



# Séquence n°3

## Système de régulation



### Fiches de synthèse mobilisées :

Fiche n°3 Système de régulation



### Sommaire des activités

ACTIVITÉ 1 : Découverte de la régulation de température d'un microprocesseur<sup>1</sup>

ACTIVITÉ 2 : Tracé des caractéristiques statiques du procédé<sup>8</sup>

ACTIVITÉ 3 : Régulation Tout Ou Rien<sup>11</sup>

ACTIVITÉ 4 : Correction Proportionnelle<sup>14</sup>

ACTIVITÉ 5 : Correction Proportionnelle et Intégrale<sup>17</sup>

## ACTIVITÉ 1 : Découverte de la régulation de température d'un microprocesseur

### Partie 1 : Pourquoi surveiller la température du microprocesseur d'un ordinateur ?

La température est une grandeur ayant une grande influence sur le comportement d'un ordinateur. En effet, les différents composants ont des plages de températures idéales. C'est le cas de la carte mère, de la carte graphique, des disques durs et du processeur.

Un composant (processeur, disque...) ayant une température trop élevée peut ralentir et même causer l'arrêt brutal du système.

D'une manière générale, les températures maximales préconisées par les fabricants sont :

- pour le processeur (CPU) : en-dessous de 90 °C.
- pour le disque dur : en-dessous de 50 °C.

Sur la plupart des cartes mères, des capteurs de température sont présents et permettent la surveillance de ces températures. Des logiciels spécialisés permettent d'afficher les valeurs mesurées (SpeedFan ou OpenHardwareMonitor par exemple).

La vidéo présentée ci dessous cherche à déterminer l'origine de l'élévation de la température du microprocesseur :

<https://ladigitale.dev/digiview/#/v/65fd42a20d92c>

Comme le montre la vidéo, la température du microprocesseur dépend directement de son activité, c'est à dire des applications ou programmes en cours d'exécution. Une élévation de température pénalisant les performances de l'ordinateur, il est donc indispensable de chercher à évacuer l'énergie thermique dissipée par le processeur. C'est le rôle de dispositifs spécifiques.



Les photos ci-après présentent plusieurs vues de l'intérieur d'un ordinateur, sur lesquelles on identifie les composants suivants :

- carte mère,
  - alimentation,
  - disque dur,
  - mémoire vive,
  - processeur.
1. Repérer et nommer les dispositifs permettant d'évacuer l'énergie thermique dissipée par le processeur et les autres composants de l'ordinateur.
  2. Sur la Photo 2, représenter par des flèches, la circulation du flux d'air à travers l'ordinateur.  
Préciser le sens de l'échange thermique ayant lieu entre l'air circulant dans l'ordinateur et le processeur.

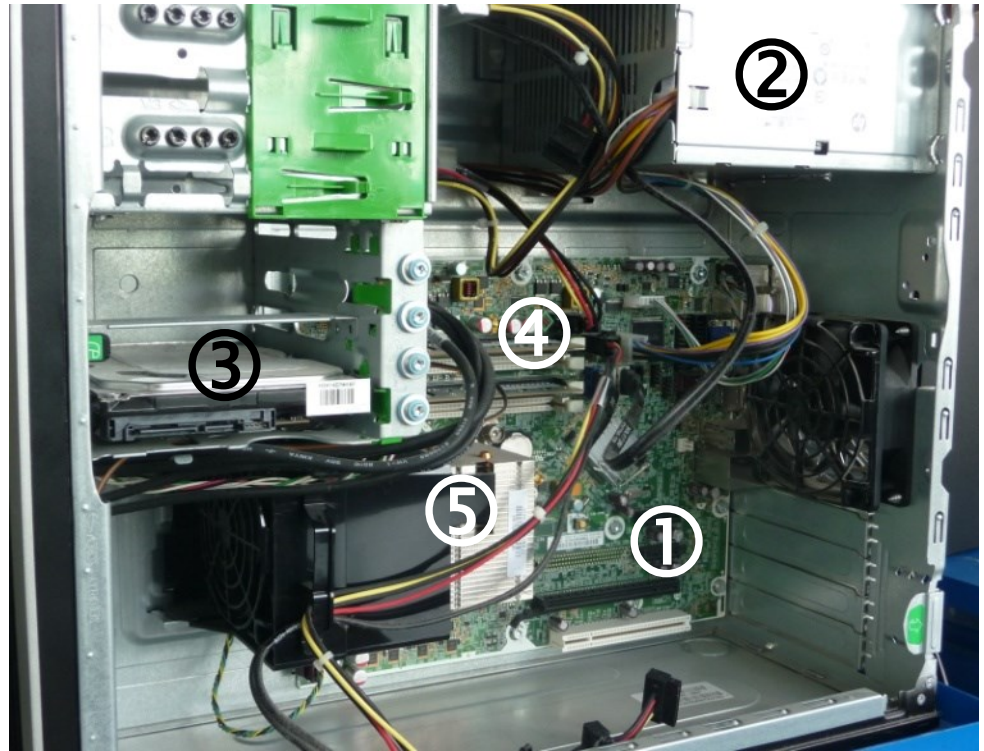


Photo 1

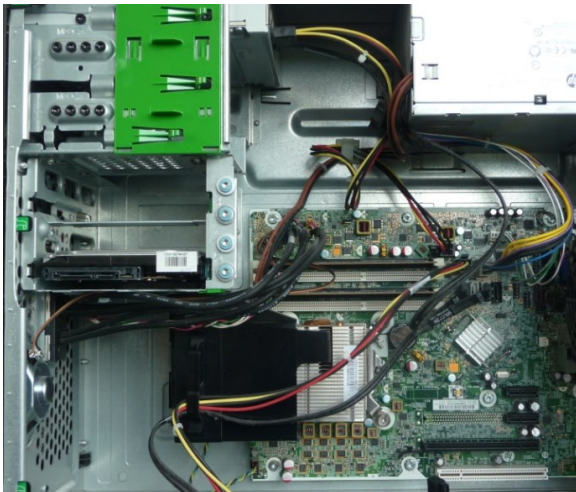


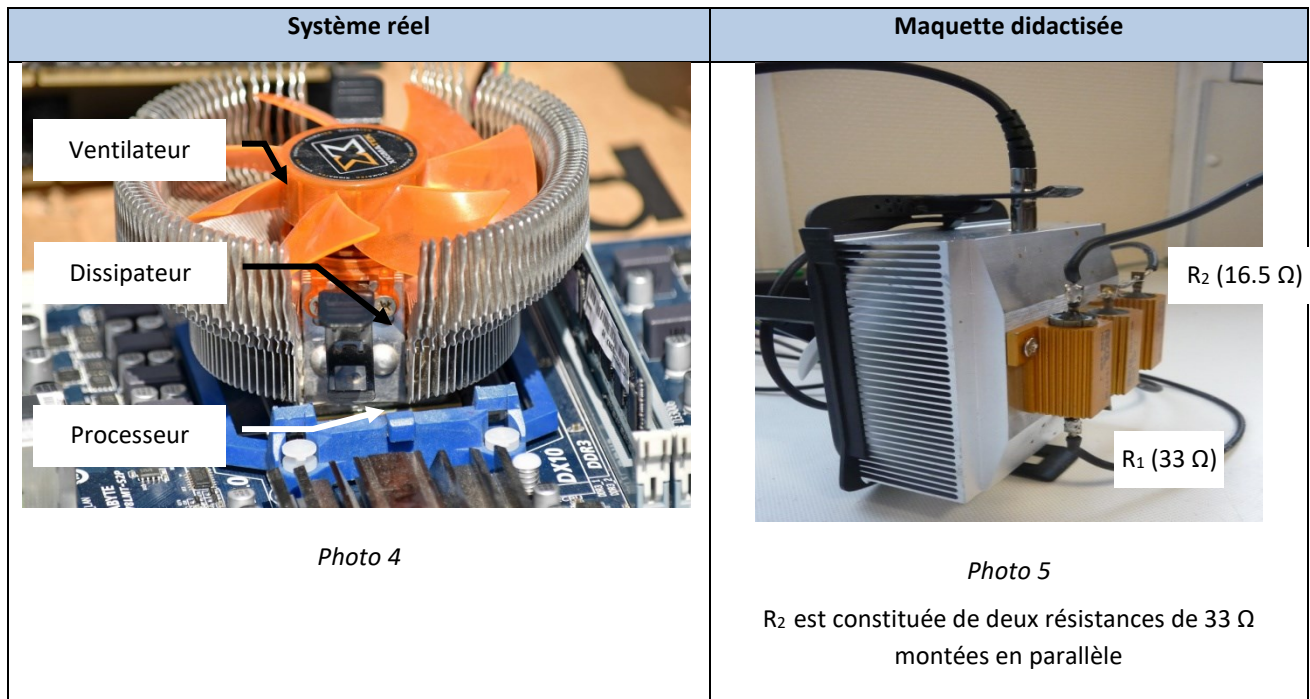
Photo 2



Photo 3

## Partie 2 : Maquette didactisée du processeur et de son système de refroidissement

Comme présenté précédemment, l'énergie thermique du processeur est évacuée vers l'extérieur grâce au couple dissipateur thermique + ventilateur (voir Photo 4).



La maquette étudiée (voir Photo 5) comporte un ventilateur, un dissipateur thermique et deux résistances électriques qui permettront de simuler l'échauffement dû au fonctionnement du processeur.

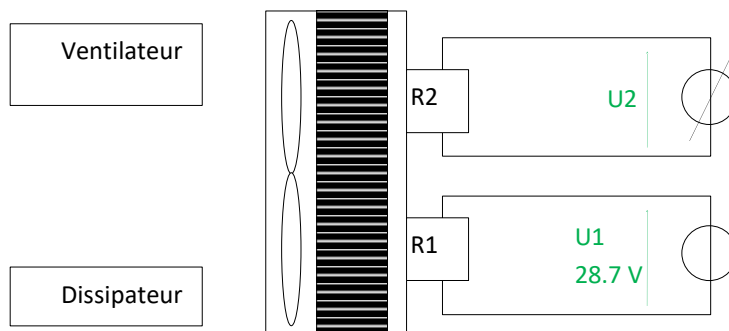
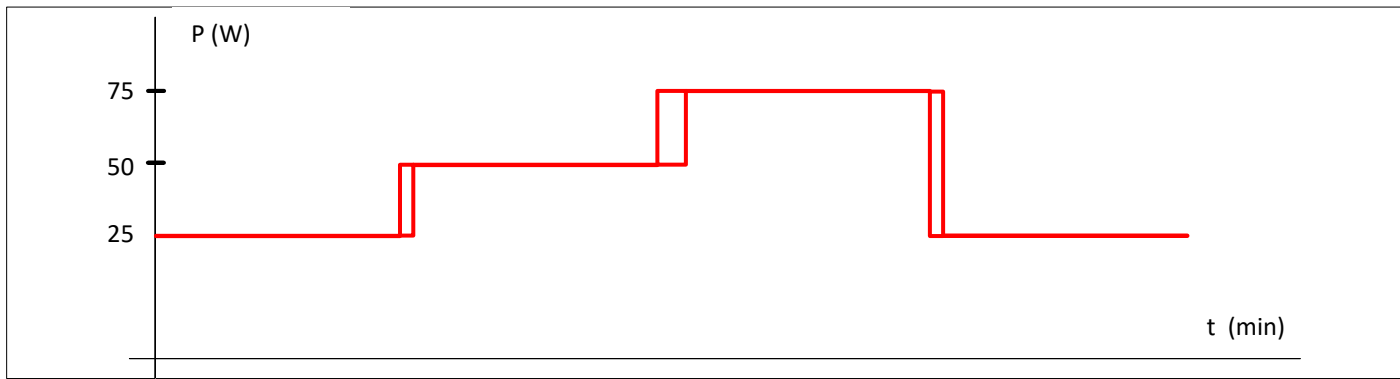


Figure 1 : schéma d'alimentation électrique des deux résistances

3. Expliquer comment les résistances électriques peuvent simuler le comportement du processeur.

Le graphe ci-dessous représente l'évolution de la puissance électrique consommée par le processeur au cours du temps en fonction des programmes exécutés par l'ordinateur.



Application en cours	Windows	Windows + Traitement de texte	Windows + Traitement de texte + Navigateur web	Windows
Puissance $R_1$				
Tension $U_1$				
Puissance $R_2$				
Tension $U_2$				

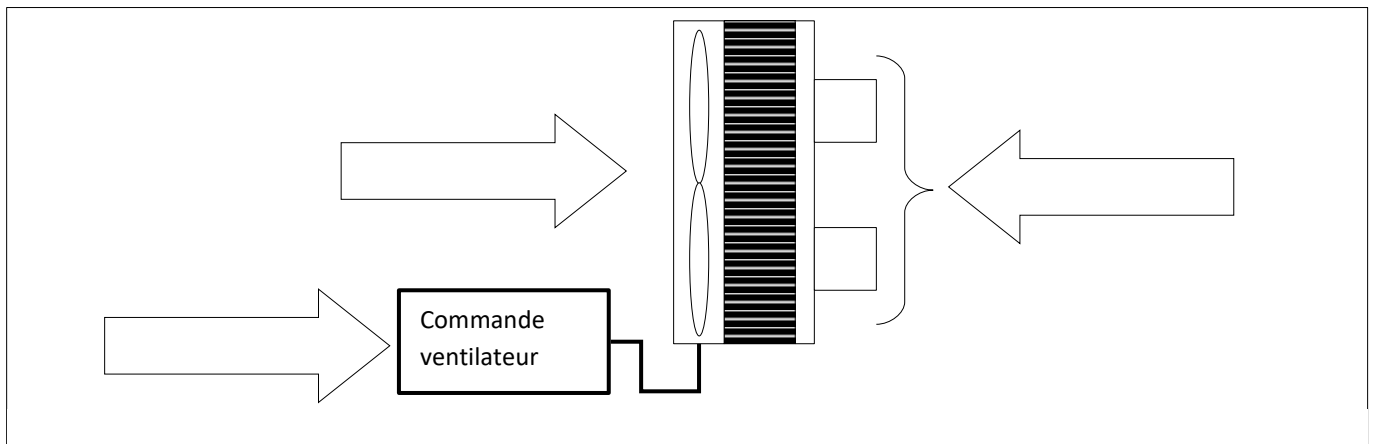
Tableau 1

- Indiquer laquelle des 3 situations est simulée par le dispositif didactique si une tension de 28,7 V est appliquée aux bornes de  $R_1$  alors que  $R_2$  n'est pas alimentée.
- En analysant la correspondance entre le graphe et le Tableau 1, compléter le Tableau 1 ci-dessus.

Dans la suite de cette étude, la résistance  $R_1$  sera toujours alimentée sous 28,7 V pour simuler le fonctionnement de base de l'ordinateur (environnement Windows). Le réglage de l'alimentation de  $R_2$  permettra de simuler l'activité du processeur lors d'exécution de programmes supplémentaires (voir Figure 1).

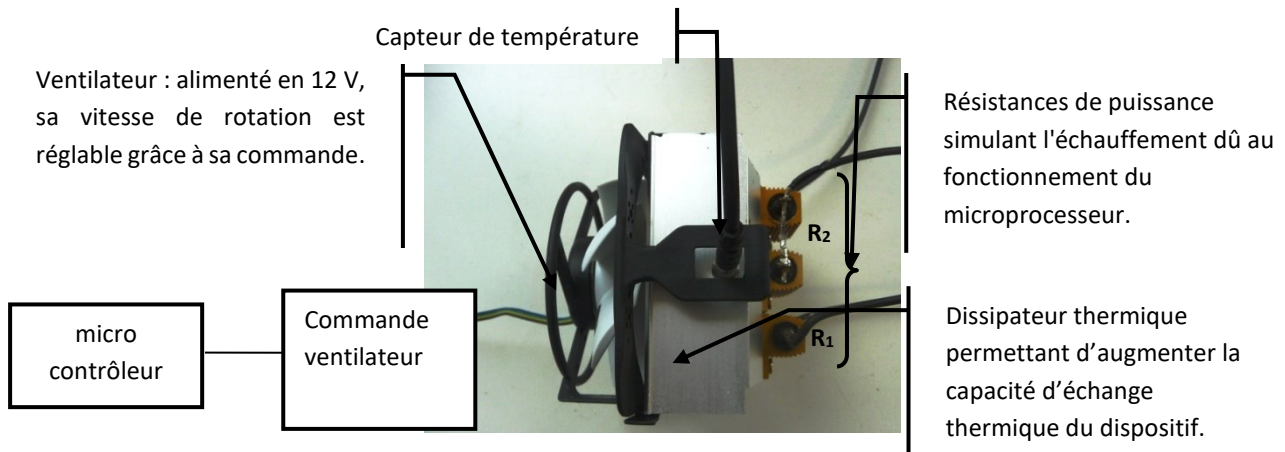
On souhaite préciser la nature des différents flux mis en jeu dans ce procédé, pour le ventilateur, sa commande, la résistance  $R_1$  et la résistance  $R_2$ .

- Compléter la figure ci-dessous en écrivant dans les flèches les termes : Flux de matière, flux d'énergie et flux d'information.





La surveillance de la température du processeur simulé sera effectuée par un capteur placé dans le dissipateur thermique comme indiqué ci-dessous :



La vitesse de rotation réglable du ventilateur permet d'avoir un débit d'air réglable à l'intérieur du dissipateur thermique de 0% (débit nul) à 100% (débit maximal).

### Partie 3 : Procédé et instrumentation

Dans notre étude, le but de la régulation est de maintenir la température du processeur égale à une consigne donnée quelle que soit son activité (puissance thermique dissipée) en agissant sur le débit d'air.

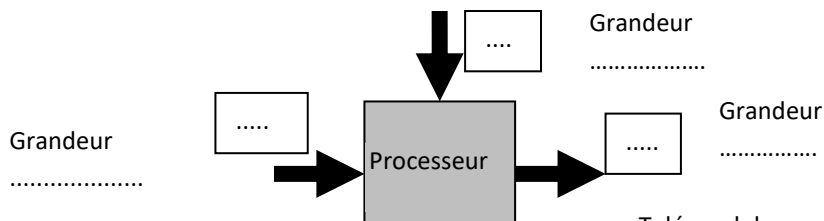
En régulation, on définit 3 types de grandeurs :

- **La grandeur réglée**, c'est la grandeur dont on cherche à piloter la valeur.
- **La grandeur réglante**, c'est la grandeur sur laquelle on peut agir pour piloter la grandeur réglée.
- **La ou les grandeurs perturbatrices**, c'est ou ce sont les grandeurs qui perturbent le pilotage de la grandeur réglée.

7. Relier par des flèches les grandeurs physiques aux grandeurs de régulation dans le tableau suivant :

Grandeurs physiques mises en jeu		Grandeurs de régulation	
•	Température T	•	Grandeur réglante
•	Débit d'air Q <sub>e</sub>	•	Grandeur perturbatrice
•	Puissance électrique P	•	Grandeur réglée

8. Compléter le schéma ci-dessous avec les 6 grandeurs du tableau précédent



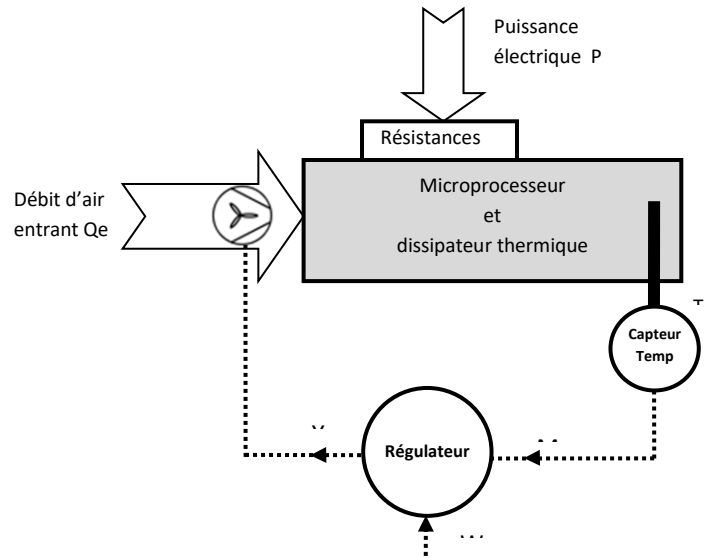


#### Partie 4 : Schéma de représentation de la boucle de régulation

9. Compléter le texte ci-dessous avec les verbes suivants : AGIR, MESURER, COMPARER.

Les éléments de la boucle de régulation doivent permettre de :

- ..... la température  $T$  (**MESURE M**)
- .....  $M$  à la **CONSIGNE (W)**
- ..... sur le ventilateur en élaborant un signal de **COMMANDE (Yr)**





## ACTIVITÉ 2 : Tracé des caractéristiques statiques du procédé

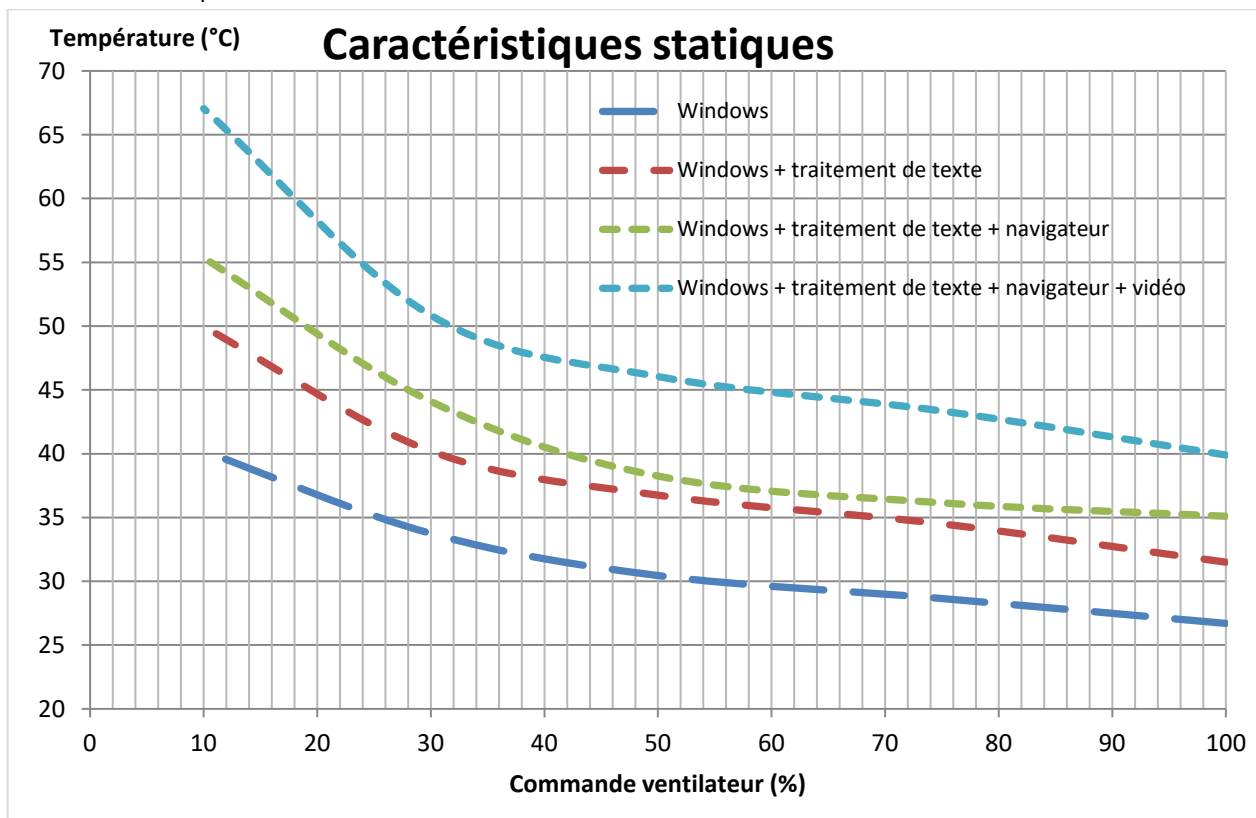
### Partie 1 : Caractéristiques statiques d'un procédé

On a vu dans la Partie 1 : ci-dessus que la température du microprocesseur dépend du nombre de programmes exécutés par l'ordinateur. Pour réguler cette température, il faut agir sur le ventilateur (Partie 2 : ACTIVITÉ 1 : )

Dans notre maquette, le nombre de programmes exécutés par l'ordinateur, entraînant une élévation de la température est simulé par la puissance électrique dissipée par les résistances  $R_1$  et  $R_2$ . Pour mettre en œuvre la régulation, il nous est nécessaire de connaître l'évolution de la température du microprocesseur en fonction du débit d'air entrant  $Q_e$  et donc de la commande du ventilateur. Cette évolution de la température en fonction de la commande du ventilateur est tracée pour une puissance électrique dissipée particulière. Cette courbe constitue **une caractéristique statique** du procédé. Lorsqu'on trace l'évolution de la température pour d'autres valeurs de la puissance électrique dissipée, on obtient un réseau de caractéristiques statiques.

On donne ci-dessous les caractéristiques statiques obtenues lors de l'étude d'un microprocesseur présent sur un ordinateur en 2002.

1. Quelle est la plage de variation de la température du microprocesseur lorsque seul l'environnement Windows est en fonctionnement ?
2. On suppose que Windows et un traitement de texte sont en fonctionnement. La commande du ventilateur est à 50%, déterminer la température atteinte par le microprocesseur.
3. On se place dans le cas où Windows, un traitement de texte, un navigateur internet et une vidéo sont en fonctionnement. Quelle valeur minimale de la commande du ventilateur permet de maintenir une température du microprocesseur inférieure à 50°C ?





## Partie 2 : Etude du dispositif permettant de tracer une caractéristique statique

Dans cette partie, l'objectif est de mesurer la température du microprocesseur et de commander la vitesse du ventilateur pour une valeur de puissance dissipée dans les résistances simulant un échauffement du microprocesseur en fonctionnement.

Le capteur de température utilisé est une CTN. On donne dans la Figure 2 son circuit de conditionnement. La tension aux bornes de la CTN est nommée  $U_{CTN}$

4. Quel est le rôle des entrées analogiques du microcontrôleur Arduino ?
5. Repérer la borne du microcontrôleur Arduino qui permet la lecture de la tension  $U_{CTN}$ .
6. De quelle grandeur physique dépend la tension  $U_{CTN}$  ?

Le ventilateur nécessite une alimentation extérieure et sa vitesse est commandée par une entrée notée PWM.

7. Quelle est la tension d'alimentation du ventilateur ?
8. Repérer la borne du microcontrôleur Arduino qui permet la commande en vitesse du ventilateur.

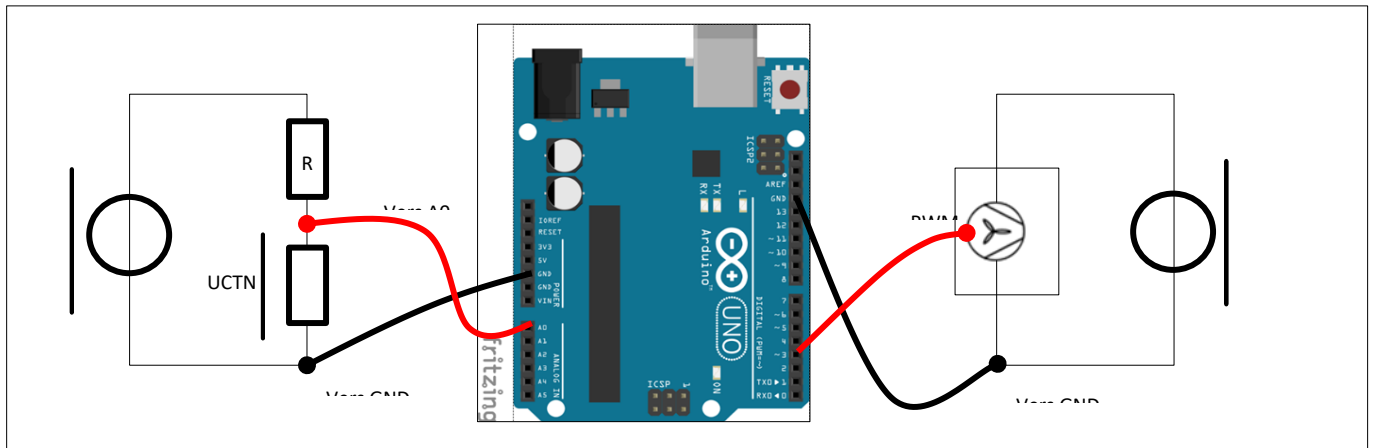


Figure 2

La CTN a été étalonnée. On a déterminé la relation entre la température  $T$  et le nombre  $N$  issu du CAN du microcontrôleur. Pour la CTN étudiée, on a trouvé :  $T = -0.1459 \times N + 120.17$ .



```
Affichage_T_et_commande_ventilateur
1 // déclaration des constantes
2 int      portTemp = 0;
3 float    a = -0.1459 ;
4 float    b = 120.17;
5 int      portVentilateur = 3;
6
7 // Définition des variables
8 int      N = 0;
9 float    Temp;
10 int     commandeVitesse = 0;
11
12
13 void setup()
14 {
15     // initialisation serial port
16     Serial.begin(9600);
17
18     pinMode(portVentilateur, OUTPUT);
19 }
20
21 void loop()
22 {
23     N = analogRead(portTemp); //Lecture N
24     Temp = a * N + b;          //calcul température T
25
26 //commande de la vitesse du ventilateur
27 // division par 500 pour que la variation de 0 à 20% soit ramenée à une variation virtuelle de 0 à 100%
28     analogWrite(portVentilateur, 255*commandeVitesse/500);
29
30
31
32     Serial.print("Température = ");
33     Serial.print(Temp);
34     Serial.println(" °C");
35     delay(1000);
36 }
```

### Programme 1

9. Repérer la ligne qui permet de fixer le numéro de port analogique sur lequel est envoyée l'information Température.
10. Repérer la ligne qui permet de fixer le numéro de port numérique pour commander le ventilateur.
11. Repérer la ligne qui permet de lire le nombre N qui dépend de la température.
12. Repérer la ligne qui permet de calculer la température.
13. Repérer la ligne qui permet de fixer la valeur de la commande en vitesse du ventilateur.
14. Repérer la ligne qui permet de commander la vitesse du ventilateur.

### Partie 3 : Tracé d'une caractéristique statique

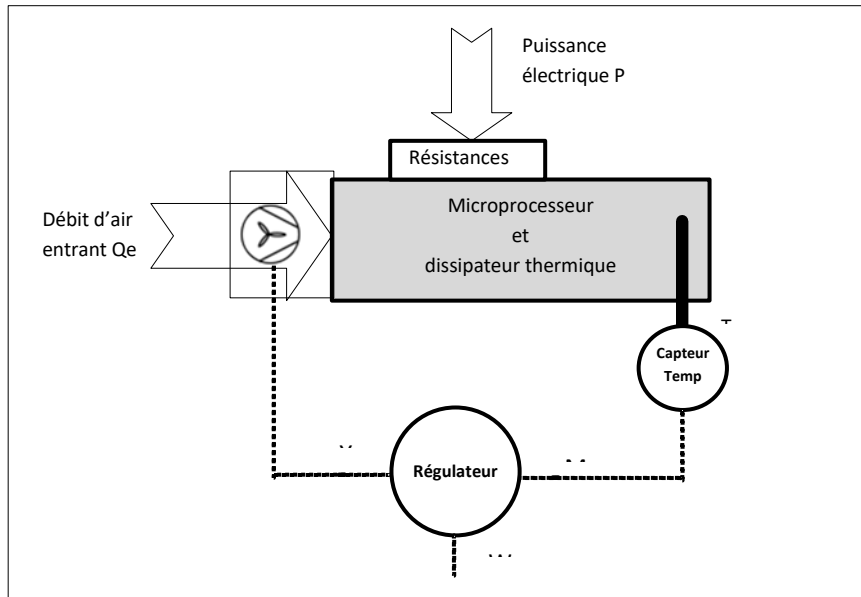
On simule un fonctionnement particulier du microprocesseur par une valeur de puissance totale de 50W.

15. Réaliser le montage donné par la Figure 2.
16. On fixe  $U_1 = 28.7$  V. Calculer la valeur de la tension  $U_2$  nécessaire pour obtenir une puissance totale de 50 W.
17. Régler ces valeurs de tension.
18. Ouvrir le programme Affichage\_T\_et\_commande\_ventilateur.ino.
19. A la ligne 10, fixer la commande de vitesse à 0% puis téléverser. Attendre quelques minutes la stabilisation de la température et la relever.
20. Répéter l'opération pour 4 à 5 valeurs de la commande de vitesse comprises entre 20 et 100%.
21. Tracer pour cette valeur de puissance, l'évolution de la température en fonction de la commande du ventilateur.



## ACTIVITÉ 3 : Régulation Tout Ou Rien

### Partie 1 : Régulation Tout Ou Rien à un seuil



On souhaite maintenir la température du microprocesseur à une consigne  $W = 60^{\circ}\text{C}$  en mettant en œuvre une régulation Tout Ou Rien à un seuil. Dans ce type de régulation, on sait que la commande ne peut prendre que deux états :  $Y_r = 0\%$  et  $Y_r = 100\%$ .

1. Compléter les deux phrases suivantes :
  - Si la température mesurée est inférieure à  $60^{\circ}\text{C}$  ALORS.....
  - Si la température mesurée est supérieure à  $60^{\circ}\text{C}$  ALORS.....
2. Le Programme 2 permet de mettre en œuvre la régulation TOR. Compléter les lignes 22 à 28 en vous aidant des résultats de la question 1.



```
regulation_TOR
1 // déclaration des constantes
2 int     portTemp = 0;           // la tension aux bornes de la CTN est envoyée sur le port température : port analogique 0
3 float   a = -0.1459 ;         // coefficient a
4 float   b = 120.17;           // coefficient b
5 int     portVentilateur = 3;   // la commande du ventilateur se fait par le port numérique 3
6 int     consigne = 60;        // valeur de la consigne
7
8 // Définition des variables
9 int     N = 0;                // variable nombre N sur port température
10 float  Temp;                 // variable température mesurée par le capteur
11 int     commandeVitesse = 0; // pourcentage de la commande de vitesse de 0 à 100%
12
13 void setup()
14 { // initialisation serial port
15   Serial.begin(9600);}
16   pinMode(portVentilateur, OUTPUT);}
17
18 void loop()
19 {
20   N = analogRead(portTemp); //Lecture N
21   Temp = a * N + b;         //calcul température T
22   if ( )
23   {
24     commandeVitesse = ;
25   }
26   if ( )
27   {
28     commandeVitesse = ;
29   }
30
31 //commande de la vitesse du ventilateur
32 // division par 500 pour que la variation de 0 à 20% soit ramenée à une variation virtuelle de 0 à 100%
33 analogWrite(portVentilateur,255*commandeVitesse/500);
34
35
36 Serial.print(consigne);
37 Serial.print(",");
38 Serial.print(Temp);
39 Serial.print(",");
40 Serial.println(commandeVitesse);
41 delay(1000);
42 }
```

### Programme 2

3. Fixer la puissance totale à 50 W, téléverser le programme régulation\_TOR.ino.
4. Observer dans le traceur série, l'évolution de la température et de la commande du ventilateur.
5. Faire une copie d'écran pour l'insérer dans le compte rendu et commenter les courbes obtenues.
6. La régulation TOR permet-elle d'atteindre l'objectif recherché ?
7. En conservant une puissance de 50 W, fixer maintenant la consigne à 50°C. Répéter les étapes 4 et 5.
8. Comparer les courbes obtenues pour une consigne de 60 et 50°C. Proposer une interprétation.
9. Quel inconvénient majeur présente la régulation TOR ?

On souhaite optimiser le fonctionnement du microprocesseur en fixant sa température à 25°C.

10. Modifier le programme, le téléverser et observer dans le traceur série l'évolution de la température et de la commande du ventilateur.
11. La température du microprocesseur respecte-t-elle la consigne imposée ? Proposer une explication en vous aidant de la caractéristique statique tracée dans l'ACTIVITÉ 2 : .

### Partie 2 : Régulation Tout Ou Rien à deux seuils

Lors de la mise en œuvre de la régulation TOR à un seuil, le fonctionnement du ventilateur consiste en une succession de phases d'arrêt et de phases de marche à pleine vitesse. La période de fonctionnement du ventilateur pouvait être très courte. Cela a des conséquences négatives sur la durée de vie du ventilateur. On cherche à limiter ce phénomène en mettant en œuvre une régulation TOR à deux seuils. Dans ce type de régulation, on fixe deux consignes : une consigne basse et une consigne haute.

12. Quel doit être l'état de la commande de vitesse du ventilateur si la température mesurée est supérieure à la consigne haute ?
13. Quel doit être l'état de la commande de vitesse du ventilateur si la température mesurée est inférieure à la consigne basse ?
14. En vous aidant des deux questions précédentes, compléter les lignes 23 à 29 du Programme 3.
15. Fixer la puissance totale à 50 W, téléverser le programme regulation\_TOR\_2seuils.



16. Observer dans le traceur série, l'évolution de la température et de la commande du ventilateur.
17. Faire une copie d'écran pour l'insérer dans le compte rendu et commenter les courbes obtenues.
18. La régulation TOR à deux seuils permet-elle de limiter le phénomène observé lors de la régulation TOR à un seuil ?
19. Quelle conséquence cela entraîne-t-il sur l'évolution de la température ?

```
regulation_TOR_2seuils
1 // déclaration des constantes
2 int     portTemp = 0;           // la tension aux bornes de la CTN est envoyée sur le port température : port analogique 0
3 float   a = -0.1459;          // coefficient a
4 float   b = 120.17;           // coefficient b
5 int     portVentilateur = 3;    // la commande du ventilateur se fait par le port numérique 3
6 int     consigneHaute = 60;    // valeur de la consigne haute
7 int     consigneBasse = 50;    // valeur de la consigne basse
8
9 // Définition des variables
10 int    N = 0;                 // variable nombre N sur port température
11 float  Temp;                  // variable température mesurée par le capteur
12 int    commandeVitesse = 0;   // pourcentage de la commande de vitesse de 0 à 100%
13
14 void setup()
15 { // initialisation serial port
16   Serial.begin(9600);
17   pinMode(portVentilateur, OUTPUT);}
18
19 void loop()
20 {
21   N = analogRead(portTemp); //Lecture N
22   Temp = a * N + b;         //calcul température T
23   if (                      )
24   {
25     commandeVitesse =      ;
26   }
27   if (                      )
28   {
29     commandeVitesse =      ;
30   }
31
32 //commande de la vitesse du ventilateur
33 // division par 500 pour que la variation de 0 à 20% soit ramenée à une variation virtuelle de 0 à 100%
34 analogWrite(portVentilateur, 255*commandeVitesse/500);
35
36
37 Serial.print(consigne);
38 Serial.print(",");
39 Serial.print(Temp);
40 Serial.print(",");
41 Serial.println(commandeVitesse);
42 delay(1000);
43 }
```

Programme 3



### ACTIVITÉ 4 : Correction Proportionnelle

L'ACTIVITÉ 3 : a permis de mettre en évidence au moins deux inconvénients de la régulation TOR. D'une part la non stabilité de la température du microprocesseur et d'autre part, le fonctionnement discontinu du ventilateur pouvant entraîner son usure prématurée. Pour éviter cela, une régulation proportionnelle est mise en œuvre. Dans ce type de régulation, la commande vitesse peut prendre toutes les valeurs de 0 à 100%. La commande vitesse est calculée avec l'une des deux relations suivantes :

$$Commande\ vitesse = K \times (Consigne - Mesure) \quad (\text{équation 1})$$

$$Commande\ vitesse = K \times (Mesure - Consigne) \quad (\text{équation 2})$$

#### Partie 1 : Mise en œuvre de la régulation proportionnelle

1. En analysant la Figure 3, justifier que l'équation 2 est celle qui doit être mise en œuvre pour permettre le fonctionnement du ventilateur lorsque la température dépasse la valeur de la consigne.

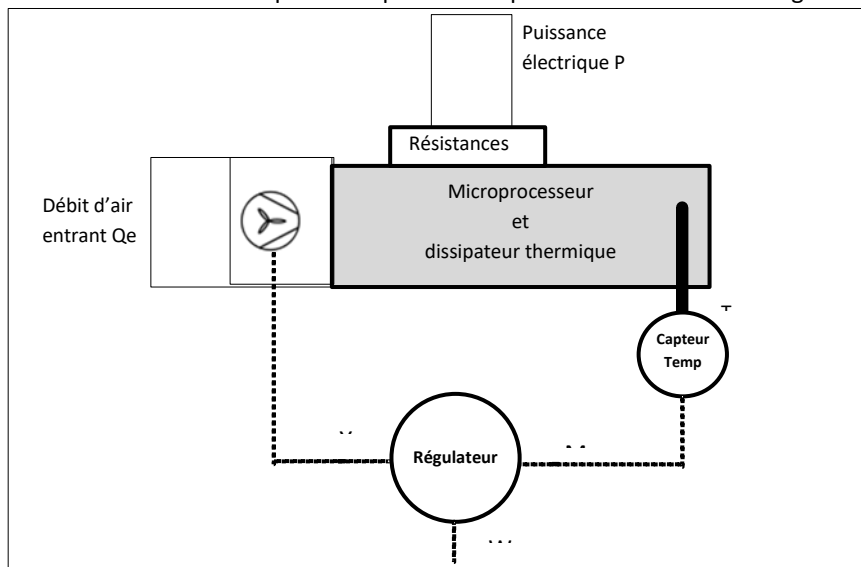


Figure 3

2. Téléverser le programme Regulation\_P.ino. (la consigne est fixée à 45°C. Le coefficient K est réglé à 5°C<sup>-1</sup>).
3. Alimenter R<sub>1</sub> et R<sub>2</sub> chacune avec une tension égale à 28V.
4. Ouvrir le traceur série et durant quelques minutes, laisser évoluer le procédé.
5. Observer et commenter les évolutions de la température et de la commande vitesse. Comparer avec ce qui avait été observé avec une régulation TOR.
6. Comparer la valeur de la température avec la consigne lorsque le procédé est stabilisé. L'écart relevé est nommé **écart statique**.
7. Citer un avantage et un inconvénient de la régulation P par rapport à la régulation TOR.

#### Partie 2 : Caractéristique statique du régulateur - Point de fonctionnement

8. On suppose que la mesure M correspondant à la température peut varier de 25 à 85°C. La consigne est fixée à 60°C. En utilisant l'équation 2 :  $Commande\ vitesse = K \times (Mesure - Consigne)$ , calculer les valeurs que prend la commande vitesse pour les valeurs 85 et 70°C de la température si on choisit le coefficient K égal à 1 °C<sup>-1</sup>.
9. Répondre de nouveau à la question 8 en choisissant K égal 2 °C<sup>-1</sup>.
10. Répondre de nouveau à la question 8 en choisissant K égal 5 °C<sup>-1</sup>.
11. Reporter les valeurs trouvées aux questions 8 à 10 dans les tableaux suivants et commenter l'influence de K sur la commande vitesse.

M = 85°C	K = 1°C <sup>-1</sup>	K = 2°C <sup>-1</sup>	K = 5°C <sup>-1</sup>
Yr			

M = 70°C	K = 1°C <sup>-1</sup>	K = 2°C <sup>-1</sup>	K = 5°C <sup>-1</sup>
Yr			



- 12. La température mesurée est maintenant égale à 50°C, le coefficient K étant fixé à 2°C<sup>-1</sup>, déterminer la valeur prise par la commande de vitesse.
- 13. Les valeurs possibles pour la commande vitesse permettent-elles de valider toutes les réponses des questions 10 et 12 ? Quelles précautions faudra-t-il prendre lors du calcul des valeurs de la commande vitesse ?
- 14. Repérer les numéros des lignes du programme Regulation\_P.ino donné ci-dessous qui permettent de calculer la commande et qui tiennent compte des précautions de la question 13.

```

regulation_P
1 // déclaration des constantes
2 int    portTemp = 0;           // la tension aux bornes de la CTN est envoyée sur le port température : port analogique 0
3 float  a = 0.1459;           // coefficient a
4 float  b = 120.17;           // coefficient b
5 int    portVentilateur = 3;   // la commande du ventilateur se fait par le port numérique 3
6 int    consigne = 45;         // valeur de la consigne
7 int    K = 5;                 // valeur du coefficient de proportionnalité
8
9 // Définition des variables
10 int   N = 0;                 // variable nombre N sur port température
11 float Temp;                 // variable température mesurée par le capteur
12 int   commandeVitesse = 0;   // pourcentage de la commande de vitesse de 0 à 100%
13
14
15 void setup()
16 {
17   // initialisation serial port
18   Serial.begin(9600);
19
20   pinMode(portVentilateur, OUTPUT);
21 }
22
23 void loop()
24 {
25   N = analogRead(portTemp); //Lecture N
26   Temp = a * N + b;         //calcul température T
27
28   // génération de la commande en regulation P
29   commandeVitesse = K*(Temp - consigne);
30
31   // conditions limites
32   if (commandeVitesse > 100) {commandeVitesse = 100 ;}
33   if (commandeVitesse < 0) { commandeVitesse = 0 ;}
34
35
36   //commande de la vitesse du ventilateur
37   analogWrite(portVentilateur,255*commandeVitesse/100);]
38
39   Serial.print(consigne);
40   Serial.print(",");
41   Serial.print(Temp);
42   Serial.print(",");
43   Serial.println(commandeVitesse);
44   delay(1000);
45 }

```

- 15. On souhaite maintenant tracer la caractéristique statique du régulateur avec une consigne de 45°C et en conservant un coefficient K=5°C<sup>-1</sup>. En vous appuyant sur les résultats des questions 10 à 13, compléter le tableau ci-dessous.

Température (°C)	30	35	40	45	50	55	60	65	70
Commande vitesse (%)									

- 16. On suppose que Windows, un traitement de texte, un navigateur et une vidéo sont en fonctionnement. La caractéristique du procédé est rappelée Figure 4. Sur ce graphique, tracer la caractéristique du régulateur (température en fonction de la commande) à partir des points calculés à la question 15.
- 17. Quelles sont les coordonnées du pont d'intersection des deux caractéristiques ?
- 18. Que représente ce point d'intersection ?
- 19. Comparer la température de ce point avec la valeur trouvée expérimentalement dans la Partie 1 : .
- 20. On fixe maintenant un coefficient K=10°C<sup>-1</sup> avec une consigne de 45°C. Compléter le tableau ci-dessous.

Température (°C)	30	35	40	45	50	55	60	65	70
Commande vitesse (%)									

- 21. Tracer sur la Figure 4, la nouvelle caractéristique du régulateur (température en fonction de la commande) à partir des points calculés.
- 22. Déterminer les coordonnées du nouveau point d'intersection. En déduire la valeur de l'écart statique.
- 23. Quelle amélioration a permis l'augmentation du coefficient K ?
- 24. Modifier la valeur de K pour qu'elle soit égale à 10°C<sup>-1</sup> dans le programme Regulation\_P.ino puis le téléverser.
- 25. Ouvrir le traceur série et durant quelques minutes, laisser évoluer le procédé.
- 26. Comparer alors l'écart statique mesuré avec celui obtenu lorsque K = 5°C<sup>-1</sup> (question 6).



27. L'observation de la valeur de température lorsque le procédé est stabilisé confirme-t-elle expérimentalement l'influence de K ?

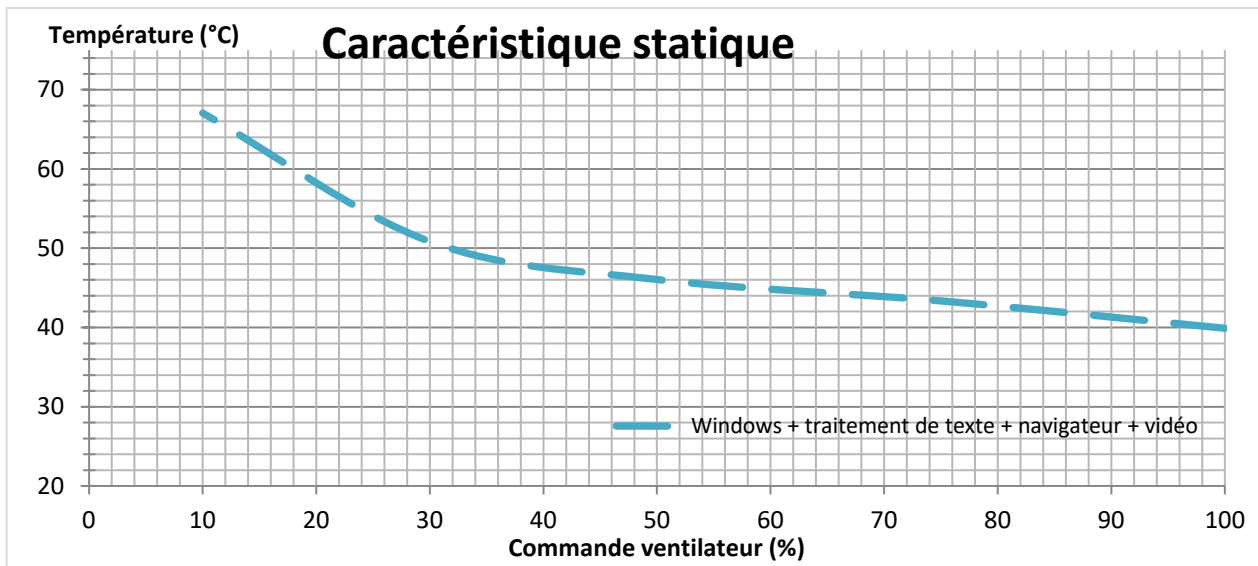


Figure 4

Partie 3 : Influence de la perturbation sur le point de fonctionnement

28. On conserve une consigne de 45°C et un coefficient  $K = 10^{\circ}\text{C}^{-1}$ . A partir du réseau de caractéristiques statiques donné Figure 5, déterminer la température du microprocesseur et l'écart statique pour les 3 modes de fonctionnement.

29. Proposer un protocole permettant de retrouver ces valeurs expérimentalement.

30. Mettre en œuvre le protocole.

31. En vous appuyant sur les observations expérimentales et sur le réseau de caractéristiques statiques donné Figure 5, expliquer comment se déplace le point de fonctionnement entre deux caractéristiques.

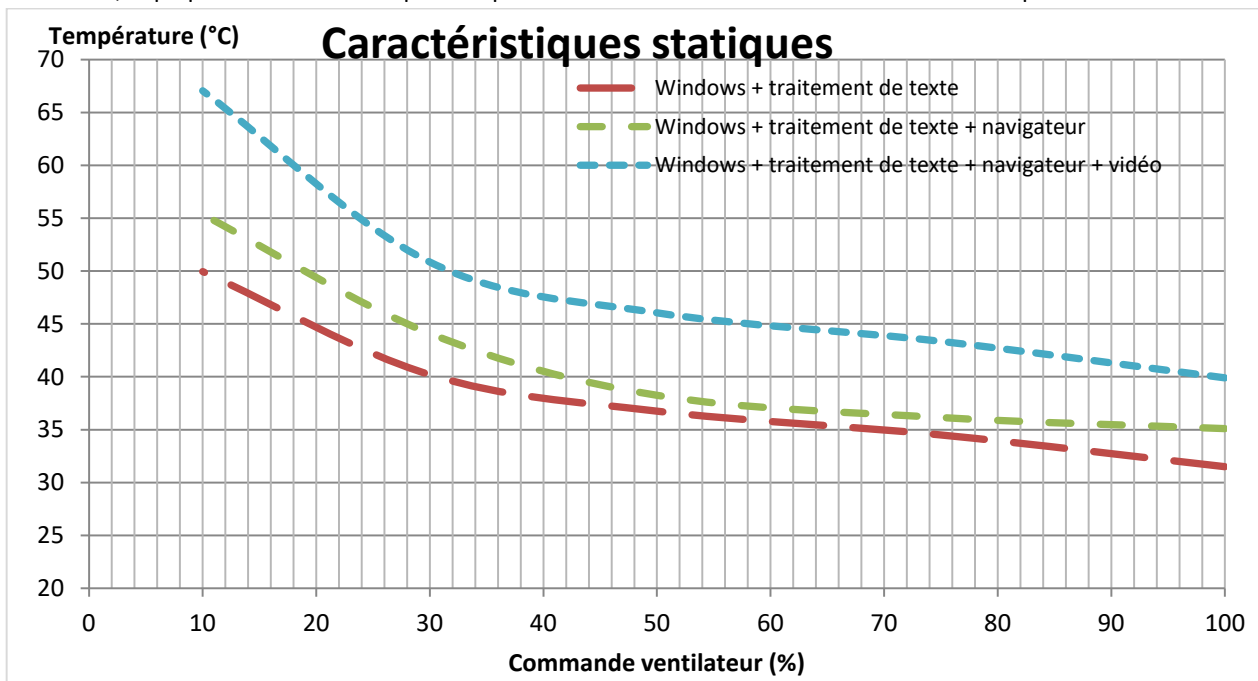


Figure 5





## ACTIVITÉ 5 : Correction Proportionnelle et Intégrale

L'activité précédente a permis de mettre en évidence un inconvénient de la correction proportionnelle. La mesure n'atteint jamais la consigne, il existe toujours un écart appelé écart statique. Pour remédier à ce problème, on met en œuvre une correction Proportionnelle et Intégrale (PI).

La correction PI s'appuie sur :

- la correction Proportionnelle (P) associée au facteur K
- la correction Intégrale (I) associée au facteur  $K_i$ .

### Partie 1 : Intérêt de la correction PI

On se place dans le cadre de la simulation d'un PC fonctionnant avec traitement de texte et un navigateur, c'est à dire une puissance dissipée de 50 W.

1. Après avoir vérifié que la consigne est égale à 45°C et que le facteur  $K_i$  est égal à 0,02, téléverser le programme "regulation\_PI.ino"
2. Patienter jusqu'à stabilisation du système. Estimer l'écart statique.
3. Que pouvez-vous en conclure sur l'intérêt de la correction PI par rapport à la correction proportionnelle (P) seule ?

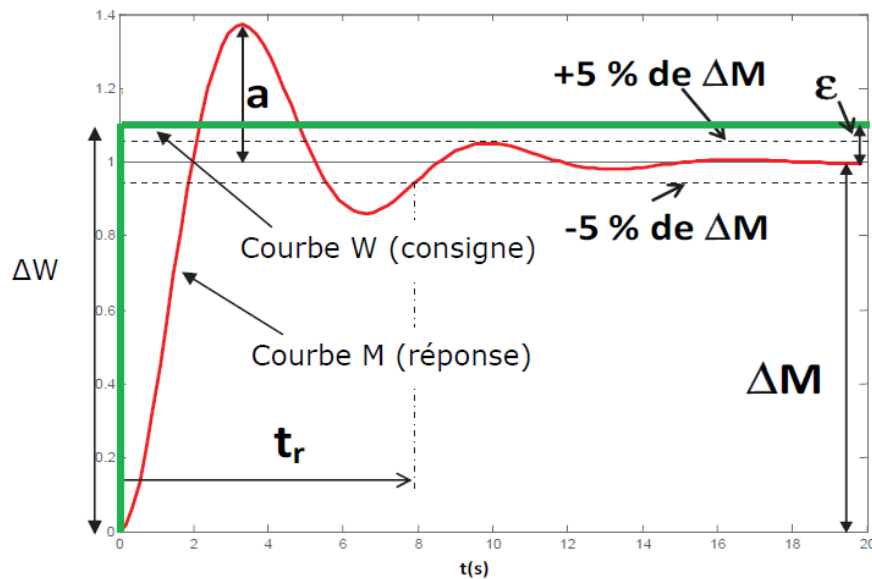
### Partie 2 : Performances de la correction PI

L'objectif de cette partie est d'observer l'influence de la valeur prise par le facteur  $K_i$  sur les performances de la boucle de régulation.

Pour comparer les performances d'une boucle de régulation, il existe trois critères présentés dans la *Figure 6*.



La réponse à un échelon de consigne  $\Delta W$  permet de déterminer les critères de performance de la boucle de régulation :



La **précision** s'évalue en mesurant l'écart  $\varepsilon = M - W$  quand le système s'est stabilisé après l'échelon de consigne. C'est l'écart statique.

On considère que le système réglé est précis si  $\varepsilon = 0$ .

La **rapidité** s'évalue en mesurant le temps de réponse à 5 % en boucle fermée ( $t_r$ ). Pour le déterminer, on trace les deux asymptotes à  $\pm 5\%$  de  $\Delta M$  (c'est à dire 95 % et 105 %). Le temps de réponse  $t_r$  est la durée séparant l'échelon de consigne et l'instant où la valeur de  $M$  reste toujours comprise entre ces deux asymptotes sans en ressortir.

L'**amortissement** est caractérisé par la valeur relative du dépassement :  $D = \frac{a}{\Delta M}$ . En général, le système est considéré comme bien amorti lorsque  $D$  est inférieur à 15 %.

Figure 6 : Critères de performance d'une boucle de régulation

4. Fixer la consigne à 45°C. et fixer  $K_i = 0,06. \text{ }^\circ\text{C}^{-1}\text{s}^{-1}$ . Laisser le système se stabiliser.
5. Modifier la consigne pour la fixer à 55 °C.
6. Laisser évoluer le système.
7. Estimer l'écart statique, le temps de réponse à 5% et le dépassement.
8. Doubler la valeur du coefficient  $K_i$  en fixant  $K_i = 0,12 \text{ }^\circ\text{C}^{-1}\text{s}^{-1}$ . Laisser le système se stabiliser.
9. Modifier la consigne pour la fixer à 45 °C.
10. Laisser évoluer le système puis fixer la consigne à 55°C.
11. Estimer l'écart statique, le temps de réponse à 5% et le dépassement.
12. Répéter les étapes précédentes en fixant  $K_i=0,03 \text{ }^\circ\text{C}^{-1}\text{s}^{-1}$ .
13. Comparer les trois résultats précédents pour en déduire l'influence de  $K_i$  sur les performances de la boucle de régulation.