



Université Abdelmalek
Essaâdi
Faculté des Sciences et
Techniques
– Al Hoceima –



Département de Physique

Filière : Système embarqué et Robotique

Module : Système embarqué : Applications et mise en œuvre

Application de l'algorithme d'Évolution
Différentielle (DE)

Optimisation d'un contrôleur PID pour un bras
manipulateur 3 DDL avec modélisation
géométrique inverse

Encadrante :
FAIZA DIB

Réalisé par :
ANASS MOUKHTARI

20 janvier 2026

Résumé

Ce rapport présente une étude sur l'application de l'algorithme d'Évolution Différentielle (DE) pour l'optimisation des paramètres d'un contrôleur PID commandant un bras manipulateur à 3 degrés de liberté. L'objectif est d'améliorer les performances de suivi de trajectoire tout en maintenant la robustesse du système.

Dans un premier temps, nous présentons les bases théoriques des bras manipulateurs robotiques, incluant leur modélisation géométrique directe et inverse. Ensuite, nous détaillons le fonctionnement du contrôleur PID et ses limitations. La partie centrale est consacrée à l'implémentation de DE pour l'optimisation des gains PID.

Les résultats montrent une amélioration significative des performances par rapport aux méthodes de réglage classiques.

Mots-clés : Bras manipulateur, 3 DDL, PID, Évolution différentielle, Cinématique inverse, Commande optimale

Table des matières

Résumé	1
Introduction Générale	8
Chapitre 1 Généralités et Problématique	9
1.1 Introduction	10
1.2 Présentation du bras manipulateur 3RRR	10
1.2.1 Définition et caractéristiques générales	10
1.2.2 Architecture mécanique	11
1.2.3 Configuration cinématique (DDL)	11
1.3 Notions de modélisation	12
1.3.1 Modèle géométrique direct (MGD)	12
1.3.2 Modèle géométrique inverse (MGI)	12
1.3.3 Jacobien cinématique (relation vitesses)	12
1.3.4 Modèle dynamique	13
1.4 Commande PID : principe et intérêt	13
1.4.1 Principe général	13
1.4.2 Intérêt et difficultés de réglage	14
1.5 Algorithme d'Évolution Différentielle (DE) : vue d'ensemble	14
1.5.1 Principe	14
1.5.2 Règles générales de DE	15
1.5.3 Pourquoi DE pour l'optimisation PID ?	15
1.6 Problématique et objectifs du travail	15
1.6.1 Problématique	15
1.6.2 Objectifs	15
1.7 Conclusion	16
Chapitre 2 Méthodologie et Implémentation	17
2.1 Modèle Géométrique Direct	18

2.1.1	Introduction	18
2.1.2	Initialisation du Tableau DH	18
2.2	Calcul des Matrices de Transformation M_{01} , M_{12} et M_{23}	19
2.2.1	Matrice M_{01} (Repère 0 vers Repère 1)	19
2.2.2	Matrice M_{12} (Repère 1 vers Repère 2)	20
2.2.3	Matrice M_{23} (Repère 2 vers Repère 3)	20
2.3	Calcul des Matrices M_{02} et M_{03}	20
2.3.1	Matrice M_{02} (Repère 0 vers Repère 2)	20
2.3.2	Matrice M_{03} (Repère 0 vers Repère 3)	20
2.4	Modèle Géométrique Inverse : Méthode de Paul	21
2.4.1	Situation du Repère R_n (organe terminal)	21
2.4.2	Première Étape : Pré-multiplication par M_{01}^{-1}	22
2.4.3	Ensemble des Solutions pour θ_1	22
2.4.4	Deuxième Étape : Pré-multiplication par $M_{01}^{-1} \cdot M_{12}^{-1}$	23
2.5	Deuxième Étape : Pré-multiplication par $M_{01}^{-1} \cdot M_{12}^{-1}$	23
2.5.1	Ensemble des Solutions pour θ_3	23
2.5.2	Ensemble des Solutions pour θ_2	24
2.6	Modèle Géométrique Inverse : Méthode Géométrique	24
2.6.1	Définition des Variables Intermédiaires	25
2.6.2	Solution pour θ_1	25
2.6.3	Solution pour θ_2	25
2.6.4	Solution pour θ_3	26
2.7	Trajectoire Calculée	26
2.7.1	Définition de la Trajectoire Articulaire	26
2.7.2	Paramètre de Progression $s(t)$	26
2.7.3	Évolution des Angles Articulaires	26
2.8	Modèle Dynamique	27
2.8.1	Rappel des Matrices de Transformation	27
2.8.2	Énergie Cinétique : Partie Translation	27
2.8.3	Position des Centres de Gravité	28
2.8.4	Calcul des Jacobiens pour les Vitesses Linéaires	28
2.8.5	Matrice d'Inertie en Translation	29
2.9	Énergie Cinétique : Partie Rotation	29
2.9.1	Jacobiens des Vitesses Angulaires	29
2.9.2	Matrices de Rotation	30
2.9.3	Tenseurs d'Inertie	30
2.9.4	Matrice d'Inertie en Rotation	31

2.10	Énergie Cinétique Totale	31
2.11	Énergie Potentielle	31
2.12	Lagrangien du Système	32
2.13	conclusion	33
Chapitre 3	Application et Simulation	34
3.1	Introduction	35
3.2	Présentation des outils de simulation	35
3.3	Commande du bras : cas de modélisation dynamique	36
3.4	Description des blocs MATLAB/Simulink	36
3.4.1	Bloc 1 : Génération de trajectoire	36
3.4.2	Bloc 2 : Modèle géométrique inverse (MGI)	36
3.4.3	Bloc 3 : Commande PID articulaire	37
3.4.4	Bloc 4 : Modèle dynamique du robot 3RRR	38
3.4.5	Bloc 5 : Calcul des métriques de performance (fonction coût) . .	39
3.5	Commande du bras : cas de géométrie inverse (MGI)	40
3.5.1	Principe global	40
3.5.2	Bloc génération de trajectoire cartésienne	40
3.5.3	Bloc MGI : conversion cartésien → articulaire	40
3.5.4	Boucles PID et robot	41
3.6	Métriques de performance et fonction coût	41
3.7	Intégration de l'algorithme DE dans la simulation	42
3.7.1	Principe général	43
3.7.2	Fonction <code>DE_Tuner_Live</code> : réglage PID en ligne	43
3.7.3	Interprétation : comparaison des deux cas	44
3.8	Conclusion	45
Conclusion Générale		46
Références		47
Chapitre A	Annexes	48
A.1	Code MATLAB/Simulink	49
A.2	Données complémentaires	49
A.3	Détails mathématiques	49

Table des figures

1.1	Bras manipulateur étudié (configuration 3RRR)	10
1.2	Architecture mécanique et repérage des articulations (q_1, q_2, q_3) du bras 3RRR.	11
1.3	Architecture de la commande PID du bras manipulateur 3RRR (boucle de rétroaction articulaire)	13
1.4	Architecture de l'optimisation des gains PID par Évolution Différentielle (DE) pour le bras 3RRR	14
2.1	Architecture de l'optimisation des gains PID par Évolution Différentielle (DE) pour le bras 3RRR	18
2.2	Modèle Géométrique Inverse	24
3.1	Environnement MATLAB/Simulink utilisé pour la modélisation et la simulation.	35
3.2	Chaîne de simulation MATLAB/Simulink	36
3.3	Bloc MGI (méthode graphique) : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés ($\theta_{1d}, \theta_{2d}, \theta_{3d}$).	37
3.4	Boucles PID articulaires	38
3.5	Chaîne incluant le modèle dynamique : appliqués au robot.	38
3.6	Sous-système robot (Simscape Multibody) : trois articulations Revolute1..3 et mesures des positions articulaires (position1..3).	39
3.7	Bloc MGI (méthode graphique) : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés ($\theta_{1d}, \theta_{2d}, \theta_{3d}$).	40
3.8	Bloc MGI : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés ($\theta_{1d}, \theta_{2d}, \theta_{3d}$).	41
3.9	Chaîne de commande basée sur la géométrie inverse MGI + multi-PID + robot.	41
3.10	Chaîne de commande basée DE	42
3.11	Architecture Simulink incluant l'algorithme DE : adaptation en ligne des gains PID pour chaque articulation.	43

3.12 Réponse du système avant intégration du DE (PID à gains fixes) :transitoire plus important et stabilisation plus lente.	44
3.13 Réponse du système après intégration du DE : amélioration de la stabilité et réduction de l'erreur de suivi.	45

Liste des tableaux

Introduction Générale

L'optimisation des performances de commande en robotique constitue un enjeu majeur, en particulier pour les bras manipulateurs où la précision, la rapidité de réponse et la stabilité conditionnent directement la qualité des mouvements. Dans ce projet, nous nous intéressons à un bras manipulateur à trois degrés de liberté (3 DDL), piloté par un contrôleur PID, et dont le mouvement est défini à partir d'une modélisation géométrique inverse.

L'objectif principal est d'appliquer l'algorithme d'Évolution Différentielle (Differential Evolution, DE) afin d'optimiser automatiquement les gains du contrôleur PID. Contrairement à un réglage manuel souvent long et dépendant de l'expérience, l'approche par DE permet d'explorer efficacement l'espace de recherche des paramètres et de trouver une combinaison de gains améliorant les performances du système. L'optimisation est conduite selon un critère de qualité basé sur l'erreur de suivi de trajectoire et, selon le besoin, sur des indicateurs tels que le dépassement, le temps de réponse et l'effort de commande.

La démarche adoptée repose sur trois éléments clés : (i) l'établissement d'un modèle exploitable du bras 3 DDL, (ii) le calcul de la loi de commande et de la trajectoire via la modélisation géométrique inverse, et (iii) la mise en œuvre de l'algorithme DE pour ajuster les gains PID afin d'assurer un suivi précis et stable. Les résultats obtenus permettent d'évaluer l'apport de l'optimisation par métaheuristique et de comparer les performances du contrôleur optimisé à celles d'un réglage conventionnel.

Le rapport est organisé comme suit : le premier chapitre présente le contexte, la problématique et les notions fondamentales. Le deuxième chapitre détaille la méthodologie et l'implémentation, notamment les calculs de cinématique inverse, la génération de trajectoires et la formulation du critère d'optimisation. Les chapitres suivants sont consacrés aux simulations, à l'analyse des résultats et à la discussion, avant de conclure sur les apports du projet et les perspectives d'amélioration.

Chapitre 1

Généralités et Problématique

1.1 Introduction

La robotique industrielle moderne exige des systèmes de commande à la fois précis, rapides et robustes. Les bras manipulateurs articulés sont au cœur de nombreuses applications (assemblage, pick-and-place, inspection, manipulation de pièces), où la qualité du suivi de trajectoire et la stabilité de la commande conditionnent directement la performance globale du système.

Dans ce rapport, nous nous intéressons à un bras manipulateur articulé de type **3RRR** (trois articulations rototoïdes), commandé par un **contrôleur PID**. L'objectif global est d'améliorer les performances de la commande en réglant automatiquement les gains PID à l'aide de l'algorithme **d'Évolution Différentielle (DE)**.

1.2 Présentation du bras manipulateur 3RRR

1.2.1 Définition et caractéristiques générales

Un bras manipulateur est un système mécanique articulé programmable, capable de positionner un effecteur dans un espace de travail. La mobilité du bras est définie par ses **degrés de liberté (DDL)**. Dans notre cas, le robot est un manipulateur **3RRR** : il comporte **trois liaisons rototoïdes** successives, permettant d'orienter la base, puis de déplacer le bras dans un plan de travail lié à la base.



FIGURE 1.1 – Bras manipulateur étudié (configuration 3RRR).

1.2.2 Architecture mécanique

La structure mécanique d'un manipulateur 3RRR peut être décrite par une chaîne cinématique série composée de :

- **Une base** : support fixe assurant la stabilité et portant la première articulation.
- **Trois articulations rototoïdes** : assurant des rotations autour de leurs axes.
- **Deux segments principaux (bras et avant-bras)** : reliant les articulations.
- **Un effecteur terminal** : extrémité opérationnelle (pointe, pince, outil, etc.).

1.2.3 Configuration cinématique (DDL)

Pour la modélisation fonctionnelle, on adopte une description standard des articulations :

1. q_1 : **rotation de base** (orientation autour de l'axe vertical).
2. q_2 : **rotation d'épaule** (élévation/abaissement du bras).
3. q_3 : **rotation de coude** (pliage/dépliage de l'avant-bras).

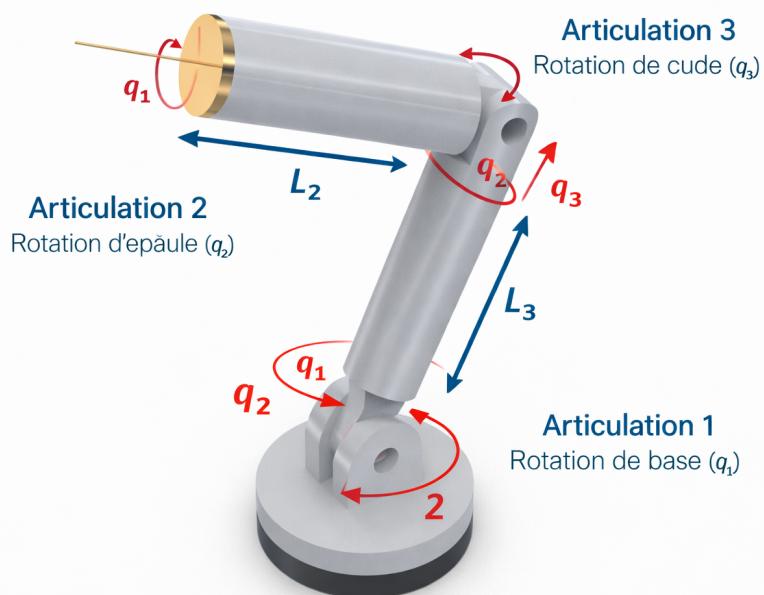


FIGURE 1.2 – Architecture mécanique et repérage des articulations (q_1, q_2, q_3) du bras 3RRR.

1.3 Notions de modélisation

1.3.1 Modèle géométrique direct (MGD)

Le **modèle géométrique direct** exprime la pose (position et éventuellement orientation) de l'effecteur en fonction des variables articulaires :

$$\mathbf{p} = f(\mathbf{q}), \quad \mathbf{q} = [q_1 \ q_2 \ q_3]^T, \quad (1.1)$$

où $\mathbf{p} = [x \ y \ z]^T$ est la position de l'effecteur dans le repère base. Dans un cadre homogène (cinématique de Denavit–Hartenberg), la transformation base–effecteur s'écrit :

$${}^0\mathbf{T}_3(\mathbf{q}) = {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) {}^2\mathbf{T}_3(q_3), \quad (1.2)$$

et la position \mathbf{p} est extraite du vecteur de translation de ${}^0\mathbf{T}_3$. Le MGD est utilisé pour simuler le mouvement, vérifier l'atteignabilité et visualiser les trajectoires.

1.3.2 Modèle géométrique inverse (MGI)

Le **modèle géométrique inverse** détermine les angles articulaires permettant d'atteindre une pose cible :

$$\mathbf{q} = f^{-1}(\mathbf{p}_d), \quad (1.3)$$

où \mathbf{p}_d est la position désirée de l'effecteur. Pour un manipulateur 3RRR, le MGI peut admettre **plusieurs solutions** (par exemple « coude haut / coude bas »), et devenir sensible aux **singularités**.

1.3.3 Jacobien cinématique (relation vitesses)

La relation entre vitesses articulaires et vitesse cartésienne de l'effecteur s'exprime par le **jacobien** :

$$\dot{\mathbf{p}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad (1.4)$$

où $\mathbf{J}(\mathbf{q})$ est une matrice dépendant de la configuration. Lorsque $\det(\mathbf{J}) = 0$ (ou lorsque \mathbf{J} perd son rang), le robot est en **singularité** et certaines directions de mouvement deviennent difficiles à contrôler.

1.3.4 Modèle dynamique

Le **modèle dynamique** relie couples moteurs et mouvement articulaire :

$$\tau = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}), \quad (1.5)$$

où :

- τ : vecteur des couples (ou efforts) aux articulations,
- $\mathbf{M}(\mathbf{q})$: matrice d'inertie,
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$: termes centrifuges/coriolis,
- $\mathbf{G}(\mathbf{q})$: gravité,
- $\mathbf{F}(\dot{\mathbf{q}})$: frottements (visqueux, secs, etc.).

1.4 Commande PID : principe et intérêt

1.4.1 Principe général

Le contrôleur **PID** (Proportionnel–Intégral–Dérivé) est largement utilisé en industrie. Pour une articulation donnée, le signal de commande (couple/effort) peut s'écrire :

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1.6)$$

où $e(t) = q_d(t) - q(t)$ est l'erreur entre consigne et mesure (dans une boucle de position articulaire).

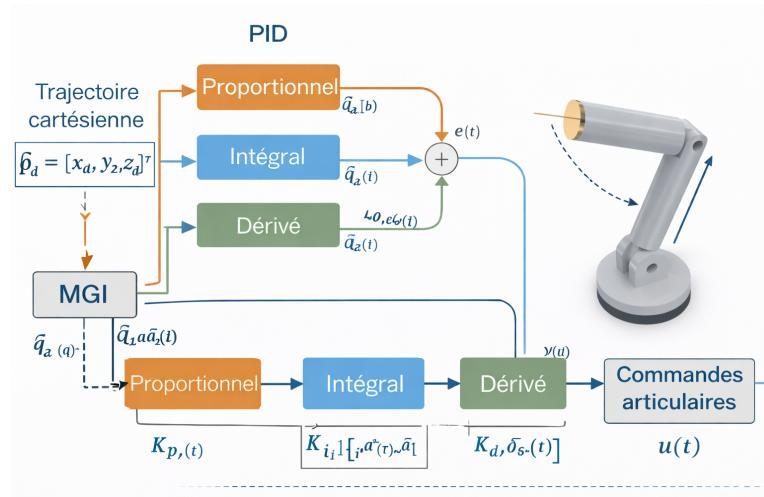


FIGURE 1.3 – Architecture de la commande PID du bras manipulateur 3RRR
(boucle de rétroaction articulaire)

1.4.2 Intérêt et difficultés de réglage

Le PID est attractif par sa simplicité, mais son réglage devient délicat pour un bras robotique car :

- le système est **non linéaire** et **couplé**,
- les paramètres varient (charge utile, frottements),
- un compromis est nécessaire entre **rapidité**, **dépassemement** et **erreur**.

Cela motive une approche de réglage automatique des gains.

1.5 Algorithme d'Évolution Différentielle (DE) : vue d'ensemble

1.5.1 Principe

DE est un algorithme d'optimisation stochastique basé sur une population de solutions candidates. Chaque individu représente un vecteur de paramètres à optimiser, ici typiquement :

$$\mathbf{x} = [K_p \ K_i \ K_d]^\top \quad (\text{ou, pour 3 articulations : } \mathbf{x} = [K_{p1}, K_{i1}, K_{d1}, \dots, K_{p3}, K_{i3}, K_{d3}]^\top). \quad (1.7)$$

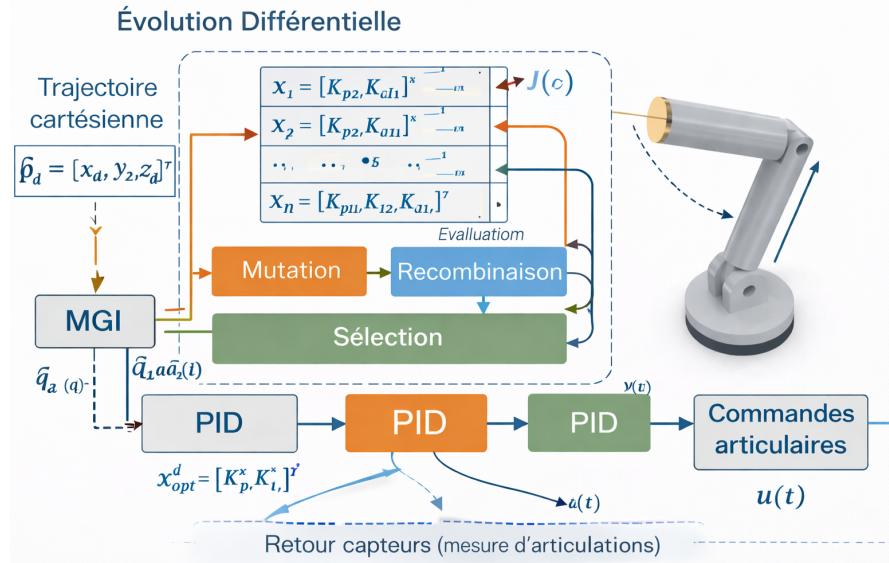


FIGURE 1.4 – Architecture de l'optimisation des gains PID par Évolution Différentielle (DE) pour le bras 3RRR

1.5.2 Règles générales de DE

Les trois opérations fondamentales sont :

- **Mutation** : création d'un vecteur mutant à partir de différences entre individus,
- **Recombinaison** : mélange du mutant avec l'individu courant,
- **Sélection** : conservation du meilleur selon une fonction coût.

Une écriture standard de la mutation est :

$$\mathbf{v}_i = \mathbf{x}_{r1} + F (\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad (1.8)$$

avec $F \in (0, 2)$ le facteur d'échelle et $r1, r2, r3$ des indices aléatoires distincts. Le candidat final est accepté s'il améliore la fonction objectif.

1.5.3 Pourquoi DE pour l'optimisation PID ?

DE est adapté car :

- il ne nécessite pas de gradient (robuste sur des fonctions coût non lisses),
- il limite le risque de rester bloqué dans des optima locaux,
- il est simple à implémenter et efficace pour des variables continues.

1.6 Problématique et objectifs du travail

1.6.1 Problématique

Comment améliorer les performances de suivi d'un bras manipulateur 3RRR, commandé par PID, en automatisant le réglage des gains via l'algorithme DE, tout en conservant stabilité et robustesse face aux perturbations et variations paramétriques ?

1.6.2 Objectifs

- Présenter le robot 3RRR et son contexte de commande.
- Poser le cadre de modélisation (MGD/MGI/Jacobien/dynamique) nécessaire à la commande.
- Définir une fonction coût de performance (erreur de suivi, dépassement, effort de commande, etc.).
- Appliquer DE afin d'obtenir un réglage PID améliorant le suivi et la robustesse.

1.7 Conclusion

Ce premier chapitre a permis de poser le cadre général du travail. Nous avons présenté le contexte et la problématique, puis défini les objectifs à atteindre ainsi que les hypothèses et notions fondamentales nécessaires à la compréhension de la suite. Cette introduction structure l'étude et prépare la transition vers le Chapitre 2, consacré à la méthodologie et à l'implémentation des calculs.

Chapitre 2

Méthodologie et Implémentation

2.1 Modèle Géométrique Direct

2.1.1 Introduction

L'objectif du modèle géométrique direct (MGD) est de déterminer la position et l'orientation de l'organe terminal en fonction des coordonnées articulaires. Pour cela, nous utilisons la convention de Denavit-Hartenberg (DH) qui permet d'établir les matrices de transformation homogène entre les repères successifs.

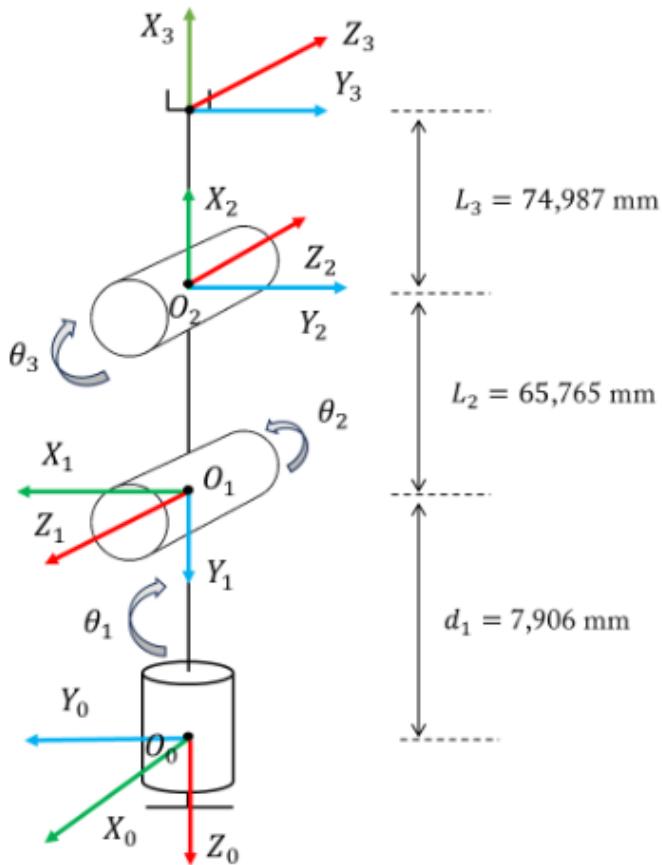


FIGURE 2.1 – Architecture de l'optimisation des gains PID par Évolution Différentielle (DE) pour le bras 3RRR

2.1.2 Initialisation du Tableau DH

La convention de Denavit-Hartenberg associe à chaque articulation quatre paramètres : a_i (longueur du segment), α_i (angle de torsion), d_i (décalage axial) et θ_i (angle articulaire). La matrice de transformation homogène générale entre deux repères consécutifs R_{i-1} et R_i est donnée par :

$$M_{i-1,i} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

Pour notre robot à 3 degrés de liberté, nous définissons les variables symboliques suivantes :

- d_1 : décalage axial de la première articulation
- L_2, L_3 : longueurs des segments 2 et 3
- $\theta_1, \theta_2, \theta_3$: angles articulaires

Le tableau des paramètres DH est le suivant :

$$\text{DH_Table} = \begin{pmatrix} a_i & \alpha_i & d_i & \theta_i \\ 0 & \frac{\pi}{2} & -d_1 & \theta_1 - \frac{\pi}{2} \\ L_2 & \pi & 0 & \theta_2 - \frac{\pi}{2} \\ L_3 & 0 & 0 & \theta_3 \end{pmatrix} \quad (2.2)$$

2.2 Calcul des Matrices de Transformation M_{01} , M_{12} et M_{23}

À partir des paramètres DH, nous calculons les matrices de transformation homogène entre repères consécutifs.

2.2.1 Matrice M_{01} (Repère 0 vers Repère 1)

$$M_{01} = \begin{pmatrix} \cos(\theta_1 - \frac{\pi}{2}) & 0 & \sin(\theta_1 - \frac{\pi}{2}) & 0 \\ \sin(\theta_1 - \frac{\pi}{2}) & 0 & -\cos(\theta_1 - \frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

En simplifiant les expressions trigonométriques, on obtient :

$$M_{01} = \begin{pmatrix} \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ -\cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ 0 & 1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

2.2.2 Matrice M_{12} (Repère 1 vers Repère 2)

$$M_{12} = \begin{pmatrix} \sin(\theta_2) & -\cos(\theta_2) & 0 & L_2 \sin(\theta_2) \\ -\cos(\theta_2) & -\sin(\theta_2) & 0 & -L_2 \cos(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

2.2.3 Matrice M_{23} (Repère 2 vers Repère 3)

$$M_{23} = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & L_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

2.3 Calcul des Matrices M_{02} et M_{03}

Les matrices de transformation cumulées permettent d'exprimer la position et l'orientation d'un repère par rapport au repère de base.

2.3.1 Matrice M_{02} (Repère 0 vers Repère 2)

La matrice M_{02} est obtenue par composition : $M_{02} = M_{01} \cdot M_{12}$

$$M_{02} = \begin{pmatrix} \sin(\theta_1) \sin(\theta_2) & -\cos(\theta_2) \sin(\theta_1) & \cos(\theta_1) & L_2 \sin(\theta_1) \sin(\theta_2) \\ -\cos(\theta_1) \sin(\theta_2) & \cos(\theta_1) \cos(\theta_2) & \sin(\theta_1) & -L_2 \cos(\theta_1) \sin(\theta_2) \\ -\cos(\theta_2) & -\sin(\theta_2) & 0 & -d_1 - L_2 \cos(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

2.3.2 Matrice M_{03} (Repère 0 vers Repère 3)

La matrice M_{03} représente la transformation complète du repère de base vers l'organe terminal : $M_{03} = M_{01} \cdot M_{12} \cdot M_{23}$

En posant $\sigma_1 = L_2 \sin(\theta_2) + L_3 \sin(\theta_2 - \theta_3)$, on obtient :

$$M_{03} = \begin{pmatrix} \sin(\theta_2 - \theta_3) \sin(\theta_1) & -\cos(\theta_2 - \theta_3) \sin(\theta_1) & \cos(\theta_1) & \sin(\theta_1) \cdot \sigma_1 \\ -\sin(\theta_2 - \theta_3) \cos(\theta_1) & \cos(\theta_2 - \theta_3) \cos(\theta_1) & \sin(\theta_1) & -\cos(\theta_1) \cdot \sigma_1 \\ -\cos(\theta_2 - \theta_3) & -\sin(\theta_2 - \theta_3) & 0 & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

où la composante P_z de la position est :

$$P_z = -d_1 - L_2 \cos(\theta_2) - L_3 \cos(\theta_2 - \theta_3) \quad (2.9)$$

2.4 Modèle Géométrique Inverse : Méthode de Paul

Le modèle géométrique inverse (MGI) permet de déterminer les coordonnées articulaires $(\theta_1, \theta_2, \theta_3)$ à partir de la position et de l'orientation désirées de l'organe terminal. La méthode de Paul consiste à pré-multiplier successivement l'équation de fermeture par les inverses des matrices de transformation.

2.4.1 Situation du Repère R_n (organe terminal)

La matrice de transformation désirée U_0 représente la situation de l'organe terminal dans le repère de base. Elle est composée des vecteurs d'orientation (S, N, A) et du vecteur position P :

$$U_0 = \begin{pmatrix} S_x & N_x & A_x & P_x \\ S_y & N_y & A_y & P_y \\ S_z & N_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

où :

- $\vec{S} = (S_x, S_y, S_z)^T$: vecteur de glissement (sliding)
- $\vec{N} = (N_x, N_y, N_z)^T$: vecteur normal
- $\vec{A} = (A_x, A_y, A_z)^T$: vecteur d'approche
- $\vec{P} = (P_x, P_y, P_z)^T$: vecteur position

L'équation de fermeture s'écrit :

$$U_0 = M_{01} \cdot M_{12} \cdot M_{23} \quad (2.11)$$

2.4.2 Première Étape : Pré-multiplication par M_{01}^{-1}

Nous calculons d'abord l'inverse de la matrice M_{01} :

$$M_{01}^{-1} = \begin{pmatrix} \sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ -\cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

En pré-multipliant l'équation de fermeture par M_{01}^{-1} , on obtient :

$$M_{01}^{-1} \cdot U_0 = M_{12} \cdot M_{23} \quad (2.13)$$

$$\left(\begin{array}{l} \begin{array}{lll} Sx \sin \theta_1 - Sy \cos \theta_1 = \sigma_2 - \sigma_3 & Nx \sin \theta_1 - Ny \cos \theta_1 = \sigma_1 & Ax \sin \theta_1 - Ay \cos \theta_1 = 0 \\ Sz = \sigma_1 & Nz = \sigma_3 - \sigma_2 & Az = 0 \\ -Sx \cos \theta_1 - Sy \sin \theta_1 = 0 & -Nx \cos \theta_1 - Ny \sin \theta_1 = 0 & -Ax \cos \theta_1 - Ay \sin \theta_1 = -1 \\ 0 = 0 & 0 = 0 & 0 = 0 \end{array} \\ \begin{array}{l} Px \sin \theta_1 - Py \cos \theta_1 = L_2 \sin \theta_2 - L_3 \cos \theta_2 \sin \theta_3 + L_3 \cos \theta_3 \sin \theta_2 \\ Pz + d_1 = -L_2 \cos \theta_2 - L_3 \cos \theta_2 \cos \theta_3 - L_3 \sin \theta_2 \sin \theta_3 \\ -Px \cos \theta_1 - Py \sin \theta_1 = 0 \\ 1 = 1 \end{array} \end{array} \right)$$

2.4.3 Ensemble des Solutions pour θ_1

En égalant l'élément (3, 4) des deux membres de l'équation, on obtient :

$$-P_x \cos(\theta_1) - P_y \sin(\theta_1) = 0 \quad (2.14)$$

La résolution de cette équation donne deux solutions possibles pour θ_1 :

$$\theta_1 = \begin{cases} 2 \arctan \left(\frac{P_y + \sqrt{P_x^2 + P_y^2}}{P_x} \right) \\ 2 \arctan \left(\frac{P_y - \sqrt{P_x^2 + P_y^2}}{P_x} \right) \end{cases} \quad (2.15)$$

2.4.4 Deuxième Étape : Pré-multiplication par $M_{01}^{-1} \cdot M_{12}^{-1}$

L'inverse de M_{12} est calculé, puis on pré-multiplie à nouveau :

$$M_{01}^{-1} \cdot M_{12}^{-1} \cdot U_0 = M_{23} \quad (2.16)$$

$$\left(\begin{array}{ll} S_x c_{12} - S_y s_{12} = c_3 & N_x c_{12} - N_y s_{12} = -s_3 \\ -S_z = s_3 & -N_z = c_3 \\ S_y c_{12} + S_x s_{12} = 0 & N_y c_{12} + N_x s_{12} = 0 \\ 0 = 0 & 0 = 0 \\ \hline A_x c_{12} - A_y s_{12} = 0 & P_x c_{12} - L_2 s_1 - P_y s_{12} = L_3 c_3 \\ -A_z = 0 & d_1 - P_z = L_3 s_3 \\ A_y c_{12} + A_x s_{12} = 1 & L_2 c_1 + P_y c_{12} + P_x s_{12} = 0 \\ 0 = 0 & 1 = 1 \end{array} \right)$$

where $c_k = \cos \theta_k$, $s_k = \sin \theta_k$, $c_{12} = \cos(\theta_1 - \theta_2)$, $s_{12} = \sin(\theta_1 - \theta_2)$.

2.5 Deuxième Étape : Pré-multiplication par $M_{01}^{-1} \cdot M_{12}^{-1}$

L'inverse de M_{12} est calculé, puis on pré-multiplie à nouveau :

$$M_{01}^{-1} \cdot M_{12}^{-1} \cdot U_0 = M_{23} \quad (2.17)$$

2.5.1 Ensemble des Solutions pour θ_3

En identifiant l'élément (2, 4) des deux côtés de l'équation matricielle, on obtient la relation suivante :

$$d_1 - P_z = L_3 \sin(\theta_3) \quad (2.18)$$

En isolant $\sin(\theta_3)$, on a $\sin(\theta_3) = \frac{d_1 - P_z}{L_3}$. La résolution de cette équation donne les solutions possibles pour θ_3 :

$$\theta_3 = \begin{cases} \arcsin\left(\frac{d_1 - P_z}{L_3}\right) \\ \pi - \arcsin\left(\frac{d_1 - P_z}{L_3}\right) \end{cases} \quad (2.19)$$

2.5.2 Ensemble des Solutions pour θ_2

En utilisant l'élément (1, 4) de l'équation matricielle :

$$P_x \cos(\theta_1 - \theta_2) - L_2 \sin(\theta_1) - P_y \sin(\theta_1 - \theta_2) = L_3 \cos(\theta_3) \quad (2.20)$$

En posant $\sigma_1 = \sqrt{P_x^2 + P_y^2 - L_2^2 \sin^2(\theta_1) - 2L_2 L_3 \cos(\theta_3) \sin(\theta_1) - L_3^2 \cos^2(\theta_3)}$, les solutions sont :

$$\theta_2 = \begin{cases} \theta_1 + 2 \arctan \left(\frac{P_y + \sigma_1}{P_x + L_3 \cos(\theta_3) + L_2 \sin(\theta_1)} \right) \\ \theta_1 + 2 \arctan \left(\frac{P_y - \sigma_1}{P_x + L_3 \cos(\theta_3) + L_2 \sin(\theta_1)} \right) \end{cases} \quad (2.21)$$

2.6 Modèle Géométrique Inverse : Méthode Géométrique

La méthode géométrique utilise des relations trigonométriques basées sur la géométrie du robot. Cette approche est souvent plus intuitive et permet une meilleure compréhension physique du problème.

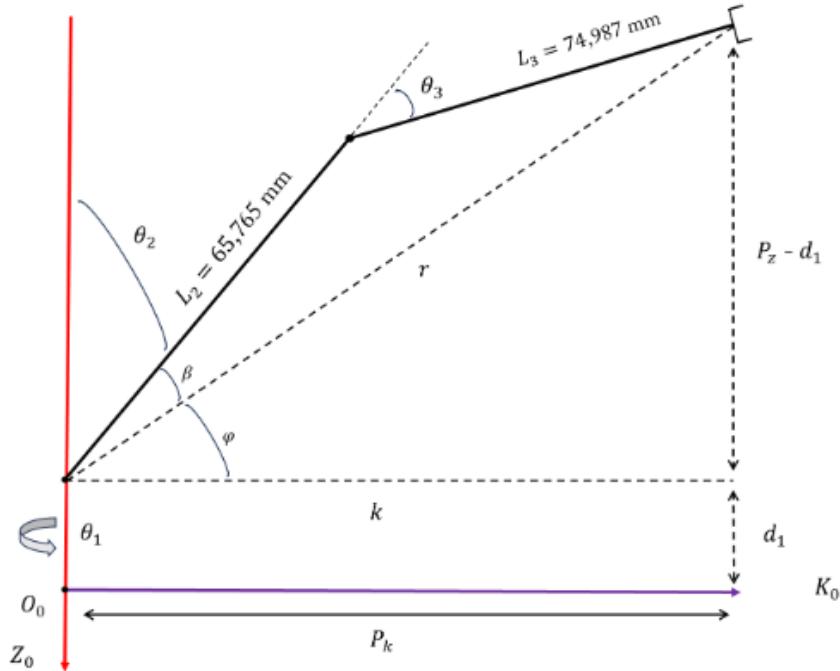


FIGURE 2.2 – Modèle Géométrique Inverse

2.6.1 Définition des Variables Intermédiaires

Nous définissons les grandeurs géométriques suivantes :

$$P_k = \sqrt{P_x^2 + P_y^2} \quad (2.22)$$

$$r = \sqrt{P_k^2 + (P_z - d_1)^2} = \sqrt{P_x^2 + P_y^2 + (P_z - d_1)^2} \quad (2.23)$$

où P_k représente la projection du point P dans le plan (x_0, y_0) et r est la distance entre le point P et l'origine du repère R_1 .

2.6.2 Solution pour θ_1

L'angle θ_1 correspond à la rotation autour de l'axe vertical z_0 . Il est directement obtenu par :

$$\boxed{\theta_1 = \text{atan2}(P_y, P_x)} \quad (2.24)$$

2.6.3 Solution pour θ_2

Pour déterminer θ_2 , nous utilisons les angles auxiliaires α et β :

$$\sin(\alpha) = \frac{P_z - d_1}{r} \Rightarrow \alpha = \arcsin\left(\frac{P_z - d_1}{\sqrt{P_x^2 + P_y^2 + (P_z - d_1)^2}}\right) \quad (2.25)$$

L'angle β est obtenu par le théorème d'Al-Kashi appliqué au triangle formé par les segments L_2 , L_3 et r :

$$\cos(\beta) = \frac{L_2^2 + r^2 - L_3^2}{2L_2r} \Rightarrow \beta = \arccos\left(\frac{L_2^2 - L_3^2 + P_x^2 + P_y^2 + (P_z - d_1)^2}{2L_2\sqrt{P_x^2 + P_y^2 + (P_z - d_1)^2}}\right) \quad (2.26)$$

La solution pour θ_2 est alors :

$$\boxed{\theta_2 = \frac{\pi}{2} - \alpha - \beta = \frac{\pi}{2} - \arcsin\left(\frac{P_z - d_1}{r}\right) - \arccos\left(\frac{L_2^2 + r^2 - L_3^2}{2L_2r}\right)} \quad (2.27)$$

2.6.4 Solution pour θ_3

L'angle θ_3 est obtenu en appliquant le théorème d'Al-Kashi au même triangle :

$$\cos(\theta_3) = -\frac{L_2^2 + L_3^2 - r^2}{2L_2L_3} \quad (2.28)$$

D'où :

$$\boxed{\theta_3 = \arccos \left(\frac{P_x^2 + P_y^2 + (P_z - d_1)^2 - L_2^2 - L_3^2}{2L_2L_3} \right)} \quad (2.29)$$

2.7 Trajectoire Calculée

2.7.1 Définition de la Trajectoire Articulaire

Pour assurer un mouvement fluide (continuité de la position, vitesse et accélération), nous utilisons un **polynôme d'ordre 5** pour le paramètre de progression temporelle.

2.7.2 Paramètre de Progression $s(t)$

Le paramètre $s(t)$ évolue de 0 à 1 sur l'intervalle $[0, T_{\text{total}}]$. Le polynôme d'ordre 5 garantit que :

- $s(0) = 0, s(T_{\text{total}}) = 1$
- $\dot{s}(0) = \dot{s}(T_{\text{total}}) = 0$ (vitesse nulle aux extrémités)
- $\ddot{s}(0) = \ddot{s}(T_{\text{total}}) = 0$ (accélération nulle aux extrémités)

$$\boxed{s(t) = 10 \left(\frac{t}{T_{\text{total}}} \right)^3 - 15 \left(\frac{t}{T_{\text{total}}} \right)^4 + 6 \left(\frac{t}{T_{\text{total}}} \right)^5} \quad (2.30)$$

2.7.3 Évolution des Angles Articulaires

Chaque angle articulaire évolue linéairement en fonction de $s(t)$ entre sa valeur initiale et sa valeur finale :

$$s(t) = 10 \left(\frac{t}{T_{\text{total}}} \right)^3 - 15 \left(\frac{t}{T_{\text{total}}} \right)^4 + 6 \left(\frac{t}{T_{\text{total}}} \right)^5$$

$$\theta_1(t) = \theta_{1,\text{start}} + s(t)(\theta_{1,\text{end}} - \theta_{1,\text{start}})$$

$$\theta_1(t) = \theta_{1,\text{start}} + (\theta_{1,\text{end}} - \theta_{1,\text{start}}) \left(\frac{10t^3}{T_{\text{total}}^3} - \frac{15t^4}{T_{\text{total}}^4} + \frac{6t^5}{T_{\text{total}}^5} \right)$$

$$\theta_2(t) = \theta_{2,\text{start}} + s(t)(\theta_{2,\text{end}} - \theta_{2,\text{start}})$$

$$\theta_2(t) = \theta_{2,\text{start}} + (\theta_{2,\text{end}} - \theta_{2,\text{start}}) \left(\frac{10t^3}{T_{\text{total}}^3} - \frac{15t^4}{T_{\text{total}}^4} + \frac{6t^5}{T_{\text{total}}^5} \right)$$

$$\theta_3(t) = \theta_{3,\text{start}} + s(t)(\theta_{3,\text{end}} - \theta_{3,\text{start}})$$

$$\theta_3(t) = \theta_{3,\text{start}} + (\theta_{3,\text{end}} - \theta_{3,\text{start}}) \left(\frac{10t^3}{T_{\text{total}}^3} - \frac{15t^4}{T_{\text{total}}^4} + \frac{6t^5}{T_{\text{total}}^5} \right)$$

Cette formulation garantit un mouvement lisse avec des phases d'accélération et de décélération progressives.

2.8 Modèle Dynamique

Le modèle dynamique établit la relation entre les forces/couples appliqués aux articulations et le mouvement résultant. Nous utilisons le formalisme de Lagrange qui repose sur les énergies cinétique et potentielle du système.

2.8.1 Rappel des Matrices de Transformation

Pour le calcul dynamique, nous rappelons les matrices de transformation homogène simplifiées :

$$M_{01} = \begin{pmatrix} \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ -\cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ 0 & 1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.32)$$

2.8.2 Énergie Cinétique : Partie Translation

L'énergie cinétique de translation est calculée à partir des vitesses des centres de gravité de chaque corps.

2.8.3 Position des Centres de Gravité

Les vecteurs position entre les origines des repères sont extraits des matrices de transformation :

$$O_0\vec{O}_1 = \begin{pmatrix} 0 \\ 0 \\ -d_1 \end{pmatrix}, \quad O_1\vec{O}_2 = \begin{pmatrix} L_2 \sin(\theta_2) \\ -L_2 \cos(\theta_2) \\ 0 \end{pmatrix}, \quad O_2\vec{O}_3 = \begin{pmatrix} L_3 \cos(\theta_3) \\ L_3 \sin(\theta_3) \\ 0 \end{pmatrix} \quad (2.33)$$

Les centres de gravité sont supposés au milieu de chaque segment :

$$O\vec{G}_1 = \frac{1}{2}O_0\vec{O}_1 = \begin{pmatrix} 0 \\ 0 \\ -\frac{d_1}{2} \end{pmatrix} \quad (2.34)$$

$$O\vec{G}_2 = O_0\vec{O}_1 + \frac{1}{2}O_1\vec{O}_2 = \begin{pmatrix} \frac{L_2 \sin(\theta_2)}{2} \\ -\frac{L_2 \cos(\theta_2)}{2} \\ -d_1 \end{pmatrix} \quad (2.35)$$

$$O\vec{G}_3 = O_0\vec{O}_1 + O_1\vec{O}_2 + \frac{1}{2}O_2\vec{O}_3 = \begin{pmatrix} L_2 \sin(\theta_2) + \frac{L_3 \cos(\theta_3)}{2} \\ -L_2 \cos(\theta_2) + \frac{L_3 \sin(\theta_3)}{2} \\ -d_1 \end{pmatrix} \quad (2.36)$$

2.8.4 Calcul des Jacobiens pour les Vitesses Linéaires

Les vitesses linéaires des centres de gravité sont obtenues par dérivation :

$$\vec{V}_{G_1} = \vec{0} \quad (2.37)$$

$$\vec{V}_{G_2} = \begin{pmatrix} \frac{L_2 \dot{\theta}_2 \cos(\theta_2)}{2} \\ \frac{L_2 \dot{\theta}_2 \sin(\theta_2)}{2} \\ 0 \end{pmatrix} \quad (2.38)$$

$$\vec{V}_{G_3} = \begin{pmatrix} L_2 \dot{\theta}_2 \cos(\theta_2) - \frac{L_3 \dot{\theta}_3 \sin(\theta_3)}{2} \\ L_2 \dot{\theta}_2 \sin(\theta_2) + \frac{L_3 \dot{\theta}_3 \cos(\theta_3)}{2} \\ 0 \end{pmatrix} \quad (2.39)$$

Les Jacobiens de vitesse linéaire $J_{v,i}$ tels que $\vec{V}_{G_i} = J_{v,i} \cdot \dot{\vec{\theta}}$ sont :

$$J_{v,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.40)$$

$$J_{v,2} = \begin{pmatrix} 0 & \frac{L_2 \cos(\theta_2)}{2} & 0 \\ 0 & \frac{L_2 \sin(\theta_2)}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.41)$$

$$J_{v,3} = \begin{pmatrix} 0 & L_2 \cos(\theta_2) & -\frac{L_3 \sin(\theta_3)}{2} \\ 0 & L_2 \cos(\theta_2) & \frac{L_3 \cos(\theta_3)}{2} \\ 0 & 0 & 0 \end{pmatrix} \quad (2.42)$$

2.8.5 Matrice d'Inertie en Translation

L'énergie cinétique de translation s'exprime sous la forme $K_{\text{trans}} = \frac{1}{2} \vec{\theta}^T M_{\text{trans}} \vec{\theta}$, où :

$$M_{\text{trans}} = m_1 J_{v,1}^T J_{v,1} + m_2 J_{v,2}^T J_{v,2} + m_3 J_{v,3}^T J_{v,3} \quad (2.43)$$

$$\begin{aligned} & \frac{L^2}{4} \cos(\theta_1) \cos\left(\theta_3 + \frac{\pi}{4}\right) \\ & \sqrt{2} LaLm \cos(\theta_2) \cos\left(\theta_3 + \frac{\pi}{4}\right) \\ & \frac{\sqrt{2}}{4} LaLm \cos(\theta_2) \cos\left(\theta_3 + \frac{\pi}{4}\right) \end{aligned}$$

2.9 Énergie Cinétique : Partie Rotation

2.9.1 Jacobiens des Vitesses Angulaires

Les Jacobiens de vitesse angulaire définissent la contribution de chaque vitesse articulaire à la vitesse angulaire de chaque corps :

$$J_{\omega,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad (2.44)$$

$$J_{\omega,2} = \begin{pmatrix} 0 & -\cos(\theta_1) & 0 \\ 0 & -\sin(\theta_1) & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad (2.45)$$

$$J_{\omega,3} = \begin{pmatrix} 0 & -\cos(\theta_1) & \cos(\theta_1) \\ 0 & -\sin(\theta_1) & \sin(\theta_1) \\ 1 & 0 & 0 \end{pmatrix} \quad (2.46)$$

2.9.2 Matrices de Rotation

Les matrices de rotation R_{0i} (partie 3×3 supérieure gauche des M_{0i}) sont :

$$R_{01} = \begin{pmatrix} \sin(\theta_1) & 0 & -\cos(\theta_1) \\ -\cos(\theta_1) & 0 & -\sin(\theta_1) \\ 0 & 1 & 0 \end{pmatrix} \quad (2.47)$$

$$R_{02} = \begin{pmatrix} \sin(\theta_1) \sin(\theta_2) & -\cos(\theta_2) \sin(\theta_1) & \cos(\theta_1) \\ -\cos(\theta_1) \sin(\theta_2) & \cos(\theta_1) \cos(\theta_2) & \sin(\theta_1) \\ -\cos(\theta_2) & -\sin(\theta_2) & 0 \end{pmatrix} \quad (2.48)$$

$$R_{03} = \begin{pmatrix} \sin(\theta_2 - \theta_3) \sin(\theta_1) & -\cos(\theta_2 - \theta_3) \sin(\theta_1) & \cos(\theta_1) \\ -\sin(\theta_2 - \theta_3) \cos(\theta_1) & \cos(\theta_2 - \theta_3) \cos(\theta_1) & \sin(\theta_1) \\ -\cos(\theta_2 - \theta_3) & -\sin(\theta_2 - \theta_3) & 0 \end{pmatrix} \quad (2.49)$$

2.9.3 Tenseurs d'Inertie

Les propriétés inertielles des corps sont définies par les tenseurs d'inertie et les masses :

Masses des composants (en kg) :

$$m_{P1} = 0,006509 \text{ kg} \quad m_{P2} = 0,008157 \text{ kg} \quad (2.50)$$

$$m_{P3} = 0,010337 \text{ kg} \quad m_{P4} = 0,009495 \text{ kg} \quad (2.51)$$

$$m_{\text{End}} = 0,000363 \text{ kg} \quad (2.52)$$

Tenseur d'inertie I_1 (pour P1 + P2) :

$$I_1 = I_{P1} + I_{P2} = 10^{-6} \times \begin{pmatrix} 1,4992 & 0 & 0 \\ 0 & 1,5154 & 0 \\ 0 & 0 & 2,6475 \end{pmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.53)$$

avec $m_1 = m_{P1} + m_{P2} = 0,014666 \text{ kg}$.

Tenseur d'inertie I_2 (pour P3) :

$$I_2 = 10^{-6} \times \begin{pmatrix} 3,2476 & -7,994 \times 10^{-6} & -2,058 \times 10^{-6} \\ -7,994 \times 10^{-6} & 3,2578 & -1,025 \times 10^{-6} \\ -2,058 \times 10^{-6} & -1,025 \times 10^{-6} & 0,2857 \end{pmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.54)$$

avec $m_2 = m_{P2} = 0,008157 \text{ kg}$.

Tenseur d'inertie I_3 (pour P4 + End) :

$$I_3 = 10^{-6} \times \begin{pmatrix} 2,4439 & 0 & 0 \\ 0 & 2,4406 & 0 \\ 0 & 0 & 0,2689 \end{pmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.55)$$

avec $m_3 = m_{P4} + m_{\text{End}} = 0,009858 \text{ kg}$.

2.9.4 Matrice d'Inertie en Rotation

L'énergie cinétique de rotation s'exprime par :

$$K_{\text{rot}} = \frac{1}{2} \dot{\vec{\theta}}^T M_{\text{rot}} \dot{\vec{\theta}} \quad (2.56)$$

où la matrice M_{rot} est donnée par :

$$M_{\text{rot}} = \sum_{i=1}^3 J_{\omega,i}^T R_{0i} I_i R_{0i}^T J_{\omega,i} \quad (2.57)$$

2.10 Énergie Cinétique Totale

L'énergie cinétique totale du système est la somme des contributions de translation et de rotation :

$$K_{\text{total}} = \frac{1}{2} \dot{\vec{\theta}}^T (M_{\text{trans}} + M_{\text{rot}}) \dot{\vec{\theta}} \quad (2.58)$$

où $\dot{\vec{\theta}} = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)^T$ est le vecteur des vitesses articulaires.

2.11 Énergie Potentielle

L'énergie potentielle gravitationnelle est calculée en fonction de la hauteur des centres de masse par rapport à un niveau de référence. En prenant l'axe z dirigé vers le haut et g l'accélération de la pesanteur :

$$V = g \cdot \vec{z}^T \cdot \left(m_1 O\vec{G}_1 + m_2 O\vec{G}_2 + m_3 O\vec{G}_3 \right) \quad (2.59)$$

Avec les valeurs numériques des masses, on obtient :

$$V \approx -0,0253 \cdot d_1 \cdot g \quad (2.60)$$

2.12 Lagrangien du Système

Le Lagrangien \mathcal{L} est défini comme la différence entre l'énergie cinétique et l'énergie potentielle :

$$\boxed{\mathcal{L} = K_{\text{total}} - V} \quad (2.61)$$

Les équations du mouvement sont obtenues par les équations de Lagrange :

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} \right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = \tau_i, \quad i = 1, 2, 3 \quad (2.62)$$

où τ_i est le couple appliqué à l'articulation i .

Ces équations peuvent être mises sous la forme matricielle standard :

$$M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) \dot{\theta} + G(\theta) = \tau \quad (2.63)$$

où :

- $M(\theta)$: matrice d'inertie généralisée
- $C(\theta, \dot{\theta})$: matrice des effets de Coriolis et centrifuges
- $G(\theta)$: vecteur des couples gravitationnels
- τ : vecteur des couples articulaires

$$\begin{aligned} & \frac{7306063574693589 d_1 g}{288230376151711744} - \frac{2539688694487295 \dot{\theta}_2 \dot{\theta}_3}{9444732965739290427392} - \frac{6021017265659 \dot{\theta}_1^2 \cos(2\theta_2)}{2361183241434822606848} \\ & - \frac{4948046337023665 \dot{\theta}_1^2 \sin(2\theta_2)}{1237940039285380274899124224} + \frac{34049915169759145 \dot{\theta}_1^2}{9444732965739290427392} + \frac{2619024451399505 \dot{\theta}_2^2}{9444732965739290427392} \\ & + \frac{2539688694487295 \dot{\theta}_3^2}{18889465931478580854784} - \frac{5095361201698625 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2}{2475880078570760549798248448} \\ & - \frac{5075554161070059 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2}{4951760157141521099596496896} + \frac{L_2^2 m_2 \dot{\theta}_2^2}{8} + L_2^2 m_3 \dot{\theta}_2^2 + \frac{L_3^2 m_3 \dot{\theta}_3^2}{8} \\ & + \frac{7791904696735 \dot{\theta}_1^2 \cos(2\theta_2) \cos(2\theta_3)}{9444732965739290427392} + \frac{7791904696735 \dot{\theta}_1^2 \sin(2\theta_2) \sin(2\theta_3)}{9444732965739290427392} \\ & - L_2^2 m_3 \dot{\theta}_2^2 \sin^2 \theta_2 + \frac{\sqrt{2} L_2 L_3 m_3 \dot{\theta}_2 \dot{\theta}_3 \cos \theta_2 \cos(\theta_3 + \frac{\pi}{4})}{2} \end{aligned}$$

2.13 conclusion

Ce chapitre a présenté la méthodologie adoptée ainsi que son implémentation. L'ensemble des étapes de calcul a été détaillé afin d'obtenir un modèle complet et exploitable : définition des paramètres, formulation des équations, et établissement des expressions nécessaires (cinématique, interpolation temporelle des angles, etc.). Cette phase constitue la base du travail, car elle garantit la cohérence mathématique du modèle et prépare directement la validation et les résultats présentés dans les chapitres suivants.

Chapitre 3

Application et Simulation

3.1 Introduction

Ce chapitre présente la mise en œuvre, sous **MATLAB/Simulink**, de l'ensemble des blocs développés au Chapitre 2 afin de **simuler** le bras manipulateur **3RRR** et d'évaluer les performances de la stratégie de commande. L'objectif est de regrouper, dans une chaîne de simulation cohérente, (i) la génération de trajectoire, (ii) la conversion cartésienne → articulaire via le MGI, (iii) la commande PID articulaire, (iv) le modèle dynamique du robot et (v) le calcul des indicateurs de performance utilisés par l'algorithme **DE**.

3.2 Présentation des outils de simulation

MATLAB/Simulink a été retenu car il offre :

- un environnement robuste pour le calcul matriciel et la modélisation robotique ;
- des outils de simulation temporelle (solveurs ODE) adaptés aux systèmes dynamiques non linéaires ;
- une intégration simple des boucles de commande (PID) et des signaux de test ;
- la possibilité de structurer le modèle en **blocs** (modularité) et de générer automatiquement les tracés.

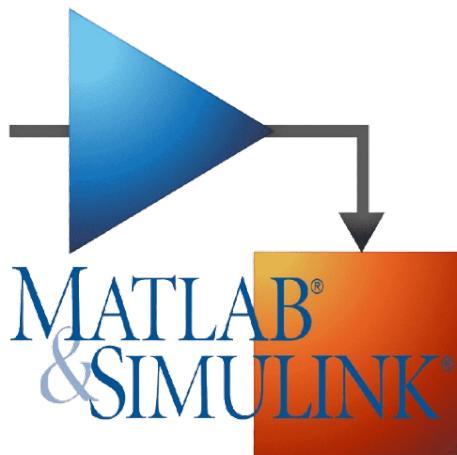


FIGURE 3.1 – Environnement MATLAB/Simulink utilisé pour la modélisation et la simulation.

3.3 Commande du bras : cas de modélisation dynamique

La Figure 3.2 illustre l'architecture adoptée. Le modèle est organisé en sous-systèmes indépendants afin de faciliter l'intégration, la validation et les tests. La chaîne complète suit la logique :

$$\mathbf{p}_d(t) \rightarrow \mathbf{q}_d(t) \rightarrow \text{PID} \rightarrow \boldsymbol{\tau}(t) \rightarrow \mathbf{q}(t)$$

où $\mathbf{p}_d(t)$ est la consigne cartésienne, $\mathbf{q}_d(t)$ la consigne articulaire, et $\boldsymbol{\tau}(t)$ les couples appliqués au robot.

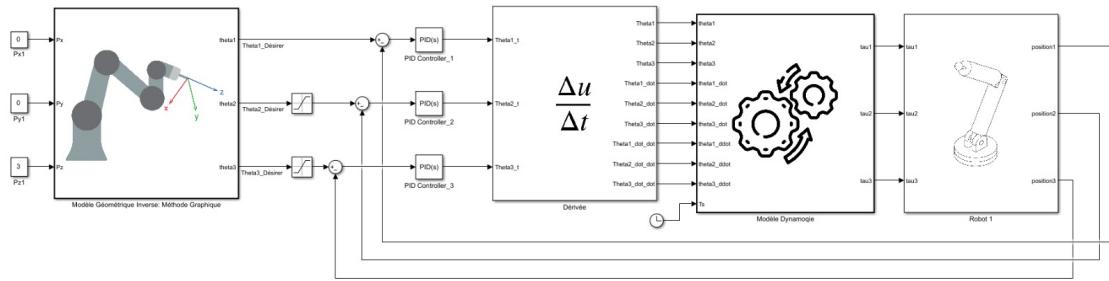


FIGURE 3.2 – Chaîne de simulation MATLAB/Simulink

3.4 Description des blocs MATLAB/Simulink

3.4.1 Bloc 1 : Génération de trajectoire

Ce bloc génère la trajectoire désirée de l'effecteur dans l'espace de travail (ex. trajectoire circulaire, point à point, trajectoire en L). En sortie, on obtient une consigne cartésienne $\mathbf{p}_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]^\top$, ainsi que, si nécessaire, ses dérivées temporelles (utiles pour l'analyse des vitesses et accélérations). Dans Simulink, ce bloc est typiquement implémenté à l'aide de générateurs de signaux, tables de consignes, ou fonctions paramétriques du temps.

3.4.2 Bloc 2 : Modèle géométrique inverse (MGI)

Le MGI convertit la trajectoire cartésienne en **consignes articulaires** :

$$\mathbf{q}_d(t) = [q_{1d}(t) \ q_{2d}(t) \ q_{3d}(t)]^\top.$$

Il réalise la transformation $(x_d, y_d, z_d) \mapsto (\theta_1, \theta_2, \theta_3)$ selon la méthode développée au chapitre précédent (méthode graphique). Ce bloc gère également :

- **atteignabilité** (position dans l'espace de travail) ;
- **limites articulaires** ;
- la **cohérence** de la solution choisie (éviter des sauts de configuration).

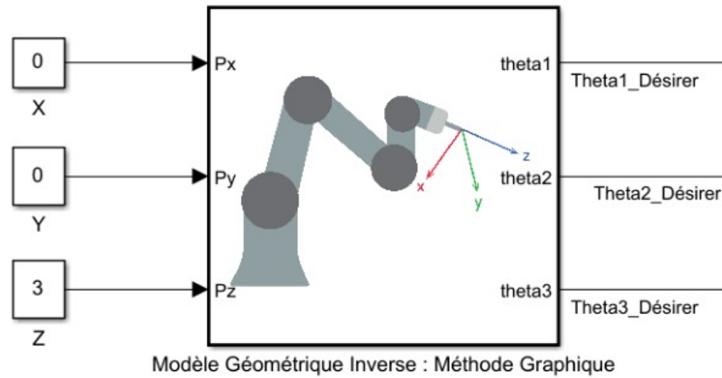


FIGURE 3.3 – Bloc MGI (méthode graphique) : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés $(\theta_{1d}, \theta_{2d}, \theta_{3d})$.

3.4.3 Bloc 3 : Commande PID articulaire

Chaque articulation est pilotée par une boucle PID indépendante (structure **multi-PID**) :

$$e_i(t) = q_{id}(t) - q_i(t), \quad i \in \{1, 2, 3\}.$$

Chaque contrôleur produit une commande (couple ou consigne d'actionneur) afin de réduire l'erreur. Le schéma de principe est :

$$\mathbf{q}_d(t) \rightarrow \underbrace{(\mathbf{q}_d - \mathbf{q})}_{\mathbf{e}(t)} \rightarrow \text{PID} \rightarrow \text{commande}$$

Dans l'implémentation, des éléments de sûreté peuvent être ajoutés :

- saturation des couples (limites actionneurs) ;
- anti-windup pour éviter l'accumulation excessive de l'intégrateur ;
- filtrage dérivé afin de réduire la sensibilité au bruit de mesure.

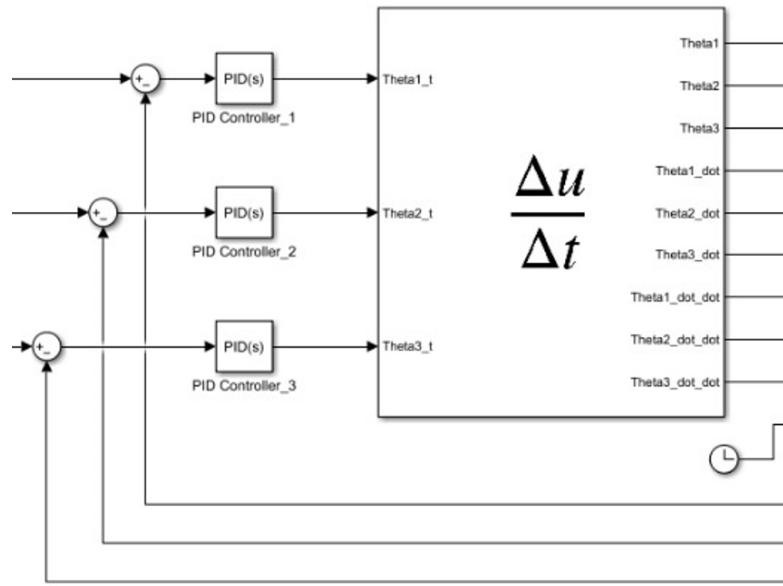


FIGURE 3.4 – Boucles PID articulaires

3.4.4 Bloc 4 : Modèle dynamique du robot 3RRR

Le modèle dynamique représente le robot et ses paramètres physiques (masses, inerties, frottements). Il reçoit la commande (ex. couples τ_1, τ_2, τ_3) et fournit l'évolution des variables d'état articulaires $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ et $\ddot{\mathbf{q}}(t)$, conformément aux équations du mouvement.

Dans la chaîne Simulink, ce bloc assure le passage :

$$\mathbf{q}(t) \rightarrow \dot{\mathbf{q}}(t) \rightarrow \ddot{\mathbf{q}}(t) \rightarrow \boldsymbol{\tau}(t)$$

via intégration temporelle (solveur). Il constitue le cœur physique de la simulation (réponse inertielle, effet des frottements, couplages dynamiques, etc.).

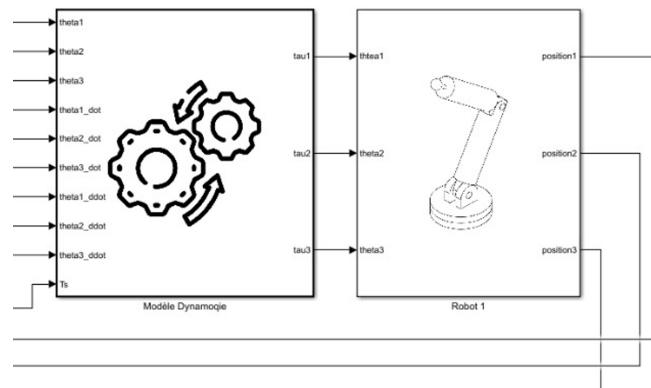


FIGURE 3.5 – Chaîne incluant le modèle dynamique : appliqués au robot.

Implémentation du robot (Simscape/Multibody). Le robot est modélisé par trois liaisons **revolute** (une par articulation), connectées à des corps rigides. Les positions articulaires mesurées (`position1..3`) sont rebouclées vers les sommateurs pour former les erreurs utilisées par les PID.

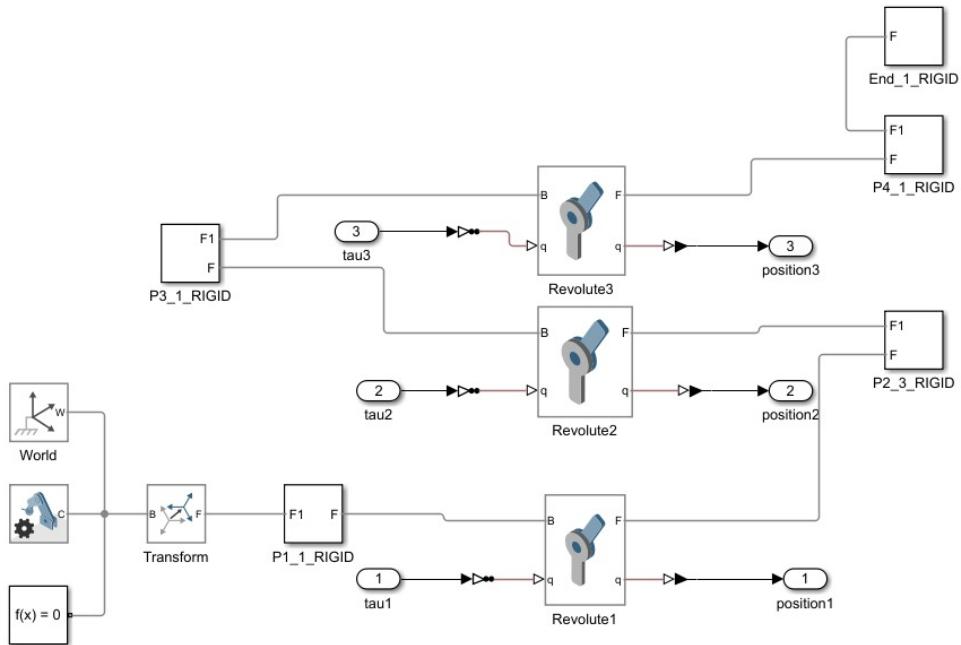


FIGURE 3.6 – Sous-système robot (Simscape Multibody) : trois articulations Revolute1..3 et mesures des positions articulaires (`position1..3`).

3.4.5 Bloc 5 : Calcul des métriques de performance (fonction coût)

Ce bloc calcule les indicateurs utilisés pour comparer les réglages PID et piloter l’optimisation. Les critères considérés incluent typiquement :

- **erreur de suivi** (ISE/IAE/ITAE) ;
- **dépassemement et temps de réponse** ;
- **effort de commande** (énergie ou norme des couples) ;
- pénalités en cas de saturation ou dépassement des limites articulaires.

Ces métriques permettent de résumer quantitativement la qualité du suivi, la rapidité de réponse et le coût énergétique associé à une configuration de gains PID.

3.5 Commande du bras : cas de géométrie inverse (MGI)

3.5.1 Principe global

Dans ce second cas, la consigne est définie dans l'espace cartésien $\mathbf{p}_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]^\top$ puis convertie en consignes articulaires via le **modèle géométrique inverse (MGI)** :

$$\mathbf{p}_d(t) \rightarrow \mathbf{q}_d(t) \rightarrow \text{PID} \rightarrow \mathbf{q}(t).$$

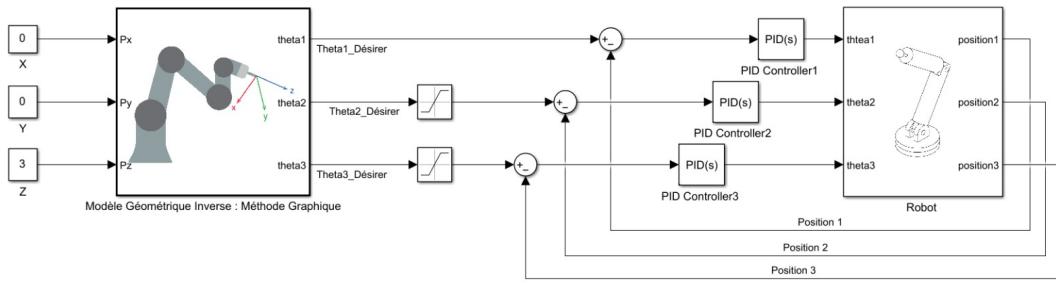


FIGURE 3.7 – Bloc MGI (méthode graphique) : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés ($\theta_{1d}, \theta_{2d}, \theta_{3d}$).

3.5.2 Bloc génération de trajectoire cartésienne

Ce bloc fournit les consignes cartésiennes (point à point, cercle, L, etc.) :

$$\mathbf{p}_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]^\top.$$

3.5.3 Bloc MGI : conversion cartésien → articulaire

Le MGI réalise la transformation :

$$(x_d, y_d, z_d) \mapsto (\theta_{1d}, \theta_{2d}, \theta_{3d}),$$

selon la méthode développée au Chapitre 2 (méthode graphique). Il peut intégrer :

- atteignabilité (workspace) ;
- limites articulaires ;
- choix cohérent de solution (éviter les sauts).

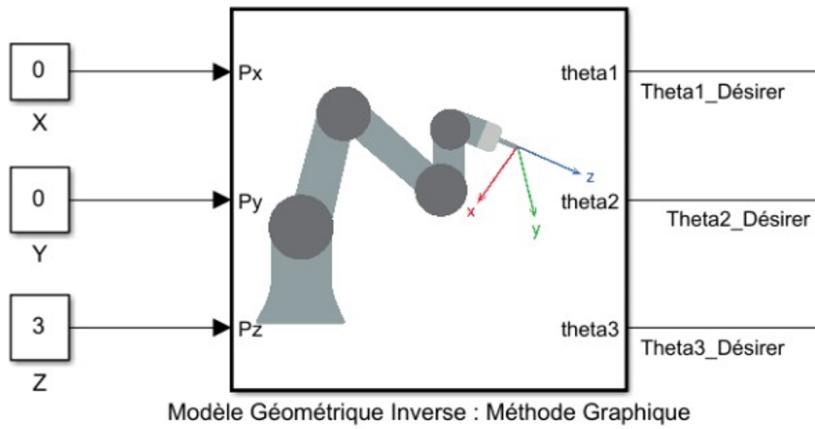


FIGURE 3.8 – Bloc MGI : conversion des consignes cartésiennes (P_x, P_y, P_z) en angles désirés ($\theta_{1d}, \theta_{2d}, \theta_{3d}$).

3.5.4 Boucles PID et robot

Les angles désirés $\mathbf{q}_d(t)$ sont comparés aux positions mesurées $\mathbf{q}(t)$, et les PID génèrent les commandes appliquées au robot Simscape.

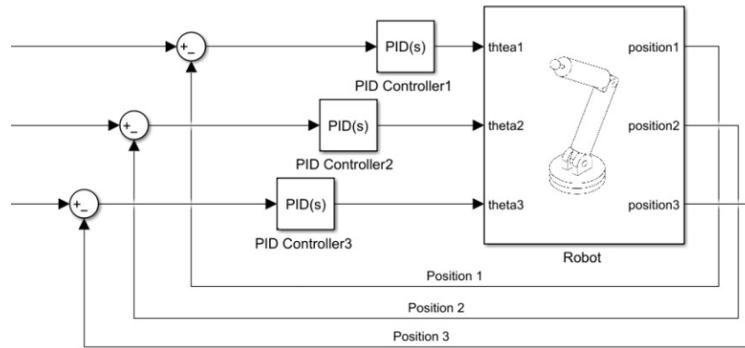


FIGURE 3.9 – Chaîne de commande basée sur la géométrie inverse MGI + multi-PID + robot.

3.6 Métriques de performance et fonction coût

Afin d'évaluer quantitativement chaque réglage PID, on calcule une fonction coût basée sur :

- erreur de suivi (ISE / IAE / ITAE) ;
- dépassement et temps de réponse ;
- effort de commande (norme/énergie des couples) ;
- pénalités en cas de saturation ou dépassement des limites articulaires.

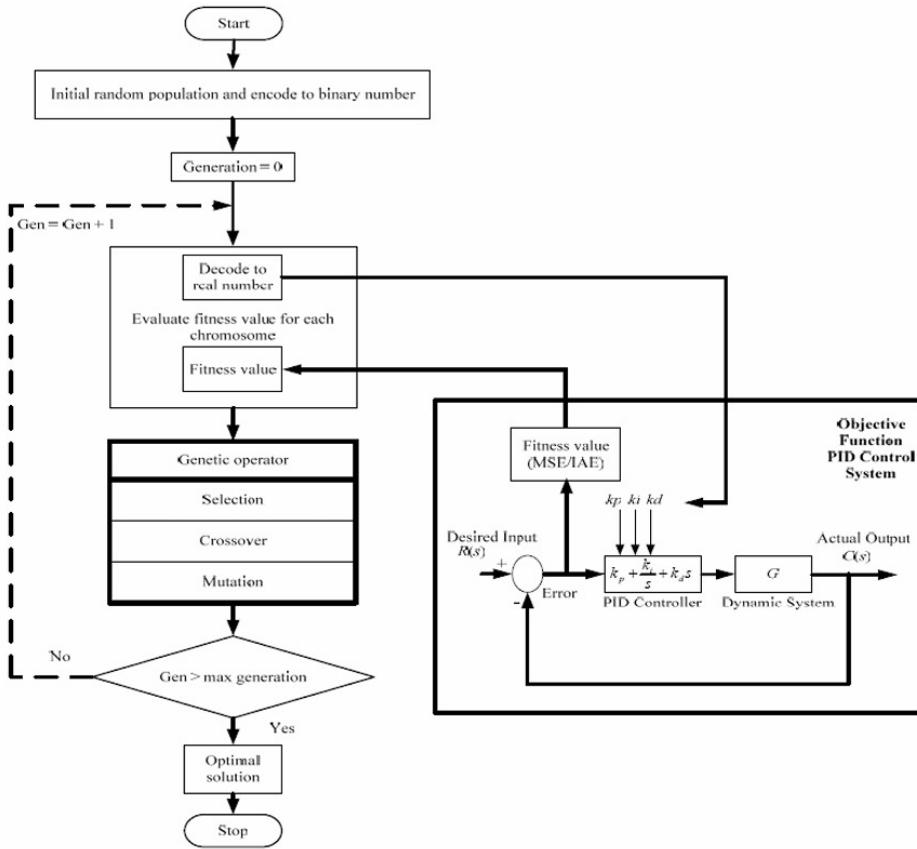


FIGURE 3.10 – Chaîne de commande basée DE

3.7 Intégration de l'algorithme DE dans la simulation

L'algorithme DE est utilisé pour optimiser les gains PID. Chaque individu de la population représente un vecteur de paramètres :

$$\mathbf{x} = [K_{p1}, K_{i1}, K_{d1}, K_{p2}, K_{i2}, K_{d2}, K_{p3}, K_{i3}, K_{d3}]^T.$$

Pour chaque individu :

1. les gains sont injectés dans les blocs PID ;
2. la simulation est exécutée sur une durée T ;
3. la fonction coût $J(\mathbf{x})$ est calculée à partir des erreurs et des contraintes ;
4. DE applique mutation, recombinaison et sélection afin d'améliorer J .

Cette boucle permet d'obtenir un réglage PID **automatique** et adapté au comportement du robot simulé.

3.7.1 Principe général

Chaque articulation est pilotée par un PID indépendant. Ainsi, l'optimisation concerne un ensemble de trois PID (un par articulation). Dans une version globale, un individu DE peut regrouper les neuf paramètres :

$$\mathbf{x} = [K_{p1}, K_{i1}, K_{d1}, K_{p2}, K_{i2}, K_{d2}, K_{p3}, K_{i3}, K_{d3}]^T.$$

Dans l'implémentation Simulink utilisée ici, le réglage est réalisé **en ligne** (*live*) via trois blocs **DE_Tuner_Live**, chacun fournissant (K_p, K_i, K_d) pour un PID.

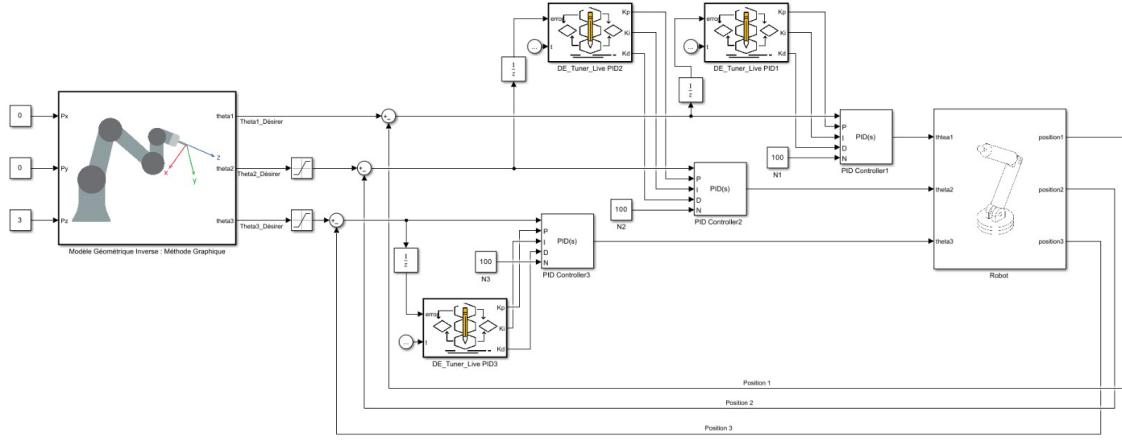


FIGURE 3.11 – Architecture Simulink incluant l'algorithme DE :adaptation en ligne des gains PID pour chaque articulation.

3.7.2 Fonction **DE_Tuner_Live** : réglage PID en ligne

Le bloc **DE_Tuner_Live** reçoit en entrée l'erreur $e(t)$ et le temps t , puis renvoie les gains (K_p, K_i, K_d). L'algorithme repose sur :

- une **population** de taille $NP = 10$ (choisie petite pour réduire le temps de calcul) ;
- une dimension $D = 3$ correspondant à (K_p, K_i, K_d) ;
- une **fonction coût** simplifiée en ligne : $J(t) = |e(t)|$ (plus l'erreur est grande, plus la fitness est mauvaise) ;
- une mise à jour périodique des individus (toutes les 0.1s) via mutation et croisement.

La mutation suit la forme classique de DE :

$$\mathbf{v} = \mathbf{x}_{r1} + F (\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad F = 0.5,$$

puis un croisement (probabilité typique $CR \simeq 0.8$) met à jour l'individu testé. Les gains sont contraints positifs (valeur absolue) afin d'éviter des réglages incohérents. Le meilleur individu (selon la fitness) est utilisé comme sortie du bloc.

3.7.3 Interprétation : comparaison des deux cas

Afin de mettre en évidence l'intérêt du DE, deux cas sont comparés :

Cas 1 : commande PID sans DE (gains fixes)

Dans ce cas, les PID utilisent des gains constants (réglage manuel ou initial). La réponse observée présente une dynamique moins favorable : présence de transitoires plus marqués (dépassement/oscillations) et un temps d'établissement plus long avant stabilisation.



FIGURE 3.12 – Réponse du système avant intégration du DE (PID à gains fixes) : transitoire plus important et stabilisation plus lente.

Cas 2 : commande PID avec DE (gains optimisés en ligne)

Après intégration du DE, les gains PID sont adaptés automatiquement en fonction de l'erreur. On observe une amélioration globale du suivi : réduction de l'erreur résiduelle, stabilisation plus rapide, et diminution des oscillations. Autrement dit, DE agit comme un mécanisme d'auto-réglage qui adapte K_p , K_i et K_d pour améliorer la qualité de la réponse.

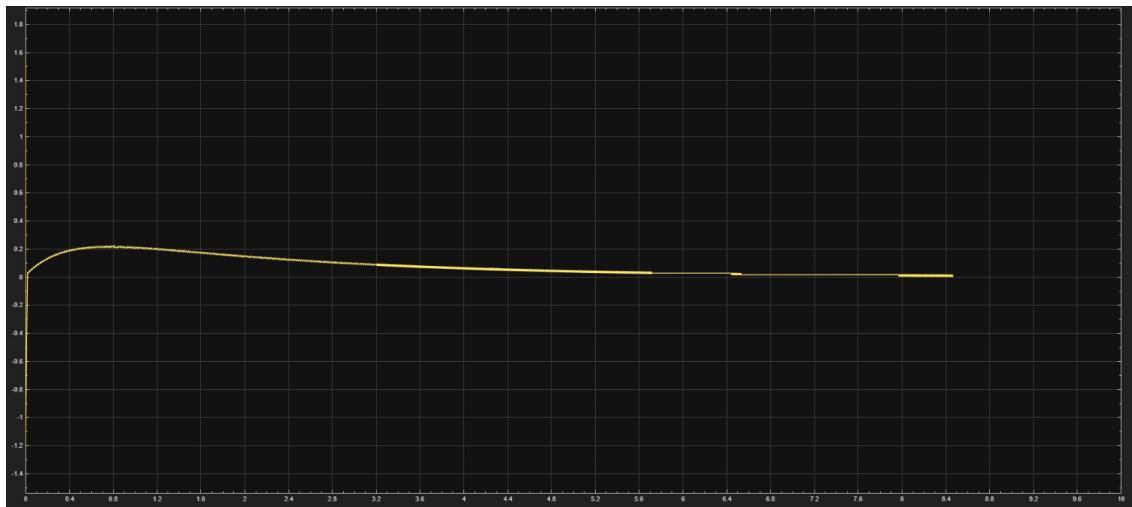


FIGURE 3.13 – Réponse du système après intégration du DE : amélioration de la stabilité et réduction de l'erreur de suivi.

3.8 Conclusion

Ce chapitre établit le cadre de simulation complet sous MATLAB/Simulink, intégrant la cinématique, la dynamique et la commande PID, ainsi que la boucle d'optimisation par DE. Le chapitre suivant présentera les résultats numériques, les graphiques et les comparaisons obtenues sur les différents scénarios de test.

Conclusion Générale

Références

1. Storn, R., & Price, K. (1997). *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization.
2. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*.
3. Åström, K. J., & Hägglund, T. (2006). *Advanced PID Control*.
4. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics : Modelling, Planning and Control*.

Chapitre A

Annexes

A.1 Code MATLAB/Simulink

Description succincte des fichiers principaux utilisés.

A.2 Données complémentaires

Tableaux supplémentaires et figures détaillées.

A.3 Détails mathématiques

Développements mathématiques complémentaires sur la cinématique inverse.