PID auto-tuning

UNICOS

MOC3002, Feedback systems & Tuning

*Enrique Blanco* (CERN, EN/ICE)

on behalf of the UNICOS team
Outline

1. Introduction
2. CERN installations (UNICOS)
3. PID auto-tuning: Solution and methods
4. Experimental results
5. Conclusions
Introduction

- The **PID** feedback control algorithm dates back to early nineteens’
- The right tuning of those controllers is essential to increase plant availability and maximize profits in many processes
- Operators and control engineers spend a considerable time in tuning those controllers
- **GOAL:** (1) Provide automatic methods to tune the PID controllers and (2) make them usable by plant operators
PID control basics

• Basics

SISO control loop schema

- PID are usually tuned by operators (or control engineers).
- The majority of the controllers in industry are PIs

\[ K_c \cdot \left( e(t) + \frac{1}{T_i} \cdot \int e(t) \, dt + T_d \cdot \frac{d}{dt} e(t) \right) \]

Parameters: \( K_c, T_i, T_d \)
CERN installations

- **Industrial facilities** for the accelerator complex and the associated experiments
- Continuous process control: temperature, pressure, levels…
- Large and/or complex dynamics

LHC 4.5 K Cryogenics refrigerators  TPC gas system  LHC cooling towers  CO2 Cooling (MARCO)
CERN installations

Enormous number of PID based controllers: > 8000
- LHC cryogenic control system: ~ 5000 PID controllers
- Cooling and Ventilation: ~ 870 PID controllers

PIDs tuned extremely conservative or initial parameters untouched
- Fine tuning systematically avoided

Implementation:
- **PLCs** (Programmable Logic Controller): > 400
- **UNICOS** (Unified Industrial Control System) framework.
PID auto-tuning

Create a tool to **automatically find the PID parameters** and tune the control loops always initiated deliberately by operators and/or control engineers.

Classification (one) on how the data is extracted from the plant

- **Open loop**: e.g. turn off PID and excite the process by changing the MV
- **Close loop**: e.g. tune online while the PID is working (SP is changed)
PID auto-tuning

Tuning methods

- Trial & Error
- Experimental based
- Model based analytical
- Automatic tuning: Auto-tuning methods
  - Relay Method
  - SIMC (*Skogestad* Internal Model Control)
  - IFC (Iterative Feedback Tuning)

The choice is not straightforward and depends mostly on the process knowledge.
PID auto-tuning: [1] Relay

Astrom and Hagglund (1995)

The process is brought to oscillation by replacing the PID controller with a relay function. The ultimate Gain ($K_{cu}$) and the ultimate period ($T_u$) are determined.

User parameters
- Maximal deviation of the control effort: MV
- Number of cycles to detect the ultimate condition

Advantages
- Single action (vs. trial and error)
- Little a priori knowledge of the process

Disadvantages
- Preferably to execute it under stable conditions
- Process must be controllable with a P-controller.

<table>
<thead>
<tr>
<th>Type</th>
<th>$K_c$</th>
<th>$T_i$</th>
<th>$T_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.5 $K_u$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>0.4 $K_u$</td>
<td>0.8 $T_u$</td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>0.6 $K_u$</td>
<td>0.5 $T_u$</td>
<td>0.125 $T_u$</td>
</tr>
</tbody>
</table>
PID auto-tuning: [2] SIMC

Method based on an internal model (Skogestad IMC)
- Two phases: Process identification (first or second order) and application of tuning rules

Parameterization
- Desired performance: Tight vs. smooth control

Advantages
- Simplicity of parameterization

Disadvantages
- Applicable to processes without complex dynamics
- Stable processes: open loop test

\[ k' = \frac{k}{\tau_1} \]
\[ \tau_c \geq 0: \text{desired closed-loop response time (tuning parameter)} \]
For robustness select: \( \tau_c, \theta \) (gives \( K_c \leq K_{c,\text{max}} \))
For disturbance rejection select: \( K_c \geq K_{c,\text{min}} = \frac{u_{dd}}{y_{\text{max}}} \)

\[ K_c = \frac{1}{k'} \cdot \frac{1}{(\theta + \tau_c)} \]
\[ \tau_I = \min(\tau_1, 4(\tau_c + \theta)) \]
PID auto-tuning: [3] IFT

*Iterative Feedback Tuning* (IFT): inspired in the iterative parametric optimization approach. Makes random perturbations on the SP. Minimize the current value of the measured value and a desired first order response.

**Parameterization**
- Just safeguards (thresholds)
- Desired 1st order response shape

**Advantages**
- close loop method with minimal disturbances
- model free

**Disadvantages**
- A local minima could be found

\[
J(\rho) = \frac{1}{2N} \left[ a \sum_{t=1}^{N} (L_y \cdot \tilde{y}_t(\rho))^2 + l \sum_{t=1}^{N} (L_u \cdot u_t(\rho))^2 \right]
\]

\[
\rho^* = \arg\min_{\rho} J(\rho)
\]

\[
\tilde{y}_t = y_t - y_d
\]
UNICOS Implementation

- Algorithms in the *WinCC OA* SCADA: Scripting language
- The *PLC* maintains the PID algorithm untouched
- HMI inside the PID controller faceplate
Experimental results *(Ghe Flow control)*

Use case: **Cryogenics flow control: PI control**

*Simulation*

Gas helium circulating to maintain the thermal shielding of the LHC superconducting magnets at 80 K.
GHe Flow control

- Found too sluggish with oscillations and overshoots when disturb by pressure changes
- Three auto-tuning methods tested: Relay, SIMC, IFT

<table>
<thead>
<tr>
<th>Tuning</th>
<th>$K_c$</th>
<th>$T_i$</th>
<th>Overshoot</th>
<th>Oscillation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1</td>
<td>200</td>
<td>22 %</td>
<td>6 %</td>
</tr>
<tr>
<td>Relay</td>
<td>9</td>
<td>12</td>
<td>0.5 %</td>
<td>1 %</td>
</tr>
<tr>
<td>SIMC</td>
<td>4.5</td>
<td>7</td>
<td>1 %</td>
<td>1.3 %</td>
</tr>
<tr>
<td>IFT</td>
<td>13</td>
<td>11</td>
<td>0.5 %</td>
<td>0.5 %</td>
</tr>
</tbody>
</table>
Experimental results (HVAC process)

- Use case: chilled water production unit providing water at 5°C for LHC
- Maintain at 25°C the condenser output temperature of a chiller
- PI controller: desired temperature deviation within 1°C
HVAC control

Operation team reported instabilities on a regulation loop action on a control valve

Results: Relay vs. IFT

| Tuning | $K_c$ | $T_i$ | $|\Delta y|$ | $|\Delta u|$ |
|--------|-------|-------|-------------|------------|
| Original | 8    | 0.5   | 2 °C        | 100 %      |
| Relay   | 8.3   | 311   | 1.3 °C      | 10 %       |
| IFT     | 50.9  | 5666  | 0.5 °C      | 20 %       |
Conclusions

- PID auto-tuning is not a dream. **Feasible** to implement
- Fully-integrated implementation: UNICOS
- **Flexible** solution: Open to new methods
- Improvement of plant availability and *engineering time*
- **Operator acceptance**
Acknowledgements

Collaboration agreement: CERN – UVA

Industrial Controls group
Engineering Department
CERN, Geneva (Switzerland)

Automatic control and systems engineering
University of Valladolid.
UVA, Valladolid (Spain)
UNICOS (UNified Industrial Control System) is a CERN-made framework to develop industrial control applications
http://www.cern.ch/unicos

Enrique Blanco: automation engineer, PhD in systems and process engineering. Head of the process control section (industrial controls group) in the engineering dpt. at CERN
Native integration advantages

**Easy integration**
- Avoid data extraction & third party tools (off line analysis)
- External connections
- Customized to our environment (event driven data, Customized scaling)

**Ease operation**
- Same philosophy & look and feel
- Ease parameterization
- Fully control and safe operation (boundaries)

**Flexible and evolutive**
- Easy integration of new algorithms