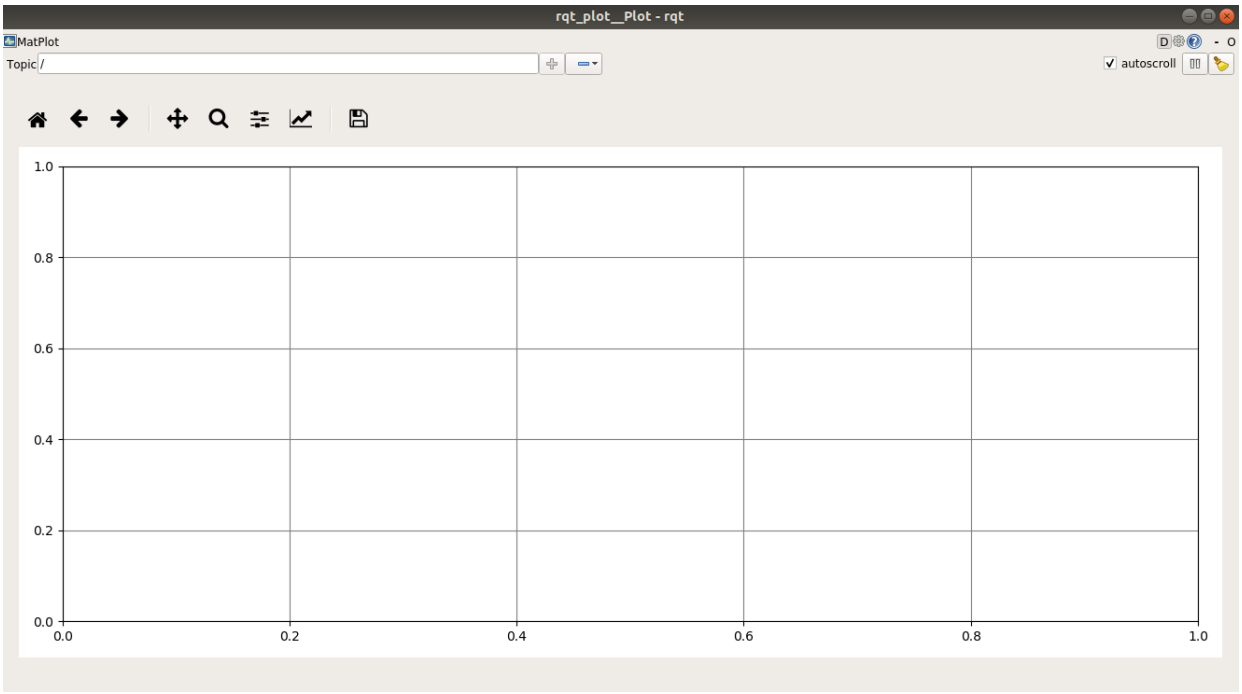

Figura C.1: rqt_plot

A interface gráfica do **rqt_plot** funciona de forma similar a um osciloscópio. O eixo das ordenadas faz a medida do sinal que é colocado como entrada (será explicado como colocar um sinal para visualização mais adiante) e o eixo das abscissa faz a medida do tempo em segundos.

C.2.1 Visualização dos estados das juntas do UR3

Como já descrito na seção 2.2.3.2, existe um tópico chamado de **/ur3/arm** (mostrado na Figura C.2 usando o comando **rostopic list**) onde podemos acessar os estados das juntas do UR3 (Posição, velocidade e Torque).

Agora, usando a interface do **rqt_plot**, vamos acessar os estados de uma das juntas, usando as legendas abaixo, sendo colocado a referência para o tópico na caixa de texto em vermelho chamada **Topic** Figura C.3.

- **i = 0**: junta **Base**

- **i = 1**: junta **Shoulder**
- **i = 2**: junta **Elbow**
- **i = 3**: junta **Wrist1**
- **i = 4**: junta **Wrist2**
- **i = 5**: junta **Wrist3**

- **Posição da junta i, com i = 0 a 5**: /ur3/arm/position[i]
- **Velocidade da junta i, com i = 0 a 5**: /ur3/arm/velocity[i]
- **Toque da junta i, com i = 0 a 5**: /ur3/arm/effort[i]

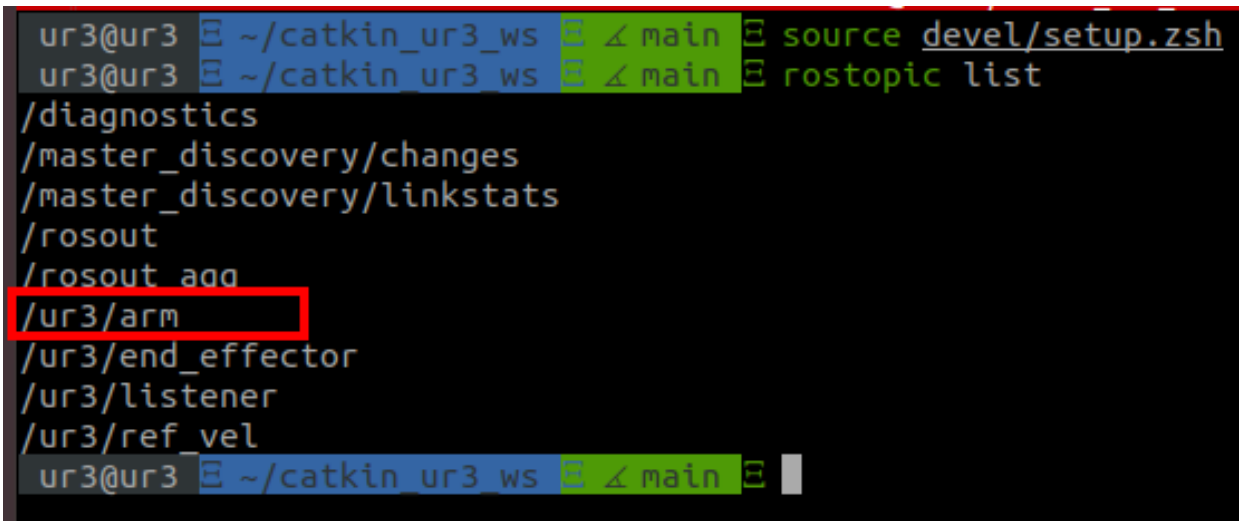

Figura C.2: Tópico /ur3/arm

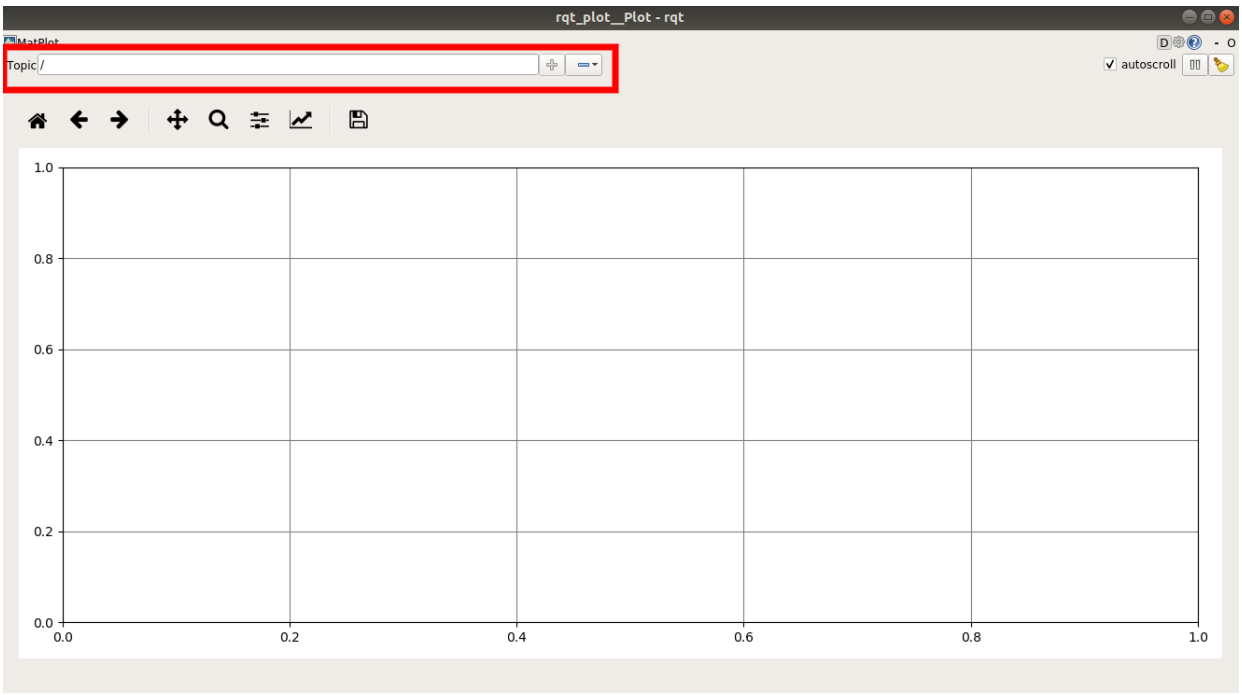

Figura C.3: caixa de texto Topic

Depois de adicionar uma das referências para uma das ondas, que no exemplo da Figura C.4 é a posição da **Base**, clique no botão demarcado pelo retângulo verde na Figura C.4.

Como a junta não está em movimento, o sinal aparecerá imóvel como mostra a Figura C.5.

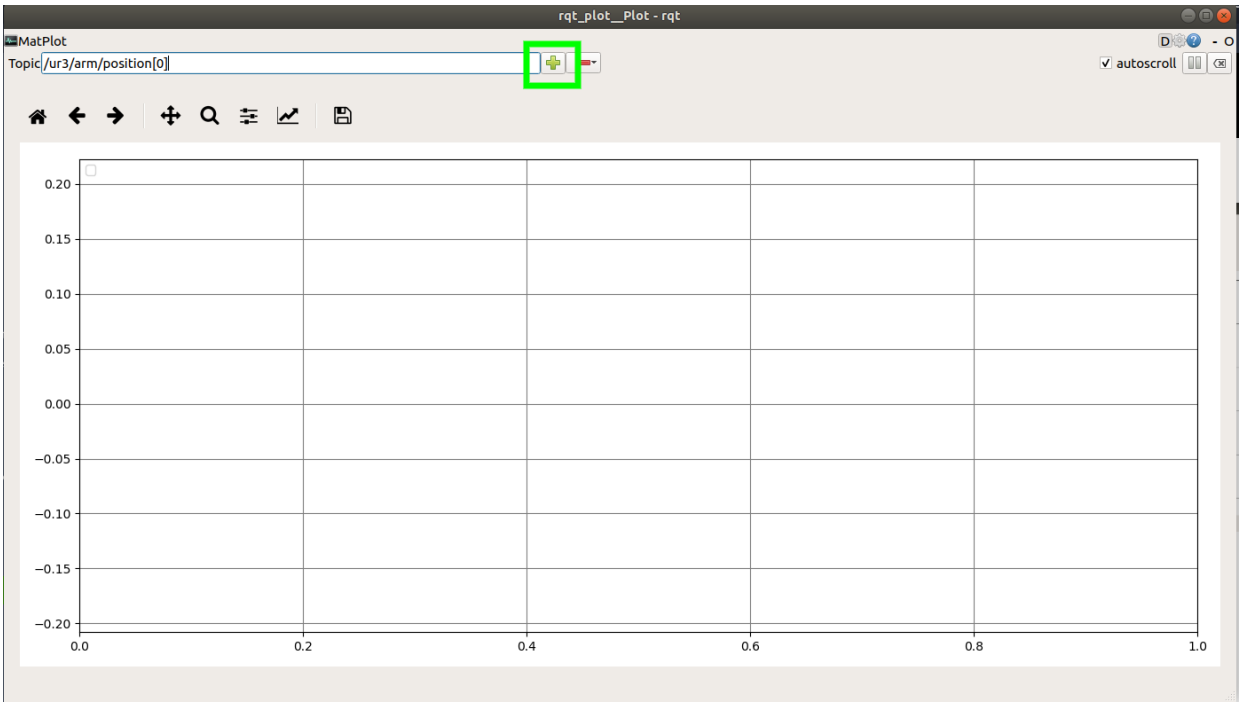

Figura C.4: Adicionando uma onda no rqt_plot

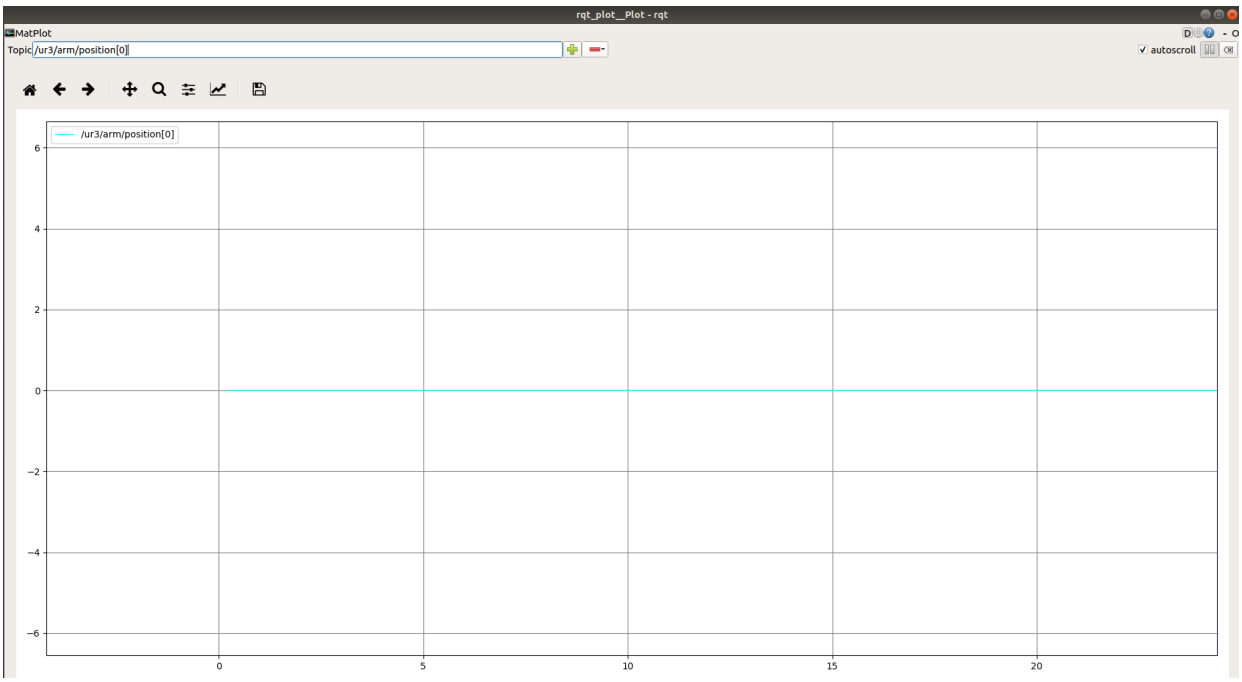

Figura C.5: Sinal de posição da Base Adicionado

Seguindo o mesmo processo usado para adicionar o sinal de posição, vamos adicionar os estados de velocidade e torque da junta da base. Para isso, basta escrever o comando abaixo na caixa de texto **Topic** e clicar no ícone mais.

Lembre-se, só é permitido adicionar um sinal por vez, ou seja, adicione o sinal de velocidade e depois adicione o sinal de torque.

ur3/arm/velocity[0]
ur3/arm/effort[0]

Depois de adicionar os sinais que foram mencionados nos parágrafos acima, abra uma nova seção no terminator e rode os comandos a seguir para fazer a junta **Base** iniciar um movimento.

source devel/setup.zsh

Na sequência inicie o nó do experimento com o comando a seguir.

roslaunch demos_ur3 exp3.launch

Neste ponto o nó **Exp3** está rodando e esperando a chamada do serviço **/demos_ur3/start_exp3**.

Para chamar o serviço **/demos_ur3/start_exp2** abra outra aba no terminal, vá para o **workspace** e insira o comando que torna essa aba um diretório ROS

source devel/setup.zsh

e rode o comando de chamada do serviço com a flag **data** como **true**.

rosservice call /demos_ur3/start_exp3 "data: true"

Para parar o experimento basta chamar o serviço **/demos_ur3/start_exp3** com a flag **data** como **false** ou esperar o experimento acabar.

rosservice call /demos_ur3/start_exp3 "data: false"

Depois desse processo o usuário verá uma imagem semelhante a da Figura C.6.

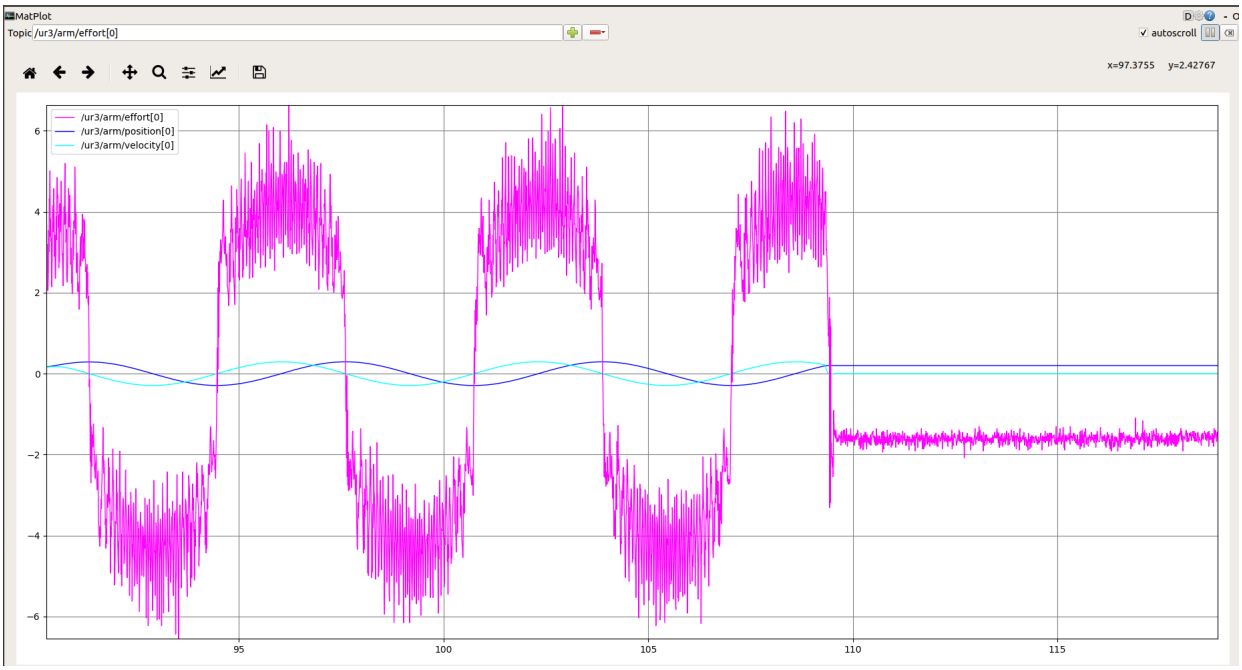

Figura C.6: Sinais de saída para junta Base

C.2.2 Visualização dos sinais de entrada

Assim como para visualizar os sinais de saída do UR3, visualizar o sinais de entrada da junta **Base** (referência de posição e sinal de controle que é uma referência de velocidade para o controlador interno do UR3), basta ir no **rqt_plot** e adicionar o sinal referente ao tópico **/ur3/ref_pos** e o sinal referente ao tópico **/ur3/ref_vel**.

Para isso, escreva os comandos abaixo na caixa de texto **Topico** do **rqt_plot** e clique no ícone mais.

/ur3/ref_pos/data[0]
/ur3/ref_vel/data[0]

Depois desse processo o usuário verá uma imagem semelhante a da Figura C.7.

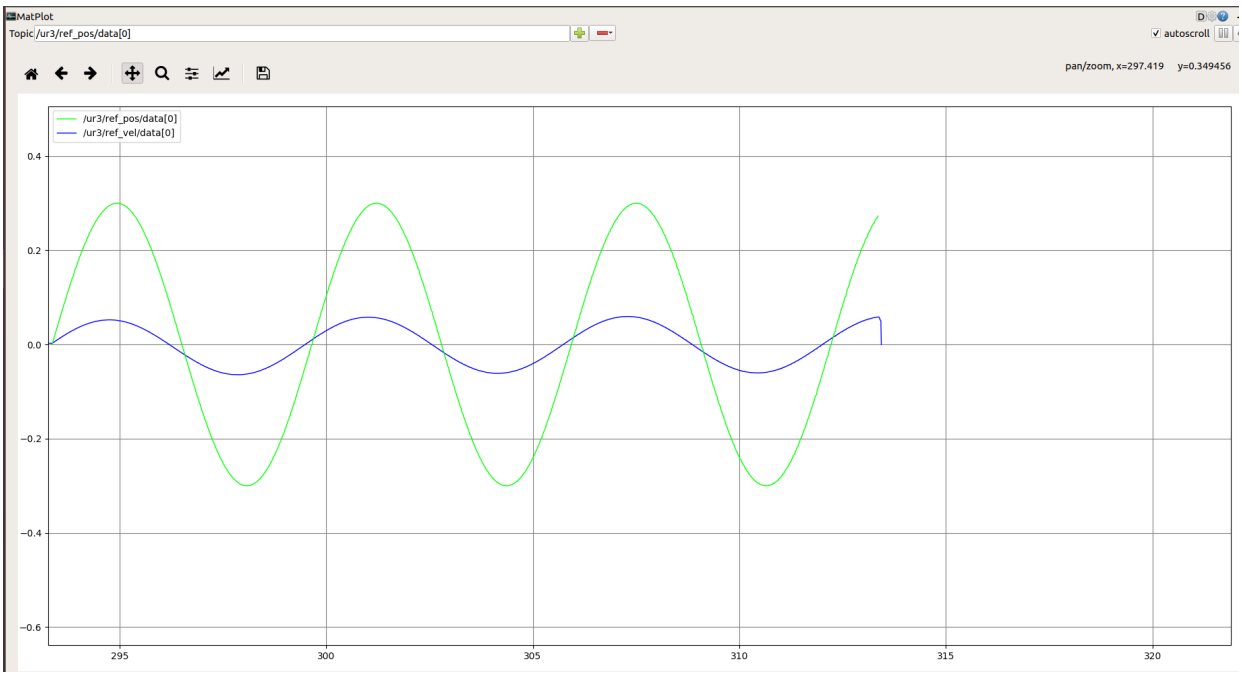

Figura C.7: Sinais de entrada para junta Base

Caso o usuário insira todas as ondas (entrada e saída da junta Base), ele verá uma imagem semelhante a da Figura C.8.

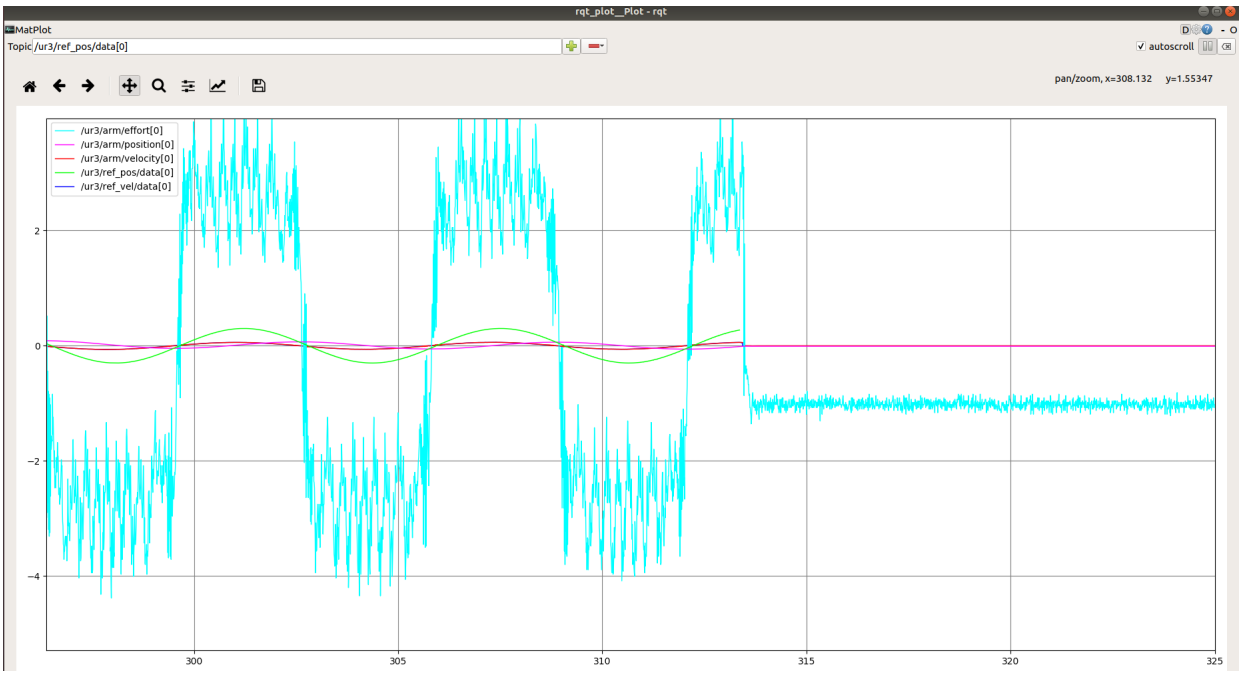

Figura C.8: Sinais de entrada e saída para junta Base

C.2.3 Gravação dos dados em um arquivo rosbag

Para realizar o gravação dos dados em um sistema de arquivo **bag**, primeiro vamos esclarecer qual os tópicos ROS que são de interesse apara uma analise mais geral dos sinais de entrada e saída do UR3.

Nos itens abaixo, encontra-se a discrição de cada tópico de interesse. Para mais detalhe, consulte o capítulo referente a Interface de Comunicação 2.2.3.2.

- **ur3/arm**: nesse tópico está contido todos os sinais de saída referente as juntas do UR3 (Posição, Velocidade e Torque).
- **ur3/ref_pose**: o tópico **ur3/ref_pos** armazeno os os sinais de referência de posição para todas as juntas do UR3.
- **ur3/ref_vel**: o tópico **ur3/ref_vel** armazeno os os sinais de referência de velocidade para todas as juntas do UR3.
- **/ur3/end_effector**: este tópico armazena os dados do end_effector do UR3, são eles:
  - **abertura da pinça do end_effector**
  - **posição no espaço tridimensional com relação a base do UR3**
  - **orientação no espaço tridimensional com relação a base do UR3**
  - **velocidade linear no espaço tridimensional com relação a base do UR3**

Como dito antes, os dados gravados serão armazenados em um arquivo **bag**[23] e serão armazenados, com o prefixo **exp**, na pasta **bags** (Figura C.9) presente no **workspace catkin_ur3_ws** com a nomenclatura sendo prefixo mais a data e hora de realização do experimento.

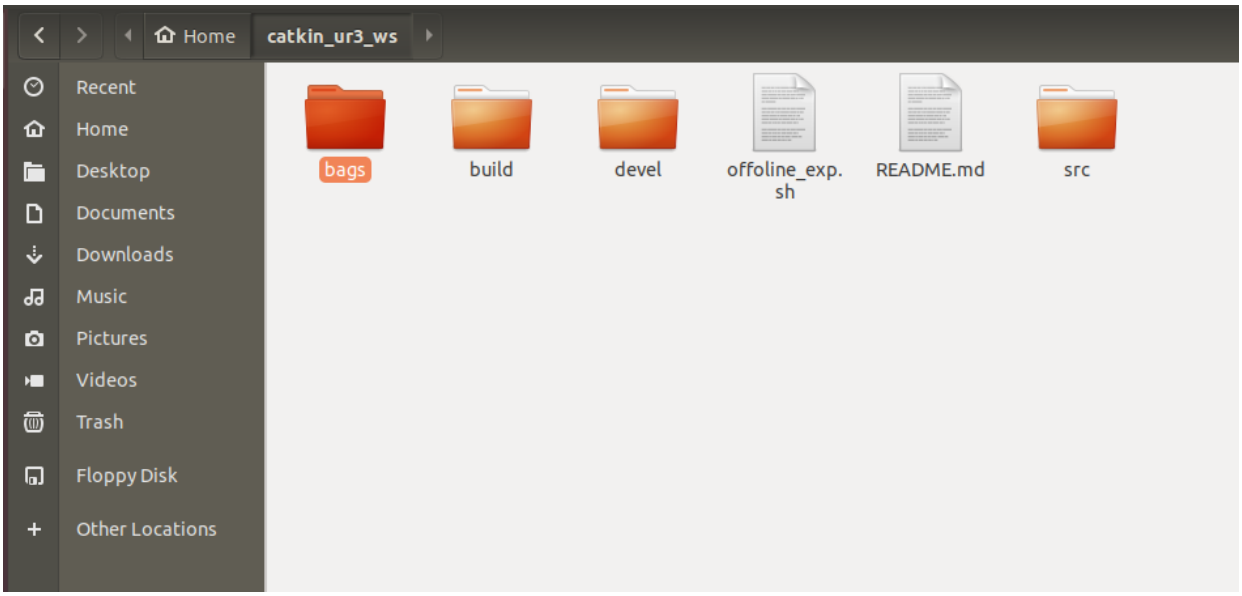

Figura C.9: Pasta bags selecionada

Para gravar os dados do experimento, abra uma nova seção no terminator, vá para o **workspace** e insira o comando que torna essa aba um diretório ROS.

source devel/setup.zsh

Inicie a gravação dos dados com o comando abaixo.

rosbag record -o bags/exp /ur3/arm /ur3/ref_pos /ur3/ref_vel /ur3/end_effector

Para parar a gravação dos dados, pressione Ctrl+C na mesma seção do terminator que foi iniciado a gravação. Depois disso, aparecerá um arquivo com os dados do experimento como mostrado na FiguraC.10 (nesse exemplo experimento foi realizado na data 25/09/2021 às 10:47:32)

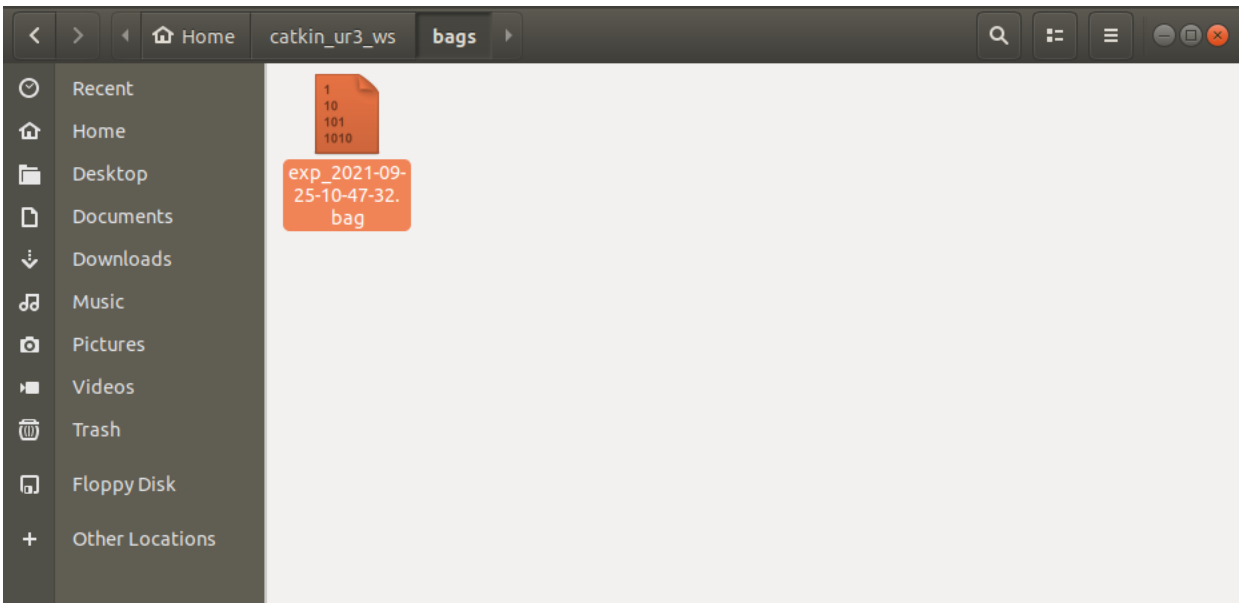

Figura C.10: Pasta bags selecionada

C.2.4 Exportação dos dados gravados para o matlab

Para fazer a exportação dos dados gravados pelo ROS para o **matlab** da forma mais simples possível, foi feito um script na linguagem matlab para agilizar o processo de exportação de dados.

O script matlab é nomeado com **export_rosbag** e se encontra na pasta **matlab_scripts** dentro do **workspace** como mostra a Figura C.11.

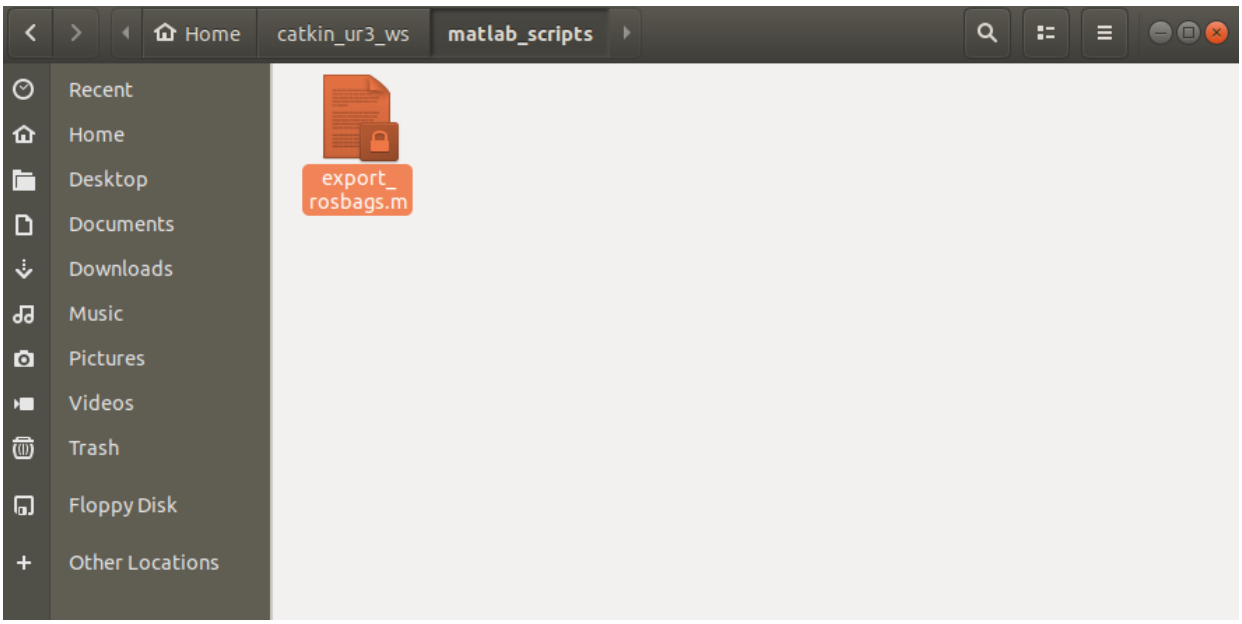

Figura C.11: Export rosbag

Para fazer a exportação dos dodos para o matlab, abra o matlab e rode o script **matlab_scripts** no formato descrito no comando abaixo.

[time, input_pose, input_vel, output_pose, output_vel, output_effort, end_effector] = export_rosbags('exp_2021-09-25-10-47-32')

A FiguraC.12 mostra como é a execução do comando acima (retângulo em vermelho) e as variáveis exportadas (retângulo em verde).

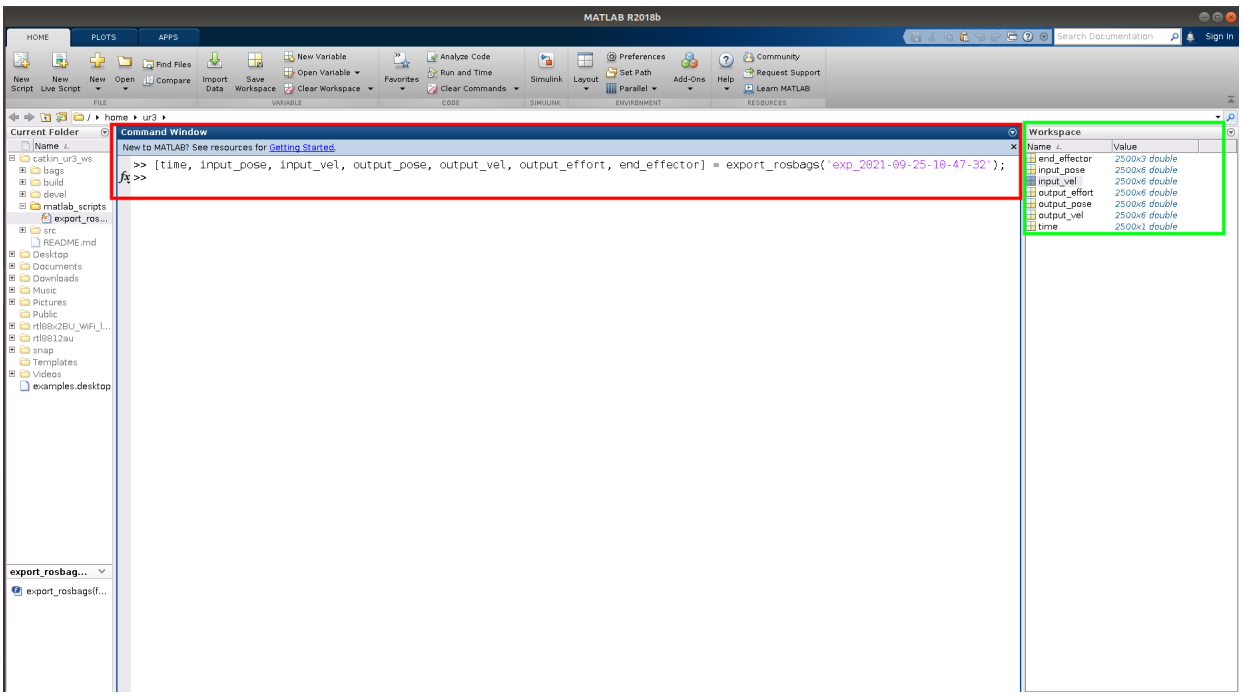

Figura C.12: Comando para exporta os dados para o matlab

Descrição de cada variável do workspace do matlab (retângulo verde na Figura C.12).

- **time**: vetor com os valores de tempo discretizado a cada 0.008 segundos.
- **input_pose**: vetor entrada de posição angular de tamanho 6, em radianos, para cada junta do UR3, sendo o primeiro elemento a posição da junta da base.
- **input_vel**: entrada de velocidade angular de tamanho 6, em rads/s, para cada junta do UR3, sendo o primeiro elemento a velocidade da junta da base.
- **output_pose**: saída de posição angular de tamanho 6, em radianos, de cada junta do UR3, sendo o primeiro elemento a posição da junta da base.
- **output_vel**: saída de velocidade angular de tamanho 6, em rads/s, de cada junta do UR3, sendo o primeiro elemento a velocidade da junta da base.
- **output_effort**: saída de torque de tamanho 6, em N.m, de cada junta do UR3, sendo o primeiro elemento a torque da junta da base.
- **end_effectror**: vetor de de saída da posição tridimensional do end_effector com tamanho 3 com relação a base do UR3. A primeira posição do vetor é a coordenada x, a segunda y e a terceira a coordenada z, todas com a unidade em metros.