

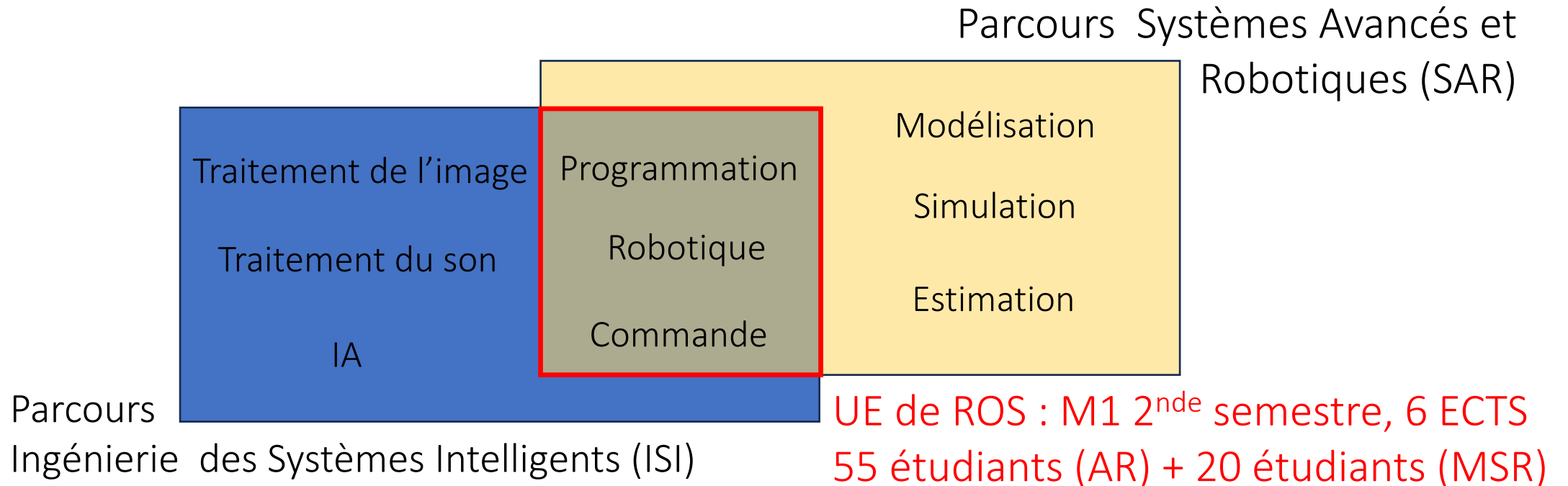
Enseigner ROS en utilisant la méthodologie ABCD

Sylvain ARGENTIERI, Ludovic SAINT BAUZEL, Fabien VÉRITÉ



Un peu de contexte

- **Objectif** : augmenter la part d'enseignements expérimentaux en robotique en Master AR
 - Création en 2020 d'une UE "ROS and experimental robotics"



Méthodologie ABCD : Alignement pédagogique

Acquis d'apprentissage visés

*“ Que doit savoir ou savoir faire l'étudiant
à la fin du cours ?
Quelle compétence doit-il avoir ? ”*

Cohérence

Activités pédagogiques

*“ Quelles activités je vais mettre en
oeuvre pour aider l'étudiant à atteindre
les objectifs définis, les acquis
d'apprentissage ? ”*

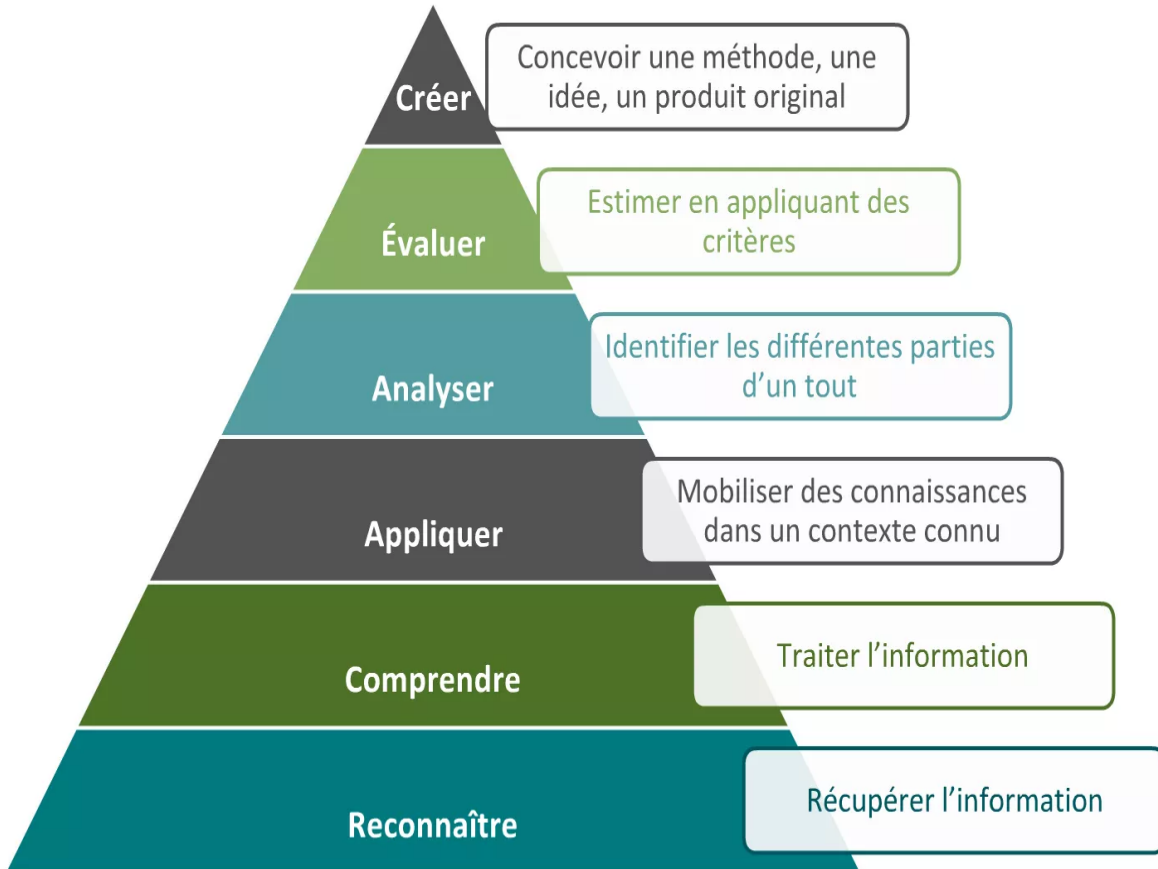
Stratégies d'Evaluation

*“ Comment est-ce que je vais évaluer
l'acquisition des apprentissage visés ? Par
le biais de quelle activité ? ”*



Méthodologie ABCD : Taxonomie de Bloom

❖ Taxonomie de Bloom (1956) – révisée par Krathwolh (2001) offre une taxonomie en 6 niveaux pour le domaine cognitif



Synthèse : concevoir, rédiger, planifier, réaliser, faire un exposé, produire, mettre au point...

Evaluation : justifier, défendre, juger, argumenter, critiquer, évaluer...

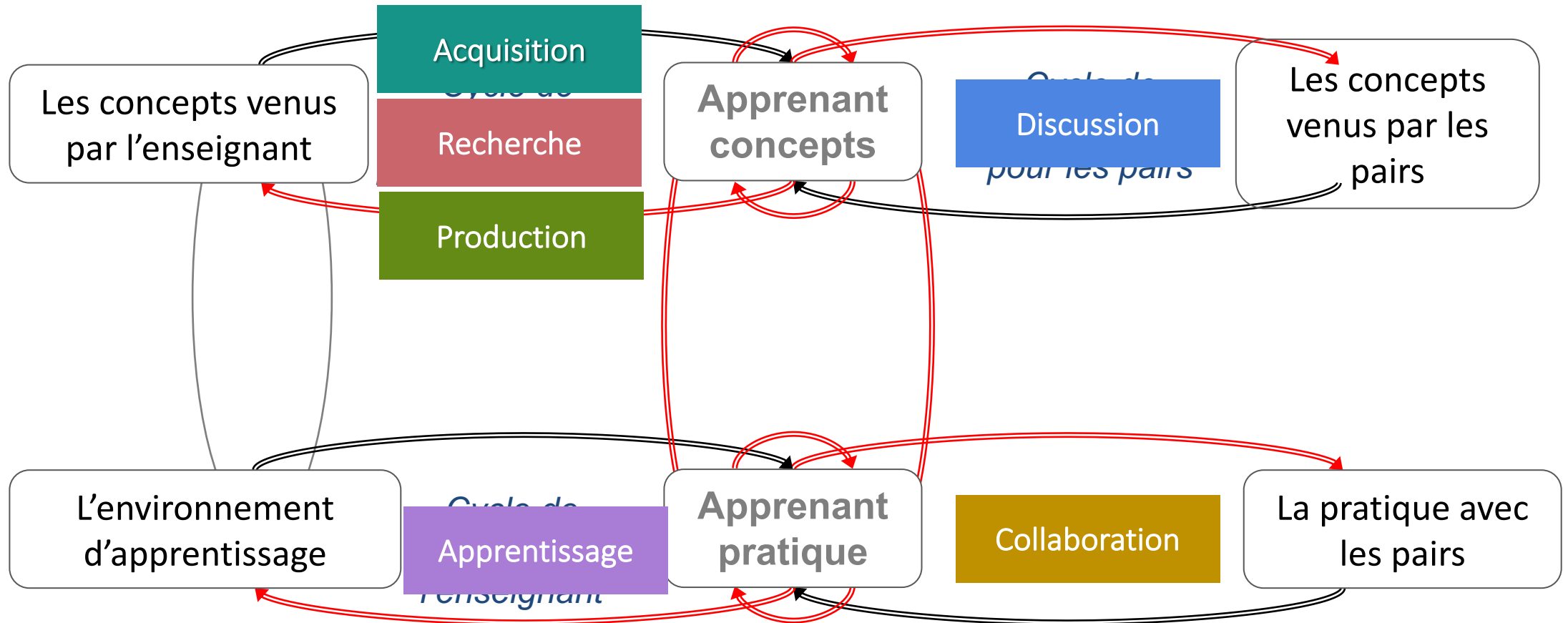
Analyse : analyser, distinguer, comparer, faire un choix...

Application : utiliser, résoudre, appliquer, construire, démontrer, calculer, dériver

Compréhension : décrire, dire avec ses propres mots, résumer, expliquer, interpréter

Connaissance : restituer une définition, identifier, nommer, énumérer

Méthodologie ABCD



Comment avons nous créé cette UE ?

- Dès le depart, volonté d'aligner pédagogiquement l'enseignement, en s'appuyant sur une nouvelle méthode pédagogique : la méthode ABCD.

Alignement pédagogique

Formulation claire des objectifs pédagogiques :

- Concis et précis
- Quantifiable (par l'étudiant et son évaluation)
- Focalisé sur les compétences

Evaluation

- Elle fait partie du processus d'apprentissage
- Seulement pour les objectifs d'apprentissage *e.g. ne pas tester la mise en pratique d'une compétence non enseignée*

Activity Based Curriculum Design

- Apprentissage centré sur l'étudiant** : les activités sont conçues sur la base des compétences utiles pour réaliser les objectifs d'apprentissage
- L'évaluation** est intégrée au processus d'apprentissage
- Apprentissage actif** : les cours sont pensés pour pousser à la participation active des étudiant et l'acquisition de compétences transverses.

Acquisition

Recherche

Apprentissage

Discussion

Production

Collaboration

Présentation des objectifs de l'UE

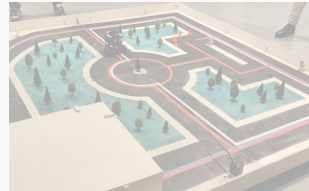
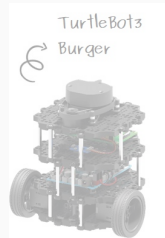
Un objectif principal*

Rendre les étudiants compétents sous ROS et capables d'y développer des éléments

**(basé sur une demande forte en labo & industrie)* :::: ROS

Une problématique réaliste présentée

Contexte du véhicule autonome : capacité à programmer un robot mobile dans un environnement encombré.



Objectifs secondaires

- Développer une application robotique sur la base d'un ensemble de spécifications
- Être confronté à la différence entre simulation et données expérimentales réelles
- Appliquer un variété des compétences théoriques en lien avec les enseignements précédents :
 - Python,
 - Traitement du signal/image,
 - commande, contrôle.

3 phases

Phase 1 : concepts de base ROS

- Définition des concepts de base, programmation événementielle, and communications client/serveur
- Programmation d'un nœud de téléopération
- Application au contrôle d'un robot simulé (turtlesim, Turtlebot 3 burger) et d'un robot réel (Turtlebot 3 burger)

Phase 2: simulation d'un robot réaliste à 2 roues

- introduction à la description géométrique des robots (URDF)
- présentation de gazebo pour rassembler des données sensorimotrices (laser, camera)
- exploitation des données laser simulées pour réaliser un comportement d'arrêt d'urgence

Phase 3 : implementation de capacités de navigation

- Projet visant à mobiliser tous les développements précédents de façon à permettre à un Turtlebot 3 burger de naviguer en autonomie dans une arène dédiée (lignes à suivre, obstacles, couloir, etc.)

3 phases

Phase 1 : concepts de base ROS

Phase 2: simulation d'un robot réaliste à 2 roues

Phase 3 : implementation de capacités de navigation



Phase 1 : les concepts de base de ROS1

Cours 1 : introduction à l'UE (1h)

Cours 2 : Introduction à ROS (2h)

Cours 3 : Introduction à la programmation événementielle (2h)

Cours 4 : session inversée Q&R (1-2h)

TP1
(4h)

TP2
(4h)

TP3
(4h)

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Tutoriel 1 ROS, suivi via un guide + forum en ligne

Evaluation :
QCM 1

Examen
pratique (2h)



Phase 1 : les concepts de base de ROS1

Cours 1 : introduction à l'UE (1h)

Cours 2 : Introduction à ROS (2h)

Cours 3 : Introduction à la programmation événementielle (2h)

Cours 4 : session inversée Q&R (1-2h)

TP1 (4h)

TP2 (4h)

TP3 (4h)

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

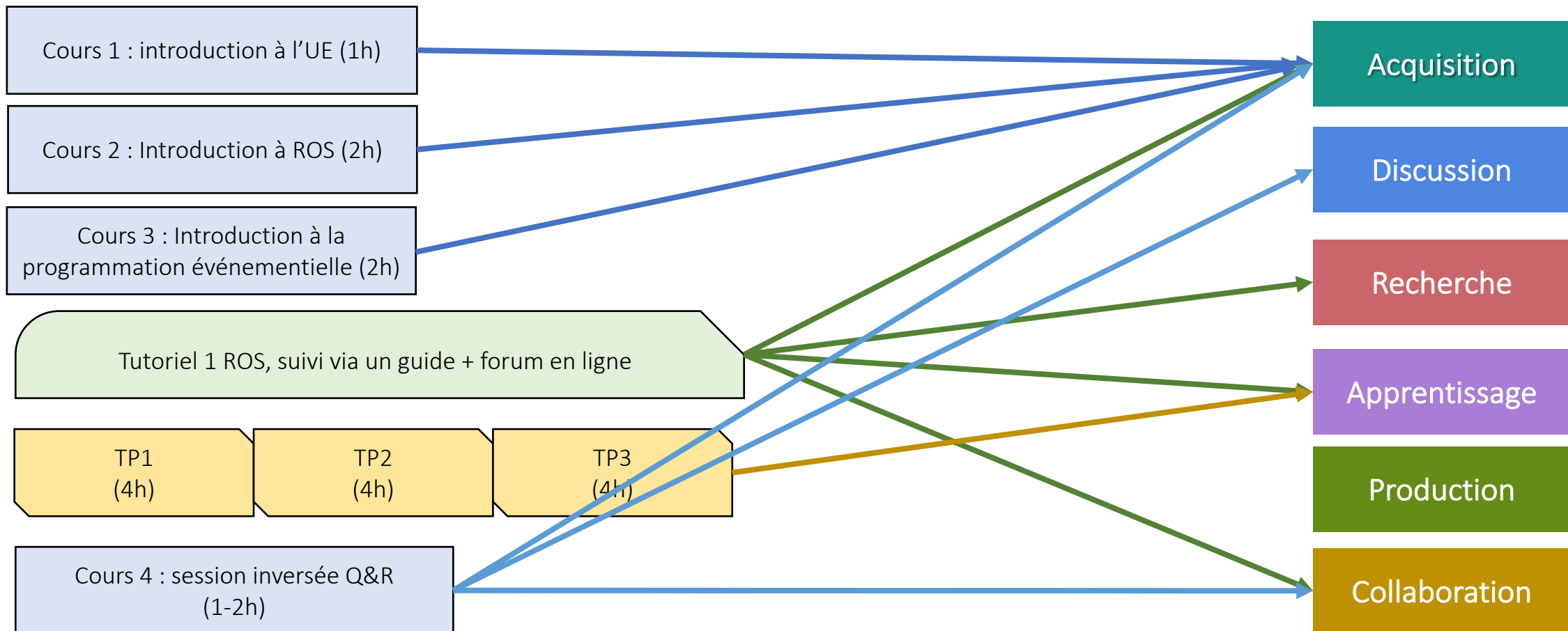
Tutoriel 1 ROS, suivi via un guide + forum en ligne

Evaluation :
QCM 1

Examen
pratique (2h)



Phase 1 : les concepts de base de ROS1



Phase 1 : tutoriel

- <https://moodle-sciences-22.sorbonne-universite.fr/mod/page/view.php?id=132167>

Work to do and tutorials to follow.

As explained in the first introductory course, this course is based on new pedagogical methods that aim to maximize your time in class to do experiments (ROS and experimental robotics, that was in the title!). For that, and after two introductory courses, you will have to follow tutorials at home.

Attention: this course is entirely based on this autonomous work.

If you don't do this work or follow the tutorials without trying to understand, you will be completely lost during the practical sessions. A good method of work would be to create a summary sheet of everything you learn during these tutorials so that you can use it during the practical sessions.

All the tutorials on the official website and on the beginner, level must be done before the first practical session (<http://wiki.ros.org/ROS/Tutorials>), only the first tutorial on the installation is not to be followed since you have a virtual machine with a complete installation.

Similarly, all the videos mentioned are to be watched before the first session.

We can't insist enough but it's not by simply reading and passively watching the videos that you will be ready for the sessions. It is necessary to redo all the manipulations (and to understand them). Don't hesitate to re-watch the introductory documents/videos once you've made some progress. Some of the concepts will be clearer after you have worked on them a bit and you will understand more and new things by doing so.

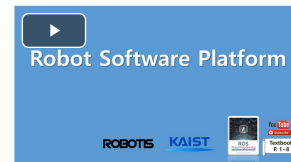
To help you during this work we propose you hereafter an organization on how to follow the tutorials of the official ROS website in parallel of the explanatory videos created by Robotis (manufacturer of the turtlebot) which also proposes an initiation to ROS. The path we propose among these tutorials is supposed to guide you to acquire the list of skills expected at the end of part 1 and which can be found on moodle (see document ..)

Recommended planning :

DAY 1	How to set-up your environment
DAY 2	Chapter 1, 2 and 3
DAY 3	Chapter 4, 5, 6
DAY 4	
DAY 5	Chapter 7: nodes, topics, publisher, subscriber
DAY 6	
DAY 7	Chapter 7: Services
DAY 8	Chapter 7: Parameters, launchfile
DAY 9	Conclusion: last tutorials + study for the exam (review of important points)
DAY 10	

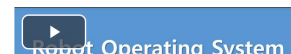
Chapter 1 Robot Software Platform

General introduction about the need for a unifying OS that can interface between all software and hardware developments



Chapter 2 Robot operating system

General introduction to ROS and the tools it provides to simplify/optimize the development of robotic applications.



Phase 1 : QCM

Quelques exemples de questions du QCM posé après les tutoriels

Question 1
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

A subscriber node

Veillez choisir une réponse.

- a. is a ROS node that can only receive data from the ROS Master.
- b. is a ROS node sending data in the form of messages to other ROS nodes.
- c. directly requests connection to the publisher node and receives messages without any interaction with the ROS Master.
- d. directly requests connection to the publisher node and receives messages from the connected publisher node after getting information on the publisher node from the ROS Master.

Question 9
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

The ROS master

Veillez choisir une réponse.

- a. monitors every currently running nodes and automatically detects when a problem, a bug, appears when running and tries to solve it (by, for instant, relaunching the faulty node).
- b. must be launched manually when launching roscore.
- c. provides naming and registration services to the rest of the nodes in the ROS system.
- d. is in charge of collecting every messages sent from a node A to a node B, and to distribute them to node B.

Page précédente

Page suivante

Question 5
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

The rosparam command-line tool:

Veillez choisir une réponse.

- a. allows you to launch or stop the ROS Parameter Server.
- b. can not be used to get a parameter value, but only to set its value.
- c. can not be used to set a parameter value, but only to get its value.
- d. allows you to store and manipulate data on the ROS Parameter Server.

Question 10
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

roscore will startup :

Veillez choisir une réponse.

- a. a ROS Parameter server, a rosout recording node and a ROS service server.
- b. a ROS Master node and a ROS topic server.
- c. a rosout logging node only, all other requirements being automatically launched when executing a ROS node.
- d. a ROS Master node, a ROS Parameter Server and a rosout logging node.

Page précédente

Terminer le test...

Phase 1 : les concepts de base de ROS1

Contenu des séances de travaux pratiques :

- Partie A : nœud de téléopération du turtlesim

→ exploration des propriétés, création d'un nœud, publisher

- Partie B : utilisation des services

→ subscriber, appel de services

- Partie C : launchfile et paramètres

→ lancement de plusieurs nœuds, paramètres, téléop du Turtlebot 3



Fin de la Phase 1 : Examen de TP

Attendus explicités :

« All documents are authorized, including your own code written during the first practical sessions. In practice, you must be able to correctly pass the exam if you are able to:

- investigate an existing architecture with the basic ros commands and interpret their outputs;
- write a publisher and subscriber node;
- write and use launchfiles and parameters;
- use an existing service. »

Sorbonne Université – Département de Sciences De l'Ingénieur ROS and experimental robotics

March 2022, 1h.

You are in charge of the development of a mobile cleaning-robot (Gaussian Robotics ECOBOT Sprayer 50). This robot is sold to automatically clean public places such as train stations or airports. The robot is equipped with a mobile base composed of two motorized wheels. In order to allow its displacements in a loaded environment, the robot is equipped with a Lidar, a 3d camera and ultrasonic sensors (abbreviated as US in the following). All these sensors allow the robot to plan its movements in the area to be cleaned. Above all, these sensors allow a safety redundancy in the detection of obstacles: it is indeed essential that the robot adapts its behavior to the proximity of an obstacle (which can be here a human).

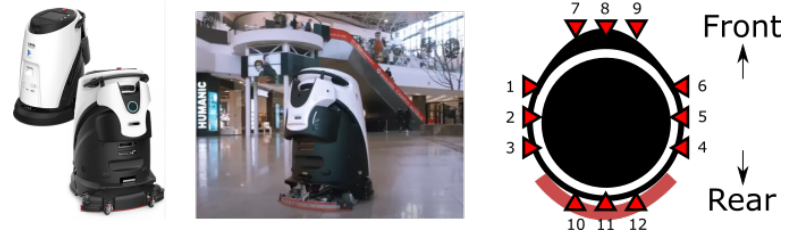


Figure 1: Views of the Gaussian Robotics ECOBOT Sprayer 50 robot. On the right, the layout of its ultrasonic sensors (in red). Each sensor is identified with an ID number inside the sensor array.

As you can see on Figure 1, the robot is equipped with 12 ultrasonic sensors (3 on each side) to detect an obstacle in each direction. Each sensor of the ultrasound (US) array aims at estimating the distance to the obstacles around the robot through a time-of-flight computation: a short high frequency burst is sent by the US emitter, which comes back to an US receiver when an obstacle is present. Each sensor actually measures the time taken by the wave to go back and forth, which is a function of the speed of sound ($c \approx 340 \text{ m.s}^{-1}$) and the distance d to the obstacle. The US sensor documentation indicates:

Each US sensor is not able to measure any distance greater than 5m, or smaller than 5cm.

In practice, the ROS interface to the sensors array relies on a node `ecobot_ultrasound` which communicates directly with the array of US sensors, and publishes (among other things) an array containing the 12 time-of-flight values (in s) for each US sensor on the topic `/usarray`. In this topic, the data are arranged in the sensor IDs order, see again Figure 1.

Sorbonne Université – Département de Sciences De l'Ingénieur ROS and experimental robotics

February 2023, 1h30.

You work in an underwater robotics company and you have to code a measurement solution. The drone in question is a monitoring drone allowing to monitor and check underwater installations (pipeline ...). The drone is equipped with 4 turbines, a battery, lighting, a camera and many sensors including a pressure sensor that allows to estimate its depth.



Figure 1: CHASING M2 PRO (<https://www.chasing.com>)

The drone is controlled from a remote which is equipped with joysticks, a pressure sensor and leds indicating the current status of different elements on the drone, like the battery charging level, or warnings about e.g. too far distance between the drone and the pilot.

You are in charge of the depth estimation of the drone, which must be computed from the pressure measured onboard and on the remote. You will have to use this measurements to create an alert to warn the pilot that it is necessary to make the drone go up before losing communication with the drone or damage it due to high pressure.

The depth (in meters) can be estimated with the following relation (considering that the water is incompressible).

$$Depth = \frac{P_{robot} - P_{remote}}{\rho * g} \quad (1)$$

with ρ the density of salt water = 1030 kg/m^3 , $g = 9.80665 \text{ N/kg}$, P_{robot} (Pa) the pressure measured on the robot and P_{remote} (Pa) the pressure measured on the remote at sea level.

Your colleagues have already developed a ROS node `submarinedrone` running on the drone, and communicating directly with the hardware on the system, and publishing the sensors values on dedicated topics. For now, the engineers are able to publish (i) the pressure measured on the robot by the corresponding sensor in the `/pressure_robot` topic (in Pa), and (ii) the pressure measured on the remote by the corresponding sensor in the `/pressure_remote` topic (in Pa).

3 phases

Phase 1 : concepts de base ROS

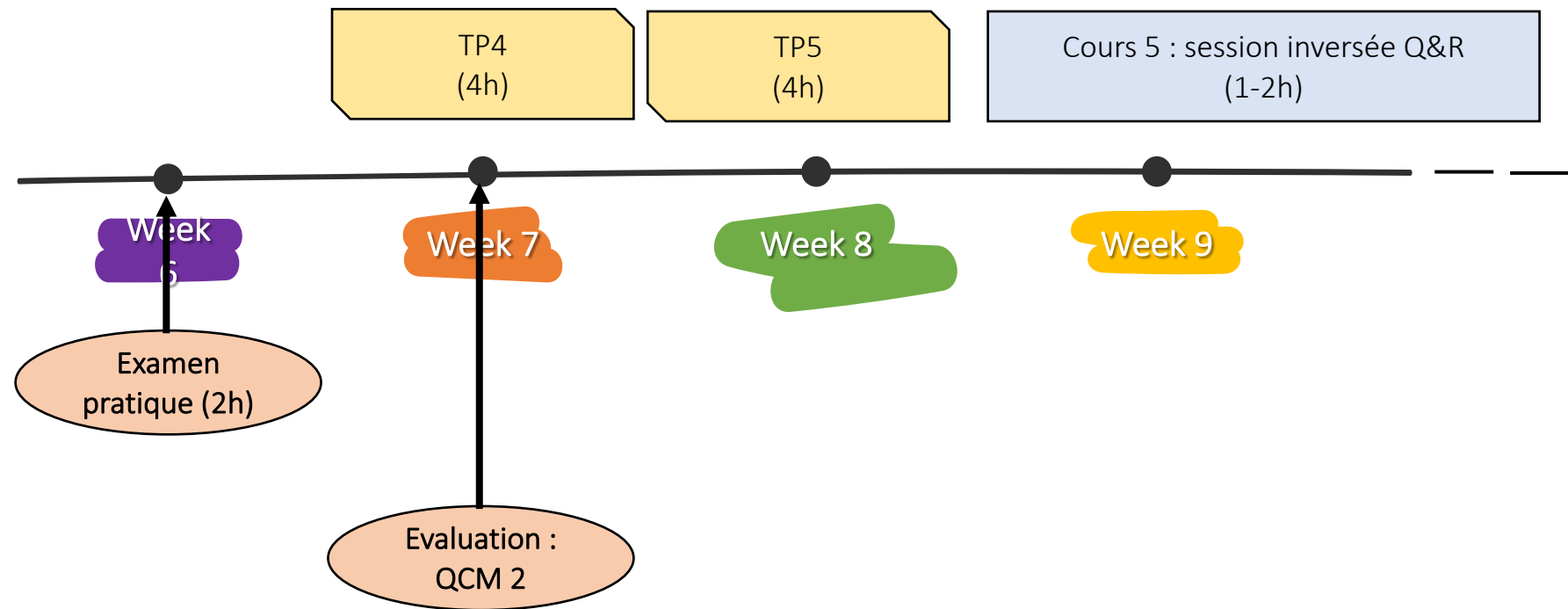
Phase 2: simulation d'un robot réaliste à 2 roues

Phase 3 : implementation de capacités de navigation



Phase 2 : URDF, visualisation, simulation

Tutoriel 2 URDF, suivi via un guide + forum en ligne



Phase 2 : URDF, visualisation, simulation

- Tutoriel + QCM à nouveau

Tutorials part 2

Tutorials Part 2 : URDF, visualisation (RVIZ) and simulation (GAZEBO)

After phase 1 of this teaching unit, you should now master the basic concepts of ROS.

The goal of this second phase is to use these basic skills to program a sensorimotor task for a simulated, then real, robot. Two tasks will be programmed: the first one corresponds to an essential safety mechanism, where the robot will have to be able to stop at any time if an obstacle appears in its path. The second task will be a line following task allowing the robot to automatically follow a pre-defined path.

In order to perform this kind of elementary tasks, it will be necessary:

- to understand and manipulate the universal declaration format that the ROS environment proposes to use, i.e. the URDF format (Unified Robot Descriptor Format) as well as the functioning of XACRO which allows to simplify the writing of this format ;
- to understand and exploit the physical simulator we will be using in the following, i.e. GAZEBO
- to be comfortable with the visualization tool (RVIZ) and ROSBAG in order to prototype and debug more easily your control algorithms

URDF

The URDF is a ROS specific standardized XML format for representing the model of a robot using basic geometries and linkages between them. It allows to have a code-independent, human-readable way to describe robots. It takes into account the dynamics of the different parts of the robot and their interactions (mechanical linkages, elasticity, viscosity, etc.). An external mesh-file could also be associated for display purposes.

As in phase 1, it is **mandatory** for you to carefully follow the following ROS tutorial as it will practically explain how URDF and XACRO work together.

This thus constitutes mandatory homework, and you have about 2 weeks to complete the tutorials.

In the meantime, a discussion forum is made available on Moodle and all your teachers will be there to respond to the questions you might have. In the end of the tutorial period, your knowledge and understanding of the URDF format and Gazebo simulation will be assessed **through multiple choice questions on Moodle**.

The tutorial is available at <http://wiki.ros.org/lurdf/Tutorials> and you have to work on the tutorials from the section "Learning URDF Step by Step" :

- [Building a Visual Robot Model with URDF from Scratch](#)
- [Building a Movable Robot Model with URDF](#)
- [Adding Physical and Collision Properties to a URDF Model](#)
- [Using Xacro to Clean Up a URDF File](#)

Question 3
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

XACRO, by using macros that expand to larger XML expressions, allows to :

Veillez choisir une réponse.

- a. construct more complex robotic system than with URDF.
- b. construct shorter and more readable XML files.
- c. reduce the number of files for a given robotic system.
- d. add sensors and actuators to the robot.

Question 5
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

URDF is a :

Veillez choisir une réponse.

- a. XML format for representing a robot model.
- b. 3d file format used to represent robot
- c. a Python file use to describe a robot
- d. an identification code to identify a robot

Question 10
Pas encore répondu
Noté sur 1,00
Marquer la question
Modifier la question

```
131 <link name="left_back_wheel">
132 <visual>
133 <origin rpy="1.57075 0 0" xyz="0 0 0"/>
134 <geometry>
135 <cylinder length="0.1" radius="0.035"/>
136 </geometry>
137 <material name="black"/>
138 </visual>
139 </link>
```

Choose the true assertion about the previous code:

Veillez choisir une réponse.

- a. The defined link is a cylinder with a diameter=0.035m.
- b. The origin of the link is shifted to the back and left.
- c. The origin of the link is shifted on the y axis.
- d. The link default orientation is shifted of $\pi/2$ around x axis.

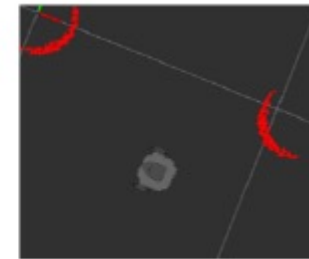
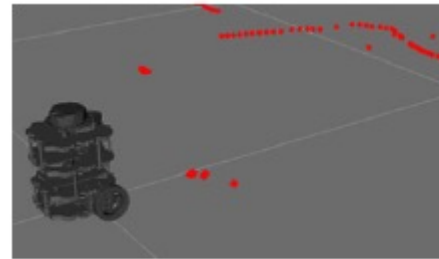
Page suivante

Phase 2 : URDF, visualisation, simulation

Contenu des séances de travaux pratiques :

- Partie A : visualisation

 - description URDF du robot, RVIZ



- Partie B : simulation

 - Gazebo, simulation des données sensorielles (LDS, camera, IMU) et exploitation dans un nœud élémentaire

- Partie C : vers le robot réel

 - affichage des données dans RVIZ, rosbag



3 phases

Phase 1 : concepts de base ROS

Phase 2: simulation d'un robot réaliste à 2 roues

Phase 3 : implementation de capacités de navigation



Phase 3 : projet !

Acquisition

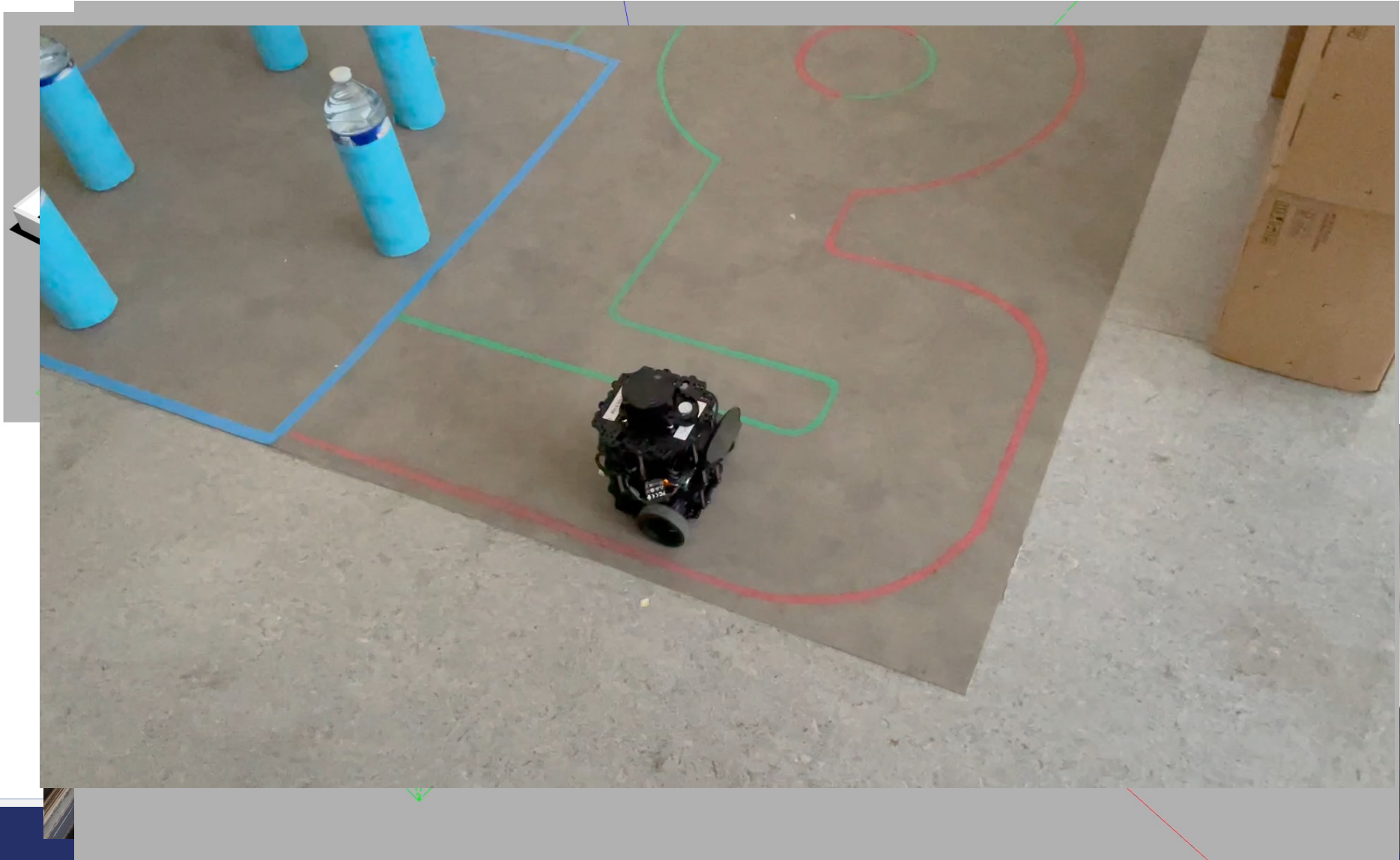
Discussion

Investigation /
research

Practice /
training

Production

Collaboration



Retour d'expérience

Retours des étudiants

- ❑ **Globalement positifs : ils apprécient particulièrement le côté expérimental et la phase de projet**
- ❑ **Nous avons compris avec les premiers retours qu'il fallait**
 - dédier une séance à l'explication de la méthode (travail individuel en autonomie...)
 - guider le tutoriel en créant une page dédiée
 - Donner des jalons (vous devriez maintenant en être là ...) ,
 - *Animer via le forum Moodle (questions/réponses, activités ...)*
- ❑ **Beaucoup de prérequis** (connaissances de base en réseau , utilisation de linux, programmation événementielle...)
- ❑ **Bon retour sur l'attractivité de la formation pour les entreprises**
- ❑ **Les étudiants utilisent spontanément ROS lors de leurs projets de M2**

Retour d'expérience

Nos retours

Abandon des grilles d'auto-évaluation au fur et à mesure car trop lourdes à utiliser pour la notation en « live », passage à un « examen » de travaux pratiques

ROS & experimental robotics: part 1

	Unacceptable : 0 point	Insufficient : 2.5 points	Objective reached : 7.5 points	Beyond : 10 points	Points
The student knows ROS architecture and its root concepts.	The student is not able to define basic ROS concepts like nodes, topics, messages, etc.	The student has correctly understood the concepts of nodes, topics, etc., but still does not master more advance concepts such as the role of the ROS master, parameters, services, messages, etc.	All the root concepts of ROS are correctly understood.	The student is able to generalize these concepts to more involved ROS architectures and masters the use of advanced concepts.	
The student is able to investigate a ROS architecture, either by CLI or graphical tools.	Classical ROS CLI tools (roslaunch, rostopic, etc.) or graphical tools (rqt) are not correctly used.	CLI or graphical tools are sometimes correctly used, the student is not able to describe them or to use them without external help (webpage, internet search, etc.)	ROS CLI andn graphical tools are correctly exploited and questions on their use are correctly answered.	The ROS CLI and graphical tools are known by heart, correctly exploited, even with advance options.	
The student is able to use the rospy client library to specify and realize a ROS node	The student is not able to develop autonomously a running node, or to explain clearly the purpose of classical intructions for the creation of a talker or subscriber node. Launch files are not understood at all.	The student is able to develop a ROS node, but still do not entirely master classical intructions for using services, parameters and the underlying concept of callback functions. Launch files are not well understood.	The student is able to code a node with communications to/from topics, services, parameters and to write a launch file to execute a ROS node with specific parameters.	The student shows great capabilities to generalize the programming concepts to other applications and context without hesitations.	
	Unacceptable : 0 point	Insufficient : 5 points	Objective reached : 15 points	Beyond : 20 points	Points
Achievements	Nothing works!	Only the teleoperation node works correctly	Both teleoperation and color nodes work as specified. The student has began to work on simulated turtlebot 3.	Teleoperation and color nodes are working with additional capabilities, and the teleoperation of the simulated turtlebot 3 is working.	
					-20

Noms des étudiants :

TOTAL (/50)

ROS & experimental robotics: part 1					
	Unacceptable : 0 point	Insufficient : 2.5 points	Objective reached : 7.5 points	Beyond : 10 points	Points
Challenge 1 : line follower	The provided launch file does not work and/or does not use the correct requirements.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.
Challenge 2 : obstacle avoidance	The provided launch file does not work and/or does not use the correct requirements.	Task 1 is working successfully, and the robot is able to avoid a moving obstacle from a fixed direction.	Task 1 is working successfully, and the robot is able to avoid a moving obstacle from a fixed direction.	Task 1 is working successfully, and the robot is able to avoid a moving obstacle from a fixed direction.	Task 1 is working successfully, and the robot is able to avoid a moving obstacle from a fixed direction.
Challenge 3 : visual and LIDAR based navigation	The provided launch file does not work and/or does not use the correct requirements.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.	Task 1 is working successfully, and the robot is able to follow a line through the "line follower" experiment. The robot is able to follow a line through the "line follower" experiment.
Code	The code is not commented at all, and contains only hard-coded values.	The code is commented at all, and contains only hard-coded values.	The code is commented at all, and contains only hard-coded values.	The code is commented at all, and contains only hard-coded values.	The code is commented at all, and contains only hard-coded values.
Noms des étudiants :					TOTAL /50

ROS & experimental robotics: part 2					
	Unacceptable : 0 point	Insufficient : 2.5 points	Objective reached : 7.5 points	Beyond : 10 points	Points
The student knows ROS architecture and is able to investigate a ROS architecture, either by CLI or graphical tools.	The student is not able to define basic ROS concepts like nodes, topics, messages, etc.	The student has correctly understood the concepts of nodes, topics, etc., but still does not master more advance concepts such as the role of the ROS master, parameters, services, messages, etc.	All the root concepts of ROS are correctly understood.	The student is able to generalize these concepts to more involved ROS architectures and masters the use of advanced concepts.	
The student is able to use the rospy client library to specify and realize a ROS node	The student is not able to develop autonomously a running node, or to explain clearly the purpose of classical intructions for the creation of a talker or subscriber node. Launch files are not understood at all.	The student is able to develop a ROS node, but still do not entirely master classical intructions for using services, parameters and the underlying concept of callback functions. Launch files are not well understood.	The student is able to code a node with communications to/from topics, services, parameters and to write a launch file to execute a ROS node with specific parameters.	The student shows great capabilities to generalize the programming concepts to other applications and context without hesitations.	
Achievements	Nothing works!	Only the teleoperation node works correctly	Both teleoperation and color nodes work as specified. The student has began to work on simulated turtlebot 3.	Teleoperation and color nodes are working with additional capabilities, and the teleoperation of the simulated turtlebot 3 is working.	
Noms des étudiants :					TOTAL /50

Retour d'expérience

Nos retours

- ❑ **A permis une transition facile vers la formation à distance car l'hybride était déjà au cœur de l'UE**
- ❑ **Alignement pédagogique très intuitif pour les étudiants**
 - on retrouve malgré tout une grande variabilité dans les notes
- ❑ **Entretien chronophage de la plateforme**
 - Beaucoup de robots à entretenir pour pouvoir les mettre à disposition des étudiants + arène « student-proof »
 - Mise à disposition de machines virtuelles pour faciliter le debug (mais doit être compatibles avec les différentes machines des étudiants)

Merci pour votre attention

[https://github.com/sylarevan/ROS and Experimental Robotics](https://github.com/sylarevan/ROS_and_Experimental_Robotics)



**SORBONNE
UNIVERSITÉ**

