

CAPACITES : ANALYSER- EXPERIMENTER ET SIMULER : PRESENTATION ACTIVITES ELEVES
Problématique technique

Analyser les programmes de tests de fonctionnement fournis et en créer d'autres avec des fonctionnalités imposées

Compétence(s) issue(s) du programme officiel

Traduire un algorithme en un programme exécutable

Connaissance(s) associée(s)

Langage de programmation Python

Prérequis

Savoir faire

Connaitre les fondamentaux du langage Python et les différentes structures de contrôle. Notions d'asservissement avec correcteurs PID

Conditions de réalisation

Durée du TP

3*2 heures

Nombre d'élèves

2 binômes

Critères et modalités d'évaluation liés aux objectifs pédagogiques

*Formative : Évaluation en fonction :
De l'organisation du groupe de travail pour :*

- *Communiquer*
- *Mutualiser les résultats*

MISE EN ŒUVRE DE L'ACTIVITE
Environnement matériel et logiciel nécessaire

Robot AppBot et environnement de programmation Myviz

Documents à utiliser

Dossier technique

Modalités propres à ces activités

Les réponses sont portées sur des documents à rédiger

Activités abordées

- *Programmation appliquée au déplacement et contrôle d'un robot*

Problématique technique

Ecrire un programme de pilotage et de contrôle du déplacement d'un robot de surveillance

En présence du robot didactisé, du **dossier technique** et du logiciel de programmation :

 **Les élèves du poste doivent :**

Phase 1 du TP

- Exécuter et identifier les séquences du programme python donné pour piloter le robot
 - Analyser les courbes enregistrées pour visualiser les mouvements effectués
 - Mesurer les valeurs obtenues, quantifier les écarts et justifier les résultats
- Ecrire des programmes en python pour exécuter des cycles de déplacement du robot

Phase 2 du TP

- Etudier et programmer le robot en utilisant différentes structures de contrôle d'une boucle d'itération
- Ecrire et tester un programme permettant au robot d'effectuer un balayage avec la caméra d'un angle de 90° à différentes vitesses de balayage (de 40 à 90°). L'objectif est de définir à quelle vitesse on obtient une image vidéo stable et exploitable

Phase 3 du TP

- Optimiser les correcteurs PID pour obtenir une image stable lors des déplacements en translation et rotation du robot
- Programmer un plan d'inspection d'une pièce regroupant toutes les fonctions abordées

Il est attendu :

- Un document de synthèse commun à l'équipe, en réponse au problème posé et mettant en œuvre les techniques de communication
- Une fiche de formalisation des connaissances et des compétences abordées durant le TP

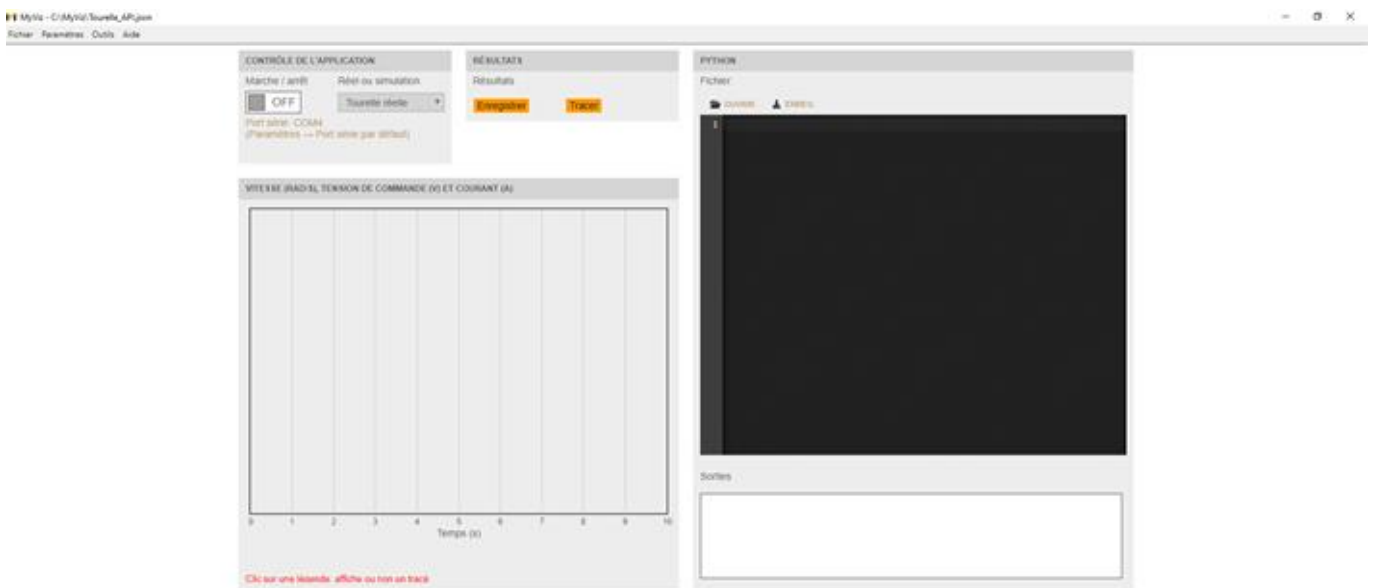
Phase 1 du TP

Les élèves du poste doivent :

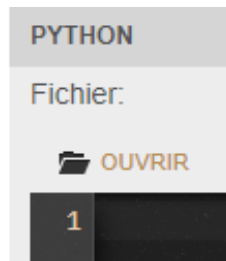
- Tester avec le logiciel Myviz, le programme de test de fonctionnement fourni.

Programme= **Test_Appbot_Api.py**

 **Lancer le logiciel Myviz**

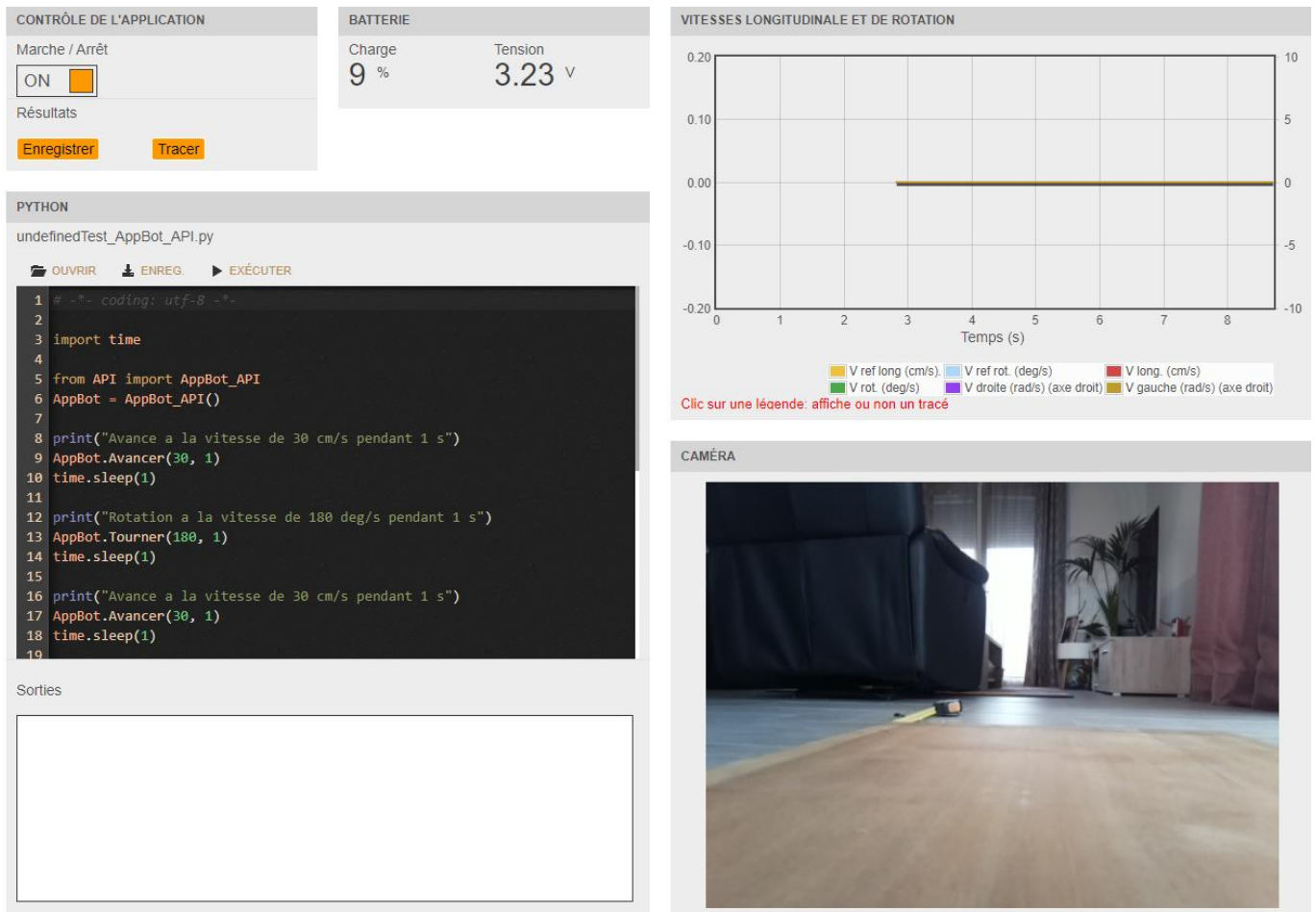


 **Cliquer sur ouvrir**



 **Choisir le programme **Test_Appbot_Api.py****

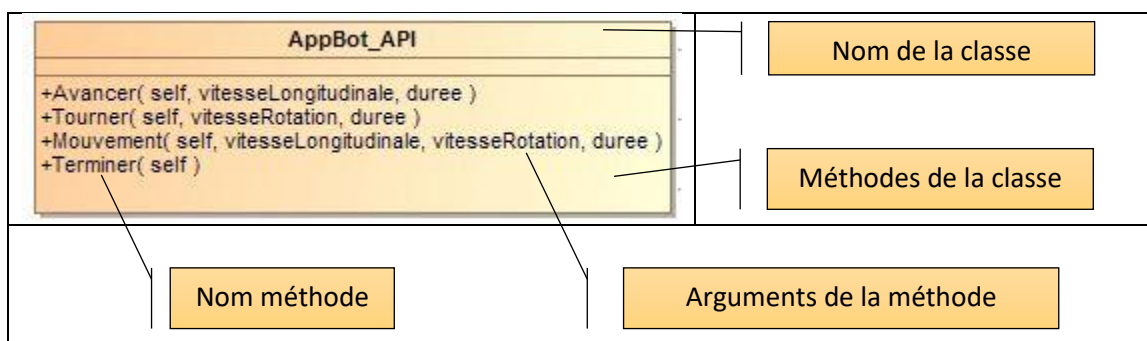
 **Cliquer sur OFF → ON**



 Cliquer sur EXCUTER

Q1-1 : Observer le déroulement des mouvements du robot et du programme et noter le type de mouvement effectué dans le tableau ci-dessous en vous référant au programme Test_Appbot_Api.py.

Rappel : modélisation Sysml et syntaxe employée pour décrire une classe :



Analyse du programme :

AppBot=AppBot_API () : Instanciation de l'objet AppBot par le constructeur AppBot_API ()

L'appel des méthodes se fait par la syntaxe **nom objet. Nom de la méthode**

Lors de l'appel d'une méthode de l'extérieur de la classe (c'est notre cas dans ce programme simple), l'argument self qui représente la référence de l'instance est ajouté automatiquement par Python, il peut être omis.

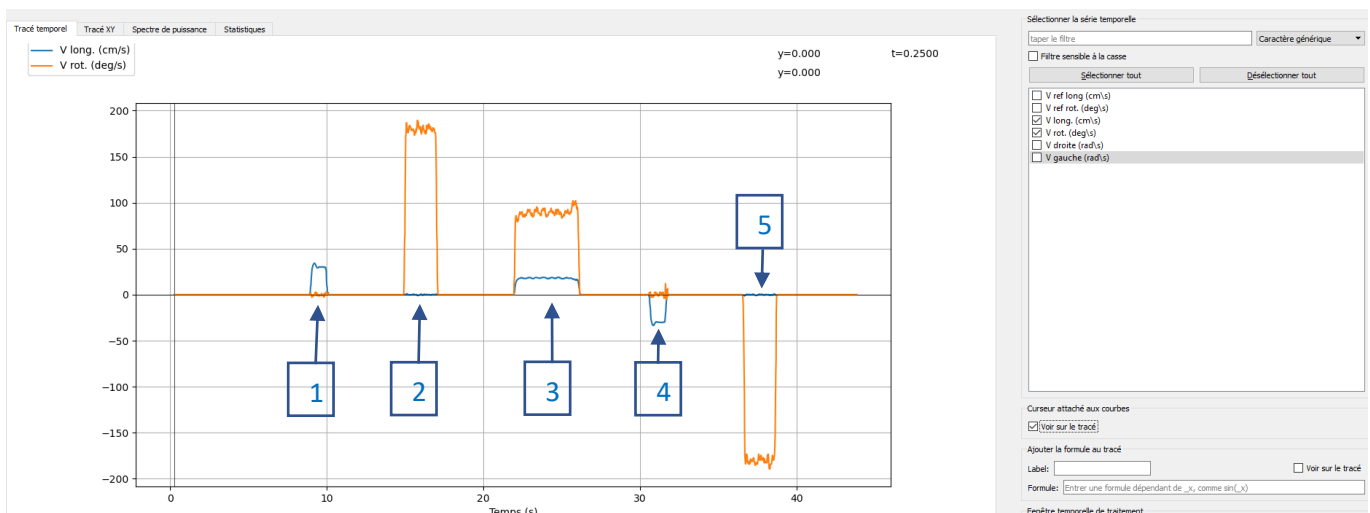
Séquences	Abréviation	Méthode et paramètres de la classe AppBot_API	Unité et valeur des paramètres
1			
2			
3			
4			
5			

Abréviation des déplacements :

- Rectiligne vers l'avant : AV
- Rectiligne vers l'arrière : AR
- Rotation vers la droite : RD
- Rotation vers la gauche : RG
- Mouvements combiné translation et rotation :MC

Analyser les courbes de réponses des consignes de mouvements


Cliquer sur le bouton TRACER pour lancer l'affichage des courbes obtenues





Q1-2 : Sélectionner successivement les bonnes courbes à visualiser et indiquer dans le tableau à quelle séquence correspondent les repères qui figurent sur l'image

Repères	Séquence	Caractéristiques mesurées : Vitesse (cm/s) et durée (s)	Distances parcourues en cm et degrés
1			
2			
3			
4			
5			

Positionner le robot au centre du cadre fourni


Cycle	Programmes	Courbes
<p>Aller/Retour</p> 	<pre>import time from API import</pre>	<p>Vitesses longitudinales</p> <p>Identifier sur les courbes les différents mouvements et indiquer la durée et les vitesses réelles de chacun d'eux.</p>
<p>Arguments</p> <p>Vitesse aller et retour = 20 cm/s</p> <p>Durée= 3s</p>		

<p> Aller/Rotation 180°/Retour/Rotation 180° </p> 	<p>import time from API import</p>	<p>Vitesses longitudinales</p>
<p> Paramètres </p> <p> Vitesse aller et retour = 20 cm/s Durée= 3s </p> <p> Vitesse rotation= 180 deg/s Durée= 1 </p>		<p>Vitesses de rotation</p> <p>Identifier sur les courbes les différents mouvements et indiquer la durée et les vitesses réelles de chacun d'eux.</p>

Cycle	Programmes	Courbes
<p>Utiliser la commande mouvement() pour ce cycle :</p> <p>Nota : la commande mouvement() réalise un mouvement combiné d'une translation et d'une rotation</p> <p>Aller/ Déplacment combiné sur 540° / Retour/Déplacement combiné sur 180°</p> 	<pre>import time from API import AppBot_API</pre>	<p>Vitesses de translation et de rotation</p> <p>Identifier sur les courbes les différents mouvements et indiquer la durée et les vitesses réelles de chacun d'eux.</p>
<p>Paramètres</p> <p><u>Aller et retour</u> Vitesse = 20 cm/s Durée = 2s</p> <p><u>Déplacement combiné 540°</u> Vitesse linéaire = 20 cm/s Vitesse angulaire = 100 deg/s Durée = 5,4 s</p> <p><u>Déplacement combiné 180°</u> Vitesse linéaire = 20 cm/s Vitesse angulaire = 180 deg/s Durée = 5,4 s</p>		

Phase 2 du TP

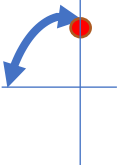
Q2-1 : Etudier et programmer plusieurs aller /retour du robot en utilisant différentes structures de contrôle d'une boucle

Cycle	Programme						
<p>Aller/Retour (3 cycles)</p> 							
<p>Consignes de programmation</p> <p>Vitesse= 3 paliers d’une durée de 2s pour chaque aller/retour</p> <table><tr><td>Itération 1</td><td>Itération 2</td><td>Itération 3</td></tr><tr><td>V=10 cm/s</td><td>V=20 cm/s</td><td>V=30 cm/s</td></tr></table> <p>Type de boucles à utiliser dans le programme</p> <p>for j in range (cycle1)</p> <p>while (cycle2)</p> <p>for j in (cycle3)</p>	Itération 1	Itération 2	Itération 3	V=10 cm/s	V=20 cm/s	V=30 cm/s	
Itération 1	Itération 2	Itération 3					
V=10 cm/s	V=20 cm/s	V=30 cm/s					

Courbes obtenues

Identifier les 3 cycles et sur chaque cycle les mouvements du robot.

Q2-2 : Ecrire et tester un programme de balayage d'un angle de 90° à différentes vitesses (de 40 à 90 deg/s) pour définir la vitesse qui permet d'avoir une image vidéo stable et exploitable lors du balayage

Cycle	Programmes	Courbes
<p>Effectuer 6 cycles</p> <p>Rotation gauche Rotation droite balayant un <u>angle</u> <u>constant de 90°</u></p> 		<p>Vitesses de rotation</p>
<p>Paramètres</p> <p>Vitesse rotation= de 40 à 90 deg/s</p>		<p>Constat :</p>

Phase 3 du TP

Q3-1 : Ecrire et tester un programme de déplacement rectiligne en aller/retour avec trois changements de valeurs des correcteurs PID.

Vitesse = 15cm/s
Durée = 4s
Kp = [0.02,0.15,0.25]
Ki = [10,4,5]
Kd = [0,0,0.005]

Q3-2 : Ecrire et tester un programme de déplacement en rotation en boucle avec trois changements de valeurs de correcteurs PID.

Vitesse = 15cm/s
Durée = 4s
Kp = [0.02,0.15,0.25]
Ki = [10,4,5]
Kd = [0,0,0.005]

Q3-3 : Ecrire et tester un programme de déplacement combinant toutes les fonctions abordées

FICHE DE FORMALISATION DES CONNAISSANCES ET DES COMPETENCES
1- CONNAISSANCES ABORDEES DU PROGRAMME

Connaissances abordées du programme	Niveau ciblé : expression	Savoir appris maîtrisé	Je saurai en parler	Non maîtrisé
<i>Langage de programmation</i>				

2- COMPETENCES ABORDEES DU PROGRAMME

Compétences abordées du programme	Acquis	Je saurai refaire avec de l'aide	Non acquis
<i>Traduire un algorithme en un programme exécutable</i>			

Programmes Python (se reporter au dossier technique pour plus de détails)**AppBot_API()**

- Description : fonction d'initialisation à exécuter impérativement au début de chaque programme. Elle lance en particulier en tâche de fond le client Websocket destiné à envoyer les ordres de pilotage au robot

Avancer(vitesseLongitudinale, duree=-1)

- Paramètres :
 - vitesseLongitudinale : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ', t étant reconnu comme le temps courant). Consigne de vitesse longitudinale (m/s), saturée en interne entre -0.5 et 0.5 m/s
 - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut : -1 (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne une consigne de vitesse longitudinale au robot pendant une certaine durée

Tourner(vitesseRotation, duree=-1)

- Paramètres :
 - vitesseRotation : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ', t étant reconnu comme le temps courant). Consigne de vitesse de rotation (deg/s), saturée en interne entre -180 et 180 deg/s
 - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut : -1 (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne au robot une consigne de vitesse de rotation autour de la verticale pendant une certaine durée

Mouvement(vitesseLongitudinale, vitesseRotation, duree=-1)

- Paramètres :
 - vitesseLongitudinale : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ', t étant reconnu comme le temps courant). Consigne de vitesse longitudinale (m/s), saturée en interne entre -0.5 et 0.5 m/s
 - vitesseRotation : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ', t étant reconnu comme le temps courant). Consigne de vitesse de rotation (deg/s), saturée en interne entre -180 et 180 deg/s
 - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut : -1 (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne une combinaison de consignes de mouvement au robot pendant une certaine durée

Terminer()

- Description : fonction à exécuter impérativement à la fin de chaque programme