

**INSTITUTO FEDERAL DE SANTA CATARINA**

**CLEITON FERIGOLLO**

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA SINTONIA  
AUTOMÁTICA DE CONTROLADORES PID EM SISTEMA  
EMBARCADO APLICADO EM CLP**

**FLORIANÓPOLIS, SC.**

**FEVEREIRO DE 2017.**

**CLEITON FERIGOLLO**

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA SINTONIA  
AUTOMÁTICA DE CONTROLADORES PID EM SISTEMA  
EMBARCADO APLICADO EM CLP**

Dissertação apresentada ao Mestrado Profissional em Mecatrônica do Instituto Federal de Santa Catarina para obtenção do diploma de Mestre em Mecatrônica.

Orientadora: Cynthia Beatriz Scheffer Dutra

**FLORIANÓPOLIS**

**FEVEREIRO DE 2017.**

CDD 629.895  
F356d

Ferigollo, Cleiton

Desenvolvimento de uma ferramenta para sintonia automática de controladores PID em sistema embarcado aplicado em CLP [DIS] / Cleiton Ferigollo; orientação de Cynthia Beatriz Scheffer Dutra – Florianópolis, 2017.

1 v.: il.

Dissertação de Mestrado (Mecatrônica) – Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Inclui referências.

1. Processos de controle computadorizados. 2. Controladores PID. 3. Controlador lógico programável. 4. Sintonia automática de PID. I. Dutra, Cynthia Beatriz Scheffer. II. Título.

Sistema de Bibliotecas Integradas do IFSC

Biblioteca Dr. Hercílio Luz – Campus Florianópolis


Catalogado por: Ana Paula F. Rodrigues Pacheco CRB 14/1117

**DESENVOLVIMENTO DE UMA FERRAMENTA  
PARA SINTONIA AUTOMÁTICA DE CONTROLADORES PID  
EM SISTEMA EMBARCADO APLICADO EM CLP**

**CLEITON FERIGOLLO**

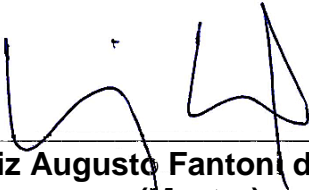
Este trabalho foi julgado adequado para obtenção do título de Mestre em Mecatrônica e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Florianópolis, 10 de março de 2017.

  
Cynthia Beatriz Scheffer Dutra  
(Doutora)

  
André Bittencourt Leal  
(Doutor)

  
Maurício Edgar Stivanello  
(Doutor)

  
Luiz Augusto Fantoni de Souza  
(Mestre)

## RESUMO

Atualmente existem diversas metodologias para obtenção dos parâmetros de um controlador PID de forma automática. Dentre os modernos métodos de sintonia automática (ou *auto-tuning*), destaca-se o dos relés por sua robustez e simplicidade de aplicação. Este método possui grande aceitação na área industrial e é estudado neste trabalho e implementado no PLC300, Controlador Lógico Programável (CLP) da empresa WEG, suprimindo uma atual necessidade dos clientes. A implementação é feita em linguagem C, em um processador Cortex-M3, e encapsulada em um bloco de linguagem *Ladder* que é utilizado para programação do CLP.

Após o ensaio do relé, regras como a de Tyreus-Luyben, Ciancone-Marlin e ITAE Mínimo são utilizadas para obtenção dos parâmetros do controlador PI ou PID para o sistema. A escolha de qual regra é aplicada pode ser feita de forma automática, com base na relação entre o atraso de transporte e a constante de tempo do sistema, ou de forma manual na qual o usuário opta por uma das regras implementadas.

Ao final são apresentados os resultados da aplicação da ferramenta em nove plantas de diversas ordens simuladas em software Matlab, algumas sem atraso de transporte, outras com atraso de transporte. Além das simulações, três plantas reais são utilizadas para a aplicação da ferramenta desenvolvida, sendo elas: controle de posição da abertura de uma válvula de uma turbina hidroelétrica, controle de velocidade de um servomotor e controle de temperatura de uma planta térmica.

Os resultados obtidos mostraram-se satisfatórios tanto para as plantas simuladas quanto para as plantas reais. As plantas simuladas de menor complexidade, primeira ordem e sem atraso de transporte, apresentaram respostas com pequeno sobressinal e pouca oscilação, atingindo o valor de regime rapidamente. As de ordens elevadas e com atraso de transporte, apresentaram respostas consideradas lentas e com sobressinal.

As plantas reais em que a sintonia automática foi aplicada mostraram que o sistema funciona bem na prática, onde há influência de diversos fatores externos, como ruídos, variações na temperatura ambiente, atritos, não linearidade, etc. Em todos os casos, ao menos uma das regras trouxe resultados estáveis para o

sistema, fornecendo um bom ponto de partida para um ajuste mais fino manualmente.

**Palavras-chave:** Controladores PID. Sintonia Automática de PID. Método dos Relés. Controlador Lógico Programável.

## ABSTRACT

There are now a several methodologies to obtain the parameters of a PID controller automatically. Amongst the modern methods of auto-tuning, the relay's method stands out by its robustness and simplicity of application. This method has great acceptance in the industrial area and is studied in this work and implemented in the PLC300, WEG company PLC, supplying a current requirement of the customers. The implementation is made in C language, on a Cortex-M3 processor, and encapsulated in a *Ladder* language block that is used for PLC programming.

After the relay test, rules such as Tyreus-Luyben, Ciancone-Marlin and ITAE minimum are used to obtain PI or PID controller parameters for the system. The choice of which rule is applied can be made automatically, based on the relationship between the dead time and the time constant, or manually allowing the user to choose which rule to apply.

Finally, the results of the application of the tool are presented in nine plants of several orders simulated in Matlab software, some without dead time and others with long dead time. Besides the virtual ones, three real plants are used for the application of the developed tool, these related to the position control of the opening of a valve of a hydroelectric turbine, speed control of a servomotor and temperature control of a thermal plant.

The results obtained were satisfactory for both the simulated and the real plants. The simulated plants of lower complexity, first order and without dead time , presented answers with little overshoot and low oscillation, getting the steady state quickly. Those of high orders and with large dead time, presented answers considered slow with overshoot.

The real plants in which the auto-tuning was applied showed that the system works well in practice, where there are influence of several factors, such as noise, variations in ambient temperature, friction, nonlinearity, etc. In all cases, at least one of the rules has brought stable results to the system, providing a good point for a fine adjustment manually.

**Keywords:** PID Controller. PID Auto-Tunig. Relay Method. Programmable Logic Controller.

## LISTA DE FIGURAS

Figura 1 – Diagrama padrão de um sistema de controle em malha fechada..	18
Figura 2 – Diagrama de blocos da estrutura série .....	22
Figura 3 – Diagrama de blocos da estrutura acadêmica.....	23
Figura 4 – Diagrama de blocos do controlador paralelo .....	23
Figura 5 – Estrutura do relé realimentado.....	26
Figura 6 – Função relé ideal .....	27
Figura 7 – Método dos relés aplicado .....	28
Figura 8 – Relé com histerese .....	29
Figura 9 – Etapas da sintonia automática do bloco TCONT_CP .....	35
Figura 10 – Modelo do PLC300 com IHM (A) e sem IHM (B) .....	37
Figura 11 – Detalhe do conector da rede CAN do PLC300 .....	39
Figura 12 – Bloco do controlador PID do PLC300 .....	41
Figura 13 – Topologia acadêmica do PID do PLC300 .....	43
Figura 14 – Topologia paralela do PID do PLC300 .....	43
Figura 15 – Algoritmo PID de controladores comerciais* .....	44
Figura 16 – Bloco PID2 desenvolvido .....	45
Figura 17 – Fluxograma da execução da sintonia automática .....	48
Figura 18 – Fluxograma do bloco PID2 .....	49
Figura 19 – Cálculo do controlador .....	50
Figura 20 – Aplicação do método dos relés (a) .....	53
Figura 21 – Aplicação do método dos relés (b) .....	54
Figura 22: bloco PID2 no WPS .....	55
Figura 23 – Função <i>trend</i> do WPS .....	56
Figura 24 – Configuração das telas do PLC300. ....	56
Figura 25 – Diagrama de blocos para simulação das plantas .....	58
Figura 26 – Processo de sintonia da planta G1 .....	60
Figura 27 – Resultado do controle com PID da planta G1 .....	61
Figura 28 – Resultado do controle com PI da planta G1 .....	61
Figura 29 – Processo de sintonia da planta G2. ....	62
Figura 30 – Resultado do controle com PID da planta G2.....	63
Figura 31 – Resultado do controle com PI da planta G2 .....	63



Figura 32 – Processo de sintonia da planta G3 .....	64
Figura 33 – Resultado do controle com PID da planta G3.....	65
Figura 34 – Resultado do controle com PI da planta G3 .....	65
Figura 35 – Processo de sintonia da planta G4 .....	67
Figura 36 – Resultado com controlador PID da planta G4 (ITAE) .....	67
Figura 37 – Resultado com controlador PI da planta G4 (ITAE).....	68
Figura 38– Resultado com controlador PID da planta G4 (Tyreus-Luyben) ...	68
Figura 39 – Resultado com controlador PI da planta G4 (Tyreus-Luyben).....	69
Figura 40 – Processo de sintonia da planta G5 .....	70
Figura 41 – Resultado com controlador PID da planta G5.....	70
Figura 42 – Resultado com controlador PI da planta G5 .....	71
Figura 43 – Processo de sintonia da planta G6 .....	72
Figura 44 – Resultado com controlador PID da planta G6.....	72
Figura 45 – Resultado com controlador PI da planta G6 .....	73
Figura 46 – Processo de sintonia da planta G7 .....	74
Figura 47 – Resultado com controlador PID da planta G7.....	74
Figura 48 – Resultado com controlador PI.....	75
Figura 49 – Processo de sintonia da planta G8 .....	76
Figura 50 – Resultado com PID da planta G8 .....	76
Figura 51 – Resultado com PI da planta G8 .....	77
Figura 52 – Processo de sintonia da planta G9 .....	78
Figura 53 – Resultado com PID da planta G9 .....	78
Figura 54 – Resultado PI da planta G9.....	79
Figura 55 – Turbina Francis com detalhe do pistão que abre a válvula.....	80
Figura 56 – Velocidade de abertura da tampa em função da corrente no atuador .....	81
Figura 57– Curva do relé para a turbina .....	82
Figura 58– Resposta ao degrau de abertura da turbina. ....	83
Figura 59 – Resposta ao degrau de fechamento da turbina .....	83
Figura 60 – Servomotores utilizados para controle de velocidade.....	84
Figura 61 – Curva do relé para o servomotor .....	85
Figura 62 – Resposta ao degrau com controlador PID(ITAE).....	86
Figura 63 – Resposta ao degrau com controlador PI (ITAE) .....	86

Figura 64 – Resposta ao degrau com controlador PID (Ciancone-Marlin).....	87
Figura 65 – Resposta ao degrau com controlador PI (Ciancone-Marlin) .....	88
Figura 66 – Resposta ao degrau com controlador PID (Tyreus-Luyben).....	88
Figura 67 – Resposta ao degrau com controlador PI (Tyreus-Luyben) .....	89
Figura 68 – Planta térmica.....	89
Figura 69 – Curva do relé para a planta térmica.....	90
Figura 70 – Resposta ao degrau positivo com controlador PID (Tyreus-Luyben) .....	91
Figura 71 – Resposta ao degrau negativo com controlador PID (Tyreus-Luyben) .....	91

## LISTA DE TABELAS

Tabela 1 – Regra de Ziegler-Nichols para controlador PID estrutura série ....	30
Tabela 2 – Regra de Tyreus-Luyben para PID estrutura série .....	31
Tabela 3 – Regra de Ciancone-Marlin para PID na estrutura série .....	31
Tabela 4 – Regra de ITAE Mínimo para PID estrutura ideal.....	31
Tabela 5 – Regra de Tyreus-Luyben para PI/PID.....	33
Tabela 6 – Regra de Ciancone-Marlin para PI/PID.....	33
Tabela 7 – Parâmetros do bloco PID do PLC300 .....	42
Tabela 8 – Opções da sintonia automática.....	46
Tabela 9 – Status da sintonia .....	47
Tabela 10 – Resultados da planta G1 para controlador Tyreus-Luyben.....	60
Tabela 11 – Resultados da planta G2 para controlador ITAE.....	62
Tabela 12 – Resultados da planta G3 para controlador Ciancone-Marlin.....	64
Tabela 13 – Resultados da planta G4 para controlador ITAE.....	66
Tabela 14 – Resultados da planta G4 para controlador Tyreus-Luyben.....	66
Tabela 15 – Resultados da planta G5 para controlador ITAE.....	69
Tabela 16 – Resultados da planta G6 para controlador Ciancone-Marlin.....	71
Tabela 17 – Resultados da planta G7 para controlador ITAE.....	73
Tabela 18 – Resultados da planta G8 para controlador Ciancone-Marlin.....	75
Tabela 19 – Resultados da planta G9 para controlador Ciancone-Marlin.....	77
Tabela 20 – Resultados da turbina para controlador Tyreus-Luyben .....	81
Tabela 21 – Resultados do servomotor para controlador ITAE .....	85
Tabela 22 – Resultados do servomotor para controlador Tyreus-Luyben .....	87
Tabela 23 – Resultados do servomotor para controlador Ciancone-Marlin ....	87
Tabela 24 – Resultados da planta térmica para controlador Tyreus-Luyben..	90

## LISTA DE ABREVIATURAS E SIGLAS

<i>PID</i>	Controlador proporcional integral derivativo
<i>PI</i>	Controlador proporcional integral
<i>P</i>	Controlador proporcional
<i>I</i>	Controlador integral
<i>D</i>	Controlador derivativo
<i>CLP</i>	Controlador lógico programável
$K_p$	Ganho proporcional
$K_i$	Ganho integral
$K_d$	Ganho derivativo
<i>IHM</i>	Interface homem máquina
<i>WPS</i>	<i>WEG Programming Suite</i>
$e(t)$	Erro
$r(t)$	Referência
$u(t)$	Sinal de controle
$y(t)$	Variável de processo
$T_i$	Tempo integral
$T_d$	Tempo derivativo
<i>ISA</i>	<i>International Society of Automation</i>
$K_u$	Ganho crítico
$P_u$	Período crítico
$W_u$	Frequência crítica
<i>D</i>	Atraso de transporte
<i>h</i>	Amplitude do relé
<i>a</i>	Amplitude da saída do processo
$\varepsilon$	Histerese do relé
<i>SP</i>	<i>Setpoint</i>
<i>MV</i>	Variável manipulada
$\tau$	Constante de tempo
$C_p$	Fator de controlabilidade
<i>ITAE</i>	<i>Integrated of Time Multiplied by Absolute Error</i>
<i>k</i>	Ganho estático

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Objetivo principal	15
1.2	Objetivos específicos	15
1.3	Justificativa e relevância	16
1.4	Abrangência e delimitação do escopo	17
<b>2</b>	<b>CONTROLADORES PID</b>	<b>18</b>
2.1	Controlador proporcional (P)	19
2.2	Controlador integral (I)	19
2.3	Controlador derivativo (D)	20
2.4	Controlador proporcional integral derivativo (PID)	20
2.5	Classificação dos controladores PID	21
2.5.1	Estrutura série	21
2.5.2	Estrutura acadêmica	22
2.5.3	Estrutura paralela	23
<b>3</b>	<b>SINTONIA AUTOMÁTICA DE CONTROLADORES PID</b>	<b>25</b>
3.1	Método dos relés	25
3.1.1	Relé ideal:	27
3.1.2	Relé com histerese	28
3.2	Regras para obtenção dos parâmetros do controlador	29
3.3	Métodos utilizados por diferentes fabricantes	33
3.3.1	ABB	33
3.3.2	Emerson Process Management	33
3.3.3	Siemens	34
3.3.4	WEG	35
<b>4</b>	<b>O PLC300</b>	<b>36</b>
4.1	O microcontrolador do PLC300 e sua programação	36
4.2	Principais características do PLC300	37
4.3	Saídas digitais, analógicas e PWM	38
4.4	Entradas digitais e analógicas	38
4.5	Entrada de encoder	38
4.6	Módulos de expansão (MANUAL PLC300, p. 2-3)	38
4.7	Interfaces de comunicação	39

4.8	Alimentação .....	40
4.9	A programação do PLC300 .....	40
4.10	O bloco PID .....	41
<b>5</b>	<b>PID2: CONTROLADOR PID DO PLC300 COM SINTONIA AUTOMÁTICA....</b>	<b>45</b>
5.1	O bloco PID2 visto pelo usuário.....	45
5.1.1	A variável Tune .....	46
5.1.2	As variáveis Delta e Hysteresis .....	46
5.1.3	A variável TuneTimeOut.....	47
5.1.4	A variável TuneStatus .....	47
5.1.5	As variáveis KpOut, KiOut e KdOut.....	47
5.2	A implementação do bloco PID2 no sistema embarcado .....	48
5.3	A chamada do bloco PID2 .....	48
5.4	A interrupção de 100us.....	51
5.4.1	O ajuste de Bias .....	51
5.4.2	O ganho estático .....	52
5.4.3	O ponto crítico de operação ( $K_u$ , $P_u$ e $W_u$ ).....	52
5.4.4	A constante de tempo do sistema .....	52
5.4.5	O fator de controlabilidade .....	52
5.4.6	O atraso de transporte.....	53
5.5	Procedimento de sintonia automática do controlador .....	55
5.5.1	Critérios para que a sintonia seja efetuada .....	57
5.5.2	Passo a passo para executar a sintonia.....	57
<b>6</b>	<b>RESULTADOS .....</b>	<b>58</b>
6.1	Plantas simuladas.....	58
6.1.1	Primeira ordem com $C_p < 0,1$ .....	59
6.1.2	Primeira ordem com $0,1 \leq C_p \leq 1$ .....	62
6.1.3	Primeira ordem com $1 < C_p$ : .....	64
6.1.4	Segunda ordem com $C_p < 0,1$ :.....	66
6.1.5	Segunda ordem com $0,1 \leq C_p \leq 1$ :.....	69
6.1.6	Segunda ordem com $1 < C_p$ : .....	71
6.1.7	Planta G7 .....	73
6.1.8	Terceira ordem com $1 < C_p$ :.....	75
6.1.9	Planta de oitava ordem.....	77

6.2	Testes práticos com plantas reais .....	79
6.2.1	Turbina hidroelétrica Francis .....	79
6.2.2	Servomotor .....	84
6.2.3	Planta Térmica .....	89
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>93</b>
<b>8</b>	<b>TRABALHOS FUTUROS .....</b>	<b>95</b>
<b>9</b>	<b>REFERÊNCIAS .....</b>	<b>96</b>
<b>10</b>	<b>BIBLIOGRAFIA CONSULTADA .....</b>	<b>99</b>

## 1 INTRODUÇÃO

Controladores PID estão presentes na grande maioria das aplicações de controle de processos existentes no mercado. Eles são a solução mais comum para problemas práticos de controle (ASTRÖM; HÄGGLUND, 2006). A sintonia dos parâmetros do controlador é uma tarefa muitas vezes árdua e demorada, além disso, é sabido que grande parte desses controladores desempenham seu papel no meio industrial de forma muito inferior àquela possível de ser obtida com uma regulação ótima das variáveis.

A eficácia do sistema de controle depende de uma boa calibração dos ganhos  $K_p$ ,  $K_i$  e  $K_d$  que geralmente é feita por uma pessoa com conhecimento na área, através de técnicas conhecidas que, muitas vezes, envolvem bastante tempo e nem sempre chegam a resultados realmente satisfatórios. Boa parte dos controladores na indústria operam em modo manual e muitos dos que operam em modo automático, possuem ação derivativa desligada pelo simples fato de ser difícil de sintonizar adequadamente (ASTRÖM; HÄGGLUND, 2006).

Dentre os diversos métodos para obtenção dos parâmetros do controlador, o método da entrada degrau em malha aberta ainda é um dos mais utilizados (LIU; WANG; HUANG, 2013). Para evitar que a saída do processo se afaste muito do *setpoint*, métodos de malha fechada como a da entrada degrau e dos relés vêm sendo utilizados para manter a saída dentro de limites admissíveis (LIU, WANG, HUANG, 2013).

Diversas técnicas de sintonia têm sido propostas na literatura, possibilitando que ajustes mais finos e precisos nas variáveis do controlador sejam feitos e implementados no próprio local de aplicação. Métodos de sintonia automática também têm sido desenvolvidos e alguns dos controladores PID podem possuir recursos de ajuste automático *on-line* (OGATA, 2002).

A sintonia automática é um método em que o controlador PID é obtido automaticamente com pouco esforço do operador que, geralmente, pressiona apenas um botão ou envia um comando ao controlador para ativar a função de sintonia. Esse processo automático deve ser capaz de obter resultados mais satisfatórios do que o processo manual, além de ser de simples aplicação.



Desta forma, foi desenvolvido neste projeto um sistema automatizado de calibração de controles PID. Toda a implementação foi feita em sistema embarcado, em um microprocessador Cortex-M3, utilizando linguagem C. O código desenvolvido foi modularizado, de forma a permitir que seja utilizado posteriormente em outros produtos, não necessitando para isso de muitas adaptações. Diversos produtos hoje comercializados pela WEG, como inversores de frequência, CLPs e servoconversores poderão contar com esta nova funcionalidade que traz melhor eficiência e desempenho em processos industriais em geral.

Um bloco de controle para utilização em linguagem *ladder* será desenvolvido e validado no Controlador Lógico Programável WEG, o PLC300. Com esse bloco, o operador poderá facilmente habilitar a sintonia automática através de uma entrada digital, um botão da IHM ou através de alguma rede de comunicação, obtendo como resultado final os parâmetros do controlador a ser utilizado, eliminando o processo moroso e muitas vezes não eficiente de sintonia manual do controlador.

### **1.1 Objetivo principal**

O objetivo principal deste trabalho é desenvolver um sistema automático de calibração de controladores PID sem a necessidade de conhecer o modelo matemático da planta. Ao final do projeto, pretende-se ter uma funcionalidade de um CLP que permita que os ganhos proporcional, integral e derivativo de um controlador, aplicado a uma determinada planta, sejam calculados automaticamente para que o sistema tenha um comportamento adequado conforme a necessidade da aplicação.

### **1.2 Objetivos específicos**

- Fazer um levantamento das principais necessidades e aplicações de sistemas de controle dos clientes do PLC300;
- Conhecer as formas de sintonia automática já existentes nos principais CLPs do mercado;
- Desenvolver um sistema de sintonia rápido e eficiente, implementado em linguagem C, em sistema embarcado para o PLC300;

- Validar a funcionalidade da ferramenta em plantas de primeira, segunda e superiores ordens simuladas no Matlab.
- Utilizar o sistema desenvolvido em plantas reais de clientes WEG para validar a aplicação prática do mesmo.

### **1.3 Justificativa e relevância**

Nos últimos anos, os requisitos de desempenho dos processos tem se tornado cada vez mais difíceis de satisfazer por conta da competitividade, regulamentações de segurança e de ambiente e também por rápidas mudanças econômicas (SEBORG et al., 2011). Tendo em vista esse enfoque na segurança e eficiência das plantas, é natural que o controle de processos ganhe cada vez mais espaço e importância no mercado. Controladores PID são os mais largamente utilizados na indústria e estão cada vez mais presentes nas mais variadas aplicações. Mesmo os mais complexos sistemas de controle industrial muitas vezes possuem um PID como controle principal (JOHNSON; MORADI, 2005). A cada dia que passa, plantas mais eficientes e com menor tempo de implementação se tornam um grande diferencial na busca por um sistema de controle. Um grande problema é que nem sempre a sintonia manual é uma tarefa fácil e muitas vezes não apresenta resultados satisfatórios, podendo até mesmo apresentar um comportamento pior do que o sistema de malha aberta. A WEG possui, entre outros, o controlador lógico programável PLC300 que, através da linguagem de programação *ladder*, pode implementar um controlador PID e aplicá-lo em um processo industrial qualquer que necessite de controle. O que ocorre hoje é o mencionado acima: boa parte dos usuários possui grande dificuldade para ajustar o controlador PID e muitos não possuem nenhuma noção de como isso é feito. Desta forma, justifica-se o desenvolvimento de um processo automatizado de calibração desses controladores, visando uma melhoria no ajuste das variáveis para propiciar bom desempenho dos processos industriais e que possa ser aplicado por clientes com pouco ou nenhum conhecimento nesta área.

#### 1.4 Abrangência e delimitação do escopo

A grande maioria das plantas em que hoje se aplica controle com produtos desenvolvidos pela WEG são de controle de temperatura, principalmente extrusoras e fornos. Além do controle de temperatura, uma parcela significativa das aplicações é destinada ao controle de pressão de tubulações, velocidade e posição de motores, turbinas, etc. O projeto do mestrado será focado nesses tipos de plantas, uma vez que, segundo a equipe de vendas e aplicações da empresa, eles compreendem cerca de 90% do mercado de controle de processos atingido pela WEG.

Primeiramente, o método dos relés será empregado para obtenção do ganho e do período críticos e de outros parâmetros do sistema. Com as informações obtidas através do método dos relés, regras como as de Tyreus-Luyben, Ciancone-Marlin, ITAE mínimo, entre outras, serão empregadas para obtenção dos parâmetros do controlador PID ( $K_p$ ,  $K_i$  e  $K_d$ ). A regra a ser utilizada para obter os resultados de cada planta será escolhida com base nos estudos feitos por Yu (2007) e por O'Dwyer (2000), que se baseiam na relação entre o atraso de transporte e a constante de tempo do sistema para optar entre uma regra e outra.

Toda implementação será encapsulada em um bloco para ser utilizado em linguagem *ladder* na programação do controlador PLC300. Este bloco poderá facilmente ser parametrizado e habilitado, possibilitando a aplicação da sintonia automática de um controlador PID. O bloco possuirá ainda uma forma fácil de chavear entre o modo de sintonia e o modo de controle, no qual os parâmetros calculados na sintonia automática poderão ser aplicados no sistema.

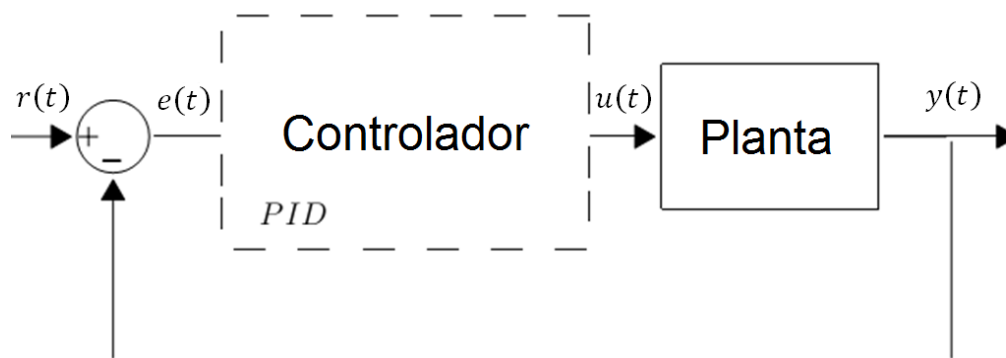
A ferramenta a ser desenvolvida será testada tanto em plantas simuladas no Matlab como também em plantas reais, podendo ser elas de temperatura, velocidade ou posição de motores, pressão, entre outras.

Por fim, após a avaliação dos resultados e validação do bloco, a funcionalidade será disponibilizada para os clientes através de uma nova versão de firmware para o PLC300.

## 2 CONTROLADORES PID

O controlador PID calcula o “erro”, que é a diferença entre a variável controlada e o valor desejado (*setpoint*), e em função dele gera um sinal de controle de forma a eliminar este desvio (PINTO, 2014). O controlador PID é uma técnica de controle que faz uso de ação de controle proporcional, do integral e do derivativo simultaneamente, que serão detalhadas nas subseções seguintes. Ele é amplamente aplicado na indústria e busca obter sistemas cada vez mais eficientes, imunes a ruídos e o mais próximo do ideal possível. Diversas bibliografias afirmam que mais de 90% dos processos industriais que operam em malha fechada, utilizam controladores PID (CARDOSO, 2002). Mesmo com o surgimento de novas técnicas, controladores PID dominarão por muito tempo os parques industriais (PINTO, 2014). A Figura 1 apresenta um diagrama de blocos básico de um sistema realimentado com controlador PID. O sinal de controle é calculado pelo PID em um determinado tempo com base no erro atual  $e(t)$ , na integral do erro e também na derivada do erro (BERNER, 2015).

Figura 1 – Diagrama padrão de um sistema de controle em malha fechada



Fonte: o autor

Na sequência, as três ações de um controlador PID são apresentadas separadamente, para facilitar o entendimento da influência de cada uma delas em um sistema de controle.

## 2.1 Controlador proporcional (P)

O controlador proporcional produz um sinal na saída que é proporcional ao erro do sistema, conforme a equação

$$P(t) = K_p e(t) \quad (1)$$

onde o erro é dado pela diferença entre a referência do usuário e a saída do sistema:  $e(t) = r(t) - y(t)$ .

Um ganho proporcional  $K_p$  elevado faz com que a saída do sistema tenha uma rápida alteração e tende a diminuir o erro de regime permanente. Contudo, um ganho muito grande pode tornar o sistema instável. Já um ganho proporcional  $K_p$  muito baixo torna o sistema muito lento. A ação proporcional por si só não elimina o erro de regime permanente, para isso é necessária a adição da ação integral.

## 2.2 Controlador integral (I)

A ação integral, como o próprio nome diz, é proporcional à integral do erro do sistema, em outras palavras, proporcional à magnitude e duração do erro, conforme equação

$$I(t) = \frac{1}{T_i} \int_0^t [e(t)] dt. \quad (2)$$

O tempo integral  $T_i$  é inversamente proporcional ao ganho integral  $K_i$ , que aparecerá posteriormente na topologia paralela. Se o ganho integral escolhido for baixo, o sistema pode demorar muito tempo para atingir o valor de regime permanente. Porém, um ganho integral elevado demais pode tornar o sistema oscilatório.

A ação integral combinada com a proporcional (PI) é altamente utilizada na indústria. Existem casos típicos de plantas sem controle derivativo, e mesmo assim obtendo resultados bastante satisfatórios (FINKE, 2016). A ação PI fornece uma opção para corrigir o erro de regime permanente não eliminado pela ação proporcional isolada, além disso, acelera a resposta do sistema, permitindo que o valor de referência seja atingido mais rapidamente.

### 2.3 Controlador derivativo (D)

A ação derivativa, basicamente, produz um sinal de saída que é proporcional à taxa de variação do erro, conforme equação

$$D(t) = T_d \frac{d}{dt} [e(t)] \quad (3)$$

A ação derivativa atua em intervalos regulares, conhecido como tempo derivativo  $T_d$ , que é inversamente proporcional ao ganho derivativo  $K_d$  que aparecerá em algumas topologias apresentadas nas subseções seguintes.

A ação derivativa é capaz de fornecer uma ação antecipativa do erro, tornando ele mais preditivo. Ela diminui o tempo de resposta e pode melhorar a estabilidade do sistema.

Ganhos derivativos muito elevados, podem causar oscilações e comportamentos inesperados no sistema.

Segundo Finke (2016), uma das desvantagens da ação derivativa é que na forma ideal ela possui um ganho muito alto para sinais de alta frequência, fazendo com que sinais de ruídos tenham grande influência no sinal de controle.

### 2.4 Controlador proporcional integral derivativo (PID)

Os controladores Proporcionais Integrais Derivativos são aqueles que obedecem à lei de controle

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{d}{dt} e(t) \right], \quad (4)$$

onde

$K_p =$  *Ganho Proporcional*,

$T_i =$  *Tempo Integral*,

$T_d =$  *Tempo Derivativo*.

O controlador PID une as vantagens da velocidade, devido à ação proporcional e derivativa, e de precisão do resultado de regime permanente devido ao integrador. Essas características, tornam os controladores PID bastante flexíveis, podendo atender praticamente qualquer sistema de controle, que via de regra, procurará atender as seguintes especificações (ALCINDO, 2010):

- a. Rapidez de operação;
- b. Amortecimento adequado;
- c. Precisão em regime (erros pequenos ou nulos);
- d. Imunidade a perturbações.

## 2.5 Classificação dos controladores PID

Os controladores PID podem ser divididos em três diferentes estruturas (ÅSTRÖM; HÄGGLUND, 1995, pag. 73):

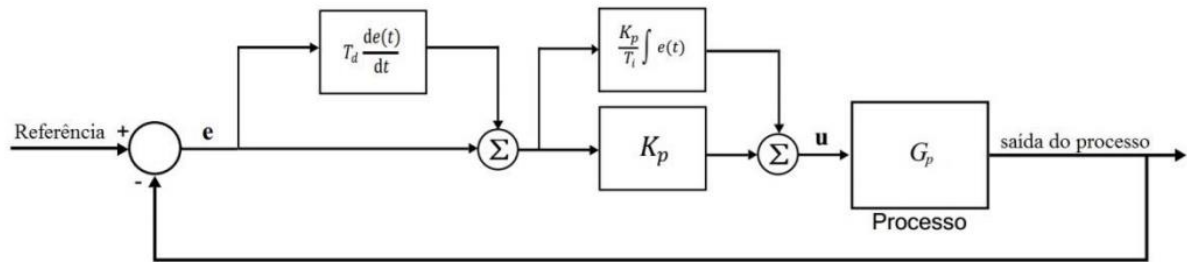
- Estrutura série;
- Estrutura acadêmica;
- Estrutura paralela.

Esta mesma divisão pode ser encontrada nos trabalhos de Boava (2016) e Hermann (2016). Os trabalhos de Yu (2007) e O'Dwyer (2009) trazem a estrutura paralela como sendo apenas uma variação da estrutura Ideal, ficando então duas divisões: série e paralela. Os produtos WEG que podem configurar um sistema de controle possuem tanto a estrutura acadêmica quanto a paralela. Desta forma, optou-se neste trabalho por manter esta divisão proposta por Åström e Hägglund (1995).

### 2.5.1 Estrutura série

A estrutura série também é conhecida como Interativa porque a ação derivativa tem influência sobre a ação integral, conforme pode ser visto nas equações (5), (6) e no diagrama de blocos da Figura 2. Ela é uma implementação muito utilizada nos sistemas analógicos, mantida também em alguns dos sistemas digitais atuais (FINKE, 2016).

Figura 2 – Diagrama de blocos da estrutura série



A equação que rege a estrutura série é a

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right] \left[ 1 + T_d \frac{d}{dt} e(t) \right]. \quad (5)$$

Após a aplicação da transformada de Laplace chega-se à equação:

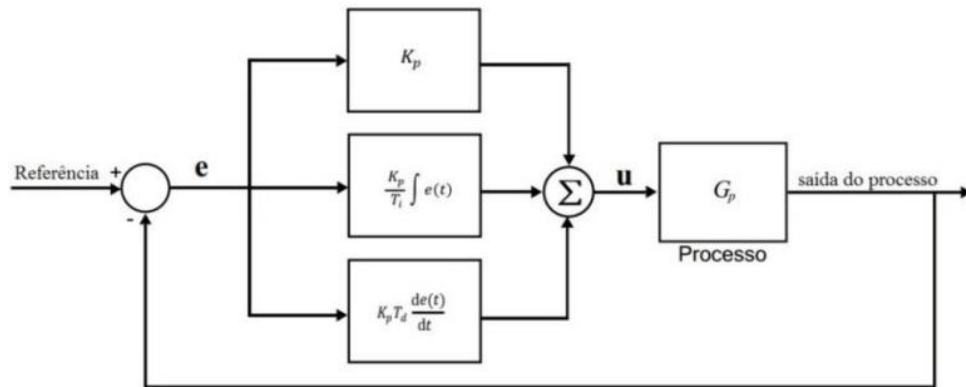
$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{sT_i} \right) (1 + sT_d). \quad (6)$$

### 2.5.2 Estrutura acadêmica

Também conhecida por ideal ou padrão, a estrutura acadêmica é a forma padronizada de implementação da ISA (*International Society of Automation*). Apesar de ser um pouco mais complexa de sintonizar, é a forma mais usada nos controladores digitais segundo Boava (2016). Nesta forma, as ações de controle proporcional, integral e derivativa são combinadas paralelamente, sendo o resultado final da ação de controle, dada pela soma das três componentes calculadas separadamente, conforme Figura 3. Nesta configuração, o ganho  $K_p$  multiplica os termos Integral e Derivativo.



Figura 3 – Diagrama de blocos da estrutura acadêmica



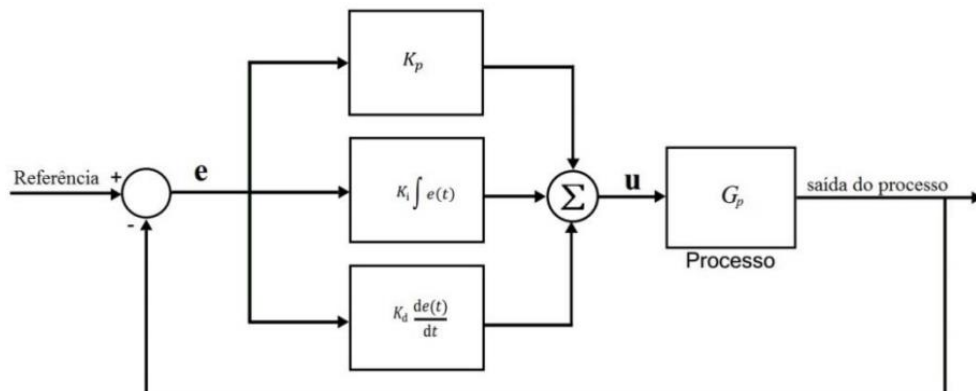
A equação que rege a estrutura acadêmica é a equação (4) já apresentada. Após a aplicação da transformada de Laplace ela apresenta-se assim:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{sT_i} + sT_d \right). \quad (7)$$

### 2.5.3 Estrutura paralela

A estrutura paralela de implementação de um controlador PID pode ser vista na Figura 4:

Figura 4 – Diagrama de blocos do controlador paralelo



Nesta implementação, o ganho proporcional não afeta no tempo integral e nem no derivativo, possibilitando que as três ações de controle ajam completamente separadas. A equação que representa essa implementação depois de aplicada a transformada de Laplace é dada por:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + sK_d. \quad (8)$$

O PLC300, CLP da WEG, pode implementar um controlador P, PI ou PID na estrutura acadêmica ou na paralela. Por este motivo, a ferramenta de sintonia automática foi desenvolvida para estas duas estruturas. Alguns detalhes sobre o PLC300 são fornecidos no capítulo 4.

### 3 SINTONIA AUTOMÁTICA DE CONTROLADORES PID

A sintonia automática implementada neste trabalho determina os ganhos Proporcional, Integral e Derivativo do sistema sem a necessidade de conhecer a modelagem da planta e sem necessitar de muitos esforços do usuário. Um dos métodos mais robustos e simples de sintonia automática de controladores PID, que permite a implementação com as características citadas acima, é conhecido por método dos relés (HANG, ASTRÖM e WANG, 2002). Na sequência será apresentado este método e algumas de suas variações.

#### 3.1 Método dos relés

O método dos relés foi introduzido pelo trabalho de Karl Johan Aström e Tore Hägglund, em 1984. Este método apresenta a característica de manter uma oscilação controlada na etapa de sintonia. O sucesso desse método deve-se à simplicidade dos procedimentos utilizados, e também à sua aplicabilidade em processos lentos ou de características altamente não lineares (NEVES, 2009).

O método dos relés é um método de malha fechada, que permite o cálculo de parâmetros que são utilizados no processo de sintonia do sistema, não necessitando que se tenha conhecimento do modelo matemático da planta em que será aplicado.

Dentre as principais vantagens do método dos relés destacam-se (HANG, ASTRÖM e WANG, 2002), (NEVES, 2009):

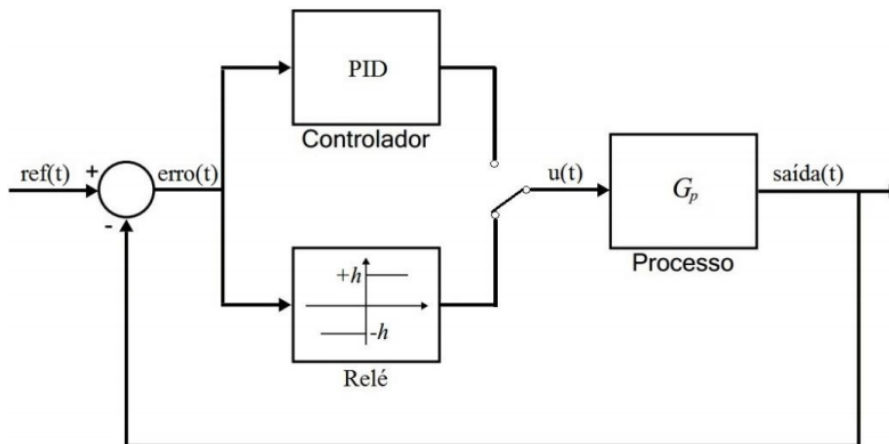
- a) não requer conhecimento prévio do modelo matemático da planta;
- b) o método é facilmente aplicado, uma vez que, após um comando do operador, ele obtém automaticamente o ganho crítico e a frequência crítica do processo que geralmente são suficientes para sintonizar o controlador;
- c) o método dos relés funciona em malha fechada, logo, com um ajuste apropriado dos parâmetros do relé, o sistema pode ser ajustado e mantido muito próximo ao ponto de operação. Isso mantém o sistema operando na região linear, permitindo aplicações em sistemas altamente não lineares;

- d) o método dos relés pode ser modificado para lidar efetivamente com perturbações e distúrbios no processo.
- e) o tempo de execução de todo o procedimento é normalmente menor em comparação a outros métodos;
- f) muito eficiente para processos com constante de tempo elevada.

O método dos relés baseia-se na obtenção do ganho crítico  $K_u$  e período crítico  $P_u$ , que podem ser utilizados para obtenção dos ganhos do controlador PID através de diversas regras de sintonia: Ziegler-Nichols, Ciancone-Marlin, Tyreus-Luyben, ITAE Mínimo, entre outras. Algumas destas regras serão apresentadas neste trabalho.

Para obtenção do ponto crítico ( $K_u$  e  $P_u$ ), o sistema é conectado conforme mostrado na Figura 5:

Figura 5 – Estrutura do relé realimentado



Fonte: o autor

Existem na literatura diversas variações do método dos relés. As mais relevantes para este trabalho são apresentadas na sequência. O método do relé ideal também é apresentado, a título de conhecimento, para facilitar o entendimento das demais variações do método dos relés.

### 3.1.1 Relé ideal:

O método dos relés de Aström-Hägglund é baseado na seguinte observação: quando a saída está atrasada  $-\pi$  radianos, a frequência crítica do sistema pode ser determinada automaticamente com o experimento do relé, conforme Figura 5 (FERDOUS, 2005).

A regra geral de comutação do relé ideal é a seguinte:

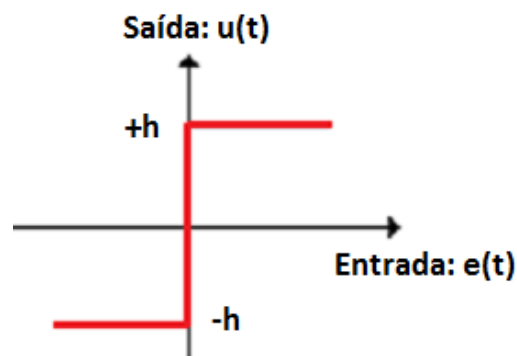
*Se*  $\{\text{erro}(t) \geq 0\}$ , *então*  $u(t) = +h$

*Se*  $\{\text{erro}(t) < 0\}$ , *então*  $u(t) = -h$

Ao iniciar, o relé passa para o valor  $+h$  e após um período  $D$  a saída  $y$  começa a aumentar. No momento em que a saída se torna superior ao valor do setpoint (erro se torna menor do que zero), o relé comuta para  $-h$  e assim sucessivamente (NEVES, 2009).

O sinal de saída da função relé ideal pode ser observado na Figura 6.

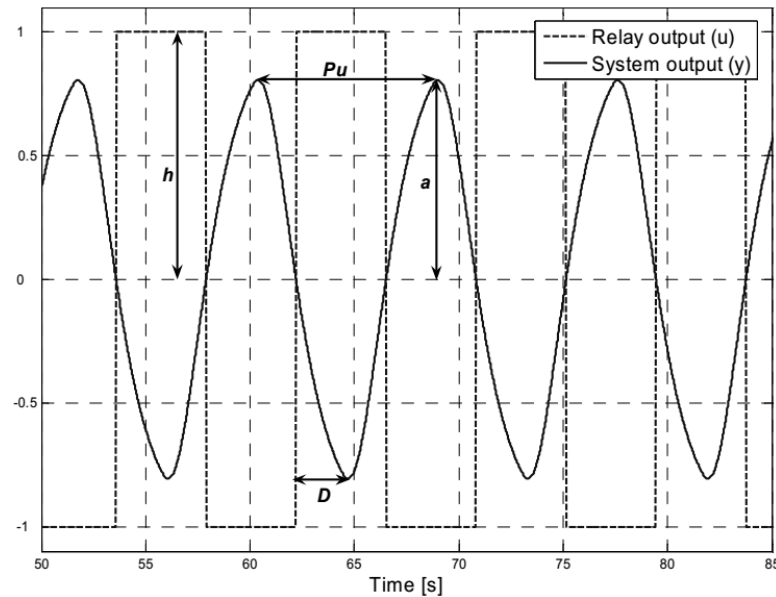
Figura 6 – Função relé ideal



Fonte: o autor

A ação do relé bem como a saída do sistema podem ser observados na Figura 7. Dela pode-se obter também o período crítico  $P_u$ , conforme indicação.

Figura 7 – Método dos relés aplicado



Fonte: Zamani et al. (s.d.)

Através da expansão em séries de Fourier, a amplitude  $a$  pode ser considerada como sendo o resultado da primeira harmônica da saída do relé (YU, 2007). Portanto, o ganho crítico pode ser aproximado pela equação

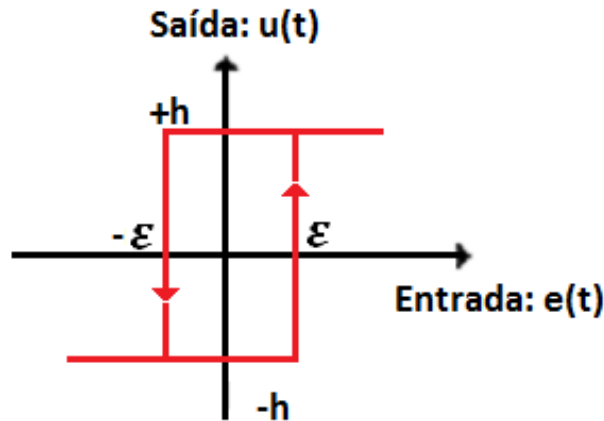
$$K_u = \frac{4h}{\pi a} \quad (9)$$

onde  $a$  é a amplitude de saída do processo e  $h$  a amplitude do relé. A dedução completa de (9) pode ser encontrada em Hang, Aström e Wang (2002), Finke (2016) e Berner (2015).

### 3.1.2 Relé com histerese

CLPs geralmente são utilizados em meios industriais altamente ruidosos, o que faz com que o método dos relés realize chaveamentos desnecessários e indesejados por causa desses ruídos. Para solucionar esse problema, costuma-se introduzir uma zona de histerese na função relé, conforme ilustrado na Figura 8:

Figura 8 – Relé com histerese



Fonte: o autor

A regra geral de comutação do relé com histerese é a seguinte:

$$\text{Se } \{\text{erro}(t) \geq \varepsilon\}, \quad \text{então } u(t) = +h$$

$$\text{Se } \{\text{erro}(t) \leq -\varepsilon\}, \quad \text{então } u(t) = -h$$

$$\text{Se } \{-\varepsilon < \text{erro}(t) < \varepsilon\}, \quad \text{então } u(t) = u(t-1)$$

A curva do relé com histerese é muito parecida com a do relé ideal (Figura 7), a diferença é que com a adição da histerese, uma pequena defasagem aparecerá entre o sinal de saída do processo e o sinal de saída do relé.

Para o relé com histerese, o ganho crítico é dado pela mesma equação do relé ideal, ou seja, pela equação (9). A dedução completa pode ser encontrada em Pinto (2014).

O período crítico  $P_u$  pode ser obtido diretamente do resultado da saída do relé, da mesma forma utilizada no relé ideal.

### 3.2 Regras para obtenção dos parâmetros do controlador

O processo de sintonia através do método dos relés pode ser finalizado com o cálculo dos parâmetros do controlador através da tabela de Ziegler-Nichols, na qual o controlador PID é dado em termos do ganho crítico e do período crítico conforme Tabela 1, obtida do livro de Yu (2007). Yu reforça ainda que se deve atentar que as regras desta tabela são para a estrutura série do controlador PID.

Tabela 1 – Regra de Ziegler-Nichols para controlador PID estrutura série

Controlador	$K_p$	$T_i$	$T_d$
<b>P</b>	$K_u/2$	–	–
<b>PI</b>	$K_u/2,2$	$P_u/1,2$	–
<b>PID</b>	$K_u/1,7$	$P_u/2$	$P_u/8$

Fonte: Yu (2007) pag. 18

A regra de Ziegler-Nichols funciona razoavelmente bem para algumas plantas, porém tende a ser um pouco agressiva, podendo apresentar um sobressinal elevado para muitas aplicações (YU, 2007). Outras regras alternativas são estudadas e implementadas neste trabalho. A escolha entre elas é feita conforme a razão entre o Atraso de Transporte ( $D$ ) e a Constante de Tempo ( $\tau$ ) do sistema, definida como fator de controlabilidade  $C_p$ :

$$C_p = \frac{D}{\tau}. \quad (10)$$

As regras de sintonia de Ziegler-Nichols funcionam bem apenas para uma pequena faixa de  $C_p$ . Seu desempenho começa a diminuir para  $C_p < 0,2$  e para  $C_p > 2$  (YU, 2007), (RIVERA, MORARI, SKOGESTAD, 1986).

Outras regras de sintonia mais conservativas são propostas nas literaturas. Para processos com constante de tempo dominante ( $C_p < 0,1$ ), recomenda-se a utilização das regras de Tyreus-Luyben, já para processos com atraso de transporte dominante ( $C_p > 1,0$ ), recomenda-se as regras de Ciancone-Marlin (YU, 2007, BOAVA, 2016).

As regras de Tyreus-Luyben e Ciancone-Marlin para a estrutura série podem ser vistos na Tabela 2 e Tabela 3, respectivamente.



Tabela 2 – Regra de Tyreus-Luyben para PID estrutura série

Controlador	$K_p$	$T_i$	$T_d$
PI	$K_u/3,2$	$P_u/0,45$	–
PID	$K_u/2,2$	$P_u/0,45$	$P_u/6,3$

Fonte: Yu (2007)

Tabela 3 – Regra de Ciancone-Marlin para PID na estrutura série

Controlador	$K_p$	$T_i$	$T_d$
PI	$K_u/3,3$	$P_u/4$	–
PID	$K_u/3,3$	$P_u/4,4$	$P_u/8,1$

Fonte: Yu (2007)

Segundo Yu (2007), as regras de Tyreus-Luyben apresentam resultados mais conservadores do que as regras de Ziegler-Nichols e além de funcionarem bem para sistemas com constante de tempo dominante, funcionam bem para sistemas multivariáveis.

As regras de Ciancone-Marlin são ligeiramente diferentes das de Ziegler-Nichols sendo a ação Proporcional mais conservadora e a Integral mais agressiva, conforme pode ser visto comparando a Tabela 1 com a Tabela 3.

Outra regra utilizada para obtenção do controlador PI ou PID é o método de ITAE proposto por Rovira (1969), o qual é apresentado na Tabela 4. Este método foi desenvolvido para a estrutura ideal (acadêmica) e para fator de controlabilidade entre 0 e 1 (PINTO, 2014).

O trabalho de O' Dwyer (2000) apresenta o método ITAE mínimo como adequado para obtenção dos parâmetros do controlador para  $0,1 \leq C_p \leq 1,0$ .

Tabela 4 – Regra de ITAE Mínimo para PID estrutura ideal

Controlador	$K_p$	$T_i$	$T_d$
PI	$\frac{0,586}{K_p} \left(\frac{D}{\tau}\right)^{-0,916}$	$\frac{\tau}{(1,03 - 0,165(D/\tau))}$	–
PID	$\frac{0,965}{K_p} \left(\frac{D}{\tau}\right)^{-0,850}$	$\frac{\tau}{(0,796 - 0,147(D/\tau))}$	$0,308\tau \left(\frac{D}{\tau}\right)^{0,929}$

Fonte: Pinto (2014)

Com base nas recomendações feitas pelas referências citadas acima, neste trabalho optou-se por utilizar as regras de Tyreus-Luyben pra plantas com  $C_p < 0,1$ , regras de ITAE para  $0,1 \leq C_p \leq 1,0$  e as regras de Ciancone-Marlin para  $C_p > 1,0$ . O usuário pode também forçar algum desses métodos caso o valor de  $C_p$  encontrado esteja fora da faixa do método desejado, conforme será apresentado na seção 3.3.4.

Como o controlador PID do PLC300 não implementa a estrutura série utilizada por Ciancone-Marlin e por Tyreus-Luyben, foram necessários alguns cálculos para adaptar as tabelas para a estrutura paralela. As regras para o controlador PID na forma paralela podem ser obtidas comparando as equações (6) e (8).

Resolvendo a equação (6) temos:

$$G_c(s) = K_p \left( 1 + \frac{1}{sT_i} \right) (1 + sT_d),$$

$$G_c(s) = \left( K_p + \frac{K_p}{sT_i} \right) (1 + sT_d),$$

isolando os termos:

$$G_c(s) = \left( K_p + \frac{K_p T_d}{T_i} \right) + \left( \frac{K_p}{T_i} \right) \frac{1}{s} + (K_p T_d) s.$$

Comparando os termos separadamente com a equação (8) chega-se às equações (11), (12) e (13) de conversão do PID série para o paralelo, onde  $K_p'$  é o ganho proporcional na forma série e  $K_p$ ,  $K_i$  e  $K_d$  estão na forma paralela.

$$K_p = K_p' + \frac{K_p' T_d}{T_i} \quad (11)$$

$$K_i = \frac{K_p'}{T_i} \quad (12)$$

$$K_d = K_p' T_d \quad (13)$$

Utilizando as fórmulas acima, pode-se facilmente chegar às regras de Tyreus-Luyben e Ciancone-Marlin para a estrutura paralela do controlador PID, conforme Tabela 5 e Tabela 6, respectivamente, e apresentadas no trabalho de Boava (2016).

Tabela 5 – Regra de Tyreus-Luyben para PI/PID

Controlador	$K_p$	$K_i$	$K_d$
PI	$0,31 K_u$	$0,14 K_u/P_u$	–
PID	$0,49 K_u$	$0,21 K_u/P_u$	$0,072 K_u P_u$

Fonte: Boava (2016)

Tabela 6 – Regra de Ciancone-Marlin para PI/PID

Controlador	$K_p$	$K_i$	$K_d$
PI	$0,30 K_u$	$1,2 K_u/P_u$	–
PID	$0,47 K_u$	$1,3 K_u/P_u$	$0,037 K_u P_u$

Fonte: Boava (2016)

### 3.3 Métodos utilizados por diferentes fabricantes

Os métodos de sintonia automática de alguns fabricantes foram estudados. No geral, o método dos relés e o método da resposta ao degrau de malha fechada são os preferidos por eles.

#### 3.3.1 ABB

O PID01A da ABB é um controlador de malha fechada com a função de sintonia automática. Com o processo em regime permanente, o usuário pode dar início à sintonia automática que se baseia no método dos relés. Ao finalizar o processo, o bloco volta a operar no estado anterior ao da sintonia. Novos parâmetros são sugeridos, podendo ou não o usuário aplicar eles ao sistema (ABB, 2005).

Para que a sintonia automática do bloco PID01A através do método dos relés seja aplicada, algumas condições deverão ser satisfeitas antes de iniciar o processo de sintonia:

- O sistema deve estar em regime permanente;
- Não podem ocorrer grandes perturbações;
- O erro do sistema deve ser menor do que 5%.

#### 3.3.2 Emerson Process Management

DeltaV InSight da Emerson Process Management é uma ferramenta para monitorar, analisar, diagnosticar e melhorar o desempenho de sistemas de controle. DeltaV InSight possui duas formas de sintonia de malha de controle, a sintonia on-demand e a sintonia adaptativa.

A Sintonia on-demand é semelhante à utilizada pelo bloco PID01A da ABB sendo também baseada no método dos relés de Aström-Hägglund. O usuário pode ajustar a velocidade desejada da resposta da malha de controle, podendo facilmente aumentar ou diminuir a mesma. Usuários avançados podem também optar por diferentes regras como a de Ziegler-Nichols, Lambda e Integral Model Control (EMERSON PROCESS MANAGEMENT, 2013).

A sintonia adaptativa automaticamente altera a sintonia da malha conforme as condições do processo vão alterando. A sintonia adaptativa do DeltaV InSight é baseada em modelos de processo que são continuamente calculados com base nas alterações efetuadas no dia a dia de operação (EMERSON PROCESS MANAGEMENT, 2013).

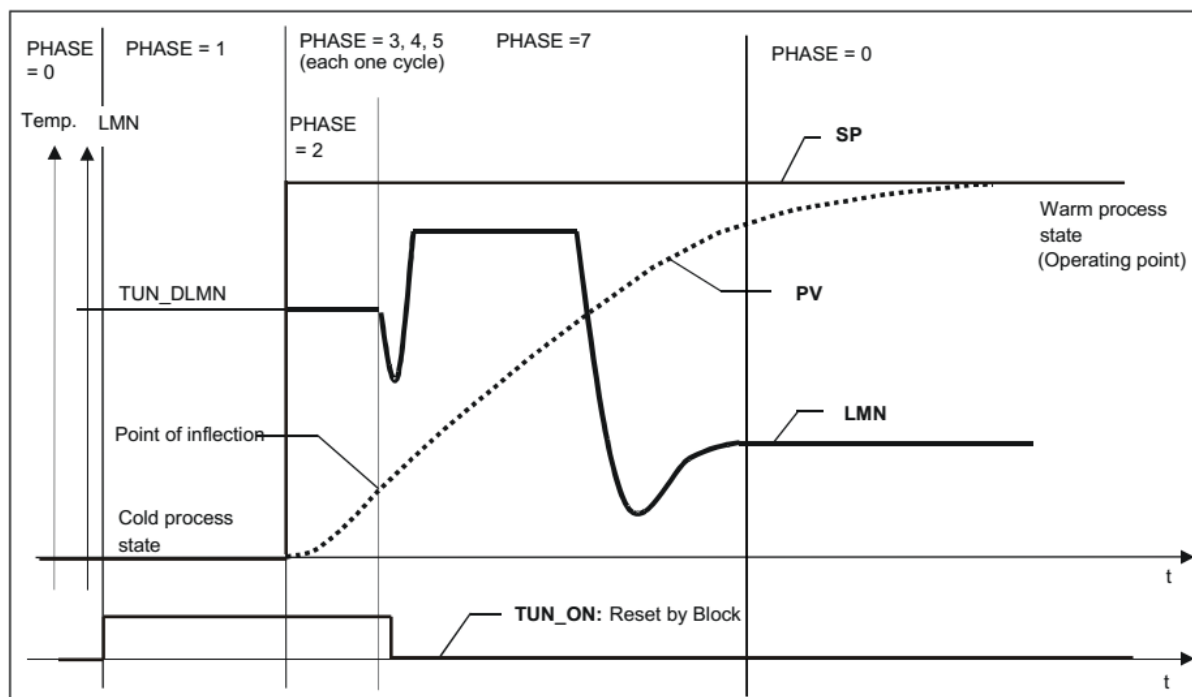
### 3.3.3 Siemens

A Siemens possui o bloco TCONT\_CP que efetua a sintonia automática de controladores em malha fechada. Ele é recomendado para sistemas com  $C_p < 0,1$ , mas pode ser aplicado também para os demais casos. Ele não se aplica em sistemas em que, após um degrau na variável manipulada, em malha fechada, a variável de processo não atinge o valor de regime permanente, ou seja, não se aplica a sistemas integradores.

Para obtenção dos parâmetros do controlador, o bloco aplica um degrau no *setpoint* saindo do valor inicial para o ponto de operação.

O processo de sintonia do bloco é dividido em sete fases, conforme exibido no sistema de controle de temperatura da Figura 9.

Figura 9 – Etapas da sintonia automática do bloco TCONT\_CP



Fonte: Manual TCONT\_CP Siemens, 2003

As fases 0 e 1 são de preparação para a sintonia, nas quais o sistema deve estar em regime permanente ou muito próximo a ele. A fase 2 é a de aplicação do degrau na variável manipulada para o sistema atingir o *setpoint*, momento em que é dado início à sintonia. As Fases 3, 4 e 5 são as fases em que o controlador é calculado e na Fase 7 o tipo de processo é verificado e dependendo dos resultados, os parâmetros do controlador podem ser recalculados (SIEMENS, 2003)

### 3.3.4 WEG

Um novo bloco, o PID2, foi desenvolvido para realizar a sintonia automática de controladores PID utilizados nos controladores WEG. Este bloco, além de conter todas as funcionalidades do bloco PID, também é capaz de realizar a sintonia automática dos seus parâmetros de controle  $K_p$ ,  $K_i$  e  $K_d$ . O capítulo seguinte descreve um pouco o CLP da empresa WEG e o capítulo 5 como foi implementado e como funciona o bloco PID2.

## 4 O PLC300

O PLC300 é um CLP com IHM integrada desenvolvido pela WEG para atender a necessidade de interface com o usuário em painéis e máquinas e ao mesmo tempo um completo CLP expansível. O PLC300 é comercializado há aproximadamente seis anos e está constantemente em evolução, recebendo novas funcionalidades para atender as mais diversas aplicações. É um CLP rápido e com vários protocolos de comunicação. Entre eles podemos citar: CANopen, Modbus RTU (rede RS232 e RS-485) e Modbus TCP (rede Ethernet) (MANUAL PLC300).

### 4.1 O microcontrolador do PLC300 e sua programação

O microcontrolador utilizado pelo PLC300 e no qual a implementação foi realizada é o LPC1778 da NXP, baseado no processador ARM Cortex-M3 de 32 Bits, que roda à 120MHz. Este microcontrolador possui memória flash de 512kBytes e SRAM de 96kBytes além de 4032 bytes de memória EEPROM. Possui comunicação USB, Ethernet, UART, entrada para leitura de *encoder*, quatro *timers*, dois PWMs, até 165 pinos de entradas/saídas entre outras características e funções (DATASHEET LPC1778). A programação do microcontrolador é feita em linguagem C.

Segundo Kenighan e Ritchie (2010), em seu prefácio:

C é uma linguagem de programação de finalidade geral que permite economia de expressão, modernos fluxos de controle e estruturas de dados e um rico conjunto de operadores. C não é uma linguagem de "muito alto nível", nem "grande" e nem específica para uma área de aplicação particular. Mas, sua falta de restrições e sua generalidade tornam-na mais conveniente e eficaz para muitas tarefas do que linguagens supostamente mais poderosas.

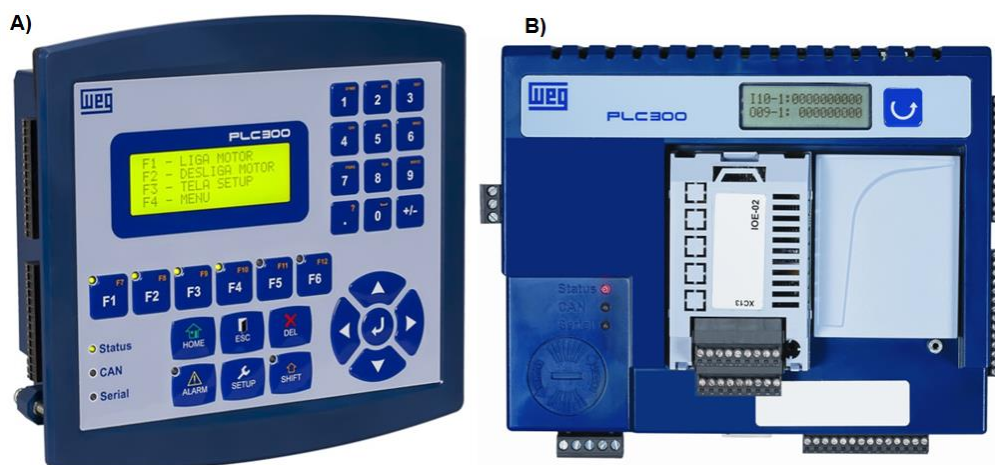
Além dos motivos expressos acima, a linguagem C foi escolhida por ser amplamente difundida no meio acadêmico e industrial. A grande maioria dos fabricantes disponibiliza uma gama muito grande de exemplos, aplicações e bibliotecas para os seus microcontroladores, implementados em linguagem C, o que facilita muito o desenvolvimento de novos projetos. Outro motivo para esta escolha é que diversos produtos da empresa utilizam esta mesma linguagem, isso permite que as funções implementadas sejam facilmente adaptadas a outros produtos.

## 4.2 Principais características do PLC300

Possui alta velocidade de processamento, proporcionada por um processador Cortex-M3 de 32 bits rodando a 120MHz.

Possui versões com e sem IHM integrada. A IHM é composta por um *display* de 4 linhas e 20 caracteres cada, teclas que permitem acessar funções do usuário, configurações, alarmes, possui teclas para navegação entre os campos ou dados do display, teclado alfanumérico entre outras, conforme pode ser visualizado na Figura 10 (A):

Figura 10 – Modelo do PLC300 com IHM (A) e sem IHM (B)



Fonte: [www.weg.net](http://www.weg.net)

A versão sem IHM permite que o PLC300 seja montado em trilho DIN. Nesta versão, a parte traseira do produto torna-se a frente e apresenta um display de 2 linhas e 16 caracteres que exibem, entre outras informações, o status das I/Os. Para navegar entre as informações esta versão apresenta também um botão, conforme pode ser visto na Figura 10 (B).

Possui funcionalidade *Hot Download* que permite carga a quente do recurso, ou seja, que o programa *Ladder*, telas, alarmes e outros dados sejam alterados no produto sem que o mesmo interrompa a execução do programa atual.

O PLC300 possui um *slot* para cartão de memória do tipo SD de até 32GBytes que permite salvar dados, fazer log de eventos, receitas, fazer backup da aplicação, etc.

### 4.3 Saídas digitais, analógicas e PWM

O PLC300 possui 8 saídas digitais de 500mA cada, em 24Vcc, do tipo PNP. Estas saídas são isoladas galvanicamente e protegidas contra surtos de tensão, curto-circuito e sobrecarga.

Possui uma saída rápida de até 300kHz com capacidade de 100mA em 24Vcc que pode ser programada para Modulação PWM e também ser usada como saída digital.

A saída analógica é de 10Bits e possui bornes independentes para corrente e tensão.

### 4.4 Entradas digitais e analógicas

O PLC300 possui 10 entradas digitais isoladas, em 24Vcc. Todas elas podem gerar interrupção de evento no programa do usuário. Duas delas são rápidas e podem ser usadas para entrada de encoder em quadratura ou para contagem de pulsos de até 100kHz. A entrada analógica diferencial é de 12 Bits e pode ser configurada em tensão (0V a 10V) ou em corrente (0mA a 20mA ou 4mA a 20mA).

### 4.5 Entrada de encoder

Além das duas entradas rápidas, o PLC300 possui uma entrada exclusiva para encoder do tipo quadratura, com sinais complementares e detecção de fio partido com alarme opcional. O PLC fornece alimentação configurável em 5 ou 12V.

### 4.6 Módulos de expansão (MANUAL PLC300, p. 2-3)

- IOA-01: 1 entrada analógica de 14 bits em tensão e corrente; 2 entradas digitais; 2 saídas analógicas de 14 bits em tensão e corrente; 2 saídas digitais tipo coletor aberto.
- IOB-01: 2 entradas analógicas isoladas em tensão e corrente; 2 entradas digitais; 2 saídas analógicas isoladas em tensão e corrente (mesma programação das saídas do CFW-11 padrão); 2 saídas digitais tipo coletor aberto.
- IOC-01: 8 Entradas digitais; optoacopladas; 24 Vcc; atuação nível alto/baixo; 4 saídas digitais; relé contatos NA; capacidade 240 Vca/1 A.
- IOC-02: 8 Entradas digitais; optoacopladas; 24 Vcc; atuação nível alto/baixo; 8 saídas digitais; coletor aberto; 24 V/0,1 A.
- IOC-03: 8 entradas digitais; optoacopladas; 24 Vcc; configuráveis, atuação nível alto/baixo; 7saídas PNP, protegidas; 24 Vcc; 500 mA.
- IOE-01: 5 entradas de termistores tipo PTC simples ou triplo com isolamento reforçada em relação ao 0 V do PLC300.



- IOE-02: 5 entradas de termistores tipo PT100 com isolação reforçada em relação ao 0 V do PLC300.
- IOE-03: 5 entradas de termistores tipo KTY84 com isolação reforçada em relação ao 0 V do PLC300.

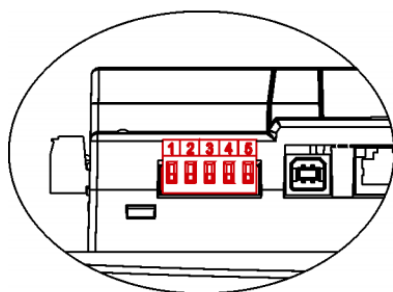
#### 4.7 Interfaces de comunicação

O PLC300 possui interface de comunicação serial RS232 e RS485 (isolada) com protocolo modbus RTU Mestre/Escravo. A comunicação serial possibilita, através do WPS, a criação de protocolos ASCII para leitura de códigos de barras ou outras aplicações. Além da comunicação serial, possui também CAN isolada com protocolo CANopen Mestre/Escravo, Ethernet com protocolo modbus TCP e USB para comunicação com o computador.

Como a rede CAN será utilizada para realizar os ensaios práticos deste trabalho, uma breve descrição sobre ela será apresentada a seguir.

O conector da rede CAN possui 5 pinos, a descrição de cada um deles pode ser vista na Figura 11.

Figura 11 – Detalhe do conector da rede CAN do PLC300



Pino	Nome	Função
1	V-	Pólo negativo da fonte de alimentação
2	CAN_L	Sinal de comunicação CAN_L
3	Shield	Blindagem do cabo
4	CAN_H	Sinal de comunicação CAN_H
5	V+	Pólo positivo da fonte de alimentação

Fonte: Manual CANopen PLC300, 2010.

Suas principais características são (MANUAL CANOPEN PLC300, 2010):

- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Alimentação externa de 24V.
- Permite a conexão de até 64 dispositivos no mesmo segmento. Uma quantidade maior de dispositivos pode ser conectada com o uso de repetidores.
- Comprimento máximo do barramento de 1000 metros.

Os dados de uma rede CAN são transmitidos através de frames compostos por um campo identificador de 11 bits e um campo de dados que pode conter até 8 bytes de informação. A especificação CANopen define como é feita a troca de dados entre os equipamentos e como cada dispositivo deve interpretar esses dados. Basicamente ela funciona com o sistema mestre/escravo, em que o telegrama é transmitido pelo mestre para todos os escravos sendo que um ou mais deles (conforme solicitação) irá interpretar e, se necessário, responder ao telegrama. A velocidade de comunicação varia de 10kbit/s até 1Mbit/s. Mais detalhes sobre o protocolo CANopen de comunicação podem ser encontrados no Manual CANopen do PLC300 (2010).

#### **4.8 Alimentação**

O PLC300 deve ser alimentado em  $24V_{cc} \pm 15\%$  com uma capacidade de corrente de pelo menos 500mA. Internamente, um conversor DC/DC fornece as demais tensões necessárias para o produto. A isolação da fonte permite que a mesma seja utilizada para alimentar outros dispositivos e até mesmo suas próprias entradas e saídas digitais.

#### **4.9 A programação do PLC300**

A programação do PLC300 é feita através da linguagem *ladder* utilizando o software gratuito desenvolvido pela WEG, o WPS, que pode ser baixado no site da empresa.

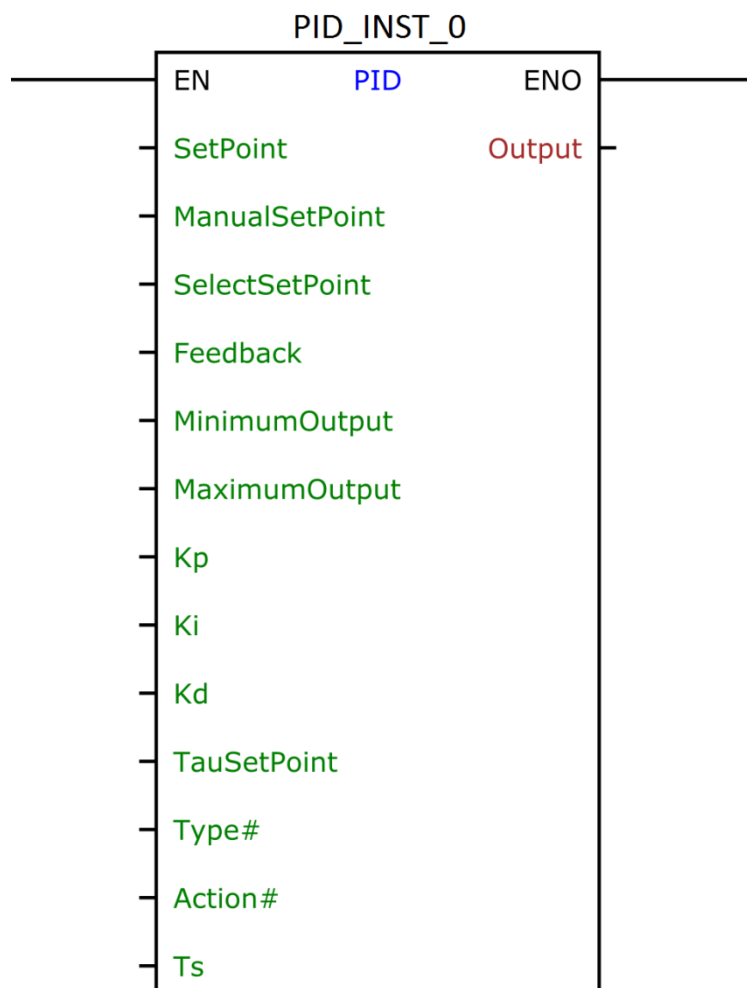
Além da programação via *Ladder*, o WPS também permite fazer a configuração de telas, atribuir funções às teclas do usuário, configurar as comunicações, atualizar o firmware do produto, entre outras funcionalidades. Com o PLC300 conectado ao WPS, é possível rodar a aplicação, monitorar e fazer gráficos das variáveis que desejar.

O PLC300 possui mais de 100 blocos com funções pré-programadas para facilitar o desenvolvimento da aplicação. Entre eles, blocos temporizadores, blocos para leitura de encoder, conversão de tipos de dados, blocos de comparação e muitos outros. Um bloco que será apresentado em mais detalhes é o bloco de controle PID.

#### 4.10 O bloco PID

O PLC300 possui um bloco que executa a função de um controlador PID discreto. A partir das variáveis de entrada, ele calcula a saída correspondente do controlador. O bloco PID pode ser visualizado na Figura 12:

Figura 12 – Bloco do controlador PID do PLC300



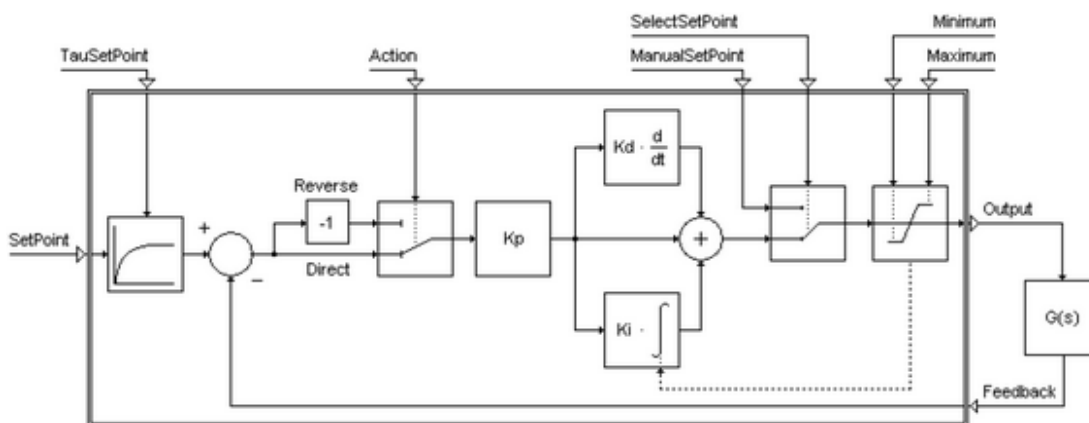
Fonte: Manual de software do produto

Tabela 7 – Parâmetros do bloco PID do PLC300

Tipo de Variável	Nome	Tipo de Dado	Descrição
Entrada	EN	BOOL	Habilitação do bloco
	SetPoint	REAL	Referência automática (pré-controle)
	ManualSetPoint	REAL	Referência forçada (pós-controle)
	SelectSetPoint	BOOL	Seleciona qual referência utilizar
	Feedback	REAL	Variável de realimentação da malha
	MinimumOutput	REAL	Valor mínimo da saída do controlador
	MaximumOutput	REAL	Valor máximo da saída do controlador
	Kp	REAL	Ganho proporcional
	Ki	REAL	Ganho integral
	Kd	REAL	Ganho derivativo
	TauSetPoint#	REAL	Constante de tempo do filtro de entrada de referência automática
	Type#	BYTE	Tipo de controlador
	Action#	BYTE	Ação de controle
Ts#	UINT	Período de amostragem [ms]	
Saída	ENO	BOOL	Habilitação de saída
	Output	REAL	Saída do controlador
Instância	PID_INST_0	PID	Instância de acesso à estrutura do bloco

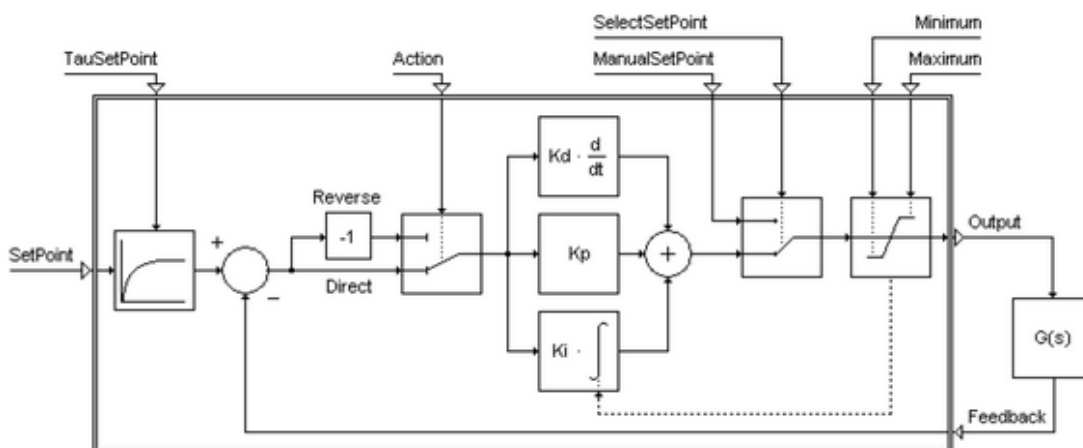
O bloco PID funciona da seguinte forma: na transição positiva da entrada EN, o bloco executa uma rotina de controle PID conforme os parâmetros  $K_p$ ,  $K_i$  e  $K_d$  escolhidos. Esse bloco possibilita a utilização de duas topologias, Acadêmica ou Paralela, que podem ser selecionadas através de *Type#*. Um esquemático de cada uma dessas topologias pode ser visualizado na Figura 13 e Figura 14. Mais detalhes sobre o funcionamento do bloco podem ser encontrados no manual de programação do produto disponível em [www.weg.net](http://www.weg.net).

Figura 13 – Topologia acadêmica do PID do PLC300



Fonte: manual de programação do PLC300

Figura 14 – Topologia paralela do PID do PLC300



Fonte: manual de programação do PLC300

O trabalho de Pinto (2014) traz uma tabela com o algoritmo do controlador PID implementado por diversos fabricantes, conforme Figura 15. Em seu trabalho, ele ressalta a importância de sempre revisar e seguir o que se encontra no manual de cada CLP ou SDCD (Sistema Digital de Controle Distribuído), pois o fabricante pode realizar melhorias, ou alterar a estratégia do algoritmo PID implementado. Além disso, séries diferentes do mesmo fabricante podem apresentar estratégias diferentes.

Os fabricantes Altus, HI-Tecnologia, Schneider, Smar e Yogogawa utilizam a estrutura acadêmica em seus controladores PID. Os CLPs da GE-Fanuc e Siemens

utilizam a estrutura paralela. O fabricante Allen Bradley, possui controladores para aplicar tanto a estrutura acadêmica quanto a paralela, assim como o PLC300 do fabricante WEG.

Figura 15 – Algoritmo PID de controladores comerciais\*

Fabricantes	Modelos	Algoritmo PID	Parâmetros do Controlador
Allen Bradley	Contrologix - PID independente	$MV = K_p e(t) + K_i \int e(t) dt + K_d \frac{\Delta PV}{dt}$	$K_p$ : Ganho; $K_i$ : ganho integral; $K_d$ : ganho derivativo
Allen Bradley	Contrologix - PID dependente	$MV = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{\Delta PV}{dt} \right)$	$K_p$ : Ganho; $T_i$ : reset time (min); $T_d$ : rate time (min)
Siemens	S7 PB41	$MV = K_p e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{e(t)}{dt}$	$K_p$ : Ganho; $T_i$ : reset time (seg); $T_d$ : tempo derivativo (seg)
Yogogawa	Field Control Station PID	$MV = \frac{100}{BP} \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$	Ganho: banda proporcional; $T_i$ : tempo integral (seg); $T_d$ : tempo derivativo (seg)
Smar	MultiLoop CD-600	$MV = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$	$K_p$ : ganho; $T_i$ : tempo integral (min); $T_d$ : tempo derivativo (min)
GE-Fanuc	Series 90-30, 90-70 independente	$MV = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$	$K_p$ : ganho (0,01); $K_i$ : reset time (0,001 rep/s); $K_d$ : ganho derivativo
Schneider	Familia Modicon	$MV = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$	$K_p$ : ganho (0-100); $T_i$ : tempo integral (seg:1-3600); $T_d$ : tempo derivativo (seg:0-900)
HI-Tecnologia	ZAP-900, 500 e Flex950	$MV = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$	$K_p$ : ganho (0-1000); $T_i$ : tempo integral (rep/min: 0.001-5000); $T_d$ : tempo derivativo (min: 0-1000)
Altus	Series: DUO, AL c FBs	$MV = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$	$K_p$ : ganho; $T_i$ : tempo integral (seg); $T_d$ : tempo derivativo(seg)

Fonte: Pinto, 2014

\* Na equação do CLP da Siemens está faltando a letra *d* no termo derivativo:

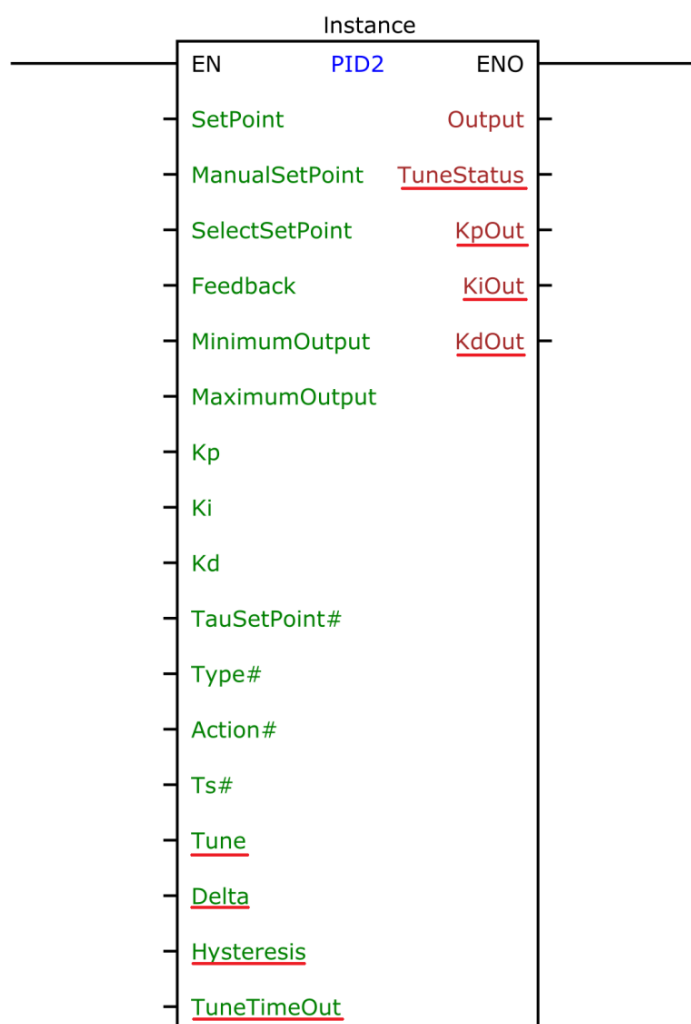
$$MV = K_p e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt}.$$

## 5 PID2: CONTROLADOR PID DO PLC300 COM SINTONIA AUTOMÁTICA

### 5.1 O bloco PID2 visto pelo usuário

O bloco PID2 desenvolvido apresenta as mesmas características de controle do bloco PID já apresentado. Adicionalmente, ele possui a funcionalidade de sintonia automática do controlador, que pode ser acionada a qualquer momento pelo operador. Para tanto, o bloco PID2 possui quatro entradas diferentes que são elas *Tune*, *Delta*, *Hysteresis* e *TuneTimeOut*, além de quatro saídas *TuneStatus*, *KpOut*, *KiOut* e *KdOut*. Estas variáveis estão em destaque na Figura 16 e uma breve descrição sobre cada uma delas está na sequência.

Figura 16 – Bloco PID2 desenvolvido



Fonte: o autor.

### 5.1.1 A variável Tune

A variável de entrada *Tune* é do tipo Byte e quando recebe um valor diferente de zero dá início ao processo de sintonia. Além disso, ela também determina qual o tipo de controlador que será utilizado. O operador pode optar pelo controlador PI ou PID e também se a regra utilizada para os cálculos será conforme  $C_p$  (forma automática), ou se alguma das regras implementadas será forçada, conforme Tabela 8.

Tabela 8 – Opções da sintonia automática.

<i>Tune</i>	Controlador
0	Desabilitado
1	Controlador PID Automático
2	Controlador PI Automático
3	Controlador PID Tyreus-Luyben
4	Controlador PI Tyreus-Luyben
5	Controlador PID ITAE
6	Controlador PI ITAE
7	Controlador PID Ciancone-Marlin
8	Controlador PI Ciancone-Marlin

Após a calibração ser concluída, o método poderá ser alterado entre as opções 1 a 8 sem que o sistema execute novamente a calibração. Isto permite que o usuário experimente diferentes resultados e escolha o que melhor se adaptar ao seu sistema. Caso *Tune* receba o valor 0 (zero), na próxima mudança de valor o sistema irá reiniciar a calibração.

### 5.1.2 As variáveis Delta e Hysteresis

As variáveis *Delta* e *Hysteresis* são parâmetros ajustáveis do método do relé. *Delta*, representado por  $h$  no método dos relés, é a variação da variável manipulada com relação ao valor que atingiu o *Setpoint* (recomenda-se um valor de 10% inicialmente). *Delta* deve ser suficientemente grande para gerar uma oscilação na saída do processo. *Hysteresis* determina o momento de chaveamento do relé. Ela deve ser ligeiramente maior do que o ruído do sistema (pode-se iniciar com 2% do ponto de operação).



### 5.1.3 A variável TuneTimeOut

A variável *TuneTimeOut* é o tempo máximo em milissegundos (*ms*) de espera até que a sintonia seja concluída. O tempo começa a contar sempre que *Tune* é habilitado. Caso o tempo seja atingido e o sistema ainda não tenha finalizado a sintonia, o processo é abortado e o bloco retorna para a situação imediatamente anterior ao início da sintonia. Os valores de saída não são alterados, com exceção de *TuneStatus* que indicará que ocorreu *timeout*.

### 5.1.4 A variável TuneStatus

Indica o status da sintonia conforme Tabela 9:

Tabela 9 – Status da sintonia

<i>TuneStatus</i>	Significado
0	Desabilitado
1	Relé em nível alto
2	Relé em nível baixo
4	Sistema estabilizado
5	Estável e Relé em nível alto
6	Estável e Relé em nível baixo
8	Sintonia concluída
16	Timeout

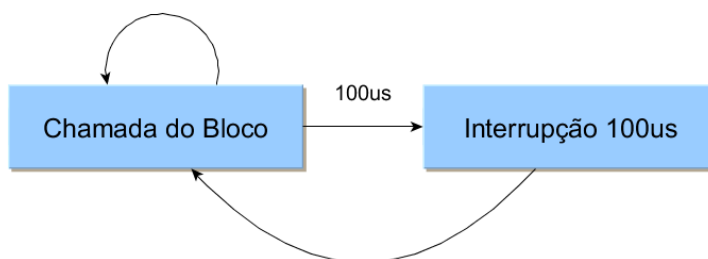
### 5.1.5 As variáveis KpOut, KiOut e KdOut

Essas três variáveis indicam os ganhos Proporcional, Integral e Derivativo, respectivamente, obtidos após a calibração do sistema. O usuário pode fazer uma lógica *ladder* simples para copiar esses parâmetros para entrada do bloco quando desejar. Outra alternativa é utilizar as mesmas variáveis das entradas (*Kp*, *Ki* e *Kd*), desta forma, ao completar o processo de sintonia, as variáveis serão automaticamente copiadas para a entrada e o controlador passará a operar com os novos ganhos obtidos. A vantagem da primeira opção é que o usuário pode avaliar os resultados obtidos antes de atualizar os valores no controlador.

## 5.2 A implementação do bloco PID2 no sistema embarcado

O código responsável pela execução do bloco PID2 foi implementado em duas rotinas separadas no processador. A primeira parte roda no laço principal do programa, sendo chamada sempre que o bloco na linguagem *ladder* é executado, dependendo assim do ciclo de *scan* da aplicação. A segunda parte roda em uma interrupção de tempo de  $100\mu\text{s}$ . Nesta segunda parte, a aquisição dos dados e os cálculos são efetuados, permitindo uma taxa de amostragem maior e independente do ciclo de *scan*, não comprometendo assim o desempenho do sistema. O fluxograma da Figura 17 ilustra a chamada do bloco e a interrupção de tempo.

Figura 17 – Fluxograma da execução da sintonia automática



Fonte: o autor

## 5.3 A chamada do bloco PID2

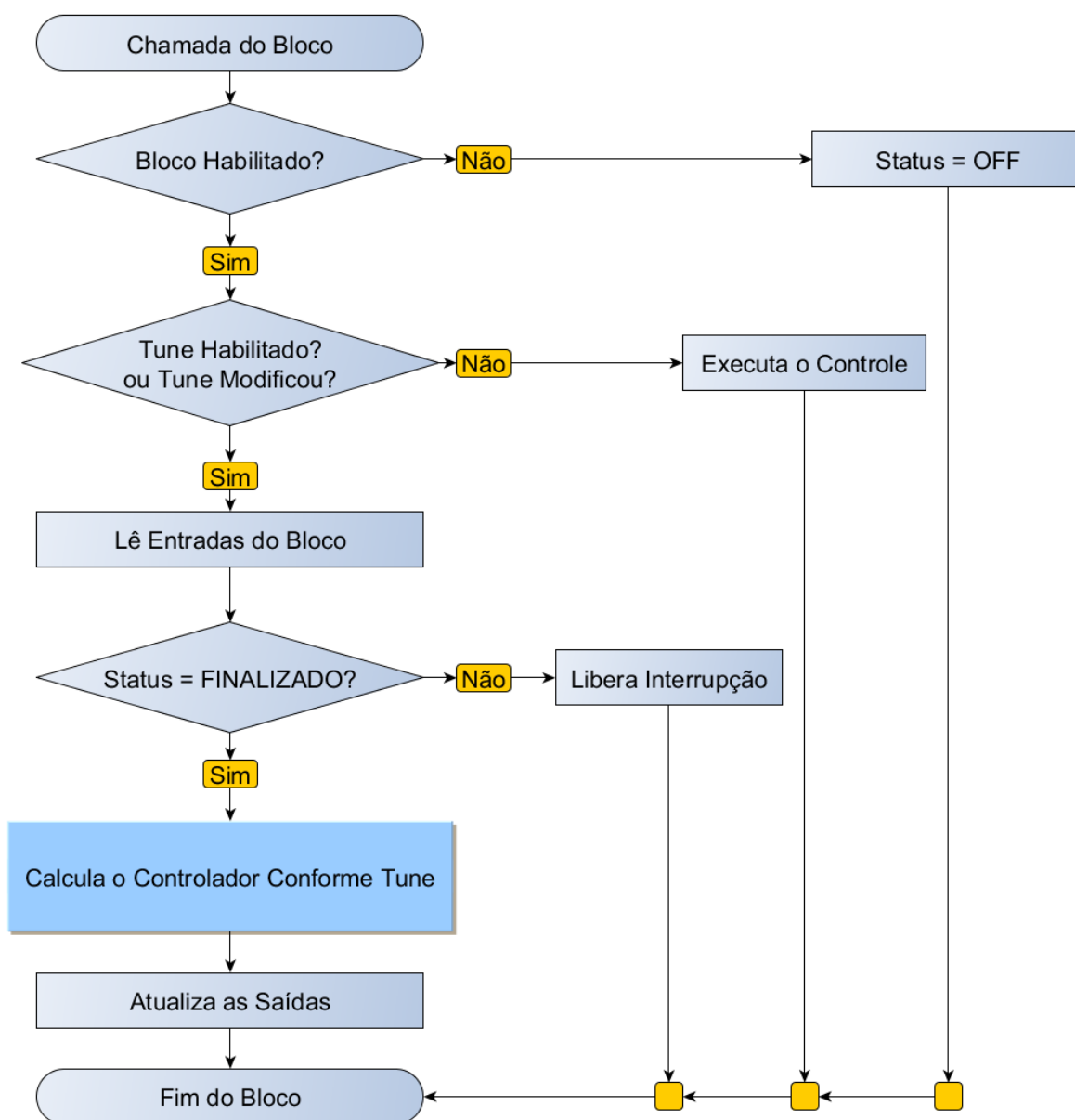
Na primeira parte, quando o bloco é executado, o *firmware* faz uma leitura da variável *Tune* e caso ela esteja com algum valor diferente de zero, as demais entradas são lidas e a interrupção de  $100\mu\text{s}$ , que irá aplicar o método dos relés, é habilitada para execução. Além destas configurações iniciais, esta rotina monitora, através de uma variável de *status*, se a sintonia automática já foi finalizada com sucesso, desta forma, os cálculos do controlador são efetuados e as saídas do bloco atualizadas. Por fim, *Tune* é desabilitado internamente e a função de controle volta a ser executada conforme configuração imediatamente anterior ao início da sintonia.

Uma grande dificuldade reportada por Finke (2016), por outros autores e também experimentada neste trabalho, é a de encontrar um método que se adapte muito bem para toda a diversidade de plantas do mercado. Pensando nisto, caso *Tune* receba algum valor diferente do escolhido para a sintonia, que seria a escolha de alguma outra regra, os cálculos são refeitos e as saídas atualizadas, sem que o

método dos relés necessite ser aplicado novamente. Essa possibilidade traz um grande diferencial para esta ferramenta, uma vez que regras diferentes podem ser aplicadas com apenas um comando, podendo o usuário escolher qual a melhor se adapta à planta, mesmo que ele não tenha conhecimento algum sobre as diferenças entre elas.

O fluxograma da Figura 18 mostra simplificada o funcionamento do bloco.

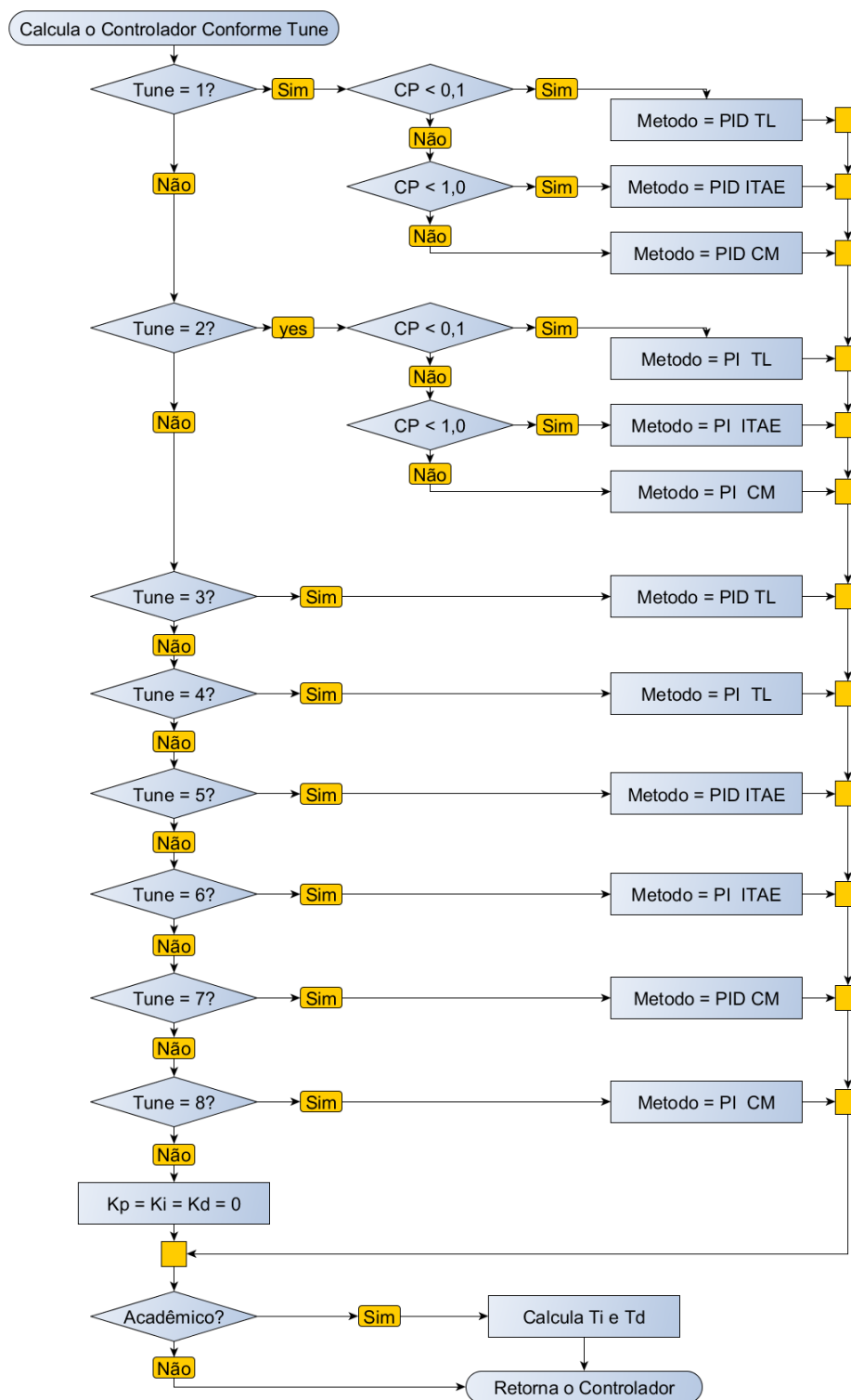
Figura 18 – Fluxograma do bloco PID2



Fonte: o autor

A escolha da regra a ser utilizada é feita conforme *Tune* e o fator de controlabilidade, representado por *CP* no fluxograma da Figura 19. As regras de Tyreus-Luyben, e Ciancone-Marlin foram abreviadas por TL e CM, respectivamente.

Figura 19 – Cálculo do controlador



Fonte: o autor

Embora o controlador paralelo tenha sido escolhido para obtenção dos resultados neste trabalho, o bloco permite que o usuário realize a sintonia para a forma acadêmica também, conforme pode ser visto no final do fluxograma da Figura 19. Caso o usuário tenha escolhido a forma acadêmica, os valores de  $T_i$  e  $T_d$  são calculados.

#### 5.4 A interrupção de 100us

Uma vez que o bloco PID2 habilita a interrupção de tempo de 100 $\mu$ s, o método dos relés com histerese é aplicado ao sistema. Na primeira execução, *Status* está em *OFF*, com isso, os dados são inicializados, o valor máximo e mínimo da Variável Manipulada (MV) são calculados com base em *Delta* e o primeiro degrau é dado. Caso  $erro > 0$  o degrau será positivo, caso  $erro < 0$  o degrau será negativo, conforme Figura 8.

Toda vez que a interrupção é executada, o tempo configurado em *TuneTimeout* é verificado e caso ultrapasse o valor definido, a sintonia é interrompida e a saída de status indica a falha.

##### 5.4.1 O ajuste de Bias

A partir do segundo semiciclo do relé, um ajuste em *bias* é efetuado com o intuito de buscar uma senóide mais uniforme como saída do relé. A equação do ajuste adaptada do trabalho de Neves (2009) é dada por:

$$bias = bias + 0,2 bias \frac{T_{up} - T_{down}}{T_{up} + T_{down}}, \quad (14)$$

$T_{up}$  e  $T_{down}$  representam o tempo em nível alto e baixo do relé, respectivamente.

A amplitude da oscilação dada por *Delta* não é alterada com este cálculo, ele apenas faz um pequeno deslocamento de *bias*, necessário caso o sistema não esteja bem estabilizado no ponto de operação, situação em que a oscilação na saída poderia trazer resultados inadequados para os cálculos.

Quando o erro entre  $T_{up}$  e  $T_{down}$  é menor do que 10%, significando que temos uma oscilação uniforme na saída, o sistema armazena dois períodos completos e efetua os cálculos/medições do ganho estático, período, frequência e ganho críticos, constante de tempo, fator de controlabilidade e atraso de transporte.

#### 5.4.2 O ganho estático

O ganho estático segundo Hang, Aström e Wang (2002) é dado pela razão da integral da saída da planta pela integral da entrada, conforme equação:

$$k = \frac{\int_0^{P_u/2} y(t) dt}{\int_0^{P_u/2} u(t) dt}. \quad (15)$$

Neste trabalho, a integral da entrada foi obtida através da regra dos trapézios.

#### 5.4.3 O ponto crítico de operação ( $K_u$ , $P_u$ e $W_u$ )

O ganho crítico  $K_u$  é dado pela Equação (9). O período crítico  $P_u$  é medido no firmware, conforme Figura 28. A frequência crítica  $W_u$  é dada por

$$W_u = \frac{2\pi}{P_u}. \quad (16)$$

#### 5.4.4 A constante de tempo do sistema

Para obtenção da constante de tempo  $\tau$  do sistema, aproxima-se a planta a ser controlada por um sistema de primeira ordem com atraso de transporte e utiliza-se a equação

$$\tau = \frac{\sqrt{(kk_u)^2 - 1}}{W_u} \quad (17)$$

apresentada por Yu (2007). Embora nem todas as plantas possam ser aproximadas por um sistema de primeira ordem, a constante de tempo é utilizada nesta implementação apenas para a escolha automática da regra a ser utilizada, não comprometendo a funcionalidade da sintonia, uma vez que o usuário tem a possibilidade de trocar de regra caso a escolhida não traga bons resultados.

#### 5.4.5 O fator de controlabilidade

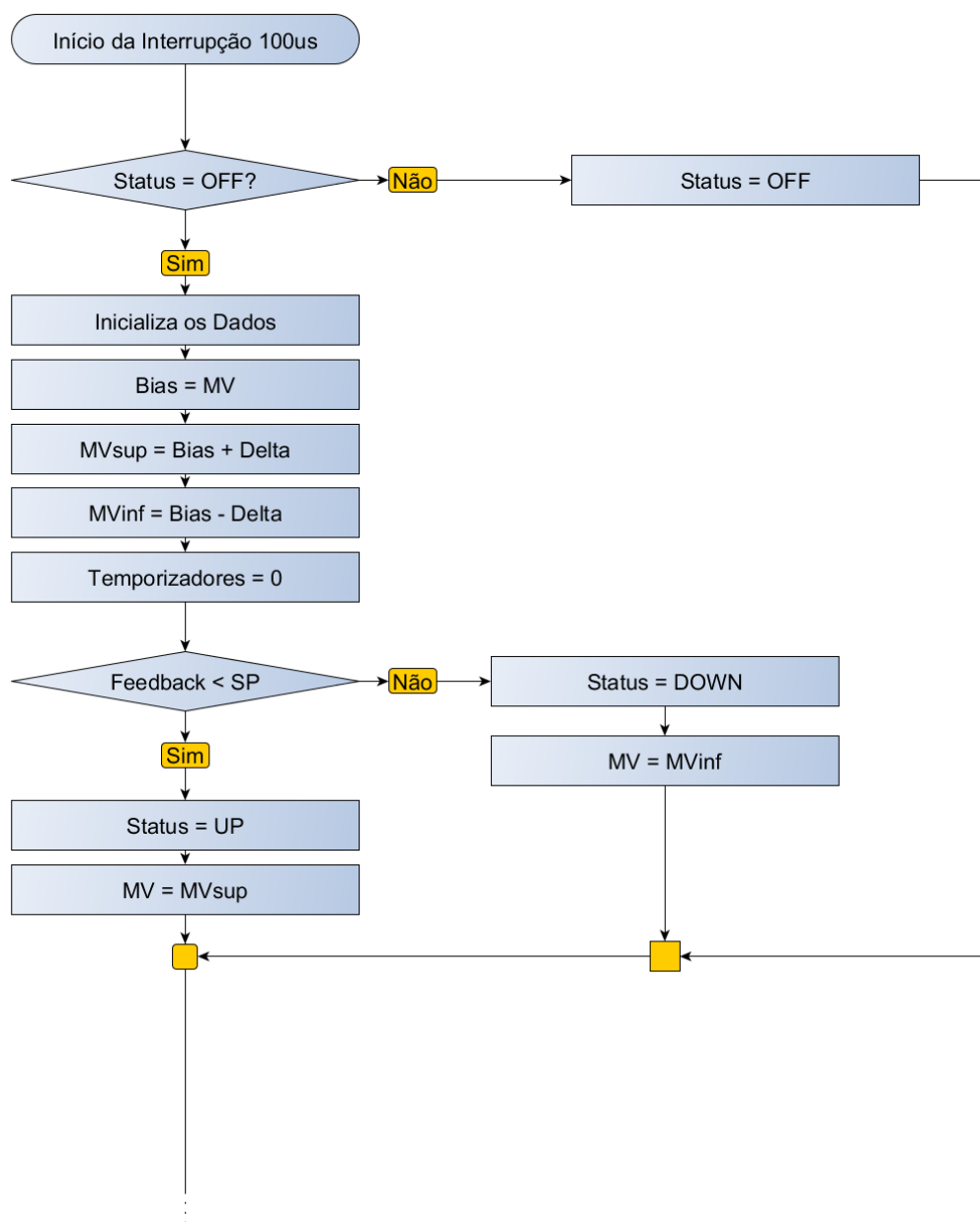
O fator de controlabilidade  $C_p$  já foi definido e apresentado na subseção 3.2 - Regras para obtenção dos parâmetros do controlador. Seu valor é obtido através da Equação (10).

#### 5.4.6 O atraso de transporte

O atraso de transporte é obtido através da medição do tempo decorrido após um chaveamento do relé, até que a variável de saída atinja seu ponto máximo ou mínimo, conforme apresentado por  $D$  na Figura 7.

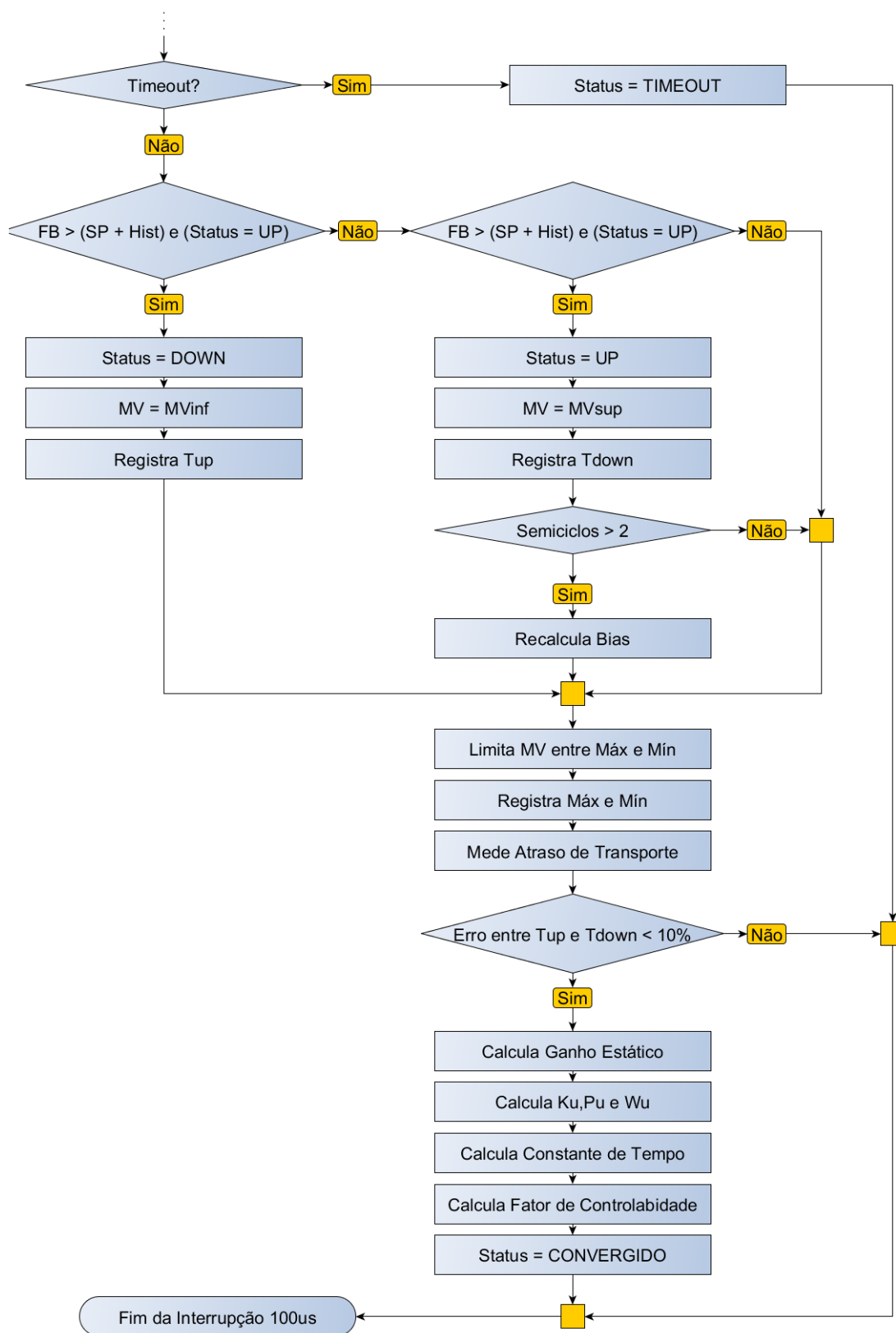
O fluxograma que inicia na Figura 20 e termina na Figura 21 mostra, resumidamente, como o método dos relés é aplicado na interrupção de tempo.

Figura 20 – Aplicação do método dos relés (a)



Fonte: o autor

Figura 21 – Aplicação do método dos relés (b)



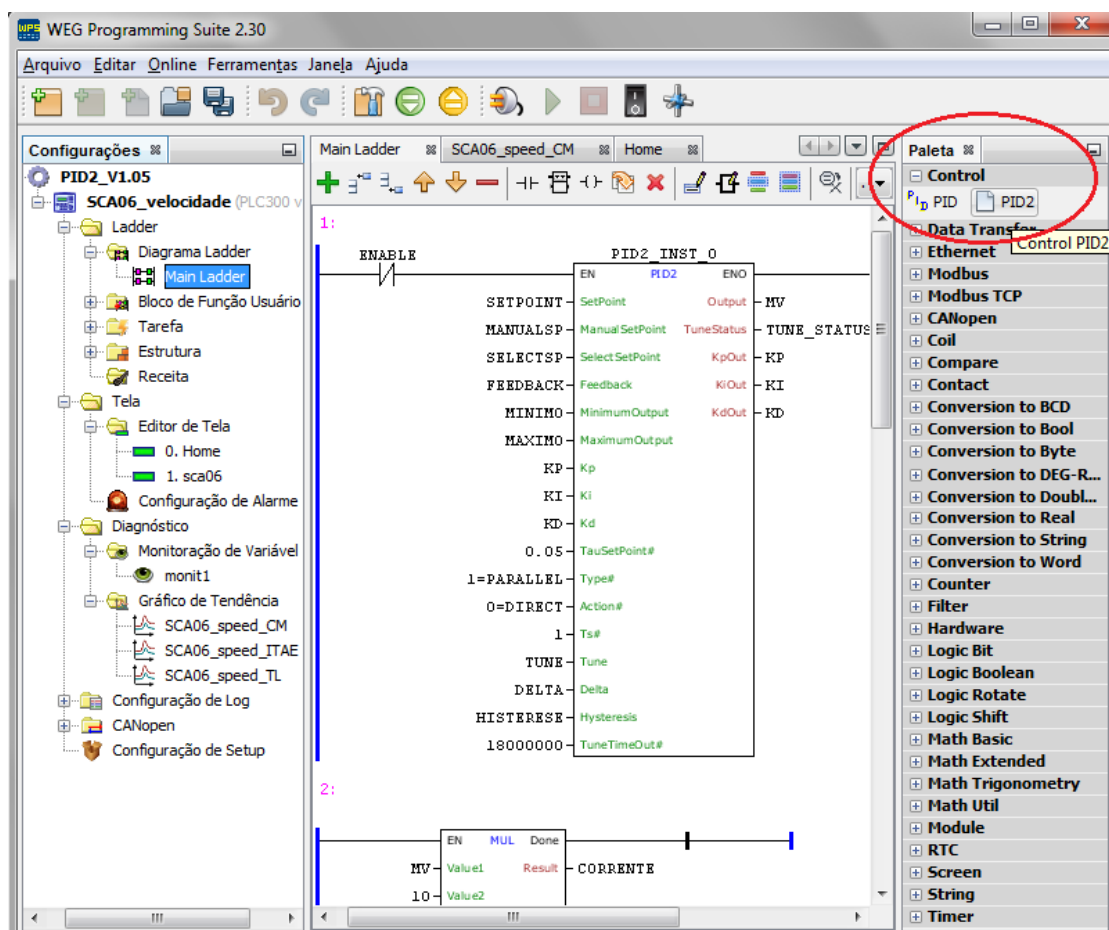
Fonte: o autor



## 5.5 Procedimento de sintonia automática do controlador

Para realizar a sintonia automática do controlador, o bloco PID2 pode ser adicionado e configurado no *software* WPS conforme exibido na Figura 22.

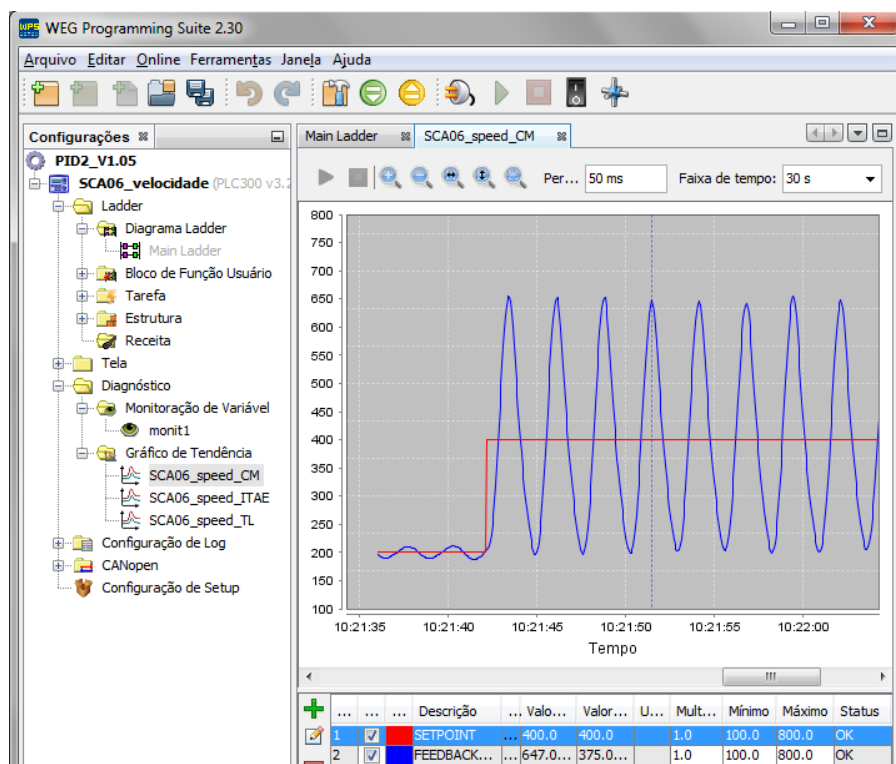
Figura 22: bloco PID2 no WPS



Fonte: o autor

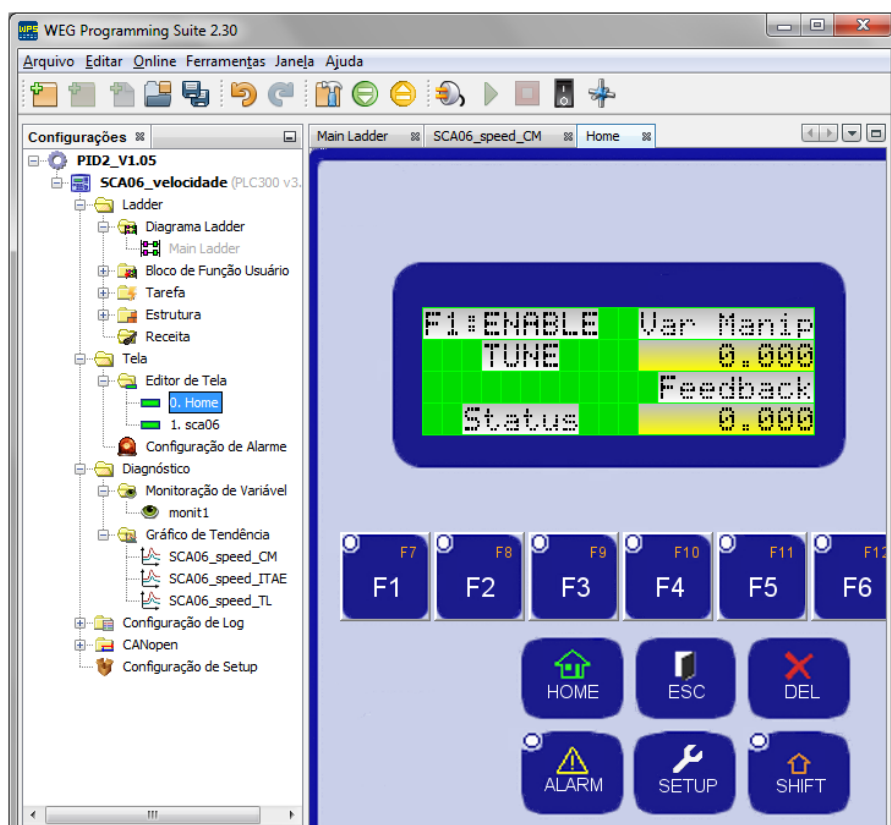
Os dados da sintonia podem ser monitorados graficamente utilizando a função *Trend* do WPS. Ela permite que um gráfico em tempo real seja efetuado com as variáveis que o usuário desejar, conforme Figura 23, no qual as variáveis *setpoint* e *feedback* são monitoradas.

Além da monitoração gráfica, o usuário pode configurar diversas telas para serem exibidas diretamente no CLP, dispensando a utilização de um computador para habilitar a função e acompanhar o status dela. Um exemplo de uma tela configurada pode ser visto na Figura 24.

Figura 23 – Função *trend* do WPS

Fonte: o autor

Figura 24 – Configuração das telas do PLC300.



### 5.5.1 Critérios para que a sintonia seja efetuada

Para que a sintonia automática possa realizar a calibração adequadamente, dois critérios precisam, obrigatoriamente, serem obedecidos:

1) colocar o sistema no ponto de operação que se deseja calibrar (a diferença deve ser menor do que 20%);

2) a variável *Output*, que é a variável manipulada, não pode estar apresentando oscilações maiores do que 20%;

OBS: pode-se usar o modo Manual do bloco para atingir o ponto de operação sem grandes oscilações (através da variável *ManualSetPoint*).

### 5.5.2 Passo a passo para executar a sintonia

Algumas etapas podem ser seguidas para realizar a calibração do sistema:

1) *Tune* deve iniciar com o valor zero;

2) Definir um valor para *Delta*. Recomenda-se um valor inicial de 10% do valor da variável manipulada (*Output*) que atingiu o *setpoint* do sistema;

3) Definir um valor inicial para *Hysteresis*. Esse valor deve ser ligeiramente maior do que o ruído presente no sistema. Pode-se utilizar um valor inicial de 2% do valor do *Setpoint* caso não se tenha conhecimento do ruído;

4) Definir um valor em milissegundos (*ms*) para *TuneTimeout*. Esse valor depende do sistema que está sendo aplicado o bloco. Sistemas mais lentos necessitarão de um tempo maior. Ele deve ser suficiente para que pelo menos 10 ciclos do relé sejam completos, embora com 5 ciclos o sistema geralmente já estabiliza;

5) Colocar o sistema no ponto de operação. Caso um pré-controle já tenha sido feito, ele pode ser aplicado. Outra opção é utilizar o modo manual e variar a entrada *ManualSetPoint* até que o sistema estabilize no ponto de operação, ou seja, *Feedback* o mais próximo possível de *SetPoint*;

6) Habilitar a sintonia escolhendo o tipo de controle (PI ou PID e a regra usada) através da variável *Tune*, conforme Tabela 8;

7) A saída *TuneStatus* irá informar qual o status do processo de sintonia automática, conforme Tabela 9. Ao final do processo, a saída *TuneStatus* receberá o valor 8 caso a calibração seja efetuada com sucesso e as saídas *KpOut*, *KiOut* e *KdOut* serão atualizadas com os valores obtidos no processo de sintonia.

## 6 RESULTADOS

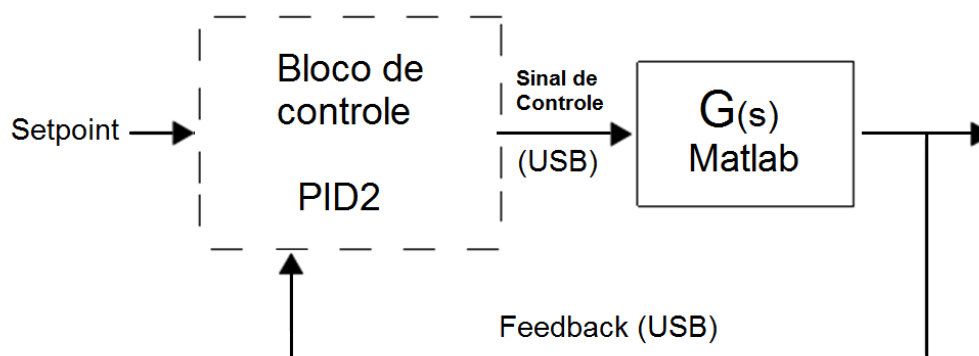
Para a coleta dos resultados e validação do bloco desenvolvido, nove plantas simuladas no Matlab e três plantas físicas reais foram escolhidas para aplicar a sintonia automática. Mais detalhes destas e os resultados são apresentados na sequência. Para as plantas simuladas e para todas as plantas reais, a realimentação do sistema era unitária, ou seja, a Variável Controlada é o próprio *Feedback* do sistema. Desta forma, sempre que *feedback* for mencionado, entende-se por ser também a Variável Controlada.

### 6.1 Plantas simuladas

Foram simuladas plantas de primeira, segunda e elevadas ordens com diversos valores do fator de controlabilidade:  $C_p < 0,1$ ,  $0,1 \leq C_p \leq 1$  e  $1 < C_p$ . As plantas  $G_1$  a  $G_6$  foram escolhidas para cobrir os diferentes valores de  $C_p$  tanto de primeira como de segunda ordem. As plantas  $G_7$ ,  $G_8$  e  $G_9$  são sugestões encontradas no trabalho de Finke (2016 apud CHIDAMBARAM e SATHE, 2014) e Yu (2007) e foram escolhidas por se assemelharem com sistemas industriais reais.

As simulações ocorreram da seguinte forma: uma comunicação modbus foi estabelecida entre o PLC300 e o Matlab através da USB; o PLC300 envia o sinal de controle que é aplicado na entrada da planta simulada, retornando o *Feedback* para o PLC; este sinal é lido pelo bloco PID2 que efetua os cálculos da nova saída de controle, seja na etapa de sintonia ou na de controle, conforme exibido na Figura 25.

Figura 25 – Diagrama de blocos para simulação das plantas



Fonte: o autor.

Após a realização da sintonia, os ganhos  $K_p$ ,  $K_i$  e  $K_d$  encontrados são atribuídos à entrada do bloco. Um degrau de amplitude 60, simbolizando 60% em uma escala de 0 a 100%, é aplicado em malha fechada para verificar como o sistema se comporta com os parâmetros calculados. Os gráficos são exibidos e interpretados na seção seguinte.

A escolha de diferentes valores para delta e histerese também foi avaliada nas plantas simuladas. A sintonia com *delta* igual a 10%, 15% e 20% e com *hysteresis* igual a 1,0% 1,5% e 2,0% foi realizado. Mesmo com estas variações, o período e o ganho críticos do sistema foram pouco influenciados pelos valores de *delta* e *hysteresis*, implicando em controladores bem semelhantes, como pode ser observado nas tabelas abaixo. Por este motivo, o resultado gráfico de apenas uma das combinações foi exibido: *Delta* = 10% e *Hysteresis* = 1%. A regra para obtenção do controlador foi escolhida automaticamente pelo bloco PID2.

As plantas simuladas apresentam o formado da equação

$$G(s) = \frac{ke^{-Ds}}{(\tau s + 1)^n} \quad (18)$$

na qual as variáveis representam (YU, 2007):

$k$ : ganho estático

$D$ : atraso de transporte

$\tau$ : constante de tempo do sistema

$n$ : ordem do sistema.

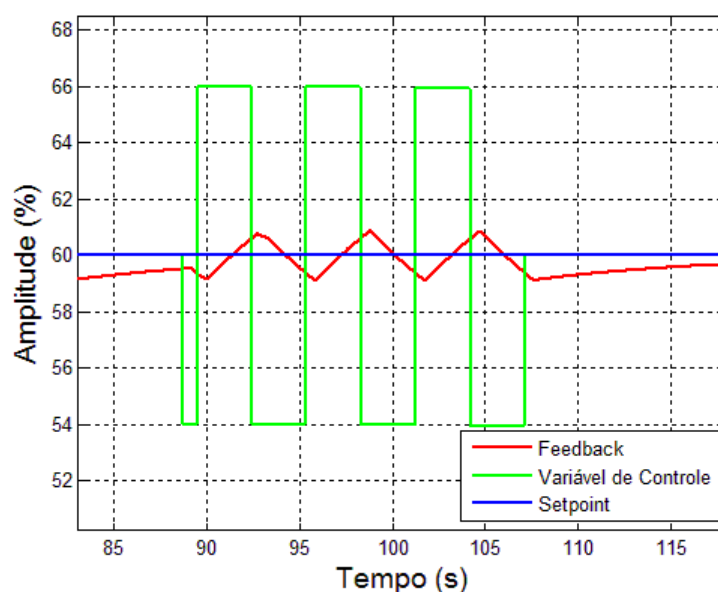
#### 6.1.1 Primeira ordem com $C_p < 0,1$

$$G_1(s) = \frac{e^{-0.5s}}{10s + 1} \quad (19)$$

A planta  $G_1$  simulada é uma planta de primeira ordem. O gráfico do resultado da aplicação da sintonia pode ser visto na Figura 26.

Primeiramente o sistema foi colocado próximo ao ponto de operação, que neste caso e nos próximos simulados representa simbolicamente 60. Em seguida, *Tune* foi habilitado e o chaveamento se iniciou. Uma oscilação pode ser observada no *Feedback* do sistema.

Figura 26 – Processo de sintonia da planta G1.



Os resultados obtidos após a simulação estão presentes na Tabela 10.

Tabela 10 – Resultados da planta G1 para controlador Tyreus-Luyben

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	10,302	9,931	0,033	5,048	0,218	7,366	3,193	0,145
15,0%	1,5%	10,342	9,917	0,033	5,068	0,219	7,384	3,206	0,146
20,0%	2,0%	10,225	9,913	0,034	5,010	0,217	7,298	3,170	0,144

Para a planta  $G_1$  o bloco PID2 optou pela regra de Tyreus-Luyben para cálculo dos parâmetros do controlador, pois  $C_p$  encontrada foi menor do que 0,1.

Os gráficos da resposta ao degrau de amplitude 60 para PID e para PI podem ser vistos na Figura 27 e Figura 28, respectivamente. Lembrando que os resultados exibidos são sempre para  $Delta = 10\%$  e  $Hysteresis = 1\%$ .

Figura 27 – Resultado do controle com PID da planta G1

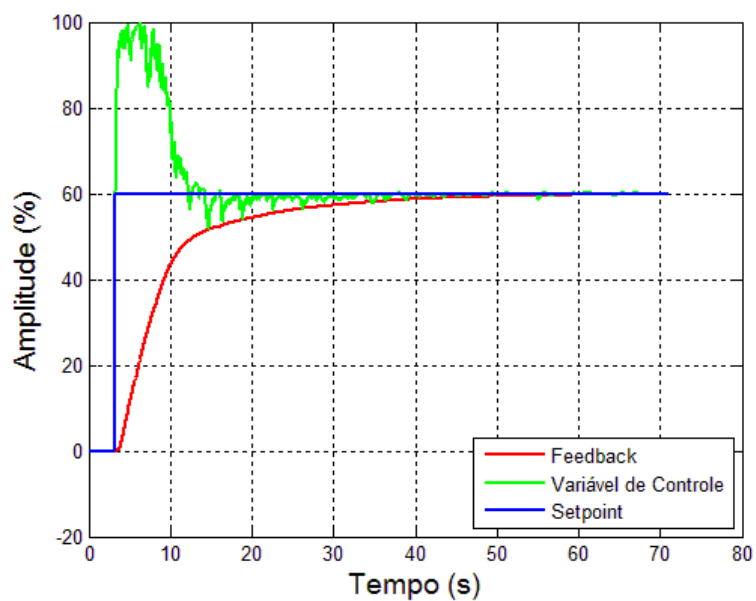
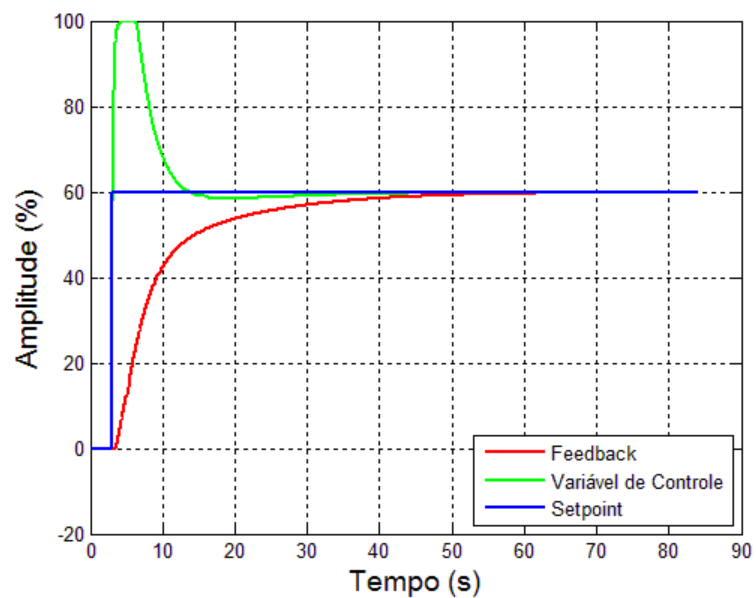


Figura 28 – Resultado do controle com PI da planta G1



Tanto para o controlador PID quanto para o PI o resultado se mostrou adequado. O *feedback* atingiu o valor do *Setpoint* sem apresentar sobressinal, mostrando que o sistema está sendo controlado com os parâmetros calculados.

O procedimento adotado para a planta  $G_1$  foi repetido para as demais plantas. Os resultados são apresentados nas subseções seguintes.

### 6.1.2 Primeira ordem com $0,1 \leq C_p \leq 1$

$$G_2(s) = \frac{e^{-1s}}{2s + 1} \quad (20)$$

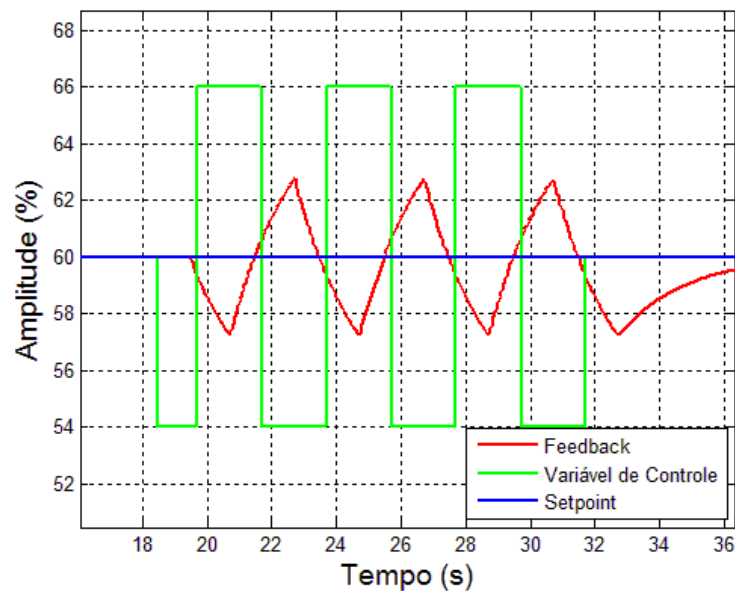
A planta  $G_2$  possui certo atraso de transporte resultando em um fator de controlabilidade entre 0,1 e 1. Após simulada e aplicada a sintonia, os resultados obtidos foram os exibidos na Tabela 11.

Tabela 11 – Resultados da planta G2 para controlador ITAE

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	2,770	4,022	0,688	1,363	0,576	0,476	0,850	0,474

O gráfico da simulação é exibido na Figura 29.

Figura 29 – Processo de sintonia da planta G2.



A regra para obtenção do controlador desta vez foi a do ITAE mínimo. O resultado da resposta ao degrau 60% pode ser visto na Figura 30 e na Figura 31.



Figura 30 – Resultado do controle com PID da planta G2

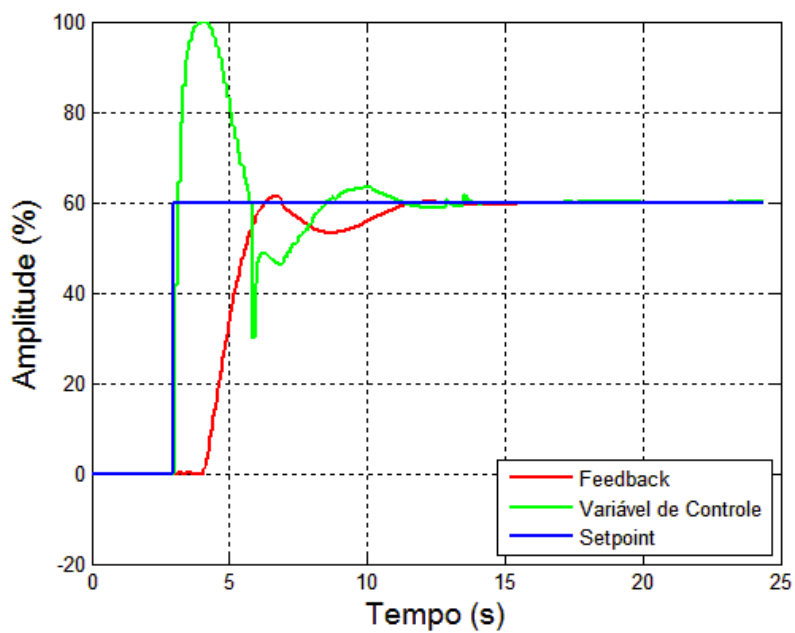
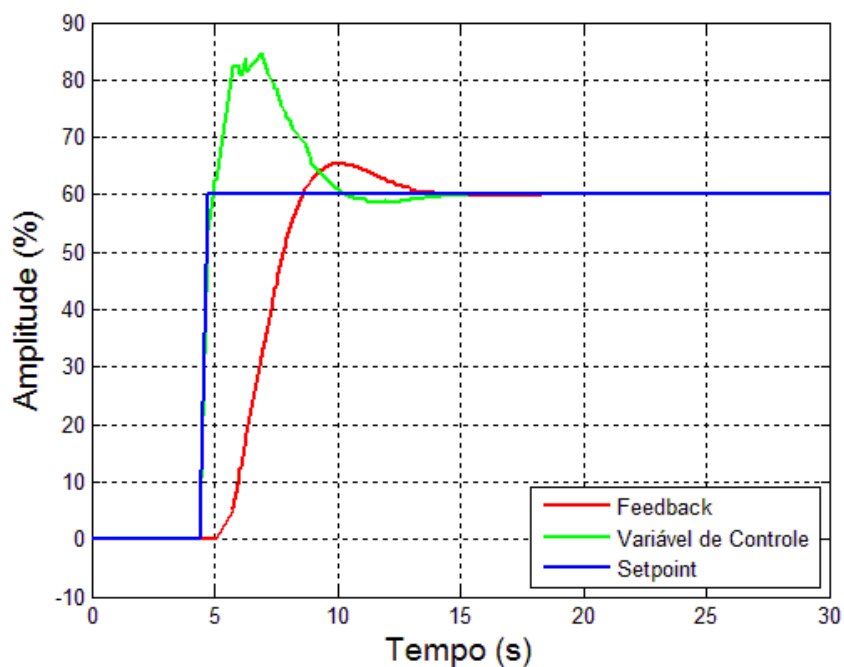


Figura 31 – Resultado do controle com PI da planta G2



O sistema foi controlado para ambos os controladores sendo que o tempo de subida foi de aproximadamente 4s e de acomodação de 10s. O controlador PID

apresentou uma pequena oscilação e o PI um sobressinal de 8%. Para melhorar esse sistema de controle, uma alternativa é diminuir o ganho proporcional.

### 6.1.3 Primeira ordem com $1 < C_p$ :

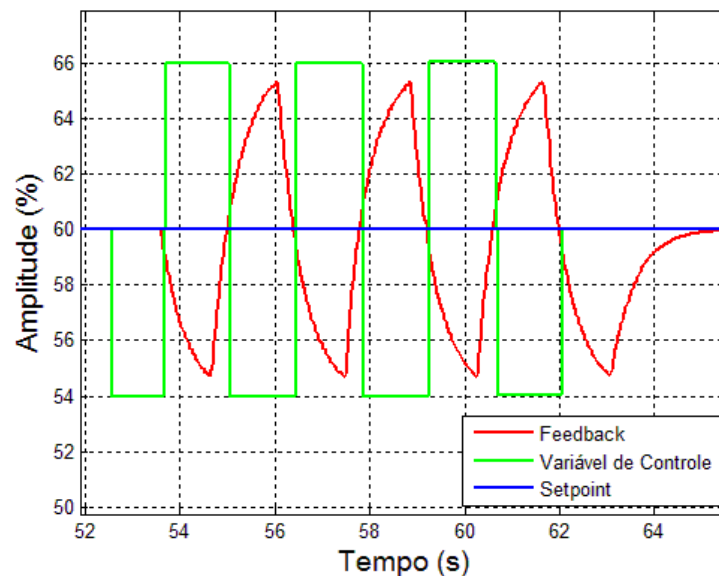
$$G_3(s) = \frac{e^{-1s}}{0.5s + 1} \quad (21)$$

A planta  $G_3$  possui grande atraso de transporte considerando-se a constante de tempo do sistema que é baixa, o que resulta em um fator de controlabilidade maior do que um. Os resultados da simulação da planta  $G_3$  podem ser vistos na Tabela 12 e na Figura 32.

Tabela 12 – Resultados da planta G3 para controlador Ciancone-Marlin

Configuração		Dados coletados			PID			PI	
Delta	Histerese	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	1,434	2,806	2,227	0,674	0,664	0,149	0,430	0,613
15,0%	1,5%	1,437	2,803	2,216	0,675	0,666	0,149	0,431	0,615
20,0%	2,0%	1,457	2,795	2,162	0,685	0,678	0,151	0,437	0,626

Figura 32 – Processo de sintonia da planta G3



Com a aplicação do controlador obtido pela regra de Ciancone-Marlin, o resultado da resposta ao degrau 60% foi o obtido na Figura 33 para controlador PID e na Figura 34 para controlador PI.

Figura 33 – Resultado do controle com PID da planta G3

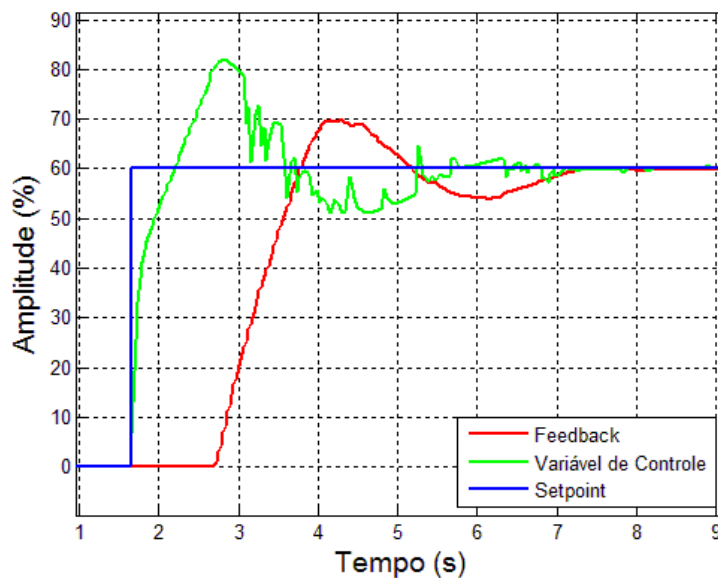
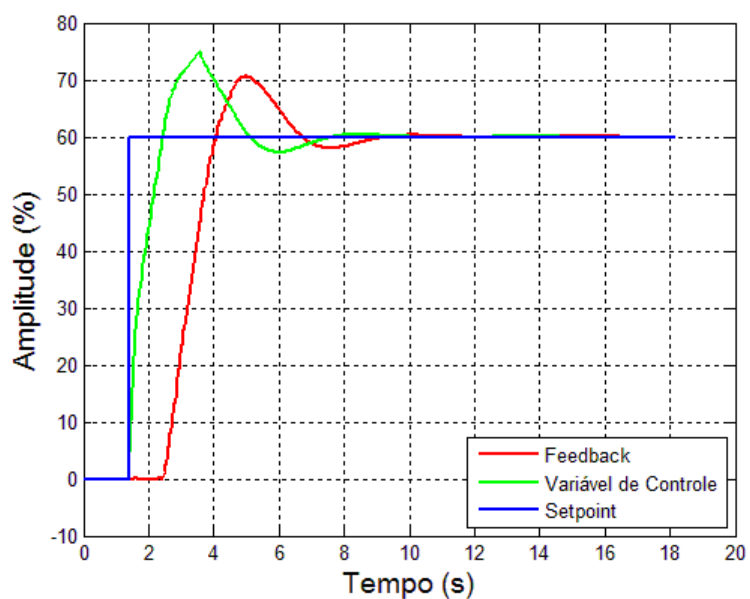


Figura 34 – Resultado do controle com PI da planta G3



Em ambos os casos, PI e PID, o sistema foi controlado, apresentou um sobressinal de aproximadamente 18% e estabilizou na sequência. Nestes resultados apresentados pode-se observar o atraso de transporte logo após a aplicação do degrau.

6.1.4 Segunda ordem com  $C_p < 0,1$ :

$$G_4(s) = \frac{e^{-0.5s}}{(10s + 1)^2} \quad (22)$$

A planta G4 é uma planta de segunda ordem. Os resultados obtidos após a simulação são os exibidos na Tabela 13. Como o valor da  $C_p$  obtida pelo algoritmo ficou entre 0 e 1, o método escolhido automaticamente foi o de ITAE mínimo.

Tabela 13 – Resultados da planta G4 para controlador ITAE

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	6,981	25,483	0,122	5,769	0,160	7,051	4,026	0,145

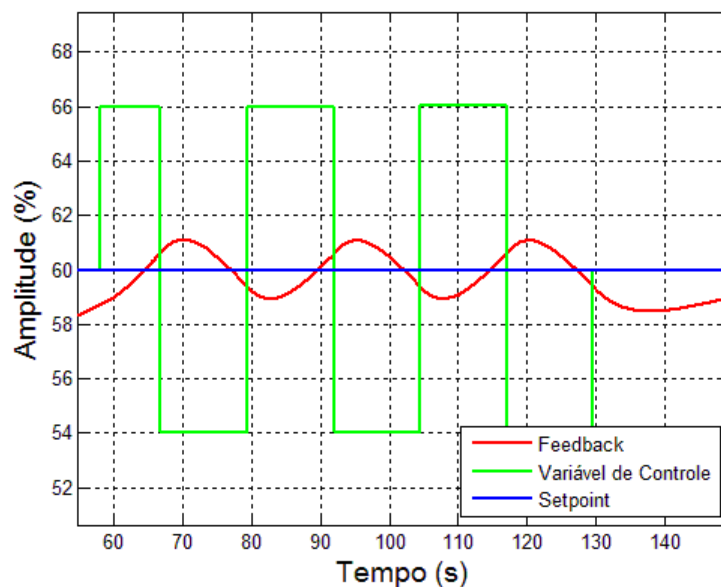
Como o valor de  $C_p$  ficou próximo de 0,1, optou-se por avaliar também os resultados para o controlador Tyreus-Luyben. Os resultados são apresentados na Tabela 14.

Tabela 14 – Resultados da planta G4 para controlador Tyreus-Luyben

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	6,981	25,483	0,122	3,421	0,058	12,809	2,164	0,038

O gráfico do processo de sintonia, que é o mesmo, independente da regra escolhida, pode ser observado na Figura 35.

Figura 35 – Processo de sintonia da planta G4



A Figura 36 e a Figura 37 mostram a resposta ao degrau 60% utilizando o controlador pela regra ITAE. Tanto o controlador PID quanto o PI foram capazes de controlar a planta sem apresentar sobressinal. Apesar disso, ambos apresentaram pequena oscilação, que pode ser corrigida diminuindo o ganho proporcional do sistema.

Figura 36 – Resultado com controlador PID da planta G4 (ITAE)

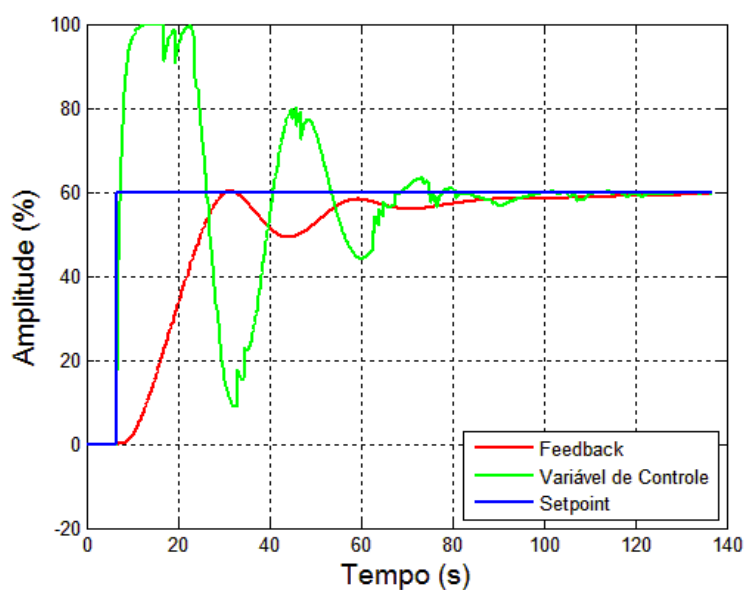
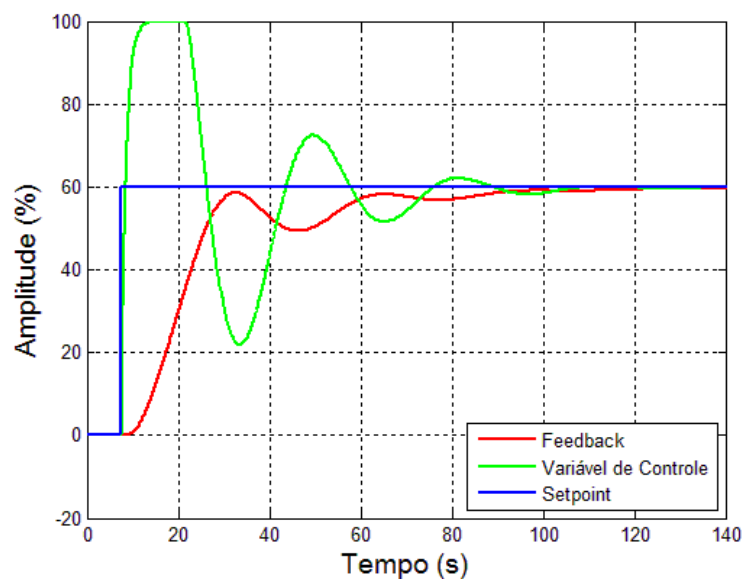


Figura 37 – Resultado com controlador PI da planta G4 (ITAE)



A Figura 38 e a Figura 39 exibem o resultado para a regra de Tyreus-Luyben para o controlador PID e PI, respectivamente. Apesar de apresentar um pequeno sobressinal de 8%, o sistema estabilizou em um tempo bem menor do que o da regra do ITAE Mínimo, em aproximadamente 50s.

Figura 38– Resultado com controlador PID da planta G4 (Tyreus-Luyben)

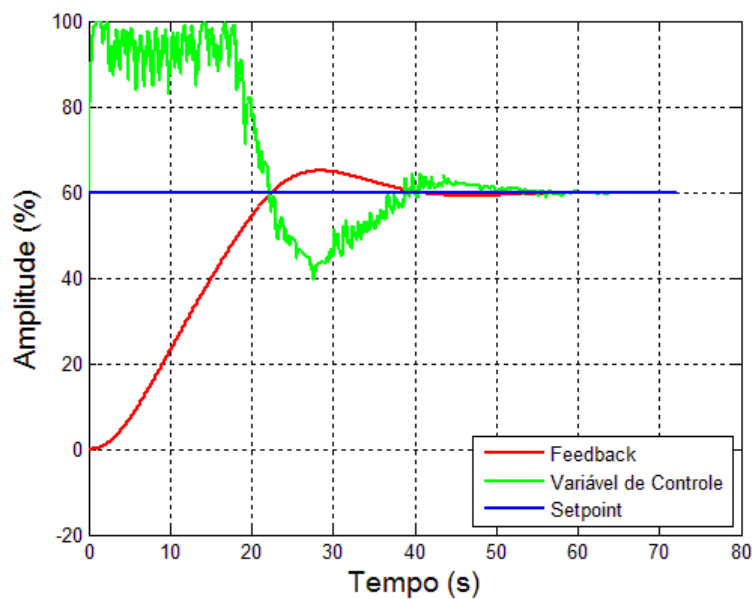
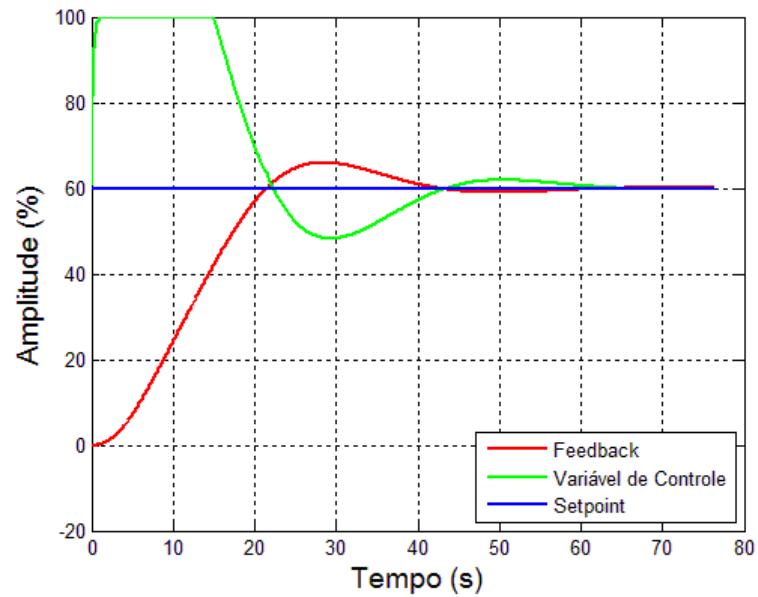


Figura 39 – Resultado com controlador PI da planta G4 (Tyreus-Luyben)



6.1.5 Segunda ordem com  $0,1 \leq C_p \leq 1$ :

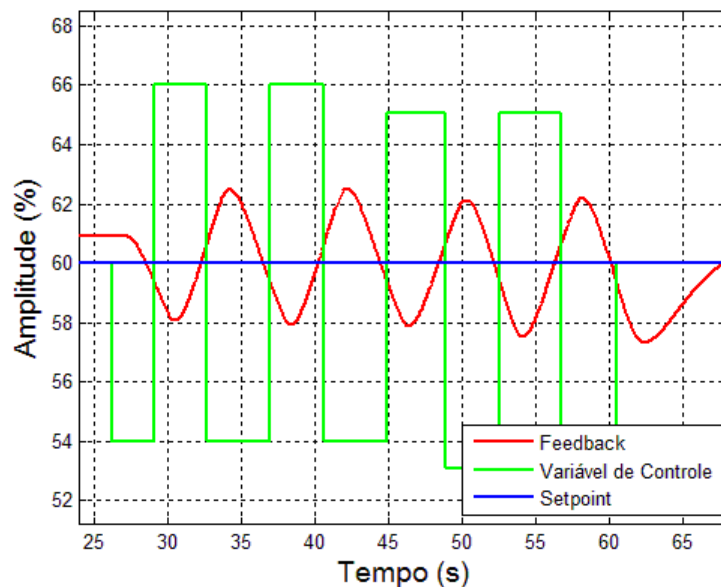
$$G_5(s) = \frac{e^{-1s}}{(2s + 1)^2} \quad (23)$$

A planta de segunda ordem com atraso de transporte simulada teve os resultados exibidos na Tabela 15 e na Figura 40.

Tabela 15 – Resultados da planta G5 para controlador ITAE

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	3,291	8,223	0,506	1,697	0,294	1,158	1,078	0,245

Figura 40 – Processo de sintonia da planta G5



Com a aplicação do controlador para a regra do ITAE, o sistema apresentou pequena oscilação, mas estabilizou na sequência. O tempo de assentamento em ambos os casos foi de 25s, conforme pode ser visto na Figura 41 e Figura 42.

Figura 41 – Resultado com controlador PID da planta G5

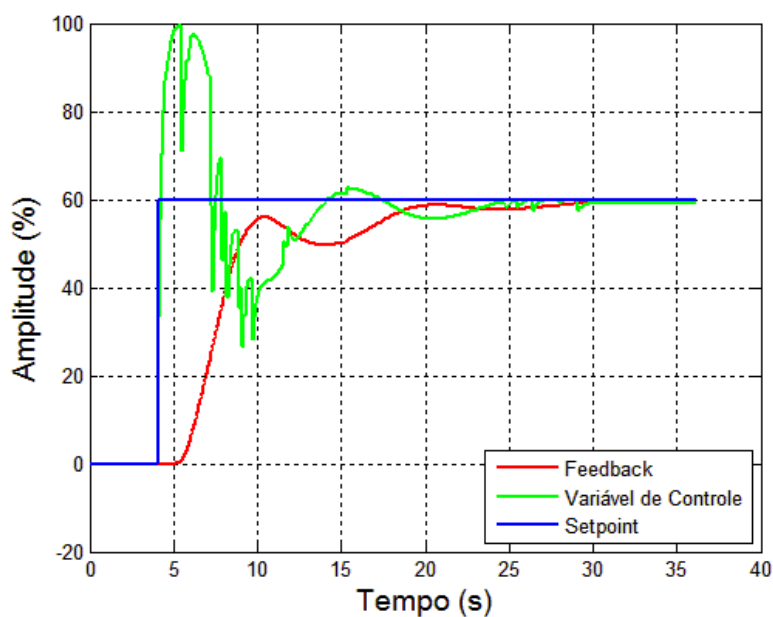
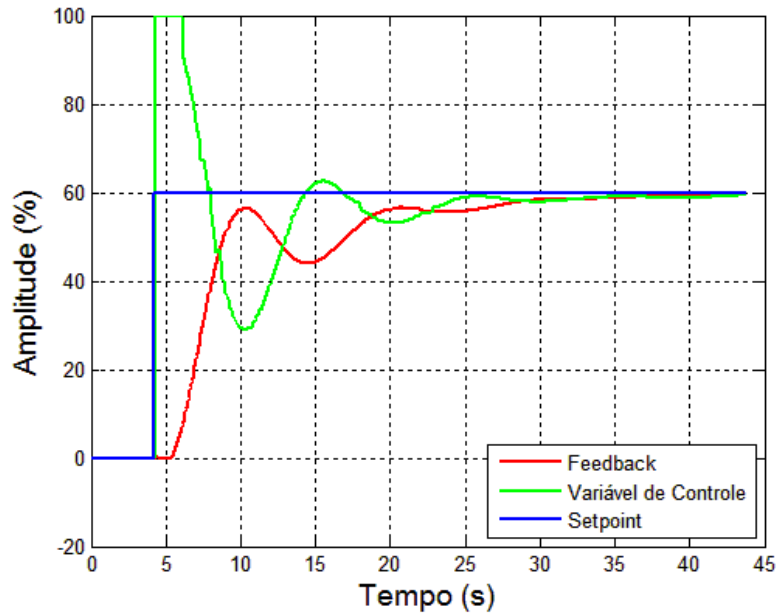




Figura 42 – Resultado com controlador PI da planta G5

6.1.6 Segunda ordem com  $1 < C_p$ :

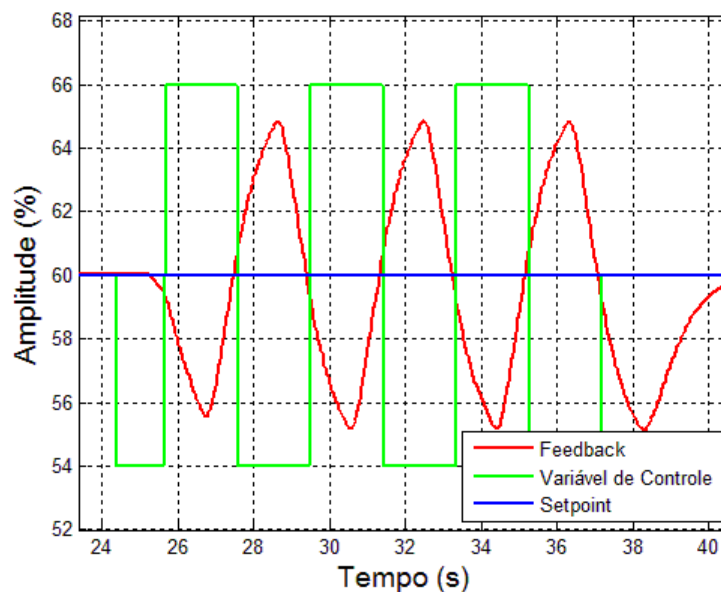
$$G_6(s) = \frac{e^{-1s}}{(0.5s + 1)^2} \quad (24)$$

A planta G6 simulada é com atraso de transporte dominante. Apesar de ter o mesmo valor de atraso de transporte da planta anterior, ele se torna mais expressivo uma vez que esta planta possui uma constante de tempo menor. Os resultados da sintonia são exibidos na Tabela 16 e Figura 43.

Tabela 16 – Resultados da planta G6 para controlador Ciancone-Marlin

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	1,580	3,838	1,438	0,743	0,535	0,224	0,474	0,494
15,0%	1,5%	1,579	3,844	1,433	0,742	0,534	0,225	0,474	0,493
20,0%	2,0%	1,580	3,846	1,463	0,743	0,534	0,225	0,474	0,493

Figura 43 – Processo de sintonia da planta G6



Os resultados da planta G6 para controlador PID e PI se mostraram satisfatórios uma vez que controlaram o sistema. O resultado não apresentou grandes oscilações, apenas um sobressinal que ficou abaixo dos 20%, conforme Figura 44 e Figura 45. Para diminuir esse sobressinal, pode-se diminuir o ganho proporcional.

Figura 44 – Resultado com controlador PID da planta G6

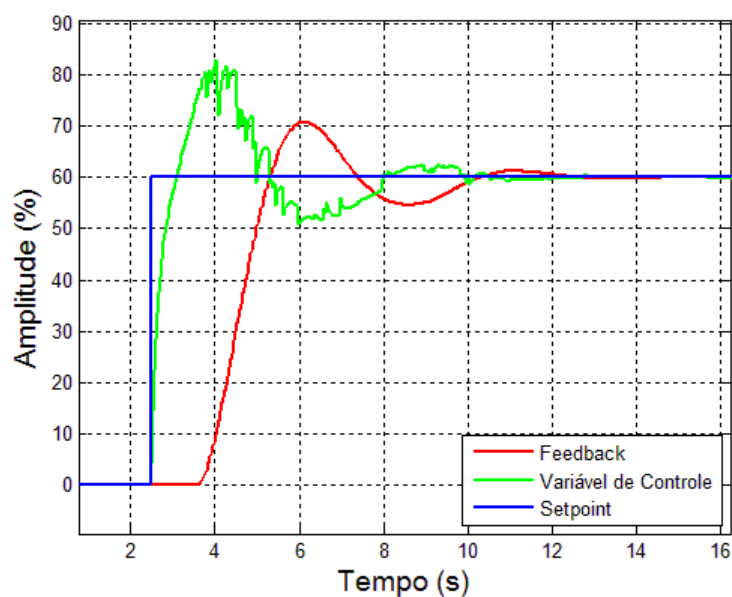
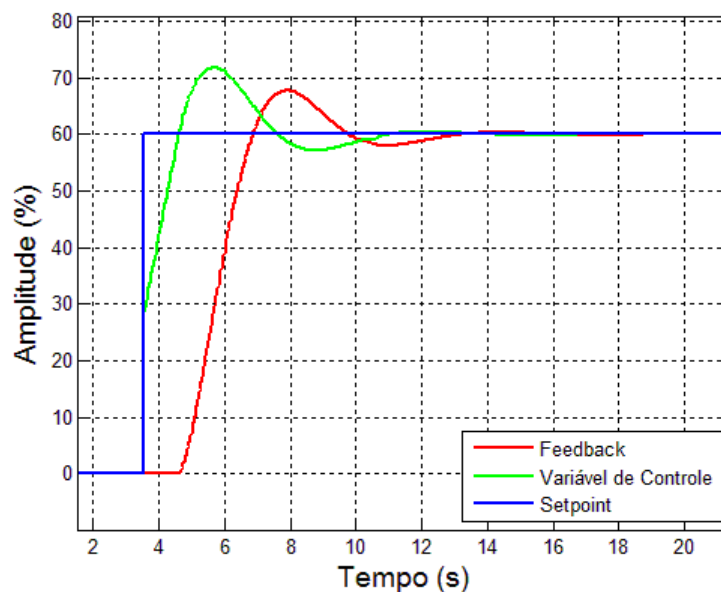


Figura 45 – Resultado com controlador PI da planta G6



## 6.1.7 Planta G7

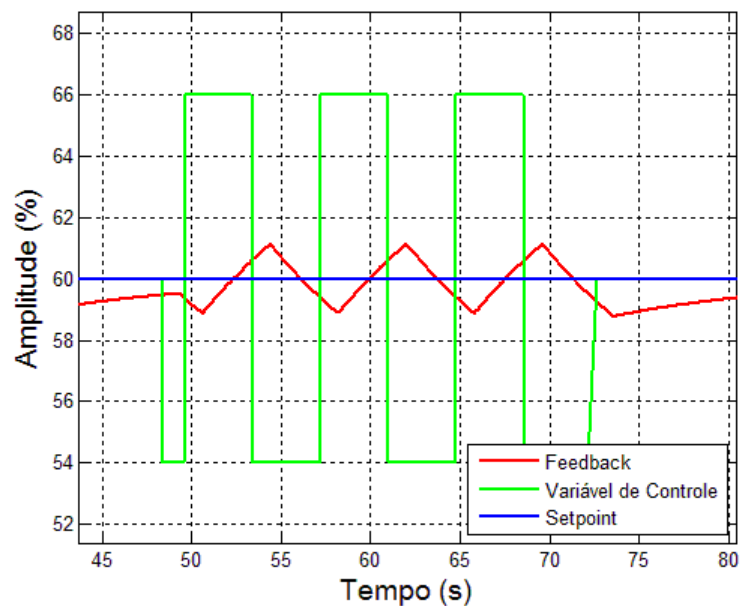
$$G_7(s) = \frac{e^{-s}}{10s + 1} \quad (25)$$

A planta  $G_7$  simulada foi apresentada no trabalho de Yu (2007). Ela é de primeira ordem com fator de controlabilidade entre 0,1 e 1, semelhante à planta  $G_2$ , porém com constante de tempo cinco vezes maior. Os resultados obtidos com a sintonia automática podem ser vistos Tabela 17 e na Figura 46.

Tabela 17 – Resultados da planta G7 para controlador ITAE

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	6,806	7,621	0,127	6,522	0,775	0,345	4,477	1,089
15,0%	1,5%	6,793	7,622	0,128	6,492	0,772	0,344	4,458	1,083
20,0%	2,0%	6,816	7,601	0,128	6,487	0,771	0,345	4,454	1,082

Figura 46 – Processo de sintonia da planta G7



Com a aplicação de um degrau de amplitude 60 no sistema controlado utilizando os valores obtidos na simulação, o resultado para controlador PID é mostrado Figura 47 e para PI na Figura 48.

Figura 47 – Resultado com controlador PID da planta G7

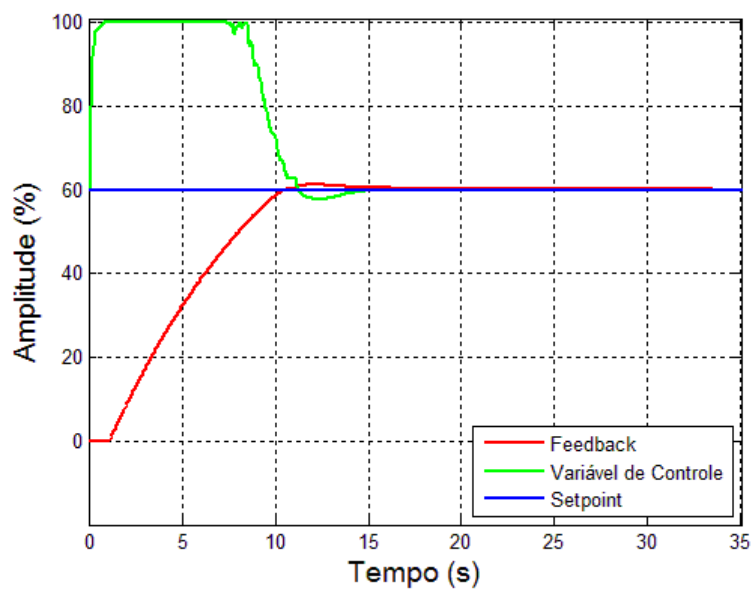
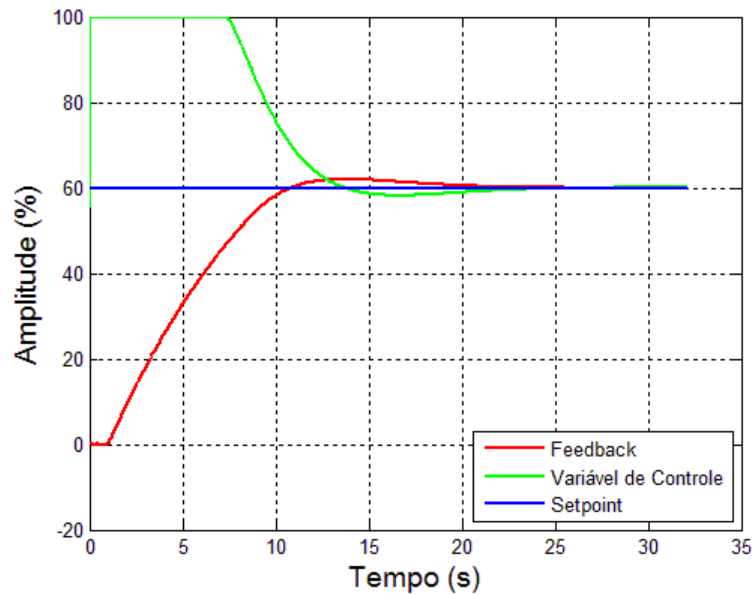


Figura 48 – Resultado com controlador PI



Ambos os controladores PID e PI pela regra ITAE Mínimo se mostraram coerentes e controlaram a planta. Um pequeno sobressinal ocorreu, mas que pode ser eliminado com um ajuste mais fino dos ganhos, caso necessário. O controlador PID estabilizou um pouco antes do que o PI.

#### 6.1.8 Terceira ordem com $1 < C_p$ :

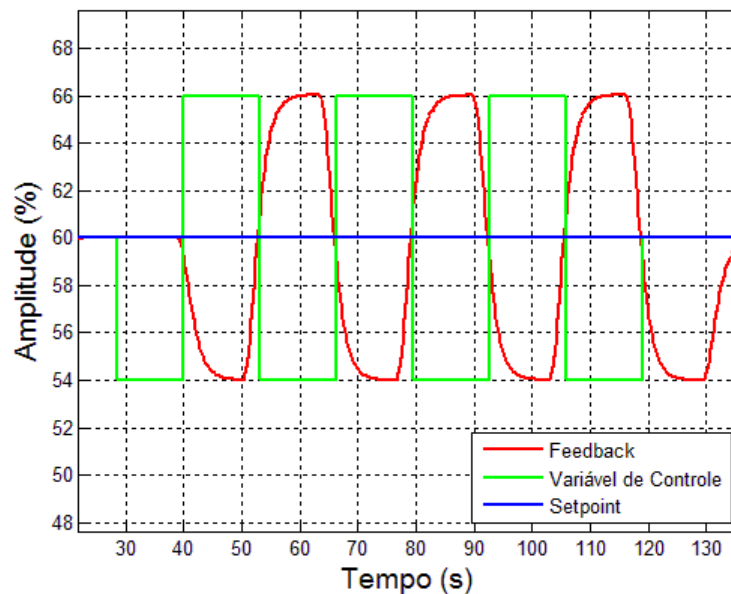
$$G_8(s) = \frac{e^{-10s}}{(s + 1)^3} \quad (26)$$

A planta  $G_8$  simulada é uma planta de terceira ordem. Os resultados obtidos após o término da sintonia são exibidos na Tabela 18 e na Figura 49.

Tabela 18 – Resultados da planta G8 para controlador Ciancone-Marlin

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	1,272	26,398	3,144	0,598	0,063	1,242	0,382	0,058
15,0%	1,5%	1,273	26,411	3,127	0,598	0,063	1,244	0,382	0,058
20,0%	2,0%	1,273	26,401	3,128	0,598	0,063	1,243	0,382	0,058

Figura 49 – Processo de sintonia da planta G8



Com a aplicação do controlador PID e PI, pela regra de Ciancone-Marlin, obteve-se a resposta exibida na Figura 50 e Figura 51 respectivamente.

Figura 50 – Resultado com PID da planta G8

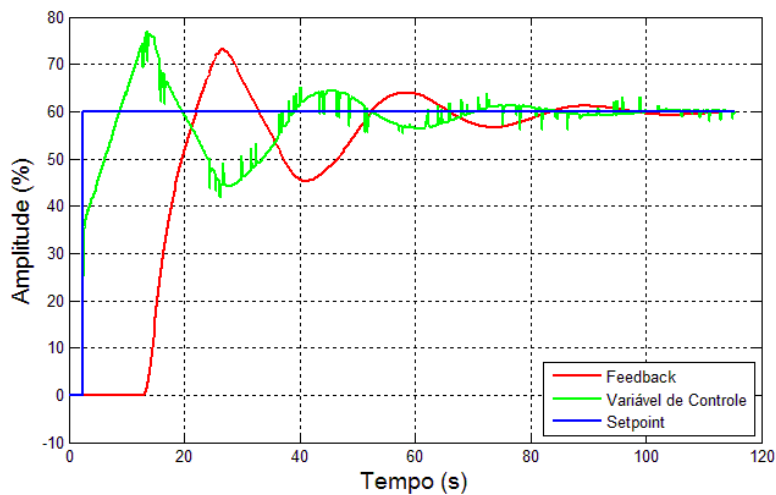
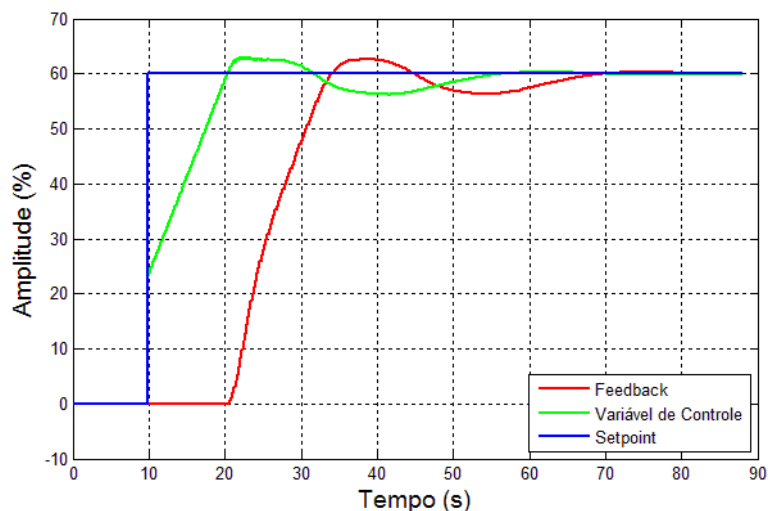


Figura 51 – Resultado com PI da planta G8



Os resultados apresentaram certa oscilação, mas com ambos os controladores o sistema estabilizou após um tempo. Com o controlador PID, o sobressinal foi de aproximadamente 20% e o tempo de assentamento de 100s. O controlador PI apresentou um sobressinal de cerca de 5% e tempo de assentamento de 60s, ambos menores do que o controlador PID.

Pode-se observar graficamente o atraso de transporte de 10s após a aplicação do degrau.

#### 6.1.9 Planta de oitava ordem

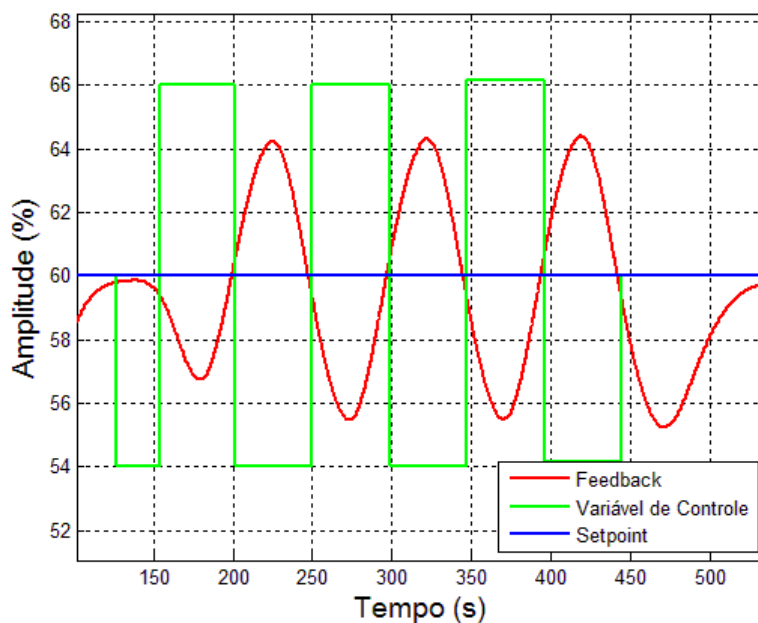
$$G_9(s) = \frac{e^{-0.6s}}{(6s + 1)^8} \quad (27)$$

A planta G9 simulada é uma planta de ordem elevada, oitava ordem. Os resultados da simulação são exibidos na Tabela 19 e na Figura 52.

Tabela 19 – Resultados da planta G9 para controlador Ciancone-Marlin

Configuração		Dados coletados			PID			PI	
Delta	Histerese	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
10,0%	1,0%	1,724	97,405	1,067	0,810	0,023	6,213	0,517	0,021
15,0%	1,5%	1,728	97,399	1,060	0,812	0,023	6,227	0,518	0,021
20,0%	2,0%	1,729	97,194	1,059	0,812	0,023	6,216	0,519	0,021

Figura 52 – Processo de sintonia da planta G9



Os resultados obtidos com o controlador PID e PI aplicados na planta G9 através da regra de Ciancone-Marlin são exibidos na Figura 53 e na Figura 54.

Figura 53 – Resultado com PID da planta G9

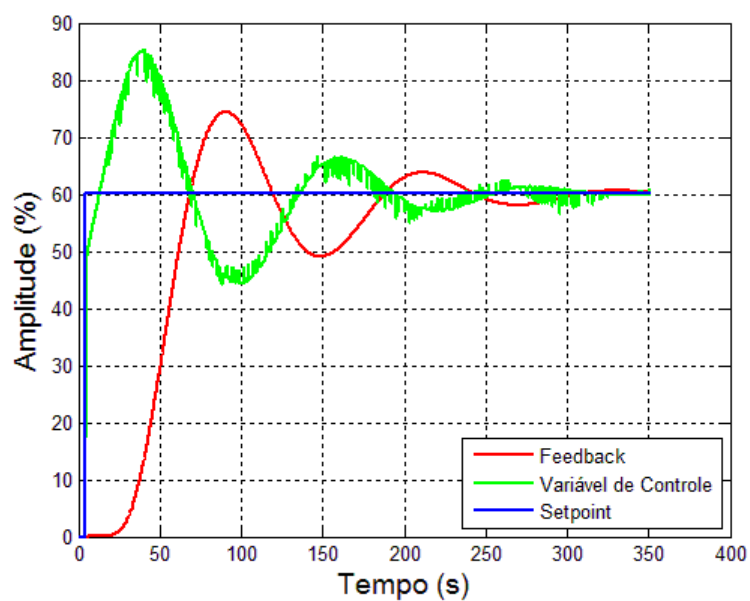
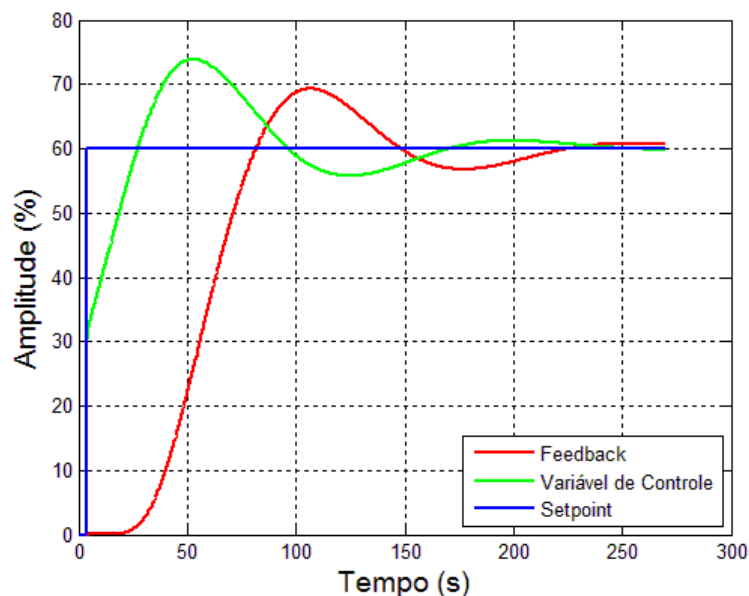




Figura 54 – Resultado PI da planta G9



O controlador PID se mostrou um pouco mais oscilatório do que o PI, porém ambos realizaram o controle do sistema, estabilizando após algum tempo. O controlador PID apresentou um sobressinal de aproximadamente 25% enquanto o do PI foi menor, aproximadamente 18%.

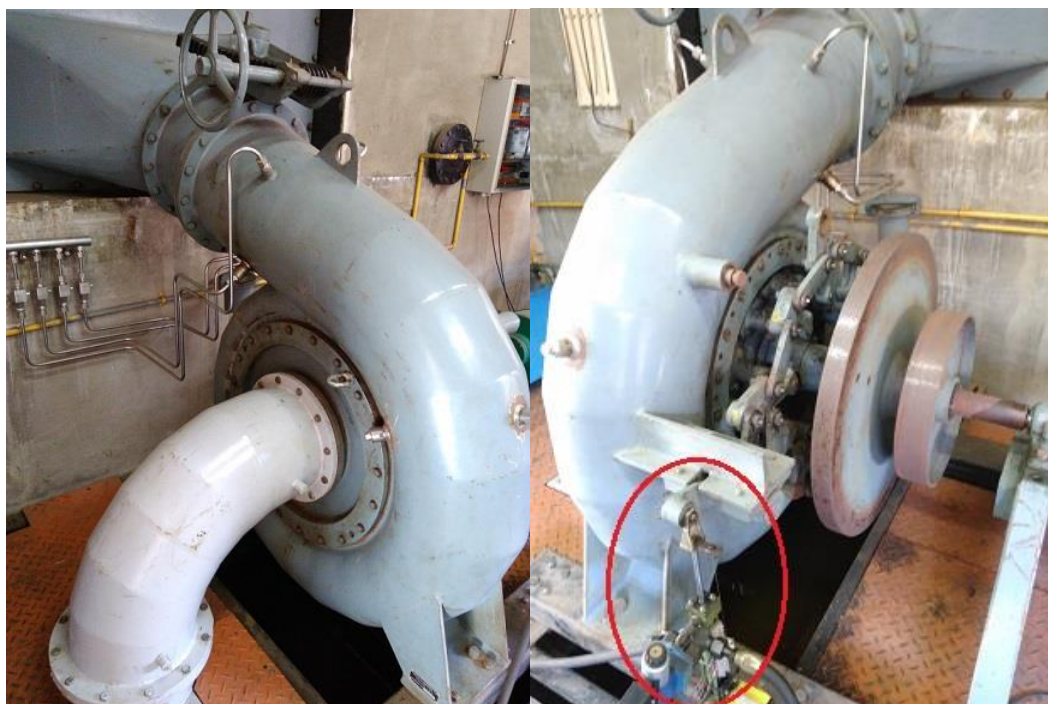
## 6.2 Testes práticos com plantas reais

Além das plantas simuladas em laboratório, a sintonia automática foi aplicada também em três plantas físicas reais. A primeira é uma turbina hidroelétrica do tipo Francis, onde é feito o controle da abertura de uma válvula, que controla o fluxo de água da turbina, através de um sinal de corrente fornecido por uma saída controlada do PLC300. A segunda foi o controle de velocidade de um servomotor e por fim o controle de temperatura de uma câmara térmica. A seguir os detalhes e resultados de cada uma delas.

### 6.2.1 Turbina hidroelétrica Francis.

A empresa Hisa, pertencente ao grupo WEG, possui um laboratório com uma turbina do tipo Francis que foi utilizada para alguns ensaios e coleta de resultados do bloco PID2. A Figura 55 mostra a imagem da turbina utilizada para os ensaios.

Figura 55 – Turbina Francis com detalhe do pistão que abre a válvula



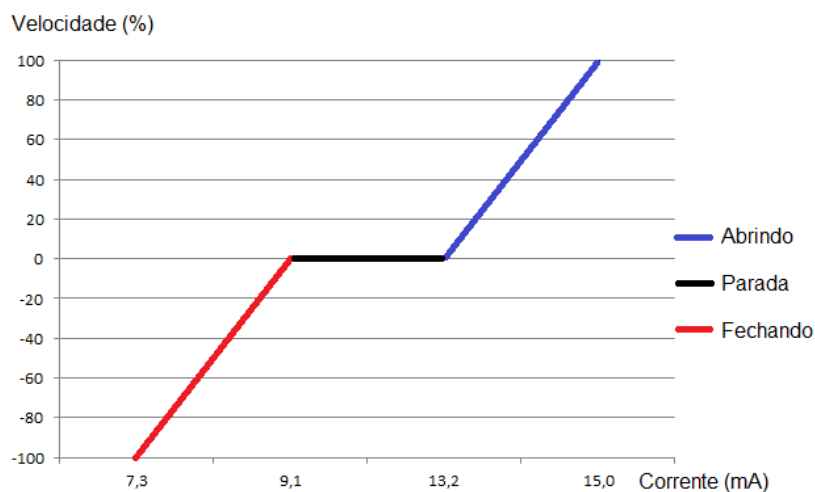
As turbinas Francis são adequadas para operar em quedas de 10m até 200m. Este modelo de turbina é o que permite sua instalação em maior faixa de aplicação relacionado à queda/vazão. São máquinas projetadas para operar tanto com eixo vertical quanto horizontal ([www.hisa.com.br/produtos/turbina-francis](http://www.hisa.com.br/produtos/turbina-francis)).

A velocidade de rotação da turbina e a potência por ela gerada dependem diretamente do fluxo de água pela mesma. O fluxo de água é controlado através da abertura e fechamento de uma válvula. Um dos desafios de controle desta turbina é na malha de posição da abertura desta válvula através de uma corrente enviada ao atuador. Esta foi a malha de controle ajustada para uma das validações do PID2 do PLC300.

O pistão em destaque na Figura 55 recebe um sinal de corrente e atua na abertura e fechamento da turbina. Com um sinal de corrente entre 9,10mA e 7,30mA a válvula começa a abrir (quanto menor o valor maior a velocidade de abertura) e entre 13,20 e 15,0mA a válvula começa a fechar (quanto maior o valor maior a velocidade de fechamento).

A Figura 56 demonstra graficamente o comportamento.

Figura 56 – Velocidade de abertura da tampa em função da corrente no atuador



Fonte: o autor

Para facilitar a aplicação do controlador e diminuir oscilações, uma lógica foi implementada no software aplicativo (em *ladder*) do sistema de controle para eliminar a “zona morta” entre 9,10mA e 13,20mA, ou seja, caso o valor esteja entre esses dois valores, a saída analógica recebe o valor 11,15mA.

O *feedback* do sistema é a posição da válvula que mede sua abertura através de um encoder. Um transdutor transforma a informação do encoder em um sinal de corrente que varia de 14,60 mA para completamente fechado (0%) e 9,15 mA para completamente aberto (100%).

Para a aplicação da sintonia os valores das correntes foram convertidos no *ladder* para PU, desta forma, o atuador varia de 0 à -100 fechando a válvula e de 0 à 100 abrindo a válvula. O *feedback*, por sua vez, varia de 0 à 100, onde 0 a válvula está completamente fechada e 100 completamente aberta.

Para o ensaio da turbina, utilizou-se um valor de 20% para Delta e 1% para histerese.

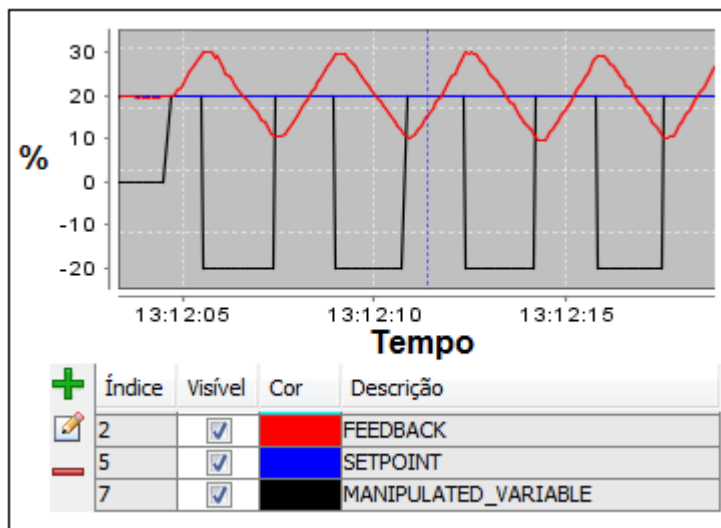
Os valores obtidos após a aplicação do método dos relés podem ser vistos na Tabela 20.

Tabela 20 – Resultados da turbina para controlador Tyreus-Luyben

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
20,0%	1,0%	2,966	0,652	0,004	1,453	0,955	0,139	0,919	0,637

A Figura 57 mostra a aplicação do método dos relés na turbina com um delta de 20% e uma histerese de 1%. Para melhor visualização, a curva do *Feedback* está em uma escala de 10 para 1.

Figura 57– Curva do relé para a turbina



Após a obtenção dos parâmetros do controlador e habilitados os mesmos para operação em modo automático, alguns degraus positivos de posição foram aplicados para verificar a abertura da válvula: de 20 para 40, de 40 para 60 e de 60 para 90 por cento. O resultado pode ser visualizado na Figura 58. Em seguida, um degrau negativo também foi aplicado para verificar o comportamento durante o fechamento da válvula. O resultado pode ser visto na Figura 59.

Os resultados obtidos foram satisfatórios tanto na abertura quanto no fechamento da válvula. O *setpoint* foi atingido rapidamente e o sistema apresentou um pequeno sobressinal. Uma pequena oscilação pode ser observada devido ao ruído do sistema. Ela pode ser facilmente eliminada com a adição de um filtro passa baixa.

Figura 58– Resposta ao degrau de abertura da turbina.

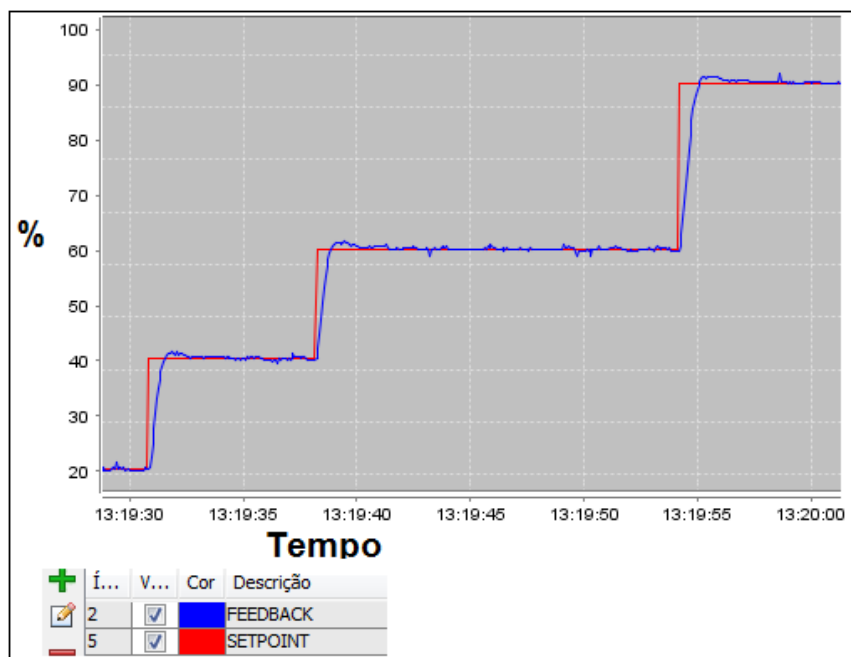
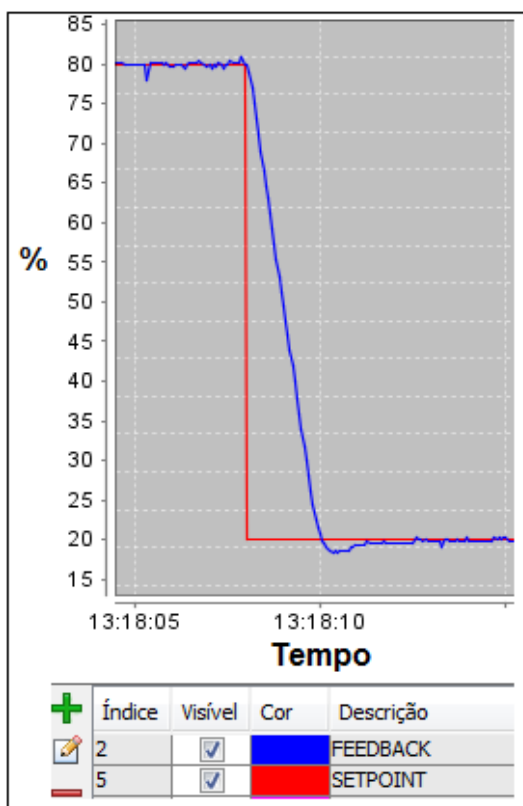


Figura 59 – Resposta ao degrau de fechamento da turbina

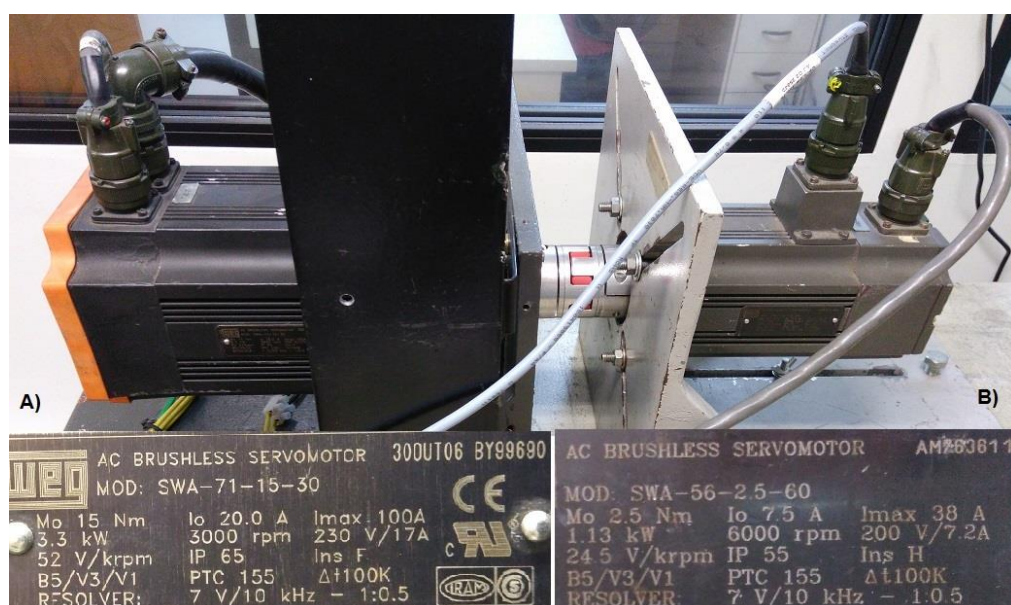


## 6.2.2 Servomotor

Um segundo teste prático realizado foi o de controle de velocidade de um servomotor, o qual é mostrado na Figura 60(B), conectado ao servoconversor SCA06 da WEG. Neste sistema, o PLC300 recebe a velocidade do servomotor (*feedback*) via rede CANopen, aplica a malha de controle, ou a sintonia, e envia para o servoconversor uma referência de corrente que alimentará o servomotor.

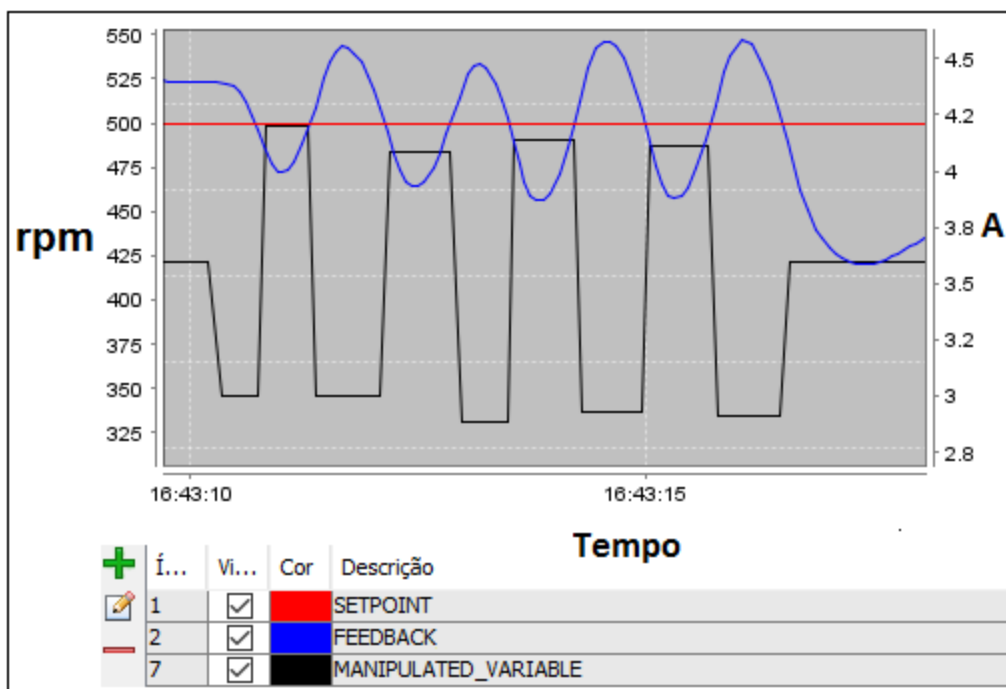
O servomotor de maior potência aplica uma carga constante de 0,5Nm no menor. Eles estão com os eixos acoplados, conforme Figura 60.

Figura 60 – Servomotores utilizados para controle de velocidade.



Depois de feitos os ajustes da comunicação entre o PLC300 e o servoconversor, definiu-se o valor de 20% para delta e 1% para histerese. O ponto de operação para este experimento foi de 500rpm. A referência de corrente enviada ao servomotor foi ajustada manualmente até o sistema atingir o ponto de operação. Na sequência a sintonia automática foi habilitada. O resultado pode ser visto na Figura 61.

Figura 61 – Curva do relé para o servomotor



Os resultados obtidos após o término da sintonia estão na Tabela 21. Ela apresenta também os ganhos do controlador pelo método ITAE.

Tabela 21 – Resultados do servomotor para controlador ITAE

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
20,0%	1,0%	0,020	1,425	0,375	0,016	0,021	0,002	0,011	0,024

Alguns degraus de velocidade foram aplicados para verificar como o sistema se comportou após a sintonia automática dos ganhos. O resultado para o controlador PID está na Figura 62 e para o controlador PI na Figura 63.

Figura 62 – Resposta ao degrau com controlador PID(ITAE)

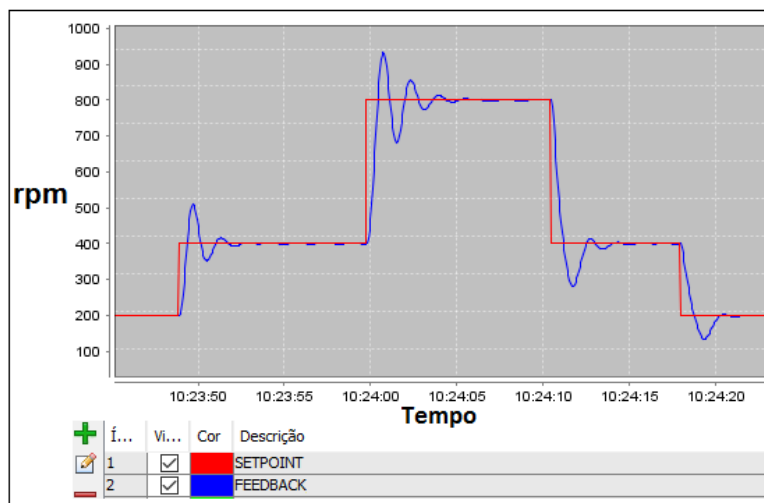
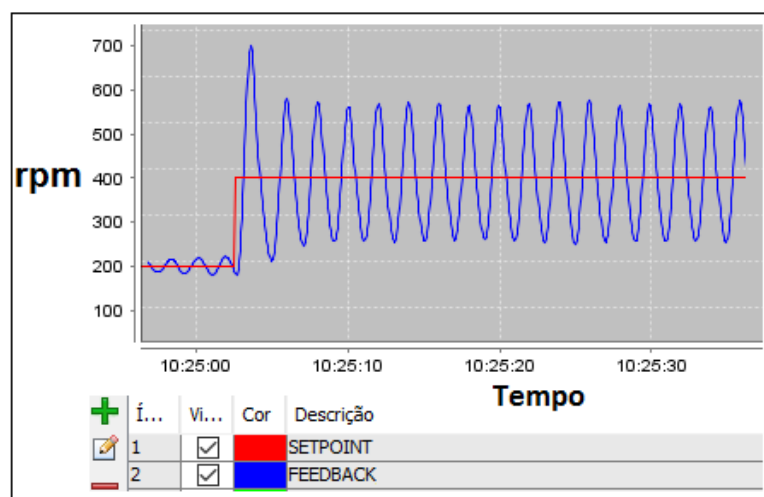


Figura 63 – Resposta ao degrau com controlador PI (ITAE)



A resposta com controlador PID apresentou um sobressinal de aproximadamente 30%. Dependendo da aplicação e das exigências do sistema, o resultado não seria satisfatório, porém, com esses valores, um ajuste fino pode ser feito para melhorar a resposta, sendo muito mais simples do que realizar a sintonia completa. Já a resposta com o controlador PI, se mostrou oscilatória, não convergindo para o valor desejado.

Testou-se também o método Tyreus-Luyben e Ciancone-Marlin para esta mesma planta, nas mesmas condições. Os resultados podem ser vistos na Tabela 22 e Tabela 23, respectivamente.



Tabela 22 – Resultados do servomotor para controlador Tyreus-Luyben

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
20,0%	1,0%	0,020	1,425	0,375	0,010	0,003	0,002	0,006	0,002

Tabela 23 – Resultados do servomotor para controlador Ciancone-Marlin

Configuração		Dados coletados			PID			PI	
<i>Delta</i>	<i>Histerese</i>	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
20,0%	1,0%	0,020	1,425	0,375	0,009	0,018	0,001	0,006	0,017

O método de Ciancone-Marlin se mostrou ainda mais oscilatório do que o método ITAE já apresentado, tanto para controlador PID quanto para o PI.

Figura 64 – Resposta ao degrau com controlador PID (Ciancone-Marlin)

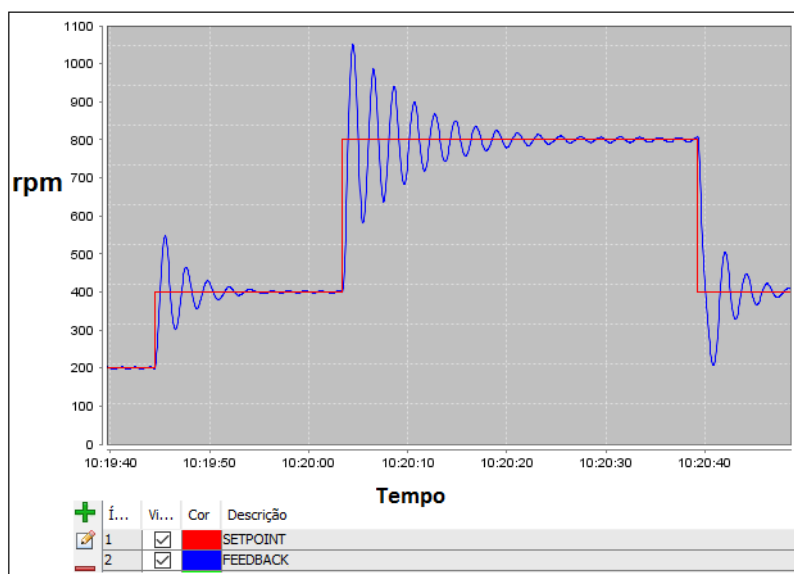
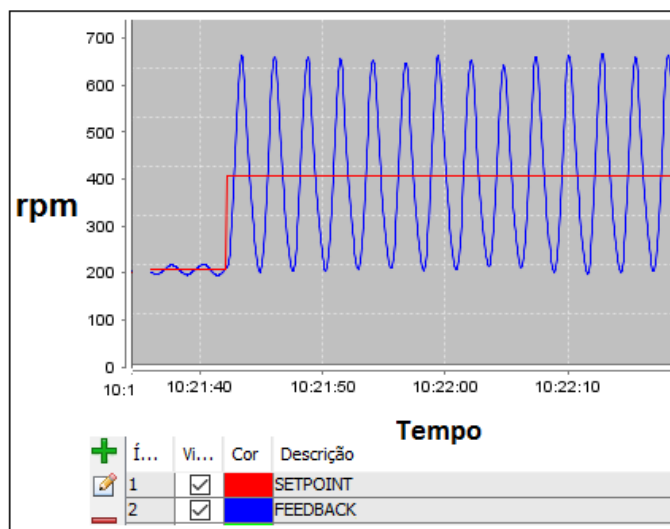


Figura 65 – Resposta ao degrau com controlador PI (Ciancone-Marlin)



O método Tyreus-Luyben apresentou-se melhor do que os demais tanto para o controlador PID quanto para o PI. Dos três métodos avaliados, este foi o único que o controlador PI também convergiu, apesar de apresentar um sobressinal maior do que o controlador PID.

Figura 66 – Resposta ao degrau com controlador PID (Tyreus-Luyben)

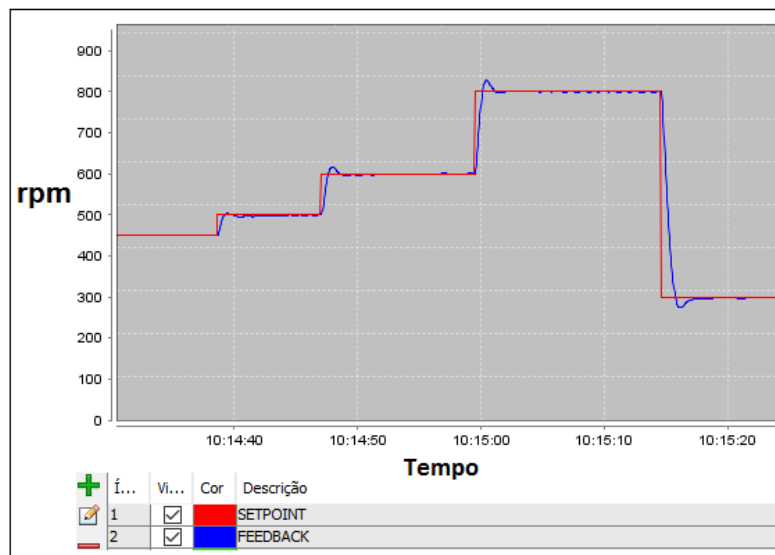
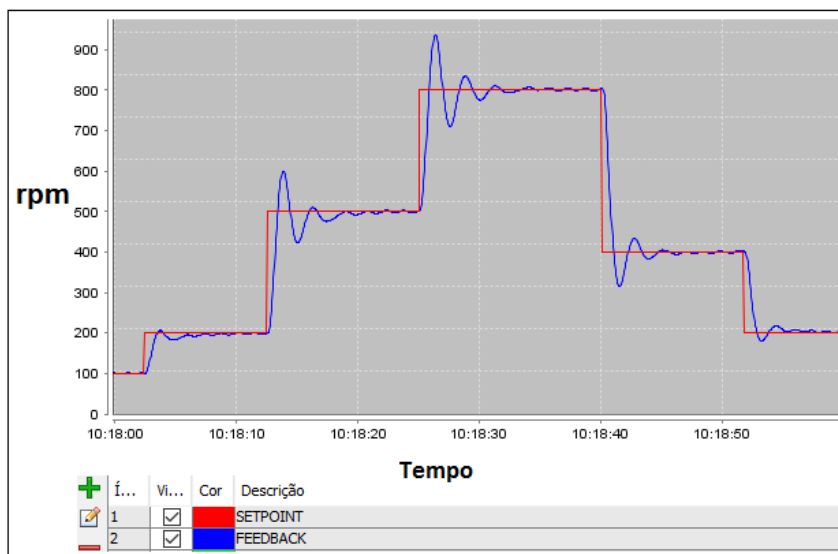


Figura 67 – Resposta ao degrau com controlador PI (Tyreus-Luyben)

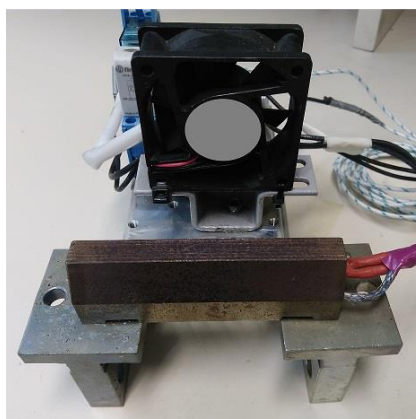


### 6.2.3 Planta Térmica

A planta térmica simulada pode ser vista na Figura 68. Ela é composta por um relé de estado sólido, por um ventilador e por um conjunto metálico que é aquecido e tem sua temperatura medida por um termistor do tipo PT100.

O acionamento do relé de estado sólido é feito via PWM do PLC300, com uma frequência de 1Hz e largura de pulso que varia de 0 à 100%. A leitura do PT100 é feita por uma Unidade Remota WEG, RUW05, que transmite o valor da temperatura via CANopen para o PLC300. Com base na informação de temperatura e as demais configurações, o PLC300 realiza a sintonia automática e o controle do sistema.

Figura 68 – Planta térmica



Delta igual a 50% e histerese de 1% foram configurados para o ensaio do relé. O valor de delta um pouco maior foi escolhido, pois com igual a 20% a oscilação na saída foi muito pequena, fazendo com que ruídos e oscilações na temperatura pudessem influenciar nos resultados obtidos.

O gráfico da aplicação da sintonia pode ser visto na Figura 69 e os resultados obtidos na Tabela 24.

Figura 69 – Curva do relé para a planta térmica

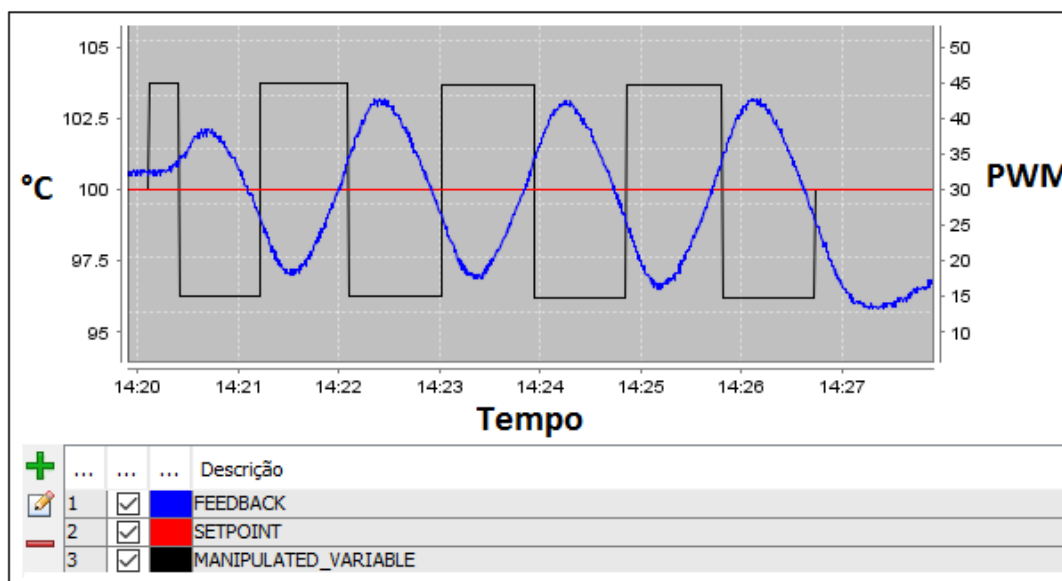


Tabela 24 – Resultados da planta térmica para controlador Tyreus-Luyben

Configuração		Dados coletados			PID			PI	
Delta	Histerese	$K_u$	$P_u$	$C_p$	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$
50,0%	1,0%	5,876	112,816	0,065	2,879	0,011	47,729	1,822	0,007

Como a constante de proporcionalidade ficou menor do que 0,1, a regra para obtenção do controlador utilizada foi a de Tyreus-Luyben.

A resposta ao degrau foi verificada com o controlador do tipo PID sugerido. O gráfico desta resposta pode ser visto na Figura 70 e na Figura 71.

Figura 70 – Resposta ao degrau positivo com controlador PID (Tyreus-Luyben)

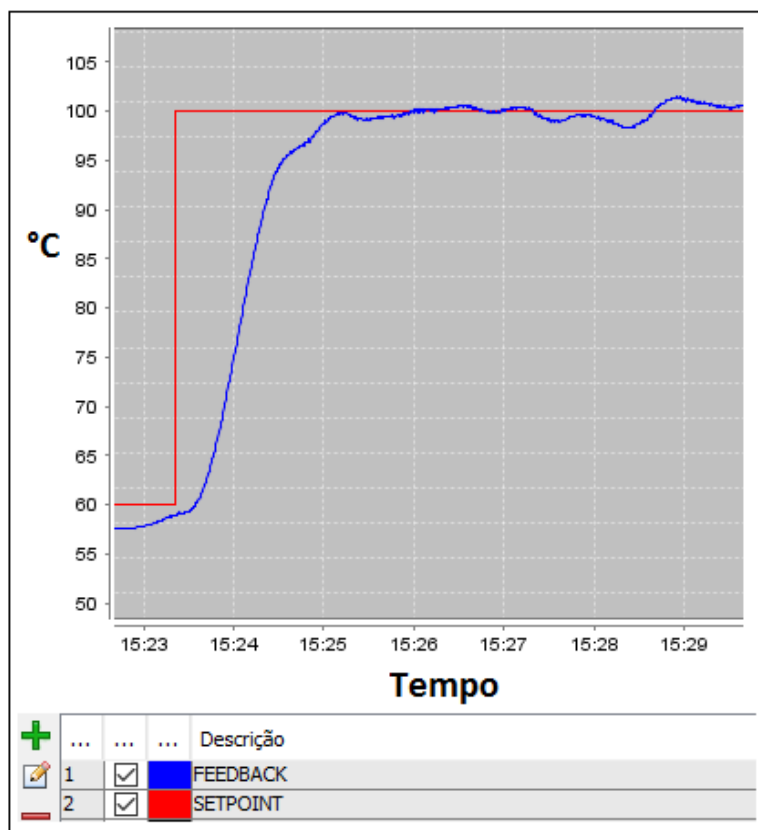
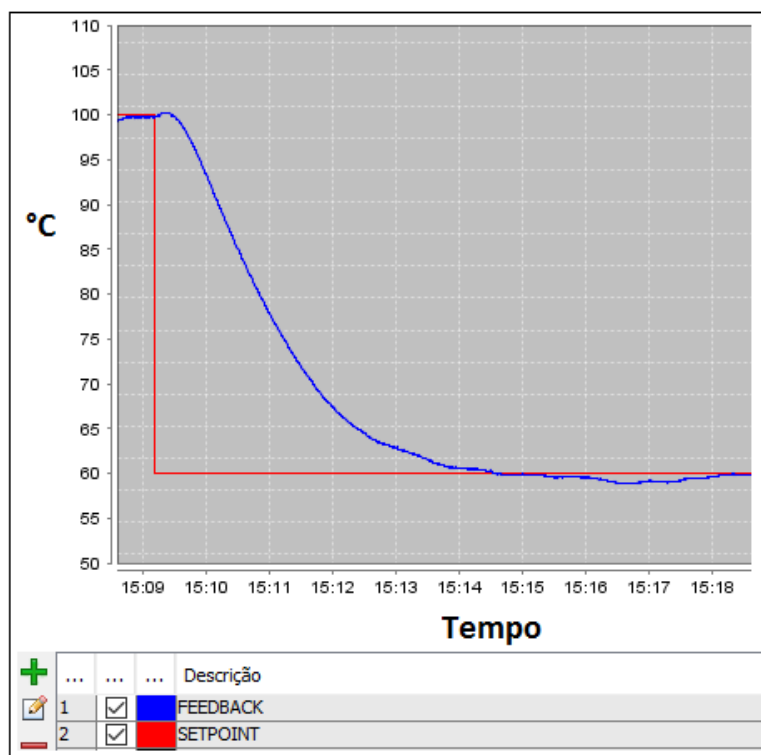


Figura 71 – Resposta ao degrau negativo com controlador PID (Tyreus-Luyben)



Cada divisão dos gráficos representa 1 minuto na escala temporal. No degrau positivo de temperatura, o sistema atingiu o *setpoint* em aproximadamente dois minutos. O resfriamento, durante o degrau negativo, levou cerca de 5 minutos. Esses tempos se devem à limitação do sistema, tanto no aquecimento quanto no resfriamento, pois a constante de tempo desta planta é grande. O sistema controlado não apresentou sobressinal. Uma oscilação de aproximadamente 2% pode ser verificada na variável controlada, valor razoavelmente baixo, que para a maioria das aplicações é aceitável.

Para o controlador PI, a resposta foi muito semelhante ao PID. A única diferença foi que ele se mostrou um pouco mais lento do que o PID. Por esta semelhança, optou-se por não exibir os gráficos do controlador PI.

## 7 CONCLUSÃO

Os resultados obtidos nas plantas reais e nas simuladas mostraram que o bloco PID2 pode ser aplicado em plantas típicas de controle dos clientes WEG, como temperatura, velocidade, pressão, entre outros. Embora pressão em particular não tenha sido analisada, ela se assemelha às outras plantas, principalmente a de controle de velocidade do servomotor.

Avaliando os resultados e analisando a forma como foi realizada a implementação desta ferramenta em um sistema embarcado, chega-se a algumas conclusões que tornam esta nova funcionalidade implementada no PLC300 muito atrativa.

Uma das vantagens desta implementação no sistema embarcado é a aquisição dos dados que pode ser efetuada com uma taxa maior do que a aquisição via software (métodos aplicados em linguagem *ladder*, por exemplo), pois, esta última, geralmente fica condicionada à taxa de comunicação do sistema podendo também ser dependente do ciclo de varredura do produto, o que pode inviabilizar a aplicação em plantas muito rápidas.

Uma segunda vantagem desta ferramenta desenvolvida em particular é a simplicidade da utilização, que depende do ajuste de apenas dois parâmetros (delta e histerese) além dos parâmetros do bloco de controle já utilizado. Esses dois parâmetros podem até vir com um valor pré-definido ou basta seguir a recomendação da ajuda do bloco.

Outra grande vantagem da implementação em firmware, é que toda a lógica está junto com o CLP, não necessitando de um computador e de cabos de comunicação para efetuar uma nova sintonia. Sempre que for necessário, basta o usuário ativar novamente a sintonia através de uma entrada digital, de alguma tecla ou outro comando qualquer, que pode ser configurado durante o desenvolvimento do software aplicativo. Além disso, há a possibilidade de o usuário trocar de regra de sintonia sem aplicar novamente o método dos relés, desta forma, mesmo que ele não tenha conhecimento das diferenças entre as regras, ou até mesmo entre o controlador PID e o PI, ele pode facilmente escolher um método diferente e avaliar como o seu sistema irá se comportar com os novos parâmetros calculados, podendo fazer isso quantas vezes achar necessário. Os resultados podem ser facilmente

exibidos na tela do produto, podendo eles serem aceitos ou não para o novo ajuste do sistema de controle.

Essa funcionalidade permite, em determinadas situações, que até mesmo o usuário da máquina final possa realizar uma nova calibração do sistema, não necessitando ter acesso ao software aplicativo ou sequer de um computador.

Por outro lado, métodos numéricos e cálculos mais avançados muitas vezes não são possíveis de programar em um sistema embarcado, devido à limitação de memória e de tempo de processamento, que pode acabar comprometendo outras funcionalidades do produto.

De maneira geral, a ferramenta desenvolvida cumpre com a necessidade da empresa. Embora em algumas aplicações o resultado obtido automaticamente possa não ser o esperado como resultado final ou o mais eficiente, mesmo assim a funcionalidade se mostra adequada para realizar uma boa sintonia inicial. Partindo desta sintonia inicial, com pequenos ajustes efetuados de maneira manual nos ganhos do controlador, um sistema de controle mais preciso e adequado para a aplicação pode ser obtido.



## 8 TRABALHOS FUTUROS

Como trabalho futuro, sugere-se a integração entre a implementação em firmware embarcado e alguma implementação em software, como a proposta por Finke, 2016. Desta forma, pode-se juntar a simplicidade e a alta taxa de amostragem da implementação em firmware, com uma análise mais avançada através de gráficos e outras informações que podem ser proporcionados pela implementação em software, após a coleta dos dados.

Sugere-se também a implementação de outras regras de obtenção do controlador PID e também para identificação da função de transferência da planta, trazendo isso como resposta do sistema e possível modelo para aplicação de outros métodos de obtenção do controlador.

Uma análise mais detalhada com relação à influência de delta e histerese poderia ser feita, com o intuito de verificar até que ponto elas não influenciam na resposta final do controlador.

Fica como sugestão fazer uma análise gráfica das curvas obtidas durante a aplicação do método, para diferentes tipos de plantas com modelo conhecido e comparar os resultados com os obtidos pelo trabalho de Yu (2007). Outra possibilidade de avaliação da qualidade do controlador seria a aplicação da ferramenta em sistemas reais que possuem seu modelo matemático determinado e com posterior comparação dos resultados obtidos com simulações da planta em software Matlab.

## 9 REFERÊNCIAS

NEVES, M.G.S. **Auto-tuning de Controladores PID pelo método Relay**: Otimização de Controlo em Automação Industrial. 2009. 72p. Dissertação (Mestrado em Engenharia Electrotécnica e de Computadores) - Instituto Superior Técnico da Universidade Técnica de Lisboa, Lisboa 2009;

Autor Desconhecido: **Realização de Auto-Tuning em Controladores de Processos**. Matéria publicada na revista Mecatrônica Atual; Ano: 10; Nº 54; Jan / Fev – 2012;

ZIEGLER, J.G. e NICHOLS, N.B. **Optimum Settings for Automatic Controllers**. Trans. ASME, pp. 759-768, 1942;

ASTROM, K.J. & HAGLUND, T. H. **New Tuning Methods for PID Controllers**. Proc. 3ª European Control Conference, p.2456-62, 1995;

OGATA, K. **Modern Control Engineering**. 4ª. Ed., Pearson/ Prentice Hall, 2002, cap. 10;

HANG, C. C.; ASTRÖM, K. J.; WANG, Q. G. **Relay feedback auto-tuning of process controllers**: a tutorial review. Journal of process control 12, 2002. 143-162;

BOAVA, C. **Desenvolvimento de uma função de autossintonia PID para um controlador automático de tensão**. 2016. 65p. Dissertação de Mestrado - Fundação Universidade Regional de Blumenau, Blumenau-SC, 2016;

YU, C.-C. **Autotuning of PID Controllers: A Relay Feedback Approach**. 2nd. ed. London: Springer-Verlag, 2007;

CARDOSO, P. M. **Estudo, proposta e avaliação de novas metodologias de sintonia automática de controladores PID baseadas no ensaio do relé realimentado**. Dissertação de Mestrado. Uberlândia, MG, 2002;

FINKE, J. H. **Sintonia automática de controladores PID pelo método do relé realimentado utilizando a técnica do relé realimentado e o método de Luyben e Yu**. 2016. 111p. Dissertação de Mestrado - Fundação Universidade Regional de Blumenau, Blumenau-SC, 2016;

TAN, K. et al. **Improved critical point estimation using a preload relay**. Journal of Process Control, Elsevier, v. 16, n. 5, p. 445–455, 2006;

PINTO, J. E. M. G. **Aplicação prática do método de sintonia de controladores PID utilizando o método do relé com histerese.** Universidade Federal do Rio Grande do Norte, 2014.

<http://www.hisa.com.br/produtos/turbina-francis>

JOHNSON, M. A. and MORADI, M. H. **PID Control: New Identification and Design Methods.** Springer, 2005, 543p;

ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced PID control.** [S.I.]: ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006;

BERNER, J. **Automatic Tuning of PID Controllers based on Asymmetric Relay Feedback.** Master Thesis., 2015.

FERDOUS, R. **Advanced Process Control and Relay Auto-tuning.** PhD Thesis. Singapore, 2005;

ÅSTRÖM, K. J. **Control system design. Lecture notes,** Department of Mechanical and Environmental Engineering, University of California Santa Barbara, 2002;

RIVERA D.E., MORARI, M., SKOGESTAD, S. **Internal model control PID controller design.** Ind.Eng. Chem. Proc. Des. Dev. 1986;25-252;

O'DWYER, A. **A summary of pi and pid controller tuning rules for processes with time delay. part 2: Pid controller tuning rules.** Dublin Institute of Technology, 2000;

ZAMANI, Z.T., MOSHIRI, B., FATEHI, A., SEDIGH A.K. **Relay feedback based monitoring and autotuning of processes with gain non linearity;** Control and Intelligence Processing Center of Excellence, School of ECE University of Tehran, Tehran, Iran; p. 16;

LIU,T., WANG, Q. G. and HUANG, H. P. - **A tutorial review on process identification from step or relay feedback test** – Journal of Process Control 23, 2013, 1597-1623;

ALCINDO, P., J. **Controle de processos industriais.** Universidade do Estado de Santa Catarina – CCT, Joinville-SC, 2010 p. 67;

AABB. **Functional Description Pid01A:** Closed Loop Controller with Auto Tuner, 2005, p.47;

EMERSON PROCESS MANAGEMENT. **DeltaV Insight for Intelligent Control**. 2013, p. 12;

SIEMENS. **Simatic PID Temperature Control: TCONT\_CP**. Manual. 2003, p. 102;

WEG - **Manual de Comunicação CANopen do PLC300**, 2010 (disponível em [www.weg.net](http://www.weg.net));

KERNIGHAN, B., W., RITCHIE, D., M. - **C a Linguagem de Programação** - 3ed, Edisa, 1987;

Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Doyle, F. J. (2011). **Process Dynamics and Control**. 2nd ed. Hoboken: Wiley;

ÅSTRÖM, K. J.; HÄGGLUND, T. – **PID Controllers: Theory, Design, and Tuning**. Instrument Society of America: Research Triangle Park; 1995.

## 10 BIBLIOGRAFIA CONSULTADA

SOLTESZ, K. **On Automation of the PID Tuning Procedure**. Licentiate Thesis., 2012;

VISIONLI, A., SHONG, Q. **Control of Integral Porcess with Dead Time**. Springer, 2011, 250 p.;

OLIVEIRA E., M. **Implementação de Técnicas de Controle Avançado a uma Planta Piloto de Controle de Vazão e Temperatura de Ar**. Universidade Federal de Minas Gerais. Belo Horizonte, 2008, 220 p.;

FRIMAN, M.; WALLER, K. V. **A two channel relay for autotuning**. Industrial & Engineering Chemistry Research (36), p. 2662–2671, 1997;

FRIMAN, M. **Extensions and Modifications of Relay Autotuning**. Academic Dissertation. Department of Chemical Engineering, Åbo Akademi University, Finland, 1997;

COLOGNI, M. A. **Estudo e avaliação de metodologias de auto-sintonia de controladores pid visando uma implementação em controlador industrial**. Dissertação de Mestrado. Florianópolis, SC, 2008;

COELHO, C. A. D. S. **Auto - ajuste de controladores PID usando o método de linearização harmônica**. Monografia de Conclusão de Curso. Rio de Janeiro, RJ, 2004;

ÅSTRÖM, K. J.; HÄGGLUND, T. **The future of pid control**. Control engineering practice, Elsevier, v. 9, n. 11, p. 1163–1175, 2001;

ZIEGLER, J. G.; NICHOLS, N. B. **Optimum settings for automatic controls**. Transactions ASME 64, Novembro 1942. 759-768;

TAN, K. K.; LEE, T. H.; JIANG, X. **On-line Relay Identification, Assessment and Tuning of PID Controller**. Journal of Process Control (11), 2001. 483-496;

WANG, Q.-G.; HANG, C.-C.; ZOU, B. **Low-order modeling from relay feedback**. Industrial & Engineering Chemistry Research (36), p. 375–381, 1997;

LUYBEN, W. L. **Getting more information from relay-feedback tests**. Industrial & engineering chemistry research, ACS Publications, v. 40, n. 20, p. 4391–4402, 2001;

HAUGEN, F. **Comparing PI Tuning Methods in a Real Benchmark Temperature Control System.** Telemark University College. Modeling, Identification and Control, Vol. 31, No. 3, 2010, pp. 79–91;

SOLTESZ, K., HÄGGLUND T. **Extending the Relay Feedback Experiment.** Department of Automatic Control, Lund University, 18th IFAC World Congress Milano (Italy), 2011, p. 6.