



Université technique de Sofia

MÉMOIRE

par Gergana Petrova, №: 211320011

**Thème: Création d'un système de contrôle à
distance d'un bras robotisé**

**Directeur scientifique :
(Nicole Christoff)**

Contenu de mémoire de projet:

Introduction	4
Chapitre 1 : Etat de l'art	7
1. Revue technique sur les bras robotisés conçus.....	7
1.1 Contrôleurs.....	9
1.2 Degrés de liberté.....	9
1.3 Télécommande.....	10
1.4 Lien de communication.....	10
1.5 Conclusion.....	11
2. Analyse comparative des filtres de traitement des données de capteurs.....	11
2.1 Filtres de visualisation utilisés (filtre utilisé).....	13
2.2 Capteur et microcontrôleur.....	13
2.3 Application et contexte d'utilisation.....	13
2.4 Raisons du choix du filtre.....	14
2.5 Conclusion.....	14
Chapitre 2 : Partie théorique.....	16
1. Introduction à la robotique et à la plateforme Arduino.....	16
2. Types de bras robotisés.....	19
3. Communications radio.....	22
4. Détecteurs de mouvement.....	26
5. Filtres - passe-haut et passe-bas.....	28
Chapitre 3 : Réalisation du système de contrôle à distance.....	30
1. Vue d'ensemble du système.....	30
2. Partie transmission.....	32
2.1 Vue générale du système.....	32
2.2 Sélection du capteur.....	34
2.3 Sélection du module radio.....	34
2.4 Connexion à Arduino.....	34

2.5 Traitement des données.....	35
2.6 Programmation d'un module de capture de mouvement.....	36
3. Communication sans fil.....	43
4. Partie réceptrice.....	45
4.1 Vue générale du système.....	45
4.2 Réception des signaux.....	47
4.3 Interprétation des commandes.....	48
4.4 Contrôle servomoteur du bras robotique.....	48
4.5 Programmation d'un module pour exécuter des mouvements.....	49
Chapitre 4 : Analyse des résultats obtenus, applicabilité et conclusions.....	54
1. Démonstration du fonctionnement du prototype.....	54
1.1 Implémentation matérielle et principaux composants du prototype.....	54
1.2 Traitement des données et résultats du moniteur série.....	57
2. Analyse comparative des résultats à l'aide du filtre de Kalman et des angles d'Euler.....	59
3. Analyse des performances.....	62
4. Avantages et limites du système.....	64
4.1 Avantages.....	64
4.2 Limitations.....	65
4.3 Opportunités d'amélioration.....	65
5. Conclusion.....	66
Bibliographie.....	69

INTRODUCTION

Le monde de la robotique se développe de manière dynamique, ce qui conduit à son application toujours plus large dans l'industrie et dans la vie. Les robots sont utilisés avec succès dans de nombreux secteurs, de l'assemblage et du soudage dans l'industrie automobile à l'emballage automatisé dans l'industrie agroalimentaire. La raison en est leur capacité à effectuer des tâches répétitives avec une grande précision, fiabilité et sans besoin de repos. De plus, les robots sont capables de travailler dans des conditions inadaptées ou dangereuses pour les humains: températures élevées, fumées toxiques, environnements radioactifs ou activités monotones et épuisantes.

Avec l'utilisation croissante des systèmes automatisés, il devient nécessaire que les humains puissent contrôler les machines de manière simple et intuitive. Cela implique le développement de technologies de contrôle vocal, visuel ou gestuel. Le contrôle gestuel se distingue par une interaction naturelle avec la machine, similaire à la communication humaine. Cette approche trouve des applications dans les environnements industriels, ainsi que dans les domaines de la médecine, de la réadaptation, de l'éducation et même du divertissement.

La technologie de contrôle gestuel utilise souvent des unités de mesure inertielles (IMU), qui incluent des accéléromètres et des gyroscopes. L'analyse des données provenant de ces capteurs permet de déterminer l'orientation et le mouvement de la main de l'utilisateur. Pour que le système soit fiable et précis, les données des capteurs nécessitent un prétraitement et un filtrage afin de garantir la précision et la fiabilité de leur interprétation. Il peut même éliminer les gestes involontaires de la main.

Dans ce contexte, le développement d'un système utilisant des capteurs pour capturer les gestes et les mouvements d'une main humaine et les convertir en commandes pour contrôler une main robotisée constitue une innovation actuelle

et pertinente. Le projet actuel illustre ce type de système : un microcontrôleur Arduino et un module de communication radio sont utilisés pour créer une plateforme abordable, efficace et fonctionnelle de contrôle de mouvement. L'utilisation de modules radio permet un contrôle à distance sans câble, offrant ainsi liberté de mouvement et flexibilité d'utilisation.

Ce développement démontre non seulement l'application de technologies modernes, mais combine également des composants matériels et logiciels facilement adaptables et évolutifs. Il est donc parfaitement adapté aux applications pédagogiques et industrielles réelles, où des solutions d'automatisation économiques et fiables via une interface gestuelle sont recherchées.

OBJECTIF DU DÉVELOPPEMENT

L'objectif de cette thèse est de concevoir et de développer un prototype de robot industriel (bras robotisé) à quatre degrés de liberté, contrôlé par un automate programmable radio. Il permet également de surveiller le traitement des données et le canal radio par lequel les deux composants du projet communiquent entre eux. Le robot reproduit les mouvements de la main de l'utilisateur grâce à des capteurs. L'objectif principal est de développer un prototype initial qui non seulement démontre les fonctionnalités clés, mais serve également de base au développement ultérieur d'un système plus complexe et avancé, adapté à des tâches pratiques spécifiques et offrant un potentiel d'application concrète dans le domaine concerné. Ce prototype permet de valider les principaux sous-systèmes, tels que le contrôle du servomoteur via un microcontrôleur. Le projet comprend deux parties principales: l'émission et la réception. L'émetteur enregistre les mouvements de la main, traite les données collectées et transmet les commandes par radio à l'émetteur, puis au récepteur, auquel la main robotique mécanique est connectée, exécutant les mouvements correspondants. Chaque geste de la main

correspond à un code généré par l'émetteur, transmis par radio à l'émetteur et exécuté par celui-ci.

CHAPITRE 1: ETAT DE L'ART

Afin de construire une compréhension des solutions existantes dans le domaine du contrôle des bras robotisés, ce chapitre présente et analyse différents projets décrits dans la littérature scientifique. Une analyse comparative de leurs caractéristiques techniques, des composants utilisés et des méthodes de contrôle permet d'identifier les pratiques et approches les plus efficaces applicables au présent travail.

1. Revue technique sur les bras robotisés conçus

Le tableau 1 présente des informations résumées sur les développements existants des bras robotiques, y compris les caractéristiques techniques clés telles que le microcontrôleur utilisé, les degrés de liberté, la méthode de contrôle à distance et le type de connexion de communication, qui servent de base à la sélection des solutions techniques dans le présent projet.

Tableau 1.1. Revue technique des bras robotiques conçus

Contrôleur	Degrés de liberté	Commande à distance	Type de connexion	Article
Arduino Uno	6	Bluetooth via application mobile	Module Bluetooth (HC-05)	[1]
Arduino ESP32	4	IoT via interface Web (Wi-Fi)	Wi-Fi (via serveur Web sur ESP32)	[2]
Arduino Uno	4	Potentiomètres (filaires)	Entrée analogique directe (potentiomètres câblés)	[3]
Arduino Mega (2 units)	6	Contrôle gestuel sans fil (NRF24L01+)	Émetteurs-récepteurs sans fil NRF24L01+	[4]
Microcontrôleur (non spécifié, basé sur Bluetooth)	Non spécifié	Bluetooth via l'application Android	Bluetooth (via l'application Android et la transformation RF)	[5]

Sur la base de l'analyse documentaire, un tableau récapitulatif (Tableau 1) a été créé, permettant une comparaison claire et systématique de cinq projets, en tenant compte de la construction et du contrôle de bras robotisés. Le point commun entre tous les développements présentés est que chacun d'eux contient trois sous-systèmes clés : mécanique, électrique et logiciel. La relation entre ces composants détermine des facteurs critiques tels que la fiabilité, le coût, les performances et l'évolutivité du système. Par conséquent, comme le soulignent Ali et al.[1], le

succès du système autonome dépend d'une intégration bien planifiée de ces trois domaines grâce à des paradigmes de conception modernes.

1.1 Contrôleurs

La comparaison des microcontrôleurs utilisés montre une nette préférence pour la plateforme Arduino - grâce à son environnement de développement open source, à faible coût et facile à utiliser.

Différents modèles de microcontrôleurs sont utilisés dans différentes études en fonction des besoins spécifiques des projets. Ali et al. [1] et Basith & Marroquin [3] utilisent l'Arduino Uno, adapté à des fins éducatives et de démonstration où plusieurs entrées/sorties analogiques et numériques sont nécessaires. Ahmed et al. [2] s'appuient sur l'Arduino ESP32, un microcontrôleur avec module Wi-Fi intégré, offrant des performances supérieures et une meilleure connectivité pour les applications IoT. Dans le projet de Wen et al. [4], deux Arduino Mega sont utilisés en raison de la nécessité de contrôler simultanément plusieurs capteurs et de maintenir la communication entre l'émetteur et le récepteur. En revanche, Reddy et al. [5] ne précisent pas le modèle spécifique du microcontrôleur, mais la description suggère l'utilisation d'une plateforme compatible Bluetooth, probablement basée sur Arduino ou un autre matériel ouvert.

1.2 Degrés de liberté

La variété des degrés de liberté (DOF) reflète les différents objectifs et niveaux de complexité des projets étudiés. Ali et al. [1] et Wen et al. [4] ont développé des bras robotisés avec 6 degrés de liberté, offrant une mobilité et une flexibilité élevées, adaptées aux manipulations complexes. D'autre part, Ahmed et al. [2] et Basith & Marroquin [3] ont limité leurs systèmes à 4 DOF, ce qui est suffisant pour des tâches plus simples comme le pick and place. Reddy et al. [5] n'ont pas précisé le nombre de degrés de liberté dans leur projet.

1.3 Télécommande

Toutes les solutions envisagées illustrent différentes approches du contrôle à distance, ce qui met en évidence la grande diversité des technologies applicables. Ali et al. [1] contrôlent le bras robotique via une application mobile et Bluetooth, offrant une interface utilisateur pratique. Ahmed et al. [2] s'appuient sur une interface HTML Web accessible via n'importe quel appareil du réseau Wi-Fi local. Basith et Marroquin [3] utilisent un contrôle analogique via des potentiomètres - une approche économique, quoique plus limitée. Reddy et al. [5] utilisent également une connexion Bluetooth avec une application Android, en mettant l'accent sur la facilité d'utilisation et les faibles coûts de mise en œuvre. Parallèlement, Wen et al. [4] proposent un contrôle gestuel via un gant capteur et une communication radio, permettant un contrôle plus naturel et intuitif.

1.4 Lien de communication

Les projets démontrent une variété d'approches pour la communication entre le contrôleur et le bras robotique, en fonction des besoins de mobilité, de portée et de fiabilité. Ali et al. [1] et Reddy et al. [5] utilisent des modules Bluetooth (HC-05), qui fournissent une connexion sans fil avec une faible complexité et une bonne compatibilité avec les appareils mobiles. Ahmed et al. [2] s'appuient sur la communication WiFi, qui offre une plus grande portée et permet l'accès via des périphériques réseau. Le projet de Basith et al. [3] a choisi la communication filaire, qui offre une grande fiabilité, bien qu'avec une mobilité limitée. D'autre part, Wen et al. [4] utilisent des modules radio NRF24L01+, qui combinent une faible consommation d'énergie et une vitesse de transmission élevée, adaptés au contrôle gestuel et au fonctionnement synchrone des appareils distants.

1.5 Conclusion

Ces projets illustrent différentes stratégies de mise en œuvre d'un bras robotisé, chacune présentant ses propres avantages selon l'application. Leur point commun est la recherche d'un équilibre entre fonctionnalité, coût et facilité d'utilisation. Le système de Wen et al. [4], qui utilise le contrôle gestuel, une forme naturelle de communication homme-machine adaptée aux environnements à haut risque tels que les zones nucléaires, est particulièrement remarquable.

Malgré les progrès réalisés, aucun des projets examinés n'inclut d'affichage permettant de visualiser le canal de communication et l'état du système, comme proposé dans le présent développement. Cet ajout améliore la traçabilité et le débogage en cours d'exploitation. Associé à des filtres de lissage du signal du capteur, un tel système est plus fiable et applicable en environnement réel.

2. Analyse comparative des filtres de traitement des données des capteurs

Pour garantir la précision et la stabilité de la reconnaissance gestuelle, il est essentiel de choisir un algorithme de filtrage approprié pour le traitement des données du capteur. Le tableau 2 présente une analyse comparative des filtres utilisés dans différents projets utilisant des capteurs inertiels pour le contrôle de mouvement.

Tableau 1.2. Analyse comparative des filtres pour le traitement des données des capteurs

Filtre utilisé	Capteur	Microcontrôleur	Application	Raison de l'utilisation du filtre	Article
Filtre de Kalman	MPU6050	NodeMCU	Contrôle de la souris par geste	Précision accrue ; moins de latence et dépendance à la caméra.	[6]
Filtre complémentaire	MPU6050	ESP8266 NodeMCU	Mesure inertielle pour les pompiers	Précision du MPU6050 (95 %) avec faible complexité.	[7]
Filtre de Kalman	MPU6050	Arduino Uno	Contrôle gestuel du drone	Réduction du bruit ; contrôle précis du drone.	[8]
Filtre de Kalman	MPU6050	Arduino Uno	Système de reconnaissance des gestes	Meilleure estimation des angles malgré le bruit.	[9]

À la suite de l'analyse de la littérature présentée dans le tableau 2, on peut conclure que lors de la mise en œuvre de projets utilisant des capteurs MPU6050 pour le contrôle des gestes et le suivi inertiel, le choix de l'algorithme de filtrage joue un rôle clé dans la précision, la robustesse et la stabilité du système.

2.1 Filtres de visualisation utilisés (filtre utilisé)

Le tableau montre que trois des quatre projets utilisent un filtre de Kalman, tandis qu'un seul, celui de Juwita et al. [7] - s'appuie sur un filtre complémentaire. Cela indique déjà une pratique établie : pour les projets nécessitant une grande précision dans l'estimation de la position angulaire et une réponse rapide, le filtre de Kalman est l'approche privilégiée. Le filtre de Kalman est choisi dans les projets de Prasanth et al. [6], Setiawan et al. [8] et Rahmad et al. [9], où un contrôle gestuel stable est requis ou dans le contexte des drones volants - des systèmes où les conditions dynamiques et la précision des capteurs sont d'une importance primordiale. Seul le projet de Juwita et al. [7] choisit le filtre complémentaire en raison de sa complexité de calcul plus faible et de sa mise en œuvre plus simple, suffisante pour le positionnement intérieur des pompiers.

2.2 Capteur et microcontrôleur

Tous les projets utilisent le MPU6050, qui combine un gyroscope et un accéléromètre 3 axes. Les différences résident dans la plateforme de contrôle : NodeMCU est utilisé dans Prasanth et al. [6] et Juwita et al. [7], offrant une connectivité Wi-Fi intégrée, pratique pour les applications IoT. Arduino Uno est utilisé dans les deux autres projets, un choix motivé par la stabilité, la simplicité de programmation et la large compatibilité. Le choix du microcontrôleur est lié aux objectifs du projet : les calculs lourds du filtre de Kalman nécessitent une plateforme légèrement plus performante ou une implémentation optimisée.

2.3 Application et contexte d'utilisation

Les conceptions diffèrent dans leur objectif, mais toutes sont unies par l'utilisation de gestes comme entrée de contrôle. Prasanth et al. [6] mettent en œuvre le contrôle de la souris d'ordinateur par des gestes, où la précision et une faible latence sont recherchées - ce que le filtre de Kalman fournit grâce à son

schéma de lissage récursif du bruit. Setiawan et al. [8] et Rahmad et al. [9] appliquent le filtre au contrôle des drones, où la précision du tangage et du roulis et la réponse rapide sont vitales. Dans Juwita et al. [7], la tâche est plus générale - suivre le mouvement dans un bâtiment, où la facilité de calcul du filtre complémentaire est tout à fait suffisante.

2.4 Raisons du choix du filtre

La colonne « Raison d'utiliser le filtre » présente des arguments importants : le filtre de Kalman est utilisé en cas de mouvement dynamique et de nécessité de stabiliser le signal de sortie. Le filtre complémentaire est une alternative plus légère lorsque le système est plus simple ou qu'une prédiction d'état complexe n'est pas requise.

2.5 Conclusion

L'analyse comparative réalisée sur quatre projets utilisant le MPU6050 pour le contrôle gestuel et le suivi inertiel montre que le choix de l'algorithme de filtrage a un impact significatif sur la précision, la stabilité et l'efficacité du système. Trois projets sur quatre utilisent le filtre de Kalman, une pratique établie pour les projets nécessitant une estimation précise des positions angulaires, un temps de réponse en temps réel et une minimisation du bruit des capteurs, notamment dans des applications telles que le contrôle de drones et les mouvements de souris d'ordinateur. Le filtre de Kalman, bien que plus complexe en termes de calcul, offre un mécanisme récursif qui le rend adapté aux environnements dynamiques avec des paramètres en constante évolution.

Le filtre complémentaire est privilégié dans les conceptions plus simples, comme celle de Juwita et al. [7], où la complexité de calcul limitée et les exigences de précision moindres en font une solution pratique, par exemple pour le positionnement intérieur des pompiers dans un bâtiment, dans un

environnement relativement stable. Cela souligne l'importance du contexte d'application dans le choix d'une méthode de filtrage appropriée.

De plus, l'analyse des microcontrôleurs utilisés montre une corrélation claire entre le choix de la plateforme NodeMCU ou Arduino Uno et la complexité de l'algorithme. Des plateformes plus puissantes comme NodeMCU sont privilégiées lorsqu'une connectivité Wi-Fi ou des performances supérieures pour le traitement de filtres tels que Kalman sont requises. Malgré ces différences, tous les projets examinés présentent une tendance commune : l'utilisation des gestes comme principale entrée de commande, ce qui nécessite un traitement fiable des données des capteurs pour obtenir un comportement système intuitif, précis et stable.

On peut conclure que la réussite de la mise en œuvre du contrôle gestuel à l'aide de capteurs inertiels dépend en grande partie d'une combinaison judicieuse d'algorithme de filtrage, de plateforme matérielle et d'objectifs spécifiques du projet. Les filtres de Kalman et complémentaires ont leur place dans la conception de tels systèmes, et la clé de leur efficacité réside dans l'équilibre entre précision, complexité et applicabilité.

CHAPITRE 2: PARTIE THEORIQUE

Afin de mettre en œuvre un système de contrôle gestuel efficace pour un bras robotisé, il est nécessaire de construire une base théorique couvrant les principaux composants technologiques et techniques. Ce chapitre aborde les principes de fonctionnement des robots industriels, les spécificités de la plateforme de microcontrôleurs utilisée, les capacités de communication radio, ainsi que les capteurs et les algorithmes de traitement de données appropriés. Cet exposé sert de base à la construction de l'architecture globale du système, présentée dans le chapitre suivant.

1. Introduction à la robotique et à la plateforme Arduino

Un robot industriel est une machine multifonctionnelle programmable et contrôlée automatiquement, dotée de plusieurs degrés de liberté, qui est installée dans un système de production automatisé. Les robots industriels modernes se caractérisent par leur capacité à être programmés dans différents langages, le plus souvent C et C++, puis Python, et parfois Java, C#/.NET et Arduino. Ils sont capables de déterminer des positions dans un système de coordonnées avec une grande précision, de gérer les entrées et les sorties, de traiter des informations et de communiquer avec les utilisateurs et d'autres appareils via des bus de terrain ou des réseaux locaux.

L'une des principales tâches actuelles est l'automatisation de la production discrète à l'aide de robots industriels, de modules technologiques automatisés et de lignes automatisées complexes. Les robots sont utilisés non seulement pour accroître l'efficacité, mais aussi pour remplacer les humains dans les activités dangereuses ou dangereuses, ainsi que dans les opérations lourdes et monotones.

Parmi les exemples d'applications modernes, on peut citer les robots industriels pour la fabrication, les robots miniers et de reconnaissance, les robots spatiaux pour l'exploration planétaire, les robots personnels pour les tâches ménagères ou sociales, les robots médicaux pour le diagnostic, la chirurgie et les soins, les robots militaires à des fins de reconnaissance et de combat, ainsi que la nanobotique pour la microchirurgie et les procédures médicales avancées.

Arduino est une carte microcontrôleur programmable. Un microcontrôleur est un composant d'un circuit électronique qui détermine le signal de sortie en fonction des données d'entrée reçues du monde physique via des périphériques d'entrée-sortie, tels que des capteurs. On peut supposer qu'il est analogue au cerveau humain.[10] Arduino est une plateforme électronique open source basée sur du matériel et des logiciels faciles à utiliser. Les cartes Arduino peuvent lire les données d'entrée, par exemple celles d'un capteur de lumière, d'un capteur de température, d'un microphone, d'un bouton, etc., et les convertir en sortie (activation d'un moteur, allumage d'une LED, etc.). Le microcontrôleur Arduino est contrôlé par l'envoi d'un ensemble d'instructions. Pour ce faire, le langage de programmation Arduino (basé sur le câblage) et le logiciel Arduino (IDE), basé sur le traitement, sont utilisés. Grâce à son interface utilisateur simple et accessible, Arduino a été utilisé dans des milliers de projets et d'applications différents. Arduino offre un environnement logiciel intuitif qui facilite la mise en œuvre d'applications, du simple contrôle de LED au développement de systèmes complexes. Il est compatible avec Mac, Windows et Linux.

Applications modernes des robots :

- **Robots industriels** - Ce type de robot convient aux tâches qui peuvent être définies avec précision et qui doivent être exécutées de la même manière sans avoir besoin de beaucoup de retour d'information pour contrôler la manière spécifique dont l'action a été effectuée.

- **Robots miniers** - En raison de la nature de l'activité dans les mines, en particulier souterraines, les robots autonomes, semi-autonomes et télécommandés ont récemment été largement utilisés.

- **Robots de reconnaissance** - Ils peuvent travailler dans des environnements difficiles ou dangereux pour l'homme, par exemple pour le déminage, les travaux sous-marins, l'exploitation minière et le nettoyage des déchets toxiques. De nombreux exemples de ce type de robots peuvent être cités, comme les véhicules autoguidés, les robots mobiles utilisés dans de grandes installations comme les entrepôts et les hôpitaux, et les ports pour la manutention de conteneurs, le transport de marchandises, ou même les patrouilles de sécurité. Un autre exemple de ce type est l'exploration d'objets spatiaux lointains, comme Mars, qui est également réalisée par des robots contrôlés depuis la Terre. Un problème dans ce cas est le long temps nécessaire au signal pour voyager vers Mars et en revenir.

- **Robots spatiaux** - Des robots humanoïdes spéciaux sont en cours de développement pour l'exploration spatiale. La NASA développe des robots en collaboration avec d'autres organisations qui seront utilisés dans des conditions de vol spatial, ainsi que pour l'exploration d'autres planètes.

- **Robots personnels** - Il existe sur le marché des robots domestiques qui effectuent des tâches simples comme passer l'aspirateur ou tondre la pelouse. D'autres robots domestiques sont conçus pour tenir compagnie aux gens ou jouer avec eux, comme un chien robot ou un robot humanoïde conçu pour les personnes âgées et handicapées.

- **Robots médicaux** - Les robots médicaux sont utilisés dans divers domaines médicaux, notamment la chirurgie, le diagnostic et les soins médicaux. Il existe différentes installations pour les soins aux patients. L'objectif de la robotique est de créer un robot capable d'assister les patients alités 24 heures sur 24, à la place d'un professionnel de la santé.

- **Robots militaires** - Robots utilisés à des fins de reconnaissance et de combat militaires. Ils peuvent se déplacer de manière autonome dans les airs, sur terre et sous l'eau. Il s'agit, par exemple, de divers types de drones militaires. Ils sont utilisés à des fins de reconnaissance et de combat.

- **Nanobotique** - La nanobotique, ou nanorobotique, est une technologie permettant de créer des machines ou des robots dont les composants sont de l'ordre du nanomètre. L'objectif est de créer des robots de la taille d'un virus ou d'une bactérie, capables d'effectuer des tâches à l'échelle microscopique. Les domaines d'application possibles incluent la microchirurgie (au niveau cellulaire), la production d'armes, etc.

2. Types de bras robotisés

Les différents types de bras robotisés comprennent 6 types principaux :

- Un bras robotique articulé (Figure 2.1) ressemble à une main humaine et permet des mouvements et des configurations mécaniques. C'est l'un des types de bras robotisés les plus courants pour l'automatisation industrielle. Il se caractérise par un bras mécanique unique fixé à une base par une articulation rotative.[11]

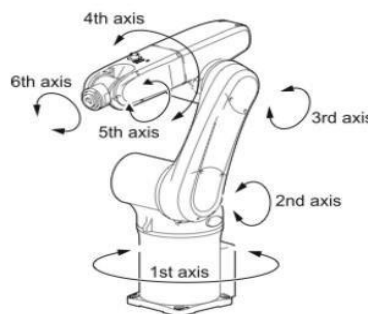


Figure 2.1 - Modèle de bras robotisé articulé[12]

- Les bras robotiques cartésiens (Figure 2.2) sont également appelés robots linéaires ou robots portiques. Ce sont des types de robots industriels qui fonctionnent sur trois axes linéaires utilisant le système de coordonnées

cartésiennes (X, Y et Z). Cela signifie qu'ils se déplacent en ligne droite le long de trois axes différents (haut et bas, intérieur et extérieur, et latéral). C'est pourquoi ce type de bras robotique est très populaire auprès des industriels et des fabricants qui recherchent une plus grande flexibilité dans leurs configurations [11].

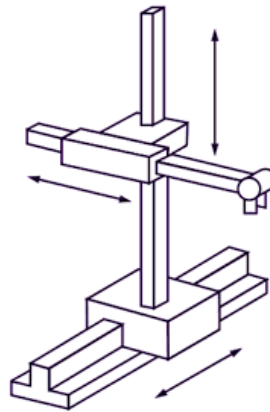


Figure 2.2 - Modèle de bras robotisé cartésien[13]

- Les robots cylindriques (Figure 2.3) sont un autre type de bras robotisé conçu autour d'un bras unique capable de se déplacer verticalement. Ils sont dotés d'une articulation pivot à la base et d'une articulation prismatique reliant les maillons. Ces robots sont très compacts et permettent des tâches simples et de petite taille, telles que l'assemblage, l'entretien de machines ou les applications de revêtement. Ce type de bras robotisé est constitué d'un arbre rotatif et d'un bras extensible permettant des mouvements verticaux et coulissants.[11]

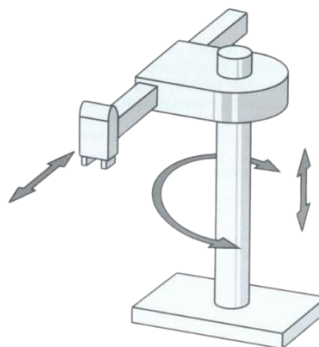


Figure 2.3 - Modèle de robot cylindrique[14]

- Les bras robotisés Delta (Figure 2.4) sont également appelés « bras robotisés parallèles » car ils permettent de connecter trois bras à une base unique montée au-dessus d'un espace de travail. Ces types de bras robotisés offrent des vitesses élevées et sont donc utilisés pour l'automatisation.[11]

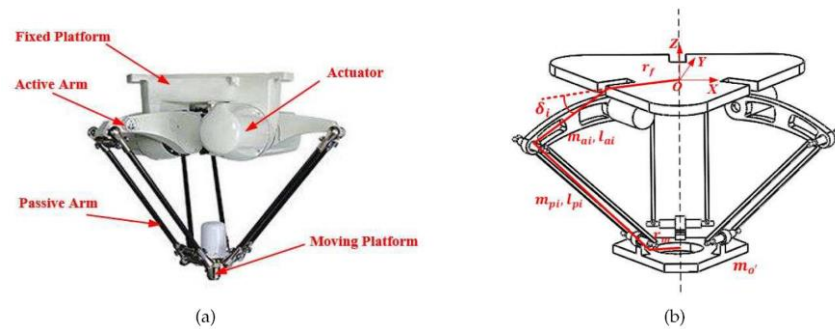


Figure 2.4 - Modèle de bras robotisé Delta[15]

- Bras robotisé d'assemblage à compliance sélective. Le bras robotisé articulé est appelé SCARA (Figure 2.5). Le mouvement à trois axes (X, Y et Z) des robots SCARA est combiné à un mouvement rotatif. Comparés aux robots cartésiens, les robots SCARA sont plus performants en mouvement latéral, plus rapides et plus faciles à intégrer. Les robots SCARA sont couramment utilisés pour les applications biomédicales, la palettisation et l'assemblage.[11]

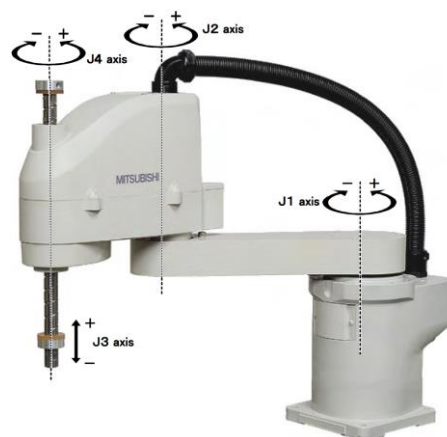


Figure 2.5 - Modèle de bras robotisé SCARA[16]

- Les robots polaires, également appelés robots sphériques (Figure 2.6), possèdent une base reliée à un bras par deux articulations rotatives et une articulation linéaire. Les axes du robot se combinent pour créer une coordonnée polaire, lui permettant d'avoir une enveloppe de travail sphérique. L'une des premières catégories de robots industriels jamais créées fut le robot polaire. Le moulage par injection, le soudage et la manutention sont des applications courantes des robots polaires [11].

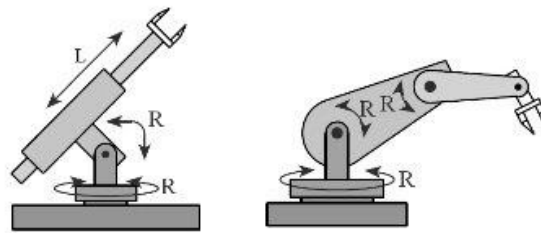


Figure 2.6 - Modèle de robot polaire[17]

3. Communications radio

La radiocommunication est une méthode de transmission d'informations utilisant des ondes électromagnétiques, capables de transporter des informations sur de longues distances sans nécessiter de connexion physique (câble). Son principe de base consiste à transmettre des signaux électriques modulés et émis sous forme d'ondes radio par l'émetteur, puis reçus et démodulés par le récepteur. Ce processus implique une modulation de fréquence ou d'amplitude, la sélection d'un canal d'émission et de réception, et le contrôle de la gamme de fréquences afin de minimiser les interférences. L'avantage de la radiocommunication réside dans sa mobilité, sa flexibilité et sa capacité à surmonter les obstacles physiques tels que les murs et les distances, ainsi que dans la possibilité de construire des réseaux complexes permettant à plusieurs appareils d'échanger des données de manière efficace et sécurisée.

Dans le domaine de la robotique, la communication radio permet le contrôle à distance, ce qui est particulièrement important lorsque l'opérateur doit rester à distance de sécurité ou lorsque le système fonctionne dans un environnement à accès limité. Avec les plateformes Arduino, on utilise généralement des modules radio compacts, offrant un bon débit, une faible consommation d'énergie et une intégration logicielle aisée.

Exemples de modules radio :

- **FS1000A / XY-MK-5V**

Les FS1000A (figure 2.7) sont populaires en raison de leur faible coût et de leur intégration facile avec Arduino, mais ils présentent des limites qui doivent être prises en compte dans les projets nécessitant une communication sans fil fiable.[18]



Figure 2.7 - Image du module FS1000A / XY-MK-5V[19]

- **HC-12**

Le module HC-12 (figure 2.8) est un émetteur-récepteur RF semi-duplex conçu pour fonctionner dans la bande ISM 433 MHz avec jusqu'à 100 canaux de communication avec une séparation de 400 kHz, une portée jusqu'à 1,8 km en plein air et une interface UART. [20]



Figure 2.8 - Image du module HC-12[21]

- **RFM69**

Le RFM69HCW(figure 2.9) est un module radio économique et polyvalent. Il permet d'envoyer du texte ou des données binaires entre deux ou plusieurs centaines de modules. [22]



Figure 2.9 - Image du module RFM69[23]

- **RFM95 / SX1278**

L'émetteur-récepteur LoRa Connect SX1278(figure 2.10) dispose d'un modem LoRa pour une communication à spectre étalé à très longue portée. [24]



Figure 2.10 - Image du module RFM95 / SX1278[25]

- **XBee Series 1 / 2**

Les modules XBee(figure 2.11) sont de petits appareils radio qui utilisent divers protocoles de communication pour envoyer des informations, telles que des données de capteurs. [26]



Figure 2.11 - Image du module XBee Series 1 / 2[27]

- **NRF24L01+**

Le module nRF24L01+(figure 2.12) est un émetteur-récepteur radio conçu pour fonctionner dans la bande ISM 2,4 GHz de 2,400 GHz à 2,525 GHz avec jusqu'à 126 canaux de communication. [28]



Figure 2.12 - Image du module NRF24L01+[29]

Le NRF24L01+ est reconnu pour sa compacité sa faible consommation et la fiabilité des échanges de données entre l'émetteur (gant capteur) et le récepteur (bras robotisé). De plus, le module permet la sélection et l'adressage des canaux, ce qui minimise les risques d'interférences et permet un fonctionnement dans des environnements avec plusieurs appareils sans fil. Grâce à sa vitesse de transmission élevée (jusqu'à 2 Mbit/s) et à ses mécanismes d'accusé de réception de paquets intégrés, ce module garantit une communication fiable, même en présence d'interférences radio. Un autre avantage est sa compatibilité avec Arduino et la richesse de la documentation et des exemples disponibles, ce qui facilite son intégration au développement. Le choix de ce module est stratégique pour le projet, car il allie caractéristiques techniques et accessibilité pratique, offrant une base solide pour un contrôle sans fil avec un délai minimal et une grande précision.[30]

4. Détecteurs de mouvement

Les capteurs de mouvement jouent un rôle essentiel dans les systèmes de contrôle gestuel et les bras robotisés. Ils permettent de capturer la position, l'orientation et la dynamique des mouvements du bras humain et de les convertir en signaux de commande pour le système robotisé. Cela offre une interface homme-machine intuitive et flexible, utilisable dans des applications industrielles, médicales ou éducatives.

Les capteurs de mouvement les plus couramment utilisés dans ce type de projet sont les unités de mesure inertielle (IMU). Elles combinent un accéléromètre (pour mesurer l'accélération linéaire) et un gyroscope (pour mesurer la vitesse angulaire), et parfois un magnétomètre (pour déterminer l'orientation par rapport au champ magnétique terrestre). Les IMU sont compactes, économes en énergie et fournissent un retour d'information rapide, ce qui les rend idéales pour une utilisation dans les appareils portables ou connectés.

Exemples de capteurs IMU :

- **BNO055- 9 DoF**

Accéléromètre + gyroscope + magnétomètre.

Le BNO055 (figure 2.13) est un capteur intelligent avec fusion de capteurs intégrée fournit directement des données d'orientation, sans nécessiter de calculs complexes par le microcontrôleur.[19]



Figure 2.13 - Image du capteur BNO055[32]

- **LSM6DSOX (6 DoF)**

Accéléromètre + Gyroscope

Le LSM6DSOX(figure 2.14) est un capteur de pointe de STMicroelectronics intégrant des fonctions de fusion de données de capteurs et d'apprentissage automatique. [20]



Figure 2.14 - Image du capteur LSM6DSOX[34]

- **MPU-9250 (9 DoF)**

Accéléromètre + gyroscope + magnétomètre.

Le MPU-9250(figure 2.15) est un système intégré (SiP) combinant deux puces : le MPU-6500, qui contient un gyroscope 3 axes, un accéléromètre 3 axes et un processeur de mouvement numérique intégré. [35]

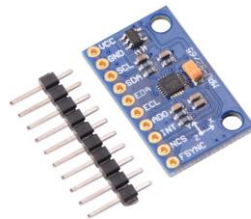


Figure 2.15 - Image du capteur MPU-9250[36]

- **MPU-6050 (6 DoF)**

Accéléromètre + gyroscope.

Le MPU-6050 (figure 2.16) combine un gyroscope 3 axes, un accéléromètre 3 axes et un processeur de mouvement numérique communiquant via une interface I2C, extrêmement populaire en raison de son prix abordable et de sa facilité d'intégration [37].



Figure 2.16 - Image du capteur MPU-6050[38]

Le MPU6050 offre un excellent compromis entre prix, performances et documentation disponible. Il fonctionne selon le principe MEMS, utilisant des structures mécaniques miniatures pour détecter les mouvements [39]. Lorsque la position change, la masse à l'intérieur du capteur se déplace, créant un signal électrique interprété par le microcontrôleur. Placé sur la main (par exemple, au poignet), le MPU6050 peut surveiller des mouvements tels que l'inclinaison, la rotation et le levage, permettant ainsi de contrôler un bras robotisé.

5. Filtres - passe-haut et passe-bas

Dans les systèmes de contrôle par capteurs, ainsi que dans le traitement des données de capteurs, le traitement du signal joue un rôle essentiel pour garantir la précision et la fiabilité des informations reçues. Les données brutes des capteurs contiennent souvent du bruit, c'est-à-dire des fluctuations ou des écarts indésirables pouvant entraîner une mauvaise interprétation des mouvements, des positions ou des états des objets. Par conséquent, différents types de filtres sont utilisés pour minimiser ces distorsions et améliorer la qualité de l'analyse.

Les filtres passe-haut sont utilisés pour supprimer les basses fréquences des signaux haute fréquence (par exemple, dans les circuits d'antenne et RF), pour séparer les signaux basse et haute fréquence en canaux séparés (comme dans les signaux audio et vidéo), pour construire des circuits résonants sélectifs pour les hautes fréquences, pour façonner les caractéristiques de transmission des systèmes électroniques en bloquant les basses fréquences indésirables et pour adapter l'impédance entre les circuits ayant des propriétés de fréquence différentes[40].

Les filtres passe-bas sont utilisés dans divers domaines, tels que : la suppression des composants interférents à haute fréquence (bruit) des signaux basse fréquence (dans les circuits audio, de télécommunication, etc.), la séparation des signaux basse fréquence et haute fréquence (par exemple, dans les signaux audio et vidéo des circuits d'antenne), la construction de circuits résonants (par exemple, dans les circuits de contrôle), l'adaptation de l'impédance entre les circuits ayant des caractéristiques de fréquence différentes, ainsi que la mise en forme des caractéristiques de transmission des canaux d'entrée/sortie des appareils[40].

Pour les applications plus complexes, des méthodes combinées ou des algorithmes adaptatifs tels que **le filtre complémentaire** et **le filtre de Kalman** sont souvent utilisés. Le filtre complémentaire combine les données de l'accéléromètre et du gyroscope, s'appuyant sur le fait que l'accéléromètre est plus précis à long terme et le gyroscope à court terme. Le filtre de Kalman est un algorithme statistique qui combine de manière optimale les mesures et les valeurs prédites afin de réduire le bruit et de fournir la meilleure estimation de l'état du système [41]. En raison de sa flexibilité et de sa précision, le filtre de Kalman a été choisi pour ce projet.

CHAPITRE 3: REALISATION DU SYSTEME DE CONTROLE A DISTANCE

Ce chapitre présente la mise en œuvre concrète du système de contrôle gestuel à distance d'un bras robotisé. Il décrit en détail les composants matériels et logiciels utilisés, ainsi que l'architecture générale du système. L'objectif est d'exposer les différentes étapes de la conception et du développement, en mettant l'accent sur les choix techniques réalisés et leur justification. Cette partie vise à illustrer comment les principes théoriques abordés précédemment ont été traduits en une solution pratique et fonctionnelle.

1. Vue d'ensemble du système

Le système de contrôle gestuel d'un bras robotique est complexe, composé de composants matériels et logiciels fonctionnant en synchronisation. Il comprend deux blocs principaux: un élément émetteur, porté par l'opérateur (un gant), et un élément récepteur, qui contrôle le bras robotique. Les deux éléments sont connectés sans fil grâce à des modules radio NRF24L01+.

L'émetteur contient des capteurs de mouvement (IMU), un microcontrôleur (Arduino Nano), un module de communication sans fil et un système d'alimentation. Le récepteur est équipé d'un microcontrôleur (Arduino Mega), d'un module radio du même modèle et d'une interface de contrôle du bras robotisé (servomoteurs, pilotes et mécanique).

La figure 3.1 représente un élément clé pour comprendre la structure et le fonctionnement de l'ensemble du système.

Le schéma fonctionnel illustre l'architecture du système de contrôle gestuel d'un bras robotisé.

À gauche, l'émetteur représente la partie émettrice, portée par l'opérateur et contenant :

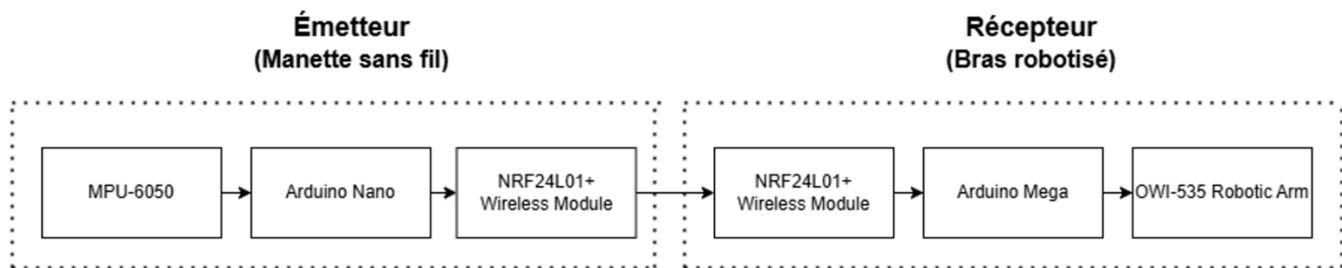


Figure 3.1 - Vue générale du système

- **MPU-6050** - capteur de mouvement qui détecte les vitesses angulaires et les accélérations linéaires ;
- **Arduino Nano** - un microcontrôleur qui lit les données du capteur et effectue le traitement ;
- **Module sans fil NRF24L01+** - un module radio qui envoie les données traitées au récepteur.

Le côté droit - Récepteur représente la partie réceptrice :

- **Module sans fil NRF24L01+** - reçoit les données sans fil de l'émetteur ;
- **Arduino Mega** - un microcontrôleur plus puissant qui interprète les commandes reçues ;
- **Bras robotique OWI-535** - la partie mécanique qui exécute les mouvements selon les instructions.

La flèche entre les deux parties indique une communication sans fil, mise en œuvre via une liaison radio.

2. Partie transmission

2.1 Vue générale du système

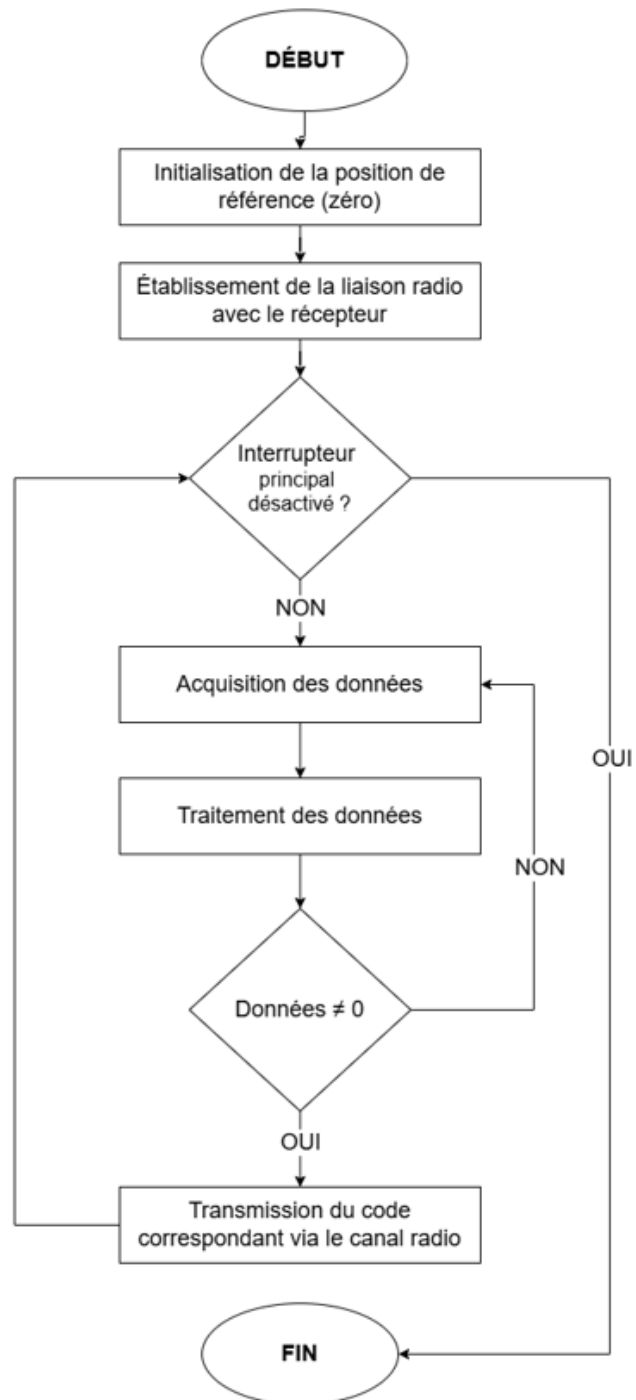


Figure 3.2 - Algorithme de fonctionnement de la partie émettrice

La partie transmission est au cœur du système. Sa fonction principale est de capturer les mouvements de la main et de les convertir en commandes de contrôle correspondantes, transmises à la main robotisée. Ce module est essentiel au fonctionnement global du système, car il assure la connexion entre les gestes humains et l'exécution mécanique des actions par le robot. Pour atteindre cet objectif, une solution matérielle et logicielle a été développée, comprenant plusieurs composants soigneusement sélectionnés et complémentaires.

Parmi les principaux éléments de la partie transmission figure le module capteur MPU6050, qui combine un accéléromètre et un gyroscope, fournissant des données complexes sur l'accélération angulaire et l'orientation dans l'espace. Ces données sont traitées par un microcontrôleur Arduino, qui sert d'unité centrale pour la collecte, le filtrage et la transmission des informations. Pour garantir une précision et une fiabilité accrues des mesures, un filtre de Kalman est intégré au système. Il permet de réduire le bruit des données brutes du capteur et de générer des valeurs plus stables et plus réalistes de la position de la main.

Après traitement, les données sont transmises sans fil via le module radio NRF24L01+, qui assure une communication rapide et fiable avec la partie réceptrice du système. Le choix de ce module est motivé par sa haute efficacité énergétique, sa faible latence et sa large application dans les communications radiofréquences à courte portée.

Dans le cadre du projet, une sélection rigoureuse de tous les composants matériels a été effectuée, en fonction de leurs caractéristiques techniques et de leur compatibilité avec l'ensemble du système. Chaque étape de la construction de la partie transmission - de la conception et de la connexion à la programmation et à l'étalonnage - a été réalisée dans le but d'atteindre une efficacité maximale et de respecter les exigences académiques et techniques en matière de fiabilité, de durabilité et de fonctionnalité.

2.2 Sélection du capteur

Le MPU6050 a été choisi pour sa combinaison d'un accéléromètre et d'un gyroscope sur une seule puce, permettant ainsi de mesurer les inclinaisons, les accélérations et les vitesses angulaires. Ce capteur est ainsi idéal pour le contrôle gestuel, car il peut détecter des mouvements complexes de la main. De plus, le MPU6050 intègre la technologie de traitement numérique du mouvement (DMP), qui facilite l'extraction des données traitées. La connexion à l'Arduino s'effectue via une interface I2C, ce qui permet d'utiliser seulement deux broches de communication.

2.3 Sélection du module radio

Le choix du module radio NRF24L01+ repose sur ses avantages : faible consommation d'énergie, débit de transmission élevé (jusqu'à 2 Mbps) et bonne intégration avec les plateformes Arduino. De plus, son format compact permet d'adresser et de sélectionner un canal, minimisant ainsi les risques d'interférences. Le NRF24L01+ utilise une interface SPI pour une connexion rapide au microcontrôleur. Il est donc particulièrement adapté aux projets nécessitant une communication sans fil stable et fiable.

2.4 Connexion à Arduino

Le MPU-6050 se connecte à l'Arduino Nano via une interface I2C (SDA, SCL). Le NRF24L01+ utilise une interface SPI (MOSI, MISO, SCK, CSN, CE). L'alimentation est assurée par une batterie via un régulateur qui fournit une tension stable de 3,3 V pour le module radio et de 5 V pour le reste des composants. Le condensateur entre VCC et GND du NRF24L01+ est important pour éviter les chutes de tension lors des pics de transmission. Les LED connectées via des résistances signalent l'activité du système, la transmission de données et la présence d'erreurs.

2.5 Traitement des données

Le filtre de Kalman a été choisi pour le traitement des données.

Le filtre de Kalman a été choisi pour ce projet car il s'agit d'un algorithme largement utilisé pour l'estimation optimale de l'état des systèmes dynamiques, notamment en présence de bruit et d'incertitude. Grâce à sa rigueur mathématique et à son adaptabilité, il est particulièrement adapté aux applications temps réel, comme c'est le cas pour le contrôle gestuel d'un bras robotique.

Le principe de base du filtre de Kalman repose sur un processus itératif comprenant deux étapes principales : la prédiction et la mise à jour. Lors de la phase de prédiction, le filtre utilise un modèle mathématique du système pour prédire l'état suivant à partir du précédent. Lors de la deuxième étape, la mise à jour, cet état prédit est mis à jour à l'aide des nouvelles mesures reçues du capteur. De cette manière, un équilibre est atteint entre la confiance dans le modèle (la prédiction) et la confiance dans les données mesurées, en tenant compte des caractéristiques statistiques du bruit.

Dans ce projet, le filtre de Kalman est appliqué aux données du capteur inertiel MPU6050, qui fournit des informations sur l'accélération linéaire et la vitesse angulaire. Malgré sa haute sensibilité, le capteur est sensible à divers types de bruit, provenant à la fois de l'environnement et des déviations électroniques internes. Ces bruits peuvent entraîner des imprécisions dans la reconnaissance gestuelle, essentielle pour le contrôle d'un système mécanique en temps réel. Le filtre de Kalman élimine ces distorsions en fournissant l'estimation la plus probable de l'état actuel de la main (par exemple, inclinaison, angle, position).

Ce modèle permet au système non seulement d'éliminer les fluctuations momentanées des données, mais aussi de s'adapter aux conditions changeantes en temps réel. Ceci est essentiel lors de l'interaction avec les commandes gestuelles,

où tout retard ou mauvaise interprétation peut entraîner un mouvement incorrect du bras robotisé. Comparé à des méthodes plus simples comme le filtre complémentaire ou le lissage de la moyenne arithmétique, le filtre de Kalman offre une précision et une fiabilité nettement supérieures, notamment dans les systèmes hautement dynamiques.

Par conséquent, sa mise en œuvre dans le projet actuel contribue à la stabilité et à la précision du système, garantissant que le contrôle gestuel est effectué avec un délai minimal et une précision maximale.

2.6 Programmation d'un module de capture de mouvement

Le code de l'émetteur est un composant clé du système de contrôle d'un robot industriel par mouvements manuels. Il utilise le capteur MPU6050, qui combine un gyroscope et un accéléromètre pour suivre les orientations spatiales (lacet, tangage, roulis), et un module sans fil NRF24L01+ pour transmettre les commandes traitées au récepteur du robot. L'architecture du programme combine plusieurs fonctions principales : l'initialisation matérielle, le traitement des données par l'algorithme de Kalman, la reconnaissance des gestes et leur transmission par radio.

- **Initialisation et étalonnage du MPU6050 et du DMP**

Dans la figure 3.3, voici un exemple du code essentiel au bon fonctionnement du module inertiel MPU6050. Initialement, le module est initialisé à l'aide de la fonction ***mpu.initialize()***, qui prépare la communication I2C et charge les registres internes nécessaires. Ensuite, le **DMP (Digital Motion Processor)** est activé: un processeur intégré au MPU6050 qui calcule des orientations spatiales complexes sans que le processeur Arduino n'ait à effectuer de lourdes opérations mathématiques.

Les valeurs de décalage du gyroscope et de l'accéléromètre sont définies manuellement pour compenser l'erreur d'usine du capteur, ce qui permet

des mesures plus précises. L'étalonnage avec *CalibrateAccel()* et *CalibrateGyro()* est un processus automatisé où le capteur « apprend » à reconnaître son état zéro.

Après une initialisation réussie (*devStatus == 0*), DMP est activé avec *setDMPEnabled(true)* et une interruption externe est définie avec *attachInterrupt*, qui avertit le programme lorsque de nouvelles données sont disponibles. Cela améliore considérablement l'efficacité du programme en éliminant la nécessité d'interroger constamment de nouvelles valeurs.

```
mpu.initialize();
pinMode(INTERRUPT_PIN, INPUT);
devStatus = mpu.dmpInitialize();

mpu.setXGyroOffset(-86);
mpu.setYGyroOffset(33);
mpu.setZGyroOffset(26);
mpu.setZAccelOffset(1820);

if (devStatus == 0) {
    mpu.CalibrateAccel(6);
    mpu.CalibrateGyro(6);
    mpu.PrintActiveOffsets();
    Serial.println(F("Enabling DMP..."));
    mpu.setDMPEnabled(true);

    Serial.print(F("Enabling interrupt detection (Arduino external interrupt "));
    Serial.print(digitalPinToInterrupt(INTERRUPT_PIN));
    Serial.println(F(")..."));
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    Serial.println(F("DMP ready! Waiting for first interrupt..."));
    dmpReady = true;
    packetSize = mpu.dmpGetFIFOPacketSize();
}
```

Figure 3.3 - Initialisation et configuration du MPU6050 et du DMP

- **Configuration de l'émetteur sans fil NRF24L01+**

Ici, le module sans fil NRF24L01+ est mis en place, qui effectue la communication radio entre la partie émettrice et réceptrice du système. *radio.begin()* démarre le module et *openWritingPipe(pipe)* définit une adresse de transmission unique, qui doit correspondre à celle du périphérique récepteur. L'utilisation de *setPALevel(RF24_PA_MAX)* garantit une puissance de transmission maximale, idéale pour une communication stable sur de longues distances. *setDataRate(RF24_1MBPS)* détermine la bande passante et la vitesse de transfert. L'activation de l'accusé de réception automatique avec *setAutoAck(1)* permet de vérifier la bonne réception du message. En l'absence de réponse, *setRetries(2, 15)* définit le nombre et l'intervalle des tentatives, augmentant ainsi la fiabilité du système.

```
radio.begin();  
radio.openWritingPipe(pipe);  
radio.setPALevel(RF24_PA_MAX);  
radio.setDataRate(RF24_1MBPS);  
radio.setAutoAck(1);  
radio.setRetries(2, 15);
```

Figure 3.4 - Configuration de l'émetteur sans fil NRF24L01+

- **Extraction et filtration du lacet, du tangage et du roulis via un filtre de Kalman**

La partie du code illustrée à la Figure 3.5 est chargée de récupérer les données d'orientation du MPU6050 et de les traiter. Un DMP (Digital Motion Processor) est utilisé pour calculer un quaternion, une représentation mathématique des rotations dans l'espace tridimensionnel. Ce quaternion permet de calculer le vecteur de gravité et les trois angles d'orientation : lacet (rotation selon l'axe vertical), tangage (inclinaison avant/arrière) et roulis (inclinaison latérale).

La sortie de `dmpGetYawPitchRoll` est en radians, donc chaque valeur est convertie individuellement en degrés en multipliant par $180 / M_PI$.

Un filtre de Kalman est ensuite appliqué à chaque axe individuellement pour lisser les fluctuations et éliminer le bruit, ce qui donne des valeurs plus stables et plus fiables qui sont essentielles pour une reconnaissance correcte des gestes et l'envoi de commandes précises au bras robotique.

```
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

ypr[0] = ypr[0] * 180 / M_PI;
ypr[1] = ypr[1] * 180 / M_PI;
ypr[2] = ypr[2] * 180 / M_PI;

// Apply Kalman filter
kalmanYaw = kalmanUpdate(ypr[0], &kalmanYaw);
kalmanPitch = kalmanUpdate(ypr[1], &kalmanPitch);
kalmanRoll = kalmanUpdate(ypr[2], &kalmanRoll);
```

Figure 3.5 - Extraction et filtrage de Yaw, Pitch et Roll via un filtre de Kalman

- **Reconnaissance des gestes et transmission des commandes**

La reconnaissance des gestes est un mécanisme de base pour le contrôle du bras robotique, mis en œuvre par la mesure et l'analyse de l'orientation spatiale du bras humain en temps réel.

Les vérifications conditionnelles présentées dans la figure 3.6 représentent un système de reconnaissance de gestes basé sur une plage de valeurs.

```

else if ((kalmanYaw < -45) && (kalmanPitch > -40) && (kalmanPitch < 40)
        && (kalmanRoll > -40) && (kalmanRoll < 40)) {
    Serial.println("Base Left");
    SendMessage[0] = 2;
    radio.writeFast(&SendMessage, sizeof(SendMessage));
}

if ((kalmanYaw > -40) && (kalmanYaw < 40) && (kalmanPitch < -30)
    && (kalmanRoll > -40) && (kalmanRoll < 40)) {
    Serial.println("Down");
    SendMessage[0] = 3;
    radio.writeFast(&SendMessage, sizeof(SendMessage));
}

else if ((kalmanYaw > -40) && (kalmanYaw < 40) && (kalmanPitch > 30)
        && (kalmanRoll > -40) && (kalmanRoll < 40)) {
    Serial.println("Up");
    SendMessage[0] = 4;
    radio.writeFast(&SendMessage, sizeof(SendMessage));
}

```

Figure 3.6 - Reconnaissance de gestes et transmission de commandes

Tableau 3.1 - Correspondance des gestes et des conditions d'activation

L'équipe	Mouvement	Conditions
1	Rotation à droite	Yaw $> 45^\circ$, Pitch et Roll dans les limites centrales
2	Rotation à gauche	Yaw $< -45^\circ$, Pitch et Roll dans les limites centrales
3	Inclinaison vers le bas	Pitch $< -30^\circ$, Yaw et Roll en zone neutre
4	Inclinaison vers le haut	Pitch $> 30^\circ$, Yaw et Roll stables
5	Inclinaison à droite	Roll $< -60^\circ$, Yaw et Pitch proches de zéro
6	Inclinaison à gauche	Roll $> 60^\circ$, Yaw et Pitch proches de zéro
7	Combinaison : droite et haut	Yaw $< -25^\circ$, Roll $< -60^\circ$
8	Combinaison : droite et bas	Yaw $> 30^\circ$, Roll $< -60^\circ$
9	Combinaison : gauche et haut	Yaw $> 30^\circ$, Roll $> 60^\circ$
10	Combinaison : gauche et bas	Yaw $< -30^\circ$, Roll $> 60^\circ$

Le tableau 3.1 présente de manière synthétique les correspondances entre les gestes de l'opérateur et les conditions spécifiques d'activation du bras robotisé. Chaque mouvement est associé à un ensemble précis de seuils sur les axes Yaw, Pitch et Roll, permettant une identification claire et fiable des commandes. Cette approche basée sur des règles prédéfinies assure un contrôle intuitif et minimise le risque d'interprétation erronée des gestes.

- **Position neutre centrale (Centre) :**

Pour de faibles écarts sur tous les axes ($\pm 20^\circ$), l'état initial est interprété comme inactif. La commande 0 est envoyée, ce qui représente un arrêt complet (absence de mouvement).

Cette approche utilise des règles prédéfinies avec des seuils clairement définis, qui permettent une mise en œuvre facile en temps réel et ne nécessitent pas d'apprentissage automatique, tout en offrant un contrôle fiable lorsque les gestes sont clairement distinguables.

- **Commande de centrage (réinitialisation de position)**

Le code illustré à la figure 3.7 correspond à l'état « Centre », où tous les angles sont proches de zéro. Cela signifie que la main est en position initiale, ou neutre. Cet état est essentiel pour les systèmes de commande gestuelle, car il permet à l'opérateur de réinitialiser la commande ou d'attendre la commande suivante sans risque d'activation involontaire. Dans un contexte industriel, une telle commande peut également servir à effectuer un arrêt d'urgence ou à boucler le mécanisme en position de sécurité.

```
if ((kalmanYaw > -20) && (kalmanYaw < 20) && (kalmanPitch > -20)
    && (kalmanPitch < 20) && (kalmanRoll > -20) && (kalmanRoll < 20)) {
    Serial.println("Center");
    SendMessage[0] = 0;
    radio.writeFast(&SendMessage, sizeof(SendMessage));
}
```

Figure 3.7 - Commande de recentrage (remise à zéro de la position)

3. Communication sans fil

L'un des principaux composants fonctionnels du système de contrôle du bras robotisé développé est le module de communication sans fil, implémenté à l'aide des modules radio NRF24L01+. La connexion sans fil entre l'émetteur et le récepteur est essentielle pour garantir flexibilité, mobilité et contrôle à distance, sans nécessiter de connexion physique entre l'interface de contrôle et le système mécanique exécutif.

Le NRF24L01+ est un module de communication radiofréquence fonctionnant dans la bande ISM (industrielle, scientifique et médicale) de 2,4 GHz. Il utilise une interface SPI pour communiquer avec le microcontrôleur et prend en charge la configuration de l'adresse de communication (pipe), de la puissance de transmission, du débit de transfert, ainsi que la confirmation automatique des messages reçus (accusé de réception automatique). Cette fonctionnalité contribue à une plus grande fiabilité lors de la transmission de commandes critiques.

Dans la partie transmission, le module est configuré en mode transmission via les fonctions *radio.begin()*, *openWritingPipe()* et *setPALevel()*, tandis que dans la partie réception le mode écoute (*startListening()*) est activé, permettant la réception continue des commandes entrantes. La communication s'effectue en temps réel par l'envoi de tableaux de données structurés, dont chaque valeur correspond à une commande moteur spécifique.

L'avantage du protocole sans fil utilisé réside non seulement dans l'élimination des connexions par câble, mais aussi dans la prise en charge de la communication bidirectionnelle, bien que le système actuel utilise un canal unidirectionnel (de l'émetteur au récepteur). Le système prend en charge une vitesse de transmission allant jusqu'à 1 Mbit/s, ce qui est largement suffisant pour contrôler un bras robotisé en temps réel, tout en préservant l'efficacité énergétique du module.

En termes de robustesse et de sécurité, le NRF24L01+ assure la vérification des données matérielles grâce au contrôle de redondance cyclique (CRC) intégré et prend en charge la communication multicanal, ce qui permet un fonctionnement dans des environnements potentiellement perturbateurs électromagnétiques. Ce module est ainsi parfaitement adapté aux applications d'ingénierie réelles nécessitant une transmission de commandes fiable et rapide.

4. Partie réceptrice

4.1 Vue générale du système

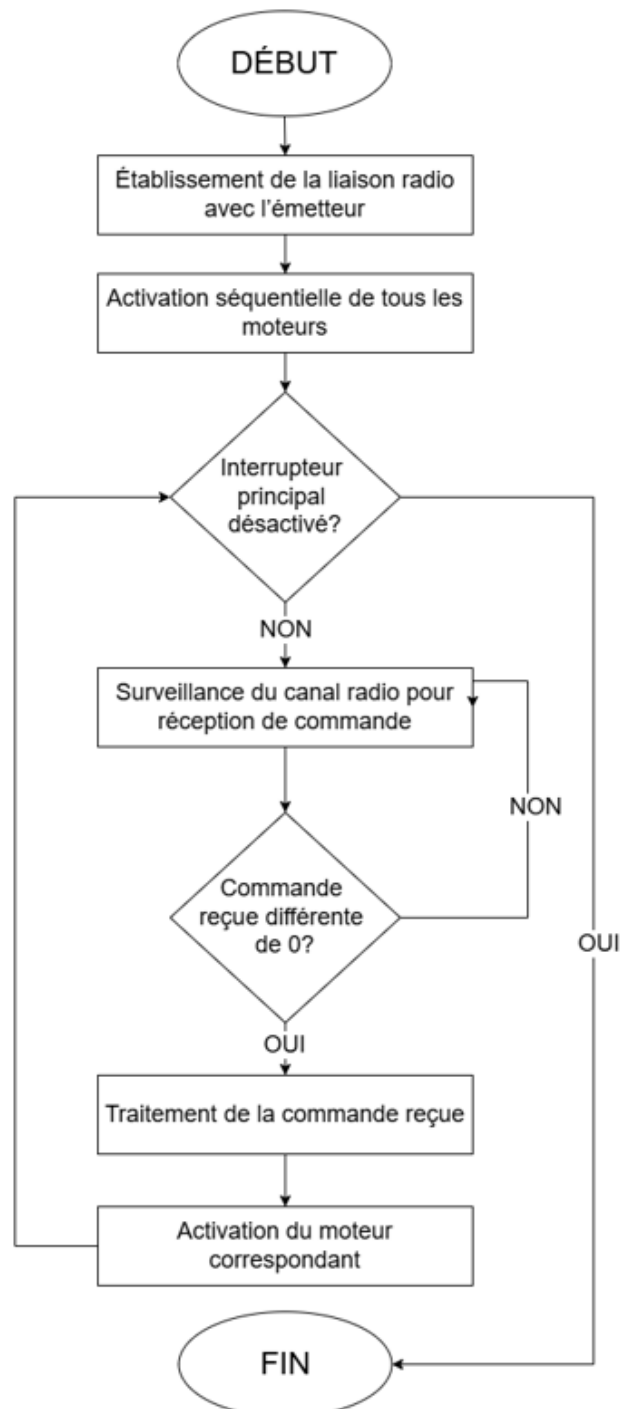


Figure 3.8 - Algorithme de fonctionnement de la partie réception

La partie réception du système représente l'étape finale du processus de communication et d'exécution du contrôle du bras robotisé. Elle est chargée d'interpréter les commandes reçues par radio et d'activer ensuite les mécanismes moteurs correspondants qui constituent la chaîne cinématique du manipulateur. La partie réception est implémentée à l'aide d'une plateforme microcontrôleur (Arduino MEGA), associée à un module radio NRF24L01+ et à trois circuits intégrés de type L293D pour le contrôle des moteurs électriques.

La fonction principale de ce bloc est de décoder les commandes de contrôle reçues de la partie transmission et de les associer aux mouvements prédéfinis des moteurs correspondants. Les commandes sont reçues sous forme de valeurs numériques (de 0 à 10), chacune correspondant à une opération spécifique du moteur : rotation de la base, élévation ou abaissement du bras, rotation du coude, mouvement du poignet ou commande du mécanisme de préhension. Après avoir identifié la valeur correspondante, le contrôleur Arduino active les sorties correspondantes, par lesquelles il envoie des signaux logiques aux moteurs électriques.

Chaque moteur est connecté à deux broches numériques, une pour chaque sens de rotation, et le sens de déplacement est déterminé par une combinaison appropriée de signaux (HAUT/BAS). Cela permet un contrôle bidirectionnel de chacun des cinq moteurs, permettant des mouvements complexes en environnement réel. Pour garantir la sécurité et la protection contre toute activation intempestive, à la fin de chaque cycle de traitement, toutes les sorties moteur sont désactivées si aucune nouvelle commande n'est reçue ou si une commande de valeur 0 est reçue, indiquant un arrêt complet.

Le récepteur fonctionne en mode d'écoute continue, activé par la fonction ***radio.startListening()***, qui permet une réponse rapide à toute commande entrante. La communication s'effectue via une interface SPI, et l'échange de données avec le module radio s'effectue grâce à la bibliothèque ***RF24***, qui assure une interaction fiable et rapide entre le microcontrôleur et le module sans fil.

Cette architecture offre modularité et évolutivité au système, permettant l'ajout facile de canaux moteur ou de capteurs supplémentaires sans modifier complètement la logique. Le récepteur fait office de module exécutif, mais peut également être adapté pour la communication de retour d'information vers l'émetteur lorsque des fonctions de surveillance, de diagnostic ou de contrôle sont nécessaires.

4.2 Réception des signaux

La partie réceptrice du système fonctionne en mode réception constante via le module sans fil NRF24L01+, configuré pour recevoir des données via un canal radio préétabli. Le module fonctionne sur la bande ISM de 2,4 GHz et utilise une interface SPI pour communiquer avec le microcontrôleur (Arduino MEGA). Après avoir établi la connexion radio, la commande ***radio.startListening()*** place le module en mode de réception passive, où il attend constamment les signaux entrants. Lorsqu'un nouveau message est reçu, la logique est activée, ce qui permet de lire le contenu du paquet dans un tableau de type ***ReceivedMessage[]***. Cela garantit une réception fiable et ponctuelle des signaux de contrôle envoyés par la partie émission.

4.3 Interprétation des commandes

Après réception des données, le microcontrôleur passe à l'étape suivante: l'interprétation de la commande, contenue dans le premier élément du tableau *ReceivedMessage[0]*. La commande est codée sous la forme d'un entier (de 0 à 10), chaque valeur étant pré-associée à un mouvement ou une action spécifique d'un élément du bras robotisé. Cette approche, basée sur une valeur discrète pour chaque commande, permet une identification rapide grâce à une série de vérifications conditionnelles (conditions conditionnelles). Par exemple, une valeur de 1 active le déplacement de la base vers la droite, et une valeur de 9 active la fermeture de la poignée. Cela traduit efficacement les signaux sans fil en actions mécaniques spécifiques.

4.4 Contrôle servomoteur du bras robotique

L'étape finale du fonctionnement de la partie réceptrice consiste à exécuter la commande en activant le moteur électrique correspondant, ce qui permet d'obtenir le mouvement souhaité du bras robotisé. Le contrôle des moteurs est assuré par trois circuits intégrés de type L293D. Chaque servomoteur est connecté à deux broches de sortie de la carte Arduino, activées selon une combinaison spécifique (niveau logique HAUT ou BAS) pour garantir le sens de rotation souhaité. Ainsi, chaque moteur peut effectuer un mouvement vers l'avant ou vers l'arrière, réalisant ainsi les fonctions clés du manipulateur : rotation de la base, élévation du bras, flexion du coude, positionnement du poignet et contrôle du mécanisme de préhension. Pour garantir la sécurité et prévenir toute activité indésirable, tous les moteurs s'arrêtent automatiquement en l'absence de nouvelle commande ou à la réception d'une commande de valeur 0, indiquant un état de repos.

4.5 Programmation d'un module pour exécuter des mouvements

- **Configuration de la communication sans fil avec NRF24L01+**

La figure 3.9 illustre la configuration du module sans fil NRF24L01+ pour un fonctionnement en mode réception. La commande *radio.begin()* lance la communication SPI avec le module. La commande *openReadingPipe()* ouvre un canal de communication avec une adresse fixe (qui doit correspondre à celle de l'émetteur) et la commande *startListening()* active le mode d'écoute des paquets entrants.

La radio est configurée pour utiliser le niveau de transmission maximal (RF24_PA_MAX), ainsi qu'un débit de transfert standard de 1 Mbit/s. L'acquittement et les tentatives automatiques garantissent la fiabilité de la réception des données. Tout cela garantit une connexion stable et sans problème entre les deux appareils.

```
radio.begin(); // Start the NRF24L01
radio.openReadingPipe(1, pipe); // Get NRF24L01 ready to receive
radio.setPALevel(RF24_PA_MAX);
radio.setDataRate(RF24_1MBPS);
radio.setAutoAck(1);           // Ensure autoACK is enabled
radio.setRetries(2, 15);
radio.startListening();        // Listen to see if information received
```

Figure 3.9 - Configuration de la communication sans fil avec le NRF24L01+

- **Initialisation des broches du moteur**

L'exemple de code de la figure 3.10 initialise les sorties connectées aux canaux du moteur L293D qui contrôlent les différentes parties du bras robotisé : rotation à la base, à l'épaule, au coude, au poignet et à la pince. Chaque moteur nécessite deux broches : une pour la rotation avant et une pour la rotation arrière, ce qui explique le besoin de 10 broches au total pour 5 moteurs. Le mode OUTPUT garantit que l'Arduino pilotera ces broches comme sorties numériques.

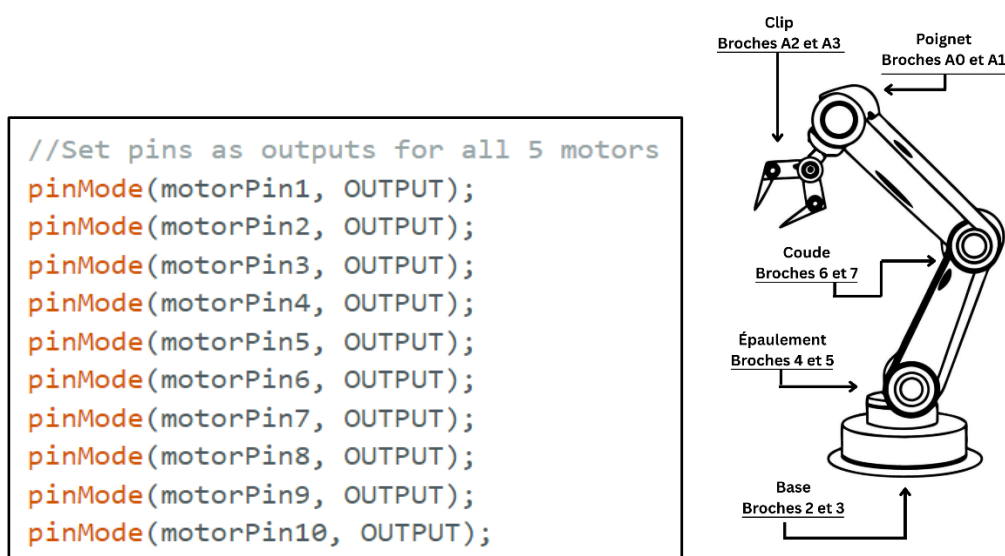


Figure 3.10 - Initialisation des broches des moteurs

Figure 3.11 - Schéma du bras robotisé avec correspondance des broches de commande

- **Lecture de messages et envoi de commandes aux moteurs**

Dans la partie du code illustrée à la figure 3.12, on vérifie si le module radio a reçu un nouveau message. Si c'est le cas, il est lu dans le tableau ***ReceivedMessage[]***. Le premier élément du tableau contient une valeur numérique comprise entre 0 et 10, correspondant à une commande spécifique envoyée par l'émetteur suite à un geste reconnu.

Grâce à des vérifications séquentielles (si (*ReceivedMessage[0]* == *X*)), le programme détermine quel moteur doit être activé et dans quelle direction.

```
while (radio.available())
{
    radio.read(&ReceivedMessage, sizeof(ReceivedMessage)); // Read information from the NRF24L01

    if (ReceivedMessage[0] == 1) //
    {
        digitalWrite(motorPin1, HIGH);
        digitalWrite(motorPin2, LOW);
        lcd.setCursor(0, 0);
        lcd.print("Center");
    }
    if (ReceivedMessage[0] == 2) //
    {
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, HIGH);
    }
}
```

Figure 3.12 - Lecture des messages et transmission des commandes aux moteurs

À la réception d'une valeur de 1, le canal moteur responsable de la rotation de la base du bras robotique dans le sens horaire est activé. Le mouvement inverse, à savoir la rotation de la base vers la gauche, est réalisé avec une valeur de 2. Ces deux commandes fournissent l'orientation horizontale de base du bras robotique par rapport à l'espace de travail.

Les commandes de valeurs 3 et 4 contrôlent le mouvement vertical du bras. Une valeur de 3 lève le bras, tandis qu'une valeur de 4 l'abaisse. Ce mouvement est essentiel pour positionner l'outil (ou la pince) à différentes hauteurs par rapport à la base.

Ainsi, les valeurs 5 et 6 contrôlent le mouvement du coude du mécanisme. La commande 5 déplace le coude vers l'intérieur ou vers la gauche, tandis que la valeur 6 active le mouvement dans la direction opposée, vers l'extérieur ou vers

la droite. Cela élargit la portée fonctionnelle du manipulateur, lui permettant d'atteindre des points plus éloignés.

Le segment suivant des commandes de contrôle - 7 et 8 - concerne le positionnement du mécanisme du poignet. La commande de valeur 7 entraîne une flexion du poignet vers le haut, et celle de valeur 8, une flexion vers le bas. Cela permet un réglage précis de l'orientation de l'effecteur terminal.

Les commandes 9 et 10 contrôlent le mécanisme de préhension (pince). À la valeur 9, la pince se ferme pour saisir un objet, tandis qu'à la valeur 10, elle amorce son ouverture, c'est-à-dire la libération de l'élément saisi. Ce contrôle bidirectionnel de la pince est essentiel pour les manipulations telles que le levage, le déplacement et le positionnement.

- **Commande d'arrêt complet de tous les moteurs**

Une commande de valeur 0 correspond à un « état neutre » ou à un arrêt complet. Toutes les broches sont positionnées sur BAS, ce qui désactive tous les moteurs simultanément. Ceci est essentiel pour la sécurité du système, car cela permet une réponse rapide lorsqu'un mouvement doit être arrêté. En pratique, cela peut servir à réinitialiser le robot, à préparer une nouvelle commande ou à effectuer un arrêt d'urgence.

```
if (ReceivedMessage[0] == 0) // FULL STOP
{
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    digitalWrite(motorPin5, LOW);
    digitalWrite(motorPin6, LOW);
    digitalWrite(motorPin7, LOW);
    digitalWrite(motorPin8, LOW);
    digitalWrite(motorPin9, LOW);
    digitalWrite(motorPin10, LOW);
}
```

Figure 3.13 - Commande d'arrêt complet de tous les moteurs

D'un point de vue technique, une telle commande est un composant obligatoire de tout système sûr, cette commande agit comme une stratégie de sécurité passive qui évite de maintenir les moteurs dans un état activé sans contrôle explicite de l'utilisateur.

- **Arrêt supplémentaire des moteurs en l'absence de signal radio**

À la fin du code du récepteur, la figure 3.14 montre un mécanisme de sécurité intégré qui garantit l'arrêt automatique de tous les moteurs en l'absence de nouvelle commande. Cela évite que le robot reste actif sans intervention de l'opérateur, ce qui pourrait entraîner des dommages ou un accident. Il s'agit d'un bon exemple de protection logicielle passive, courante dans les systèmes industriels.

```
digitalWrite(motorPin1, LOW);  
digitalWrite(motorPin2, LOW);  
digitalWrite(motorPin3, LOW);  
digitalWrite(motorPin4, LOW);  
digitalWrite(motorPin5, LOW);  
digitalWrite(motorPin6, LOW);  
digitalWrite(motorPin7, LOW);  
digitalWrite(motorPin8, LOW);  
digitalWrite(motorPin9, LOW);  
digitalWrite(motorPin10, LOW);
```

Figure 3.14 - Arrêt de secours des moteurs en cas de perte de signal radio

CHAPITRE 4: ANALYSE DES RÉSULTATS OBTENUS, APPLICABILITÉ ET CONCLUSIONS

1. Démonstration du fonctionnement du prototype

1.1. Implémentation matérielle et principaux composants du prototype

Le prototype se compose d'une partie émettrice (gant capteur avec module de contrôle) et d'une partie réceptrice (bras robotisé avec carte de contrôle et alimentation).

La figure 4.1 montre la partie transmission :

- Microcontrôleur Arduino Nano connecté au module radio NRF24L01+, capteur de mouvement MPU6050 et fils nécessaires à la connexion.
- Tout cela est intégré au gant et fixé solidement pour plus de confort lors du port et du travail.

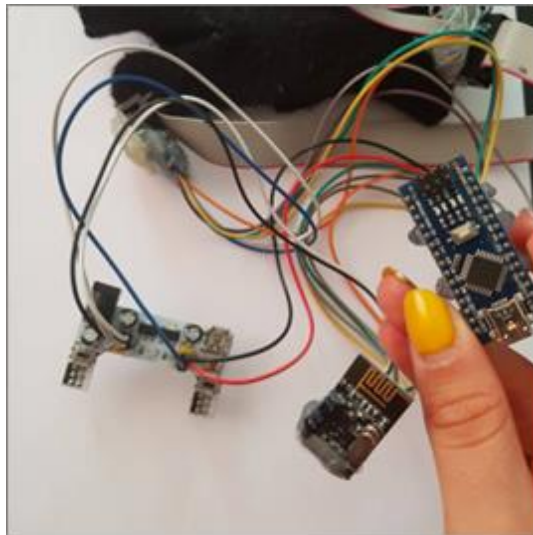


Figure 4.1 - Partie émettrice du système

La figure 4.2 présente le gant capteur :

- Il dispose d'un capteur MPU6050 installé pour détecter les mouvements de la main (lacet, tangage, roulis).
- Le capteur est fixé au gant avec du silicone chaud pour plus de stabilité, et les câbles sont connectés au module de commande.

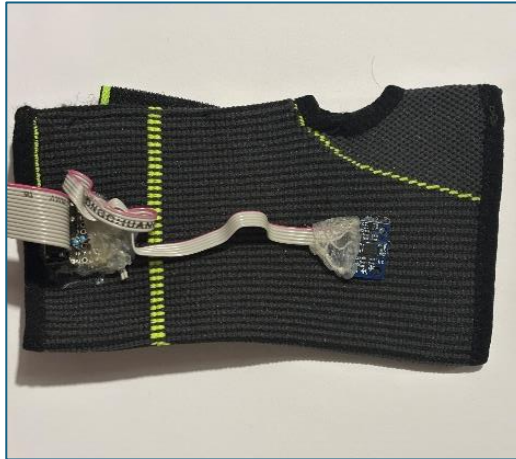


Figure 4.2 - Vue de la gant capteur avec le module MPU6050 fixé

La figure 4.3 montre le bras robotisé OWI-535, équipé de moteurs électriques et de câbles de connexion. Il s'agit de la partie mécanique du système, qui exécute les commandes reçues et réalise les mouvements dictés par les gestes de l'opérateur.



Figure 4.3 - Bras robotisé OWI-535

La figure 4.4 montre la partie réceptrice :

- Microcontrôleur Arduino MEGA connecté à un module radio NRF24L01+, ainsi qu'aux pilotes de contrôle moteur (L293D sur une carte prototype).
- L'alimentation est fournie par une batterie externe avec une tension appropriée pour les composants électroniques.

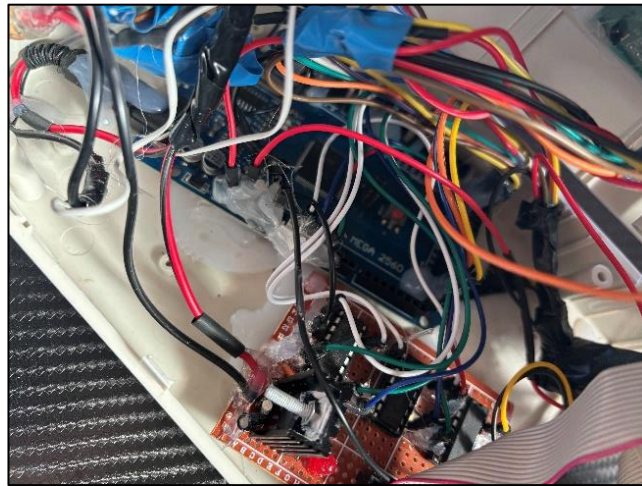


Figure 4.4 - Partie réceptrice du système

La construction globale du prototype présente une conception modulaire, permettant une maintenance et une mise à niveau faciles du système si nécessaire.

1.2. Traitement des données et résultats du moniteur série

Les données du capteur MPU6050 sont traitées en temps réel grâce à un algorithme spécialement développé, dans lequel le filtre de Kalman joue un rôle essentiel. Le MPU6050 fournit des données brutes d'accéléromètre et de gyroscope, elles-mêmes sujettes à du bruit dû aux vibrations, aux influences externes et aux imprécisions des mesures du capteur. Sans traitement approprié, les résultats seraient hésitants et instables, ce qui rendrait difficile la reconnaissance correcte des gestes.

Le filtre fournit trois paramètres principaux qui décrivent la position de la main dans l'espace :

- **Yaw** - rotation autour de l'axe vertical;
- **Pitch** - incliner d'avant en arrière ;
- **Roll** - incliner sur le côté.

À chaque changement de geste, les données sont visualisées en temps réel sur le moniteur série de l'Arduino, permettant une vérification facile du fonctionnement du système et de l'exactitude de la reconnaissance.

- **Position centrale (Center)**

Sur la figure 4.5, les valeurs des trois axes sont proches de zéro, avec des écarts minimes. Cela témoigne de la stabilité du système en l'absence de mouvement et de la grande efficacité du filtre lorsqu'il est maintenu en position centrale.

```
Kalman Filtered angles: -5.24  -1.54  -2.23
Center
Kalman Filtered angles: -5.25  -1.54  -2.23
Center
Kalman Filtered angles: -5.25  -1.54  -2.23
Center
Kalman Filtered angles: -5.26  -1.55  -2.23
Center
Kalman Filtered angles: -5.26  -1.55  -2.23
Center
```

Figure 4.5 - Position neutre

- **Pente ascendante (Up)**

Dans la figure 4.6, les valeurs de hauteur supérieures à 50° sont caractéristiques d'un mouvement « vers le haut ». Les données sont fluides et sans sauts brusques, ce qui résulte des bonnes performances du filtre.

```
Kalman Filtered angles: -31.89  54.99  7.95
Up
Kalman Filtered angles: -31.81  54.84  7.92
Up
Kalman Filtered angles: -31.92  54.70  7.82
Up
Kalman Filtered angles: -32.05  54.46  7.69
Up
Kalman Filtered angles: -31.93  53.75  7.45
Up
```

Figure 4.6 - Inclinaison vers le haut

- Tournez la base vers la droite (Base Right)

Sur la figure 4.7, les valeurs de lacet sont supérieures à 50° avec de faibles variations de tangage et de roulis. Ceci est typique d'un geste « Base Right », où la main est tournée vers la droite sur le plan horizontal.

```
Kalman Filtered angles: 52.66  -2.18  0.50
Base Right
Kalman Filtered angles: 52.81  -2.26  0.43
Base Right
Kalman Filtered angles: 52.92  -2.32  0.39
Base Right
Kalman Filtered angles: 53.00  -2.35  0.36
Base Right
```

Figure 4.7 - Rotation de la base vers la droite

Ces résultats de surveillance série démontrent les performances fiables du système sur une variété de gestes et confirment l'importance du traitement des données du filtre de Kalman pour atteindre la précision et la stabilité.

2. Analyse comparative des résultats à l'aide du filtre de Kalman et des angles d'Euler

Une analyse comparative a été réalisée entre deux méthodologies principales de traitement et d'interprétation des données des capteurs : l'utilisation d'un filtre de Kalman et l'utilisation directe des angles d'Euler.

Les angles d'Euler ont été utilisés lors d'une étape antérieure du développement, lors de la mise en œuvre d'un premier prototype du système. Leur principal avantage réside dans leur relative simplicité de calcul et leur interprétation directe de l'orientation de la main selon les trois axes principaux (lacet, tangage, roulis). Ils sont donc adaptés aux démonstrations et aux

expériences rapides. Malgré ces avantages, les résultats ont montré une sensibilité importante au bruit et aux fluctuations des mesures, notamment lors des mouvements dynamiques.

- Dans la figure 4.8, nous pouvons remarquer des fluctuations dans les valeurs malgré la main stable.

```
euler    2.67    -1.77    2.17
Center
euler    0.71    1.45    -3.13
Center
euler    1.96    -3.11    3.17
Center
euler    1.85    -0.28    -0.19
Center
```

Figure 4.8 - Données d'angle d'Euler en position neutre (« centre »)

- Dans la figure 4.9, nous pouvons remarquer que les données montrent des changements brusques et du bruit malgré la main levée stable.

```
euler    -8.53    55.74    -14.19
Up
euler    -13.37    60.83    -17.79
Up
euler    -16.21    64.13    -20.47
Up
euler    -19.36    66.91    -22.66
Up
```

Figure 4.9 - Données d'angle d'Euler pour le mouvement „Up“.

- En tournant la main vers la droite, des oscillations instables sont observées, comme on peut le voir sur la figure 4.10

```
euler    81.57    -1.17    -0.84
Base Right
euler    85.75    -1.24    -0.24
Base Right
euler    83.27    -1.09    -0.50
Base Right
euler    87.71    -1.10    -0.35
Base Right
```

Figure 4.10 - Données d'angle d'Euler pour le mouvement “Base Right”

En revanche, l’application d’un filtre de Kalman entraîne un lissage significatif des données et une réduction du bruit.

Le filtre de Kalman combine prédiction et correction d’état basées sur de nouvelles mesures, ce qui garantit une précision et une fiabilité élevées en temps réel. Cela minimise le risque d'activation involontaire des commandes, et le système global fait preuve d'une plus grande résilience, même en cas de mouvements rapides et brusques de la main.

Par conséquent, on peut conclure que si les angles d'Euler constituent une base utile pour le développement d'un prototype initial, l'utilisation d'un filtre de Kalman offre le niveau de précision et de stabilité nécessaire, essentiel aux applications pratiques du système. L'intégration d'un filtre de Kalman dans l'architecture de conception contribue significativement à la fiabilité du contrôle et au bon fonctionnement du bras robotisé en conditions réelles.

3. Analyse des performances

- **Précision de la reconnaissance**

Le système de reconnaissance gestuelle atteint un haut degré de précision en conditions de fonctionnement normales, lorsque les mouvements de la main sont fluides et que sa position est stable par rapport au système de coordonnées terrestres. Le capteur inertiel MPU6050 utilisé fournit des données sur la vitesse angulaire et l'accélération linéaire, interprétées par une logique algorithmique intégrée basée sur les valeurs de lacet, de tangage et de roulis. Pour améliorer la précision des mesures et éliminer les vibrations et le bruit indésirables, le système intègre un filtre de Kalman, qui fournit une estimation optimale de l'orientation en combinant les données du gyroscope et de l'accéléromètre.

Lors des tests du système, il a été constaté qu'avec des gestes clairement définis et un impact externe limité (par exemple, mouvements brusques ou chocs), les commandes correctement reconnues sont plus nombreuses que les commandes non reconnues. De légers écarts de précision sont observés en cas d'instabilité de la main, de changement rapide de direction du mouvement ou de chevauchement partiel des conditions entre deux gestes. Néanmoins, la résistance du système aux écarts à court terme par rapport aux valeurs attendues est bonne, ce qui le rend adapté aux applications concrètes où une précision absolue n'est pas requise.

- **Retard et perte de données**

Le délai moyen entre l'exécution d'un geste et l'activation de la fonction correspondante dans le bras robotisé est d'environ 500 millisecondes. Ce délai comprend à la fois le temps de collecte et de traitement des données des capteurs, ainsi que le temps de transmission sans fil et d'exécution de la commande. Dans le contexte des systèmes de contrôle à distance en temps réel, cet intervalle est acceptable et n'entraîne pas de retard notable pour l'opérateur.

Concernant la fiabilité de la transmission, le module radio NRF24L01+ utilisé prend en charge l'accusé de réception automatique des paquets reçus, ainsi que la retransmission en l'absence de réponse du récepteur. Ces fonctions minimisent le risque de perte de données sur le canal de communication. Lors des tests expérimentaux, aucun cas significatif de commande non exécutée ni de perte de paquet n'a été enregistré, ce qui témoigne du fonctionnement stable et fiable du système en conditions normales. En cas d'instabilité temporaire de la communication, comme une sortie hors de portée ou des interférences électromagnétiques, la réémission des paquets assure le rétablissement de la communication sans intervention externe.

- **Stabilité de la connexion**

Lors de tests réalisés dans un environnement contrôlé avec visibilité directe entre les modules, la liaison radio a démontré une communication stable et ininterrompue dans un rayon d'environ 10 mètres. En présence d'obstacles physiques, notamment de structures métalliques, de murs en béton ou de sources électromagnétiques, une baisse de la qualité du signal est observée, ce qui peut entraîner des interruptions temporaires ou des retards dans la transmission des commandes.

Cependant, grâce à la configuration de répétition automatique et au niveau élevé de sensibilité du récepteur, le système compense avec succès les pertes temporaires, ce qui offre une fiabilité suffisante pour la plupart des applications pratiques où un contrôle dans un espace confiné ou un environnement de laboratoire est attendu.

4. Avantages et limites du système

4.1 Avantages

- **Communication sans fil:**

L'un des principaux avantages du système développé est l'utilisation de la communication sans fil via les modules radio NRF24L01+. Cette solution élimine le besoin de connexion physique entre le dispositif de commande et le bras mécanique, ce qui accroît considérablement la flexibilité de l'application et permet le contrôle à distance. Cette approche est particulièrement adaptée aux situations où l'opérateur doit se trouver hors de portée du système robotisé, par exemple lors de travaux avec des matières dangereuses ou dans des zones difficiles d'accès.

- **Fonctionnement en temps réel:**

Le système maintient un délai de réponse négligeable entre la reconnaissance gestuelle et l'exécution du mouvement correspondant par le bras robotisé. Ce délai moyen d'environ 500 ms se situe dans les limites acceptables pour la plupart des applications interactives. Cela assure une fluidité et une prévisibilité de contrôle suffisantes, essentielles pour les manipulations manuelles.

- **Conception modulaire:**

La conception du système repose sur un principe modulaire : les composants matériels et logiciels sont clairement séparés et fonctionnellement indépendants. Cela facilite la maintenance, la mise à niveau et l'extension ultérieure du système. Par exemple, le remplacement d'un module de communication ou la mise à jour d'un algorithme logiciel

peuvent être effectués sans réorganisation majeure de l'architecture globale.

4.2 Limitations

- **Sensibilité aux mouvements dynamiques et incontrôlés:**

Le contrôle gestuel nécessite des mouvements fluides et distincts pour être interprété correctement. Des mouvements plus brusques ou imprévisibles, notamment s'ils sont accompagnés de vibrations ou de bruits externes, augmentent le risque de mauvaise reconnaissance ou activation d'une commande. Cela peut limiter l'utilisation du système dans des environnements dynamiques ou bruyants.

- **Portée de communication radio limitée:**

Bien que fiable à courte distance, la connexion radio mise en œuvre via NRF24L01+ est sensible aux obstacles physiques tels que les murs, les surfaces métalliques ou les interférences électromagnétiques.

4.3 Opportunités d'amélioration

Le système est implémenté avec des solutions matérielles et logicielles soigneusement sélectionnées, garantissant une fiabilité et une précision de contrôle élevées. Le filtre de Kalman utilisé pour traiter les données des capteurs s'est avéré le plus adapté à ce type d'application, car il offre une combinaison optimale de précision, de résistance au bruit et de vitesse de traitement. Il assure une reconnaissance gestuelle stable, même avec des mouvements de main dynamiques et dans des environnements soumis à des interférences externes.

Les principales perspectives d'amélioration future du système résident dans l'utilisation de composants matériels plus coûteux et de meilleure qualité. Le remplacement des capteurs, modules radio, microcontrôleurs et éléments mécaniques par des versions industrielles plus fiables permettrait d'améliorer encore la précision et la durabilité du système, notamment dans des conditions difficiles ou en exploitation prolongée. Cette approche réduirait le risque de pannes matérielles et permettrait d'appliquer le système dans des environnements industriels plus exigeants.

5. Conclusion

Dans le cadre de cette thèse, un prototype de bras robotique industriel contrôlé par gestes a été développé, intégrant pleinement les composants matériels et logiciels de la plateforme Arduino. Le système comprend une partie émettrice (gant capteur), qui capte les mouvements de la main humaine, les traite à l'aide d'un capteur inertiel (MPU6050) et d'un filtre de Kalman, et transmet sans fil les commandes reçues à la partie réceptrice, qui contrôle la structure mécanique du bras robotique. Un module radio NRF24L01+ fiable a été utilisé pour la transmission des données, assurant une communication stable avec une faible latence et une grande précision des signaux transmis.

Le projet a franchi plusieurs étapes clés : analyse théorique, sélection et raisonnement des composants, implémentation matérielle, programmation et tests. L'analyse documentaire du chapitre 1 a fourni une base de comparaison avec les solutions déjà mises en œuvre, en mettant l'accent sur les degrés de liberté, les méthodes de contrôle, les interfaces de communication utilisées et les filtres appliqués au traitement des données des capteurs. L'analyse comparative des projets existants a montré que le choix d'une combinaison appropriée de microcontrôleur, de capteur et d'algorithme de traitement est essentiel pour garantir un fonctionnement fiable et précis du système.

Le chapitre 2 a présenté la solution théorique du problème, en examinant les concepts fondamentaux liés à la robotique industrielle, à la plateforme Arduino, aux technologies de communication, aux capteurs inertiels et aux algorithmes de filtrage des données. Une attention particulière a été portée à l'applicabilité du filtre de Kalman pour le lissage du bruit dans le signal du capteur, car il a été démontré qu'il offre une grande précision dans les mouvements dynamiques et qu'il est adapté au contrôle en temps réel.

Au chapitre 3, la solution d'ingénierie a été présentée, incluant le développement des parties émettrice et réceptrice. La programmation du code correspondant à chaque partie a permis de reconnaître les gestes de base – rotation, inclinaison, mouvements combinés – et de les transformer en commandes pour les servomoteurs composant le bras robotique. Le système fonctionne en temps réel, chaque mouvement du bras entraînant une action correspondante du robot, démontrant ainsi la précision, la réactivité et l'applicabilité pratique du développement.

Les résultats obtenus démontrent qu'en utilisant des composants matériels accessibles et peu coûteux, il est possible de construire un système fiable de contrôle à distance d'un robot industriel. Outre son potentiel de formation et de démonstration, la solution peut également servir de base à des développements avancés offrant un degré d'autonomie accru, un retour d'information, une visualisation des données ou même une intégration avec l'apprentissage automatique pour la reconnaissance automatique des gestes. À l'avenir, il sera possible d'ajouter une interface de surveillance visuelle des commandes exécutées, ainsi que des degrés de liberté supplémentaires, ce qui étendra les fonctionnalités du système.

En conclusion, cette thèse démontre une approche réussie pour la construction d'un système flexible et efficace de contrôle gestuel d'un bras robotique.

L'harmonie entre matériel et logiciel, entre théorie et pratique, a été atteinte, transformant ce développement en une véritable démonstration du potentiel de l'électronique moderne, des technologies de capteurs et de la communication sans fil pour les futures applications industrielles et éducatives.

BIBLIOGRAPHIE:

- [1] Mohammed Ali, H., Hashim, Y., & A AL-Sakkal, G. (2022). Design and implementation of Arduino based robotic arm. *International Journal of Electrical and Computer Engineering*, 12(2), 1411-1411.
- [2] Ahmed, A. S., Marzog, H. A., & Abdul-Rahaim, L. A. (2021). Design and implement of robotic arm and control of moving via IoT with Arduino ESP32. *International Journal of Electrical & Computer Engineering* (2088-8708), 11(5).
- [3] Basith, I. I., & Marroquin, B. (2020, June). Byoe: Design and development of a simple robotic arm. In *2020 ASEE Virtual Annual Conference Content Access*.
- [4] Wen, C. K., Yong, L. C., & Nataraj, C. (2022). Gesture controlled robotic arm for hazardous chemical control. *Journal of Applied Technology and Innovation*, 6(3), 47-54.
- [5] Reddy, K. M., Maddilety, S., Reshma, S., Babu, P. S., & Srikanth, V. (2023). Design and Implementation of Robotic Arm for Pick and Place by using Bluetooth Technology.
- [6] Prasanth, N., Shrivastava, K., Sharma, A., Basu, A., Sinha, R. A., & Raja, S. P. (2023). Gesture-based mouse control system based on MPU6050 and Kalman filter technique. *International Journal of Intelligent Systems Technologies and Applications*, 21(1), 56-71.
- [7] Sultan, J. M., Zani, N. H., Azuani, M., Ibrahim, S. Z., & Yusop, A. M. (2022). Analysis of inertial measurement accuracy using complementary filter for MPU6050 sensor. *Jurnal Kejuruteraan*, 34(5), 959-964.
- [8] Setiawan, R., Triharminto, H. H., & Fahrurozi, M. (2021, November). Gesture control menggunakan IMU MPU 6050 metode kalman filter sebagai kendali quadcopter. In *Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia p-ISSN* (Vol. 2086, p. 5805).

- [9] Rahmad, S. A., Ibrahim, Z., & Md Yusof, Z. (2022). Simulated Kalman filter with modified measurement, substitution mutation and hamming distance calculation for solving traveling salesman problem. In *Enabling Industry 4.0 through Advances in Mechatronics: Selected Articles from iM3F 2021, Malaysia* (pp. 309-320). Singapore: Springer Nature Singapore.
- [10] Kakoty, N.M., Goswami, R., Vinjamuri, R. (2024). Microcontrollers in Robotics. In: Introduction to Embedded Systems and Robotics. Springer, Cham. https://doi.org/10.1007/978-3-031-73098-6_4
- [11] <https://www.universal-robots.com/in/blog/types-of-robotic-arms/>
- [12] https://www.researchgate.net/figure/Model-of-Articulated-robotic-arm_fig5_276062380
- [13] <https://electricalworkbook.com/cartesian-robot/>
- [14] https://www.researchgate.net/figure/The-Configuration-of-a-Cylindrical-Arm-Robot_fig2_336702692
- [15] <https://acrome.net/post/what-are-delta-robots-used-for>
- [16] <https://en.wikipedia.org/wiki/SCARA>
- [17] <https://allonrobots.com/spherical-robots/>
- [18] <https://blog.quadmeup.com/2017/05/24/fs1000a-and-xy-mk-5v-433mhz-rf-modules-overview/>
- [19] <https://www.amazon.co.uk/433MHz-Transmitter-Receiver-Module-FS1000A/dp/B07T9F6LT2>
- [20] <https://datacapturecontrol.com/articles/data-communication/wireless/rf/hc12/hc12-module>
- [21] <https://www.smart-prototyping.com/HC-12-Wireless-Transceiver-Module-SI4438-433MHz-1km>
- [22] <https://learn.sparkfun.com/tutorials/rfm69hcx-hookup-guide/all>

- [23] <https://www.graylogix.in/product/rfm69-433mhz-transceiver-module>
- [24] <https://www.semtech.com/products/wireless-rf/lora-connect/sx1278>
- [25] <https://in.element14.com/hoperf/rfm95w-868s2/transceiver-868mhz-fsk-ook/dp/2759294>
- [26] <https://www.digi.com/blog/post/xbee-buying-guide>
- [27] <https://www.digi.com/resources/examples-guides/basic-xbee-802-15-4-%28series-1%29-chat>
- [28] <https://datacapturecontrol.com/articles/data-communication/wireless/rf/nrf24l01/nrf24l01p-module>
- [29] <https://www.faranux.com/product/arduino-nrf24l01-2-4ghz-wireless-rf-transceiver-module-com42/>
- [30] <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>
- [31] <https://www.adafruit.com/product/2472>
- [32] <https://www.amazon.com/Adafruit-Absolute-Orientation-Fusion-Breakout/dp/B017PEIGIG>
- [33] <https://www.st.com/en/mems-and-sensors/lsm6dsox.html>
- [34] <https://www.elfadistelec.no/en/lsm6dsox-accelerometer-and-gyroscope-module-adafruit-4438/p/30172907>
- [35] <https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/>
- [36] <https://www.emag.bg/mpu9250-9-osov-modul-cl451/pd/DCZSXVBBM/>
- [37] https://www.ardboard.com/index.php?route=product/product&product_id=343
- [38] <https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>
- [39] <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>

[40] <https://www.tme.eu/bg/news/library-articles/page/61474/pasivni-filtri-kakvo-si-struva-da-znaete-za-tyah-v-nachaloto/>

[41] <https://www.kalmanfilter.net/>

APPLICATIONS

Figure 2.1 - Model of Articulated robotic arm

Figure 2.2 - Model of Cartesian robot arm

Figure 2.3 - Model of Cylindrical robot

Figure 2.4 - Model of Delta robot arm

Figure 2.5 - Model of SCARA robot arm

Figure 2.6 - Model of Polar robot

Figure 2.7 - Image de FS1000A / XY-MK-5V

Figure 2.8 - Image de HC-12

Figure 2.9 - Image de RFM69

Figure 2.10 - Image de RFM95 / SX1278

Figure 2.11 - Image de XBee Series 1 / 2

Figure 2.12 - Image de NRF24L01+

Figure 2.13 - Image de BNO055

Figure 2.14 - Image de LSM6DSOX

Figure 2.15 - Image de MPU-9250

Figure 2.16 - Image de MPU-6050

Figure 3.1 - Vue générale du système

Figure 3.2 - Partie émettrice - bloc schéma

Figure 3.3 - Initialisation et configuration du MPU6050 et du DMP

Figure 3.4 - Configuration de l'émetteur sans fil NRF24L01+

Figure 3.5 - Extraction et filtrage de Yaw, Pitch et Roll via un filtre de Kalman

Figure 3.6 - Reconnaissance de gestes et transmission de commandes

Figure 3.7 - Commande de recentrage (remise à zéro de la position)

Figure 3.8 - Partie réceptrice - bloc schéma

Figure 3.9 - Configuration de la communication sans fil avec le NRF24L01+

Figure 3.10 - Initialisation des broches des moteurs

Figure 3.11 - Schéma du bras robotisé avec correspondance des broches de commande

Figure 3.12 - Lecture des messages et transmission des commandes aux moteurs

Figure 3.13 - Commande d'arrêt complet de tous les moteurs

Figure 3.14 - Arrêt de secours des moteurs en cas de perte de signal radio

Figure 4.1 - Partie émettrice du système

Figure 4.2 - Vue de la gant capteur avec le module MPU6050 fixé

Figure 4.3 - Bras robotisé OWI-535

Figure 4.4 - Partie réceptrice du système

Figure 4.5 - Position neutre

Figure 4.6 - Inclinaison vers le haut

Figure 4.7 - Rotation de la base vers la droite

Figure 4.8 - Données d'angle d'Euler en position neutre (« centre »)

Figure 4.9 - Données d'angle d'Euler pour le mouvement „Up“.

Figure 4.10 - Données d'angle d'Euler pour le mouvement “Base Right”