



ROBOTIQUE-VISION

Thomas FIOLET
thomas.fiolet@unilasalle.fr

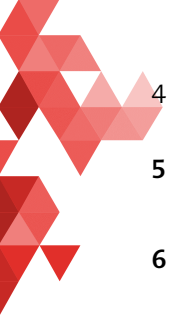
13 octobre 2021





Table des matières

I	Introduction	5
1	Contexte et portée	7
1.1	Contexte	7
1.2	Qu'est ce la robotique?	7
1.3	Qu'est-ce que la vision?	8
1.4	Portée du cours	9
1.5	Notations	9
2	Généralités	11
2.1	Exemples de systèmes de robotique et vision	11
2.1.1	Robots industriels	11
2.1.2	Robots à charges lourdes	13
2.1.3	Robots médicaux	14
2.1.4	Drones	15
2.1.5	Robots mobiles	16
2.1.6	Humanoïdes/Prothèses	17
II	Géométrie	19
3	Outils mathématiques I : repérer les objets	21
3.1	Comment repérer un objet dans le plan?	21
3.1.1	Plan, point et translation	22
3.1.2	Repère	25
3.1.3	Matrice de translation	27
3.1.4	Matrice de rotation	30
3.1.5	Matrice de transformation	33
3.2	Et dans l'espace?	35
3.2.1	L'espace	35
3.2.2	Matrice de translation	35
3.2.3	Matrice de rotation	36
3.2.4	Matrice de transformation	37
3.3	Caractérisation des rotations	38
3.3.1	Angles d'Euler	38
3.3.2	Représentation u-theta	39
4	Liaisons mécaniques	41
4.1	Robot : point de vue mécanique	41
4.2	Liaison prismatique	42
4.3	Liaison rotoïde	42
4.4	Coordonnées articulaires, coordonnées opérationnelles	42



5	Système, modèle, topologie	45
5.1	Notions	45
6	Modélisation géométrique des robots	47
6.1	Paramétrage de Denavit-Hartenberg modifié (Khalil-Kleinfinger)	47
6.1.1	Définition	47
6.1.2	Placement des repères	48
6.2	Modèle géométrique direct (MGD)	48
6.3	Modèle géométrique inverse (MGI)	49
III	Annexes	51
A	Outils mathématiques complémentaires	53
A.0.1	Rappels sur le produit matriciel	53



I Introduction





1 Contexte et portée

1.1 Contexte

Ce support de cours est le cœur des enseignements Robotique S7 et Robotique S8 enseigné à Unilasalle Amiens (anciennement ESIEE Amiens), au sein de la filière PAUC (Production Automatisée et Usine Connectée) FISE (anciennement GSP (Genie des Systèmes de Production)). Ces deux cours sont vus comme un seul qui se déroule tout au long de la quatrième année d'étude. Les cours magistraux et les travaux dirigés seront confondus.

L'objectif est, à terme, de proposer aux étudiants d'apprendre la robotique et la vision à travers de nombreux travaux pratiques et projets, accompagnés de quelques cours magistraux qui serviront à donner une base de connaissances aux étudiants. La mise en place d'un cours sous ce format est un travail long qui ne peut pas être réalisé au moment où ce cours est dispensé pour la première fois. C'est pourquoi la première version de cet enseignement prendra principalement la forme de cours magistraux qui auront pour but de donner une culture de la robotique-vision aux étudiants ingénieurs plutôt que des compétences théoriques poussées et peu applicables, comme cela est fait habituellement dans les cours de robotique. On notera aussi que ce cours ne porte pas uniquement sur la robotique mais aussi sur la vision, contrairement au nom de l'enseignement. L'utilisation de capteurs tels que les caméras et les lidars étant devenu indispensable dans la robotique, d'où le nom du support de cours : robotique-vision. L'introduction de la vision dans cet enseignement permet de s'intéresser à de nombreux aspects intéressants et modernes de la discipline : la calibration de caméra, la cartographie et localisation simultanée (ou SLAM en anglais pour Simultaneous Localization And Mapping), la robotique mobile ou encore les drones.

Il est aussi à noter que ce cours est toujours en cours d'écriture au moment où il est dispensé. Certaines parties seront donc incomplètes et tout le support peut être sujet à modification. Tout retour que les étudiants pourront faire sur le cours sera vivement apprécié¹.

1.2 Qu'est ce la robotique ?

La robotique est une discipline particulière en ceci qu'elle est pluridisciplinaire : alliance de la construction mécanique, la mécanique newtonienne, la géométrie projective, l'analyse numérique, l'algorithmique, plus récemment de la vision et pour aller plus loin : de l'automatique et de notions mathématiques plus avancées comme les groupes de Lie et les quaternions. C'est pour cela qu'elle fait partie de ces disciplines qui, grâce à son aspect interdisciplinaire, représentent bien ce que peut être le métier d'ingénieur. On peut aussi voir la robotique comme une sous-branche de la mécatronique, la mécatronique étant elle-même la fusion entre la mécanique, l'électronique, l'automatique et l'informatique.

L'aspect interdisciplinaire et les très nombreuses applications de la robotique font d'elle une discipline très difficile à définir. Comme nous le verrons dans la suite du cours, les robots industriels, les drones, les véhicules autonomes, les robots mobiles, les systèmes mécaniques équipés de systèmes de vision et bien d'autres choses rentrent dans le cadre de la robotique. De plus, donner une définition précise de la robotique serait assez peu utile en pratique.

D'un autre côté, la robotique peut être vue comme un des pendants physiques de l'informatique. Les systèmes informatiques sont extrêmement utiles, mais ils ne permettent que le traitement de l'information. La robotique permet aux algorithmes, non plus d'agir uniquement sur de l'information, mais aussi sur de la matière. L'algorithmique permet d'automatiser le traitement de l'information, la robotique permet désormais d'automatiser le traitement

1. Les retours pouvant être fait, entre autres, via l'adresse mail suivante : thomas.fiolet@unilasalle.fr

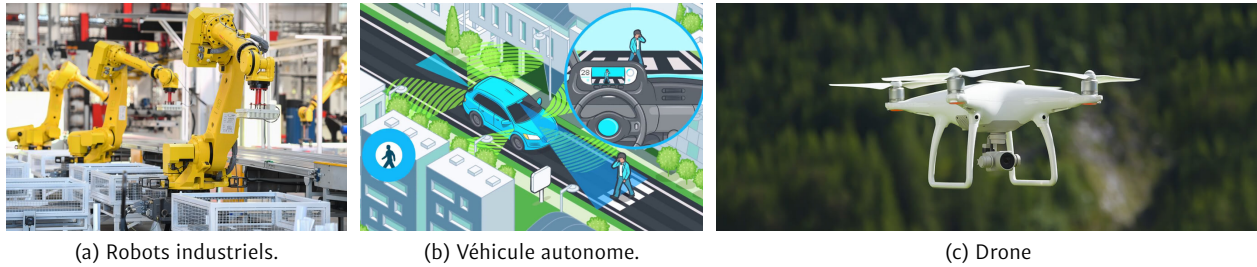


FIGURE 1.1 – Les différentes applications de la robotique. (itsocial.fr|usine-digitale.fr|orange.com)

de la matière, comme par exemple, la fabrication de pièces en série sans l'intervention de l'humain. Par ailleurs, la robotique profite de toutes les avancées modernes en informatique (puissance de calcul disponible, apprentissage automatique, etc.), qui permettent un traitement de l'information beaucoup plus complexe et beaucoup moins séquentiel qu'auparavant.

Tous les aspects de la robotique ne seront pas abordés ici. Ce sont les aspects mathématiques et algorithmiques qui seront privilégiés ici, et ce, pour plusieurs raisons :

- D'autres cours présents dans le cursus présentent les autres disciplines (comme l'électronique, la mécanique ou encore l'automatique). Ce cours peut donc se concentrer sur ce qui fait le cœur de la robotique et qui n'est pas présenté dans les autres enseignements.
- Les outils de modélisation mathématique et les algorithmes présentés ici sont assez fondamentaux, et s'appliquent à des champs beaucoup plus large que la robotique, il est donc intéressant de se pencher plus en détails sur ceux-ci.
- La plupart des missions de robotique qui sont confiées aux ingénieurs sont plutôt informatiques. Les robots étant la plupart du temps déjà présents, les aspects mécaniques et électroniques n'ont pas forcément à être pris en compte. Même si une culture de ces sujets n'est jamais de trop.

1.3 Qu'est-ce que la vision ?

La vision est un domaine tout aussi large que la robotique. Elle inclut toutes les tâches qui permettent d'extraire de l'information de données visuelles : du choix du capteur (caméra, lidar, etc.) jusqu'au traitement des données extraites. On peut aussi voir la vision comme une tentative de compréhension et d'automatisation de la vision humaine. Tout comme la robotique, elle inclut de nombreuses disciplines différentes comme : l'optique, l'électronique, l'informatique, l'analyse numérique, l'électromagnétisme jusqu'aux notions de mathématiques avancées comme les groupes de Lie. Les sous-domaines de la vision par ordinateur comprennent la détection d'événements, le suivi vidéo, la reconnaissance d'objets, l'apprentissage, l'indexation, l'estimation de mouvement, la modélisation de scènes 3D et la restauration d'image².

2. https://fr.wikipedia.org/wiki/Vision_par_ordinateur

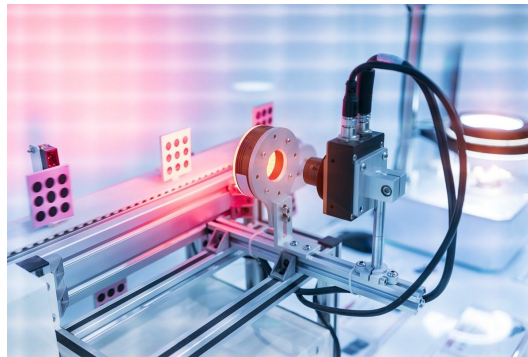


FIGURE 1.2 – Systèmes de contrôle qualité par vision (elementsindustriels.fr)

La vision est utilisée pour le contrôle qualité, la robotique, les véhicules autonomes, les drones, la réalité augmentée. Le traitement d'image, qui est un sous-domaine de la vision, est utilisé par tous les outils numériques de la vie de tous les jours : les réseaux sociaux et les applications de stockage d'images pour la reconnaissance d'images, les consoles de jeux, entre autres, pour la reconnaissance de mouvements, etc. On y retrouve beaucoup d'applications en commun avec la robotique.

À l'instar des humains, l'intérêt de posséder un système de vision réside dans la richesse des données qu'il est possible d'en extraire. Les systèmes de vision sont capables de récolter de nombreuses données qu'il est possible d'exploiter de nombreuses manières différentes, les rendants très utiles lorsqu'il s'agit d'offrir aux robots la possibilité de percevoir leurs environnements.

On notera aussi que rares sont les ingénieurs et chercheurs en robotique qui ne font pas aussi de la vision. De plus, il est aujourd'hui de plus en plus compliqué, voir impossible, de faire de la robotique sans système de vision. D'où l'inclusion de la vision dans le cours.

1.4 Portée du cours

Pour résumer ce qui est dit précédemment, l'objectif de ce cours est avant tout de donner une culture générale de la robotique aux étudiants, ainsi qu'un aperçu des concepts mathématiques sous-jacents à la discipline. Les étudiants doivent repartir du cours avec les bases nécessaires pour prendre en main des projets de robotique-vision et se former seuls. De nombreux contenus (livres, articles, vidéos) seront proposés dans la partie bibliographie.

Les outils utilisés en robotique (les matrices de transformation) reposent sur des notions de géométrie projectives. Or, il serait trop long, trop compliqué et inutile de présenter la géométrie projective rigoureusement. Seules les notations seront conservées, en ajoutant une composante aux vecteurs et aux points (égale à 0 pour les vecteurs et égale à 1 pour les points). Une explication intuitive de ce que représentent ces composantes supplémentaires, de comment les interpréter, sera par contre donnée.

Bien que les robots étudiés dans ce cours seront principalement des bras robots, les outils mathématiques étant très adaptés à ces derniers, les nombreuses autres applications de la robotique-vision seront abordées.

Il est à noter que ce support représente la première étape de la construction d'un enseignement bien plus ambitieux. Un enseignement capable de donner à chaque ingénieur les compétences dont il a besoin : d'une simple culture générale sur la robotique-vision, jusqu'aux connaissances théoriques les plus pointues, en passant par des compétences pratiques et utilisables immédiatement.

1.5 Notations

Les notations utilisées dans ce cours peuvent être déroutantes pour les non connaisseurs de la robotique, elles varient aussi d'un cours à l'autre, d'un professeur à l'autre. L'objectif de cette section est donc présenter les nota-

tions qui seront utilisées tout au long de ce cours. Un effort particulier sera fourni afin de proposer un cours cohérent.

- **Point** : les points du plan et de l'espace seront notés avec des lettres minuscules comme p . Les coordonnées d'un point peuvent changer d'un repère à l'autre, le point p dont les coordonnées sont exprimées dans le repère R_i sera noté p^i . Les points sont notés comme les vecteurs, en colonne. On les distingue de ces derniers en ajoutant une composante supplémentaire qui vaut 1. Par exemple :

$$p^i = \begin{pmatrix} 1.5 \\ 1 \\ 2 \\ 1 \end{pmatrix} \quad (1.1)$$

Ce point se trouve dans un espace à trois dimensions. La dernière composante indique simplement que c'est un point (elle trouvera une autre utilité dans la suite du cours). Il est parfois utile d'écrire les points directement dans des textes, on les écrira alors en ligne en indiquant qu'ils sont transposés en ajoutant un t en exposant : $p^i = (1.5 \ 1 \ 2 \ 1)^t$

- **Vecteur** : les vecteurs sont notés comme les points, à deux exceptions près : ils sont notés en gras et la composante supplémentaire vaut 0 :

$$\mathbf{u}^i = \begin{pmatrix} 3 \\ -1.2 \\ 1.7 \\ 0 \end{pmatrix} \quad (1.2)$$

Les composantes des vecteurs seront notées entre crochet, pour se rapprocher de la notation utilisée en programmation (en C++ notamment) :

$$\mathbf{u}^i[x] = 3 \quad (1.3)$$

La dernière équation représente donc la composante en x du vecteur $\mathbf{u}$. La composante étant exprimée selon l'axe x^i .

Certains vecteurs forment une base de l'espace, il se verront attribuer un indice x^j , y^j ou z^j pour savoir quel axe ils portent :

$$\mathbf{u}_{y_j}^i \quad (1.4)$$

Ce vecteur est donc celui qui porte l'axe y du repère R_j , exprimé dans le repère R_i .

- **Matrices et matrices de transformation** : les matrices seront notées avec des lettres majuscules. Les matrices de transformation représentent des repères, elles seront notées avec la lettre T , et posséderont un indice et un exposant qui permettront d'identifier le repère correspondant ainsi que le repère considéré comme canonique. Par exemple, la matrice associée au repère R_i dont les éléments sont exprimés dans R_j sera notée T_i^j .

2 Généralités

Comme énoncé précédemment, la vision et la robotique sont deux domaines très compliqués à définir précisément et rigoureusement. C'est pourquoi la prochaine section de cours se concentrera sur de nombreux exemples de systèmes. Ce qui permettra, à défaut de donner des définitions précises, de donner une vue d'ensemble de la robotique-vision.

2.1 Exemples de systèmes de robotique et vision

Comme dit précédemment, les robots peuvent être très différents les uns des autres. Du bras robot industriel six axes jusqu'à des modèles beaucoup plus exotiques. L'objectif de cette première section est de mettre à jour toute la diversité, que ce soit dans les types de modèles, les applications, ou encore les marques, qui existent sur le marché.

2.1.1 Robots industriels

Les robots six axes sont en un sens les plus classiques des robots. Ce sont ceux qui sont présents au laboratoire **Production Automatisée et Usine Connectée** du Campus d'Amiens. Ils sont très versatiles, c'est pour cela qu'on les retrouve beaucoup dans l'industrie. Les bras robots de ce type peuvent travailler sur six degrés de liberté, ce qui leur permet d'atteindre un grand nombre de positions différentes. Ils peuvent aussi bien servir à faire de l'assemblage que de la peinture ou encore de l'usinage.

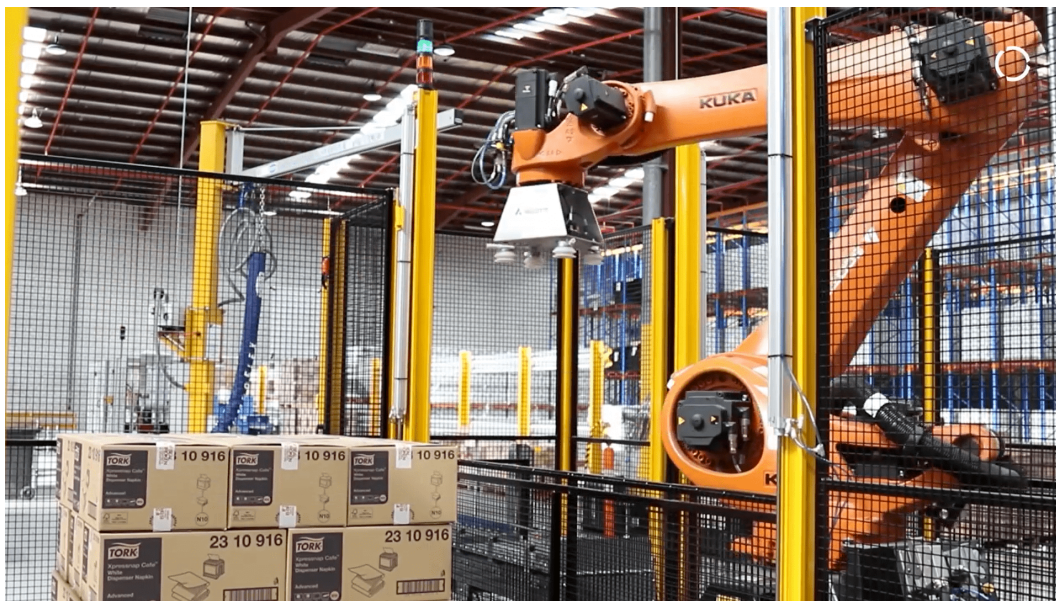


FIGURE 2.1 – Un robot de palettisation Kuka (► Productive and flexible way of palletizing consumer goods | KUKA - Robots & Automation).

La figure 2.1 montre un robot de palettisation de marque Kuka. Sa tâche est simple, le robot saisit les cartons provenant de la chaîne de production pour les empiler sur des palettes. Attribuer cette tâche à un robot permet

plusieurs choses : éviter la fatigue et les accidents du travail (les cartons étant assez lourds à transporter) et permet un gain de temps (le robot peut être plus efficace sur ce type de tâche). Bien que la palettisation soit simple à définir, il y a en fait quelques subtilités à prendre en compte. Le robot doit faire preuve de précision, il ne doit pas bousculer les cartons déjà entreposés. Au lieu de se placer directement au-dessus de la position finale du carton puis de descendre tout droit vers cette dernière, il vaut mieux descendre "en diagonale", afin de bien caler la boîte.



FIGURE 2.2 – Des robots industriels Fanuc chez Audi (📺 FANUC Industrial Robots at AUDI | FANUC Europe).

La figure 2.2 montre l'usine Audi en Allemagne, où toute une flotte de robot Fanuc fabriquent les voitures, beaucoup de tâches y sont automatisées, de l'assemblage à la peinture. C'est une tendance très forte dans l'industrie automobile de pousser très loin l'automatisation des tâches. La demande étant encore très forte, certaines usines devant sortir plus d'une voiture par minute.

2.1.2 Robots à charges lourdes

Certains robots sont conçus pour soulever des charges lourdes. C'est par exemple le cas du Fanuc M-2000iA/1700L présent sur la figure 2.3, qui peut porter jusqu'à 1700 kg. La difficulté pour concevoir de tels robots est ici mécanique, la poignée du robot doit être capable de supporter des couples très importants. On notera aussi la présence d'un contrepoids.

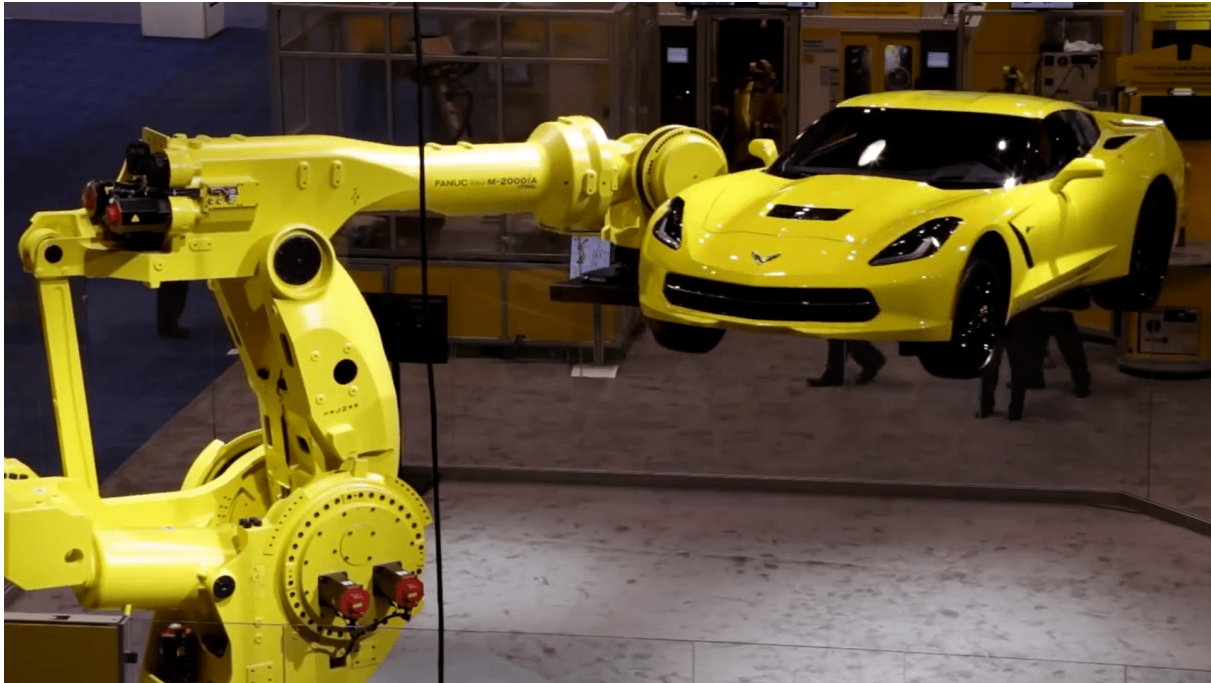


FIGURE 2.3 – Un Fanuc M-2000iA/1700L soulevant une Chevrolet Corvette C7 pesant 1539 kg à vide (► New FANUC M-2000iA/1700L Robot Lifts Corvette Demonstrating Long Reach & Heavy Payload | FANUC America Corporation).

Les robots Fanuc de l'exemple précédent, avec leur capacité à soulever plusieurs centaines de kilos, peuvent rentrer dans cette catégorie.

2.1.3 Robots médicaux

Un secteur en plein essor est celui de l'assistance chirurgicale. De nombreux développements sont en cours dans ce domaine. Une des plateformes les plus connues est ROSA par Stäubli (voir 2.4). Cette plateforme assiste le chirurgien pendant son intervention en guidant son geste. Le robot calcule le mouvement à effectuer à partir de l'IRM du patient et des instructions du chirurgien. Il est capable de calculer sa position relativement au patient.

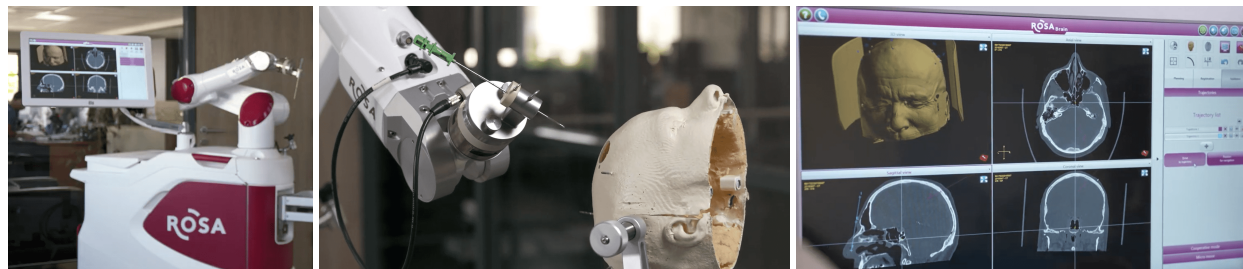


FIGURE 2.4 – Plateforme robotique d'assistance chirurgicale ROSA par Stäubli (YouTube First robotic platform for surgical assistance | RoboticsStaubli | 2020).

[14] montre que les performances varient beaucoup entre les chirurgiens. Plus précisément, la gravité des complications post-opératoires varie beaucoup d'un chirurgien à l'autre. L'utilisation des technologies numériques permet de lisser ces différences de performances et de maximiser les chances de tous les patients.

Les figures suivantes montrent les nombreux robots présents dans un laboratoire de biochimie. On y voit notamment des robots cartésiens et des scara. Les procédures d'analyse dans ces laboratoires sont souvent longues et demandent une grande précision et une grande précaution lors de la manipulation des échantillons.

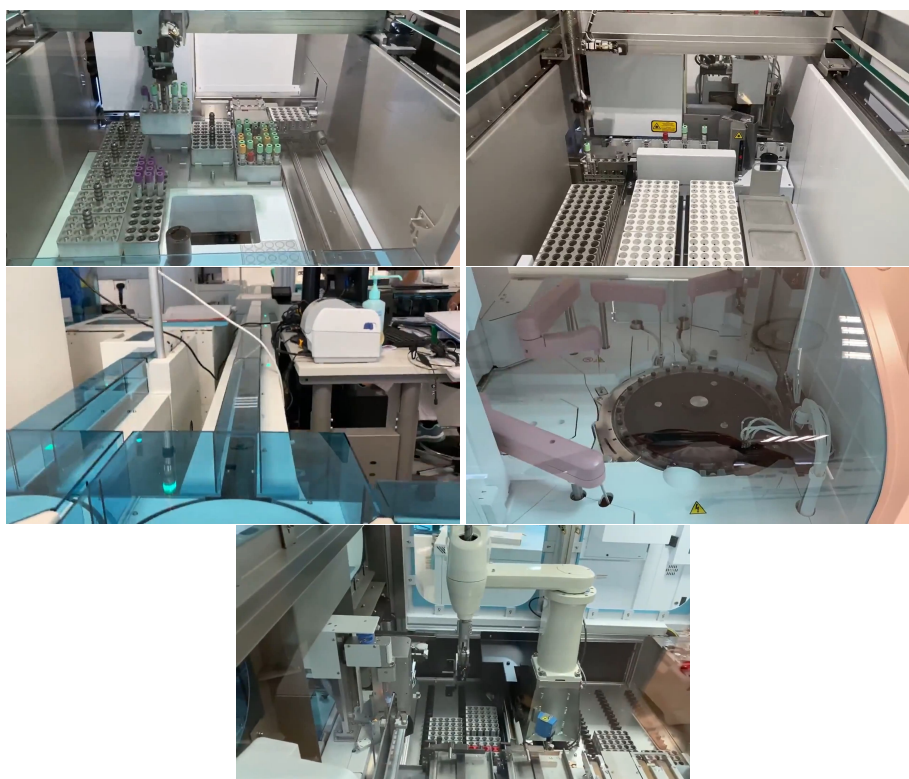


FIGURE 2.5 – Laboratoire de biochimie (Twitter Thread sur la visite d'un laboratoire de biochimie | 2021).

2.1.4 Drones

Les drones font partie intégrante de la robotique d'aujourd'hui, ce sont surtout les outils d'automatique et de vision qui sont utilisés dans cette discipline. Les drones étant par nature instables et non localisés (ils ne sont pas fixés à un endroit précis comme les robots industriels), le défi est plutôt de les contrôler et de les localiser dans leur environnement. La figure 2.6 montre un drone conçu par l'UZH Robotics and Perception Group transportant du liquide dans une canette ou une bouteille grâce à une commande prédictive non-linéaire [25] [24].

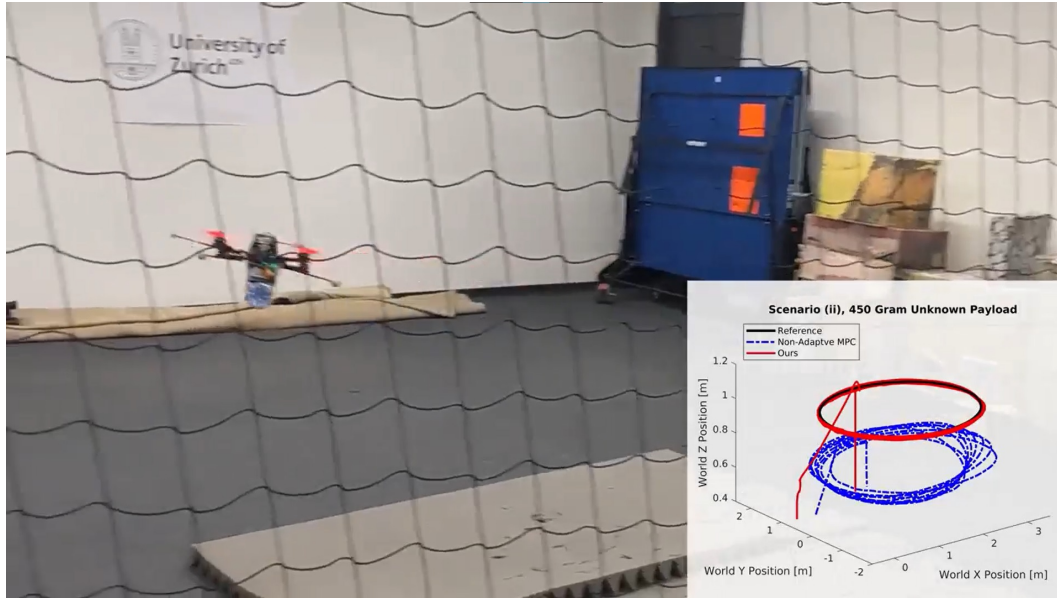


FIGURE 2.6 – Un quadrotor transportant des bouteilles et des canettes en utilisant la commande prédictive (in Agile quadrotor under uncertainty - Davide SCARAMUZZA | 2021).

Les chercheurs de l'UZH Robotics and Perception Group proposent aussi une technologie permettant aux drones d'atterrir sur une cible mouvement, sans connaître sa position à l'avance (voir 2.7).

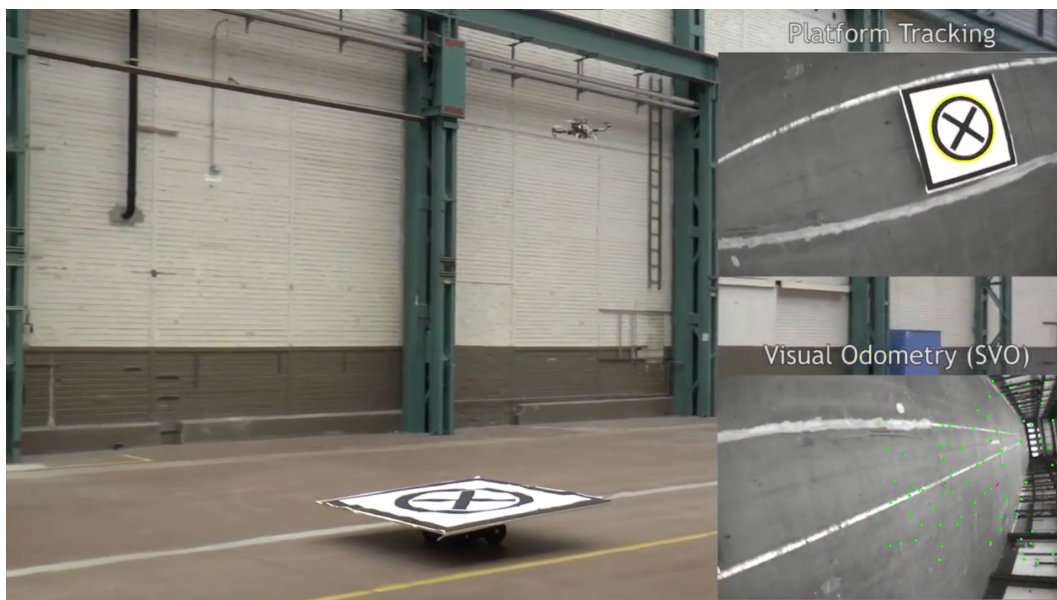


FIGURE 2.7 – Un drone capable d'atterrir sur une cible en mouvement (Vision based Autonomous Quadrotor Landing on a Moving Platform - UZH Robotics and Perception Group | 2017).

2.1.5 Robots mobiles

La robotique mobile est une part non-négligeable de la robotique-vision moderne, tout spécialement avec le développement fort des véhicules autonomes. À l'instar des drones, les robots mobiles demandent surtout des compétences en automatique et en vision.



FIGURE 2.8 – Véhicule Tesla complètement autonome sur route (▶ Full Self-Driving - Tesla | 2019).

Le SLAM (Simultaneous Localization and Mapping), ou encore Localisation et Cartographie Simultanée est une technologie très utilisée. Elle permet à l'agent de créer une carte de son environnement au fur et à mesure que les données sont récoltées, tout en situant l'agent dans cette carte. La figure 2.9 montre l'exécution de l'algorithme LDSO (Direct Sparse Odometry with Loop-Closure), un algorithme de SLAM.



FIGURE 2.9 – Execution de l'algorithme LDSO sur le dataset Kitty (▶ LDSO : Direct Sparse Odometry with Loop Closure - Gao, Wang, Demmel, Cremers | 2018).

2.1.6 Humanoïdes/Prothèses

"Hugh Herr (voir figure 2.10), né le 25 octobre 1964 à Lancaster (Pennsylvanie), est un grimpeur, ingénieur et biophysicien américain. Grimpeur de haut niveau, il doit être amputé de ses deux jambes au niveau du genou à la suite d'une mésaventure lors d'une ascension. Il cherche malgré tout à continuer à escalader et crée des jambes prothétiques lui permettant d'escalader à nouveau au plus haut niveau. Il travaille toujours dans la recherche dans le secteur de la biophysique au sein du MIT Media Lab."¹



FIGURE 2.10 – Hugh Herr et ses prothèses de jambes (▶ Hugh Herr, l'homme « bionique » - FUTUREMAG - ARTE | 2016).

La figure 2.11 montre la dernière génération de robots Atlas de la société Boston Dynamics. Atlas est capable de se déplacer dans des environnements complexes.

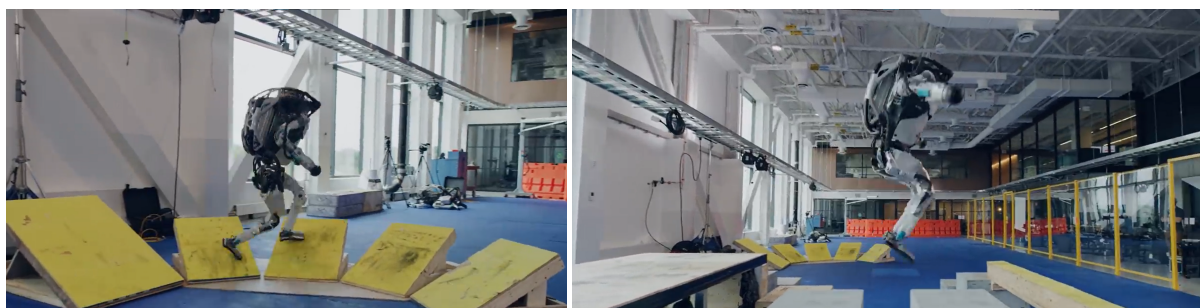


FIGURE 2.11 – Robot Atlas de Boston Dynamics (▶ Atlas | Partners in Parkour | 2021).

1. https://fr.wikipedia.org/wiki/Hugh_Herr





II Géométrie



3 Outils mathématiques I : repérer les objets

3.1 Comment repérer un objet dans le plan ?

Pratiquer la robotique demande de ne se poser au final qu'une seule question : où ? Où se trouve l'effecteur du robot ? Où se trouve une liaison particulière du robot ? Où doit aller la pince attachée au robot ? Par où doit-elle passer ? Où se trouvent les obstacles qui pourraient gêner le robot ? Il faut pouvoir répondre à ces questions de manière précise et quantitative. Il existe pour cela des outils mathématiques puissants qui vont être présentés dans ce premier chapitre. Les outils seront introduits en deux dimensions puis adaptés en trois dimensions. Les concepts sous-jacents étant bien plus abordables en deux qu'en trois dimensions, tout en restant exactement les mêmes. L'étude de ces outils mathématiques demande par contre quelques capacités d'abstraction, l'objectif étant d'extraire l'information minimale requise pour décrire le système étudié.

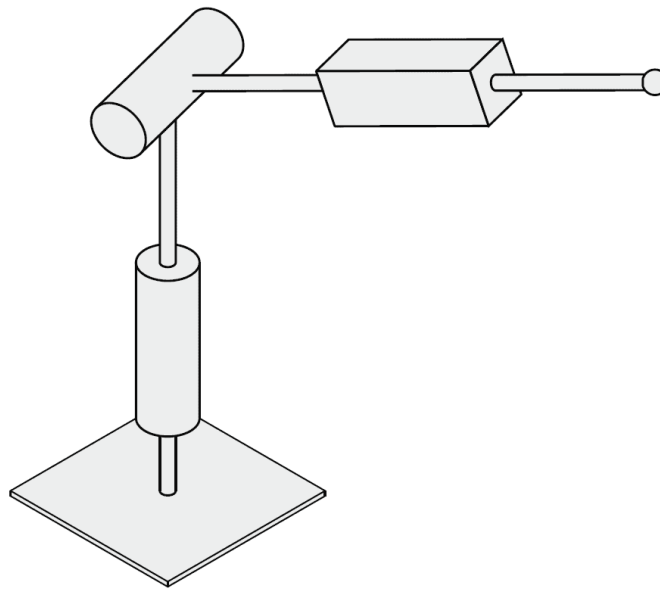


FIGURE 3.1 – Schéma d'un robot RRT (Rotation-Rotation-Translation). La question centrale en robotique est de savoir où se situe, dans l'espace, les différentes parties du robot par rapport, par exemple, à sa base (📖 Robotics - T. Bajd et al. - 2010).

Le caractère abstrait de ces outils fait aussi leur force, ils peuvent être utilisés dans de nombreux autres domaines : modéliser des caméras pour prédire la position des objets filmés dans l'image, quantifier la position des objets dans un jeu vidéo et placer les caméras aux bons endroits, réaliser des simulations physiques, etc...

Dans la suite de ce cours, le système physique sera réduit au modèle mathématique que nous manipulerons.

3.1.1 Plan, point et translation

Les premiers outils utilisés forment plus un cadre abstrait, presque universel, plutôt que des méthodes de travail spécifiques. Les sections suivantes présenteront la base de ce cadre, en repartant de zéro, pour construire étape par étape les outils finaux.

Définition

La scène sur laquelle tous les événements prennent place sera ici le plan $\mathbb{R}^2$. Il est constitué d'un ensemble infini et non-dénombrable¹ de points, chacun étant décrit par deux nombres réels. C'est le nombre de paramètres libres, ici les nombres réels qui décrivent chaque point, qui donne sa dimension au plan. A titre de comparaison, l'espace $\mathbb{R}^3$, est un espace à trois dimensions, car les points de l'espace sont entièrement définis par trois nombres réels. Le nombre de dimensions est aussi un nombre de degrés de liberté. Les deux coordonnées d'un point sont indépendantes l'une de l'autre².

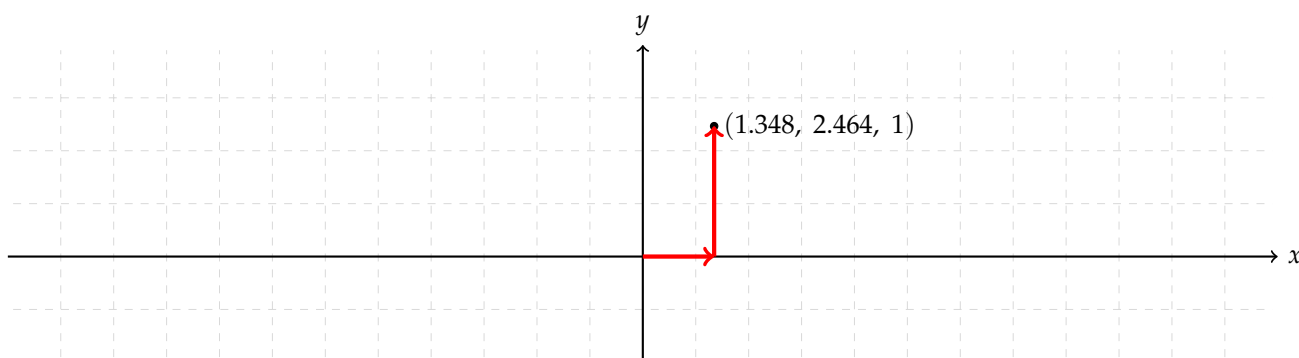


FIGURE 3.2 – Représentation graphique du plan $\mathbb{R}^2$. Un point est sélectionné sur le plan. Les coordonnées de ce point représentent les déplacements qu'il faut effectuer depuis l'origine et le long des deux directions x et y pour l'atteindre.

L'outil abstrait qu'est le plan peut paraître très simple aux premiers abords, mais il permet déjà de repérer les objets de manière précise et quantifiée. Mais avant de pouvoir repérer des objets, un point de référence, l'origine, doit être sélectionné parmi tous les points disponibles. Ce point est arbitraire, il faut donc le sélectionner soigneusement (dans le cas d'utilisation d'un robot, la base du robot est souvent un choix judicieux). Les axes x et y , eux, définissent les directions avec lesquelles on travaille. Tout comme l'origine, ces directions sont arbitraires, il faut donc les choisir soigneusement. Bien heureusement, des méthodes existent pour choisir une origine et des axes adéquats (voir le chapitre sur le paramétrage de Denavit-Hartenberg modifié).

Repérer un objet revient alors à indiquer les déplacements à effectuer depuis l'origine et le long des deux dimensions pour l'atteindre. Considérons un premier exemple (voir figure 3.2) : un objet se trouve au point $(x, y, 1)^t = (1.348, 2.464, 1)^t$ ³. Lorsque l'on dit qu'un point est situé en $(x, y, 1)^t = (1.348, 2.464, 1)^t$, cela veut dire qu'il faut se déplacer de 1.348 *unités* selon l'axe x et de 2.464 *unités* selon l'axe y .

Le plan présenté au-dessus permet aussi de repérer des objets non-ponctuels (qui ont un volume non nul, donc qui occupent plusieurs points dans le plan), qui vont occuper un sous-ensemble du plan.

1. Il y a toujours une infinité de points entre deux points du plan.

2. Un objet du plan peut en fait avoir plus que deux degrés de liberté, tout comme il est possible de le contraindre à un seul degré de liberté.

3. On rappelle que, pour des raisons qui seront explicitées dans la suite, les points se voient attribuer une composante supplémentaire valant 1

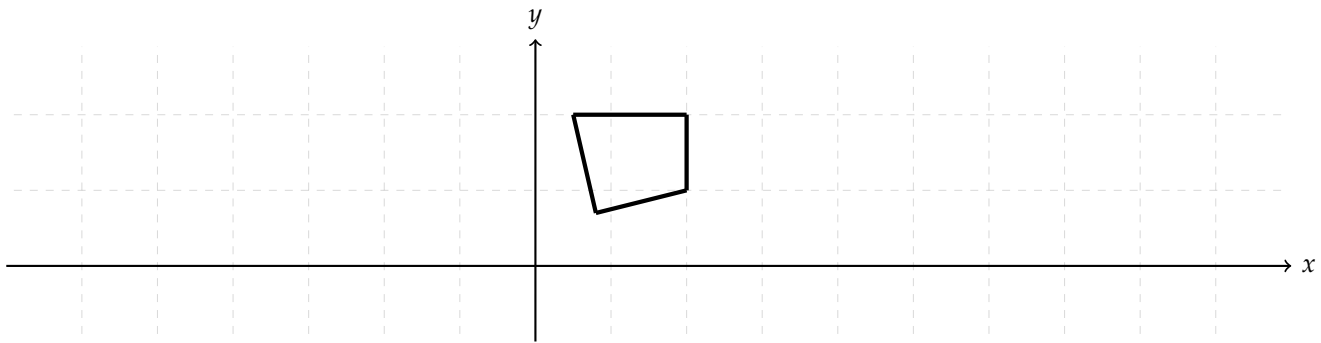


FIGURE 3.3 – Exemple de représentation d'un objet avec un volume non nul.

Bien qu'il soit souvent pratique de représenter les objets par des points ponctuels, souvent placés au niveau de leur centre de gravité, aucun objet n'est un point ponctuel, même pas les particules élémentaires qui forment les fondations de notre réalité (jusqu'à preuve du contraire!).

Distances

Il est possible, directement à partir des coordonnées des points, de mesurer des distances entre ceux-ci. La distance n'est caractérisée que par un seul nombre, elle permet, en un sens, de compacter l'information contenue dans deux points. Il existe beaucoup de définitions différentes pour la distance⁴, la plus connue et la plus utilisée est la distance euclidienne. Elle correspond à la distance parcourue par un agent qui suit un segment dont les extrémités sont les deux points d'intérêt. La procédure pour calculer la distance entre deux points est très simple, elle est présentée ici en pseudo-code :

Algorithm 1 Euclidean distance

Require : $p_1, p_2 \in \mathbb{R}^2$

return $\text{sqrt}((p_1[x] - p_2[x])^2 + (p_1[y] - p_2[y])^2)$

Et elle est présentée ci-dessous dans différents langages de programmation (C++, Python et Matlab).

Listing 3.1 – Distance euclidienne, C++

```
1 #include <math.h>
2
3 real euclideanDistance(p1, p2){
4     return sqrt(pow((p1.x - p2.x),2) + pow((p1.y - p2.y),2))
5 }
```

Listing 3.2 – Distance euclidienne, Python

```
1 from math import *
2
3 def euclideanDistance(p1, p2)
4     return sqrt((p1.x - p2.x)**2 + (p1.y - p2.y)**2)
```

Listing 3.3 – Distance euclidienne, Matlab

```
1 function[d] = euclideanDistance(p1, p2)
2     d = sqrt((p1(1) - p2(1)).^2 + (p1(2) - p2(2)).^2);
3 end
```

La figure 3.4 donne une représentation graphique de la distance euclidienne.

4. Choisir une définition différente donnera d'ailleurs des propriétés différentes au plan.

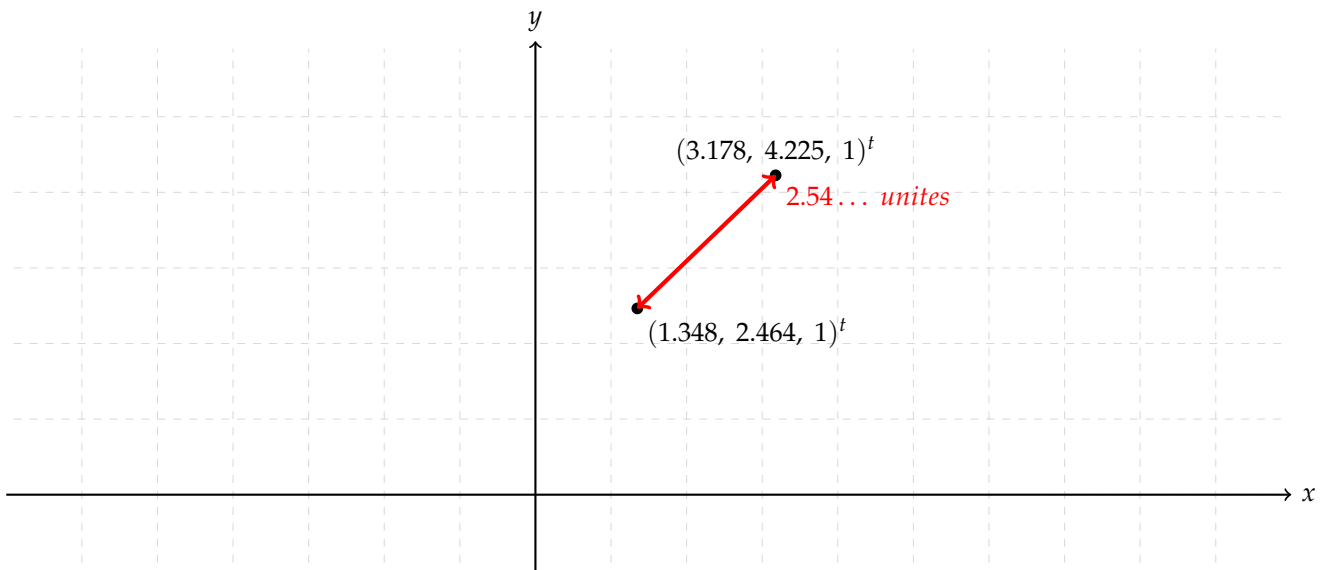


FIGURE 3.4 – Distance euclidienne entre deux points.

On notera que dans beaucoup de cas pratiques, ce sont les mesures qui permettent d'obtenir les coordonnées des points d'intérêts, souvent avec de l'incertitude...

Les deux autres distances les plus notables, et qu'il peut être utile de connaître, sont les distances de Manhattan et de Tchebychev, encore une fois illustrée en C++, Python et Matlab ci-dessous.

Algorithm 2 Manhattan distance

Require : $p_1, p_2 \in \mathbb{R}^2$

return $abs(p_1[x] - p_2[x]) + abs(p_1[y] - p_2[y])$

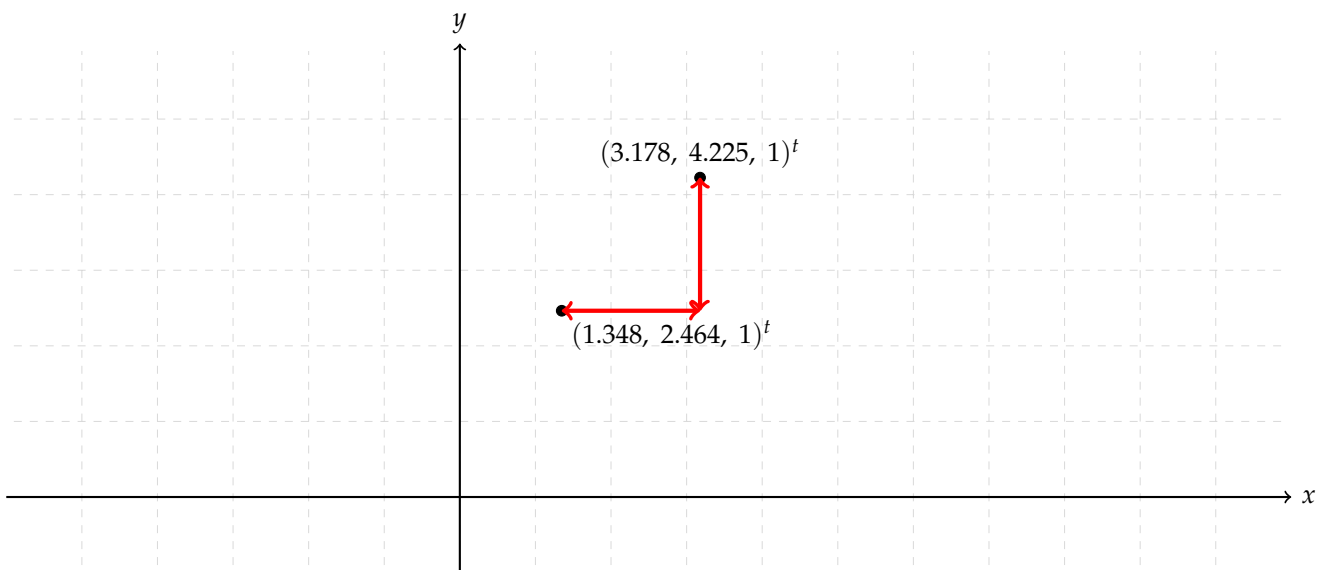


FIGURE 3.5 – Distance de Manhattan entre deux points.

Le nom de cette distance vient du fait qu'elle permet de mesurer la distance à parcourir pour aller d'un endroit à un autre dans une ville comme Manhattan, constituée de routes droites qui se croisent en formant des angles droits. Là où la distance Euclidienne représente la distance "à vol d'oiseau".

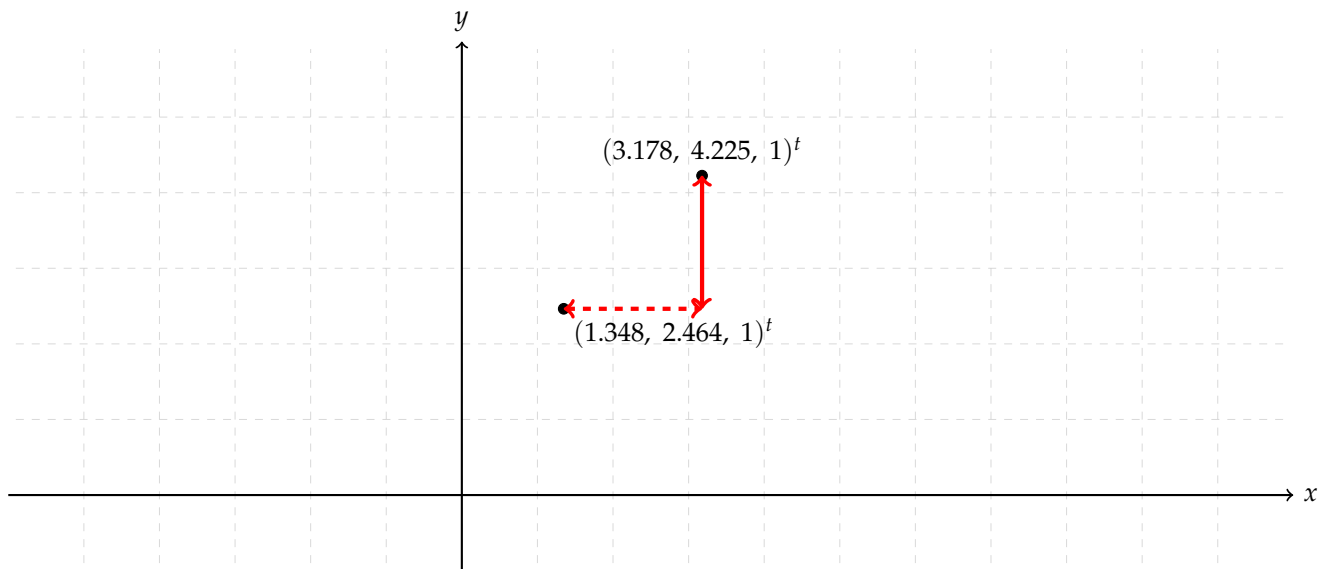
Algorithm 3 Tchebychev distance**Require :** $p_1, p_2 \in \mathbb{R}^2$ **return** $\sup_{i=x,y,z} |p_1[i] - p_2[i]|$ 

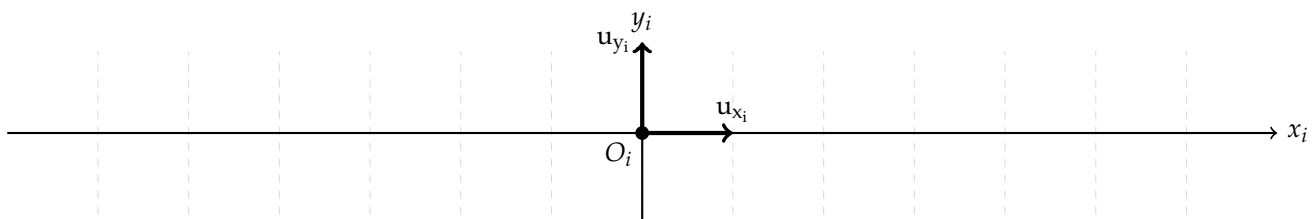
FIGURE 3.6 – Distance de Tchebychev entre deux points.

La distance de Tchebychev sélectionne la dimension qui renvoie la plus grande distance.

Calculer des distances permet de créer, par exemple, des zones interdites autour d'un point. On peut interdire à un objet de se trouver à moins d'une certaine distance d'un point. En changeant la façon de mesurer les distances, on change la forme de la zone interdite. Avec la distance euclidienne, la zone interdite est un disque, avec les distances de Manhattan et de Tchebychev, la zone interdite est un carré.

3.1.2 Repère

Dans la section précédente, différents objets mathématiques ont été introduits, notamment l'origine et les axes, la notion de repère tente de tous les rassembler. Un repère sera ici défini comme un ensemble composé d'un point, qui représente l'origine du plan et de deux vecteurs, qui donne la direction des deux axes ainsi que la taille de l'unité pour chaque dimension. Dans ce cours, les repères seront toujours orthonormés, les vecteurs de chaque repère seront orthogonaux les uns aux autres et unitaires (leurs normes vaudront toujours 1).

FIGURE 3.7 – $\mathbb{R}^2$ muni du repère orthonormé canonique.

L'exemple donné sur la figure 3.7 est celui du repère canonique où $O_i = (0, 0, 1)^t$, $u_{x_i} = (1, 0, 0)^t$ et $u_{y_i} = (0, 1, 0)^t$ (on rappelle ici que l'exemple est en deux dimensions, la composante supplémentaire prendra son sens dans les sections suivantes). Il est tout à fait possible de travailler avec un repère qui n'est pas canonique, mais l'existence d'un repère non-canonique implique l'existence d'un repère canonique, dont les vecteurs sont alignés aux axes, de

norme 1, orthogonaux entre eux et dont l'origine est le point $(0,0,1)$. La figure 3.8 montre un repère non-canonique. Ce dernier est tourné et translaté par rapport à l'origine.

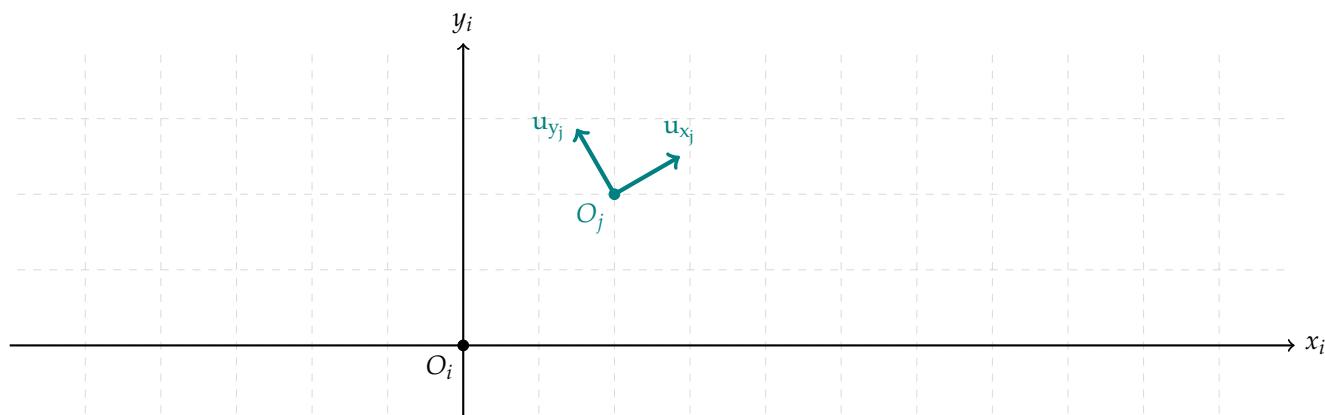


FIGURE 3.8 – $\mathbb{R}^2$ muni du repère orthonormé non canonique.

Pour conclure, on remarquera que le plan n'est qu'un ensemble infini de point, sans référence à laquelle se rattacher pour placer un repère, et en particulier le repère canonique. Le choix de son placement et de son orientation est donc complètement arbitraire. Ce qui nous donne l'avantage de pouvoir placer l'origine et l'orientation des axes comme on le souhaite. C'est aussi ce qui nous permet de travailler relativement à n'importe quel repère, et de changer de point de vue autant de fois que nécessaire. Il est tout à fait possible de travailler avec un repère R_i , canonique, et un repère R_j , qui lui ne l'est pas, et de changer de point de vue pour considérer R_j comme le repère canonique et R_i comme ne l'étant pas.

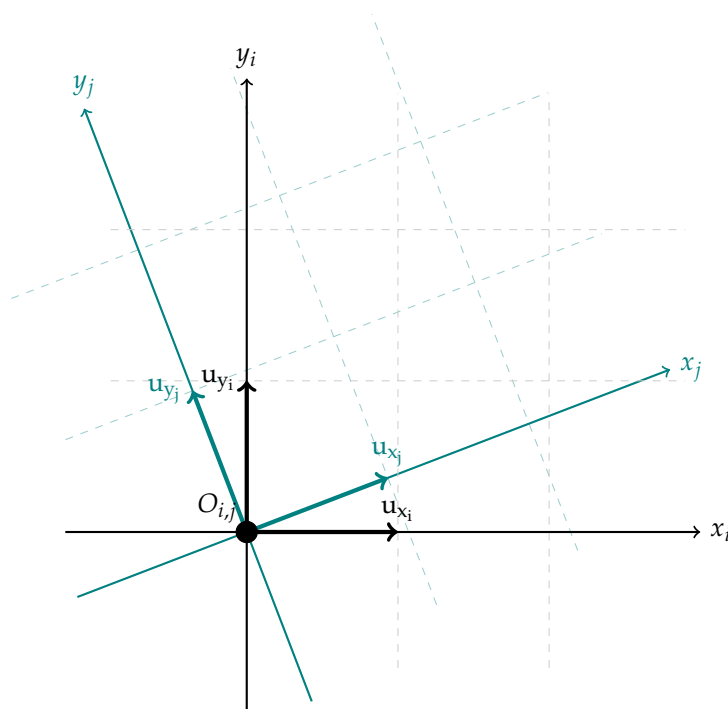


FIGURE 3.9 – Illustration de la relativité de deux repères tournés l'un par rapport à l'autre.

3.1.3 Matrice de translation

Les conclusions établies dans les sections précédentes vont nous permettre de faire beaucoup de choses. En effet, il est souvent pratique de placer un repère de travail au niveau de l'objet d'étude, pour repérer les objets localement, puis de changer de point de vue pour repérer les objets par rapport à un repère plus générale. On peut par exemple considérer un entrepôt dans lequel des robots mobiles s'occupent de transporter des objets. Lorsqu'un robot manipule un objet, il est beaucoup plus utile de travailler avec un repère local, attaché au robot. Ce qui nous intéresse ici étant la position de l'objet par rapport au robot. Mais lorsque le robot a terminé sa manipulation, il est plus utile de repasser dans un repère global attaché à l'entrepôt⁵.

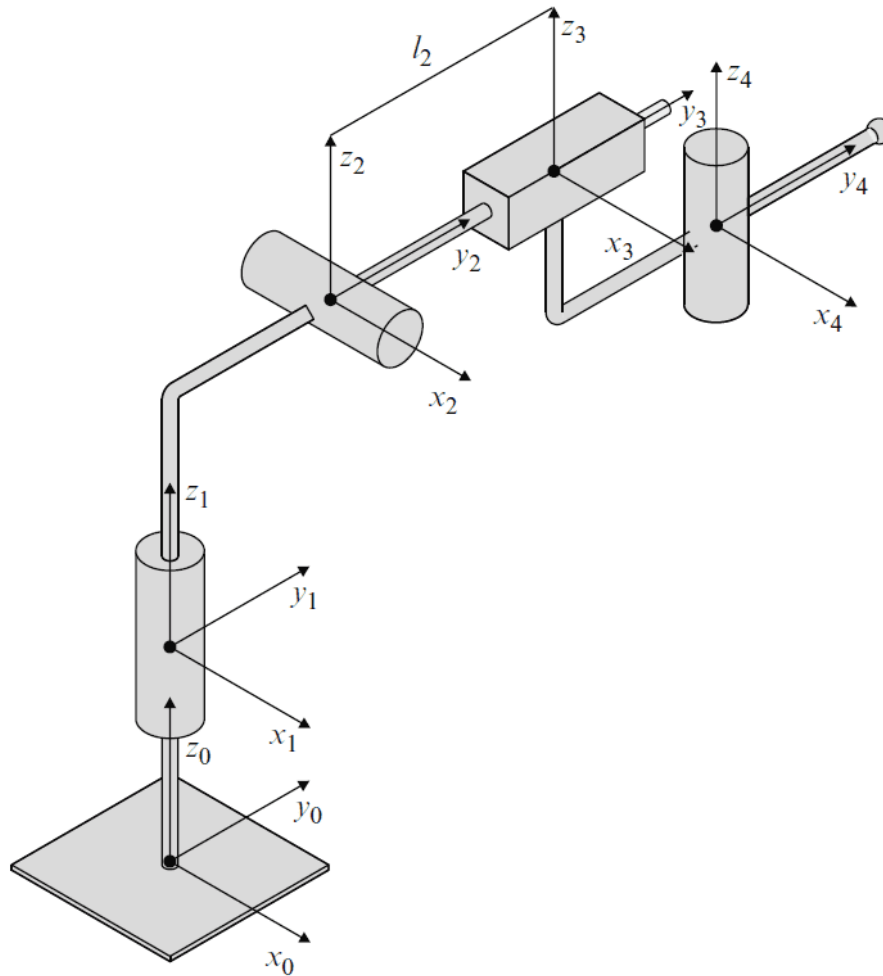
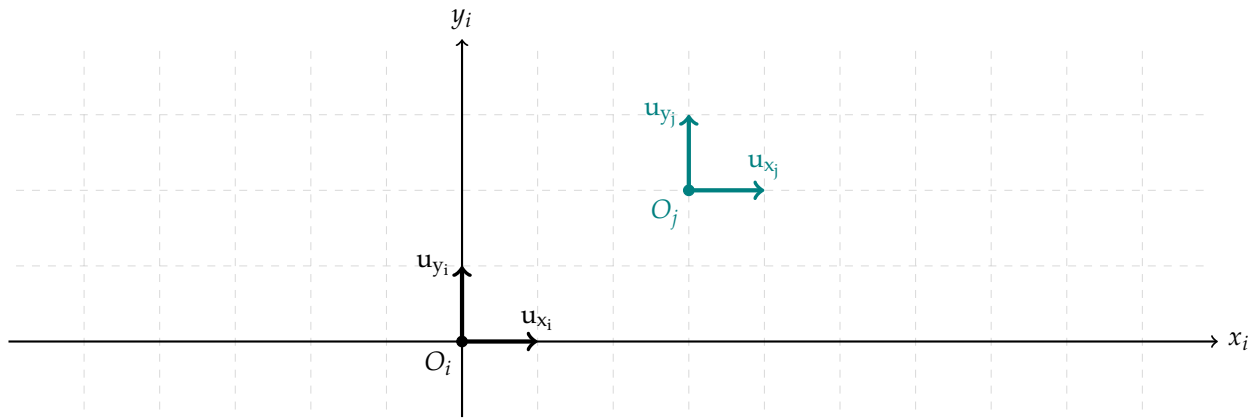


FIGURE 3.10 – Schéma d'un robot RRTR (Rotation-Rotation-Translation-Rotation) sur lequel sont placés cinq repères (Robotics – T. Bajd et al. – 2010).

La figure 3.10 montre le schéma d'un robot RRTR sur lequel sont placés cinq repères. Le repère R_0 peut être considéré comme le repère global ici, il est attaché à la base du robot. Le repère R_4 permet de travailler localement. On peut imaginer qu'une caméra soit embarquée sur le robot au niveau du repère R_4 . Les objets pourront être localisés, mais uniquement par rapport au repère R_4 , il faudra alors changer de repère.

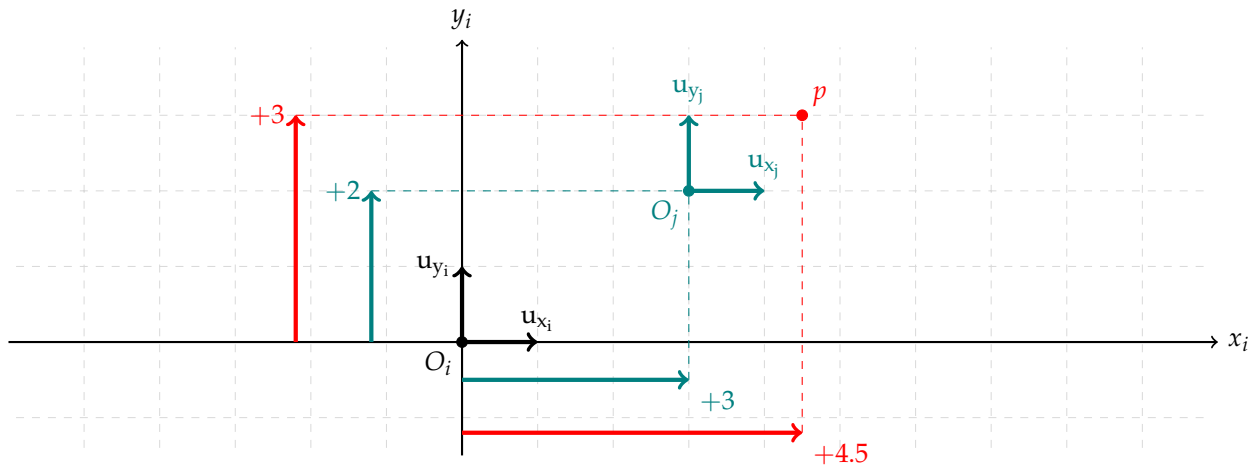
Repassons dans le plan et considérons un exemple simple (donné par la figure 3.11).

5. On notera que, mathématiquement parlant, les repères ne sont ni locaux ni globaux. Un repère est considéré ici comme local ou global en fonction de ce à quoi on l'attache.

FIGURE 3.11 – $\mathbb{R}^2$ muni de deux repères traduits l'un par rapport à l'autre.

Passer d'un repère à l'autre permet, certes, de simplifier souvent le problème considéré, mais il faut pour ça pouvoir exprimer les coordonnées d'un point relativement à chaque repère. C'est-à-dire ne plus considérer que le point de coordonnées $(0, 0, 1)^t$ n'est plus en O_i mais bien en O_j . Et que c'est le point O_i qui n'est plus à l'origine du plan.

Travaillons dans un premier temps à partir de l'exemple de la figure 3.12. L'origine du repère R_i est placée en $O_i^i = (0, 0, 1)^t$; l'origine du repère R_j est placée en $O_j^i = (3, 2, 1)^t$ (l'exposant indique que les coordonnées des origines sont exprimées dans le repère R_i).

FIGURE 3.12 – $\mathbb{R}^2$ muni de deux repères traduits l'un par rapport à l'autre.

Le point p^i (la notation en exposant i indique que les coordonnées du point p sont exprimées dans R_i) est placé aux coordonnées $(4.5, 3, 1)^t$. On remarque immédiatement que le point p , mais dont les coordonnées sont exprimées dans R_j est alors : $p^j = (1.5, 1, 1)^t$. Il est aussi possible de changer le repère dans lequel les coordonnées des origines des repères sont exprimées. On obtient alors $O_j^j = (0, 0, 1)^t$ et $O_i^j = (-3, -2, 1)^t$. La question qu'il faut se poser maintenant est : quelle est la relation entre toutes ces grandeurs, la position de p exprimée dans différents repères, et la position relative des repères entre eux. En ne tenant pas compte de la dernière composante, il est possible, grâce à des sommes vectorielles (les points étant ici considérés comme des vecteurs, se distinguant de ces derniers par la coordonnée supplémentaire), de passer d'un repère à l'autre :

$$\begin{pmatrix} 4.5 \\ 3 \\ 1 \end{pmatrix} + \begin{pmatrix} -3 \\ -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1 \\ 1 \end{pmatrix} \quad (3.1)$$

L'équation 3.1 peut être ré-écrite sous la forme suivante :

$$p^i + O_i^j = p^j \quad (3.2)$$

On obtient ici une première relation entre les coordonnées d'un point dans un repère et ses coordonnées dans un autre! Il faut pour cela connaître la position relative des deux repères. On remarquera aussi que pour passer du repère R_j au repère R_i , il faut connaître les coordonnées du repère R_i dans le repère R_j . Il est important de préciser ce résultat, car il peut aller contre l'intuition.

L'équation 3.2, bien qu'utile, n'est pas entièrement satisfaisante. En effet, il est assez contre-intuitif d'effectuer des sommes vectorielles en prenant en compte certaines composantes, mais pas d'autres. On se propose donc une autre formulation, matricielle, qui s'avérera très puissante dans la suite du cours :

$$\begin{pmatrix} 1 & 0 & -3 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4.5 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 4.5 + 0 \cdot 3 - 3 \cdot 1 \\ 0 \cdot 4.5 + 1 \cdot 3 - 2 \cdot 1 \\ 0 \cdot 4.5 + 0 \cdot 3 + 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1 \\ 1 \end{pmatrix} \quad (3.3)$$

Cette formulation peut paraître lourde aux premiers abords, mais elle possède plusieurs avantages :

- Pas besoin d'ajouter des règles ad-hoc au produit matriciel pour arriver au bon résultat. Le produit matriciel classique suffit.
- La matrice qui remplace le vecteur caractérisant la position du repère R_i dans le repère R_j contient de précieuses informations sur ce repère. La dernière colonne contient les coordonnées du point d'origine du repère. Les deux premières colonnes contiennent les deux vecteurs qui forment le repère. On remarque d'ailleurs que leurs dernières composantes valent 0, ce qui confirme bien que l'on a bien affaire à des vecteurs.

La matrice précédemment exposée peut donc s'écrire :

$$\begin{pmatrix} u_{x_i}^j & u_{y_i}^j & p_i^j \end{pmatrix} \quad (3.4)$$

La formule 3.4 donne un premier résultat important du cours : on peut regrouper toutes les informations importantes sur un repère dans une matrice, que l'on peut elle-même utiliser pour effectuer des changements de repères.

3.1.4 Matrice de rotation

Un objet peut être traduit par rapport à l'origine du repère, dans ce cas, un point suffit à caractériser sa position. Mais est-ce suffisant en pratique ?

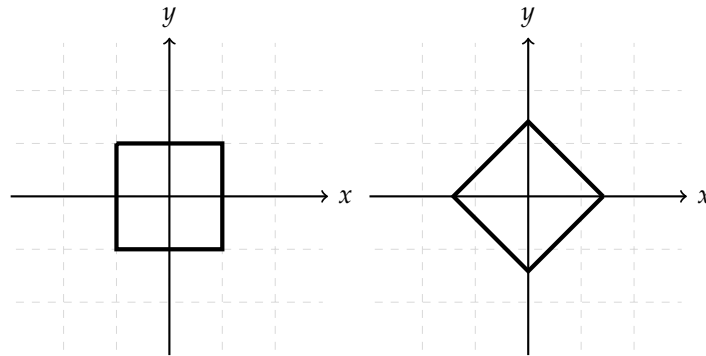


FIGURE 3.13 – A gauche, un objet cubique aligné aux axes du plan. A droite, le même objet tourné de 45° .

La réponse est non, il est aussi nécessaire de quantifier l'orientation des objets précisément pour pouvoir faire de la robotique. Là où le point nous permettait de caractériser les translations, les vecteurs nous permettront ici de caractériser les rotations.

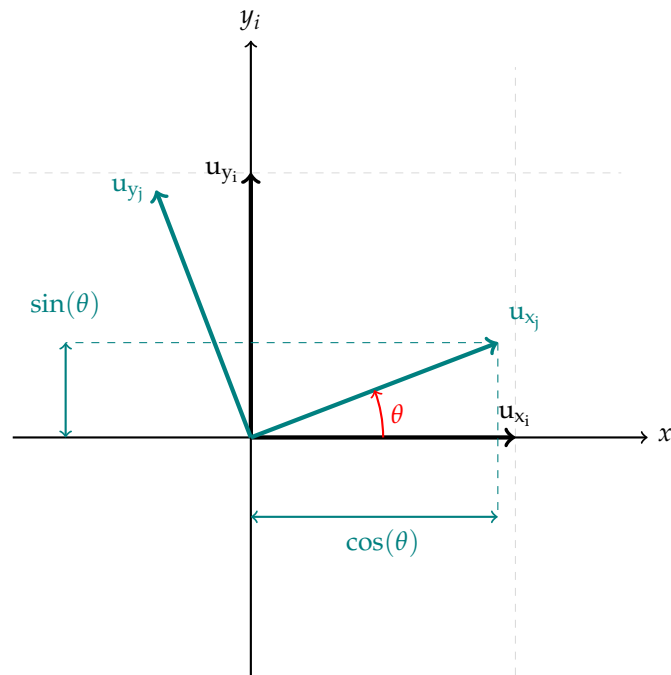
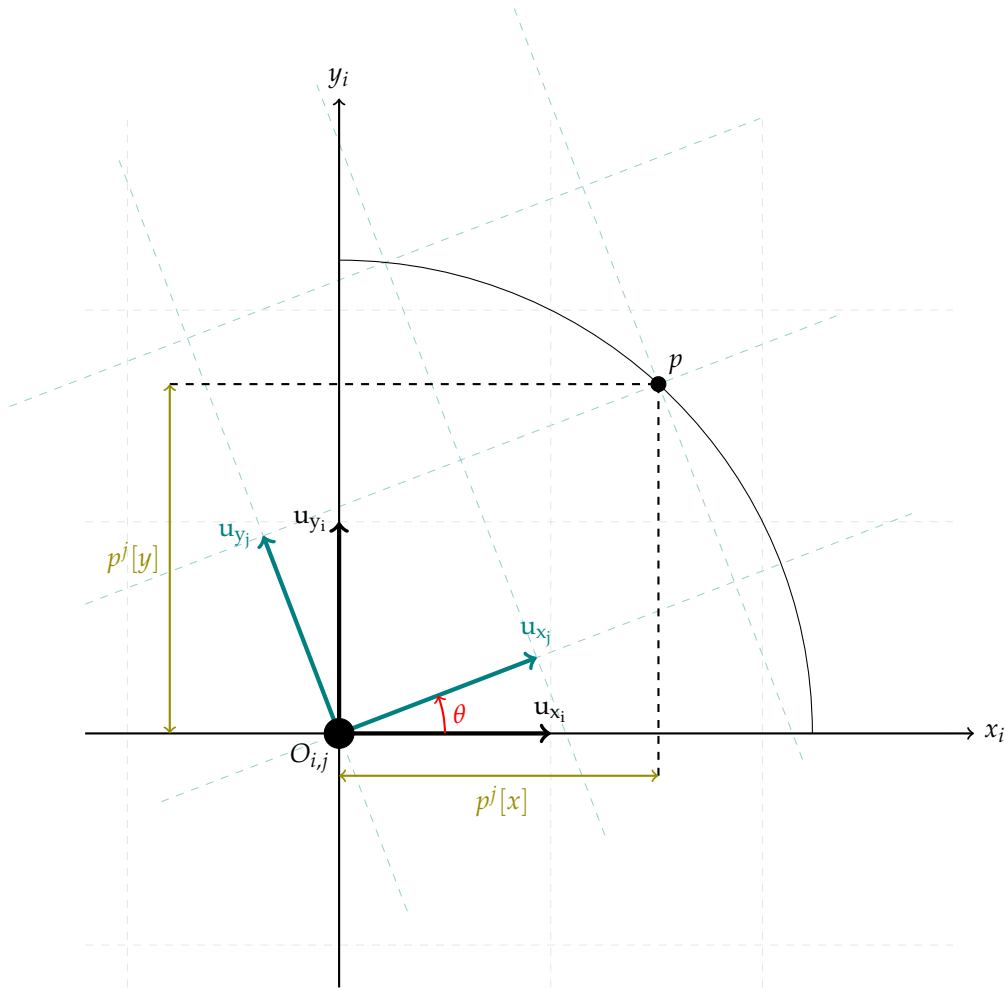


FIGURE 3.14 – Rotation du repère d'un angle θ .

Comme le montre la figure 3.14, ce sont les vecteurs du repère qui seront influencés par la rotation. Ce sont eux qui contiennent l'information d'orientation. Contrairement à la translation où c'est l'origine du repère qui contient l'information importante. Pour les translations, on a cherché à trouver un opérateur, une action à effectuer sur un objet dont on connaît les coordonnées dans un repère pour connaître ses coordonnées dans un autre repère. C'est ce qui nous a amené à introduire la matrice de translation. De manière analogue, il est possible d'introduire des matrices de rotation. On considère deux repères : R_i , le repère canonique et R_j , qui est tourné d'un angle θ par rapport à R_i . On y ajoute un point p , dont les coordonnées dans le repère R_j sont $p^j = (2, 1, 1)$. On souhaite connaître ses coordonnées dans R_i . Les informations se trouvent sur la figure 3.15.

FIGURE 3.15 – Rotation du repère par un angle θ .

Avec un peu de trigonométrie, on trouve que les coordonnées du point p dans le repère R_i sont :

$$p^i = \begin{pmatrix} 2 \cos(\theta) - 1 \sin(\theta) \\ 2 \sin(\theta) + 1 \cos(\theta) \\ 1 \end{pmatrix} \quad (3.5)$$

Analysons cette dernière expression. On y retrouve les coordonnées des vecteurs du repère R_j exprimées dans le repère R_i :

$$u_{x_j}^i = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix}; u_{y_j}^i = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix} \quad (3.6)$$

On y retrouve aussi les coordonnées du point p dans R_j . La question est maintenant de savoir s'il existe, à l'instar de la matrice de translation, un opérateur qui transforme un point donné dans un repère en un autre. La réponse est oui ! C'est la matrice de rotation, qui se présente sous cette forme :

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

Il suffit alors de multiplier p^j par la matrice de rotation pour le changer de repère :

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2\cos(\theta) - 1\sin(\theta) \\ 2\sin(\theta) + 1\cos(\theta) \\ 1 \end{pmatrix} \quad (3.8)$$

On obtient bien le résultat attendu. On remarque aussi que la matrice de rotation n'est pas construite n'importe comment : elle est composée de trois vecteurs colonnes qui sont respectivement $\mathbf{u}_{x_j}^i$, $\mathbf{u}_{y_j}^i$ et O_j^i , comme la matrice de translation...

3.1.5 Matrice de transformation

La matrice de transformation est la matrice qui combine à la fois les principes de la matrice de translation et de la matrice de rotation. Ces deux dernières sont d'ailleurs des cas particuliers de la matrice de transformation. Ce type de matrice est l'objet mathématique qui est au cœur de la théorie derrière la modélisation des robots et des outils de vision. Afin d'introduire les matrices de transformation, considérons l'exemple 3.16, le repère R_i est considéré comme étant le repère canonique ici. Un autre repère R_j , est tourné de θ par rapport à R_i et translaté de 3 unités selon x_i et de 2 unités selon y_i .

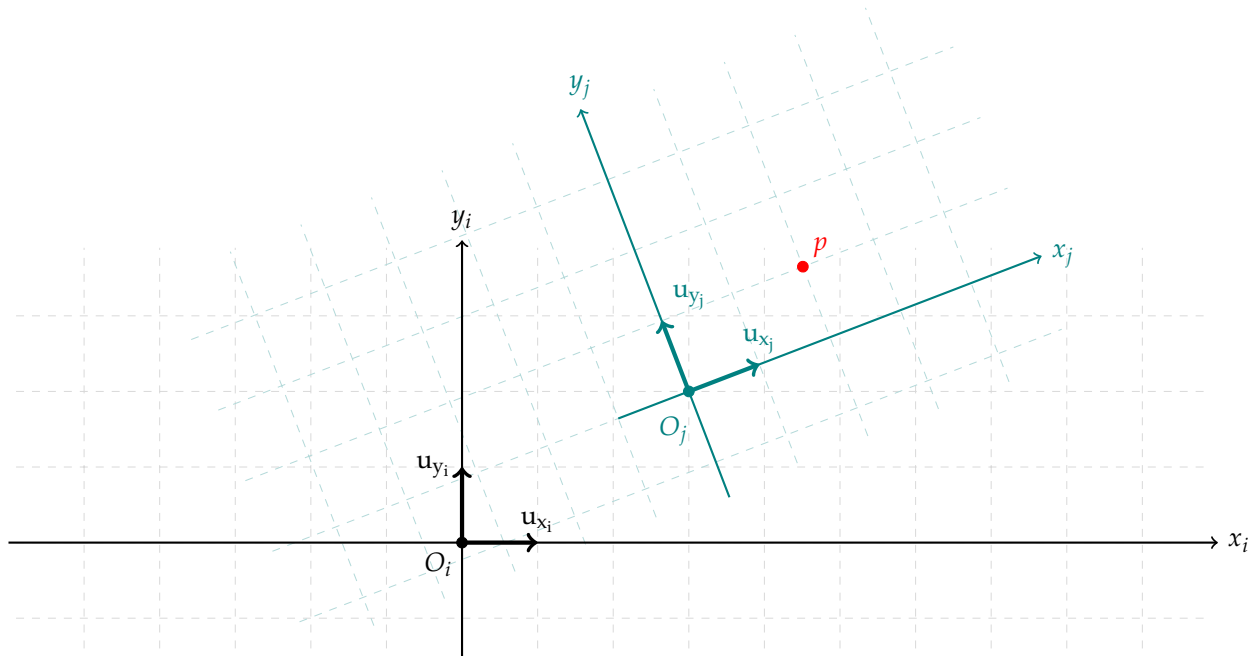


FIGURE 3.16 – $\mathbb{R}^2$ muni de deux repères transformés l'un par rapport à l'autre.

On peut associer deux matrices de transformations à cet exemple, une par repère. Celle du repère R_i , notée T_i^i est de la forme suivante :

$$T_i^i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.9)$$

Celle du repère R_j , exprimée dans le repère R_i , est de la forme suivante :

$$T_j^i = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 3 \\ \sin(\theta) & \cos(\theta) & 2 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.10)$$

avec $\theta = 21^\circ$ ici. On introduit une notation très utilisée en robotique, qui permet un gain de place :

$$T_j^i = \begin{pmatrix} c(\theta) & -s(\theta) & 3 \\ s(\theta) & c(\theta) & 2 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.11)$$

où les fonctions sinus et cosinus sont notées s et c (on évitera donc d'utiliser ces lettres pour nommer des variables).

Quelques remarques sur les matrices de transformation :

- Elles sont constituées des coordonnées des points et des vecteurs qui composent leurs repères associés. Ces coordonnées étant exprimées dans R_i (bien que tout pourrait être exprimé dans R_j , c'est l'avantage de ce formalisme).
- Elles sont constituées, à gauche, d'une partie rotation et à droite, d'une partie translation, chacune bien identifiée :

$$T_j^i = \begin{pmatrix} \mathbf{R}_\theta & t \\ \mathbf{0}_{1 \times 2} & 1 \end{pmatrix} \quad (3.12)$$

- Comme pour les matrices de rotation et de translation, la matrice T_j^i permet ici de transformer les coordonnées d'un point exprimés dans le repère R_j vers le repère R_i . Alors que la matrice est constituée des coordonnées des vecteurs et de l'origine du repère R_j dans le repère R_i . Cette matrice indique donc comment transformer R_i pour obtenir R_j .

$$p^i = T_j^i p^j = T_j^i p^j = \begin{pmatrix} c(\theta) & -s(\theta) & 3 \\ s(\theta) & c(\theta) & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad (3.13)$$

- Les notations en indice et en exposant permettent de tout de suite bien identifier les changements de repère, que ce soit lors de la multiplication d'un vecteur par une matrice de transformation ou d'une matrice de transformation par une autre : $T_i^q T_j^i T_k^j = T_k^q$. Les indices et les exposants en diagonale s'annulent : $T_i^q T_j^i T_k^j = T_k^q$.
- On peut différencier les points des repères en remarquant que les repères se transforment par la droite (il faut multiplier à droite la matrice correspondant au repère par la matrice de transformation), alors que les points se transforment par la gauche (il faut multiplier à gauche le point par la matrice de transformation)

3.2 Et dans l'espace ?

Les outils mathématiques précédemment décrits en deux dimensions se généralisent très bien en trois dimensions. Bien que nous allons convertir les outils en trois dimensions au fur et à mesure, le principe de base est toujours le même : ajouter une coordonnée aux vecteurs et aux points et de passer sur des matrices de transformation de taille 4×4 .

3.2.1 L'espace

La majorité des problèmes réels rencontrés prennent place dans un espace en trois dimensions et non en deux dimensions. Certains problèmes peuvent se résumer à des problèmes plans, mais ces cas ne sont pas les plus communs. Il faut donc adapter notre outil de travail mathématique. On introduit alors l'espace, qui possède les mêmes propriétés que le plan, à ceci près qu'il possède une dimension supplémentaire : les points et les vecteurs seront donc décrits complètement avec trois nombres réels et non plus deux, et leur représentation matricielle posséderont quatre chiffres (en comptant la composante supplémentaire).

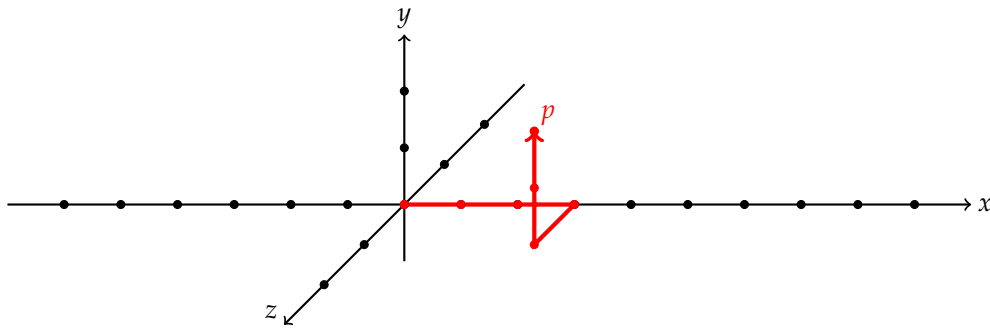


FIGURE 3.17 – L'espace $\mathbb{R}^3$.

Dans la figure 3.17, les coordonnées du point p s'expriment comme ceci : $p = (3 \ 2 \ 1)^t$.

3.2.2 Matrice de translation

Les matrices de transformation sont construites de la même façon en trois dimensions qu'en deux dimensions. C'est donc aussi le cas pour les matrices de translation. Il suffit de connaître les coordonnées de l'origine d'un repère par rapport à un autre pour caractériser la translation. Il faudra pour cela trois nombres réels. La matrice sera elle aussi plus grande (4×4).

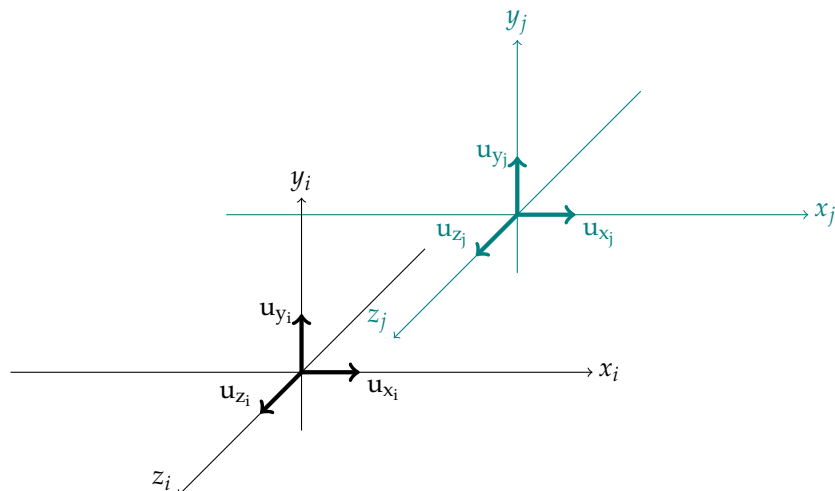


FIGURE 3.18 – Deux repères de $\mathbb{R}^3$ traduits l'un par rapport à l'autre.

On considère l'exemple donné par la figure 3.18. L'origine du repère R_j est translatée de $t_j^i[x]$ par rapport à l'origine du repère R_i dans la direction x_i . Idem pour les directions y_i et z_i dans lesquelles elle est translatée des quantités $t_j^i[y]$ et $t_j^i[z]$.

$$\begin{pmatrix} 1 & 0 & 0 & t_j^i[x] \\ 0 & 1 & 0 & t_j^i[y] \\ 0 & 0 & 1 & t_j^i[z] \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.14)$$

Une matrice de translation en trois dimensions est donc une matrice 4×4 , la dernière ligne est une suite de 0 se terminant par un 1 au niveau de la partie translation. La partie translation étant composée des trois déplacements à effectuer dans chacune des trois directions pour changer de repère.

3.2.3 Matrice de rotation

Encore une fois, les matrices de rotation fonctionnent de la même manière en deux et trois dimensions. À ceci près qu'il y a plus de rotation différentes possible dans l'espace.

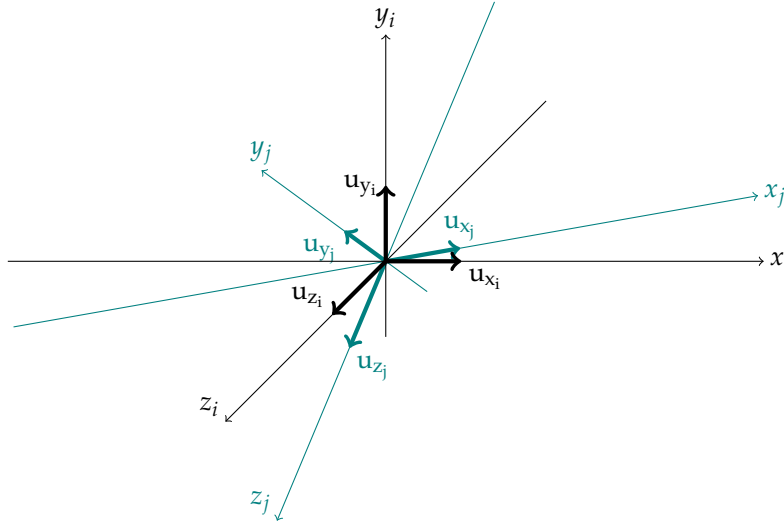


FIGURE 3.19 – Deux repères de $\mathbb{R}^3$ tournés l'un par rapport à l'autre.

Là où dans le plan, il est seulement possible de tourner dans (x, y) , dans l'espace, il est possible de tourner dans les plans suivants : (x, y) , (y, z) et (z, x) . On parlera d'ailleurs de rotation autour d'un axe plutôt que dans un plan, par soucis de compacité d'écriture. On peut donc tourner autour des axes x , y et z . Il est en fait possible de tourner autour d'un axe quelconque (ou dans un plan quelconque), mais comme il est possible de décomposer un vecteur dans la base de l'espace, il est possible de décomposer une rotation en différentes rotations élémentaires autour des axes de la base.

— Matrice de rotation d'angle θ autour de l'axe x :

$$R_{u_x, \theta} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) & 0 \\ 0 & s(\theta) & c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.15)$$

— Matrice de rotation d'angle θ autour de l'axe y :

$$R_{u_y, \theta} = \begin{pmatrix} c(\theta) & 0 & s(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\theta) & 0 & c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.16)$$

— Matrice de rotation d'angle θ autour de l'axe z :

$$R_{u_z, \theta} = \begin{pmatrix} c(\theta) & -s(\theta) & 0 & 0 \\ s(\theta) & c(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.17)$$

— Matrice de rotation d'angle θ autour de l'axe porté par un vecteur unitaire quelconque $\mathbf{u}$ (bien qu'il ne soit pas utile de connaître cette matrice par coeur, il est utile de connaître son existence) :

$$R_{\mathbf{u}, \theta} = \begin{pmatrix} u_x^2(1 - c(\theta)) + c(\theta) & u_x u_y(1 - c(\theta)) - u_z s(\theta) & u_x u_z(1 - c(\theta)) + u_y s(\theta) & 0 \\ u_x u_y(1 - c(\theta)) + u_z s(\theta) & u_y^2(1 - c(\theta)) + c(\theta) & u_y u_z(1 - c(\theta)) - u_x s(\theta) & 0 \\ u_x u_z(1 - c(\theta)) - u_y s(\theta) & u_y u_z(1 - c(\theta)) + u_x s(\theta) & u_z^2(1 - c(\theta)) + c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.18)$$

$$\mathbf{u}\theta = (\theta u_x \ \theta u_y \ \theta u_z)^t \quad (3.19)$$

3.2.4 Matrice de transformation

Tout comme dans le cas du plan, les matrices de transformation sont une combinaison des matrices de translations et des matrices de rotation.

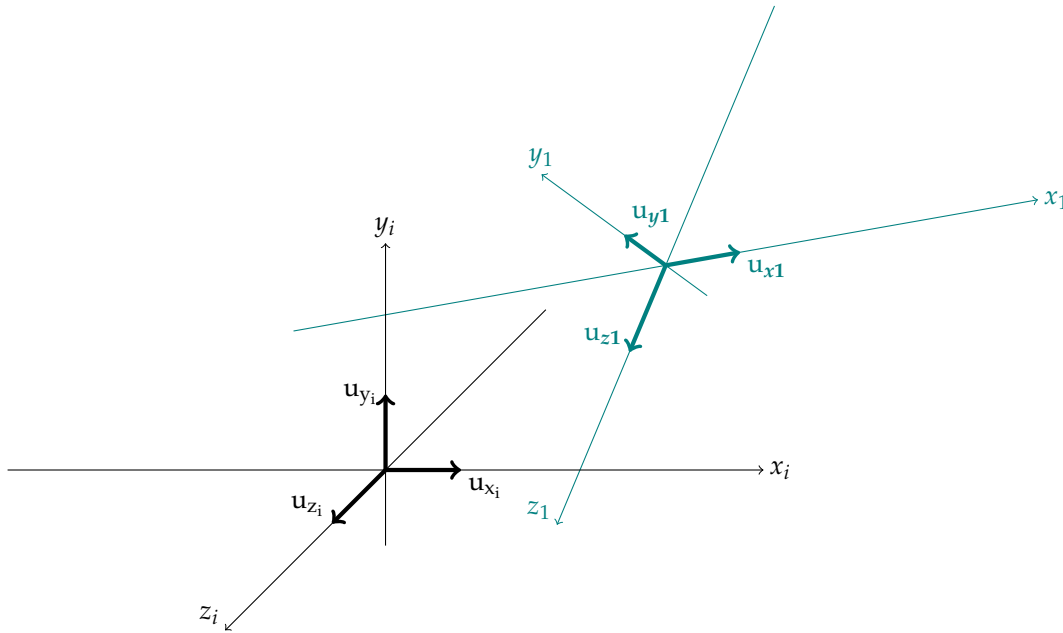


FIGURE 3.20 – Deux repères de $\mathbb{R}^3$ traduits et tournés l'un par rapport à l'autre.

— Matrice de transformation avec rotation autour de l'axe x_i :

$$T_j^i = \begin{pmatrix} 1 & 0 & 0 & t_j^i[x] \\ 0 & c(\theta) & -s(\theta) & t_j^i[y] \\ 0 & s(\theta) & c(\theta) & t_j^i[z] \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.20)$$

— Matrice de transformation avec rotation autour de l'axe y_i :

$$T_j^i = \begin{pmatrix} c(\theta) & 0 & s(\theta) & t_j^i[x] \\ 0 & 1 & 0 & t_j^i[y] \\ -s(\theta) & 0 & c(\theta) & t_j^i[z] \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.21)$$

— Matrice de transformation avec rotation autour de l'axe z_i :

$$T_j^i = \begin{pmatrix} c(\theta) & -s(\theta) & 0 & t_j^i[x] \\ s(\theta) & c(\theta) & 0 & t_j^i[y] \\ 0 & 0 & 1 & t_j^i[z] \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.22)$$

— Matrice de transformation avec rotation autour du vecteur unitaire u :

$$T_j^i = \begin{pmatrix} u_x^2(1 - c(\theta)) + c(\theta) & u_x u_y(1 - c(\theta)) - u_z s(\theta) & u_x u_z(1 - c(\theta)) + u_y s(\theta) & t_j^i[x] \\ u_x u_y(1 - c(\theta)) + u_z s(\theta) & u_y^2(1 - c(\theta)) + c(\theta) & u_y u_z(1 - c(\theta)) - u_x s(\theta) & t_j^i[y] \\ u_x u_z(1 - c(\theta)) - u_y s(\theta) & u_y u_z(1 - c(\theta)) + u_x s(\theta) & u_z^2(1 - c(\theta)) + c(\theta) & t_j^i[z] \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.23)$$

3.3 Caractérisation des rotations

A l'instar des translations, les rotations peuvent être caractérisées par trois paramètres, il est cependant nécessaire de transformer ces paramètres pour obtenir un objet mathématique utilisable en pratique, typiquement, une matrice de rotation. Cette section s'attardera sur deux représentations très utilisées : les angles d'Euler et le vecteur u -theta.

3.3.1 Angles d'Euler

Les angles d'Euler permettent de caractériser n'importe quelle rotation dans l'espace en la découpant en trois rotations élémentaires : la précession, d'angle ψ , autour de l'axe z , suivi de la nutation, d'angle θ ; autour du nouvel axe x (parfois appelé u) obtenu précédemment et enfin la giration, d'angle ϕ autour du nouvel axe z précédemment obtenu.

Les trois matrices de rotations à appliquer les unes à la suite des autres sont les suivantes :

$$\begin{pmatrix} c(\phi) & -s(\phi) & 0 & 0 \\ s(\phi) & c(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) & 0 \\ 0 & s(\theta) & c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c(\psi) & -s(\psi) & 0 & 0 \\ s(\psi) & c(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.24)$$

Une représentation graphique est donnée par les figure 3.21. Le repère d'origine est le repère bleu, le repère final est le rouge. Une première rotation autour de l'axe z bleu permet d'orienter l'axe x bleu sur l'axe en vert. La seconde rotation permet au plan (x, y) de se placer sur la surface rouge. La dernière rotation fait tourner le repère dans le nouveau plan précédemment obtenu.

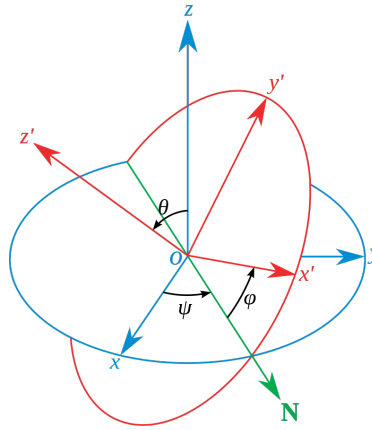


FIGURE 3.21 – Représentation graphique des angles d'Euler | Par Lionel Brits, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=84068570>.

La matrice finale obtenue est la suivante :

$$A = \begin{pmatrix} \cos \psi \cos \varphi - \sin \psi \cos \theta \sin \varphi & -\cos \psi \sin \varphi - \sin \psi \cos \theta \cos \varphi & \sin \psi \sin \theta & 0 \\ \sin \psi \cos \varphi + \cos \psi \cos \theta \sin \varphi & -\sin \psi \sin \varphi + \cos \psi \cos \theta \cos \varphi & -\cos \psi \sin \theta & 0 \\ \sin \theta \sin \varphi & \sin \theta \cos \varphi & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.25)$$

Elle est directement implémentable, l'ingénieur devra s'occuper de paramétrer la rotation avec les trois angles.

De nombreuses bibliothèque de développement informatique utilisent cette représentation, c'est par exemple le cas d'OpenGL⁶.

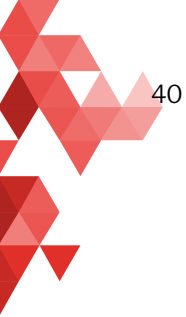
3.3.2 Représentation u-theta

La représentation en u-theta est surement plus intuitive. Elle est composée de trois paramètres qui sont $\theta \mathbf{u}[x]$, $\theta \mathbf{u}[y]$, $\theta \mathbf{u}[z]$. Avec θ l'angle de rotation et $\mathbf{u}$ un vecteur unitaire autour duquel la rotation est effectuée. La formule permettant de retrouver la partie rotation de la matrice de transformation est la suivante :

$$R = \cos(\theta) \mathbf{Id}_{3 \times 3} + (1 - \cos(\theta)) \mathbf{u} \mathbf{u}^t + \sin(\theta) U_{\times} \quad (3.26)$$

avec $\mathbf{Id}_{3 \times 3}$ la matrice identité de taille 3×3 et $U_{\times}$ la matrice de pré-produit vectoriel associé au vecteur $\mathbf{u}$.

6. <http://www.opengl-tutorial.org/fr/intermediate-tutorials/tutorial-17-quaternions/>



4 Liaisons mécaniques

Le deuxième ingrédient qui nous permettra de modéliser les robots, en plus de faire un lien avec la mécanique, est la liaison. Les liaisons sont utilisés dans des chaînes cinématique pour relier des solides considérés comme indéformables. Il existe de nombreuses liaisons différentes mais seul deux nous intéresseront aujourd'hui : la liaison prismatique et la liaison rotoïde.

4.1 Robot : point de vue mécanique

Les robots peuvent, d'un point de vue mécanique, être considérés comme des chaînes de solides indéformables reliés entre eux par des liaisons mécaniques. Les solides indéformables peuvent prendre des formes diverses. Les liaisons permettent d'attacher ensemble plusieurs solides, tout en leur laissant une certaine liberté de mouvement. Les liaisons prismatiques, par exemple, permettent à deux solides de translater entre eux, contrairement aux liaisons rotoïdes, qui leur permettent de pivoter.

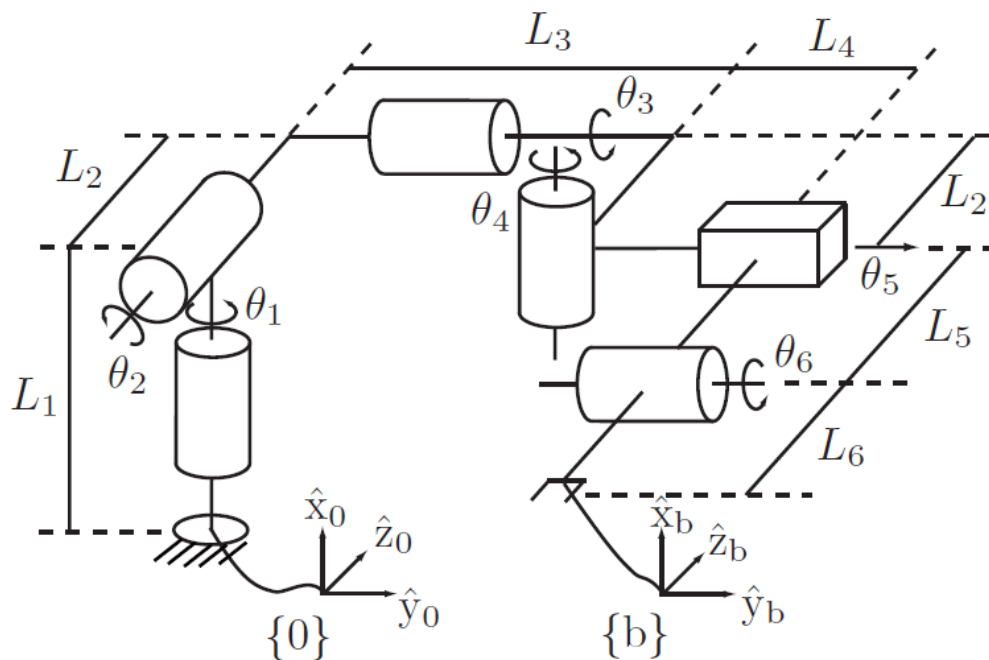


FIGURE 4.1 – Chaînes cinématique d'une boucle ouvert RRRRPR (Modern Robotics : Mechanics, Planning, and Control - F. Park et al. - 2017).

La figure 4.1 montre une chaîne cinématique composée de liaisons rotoïdes et prismatiques (rotation et translation)

4.2 Liaison prismatique

Comme dit précédemment, les liaisons prismatiques permettent de relier deux solides en leur permettant de se translater entre eux. Si on considère que l'un des deux solides est fixe, l'autre ne peut que se translater selon une direction.

Afin de modéliser les robots, on associera des repères aux différentes liaisons qui le composent. Les axes z des repères associés aux liaisons prismatiques seront placés dans le sens de la translation. Nous verrons dans le prochain chapitre, qui présente les conventions permettant de modéliser les robots, que les axes z des repères associés seront toujours placés dans les axes des liaisons.

Les liaisons prismatiques seront toujours représentées par des parallélépipèdes rectangles, la translation se faisant dans le sens de la longueur des parallélépipèdes.



FIGURE 4.2 – Schéma cinématique d'une liaison prismatique.

4.3 Liaison rotoïde

Les liaisons rotoïdes, quant à elles, permettent de pivoter deux solides l'un par rapport à l'autre. Le repère sera placé de telle sorte à ce que les solides tournent autour de l'axe z .

Le corps humain donne de très bons exemples de liaisons rotoïdes. Les coudes en sont un. Un premier solide est formé par le Radius et l'Ulna (os de l'avant-bras), un autre est formé par l'humérus (os qui relie l'épaule au coude). Le coude relie alors ces deux solides et leur permet de pivoter l'un par rapport à l'autre.



FIGURE 4.3 – Schéma cinématique d'une liaison rotoïde.

4.4 Coordonnées articulaires, coordonnées opérationnelles

Dans la suite, nous utiliserons deux types de coordonnées : les coordonnées articulaires et les coordonnées opérationnelles. Ces dernières sont les coordonnées que nous avons l'habitude d'utiliser, les coordonnées de l'espace : x , y et z , auxquelles on ajoute trois paramètres qui caractérisent la rotation, afin d'obtenir une information complète à propos de la position de l'objet. Pour être plus précis encore, lorsque l'on parle de coordonnée opérationnelle, on parle des coordonnées de l'effecteur dans le repère global. Les coordonnées articulaires sont, quant à elle, les

différents paramètres des différentes liaisons du robot : un angle pour une liaison rotoïde, une longueur pour une liaison prismatique. Au même titre que les coordonnées opérationnelles existent dans un espace en six dimensions, les coordonnées articulaires, si elles sont au nombre de n , peuvent être représentées dans un espace à n dimensions¹.

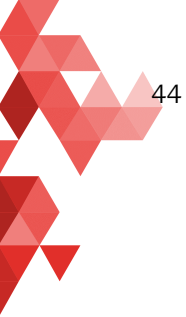
On introduit une notation générale pour les variables articulaires :

$$q_i = (1 - \sigma_i)\theta_i + \sigma_i r_i \quad (4.1)$$

avec $\sigma_i = 0$ pour une liaison rotoïde et $\sigma_i = 1$ pour une liaison prismatique.

Cela permet de définir un vecteur $\mathbf{q}$ qui contient toutes les variables articulaires.

1. Ces notions seront développées dans le chapitre traitant des trajectoires



5 Système, modèle, topologie

5.1 Notions

Les chapitres précédents présentent au lecteur les outils mathématiques et les notions de mécanique nécessaires à la modélisation des robots et systèmes de vision. Cependant, certaines considérations sont à prendre en compte afin de ne pas confondre le système avec son modèle (sa représentation ou sa topologie).

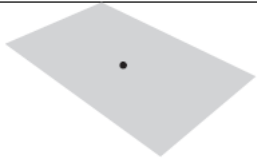
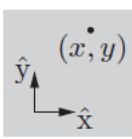
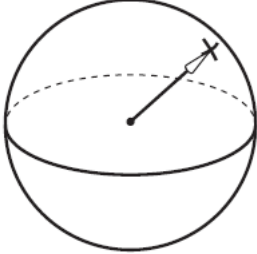
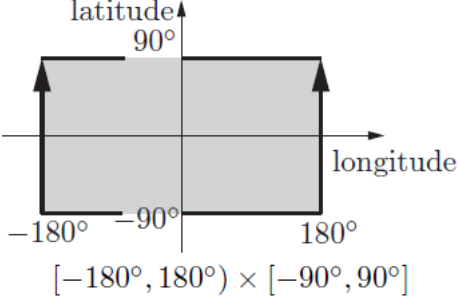
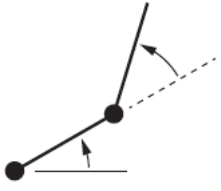
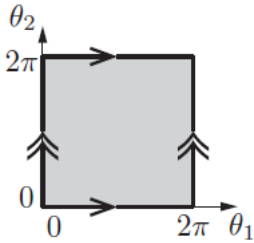
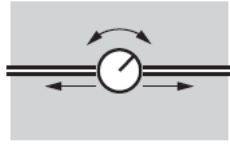
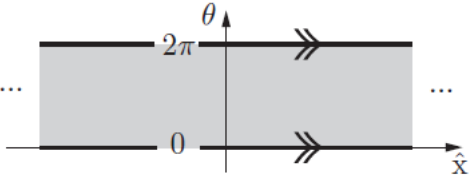
system	topology	sample representation
 point on a plane	 $\mathbb{E}^2$	 $\mathbb{R}^2$
 spherical pendulum	 S^2	 $[-180^\circ, 180^\circ] \times [-90^\circ, 90^\circ]$
 2R robot arm	 $T^2 = S^1 \times S^1$	 $[0, 2\pi) \times [0, 2\pi)$
 rotating sliding knob	 $\mathbb{E}^1 \times S^1$	 $\mathbb{R}^1 \times [0, 2\pi)$

FIGURE 5.1 – Différence entre le système, la topologie du modèle et la représentation du modèle utilisée (Modern Robotics : Mechanics, Planning, and Control - F. Park et al. - 2017).

La figure 5.1 illustre bien ce problème en séparant trois éléments bien distincts : le système, la topologie et la

représentation. Considérons le dernier exemple donné dans la figure 5.1. Il s'agit d'un bouton capable de se translater et de tourner sur lui-même. Il a donc deux degrés de liberté, que l'on peut caractériser par deux variables : θ pour la rotation et x pour la translation. On peut placer ces deux variables dans le plan, on obtient alors la représentation illustrée à droite du schéma. On remarque que la variable x peut évoluer comme elle le souhaite sur $\mathbb{R}^1$, ou $\mathbb{E}^1$ comme il est appelé ici. Alors que θ est périodique, les droites d'équation $\theta = 2\pi$ et $\theta = 0$ sont confondues. La configuration du système physique est alors ici représentée par un point qui se balade dans cette représentation. La représentation d'un système n'est pas absolue, elle peut changer d'un ingénieur à l'autre. Tout comme le modèle d'un robot peut changer en fonction de la convention utilisée et de l'ingénieur qui le modélise. Ce qui ne change pas, c'est la topologie du système. C'est d'ailleurs tout l'intérêt de la topologie : étudier ces invariants. Ici, le système évolue dans l'espace $\mathbb{E}^1 \times S^1$. Cet espace a la forme d'un tube. Un point qui bouge sur la partie circulaire du tube représente le bouton qui tourne sur lui-même, un point qui bouge le long du tube représente la translation du bouton.

Il est important de ne pas confondre le système physique et son modèle (bien que cette position ne soit pas le seul dans l'état de l'art sur les questions épistémiques, mais ce débat dépasse largement le cadre du cours). Le modèle est une représentation imparfaite de la réalité, mais qui représente assez bien le système pour pouvoir être utile en pratique. Le modèle permet aussi de quantifier précisément les choses et d'appliquer des outils mathématiques abstraits.



6 Modélisation géométrique des robots

6.1 Paramétrage de Denavit-Hartenberg modifié (Khalil-Kleinfinger)

6.1.1 Définition

Caractériser un changement de repère demande de connaître six paramètres : trois pour les translations, trois pour les rotations. Or, en robotique, toute cette information n'est pas nécessaire, les liaisons utilisées contraignant beaucoup les mouvements possibles. Le paramétrage de Denavit-Hartenberg modifié (Khalil-Kleinfinger, que l'on nommera DHm), permet de caractériser une transformation avec seulement quatre paramètres. C'est celle-ci qui sera utilisée dans ce cours pour modéliser les robots. De plus, la plupart des paramètres seront fixes, seul une petite partie seront des variables.

Le paramétrage DHm prend comme hypothèse que les repères sont placés d'une certaine façon d'une liaison à l'autre et que pour passer d'un repère à l'autre, quatre transformations élémentaires sont nécessaires : une translation, une rotation, une translation et de nouveau une rotation, autour d'axes bien définis.

Le passage d'un repère à l'autre se fait grâce aux transformations élémentaires suivantes :

- Une rotation autour de l'axe x , d'un angle α . On obtient alors un nouveau repère, les nouveaux axes seront notés x' , y' et z' .
- Une translation selon l'axe x' , d'une distance d . On obtient alors un nouveau repère, les nouveaux axes seront notés x'' , y'' et z'' .
- Une translation selon l'axe z'' , d'une distance r . Dans le cas d'une liaison prismatique, r est la variable articulaire. On obtient alors un nouveau repère, les nouveaux axes seront notés x''' , y''' et z''' .
- Une rotation autour de l'axe z''' , d'un angle θ (il est possible d'ajouter un offset à l'angle de rotation). Dans le cas d'une liaison rotoïde, θ est la variable articulaire. On obtient alors un nouveau repère final voulu.

On insistera sur le fait que chaque transformation donne un repère intermédiaire, depuis lequel la transformation suivante va agir.

Ces transformations prennent la forme matricielle suivante :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.1)$$

$$= \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & d \\ \cos(\alpha)\sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha) & -r\sin(\alpha) \\ \sin(\alpha)\sin(\theta) & \sin(\alpha)\cos(\theta) & \cos(\alpha) & r\cos(\alpha) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

On rappelle aussi qu'il faut lire les équations de droite à gauche, les premières transformations sont celles de droite, les dernière celles de gauche. La matrice correspondant au repère transformée serait tout à droite ici.

6.1.2 Placement des repères

Afin de pouvoir utiliser la transformation précédemment exposée pour modéliser les robots, il faut respecter certaines règles dans le placement des repères sur le robot. Elles sont données dans la méthode suivante [21] (on fera attention à la position initiale du robot, c'est-à-dire la position dans laquelle le robot est quand les variables articulaires ont une valeur nulle) :

- On note i l'axe de la liaison qui relie le solide $i - 1$ au solide i , $i = 1, \dots, n$.
- On attache un repère R_i au bout du solide $i - 1$ (et non pas le solide i).
- L'axe z_i est placé le long de la liaison i , la direction du vecteur associé $\mathbf{u}_{z_i}$ est arbitraire.
- Si z_i et z_{i-1} possèdent une normale commune unique :
 - L'origine O_i se place à l'intersection de l'axe z_i et de la normale commune à z_{i-1} et z_i .
 - L'axe x_i est placé le long de la normale commune à z_{i-1} et z_i , le vecteur associé $\mathbf{u}_{x_i}$ part de z_{i-1} pour aller vers z_i . On vérifiera qu'en position initiale, le solide $i + 1$ est dans l'axe x_i .
- Si z_i et z_{i-1} possèdent une infinité de normales communes (les axes sont parallèles) :
 - L'origine O_i se place à l'intersection de l'axe z_i et d'une des normales communes à z_{i-1} et z_i .
 - L'axe x_i est placé le long de la normale commune choisie à l'étape précédente à z_{i-1} et z_i , le vecteur associé $\mathbf{u}_{x_i}$ part de z_{i-1} pour aller vers z_i .
- Si z_i et z_{i-1} possèdent une normale commune unique est réduite à un point (les axes s'intersectent) :
 - L'origine O_i se place sur l'axe z_i .
 - L'axe x_i est placé perpendiculairement à z_i . On vérifiera qu'en position initiale, le solide $i + 1$ est dans l'axe x_i .
- Si z_i et z_{i-1} possèdent une infinité de normales communes qui se réduisent à un point chacune (les axes sont confondus) :
 - L'origine O_i se place sur l'axe z_i .
 - L'axe x_i est placé perpendiculairement à z_i . On vérifiera qu'en position initiale, le solide $i + 1$ est dans l'axe x_i .
- Choisir $\mathbf{u}_{y_i}$ afin d'obtenir un repère orthonormé direct.

6.2 Modèle géométrique direct (MGD)

Le modèle géométrique direct permet, en connaissant la valeur des coordonnées articulaires, de connaître la valeur des coordonnées opérationnelles. En connaissant les valeurs des angles et des translations des liaisons, on peut connaître la pose de l'effecteur dans le repère global du robot et la position d'un point dans le repère global connaissant sa position dans le repère de l'effecteur.

Pour faire cela, il faut d'abord modéliser le robot, en utilisant par exemple la méthode de DHm. La méthode nous renvoie un ensemble de matrices de transformation, une par liaison et éventuellement deux de plus : une pour le repère global du robot et une pour l'effecteur.

La pose de l'effecteur est donnée par la matrice de transformation T_{n+1}^0 . Il suffit pour l'obtenir de prendre toutes les matrices obtenues grâce à DHm et de les multiplier :

$$T_{n+1}^0 = T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} T_{n+1}^n \quad (6.2)$$

$$p^0 = T_{n+1}^0 p^{n+1} = T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} T_{n+1}^n p^{n+1} \quad (6.3)$$

Il est souvent commode de changer le nom de la matrice qui caractérise la position de l'effecteur E^n et d'ajouter une matrice qui donne la position du robot B_0 :

$$T_E^W = W T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} E^n = W T_n^0 E \quad (6.4)$$

La matrice obtenue est donc celle de l'effecteur dans le repère monde.

On remarque bien ici le côté "direct" du modèle. Les matrices prennent en entrée les paramètres q du robot pour donner une position sous forme de matrice de transformation.

6.3 Modèle géométrique inverse (MGI)

Le modèle géométrique indirect permet, en connaissant les valeurs des coordonnées opérationnelles, de connaître les valeurs nécessaires que les coordonnées articulaires doivent avoir pour atteindre les coordonnées opérationnelles. Il se calcule à partir du MGD.

Le problème que pose le MGI est la non-linéarité du modèle, les rotations sont caractérisées par les fonctions sinus et cosinus, ce qui rend difficile d'isoler les variables θ_i . Même considérant les fonctions comme les inconnus (ce qui multiplie par deux le nombre de variable, car il faut une variable pour le cosinus et une variable pour le sinus), il y a des multiplications qui rendent aussi les équations non-linéaires.

On peut définir le MGI ainsi :

$$q = MGI(T_E^W) \quad (6.5)$$

Trouver le MGI demande de résoudre le système d'équations suivantes :

$$T_E^W = W T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} E = W T_n^0 E \quad (6.6)$$

On souhaite alors trouver q en fonction T_E^W qui est donné. Il est possible déjà de modifier l'équation pour ne garder à droite que des termes qui contiennent q :

$$W^{-1} T_E^W E^{-1} = U = T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} \quad (6.7)$$

Résoudre ce genre d'équation demande beaucoup d'intuition, mais il existe une méthode[16] qui permet de faciliter les calculs. Chaque matrice qui membre de droite ne contient qu'une seule variable q_i . On passe alors cette matrice dans le membre de gauche (en multipliant par son inverse) pour résoudre les équations pour cette variable en fonction des autres, puis on recommence :

$$U_0 = T_1^0 T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} \rightarrow T_0^1 U_0 = T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} \quad (6.8)$$

avec T_0^1 l'inverse de T_1^0 .

Au final, il faut résoudre à la suite les systèmes d'équations suivants :

$$T_0^1 U_0 = T_2^1 T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1} \quad (6.9)$$

$$T_1^2 U_1 = T_3^2 \dots T_{n-1}^{n-2} T_n^{n-1}$$

$$T_2^3 U_2 = \dots T_{n-1}^{n-2} T_n^{n-1}$$

...



$$T_{n-1}^n U_{n-1} = T_n^{n-1}$$

avec $U_i = T_n^i = T_{i-1}^i U_{i-1}$.



III Annexes





A Outils mathématiques complémentaires

A.0.1 Rappels sur le produit matriciel

Les étudiants qui sont à l'aise avec le produit matriciel peuvent passer à la sous-section suivante.

Une matrice peut être considérée comme un tableau de nombres rectangulaire. Les matrices sont bien évidemment des objets beaucoup plus complexes que cela, et certains de leurs aspects seront explorés pendant ce cours, mais cette définition suffira pour définir le produit matriciel. Premièrement, on rappelle que le produit matriciel n'est pas commutatif, contrairement au produit de nombres réels, donc si on considère deux matrices $M \in \mathbb{R}^{n \times n}$ et $P \in \mathbb{R}^{n \times n}$ (les matrices M et P possèdent donc n lignes et n colonnes), on a :

$$MP \neq PM \tag{A.1}$$

Ensuite, pour pouvoir multiplier deux matrices entre elles, le nombre de colonnes de la matrice de gauche doit correspondre au nombre de lignes de la matrice de droite.

La figure A.1 donne une représentation graphique du produit matriciel.

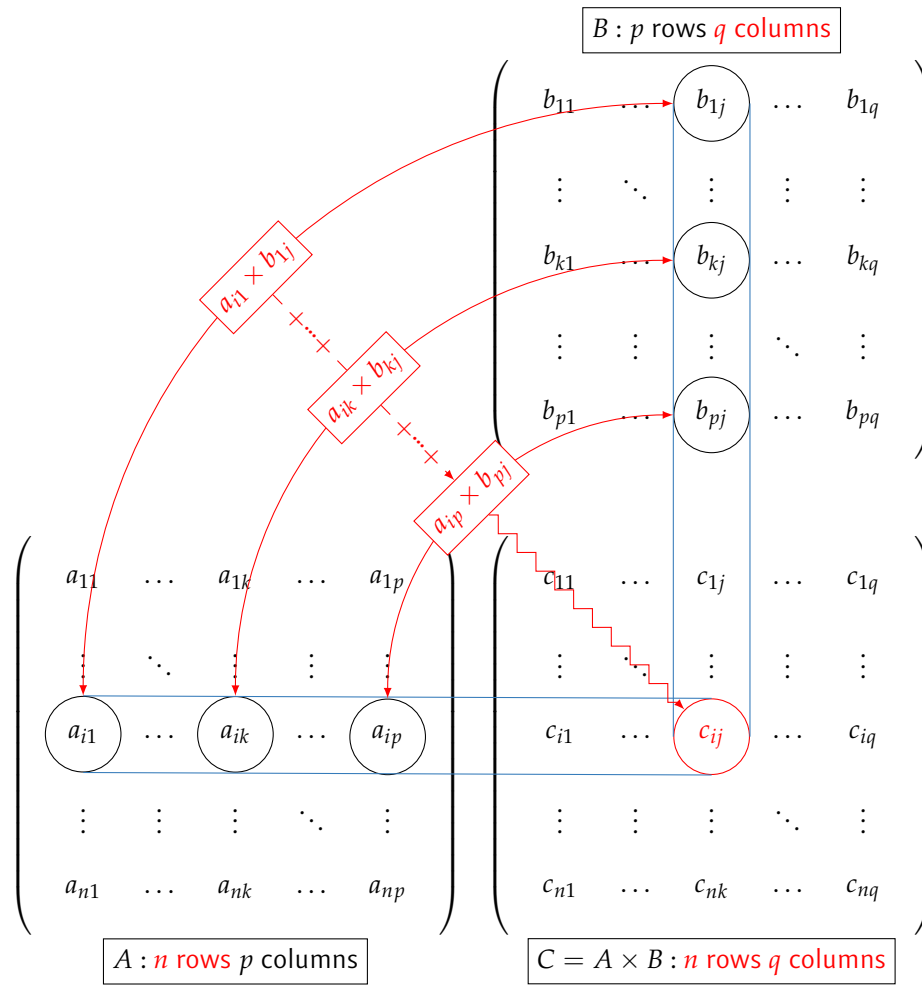


FIGURE A.1 – Représentation graphique du produit matriciel

La formule générale permettant de trouver les coefficients de la matrice C est :

$$c_{ij} = \sum_k a_{ik} b_{kj} \quad (\text{A.2})$$

Mais le plus intéressant est de produire un algorithme qui, prenant en entrée les deux matrices A et B , renvoie leur produit C :

Algorithm 4 Matrix product

Require : $A \in \mathbb{R}^{n \times p}, B \in \mathbb{R}^{p \times q}$
for $0 < i < n - 1$ **do**
 for $0 < k < q - 1$ **do**
 $C[i][j] \leftarrow 0$
 for $0 < k < p - 1$ **do**
 $C[i][j] \leftarrow C[i][j] + A[i][k] \cdot B[k][j]$
 end for
 end for
end for
return C



Table des figures

1.1	Les différentes applications de la robotique. (itsocial.fr usine-digitale.fr orange.com)	8
1.2	Systèmes de contrôle qualité par vision (elementsindustriels.fr)	9
2.1	Un robot de palettisation Kuka (▶ Productive and flexible way of palletizing consumer goods KUKA - Robots & Automation).	11
2.2	Des robots industriels Fanuc chez Audi (▶ FANUC Industrial Robots at AUDI FANUC Europe).	12
2.3	Un Fanuc M-2000iA/1700L soulevant une Chevrolet Corvette C7 pesant 1539 kg à vide (▶ New FANUC M-2000iA/1700L Robot Lifts Corvette Demonstrating Long Reach & Heavy Payload FANUC America Corporation).	13
2.4	Plateforme robotique d'assistance chirurgicale ROSA par Stäubli (▶ First robotic platform for surgical assistance RoboticsStaubli 2020).	14
2.5	Laboratoire de biochimie (🔗 Thread sur la visite d'un laboratoire de biochimie 2021).	14
2.6	Un quadrotor transportant des bouteilles et des canettes en utilisant la commande prédictive (in Agile quadrotor under uncertainty - Davide SCARAMUZZA 2021).	15
2.7	Un drone capable d'atterrir sur une cible en mouvement (▶ Vision based Autonomous Quadrotor Landing on a Moving Platform - UZH Robotics and Perception Group 2017).	15
2.8	Véhicule Tesla complètement autonome sur route (▶ Full Self-Driving - Tesla 2019).	16
2.9	Execution de l'algorithme LDSO sur le dataset Kitti (▶ LDSO : Direct Sparse Odometry with Loop Closure - Gao, Wang, Demmel, Cremers 2018).	16
2.10	Hugh Herr et ses prothèses de jambes (▶ Hugh Herr, l'homme « bionique » - FUTUREMAG - ARTE 2016).	17
2.11	Robot Atlas de Boston Dynamics (▶ Atlas Partners in Parkour 2021).	17
3.1	Schéma d'un robot RRT (Rotation-Rotation-Translation). La question centrale en robotique est de savoir où se situe, dans l'espace, les différentes partie du robot par rapport, par exemple, à sa base (📖 Robotics - T. Bajd et al. - 2010).	21
3.2	Représentation graphique du plan $\mathbb{R}^2$. Un point est sélectionné sur le plan. Les coordonnées de ce point représentent les déplacements qu'il faut effectuer depuis l'origine et le long des deux directions x et y pour l'atteindre.	22
3.3	Exemple de représentation d'un objet avec un volume non nul.	23
3.4	Distance euclidienne entre deux points.	24
3.5	Distance de Manhattan entre deux points.	24
3.6	Distance de Tchebychev entre deux points.	25
3.7	$\mathbb{R}^2$ muni du repère orthonormé canonique.	25
3.8	$\mathbb{R}^2$ muni du repère orthonormé non canonique.	26
3.9	Illustration de la relativité de deux repères tournés l'un par rapport à l'autre.	26
3.10	Schéma d'un robot RRTR (Rotation-Rotation-Translation-Rotation) sur lequel sont placés cinq repères (📖 Robotics - T. Bajd et al. - 2010).	27
3.11	$\mathbb{R}^2$ muni de deux repères translatés l'un par rapport à l'autre.	28
3.12	$\mathbb{R}^2$ muni de deux repères translatés l'un par rapport à l'autre.	28
3.13	A gauche, un objet cubique aligné aux axes du plan. A droite, le même objet tourné de 45°	30
3.14	Rotation du repère d'un angle θ	30
3.15	Rotation du repère par un angle θ	31
3.16	$\mathbb{R}^2$ muni de deux repères transformées l'un par rapport à l'autre.	33
3.17	L'espace $\mathbb{R}^3$	35

3.18	Deux repères de $\mathbb{R}^3$ traduits l'un par rapport à l'autre.	35
3.19	Deux repères de $\mathbb{R}^3$ tournés l'un par rapport à l'autre.	36
3.20	Deux repères de $\mathbb{R}^3$ traduits et tournés l'un par rapport à l'autre.	37
3.21	Représentation graphique des angles d'Euler Par Lionel Brits, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?rid=84068570	39
4.1	Chaînes cinématique d'une boucle ouvert RRRRPR ( Modern Robotics : Mechanics, Planning, and Control – F. Park et al. – 2017).	41
4.2	Schéma cinématique d'une liaison prismatique.	42
4.3	Schéma cinématique d'une liaison rotoïde.	42
5.1	Différence entre le système, la topologie du modèle et la représentation du modèle utilisée ( Modern Robotics : Mechanics, Planning, and Control – F. Park et al. – 2017).	45
A.1	Représentation graphique du produit matriciel	54



Bibliographie

- [1] FUTUREMAG ARTE. ▶ *Hugh Herr, l'homme « bionique »* - FUTUREMAG - ARTE. URL : https://www.youtube.com/watch?v=rEKe_ShtqaI&.
- [2] IAI AUTOMATION. ⚙️ *Modular Cartesian Robots Systems*. URL : <https://www.iai-automation.com/en/cartesian-robots.html#3-axes-combinations>.
- [3] T. BAJD et al. 📖 *Robotics. Intelligent Systems, Control, Automation : Science et Engineering*, 2010.
- [4] @BIOHOSPITALIX. 🗨️ *Thread sur la visite d'un laboratoire de biochimie*. URL : <https://twitter.com/BioHospitalix/status/1408143717161213954>.
- [5] FANUC America CORPORATION. ▶ *New FANUC M-2000iA/1700L Robot Lifts Corvette Demonstrating Long Reach & Heavy Payload*. URL : <https://www.youtube.com/watch?v=69RtLBImXiU>.
- [6] Boston DYNAMICS. ▶ *Atlas | Partners in Parkour*. URL : <https://www.youtube.com/watch?v=tF4DML7FIWk>.
- [7] FANUC EUROPE. ▶ *FANUC Industrial Robots at AUDI*. URL : <https://www.youtube.com/watch?v=rbki4HR41-4>.
- [8] 📖 *FANUC Robot LR Mate 200iD FANUC Robot LR Mate 200iD/7L FANUC Robot LR Mate 200iD/4S. Livret intégrateur. FANUC, 2013 v3. FANUC Robotics. 2013.* URL : <https://docplayer.fr/20527577-Fanuc-robotics-fanuc-robot-lr-mate-200id-fanuc-robot-lr-mate-200id-7l-fanuc-robot-lr-mate-200id-4s-livret-integrateur-fanuc-2013-v3.html>.
- [9] FANUC. ⚙️ *Fanuc M-2000iA/1700L*. URL : <https://www.fanuc.eu/fr/fr/robots/page-de-filtre-des-robots/m-2000-series/m-2000ia-1700L>.
- [10] Xiang GAO et al. *LDSO : Direct Sparse Odometry with Loop Closure (pre-print. 2018. arXiv : 1808.01111 [cs.CV]*.
- [11] Andreas GEIGER, Philip LENZ et Raquel URTASUN. « Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite ». In : *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [12] INDUSTRIE-ROBOTER.COM. ▶ *The fastest robot of the world from Stäubli High Speed*. URL : <https://www.youtube.com/watch?v=e9geaPrEW3E>.
- [13] IAI America INTELLIGENTACTUATOR.COM. ▶ *CT-4 High-Speed Cartesian Robot*. URL : <https://www.youtube.com/watch?v=rstvgiZFIXE>.
- [14] Eelke Bosma & Eelco Veen & Mariska de JONGH & JAN ROUKEMA. « Variable impact of complications in generale surgery : a prospective cohort study ». In : *Canadian Journal of Surgery* (2012).
- [15] Jean-François KHALIL Wisama & Kleinfinger. « A new geometric notation for open and closed-loop robot ». In : *IEEE International Conference on Robotics and Automation* (1986).
- [16] Wisama KHALIL et Étienne DOMBRE. 📖 *Modeling, Identification and Control of Robots*. Kogan Page Science, 2004.
- [17] KUKA. ▶ *Productive and flexible way of palletizing consumer goods*. URL : https://www.youtube.com/watch?v=2SEYgp8cW_A&list=PLcmh-lxe_PW5pe4JDS-m5LPexxbB1jaRr&index=14.
- [18] Frank C. PARK et Kevin M. LYNCH. 📖 *Modern Robotics : Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [19] UZH ROBOTICS et Perception GROUP. ▶ *Vision based Autonomous Quadrotor Landing on a Moving Platform*. URL : <https://www.youtube.com/watch?v=Tz5ubwoAfNE>.
- [20] ROBOTICSSTAUBLI. ▶ *First robotic platform for surgical assistance*. URL : <https://www.youtube.com/watch?v=W53EmrgxJT4>.
- [21] Subir Kumar SAHA. 📖 *Introduction to Robotics*. McGraw Hill, 2008.

- [22] Subir Kumar SAHA.  *RoboAnalyzer : 3D Model Based Robotics Learning Software : Home Page*. URL : <http://www.roboanalyzer.com/>.
- [23] Grant SANDERSON.  *Essence of linear algebra*. URL : <https://www.youtube.com/playlist?list=PLZHQ0b0WTQDPD3MizzM2xab>.
- [24] Davide SCARAMUZZA. **in** *Agile quadrotor control under uncertainty*. URL : <https://www.linkedin.com/feed/update/urn:li:activity:6843221234293993473/>.
- [25] Drew Hanover & Philipp Foehn & Sihao Sun & Elia Kaufmann & Davide SCARAMUZZA. «  Performance, Precision and Payloads : Adaptive Nonlinear MPC for Quadrotors (pre-print) ». In : *arXiv* (2021).
- [26] SCOTT.  *Centralised Robotic Palletising*. URL : <https://www.scottautomation.com/products/centralised-robotic-palletising/>.
- [27] Cyrill STACHNISS.  *SLAM - 5 Minutes with Cyrill | Partners in Parkour*. URL : <https://www.youtube.com/watch?v=BuRCJ2fegcc>.
- [28] Michael STEVENS.  *Laws & Causes*. URL : https://www.youtube.com/watch?v=_WHRWLnVm_M&.
- [29] STÄUBLI.  *Plateforme d'assistance chirurgicale ROSA*. URL : <https://www.staubli.com/fr-fr/robotics/solution-application/applications-pharmaceutiques/medical-robotics/success-story-zimmer-biomet/>.
- [30] STÄUBLI.  *Robots industriels SCARA*. URL : <https://www.staubli.com/fr-be/robotics/gamme-produits/robots-industriels/robots-scara-4-axes/>.