

Déploiement de ROS2

Pour le déploiement d'environnements pédagogique et concours

On voit plusieurs approches :

- déploiement de machine native Ubuntu, par clonage de disque par exemple
- déploiement de VM VirtualBox
- déploiement de Docker container

Veut-on enseigner la robotique ou l'ingénierie logiciel sous Linux ? Selon moi, il faut différencier deux types de public :

- On veut enseigner la robotique Open Source, par exemple à des électroniciens
 - on n'aborde pas l'installation ROS, la création de package dans un workspace et sa compilation
 - on fournit une machine dont l'enseignant maîtrise l'état
 - tout est fait dans un IDE type VSCode.
 - les apprenants peuvent se concentrer sur la génération de codes et d'algorithmes pour résoudre un problème robotique
 - on reste en simulation ou on déploie sur un robot en fin de parcours
 - on bascule d'un conteneur à un autre depuis VSCode pour éviter les erreurs de config
- On veut enseigner l'ingénierie logiciel appliquée à la robotique avec ROS
 - initiation à Ubuntu et à la ligne de commande
 - on déroule tout le tutoriel ROS en encourageant à faire un dual-boot (ou éventuellement un WSL) sur une machine personnelle
 - on insiste sur les réflexes de compilation dans le terminal
 - on laisse naviguer entre plusieurs workspaces sans Docker

Besoin pour la robotique industrielle

Par ordre croissant de complexité :

- Visualisation et planification trajectoire avec MoveIt2 via plugin motion_planning dans RViz2

https://moveit.picknik.ai/main/doc/tutorials/quickstart_in_rviz/quickstart_in_rviz_tutorial.ht

ml

- cellule SO-ARM100 ou Panda (moveit2_tutorial officiel) ou cellule UR5e
- Moveit2 Python API
https://moveit.picknik.ai/main/doc/examples/motion_planning_python_api/motion_planning_python_api_tutorial.html
- Connexion à docker container URSim <https://innovation.iha.unistra.fr/books/robotique-open-source/page/universal-robot-ros2-driver#bkmrk-installation-du-simu>
- Contrôle vrai SO-ARM100
- Contrôle vrai UR5e
- Contrôle local de trajectoire avec Moveit Servo : téléopération via manette ou asservissement visuel (TOF sensor)
https://moveit.picknik.ai/main/doc/how_to_guides/controller_teleoperation/controller_teleoperation.html
- Gazebo et simulation traitement de l'image 2D/3D
- Pour aller plus loin : [Cartesian Path Planning](https://moveit.picknik.ai/main/doc/tutorials/pick_and_place_with_moveit_task_constructor/pick_and_place_with_moveit_task_constructor.html), [Hybrid Planning](https://moveit.picknik.ai/main/doc/tutorials/pick_and_place_with_moveit_task_constructor/pick_and_place_with_moveit_task_constructor.html), Moveit Task Constructor (https://moveit.picknik.ai/main/doc/tutorials/pick_and_place_with_moveit_task_constructor/pick_and_place_with_moveit_task_constructor.html , UR10e_welding_demo)

Besoin pour la robotique mobile

Par ordre croissant de complexité :

- Tutoriels Nav2 avec paquets tb3 https://docs.nav2.org/getting_started/index.html
- Tutoriels ROBOTIS Turtlebot3 en simulation https://github.com/ROBOTIS-GIT/turtlebot3_simulations
 - Cartographie et Navigation autonome avec GZ Sim
 - Suivi de ligne avec OpenCV <https://innovation.iha.unistra.fr/books/robotique-open-source/page/suivi-de-ligne-ros2-humble>
- Tutoriels ROBOTIS Turtlebot3 avec le vrai robot
<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>
- Architecture de flotte :
 - temps réel sur le Raspberry : drivers actionneurs et capteurs, planification locale de trajectoire ?
 - calcul lourds sur PC déporté : cartographie, SLAM, planification globale de trajectoire
- Déploiement contrôleur sur le Raspberry du TurtleBot3
- Setup communication via access point ou hotspot wifi
- Environnement sur le PC de développement capable de communiquer via le réseau
 - PC portable pour le concours de robotique
- Contrôle vrai robot
- Cartographie vrai robot

- Calibration vraie caméra
- Pour aller plus loin : Traitement de l'image vrai robot, suivi de ligne, Behavior Trees Demo

Pour le déploiement de robots dans l'industrie

Dev Container

Pour un développement local

- Installer Visual Studio Code
- Installer Docker avec le support de l'accélération graphique
- Installer l'extension `Dev Container`

Pour un développement sur un serveur/PC distant

- Installer le pack d'extensions `Remote Development`

Lancer le conteneur de développement :

- Optionnel : se connecter à la machine de développement distante depuis Visual Studio Code
 - Ouvrir la Command Palette `Ctrl+Shift+P`
 - Lancer `Remote-SSH: Connect to host...`
 - Configurer la connexion ssh par mot-de-passe ou clé si ce n'a pas encore été fait
- Cloner le dépôt <https://github.com/rplayground/sandbox>
- Lancer `Dev Container: Reopen in Container`

<https://articulatedrobotics.xyz/tutorials/docker/dev-containers/>

Nav2, container, PWA <https://discourse.rRemote-SSH: Connect to host...os.org/t/mini-workshop-developing-and-teaching-ros-from-a-web-browser-using-dev-containers-and-pwas/31533>

<https://discourse.ros.org/t/repeatable-reproducible-and-now-accessible-ros-development-via-dev-containers/31398> https://github.com/ros-navigation/docs.nav2.org/blob/master/development_guides/devcontainer_docs/devcontainer_guide.md

Docker Containers + VSCode

Quelques conseils de Ragesh (<https://robotair.io/>) que j'ai rencontré au Fraunhofer :

https://github.com/athackst/workstation_setup

<https://docs.ros.org/en/iron/How-To-Guides/Setup-ROS-2-with-VSCode-and-Docker-Container.html>

<https://www.allisonthackston.com/articles/vscode-docker-ros2.html>

MoveIt2, docker <https://fr.slideshare.net/secret/vdPbBCNB3LamRy>

https://moveit.picknik.ai/main/doc/how_to_guides/how_to_setup_docker_containers_in_ubuntu.html

Ansible

<https://www.linkedin.com/pulse/deploying-ros2-packages-using-ansible-ragesh-ramachandran/>

<https://blog.robotair.io/best-way-to-ship-your-ros-app-a53927186c35>

https://github.com/swarmBots-ipa/ansible_automation

<https://github.com/ipa-rar/ros2-playbooks>

<https://github.com/Laura7089/ros2-ansible>

https://github.com/jeremyfix/ros2_ansible_turtlebot

<https://github.com/rarraais/ansible-role-ros2>

Multipass

https://artivis.github.io/post/2023/multipass_ros_blueprint/

YOCTO

<https://www.yoctoproject.org/>

JupyterLab

<https://blog.jupyter.org/jupyterlab-ros-3dc9dab7f421>

Scripts bash et Cluster SSH

https://github.com/runtimerobotics/ros2_online_install

Scripts de Loïc Cuvillon : <https://seafile.unistra.fr/d/50662484c5f641709cd7/>

Scripts Gauthier Hentz : <https://innovation.iha.unistra.fr/books/robotique-open-source/page/installation-pc-ros2#bkmrk-installation-de-jazz>

Discussions

<https://discourse.ros.org/t/easy-way-to-distribute-an-instance-of-ubuntu-w-ros-to-students/31824/4>

Singularity, Robolaunch, ROSblox <https://discourse.ros.org/t/what-environments-do-you-use-for-training-and-courses/26473/6>

Python <https://discourse.ros.org/t/teaching-with-ros-or-zero-ros-at-university/32124>

Sources

Revision #12

Created 15 April 2025 08:07:11 by admin_idf

Updated 15 July 2025 13:24:31 by admin_idf