Vamos utilizar o Ziegler-Nichols em malha aberta, pois ele funciona bem para processos com constante de tempo de pelo menos duas vezes a de “tempo morto”, e isso costuma ocorrer para temperatura. Além disso, não é necessário escolher aleatoriamente os ganhos (Kp , Ki , Kd) e a implementação é relativamente simples.

Os parâmetros iniciais obtidos com esta técnica fornecerão uma base sólida para ajustes subsequentes, garantindo que o sistema atenda aos requisitos específicos de overshoot máximo de 2°C e aquecimento rápido. A técnica de Ziegler-Nichols é amplamente validada na literatura e em aplicações industriais, proporcionando confiança adicional na sua aplicação para este projeto de controle de temperatura.

Uma das desvantagens do método Ziegler-Nichols observada na literatura é que ele pode resultar em sintonização menos precisa e overshoot elevado, vamos observar se isso irá ocorrer.

Vamos alterar o código fornecido pelo Prof. Rodrigo M Bacurau para transformar o algoritmo de controlador paralelo (Figura 1) em um algoritmo de controlador interativo (Figura 2), pois o processo de Tuning é feito para a representação interativa.

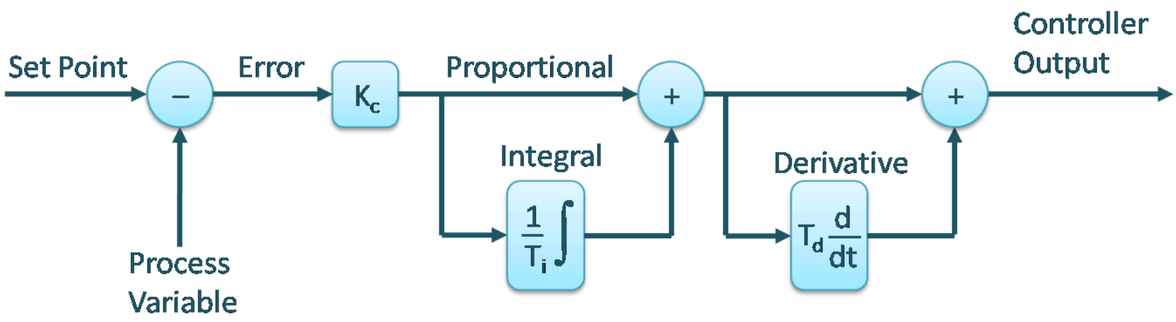

Figura 1: Modo iterativo

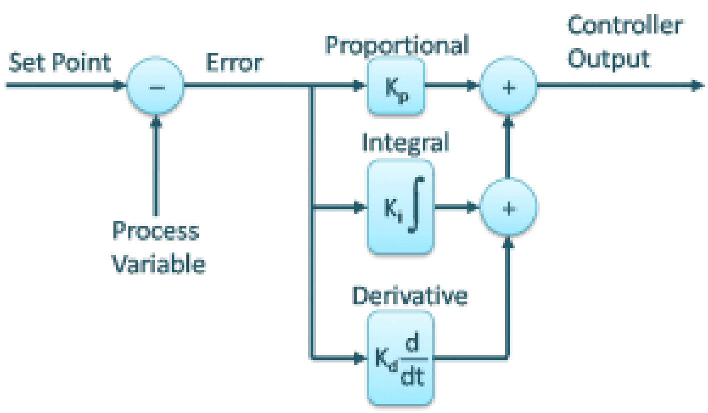

Figura 2: Modo paralelo

Para determinar t_d, tau, e g_p que são necessários para encontrar K_c, T_i, e T_d desenvolvemos um código em Python localizado na pasta **/jupyterNotebookForPlottingData**, baseado no tutorial https://blog.opticontrols.com/archives/477.

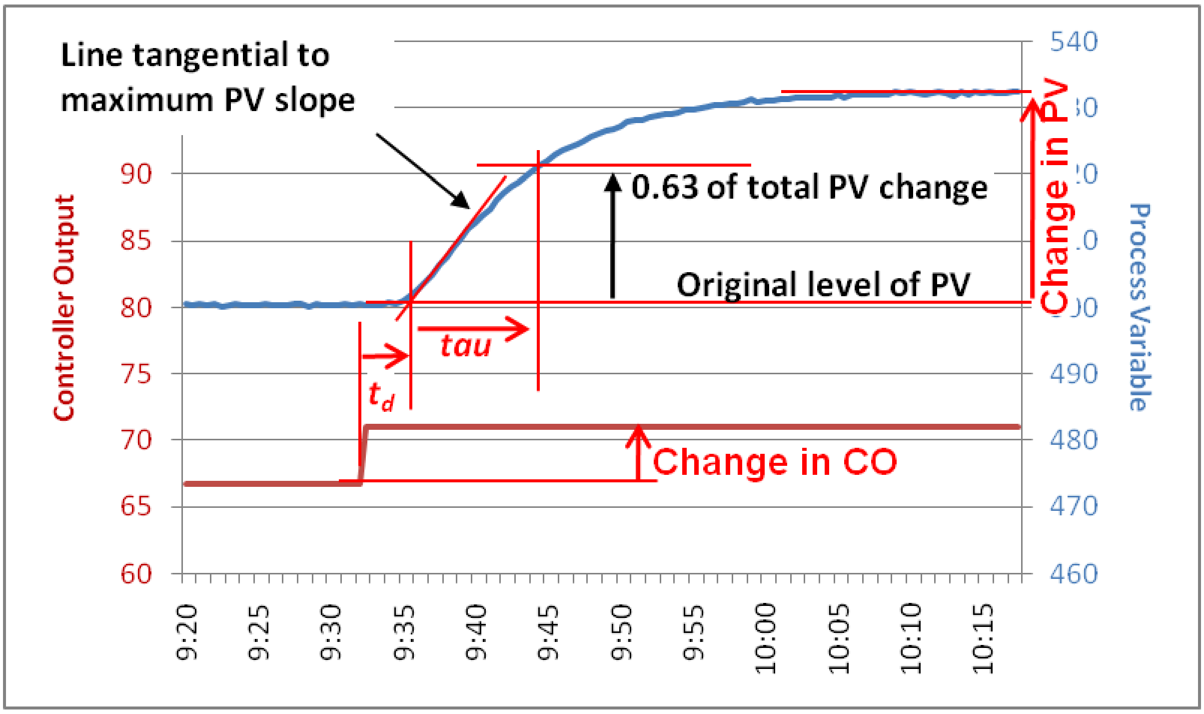

Figura 3: Variáveis necessárias para encontrar os valores do modo iterativo do PID pelo método de Ziegler-Nichols de malha aberta

Como as especificações dadas pelo Professor referem-se ao modo paralelo do PID, foi necessário converter K_c, T_i, e T_d em K_P, K_i, e K_d.

Os valores encontrados foram:

$K_c = 31.35$, $T_i = 12.5$, $T_d = 3.125$

$K_P = 31.35$, $K_i = 2.50$, $K_d = 98$

A *primeira* tentativa de sintonia com o método de Ziegler-Nichols de malha aberta não funcionou, pois o overshoot foi de **3,3%**, um overshoot elevado, situação já esperada, sendo ela uma desvantagem do método. Os dados desse teste estão em **controlOutputWithValuesFromNiegler-Zichols.csv**

A primeira otimização sugerida pelo artigo é dividir o K_c por 2, obtendo os valores

$K_P = 15.67$, $K_i = 1.25$, $K_d = 49$

A *segunda* tentativa de sintonia com o método de Ziegler-Nichols não atingiu o setpoint. Os dados desse teste estão em **controlOutputWithValuesFromNiegler-ZicholsDividedByTwo.csv**

A *terceira* tentativa de sintonia com o método de Ziegler-Nichols foi aumentar o K_p para 130, de forma a diminuir o overshoot. Os valores testados foram:

$K_P = 31.35$ **,** $K_i = 2.50$ **,** $K_d = 130$

A terceira tentativa de sintonia com o método de Ziegler-Nichols resultou em um overshoot de **3.18%**.

Os dados desse teste estão em **controlOutputWithValuesIncreasingKd.csv**

A “*quarta”* tentativa de sintonia com o método de Ziegler-Nichols foi feita por tentativa por tentativa e erro, aumentando o K_p e K_i. Como aumentar somente o $K_d = 130$ não surtiu o efeito esperado, nem diminuir K_p e K_i, fomos aumentando K_p e K_i e observando o controle, os últimos valores testados foram:

$K_P = 400$ **,** $K_i = 4.6$ **,** $K_d = 98$

Para chegar nesses dados tomamos como base inicial os valores encontrados na malha aberta de Ziegler-Nichols $K_P = 31.35$ **,** $K_i = 2.50$ **,** $K_d = 98$ e fomos aumentando e testando o K_P e o K_i múltiplas vezes, sempre observando o erro estacionário e o overshoot até chegarmos em um valor satisfatório. Os dados desse teste estão em **controlOutputWorkingWellAfterTryAndError.csv**

O overshoot desse teste foi de **1,58%**, abaixo do máximo de 2%.