

# **Spécifications Techniques/analyse fonctionnelle pour le Projet de Robot Assistant Dirigé par Commandes Vocales**

**Présenté par Marie Doms, Basile Lorient**

<b>Spécifications Techniques/analyse fonctionnelle pour le Projet de Robot Assistant Dirigé par Commandes Vocales</b>	<b>1</b>
<b>Présenté par Marie Doms, Basile Lorient</b>	<b>1</b>
<b>1 - Intended Purpose (Objectif visé)</b>	<b>2</b>
1a - Operational Domain Design (ODD)	2
<b>2 - Spécifications du Système</b>	<b>4</b>
2a - Interaction avec l'Utilisateur	4
2b - Interaction avec l'Environnement	5
2c - Performances du Système	5
2d - Synthèse des Besoins des Parties Prenantes	6
2e - Objectif d'Automatisation	7
Contraintes Opérationnelles (ODC) : Gestion des Ressources Technologiques	7
<b>3 - Définition de l'architecture de votre système et de son design.</b>	<b>7</b>
3a - Choix Technologiques	7
3b - Architecture du Système	8
<b>4 - Spécifications détaillées de votre composant porteur du Machine Learning.</b>	<b>9</b>
Composant Hardware pour le Machine Learning (Voiture sans fil) :	9
Processeur Central (CPU) :	9
Mémoire :	10
Connectivité :	10
Alimentation :	10
<b>5 - Architecture du Composant Machine Learning pour la Voiture Autonome</b>	<b>12</b>
Acquisition et Transmission des Données :	12
Prétraitement et Traitement des Données :	13
Stockage et Inférence des Modèles :	14
Contrôle et Navigation :	14
Interface Utilisateur :	15
Gestion des Données :	15
<b>6 - Partie Data Engineering</b>	<b>15</b>
Partie 2: Développement, NLP/Prompt Engineering	17
Logiciel Speech to Text	17
Speech Synthesis from Text	17

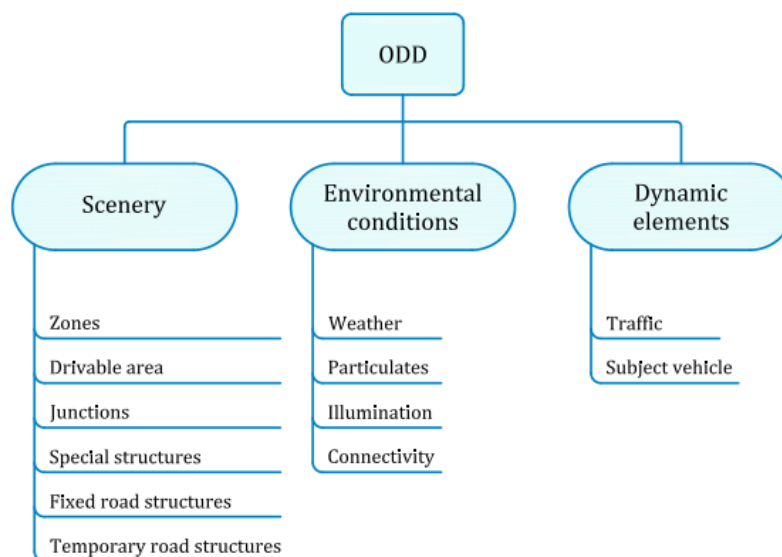
# 1 - Intended Purpose (Objectif visé)

L'objectif de l'utilisateur final est de disposer d'un système de commande vocale robuste et intuitif pour contrôler les mouvements du robot assistant médical dans les hôpitaux.

Indépendamment de l'implémentation, le système doit permettre à l'utilisateur de communiquer efficacement avec le robot, en lui transmettant des instructions claires et précises. La robustesse du système est cruciale pour assurer un fonctionnement fiable, même dans des conditions d'utilisation variées et parfois bruyantes. De plus, l'explicabilité du système est importante pour permettre à l'utilisateur de comprendre les réponses et les actions du robot.

## 1a - Operational Domain Design (ODD)

Le domaine opérationnel dans lequel le système va fonctionner comprend les hôpitaux et les environnements médicaux. Les conditions et les contraintes spécifiques de ce domaine incluent :



- Environnement bruyant:
  - Les hôpitaux sont des environnements caractérisés par un niveau élevé de bruit. Les activités médicales telles que les machines, les alarmes, les discussions entre le personnel médical et les patients, ainsi que le déplacement constant du personnel contribuent à ce bruit ambiant. Le système de commande vocale doit être capable de fonctionner efficacement dans un tel environnement bruyant, en utilisant des techniques de traitement

du signal robustes pour filtrer le bruit de fond et identifier avec précision les commandes vocales de l'utilisateur.

- Mobilité restreinte :
  - Le robot doit être capable de naviguer de manière autonome à travers des espaces restreints, tels que des couloirs étroits, des portes, et potentiellement des espaces encombrés par des équipements médicaux, des chariots et d'autres obstacles. Il devra également être capable de détecter et d'éviter les personnes et les objets qui se trouvent sur son chemin, en utilisant des capteurs de proximité et des algorithmes de planification de trajectoire adaptés à des environnements complexes et dynamiques.
- Sécurité des patients :
  - La sécurité des patients est une priorité absolue. Le robot doit être conçu pour éviter les collisions avec les patients, le personnel médical et les équipements médicaux. Il doit être doté de capteurs de détection d'obstacles à haute précision et de systèmes d'évitement de collision robustes. De plus, le robot doit être programmé pour interagir de manière respectueuse et non intrusive avec les patients, en évitant tout comportement qui pourrait causer de l'anxiété ou du stress.
- Confidentialité des données :
  - Tout échange vocal entre l'utilisateur et le robot doit garantir la confidentialité des informations médicales des patients. Le système doit être conforme aux réglementations en matière de protection des données, telles que le RGPD (Règlement Général sur la Protection des Données) dans l'Union européenne.
- Fiabilité et disponibilité :
  - Le système doit être disponible en permanence pour répondre aux besoins des patients et du personnel médical, avec une fiabilité élevée pour éviter les interruptions de service. Cela nécessite une conception robuste et des mécanismes de surveillance en temps réel pour détecter et résoudre rapidement les éventuelles défaillances. Le système doit également être doté de fonctionnalités de redondance et de sauvegarde pour assurer une continuité de service en cas de panne matérielle ou logicielle. En outre, des procédures de maintenance préventive doivent être mises en place pour garantir le bon fonctionnement à long terme du système.



En prenant en compte ces aspects, le système doit être conçu pour fonctionner de manière efficace et sécurisée dans l'environnement complexe et dynamique des hôpitaux, en garantissant une interaction intuitive et sûre entre l'utilisateur et le robot assistant.

## **2 - Spécifications du Système**

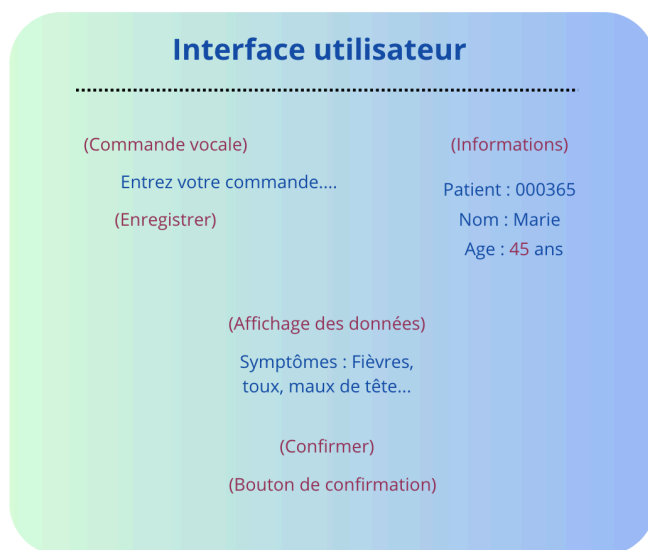
### **2a - Interaction avec l'Utilisateur**

- **Système de Commande vocale:** Le robot sera équipé d'un système de commande vocale permettant à l'utilisateur de contrôler ses mouvements et d'interagir avec lui en utilisant des commandes vocales simples et intuitives.
- **Technologie de Reconnaissance Vocale Avancée :**
  - Le système utilisera des algorithmes de reconnaissance vocale avancés, basés sur des réseaux neuronaux, pour détecter et comprendre les commandes vocales de l'utilisateur avec une grande précision.
- **Commandes Intuitives et Naturelles :**
  - Les commandes vocales seront conçues pour être intuitives et naturelles, reflétant le langage courant utilisé par les utilisateurs pour contrôler le robot. Par exemple, "Robot, avance" ou "Robot, arrête-toi".
- **Traitement du Langage Naturel (NLP):** Des techniques avancées de NLP seront utilisées pour comprendre et interpréter les conversations entre l'utilisateur et le robot, afin de capturer, segmenter et résumer les problèmes des patients de manière précise.

Le traitement du langage naturel (NLP) permettra au robot de comprendre et d'interpréter les conversations entre l'utilisateur et lui-même, afin de capturer, segmenter et résumer les problèmes des patients de manière précise. Voici des détails supplémentaires sur cette fonctionnalité :

- **Analyse des Intentions et des Entités :**
  - Le système utilisera des techniques avancées de NLP pour analyser les intentions derrière les phrases de l'utilisateur et extraire les entités pertinentes, telles que les symptômes, les demandes d'assistance, etc.
- **Segmentation des Conversations :**
  - Les conversations seront segmentées en unités logiques, telles que les questions, les réponses et les demandes, pour faciliter le traitement et la compréhension par le robot.
- **Résumé des Problèmes des Patients :**
  - Le système générera un résumé clair et concis des problèmes des patients à partir des conversations avec l'utilisateur, facilitant ainsi la communication avec le personnel médical.
- **Utilisation de Modèles de Langage Pré-Entraînés :**

- Des modèles de langage pré-entraînés seront utilisés pour améliorer la précision et la performance du traitement du langage naturel, permettant ainsi une compréhension plus contextuelle des interactions.
- Interface Utilisateur Conviviale: Une interface utilisateur conviviale sera développée pour permettre à l'utilisateur d'interagir facilement avec le robot, en confirmant les commandes vocales et en affichant les informations pertinentes de manière claire et compréhensible.



## 2b - Interaction avec l'Environnement

- Navigation et Mouvement: Le robot sera capable de se déplacer de manière autonome dans l'environnement hospitalier, en utilisant des techniques de navigation avancées pour gérer des virages et recalibrer sa position en cas de besoin.
- Détection et Évitement d'Obstacles: Des capteurs seront intégrés au robot pour détecter les obstacles sur son chemin et éviter les collisions, assurant ainsi une navigation sûre et fluide dans les couloirs et les chambres des patients.

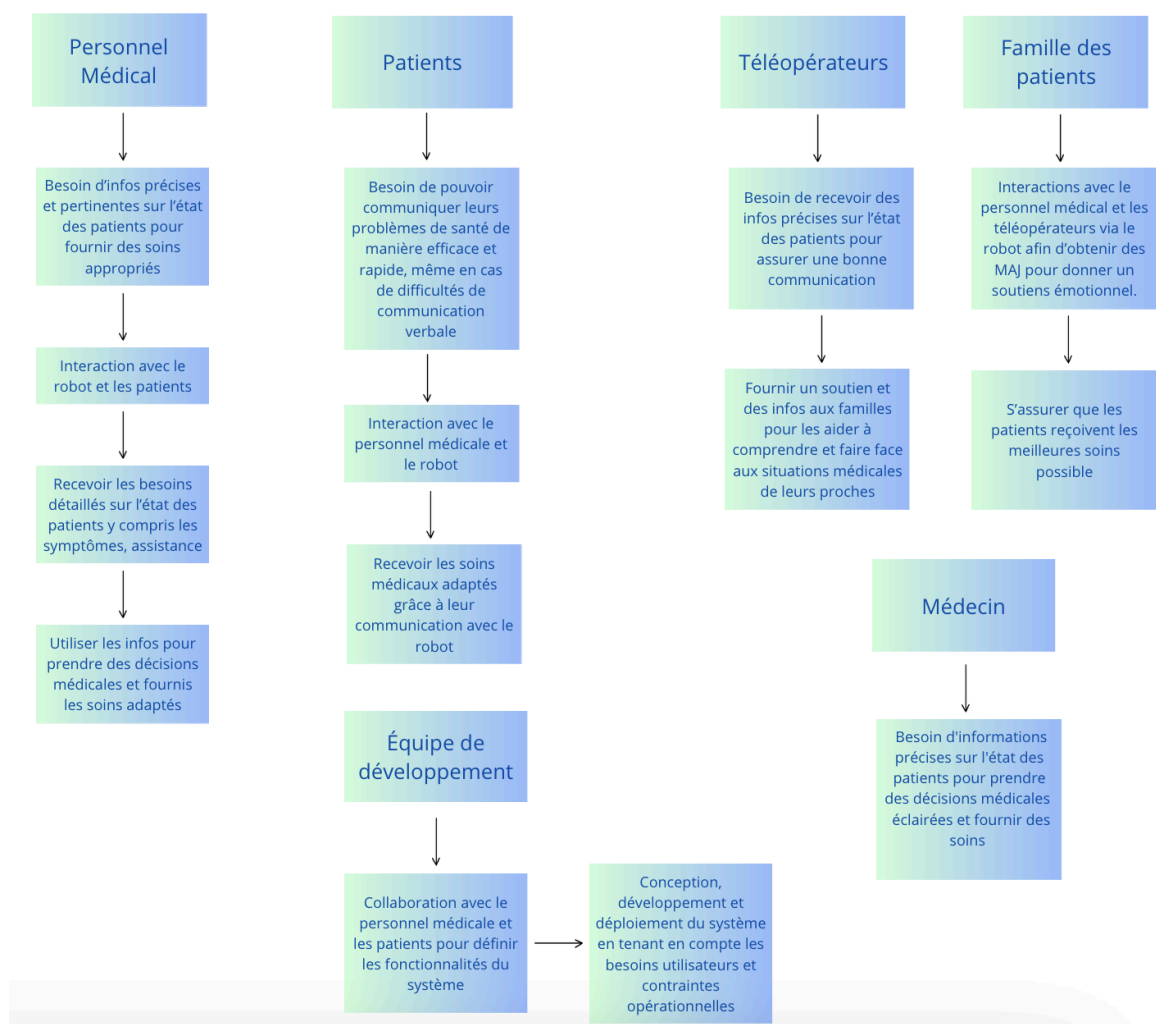
## 2c - Performances du Système

- Précision de la Reconnaissance Vocale: Le système de reconnaissance vocale devra avoir une précision élevée pour comprendre et exécuter correctement les instructions de l'utilisateur, garantissant ainsi une communication efficace.
- Réactivité du Robot: Le robot devra répondre rapidement aux commandes de l'utilisateur et aux situations d'urgence, assurant ainsi une interaction fluide et réactive.
- Fiabilité de la navigation: La capacité du robot à naviguer de manière fiable dans l'environnement hospitalier, en évitant les obstacles et en recalibrant sa position au besoin, est cruciale pour son efficacité opérationnelle.

## 2d - Synthèse des Besoins des Parties Prenantes

- Personnel médical: Besoin de recevoir des informations précises et pertinentes sur l'état des patients pour fournir des soins appropriés et rapides.
- Patients: Besoin de pouvoir communiquer leurs problèmes de santé de manière efficace et rapide, même en cas de difficultés de communication verbale.
- Équipe de Développement: Besoin de comprendre les exigences fonctionnelles et techniques du projet pour concevoir et mettre en œuvre le système de manière appropriée.
- Téléopérateur : Personne qui sera chargé de gérer par guide vocal la voiture autonome.

Pour plus de détails voici un schéma explicatif :



## 2e - Objectif d'Automatisation

L'objectif principal de l'automatisation est de réduire la charge de travail du personnel médical en permettant aux robots assistants de recueillir et de transmettre les informations des patients de manière autonome. Cela permettra une utilisation plus efficace des ressources humaines et une meilleure prise en charge des patients.

Contraintes Opérationnelles (ODC) : Gestion des Ressources Technologiques

- Budget Limité : Les choix technologiques doivent être optimisés pour respecter les contraintes budgétaires, en privilégiant les solutions rentables et les technologies accessibles financièrement.
- Disponibilité limitée des compétences : Les compétences requises pour la mise en œuvre des technologies choisies peuvent être limitées. Il est essentiel de maximiser l'utilisation des compétences disponibles et de former le personnel si nécessaire.
- Compatibilité Technologique : Le système doit être compatible avec les infrastructures technologiques existantes dans les hôpitaux pour faciliter son intégration et son déploiement. Cela nécessite une évaluation approfondie des technologies disponibles sur le marché et de leur compatibilité avec l'environnement hospitalier.
- Performances Requises : Les technologies choisies doivent répondre aux exigences de performance du système, telles que la vitesse de traitement des commandes vocales et la réactivité du robot. Il est crucial de sélectionner des technologies capables de fournir les performances nécessaires tout en respectant les contraintes budgétaires et temporelles.

## **3 - Définition de l'architecture de votre système et de son design.**

### 3a - Choix Technologiques

Pour concevoir l'architecture du système de commande vocale et de contrôle du robot assistant médical, nous avons sélectionné les technologies suivantes :

- Reconnaissance Vocale :

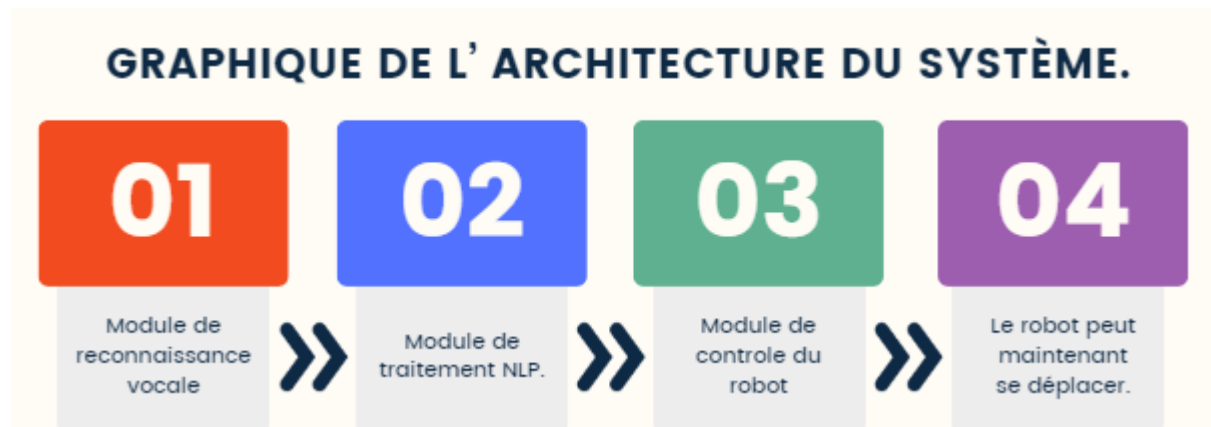
- Nous utiliserons la librairie “speech\_recognition” pour la reconnaissance vocale. Cette librairie offre une excellente précision et prend en charge plusieurs langues. De plus, elle est facile à intégrer dans notre système.
- Traitement du Langage Naturel (NLP) :
  - Pour le traitement du texte généré par la reconnaissance vocale, nous utiliserons la bibliothèque NLTK (Natural Language Toolkit) en Python. NLTK offre une large gamme d'outils pour le traitement du langage naturel, y compris la tokenization, la lemmatisation et la classification de texte.
- Communication avec le Robot\*\* :
  - La communication avec le robot se fera via une connexion Wi-Fi. Nous utiliserons la bibliothèque `socket` en Python pour établir une connexion TCP/IP entre le système de commande vocale et le Raspberry Pi du robot. Cette approche permettra un contrôle en temps réel et une communication bidirectionnelle entre le système et le robot.

### 3b - Architecture du Système

L'architecture du système se compose des modules suivants :

- Module de Reconnaissance Vocale :
  - Ce module sera chargé de convertir les commandes vocales de l'utilisateur en texte. Nous utiliserons l'API Google Speech-to-Text pour cette tâche.
- Module de Traitement NLP :
  - Le texte généré par le module de reconnaissance vocale sera traité par ce module. NLTK sera utilisé pour capturer, segmenter et résumer les conversations entre l'utilisateur et le robot.
- Module de Contrôle du Robot :
  - Ce module sera responsable de la génération d'instructions logiques à partir des commandes vocales traitées. Ces instructions seront ensuite transmises au robot via la connexion Wi-Fi établie.





Ce diagramme illustre les principaux modules du système et la séquence de traitement des données. Les commandes vocales de l'utilisateur sont converties en texte, puis traitées par le module NLP pour générer des instructions de contrôle du robot. Enfin, ces instructions sont envoyées au robot via la connexion Wi-Fi établie.

## **4 - Spécifications détaillées de votre composant porteur du Machine Learning.**

Stockage et Inférence des Modèles :

Une fois entraînés, les modèles de Machine Learning sont stockés sur l'ordinateur central. Lors de l'opération en temps réel de la voiture, ces modèles sont utilisés pour l'inférence afin de prendre des décisions de contrôle et de navigation en fonction des données sensorielles actuelles.

Contrôle et Navigation :

Les prédictions issues des modèles de Machine Learning sont utilisées pour générer des commandes de mouvement pour la voiture autonome, permettant ainsi une navigation autonome en évitant les obstacles détectés et en suivant une trajectoire sûre et efficace.

Interface Utilisateur :

Une interface utilisateur graphique (GUI) est intégrée à l'ordinateur central pour afficher les informations sur l'état de la voiture, les obstacles détectés et les instructions de navigation. De plus, un système de reconnaissance vocale est également inclus pour permettre aux utilisateurs de donner des instructions vocales à la voiture

Composant Hardware pour le Machine Learning (Voiture sans fil) :

Processeur Central (CPU) :

- Architecture à Faible Consommation : Le système embarqué doit être équipé d'un processeur basse consommation pour prolonger l'autonomie de la batterie tout en fournissant des performances adéquates.
- Support d'Instructions SIMD : Bien que le processeur ne soit pas aussi puissant que les versions de bureau, un support d'instructions SIMD peut accélérer certaines opérations de traitement des données.

Mémoire :

- Mémoire Vive (RAM) : Une quantité suffisante de RAM embarquée est nécessaire pour charger et manipuler les informations reçues de l'ordinateur. Le LLM sera lui géré depuis l'ordinateur car il va demander trop de RAM, et cela pourrait compromettre le système du raspberry PI4 qui lui, possède que 8go RAM.
- Batterie du Raspberry : Pour une utilisation de 1 heure, la capacité de la batterie d'un Raspberry Pi pour une voiture autonome à assistance vocale pourrait varier entre 2000 et 5000 milliampères-heures, en fonction de la consommation d'énergie du système et des composants périphériques.

La consommation typique d'un Raspberry Pi en fonctionnement est d'environ 2 à 5 watts, selon le modèle spécifique et les opérations effectuées. Pour simplifier, nous utiliserons une valeur moyenne de 3 watts.

Pour estimer l'énergie consommée en une heure (en wattheures, Wh), nous multiplions la puissance en watts par le temps en heures :  $\text{Energie consommée (Wh)} = \text{Puissance (W)} \times \text{Temps (h)}$

Donc, pour une utilisation de 1 heure :  $\text{Energie consommée (Wh)} = 3 \text{ watts} \times 1 \text{ heure} = 3 \text{ Wh}$

Cette estimation est basée sur une consommation moyenne et peut varier en fonction des composants supplémentaires, des charges de travail spécifiques et des conditions environnementales.

Connectivité :

- Wi-Fi : Le système embarqué doit être équipé d'une connectivité Wi-Fi qui ici est optimale pour notre salle. Elle permettra la communication sans fil avec d'autres appareils, le cloud ou le réseau local de l'hôpital. Il est nécessaire d'utiliser le même réseau pour notre projet. Un répéteur bridge permettrait de le connecter sans avoir besoin de réintégrer les mêmes specs réseaux sur notre Raspberry py.
- Bluetooth Low Energy (BLE) : La technologie BLE peut être utilisée pour la communication avec d'autres appareils ou capteurs de proximité, facilitant ainsi les interactions avec l'environnement de l'hôpital. Nous ne savons pas exactement encore quel type de bluetooth est utilisé dans le raspberry PI4.

Alimentation :

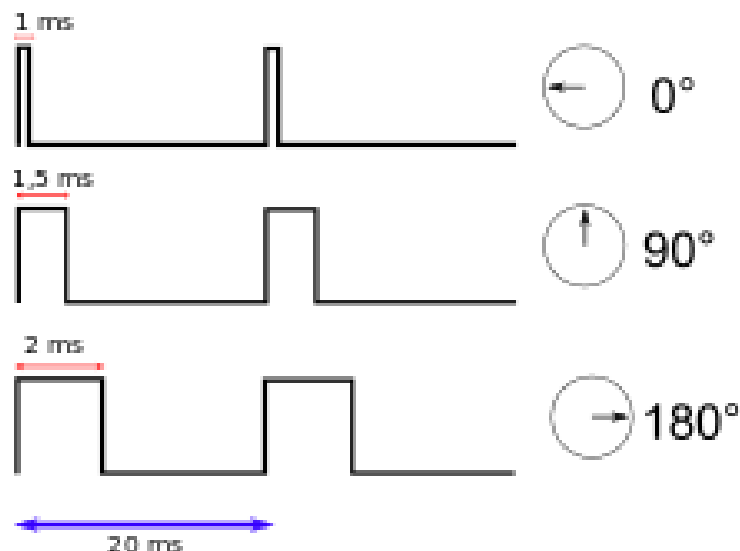
- **Batterie Rechargeable** : Une batterie rechargeable intégrée offre une autonomie de fonctionnement sans fil, permettant à la voiture de se déplacer dans l'hôpital sans être limitée par des câbles d'alimentation.

Une fois la technologie développée, dans un second temps, il serait peut-être intéressant de pousser encore plus la solution en lui intégrant des capteurs pour une V2 :

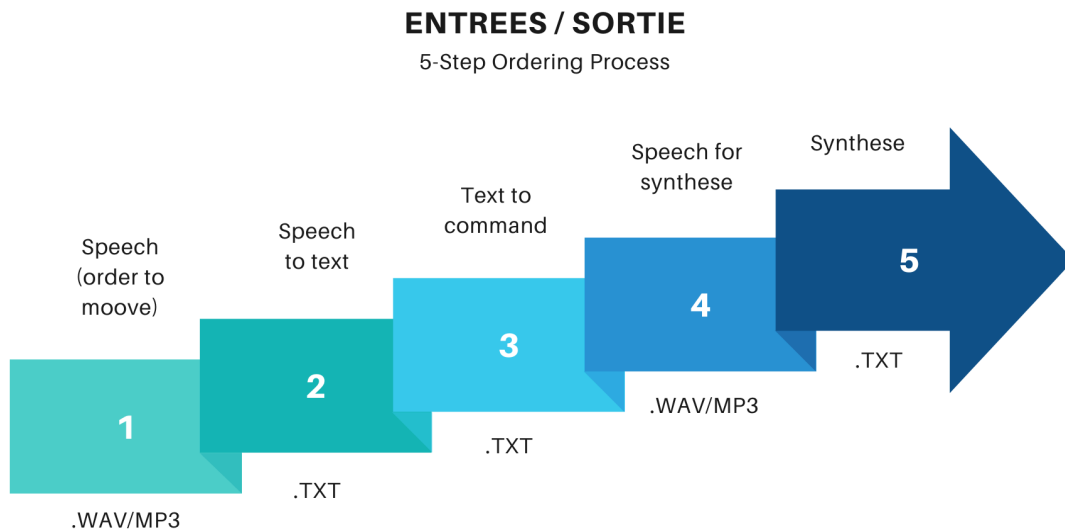
- **Caméra** : Une caméra embarquée peut être utilisée pour la perception de l'environnement, la détection d'obstacles et la navigation autonome de la voiture dans l'hôpital.
- **Capteurs à ultrasons** : Les capteurs à ultrasons peuvent compléter la perception de l'environnement en détectant les obstacles à proximité et en permettant d'éviter les collisions.

Dans notre projet, le capteur à ultrason est pertinent pour plusieurs raisons :

1. **Détection d'obstacles et navigation autonome** : Le capteur à ultrasons monté sur le servomoteur permettra au robot de détecter les obstacles dans son environnement et d'ajuster son orientation en conséquence. Cela est essentiel pour assurer une navigation autonome sûre dans les couloirs et les chambres d'hôpital, en évitant les collisions avec les patients, le personnel médical et les équipements médicaux.
2. **Précision de l'axe visé** : L'explication fournit des informations sur la précision de l'axe visé par le capteur à ultrason monté sur le servomoteur. Avec une précision angulaire de  $\pm 20^\circ$ , le robot peut orienter le capteur de manière à obtenir des mesures plus précises dans une direction spécifique. Cela peut être utile pour cibler des zones spécifiques dans l'environnement hospitalier ou pour effectuer des tâches de surveillance ciblées.
3. **Gestion des contraintes matérielles** : La description du montage sur un servomoteur bas coût et de la faible précision angulaire du capteur à ultrason souligne les contraintes matérielles auxquelles le système est confronté. Comprendre ces limitations est important lors de la conception et du développement du système, car cela permet de prendre des décisions appropriées en matière de conception et de calibration pour maximiser l'efficacité et la fiabilité du capteur.



## 5 - Architecture du Composant Machine Learning pour la Voiture Autonome



L'architecture du composant Machine Learning pour la voiture autonome repose sur plusieurs couches fonctionnelles, depuis l'acquisition des données jusqu'à la prise de décision et la navigation autonome. Voici une présentation détaillée de notre architecture :

Acquisition et Transmission des Données :

Les données sensorielles sont collectées par les capteurs embarqués de la voiture autonome, notamment des caméras pour la perception visuelle de l'environnement et des capteurs à ultrasons pour la détection d'obstacles proches. Ces données sont ensuite transmises à l'ordinateur central via une connexion sans fil, telle que le Wi-Fi ou le Bluetooth, pour un traitement ultérieur.

**Entrées/Sorties (E/S) :**

- Entrées :
  - Flux audio des commandes vocales captées par les microphones embarqués dans la voiture autonome.
- Sorties :
  - Texte transcrit des commandes vocales.
  - Segmentation des conversations en unités logiques telles que les questions, les réponses et les demandes.

**Modèle :**

- Réseau de neurones récurrents (RNN) :

- Les RNN sont bien adaptés pour traiter des séquences de données, ce qui en fait un choix approprié pour la transcription vocale et la segmentation des conversations.
- Ce modèle peut être utilisé pour encoder les flux audio en texte et pour segmenter les conversations en unités logiques.

### **Algorithme :**

- Algorithme de transcription vocale :
  - Les algorithmes de transcription vocale convertissent les flux audio en texte.
  - Des modèles basés sur des réseaux de neurones tels que les réseaux de neurones récurrents (RNN) ou les réseaux de neurones transformer (Transformer) sont couramment utilisés pour cette tâche.
- Algorithme de segmentation de conversations :
  - Les algorithmes de segmentation de conversations divisent les flux de texte en unités logiques, telles que les questions, les réponses et les demandes.
  - Des techniques de traitement du langage naturel (NLP) comme la tokenization et la détection de phrases peuvent être utilisées pour cette tâche.

### **Prétraitement et Traitement des Données :**

Une fois les données reçues par l'ordinateur central, elles subissent un processus de prétraitement visant à éliminer le bruit et à normaliser les informations sensorielles. Ensuite, les caractéristiques pertinentes sont extraites des données traitées à l'aide de techniques de traitement d'images et d'analyse de données.

### **Entrées/Sorties (E/S) :**

- Entrées :
  - Données brutes collectées par les capteurs embarqués, telles que des flux audio pour la reconnaissance vocale ou des données de capteurs pour la perception de l'environnement.
- Sorties :
  - Données prétraitées et nettoyées, prêtes pour l'extraction de caractéristiques.
  - Caractéristiques pertinentes extraites des données pour être utilisées dans les modèles de Machine Learning ultérieurs.

### **Modèle :**

- Aucun modèle spécifique n'est utilisé dans cette phase. Le prétraitement et le traitement des données sont généralement réalisés à l'aide de techniques de traitement du signal et d'analyse de données plutôt que de modèles d'apprentissage automatique.

### **Entraînement des Modèles de Machine Learning :**

Les données prétraitées et leurs caractéristiques extraites sont utilisées pour entraîner les modèles de Machine Learning, tels que les réseaux de neurones convolutifs (CNN) pour la détection d'objets et la segmentation d'images, ainsi que les réseaux de neurones récurrents (RNN) pour la prédiction de trajectoire et la planification de mouvement. Ces modèles sont entraînés sur l'ordinateur central à partir des données collectées lors des tests et des opérations de la voiture autonome.

### **Entrées/Sorties (E/S) :**

- Entrées :
  - Données audio représentant les commandes vocales captées par les microphones embarqués dans la voiture autonome.
- Sorties :
  - Transcriptions textuelles des commandes vocales converties par le modèle de Voice-to-Text.

### **Modèle :**

- Modèle de Conversion de la Voix en Texte :
  - Il s'agit d'un modèle de reconnaissance vocale entraîné spécifiquement pour convertir les données audio en texte.
  - Ce modèle peut être basé sur des architectures de réseaux neuronaux récurrents (RNN) ou de transformer, adaptées à la conversion de la voix en texte.

### **Algorithme :**

- L'algorithme principal utilisé dans ce modèle est généralement le traitement automatique de la parole (ASR), qui peut être implémenté à l'aide d'architectures de réseaux neuronaux profonds.
- Les réseaux récurrents (RNN) et les réseaux de transformer sont couramment utilisés pour cette tâche, avec des techniques telles que le mécanisme d'attention pour se concentrer sur les parties pertinentes de l'entrée audio lors de la conversion en texte.
- Pour la segmentation des données, des algorithmes de traitement du signal numérique peuvent être utilisés pour découper les enregistrements audio en segments appropriés pour l'entraînement du modèle de Voice-to-Text.

### **Stockage et Inférence des Modèles :**

Une fois entraînés, les modèles de Machine Learning sont stockés sur l'ordinateur central. Lors de l'opération en temps réel de la voiture, ces modèles sont utilisés pour l'inférence afin de prendre des décisions de contrôle et de navigation en fonction des données sensorielles actuelles.

### **Contrôle et Navigation :**

Les prédictions issues des modèles de Machine Learning sont utilisées pour générer des commandes de mouvement pour la voiture autonome, permettant ainsi une navigation autonome en évitant les obstacles détectés et en suivant une trajectoire sûre et efficace.

Interface Utilisateur :

Une interface utilisateur graphique (GUI) est intégrée à l'ordinateur central pour afficher les informations sur l'état de la voiture, les obstacles détectés et les instructions de navigation. De plus, un système de reconnaissance vocale est également inclus pour permettre aux utilisateurs de donner des instructions vocales à la voiture.

Gestion des Données :

L'ordinateur central est également responsable de la gestion des données collectées par la voiture, y compris le stockage, la sauvegarde et l'analyse pour améliorer les performances du système.

Cette architecture complète du composant Machine Learning assure une intégration transparente et efficace des fonctionnalités de traitement des données, de prise de décision et de navigation autonome dans le cadre d'un hôpital.

## **6 - Partie Data Engineering**

Pour entraîner le modèle final de reconnaissance vocale et de traitement NLP, nous aurons besoin des types de données suivants :

- **Données de Reconnaissance Vocale :**
  - Des enregistrements audio contenant des commandes vocales dans différents contextes hospitaliers.
  - Ces enregistrements doivent couvrir une variété de conditions environnementales, y compris le bruit de fond typique des hôpitaux.
- **Données Textuelles :**
  - Des transcriptions des enregistrements audio correspondants, obtenues à l'aide de la reconnaissance vocale.
  - Ces transcriptions serviront de données d'entraînement pour le traitement NLP.
- **Données d'Annotation :**
  - Des annotations manuelles des transcriptions pour identifier les commandes vocales pertinentes et les intentions de l'utilisateur.
  - Ces annotations aideront à superviser l'apprentissage du modèle et à évaluer sa performance.
- **Données de Validation :**

- Un ensemble de données distinct pour évaluer la performance du modèle après l'entraînement.
- Ces données doivent être représentatives des conditions réelles rencontrées par le système lors de son déploiement dans un environnement hospitalier.
- Données Supplémentaires :
  - Des données supplémentaires telles que des annotations de contexte (par exemple, des informations sur l'emplacement dans l'hôpital, l'heure de la journée, etc.) peuvent également être utiles pour améliorer la capacité du modèle à interpréter les commandes vocales dans des contextes spécifiques.
- Données de Reconnaissance Vocale :
  - Format : Fichiers audio (par exemple, WAV, MP3).
  - Exemple : Enregistrements audio des commandes vocales prononcées dans des contextes hospitaliers.
  - Stockage : Ces fichiers audio peuvent être stockés dans un système de fichiers ou une base de données spécialisée pour la gestion des données audio.
- Données Textuelles :
  - Format : Fichiers texte (par exemple, TXT, CSV).
  - Exemple : Transcriptions des enregistrements audio obtenues à l'aide de la reconnaissance vocale.
  - Stockage : Les transcriptions textuelles peuvent être stockées dans des fichiers texte ou des bases de données relationnelles, associées aux enregistrements audio correspondants.
- Données d'Annotation :
  - Format : Fichiers texte (par exemple, CSV) ou bases de données.
  - Exemple : Annotations manuelles des transcriptions pour identifier les commandes vocales pertinentes et les intentions de l'utilisateur.
  - Stockage : Les annotations peuvent être stockées dans des fichiers CSV avec des informations sur les segments de texte annotés ou dans des bases de données avec une structure adaptée.
- Données de Validation :
  - Format : Fichiers audio et texte.
  - Exemple : Enregistrements audio et transcriptions correspondantes pour évaluer la performance du modèle.
  - Stockage : Ces données peuvent être stockées dans des répertoires séparés ou dans une base de données, organisées pour faciliter l'évaluation du modèle.
- Données Supplémentaires :
  - Format : Dépend du type de données supplémentaires (annotations, métadonnées, etc.).
  - Exemple : Annotations de contexte telles que l'emplacement dans l'hôpital, l'heure de la journée.
  - Stockage : Ces données peuvent être stockées dans des fichiers structurés ou des bases de données en fonction de leur nature et de leur volume.



Pipeline de Traitement des Données :

1. Acquisition des Données : Collecte des enregistrements audio et de leurs transcriptions correspondantes dans différents contextes hospitaliers.
2. Prétraitement des Données : Élimination du bruit de fond, normalisation des données audio, prétraitement des transcriptions (nettoyage, tokenization, etc.).
3. Annotation des Données : Annotation manuelle des transcriptions pour identifier les commandes vocales pertinentes et les intentions de l'utilisateur.
4. Division des Données : Séparation des données en ensembles d'entraînement et de validation pour l'entraînement et l'évaluation du modèle.
5. Stockage des Données : Stockage organisé des données dans des formats adaptés pour une utilisation ultérieure lors de l'entraînement du modèle.
6. Intégration des Données : Fusion des données de reconnaissance vocale, des transcriptions, des annotations et des données supplémentaires pour former un ensemble de données complet et diversifié.

En résumé, pour entraîner le modèle final, nous aurons besoin d'un ensemble de données diversifié et étiqueté, comprenant à la fois des enregistrements audio et leurs transcriptions textuelles, ainsi que des annotations pour guider l'apprentissage et l'évaluation du modèle.

## Partie 2: Développement, NLP/Prompt Engineering

### FICHER DE TEST

#### Logiciel Speech to Text

Lors des premières étapes de développement de notre système de reconnaissance vocale, nous nous concentrons sur la transformation de la parole en texte, une phase essentielle pour comprendre les commandes vocales de l'utilisateur. En utilisant des techniques avancées de traitement du signal audio et de modélisation du langage, notre objectif est de garantir une conversion précise et rapide de la parole en texte, même dans des environnements bruyants et complexes.

Voir fichier "*transcription\_vocale*"

#### Logiciel Text to command

Une fois que nous avons réussi à transcrire avec précision la parole en texte, nous passons à la phase suivante : la conversion du texte en commandes exécutables par le système. Cette étape implique l'analyse syntaxique et sémantique du texte pour identifier les intentions de l'utilisateur et générer des instructions appropriées. Notre objectif est de permettre aux utilisateurs de contrôler intuitivement le système en utilisant des commandes textuelles simples et naturelles.

Voir fichier "*analyse\_commande*"

#### Speech Synthesis from Text

Après avoir réussi à transformer le texte en commandes, nous mettons en place la synthèse de parole à partir de ce texte. Cette fonctionnalité permet au système de répondre verbalement aux utilisateurs en utilisant une synthèse textuelle. L'objectif est de fournir des réponses adaptées aux commandes de l'utilisateur en utilisant un texte préalablement généré.

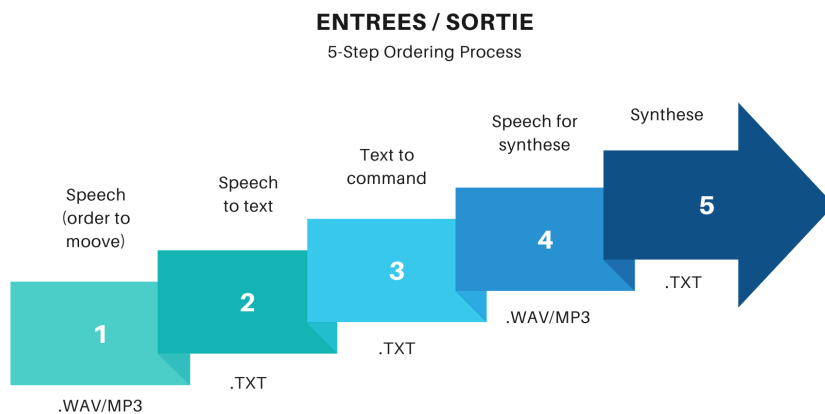
Voir fichier "*controle\_moteurs*"

## Report synthese

Enfin, nous développons une fonctionnalité de synthèse pour générer des rapports textuels ou des résumés des interactions avec le système. Par exemple, après avoir reçu et traité les informations médicales d'un patient, le système peut synthétiser un rapport de diagnostic clair et concis pour le personnel médical. Cette synthèse résume les données pertinentes et permet une communication efficace et rapide. Pour synthétiser le texte du patient nous utilisons l'API de openai GTP 3.5 Turbo avec un prompt qui nous permet d'obtenir des synthèses les plus claires possible et compréhensible pour le médecin qui lira cette synthèse.

Voir fichier "*synthese\_vocale*"

Vous pouvez vous référer à l'architecture de ce schéma :



Ajouter les PINS 13 / 15 / 16 18 EN HIGH