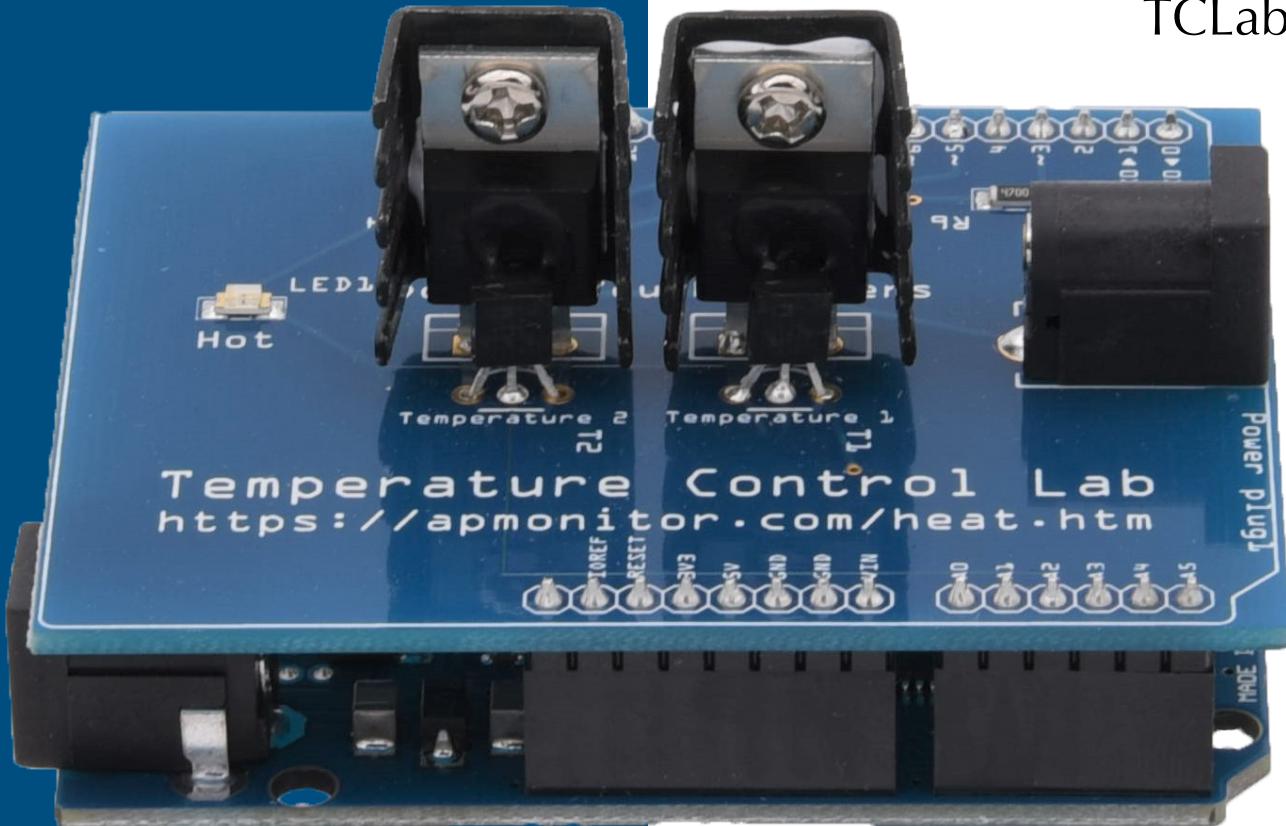


Control Theory

Rapport de laboratoire

Analyse & Contrôle de la plate-forme
TCLab



Groupe 8

Frédéric DRUPPEL 18053
Hugo DUBOIS 195347

01-04-2021

Sommaire

<u>Sommaire</u>	2
<u>1 Introduction</u>	4
<u>1.1 Explication du Kit et démarche</u>	4
<u>1.2 Objectifs du laboratoire</u>	4
<u>2 Expériences</u>	5
<u>2.1 Schéma bloc de l'expérience en boucle ouverte</u>	5
<u>2.2 Expérience sur MV (Step response)</u>	6
<u>2.3 Expérience sur DV (Step response)</u>	7
<u>2.4 Interprétation des graphes</u>	7
<u>3 Analyse des données d'expériences</u>	8
<u>3.1 Modèle d'approximation graphique</u>	8
<u>3.1.1 Résultats pour MV</u>	8
<u>3.1.2 Résultats pour DV</u>	10
<u>4 Création du contrôleur</u>	11
<u>4.1 Schémas blocs</u>	11
<u>4.1.1 PID</u>	11
<u>4.1.2 Feed-Forward</u>	12
<u>4.1.3 Schéma bloc du système avec PID et Feed-Forward</u>	13
<u>4.2 Discréttisation</u>	13
<u>4.3 Mode de fonctionnement</u>	14
<u>4.4 Réponses indicielles</u>	14
<u>4.4.1 PID</u>	14
<u>4.4.2 Lead-Lag avec délai (pour Feed-Forward)</u>	16
<u>4.5 « IMC Tuning »</u>	17
<u>5 Simulation</u>	18
<u>5.1 Boucle ouverte</u>	18
<u>5.2 Analyse fréquentielle du processus</u>	19
<u>5.3 Boucle fermée</u>	20
<u>6 Test sur la plate-forme TCLab</u>	21
<u>6.1 Plateforme TCLAB avec PID et FF en mode manuel puis mode automatique</u>	21
<u>6.2 Plateforme TCLAB en mode manuel avec FF</u>	22
<u>6.3 Plateforme TCLAB avec PID sans FF en mode manuel puis automatique</u>	23
<u>6.4 Plateforme TCLAB en mode manuel sans FF</u>	25
<u>7 Robustesse de la boucle</u>	26
<u>8 Problèmes rencontrés</u>	27
<u>8.1 Problème de détermination du modèle à utiliser pour implémenter l'IMC tuning.</u>	27
<u>8.2 Problème d'identification des paramètres dynamiques de (DV)</u>	28
<u>8.3 Problèmes de bruit de mesure & de stabilisation de PV</u>	28

<u>9 Conclusion</u>	<u>29</u>
<u>Sources</u>	<u>30</u>
<u>Annexes</u>	<u>31</u>

1 Introduction

1.1 Explication du Kit et démarche

Dans ce laboratoire nous avons utilisé un Kit de contrôle de température (Temperature Control Lab Kit) composé d'un « shield » avec deux sources de chaleur électroniques connectées chacun à un capteur de température, le tout branché à un Arduino pour gérer la communication avec un ordinateur. Les deux radiateurs ont été reliés entre eux par une plaque métallique (conducteur thermique) pour permettre un meilleur transfert de chaleur entre les deux sources, et ainsi l'influence de chacun des radiateurs sur les capteurs.

Nous y avons manipulé les variables MV et DV qui sont respectivement la puissance thermique du premier radiateur, et la puissance thermique du deuxième radiateur. Ce deuxième radiateur permet d'introduire une perturbation thermique « contrôlée » sur le premier.

En effet bien que le TCLab se comporte comme un processus non linéaire à variables multiples (HP_1 , HP_2 , T_1 et T_2) nous avons simplifié l'analyse de ce processus en considérant HP_2 [%] comme une perturbation sur la mesure du premier capteur (T_1).

Nous avons donc uniquement pris en compte la prise de mesure du premier capteur de température (T_1 [$^{\circ}$ C]).

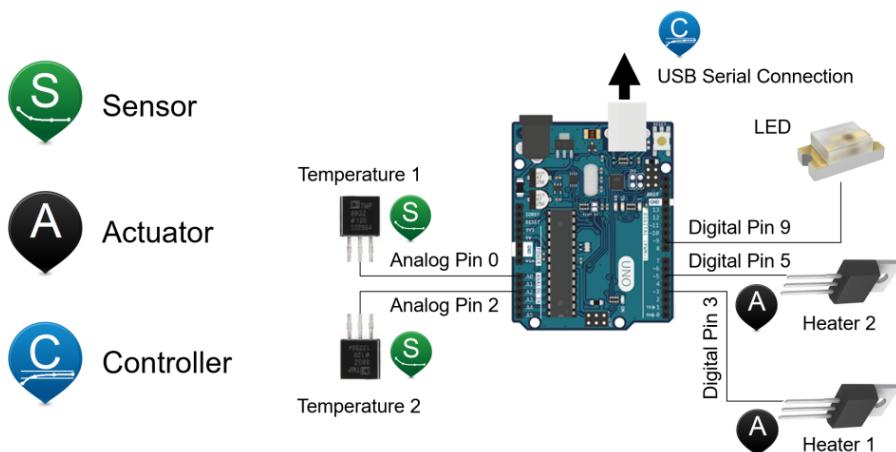


Figure 1: TCLab structure

1.2 Objectifs du laboratoire

Durant ce laboratoire nous avons analysé le processus non linéaire du contrôleur de température (TCLab). Pour ce faire nous avons utilisé l'environnement de développement JupyterLab ainsi que la librairie Python « `tclab` » qui nous ont permis d'implémenter une série de fonction (sous forme de package) nécessaire pour établir à terme un régulateur PID (avec ou sans action Feed-Forward).

2 Expériences

2.1 Schéma bloc de l'expérience en boucle ouverte

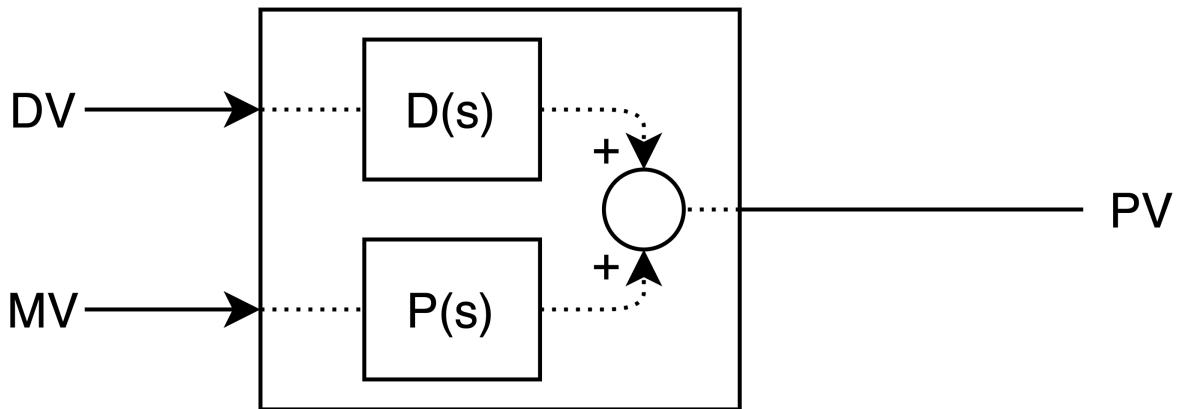


Figure 2 : Schéma bloc du processus TCLab

La figure 2 ci dessus représente la plateforme TCLab « brute »; On peut y voir deux entrées et une sortie.

La première entrée DV correspond sur TCLab à la source de chaleur H2, et nous permet d'introduire une perturbation connue. Ceci nous permettra d'implémenter dans le contrôleur une action Feed-Forward qui compensera cette perturbation. DV joue sur le processus $D(s)$ qui doit être identifié (voir section 3).

La seconde entrée MV correspond sur TCLab à la source de chaleur H1, et c'est cette entrée que nous allons manipuler avec le contrôleur pour réguler la sortie. MV joue sur le processus $P(s)$ qui doit également être identifié (voir section 3).

Enfin, la sortie PV est la somme des processus $P(s)$ et $D(s)$, qui correspond à la température du système.

2.2 Expérience sur MV (Step response)

Pour cette expérience, nous avons observé les variations en sortie de processus en boucle ouverte (PV [$^{\circ}\text{C}$], c'est à dire le capteur T1) en fonction des signaux d'entrée suivants :

- Une puissance de chauffe DV[%] (H2) à 50%, ce qui correspond à DV0 (constante durant l'expérience)
- Une puissance de chauffe MV[%] (H1) à 30% jusqu'à stabilisation de PV
- 10 secondes après la stabilisation de PV, MV subit un échelon de 40% et passe donc à 70% (valeur conservée jusqu'à ce que PV soit stable)
- Une fois PV stable à la nouvelle valeur de MV, MV subit un autre échelon de 40% vers le bas pour retourner à 30%

Nous pouvons observer le résultat de cette expérience dans la figure 2.

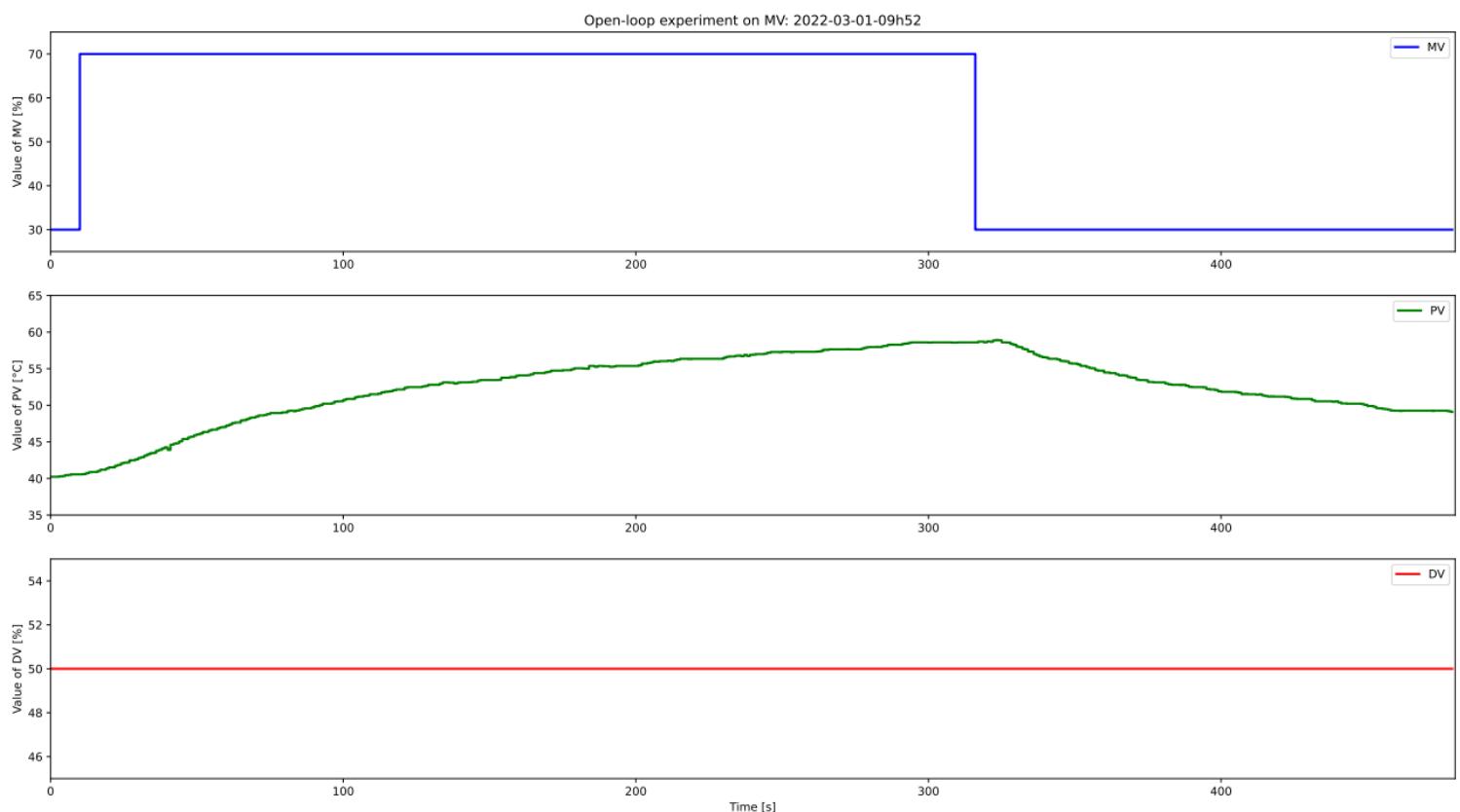


Figure 3 : Graphe de l'expérience sur MV

Afin d'optimiser la détermination des constantes de temps du processus, nous avons réitéré l'expérience à d'autres points de fonctionnement (par exemple à un point MV=60% avec un échelon de 15%). On reparle de ceci dans la section « 3 Analyse des données d'expériences » de ce rapport

2.3 Expérience sur DV (Step response)

Cette expérience se base sur le même principe de l'expérience sur MV, à l'exception qu'ici nous appliquons un échelon sur DV et gardons MV constant. Voici les signaux d'entrée de l'expérience sur DV (figure 3) :

- Une puissance de chauffe MV[%] (H1) 50% (constante durant l'expérience)
- Une puissance de chauffe de DV[%] (H2) à 30% jusqu'à stabilisation de PV
- 10 secondes après la stabilisation de PV, DV subit un échelon de 40% et passe donc à 70% (valeur conservée jusqu'à ce que PV soit stable)
- Une fois PV stable à la nouvelle valeur de DV, DV subit un autre échelon de 40% vers le bas pour retourner à 30%

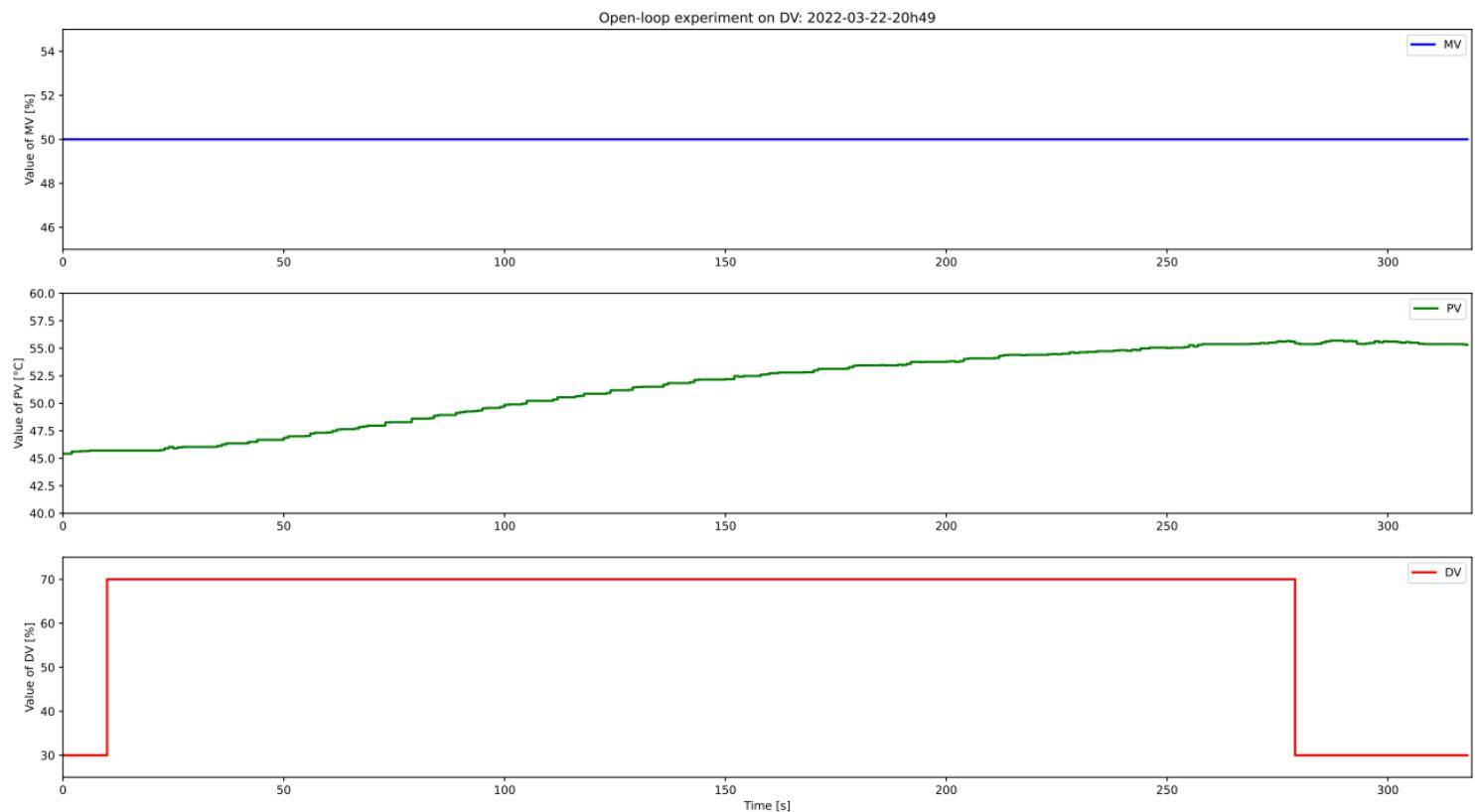


Figure 4 : Graphe de l'expérience sur DV

2.4 Interprétation des graphes

Les deux expériences ci-dessus comportent un pré-chauffage.

- 1) Premièrement il est facilement observable sur les figures 3 et 4 que le processus réel comporte bien un temps mort. En effet pour la figure 3, après avoir appliqué un échelon sur le signal d'entrée MV nous pouvons constater que la réponse PV (signal de sortie) met un certain temps $\theta = 1.72$ [s] avant de voir sa valeur être modifiée. De même à la figure 4, lorsque l'on fait subir un échelon sur le signal de perturbation DV où cette fois ci $\theta = 1.99$ [s].
- 2) Ensuite, on peut reconnaître la courbe d'un système dynamique du second ordre car une fois le temps mort dépassé, la pente de PV est nulle et augmente progressivement (contrairement à un système du premier ordre où la pente après le temps mort ne serait pas nulle).
- 3) Finalement, on voit que PV tend vers le gain statique du point de fonctionnement lorsqu'il se stabilise.

3 Analyse des données d'expériences

3.1 Modèle d'approximation graphique

Afin de déterminer les paramètres du processus, il a été possible d'utiliser les fonctions FOPDT et SOPDT mises à notre disposition via le package « package_DBR.py » toutes deux faisant appel à une autre fonction FO_RT. Cette dernière fonction utilise la méthode des rectangles à droite « Euler Backward difference » par défaut¹.

En utilisant ainsi les données des expériences dur DV et MV et les fonctions mentionnées ci-dessus, il a été possible de déterminer les différents paramètres (K_p , T , θ , T_1 et T_2).

3.1.1 Résultats pour MV

1) Pour une approximation selon un système respectant une fonction du premier ordre avec délai (FOPDT) respectant l'équation suivante :

$$PV = \frac{K_p}{Ts + 1} \cdot e^{(-\theta s)} \cdot MV$$

Equation 1 : Fonction de transfert du premier ordre avec délai (domaine de Laplace)

Nous avons obtenu les valeurs suivantes :

- $K_p = 0.56$
- $T = 166.20$ [s]
- $\theta = 1.72$ [s]

Où :

- K_p est appelé le gain statique du système, calculé en $s = 0$ et donc en $K_p = \frac{PV}{MV}$.
- T est appelé la constante de temps du système, telle que $\omega_c = \frac{1}{T}$ (la fréquence de coupure du système pour lequel le gain vaut -3dB).
- θ est appelé temps mort du processus, il s'agit du temps durant lequel, pour une variation du signal d'entrée, le signal de sortie du processus n'est pas modifié.

La deuxième manière de déterminer ces paramètres est le modèle d'approximation graphique appelée dans le cas d'un FOPDT, le modèle Broida.²

Via la méthode la plus simple nous avons obtenu :

- $T_g = T = 117.59$ [s]
- $T_u = \theta = 7.41$ [s]

Et via la méthode la plus complexe :

- $T_1 = 47.06$ [s]
- $T_2 = 72.55$ [s]
- $T = 5.5(T_2 - T_1) = 140.19$ [s]
- $\theta = 2.8T_1 - 1.8T_2 = 1.18$ [s]

¹ Slides 88/224 à 92/224 du cours de Control Theory. (Voir sources pour les références)

² La détermination graphique se trouve dans l'annexe 1

2) Toujours pour MV, mais cette fois-ci en utilisant une approximation par un système suivant une fonction du deuxième ordre avec délai (SOPDT) respectant l'équation suivante :

$$PV = \frac{K_p}{(T_{lag1}s + 1) \cdot (T_{lag2}s + 1)} \cdot e^{(-\theta s)} \cdot MV$$

Equation 2 : Fonction de transfert du second ordre avec délai (domaine de Laplace)

- $K_p = 0.57$
- $T_{lag1} = 172.41$ [s]
- $T_{lag2} = 2.85$ [s]
- $\theta = 1.99$ [s]

Et via la méthode graphique avec le modèle Van der Grinten³ pour un SOPDT :

$$\bullet T_{lag1} = T_g \frac{3ae - 1}{1 + ae} = 18.65 \text{ [s]}$$

$$\bullet T_{lag2} = T_g \frac{1 - ae}{1 + ae} = 7.85 \text{ [s]}$$

$$\bullet \theta = T_u - \frac{T_1 T_2}{T_1 + 3T_2} = 3.94 \text{ [s]}$$

Pour finir via la méthode graphique avec le modèle de Strejc :

$$P_S(s) = \frac{K_p}{(Ts + 1)^n} \cdot e^{(-\theta s)}$$

Equation 3 : Modèle de Strejc

Nous avons déterminé que le processus était du $n = 1^{er}$ ordre tel que :

$$a_n \leq \frac{T_u}{T_g} < a_{n+1}$$

C'est à dire :

$$a_1 = 0 \leq \frac{T_u}{T_g} \approx 0.063 < a_2 = 0.1$$

Pour lequel :

$$b_n = 1 = \frac{T_g}{T}, \text{ tel que } T = \frac{T_g}{b_n} = 117.59[\text{s}] \text{ avec } T_{u_{th}} = a_n \cdot T_g = 0[\text{s}]$$

Et :

$$\theta = T_u - T_{u_{th}} = T_u = 7.41[\text{s}]$$

³ La détermination graphique se trouve dans l'annexe 1

3.1.2 Résultats pour DV

1) Pour une approximation selon un système respectant une fonction du premier ordre avec délai (FOPDT), nous avons obtenus pour la dynamique de DV les valeurs suivantes :

- $K_d = 0.31$
- $T = 157.05$ [s]
- $\theta = 25.53$ [s]

2) Et pour finir toujours avec DV mais cette fois-ci en utilisant une approximation du système selon un système respectant une fonction du second ordre avec délai (SOPDT)

- $K_d = 0.29$
- $T_1 = 127.86$ [s]
- $T_2 = 21.74$ [s]
- $\theta = 12.38$ [s]

4 Création du contrôleur

4.1 Schémas blocs

4.1.1 PID

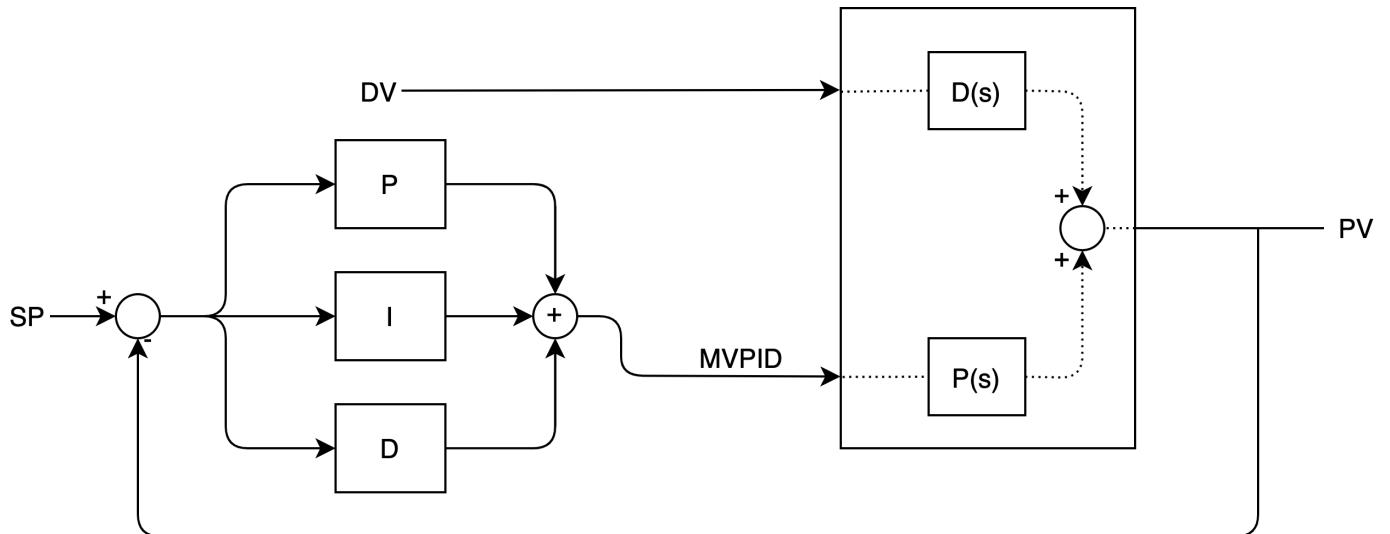


Figure 5 : Schéma bloc du régulateur PID

Ci-dessus nous pouvons voir dans la figure 5 le schéma bloc d'un système avec un contrôleur PID (décomposé en ses 3 actions).

En effet, « PID » est un acronyme de « Proportionnel », « Intégrateur » et « Dérivateur ». Chaque action a un paramètre permettant de contrôler leur influence dans la régulation. Dans ce laboratoire, ces paramètres découlent directement de la fonction de transfert du régulateur PID⁴ :

$$M V_{PID} = K_c \cdot \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}\right) E$$

Equation 4 : Fonction de transfert du régulateur PID

Où K_c est le gain du contrôleur, T_i est la constante de temps de l'action intégrante, T_d la constante de temps de l'action dérivative, α un facteur permettant de limiter le gain en haute fréquence à $\frac{1}{\alpha}$ (pour $s \rightarrow \infty$), E l'erreur entre SP et PV, et MV la valeur de sortie du contrôleur PID.

Les rôles des différentes actions sont les suivants :

- Proportionnel :
 - Permet de gérer l'« agressivité » du contrôleur. C'est l'action qui gère la vitesse de réponse.
- Intégrateur :
 - Permet de faire en sorte que l'erreur statique tende vers zéro en régime.
- Dérivateur :
 - Permet d'approximer l'erreur dans le « futur », et ainsi ne pas dépasser la consigne.

4 Dans certains cours et autres littératures (Source 7), on utilisera plutôt un gain pour chaque action, ce qui nous donnerait K_p, K_i, K_d

4.1.2 Feed-Forward

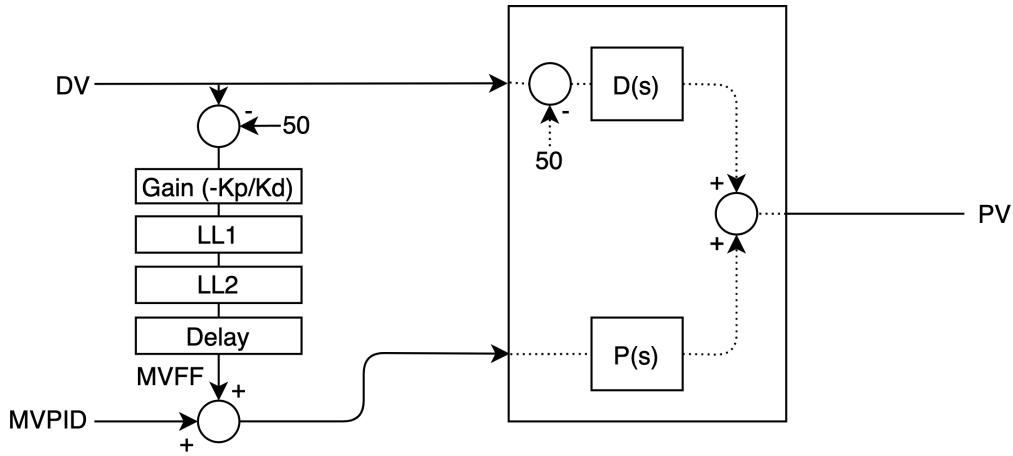


Figure 6 : Schéma bloc de l'action feed-forward

Ci-dessus nous pouvons voir dans la figure N le schéma bloc d'un système avec une action feed-forward. On voit dans ce système le feed-forward possède une partie gain, deux parties Lead-Lag, et une partie délai. Sa fonction de transfert est la suivante :

$$MV_{FF} = DV \cdot K_{FF} \cdot \frac{T_{P_1}s + 1}{T_{D_1}s + 1} \cdot \frac{T_{P_2}s + 1}{T_{D_2}s + 1} \cdot e^{-\theta_{FF}s}$$

Equation 5 : Fonction de transfert du feed forward

Où DV est la perturbation par rapport à DV_0 , K_{FF} le gain du feed-forward (correspondant à $\frac{-K_p}{K_d}$), T_{P_1} et T_{P_2} les constantes de temps du processus $\hat{P}(s)$, T_{D_1} et T_{D_2} les constantes de temps du processus $\hat{D}(s)$ ⁵, et θ_{FF} le temps-mort du feed-forward

⁵ $\hat{P}(s)$ et $\hat{D}(s)$ sont les approximations de $P(s)$ et $D(s)$ identifiées avec les modèles du premier ou second ordre.

4.1.3 Schéma bloc du système avec PID et Feed-Forward

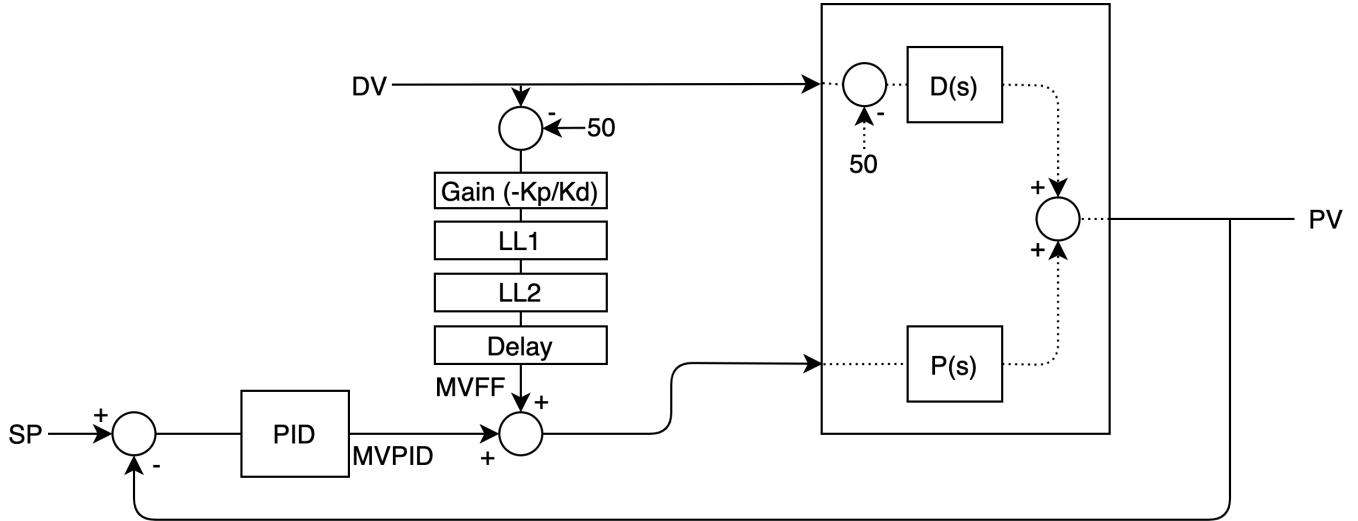


Figure 7 : Schéma bloc du système PID + FF

En combinant le système avec un régulateur PID et le système avec action feed-forward, nous obtenons le système représenté sur la figure 7.

4.2 Discréétisation

Afin de pouvoir implémenter le système dans un code, nous devons passer du domaine de Laplace (domaine en temps continu en s) au domaine de z (domaine en temps discret).

Pour ce faire, nous devons substituer dans les fonctions de transfert du PID et du feed-forward la variable s par un équivalent en z , et nous disposons (dans le cadre de ce cours) de 3 méthodes pour faire cela : La méthode « Euler Backward Difference » (EBD), la méthode « Euler Forward Difference » (EFD), et la méthode des trapèzes (TRAP).⁶

Sans entrer trop dans les détails, nous n'utiliserons que les méthodes EBD et TRAP pour discréétiser le système, car la méthode EFD peut rendre un système instable en temps discret, alors qu'il est stable en temps continu.

L'équivalence $s - z$ pour la méthode EBD est :

$$s = \frac{z - 1}{z T_s}$$

Et, l'équivalence $s - z$ pour la méthode TRAP est :

$$s = \frac{2}{T_s} \frac{z - 1}{z + 1}$$

Avec s étant la variable de Laplace, z étant la variable du domaine en Z , et T_s le temps d'échantillonnage.

Maintenant, il suffit de remplacer les « s » dans les fonctions de transfert par l'équivalence de la méthode choisie, et faire la transformée en Z^{-1} pour obtenir les équations aux différences⁷ qui nous permettront d'implémenter le tout dans un code en temps discret, en fonction d'un temps d'échantillonnage.

⁶ « Numerical integration methods », slide 88/224 du cours de Control Theory & Applications, Franky De Bruyne

⁷ Cet exercice est laissé à la convenance du lecteur

4.3 Mode de fonctionnement

Maintenant que les différents éléments du système ont été discrétisés, on peut implémenter le tout dans un code en Python. Néanmoins, nous devons également tenir compte de certains critères que le contrôleur doit respecter :

1. MV ne peut pas dépasser une valeur MV_{\min} et MV_{\max}
2. MV doit pouvoir être contrôlée de manière manuelle (donc il faut un mode manuel pour le contrôleur)
3. L'action feed-forward doit pouvoir être activée ou non selon le choix de l'utilisateur
4. Le mode de discréétisation doit pouvoir être choisi pour la partie PID et la partie feed-forward (EBD ou TRAP)

Une fois ces critères définis, le code du contrôleur peut être créé, et le système peut être simulé en tenant compte du schéma bloc sur la figure 7, avec quelques différences comme par exemple MV_{FF} qui sera à une entrée du contrôleur PID afin de respecter le critère 1 ci-dessus.

Le code peut-être retrouvé dans le fichier « package_LAB.py » avec un fichier de test « package_LAB.ipynb » permettant d'avoir des exemples rapides et explications des paramètres des méthodes & fonctions créées.

4.4 Réponses indicielles

4.4.1 PID

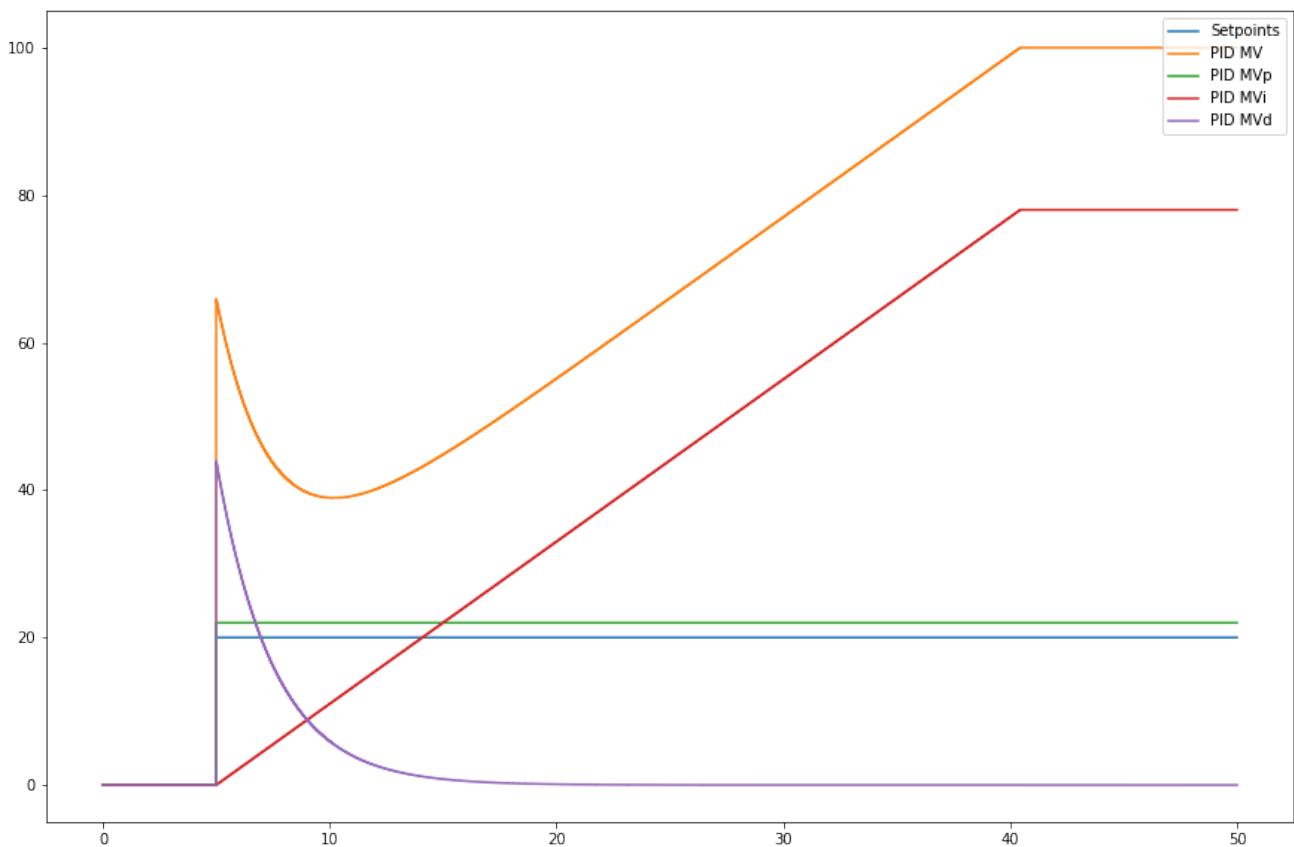


Figure 8 : Réponse indicuelle d'un contrôleur PID en simulation

On peut observer dans la figure 8 la réaction d'un contrôleur PID lorsqu'on applique un échelon à son entrée. Les paramètres dans le cas de la figure mentionnée ci-dessus sont les suivants⁸ :

$$K_c = 1.1, T_i = 10, T_d = 5, \alpha = 0.5$$

En analysant le graphe courbe par courbe nous pouvons observer que :

- A l'échelon sur le set point correspond une réaction immédiate de l'action proportionnelle (valeur de la variable MV_p) du PID pour atteindre une valeur équivalente à $K_c \cdot S_p$. Nous pouvons remarquer aussi qu'étant en simulation et comme vu au cours⁹, il n'y a pas de délai entre les variations de SP et les variations du PID (ce qui n'est pas le cas en réalité).
- Concernant l'action intégrale du PID, c'est-à-dire la valeur de la variable MV_i , nous pouvons constater que celle-ci suit une évolution graduelle (démarrant au moment de l'échelon sur SP) et continu jusqu'à ce que PV (non présente ici) atteigne la valeur de consigne en régime permanent. C'est à dire jusqu'à ce que l'erreur $E = SP - PV$ soit nulle (ce qui n'arrive pas dans ce cas-ci car nous sommes en boucle ouverte).
- L'action dérivative du PID quant-à elle suit une évolution brutale au moment de l'échelon sur SP car au même moment MV (en sortie du PID) est nul. Comme expliqué au cours le rôle de cette action est d'anticiper l'évolution de l'erreur du contrôleur $E = SP - PV$. Il est alors simple de comprendre qu'à une évolution brutale de SP (pour un PV constant à cet instant), et donc pour une évolution de E , correspond une augmentation de la valeur MV_d (voir développement de Taylor du 1^{er} ordre).

⁸ Le fichier « package_LAB.ipynb » possède une section avec des curseurs permettant de modifier ces paramètres et afficher le résultat sur un graphique de manière dynamique.

⁹ Slides 155/224 à 177/224 du cours de Control Theory. (Voir sources pour les références)

4.4.2 Lead-Lag avec délai (pour Feed-Forward)

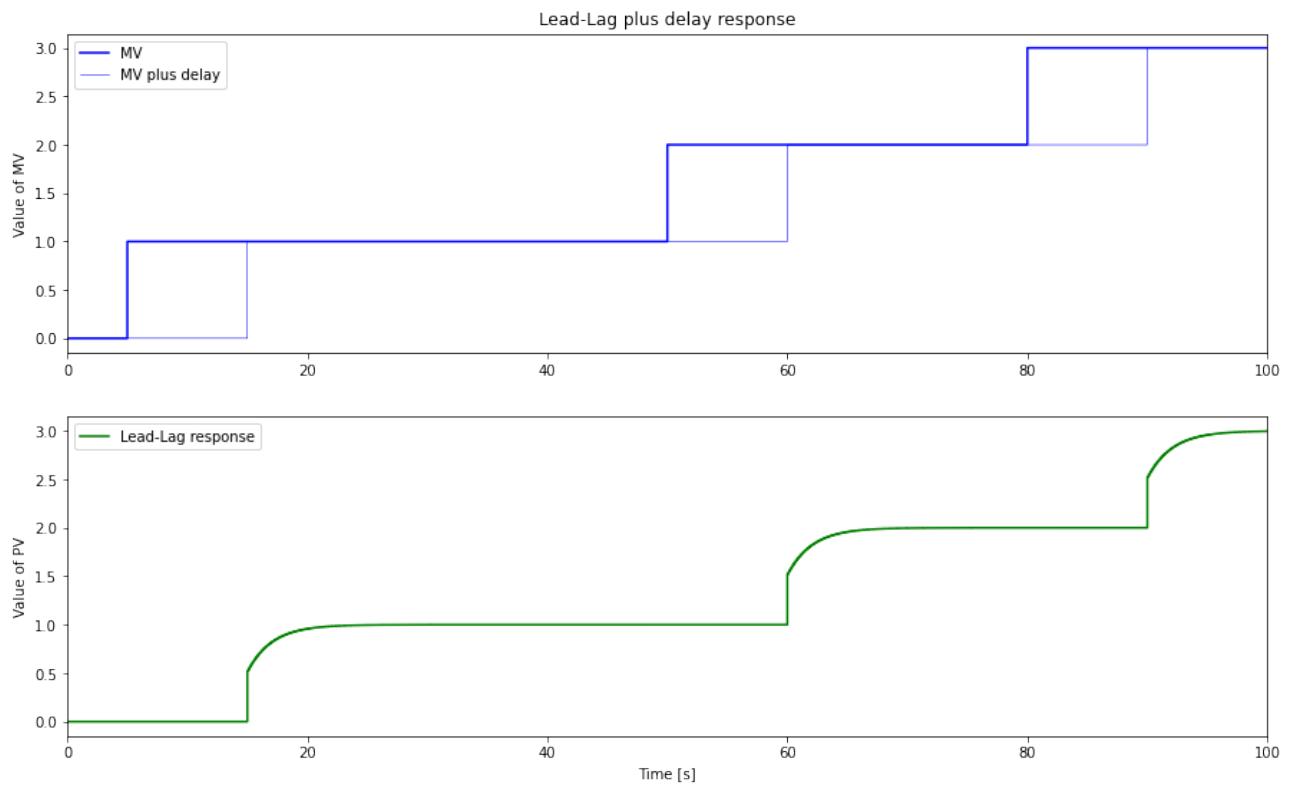


Figure 9 : réponse d'un système lead-lag avec délai à plusieurs échelons

On peut observer dans la figure 9 la réaction d'un système lead-lag à plusieurs échelons. Les paramètres dans ce cas-ci sont les suivants¹⁰ :

$$K_p = 1, T_{lead} = 1, T_{lag} = 2, \theta = 10$$

¹⁰ Le fichier « package_LAB.ipynb » possède une section avec des curseurs permettant de modifier ces paramètres et afficher le résultat sur un graphique de manière dynamique.

4.5 « IMC Tuning »

Afin que notre processus puisse rester dans une limite de tolérance de production et afin de minimiser l'impact du bruit DV sur le signal de sortie PV, il a été nécessaire d'implémenter l'IMC tuning.

L'acronyme IMC signifie « Internal Model Control ».

Pour ce faire, il nous a fallu déterminer à quel modèle nous avions à faire. Cette détermination se fait par expérimentation sur le processus.

Ainsi en nous basant sur nos résultats¹¹ et sur des valeurs de α et de γ telles que :

$$\alpha = 0.5 \text{ et } \gamma = 0.4$$

Dans notre cas les valeurs de $T_{lag1} = 172.41$ [s] et $T_{lag2} = 2.85$ [s] nous laissent apercevoir que la formule de l'équation 2 tend à se simplifier par le fait que $T_{lag2} < < T_{lag1}$ pour donner une équation similaire au cas G¹², c'est à dire :

$$\frac{Ke^{-\theta s}}{Ts + 1}$$

Ainsi, nous obtenons en sortie de l'IMC tuning nous obtenons les valeurs de :

- $K_c = 4.26$
- $T_i = 172.4$
- $T_d = 0$

Ces valeurs sont alors reprises en paramètres du contrôleur PID.

¹¹ Voir 3.1.1 Résultats pour MV et 3.1.2 Résultats pour DV

¹² Slides 186/224 à 187/224 du cours de Control Theory. (Voir sources pour les références)

5 Simulation

5.1 Boucle ouverte

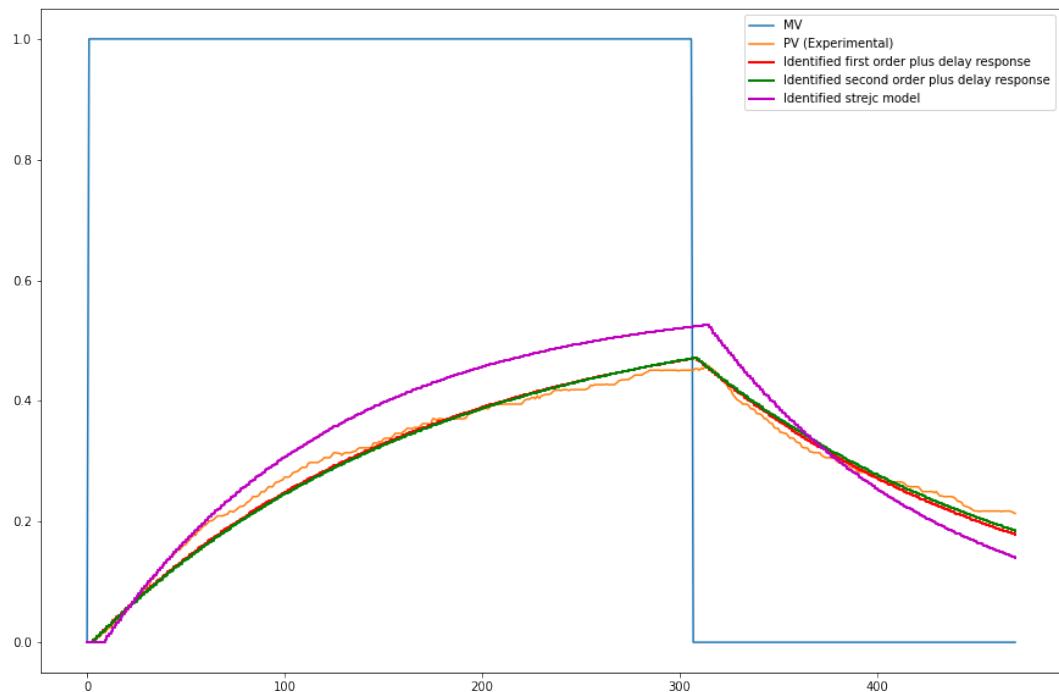


Figure 10 : Réponse de PV à un échelon sur MV (valeurs expérimentales et modèles mathématiques)

Comme nous avons trouvé les paramètres dynamiques du système dans la section 3, nous pouvons vérifier que ces paramètres correspondent (ou s'approchent) des valeurs expérimentales enregistrées dans les expériences de la section 2.

On voit dans la figure 10 en jaune les valeurs expérimentales, en rouge la réponse du modèle du premier ordre avec délai, en vert la réponse du modèle du second ordre avec délai, et en mauve la réponse du modèle de Strejc¹³. De ces courbes nous pouvons voir que le modèle du premier ordre colle le mieux aux valeurs expérimentales, suivi de près du modèle du second ordre.

13 La figure 10 a été obtenue à partir du fichier « Simulation_OLP.ipynb »

5.2 Analyse fréquentielle du processus¹⁴

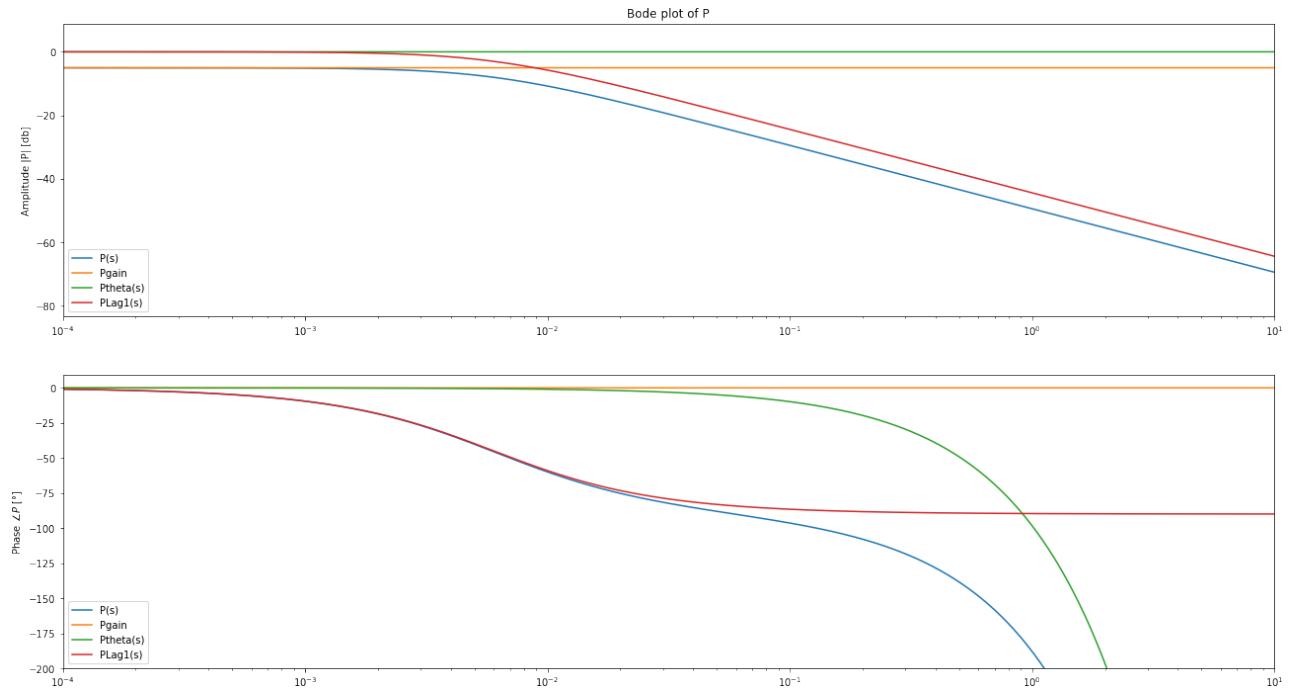


Figure 11 : Diagramme de Bode du modèle du premier ordre avec comme paramètres :
 $K_p = 0.56$, $T_1 = 166.2$, $\theta = 1.71$

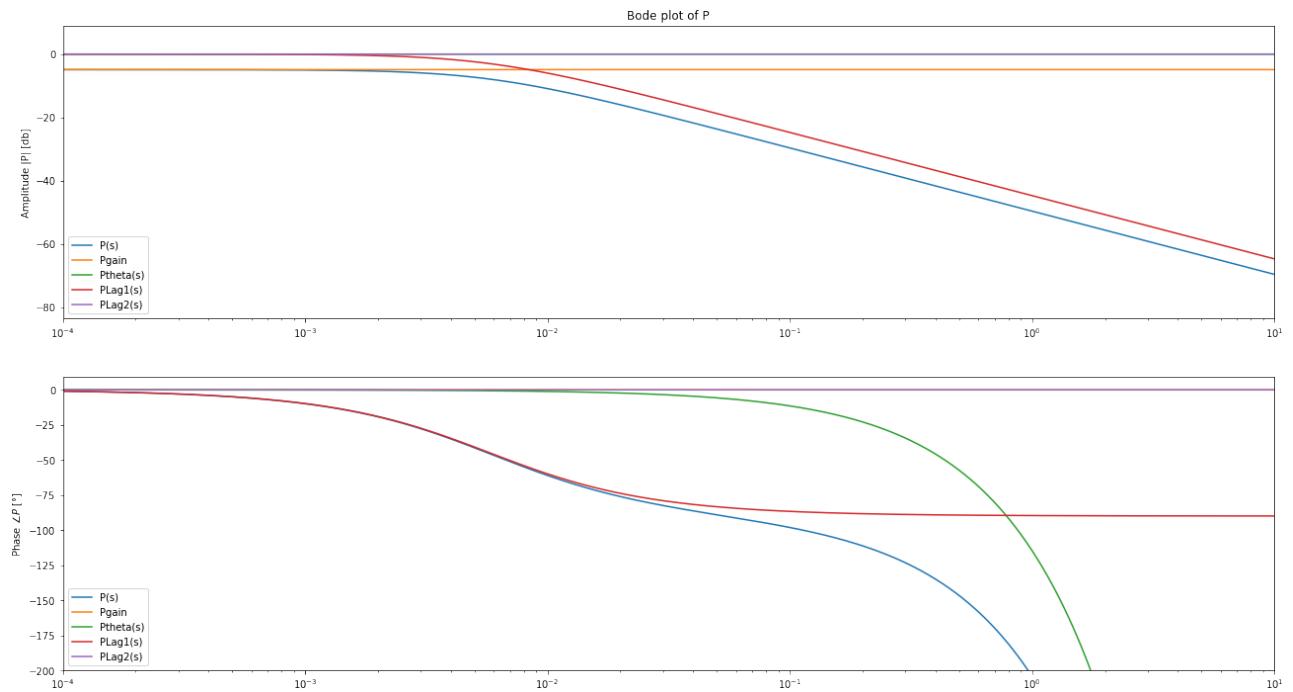
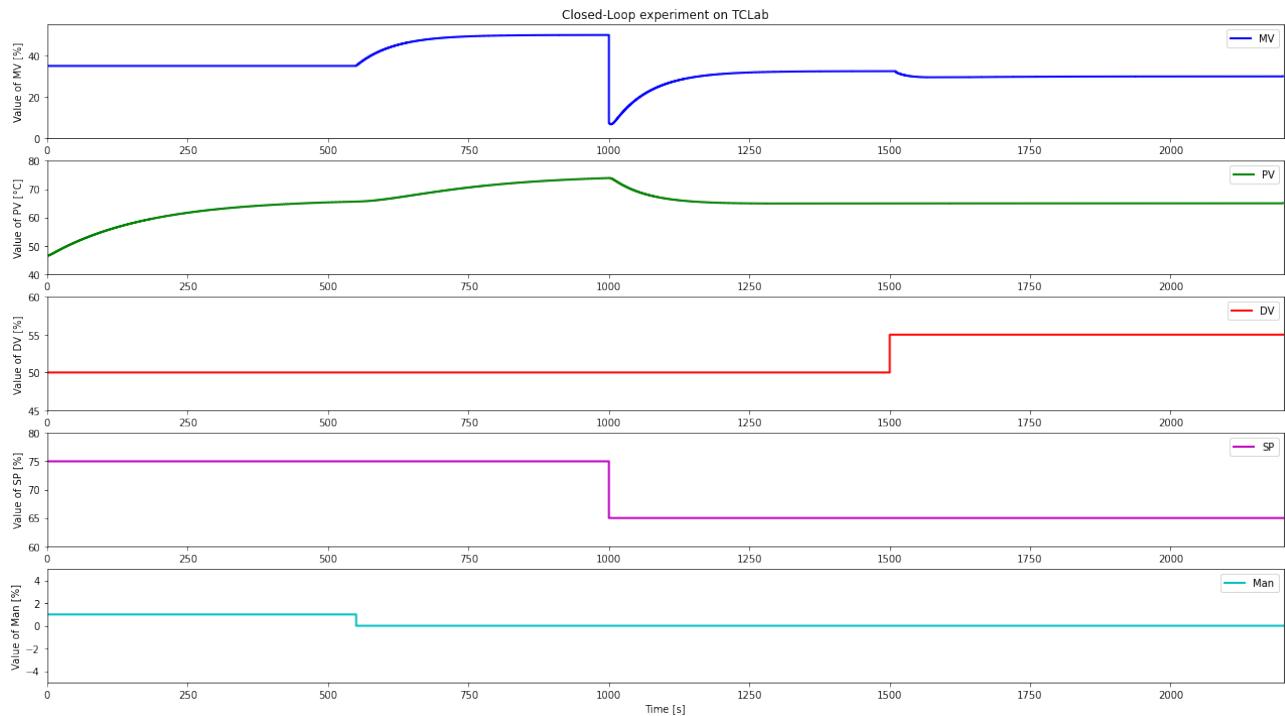


Figure 12 : Diagramme de Bode du modèle du premier ordre avec comme paramètres :
 $K_p = 0.57$, $T_1 = 172.4$, $T_2 = 2.85$, $\theta = 1.99$

14 Les diagrammes de Bode (figures 11 et 12) ont été obtenues à partir du fichier « Simulation_OLP.ipynb »

5.3 Boucle fermée¹⁵



Sur la figure 13, nous observons un comportement du système selon les étapes suivantes¹⁶ :

1. 0s à 550s : Mode manuel avec MV = 35%
2. 550s : Passage en mode automatique avec SP = 75°C
3. 1000s : Variation sur SP (passe de 75°C à 65°C)
4. 1500s : Variation sur DV (Passe de DV0 (50%) à DV0+5 (55%)

On observe sur la courbe de MV et PV :

- Pour l'étape 1 :
 - MV est constante car le système est en mode manuel.
 - PV réagit comme si le système était en boucle ouverte avec un échelon sur MV.
- Pour l'étape 2 :
 - MV varie en fonction des paramètres du contrôleur PID (et de SP).
 - PV tend vers le SP (qui vaut donc 75°C).
- Pour l'étape 3 :
 - MV varie en fonction du nouveau SP (65°C) qu'on applique.
 - PV tend vers le nouveau SP de 65°C
- Pour l'étape 4 :
 - MV varie grâce à l'action feed-forward du système qui a pris en compte la variation de DV.
 - PV ne varie pas et reste au SP de 65°C, ce qui prouve que le feed-forward a bien été implémenté.

¹⁵ Les diagrammes ont été obtenu à partir du fichier « Simulation_CLP_PID_FF.ipynb »

¹⁶ Le résultat de la simulation représentée par la figure 13 correspond au scénario 4 dans le fichier « Simulation_CLP_PID_FF.ipynb ». D'autres scénarios avec d'autres paramètres sont présents dans ce fichier afin de simuler un système avec ou sans feed-forward, mode manuel, etc.

6 Test sur la plate-forme TCLab

6.1 Plateforme TCLAB avec PID et FF en mode manuel puis mode automatique

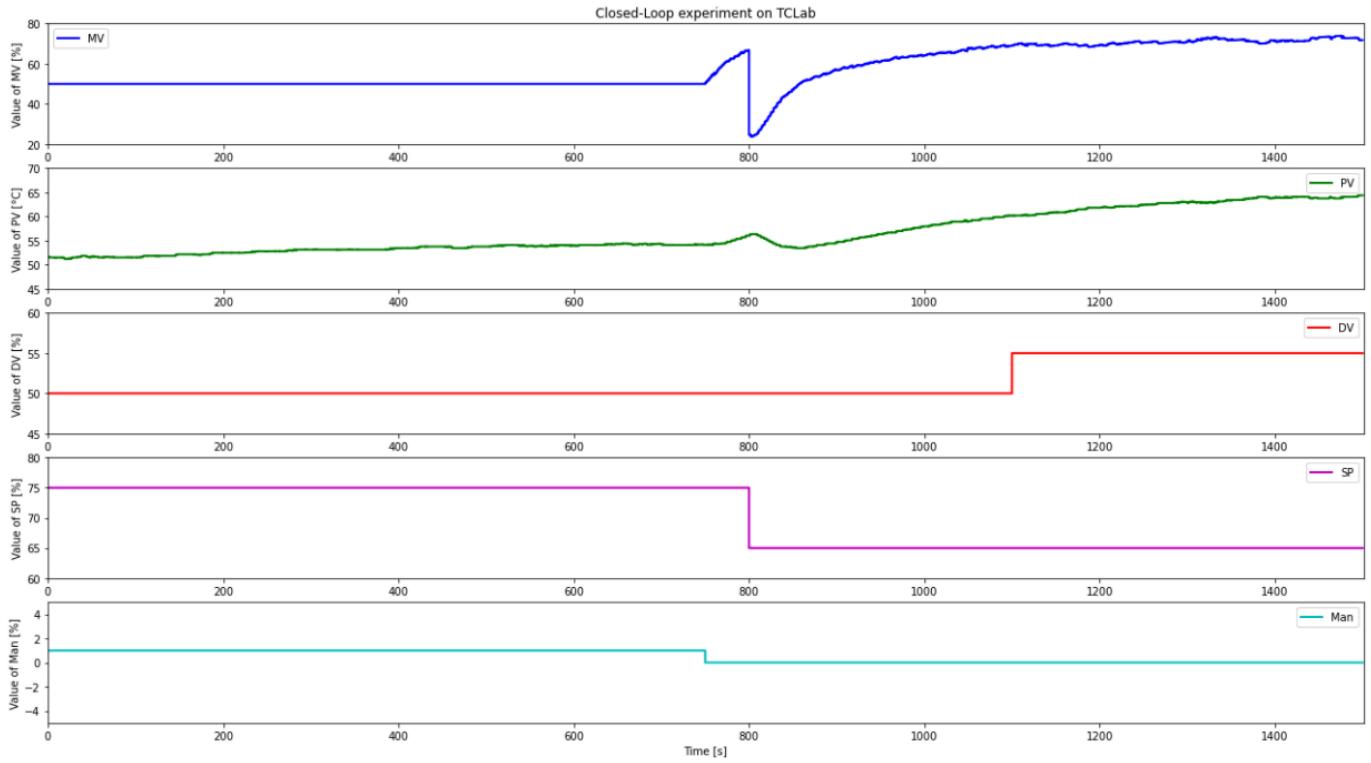


Figure 14 : Graphes des données du PID + FF sur le TCLAB avec passage en mode automatique

Cette expérience nécessite une combinaison du PID avec feed-forward et du rendu graphique en live des paramètres du processus. Elle a été réalisée sur un temps de simulation $T_{sim} = 1500$ [s] et avec une température de démarrage égale à 51°C.

Sur la figure 14, nous observons un comportement du système selon les étapes suivantes :

1. 0s à 750s : Mode manuel avec $MV = 50\%$
2. 750s : Passage au mode automatique (système en boucle fermée)
3. 800s : Variation sur SP (passe de 75°C à 65°C)
4. 1100s : Variation sur DV (Passe de DV0 (50%) à DV0+5 (55%)

Interprétations graphiques :

- Pour l'étape 1 :
 - MV est constante car le système est en mode manuel.
 - PV réagit comme si le système était en boucle ouverte avec un échelon sur MV.
- Pour l'étape 2 :
 - MV varie maintenant grâce au PID pour amener PV à la consigne SP.
 - PV varie en fonction de la sortie du contrôleur PID pour atteindre SP (75°C).
- Pour l'étape 3 :
 - MV varie brutalement grâce au PID suite à la variation brutale de la valeur de consigne.
 - PV diminue suite à la variation de MV pour tendre vers la nouvelle valeur de consigne SP=65°C.
- Pour l'étape 4 :
 - MV corrige la perturbation générée par DV via le feedforward.
 - PV ne subit pas de perturbation et continue à tendre vers la valeur de consigne.

6.2 Plateforme TCLAB en mode manuel avec FF

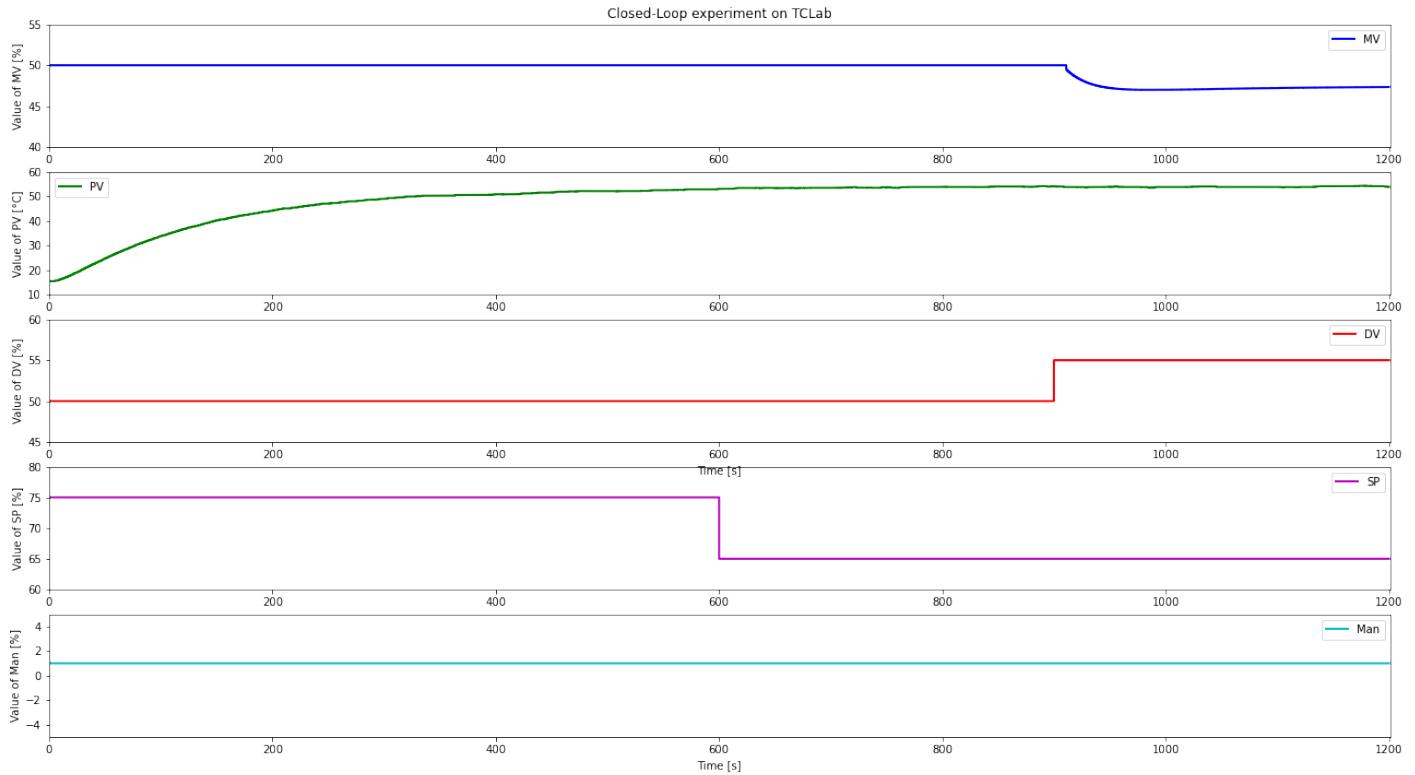


Figure 15 : Graphes des données du PID + FF sur le TCLAB avec mode manuel

Cette expérience nécessite une combinaison du PID avec feed-forward mais avec un mode manuel (boucle ouverte) + rendu graphique en live des paramètres du processus. Elle a été réalisée sur un temps de simulation $T_{sim} = 1200$ [s] et avec une température de démarrage égale à 15°C.

Sur la figure 15, nous observons un comportement du système selon les étapes suivantes :

1. 0s à 1200s : Mode manuel avec $MV = 35\%$
2. 600s : Variation sur SP (passe de 75°C à 65°C)
3. 900s : Variation sur DV (Passe de DV0 (50%) à DV0+5 (55%)

Interprétations graphiques :

- Pour l'étape 1 :
 - MV est constante car le système est en mode manuel.
 - PV réagit comme si le système était en boucle ouverte avec un échelon sur MV.
- Pour l'étape 2 :
 - MV ne varie pas car nous restons en mode Manuel, SP est donc ignoré.
 - PV reste constante.
- Pour l'étape 3 :
 - MV varie grâce à l'action feed-forward du système qui a pris en compte la variation de DV.
 - PV ne varie pas, ce qui prouve que le feed-forward a bien été implémenté et est fonctionnel pour le système réel.

6.3 Plateforme TCLAB avec PID sans FF en mode manuel puis automatique

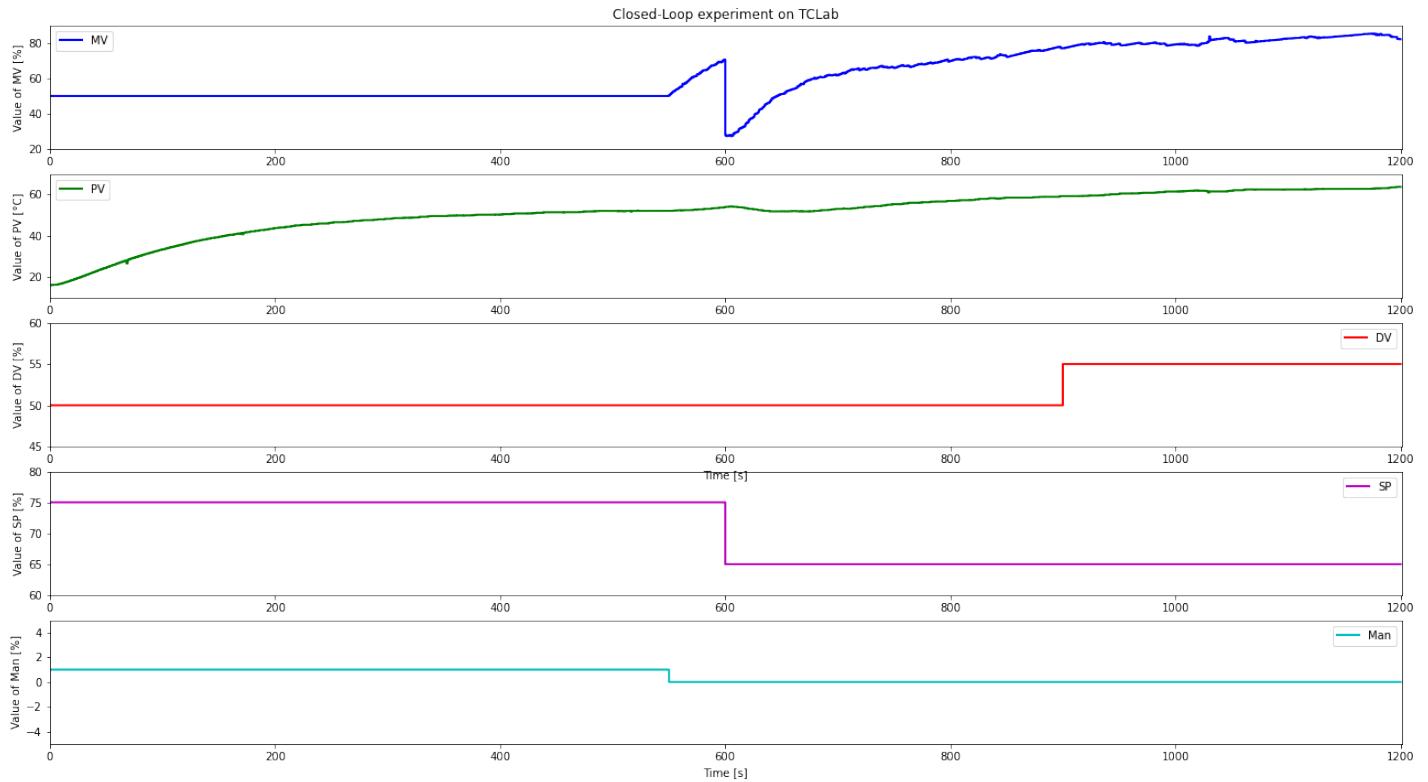


Figure 16 : Graphes des données du PID + No FF sur le TCLAB avec passage en mode automatique

Cette expérience nécessite une combinaison du PID sans feed-forward mais avec un passage mode automatique (boucle fermée) + rendu graphique en live des paramètres du processus. Elle a été réalisée sur un temps de simulation $T_{sim} = 1200$ [s] et avec une température de démarrage égale à 15°C.

Sur la figure 13, nous observons un comportement du système selon les étapes suivantes¹⁷ :

1. 0s à 550s : Mode manuel avec MV = 50%
2. 550s : Passage an mode automatique avec SP = 75°C
3. 600s : Variation sur SP (passe de 75°C à 65°C)
4. 900s : Variation sur DV (Passe de DV0 (50%) à DV0+5 (55%)

¹⁷ Le résultat de la simulation représentée par la figure 13 correspond au scénario 4 dans le fichier « Simulation_CLP_PID_FF.ipynb ». D'autres scénarios avec d'autres paramètres sont présents dans ce fichier afin de simuler sin système avec ou sans feed-forward, mode manuel, etc.

Interprétations graphiques :

- Pour l'étape 1 :
 - MV est constante car le système est en mode manuel.
 - PV réagit comme si le système était en boucle ouverte avec un échelon sur MV.
- Pour l'étape 2 :
 - MV varie maintenant grâce au PID pour amener PV à la consigne SP.
 - PV varie en fonction de la sortie du contrôleur PID pour atteindre SP (75°C).
- Pour l'étape 3 :
 - MV varie en fonction du nouveau SP (65°C) qu'on applique.
 - PV tend vers le nouveau SP de 65°C
- Pour l'étape 4 :
 - MV varie en fonction du contrôleur PID pour essayer de rectifier l'erreur.
 - PV continue d'augmenter car il n'y a pas d'action feed-forward pour anticiper l'effet de la variation de DV.

6.4 Plateforme TCLAB en mode manuel sans FF

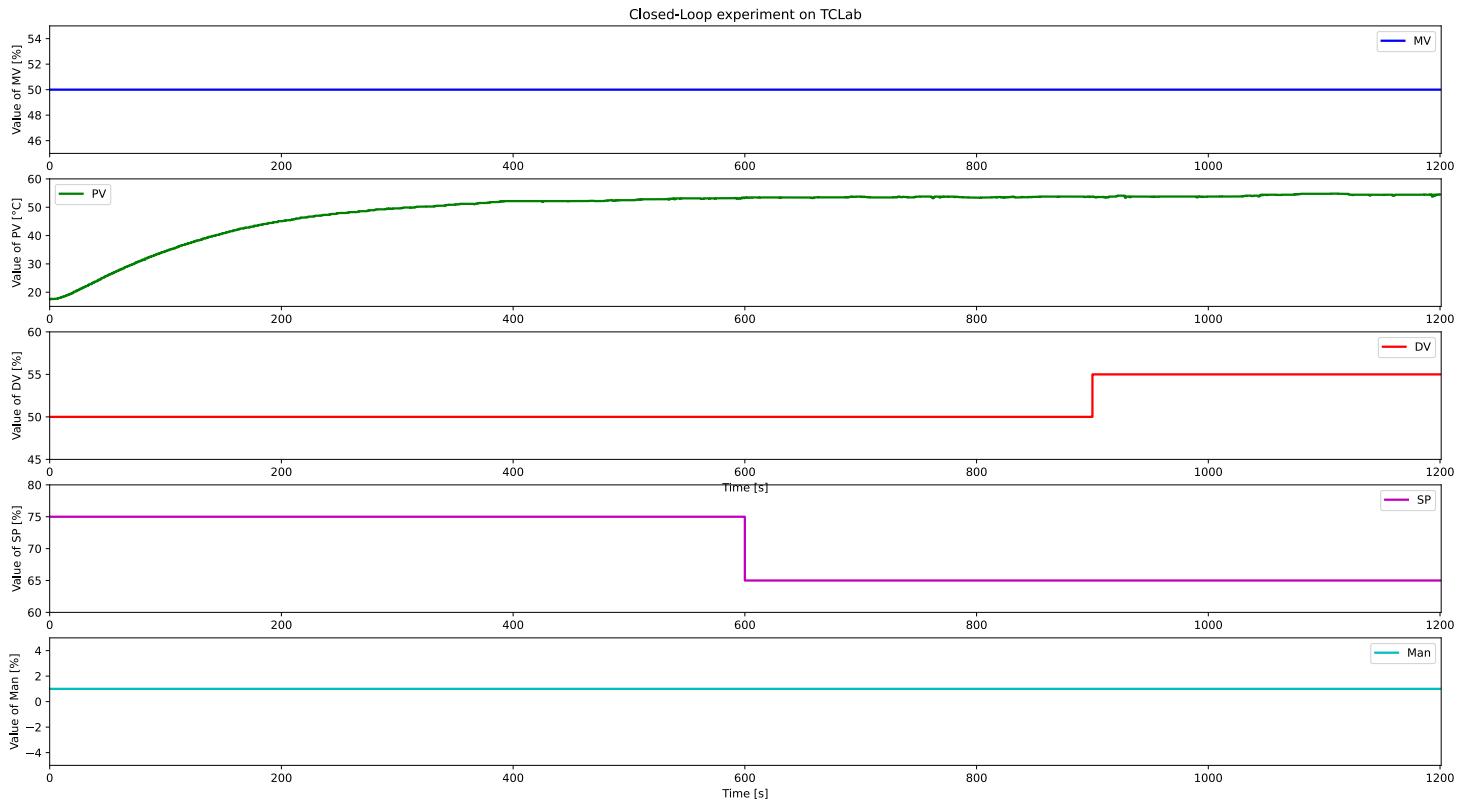


Figure 17 : Graphes des données du PID + FF sur le TCLAB avec passage en mode manuel

Cette expérience nécessite une combinaison du PID sans feed-forward mais avec un mode manuel (boucle ouverte) + rendu graphique en live des paramètres du processus. Elle a été réalisée sur un temps de simulation $T_{sim} = 1200$ [s] et avec une température de démarrage égale à 15°C.

Sur la figure 13, nous observons un comportement du système selon les étapes suivantes¹⁸ :

1. 0s à 550s : Mode manuel avec MV = 50%
2. 600s : Variation sur SP (passe de 75°C à 65°C)
3. 900s : Variation sur DV (Passe de DV0 (50%) à DV0+5 (55%)

Interprétations graphiques :

- Pour l'étape 1 :
 - MV est constante car le système est en mode manuel.
 - PV réagit comme si le système était en boucle ouverte avec un échelon sur MV.
- Pour l'étape 2 :
 - MV ne varie pas car nous restons en mode Manuel, SP est donc ignoré.
 - PV reste constante.
- Pour l'étape 3 :
 - MV reste constante car nous sommes toujours en mode manuel
 - PV augmente car il n'y a pas d'action feed-forward pour anticiper l'effet de la variation de DV.

¹⁸ Le résultat de la simulation représentée par la figure 13 correspond au scénario 4 dans le fichier « Simulation_CLP_PID_FF.ipynb ». D'autres scénarios avec d'autres paramètres sont présents dans ce fichier afin de simuler un système avec ou sans feed-forward, mode manuel, etc.

7 Robustesse de la boucle¹⁹

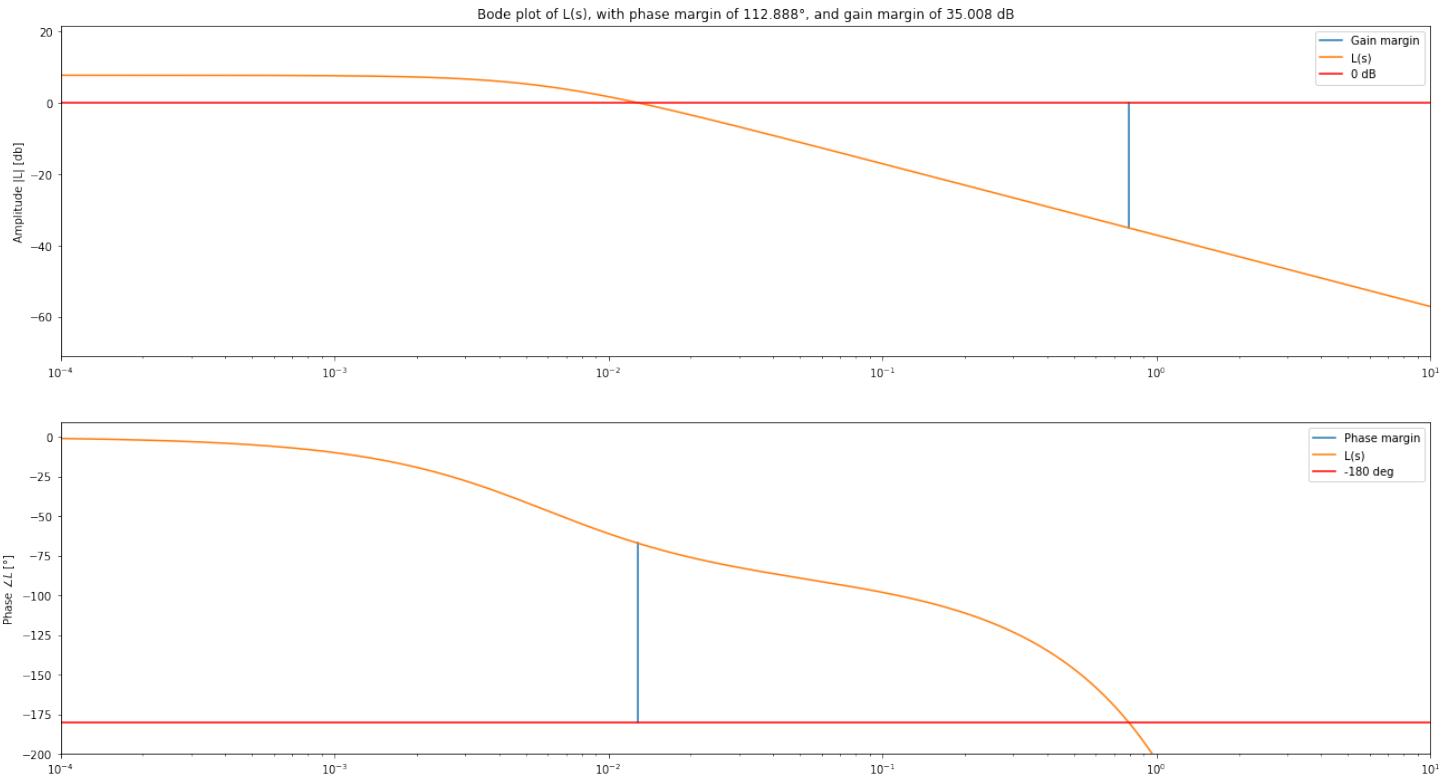


Figure 18 : Diagramme de Bode avec marges de gain et de phase de la boucle

En fonction de tous les paramètres du système représenté dans la figure 7 (section 4.1.3), on peut calculer la réponse fréquentielle de la boucle, et en tracer le diagramme de Bode. En suite, nous pouvons déterminer la marge de gain et la marge de phase, qui dans la figure 18 valent 35.008dB et 112.888° respectivement.

On peut déduire de cela que nous pourrions augmenter encore le gain de la boucle de manière non négligeable²⁰ avant que cela pose des problèmes d'instabilité, mais par manque de temps nous n'avons pas pu réaliser ce test.

¹⁹ Le diagramme de Bode (figure 18) a été obtenu à partir du fichier « package_LAB_Margins.ipynb »

²⁰ Le slide 70/224 du cours « Control Theory & Applications » nous donne comme valeurs indicatives « 5 à 14dB » comme marge de gain et « 30° à 60° » comme marge de phase pour considérer qu'une boucle est robuste aux changements.

8 Problèmes rencontrés

8.1 Problème de détermination du modèle à utiliser pour implémenter l'IMC tuning.

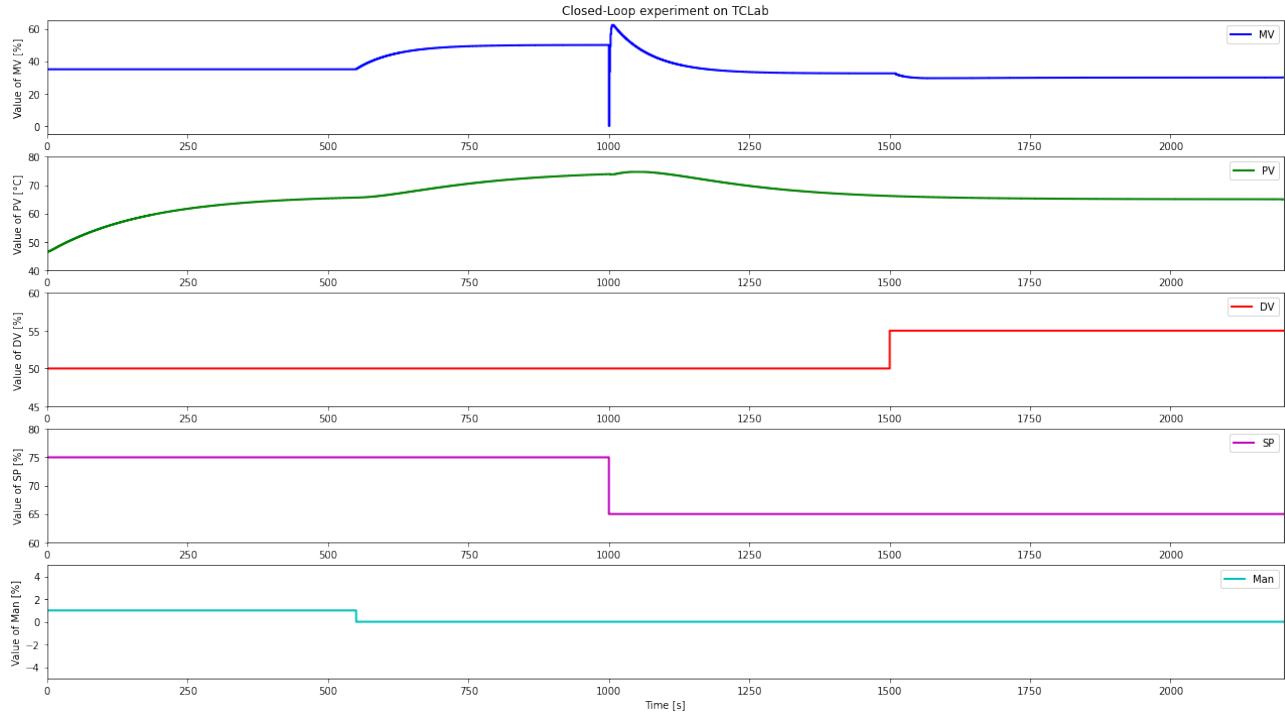


Figure 19 : Diagramme du PID + FF en auto mode avec IMC tuning basé sur un second ordre

En effet après essai il a été remarqué que lorsque nous prenions un modèle SOPDT nous rencontrions des problèmes en simulation. En sortie du PID le signal MV tendait vers des valeurs qui ne correspondaient pas à ce qui était attendu, et donc le signal PV commençait d'abord par augmenter avant de tendre vers la valeur de consigne. Sur la figure 19, il est facilement observable que tout d'abord MV réagit de manière non-intuitive à la variation de la consigne.

8.2 Problème d'identification des paramètres dynamiques de $D(s)$ (DV)

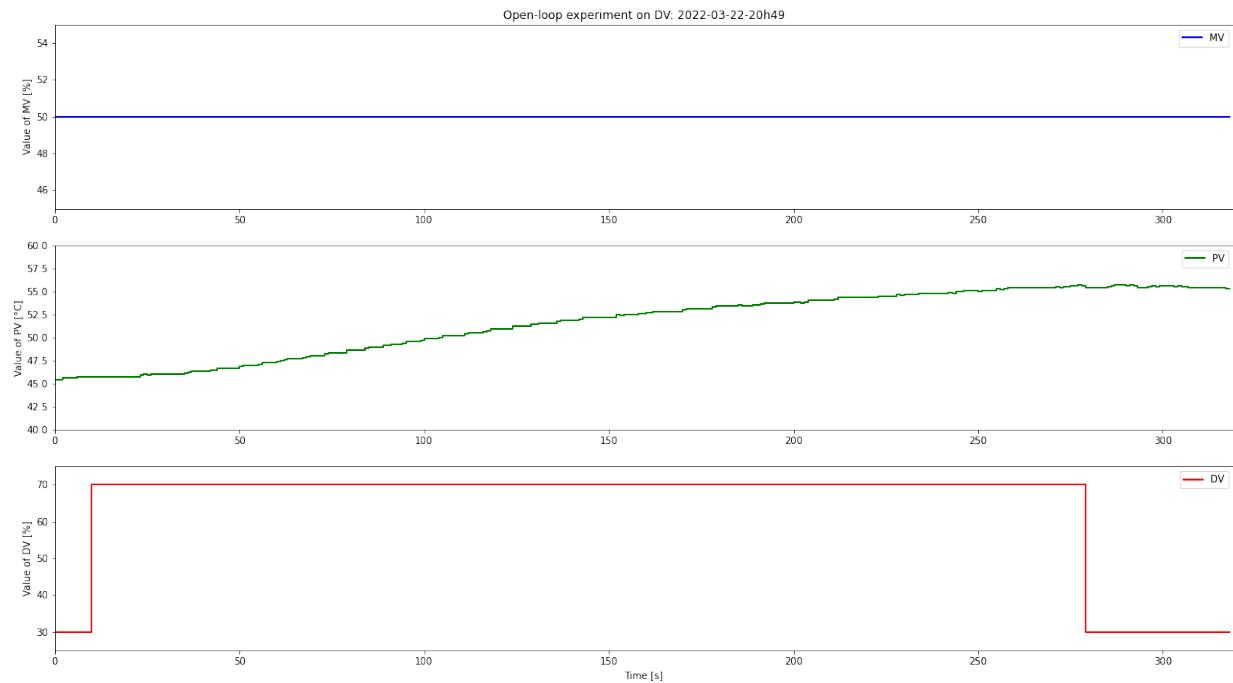


Figure 20 : Réponse indicielle de DV

Nous pouvons constater que malgré l'abaissement (échelon vers le bas) du signal DV le programme n'attendait pas assez longtemps pour percevoir la diminution de PV. En effet, lorsque PV atteignait une valeur stable l'expérience s'arrêtait.

8.3 Problèmes de bruit de mesure & de stabilisation de PV

Au fur et à mesure de nos itérations, il a été constaté que le capteur de température T1 était très sensible au perturbations extérieures. Il est également possible que ce capteur T1 ait un niveau de bruit interne non négligeable, ou que le bruit provienne du convertisseur analogique-numérique (ADC) du microcontrôleur utilisé.

9 Conclusion

Au vu des expériences menées et des conclusions établies, nous pouvons constater que l'objectif principal du laboratoire à savoir la compréhension, l'élaboration, l'implémentation et le déploiement d'un régulateur PID avec feed-forward a été rempli correctement.

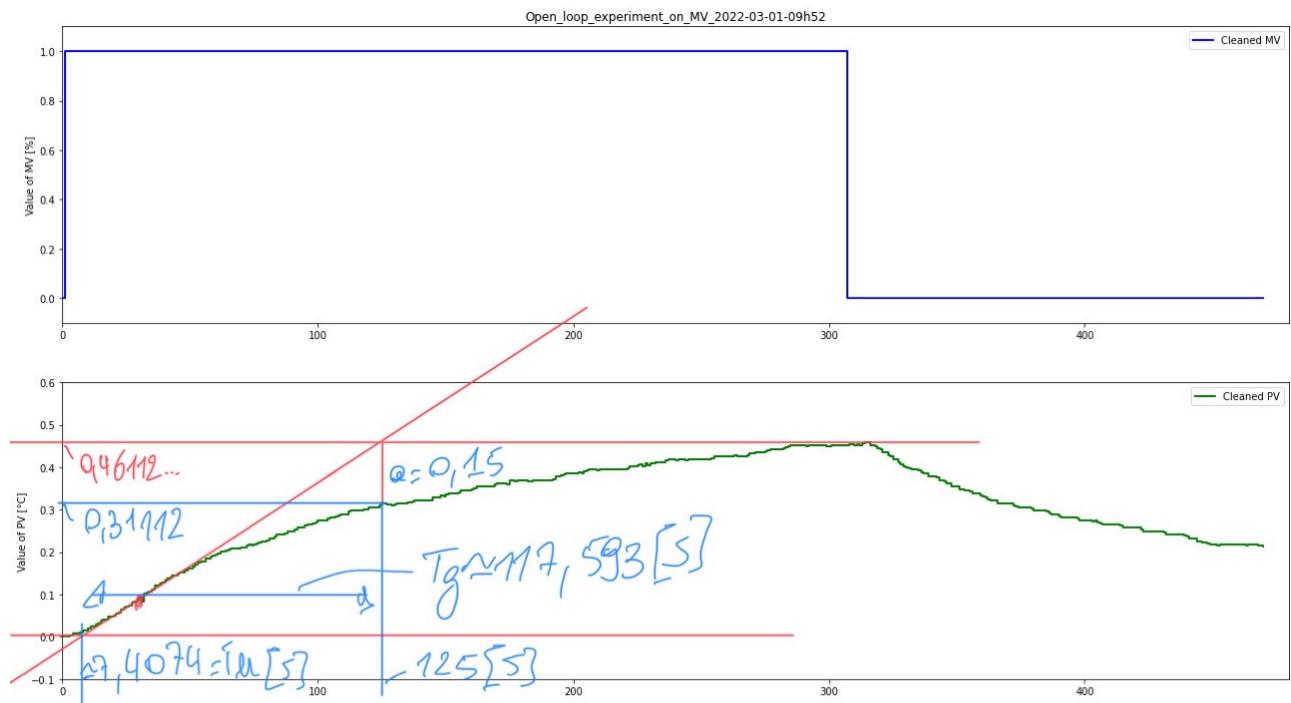
Nous pourrions néanmoins avec un peu plus de temps des améliorations à notre système. En effet comme mentionnés au point 8, plusieurs problèmes ont été rencontrés lors de ce laboratoire. Il aurait été plus précis de pouvoir bénéficier d'un IMC tuning utilisant un modèle SOPDT se rapprochant ainsi plus du modèle réel.

Ajoutons à cela que malgré nos tentatives pour isoler thermiquement le TCLAB des perturbations externes non désirées, il était relativement complexe de pouvoir obtenir un signal PV ne dépendant que des paramètres mentionnés aux points 1 et 2.

Sources

1. Support de cours « Control Theory & Applications », Franky De Bruyne
2. Consignes de laboratoire « Assignment 3beau3C 4meoau4C 4minau4C 2021 2022 », Franky De Bruyne
3. Matplotlib reference : <https://matplotlib.org/stable/api/index.html>
4. Numpy reference <https://numpy.org/doc/stable/reference/>
5. « Control of a First-Order Process with Dead Time », K. Craig : http://engineering.nyu.edu/mechatronics/Control_Lab/Craig/Craig_RPI/2002/Week2/First-Order_Process_Time_Delay_2002.pdf
6. « Dynamics and Control; First Order Plus Dead Time (FOPDT) », John D. Hedengren, <https://apmonitor.com/pdc/index.php/Main/FirstOrderSystems>
7. PID pour différents paramètres Kp Ki et Kd : https://en.wikipedia.org/wiki/PID_controller#Mathematical_form
- 8.

Annexes



Annexe 1 : Détermination graphique des paramètres