SIMULAÇÃO DE CONTROLE DE PROCESSOS INDUSTRIAIS REFERENCIAIS PELA WEB: UMA FERRAMENTA FOSS

Rafael G. Martins – rafael@rafaelmartins.eng.br
Bruno R. Carneiro – brunorc@live.com
Thales C. Lanza – thaleslanza@gmail.com
Marlon J. Carmo - marloncarmo@ieee.org
Centro Federal de Educação Tecnológica de Minas Gerais – Campus III
Rua José Peres 558 - Centro
36700-000 – Leopoldina - MG

Resumo: *Os controladores PID têm sido utilizados em larga escala na indústria e nas universidades, por serem na maioria das vezes o recurso mais viável para controle de certas plantas de processos. Devido à limitação do conhecimento sobre os métodos de identificação e sintonia destes controladores, este trabalho apresenta um ambiente alternativo multifuncional, que utiliza linguagem Python, plataforma WEB (Internet), para simulação de controladores PID com base em catorze processos referenciais, capazes de representar uma gama extensa dos processos reais existentes na indústria atual.*

Palavras-chave: *Controladores PID, sintonia, malhas industriais, simulação.*

1. INTRODUÇÃO

Os controladores Proporcional Integral (PI) e Proporcional Integral Derivativo (PID) são utilizados em grande parte das malhas industriais por serem capazes de compensar a maioria dos processos práticos industriais (O'DWYER, 2006), garantindo a estabilidade da operação. A maioria dos controladores utilizados na indústria são do tipo PID (ÅSTRÖM & HÄNGGLUND, 2005).

Entre os vários métodos de sintonia para controladores PID, destacam-se os tradicionais baseados na resposta ao degrau: Ziegler e Nichols (ZIEGLER & NICHOLS, 1942), Cohen-Coon (COHEN & COON, 1953) e López (LÓPEZ, 1967), que continuam sendo referência para o estudo de controladores PID, mesmo com o desenvolvimento de tecnologias mais avançadas para tal finalidade.

Para obtenção do modelo dinâmico de um sistema, os métodos mais tradicionais são a resposta ao degrau e o método do ganho crítico, dos quais se obtém um modelo de primeira ordem contendo três parâmetros: ganho estático do processo, tempo de atraso e constante de tempo do sistema (CARMO, 2006).

Devido à evolução dos controladores, torna-se necessário a constante atualização dos

profissionais, principalmente em relação aos métodos de identificação e sintonia. Professores de engenharia de controle precisam complementar as aulas teóricas inserindo uma abordagem mais prática e proporcionando aos alunos conhecimento para lidar com situações de problema em plantas industriais.

Com base nessa necessidade, esse trabalho apresenta uma ferramenta didática desenvolvida para simulação de controladores PID, utilizando software livre, visto que uma das limitações na abordagem prática desse estudo é devido ao fato de a maioria dos softwares disponíveis para este fim serem proprietários, com alto custo de aplicação.

Trata-se de um ambiente multifuncional que utiliza linguagem Python (MILLMAN & AIVAZIS, 2011), plataforma WEB (Internet) e tem como base para simulação catorze processos referenciais, que representam uma grande parte dos processos reais utilizados no meio industrial.

O presente trabalho está dividido da seguinte forma:

Na seção 2 apresenta-se os processos e métodos utilizados no software.

Na seção 3 é feita uma descrição do software.

Na seção 4 apresentam-se os resultados obtidos com o uso desta ferramenta.

A seção 5 relata as conclusões obtidas com o trabalho e algumas ideias do que se pode ser feito para aperfeiçoá-lo.

2. PROCESSOS E MÉTODOS UTILIZADOS NO SOFTWARE

2.1. Processos Referenciais Industriais

Como base para o software foram utilizados catorze processos referenciais que conseguem se aproximar de muitos dos processos reais presentes nas malhas industriais. Esses processos são apresentados a seguir.

- Processo de primeira ordem;
- Processo de segunda ordem;
- Processo de segunda ordem de fase não-mínima;
- Processo de terceira ordem com tempo morto ajustável;
- Processo de polos múltiplos e iguais;
- Processo de quarta ordem;
- Processo com três polos iguais e um zero no semiplano direito;
- Processo de primeira ordem com tempo morto;
- Processo de segunda ordem com tempo morto;
- Processo com características dinâmicas assimétricas;
- Processo condicionalmente estável;
- Processo oscilatório;
- Processo instável;
- Processo de primeira ordem mais tempo morto com a presença de integrador;

Cabe ressaltar que nem todos os processos conseguem ser identificados e controlados a partir dos métodos utilizados neste trabalho, visto que se trata de um projeto didático. A análise da eficiência dos métodos do software também faz parte do aprendizado do aluno.

2.2. Métodos de identificação e sintonia para controladores PID baseados no modelo FODT

No projeto apresentado neste trabalho foram utilizados métodos de identificação e sintonia baseados no modelo FODT, que serão apresentados nesta seção.

O modelo First Order Delay Time (FODT) trata-se de um modelo aproximado, utilizado para identificação de plantas em malha aberta. Este possui três parâmetros: ganho do sistema K, a constante de tempo τ (tau) e o tempo morto L, que são aplicados na Equação (1).

$$G(s) = \frac{K}{1 + s\tau} e^{-sL} \qquad (1)$$

Pelo fato de se basear no tempo morto, esse modelo não é capaz de identificar sistemas sem tempo morto, ou com tempo morto muito pequeno.

Métodos baseados na curva de reação para identificação de processos

Geralmente os métodos de identificação baseados na curva de reação de processos utilizando-se o modelo FODT são baseados em dois pontos (p1 e p2 na figura 1), que correspondem a certos níveis alcançados pela curva do gráfico da resposta ao degrau em malha aberta em relação à amplitude de estabilização do sistema. Esses níveis dependem do método de identificação utilizado.

Como pode ser visto na Figura 1, os dois pontos definem uma reta. O tempo morto L é definido pela distância entre o ponto onde a reta toca o eixo do tempo e a sua origem.

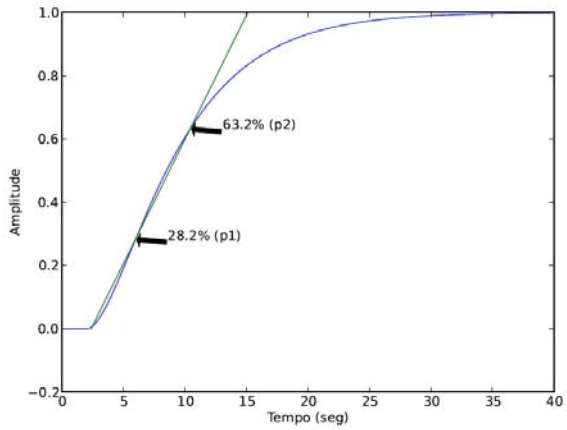

Figura 1 - Modelo FODT – Método de Smith – Processo de terceira ordem com tempo morto ajustável.

O valor de τ é a constante de tempo, que é definida pela medida da projeção do extremo da reta, alinhado com o ponto máximo da curva, até o ponto onde a reta toca o eixo do tempo.

A Tabela 1 apresenta os dois pontos utilizados por cada método de identificação.

Tabela 1 - Constantes para os métodos de identificação.

Método	%p1(t1)	%p2(t2)
Alfaro	25.0	75.0
Broida	28.0	40.0
Chen & Yang	33.0	67.0
Ho *et al.*	35.0	85.0
Smith	28.3	63.2
Vitecková *et al.*	33.0	70.0

O ganho estático, o parâmetro k, é representado pela amplitude da curva quando em regime permanente, que é geralmente representada pela amplitude do último ponto da curva nos gráficos.

Os valores de τ e L podem ser obtidos a partir das Equações 2 e 3, respectivamente, utilizando os valores de t1 e t2, que são os tempos em que a curva atinge os pontos p1 e p2 de cada método.

$$\tau = 1.5(t2 - t1) \qquad (2)$$

$$L = (t2 - \tau) \qquad (3)$$

Os valores de τ, L e K são utilizados nos métodos de sintonia para calcular os ganhos do controlador desejado.

Métodos de sintonia para controladores PID utilizando curva de reação

No software apresentado neste trabalho foram implementados quatro métodos de sintonia para controladores PID que utilizam a curva de reação dos sistemas. Estes métodos e as equações para cálculo de seus parâmetros estão descritos nas tabelas a seguir.

Tabela 2 - Método de Ziegler-Nichols.

Controlador	Fórmulas		
P	$Kp = \frac{\tau}{kL}$		
PI	$Kp = 0.9\frac{\tau}{kL}$	$Ti = 3.33L$	
PID	$Kp = 1.2\frac{\tau}{kL}$	$Ti = 2L$	$Td = \frac{L}{2}$

Tabela 3 - Método de Cohen-Coon.

Controlador	Fórmulas		
P	$Kp = \frac{\tau}{kL\left(1+\frac{R}{3}\right)}$		
PI	$Kp = \frac{\tau}{kL\left(\frac{10}{9}+\frac{R}{12}\right)}$	$Ti = L\left(\frac{30+3R}{9+20R}\right)$	
PD	$Kp = \frac{\tau}{kL\left(\frac{5}{4}+\frac{R}{6}\right)}$		$Td = L\left(\frac{6-2R}{22+3R}\right)$
PID	$Kp = \frac{\tau}{kL\left(\frac{4}{3}+\frac{R}{4}\right)}$	$Ti = L\left(\frac{32+6R}{13+8R}\right)$	$Td = L\left(\frac{4}{13+8R}\right)$

Onde

$$R = \frac{L}{\tau} \tag{4}$$

Tabela 4 - Método Chien, Hrones e Reswick 0%.

Controlador	Fórmulas		
P	$Kp = 0.3\frac{\tau}{kL}$		
PI	$Kp = 0.35\frac{\tau}{kL}$	$Ti = 1.2\tau$	
PID	$Kp = 0.6\frac{\tau}{kL}$	$Ti = \tau$	$Td = 0.5L$

Tabela 5 - Método Chien, Hrones e Reswick 20%.

Controlador	Fórmulas
P	$Kp = 0.7\frac{\tau}{kL}$

PI	$Kp = 0.6\frac{\tau}{kL}$	$Ti = \tau$	
PID	$Kp = 0.95\frac{\tau}{kL}$	$Ti = 1.4\tau$	$Td = 0.47L$

3. DESCRIÇÃO DO SOFTWARE

O trabalho apresentado trata-se de um software, desenvolvido em linguagem Python e disponível via World Wide Web (WWW). Esse software, que recebeu o nome de PIDSIM, conta com catorze processos referenciais industriais, já apresentados no capítulo 2, assim como vários métodos numéricos para modelagem e identificação de sistemas e métodos de sintonia de controladores PID (que também foram apresentados no capítulo 2.

Possuindo fins didáticos e com o objetivo de facilitar os estudos em disciplinas de controle, o software começou a ser desenvolvido em 2009 e passou por várias modificações. Todos os recurso utilizados são livres e estão disponíveis na internet, assim como o código-fonte dos módulos que formam o software.

Existe uma instância pública do software funcionando no endereço:

http://pidsim.rafaelmartins.eng.br

O código-fonte do software está disponível em repositórios do Sistema de Controle de Versão Mercurial (MACKALL, 2011)

http://hg.rafaelmartins.eng.br/pidsim/

Este software está licenciado sob a licença GNU General Public License, version 2 (GLP-2, 1991), e pode ser redistribuído livremente, de acordo com os termos da licença.

3.1. A linguagem Python

A linguagem Python é uma linguagem geral, orientada a objetos, que é utilizada em grande escala e para os mais diversos fins, seja para a criação de um web site ou uma aplicação para desktop, ou para a criação de um aplicativo científico com cálculos complexos.

3.2. Arquitetura do software

O PIDSIM tem como uma de suas principais características, justamente por ser um software livre, a possibilidade de usuários trabalharem em seu código-fonte e implementarem novas funcionalidades. Essa tarefa é facilitada quando o software é bem planejado e com uma divisão sistemática de funcionalidades entre as classes e pacotes que o compõem.

Com o objetivo de facilitar a expansão por meio de usuários, o software foi construído com uma estrutura modular, dividido em pacotes individuais com o mínimo de interdependência possível entre si. A Figura 2 mostra a estrutura de dependência entre os pacotes.

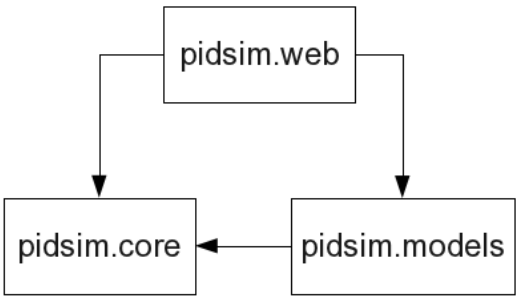

Figura 2 - Dependência entre os pacotes do software.

A divisão foi feita em três pacotes Python:

- pidsim.core
- pidsim.models
- pidsim.web

Cada pacote possui uma funcionalidade especifica e pode ser aprimorado separadamente, o que facilita muito o desenvolvimento distribuído do software. A seguir são apresentados os pacotes e suas funções.

pidsim.core

O pidsim.core é o pacote principal do software, responsável por toda a implementação básica, como os tipos de dados, os métodos numéricos, os algoritmos de controle, entre outras.

Esse pacote foi implementado em Python puro, com o objetivo de se poder utilizá-lo em sistemas onde não é possível a compilação de módulos Python escritos em linguagem C.

pidsim.models

No pacote pidsim.models estão implementados os catorze processos referenciais industriais discutidos anteriormente no item 2.1.

Cada processo foi implementado como uma classe Python, sendo que cada uma possui atributos específicos. Dentre esses estão a expressão Latex que representa o processo, um método Python que representa a função de transferência do processo, uma lista de argumentos que esta função recebe, entre outros.

pidsim.web

Este pacote implementa a interface web, desenvolvida utilizando-se o framework web Flask, responsável pela geração das páginas WEB do software e o toolkit de plotagem de gráficos Matplotlib, responsável pela geração dos gráficos.

3.3. A interface WEB

No momento a única interface disponível entre o software e o usuário é feita através da internet, utilizando-se um aplicativo Web, desenvolvido com o framework web Flask.

A Figura 3 ilustra a tela inicial do software.

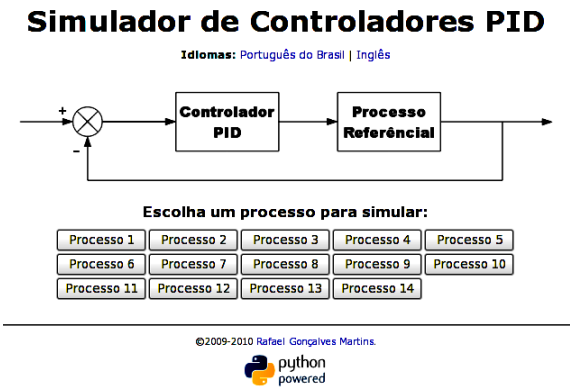

Figura 3 - Tela inicial do software.

Atualmente estão implementados os suportes ao Inglês e ao Português.

4. RESULTADOS

Como resultados do trabalho desenvolvido, são apresentados exemplos de simulações e cálculos feitos pelo software.

A Figura 4 a seguir ilustra a tela inicial do software quando selecionado o processo referencial de quarta ordem (processo 6), no qual os seguintes parâmetros foram aplicados:

- Alpha = 0.2;
- Método numérico para discretização: Runge-Kutta de quarta ordem;
- Método de identificação para o controlador: Smith
- Método de sintonia para o controlador: Ziegler Nichols
- Tempo de amostragem = 1e-2s
- Tempo total = 30s

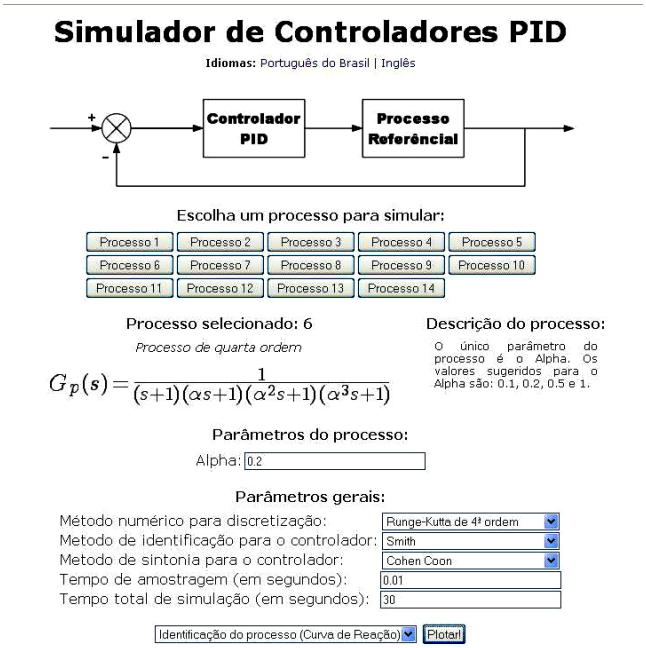

Figura 4 - Tela inicial do software preenchida.

Para obter a curva de identificação do sistema, deve-se clicar no botão “Plotar” após o preenchimento dos parâmetros. A Figura 5 ilustra o gráfico de identificação obtido do software.

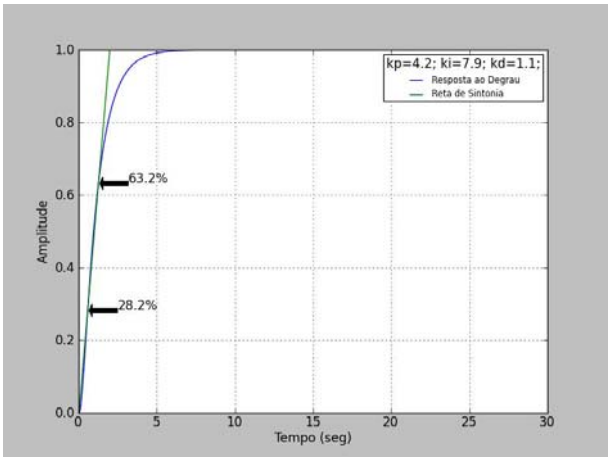

Figura 5 - Gráfico da identificação do processo de exemplo.

De acordo com os parâmetros selecionados, os ganhos calculados para o controlador foram:

kp=4.2;

ki=7.9;

kd=1.1;

É possível também a simulação do controlador em operação, utilizando-se o menu que fica ao lado do botão “Plotar”, escolhendo a opção “Simulação do Controlador PID” e em seguida clicando no botão “Plotar”. A Figura 6 ilustra o grafico resultante dessa simulação para os parâmetros selecionados anteriormente.

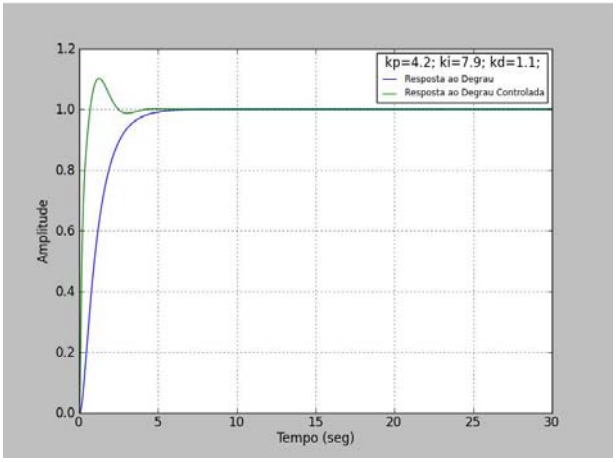

Figura 6 - Gráfico da simulação do controlador para o processo de exemplo.

Selecionando o método de sintonia como “Manual”, é possível alterar manualmente os valores dos ganhos kp, ki e kd.

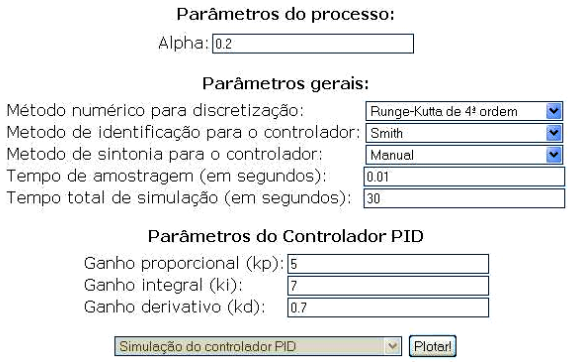

Figura 7 - Valores dos ganhos do controlador para o processo de exemplo, escolhidos manualmente.

Selecionando como ganhos os valores: kp=5.0; ki=7.0 e kd=0.7, se obtém o gráfico ilustrado pela Figura 8 a seguir.

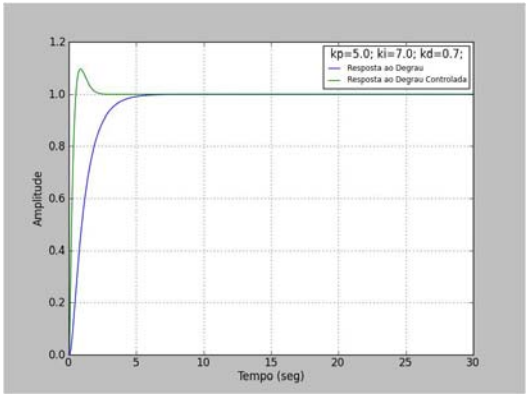

Figura 8 - Gráfico da simulação do controlador para o processo de exemplo, com ganhos escolhidos manualmente.

A partir dos gráficos obtidos, o usuário poderá analisar o desempenho do controlador para o processo referencial selecionado e, de acordo com os ganhos e métodos escolhidos, obter um controlador que melhor corresponda ao resultado desejado.

5. CONSIDERAÇÕES FINAIS

A ferramenta desenvolvida que foi apresentada neste trabalho é capaz de proporcionar várias formas de aprendizado a alunos e outras pessoas envolvidas com controle, seja mostrando os resultados obtidos a partir do software com métodos que funcionam, ou provocando a curiosidade e vontade de desenvolver soluções para os métodos que precisam ser aperfeiçoados.

O software PIDSIM pode ser aprimorado em diversas áreas, recebendo, por exemplo, uma nova interface gráfica, novos métodos de identificação e sintonia e até mesmo implementações de novos idiomas além do português e do inglês, entre outros. Essa vantagem existe devido ao fato do software possuir uma arquitetura bem definida, e também por ser bem simples e fácil de utilizar.

O desenvolvimento de um software livre com essa finalidade contribui para um avanço mais rápido nessa área, permitindo que várias pessoas colaborem facilmente.

Apesar do uso de software livre ter crescido nos últimos tempos, a quantidade desse tipo de ferramenta sendo utilizada na área de controle é pequena. Porém, existem ferramentas importantes desse tipo, indicando que novas idéias podem surgir para diversas aplicações.

Por ser desenvolvido com uma arquitetura modular, esse trabalho pode ser aprimorado de várias formas, sendo adicionados novos métodos de identificação, novos métodos de sintonia de controladores, criadas novas interfaces para Desktop utilizando Qt ou GTK, entre outras.

Agradecimentos

Os autores agradecem ao MEC/SESu, FNDE, CAPES, CnPq, COPPE/LASUP/UFRJ, FAPEMIG, FAPERJ Fundação CEFETMINAS e CEFET-MG pelo apoio ao desenvolvimento deste trabalho.

6. REFERÊNCIAS / CITAÇÕES

ÅSTRÖM, K. J. & HÄNGGLUND, T. (2005), PID controllers: Theory, Design and Tuning. 2nd. ed. North Carolina: Instrument Society of America.

CARMO, M. J. do. Ambiente educacional multifuncional integrado para sintonia e avaliação do desempenho de malhas industriais de controle. 2006.

COHEN, G. H.; COON, G. A. Theoretical considerations of retarded control. ASME Transactions, v. 75, p. 827–834, 1953.

FREE SOFTWARE FOUNDATION. GNU General Public License, version 2. 1991.

LÓPEZ, A. M., Miller, J. A., Smith, C. L. and Murril, P. W. (1967), Tuning controllers with error-integral critteria. In: Instrumentation Technology.

MACKALL, M. Mercurial. 2011

MILLMAN, K. J.; AIVAZIS, M. Python for scientists and engineers. Computing in Science Engineering, v. 13, n. 2, p. 9 –12, march-april 2011. ISSN 1521-9615.

O'DWYER, A. Handbook of PI and PID Controller Tuning Rules. 2nd. ed. [S.l.]: Imperial College Press, 2006.

ZIEGLER, J. B. and NICHOLS, N. B. (1942), Optimum settings for automatic controls. ASME Transactions, v. 64, p. 759–768.

SIMULATION OF INDUSTRIAL PROCESS CONTROL BY REFERENCE WEB: A TOOL FOSS

Abstract: *The PID controllers have been used extensively in industry and in universities, because they most often use the more feasible for control of various process plants. Due to limited knowledge about the methods of identification and tuning of these controllers, This work presents an alternative environment multifunctional, which uses the Python language, platform WEB (Internet), for simulation of PID controllers based on fourteen benchmark processes, capable of representing a wide range of real industrial processes today.*

Key-words: *PID Controllers, Tuning, Industrial Loops, Simulation*