



Robótica Espacial

Aula prática nº 8 - Componente experimental

Introdução ao uso e programação do robô UR3e
Atividade para realizar em grupo com os robôs reais

Vitor Santos

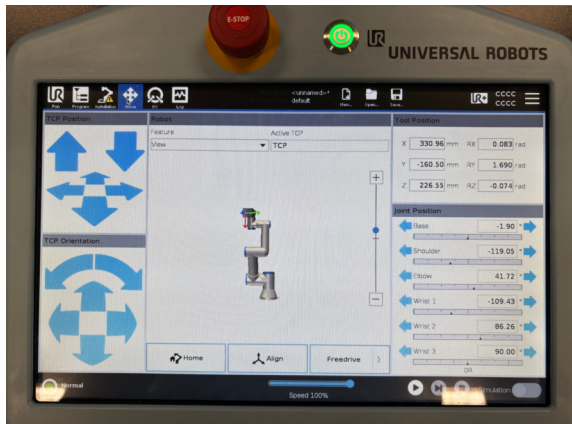
Universidade de Aveiro

3 Abril 2025

- 1 Primeiro contacto com o robô UR3e
- 2 Programa de movimentação simples no robô UR3e
- 3 Movimento condicionado pelo contacto
- 4 Programa com uso da garra no robô UR3e

Exercício 1 - Interação com o robô através da consola

- Ligar o robô UR3e, aguardar pelo arranque e ativar o braço na consola
- Mover o robô junta a junta (*jog*) até chegar a 5 cm da mesa na vertical
- Voltar ao ponto de partida e repetir o *jog* mas no sistema de coordenadas global
- Repetir o movimento em modo "free-drive"

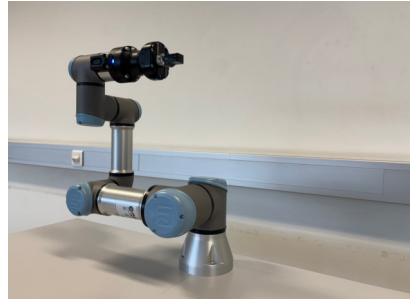
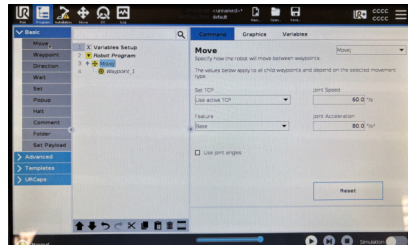


Exercício 2 - Programa de movimento em espaço de juntas e espaço cartesiano no robô UR3e

- O objetivo do exercício é que os estudantes programem o robô UR3e utilizando o *teach pendant* (a consola de programação do robô) para ele executar diversos movimentos.
- O programa guiará o robô para:
 - ➊ mover-se para uma posição inicial pré-definida com movimento em espaço de juntas;
 - ➋ executar movimentos em espaço de juntas;
 - ➌ seguir um caminho linear em espaço cartesiano;
 - ➍ repetir o ciclo em contínuo.
- Também se pode incluir a ida para posição de "Home".

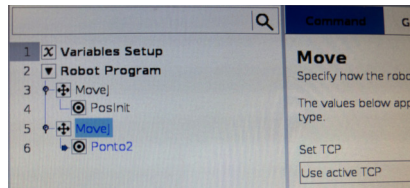
Exercício 2a) - Primeiro Waypoint em Juntas

- Aceder ao *Teach Pendant* do UR3e, navegar até ao ecrã **Program** para iniciar a criação de um novo programa.
- Definir uma posição inicial, inserindo um comando **MoveJ** (movimento em juntas) para mover o robô para uma posição de partida com os seguintes valores:
 - $q1 = 0^\circ$, $q2 = 0^\circ$, $q3 = -90^\circ$, $q4 = -90^\circ$, $q5 = -270^\circ$, $q6 = 90^\circ$
 - onde $q1$ é a junta da base, $q2$ o ombro, $q3$ o cotovelo, etc.
 - Esta posição inicial será um *WayPoint* que deverá ser chamado "PosInit"
- Estabelecer a velocidade e a aceleração para valores moderados para uma operação segura (ex: velocidade = $60^\circ/\text{s}$, aceleração = $80^\circ/\text{s}^2$).



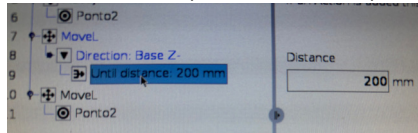
Exercício 2b) - Inserção de um segundo Waypoint em juntas

- Inserir um segundo comando **MoveJ** para mover o robô para uma posição de passagem:
 - $q1 = -90^\circ$, $q2 = -95^\circ$, $q3 = -52^\circ$, $q4 = -123^\circ$, $q5 = -270^\circ$, $q6 = 0^\circ$
 - Esta posição será um *Waypoint* que deverá ser chamado "Ponto2"
 - Manter a velocidade e a aceleração em valores similares aos anteriores.
- Neste momento é possível executar o programa de forma contínua entre estes dois *Waypoints*: o inicial "PosInit" e o "Ponto2".
- Todavia, deve-se fazer primeiro em simulação na consola e só depois executar no robô real!

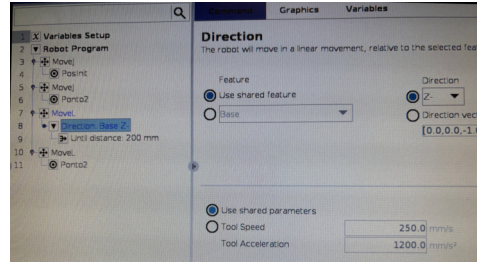


Exercício 2c) Inserir movimento linear

- Adicionar um movimento em espaço cartesiano inserindo um comando **MoveL** (movimento linear) para mover a ponta em linha reta sem alterar a sua orientação:
 - movimento na direção negativa dos Z com a distância limite (**Until Distance**) de 200 mm.

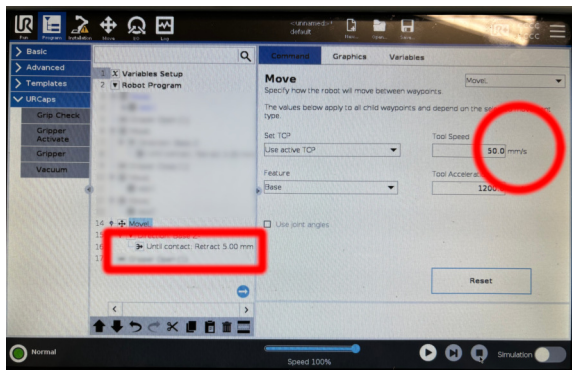


- Adicionar um segundo comando linear para regresso ao "Ponto2"
 - Este movimento é linear mas já não é dado por uma distância mas sim um *Waypoint*
- Pode ser recomendado limitar a velocidade e a aceleração da ponta.



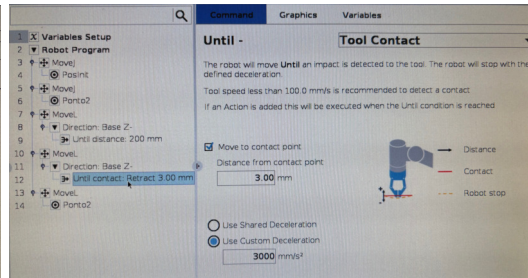
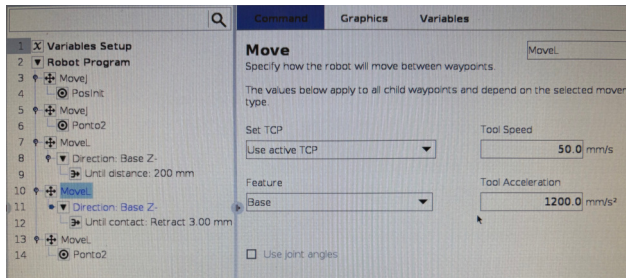
Exercício 3 - Movimento até haver contacto

- As instruções mais delicadas são as de aproximação até a garra fazer contacto.
- A deteção do contacto é possível porque o UR3e é um robô colaborativo.
- Estes movimentos devem ser feitos a muito baixa velocidade e devem contemplar uma retração da garra para não perturbar o fecho ou abertura depois do contacto
- São movimentos lineares ao longo de uma direção (Z+ ou Z- neste exercício)



Movimento até haver contacto - detalhes

- Inserir uma instrução de movimento linear na direção Z- até haver contacto.
 - MoveL - Tool Speed: **⚠50 mm/s⚠**
 - Direction Base Z-
 - Add Until – tool contact
 - Retract 3 mm
- Essa instrução deve ficar entre os dois movimentos lineares criados antes.



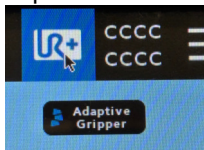
Movimento até haver contacto - Resultado

- Para testar o contacto, colocar um pequeno objeto no percurso da garra, como ilustrado:

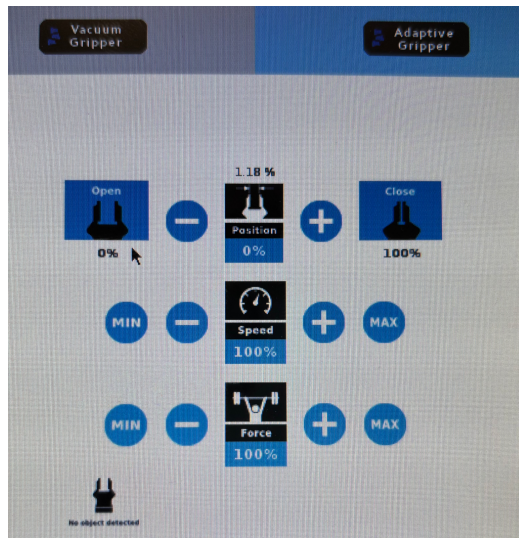


Utilização da garra

- Antes de usar a garra pela primeira vez no programa é preciso fazer a sua ativação.
- Aceder ao menu UR+ no canto superior direito da consola:

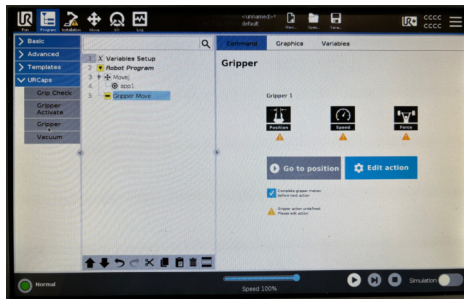


- Depois é preciso ativar a garra e podem-se fazer testes de fecho e abertura.



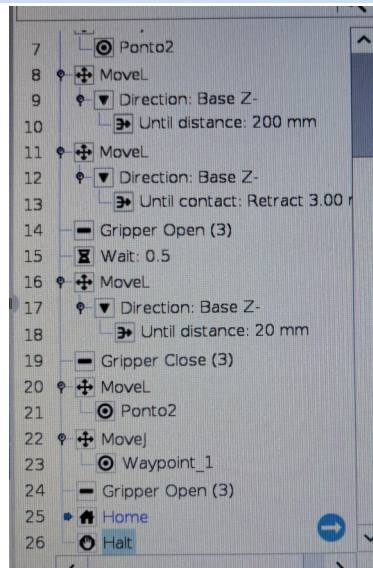
Exercício 4 - Incluir atuação da garra do robô

- Após o movimento com contacto, abrir a garra.
- A atuação da garra introduz-se a partir do menu dos URCaps - opção Gripper
- Deve-se definir qual a ação a desenvolver (*edit action*) e fazer o "save"
- Antes de avançar para o próximo movimento é recomendado fazer uma pequena pausa (Wait) para dar tempo à garra de abrir (0.5 s será em princípio suficiente)
- Para o programa poder funcionar bem em contínuo recomenda-se colocar uma instrução preventiva do fecho da garra no início do programa (primeira instrução, por exemplo)

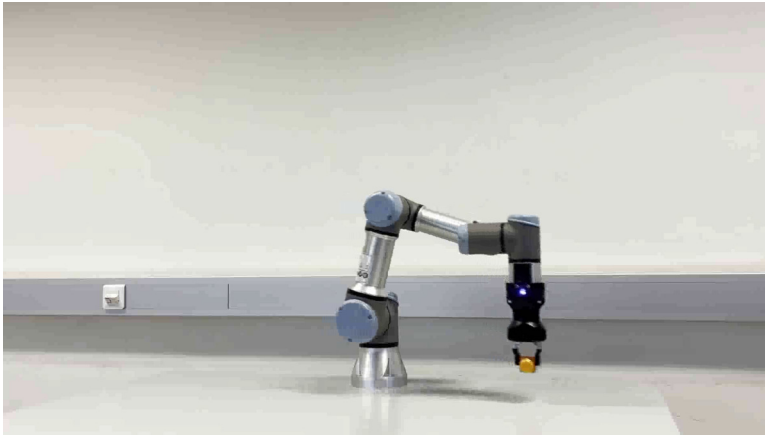


Exercício 5 - Apanhar o objeto e reposicioná-lo

- Para completar o programa, a tarefa é pegar no objeto e colocá-lo num *waypoint* a indicar e terminar efetivamente o programa sem repetição.
- As instruções específicas são enumeradas a seguir.
- Após abertura da garra e do tempo de espera proposto, inserir instruções para:
 - Avançar 20 mm em direção ao objeto (Z-)
 - Fechar a garra
 - Voltar ao "Ponto2" (esta instrução já existe)
- Criar um movimento de juntas (MoveJ) para o *waypoint* atribuído a cada grupo no e-learning.
- Libertar a peça nesse *waypoint* (abertura da garra)
- Fechar a garra
- Voltar à home position com a instrução Home.
- Terminar o programa com o comando Halt.



Resultado final completo



- O Waypoint final de destino deve ser previamente configurado para cada grupo.
- Esse *Waypoint* pode ser descarregado do e-learning e é diferente para cada grupo.
- Para confirmar que o grupo conseguiu concretizar o *waypoint*, pelo menos um dos elementos do grupo deve responder à questão no e-learning com as coordenadas da ponta do robô.