

# Master 1 Sciences Cognitives

---

## Rapport TD2 Robotique Social

---

**Auteurs :**

Blondelle Corentin  
Baptiste Billard  
Alexis Beer  
Aaron Costescu

**Professeur :**

Ferreira Chame

Année universitaire 2024-2025
-------------------------------

1. Introduction :	3
2. TD2 A – Introduction à Nao et Pepper :	3
3. TD2 B – Introduction à la communication en IHR :	4
3.1. Partie 1 : Production de parole animé :	4
3.2. Partie 2 : Gestionnaire de dialogues :	4
4. TD2 C – Localisation et gestion de l’espace en IHR :	5
4.1. Partie 1 : Zones d’interaction et dialogue :	5
4.2. Partie 2 : Référencer un objet connu :	5
5. Conclusion :	6

## 1. Introduction :

Ce rapport présente les différentes étapes du TD2 de robotique sociale, réalisé avec les robots Nao et Pepper. Le TD était divisé en trois grandes parties : une introduction à la programmation de comportements de base (TD2-A), une exploration de la communication verbale et animée (TD2-B), puis un approfondissement autour de la gestion de l'espace et de la perception (TD2-C). Chaque partie comportait plusieurs exercices pratiques, développés en langage Python en s'appuyant sur le SDK NaoQi.

Pour chaque étape, nous exposerons les objectifs fixés et atteints, puis nous détaillerons les éventuelles difficultés rencontrées au cours de la mise en œuvre. Ce rapport vise ainsi à rendre compte de notre progression, de notre compréhension des concepts abordés, ainsi que des solutions mises en œuvre face aux obstacles rencontrés.

## 2. TD2 A – Introduction à Nao et Pepper :

La première partie du TD2 avait pour objectif de nous initier à la programmation des robots Nao et Pepper, en utilisant à la fois l'environnement graphique Choregraphe et des scripts Python autonomes. Le but principal était de faire prononcer au robot une phrase simple comme « Hello, world! », d'abord via une interface graphique, puis directement par un script Python. Par ailleurs, nous avons découvert l'utilisation de modules fondamentaux tels que ALTextToSpeech, et appris à faire pointer et orienter la tête du robot vers une position donnée.

L'ensemble des objectifs a été atteint avec succès. Nous avons tout d'abord programmé le robot pour qu'il dise une phrase via Choregraphe. Ensuite, nous avons reproduit ce comportement dans un script Python autonome que nous avons exécuté en local sur notre machine. Ce script utilisait un proxy vers le module ALTextToSpeech pour déclencher la parole du robot. Enfin, nous avons également exploré les modules permettant au robot de pointer vers une position spécifique et d'orienter sa tête, comme demandé dans les consignes.

Nous avons néanmoins rencontré plusieurs difficultés techniques pendant cette séance. Dès le début, Choregraphe plantait régulièrement sur les ordinateurs de la faculté, ce qui a ralenti nos essais et nous a contraints à privilégier les tests sur le simulateur virtuel plutôt que sur les robots physiques.

### 3. TD2 B – Introduction à la communication en IHR :

#### 3.1. Partie 1 : Production de parole animé :

Cette première sous-partie du TD2-B avait pour objectif de renforcer les capacités expressives du robot en liant la parole à des gestes physiques. Pour cela, nous avons utilisé le module `ALAnimatedSpeech`, qui permet d'annoter un texte avec des balises déclenchant des mouvements du robot. Le but était de prototyper différentes émotions (la confusion, l'excitation, la tristesse et le doute) en combinant parole et animation.

L'ensemble des objectifs fixés a été entièrement réalisé. Nous avons conçu un script Python qui envoie plusieurs phrases annotées au robot. Chaque phrase exprimait une émotion précise, associée à un comportement corporel du robot via des balises comme `^start()` et `^wait()`. Par exemple, pour exprimer la confusion, le robot effectuait le geste « Think\_1 » tout en disant qu'il ne comprenait pas bien. Pour l'excitation, un geste enthousiaste était combiné à une exclamation verbale, et ainsi de suite pour la tristesse et le doute.

Concernant les difficultés pour cette partie, on en a eu qu'une seule. Elle consistait à comprendre comment fonctionnaient précisément les balises d'annotation dans `ALAnimatedSpeech`, car la documentation n'était pas toujours explicite sur certains comportements ou combinaisons possibles.

#### 3.2. Partie 2 : Gestionnaire de dialogues :

Dans cette seconde sous-partie du TD2-B, nous avons découvert le module `ALDialog` et la syntaxe `QiChat`, qui permet de concevoir des dialogues structurés avec le robot. L'objectif était de créer un fichier de règles conversationnelles et de l'intégrer dans un programme Python autonome, sans dépendre de Choregraphe.

Nous avons entièrement atteint l'objectif. Le robot a été capable de gérer un petit dialogue sur le thème des animaux, en répondant à plusieurs types de questions et en relançant la conversation. L'écriture des règles en `QiChat` a été facilitée par les exemples disponibles, et l'intégration via le service `ALDialog` s'est bien déroulée.

Nous avons toutefois rencontré une difficulté principale : comprendre la syntaxe exacte pour écrire les topics en `QiChat`. Nous avons notamment tenté d'ajouter une réponse par défaut en cas de phrase hors sujet, mais nous n'avons pas réussi à la faire fonctionner correctement.

## 4. TD2 C – Localisation et gestion de l'espace en IHR :

### 4.1. Partie 1 : Zones d'interaction et dialogue :

Dans cette première partie du TD2-C, nous avons travaillé sur l'intégration des zones d'engagement dans un prototype conversationnel. L'objectif était de configurer les distances des zones sociales du robot, détecter l'entrée d'une personne dans la zone intime, et déclencher un dialogue de manière proactive.

Nous avons atteint tous les objectifs. Le robot détectait correctement lorsqu'un individu entrait dans la zone proche et lançait automatiquement une interaction verbale. Nous avons utilisé le module `ALEngagementZones` pour configurer les seuils de distance, et connecté un événement de perception à un script de dialogue, déjà implémenté dans la partie TD2-B2.

Les principales difficultés rencontrées étaient liées à la compréhension de la façon dont les événements sont structurés dans `ALMemory` et comment s'abonner correctement à ces événements. De plus, comme nous n'avons pas pu tester cette partie sur un vrai robot, nous nous sommes basés uniquement sur la documentation et la simulation. Cela a limité notre capacité à valider pleinement la réactivité du système dans un contexte réel.

### 4.2. Partie 2 : Référencer un objet connu :

Dans cette seconde partie du TD2-C, nous avons appris à faire pointer le robot vers un objet détecté dans son environnement, en l'occurrence un marqueur visuel (Naomark). L'objectif était d'intégrer les modules de perception visuelle, d'estimer la position du marqueur par rapport au robot, puis d'utiliser les fonctions `lookAt` et `pointAt` pour qu'il oriente sa tête et son bras vers celui-ci.

Nous avons réussi à atteindre les objectifs. Le code met en place un détecteur de marqueur en s'abonnant à l'événement `LandmarkDetected` via `ALMemory`, puis calcule la position du marqueur à l'aide de transformations géométriques utilisant la bibliothèque `almath`. Une fois la position estimée, le robot tourne la tête et pointe avec son bras vers le marqueur détecté, ce qui permet un comportement spatialement ciblé.

Nous avons toutefois rencontré une difficulté technique importante au moment de tester le script : l'importation du module `almath` a posé problème sur notre environnement de test. Le module ne se chargeait pas correctement, ce qui nous a empêchés de vérifier pleinement le bon déroulement des calculs et de la transformation des coordonnées. Bien que nous ayons compris l'essentiel du fonctionnement du script et son lien avec la géométrie des référentiels, cette limitation a restreint notre capacité à valider l'exécution complète de cette partie.

## 5. Conclusion :

Nous avons largement préféré ce TD2 par rapport au TD1, principalement parce qu'il nous a permis d'interagir concrètement avec un robot. Le fait de pouvoir observer des comportements réels, même en simulation, rend l'apprentissage bien plus engageant. Travailler sur des aspects concrets de la robotique sociale, comme la parole animée, la gestion du dialogue ou encore le pointage vers un objet détecté, nous a permis de mieux comprendre l'intérêt des concepts théoriques en les mettant en pratique.

Tout au long du TD2, nous avons atteint l'ensemble des objectifs proposés. Nous avons appris à utiliser les principaux modules du SDK NaoQi, à manipuler les fonctions d'expression verbale et non verbale, à gérer des dialogues en QiChat, et à intégrer des éléments de perception pour enrichir l'interaction. En général ce TD a été motivant, formateur, et très satisfaisant à réaliser.