

Application du Machine Learning à la prédiction des  
paramètres optimaux des politiques de production d'un système  
manufacturier soumis à une demande périodique

par

Laurent Davy NDZIE ENAMA

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE  
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE  
M. Sc. A.

MONTRÉAL, LE 20 MARS 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Laurent Davy NDZIE ENAMA, 2026



© Tous droits réservés

Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Jean-Pierre KENNE, directeur de mémoire  
Département du génie mécanique, École de Technologie Supérieure

M. Ali GHARBI, codirecteur  
Département du génie des systèmes, École de Technologie Supérieure

M. Christian BELLEAU, président du jury  
Département du génie mécanique, École de Technologie Supérieure

M. Lucas HOF, membre du jury  
Département du génie mécanique, École de Technologie Supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 17 MARS 2026

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## REMERCIEMENTS

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de recherche, M. Jean-Pierre KENNE, ainsi qu'à mon codirecteur, M. Ali GHARBI, pour la qualité de leur encadrement tout au long de ce travail. Leurs conseils avisés, leur disponibilité, leur rigueur scientifique et leur soutien financier ont été déterminants dans la réalisation de ce mémoire. Leur confiance et leurs encouragements constants m'ont permis de mener ce projet à terme dans les meilleures conditions.

Je souhaite également remercier chaleureusement M. Vladimir Polotski pour ses conseils précieux, sa grande disponibilité et l'aide qu'il m'a apportée à différentes étapes de ce travail. Ses remarques pertinentes et ses échanges constructifs ont grandement contribué à l'amélioration de ce mémoire.

Je remercie l'École de technologie supérieure pour la qualité de l'environnement académique et scientifique qu'elle offre, ainsi que pour les ressources mises à disposition des étudiants, favorisant un cadre propice à la recherche et à l'apprentissage.

Je tiens à adresser une reconnaissance toute particulière à mes parents, M. et Mme ENAMA, mes véritables piliers. Leur amour inconditionnel, leurs prières, leurs conseils et leur soutien indéfectible ont toujours été une source de force et de motivation. Sans eux, rien de ce parcours n'aurait été possible.

Mes sincères remerciements vont également à mes frères et sœurs pour leur présence, leur patience et leur encouragement tout au long de mes études.

Je remercie enfin mes amis, ainsi que toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail, par leur soutien moral, leurs échanges ou leur bienveillance.



# Application du Machine Learning à la prédiction des paramètres optimaux des politiques de production d'un système manufacturier soumis à une demande périodique

Laurent Davy NDZIE ENAMA

## RÉSUMÉ

Les systèmes de production soumis à une demande variable et à des pannes et réparations aléatoires posent un défi majeur de pilotage, car les décisions optimales doivent s'adapter à une dynamique incertaine et non stationnaire. La programmation dynamique stochastique et les équations d'Hamilton–Jacobi–Bellman (HJB) constituent un outil rigoureux pour la détermination des politiques optimales de production. Toutefois, dans des contextes non-stationnaires, la résolution numérique de ces équations nécessite un temps de calcul important, ce qui rend difficile l'exploration de multiples scénarios dans un contexte industriel où la réactivité est essentielle.

Ce mémoire propose de recourir à l'apprentissage automatique (*Machine Learning*) pour prédire rapidement les paramètres des politiques de production optimales, à partir de solutions d'HJB préalablement calculées. Les solutions obtenues par résolution des équations d'Hamilton–Jacobi–Bellman servent de base de données pour entraîner des modèles d'apprentissage supervisé, qui apprennent la relation entre des paramètres du système et les caractéristique des politiques optimales. Une fois entraînés, ces modèles fournissent directement une approximation des paramètres optimaux, sans nécessiter de nouvelle résolution numérique des équations d'HJB, ce qui réduit considérablement le temps de calcul. Pour mettre en place et valider cette approche, une méthodologie progressive est développée sur deux systèmes de complexité différente, le second représentant une extension plus du premier.

Dans le cas d'un système constitué d'une machine produisant un type de pièce, plusieurs algorithmes sont comparés : l'algorithme des k-plus proches voisins (kNN), les réseaux de neurones profonds (DNN) et les réseaux de neurones à base radiale (RBF). L'espace d'entrée comprend d'abord les caractéristiques de la demande (moyenne, amplitude, fréquence), puis est étendu pour intégrer le coût de rupture de stock et le taux de réparation. Les résultats montrent que le kNN offre un excellent compromis entre précision et rapidité, reproduisant fidèlement les solutions issues des équations d'HJB tout en réduisant significativement le temps de calcul. Cela rend possible l'exploration rapide de plusieurs scénarios, permettant au décideur d'évaluer l'impact de différentes conditions opérationnelles sur la politique de production.

Pour valider la pertinence de cette approche sur un système plus complexe, l'étude est étendue à une machine produisant deux types de pièces. Cette configuration permet également d'analyser comment l'augmentation du nombre de variables d'entrée influence le choix de l'algorithme. Les résultats révèlent que le kNN devient moins performant lorsque la dimension de l'espace d'entrée augmente, tandis que les réseaux de neurones profonds maintiennent une bonne précision. Ces travaux confirment que l'apprentissage automatique constitue une alternative efficace aux résolutions répétées des équations d'HJB, permettant des prédictions quasi instantanées tout en guidant le choix de l'algorithme selon la complexité du système étudié.

## VIII

**Mots-clés:** Commande optimale, Demande périodique, Machine Learning, Systèmes manufacturiers, k-plus proches voisins, Réseaux de neurones

# **Prediction of Optimal Production Policy Parameters in Manufacturing Systems under Periodic Demand**

Laurent Davy NDZIE ENAMA

## **ABSTRACT**

Production systems subject to time-varying demand as well as random breakdowns and repairs pose a major control challenge, as optimal decisions must adapt to uncertain and non-stationary dynamics. Stochastic dynamic programming and Hamilton–Jacobi–Bellman (HJB) equations provide a rigorous framework for determining optimal production policies. However, in non-stationary settings, the numerical resolution of these equations entails significant computational effort, which makes the exploration of multiple scenarios difficult in industrial environments where responsiveness is essential.

This thesis proposes the use of Machine Learning to rapidly predict the parameters of optimal production policies based on previously computed HJB solutions. The solutions obtained from the resolution of the Hamilton–Jacobi–Bellman equations are used to build a training dataset for supervised learning models, which learn the relationship between system parameters and the characteristics of optimal policies. Once trained, these models directly provide approximations of the optimal parameters without requiring further numerical resolution of the HJB equations, thereby significantly reducing computation time. To implement and validate this approach, a progressive methodology is developed on two systems of different complexity, the second representing an extension of the first.

In the case of a system consisting of a single machine producing one type of part, several algorithms are compared : the k-nearest neighbors (kNN) algorithm, deep neural networks (DNN), and radial basis function (RBF) neural networks. The input space initially includes demand characteristics (mean, amplitude, and frequency) and is subsequently extended to incorporate the backlog cost and the repair rate. The results show that kNN offers an excellent trade-off between accuracy and computational efficiency, closely reproducing the solutions obtained from the HJB equations while significantly reducing computation time. This enables rapid exploration of multiple scenarios, allowing decision-makers to assess the impact of different operating conditions on the production policy.

To assess the relevance of this approach on a more complex system, the study is extended to a machine producing two types of parts. This configuration also makes it possible to analyze how the increase in the number of input variables influences the choice of the learning algorithm. The results reveal that kNN performance deteriorates as the dimensionality of the input space increases, whereas deep neural networks maintain good predictive accuracy. These findings confirm that Machine Learning constitutes an effective alternative to repeated numerical solutions of the HJB equations, enabling near-instantaneous predictions while guiding the choice of the most suitable algorithm according to the complexity of the system under study.

**Keywords:** Optimal control, Periodic demand, Machine Learning, Manufacturing systems, k-nearest neighbors, Neural networks

## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE .....	5
1.1 Introduction .....	5
1.2 Systèmes de production et politique de commande optimale .....	6
1.2.1 Systèmes de production et notion de politique de commande optimale ....	6
1.2.2 Commande optimale stochastique .....	8
1.3 Cadre théorique du Machine Learning .....	9
1.3.1 Types d'apprentissage en Machine Learning .....	9
1.3.2 Types de problèmes en Machine Learning .....	10
1.3.3 Étapes d'un processus d'apprentissage supervisé .....	11
1.4 Algorithmes de Machine Learning .....	13
1.4.1 Algorithme des $k$ -plus proches voisins (kNN) pour la régression .....	13
1.4.2 Réseaux de neurones artificiels et réseaux à base radiale (ANN/RBF) ....	15
1.4.3 Réseaux de neurones profonds (DNN) .....	16
1.4.4 Métriques de performance pour les modèles de régression .....	18
1.5 Revue critique de la littérature .....	19
1.5.1 Commande optimale des systèmes manufacturiers sous demande constante .....	19
1.5.1.1 Systèmes mono-produit .....	20
1.5.1.2 Systèmes multi-produits .....	21
1.5.2 Intégration de la variabilité de la demande .....	22
1.5.3 Approches par Machine Learning .....	23
1.5.3.1 Méthodes basées sur les distances (kNN) .....	24
1.5.3.2 Réseaux de neurones (ANN, RBF) .....	25
1.5.3.3 Apprentissage profond (DNN, Deep RL) .....	26
1.6 Synthèse, problématique, méthodologie et contributions .....	28
1.6.1 Synthèse critique de la littérature .....	28
1.6.2 Problématiques et objectifs de recherche .....	31
1.6.3 Méthodologie .....	32
1.6.4 Contributions .....	34
1.7 Conclusion .....	35
CHAPITRE 2 DÉVELOPPEMENT DE L'APPROCHE POUR UN SYSTÈME MANUFACTURIER PRODUISANT UN SEUL TYPE DE PIÈCE .....	37
2.1 Introduction .....	37
2.2 Modélisation du système et commande optimale stochastique .....	37
2.2.1 Diagramme des taux de transition .....	38
2.2.2 Probabilités limites et conditions de faisabilité .....	40
2.2.3 Conditions d'optimalité .....	41

2.2.4	Résolution numérique .....	43
2.2.5	Limites de la résolution numérique des équations d'HJB .....	47
2.3	Reformulation du problème de commande optimale pour l'apprentissage automatique .....	48
2.4	Méthodologie de prédiction des paramètres optimaux de la politique de production	49
2.5	Analyse comparative des algorithmes .....	52
2.5.1	Les réseaux de neurones profonds (DNN) .....	52
2.5.2	Réseaux de neurones à base radiale (RBF) .....	53
2.5.3	$k$ -Nearest Neighbors (kNN) .....	55
2.5.4	Analyses de sensibilité .....	57
2.5.4.1	Variation du taux moyen de demande .....	57
2.5.4.2	Variation de l'amplitude du taux de demande .....	59
2.5.4.3	Variation de la fréquence du taux de demande .....	61
2.5.5	Synthèse des résultats et gain computationnel .....	63
2.6	Considération de la variation du coût de rupture de stock et du taux de réparation ..	65
2.6.1	Analyse globale des prédictions .....	65
2.6.2	Validation sur scénarios représentatifs .....	67
2.6.2.1	Scénario particulier .....	68
2.6.2.2	Validation du modèle kNN .....	69
2.6.3	Analyses de sensibilité .....	71
2.6.3.1	Analyse de sensibilité par rapport au taux de réparation .....	71
2.6.3.2	Analyse de sensibilité par rapport au coût de rupture de stock ..	74
2.6.4	Implémentation de la politique optimale .....	77
2.7	Conclusion .....	80
CHAPITRE 3	EXTENSION DE L'APPROCHE À UN SYSTÈME MANUFACTURIER PRODUISANT DEUX TYPES DE PIÈCES .....	81
3.1	Introduction .....	81
3.2	Modélisation du système multi-produits et commande optimale stochastique .....	82
3.2.1	Diagramme des taux de transition .....	83
3.2.2	Probabilités limites et conditions de faisabilité .....	84
3.2.3	Conditions d'optimalité .....	85
3.2.4	Résolution numérique et politique optimale .....	86
3.2.4.1	Résolution numérique .....	86
3.2.4.2	Détermination du coût optimal .....	90
3.2.4.3	Politiques optimales de production .....	91
3.3	Prédiction des paramètres optimaux par Machine Learning .....	93
3.3.1	Reformulation du problème HJB en tâche d'apprentissage .....	93
3.3.2	Approche méthodologique .....	94
3.4	Résultats de la première expérience .....	97
3.4.1	Performances globales du kNN .....	98
3.4.2	Analyse de scénarios .....	101
3.4.2.1	Validation du modèle .....	102

3.5	Résultats de la deuxième expérience et transition vers les réseaux de neurones profonds .....	104
3.5.1	Comparaison des performances kNN et DNN .....	104
3.5.1.1	Performances du kNN .....	104
3.5.1.2	Performances du DNN .....	107
3.5.2	Analyse de scénarios et validation .....	110
3.5.2.1	Scénario particulier .....	111
3.5.2.2	Validation du modèle .....	113
3.5.3	Analyses de sensibilité .....	113
3.5.3.1	Analyse de sensibilité sur le coût de rupture de stock .....	114
3.5.3.2	Analyse de sensibilité par rapport au taux de réparation .....	116
3.5.4	Mise en œuvre des politiques de production optimales .....	119
3.5.5	Domaine de validité de l'approche .....	123
3.5.5.1	Domaine de validité du modèle d'apprentissage .....	123
3.5.5.2	Hypothèses de modélisation .....	124
3.5.5.3	Compromis de calcul et lien avec les contributions .....	124
3.5.6	Analyse et discussion .....	125
3.5.6.1	Synthèse comparative des algorithmes à travers les deux systèmes .....	125
3.5.6.2	Effet de la dimensionnalité et seuil de transition algorithmique .....	126
3.5.6.3	Positionnement par rapport à la littérature .....	127
3.6	Conclusion .....	128
	CONCLUSION GÉNÉRALE .....	129
	PERSPECTIVES .....	133
ANNEXE I	VALIDATION ÉTENDUE DU MODÈLE KNN POUR LE SYSTÈME MONO-PRODUIT (CHAPITRE 2) .....	135
ANNEXE II	VALIDATION ÉTENDUE DU MODÈLE KNN POUR LE SYSTÈME BI-PRODUITS — PHASE 1 (CHAPITRE 3) .....	139
ANNEXE III	VALIDATION ÉTENDUE DU MODÈLE DNN POUR LE SYSTÈME BI-PRODUITS — PHASE 2 (CHAPITRE 3) .....	143
	BIBLIOGRAPHIE .....	145



## LISTE DES TABLEAUX

	Page
Table 1.1	Synthèse des travaux de la littérature et positionnement de la contribution du mémoire ..... 30
Table 2.1	Paramètres numériques utilisés pour la Figure 2.3 ..... 45
Table 2.2	Paramètres utilisés pour la création de l'ensemble de données ..... 52
Table 2.3	Comparatif des performances des modèles ML (MAE pour $Z^*$ et erreur relative pour $C$ ) ..... 57
Table 2.4	Paramètres variables utilisés pour la génération de l'ensemble de données dans l'Expérience 2 ..... 65
Table 2.5	Performances du modèle kNN sur 10 scénarios : MAE sur $z^*$ , coûts optimaux HJB et prédicts, ainsi que l'erreur relative ..... 70
Table 3.1	Valeurs numériques des paramètres utilisés pour l'exemple ..... 91
Table 3.2	Paramètres numériques pour la Phase 1 ..... 98
Table 3.3	Performances du modèle kNN sur 10 scénarios représentatifs : MAE sur $z_1^*$ et $z_2^*$ , coûts optimaux HJB et prédicts, erreur relative ..... 103
Table 3.4	Paramètres numériques de la Phase 2 ..... 104
Table 3.5	Hyperparamètres et structure du modèle DNN ..... 108
Table 3.6	Performances du modèle DNN sur 10 scénarios représentatifs : MAE sur $z_1^*$ et $z_2^*$ , coûts HJB et prédicts, erreur relative ..... 113
Table 3.7	Synthèse des performances des algorithmes selon le système et le nombre de variables d'entrée ..... 125



## LISTE DES FIGURES

	Page
Figure 1.1	Schéma général d'un système de production avec une unité de production, plusieurs stocks et plusieurs marchés ..... 7
Figure 1.2	Schéma de synthèse d'un processus d'apprentissage supervisé ..... 12
Figure 1.3	Schéma illustratif du principe de prédiction de l'algorithme des $k$ -plus proches voisins (kNN) en régression ..... 14
Figure 1.4	Architecture schématique d'un réseau de neurones artificiels (ANN) pour un problème de régression ..... 15
Figure 1.5	Architecture schématique d'un réseau de neurones profond (DNN) pour un problème de régression ..... 17
Figure 2.1	Structure du système manufacturier ..... 38
Figure 2.2	Diagramme des taux de transition entre les deux états de la machine .... 39
Figure 2.3	Illustration de l'effet d'anticipation ..... 46
Figure 2.4	Représentation schématique de la méthodologie proposée ..... 50
Figure 2.5	Prédictions de $z^*$ par le modèle DNN ..... 53
Figure 2.6	Prédictions du coût optimal par le modèle DNN ..... 53
Figure 2.7	Prédictions de $z^*$ par le modèle RBF ..... 54
Figure 2.8	Prédictions du coût optimal par le modèle RBF ..... 55
Figure 2.9	Prédictions de $z^*$ par le modèle kNN ..... 56
Figure 2.10	Prédictions du coût optimal par le modèle kNN ..... 56
Figure 2.11	Trajectoire de $z^*(t)$ pour différents niveaux de $d_m$ ..... 58
Figure 2.12	Coût optimal en fonction de $d_m$ ..... 59
Figure 2.13	Trajectoire de $z^*(t)$ pour différents niveaux de $d_a$ ..... 60
Figure 2.14	Coût optimal en fonction de $d_a$ ..... 61
Figure 2.15	Trajectoire de $z^*(t)$ pour différentes valeurs de $\omega$ ..... 62

Figure 2.16	Coût optimal pour différentes valeurs de $\omega$ .....	63
Figure 2.17	Comparaison entre la moyenne réelle et prédite de $Z^*(t)$ (kNN, expérience 2) .....	66
Figure 2.18	Comparaison entre le coût optimal réel et prédit (kNN, expérience 2) ...	67
Figure 2.19	Comparaison des trajectoires optimales $Z^*(t)$ issues des équations HJB et du modèle kNN (scénario particulier .....	68
Figure 2.20	Trajectoire optimale $z^*(t)$ pour différents niveaux de $q_{21}$ .....	72
Figure 2.21	Coût optimal en fonction de $q_{21}$ : comparaison HJB / kNN .....	73
Figure 2.22	Erreur relative sur le coût optimal en fonction du taux de réparation $q_{21}$	74
Figure 2.23	Trajectoire optimale $z^*(t)$ pour différents niveaux de $c^-$ .....	75
Figure 2.24	Coût optimal en fonction de $c^-$ : comparaison HJB / kNN .....	76
Figure 2.25	Erreur relative sur le coût optimal en fonction du coût de rupture $c^-$ .....	77
Figure 2.26	Évolution de la trajectoire optimale $Z^*(t)$ .....	78
Figure 2.27	Arbre de décision appliqué à l'instant $t_4$ : le chemin suivi est indiqué en rouge .....	79
Figure 3.1	Structure du système étudié .....	82
Figure 3.2	Diagramme des taux de transition entre les deux états de la machine ....	83
Figure 3.3	Effet d'anticipation pour le produit 1 : évolution du seuil critique optimal $z_1^*(t)$ et du taux de demande $d_1(t)$ .....	92
Figure 3.4	Effet d'anticipation pour le produit 2 : évolution du seuil critique optimal $z_2^*(t)$ et du taux de demande $d_2(t)$ .....	92
Figure 3.5	Processus d'enrichissement du dataset pour la Phase 2 .....	95
Figure 3.6	Visualisation de la méthodologie proposée .....	97
Figure 3.7	Comparaison entre le coût optimal $C$ prédit par le modèle kNN et les valeurs de référence issues des HJB sur l'ensemble de test .....	99
Figure 3.8	Comparaison entre les valeurs de $z_1^*(t)$ prédites par le kNN et les valeurs de référence issues des HJB .....	100

Figure 3.9	Comparaison entre les valeurs de $z_2^*(t)$ prédites par le kNN et les valeurs de référence issues des HJB .....	100
Figure 3.10	Comparaison des trajectoires de $z_1^*(t)$ obtenues par HJB (référence) et prédites par le modèle kNN .....	101
Figure 3.11	Comparaison des trajectoires de $z_2^*(t)$ obtenues par HJB (référence) et prédites par le modèle kNN .....	102
Figure 3.12	Performances du modèle kNN ( $k = 7$ ) pour la prédiction du coût optimal $C$ sur l'ensemble de test .....	105
Figure 3.13	Performances du modèle kNN pour la prédiction de la trajectoire $z_1^*(t)$ sur l'ensemble de test .....	106
Figure 3.14	Performances du modèle kNN pour la prédiction de la trajectoire $z_2^*(t)$ sur l'ensemble de test .....	107
Figure 3.15	Performances du modèle DNN pour la prédiction de la trajectoire $z_1^*(t)$ sur l'ensemble de test .....	108
Figure 3.16	Performances du modèle DNN pour la prédiction de la trajectoire $z_2^*(t)$ sur l'ensemble de test .....	109
Figure 3.17	Performances du modèle DNN pour la prédiction du coût optimal $C$ sur l'ensemble de test .....	110
Figure 3.18	Scénario particulier : comparaison des trajectoires de $z_1^*(t)$ obtenues par HJB (référence) et prédites par le modèle DNN .....	111
Figure 3.19	Scénario particulier : comparaison des trajectoires de $z_2^*(t)$ obtenues par HJB (référence) et prédites par le modèle DNN .....	112
Figure 3.20	Sensibilité de la trajectoire $z_1^*(t)$ aux variations de $c^-$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN	114
Figure 3.21	Sensibilité de la trajectoire $z_2^*(t)$ aux variations de $c^-$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN	115
Figure 3.22	Sensibilité du coût optimal aux variations de $c^-$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN .....	116
Figure 3.23	Sensibilité de la trajectoire $z_1^*(t)$ aux variations de $q_{21}$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN	117

Figure 3.24	Sensibilité de la trajectoire $z_2^*(t)$ aux variations de $q_{21}$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN	118
Figure 3.25	Sensibilité du coût optimal aux variations de $q_{21}$ : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN	.....119
Figure 3.26	Évolution de la trajectoire optimale $z_1^*(t)$ prédite par le modèle DNN.	..120
Figure 3.27	Évolution de la trajectoire optimale $z_2^*(t)$ prédite par le modèle DNN	..121
Figure 3.28	Processus de décision pour la mise en œuvre de la politique de production optimale à $t = 600$	.....122

## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

DNN	Deep Neural Network
ERM	Erreur relative moyenne
HJB	Hamilton–Jacobi–Bellman
kNN	k-Nearest Neighbors
MAE	Mean Absolute Error
ML	Machine Learning
RBF	Radial Basis Function
RL	Reinforced Learning



## LISTE DES SYMBOLES ET UNITÉS DE MESURE

$\alpha(t)$	Mode de la machine au temps $t$
$\xi(t)$	Processus stochastique représentant l'état de la machine à l'instant $t$
$\pi_\alpha$	Probabilité stationnaire (limite) que la machine soit dans le mode $\alpha$
$c_i^+$	Coût unitaire de stockage du produit $i$
$c_i^-$	Coût unitaire de rupture du produit $i$
$d_i(t)$	Taux de demande du produit $i$
$d_{mi}$	Taux de demande moyen du produit $i$
$d_{ai}$	Amplitude du taux de demande du produit $i$
	$g(\cdot)$ : Coût instantané
$J(\cdot)$	Coût total actualisé associé à une politique de production
$N$	Nombre de points de discrétisation temporelle
$t_j$	$j$ -ième instant de discrétisation temporelle
$q_{\alpha\beta}$	Taux de transition de l'état $\alpha$ vers l'état $\beta$
$u_i(t)$	Taux de production du produit $i$
$U_{\max}^i$	Taux de production maximal du produit $i$
$v(\cdot)$	Fonction valeur associée au problème de commande optimale
$x_i(t)$	Niveau d'inventaire du produit $i$
$z_i^*(t)$	Seuil critique optimal du produit $i$
$\rho$	Taux d'actualisation
$\omega$	Fréquence de variation de la demande
$\varphi$	Phase initiale de la demande



## INTRODUCTION

Les systèmes manufacturiers contemporains évoluent dans des environnements de plus en plus contraints, marqués par une intensification de la concurrence, une exigence accrue de flexibilité et une variabilité croissante de la demande. Dans ce contexte, la performance des systèmes de production ne dépend plus uniquement de leur capacité nominale, mais également de leur aptitude à s'adapter efficacement à des dynamiques incertaines, tout en maîtrisant les coûts liés au stockage, aux ruptures et à la fiabilité des équipements. Ces enjeux sont au cœur des paradigmes de l'industrie 4.0 et de l'industrie 5.0, qui promeuvent des systèmes de production intelligents, résilients et orientés vers l'aide à la décision.

La commande optimale des systèmes de production constitue, depuis plusieurs décennies, un outil privilégié pour l'élaboration de politiques de pilotage performantes. En particulier, la programmation dynamique stochastique et la résolution des équations de Hamilton–Jacobi–Bellman permettent de déterminer des politiques optimales de production en tenant compte de l'état du système, des paramètres économiques et des phénomènes aléatoires tels que les pannes et réparations. Ces approches ont permis d'obtenir des résultats fondamentaux sur la structure des politiques optimales, notamment en régime stationnaire où la demande est supposée constante.

Toutefois, la considération d'une demande constante constitue une simplification importante au regard des réalités industrielles. Dans de nombreux secteurs, la demande présente des variations temporelles marquées, parfois de nature périodique, qui rendent le problème de contrôle non stationnaire (Polotski *et al.*, c; Abdelilah *et al.*, 2018). L'introduction explicite de cette variabilité complique considérablement la résolution des équations de Hamilton–Jacobi–Bellman, en raison de l'apparition de termes de dérivée temporelle et de la nécessité d'une discrétisation. Ces résolutions numériques deviennent alors rapidement coûteuses en temps de calcul, en particulier lorsque plusieurs paramètres du système doivent être explorés simultanément.

Cette complexité computationnelle rend difficile l'utilisation des approches classiques de commande optimale stochastique dès lors qu'une exploration rapide de scénarios ou des analyses de sensibilité sont nécessaires. Or, ces capacités sont devenues indispensables pour accompagner les décideurs industriels confrontés à des paramètres économiques et opérationnels en constante évolution. Il en résulte un écart entre la rigueur théorique de ces méthodes et leur capacité à être mobilisées de façon flexible et réactive pour l'analyse de scénarios dans les systèmes de production modernes.

Dans ce contexte, les méthodes d'apprentissage automatique ont suscité un intérêt croissant en tant qu'outils complémentaires aux approches analytiques classiques. En particulier, l'apprentissage supervisé offre la possibilité d'approximer des relations complexes à partir de données, ouvrant la voie à des prédictions rapides et à une exploitation en temps quasi réel. Toutefois, l'usage conjoint de ces méthodes et des approches de commande optimale demeure encore peu formalisé, en particulier pour des systèmes soumis à une demande variable dans le temps. De nombreux travaux mobilisent des algorithmes d'apprentissage sans établir de lien explicite avec des solutions optimales de référence, ou sans analyser l'adéquation du choix de l'algorithme d'apprentissage automatique à la complexité du système étudié.

Le présent mémoire aborde cette problématique en combinant commande optimale stochastique et apprentissage automatique, avec pour objectif de rendre les politiques optimales de production plus facilement exploitables dans des contextes industriels non stationnaires, où la demande varie de manière périodique. L'approche proposée s'appuie sur la rigueur des solutions issues de la résolution des équations d'Hamilton–Jacobi–Bellman, tout en permettant une utilisation rapide et flexible, adaptée aux exigences des systèmes de production intelligents.

Afin de mener à bien ce travail, ce mémoire est structuré en trois chapitres comme suit. Le chapitre un (1) présente d'abord les cadres conceptuels et les outils méthodologiques mobilisés dans la suite du document, en introduisant notamment les notions de commande

optimale stochastique, les grandes lignes des approches numériques de résolution des équations d'Hamilton–Jacobi–Bellman, ainsi que les principes de l'apprentissage automatique supervisé et des algorithmes considérés (algorithme des k-plus proches voisins, réseaux de neurones à base radiale et réseaux de neurones profonds). Il se poursuit par une revue critique de la littérature consacrée aux travaux portant sur la commande optimale des systèmes de production ainsi que ceux portant sur l'application du Machine Learning, et se conclut par une synthèse critique et par le positionnement du travail proposé.

Le chapitre deux (2) est consacré à l'étude d'un système de production constitué d'une machine produisant un type de pièce, soumis à des pannes et réparations aléatoires et à une demande variable périodique. La méthodologie développée y est détaillée et appliquée afin d'analyser la capacité des modèles d'apprentissage à approximer les politiques optimales de production dans ce cadre mono-produit.

Le chapitre trois (3) étend cette méthodologie à un système constitué d'une machine produisant deux types de pièces. Il permet d'abord de valider la capacité de l'approche par apprentissage automatique à se généraliser à un système manufacturier plus complexe, puis d'analyser l'effet de l'augmentation du nombre de variables d'entrée du modèle sur le comportement des algorithmes d'apprentissage et sur le choix de l'algorithme le plus adapté. Et enfin, une partie est consacrée aux pistes de recherche et aux applications futures.



# CHAPITRE 1

## REVUE DE LA LITTÉRATURE

### 1.1 Introduction

La commande optimale des systèmes de production constitue un champ de recherche central pour l'aide à la décision industrielle, en particulier lorsque les systèmes sont soumis à des pannes aléatoires, des coûts économiques contradictoires et des évolutions de la demande dans le temps. Les approches fondées sur la programmation dynamique stochastique et la résolution des équations de Hamilton–Jacobi–Bellman offrent un cadre théorique rigoureux pour déterminer des politiques de production optimales, mais leur mise en œuvre devient rapidement complexe dès que la dimension du système augmente ou que la demande présente une dynamique non stationnaire.

Dans ce contexte, ce chapitre établit les fondements conceptuels et méthodologiques nécessaires à la suite du mémoire. Il introduit d'abord les notions essentielles liées aux systèmes de production, au contrôle optimal stochastique et aux politiques optimales issues des équations HJB. Il présente ensuite les principes de l'apprentissage automatique supervisé et les algorithmes considérés dans ce travail, en mettant l'accent sur leur usage pour des problèmes de régression.

Le cœur du chapitre est consacré à une revue critique de la littérature, structurée autour de trois axes complémentaires : la commande optimale stochastique des systèmes manufacturiers sous demande constante, l'intégration explicite de la variabilité temporelle de la demande et l'émergence des approches par Machine Learning pour contourner les limitations computationnelles des méthodes classiques. Cette analyse met en évidence un écart persistant entre la rigueur des solutions issues du contrôle optimal et leur exploitabilité pratique dans des contextes industriels dynamiques.

Enfin la synthèse critique de la littérature, la formulation des problématiques de recherche, la présentation des objectifs du mémoire, la méthodologie proposée et les contributions, qui

ouvrent la voie aux développements méthodologiques et aux études de cas présentées dans les chapitres suivants.

## **1.2 Systèmes de production et politique de commande optimale**

Cette section a pour objectif de poser le cadre de modélisation et de commande optimale dans lequel s'inscrit l'ensemble des travaux présentés dans ce mémoire. Elle introduit les éléments nécessaires à la compréhension des systèmes de production considérés, ainsi que la notion de politique optimale de contrôle qui constitue le point de départ des approches développées par la suite.

Dans un premier temps, une représentation générique des systèmes de production est présentée afin de décrire les principaux flux physiques et décisionnels, ainsi que les variables d'état et de contrôle associées. Cette modélisation permet de définir clairement le cadre des politiques de production étudiées et de situer les configurations particulières analysées dans la suite du mémoire. Dans un second temps, le cadre du contrôle optimal stochastique est introduit, ainsi que les principes de résolution numérique utilisés pour obtenir des politiques optimales de référence. Ces solutions constituent la base sur laquelle reposent les approches d'apprentissage automatique développées dans les chapitres suivants.

### **1.2.1 Systèmes de production et notion de politique de commande optimale**

Dans ce mémoire, les systèmes de production considérés sont représentés par le schéma général présenté à la Figure 1.1. Ce schéma met en évidence les principaux flux physiques et décisionnels du système, indépendamment du nombre de types de pièces effectivement traités.

Le système, soumis à des pannes et réparations aléatoires, est alimenté en matières premières, qui sont transformées par une unité de production centrale. Cette unité de production représente l'ensemble des ressources productives nécessaires à la fabrication des pièces, et peut être assimilée, selon le contexte étudié, à une machine unique ou à un ensemble de moyens de

production agrégés. Les décisions de production portent sur la manière dont cette unité est exploitée au cours du temps afin d'alimenter différents flux de sortie.

Les produits fabriqués sont dirigés vers plusieurs stocks, notés de manière générique  $x_1, x_2, \dots, x_n$ , chacun étant associé à un type de pièce distinct. Ces stocks permettent d'assurer la satisfaction des différentes demandes associées à chaque type de pièce. Chaque stock est ainsi relié à un marché spécifique, caractérisé par une demande propre, notée  $d_1, d_2, \dots, d_n$ .

Une politique de commande est définie comme une règle de décision associant à chaque état du système une action de production admissible (Bensoussan, 1982; Sethi *et al.*, 2005). L'objectif est de déterminer une politique de commande optimale qui minimise un critère de performance global, intégrant les coûts liés aux niveaux de stock, aux ruptures ou retards de livraison, ainsi que les effets des aléas de fonctionnement du système (Akella *et al.*, 1986; Gharbi *et al.*, 2000). Cette politique vise à assurer un compromis optimal entre satisfaction de la demande et maîtrise des coûts dans un environnement incertain et évolutif.

La représentation générique présentée ici permet de couvrir un large éventail de configurations. Dans la suite de ce mémoire, l'analyse se concentre plus particulièrement sur deux cas spécifiques : un système constitué d'une machine produisant un seul type de pièce, puis un système constitué d'une machine produisant deux types de pièces.

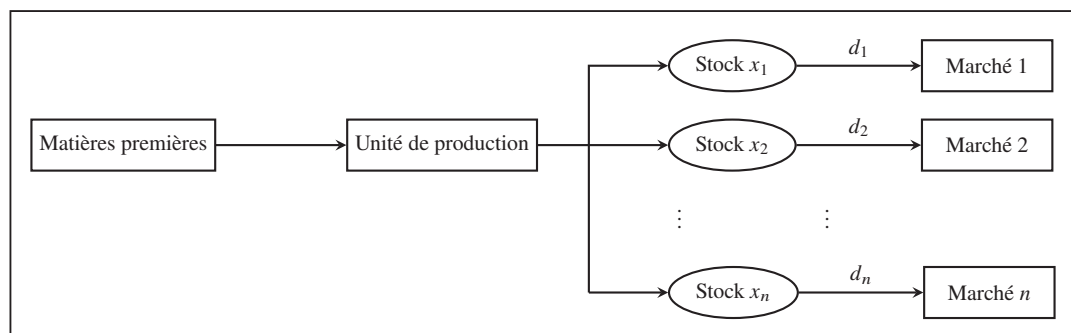


Figure 1.1 Schéma général d'un système de production avec une unité de production, plusieurs stocks et plusieurs marchés

### 1.2.2 Commande optimale stochastique

Les systèmes de production étudiés dans ce mémoire sont soumis à des sources d'incertitude liées notamment aux pannes et réparations aléatoires des équipements. Ces aléas conduisent à modéliser l'évolution du système dans un cadre stochastique, dans lequel la performance d'une politique de commande est évaluée en termes de coût espéré sur un horizon de temps donné (Bensoussan, 1982; Sethi *et al.*, 2005).

La programmation dynamique stochastique offre un outil privilégié pour traiter ce type de problèmes. Elle s'appuie sur le principe d'optimalité de Bellman, selon lequel une politique est optimale si, à partir de tout état atteint, les décisions futures restent elles-mêmes optimales. Ce principe conduit à l'introduction de la fonction valeur, qui représente le coût minimal associé à un état donné du système et permet de caractériser les politiques optimales de contrôle (Bellman, 1957).

L'application de la programmation dynamique conduit aux équations de Hamilton–Jacobi–Bellman (HJB). Ces équations permettent de déterminer les politiques de production optimales en fonction de l'état du système et des paramètres considérés (Sethi *et al.*, 2005). En pratique, l'obtention des politiques optimales repose sur des méthodes numériques. Dans le cas où tous les paramètres caractéristiques du système (taux de panne et de réparation, taux de demande, coûts de stockage et de rupture de stock, etc.) sont constants dans le temps, on considère que le système évolue en régime stationnaire. Les équations de Hamilton–Jacobi–Bellman se réduisent alors à un problème stationnaire, qui peut être résolu numériquement à l'aide de l'algorithme de Kushner (Kushner *et al.*, 1992; Boukas *et al.*, 1998). L'espace continu des niveaux de stock est discrétisé en un maillage, et les fonctions valeur sont mises à jour itérativement jusqu'à convergence, ce qui permet de déterminer les politiques optimales.

Lorsque certains paramètres deviennent explicitement dépendants du temps, par exemple un taux de demande périodiquement variable, les équations de Hamilton–Jacobi–Bellman deviennent non stationnaires et comportent un terme de dérivée temporelle de la fonction valeur. Une stratégie consiste alors à approcher ce problème par une succession de d'équations d'HJB

stationnaires pour des niveaux de demande « gelés », puis à estimer a posteriori les dérivées temporelles à partir de différences finies entre fonctions valeur associées à des demandes voisines, avant de recomputer les HJB avec ces termes correctifs (Polotski *et al.*, 2019).

Dans ce contexte, les solutions obtenues par la résolution numérique des équations HJB sont utilisées dans ce mémoire comme des solutions de référence. Elles servent de base à la construction des jeux de données nécessaires à l'apprentissage de modèles prédictifs capables d'approximer les paramètres des politiques optimales, tout en réduisant significativement le coût computationnel associé à leur exploitation.

### 1.3 Cadre théorique du Machine Learning

Le Machine Learning (ML) ou apprentissage machine constitue un champ de recherche majeur de l'intelligence artificielle, dont l'objectif est de concevoir des modèles capables d'apprendre automatiquement des relations à partir de données, sans être explicitement programmés pour chaque situation (Mitchell, 1997; Hastie *et al.*, 2009). Contrairement aux approches analytiques classiques, le ML repose sur l'exploitation de jeux de données afin d'approximer des fonctions d'intérêt, d'identifier des structures sous-jacentes ou de prédire des sorties associées à de nouvelles observations.

#### 1.3.1 Types d'apprentissage en Machine Learning

Les principaux types d'apprentissage utilisés en Machine Learning peuvent être résumés comme suit :

- **Apprentissage supervisé** : utilise des données annotées, sous forme de couples entrée–sortie. L'objectif est d'apprendre une fonction reliant les variables d'entrée aux variables de sortie, de manière à généraliser correctement cette relation à de nouvelles données non observées (Mitchell, 1997). Ce cadre est largement utilisé dans les problèmes de prédiction, d'estimation de paramètres et d'approximation de fonctions.

- **Apprentissage non supervisé** : repose sur des données dépourvues de labels. Il vise principalement à extraire des structures latentes, telles que des regroupements d'observations ou des représentations de dimension réduite, sans référence explicite à une sortie cible (Hastie *et al.*, 2009).
- **Apprentissage semi-supervisé** : se situe entre les deux précédents en combinant un nombre limité de données annotées avec un ensemble plus large de données non annotées. Cette approche permet d'améliorer les performances d'apprentissage lorsque l'obtention de labels est coûteuse ou difficile, tout en exploitant l'information contenue dans la structure globale des données (Zhu, 2009).

Dans ce travail, les données disponibles associent explicitement des paramètres d'entrée à des sorties de référence issues de méthodes de contrôle optimal. Nous utiliserons donc l'apprentissage supervisé.

### 1.3.2 Types de problèmes en Machine Learning

Dans le cadre de l'apprentissage supervisé, les problèmes traités en Machine Learning sont généralement classés en deux grandes catégories, selon la nature des sorties à prédire : les problèmes de classification et les problèmes de régression (Hastie *et al.*, 2009; Mitchell, 1997).

- **Problèmes de classification** : il s'agit d'attribuer à chaque observation une étiquette appartenant à un ensemble fini de classes. L'objectif est d'apprendre une règle de décision permettant de séparer ou de discriminer ces classes à partir des variables d'entrée. Ce type de problème est couramment rencontré dans des contextes tels que la reconnaissance de formes, la détection d'événements ou le diagnostic automatique (Bishop, 2006).
- **Problèmes de régression** : ils visent à prédire une variable de sortie continue à partir des variables d'entrée. L'objectif est d'approximer une fonction réelle reliant les entrées aux sorties, de manière à fournir des estimations quantitatives précises pour de nouvelles données (Hastie *et al.*, 2009). Les problèmes de régression sont particulièrement adaptés lorsque

les sorties d'intérêt correspondent à des paramètres, des coûts ou des grandeurs physiques continues.

Dans ce mémoire, le problème étudié relève explicitement de la régression. Les modèles de Machine Learning sont en effet mobilisés pour approximer des relations continues entre des paramètres d'entrée et des sorties de référence issues de méthodes de contrôle optimal.

### 1.3.3 Étapes d'un processus d'apprentissage supervisé

Un processus d'apprentissage supervisé suit généralement une séquence d'étapes bien définies, visant à construire un modèle capable de généraliser correctement à partir de données annotées (Hastie *et al.*, 2009; Goodfellow *et al.*, 2016). Bien que les implémentations puissent varier d'un algorithme à l'autre, ces étapes constituent un cadre méthodologique commun largement admis dans la littérature.

De manière synthétique, les principales étapes sont les suivantes (Bishop, 2006; Arlot *et al.*, 2010) :

- **Constitution du jeu de données** : rassemblement d'un ensemble de couples entrée–sortie, dans lequel les variables d'entrée décrivent les caractéristiques du problème étudié, tandis que les sorties correspondent à des valeurs de référence. La qualité, la représentativité et la couverture de ce jeu de données conditionnent fortement les performances du modèle appris.
- **Séparation en sous-ensembles** : division du jeu de données en au moins deux parties distinctes :
  - un *ensemble d'entraînement*, utilisé pour ajuster les paramètres du modèle ;
  - un *ensemble de test*, réservé à l'évaluation des performances sur des données non vues durant l'apprentissage.

Dans certains cas, un *ensemble de validation* supplémentaire est introduit pour guider le choix des paramètres du modèle et améliorer leur performance.

- **Apprentissage du modèle** : ajustement des paramètres du modèle en minimisant une fonction de coût mesurant l'écart entre les sorties prédites et les valeurs de référence sur l'ensemble d'entraînement. L'objectif est de trouver un compromis entre fidélité aux données d'apprentissage et capacité de généralisation.
- **Validation croisée** : répétition du processus d'entraînement et d'évaluation sur plusieurs partitions du jeu de données, afin d'obtenir une estimation plus robuste des performances et de réduire la sensibilité aux choix de séparation (Arlot *et al.*, 2010).
- **Évaluation des performances** : quantification de la qualité des prédictions du modèle à l'aide de métriques adaptées à la nature du problème considéré. Pour les problèmes de régression, ces métriques reposent généralement sur des mesures d'erreur entre les valeurs prédites et les valeurs de référence, telles que l'erreur absolue ou l'erreur relative, ce qui permet d'apprécier la précision du modèle et sa pertinence pour une utilisation en aide à la décision.

Les principales étapes de ce processus sont synthétisées à la Figure 1.2, qui illustre la logique générale reliant la constitution du jeu de données, l'apprentissage du modèle et l'évaluation de ses performances, indépendamment des algorithmes spécifiques considérés.

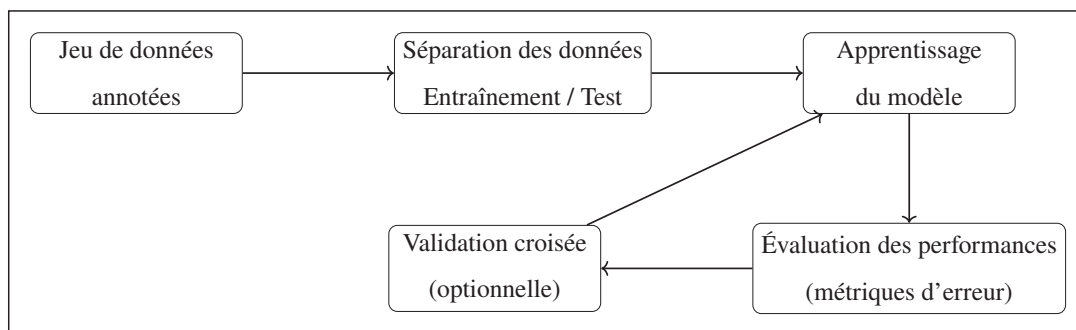


Figure 1.2 Schéma de synthèse d'un processus d'apprentissage supervisé

## 1.4 Algorithmes de Machine Learning

Dans le cadre de l'apprentissage supervisé appliqué à des problèmes de régression, plusieurs familles d'algorithmes peuvent être mobilisées afin d'approximer la relation entre les variables d'entrée et les sorties de référence. Ce mémoire s'intéresse en particulier à trois approches largement documentées dans la littérature : l'algorithme des  $k$ -plus proches voisins (kNN), les réseaux de neurones artificiels incluant les réseaux à base radiale (ANN/RBF), et les réseaux de neurones profonds (DNN) (Delpla *et al.*, 2024; Behnamfar *et al.*, 2022b).

Ces algorithmes reposent sur des principes d'approximation distincts et font intervenir des mécanismes d'apprentissage différents. Dans tous les cas, leur comportement est fortement conditionné par le choix d'un certain nombre *d'hyperparamètres*. Par hyperparamètres, on entend des paramètres fixés avant l'apprentissage (tels que la taille du modèle, le nombre de voisins ou le taux d'apprentissage), qui ne sont pas directement ajustés par l'algorithme à partir des données, mais qui contrôlent la complexité effective du modèle, la dynamique d'optimisation et sa capacité de généralisation.

### 1.4.1 Algorithme des $k$ -plus proches voisins (kNN) pour la régression

L'algorithme des  $k$ -plus proches voisins (kNN) est une méthode d'apprentissage supervisé non paramétrique, applicable aussi bien aux problèmes de classification qu'aux problèmes de régression (Cover *et al.*, 1967; Hastie *et al.*, 2009). Dans le cadre de ce mémoire, kNN est utilisé exclusivement dans un contexte de régression, où l'objectif est de prédire une variable de sortie continue à partir des variables d'entrée.

Le principe de la régression par kNN repose sur l'hypothèse selon laquelle des observations proches dans l'espace des variables d'entrée sont associées à des valeurs de sortie similaires. Contrairement aux modèles paramétriques, kNN ne construit pas explicitement de fonction globale lors de la phase d'apprentissage. L'ensemble des données d'entraînement est conservé, et la prédiction est réalisée localement au moment de l'interrogation du modèle.

La Figure 1.3 illustre le mécanisme de prédiction du kNN en régression. Pour une nouvelle observation, appelée *point requête*, les  $k$  observations les plus proches dans l'espace des entrées sont identifiées à partir d'une mesure de distance donnée. La valeur prédite est ensuite obtenue en combinant les valeurs de sortie associées à ces voisins, généralement par une moyenne arithmétique ou une moyenne pondérée par la distance.

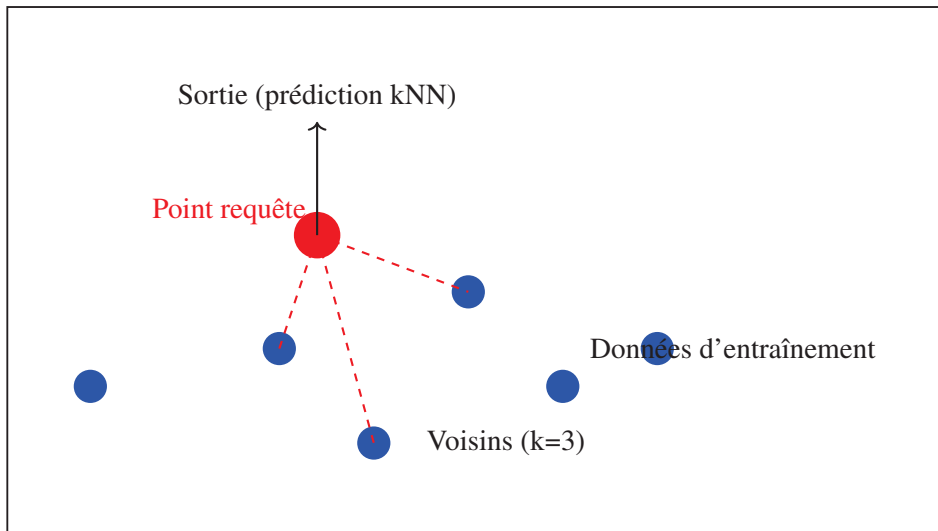


Figure 1.3 Schéma illustratif du principe de prédiction de l'algorithme des  $k$ -plus proches voisins (kNN) en régression

Dans le cas du kNN, les principaux hyperparamètres sont :

- le *nombre de voisins*  $k$ , qui contrôle le compromis entre biais et variance ;
- la *mesure de distance* utilisée pour définir la proximité entre les observations (par exemple distance euclidienne ou distance de Minkowski) ;
- le *schéma de pondération* des voisins, pouvant accorder un poids uniforme ou décroissant avec la distance.

Un aspect fondamental du kNN concerne la dimension de l'espace des variables d'entrée. Lorsque cette dimension augmente, la notion de proximité devient moins discriminante, phénomène souvent désigné sous le terme de *malédiction de la dimension* (Hastie *et al.*, 2009). En effet, dans des espaces de données d'entrées de grande dimension, les distances entre les observations

tendent à devenir homogènes, ce qui dégrade la pertinence de la sélection des voisins et peut nuire aux performances de prédiction. Ce point doit être pris en compte lors de l'application de kNN, en particulier dans les problèmes comportant un nombre élevé de variables d'entrée.

### 1.4.2 Réseaux de neurones artificiels et réseaux à base radiale (ANN/RBF)

Les réseaux de neurones artificiels (ANN) constituent une famille de modèles d'approximation non linéaire reposant sur l'interconnexion de neurones élémentaires organisés en couches successives (Bishop, 2006). Dans le contexte de l'apprentissage supervisé pour la régression, les réseaux de type perceptron multicouche sont couramment utilisés afin d'approximer une fonction continue reliant les variables d'entrée aux variables de sortie.

La Figure 1.4 présente une architecture schématique simple d'un réseau de neurones artificiels adapté à un problème de régression. Le réseau est composé d'une couche d'entrée, qui reçoit les différentes composantes du vecteur d'entrée, d'une couche cachée, constituée de neurones appliquant chacun une transformation affine suivie d'une fonction d'activation non linéaire, et d'une couche de sortie, qui produit la valeur prédite. Les connexions pondérées entre les neurones permettent de modéliser des relations non linéaires entre les variables, les poids de ces connexions étant ajustés lors de la phase d'apprentissage.

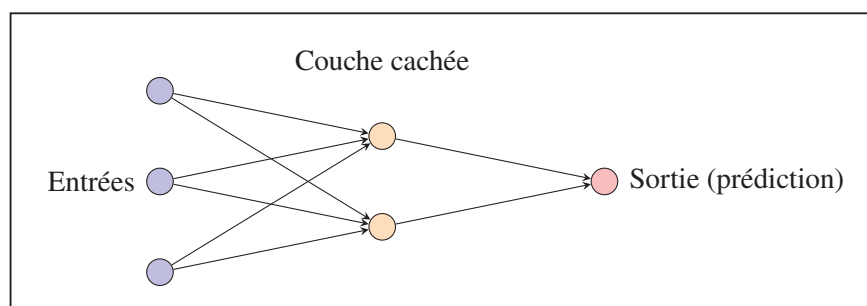


Figure 1.4 Architecture schématique d'un réseau de neurones artificiels (ANN) pour un problème de régression

Dans le cas des ANN, les hyperparamètres peuvent être regroupés en deux grandes catégories :

- des *hyperparamètres architecturaux*, qui définissent la structure du réseau :

- le nombre de couches cachées ;
- le nombre de neurones dans chaque couche cachée ;
- le choix des fonctions d'activation (par exemple sigmoïde, tangente hyperbolique ou ReLU) ;
- des *hyperparamètres d'apprentissage*, qui régissent la manière dont les poids sont ajustés :
  - le taux d'apprentissage ;
  - le nombre d'époques d'entraînement ;
  - la taille des mini-lots ;
  - les paramètres de régularisation, tels que les coefficients de pénalisation des poids ou la probabilité d'abandon (*dropout*) (Goodfellow *et al.*, 2016).

Le choix de ces hyperparamètres influe directement sur la capacité de représentation du modèle, la stabilité de l'optimisation et la tendance éventuelle au surapprentissage.

Les réseaux à base radiale (RBF) constituent une variante particulière des réseaux de neurones, dans laquelle la couche cachée est composée de neurones appliquant des fonctions de base radiales, le plus souvent de type gaussien (Broomhead *et al.*, 1988). La sortie du réseau est obtenue comme une combinaison linéaire des réponses de ces fonctions de base. Dans ce cas, les hyperparamètres spécifiques incluent :

- le *nombre de fonctions de base* (ou neurones radiaux) ;
- la *position des centres* des fonctions de base dans l'espace des entrées ;
- la *largeur* (ou échelle) associée à chaque fonction radiale.

Ces hyperparamètres déterminent la capacité du réseau RBF à modéliser des variations locales ou plus globales de la fonction cible, en fonction du nombre de neurones radiaux et de l'étendue de leur zone d'influence.

### 1.4.3 Réseaux de neurones profonds (DNN)

Les réseaux de neurones profonds (DNN) étendent le principe des réseaux de neurones artificiels en augmentant le nombre de couches cachées, ce qui permet de construire des représentations

hiérarchiques à partir des données d'entrée (Goodfellow *et al.*, 2016; LeCun *et al.*, 2015). Dans un cadre d'apprentissage supervisé pour la régression, cette profondeur accrue confère aux DNN une capacité élevée à approximer des relations non linéaires complexes entre les variables d'entrée et les variables de sortie.

La Figure 1.5 présente une architecture schématique d'un réseau de neurones profond, caractérisée par la présence de plusieurs couches cachées successives situées entre la couche d'entrée et la couche de sortie. Chaque couche cachée transforme la représentation issue de la couche précédente au moyen de connexions pondérées et de fonctions d'activation non linéaires. Cette composition de transformations successives permet au réseau de capturer progressivement des structures de plus en plus abstraites dans les données.

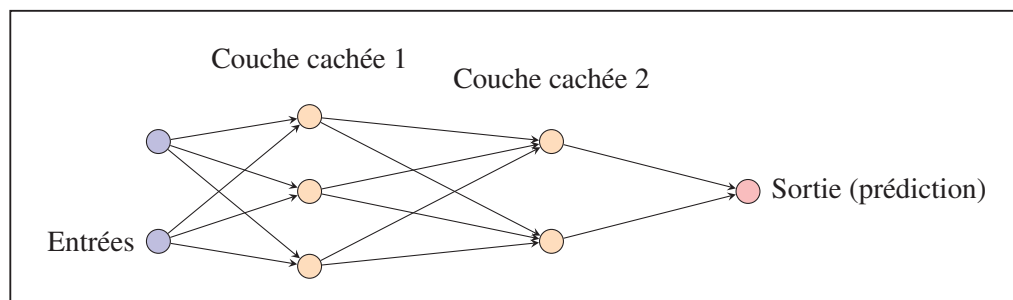


Figure 1.5 Architecture schématique d'un réseau de neurones profond (DNN) pour un problème de régression

Comme pour les réseaux de neurones artificiels classiques, le comportement et les performances des DNN dépendent fortement du choix des hyperparamètres. En raison de la profondeur du réseau, leur réglage revêt toutefois une importance accrue. Les principaux hyperparamètres peuvent être regroupés comme suit :

- des *hyperparamètres architecturaux*, tels que le nombre de couches cachées, le nombre de neurones par couche et le choix des fonctions d'activation ;
- des *hyperparamètres d'optimisation*, tels que le taux d'apprentissage, le choix de l'algorithme d'optimisation (par exemple descente de gradient stochastique ou méthodes adaptatives de type Adam), la taille des mini-lots et le nombre d'époques d'entraînement ;

- des *hyperparamètres de régularisation*, tels que la probabilité d'abandon (*dropout*), la normalisation de lots (*batch normalization*) ou les coefficients de pénalisation des poids.

Le choix de ces hyperparamètres conditionne la stabilité de l'apprentissage, la vitesse de convergence et la capacité du modèle à généraliser à de nouvelles données.

#### 1.4.4 Métriques de performance pour les modèles de régression

Dans le cadre de ce mémoire, qui porte sur des problèmes de régression, l'évaluation des modèles de Machine Learning repose sur des métriques de performance quantitatives permettant de mesurer la qualité des prédictions produites. Ces métriques jouent un rôle central pour comparer différents algorithmes et pour apprécier la pertinence d'un modèle dans un contexte donné (Hastie *et al.*, 2009; Bishop, 2006). Plus précisément, deux indicateurs sont retenus : l'erreur absolue moyenne et l'erreur relative moyenne.

- **Erreur absolue moyenne (MAE, Mean Absolute Error)**

La MAE mesure l'erreur moyenne, en valeur absolue, entre les prédictions et les valeurs de référence. Elle est définie par :

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad (1.1)$$

où  $N$  désigne le nombre de prédictions,  $\hat{y}_i$  la  $i$ -ème valeur prédite et  $y_i$  la  $i$ -ème valeur de référence. Plus la MAE est faible, plus les prédictions sont proches des valeurs réelles. Cette métrique pondère de la même manière les petites et les grandes erreurs et s'exprime dans la même unité que la grandeur prédite.

- **Erreur relative moyenne (ERM)**

L'erreur relative moyenne permet de mesurer, en termes relatifs, l'écart entre les prédictions et les valeurs de référence. Elle est définie par :

$$\text{ERM} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right|. \quad (1.2)$$

Cette métrique exprime l'erreur en proportion de la valeur réelle et peut être interprétée comme un pourcentage moyen d'erreur. Elle facilite la comparaison des performances du modèle sur des plages de valeurs différentes, en tenant compte de l'ordre de grandeur des quantités prédites.

Dans la suite de ce mémoire, ces métriques seront systématiquement utilisées pour évaluer et comparer les modèles d'apprentissage supervisé considérés et leurs performances.

La section suivante propose une revue critique de la littérature, en examinant comment ces outils ont été mobilisés dans les travaux existants et en mettant en évidence les limites qui motivent l'approche développée dans ce mémoire.

## **1.5 Revue critique de la littérature**

Cette section propose une revue critique des travaux sur le contrôle optimal des systèmes manufacturiers soumis à des pannes et réparations aléatoires. Elle examine d'abord les modèles sous demande constante, puis les approches intégrant une demande variable et la complexité numérique associée aux équations de Hamilton–Jacobi–Bellman. Enfin, elle présente les travaux fondés sur le Machine Learning, afin d'en dégager les apports, les limites et le positionnement du présent mémoire.

### **1.5.1 Commande optimale des systèmes manufacturiers sous demande constante**

Le contrôle optimal des systèmes manufacturiers sujets aux pannes et réparations aléatoires a d'abord été étudié en supposant un taux de demande constant. Cette hypothèse simplifie la formulation mathématique, conduit à des équations de Hamilton–Jacobi–Bellman (HJB) stationnaires et permet d'obtenir des politiques de type seuil critique (Akella *et al.*, 1986). Dans ce cadre, les décisions optimales dépendent uniquement de l'état instantané du système (stock, mode de la machine), et non de l'évolution future de la demande. Cette section distingue les

systèmes mono-produit, qui constituent le cadre de base, et les systèmes multi-produits, qui ajoutent la complexité de la gestion simultanée de plusieurs flux.

### 1.5.1.1 Systèmes mono-produit

Dans les systèmes mono-produit, plusieurs travaux ont établi et raffiné les politiques à seuil critique sous demande constante. En s'appuyant sur Akella *et al.* (1986), Gharbi *et al.* (2006) proposent une *Modified Hedging Corridor Policy* obtenue par programmation dynamique stochastique, qui fournit une politique de production robuste pour des systèmes soumis aux pannes. Ces travaux montrent que la structure en seuil reste pertinente dès lors que la demande est stationnaire.

Les études ultérieures enrichissent ce cadre en intégrant différentes contraintes opérationnelles. Rivera-Gómez *et al.* (2013) tiennent compte de la dégradation de la machine et de la détérioration de la qualité, en couplant production, inspection et décisions de réparation ou de révision. Kouedou *et al.* (2014) étendent l'analyse à un système hybride manufacturing–remanufacturing, où les taux de défaillance dépendent du rythme de production et où une politique multi-seuils pilote à la fois les pièces neuves et les produits retournés.

D'autres contributions se concentrent sur la qualité, la péremption et la durabilité. Les travaux de Polotski *et al.* (2021), Kaddachi *et al.* (2021) et Megoze *et al.* (2022) combinent contrôle de production, maintenance préventive et gestion de la qualité dans des systèmes qui se dégradent au cours du temps, souvent avec inventaires périssables. Plus récemment, Kenné *et al.* (2024) proposent une *Environmental Hedging Point Policy* intégrant un coût environnemental, tandis que Gharbi *et al.* (2024) utilisent la proportion de produits défectueux comme indicateur de dégradation pour optimiser conjointement production et maintenance.

Ces travaux montrent que le cadre HJB sous demande constante est suffisamment flexible pour intégrer dégradation, remanufacturing, qualité et environnement. Cependant, la structure des politiques reste stationnaire : les seuils critiques ne dépendent pas explicitement du temps, ce qui

est cohérent avec l'hypothèse d'une demande constante mais limite la pertinence de ces résultats lorsque la demande suit des profils saisonniers ou périodiques.

### 1.5.1.2 Systèmes multi-produits

Dans les systèmes multi-produits, la difficulté principale réside dans le partage de la capacité entre plusieurs flux de production. Les travaux de Hajji *et al.* (2003) introduisent une *Modified Hedging Corridor Policy* pour un système mono-machine multi-produits avec pannes et coûts de *setup*. La politique optimale résulte d'une programmation dynamique stochastique couplée à la simulation, et détermine conjointement les seuils de production et les instants de changement de produit. Kenne *et al.* (2003) généralisent cette approche à des systèmes multi-machines multi-produits, avec contrôle conjoint des taux de production et de la maintenance corrective.

Dans une perspective plus intégrée, Mifdal *et al.* (2013) et Mifdal *et al.* (2014b) relient explicitement le profil de production multi-produits à la dégradation des machines, et optimisent simultanément production et maintenance préventive sur un horizon fini. Assid *et al.* (2015) confirment l'intérêt de politiques hybrides combinant production, maintenance et gestion des *setups* pour réduire les coûts globaux. Plus récemment, Yu *et al.* (2024) et Zhang *et al.* (2025) intègrent la qualité des produits et la fiabilité dans des modèles multi-machines multi-produits se dégradant, montrant que les politiques optimales restent de type seuils couplés.

Une grande partie de la littérature sur le contrôle optimal des systèmes manufacturiers suppose un taux de demande constant, ce qui conduit à des équations de Hamilton–Jacobi–Bellman stationnaires et à des politiques de type seuil critique (*hedging point policy*). Cette hypothèse simplifie l'analyse, mais elle est en décalage avec de nombreux contextes industriels où la demande présente des variations saisonnières ou cycliques (Black *et al.*, 2006; Yang, 2016; Yang *et al.*, 2017). Dans ces situations, des politiques stationnaires peuvent être nettement sous-optimales, alors que les politiques véritablement optimales deviennent dépendantes du temps, comme l'illustrent les travaux sur la gestion des stocks et le contrôle de systèmes sous demande périodique (Graves *et al.*, 2008; Tunc *et al.*, 2011; Polotski *et al.*, 2019).

### 1.5.2 Intégration de la variabilité de la demande

La variabilité de la demande constitue une réalité industrielle incontournable dans de nombreux secteurs — agroalimentaire, textile, électronique, automobile — où les fluctuations saisonnières ou cycliques nécessitent une adaptation continue des décisions de production. L'intégration de cette variabilité dans les modèles de contrôle optimal, généralement via des processus stochastiques ou des fonctions périodiques, complexifie significativement la résolution des équations de Hamilton-Jacobi-Bellman qui deviennent non stationnaires.

Ouaret *et al.* (2011) ont proposé un modèle hiérarchique de contrôle optimal pour un système avec demande stochastique, combinant un contrôle linéaire-quadratique au premier niveau et un contrôle prédictif au second niveau pour ajuster la production. Ouaret *et al.* (2013) ont étendu ces principes à un système manufacturier-remanufacturier avec un taux de demande modélisé par un processus de diffusion, développant ainsi des politiques plus réactives intégrant l'incertitude dynamique.

Mifdal *et al.* (2013) ont abordé les systèmes mono-machine multi-produits avec demande aléatoire en modélisant la dépendance entre rythme de production et dégradation matérielle, proposant une stratégie conjointe production-maintenance. Mifdal *et al.* (2014b) ont affiné ces politiques en intégrant la maintenance préventive avec une résolution par programmation dynamique stochastique.

Polotski *et al.* (2016) ont introduit une estimation dynamique de la demande via des observateurs d'état, permettant une adaptation continue des politiques de contrôle. Ouaret *et al.* (2018) ont proposé une extension intégrant fiabilité des machines et qualité des produits, avec un pilotage croisé production-maintenance obtenu par résolution numérique des équations d'HJB.

Un tournant méthodologique important est apporté par Polotski *et al.* (2019), qui modélisent le taux de demande par une fonction sinusoïdale périodique dans un cadre de contrôle optimal stochastique non stationnaire. La politique optimale obtenue, appelée *hedging-curve policy*, généralise la politique classique *hedging point* en prescrivant un niveau de stock optimal variable

dans le temps dont la trajectoire anticipe les fluctuations de la demande. Plus récemment, Luo *et al.* (2023) ont proposé une approche intégrée de tarification dynamique et de contrôle de production pour des produits périssables, tandis que Wang *et al.* (2024) ont développé un modèle pour optimiser les stratégies dans un contexte de demande volatile liée aux tendances de la mode.

Les travaux qui abandonnent l'hypothèse de demande constante se regroupent globalement en deux familles. La première modélise la demande comme un processus stochastique, ce qui permet d'introduire l'incertitude dans les équations de Hamilton–Jacobi–Bellman. Cette approche reste toutefois mal adaptée aux demandes périodiques ou saisonnières : traitée comme un bruit aléatoire, la demande n'exploite pas pleinement les régularités temporelles observées dans de nombreux secteurs, où les profils suivent des schémas prévisibles (Ye *et al.*, 2024). Les politiques obtenues demeurent alors surtout réactives.

La seconde famille représente au contraire la demande comme une fonction déterministe du temps, souvent périodique. Elle permet de construire des politiques qui anticipent les pics et creux de demande, mais au prix d'équations HJB non stationnaires nécessitant une discrétisation fine de l'horizon et une résolution à chaque instant (Polotski *et al.*, 2019). Le coût de calcul associé rend l'exploration de scénarios et les analyses de sensibilité difficilement compatibles avec une utilisation courante en contexte industriel. Ce compromis entre réalisme de la demande et complexité numérique motive ainsi le recours à des approches complémentaires, notamment basées sur le Machine Learning, pour approximer plus rapidement les paramètres des politiques optimales.

### **1.5.3 Approches par Machine Learning**

Face aux limitations computationnelles inhérentes à la résolution numérique des équations d'Hamilton-Jacobi-Bellman, particulièrement accentuées dans les contextes non stationnaires où toute modification paramétrique impose une résolution complète, l'apprentissage automatique émerge comme une alternative prometteuse. Cette approche permet potentiellement de contourner les coûts de calcul élevés tout en préservant la qualité des solutions. La littérature récente

témoigne d'un intérêt croissant pour l'application du machine learning à l'optimisation des systèmes manufacturiers, avec une diversité d'architectures allant des méthodes basées sur les distances à l'apprentissage profond.

### 1.5.3.1 Méthodes basées sur les distances (kNN)

L'algorithme des k plus proches voisins (k-Nearest Neighbors, kNN) représente une approche d'apprentissage supervisé simple mais efficace, reposant sur le principe que des configurations similaires d'un système conduisent à des décisions optimales comparables. Cette méthode ne nécessite aucune hypothèse préalable sur la forme de la relation entrées-sorties et s'avère particulièrement adaptée aux problèmes de faible dimensionnalité.

Dans une démarche appliquée, Bensaci *et al.* (2021) proposent une architecture hybride combinant un réseau de neurones convolutionnel et une régression kNN pour l'annotation automatique d'images industrielles, permettant de classifier efficacement les données visuelles issues de capteurs dans divers milieux de production. Bien que cette contribution témoigne de la polyvalence du kNN, elle ne s'inscrit pas dans une problématique de commande optimale ou d'optimisation en environnement manufacturier au sens strict.

Dans une démarche qui s'apparente au contrôle, Ho *et al.* (2021) appliquent le kNN à l'optimisation énergétique d'un système de refroidissement industriel, démontrant la capacité de la méthode à ajuster dynamiquement les réglages afin de minimiser les émissions de carbone. Bien que l'étude s'appuie sur le kNN pour optimiser la consommation énergétique d'un système industriel spécifique, elle ne formalise pas une politique de commande optimale.

David *et al.* (2024) appliquent le kNN à l'estimation du taux de défauts dans un processus de production standard, soulignant la capacité de l'algorithme à fonctionner efficacement dans un environnement incertain. Toutefois, l'approche reste purement descriptive et prédictive, sans formulation explicite d'une stratégie de commande optimale ou de prise de décision. De manière similaire, d'autres applications du kNN à des tâches de classification ou de prédiction dans des

contextes manufacturiers démontrent son utilité pratique, bien que son application directe au contrôle optimal reste limitée par la malédiction de la dimensionnalité.

### 1.5.3.2 Réseaux de neurones (ANN, RBF)

Les réseaux de neurones artificiels (Artificial Neural Networks, ANN) occupent une place significative dans la littérature, avec des applications allant de l'aide à la décision à la prédiction de seuils optimaux. Leur capacité à approximer des fonctions non linéaires complexes en fait des candidats naturels pour modéliser les relations entre paramètres de systèmes manufacturiers et décisions optimales.

Priore *et al.* (2019) développent une architecture qui intègre plusieurs politiques d'approvisionnement alternatives et choisit la meilleure en temps réel selon un classificateur — réseau de neurones, forêts aléatoires, arbres de décisions — appris sur des données simulées. L'objectif est d'améliorer la performance opérationnelle dans des environnements dynamiques où la demande, les délais et autres caractéristiques évoluent rapidement. Toutefois, la démarche s'apparente davantage à un apprentissage supervisé de classification pour choisir parmi des règles existantes, sans construire une stratégie adaptative optimale découlant d'un modèle dynamique.

Behnamfar *et al.* (2022b) adoptent une démarche basée sur la commande optimale stochastique passant par la résolution numérique des équations d'HJB, qu'ils utilisent pour entraîner un modèle de réseaux de neurones à base radiale (Radial Basis Function, RBF) servant à prédire les taux de production et d'émission. Cette approche combine la rigueur de la commande optimale avec la capacité d'apprentissage des réseaux neuronaux. Toutefois, l'aspect continu du taux de demande est négligé car il est modélisé par des variations discrètes, limitant ainsi la capacité du modèle à gérer efficacement des fluctuations continues.

Subramonian *et al.* (2023) exploitent des réseaux de neurones artificiels pour modéliser et prédire la qualité des pièces fabriquées par impression 3D dans un contexte manufacturier. Leur approche se concentre sur la capacité prédictive des réseaux à capturer les relations complexes

entre paramètres de fabrication et caractéristiques mécaniques, sans formuler un problème de commande optimale stochastique.

### 1.5.3.3 Apprentissage profond (DNN, Deep RL)

L'apprentissage profond (Deep Learning) et l'apprentissage par renforcement profond (Deep Reinforcement Learning, DRL) représentent les développements les plus récents et prometteurs pour l'optimisation de systèmes manufacturiers complexes. Ces approches se distinguent par leur capacité à traiter des espaces de décision de haute dimensionnalité et à apprendre directement des politiques de contrôle dans des environnements incertains.

Petsagkourakis *et al.* (2020b) proposent un algorithme d'apprentissage profond par renforcement pour l'optimisation de processus chimiques sous contraintes probabilistes, démontrant la capacité de leur approche à gérer efficacement les contraintes stochastiques dans plusieurs études de cas industriels. Bien que rigoureuse et adaptée à ces environnements incertains, cette méthodologie ne traite pas directement des politiques optimales de production ou de maintenance, ce qui limite sa portée pour les problématiques classiques de production manufacturière.

Gijsbrechts *et al.* (2022b) modélisent des problèmes classiques de gestion des stocks sous demande aléatoire en processus de décision markovien, puis entraînent un agent via le Deep Reinforcement Learning afin d'optimiser la quantité à commander. Leur approche démontre la capacité du DRL à découvrir des politiques performantes par interaction directe avec l'environnement, bien que la garantie d'optimalité reste difficile à établir formellement.

Paraschos *et al.* (2024b) proposent un réseau de neurones profond pour l'optimisation conjointe de production, maintenance et remanufacturing dans un système bi-machines soumis aux dégradations. Leur contribution démontre l'applicabilité des DNN à des problématiques intégrées de contrôle de production.

Yan *et al.* (2025) appliquent l'apprentissage profond par renforcement à un système d'inventaire avec délais d'approvisionnement, générant des politiques quasi-optimales tout en réduisant

drastiquement le temps de calcul. Peng *et al.* (2025) utilisent des réseaux de neurones graphiques pour l'ordonnancement d'atelier flexible multi-produits, surpassant significativement les règles de dispatching classiques.

Dans l'ensemble, les travaux mobilisant l'apprentissage automatique pour l'optimisation des systèmes de production présentent plusieurs limites récurrentes. D'abord, les politiques apprises ne sont que rarement confrontées directement aux solutions issues des équations de Hamilton–Jacobi–Bellman, de sorte que leur optimalité reste le plus souvent inconnue ; le travail de Behnamfar *et al.* (2022b), fondé sur des solutions des équations d'HJB, fait figure d'exception mais demeure limité à des cas où la demande est stationnaire. Ensuite, le Machine Learning est fréquemment utilisé pour sélectionner, parmi un ensemble fini de politiques ou d'heuristiques prédéfinies (Priore *et al.*, 2019; Yan *et al.*, 2025), plutôt que pour prédire directement les paramètres de politiques continues. Cette formulation bornée de l'espace de décision empêche de reconstruire des politiques dépendantes du temps, telles que celles qui apparaissent dans les contextes de demande périodiquement variable.

Par ailleurs, la demande est le plus souvent supposée constante ou modélisée comme une variable aléatoire sans dépendance explicite au temps, ce qui ne permet pas de tirer parti de la structure périodique ou saisonnière observée en pratique. Enfin, le choix des algorithmes d'apprentissage (k plus proches voisins, réseaux de neurones classiques ou profonds, apprentissage par renforcement, etc.) est rarement justifié en fonction de la dimension de l'espace d'état ou du nombre de paramètres à prédire. Ces limites soulignent l'absence d'un cadre méthodologique unifié qui ancre l'apprentissage automatique dans le contrôle optimal, tienne compte explicitement de la variabilité temporelle de la demande et relie la complexité du système au choix des modèles d'apprentissage.

## **1.6 Synthèse, problématique, méthodologie et contributions**

### **1.6.1 Synthèse critique de la littérature**

L'analyse des travaux existants montre la richesse des approches développées pour la commande optimale stochastique des systèmes de production soumis à l'incertitude, en particulier grâce à la programmation dynamique stochastique. De nombreux modèles analytiques rigoureux intègrent coûts économiques et fiabilité, et mettent en évidence des politiques optimales de type seuil qui constituent une base théorique solide pour l'étude des systèmes manufacturiers.

Cette revue met toutefois en lumière plusieurs limites récurrentes. D'une part, une large part de la littérature repose sur une demande constante ou faiblement variable, ou bien traite la variabilité par scénarios discrets, ce qui reste éloigné des environnements où la demande évolue de manière marquée dans le temps. Lorsque la demande est modélisée comme une fonction du temps, les équations de Hamilton–Jacobi–Bellman deviennent non stationnaires et leur résolution numérique nécessite une discrétisation fine et des calculs répétés, ce qui rend coûteuse l'exploration systématique de différentes configurations et limite les analyses de sensibilité.

D'autre part, les solutions optimales sont le plus souvent obtenues pour des jeux de paramètres fixes, sans mécanisme de généralisation à de nouvelles configurations, et les approches récentes par apprentissage automatique restent en grande partie déconnectées des solutions issues des équations HJB, en particulier lorsque la demande varie explicitement dans le temps. Il en résulte un écart persistant entre la rigueur des méthodes de commande optimale et les exigences de réactivité et de flexibilité des systèmes manufacturiers modernes, qui motive le développement d'approches capables de capitaliser sur les solutions existantes tout en facilitant l'exploitation opérationnelle dans des contextes complexes et non stationnaires. Le Tableau 1.1 synthétise l'ensemble des travaux recensés selon cinq dimensions principales : le type de système (mono-produit ou multi-produits), la nature de la demande (constante ou variable), l'objectif de l'étude (obtention d'une politique de commande optimale ou réduction du temps de calcul), l'utilisation d'approches d'apprentissage automatique (kNN, ANN/RBF DNN) et la justification du choix

des algorithmes. Cette organisation permet d'identifier les espaces de recherche encore peu explorés et positionne la contribution du présent mémoire par rapport à la littérature existante.



### 1.6.2 Problématiques et objectifs de recherche

L'introduction d'une demande périodiquement variable dans les modèles de contrôle optimal conduit à des équations de Hamilton–Jacobi–Bellman non stationnaires, dont la résolution numérique est coûteuse. La politique optimale, obtenue grâce à une discrétisation temporelle de la demande, ne se limite alors plus à un seuil fixe, mais prend la forme d'une trajectoire dépendante du temps. Cela augmente le temps de calcul et c'est cette contrainte qui limite fortement l'exploration de scénarios multiples et l'utilisation directe de ces approches en aide à la décision industrielle.

Dans ce contexte, l'apprentissage automatique apparaît comme un levier pertinent pour approximer rapidement les paramètres de politiques optimales à partir de solutions de référence issues des équations HJB. Toutefois, la littérature propose peu de cadres permettant de relier le choix des algorithmes d'apprentissage à la complexité du problème, notamment en fonction de la dimension de l'espace des variables d'entrée et du nombre de paramètres à prédire. Il demeure ainsi difficile d'identifier quels types de modèles sont les plus adaptés selon le niveau de complexité du système considéré.

Dans ce contexte, la problématique de ce mémoire se résume en deux questions de recherche : comment prédire efficacement les paramètres d'une politique optimale de production dans un système manufacturier soumis à une demande périodiquement variable, en s'appuyant sur des solutions de commande optimale tout en réduisant le temps de calcul ? Et comment relier ce choix d'algorithmes de Machine Learning à la complexité des systèmes considérés, afin d'identifier les approches les plus adaptées selon le niveau de dimensionnalité du problème ?

Pour répondre à ces questions, ce mémoire a pour objectifs :

- De proposer une approche hybride combinant commande optimale stochastique et apprentissage supervisé pour prédire les paramètres d'une politique de production optimale dans un système soumis à une demande périodique variable ;
- D'évaluer et de comparer la capacité de différents algorithmes d'apprentissage supervisé, en particulier l'algorithme des k-plus proches voisins (kNN), les réseaux de neurones profonds

(DNN) et les réseaux de neurones à base radiale (RBF), à approximer fidèlement les solutions de référence issues de la résolution numérique des équations HJB ;

- D'analyser l'impact de la complexité et de la dimensionnalité du système sur les performances des modèles prédictifs, au moyen d'analyses de sensibilité et d'analyses de scénarios, afin de dégager des éléments de justification du choix algorithmique en fonction du problème considéré ;
- De valider l'approche proposée sur des systèmes manufacturiers de complexité croissante, en passant d'un système mono-produit à un système multi-produits, et d'apprécier dans chaque cas la capacité des modèles appris à fournir des politiques de production directement exploitables en pratique.

### 1.6.3 Méthodologie

La démarche adoptée dans ce mémoire repose sur une méthodologie structurée, appliquée de manière cohérente aux différents cadres étudiés. Elle vise à combiner des méthodes de commande optimale avec des outils d'apprentissage automatique afin de répondre aux exigences d'analyse, d'exploration de scénarios et d'aide à la décision dans les systèmes de production. Cette méthodologie s'organise autour des étapes principales suivantes :

- **Modélisation du système de production étudié.** Chaque cadre est d'abord formalisé par une modélisation explicite du système de production considéré, incluant la structure du système, la dynamique des stocks, la fiabilité des équipements (pannes et réparations) ainsi que les caractéristiques de la demande. Cette étape permet de définir les états du système, les décisions de commande et les paramètres économiques pertinents.
- **Formulation du problème de commande optimale.** À partir du modèle établi, le pilotage du système est formulé comme un problème de commande optimale stochastique visant à minimiser un critère de coût global intégrant les coûts de stockage, de rupture et d'exploitation. Cette formulation fournit le cadre théorique nécessaire à la détermination de politiques de production optimales.

- **Construction des bases de données à partir des résolutions numériques.** Les bases de données utilisées pour l'apprentissage sont construites en résolvant à plusieurs reprises les équations de commande optimale pour différentes configurations du système. Les paramètres d'entrée des modèles sont ainsi générés en faisant varier les caractéristiques de la demande, les paramètres économiques et les paramètres de fiabilité, tandis que les sorties correspondent aux paramètres optimaux des politiques de production obtenus par résolution numérique. Cette étape permet de constituer un ensemble de données représentatif des comportements optimaux du système.
- **Apprentissage et validation des modèles.** Les bases de données construites sont exploitées pour entraîner des modèles d'apprentissage supervisé capables d'établir une relation entre les paramètres du système et les paramètres des politiques optimales. La validation des modèles repose sur une stratégie en trois niveaux complémentaires : (i) une validation croisée pour le réglage des hyperparamètres de chaque algorithme, (ii) une évaluation quantitative sur un ensemble de test indépendant à l'aide de l'erreur absolue moyenne (MAE) pour les trajectoires optimales et de l'erreur relative moyenne (ERM) pour le coût optimal, et (iii) une validation externe par comparaison directe des prédictions aux solutions de référence issues des équations d'HJB sur des scénarios représentatifs. Cette démarche permet d'évaluer à la fois la précision prédictive et la capacité de généralisation des modèles au-delà des données d'entraînement.
- **Analyses de sensibilité et analyses de scénarios.** Les modèles entraînés sont ensuite mobilisés pour mener des analyses de sensibilité et des analyses de scénarios, afin d'étudier l'influence des différents paramètres du système sur les politiques de production optimales. Ces analyses permettent d'évaluer la robustesse des approches proposées et d'explorer rapidement un large éventail de configurations pertinentes d'un point de vue industriel.
- **Implémentation de la politique optimale.** Enfin, les modèles validés sont utilisés pour fournir des politiques de production directement exploitables, permettant une prise de décision rapide et cohérente avec les solutions optimales issues des équations d'HJB, tout en répondant aux contraintes opérationnelles des systèmes de production.

#### 1.6.4 Contributions

Les contributions de ce mémoire s'articulent autour de deux volets complémentaires :

- i)* **Système constitué d'une machine produisant un type de pièces soumis à une demande périodiquement variable** : les équations de Hamilton–Jacobi–Bellman sont d'abord résolues numériquement afin de constituer une base de données de solutions optimales pour un système mono-produit. Cette base sert ensuite à entraîner et comparer plusieurs modèles de régression supervisée (k-plus proches voisins, réseaux de neurones à base radiale, réseaux de neurones profonds) pour la prédiction des paramètres d'une politique de production optimale. Les résultats montrent que ces modèles permettent de reproduire fidèlement les solutions de référence tout en réduisant fortement le temps de calcul, ce qui rend possible l'exploration rapide de scénarios dans lesquels plusieurs paramètres du système varient simultanément.
- ii)* **Extension à un système constitué d'une machine produisant deux types de pièces** : l'approche est ensuite étendue à un système de production plus complexe, dans lequel l'espace d'état et l'espace des paramètres sont significativement plus riches. Le Machine Learning est étudié comme moyen de déterminer les paramètres optimaux de la politique de production même dans ce contexte avec un système plus complexe, afin de confirmer la pertinence de cette approche au-delà du cas mono-produit. Cette extension permet en outre d'analyser l'influence de l'augmentation de la dimension des variables d'entrée sur les performances des algorithmes considérés et de dégager des éléments de justification du choix algorithmique en fonction de la complexité du problème traité.

Les deux contributions présentées ci-dessus sont développées respectivement aux chapitres 2 et 3.

## 1.7 Conclusion

Ce chapitre a posé les fondements conceptuels et méthodologiques du mémoire. La revue de la littérature a mis en évidence le coût computationnel élevé des résolutions répétées des équations d'HJB sous demande périodique, et l'intérêt de l'apprentissage automatique pour approximer rapidement les paramètres des politiques optimales. Le chapitre suivant applique cette approche à un système constitué d'une machine produisant un type de pièces, en comparant plusieurs algorithmes d'apprentissage supervisé. Le chapitre 3 étend ensuite l'étude à une machine produisant deux types de pièces, afin d'analyser l'influence de la dimensionnalité sur le choix algorithmique.



## CHAPITRE 2

### DÉVELOPPEMENT DE L'APPROCHE POUR UN SYSTÈME MANUFACTURIER PRODUISANT UN SEUL TYPE DE PIÈCE

#### 2.1 Introduction

Ce chapitre porte sur la prédiction des paramètres optimaux d'un système de production constitué d'une machine produisant un type de pièce et soumis à une demande périodiquement variable. Bien que les équations de Hamilton–Jacobi–Bellman (HJB) fournissent une base théorique solide pour la détermination de politiques de production optimales, leur résolution numérique devient coûteuse lorsque la demande varie dans le temps, ce qui limite leur utilisation dans un contexte industriel.

Afin de contourner cette difficulté, ce chapitre explore l'utilisation du Machine Learning comme outil de prédiction des paramètres optimaux, à partir de données issues de résolutions numériques des équations d'HJB. L'objectif est d'évaluer la capacité de différents algorithmes d'apprentissage supervisé à reproduire fidèlement ces paramètres dans un cadre de dimensionnalité modérée.

Deux expériences successives sont menées. La première compare plusieurs algorithmes de Machine Learning lorsque seuls les paramètres liés à la demande varient. La seconde approfondit l'analyse en se concentrant sur l'algorithme des  $k$  plus proches voisins, en intégrant des paramètres économiques et de fiabilité.

Le chapitre s'articule autour de la modélisation du système (section 2.2), de la reformulation du problème de commande optimale en une tâche d'apprentissage automatique (section 2.3), de la méthodologie adoptée (section 2.4), de l'analyse comparative des algorithmes (section 2.5), de l'extension du modèle étudié (section 2.6), et se conclut par une synthèse des résultats.

#### 2.2 Modélisation du système et commande optimale stochastique

Le système étudié se compose d'une machine sujette à des pannes et réparations aléatoires, produisant un type de pièce dont le stock est utilisé pour répondre à une demande de marché

variant dans le temps. Le décideur contrôle le taux de production  $u$  pour optimiser la performance globale du système. La Figure 2.1 illustre ce système manufacturier.

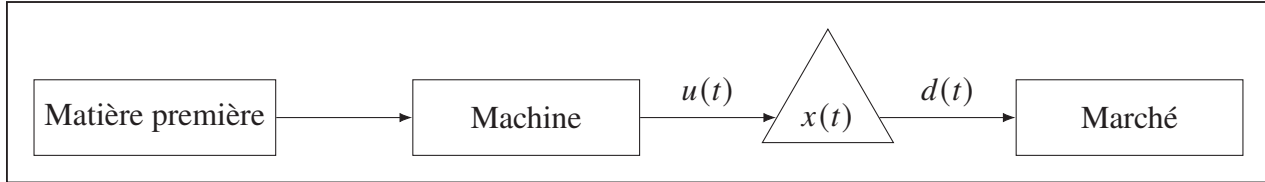


Figure 2.1 Structure du système manufacturier

La dynamique du stock est définie par :

$$\frac{dx}{dt} = u(t) - d(t) \quad (2.1)$$

Le taux de demande périodique est modélisé par une fonction sinusoïdale :

$$d(t) = d_m + d_a \sin(\omega t + \varphi) \quad (2.2)$$

où  $d_m$  est le taux de demande moyen,  $d_a$  l'amplitude des variations,  $\omega$  la fréquence et  $\varphi$  la phase initiale. Pour garantir la positivité du taux de demande, on impose  $d_m > d_a$ .

### 2.2.1 Diagramme des taux de transition

Soit  $\{\xi(t), t \geq 0\}$  le processus stochastique décrivant l'état de la machine à l'instant  $t$ , avec espace d'états  $\mathcal{E} = \{1, 2\}$  :

$$\xi(t) = \begin{cases} 1, & \text{machine en fonctionnement,} \\ 2, & \text{machine à l'arrêt (en panne / en réparation)} \end{cases} \quad (2.3)$$

On utilise  $\alpha \in \mathcal{E}$  pour désigner un état générique de la machine.

Les changements d'état de la machine sont modélisés par un processus de Markov à temps continu sur l'espace d'états  $\mathcal{E} = \{1, 2\}$ . On note  $q_{\alpha\beta}$  le taux de transition de l'état  $\alpha$  vers l'état  $\beta$ , pour  $\alpha, \beta \in \mathcal{E}$ ,  $\alpha \neq \beta$ . Dans le cas étudié, on dispose de deux taux de transition :  $q_{12}$  est le passage de l'état 1 à l'état 2 et  $q_{21}$  est le passage de l'état 2 à l'état 1.

Le diagramme de la Figure 2.2 illustre ces transitions entre les deux états de la machine.

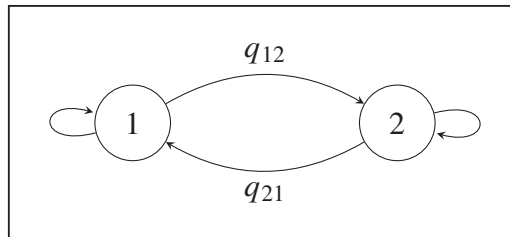


Figure 2.2 Diagramme des taux de transition entre les deux états de la machine

La matrice des taux de transition (ou matrice génératrice) associée au processus  $\{\xi(t), t \geq 0\}$  s'écrit alors :

$$Q = (q_{\alpha\beta})_{\alpha, \beta \in \mathcal{E}} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix} = \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix}. \quad (2.4)$$

Les probabilités de transition sur un petit intervalle de temps  $[t, t + dt]$  s'en déduisent :

$$\begin{aligned} \mathbb{P}(\xi(t + dt) = 2 \mid \xi(t) = 1) &= q_{12} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 1 \mid \xi(t) = 2) &= q_{21} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 1 \mid \xi(t) = 1) &= 1 - q_{12} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 2 \mid \xi(t) = 2) &= 1 - q_{21} dt + o(dt). \end{aligned} \quad (2.5)$$

Ces expressions traduisent le fait que, sur un intervalle de durée infinitésimale  $dt$ , la probabilité d'un changement d'état est proportionnelle au taux de transition correspondant, tandis que la probabilité de rester dans le même état est proche de 1.

### 2.2.2 Probabilités limites et conditions de faisabilité

Lorsque les taux de panne et de réparation  $q_{12}$  et  $q_{21}$  sont constants, le processus de Markov  $\{\xi(t)\}$  admet une distribution stationnaire  $\pi = (\pi_1, \pi_2)$ , où  $\pi_\alpha$  représente la probabilité limite d'être dans l'état  $\alpha \in \{1, 2\}$ . Cette distribution est définie par le système linéaire classique

$$\pi Q = \mathbf{0}, \quad \pi_1 + \pi_2 = 1, \quad (2.6)$$

soit explicitement

$$(\pi_1 \ \pi_2) \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix} = (0 \ 0), \quad \pi_1 + \pi_2 = 1. \quad (2.7)$$

On en déduit les probabilités limites

$$\pi_1 = \frac{q_{21}}{q_{12} + q_{21}}, \quad \pi_2 = \frac{q_{12}}{q_{12} + q_{21}}. \quad (2.8)$$

La quantité  $\pi_1$  représente le taux de disponibilité de la machine, c'est-à-dire la proportion de temps, à long terme, durant laquelle celle-ci est en état de fonctionnement.

La demande périodique présentée plus haut atteint sa valeur maximale

$$d_{\max} = d_m + d_a.$$

Pour que le système soit faisable, la capacité de production moyenne, lorsque la machine est disponible et fonctionne au taux de production maximal  $U_{\max}$ , doit être supérieure à cette valeur maximale. En notant  $\pi_1$  la probabilité limite de disponibilité de la machine (voir (2.8)), on obtient la condition de faisabilité

$$\pi_1 U_{\max} > d_m + d_a. \quad (2.9)$$

En remplaçant  $\pi_1$  par son expression en fonction des taux de transition, on arrive à

$$\frac{q_{21}}{q_{12} + q_{21}} U_{\max} > d_m + d_a \iff (U_{\max} - d_m - d_a) q_{21} > (d_m + d_a) q_{12}. \quad (2.10)$$

Cette inégalité exprime que, à long terme, la capacité moyenne effective de la machine (pondérée par sa disponibilité) doit excéder la demande maximale.

On suppose par ailleurs que la demande varie « lentement » par rapport à l'horizon effectif induit par le facteur d'actualisation  $e^{-\rho t}$ . La demi-période de la demande sinusoïdale vaut

$$T_{\frac{1}{2}} = \frac{\pi}{\omega}, \quad (2.11)$$

tandis que le temps caractéristique  $t_{10}$  à partir duquel le facteur d'actualisation est réduit à 10% est donné par

$$e^{-\rho t_{10}} = 0.1 \iff t_{10} = \frac{\ln 10}{\rho} \approx \frac{2.3}{\rho}. \quad (2.12)$$

La condition de « lente variation » adoptée est alors

$$\frac{\pi}{\omega} > \frac{2.3}{\rho}, \quad (2.13)$$

ce qui garantit que la demande évolue sur une échelle de temps plus grande que celle sur laquelle les coûts sont fortement atténués par l'actualisation.

### 2.2.3 Conditions d'optimalité

On définit les parties positive et négative du niveau de stock  $x$  par

$$x^+ = \max(0, x), \quad x^- = \max(0, -x).$$

Le coût instantané est alors donné par

$$h(.) = c^+ x^+ + c^- x^-, \quad (2.14)$$

où  $c^+$  et  $c^-$  désignent respectivement les coûts unitaires de possession (stockage) et de rupture (backlog).

Pour une commande admissible  $u(\cdot)$ , le coût total actualisé est défini par

$$J(u(\cdot), x, t, \alpha) = \mathbb{E} \left[ \int_t^\infty e^{-\rho(s-t)} h(x(s)) ds \mid x(0) = x_0, \xi(t) = \alpha \right], \quad \alpha \in \{1, 2\}, \quad (2.15)$$

où  $\rho > 0$  est le taux d'actualisation.

L'ensemble des commandes admissibles est défini par :

$$\Gamma(\alpha) = \{u(\cdot) \in \mathbb{R} \mid 0 \leq u(t) \leq U_{\max} \mathbf{1}_{\{\alpha=1\}}, \forall t \geq 0\}, \quad \alpha \in \{1, 2\}. \quad (2.16)$$

Soit  $v(x, t, \alpha)$  la fonction valeur :

$$v(x, t, \alpha) = \min_{u(\cdot) \in \Gamma(\alpha)} J(u(\cdot), x, t, \alpha), \quad \alpha \in \{1, 2\}. \quad (2.17)$$

Selon Polotski *et al.* (2019), les équations de Hamilton–Jacobi–Bellman (HJB) permettant de déterminer la politique optimale sont non stationnaires et s'écrivent :

$$\begin{aligned} \rho v(x, t, \alpha) = & \frac{\partial v(x, t, \alpha)}{\partial t} \\ & + \min_{u \in \Gamma(\alpha)} \left\{ (u \text{Ind}_{\{\alpha=1\}} - d(t)) \frac{\partial v(x, t, \alpha)}{\partial x} + h(x, u, \alpha) \right. \\ & \left. + q_{\alpha\beta} (v(x, t, \beta) - v(x, t, \alpha)) \right\}. \end{aligned} \quad (2.18)$$

Les équations d'Hamilton–Jacobi–Bellman permettent de déterminer la fonction valeur, à partir de laquelle la politique optimale est déduite. La variabilité de la demande rend toutefois le système non stationnaire, ce qui impose une résolution numérique.

### 2.2.4 Résolution numérique

La résolution analytique des équations de Hamilton–Jacobi–Bellman non stationnaires étant intractable dans la plupart des cas, une approximation numérique est nécessaire. L’approche retenue repose sur la méthode de discrétisation proposée par Kushner *et al.* (1992), quiconsiste à approximer les dérivées spatiales par des différences finies sur une grille de l’espace des états.

Soit  $h > 0$  le pas de discrétisation de l’espace des niveaux de stock  $x$ . La fonction valeur  $v(x, t, \alpha)$  est alors approchée par une fonction discrète  $v^h(x, t, \alpha)$  définie sur la grille. La dérivée partielle par rapport au stock est approximée par un schéma de différences finies amont (upwind), dépendant du signe de la dérivée temporelle du stock :

$$\frac{\partial v(x, t, \alpha)}{\partial x} \approx \begin{cases} \frac{v^h(x + h, t, \alpha) - v^h(x, t, \alpha)}{h}, & \text{si } \dot{x}(t) \geq 0, \\ \frac{v^h(x, t, \alpha) - v^h(x - h, t, \alpha)}{h}, & \text{si } \dot{x}(t) < 0, \end{cases} \quad (2.19)$$

où  $\dot{x}(t) = u(t) - d(t)$ .

En injectant l’approximation (2.19) dans les équations de Hamilton–Jacobi–Bellman non stationnaires, on obtient, pour chaque état  $\alpha \in \{1, 2\}$ , une équation aux différences finies.

Lorsque la machine est opérationnelle, la dynamique du stock est donnée par  $\dot{x}(t) = u(t) - d(t)$ , avec  $u(\cdot) \in \Gamma(1)$ . Pour alléger

On introduit les fonctions indicatrices

$$I^+ = \text{Ind}\{u - d(t) \geq 0\}, \quad I^- = \text{Ind}\{u - d(t) < 0\},$$

permettent de sélectionner la différence finie amont appropriée dans le schéma de Kushner (2.19).

Les équations de Hamilton–Jacobi–Bellman numériques associées au mode  $\alpha = 1$  s’écrivent

alors :

$$\begin{aligned} \rho v^h(x, t, 1) = & \frac{\partial v^h(x, t, 1)}{\partial t} + \min_{u(\cdot) \in \Gamma(1)} \left\{ h(x) + q_{12}(v^h(x, t, 2) - v^h(x, t, 1)) \right. \\ & + (u - d(t)) \frac{v^h(x + h, t, 1) - v^h(x, t, 1)}{h} I^+ \\ & \left. + (u - d(t)) \frac{v^h(x, t, 1) - v^h(x - h, t, 1)}{h} I^- \right\}. \end{aligned} \quad (2.20)$$

Lorsque la machine est en panne, aucune production n'est possible ( $u(t) = 0$ ), et la dynamique du stock devient  $\dot{x}(t) = -d(t)$ . L'équation HJB discrétisée associée est alors :

$$\begin{aligned} \rho v^h(x, t, 2) = & \frac{\partial v^h(x, t, 2)}{\partial t} + h(x) + q_{21}(v^h(x, t, 1) - v^h(x, t, 2)) \\ & - d(t) \left[ \frac{v^h(x + h, t, 2) - v^h(x, t, 2)}{h} \right] + \frac{v^h(x, t, 2) - v^h(x - h, t, 2)}{h} \left. \right]. \end{aligned} \quad (2.21)$$

La présence du terme de dérivée partielle par rapport au temps  $\partial v(x, t, \alpha) / \partial t$  dans les équations de Hamilton–Jacobi–Bellman rend leur résolution numérique particulièrement coûteuse. L'approche proposée par Polotski *et al.* (2019) repose sur une procédure en quatre étapes, décrite ci-dessous.

1. **Discrétisation du taux de demande.** L'intervalle des valeurs possibles du taux de demande est discrétisé en  $N$  niveaux  $d^j$ ,  $j = 1, \dots, N$ , tels que

$$d^1 = d_m - d_a, \quad d^N = d_m + d_a.$$

Chaque niveau de demande  $d^j$  est associé à un ou plusieurs instants  $t_j$  du domaine temporel, déterminés à partir du modèle sinusoïdal de la demande.

2. **Résolution des équations HJB stationnaires.** Pour chaque niveau de demande discret  $d^j$ , on résout les équations de Hamilton–Jacobi–Bellman stationnaires, obtenues en négligeant le terme de dérivée temporelle. Cette étape permet d'obtenir une famille de fonctions valeur

numériques

$$v^j(x, \alpha), \quad \alpha \in \{1, 2\},$$

associées à des valeurs constantes de la demande.

3. **Approximation du terme de dérivée temporelle.** Le terme de dérivée temporelle est approché par une différence finie entre deux fonctions valeur correspondant à deux niveaux de demande consécutifs. Plus précisément, pour tout  $(x, \alpha)$ , on utilise

$$\frac{\partial v(x, t, \alpha)}{\partial t} \approx \frac{v^{j+1}(x, \alpha) - v^j(x, \alpha)}{t_{j+1} - t_j}, \quad (2.22)$$

où  $t_j$  et  $t_{j+1}$  sont les instants associés respectivement aux niveaux de demande  $d^j$  et  $d^{j+1}$ .

4. **Réintégration dans les équations HJB numériques.** Les estimations obtenues pour le terme de dérivée temporelle sont ensuite réinjectées dans les équations de Hamilton–Jacobi–Bellman numériques. Les fonctions valeur sont recalculées sur la grille spatiale en intégrant explicitement l’approximation (2.22), ce qui conduit à une politique de production optimale dépendant du temps.

Leurs résultats démontrent que la politique optimale de production est une politique à courbe critique qui généralise la politique à seuil critique. La quantité optimale de produits à stocker  $Z^*(t)$  est une fonction du temps, dont la courbe suit la tendance du taux de demande, mais anticipe également ses variations, comme l’illustre la Figure 2.3. Les paramètres numériques utilisés pour obtenir la Figure 2.3 sont regroupés dans le Tableau 2.1.

Table 2.1 Paramètres numériques utilisés pour la Figure 2.3

$U_{\max}$	$\rho$	$q_{12}$	$q_{21}$	$c^+$	$d_m$	$d_a$	$\omega$	$\varphi$	$c^-$	$N$
1.75	0.01	0.05	0.15	1	0.75	0.05	$5 \times 10^{-3}$	$-\frac{\pi}{2}$	20	10

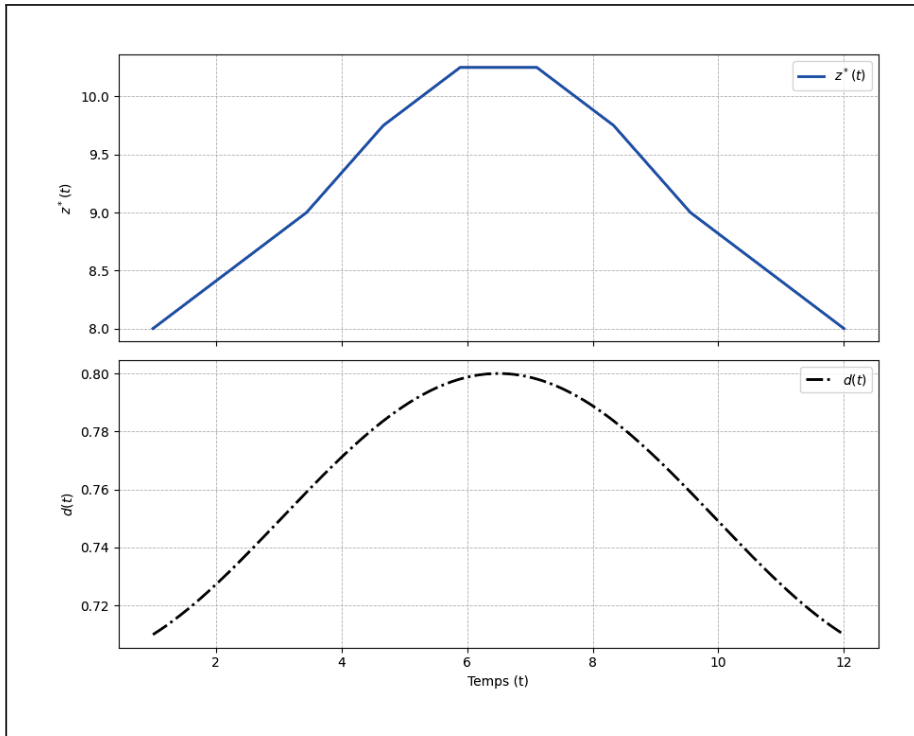


Figure 2.3 Illustration de l'effet d'anticipation

La politique optimale de production obtenue se résume à cette équation :

$$u^*(x, t) = \begin{cases} U_{\max} & \text{si } x(t) < Z^*(t) \\ d(t) & \text{si } x(t) = Z^*(t) \\ 0 & \text{si } x(t) > Z^*(t) \end{cases} \quad (2.23)$$

Pour évaluer le coût optimal associé à cette politique, nous avons développé une approche permettant d'approximer le coût du système à partir des fonctions valeur mises à jour. On note  $\bar{v}^j(x, \alpha)$  la fonction valeur numérique obtenue après intégration du terme de dérivée temporelle, au temps  $t_j$  et pour l'état de mode  $\alpha \in \{1, 2\}$ . Comme établi par Akella *et al.* (1986), la fonction valeur représente le coût total actualisé minimal du système pour une condition initiale donnée. Nous considérons donc que le coût optimal du système, noté  $C$ , peut être approché par la moyenne temporelle de la fonction valeur mise à jour sur l'intervalle de discrétisation.

En notant  $T_{\min} = t_1$  et  $T_{\max} = t_n$ , avec  $t_1 < \dots < t_N$ , le coût optimal est défini par

$$C = \frac{1}{T_{\max} - T_{\min}} \int_{T_{\min}}^{T_{\max}} \bar{v}(t, x, \alpha) dt, \quad (2.24)$$

par définition de la valeur moyenne d'une fonction sur un intervalle. Cependant, la fonction  $\bar{v}(t, x, \alpha)$  n'est disponible que sous forme discrète aux temps  $t_j$  via les valeurs  $\bar{v}^j(x, \alpha)$ . L'intégrale (2.24) est alors approchée par la méthode des trapèzes :

$$C \approx \hat{C} = \frac{1}{T_{\max} - T_{\min}} \sum_{j=1}^{n-1} \frac{t_{j+1} - t_j}{2} \left( \bar{v}^j(x, \alpha) + \bar{v}^{j+1}(x, \alpha) \right). \quad (2.25)$$

Cette approximation constitue la base du calcul du coût optimal associé à la politique de production obtenue par résolution numérique des équations de Hamilton–Jacobi–Bellman.

### 2.2.5 Limites de la résolution numérique des équations d'HJB

La résolution numérique développée dans cette section, fondée sur les équations de Hamilton–Jacobi–Bellman, permet désormais de déterminer non seulement la courbe critique  $Z^*(t)$ , mais également une approximation du coût optimal associé à la politique de production. Cet apport enrichit la contribution de Polotski *et al.* (2019). Néanmoins, malgré cela, plusieurs limites structurelles subsistent.

En premier lieu, le coût de calcul demeure élevé. Le temps de résolution croît avec le nombre de discrétisations temporelles de la demande ( $N$ ), le nombre de points de grille pour le niveau de stock ( que nous notons  $N_x$ ), le nombre de modes de fonctionnement de la machine (que nous notons  $N_\alpha$ ), le nombre de valeurs de commande testées ( que nous notons  $N_u$ ) et le nombre d'itérations nécessaires à la convergence (que nous notons  $K$ ). Le calcul du coût optimal, qui repose sur une intégration numérique de la fonction valeur sur l'intervalle temporel considéré, contribue également à alourdir la charge de calcul.

En second lieu, cette lourdeur numérique limite fortement la capacité d'exploration de scénarios. Toute modification d'un paramètre du système (caractéristiques de la demande, coûts, fiabilité de la machine, etc.) nécessite de reprendre l'ensemble de la procédure de calcul : résolution des équations HJB pour chaque niveau de demande, estimation du terme de dérivée temporelle, mise à jour des fonctions valeur et calcul du coût optimal. Dans ce contexte, la réalisation d'analyses de sensibilité approfondies ou la comparaison rapide de nombreuses configurations devient difficilement compatible avec des contraintes de temps industrielles.

Ces limites motivent l'introduction de méthodes basées sur l'apprentissage automatique, visant à exploiter les solutions issues des équations HJB comme références, afin de construire des modèles prédictifs capables de fournir rapidement des estimations fiables des paramètres optimaux de la politique de production et du coût associé, sans recourir systématiquement à une nouvelle résolution complète des équations HJB.

### **2.3 Reformulation du problème de commande optimale pour l'apprentissage automatique**

Le contrôle optimal d'un système de production soumis à une demande variable s'appuie classiquement sur la résolution des équations de Hamilton–Jacobi–Bellman (HJB), qui fournissent la politique optimale en fonction des paramètres du système et de l'état du stock. Toutefois, la complexité computationnelle associée à leur résolution, aggravée par l'ajout de variables industrielles pertinentes, rend difficile leur utilisation pratique pour explorer un large éventail de scénarios.

Afin de dépasser cette contrainte, le problème est reformulé comme une tâche d'apprentissage automatique supervisé. L'objectif est d'entraîner un modèle prédictif à restituer, à partir d'un ensemble de paramètres d'entrée observables, les sorties optimales obtenues par résolution HJB. Ce type de mapping autorise une prédiction rapide des trajectoires et coûts optimaux, même face à des variations importantes du contexte industriel.

Le processus expérimental est structuré sur l'idée d'une intégration progressive des variables d'entrée au modèle. Dans un premier temps, l'analyse porte sur les caractéristiques fondamentales du taux de demande, à savoir sa moyenne ( $d_m$ ), son amplitude ( $d_a$ ) et sa fréquence ( $\omega$ ). Cette étape vise à évaluer la capacité des modèles de Machine Learning à saisir les dynamiques essentielles d'un système soumis à une demande périodique.

L'étude se poursuit par l'intégration du coût unitaire de rupture de stock ( $c^-$ ), paramètre économique dont l'influence opérationnelle est majeure : il conditionne directement les stratégies de gestion des stocks et les risques encourus en cas d'approvisionnement insuffisant. Ce coût est souvent bien supérieur à celui de stockage dans l'industrie, traduisant la nécessité de stratégies proactives et robustes.

La complexité du modèle est finalement accrue avec l'ajout du taux de réparation de la machine ( $q_{21}$ ), qui introduit la notion de fiabilité et disponibilité industrielle au cœur des décisions optimales, modulant la capacité du système à répondre aux fluctuations de la demande.

Le modèle prend comme variables d'entrée  $[d_m, d_a, c^-, q_{21}]$  et fournit en sortie  $(Z^*(t), C)$ , où  $Z^*(t)$  désigne la trajectoire optimale (vectorisée) des stocks à maintenir au cours du temps et  $C$  le coût total associé à la politique optimale.

Cette démarche séquentielle, qui complexifie graduellement l'espace des paramètres, permet d'interroger la robustesse et la précision des modèles de Machine Learning face à des contraintes industrielles croissantes, tout en fournissant un cadre méthodologique adaptable à des besoins opérationnels variés.

## **2.4 Méthodologie de prédiction des paramètres optimaux de la politique de production**

Cette section présente la démarche adoptée pour prédire les paramètres optimaux de la politique optimale d'un système de production soumis à un taux de demande périodiquement variable, via l'apprentissage automatique supervisé, afin de contourner la complexité computationnelle des équations HJB. L'approche s'articule en deux phases progressives synthétisées dans la

Figure 2.4 : la première identifie l’algorithme le plus performant sur des variables de base (les paramètres du taux de demande), la seconde enrichit le modèle en intégrant des contraintes économiques et de fiabilité.

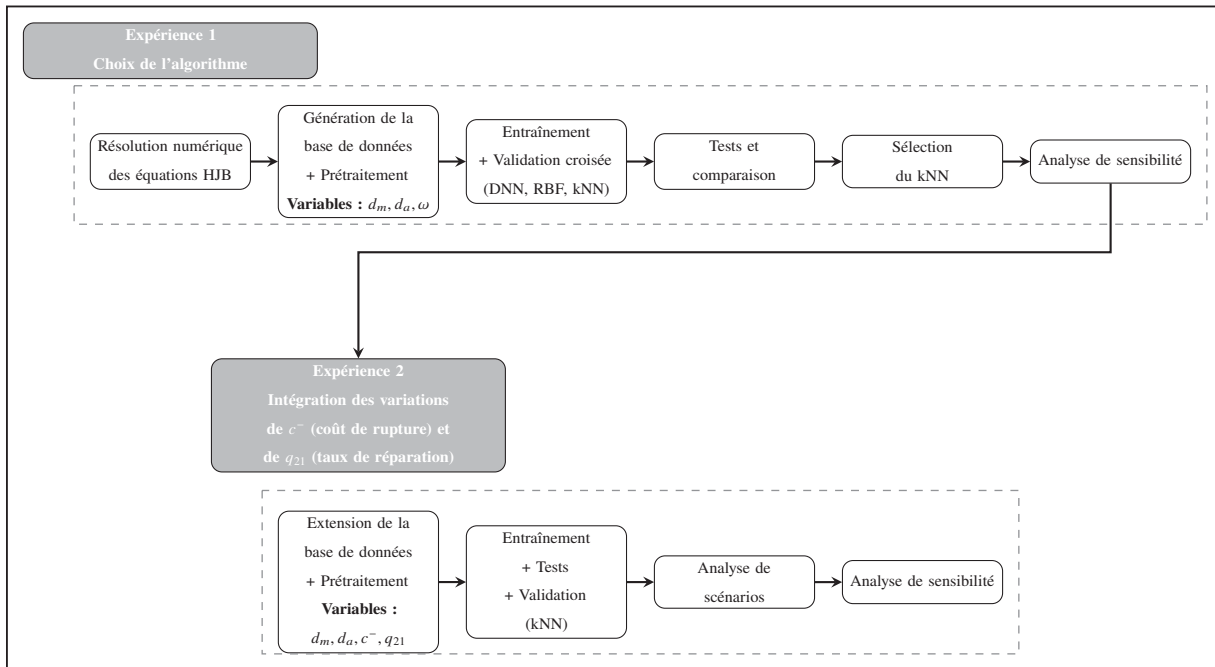


Figure 2.4 Représentation schématique de la méthodologie proposée

Dans la phase 1, Les paramètres optimaux (trajectoire critique  $Z^*(t)$  et coût) sont obtenus par résolution numérique des équations HJB pour différentes configurations des paramètres de demande ( $d_m, d_a, \omega$ ). Ces données constituent l’ensemble d’entraînement après normalisation. Trois algorithmes de régression supervisée sont comparés : réseaux de neurones profonds (DNN), réseaux à base radiale (RBF) et k-Nearest Neighbors (kNN). L’évaluation par validation croisée 10-fold révèle que le kNN offre la meilleure performance. Une analyse de sensibilité sur les trois variables d’entrée démontre que  $\omega$  exerce une influence négligeable sur les paramètres optimaux, justifiant son retrait pour les expériences suivantes.

Dans la phase 2, on enrichit le modèle en considérant la variation du coût de rupture de stock ( $c^-$ ) et celle du taux de réparation ( $q_{21}$ ), deux paramètres stratégiques aux impacts distincts. Le coût de rupture  $c^-$  est privilégié car, dans la majorité des contextes industriels (Silver *et al.*

(1998)), il excède largement le coût de stockage et influence fortement les décisions optimales : une rupture entraîne des ventes perdues, des pénalités contractuelles et une dégradation de la relation client. Le taux de réparation  $q_{21}$  est retenu plutôt que le taux de panne  $q_{12}$  car il constitue un levier d'action managérial direct, améliorable par une organisation performante de la maintenance. Plusieurs études (notamment Samir (2012)) ont montré que optimisation des processus de maintenance, la formation des techniciens, et la gestion efficace des pièces détachées permettent de réduire significativement le temps de réparation et d'améliorer la disponibilité des équipements, impactant ainsi favorablement le taux de réparation. À l'inverse,  $q_{12}$  dépend de facteurs structurels (âge, usure) moins maîtrisables à court terme et est supposé constant dans cette étude (Oualet *et al.* (2017)).

L'enrichissement s'effectue en deux couches. La première génère des configurations avec  $dm$ ,  $da$  et  $c^-$  en maintenant  $q_{21}$  fixe. La deuxième ajoute  $q_{21}$  en variant différemment  $dm$ ,  $da$  et  $c^-$  pour affiner la couverture de l'espace paramétrique. On résout numériquement les équations d'HJB non stationnaires pour chaque configuration pour obtenir les paramètres optimaux correspondants.

Le kNN, identifié comme meilleur algorithme en Phase 1, est entraîné sur les données enrichies après normalisation. Une validation croisée 10-fold garantit la robustesse du modèle. Les performances sont évaluées par l'erreur absolue moyenne (MAE) et l'erreur relative.

La robustesse du modèle est testée sur un ensemble de scénarios représentatifs couvrant différentes combinaisons de  $dm$ ,  $da$ ,  $c^-$  et  $q_{21}$ . Cette validation confirme la capacité du modèle à généraliser sur l'espace paramétrique étendu.

Des études de sensibilité sont menées pour quantifier l'influence respective de  $c^-$  et  $q_{21}$  sur les prédictions. Ces analyses révèlent les contributions relatives de chaque paramètre et confirment la pertinence de leur intégration dans le modèle et permettent de valider l'approche par apprentissage automatique comme méthode d'approximation des paramètres de la politique optimale.

## 2.5 Analyse comparative des algorithmes

Pour la première expérience, nous comparons trois algorithmes de machine learning en fonction de leur capacité à prédire les paramètres optimaux du système de production. L'ensemble de données a été généré selon les paramètres présentés dans le tableau 2.2.

Table 2.2 Paramètres utilisés pour la création de l'ensemble de données

$U_{\max}$	$\rho$	$q_{12}$	$q_{21}$	$c^+$	$d_m$	$d_a$	$\omega$	$\varphi$	$c^-$
1.75	0.01	0.05	0.25	1	[0.1, 1]	[0.01, 0.1]	$[10^{-3}, 10^{-2}]$	$-\frac{\pi}{2}$	50

### 2.5.1 Les réseaux de neurones profonds (DNN)

Le modèle DNN affiche une dispersion importante des prédictions pour  $z^*$ , avec une erreur moyenne absolue (MAE) de 1,45. Une observation importante est que le nuage de points est particulièrement volumineux pour le graphique de  $z^*$  comparé à celui du coût optimal. Cela s'explique par le fait que, pour chaque combinaison des variables d'entrée ( $d_m, d_a, \omega$ ), le modèle prédit un vecteur de 10 valeurs correspondant aux différents instants de la discrétisation temporelle du taux de demande. En revanche, l'erreur relative moyenne (ERM) pour le coût optimal atteint seulement 0,882 %, montrant une meilleure maîtrise de cette variable agrégée.

La Figure 2.5 présente la comparaison entre les valeurs réelles et prédites de  $z^*$ , tandis que la Figure 2.6 illustre les performances du modèle pour le coût optimal.

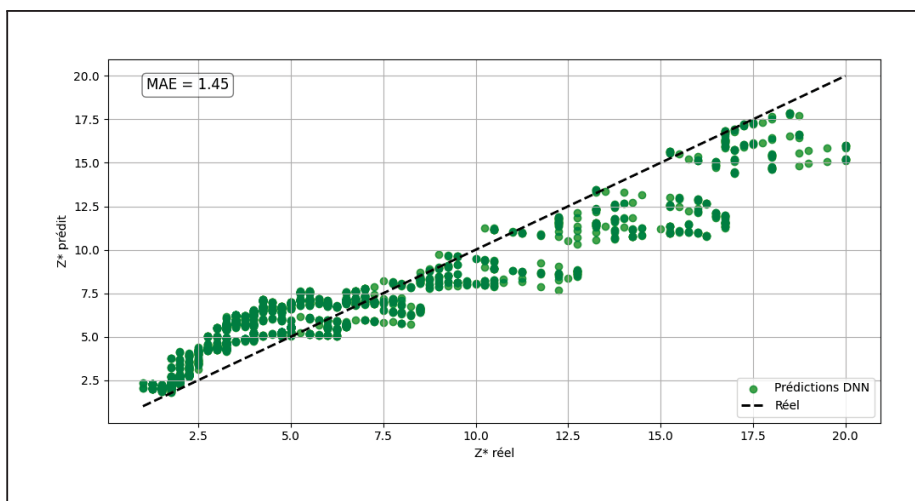


Figure 2.5 Prédictions de  $z^*$  par le modèle DNN

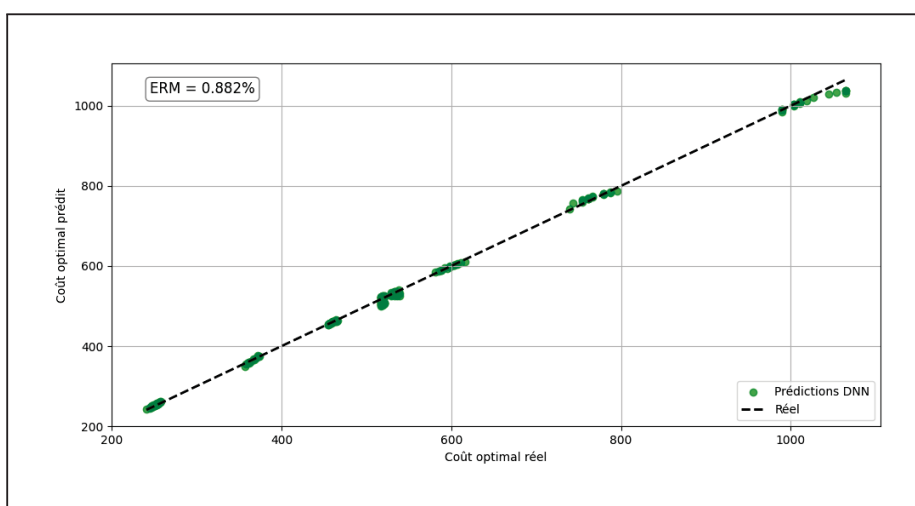


Figure 2.6 Prédictions du coût optimal par le modèle DNN

### 2.5.2 Réseaux de neurones à base radiale (RBF)

Le modèle à base de fonctions radiales (RBF) présente une amélioration marquée des performances par rapport au modèle DNN pour la prédiction du niveau de stock optimal  $z^*$ . L'erreur moyenne absolue (MAE) diminue nettement pour atteindre une valeur de 0,45, et les prédictions sont davantage concentrées autour de la diagonale de référence, traduisant une

meilleure précision globale. Comme pour le modèle DNN, le nuage de points reste relativement étendu, ce qui s'explique par le fait que le modèle prédit, pour chaque combinaison des variables d'entrée ( $d_m, d_a, \omega$ ), un vecteur de 10 valeurs correspondant aux différents instants de la discrétisation temporelle du taux de demande.

En revanche, les performances du modèle RBF sont plus contrastées pour la prédiction du coût optimal. L'erreur relative moyenne (ERM) atteint 4,95 %, ce qui demeure supérieur à celle obtenue avec le DNN. Cette différence suggère que le modèle RBF capture plus finement les variations temporelles locales du stock optimal  $z^*(t)$ , mais qu'il reproduit avec moins de précision la dynamique globale du coût, qui constitue une variable plus agrégée.

La Figure 2.7 illustre les performances du modèle RBF pour la prédiction de  $z^*$ , tandis que la Figure 2.8 présente les résultats obtenus pour le coût optimal.

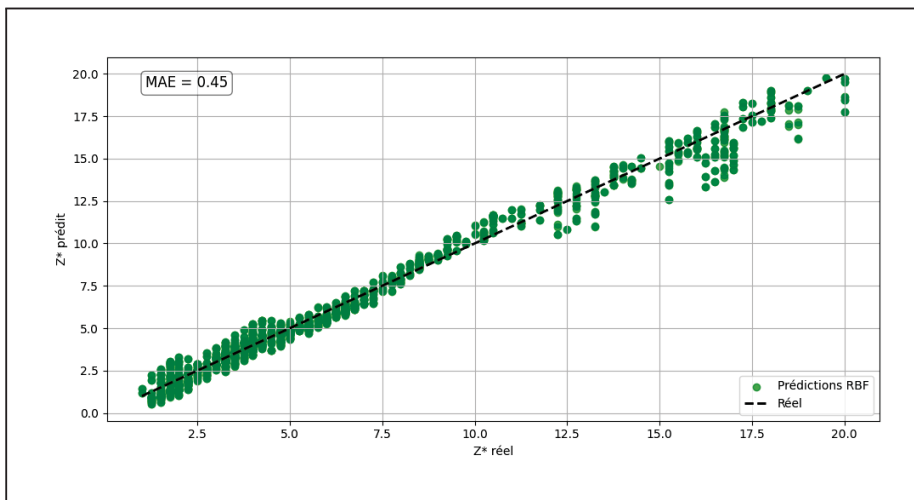


Figure 2.7 Prédications de  $z^*$  par le modèle RBF

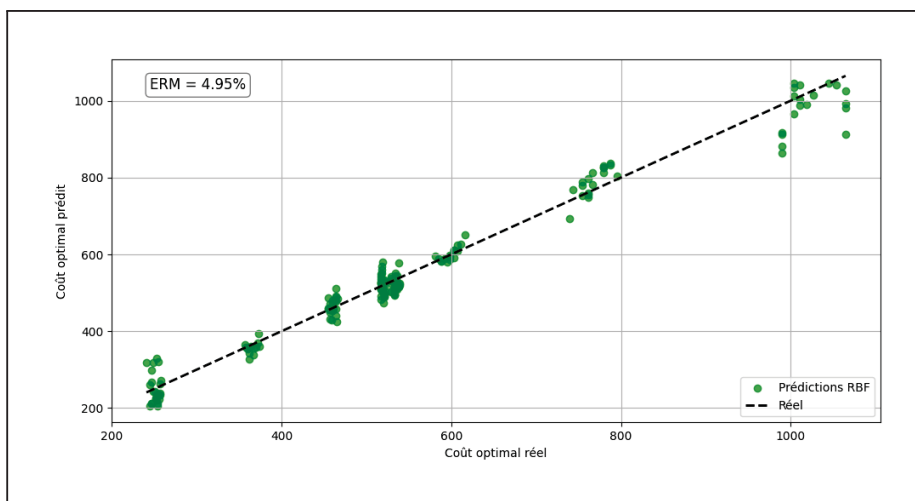


Figure 2.8 Prédictions du coût optimal par le modèle RBF

### 2.5.3 $k$ -Nearest Neighbors (kNN)

L'algorithme kNN présente les meilleures performances parmi les modèles testés. Pour le niveau de stock optimal  $z^*$ , l'erreur moyenne absolue (MAE) n'est que de 0,07, ce qui traduit une prédiction quasiment parfaite du profil optimal. Le nuage de points est très fortement concentré autour de la diagonale de référence, avec une dispersion minimaliste.

Les résultats sont tout aussi convaincants pour le coût optimal. L'erreur relative moyenne (ERM) atteint 0,322 %, ce qui reflète une précision élevée sur cette variable agrégée. Malgré la prédiction, pour chaque combinaison des variables d'entrée  $(d_m, d_a, \omega)$ , d'un vecteur de 10 valeurs correspondant aux différents instants de la discrétisation temporelle de la demande, le modèle kNN parvient à maîtriser efficacement la variabilité temporelle et spatiale des données.

La Figure 2.9 illustre les performances du modèle kNN pour la prédiction de  $z^*$ , tandis que la Figure 2.10 présente les résultats obtenus pour le coût optimal.

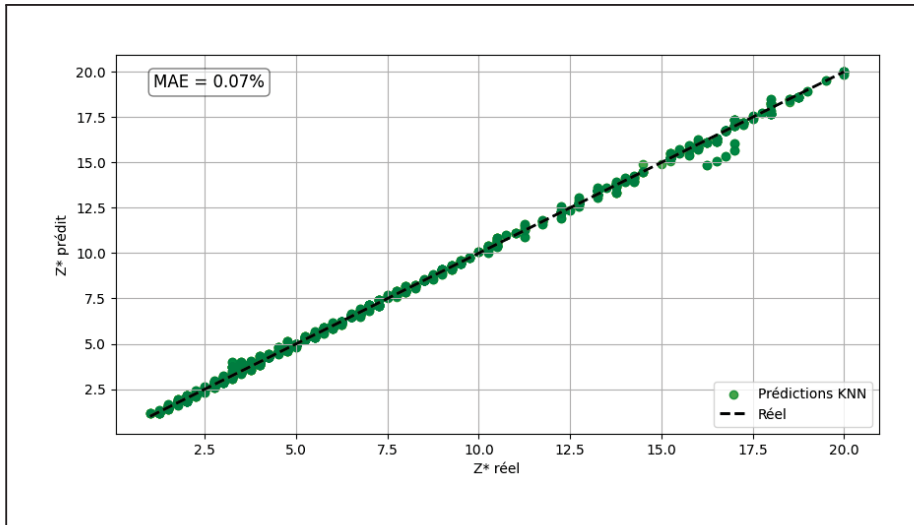


Figure 2.9 Prédictions de  $z^*$  par le modèle kNN

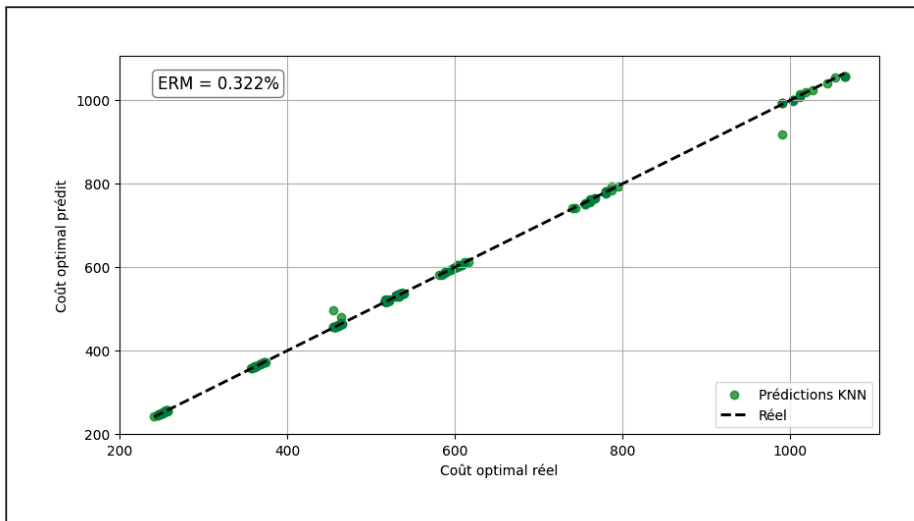


Figure 2.10 Prédictions du coût optimal par le modèle kNN

Le tableau 2.3 synthétise les performances des trois algorithmes testés lors de la phase 1.

Table 2.3 Comparatif des performances des modèles ML (MAE pour  $Z^*$  et erreur relative pour  $C$ )

Modèle de ML	MAE $Z^*$	Erreur relative $C$
MLP	0,712	1,45%
RBF	0,407	4,35%
kNN	0,07	0,32%

Le kNN s'avère être l'algorithme le plus performant, combinant une excellente précision sur les deux paramètres optimaux à prédire. Avec une MAE de 0,07 pour  $z^*$  et une ERM de 0,322% pour le coût optimal, il surpasse significativement le RBF et le DNN. Cette supériorité justifie le choix du kNN comme algorithme de référence pour les phases ultérieures de notre étude.

#### 2.5.4 Analyses de sensibilité

L'analyse de sensibilité examine comment les variations des paramètres d'entrée influencent la trajectoire optimale du stock et le coût global du système. Cette investigation valide la capacité du modèle kNN à anticiper les ajustements nécessaires face à des changements des conditions opérationnelles.

##### 2.5.4.1 Variation du taux moyen de demande

Sur le plan théorique, une augmentation de  $d_m$  doit se traduire par une élévation du niveau de stock optimal, afin de satisfaire une demande moyenne plus élevée. Cela conduit à une augmentation du volume de production et, par conséquent, du coût global du système. Le modèle kNN reproduit ce comportement de manière cohérente : lorsque  $d_m$  augmente, la trajectoire  $z^*(t)$  se déplace vers le haut tout en conservant sa structure temporelle, ce qui confirme la présence de l'effet d'anticipation mis en évidence précédemment. La Figure 2.11 illustre l'évolution de  $z^*(t)$  pour différents niveaux de  $d_m$ . Les valeurs de l'amplitude et de la fréquence sont  $d_a = 0,09$  et  $\omega = 0,005$ .

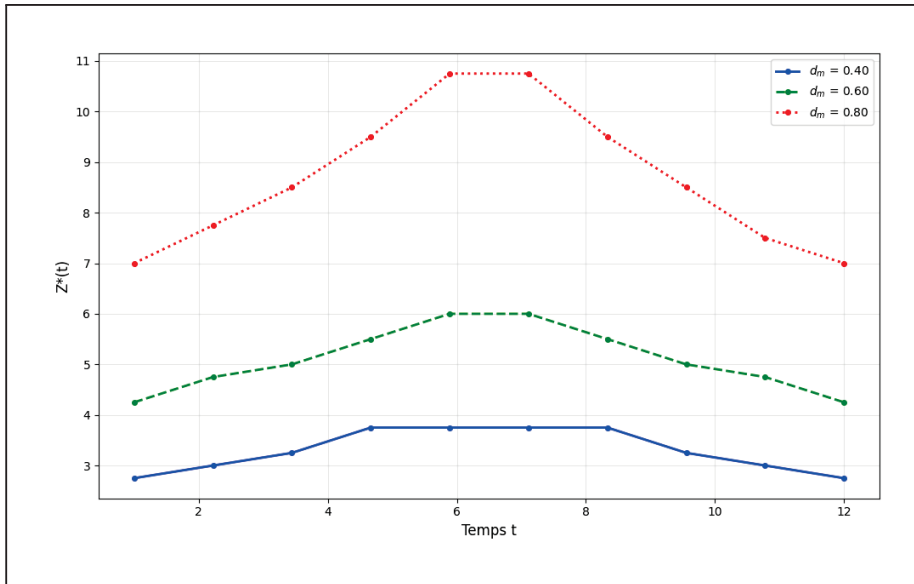


Figure 2.11 Trajectoire de  $z^*(t)$  pour différents niveaux de  $d_m$

Pour le coût optimal, le modèle capture également l'augmentation attendue lorsque  $d_m$  croît. La croissance du coût devient plus marquée pour les valeurs élevées de la demande moyenne, ce qui reflète l'impact combiné d'une production accrue et d'un risque plus important de rupture de stock. La Figure 2.12 présente la variation du coût optimal en fonction de  $d_m$  pour les mêmes paramètres  $d_a$  et  $\omega$ .

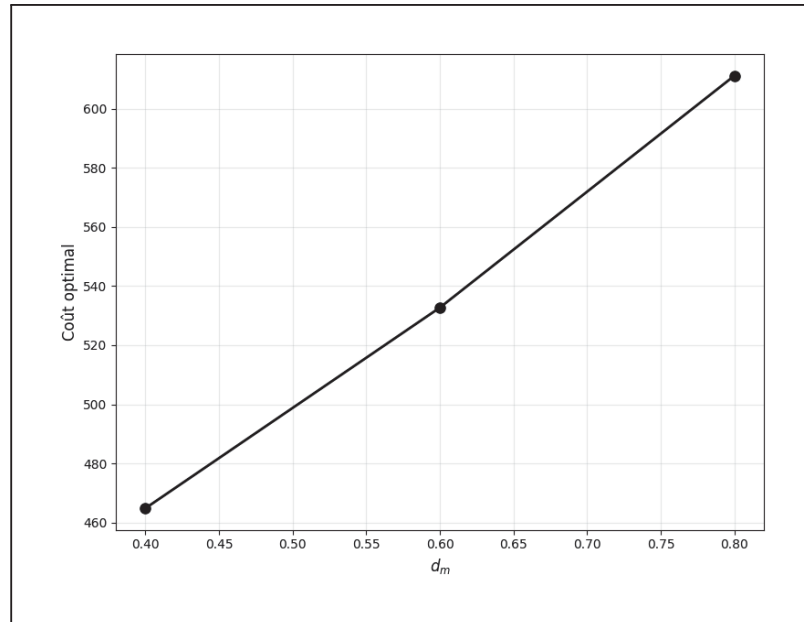


Figure 2.12 Coût optimal en fonction de  $d_m$

#### 2.5.4.2 Variation de l'amplitude du taux de demande

L'augmentation de l'amplitude  $d_a$  intensifie les oscillations de la demande autour de sa moyenne. Contrairement au paramètre  $d_m$ , qui agit directement sur le niveau global de la demande,  $d_a$  modifie surtout sa variabilité. Le modèle kNN restitue bien cet effet : lorsque  $d_a$  augmente, la trajectoire  $z^*(t)$  se relève progressivement tout en conservant sa structure temporelle, ce qui traduit une augmentation du niveau de stock nécessaire pour absorber des fluctuations plus marquées de la demande. La Figure 2.13 illustre l'évolution de  $z^*(t)$  pour différents niveaux de  $d_a$ . Les valeurs de la moyenne et de la fréquence sont : ( $d_m = 1,0$  et  $\omega = 0,005$ )

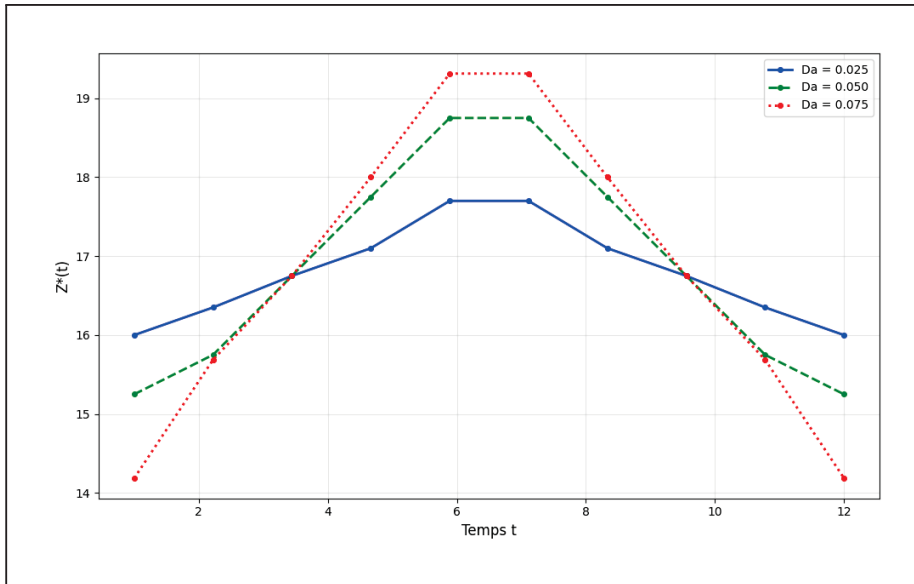


Figure 2.13 Trajectoire de  $z^*(t)$  pour différents niveaux de  $d_a$

L'impact sur le coût global reste toutefois plus modéré que dans le cas d'une variation de  $d_m$ . L'augmentation de  $d_a$  renforce surtout la volatilité de la demande plutôt que son volume moyen, de sorte que l'accroissement du coût reflète davantage le risque accru de rupture et de surstock que une hausse systématique des quantités produites. La Figure 2.14 présente la variation du coût optimal en fonction de  $d_a$  pour les mêmes valeurs de  $d_m$  et  $\omega$ .

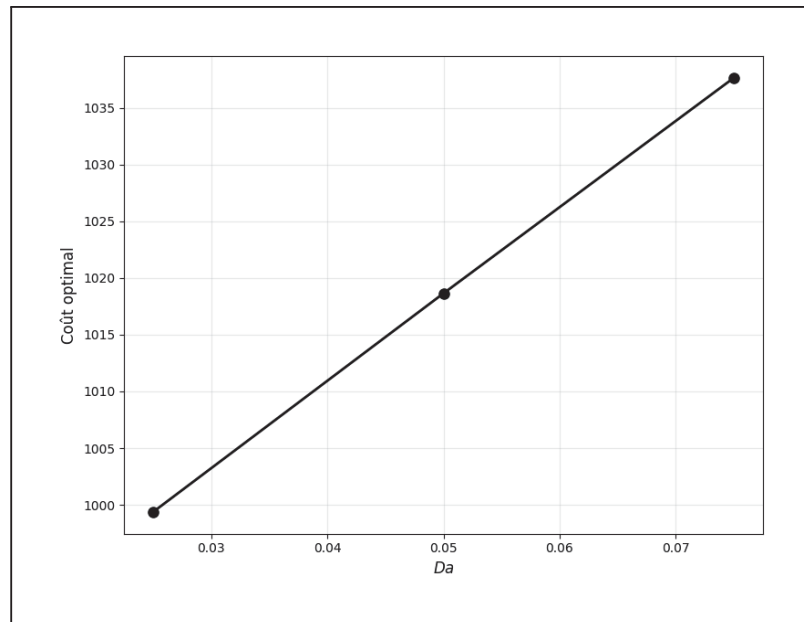


Figure 2.14 Coût optimal en fonction de  $d_a$

### 2.5.4.3 Variation de la fréquence du taux de demande

Le paramètre  $\omega$  contrôle la fréquence des variations du taux de demande. Les résultats numériques montrent que, dans l'intervalle testé, ce paramètre n'a pratiquement aucun impact sur les paramètres optimaux. La trajectoire  $z^*(t)$  reste quasi identique pour toutes les valeurs de  $\omega$  considérées, ce qui indique que le profil optimal de stock est principalement gouverné par le niveau de demande instantané plutôt que par la vitesse à laquelle le cycle est parcouru. La Figure 2.15 illustre cette insensibilité de  $z^*(t)$  vis-à-vis de  $\omega$  avec les valeurs de la moyenne et de l'amplitude qui sont de : ( $d_m = 1,0$  et  $d_a = 0,09$ ).

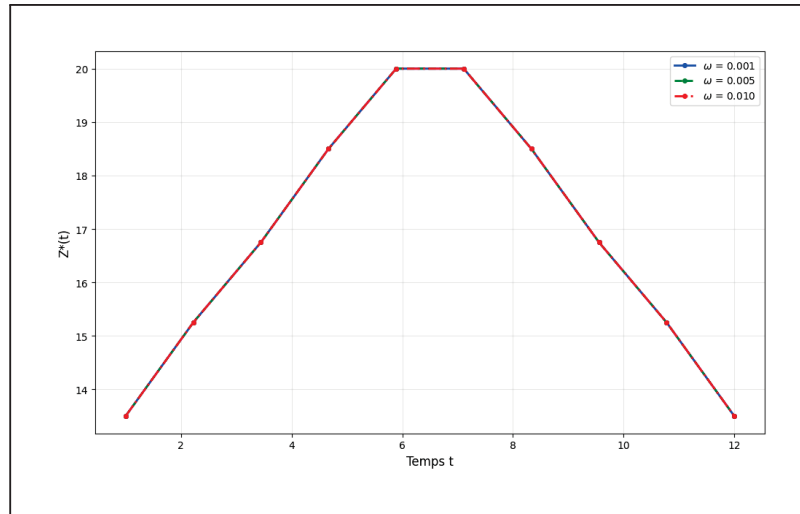


Figure 2.15 Trajectoire de  $z^*(t)$  pour différentes valeurs de  $\omega$

Le paramètre  $\omega$  détermine la période

$$T = \frac{2\pi}{\omega}$$

de la demande  $d(t) = d_m + d_a \cos(\omega t + \varphi)$ . Dans la plage de valeurs étudiée, la variation de  $\omega$  modifie essentiellement l'échelle temporelle (compression ou dilatation du cycle), tandis que l'ensemble des niveaux de demande visités au cours du cycle reste strictement déterminé par  $(d_m, d_a)$ . Le coût optimal reflète cette propriété : il demeure pratiquement constant lorsque  $\omega$  varie, comme le montre la Figure 2.16.

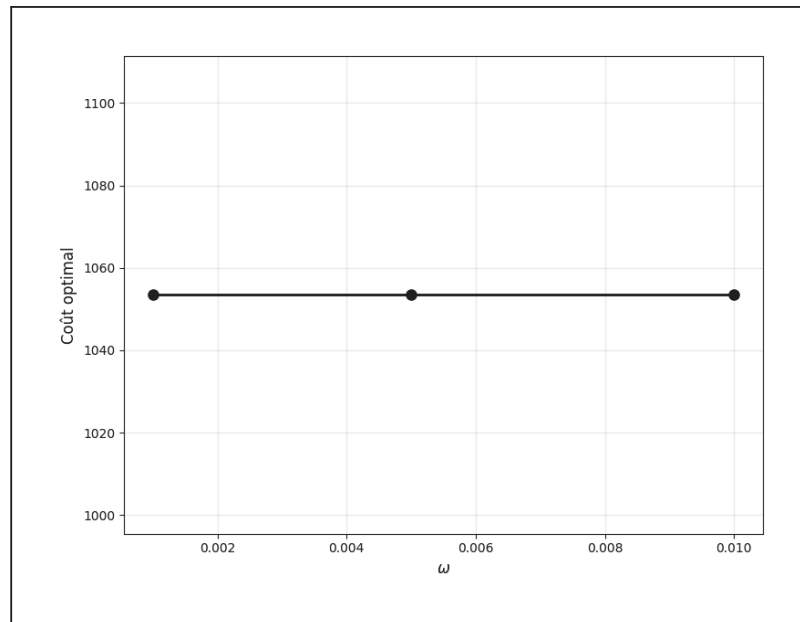


Figure 2.16 Coût optimal pour différentes valeurs de  $\omega$

On en conclut que, dans ce cadre,  $\omega$  n'apparaît pas comme un paramètre explicatif majeur des politiques optimales et du coût associé. Il est essentiellement conservé pour projeter le profil de demande sur l'axe temporel lors de la représentation de  $z^*(t)$ , tandis que les effets significatifs sur les décisions de production sont principalement induits par  $d_m$  et  $d_a$ .

### 2.5.5 Synthèse des résultats et gain computationnel

Les expériences menées dans cette section démontrent la pertinence de l'approche par apprentissage automatique pour prédire les paramètres optimaux d'un système de production soumis à une demande périodique. Trois constats majeurs émergent de cette analyse comparative.

Premièrement, le modèle kNN s'impose comme l'algorithme le plus performant parmi les trois approches testées. Avec une MAE de 0,07 pour  $Z^*$  et une erreur relative de 0,322% sur le coût optimal, il surpasse significativement les réseaux de neurones profonds (DNN) et les réseaux à base radiale (RBF). Cette supériorité s'explique par la capacité du kNN à capturer efficacement

les relations non linéaires entre les paramètres d'entrée et les sorties optimales, sans nécessiter d'hypothèses structurelles complexes sur la forme de ces relations.

Deuxièmement, les analyses de sensibilité valident la cohérence physique et économique du modèle. Les variations de  $d_m$  et  $d_a$  produisent les effets attendus sur les trajectoires optimales et les coûts : une augmentation du taux moyen de demande élève proportionnellement le stock optimal et le coût total, tandis qu'une amplitude accrue intensifie la variabilité du stock sans impact majeur sur le coût moyen. L'insensibilité constatée vis-à-vis de  $\omega$  confirme que la politique optimale dépend essentiellement du niveau et de l'amplitude de la demande, et non de sa fréquence de variation. Ces résultats démontrent que le modèle kNN ne se contente pas d'interpoler les données, mais restitue fidèlement la logique économique sous-jacente au problème de contrôle optimal.

Troisièmement, et c'est là un apport décisif de l'approche, le gain computationnel est considérable. Les temps d'exécution ont été mesurés en temps mur (wall-clock) dans un environnement standard Google Colab (CPU). Dans ces conditions, la résolution numérique des équations HJB requiert typiquement 3 à 5 minutes par configuration, tandis que le modèle kNN fournit une prédiction quasi instantanée (moins d'une seconde). Le gain obtenu est ainsi de l'ordre de 180 à 300 fois, ce qui change l'échelle de l'analyse de scénarios : là où l'approche HJB classique limite l'exploration à quelques configurations soigneusement sélectionnées, le modèle kNN permet d'évaluer rapidement des centaines, voire des milliers de combinaisons paramétriques. Cette flexibilité est particulièrement précieuse dans un contexte industriel, où les paramètres opérationnels évoluent fréquemment et où la rapidité de décision constitue un avantage compétitif.

En combinant précision prédictive, cohérence théorique et efficacité computationnelle, l'approche par kNN se positionne comme une alternative crédible et opérationnelle aux méthodes numériques classiques pour le contrôle optimal des systèmes de production sous demande variable. Les résultats obtenus justifient pleinement son extension à des configurations plus complexes, intégrant des paramètres économiques et de fiabilité supplémentaires, comme nous le verrons dans la section suivante.

## 2.6 Considération de la variation du coût de rupture de stock et du taux de réparation

Cette phase enrichit l'expérience précédente en intégrant progressivement deux paramètres critiques du système de production : le coût de rupture de stock ( $c^-$ ) et le taux de réparation ( $q_{21}$ ). L'objectif est de valider que le modèle kNN maintient ses excellentes performances lorsque ces dimensions opérationnelles sont intégrées dans la prédiction des paramètres optimaux. Ce travail teste également la robustesse du modèle face à une complexité accrue du problème d'optimisation.

Cette expérience prolonge l'analyse précédente en intégrant le coût de rupture  $c^-$  et le taux de réparation  $q_{21}$  comme variables d'entrée supplémentaires. Les paramètres qui varient dans cette expérience sont présentés au Tableau 2.4. Tous les autres paramètres ( $U_{\max}$ ,  $\rho$ ,  $q_{12}$  et  $\varphi$ ) sont conservés identiques à ceux utilisés dans l'Expérience 1 (Tableau 2.2).

Table 2.4 Paramètres variables utilisés pour la génération de l'ensemble de données dans l'Expérience 2

$d_m$	$d_a$	$q_{21}$	$c^-$
[0.6, 0.95]	[0.05, 0.1]	[0.15, 0.35]	[20, 100]

Les quatre paramètres sont combinés afin de générer un large éventail de configurations pour l'entraînement et l'évaluation.

### 2.6.1 Analyse globale des prédictions

Cette section présente une vision synthétique des performances du modèle kNN sur l'ensemble des configurations testées. Chaque point des nuages de points correspond à une combinaison de paramètres ( $d_m, d_a, \omega$ ), tandis que la droite en pointillés représente une prédiction parfaite (*prédit = réel*).

La Figure 2.17 compare les valeurs réelles et prédites de la moyenne temporelle de  $Z^*(t)$ . Les points sont fortement alignés le long de la diagonale, avec une erreur absolue moyenne (MAE) de 0,391, ce qui indique que le modèle restitue fidèlement le niveau moyen de stock optimal sur l'horizon considéré.

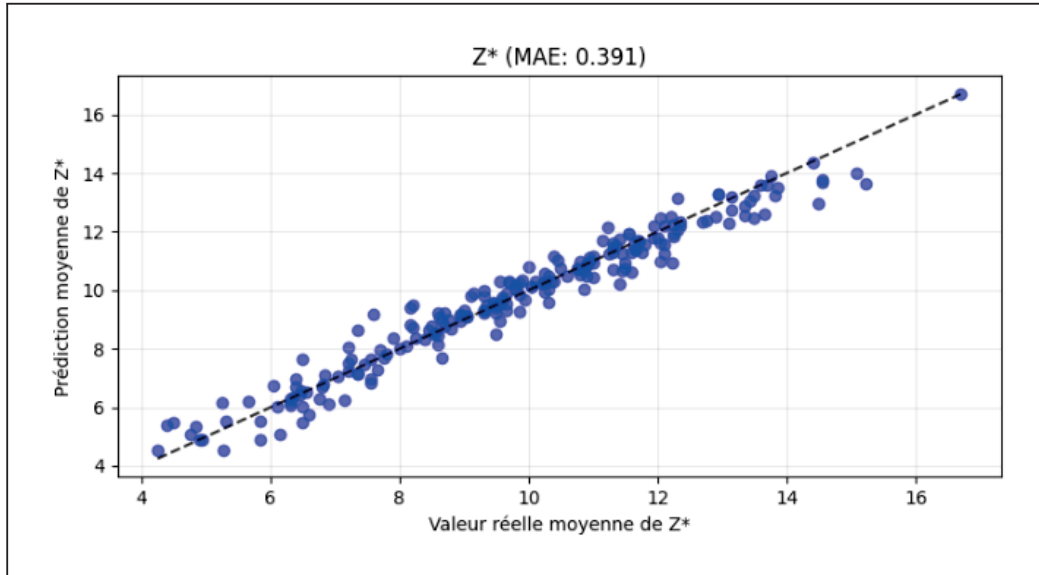


Figure 2.17 Comparaison entre la moyenne réelle et prédite de  $Z^*(t)$  (kNN, expérience 2)

La Figure 2.18 illustre la qualité des prédictions pour le coût optimal  $C$ . Malgré une plage de variation importante des coûts, les points restent concentrés autour de la diagonale, avec une erreur relative moyenne (ERM) de 4,216 %. L'absence de biais systématique marqué suggère que le modèle demeure fiable sur l'ensemble des configurations économiques explorées.

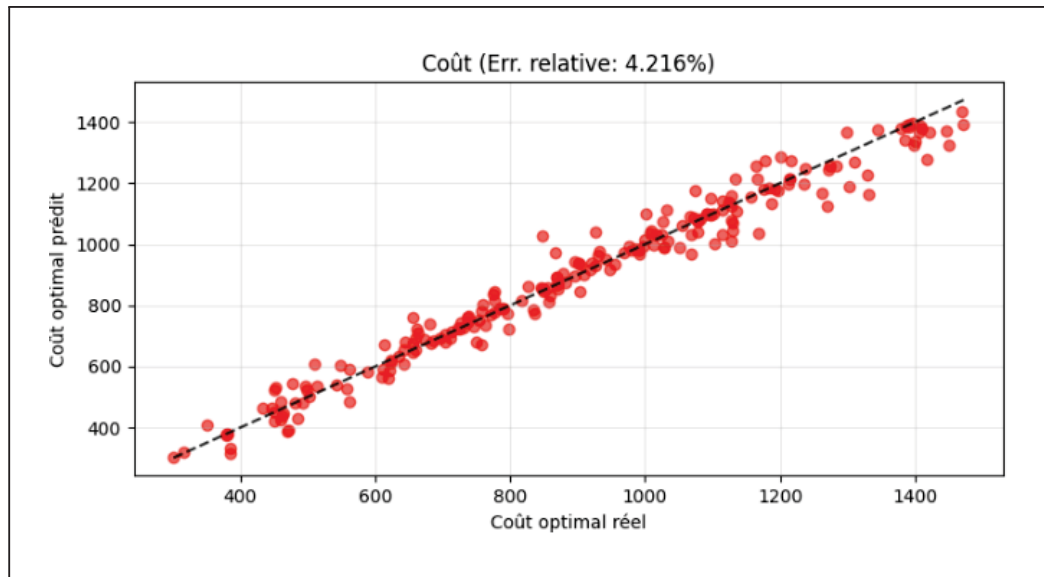


Figure 2.18 Comparaison entre le coût optimal réel et prédit (kNN, expérience 2)

Dans l'ensemble, ces résultats confirment que le kNN reproduit avec une grande précision les solutions issues des équations HJB, tout en offrant un temps de calcul quasi instantané. Il constitue ainsi un outil prédictif robuste pour l'approximation des paramètres optimaux dans un contexte où la résolution numérique directe des HJB serait trop coûteuse.

## 2.6.2 Validation sur scénarios représentatifs

Cette section évalue la performance du modèle kNN à travers une sélection de scénarios représentatifs, en comparant ses prédictions aux solutions de référence obtenues par résolution numérique des équations d'Hamilton-Jacobi-Bellman.

La démarche s'articule en deux temps. Premièrement, nous examinons en détail un scénario particulier, illustrant comment le modèle restitue les trajectoires optimales  $z^*(t)$  et les coûts associés. Deuxièmement, l'analyse est étendue à 10 configurations différentes, obtenues par variation des paramètres  $d_m$ ,  $d_a$ ,  $c^-$  et  $q_{21}$ . Les résultats sont synthétisés sous forme de tableau récapitulatif, reportant pour chaque scénario la MAE sur  $z^*$ , les coûts optimaux HJB et prédits,

ainsi que l'erreur relative correspondante. Cette double approche offre une vision complète des performances du modèle kNN pour la prédiction des politiques optimales.

### 2.6.2.1 Scénario particulier

Dans ce scénario, la demande moyenne est fixée à  $d_m = 0,94$  avec une amplitude marquée  $d_a = 0,10$ . Le coût unitaire de rupture ( $c^- = 81$ ) est élevé, ce qui incite à maintenir un stock de sécurité important. Le taux de réparation  $q_{21} = 0,32$  traduit une disponibilité intermédiaire de la machine.

La Figure 2.19 compare, pour ce jeu de paramètres, la trajectoire optimale  $Z^*(t)$  obtenue par la méthode HJB à celle prédite par le modèle kNN. Les deux courbes présentent une dynamique très proche, avec un suivi précis des phases de montée, de plateau et de descente du stock. On observe uniquement une légère surestimation de  $Z^*(t)$  par le modèle kNN autour du niveau maximal, sans modification qualitative de la forme de la trajectoire.

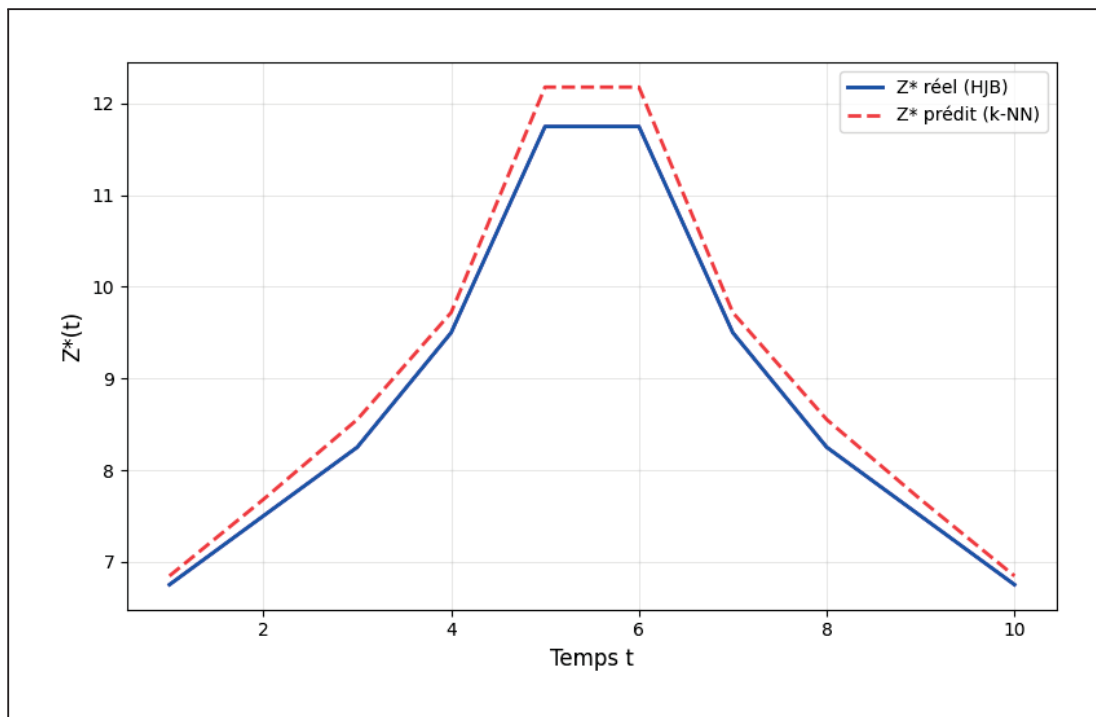


Figure 2.19 Comparaison des trajectoires optimales  $Z^*(t)$  issues des équations HJB et du modèle kNN (scénario particulier)

Sur le plan économique, les coûts obtenus sont

$$C_{\text{HJB}} = 524,38, \quad C_{\text{kNN}} = 537,10, \quad (2.26)$$

soit une erreur relative d'environ

$$\frac{|524,38 - 537,10|}{524,38} \times 100 \approx 2,42 \% \quad (2.27)$$

Cet écart limité constitue une validation importante : le modèle kNN fournit une approximation rapide et fiable des paramètres optimaux, tout en évitant la lourdeur computationnelle des méthodes numériques basées sur les équations HJB et en maintenant un niveau de précision satisfaisant pour l'aide à la décision.

### **2.6.2.2 Validation du modèle kNN**

Afin d'illustrer la performance du modèle kNN de façon plus générale, l'analyse est ensuite étendue à un ensemble de 10 scénarios représentatifs, sélectionnés parmi 50 scénarios testés (voir annexe I), générés aléatoirement par interpolation des paramètres d'entrée au sein de l'espace couvert par l'ensemble de test.

Table 2.5 Performances du modèle kNN sur 10 scénarios : MAE sur  $z^*$ , coûts optimaux HJB et prédits, ainsi que l'erreur relative

$d_m$	$d_a$	$c^-$	$q_{21}$	MAE_ $z^*$	Coût_réel	Coût_prédit	ERM(%)
0,69	0,05	30	0,15	0,62	1123,03	1136,63	1,21
0,74	0,07	61	0,16	0,33	1384,93	1339,05	3,31
0,78	0,06	91	0,20	0,00	1182,96	1182,96	0,00
0,82	0,08	40	0,20	0,31	751,58	746,42	0,69
0,85	0,09	31	0,27	0,00	334,88	334,88	0,00
0,88	0,09	21	0,25	0,00	321,48	321,48	0,00
0,89	0,10	80	0,20	0,27	1043,64	1027,39	1,56
0,93	0,07	40	0,29	0,15	445,80	442,90	0,65
0,95	0,09	40	0,26	0,21	431,67	435,00	0,77
0,94	0,08	80	0,25	0,18	609,58	604,57	0,82

Le tableau 2.5 met en évidence les excellentes performances du modèle kNN sur l'ensemble des 10 scénarios testés. L'erreur relative moyenne (ERM) sur le coût optimal s'établit à 1,29%, reflétant une précision remarquable.

L'analyse révèle également que le MAE sur  $z^*$  reste modéré, ne dépassant jamais 0,62 sur l'ensemble des scénarios. Cette stabilité confirme que le modèle kNN maintient sa robustesse face à la variabilité des paramètres d'entrée, y compris pour les valeurs basses de  $d_m$  (0,69 et 0,74) et les coûts de rupture élevés ( $c^- \geq 80$ ).

Ces résultats valident que le kNN constitue une alternative efficace et fiable pour l'approximation rapide des paramètres optimaux de la politique de production dans un contexte où la complexité opérationnelle s'accroît, notamment avec l'intégration du coût de rupture et du taux de réparation comme variables d'entrée.

### 2.6.3 Analyses de sensibilité

Cette partie approfondit l'évaluation du modèle kNN en examinant sa sensibilité à deux paramètres clés du système : le coût unitaire de rupture de stock ( $c^-$ ) et le taux de réparation de la machine ( $q_{21}$ ). L'étude consiste à observer l'évolution des politiques optimales de gestion des stocks et du coût global associé lorsque ces variables sont modifiées, tout en maintenant constantes les autres caractéristiques du système, notamment les paramètres du taux de demande.

Cette approche permet de valider la capacité du modèle à reproduire les effets attendus des variations de ces paramètres opérationnels, renforçant ainsi la confiance dans ses performances en contexte industriel complexe.

#### 2.6.3.1 Analyse de sensibilité par rapport au taux de réparation

L'analyse de sensibilité du taux de réparation  $q_{21}$  est menée en faisant varier ce paramètre dans l'intervalle  $[0,20, 0,30]$ , tandis que les autres caractéristiques du système sont maintenues constantes ( $d_m = 0,94$ ,  $d_a = 0,06$ ,  $c^- = 41$ ). Ce cadre permet d'isoler l'impact de  $q_{21}$  sur la politique de production optimale et sur le coût global, tout en évaluant la robustesse du modèle kNN.

La Figure 2.20 illustre l'effet de  $q_{21}$  sur la trajectoire optimale  $z^*(t)$ . Un comportement conforme à l'intuition est observé : lorsque le taux de réparation augmente, les niveaux optimaux de stock diminuent. Un système plus facilement réparable requiert en effet une politique de stockage moins conservatrice, les interruptions de production étant plus courtes et moins pénalisantes.

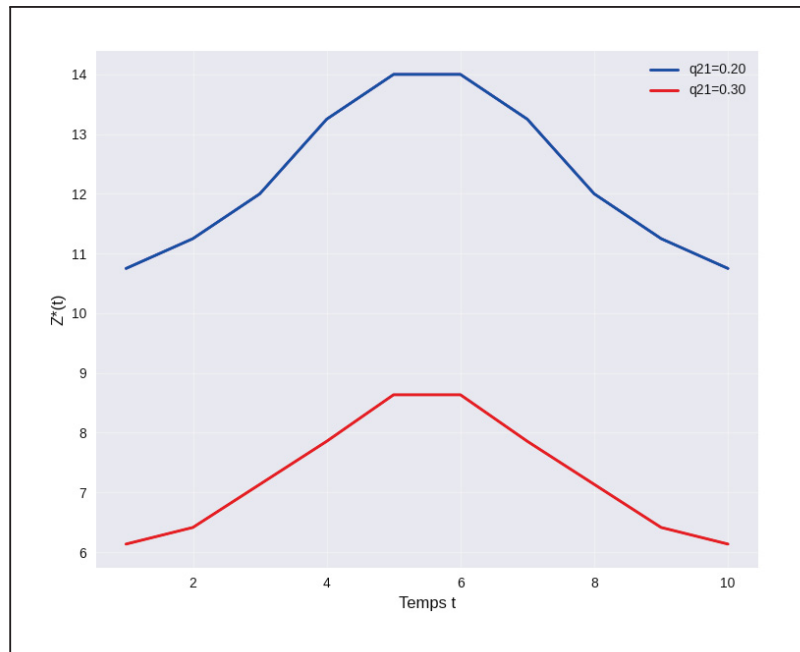


Figure 2.20 Trajectoire optimale  $z^*(t)$  pour différents niveaux de  $q_{21}$

La Figure 2.21 présente l'évolution du coût optimal en fonction de  $q_{21}$ , en comparant les valeurs issues des équations HJB et les prédictions du modèle kNN. Sur l'ensemble de l'intervalle étudié, les courbes restent très proches, ce qui traduit une bonne capacité du modèle à reproduire l'impact de  $q_{21}$  sur la performance économique du système.

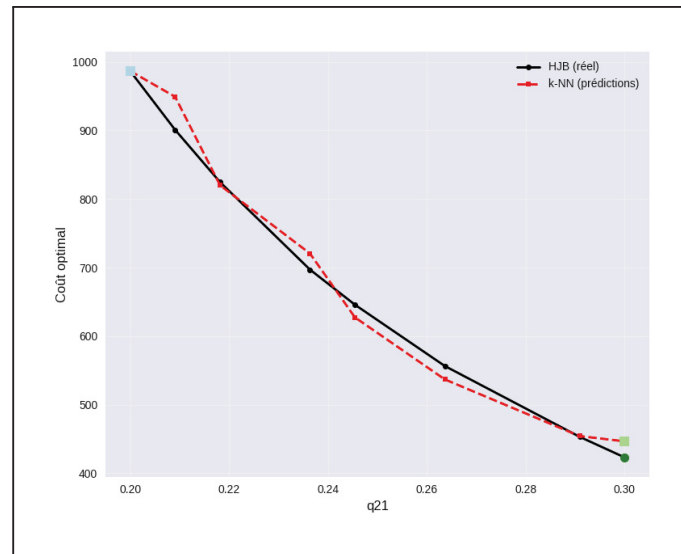


Figure 2.21 Coût optimal en fonction de  $q_{21}$  : comparaison HJB / kNN

Cette observation est confirmée par l'analyse de la variation de l'erreur relative le long de l'intervalle, représentée à la Figure 2.22. La ligne horizontale en pointillés indique une erreur relative moyenne d'environ 2,7 % sur l'intervalle  $[0,20, 0,30]$ . La plupart des points de test se situent au-dessous de 4 %, seuls quelques pics isolés atteignant des valeurs comprises entre 5 et 6 %. Globalement, le profil d'erreur observé confirme que le modèle kNN constitue un substitut fiable pour l'exploration paramétrique rapide du système vis-à-vis du taux de réparation  $q_{21}$ , sans qu'il soit nécessaire de répéter systématiquement les résolutions numériques des équations HJB.

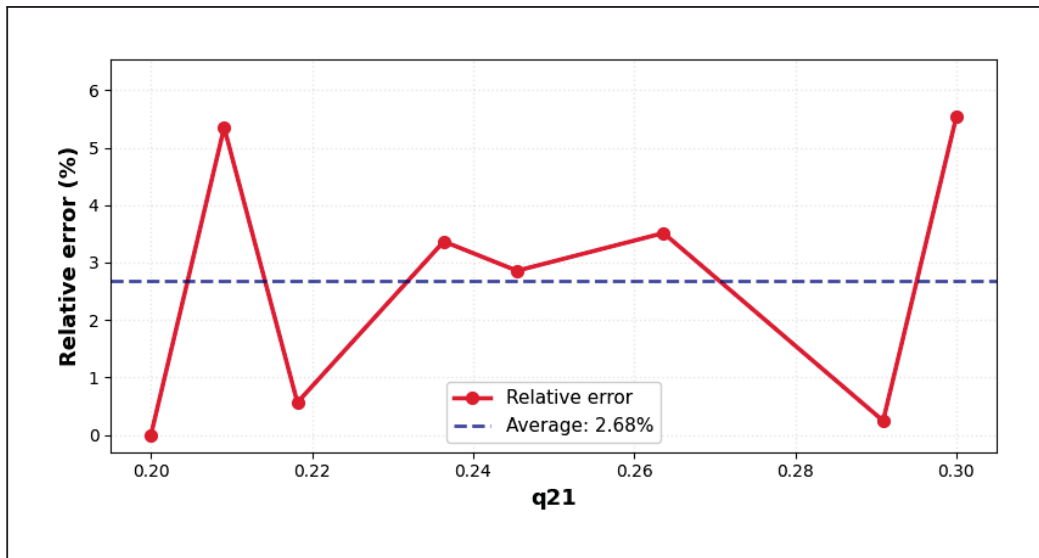


Figure 2.22 Erreur relative sur le coût optimal en fonction du taux de réparation  $q_{21}$

### 2.6.3.2 Analyse de sensibilité par rapport au coût de rupture de stock

L'analyse de sensibilité au coût unitaire de rupture de stock  $c^-$  est conduite en faisant varier ce paramètre dans l'intervalle  $[25, 50]$ , tandis que les autres paramètres du système sont maintenus constants ( $d_m = 0,82$ ,  $d_a = 0,09$ ,  $q_{21} = 0,22$ ). Ce cadre permet d'isoler l'effet de  $c^-$  sur la politique de production optimale et sur le coût total, tout en évaluant la robustesse du modèle kNN.

La Figure 2.23 illustre l'impact de  $c^-$  sur la trajectoire optimale  $z^*(t)$ . Un comportement économiquement attendu est observé : lorsque  $c^-$  augmente, les niveaux optimaux de stock s'élèvent, traduisant une politique plus conservatrice visant à limiter des ruptures devenues plus pénalisantes. Le modèle kNN reproduit fidèlement cette tendance sur l'ensemble de l'intervalle étudié.

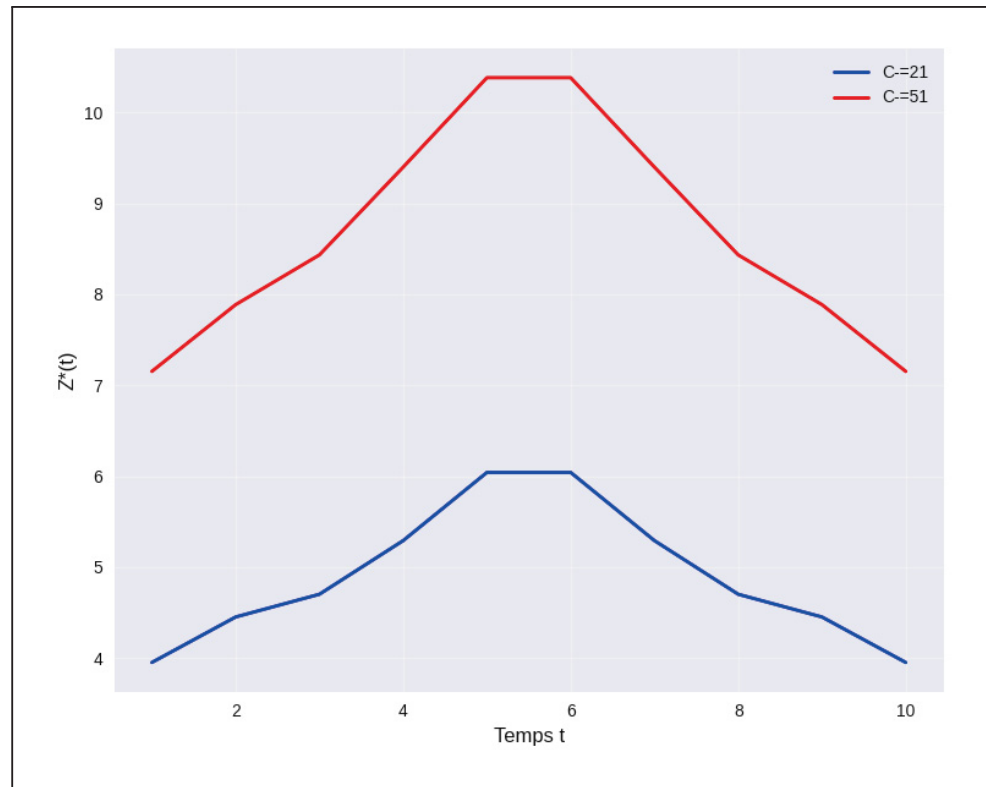


Figure 2.23 Trajectoire optimale  $z^*(t)$  pour différents niveaux de  $c^-$

La Figure 2.24 présente l'évolution du coût optimal en fonction de  $c^-$ , en comparant les valeurs obtenues par la méthode HJB et les prédictions du modèle kNN. Sur l'intervalle  $[25, 50]$ , les deux courbes restent très proches, ce qui se traduit par une erreur relative moyenne d'environ  $\bar{\varepsilon} \approx 1,7\%$ . Ces résultats confirment que le modèle kNN est capable de suivre correctement l'impact des variations du coût de rupture sur la performance économique du système, tout en conservant un temps de calcul très réduit.

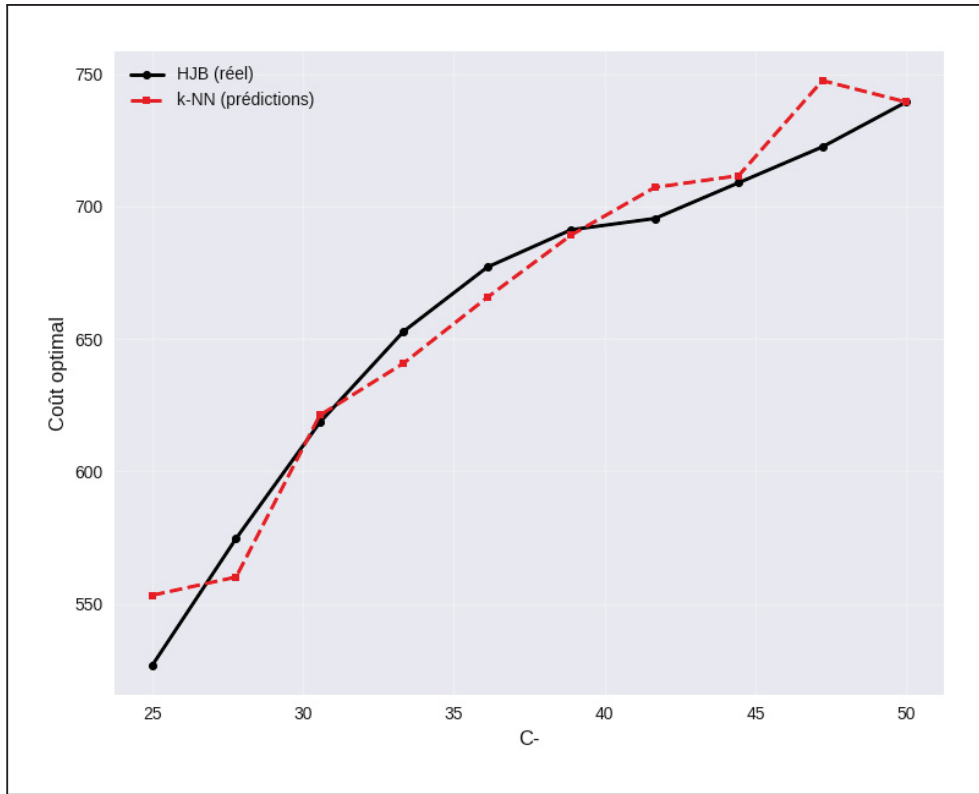


Figure 2.24 Coût optimal en fonction de  $c^-$  : comparaison HJB / kNN

La variation de l'erreur relative sur le coût optimal le long de l'intervalle de sensibilité est représentée à la Figure 2.25. La ligne horizontale en pointillés indique une erreur relative moyenne d'environ 1,7 %. La plupart des points de sensibilité se situent en dessous de 2 %, avec un seul écart plus marqué atteignant environ 5 % à l'extrémité inférieure de la plage de  $c^-$ . Dans l'ensemble, le profil d'erreur observé confirme que le modèle kNN reste à la fois stable et précis sur tout l'intervalle considéré, et qu'il constitue un surrogat robuste pour l'exploration rapide de scénarios impliquant des variations du coût de rupture.

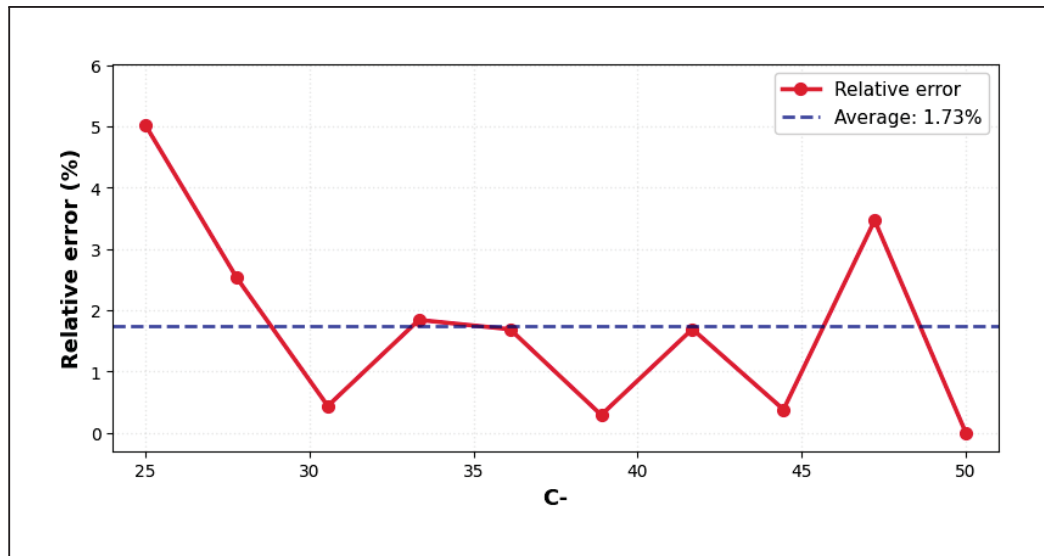


Figure 2.25 Erreur relative sur le coût optimal en fonction du coût de rupture  $c^-$

#### 2.6.4 Implémentation de la politique optimale

Une fois les paramètres optimaux obtenus via le modèle kNN, le gestionnaire doit traduire ces prédictions en décisions concrètes de production. Cette section illustre le processus décisionnel à travers un exemple numérique.

En fournissant le quadruplet  $(d_m, d_a, c^-, q_{21}) = (0.75, 0.06, 20, 0.15)$  au modèle kNN (les autres paramètres étant ceux du tableau 2.4), le gestionnaire obtient instantanément la trajectoire optimale  $Z^*(t)$  illustrée par la figure 2.26 ainsi que le coût optimal  $C = 935.95$  \$.

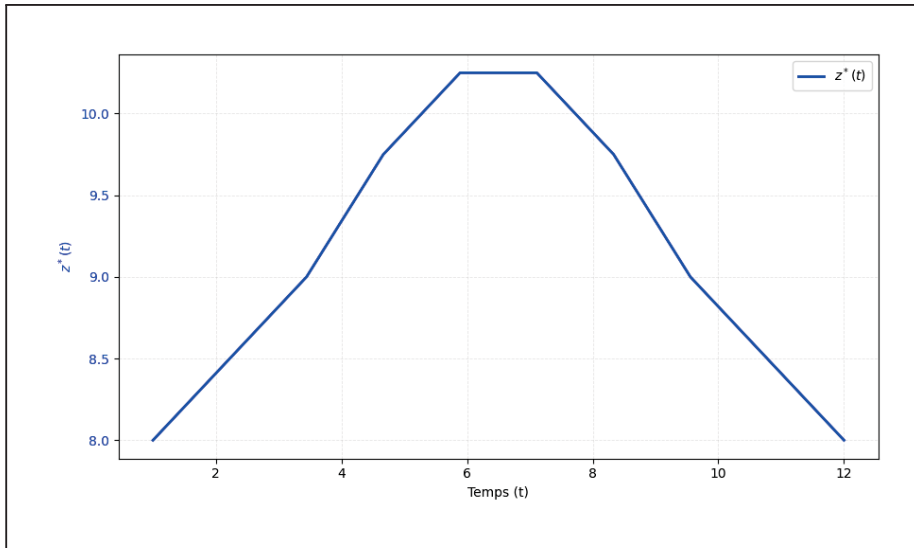


Figure 2.26 Évolution de la trajectoire optimale  $Z^*(t)$

Considérons maintenant l'instant  $t_4$  correspondant au quatrième point de discrétisation temporelle, où le taux de demande instantanée est  $d(t_4) \approx 0.81$  et le seuil critique prédit est  $Z^*(t_4) = 9.8$  unités. La figure 2.27 illustre l'algorithme décisionnel appliqué à cet instant, en supposant que la machine est opérationnelle ( $\alpha = 1$ ) et que le stock mesuré est  $x(t_4) = 9.2$  unités.

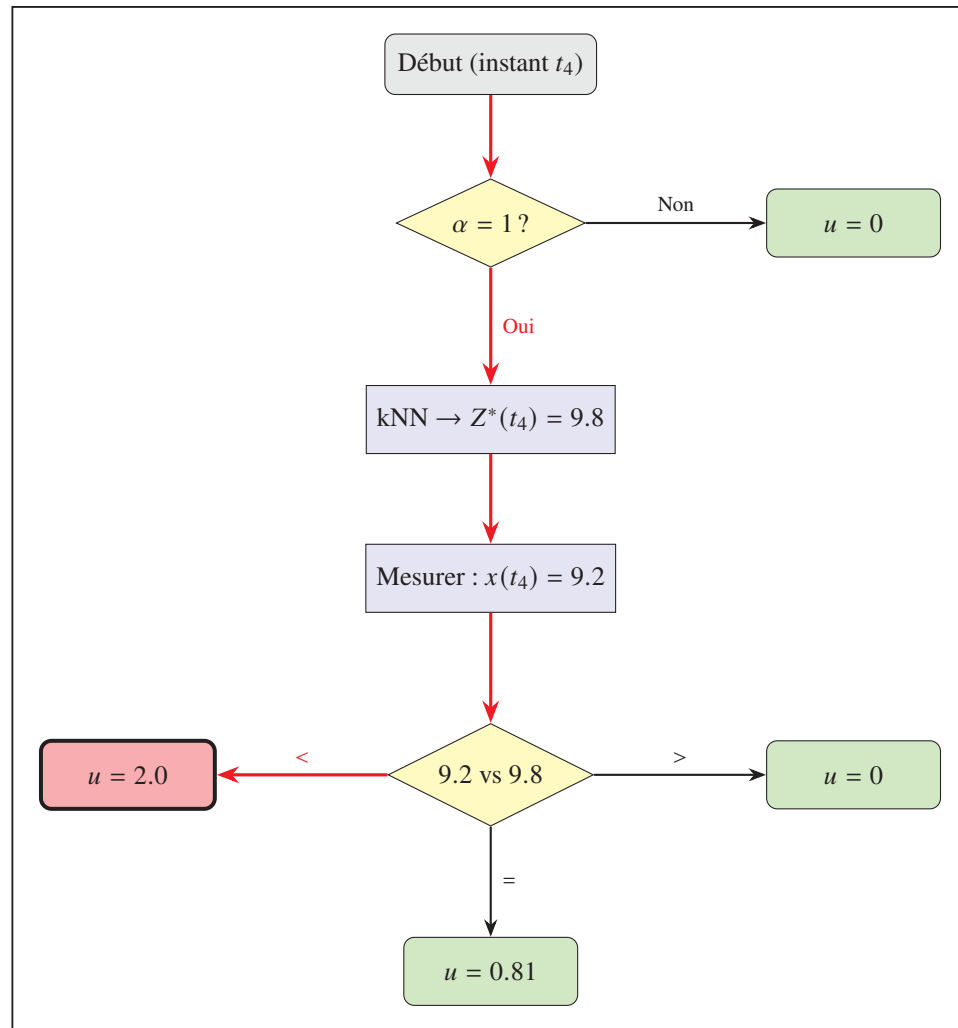


Figure 2.27 Arbre de décision appliqué à l'instant  $t_4$  : le chemin suivi est indiqué en rouge

Le processus décisionnel se déroule en quatre étapes. La première consiste à vérifier l'état de la machine : puisque  $\alpha = 1$ , la machine est opérationnelle et le processus se poursuit. La deuxième étape utilise le modèle kNN pour obtenir le seuil critique  $Z^*(t_4) = 9.8$  unités. La troisième étape mesure le stock actuel  $x(t_4) = 9.2$  unités via le système d'information. Enfin, la quatrième étape compare ces deux valeurs : puisque  $9.2 < 9.8$ , le stock est insuffisant par rapport au niveau optimal. Le système produit donc à taux maximal  $u(t_4) = U_{max} = 2.0$  pour reconstituer rapidement le stock avant le pic de demande anticipé.

De manière générale, la comparaison entre  $x(t)$  et  $Z^*(t)$  conduit à l'une des trois actions : produire à taux maximal si le stock est insuffisant, produire au taux de la demande si le stock est au niveau optimal, ou arrêter la production si le stock est excédentaire.

## 2.7 Conclusion

Ce chapitre a démontré que l'algorithme k-Nearest Neighbors constitue une alternative efficace pour approximer les paramètres optimaux de la politique de production d'un système mono-produit soumis à une demande périodiquement variable. La comparaison de trois algorithmes de régression a révélé que le kNN présente les meilleures performances parmi les modèles testés, devant le DNN et le RBF, et qu'il maintient ces performances même après l'intégration du coût de rupture et du taux de réparation comme variables supplémentaires. L'avantage décisif réside dans le gain computationnel permettant l'exploration rapide de multiples scénarios. Les résultats présentés sont valides dans le domaine couvert par les données d'apprentissage et reposent sur les mêmes hypothèses de modélisation de la demande périodique. Toutefois, cette approche se limite ici à un système mono-produit dont la dimension de l'espace des variables d'entrées est modérée.

Le chapitre suivant étend l'analyse à un système multi-produits, dans lesquels l'augmentation du nombre de variables d'entrée soulève une question méthodologique centrale : les méthodes d'apprentissage basées sur les distances, telles que le k plus proches voisins, demeurent-elles performantes face à cette complexité accrue, ou est-il nécessaire de recourir à des modèles plus sophistiqués, comme les réseaux de neurones profonds ?

## CHAPITRE 3

### EXTENSION DE L'APPROCHE À UN SYSTÈME MANUFACTURIER PRODUISANT DEUX TYPES DE PIÈCES

#### 3.1 Introduction

Le chapitre précédent a montré que l'algorithme des  $k$  plus proches voisins permet de prédire efficacement les paramètres optimaux d'un système mono-produit soumis à une demande périodique. Toutefois, dans de nombreux contextes industriels, plusieurs produits partagent une même capacité de production, ce qui accroît significativement la dimensionnalité du problème d'optimisation. Cette configuration soulève alors une question méthodologique essentielle : l'algorithme kNN, performant en dimension modérée, demeure-t-il adapté lorsque le nombre de variables d'entrée augmente ?

Ce chapitre vise d'abord à valider la pertinence de l'approche par Machine Learning sur un manufacturier produisant deux types de pièces, puis à analyser l'influence du nombre de variables d'entrée sur le choix de l'algorithme d'apprentissage. Une méthodologie en deux phases est adoptée, reposant sur l'utilisation de solutions issues de la résolution numérique des équations de Hamilton–Jacobi–Bellman comme données d'apprentissage. La première phase évalue les performances du kNN dans un contexte où le nombre de variables d'entrée reste modéré, confirmant ainsi la capacité de l'approche à s'étendre au cas multi-produits. La seconde phase intègre des paramètres économiques et de fiabilité supplémentaires, mettant en évidence les limites du kNN face à l'augmentation du nombre de variables et justifiant le recours aux réseaux de neurones profonds

Le chapitre s'articule autour de la modélisation du système multi-produits et de la commande optimale (section 3.2), de l'application du Machine Learning à la prédiction des paramètres optimaux (section 3.3), des résultats de la première phase en dimension modérée (section 3.4), des résultats de la seconde phase illustrant la nécessité des réseaux de neurones profonds (section 3.5), et se conclut par une synthèse des résultats.

### 3.2 Modélisation du système multi-produits et commande optimale stochastique

Cette section présente la modélisation du système manufacturier considéré ainsi que la formulation du problème de contrôle optimal associé.

Le système de production étudié est illustré à la Figure 3.1. La machine unique produit deux types de produits finis dont les niveaux de stock sont notés respectivement  $x_1(t)$  et  $x_2(t)$ . Ces produits sont soumis à des demandes variables dans le temps, modélisées par les fonctions  $d_1(t)$  et  $d_2(t)$ . Le décideur contrôle les taux de production  $u_1(t)$  et  $u_2(t)$  pour chaque produit, sous des contraintes de capacité maximale  $U_{\max}^1$  et  $U_{\max}^2$ .

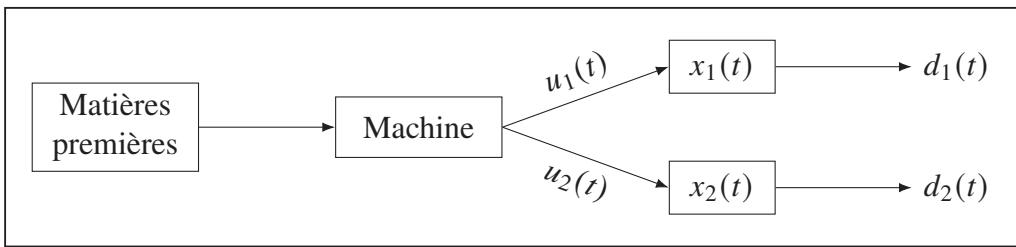


Figure 3.1 Structure du système étudié

La dynamique des stocks est décrite par les équations différentielles :

$$\dot{x}_i(t) = u_i(t) - d_i(t), \quad i = 1, 2. \quad (3.1)$$

avec les conditions initiales  $x_1(0) = x_{10}$ ,  $x_2(0) = x_{20}$ , et les contraintes :

$$0 \leq u_i(t) \leq U_{\max}^i \quad i = 1, 2. \quad (3.2)$$

Les taux de demande sont modélisés par des fonctions sinusoïdales de la forme :

$$d_i(t) = d_{mi} + d_{ai} \sin(\omega_i t + \varphi_i), \quad i = 1, 2, \quad (3.3)$$

où  $d_{mi}$  et  $d_{ai}$  représentent respectivement la moyenne et l'amplitude des variations du taux de demande du produit  $i$ , tandis que  $\omega_i$  et  $\varphi_i$  désignent la fréquence et la phase initiale.

Afin de simplifier la modélisation, on suppose que les paramètres de fréquence et de phase sont identiques pour les deux produits :

$$\omega_1 = \omega_2 = \omega, \quad \varphi_1 = \varphi_2 = \varphi. \quad (3.4)$$

Les demandes deviennent alors :

$$d_i(t) = d_{mi} + d_{ai} \sin(\omega t + \varphi), \quad i = 1, 2. \quad (3.5)$$

Pour garantir la positivité des taux de demande, on impose  $d_{mi} > d_{ai}$  pour  $i = 1, 2$ .

### 3.2.1 Diagramme des taux de transition

Le fonctionnement de la machine est modélisé par un processus de Markov à temps continu à deux états  $\xi(t) \in \{1, 2\}$ , où le mode 1 correspond à l'état opérationnel et le mode 2 à l'état de défaillance. Le diagramme de la Figure 3.2 illustre ces transitions entre les deux états de la machine.

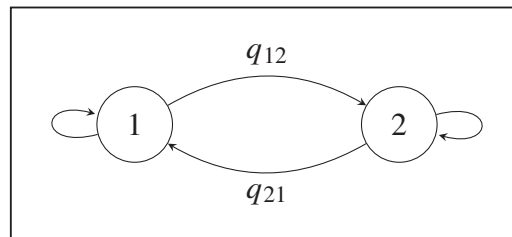


Figure 3.2 Diagramme des taux de transition entre les deux états de la machine

La matrice des taux de transition (ou matrice génératrice) associée au processus  $\{\xi(t), t \geq 0\}$  s'écrit alors :

$$Q = (q_{\alpha\beta})_{\alpha,\beta \in \mathcal{E}} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix} = \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix}. \quad (3.6)$$

Les probabilités de transition sur un petit intervalle de temps  $[t, t + dt]$  s'en déduisent :

$$\begin{aligned} \mathbb{P}(\xi(t + dt) = 2 \mid \xi(t) = 1) &= q_{12} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 1 \mid \xi(t) = 2) &= q_{21} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 1 \mid \xi(t) = 1) &= 1 - q_{12} dt + o(dt), \\ \mathbb{P}(\xi(t + dt) = 2 \mid \xi(t) = 2) &= 1 - q_{21} dt + o(dt). \end{aligned} \quad (3.7)$$

### 3.2.2 Probabilités limites et conditions de faisabilité

Lorsque les taux de panne et de réparation  $q_{12}$  et  $q_{21}$  sont constants, le processus de Markov à temps continu  $\{\xi(t)\}$  admet une distribution stationnaire  $\boldsymbol{\pi} = (\pi_1, \pi_2)$ , où  $\pi_\alpha$  désigne la probabilité limite que la machine se trouve dans l'état  $\alpha \in \{1, 2\}$ . Cette distribution vérifie le système linéaire classique :

$$\boldsymbol{\pi}Q = \mathbf{0}, \quad \pi_1 + \pi_2 = 1, \quad (3.8)$$

où  $Q$  est la matrice génératrice définie précédemment. On en déduit les probabilités limites explicites :

$$\pi_1 = \frac{q_{21}}{q_{12} + q_{21}}, \quad \pi_2 = \frac{q_{12}}{q_{12} + q_{21}}. \quad (3.9)$$

La quantité  $\pi_1$  représente la disponibilité moyenne de la machine à long terme.

Dans le cas d'un système constitué d'une machine produisant deux types de pièces, la capacité de production effective doit être suffisante pour couvrir la demande maximale agrégée des deux produits. En notant  $U_{\max}^i$  le taux de production maximal du produit  $i$ , la capacité maximale totale de la machine est donnée par :

$$U_{\max}^1 + U_{\max}^2 \quad (3.10)$$

Les taux de demande agrégés s'écrivent :

$$d_m = d_{m1} + d_{m2}, \quad d_a = d_{a1} + d_{a2}, \quad (3.11)$$

de sorte que la demande maximale totale vaut  $d_m + d_a$ .

Afin de garantir que la production moyenne, pondérée par la disponibilité de la machine, permet de satisfaire cette demande maximale, la condition de faisabilité suivante doit être vérifiée :

$$\pi_1 \left( U_{\max}^1 + U_{\max}^2 \right) > d_m + d_a. \quad (3.12)$$

Cette condition exprime que, à long terme, la capacité moyenne effective de la machine doit excéder la demande maximale agrégée des deux produits, assurant ainsi l'existence d'une politique de production admissible.

### 3.2.3 Conditions d'optimalité

La fonction de coût instantané  $g(\alpha, \cdot)$  dépend des coûts de stockage et de rupture, et s'écrit :

$$g(\alpha, \cdot) = c_1^+ x_1^+ + c_2^+ x_2^+ + c_1^- x_1^- + c_2^- x_2^-. \quad (3.13)$$

où :

$$x_j^+ = \max(0, x_j), \quad x_j^- = \max(0, -x_j), \quad j = 1, 2, \quad (3.14)$$

et  $c_i^+$ ,  $c_i^-$  désignent respectivement les coûts unitaires de possession (stockage) et de rupture (backlog) du produit  $i$ .

Le coût total actualisé est donné par :

$$J(\cdot) = \mathbb{E} \left[ \int_t^\infty e^{-\rho t} g(\alpha, \cdot) dt \mid x_1(0) = x_{01}, x_2(0) = x_{02}, \xi(0) = \alpha \right], \quad \forall \alpha \in \{1, 2\}, \quad (3.15)$$

où  $\rho$  représente le taux d'actualisation et  $g(\alpha, \cdot)$  désigne le coût instantané.

Le domaine des commandes admissibles est défini par :

$$\mathcal{A}(\alpha) = \{(u_1(\cdot), u_2(\cdot)) \in \mathbb{R}^2 \mid 0 \leq u_1(\cdot) \leq U_{\max}^1, 0 \leq u_2(\cdot) \leq U_{\max}^2\}, \quad \alpha \in \{1, 2\}. \quad (3.16)$$

Soit  $v(\alpha, \cdot)$  la fonction valeur :

$$v(\alpha, \cdot) = \min_{(u_1, u_2) \in \mathcal{A}(\alpha)} J(u_1, u_2, \alpha, t), \quad \alpha \in \{1, 2\}. \quad (3.17)$$

Selon Polotski *et al.* (2019), les équations de Hamilton–Jacobi–Bellman permettant de déterminer la politique optimale s'écrivent :

$$\begin{aligned} \rho v(t, x_1, x_2, \alpha) = & \frac{\partial v}{\partial t}(t, x_1, x_2, \alpha) \\ & + \min_{(u_1, u_2) \in \mathcal{A}(\alpha)} \left\{ g(\alpha, \cdot) + \frac{\partial v}{\partial x_1} (u_1 \text{Ind}_{\{\alpha=1\}} - d_1(t)) \right. \\ & \quad + \frac{\partial v}{\partial x_2} (u_2 \text{Ind}_{\{\alpha=1\}} - d_2(t)) \\ & \quad \left. + \sum_{\beta \neq \alpha} q_{\alpha\beta} (v(t, x_1, x_2, \beta) - v(t, x_1, x_2, \alpha)) \right\} \end{aligned} \quad (3.18)$$

La résolution analytique de ces équations étant inaccessible en raison de leur caractère non stationnaire, on présente dans ce qui suit l'approche numérique adoptée.

### 3.2.4 Résolution numérique et politique optimale

#### 3.2.4.1 Résolution numérique

Les équations de Hamilton–Jacobi–Bellman non stationnaires restent difficiles, voire impossibles, à traiter analytiquement. On recourt donc à une approximation numérique fondée sur la méthode de discrétisation proposée par Kushner *et al.* (1992), qui consiste à remplacer les dérivées spatiales par des schémas aux différences finies sur une grille de l'espace d'état.

On considère une grille rectangulaire dans le plan des niveaux de stock  $(x_1, x_2)$ , définie à partir de pas de discrétisation  $h_1 > 0$  et  $h_2 > 0$ . La fonction valeur continue  $v(x_1, x_2, t, \alpha)$  est alors approchée par une fonction discrète  $v^h(x_1, x_2, t, \alpha)$  définie sur cette grille, pour tout  $\alpha \in \{1, 2\}$ .

La dynamique des stocks dans le mode opérationnel s'écrit, pour chaque produit  $i$ ,

$$\dot{x}_i(t) = u_i(t) - d_i(t), \quad i \in \{1, 2\},$$

où  $u_i(t)$  est le taux de production et  $d_i(t)$  le taux de demande du produit  $i$ . Comme dans le cas mono-produit, on utilise un schéma amont (upwind) pour approximer les dérivées spatiales, en choisissant la direction de la différence finie en fonction du signe de la dérive  $\dot{x}_i(t)$ .

Pour  $i = 1$ , la dérivée partielle de la fonction valeur par rapport à  $x_1$  est approchée par :

$$\frac{\partial v(x_1, x_2, t, \alpha)}{\partial x_1} \approx \begin{cases} \frac{v^h(x_1 + h_1, x_2, t, \alpha) - v^h(x_1, x_2, t, \alpha)}{h_1}, & \text{si } \dot{x}_1(t) \geq 0, \\ \frac{v^h(x_1, x_2, t, \alpha) - v^h(x_1 - h_1, x_2, t, \alpha)}{h_1}, & \text{si } \dot{x}_1(t) < 0, \end{cases} \quad (3.19)$$

et, de manière analogue, pour  $x_2$  :

$$\frac{\partial v(x_1, x_2, t, \alpha)}{\partial x_2} \approx \begin{cases} \frac{v^h(x_1, x_2 + h_2, t, \alpha) - v^h(x_1, x_2, t, \alpha)}{h_2}, & \text{si } \dot{x}_2(t) \geq 0, \\ \frac{v^h(x_1, x_2, t, \alpha) - v^h(x_1, x_2 - h_2, t, \alpha)}{h_2}, & \text{si } \dot{x}_2(t) < 0. \end{cases} \quad (3.20)$$

Pour alléger l'écriture, on introduit les indicatrices suivantes, qui sélectionnent la différence finie amont appropriée pour chaque direction :

$$\begin{cases} I_1^+ = \text{Ind}\{u_1(t) - d_1(t) \geq 0\}, & I_1^- = \text{Ind}\{u_1(t) - d_1(t) < 0\} \\ I_2^+ = \text{Ind}\{u_2(t) - d_2(t) \geq 0\}, & I_2^- = \text{Ind}\{u_2(t) - d_2(t) < 0\} \end{cases} \quad (3.21)$$

Dans le mode opérationnel  $\alpha = 1$ , la dynamique complète du système est décrite par le couple  $(x_1, x_2)$  et le contrôle  $(u_1, u_2)$ , contraint à appartenir à l'ensemble admissible  $\Gamma(1)$  (voir section de faisabilité). En remplaçant les dérivées spatiales par les approximations ci-dessus dans les équations de Hamilton–Jacobi–Bellman, on obtient, pour tout point  $(x_1, x_2)$  de la grille, l'approximation numérique suivante :

$$\begin{aligned} \rho v^h(x_1, x_2, t, 1) = & \frac{\partial v^h(x_1, x_2, t, 1)}{\partial t} \\ & + \min_{(u_1, u_2) \in \Gamma(1)} \left\{ h(x_1, x_2) + q_{12}(v^h(x_1, x_2, t, 2) - v^h(x_1, x_2, t, 1)) \right. \\ & + (u_1 - d_1(t)) \left[ \frac{v^h(x_1 + h_1, x_2, t, 1) - v^h(x_1, x_2, t, 1)}{h_1} I_1^+ \right. \\ & \quad \left. + \frac{v^h(x_1, x_2, t, 1) - v^h(x_1 - h_1, x_2, t, 1)}{h_1} I_1^- \right] \\ & \left. + (u_2 - d_2(t)) \left[ \frac{v^h(x_1, x_2 + h_2, t, 1) - v^h(x_1, x_2, t, 1)}{h_2} I_2^+ \right. \right. \\ & \quad \left. \left. + \frac{v^h(x_1, x_2, t, 1) - v^h(x_1, x_2 - h_2, t, 1)}{h_2} I_2^- \right] \right\}. \end{aligned} \quad (3.22)$$

Dans le mode de panne  $\alpha = 2$ , la production est interrompue ( $u_1(t) = u_2(t) = 0$ ), et les stocks décroissent uniquement sous l'effet des demandes, selon  $\dot{x}_i(t) = -d_i(t)$ . Les signes des dérivées sont alors déterminés par les seuls termes  $-d_i(t)$ , qui sont strictement négatifs. Dans cette configuration, les dérivées spatiales sont naturellement approchées à l'amont en utilisant les points  $(x_1 - h_1, x_2)$  et  $(x_1, x_2 - h_2)$ . L'équation d'Hamilton–Jacobi–Bellman discrétisée associée au mode  $\alpha = 2$  s'écrit alors :

$$\begin{aligned} \rho v^h(x_1, x_2, t, 2) = & \frac{\partial v^h(x_1, x_2, t, 2)}{\partial t} + h(x_1, x_2) \\ & + q_{21}(v^h(x_1, x_2, t, 1) - v^h(x_1, x_2, t, 2)) \\ & - d_1(t) \frac{v^h(x_1, x_2, t, 2) - v^h(x_1 - h_1, x_2, t, 2)}{h_1} \\ & - d_2(t) \frac{v^h(x_1, x_2, t, 2) - v^h(x_1, x_2 - h_2, t, 2)}{h_2}. \end{aligned} \quad (3.23)$$

Les équations (3.22) et (3.23) définissent un système couplé d'équations aux différences finies en  $(x_1, x_2, t)$ , à résoudre numériquement sur la grille de l'espace des états et sur l'horizon temporel considéré.

Pour prendre en compte le terme de dérivée partielle par rapport au temps dans les équations de Hamilton–Jacobi–Bellman, on adopte une procédure numérique inspirée de Polotski *et al.* (c), qui se déroule en plusieurs étapes successives :

**1. Discrétisation de l'horizon temporel.**

L'horizon d'étude est discrétisé en  $N$  instants  $(t_j)_{j=0,\dots,N}$ , répartis sur une période complète de variation de la demande. À chaque instant  $t_j$ , la demande est considérée comme figée.

**2. Évaluation des taux de demande aux instants discrétisés.**

Pour chaque instant  $t_j$ , les taux de demande des deux produits sont calculés selon :

$$d_1^j = d_{m1} + d_{a1} \sin(\omega t_j + \varphi), \quad d_2^j = d_{m2} + d_{a2} \sin(\omega t_j + \varphi) \quad (3.24)$$

de sorte que le problème de contrôle est étudié pour un couple de demandes  $(d_1^j, d_2^j)$  considéré constant à l'instant  $t_j$ .

**3. Résolution des équations HJB pour chaque instant figé.**

Pour chaque  $t_j$ , on résout numériquement les équations d'Hamilton–Jacobi–Bellman associées au système avec demandes constantes  $(d_1^j, d_2^j)$ , ce qui conduit à une famille de fonctions valeur discrètes  $\{v^j(x_1, x_2, \alpha)\}_{j=0,\dots,N}$ .

**4. Approximation de la dérivée temporelle et réinjection dans les HJB.**

Les dérivées partielles par rapport au temps sont ensuite approchées par un schéma de différences finies entre instants successifs :

$$\frac{\partial v(x_1, x_2, \alpha)}{\partial t} \simeq \frac{v^{j+1}(x_1, x_2, \alpha) - v^j(x_1, x_2, \alpha)}{t_{j+1} - t_j}, \quad (3.25)$$

puis réintégrées dans les équations de Hamilton–Jacobi–Bellman non stationnaires afin d'obtenir des politiques de production optimales adaptées à la demande périodiquement variable.

### 3.2.4.2 Détermination du coût optimal

Le calcul du coût optimal pour le système à deux produits s'appuie sur la résolution numérique des équations de Hamilton–Jacobi–Bellman, comme démontré explicité plus haut.

À chaque instant  $t_j$ , la résolution des équations d'HJB conduit à deux fonctions valeur discrètes  $v_1^j(x_1, x_2)$  et  $v_2^j(x_1, x_2)$ , correspondant respectivement aux modes de fonctionnement et de panne de la machine. La valeur optimale retenue à l'instant  $t_j$  est associée à la fonction valeur minimale, qui représente le coût total actualisé le plus faible pour le système. En notant  $v_{\min}^j$  cette valeur optimale, on définit par exemple

$$v_{\min}^j = \min_{x_1, x_2} v_1^j(x_1, x_2) \quad (3.26)$$

lorsque l'on suppose que le système démarre en mode opérationnel.

Le coût optimal global  $C$  est obtenu en moyennant la fonction valeur sur l'intervalle de discrétisation temporelle  $[T_{\min}, T_{\max}]$ , où  $T_{\min} = t_1$  et  $T_{\max} = t_n$  :

$$C = \frac{1}{T_{\max} - T_{\min}} \int_{T_{\min}}^{T_{\max}} v(t, x_1, x_2, \alpha) dt. \quad (3.27)$$

Dans la pratique, la fonction valeur n'est connue qu'en un nombre fini d'instantes  $\{t_j\}_{j=1, \dots, n}$ , sous la forme des valeurs discrètes  $\{v_{\min}^j\}_{j=1, \dots, n}$ . L'intégrale précédente est alors approchée par la méthode des trapèzes :

$$C \approx \hat{C} = \frac{1}{T_{\max} - T_{\min}} \sum_{j=1}^{n-1} \frac{t_{j+1} - t_j}{2} (v_{\min}^j + v_{\min}^{j+1}), \quad (3.28)$$

où  $v_{\min}^j$  représente le coût optimal à l'instant  $t_j$ . Cette quantité  $\hat{C}$  est utilisée comme estimation numérique du coût total actualisé associé à la politique de production optimale du système.

### 3.2.4.3 Politiques optimales de production

D'après Polotski *et al.* (2019), les politiques optimales de production obtenues sont des politiques à courbes critiques pour chaque produit. Ces politiques définissent des seuils qui varient avec le temps des niveaux de stocks, notés  $z_1^*(t)$  et  $z_2^*(t)$  pour les produits 1 et 2 respectivement, qui varient dans le temps en réponse aux fluctuations de la demande. La politique optimale de production s'exprime comme suit :

$$u_i^*(x_1, x_2, t) = \begin{cases} U_{\max}^i & \text{si } x_i(t) < z_i^*(t) \\ d_i(t) & \text{si } x_i(t) = z_i^*(t) \\ 0 & \text{si } x_i(t) > z_i^*(t) \end{cases} \quad i = 1, 2. \quad (3.29)$$

Une caractéristique centrale de cette politique est sa capacité d'**anticipation** : les seuils critiques de stock évoluent de manière à précéder les fluctuations des taux de demande. La production ne réagit donc pas seulement au niveau de stock instantané, mais tient compte de l'évolution prévisible des demandes périodiques, ce qui permet de limiter à la fois les ruptures et les surstockages inutiles.

La Figure 3.3 illustre cet effet pour le produit 1, en comparant la trajectoire du seuil critique optimal  $z_1^*(t)$  à celle du taux de demande  $d_1(t)$ . De façon analogue, la Figure 3.4 présente l'évolution du seuil  $z_2^*(t)$  en regard du profil de demande  $d_2(t)$  pour le produit 2. Les valeurs numériques des paramètres utilisés pour générer les courbes présentées sont rassemblées dans le Tableau 3.1.

Table 3.1 Valeurs numériques des paramètres utilisés pour l'exemple

$D_{m1}$	$D_{a1}$	$\omega_1$	$\varphi_1$	$D_{m2}$	$D_{a2}$	$\omega_2$	$\varphi_2$
0.50	0.066	0.005	$-\frac{\pi}{2}$	0.49	0.059	0.005	$-\frac{\pi}{2}$
$U_{m1}$	$U_{m2}$	$q_{12}$	$q_{21}$	$\rho$	$h_{x_1}$	$h_{x_2}$	$N$
1.50	1.25	0.05	0.15	0.01	0.25	0.25	10

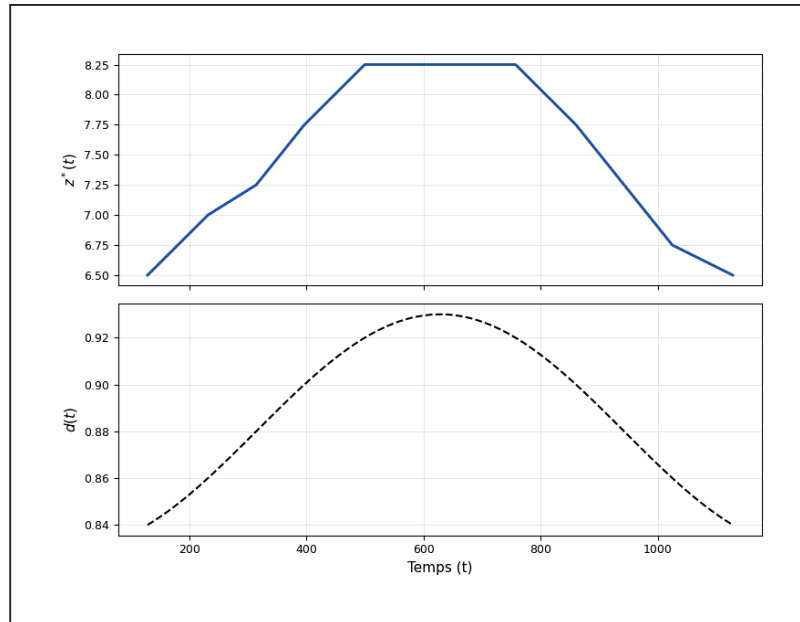


Figure 3.3 Effet d'anticipation pour le produit 1 : évolution du seuil critique optimal  $z_1^*(t)$  et du taux de demande  $d_1(t)$

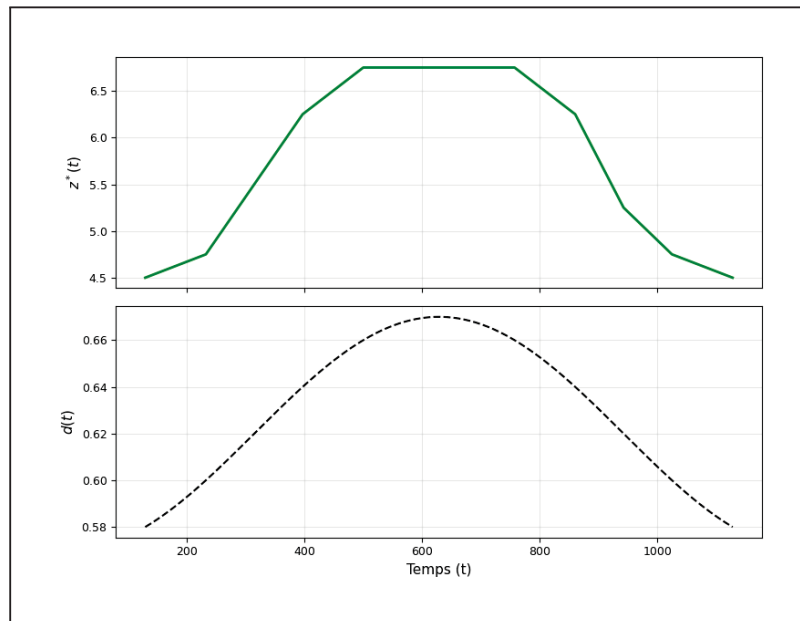


Figure 3.4 Effet d'anticipation pour le produit 2 : évolution du seuil critique optimal  $z_2^*(t)$  et du taux de demande  $d_2(t)$

La résolution numérique des équations d’HJB pour ce système amplifie considérablement la charge computationnelle : l’espace d’état bidimensionnel  $(x_1, x_2)$  et le contrôle simultané de deux taux de production  $(u_1, u_2)$  multiplient les combinaisons à explorer lors de la discrétisation. Une configuration paramétrique unique nécessite déjà un temps de calcul conséquent, rendant l’exploration de scénarios multiples impraticable. C’est dans ce contexte que l’apprentissage automatique supervisé s’impose comme alternative : en s’appuyant sur une base de solutions HJB pré-calculées, il permet de prédire quasi-instantanément les paramètres optimaux chaque fois que des paramètres du système changent (qui sont les variables du modèle de machine learning).

### **3.3 Prédiction des paramètres optimaux par Machine Learning**

La résolution numérique des équations d’Hamilton-Jacobi-Bellman pour un système bi-produits amplifie considérablement la charge computationnelle comparativement au système mono-produit étudié au chapitre précédent. L’espace d’état bidimensionnel  $(x_1, x_2)$  et le contrôle simultané de deux taux de production  $(u_1, u_2)$  multiplient les combinaisons à explorer, rendant l’exploration de scénarios multiples impraticable. Face à cette complexité, l’apprentissage automatique supervisé offre une alternative prometteuse : en s’appuyant sur une base de solutions HJB pré-calculées, il permet de prédire quasi-instantanément les paramètres optimaux pour toute nouvelle configuration.

Cette section présente d’abord la reformulation du problème de contrôle optimal en tâche d’apprentissage supervisé (section 3.3.1), puis détaille l’approche méthodologique en deux phases permettant d’évaluer la robustesse des algorithmes face à l’augmentation de la dimensionnalité (section 3.3.2).

#### **3.3.1 Reformulation du problème HJB en tâche d’apprentissage**

La méthodologie développée au chapitre précédent sert ici de base pour reformuler le problème de commande optimale originel, défini par les équations d’Hamilton-Jacobi-Bellman (HJB),

comme une tâche d'apprentissage supervisé. Concrètement, le but est d'entraîner un modèle d'apprentissage automatique à prédire, à partir d'un certain ensemble de variables d'entrée caractérisant la demande et les paramètres du système, les paramètres optimaux déjà obtenus par résolution numérique des HJB.

Deux configurations sont considérées pour évaluer l'impact de la dimensionnalité. En phase 1, les entrées sont  $[d_{m1}, d_{a1}, d_{m2}, d_{a2}]$ . En phase 2, deux variables s'ajoutent : le coût de rupture  $c^-$  et le taux de réparation  $q_{21}$ . Les sorties restent identiques : les trajectoires optimales  $z_1^*(t)$ ,  $z_2^*(t)$  et le coût global  $C$ .

Cette représentation transforme la résolution complexe et coûteuse des équations HJB en un problème de régression supervisée, dans lequel le modèle d'apprentissage automatique approxime la fonction d'optimisation reliant les paramètres de demande aux politiques optimales et au coût associé. La phase 1 évalue l'efficacité de l'algorithme k-Nearest Neighbors (kNN) en dimension modérée, tandis que la phase 2 teste sa robustesse face à l'augmentation dimensionnelle et compare ses performances à celles d'un réseau de neurones profond (DNN). Cette démarche progressive permet d'identifier empiriquement le seuil de transition algorithmique au-delà duquel des méthodes plus sophistiquées deviennent nécessaires.

Ainsi, une fois entraîné, le modèle retenu permet de prédire rapidement ces paramètres optimaux pour de nouvelles configurations de demande, contournant la nécessité de résoudre à nouveau les équations HJB pour chaque scénario.

### 3.3.2 Approche méthodologique

La méthodologie proposée reformule le problème de commande optimale, défini par les équations d'HJB, en une tâche d'apprentissage supervisé visant à prédire les paramètres optimaux du système à partir de variables d'entrée caractérisant la demande et les paramètres opérationnels. L'approche s'articule en deux phases complémentaires.

La Phase 1 s'appuie sur l'algorithme des k plus proches voisins (kNN) avec quatre variables d'entrée :  $[d_{m1}, d_{a1}, d_{m2}, d_{a2}]$ . Cette méthode constitue une bonne base en raison de sa simplicité et de son efficacité reconnue pour les problèmes de prédiction, notamment dans les travaux de Ho *et al.* (2021). Les paramètres optimaux sont obtenus par résolution numérique des équations d'HJB avec un filtrage basé sur la condition de faisabilité (3.9). Les variables de sortie sont les trajectoires optimales de stock  $z_1^*(t)$  et  $z_2^*(t)$  (vectorisées) ainsi que le coût global  $C$ .

La Phase 2 enrichit le dataset de la première expérience en ajoutant  $c^-$  (coût de pénurie commun) et  $q_{21}$  (taux de réparation), portant à six le nombre de variables d'entrée. Cette augmentation dimensionnelle teste la robustesse du kNN et justifie la comparaison avec un réseau de neurones profond (DNN). La construction du dataset de la Phase 2 repose sur un enrichissement structuré en deux couches complémentaires visant à maximiser la couverture de l'espace paramétrique à six dimensions (voir figure 3.5).

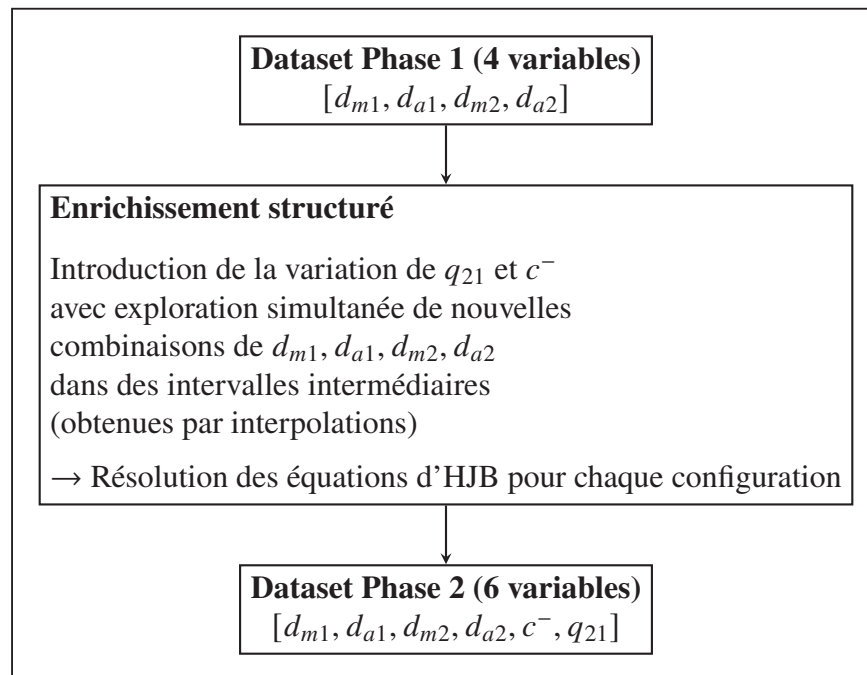


Figure 3.5 Processus d'enrichissement du dataset pour la Phase 2

Les données sont normalisées puis réparties en ensemble d'apprentissage (80 %) et de test (20 %). La Phase 1 évalue le kNN par rapport aux solutions HJB sur un ensemble de scénarios

de test. La Phase 2 compare d'abord le kNN au DNN pour établir la supériorité de ce dernier face à l'augmentation dimensionnelle, puis valide le DNN par rapport aux solutions HJB via des analyses de scénarios et de sensibilité. Ce choix méthodologique est central : le kNN, efficace en faibles dimensions, voit ses performances se dégrader en haute dimension ( phénomène de *malédiction de la dimensionnalité*)(Murphy (2012), Pestov (2011)) tandis que les DNN excellent dans l'exploitation de données multi-paramétriques grâce à leur capacité à modéliser des interactions complexes Goodfellow *et al.* (2016).

Les performances sont quantifiées par l'erreur moyenne absolue (MAE) pour  $z_1^*(t)$  et  $z_2^*(t)$  et l'erreur relative moyenne (ERM) pour le coût  $C$ . La figure 3.6 présente une vue d'ensemble de cette méthodologie.

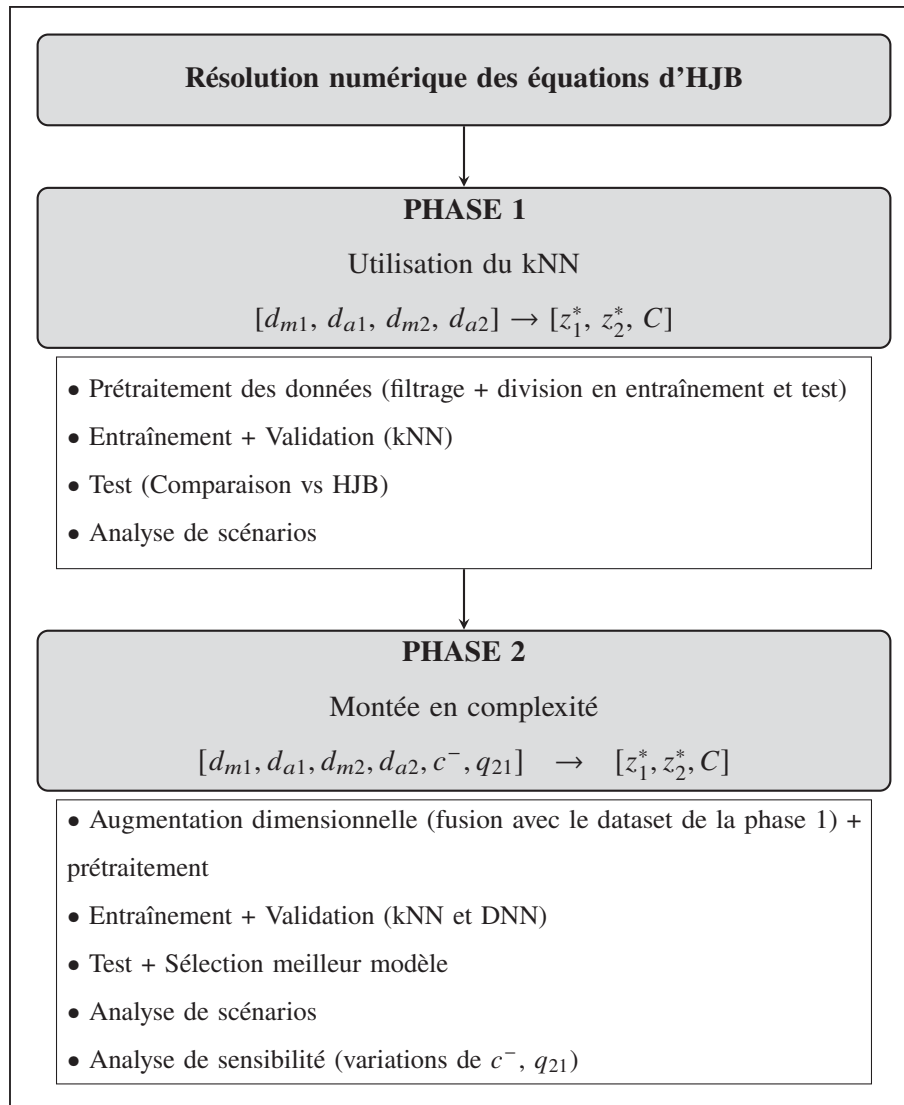


Figure 3.6 Visualisation de la méthodologie proposée

### 3.4 Résultats de la première expérience

Cette section présente les résultats obtenus lors de l'application du kNN pour la prédiction des paramètres optimaux du système multi-produits soumis à un taux de demande variable, en ne considérant que la variation des quatre paramètres de demande. Nous détaillons d'abord les paramètres numériques utilisés pour la génération des données, puis analysons les performances prédictives du modèle à travers différents scénarios représentatifs.

Les paramètres numériques utilisés pour générer les données sont synthétisés dans le tableau 3.2. Les paramètres opérationnels du système sont fixés, tandis que les paramètres de demande ( $d_{m1}$ ,  $d_{a1}$ ,  $d_{m2}$ ,  $d_{a2}$ ) varient sur les intervalles indiqués pour constituer un jeu de données diversifié.

Table 3.2 Paramètres numériques pour la Phase 1

$\omega$	$\varphi$	$U_{\max}^1$	$U_{\max}^2$	$q_{12}$	$q_{21}$	$\rho$	$c_1^+$	$c_1^-$	$c_2^+$
0.005	$-\frac{\pi}{2}$	1.5	1.25	0.05	0.15	0.01	1.0	15.0	0.5
$c_2^-$	$d_{m1}$	$d_{a1}$	$d_{m2}$	$d_{a2}$					
15.0	[0.384, 1.017]	[0.025, 0.075]	[0.303, 0.849]	[0.025, 0.075]					

### 3.4.1 Performances globales du kNN

L'évaluation du modèle kNN repose sur une comparaison systématique entre les prédictions et les valeurs de référence issues de la résolution numérique des équations d'Hamilton–Jacobi–Bellman. Une validation croisée a été réalisée pour ajuster les hyperparamètres du modèle. La configuration retenue utilise la distance euclidienne, un nombre de voisins fixé à  $k = 7$  et une pondération des voisins par l'inverse de la distance. Ce choix offre un compromis satisfaisant entre biais et variance.

La Figure 3.7 illustre la qualité de la prédiction du coût optimal  $C$  sur l'ensemble de test. Chaque point compare une valeur prédite par le modèle à la valeur de référence correspondante, et la droite en pointillés représente la prédiction parfaite ( $y = x$ ). L'alignement serré des points autour de cette bissectrice indique que le kNN reproduit fidèlement le coût optimal calculé à partir des équations d'HJB.

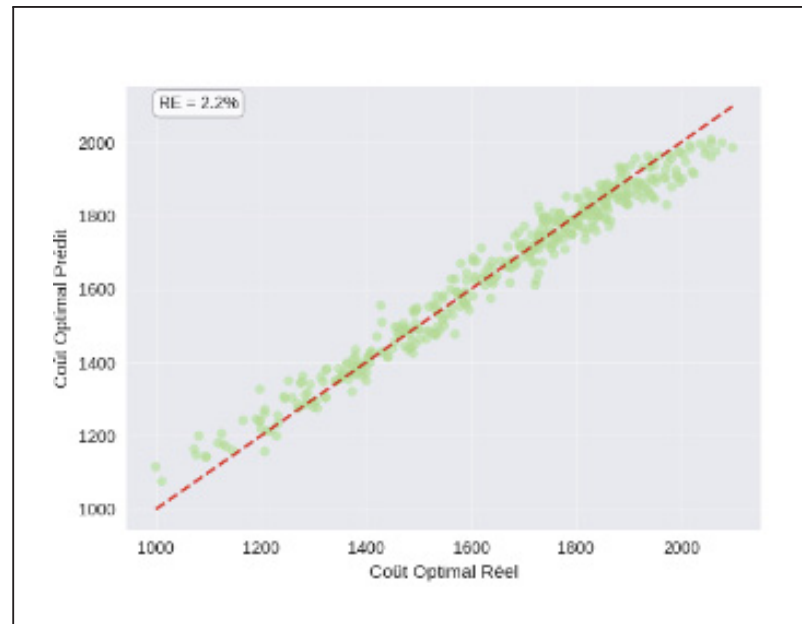


Figure 3.7 Comparaison entre le coût optimal  $C$  prédit par le modèle kNN et les valeurs de référence issues des HJB sur l'ensemble de test

Les Figures 3.8 et 3.9 présentent des résultats analogues pour les trajectoires des seuils critiques de stock  $z_1^*(t)$  et  $z_2^*(t)$ . Dans ces cas, chaque configuration génère plusieurs points, car les trajectoires sont discrétisées en une série de valeurs au cours du temps. Les nuages de points restent néanmoins concentrés autour de la diagonale, ce qui montre que le modèle capture correctement la forme globale des courbes optimales.

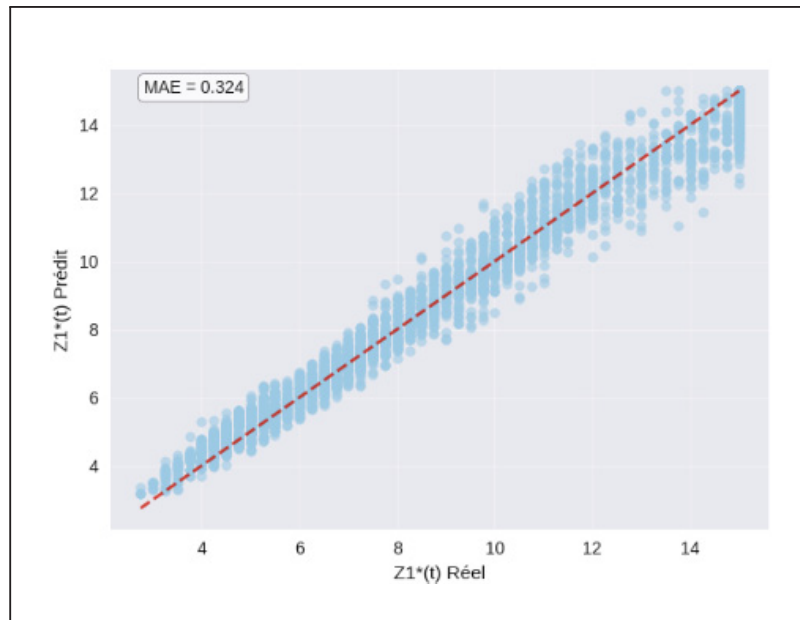


Figure 3.8 Comparaison entre les valeurs de  $z_1^*(t)$  prédites par le kNN et les valeurs de référence issues des HJB

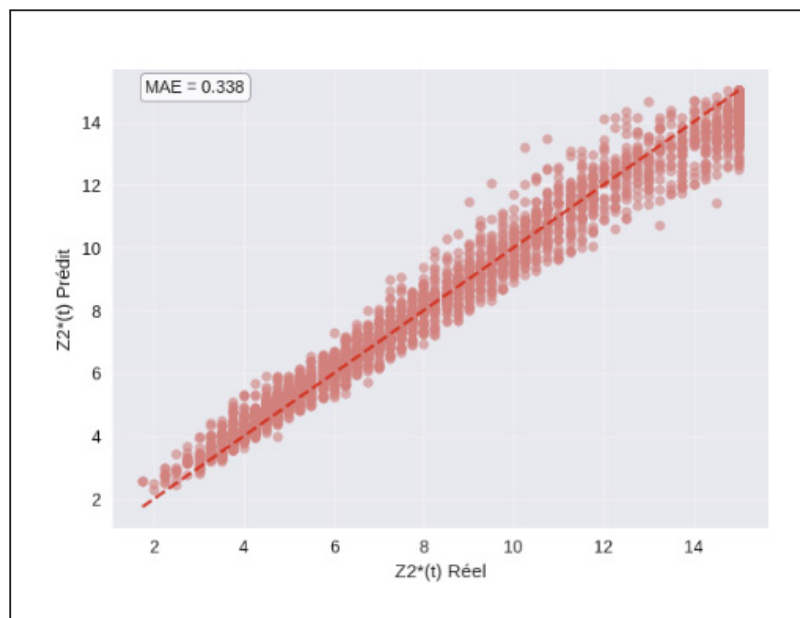


Figure 3.9 Comparaison entre les valeurs de  $z_2^*(t)$  prédites par le kNN et les valeurs de référence issues des HJB

Les indicateurs quantitatifs confirment la bonne précision globale du modèle : l'erreur relative moyenne sur le coût optimal  $C$  est de 2,2 %, tandis que l'erreur absolue moyenne (MAE) vaut 0,324 pour  $z_1^*(t)$  et 0,338 pour  $z_2^*(t)$ . Ces résultats montrent que le modèle kNN constitue un surrogate fiable pour la prédiction des sorties optimales du système bi-produits.

### 3.4.2 Analyse de scénarios

Cette section évalue la performance du modèle kNN à travers une sélection de scénarios représentatifs. Un premier scénario illustre en détail la capacité du modèle à reproduire fidèlement les trajectoires optimales et le coût associé, comparés aux résultats HJB. L'évaluation est ensuite étendue à 10 scénarios extraits des 50 scénarios de l'annexe II, dont les résultats synthétiques attestent de la capacité généralisante du modèle.

La Figure 3.10 compare, pour le produit 1, la trajectoire du seuil critique optimal obtenue par résolution HJB et celle prédite par le modèle kNN. Les profils temporels sont globalement bien reproduits, avec un léger décalage en niveau.

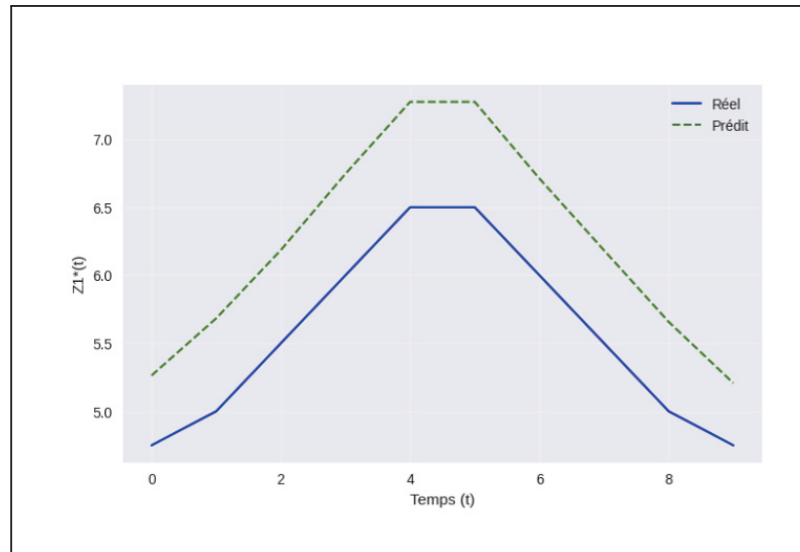


Figure 3.10 Comparaison des trajectoires de  $z_1^*(t)$  obtenues par HJB (référence) et prédites par le modèle kNN

La Figure 3.11 présente le même type de comparaison pour le produit 2. Là encore, la dynamique globale de la trajectoire est correctement capturée, malgré une légère surestimation des niveaux de stock.

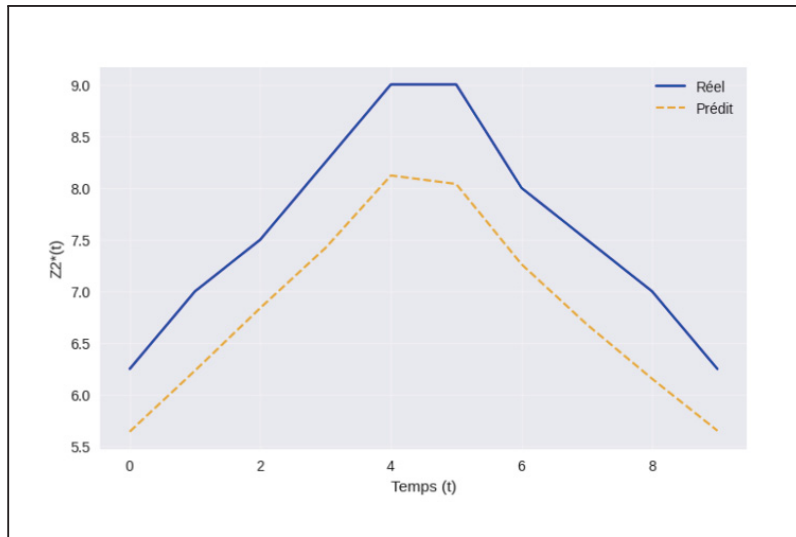


Figure 3.11 Comparaison des trajectoires de  $z_2^*(t)$  obtenues par HJB (référence) et prédites par le modèle kNN

### 3.4.2.1 Validation du modèle

L'évaluation est étendue à 10 scénarios représentatifs générés par interpolation sur les valeurs de l'ensemble de test, couvrant une large diversité de configurations. Le tableau 3.3 présente pour chaque scénario la MAE sur  $z_1^*$  et  $z_2^*$ , le coût optimal obtenu par HJB, le coût prédit par kNN ainsi que l'erreur relative observée. L'erreur relative moyenne sur ces scénarios est de 1,05 %, confirmant la précision du modèle. Les MAE moyennes pour  $z_1^*$  et  $z_2^*$  s'élèvent respectivement à 0,25 et 0,33, témoignant de la capacité du modèle à restituer fidèlement les trajectoires optimales. Ces résultats attestent de la robustesse du kNN pour généraliser au-delà de l'ensemble d'entraînement initial. Les 50 scénarios complets sont disponibles en annexe II.

Table 3.3 Performances du modèle kNN sur 10 scénarios représentatifs : MAE sur  $z_1^*$  et  $z_2^*$ , coûts optimaux HJB et prédits, erreur relative

<b>dm<sub>1</sub></b>	<b>da<sub>1</sub></b>	<b>dm<sub>2</sub></b>	<b>da<sub>2</sub></b>	<b>MAE <math>z_1^*</math></b>	<b>MAE <math>z_2^*</math></b>	<b>Coût HJB</b>	<b>ERM (%)</b>
0.6475	0.0492	0.6029	0.0527	0.136	0.344	1718.88	0.03
0.8397	0.0741	0.4510	0.0393	0.058	0.107	1790.27	0.06
0.9121	0.0357	0.4611	0.0452	0.000	0.613	1678.47	0.09
0.5418	0.0479	0.8193	0.0547	0.264	0.000	1481.11	0.15
0.4826	0.0354	0.4239	0.0564	0.039	0.227	1210.96	0.18
0.7153	0.0514	0.7570	0.0458	0.162	0.000	1823.73	0.24
0.8412	0.0480	0.6021	0.0478	0.151	0.565	1995.23	1.06
0.9870	0.0614	0.7848	0.0481	0.000	0.000	1775.55	1.12
0.7195	0.0594	0.5973	0.0636	0.612	1.140	1879.84	3.01
0.5913	0.0272	0.4782	0.0552	0.588	0.098	1454.09	3.08

L'analyse des scénarios s'articule en deux temps. Un premier scénario illustre la capacité du kNN à reproduire fidèlement les trajectoires optimales  $z_i^*(t)$  et à fournir des prédictions de coûts très proches de celles obtenues par résolution numérique des équations d'HJB. Cette observation est confirmée par une validation élargie sur 50 scénarios (annexe II), dont 10 sont présentés dans le tableau 3.3. L'erreur relative moyenne de 1,72 % sur l'ensemble des 50 scénarios atteste de la précision du modèle. Au-delà de la justesse des prédictions, l'apport majeur du kNN réside dans sa rapidité d'exécution. Alors que la résolution numérique des équations d'HJB nécessite entre 5 et 10 minutes par scénario, la prédiction par le modèle entraîné s'effectue de manière quasi-instantanée (moins d'une seconde), permettant une exploration efficace de nombreuses configurations sans recalculer systématiquement des solutions coûteuses. Ce gain en temps de calcul constitue un atout décisif pour l'aide à la décision en temps réel en environnement industriel.

### 3.5 Résultats de la deuxième expérience et transition vers les réseaux de neurones profonds

Cette section présente les résultats de la seconde expérience, où la prédiction des paramètres optimaux intègre désormais six variables d'entrée : les quatre paramètres de demande ( $d_{m1}$ ,  $d_{a1}$ ,  $d_{m2}$ ,  $d_{a2}$ ), le coût de rupture de stock  $c^-$  (identique pour les deux produits par souci de simplification) et le taux de réparation  $q_{21}$ . Cette expérience compare les performances de deux algorithmes : le kNN et le réseau de neurones profond (DNN). Les paramètres numériques utilisés sont synthétisés dans le tableau 3.4, où les paramètres opérationnels du système sont fixés tandis que les variables d'entrée varient sur les intervalles indiqués pour constituer un jeu de données diversifié.

Table 3.4 Paramètres numériques de la Phase 2

$\omega$	$\varphi_1$	$U_{\max}^1$	$U_{\max}^2$	$q_{12}$	$\rho$	$c_1^+$	$c_2^+$
0.005	$-\frac{\pi}{2}$	1.5	1.25	0.05	0.01	1.0	0.5
$d_{m1}$	$d_{a1}$	$d_{m2}$	$d_{a2}$	$q_{21}$	$c^-$		
[0.51, 0.95]	[0.025, 0.075]	[0.4, 0.8]	[0.025, 0.075]	[0.15, 0.30]	[10, 50]		

#### 3.5.1 Comparaison des performances kNN et DNN

Cette section compare les performances des modèles kNN et DNN à partir de métriques quantitatives : l'erreur absolue moyenne (MAE) pour les trajectoires optimales  $z_1^*(t)$  et  $z_2^*(t)$ , et l'erreur relative moyenne pour le coût optimal. Ces indicateurs permettent de déterminer objectivement lequel des deux algorithmes offre la meilleure précision prédictive.

##### 3.5.1.1 Performances du kNN

Après optimisation des hyperparamètres, le modèle kNN retenu utilise un nombre de voisins fixé à  $k = 7$ . La qualité de la prédiction du coût optimal  $C$  sur l'ensemble de test est illustrée à la

Figure 3.12, qui compare, pour chaque configuration, la valeur issue des équations d'HJB et celle fournie par le modèle. La droite en pointillés représente la prédiction parfaite ( $y = x$ ) : la majorité des points restant proches de cette bissectrice, le kNN reproduit correctement le coût optimal de référence.

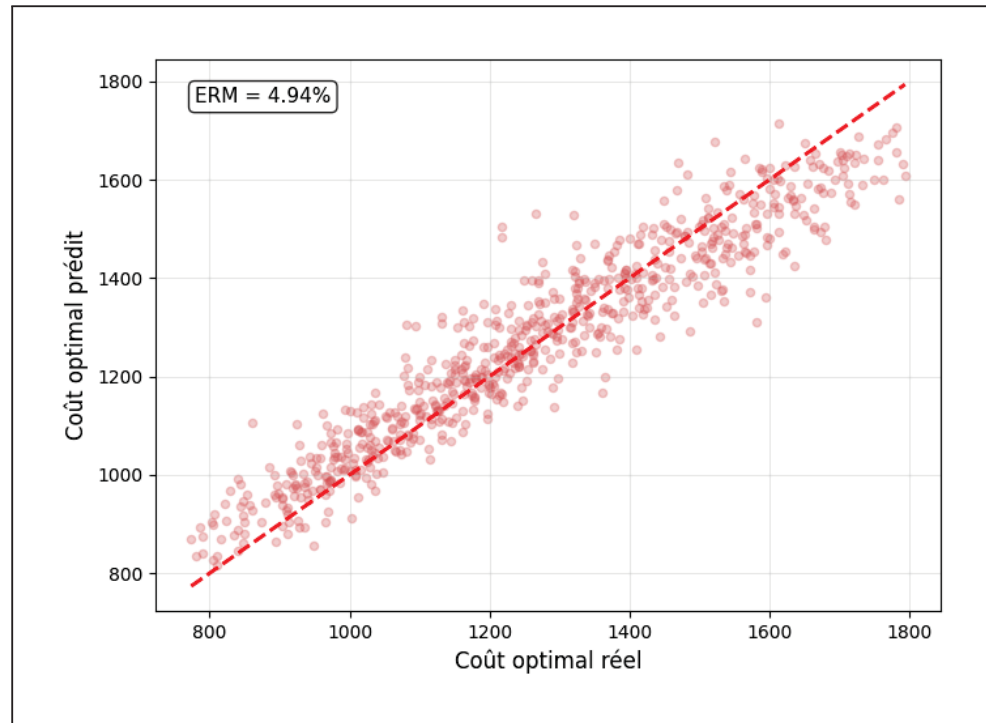


Figure 3.12 Performances du modèle kNN ( $k = 7$ ) pour la prédiction du coût optimal  $C$  sur l'ensemble de test

Les Figures 3.13 et 3.14 présentent les résultats obtenus pour les trajectoires des seuils critiques de stock  $z_1^*(t)$  et  $z_2^*(t)$ . Chaque point correspond à une valeur de seuil à un instant donné, comparant la prédiction du kNN à la valeur issue de la résolution des équations d'HJB. Là encore, la concentration des points autour de la diagonale témoigne de la bonne capacité du modèle à reproduire les trajectoires optimales.

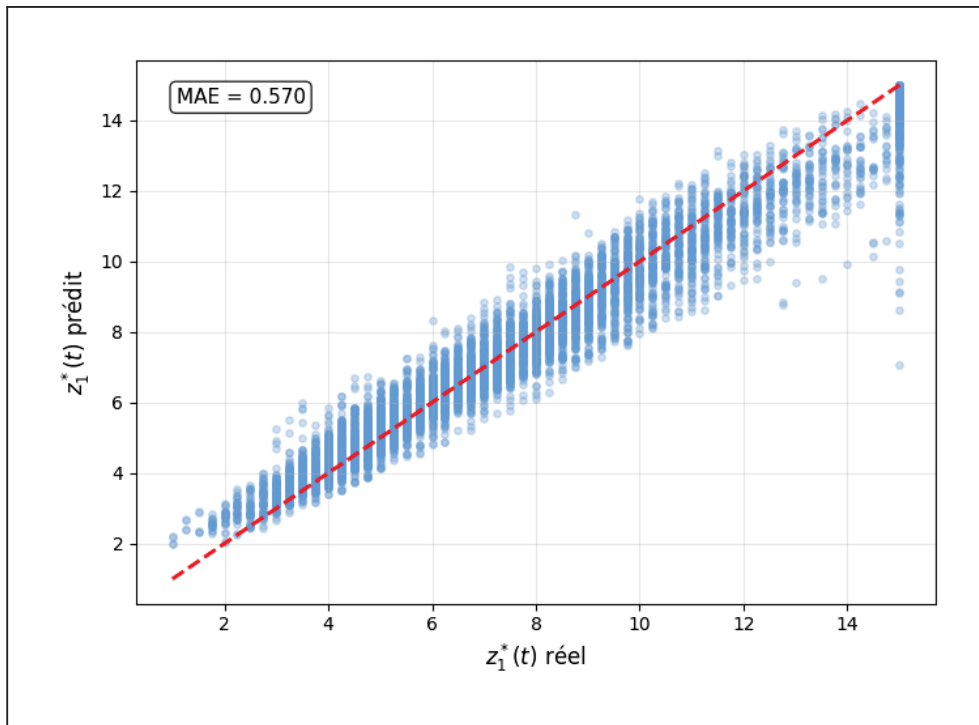


Figure 3.13 Performances du modèle kNN pour la prédiction de la trajectoire  $z_1^*(t)$  sur l'ensemble de test

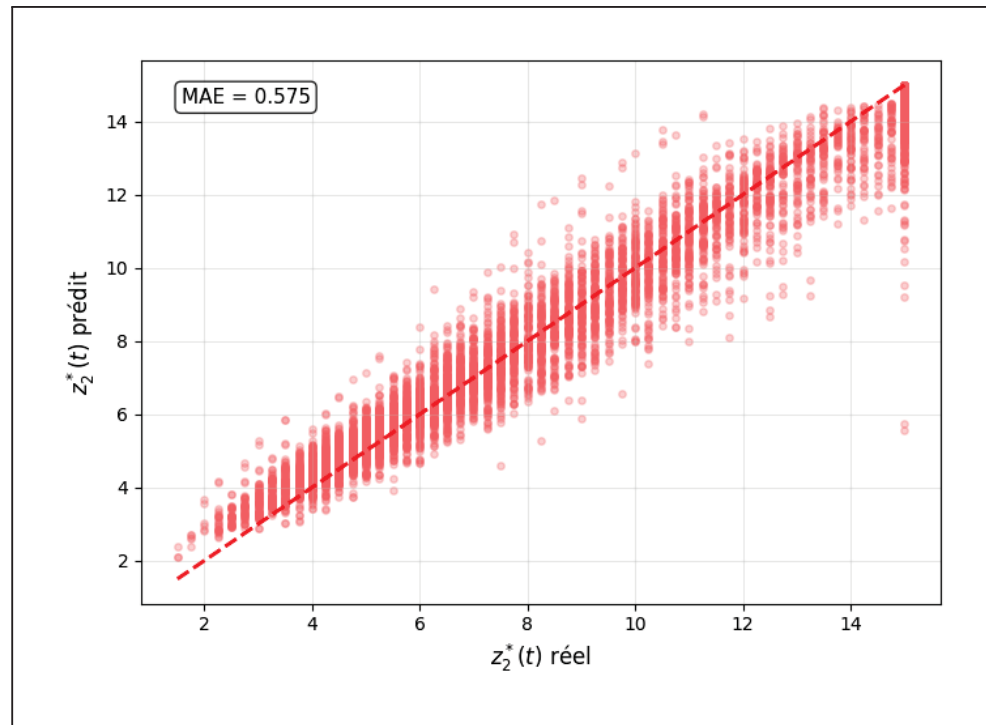


Figure 3.14 Performances du modèle kNN pour la prédiction de la trajectoire  $z_2^*(t)$  sur l'ensemble de test

Les métriques quantitatives confirment ces observations : l'erreur absolue moyenne (MAE) vaut 0,570 pour  $z_1^*(t)$  et 0,575 pour  $z_2^*(t)$ , tandis que l'erreur relative moyenne sur le coût optimal  $C$  est de 4,94 %. Ces résultats montrent que le modèle kNN fournit un surrogate suffisamment précis pour représenter les sorties optimales du système bi-produits dans cette deuxième phase d'analyse.

### 3.5.1.2 Performances du DNN

Après optimisation des hyperparamètres et validation croisée, le modèle DNN retenu présente la structure synthétisée dans le tableau 3.5, comprenant cinq couches, les fonctions d'activation employées, les stratégies de normalisation et de régularisation, ainsi que l'optimiseur utilisé pour l'entraînement.

Table 3.5 Hyperparamètres et structure du modèle DNN

Hyperparamètre	Valeur
Architecture du réseau	5 couches : entrée (6), cachées (256, 128, 64, 32), sortie (21)
Fonction d'activation	ReLU pour les couches cachées, linéaire pour la sortie
Normalisation	Batch Normalization après chaque couche dense
Régularisation	Dropout (30%, 25%, 20%, 15%)
Optimiseur	Adam (taux d'apprentissage initial = 0,001)

La Figure 3.15 illustre les performances du modèle DNN optimisé pour la prédiction de la trajectoire du seuil critique  $z_1^*(t)$  sur l'ensemble de test. Chaque point compare une valeur prédite à la valeur de référence issue des équations d'HJB, la droite en pointillés représentant la prédiction parfaite ( $y = x$ ). La concentration des points autour de cette diagonale montre que le modèle parvient à reproduire fidèlement la dynamique optimale du premier produit.

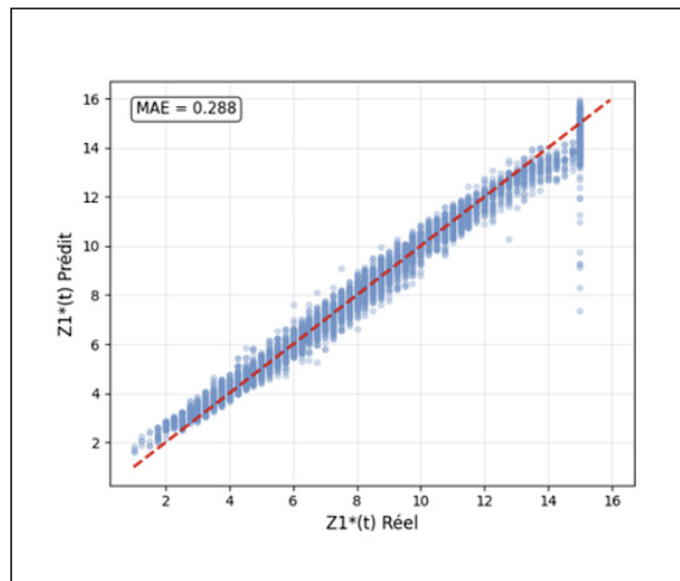


Figure 3.15 Performances du modèle DNN pour la prédiction de la trajectoire  $z_1^*(t)$  sur l'ensemble de test

La Figure 3.16 présente un résultat analogue pour la trajectoire du seuil  $z_2^*(t)$ . Là encore, les points restent globalement proches de la bissectrice, indiquant que le DNN capture correctement la forme des trajectoires optimales associées au second produit. Les erreurs absolues moyennes sont de 0,288 pour  $z_1^*(t)$  et de 0,303 pour  $z_2^*(t)$ , ce qui traduit une bonne précision sur les niveaux de stock recommandés.

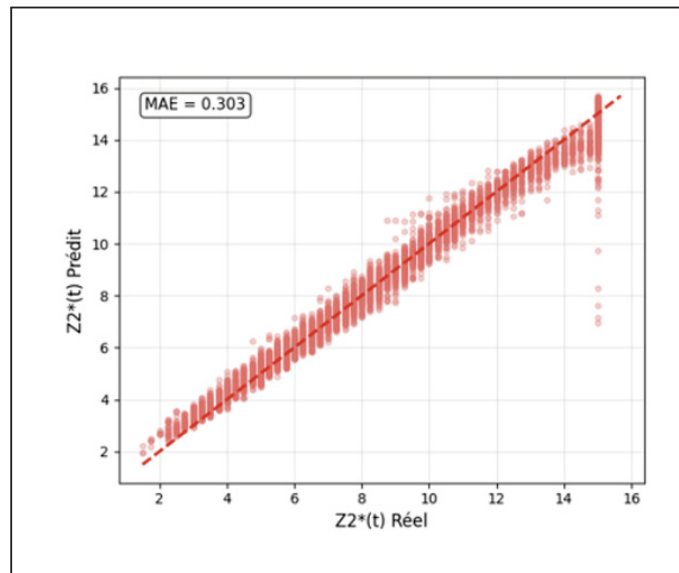


Figure 3.16 Performances du modèle DNN pour la prédiction de la trajectoire  $z_2^*(t)$  sur l'ensemble de test

La Figure 3.17 rend compte des performances du DNN pour la prédiction du coût optimal global  $C$ . Les points y restent également bien alignés autour de la droite  $y = x$ , avec une erreur relative moyenne de 3,62 % sur l'ensemble de test. Comparé au kNN, le DNN améliore l'ensemble des métriques, tant sur les trajectoires de  $z_1^*(t)$  et  $z_2^*(t)$  que sur le coût optimal.

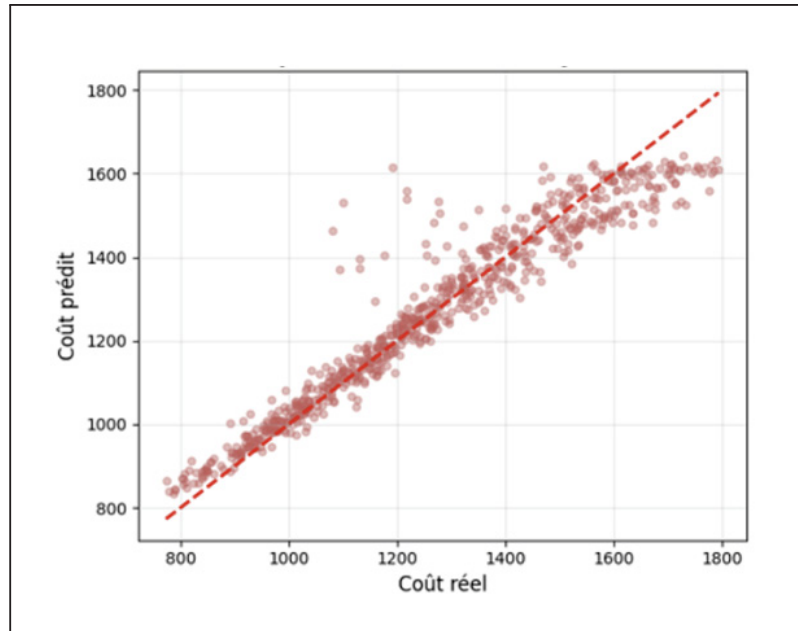


Figure 3.17 Performances du modèle DNN pour la prédiction du coût optimal  $C$  sur l'ensemble de test

La supériorité du DNN peut s'expliquer par la « malédiction de la dimensionnalité » qui pénalise les méthodes fondées sur des distances, comme le kNN : lorsque le nombre de variables d'entrée augmente, les distances entre points deviennent moins discriminantes et les voisins les plus proches ne sont plus nécessairement représentatifs (Murphy, 2012; Pestov, 2011). Avec six variables d'entrée, le DNN surmonte cette limitation en apprenant des représentations latentes et des corrélations complexes dans l'espace des données, sans se reposer exclusivement sur une métrique géométrique (Goodfellow *et al.*, 2016). Ces résultats confirment la pertinence des réseaux de neurones profonds dans des contextes fortement multidimensionnels, alors que le kNN, bien que performant en faible dimension, montre ici ses limites. Le modèle DNN est donc retenu pour la suite des analyses.

### 3.5.2 Analyse de scénarios et validation

Cette section évalue la pertinence du modèle DNN sur des cas spécifiques, en comparant les prédictions aux solutions de référence HJB. L'analyse débute par un scénario particulier

illustrant les trajectoires optimales et les coûts associés, puis s'étend à 10 scénarios extraits des 50 scénarios de l'annexe III.

### 3.5.2.1 Scénario particulier

Pour illustrer le comportement du modèle DNN sur un cas concret, on considère un scénario caractérisé par  $d_{m1} = 0.503$ ,  $d_{a1} = 0.060$ ,  $d_{m2} = 0.631$ ,  $d_{a2} = 0.059$ ,  $c^- = 50$ ,  $q_{21} = 0.260$ . La Figure 3.18 compare, pour le produit 1, la trajectoire du seuil critique  $z_1^*(t)$  obtenue par résolution HJB (référence) à celle prédite par le modèle DNN. Les deux courbes présentent des profils temporels très proches, avec des écarts de niveau limités sur l'ensemble de l'horizon considéré.

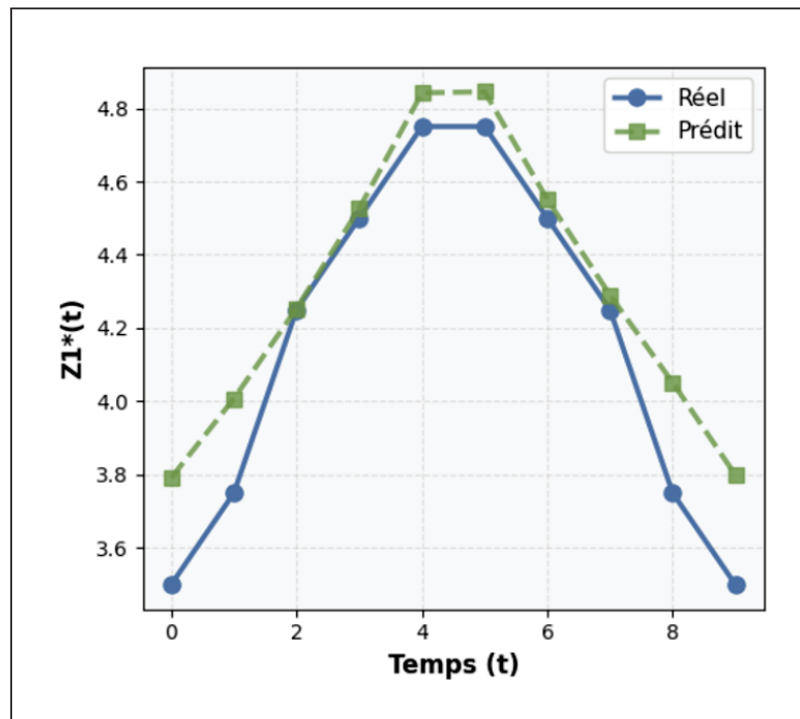


Figure 3.18 Scénario particulier : comparaison des trajectoires de  $z_1^*(t)$  obtenues par HJB (référence) et prédites par le modèle DNN

La Figure 3.19 présente la même comparaison pour le seuil critique du produit 2,  $z_2^*(t)$ . Là encore, la dynamique des trajectoires est correctement restituée : les phases, maxima et minima

sont bien alignés entre la solution HJB et la prédiction du DNN, ce qui traduit une bonne capacité du modèle à reproduire la politique optimale sur ce scénario.

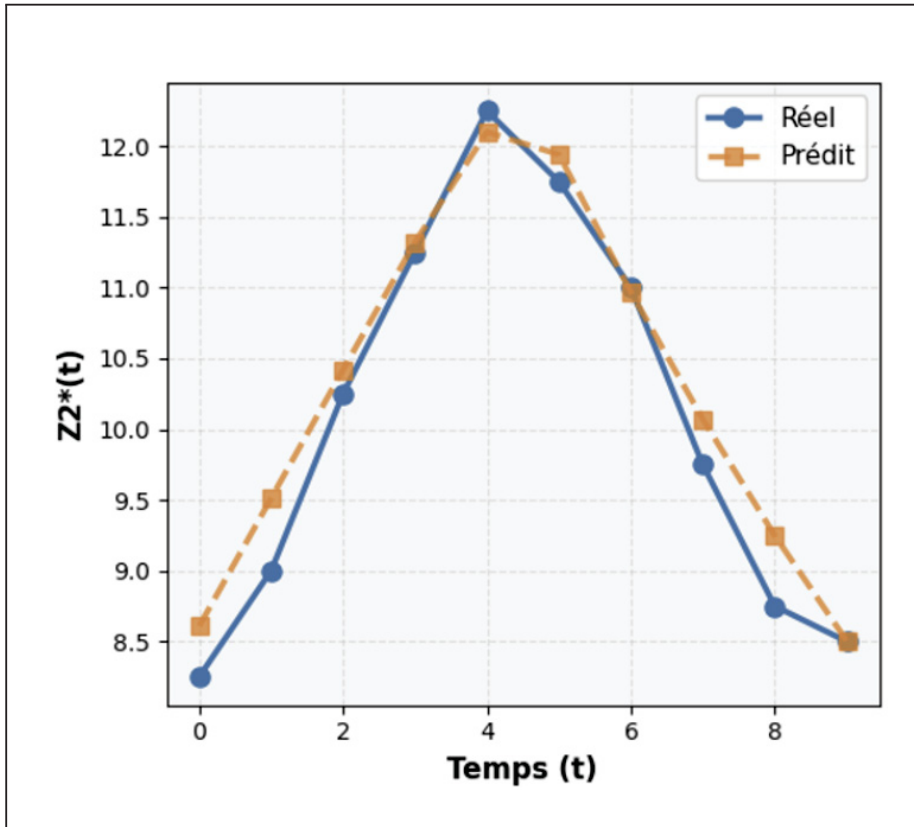


Figure 3.19 Scénario particulier : comparaison des trajectoires de  $z_2^*(t)$  obtenues par HJB (référence) et prédites par le modèle DNN

Cette fidélité se retrouve au niveau des coûts : le coût optimal calculé par HJB vaut 1109,3, contre 1114,7 pour la prédiction du DNN, soit une erreur relative de seulement 0,49 %. Ce résultat confirme que, pour ce type de configuration, le modèle DNN est capable de produire des décisions de production très proches de la solution optimale, tant du point de vue des trajectoires de stock que de leur impact économique.

### 3.5.2.2 Validation du modèle

L'évaluation est étendue à 10 scénarios représentatifs extraits des 50 scénarios de l'annexe III, générés par variation des paramètres d'entrée. Le tableau 3.6 présente pour chaque configuration la MAE sur  $z_1^*(t)$  et  $z_2^*(t)$ , le coût optimal HJB, le coût prédit et l'erreur relative. Sur l'ensemble des 50 scénarios, le modèle DNN conserve de très bonnes performances avec une MAE moyenne de 0,26 pour  $z_1^*(t)$ , 0,29 pour  $z_2^*(t)$ , et une erreur relative moyenne de 1,62 % pour le coût optimal, confirmant sa robustesse et sa capacité généralisante. L'avantage majeur du DNN réside dans sa rapidité d'exécution : alors que la résolution numérique HJB demande plusieurs minutes par scénario, la prédiction s'effectue en moins d'une seconde, permettant une exploration rapide de nombreuses configurations sans recalcul coûteux.

Table 3.6 Performances du modèle DNN sur 10 scénarios représentatifs : MAE sur  $z_1^*$  et  $z_2^*$ , coûts HJB et prédits, erreur relative

$dm_1$	$da_1$	$dm_2$	$da_2$	$c^-$	$q_{21}$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	ERM (%)
0.752	0.066	0.564	0.066	34	0.225	0.451	0.221	1478.05	0.10
0.684	0.070	0.792	0.034	49	0.255	0.234	0.245	1515.00	5.01
0.675	0.033	0.749	0.059	16	0.235	0.176	0.300	1161.52	1.11
0.556	0.033	0.523	0.047	21	0.235	0.266	0.109	903.02	0.01
0.609	0.070	0.653	0.073	40	0.300	0.168	0.279	1012.49	5.04
0.769	0.045	0.525	0.058	28	0.230	0.113	0.084	1342.29	1.11
0.889	0.058	0.402	0.072	22	0.150	0.474	0.720	1595.37	0.02
0.770	0.040	0.531	0.053	25	0.200	0.127	0.089	1524.60	0.01
0.705	0.028	0.733	0.036	33	0.280	0.245	0.260	1214.76	5.00
0.586	0.039	0.782	0.026	29	0.285	0.078	0.314	1073.76	5.07

### 3.5.3 Analyses de sensibilité

Cette section approfondit la validation du modèle en étudiant sa sensibilité aux paramètres clés  $c^-$  et  $q_{21}$ . L'approche consiste à analyser le comportement des politiques optimales et des coûts

lorsqu'un paramètre varie, les autres restant constants. Cette démarche complémentaire vérifie la capacité du modèle DNN à restituer fidèlement les tendances attendues en cohérence avec les solutions HJB.

### 3.5.3.1 Analyse de sensibilité sur le coût de rupture de stock

L'impact du coût unitaire de rupture de stock  $c^-$  est analysé en faisant varier ce paramètre dans l'intervalle  $[13, 21]$ , les autres paramètres restant fixes :  $d_{m1} = 0.874$ ,  $d_{m2} = 0.534$ ,  $d_{a1} = 0.033$ ,  $d_{a2} = 0.063$  et  $q_{21} = 0.225$ . L'objectif est de vérifier si le modèle DNN reproduit l'effet économique attendu : une augmentation de  $c^-$  doit conduire à des niveaux de stock plus élevés afin de limiter les ruptures.

La Figure 3.20 montre l'évolution de la trajectoire optimale  $z_1^*(t)$  en fonction de  $c^-$ , pour la solution de référence (HJB) et pour les prédictions du DNN. On observe une augmentation nette du seuil critique lorsque  $c^-$  passe de 13 à 21, ce qui traduit un renforcement de la politique de stockage pour le premier produit.

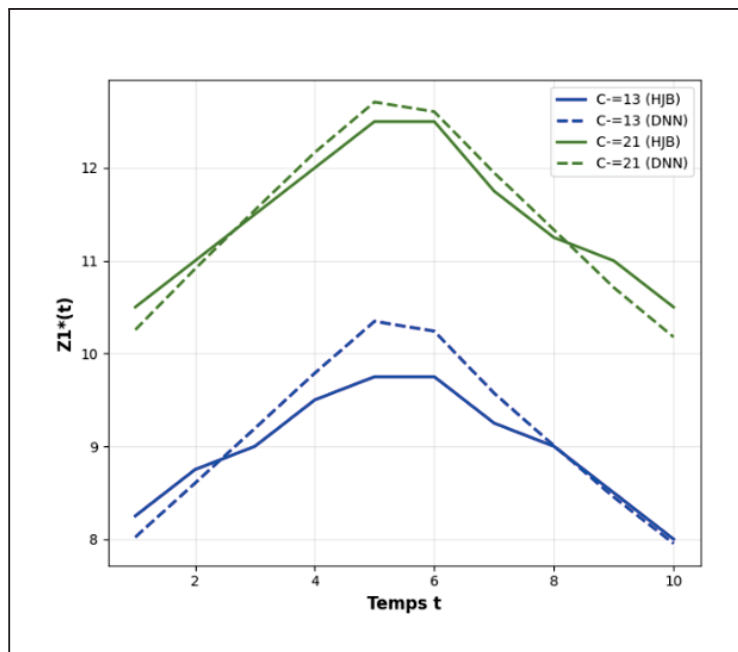


Figure 3.20 Sensibilité de la trajectoire  $z_1^*(t)$  aux variations de  $c^-$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

La Figure 3.21 présente le même type d'analyse pour la trajectoire  $z_2^*(t)$ . Là encore, les courbes HJB et DNN restent très proches, et l'augmentation de  $c^-$  se traduit par des niveaux de stock plus élevés pour le second produit, conformément à l'intuition économique.

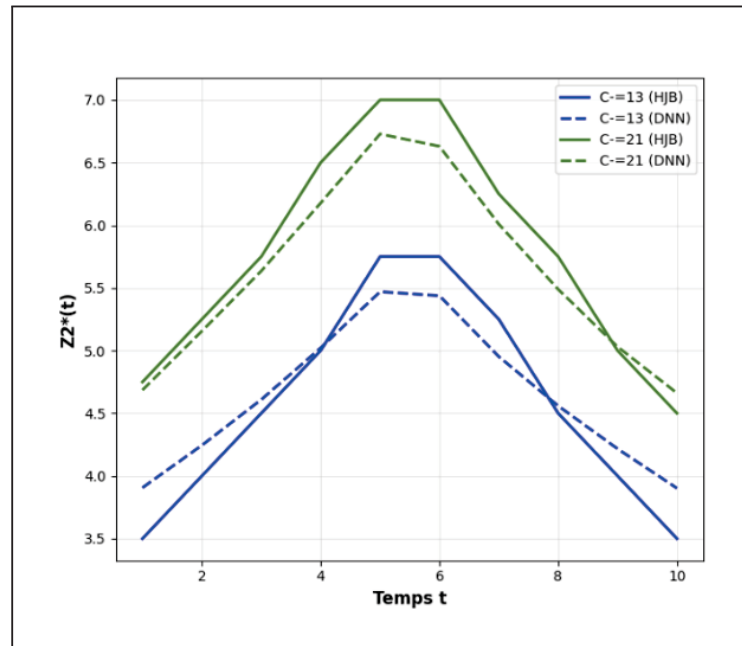


Figure 3.21 Sensibilité de la trajectoire  $z_2^*(t)$  aux variations de  $c^-$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

La Figure 3.22 illustre enfin l'évolution du coût optimal en fonction de  $c^-$ . L'augmentation du coût unitaire de rupture de 13 à 21 (soit +62 %) entraîne une hausse marquée des niveaux de stock, de l'ordre de 25 à 30 % pour  $z_1^*(t)$  et de 20 à 25 % pour  $z_2^*(t)$ , ce qui permet de réduire la fréquence des ruptures. Sur tout le domaine étudié, l'erreur relative entre les coûts prédits par le DNN et ceux obtenus par HJB reste très faible, comprise entre 0 % et 1,2 %. Cette concordance entre les courbes de référence et les prédictions confirme la robustesse du DNN pour analyser rapidement l'impact des variations de  $c^-$  sur la politique optimale et sur le coût global.

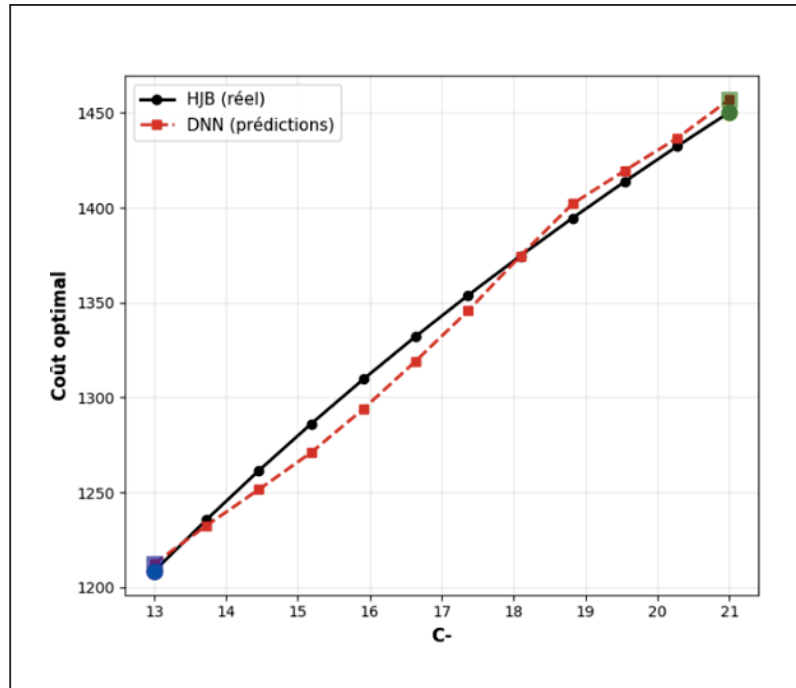


Figure 3.22 Sensibilité du coût optimal aux variations de  $c^-$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

### 3.5.3.2 Analyse de sensibilité par rapport au taux de réparation

L'impact du taux de réparation  $q_{21}$  est analysé en faisant varier ce paramètre dans l'intervalle  $[0.20, 0.25]$ , les autres paramètres restant fixes :  $d_{m1} = 0.58$ ,  $d_{m2} = 0.71$ ,  $d_{a1} = 0.072$ ,  $d_{a2} = 0.035$  et  $c^- = 19$ . L'objectif est de vérifier si le DNN reproduit l'effet attendu : une augmentation de  $q_{21}$  doit conduire à une réduction des niveaux de stock optimaux, des réparations plus rapides diminuant l'intérêt de stratégies de stockage préventif coûteuses.

La Figure 3.23 présente la sensibilité de la trajectoire  $z_1^*(t)$  aux variations de  $q_{21}$ , en comparant la solution de référence (HJB) et les prédictions du DNN.

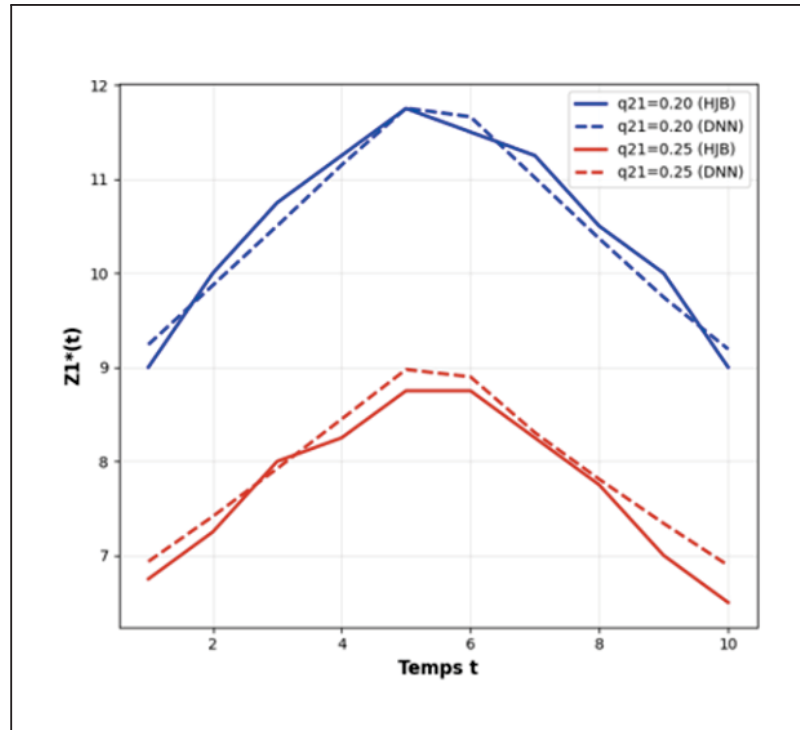


Figure 3.23 Sensibilité de la trajectoire  $z_1^*(t)$  aux variations de  $q_{21}$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

La Figure 3.24 montre un résultat analogue pour la trajectoire  $z_2^*(t)$ . Dans les deux cas, l'augmentation du taux de réparation de  $q_{21} = 0.20$  à  $q_{21} = 0.25$  (+25 %) entraîne une diminution significative des niveaux de stock : de l'ordre de 20 à 25 % pour  $z_1^*(t)$  et de 15 à 20 % pour  $z_2^*(t)$ . Cette tendance est conforme à l'intuition : un système plus rapidement réparable nécessite des stocks plus faibles pour assurer le même niveau de service.

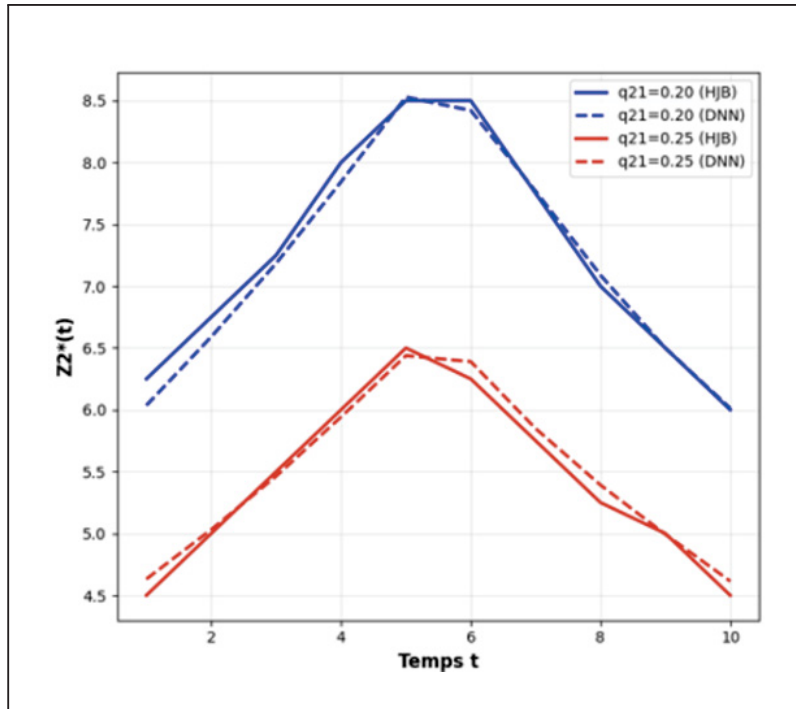


Figure 3.24 Sensibilité de la trajectoire  $z_2^*(t)$  aux variations de  $q_{21}$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

La Figure 3.25 illustre l'évolution du coût optimal en fonction de  $q_{21}$ , toujours en comparant les valeurs issues des HJB et celles prédites par le DNN. Sur l'ensemble du domaine étudié, l'erreur relative entre les coûts prédits et les coûts de référence reste exceptionnellement faible, comprise entre 0 % et 0,5 %. Cette concordance confirme la précision de l'approximation neuronale pour l'évaluation de l'impact du taux de réparation sur le coût global.

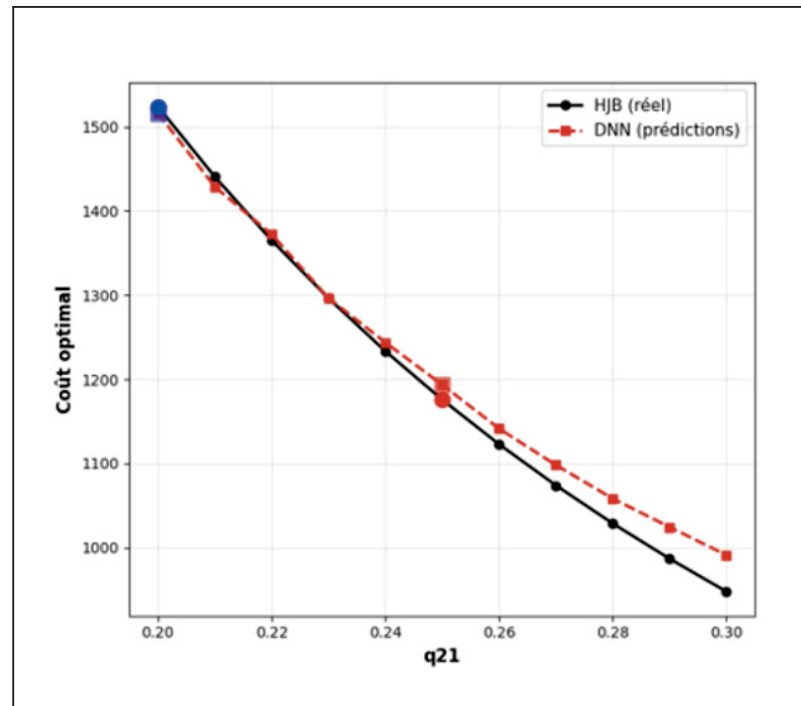


Figure 3.25 Sensibilité du coût optimal aux variations de  $q_{21}$  : comparaison entre la solution de référence (HJB) et les prédictions du modèle DNN

Les analyses de sensibilité confirment ainsi la robustesse du modèle DNN pour la prédiction des paramètres optimaux. Que ce soit pour le coût de rupture  $c^-$  ou pour le taux de réparation  $q_{21}$ , le réseau reproduit correctement les tendances attendues : augmentation des niveaux de stock lorsque les ruptures deviennent plus coûteuses, diminution lorsque le système est plus réparable. Les erreurs relatives restent très faibles sur l'ensemble des domaines étudiés ( $< 1,2\%$  pour  $c^-$  et  $< 0,5\%$  pour  $q_{21}$ ), en cohérence avec les résultats obtenus lors de l'analyse des scénarios (erreur relative moyenne de  $1,62\%$ ). Ces éléments valident l'utilisation du DNN comme alternative efficace à la résolution HJB pour des analyses paramétriques rapides et l'aide à la décision en contexte industriel.

### 3.5.4 Mise en œuvre des politiques de production optimales

Une fois les paramètres optimaux obtenus à partir du modèle DNN, le décideur peut les traduire en décisions de production concrètes pour chaque produit. Les six variables d'entrée

$(d_{m1}, d_{a1}, d_{m2}, d_{a2}, c^-, q_{21})$  sont fournies au modèle, qui prédit les courbes critiques optimales  $z_1^*(t)$  et  $z_2^*(t)$ , ainsi que le coût optimal  $C$ .

Considérons un scénario caractérisé par  $d_{m1} = 0.5$ ,  $d_{a1} = 0.066$ ,  $d_{m2} = 0.49$ ,  $d_{a2} = 0.059$ ,  $q_{21} = 0.15$  et  $c^- = 15$ , les autres paramètres étant ceux du Tableau 3.4. Les trajectoires optimales prédites par le DNN sont illustrées aux Figures 3.26 et 3.27 : la première montre l'évolution de  $z_1^*(t)$ , la seconde celle de  $z_2^*(t)$ .

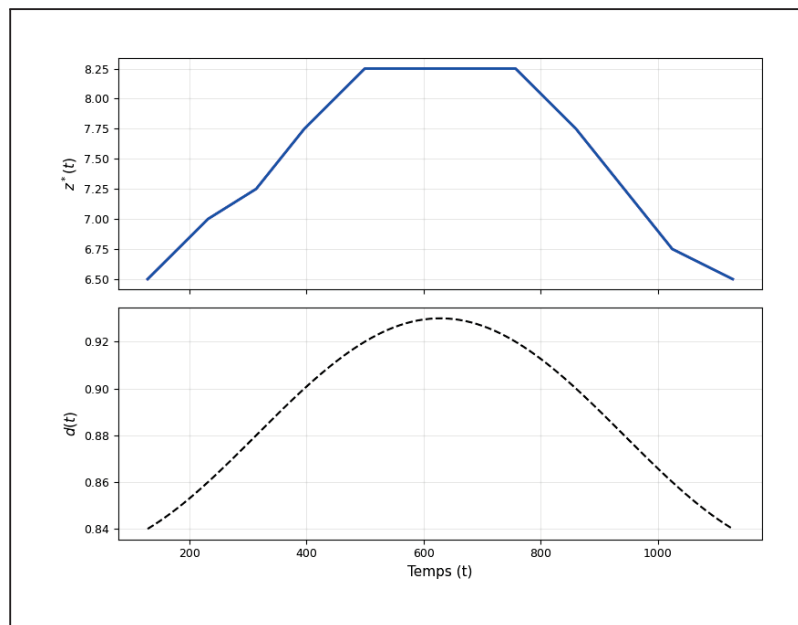


Figure 3.26 Évolution de la trajectoire optimale  $z_1^*(t)$  prédite par le modèle DNN.

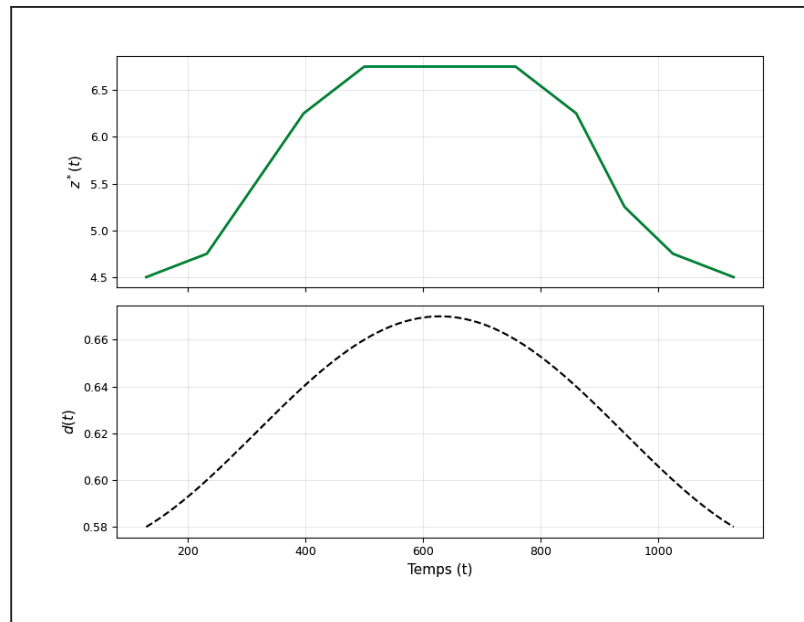


Figure 3.27 Évolution de la trajectoire optimale  $z^*(t)$  prédite par le modèle DNN

À l'instant  $t = 600$ , le modèle prédit par exemple  $z_1^*(600) \approx 5.5$  et  $z_2^*(600) \approx 9.0$ . La Figure 3.28 illustre le processus de décision permettant de déterminer les taux de production lorsque les niveaux de stock observés sont  $x_1(600) = 5.0$  et  $x_2(600) = 9.5$ .

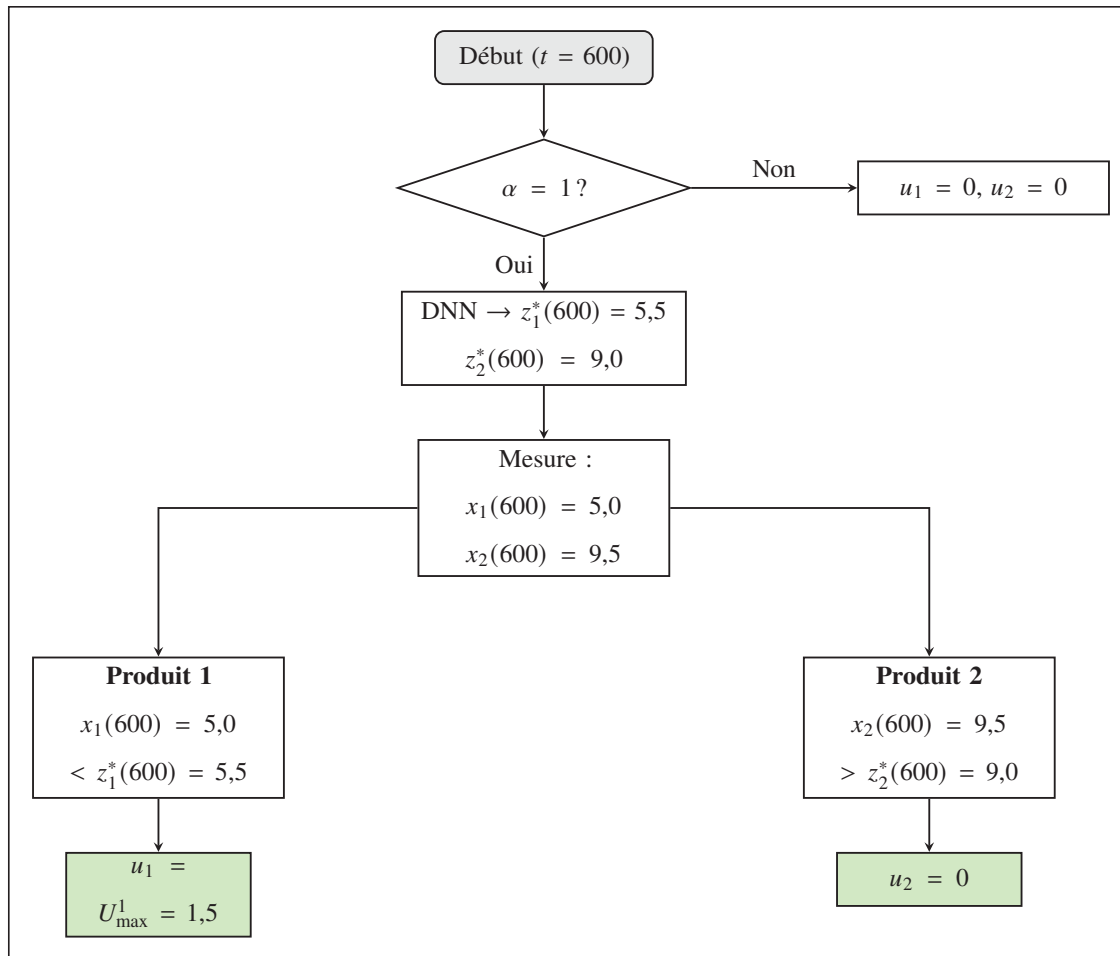


Figure 3.28 Processus de décision pour la mise en œuvre de la politique de production optimale à  $t = 600$

Le processus de décision se déroule alors en quatre étapes :

1. **Vérification de l'état de la machine.**

On identifie d'abord le mode de fonctionnement de la machine : si  $\alpha = 1$ , la machine est opérationnelle ; si  $\alpha = 2$ , elle est en panne. En cas de panne, aucune production n'est possible et l'on fixe  $u_1 = u_2 = 0$ .

2. **Récupération des seuils optimaux auprès du DNN.**

Si la machine est opérationnelle, le modèle DNN fournit les seuils critiques optimaux à l'instant considéré, ici  $z_1^*(600) = 5,5$  et  $z_2^*(600) = 9,0$ .

### 3. Mesure des niveaux de stock.

Les niveaux de stock réels sont ensuite mesurés. Dans cet exemple, on observe  $x_1(600) = 5,0$  et  $x_2(600) = 9,5$ .

### 4. Décision de production pour chaque produit.

Pour chaque produit, on compare le stock observé au seuil optimal :

- Produit 1 :  $x_1(600) = 5,0 < z_1^*(600) = 5,5 \Rightarrow$  on produit au maximum,  $u_1 = U_{\max}^1 = 1,5$ .
- Produit 2 :  $x_2(600) = 9,5 > z_2^*(600) = 9,0 \Rightarrow$  on arrête la production,  $u_2 = 0$ .

Cette politique permet de ramener progressivement les stocks vers leurs niveaux optimaux, en limitant à la fois les coûts de possession et les coûts de rupture, tout en tenant compte des fluctuations de la demande. L'exemple illustre la capacité du modèle à guider des décisions simultanées mais différenciées pour chaque produit, en fonction de l'écart entre le stock observé et le seuil optimal à chaque instant du cycle de demande.

## 3.5.5 Domaine de validité de l'approche

L'approche proposée est développée et validée dans un cadre bien défini, ce qui permet d'en préciser à la fois le périmètre d'application et l'apport pratique.

### 3.5.5.1 Domaine de validité du modèle d'apprentissage

Le DNN fournit des prédictions fiables principalement en tant qu'*interpolateur* à l'intérieur du domaine d'entrée couvert par l'ensemble d'apprentissage, défini par les plages de paramètres reportées au Tableau 3.4. Ce point est essentiel d'un point de vue industriel : dans ces intervalles, le modèle peut être interrogé en temps réel pour soutenir des analyses *what-if* rapides, tout en restant ancré aux solutions optimales issues des équations HJB utilisées comme référence lors de l'entraînement. Lorsque les entrées sortent de ce domaine, le modèle opère en *extrapolation*, pour laquelle la précision n'est pas garantie. Dans ce cas, une mesure de sûreté simple consiste à vérifier que chaque paramètre appartient aux bornes d'apprentissage ; sinon, des résolutions

HJB supplémentaires peuvent être réalisées afin d'étendre la base de données et de mettre à jour le modèle substitut.

### 3.5.5.2 Hypothèses de modélisation

Les résultats sont obtenus sous un ensemble structuré d'hypothèses qui définit la classe de systèmes étudiée. La demande est modélisée, pour les deux produits, par une fonction sinusoïdale variant dans le temps, et l'on suppose des cycles de demande synchronisés avec des fréquences égales ( $\omega_1 = \omega_2$ ) et des phases identiques ( $\phi_1 = \phi_2$ ). Ce choix offre un cadre d'évaluation contrôlé et reproductible pour la politique à seuils critiques anticipative, et il est cohérent avec des contextes où plusieurs produits sont influencés par des schémas saisonniers communs, des effets calendaires et des actions promotionnelles coordonnées (Fildes *et al.* (2022); Wolters *et al.* (2021)). Des cadres plus généraux, avec des fréquences et des phases distinctes, sont laissés à des travaux futurs.

### 3.5.5.3 Compromis de calcul et lien avec les contributions

La génération de la base de données requiert une phase initiale fondée sur la résolution numérique des équations HJB, qui peut prendre plusieurs heures selon les réglages de grille et l'échantillonnage paramétrique. Ce coût ponctuel est précisément celui que le cadre substitut proposé vise à amortir. Une fois entraîné, le DNN remplace les résolutions HJB répétées par des prédictions quasi instantanées des paramètres de la politique optimale (seuils critiques et coût optimal), ce qui rend possibles des analyses de scénarios et de sensibilité à grande échelle. En ce sens, l'approche opérationnalise les contributions principales de ce travail : (i) une réduction drastique du temps de calcul pour l'exploration multi-paramétrique et (ii) une méthodologie reliant le choix de l'algorithme à la taille de l'ensemble de variables d'entrée, le kNN convenant au cadre de plus faible dimension et le DNN offrant de meilleures performances lorsque des paramètres supplémentaires sont introduits.

### 3.5.6 Analyse et discussion

Cette section propose une discussion transversale des résultats obtenus aux chapitres 2 et 3, en mettant en perspective les choix algorithmiques, l'effet de la dimensionnalité sur les performances prédictives et le positionnement de l'approche développée par rapport à la littérature existante.

#### 3.5.6.1 Synthèse comparative des algorithmes à travers les deux systèmes

Les expériences menées dans ce mémoire ont mobilisé trois algorithmes de régression supervisée — le kNN, le RBF et le DNN — évalués dans des contextes de complexité croissante. Le Tableau 3.7 rassemble les principales métriques de performance obtenues au fil des différentes phases expérimentales, permettant une lecture comparative directe. Plusieurs constats se

Table 3.7 Synthèse des performances des algorithmes selon le système et le nombre de variables d'entrée

Système	Algorithme	Variables	MAE $z^*$	ERM coût (%)	Référence
Mono-produit (Phase 1)	DNN	3	1,45	0,88	Tableau 2.3
	RBF	3	0,41	4,35	Tableau 2.3
	kNN	3	0,07	0,32	Tableau 2.3
Mono-produit (Phase 2)	kNN	4	—	2,22 <sup>a</sup>	Section 2.6
Bi-produits (Phase 1)	kNN	4	0,32 / 0,34 <sup>b</sup>	2,20	Section 3.4
Bi-produits (Phase 2)	kNN	6	0,57 / 0,58 <sup>b</sup>	4,94	Section 3.5
	DNN	6	0,29 / 0,30 <sup>b</sup>	3,62	Section 3.5

<sup>a</sup> Erreur relative moyenne sur 50 scénarios de validation.

<sup>b</sup> MAE pour  $z_1^*(t)$  /  $z_2^*(t)$  respectivement.

dégagent de cette mise en parallèle. Premièrement, dans le système mono-produit avec trois variables d'entrée ( $d_m, d_a, \omega$ ), le kNN présente les meilleures performances parmi les trois algorithmes testés, avec une MAE de 0,07 pour  $z^*$  et une erreur relative de 0,32 % sur le coût optimal, devant le RBF (MAE de 0,41 ; ERM de 4,35 %) et le DNN (MAE de 1,45 ; ERM de 0,88 %). Ce résultat s'explique par la nature du problème en dimension modérée : l'espace des entrées reste suffisamment compact pour que la notion de voisinage, sur laquelle repose le kNN, demeure discriminante. Le kNN exploite directement la structure locale des données sans imposer d'hypothèse paramétrique sur la forme de la relation entrées-sorties, ce qui lui

confère un avantage lorsque les données sont suffisamment denses relativement à la dimension de l'espace.

Deuxièmement, le comportement du RBF mérite une attention particulière. Ce modèle obtient une meilleure MAE que le DNN sur  $z^*(t)$  (0,41 contre 1,45), ce qui témoigne de sa capacité à capter les variations locales de la trajectoire optimale. Toutefois, son erreur relative sur le coût optimal est la plus élevée des trois algorithmes (4,35 %). Ce profil de performance contrasté suggère que le RBF reproduit la forme des trajectoires avec une précision intermédiaire, mais accumule des erreurs lorsqu'il s'agit de prédire une grandeur agrégée comme le coût, qui intègre l'ensemble du comportement dynamique du système sur l'horizon considéré.

Troisièmement, le passage du système mono-produit au système bi-produits avec quatre variables d'entrée confirme la robustesse du kNN : les MAE sur  $z_1^*(t)$  et  $z_2^*(t)$  restent modérées (0,32 et 0,34) et l'erreur relative sur le coût est de 2,20 %. Le kNN conserve donc des performances satisfaisantes tant que la dimension de l'espace d'entrée demeure modérée, ce qui est cohérent avec les propriétés théoriques de cet algorithme.

### 3.5.6.2 Effet de la dimensionnalité et seuil de transition algorithmique

Le résultat le plus structurant de ce mémoire est la mise en évidence d'un seuil de transition algorithmique lié à la dimensionnalité du problème. Lorsque le nombre de variables d'entrée passe de quatre à six, par l'ajout du coût de rupture de stock  $c^-$  et du taux de réparation  $q_{21}$ , les performances du kNN se dégradent de manière significative : la MAE sur  $z_1^*(t)$  passe de 0,32 à 0,57 et l'erreur relative sur le coût augmente de 2,20 % à 4,94 %. Dans le même contexte, le DNN atteint une MAE de 0,29 pour  $z_1^*(t)$  et une erreur relative de 3,62 %, prenant ainsi le relais du kNN.

Ce phénomène est directement lié à la *malédiction de la dimensionnalité* (Murphy, 2012; Pestov, 2012) : en dimension élevée, les distances euclidiennes entre points deviennent moins discriminantes, de sorte que les voisins les plus proches ne sont plus nécessairement représentatifs de la structure locale des données. Le DNN surmonte cette limitation en apprenant des

représentations latentes et des corrélations non linéaires complexes dans l'espace des données, sans dépendre exclusivement d'une métrique géométrique (Goodfellow *et al.*, 2016).

Ce constat a une portée méthodologique importante. Il ne s'agit pas simplement d'affirmer qu'un algorithme est globalement meilleur qu'un autre, mais de montrer que le choix de l'algorithme d'apprentissage doit être guidé par la complexité dimensionnelle du problème considéré. Dans le cadre étudié, cette transition s'opère entre quatre et six variables d'entrée, ce qui fournit un repère pratique pour les systèmes manufacturiers de structure similaire. Pour des systèmes encore plus complexes — par exemple, des systèmes à trois produits ou davantage, ou intégrant des variables de maintenance — on peut raisonnablement anticiper que les réseaux de neurones profonds, voire des architectures plus spécialisées, deviendraient indispensables.

### **3.5.6.3 Positionnement par rapport à la littérature**

L'approche développée dans ce mémoire se distingue des travaux existants sur plusieurs aspects. Contrairement aux travaux de commande optimale classiques (Gharbi *et al.*, 2006; Rivera-Gómez *et al.*, 2013; Kouedou *et al.*, 2014), qui produisent des solutions pour des jeux de paramètres fixes sans mécanisme de généralisation, l'approche proposée construit un modèle substitut capable de prédire les paramètres optimaux pour toute configuration à l'intérieur du domaine d'apprentissage. Le coût de résolution des équations d'HJB est ainsi supporté une seule fois lors de la constitution de la base de données, puis amorti par les prédictions quasi instantanées du modèle entraîné.

Par rapport aux approches par apprentissage par renforcement profond (Petsagkourakis *et al.*, 2020a; Gijssbrechts *et al.*, 2022a; Paraschos *et al.*, 2024a), l'approche adoptée ici présente un avantage d'ancrage théorique : les données d'entraînement proviennent directement des solutions optimales des équations d'HJB, ce qui garantit que le modèle substitut apprend à reproduire des politiques dont l'optimalité est établie analytiquement. L'apprentissage par renforcement, en revanche, découvre les politiques par exploration, ce qui peut nécessiter un grand nombre

d'interactions et ne garantit pas toujours la convergence vers l'optimum global, en particulier pour des problèmes à horizon continu avec des dynamiques stochastiques complexes.

Enfin, à la différence des travaux qui appliquent le Machine Learning aux systèmes de production sans lien explicite avec les solutions de commande optimale (Bensaci *et al.*, 2021; Ho *et al.*, 2021; Priore *et al.*, 2019), ce mémoire propose un cadre méthodologique intégré qui relie explicitement la résolution numérique des équations d'HJB à l'apprentissage supervisé. Ce lien assure la cohérence entre la rigueur des solutions de référence et la flexibilité des modèles prédictifs, tout en fournissant des éléments de justification du choix algorithmique en fonction de la complexité du système.

En somme, les résultats obtenus confirment que l'apprentissage automatique, lorsqu'il est ancré à des solutions de commande optimale, constitue une approche à la fois rigoureuse et opérationnelle pour la prédiction des politiques de production sous demande périodique. Le choix de l'algorithme ne doit pas être figé a priori, mais adapté à la dimensionnalité du problème, selon un principe que les expériences de ce mémoire ont permis de quantifier et d'illustrer.

### **3.6 Conclusion**

Ce chapitre a établi le seuil de transition algorithmique guidé par la dimensionnalité du problème. L'algorithme k-Nearest Neighbors, efficace pour prédire les paramètres optimaux d'un système bi-produits lorsque l'on considère quatre variables d'entrée, voit ses performances se dégrader significativement lors du passage à six variables d'entrée. Les réseaux de neurones profonds s'imposent alors comme solution, atteignant une précision remarquable tout en maintenant un gain computationnel décisif par rapport à la résolution des équations d'HJB.

Combinés aux résultats du chapitre précédent, ces travaux démontrent que l'apprentissage automatique constitue une alternative viable et efficace pour contourner la complexité computationnelle des méthodes de contrôle optimal classiques, avec un choix algorithmique justifié par la complexité dimensionnelle du système considéré.

## CONCLUSION GÉNÉRALE

Ce mémoire s'est inscrit dans le contexte des systèmes de production soumis à une demande périodiquement variable, où la résolution numérique des équations de Hamilton–Jacobi–Bellman (HJB), bien que rigoureuse sur le plan théorique, reste lourde et difficile à exploiter pour l'exploration rapide de scénarios. Dans ce cadre, la politique optimale ne se réduit plus à un seuil fixe, mais prend la forme d'une trajectoire critique dépendante du temps, dont le calcul doit être intégralement répété dès qu'un paramètre du système varie. Cette contrainte limite fortement l'usage direct de la commande optimale stochastique dans un contexte industriel où la réactivité et la flexibilité sont essentielles. La question centrale posée par ce mémoire était donc la suivante : *comment prédire efficacement les paramètres d'une politique de production optimale dans un système manufacturier soumis à une demande périodiquement variable, tout en conservant la rigueur du contrôle optimal et en réduisant significativement le temps de calcul ?*

Le chapitre 1 a posé les fondements conceptuels et méthodologiques de ce travail. Il a rappelé le cadre de la commande optimale stochastique pour les systèmes de production soumis à des pannes et réparations aléatoires, en mettant en avant le rôle central des équations d'Hamilton–Jacobi–Bellman pour la détermination de politiques optimales. Il a ensuite présenté les principes de l'apprentissage supervisé et les principaux algorithmes étudiés (kNN, réseaux de neurones à base radiale, réseaux de neurones profonds), avant de proposer une revue critique de la littérature. Cette revue a mis en évidence un double constat : d'une part, la difficulté d'exploiter en pratique les solutions des équations d'HJB lorsque la demande varie dans le temps ; d'autre part, l'absence d'un cadre méthodologique unifié qui relie explicitement l'apprentissage automatique aux solutions de référence issues de la commande optimale, tout en tenant compte de la complexité des systèmes étudiés, pour le choix des algorithmes d'apprentissage.

Le chapitre 2 s'est concentré sur un système constitué d'une machine produisant un type de pièce, soumise à des pannes et réparations aléatoires et à une demande périodiquement variable. Dans ce cas, le problème de commande optimale stochastique a été reformulé en une tâche de régression supervisée, où les solutions des équations d'HJB (seuil critique optimal  $Z^*(t)$  et coût optimal associé) servent de données d'entraînement. Plusieurs algorithmes de Machine Learning

ont été comparés pour prédire ces paramètres à partir des caractéristiques de la demande, puis à partir d'un nouvel ensemble de données d'entrée incluant le coût de rupture de stock et le taux de réparation. Les résultats ont montré que l'algorithme des  $k$  plus proches voisins offre un compromis très satisfaisant entre précision et simplicité. Des analyses de scénarios et de sensibilité ont confirmé que le modèle kNN permet d'approximer fidèlement les politiques optimales sur un large ensemble de configurations, tout en réduisant fortement le temps de calcul nécessaire à l'exploration multi-paramétrique.

Le chapitre 3 a étendu cette démarche à un système constitué d'une machine produisant deux types de pièces, soumis à des demandes périodiques synchronisées. Dans ce contexte multi-produits, la dimension de l'espace d'entrée et le nombre de paramètres à prédire augmentent significativement, ce qui modifie les exigences imposées aux modèles d'apprentissage. Une première phase a évalué les performances du kNN dans ce nouveau contexte, en montrant que cet algorithme reste utilisable mais que sa précision se dégrade lorsque le nombre de variables d'entrée augmente. Une seconde phase a alors introduit des réseaux de neurones profonds, mieux adaptés à la modélisation de relations non linéaires complexes en dimension plus élevée. Les résultats ont mis en évidence la capacité des DNN à maintenir une précision élevée sur la prédiction des paramètres optimaux, tout en conservant un gain computationnel décisif par rapport aux résolutions numériques répétées. Ce chapitre a également précisé le domaine de validité du modèle d'apprentissage, discuté les hypothèses de modélisation et mis en lien le compromis de calcul avec les contributions globales du mémoire.

Au regard de ces développements, les objectifs formulés en introduction ont été atteints. D'une part, une approche hybride combinant commande optimale stochastique et apprentissage supervisé a été proposée pour prédire les paramètres optimaux des politiques de production sous demande périodique. D'autre part, plusieurs algorithmes de Machine Learning ont été comparés et évalués dans des contextes mono- et multi-produits, ce qui a permis d'analyser l'impact de la complexité du système sur le choix des algorithmes d'apprentissage. Enfin, ces travaux permettent de dégager un lien clair entre la dimensionnalité du problème et le choix de l'algorithme d'apprentissage : des méthodes simples et interprétables comme le kNN

apparaissent bien adaptées lorsque la dimension des données d'entrée reste modérée, tandis que les réseaux de neurones profonds deviennent nécessaires lorsque le nombre de variables augmente.

Ce mémoire montre que l'apprentissage automatique peut servir à approximer efficacement les paramètres d'une politique optimale de production dans différentes configurations de systèmes, en s'appuyant sur des solutions des équations d'HJB préalablement. Une fois entraînés, les modèles de Machine Learning fournissent des prédictions quasi instantanées, ce qui permet de tester rapidement de multiples scénarios et d'analyser la sensibilité des politiques de production lorsque la demande varie périodiquement, sans avoir à résoudre à nouveau les équations d'HJB à chaque changement de paramètres.



## PERSPECTIVES

Les travaux présentés dans ce mémoire ouvrent plusieurs pistes de recherche, en continuité avec le cadre développé et dans le respect de son domaine de validité. Les principales perspectives peuvent être résumées comme suit :

- La demande a été modélisée sous une forme périodique sinusoïdale, ce qui a facilité la résolution des équations de Hamilton–Jacobi–Bellman et la construction de la base d’apprentissage. Une extension serait de considérer des profils de demande périodiques non sinusoïdaux ou issus de données historiques.
- Dans le cadre de ce mémoire, seule la production est contrôlée. Une perspective consiste à considérer d’autres aspects décisionnels, tels que la maintenance (préventive ou corrective), la qualité et même la périssabilité des produits, en plus du contrôle de la production.
- La considération de la logistique inverse dans une optique d’économie circulaire, en tenant compte des produits retournés. Cela conduirait à considérer conjointement une machine de fabrication et une machine de refabrication dédiée au traitement de ces retours.
- Enfin, les modèles développés opèrent principalement en régime d’interpolation, à l’intérieur de l’enveloppe paramétrique couverte par la base d’apprentissage. L’extension progressive de cette base, de manière contrôlée, afin de couvrir une gamme plus large de paramètres (demande, coûts, fiabilité), permettrait d’élargir le domaine de validité et d’améliorer les capacités de prédiction des modèles.



## ANNEXE I

### VALIDATION ÉTENDUE DU MODÈLE KNN POUR LE SYSTÈME MONO-PRODUIT (CHAPITRE 2)

Le tableau suivant présente les résultats complets de la validation du modèle kNN sur 50 scénarios représentatifs couvrant un large éventail de configurations opérationnelles pour le système mono-produit (Chapitre 2, Phase 2). Pour chaque scénario, les paramètres d'entrée ( $d_m, d_a, c^-, q_{21}$ ), la MAE sur  $z^*$ , les coûts optimaux (HJB et prédit par kNN) ainsi que l'erreur relative sont reportés. Cette validation étendue confirme la robustesse et la précision du modèle kNN pour approximer les paramètres optimaux du système de production mono-produit en dimension modérée.

Tableau-A I-1 Performances du modèle kNN sur 50 scénarios représentatifs du système mono-produit : MAE sur  $z^*$ , coûts optimaux HJB et prédits, ainsi que l'erreur relative

$d_m$	$d_a$	$c^-$	$q_{21}$	MAE $z^*$	Coût HJB	Coût kNN	ERM (%)
0.94	0.06	21	0.22	0.00	542.01	542.01	0.00
0.91	0.06	60	0.22	0.40	903.22	933.51	3.35
0.94	0.07	91	0.29	0.32	654.94	645.79	1.40
0.91	0.10	40	0.22	0.44	777.38	817.51	5.16
0.78	0.06	91	0.20	0.00	1182.96	1182.96	0.00
0.91	0.10	80	0.28	0.26	641.49	607.76	5.26
0.87	0.06	60	0.18	0.33	1272.65	1256.09	1.30
0.87	0.10	40	0.22	0.12	735.71	744.83	1.24
0.86	0.07	31	0.22	0.18	623.49	614.12	1.50
0.94	0.06	91	0.20	0.00	1395.82	1395.82	0.00
0.82	0.08	40	0.20	0.31	751.58	746.42	0.69
0.79	0.07	40	0.20	0.19	766.43	782.80	2.14
0.85	0.07	80	0.26	0.17	676.98	668.02	1.32
0.95	0.07	40	0.29	0.00	430.36	430.36	0.00

$d_m$	$d_a$	$c^-$	$q_{21}$	MAE $z^*$	Coût HJB	Coût kNN	ERM (%)
0.82	0.07	80	0.18	0.18	1132.82	1115.78	1.50
0.95	0.08	60	0.23	0.34	556.03	547.68	1.50
0.95	0.06	60	0.18	0.00	905.43	905.43	0.00
0.95	0.09	40	0.26	0.21	431.67	435.00	0.77
0.88	0.07	91	0.27	0.00	752.00	752.00	0.00
0.87	0.06	31	0.29	0.21	444.11	444.62	0.11
0.78	0.06	60	0.22	0.20	847.69	838.81	1.05
0.86	0.10	80	0.19	0.40	1171.65	1172.99	0.11
0.82	0.10	21	0.29	0.19	330.15	347.18	5.16
0.94	0.10	60	0.20	0.48	992.67	991.67	0.10
0.91	0.09	60	0.24	0.32	620.95	616.22	0.76
0.94	0.09	21	0.29	0.00	310.46	310.46	0.00
0.79	0.08	21	0.26	0.36	328.94	327.33	0.49
0.85	0.06	40	0.23	0.29	521.99	526.11	0.79
0.82	0.06	31	0.18	0.20	773.09	778.69	0.72
0.78	0.10	40	0.18	0.36	1010.94	1021.21	1.01
0.82	0.09	60	0.27	0.10	562.67	566.60	0.70
0.86	0.08	60	0.29	0.26	498.66	498.37	0.06
0.88	0.09	21	0.25	0.00	321.48	321.48	0.00
0.91	0.08	31	0.24	0.19	378.86	387.31	2.23
0.79	0.09	31	0.18	0.25	733.71	730.04	0.50
0.85	0.08	21	0.28	0.00	283.89	283.89	0.00
0.79	0.06	80	0.29	0.18	679.60	665.74	2.04
0.95	0.06	80	0.28	0.00	593.59	593.59	0.00
0.79	0.09	60	0.23	0.33	666.83	656.66	1.53
0.88	0.08	80	0.20	0.27	1043.64	1027.39	1.56
0.88	0.06	80	0.26	0.18	735.48	726.77	1.18
0.85	0.09	31	0.27	0.00	334.88	334.88	0.00

$d_m$	$d_a$	$c^-$	$q_{21}$	MAE $z^*$	Coût HJB	Coût kNN	ERM (%)
0.95	0.10	80	0.27	0.41	649.32	647.27	0.32
0.78	0.08	80	0.24	0.12	792.99	783.94	1.14
0.82	0.08	31	0.19	0.12	595.77	599.50	0.63
0.91	0.07	21	0.18	0.26	640.21	647.32	1.11
0.85	0.10	60	0.25	0.40	652.38	643.90	1.30
0.79	0.10	91	0.22	0.77	1032.63	1035.78	0.30
0.88	0.10	31	0.19	0.10	692.36	707.59	2.20
0.94	0.08	80	0.25	0.18	609.58	604.57	0.82



## ANNEXE II

### VALIDATION ÉTENDUE DU MODÈLE KNN POUR LE SYSTÈME BI-PRODUITS — PHASE 1 (CHAPITRE 3)

Le tableau suivant présente les résultats complets de la validation du modèle kNN sur 50 scénarios représentatifs couvrant un large éventail de configurations opérationnelles pour le système bi-produits (Chapitre 3, Phase 1). Pour chaque scénario, les paramètres d'entrée ( $d_{m1}$ ,  $d_{a1}$ ,  $d_{m2}$ ,  $d_{a2}$ ), les MAE sur  $z_1^*$  et  $z_2^*$ , les coûts optimaux (HJB et prédit par kNN) ainsi que l'erreur relative sont reportés. Cette validation étendue confirme la robustesse et la précision du modèle kNN pour approximer les paramètres optimaux du système multi-produits en dimension modérée.

Tableau-A II-1 Performances du modèle kNN sur 50 scénarios représentatifs du système bi-produits (Phase 1) : MAE sur  $z_1^*$  et  $z_2^*$ , coûts optimaux HJB et prédits, erreur relative

$dm_1$	$da_1$	$dm_2$	$da_2$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	Coût kNN	ERM (%)
0.6475	0.0492	0.6029	0.0527	0.136	0.344	1718.88	1718.35	0.03
0.7084	0.0497	0.7278	0.0618	0.264	0.163	1833.63	1834.29	0.04
0.4975	0.0562	0.8140	0.0369	0.102	0.000	1376.05	1375.35	0.05
0.6563	0.0494	0.4611	0.0568	0.093	0.139	1578.19	1579.03	0.05
0.8397	0.0741	0.4510	0.0393	0.058	0.107	1790.27	1791.39	0.06
0.7635	0.0529	0.4285	0.0537	0.343	0.440	1807.79	1809.07	0.07
0.7027	0.0320	0.7571	0.0590	0.133	0.000	1784.71	1786.04	0.07
0.9121	0.0357	0.4611	0.0452	0.000	0.613	1678.47	1679.96	0.09
0.9327	0.0360	0.6203	0.0684	0.000	0.386	1858.98	1857.26	0.09
0.8016	0.0682	0.4072	0.0332	0.259	0.680	1801.78	1800.10	0.09

$dm_1$	$da_1$	$dm_2$	$da_2$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	Coût kNN	ERM (%)
0.7098	0.0714	0.4687	0.0670	0.188	0.331	1720.25	1721.97	0.10
0.8716	0.0447	0.7977	0.0439	0.000	0.000	1858.18	1855.91	0.12
0.7108	0.0270	0.4769	0.0560	0.133	0.277	1704.34	1706.60	0.13
0.5418	0.0479	0.8193	0.0547	0.264	0.000	1481.11	1478.83	0.15
0.6847	0.0253	0.4955	0.0449	0.218	0.688	1663.08	1665.75	0.16
0.6160	0.0639	0.6563	0.0286	0.180	0.332	1729.95	1727.13	0.16
0.9000	0.0491	0.8242	0.0280	0.000	0.000	1790.08	1787.09	0.17
0.7674	0.0377	0.7863	0.0522	0.280	0.000	1939.93	1936.46	0.18
0.4826	0.0354	0.4239	0.0564	0.039	0.227	1210.96	1213.18	0.18
0.6107	0.0325	0.4092	0.0675	0.142	0.289	1433.22	1430.50	0.19
0.9175	0.0533	0.6096	0.0572	0.000	0.412	1864.21	1867.77	0.19
0.5399	0.0733	0.3771	0.0495	0.069	0.087	1283.41	1285.99	0.20
0.4205	0.0744	0.6215	0.0597	0.195	0.310	1368.63	1371.42	0.20
0.9059	0.0717	0.7470	0.0476	0.000	0.000	1796.79	1792.97	0.21
0.5742	0.0431	0.8159	0.0543	0.164	0.000	1536.42	1533.10	0.22
0.8519	0.0361	0.5562	0.0613	0.289	0.569	1940.40	1936.02	0.23
0.4605	0.0719	0.6233	0.0662	0.185	0.217	1426.00	1429.36	0.24
0.7153	0.0514	0.7570	0.0458	0.162	0.000	1823.73	1828.06	0.24
0.4503	0.0672	0.4825	0.0657	0.128	0.298	1230.94	1233.92	0.24

<b>dm<sub>1</sub></b>	<b>da<sub>1</sub></b>	<b>dm<sub>2</sub></b>	<b>da<sub>2</sub></b>	<b>MAE <math>z_1^*</math></b>	<b>MAE <math>z_2^*</math></b>	<b>Coût HJB</b>	<b>Coût kNN</b>	<b>ERM (%)</b>
0.9393	0.0318	0.6866	0.0750	0.000	0.183	1842.03	1846.55	0.25
0.8412	0.0480	0.6021	0.0478	0.151	0.565	1995.23	1974.18	1.06
0.9788	0.0456	0.6542	0.0470	0.000	0.231	1913.52	1893.20	1.06
0.5446	0.0413	0.5642	0.0260	0.247	0.125	1463.89	1479.46	1.06
0.9424	0.0314	0.7350	0.0640	0.000	0.012	1768.85	1787.79	1.07
0.6909	0.0266	0.3928	0.0445	0.165	0.075	1584.76	1567.67	1.08
0.5009	0.0533	0.7414	0.0495	0.203	0.000	1404.85	1420.00	1.08
0.8743	0.0590	0.3797	0.0352	0.000	0.085	1675.84	1657.60	1.09
0.5018	0.0477	0.5542	0.0609	0.393	0.502	1391.47	1376.08	1.11
0.9870	0.0614	0.7848	0.0481	0.000	0.000	1775.55	1755.64	1.12
0.8217	0.0591	0.4291	0.0338	0.330	0.123	1821.92	1801.46	1.12
0.9501	0.0563	0.5582	0.0385	0.000	0.152	1773.94	1794.06	1.13
0.8921	0.0631	0.6117	0.0332	0.000	0.374	1896.98	1875.36	1.14
0.6109	0.0505	0.6623	0.0358	0.216	1.325	1708.60	1689.02	1.15
0.9507	0.0581	0.6792	0.0656	0.000	0.231	1855.40	1876.75	1.15
0.7129	0.0447	0.4511	0.0506	0.257	0.246	1693.91	1713.57	1.16
0.7195	0.0594	0.5973	0.0636	0.612	1.140	1879.84	1936.43	3.01
0.5393	0.0258	0.7295	0.0487	0.173	0.032	1479.97	1435.21	3.02
0.8196	0.0388	0.5373	0.0424	0.388	0.294	1966.27	1906.57	3.04

$dm_1$	$da_1$	$dm_2$	$da_2$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	Coût kNN	ERM (%)
0.8357	0.0585	0.4389	0.0724	0.729	0.377	1820.69	1764.71	3.07
0.5913	0.0272	0.4782	0.0552	0.588	0.098	1454.09	1498.93	3.08

### ANNEXE III

#### VALIDATION ÉTENDUE DU MODÈLE DNN POUR LE SYSTÈME BI-PRODUITS — PHASE 2 (CHAPITRE 3)

Le tableau suivant présente les résultats complets de la validation du modèle DNN sur 50 scénarios représentatifs couvrant un large éventail de configurations opérationnelles pour le système bi-produits (Chapitre 3, Phase 2). Pour chaque scénario, les paramètres d'entrée ( $d_{m1}$ ,  $d_{a1}$ ,  $d_{m2}$ ,  $d_{a2}$ ,  $c^-$ ,  $q_{21}$ ), les MAE sur  $z_1^*$  et  $z_2^*$ , le coût optimal HJB ainsi que l'erreur relative sont reportés. Cette validation étendue confirme la robustesse et la précision du modèle DNN pour approximer les paramètres optimaux du système multi-produits en contexte de dimensionnalité accrue.

Tableau-A III-1 Performances du modèle DNN sur 50 scénarios représentatifs du système bi-produits (Phase 2) : MAE sur  $z_1^*$  et  $z_2^*$ , coût HJB, erreur relative

$dm_1$	$da_1$	$dm_2$	$da_2$	$c^-$	$q_{21}$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	ERM (%)
0.752	0.066	0.564	0.066	34	0.225	0.451	0.221	1478.05	0.10
0.565	0.043	0.433	0.059	33	0.180	0.172	0.232	1350.48	3.04
0.741	0.042	0.655	0.061	16	0.205	0.256	0.438	1338.55	1.12
0.684	0.070	0.792	0.034	49	0.255	0.234	0.245	1515.00	5.01
0.732	0.057	0.761	0.035	49	0.275	0.231	0.206	1480.42	0.12
0.608	0.053	0.548	0.058	39	0.185	0.146	0.184	1582.17	5.07
0.731	0.050	0.515	0.028	39	0.215	0.176	0.304	1511.29	1.11
0.675	0.033	0.749	0.059	16	0.235	0.176	0.300	1161.52	1.11
0.705	0.062	0.657	0.065	20	0.230	0.210	0.304	1226.10	1.15
0.747	0.050	0.497	0.056	44	0.225	0.277	0.088	1511.78	0.14
0.861	0.068	0.439	0.053	40	0.260	0.316	0.289	1431.72	0.08
0.556	0.033	0.523	0.047	21	0.235	0.266	0.109	903.02	0.01
0.670	0.057	0.422	0.069	39	0.285	0.073	0.212	966.33	3.03
0.779	0.066	0.609	0.054	41	0.235	0.459	0.349	1580.19	0.07
0.874	0.033	0.534	0.063	13	0.225	0.294	0.239	1208.94	0.01
0.551	0.033	0.648	0.026	23	0.215	0.179	0.340	1150.86	0.02
0.557	0.074	0.493	0.072	40	0.220	0.151	0.190	1192.58	3.04
0.609	0.070	0.653	0.073	40	0.300	0.168	0.279	1012.49	5.04

$dm_1$	$da_1$	$dm_2$	$da_2$	$c^-$	$q_{21}$	MAE $z_1^*$	MAE $z_2^*$	Coût HJB	ERM (%)
0.530	0.071	0.701	0.043	23	0.195	0.267	0.594	1353.24	5.05
0.616	0.066	0.789	0.027	11	0.195	0.653	0.651	1242.34	0.10
0.654	0.068	0.760	0.057	21	0.245	0.209	0.352	1217.87	0.07
0.540	0.056	0.560	0.054	33	0.170	0.419	0.456	1444.21	0.12
0.769	0.045	0.525	0.058	28	0.230	0.113	0.084	1342.29	1.11
0.506	0.029	0.661	0.050	25	0.260	0.241	0.143	921.55	3.01
0.547	0.074	0.681	0.028	21	0.215	0.177	0.197	1161.50	1.15
0.754	0.027	0.533	0.046	19	0.225	0.251	0.059	1178.50	0.01
0.713	0.026	0.428	0.070	31	0.185	0.415	0.321	1555.43	5.01
0.552	0.070	0.663	0.027	45	0.180	0.292	0.252	1449.37	1.12
0.667	0.062	0.734	0.062	44	0.295	0.099	0.204	1214.65	3.01
0.889	0.058	0.402	0.072	22	0.150	0.474	0.720	1595.37	0.02
0.582	0.041	0.637	0.059	19	0.245	0.172	0.265	951.83	1.12
0.581	0.034	0.715	0.060	49	0.255	0.217	0.288	1322.44	0.07
0.509	0.063	0.684	0.028	30	0.280	0.155	0.228	924.37	3.02
0.770	0.040	0.531	0.053	25	0.200	0.127	0.089	1524.60	0.01
0.640	0.058	0.608	0.048	35	0.260	0.104	0.254	1131.73	1.07
0.944	0.048	0.518	0.061	31	0.265	0.371	0.149	1515.34	0.01
0.705	0.028	0.733	0.036	33	0.280	0.245	0.260	1214.76	5.00
0.671	0.061	0.720	0.059	16	0.195	0.125	0.389	1404.42	3.01
0.656	0.041	0.489	0.071	13	0.165	0.237	0.430	1260.65	1.08
0.937	0.026	0.410	0.066	49	0.225	0.600	0.439	1557.88	3.04
0.778	0.062	0.470	0.046	33	0.150	0.745	1.039	1577.68	0.07
0.651	0.066	0.451	0.043	25	0.230	0.113	0.112	1060.76	1.07
0.519	0.035	0.566	0.044	33	0.250	0.249	0.255	973.10	0.13
0.586	0.039	0.782	0.026	29	0.285	0.078	0.314	1073.76	5.07
0.807	0.067	0.498	0.028	23	0.195	0.407	0.252	1585.43	1.03
0.712	0.060	0.614	0.073	43	0.260	0.195	0.516	1322.88	0.03
0.622	0.041	0.659	0.038	16	0.255	0.219	0.154	903.20	3.06
0.577	0.072	0.709	0.035	19	0.225	0.137	0.172	1131.44	0.10
0.624	0.067	0.542	0.037	22	0.240	0.108	0.198	994.64	1.12
0.503	0.042	0.659	0.038	11	0.155	0.529	0.255	1240.78	3.04

## BIBLIOGRAPHIE

- Abdelilah, B., El Korchi, A. & Balambo, M. A. (2018). Flexibility and agility : evolution and relationship. *Journal of Manufacturing Technology Management*. doi : 10.1108/JMTM-03-2018-0090.
- Adul-Kader, W. (2001). *Modélisation du déploiement du stock-tampon dans une ligne de production sujette à des défaillances*. (Mémoire de maîtrise, Université Laval). Repéré à Dépôt institutionnel.
- Akella, M. et al. (1986). Optimal Control of Production Systems. *Some Journal*, 1(1), 1-10.
- Arlot, S. & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79.
- Assid, M., Gharbi, A. & Hajji, A. (2015). Joint Production, Setup and Preventive Maintenance Policies of Unreliable Two-Product Manufacturing Systems. 53(15), 4668–4683. doi : 10.1080/00207543.2015.1030468.
- Behnamfar, R., Sajadi, S. M. & Tootoonchy, M. Developing Environmental Hedging Point Policy with Variable Demand : A Machine Learning Approach. 254, 108640. doi : 10.1016/j.ijpe.2022.108640.
- Behnamfar, R., Sajadi, S. M. & Tootoonchy, M. (2022a). Developing Environmental Hedging Point Policy with Variable Demand : A Machine Learning Approach. *International Journal of Production Economics*, 254, 108640. doi : 10.1016/j.ijpe.2022.108640.
- Behnamfar, R., Sajadi, S. M. & Tootoonchy, M. (2022b). Developing Environmental Hedging Point Policy with Variable Demand : A Machine Learning Approach. 254, 108640. doi : 10.1016/j.ijpe.2022.108640.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- Bensaci, R., Khaldi, B., Aiadi, O. & Benchabana, A. Deep Convolutional Neural Network with KNN Regression for Automatic Image Annotation. 11(21), 10176. doi : 10.3390/app112110176.
- Bensaci, R., Khaldi, B., Aiadi, O. & Benchabana, A. (2021). Deep Convolutional Neural Network with KNN Regression for Automatic Image Annotation. *Applied Sciences*, 11(21), 10176. doi : 10.3390/app112110176.
- Bensoussan, A. (1982). *Stochastic Control of Partially Observable Systems*. Cambridge University Press.

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York : Springer.
- Black, D. C. & Donovan, J. (2006). *SystemC : From the Ground up ; [Includes New Features of SystemC Version 2.1]* (éd. 1. softcover print). Springer US.
- Boukas & Eurie. (1998). Hedging Point Policy Improvement. *Journal of Optimization Theory and Applications*, 97, 47–70. doi : 10.1023/A:1022670932536.
- Bouslah, B., Gharbi, A., Pellerin, R. & Hajji, A. (2013). Optimal Production Control Policy in Unreliable Batch Processing Manufacturing Systems with Transportation Delay. *International Journal of Production Research*, 51(1), 264–280. doi : 10.1080/00207543.2012.676217.
- Broomhead, D. S. & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2(3), 321–355.
- Büttner, K., Antons, O. & Arlinghaus, J. C. (2022). Applied Machine Learning for Production Planning and Control : Overview and Potentials. *IFAC-PapersOnLine*, 55(10), 2629–2634. doi : 10.1016/j.ifacol.2022.10.106.
- Cavalcanti, J. H., Kovacs, T., Ko, A. & Pocsarovszky, K. Production System Efficiency Optimization through Application of a Hybrid Artificial Intelligence Solution. 37(6), 790–807. doi : 10.1080/0951192X.2023.2257661.
- Covei, D.-P. Stochastic Production Planning in Manufacturing Systems. 14(10), 766. doi : 10.3390/axioms14100766.
- Cover, T. M. & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.
- Cruz, Y. J., Villalonga, A., Castaño, F., Rivas, M. & Haber, R. E. Automated Machine Learning Methodology for Optimizing Production Processes in Small and Medium-Sized Enterprises. 12, 100308. doi : 10.1016/j.orp.2024.100308.
- Darendeliler, A., Claeys, D. & Aghezzaf, E.-H. (2023). Joint Multi-Item Production and Condition-Based Maintenance Control of a System with Setup Times and Stochastic Demand :. *Proceedings of the 12th International Conference on Operations Research and Enterprise Systems*, pp. 185–192. doi : 10.5220/0011741300003396.
- David, M. & Firdaus, M. A. Defect Rate Prediction in Manufacturing Process Using K-Nearest Neighbor Algorithm. 3(2), 161–173. doi : 10.52362/ijjiems.v3i2.1599.

- David, M. & Firdaus, M. A. (2024). Defect Rate Prediction in Manufacturing Process Using K-Nearest Neighbor Algorithm. *International Journal of Informatics, Economics, Management and Science*, 3(2), 161. doi : 10.52362/ijiems.v3i2.1599.
- Delpla, V., Chapron, K., Kenné, J.-P. & Hof, L. A. (2024). A Novel Approach for Predicting Lockout/Tagout Safety Procedures for Smart Maintenance Strategies. *International Journal of Production Research*, 62(13), 4754–4775. doi : 10.1080/00207543.2023.2275635.
- Fildes, R., Ma, S. & Kolassa, S. (2022). Retail forecasting : Research and practice. *International Journal of Forecasting*, 38(4), 1283–1318. doi : 10.1016/j.ijforecast.2019.06.004.
- Gharbi, Kenné, J.-P. & Hajji, A. (2006). Operational Level-Based Policies in Production Rate Control of Unreliable Manufacturing Systems with Set-Ups. *International Journal of Production Research*, 44(3), 545–567. doi : 10.1080/00207540500270364.
- Gharbi, Kenné, J.-P., Takengny, A. L. K. & Assid, M. (2024). Joint Emission-Dependent Optimal Production and Preventive Maintenance Policies of a Deteriorating Manufacturing System. *Sustainability*, 16(14), 6146. doi : 10.3390/su16146146.
- Gharbi, A. & Jean-Pierre, K. (2000). Production and Preventive Maintenance Rates Control for a Manufacturing System : An Experimental Design Approach. *International Journal of Production Economics*, 65, 275–287. doi : 10.1016/S0925-5273(99)00079-1.
- Gijsbrechts, J., Boute, R. N., Van Mieghem, J. A. & Zhang, D. J. (2022a). Can Deep Reinforcement Learning Improve Inventory Management? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems. *Manufacturing & Service Operations Management*, 24(3), 1349–1368. doi : 10.1287/msom.2021.1064.
- Gijsbrechts, J., Boute, R. N., Van Mieghem, J. A. & Zhang, D. J. (2022b). Can Deep Reinforcement Learning Improve Inventory Management? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems. 24(3), 1349–1368. doi : 10.1287/msom.2021.1064.
- Goodfellow & Yoshua Bengio. (2016). *Deeplearningbook.Org/Contents/ML.Html*. Repéré le 2025-11-20 à <https://www.deeplearningbook.org/contents/ml.html>.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. Repéré à <https://www.deeplearningbook.org/>.
- Graves, S. & Willems, S. (2008). Strategic Inventory Placement in Supply Chains : Nonstationary Demand. 10, 278–287. doi : 10.1287/msom.1070.0175.

- Hajji, A., Gharbi, A. & Jean-Pierre, K. (2003). *Production and Setup Planning of a Failure Prone Multiple Parts Manufacturing System*.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction* (éd. 2). New York : Springer.
- Ho, W. & Yu, F. (2021). Chiller System Optimization Using k Nearest Neighbour Regression. *Journal of Cleaner Production*, 303, 127050. doi : 10.1016/j.jclepro.2021.127050.
- Hosseinzadeh, A., Frank Chen, F., Shahin, M. & Bouzary, H. A Predictive Maintenance Approach in Manufacturing Systems via AI-based Early Failure Detection. 35, 1179–1186. doi : 10.1016/j.mfglet.2023.08.125.
- Huang, S. & Hong-Chao Zhang. (1994). Artificial Neural Networks in Manufacturing : Concepts, Applications, and Perspectives. *IEEE Transactions on Components, Packaging, and Manufacturing Technology : Part A*, 17(2), 212–228. doi : 10.1109/95.296402.
- Hui, G., Chen, Z., Wang, Y., Zhang, D. & Gu, F. (2023a). An Integrated Machine Learning-Based Approach to Identifying Controlling Factors of Unconventional Shale Productivity. *Energy*, 266, 126512. doi : 10.1016/j.energy.2022.126512.
- Hui, G., Chen, Z., Wang, Y., Zhang, D. & Gu, F. (2023b). An Integrated Machine Learning-Based Approach to Identifying Controlling Factors of Unconventional Shale Productivity. *Energy*, 266, 126512. doi : 10.1016/j.energy.2022.126512.
- Höse, K., Amaral, A., Götze, U. & Peças, P. Manufacturing Flexibility through Industry 4.0 Technological Concepts—Impact and Assessment. 24(2), 271–289. doi : 10.1007/s40171-023-00339-y.
- Ian Goodfellow, Yoshua Bengio & Aaron Courville. (2016). *DeepLearningBook.Org/Contents/Mlp.Html*. Repéré le 2025-04-15 à <https://www.deeplearningbook.org/contents/mlp.html>.
- Infotech, I. T. C. Product Quality Improvement in Manufacturing Using Machine Learning and Stochastic Optimization. Repéré le 2025-11-13 à <https://www.itcinfotech.com/blogs/product-quality-improvement-in-manufacturing-using-machine-learning-and-stochastic-optimization/>.
- Jean-Pierre, K. & Boukas, E. (2003). Hierarchical Control of Production and Maintenance Rates in Manufacturing Systems. *Journal of Quality in Maintenance Engineering*, 9, 66–82. doi : 10.1108/13552510310466927.

- Kaddachi, Ali Gharbi & Jean-Pierre Kenné. (2021). Integrated Production and Maintenance Control Policies for Failure-prone Manufacturing Systems Producing Perishable Products. *The International Journal of Advanced Manufacturing Technology*.
- Kairo, J. Machine Learning Algorithms for Predictive Maintenance in Manufacturing. 6(4), 66–79. doi : 10.47941/jts.2144.
- Kenne, J. P., Boukas, E. K. & Gharbi, A. (2003). Control of Production and Corrective Maintenance Rates in a Multiple-Machine, Multiple-Product Manufacturing System. 38(3), 351–365. doi : 10.1016/S0895-7177(03)90093-8.
- Kenné, J.-P., Gharbi, A., Takengny, A. L. K. & Assid, M. (2024). Optimal Control Policy of Unreliable Production Systems Generating Greenhouse Gas Emission. *Sustainability*, 16(13), 5760. doi : 10.3390/su16135760.
- Kouedeu, A. F., Kenné, J.-P., Dejax, P., Songmene, V. & Polotski, V. (2014). Production Planning of a Failure-Prone Manufacturing/Remanufacturing System with Production-Dependent Failure Rates. *Applied Mathematics*, 05(10), 1557–1572. doi : 10.4236/am.2014.510149.
- Kushner, H. J. & Dupuis, P. (1992). *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep Learning. *Nature*, 521, 436–444.
- Lei, L., Pang, R., Han, Z., Wu, D., Xie, B. & Su, Y. (2023). Current Applications and Future Impact of Machine Learning in Emerging Contaminants : A Review. *Critical Reviews in Environmental Science and Technology*, 53(20), 1817–1835. doi : 10.1080/10643389.2023.2190313.
- Liu, J. Semiconductor Production Scheduling and Industrial Economic Coordination Based on Deep Learning and Heat Loss Optimization. 54, 102831. doi : 10.1016/j.tsep.2024.102831.
- Luo, X. & Chu, Y. (2023). Optimal Dynamic Pricing and Production Policy for a Stochastic Inventory System with Perishable Products and Inventory-Level-Dependent Demand. *Journal of Industrial and Management Optimization*, 19(12), 8541–8557. doi : 10.3934/jimo.2023050.
- Mazzei, D. & Ramjattan, R. Machine Learning for Industry 4.0 : A Systematic Review Using Deep Learning-Based Topic Modelling. 22(22), 8641. doi : 10.3390/s22228641.

- Megoze, P., Kibouka, G.-R., Kenné, J.-P. & Hof, L. A. (2022). Production, Maintenance and Quality Inspection Planning of a Hybrid Manufacturing/Remanufacturing System under Production Rate-Dependent Deterioration. *The International Journal of Advanced Manufacturing Technology*, 121(1-2), 1289–1314. doi : 10.1007/s00170-022-09078-3.
- Mifdal, Hajej, Z. & Dellagi, S. (2014a). Production/Maintenance Control of Multiple-Product Manufacturing System. *2014 Second World Conference on Complex Systems (WCCS)*, pp. 304–310. doi : 10.1109/ICoCS.2014.7060904.
- Mifdal, L., Hajej, Z., Dellagi, S. & Rezg, N. (2013). An Optimal Production Planning and Maintenance Policy for a Multiple-Product and Single Machine under Failure Rate Dependency. 46(9), 507–512. doi : 10.3182/20130619-3-RU-3018.00246.
- Mifdal, L., Hajej, Z. & Dellagi, S. (2014b). Joint Optimization Approach of Maintenance Planning and Production Scheduling for a Multiple-Product Manufacturing System. 47(3), 8042–8047. doi : 10.3182/20140824-6-ZA-1003.02161.
- Mitchell, T. M. (1997). *Machine Learning*. New York : McGraw-Hill.
- Murphy, K. P. *Probabilistic Machine Learning : An Introduction* (éd. Third printing). The MIT Press.
- Murphy, K. P. (2012). *Machine Learning : A Probabilistic Perspective*. MIT Press. Repéré à <https://probml.github.io/pml-book/>.
- Mutlu, A. Y. & Yucel, O. (2018). An Artificial Intelligence Based Approach to Predicting Syngas Composition for Downdraft Biomass Gasification. *Energy*, 165, 895–901. doi : 10.1016/j.energy.2018.09.131.
- Njike, A. N., Pellerin, R. & Kenne, J. P. (2012). Simultaneous control of maintenance and production rates of a manufacturing system with defective products. *Journal of Intelligent Manufacturing*, 23(2), 323–332. doi : 10.1007/s10845-009-0354-y.
- Norouzi, A., Shahpouri, S., Gordon, D., Winkler, A., Nuss, E., Abel, D., Andert, J., Shahbakhti, M. & Koch, C. R. (2022a). Machine Learning Integrated with Model Predictive Control for Imitative Optimal Control of Compression Ignition Engines. *IFAC-PapersOnLine*, 55(24), 19–26. doi : 10.1016/j.ifacol.2022.10.256.
- Norouzi, A., Shahpouri, S., Gordon, D., Winkler, A., Nuss, E., Abel, D., Andert, J., Shahbakhti, M. & Koch, C. R. (2022b). Machine Learning Integrated with Model Predictive Control for Imitative Optimal Control of Compression Ignition Engines. *IFAC-PapersOnLine*, 55(24), 19–26. doi : 10.1016/j.ifacol.2022.10.256.

- Omurlu, Fulden Cantas & Mevlut Ture. (2020). An Empirical Study on Performances of Multilayer Perceptron, Logistic Regression, ANFIS, KNN and Bagging CART. Repéré le 2025-11-05 à <https://www.tandfonline.com/doi/epdf/10.1080/09720510.2019.1696924?needAccess=true>.
- Ouaret, S., Gharbi, Ali & Kenne, Jean-Pierre. Hierarchical Control of Production with Stochastic Demand in Manufacturing Systems.
- Ouaret, S., Kenné, J.-P. & Gharbi, A. Production and Replacement Policies for a Deteriorating Manufacturing System under Random Demand and Quality. 264(2), 623–636. doi : 10.1016/j.ejor.2017.06.062.
- Ouaret, S., Polotski, V., Kenné, J.-P. & Gharbi, A. Manufacturing-Remanufacturing Failure-Prone Systems under Random Demands. 46(9), 1620–1625. doi : 10.3182/20130619-3-RU-3018.00338.
- Ouaret, S., Kenné, J.-P. & Gharbi, A. (2011). Hierarchical Control of Production with Stochastic Demand in Manufacturing Systems.
- Ouaret, S., Polotski, V., Kenné, J.-P. & Gharbi, A. (2013). Manufacturing-Remanufacturing Failure-Prone Systems under Random Demands. *IFAC Proceedings Volumes*, 46(9), 1620–1625. doi : 10.3182/20130619-3-RU-3018.00338.
- Ouaret, S., Kenné, J.-P., Gharbi, A. & Polotski, V. (2017). Age-Dependent Production and Replacement Strategies in Failure-Prone Manufacturing Systems. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 231(3), 540–554. doi : 10.1177/0954405415616060.
- Ouaret, S., Kenné, J.-P. & Gharbi, A. (2018). Production and Replacement Policies for a Deteriorating Manufacturing System under Random Demand and Quality. *European Journal of Operational Research*, 264(2), 623–636. doi : 10.1016/j.ejor.2017.06.062.
- Paraschos, P. D., Gasteratos, A. C. & Koulouriotis, D. E. (2024a). Deep Learning Model for Optimizing Control and Planning in Stochastic Manufacturing Environments. *Expert Systems with Applications*, 257, 125075. doi : 10.1016/j.eswa.2024.125075.
- Paraschos, P. D., Gasteratos, A. C. & Koulouriotis, D. E. (2024b). Deep Learning Model for Optimizing Control and Planning in Stochastic Manufacturing Environments. 257, 125075. doi : 10.1016/j.eswa.2024.125075.
- Paull, S., Bubak, A. & Stuckenschmidt, H. Machine Learning for Master Production Scheduling : Combining Probabilistic Forecasting with Stochastic Optimisation. 271, 126586. doi : 10.1016/j.eswa.2025.126586.

- Peng, Y., Lyu, Y., Zhang, J. & Chu, Y. (2025). Heterogeneous Graph Neural-Network-Based Scheduling Optimization for Multi-Product and Variable-Batch Production in Flexible Job Shops. 15, 5648. doi : 10.3390/app15105648.
- Pestov, V. (2011). Is the k-NN classifier in high dimensions affected by the curse of dimensionality? *arXiv preprint arXiv :1110.4347*. Repéré à <https://arxiv.org/abs/1110.4347>.
- Pestov, V. (2012). Is the K-NN Classifier in High Dimensions Affected by the Curse of Dimensionality? Repéré le 2025-11-20 à <http://arxiv.org/abs/1110.4347>.
- Petsagkourakis, P., Sandoval, I. O., Bradford, E., Galvanin, F., Zhang, D. & del Rio-Chanona, E. A. (2020a). Chance Constrained Policy Optimization for Process Control and Optimization. arXiv.
- Petsagkourakis, P., Sandoval, I. O., Bradford, E., Galvanin, F., Zhang, D. & del Rio-Chanona, E. A. (2020b). Chance Constrained Policy Optimization for Process Control and Optimization. Repéré le 2025-11-20 à <http://arxiv.org/abs/2008.00030>.
- Polotski, Kenne, J.-P. & Gharbi, A. (2011). ADAPTIVE CONTROL OF HYBRID FAILURE-PRONE MANUFACTURING SYSTEMS UNDER DEMAND VARIATION AND UNCERTAINTY.
- Polotski, Jean-Pierre, K. & Gharbi, A. (2016). *ADAPTIVE CONTROL OF HYBRID FAILURE-PRONE MANUFACTURING SYSTEMS UNDER DEMAND VARIATION AND UNCERTAINTY*.
- Polotski, Kenne, J.-P. & Gharbi, A. (2019). Optimal Production and Corrective Maintenance in a Failure-Prone Manufacturing System under Variable Demand. *Flexible Services and Manufacturing Journal*, 31(4), 894–925. doi : 10.1007/s10696-019-09337-8.
- Polotski, Gharbi, A. & Kenne, J.-P. (2022). Production Control in Manufacturing Systems with Perishable Products under Periodic Demand. *Journal of Manufacturing Systems*, 63, 288–303. doi : 10.1016/j.jmsy.2022.03.013.
- Polotski, V., Gharbi, A. & Kenne, J.-P. Production Control in Manufacturing Systems with Perishable Products under Periodic Demand. 63, 288–303. doi : 10.1016/j.jmsy.2022.03.013.
- Polotski, V., Kenne, J.-P. & Gharbi, A. ADAPTIVE CONTROL OF HYBRID FAILURE-PRONE MANUFACTURING SYSTEMS UNDER DEMAND VARIATION AND UNCERTAINTY.

- Polotski, V., Kenne, J.-P. & Gharbi, A. Optimal Production and Corrective Maintenance in a Failure-Prone Manufacturing System under Variable Demand. 31(4), 894–925. doi : 10.1007/s10696-019-09337-8.
- Polotski, V., Gharbi, A. & Kenne, J.-P. (2021). Production Control of Unreliable Manufacturing Systems with Perishable Inventory. *The International Journal of Advanced Manufacturing Technology*, 116(7-8), 2473–2496. doi : 10.1007/s00170-021-07503-7.
- Priore, P., Ponte, B., Rosillo, R. & de la Fuente, D. (2019). Applying Machine Learning to the Dynamic Selection of Replenishment Policies in Fast-Changing Supply Chain Environments. *International Journal of Production Research*, 57(11), 3663–3677.
- Pyke, D., Peterson, R. & Silver, E. (2001). Inventory Management and Production Planning and Scheduling (Third Edition). *Journal of The Operational Research Society - J OPER RES SOC*, 52, 845–845.
- Rivera-Gómez, H., Gharbi, A. & Kenné, J. P. (2013). Joint Production and Major Maintenance Planning Policy of a Manufacturing System with Deteriorating Quality. *International Journal of Production Economics*, 146(2), 575–587. doi : 10.1016/j.ijpe.2013.08.006.
- Rivera-Gómez, H., Gharbi, A., Kenné, J.-P., Ortiz-Zarco, R. & Corona-Armenta, J. R. (2021). Joint Production, Inspection and Maintenance Control Policies for Deteriorating System under Quality Constraint. *Journal of Manufacturing Systems*, 60, 585–607. doi : 10.1016/j.jmsy.2021.07.018.
- Samir, H. (2012). MODELE D’EVALUATION DES PERFORMANCES D’UNE LIGNE DE PRODUCTION D’UNE SÉQUENCE DE N PRODUITS DISTINCTS SUR M MACHINES AVEC ( M 1 ) STOCKS TAMPONS.
- Sebayang, A. H., Masjuki, H. H., Ong, H. C., Dharma, S., Silitonga, A. S., Kusumo, F. & Milano, J. (2017). Optimization of Bioethanol Production from Sorghum Grains Using Artificial Neural Networks Integrated with Ant Colony. *Industrial Crops and Products*, 97, 146–155. doi : 10.1016/j.indcrop.2016.11.064.
- Sethi, A. K. & Sethi, S. P. Flexibility in Manufacturing : A Survey. 2(4), 289–328. doi : 10.1007/BF00186471.
- Sethi, S. P. & Thompson, G. L. (2005). *Optimal Control Theory : Applications to Management Science and Economics*. Springer.
- Silver, E. A., Pyke, D. F. & Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling* (éd. 3rd). New York : Wiley.

- Soori, M., Arezoo, B. & Dastres, R. (2023). Artificial Neural Networks in Supply Chain Management, a Review. *Journal of Economy and Technology*, 1, 179–196. doi : 10.1016/j.ject.2023.11.002.
- Subramonian, S., Kadirgama, K., Al-Obaidi, A. S. M., Salleh, M. S. M., Vatesh, U. K., Pujari, S., Rao, D. & Ramasamy, D. (2023). Artificial Neural Network Performance Modeling and Evaluation of Additive Manufacturing 3D Printed Parts. *Engineering, Technology & Applied Science Research*, 13(5), 11677–11684. doi : 10.48084/etasr.6185.
- Tibshirani, S. & Friedman, H. (2017). Valerie and Patrick Hastie. 764.
- Tunc, H., Kilic, O., Tarim, S. A. & Eksioglu, B. (2011). The Cost of Using Stationary Inventory Policies When Demand Is Non-Stationary. 39, 410–415. doi : 10.1016/j.omega.2010.09.005.
- Umer, M. A., Junejo, K. N., Jilani, M. T. & Mathur, A. P. (2022). Machine Learning for Intrusion Detection in Industrial Control Systems : Applications, Challenges, and Recommendations. *International Journal of Critical Infrastructure Protection*, 38, 100516. doi : 10.1016/j.ijcip.2022.100516.
- Vihar Kurama. Regression in Machine Learning : Definition and Examples. Repéré le 2025-11-05 à <https://builtin.com/data-science/regression-machine-learning>.
- Wang, C., Zhang, X., Zhao, G. & Chen, Y. (2023). Mechanisms, Methods and Applications of Machine Learning in Bio-Alcohol Production and Utilization : A Review. *Chemosphere*, 342, 140191. doi : 10.1016/j.chemosphere.2023.140191.
- Wang, X. & Chen, Q. (2024). Stochastic Optimal Control for Dynamic Pricing and Production in Fashion Retailing : An Economic Sustainability Approach. *Discrete Dynamics in Nature and Society*, 2024(1), 4854557. doi : 10.1155/ddns/4854557.
- Wolters, J. & Huchzermeier, A. (2021). Joint In-Season and Out-of-Season Promotion Demand Forecasting in a Retail Environment. *Journal of Retailing*, 97(4), 726–745. doi : 10.1016/j.jretai.2021.01.003.
- Yan, Y., Chen, F. Y., Fu, Z. & Bi, W. (2025). Heuristics and Deep Reinforcement Learning for the Inventory Problem with an All-or-Nothing Yield Pattern and Non-Zero Leadtimes. 186, 107305. doi : 10.1016/j.cor.2025.107305.
- Yang, L. (2016). Optimizing Inventory for Profitability and Order Fulfillment Improvement.
- Yang, L., Li, H., Campbell, J. F. & Sweeney, D. C. (2017). Integrated Multi-Period Dynamic Inventory Classification and Control. 189, 86–96. doi : 10.1016/j.ijpe.2017.04.010.

- Ye, L., Xie, N., Boylan, J. E. & Shang, Z. (2024). Forecasting Seasonal Demand for Retail : A Fourier Time-Varying Grey Model. 40(4), 1467–1485. doi : 10.1016/j.ijforecast.2023.12.006.
- Yu, W., Xia, T., Xu, Y., Si, G., Wang, D., Pan, E. & Xi, L. (2024). Quality-Centered Production and Maintenance Scheduling for Multi-Machine Manufacturing Systems under Variable Operating Conditions. 250, 110264. doi : 10.1016/j.res.2024.110264.
- Zhang, J., Chen, X., An, Y., Zhang, L., Shi, H. & Xu, W. (2025). Joint Optimization of Preventive Maintenance and Product Quality Improvement Policies for Deteriorating Manufacturing Systems with Quality-Reliability Dependency. 255, 110661. doi : 10.1016/j.res.2024.110661.
- Zhu, X. (2009). Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1), 1–130.
- Zipkin, P. (2000). *Foundations of Inventory Management*. McGraw-Hill Companies, Incorporated.