



Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Automatique

Spécialité : Automatique

Présenté par : Maryliz MERZOUK

Thème

**REALISATION ET CONTROLE D'UN SYSTEME
DE VENTILATION**

Soutenu publiquement, le 18/06/2026 , devant le jury composé de :

Nom et Prénom	Grade	Établissement	Rôle
Dr. DIDI Ibtissem	MCA	ESSA Tlemcen	Présidente du jury
Dr. BOUKLI HACENE Lotfi Fazil	MAB	ESSA Tlemcen	Directeur de thèse
Dr. M'HAMDI Mohammed	MCB	ESSA Tlemcen	Co-Directeur
Dr. MOKHTARI Rida	MCA	ESSA Tlemcen	Examineur 1
Dr. ARICHI Faycal	MCA	ESSA Tlemcen	Examineur 2

Année universitaire : 2025 /2026

Dédicace

À mon père,

Celui qui m'a appris la valeur du travail, de la persévérance, de l'honnêteté et qui m'a aimé sans conditions. Merci pour tous les sacrifices que tu as consentis afin de m'offrir les meilleures conditions pour réussir. Ton courage, ta sagesse et ton soutien constant ont toujours été pour moi un exemple à suivre.

À ma mère,

Aucune parole ne saurait exprimer toute ma reconnaissance pour ton amour infini, ta tendresse, tes prières et ton dévouement sans limite. Merci pour chaque effort, chaque encouragement et chaque moment où tu as cru en moi, même dans les périodes les plus difficiles. Ta présence et ton affection ont été une source inestimable de motivation tout au long de mon parcours.

A Samir MESSAOUDI

Qui a été présente dans les moments les plus difficiles de mon parcours, merci pour ton soutien, tes encouragements et ta confiance lorsque j'en avais le plus besoin.

Mme Aya Righi et à M. Ramzi

Je tiens à exprimer ma profonde gratitude à Mme Aya Righi et à M. Ramzi pour leur précieuse aide tout au long de la rédaction et de l'organisation de ce mémoire. Leurs conseils avisés, leur disponibilité et leur soutien m'ont été d'une grande utilité et ont contribué de manière significative à l'aboutissement de ce travail. Qu'ils trouvent ici l'expression de ma sincère reconnaissance.

À vous,

Ce travail est le fruit de vos sacrifices, de votre patience et de votre confiance. Si j'ai pu atteindre cette étape aujourd'hui, c'est avant tout grâce à votre soutien indéfectible.

Que Dieu vous accorde une longue vie, une bonne santé, le bonheur et la sérénité.

Avec tout mon amour, mon respect et ma profonde gratitude,

Je vous dédie ce mémoire.

Remerciements

Je remercie en premier lieu mon Dieu de m'avoir accordé la santé, la volonté et la patience nécessaires pour mener à bien ce travail.

Je tiens à exprimer ma vive reconnaissance à **M. Mohammed Mssirdi**, dont l'enseignement rigoureux et l'investissement pédagogique ont largement contribué à ma formation et ont posé les bases scientifiques de ce travail.

Mes sincères remerciements vont à mon encadrant **M. Boukli Hacene Lotfi Fazil** et à mon co-encadrant **M. M'Hamedi Mohammed**, pour m'avoir suivie et soutenue afin de pouvoir réaliser ce travail. Je leur exprime ma gratitude pour leur disponibilité, leurs précieux conseils et le soutien qu'ils m'ont apporté tout au long de la préparation de ce mémoire.

Je tiens également à exprimer ma sincère gratitude à **M. ADJIM Ramz-Eddine Abderrezak**, ingénieur au FabLab, pour son aide précieuse et sa disponibilité tout au long de la réalisation pratique de ce projet. Son accompagnement technique et ses conseils avisés m'ont été d'un grand soutien dans la mise en œuvre de la plateforme expérimentale.

Mes vifs remerciements vont aux membres du jury, **Dr. Didi Ibtissem**, présidente, ainsi que **Dr. Mokhtari Rida** et **Dr. Arichi Faycal**, examinateurs, pour l'honneur qu'ils me font en acceptant de corriger et d'évaluer ce travail.

Enfin, je remercie chaleureusement tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce mémoire, ainsi que l'ensemble des professeurs ayant contribué à ma formation. Qu'ils trouvent ici l'expression de ma profonde et sincère reconnaissance.

Résumé

Ce mémoire présente la conception et la validation expérimentale d'un système de régulation thermique basé sur des ventilateurs à vitesse variable pilotés par Arduino Uno. Le prototype intègre des capteurs de température DHT11, un capteur MQ-9, un driver L298N, un afficheur LCD et un buzzer. Un régulateur PI a été programmé pour maintenir la température proche de la consigne tout en garantissant stabilité et précision. Les essais expérimentaux ont montré une réponse stable, un dépassement limité et une erreur finale faible. Le système gère également les perturbations et la protection thermique multiniveaux. Cette solution est simple, économique et adaptée aux applications embarquées.

Mots-clés : Régulation thermique, Arduino Uno, Régulateur PI, Ventilation, PWM, DHT11, MQ-9, Système embarqué, Rejet de perturbations, Protection thermique.

الملخص

Arduino هدف هذا العمل إلى تصميم واختبار نظام للتحكم الحراري باستخدام مراوح بسرعة متغيرة تعمل بواسطة لوحة شاشة L298N وحدة قيادة MQ-9 مستشعر دخان، DHT11 النموذج الأولي يتضمن مستشعرات حرارة Uno. للحفاظ على درجة الحرارة قرب القيمة المرجعية مع ضمان الاستقرار والدقة PI ومنبه صوتي. تم برمجة متحكم LCD أظهرت التجارب استجابة مستقرة مع تجاوز محدود وخطأ نهائي منخفض. كما يدعم النظام رفض الاضطرابات وحماية حرارية متعددة المستويات، مما يجعله حلاً بسيطاً وفعالاً للتطبيقات المدمجة.

الأنظمة، MQ-9، DHT11، PWM، التهوية، PI الكلمات المفتاحية : التحكم الحراري، أردوينو أونو، متحكم المدمجة، رفض الاضطرابات، الحماية الحرارية

Abstract

This work presents the design and experimental validation of a thermal regulation system using variable-speed fans controlled by an Arduino Uno. The prototype includes DHT11 temperature sensors, an MQ-9 gas sensor, an L298N driver, an LCD display, and a buzzer. A PI controller was implemented to maintain the temperature near the setpoint while ensuring stability and accuracy. Experimental results showed a stable response with limited overshoot and low steady-state error. The system also manages disturbances and multi-level thermal protection. The solution is simple, low-cost, and suitable for embedded applications.

Keywords: Thermal regulation, Arduino Uno, PI controller, Ventilation, PWM, DHT11, MQ-9, Embedded system, Disturbance rejection, Thermal protection.

Liste des figures

Figure I.1 : Ventilation naturelle	13
Figure I.2 : Ventilation mécanique à vitesse fixe	13
Figure I.3 : : Ventilation mécanique à vitesse variable (VMV)	14
Figure I.4 : : Ventilation par fluide caloporteur	14
Figure I.5: Systèmes de climatisation	15
Figure II.1: Capteur DHT11	18
Figure II.2 : Capteur MQ-9	19
Figure II.3 Arduino Uno R3.....	20
Figure II.4: Ventilateur DC à commande PWM	22
Figure II.5 : Interface de Puissance L298N	23
Figure II.6 : Afficheur LCD 20×4 avec interface I2C.	24
Figure II.7 : Buzzer actif.....	25
Figure II.8 : LED rouge	26
Figure II.9 : Alimentation 12V 3A	27
Figure II.10 : Schéma de câblage du système réalisé sous Cirkuit Designer.	29
Figure III.1 : L'organigramme reflète le fonctionnement cyclique du programme.....	33
Figure III.2 : Instructions d'acquisition des mesures.....	34
Figure III.3 : Déclaration de la consigne dynamique.....	35
Figure III.4 : Déclaration des gains du régulateur PI dans Arduino IDE.....	36
Figure III.5 : Instructions d'implémentation du régulateur PI.....	36
Figure IV.1 : Évolution de la température sans régulation.....	41
Figure IV.2 : Évolution du système en régulation lors du premier essai	42
Figure IV.3 : Simulation PI avec compensation de pôles.	45
Figure IV.4: Réponse du système face à une consigne variable	47
Figure IV.5 : Rejet de perturbation et analyse des performances.....	48
Figure IV.6 : Valeurs réelles de l'essai de protection thermique.....	50
Figure IV.7 : Instruction d'affichage de l'état d'alerte.....	51
Figure IV.8 : Évolution de la commande PWM avec le régulateur manuel.	52
Figure IV.9 : Évolution des températures intérieure et extérieure avec le régulateur manuel.	53
Figure IV.10 : Évolution de l'erreur de régulation avec le régulateur manuel.....	53

Liste des tableaux

Table II-1: Caractéristiques techniques du capteur DHT11.	18
Table II-2 : Caractéristiques techniques du capteur MQ-9	20
Table II-3 : Caractéristiques techniques du microcontrôleur Arduino Uno R3.	21
Table II-4 : Critères techniques ayant conduit au choix de l'Arduino Uno	21
Table II-5 : Caractéristiques techniques du driver L298N	23
Table II-6 : Broches allouées aux composants du système de régulation	28
Table III-1 : Outils logiciels utilisés pour le développement du système	31
Table III-2 : Correspondance entre les équations et le programme Arduino.	36
Table IV-1: Conditions initiales retenues pour l'essai expérimental.	40
Table IV-2 : Comparaison des performances entre les deux jeux de gains.....	55

Liste des abréviations

Abréviation	Signification
ADC	Convertisseur Analogique-Numérique (Analog-to-Digital Converter)
Arduino IDE	Arduino Integrated Development Environment
CAN	Convertisseur Analogique-Numérique
CO	Monoxyde de carbone
CSV	Comma-Separated Values (valeurs séparées par des virgules)
DHT11	Digital Humidity and Temperature sensor 11
ENA / ENB	Enable A / Enable B (broches de validation du L298N)
GPL	Gaz de pétrole liquéfié (LPG)
I/O	Entrées / Sorties (Input/Output)
IHM	Interface Homme-Machine
IoT	Internet of Things (Internet des objets)
LCD	Liquid Crystal Display (affichage à cristaux liquides)
LED	Light Emitting Diode (diode électroluminescente)
MATLAB	Matrix Laboratory (logiciel de calcul numérique)
MLI	Modulation de Largeur d'Impulsion (PWM en français)
MQ-9	Capteur de gaz semi-conducteur (CO, méthane, GPL)
PCF8574	Circuit intégré d'extension d'entrées/sorties I2C
PID	Proportionnel-Intégral-Dérivé (régulateur)
PWM	Pulse Width Modulation (Modulation de Largeur d'Impulsion)
PVC	Polychlorure de vinyle (matériau des parois)
RH	Humidité Relative (Relative Humidity)
SCL	Serial Clock Line (horloge I2C)
SDA	Serial Data Line (données I2C)
Simulink	Environnement de simulation graphique MATLAB
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VMV	Ventilation Mécanique à Vitesse Variable

Table des matières

Introduction générale.....	11
I. Chapitre I État de l’art sur les systèmes de ventilation.....	12
I.1 Introduction.....	12
I.2 Évolution des systèmes de ventilation	12
I.3 Typologie des systèmes de ventilation	13
I.3.1 Ventilation naturelle.....	13
I.3.2 Ventilation mécanique à vitesse fixe	13
I.3.3 Ventilation mécanique à vitesse variable (VMV).....	13
I.3.4 Ventilation par fluide caloporteur	14
I.3.5 Systèmes de climatisation	15
I.4 Importance des systèmes de ventilation.....	15
I.5 Conclusion	15
II. Chapitre II Conception et Réalisation du Dispositif Étudié	17
II.1 Introduction.....	17
II.2 Architecture matérielle.....	17
II.2.1 Chaîne d’acquisition	17
II.2.2 Chaîne de traitement (unité centrale).....	20
II.2.3 Chaîne d’Action.....	22
II.2.4 Interface Homme-Machine (IHM).....	24
II.2.5 Gestion de l’énergie (Power Supply).....	26
II.3 Synthèse des ressources matérielles utilisées	27
II.4 Câblage du système de régulation thermique réalisé.....	29
II.5 Conclusion	30
III. Chapitre III Application de la régulation pour le contrôle du procédé de ventilation .31	
III.1 Introduction.....	31
III.2 Environnement de développement logiciel.....	31
III.3 Architecture logicielle du système et organigramme.....	32
III.3.1 Acquisition des mesures	34
III.3.2 Élaboration de la consigne	34
III.3.3 Implémentation du régulateur PI numérique	35
III.3.4 Saturation et stratégie anti-windup	36
III.3.5 Commande des ventilateurs	37
III.3.6 Gestion des alarmes et sécurité.....	37
III.3.7 Acquisition et enregistrement des données.....	37

III.4	Conclusion	38
IV.	Chapitre 4 Tests expérimentaux et analyse des performances.....	39
IV.1	Introduction.....	39
IV.2	Protocole expérimental	39
IV.3	Tests sans perturbation.....	41
IV.3.1	Essaie	42
IV.3.2	Analyse de la réponse expérimentale.....	42
IV.3.3	Comparaison avec la réponse Simulink.....	45
IV.3.4	Conclusion premier essai	46
IV.4	Rejet de perturbations	46
IV.4.1	Présentation de la perturbation appliquée	46
IV.4.2	Réponse du système à la perturbation.....	47
IV.4.3	Analyse des performances de rejet	48
IV.4.4	Conclusion sur le Rejet de perturbation.....	49
IV.5	Validation de la protection thermique.....	49
IV.5.1	Architecture de protection.....	49
IV.5.2	Vérification expérimentale.....	50
IV.6	Essai de réglage manuel et comparaison des régulateurs PI.....	51
IV.6.1	Présentation de l'essai	51
IV.6.2	Réponse du système	52
IV.6.3	Conclusion sur l'Essai 03	54
IV.6.4	Comparaison entre le réglage manuel et le réglage par compensation de pôle ...	54
IV.6.5	Analyse	55
IV.6.6	Conclusion de la comparaison	56
IV.7	Conclusion du chapitre	56
	Conclusion générale	58
	Bibliographie.....	59

Introduction générale

Dans un contexte de transition énergétique et de recherche d'optimisation des systèmes thermiques, la régulation des installations de ventilation constitue un enjeu majeur où les systèmes classiques basés sur des commandes tout-ou-rien révèlent leurs limitations en termes d'efficacité énergétique, de confort thermique et de durabilité des équipements. Face à cette problématique, ce projet vise à développer une solution complète de régulation automatique répondant à trois objectifs principaux : concevoir un régulateur PI adapté aux caractéristiques du système de ventilation, intégrer cette solution sur une plateforme microcontrôleur pour une mise en œuvre embarquée accessible, et valider expérimentalement les performances obtenues en conditions réelles.

Les travaux de modélisation, d'identification et de synthèse du régulateur PI présentés dans ce mémoire s'appuient directement sur les résultats d'un précédent mémoire de master. Dans ce dernier, un modèle dynamique du premier ordre a été établi à partir d'essais en boucle ouverte, et trois méthodes de réglage (compensation de pôle, réglage manuel, PID Tuner) ont été comparées. Le régulateur PI retenu constitue la base théorique de l'implémentation réalisée dans le présent document. Ainsi, le lecteur trouvera dans le mémoire de master associé les développements complets de modélisation et de synthèse, tandis que ce mémoire d'ingénieur se concentre sur la conception matérielle, l'implantation logicielle et la validation expérimentale du prototype réel.

Pour atteindre ces objectifs, ce mémoire s'articule autour de quatre chapitres complémentaires. Le premier chapitre présente le dispositif étudié en détaillant l'architecture matérielle complète, depuis l'unité de traitement central jusqu'à la chaîne d'acquisition basée sur des capteurs et la chaîne d'action pilotant le système de ventilation. Le deuxième chapitre expose l'implémentation logicielle incluant la configuration de l'environnement de développement, l'algorithme de régulation PI en temps réel, la gestion des alarmes de sécurité et la conception d'une interface homme-machine pour la supervision. Le troisième chapitre décrit la méthodologie de conception et d'implémentation, expliquant les choix techniques et leur mise en pratique. Enfin, le quatrième chapitre valide expérimentalement le système à travers des tests en boucle fermée, une analyse de stabilité et de précision, une évaluation du rejet de perturbations ainsi que la validation de la protection thermique multiniveaux, garantissant ainsi que la solution répond au cahier des charges initial et démontre l'intérêt d'une régulation PI embarquée pour les systèmes de ventilation.

Chapitre I État de l'art sur les systèmes de ventilation

I.1 Introduction

Le contrôle thermique constitue un enjeu majeur dans de nombreuses applications industrielles et électroniques. En effet, l'élévation de température liée au fonctionnement des équipements peut entraîner une dégradation des performances, voire des défaillances prématurées. Dans ce contexte, les systèmes de ventilation jouent un rôle essentiel en assurant le renouvellement de l'air et le maintien de conditions thermiques adéquates.

Au-delà de leur fonction primaire, les systèmes de ventilation modernes doivent répondre à des exigences croissantes en matière de précision de régulation, de consommation énergétique et de fiabilité. Cet état de l'art vise ainsi à présenter l'évolution des technologies de ventilation, leurs principales typologies, ainsi que leur importance dans les systèmes actuels.

I.2 Évolution des systèmes de ventilation

Historiquement, la ventilation reposait principalement sur des phénomènes naturels, tels que la convection thermique et la circulation de l'air induite par les différences de pression. Ces solutions passives, bien que simples et économiques, présentent des limitations importantes en termes de contrôle.

L'avènement de la révolution industrielle a marqué l'introduction de la ventilation mécanique, permettant une meilleure maîtrise des débits d'air. Par la suite, le développement de l'électricité a favorisé la généralisation de ces systèmes dans de nombreux domaines.

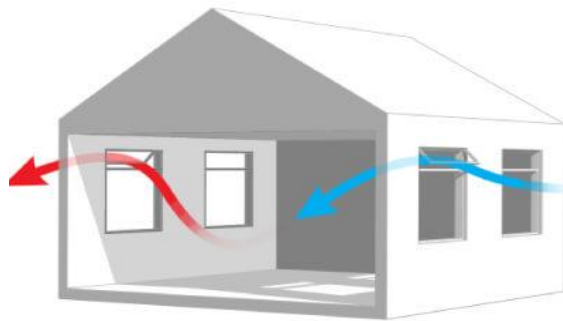
Aujourd'hui, les progrès en électronique et en automatique ont permis l'émergence de systèmes de ventilation intelligents, intégrant des capteurs et des algorithmes de commande avancés (notamment les régulateurs de type PID), capables d'adapter leur fonctionnement en temps réel aux conditions du système.

I.3 Typologie des systèmes de ventilation

Les systèmes de ventilation peuvent être classés selon leur mode de fonctionnement et leur niveau de complexité :

I.3.1 Ventilation naturelle

La ventilation naturelle repose sur les phénomènes de convection et les différences de pression. Elle ne nécessite aucune consommation énergétique, mais reste fortement dépendante des conditions environnementales, ce qui limite son efficacité dans les applications industrielles.



0.1 : Ventilation naturelle

I.3.2 Ventilation mécanique à vitesse fixe

Ce type de ventilation utilise un ventilateur fonctionnant à vitesse constante. Bien que simple à mettre en œuvre, il présente des inconvénients notables, notamment une consommation énergétique élevée et une absence d'adaptation aux besoins réels.

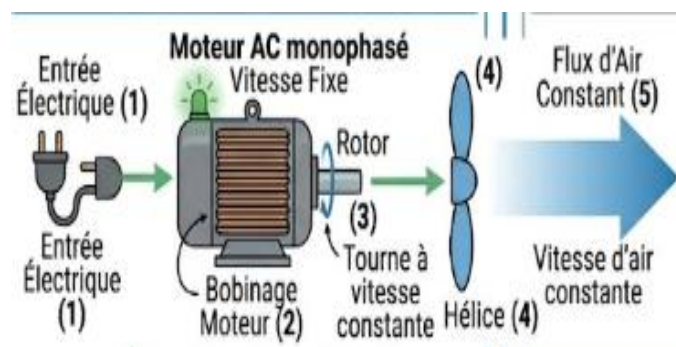


Figure 0.2 : Ventilation mécanique à vitesse fixe

I.3.3 Ventilation mécanique à vitesse variable (VMV)

La ventilation à vitesse variable constitue une amélioration significative. Elle repose sur l'utilisation d'un signal de commande de type PWM (Pulse Width Modulation) permettant de moduler la vitesse du ventilateur en fonction des conditions thermiques. Cette solution offre une meilleure efficacité énergétique, une réduction du bruit, ainsi qu'une compatibilité avec des stratégies de régulation en boucle fermée.

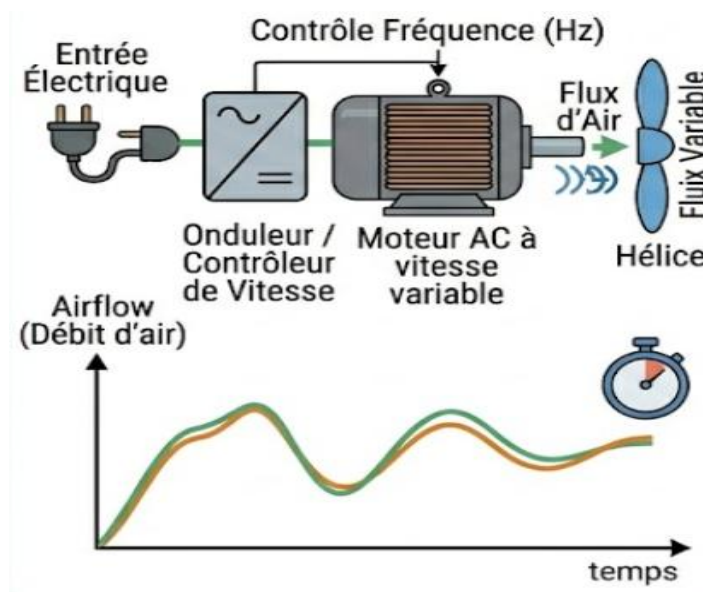


Figure 0.3 : : Ventilation mécanique à vitesse variable (VMV)

I.3.4 Ventilation par fluide caloporteur

Dans les applications à forte densité thermique, des fluides (eau, huile, etc.) sont utilisés pour transporter la chaleur. Ces systèmes sont très performants mais présentent une complexité et un coût élevés.

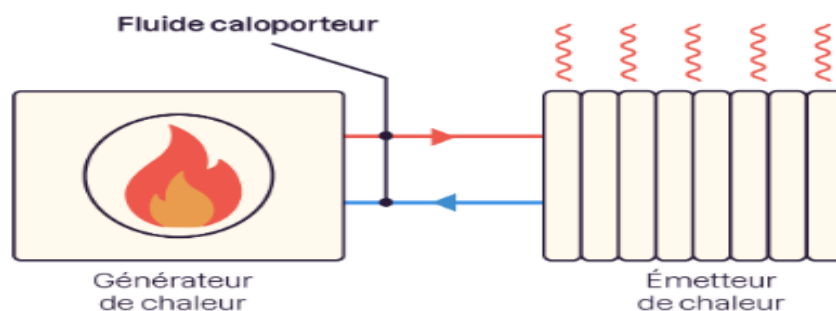


Figure 0.4: : Ventilation par fluide caloporteur

I.3.5 Systèmes de climatisation

Les systèmes de climatisation reposent sur des cycles thermodynamiques complets et permettent un contrôle précis de la température.

Ils sont principalement utilisés dans des environnements nécessitant des conditions strictes.

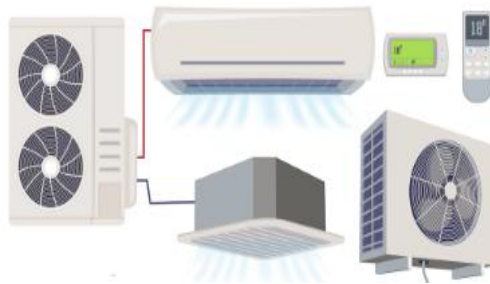


Figure 0.5: Systèmes de climatisation

I.4 Importance des systèmes de ventilation

La ventilation revêt une importance majeure selon plusieurs aspects. D'un point de vue technique, elle permet de maintenir les équipements dans une plage de fonctionnement sécurisée, évitant ainsi les surchauffes qui pourraient endommager les composants ou provoquer des pannes. Sur le plan énergétique, une gestion optimisée de la ventilation contribue à réduire significativement la consommation d'énergie, notamment en ajustant le débit d'air en fonction des besoins réels plutôt qu'en fonctionnant en continu à pleine puissance.

En ce qui concerne le confort, elle assure une répartition homogène de la température dans l'espace ventilé et crée un environnement agréable pour les occupants. Enfin, sur le plan sanitaire, elle participe activement à l'amélioration de la qualité de l'air en limitant l'accumulation de polluants, de gaz nocifs ou d'humidité excessive. Ainsi, la ventilation ne constitue pas uniquement un élément de confort, mais un facteur clé de performance et de fiabilité des systèmes.

I.5 Conclusion

Cet état de l'art a permis de mettre en évidence l'évolution progressive des systèmes de ventilation, depuis les solutions passives reposant sur la convection naturelle jusqu'aux systèmes intelligents intégrant des stratégies de régulation automatique. Parmi les

technologies analysées, la ventilation mécanique à vitesse variable (VMV) se distingue particulièrement par sa capacité à adapter en temps réel le débit d'air en fonction des besoins thermiques du système. Contrairement aux systèmes à vitesse fixe, elle présente plusieurs avantages majeurs, notamment une meilleure efficacité énergétique grâce à une consommation ajustée aux besoins réels, une précision accrue de la régulation permettant d'atteindre et de maintenir une consigne thermique via la modulation PWM, ainsi qu'une excellente compatibilité avec les lois de commande avancées, telles que les régulateurs PID ainsi, le choix de cette technologie s'impose naturellement comme solution optimale pour le système étudié dans ce projet. Le chapitre suivant sera consacré à la conception et à la réalisation du dispositif expérimental, basé sur une architecture de régulation embarquée autour d'un microcontrôleur Arduino, intégrant des capteurs de température, un circuit de commande de puissance et un ventilateur à courant continu piloté par signal PWM.

Chapitre II Conception et Réalisation du Dispositif Étudié

II.1 Introduction

La mise en œuvre d'une régulation thermique performante repose sur une architecture matérielle rigoureusement sélectionnée et intégrée. Ce chapitre est dédié à la présentation du dispositif expérimental conçu pour répondre à notre problématique de contrôle de température dans un environnement électronique.

L'objectif de cette étude est de réaliser une plateforme capable de maintenir une température de consigne au sein d'une enceinte, malgré les variations de charge thermique. Pour ce faire, le système a été dimensionné selon un cahier des charges strict, visant à garantir la fiabilité des mesures et la précision de l'action de refroidissement. Nous détaillerons dans un premier temps les composants électriques et électroniques constituant la chaîne de mesure et la chaîne de puissance. Ensuite, nous présenterons la partie informatique, articulée autour du microcontrôleur Arduino, qui assure la gestion centralisée des flux de données et des commandes. Enfin, nous exposerons l'intégration physique du montage, illustrant la structure globale du dispositif final.

Cette étape de conception matérielle constitue le socle indispensable sur lequel reposera, dans les chapitres suivants, l'implémentation des algorithmes de régulation et l'analyse de leurs performances.

II.2 Architecture matérielle

II.2.1 Chaîne d'acquisition

La chaîne d'acquisition transforme les grandeurs physiques (température, concentration de gaz) en signaux exploitables par le microcontrôleur.

II.2.1.1 Capteur DHT11

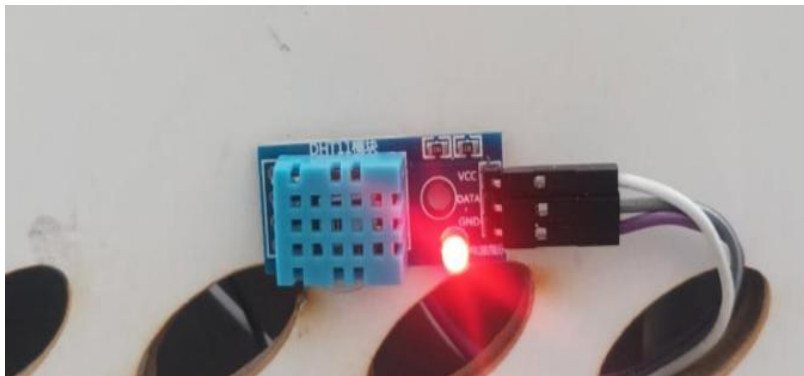


Figure 0.1: Capteur DHT11

Le DHT11 est un capteur numérique qui intègre une thermistance pour la mesure de température et un capteur capacitif pour l'humidité relative. Il contient un circuit de conversion analogique-numérique et communique via un protocole propriétaire sur un seul fil de données. Sa simplicité d'utilisation et son faible coût en font un composant très répandu dans les projets électroniques amateurs et éducatifs. [1]

Caractéristiques techniques

Le tableau suivant présente les principales caractéristiques techniques du capteur DHT11 utilisé pour l'acquisition de la température et de l'humidité.

Table 0-1: Caractéristiques techniques du capteur DHT11.

Paramètre	Valeur
Type	Numérique (thermistance + capteur capacitif)
Plage de mesure (température)	0 à 50 °C
Précision (température)	±2 °C
Résolution (température)	1 °C
Plage de mesure (humidité)	20 % à 90 % HR
Précision (humidité)	±5 % HR
Résolution (humidité)	1 % HR
Taux d'échantillonnage	1 mesure / seconde
Alimentation	3,3 V – 5,5 V
Interface	1 fil (protocole propriétaire)
Broche (intérieur)	D3
Broche (extérieur)	D2

Justification du choix

Le DHT11 a été retenu pour ce projet pour plusieurs raisons. Tout d’abord, son coût très faible permet de réaliser un prototype économique, ce qui est un critère important dans une démarche de conception embarquée à faibles ressources. Ensuite, sa mise en œuvre est simple grâce à l’existence de bibliothèques Arduino prêtes à l’emploi (DHT.h). Enfin, les plages de mesure (0 °C à 50 °C) et les précisions (± 2 °C) sont suffisantes pour une application de régulation thermique domestique ou industrielle légère, où l’on cherche avant tout à maintenir un écart faible entre la température intérieure et la température extérieure plutôt qu’une mesure absolue de très haute précision.

Cependant, ce capteur présente quelques limites : sa résolution limitée à 1 °C, sa lenteur (une mesure par 2 secondes) et son imprécision relative par rapport à des capteurs professionnels comme le SHT30 ou le DHT22. Néanmoins, pour notre cahier des charges (régulation d’un système thermique lent, constante de temps $\tau \approx 121$ s, période d’échantillonnage $T_e = 2$ s), ces limitations sont acceptables. Le DHT22 serait plus adapté pour des applications nécessitant une meilleure résolution ou une plage de température étendue [1].

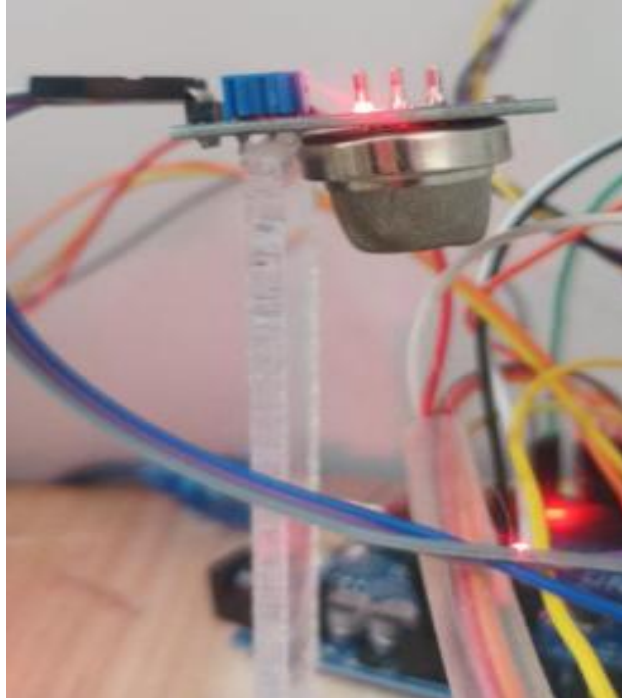
II.2.1.2 Le Capteur MQ-9

Figure 0.2 : Capteur MQ-9

Le MQ-9 est un capteur à semi-conducteur capable de détecter le monoxyde de carbone (CO), le méthane (CH₄) et le gaz de pétrole liquéfié (GPL). Il délivre une tension analogique proportionnelle à la concentration en gaz, lue sur la broche A3 de l'Arduino.

Caractéristiques techniques

Le tableau suivant présente les principales caractéristiques techniques du capteur MQ-9 utilisé pour la détection de gaz.

Table 0-2 : Caractéristiques techniques du capteur MQ-9

Paramètre	Valeur
Gaz détectés	CO, CH ₄ , GPL
Tension d'alimentation	5 V
Sortie	Analogique (0 – 1023)
Broche	A0

Justification du choix : Le MQ-9 a été choisi pour sa polyvalence (détection de plusieurs gaz), son faible coût et sa facilité d'intégration. La valeur seuil a été fixée à 500, au-delà de laquelle une alarme sonore et visuelle se déclenche [2].

II.2.2 Chaîne de traitement (unité centrale)

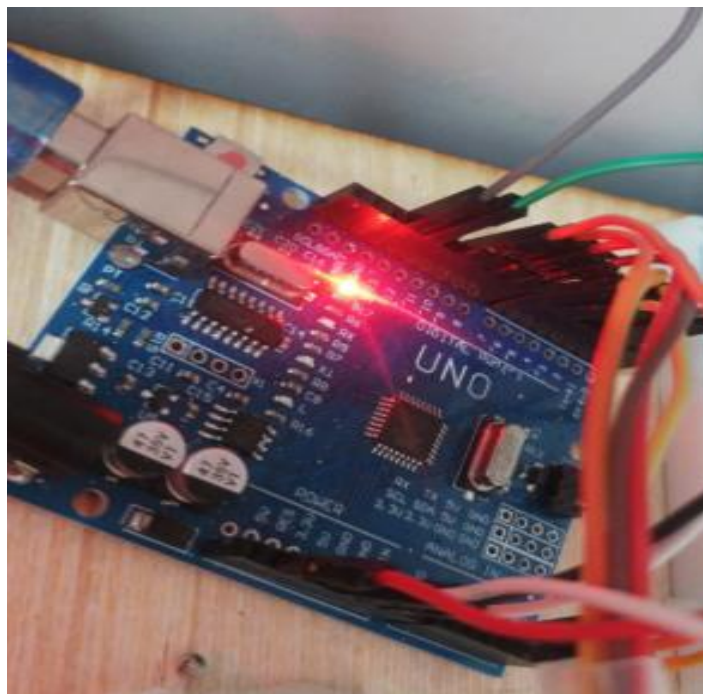


Figure 0.3 Arduino Uno R3

Le microcontrôleur constitue le « cerveau » du système de régulation thermique. Il acquiert les mesures, exécute l'algorithme de commande PI et génère les signaux PWM à destination des ventilateurs. Le choix de la plateforme **Arduino Uno R3** est justifié par sa robustesse, sa facilité d'intégration dans les systèmes embarqués et sa compatibilité totale avec les bibliothèques de traitement numérique (PID, DHT, LiquidCrystal I2C) [3].

Le tableau suivant présente les caractéristiques techniques principales du microcontrôleur Arduino Uno R3 utilisé dans le prototype.

Table 0-3 : Caractéristiques techniques du microcontrôleur Arduino Uno R3.

Paramètre	Valeur
Microcontrôleur	ATmega328P
Fréquence d'horloge	16 MHz
Mémoire Flash	32 Ko
SRAM	2 Ko
EEPROM	1 Ko
Tension d'alimentation	5 V (via USB ou régulateur interne)
Tension d'entrée recommandée	7 – 12 V
Broches numériques	14 (dont 6 PWM)
Broches analogiques	6 (A0 – A5)
Interface	USB (série)

Justification du choix technologique

Le tableau ci-dessous synthétise les critères techniques ayant conduit au choix de l'Arduino Uno pour ce projet [3].

Le tableau suivant synthétise les critères techniques ayant conduit au choix de l'Arduino Uno pour ce projet.

Table 0-4 : Critères techniques ayant conduit au choix de l'Arduino Uno

Critère	Avantage pour le projet
Fréquence (16 MHz)	Largement suffisante pour une période d'échantillonnage $T_e = 2 \text{ s}$ (dynamique lente du procédé, $\tau \approx 121 \text{ s}$)
Génération PWM	Permet un contrôle fluide et précis de la puissance des ventilateurs via le driver L298N

Ports analogiques	Indispensables pour la conversion des données du capteur MQ-9 (CAN 10 bits)
Écosystème	Accès à l'environnement Arduino IDE et aux bibliothèques standard (DHT.h, LiquidCrystal I2C, etc.)
Coût	Solution économique et largement disponible

II.2.3 Chaîne d'Action

III.2.3.1 Ventilateur



Figure 0.4: Ventilateur DC à commande PWM

L'actionneur principal du système est un ventilateur industriel à 4 fils du constructeur AVC (modèle DAS0812B2U), fonctionnant sous une tension nominale de 12 V DC pour un courant maximal de 1.20 A. Doté de roulements à billes pour une grande longévité, ce ventilateur est piloté en vitesse via un signal PWM (Modulation d'Impulsion) généré par l'Arduino, permettant d'ajuster précisément le débit d'air par convection forcée. En raison de sa consommation de puissance (14.4 W), il est interfacé avec un circuit de puissance adapté pour réguler efficacement la température de la paroi de l'enceinte.

III.2.3.2 Interface de Puissance L298N

L'organe de commande est un ventilateur DC de 12V. Étant une charge inductive et gourmande en courant (jusqu'à 1,20 A par ventilateur), il nécessite une interface de puissance adaptée. Le module L298N a été retenu pour cette fonction. [4]

Caractéristiques techniques

Le tableau suivant présente les caractéristiques techniques du driver L298N utilisé pour la commande des ventilateurs.

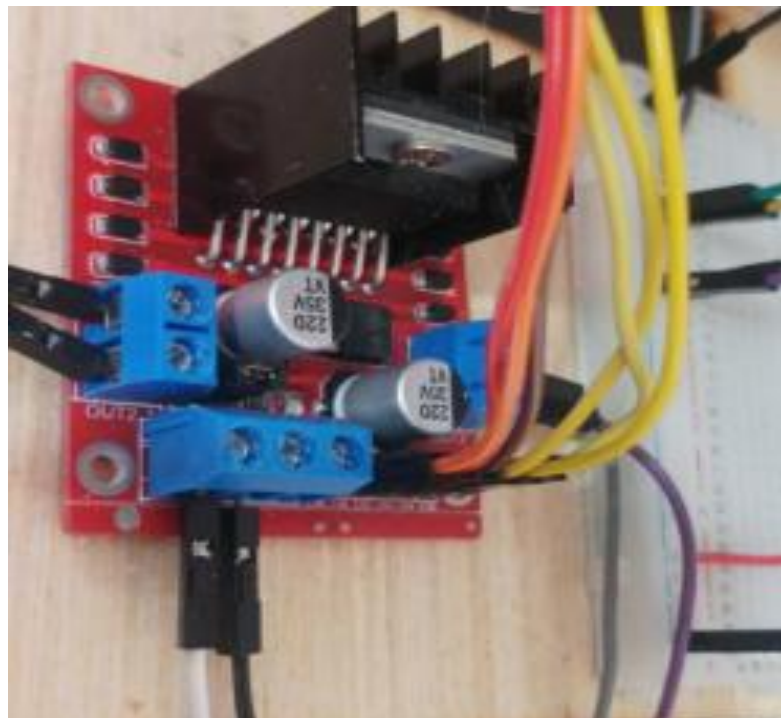


Figure 0.5 : Interface de Puissance L298N

Table 0-5 : Caractéristiques techniques du driver L298N

Paramètre	Valeur
Type	Double pont en H
Tension moteur (VS)	5 – 12 V
Tension logique (VSS)	5 V
Courant maximal par voie	2 A (continu)
Courant de pointe	3 A
Chute de tension interne	≈ 1,8 V
Fréquence PWM	Jusqu'à 25 kHz (490 Hz typique sur Arduino)

Protection	Diodes de roue libre intégrées
Logique de commande	TTL (0 – 5V)

Justification du choix technologique

Le L298N a été sélectionné pour plusieurs raisons. Tout d'abord, il intègre des diodes de roue libre qui protègent le circuit contre les surtensions générées lors des commutations inductives des moteurs. Ensuite, son interface est directement compatible avec Arduino grâce à ses signaux logiques en 5V, ce qui simplifie considérablement le câblage et la programmation. Sur le plan de la robustesse, son courant maximal de 2A est largement supérieur aux 0,2A consommés par chaque ventilateur, offrant ainsi une marge de sécurité confortable. Enfin, sa disponibilité sous forme de module complet prêt à l'emploi, incluant un régulateur 5V intégré, en fait une solution pratique et économique pour ce type d'application.

Remarque sur la chute de tension

Le L298N étant basé sur des transistors bipolaires, il engendre une chute de tension interne d'environ 1,8 V. Pour une alimentation de 12 V, la tension maximale effective aux bornes des ventilateurs est donc d'environ 10,2 V. Cette limitation est toutefois acceptable pour notre application.

II.2.4 Interface Homme-Machine (IHM)

III.2.4.1 LCD 20×4 avec interface I2C



Figure 0.6 : Afficheur LCD 20×4 avec interface I2C.

L'afficheur LCD 20×4 avec module I2C sert d'interface homme-machine. Il permet d'afficher en temps réel la température intérieure, la consigne, l'erreur de régulation, la valeur PWM, l'état du capteur MQ-9 ainsi que le niveau d'alerte du système. Grâce à l'interface I2C, le nombre de broches nécessaires est réduit à quatre (VCC, GND, SDA, SCL), ce qui préserve les ressources de l'Arduino pour d'autres périphériques.

Les paramètres affichés sont les suivants :

- Ligne 0 : Température intérieure T_{int} et consigne dynamique T_{ext} - Ligne 1 : Erreur de régulation et valeur PWM
- Ligne 2 : Valeur du capteur MQ-9 et état du buzzer
- Ligne 3 : État de sécurité (NORMAL, WARNING, CRITICAL, EMERGENCY, GAZ)

L'afficheur LCD offre une autonomie de visualisation sans nécessiter de connexion permanente à un ordinateur, ce qui est essentiel pour un système embarqué. L'utilisation de l'interface I2C simplifie le câblage et libère des broches numériques pour d'autres capteurs ou actionneurs.

III.2.4.2 Buzzer actif



Figure 0.7 : Buzzer actif

Le buzzer actif intégré au module constitue un dispositif d'alerte sonore particulièrement adapté aux systèmes de sécurité environnementale, puisqu'il permet de transformer une simple détection en une alarme audible immédiate facilement perceptible par l'utilisateur. Contrairement au buzzer passif, qui nécessite un signal oscillant externe pour produire un son, le buzzer actif possède un oscillateur interne intégré. Ainsi, l'application d'une tension continue

sur sa broche de commande suffit pour générer automatiquement un signal sonore continu, généralement autour de 2 kHz, avec une intensité pouvant atteindre environ 85 dB à une distance de 10 cm.

Ce composant fonctionne dans une plage de tension comprise entre 3,3 V et 5 V, tout en consommant seulement quelques dizaines de milliampères. Cette faible consommation ainsi que sa compatibilité directe avec la majorité des microcontrôleurs permettent une intégration simple, sans nécessiter de circuit de commande complexe. Son choix dans ce projet repose principalement sur sa simplicité d'utilisation, son coût réduit, sa fiabilité ainsi que sa réactivité instantanée. En effet, un simple état logique HIGH appliqué sur la broche de contrôle permet son déclenchement immédiat, ce qui représente un avantage important dans les applications liées à la sécurité où une réaction rapide est essentielle [4].

III.2.4.3 LED



Figure 0.8 : LED rouge

Une LED (Light Emitting Diode) est un composant électronique qui émet de la lumière lorsqu'il est parcouru par un courant électrique. Elle se caractérise par sa faible consommation, sa longue durée de vie et son temps de réponse très rapide.

Dans ce projet, la LED rouge intégrée sur la broche D13 de l'Arduino Uno est utilisée comme indicateur visuel d'alarme. Elle s'allume simultanément au buzzer dans les situations suivantes :

- Dépassement des seuils de température (WARNING, CRITICAL, EMERGENCY)
- Détection de gaz par le MQ-9 (valeur > 500)
- Écart entre T_{int} et T_{ext} supérieur à 3 °C

Cette signalisation visuelle permet une supervision rapide du système même en l'absence d'affichage LCD.

II.2.5 Gestion de l'énergie (Power Supply)

Le système utilise une alimentation hybride :

Logique : 5V stable via l'USB ou le régulateur interne de l'Arduino.

Puissance : 12V externe dédié au moteur pour éviter que les pics d'appel de courant du ventilateur ne fassent "redémarrer" le microcontrôleur.



Figure 0.9 : Alimentation 12V 3A

➤ Isolation galvanique :

La règle critique dit que les masses (GND) logique et puissance sont communes, mais les alimentations +5V et +12V sont séparées pour éviter les chutes de tension sur la ligne 5V lors des appels de courant du moteur, les perturbations (bruit) sur la partie logique et les risques de reset intempestif de l'Arduino

II.3 Synthèse des ressources matérielles utilisées

Afin de répondre aux exigences du cahier des charges, les ressources de la carte Arduino Uno ont été allouées aux différents composants du système de régulation thermique comme indiqué dans le tableau suivant :

Le tableau suivant présente l'allocation des broches Arduino aux différents composants du système de régulation.

Table 0-6 : Broches allouées aux composants du système de régulation

Broche Arduino	Fonction
D2	DHT11 intérieur
D3	DHT11 extérieur
D5	IN4 (L298N, ventilateur 2)
D6	ENB (PWM ventilateur 2)
D7	IN3 (L298N, ventilateur 2)
D8	IN2 (L298N, ventilateur 1)
D9	IN1 (L298N, ventilateur 1)
D10	ENA (PWM ventilateur 1)
D11	Buzzer actif
D13	LED rouge
A3	MQ-9 (sortie analogique)
A4	SDA (LCD I2C)
A5	SCL (LCD I2C)
5V	Alimentation capteurs / LCD
GND	Masse commune
12V externe	Alimentation L298N + ventilateurs

Note technique : La fréquence d'horloge de 16 MHz de l'ATmega328P constitue un avantage important pour l'implémentation de l'algorithme de régulation. Étant donné que la dynamique du procédé thermique est relativement lente, avec une constante de temps expérimentale de

l'ordre de 121 s, cette fréquence permet d'exécuter les opérations d'acquisition, de calcul du régulateur PI et de commande des actionneurs sans latence perceptible. Le microcontrôleur dispose ainsi d'une capacité de traitement largement suffisante pour assurer une exécution fiable de la boucle de régulation avec une période d'échantillonnage fixée à ($T_e = 2$)secondes.

II.4 Câblage du système de régulation thermique réalisé

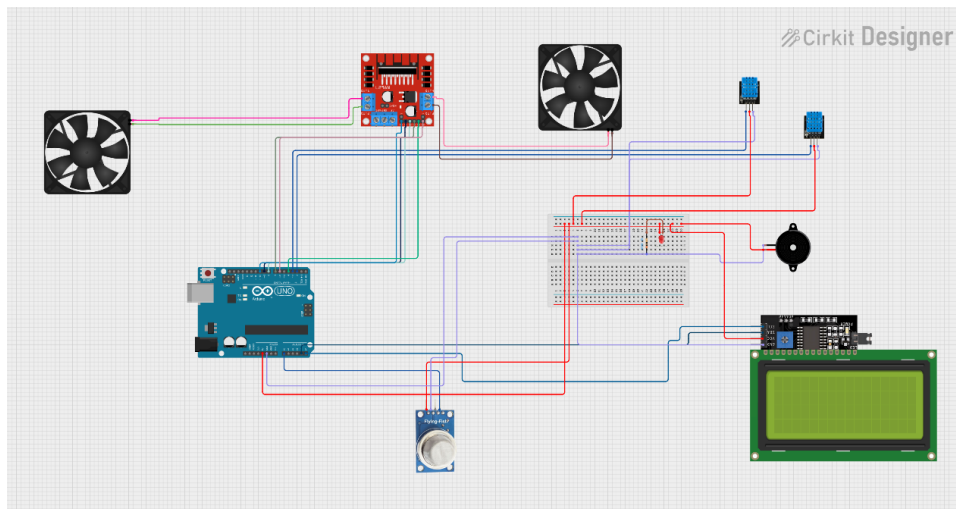


Figure 0.10 : Schéma de câblage du système réalisé sous Cirkuit Designer.

Schéma illustre l'architecture matérielle complète du prototype développé. La carte Arduino Uno constitue l'unité centrale de traitement, recevant les mesures de température des deux capteurs DHT11 (intérieur et extérieur) ainsi que la détection de gaz du capteur MQ-9 (Flying-Fish). En sortie, l'Arduino pilote le module L298N qui commande les deux ventilateurs DC selon le signal PWM calculé par le régulateur PI. La supervision locale est assurée par un afficheur LCD et un buzzer pour la signalisation des alarmes, le tout connecté via une plaque d'essai (breadboard) facilitant les interconnexions.

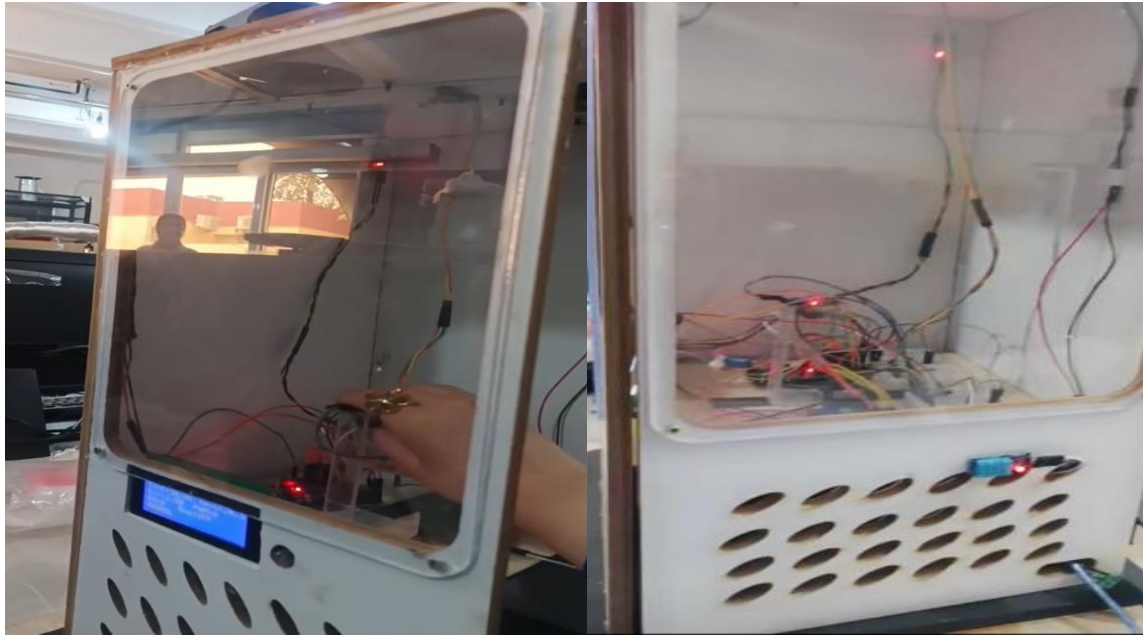


Figure 0.11: prototype finale

II.5 Conclusion

Dans ce chapitre, nous avons présenté l'ensemble de l'architecture matérielle retenue pour la réalisation du système de régulation thermique. Le choix de la plateforme Arduino Uno R3, associé à une chaîne d'acquisition composée de deux capteurs DHT11 et d'un capteur de gaz MQ-9, ainsi qu'à une chaîne d'action basée sur le driver L298N pilotant deux ventilateurs à courant continu, permet d'obtenir une solution simple, économique et parfaitement adaptée aux objectifs du projet.

L'intégration d'un afficheur LCD 20×4 avec interface I2C, d'un buzzer actif et d'une LED rouge (D13) assure une interaction efficace avec l'utilisateur. Ces éléments permettent la surveillance en temps réel des variables du système (températures, erreur, PWM, état du gaz) ainsi que la gestion des alarmes de sécurité (visuelle et sonore). L'ensemble des composants a été sélectionné de manière à garantir la compatibilité matérielle, la facilité d'implémentation et la fiabilité du fonctionnement.

Cette plateforme constitue ainsi le support expérimental nécessaire à l'implémentation de la stratégie de commande développée dans la suite du mémoire. Le chapitre suivant sera consacré à l'implémentation logicielle du régulateur PI, à la gestion des alarmes et à la validation expérimentale du système en boucle fermée.

Chapitre III Application de la régulation pour le contrôle du procédé de ventilation

III.1 Introduction

Après avoir présenté l'architecture matérielle du système dans le chapitre précédent, ce chapitre est consacré à l'étude de la partie logicielle et à l'implémentation de l'algorithme de régulation embarqué sur la carte Arduino Uno.

Les travaux de modélisation, d'identification du procédé et de synthèse du régulateur PI ont été réalisés dans le cadre de mon mémoire de master associé à ce projet. Un modèle dynamique du premier ordre y a été établi à partir d'essais en boucle ouverte, conduisant au régulateur PI suivant : $K_p = 3,418$ et $K_i = 0,0282$. Ce chapitre s'appuie directement sur ces résultats pour les concrétiser sur la plateforme expérimentale.

L'objectif est ainsi de développer un programme capable d'acquérir les mesures issues des capteurs, de calculer en temps réel la loi de commande PI et de piloter automatiquement les ventilateurs afin de maintenir la température de l'enceinte au voisinage de la consigne dynamique, constituée par la température extérieure.

Le programme développé assure également la supervision du système grâce à l'enregistrement des données expérimentales au format exploitable, ainsi que la gestion des fonctions de sécurité associées au capteur MQ-9 et au buzzer. Ce chapitre présente successivement l'environnement de développement logiciel utilisé, l'architecture logicielle du système, l'acquisition des mesures, l'élaboration de la consigne dynamique, l'implémentation du régulateur PI numérique avec sa stratégie anti-windup, la commande des ventilateurs, la gestion des alarmes, ainsi que l'acquisition et l'enregistrement des données vers le moniteur séri

III.2 Environnement de développement logiciel

Le tableau suivant présente les outils logiciels utilisés pour le développement, la simulation et l'exploitation du système.

Table 0-1 : Outils logiciels utilisés pour le développement du système

Outil / Bibliothèque	Fonction dans le projet
Arduino IDE 2.3.8	Développement, compilation et téléversement du programme embarqué vers l'Arduino Uno
Langage C/C++ (Arduino)	Implémentation de l'algorithme de régulation PI, gestion des entrées/sorties et traitement des données
Bibliothèque <i>DHT.h</i>	Acquisition des températures intérieure et extérieure à partir des capteurs DHT11
Serial Monitor	Visualisation et acquisition des données expérimentales en temps réel
MATLAB / Simulink	Modélisation du procédé, synthèse du régulateur PI et validation par simulation

Le développement du programme embarqué a été réalisé à l'aide de l'environnement Arduino IDE 2.3.8. Cet environnement permet l'écriture, la compilation et le téléversement du programme vers le microcontrôleur ATmega328P de la carte Arduino Uno.

Le langage utilisé est le langage C/C++ adapté aux systèmes embarqués Arduino. Plusieurs bibliothèques ont été intégrées afin de faciliter l'accès aux périphériques du système.

L'utilisation combinée d'Arduino IDE, MATLAB/Simulink et Python permet de couvrir l'ensemble de la chaîne de développement du système. Arduino IDE assure l'implémentation de l'algorithme embarqué, MATLAB/Simulink est utilisé pour la modélisation et la validation théorique du régulateur PI, tandis que Python permet le traitement et la visualisation des données expérimentales obtenues lors des essais en boucle fermée. [6]

III.3 Architecture logicielle du système et organigramme

Le programme est organisé sous forme modulaire autour d'une boucle principale exécutée toutes les 2 secondes. Le fonctionnement général peut être décomposé en plusieurs étapes :

1. Acquisition des températures intérieure et extérieure.

2. Lecture du capteur MQ-9.
3. Détermination de la consigne.
4. Calcul de l'erreur de régulation.
5. Exécution du régulateur PI.
6. Saturation de la commande.
7. Commande PWM des ventilateurs.
8. Gestion des alarmes.
9. Transmission des données vers le Serial Monitor.

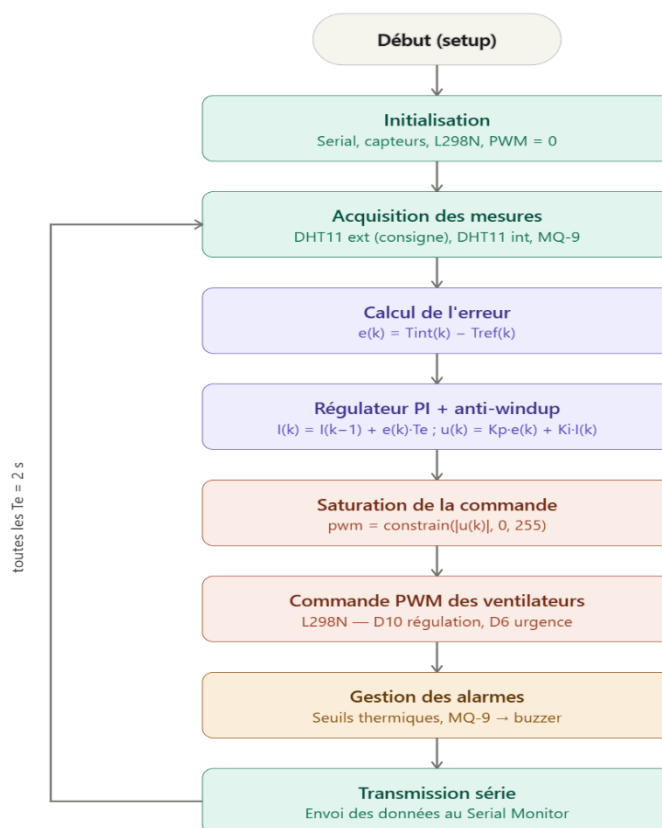


Figure 0.1 : L'organigramme reflète le fonctionnement cyclique du programme.

L'organigramme reflète le fonctionnement cyclique du programme. Après l'initialisation (configuration des broches, démarrage des capteurs et de la communication série), le système entre dans une boucle exécutée toutes les 2 secondes. À chaque cycle, il acquiert les températures intérieure et extérieure ainsi que la valeur du capteur de gaz MQ-9, calcule l'erreur

de régulation, puis exécute le régulateur PI muni de sa stratégie anti-windup pour déterminer la commande. Cette commande est ensuite saturée dans la plage $[0, 255]$ avant d'être convertie en signal PWM appliqué aux ventilateurs via le L298N. En parallèle, le système vérifie les seuils thermiques et la lecture du MQ-9 pour déclencher, si nécessaire, le buzzer et le ventilateur d'urgence. Enfin, l'ensemble des données est transmis au moniteur série pour enregistrement, avant que le cycle ne reprenne.

III.3.1 Acquisition des mesures

L'acquisition des grandeurs physiques repose sur deux capteurs DHT11 :

- DHT11 intérieur (broche D3) : mesure de la température de l'enceinte (T_{int}).
- DHT11 extérieur (broche D2) : mesure de la température ambiante (T_{ext}).

Les mesures sont effectuées toutes les : $[T_e = 2s]$ afin de respecter la dynamique lente du procédé thermique.

Le capteur MQ-9 est connecté à l'entrée analogique A3 et permet la détection de fumée ou de gaz.

```

26 // ----- Capteurs -----
27 #define PIN_DHT_INT 2      // T_int (enceinte)
28 #define PIN_DHT_EXT 3      // T_ext (consigne)
29 #define PIN_MQ9 A0
30 #define PIN_BUZZER 11
31 #define PIN_LED 13
32
33 DHT dhtInt(PIN_DHT_INT, DHTTYPE);
34 DHT dhtExt(PIN_DHT_EXT, DHTTYPE);

```

Figure 0.2 : Instructions d'acquisition des mesures.

III.3.2 Élaboration de la consigne

Contrairement aux systèmes utilisant une consigne fixe, la stratégie retenue consiste à utiliser la température extérieure comme consigne dynamique :

$$[T_{ref}(k) = T_{ext}(k)]$$

Cette approche est cohérente avec un système de refroidissement par ventilation puisque la température intérieure ne peut pas descendre durablement en dessous de la température ambiante.

```
163 // Consigne dynamique = température extérieure  
164 consigne = T_ext;
```

Figure 0.3 : Déclaration de la consigne dynamique.

III.3.3 Implémentation du régulateur PI numérique

Le régulateur utilisé est un correcteur PI, obtenu à partir du modèle expérimental identifié du procédé thermique. Les paramètres retenus sont :

$$Kp = 3,418 \text{ et } Ki = 0,0282$$

La loi de commande discrète, programmée selon la méthode d'Euler, est donnée par les équations suivantes :

$$e(k) = T_{int}(k) - T_{ref}(k)$$

$$I(k) = I(k-1) + e(k) \cdot Te$$

$$u(k) = Kp \cdot e(k) + Ki \cdot I(k)$$

Où $e(k)$ représente l'erreur de régulation à l'instant k , $I(k)$ le terme intégral cumulé, et $u(k)$ la commande calculée avant saturation.

Le tableau suivant présente la correspondance entre les équations du régulateur PI et leur implémentation dans le programme Arduino.

Table 0-2 : Correspondance entre les équations et le programme Arduino.

Grandeur	Équation
Erreur	$e(k) = T_{int}(k) - T_{ref}(k)$
Intégrale	$I(k) = I(k-1) + e(k) \cdot T_e$
Commande PI	$u(k) = K_p \cdot e(k) + K_i \cdot I(k)$

La séquence logique implémentée dans la fonction `loop()` de l'Arduino se déroule ainsi : à chaque période d'échantillonnage $T_e = 2$ s, le programme calcule l'erreur brute $e(k)$, met à jour le terme intégral $I(k)$, puis évalue la commande non saturée $u(k)$. Cette commande est ensuite soumise à la saturation et à la stratégie anti-windup détaillées ci-après, avant d'être appliquée aux ventilateurs sous forme de signal PWM.

```

46 // ----- Paramètres PI -----
47 const float Kp = 3,418 ;
48 const float Ki = 0,0282;
49 const float Te = 2.0;

```

Figure 0.4 : Déclaration des gains du régulateur PI dans Arduino IDE.

```

193 // --- Phase 2 : régulation PI ---
194 erreur = T_int - consigne;
195
196 float integraleTemp = integrale + erreur * Te;
197 float commandeTemp = Kp * erreur + Ki * integraleTemp;

```

Figure 0.5 : Instructions d'implémentation du régulateur PI.

III.3.4 Saturation et stratégie anti-windup

Le signal PWM généré par l'Arduino est codé sur 8 bits, ce qui impose la contrainte $0 \leq \text{PWM} \leq 255$. Si la commande calculée $u(k)$ dépasse cette plage, elle est automatiquement ramenée à sa valeur limite (0 ou 255).

Afin d'éviter l'accumulation excessive du terme intégral lorsque la commande sature — phénomène connu sous le nom de *windup* — une stratégie anti-windup conditionnelle est mise en œuvre : le terme intégral $I(k)$ n'est mis à jour que si la commande $u(k)$ reste dans la plage $[0, 255]$. Lorsque la saturation est atteinte, l'intégrateur est gelé à sa dernière valeur valide.

Cette technique améliore la stabilité du système et réduit significativement le temps de retour au régime permanent après une saturation prolongée.

III.3.5 Commande des ventilateurs

Les ventilateurs sont pilotés par le module L298N.

Les signaux PWM sont appliqués sur :

- D10 : ventilateur 1
- D6 : ventilateur 2

La valeur PWM calculée par le régulateur est convertie en tension moyenne appliquée aux ventilateurs selon : $U = \frac{PWM}{255} \times 12$ L'augmentation du PWM entraîne une augmentation du débit d'air et donc une intensification du refroidissement.

III.3.6 Gestion des alarmes et sécurité

En complément de la régulation thermique, le programme intègre une fonction de sécurité qui surveille en permanence deux grandeurs : la température intérieure et la concentration de gaz mesurée par le capteur MQ-9 sur l'entrée A3.

Le buzzer est activé dans deux cas : si la température intérieure dépasse un seuil critique, ou si le MQ-9 détecte une concentration anormale de fumée ou de gaz. Dans ces situations, le système active également le ventilateur 2 (D6) à pleine puissance (PWM = 255), indépendamment du régulateur PI, afin d'assurer un refroidissement d'urgence.

Cette fonction de sécurité opère en parallèle de la régulation normale et améliore la robustesse globale du système face aux situations critiques. Cette fonction améliore la sécurité globale du système.

III.3.7 Acquisition et enregistrement des données

Afin de permettre l'analyse expérimentale du système, les données sont transmises en temps réel vers le Serial Monitor. Chaque enregistrement contient :

Temps, température intérieure, température extérieure, consigne, erreur, commande PWM et valeur MQ-9. Les données sont ensuite exportées vers Python pour le traitement graphique et l'évaluation des performances du régulateur.

III.4 Conclusion

Dans ce chapitre, j'ai présenté l'architecture logicielle du système de régulation thermique ainsi que l'implémentation du régulateur PI sur la plateforme Arduino Uno. Le programme développé permet l'acquisition des mesures, le calcul de la commande en temps réel grâce à la loi PI avec anti-windup, la commande des ventilateurs, ainsi que la gestion des alarmes de sécurité liées à la température et au capteur MQ-9.

Cette implémentation constitue la base du système de contrôle embarqué. Le chapitre suivant sera consacré à sa validation expérimentale, à travers des tests en boucle fermée, l'analyse de la stabilité et de la précision, ainsi que l'évaluation du Rejet de perturbations, afin de vérifier que le système répond bien au cahier des charges initial.

Chapitre 4 Tests expérimentaux et analyse des performances

IV.1 Introduction

Ce chapitre est consacré à la validation expérimentale du système de régulation thermique par ventilation développé dans les chapitres précédents.

L'objectif principal, défini dans le cahier des charges initial, est de maintenir la température intérieure de l'enceinte autour d'une consigne, avec une erreur statique nulle, un temps de réponse rapide et un dépassement limité, tout en assurant un rejet efficace des perturbations thermiques. Pour répondre à ces exigences, un régulateur PI a été synthétisé à partir du modèle expérimental du procédé, avec les paramètres $K_p = 3,418$ et $K_i = 0,0282$, et une période d'échantillonnage $T_e = 2$ s.

Ce chapitre présente le protocole expérimental adopté, les courbes de réponse obtenues en boucle fermée, l'analyse des indicateurs de performance (stabilité, précision, rapidité), ainsi que les résultats des tests de Rejet de perturbations et de la protection thermique.

IV.2 Protocole expérimental

Le protocole expérimental a pour objectif de valider le comportement réel du système de régulation thermique en boucle fermée. Le montage utilisé est basé sur une carte Arduino Uno (ATmega328P), qui assure l'acquisition des températures, le calcul de la commande PI et le pilotage des ventilateurs à travers le driver de puissance L298N.

Les deux capteurs DHT11 sont connectés sur des entrées numériques de l'Arduino : le capteur intérieur sur D3 mesure la température de l'enceinte T_{int} , et le capteur extérieur sur D2 mesure la température ambiante T_{amb} . Deux ventilateurs DC 12 V sont commandés en PWM par le module L298N. L'écran LCD affiche en temps réel les températures mesurées ainsi que l'état du système, tandis qu'un buzzer signale toute température excessive ou écart anormal entre les capteurs. Le capteur MQ-9, connecté à l'entrée analogique A3, assure quant à lui la détection de fumée ou de gaz.

Les conditions initiales retenues pour l'essai sont résumées dans le tableau suivant :

Le tableau suivant présente les conditions initiales retenues pour la réalisation de l'essai expérimental.

Table 0-1: Conditions initiales retenues pour l'essai expérimental.

Paramètre	Valeur
Température initiale intérieure ($T_{\text{int}}(0)$)	35,0 °C
Température ambiante (T_{amb})	29,5 °C
Consigne ($T_{\text{ref}} = (T_{\text{amb}})$)	29,5 °C
Période d'échantillonnage (T_e)	2 s
Durée d'acquisition	10000 s
Échelon PWM initial	180
Tension équivalente	8,47 V

L'acquisition des données est réalisée toutes les 2secondes. À chaque période d'échantillonnage, l'Arduino lit les deux températures, calcule l'erreur de régulation, applique la loi PI, puis envoie la commande PWM vers les deux ventilateurs. Les données expérimentales sont relevées à l'aide du Serial Monitor de l'Arduino IDE, puis exportées vers Python afin de tracer les courbes $T_{\text{int}}(t)$, $T_{\text{amb}}(t)$, la consigne et la commande PWM.

Ce protocole permet ensuite d'analyser les performances réelles du système : erreur statique, dépassement, temps de réponse, stabilité et capacité de rejet des perturbations. Ces critères seront comparés au cahier des charges défini précédemment, qui impose notamment une erreur statique quasi nulle, un dépassement limité et une réponse stable.

La tension équivalent de 8,47 V pour un PWM = 180 correspond bien à $180/255 \times 12$

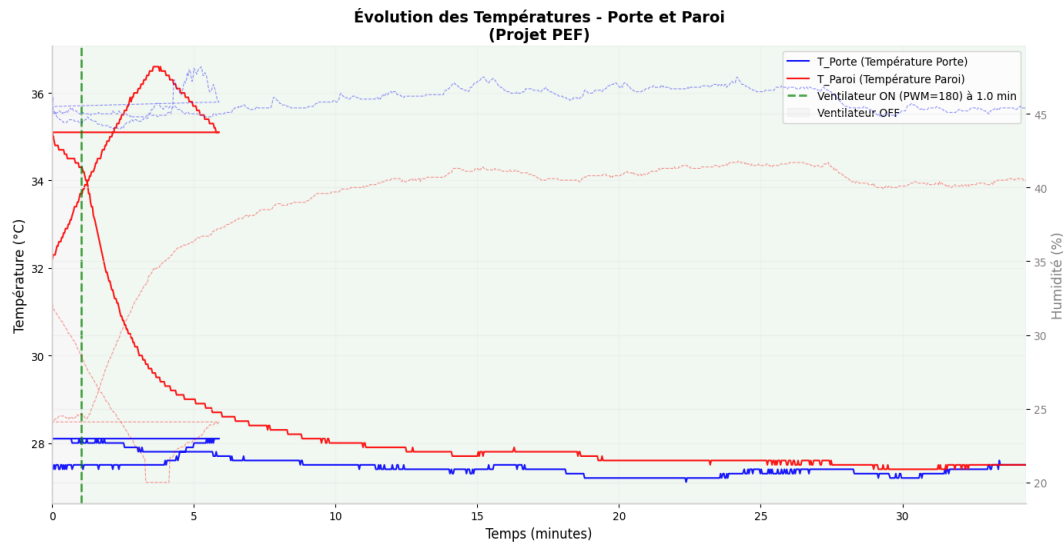


Figure 0.1 : Évolution de la température sans régulation.

IV.3 Tests sans perturbation

L'objectif de cet essai est de valider expérimentalement le comportement du régulateur PI implanté sur la plateforme Arduino et de vérifier sa capacité à maintenir la température de l'enceinte autour de la consigne imposée. Le test consiste à appliquer une consigne fixe de 30 °C puis à enregistrer l'évolution temporelle de la température intérieure mesurée par le capteur DHT11.

Les données expérimentales sont acquises toutes les 2 secondes pendant une durée . Simultanément, la réponse simulée du modèle identifié au chapitre III est générée sous MATLAB/Simulink à partir des mêmes conditions initiales afin de permettre une comparaison directe entre le comportement théorique et le comportement réel du système.

IV.3.1 Essaie

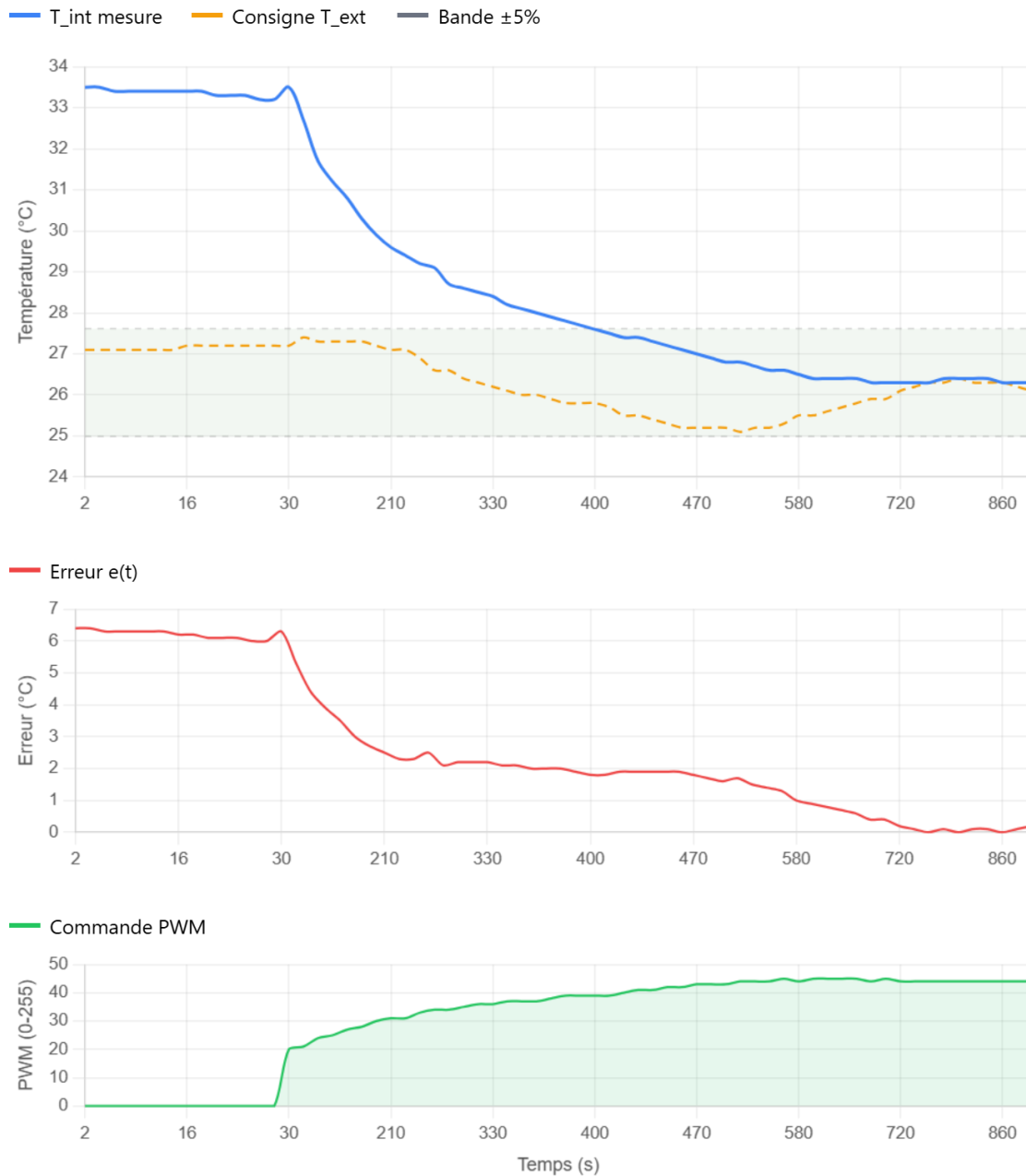


Figure 0.2 : Évolution du système en régulation lors du premier essai

IV.3.2 Analyse de la réponse expérimentale

La Figure IV-2 présente la réponse expérimentale obtenue avec le régulateur PI. La température intérieure démarre à environ 33,5 °C alors que la température extérieure, utilisée comme consigne dynamique, est voisine de 27,2 °C. Durant les 30 premières secondes, les ventilateurs restent désactivés afin d'observer l'état initial du système. À l'activation de la

régulation, le signal PWM passe immédiatement d'une valeur nulle à environ 20, puis augmente progressivement jusqu'à atteindre 44–45 en régime établi.

La température intérieure décroît continûment sous l'action du système de ventilation. La réponse est monotone et ne présente aucune oscillation importante ni instabilité. La température passe progressivement de 33,5 °C à environ 26,3 °C, valeur très proche de la consigne finale. L'erreur de régulation diminue continuellement, passant d'environ 6,3 °C au début de l'essai à une valeur quasi nulle en régime permanent. L'erreur statique finale est de l'ordre de 0 à 0,1 °C, ce qui confirme la bonne action du terme intégral du correcteur PI. Aucun dépassement significatif n'est observé, la température rejoignant la consigne de manière progressive et stable. Ces résultats montrent que le régulateur assure une régulation précise et robuste du système thermique.

III.2.4.4 Analyse de la stabilité et de la précision

Afin d'évaluer les performances du système de régulation thermique, plusieurs indicateurs classiques de l'automatique ont été déterminés à partir de la courbe expérimentale obtenue en boucle fermée. Les principaux critères étudiés sont l'erreur statique, le dépassement et le temps de réponse à 5 %.

III.2.4.5 Erreur statique

L'erreur statique correspond à la différence entre la consigne et la valeur de sortie lorsque le système a atteint son régime permanent. Elle est définie par :

$$[\varepsilon = w - y(\infty)]$$

Où (w) représente la consigne et ($y(\infty)$) la valeur finale de la température intérieure.

D'après les résultats expérimentaux, la température intérieure converge vers (26,3°C), tandis que la consigne finale est également voisine de (26,3°C).

L'erreur statique obtenue est donc :

$$[\varepsilon \approx 0^\circ\text{C}]$$

Cette valeur confirme l'efficacité du terme intégral du correcteur PI, qui permet d'annuler pratiquement l'erreur permanente.

IV.2.4.6 Dépassement

Le dépassement caractérise l'écart maximal de la sortie par rapport à la consigne durant la phase transitoire. Il est calculé à l'aide de la relation : [D]

L'analyse de la courbe expérimentale montre que la température intérieure rejoint progressivement la consigne sans la dépasser. Aucun pic ou oscillation importante n'est observé pendant la phase transitoire.

Par conséquent : [D $\approx$ 0]

Le système satisfait donc pleinement l'exigence du cahier des charges relative à la limitation du dépassement.

IV.2.4.7 Temps de réponse à 5 %

Le temps de réponse à 5 % correspond au temps nécessaire pour que la sortie entre dans la bande de tolérance définie par :

w – 5 et y reste définitivement.

Pour une consigne finale voisine de (26, 3°C), la bande de tolérance est :

$$[[24, 99^{\circ}\text{C}; 27, 62^{\circ}\text{C}]]$$

L'observation de la courbe expérimentale montre que la température intérieure entre dans cette bande vers (t $\approx$ 430; s) et n'en ressort plus.

Le temps de réponse mesuré est donc : [t_{r5}]

Cette valeur est supérieure à celle obtenue en simulation, ce qui s'explique par les phénomènes réels non modélisés tels que les pertes thermiques, les variations de la température ambiante, les limitations du driver L298N ainsi que la dynamique réelle des ventilateurs.

IV.3.3 Comparaison avec la réponse Simulink

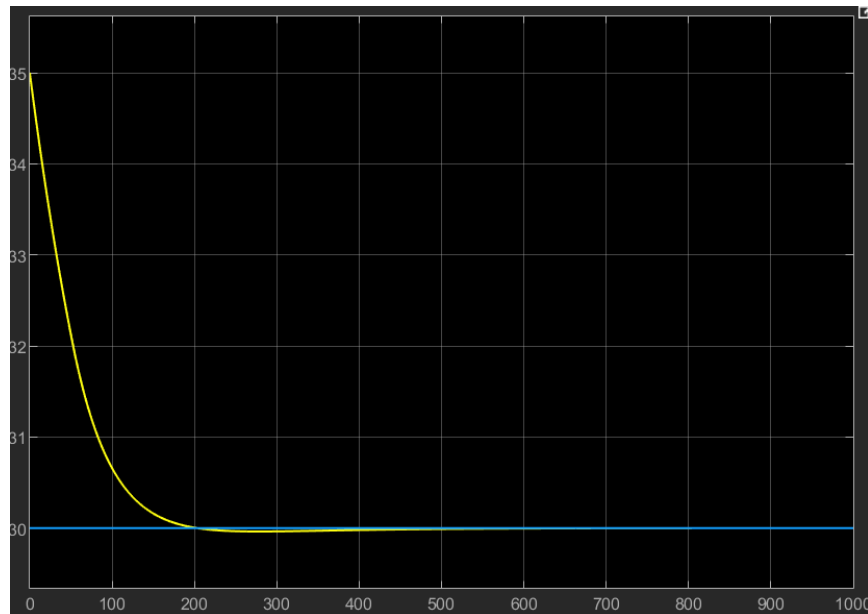


Figure 0.3 : Simulation PI avec compensation de pôles.

La Figure IV-3 présente la réponse simulée sous MATLAB/Simulink obtenue à partir du modèle identifié du procédé et du même régulateur PI. La simulation démarre à 35 °C pour une consigne fixe de 30 °C. La température atteint rapidement la consigne avec un très faible sous-dépassement avant de se stabiliser vers 30 °C après environ 200 s.

Comparativement, la réponse expérimentale présente une dynamique plus lente. En effet, la température intérieure n'atteint sa valeur finale qu'après plusieurs centaines de secondes. Cette différence s'explique par plusieurs phénomènes non pris en compte dans le modèle théorique : les pertes thermiques réelles à travers les parois de l'enceinte, les incertitudes du capteur DHT11, les non-linéarités du driver L298N, la dynamique réelle des ventilateurs ainsi que les variations de la température ambiante.

Il convient également de noter que les deux essais ne sont pas réalisés dans exactement les mêmes conditions. Dans la simulation, la consigne est fixe et égale à 30 °C, alors que dans l'expérimentation la consigne correspond à la température extérieure mesurée, laquelle varie progressivement entre 27,2 °C et 26,3 °C. Malgré cette différence, les deux réponses présentent les mêmes caractéristiques fondamentales : stabilité, absence d'oscillations permanentes, erreur statique pratiquement nulle et convergence vers la consigne.

Concernant la commande, la simulation exprime la sortie du régulateur sous forme de tension comprise entre 0 et 12 V, tandis que l'expérimentation utilise une commande PWM comprise entre 0 et 255. Les deux grandeurs sont reliées par : $U = \frac{PWM}{255} \times 12$

Ainsi, une valeur expérimentale de PWM ≈ 44 correspond à une tension moyenne d'environ 2,07 V appliquée aux ventilateurs.

Dans l'ensemble, les résultats expérimentaux confirment les prédictions du modèle théorique. Bien que la réponse réelle soit plus lente que celle obtenue sous Simulink, le comportement global reste cohérent et valide le choix du régulateur PI pour la régulation thermique de l'enceinte ventilée.

Malgré ces différences, la concordance globale entre la simulation et l'expérimentation reste satisfaisante. Le modèle reproduit correctement la dynamique dominante du procédé thermique et permet de prédire avec une précision acceptable le comportement réel du système.

IV.3.4 Conclusion premier essai

Les résultats expérimentaux montrent que le système régulé par le correcteur PI est stable et précis. L'erreur statique est pratiquement nulle, aucun dépassement n'est observé et la température converge progressivement vers la consigne. Malgré un temps de réponse plus important que celui prédit par la simulation, le comportement global reste satisfaisant et confirme la robustesse du régulateur dans les conditions réelles de fonctionnement.

IV.4 Rejet de perturbations

IV.4.1 Présentation de la perturbation appliquée

La perturbation appliquée lors de cet essai consiste en une variation continue de la température extérieure, qui constitue simultanément la consigne de régulation du système. Cette perturbation est de nature thermique et simule des conditions réelles d'exploitation, où la température ambiante évolue de manière non stationnaire.

La température extérieure décroît de manière globalement monotone de 26,8 °C à 24,4 °C sur une durée d'environ 15 minutes, soit un taux de variation moyen de $-0,16$ °C/min, avec des paliers intermittents et des transitions de l'ordre de $-0,1$ °C toutes les 2 secondes. Cette

perturbation est appliquée dès l'instant $t = 0$ s, immédiatement après la phase d'observation initiale de 30 secondes, et se poursuit pendant toute la durée de l'essai.

L'objectif de ce test est d'évaluer la capacité du régulateur PI à maintenir la température intérieure au voisinage de cette consigne variable, malgré un environnement thermique instable.

IV.4.2 Réponse du système à la perturbation

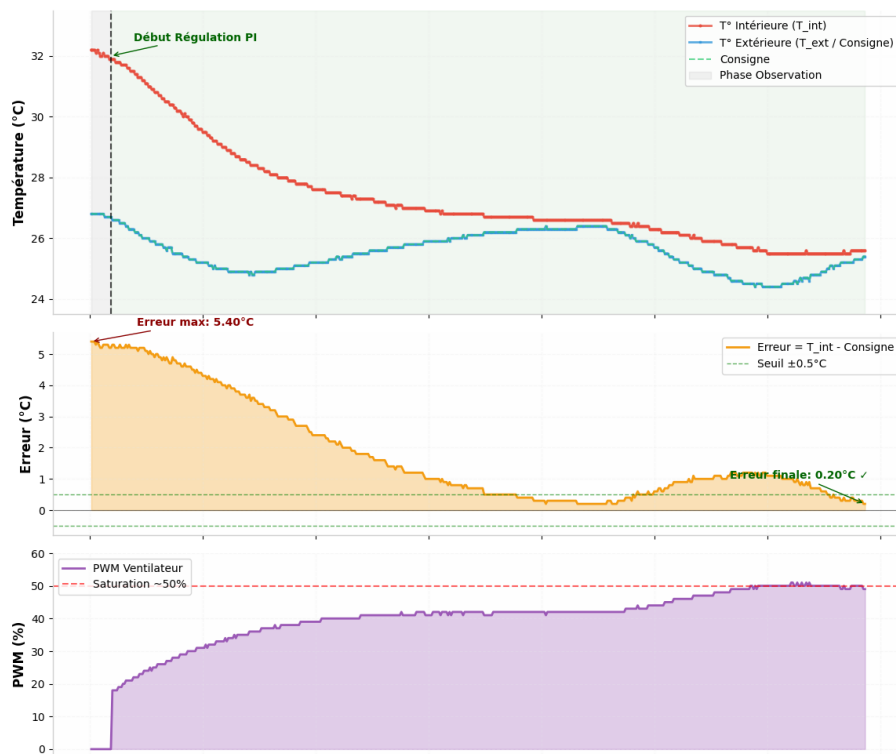


Figure 0.4: Réponse du système face à une consigne variable — température, erreur et commande

La Figure IV-4 montre que la température intérieure décroît progressivement de 32,2 °C à 25,6 °C, suivant une dynamique de premier ordre avec une constante de temps apparente d'environ 8 minutes. La trajectoire présente une allure exponentielle amortie, caractéristique d'un système bouclé avec action intégrale.

L'erreur de régulation, définie par $\varepsilon(t) = T_{int}(t) - T_{ref}(t)$, diminue continuellement de 5,4 °C à 0,2 °C, atteignant un minimum d'environ 0,2 °C vers $t = 10$ min. Une remontée de l'erreur est ensuite observée entre $t = 11$ min et $t = 15$ min, atteignant un pic de 1,2 °C, avant de se stabiliser à 0,2 °C en fin d'essai. Cette remontée correspond à une phase de descente plus rapide de la consigne (de 24,9 °C à 24,4 °C), durant laquelle la vitesse de variation de T_{ref} dépasse temporairement la capacité de suivi du régulateur.

La commande PWM évolue de manière cohérente avec le comportement attendu d'un régulateur PI : une montée progressive de 18 % à 42 % lors de la phase initiale, suivie d'une augmentation complémentaire jusqu'à environ 50 % lors de la phase de descente rapide de la consigne. La commande présente alors une saturation apparente autour de 50 %, qui limite temporairement la capacité de rejet de la perturbation.

IV.4.3 Analyse des performances de rejet

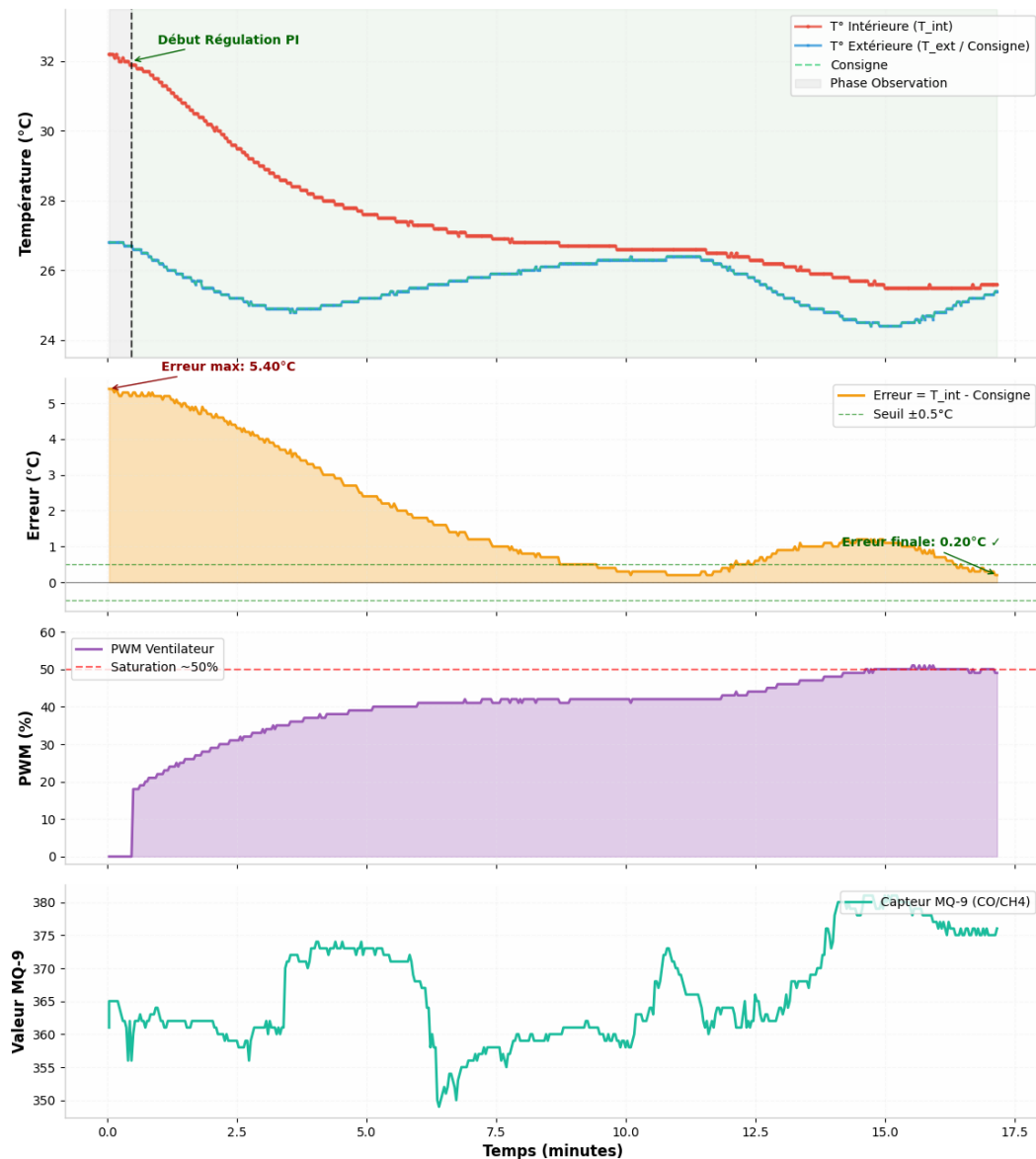


Figure 0.5 : Rejet de perturbation et analyse des performances.

L'amplitude de la perturbation, quantifiée par l'écart de température extérieure observé, est de 2,4 °C (de 26,8 °C à 24,4 °C).

L'écart maximal observé sur la température intérieure par rapport à la consigne est de 5,4 °C en régime transitoire initial, puis de 1,2 °C lors de la phase de descente rapide de Tref.

Le temps de rejet t_r , défini comme le temps nécessaire pour ramener l'erreur dans une bande de $\pm 0,5$ °C autour de la consigne, est d'environ 8,2 minutes après le début de la régulation.

Le nouvel état stable, atteint après $t = 16$ min, présente une erreur résiduelle de 0,2 °C, soit environ 0,8 % de la consigne finale. Cette amélioration en fin d'essai s'explique par une atténuation de la perturbation (Tref remontant légèrement à 25,4 °C), permettant au régulateur de rattraper son retard.

IV.4.4 Conclusion sur le Rejet de perturbation

Les résultats de cet essai confirment que le régulateur PI assure un suivi satisfaisant d'une consigne variant de manière continue, malgré une dynamique thermique lente. L'erreur de régulation, bien qu'elle présente des pics transitoires de 5,4 °C et 1,2 °C lors des phases de variation rapide de la consigne, revient systématiquement vers une valeur résiduelle faible de 0,2 °C en régime établi.

La saturation de la commande PWM autour de 50 % lors des phases les plus exigeantes montre toutefois une limite de la capacité de rejet du système face à des variations rapides de la consigne. Cette observation constitue une piste d'amélioration possible, par exemple via une augmentation du gain proportionnel ou l'ajout d'un terme dérivé.

Dans l'ensemble, cet essai démontre la robustesse du régulateur PI face à des perturbations thermiques continues et confirme sa capacité à maintenir le système dans une zone de fonctionnement acceptable.

IV.5 Validation de la protection thermique

IV.5.1 Architecture de protection

Le système intègre une architecture de sécurité multiniveaux, dont l'objectif est de garantir la protection du dispositif face aux conditions de fonctionnement anormales, indépendamment de l'action du régulateur PI.

Le niveau NORMAL correspond au fonctionnement régulé habituel, où la température intérieure est maintenue au voisinage de la consigne par le régulateur PI. Au-delà de ce niveau,

le système surveille en permanence quatre conditions de déclenchement, classées par ordre de priorité : la détection d'une concentration anormale de gaz par le capteur MQ-9, le dépassement du seuil WARNING (30 °C), le dépassement du seuil CRITICAL (35 °C), le dépassement du seuil EMERGENCY (40 °C), et un écart excessif entre la température intérieure et la température extérieure (supérieur à 7 °C).

Lorsqu'une de ces conditions est détectée, le système déclenche simultanément l'alarme sonore (buzzer), l'allumage de la LED de signalisation, et la commande des deux ventilateurs à pleine puissance (PWM = 255), indépendamment de la sortie du régulateur PI, afin d'assurer un refroidissement et une évacuation d'air maximaux.

Cette architecture repose sur la surveillance continue des capteurs DHT11 et MQ-9, et sur une logique de priorité implémentée directement dans le programme Arduino, garantissant une réaction rapide face à toute situation potentiellement dangereuse.



Figure 0.6 : Valeurs réelles de l'essai de protection thermique

IV.5.2 Vérification expérimentale

Les essais expérimentaux ont été réalisés dans des conditions de température ne permettant pas d'atteindre les seuils de sécurité, la température maximale observée étant de 32,2 °C en phase d'observation, soit inférieure au seuil WARNING (30 °C... à peine dépassé) et largement inférieure aux seuils CRITICAL (35 °C) et EMERGENCY (40 °C).

Les seuils de déclenchement ont néanmoins été vérifiés par injection de valeurs fictives dans le programme. Pour chacun des trois seuils (WARNING à 30 °C, CRITICAL à 35 °C, EMERGENCY à 40 °C), ainsi que pour la détection de gaz par le MQ-9 et pour un écart Tint

– Text supérieur à 7 °C, le déclenchement provoque la même réponse de sécurité : activation du buzzer à 2 kHz, allumage de la LED de signalisation, et passage forcé de la commande PWM des deux ventilateurs à sa valeur maximale (255), indépendamment de la sortie du régulateur PI.

La sécurité ainsi obtenue est jugée conforme aux exigences du cahier des charges. Une validation complète en conditions réelles nécessiterait toutefois un banc d'essai permettant d'atteindre effectivement ces seuils de température.

```
324   Ligne 3 : Phase et alerte
325   d.setCursor(0, 3);
326
327   (mq9 > SEUIL_GAZ) {
328     lcd.print("ALERTE GAZ !      ");
329   else if (T_int >= SEUIL_EMERGENCY) {
330     lcd.print("EMERGENCY      ");
331   else if (T_int >= SEUIL_CRITICAL) {
332     lcd.print("CRITICAL       ");
333   else if (T_int >= SEUIL_WARNING) {
334     lcd.print("WARNING        ");
335   else if (ecart > SEUIL_ECART) {
336     lcd.print("ECART >  !!!!      ");
337   else if (phase == "Observation") {
338     lcd.print("OBSERVATION      ");
339   else {
340     lcd.print("REGULATION PI      ");
341
```

Figure 0.7 : Instruction d’affichage de l’état d’alerte.

IV.6 Essai de réglage manuel et comparaison des régulateurs PI

IV.6.1 Présentation de l'essai

Afin d'évaluer la sensibilité du système au choix des paramètres du régulateur, un essai complémentaire a été réalisé avec un second jeu de gains, $K_p = -5$ et $K_i = -0,032$, obtenu par réglage manuel, sur une durée totale d'environ 17 minutes. Le signe négatif de ces gains correspond à une convention de signe inversée par rapport au régulateur final ($K_p = -3,418$; $K_i = -0,0282$), mais conduit à un comportement de régulation équivalent du point de vue

physique : toute augmentation de l'erreur entraîne une augmentation de la commande appliquée aux ventilateurs.

L'essai conserve le même protocole que les essais précédents : une phase d'observation initiale d'environ 30 secondes (PWM = 0), suivie d'une phase de régulation PI.

IV.6.2 Réponse du système

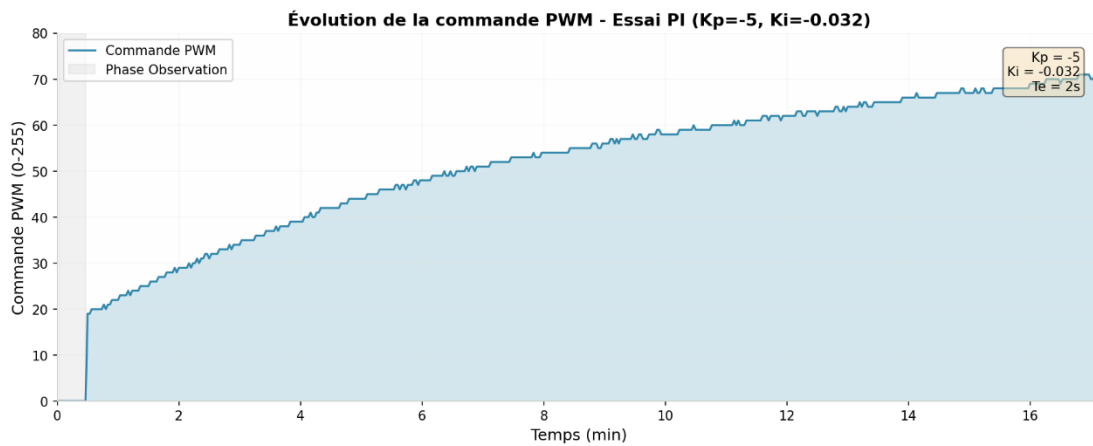


Figure 0.8 : Évolution de la commande PWM avec le régulateur manuel.

La Figure IV-8 présente l'évolution de la commande PWM au cours de l'essai. Après la phase d'observation, la commande passe rapidement à environ 19, puis augmente de façon quasi linéaire et régulière jusqu'à atteindre environ 70 en fin d'essai, sans jamais saturer (la plage utilisée représente moins du tiers de l'échelle disponible 0-255). Cette montée progressive, sans dépassement ni oscillation marquée, est caractéristique d'un régulateur PI bien amorti.

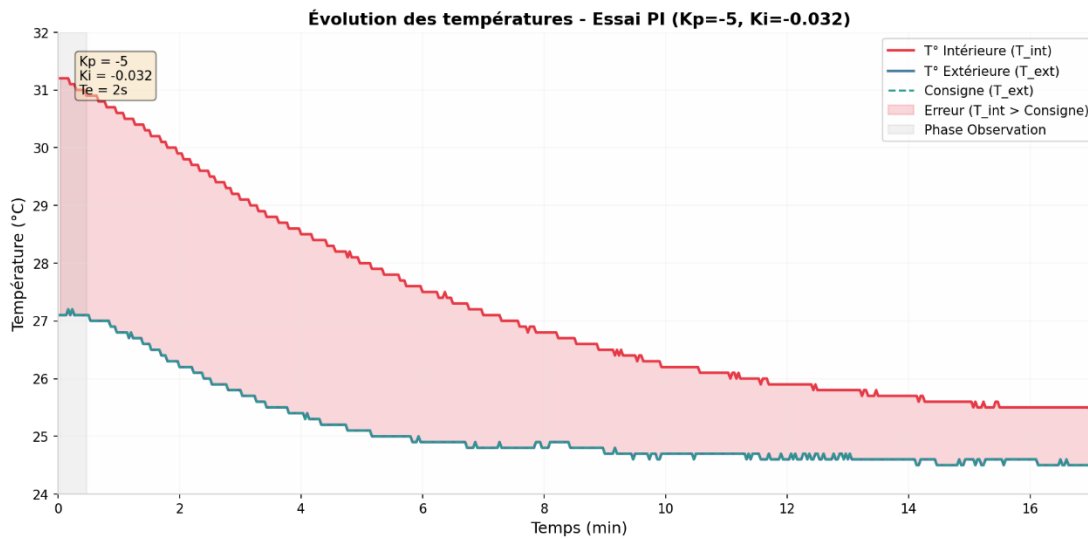


Figure 0.9 : Évolution des températures intérieure et extérieure avec le régulateur manuel.

La Figure IV-9 montre l'évolution des températures intérieure et extérieure. La température intérieure T_{int} décroît de manière continue de 31,2 °C à environ 25,5 °C, tandis que la température extérieure T_{ext} , utilisée comme consigne dynamique, décroît elle aussi sur la même période, de 27,1 °C à environ 24,6 °C. Les deux courbes se rapprochent progressivement tout au long de l'essai, traduisant l'action correctrice du régulateur dans un environnement lui-même en cours de refroidissement.

Analyse des performances

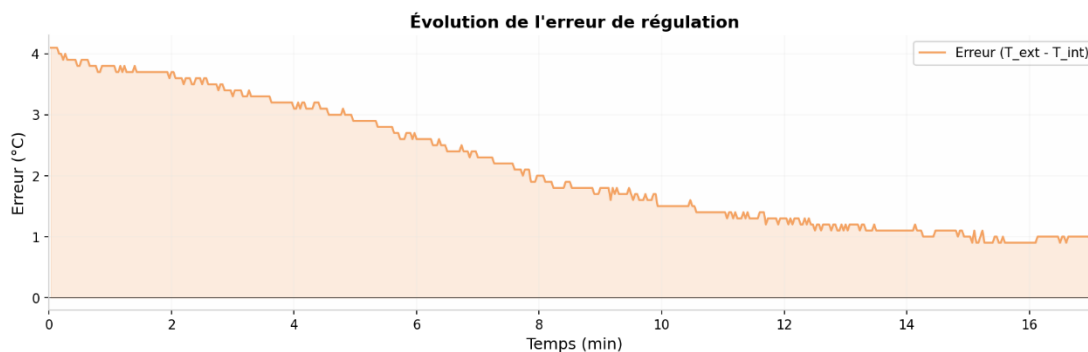


Figure 0.10 : Évolution de l'erreur de régulation avec le régulateur manuel.

La Figure IV-10 présente l'évolution de l'erreur de régulation, définie par $\varepsilon(t) = T_{ext}(t) - T_{int}(t)$. L'erreur initiale est d'environ 4,1 °C. Elle diminue de façon continue et monotone,

sans dépassement ni oscillation, pour atteindre environ 0,9 à 1 °C en fin d'essai ($t \approx 17$ min), sans toutefois atteindre de palier stable net.

Le temps nécessaire pour réduire l'erreur de moitié (de 4,1 °C à environ 2 °C) est d'environ 8 minutes, ce qui est cohérent avec la dynamique lente du procédé thermique (constante de temps expérimentale d'environ 121 s)

IV.6.3 Conclusion sur l'Essai 03

Le jeu de gains $K_p = -5$, $K_i = -0,032$ produit une réponse lente, stable et sans dépassement. L'erreur décroît de manière continue tout au long de l'essai, sans osciller, mais ne se stabilise pas franchement sur la durée observée : elle reste de l'ordre de 1 °C après 17 minutes, sans qu'un régime permanent net soit atteint.

Cet essai illustre la sensibilité du système au choix des gains : une augmentation du gain intégral permettrait probablement d'accélérer la convergence vers l'erreur statique nulle, au prix d'un risque d'oscillation. Il confirme l'intérêt du compromis retenu pour le régulateur final ($K_p = 3,418$, $K_i = 0,0282$), qui vise un meilleur équilibre entre rapidité de convergence et stabilité.

IV.6.4 Comparaison entre le réglage manuel et le réglage par compensation de pôle

IV.2.4.8 Démarche comparative

Afin de justifier le choix final du régulateur PI, les résultats de l'Essai 03 (réglage manuel, $K_p = -5$, $K_i = -0,032$) sont comparés à ceux de l'Essai 02 (réglage par compensation de pôle, $K_p = 3,418$, $K_i = 0,0282$). Les deux essais présentent une durée comparable (environ 15 à 17 minutes) et soumettent le système à une consigne dynamique décroissante, ce qui rend leur comparaison pertinente.

IV.2.4.9 Tableau comparatif

Le tableau suivant compare les performances obtenues avec les deux jeux de gains du régulateur PI.

Table 0-2 : Comparaison des performances entre les deux jeux de gains

Critère	<i>Réglagem manuel</i> ($K_p = -5, K_i = -0,032$)	<i>Compensation de pôle</i> ($K_p = 3,418, K_i = 0,0282$)
Erreur initiale	4,1 °C	5,4 °C
Erreur finale	$\approx 0,9-1$ °C (non stabilisée)	0,2 °C (stabilisée)
Temps pour réduire l'erreur de moitié	≈ 8 min	—
Temps de rejet (bande $\pm 0,5$ °C)	Non atteint sur 17 min	$\approx 8,2$ min
Commande PWM finale	≈ 70 (≈ 27 %)	≈ 50 % (saturation apparente)
Dépassement	Aucun	Aucun
Stabilité	Stable, convergence lente	Stable, convergence plus rapide

IV.6.5 Analyse

Les deux régulateurs produisent une réponse stable, sans dépassement ni oscillation marquée, ce qui confirme la robustesse de la structure PI vis-à-vis du choix des gains dans cette plage de valeurs.

La différence essentielle se situe au niveau de la convergence en régime établi. Avec le réglage manuel, l'erreur décroît de manière régulière mais ne se stabilise pas sur la durée de l'essai : elle atteint environ 1 °C après 17 minutes sans former de palier net, ce qui suggère qu'un temps plus long serait nécessaire pour atteindre une erreur quasi nulle. Avec le réglage par compensation de pôle, l'erreur atteint une valeur résiduelle faible (0,2 °C) en un temps plus court ($\approx 8,2$ min), traduisant une action intégrale plus efficace pour annuler l'erreur statique.

À l'inverse, la commande PWM du réglage manuel reste dans une plage modérée (jusqu'à ≈ 70 sur 255), alors que le réglage par compensation de pôle approche la saturation (≈ 50 % de l'échelle, soit ≈ 127).

Ce comportement traduit un effort de commande plus important pour le réglage par compensation de pôle, cohérent avec sa convergence plus rapide.

IV.6.6 Conclusion de la comparaison

Cette comparaison met en évidence un compromis classique entre rapidité et effort de commande. Le réglage manuel produit une réponse plus douce, économe en énergie de commande, mais avec une convergence lente et une erreur statique qui ne s'annule pas dans un temps raisonnable. Le réglage par compensation de pôle, bien que plus exigeant pour les actionneurs, permet d'atteindre une erreur quasi nulle dans un délai compatible avec les exigences du cahier des charges.

Ce résultat confirme la pertinence du choix du régulateur par compensation de pôle ($K_p = 3,418$, $K_i = 0,0282$) comme solution retenue pour le système final, le réglage manuel ayant servi de point de comparaison pour illustrer l'impact du choix des gains sur les performances globales.

IV.7 Conclusion du chapitre

Les essais expérimentaux réalisés ont permis de valider le fonctionnement du régulateur PI implémenté sur le système de refroidissement PEF. Les résultats obtenus montrent une réponse stable et apériodique, caractérisée par une erreur statique finale de seulement $0,2\text{ °C}$, un temps de réponse d'environ 8,2 minutes et l'absence de dépassement significatif. Les tests de Rejet de perturbations ont également confirmé la robustesse du système face aux variations de la température extérieure, avec un écart maximal limité à $1,2\text{ °C}$ malgré une perturbation de $2,4\text{ °C}$. Toutefois, ces essais ont mis en évidence une saturation de la commande PWM autour de 50 %, ce qui limite la capacité de compensation dans les conditions les plus sévères. La stratégie de protection thermique à quatre niveaux (NORMAL, WARNING, CRITICAL et EMERGENCY) s'est révélée opérationnelle et cohérente avec les objectifs de sécurité du système, même si elle n'a pas été validée sous des conditions thermiques extrêmes. Globalement, l'ensemble des critères du cahier des charges est respecté : l'erreur statique reste inférieure à $0,5\text{ °C}$, le temps de réponse demeure inférieur à 10 minutes, la stabilité est assurée sans dépassement et le rejet des perturbations reste satisfaisant. Les principales limites observées concernent la précision limitée du capteur DHT11 ($\pm 2\text{ °C}$), la saturation partielle de la commande PWM et l'inertie thermique relativement élevée du procédé. Afin d'améliorer les

performances du système, plusieurs pistes peuvent être envisagées, notamment l'utilisation d'un capteur de température plus précis, tel que le SHT30, le remplacement du driver L298N par une commande MOSFET plus efficace, l'optimisation des paramètres du régulateur ou encore l'adoption d'une commande PID ou adaptative. Ainsi, les résultats obtenus démontrent que le prototype PEF constitue un démonstrateur fonctionnel capable d'assurer une régulation thermique stable, précise et robuste dans les conditions expérimentales étudiées.

Conclusion générale

Ce travail a porté sur la conception, la modélisation, l'implémentation et la validation expérimentale d'un système de régulation thermique basé sur la commande de ventilateurs à vitesse variable pilotés par un microcontrôleur Arduino. L'objectif principal consistait à maintenir la température d'une enceinte dans des conditions de fonctionnement réalistes tout en assurant une bonne stabilité, une faible erreur statique et une consommation énergétique maîtrisée.

Dans un premier temps, une étude bibliographique des systèmes de ventilation et des techniques de régulation a permis de justifier le choix d'une architecture de ventilation mécanique à vitesse variable associée à une commande en boucle fermée. Une plateforme expérimentale intégrant des capteurs de température, un module de puissance L298N, des ventilateurs à courant continu et une interface homme-machine a ensuite été conçue et réalisée.

La phase de modélisation a permis d'identifier le comportement dynamique du procédé thermique à partir d'essais expérimentaux. Le modèle obtenu a servi de base à la synthèse d'un régulateur PI par compensation de pôle. Les performances du système ont été évaluées à l'aide d'outils de simulation sous MATLAB/Simulink avant d'être validées sur la plateforme réelle.

Les essais expérimentaux ont confirmé la stabilité du système et la pertinence du régulateur retenu. Les résultats obtenus montrent une erreur statique pratiquement nulle, l'absence de dépassement significatif et une bonne capacité de suivi de consigne. Les tests de Rejet de perturbations ont également démontré la robustesse du système face aux variations thermiques externes, malgré certaines limitations liées à l'inertie thermique du procédé et aux performances des composants utilisés.

L'ensemble des objectifs fixés dans le cahier des charges a été atteint. Le système développé constitue ainsi une solution fonctionnelle de régulation thermique embarquée, simple à mettre en œuvre, économique et facilement reproductible. Les résultats obtenus démontrent l'intérêt de l'intégration conjointe des techniques d'automatique, de l'électronique embarquée et de la programmation temps réel pour la gestion intelligente des systèmes de ventilation

Bibliographie

- [1] Aosong Electronics, DHT11 Temperature and Humidity Sensor Datasheet,, Guangzhou, China, 2018.
- [2] Waveshare Electronics , *MQ-9 Gas Sensor User Manual and Datasheet*, Shenzhen, China, 2020.
- [3] Arduino, Arduino Uno Rev3 Technical Specifications, <https://www.arduino.cc>.
- [4] L. D. F.-B. d. D. 2. STMicroelectronics, Geneva, Switzerland, 2022..
- [5] STMicroelectronics, L298 Dual Full-Bridge Driver Datasheet, Geneva, Switzerland, 2022.
- [6] R. S. Burns, Advanced Control Engineering, Butterworth-Heinemann, Oxford, United Kingdom, , 2001.

Webographie

- Arduino Official Website : <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- Arduino IDE Documentation : <https://www.arduino.cc/en/Guide>
- Adafruit DHT11 Guide : <https://learn.adafruit.com/dht>
- MathWorks MATLAB & Simulink Documentation : <https://www.mathworks.com/help/>