Exercises simulator automatisation 3th Bachelor marine engineering

Lab exercises sessions 3 and 4: AUTOTUNING

**Adjusting the controlparameters by the method of Ziegler-Nichols (and Strejc-Broida)**

**a)Ziegler-Nichols Method:** (tuning a controller) 

1. First, note whether the required proportional control gain is positive or negative. To do so, increase a little, under manual control, to see if the resulting steady state value of the process output has also moved up (increased). If so, then the steady-state process gain is positive and the required Proportional control gain, *K*c, has to be positive as well.
2. Turn the controller to P-only mode, i.e. turn both the Integral and Derivative modes off.
3. Turn the controller gain, *Kc*, up slowly (more positive if *Kc* was decided to be so in step 1, otherwise more negative if *Kc* was found to be negative in step 1) and observe the output response. Note that this requires changing *Kc* in step increments and waiting for a steady state in the output, before another change in *Kc* is implemented.
4. When a value of *Kc* results in a **sustained periodic oscillation** in the output (or close to it), mark this critical value of *Kc* as *Ku*, the ultimate gain. Also, measure the period of oscillation, *Pu*, referred to as the ultimate period. Using the values of the ultimate gain, *Ku*, and the ultimate period, *Pu*, Ziegler and Nichols prescribes the following values for *Kc*, *tI* and *tD*, depending on which type of controller is desired:

**Ziegler-Nichols Tuning Chart:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Kc*** | ***I*** | ***D*** |
| **P control** | *Ku*/2 |  |  |
| **PI control** | *Ku*/2.2 | *Pu*/1.2 |  |
| **PID control** | *Ku*/1.7 | *Pu*/2 | *Pu*/8 |

**(original values:P=12;I=60;D=3)**

**Use the values of the PID settings of the cases where the system was instable in exercise 1 and try to stabilize the system in this exercise using the above formula’s.**

**b) Method by Strejc-Broida:** (systemidentification)

Strejc and Broida take a arbitrary system as follows H(s)=Keτ/(1+τv s)n

|  |
| --- |
| \includegraphics[width=12cm]{Courbes/FigRepStrejc.eps} |

1. Take a trend of a change in temperature and print it out
2. Draw on your print out the tangent line as in the figure above and measure T1 and T2
3. Calculate n by calculating T1/T2
4. This value for n gives you the values for τ and τv

|  |
| --- |
|  |
| |  |  |  |  | | --- | --- | --- | --- | | $ n$ | T1/τv | T2/τ | T1/T2 | | 2 | 0,2817 | 2,718 | 0,1036 | | 3 | 0,8055 | 3,695 | 0,2180 | | 4 | 1,425 | 4,464 | 0,3194 | | 5 | 2,102 | 5,112 | 0,4103 | | 6 | 2,811 | 5,699 | 0,4933 | | 7 | 3,549 | 6,226 | 0,5700 | | 8 | 4,307 | 6,711 | 0,6417 | | 9 | 5,081 | 7,164 | 0,7092 | | 10 | 5,869 | 7,590 | 0,7732 | |

**Note:**

ThePID-controller uses the next controlalgorithm for the output u(t):


u(t) = Kr \cdot \left(e(t) + \frac {\int e(t)dt} {Ti} + Td \cdot \frac{d e(t)}{dt}\right)


e(t) :difference between procesvalue PV(t) and setpunt SP(t)

e(t) = PV(t) - SP(t)

Laplacetransformation:


H(s) = Kr \cdot (1 + \frac {1} {Ti \cdot s} + Td \cdot s)


* H(s) = transferfunction for the controller
* Kr = proportional action (P-actie)
* Ti = [integrationtime](http://nl.wikipedia.org/wiki/Integraalrekening) (I-action): the smaller Ti, the more I-action
* Td = [differentiati](http://nl.wikipedia.org/wiki/Differentiaalrekening)ontime (D-action): the larger Td, the more D-action

**3) Reaction of the actuator**

Not only controlling the process is important also the reaction of the actuator. In this case we look at the opening(closing) of a valve.

If we set the parameters of the controller thus to have in the end a fast system without too much overshoot it just could be that the valve reacts in an aggressive manner. This could result in damages to the actuator (valve).

Look at the reaction for P=12/I=80/D=8

P=12/I=80/D=1

P=12/I=80/D=18

P=12/I=80/D=25