RAPPORT TCLAB

DREWNOWSKI B.
KRASOWSKI D.

Table des matières

Intro

Ce document a pour but de documenter nos recherches, résultats et réflexions autour du laboratoire de Control Theory, qui vise à mettre en pratique les notions vues au cours théorique. Cela comprend l'identification de processus dynamiques à travers diverse méthodes, la mise en place d'un PID, l'optimisation de ses paramètres grâce à IMC tuning, l'implémentation d'un système de Feedforward, la notion de marges de gain et de phase, l'influence des paramètres tels que alpha, gamma…

Identification processus et perturbations dynamiques

Avant de pouvoir se lancer dans la régulation d'un process, il est impératif d'apprendre davantage sur ce dernier, nous ferons cela en utilisant la plateforme TCLab que nous ferons fonctionner en boucle ouverte avec des conditions fixées. Nous obtiendrions ainsi une courbe qui représente la sortie PV, pour processed value.

Le TCLab dispose de deux sources chauffantes HP1 et HP2, ainsi que deux sondes de température, T1 et T2. Nous sommes face à un système multivariables. Cependant, nous considérons que c'est un processus monovariable, avec HP1 qui sera MV, T1 comme PV et HP2 sera le DV.

De plus, notre système est de type non linéaire, pour cette raison, nous allons travailler autour d'un point de fonctionnement qui est le suivant :

Variables	Valeur
MV0	50%
DV0	50%

Ainsi, nous obtiendrons la réponse indicielle du processus et des perturbations.

Variation DV

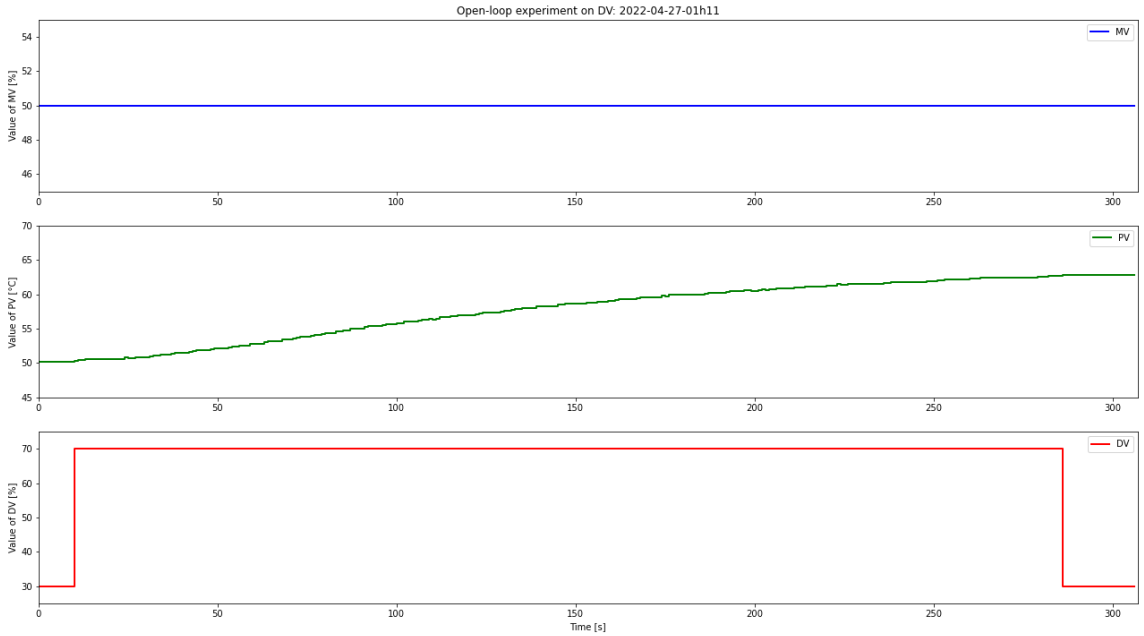


Figure : Changement de DV en boucle ouverte

Ici nous constatons que nous avons appliqué une variation sur DV pour constater la réaction du système en boucle ouverte, cette réponse indicielle nous permettra de modéliser les perturbations D(s) avec la méthode python que nous expliquerons plus tard.

Variation MV

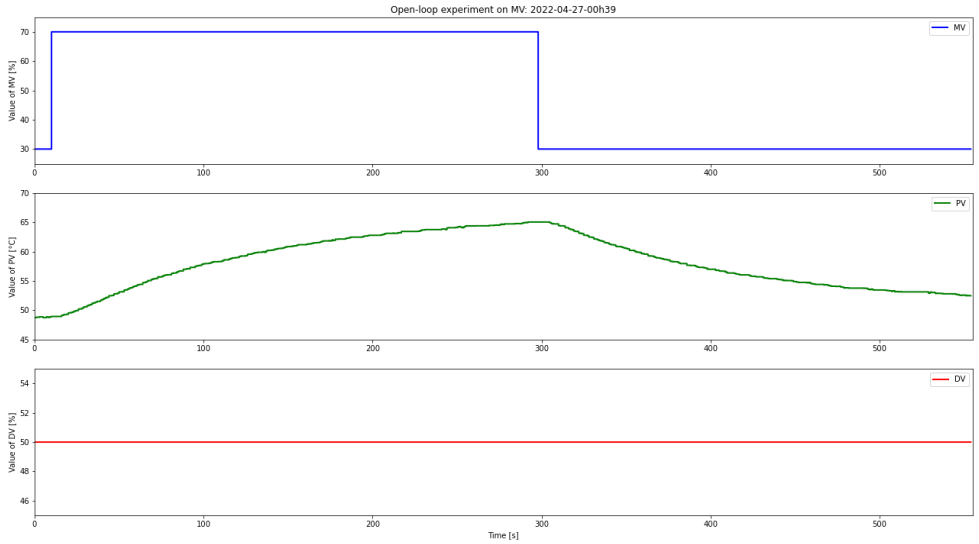


Figure : Changement de MV en boucle ouverte

Dans le cas présent, nous avons la réponse indicielle qui permet de trouver P(s).

Méthode programmation python

La méthode suivante utilise un code python qui se base sur les minimums et maxima locaux, en faisant une série d'itérations, on obtient une courbe qui s'approche du processus réel. La fonction nous renvoie les paramètres de gain K, les constantes de temps T1, T2 ainsi que le déphasage Thêta pour une fonction du second ordre. Pour une fonction du premier ordre, nous obtenons le gain K, la constante de temps T et le déphasage Thêta.

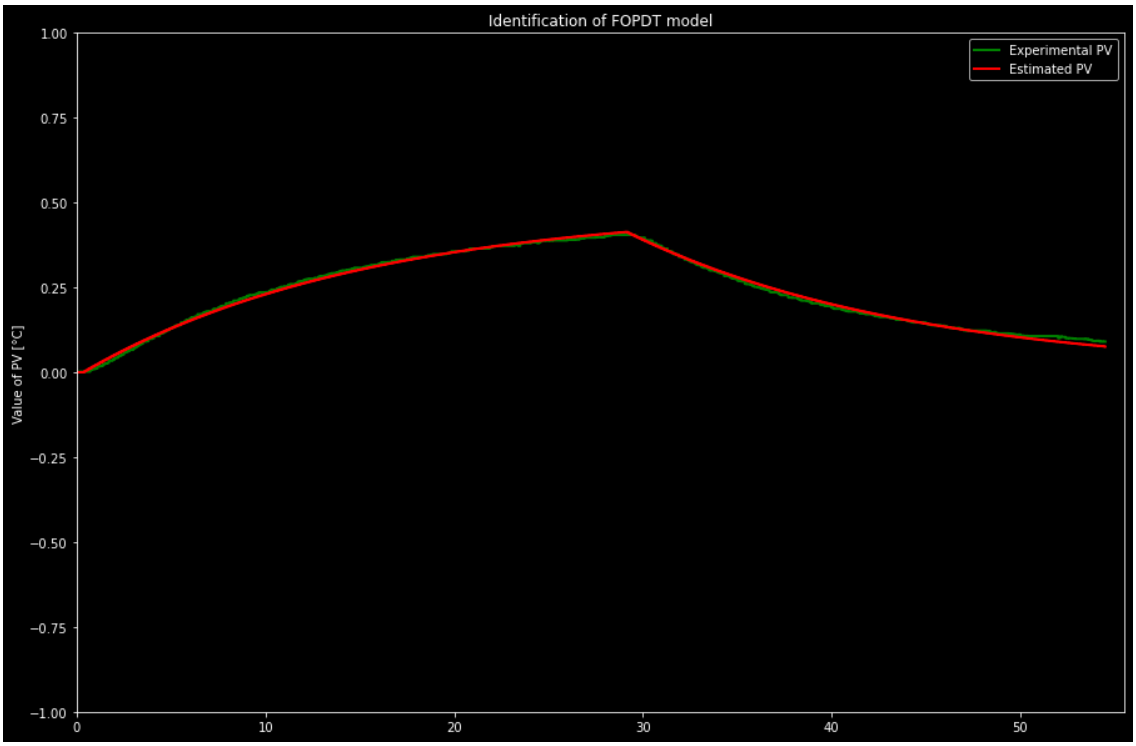


Figure: Résultat de l'identification premier ordre

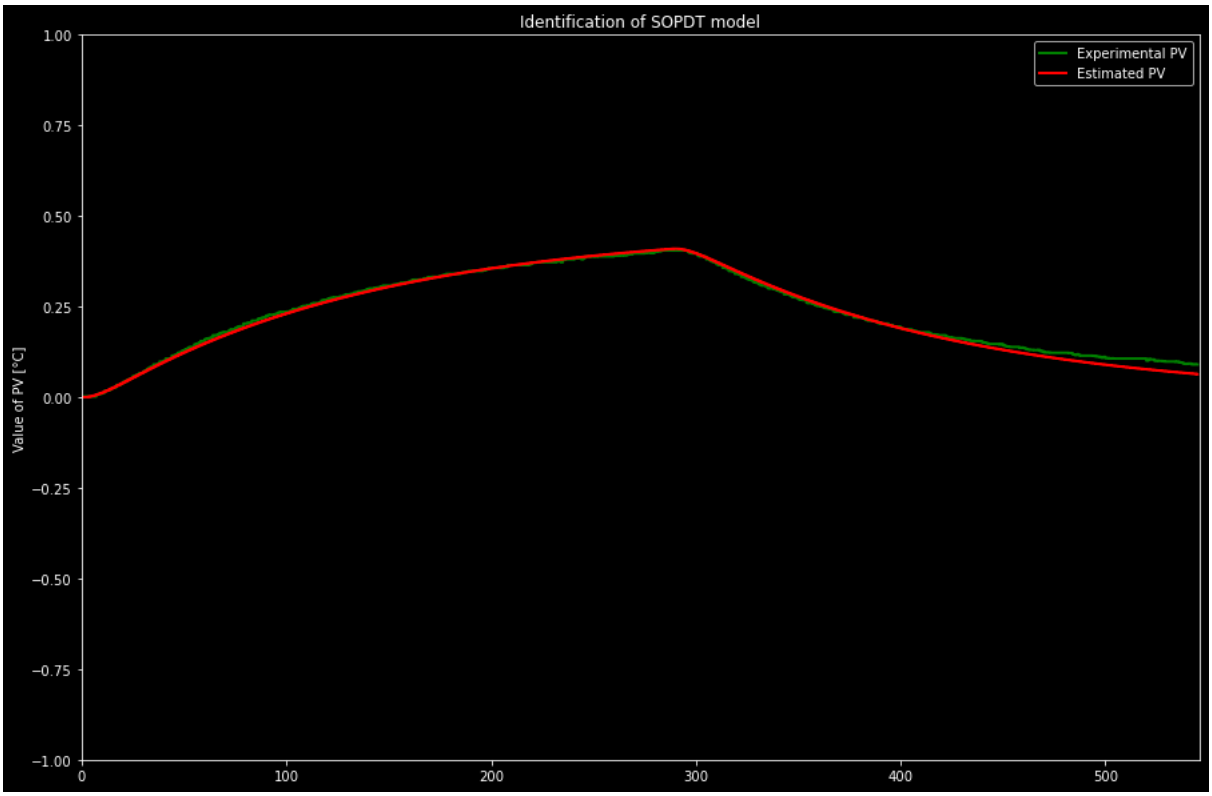


Figure: Résultat de l'identification du second ordre

Nous pouvons constater que les deux résultats sont proches, nous allons les comparer avec les méthodes d'identification graphiques.

Paramètres obtenus par méthode graphique

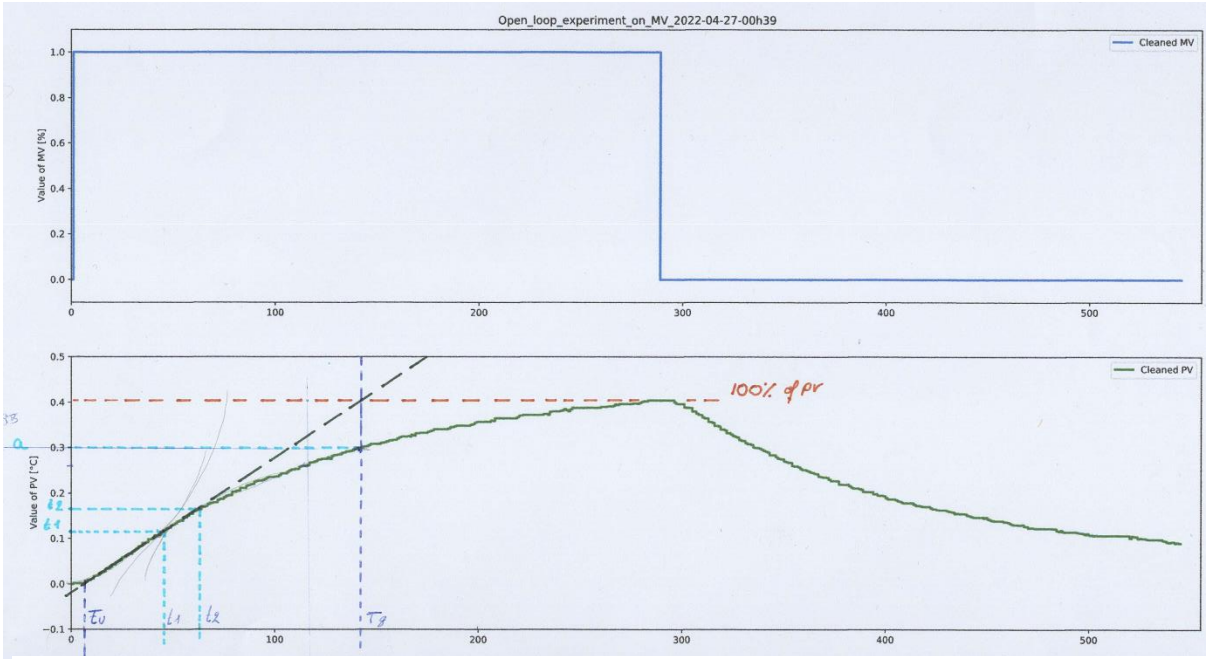


Figure: Outil d'obtention paramètres graphiques

Variables	Mesures
t1	45.58
t2	63.3
Tu	6.33
Tg	136.73
a	0.123
Kp	0.4

Les résultats ci-dessous vont nous permettre de trouver les valeurs de gain, constantes de temps et déphase via les méthodes de Broida, Van de Grinten et Strejc. Ensuite nous obtiendrons les fonctions de transfert pour notre processus TCLab.

Broida

Pour calculer le processus du premier ordre avec le temps mort (FOPDT), on va utiliser l'équation de Broida.

La fonction de transfert du modèle de Broida: $P_B(s) = \frac{K_P\, e^{-\theta s}}{(T\, s+1)}$

Afin de trouver les paramètres du processus : constante de temps T et le délai du processus θ en utilise deux méthodes Broida simple et Broida compliqué.

Par la méthode simple les paramètres sont égaux au temps de monté Tg et le temps de démarrage.

$$T = T_g \;\&\; \theta = T_u$$

Pour cette première méthode on obtient : $T = 136.73 \;\&\; \theta = 6.33$

Par la méthode plus compliquée en utilise l'équation : $T = 5{,}5\,(t_2 - t_1) \;\&\; \theta = 2{,}8t_1 - 1{,}8t_2$

Pour cette deuxième méthode en obtient : $T = 97.43 \;\&\; \theta = 13.69$

Van der Grinten

Il est important mentionner que la méthode de Van der Grinten n'est pas adaptée à notre processus, car nous obtenons une constante de temps négative ce qui est impossible. L'approximation du processus pour le deuxième ordre avec temps mort (SOPDT) ce fait avec la méthode de van der Grinten dans la fonction de transfert est :

$$P_{vdG}(s) = \frac{K_P}{(T_1 s + 1)(T_2 s + 1)} e^{-\theta s}$$

Pour calculer les paramètres du modèle T_1, T_2 et θ on utilise les équations suivantes :

Équations	Résultats
$T_1 = T_g \frac{3\,a\,e - 1}{1 + a\,e}$	-15.289
$T_2 = T_g \frac{1 - a\,e}{1 + a\,e}$	81.91
$\theta = T_u - \frac{T_1\,T_2}{T_1 + 3\,T_2}$	13.89

Pour le résultat du processus par la méthode vdG nous trouvons :

$$P_{vdG}(s) = \frac{0.4}{(-15.289s + 1)(81.91s + 1)} e^{-13.89s}$$

Strejc

Le model de Strejc nous permet d'approximer le processus pour un nième ordre

Fonction de transfer : $P_s(s) = \frac{Kp\, e^{-\theta s}}{(Ts+1)^n}$

Premièrement on détermine le rapport de Tu sur Tg : $\frac{Tu}{Tg} = 0.046$

La valeur obtenue permet de déterminer l'ordre du système qui veut n = 1.

Ensuite, on calcule le paramètre T : $T = \frac{Tg}{b_n} = 136.73$

Une fois T calculé en calcule Tu_{th}qui est le temps mort apparent résultant du système nième ordre.

$$a_n = \frac{Tu_{th}}{Tg} \Rightarrow Tu_{th} = a_n Tg = 0$$

Avec Tu_{th}on peut obtenir le délai θ : $\theta = Tu - Tu_{th} = 6.33$

Finalement, notre fonction de transfert via le modèle de Strejc veut : $P_s(s) = \frac{0.4\, e^{-6.33s}}{(136.73s+1)^n}$

Comparaison des graphes

Il est important d'avoir un esprit critique et observer les résultats obtenus à l'aide d'un graphe. Nous avons superposé les résultats obtenus graphiquement ainsi qu'avec le script python. Les méthodes graphiques sont utiles et faciles à appliquer certes, mais il faut mentionner que ces dernières sont moins précises et le résultat obtenu est moins fidèles comparé aux scripts python utilisés.

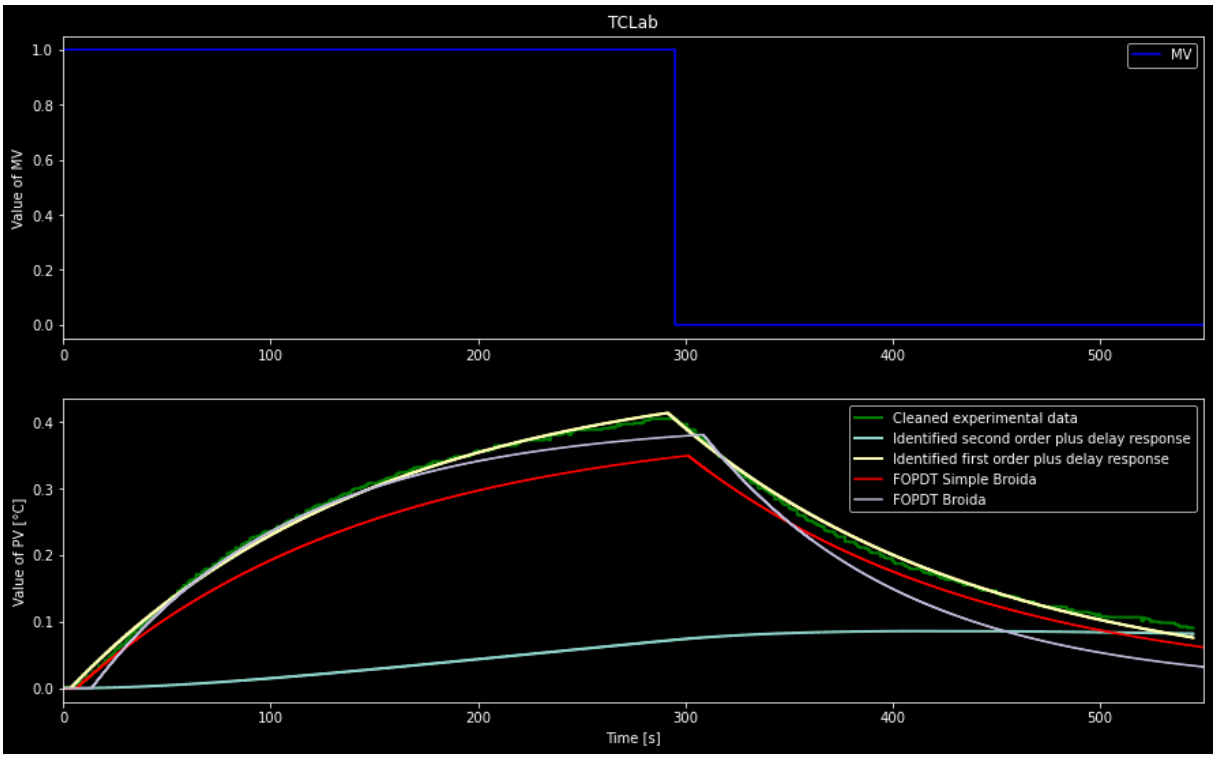


Figure : comparaison des modèles

Le graphe ci-dessus nous monter en vert le processus dans la vie réelle comparé aux résultats obtenus.

Diagramme de Bode

Dans le diagramme ci-dessous, nous obtenons les gains et la phase de chaque modèle, pour les comparer entre eux. On constate que les allures du modèle de premier ordre, Broida et Strejc sont similaires. Le diagramme pour le modèle de second ordre nous semble le plus en adéquation avec ce que nous avons vu au cours.

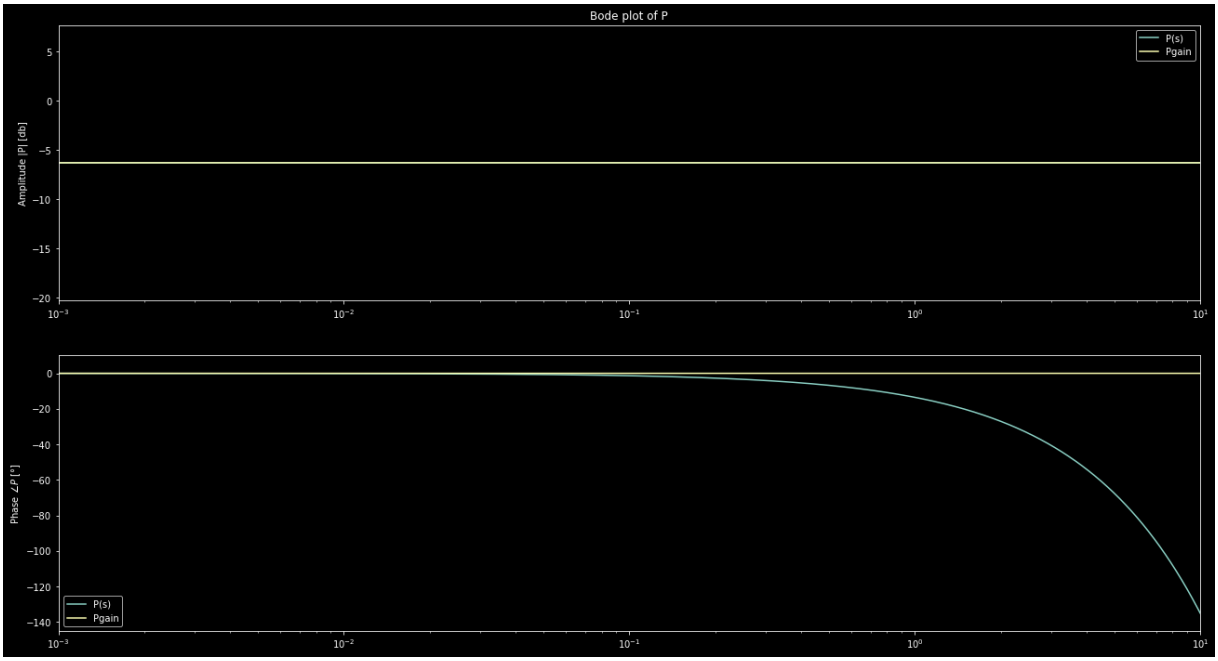


Figure diag Bode modèle premier ordre

PID

Implémentation du IMC Tuning

Pour que notre PID puisse être le plus efficace possible, nous devons trouver les paramètres Kc, Ti, Td optimaux. La méthode de l'IMC tuning propose des formules pour y parvenir. Il faut noter que nous avons également un paramètre nommé gamma qui permet de tuner notre PID. Nous expliquerons son effet plus tard.

Variable	Valeur
Kc	2.523753764240984
Ti	139.6892400432098
Td	7.229030808202924

Analyse du gamma

Nous avons vu que gamma influence l'action proportionnelle. Le fait de diminuer la valeur de γ rends le PID plus agressif au début. Cela signifie un kick du contrôleur qui est plus grand et un temps de montée réduit. Nous avons gardé un γ égal à 1 et nous n'avons pas eu d'oscillations lors de nos tests.

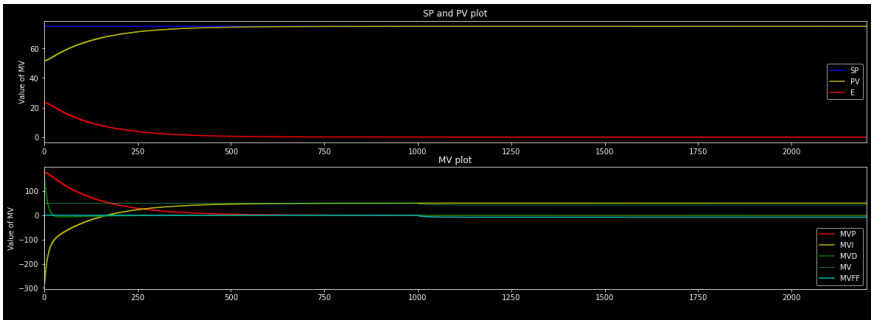


Figure gamma = 0.3

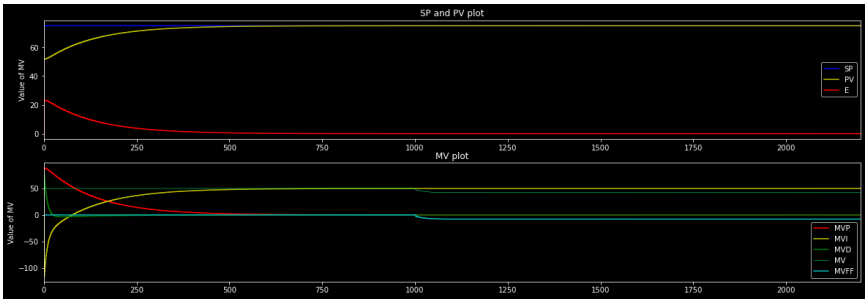


Figure gamma = 0.6

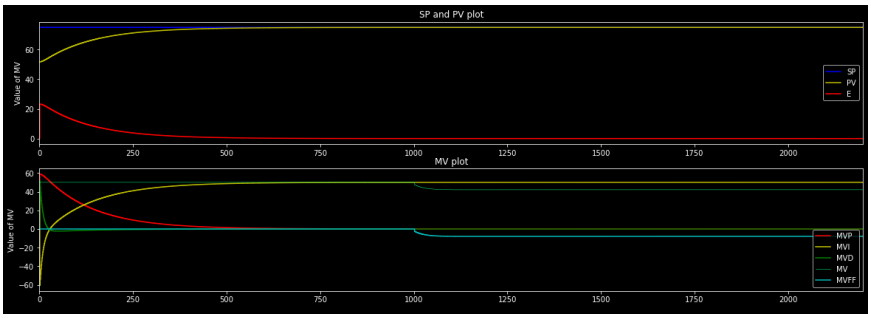


Figure 1gamma = 0.9

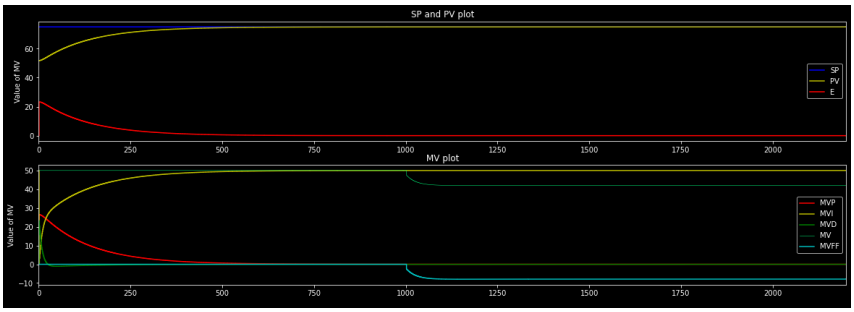


Figure 2 gamma = 2

Marges de gain et phase

Les marges de phase et de gain correspondent à la distance depuis les points auxquels l'instabilité se produise. Dans le but d'avoir une meilleur stabilité les distances de ces deux marges devant être le plus élevé que possible.

La marge de gain est mesurée à la fréquence où le décalage de phase est égal à 180°

$$Am = \frac{1}{|L(j\omega_u)|}$$

La marge de phase est mesurée à la fréquence où le gain est égal à 0dB

$$\emptyset m = Arg\big(C(j\omega_c)\big)$$

Essai 1 : Avec un alpha de 0.1 et un gamma de 0.9, nous obtenant 56.53dB pour la marge du gain et 91.42° à 0.01rad/s pour la marge de phase

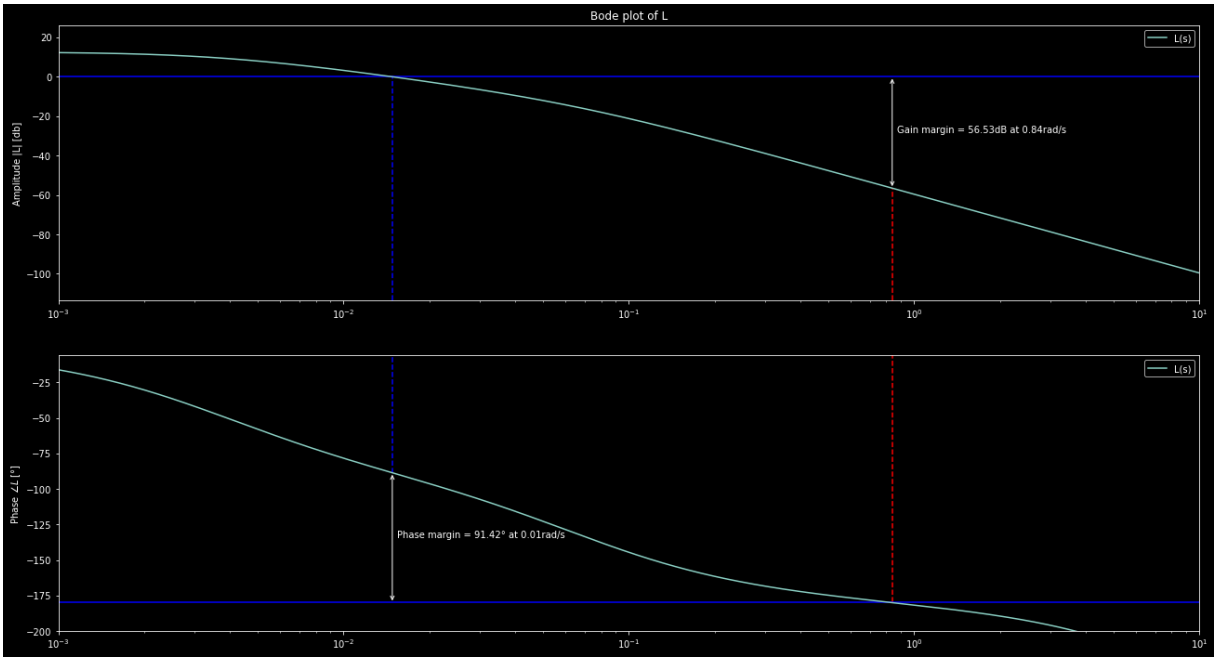
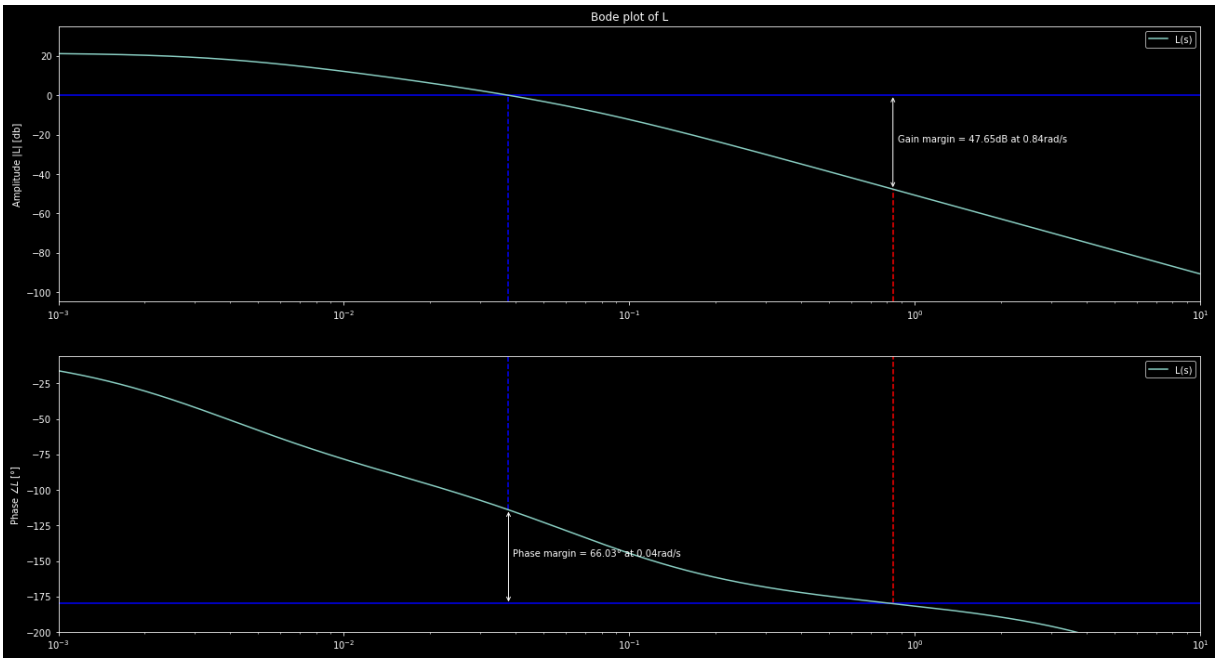


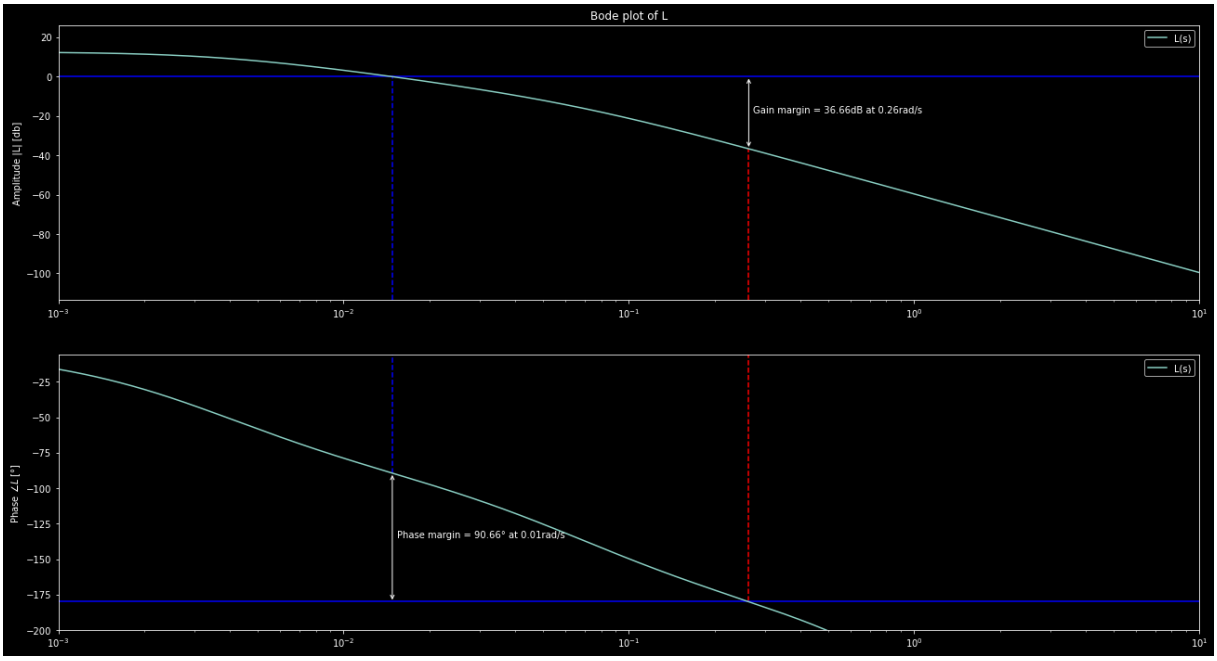
Figure marges de gain & phase, α=0.1, γ=0.9

Essai 2 : Avec un alpha de 0.1 et un gamma de 0.2, nous obtenant 46.65dB à 0.84rad/s pour la marge du gain et 66.03° à 0.04rad/s pour la marge de phase



En comparent l'essai 1 et 2, nous peuvent déduire que les marges de gain et de phase se sont détériorées quand on diminue le gamma.

Essai 3 : Avec un alpha de 1 et un gamma de 0.9, nous obtenant 36.66dB à 0.26rad/s pour la marge du gain et 90.66° à 0.01rad/s pour la marge de phase



À la suite de cet essai, on constate que changer la valeur d'alpha n'a pas d'incidence sur la marge de phase, mais diminue seulement la marge de gain.

Il faut prendre un gamma plus petit possible pour éviter d'avoir un système instable et un alpha petit également. Néanmoins, un gamme petit conduit à un temps de montée plus conséquent pour atteindre SP. Dès lors, il faut trouver un bon compromis entre rapidité et stabilité.

Feedforward

Le Feedforward permet d'anticiper les perturbations DV qui se rajoutent après le processus pour les compenser. Pour que la compensation soit possible le θs doit être plus grand ou égale à θp. Dans le cas contraire les perturbations vont arriver plus rapidement que la compensation.

La fonction de transfert du FeedForward est : $-\frac{D(s)}{P(s)}$

$$\frac{D(s)}{P(s)} = \frac{Kd}{Kp} \cdot \frac{T_1 \cdot s + 1}{T_{1d} \cdot s + 1} \cdot \frac{T_2 \cdot s + 1}{T_{2d} \cdot s + 1} \cdot e^{-(\theta d - \theta) \cdot s}$$

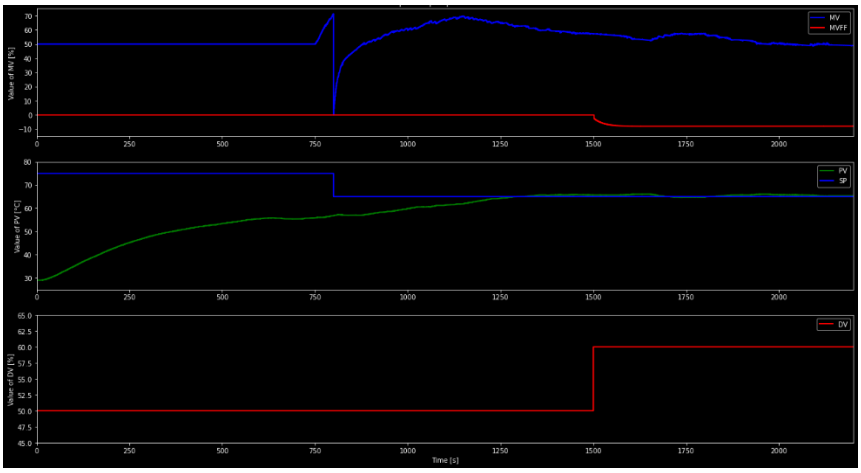
Dans l'analyse en peut distinguer de cette équation deux Lead Lag et un délai. Cette implémentation de deux LL et du délai est fait dans le code dans la boucle fermé.

Expérimentation sur le TCLab

Une fois que nous avons mis en place le PID ainsi que le Feedforward, il est temps de tester sur le TCLab. Il faut savoir que c'est un processus qui a une constante de temps élevée, il faut être patient pour réaliser ces tests.

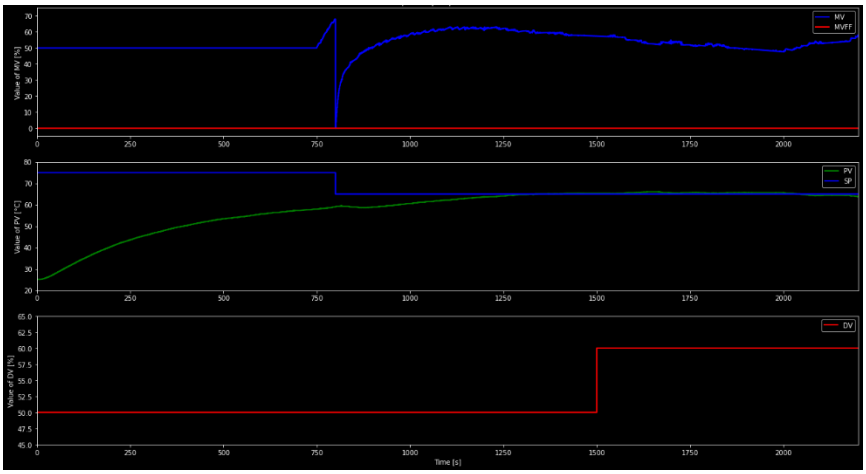
Mode auto avec Feedforward

Dans ce mode, nous pouvons voir que le changement de consigne SP induit une anticipation de l'action dérivée aux alentours de t = 750. On constate un léger dépassement de la consigne qui sera comblé par l'action intégrale, qui agit tant que l'erreur est non-nulle. Un changement sur DV géré par le Feedforward qui s'additionne au MV en sortie du régulateur. Une diminution de MVFF induit une diminution de MV également pour que PV reste inchangé.



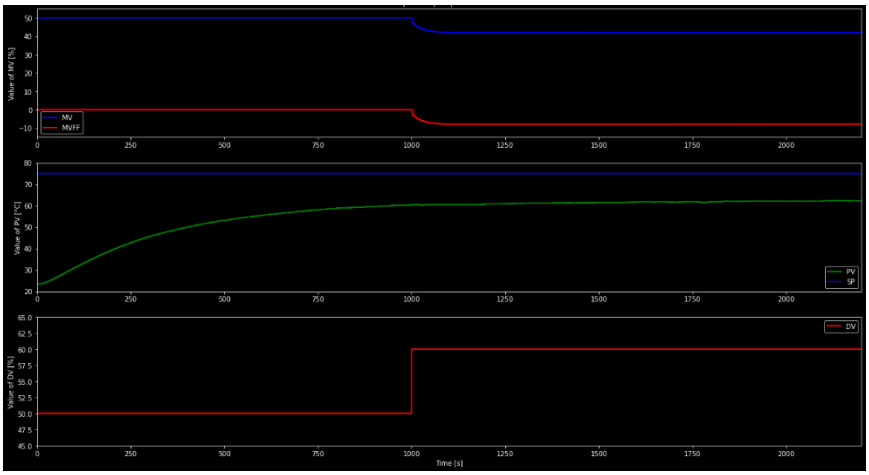
Mode auto sans Feedforward

Dans le second cas, nous avons le PID qui agit sur le système, mais sans l'aide du Feedforward. Les mêles constatations sont faites pour le léger dépassement de consigne. Pour le changement de DV, nous pouvons voir que MV en sortie du régulateur arrive à absorber cette fluctuation et éviter une variation de PV, ici c'est sur la courbe bleue de MV qu'on voit la régulation de faire, or sur le cas précédent, le PID n'était pas au courant de la perturbation DV qui fut gérée par le Feedforward.



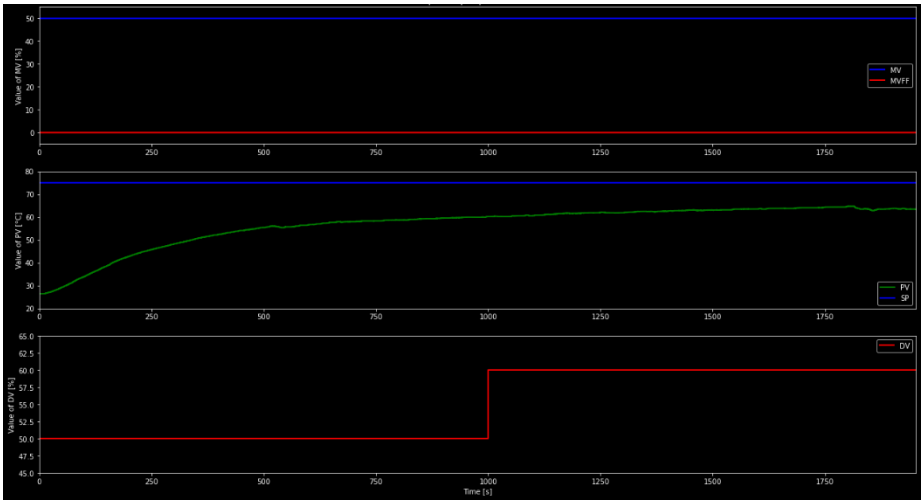
Mode manuel avec Feedforward

Dans ce troisième cas, nous sommes en mode auto avec Feedforward, cela veut dire que nous avons fixé une valeur MV manuel et nous avons le Feedforward qui permet de contrer les perturbations DV. MV est la somme de MVFF fourni par le Feedforward ainsi que MV manuel. Avec une valeur de MV fixe, nous ne parvenons pas à la consigne mais la variation de DV n'exerce aucune influence sur PV ce qui est une bonne chose.



Mode manuel sans Feedforward

Dans ce dernier mode, nous n'avons ni PID, ni Feedforward qui est d'application. La valeur de MV ne changera en aucun cas. Nous constatons que nous ne pouvons pas atteindre la consigne et que le changement en DV perturbe PV. PV continue de monter là ou dans le mode précédent PV ne montait pas par suite du delta de DV.



Conclusion

En conclusion nous avons atteint tous les objectifs de ces quatre labos. Nous avons pu réaliser l'expérimentation de quatre situations avec de résultat aux théoriquement on s'attendais.

Premièrement, il était indispensable de bien réaliser le PID afin de simuler les diffèrent scénarios de contrôle. Nous avons pu s'intéressé au FeedForward et à son utilité pour compenser les perturbations dans un système stable. L'action proportionnelle permet de jouer sur la vitesse, la dérivée d'anticiper et l'intégrale à supprimer complétement l'erreur.

Finalement, ce labo nous a permis la compréhension du cours théorique sur l'implémentation d'un PID et l'action du féedforward. Nous avons compris de manière plus approfondie l'effet de alpha et gamme sur notre processus.
Alpha a une incidence sur l'action dérivée, il permet de filtrer le gain en haute fréquence avec un rapport de 1/alpha car le bruit peut rendre instable notre contrôleur.
L'action de gamma joue comme dit précédemment sur l'action proportionnelle, donc sur le temps de montée par conséquent.
Ces deux valeurs sont à choisir avec précaution pour éviter une instabilité du système.

Nous avons rencontre quelques soucis avec les plots ainsi que le package_DBR qui agissait comme si nos fonctions n'existaient pas. Nous souhaiterions mentionner que la répartition de laboratoires nous a compliqué un peu l'organisation avec d'autres UE et a réduit le temps de travail entre les séances, cependant nous arrivons à un résultat concluant de notre point de vue.