Titre: Title:	Commande décentralisée de la puissance de chauffage de charges thermostatiques par apprentissage automatique
Auteur: Author:	Jonathan Briand
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Briand, J. (2025). Commande décentralisée de la puissance de chauffage de charges thermostatiques par apprentissage automatique [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/65814/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/65814/
Directeurs de recherche: Advisors:	Roland P. Malhamé
Programme: Program:	Génie électrique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Commande décentralisée de la puissance de chauffage de charges thermostatiques par apprentissage automatique

JONATHAN BRIAND
Département de génie électrique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie électrique

Mai 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Commande décentralisée de la puissance de chauffage de charges thermostatiques par apprentissage automatique

présenté par **Jonathan BRIAND**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Jérôme LE NY, président
Roland MALHAMÉ, membre et directeur de recherche
Antoine LESAGE-LANDRY, membre

REMERCIEMENTS

Je voudrais remercier mon directeur de recherche, Roland Malhamé, à la fois pour ses conseils sur ma recherche et pour avoir encouragé ma curiosité dans les domaines de l'apprentissage automatique et de la commande optimale, mais aussi pour les nombreuses conversations enrichissantes que j'ai eu le privilège de partager avec lui.

Je tiens aussi à remercier les professeurs Jérôme Le Ny et Antoine Landry-Lesage pour avoir gracieusement accepté de constituer mon jury.

Finalement, je suis reconnaissant du soutien constant de ma famille et de mes amis, sans lesquels la réalisation de ce mémoire n'aurait pas été possible.

RÉSUMÉ

L'utilisation grandissante de sources d'énergie renouvelable telles que l'énergie solaire et éolienne introduit de nouveaux défis pour les réseaux de distribution électriques. En effet, contrairement aux méthodes de production d'électricité traditionnelles, ces sources d'énergie renouvelable sont intermittentes et imprévisibles. Afin de mitiger les fluctuations causées par ces sources, les réseaux de distribution électrique intelligents devront intégrer différentes solutions comme l'introduction de batteries dans le réseau ou l'utilisation de centrales de pointe. Ce mémoire s'intéresse plus spécifiquement à une méthode de maîtrise de la demande d'énergie ; c'est-à-dire une méthode qui ajuste la demande du réseau selon la production des sources renouvelables.

Les charges thermostatiques, à cause de leur association à un stockage d'énergie, peuvent servir de batteries thermiques et pourraient être utilisées à grande échelle pour aider à mitiger les fluctuations dans la génération d'électricité. Dans ce mémoire, on considère une population de charges thermostatiques dont la puissance de chauffage est contrôlée par un agrégateur. On élabore une méthode permettant à l'agrégateur de contrôler la puissance agrégée consommée par la population de charges tout en garantissant que la température de celles-ci reste à l'intérieur d'un intervalle acceptable. Selon notre méthode de contrôle, les charges thermostatiques vont calculer de façon décentralisée la puissance de chauffage qu'elles vont consommer en résolvant un problème de commande optimale. La puissance de chauffage est influencée par un terme de pression, dicté par l'agrégateur, qui modifie la fonction de coût minimisée par les charges.

Un avantage important de notre méthode est qu'elle ne nécessite pas que l'agrégateur connaisse au préalable les paramètres thermiques des charges thermostatiques de la population. À l'aide d'expérimentations et d'un algorithme d'apprentissage automatique, l'agrégateur apprend comment les charges réagissent à un signal de pression donné. L'agrégateur utilise ensuite ce savoir pour calculer la valeur du terme de pression à envoyer à la population qui amène la puissance agrégée au niveau désiré. Étant donné que l'agrégateur doit expérimenter sur la population, il est important que la méthode d'apprentissage automatique développée soit très efficace.

Finalement, à l'aide de simulations, nous testons et analysons la performance de notre méthode pour des périodes de contrôle de différentes durées.

ABSTRACT

The increasing prevalence of renewable energy sources such as solar and wind energy in electrical grids brings new challenges. These renewable sources are more unpredictable and unreliable than their non-renewable counterparts. To compensate for the fluctuations in energy production originating from these sources, smart electrical grids will need to employ a variety of different methods such as battery storage and peaking power plants, among others. This master's thesis focuses on an energy demand management method; we seek to influence the consumers' demand to balance it with the energy production from renewables.

Thermostatically controlled loads can accumulate thermal energy. Thus, they can potentially be used as thermal batteries to help compensate for fluctuations in energy production. In this work, we consider the case of a population of such loads that is controlled by an aggregator. We develop a method that lets the aggregator control the total power consumption of the population of loads in such a way that guarantees that the temperature of the loads remains in an acceptable interval. Our method entails that the heating power of the individual loads be calculated in a decentralized manner by the loads themselves, by means of solving an optimal control problem. The heating power is influenced by a pressure factor within the cost function minimized by the loads. This pressure factor is uniform for all loads and is prescribed by the aggregator.

A significant advantage of using our method is that the aggregator does not need to know the thermal parameters of the thermostatically controlled loads beforehand. Using machine learning, the aggregator learns how the loads respond to different pressure factors by interacting with the population. Using this knowledge, the aggregator is able to find the correct pressure factor to send the population to achieve any desired target power consumption. Given that the aggregator learns by interacting with the population, the efficiency of learning is a major concern.

Using the method we developed, we run simulations and analyze its performance in different scenarios.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

Abréviations :

DR	Maîtrise de la demande en énergie (Demand response)
HJB	Équation de Hamilton-Jacobi-Bellman
KAN	Réseau neuronal Kolmogorov-Arnold
LCOE	Coût actualisé de l'énergie (Levelized cost of electricity)
MARL	Apprentissage par renforcement multi-agents
MBRL	Méthode d'apprentissage par renforcement basée sur un modèle
MDP	Processus de décision markovien
MLP	Perceptron multicouche
POMDP	Processus de décision markovien partiellement observable
RL	Apprentissage par renforcement

Symboles :

x_t^i	Température de la charge i
$x_{rel,t}^i$	Température de la charge i relative à sa température idéale z_{ideal}^i
y^i	Température externe à la charge i
z_{ideal}^i	Température idéale de la charge i
z_{min}^i	Température minimale admissible de la charge i
z_{diff}^i	Valeur constante de la différence entre z_{ideal}^i et z_{min}^i
$z_{min,alt}^i$	Température minimale remplaçant z_{min}^i dans la fonction de coût J^i
c_t^i	Puissance de chauffage de la charge i, égale à $u_{ideal}^i + u_t^i$
$c_{est,t}^i$	Estimation de c_t^i obtenue avec le réseau de neurones pour la charge i
$c_{obs,t}^i$	Puissance de chauffage de la charge i observée par l'agrégateur
u_{ideal}^i	Puissance pour maintenir la charge i en régime permanent à z_{ideal}^i
u_t^i	Effort exercé par la charge i
$u_{est,t}^i$	Estimation de u_t^i obtenue avec le réseau de neurones pour la charge i
C_t	Puissance de chauffage agrégée moyenne de la population
$C_{est,t}$	Estimation de C_t obtenue avec les réseaux de neurones
$C_{obj,n}$	Objectif de puissance agrégée moyenne pour l'intervalle n
$C_{alt,n}$	Objectif alternatif de puissance agrégée moyenne réalisable pour n

C_{ideal}	Puissance agrégée moyenne quand les charges sont à z^i_{ideal} en régime permanent
C_{min}	Puissance agrégée moyenne quand les charges sont à z^i_{min} en régime permanent
q_n	Fonction de pression dictée par l'agrégateur pour influencer C_t
q_{max}	Valeur maximale de la fonction de pression
$\vec{p}_n$	Vecteur définissant la fonction de pression q_n sur l'intervalle n
$\vec{P}$	Vecteur définissant les fonctions de pression q_n sur tous les intervalles d'un épisode
r, q_{ideal}	Coefficients faisant partie de la fonction de coût J
s^i_n	État de la charge i au début de l'intervalle n
$s^i_{est,n+1}$	Estimation de l'état de la charge i au début de l'intervalle $n+1$
S_n	État de la population au début de l'intervalle n
$S_{est,n+1}$	Estimation de l'état de la population au début de l'intervalle $n+1$
$\vec{\mu}^{\,i}$	Paramètres internes du réseau de neurones pour la charge i
Δt	Durée d'un intervalle
h_n	État markovien de la population à l'intervalle n
H	Espace des états possibles de la population
a_n	Action prise par l'agrégateur à l'intervalle n
$A(h_n)$	Espace des actions possibles dans l'état h_n
$\hat{R}$	Fonction de coût apprise par les réseaux neuronaux de l'agrégateur
$\hat{G}$	Fonction de transition apprise par les réseaux neuronaux de l'agrégateur

Fonctions :

J^i	Fonction de coût minimisée par la charge i pour calculer l'effort u^i_t
f^i_{effort}	Fonction décrivant l'effort instantané de la charge i
$\vec{f}^{\,i}_{effort}$	Fonction décrivant le vecteur de l'effort de la charge i sur un intervalle
f^i_{temp}	Fonction décrivant la température de la charge i à la fin d'un intervalle
$\vec{f}^{\,i}_{neural}$	Fonction composée de $\vec{f}^{\,i}_{effort}$ et de f^i_{temp}
E_{neural}	Erreur d'estimation de la puissance consommée
$E_{est,n}$	Estimation de l'erreur entre $\vec{C}_{est,n}$ et $C_{obj,n}$ sur l'intervalle n
$E_{est,ep}$	Estimation de l'erreur entre $\vec{C}_{est}$ et $\vec{C}_{obj}$ sur l'épisode

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Les sources d'énergie renouvelable occupent une part grandissante de la production d'électricité. Bien qu'elles soient avantageuses au niveau de leurs impacts environnementaux, les sources d'énergie renouvelables ont traditionnellement toujours été beaucoup plus coûteuses que leurs alternatives non renouvelables. Cependant, avec l'avancement des technologies de fabrication modernes, certaines sources d'énergie renouvelable comme les panneaux photovoltaïques et les éoliennes sont maintenant compétitives en termes de coût avec les centrales thermiques classiques au charbon ou au gaz naturel. Le coût actualisé de l'énergie (Levelized cost of electricity - LCOE) représente le coût de production de l'électricité en prenant en considération les coûts de l'équipement, les coûts de maintenance ainsi que la durée de vie. Le LCOE permet de comparer les réalités économiques de différentes formes de génération d'électricité. Pour l'énergie éolienne, le LCOE est passé de 186 USD/MWh en 2009 à 49 USD/MWh en 2024. Plus impressionnant encore, le LCOE de l'énergie solaire photovoltaïque est passé de 496 USD/MWh en 2009 à 60 USD/MWh en 2024. Pour référence, le LCOE d'une centrale au charbon est 115 USD/MWh. En réponse à ces tendances dans le prix de production de l'énergie renouvelable, l'adoption mondiale de ces technologies augmente rapidement. En 2023, 11.8% de l'électricité générée mondialement était d'origine solaire ou éolienne [1, 2].

Cependant, la proportion grandissante de la production d'électricité provenant de panneaux photovoltaïques et d'éoliennes introduit de nouveaux défis pour les réseaux de distribution d'électricité modernes. En effet, ces techniques de production d'électricité sont dépendantes des conditions météorologiques, ce qui rend leur production d'énergie intermittente et difficile à prévoir [3]. Afin d'accommoder les fluctuations de ces sources d'énergie renouvelables, les réseaux de distribution d'électricité devront continuer à devenir de plus en plus intelligents et employer diverses méthodes pour équilibrer la demande énergétique de la population et la production.

Dans ce chapitre, nous présentons différents concepts essentiels sur lesquels repose la recherche présentée dans ce mémoire.

1.1 Maîtrise de la demande d'énergie

La maîtrise de la demande d'énergie (Demand response - DR) regroupe différentes méthodes utilisées par les réseaux électriques intelligents pour influencer la demande électrique de la

population. Les différentes techniques de DR ont historiquement été élaborées pour limiter les pointes de consommation électrique. Cependant, les techniques de DR peuvent aussi contribuer à contrebalancer les fluctuations dans la production d'électricité dues aux sources d'énergie renouvelables.

1.2 Le rôle de l'agrégateur

On définit l'agrégateur comme étant l'entité chargée de coordonner la production et la demande énergétique de la population. Dans un réseau de distribution électrique intelligent, on suppose que l'agrégateur possède certaines capacités de communication afin de coordonner intelligemment la génération ainsi que la consommation d'énergie dans le réseau. C'est l'agrégateur qui va appliquer les méthodes de DR pour influencer la demande.

1.3 Charges thermostatiques

Selon Hydro-Québec [4], 54% de l'électricité consommée par les habitations québécoises est utilisée pour réguler la température. Les systèmes de climatisation chargés de réguler la température d'habitations sont un exemple de charges thermostatiques. C'est-à-dire que ce sont des charges qui utilisent l'électricité pour chauffer ou refroidir. Une portion non négligeable de la demande électrique totale peut donc être attribuée aux charges thermostatiques.

Les charges thermostatiques peuvent emmagasiner de l'énergie thermique à travers leur température. De plus, il est généralement acceptable pour la température d'une charge thermostatique d'augmenter ou de diminuer temporairement sans nuire de façon excessive au confort de ses occupants. Cette capacité de stockage pourrait potentiellement être utilisée à grande échelle pour ajuster la demande électrique instantanée des charges thermostatiques en fonction des besoins du réseau.

1.4 Types de méthodes de contrôle de la demande

Dans les dernières années, différentes méthodes de DR utilisant la capacité de stockage énergétique d'une population de charges thermostatiques pour contrôler la demande énergétique ont été développées. Les trois sections suivantes présentent certaines caractéristiques à travers lesquelles on peut différencier ces méthodes.

1.4.1 Modélisation physique des charges ou apprentissage par renforcement

Les méthodes plus traditionnelles définissent un modèle physique représentant une population de charges thermostatiques. Ce modèle est ensuite utilisé dans un schéma d'optimisation tel que la commande optimale ou la commande prédictive. Un désavantage de cette approche est qu'il est nécessaire de connaître les paramètres thermiques des charges thermostatiques de la population pour utiliser le modèle physique dans le schéma d'optimisation choisi.

Les méthodes utilisant l'apprentissage par renforcement (RL) ne nécessitent pas de modèle physique. À partir d'expérimentations sur une population de charges thermostatiques, les méthodes de RL apprennent directement l'action optimale qui va minimiser le coût encouru par l'agrégateur.

1.4.2 Contrôle centralisé ou décentralisé

Dans les méthodes de contrôle centralisées, l'agrégateur contrôle directement la puissance de chauffage consommée par chacune des charges thermostatiques de la population. Un désavantage de cette approche est que, si le modèle utilisé par l'agrégateur n'est pas exact ou si l'agrégateur ne connaît pas exactement l'état des charges, il est possible que la puissance prescrite par l'agrégateur pousse les charges vers des températures non confortables ou non sécuritaires. Pour empêcher cette situation de se produire, les méthodes de contrôle centralisé considèrent souvent que les charges possèdent un mécanisme pouvant ignorer le contrôle de l'agrégateur si celui-ci n'est pas sécuritaire [5].

Les méthodes de contrôle décentralisées ne contrôlent pas directement la puissance de chauffage consommée par les charges thermostatiques. Plutôt, dans les méthodes décentralisées, l'agrégateur va posséder un mécanisme pour influencer la façon dont les charges thermostatiques individuelles vont calculer localement leur puissance de chauffage. Les méthodes de contrôle décentralisées ont plusieurs avantages et désavantages potentiels face aux méthodes centralisées. Contrairement aux méthodes centralisées, il n'est pas trivial de garantir l'équité de la participation des charges dans la réduction de la demande. Cependant, alors qu'une méthode centralisée doit dicter le comportement individuel de chaque charge de la population, certaines méthodes décentralisées permettent l'utilisation d'un signal de contrôle uniforme, ce qui permet de réduire la complexité de la communication nécessaire.

1.4.3 Puissance électrique binaire ou scalaire

Certaines méthodes considèrent que la puissance de chauffage consommée par les charges est binaire. C'est-à-dire, les charges consomment soit une puissance fixe, soit une puissance

nulle.

D'autres méthodes considèrent que la puissance est scalaire et peut être précisément contrôlée. La puissance varie alors entre une puissance nulle et une puissance maximale.

CHAPITRE 2 REVUE DE LITTÉRATURE

Dans le domaine de la DR, il existe un grand nombre d'approches et de méthodes différentes pour influencer la consommation électrique des consommateurs [6]. Plus récemment, avec la disponibilité de plus en plus grande d'ensembles de données décrivant la consommation électrique d'usagers, beaucoup de méthodes de DR contemporaines font utilisation de l'apprentissage automatique ou de l'apprentissage profond [7]. Dans cette revue de littérature, on va évaluer spécifiquement les méthodes de DR qui utilisent la flexibilité des charges thermostatiques.

Ce chapitre présente différentes méthodes de DR ainsi que leurs caractéristiques, avantages et désavantages. Ensuite, à partir des caractéristiques de ces méthodes, on élabore les objectifs que la nouvelle méthode développée dans ce mémoire devra atteindre.

2.1 Nature du signal de contrôle

Une caractéristique fondamentale de toute méthode de DR pour une population de charges thermostatiques est la nature du signal de contrôle envoyé par un agrégateur à la population. C'est pourquoi nous séparons les différentes méthodes présentées dans cette section selon le type de signal de contrôle utilisé.

2.1.1 Commande binaire probabiliste

Les approches de cette catégorie considèrent qu'une charge thermostatique est soit ON si la charge consomme une puissance fixe, soit OFF si elle consomme une puissance nulle. De plus, le domaine des températures possibles des charges est discrétisé en un nombre fini de sous-domaines. L'état d'une charge individuelle est la combinaison de son statut ON ou OFF et du sous-domaine de température dans lequel elle se trouve. L'état de la population de charges thermostatiques est alors représenté par une fonction de densité de probabilité représentant la proportion de la population de charges dans chaque état individuel possible. Le signal de contrôle envoyé par l'agrégateur est un vecteur contenant pour chaque sous-domaine de température la probabilité d'allumer (passer de OFF à ON) ou d'éteindre le dispositif de chauffage ou de climatisation [8–10].

Ces approches divergent dans la façon dont elles approchent l'obtention de l'état de la population ainsi que le calcul du signal de contrôle optimal à envoyer à la population pour obtenir une puissance agrégée égale à un signal de référence.

Dans [8], le problème de contrôle de la puissance agrégée est formulé comme un processus de décision markovien (MDP). Afin de pouvoir utiliser une méthode d'apprentissage tabulaire, les éléments des vecteurs contenant l'action et l'état de la population sont discrétisés. Pour chaque objectif de consommation agrégée, l'agrégateur apprend une table des valeurs du coût à venir (Q value) selon l'état de la population et l'action choisie. Pour ce faire, il simule la réponse de la population de charges en réponse à toutes les actions possibles dans tous les états possibles. L'agrégateur doit donc posséder un modèle de la dynamique thermique de la population. Une limitation de cette méthode est la nécessité de mesurer la température de chaque charge à chaque étape du MDP pour connaître l'état de la population. De plus, l'apprentissage des valeurs de Q demande un grand nombre d'expérimentations et doit être répété pour chaque nouvel objectif de puissance agrégée que l'agrégateur désire atteindre.

Dans [9], l'agrégateur possède un modèle de la dynamique thermique de la population de charges. Utilisant ce modèle, l'agrégateur calcule la proportion de la population qui doit passer de ON à OFF pour une réduction de puissance agrégée désirée et vice-versa. Selon cette proportion, l'agrégateur calcule un vecteur contenant les probabilités de passer de ON à OFF selon l'état. Un filtre de Kalman est utilisé pour obtenir une estimation de l'état à partir d'une mesure imprécise de la consommation agrégée des charges de la population, ce qui permet de limiter le transfert d'informations nécessaire. Cependant, cette approche ne considère que la réponse instantanée de la population et ne permet donc pas de planifier une trajectoire de consommation agrégée optimale pour un objectif de consommation donné sur un long horizon.

Finalement, dans [10], la dynamique thermique de la population est discrétisée puis exprimée comme une chaîne de Markov. L'agrégateur possède une trajectoire désirée de consommation agrégée sur un horizon de temps de 6 heures. Afin de trouver la trajectoire de puissance agrégée optimale qui minimise la différence carrée avec la trajectoire désirée, l'agrégateur résout un problème de minimisation dans lequel la dynamique de la population est contrainte par la chaîne de Markov. Il est démontré que ce problème de minimisation peut être exprimé sous forme convexe. Une simulation démontre que la commande optimale sur un horizon de 6 heures contenant 360 étapes discrètes peut être calculée en une minute. Un avantage de cette méthode est qu'elle permet d'obtenir la commande optimale sur un horizon de plusieurs heures. Dans le cas où la trajectoire de puissance agrégée désirée n'est pas réalisable, cette méthode permet à un agrégateur de répartir l'erreur sur tout l'horizon. Cependant, l'agrégateur a besoin de connaître les paramètres thermiques ainsi que l'état de la population pour calculer la commande optimale.

On remarque que, dans toutes ces méthodes, l'équité de la participation des charges indivi-

duelles dans la réduction de la consommation agrégée n'est pas garantie. En effet, une charge individuelle pourrait être malchanceuse et aléatoirement sélectionner l'option de couper son chauffage plus fréquemment que la moyenne des charges.

2.1.2 Commande par manipulation de la consigne thermostatique

Plutôt que de dicter la probabilité d'allumer ou d'éteindre le chauffage des charges thermostatiques selon leur température, la méthode décrite dans [11] manipule la consigne thermostatique des charges. Un modèle linéaire est développé décrivant la variation dans la consommation électrique agrégée selon la variation de la consigne thermostatique. À partir de ce modèle, il est possible de calculer la consigne thermostatique nécessaire pour obtenir une puissance agrégée résultante égale à la puissance agrégée désirée par l'agrégateur. Cette méthode est plus équitable que celles présentées dans la section précédente puisque toutes les charges diminuent leurs consignes thermostatiques. Cependant, si la population contrôlée est assez homogène, le contrôle par changement de la consigne peut induire des oscillations indésirables dans la puissance agrégée. De plus, cette approche ne considère que la réponse instantanée de la population.

2.1.3 Commande par communication d'un objectif de puissance agrégée

Dans [12], les auteurs introduisent une approche dans laquelle les charges thermostatiques calculent localement la stratégie optimale de façon à mener la puissance agrégée à un objectif de puissance désiré. Pour ce faire, le problème est décrit par un processus de décision markovien partiellement observable (POMDP) et l'approche choisie pour résoudre ce POMDP est l'apprentissage par renforcement multi-agents (MARL). À chaque étape du POMDP, chaque charge thermostatique mesure son propre état et reçoit l'information décrivant l'état de N charges voisines ainsi que l'objectif de puissance agrégée et la puissance agrégée actuelle. À partir de cette information, chaque charge utilise un réseau neuronal profond pour choisir si elle va s'allumer ou s'éteindre. Le réseau neuronal de chaque charge est entraîné de façon à minimiser un coût correspondant à l'écart entre la température de la charge et la température idéale ainsi que l'écart entre la puissance agrégée et l'objectif de puissance. Cette formulation du coût implique une certaine équité dans la participation des charges puisqu'elle pénalise les écarts de température locaux.

Cette approche est unique parmi les approches présentées dans cette revue de littérature puisque le rôle de l'agrégateur se limite à communiquer l'objectif de puissance agrégée et la puissance agrégée actuelle. Tout le calcul de coordination de la puissance agrégée est effectué implicitement de façon locale et décentralisée par les réseaux neuronaux profonds des charges.

De plus, à l'aide de simulations, les auteurs déterminent que communiquer avec 9 charges voisines dans le cas homogène ou 49 charges voisines dans le cas hétérogène permet d'obtenir la meilleure performance. Ce résultat indique que la communication requise pour calculer la commande optimale est minimale et locale.

Cette approche a été développée afin de contrebalancer les fluctuations de haute fréquence des sources d'énergies renouvelables. Cependant, elle ne permet pas de planifier des stratégies sur de longs horizons.

2.1.4 Commande par choix du prix de l'électricité

Dans [5], les auteurs développent une méthode d'apprentissage par renforcement pour que les charges thermostatiques contrôlent leur puissance de chauffage de façon décentralisée. Pour une tarification de l'électricité donnée dans le temps, l'objectif de cet article est de minimiser le coût encouru par chaque charge individuelle et non de contrôler la consommation agrégée de la population de charges. Le coût individuel correspond à la puissance multipliée par le coût instantané de l'électricité.

Cependant, plutôt que d'observer directement l'état complet des charges thermostatiques, les auteurs proposent d'observer seulement la consommation et la température extérieure des charges. Ce problème est donc un POMDP. Plutôt que de chercher à calculer une distribution de probabilité sur l'état actuel des charges, la méthode développée utilise l'ensemble des observations effectuées depuis le début de la période de contrôle pour calculer l'action à prendre. Finalement, le choix binaire entre puissance nulle et puissance fixe est effectué à l'aide d'un réseau de neurones.

Un désavantage de cette méthode est la quantité d'expérimentation requise afin d'apprendre à calculer l'action optimale. En effet, la phase d'apprentissage présentée dans l'article dure 100 jours. Une autre limitation est que cet article n'introduit pas de méthode permettant à un agrégateur de contrôler précisément la puissance agrégée d'une population de charges. Un agrégateur pourrait potentiellement influencer la puissance agrégée de la population en dictant le coût de l'électricité. Cependant, cette possibilité n'est pas discutée dans l'article. Cette méthode est discutée car son approche pour obtenir l'état des charges est pertinente pour notre méthode.

2.1.5 Commande par manipulation d'un coefficient de pression

La méthode de contrôle présentée dans ce mémoire est basée sur la méthode par équilibre de Nash inverse introduite dans [13], elle-même basée sur [14]. La méthode de Nash Inverse

nécessite de connaître tous les paramètres thermiques de chacune des charges de la population puisqu'elle se base sur un modèle de la dynamique thermique des charges plutôt que d'utiliser l'apprentissage par renforcement. De plus, à l'exception de leurs températures idéales, les charges de la population doivent être uniformes.

Dans un jeu non coopératif, un ensemble de stratégies pour chacun des joueurs est un équilibre de Nash s'il est impossible pour un joueur agissant seul d'améliorer sa situation en changeant unilatéralement de stratégie [15].

Dans le contexte de la coordination de la consommation de charges thermostatiques, la méthode par équilibre de Nash inverse suppose que les charges calculent leur puissance de chauffage consommée en résolvant un problème de commande optimale. À l'intérieur de la fonction de coût se trouve un coefficient, variable dans le temps, dont la valeur est dictée par l'agrégateur. Ce coefficient est appelé la fonction de pression et est uniforme pour toutes les charges de la population. C'est à l'aide de la fonction de pression que l'agrégateur influence la puissance de la population.

Puisque l'agrégateur connaît les paramètres thermiques et les températures initiales des charges, il calcule la trajectoire de la température moyenne correspondant à la puissance agrégée désirée. Étant donné que la puissance consommée par les charges est calculée par résolution du problème de commande optimal local, l'agrégateur calcule la fonction de pression pour laquelle la stratégie optimale pour chacune des charges va induire cette trajectoire de température moyenne. La trajectoire de température moyenne désirée est donc l'équilibre de Nash correspondant à la fonction de pression optimale.

Les désavantages principaux de cette approche sont qu'elle requiert une connaissance préalable des paramètres thermiques des habitations et une connaissance des conditions initiales moyennes de température, et de la température externe. Également, sa performance n'est pas garantie pour une population non homogène.

2.2 Objectifs de notre méthode

Dans ce mémoire, nous développons une méthode pour coordonner la puissance agrégée d'une population de charges thermostatiques qui utilise le même signal de contrôle que la méthode de Nash inverse. Cependant, nous allons incorporer de l'apprentissage automatique dans notre méthode de façon à ce qu'elle soit plus générale et polyvalente que la méthode de Nash inverse [14]. Notre méthode devra :

— Se dégager des hypothèses d'homogénéité des charges et de la connaissance de leurs paramètres physiques. Elles sont dictées dans [14] par la nécessité de calculer analy-

tiquement la trajectoire du coefficient de pression exercé dans les fonctions coûts des usagers par l'agrégateur (algorithme Nash Inverse).

— Réaliser les objectifs ci-dessus par un double calcul : un premier calcul en ligne d'une famille de réseaux neuronaux, un par charge, représentant l'impact d'une trajectoire donnée de coefficient de pression, sur la consommation de la charge. Un deuxième calcul utilisant la connaissance de la famille de réseaux neuronaux en question pour estimer par un calcul numérique la trajectoire de coefficient de pression qui permettrait à l'agrégateur d'atteindre les objectifs de consommation globaux recherchés, à condition que ces derniers soient réalisables (respect des contraintes de confort des clients).

— Dans le cas où la trajectoire de consommation globale désirée ne serait pas réalisable, développer un algorithme pour construction d'un "compromis optimal" de trajectoire réalisable qui serait alors atteignable à partir des méthodes précédentes.

— Réaliser les tâches ci-dessus, en ne requérant d'informations que les mesures de consommation individuelles des charges thermostatiques.

Plus précisément, étant donné que l'agrégateur ne connaît pas les paramètres thermiques des charges, nous faisons appel à l'apprentissage automatique pour apprendre la dynamique de celles-ci. Cependant, l'agrégateur ne possède pas de banque de données préexistante décrivant le comportement des charges lorsqu'elles réagissent à différentes fonctions de pression. De plus, il ne peut pas faire de simulations pour obtenir une telle banque de données. Donc, peu importe la méthode choisie, l'apprentissage devra se faire en ligne. C'est-à-dire que l'agrégateur va apprendre à calculer la fonction de pression optimale en expérimentant avec la vraie population de charges. À cause de cette contrainte, il est important que notre solution soit efficace et puisse apprendre à calculer la fonction de pression optimale en ayant besoin de faire le moins d'expérimentations possible.

Pour se dégager de la nécessité de mesurer les températures internes et externes des charges, nous élaborons une représentation alternative de l'état qui ne nécessite pas de connaître ces températures pour prédire le comportement de consommation de la population. Plutôt, l'agrégateur doit seulement mesurer la consommation de chaque charge pour connaître son état.

CHAPITRE 3 FORMULATION DU MODÈLE DES CHARGES

Dans ce chapitre, nous présentons l'approche que nous utilisons pour modéliser une population de charges thermostatiques dont la consommation d'électricité est influencée par un agrégateur.

En premier lieu, nous présentons le modèle simplifié de la dynamique thermique des charges que nous utilisons (3.1).

Ensuite, le mécanisme par lequel l'agrégateur exerce une influence sur la consommation des charges est introduit (3.2).

Finalement, la façon dont les charges calculent leur puissance de chauffage en réponse à un signal envoyé par l'agrégateur est décrite (3.3).

3.1 Dynamique thermique des charges

Nous cherchons à modéliser la consommation électrique d'une population P_I de I charges thermostatiques sur un intervalle de contrôle $t \in [0, T]$. Pour fixer les idées, nous supposons qu'il s'agit d'une population de charges thermostatiques chauffantes.

La dynamique thermique d'une charge thermostatique $i \in P_I$ à l'intérieur d'un intervalle de contrôle peut être modélisée par l'équation différentielle suivante [16],

$$dx_t^i = \frac{1}{V^i}\left[-U^i\left(x_t^i - y_t^i\right) + c_t^i\right]dt + \sigma^i dw_t^i \tag{3.1}$$

dans laquelle t indique le temps à l'intérieur de l'intervalle de contrôle et :

- x_t^i est la température de la charge en $°C$
- y_t^i est la température extérieure à la charge en $°C$
- c_t^i est la puissance efficace de chauffage en kW
- U^i est la conductivité thermique de la coquille de l'habitation en $kW/°C$
- V^i est la capacité thermique de la charge en $kWh/°C$
- w_t^i est un processus standard de Wiener qui représente les gains et pertes d'énergie aléatoires de la charge, comme l'ouverture de portes et fenêtres ou la chaleur dégagée par l'utilisation d'électroménagers
- σ^i est le coefficient de volatilité associé à ces évènements aléatoires

La puissance de chauffage c_t^i de chaque charge i peut varier entre 0 kW et une quantité

maximale c^i_{max}, différente pour chaque charge.

$$0 \leq c^i_t \leq c^i_{max} \tag{3.2}$$

Afin de simplifier la notation, on introduit deux nouvelles variables,

$$a^i = \frac{U^i}{V^i}, \quad b^i = \frac{1}{V^i} \tag{3.3}$$

On obtient donc la forme suivante pour la dynamique thermique d'une charge,

$$dx^i_t = \left[-a^i \left(x^i_t - y^i\right) + b^i c^i_t\right] dt + \sigma^i dw^i_t \tag{3.4}$$

On suppose que la température extérieure à chaque charge y^i_t reste constante sur la durée du contrôle, soit $y^i_t = y^i_0 \ \forall t \in [0, T]$.

3.2 Mécanisme de contrôle de la puissance

L'objectif de l'agrégateur est d'obtenir une courbe de puissance agrégée désirée pour la population P_I sur l'intervalle de contrôle $t \in [0, T]$. Plus précisément, l'agrégateur cherche à minimiser l'intégrale du carré de la différence entre la puissance agrégée moyenne instantanée de la population C_t et un profil désiré de puissance moyenne C_{obj} que l'on suppose constant sur la durée de l'intervalle.

$$C_t = \frac{1}{I} \sum_{i=0}^{I} \ c^i_t \tag{3.5}$$

$$\boldsymbol{min} \ E\left[\int_0^T \left[\ C_t - C_{obj}\ \right]^2 dt\right] \tag{3.6}$$

Le mécanisme de contrôle qu'on utilise a été élaboré dans [13]. On suppose qu'il existe un accord client-agrégateur qui stipule que les charges doivent consommer une quantité d'énergie de chauffage prescrite par la minimisation d'une fonction de coût J locale à chaque charge.

La forme de cette fonction de coût J est prescrite par l'agrégateur et est la même pour toutes les charges. Cependant, les valeurs de certains coefficients sont choisies par les charges thermostatiques elles-mêmes de façon à ce que leur comportement soit sécuritaire et adapté à leurs besoins.

Une fonction de pression q_t définie par l'agrégateur, variable dans le temps durant la période de contrôle, est envoyée à toutes les charges de la population. Cette fonction est un paramètre dans la fonction de coût locale minimisée par les charges. C'est à l'aide de cette fonction de pression q_t, uniforme pour toutes les charges, que l'agrégateur influence la puissance agrégée de la population. L'objectif de ce mémoire est de présenter une méthode pour que l'agrégateur puisse calculer la fonction q_t optimale qui minimisera la différence entre la puissance agrégée moyenne et le profil désiré.

3.2.1 Températures limites

Chaque charge thermostatique possède deux valeurs qui influencent son comportement en réponse à un signal de commande. Ces valeurs sont différentes pour chaque charge et dépendent des besoins spécifiques de la charge.

— z^i_{ideal} est la température idéale désirée pour la charge en $°C$
— z^i_{min} est la température minimale admissible pour la charge en $°C$

Par définition, la température z^i_{min} est toujours plus froide que z^i_{ideal} de $z^i_{diff} > 0$ degrés,

$$z^i_{min} = z^i_{ideal} - z^i_{diff} \tag{3.7}$$

Chaque charge est libre de changer la valeur de z^i_{ideal} à tout moment selon ses besoins. Cependant, quand l'agrégateur envoie un signal à la charge indiquant le début d'une période de contrôle, la charge doit garder ses valeurs de z^i_{ideal} et par extension z^i_{min} fixes jusqu'à la fin du contrôle.

La valeur de z^i_{diff} est toujours constante.

Dynamique thermique modifiée

On définit la quantité u^i_{ideal} comme étant la puissance de chauffage nécessaire à maintenir la température de la charge à z^i_{ideal} en régime permanent. Étant donné que u^i_{ideal} ne dépend que de la température extérieure y^i et de z^i_{ideal} et que ces valeurs sont supposées fixes sur l'intervalle de contrôle, u^i_{ideal} est constante sur la durée du contrôle,

$$u^i_{ideal} = \frac{a^i}{b^i}\left(z^i_{ideal} - y^i\right) \tag{3.8}$$

On définit ensuite l'effort de la charge u^i_t comme étant la différence entre la puissance de la charge c^i_t et la puissance idéale u^i_{ideal},

$$u_t^i = c_t^i - u_{ideal}^i \tag{3.9}$$

Les limites sur la puissance de chauffage de la charge sont transférées à la valeur de l'effort u_t^i,

$$-u_{ideal}^i \leq u_t^i \leq c_{max}^i - u_{ideal}^i \tag{3.10}$$

On remplace ces deux nouvelles variables dans l'équation (3.4) pour obtenir la forme suivante pour la dynamique thermique d'une charge,

$$dx_t^i = \left[-a^i \left(x_t^i - y^i\right) + b^i \left(u_t^i + u_{ideal}^i\right)\right] dt + \sigma^i dw_t^i \tag{3.11}$$

et, à partir de (3.8) :

$$dx_t^i = \left[-a^i \left(x_t^i - z_{ideal}^i\right) + b^i u_t^i\right] dt + \sigma^i dw_t^i \tag{3.12}$$

Dans laquelle :

- u_{ideal}^i est la puissance nécessaire pour maintenir la charge à z_{ideal}^i en kW
- u_t^i est l'effort, la différence entre la puissance consommée à l'instant t et u_{ideal}^i en kW. C'est la variable calculée à partir de la minimisation de la fonction de coût

3.2.2 Fonction de coût locale

Afin de calculer la puissance de chauffage consommée pendant un intervalle de contrôle $t \in [0, T]$, chaque charge thermostatique résout le problème de commande optimale minimisant la fonction de coût suivante,

$$J\left(x_0^i, u_t^i, q_t\right) = E\left[\int_0^T \left[\frac{q_t}{2}\left(x_t^i - z_{min}^i\right)^2 + \frac{q_{ideal}}{2}\left(x_t^i - z_{ideal}^i\right)^2 + \frac{r}{2}\left(u_t^i\right)^2\right] dt + D(x_T^i)\right] \tag{3.13}$$

Dans laquelle :

- r est un coefficient fixe qui limite l'ampleur de l'effort u_t^i
- q_{ideal} est un coefficient fixe qui pousse la température de la charge vers z_{ideal}
- q_t est une fonction du temps dictée par l'agrégateur pour la durée de l'intervalle. q_t pousse la température de la charge vers z_{min}^i et permet à l'agrégateur d'influencer la

puissance de toutes les charges

— $D(x_T^i)$ est un coût final qui est fonction de la température à la fin de l'intervalle

C'est le choix de la fonction de pression q_t dans (3.13) qui permet à l'agrégateur d'influencer la puissance de chauffage des charges thermostatiques. La fonction q_t est uniforme pour toutes les charges de la population P_I et une nouvelle fonction q_t est communiquée par l'agrégateur pour chaque intervalle de contrôle.

Afin d'alléger la présentation des équations, l'indice i sera maintenant omis et les équations feront référence à une charge unique à l'intérieur de la population P_I, sauf indication contraire.

3.3 Calcul local de l'effort optimal

Au début de l'intervalle de contrôle, la charge thermostatique reçoit la fonction de pression q_t pour l'intervalle.

L'effort optimal u_t de la charge thermostatique est calculé en résolvant l'équation de Hamilton-Jacobi-Bellman (HJB) correspondant à la dynamique décrite par l'équation (3.12) et la fonction de coût décrite par l'équation (3.13) [17]. Ce calcul est effectué localement au niveau des charges thermostatiques selon la valeur de la fonction q_t communiquée par l'agrégateur pour l'intervalle. La forme optimale de u_t, l'effort instantané à t, est la suivante,

$$u_t = -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) \tag{3.14}$$

Une fois que la valeur de l'effort optimal u_t est calculée par la charge, la charge obtient sa puissance de chauffage à l'instant t en additionnant u_t à u_{ideal}. Cependant, étant donné que la valeur de u_t calculée par la résolution de l'équation de HJB ne tient pas compte des limites sur la puissance de chauffage de la charge, en réalité, la charge va consommer une puissance égale à,

$$c_t = min(\ max(\ u_t + u_{ideal},\ 0\ ),\ c_{max}\ ) \tag{3.15}$$

Calcul de π_t et β_t

π_t et β_t sont calculés numériquement à temps inverse selon les équations de Riccati suivantes. La résolution détaillée de l'équation de Hamilton-Jacobi-Bellman se trouve à l'annexe B.

$$\frac{d\pi_t}{dt} = \frac{b^2}{r}\pi_t^2 + 2a\pi_t - q_{ideal} - q_t \tag{3.16}$$

$$\frac{d\beta_t}{dt} = \left(a + \frac{b^2}{r}\pi_t\right)\beta_t - (a\pi_t - q_{ideal})\,(z_{ideal} - z_{min}) \tag{3.17}$$

Calcul de π_T et β_T

Les valeurs de π_T et β_T sont calculées selon le coût final $D(x_T)$. Étant donné que l'objectif de l'agrégateur est d'obtenir un profil de puissance agrégée constant sur l'intervalle, il est nécessaire que la variation de l'effort dans le temps soit nulle sur tout l'intervalle. Exprimant cette contrainte à l'instant T, nous obtenons les valeurs de π_T et β_T.

$$\frac{du_T}{dt} = 0 \tag{3.18}$$

$$\frac{d\pi_T}{dt}(x_T - z_{min}) + \frac{dx_T}{dt}\pi_T + \frac{d\beta_T}{dt} = 0 \tag{3.19}$$

Étant donné que la charge calcule son effort avec l'équation 3.14, remplacer l'effort u_T dans l'équation 3.12 donne,

$$\frac{dx_T}{dt} = \left(-a - \frac{b^2}{r}\pi_T\right)x_T + az_{ideal} + \frac{b^2}{r}\,(z_{min}\pi_T - \beta_T) \tag{3.20}$$

Ensuite, remplaçant $\dfrac{dx_T}{dt}$ dans l'équation 3.19,

$$\frac{d\pi_T}{dt}(x_T - z_{min}) + \left[\left(-a - \frac{b^2}{r}\pi_T\right)x_T + az_{ideal} + \frac{b^2}{r}\,(z_{min}\pi_T - \beta_T)\right]\pi_T + \frac{d\beta_T}{dt} = 0 \tag{3.21}$$

$$\left(\frac{d\pi_T}{dt} - a\pi_T - \frac{b^2}{r}\pi_T^2\right)x_T + \left[-\frac{d\pi_T}{dt}z_{min} + az_{ideal}\pi_T + \frac{b^2}{r}\left(z_{min}\pi_T^2 - \beta_T\pi_T\right) + \frac{d\beta_T}{dt}\right] = 0 \tag{3.22}$$

Cette équation est linéaire en x_T.

$$\gamma x_T + \mu = 0 \tag{3.23}$$

Afin que cette identité soit valide pour toute valeur de température finale x_T, nous fixons les valeurs de γ et μ à 0. Nous commençons par résoudre l'équation $\gamma = 0$.

$$\gamma = \frac{b^2}{r}\pi_T^2 + 2a\pi_T - q_{ideal} - q_T - a\pi_T - \frac{b^2}{r}\pi_T^2 = 0 \tag{3.24}$$

$$\pi_T = \frac{q_T + q_{ideal}}{a} \tag{3.25}$$

Nous résolvons ensuite $\mu = 0$.

$$\mu = -\left(\frac{b^2}{r}\pi_T^2 + 2a\pi_T - q_{ideal} - q_T\right) z_{min} + az_{ideal}\pi_T + \frac{b^2}{r}\left(z_{min}\pi_T^2 - \beta_T\pi_T\right) + \left(a + \frac{b^2}{r}\pi_T\right)\beta_T - (a\pi_T - q_{ideal})\,(z_{ideal} - z_{min}) = 0 \tag{3.26}$$

$$-\left(2a\pi_T - q_{ideal} - q_T\right) z_{min} + az_{ideal}\pi_T + a\beta_T - a\pi_T\,(z_{ideal} - z_{min}) + q_{ideal}\,(z_{ideal} - z_{min}) = 0 \tag{3.27}$$

Insérant 3.25,

$$-\left(q_{ideal} + q_T\right) z_{min} + z_{ideal}(q_{ideal} + q_T) + a\beta_T - (q_{ideal} + q_T)\,(z_{ideal} - z_{min}) + q_{ideal}\,(z_{ideal} - z_{min}) = 0 \tag{3.28}$$

$$\beta_T = -\frac{q_{ideal}\,(z_{ideal} - z_{min})}{a} \tag{3.29}$$

Forme du coût final $D(x_T)$

Ces valeurs de π_T et β_T sont obtenues avec un coût final de la forme suivante :

$$D(x_T) = \frac{q_T + q_{ideal}}{a}(x_T - z_{min})^2 - \frac{q_{ideal}(z_{ideal} - z_{min})}{a}(x_T - z_{min}) \tag{3.30}$$

La démonstration se trouve à l'annexe B.

3.3.1 Justification du mécanisme de contrôle

Utiliser la fonction de pression q_t pour influencer la consommation électrique des charges plutôt que de contrôler directement la puissance de chauffage garantit que, sauf en cas de

perte d'énergie aléatoire, la température des charges va toujours se maintenir entre z_{min} et z_{ideal}. Peu importe la valeur de la fonction de pression q_t, la température des charges n'est jamais poussée à des niveaux inacceptables. Cette assurance permet à l'agrégateur d'expérimenter avec la fonction q_t librement, ce qui sera nécessaire dans le chapitre 4. La preuve de cette assertion se trouve à l'annexe A.

De plus, influencer la consommation des charges avec la fonction q_t permet de répartir équitablement la contribution des charges à la réduction de la puissance agrégée. En effet, les charges réduisent leur puissance de chauffage proportionnellement à leurs écarts individuels de température par rapport à z^i_{min}. Plus la température x^i_t est éloignée de z^i_{min}, plus une charge sera appelée à céder de la chaleur (pénalité progressive).

3.3.2 Fonction décrivant l'effort

Dans cette section, nous cherchons à déterminer les variables qui influencent la valeur de l'effort calculé,

$$u_t = -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) \tag{3.31}$$

De l'équation (3.16), nous remarquons que π_t dépend uniquement de la fonction de pression q_t.

De même, de l'équation (3.17), nous remarquons que β_t dépend uniquement de π_t et donc indirectement de q_t,

La température relative $x_{rel,t}$ est définie comme étant la différence entre x_t et z_{ideal},

$$x_{rel,t} = x_t - z_{ideal} \tag{3.32}$$

z_{diff} étant supposé constant, l'effort u_t à l'instant t ne dépend que de q_t et de la température relative $x_{rel,t}$,

$$u_t = -\frac{b}{r}\left[\ \pi(q,t)\ \left(x_{rel,t} + z_{diff}\right) + \beta(q,t)\ \right] \tag{3.33}$$

Nous cherchons maintenant une expression pour $x_{rel,t}$. Remplaçant l'effort dans la dynamique thermique de la charge (3.12), on obtient,

$$dx_t = \left(-ax_{rel,t} + b\left(-\frac{b}{r}\left(\ \pi(q,t)\ \left(x_{rel,t} + z_{diff}\right) + \beta(q,t)\ \right)\right)\right)dt + \sigma dw_t \tag{3.34}$$

On résout ensuite cette équation différentielle stochastique linéaire pour trouver la température relative $x_{rel,t}$ à t.

$$x_{rel,t} = \Phi_t \left(x_{rel,0} + \int_0^t -\Phi_s^{-1} \frac{b^2}{r} \left(\pi(q,s) z_{diff} + \beta(q,s)\right)\ ds \ + \ \int_0^t \Phi_s^{-1} \sigma\ dw_s \right) \tag{3.35}$$

dans laquelle,

$$\Phi_t = e^{\int_0^t -a - \frac{b^2}{r}\pi(q,s)\ ds} \tag{3.36}$$

On calcule ensuite l'espérance de la température relative $x_{rel,t}$,

$$E[x_{rel,t}] = \Phi_t \left(x_{rel,0} + \int_0^t -\Phi_s^{-1} \frac{b^2}{r} \left(\pi(q,s) z_{diff} + \beta(q,s)\right)\ ds \right) \tag{3.37}$$

Finalement, on définit la fonction $f_{effort}(x_{rel,0}, q, t)$ comme étant l'espérance de l'effort u_t,

$$E[u_t] = f_{effort}(x_{rel,0}, q, t) := -\frac{b}{r} \left[\ \pi(q,t)\ \left(E[x_{rel,t}] + z_{diff}\right) + \beta(q,t)\ \right] \tag{3.38}$$

Cette fonction f^i_{effort} est unique pour chaque charge i. On note que f^i_{effort} est indépendante de la température extérieure y^i et de la température idéale z_{ideal} de la charge.

Étant donné que l'agrégateur s'intéresse à une population contenant un grand nombre de charges thermostatiques dont les perturbations aléatoires sont indépendantes, un effet de moyennage statistique va rendre l'effet de la variance de u_t négligeable sur la puissance agrégée. On fait donc l'hypothèse simplificatrice suivante, nous permettant d'ignorer l'effet du bruit sur la dynamique des charges lors de l'apprentissage :

$$u_t \approx E[u_t] = f_{effort}(x_{rel,0}, q, t) \tag{3.39}$$

3.4 Charge thermostatique générique

On définit une charge thermostatique générique dont les paramètres thermiques correspondent à ceux d'une habitation typique. Cette charge générique sera utilisée pour présenter les résultats numériques des algorithmes présentés dans les prochaines sections. Les paramètres de la charge générique proviennent de l'article [13],

Charge générique :

— $U = 0.0967\ kW/°C$
— $V = 0.19\ kWh/°C$
— $z_{ideal} = 22\ °C$
— $z_{min} = 17\ °C$
— $y = 0\ °C$
— $c_{min} = 0\ kW$
— $\sigma = 0.1\ °C/\sqrt{h}$

On définit aussi les paramètres de la fonction de coût, uniformes pour toutes les charges de la population :

— $r = 1$
— $q_{ideal} = 0.2$

CHAPITRE 4 APPRENTISSAGE DE LA DYNAMIQUE DES CHARGES

4.1 Objectifs du chapitre

Bien que notre objectif ultime au chapitre 5 soit de générer une fonction de pression adéquate sur un horizon de longue durée, il s'avère qu'il est utile de décomposer cet horizon en intervalles beaucoup plus courts. Dans ce chapitre, nous développons la théorie de l'apprentissage sur un intervalle générique. Ces résultats vont servir de blocs de construction pour obtenir la fonction de pression nécessaire sur l'horizon complet. Plusieurs défis sont à relever dans cet effort :

Premièrement, l'agrégateur ne peut observer que les puissances individuelles consommées ; cette contrainte nécessite une redéfinition de l'état du système (4.5).

De plus, on doit élaborer une méthode d'apprentissage automatique efficace en terme d'expérimentations en vue de simuler la réponse par commande optimale des charges aux signaux de pression. Notre choix s'est arrêté sur l'utilisation d'un réseau de neurones individualisé par charge (4.6).

Finalement, à partir de ces réseaux de neurones, on développe une méthode pour obtenir la fonction de pression minimisant l'erreur de poursuite sur un seul intervalle (4.7). Cette méthode sera un bloc de construction que nous utiliserons dans le chapitre 5.

4.2 Définition d'un épisode

Comme mentionné à la section 2.2, on désire que notre méthode permette à l'agrégateur de changer l'objectif de puissance agrégée fréquemment en réponse aux besoins du réseau tout en permettant aussi de calculer d'avance la commande optimale sur une longue période de temps.

Pour ce faire, on définit un épisode comme étant une séquence de N petits intervalles de contrôle les uns à la suite des autres. L'épisode a une durée de T heures et chacun des intervalles dure $\Delta t = T/N$ heures. Chacun de ces intervalles peut avoir un objectif de puissance agrégée différent, ce qui permet à l'agrégateur de répondre aux besoins du réseau rapidement. On suppose que l'objectif de puissance agrégée $C_{obj,n}$ est constant sur la durée de chaque intervalle.

On considère l'objectif de l'agrégateur comme étant de réduire la somme des intégrales des différences carrées entre la puissance agrégée moyenne de la population et l'objectif de puis-

sance moyenne pour chaque intervalle.

$$\boldsymbol{min}\ E\left[\sum_{n=1}^{N}\int_{(n-1)\Delta t}^{n\Delta t}\left[\ C_t - C_{obj,n}\ \right]^2 dt\right] \tag{4.1}$$

Par exemple, si l'agrégateur connaît déjà les objectifs de puissance pour les prochaines heures, il devra pouvoir calculer d'avance la commande optimale sur l'épisode pour chacun des intervalles de cet épisode. Si, au milieu de cet épisode, l'agrégateur veut réagir à une fluctuation inattendue du réseau, il pourra changer son objectif de puissance pour le prochain intervalle. Étant donné que les intervalles sont courts, cela permet à l'agrégateur de réagir rapidement aux besoins du réseau.

4.3 Échantillonnage durant un intervalle

Pendant chaque intervalle, on suppose que l'agrégateur va mesurer la puissance consommée par les charges à K moments également répartis sur tout l'intervalle, incluant le premier et le dernier instant de l'intervalle. La figure 4.1 montre l'échantillonnage de la puissance dans le temps sur l'intervalle n.

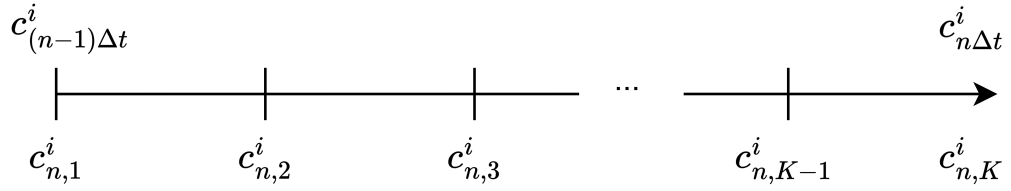

FIGURE 4.1 Schéma de l'échantillonnage sur un intervalle

L'agrégateur possède donc K mesures discrètes de la puissance de chauffage de la charge i sur l'intervalle n. On définit le vecteur $\vec{c}_n^{\,i}$ comme étant le vecteur contenant la puissance de la charge i échantillonnée à K moments sur l'intervalle n.

La notation n, k avec $k \in [1, K]$ pour l'indice sera réutilisée pour d'autres variables dont la valeur n'est pas échantillonnée par l'agrégateur. Dans ce cas, l'indice va simplement indiquer la valeur de cette variable au même instant que l'agrégateur mesure la puissance. Par exemple, la température $x^i_{n,3}$ représente la température de la charge i à l'instant où l'agrégateur fait sa troisième mesure de la puissance à l'intervalle n.

4.4 Paramétrisation de la fonction de pression

Sans forme analytique pour q_t, il est nécessaire de construire une paramétrisation pour la fonction de pression afin qu'elle soit continue sur $[0, T]$. Afin de permettre à q_t d'approximer une fonction de n'importe quelle forme et d'approcher la valeur de la fonction optimale q_t^*, on définit celle-ci comme une fonction linéaire par morceaux avec $K - 1$ segments. Ainsi, la fonction de pression à l'intervalle n, $q_{n,t}$, peut être représentée par $\vec{p}_n$, un vecteur contenant K points. On choisit délibérément un nombre de points égal au nombre de mesures de la puissance sur l'intervalle afin de simplifier le développement des équations.

$$\vec{p}_n \in \mathbb{R}^K_{\geq 0} \tag{4.2}$$

Avec $interp(\vec{p}_n, t)$ étant l'interpolation linéaire du vecteur $\vec{p}_n$ sur l'intervalle n, on définit la fonction de pression q_n comme,

$$q_n(t) := interp(\vec{p}_n, t) \tag{4.3}$$

La figure 4.2 montre un vecteur $\vec{p}_n$ possible et la fonction de pression q_n résultante.

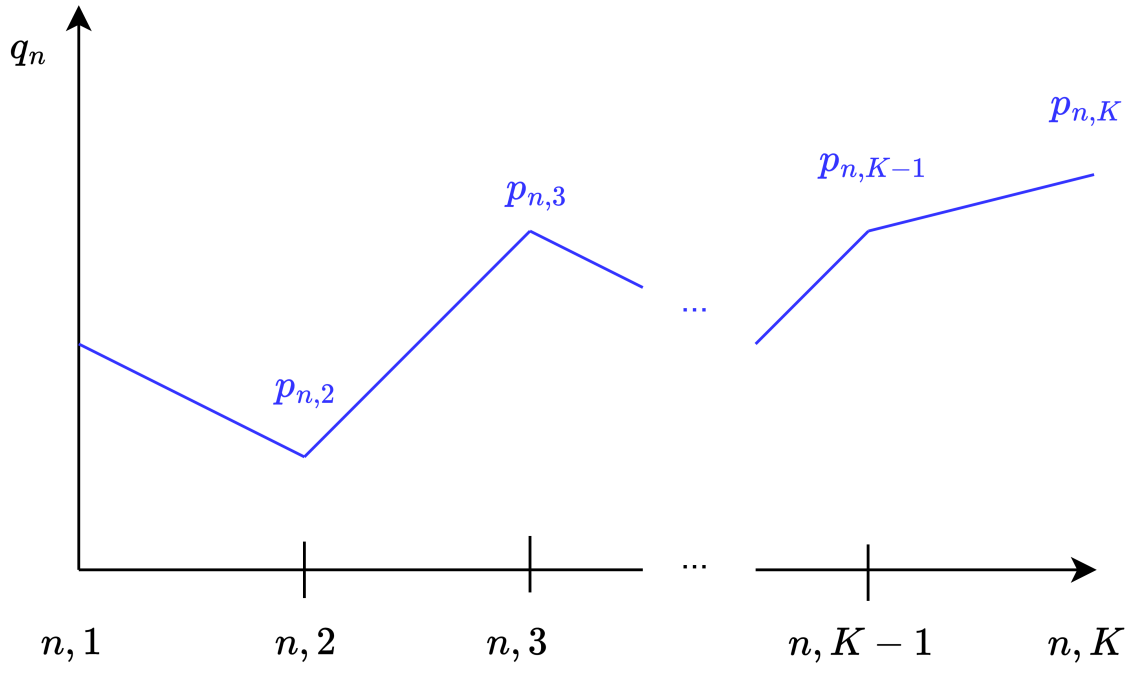

FIGURE 4.2 Exemple de fonction de pression q_n sur un intervalle

4.5 Représentation alternative de l'état

On définit l'état de la charge i comme étant l'information suffisante pour permettre de calculer la puissance consommée par cette charge sur un intervalle en réponse à une fonction de pression donnée. Pour l'intervalle n, la charge calcule localement sa puissance de chauffage en résolvant l'équation de HJB avec la fonction de pression reçue. Pour faire ce calcul, la charge a besoin de connaître tous ses paramètres thermiques et sa valeur de z^i_{ideal}. De plus, la charge a besoin de mesurer les valeurs de sa température $x^i_{n,1}$ et de la température externe y^i. Donc, l'état de la charge i au début de l'intervalle n peut être défini comme suit,

$$s^i_n = \{x^i_{n,1}\ ,\ z^i_{ideal}\ ,\ y^i\} \tag{4.4}$$

Cependant, comme mentionné dans la section 2.2, l'agrégateur ne connaît pas z^i_{ideal} et ne mesure pas les températures internes et externes $x^i_{n,1}$ et y^i des charges. On doit donc trouver une représentation alternative de l'état qui permette à l'agrégateur d'utiliser les observations dont il dispose pour prédire la puissance de chauffage de chaque charge en réponse à une fonction de pression donnée.

4.5.1 Fonction décrivant la température d'une charge

Comme démontré précédemment, chaque charge calcule son effort selon la formule suivante,

$$u_t = -\frac{b}{r}\left[\ \pi_t\ (x_{rel,t} + z_{diff}) + \beta_t\ \right] \tag{4.5}$$

De (4.5), nous déduisons que la température relative de la charge au dernier moment de l'intervalle n peut être exprimée comme ceci,

$$x_{rel,n,K} = x_{n,K} - z_{ideal} = f_{temp}\left(u_{n,K}\ ,\ q_{n,K}\right) := -\left(\frac{r}{b\ \pi_{n,K}}\ u_{n,K} + \frac{\beta_{n,K}}{\pi_{n,K}} + z_{diff}\right) \tag{4.6}$$

La fonction f_{temp} permet de calculer rétrospectivement la température relative d'une charge thermostatique au dernier moment d'un intervalle si on connaît son effort et la valeur de la fonction de pression à cet instant. La valeur de la température relative finale $x_{rel,n,K}$ d'un intervalle est aussi la température relative initiale $x_{rel,n+1,1}$ à l'intervalle suivant,

$$x_{rel,n+1,1} = f_{temp}\left(u_{n,K}\ ,\ q_{n,K}\right) \tag{4.7}$$

Étant donné que $q_{n,K} = p_{n,K}$

$$x_{rel,n+1,1} = f_{temp}\left(u_{n,K}\ ,\ p_{n,K}\right) \tag{4.8}$$

4.5.2 Nouvelle formulation de l'état

La puissance chauffante consommée par chaque charge durant l'intervalle est égale à l'effort plus la puissance en régime permanent à température idéale,

$$c_t^i = min(\ max(\ u_t^i + u_{ideal}^i,\ 0\ ),\ c_{max}\ ) \tag{4.9}$$

Si on s'intéresse à la puissance échantillonnée par l'agrégateur sur l'intervalle n, on obtient l'équation suivante,

$$\vec{c_n}^{\,i} = \vec{min}(\ \vec{max}(\ \vec{u_n}^{\,i} + u_{ideal}^i\vec{1},\ \vec{0}\ ),\ c_{max}\vec{1}\ ) \tag{4.10}$$

Dans laquelle :

- $\vec{c_n}^{\,i}$ est le vecteur contenant la puissance de la charge i échantillonnée à K instants
- $\vec{u_n}^{\,i}$ est le vecteur contenant l'effort calculé par la charge i à K instants
- $\vec{1}$ est un vecteur contenant K composantes égales à 1
- $\vec{0}$ est un vecteur contenant K composantes égales à 0
- Les fonction $\vec{min}$ et $\vec{max}$ retournent un vecteur contenant le minimum ou le maximum de deux vecteurs calculé composante par composante.
  $\vec{min}(\vec{a}, \vec{b}) := (\ min(a_k,\ b_k)\ \forall k \in [1, K]\ )$

On définit la fonction $\vec{f}_{effort}^{\,i}$ comme étant le vecteur de composantes $f_{effort,n,K}^i$ évaluées aux K instants de l'échantillonnage,

$$\vec{u_n}^{\,i} \approx \vec{f}_{effort}^{\,i}(x_{rel,n,1}^i\ ,\ \vec{p}_n) := \left(f_{effort}^i(x_{rel,n,1}^i\ ,\ interp(\vec{p}_n)\ ,\ k)\ \forall k \in [1, K]\ \right) \tag{4.11}$$

Remplaçant l'effort $\vec{u_n}^{\,i}$ par la fonction vectorielle $\vec{f}_{effort}^{\,i}$, on obtient,

$$\vec{c_n}^{\,i} \approx \vec{min}(\ \vec{max}(\ \vec{f}_{effort}^{\,i}(x_{rel,n,1}^i\ ,\ \vec{p}_n) + u_{ideal}^i\vec{1},\ \vec{0}\ ),\ c_{max}\vec{1}\ ) \tag{4.12}$$

Dans cette équation, on remplace $x_{rel,n,1}^i$ pour obtenir,

$$\vec{c}_n^{\,i} \approx \vec{min}(\ m\vec{a}x(\ \vec{f}^{\,i}_{effort}(f^i_{temp}(u^i_{n-1,\ K},\ p_{n-1,\ K})\ ,\ \vec{p}_n) + u^i_{ideal}\vec{1},\ \vec{0}\ ),\ c_{max}\vec{1}\ ) \tag{4.13}$$

On observe donc qu'il est possible de calculer la puissance de chauffage d'une charge thermostatique en réponse à une paramétrisation $\vec{p}_n$ sans connaître $x^i_{n,\ 1}$, y^i ou z^i_{ideal}. Ensemble, u^i_{ideal}, $u^i_{n-1,\ K}$ et $p_{n-1,\ K}$ constituent l'état s^i_n de la charge thermostatique au début de l'intervalle de contrôle n,

$$s^i_n = \{u^i_{ideal}\ ,\ u^i_{n-1,\ K}\ ,\ p_{n-1,\ K}\} \tag{4.14}$$

Pour obtenir l'état de la population entière, on regroupe simplement les valeurs de u^i_{ideal} et $u^i_{n-1,\ K}$ des charges de la population dans les vecteurs $\vec{u}_{ideal}$ et $\vec{u}_{n-1,\ K}$. On définit donc l'état de la population comme,

$$S_n = \{\vec{u}_{ideal}\ ,\ \vec{u}_{n-1,\ K}\ ,\ p_{n-1,\ K}\} \tag{4.15}$$

Cependant, la quantité mesurée par l'agrégateur est la puissance $\vec{c}_n^{\,i}$ et non l'effort $\vec{u}_n^{\,i}$. Les deux sections suivantes décrivent comment obtenir l'effort et donc l'état à partir de mesures de la puissance.

4.5.3 État au premier intervalle de l'épisode

On cherche à connaître l'état de la charge i au début de l'épisode. Avant le début de l'épisode, l'agrégateur ne demandait aucune réduction de puissance à la population. On suppose donc que les charges sont en régime permanent à leurs températures idéales, ce qui veut dire que leur puissance de chauffage au début du premier intervalle de l'épisode $c^i_{1,1}$ va être exactement égale à leur puissance en régime permanent à température idéale u^i_{ideal}. De plus, on déduit que l'effort au début du premier intervalle est nul. La notation $0, K$ indique l'instant immédiatement avant le début de l'épisode.

$$u^i_{ideal} = c^i_{0,K} \tag{4.16}$$

$$u^i_{0,K} = 0 \tag{4.17}$$

Les charges calculent leur effort comme si elles avaient reçu une fonction de pression constante égale à 0. L'agrégateur connaît donc la valeur de $p_{0,K} = 0$.

L'agrégateur connaît alors l'état de la charge i au début du premier intervalle de l'épisode,

$$s_1^i = \{u_{ideal}\ ,\ u_{0,K}\ ,\ p_{0,K}\} = \{c_{0,K}\ ,\ 0\ ,\ 0\} \tag{4.18}$$

Connaissant l'état de chaque charge parmi la population, l'agrégateur connaît l'état de la population,

$$S_1 = \{\vec{u}_{ideal}\ ,\ \vec{u}_{0,K}\ ,\ p_{0,K}\} = \{\vec{c}_{0,K}\ ,\ \vec{0}\ ,\ 0\} \tag{4.19}$$

On remarque que cette représentation de l'état permet aux charges de varier leur température z_{ideal} en dehors des épisodes de contrôle. Cela permet aux charges de maintenir une température confortable sans avoir besoin d'informer l'agrégateur, puisque l'agrégateur va construire une représentation de l'état des charges en mesurant leurs puissances de chauffage au début de l'épisode.

4.5.4 Calcul de l'état durant l'épisode

On cherche à connaître l'état de la charge i au début de l'intervalle n. L'agrégateur connaît la valeur de $p_{n-1,K}$ car c'est simplement la valeur finale de la fonction de pression envoyée par l'agrégateur lors de l'intervalle précédent, l'intervalle $n-1$. L'agrégateur connaît aussi la valeur de u^i_{ideal} car on suppose que la température extérieure y^i ne change pas durant l'épisode et l'agrégateur a mesuré u^i_{ideal} au début de l'épisode.

Comme mentionné précédemment, l'agrégateur mesure directement la puissance $\vec{c}^{\,i}_{n-1}$ de la charge. Cependant, l'état s^i_n de la charge contient l'effort à l'instant final de l'intervalle précédent $u^i_{n-1,K}$. Il faut donc trouver comment obtenir l'effort $u^i_{n-1,K}$ à partir de la puissance $c^i_{n-1,K}$.

La puissance de chauffage des charges est limitée entre $[0, c^i_{max}]$,

$$c^i_{n-1,K} = min(\ max(\ u^i_{n-1,K} + u^i_{ideal},\ 0\ ),\ c^i_{max}\ ) \tag{4.20}$$

Étant donné que l'agrégateur connaît la valeur de u^i_{ideal}, si la puissance mesurée se trouve entre 0 et c_{max}, il est possible de calculer directement l'effort par,

$$u^i_{n-1,K} = c^i_{n-1,K} - u^i_{ideal} \quad \forall c^i_{n-1,K} \in (0, c^i_{max}) \tag{4.21}$$

Cependant, si l'agrégateur mesure une puissance égale à 0 ou c^i_{max}, il est impossible de

connaître la valeur de l'effort calculé par la charge. Dans une telle situation, il n'est pas possible pour l'agrégateur de connaître l'état de la charge au début de l'intervalle suivant à partir de (4.21).

On développera une méthode pour obtenir une approximation de l'état de la charge dans cette situation dans la section 4.8. Pour l'instant, on suppose simplement que toutes les puissances de chauffage appartiennent à $(0, c^i_{max})$.

4.6 Apprentissage automatique avec réseau de neurones

L'agrégateur a besoin de pouvoir approximer la puissance de chauffage de la charge i en réponse à une fonction de pression donnée.

Pour ce faire, on choisit d'entraîner un perceptron multicouche (MLP). Un MLP est un réseau de neurones dense dans lequel tous les neurones d'une couche sont connectés à tous les neurones de la couche suivante. Un MLP contient une couche d'entrée, une couche de sortie et au moins une couche cachée au milieu. Selon le théorème d'approximation universelle [18], un MLP avec une seule couche cachée peut apprendre à approximer n'importe quelle fonction avec une précision arbitraire si le nombre de neurones dans la couche cachée est suffisamment grand. On choisit d'utiliser un MLP avec 2 couches cachées de 256 neurones et la fonction d'activation tangente hyperbolique [19].

On se rappellera (voir 4.13) que notre nouvelle représentation de l'état ainsi que la paramétrisation de la fonction de pression nous permettent d'exprimer la puissance de la charge i comme suit,

$$\vec{c_n}^{\,i} \approx \vec{min}(\ \vec{max}(\ \vec{f}^{\,i}_{effort}(f^i_{temp}(u^i_{n-1,\ K},\ p_{n-1,\ K})\ ,\ \vec{p}_n) + u^i_{ideal}\vec{1},\ \vec{0}\ ),\ c^i_{max}\vec{1}\ ) \tag{4.22}$$

Cependant, plutôt que d'approximer directement la puissance $\vec{c_n}^{\,i}$ avec le réseau de neurones, on choisit d'approximer la fonction composée suivante,

$$\vec{f}^{\,i}_{neural} := \vec{f}^{\,i}_{effort} \circ f^i_{temp} \tag{4.23}$$

Ce choix est expliqué en détail à la section 4.6.5.

Le réseau de neurones pourra donc calculer une estimation de l'effort calculé par la charge i en réponse à une paramétrisation $\vec{p}_n$,

$$\vec{u}^{\,i}_{est,n} = \vec{f}^{\,i}_{neural}(u^i_{n-1,K}\ ,\ p_{n-1,K}\ ,\ \vec{p}_n\ ,\ \vec{\mu}^{\,i}) \tag{4.24}$$

Où :

— $\vec{u}^{\,i}_{est,n}$ est un vecteur calculé par le réseau neuronal approximant $\vec{f}^{\,i}_{effort} \circ f^i_{temp}$
— $\vec{\mu}^{\,i}$ est le vecteur contenant l'ensemble des paramètres internes du réseau de neurones

Une fois que l'agrégateur obtient l'estimation de l'effort de la charge avec le réseau de neurones, il peut y additionner u^i_{ideal} et appliquer les limites sur la puissance pour obtenir une estimation de la puissance de la charge.

$$\vec{c}^{\,i}_{est,n} = \vec{min}(\ \vec{max}(\ \vec{f}^{\,i}_{neural}(u^i_{n-1,K}\ ,\ p_{n-1,K}\ ,\ \vec{p}_n\ ,\ \vec{\mu}^{\,i}) + u^i_{ideal}\vec{1},\ \vec{0}\ ),\ c^i_{max}\vec{1}\ ) \tag{4.25}$$

On suppose que l'agrégateur connaît la valeur de c^i_{max} pour chaque charge de la population puisque cette valeur est constante et peut être facilement mesurée si l'agrégateur demande simplement à toutes les charges de consommer leur puissance de chauffage maximale.

4.6.1 Modèle de la population

On vient de décrire la méthode pour apprendre à approximer la puissance de chauffage d'une seule charge parmi la population. Pour estimer la puissance agrégée de la population comprenant I éléments, l'agrégateur va devoir entraîner un réseau de neurones pour chaque charge. On note qu'il serait aussi possible de regrouper les charges de la population en groupes possédant des paramètres thermiques similaires et de n'utiliser qu'un seul réseau de neurones par groupe.

Pour obtenir une estimation de la puissance agrégée moyenne en réponse à une paramétrisation $\vec{p}_n$ dans un état particulier de la population, l'agrégateur additionne simplement ses estimations de la puissance de chacune des charges de la population.

$$\vec{C}_{est,n} = \frac{1}{I}\sum_{i=1}^{I} \vec{c}^{\,i}_{est,n} \tag{4.26}$$

4.6.2 Domaine de la fonction de pression

Étant donné qu'on cherche à réduire la puissance agrégée de la population, la fonction de pression sera toujours non négative. Cependant, il n'existe à priori pas de limite sur la valeur maximale de la fonction de pression.

Idéalement, on voudrait fixer un domaine fini pour la fonction de pression. De cette façon, les réseaux de neurones de l'agrégateur pourraient apprendre le comportement des charges pour toute valeur de $q_t \in [0, q_{max}]$.

Cependant, la preuve à l'annexe A indique qu'une charge va maintenir une température constante égale à sa température minimale si elle reçoit une fonction de pression de valeur infinie. Aucune valeur finie de la fonction de pression ne pourra pousser la charge à atteindre sa température minimale. Si une charge ne peut pas atteindre sa température minimale, il n'est pas possible d'utiliser toute l'énergie thermique disponible et notre méthode n'est pas optimale. Dans la sous-section suivante, nous illustrons numériquement le problème.

Visualisation de la variation de température sur un exemple numérique

Afin de visualiser ce résultat, on utilise la charge générique dont les paramètres thermiques ont été définis à la section 3.4. On dessine le graphique décrivant le taux de variation de la température de la charge selon sa température et la valeur de la fonction de pression. Pour ce faire, on suppose une fonction de pression dont la valeur est constante, ce qui permet de calculer les valeurs de β_t et π_t en régime permanent. Connaissant les valeurs de β_t et π_t, on calcule la puissance de chauffage comme suit,

$$c_t = min(\ max(\ -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) + u_{ideal},\ 0\ ),\ c_{max}\ ) \tag{4.27}$$

Finalement, on calcule le taux de variation de la température de la charge comme suit,

$$\frac{dx_t}{dt} = \left[-a\left(x_t - y\right) + b\left(c_t\right)\right] \tag{4.28}$$

Si on limite arbitrairement la valeur de la fonction de pression entre $[0, 1]$, on obtient la figure 4.3. Dans cette figure, la couleur correspond au taux de variation instantané de la température d'une charge étant à la température x_t et ayant reçu la fonction de pression constante q_t. Rouge indique une augmentation de température et bleu une diminution. La courbe noire représente un taux de variation de la température dans le temps nul. C'est-à-dire, la courbe noire représente la valeur de la fonction de pression constante pour maintenir la température en abscisse en régime permanent. On observe clairement qu'une fonction de pression $q_t = 1\ \forall t \in [0, \Delta t]$ n'est pas suffisante pour que la charge atteigne éventuellement sa température minimale $z_{min} = 17\ °C$.

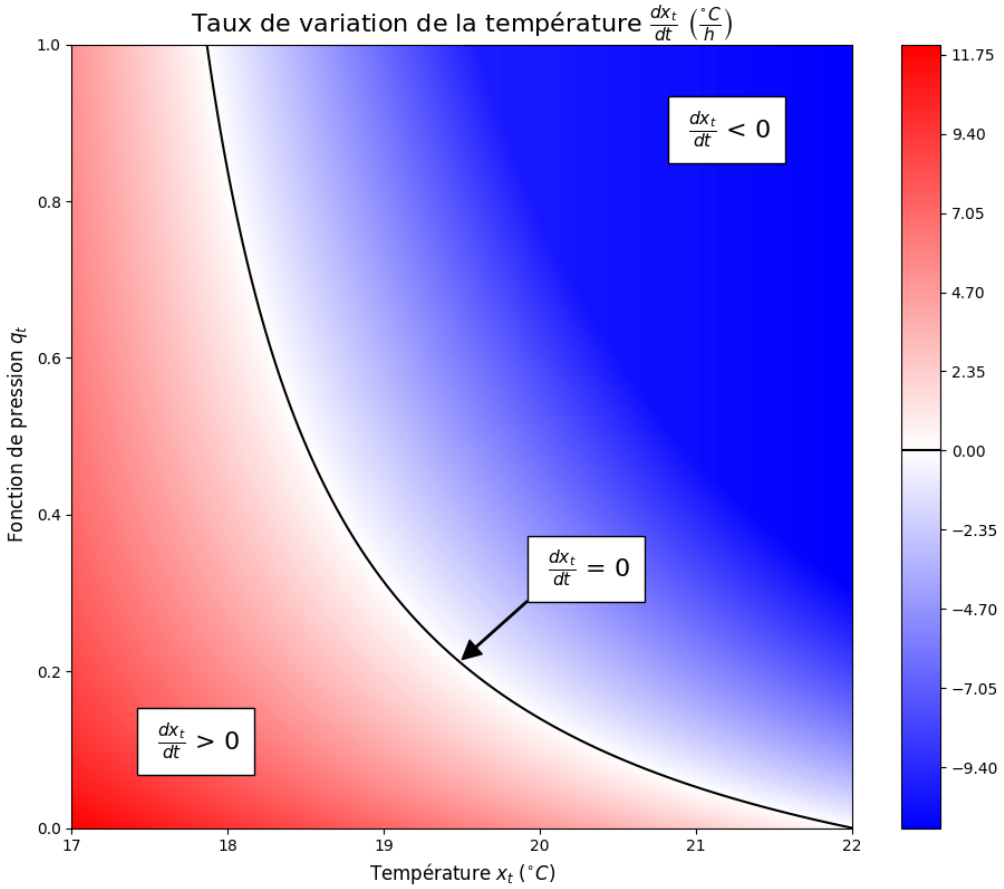

FIGURE 4.3 Taux de variation de la température $\frac{dx_t}{dt}$

Température minimale alternative

Dans cette section, on développe une méthode qui permet d'amener la température d'une charge à sa température minimale z_{min} en un temps fini et avec une fonction de pression de valeur finie.

On considère une fonction de pression dont les valeurs se situent entre $[0, q_{max}]$. Lors de la résolution de l'équation de HJB pour le calcul de l'effort optimal, plutôt que d'utiliser la vraie valeur de z_{min}, la charge va utiliser une valeur alternative $z_{min,alt}$. La charge va donc minimiser la fonction de coût suivante,

$$J\left(x_0, u, q(t)\right) = E\left[\int_0^T \left(\frac{q_t}{2}\left(x_t - z_{min,alt}\right)^2 + \frac{q_{ideal}}{2}\left(x_t - z_{ideal}\right)^2 + \frac{r}{2}\left(u_t\right)^2\right) dt + D(x_T^i)\right] \tag{4.29}$$

On définit $z_{min,alt}$ comme étant la valeur qui va maintenir la température de la charge en

régime permanent quand la charge est à sa température minimale $x_t = z_{min}$ et que la fonction de pression est constante et égale à sa valeur maximale $q_t = q_{max}$.

$$\left(E\left[\frac{dx_t}{dt}\right] \;\middle|\; (x_t, q_t) = (z_{min}, q_{max}) \right) = 0 = -a(\ z_{min} - z_{ideal}\ ) + b(\ u_t\ ) \tag{4.30}$$

Si on remplace l'effort u_t (voir 3.14) dans cette équation, en choisissant cette fois-ci dans la fonction de coût $z_{min,alt}$ à la place de z_{min}, on a :

$$0 = -a(\ z_{min} - z_{ideal}\ ) + b\left(-\frac{b}{r}(\ \pi_t(\ z_{min} - z_{min,alt}\ ) + \beta_t\ ) \right) \tag{4.31}$$

En remplaçant β_t, en régime permanent (voir 3.29) il vient :

$$0 = a(z_{ideal} - z_{min}) + b\left(-\frac{b}{r}\left(\pi_t(z_{min} - z_{min,alt}) + \frac{z_{ideal}(a\pi_t - q_{ideal}) - z_{min,alt}(a\pi_t - q_{ideal})}{a + \pi_t \frac{b^2}{r}} \right) \right) \tag{4.32}$$

Finalement, on isole la valeur de $z_{min,alt}$ pour obtenir,

$$z_{min,alt} = \frac{\frac{a}{b}(z_{min} - z_{ideal}) + \frac{b}{r}\pi_t z_{min} + \frac{b}{r}\frac{z_{ideal}(a\pi_t - q_{ideal})}{a + \pi_t \frac{b^2}{r}}}{\frac{b}{r}(\pi_t + \frac{a\pi_t - q_{ideal}}{a + \pi_t \frac{b^2}{r}})} \tag{4.33}$$

dans laquelle on utilise la valeur de π_t en régime permanent (voir 3.25) :

$$\pi_t = \frac{r}{b^2}\left[-a + \sqrt{a^2 + b^2 \frac{q_{ideal} + q_{max}}{r}} \right] \tag{4.34}$$

L'agrégateur peut choisir une valeur maximale pour la fonction de pression q_{max} et la communiquer aux charges de la population. Cette valeur de q_{max} est constante. Ensuite, utilisant la valeur de q_{max} et l'équation (4.33), chaque charge thermostatique de la population va calculer sa température minimale alternative. Cette température minimale alternative va remplacer la vraie température minimale z_{min} dans la fonction de coût minimisée par la charge thermostatique.

Utilisant une valeur de $q_{max} = 0.2$ et la température minimale alternative résultante avec la charge générique, on obtient la figure 4.4. On observe qu'utiliser la température minimale alternative $z_{min,alt}$ dans la fonction de coût permet de pousser la température de la charge vers z_{min} quand la fonction de pression $q_t = q_{max}$.

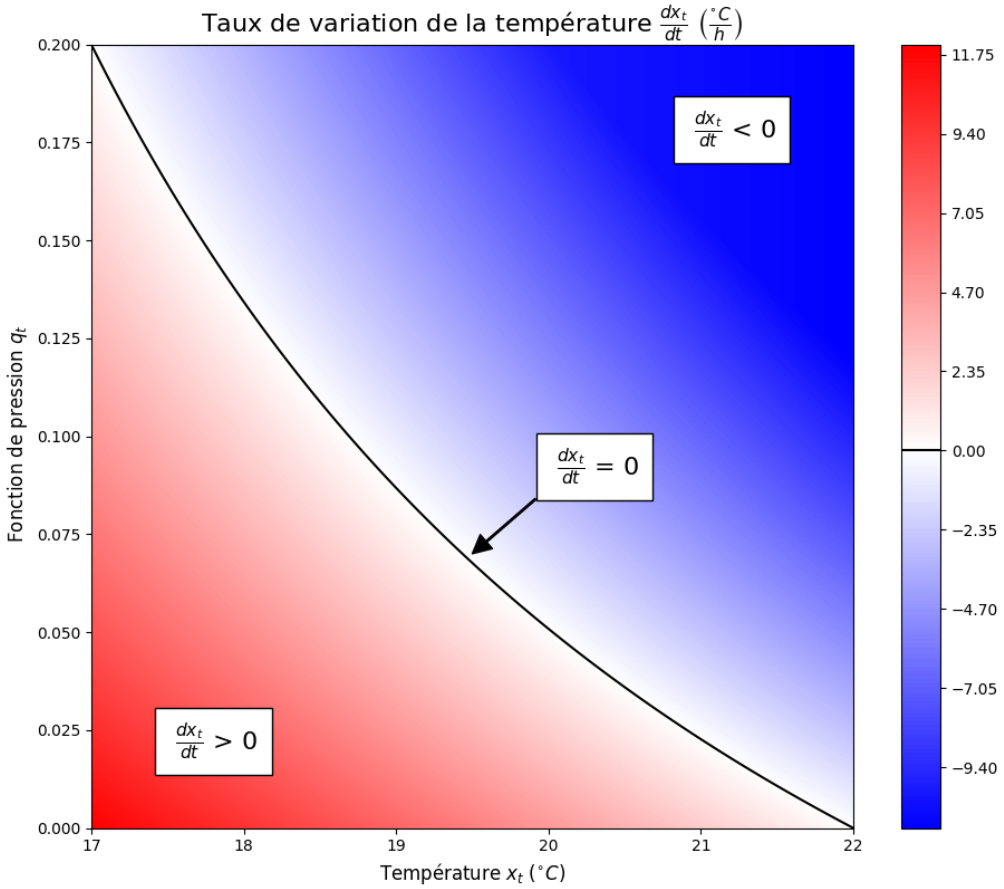

FIGURE 4.4 Taux de variation de la température $\frac{dx_t}{dt}$ en utilisant $z_{min,alt}$

4.6.3 Apprentissage par rétropropagation du gradient

Quand notre méthode est mise en place pour la première fois, les réseaux de neurones de l'agrégateur sont initialisés avec des paramètres internes aléatoires. Donc, initialement, les réseaux de neurones ne sont pas capables d'estimer la consommation des charges.

Pour entraîner les réseaux de neurones, l'agrégateur effectue une phase d'apprentissage. Durant cette phase d'apprentissage, l'agrégateur envoie une fonction de pression à la charge et échantillonne la puissance résultante à K moments durant l'intervalle n d'un épisode générique. On nomme le vecteur contenant les valeurs de puissance observées $\vec{c}^{\,i}_{obs,n}$. Ensuite, l'agrégateur va reconstruire l'effort calculé par la charge à partir de la puissance observée. Cependant, tel que mentionné à la section 4.5.4, si $\vec{c}^{\,i}_{obs,n}$ contient au moins une valeur égale à 0 ou c^i_{max}, l'agrégateur ne peut pas déterminer la valeur de cet effort puisque la puissance de la charge a été limitée.

$$\vec{u}^{\,i}_{obs,n} = \vec{c}^{\,i}_{obs,n} - u^i_{ideal}\vec{1} \quad , \quad c^i_{obs,n,k} \in (0, c^i_{max}) \; \forall k \in [1, K] \tag{4.35}$$

Si l'agrégateur ne peut pas déterminer $\vec{u}^{\,i}_{obs,n}$, il n'est pas possible d'entraîner le réseau de neurones avec les observations faites durant cet intervalle. L'effort $\vec{u}_n^{\,i}$ correspondant à cette paramétrisation pourra possiblement être mesuré à un autre moment si, par exemple, la température extérieure est plus froide, ce qui augmente la valeur de u_{ideal} et permet à la charge d'exercer un effort plus négatif sans atteindre une puissance de chauffage de 0.

Si l'agrégateur peut calculer $\vec{u}^{\,i}_{obs,n}$, il va ensuite obtenir une estimation de l'effort avec le réseau de neurones associé à la charge.

$$\vec{u}^{\,i}_{est,n} = \vec{f}^{\,i}_{neural}(u^i_{n-1,K} \; , \; p_{n-1,K} \; , \; \vec{p}_n \; , \; \vec{\mu}^{\,i}) \tag{4.36}$$

L'estimation de l'effort et la valeur observée de l'effort sont ensuite comparées. L'erreur du réseau neuronal est la somme des différences carrées entre chacune des K valeurs discrètes des vecteurs contenant l'effort observé et l'effort estimé,

$$E_{neural,n}(\vec{\mu}^{\,i}) := \sum_{k=1}^{K} (u^{\,i}_{est,n,k} - u^{\,i}_{obs,n,k})^2 \tag{4.37}$$

Étant donné que $\vec{u}^{\,i}_{est,n}$ dépend des paramètres internes du réseau de neurones, on peut calculer le gradient de l'erreur en fonction des paramètres internes. Utilisant ce gradient, l'agrégateur va modifier les paramètres internes du réseau de neurones pour minimiser la différence entre sa prédiction et l'effort observé,

$$\vec{\mu}^{\,i} = \vec{\mu}^{\,i} - \alpha \vec{\nabla} E_{neural,n}(\vec{\mu}^{\,i}) \tag{4.38}$$

Où α est le taux d'apprentissage.

Sélectionner aléatoirement les paramétrisations $\vec{p}_n$ à envoyer aux charges pendant l'apprentissage serait extrêmement inefficace. En effet, seule une toute petite fraction des paramétrisations possibles entraîne une puissance agrégée constante.

La méthode pour choisir judicieusement quelles paramétrisations tester lors de l'apprentissage sera développée à la section 4.7.3.

4.6.4 Banque de données d'observations

La méthode d'apprentissage décrite dans la section précédente observe la réponse de la charge au cours de l'intervalle et utilise cette information une seule fois pour améliorer le réseau de neurones associé à cette charge. Cependant, il existe une meilleure approche. En effet, en utilisant chaque observation une seule fois, l'efficacité de l'apprentissage est limitée par la vitesse à laquelle on descend le gradient de l'erreur de prédiction des réseaux de neurones. Cette vitesse de descente du gradient est proportionnelle au coefficient de taux d'apprentissage α dans l'équation de la rétropropagation du gradient. On ne peut pas simplement augmenter la valeur de α pour accélérer l'apprentissage car le gradient de l'erreur du réseau de neurones est local à la paire état-action et un taux d'apprentissage trop élevé cause de l'instabilité dans l'apprentissage et limite la précision du réseau de neurones.

De plus, il est possible que le réseau de neurones n'observe pas de réponses provenant d'une certaine partie du domaine état-action pendant de nombreux intervalles. Dans une telle situation, il est possible que le réseau de neurones oublie comment approximer l'effort résultant de ces paires état-action.

Pour remédier à ces deux problèmes, on enregistre chaque observation dans une banque d'observations. Cette approche est régulièrement utilisée dans la littérature sur l'apprentissage par renforcement. L'article [20], dans lequel cette approche a été introduite, appelle cette banque de données d'observations le "experience replay".

Durant l'apprentissage, on va régulièrement sélectionner aléatoirement un certain nombre d'observations dans la banque pour effectuer la rétropropagation des gradients et ainsi améliorer le réseau de neurones. Cette approche permet de réutiliser chaque observation autant de fois qu'il est nécessaire et donc la vitesse de l'apprentissage n'est plus limitée par α. Le facteur limitant est maintenant simplement qu'il faut observer les réponses de la charge provenant d'une portion significative de l'espace état-action pour approximer l'effort de la charge dans n'importe quelle situation. Cette approche règle aussi le problème de l'oubli puisque les observations seront réutilisées continuellement.

Sélection des observations

Précédemment, nous avons souligné les difficultés causées par la nature hautement dimensionnelle et continue de l'espace état-action de notre problème. Aussi, nous venons d'affirmer que le facteur limitant l'apprentissage est la nécessité d'explorer une portion significative de cet espace état-action. À première vue, cela semble problématique. Cependant, on remarque qu'il n'y a qu'une portion très restreinte de l'espace état-action qui est importante à ap-

prendre pour le réseau de neurones. En effet, l'objectif de l'agrégateur est toujours d'obtenir une courbe de puissance agrégée constante. À l'intérieur de l'espace d'action pour un état donné, seule une très petite région parmi toutes les paramétrisations possibles de $\vec{p}_n$ va résulter en une puissance agrégée constante. Donc, l'espace état-action qu'il est important de couvrir lors de l'apprentissage est en réalité une fraction minuscule de l'espace état-action total.

Dans l'article [21], il a été démontré que d'utiliser une stratégie pour prioriser certaines observations du "experience replay" lors de l'apprentissage plutôt que de sélectionner entièrement aléatoirement les observations a le potentiel d'améliorer l'efficacité de l'apprentissage. On va donc développer une méthode de priorisation des observations adaptée à notre situation.

On remarque que, dans les premiers temps de l'apprentissage, quand le réseau de neurones associé à la charge est encore très imprécis, l'agrégateur va tester des paramétrisations $\vec{p}_n$ qui ne vont pas résulter en une courbe de puissance constante. Cependant, au fur et à mesure que l'apprentissage se poursuit, les courbes de puissance obtenues vont devenir de plus en plus constantes. Afin de concentrer l'apprentissage du réseau de neurones dans la région de l'espace état-action d'intérêt, lorsque l'on sélectionne aléatoirement des observations pour l'apprentissage, on favorise la sélection des observations plus récentes de la banque d'observations. Cette modification spécifique à notre problème permet d'augmenter la précision du réseau de neurones puisqu'il doit apprendre à approximer la dynamique de la charge sur un domaine état-action plus restreint.

4.6.5 Avantages d'approximer l'effort plutôt que la puissance

Approximer l'effort plutôt que la puissance avec le réseau de neurones permet de ne pas avoir à fournir la puissance idéale u^i_{ideal} en entrée au réseau. Cela réduit beaucoup la taille du domaine sur lequel le réseau de neurones doit apprendre la fonction $\vec{f}^{\,i}_{neural}$ puisque celle-ci ne dépend pas de u_{ideal}. *Ce domaine réduit va permettre au réseau neuronal d'apprendre avec beaucoup moins d'expérimentations.* Étant donné qu'un des objectifs les plus importants de notre méthode est de minimiser la durée de l'apprentissage, cet avantage est très important.

De plus, approximer l'effort permet d'obtenir facilement une prédiction de l'état de la charge au début de l'intervalle $n+1$. En effet, connaissant l'état s^i_n et une paramétrisation $\vec{p}_n$, l'agrégateur peut utiliser le réseau de neurones pour estimer l'effort résultant $\vec{u}^{\,i}_{est,n}$. Ainsi, l'agrégateur obtient une estimation de l'état s^i_{n+1} (voir 4.14) :

$$s^i_{est,n+1} = \{u^i_{ideal}\ ,\ u^i_{est,n,K}\ ,\ p_{n,K}\} \tag{4.39}$$

Cette estimation $s^i_{est,n+1}$ de l'état à l'intervalle suivant sera essentielle pour trouver la commande optimale sur l'épisode dans le chapitre 5.

Finalement, il sera essentiel d'avoir une approximation de l'effort pour l'algorithme qui sera présenté à la section 4.8.

4.7 Commande optimale sur un intervalle

On rappelle que l'objectif de l'agrégateur est d'influencer la puissance agrégée moyenne de la population de façon à ce qu'elle soit égale à l'objectif de puissance pour chaque intervalle de l'épisode. Cependant, on va commencer par développer une méthode pour résoudre le problème pour un seul intervalle, sans tenir compte de l'état de la population après cet intervalle. Pour ce faire, l'agrégateur doit trouver la paramétrisation optimale $\vec{p}_n{}^*$ qui minimise la somme des carrés de la différence entre l'objectif de puissance agrégée $C_{obj,n}$, constant sur l'intervalle, et les valeurs de chacune des composantes du vecteur $\vec{C}_{est,n}$, qui approxime la puissance agrégée moyenne sur l'intervalle n.

$$\vec{p}_n{}^* = \underset{\vec{p}_n \in [0, q_{max}]^K}{\boldsymbol{arg\ min}} \quad E_{est,n} := \sum_{k=1}^{K} [\ C_{est,n,k} - C_{obj,n}\ ]^2 \tag{4.40}$$

Il sera nécessaire de pouvoir calculer la commande optimale sur un intervalle lors du calcul de la commande optimale sur l'épisode complet qui sera présenté dans le chapitre 5.

4.7.1 Calcul la fonction de pression optimale

Même s'il possède un ensemble de réseaux de neurones qui permet d'approximer parfaitement la puissance agrégée $\vec{C}_{est,n}$ de la population, comment l'agrégateur peut-il calculer la paramétrisation optimale $\vec{p}_n{}^*$ qui va minimiser l'erreur estimée $E_{est,n}$?

Connaissant l'état actuel de la population S_n, il est trivial pour l'agrégateur de tester une paramétrisation $\vec{p}_n$ pour estimer la puissance agrégée résultante. Cependant, l'espace des paramétrisations $\vec{p}_n$ possibles est hautement dimensionnel et continu. Donc, obtenir la paramétrisation optimale $\vec{p}_n{}^*$ par une recherche systématique sur l'entièreté de l'espace d'action n'est pas réaliste. Si on voulait tester j valeurs différentes pour chacune des K composantes de $\vec{p}_n$, il faudrait tester un nombre de paramétrisations égal à j^K. Ce nombre de tests étant trop grand pour être réalisable, il faut trouver une méthode qui permet de calculer $\vec{p}^*$ plus efficacement.

Une approche possible est la descente de gradient. Si on peut trouver une façon fiable de

calculer le gradient de l'erreur estimée $E_{est,n}$ en fonction des composantes de $\vec{p}_n$, on pourrait utiliser ce gradient pour converger itérativement vers $\vec{p}_n{}^*$ en suivant la direction inverse du gradient.

Gradient de l'erreur estimée

L'erreur de puissance estimée sur un intervalle n est égale à,

$$E_{est,n} := \sum_{k=1}^{K} [\ C_{est,n,k} - C_{obj,n}\ ]^2 \tag{4.41}$$

On cherche à obtenir le gradient de cette erreur selon le vecteur $\vec{p}_n$. Pour cela, on doit calculer la dérivée partielle de l'erreur selon chaque composante de $\vec{p}_n$. Si on applique la dérivation en chaîne pour un élément générique $p_{n,j}$ à l'intérieur de $\vec{p}_n$, on obtient l'équation suivante,

$$\frac{\partial E_{est,n}}{\partial p_{n,j}} = \sum_{k=1}^{K} \frac{\partial E_{est,n}}{\partial C_{est,n,k}} \sum_{i=1}^{I} \frac{\partial C_{est,n,k}}{\partial u^i_{est,n,k}} \frac{\partial u^i_{est,n,k}}{\partial p_{n,j}} \quad \forall j = 1, \ldots, K \tag{4.42}$$

Dans laquelle la dérivée partielle de l'erreur en fonction de la puissance agrégée estimée $C_{est,n,k}$ est la suivante,

$$\frac{\partial E_{est,n}}{\partial C_{est,n,k}} = 2(C_{est,n,k} - C_{obj,n}) \tag{4.43}$$

Dérivée partielle de la puissance agrégée

La dérivée partielle de la puissance agrégée estimée en fonction de l'effort estimé des charges individuelles est la suivante,

$$\frac{\partial C_{est,n,k}}{\partial u^i_{est,n,k}} = \begin{cases} 0, & \text{si } u^i_{est,n,k} < -u^i_{ideal} \\ \frac{1}{N}, & \text{si } u^i_{est,n,k} \in [-u^i_{ideal},\ c^i_{max} - u^i_{ideal}) \\ 0, & \text{si } u^i_{est,n,k} \geq c^i_{max} - u^i_{ideal} \end{cases} \tag{4.44}$$

(4.44) découle du fait que si le réseau de neurones estime que la charge a déjà dépassé les limites sur la puissance, un changement dans la valeur de l'effort calculé ne résultera pas en un changement de la puissance de la charge.

Étant donné qu'on cherche à obtenir une puissance agrégée constante sur un intervalle, la puissance des charges individuelles a tendance à ne pas varier beaucoup durant un intervalle. On fait donc l'approximation que la puissance de chaque charge va soit être restreinte par

les limites sur la puissance durant tout l'intervalle, soit ne pas être restreinte du tout durant l'intervalle. On définit donc la variable L_n^i, dont la valeur est constante sur tout l'intervalle,

$$\frac{\partial C_{est,n,k}}{\partial u^i_{est,n,k}} \approx L_n^i \in \{0, \frac{1}{N}\} \quad \forall k = 1, \dots, K \tag{4.45}$$

Dérivée partielle de l'effort

La difficulté de calculer le gradient de l'erreur estimée réside principalement dans le calcul de la dérivée partielle de l'effort de la charge i au moment k en fonction de l'élément j de la paramétrisation de la fonction de pression.

$$\frac{\partial u^i_{est,n,k}}{\partial p_{n,j}} \tag{4.46}$$

Le calcul analytique de cette dérivée partielle exigerait de connaître les paramètres thermiques de la charge i, ce qui n'est pas possible pour l'agrégateur. Cependant, l'agrégateur possède le réseau de neurones $\vec{f}^{\,i}_{neural}$ qui estime l'effort de la charge i,

$$\vec{u}^{\,i}_{est,n} = \vec{f}^{\,i}_{neural}(u^i_{n-1,K}\ ,\ p_{n-1,K}\ ,\ \vec{p}_n\ ,\ \vec{\mu}^{\,i}) \tag{4.47}$$

Similairement à la rétropropagation du gradient lors de l'apprentissage du réseau de neurones, il est possible d'obtenir les dérivées partielles des composantes de $\vec{u}^{\,i}_{est,n}$ par rapport aux composantes de $\vec{p}_n$ en entrée. Cependant, même si le réseau de neurones approxime très précisément $\vec{f}^{\,i}_{neural}$, les dérivées partielles obtenues avec les paramètres internes du réseau ne vont pas nécessairement être suffisamment précises pour utiliser directement la méthode de descente de gradient.

Bien qu'une simple méthode de descente de gradient ne soit pas possible, l'article [22] propose une méthode pour résoudre ce type de problème d'optimisation. Afin de minimiser un coût estimé par un réseau neuronal, l'algorithme QT-Opt est une méthode d'optimisation aléatoire qui va converger vers un minimum de la fonction de coût en évaluant itérativement le coût en réponse à des paramètres qui sont échantillonnés depuis une loi normale multidimensionnelle. La loi normale multidimensionnelle est ajustée itérativement pour favoriser les combinaisons de paramètres qui offrent le coût le plus faible.

Cependant, étant donné que cette méthode est assez complexe et que l'optimisation aléatoire n'est pas le sujet de ce mémoire, on choisit d'utiliser une méthode plus simple, basée sur notre connaissance du modèle des charges thermostatiques.

Heuristique pour la dérivée partielle de l'effort

Pour obtenir une approximation de $\frac{\partial u^i_{est,n,k}}{\partial p_{n,j}}$, on utilise encore une fois notre connaissance du mécanisme par lequel les charges calculent leur effort. Augmenter la valeur de la fonction de pression à l'instant t a pour effet de pénaliser la différence entre la température de la charge à t et la température minimale. La charge est donc poussée à réduire sa puissance de chauffage entre 0 et t pour atteindre une température plus basse à t. Sur la base de cette observation, on élabore l'heuristique suivante pour approximer la dérivée partielle,

$$\frac{\partial u^i_{est,n,k}}{\partial p_{n,j}} \approx \begin{cases} \frac{1}{j} D^i, & \text{si } k \leq j \\ 0, & \text{si } k > j \end{cases} \tag{4.48}$$

Dans laquelle la constante $D^i < 0$ est inconnue mais strictement négative. Cette constante négative représente le fait que la valeur de la fonction de pression à l'instant t d'un intervalle va influencer négativement l'effort entre 0 et t.

De plus, on suppose que les impacts que les composantes de $\vec{p}_n$ ont sur l'effort $\vec{u}_n$ sont dans un même ordre de grandeur. On multiplie donc la constante D^i par $1/j$ afin de normaliser l'impact total de chaque composante de $\vec{p}_n$ sur l'erreur $E_{est,n}$.

Cette heuristique permet à l'agrégateur d'obtenir une approximation des dérivées partielles afin de pouvoir calculer le gradient de l'erreur de puissance. Cette heuristique n'a pas besoin d'être exacte, elle doit simplement être suffisamment correcte afin de permettre à la méthode de descente du gradient de converger vers la paramétrisation optimale. Une analyse mathématique de cette heuristique se trouve à l'annexe C.

Formule de la descente du gradient

Avec notre heuristique, on peut maintenant calculer la dérivée partielle de l'erreur,

$$\frac{\partial E_{est,n}}{\partial p_{n,j}} = \sum_{k=1}^{K} \frac{\partial E_{est,n}}{\partial C_{est,n,k}} \sum_{i=1}^{I} \frac{\partial C_{est,n,k}}{\partial u^i_{est,n,k}} \frac{\partial u^i_{est,n,k}}{\partial p_{n,j}} \quad \forall j = 1, \ldots, K \tag{4.49}$$

Remplaçant les termes, on obtient,

$$\frac{\partial E_{est,n}}{\partial p_{n,j}} \approx \sum_{k=1}^{j} 2(C_{est,n,k} - C_{obj,n}) \sum_{i=1}^{I} L^i_n \frac{1}{j} D^i \tag{4.50}$$

$$\frac{\partial E_{est,n}}{\partial p_{n,j}} \approx 2\sum_{i=1}^{I} L_n^i D^i \frac{1}{j}\sum_{k=1}^{j}(C_{est,n,k} - C_{obj,n}) \tag{4.51}$$

On peut remplacer la constante inconnue $2\sum_{i=1}^{I} L_n^i D^i$ par H_n. H_n est inconnue mais strictement négative,

$$\frac{\partial E_{est,n}}{\partial p_{n,j}} \approx H_n \frac{1}{j}\sum_{k=1}^{j}(C_{est,n,k} - C_{obj,n}) \tag{4.52}$$

Pour obtenir la paramétrisation optimale $\vec{p}_n^*$ qui minimise l'erreur sur l'intervalle, l'agrégateur commence par choisir une paramétrisation $\vec{p}_n$ initiale arbitraire. L'agrégateur obtient ensuite une estimation de la puissance agrégée et de l'erreur résultante en utilisant les réseaux de neurones entraînés sur la population. L'agrégateur calcule ensuite le gradient de l'erreur estimée en fonction des composantes de $\vec{p}_n$ Finalement, l'agrégateur modifie itérativement $\vec{p}_n$ dans la direction inverse du gradient, jusqu'à ce que la paramétrisation converge vers la paramétrisation optimale. Le coefficient H_n inconnu est absorbé dans le taux de descente du gradient α. Étant donné que H_n est négatif, α est positif. De plus, on limite la valeur de la fonction de pression entre 0 et q_{max}.

$$p_{n,j} = min\left(max\left(p_{n,j} + \alpha\frac{1}{j}\sum_{k=1}^{j}(C_{est,n,k} - C_{obj,n})\ ,\ 0\right),\ q_{max}\right) \quad \forall j = 1, \ldots, K \tag{4.53}$$

4.7.2 Exemple de descente de gradient

Pour cet exemple, on utilise un réseau de neurones qui approxime l'effort d'une charge avec les paramètres de la charge générique introduite dans la section 3.4.

Dans la figure 4.5, le graphique du bas montre chacune des paramétrisations $\vec{p}$ testées séquentiellement lors de la recherche de la paramétrisation optimale par descente de gradient.

Le graphique du haut montre les trajectoires de puissance estimées $\vec{c}_{est}$ associées aux paramétrisations $\vec{p}$ testées. La ligne rouge montre l'objectif de puissance de l'agrégateur pour cet intervalle. La puissance estimée converge vers l'objectif de puissance, ce qui valide notre approche de descente de gradient.

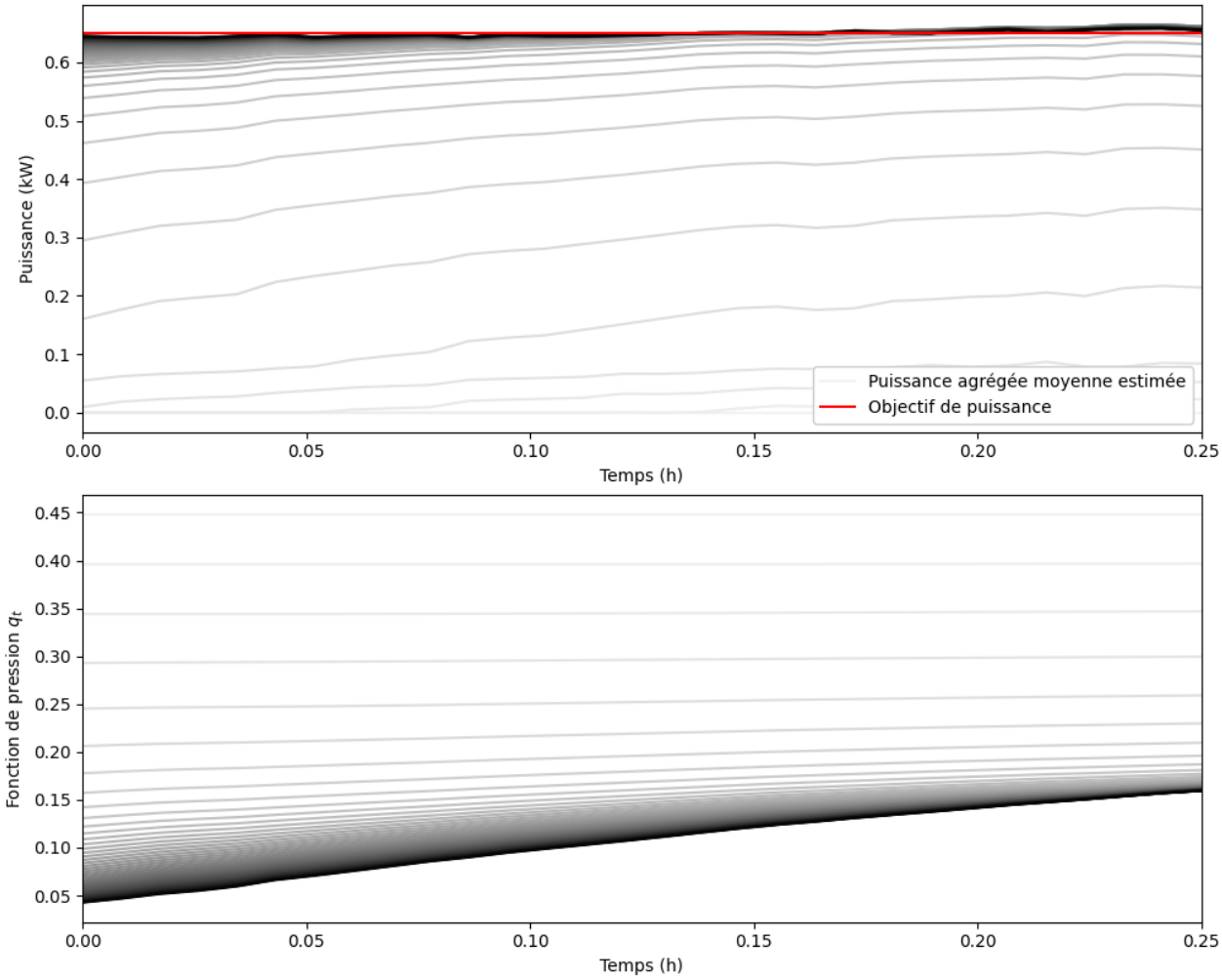

FIGURE 4.5 Descente du gradient pour calculer la fonction de pression optimale

4.7.3 Stratégie d'exploration pour l'apprentissage

Maintenant qu'on possède une méthode pour calculer la paramétrisation optimale sur un intervalle pour un objectif de puissance agrégée donné, on peut élaborer une stratégie d'exploration pour faire apprendre la dynamique des charges de la population aux réseaux de neurones de l'agrégateur.

Plutôt que de tester aléatoirement des paramétrisations $\vec{p}_n$ pendant l'apprentissage, l'agrégateur va sélectionner un objectif de puissance agrégée $C_{obj,n}$ à chaque intervalle. Pour choisir une paramétrisation $\vec{p}_n$ à envoyer à la population, l'agrégateur performe la méthode de descente de gradient décrite dans la section précédente pour chercher $\vec{p}_n{}^*$. Étant donné que les réseaux de neurones $\vec{f}_{neural}^{\,i}$ sont en phase d'apprentissage et qu'ils n'approximent pas correctement la puissance de chauffage des charges, il est peu probable que la puissance agrégée

résultante soit conforme à l'objectif de puissance. Cependant, les réseaux de neurones vont améliorer $\vec{f}^{\,i}_{neural}$ avec cette expérience. La prochaine fois que l'agrégateur va sélectionner un objectif de puissance similaire dans un état similaire de la population, l'approximation $\vec{f}^{\,i}_{neural}$ sera meilleure et la méthode de descente de gradient permettra de trouver une paramétrisation plus proche de la paramétrisation optimale.

Cette approche est avantageuse car, même durant l'apprentissage, l'agrégateur pourra influencer la puissance agrégée de la population vers l'objectif $C_{obj,n}$. En effet, à chaque intervalle, l'agrégateur va envoyer aux charges sa meilleure estimation de $\vec{p}_n^{\,*}$. Même si les réseaux de neurones ne sont pas très précis, l'estimation de $\vec{p}_n^{\,*}$ va tout de même influencer la puissance agrégée de la population vers l'objectif $C_{obj,n}$.

4.8 Calcul de l'état aux valeurs de puissance limites

Lorsque l'agrégateur mesure la puissance de chauffage d'une charge durant un intervalle n, il doit calculer rétrospectivement l'effort $u^i_{n,K}$ calculé par la charge car $u^i_{n,K}$ fait partie de l'état de la charge au début de l'intervalle suivant. Comme mentionné précédemment, la charge peut seulement consommer une puissance entre 0 et c^i_{max}. Donc, quand l'agrégateur mesure une puissance qui est égale à 0 ou c^i_{max}, il n'est pas possible d'obtenir l'effort calculé par la charge puisque la puissance que l'agrégateur mesure a probablement été limitée par 0 ou c^i_{max} et ne représente pas l'effort calculé par la charge.

Pour obtenir une estimation $s^i_{est,n+1}$ de l'état d'une charge après un intervalle dans lequel sa puissance a été limitée, l'agrégateur peut utiliser le réseau de neurones $\vec{f}^{\,i}_{neural}$ associé à la charge. On rappelle que $\vec{u}^{\,i}_{est,n}$ est l'approximation du réseau de neurones de l'effort calculé par la charge,

$$\vec{u}^{\,i}_{est,n} = \vec{f}^{\,i}_{neural}(u^i_{n-1,K}\ ,\ p_{n-1,K}\ ,\ \vec{p}_n\ ,\ \vec{\mu}^{\,i}) \tag{4.54}$$

On définit l'effort efficace $\vec{u}^{\,i}_{eff,n}$ comme étant la portion de l'effort calculé par la charge qui n'a pas été limité par les limites sur la puissance. Par définition, l'effort efficace est égal à la différence entre la puissance observée $\vec{c}^{\,i}_{obs,n}$ et la puissance en régime permanent à température idéale u^i_{ideal},

$$\vec{u}^{\,i}_{eff,n} = \vec{c}^{\,i}_{obs,n} - u^i_{ideal}\vec{1} \tag{4.55}$$

L'agrégateur peut donc facilement calculer la valeur de l'effort efficace à partir de son observation de la puissance de la charge.

L'agrégateur va ensuite calculer rétrospectivement une paramétrisation différente $\vec{p}^{\,i}_{eff,n}$ pour l'intervalle n qui aurait poussé la charge à calculer un effort exactement égal à $\vec{u}^{\,i}_{eff,n}$. Pour calculer cette paramétrisation, l'agrégateur doit résoudre le problème d'optimisation suivant,

$$\vec{p}^{\,i}_{eff,n} = \underset{\vec{p}_n \in [0, q_{max}]^K}{\boldsymbol{arg\ min}} \sum_{k=1}^{K} \left[u^i_{est,n,k} - u^i_{eff,n,k} \right]^2 \tag{4.56}$$

On remarque que ce problème est extrêmement similaire au problème de la recherche de la paramétrisation optimale pour un intervalle décrit par l'équation (4.40). On peut donc utiliser la même méthode de descente du gradient pour arriver itérativement à la valeur de $\vec{p}^{\,i}_{eff,n}$. L'équation décrivant la modification itérative des composantes de $\vec{p}_n$ est la suivante,

$$p_{n,j} = p_{n,j} + \alpha \frac{1}{j} \sum_{k=1}^{j} (u^i_{est,n,k} - u^i_{eff,n,k}) \ \forall j \in [1, K] \tag{4.57}$$

Après un certain nombre d'itérations, l'agrégateur obtient une approximation de $\vec{p}^{\,i}_{eff,n}$. On remarque que si la charge avait reçu la paramétrisation $\vec{p}^{\,i}_{eff,n}$ au lieu de $\vec{p}_n$ pour l'intervalle n, la charge aurait consommé une puissance égale à la puissance $\vec{c}^{\,i}_{obs,n}$ observée, sans toutefois être limitée par les limites sur la puissance.

Les seuls facteurs ayant un impact sur la température d'une charge à la fin d'un intervalle sont sa température au début de l'intervalle et sa puissance de chauffage durant l'intervalle. Étant donné que $\vec{p}^{\,i}_{eff,n}$ aurait causé une puissance égale à la vraie puissance observée $\vec{c}^{\,i}_{obs,n}$, on peut approximer l'état de la charge à l'intervalle $n+1$ comme suit,

$$s^i_{est,n+1} = \{u^i_{ideal}\ ,\ u^i_{eff,n,K}\ ,\ p^i_{eff,n,K}\} \tag{4.58}$$

4.9 Simulation de l'apprentissage

4.9.1 Méthodologie

Finalement, on teste la méthode développée avec une simulation de l'apprentissage pour une population de 20 charges distinctes. Afin de simuler une population hétérogène, les paramètres thermiques des charges sont distribués uniformément entre ±10% de la valeur des paramètres de la charge générique définie dans la section 3.4. Pendant l'apprentissage, l'agrégateur utilise des intervalles de 15 minutes et des épisodes contenant 12 intervalles.

On rappelle que les charges peuvent changer leur température idéale en dehors des périodes de contrôle. Cependant, lorsqu'un épisode de contrôle commence, elles doivent maintenir

leur température idéale pour tout l'épisode. Donc, lors de la simulation, nous sélectionnons aléatoirement une température idéale ainsi qu'une température externe pour chacune des charges de la population au début de chaque épisode.

À chaque intervalle, l'agrégateur choisit un objectif de puissance agrégée $C_{obj,n}$ aléatoirement entre 0 et C_{ideal}. C_{ideal} est la puissance agrégée moyenne quand toutes les charges de la population sont en régime permanent à leurs températures idéales.

Connaissant $C_{obj,n}$ et l'état S_n de la population, l'agrégateur utilise la méthode de descente de gradient pour calculer $\vec{p}_n^*$, la fonction de pression optimale sur l'intervalle n.

Étant donné que l'apprentissage se fait avec des charges thermostatiques simulées, nous résolvons l'équation de HJB pour chaque charge en utilisant $\vec{p}_n^*$. Ensuite, connaissant les valeurs de π_t et β_t $\forall t \in [(n-1)\Delta t,\ n\Delta t]$, nous utilisons la méthode d'intégration numérique d'Euler-Maruyama pour calculer la température et la puissance de chauffage de chaque charge sur l'intervalle.

Finalement, comme décrit dans la section 4.6.3, l'agrégateur utilise ses observations de la puissance consommée par les charges pour améliorer les réseaux de neurones. Il utilise aussi l'observation de la puissance des charges pour calculer l'état au début de l'intervalle suivant.

4.9.2 Résultats

Apprentissage des réseaux de neurones

La figure 4.6 contient l'erreur des réseaux de neurones au fil des épisodes durant l'apprentissage. Cette erreur est la moyenne des différences carrées entre la puissance agrégée estimée par les réseaux de neurones et la puissance agrégée mesurée par l'agrégateur sur tout l'intervalle.

$$E_{neural} := \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(C^{\ i}_{est,n,k} - C^{\ i}_{obs,n,k})^2 \tag{4.59}$$

On observe que l'erreur des réseaux de neurones diminue rapidement jusqu'à atteindre un plateau autour de $2{*}10^{-3}$. Cette limite sur la précision de l'estimation de la puissance agrégée par les réseaux de neurones est principalement due au bruit dans la dynamique thermique des charges.

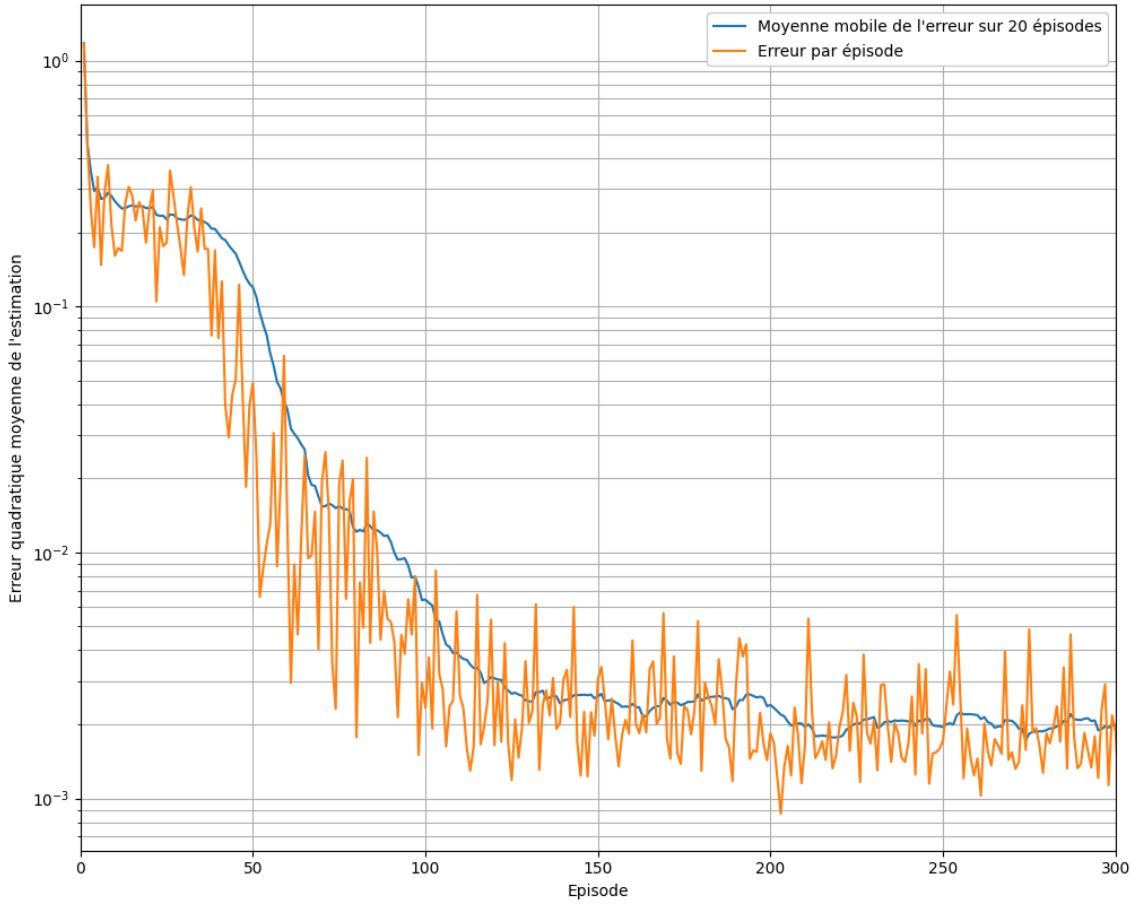

FIGURE 4.6 Erreur d'estimation des réseaux de neurones durant l'apprentissage

Simulation d'un épisode de contrôle

La figure 4.7 montre un exemple d'épisode qui a lieu après que l'erreur des réseaux de neurones ait atteint son plateau minimum.

Le graphique en haut à gauche présente l'évolution de la température de 3 charges sélectionnées aléatoirement parmi la population de 20 charges. Chaque couleur représente une des 3 charges. La ligne pleine suit la trajectoire de la température de la charge à travers les intervalles de l'épisode. La ligne en tirets indique la température idéale z^i_{ideal} tandis que la ligne pointillée indique la température minimale acceptable z_{min}. L'effort calculé par les charges est proportionnel à la différence entre leurs températures et leurs températures minimales (3.14). Les températures des charges devraient donc évoluer similairement à l'intérieur de leurs intervalles de température $[z_{min}, z_{ideal}]$ respectifs. Cette équité de participation est confirmée par le graphique qui montre que les températures évoluent selon nos attentes.

Le graphique en bas à gauche présente les trajectoires de puissance individuelles. La ligne

pleine suit la trajectoire de puissance réelle tandis que la ligne pointillée suit la trajectoire prédite par le réseau de neurones. Sur ce graphique, l'effet du bruit est très visible puisque chaque petite déviation de température entraîne une réponse dans le calcul de l'effort (3.14). De plus, l'erreur entre la vraie trajectoire de la puissance et la trajectoire prédite reste à peu près constante sur l'entièreté de l'épisode. Cela confirme que notre formulation alternative de l'état est correcte et permet à l'agrégateur de connaître l'état des charges pour approximer leur puissance correctement. Un intervalle particulièrement intéressant est l'intervalle allant de 2.25 heures à 2.5 heures. Lors de cet intervalle, l'effort de la charge bleue et de la charge verte pousse leur puissance à 0 kW. Comme mentionné dans la section 4.5.4, il n'est pas possible pour l'agrégateur d'obtenir rétrospectivement l'effort calculé par les charges dans cette situation. Pour obtenir une estimation de l'état de la charge bleue et de la charge verte, l'agrégateur utilise la méthode présentée à la section 4.8. Dans l'intervalle suivant (allant de 2.5 heures à 2.75 heures), nous observons que les prédictions de l'agrégateur pour la charge bleue et la charge verte sont très précises, ce qui valide l'efficacité de la méthode d'estimation de l'état.

Dans le graphique en haut à droite, la ligne pleine rouge suit la trajectoire de puissance agrégée et la ligne en tirets noire indique l'objectif de puissance agrégée pour chaque intervalle. L'erreur entre la puissance agrégée et l'objectif est petite sur tout l'épisode. L'effet du bruit, très visible sur les trajectoires de puissances individuelles, est atténué sur la trajectoire agrégée moyenne puisque les bruits sont indépendants.

Finalement, le graphique en bas à droite présente la trajectoire de la fonction de pression q_t sur chaque intervalle de l'épisode. La fonction de pression q_t est monotone croissante sur tous les intervalles sauf pour l'intervalle allant de 1.75 heures à 2 heures. Sur cet intervalle, l'objectif de puissance agrégée est élevé, ce qui fait que la température des charges augmente. Nous rappelons que l'effort est calculé selon l'équation suivante :

$$u_t = -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) \tag{4.60}$$

Étant donné que l'objectif de l'agrégateur est d'obtenir une puissance agrégée constante sur la durée d'un intervalle, la variation de l'effort dans le temps devrait être nulle. Si la puissance agrégée demandée a pour effet de faire diminuer la température des charges, les valeurs de π_t et β_t doivent augmenter pour maintenir un effort u_t constant. Pour ce faire, la fonction de pression q_t doit être croissante sur l'intervalle.

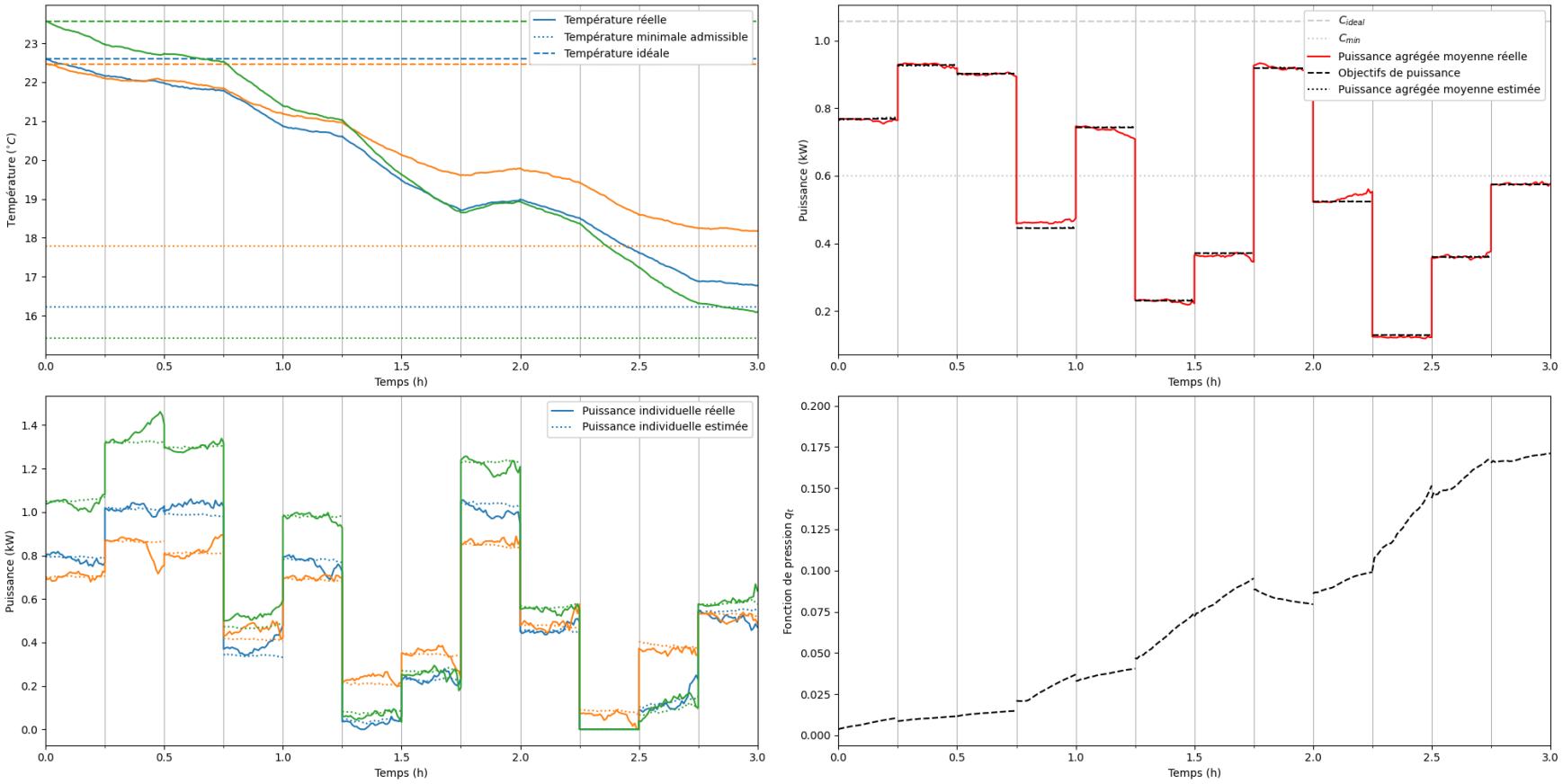

FIGURE 4.7 Commande de la population intervalle par intervalle

CHAPITRE 5 COMMANDE OPTIMALE SUR UN ÉPISODE

5.1 Exemple de consommation agrégée non optimale sur un épisode

Dans le chapitre 4, nous avons présenté la méthode pour calculer la fonction de pression optimale sur un intervalle. Cependant, celle-ci n'est pas nécessairement optimale si l'épisode complet est considéré. La simulation présentée dans cette section utilise la même population de charges que dans la section 4.9.

La figure 5.1 illustre une situation possible dans laquelle la méthode présentée dans la section précédente n'est pas appropriée. La puissance agrégée de la population sur les 9 premiers intervalles est très proche de l'objectif de puissance agrégée. Sur ces 9 intervalles, l'erreur est due aux pertes et gains aléatoires des charges et aux imprécisions des réseaux neuronaux de l'agrégateur. Cependant, dans les 3 derniers intervalles, quand un nombre critique de charges parmi la population atteint des températures proches de leurs températures minimales, l'erreur de puissance augmente rapidement. Finalement, quand toutes les charges atteignent leurs températures minimales, la puissance agrégée de la population se stabilise à C_{min}. C_{min} étant la puissance agrégée moyenne quand toutes les charges de la population sont en régime permanent à leurs températures minimales admissibles. La population va toujours se comporter de façon similaire à cet exemple quand plusieurs intervalles de suite ont des objectifs de puissance $C_{obj,n} < C_{min}$. Cette trajectoire de puissance agrégée ne minimise pas l'erreur quadratique sur tout l'épisode. L'erreur quadratique dans les derniers intervalles est démesurément plus grande que dans tous les intervalles précédents et il existe certainement une stratégie de contrôle pour l'épisode entier qui permettrait de répartir l'erreur et donc de diminuer l'erreur quadratique. Ce résultat n'est pas surprenant puisque la méthode utilisée par l'agrégateur pour calculer la fonction de pression à envoyer à la population ne considère qu'un seul intervalle à la fois.

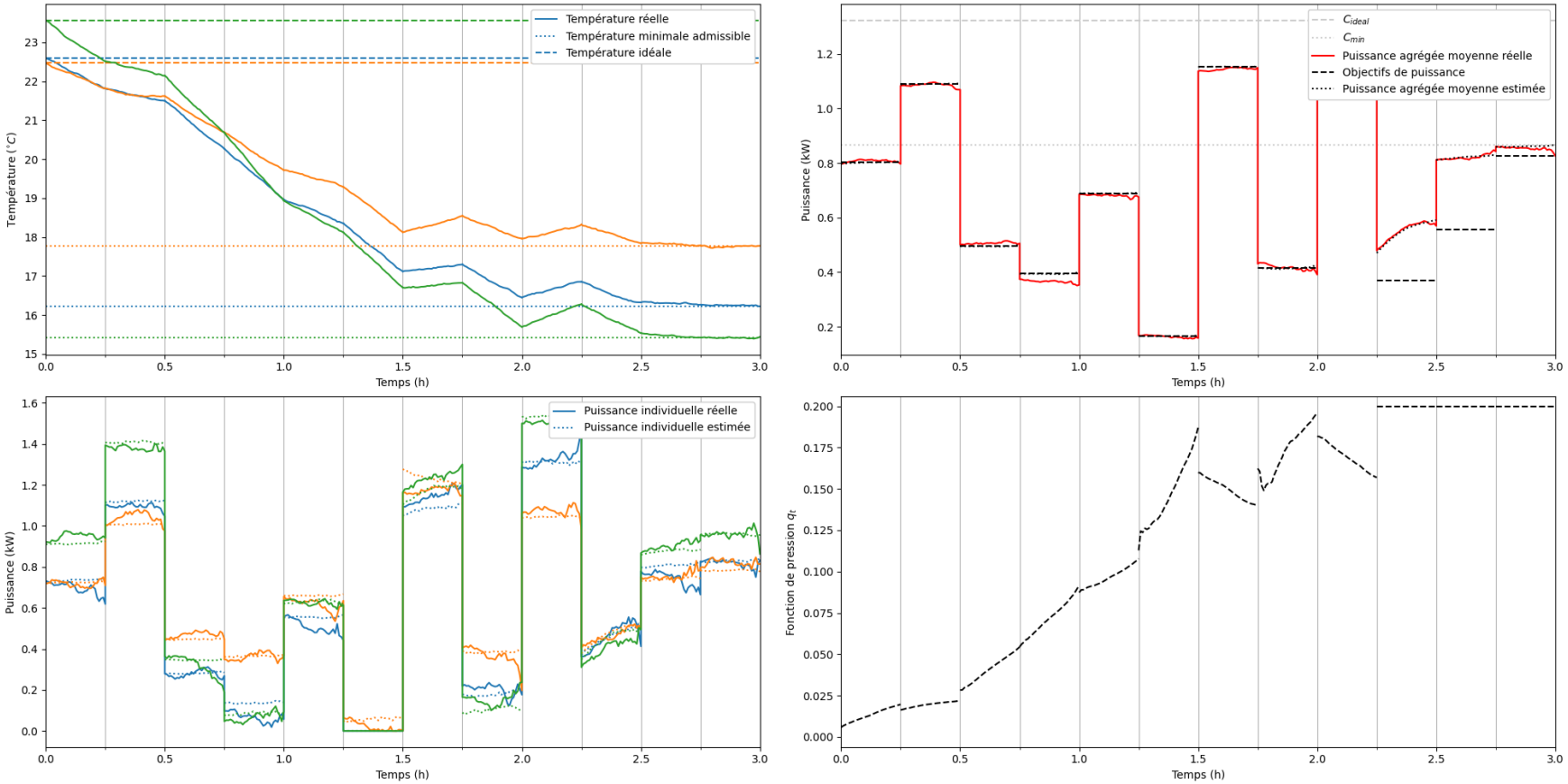

FIGURE 5.1 Exemple de commande non optimale sur un épisode

5.2 Optimalité sur l'épisode

On définit l'erreur estimée par l'agrégateur sur l'épisode,

$$E_{est,ep}(S_1, \vec{P}, \vec{C}_{obj}) := \sum_{n=1}^{N} \sum_{k=1}^{K} [\ C_{est,n,k} - C_{obj,n}\ ]^2 \tag{5.1}$$

où $\vec{P}$ est le vecteur contenant les paramétrisations $\vec{p}_n$ pour tous les intervalles de l'épisode,

$$P_n = \vec{p}_n \quad \forall n = 1, \dots, N \tag{5.2}$$

et $\vec{C}_{obj}$ est le vecteur contenant les objectifs de puissance agrégée pour tous les intervalles de l'épisode.

Imaginons une situation possible dans laquelle l'agrégateur pourrait se trouver. Il y a une forte couverture nuageuse et un faible vent sur une partie significative des installations de production d'énergie renouvelable. Les prédictions météorologiques estiment que cette situation va persister pendant environ 1 heure. Cette situation coïncide avec le retour du travail d'une grande partie de la population et l'agrégateur prévoit une surcharge, donc il voudrait réduire la puissance agrégée des charges thermostatiques de la population pour réduire la demande sur le réseau.

Dans une telle situation, on cherche à minimiser l'erreur estimée sur l'épisode pour un épisode de 1 heure comportant $1/\Delta t$ intervalles. L'agrégateur connaît déjà $\vec{C}_{obj}$, le profil de puissance recherché pour l'heure suivante. Aussi, l'agrégateur connaît l'état initial de la population S_1. L'agrégateur cherche donc à trouver $\vec{P}^*$, le vecteur contenant les paramétrisations $\vec{p}_n$ pour tous les intervalles de l'épisode qui minimisent l'erreur estimée sur l'épisode,

$$\vec{P}^* = \underset{\vec{P} \in [0, q_{max}]^{NK}}{\boldsymbol{arg\ min}} \quad E_{est,ep}(S_1, \vec{P}, \vec{C}_{obj}) := \sum_{n=1}^{N} \sum_{k=1}^{K} [\ C_{est,n,k} - C_{obj,n}\ ]^2 \tag{5.3}$$

Contrairement à la recherche de la fonction de pression optimale sur un intervalle, on ne peut pas utiliser une méthode de descente de gradient pour calculer $\vec{P}^*$ puisque l'on ne possède pas de gradient décrivant la variation de l'erreur estimée sur l'épisode $E_{est,ep}$ selon les composantes de $\vec{P}$.

5.3 Formulation du processus de décision markovien

Le problème de la recherche de la commande optimale pour un épisode ayant N intervalles se modélise parfaitement par un processus de décision markovien (MDP) dans lequel chaque intervalle de l'épisode correspond à une étape du MDP. Un MDP est défini par le quadruplet suivant :

- L'ensemble d'états possibles H
- L'ensemble d'actions possibles A
- La fonction de transition $G(h_n, h'_{n+1}, a_n)$ qui donne la probabilité de passer de l'état h à l'état h' avec l'action $a \in A$
- La fonction de coût $R(h_n, a_n)$ qui donne l'espérance de la valeur du coût engendré par l'action a à l'état h

Le MDP que l'on définit dans cette section ne représente pas la vraie population de charges. Plutôt, le MDP représente l'approximation de la population que l'agrégateur possède à l'aide de ses réseaux de neurones. En effet, l'agrégateur ne connaît pas les vraies fonctions G et R puisqu'il ne connaît pas les paramètres thermiques des charges. Cependant, à l'aide des réseaux de neurones, l'agrégateur possède la fonction $\hat{G}$ qui approxime G et la fonction $\hat{R}$ qui approxime R. On suppose que les réseaux de neurones ont terminé leur apprentissage.

L'action a_n à l'étape n du MDP est le choix de la fonction de pression pour cet intervalle,

$$a_n = \vec{p}_n \in A = [0, q_{max}]^K \tag{5.4}$$

Jusqu'à présent, on a considéré l'état de la population comme étant,

$$S_n = \{\vec{u}_{ideal},\ \vec{u}_{n-1,\ K},\ p_{n-1,\ K}\} \tag{5.5}$$

Cependant, l'état markovien du système h_n et l'action a_n doivent être suffisants pour évaluer la fonction de coût $\hat{R}(h_n, a_n)$ et la fonction de transition $\hat{G}(h_n, h'_{n+1}, a_n)$. On définit donc l'état markovien comme,

$$h_n = \{S_n,\ \vec{C}_{obj,n:N}\} \in H \tag{5.6}$$

où $\vec{C}_{obj,n:N}$ est le vecteur contenant les objectifs de puissance agrégée pour les étapes de n à N.

Le coût estimé $\hat{R}(h_n, a_n)$ pour l'étape n du MDP est donc le suivant,

$$\hat{R}(h_n, a_n) = E_{est,n}(S_n, \vec{p}_n, C_{obj,n}) := \sum_{k=1}^{K} [\ C_{est,n,k} - C_{obj,n}\ ]^2 \tag{5.7}$$

La fonction de transition $G(h_n, h'_{n+1}, a_n)$ doit donner la probabilité de passer à l'état h_{n+1} en choisissant l'action a_n dans l'état h_n. Examinons l'état h_{n+1} composante par composante.

$$h_{n+1} = \{S_{n+1},\ \vec{C}_{obj,n+1:N}\} \tag{5.8}$$

Le vecteur $\vec{C}_{obj,n+1:N}$ est simplement obtenu en retirant la première composante du vecteur $\vec{C}_{obj,n:N}$. Comme démontré à la section 4.5.2, S_{n+1} dépend entièrement de S_n et de la fonction de pression $\vec{p}_n$.

$$s^i_{est,n+1} = \{u^i_{ideal}\ ,\ u^i_{est,n,K}\ ,\ p_{n,K}\}\ \forall i \in 1, \ldots, I \tag{5.9}$$

Remplaçant $u^i_{est,n,K}$ par $f^{\ i}_{neural,K}$, on obtient,

$$s^i_{est,n+1} = \{u^i_{ideal}\ ,\ f^{\ i}_{neural,K}(u^i_{n-1,K}\ ,\ p_{n-1,K}\ ,\ \vec{p}_n\ ,\ \vec{\mu}^{\ i})\ ,\ p_{n,K}\}\ \forall i \in 1, \ldots, I \tag{5.10}$$

Nous remarquons que l'agrégateur calcule directement l'état estimé $s^i_{est,n+1}$ plutôt que de calculer une distribution de probabilité sur les états possibles. Donc, la fonction de transition estimée $\hat{G}$ est déterministe. Elle calcule l'état h_{n+1} pour un état h_n et une action a_n donnés.

$$h_{n+1} = \hat{G}(h_n, a_n) \tag{5.11}$$

5.3.1 Équation d'optimalité de Bellman

Si on possède un MDP dont on connaît les fonctions de coût R et de transition G, il est possible de calculer la stratégie optimale à partir de l'équation d'optimalité de Bellman [17],

$$v^*(h_n) = \underset{a_n \in A}{\boldsymbol{min}} \sum_{h'_{n+1}} G(h_n, h'_{n+1}, a_n) \left[\ R(h_n, a_n) + v^*(h'_{n+1})\ \right] \tag{5.12}$$

Dans laquelle $v^*(h_n)$ représente la valeur optimale à l'état h_n. La valeur optimale est la somme minimale des coûts pour les intervalles de $n+1$ à N qu'il est possible d'obtenir avec une stratégie parfaite.

Étant donné que notre MDP représente un épisode de N intervalles, on peut écrire pour le

dernier intervalle de l'épisode,

$$v^*(h_N) = \underset{a_N \in A}{\boldsymbol{min}} \; R(h_N, a_N) \tag{5.13}$$

Remplaçant les fonctions G et R par les fonctions approximées de l'agrégateur $\hat{G}$ et $\hat{R}$, on obtient,

$$v^*(h_n) = \underset{a_n \in A}{\boldsymbol{min}} \; \left[\hat{R}(h_n, a_n) + v^*(\hat{G}(h_n, a_n))\right] \tag{5.14}$$

La somme sur tous les états $h'_{n+1} \in H$ possibles disparaît puisque la fonction de transition estimée $\hat{G}$ est déterministe.

5.4 Optimisation de l'algorithme

Afin de calculer la stratégie optimale pour l'épisode complet et résoudre l'équation (5.14), il reste encore le problème de la minimisation sur l'espace d'action hautement dimensionnel et continu. Pour simplifier la résolution de cette équation, on va réduire la dimensionnalité de l'espace d'action A.

5.4.1 Séquence d'objectifs réalisable

On remarque que, dépendamment de l'état S_n de la population, il est possible de satisfaire différents objectifs de puissance agrégée. Au début d'un épisode, une population dans laquelle toutes les charges sont à leur température idéale peut atteindre un objectif de puissance agrégée $C_{obj} = 0$ avec une fonction de pression suffisamment grande. Par contre, une population dont toutes les charges sont à leurs températures minimales ne peut pas diminuer sa puissance agrégée en dessous de C_{min}.

Pour un état S_n donné de la population, on qualifie un objectif de puissance $C_{obj,n}$ de réalisable s'il est possible pour la puissance agrégée de la population d'atteindre cet objectif sur l'entièreté de l'intervalle n. On définit $C_{obj,min}(S_n)$ comme étant l'objectif de puissance agrégée le plus bas qui est réalisable pour S_n.

Similairement, une séquence d'objectifs est réalisable si les objectifs dans la séquence sont réalisables les uns à la suite des autres.

Notons que, si $\vec{C}_{obj}$, le vecteur contenant les objectifs de puissance agrégée pour tous les intervalles de l'épisode, est réalisable, cela implique que $\vec{p}_n^{\,*}$, la fonction de pression optimale

pour un intervalle, fait partie de $\vec{P}^*$, le vecteur contenant les fonctions de pression optimales pour l'épisode.

Si $\vec{C}_{obj}$ est réalisable, cela implique que,

$$\vec{p}_n{}^* = \underset{\vec{p}_n \in [0, q_{max}]^K}{\boldsymbol{arg\ min}} E_{est,n}(S_n, \vec{p}_n, C_{obj,n}) = \left(\underset{\vec{P} \in [0, q_{max}]^{NK}}{\boldsymbol{arg\ min}} E_{est,ep}(S_1, \vec{P}, \vec{C}_{obj}) \right)_n \quad \forall n \in 1, \ldots, N \tag{5.15}$$

C'est à dire que la stratégie optimale pour l'intervalle n calculée en ne considérant que l'intervalle n est aussi optimale considérant l'épisode entier si la séquence d'objectifs de puissance agrégée $\vec{C}_{obj}$ est réalisable.

Remarque : Il peut paraître contre-intuitif d'affirmer que la solution optimale peut être calculée intervalle par intervalle. En effet, une caractéristique essentielle de la commande optimale est que les coûts futurs peuvent venir moduler les actions passées en vue d'atteindre une optimalité globale.

Cependant, si la séquence d'objectifs de puissance agrégée est réalisable, cela implique qu'il existe une solution qui réduit l'erreur sur l'objectif de puissance à zéro sur tout l'épisode. Dans ce cas, il n'est pas nécessaire de balancer l'erreur instantanée et l'erreur à venir. La solution optimale sera donc indépendante de la longueur de l'intervalle considéré.

Donc, dans le cas où $\vec{C}_{obj}$ est réalisable, l'agrégateur peut simplement calculer la fonction de pression optimale pour l'intervalle n avec la méthode décrite dans la section 4.7. La fonction de pression résultante sera optimale en considérant l'épisode dans son intégralité.

5.4.2 Objectifs alternatifs réalisables

Sur la base de la notion de réalisabilité d'une séquence d'objectifs et des implications de la réalisabilité sur l'optimalité de la fonction de pression, on va modifier l'espace d'action A du MDP afin de réduire massivement la dimensionnalité de l'action.

Plutôt que d'être directement le choix d'une paramétrisation $\vec{p}_n$, l'action a_n à l'étape n du MDP sera le choix d'un objectif de puissance alternatif $C_{alt,n}$. La paramétrisation $\vec{p}^*_{alt,n}$ sera la paramétrisation qui minimise l'erreur estimée entre $\vec{C}_{est,n}$ et $C_{alt,n}$ sur l'intervalle n. On calcule $\vec{p}^*_{alt,n}$ avec la méthode de descente de gradient optimale pour un intervalle décrite à la section 4.7.

$$\vec{p}^{\,*}_{alt,n} = \underset{\vec{p}_n \in [0, q_{max}]^K}{\boldsymbol{arg\ min}} \; E_{est,n}(S_n, \vec{p}_n, C_{alt,n}) \tag{5.16}$$

Cependant, on impose la condition que l'objectif de puissance alternatif choisi dans un état h_n soit réalisable. Ce nouvel espace d'action $A(h_n)$ dépend donc maintenant de l'état de la population et contient l'ensemble des objectifs de puissance agrégée réalisables dans l'état h_n.

$$a_n = C_{alt,n} \;\in\; [\; C_{obj,min}(S_n) \;,\; C_{obj,max}(S_n) \;] \tag{5.17}$$

Ce nouvel espace d'action $A(h_n) \subset \mathbb{R}$ est beaucoup plus restreint que l'espace d'action précédent $A = [0, 1]^K$.

La séquence $\vec{C}_{alt}$ des N objectifs de puissance alternatifs pour l'épisode est réalisable. Donc, chacune des N fonctions de pression $\vec{p}^{\,*}_{alt,n}$ est optimale pour l'épisode, même si on calcule $\vec{p}^{\,*}_{alt,n}$ avec la méthode optimale développée pour un seul intervalle.

Si on définit A^N_{real} comme étant l'espace des séquences d'objectifs de puissance agrégée réalisables, on peut écrire que,

$$\vec{C}^{\,*}_{alt} = \underset{\vec{C}_{alt} \in A^N_{real}}{\boldsymbol{arg\ min}} \; E_{est,ep}(S_1, \vec{C}_{alt}) := \sum_{n=1}^{N} \sum_{k=1}^{K} \Big[\; C_{est,n,k}(\vec{C}_{alt}) - C_{obj,n} \;\Big]^2 \tag{5.18}$$

On remarque que ces objectifs de puissance alternatifs sont constants à l'intérieur de chaque intervalle. Cependant, la trajectoire de puissance agrégée optimale minimisant l'erreur sur l'épisode $E_{est,ep}$ n'est pas nécessairement constante à l'intérieur de chaque intervalle. Les fonctions de pression optimales $\vec{p}^{\,*}_{alt,n}$ calculées avec cet espace d'action alternatif ne seront donc pas parfaitement optimales. Mais, étant donné que les intervalles sont courts, on suppose que l'erreur sera acceptable.

5.5 Méthode de résolution du MDP

On a maintenant un MDP représentant l'approximation de la population de charges dont on connaît les fonctions de coût et de transition $\hat{R}$ et $\hat{G}$. On présente deux approches possibles pour résoudre un problème de ce type.

5.5.1 Méthode d'apprentissage par renforcement basée sur un modèle

Les méthodes d'apprentissage par renforcement basées sur un modèle (MBRL) cherchent à résoudre les MDPs dont les fonctions de transition et de coût sont approximées par un modèle de la dynamique du système. Dans notre cas, notre modèle de la dynamique du système est l'ensemble des réseaux de neurones de l'agrégateur.

Les méthodes de MBRL utilisent ce modèle de la dynamique du système pour simuler le comportement du système dans des situations hypothétiques, puis utilisent ces simulations avec une méthode d'apprentissage par renforcement pour minimiser le coût du MDP. Par exemple, une méthode de MBRL classique est Dyna-Q [23]. Dyna-Q utilise un modèle de la dynamique du système qui est appris. Utilisant ce modèle pour simuler le système, Dyna-Q utilise le Q-learning pour apprendre la stratégie optimale et résoudre le MDP.

Pour certains MDPs, apprendre un modèle de la dynamique du système est facile et demande une quantité limitée d'expérimentations sur le vrai système. Cependant, apprendre la stratégie optimale pour minimiser le coût du MDP peut être beaucoup plus compliqué, même si la dynamique du système est simple.

C'est le cas de notre problème dans ce mémoire. Il est possible pour un ensemble de réseaux de neurones d'approximer la dynamique de l'effort d'une population de charges thermostatiques avec un apprentissage d'une durée limitée. On pourrait ensuite utiliser ce modèle pour entraîner un autre réseau de neurones chargé d'apprendre la valeur de Q selon les paires état-action.

5.5.2 Algorithme récursif d'exploration des états possibles

Il est possible d'utiliser Dyna-Q ou un autre algorithme de MBRL pour résoudre notre MDP. Cependant, étant donné qu'on a réduit l'espace d'action à une valeur scalaire, on choisit de résoudre explicitement l'équation de Bellman pour obtenir la stratégie optimale du MDP.

$$v^*(h_n) = \underset{a_n \in A}{\boldsymbol{min}} \quad \hat{R}(h_n, a_n) + v^*(\hat{G}(h_n, a_n)) \tag{5.19}$$

Pour ce faire, on va discrétiser l'espace d'action $A(h_n)$ en $A_{disc}(h_n)$. On rappelle que l'action à l'étape n du MDP est le choix de l'objectif de puissance alternatif réalisable $C_{alt,n}$.

Cependant, l'espace d'état H est continu et hautement dimensionnel. Il n'est donc pas possible de discrétiser l'espace d'état et de calculer la valeur optimale $v^*(h_n)$ pour chaque état possible. Plutôt, on va seulement explorer les états qui sont atteignables à partir de l'état initial de la population h_1 au début du MDP. C'est-à-dire, on va explorer toutes les séquences

d'actions possibles allant du début jusqu'à la fin de l'épisode.

Pour explorer tous les états atteignables, on utilise un algorithme récursif qui reçoit un état h_n et va ensuite explorer toutes les actions possibles $a_n \in A_{disc}(h_n)$. À partir de l'état h_n et d'une action a_n, on utilise les réseaux de neurones pour approximer le coût sur l'intervalle ainsi que l'état suivant de la population h_{n+1}. Finalement, on continue l'algorithme récursif à partir de l'état h_{n+1}. Une fois que tous les états atteignables depuis h_1 ont été explorés, la stratégie optimale est simplement de choisir la séquence d'actions pour laquelle le coût total sur l'épisode est minimisé.

En pratique, l'agrégateur ne connaît pas l'espace des objectifs alternatifs réalisables dans un état h_n donné de la population de charges. Afin de déterminer si un objectif alternatif $C_{alt,n}$ est réalisable, l'agrégateur peut simplement calculer la fonction de pression optimale $\vec{p}^{\,*}_{alt,n}$ et la puissance agrégée estimée $\vec{C}_{est,n}$ résultante. Finalement, si l'erreur estimée sur l'intervalle est plus petite qu'une tolérance ϵ, l'agrégateur considère que cet objectif de puissance alternatif $C_{alt,n}$ est réalisable.

$$E_{est,n}(S_n, C_{alt,n}) := \sum_{k=1}^{K} [\ C_{est,n,k} - C_{alt,n}\ ]^2 \tag{5.20}$$

Cependant, si l'objectif de puissance alternatif $C_{alt,n}$ n'est pas réalisable, l'agrégateur va simplement ignorer cet objectif et tester l'objectif alternatif suivant.

5.6 Durée des intervalles

Le choix de la durée Δt des intervalles est très important. En effet, la valeur de Δt a un impact sur la performance et la polyvalence de la méthode de contrôle développée dans ce mémoire.

Lors de l'apprentissage, l'agrégateur observe la puissance consommée par les charges de la population en réponse à la fonction de pression à chaque intervalle. Un plus grand nombre d'intervalles de plus courte durée permet à l'agrégateur d'obtenir plus d'observations plus rapidement. Donc, plus la valeur de Δt est petite, plus le nombre d'intervalles est grand et plus l'apprentissage des réseaux de neurones est rapide.

Une autre raison de choisir une petite valeur de Δt est que cela permet à l'agrégateur de réagir plus rapidement aux besoins du réseau. Pour changer l'objectif de puissance agrégée C_{obj}, l'agrégateur doit attendre le début du prochain intervalle car l'agrégateur a besoin de mesurer la puissance consommée au dernier moment d'un intervalle $c^i_{n,K}$ afin de calculer l'état de la charge i.

Cependant, la complexité du calcul de la fonction de pression optimale sur un épisode de N intervalles est exponentielle en N. Le nombre maximum d'intervalles dans un épisode est donc limité. Donc, la durée totale sur laquelle l'agrégateur peut calculer la fonction de pression optimale est égale à $\Delta t * N$.

5.7 Simulation de commande optimale sur l'intervalle

Dans cette section, la méthodologie utilisée pour simuler le comportement de la population est la même que dans la section 4.9. Cependant, pour cette section, la population contient seulement 3 charges afin de faciliter le calcul. L'apprentissage des réseaux de neurones est terminé.

5.7.1 Stratégie non optimale

La figure 5.2 présente le comportement de la population sur un épisode contenant 4 intervalles ayant chacun un objectif de puissance agrégée égal à 5% de C_{ideal}. Comme dans la section 5.1, les charges ne peuvent pas suivre l'objectif de puissance agrégée durant tout l'épisode, ce qui cause une grande erreur quadratique dans les derniers intervalles.

5.7.2 Stratégie alternative optimisée

La figure 5.3 montre le résultat d'une simulation similaire à la figure 5.2. Cependant, avant le début de l'épisode, l'agrégateur va utiliser l'algorithme récursif d'exploration des états possibles présenté à la section 5.5.2 pour calculer la séquence optimale d'objectifs alternatifs réalisables $\vec{C}^*_{alt}$. Durant l'épisode, l'agrégateur va calculer la fonction de pression optimale par intervalle qui minimise l'erreur avec $Calt, n$ et non $C_{obj,n}$. La ligne verte dans le graphique en haut à droite représente la séquence optimale d'objectifs alternatifs réalisables $\vec{C}^*_{alt}$ calculée par l'agrégateur avant le début de l'épisode.

On observe que la population répartit sa puissance de chauffage plus également sur l'épisode afin de réduire l'erreur quadratique entre la puissance agrégée et l'objectif de puissance. De plus, étant donné que la séquence d'objectifs alternatifs est réalisable, la puissance agrégée de la population est capable d'atteindre les objectifs de consommation alternatifs sur tout l'épisode, sauf pour une petite déviation à la fin de l'épisode, due au bruit et à l'imprécision des réseaux de neurones. On arrive ainsi à réutiliser les algorithmes de commande optimale sur intervalle sans perte d'optimalité sur l'épisode complet.

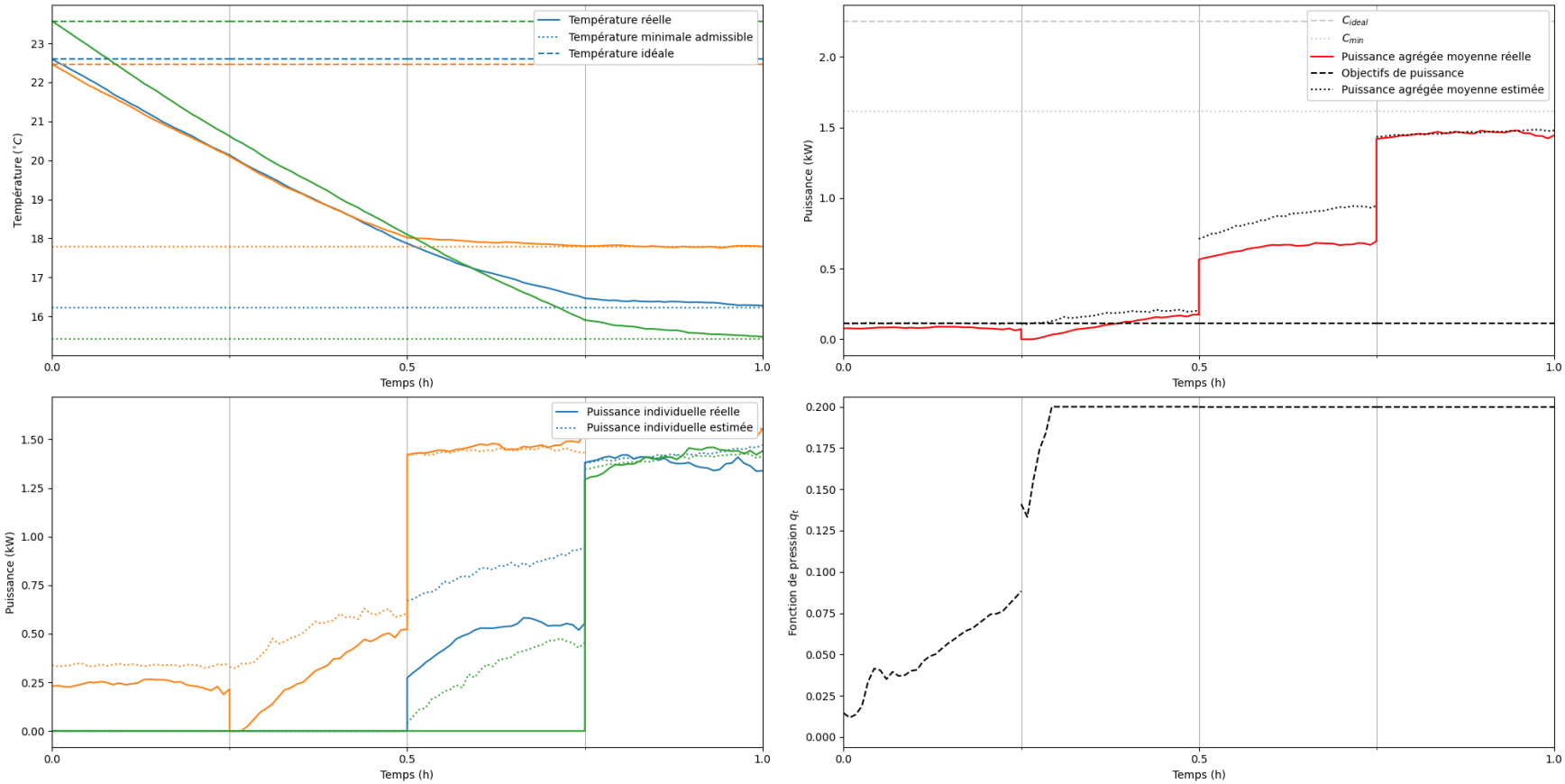

FIGURE 5.2 Commande non optimale due à la non faisabilité de l'objectif

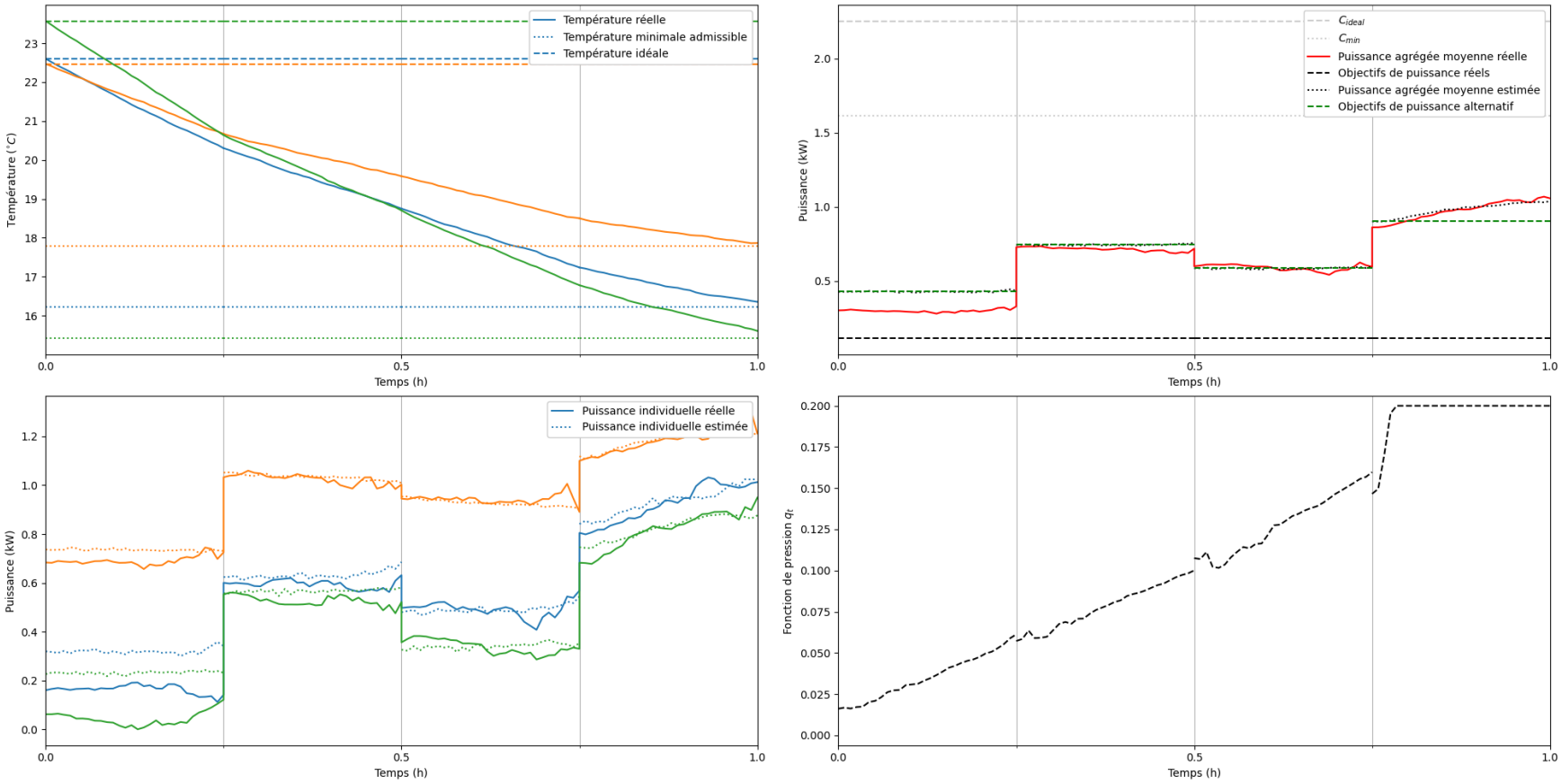

FIGURE 5.3 Commande alternative optimisée basée sur un rééquilibrage des erreurs dues à la non faisabilité de l'objectif

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

L'objectif de ce mémoire était d'élaborer une méthode pour la commande décentralisée de charges thermostatiques. On désirait obtenir une méthode pouvant permettre l'ajustement rapide de la consommation de la population de charges en réponse aux besoins du réseau. De plus, on désirait que cette méthode puisse fonctionner avec une population hétérogène de charges sans avoir besoin de connaître au préalable les paramètres thermiques des charges. Finalement, par souci de confidentialité et afin de limiter la quantité d'informations échangées entre les charges et l'agrégateur, on désirait que l'agrégateur n'ait à aucun moment besoin de connaître les températures intérieures et extérieures des charges.

Premièrement, on a établi les équations modélisant la dynamique des charges ainsi que le mécanisme par lequel l'agrégateur peut influencer la consommation énergétique de celles-ci. Les charges calculent leur consommation de façon décentralisée en résolvant un problème de commande optimale local. À chaque intervalle de contrôle, l'agrégateur dicte une fonction de pression à l'ensemble de la population. Cette fonction de pression, uniforme pour toutes les charges de la population, modifie la fonction de coût minimisée par les charges et permet donc à l'agrégateur d'influencer la consommation énergétique agrégée de la population. Ensuite, on a pu établir que la température d'une charge thermostatique à la fin d'un intervalle de contrôle est fonction de l'effort et de la valeur de la fonction de pression à la fin de l'intervalle. Cette fonction n'est cependant pas connue par l'agrégateur puisqu'il ne connaît pas les paramètres thermiques des charges.

On a ensuite élaboré une méthode basée sur l'apprentissage automatique pour apprendre la dynamique thermique de charges thermostatiques en réponse à une fonction de pression. L'agrégateur entraîne un réseau neuronal différent pour chaque charge de la population afin d'estimer l'effort des charges en réponse à différentes fonctions de pression. Ce réseau de neurones remplit deux rôles essentiels. Étant donné que l'agrégateur ne mesure pas les températures des charges, le réseau doit reconstruire l'état de la charge à partir de l'effort mesuré et de la valeur de la fonction de pression à l'intervalle précédent. De plus, le réseau doit estimer l'effort en réponse à une fonction de pression, ce qui dépend de l'état de la charge. Le réseau de neurones internalise donc le calcul de l'état et le calcul de l'effort en fonction de l'état.

Utilisant ces réseaux neuronaux, on a élaboré une méthode pour calculer la fonction de pres-

sion optimale permettant de minimiser la différence carrée entre la consommation agrégée de la population et l'objectif de consommation de l'agrégateur sur un intervalle. Cette méthode utilise une heuristique basée sur la connaissance du mécanisme par lequel les charges calculent leur effort sur un intervalle pour approximer la dérivée partielle de l'effort des charges en fonction de la fonction de pression. À l'aide de cette heuristique, l'agrégateur peut utiliser l'algorithme de descente de gradient pour calculer la fonction de pression optimale.

Finalement, on a élaboré une méthode permettant d'obtenir la séquence de fonctions de pression optimales permettant à l'agrégateur de minimiser la différence carrée entre la consommation agrégée de la population et l'objectif de consommation de l'agrégateur sur un épisode composé de multiples intervalles. La méthode optimale sur un intervalle permet de réduire l'état d'action du MDP correspondant afin de permettre une recherche récursive sur tous les états atteignables.

6.2 Limitations de la solution proposée et améliorations futures

La méthode développée dans ce mémoire possède certains désavantages et contraintes qui limitent son utilité. Certaines de ces limitations pourraient cependant être corrigées en améliorant la méthode développée dans ce mémoire dans un travail futur.

6.2.1 Température extérieure variable

La méthode proposée suppose que y^i, la température extérieure à chaque charge, reste constante sur tout l'épisode. En réalité, la température extérieure va changer et les charges vont adapter leurs valeurs de u^i_{ideal} pour contrer le changement dans les pertes dues à l'environnement.

Cependant, l'agrégateur ne possède pas de mécanisme pour ajuster sa valeur estimée $u^i_{est,ideal}$ durant l'épisode. L'agrégateur mesure u^i_{ideal} au début de l'épisode et utilise cette même valeur pour l'intégralité de l'épisode. Donc, plus l'épisode progresse, plus la température extérieure va changer et plus la différence entre u^i_{ideal} et $u^i_{est,ideal}$ va augmenter. En pratique, cela limite la durée possible des épisodes, ce qui limite la capacité de l'agrégateur à influencer la consommation de la population.

6.2.2 Calcul de la fonction de pression optimale pour l'épisode

Comme mentionné précédemment, la complexité de la méthode développée pour le calcul de la fonction de pression optimale sur un épisode de N intervalles est exponentielle en N. En

pratique, cela impose une limite sur le nombre maximum d'intervalles dans un épisode.

Si l'agrégateur désire optimiser pour des épisodes contenant un grand nombre d'intervalles, une solution possible serait d'utiliser une méthode de MBRL, tel que mentionné à la section 5.5.1. La complexité du calcul de l'action à l'aide d'une méthode de MBRL ne dépend pas du nombre d'intervalles dans l'épisode et ne va donc pas limiter la durée maximale de celui-ci.

6.2.3 Apprentissage par transfert

Lors de l'apprentissage des fonctions $\vec{f}^{\,i}_{neural}$ pour chacune des charges de la population, l'agrégateur ne peut pas atteindre ses objectifs de consommation de façon satisfaisante. Il serait bénéfique de pouvoir réduire la durée de cette phase d'apprentissage pour permettre à l'agrégateur de contrôler la population plus précisément.

Une amélioration possible pour accélérer l'apprentissage serait l'utilisation de l'apprentissage par transfert [24]. L'agrégateur pourrait simuler le comportement de différentes charges fictives représentant différentes catégories de charges thermostatiques et entraîner des réseaux de neurones pour apprendre la dynamique de ces charges fictives. Par exemple, l'agrégateur pourrait entraîner un ensemble de réseaux de neurones correspondant à différents types de charges : habitations, espaces commerciaux, entrepôts, etc.

Au début de la phase d'apprentissage, plutôt que d'initialiser aléatoirement les paramètres internes du réseau de neurones chargé d'approximer la dynamique d'une charge, l'agrégateur pourrait initialiser le réseau avec les paramètres internes du réseau ayant été entraîné à approximer la charge fictive de la catégorie similaire à la charge.

6.2.4 Architecture alternative pour les réseaux neuronaux

On a choisi d'utiliser le MLP, qui est une architecture très simple de réseau neuronal, afin de ne pas introduire de complexité supplémentaire dans cette portion de la méthodologie. Cependant, il pourrait être bénéfique d'utiliser une architecture plus performante pour l'approximation de fonction. Dans [25], les auteurs introduisent les réseaux de neurones Kolmogorov-Arnold (KAN). Les KANs apprennent les fonctions d'activation des neurones plutôt que les valeurs des poids et biais connectant les neurones de différentes couches. Dans l'article, les auteurs démontrent que cette architecture peut approximer la même fonction avec moins de paramètres et une meilleure précision qu'un MLP.

RÉFÉRENCES

[1] H. Ritchie et P. Rosado, "Electricity mix," *Our World in Data*, 2020, https ://ourworldindata.org/electricity-mix.

[2] Lazard, "Lazard's levelized cost of energy analysis—version 17.0," 2024. [En ligne]. Disponible : https://www.lazard.com/research-insights/levelized-cost-of-energyplus/

[3] J. Tastu, P. Pinson, P.-J. Trombe et H. Madsen, "Probabilistic forecasts of wind power generation accounting for geographically dispersed information," *IEEE Transactions on Smart Grid*, vol. 5, nº. 1, p. 480–489, 2014.

[4] Hydro-Québec, "Breakdown of a household's electricity use," 2025. [En ligne]. Disponible : https://www.hydroquebec.com/residential/customer-space/electricity-use/electricity-consumption-by-use.html

[5] F. Ruelens, B. J. Claessens, P. Vrancx, F. Spiessens et G. Deconinck, "Direct load control of thermostatically controlled loads based on sparse observations using deep reinforcement learning," *CSEE Journal of Power and Energy Systems*, vol. 5, nº. 4, p. 423–432, 2019.

[6] P. Siano, "Demand response and smart grids—a survey," *Renewable and Sustainable Energy Reviews*, vol. 30, p. 461–478, 2014. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1364032113007211

[7] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez et S. Wattam, "Artificial intelligence and machine learning approaches to energy demand-side response : A systematic review," *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109899, 2020. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S136403212030191X

[8] E. C. Kara, M. Berges, B. Krogh et S. Kar, "Using smart devices for system-level management and control in the smart grid : A reinforcement learning framework," dans *Proceedings of IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012, p. 85–90.

[9] J. L. Mathieu, S. Koch et D. S. Callaway, "State estimation and control of electric loads to manage real-time energy imbalance," *IEEE Transactions on Power Systems*, vol. 28, nº. 1, p. 430–440, 2013.

[10] A. Coffman, A. Bušić et P. Barooah, "A unified framework for coordination of thermostatically controlled loads," *Automatica*, vol. 152, p. 111002, 2023. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0005109823001553

[11] D. S. Callaway, “Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy,” *Energy Conversion and Management*, vol. 50, nº. 5, p. 1389–1400, 2009. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0196890408004780

[12] V. Mai, P. Maisonneuve, T. Zhang, H. Nekoei, L. Paull et A. Lesage-Landry, “Multi-agent reinforcement learning for fast-timescale demand response of residential loads,” *Machine Learning*, vol. 113, nº. 8, p. 5203–5234, Aug 2024. [En ligne]. Disponible : https://doi.org/10.1007/s10994-023-06460-4

[13] Q. Lénet, M. S. Nazir et R. P. Malhamé, “An inverse nash mean field game-based strategy for the decentralized control of thermostatic loads,” dans *Proceedings of 60th IEEE Conference on Decision and Control (CDC)*, 2021, p. 4929–4935.

[14] A. C. Kizilkale, R. Salhab et R. P. Malhamé, “An integral control formulation of mean field game based large scale coordination of loads in smart grids,” *Automatica*, vol. 100, p. 312–322, 2019. [En ligne]. Disponible : https://publications.polymtl.ca/41901/

[15] J. F. Nash, “Equilibrium points in n-person games,” *Proceedings of the National Academy of Sciences*, vol. 36, nº. 1, p. 48–49, 1950. [En ligne]. Disponible : https://www.pnas.org/doi/abs/10.1073/pnas.36.1.48

[16] R. C. Sonderegger, “Dynamic models of house heating based on equivalent thermal parameters,” Thèse de doctorat, Princeton University, New Jersey, déc. 1978.

[17] R. E. Bellman et S. E. Dreyfus, *Applied Dynamic Programming*. Princeton : Princeton University Press, 1962. [En ligne]. Disponible : https://doi.org/10.1515/9781400874651

[18] M. Leshno, V. Y. Lin, A. Pinkus et S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural Networks*, vol. 6, nº. 6, p. 861–867, 1993. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0893608005801315

[19] Y. A. LeCun, L. Bottou, G. B. Orr et K.-R. Müller, *Efficient BackProp*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, p. 9–48. [En ligne]. Disponible : https://doi.org/10.1007/978-3-642-35289-8_3

[20] L.-J. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” *Machine Learning*, vol. 8, nº. 3–4, p. 293–321, mai 1992. [En ligne]. Disponible : https://doi.org/10.1007/BF00992699

[21] T. Schaul, J. Quan, I. Antonoglou et D. Silver, “Prioritized experience replay,” dans *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio et Y. LeCun, édit., 2016. [En ligne]. Disponible : http://arxiv.org/abs/1511.05952

[22] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke et S. Levine, “Scalable deep reinforcement learning for vision-based robotic manipulation,” dans *Proceedings of The 2nd Conference on Robot Learning*, A. Billard, A. Dragan, J. Peters et J. Morimoto, édit., vol. 87. PMLR, 29–31 Oct 2018, p. 651–673. [En ligne]. Disponible : https://proceedings.mlr.press/v87/kalashnikov18a.html

[23] R. S. Sutton, “Dyna, an integrated architecture for learning, planning, and reacting,” *SIGART Bulletin*, vol. 2, n°. 4, p. 160–163, juill. 1991. [En ligne]. Disponible : https://doi.org/10.1145/122344.122377

[24] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong et Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, n°. 1, p. 43–76, 2021.

[25] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou et M. Tegmark, “KAN : Kolmogorov–arnold networks,” dans *The Thirteenth International Conference on Learning Representations*, 2025. [En ligne]. Disponible : https://openreview.net/forum?id=Ozo7qJ5vZi

ANNEXE A RESPECT DE LA TEMPÉRATURE MINIMALE ADMISSIBLE

Afin de prouver que la température de la charge va généralement se maintenir à une température supérieure ou égale à z_{min}, on doit démontrer que si une charge thermostatique atteint exactement sa température minimale acceptable z_{min}, l'effort optimal obtenu par la résolution du HJB doit toujours être suffisamment grand pour que l'espérance du changement de la température de la charge soit ≥ 0.

$$\left(E\left[\frac{dx_t}{dt}\right] \;\middle|\; x_t = z_{min} \right) \geq 0 \tag{A.1}$$

Remplaçant x_t par z_{min} dans l'équation de la dynamique thermique de la charge, on obtient,

$$-a\left(z_{min} - z_{ideal}\right) + b\left(u_t\right) \geq 0 \tag{A.2}$$

$$u_t \geq \frac{a}{b}\left(z_{min} - z_{ideal}\right) \tag{A.3}$$

Insérant cette contrainte sur l'effort dans l'équation décrivant l'effort optimal obtenu par la résolution du HJB, on obtient,

$$u_t^* = -\frac{b}{r}\left(\pi_t\left(z_{min} - z_{min}\right) + \beta_t\right) \geq \frac{a}{b}\left(z_{min} - z_{ideal}\right) \tag{A.4}$$

$$\beta_t \leq \frac{ar(z_{ideal} - z_{min})}{b^2} \tag{A.5}$$

Finalement, β_t dépend de π_t qui dépend de q_t. Pour une valeur de q_t constante, β_t est égale à,

$$\beta_t = \frac{(a\pi_t - q_{ideal})(z_{ideal} - z_{min})}{a + \pi_t \frac{b^2}{r}} \tag{A.6}$$

Étant donné que $\frac{d\beta_t}{d\pi_t} > 0$ et $\frac{d\pi_t}{dq_t} > 0$, on sait que $\frac{d\beta_t}{dq_t} > 0$. Si on prend la limite quand q_t tend vers l'infini, on obtient,

$$\lim_{q_t \to +\infty} \pi_t = +\infty \tag{A.7}$$

$$\lim_{q_t \to +\infty} \beta_t = \frac{ar(z_{ideal} - z_{min})}{b^2} \tag{A.8}$$

Cela prouve que peu importe la valeur de q_t, β_t va toujours satisfaire la contrainte nécessaire à assurer que la température de la charge ne descende pas significativement en dessous de la température minimale acceptable.

ANNEXE B RÉSOLUTION DE L'ÉQUATION DE HJB

On cherche à calculer la commande optimale u_t^* pour une charge thermostatique possédant la dynamique et la fonction de coût à minimiser suivantes,

$$dx_t = \left[-a\left(x_t - z_{ideal}\right) + b\left(u_t\right)\right] dt + \sigma dw_t \tag{B.1}$$

$$J\left(x_0, u, q\right) = E\left[\int_0^T \left[\frac{q_t}{2}\left(x_t - z_{min}\right)^2 + \frac{q_{ideal}}{2}\left(x_t - z_{ideal}\right)^2 + \frac{r}{2}\left(u_t\right)^2\right] dt\right] \tag{B.2}$$

Premièrement, on remplace x par $s = x - z_{min}$ ce qui nous donne les équations suivantes,

$$ds_t = \left[-a\left(s_t + z_{min} - z_{ideal}\right) + b\left(u_t\right)\right] dt + \sigma dw_t \tag{B.3}$$

$$J\left(x_0, u, q\right) = E\left[\int_0^T \left[\frac{q_t}{2}\left(s_t\right)^2 + \frac{q_{ideal}}{2}\left(s_t + z_{min} - z_{ideal}\right)^2 + \frac{r}{2}\left(u_t\right)^2\right] dt\right] \tag{B.4}$$

On teste une forme quadratique pour le coût optimal J^* dont on obtient les dérivées partielles suivantes :

$$J^* = \frac{1}{2}\pi s^2 + \beta s + \gamma \tag{B.5}$$

$$J_t^* = \frac{1}{2}\dot{\pi} s^2 + \dot{\beta} s + \dot{\gamma} \tag{B.6}$$

$$J_s^* = \pi s + \beta \tag{B.7}$$

On obtient ensuite le Hamiltonien suivant,

$$\begin{aligned} H\left(s_t, u_t, q_t\right) = \frac{q_t}{2}\left(s_t\right)^2 + \frac{q_{ideal}}{2}\left(s_t + z_{min} - z_{ideal}\right)^2 + \frac{r}{2}\left(u_t\right)^2 \\ + \left[\pi s + \beta\right] * \left[-a\left(s_t + z_{min} - z_{ideal}\right) + b\left(u_t\right)\right] \end{aligned} \tag{B.8}$$

On minimise le Hamiltonien par rapport à u_t,

$$\frac{dH}{du_t} = 0 \tag{B.9}$$

$$u_t^* = -\frac{b}{r}\left(\pi_t s_t + \beta_t\right) \tag{B.10}$$

Ayant tous les termes nécessaires, on évalue maintenant l'équation de Hamilton-Jacobi-Bellman,

$$-J_t^* = H\left(s_t, u_t^*, q_t\right) \tag{B.11}$$

$$\begin{gathered} -[\frac{1}{2}\dot{\pi}s^2 + \dot{\beta}s + \dot{\gamma}] = \\ \frac{q_t}{2}\left(s_t\right)^2 + \frac{q_{ideal}}{2}\left(s_t + z_{min} - z_{ideal}\right)^2 + \frac{r}{2}\left(-\frac{b}{r}\left(\pi_t s_t + \beta_t\right)\right)^2 \\ + \left[\pi s + \beta\right] * \left[-a\left(s_t + z_{min} - z_{ideal}\right) + b\left(-\frac{b}{r}\left(\pi_t s_t + \beta_t\right)\right)\right] \end{gathered} \tag{B.12}$$

$$\begin{gathered} -[\frac{1}{2}\dot{\pi}s^2 + \dot{\beta}s + \dot{\gamma}] = \\ [\frac{q_t}{2} + \frac{q_{ideal}}{2} - \frac{(b)^2\pi^2}{2r} - \pi a]s^2 + \\ [q_{ideal}z_{min} - q_{ideal}z_{ideal} - a\pi z_{min} + a\pi z_{ideal} - \frac{(b)^2}{r}\pi\beta - a\beta]s \\ +[...] \end{gathered} \tag{B.13}$$

Le terme γ n'apparaît pas dans la commande optimale, on peut donc l'ignorer. De cette équation, on obtient les équations différentielles suivantes,

$$\dot{\pi_t} = \frac{(b)^2}{r}\left(\pi_t\right)^2 + 2a\pi_t - q_{ideal} - q_t \tag{B.14}$$

$$\dot{\beta_t} = \left(a + \frac{(b)^2}{r}\pi_t\right)\beta_t - \left(a\pi_t - q_{ideal}\right)\left(z_{ideal} - z_{min}\right) \tag{B.15}$$

En remplaçant s par x on obtient finalement la loi de commande optimale suivante,

$$u_t^* = -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) \tag{B.16}$$

Équation du coût final

Dans la section 3.3, nous avons calculé les valeurs de π_T et β_T pour obtenir une puissance constante à T.

$$\pi_T = \frac{q_T + q_{ideal}}{a} \tag{B.17}$$

$$\beta_T = -\frac{q_{ideal}\left(z_{ideal} - z_{min}\right)}{a} \tag{B.18}$$

Nous allons maintenant démontrer que ces valeurs de π_T et β_T sont obtenues avec le coût final suivant :

$$D(x_T) = \frac{q_T + q_{ideal}}{a}\left(x_T - z_{min}\right)^2 - \frac{q_{ideal}(z_{ideal} - z_{min})}{a}\left(x_T - z_{min}\right) \tag{B.19}$$

Pour résoudre l'équation de HJB, nous avions remplacé x_t par $s_t + z_{min}$. Le coût final en fonction de s_T est alors :

$$D(s_T) = \frac{q_T + q_{ideal}}{a}s_T^2 - \frac{q_{ideal}(z_{ideal} - z_{min})}{a}s_T \tag{B.20}$$

Le coût optimal que nous avons utilisé lors de la résolution de l'équation de HJB était la suivante :

$$J^* = \frac{1}{2}\pi s^2 + \beta s + \gamma \tag{B.21}$$

Ce qui donne par identification :

$$\pi_T = \frac{q_T + q_{ideal}}{a} \tag{B.22}$$

$$\beta_T = -\frac{q_{ideal}\left(z_{ideal} - z_{min}\right)}{a} \tag{B.23}$$

ANNEXE C ANALYSE DE L'HEURISTIQUE DÉCRIVANT LA VARIATION DE L'EFFORT EN FONCTION DE LA PRESSION

Comme décrit précédemment, les charges calculent leur effort selon la fonction suivante,

$$u_t = -\frac{b}{r}\left(\pi_t\left(x_t - z_{min}\right) + \beta_t\right) \tag{C.1}$$

π_t et β_t sont calculés numériquement à temps inverse selon les équations de Riccati suivantes,

$$\dot{\pi_t} = \frac{b^2}{r}\left(\pi_t\right)^2 + 2a\pi_t - q_{ideal} - q_t \tag{C.2}$$

$$\dot{\beta_t} = \left(a + \frac{b^2}{r}\pi_t\right)\beta_t - \left(a\pi_t - q_{ideal}\right)\left(z_{ideal} - z_{min}\right) \tag{C.3}$$

Dans l'équation de Riccati pour π_t, on observe que,

$$\frac{d\dot{\pi}_t}{dq_t} < 0 \tag{C.4}$$

Étant donné que π_t est résolu à temps inverse, cela implique que,

$$\frac{\partial \pi_{t_1}}{\partial q_{t2}} \geq 0 \ , \ t1 < t2 \tag{C.5}$$

Similairement, dans l'équation de Riccati pour β_t, on observe que,

$$\frac{d\dot{\beta}_t}{d\pi_t} < 0 \tag{C.6}$$

Cela implique que,

$$\frac{\partial \beta_{t_1}}{\partial q_{t2}} \geq 0 \ , \ t1 < t2 \tag{C.7}$$

Finalement, on évalue l'impact sur l'effort,

$$\frac{\partial u_{t_1}}{\partial q_{t2}} = \frac{\partial u_{t_1}}{\partial \pi_{t_1}}\frac{\partial \pi_{t_1}}{\partial q_{t2}} + \frac{\partial u_{t_1}}{\partial \beta_{t_1}}\frac{\partial \beta_{t_1}}{\partial q_{t2}} + \frac{\partial u_{t_1}}{\partial x_{t_1}}\frac{\partial x_{t_1}}{\partial q_{t2}} \tag{C.8}$$

On fait l'hypothèse que l'impact du terme, $\frac{\partial u_{t_1}}{\partial x_{t_1}} \frac{\partial x_{t_1}}{\partial q_{t2}}$ est moins important, donc on l'ignore. On obtient alors,

$$\frac{\partial u_{t_1}}{\partial q_{t2}} \approx -\frac{b}{r}(x_t - z_{min}) \frac{\partial \pi_{t_1}}{\partial q_{t2}} - \frac{b}{r} \frac{\partial \beta_{t_1}}{\partial q_{t2}} \tag{C.9}$$

$$\frac{\partial u_{t_1}}{\partial q_{t_2}} \approx \begin{cases} \leq 0, & \text{si } t_1 \leq t_2 \\ = 0, & \text{si } t_1 > t_2 \end{cases} \tag{C.10}$$

Si on s'intéresse maintenant à la discrétisation de l'effort sur l'intervalle selon les composantes de la paramétrisation $\vec{p}$, on obtient,

$$\frac{\partial u_k}{\partial p_j} \approx \begin{cases} \leq 0, & \text{si } k \leq j \\ = 0, & \text{si } k > j \end{cases} \tag{C.11}$$

Étant donné qu'on ne peut pas calculer la magnitude de ces dérivées partielles, on l'approxime par une constante $D < 0$ inconnue mais strictement négative. Cette constante négative représente le fait que la valeur de la fonction de pression à l'instant t d'un intervalle va influencer négativement l'effort entre 0 et t.

De plus, on suppose que les impacts que les composantes de $\vec{p}_n$ ont sur l'effort $\vec{u}_n$ sont dans un même ordre de grandeur. On multiplie donc la constante D^i par $1/j$ afin de normaliser l'impact total de chaque composante de $\vec{p}_n$ sur l'erreur $E_{est,n}$.

On obtient alors la forme suivante pour la dérivée partielle de l'effort à k selon la fonction de pression à j,

$$\frac{\partial u_k}{\partial p_j} \approx \begin{cases} \frac{1}{j} D, & \text{si } k \leq j \\ 0, & \text{si } k > j \end{cases} \tag{C.12}$$