

## TP8 : Régime transitoire du premier ordre

### 1 Objectif du TP

L'objectif de ce TP est de faire l'acquisition de la tension aux bornes du condensateur au cours de sa charge à travers une résistance, et de confronter les résultats expérimentaux aux expressions théoriques.

*Ne pas oublier qu'une mesure physique doit toujours être associée à une incertitude expérimentale. Penser à lire la notice des appareils pour connaître l'incertitude liée aux valeurs qu'ils fournissent.*

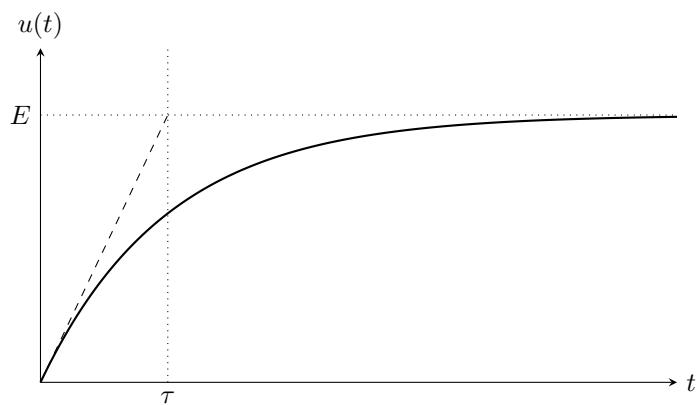
### 2 Acquisition de la tension aux bornes du condensateur

#### 2.1 Visualisation à l'oscilloscope

Réaliser un montage permettant de visualiser la charge du condensateur à l'oscilloscope.

Déterminer par cette méthode la constante de temps  $\tau$  de la charge et la confronter à la valeur théorique qui est :  $\tau = RC$ . Ne pas oublier d'évaluer l'incertitude associée à cette mesure.

On rappelle que la valeur de la constante de temps associée à la charge du condensateur peut être déterminée grâce à la méthode de la tangente représentée sur la figure ci-dessous.



#### 2.2 Acquisition informatique

À l'aide de l'interface d'acquisition et du logiciel *LatisPro*, faire la même acquisition de la charge du condensateur.

L'acquisition informatique permet de confronter directement l'évolution temporelle de la tension aux bornes du condensateur avec l'évolution prévue théoriquement.

On rappelle que la théorie prévoit que la tension  $u(t)$  aux bornes d'un condensateur chargé sous une tension  $E$  à travers une résistance  $R$  est :

$$u(t) = E \left( 1 - \exp \left( -\frac{t}{\tau} \right) \right) \quad (1)$$

Pour pouvoir traiter les données enregistrées par *LatisPro* il va falloir les importer dans Python. Pour cela on effectuera les étapes suivantes :

- Fichier → Exportation ;
- Format : TXT ; Décimal : point ; Entre les données : tabulation
- Enregistrer le fichier, par exemple en lui donnant le nom **data.txt**.

On pourra ensuite importer les données dans un script Python avec la commande suivante

```
import numpy as np
# skiprows permet d'ignorer la première ligne du fichier qui contient les titres des colonnes.
data = np.loadtxt("data.csv", skiprows=1)
```

Il faudra très certainement spécifier le chemin complet du fichier **data.csv** dans la fonction **loadtxt**. Si les valeurs du temps  $t$  et de la tension  $u(t)$  sont contenues dans les deux premières colonnes du fichier **data.csv**, on pourra les récupérer dans des tableaux numpy **t** et **u** avec les commandes

```
t = data[:,0]
u = data[:,1]
```

On pourra estimer la valeur de  $E$  grâce à la valeur limite atteinte par  $u(t)$ . Par exemple, si on veut prendre la valeur moyenne des 10 dernières valeurs de  $u(t)$ , on pourra écrire

```
E = np.mean(u[-10:])
```

On peut ensuite transformer les données pour obtenir une droite. Par exemple on pourra tracer la fonction

$$f(t) = \ln(E - u(t)) \quad (2)$$

en fonction de  $t$ . Dans ce cas la théorie prévoit que l'on obtient une droite décroissante de pente  $-\frac{1}{\tau}$ . Une régression linéaire devrait permettre de confronter la théorie et l'expérience et de déterminer la valeur de  $\tau$ .

On pourra enfin tracer sur le même graphique les mesures de  $u(t)$  ainsi que la fonction théorique avec les paramètres déterminés par la régression linéaire.