Assignment 6 - Project Kårstø Statpipe Butane Splitter

TTK4210 - Advanced Control of Industrial Processes

Sondre Myrberg

April 9, 2018

Contents

List of Figures

1 Identification and tuning of the controllers

In this chapter we will look at the tuning of the different controllers of the system. Firstly we will look at the secondary controllers, before we later move on to the higher level controlllers. A summary of the different controller parameters can be found in Table 1.1.

Table 1.1: Summary of controller parameters in the plant

Controller	K_p	T_i	T_d
`24_FC1005`	1.18125	5	-
`24_FC1015`	0.2	5	-
`24_FC1019`	0.27225	5	-
`24_LC1028`	47.9	128	-
`24_PC1024`	17.5	2400	-

1.1 Tuning of secondary controllers

Firstly we want to tune the secondary controllers, which do not depend much on which control structure is used for composition control. The controllers this applies to is the flow controllers `24_FC1005`, `24_FC1015` and `24_FC1019`, the level controller `24_LC1028`, and also the pressure controller `24_PC1024`. These controllers all have fast dynamics compared to the composition control and primary level controllers.

1.1.a Tuning of `24_FC1019`

Firstly we tune the controller `24_FC1019`, which is a flow controller controlling the bottom flow B of N-butane. When trying to apply Skogestad IMC-tuning to a first order model, the results were poor as there are close to none time delay, as expected in a valve, which lead to this being a unprecise model. Ziegler-Nichols method, as described in [1], was then a natural choice, and with a critical gain of $K_{p_k} = 0.605$ and integral time $T_i \to \infty$ we got the osciallations shown in Figure 1.1.1. By zooming into this we read the period of the oscillations as $T_k = 6\,\mathrm{s}$. This yields the gain and integral time $K_p = 0.27225$ and $T_i = 5\,\mathrm{s}$. With this tuning we get the response shown in Figure 1.1.2. This is a good response to a relatively large step reference change. Before the reference step the controller has an external set point, that is why the set point is changing dynamically. When doing the step change, the controller settings was changed from external to internal set point, and then back to external for the second step.

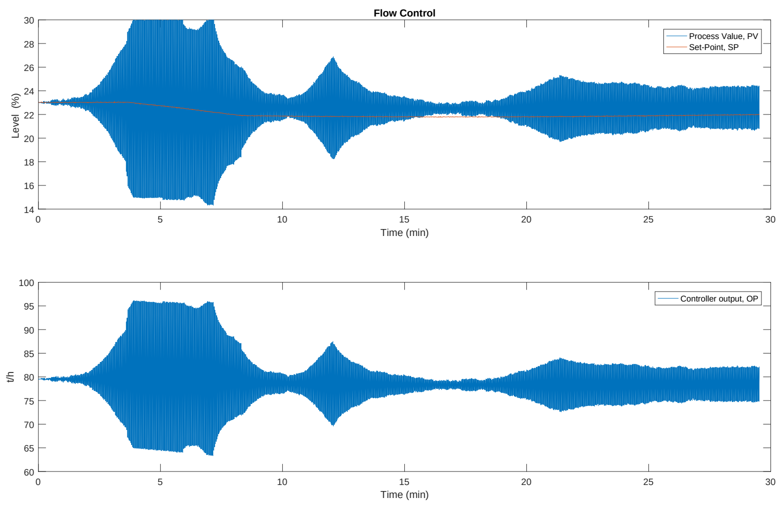

Figure 1.1.1: Oscillations of the flow controller `24_FC1019` with $K_{p_k} = 0.605$ from about 22 min

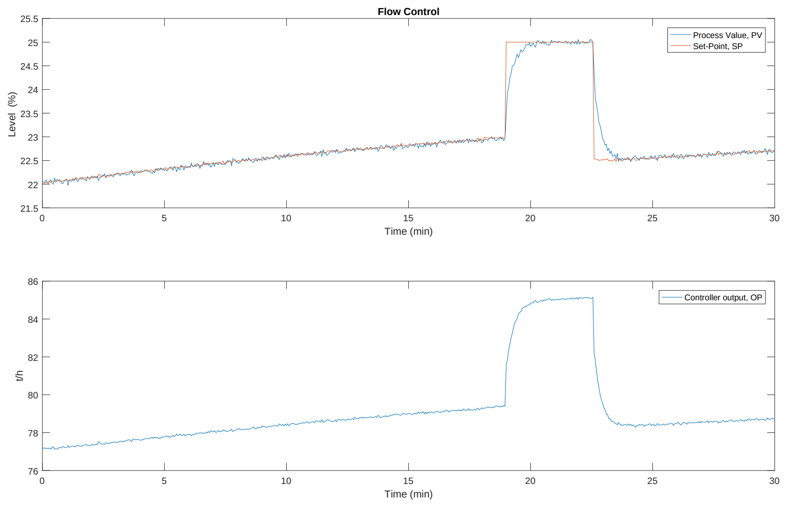

Figure 1.1.2: Response of `24_FC1019` tuned using Ziegler-Nichols method

1.1.b Tuning of 24_FC1005

We now tune the controller 24_FC1005, another flow controller, this one controlling D, the distillate flow of Iso-butane. Using the same method as in the tuning of 24_FC1019, we find the critical values $K_{p_k} = 2.625$ and $T_k = 6$s. This yields the parameters $K_p = 1.18125$ and $T_i = 5$s. As we see in Figure 1.1.4, the we get a slow oscillation when switching back to external reference, but this is due to oscillations in the set point, and not the process value. We see that the process nicely follows the reference with high accuracy.

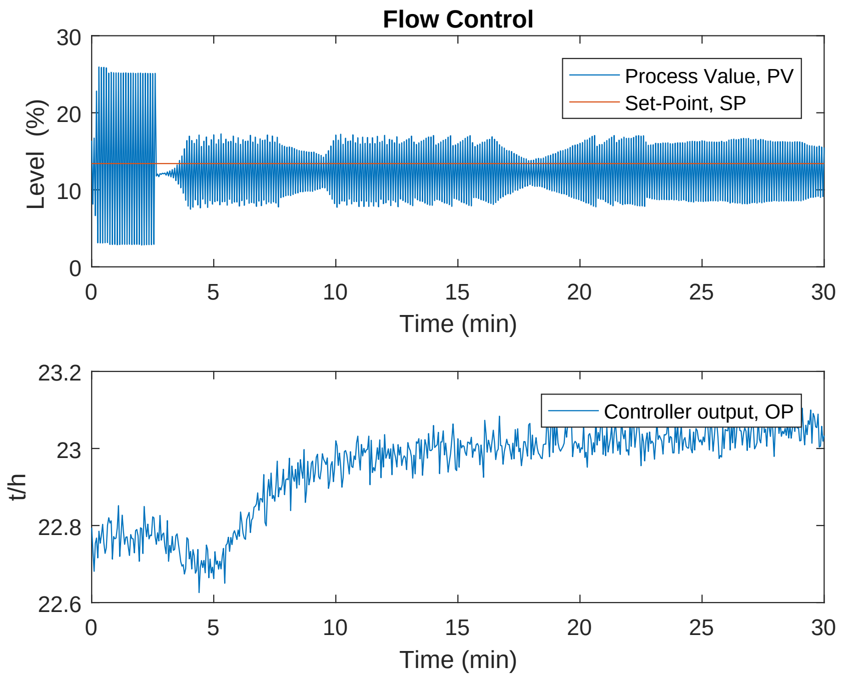

Figure 1.1.3: Oscillations of the flow controller 24_FC1005 with critical gain $K_{p_k} = 2.625$ from about 23 min

1.1.c Tuning of 24_LC1028

Further we tune the level controller 24_LC1028 wich controls the area for heat exhange in the bottom heat exchanger. This makes it indirectly control the bottom vapour flowrate V. Here we use the SIMC-method for tuning of PI(D)-controllers. Using this method we now need to take into account the internal scaling in the controller. By setting the integral time $T_i \to \infty$ we get a simple P-controller which we apply a step in the reference. By using the SIMC-tuning rules from [2] applied on a integrating step response we get

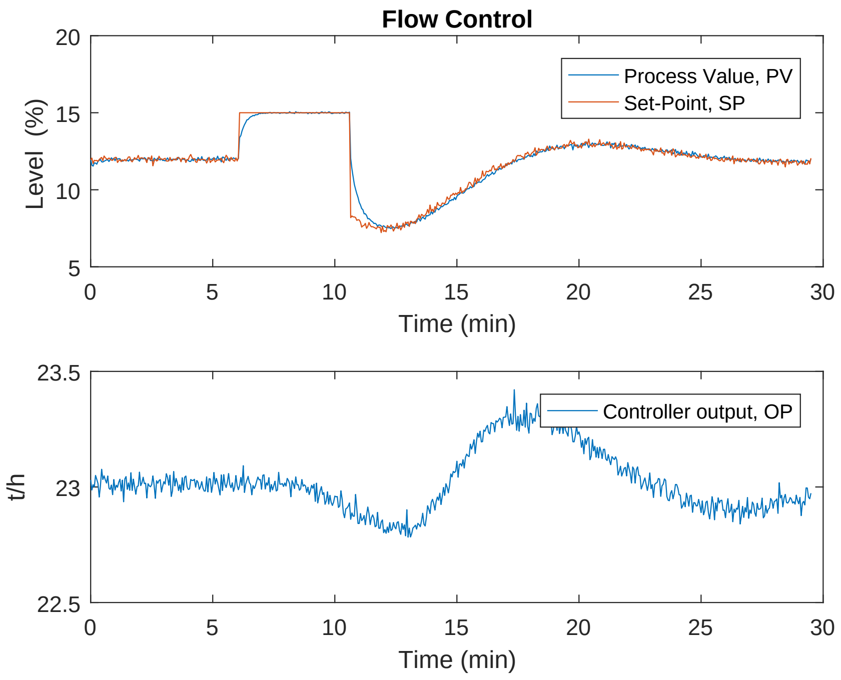

Figure 1.1.4: Response of `FC_1005` tuned using Ziegler-Nichols method

$$
\begin{aligned}
k' &= \frac{\Delta y}{\Delta t \Delta u} \\
K_c &= \frac{1}{k'} \frac{1}{\tau_c + \theta} \\
T_i &= 4(\tau_c + \theta)
\end{aligned}
\tag{1.1.1}
$$

where τ_c is chosen as 3θ to achieve robustness. Smaller τ_c improves preformance, but makes the system less robust, as described in [2]. By measuring on Figure 1.1.5 we get the observations

$$
\begin{aligned}
\Delta u &= 1 \\
\Delta y &= -0.3413 \\
\Delta t &= 462 \\
\theta &= 8
\end{aligned}
\tag{1.1.2}
$$

which again gives the results

$$
\begin{aligned}
K_c &= -42.3 \\
T_i &= 128
\end{aligned}
\tag{1.1.3}
$$

The scaling factor can be calculated as shown in (1.1.6), and when applying a step input with $\Delta y_{\text{ref}} = 2$, we get the immediate response $\Delta u_{\text{meas}} = -17.68$. $\Delta u_{\text{exp}} = K_p \Delta y_{\text{ref}} =$

20, which gives the scaling factor

$$G = \frac{u_{\text{exp}}}{u_{\text{meas}}} = -1.13 \tag{1.1.4}$$

which again gives the controller parameters

$$\begin{aligned} K_{p,\text{applied}} &= GK_c = 47.9 \\ T_i &= 128 \end{aligned} \tag{1.1.5}$$

This again leads to fairly good response, although the controller is maybe slightly aggressive. This is shown in Figure 1.1.6. The equations for the scaling parameter are given as

$$\begin{aligned} \Delta u_{\text{exp}} &= K_p \Delta y_{\text{ref}} \\ G &= \frac{\Delta u_{\text{exp}}}{\Delta u_{\text{meas}}} \\ \implies K_{p,\text{applied}} &= GK_p \end{aligned} \tag{1.1.6}$$

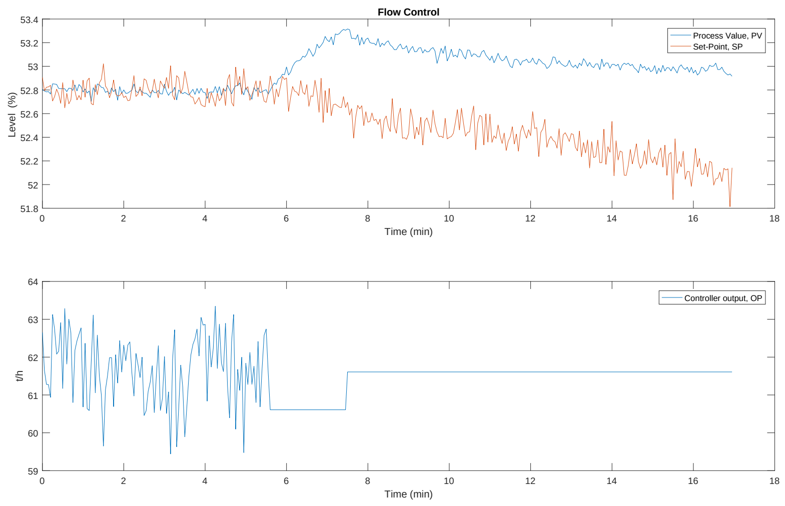

Figure 1.1.5: Output of `24_LC1028` with simple P-controller and applied step at $t \approx$ 7 min

1.1.d Tuning of 24_PC1024

When applying a step oon the input of the pressure controller `PC_1024` we get the output response shown in Figure 1.1.7. When then applying the SIMC method to this output response, we see that we do not have a proper first order model with delay or a clean integrating response with delay, we instead treat the time from the step until the

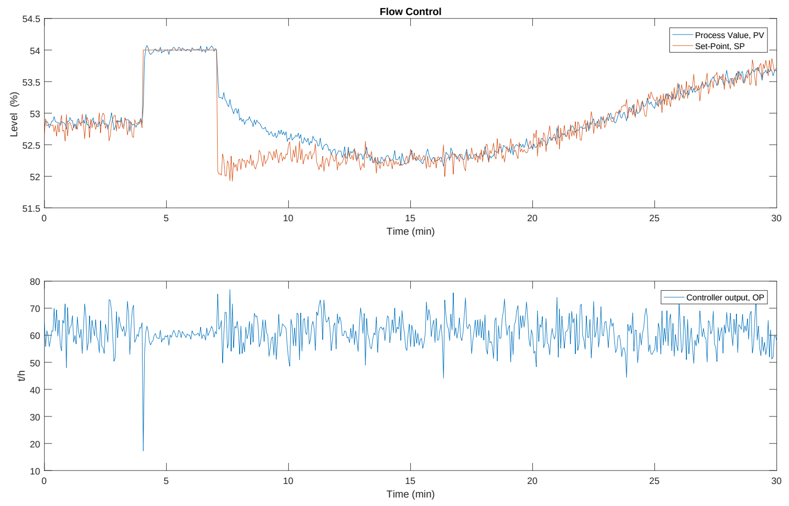

Figure 1.1.6: Output of `24_LC1028` tuned using SIMC and PID-scaling

integration starts properly as the time delay θ. We also calculate the internal gain as in (1.1.6) and obtain the parameters $G = -1.62$, $\delta u = 0.3$, $\delta t = 900$, $\delta y = -0.021$ and $\theta = 300$. Using the rules from (1.1.1) and chosing $\tau_c = \theta$ for faster control we get the controller parameters

$$\begin{aligned} K_p &= GK_c = 17.5 \\ T_i &= 4(\tau_c + \theta) = 2400 \end{aligned} \tag{1.1.7}$$

This integral time seems quite large, but when looking at the response of the `PC_1024` after tuning the parameters, shown in Figure 1.1.8, we see that the controller response is fairly good, and keeps the pressure well within acceptable deviations.

1.1.e Tuning of `24_FC1015`

When it came to the controller `24_FC1015`, I came to the conclusion that all responses were unstable when trying to use the classic tuning methods. I therefore went back to the even more classic method of trial and error. By tweaking K_p and T_i within reasonable limits we hit a set of parameters, $K_p = 0.2$ and $T_i = 5$, which gave fairly good response. This is shown in Figure 1.1.9. We see that the process value follows the reference pretty good.

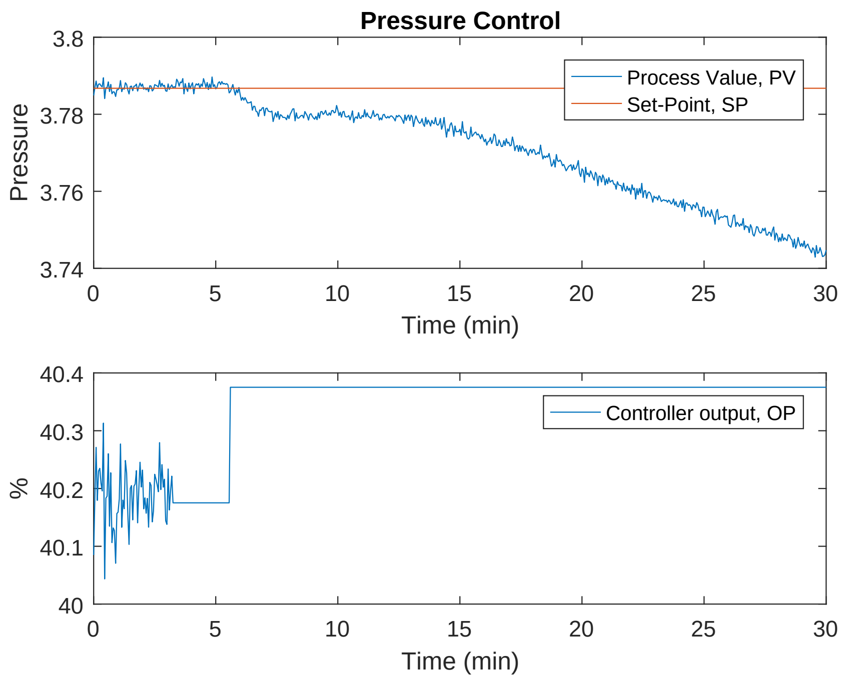

Figure 1.1.7: Response of the controller `PC_1024` when applying a step on input

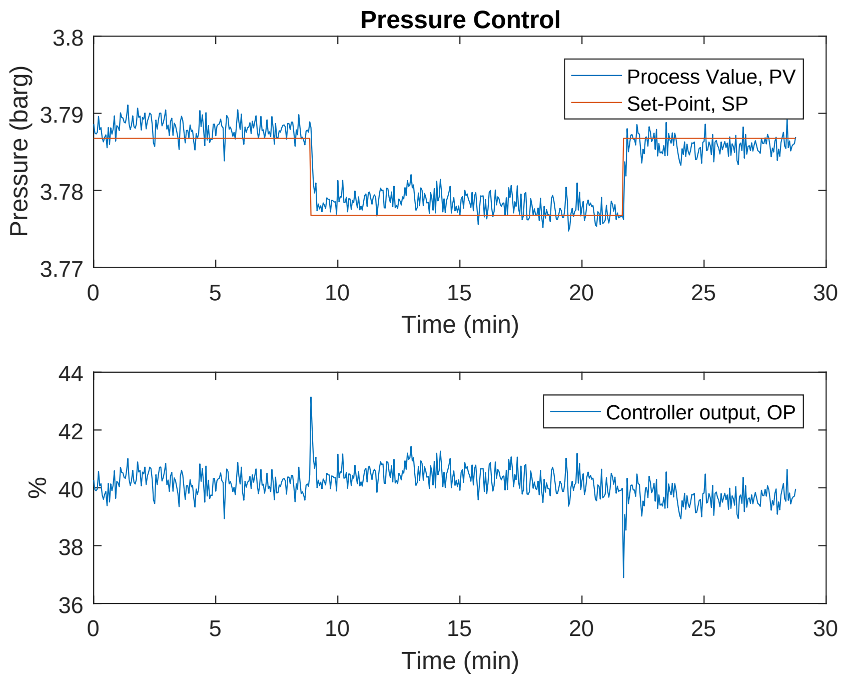

Figure 1.1.8: Response of the pressure controller PC_1024 after tuning and applying a step in the reference value

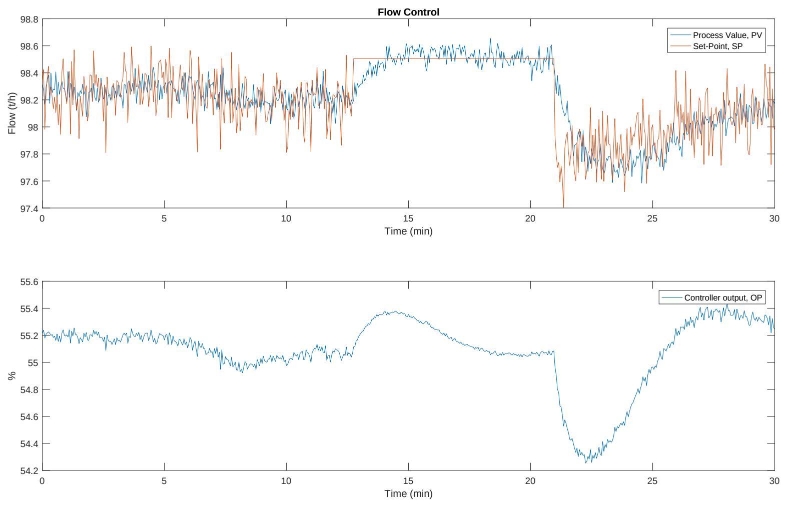

Figure 1.1.9: Response of the controller 24_FC1015 after tuning using trial and error

1.2 Tuning and identification of level controllers

We now move on to the identification and tuning of the level controllers `24_LC1015` and `24_LC1016`. The base code for this is shown in Listing A.1, where minor adjustments is done to the .txt-file we import based on what data we have exported from K-Spice and which contoller we have excited.

1.2.a Level controller `24_LC1016`

We start by doing an set point exciting experiment on the controller `24_LC1016`, where we do several step changes in the set point, and then observe the output. We then import this to MATLAB and use the built in function `n4sid` to estimate a state space realization of the system we are looking at. In Figure 1.2.1 we see the actual response of the controller normalized and compared to the identified model. We see that the identified model is fairly good.

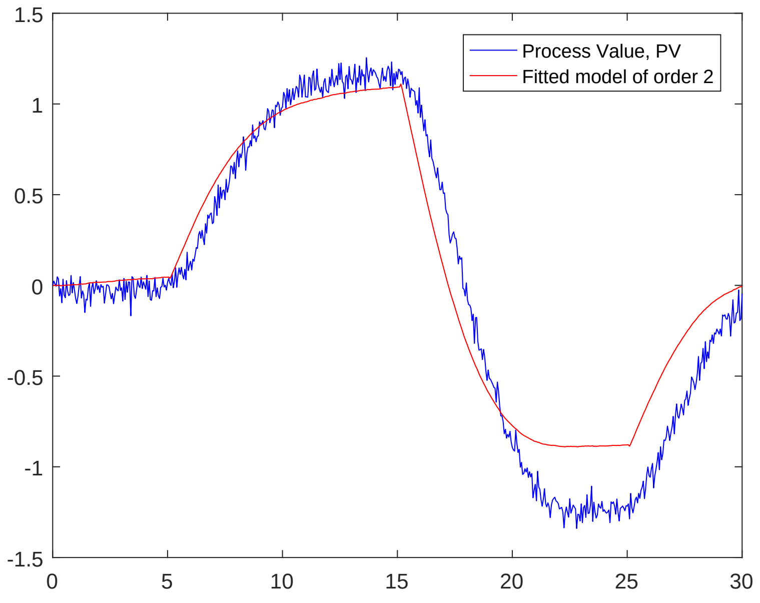

Figure 1.2.1: Actual response and response of identified model of order 2

Bibliography

[1] J.G. Balchen, T. Andresen, and B.A. Foss. *Reguleringsteknikk*. Intitutt for teknisk kybernetikk, NTNU, 2003. ISBN: 8247151472.

[2] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, 2005. ISBN: 9780470011676. URL: https://books.google.no/books?id=3dxSAAAAMAAJ.

A MATLAB code

Listing A.1: MATLAB code to import and identify system of order nx using the command `n4sid`, before plotting estimated response together with the actual response of the system

```
%% cleanup
close all
clear variables
clc

%% Get data from the log file
% Make sure to use the correct path for the log file
fileID=fopen('Logging.lst_lc1016_exp_2.txt','r'); % This loads
   the data log file
for m = 1:35
    String_Row=fgetl(fileID); % Ignore first 35 rows in the txt
        file
end

i = 1;
while(ischar(String_Row));          % Continue until the end of
   file
       String_Row=fgetl(fileID);   % Read row from txt file
    if ischar(String_Row) ~= 0
        Num_Vector = str2num(String_Row);    % Converts string
           number a vector
        Data(i,:) = Num_Vector;              % Store rows into a
            "Data" Matrix
    end
     i = i + 1;
end
fclose(fileID); % Close the data log file

%% Controller data
Time = Data(:,1);            % Time
PC1024 = Data(:,2:4);        % Controller: 24_PC1024
FC1005 = Data(:,5:7);        % Controller: 24_FC1005
```

```
FC1019 = Data(:,8:10);      % Controller: 24_FC1019
LC1016 = Data(:,11:13);     % Controller: 24_LC1016
LC1015 = Data(:,14:16);     % Controller: 24_LC1015
FC1015 = Data(:,17:19);     % Controller: 24_FC1015
LC1028 = Data(:,20:22);     % Controller: 24_LC1028
TC1015 = Data(:,23:25);     % Controller: 24_TC1015
TC1088 = Data(:,26:28);     % Controller: 24_TC1088
% Where the:
% First column is the Process Value, PV
% Second column is the Set-Point, SP
% Third column is the Control Signal, OP
% Ex.
% PC1024(:,1) = Process Value, PV
% PC1024(:,2) = Set-Point, SP
% PC1024(:,3) = Control signal, OP

%% normalize data
cut = 1;

controller = LC1016;
controller_cut = controller(cut:end,:);
y = controller_cut(:,1) - controller_cut(1,1);
y_ref =  controller_cut(:,2) - controller_cut(1,2);
u = controller_cut(:,3) - controller_cut(1,3);
Time_cut = Time(cut:end) - Time(cut);


%% system identification
Ts = 1;
data = iddata(y,u,Ts);
nx = 2;
sys = n4sid(data,nx);

%%
y_sim = lsim(sys,u);

%% plot
figure
plot(Time_cut./60, y, 'b'); hold on;
plot(Time_cut./60, y_sim, 'r');
%plot(Time_cut./60, y_ref);
legend('Process Value, PV', ['Fitted model of order ' num2str(
   nx)] , 'Reference value');
```