

## Atividade de Drones

Esta atividade conta como opcional em sua nota de atividades de grupo. Pode ser enviada até 30/04.

### Execução

Se você usa a imagem do Ubuntu 18.04 fornecida pela disciplina, use o guia de execução

### Instalação

Caso seja uma instalação sua do Linux, vá para o guia de instalação

### Simulador

Sobre como executar o simulador de drones Bebop (Ubuntu 16.04 somente) . Peça aos técnicos ou professor a máquina virtual do VirtualBox que já tem o simulador funcionando.

Vídeo com instruções do simulador - parte 1 Vídeo com instruções do simulador - parte 2

### Informações de segurança

Todos os que estiverem na quadra junto com o *drone* precisam usar equipamento de segurança: \* Óculos de proteção \* Sapato fechado e antiderrapante \* Calça comprida \* Vestimenta que cubra os braços

Além disso **ninguém deve tocar um drone em operação**

Tenha sempre um terminal (não pode ser uma aba) dedicado ao comando de parar e outro ao comando de pouso

## Comandos importantes

### Conectar

Antes anote o número de série do *drone* e desabilite ROS\_MASTER\_URI e ROS\_IP no `.bashrc`

```
roslaunch bebop_driver bebop_node.launch
```

## Decolar

```
rostopic pub --once bebop/takeoff std_msgs/Empty
```

## Teleop

```
roslaunch teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=bebop/cmd_vel
```

## Pousar

Tenha sempre este comando em seu histórico

```
rostopic pub --once bebop/land std_msgs/Empty
```

## Parar o drone

Tenha este comando em seu histórico e numa janela dedicada

```
rostopic pub -1 /bebop/cmd_vel geometry_msgs/Twist -- '[0.0, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
```

## Fazer flip

Para fazer o flip o *drone* deve estar voando a pelo menos 2.5m de altura

```
rostopic pub --once /bebop/flip std_msgs/UInt8 "0"
```

## Precauções

Bater o drone em objetos rígidos é **uma tragédia**. O equipamento pode ser danificado permanentemente

O grupo deve adotar uma estratégia de resolução de problemas que preserve o equipamento

Na quadra, faça vôos em direção às redes do gol enquanto estiver testando

Use baixas velocidades de deslocamento

Antes de fazer uma missão totalmente autônoma valide muito bem suas leituras de sensores e parâmetros (velocidades e tempos)

## Atividades

Os arquivos `.py` precisam ter o nome dos integrantes do grupo

### 1. Teleop e foto

Faça o *teleop* do seu drone e *capture* uma imagem a partir da câmera do drone em que se vê seu grupo

Nota: Pode ser com o `rqt_image_view` mesmo

### 2. Modificação de código

Clone <https://github.com/insper/robot19> dentro de seu `~/catkin_ws/src` e dê um `'catkin_make'`

Para rodar os exemplos faça:

```
roslaunch insperbop takeoff.py
```

```
roslaunch insperbop fly.py
```

Altere o código fornecido na pasta `drones/insperbop/scripts` para fazer um script que faz o drone decolar, avançar 6m e depois parar. Você *pode* fazer em malha aberta, mas se houver tempo use o tópico de odometria `/bebop/odom` como mostrado no exemplo `print_odom.py`

```
roslaunch insperbop print_odom.py
```

**Atenção:** as coordenadas  $x$ ,  $y$ ,  $z$  de odometria **não coincidem** com as do drone

Dê o comando a seguir para entender como funciona a odometria enquanto estiver fazendo *teleop*

```
rostopic echo /bebop/odom
```

### 3. Avançado

Faça o drone abaixar sua altitude um pouco e pousar após ter decolado e voador 6m.

Não precisa usar o `/bebop/odom`

#### **4. Encerramento**

Pergunte se o professor tem algo para você

#### **Entrega**

Mostre seu programa funcionando para um dos professores ainda durante a atividade.

Entregue os arquivos `.py` e a foto do grupo via Blackboard. Todos os integrantes precisam entregar.