

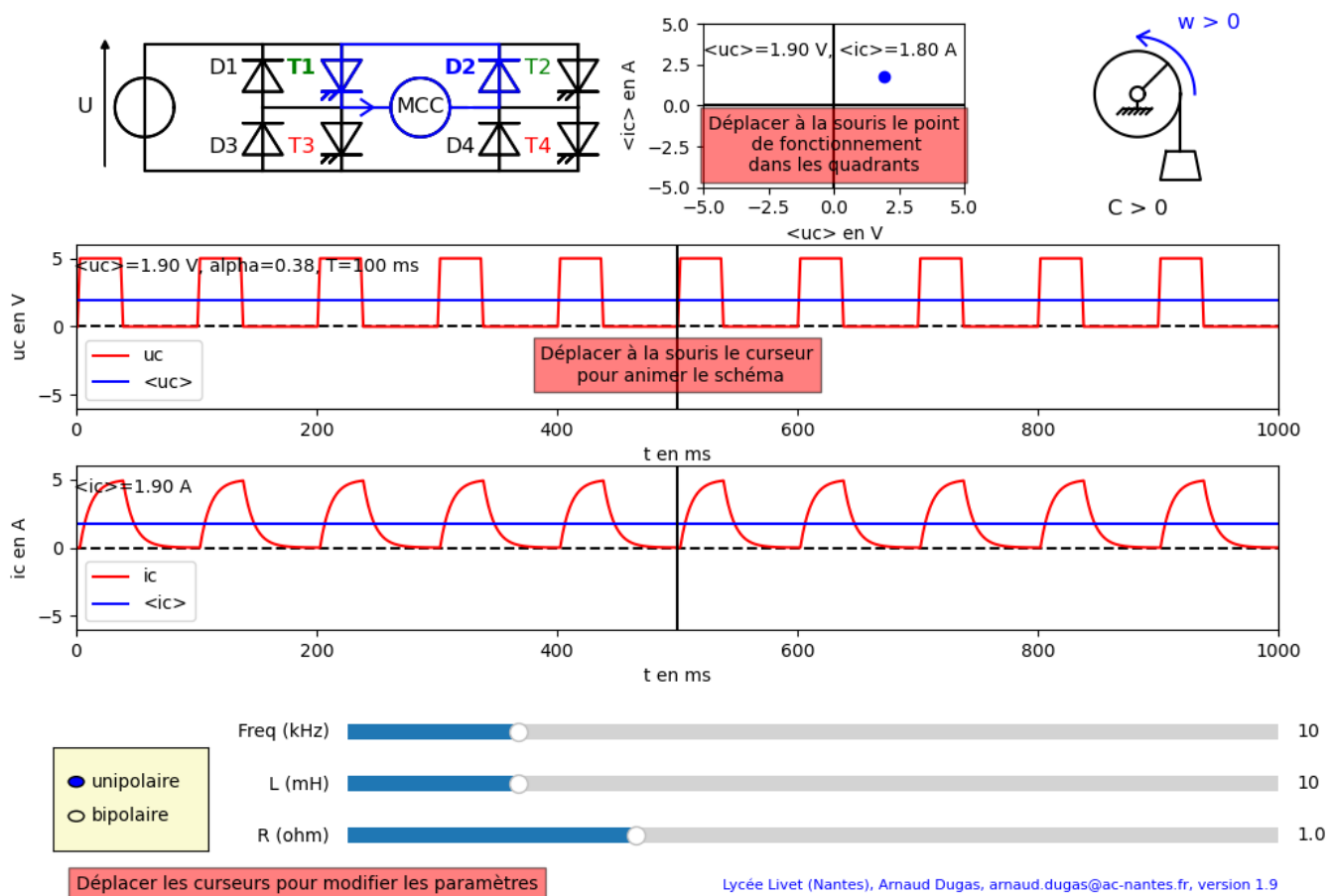
Simulateur python d'un hacheur

Document d'accompagnement

La ressource proposée est un programme informatique écrit en python qui permet de simuler le comportement d'un **hacheur unipolaire ou bipolaire** réversible en courant. Pour fonctionner, le programme nécessite uniquement l'installation de python (logiciel libre).

Le programme permet de faire varier les différents paramètres du hacheur comme la fréquence, le rapport cyclique, la résistance et l'inductance grâce à une interface graphique. Le programme permet de visualiser les signaux obtenus (courant, tension) ainsi que les transistors en commutation. Tous les paramètres sont ajustables par curseurs ce qui permet de voir l'évolution des signaux en temps réel.

Hacheur en commande unipolaire



Cette ressource est destinée aux étudiants et enseignants de sciences de l'ingénieur de CPGE et de BTS.

La ressource est composée :

- du programme python,
- d'une vidéo de prise en main rapide,
- d'un TD au niveau CPGE (avec corrigé) utilisant la ressource,
- d'un document d'accompagnement précisant les fonctionnalités et le prérequis d'installation.

1 INSTALLATION DE PYTHON

Pour fonctionner, le programme nécessite que python soit installé sur l'ordinateur accompagné des bibliothèques numpy, scipy et matplotlib. Le programme a été testé sur plusieurs versions de python et en particulier sur les versions :

Python 3.7.1, numpy 1.14.6, scipy 1.1.0, matplotlib 3.0.1 (version de 2018)

Python 3.9.10, numpy 1.21.5, scipy 1.7.3, matplotlib 3.5.1 (version de 2022)

Il est possible que sur des versions plus anciennes ou plus récentes il y est des incompatibilités.

Python peut être téléchargé ici : <https://www.python.org/downloads/>

Pour installer les bibliothèques, il faut utiliser le commande pip : `pip install matplotlib`, pour chacune des bibliothèques comme décrit ici : https://matplotlib.org/stable/users/getting_started/index.html.

Une autre solution consiste à utiliser une distribution contenant python et toutes les bibliothèques nécessaires comme WinPython, Anaconda ou EduPython par exemple.

La version Winpython 3.9.10 utilisée pour le développement de la ressource peut être téléchargée ici :

https://sourceforge.net/projects/winpython/files/WinPython_3.9/3.9.10.0/

C'est une version portable **qui ne nécessite donc pas de droits administrateurs**.

2 LANCENT DE L'APPLICATION

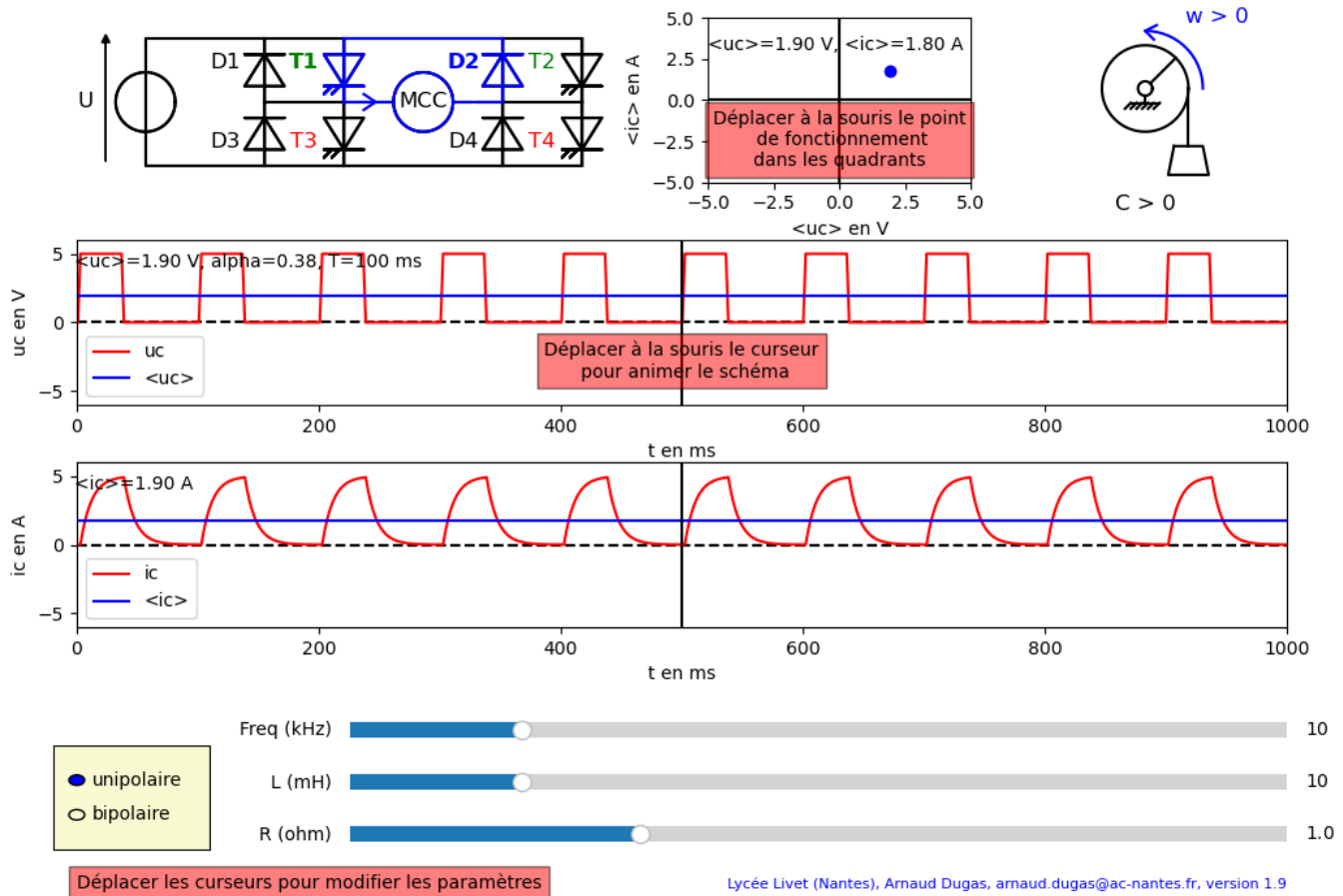
Si python a été installé, il suffit de double-cliquer sur le fichier en .py.

Si vous utilisez une version portable, il faut lancer **idle.exe**, puis ouvrir le fichier en **.py** dans idle (menu File/Open...) et enfin lancer le programme par la commande « Run Module » (menu Run).

3 UTILISATION

Une fois lancée, on obtient la fenêtre suivante :

Hacheur en commande unipolaire



En haut, on trouve de gauche à droite :

- ✚ le schéma électrique animé,
- ✚ le diagramme UI permettant de régler le point de fonctionnement,
- ✚ le schéma mécanique animé.

Au milieu, on trouve les chronogrammes $u_c = f(t)$ et $i_c = f(t)$.

En bas, on trouve les différents réglages possibles.

On peut déplacer le point de fonctionnement du diagramme UI, ce qui anime : les chronogrammes et le schéma mécanique. Sur les chronogrammes, on peut déplacer le curseur pour animer le schéma électrique.

Sur le schéma électrique, les transistors sont verts ou rouges suivant que le composant est activé ou non. Les composants sont en gras lorsqu'ils sont effectivement traversés par du courant.

Sur la figure ci-dessus, T1 et T2 sont actifs (ils sont en vert). T1 et D2 sont traversés par le courant (ils sont en gras). Le transistor T2, bien qu'actif n'est pas traversé par le courant (il est vert mais pas en gras).

Les curseurs en bas permettent de visualiser l'effet de la fréquence de hachage, de l'inductance du moteur et de la résistance du moteur. On peut comparer le fonctionnement en unipolaire et bipolaire.

On peut éventuellement augmenter le nombre de points des chronogrammes au détriment de la fluidité en modifiant le programme.

4 REMARQUES

Sur certaines machines anciennes, le programme peut manquer de la fluidité.

Sur un écran trop petit, certains éléments graphiques peuvent se chevaucher.