

Apêndice B

Experimento 2: Usando controlador do Capítulo 4

Assumindo que o usuário tenha feito o processo de *setup* do ambiente de experimento como é feito no Apêndice A da seção A.1.1 até a seção A.1.2, vamos direto para o experimento em si. Feito os procedimentos da seção A, abra uma aba do terminal e vá para o **workspce** do UR3 que nada mais é a pasta **catkin_ur3_ws**.

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.
```

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**.

B.1 Gerando velocidades de referência para o UR3

Neste seção vamos explicar o processo de como gerar uma referência de velocidade para umas das juntas do robô UR3. Para isso, será necessário que o usuário tenha em mãos a ferramenta Matlab para rodar o código mostrado abaixo (o Matlab está instalado no computador reservado para o UR3).

B.1.1 Explicação do *script* de geração de velocidade

```
1 % Arquivo para gerar referencia de velocidade para as juntas do robo UR3
2 clear ;
3 close all;
4 periodo = 2; % Em pi rad
5 taxa_de_amostragem = 125; % Em Hz do processo
6 duracao = 4; % Tempo de duracao da onda de entrada em pi segundos
7 amplitude = 0.5; % Amplitude da onda de referencia em rad
8 % Foi usado a funcao senoide, mas pode ser a funcao de sua preferencia.
9 % Evite gerar referencias que tenham trasicoes rapidas
10 tempo = (0:(1/taxa_de_amostragem):duracao*pi);
11 ref_vel = amplitude*sin(periodo*tempo);
12 % ref_vel ser a vari vel que cont m os dados de velocidade para o robo
13 figure('units','normalized','outerposition',[0 0 1 1]);
14 plot(tempo, ref_vel, 'LineWidth',1.5); % plotando a referencia de velocidade
15 grid on ;
16 ylabel('Amplitude(rad)'); xlabel('tempo(s)')
17 lg = legend('Refer ncia de velocidade'); lg.FontSize = 11;
18 ti = title('Onda senoidal de refer ncia '); ti.FontSize = 20;
19 csvwrite('ref_vel.csv', ref_vel);% escrevendo em um arquivo do tipo csv
```

Nesse código existem alguns parâmetros que o usuário pode alterar conforma sua necessidade. São eles:

- Período da função seno (linha 4 do código matlab)
- O tempo de duração da referência de velocidade (linha 6 do código matlab)
- A amplitude do sinal (linha 5 do código matlab)
- A função geradora do sinal (linha 11 do código matlab) que nesse caso é a função seno.

Após rodar esse *script* Matlab, será mostrado uma imagem semelhante Figura B.1 onde é plotado os dados de referência de velocidade.



Figura B.1: Referência de Velocidade

Na linha 19 do *script* Matlab é onde acontece a criação do arquivo que será usado pela Interface de Comunicação 2.2.3 , que nesse caso chama-se **ref_vel.csv**, e pode ser visto na pasta que é usado no Matlab exemplificada na FiguraB.2 .



Figura B.2: ref_vel.csv no diretório do Matlab

B.1.2 Movendo o arquivo CSV para pasta csv do pacote demos_ur3

Dentre os pacotes da Interface de Comunicação, foi projetado um pacote que é específico para enviar referências de velocidade para o UR3. Para isso, basta utilizar o arquivo .csv, gerado no passo anterior, e colocá-lo dentro do pacote **demos_ur3** na pasta **csv** (pasta selecionada na Figura B.3). Para realizar todo o manuseio do robô e da Interface de Comunicação para o experimento de envio e coleta de dados, basta seguir os passos do Apêndice A.

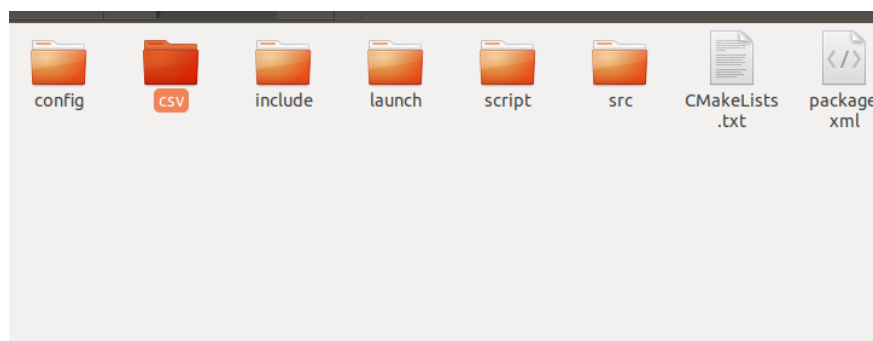


Figura B.3: Pasta csv selecionada

B.1.3 Alterando o arquivo de setup

Agora vamos informar para o pacote **demos_ur3**, nas configurações, que o arquivo que queremos que seja lido para mandar as referências de velocidade, seja o arquivo que acabamos de criar na seção anterior B.1.2. Para isso, basta ir na pasta **config** (pasta selecionada na Figura B.4) do pacote **demos_ur3** e mudar a variável **csv_name** que se encontra no arquivo **setup1.yaml** para **ref_vel.csv** como mostra a Figura B.5.

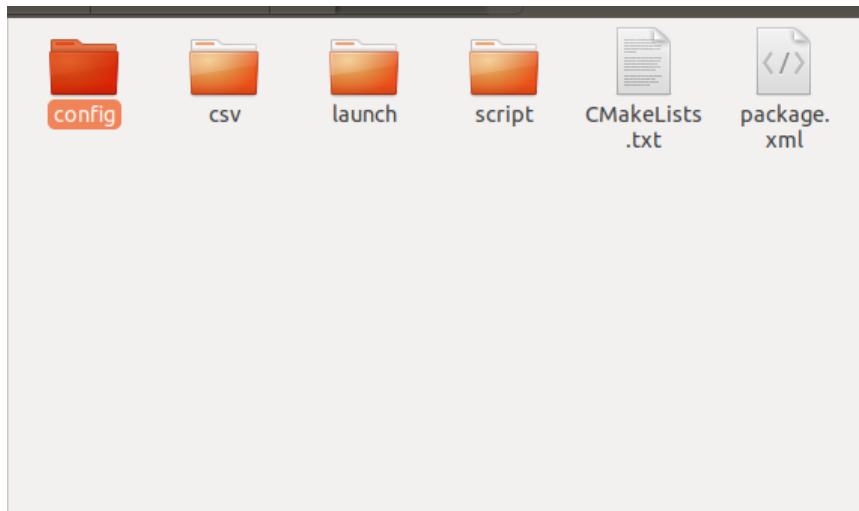


Figura B.4: Pasta config selecionada

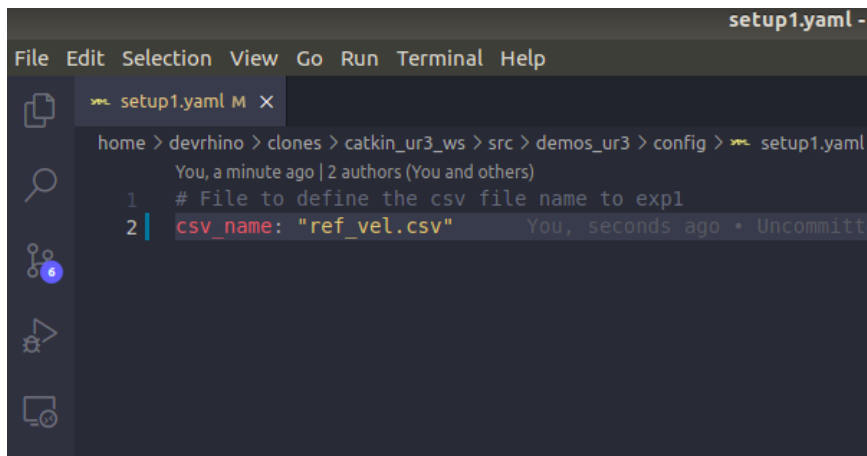


Figura B.5: Arquivo setup1

Feito isso, basta executar os mesmos comandos do Apêndice A referente a inicialização da Interface de Comunicação e execução do experimento.

B.2 Usando o controlador

Neste experimento o usuário tem a opção de enviar ondas de referência de posição de forma offline, como é feito no experimento 1 mostrado no Apêndice A, ou mandar o comando especificando a posição da junta via terminal.

No arquivo **setup2.yaml** (Figura B.7), localizado na pasta **config** (pasta selecionada na Figura B.6) do pacote **demos_ur3**, há alguns parâmetros que serão comentados em itens a seguir.

- **csv__name**: No parâmetro **csv__name** o leitor deve escrever no sub-parâmetro **your__wave**

o nome de um arquivos **csv** que tenha ondas de referência de posição geradas, explicado no seção B.1, para realizar seu próprio experimento. Os sub-parâmetros **sine** e **square** não devem ser alterados, pois foram feitos para fazer demonstrações de funcionamento do UR3.

- **ref_type**: No parâmetro **ref_type** o usuário decide se quer que as mensagens de referência, para as juntas, sejam enviadas online ou offline, ou seja, caso o sub-parâmetro **online** seja dado com **True** o usuário enviará a referência de posição pelo terminal. Por outro lado, caso o sub-parâmetro **online** seja dado com **False**, o sub-parâmetro **type** irá informar qual onda de referência de posição será enviada para o UR3, que na Figura B.7 mostra que é a senoidal.

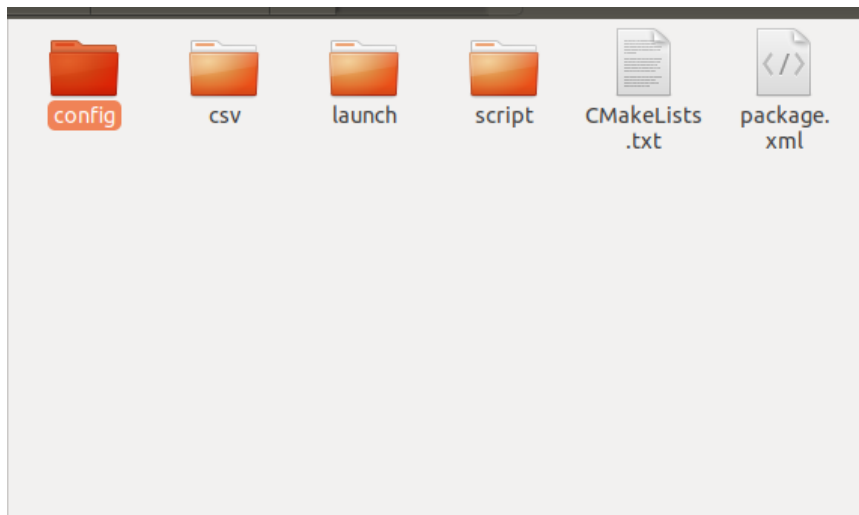


Figura B.6: Pasta config selecionada

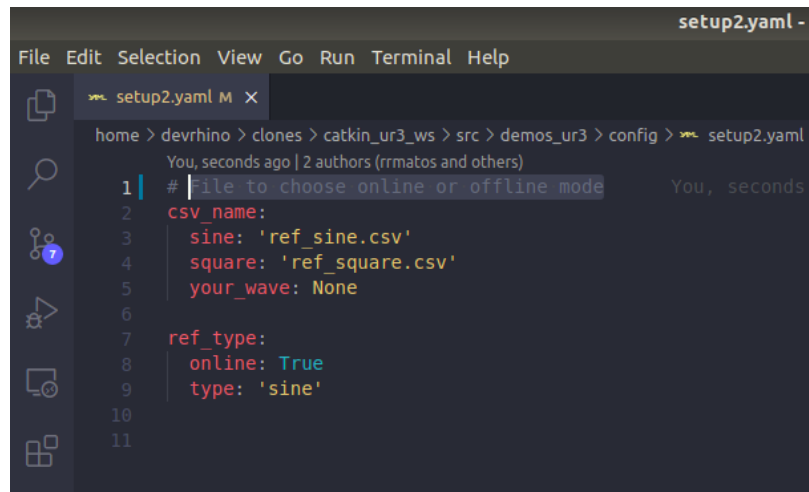


Figura B.7: Arquivo de setup para o experimento 2

B.2.1 Experimento offline

No experimento **offline** o usuário deve colocar o sub-parâmetro **online** como **False**, escolher qual onda de referência será enviada ao UR3 e colocá-la no sub-parâmetro **type**. As opções de onda de referência são:

- **sine**: Onda de referência será uma senoidal.
- **square**: Onda de referência será uma onda quadrada.
- **your__wave**: Uma onda gerada pelo próprio usuário.

Para este experimento foi feito, em **Python**, um nó chamado **exp2** em ROS para publicar as referências de velocidade para o robô. O arquivo em questão é chamado de **demo_exp2.py**, pertencente ao pacote **demos_ur3**. Para ser executado, abra outra aba no terminal, vá para o **workspace** e digite o comando que torna essa aba um diretório ROS.

```
source devel/setup.zsh
```

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

```
roslaunch demos_ur3 exp2.launch
```

Neste ponto o nó **Exp2** está rodando e esperando a chamada do serviço **/demos_ur3/start_exp2**.

Para chamar o serviço **/demos_ur3/start_exp2** abra outra aba no terminal, vá para o **workspace** e digite 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_exp2 "data: true"
```

Para parar o experimento basta chamar o serviço **/demos_ur3/start_exp2** com a flag **data** como **false** ou esperar o experimento acabar.

```
rosservice call /demos_ur3/start_exp2 "data: false"
```

B.2.2 Experimento online

No experimento **online** o usuário deve colocar o sub-parâmetro **online** como **true**. Feito isso, o usuário não precisa se preocupar com os outros parâmetros.

Para este experimento foi feito, em **Python**, um nó chamado **exp2** em ROS para publicar as referências de velocidade para o robô. O arquivo em questão é chamado de **demo_exp2.py**, pertencente ao pacote **demos_ur3**, e para ser executado, abra outra aba no terminal, vá para o **workspace** e digite o comando que torna essa aba um diretório ROS.

```
source devel/setup.zsh
```

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

```
roslaunch demos_ur3 exp2.launch
```

Neste ponto o nó **Exp2** está rodando e esperando a chamada do serviço **/demos_ur3/start_exp2**.

Para chamar o serviço **/demos_ur3/start_exp2** abra outra aba no terminal, vá para o **workspace** e digite 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_exp2 "data: true"
```

O envio de referência via terminal também é feito via serviço do ROS. O nome do serviço que envia os dados de referência de posição é **/demos_ur3/target_position** e nele o leitor teve passar como argumento do parâmetro **newFloat** a posição, em radianos, para a junta do UR3.

O comando a seguir é um exemplo de como o serviço é chamado passando uma posição de referência de 0.2 rad.

```
rosservice call /demos_ur3/target_position "newFloat: 0.2"
```

Para parar o experimento basta chamar o serviço **/demos_ur3/start_exp2** com a flag **data** como **false** ou esperar o experimento acabar.

```
rosservice call /demos_ur3/start_exp2 "data: false"
```

Depois disso é esperado que o robô UR3 faça um movimento repetitivo parecido com o movimento do seguinte vídeo que está postado no youtube:

https://youtu.be/pkN1GS0wX_o

FIM!