

LYCÉE LA MARTINIÈRE MONPLAISIR LYON

SCIENCES INDUSTRIELLES POUR L'INGÉNIEUR

CLASSE PRÉPARATOIRE P.S.I.

ANNÉE 2023 - 2024



C8 : RÉOLUTION DES MODÈLES PAR L'UTILISATION DE L'INTELLIGENCE ARTIFICIELLE

## TD 20 - Apprentissage supervisé et Machine Learning et apprentissage profond (C8-1)

### Compétences

- **Analyser**
  - Analyser un algorithme.
  - Analyser les principes d'intelligence artificielle.
  - Extraire un indicateur de performance pertinent à partir du cahier des charges ou de résultats issus de l'expérimentation ou de la simulation.
  - Caractériser les écarts entre les performances.
  - Interpréter et vérifier la cohérence des résultats obtenus expérimentalement, analytiquement ou numériquement.
  - Rechercher et proposer des causes aux écarts constatés.
- **Résoudre**
  - Choisir une démarche de résolution d'un problème d'ingénierie numérique ou d'intelligence artificielle.
  - Résoudre un problème en utilisant une solution d'intelligence artificielle.
- **Communiquer**
  - Trier les informations selon des critères.
- **Concevoir**
  - Modifier la commande pour faire évoluer le comportement du système.

## Exercice 1 : Apprentissage automatique du robot Alphai

Source : Emilien DURIF

## 1 Présentation

On s'intéresse ici au robot **Alphai** développé pour la société **learnin-grobots** qui permet d'appréhender les enjeux de l'intelligence artificielle et du **Machine Learning** sur la robotique.

On peut utiliser ce robot pour lui faire apprendre par exemple à faire le plus grand parcours dans une enceinte sans toucher les murs par les méthode d'apprentissage supervisé selon :

- Recherche de plus proche voisin : machine learning
- Réseau de neurones : apprentissage profond



Le scénario d'apprentissage est le suivant :

On dispose d'une enceinte et d'un mur central. On souhaite que le robot alphai fasse le plus de tour consécutif du mur central en le moins de temps possible.



Pour cela on dispose de 3 actions :

Rotation sur place à gauche    Déplacement tout droit    Rotation sur place à droite



On dispose de données d'entrée provenant de capteurs :

- caméra : jusque 640x380 pixels
- capteur de robot bloqué

## 2 Apprentissage supervisé par recherche du plus proche voisin

**Q 1 : Mettre en oeuvre l'apprentissage du robot alphai par l'apprentissage supervisé par recherche du plus court chemin en suivant la documentation en annexe.**

Exporter les données d'apprentissage dans le menu IA/sauver la mémoire d'expérience. On pourra exporter les données sous le nom *alphai\_knn.buffer.npz*.

**Q 2 : Exploiter les données d'apprentissage à l'aide du script *traiter\_alphai.py* qui doit être sauvegardé dans le même répertoire que *alphai\_knn.buffer.npz*.**

**Q 3 : Faire varier le nombre de plus proches voisins et observer les résultats sur la matrice de confusion, le score d'apprentissage et la représentation graphique des plus proches voisins.**

**Q 4 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

## 3 Apprentissage supervisé par réseau de neurones

**Q 5 : Mettre en oeuvre l'apprentissage du robot alphai par l'apprentissage supervisé réseau de neurones en suivant la documentation en annexe.**

**Q 6 : Faire varier le nombre de neurones, de couches et de données d'entrées.**

**Q 7 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

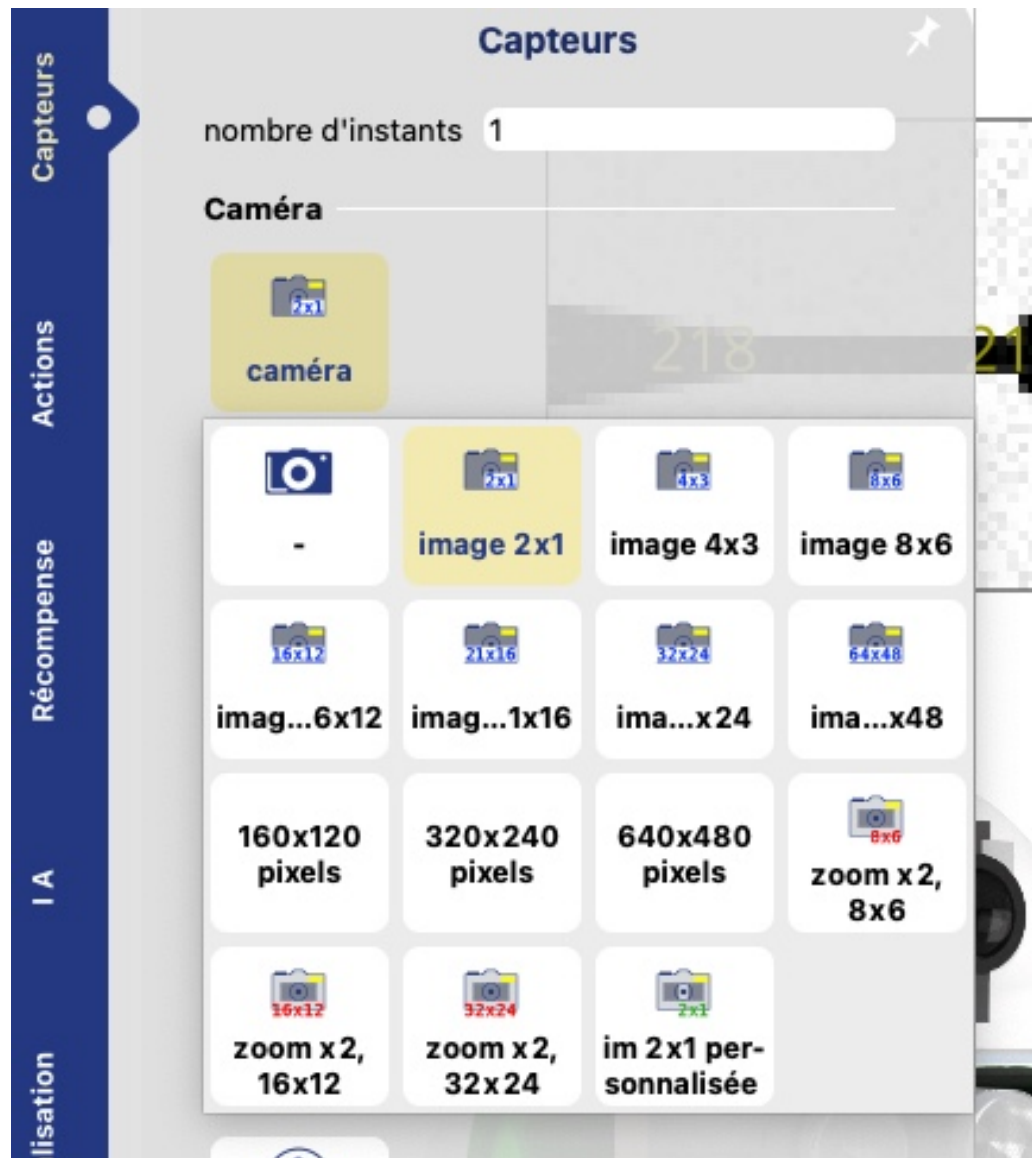
## 4 Annexe : utilisation du logiciel alphai

Lancer le logiciel **alphai** en mode simulation



### a) Apprentissage supervisé par recherche du plus proche voisin

1. Dans l'onglet **capteur** choisir en premier lieu (pour le mode plus proche voisin) :
  - camera image : 2x1,
  - précalcul image : canal vert



2. Dans l'onglet **actions** choisir la rotation sur place et la translation vers l'avant

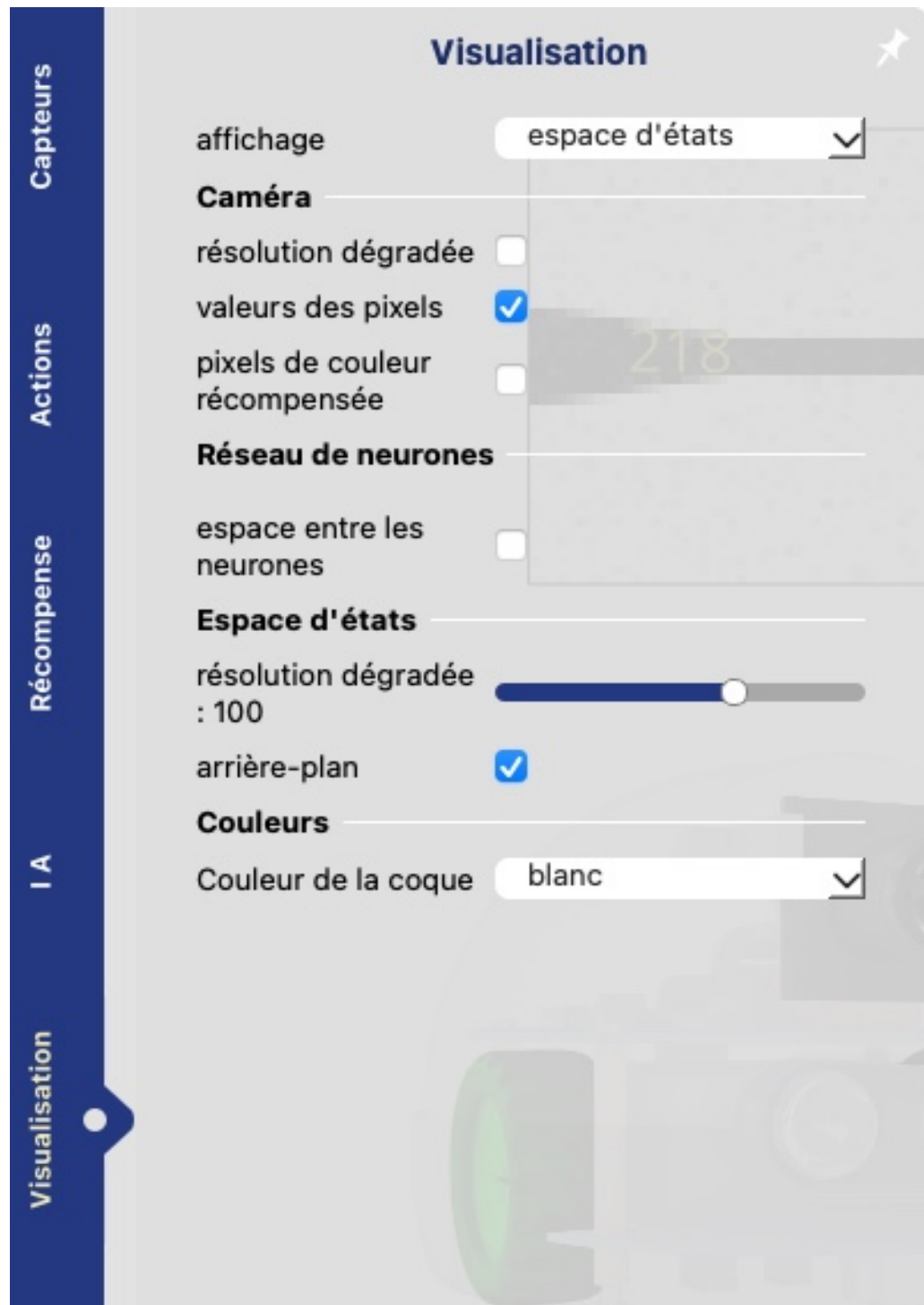


### 3. Dans l'onglet **IA**

- type d'apprentissage : apprentissage supervisé
- algorithme : K plus proche voisin



4. Dans l'onglet **Visualisation** cocher la case "arrière plan"

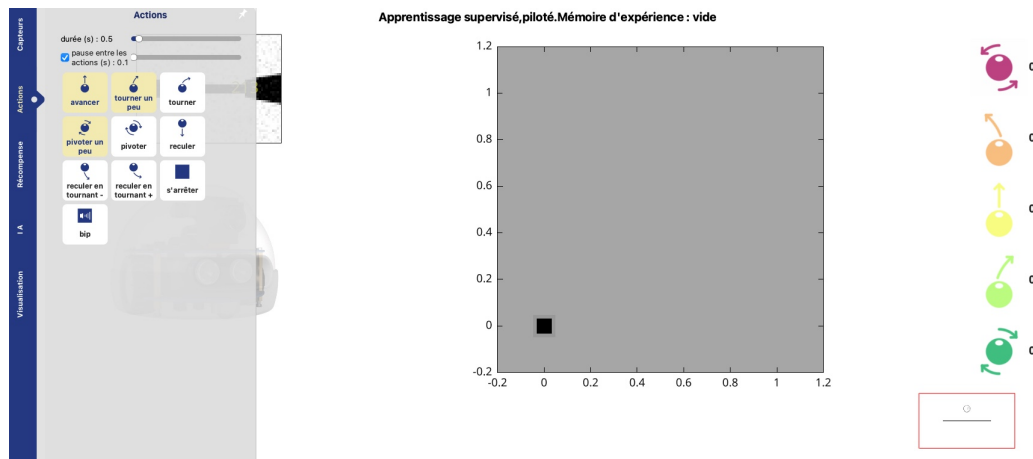


5. Vous pouvez alors activer la fonction apprentissage (en bas à gauche) puis diriger le robot pour lui faire apprendre à effectuer le circuit désiré.
6. Un fois l'apprentissage effectué, vous pouvez vérifier que votre apprentissage a bien fonctionné en faisant travailler le robot en autonomie en cliquant sur le mode **Autonome** en bas à gauche.

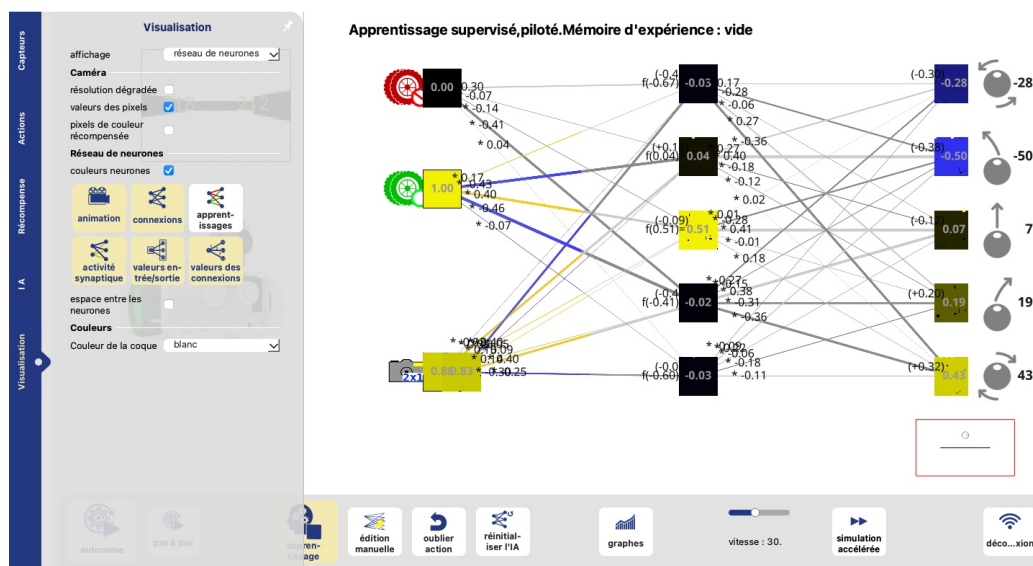
#### b) Apprentissage supervisé par réseau de neurones

1. Dans l'onglet **capteur** vous pouvez ajouter des données d'entrée : plus de pixels sur la caméra et ajouter le capteur de vitesse "bloqué/mouvement"
2. Dans l'onglet **actions** choisir la rotation sur place et la translation vers l'avant mais on peut également ajouter tourner un peu ce qui ajoute deux données de sortie.



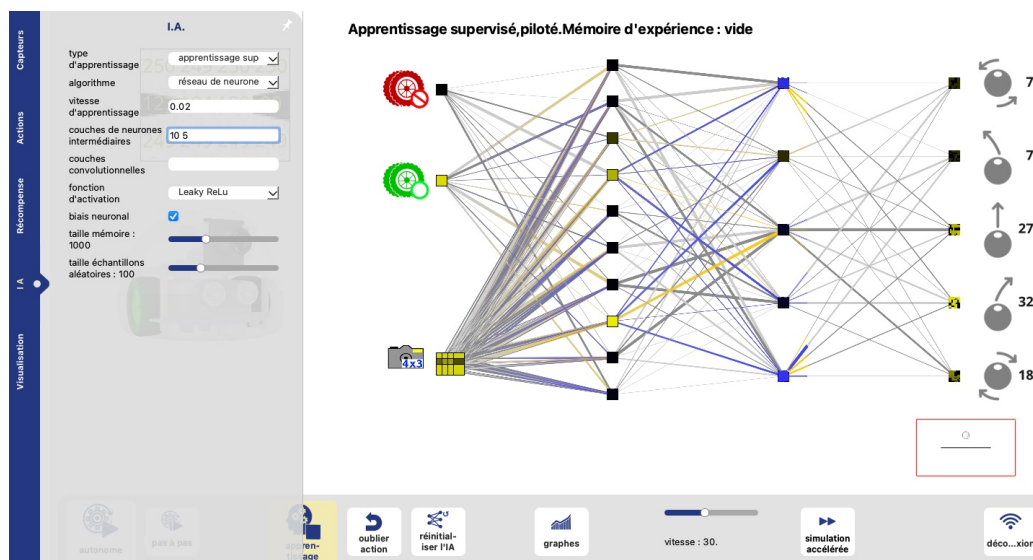


3. Dans l'onglet **Visualisation** choisir dans le menu déroulant affichage : réseau de neurones. On peut aussi ajouter les valeurs entrée/sortie et valeurs de connexion



4. Dans l'onglet **IA**

- type d'apprentissage : apprentissage supervisé
- algorithme : réseau de neurones
- Couches de neurones : séparés d'espace on précise par couche de neurones intermédiaires le nombre de neurones. Par exemple en mettant "10 5" on aura deux couches intermédiaires de respectivement 10 et 5 neurones.





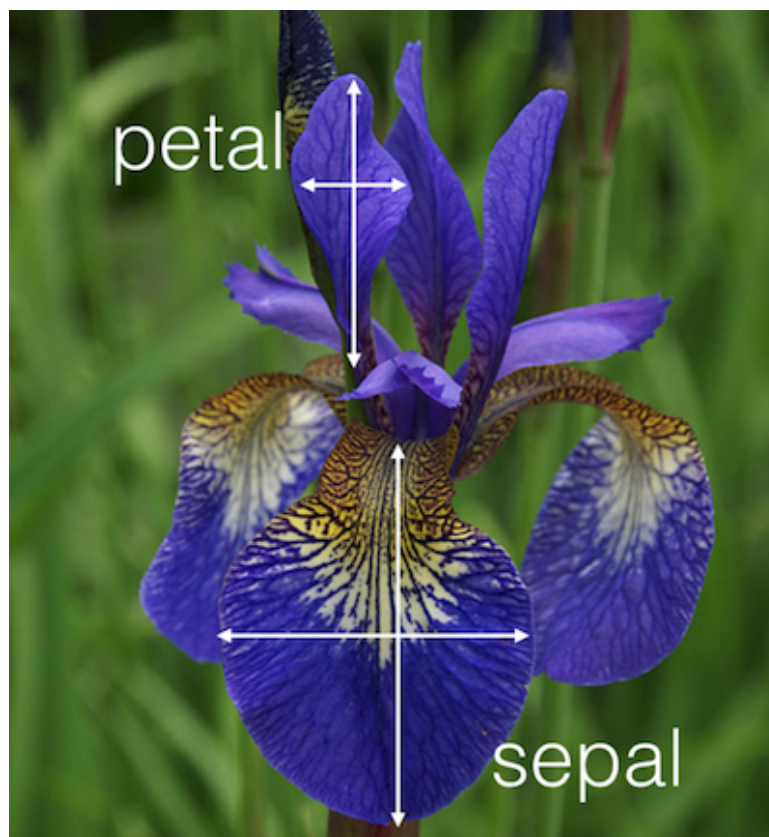
5. Vous pouvez alors activer la fonction apprentissage (en bas à gauche) puis diriger le robot pour lui faire apprendre à effectuer le circuit désiré.

## Exercice 2 : Classification d'images d'IRIS

**Source :** Xavier Pessoles

### 1 Présentation

On s'intéresse ici à la base de donnée des Iris, présente par exemple dans `scikit-learn`, les données sont composées de 150 observations, chacune composée de 4 caractéristiques : longueur et largeur du sépale ainsi que longueur et largeur du pétale.



Pour la mise en oeuvre du TP on pourra utiliser un notebook développé sur Capytal. Avec le QR code ci-contre ou l'url ci-dessous, vous pourrez vous connecter via votre identifiant ENT.

<https://capytale2.ac-paris.fr/web/c/42ed-1417016>

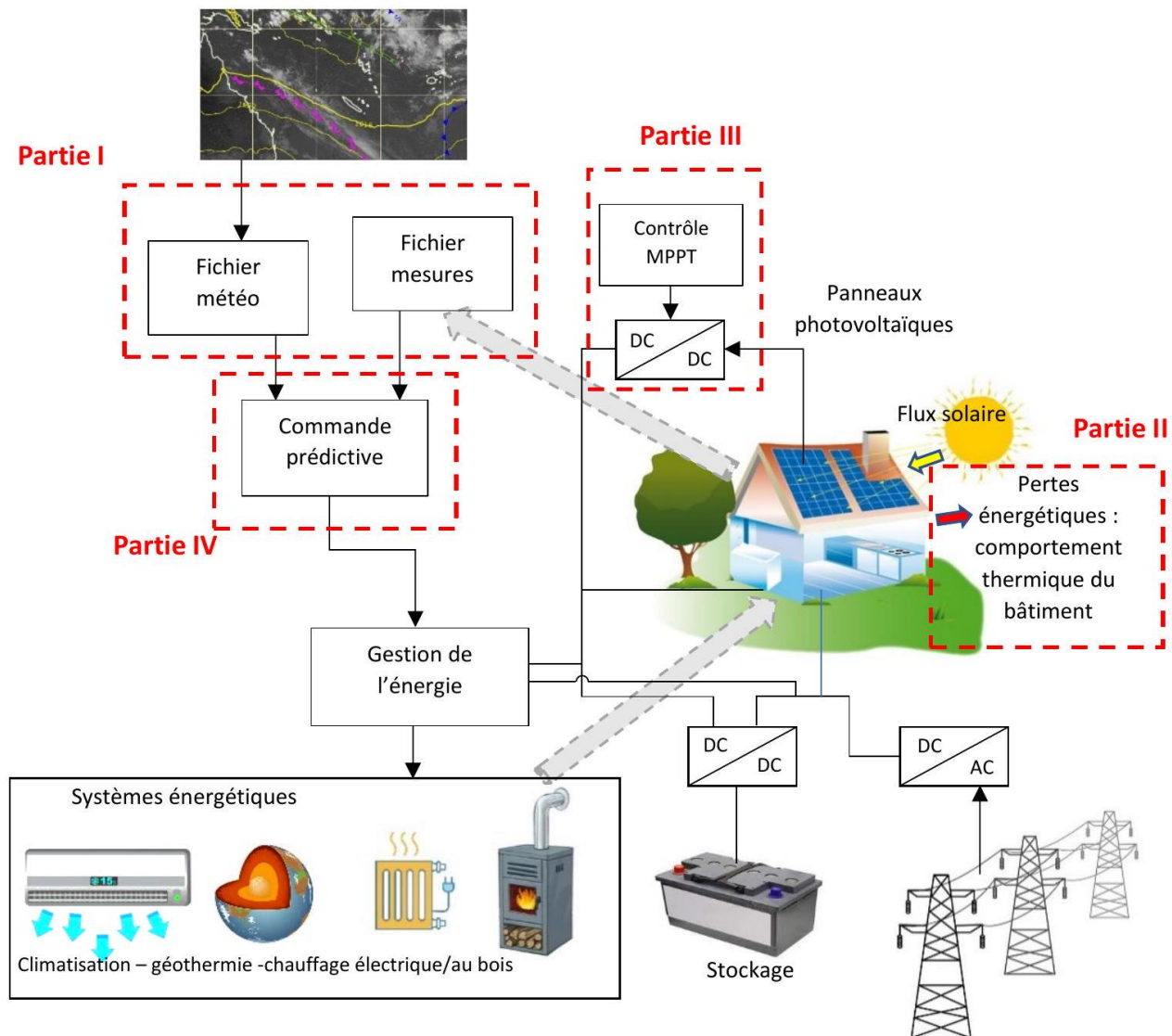


## Exercice 3 :

---

### 1 Présentation du sujet

- La partie 1 présente l'étude et demande d'extraire des informations d'une base de données qui servira ensuite de support pour l'optimisation de la gestion de l'énergie.
- La partie 2 traite du dimensionnement d'une méthode de commande haut niveau permettant de contrôler la thermique du bâtiment et d'une méthode issue de Machine Learning.



## 2 Présentation de l'étude

### a) Contexte et problématique de l'étude

Le rapport du GIEC 2022 a mis en exergue l'importance du secteur du bâtiment dans la lutte contre le réchauffement climatique. Le secteur du bâtiment représente 44% de l'énergie consommée en France et émet plus de 123 millions de tonnes de CO<sub>2</sub> (chiffres officiels du gouvernement Français en 2021). Ainsi, les experts du GIEC estiment qu'en mobilisant des solutions performantes, 61% des émissions de polluants pourraient être évitées dues au secteur du bâtiment en s'appuyant sur trois stratégies :

- Une politique d'efficacité énergétique (42%);
- Réduction de la demande en matériaux et en énergie (10%);
- Favoriser les énergies renouvelables (9%).

Plusieurs éléments viennent aider cette transition comme l'émergence des nouveaux réseaux « intelligents » : smart grid, éco maisons, green buildings, éco quartiers. Ces réseaux possèdent une vision locale du rapport producteur - consommateur d'énergie.

Dans ce sujet, nous allons aborder les différentes problématiques liées au green building. Le green building est conçu pour réduire l'impact global du bâtiment sur l'environnement tout en garantissant un niveau de confort important aux utilisateurs.

Pour cela, les green buildings produisent et stockent une partie de leur énergie nécessaire sous différentes formes comme thermique, électro-chimique, etc. Ces réseaux sont également guidés par l'IOT (Internet of Things) formant

un écosystème digital. Les capteurs de température, de consommation d'énergie, de présence sont connectés à internet et permettent une gestion optimale des énergies pour un ensemble de bâtiments.

La première étape pour concevoir un éco-bâtiment est de prévoir dès la phase de conception du bâtiment des matériaux respectueux de l'environnement, de l'extraction de la matière première, à la fabrication, l'utilisation puis le recyclage. Cette démarche est appelée éco-conception. Nous ne la traiterons pas dans ce sujet.

Dans un second temps, pour mettre en place des stratégies optimales d'énergies, il est nécessaire de modéliser la thermique du bâtiment. Cette étape est primordiale mais néanmoins très complexe. La modélisation dépend de l'environnement (ensoleillement, températures, pluviométrie, etc.), de la structure du bâtiment et des moyens de contrôle des énergies.

À partir des besoins en énergies exprimés pour le fonctionnement du bâtiment, il est nécessaire de prévoir les différents moyens d'approvisionnement en énergie du bâtiment du point de vue thermique, électrique, solaire, en fluide, etc.

Une fois tout cela exprimé, il est possible de définir une stratégie haut niveau de contrôle du bâtiment prenant en compte toutes les sources d'énergie et la maximisation du confort de l'utilisateur.

### b) Rôle des différents capteurs

Le green building ne peut être efficient sans un nombre important de capteurs pour contrôler, anticiper, réguler les différentes sources d'énergies. Le développement des IOT permet de centraliser toutes ces informations de manière efficace mais également de traiter un grand nombre d'informations.

Une solution possible par la gestion des informations des différents capteurs est l'utilisation d'une base de données, Tableau 1, où l'on vient enregistrer périodiquement les informations issues des capteurs.

| id  | adresse_I2C | date       | heure_UTC | mesure |
|-----|-------------|------------|-----------|--------|
| 1   | 11110yz0    | 2022-09-01 | 14:00:00  | 20,5   |
| ... | ...         | ...        | ...       | ...    |
| ... | ...         | ...        | ...       | ...    |
| ... | ...         | ...        | ...       | ...    |
| 901 | 11110yz0    | 2022-09-14 | 14:00:00  | 21     |
| 902 | 11110yz2    | 2022-09-14 | 14:05:00  | 27     |
| 903 | 11110yz4    | 2022-09-14 | 14:10:00  | 166    |
| 904 | 11110yz6    | 2022-09-14 | 14:15:00  | 54     |
| 905 | 11110yz8    | 2022-09-14 | 14:20:00  | 14     |
| 906 | 11110yz0    | 2022-09-14 | 14:30:00  | 21     |
| 907 | 11110yz2    | 2022-09-14 | 14:35:00  | 27,5   |
| 908 | 11110yz4    | 2022-09-14 | 14:40:00  | 160    |
| 909 | 11110yz6    | 2022-09-14 | 14:45:00  | 50     |
| 910 | 11110yz8    | 2022-09-14 | 14:50:00  | 14     |
| 911 | 11110yz0    | 2022-09-14 | 15:00:00  | 21,5   |
| 912 | 11110yz2    | 2022-09-14 | 15:05:00  | 28     |
| 913 | 11110yz4    | 2022-09-14 | 15:10:00  | 155    |
| 914 | 11110yz6    | 2022-09-14 | 15:15:00  | 42     |
| 915 | 11110yz8    | 2022-09-14 | 15:20:00  | 13     |

TABLE 1 – Base de données capteurs regroupant la mesure de tous les capteurs.

Sur le Tableau 2, il est indiqué la correspondance entre l'adresse du capteur sur un bus I2C et le type de mesure effectué. La mise à jour de la base de données et l'extraction des données des capteurs ne sont pas demandées. Pour chaque capteur, une mesure est effectuée toutes les demi-heures.

**Q 8 : A partir de la base de données capteurs, écrire une requête SQL permettant d'extraire les valeurs de la température interne  $T_{\text{int}}$  et externe  $T_{\text{ext}}$  du bâtiment à une date souhaitée. On propose ici de prendre les valeurs de température le 14 septembre 2022 sur l'intervalle [14h, 14h30[.**

Les mesures de vitesse à l'extérieur du bâtiment peuvent être fortement bruitées, voir contenir des données aberrantes. On propose de réaliser un filtrage de la mesure sur les trois dernières mesures.

**Q 9 : A partir de la base de données capteurs, écrire une requête SQL permettant d'extraire les trois dernières mesures de vitesse de vent  $V_{\text{vent}}$  puis d'en faire la moyenne. On propose ici de prendre comme valeur finale de vitesse de vent, la vitesse de vent mesurée le 14 septembre 2022 sur l'intervalle [14h, 14h30[. Cette valeur sera la**

| Adresse I2C capteur | Type de capteur                                     |
|---------------------|---|
| 11110yz0            | Mesure température interne du bâtiment [°C] (float) |
| 11110yz2            | Mesure température externe [°C] (float)             |
| 11110yz4            | Mesure éclairage [W/m <sup>2</sup> ] (float)        |
| 11110yz6            | Mesure humidité [%] (float)                         |
| 11110yz8            | Mesure vitesse du vent [m/s] (float)                |

TABLE 2 – Correspondance entre l'adresse du capteur au format I2C et le capteur de mesure.

**dernière valeur à considérer pour effectuer le filtrage.**

Il est nécessaire de prévoir la puissance solaire accessible à un instant donné afin de prévoir les différents échanges d'énergies.

**Q 10 : A partir de la base de données capteurs, écrire une requête SQL permettant de récupérer l'éclairement  $E_{\text{solaire}}$  à une date souhaitée puis de calculer la puissance solaire disponible pour le bâtiment en considérant qu'il y a 20 m<sup>2</sup> de panneaux solaires ayant un rendement solaire électrique de 20%. On propose ici de prendre les valeurs de température le 14 septembre 2022 sur l'intervalle [14h, 14 h30[.**

À partir de ces données capteurs, il est possible de définir des stratégies haut niveau de contrôle du bâtiment en y incluant la thermique et la gestion d'énergie électrique.

### 3 Construction d'une loi de pilotage prédictive

Dans cette étude, nous allons mettre en place une loi de commande qui pilotera les différents systèmes de régulation énergétiques afin d'obtenir une température consigne du bâtiment. En s'appuyant sur le modèle développé en seconde partie, cette commande devra être prédictive et exploitera la base de données mise en place au début de ce sujet : elle prendra en compte les variations météorologiques futures issues de prévisions et/ou données des années précédentes.

Cette commande doit répondre aux besoins de confort thermique et de consommation énergétique.

On souhaite que la température à l'intérieur du bâtiment se rapproche d'une température consigne  $T_c$ . On note  $y_c = T_c - T_{\text{initiale}}$  la variation de température correspondante, ce qui permet d'avoir des conditions initiales nulles.

La commande des systèmes de régulation sera notée  $x(t)$ .

La méthode consiste à déterminer la séquence future et la commande à appliquer en vue de forcer la sortie à suivre une trajectoire prédéfinie : la trajectoire de référence, Figures 1 et 2.

En se basant sur le travail réalisé dans une autre partie du sujet on propose :

$$G_r(p) = \frac{Y_r(p)}{X(p)} = \frac{K_r e^{-\eta_r p}}{1 + \tau_r p} \quad G_m(p) = \frac{Y_m(p)}{X(p)} = \frac{K_m}{1 + \tau_m p}$$

Le retard  $\eta_r$  provient de la réactivité des systèmes de régulation. L'incrément de température qui doit être atteint, entre l'instant  $t_k$  et l'horizon de prédiction  $t_{k+H}$  est :

$$\Delta_r(t_{k+H}) = y_r(t_{k+H}) - y_r(t_k)$$

Où  $t_k$  représente l'instant présent. On peut également fixer l'incrément désiré dans l'espace du modèle :

$$\Delta_m(t_{k+H}) = y_m(t_{k+H}) - y_m(t_k)$$

La loi de commande prédictive s'obtient à partir de l'égalité entre la trajectoire de référence et la sortie prédite du processus en un point de coïncidence  $t_{k+H}$  :

$$\Delta_r(t_{k+H}) = \Delta_m(t_{k+H}) = y_{ref}(t_{k+H}) - y_{ref}(t_k)$$

Il s'agit maintenant de construire le modèle échantillonné.

La période d'échantillonnage est constante ( $T_e$ ) et on utilise un bloqueur d'ordre zéro (BOZ), qui maintient le signal constant jusqu'à l'échantillon suivant, Figures 3 et 4 :

La réponse impulsionnelle du bloqueur est représentée sur la Figure 21 :

**Q 11 : Exprimer la réponse impulsionnelle du  $u_{BOZ}(t)$  bloqueur. En déduire sa fonction de transfert  $U_{BOZ}(p)$ .**

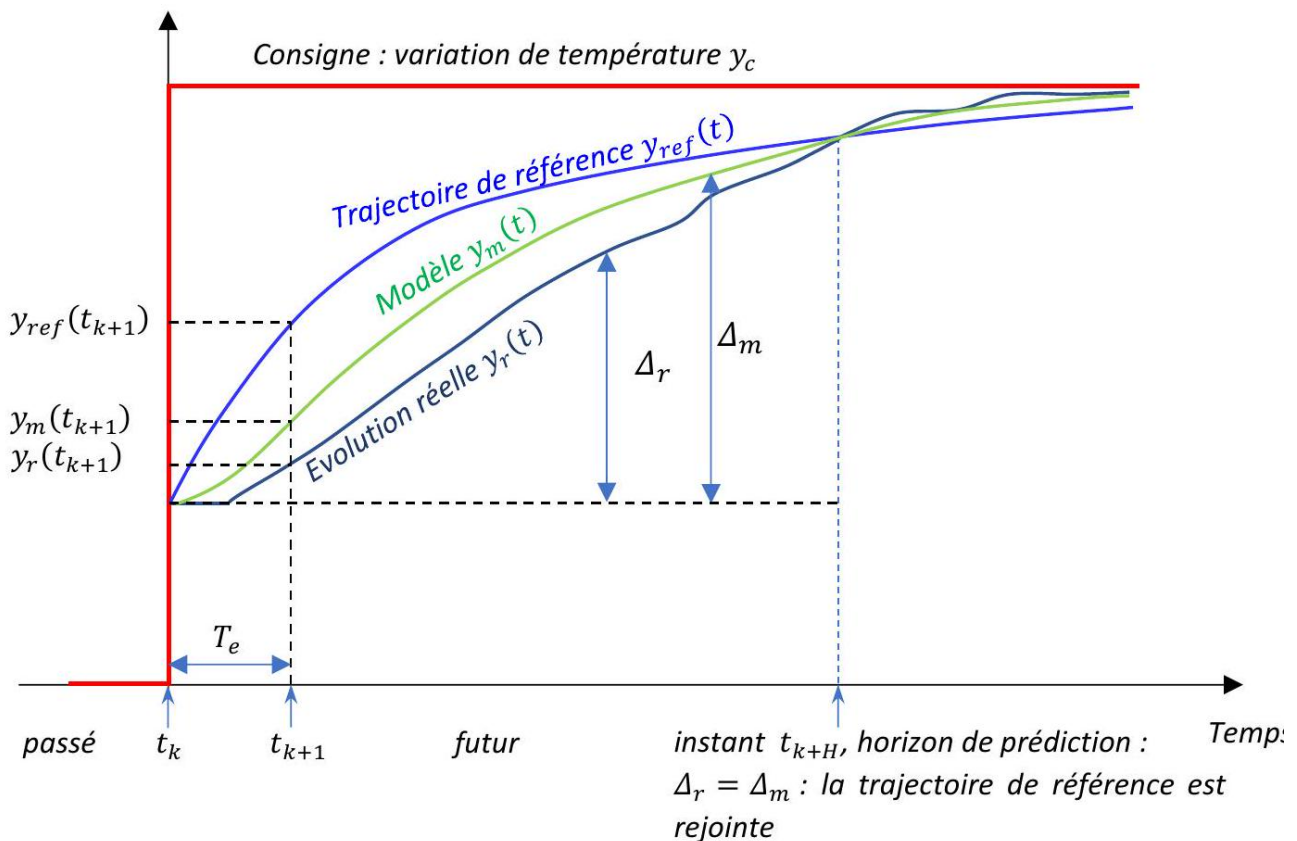


FIGURE 1 – Evolutions de la température réelle, de la température issue du modèle et de la trajectoire de référence

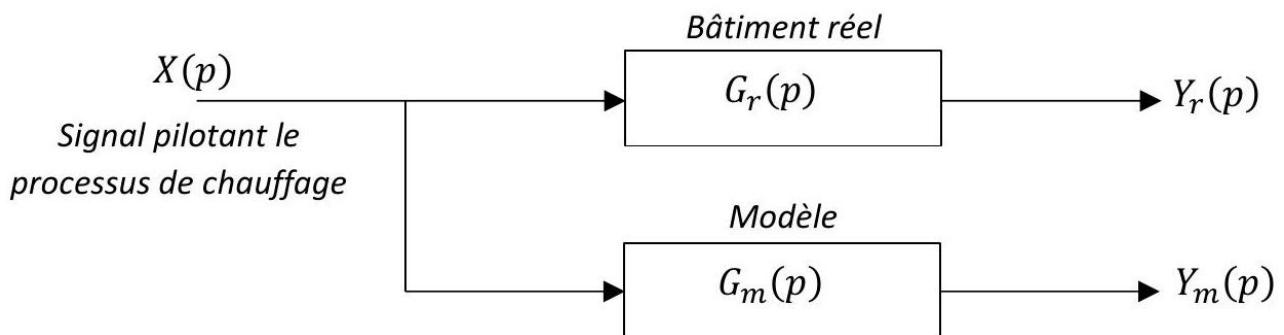


FIGURE 2 – Schéma du système avec le bâtiment réel et le modèle

On peut ainsi obtenir l'équation de récurrence du système, mais en utilisant une méthode (la transformée en  $z$ ) qui n'est pas au programme de PSI. Nous ne la développerons donc pas plus ici. Cette équation de récurrence, exacte, est la suivante :

$$y_r(t_{k+1}) = \alpha_r y_r(t_k) + K_r (1 - \alpha_r) x(t_{k+1} - \eta_r) \quad \text{avec} \quad \alpha_r = e^{-\frac{T_e}{\tau_r}}$$

(On considère que  $\eta_r$  est un multiple de la période  $T_e$ ).

Nous pouvons en revanche obtenir une approximation de cette équation, en utilisant la méthode d'Euler.

**Q 12 :** Ecrire l'équation différentielle à l'origine de la fonction de transfert  $G_r(p)$  puis, à l'aide de la méthode d'Euler explicite, formuler le problème sur un intervalle  $[t_k, t_{k+1}]$ .

**Q 13 :** En déduire l'équation de récurrence approchée à l'ordre 1, sous la forme :  $y_r(t_{k+1}) = \alpha_E y_r(t_k) + \beta_E x(t_{k+1} - \eta_r)$ .

**Q 14 :** Comparer les coefficients  $(\alpha_r, \beta_r)$  et  $(\alpha_E, \beta_E)$  et formuler une relation entre  $T_e$  et  $\tau_r$  pour que le schéma d'Euler soit précis.



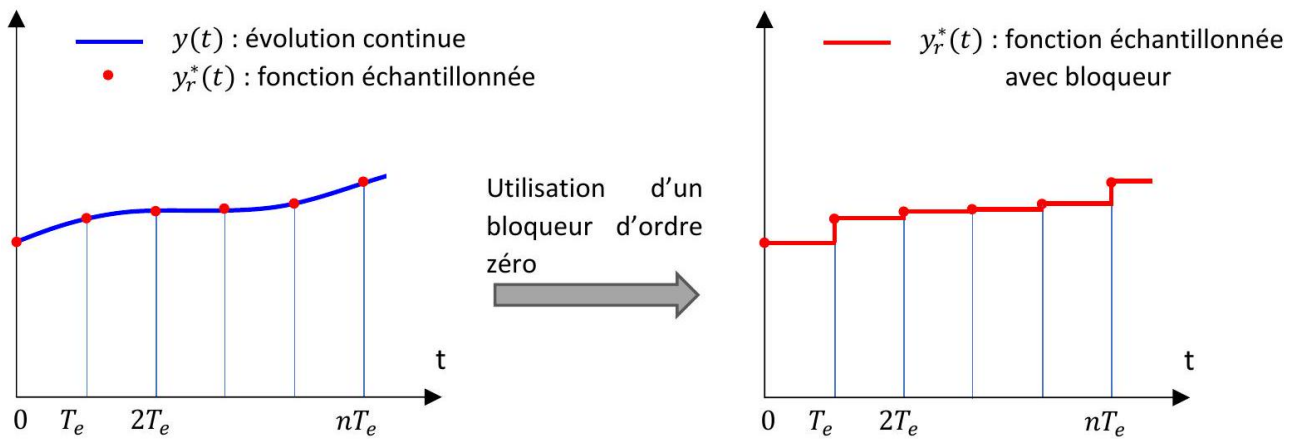


FIGURE 3 – Fonctionnement du bloqueur d'ordre 0

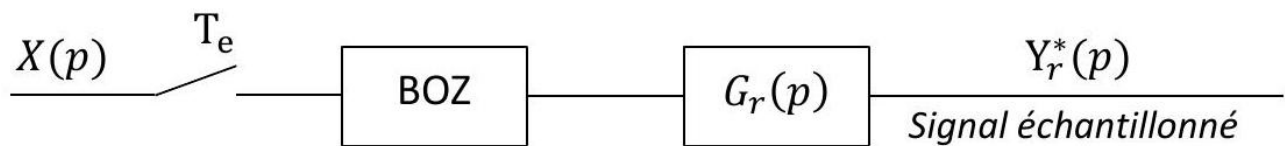


FIGURE 4 – Modèle dans le domaine de Laplace du bloqueur et du système

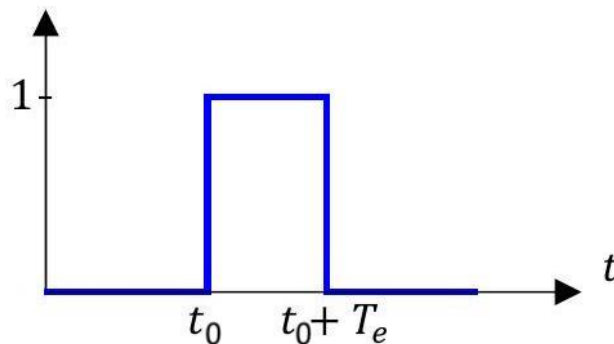


FIGURE 5 – Réponse impulsionnelle du bloqueur d'ordre 0

On considère la commande comme constante, égale à une valeur  $x(k)$  après son application à l'instant  $k$  :

$$x(t_k) = x(t_{k+1}) = x(t_{k+2}) = \dots = x(t_{k+H})$$

**Q 15 : Déterminer la loi de commande prédictive du système de régulation :**

$$y_r(t_{k+H}) = f(y_r(t_k), x(t_k), \alpha_E, K_r)$$

**Q 16 : Par analogie, exprimer la commande prédictive du modèle :**  $y_m(t_{k+H}) = f(y_m(t_k), x(t_k), \alpha_E, K_m)$ .

La trajectoire de référence est choisie de forme exponentielle, avec une constante de temps  $\tau_c$  liée au délai souhaité pour que la température consigne soit atteinte :

$$y_{ref}(t) = y_{ref}(t_k) + (y_c - y_{ref}(t_k)) \left( 1 - e^{-\frac{t-t_k}{\tau_c}} \right)$$

**Q 17 : En exploitant la coïncidence à  $t_{k+H}$ , déterminer l'incrément  $\Delta_r(t_{k+H})$  en fonction de  $y_c$ ,  $y_{ref}(t_k)$  et  $\lambda = e^{-T_e/\tau_c}$ .**

**Q 18 : Exprimer l'incrément lié au modèle  $\Delta_m(t_{k+H})$  en fonction de  $y_m(t_k)$ ,  $x(t_k)$ ,  $\alpha_E$ ,  $K_m$  et  $H$ .**



La première partie du sujet initial a mis en évidence l'utilisation d'une base de données, actualisée régulièrement. L'intérêt de la commande développée ici est d'utiliser ces données pour recalculer la trajectoire de référence à chaque pas de temps. Ce recalage s'exprime par :

$$y_{ref}(t_k) = y_r(t_k)$$

**Q 19 : En exprimant la coïncidence, à l'horizon  $t_{k+H}$ , entre l'écart réel et l'écart du modèle, exprimer la loi de commande  $x(t_k)$  en fonction de  $y_m(t_k)$ ,  $y_r(t_k)$ ,  $y_c$ ,  $\alpha_E$ ,  $K_m$  et  $H$ .**

La période d'échantillonnage étant faible devant le temps de réponse des systèmes de régulation, on admet que le modèle continu est similaire au modèle discret développé.

La figure 6 présente une ébauche de la commande prédictive.

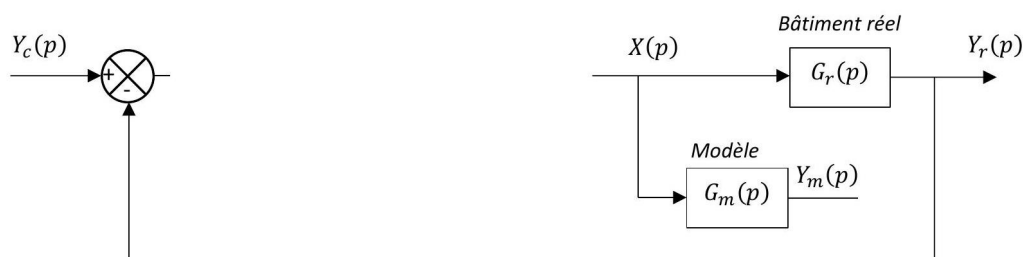


FIGURE 6 – Schéma bloc (incomplet) de la commande prédictive

**Q 20 : Compléter le schéma bloc modélisant cette commande prédictive.**

## 4 Modèle obtenu par Machine Learning

On veut maintenant simplifier le système de commande en améliorant la qualité de ses prédictions : la température de sortie, prédite ou réelle, appartiendra à une des 6 classes décrites par le tableau suivant :

| Classe | Température $y_{k+1}$ | Classe | Température $y_{k+1}$ |
|--------|-----------------------|--------|-----------------------|
| 1      | $[+7; +20[$           | 4      | $[-2; 0[$             |
| 2      | $[+2; +7[$            | 5      | $[-7; -2[$            |
| 3      | $[0; +2[$             | 6      | $[-20; -7[$           |

FIGURE 7 – Catégories des températures de sortie

On commence par enrichir la base de données d'observations créée dans la première partie de ce sujet avec des données issues des diagnostics énergétiques (DPE) : prise en compte du type et de l'année de construction, de la surface habitable, de la hauteur de réalisation, du nombre de niveaux, de la surface des parois opaques verticales déperditives, etc...

Le modèle prédictif précédent a également permis d'associer à cette base de données les valeurs de sortie pendant une période d'observation.

On se tourne donc vers un modèle d'apprentissage (Machine Learning) supervisé (Figure 8) :

**Q 21 : Cet apprentissage doit-il utiliser une méthode de régression ou de classification ?**

**Q 22 : Sans rentrer dans le niveau de détail d'un algorithme, donner les différentes étapes de cet apprentissage et associer une méthode de classification ou de régression (selon la réponse à la question précédente).**

**Q 23 : Evaluer la sensibilité de chaque classe de la matrice. Cet apprentissage est-il performant (sensibilité > 80%) ?**

**Q 24 : Les caractéristiques du logement sont susceptibles d'évoluer au cours du temps. Proposer un autre choix d'apprentissage permettant d'améliorer les performances de ce modèle.**

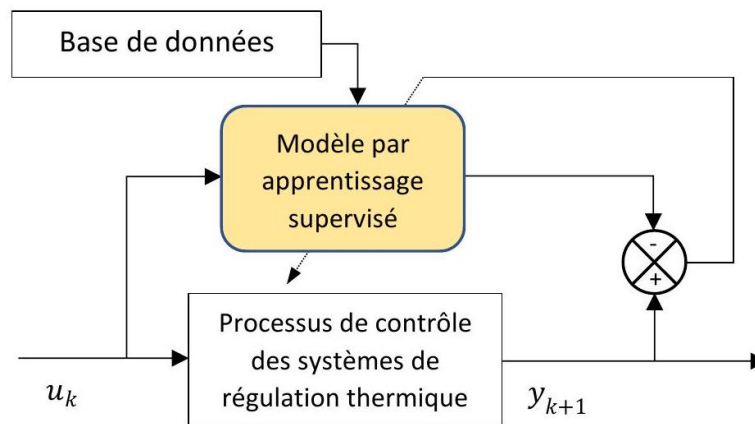


FIGURE 8 – Schéma de l'identification