

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière Automatique  
Spécialité Automatique et Informatique Industrielle

réalisé par

Alem Abdelbasset

&

Rahali Hichem

---

# Implémentation des modules de marche d'un robot bipède NAO sur V-rep

---

Proposé par : Dr.Benslama Zoubir & Krabi Abdellah

Année Universitaire 2018-2019

# *Remerciements*

*Au terme de ce modeste travail, nous tenons à remercier et glorifier en premier lieu, notre Allah le tout puissant et miséricordieuses pour nous avoir donné la patience et la volonté durant le cursus universitaire.*

*Nos vifs remerciements vont également à notre promoteur : Dr Z.BENSELAMA, Qui nous a encadré d'une manière exemplaire, pour son aide, sa patience, son appui et ses conseils tout à la file de cette tâche aussi bien que Mr A.KRABI , qu'il soit assuré de notre reconnaissance et de nos sentiments les plus respectueux.*

*Nos remerciements vont également à tous les enseignants du département de Génie Electrique.*

*Enfin, nous remercions tous ceux qui nous ont apporté de l'aide dans la réalisation de ce modeste travail.*

## *Dédicace*

Je dédie ce travail à mon encadreur Dr Z.Benselama qui m'a orienté

durant ce travail et qui a offert son temps pour moi,

aussi je dédie ce travail qui n'aura jamais pu voir le jour sans les soutiens indéfectibles de mes chers parents qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui, Que dieux vous protège et que la réussite soit toujours à ma portée pour que je puisse vous combler de bonheur.

Je dédie aussi ce travail à ma grand-mère, mon frère, mes sœurs aussi bien mes oncles, mes tantes et tous mes cousins et cousines.

Tous mes amis, mes collègues et tous ceux qui m'estiment.

*Abdelbasset*

# **Dédicaces**

*Je dédie ce modeste travail à :*

*Mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.*

*A mes chers frères, pour leur appui et leurs encouragements.*

*A mes chères sœurs, pour leurs encouragements permanents et leur soutien moral.*

*A mon adorable frère Mohamed, merci d'être mon frère, mon ami, mon confident.*

*A tous ceux qui m'aiment,*

*A tous ceux ce que j'aime,*

*Aux familles RAHALI & ALEM.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués et le fruit de votre soutien infaillible.*

*Merci d'être toujours là pour moi.*

**HICHEM**

# Liste des Figures

## Chapitre 1

<b>Figure 1.1</b>	historique de robot.....	5
<b>Figure 1.2</b>	Robots manipulateurs.....	6
<b>Figure 1.3</b>	Architecture d'un robot mobile.....	8
<b>Figure 1.4</b>	robots mobiles à roues.....	9
<b>Figure 1.5</b>	Robots mobiles à chenilles.....	9
<b>Figure 1.6</b>	Robots mobiles marcheurs.....	10
<b>Figure 1.7</b>	autres moyens de locomotion des robots mobiles.....	10
<b>Figure 1.8</b>	Synoptique de sécurité.....	11
<b>Figure 1.9</b>	Navigation de robot mobile en environnement.....	13
<b>Figure 1.10</b>	robot manipulateur dans un environnement pharmaceutique sensible.....	14
<b>Figure 1.11</b>	robot agricole.....	15
<b>Figure 1.12</b>	Robot utilisé dans le domaine.....	15
<b>Figure 1.13</b>	robot de service ASIMO.....	16
<b>Figure 1.14</b>	la diversité des robots utilisés en robotique humanoïde.....	17
<b>Figure 1.15</b>	robot humanoïde de loisir NAO.....	19
<b>Figure 1.16</b>	robot humanoïde de service ROMEO.....	19

## Chapitre 2

<b>Figure 2.1</b>	Robot humanoïde NAO.....	21
<b>Figure 2.2</b>	Robot humanoïde NAO V 4.0 avec tous ses degrés de liberté.....	22
<b>Figure 2.3</b>	emplacement des capteurs de robot humanoïde NAO.....	23
<b>Figure 2.4</b>	NAO participant à une chorégraphie.....	23
<b>Figure 2.5</b>	NAO à la RoboCup2015.....	23
<b>Figure 2.6</b>	polygone de sustentation.....	25
<b>Figure 2.7</b>	Marche quasi-statique bipède (carré : CdM ) projeté sur le polygone de sustentation.....	26
<b>Figure 2.8</b>	la marche dynamique, position des pieds et projection de la trajectoire du CdM.....	26
<b>Figure 2.9</b>	Cycle de marche humanoïde.....	27
<b>Figure 2.10</b>	Séquence des phases dans la marche humanoïde.....	27
<b>Figure 2.11</b>	modèles concentrés en masse.....	28
<b>Figure 2.12</b>	Modèle à chariot sur table pour le NAO.....	29
<b>Figure 2.13</b>	vue schématique du modèle à chariot sur table.....	29
<b>Figure 2.14</b>	vue schématique du modèle à pendule inversé linéaire pour NAO.....	30
<b>Figure 2.15</b>	l'affectation des repères pour l'analyse géométrique de robot humanoïde.....	32
<b>Figure 2.16</b>	Chaîne série de jambe de l'humanoïde NAO.....	36

### Chapitre 3

<b>Figure 3.1</b>	illustration d'une vue typique de l'application V-rep.....	41
<b>Figure 3.2</b>	représentation du fonctionnement de chaque bouton de la barre d'outils {1}.....	42
<b>Figure 3.3</b>	représentation du fonctionnement de chaque bouton de la barre d'outils {2}....	42
<b>Figure 3.4</b>	représentation de différent élément de la hiérarchie de scènes.....	43
<b>Figure 3.5</b>	Les approches de programmation prise en charge.....	45
<b>Figure 3.6</b>	Architecture de contrôle V-rep.....	59
<b>Figure 3.7</b>	quelques exemples des objets utilisés dans V REP et leur représentation en 3D.....	51
<b>Figure 3.8</b>	L'interface graphique.....	52
<b>Figure 3.9</b>	les logiciels du NAO.....	53
<b>Figure 3.10</b>	Diagramme de flux dédié au modèle Straight forward.....	55
<b>Figure 3.11</b>	Diagramme de flux dédié au modèle Walk and Turn.....	56
<b>Figure 3.12</b>	Diagramme de flux dédié au modèle Omnidirectional walk.....	56
<b>Figure 3.13</b>	le diagramme des séquences de connexion.....	57
<b>Figure 3.14</b>	simulation de robot NAO sur V-rep.....	59
<b>Figure 3.15</b>	Organigramme des modules implémentés.....	60

## Liste des tableaux

<b>Tableaux 2.1</b>	Architecture matérielle du robot humanoïde NAO.....	<b>22</b>
<b>Tableaux 2.2</b>	les paramètres D-H de la chaîne de jambe gauche de robot NAO.....	<b>37</b>
<b>Tableaux 2.3</b>	les paramètres D-H de la chaîne de jambe droite de robot NAO.....	<b>37</b>

## Liste des abréviations

<b>R.U.R</b>	Rossum's Universal Robots
<b>A.F.N.O.R</b>	Association Française de Normalisation
<b>OT</b>	Organe Terminale
<b>FBH</b>	Full Body Human
<b>ddl</b>	degrés de liberté
<b>CdM</b>	Centre de Masse
<b>ZMP</b>	Zéro Moment Point
<b>CdP</b>	Centre de Pression
<b>MGD</b>	Modélisation Géométrique Directe
<b>MCD</b>	Modélisation cinématique Directe
<b>D-H</b>	Denavit-Hartenberg
<b>V-rep</b>	Virtual robot experimental platform
<b>API</b>	Application Programming Interface
<b>GPU</b>	Graphics Processing Unit
<b>ROS</b>	Robot Operation System
<b>3D</b>	3 Dimensions



## Table des matières

Liste des figures.....	i
Liste des tableaux.....	iii
Liste des Abréviations.....	iv
Introduction générale.....	1

### Chapitre 1 : Généralités sur la robotique

1.1	Introduction.....	3
1.2	Notions de base.....	3
1.2.1	Origine de terme.....	3
1.2.2	Définition de la robotique.....	3
1.2.3	Définition d'un robot.....	3
1.3	Historique.....	4
1.4	Types de robots.....	6
1.4.1	Les robots manipulateurs.....	6
1.4.2	Les robots mobiles.....	7
	a) Classification.....	7
	b) Les moyens de perception en robotique mobile.....	8
	c) Architectures des robots mobiles.....	8
1.5	Caractéristiques d'un robot.....	13
1.6	Domaines d'applications.....	14
1.6.1	Domaine industriel.....	14
1.6.2	Domaine agriculture.....	14
1.6.3	Domaine militaire.....	15
1.6.4	Domaine de service.....	15
1.7	Robot humanoïde.....	16
1.7.1	Définition du robot humanoïde.....	16
1.7.2	Les intérêts de la robotique humanoïde.....	17

1.7.3	Les types des robots humanoïdes.....	18
a)	Les robots humanoïdes de loisir.....	18
b)	Les robots de service de taille humain.....	19
1.8	Conclusion.....	19

## **Chapitre 2 : Modélisation de robot humanoïde**

2.1	Introduction.....	21
2.2	Présentation du robot humanoïde NAO.....	21
2.3	La marche humanoïde.....	24
2.3.1	Les indicateurs de stabilité de la marche humanoïde.....	24
a)	Le centre de masse.....	24
b)	Zéro moment point.....	24
c)	Centre de pression.....	25
d)	Polygone de sustentation.....	25
2.3.2	Les modes de marche humanoïde.....	26
2.3.3	Description et principe de la marche humanoïde.....	26
a)	Phase de simple support.....	26
b)	Phase de double support.....	26
c)	Phase d'impact.....	26
2.3.4	Modèle de marche humanoïde.....	27
2.3.5	Degrés de liberté nécessaires à la marche humanoïde.....	28
2.3.6	Contrôle de la marche à l'aide des modèles approximatifs.....	28
a)	Modèle de chariot sur table.....	28
b)	Modèle à pendule inversé linéaire.....	30
2.4	Modélisation de robot humanoïde.....	31
2.4.1	Modélisation géométrique.....	31
a)	Modélisation géométrique direct.....	32
b)	Modélisation géométrique inverse.....	35

2.4.2	Modélisation cinématique.....	35
a)	Modélisation cinématique directe.....	36
b)	Modélisation cinématique inverse.....	37
2.5	Conclusion.....	38

### **Chapitre 3 : Implémentation**

3.1	Introduction.....	39
3.2	La plateforme V-rep.....	39
3.2.1	Différents composants de V-rep.....	40
3.2.2	Mise en œuvre du V-rep.....	44
a)	Scripts intégrés.....	45
b)	Adds on.....	47
c)	Plug-ins.....	47
d)	Remote API clients.....	48
e)	ROS Nodes.....	49
f)	NoeudBlueZero.....	49
3.2.3	Les objets de scène.....	50
3.3	Choregraphe.....	51
3.4	Naoqi.....	52
3.5	Implémentation.....	54
3.5.1	Les modules de marche.....	54
a)	Straight forward.....	55
b)	Walk and Turn.....	55
c)	Omni-directional Walk.....	56
3.5.2	Etablissement de connexion.....	57
3.5.3	Simulation.....	58
3.6	Conclusion.....	60

Conclusion Générale.....	61
Bibliographie.....	62

*Introduction*  
*Générale*

## Introduction générale

---

D'un point de vue scientifique, parmi les nombreuses définitions possibles du robot, on peut retenir qu'il s'agit d'une machine dont l'état de perfection future lui permettra de rendre tous les services qu'attend un homme d'un autre homme en matière d'actions matérielles. Et ce service pourra être rendu par la machine soit de manière complètement autonome (robot de substitution à l'homme), soit en collaboration avec un homme (robot de coopération). Le chemin est encore long avant d'atteindre la perfection souhaitée, car la machine doit associer des propriétés physiques complexes avec des propriétés intellectuelles non moins compliquées. Cependant, les verrous scientifiques et technologiques sautent les uns après les autres et on peut être très optimiste sur l'aboutissement de cette recherche.

S'il est une chose claire en robotique scientifique, c'est la suivante : on cherche avec une machine, si possible automatique, à réaliser des tâches qui, jusqu'ici, étaient exécutées par des hommes, soit à mains nues, soit en s'aidant d'outils manuels simples et passifs. Cette substitution à l'homme ne saurait se faire sans rappel des fonctions et caractéristiques de l'homme. Le développement de la robotique se trouve donc entraîné, peut-être par manque d'imagination, sur un terrain qui croise des données de l'anthropomorphisme dont celles ayant trait au fait que l'homme est un être vivant. Un pas décisif dans cette direction est initié par l'évolution des tâches qu'on souhaite confier au robot. On est passé de tâches industrielles à poste fixe, relativement répétitives et programmables à l'avance, à des missions où l'adaptation permanente et immédiate du robot aux nouvelles données d'un environnement relativement inconnu à l'avance exige des capacités physiques, sensorielles et mentales de plus en plus proches de celles de l'homme. Par ailleurs, une autre ambition récente de la robotique est l'intégration des robots dans un environnement humain. On constate alors qu'agir dans cet environnement est plus aisé si le mode de locomotion du robot offre les mêmes performances que celle de l'homme, en volume occupé, en agilité et en capacité de franchissement d'obstacles divers [1].

La recherche en robotique humanoïde ne cesse d'évoluer ces dernières années. De nombreux prototypes sont réalisés de par le monde et les tâches effectuées par ceux-ci sont de plus en plus complexes. Concernant la locomotion bipède en robotique, les dernières études se concentrent sur la commande et la stabilisation des robots, la gestion de l'économie d'énergie et l'innovation conceptuelle des corps de ces prototypes. Un seul but ressort de toutes ces études, c'est de tenter de reproduire la marche humaine le plus fidèlement. Il n'y a certes pas que la locomotion bipède humaine (que l'on transforme parfois en locomotion quadrupède quand on est à quatre pattes ou en locomotion serpentiforme quand on rampe...) qui puisse satisfaire les contraintes environnementales de la société humaine. Mais on ne peut nier l'intérêt du mode de locomotion de l'homme si on peut l'imiter correctement.

L'objectif principal de ce projet de fin étude est d'implémenter les différents modules de marche de robot bipède NAO sur la plateforme V-rep (Virtual robot experimentation platform).

Pour atteindre cet objectif, on va essayer d'utiliser Framework API V-rep afin de contrôler et simuler la marche de robot bipède NAO.

Ce travail comporte trois chapitres : Le premier chapitre représentera des généralités sur les robots, leurs domaines d'utilisation et leurs différents types tout en détaillant les robots mobiles.

Dans le deuxième chapitre, nous donnerons une description technique du robot humanoïde NAO. Ainsi, le concept de la marche robotique et l'état de l'art sur la modélisation des robots humanoïde.

Le troisième chapitre, sera consacré pour la plateforme virtuelle V-rep et la mise en œuvre de cette dernière. Nous présenterons aussi la simulation de nos modules implémentés.

# *Chapitre I*



#### 1.1 Introduction

Dans les années 1970, quand les chercheurs ont commencé à réfléchir sur la conception de robots, on appelait « robotique » la science des robots et/ou l'art de concevoir et fabriquer des robots. Les succès des robots industriels, qui travaillaient initialement en poste isolé et qui ont été intégrés à des chaînes de production comme des machines parmi d'autres, ont conduit le public à élargir le sens du terme « robotique » et à le rendre désormais presque synonyme d'automatisation. Le correspondant de robotique devrait naturellement être « automatique » mais ce n'est pas le cas dans le langage courant. Pour compenser cette lacune, ils se sont créés le mot « robotisation » et le verbe « robotiser » pour désigner la plupart des automatisations et leur mise en œuvre, même en l'absence de véritables robots. Dans certains champs d'applications, des mots calqués sur robotique sont apparus. Il en est ainsi de « domotique » qui désigne tout ce qui concerne l'automatisation dans l'habitat ou « productique » pour tout ce qui a trait aux moyens de production [1].

On va entamer ce chapitre par un aperçu historique sur les robots, leur classification, en détaillant les robots mobiles et leurs types. Enfin, les robots humanoïdes seront l'objet de notre travail.

#### 1.2 Notions de base

Dans cette partie nous allons définir quelque notion concernons notre sujet, on peut citer les suivants :

##### 1.2.1 Origine de terme

Robot a été utilisé pour la première fois en 1921 par Karel Capek dans sa pièce R.U.R. Il provient du terme tchèque "robota" qui signifie corvée, travail obligatoire. Le terme robotique a été employé pour la première fois par Asimov en 1941 [2].

##### 1.2.2 Définition de la robotique

La robotique peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles [3].

##### 1.2.3 Définition d'un robot

Selon l'institut américain du robot, un robot est un opérateur reprogrammable et multifonctionnel, conçu pour mouvoir des pièces, des outils ou appareils spécialisés, suivant la séquence de mouvement variable et programmés en vue de l'exécution d'une variété de tâches [4].

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des

dispositifs spécialisés, pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel.

Dans cette définition, on retrouve les différentes composantes d'une cellule robotisée :

- **Le mécanisme** : ayant une structure plus ou moins proche de celle du bras humain, il permet de remplacer ou de prolonger son action. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.
- **La perception** : qui permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des capteurs dits proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, mesure de distance, vision artificielle).
- **La commande**: qui synthétise les consignes des asservissements pilotant les actionneurs. A partir de la fonction de perception et des ordres de l'utilisateur, elle permet d'engendrer les actions de robot.
- **L'interface homme-machine** : à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter.
- **Le poste de travail et les dispositifs** : qui constituent l'environnement dans lequel évolue le robot [3].

### 1.3 Historique

Le philosophe grec Aristote a fait cette célèbre citation:

*«Si chaque outil, quand il était commandé, ou même de son propre accord, pouvait faire le travail qui lui convenait ... alors il n'y aurait plus besoin d'apprentis pour les maîtres ouvriers ni d'esclaves pour les seigneurs».*

Dans cette chronologie, nous présentons un certain nombre d'événements historiques du passé, ainsi que des développements récents qui ont contribué à façonner le monde des robots.

**1941:**Le légendaire écrivain de science-fiction Isaac Asimov écrit la nouvelle «Liar!» Dans laquelle il décrit les Trois lois de la robotique :

1. Un robot ne peut pas blesser un être humain.
2. Un robot doit obéir aux ordres donnés par des êtres humains, sauf si de tels ordres sont contraires à la Première Loi.
3. Un robot doit protéger sa propre existence tant que cette protection n'entre pas en conflit avec la première ou la deuxième loi.

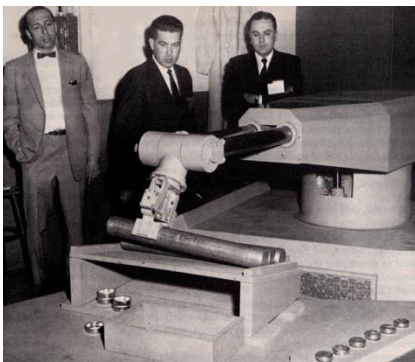
**1954:**George Devol et Joe Engleberger conçoivent le premier «bras» de robot programmable. Ce dernier est devenu plus tard le premier robot industriel chez General Motors (1962).

**1994 :**Le robot marcheur à huit jambes des universités Carnegie, Dante II, descend avec succès dans Mt Spur pour collecter des échantillons de gaz volcanique.

**1999 :** Sony lance la première version d'AIBO, un chien robotique capable d'apprendre, de divertir et de communiquer avec son propriétaire.

**2004 :** Epsom lance le plus petit robot connu, mesurant 7 cm de haut et pesant 10 grammes. L'hélicoptère robot est conçu pour être utilisé comme une «caméra volante» lors de catastrophes naturelles.

**2008 :**Après avoir été introduit pour la première fois en 2002, le populaire aspirateur robotique ROOMBA a été vendu à plus de 2,5 millions d'unités, ce qui prouve qu'il existe une forte demande pour ce type de technologie robotique domestique [5].



a) le premier «bras» de robot de George Devol & Joe Engleberger  
1954



b) Sony AIBO 2004



c) ROOMBA 2008

**Figure 1.1:** historique des robots

#### 1.4 Types de robots

Ils peuvent être divisés en robots fixes, c'est-à-dire manipulateurs, et robots mobiles.

##### 1.4.1 Les robots manipulateurs

Un robot manipulateur, quelle que soit la fonction qui lui est attribuée (transfert d'objet, soudage, assemblage), est un mécanisme capable de déplacer et de situer un objet appelé « organe terminal » ou « point outil » dans une partie de l'espace appelé « volume de travail », on peut classer les robots fixes selon 2 groupes principaux [6]:

- ✓ **les robots sériels** : La plupart des robots industriels construits à ce jour est de type sériel, c'est-à-dire que leur structure mobile est une seule chaîne ouverte formée d'une succession de segments reliés entre eux par des liaisons à un degré de liberté. Chaque articulation est commandée par un actionneur situé à l'endroit de l'articulation ou sur un des segments précédents, la figure a).1.2 montre ce type de robots.
- ✓ **les robots parallèles** : caractérisés par le fait que plus d'une chaîne cinématique assure la liaison entre la base fixe et l'organe terminal mobile, on parle également de cinématique bouclée, ce type de robots est illustré par la figure b).1.2.

Certains robots peuvent être une combinaison de cinématiques sérielles et parallèles; on parle dans ce cas de robots hybrides (voir figure c).1.2). Vu les caractéristiques de ces cinématiques, nous classerons pour simplifier les robots hybrides avec les robots parallèles [7].



a) Robot sériel.



b) Robot parallèle.



c) Robot hybride.

Figure 1.2 : Robots manipulateurs.

#### 1.4.2 Les robots mobiles

Contrairement au robot industriel qui est généralement fixé, le robot mobile est doté de moyens qui lui permettent de se déplacer dans son espace de travail. Suivant son degré d'autonomie ou degré d'intelligence, il peut s'adapter et de prendre des décisions afin de réaliser une tâche malgré un manque d'information préliminaires ou éventuellement erronées. Certains sont capables de modéliser leur espace de travail et de planifier un chemin dans un environnement qu'ils ne connaissent pas forcément d'avance [8].

L'aspect particulier de la mobilité impose une complexité technologique et méthodologique qui s'ajoute en général aux problèmes rencontrés par les robots manipulateurs. La résolution de ces problèmes passe par l'emploi de toutes les ressources disponibles tant au niveau technologique (capteurs, motricité, énergie) qu'à celui du traitement des informations par l'utilisation des techniques de l'intelligence artificielle [9].

Les possibilités sont plus vastes, du fait de leur mobilité, ils peuvent être utilisés dans des zones dangereuses (nucléaire, incendie, sécurité civile, déminage), inaccessibles (océanographie, spatial).

Les principaux problèmes particuliers liés à la conception de tels robots sont :

1. La conception mécanique liée à la mobilité
2. La détermination de la position et de l'orientation
3. La détermination du chemin optimal.

##### a) Classification

La classification des robots mobiles se fait suivant plusieurs critères (degré d'autonomie, système de locomotion, énergie utilisée, etc.) [10]. La classification la plus intéressante, et la plus utilisée est selon leur degré d'autonomie, on peut citer quelques type :

- Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire à réaliser.
- Véhicule télécommandé au sens de la tâche à réaliser. Le véhicule contrôle automatiquement ses actions.
- Véhicule semi-autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies.
- Véhicule autonome qui réalise des tâches semi- définies.

**b) Les moyens de perception en robotique mobile**

La perception est un domaine crucial de la robotique. C'est autour de ce concept qui est bâtie la structure d'un robot apte à exécuter des tâches complexes ou à évoluer dans un univers inconnu ou mal connu. L'élément de base du système de perception est le capteur qui a pour objet de traduire en une information exploitable des données représentant des caractéristiques de l'environnement. Les moyens utilisés pour la perception de l'environnement sont nombreux. Nous citons :

- Les systèmes de vision globale.
- Les télémètres laser et ultrasonores.
- Les capteurs optiques et infrarouges.
- Les capteurs tactiles.

**c) Architectures des robots mobiles**

L'architecture des robots mobiles se structure en quatre éléments :

- ✓ La structure mécanique et la motricité.
- ✓ Les organes de sécurité.
- ✓ Le système de traitement des informations et gestion des tâches.
- ✓ Le système de localisation [11].

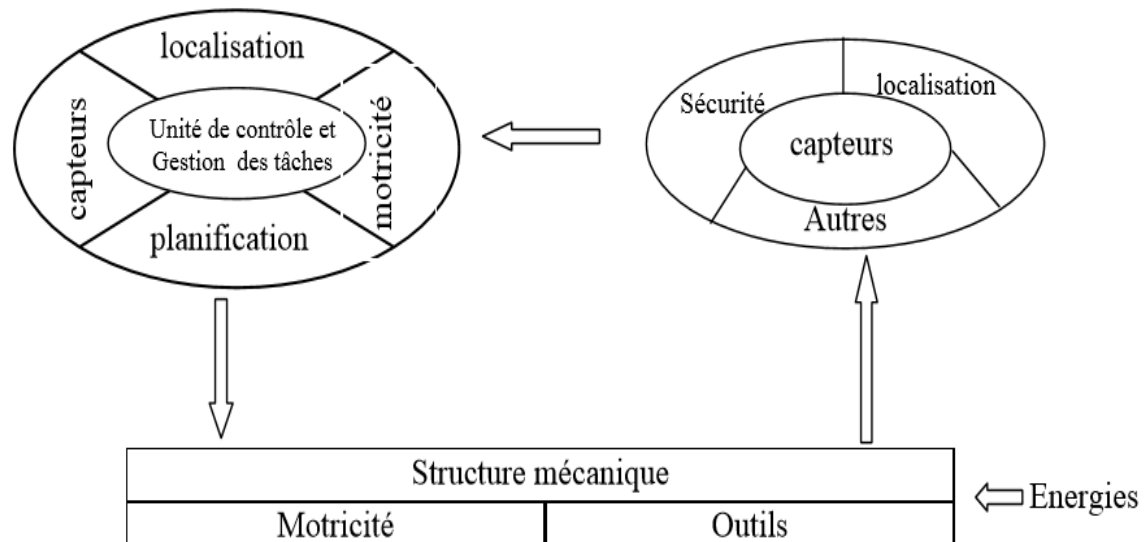


Figure 1.3: Architecture d'un robot mobile

#### ✓ La structure mécanique et la motricité

On peut dénombrer trois types de structures mécaniques assurant la motricité [10].

- **Les robots mobiles à roues**

La mobilité par roues est la structure mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importante. Le franchissement d'obstacles ou l'escalade de marches d'escalier est possible dans une certaine mesure.



Figure 1.4: robots mobiles à roues

- **Les robots mobiles à chenilles**

Lorsque le terrain est accidenté, les roues perdent leur efficacité de locomotion. Ceci limite la capacité d'évolution du robot mobile équipé de ce type de système de locomotion. Dans ces conditions, les chenilles sont plus intéressantes, car elles présentent l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles plus importants. L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...).



Figure 1.5: Robots mobiles à chenilles

- **Les robots mobiles marcheurs**

Les robots mobiles marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile, dangereux ou impossible à l'homme. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec les robots manipulateurs. La locomotion est commandée en termes de coordonnées articulaires. Les méthodes de commande des articulations définissent le concept d'allure qui assure le déplacement stable de l'ensemble. Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle des êtres humains.

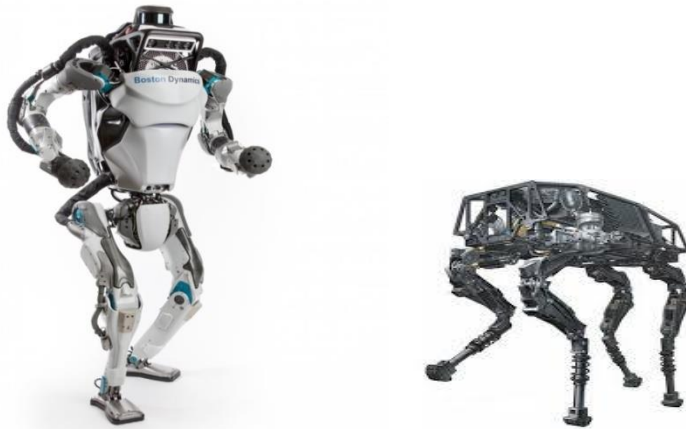


Figure 1.6: robots mobiles marcheurs

- **Autres moyens de locomotion**

Il est d'usage de mettre dans ce groupe tous les robots mobiles qui utilisent un moyen de locomotion différent des trois précédents. Par exemple, les robots mobiles qui se déplacent par reptation, les robots sous-marins, les robots volants également appelés « drones » etc. Les applications de ce type de robots sont très spécialisées et les architectures des robots sont en général spécifiques à l'application visée.



- **La motricité et l'énergie**

Figure 1.7: autres moyens de locomotion des robots mobiles



Les déplacements des robots mobiles sont réalisés par des moteurs de types électrique, thermique ou hydraulique. L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée, par contre le transport et la génération présentent des difficultés. Plusieurs méthodes sont employées :

- Par batteries qui sont soit rechargées périodiquement de manière automatique ou manuelle.
- Par groupe électrogène embarqué dont l'inconvénient constitue la masse élevée.
- Par cordon ombilical qui réduit l'autonomie du robot.

#### ✓ Les organes de sécurité

Un robot, selon la tâche qui lui est confiée, peut être amené à travailler au voisinage du personnel. A ce titre, il est obligatoire qu'il soit doté d'organe garantissant la sécurité de ce dernier. Deux types de capteurs sont disponibles tout autour du mobile afin de détecter un obstacle dans un domaine le plus étendu possible :

- **les capteurs proximétriques** : assurant la détection avant collision (ultra-son, hyper fréquence, infrarouge...).
- **les capteurs de contact** : détectant une collision ou un choc avec l'environnement (contact électrique sur pare-chocs, résistance variable...). Ce sont des dispositifs redondants par rapport aux capteurs précédents.

L'organisation de la sécurité est représentée sur le schéma de la figure 1.8 :

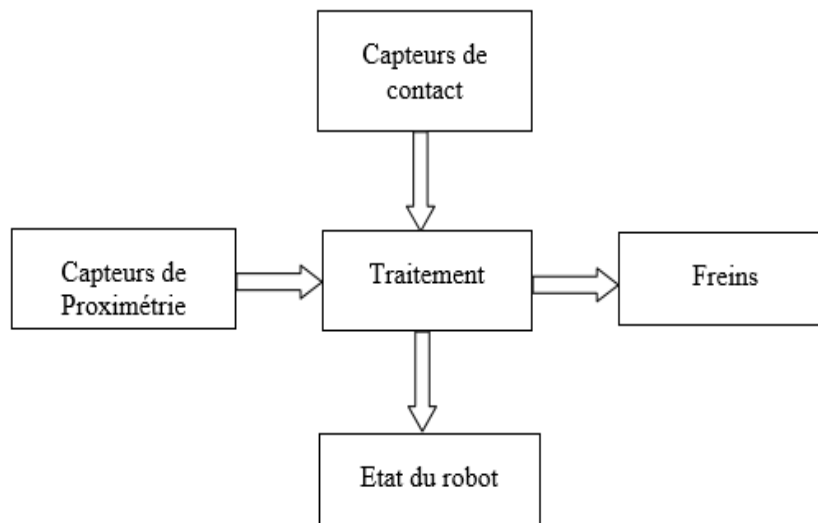


Figure 1.8:synoptique de la sécurité

#### ✓ **Traitement des informations et gestion des tâches :**

L'ensemble de traitement des informations établissent les commandes permettant au mobile de réaliser un déplacement et d'activer les divers organes en accord avec l'objectif. Le problème de génération de plan consiste à établir la manière dont le robot se déplace par rapport à des connaissances statique ou dynamique. La génération repose sur trois concepts :

- **La stratégie de navigation :** La navigation est une étape très importante en robotique mobile, elle permet de donner une large autonomie. Le système de navigation comporte plusieurs modules parmi ces modules on distingue celui de la localisation et celui de l'évitement d'obstacles. On utilise en général un capteur à ultrasons qui permet de renseigner sur la présence d'un obstacle sur le chemin d'évolution (figure 1.9).
- **La modélisation de l'espace :** La connaissance du milieu dans lequel évolue le robot mobile n'est établie en général qu'après avoir effectué une campagne de mesure de l'ensemble des éléments constituant l'environnement. Cette procédure fastidieuse peut être évitée si le robot construit lui-même son modèle d'environnement de manière dynamique.
- **La planification :** la planification de trajectoire n'est pas utilisable tant que le robot ne dispose pas d'un modèle de l'espace d'évolution ce qui handicape très fortement l'utilisation du robot. Différentes approches sont envisageables selon que le robot évolue en milieu connu ou inconnu :
  - L'évolution en territoire cartographié simplifie évidemment la tâche des concepteurs, une fois la carte de la zone d'évolution rentrée dans la mémoire d'un ordinateur communiquant avec le robot ou bien dans une mémoire intégrée au robot lui-même, des algorithmes de routage permettent de diriger le robot.
  - Il en va tout autrement dans le cas de l'évolution en territoire inconnu, Le robot doit alors analyser son environnement au moyen de différents capteurs, détecter sa position par rapport à son but, et décider de sa trajectoire [11].

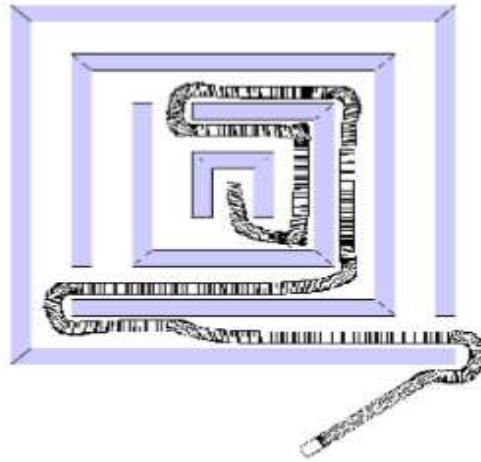


Figure 1.9 : Navigation de robot mobile en environnement

#### ✓ Le système de localisation

La localisation instantanée est un des points les plus importants et les plus délicats des robots mobiles. Les techniques employées sont de deux types :

- **La localisation relative** : La localisation relative ou localisation à l'estime consiste à évaluer la position, l'orientation, et éventuellement la vitesse du robot mobile par intégration des informations fournies par des capteurs dits proprioceptifs. L'intégration se fait par rapport au point de départ du robot. Ces données peuvent être des informations de déplacement (odomètre), de vitesse (vélocimétrie) ou d'accélération (accéléromètre).
- **La localisation absolue** : La localisation absolue est une technique qui permet à un robot de se repérer directement dans son milieu d'évolution, que ce soit en environnement extérieur (mer, terre, espace), ou en environnement intérieur (ateliers, immeubles, centrales nucléaires...). Cette méthode de localisation est basée sur l'utilisation de capteurs extéroceptifs. Elle assure la mesure de la position et de l'orientation du mobile à tout instant[12].

#### 1.5 Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre en compte :

- La charge maximale transportable (de quelques kilos à quelques tonnes).
- L'architecture du système mécanique articulé, le choix est guidé par la tâche à réaliser.
- Le volume de travail, défini comme l'ensemble des points atteignables par OT.
- Le type du robot et la mission à effectuer.
- La vitesse de déplacement.
- La masse du robot.
- Le coût du robot.

#### 1.6 Domaines d'applications

La robotique est un domaine en plein essor depuis quelques années, utilisée dans des domaines extrêmement rigoureux et exigeants. Nous allons explorer ces différents domaines.

##### 1.6.1 Domaine industriel

Depuis leur apparition sur les chaînes de production dans les années 1970, les robots industriels ont pris une place de plus en plus importante dans la plupart des secteurs où les opérations d'assemblage et les manipulations de produits sont récurrentes. Durant les trois dernières décennies, les progrès technologiques ont participé à rendre les robots plus flexibles, plus rapides et plus précis. Aujourd'hui, ils jouent un rôle prépondérant pour :

- ✓ Les manipulations de produits à haute cadence.
- ✓ Les opérations de packaging avec des manipulations de produits et d'emballage.
- ✓ L'assemblage de produits (secteurs automobile, pharmaceutique, cosmétique...etc.). ci-dessous (figure 1.10).

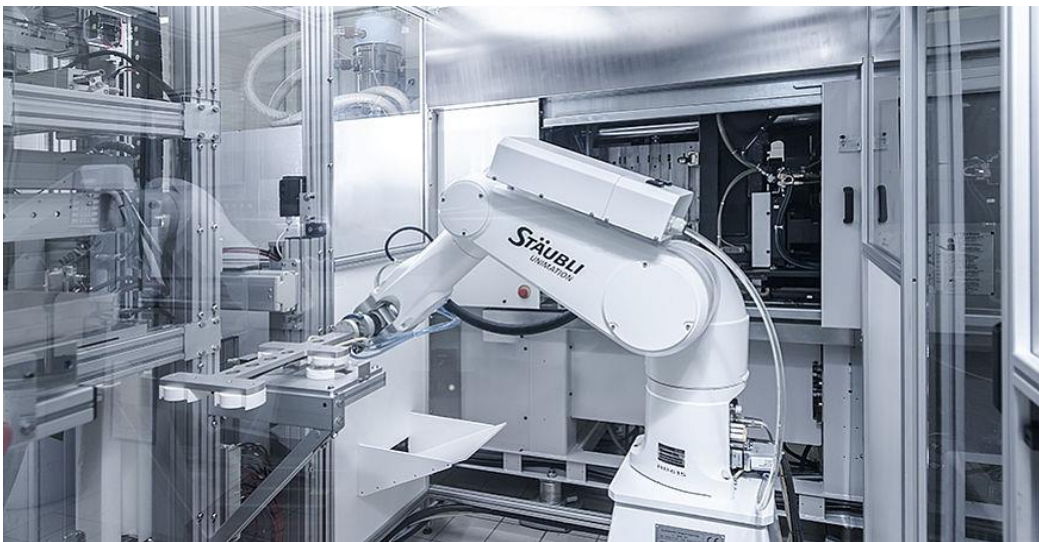


Figure 1.10: robot manipulateur dans un environnement pharmaceutique sensible.

##### 1.6.2 Domaine agriculture

Après des décennies d'expérimentation, les robots ont enfin fait leur entrée à la ferme. Cette machine totalement autonome fonctionne grâce à l'énergie solaire et circule dans les rangées de plantations pour surveiller et analyser les plantes (voir figure 1.11). Ce robot a déjà passé avec succès de nombreux tests réalisés dans des champs de légumes mais se contente de surveiller la « bonne santé » et plantations. Grâce à ces nombreux capteurs, senseurs et caméras, il détecte rapidement d'éventuelles anomalies (présence de mauvaises herbes, animaux nuisibles, croissance trop faible) et avertit l'exploitant agricole qui peut ainsi prendre immédiatement les mesures appropriées.



Figure 1.11: robot agricole.

#### 1.6.3 Domaine militaire

Les robots sont de plus en plus utilisés dans le domaine militaire. En effet, la miniaturisation permet aujourd'hui de créer des robots discrets mais dotés de nombreux capteurs, ce qui est idéal pour des missions d'espionnage ou d'éclairage. De plus, certains robots sont équipés d'un armement pour évoluer en milieu hostile, dans le but de remplacer les soldats pour limiter les pertes humaines comme le montre la figure 1.12.



Figure 1.12: Robot utilisé dans le domaine militaire

#### 1.6.4 Domaine de service

La révolution de la robotique a conduit ces dernières années à voir de nombreux robots s'installer chez les particuliers pour effectuer des tâches à la place de leur possesseur, la figure 1.13 illustre le robot de service ASIMO. En effet, ceux-ci sont capables de faire le ménage, tondre la pelouse, nettoyer la piscine etc. La robotique autrefois réservée à des applications précises ou coûteuses, est aujourd'hui de plus en plus utilisée à titre ludique [10]. En effet, les robots compagnons par exemple sont des objets de plus en plus convoités, les applications de jouet pour enfant, jusqu'à l'humanoïde destiné à remplacer une présence humaine.



Figure 1.13:robot de service ASIMO

Dans la section suivante nous allons explorer les robots humanoïdes et plus précisément le robot NAO.

### 1.7 Robot humanoïde

Les études récentes de l'impact de la robotique dans la société ont clairement établi qu'il sera nécessaire d'utiliser des robots pour effectuer dans un proche avenir des tâches de service et d'aide dans les domaines industriels, de l'inspection en milieux difficiles et dangereux, de sécurité et de surveillance et de service aux personnes âgées et/ou dépendantes. Les robots à pattes et en particulier les robots humanoïdes sont amenés à se développer très fortement dans les dix prochaines années. Leur capacité à se mouvoir dans un environnement conçu pour les êtres humains constitue leur principal atout face à d'autres structures robotiques comme celles à roues. Il est nécessaire au robot humanoïde de maîtriser les modes de déplacement de marche dans un environnement d'habitation avec des escaliers et des sols inclinés. Car ils sont de plus en plus souvent cités comme une solution potentielle pour répondre aux problèmes liés au vieillissement de la population (aides à la personne). Les chercheurs imaginent qu'un robot capable de se déplacer dans un environnement initialement conçu pour les humains et capable de réaliser des tâches de service à domicile, est une solution économiquement très intéressante pour répondre aux besoins. La mise au point d'un tel robot humanoïde répond dès lors, non seulement à ce problème d'aide et de service à la personne, mais permet également l'utilisation de ce même robot pour des applications de surveillance et d'intervention en milieux dangereux ou en situation de catastrophes [13].

#### 1.7.1 Définition du robot humanoïde

Le terme humanoïde signifie « de forme humaine », il évoque principalement la bipédie, la présence de deux bras et d'une tête. Il s'agit donc uniquement d'un critère phénotypique et plus précisément morphologique. Un robot humanoïde est un robot dont l'apparence générale rappelle celle d'un corps humain. Généralement, les robots humanoïdes ont un torse avec une tête, deux

### Généralités sur la robotique

bras et deux jambes, bien que certains modèles ne représentent qu'une partie du corps, par exemple à partir de la taille. Certains robots humanoïdes peuvent avoir un « visage », avec des « yeux » et une « bouche ».

La robotique humanoïde étudie donc les robots qui ressemblent à l'être humain. Cette ressemblance dépend des objectifs de la recherche effectuée. Par exemple les travaux qui portent sur la manipulation des objets, l'interaction homme-machine et l'intelligence artificielle utilisent souvent des robots dont seul le haut du corps ressemble à celui des humains. Le robot TETESAR le projet COG du MIT, et le WE-4RII, illustrent ce type de recherches. D'autre part, les robots bipèdes sont utilisés depuis les années 70 pour la recherche sur la locomotion des machines et des personnes. Les travaux sur les exosquelettes, les marcheurs semi-passifs et les mécanismes bipèdes capables de porter une personne sont quelques exemples de cette catégorie de robots. Récemment, les progrès technologiques ont permis l'émergence d'humanoïdes complets (FBH). La figure 1.14 illustre la diversité des robots utilisés en robotique humanoïde [14].

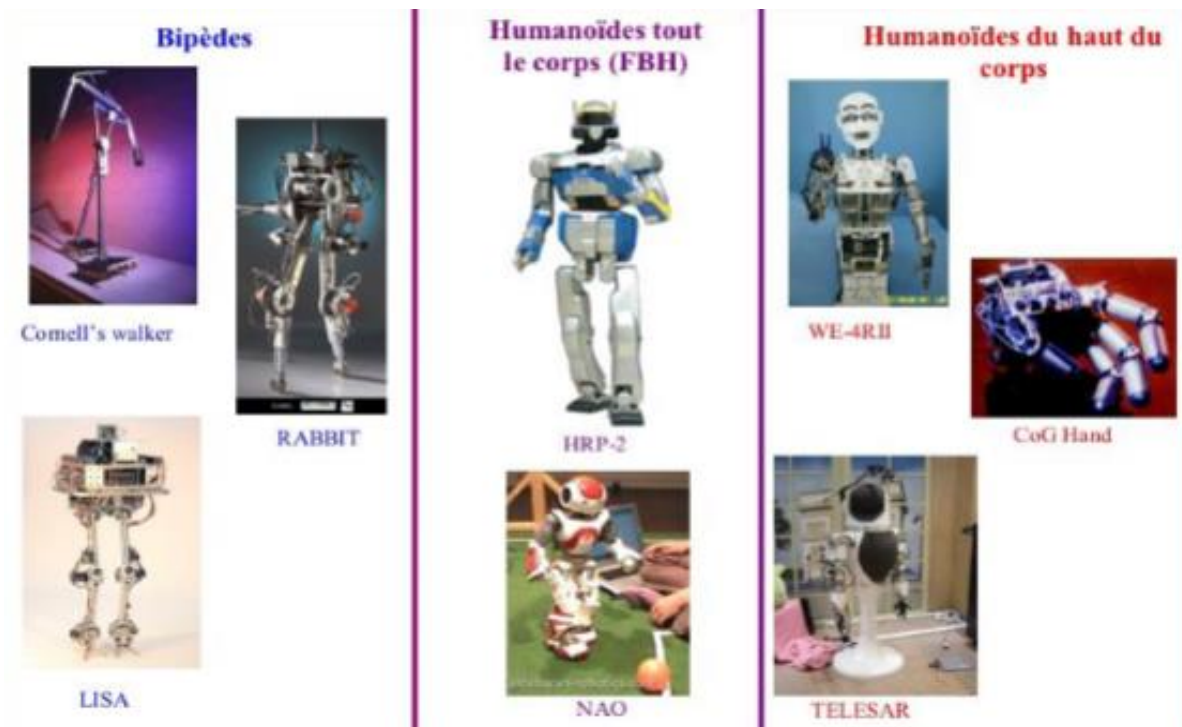


Figure 1.14: la diversité des robots utilisés en robotique humanoïde

#### 1.7.2 Les intérêts de la robotique humanoïde

Il existe trois intérêts majeurs qui motivent l'étude et le développement des robots humanoïdes :

- ✓ comprendre le mieux possible le fonctionnement des humains qui demeurent les entités dont les fonctionnements, en particulier mécaniques. La construction d'un robot avec des

capacités similaires et même supérieures sur certains aspects, demeure un des rêves de nombreux chercheurs dans ce domaine.

- ✓ élargir la présence d'une personne. De la même façon que le téléphone permet d'augmenter la portée de notre voix, nous pouvons demander à un robot humanoïde de réaliser une tâche spécifique. Amener une boisson du réfrigérateur, jeter les poubelles ou faire des courses sont quelques exemples de cette idée de présence élargie qu'un robot humanoïde peut offrir. Ceci a une application immédiate pour des tâches dangereuses ou peu appropriées aux humains [16].
- ✓ Une plate-forme de recherche d'intégration commune pour le développement scientifique et technologique. Parmi les thématiques abordées, on peut citer, la planification et la génération de mouvements, la coordination perception action/locomotion, l'optimisation mécatronique de robots complexes, l'intelligence artificielle, etc. Cette richesse due à l'interaction de chercheurs de domaines différents qui fait de la robotique humanoïde un sujet passionnant et en progrès rapide.

#### 1.7.3 Les types des robots humanoïdes

Une classification des robots humanoïdes existants est difficile à faire du fait de la diversité croissante de projets et modèles. Pourtant on peut distinguer deux types de robots qui ont des tailles et une structure différentes.

- a) **Les robots humanoïdes de loisir** : L'application principale de ces humanoïdes est d'être utilisée comme jouet ou compagnon disposant de capacités multimédia et d'accès à internet. Du point de vue de la locomotion, la marche et parfois la danse sont étudiées. Ces ordinateurs mobiles, autonomes et intelligents, représentent des marchés potentiels importants à court terme et des recherches sont menées sur la locomotion ou encore l'apprentissage. Ces robots comprennent des tailles à partir d'une quinzaine de cm jusqu'à une cinquantaine et possèdent des moteurs à chaque articulation et la majorité aux pieds plats et rigides.

Le robot NAO fait partie de cette catégorie, une brève description sera présentée. Ensuite, une autre bien détaillée dans le chapitre 2.

**NAO** : un robot de 58 cm de haut et de 5 kg développé par la société française Aldebaran Robotics, créée en 2005 et devenue en cinq ans un des leaders mondiaux de la robotique humanoïde. Véritable combinaison de matériel et de logiciel, NAO est un robot entièrement programmable, autonome et interactif utilisant les dernières technologies innovantes. Il peut voir, entendre, parler, sentir et communiquer. Présent dans les centres de recherche du monde entier ainsi que dans de grandes universités, il participe à la recherche et à l'éducation. Bien qu'il ait été conçu à l'origine pour tenir compagnie à des particuliers, NAO est surtout utilisé pour la recherche en robotique (voir figure 1.15).

- b) **Les robots de service de taille humaine** : Une de principales applications de ces robots est l'assistance de personnes handicapées et âgées, ainsi que la substitution des



personnes pour des tâches dangereuses ou pénibles. Pour cela, ces robots humanoïdes doivent être capables d'évoluer et d'effectuer des tâches dans un environnement conçu pour les humains. Ils doivent aussi être performants au niveau de l'interaction avec l'homme. Le robot ROMEO est un parfait exemple de ce type de robots (voir figure 1.16).

**ROMEO**:un robot issu d'un projet visant à développer la robotique liée à l'aide et à l'assistance aux personnes âgées ou en perte d'autonomie.il est de 1m40 de long et 40kg de poids il dispose d'une quarantaine d'articulation. ROMEO est de taille parfaite pour servir dans une maison ou un appartement. Il aide aux tâches qui peuvent vous être pénible et à la cuisine.il peut également vous rappeler la liste des choses à passer prendre au supermarché, ou un rendez-vous. Il est aussi capable de détecter la fatigue et de vous suggérer un repos. Le rappel des médicaments fait aussi parti de ses attributions [15].



Figure 1.15: robot humanoïde de loisir NAO

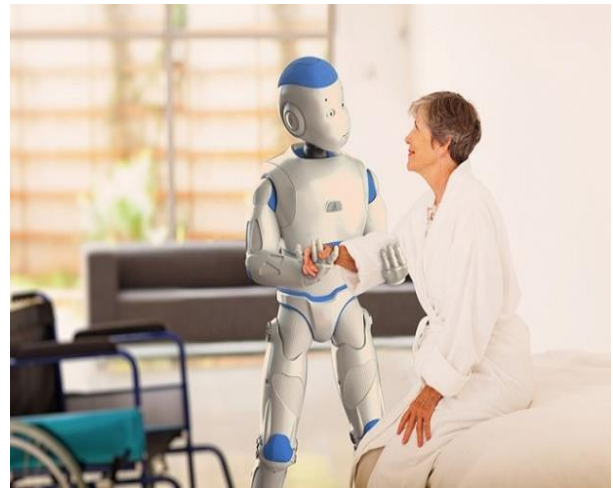


Figure 1.16:robot humanoïde de service ROMEO

### 1.8 Conclusion

La robotique est un domaine pluridisciplinaire par excellence, effectivement il implique de nombreuses thématiques telles que la mécanique, la mécatronique, l'électronique, l'automatique, l'informatique et l'intelligence artificielle. L'expansion des robots dans les différents domaines est une excellente chose, car elle permet d'accomplir des tâches dangereuses et monotones pour l'homme et leur rendement est beaucoup plus avantageux et stimulant pour l'économie.

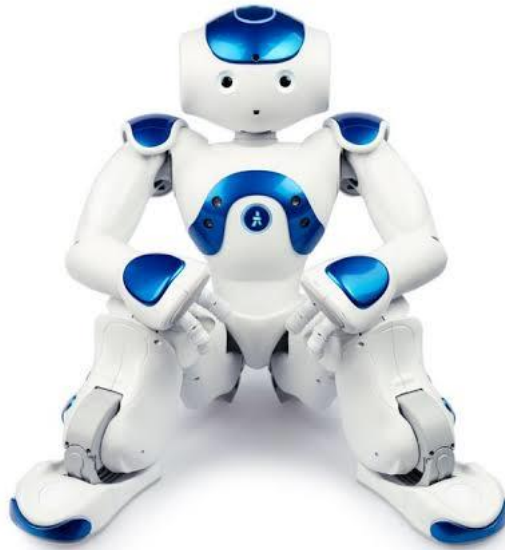
La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). Dans le chapitre suivant nous allons explorer ces modèles et donner un aperçu sur la modélisation du robot humanoïde.

# *Chapitre II*

#### 2.1 Introduction

Dans ce chapitre nous allons présenter l'étude théorique globale concernant la modélisation de robot humanoïde. Ainsi, un état de l'art sur la marche robotique sera présenté. On entamera le chapitre par une description technique du robot NAO. Puis, les concepts décrivant la marche bipède. Ensuite, avant de parler sur le modèle géométrique et cinématique, on explique les modèles de contrôle de marche.

#### 2.2 Présentation du robot humanoïde NAO



**Figure 2.1 :** Robot humanoïde NAO

Le robot NAO est l'un des célèbres robots humanoïdes disponibles pour la recherche et le développement. La société française Aldebaran l'a introduite en 2004. Il existe quatre modèles de NAO qui diffèrent par leur structure, tels que les modèles H25, H21, T21 et T25. Le modèle H est un humanoïde complet, tandis que le modèle T est constitué uniquement des parties supérieures du corps. Il existe différentes versions de chaque modèle[17]. La version à l'étude est H25 V4.0, présente un total de 25 ddl (degrés de liberté), une hauteur de 58 cm et une masse d'environ 5 kg. La structure de NAO comprend une tête à deux articulations, deux membres supérieurs et inférieurs symétriques à six et cinq articulations respectivement et un tronc avec une articulation dans le bassin. Dans chacun des bras de NAO, 2 ddl sont situés à l'épaule, 2 ddl au coude et 1 ddl au poignet, tandis qu'un ddl supplémentaire est affecté à la main pour la préhension. Chaque jambe a 2 ddl situés à la hanche, 1 ddl au genou et 2 ddl à la cheville. Contrairement aux autres robots humanoïdes, le degré de liberté du bassin est partagé entre les deux jambes via un mécanisme spécial. La figure 2.2 montre les 25 ddl montés et étiquetés sur le robot NAO [18].

Modélisation de robots Humanoïde

Articulation	Nom	Emplacement des articulations
Tête	HeadYaw	
	HeadPitch	
Bras (x2)	ShoulderPitch	
	ShoulderRoll	
	ElbowYaw	
	ElbowRoll	
	WristYaw	
Main (x2)	Hand	
Bassin	HipYawPitch	
Jambe (x2)	HipRoll	
	HipPitch	
	KneePitch	
	AnklePitch	
	AnkleRoll	

Figure 2.2 : Robot humanoïde NAO V 4.0 avec tous ses degrés de liberté étiquetés.

La plate-forme matérielle est concentrée de technologie, le tableau 2.1 montre l'architecture matérielle de NAO :

Carte mère	CPU Atom E3845 – 1,91 GHz – Cache 2MB – RAM 8 Go DDR3 – Stockage 32 Go eMMC
Tête	3 capteurs tactiles (sur le tête)
	4 microphones (plage de fréquence de 100 Hz à 10KHz / sensibilité 250mV/Pa)
	2 haut-parleurs (plage de fréquence jusqu'à 20 kHz, puissance 83 dB impédance 4 Ω entrée 2W)
	2 caméra CMOS - 5 MP – Autofocus – sortie YUY & RGB – 640x480 30fps ou 2560 x 1920 1fps
	2 x 10 Leds bleues pour les oreilles – 16 Leds blanches pour la tête – 2 x 8 Leds RGB pour les yeux
	2 moteurs coreless et 2 capteurs à effet hall pour le mouvement de la tête
Torse	2 émetteurs et 2 récepteurs à ultrasons – Fréquence 40 KHz – Plage de détection de 0,2 à 0,8 m
	1 centrale inertielle avec accéléromètre 3 axes (2g) et gyromètre 3 axes (500°/s)
	1 bouton poussoir « Marche/Arrêt » avec 1 Led RGB
	1 batterie Lithium-Ion 21,6V 2,9 Ah - Energie 62,5 Wh – Autonomie 60 à 90 min – recharge 90 min
Bras & Mains	2 x 5 moteurs coreless et 2 x 5 capteurs à effet hall pour les mouvements des bras et des mains
	2 x 1 mécanisme pour le préhension avec les trois doigts ( 2 x 1 moteurs coreless et 2 x 1 capteurs)
	2 x 3 capteurs tactiles (sur les mains)
Jambes	2 x 3 + 1 moteurs coreless et 2 x 2 x 3 + 2 capteurs à effet hall pour le mouvement des jambes
Pieds	2 x 2 moteurs coreless + 2 x 2 x 2 capteurs à effet hall pour le mouvement des pieds
	2 x 4 capteurs de pression (FSR) – plage de mesure de 0 à 25 N
	2 x 2 capteurs mécaniques (bumpers) pour la détection des chocs
Connectivité	Ethernet 1 port RJ45 - 10/100/1000 base T
	Wi-Fi - IEEE 802.11 a/b/g/n avec sécurité 64 / 128 bit : WEP, WPA / WPA2.
	1 port USB

Tableau 2.1 : Architecture matérielle du robot humanoïde NAO.

Modélisation de robots Humanoïde

La figure 2.3 montre l'emplacement de ces capteurs :

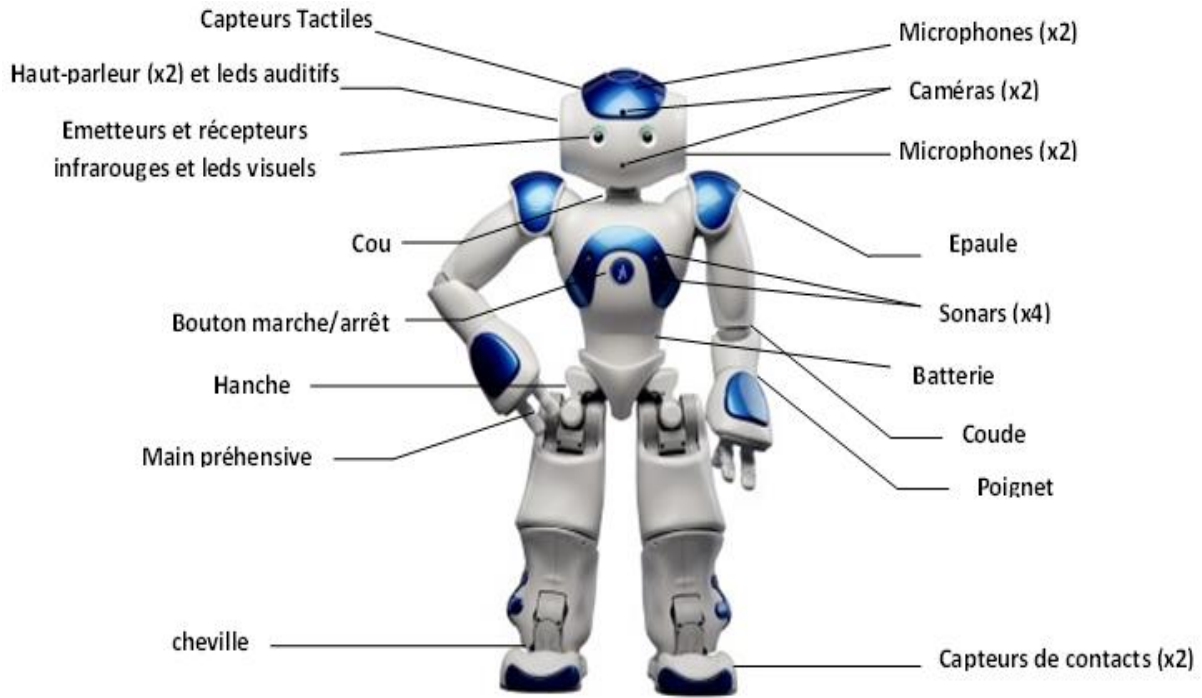


Figure 2.3 : emplacement des capteurs de robot humanoïde NAO.

NAO est un robot compagnon interactif capable d'interagir avec les humains, ce petit robot est capable d'entendre, analyser, marcher, voir, sentir et communiquer. Il peut faire la réception pour accueillir des clients où parler plusieurs langues. Il est aussi utilisé dans l'évènementiel en devenant parfois un support d'interaction. Il a également participé à un spectacle de danse avec des chorégraphes de Blanca Li en 2013 et à la RoboCup2015[19].



Figure 2.4 : NAO participant à une chorégraphie



Figure 2.5 : NAO à la RoboCup2015

## 2.3 La marche humanoïde

**Marcher** :« se déplacer par mouvements et appuis successifs des jambes et des pieds sans quitter le sol ». Cette définition issue du dictionnaire *Le Petit Robert* est très succincte et nous permet de comprendre qu'au moins un pied reste en contact avec le sol à tout moment de la marche.

### 2.3.1 Les indicateurs de stabilité de la marche humanoïde

La stabilité de la marche humanoïde peut être évaluée en mesurant la distance relative entre des points spécifiques au sol. Dans cette section, on définit ces points et explique comment il pourrait être utilisé pour évaluer la stabilité de la marche à tout moment.

#### a) Le centre de masse

La définition physique du centre de masse (CdM) pour un corps de masse distribué est le point unique où la position relative pondérée de la masse distribuée est égale à zéro. En d'autres termes, c'est un point théorique où l'on peut supposer que la masse entière d'un objet concentré. Le CdM est utile pour visualiser le mouvement d'un objet composé à liaisons multiples tel qu'un robot humanoïde. Le CdM d'un corps multi-liaisons en 3 dimensions peut être calculé à l'aide de l'équation (1) :

$$x = \sum_{i=1}^N \frac{m_i X_i}{M}, y = \sum_{i=1}^N \frac{m_i Y_i}{M}, z = \sum_{i=1}^N \frac{m_i Z_i}{M} \quad (1)$$

Où  $x$ ,  $y$  et  $z$  sont les coordonnées cartésiennes 3D de la CdM.  $X_i$ ,  $Y_i$  et  $Z_i$  sont les coordonnées cartésiennes de la CdM de la  $i^{\text{ème}}$  liaison,  $m_i$  représente la masse de la  $i^{\text{ème}}$  liaison,  $M$  est la masse totale du corps et  $N$  le nombre des liaisons du corps [20].

#### b) Zéro moment point

Le zéro moment point (ZMP) est un point sur le sol où le moment total généré en raison de la gravité et de l'inertie est égal à zéro. Le ZMP est influencé par la position et l'inertie des liaisons du robot, qui changent continuellement au cours du mouvement. Le ZMP d'un corps multi-liaisons peut être calculé à l'aide des équations (2.1) et (2.2), où  $P_x$  et  $P_y$  sont la position du ZMP dans le domaine cartésien.

$$P_x = \frac{\sum_{i=1}^n m_i (\ddot{z} + g) x_i - \sum_{i=1}^n m_i \ddot{x}_i z_i - \sum_{i=1}^n I_{iy} \ddot{\Omega}_{iy}}{\sum_{i=1}^n (\ddot{z}_i + g) m_i} \quad (2.1)$$

$$P_y = \frac{\sum_{i=1}^n m_i (\ddot{z} + g) y_i - \sum_{i=1}^n m_i \ddot{y}_i z_i - \sum_{i=1}^n I_{ix} \ddot{\Omega}_{ix}}{\sum_{i=1}^n (\ddot{z}_i + g) m_i} \quad (2.2)$$

Dans cette équation  $x_i$ ,  $y_i$  et  $z_i$  sont les sommets CdM de la  $i^{\text{ème}}$  liaison,  $m_i$  est la masse de la  $i^{\text{ème}}$  liaison,  $g$  est la constante de gravitation,  $I_{ix}$  et  $\Omega_{ix}$  sont le moment d'inertie et le déplacement angulaire autour de l'axe des X, respectivement [21].

**c) Centre de pression**

Le centre de pression (CdP) peut être défini comme un point sur le sol où la force de réaction au sol agit. En termes plus simples, le CdP est la distribution moyenne de la pression sur une surface. Si le robot est équilibré, le CdP et le ZMP indiquent le même point. La principale différence entre le CdP et le ZMP est que le CdP ne quitte jamais la zone couverte par les pieds du robot, alors que le ZMP peut quitter temporairement cette zone [22]. Le CdP d'un corps multi-bras peut être calculée à l'aide de l'équation (3).

$$OP = \frac{\sum_{i=1}^N q_i F_{ni}}{\sum_{i=1}^N F_{ni}} \quad (3)$$

Où  $OP$  est le vecteur de l'origine du système de coordonnées O à la position du centre de pression,  $q_i$  est le vecteur au point où la force  $F_{ni}$  agit perpendiculairement à la surface.

**d) Polygone de sustentation**

Le polygone de sustentation est la zone couverte par un pied simple ou la coque convexe de deux pieds en phase de simple support et double respectivement, qui seront expliqués par la suite. La stabilité dynamique et statique peut être analysée en fonction de la distance d'un point particulier à la limite de la zone de sustentation. La taille du polygone de sustentation est proportionnelle à la stabilité du robot. Si un robot avec de grands pieds est plus facile à garder en équilibre. De plus, le robot humanoïde est plus stable en phase de double support comparé au simple support car la surface du polygone est plus étendue [23]. La figure 2.6 présente un exemple de polygone de sustentation pour les phases de simple et double support.

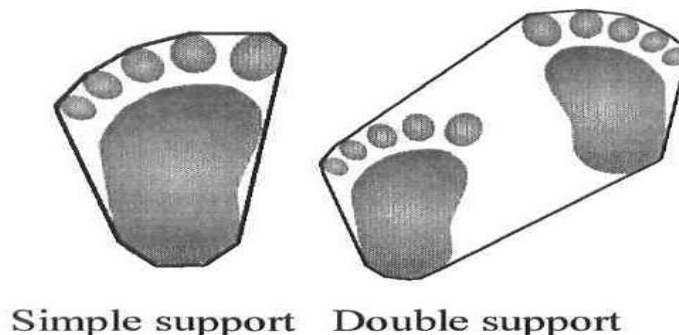


Figure 2.6 : polygone de sustentation.

### 2.3.2 Les modes de marche humanoïde

Dans le contexte de la marche robotique, on distingue deux types de marches qui sont étroitement liés à la stabilité du corps du robot pendant le déplacement. Il s'agit de la marche quasi-statique et dynamique qui seront brièvement décrites [24].

- ✓ **La marche quasi-statique:** elle est caractérisée par le fait que la projection du CdM sur le sol reste en permanence dans le polygone de sustentation. Le robot restera donc indéfiniment en équilibre s'il s'arrête brutalement. Ce type de marche est illustré sur la figure 2.7
- ✓ **La marche dynamique :** elle correspond à la marche la plus ressemblante à celle de l'humain. La projection du CdM sur le sol peut se trouver localement en dehors du polygone de sustentation comme le montre la figure 2.8



Figure 2.7 : Marche quasi-statique bipède, (carré : CdM ) projeté sur le polygone de sustentation.

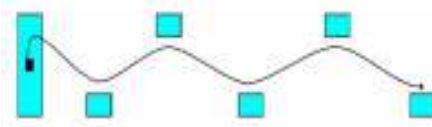


Figure 2.8 : la marche dynamique, position des pieds et projection de la trajectoire du CdM.

### 2.3.3 Description et principe de la marche humanoïde

Durant la marche, le robot bipède alterne les rôles de ses deux jambes, ce qui lui permet de se déplacer sur la surface de marche. La manière d'interaction du robot marcheur avec le sol définit les différentes phases de marche, elles peuvent être classées comme suit :

- a) **Phase de simple support :** le robot est en contact avec le sol à travers une seule jambe. Un des pieds reste en appui sur le sol d'où le nom simple support ou simple appui, alors que l'autre est en mouvement de balancement de l'arrière vers l'avant sans toucher le sol, d'où elle est appelée jambe de vol, ou jambe de balancement.
- b) **Phase de double support :** c'est la période au cours de laquelle les deux jambes sont simultanément en contact avec le sol, d'où le nom double support ou double appui. Durant cette phase le bipède fait avancer son centre de gravité et prépare les conditions initiales de la prochaine phase de simple support.
- c) **Phase d'impact :** elle caractérise le phénomène de choc, entre le pied de balancement et le sol. Elle intervient à la fin de la phase de simple support [25].



Modélisation de robots Humanoïde

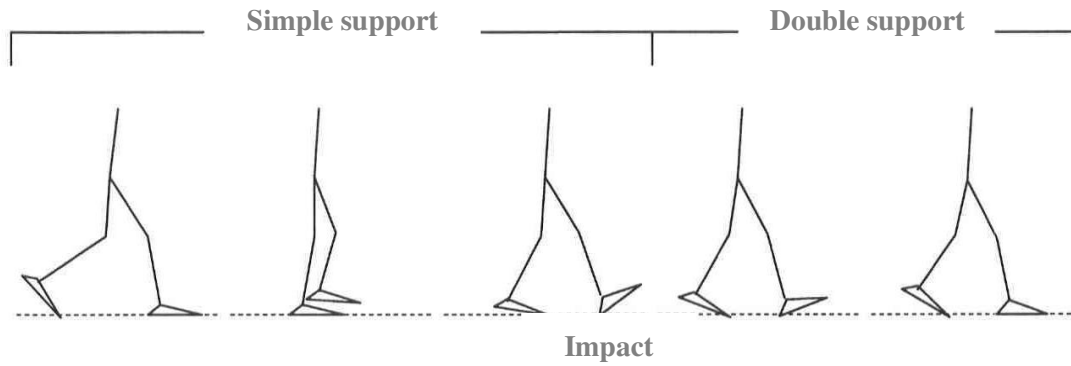


Figure 2.9: Cycle de marche humanoïde

Par conséquent un cycle de marche consiste en un enchaînement de phases de simple support, d'impact, et de double support.

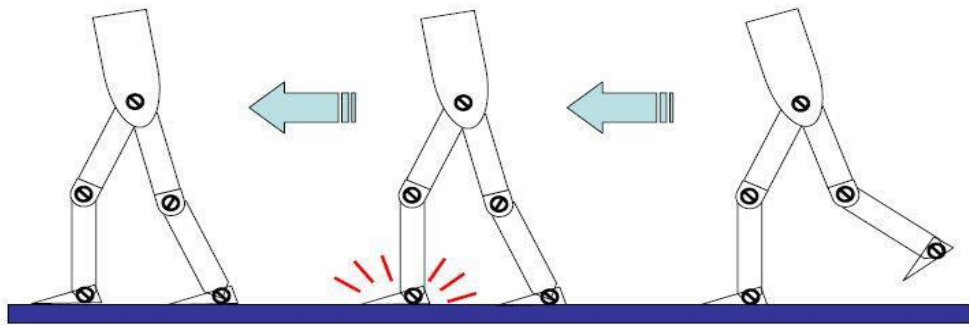


Figure 2.10 : Séquence des phases dans la marche humanoïde

2.3.4 Modèle de marche humanoïde

L'ensemble des trajectoires articulaires du robot défini sur des intervalles de temps, se succédant pour créer la marche, est appelé modèle de marche. Pour créer ce modèle, nous faisons appel à un générateur de modèle de marche. Dans une situation idéale, la marche bipède peut être réalisée en fournissant uniquement le modèle de marche au robot. Afin de réaliser une telle marche dans des conditions idéales, nous devons au préalable disposer d'un modèle précis du robot, sous forme d'un mécanisme rigide se déplaçant exactement selon la commande et sur un sol parfaitement plat et horizontal.

Dans le cas réel, un robot humanoïde peut facilement tomber à cause d'une irrégularité du sol de quelques millimètres seulement. La distribution de masse et la géométrie des structures humanoïdes créent une amplification rapide des erreurs de posture dues aux irrégularités du support. Il est nécessaire d'implémenter un programme supplémentaire pour éviter cette situation, et qui va modifier le modèle de marche en tenant compte des informations issues du gyroscope, de l'accéléromètre, des capteurs d'efforts et autres dispositifs d'information embarqués sur le robot. Ce programme est appelé stabilisateur.

### 2.3.5 Degrés de liberté nécessaires à la marche humanoïde

Une notion importante dans l'analyse d'un système mécanique est celle du degré de liberté (ddl). Les ddl d'un système sont les paramètres indépendants qui permettent de définir complètement la position et l'orientation de chacun de ses corps dans l'espace. Ainsi, pour un système à  $n$  ddl, il faudra  $n$  coordonnées pour définir sa position [26]. La marche humaine comporte naturellement des dizaines de ddl répartis dans les membres inférieurs et supérieurs du corps. Cependant, lors de l'étude de la marche bipède, le nombre de ddl de la représentation schématique du bipède doit être diminué pour en simplifier l'analyse. Plus le nombre de ddl est élevé, plus le modèle sera complexe et difficile à analyser, mais plus il aura de possibilités de mouvement.

### 2.3.6 Contrôle de la marche à l'aide des modèles approximatifs

Pour contrôler un robot humanoïde, ses mouvements doivent être générés en temps réel conformément à sa dynamique. Bien que plus précis, l'utilisation de la dynamique humanoïde réelle nécessite un coût de calcul élevé. Pour réduire ce coût, les approches courantes reposent sur des modèles simplifiés basées sur l'indicateur de stabilité ZMP. Le ZMP ne peut pas générer de trajectoires de référence pour la marche directe, mais il peut indiquer si les trajectoires de marche générées maintiendront ou non l'équilibre d'un robot. En particulier lorsque le robot se tient sur une jambe. L'idée est d'utiliser des modèles concentrés en masse, tels que le modèle à pendule inversé ou le modèle à chariot sur la table (voir figure 2.11).

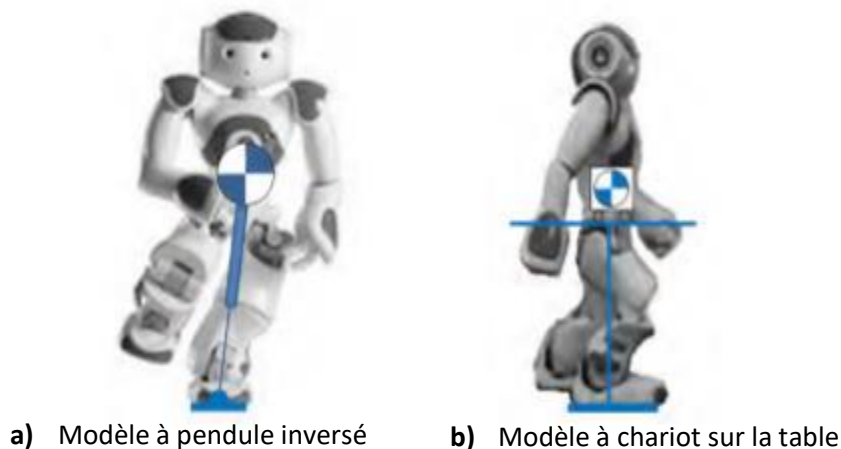


Figure 2.11 : modèles concentrés en masse

#### a) Modèle de chariot sur table

Le modèle de chariot sur table est un modèle physique qui simplifie la dynamique du robot et fournit une relation mathématique entre la position de CdM et le ZMP. Comme le montre la figure 2.12 la marche de l'humanoïde peut être représentée par un modèle à chariot sur table au cours de la phase de simple support.

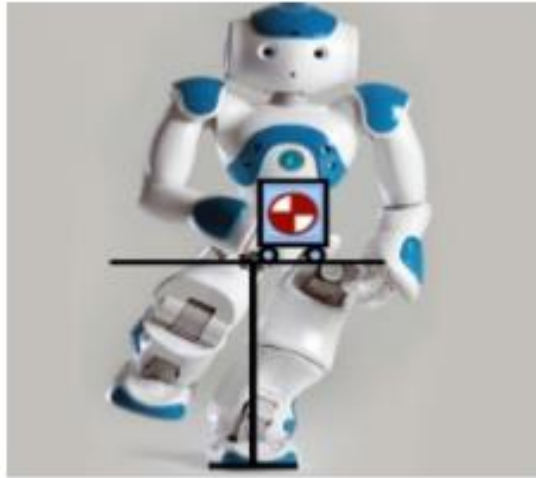


Figure 2.12 : Modèle à chariot sur table pour le NAO

Le modèle suppose que toutes les masses sont concentrées dans un chariot et que la jambe de support du robot est sans masse. Ces hypothèses sont loin de la réalité mais pourraient tout de même fournir une bonne approximation car la majeure partie de la masse du robot est concentrée dans le tronc et les jambes [23]. Les effets de poids sont faibles par rapport au tronc. Le principe du modèle est qu'il équilibre un objet en soutenant son centre de gravité et que le ZMP peut se déplacer dans le pied modelé par le contact de la table avec le sol. Pendant la locomotion, le CdM se déplace le long des deux plans. Si la position de la CdM sur un plan dépasse la zone prise en charge, alors un moment horizontal sera exercé sur l'axe et la table basculera. La figure 2.13 fournit une vue schématique du modèle.

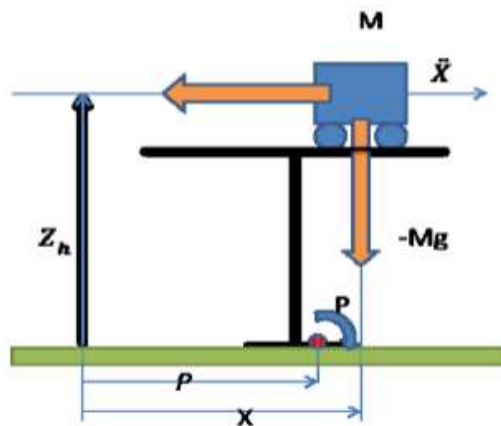


Figure 2.13 : vue schématique du modèle à chariot sur

$x$  et  $z_h$  représentent la position de CdM dans le plan  $X$  et  $Z$ .  $\ddot{x}$  et  $g$  sont l'accélération du chariot et la gravité respectivement.  $T_p$  est le moment de basculement autour de CdM et  $M$  est la masse de la CdM. Le moment autour du point  $P$  peut être calculé à l'aide de l'équation (4).

$$T_p = M_g(x - P) - M\ddot{x}z_h \quad (4)$$

Comme mentionné dans [27], le critère de stabilité dynamique exige que le moment au ZMP soit égal à zéro. En substituant  $T_p = 0$  dans l'équation (4), les équations (4.1) et (4.2) décrivent le ZMP sur les axes X et Y de manière répétitive.

$$P_x = x - \frac{z_h}{g} \ddot{x} \quad (4.1)$$

$$P_y = y - \frac{z_h}{g} \ddot{y} \quad (4.2)$$

Le chariot sur table décrit le mouvement de la CdM dans un seul plan et deux modèles à chariot sur table augmenté sont nécessaires pour décrire le mouvement du robot en trois dimensions. Le modèle de chariot sur table a pour inconvénient de ne pas envisager de changement de moment angulaire au niveau de la CdM.

**b) Modèle à pendule inversé linéaire**

La marche bipède peut être modélisée à travers le mouvement de ZMP et de CdM. Le robot est en équilibre lorsque la position du ZMP est à l'intérieur du polygone de sustentation. Lorsque le ZMP atteint le bord de ce polygone, le robot perd son équilibre. Les trajectoires de marche bipède peuvent être dérivées de la trajectoire ZMP souhaitée en calculant une trajectoire de CdM réalisable. L'oscillation possible du corps peut être estimée à l'aide de modèles dynamiques tels que le modèle à pendule inversé.

Le modèle à pendule inversé comporte certaines hypothèses et simplifications. Premièrement, il simplifie que le robot en tant qu'un point de masse unique sur son CdM. Deuxièmement, cela suppose que la jambe de force n'a aucune masse. Bien que ces hypothèses semblent être loin de la réalité, les robots humanoïdes modernes ont généralement de lourds troncs contenant des circuits électroniques et des batteries. Par conséquent, l'effet de la masse des jambes est relativement faible.

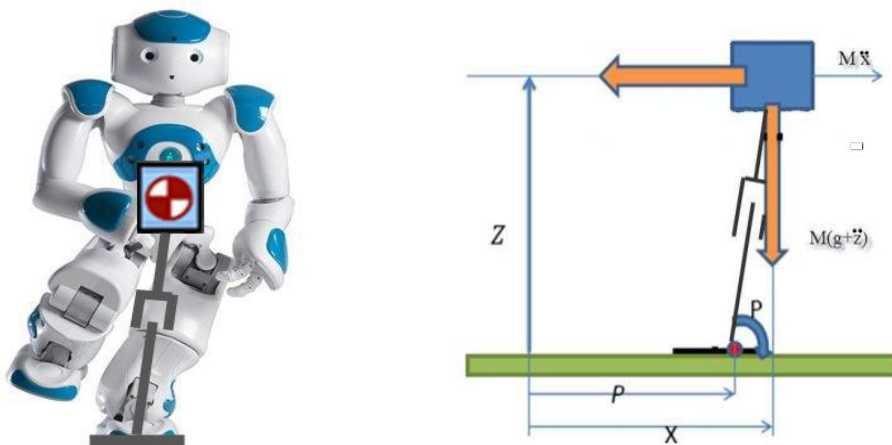


Figure 2.14 : vue schématique du modèle à pendule inversé linéaire

La figure 2.14 montre une vue schématique du modèle de pendule inversé linéaire dans le plan X et Z ou dans le plan sagittal [27]. Deux ensembles de pendule inversé sont utilisés pour modéliser une marche en 3D. L'une concerne les mouvements dans le plan frontal, une autre concerne les mouvements dans le plan sagittal.

$$T_p = Mg(x + \ddot{z} - P_x) - M\ddot{x}z(5)$$

Nous savons que lorsque le robot est équilibré dynamiquement, ZMP et CdP sont identiques, la quantité de moment dans le point CdP doit donc être nulle,  $T_p = 0$ . En supposant que le côté gauche de l'équation (5) est égal à zéro, l'équation (5.1) fournit la position du ZMP en fonction de la position et de l'accélération de CdM. Pour générer une marche en 3D, le CdM doit également se déplacer dans le plan frontal. Par conséquent, un autre pendule inversé doit être utilisé dans la direction y. En utilisant les mêmes hypothèses, l'équation (5.2) est donnée pour les mouvements dans le plan frontal.

$$P_x = x - \frac{z}{g+\ddot{z}} \ddot{x}(5.1)$$

$$P_y = y - \frac{z}{g+\ddot{z}} \ddot{y}(5.2)$$

## 2.4 Modélisation de robot humanoïde

La modélisation de Système poly-articulé (robot) a pour but de représenter au mieux le robot dans son environnement pour ensuite lui programmer des trajectoires avec la planification de mouvement. En tant que structure mécanique, un robot comprend un ensemble de corps rigides, appelés liaisons, reliés entre eux par des articulations [28]. Les articulations sont généralement rotatives qui permettent un mouvement relatif entre deux liaisons. Ils sont souvent actionnés de sorte que le mouvement global du robot puisse être contrôlé afin d'effectuer une tâche. Une tâche robotique est généralement définie en fonction de l'emplacement du l'organe terminal, caractérisé par une position et une orientation. Contrairement à un robot fixe, un robot humanoïde se caractérise par une base mobile et de multiples organes terminaux pouvant interagir avec son environnement. Sa locomotion bipède ne peut être réalisée que par des forces de contact quand il marche sur le sol.

La modélisation d'un robot consiste à dériver un ensemble d'expressions mathématiques décrivant les propriétés géométriques et temporelles du mouvement du robot [29]. Le modèle complet d'un mécanisme de robot est généralement divisé en 2 parties : la géométrie du robot et la cinématique du robot.

### 2.4.1 Modélisation Géométrique

La structure des robots humanoïde est décomposée selon le nombre de segments reliés entre eux par le biais des articulations, la modélisation géométrique consiste à définir les coordonnées

locales qui permettent de repérer les positions et orientations de chaque corps. Il existe deux types de modèles, modèle géométrique directe et inverse [24].

a) **Modélisation géométrique directe**

La Modélisation Géométrique Directe ou MGD formalise mathématiquement la chaîne cinématique et s'intéresse à relier l'espace articulaire associé aux différentes liaisons du robot  $q$ , à l'espace opérationnel associé à la configuration  $X$  de l'organe terminal pour une tâche robotique à réaliser [30]. Cette relation vectorielle exprime les positions et orientations des  $m$  coordonnées opérationnelles de  $X$  en fonction des  $n$  variables articulaires de  $q$  :  $X = f(q)$

avec  $f()$  une fonction vectorielle non-linéaire la plupart du temps de  $q$ . Le calcul du modèle géométrique est souvent basé sur le formalisme de Denavit-Hartenberg. La figure 2.15 illustre l'affectation des repères pour l'analyse géométrique de robot humanoïde NAO.

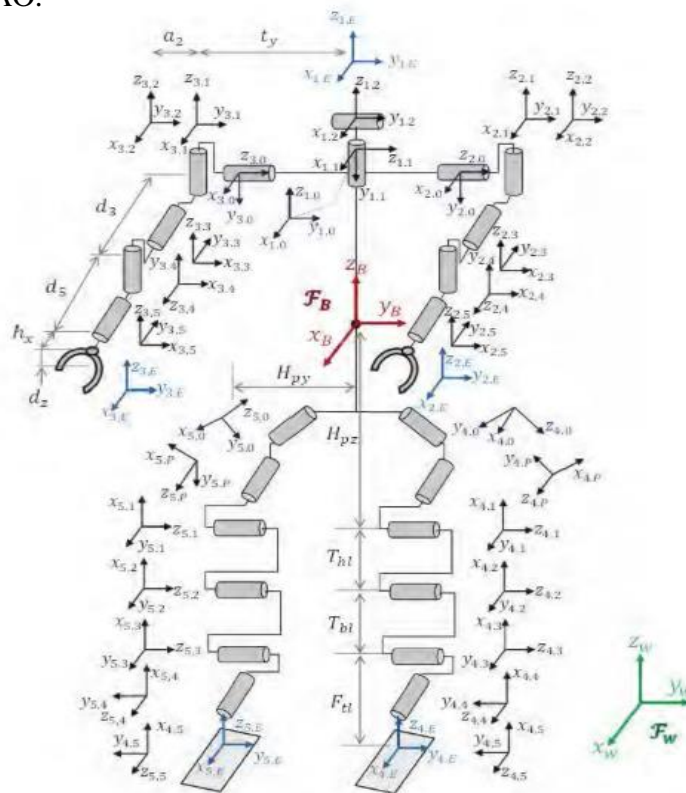


Figure 2.15 : l'affectation des repères pour l'analyse géométrique de robot humanoïde

- **Formalisme de Denavit-Hartenberg (D-H)**

Quand l'architecture de la structure mécanique est plus complexe, il faut une méthode systématique. La convention de Denavit-Hartenberg (D-H) permet de construire itérativement l'ensemble des transformations de la structure. Elle s'appuie sur l'hypothèse de liaisons glissières/prismatiques ou pivot/rotoïde entre les corps [30]. Denavit et Hartenberg ont mis au point un moyen de créer une matrice de transformation décrivant des points situés à une extrémité

d'une articulation par rapport à un système de coordonnées fixé à l'autre extrémité de l'articulation, en fonction de l'état de l'articulation. Ils ont conclu que nous pouvons décrire complètement cette matrice de transformation en utilisant seulement quatre paramètres, appelés paramètres de Denavit-Hartenberg (DH) ( $a$ ,  $\alpha$ ,  $d$  et  $\theta$ ). Avant de pouvoir expliquer ces paramètres, nous devons d'abord établir le cadre de référence de chaque joint  $i$  par rapport au cadre de référence de son joint précédent[31].

- ✓ le vecteur  $\vec{z}_i$  est aligné à la direction de l'axe de l'articulation (le sens de rotation).
- ✓ le vecteur  $\vec{x}_i$  de  $(R_i)$  est perpendiculaire à  $\vec{z}_{i-1}$  de  $(R_{i-1})$ .
- ✓ l'axe porté par  $\vec{x}_i$  intersecte l'axe porté par  $\vec{z}_{i-1}$ .

Nous pouvons maintenant décrire les paramètres DH :

- **a**: longueur de la normale commune.
- **$\alpha$** : angle autour de la normale commune, de  $\vec{z}_{i-1}$  à  $\vec{z}_i$ .
- **d**: offset le long de l'axe  $\vec{z}_{i-1}$  à la normale commune.
- **$\theta$** : angle autour de l'axe  $\vec{z}_{i-1}$ , de l'axe  $\vec{x}_{i-1}$  à l'axe  $\vec{x}_i$ .

Chaque transformation géométrique  $T_i^{i-1}$  entre les repères ortho-normés  $(R_{i-1})$  et  $(R_i)$  des corps rigides successifs  $(C_{i-1})$  et  $(C_i)$  est décomposé en quatre transformations élémentaires (RTTR).

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ces quatre transformations sont successivement :

1. une rotation d'angle  $\theta_i$  autour de  $\vec{z}_{i-1}$  (variable articulaire pour une liaison pivot).
2. une translation de longueur  $d_i$  selon  $\vec{z}_{i-1}$  (variable articulaire pour une liaison glissière/prismatique).
3. une translation de longueur  $a_i$  selon  $\vec{x}_i$ .
4. une rotation d'angle  $\alpha_i$  autour de  $\vec{x}_i$ .

- **Matrice de rotation**

La rotation dans un espace cartésien est une fonction qui fait pivoter les vecteurs d'un angle déterminé autour d'une direction spécifiée. Une rotation dans l'espace à  $n$  dimensions est décrite comme une matrice ( $n \times n$ ) orthogonale  $R$  avec le déterminant égale à 1:

$$R^T = R^{-1} \quad RR^T = R^T R = I \quad \det(R) = 1$$

Dans l'espace cartésien tridimensionnel, il existe trois matrices de rotation distinctes, chacune effectuant une rotation de  $\theta_x$ ,  $\theta_y$ ,  $\theta_z$  autour des axes x, y et z, respectivement.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \quad R_y = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad R_z = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pour faire pivoter le vecteur d'abord autour de l'axe x, puis autour de l'axe y, il faut multiplier par les matrices de rotation correspondantes dans l'ordre suivant:

$$p' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = R_y R_x p = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

- **Matrice de translation**

La translation dans un espace cartésien est une fonction qui déplace (translate) chaque point d'une distance déterminée dans une direction spécifiée. Nous pouvons décrire une translation dans l'espace tridimensionnel avec une matrice (4 × 4) de la forme suivante:

$$A = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

où  $d_x$ ,  $d_y$  et  $d_z$  définissent la distance de translation le long des axes x, y et z respectivement.

- **Matrice de transformation**

Une matrice de transformation est une matrice  $(n + 1) \times (n + 1)$ , où n est le nombre de dimensions dans l'espace sur lequel la transformation est définie. En général, une matrice de transformation est une matrice de la forme:

$$T = \begin{bmatrix} X & Y \\ [0 \dots 0] & 1 \end{bmatrix}$$

Pour les besoins de la cinématique, nous utilisons des matrices de rotation et de translation, afin de pouvoir transformer des points dans l'espace tridimensionnel. Nous considérons une matrice de transformation qui combine rotation et translation, le bloc X de la matrice définit la rotation, tandis que le bloc Y de la matrice définit la translation:

$$T = \begin{bmatrix} \hat{R} & \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \\ [0 \ 0 \ 0] & 1 \end{bmatrix}$$



La matrice résultante de la multiplication de plusieurs matrices de transformation est toujours une matrice de transformation.

$$\hat{T} = T_1 T_2 T_3 \dots T_k = \begin{bmatrix} \hat{X} & \hat{Y} \\ [0 \dots 0] & 1 \end{bmatrix}$$

Une matrice de transformation est inversible, si et seulement si X est inversible, et prend la forme:

$$T^{-1} = \begin{bmatrix} X^{-1} & -X^{-1}Y \\ [0 \dots 0] & 1 \end{bmatrix}$$

### b) Modélisation géométrique inverse

Le modèle géométrique inverse, comme son nom l'indique, est l'inverse du modèle géométrique direct. [24] Il sert à calculer les positions articulaires en fonction de la position et orientation des corps dans l'espace opérationnel, le modèle obtenu s'écrit :

$$q = f^{-1}(X)$$

### 2.4.2 Modélisation cinématique

Les robots articulés à plusieurs degrés de liberté, tels que les robots humanoïdes, sont devenus des plateformes de recherche populaires en robotique et en intelligence artificielle. De tels robots peuvent effectuer des mouvements complexes, y compris la mise en équilibre, la marche. La conception de mouvements dynamiques complexes n'est réalisable que par l'utilisation de la cinématique robotique, qui est une application de la géométrie à l'étude de chaînes robotiques arbitraires. [33] Plus spécifiquement, la cinématique robotique assure la transformation de l'espace articulaire, où les chaînes cinématiques sont définies, vers l'espace cartésien, où le manipulateur robotique se déplace, et inversement. La cinématique de robot est très utile, car elle peut être utilisée pour planifier et exécuter des mouvements, ainsi que pour calculer les forces et les couples de l'actionneur.

#### a) Modélisation cinématique direct

A partir du modèle géométrique direct  $X = f(q)$ , où  $q = (q_1, \dots, q_n)^T$ . [30] la Modélisation Cinématique Directe ou MCD décrit les m vitesses opérationnelles  $\dot{X}$ , en fonction des n vitesses articulaires  $\dot{q}$  :

$$\dot{X} = J(q)\dot{q}$$

$f$  étant une fonction vectorielle,  $f = (f_1, \dots, f_m)^T$ , chaque composante  $f_i$  peut dépendre des variables articulaires  $q_j$ . La matrice jacobienne  $J = J(q)$ , de dimensions  $(m \times n)$  est définie par l'ensemble des  $(m \times n)$  dérivées premières de  $f$  par rapport à  $q$  :

$$J(q) = \left( \frac{\partial f}{\partial q^T} \right) = \left[ \begin{array}{ccc} \frac{\partial f}{\partial q_1} & \cdots & \frac{\partial f}{\partial q_n} \end{array} \right] = \left[ \begin{array}{ccc} J_1 & \cdots & J_n \end{array} \right]$$

Comme on l'a déjà mentionné, on s'intéresse sur la partie inférieure de robot humanoïde NAO, la figure 2.16 montre la chaîne cinématique de la jambe de NAO.

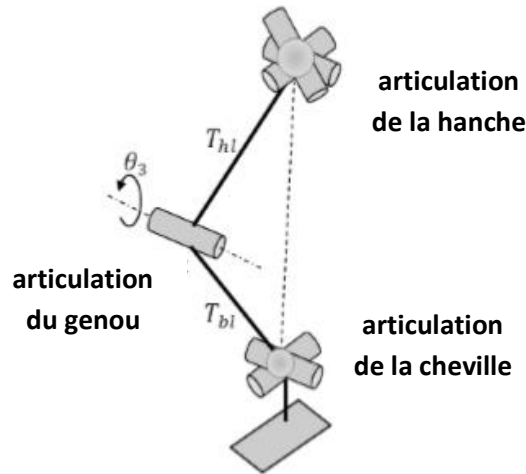


Figure 2.16 :Chaîne série de jambe de l'humanoïde NAO

- **Cinématique direct pour la jambe gauche**

Les jambes de la NAO ont chacune cinq articulations et sont reliées au repère de base (torse) par l'articulation du bassin, ce qui forme une chaîne de six articulations reliant chaque pied au repère de tors. [33] Par conséquent, à partir du repère attribué à la chaîne de la jambe gauche. Le tableau 2.2 présente les paramètres D-H pour l'ensemble de la chaîne cinématique de la jambe gauche de NAO et la matrice de transformation finale est la suivante:

$$T_{Base}^{End} = T_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z \left( -\frac{\pi}{2} \right) T_6^{End}$$

Modélisation de robots Humanoïde

Frame (Joint)	a	$\alpha$	d	$\theta$
Base	$A(0, \text{HipOffsetY}, -\text{HipOffsetZ})$			
LHipYawPitch	0	$-\frac{3\pi}{4}$	0	$\theta_1 - \frac{\pi}{2}$
LHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_2 + \frac{\pi}{4}$
LHipPitch	0	$\frac{\pi}{2}$	0	$\theta_3$
LKneePitch	$-\text{ThighLength}$	0	0	$\theta_4$
LAnklePitch	$-\text{TibiaLength}$	0	0	$\theta_5$
LAnkleRoll	0	$-\frac{\pi}{2}$	0	$\theta_6$
Rotation	$R_z(\pi)R_y(-\frac{\pi}{2})$			
End effector	$A(0, 0, -\text{FootHeight})$			

Tableau 2.2 : les paramètres D-H de la chaîne de jambe gauche de robot NAO [33]

• Cinématique direct pour la jambe droite

Les chaînes cinématiques des jambes sont totalement symétriques par rapport au plan défini par les axes X et Z. Ainsi, les différences entre les deux chaînes ne concernent que les distances le long de l'axe Y et les articulations qui tournent autour de lui. Le tableau 2.3 présente tous les paramètres D-H de la chaîne de jambe droite et de la matrice de transformation finale.

$$T_{Base}^{End} = T_{Base}^0 T_1^1 T_2^2 T_3^3 T_4^4 T_5^5 T_6^6 R_z(-\frac{\pi}{2}) T_6^{End}$$

Frame (Joint)	a	$\alpha$	d	$\theta$
Base	$A(0, -\text{HipOffsetY}, -\text{HipOffsetZ})$			
RHipYawPitch	0	$-\frac{\pi}{4}$	0	$\theta_1 - \frac{\pi}{2}$
RHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{4}$
RHipPitch	0	$\frac{\pi}{2}$	0	$\theta_3$
RKneePitch	$-\text{ThighLength}$	0	0	$\theta_4$
RAnklePitch	$-\text{TibiaLength}$	0	0	$\theta_5$
RAnkleRoll	0	$-\frac{\pi}{2}$	0	$\theta_6$
Rotation	$R_z(\pi)R_y(-\frac{\pi}{2})$			
End effector	$A(0, 0, -\text{FootHeight})$			

Tableau 2.2 : les paramètres D-H de la chaîne de jambe droite de robot NAO

b) Modélisation cinématique inverse

La cinématique inverse définit les moyens de passer de l'espace tridimensionnel à l'espace articulaire. En particulier, la cinématique inverse définit une relation entre des points de l'espace

tridimensionnel (position  $(p_x, p_y, p_z)$  et orientation  $(a_x, a_y, a_z)$  et des valeurs / angles des articulations  $(\theta_1, \theta_2, \dots, \theta_m)$  dans l'articulation d'une chaîne cinématique à m articulations. il est obtenu en inversant le modèle cinématique directe et s'écrit:

$$\dot{q} = J(q)^+ \dot{X}$$

où  $J(q)^+$  désigne la matrice pseudo-inverse de la matrice jacobienne. Le calcul du MCI peut se faire de deux façons soit analytique ou bien numérique :

- ✓ L'avantage de la solution analytique est de diminuer considérablement le temps de calcul au travers le nombre d'opérations réduit. Cependant, les singularités doivent être traitées séparément.
- ✓ La méthode numérique, quant à elle, est la plus utilisée, elle repose sur des algorithmes numériques permettant de calculer itérativement les vitesses articulaires. ces algorithmes traitent de façon unifiée les cas réguliers, redondants, et singularités. L'inconvénient de cette méthode est qu'elles nécessitent un temps de calcul relativement important. [24]

### 2.5 Conclusion

La modélisation de robot humanoïde était et reste à nos jours un modèle ambiguë. On a vu dans ce chapitre le concept de la modélisation de robot humanoïde et particulièrement les deux jambes de robot NAO, on a présenté également les modèles approximatifs utilisés pour le contrôle de la marche humanoïde. Ainsi, leurs indicateurs de stabilité.

# *Chapitre III*

### **3.1 Introduction**

Comme le matériel de robot est relativement coûteux, il est difficile pour de nombreux collègues de mettre en place un cours d'expérience sur le robot. Face à ce problème, de nombreux logiciels de simulation de robots sont développés, tels que MATLAB RoboticsToolbox, RoboAnalyzer, ARTE et d'autres. Des nombreux logiciels proposés pour l'enseignement des connaissances de base en robotique. À l'heure actuelle, la plus grande difficulté de l'enseignement de la robotique est de montrer clairement les significations de la physique mathématique de ces concepts et de combiner les connaissances théoriques et cours expérimentaux [34]. V-rep est une plate-forme de simulation virtuelle de robots développée rapidement ces dernières années. Bien qu'il n'ait pas de fonction d'enseignement directe, sa capacité d'expansion abondante nous fournit un meilleur outil.

V-rep est le résultat d'un effort visant à intégrer toutes les exigences dans un cadre de simulation polyvalent et évolutif. En plus d'offrir les approches traditionnelles que l'on retrouve également dans d'autres simulateurs, V-rep ajoute plusieurs approches supplémentaires. Il permet d'écrire des fonctions et des contrôleurs dans divers langages de programmation tels que C / C ++, Python, Java, Lua, Matlab ou Octave.

### **3.2 La plateforme V-rep**

La plate-forme d'expérimentation de robot virtuel (V-rep) est créée en 2010 et s'est développé au cours des dernières années. De nos jours, c'est le logiciel le plus utilisés dans le domaine de la robotique (avec une licence gratuite pour un but éducatif). Comme le disent ses auteurs: "Le V-rep est le couteau suisse parmi les simulateurs de robots".

V- rep est un simulateur de robot 3D basé sur une architecture de contrôle distribuée : les programmes de contrôle (ou scripts) peuvent être directement attachés aux objets de la scène et s'exécuter simultanément de manière fileté ou non fileté. Cela rend le V-REP très polyvalent et idéal pour les applications multi robots, et permet aux utilisateurs de modéliser des systèmes robotiques de la même manière qu'en réalité où le contrôle est la plupart du temps également distribué. V-rep vous permet d'éditer et de simuler des systèmes robotiques complets ou des sous-systèmes (capteurs, mécanismes, etc.). Il offre une multitude de fonctionnalités qui peuvent être facilement intégrées et combinées grâce à une API et une fonctionnalité de script exhaustive, V-rep est également une multi plate-forme qui peut être utilisée sous Windows, Linux et Mac OS. Il propose quatre moteurs de physique différents, notamment Bullet Physics, Open Dynamics, Newton et Vortex Dynamics, qui permettent un calcul dynamique rapide et personnalisable permettant de simuler des interactions réelles

## Chapitre 3

### Implémentation

entre la physique et les objets. Le modèle NAO est disponible en V-rep et il existe des programmes de transition permettant d'intégrer V-rep à NAOqi. Il présente plusieurs avantages par rapport à d'autres simulateurs, tels qu'une licence gratuite et ne nécessitant pas de carte graphique ou de GPU spécifique. Excellente solution pour :

- Prototypage et vérification rapides.
- Simulation de systèmes d'automatisation industrielle.
- Développement rapide d'algorithmes.
- Enseignement lié à la robotique.
- Surveillance à distance.
- Contrôle du matériel.
- Surveillance de la sécurité.
- Présentation du produit.

#### 3.2.1 Différents composants de V-rep

V-rep est composé de plusieurs éléments. Ses principaux éléments sont:

- **Une fenêtre consol:** au démarrage de l'application V-rep, une fenêtre de console est créée mais directement masquée à nouveau. Ce comportement par défaut de masquage de la fenêtre de la console peut être modifié dans la boîte de dialogue des paramètres utilisateur. La fenêtre de la console ou du terminal affiche quels plugins ont été chargés et si leur procédure d'initialisation a réussi. elle n'est pas interactive et sert uniquement à générer des informations. En plus, l'utilisateur peut créer par programme des fenêtres de console auxiliaires pour afficher des informations spécifiques à une simulation.
- **Une fenêtre d'application:** c'est la fenêtre principale de l'application. Elle est utilisée pour afficher, éditer, simuler et interagir avec une scène. Les boutons gauche et droit de la souris, la molette de la souris ainsi que le clavier ont des fonctions spécifiques lorsqu'ils sont activés dans la fenêtre de l'application. Dans la fenêtre de l'application, les fonctions des périphériques d'entrée (souris et clavier) peuvent varier en fonction du contexte ou de l'emplacement d'activation.
- **Plusieurs dialogues:** à côté de la fenêtre de l'application, l'utilisateur peut également éditer et interagir avec une scène en ajustant les paramètres ou les paramètres de la boîte de dialogue. Chaque boîte de dialogue regroupe un ensemble de fonctions associées, ou des fonctions s'appliquant à un même objet cible. Le contenu d'une boîte



## Chapitre 3

### Implémentation

de dialogue peut être sensible au contexte (par exemple, en fonction de l'état de sélection d'objet).

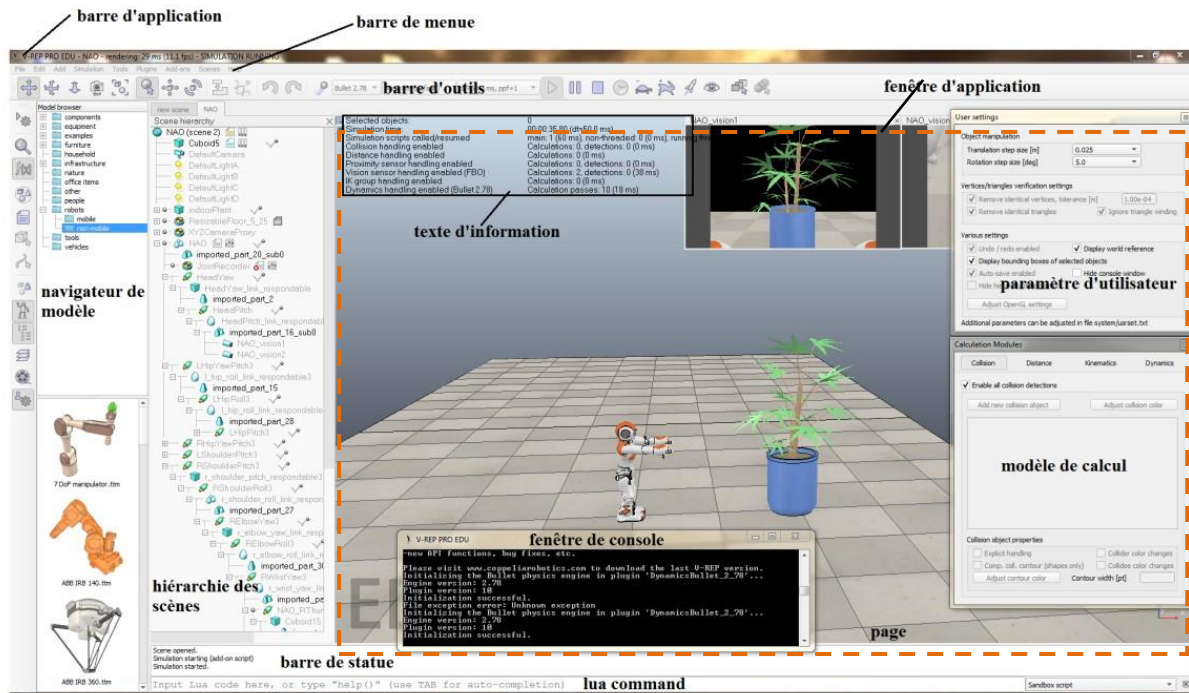


Figure 3.1 : illustration d'une vue typique de l'application V-rep

Lorsque vous lancez l'application V-rep, une scène s'initialise par défaut. L'utilisateur est libre d'ouvrir plusieurs scènes en parallèle. Chaque scène partage la fenêtre de l'application et les boîtes de dialogue avec les autres scènes, mais seul le contenu de la scène active est visible dans la fenêtre de l'application ou dans les boîtes de dialogue (une seule scène est visible à la fois).

- **Barre d'application:** elle indique le type de licence de votre copie V-rep, le nom du fichier de la scène en cours d'affichage, le temps utilisé pour une passe de rendu (une passe d'affichage) et l'état actuel du simulateur (état ou type de simulation). La barre d'application, ainsi que toute surface de la fenêtre de l'application, peuvent également être utilisées pour glisser-déposer des fichiers associés à V-rep dans la scène. Les fichiers pris en charge incluent les fichiers "\*" .tnt" (fichiers de scène V-rep), les fichiers "\*" .ttm" (fichiers de modèle V-rep) et les fichiers "\*" .ttb" (fichiers d'interface utilisateur V-rep).
- **Barre de menu:** la barre de menus permet d'accéder à presque toutes les fonctionnalités du simulateur. La plupart du temps, les éléments de la barre de menus activent une boîte de dialogue. Le contenu de la barre de menu dépend du contexte. La



plupart des fonctions de la barre de menus sont également accessibles via un menu contextuel.

- **Barres d'outils:** les barres d'outils présentent des fonctions auxquelles on accède souvent (par exemple, changer le mode de navigation, sélectionner une autre page, etc.). Certaines fonctions de la barre d'outils {1} et toutes les fonctions de la barre d'outils {2} sont également accessibles via la barre de menus ou le menu contextuel. Les deux barres d'outils peuvent être amarrées et désamarrées, mais l'amarrage ne fonctionne qu'avec leurs positions initiales respectives

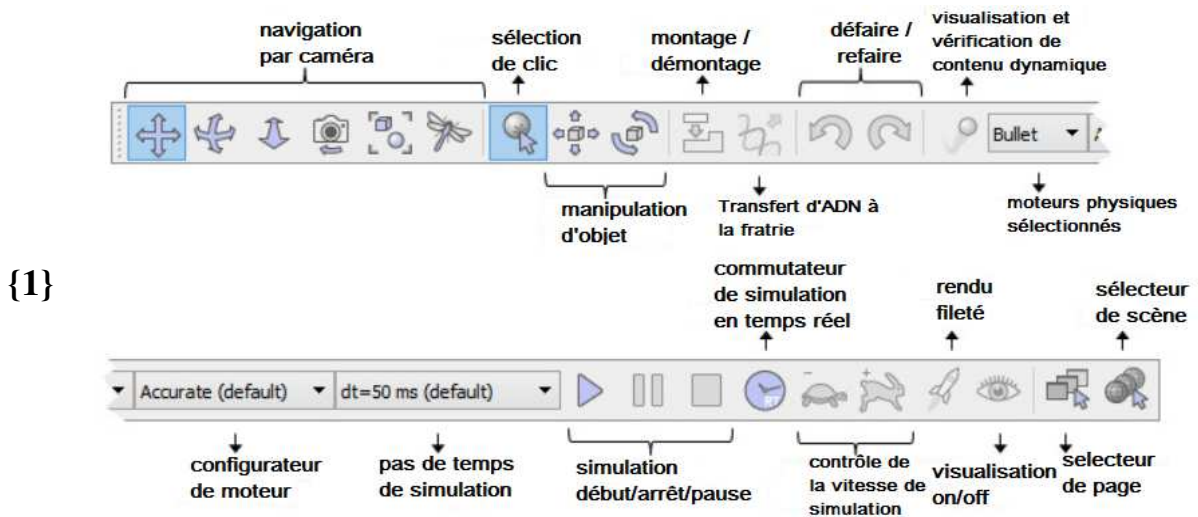
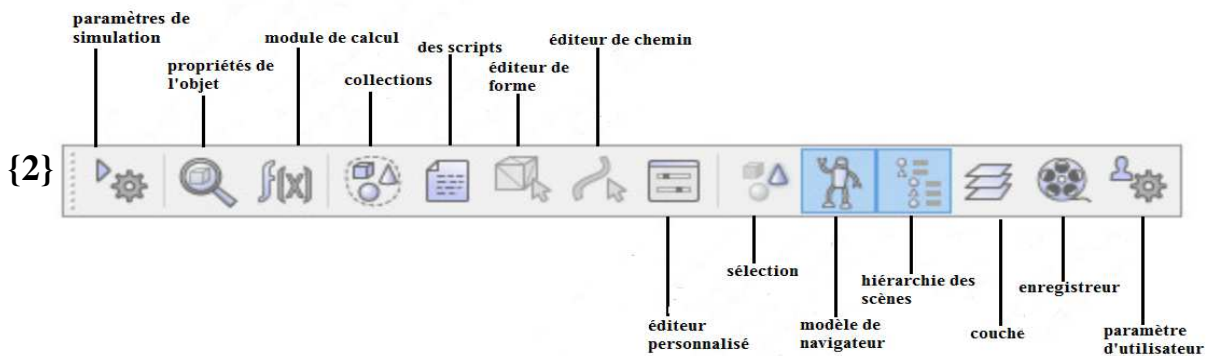


Figure 3.2 : représentation du fonctionnement de chaque bouton de la barre d'outils {1}



- 1 **Figure 3.3 :** représentation du fonctionnement de chaque bouton de la barre d'outils {2} basculé à l'aide du bouton correspondant de la barre d'outils. Il affiche dans sa partie supérieure une structure de dossiers de modèles V-rep et dans sa partie inférieure, Les miniatures peuvent être glissées et déposées dans la scène pour charger

## Chapitre 3

### Implémentation

automatiquement le modèle associé. Les vignettes capturées apparaissent sombres si la zone de dépôt n'est pas prise en charge ou n'est pas appropriée.

- **Hierarchie de scènes:** la hiérarchie de scènes est visible par défaut, mais peut être basculée avec le bouton correspondant de la barre d'outils. Il affiche le contenu d'une scène (c'est-à-dire tous les objets de la scène composant une scène). Étant donné que les objets de scène sont construits dans une structure semblable à une hiérarchie, la hiérarchie de la scène affiche une arborescence de cette hiérarchie et des éléments individuels peuvent être développés ou réduits. Un double clic sur une icône ouvre / ferme une boîte de dialogue de propriété associée à l'icône sur laquelle vous avez cliqué. Un double clic sur un nom d'objet permet de le modifier. La molette de la souris ainsi qu'un glissement des barres de défilement de la vue hiérarchique de la scène permettent de déplacer le contenu vers le haut / le bas ou vers la gauche / la droite. La sélection de contrôle et de décalage est toujours prise en charge. Les objets de la hiérarchie de la scène peuvent être glissés et déposés sur un autre objet afin de créer une relation parent-enfant.

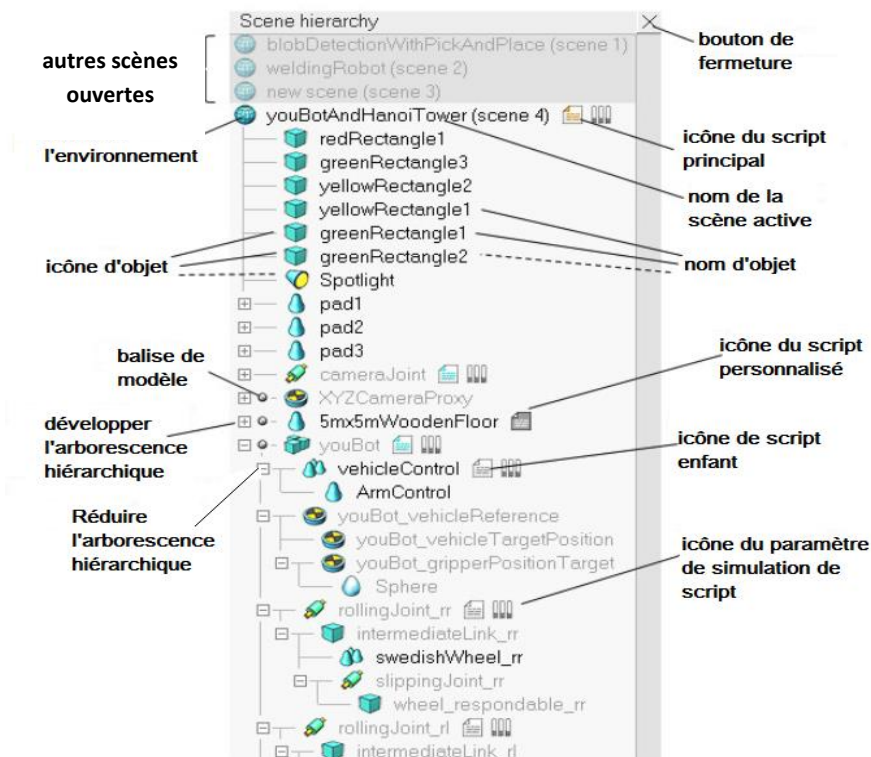


Figure 3.4 : représentation de différent élément de la hiérarchie de scènes

- **Page:** chaque scène peut contenir jusqu'à 8 pages, chacune d'elles pouvant contenir un nombre illimité de vues. Une page peut être vue comme conteneur pour les vues.
- **Vues:** il peut y avoir un nombre illimité de vues contenues dans une page. Une vue est utilisée pour afficher la scène (contenant elle-même un environnement et des objets), vue à travers un objet visible par exemple: (caméras, graphiques ou capteurs de vision,).
- **Texte d'information:** il affiche des informations relatives à la sélection d'objet / d'élément en cours et aux états ou paramètres de simulation en cours. L'affichage du texte peut être basculé avec l'un des deux petits boutons situés dans le coin supérieur gauche de la page. L'autre bouton peut être utilisé pour basculer sur un arrière-plan blanc.
- **Barre d'état:** la barre d'état affiche des informations sur les opérations effectuées, les commandes et les messages d'erreur de l'interpréteur Lua. Depuis un script, l'utilisateur peut également générer des chaînes dans la barre d'état à l'aide de la fonction `sim.addStatusbarMessage`. La barre d'état n'affiche que deux lignes par défaut, mais elle peut être redimensionnée à l'aide de sa poignée de séparation horizontale.
- **Menu contextuel:** les menus contextuels sont les menus qui apparaissent après un clic du bouton droit de la souris. Pour activer un menu contextuel, assurez-vous que la souris ne bouge pas lors du clic, sinon le mode de rotation de la caméra risque d'être activé. Chaque surface de la fenêtre de l'application (vue hiérarchique des scènes, page, vue, etc.) peut déclencher un menu contextuel différent (sensible au contexte). Le contenu des menus contextuels peut également changer en fonction de l'état actuel de la simulation ou du mode d'édition. La plupart des fonctions du menu contextuel sont également accessibles via la barre de menus, à l'exception de l'élément du menu Afficher qui apparaît uniquement lorsque le menu contextuel est activé sur une vue ou une page.

#### 3.2.2 Mise en œuvre du V-rep

V-rep est un simulateur hautement personnalisable, chaque aspect d'une simulation peut être personnalisé. De plus, le simulateur lui-même peut être personnalisé et adapté afin de se comporter exactement comme vous le souhaitez. Ceci est autorisé via une API (Application Programming Interface) élaborée. Six approches de programmation ou de codage différentes sont prises en charge, chacune présentant des avantages particuliers (et évidemment aussi des

## Chapitre 3

### Implémentation

inconvenients) par rapport aux autres, mais toutes les six sont compatibles les unes des autres (c'est-à-dire qu'elles peuvent être utilisées simultanément ou même ensemble). L'entité de contrôle d'un modèle, d'une scène ou du simulateur lui-même peut être située à l'intérieur.

a) **Scripts intégrés:** c'est un script intégré dans une scène (ou un modèle), c'est-à-dire un script faisant partie de la scène, qui sera enregistré et chargé avec le reste de la scène (ou du modèle).

Deux principaux types de scripts intégrés sont supportés:

- ✓ **Scripts de simulation:** les scripts de simulation sont des scripts exécutés uniquement lors de la simulation et utilisés pour personnaliser une simulation ou un modèle de simulation. La boucle de simulation principale est gérée via le script principal et les modèles / robots sont contrôlés via des scripts enfants.
- ✓ **Scripts de personnalisation:** ce sont des scripts qui peuvent également être exécutés lorsque la simulation n'est pas en cours d'exécution et qui permettent de personnaliser une scène de simulation ou le simulateur lui-même.

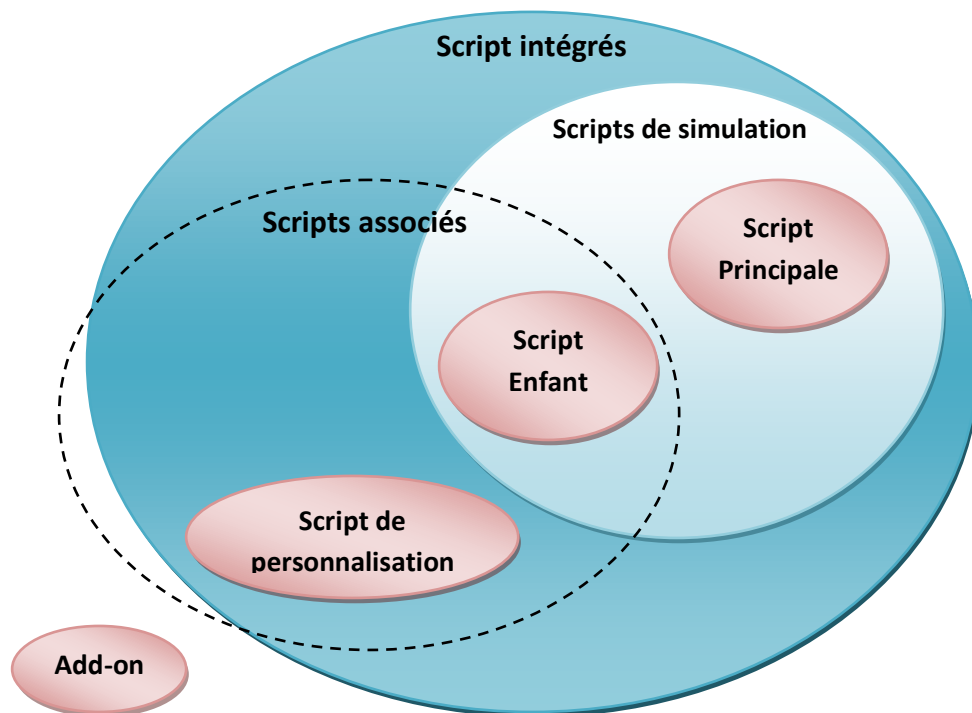


Figure 3.5 : Les approches de programmation prise en charge

- **Script principal:** c'est un script de simulation. Par défaut, chaque scène de V-rep aura un script principal. Il contient le code de base permettant l'exécution d'une simulation. Sans script principal, une simulation en cours ne fera rien. Le script principal contient des fonctions appelées de manière appropriée par le système. Si une fonction donnée n'est pas définie, l'appel sera ignoré. À l'exception de la fonction d'initialisation, toutes les autres fonctions sont facultatives.

Le script principal n'est pas censé être modifié. La raison en est la suivante: l'un des points forts de V-rep est que tout modèle (robot, actionneur, capteur, etc.) peut être copié dans une scène et est immédiatement opérationnel. Lorsque vous modifiez le script principal, vous courez le risque que les modèles ne fonctionnent plus comme prévu (par exemple, si la commande **sim.handleChildScripts** manque dans votre script principal, tous les modèles copiés dans la scène ne fonctionneront pas du tout). Une autre raison est que le fait de conserver un script principal par défaut permet aux anciennes scènes de s'adapter facilement à de nouvelles fonctionnalités (par exemple, si une nouvelle version de V-rep introduit une commande soignée **sim.doMagic()**, les anciennes scènes seront automatiquement mises à jour pour que cette commande appelé dans leur script principal).

- **Script enfant :** c'est un script de simulation. V-rep prend en charge un nombre illimité de scripts enfants par scène. Chaque script enfant représente un petit ensemble de routines écrites en Lua permettant de gérer une fonction particulière dans une simulation. Les scripts enfants sont associés aux objets de la scène et peuvent être facilement reconnus à partir de leur icône de script dans la hiérarchie des scènes. Les scripts enfants peuvent être associés à une liste de paramètres de simulation, appelés paramètres de simulation de script. Ces paramètres peuvent être utilisés comme moyen rapide d'ajuster les valeurs d'un modèle de simulation spécifique (par exemple, la vitesse maximale d'un robot mobile ou la résolution d'un capteur). Les scripts enfants peuvent être de deux types différents: les scripts enfants non-filetés ou les scripts enfants filetés:

- ✓ **scripts enfants non-filetés:** Les scripts enfants non-filetés contiennent une collection de fonctions de blocage. Cela signifie que chaque fois qu'ils sont appelés, ils doivent effectuer une tâche, puis renvoyer le contrôle. Si le contrôle n'est pas renvoyé, alors toute la simulation s'arrête. ils suivent un ordre d'appels ou d'exécution précis, par défaut, les scripts enfants sont appelés en commençant par les objets sans enfant et en terminant par les objets racine (ou objets sans

parent). La commande **sim.handleChildScripts**, appelée par défaut à partir du script principal, gère l'appel de scripts enfants non-filetés.

- ✓ **scripts enfants filetés:** Les scripts enfants filetés sont des scripts qui se lanceront dans un fil. Le lancement (et la reprise) des scripts enfants filetés est géré par le code de script principal par défaut, via les fonctions **sim.launchThreadedChildScripts** et **sim.resumeThreads**. Le lancement / la reprise des scripts enfants filetés est exécuté dans un ordre précis. Lorsque l'exécution d'un script enfant fileté est toujours en cours, il ne sera pas lancé une seconde fois. Lorsqu'un script enfant fileté est terminé, il ne peut être relancé que si l'option *Exécuter une fois* dans les propriétés du script n'est pas cochée. Une icône de script enfant fileté dans la hiérarchie de la scène est affichée en bleu clair au lieu de blanc pour indiquer qu'il sera lancé dans un fil.

Les scripts enfants filetés présentent plusieurs faiblesses par rapport aux scripts enfants non-filetés s'ils ne sont pas programmés correctement: ils utilisent plus de ressources, ils peuvent perdre du temps de traitement et peuvent être un peu moins sensibles à une commande d'arrêt de la simulation.

- b) **Add-ons:** elle consiste à écrire des scripts Lua, permet de personnaliser rapidement le simulateur lui-même. Les modules complémentaires peuvent démarrer automatiquement et s'exécuter en arrière-plan, ou ils peuvent être appelés en tant que fonctions (par exemple, pratique lors de la rédaction d'importateurs / exportateurs). Les modules complémentaires ne doivent pas être spécifiques à une simulation ou à un modèle donné, ils doivent plutôt offrir une fonctionnalité plus générique, liée au simulateur.
- c) **Plug-ins:** cette méthode consiste essentiellement à écrire un plugin pour V-rep. Souvent, les plugins ne sont utilisés que pour fournir une simulation avec des commandes Lua personnalisées, ils sont donc utilisés avec la première méthode. D'autres fois, les plugins fournissent à V-rep une fonctionnalité spéciale nécessitant soit une capacité de calcul rapide (les scripts sont généralement plus lents que les langages compilés), une interface spécifique avec un périphérique matériel (par exemple, un vrai robot), ou une fonction spéciale, interface de communication avec le monde extérieur.

**d) Remote API clients:** cette méthode permet à une application externe de se connecter très facilement à V-rep à l'aide de commandes d'API distantes. Il est composé de services de serveur d'API distants et de clients API distants. Le côté client peut être intégré sous forme de code à faible encombrement (C / C ++, Python, Java, Matlab et Urbi) dans pratiquement tout matériel, y compris les robots réels.

- **Activation de l'API distante - côté serveur :** Pour activer l'API distante côté serveur (c'est-à-dire côté V-rep), vous devez assurer que le plug-in d'API distant a été chargé avec succès au démarrage de V-rep (`v_repExtRemoteApi.dll` (windows), `libv_repExtRemoteApi.dylib` (mac) ou `libv_repExtRemoteApi.so` (linux)). Un service de serveur peut être démarré de deux manières différentes:

- ✓ **Au démarrage de V-rep (service de serveur API distant continu) :** Le plug-in API distant essaiera de lire un fichier de configuration nommé `remoteApiConnections.txt` et, selon son contenu, démarrera les services de serveur appropriés. Utilisez cette méthode pour contrôler à distance le simulateur lui-même. Avec cette méthode, les fonctions d'API distantes seront toujours exécutées côté serveur, même si la simulation n'est pas en cours d'exécution (ce qui n'est pas toujours le cas avec la méthode suivante ci-dessous).

- ✓ **À partir d'un script (service de serveur d'API distant temporaire) :** C'est la plupart du temps la méthode recommandée pour démarrer un service de serveur API distant. L'utilisateur a le contrôle lorsque le service est démarré ou arrêté. Cependant, lorsqu'un service de serveur d'API distant temporaire est démarré à partir d'un script de simulation, le service sera automatiquement arrêté à la fin de la simulation. Un service de serveur d'API distant temporaire peut être démarré ou arrêté avec les 2 fonctions Lua suivantes (les 2 fonctions sont exportées par le plug-in): **`simRemoteApi.start`**, **`simRemoteApi.stop`**. Vous pouvez collecter des informations sur tout service de serveur API distant avec la fonction **`simRemoteApi.status`**.

- **Activation de l'API distante - côté client :**

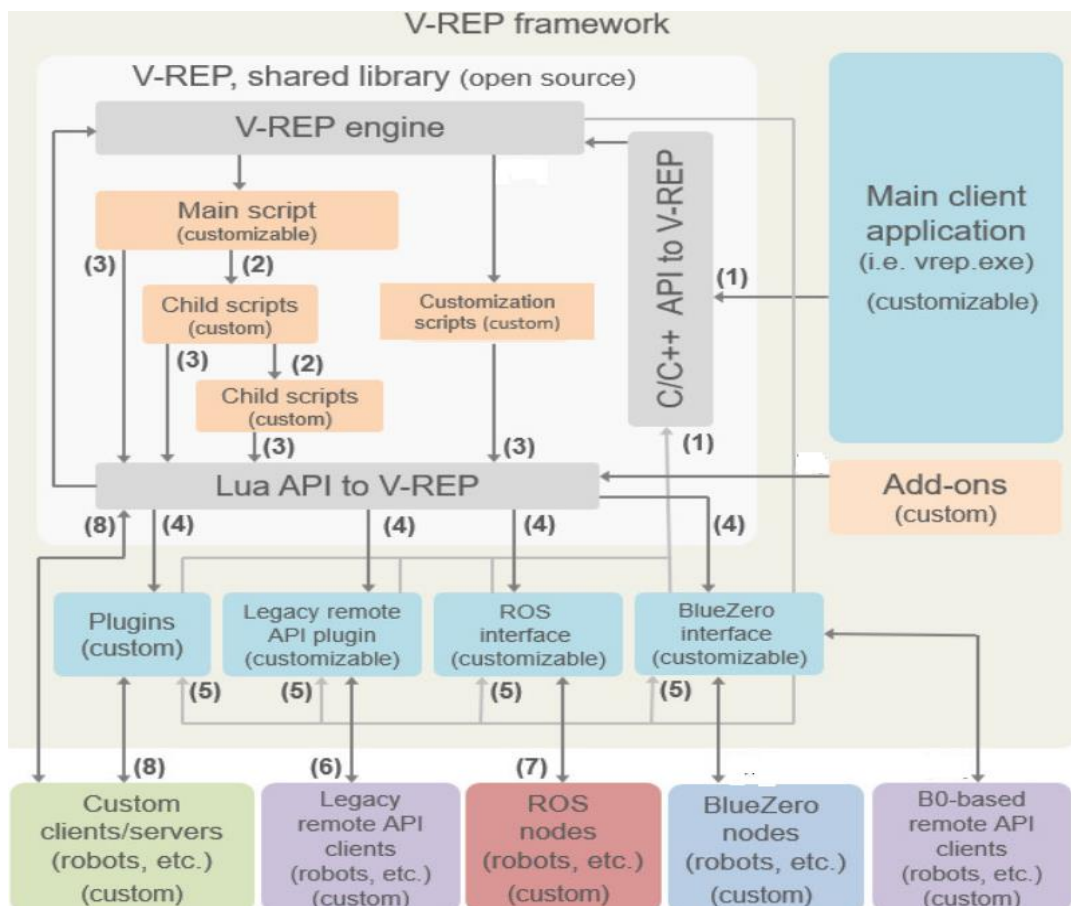
**Python client :** Pour utiliser la fonctionnalité d'API distante dans votre script Python, vous aurez besoin des 3 éléments suivants:

1. `vrep.py`
2. `vrepConst.py`
3. `remoteApi.dll`, `remoteApi.dylib` ou `remoteApi.so` (selon la plate-forme cible).

Les fichiers ci-dessus sont situés dans le répertoire d'installation de V-REP, sous `programming / remoteApiBindings / python`.

Une fois que vous avez les éléments ci-dessus dans un répertoire connu de Python, appelez `import vrep` pour charger la bibliothèque. Pour activer l'API distante côté client (c'est-à-dire votre application), appelez `vrep.simxStart`.

- e) **ROS nodes:** V-rep implémente un nœud ROS avec un plug-in qui permet aux ROS d'appeler des commandes V-rep via des services ROS ou de transmettre des données.
- f) **NoeudBlueZero:** Ce procédé permet à une application externe (située par exemple sur un robot, une autre machine, etc.) de se connecter à V-rep via BlueZero.



**Figure3.6 :** Architecture de contrôle V-REP

Dans la figure 3.6 on trouve : (1) appels API C / C ++, (2) exécution de script enfant en cascade, (3) appels API Lua, (4) rappels personnalisés de Lua API, (5) rappels d'événements V-rep, (6) appels de fonction API distants, (7) ROS transit, (8) communication personnalisée (prise, série, tuyaux, etc.) [35].



#### 3.2.3 Les objets de scène

Les principaux éléments de V-rep qui sont utilisés pour créer une scène de simulation sont des objets de scène. Une scène ou un modèle de simulation V-rep contient plusieurs objets de scène ou objets élémentaires assemblés dans la hiérarchie arborescente. Les objets de scène suivants sont pris en charge dans V-rep:

- **Joints:** c'est un objet commun ou un actionneur. Ils sont des éléments qui relient deux objets de scène ou plus il existe 4 types (prismatique, révolutionnaire, en forme de vis ou sphérique).
- **Graphes:** un graphique est utilisé pour enregistrer et visualiser les données de simulation et l'affichés directement ou des courbes 3D.
- **Caméras:** c'est un objet qui permet de visualiser la scène de simulation sous différents angles.
- **Lumières:** les lumières éclairent une scène ou des objets de la scène et influencent directement les caméras ou les capteurs de vision.
- **Chemins:** un chemin est un objet qui définit un chemin ou une trajectoire dans l'espace. Il peut être utilisé pour des buts divers, également en tant que joint ou actionneur personnalisé.
- **Formes:** une forme est un maillage rigide composé de faces triangulaires, utilisées pour la simulation et la visualisation de corps rigide. Ils peuvent être optimisés pour le calcul rapide de la réponse dynamique aux collisions.
- **Capteurs de vision:** un capteur de vision est un capteur de type caméra réagissant à la lumière, aux couleurs et aux images. Permettent d'extraire des informations d'image complexes d'une scène de simulation (couleurs, tailles d'objet, cartes de profondeur, etc.)
- **Capteurs de proximité:** un capteur de proximité détecte des objets de manière géométriquement exacte dans son volume de détection. V-rep prend en charge les capteurs de proximité de type pyramide, cylindre, disque, cône et rayon.
- **Mannequins (nuls):** un mannequin est un point d'orientation. Les nuls sont des objets polyvalents pouvant avoir de nombreuses applications différentes.
- **Capteurs de force:** un capteur de force est un objet capable de mesurer les forces et les couples qui lui sont appliqués. Il a également la capacité de casser si un seuil donné est dépassé [36].

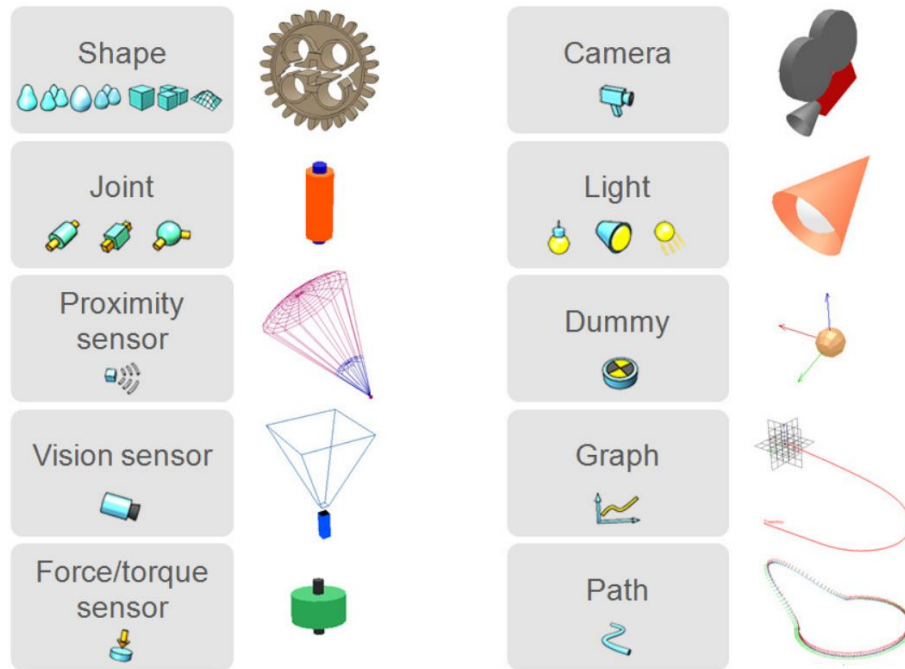


Figure 3.7 : quelques exemples des objets utilisés dans V-REP et leur représentation en 3D

### 3.3 Choregraphe

Choregraphe est un environnement graphique très intuitif qui permet une programmation simple de Nao. Lorsque le logiciel est lancé (les versions sont disponibles sous Windows XP, Linux Ubuntu et Mac OS X Leopard), l'interface utilisateur graphique illustrée à la figure 3.8 apparaît à l'écran. La fenêtre de l'application est divisée en trois zones: Zone 1, la bibliothèque de boîtes qui regroupe la liste des comportements disponibles. Zone 2, le diagramme de flux qui permet à l'utilisateur de présenter graphiquement des comportements composés de boîtes de bibliothèque et de liens entre elles. Dans la zone 3 est une représentation graphique de Nao capable d'exécuter le comportement mis en œuvre.

Programmer Nao consiste à mettre en œuvre des comportements, eux-mêmes composés de plusieurs comportements. Dans la bibliothèque de boîtes, un ensemble de comportements préprogrammés classiques sont proposés, des fonctions de haut niveau (marcher, danser, tourner, s'allonger, se lever, synthèse de la parole, reconnaissance vocale) jusqu'à des comportements de très faible niveau (lire les capteurs, allumer et éteindre les leds). En assemblant ces comportements de base, il est possible de créer un comportement original. L'assemblage des comportements est effectué dans le diagramme de flux. En «glissant-déposant» des icônes de comportement de la zone 1 à la zone 2. Dans la zone 2, un comportement est représenté par une icône dotée de petits carrés à gauche (zones d'entrée) et de petits carrés à droite (zones de sortie). Connecter la boîte de sortie d'un comportement à la boîte d'entrée d'un autre signifie que le second sera exécuté lorsque le premier sera terminé (programmation d'événement). En haut à gauche de la zone 2, la «zone d'entrée» globale représente le point de départ du comportement. En haut à droite, la «boîte de sortie» globale représente la fin du comportement. Le principe de la programmation de Nao consiste à

## Chapitre 3

### Implémentation

connecter des comportements séquentiels ou parallèles entre la "zone d'entrée" et la "zone de sortie" globale.

Si un vrai Nao est connecté, via Wi-Fi, au PC qui exécute Choregraphe, le vrai Nao exécutera le comportement global. Si aucun Nao réel n'est disponible ou si le comportement doit être testé avant son utilisation réelle, Choregraphe est connecté au NAO graphique 3D de la zone 3. Le Nao virtuel exécute le comportement programmé. Afin de donner un aperçu global de Choregraphe, deux autres zones sont présentées à la figure 2. Zone 4 est un éditeur de script pour les instructions Python permettant de développer de nouvelles boîtes de manière plus classique (calcul, branches conditionnelles...). La zone 5 est la chronologie qui permet la programmation en fonction du temps [37].

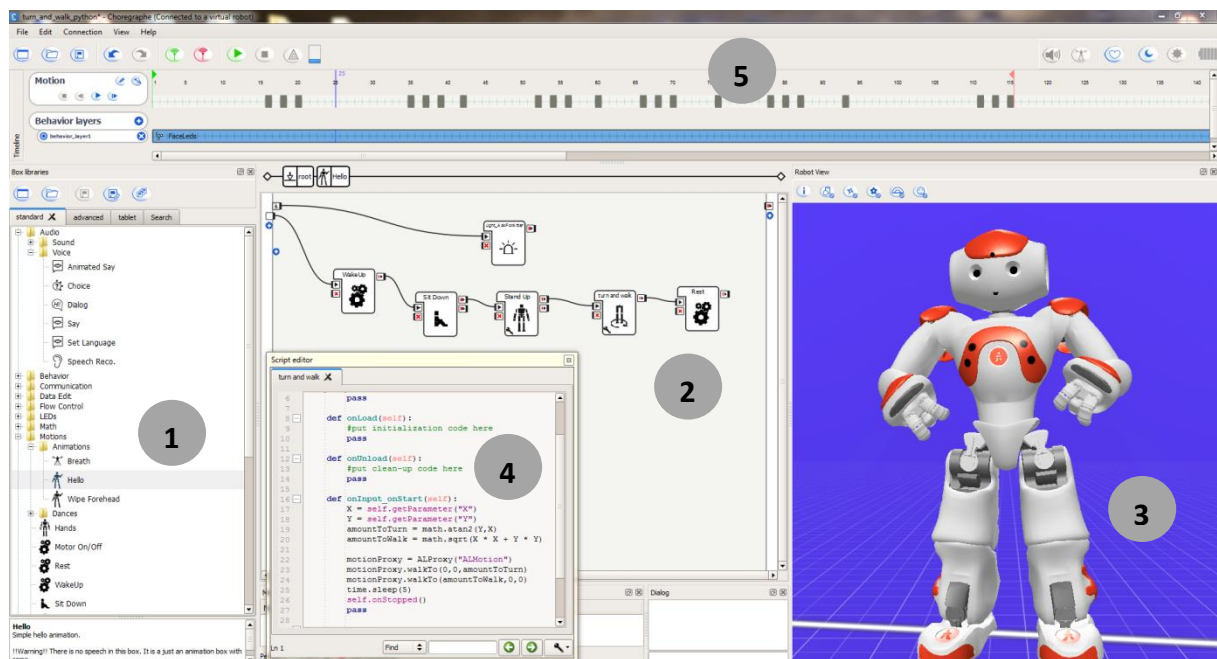


Figure 3.8 : L'interface graphique

### 3.4 Naoqi

NAOqi est le logiciel principal exécuté sur le robot NAO. NAOqi est multi-plateforme, ce qui signifie qu'il peut être utilisé sous Windows, Linux et Mac OS. De plus, c'est un langage croisé qui permet de créer un logiciel qui s'exécute sur le robot en utilisant C++ et python. Le framework permet de créer des programmes distribués qui s'exécutent à distance à partir d'un ordinateur ou localement sur le robot. [38] Il permet également une communication homogène entre différents modules, tels que le mouvement, l'audio et la vidéo. Comme une programmation et un partage d'informations similaires, Naoqi est livré avec une liste de modules de base fournissant toutes les interfaces de programmation d'application (API) nécessaires à la programmation du NAO.

Des kits de développement logiciel (SDK) sont disponibles dans différents langages de programmation, qui incluent Python, C++, Java, MATLAB et Octave. en utilisant le package SDK de simulation, les développeurs peuvent utiliser toutes les API NAOqi pour programmer le robot simulé et obtenir avec précision toutes les lectures de capteurs et de courants. Ce

## Chapitre 3

### Implémentation

package SDK est disponible au téléchargement sur le site Web d'Aldebaran. Un plug-in est nécessaire pour utiliser le package SDK dans tout simulateur 3D afin que celui-ci sache quelles bibliothèques utiliser et où les trouver.

Python est le langage de programmation le plus simple pour commencer, car il n'est pas nécessaire de compiler le code sur un ordinateur avant d'envoyer l'exécutable au robot comme dans le cas de C++.

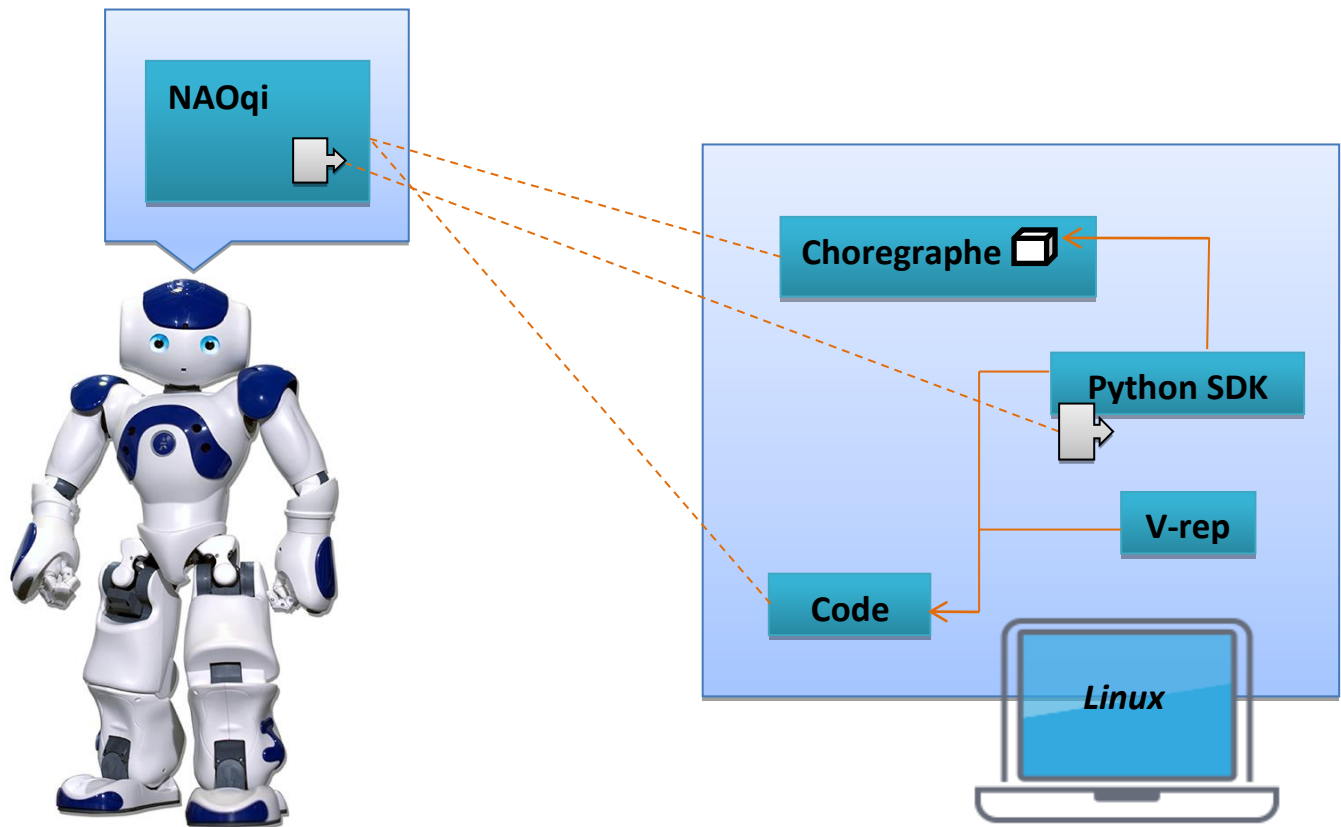


Figure 3.9 : les logiciels du NAO.

#### 3.5 Implémentation

Ce projet a pour objectif de simuler un NAO sur V-rep. L'idée principale est de pouvoir implémenter des modules de marche dans un environnement virtuel. En plus de V-rep, nous allons utiliser la Choregraphe suite et le SDK Python NAOqi d'Aldebaran. Notre projet est basé sur le projet de PierreJac/Project-NAO-Control. Nous avons modifié le script `single_ao_control.py` en ajoutant nos modules de marche.

##### 3.5.1 Les modules de marche

La locomotion de NAO est stabilisée grâce aux retours de ses capteurs articulaires. Cela rend la marche robuste contre les petites perturbations et absorbe les oscillations du torse dans les plans frontal et latéral. Il existe 4 moyens de haut niveau pour contrôler la locomotion de robot NAO :

- `ALMotionProxy::move(const float& x, const float& y, const float& theta)`: elle définit la vitesse instantanée de NAO (direction et intensité) en unités SI  
*x*: vitesse le long de l'axe X, en mètres par seconde. Utiliser des valeurs négatives pour le recul  
*y*: vitesse suivant l'axe Y, en mètres par seconde. Utilisez des valeurs positives pour aller à gauche  
*theta* : vitesse autour de l'axe Z, en radians par seconde. Utilisez des valeurs négatives pour tourner dans le sens des aiguilles d'une montre.
- `ALMotionProxy::moveTo(const float& x, const float& y, const float& theta)`: pour définir une pose cible sur le plan de sol, NAO se dirigera vers la cible donnée  
*x* : Distance le long de l'axe X en mètres.  
*y* : Distance le long de l'axe des Y en mètres  
*theta* : Rotation autour de l'axe Z en radians [-3.1415 à 3.1415].
- `ALMotionProxy::moveToward(const float& x, const float& y, const float& theta)`: pour définir de manière interactive la vitesse instantanée normalisée de NAO  
*x*: vitesse normalisée, sans unité, le long de l'axe X. +1 et -1 correspondent à la vitesse maximale dans les directions avant et arrière, respectivement.  
*y*: vitesse normalisée, sans unité, le long de l'axe des ordonnées. +1 et -1 correspondent à la vitesse maximale dans les directions gauche et droite, respectivement  
*theta*: normalisé, sans unité, vitesse autour de l'axe Z. +1 et -1 correspondent à la vitesse maximale dans le sens antihoraire et dans le sens horaire, respectivement.
- `ALMotionProxy::setWalkTargetVelocity(const float & x, const float & y, const float & theta, const float & frequency, const AL::ALValue & leftFootGaitConfig, const AL::ALValue & rightFootGaitConfig)`: elle définit la longueur et la fréquence du pas instantané (normalisé) de NAO, et ainsi contrôler sa vitesse indirectement. avec une configuration précise de chaque pied.  
*x* : Fraction de MaxStepX. Utilisez négatif pour se déplacer en arrières. [-1.0 à 1.0]  
*y* : Fraction de MaxStepY. Utilisez négatif pour la direction droit. [-1.0 à 1.0]  
*theta* : Fraction de MaxStepTheta. Utilisez négatif pour sens horaire [-1.0 à 1.0]  
*frequency* : Fraction de MaxStepFrequency [0.0 à 1.0]

## Chapitre 3

### Implémentation

*leftFootGaitConfig* : Une valeur de ALValue avec une configuration de démarche personnalisée pour le pied gauche

*rightFootGaitConfig* : Une valeur de ALValue avec une configuration de démarche personnalisée pour le pied droit.

En utilisant les moyens de locomotion précédents nous avons décidé d'implémenter trois modules de marche.

#### a) Straight forward

Dans ce module nous allons faire marcher NAO tout droit, en lui donnant une configuration de déplacement personnalisée. C'est-à-dire nous allons personnaliser les paramètres de la démarche en planifiant les pas de robot. Pour cela nous avons implémenté le programme suivant :

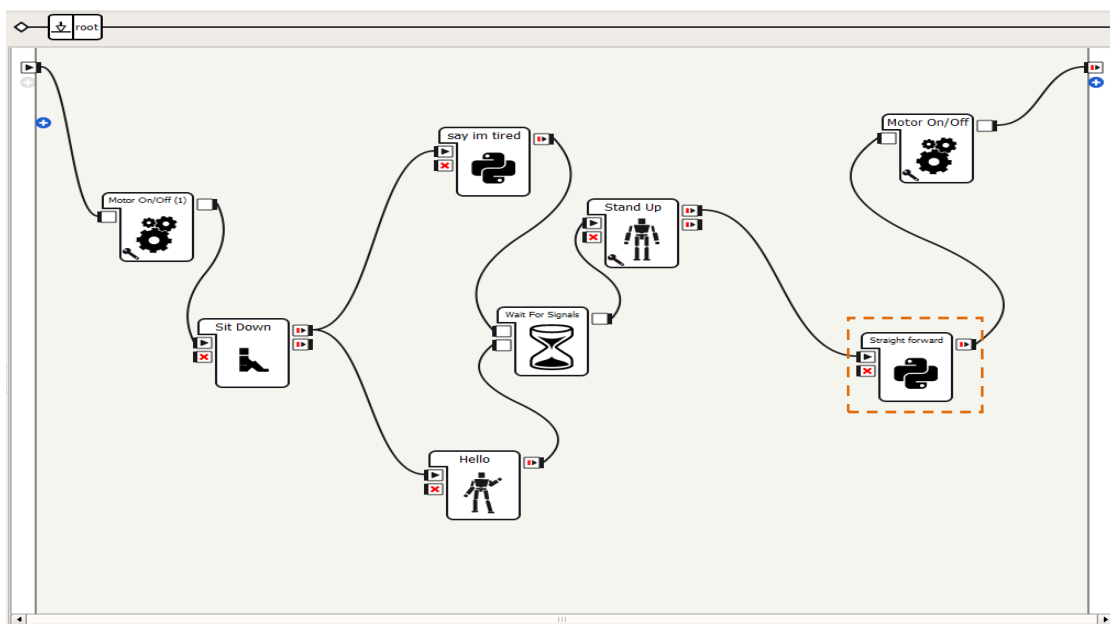


Figure 3.10 : Diagramme de flux dédié au modèle Straight forward

#### b) Walk and turn

Il s'agit de faire tourner NAO tout en gardant la marche vers des coordonnées attribués, puis retourner et revenir au point de départ. La figure 3.10 montre le code utilisé.

## Chapitre 3

### Implémentation

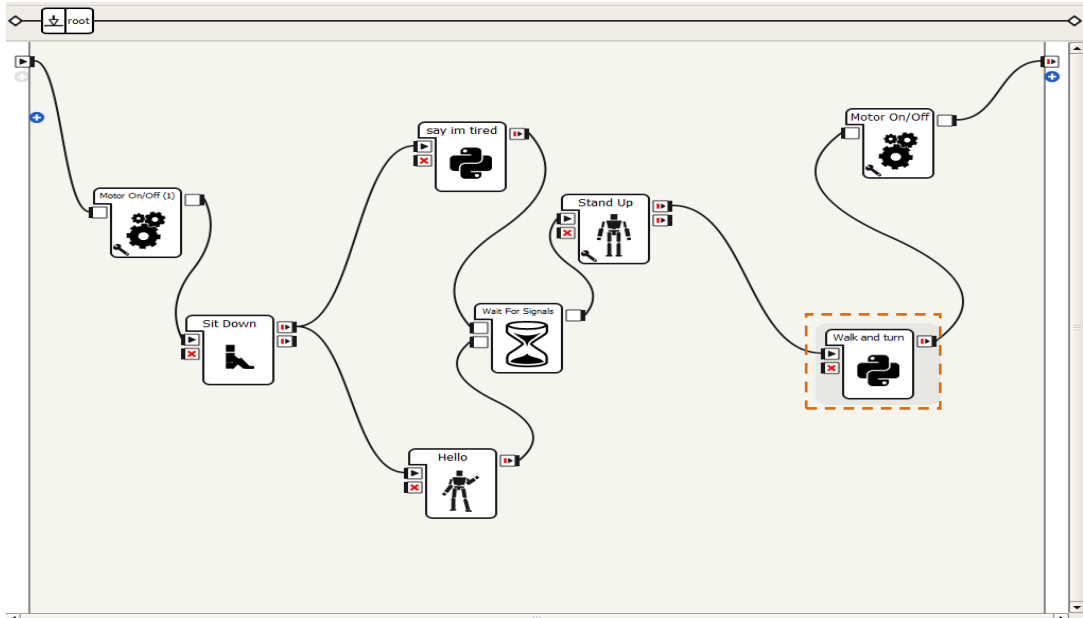


Figure 3.11 : Diagramme de flux dédié au modèle Walk and Turn

#### c) Omni-directional walk

Le robot NOA est supposé marcher dans toutes les directions, Le planificateur de pied utilisé dans le moteur de marche est très simple et permet une marche omnidirectionnelle pour NAO. La figure 3.11 montre le programme utilisé.

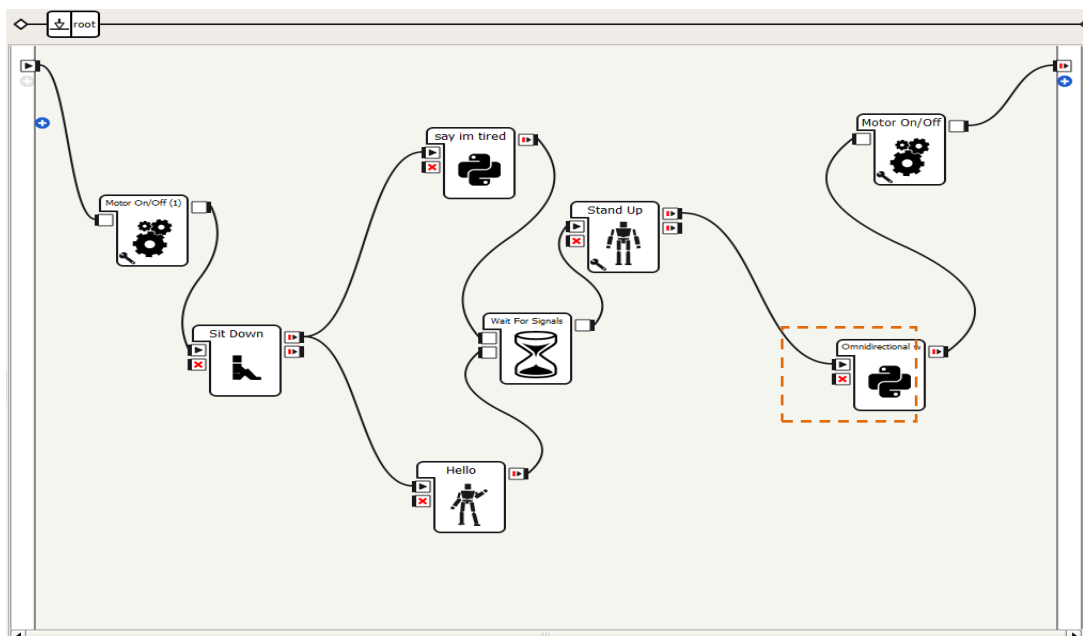
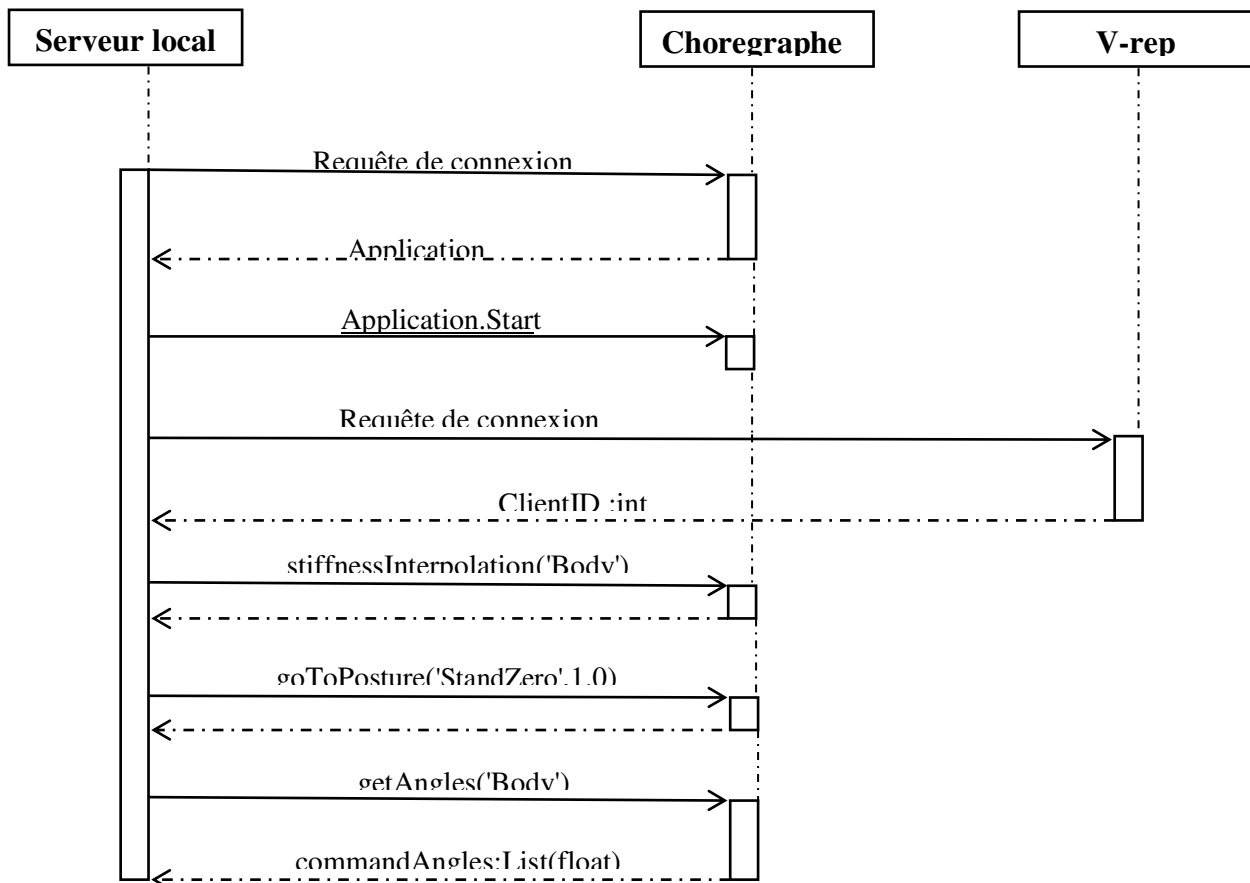


Figure 3.12 : Diagramme de flux dédié au modèle Omnidirectional walk

### 3.5.2 Etablissement de connexion

Dans ce paragraphe nous allons expliquer le concept d'établissement de connexion entre les plateformes et les packages utilisés. La figure 3.12 représente le diagramme des séquences de connexion.



**Figure 3.13:** le diagramme des séquences de connexion.

1. Le serveur local envoie une demande de connexion à NAO qi, avec l'adresse IP et le numéro de port de NAO qi, puis NAO qi renvoie une référence à son objet d'application. Sur la figure 3.9, l'appel de demande de connexion effectué par le serveur local à NAO qi est une abstraction de ce processus.
2. À l'aide de la référence de NAO qi, le serveur local essaie d'envoyer la commande START à NAO qi, qui, si elle réussit, initialise NAO qi et crée une session. Une session est ce qui permet au serveur de connecter les services de NAO qi ensemble localement ou sur le réseau.
3. Si la session a été créée avec succès, le serveur local envoie une demande de connexion au serveur V-rep, avec l'adresse IP du serveur V-rep, le numéro de port et quelques autres options de connexion. Si la demande a été acceptée, l'API du serveur V-rep renvoie une valeur entière - ClientID - différent de -1. Sur la figure 3.9, l'appel de demande de connexion effectué par le serveur local auprès du serveur V-rep est une abstraction de ce processus.



4. Après l'activation de l'API du serveur V-rep, le serveur local envoie la commande `stiffnessInterpolation("Body")`. C'est la limitation de couple dans les moteurs, Si elle est définie sur 0.0, le contrôleur de joint ne fait rien et le joint est libre. Sinon avec 1.0 le joint est autorisé à utiliser toute la puissance du couple pour atteindre une position donnée.
5. Le serveur local tente d'envoyer la commande d'appel de blocage `goToPosture("StandZero", 1.0)` à NAO qi, si elle réussit, fait passer le robot simulé NAO qi à la position prédéfinie demandée dans le premier paramètre, c'est-à-dire «StandZero», qui est la position initiale du NAO simulé V-REP. Le deuxième paramètre de cette méthode est la vitesse relative du mouvement dont la valeur est comprise entre 0,0 et 1,0.
6. Maintenant, à ce stade, le serveur local essaie d'envoyer la commande `getAngles("Body", false)` à NAO qi, qui, si elle réussit, renvoie la liste des angles de joint du robot simulé NAO qi [35].

#### 3.5.3 Simulation

Nous allons maintenant simuler notre robot NAO sur la plateforme V-rep. Pour cela nous effectuons les étapes suivantes :

- On lance V-rep et on charge la scène contenue dans le dossier \* Vrep-scene \*.
- dans le dossier chorégraphe suite, puis dans \*naoqi-bin\*, on lance la commande `./naoqi-bin -p [numéro de port] &`.
- On lance chorégraphe suite pour visualiser notre NAO virtuel et vérifier son IP et son port en utilisant le bouton de connexion.
- On Lance la simulation V-rep (ou les scripts ne fonctionneront pas). Ensuite, on lance le script `single_ao_control.py` en donnant toutes les informations nécessaires (IP et port) et on attend que NAO soit en écoute.

La figure 3.14 montre en détail les étapes précédentes.

# Chapitre 3

## Implémentation

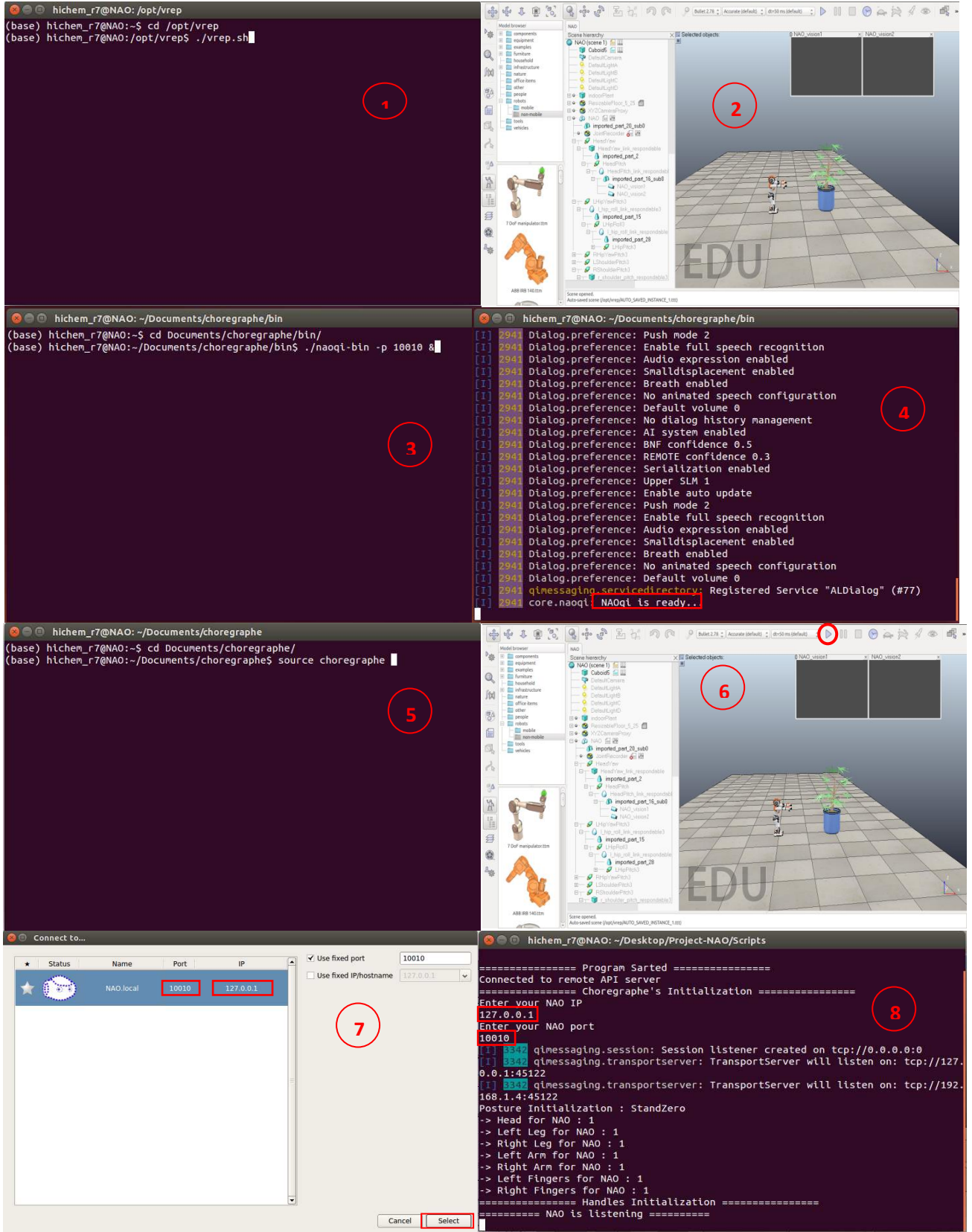


Figure 3.14 : simulation de robot NAO sur V-rep.

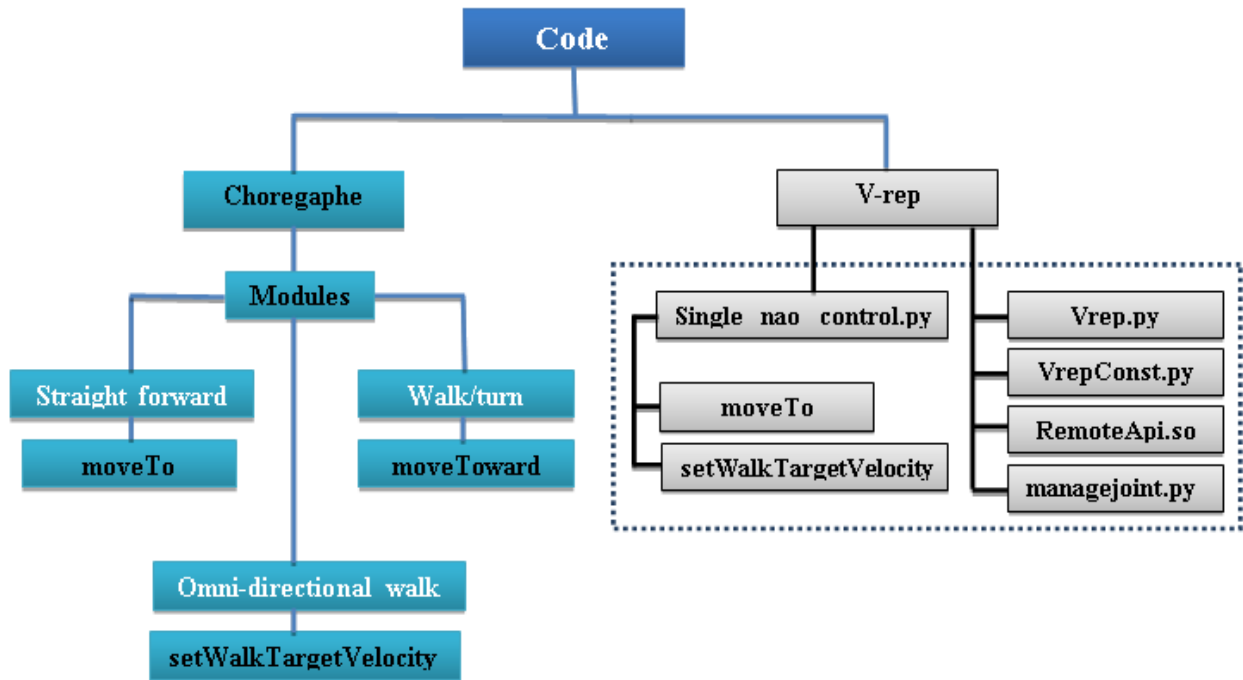


Figure 3.15 : Organigramme des modules implémentés.

### 3.6 Conclusion:

Actuellement, V-rep s'est développé pour devenir un simulateur et un contrôleur de robot robuste et largement utilisé, présent aussi bien dans le domaine universitaire que dans le secteur industriel.

Le laboratoire virtuel robotique montre les concepts obscurs de la robotique sous forme d'interaction simple et sous forme de simulation virtuelle. Le laboratoire virtuel économise plus de fonds pour l'équipement expérimental de plusieurs collèges et universités, économise de l'espace d'enseignement et amélioreront l'efficacité de l'enseignement.

*Conclusion*  
*Générale*

## Conclusion Générale

---

En conclusion nous devons avouer que nous sommes très satisfait de travail réalisé, car nous avons atteint notre objectif d'implémenter des modules de marche au robot NAO sur la plate-forme de robot virtuel V-rep, donc notre robot est en état de marche. Ceci est réalisé au prix d'immense sacrifice, tant sur le plan intellectuel que matériels.

En effet, ce projet nous a été une expérience très fructueuse, bénéfique et enrichissante. Il nous a permis de comprendre le mécanisme qui permet au robot humanoïde d'imiter la marche humaine et apprendre à maîtriser un laboratoire virtuel qui se dispose d'une structure très symétrique à l'environnement réel. Il nous a permis également de découvrir avec passion le domaine de la robotique humanoïde et mettre le premier pas dans le monde pratique de la robotique.

Sur le plan software, ce projet nous a mis directement en contact avec des langages de programmation puissants tel que le langage PYTHON, ainsi que différents logiciels spécialisés dans ce domaine, à savoir : V-rep, Choregraphe et Linux.

Il pourrait être intéressant de retravailler sur certaines parties des mouvements de robot NAO afin de le rendre plus robuste ou d'améliorer le déplacement afin de pouvoir détecter les obstacles et les éviter.



# *Bibliographie*

## Bibliographie

- [1] : philippe COIFFET. "ROBOTS" IEEE/RSJ 1992. ICF . San Diego, CA, USA.
- [2] : Introduction à la robotique par Laetitia Maignon Université de Caen, France.
- [3] : Robotique - ISTIA, Université Angers Jean-Louis Boimond.
- [4] : Lazaric Nathalie. Origines et développement de la robotique. In: *Revue d'économie industrielle*, vol. 61, 3e trimestre 1992. pp. 54-67.
- [5]: <http://www.sciencekids.co.nz/sciencefacts/technology/historyofrobotics.html>.
- [6]: Contribution à l'étalonnage géométrique des robots industriels Par : Toufik BENTALEB  
Ingénieur d'Etat en Génie Mécanique
- [7] : I.Oestman, C-A Allared, U.Holmqvist, "Pendulum robot", reprint of Asea Journal 3.4.1985
- [8]: « Contrôle d'un robot mobile » par ABDELLAOUI Zakarya et BENKHELIFA Badreddine  
Université Abou Bekr Belkaïd Tlemcen, année 2016.
- [9] : « Etude et conception d'un Robot marchant » par HEDDOUCHE Kamel Université  
Mohamed Khider Biskra, année 2014.
- [10] : Commande d'un robot mobile Khepera IV par BOUR Djilali & CHERIEF Makhoulf  
Université Djilali BOUNAAMA Khemis Miliana, année 2016.
- [11] : Etude et conception d'un Robot marchant par HEDDOUCHE Kamel Université Mohamed  
Khider Biskra, année 2014.
- [12] : Système de localisation pour robots mobiles par SLIMANE Noureddine université de  
batna, 2005.
- [13] : [http://fr.wikipedia.org/wiki/Trois\\_lois\\_de\\_la\\_robotique](http://fr.wikipedia.org/wiki/Trois_lois_de_la_robotique).
- [14] : Bedard, "Modeling and simulation of dynamic biped Plan 9 DOF", PhD thesis. University  
Laval Quebec. 2006.
- [15] : V. Nuñez « Etude de la commande des mouvements dynamique d'un robot humanoïde ». (ITL-LISV).
- [16] : [http://www.futura-sciences.com/fr/doc/t/technologie/d/hrp2-le-robot-humanoide-franco-japonais\\_688/c3/221/p2/](http://www.futura-sciences.com/fr/doc/t/technologie/d/hrp2-le-robot-humanoide-franco-japonais_688/c3/221/p2/).



## Bibliographie

- [17]: D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The nao humanoid: a combination of performance and affordability," CoRR abs/0807.3223, 2008.
- [18]: D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of nao humanoid," in Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 769–774, IEEE, 2009.
- [19] : Article écrit par Doriane Baillarget, le Mercredi 15 Février 2017, 13:53 dans la rubrique "Autres".
- [20] : Wikipedia, "Center of mass," 2018. [https://en.wikipedia.org/w/index.php?title=Center\\_of\\_mass&oldid=826373631](https://en.wikipedia.org/w/index.php?title=Center_of_mass&oldid=826373631)
- [21]: N. Shafii, "Development of an optimized omnidirectional walk engine for humanoid robots," PhD diss, University of Porto, Portugal, 2015.
- [22]: P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure zero moment point," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 34, pp. 630–637, Sept 2004.
- [23]: N. Shafii, "Development of an optimized omnidirectional walk engine for humanoid robots," PhD diss, University of Porto, Portugal, 2015
- [24]: Souhila BACHA. « Conception et développement d'un générateur de trajectoires pour des allures de marche dynamique stable en robotique humanoïde ». Montpellier, 2010.
- [25] : Wieber Pierre-Brice, Modélisation et Commande d'un Robot Marcheur Anthropomorphe, Thèse de Doctorat, École des Mines de Paris, décembre 2000.
- [26] : Sias Fred R. & Zheng Yuan F., « How many degrees-of-freedom does a biped need?», IEEE International Workshop on Intelligent Robots and Systems, 1990, p. 297-302.
- [27]: S. Kajita, F. Kanehiro, K. Kaneko, and K. Fujiwara, "Biped walking pattern generation by using preview control of zero-moment point" in IEEE International Conference on Robotics and Automation ,ICRA 2003, 2003, pp. 1620–1626.
- [28]: K. Waldron and J. Schmiedeler, "Kinematics," Springer Handbook of Robotics, pp. 9–33, 2008
- [29]: R.M.Murray, Z.Li, andS.S.Sastry, A Mathematical Introduction to Robotic Manipulation, vol. 29. CRC Press, 1994.
- [30]: Christophe DOIGNON. « Robotique et Recalage 3-D », Université de Strasbourg, France
- [31] : Craig, J.J.: Introduction to Robotics: Mechanics and Control. 3rd edn. Pearson Hall (2004) 1, 14.

## Bibliographie

[33] : Nikolaos Kofinas .“Forward and Inverse Kinematics for the NAO Humanoid Robot”, technical university of crete, Greece. Chania, July 2012.

[34]: Corke P I. 1996 A roboticstoolbox for MATLAB IEEE Robotics& Automation Magazine pp 24-32

[35]: <http://www.coppeliarobotics.com/helpFiles/index.html>

[36]: Eric Rohmer, Surya P. N. Singh and Marc Freese «V-REP: a Versatile and Scalable Robot Simulation Framework » 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) November 3-7, 2013. Tokyo, Japan

[37]: E. Pot, J. Monceaux et al " Choregraphe: a graphical tool for humanoid robot programming". IEEE Int. Conf. Robot and Human Communication (ROMAN) 27 septembre 2009. Toyama, Japan

[38]: Emile Bahdi.” Development of a Locomotion and Balancing Strategy for Humanoid Robots”, university of Denver, 2018.

## ملخص

البحث في مجال الروبوتات البشرية لا يتوقف عن التطور في السنوات الأخيرة ، يتألف العمل المقدم في هذه المذكرة من جزأين رئيسيين.

يقدم الجزء الأول الدراسة النظرية العامة حول طريقة مشي الإنسان الآلي ذو 25 درجة من الحرية. (NAO H25 v4.0)

الجزء الثاني مكرس لتنفيذ وحدات المشي في بيئة افتراضية. من أجل تحقيق هذا الهدف ، اخترنا العمل على منصتي الروبوت الآلي

Linux و V-REP و choregraphe على

**الكلمات المفتاحية :** تنقل ذو قدمين ، نموذج هندسي ، نموذج حركي ، ZMP ، روبوت بشري ، سند بسيط ، سند مزدوج ،

## Résumé

La recherche en robotique humanoïde ne cesse d'évoluer ces dernières années, le travail présenté dans ce mémoire se compose de deux parties principales.

La première partie présente l'étude théorique globale concernant la marche d'un robot humanoïde à 25 degrés de liberté (NAO H25 v4.0).

La deuxième partie est consacrée à l'implémentation des modules de marche dans un environnement virtuel. Afin d'aboutir à nos objectifs nous avons choisi de travailler sur les deux plateformes robotique virtuelles V-REP et choregraphe sous Linux.

**Mots clés :** robot humanoïde, simple support, double support, ZMP, locomotion bipède, modèle géométrique, modèle cinématique.

## Summary

Research in humanoid robotics has been evolving in recent years, the work presented in this memoir consists of two main parts.

The first part presents the global theoretical study on the walking of a humanoid robot with 25 degrees of freedom (NAO H25 v4.0).

The second part is dedicated to the implementation of walk modules in a virtual environment. In order to achieve this goal, we have chosen to work on the two virtual robot platforms V-REP and choregraphe under Linux.

**Keywords:** humanoid robot, simple support, double support, ZMP, bipedal locomotion, geometric model, kinematic model.