Rapport de Laboratoire - Control Theory

Nicolas Juruet (21211) - Xavier Allaud(21151)

March 2024

1 Introduction

Le but de ce rapport est d'expliquer le processus de conception d'un régulateur PID avec une boucle de rétroaction en utilisant un kit TCLab. Pour se faire, il faudra identier la dyamique d'entrée/sortie du système afin de construire un modèle de celui-ci. Sur base de ce modèle, il sera possible d'implémenter un PID discret qu'il faudra ensuite optimiser puis tester en conditions réelles.

Table des matières

1 Introduction **1**

2 Identification de la Dynamique du Processus **2**

2.1 Graphique 2

2 Identification de la Dynamique du Processus

2.1 Graphique

Application d'un step $X(s) = \frac{1}{s}$ lorsque le système se situe en régime établi pour analyser la "step response" $Y(s) = \frac{P(s)}{s}$iiiii.

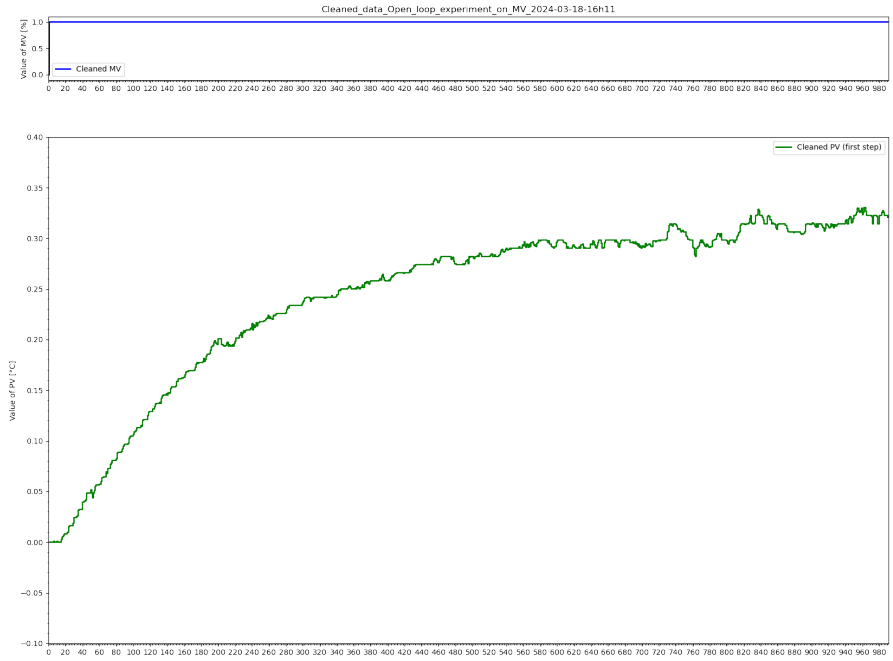

FIGURE 1 – Graphique step response

1. Le graphique (1) obtenu permet (presque) d'affirmer que le processus est un système du 1^e ordre avec un délai. L'allure de la "step response" est donc $y(t) = K_p(1 - e^{-t/T})u(t)$.
2. Le but est maintenant d'obtenir les paramètres de la fonction de transfert du processus $P(s)$ ce qui permettra de prédire $Y(s)$ pour n'importe quel $X(s)$ puisque $Y(s) = P(s) \cdot X(s)$ ou $PV = P(s) \cdot MV$. Pour se faire, on utilise une fonction de minimisation d'erreur :

```
minimize(FOPDT_cost,
    p0,args=(MVm,PVm,Ts,
    (fig,ax1,l1,l2)),
    method='Powell',
    bounds=bnds,
    options={'maxiter': maxIter})
```

Où `p0` contient les points de départ d'optimisation pour les paramètres `Kp, T` et `theta`. La fonction de minimisation va faire varier ces paramètres afin que la fonction de coût, `FOPDT_cost`, renvoie la valeur la plus basse possible.
`FOPDT_cost` :

```
for i in range(0,len(MV)):
    t.append(i*Ts) #calcule le temps correspondant au point actuel
    ↪  de la donnée en se basant sur la période de sample
```

```
    MVTemp.append(MV[i]) #itère sur la longueur du vecteur d'entrée

    Delay_RT(MVTemp,theta,Ts,MVDelay) #application d'un délai à MV
    FO_RT(MVDelay,Kp,T,Ts,PVSim) #On applique un délai au premier ordre via le
    ↪  délai appliqué à MV puis on calcule un point du vecteur de sortie à
    ↪  chaque itération
    objective = objective + (PV[i] - PVSim[i])**2 #on ajoute le carré de
    ↪  l'erreur actuel à la somme des carré des erreurs précédentes
```