

Séances 10 et 11 : chauffage d'appoint pour les abeilles "BeeWarm"

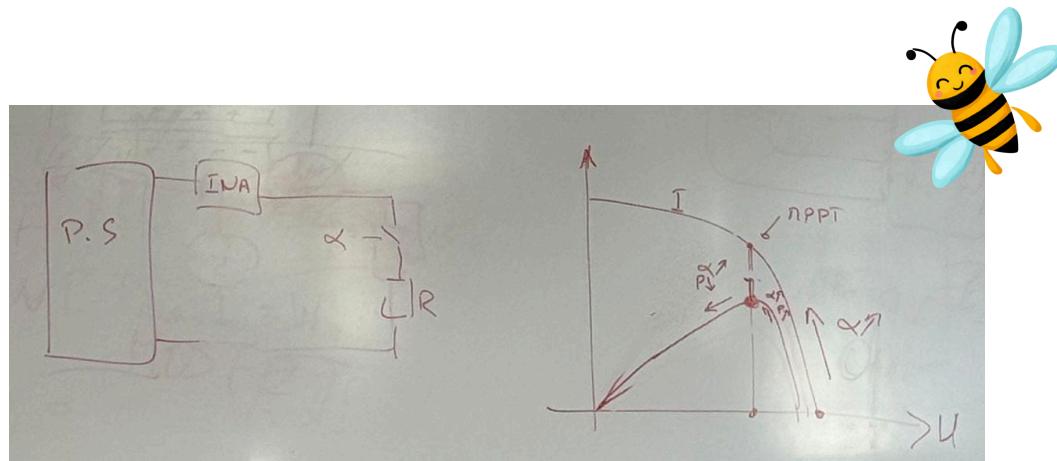


Introduction :

Pendant les séances 10 et 11, j'ai travaillé sur l'intégration du MPPT dans le programme Arduino afin d'optimiser la récupération d'énergie du panneau solaire. L'objectif était d'adapter la puissance consommée par le système de chauffage pour utiliser au mieux l'énergie disponible. En parallèle, j'ai également participé à l'impression 3D des différentes pièces mécaniques permettant de fixer et maintenir le panneau solaire sur le système.

1) Fonctionnement du MPPT dans le programme Arduino :

Le MPPT est une technique permettant d'extraire la puissance maximale d'un panneau solaire. En effet, la puissance fournie par un panneau dépend de la tension et du courant, et il existe un point optimal où la puissance $P = V * I$ est maximale.



Ce point varie en fonction de l'ensoleillement, de la température et de la charge connectée en sortie. Le rôle du MPPT est donc d'adapter la charge pour maintenir le panneau proche de ce point optimal.

l'ESP32 assure le fonctionnement du MPPT. Il mesure la tension et la puissance grâce au capteur INA219, puis ajuste le rapport cyclique du PWM commandant les résistances chauffantes. En modifiant le PWM, le système ajuste la puissance consommée, ce qui permet de trouver le point de fonctionnement optimal du panneau solaire.

Le programme fait tourner un cycle toutes les 3 secondes. À chaque cycle, il lit la température de la ruche (DS18B20) et la compare à la consigne, lit la tension et la puissance disponible via l'INA219, décide si on a le droit de chauffer (selon la température) et si on chauffe, il règle le PWM pour extraire un maximum d'énergie du panneau (MPPT).

Pour cela, deux boucles de contrôle travaillent ensemble.

La boucle autorisant le chauffage ne règle pas la puissance fine : elle décide juste d'activer ou de désactiver le chauffage avec hystérésis pour éviter de basculer sans arrêt autour de la consigne.

```
if (tRuche < (temperatureConsigne - bande)) {
    chauffe = true;
} else if (tRuche > (temperatureConsigne + bande)) {
    chauffe = false;
}
```

- Si la ruche est trop froide, alors : chauffe = true (PWM activé)
- Si elle est assez chaude, alors : chauffe = false (PWM = 0)

Au niveau de la boucle réglant le PWM, quand le chauffage est autorisé, on cherche le réglage de PWM qui permet de tirer le plus de puissance du panneau solaire.

Ici, le PWM agit comme une charge variable : plus le PWM est élevé, plus les résistances consomment, donc le MPPT du panneau se déplace.

Mais, le MPPT a besoin de savoir comment évoluent la tension et la puissance. Le INA219 mesure la tension, le courant et la puissance fournie par le panneau solaire grâce aux fonctions suivantes :

```
float busVoltage = ina219.getBusVoltage_V();
float current_mA = ina219.getCurrent_mA();
float power_mW = ina219.getPower_mW();
```

```
float P = power_mW;
float V = busVoltage;
```

Puis on calcule les variations par rapport au cycle précédent :

```
float dP = P - P_old;
float dV = V - V_old;
```

Le MPPT ici s'appuie sur le principe suivant : on teste un petit changement de direction de la variable **alpha**, on observe si la puissance monte ou descend, puis on continue ou on inverse, afin de converger vers le MPP et osciller légèrement autour de ce point.

```
if (dP > 0) {
    if (dV > 0) {
        direction = -1;
    } else if (dV < 0) {
        direction = 1;
    }
}
else if (dP < 0) {
    if (dV > 0) {
        direction = 1;
    } else if (dV < 0) {
        direction = -1;
    }
}
```

Si la dernière variation a amélioré la puissance, on continue globalement à pousser dans le sens qui rapproche du MPP. Si la puissance diminue, on change de sens.

Ensuite, on modifie **alpha** suivant l'ensoleillement. Si la lumière bouge peu ou si les mesures sont bruitées, **dP** et **dV** peuvent devenir très petits. Donc le code met un filtre : on ne bouge **alpha** que si les variations dépassent un seuil.

```
float seuil_dV = 0.01; // 10 mV
float seuil_dP = 50; // 50 mW

if (abs(dP) > seuil_dP && abs(dV) > seuil_dV) {
    // mettre à jour alpha
    alpha += direction * delta;
}
```

En effet, **alpha** représente le duty cycle normalisé ($0 \rightarrow 1$) et **delta** est le pas de variation.

```
static float alpha = 0.5;
static float delta = 0.02;
```

On limite ensuite **alpha** pour éviter 0% ou 100%, afin d'éviter de couper totalement la charge ou de saturer à fond :

```
if (alpha > 0.95) alpha = 0.95;
if (alpha < 0.05) alpha = 0.05;
```

Comme le PWM matériel ESP32 attend une valeur 0..255 (résolution 8 bits), alors:

```
int duty = alpha * 255;
```

Ensuite, on commande le PWM. Si la ruche a besoin de chaleur, alors le PWM sera égal à duty, sinon il reste à 0.

```
if (chauffe) {
    ledcWrite(SIGNAL1, duty);
    ledcWrite(SIGNAL2, duty);
} else {
    ledcWrite(SIGNAL1, 0);
    ledcWrite(SIGNAL2, 0);
}
```

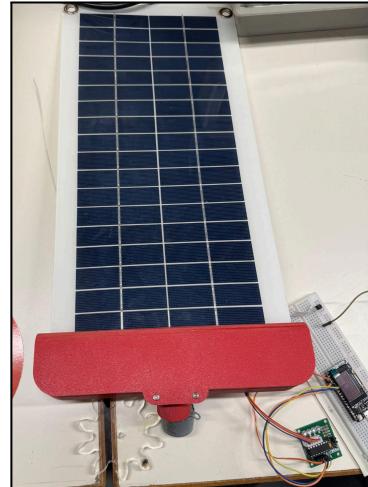
Finalement, on mémorise les valeur la puissance et de tension du MPP :

```
P_old = P;
V_old = V;
```

2) Impression 3D du support du panneau solaire + moteur :

On a réussi à imprimer en 3D les pièces de support du panneau solaire, permettant de stabiliser et faciliter sa rotation. Ces pièces avaient été modélisées au préalable, et mon travail a consisté à préparer et lancer leur fabrication afin de permettre l'intégration mécanique du système.

Une fois l'impression terminée, les pièces obtenues ont été vérifiées afin de s'assurer qu'elles correspondent bien aux dimensions attendues et qu'elles permettent le montage correct du panneau solaire et des autres composants.



Conclusion :

Cette séance a permis de mettre en place l'algorithme du MPPT, ainsi que de réaliser l'impression 3D des pièces nécessaires pour le support du panneau et du moteur. De plus, le système est désormais capable d'adapter le PWM en fonction de la puissance disponible.

Nous sommes maintenant en attente du schéma final de la carte électronique afin de pouvoir lancer la fabrication du circuit imprimé. Une fois la carte réalisée, des tests complets seront effectués pour vérifier le bon fonctionnement de l'ensemble du système, notamment la rotation du moteur, la commande PWM des résistances, ainsi que le bon fonctionnement de l'algorithme MPPT et la gestion globale de l'énergie solaire. Ces tests permettront de valider le fonctionnement réel du système et d'apporter les ajustements nécessaires si besoin.