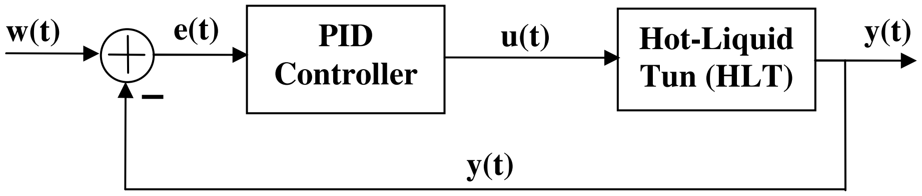
1. Introduction

This document describes the derivation of a PID controller that can be implemented in the STC1000p. The PID controller should be capable of controlling the temperature to within 0.1 °C.

This document contains the following information

- Chapter 2: the time-continuous representation for a PID controller
- Chapter 3: Derivation of an improved algorithm (a so-called ‘type C’ PID controller)
- Chapter 4: Description of algorithms for finding the optimum set of K_c, T_i, T_d and T_s values for the PID controller

2. time-continuous representation of a PID controller



The generic equation[1] for a PID controller in the time-continuous domain is:

$$u(t) = K_c . \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d . \frac{de(t)}{dt} \right) \qquad \textit{(eq. 01)}$$

With:	$K_c = K_p$	Proportional Gain (for our temperature controller, unity is [% / °C])
	$T_i = K_c / K_i$	Time-constant Integral gain [sec.]
	$T_d = K_d / K_c$	Time-constant Derivative gain [sec.]
	T_s	Sample period (default value is 20 seconds)
	w(t)	Setpoint (SP) value for temperature
	e(t)	error signal = setpoint w(t) – process variable y(t)
	u(t)	PID output signal, also called Gamma, ranges from [0..100 %]
	y(t)	process variable PV = measured temperature

3. Derivation of a Type C PID controller

There are three types of PID equations (see http://bestune.50megs.com/typeabc.htm) , with type C being the preferred one. Equation eq. 01 are type A equations, since the P- and the D-term both contain the setpoint. Any changes in the set-point may cause an unwanted change in the PID output u(t).
Removing the set-point from the D-term results in a type B controller. The type C controller has also removed the set-point from the P-term, resulting in an even better PID controller implementation.

Start with the generic equation for a PID controller again:

$$u(t) = K_c . \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d . \frac{de(t)}{dt} \right) \qquad \textit{(eq. 01)}$$

[1] This is the ideal, textbook version of a continuous-time PID controller. See “*Digital Self-Tuning Controllers*”, ISBN 1-85233-980-2, Bobál et.al., page 54.

Now differentiate both sides:

$$\frac{du(t)}{dt} = K_c.\left(\frac{de(t)}{dt} + \frac{e(t)}{T_i} + T_d.\frac{d^2e(t)}{dt^2}\right) \qquad \textit{(eq. 02)}$$

Now transform equation eq. 02 to the time-discrete domain, using backwards differentiation:

$$\frac{u_k - u_{k-1}}{T_s} = +K_c.\left[\frac{e_k - e_{k-1}}{T_s} + \frac{e_k}{T_i} + \frac{T_d}{T_s^2}.(e_k - 2.e_{k-1} + e_{k-2})\right] \qquad \textit{(eq. 03)}$$

Equation eq.03 can now also be written as:

$$u_k = u_{k-1} + K_c.\left[(e_k - e_{k-1}) + \frac{T_s.e_k}{T_i} + \frac{T_d}{T_s}.(2.y_{k-1} - y_k - y_{k-2})\right] \qquad \textit{(eq. 04)}$$

Equation 04 is a type B equation, because the differential term no longer depends upon w_k (remember that $e_k = w_k - y_k$).

We will now transform equation eq.04 into a type C equation (eq. 05), by also removing the setpoint w_k from the proportional part (again: $e_k = w_k - y_k$):

$$\boxed{u_k = u_{k-1} + K_c.\left[y_{k-1} - y_k + \frac{T_s}{T_i}.e_k + \frac{T_d}{T_s}.(2.y_{k-1} - y_k - y_{k-2})\right]} \qquad \textit{(eq. 05)}$$

Here, y_k is the process variable, which is the actual temperature of the HLT. This is the well-known relation given by Takahashi[2], also known as the Takahashi PID controller.

Equation 05 is used directly in pid_ctrl().

[2] Y. Takahashi, C. Chan and D. Auslander, "*Parametereinstellung bei linearen DDC-algorithmen*", Regelunstechnik und Prozessdatenverarbeitung, vol. 19, pp. 237-284, 1971

4. Finding the optimum set of PID parameters

Finding the optimum parameters for a PID controller can be difficult. Optimum means that the set-point temperature is reached as quickly as possible with overshoot minimised.

Three well-known algorithms for determining the PID parameters are described here:

- Ziegler-Nichols open-loop: set PID controller to a certain output and determine slope and dead-time of HLT system
- Ziegler-Nichols closed-loop: measure step-response
- Cohen-Coon: also a closed-loop method. Measure step-response
- Integral of the time weighted absolute error (ITAE): results in the best performance. The error signal is minimised (over time).

Some terms are frequently used in this document:

- **Dead-time θ:** this is the time-delay between the initial step and a 10% increase in the process variable (the HLT temperature in our case).
- **K_{hlt}:** the gain of the HLT-system. The HLT-system receives the Gamma value (PID output) as input and has the HLT temperature as output. Unity of K_{hlt} is [°C / %].
- **τ_{hlt}:** the time-constant of the HLT-system. The HLT-system can be described with a first-order process model with time-delay (FOPTD). The transfer function for this model is:

$$HLT(s) = \frac{K_{hlt}.e^{-\theta.s}}{\tau_{hlt}.s+1}$$

- **a***: the normalised slope of the step response. Equal to ΔT / (Δt. Δp) with:
  - ΔT: change in temperature [°C]
  - Δt : change in time [seconds]
  - Δp: change in PID controller output [%]

With these three parameters, the optimum PID parameters are determined using table 1 on the next page (values are given both for PID operation and for PI-only operation):

Method: / Parameter:	**K_c [% / °C]**	**T_i [seconds]**	**T_d [seconds]**
Ziegler-Nichols Open-loop	$K_c = \frac{1{,}2}{\theta . a*}$	$T_i = 2{,}0.\theta$	$T_d = 0{,}5.\theta$
Ziegler-Nichols Open-loop	$K_c = \frac{0{,}9}{\theta . a*}$	$T_i = 3{,}33.\theta$	**- -**
Ziegler-Nichols Closed-loop	$K_c = \frac{1{,}2.\tau_{hlt}}{K_{hlt}.\theta}$	$T_i = 2{,}0.\theta$	$T_d = 0{,}5.\theta$
Ziegler-Nichols Closed-loop	$K_c = \frac{0{,}9.\tau_{hlt}}{K_{hlt}.\theta}$	$T_i = 3{,}33.\theta$	**- -**
Cohen-Coon	$K_c = \frac{\tau_{hlt}}{K_{hlt}.\theta}.\left(\frac{\theta}{4.\tau_{hlt}} + \frac{4}{3}\right)$	$T_i = \theta.\frac{32.\tau_{hlt} + 6.\theta}{13.\tau_{hlt} + 8.\theta}$	$T_d = \theta.\frac{4.\tau_{hlt}}{2.\theta + 11.\tau_{hlt}}$
Cohen-Coon	$K_c = \frac{\tau_{hlt}}{K_{hlt}.\theta}.\left(\frac{\theta}{12.\tau_{hlt}} + \frac{9}{10}\right)$	$T_i = \theta.\frac{30.\tau_{hlt} + 3.\theta}{9.\tau_{hlt} + 20.\theta}$	- -
ITAE-Load	$K_c = \frac{1{,}357}{K_{hlt}}.\left(\frac{\theta}{\tau_{hlt}}\right)^{-0{,}947}$	$T_i = \frac{\tau_{hlt}}{0{,}842}.\left(\frac{\theta}{\tau_{hlt}}\right)^{0{,}738}$	$T_d = 0{,}381.\tau_{hlt}.\left(\frac{\theta}{\tau_{hlt}}\right)^{0{,}995}$
ITAE-Load	$K_c = \frac{0{,}859}{K_{hlt}}.\left(\frac{\theta}{\tau_{hlt}}\right)^{-0{,}977}$	$T_i = \frac{\tau_{hlt}}{0{,}674}.\left(\frac{\theta}{\tau_{hlt}}\right)^{0{,}680}$	- -

Table 1: optimum PID parameters for the various methods

To be able to find these three parameters accurately, two experiments need to be conducted. These two experiments are described in the next two paragraphs.
The last paragraph (§ 4.3) shows the calculated PID parameters for all these methods.

4.1 Experiment 1: Determine dead-time Θ of HLT-system

Manually set the PID-controller output ("gamma") to a certain value (e.g. 20 %). In case of a heavy load, 100 % is recommended (more accurate), but if you don't know the performance of the system, a lower value to start with is better. The temperature starts to increase and follows the curve in figure 1.

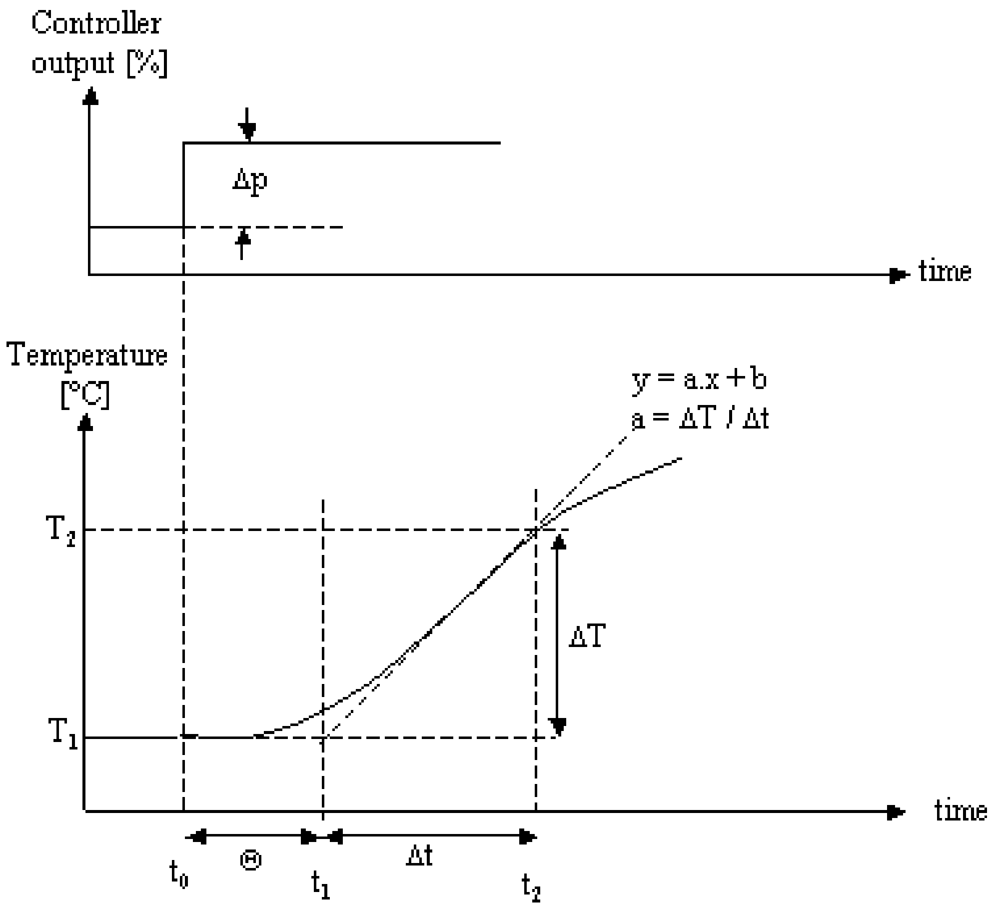


Figure 1: Open-loop response of the system

Calculate the average slope of the rise ($a = \Delta T / \Delta t$) by using regression analysis if possible.
Calculate the normalised slope a*, which is defined as: $a^* = a / \Delta p$.
The dead-time **θ** is defined as the time between t_0 and t_1 (by using regression analysis, this can be done quite accurately).

<u>Experimental data: HLT filled with 85 L water</u>

19:08:20	t_0, the PID controller output gamma was set to 100 %, T_1 = 47,41 °C
	Step-response up to 55.00 °C, ΔT = 7,59 °C
19:12:25	T_{hlt} = 48,18 °C (≥ T_1 + 10%.ΔT = 48,17 °C)
19:27:21	T_{hlt} = 54,25 °C (≥ T_1 + 90%.ΔT = 54,24 °C)

Regression analysis of the data between 19 :12 :25 and 19:27 :21 resulted in the following:
y = a.x + b = 0,0334x + 48,278 R^2 = 0,9995
Here, every data-point for x represents 5 seconds. Therefore, the average slope a is equal to:
a = 0,0334 °C / 5 sec. = 6,68E-03 °C/second (which is 0,4 °C/minute)
Now solve where this curve hits the Tenv line:
x = (47.41 – 48.278) / 6.68E-03 = -130 seconds (or 2 minutes and 10 seconds)
The dead-time moment is then 19:12:25 – 2:10 = 19:10:15
Therefore, the **dead-time θ = 19:10:15 – 19:08:20 = 1:55 = 115 seconds**

The normalised slope a* is equal to :

$a^* = a / \Delta p = 6{,}68E\text{-}05$ °C/(%.second)

4.2 Experiment 2: Determine gain K_{hlt} and time-constant τ_{hlt} of HLT-system

Manually set the PID-controller output (“gamma”) to a certain value (e.g. 20 %). The temperature starts to increase and follows the curve in figure 2.

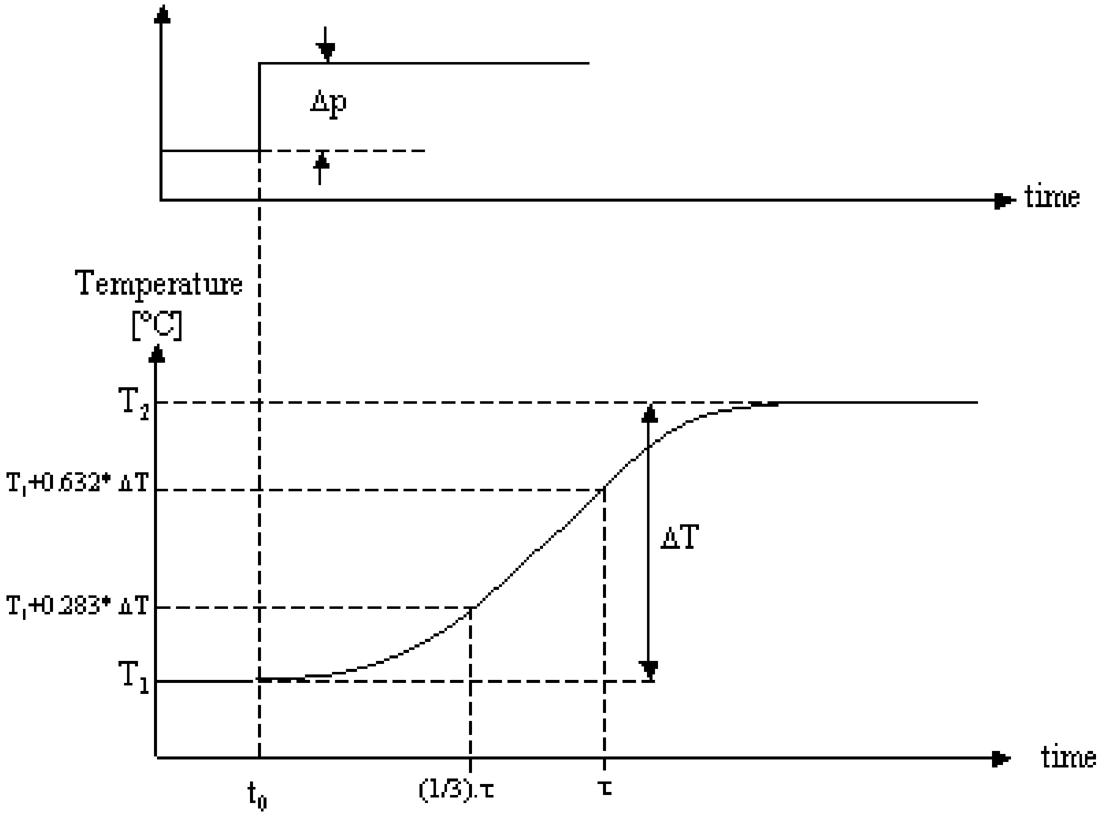


Figure 2: Step response of the system

Because of the large time-constant presumably present in the HLT system, the step response is not very accurate in determining the dead-time **θ**. This is the mean reason to conduct two experiments (theoretically, the step-response would give you all the required information to determine the three parameters).

Experimental data: HLT filled with 85 L water

09:49:04 t_0, the PID controller output gamma was set to 20 %, $T_1 = 19{,}20$ °C

20:24:03 Experiment stopped, T = 52,98 °C.

Regression analysis (2nd order polynomial) used to find maximum.

Maximum found at 21:20:44 (8300 ticks of 5 seconds). **T_2 = 52,989 °C**

$\Delta T = 52{,}989 - 19{,}20 = 33{,}789$ °C

$T(t = 1/3.\tau_{hlt}) = T_1 + 0{,}283 * \Delta T = 28{,}762$ °C => $1/3.\tau_{hlt} = 6599$ seconds

$T(t = \tau_{hlt}) = T_1 + 0{,}632 * \Delta T = 40{,}555$ °C => $\tau_{hlt} = 16573$ seconds

Solve for τ_{hlt}: $\tau_{hlt} - 1/3.\ \tau_{hlt} = 16573 - 6599$ => **τ_{hlt} = 14961 seconds**

Solve for Gain K_{hlt}: $K_{hlt} = \Delta T / \Delta p = 33{,}789$ °C / 20 % = **1,689 °C / %**

4.3 Calculation of PID parameters for the various methods

In the previous paragraphs, the following parameters were found:

- **dead-time θ = 115 seconds**
- **Gain K_{hlt} = 1,689 °C / %**
- **Time-constant τ_{hlt} = 14961 seconds**
- **Normalised slope a* = a / Δp = 6,68E-05 °C/(%.second)**

	Kc [%/°C]	Ti [sec]	Td [sec]	
Ziegler-Nichols Open Loop	156,2	230,0	57,5	PID
	117,2	383,0	- -	PI
Ziegler-Nichols Closed Loop	92,4	230,0	57,5	PID
	69,3	383,0	- -	PI
Cohen-Coon	102,8	282,2	41,8	PID
	69,4	377,2	- -	PI
ITAE-Load	80,8	489,0	44,9	PID
	59,2	810,2	- -	PI

Table 2: Calculation of parameters for the various methods

Version History

Date	Version	Description
8-11-2016	V0.1	First version for Github STC1000p-STM8