

LABORATÓRIO DE AUTOMAÇÃO DE SISTEMAS ELÉTRICOS



2012

3. Controle de Reator Químico

Prof. Eduardo Lorenzetti Pellini

PEA/EPUSP

Nesta experiência o aluno desenvolverá um sistema automático de controle para um reator químico hipotético. A atividade envolve a implantação do sequenciamento e intertravamento de operações do reator, além do desenvolvimento e ajuste de um controlador do tipo PID. Para essa estratégia de controle, os alunos terão contato com a linguagem de programação FBD (Function Block Diagram) da IEC 61131-3.

Versão 2.1

Baseado em material originalmente elaborado pelo Prof. Eduardo Cesar Senger.
Telas de IHM e sistema supervisorio pelo Eng. Marcos Yukio Yamaguchi.

Controle de Reator Químico

1. Apresentação

O sistema a ser controlado e supervisionado consiste em um tanque-reator químico, perfeitamente agitado, ou CSTR (*Continuous Stirring Tank Reactor*). Tal reator é composto por um vaso cilíndrico que recebe o fluxo de dois reagentes em fase líquida, para que reajam quimicamente de forma a produzir um determinado composto final. A admissão dos reagentes, sua mistura, o tempo de reação química e a temperatura da reação devem ser controlados para que a eficiência e a qualidade do processo sejam maximizadas. Esse esquema pode ser visto com mais detalhes na Fig. 1.

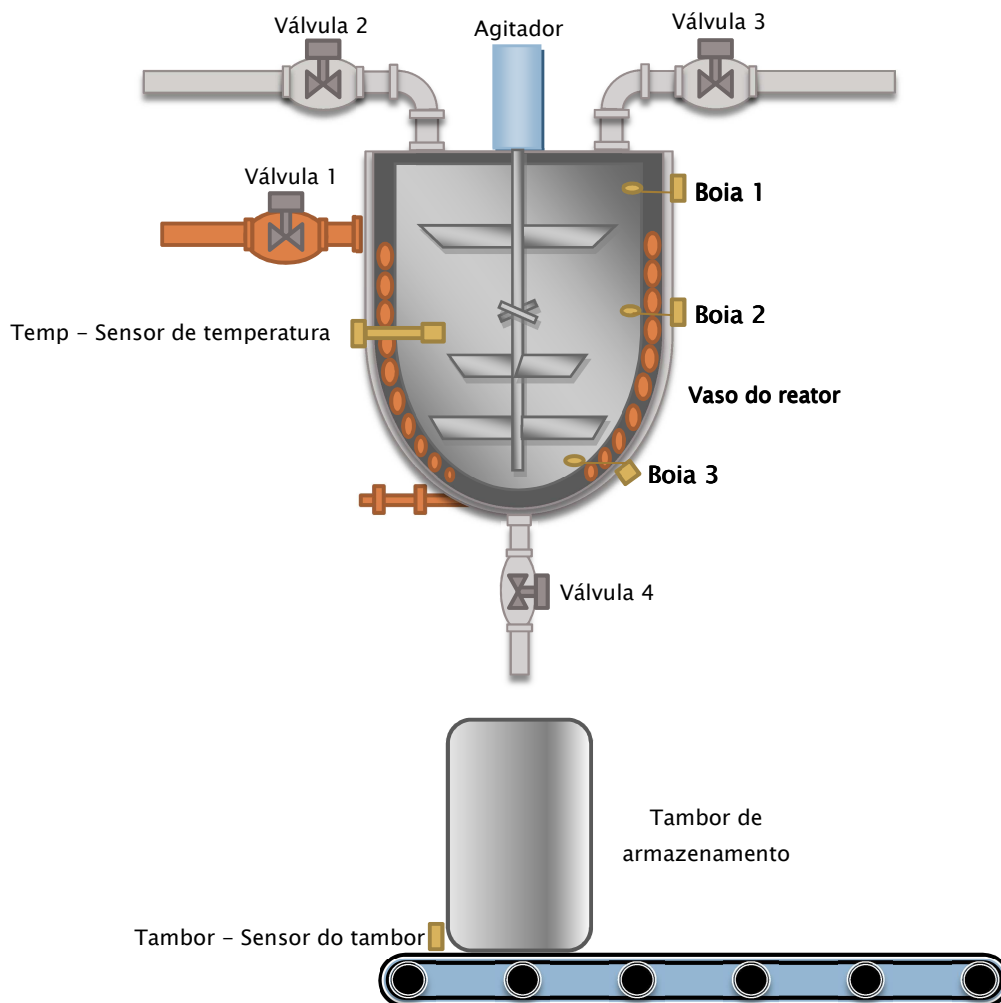


Fig. 1 - Esquema do reator CSTR.

Na Fig. 1 podem ser observados os seguintes elementos:

- **Válvula 1** – uma eletroválvula para o controle do fluxo de vapor, responsável pelo aquecimento do vaso do reator até uma temperatura desejada;
- **Válvula 2** – uma eletroválvula para o controle do fluxo do reagente 1 para dentro do volume do vaso;
- **Válvula 3** – uma eletroválvula para o controle do fluxo do reagente 2 para dentro do volume do vaso;
- **Válvula 4** – uma eletroválvula para o controle do fluxo de saída do composto final presente no vaso, para armazenamento em um tambor posicionado sob o reator;
- **Agitador** – uma hélice agitadora ligada a um motor de indução, responsável por misturar os reagentes durante a produção do composto final;
- **Temp** – um sensor de temperatura, posicionado dentro do vaso, para detecção da temperatura da reação;
- **Boia 1, Boia 2 e Boia 3** – três sensores do tipo boias, para detecção do nível do composto dentro do vaso;
- **Tambor** – local sob o reator para o posicionamento de um tambor de armazenamento, para receber a saída de produto do vaso. Nesse local há um sensor de proximidade para a detecção do correto posicionamento do tambor, antes do início do processo e seu posterior enchimento.

Deseja-se que o processo de administração dos reagentes, sua mistura, o aquecimento, o controle de temperatura do vaso do reator e a liberação do produto para armazenamento sejam totalmente automatizados. Para isso será empregado um Controlador Programável (CP) com uma interface IHM de LCD, tipo *touchscreen*, além dos sensores e atuadores mostrados na Fig. 1.

2. Requisitos do processo

O processo a ser automatizado possui os seguintes requisitos de operação:

- a) Antes do início da mistura dos reagentes, a **Válvula 4** deve estar fechada e um tambor de armazenamento deve estar corretamente posicionado sob a saída dessa válvula. A posição do tambor é detectada pela sensibilização de um sensor de proximidade denominado **Tambor**. Estando em posição, dá-se início ao processo após o pressionamento de um botão **LIGAR** presente na IHM *touchscreen*.
- b) No início de suas operações, o reator deve ser inicialmente aquecido até atingir a temperatura de 100,0 [°C]. Este aquecimento é conseguido através da abertura da válvula de admissão de

- vapor **Válvula 1**, de forma controlada, através do monitoramento do sensor de temperatura **Temp**.
- c) Após o aquecimento do vaso, o controlador programável deve abrir a **Válvula 2**, para permitir a entrada de 400 litros do reagente 1. Quando essa quantidade for alcançada dentro do vaso, essa situação poderá ser detectada, através do monitoramento do sensor **Bóia 2**.
 - d) Com o acionamento da **Bóia 2**, a **Válvula 2** deve ser fechada. Simultaneamente, a **Válvula 3** deve ser aberta para permitir a entrada de 200 litros do segundo reagente. Essa quantidade poderá ser detectada através do monitoramento do sensor **Bóia 1**.
 - e) Durante todo o processo de alimentação do vaso, a temperatura deve ser mantida em 100,0[°C], com uma faixa de variação de $\pm 10,0$ [°C]. Se a temperatura cair abaixo dessa faixa, a entrada dos reagentes deve ser suspensa e o vapor deve ser liberado automaticamente para reaquecer o sistema. Se a temperatura subir para valores acima dessa faixa, a entrada dos reagentes deve ser suspensa e deve-se aguardar o resfriamento natural do tanque. O processo de entrada de efluentes deverá ser reiniciado assim que a temperatura retorne à sua faixa adequada de operação.
 - f) Após as duas substâncias terem sido adicionadas, a temperatura deverá ser elevada para 200,0[°C], com um erro de $\pm 10,0$ [°C]. Isso deverá ser feito com a abertura da **Válvula 1** de admissão de vapor.
 - g) Em seguida, o **Agitador** deve ser ligado por **60 segundos** para homogeneizar as duas misturas. A temperatura deverá ser continuamente controlada durante esse processo, sendo mantida em 200,0[°C], com um erro de $\pm 10,0$ [°C]. Fora dessa faixa de tolerância, a atividade de agitação deve ser pausada. Caso a temperatura exceda a o limite de tolerância superior, deve-se aguardar o resfriamento do reator. Caso a temperatura caia para um limiar inferior à temperatura mínima aceitável, a **Válvula 1** deve ser novamente acionada para aquecer o processo. A agitação deve ser retomada assim que a temperatura esteja nos limites aceitáveis.
 - h) Quando o agitador for desligado, o controlador programável deve abrir a **Válvula 4** do dreno, dando início à transferência do produto para o tambor de transporte.
 - i) Após o completo esvaziamento do vaso do reator (detectada pelo desligamento do sensor da **Boia 3**), a **Válvula 4** do dreno deve ser fechada. Nessa ocasião, deve-se aguardar a retirada do Tambor cheio (desligamento do sensor **Tambor**).
 - j) Após a retirada do Tambor cheio, uma nova corrida do processo pode ser realizada através do posicionamento de outro tambor e um novo pressionamento do botão de **Iniciar** na IHM.
 - k) Em qualquer situação, se o botão **Desligar** na IHM por pressionado o sistema deve ser paralisado, ou seja, a admissão de vapor, a entrada de reagentes, o esvaziamento do tanque e

a agitação devem ser interrompidos. O sistema deve retornar à situação operativa anterior caso o botão **Ligar** seja novamente pressionado.

Tais requisitos podem ser modificados pelo aluno caso alguma condição de controle não seja plenamente segura ou coerente. Qualquer modificação deve ser documentada no relatório final.

3. Requisitos para o controlador

O controlador terá as entradas e saídas analógicas e digitais conforme a especificação citada a seguir. Mais detalhes do CP podem ser obtidos em (Bradley, 2012).

3.1. Sinais digitais de entrada

O controlador deverá monitorar entradas digitais físicas, provenientes de sensores situados em campo no reator, além de variáveis internas (TAGs), booleanas, controladas pela IHM. Essas entradas são mostradas com mais detalhes na Tabela 1, onde também é listado o nome da TAG que poderá ser adotada na lógica do CP. Todas essas TAGs deverão ser do tipo 'BOOL'.

Tabela 1 – Entradas digitais do controlador programável.

Tipo	Nome da TAG	Descrição	Características
Entrada digital 1	InBoia1	Sensor do nível 1 do vaso (cheio)	Sensor NA
Entrada digital 2	InBoia2	Sensor do nível 2 do vaso	Sensor NA
Entrada digital 3	InBoia3	Sensor do nível 3 do vaso (vazio)	Sensor NA
Entrada digital 4	InTambor	Sensor de tanque em posição	Sensor NA
Tag controlada pela IHM	IHM_Ligar	Sinal para partida do processo	Ativo Alto
Tag controlada pela IHM	IHM_Desligar	Sinal para parada do processo	Ativo Alto

A associação das entradas digitais físicas do CP a essas variáveis pode ser feita da forma habitual, em LADDER ou qualquer outra linguagem, em alguma tarefa periódica do CP.

3.2. Sinais digitais de saída

O controlador poderá acionar as seguintes saídas físicas durante o processo de controle da planta. Essas saídas são mostradas com mais detalhes na Tabela 2, onde também é listado o nome da TAG que poderá ser adotada na lógica do CP. Todas essas TAGs deverão ser do tipo 'BOOL'.

Tabela 2 – Entradas digitais do controlador programável.

Tipo	Nome da TAG	Descrição	Características
Saída digital 1	OutValvula1	Liberação de vapor para aquecimento	Saída digital de estado sólido
Saída digital 2	OutValvula2	Admissão do reagente 1 ao vaso do reator	Saída digital de estado sólido
Saída digital 3	OutValvula3	Admissão do reagente 2 ao vaso do reator	Saída digital de estado sólido
Saída digital 4	OutValvula4	Liberação do composto final presente no vaso do reator	Saída digital de estado sólido
Saída digital 5	OutAgitador	Agitador da mistura presente no vaso do reator	Saída digital de estado sólido

A associação dessas variáveis às saídas digitais físicas do CP pode ser feita da forma habitual, em LADDER ou qualquer outra linguagem, em alguma tarefa periódica do CP.

3.3. Sinal analógico de entrada

O monitoramento gradual da temperatura do vaso do reator requer que o CP tenha a capacidade de ler sinais analógicos provenientes de um sensor de temperatura presente em campo. Neste caso, foi empregado um transdutor de temperatura do tipo RTD (resistor variável com temperatura) com uma ponte de medição incorporada, que gera em sua saída um sinal analógico de tensão, com faixa dinâmica entre 0,0 a 10,0 [V] de tensão de saída, para uma temperatura na entrada entre 0,0 a 300,0 [°C], com um comportamento linear, ou seja a relação entre temperatura e tensão é dada por um ganho 'G' de 30,0 [°C/V].

Uma entrada analógica do CP, com faixa de tensão entre $\pm 10,0$ [V] (bipolar), irá converter os sinais de tensão do termômetro em um valor binário, através de um processo de digitalização, com resolução de 16 bits e codificação em complemento de dois. A tabela 3 a seguir mostra a variação desses números convertidos pelo controlador programável em função da faixa de tensão analógica na entrada.

Tabela 3 – Entradas digitais do controlador programável.

Faixa de tensão na entrada [V]	Valor Convertido (representação decimal)	Nº de Bits Significativos
-10,0 a +10,0	-32768 a +32767	16
0,0 a +10,0	0 a +32767	15
0,0 a +5,0	0 a +16383	14

Ou seja, pode-se inferir a seguinte relação:

$$\text{valor_convertido} = (\text{tensão_analógica}) \times 32767/10,0$$

Para o transdutor considerado, com a temperatura 'T' entre 0,0 a 300,0 [°C] e tensão de saída entre 0,0 a 10,0 [V] (ganho 'G' de 30,0 [°C/V]), obtém-se:

$$\text{tensão_analógica} = T \times (1,0/G) = T \times (1,0/30,0)$$

Ou seja,

$$\text{valor_convertido} = (\text{int}) (32767 \times T/300,0)$$

Alguns exemplos de leituras do sensor de temperatura codificados dessa forma são mostrados na Tabela 4.

Tabela 4 - Exemplos de leituras da entrada analógica de temperatura codificada para um número decimal.

Temperatura [°C]	Tensão [V]	Valor Convertido
90,00	3,00	9830
100,0	3,33	10922
110,0	3,67	12015
190,0	6,33	20752
200,0	6,67	21845
210,0	7,00	22937

Até gerações mais antigas de CPs, era muito comum o uso de variáveis codificadas dessa forma, em uma unidade de medida com escala fictícia, em representação inteira, sem frações decimais. Assim, uma temperatura de 90,0 [°C] era utilizada em um controlador do tipo PID, por exemplo, como um valor decimal equivalente de 9830 unidades. Todos os ganhos e ajustes eram feitos com base nessas unidades fictícias, dificultando a análise, o projeto do controlador e até seu desempenho dinâmico, devido a baixa precisão e possíveis erros de truncamento.

Nos CPs modernos que dispõem de mais recursos computacionais, para tornar o controle mais simples de ser projetado, analisado e ajustado, as variáveis que armazenam os valores de entradas analógicas podem ser convertidas, dentro do programa, de volta para uma representação em ponto flutuante, consistente com a unidade de engenharia real. Assim, por exemplo, ao invés de representar a temperatura nas malhas de controle como um número decimal inteiro, como 21845, o controlador poderá utilizar seu valor real em [°C] de 200,0.

No programa que será desenvolvido, deverão ser utilizadas duas variáveis:

- Uma variável denominada **InTermRaw**, ou seja, a temperatura segundo sua leitura “pura”, vinda do cartão de entradas analógicas, em codificação inteira (INT), resolução de 16 bits, e;
- Uma variável auxiliar **InTerm**, em ponto flutuante (REAL), calculada no programa do CP através da multiplicação do valor instantâneo de InTempRaw por um ganho constante, conhecido, de $300/32767$, ou aproximadamente 0,00916.

Essas variáveis são mostradas na Tabela 5.

Tabela 5 – Variáveis do controlador programável para as entradas analógicas.

Tipo	Nome	Descrição	Características
Entrada analógica 1	InTermRaw	Leitura do sensor de temperatura, em representação inteira, dada pelo conversor AD do CP	Entrada analógica de $\pm 10,0$ [V] representada em 16 bits, com sinal
Variável interna	InTerm	Leitura do sensor de temperatura, convertida para valores de engenharia em [°C]	Número real, de ponto flutuante

A associação das entradas analógicas físicas do CP a essas variáveis pode ser feita da forma habitual, em LADDER ou qualquer outra linguagem, em alguma tarefa periódica do controlador. Deve-se atentar que em LADDER, a atribuição de um valor inteiro (uma palavra de vários bits) ou uma variável, a outra variável, deve ser feita através de uma função tipo MOV, como será mostrado adiante.

É importante salientar que em projetos reais onde são utilizadas leituras de grandezas analógicas provenientes de sensores em campo, é comum criar blocos ou funções no programa para a calibragem e a filtragem digital das leituras obtidas, de forma a eliminar ruídos e componentes de frequência indesejada presente nos sinais de campo.

3.4. Controlador de temperatura do tipo PID

Devido às características dinâmicas do sistema térmico presente nessa planta, a estratégia de controle que será empregada é um controlador de temperatura do tipo PID. Nesse esquema, a medida atual de temperatura da planta (*feedback*) será comparada com um valor de referência desejado. Essa comparação gera um sinal de erro de temperatura, ErrT, que é então aplicado ao controlador PID, gerando em sua saída um sinal de comando, Cmd, responsável por atuar sobre a eletroválvula de vapor, para elevar a temperatura do sistema quando necessário. Esse esquema é mostrado na Fig. 2.

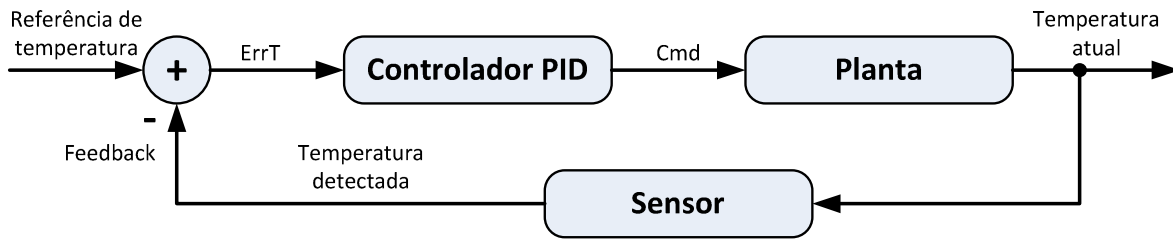


Fig. 2 – Controlador PID aplicado ao processo do reator.

A estrutura típica e a teoria geral para um controlador PID podem ser vistas em mais detalhes em (Ogata, 2011).

A estrutura interna do PID é mostrada detalhadamente na Fig. 3. Nessa é possível notar que a saída **Cmd** é constituída pela soma de três termos:

- Um termo de comando de ação instantânea, diretamente proporcional à magnitude do sinal de erro, e a um ganho constante K_P ;
- Um termo de comando de ação atrasada, diretamente proporcional à integral do sinal de erro, e a um determinado ganho K_I ;
- Um termo de comando de ação adiantada, diretamente proporcional à derivada do sinal de erro, e a um determinado ganho K_D .

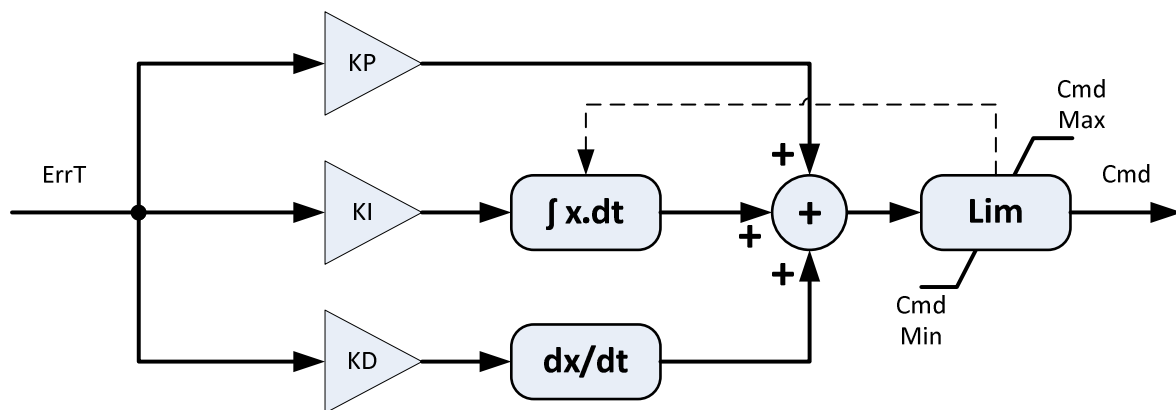


Fig. 3 – Estrutura interna típica de um controlador PID.

A forma de codificação do algoritmo de um PID em um controlador programável é bastante variada. Ela pode ser realizada em qualquer uma das linguagens da IEC 61131-3, de forma discreta (envolvendo blocos elementares de soma, produto, limitadores, etc.), ou então realizada com um algoritmo especialmente desenvolvido pelo fabricante do CP, com melhorias e otimizações nas rotinas de cálculo, com aritmética de precisão dupla, simples ou inteira, etc., e disponibilizado como um bloco elementar das linguagens de programação. Nos CP modernos, existem blocos de controladores PI ou PID que também possuem recursos incorporados para controle adaptativo, para ajuste automático de

parâmetros, controle robusto e multivariável, etc.

No experimento será empregado um PID construído discretamente, com um bloco PI e um bloco derivativo (DERV) criados pela Rockwell Automation em seu software RSLogix. Mais detalhes a respeito dos blocos funcionais PI e DERV podem ser obtidos em (Automation, 2012).

O ajuste do controlador será realizado pelos próprios alunos com base nos critérios que julgarem relevantes. A atuação do controlador poderá ser observada através de gráficos gerados pela ferramenta de parametrização RSLogix, para posterior documentação.

3.5. Eletroválvula de vapor tipo solenoide (ON/OFF)

A eletroválvula de controle de vapor na planta é operada através de um atuador do tipo solenoide, resultando em uma saída do tipo booleano, ou ON/OFF (válvula aberta/válvula fechada). Por outro lado, a variável de comando do PID que aciona a válvula apresenta valores que podem excursionar por todo o domínio dos números reais do CP, e que não poderia ser aplicado diretamente como o comando de saída booleano da eletroválvula.

Para compatibilizar a saída do PID com a entrada booleana da válvula, será empregada uma codificação do tipo PWM (*Phase Width Modulation*), ou seja, será aplicado um trem de pulsos entre ON e OFF, com período constante mas com um *duty-cycle* variável, cujo valor médio no período é diretamente proporcional ao valor de comando do PID.

Nesse tipo de atuação, entretanto, devem ser tomados alguns cuidados para que não sejam gerados pulsos intempestivos, ou seja, pulsos de abertura da válvula muito curtos, que podem estressar o equipamento, causando seu desgaste prematuro, ou ocasionar fadiga e sobrepressões nas tubulações da planta.

Um codificador PWM pode ser realizado através da comparação do valor de comando desejado com uma onda portadora do tipo “dente de serra”, cujo período seja adequado à dinâmica do atuador da malha de controle projetada. No caso em questão, um período de 1,0 ou 2,0 segundos é apropriado para a eletroválvula **Válvula 1**, significando que a mesma será acionada em uma frequência máxima de 0,5 ou 1,0 [Hz].

Em cada acionamento, será produzido um pulso com largura (*duty-cycle*) ajustável (entre 0,0 e 100,0[%]), a todo o momento em que o valor instantâneo da rampa portadora for inferior ao valor de comando ajustado pelo PID. Caso o valor da rampa seja superior ao valor de comando do PID, o pulso de saída permanece desligado. Isso é mostrado na Fig. 4 a seguir.

Alterando-se a forma da onda portadora é possível criar comportamentos dos mais diversos, como não linearidades ou características específicas de acionamento. Por exemplo, para fazer com que os pulsos produzidos não sejam menores que um dado período (o que poderia danificar a eletroválvula de vapor ou sua tubulação), pode ser usada uma onda portadora como a mostrada na Fig. 5.

A forma de implementação desse modulador PWM no CP pode ser feita de várias formas, desde uma tabela de valores para um temporizador como um gerador de rampa discreto e comparadores.

É importante ressaltar que o engenheiro projetista pode contar com outros tipos de eletroválvulas para comando e atuação em sistemas a hidráulicos ou a vapor, como uma válvula proporcional, cuja proporção de abertura é controlada continuamente entre 0,0 a 100,0[%], através de um sinal de tensão ou corrente enviado pelo CP.

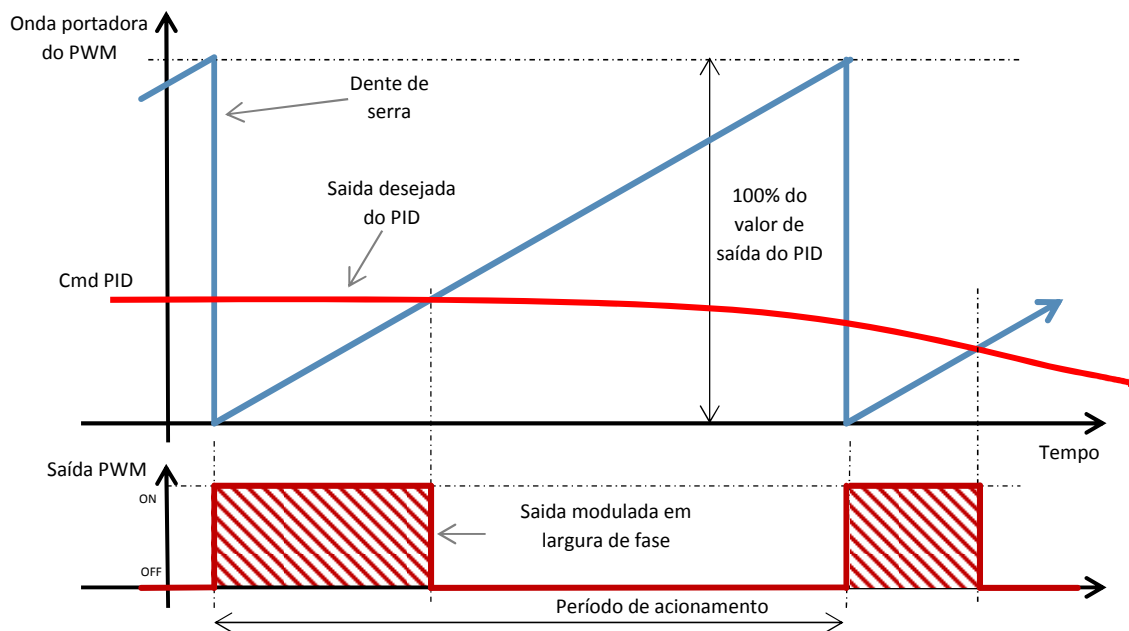


Fig. 4 - Codificador PWM para a saída do PID.

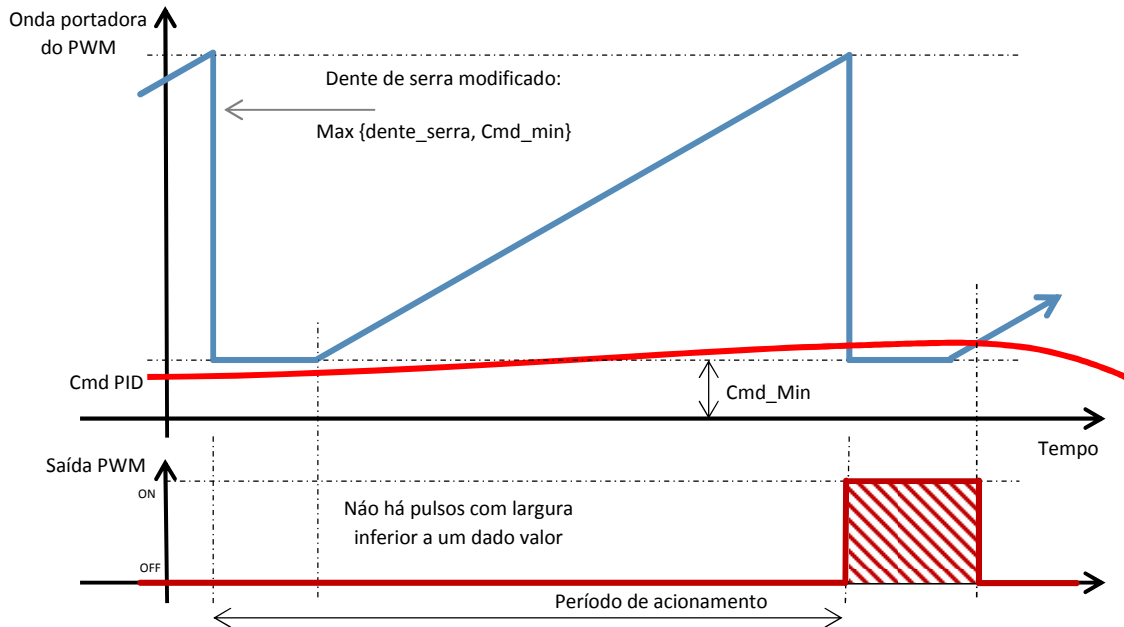


Fig. 5 – Codificador PWM modificado para a saída da eletroválvula.

3.6. Controlador de temperatura por histerese

Outra estratégia de controle mais simples que pode ser empregada é um controlador de temperatura do tipo histerese, onde a variável de controle da válvula de vapor, com saída tipo ON/OFF (válvula aberta/válvula fechada), será comandada de forma direta pelo sensor de temperatura do vaso do reator, como mostrado na Fig. 6.

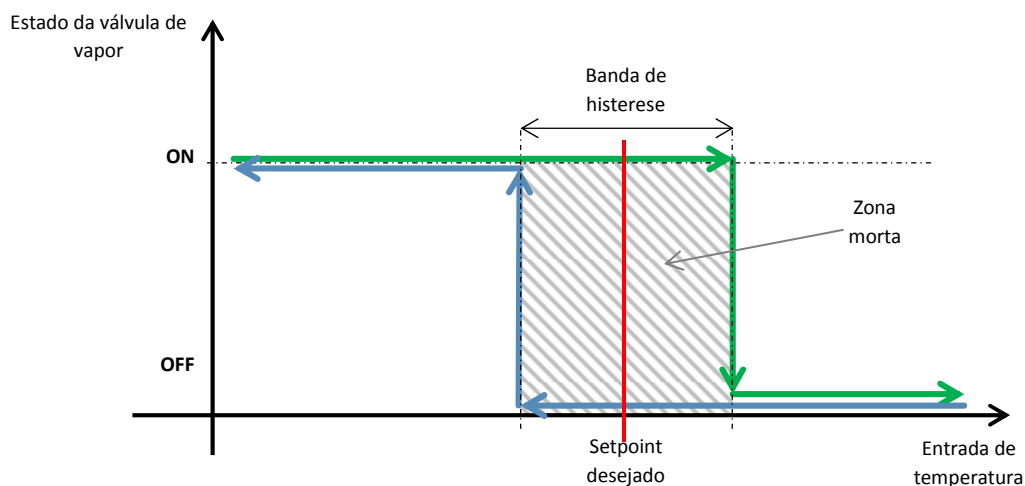


Fig. 6 – Controlador tipo histerese para a válvula de vapor.

O funcionamento desse controlador é intuitivo, ou seja, se a temperatura está baixa, o fluxo de vapor deve ser ligado; caso a temperatura esteja elevada, o fluxo de vapor deve ser desligado. Uma região de zona morta é inserida ao redor do valor desejado de *setpoint* para evitar a ação de ligar/desligar

constante da variável de saída. O resultado é uma característica oscilatória para a variável controlada de temperatura, com um *ripple* igual à banda programada, e cujo valor médio aproxima-se do *setpoint* desejado. Isso pode ser visto na Fig. 7.

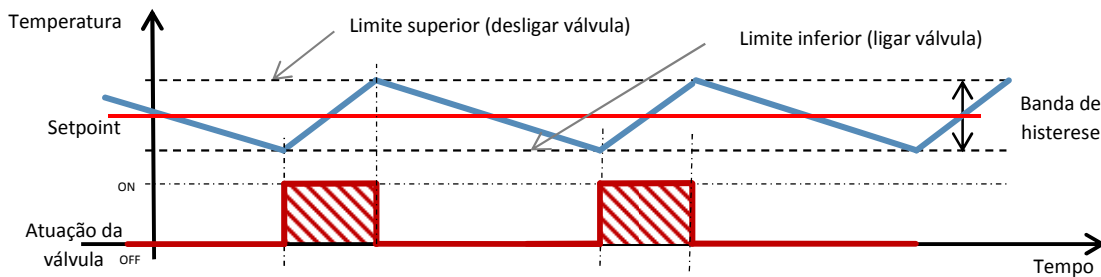


Fig. 7 – Funcionamento do controlador de temperatura por histerese.

Os alunos serão desafiados a realizar o controle do tipo histerese como atividade extra-classe para entrega junto ao relatório da experiência de laboratório, como descrito adiante.

4. Desenvolvimento em laboratório

A seguir são apresentadas as etapas de desenvolvimento do *software* no controlador programável disponível no laboratório. Os alunos devem atentar que o experimento possui diversas falhas de implementação e outros pontos questionáveis, que devem ser observados e comentados no relatório final.

Nos blocos designados como “A definir”, os alunos devem determinar as operações envolvidas e escrever as linhas de código necessárias para o funcionamento do programa. Todos os códigos devem ser apresentados no relatório da experiência. Os alunos também estão livres para adotar outros esquemas de tarefas e rotinas conforme desejado, mas devem apresentar toda a documentação de seu código.

4.1. Preparação do projeto

No software RSLogix 5000, abra o projeto PEA2509_EXP3_LAB.ACD. O projeto está incompleto e precisa receber programas e rotinas para atender aos requisitos apresentados no capítulo 2.

Inicialmente, verifique no projeto se o controlador configurado e seus cartões estão de acordo com a estrutura do CP presente no laboratório (Backplane CompactLogix, CPU 1769-L32E, Cartão 1 de entradas digitais tipo 1769-IQ16, Cartão 2 de saídas digitais tipo 1769-OB16, Cartão 3 de entradas e

saídas analógicas tipo 1769-IF4XOF2).

Clique com o botão direito do mouse sobre o cartão 3 de entradas e saídas analógicas, e escolha propriedades. Assegure-se que na tabuleta “Input Configuration” o canal 0 está habilitado, como mostrado na Fig.8. Esse canal receberá o sinal do sensor de temperatura presente no campo.

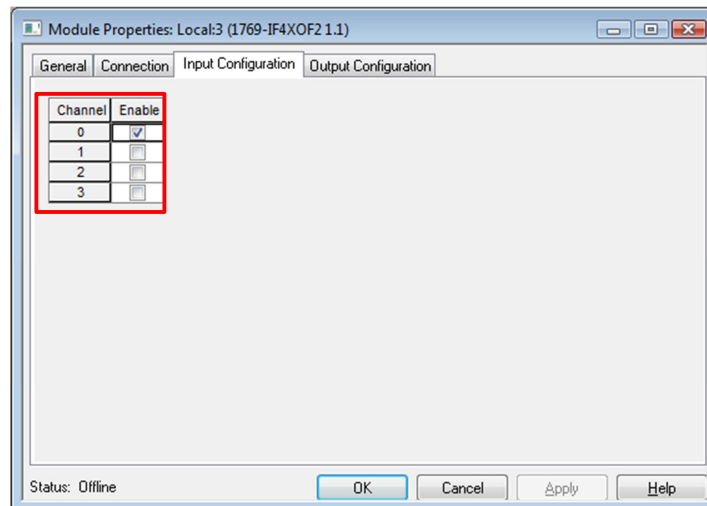


Fig. 8 – Configuração do cartão de entrada analógica.

4.2. Tarefas do controlador

O seguinte esquema de tarefas é sugerido para a implantação das funcionalidades comentadas no controlador programável:

- I. Uma tarefa principal (**MainTask**), em execução contínua, onde deve constar um programa principal (**MainProgram**), que possui duas rotinas:
 - a. A rotina principal (**MainRoutine**), que contém um código de mapeamento de entradas e saídas do controlador, feito na linguagem *Ladder*, além da chamada de outra rotina de sequenciamento, caso o sistema esteja habilitado para operação;
 - b. A rotina de sequenciamento das atividades do reator (**StateMach**), que contém uma máquina de estados codificada na linguagem SFC (*Structured Flow Chart*), com as respectivas ações por estado designadas na linguagem ST (*Structured Text*).
- II. Uma tarefa periódica (**ControlTask**), executada a cada 100 milissegundos, onde devem constar dois programas escalonados para execução:
 - a. O programa **TempControl**, que contém uma rotina de controle tipo PID (**PIDControl**), descrito através da linguagem FBD (*Function Block Diagram*);
 - b. O programa **TempCheck**, que analisa a faixa de operação de temperatura do reator em cada condição operativa, para sinalizar para a máquina de estados principal se o reator

está em condições de temperatura corretamente regulada.

- III. Uma tarefa periódica rápida (**PWMValve**), executada a cada 10 milissegundos, que contém um programa (**ValveControl**), com uma rotina (**SteamValve**) de codificação da saída de comando do PID em um padrão PWM para acionamento da válvula de vapor.

Essa estrutura é mostrada na Fig. 9.

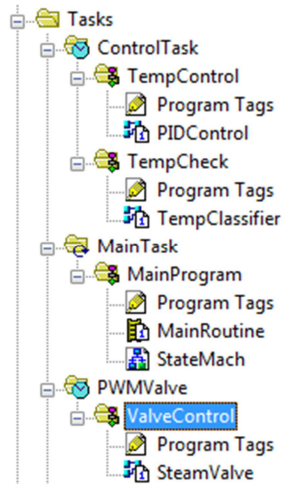


Fig. 9 - Esquema de tarefas.

4.3. Estruturas de dados e TAGs para a aplicação

As figuras mostradas a seguir mostram as TAGs que devem ser criadas nos programas e rotinas. Da forma atual são utilizados apenas dados primitivos (INT, BOOL, REAL) ao invés de estruturas de dados. Os alunos podem modificar conforme sua conveniência. Convém observar que algumas TAGs devem ser marcadas como constantes, e seus valores padrões devem ser inseridos na tableta “Monitor TAGs” da interface do RSLogix, conforme necessário.

4.3.1. Controller TAGs

Os dados globais de TAGs para o controlador são mostrados na Fig. 10.

Name	Alias For	Base Tag	Data Type	Description	External Acce	Constant	Style
HabilitaOperacao			BOOL	Condicao de operacao OK	Read/Write	<input type="checkbox"/>	Binary
IHMDesligar			BOOL	Botao IHM Desligar	Read/Write	<input type="checkbox"/>	Binary
IHMLigar			BOOL	Botao IHM Ligar	Read/Write	<input type="checkbox"/>	Binary
InBoia1			BOOL	Boia - nivel alto	Read/Write	<input type="checkbox"/>	Binary
InBoia2			BOOL	Boia - nivel medio	Read/Write	<input type="checkbox"/>	Binary
InBoia3			BOOL	Boia - nivel baixo	Read/Write	<input type="checkbox"/>	Binary
InTambor			BOOL	Tambor em posicao	Read/Write	<input type="checkbox"/>	Binary
InTerm			REAL	Temperatura Grau Celcius	Read/Write	<input type="checkbox"/>	Float
+ InTermRaw			DINT	Temperatura ADC	Read/Write	<input type="checkbox"/>	Decimal
OutAgitador			BOOL	Agitador do reator	Read/Write	<input type="checkbox"/>	Binary
OutValvula1			BOOL	Valvula de vapor	Read/Write	<input type="checkbox"/>	Binary
OutValvula2			BOOL	Valvula reagente 1	Read/Write	<input type="checkbox"/>	Binary
OutValvula3			BOOL	Valvula reagente 2	Read/Write	<input type="checkbox"/>	Binary
OutValvula4			BOOL	Valvula de saida	Read/Write	<input type="checkbox"/>	Binary
TermOK_PreHeat			BOOL	Temperatura ok no pre aquecimento	Read/Write	<input type="checkbox"/>	Binary
TermOK_PreMix			BOOL	Temperatura ok na fase de pre mistura	Read/Write	<input type="checkbox"/>	Binary
TermSetpoint			REAL	SetPoint de temperatura do controlador	Read/Write	<input type="checkbox"/>	Float
ValvePIDCmd			REAL	Comando da valvula de vapor pelo PID	Read/Write	<input type="checkbox"/>	Float

Fig. 10 - TAGs do controlador.

Basicamente as variáveis **InXXXXXX** são os mapeamentos das entradas físicas do CP para dentro da lógica. As variáveis **IHMXXXXXX** são estados enviados pela IHM PanelView ao CP para o comando de suas operações. As variáveis **OutXXXXXX** são os mapeamentos das saídas digitais da lógica do CP para os cartões de saída físicos. As demais variáveis são variáveis temporárias, tais como:

- TermOK_PreHeat – sinalizador para indicar que a temperatura do vaso está no setpoint desejado de pré-aquecimento ($100,0 \pm 10,0$ [°C]);
- TermOK_PreMix – sinalizador para indicar que a temperatura do vaso está no setpoint desejado para mistura ($200,0 \pm 10,0$ [°C]);
- TermSetpoint – variável que contém o valor desejado para o setpoint de temperatura a ser solicitado ao controlador de temperatura PID;
- ValvePIDCmd – variável que contém o valor calculado de comando para a eletroválvula pelo PID. Essa variável é contínua e deverá ser codificada para o trem de pulsos PWM para finalmente acionar a saída digital da válvula de valor (OutValvula1);

4.3.2. TempControl TAGs

As TAGs para a rotina de controle de temperatura que contém o PID são mostradas na Fig. 11.

PI_KD			REAL	Ganho do termo derivativo	Read/Write	<input checked="" type="checkbox"/>	Float
PI_KP			REAL	Ganho PI KP	Read/Write	<input checked="" type="checkbox"/>	Float
PI_WLD			REAL	Freq. PI	Read/Write	<input checked="" type="checkbox"/>	Float
						<input type="checkbox"/>	

Fig. 11 - TAGs para o PID.

As TAGs do PID mostradas acima são constantes que armazenam os ganhos do controlador PI (PI_KP e

PI_WLD) e o ganho do termo derivativo (PI_KD). Após criadas as TAGs, na parte inferior da tela, em “Monitor TAGs”, preencha os valores padrão da coluna “Value” com os valores mostrados na Fig. 12.

Name	Value	Force Mask	Style	Data Type	Description	Constant
PI_KD	1.0		Float	REAL	Ganho do termo ...	<input checked="" type="checkbox"/>
PI_KP	1.0		Float	REAL	Ganho PI KP	<input checked="" type="checkbox"/>
PI_WLD	0.2		Float	REAL	Freq. PI	<input checked="" type="checkbox"/>

Fig. 12 – Valores dos ganhos para o PID.

O significado físico desses ganhos depende da forma como o controlador PID é construído. Mais detalhes podem ser obtidos em (Automation, 2012).

Para melhor análise do desempenho e funcionamento do controlador, o aluno poderá criar outras TAGs para armazenar os valores calculados pelo controlador ao longo do tempo. O programa do RSLogix possui gráficos de tendências configuráveis para esse papel de comissionamento.

4.4. Rotina principal do programa principal (MainRoutine)

O programa principal, responsável por mapear as entradas e saídas digitais e analógicas é mostrado nas Fig.13, 14, 15 e 16 a seguir, em linguagem LADDER.

Nos trechos de código mostrado nas figuras, deve-se atentar que:

- Os nomes (endereços) das entradas digitais físicas do CP devem ser corretamente atribuídos às respectivas variáveis internas (ao invés de Local:3:I ... coloque o endereço e bit corretos da sua montagem);
- O nome (endereço) da entrada analógica 0 do cartão de entradas deve ser corretamente atribuído no código à variável interna InTempRaw, mais especificamente na instrução MOV mostrada no final da Fig. 13.

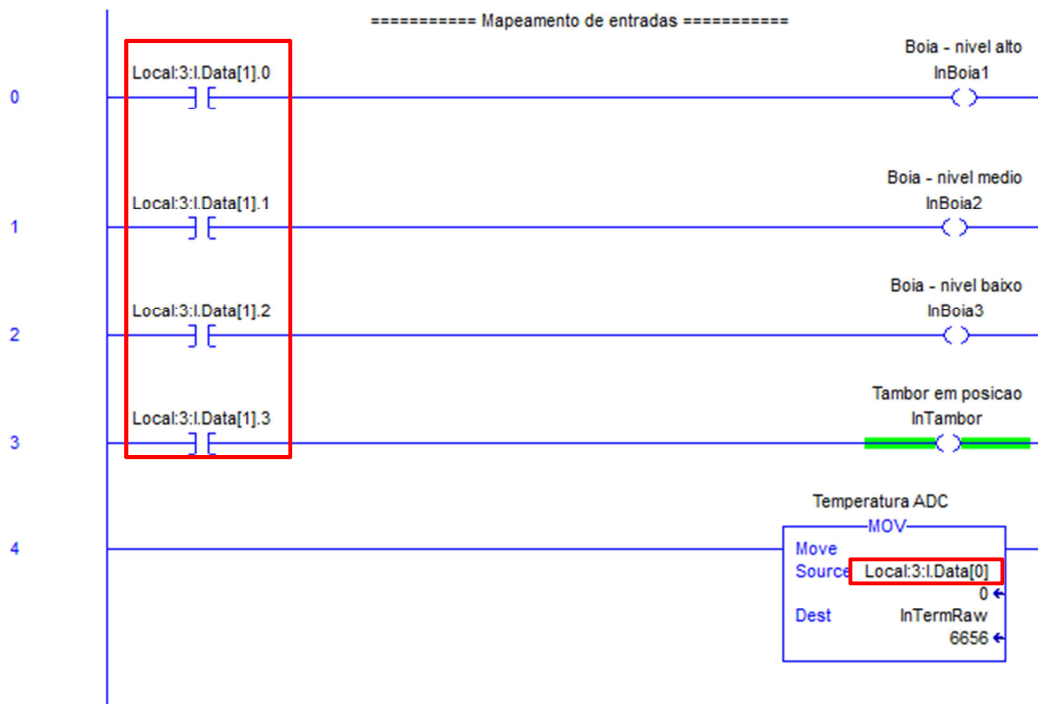


Fig. 13 – Atribuição de entradas digitais e analógica.

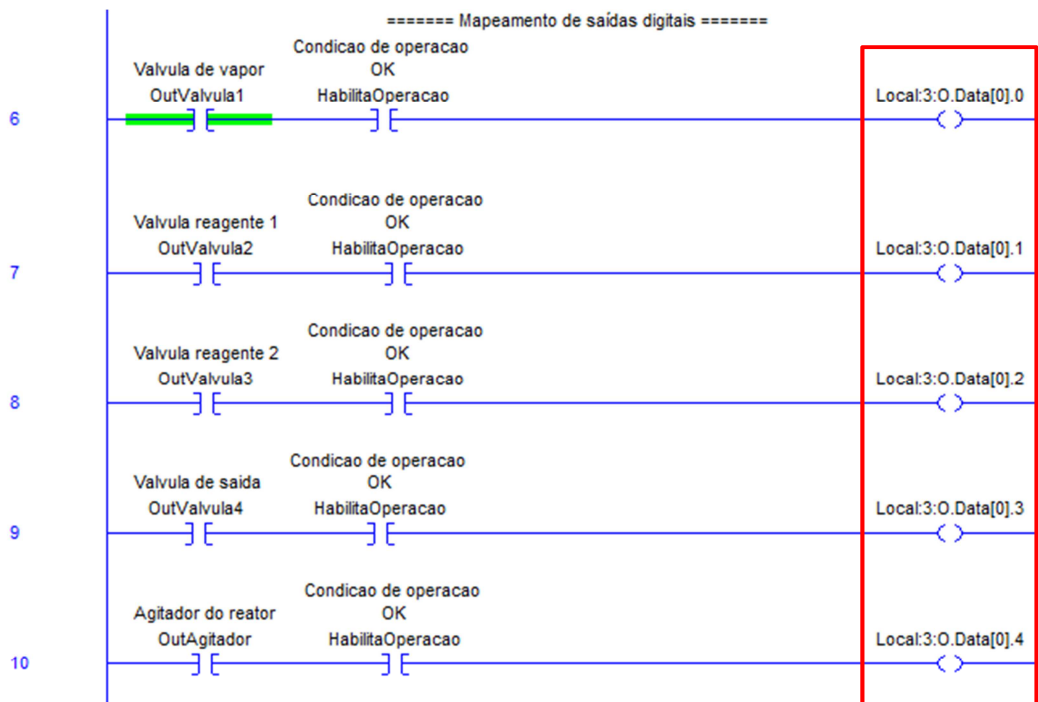


Fig. 14 – Atribuição de saídas digitais.

Note que o acionamento das saídas digitais reais do CP estão condicionadas ao estado de uma variável interna da lógica denominada *HabilitaOperacao*. A lógica que define essa variável é mostrada adiante.

Na Fig. 15 é mostrado o cálculo conveniente da variável interna InTerm, em valor de ponto flutuante, no Sistema Internacional de unidades, para o uso de valores de engenharia nas malhas de controle.

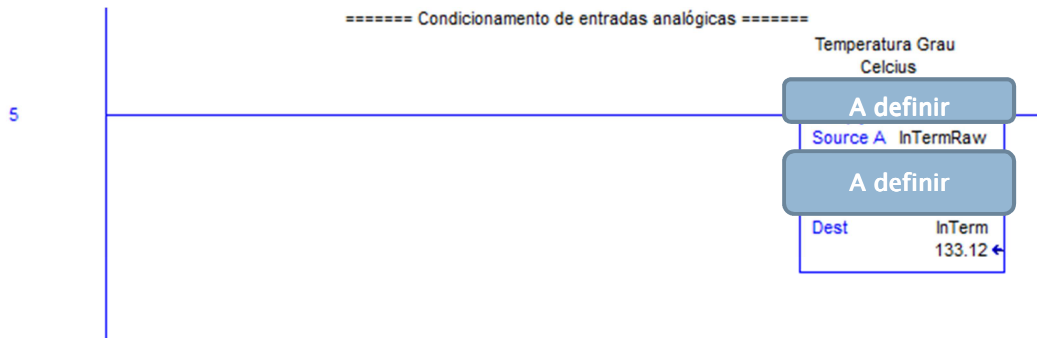


Fig. 15 – Cálculo da variável representando o sensor de temperatura em valores de engenharia.

No trecho de código a seguir é mostrada a lógica de operação do sistema e da rotina de máquina de estados descrita adiante.

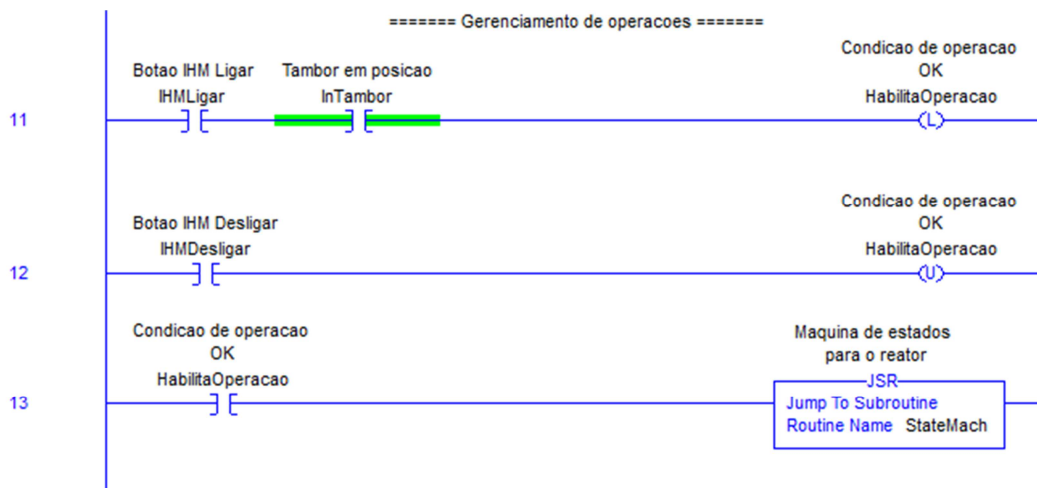


Fig. 16 – Lógica de operação e execução de outras rotinas.

4.5. Rotina de sequenciamento do programa principal (StateMach)

O programa responsável pelo sequenciamento de operações é mostrado na Fig.17 a seguir, em linguagem SFC. Foram utilizados diversos recursos de documentação (como comentários, descrição de variáveis, estados, ações, mudança do nome dos estados, etc.), para favorecer o entendimento de seu funcionamento e posterior depuração de sua operação.

Deve-se notar que a interconexão de alguns estados foi ocultada para melhorar a diagramação visual. Isso pode ser feito clicando-se com o botão direito sobre um caminho entre estado e/ou ação e escolhendo-se a opção “Hide Wire”.

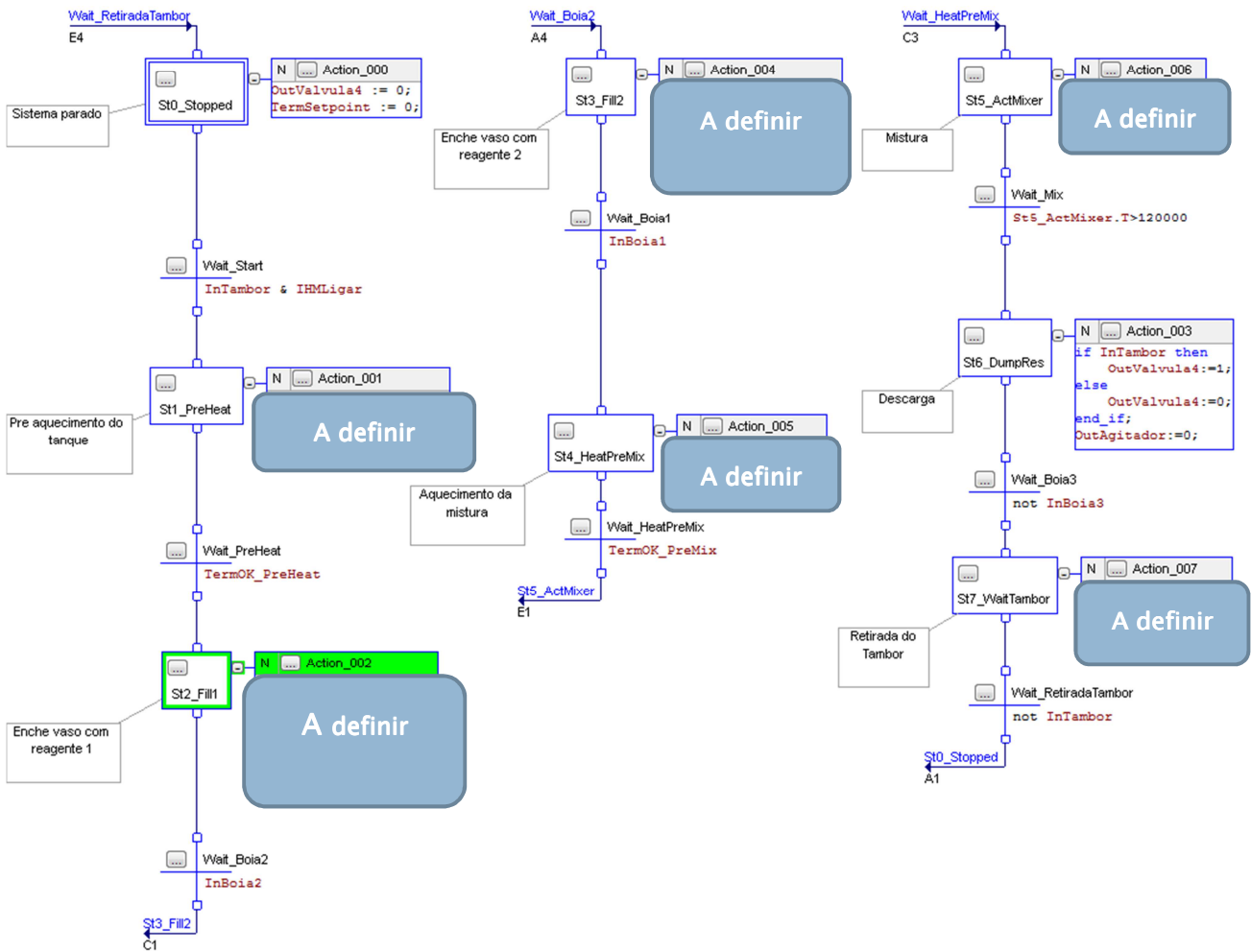


Fig. 17 - Máquina de estados de sequenciamento.

Para aguardar o período de tempo de 120 segundos necessário para a mistura dos reagentes, nessa experiência, ao invés do emprego de um Timer usual, foi utilizada a propriedade 'T' do estado ST5_ActMixer. Essa propriedade designa o período de tempo que o respectivo estado está ativo. Na condição Wait_Mix é colocada a expressão lógica `St5_ActMixer.T > 120000` como condicionante para a transição para o estado seguinte.

4.6. Rotina do classificador de temperatura (TempClassifier)

A rotina **TempClassifier** deverá ser desenvolvida com a linguagem FBD (*Function Block Diagram*). O FBD descreve um fluxo de dados entre entradas e saídas de quaisquer tipos, através de blocos de controle e funções encapsuladas. O resultado é um verdadeiro diagrama de blocos, que pode utilizar variáveis do CP (Reais, Booleanas, etc.) e todo o conjunto de blocos funcionais criado pela fabricante do CP. Entre as vantagens dessa descrição algorítmica por meio de diagramas de blocos estão as evidentes facilidades para sua interpretação, manutenção e alteração.

Para o CP do laboratório, entre as inúmeras bibliotecas de blocos disponíveis, constam blocos

aritméticos combinatórios (*Compute/Math*), cujas saídas só dependem das entradas; até blocos sequenciais e aritméticos complexos (como *Timers*, *Filtros*, *Drives*, etc.), que empregam valores e estados anteriores para constituir uma dada saída.

Para a rotina *TempClassifier*, o objetivo é criar um classificador para a temperatura do reator, para verificar se a mesma se encontra entre os limiares de operação desejados, seja para a fase de admissão de reagentes ($100,0 \pm 10,0$ [°C]), seja na fase de mistura final ($200,0 \pm 10,0$ [°C]).

Para se criar esse fluxo de dados no FBD, basta colocar blocos conectores para as representar as entradas de dados do diagrama (*Input Reference*), os respectivos blocos funcionais que constituem o fluxo de processamento, e blocos conectores para armazenar os resultados em saídas (*Output Reference*). As ligações entre esses elementos é feita com fios (*wires*) através de recursos simples de arrastar e soltar. Fios contínuos representam dados REAIS (ponto flutuante) e fios tracejados, dados BOOLEANOS. No caso da rotina, será utilizado um bloco do FBD denominado **LIM**, responsável por checar de uma variável REAL de entrada (*Test*) está situada entre dois limites, um limite inferior (*LowLimit*) e um limite superior (*HighLimit*). O resultado dessa classificação (*Dest*) é disponibilizado como saída do bloco, que por sua vez pode ser atribuída a uma variável booleana (TAG) do CP. Isso pode ser visto na Fig. 18.

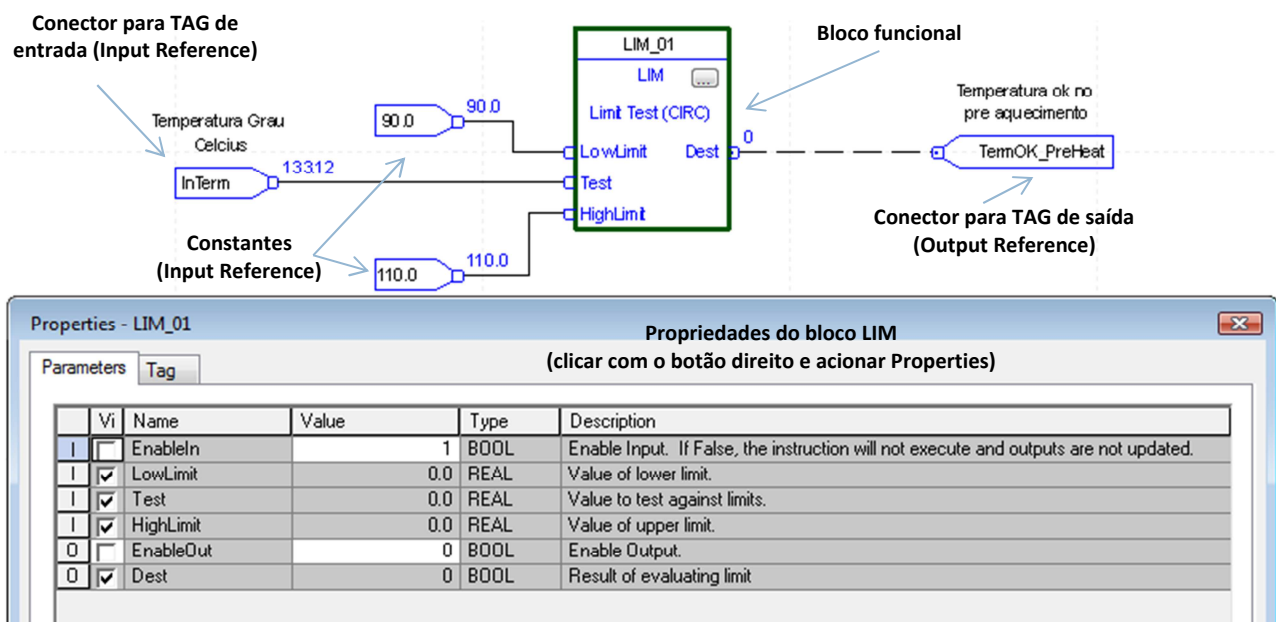


Fig. 18 - Implementação de um classificador de temperatura no FBD.

Na Fig. 18 pode-se notar a presença de um bloco do tipo *Input Reference*, responsável por citar a TAG para entrada de dados no FBD, no caso, com o nome da variável 'InTerm'. Além disso, outros dois blocos *Input References* foram criados para armazenar as constantes 90,0 e 110,0, que representam os limites baixo e alto para a lógica. O bloco LIM é obtido da paleta de blocos (*Element Group*)

denominada *Compare*. A saída do bloco LIM é ligada a um bloco tipo *Output Reference*, que indica em que TAG o resultado do bloco deve ser armazenado, no caso na variável booleana 'TempOK_PreHeat'.

Atenção: Além do mostrado na Fig. 18, o aluno **TAMBÉM** deve criar outra estrutura semelhante, para classificar a mesma variável 'InTerm' entre os limites de 190,0 e 210,0 [°C], para atribuição à variável 'TermOK_PreMix'.

É importante frisar que os blocos funcionais possuem instruções de ajuda, exibidas com a tecla F1, além de uma extensa folha de propriedades, obtida pressionando-se o botão direito do mouse sob o bloco, e escolhendo *Properties* no menu. Nessa folha de propriedades, pode-se habilitar ou desabilitar entradas, além de se fornecer valores para parâmetros internos de funcionamento de cada bloco. Isso é mostrado na parte inferior da Fig. 18.

4.7. Rotina do codificador PWM (SteamValve)

A rotina **SteamValve** deverá ser desenvolvida com a linguagem FBD (*Function Block Diagram*) para criar o codificador PWM do valor de saída que será produzido pelo controlador PID. Para isso, utilizaremos um bloco de timer (idêntico àqueles já usados nas experiências anteriores em LADDER), com comparadores e seletores, para criar a variável de saída OutValvula1 que aciona o fluxo de vapor no reator. Na Fig. 19 é mostrado o diagrama de blocos resultante para o codificador PWM de **SteamValve**.

Deve-se atentar que na construção de um diagrama FBD, as interconexões internas entre os inúmeros blocos, entradas e saídas podem se tornar bastante confusas, com claro prejuízo para a diagramação visual, como por exemplo quando os fios de conexão passam a cortar e atravessar blocos, se interseccionar, etc. Para facilitar essa diagramação, além de entradas e saídas comuns existem conectores internos de entrada e saída temporárias (*Input Wire Connector* e *Output Wire Connector*). Quando usados, indicam ao software de/para onde os sinais podem trafegar dentro do diagrama FBD, sem que sejam criadas variáveis temporárias. Esses blocos podem ser posicionados livremente dentro do diagrama FBD para facilitar a interconexão dos seus elementos. Alguns desses elementos podem ser vistos na Fig. 19.

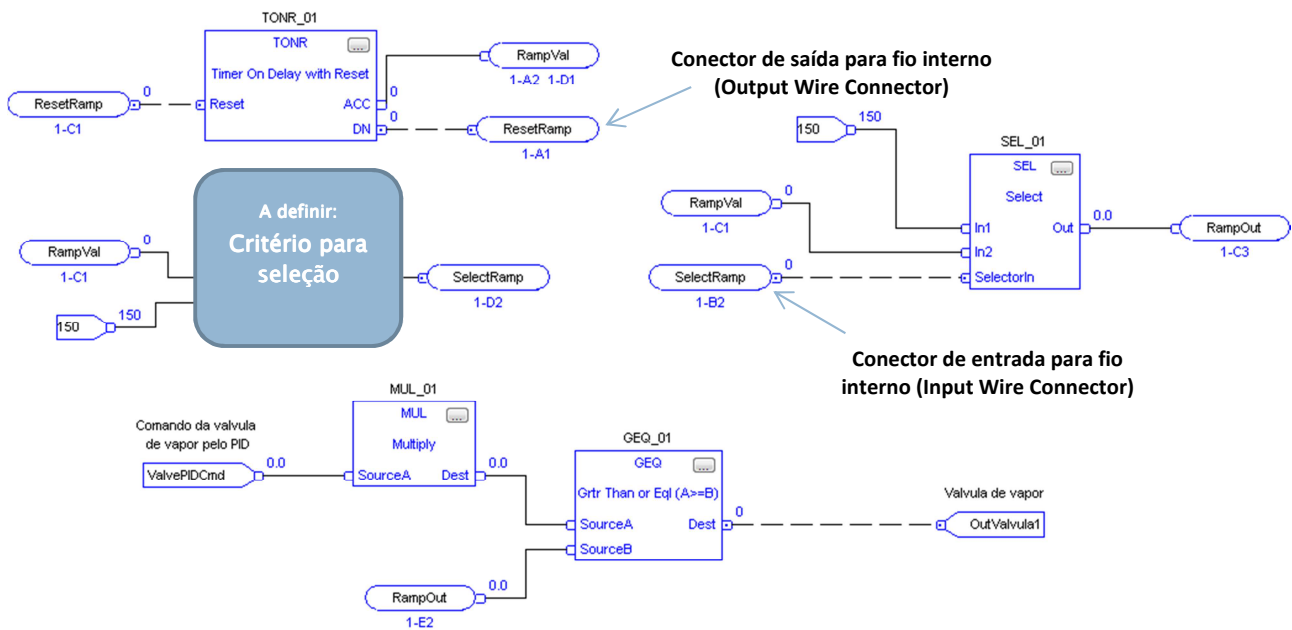


Fig. 19 – Lógica FBD para o codificador PWM do CP.

Deseja-se obter uma rampa crescente, auto-resetável, com amplitude de 1000 unidades e período de 1,0 [s]. A rampa deve ser limitada a um valor mínimo de 150 unidades, para obter-se o efeito mostrado na Fig. 5. A saída de comando do controlador PID ‘ValvePIDCmd’ é uma entrada para esse FBD, com amplitude entre 0,0 a 1,0. Essa TAG deve ser multiplicada por um ganho (multiplicador) com valor 1000,0, para então ser comparado com a rampa para a geração do sinal ‘OutValvula1’. Os alunos devem desenvolver o diagrama mostrado na Fig. 19 e atentar para as propriedades internas de cada bloco para satisfazer esses requisitos.

4.8. Rotina do controlador PID (PIDControl)

A rotina **PIDControl** deverá ser desenvolvida com a linguagem FBD (*Function Block Diagram*) para criar o controlador PID de temperatura do reator. Para isso será desenvolvido um controlador PID utilizando-se um bloco elementar de controlador Proporcional-Integral (PI), um bloco de filtragem derivativa (DERV), um bloco somador (ADD) e um bloco de limitação/saturação (HLL). O bloco HLL garante que a saída do PID fique restrita a um dado domínio desejado, no caso entre 0,0 e 1,0.

Deve-se atentar que o bloco PI utiliza como entrada o sinal de erro entre o valor de referência do controlador de temperatura ‘TermSetpoint’ e o valor instantâneo de feedback de temperatura da planta ‘InTerm’. Esse sinal de erro também precisa ser constituído através de uma operação algébrica entre essas entradas.

O aluno deve desenvolver esse diagrama FBD como mostrado na Fig. 20, novamente atentando para os parâmetros internos de um ou outro bloco, para resultar no desempenho desejado.

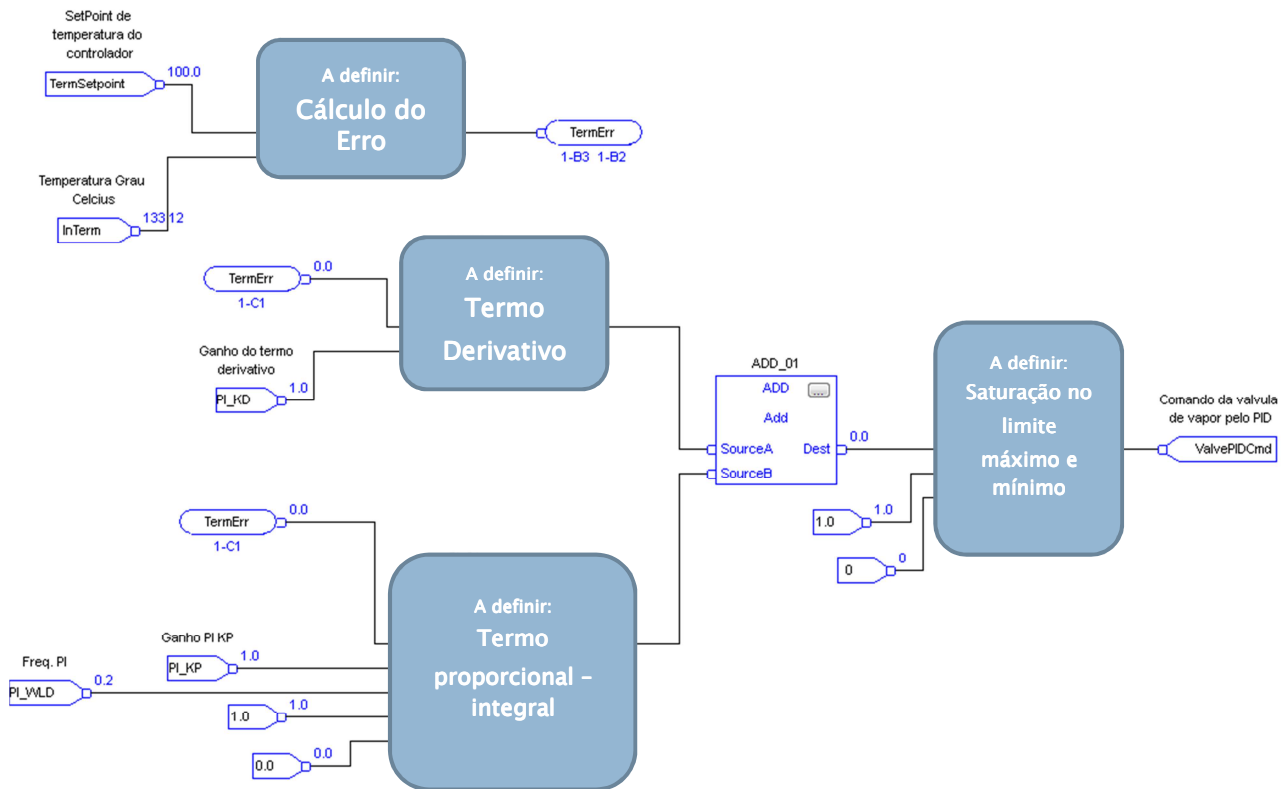


Fig. 20 – Lógica FBD para o controlador PID.

É importante que entre as propriedades do bloco do RSLogix PI sejam ajustados os limites máximos e mínimos do termo integral para os valores desejados de 1,0 e 0,0 para evitar o efeito de “Wind up”.

4.9. Interface IHM

Os alunos devem utilizar uma interface IHM desenvolvida para o PanelView presente na bancada. Tal interface foi desenvolvida para um determinado conjunto de TAGs presentes na aplicação do CP. Caso as TAGs criadas pelos alunos tenham nomes diferentes das especificadas nessa guia de laboratório, os mostradores gráficos poderão funcionar de forma incorreta. Nesse caso, sugere-se a mudança nas TAGs na aplicação ou no software do FactoryTalk View ME.

Para proceder, o aluno deve clicar duas vezes sobre o arquivo EXP3_IHM_PV.APA presente na pasta de arquivos para alunos. Será aberto o software *Application Manager*. Nesse, escolha a primeira opção “*Restore the Factory Talk View Machine Edition Application*”. Proceda através das telas de confirmação, escolha um nome para a aplicação (pode ser aceito o nome padrão se esse já não existir), e clique em *Finnish*.

Em seguida, abra o software *Factory Talk View Studio*, escolha as aplicações tipo *Machine Edition*, e selecione na lista de “*New/Open Machine Edition Application*” o nome escolhido anteriormente para sua aplicação IHM. Será aberta a aplicação, com uma tela principal como a mostrada na Fig. 21.

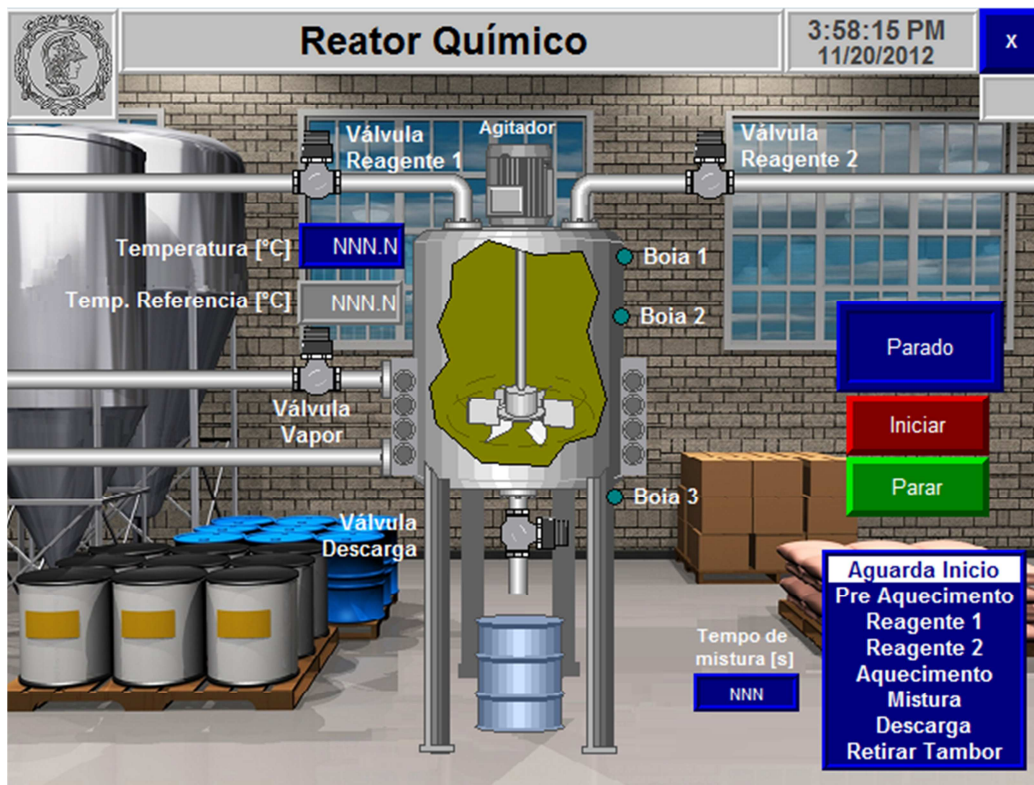


Fig. 21 – Tela da IHM PanelView para o experimento.

Os alunos podem testar e modificar a aplicação no software *FactoryTalk View Studio*, conforme a necessidade. Ao final, deve-se criar a aplicação embarcada que é carregada no PanelView, acionando-se o menu “*Application*”, opção “*Create Runtime Application*”. Será gerado um arquivo com extensão .MER em algum diretório segundo a escolha do usuário.

Com o arquivo .MER criado, pode-se descarregar a aplicação desenvolvida no PanelView clicando-se no ícone da barra de ferramentas denominado “*Transfer Utility*”. Na tabuleta “*Download*”, escolha a aplicação .MER desenvolvida, coloque como destino “*Internal Storage*” e escolha o terminal para a transferência na árvore de opções mostrada na parte de baixo da janela. Assegure-se que a opção “*Run application at start-up*” esteja desabilitada e acione a tecla “*Download*”.

Reinicialize o PanelView para que as mudanças surtam efeito. Durante a inicialização, na tela do PanelView, siga as instruções na tela para executar o aplicativo.

4.10. Simulador do reator para ligação no CP

O laboratório dispõe de uma giga de testes para emulação da planta do reator químico como mostrado na Fig. 22. Os alunos devem fazer as ligações corretas, incluindo os sinais de +24VCC e -24VCC que podem ser obtidos no painel de montagem do CP.

Com essa giga de testes um dos alunos do grupo fará o papel de PLANTA, emulando os efeitos dos

sinais das boias (chaves), da posição do tambor (slider acrílico com a figura do tambor) e do sensor de temperatura (potenciômetro).

Os alunos devem verificar a atuação do CP nas saídas que atuam sobre as válvulas de 1 a 4 e sobre o agitador.

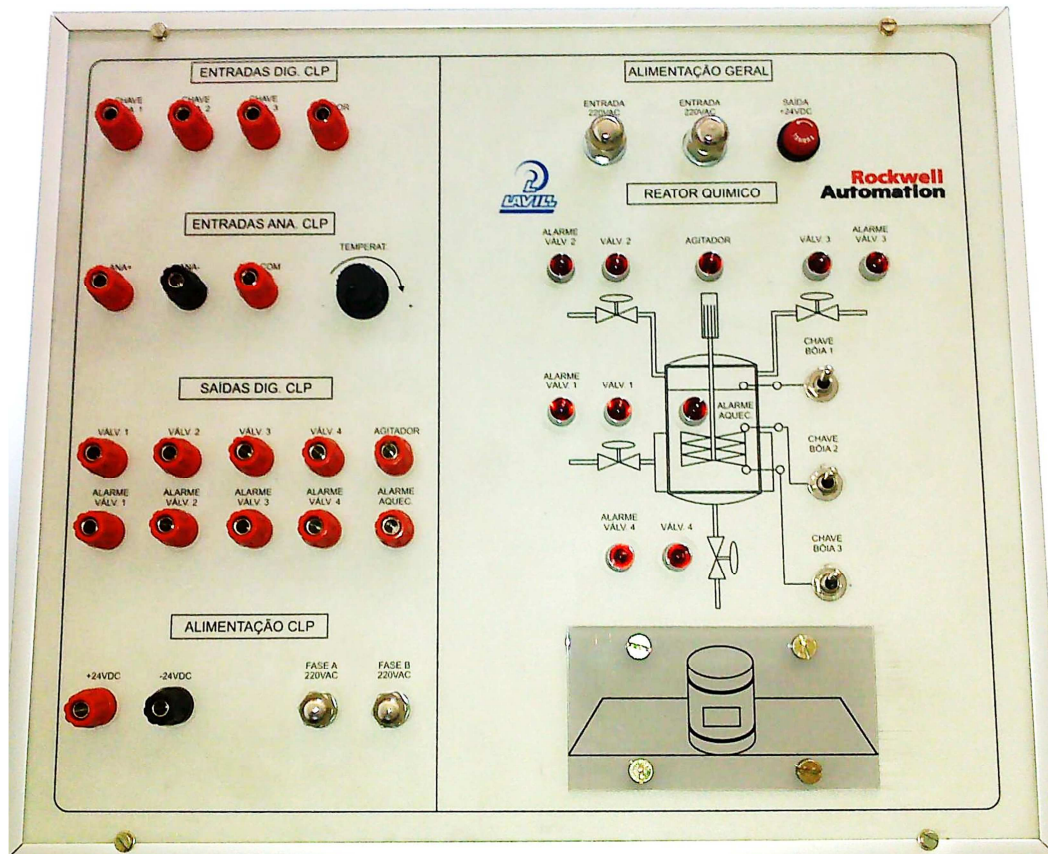
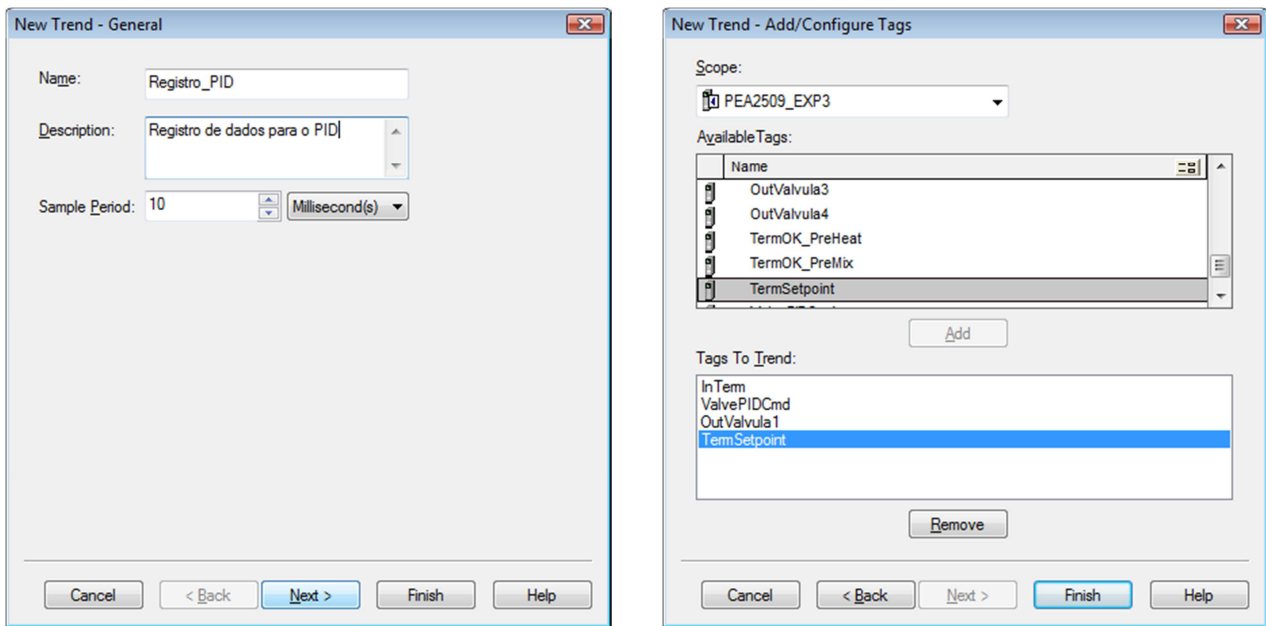


Fig. 22 – Giga de testes para simulação das operações do CP com o reator.

4.11. Gráficos e registros para análise do PID

O software de programação do RSLogix5000 possui uma ferramenta de depuração que permite o registro de formas de onda para as variáveis internas do CP durante sua execução. Esse recurso pode ser acessado na árvore de opções do projeto, no item “Trends”. Nesse item, clique com o botão direito e escolha “New Trend”, forneça um nome para o registro, descrição e sua periodicidade de coleta de dados.

Em seguida, dentro dos diversos escopos de tarefas e programas do CP, escolha as TAGs que deseja registrar e clique em “Add”, colocando-as na lista na parte inferior da tela, como mostrado nas Fig. 23 a e b.



(a) Nome do registro e periodicidade de captura.

(b) TAGs a capturar. Até 8 podem ser escolhidas.

Fig. 23 – Criação de um registro gráfico de tendência para as variáveis desejadas.

Clicando-se com o botão direito na “Trend” recém criada, e acionando-se suas propriedades, é possível configurar outros recursos, como cores para as linhas, escalas, etc. Nessa propriedades, na tabuleta “X-Axis”, escolha um “Time Span” de 60 a 100 segundos. Explore as demais propriedades.

Quando um código é enviado ao CP (*Download*) e o mesmo é colocado em modo *Remote Run*, com supervisão *Online*, ao clicar-se duas vezes com o botão esquerdo sobre uma “Trend”, o usuário tem acesso a uma janela como a mostrada na Fig. 24.



Fig. 24 – Trend pronta para registrar os dados do CP.

Clicando-se sobre o botão *Run* da *Trend* é dado início ao processo de coleta de dados. O engenheiro pode então analisar o desempenho da planta com dados quantitativos sobre tempos de subida, descida, sobressinal e outros dados da dinâmica do processo sob controle. Os registros podem ser parados, exportados e impressos.

Para verificar o funcionamento do PID, os alunos são motivados a criar uma ou mais Trends para registrar o comportamento do controlador mediante as variações manuais de temperatura da planta. Sugere-se que sejam criadas outras variáveis (TAGs) do tipo REAL no CP para registrar outros dados do controlador PID, como o valor do Erro, valor do comando enviado pelo termo proporcional-integral, valor do comando do termo derivativo, etc. Os registros devem ser então ser capturados e anexados ao relatório final.

5. Atividades para execução do relatório

Os seguintes pontos devem constar no relatório da experiência:

- Data, nome dos integrantes e número da bancada.
- Captura das telas dos códigos implementados em Ladder, SFC e FBD para o projeto, com as respectivas explicações e comentários sobre seu funcionamento;
- Comentários sobre o funcionamento da planta e dos algoritmos implementados;
- Verifique se todos os requisitos do processo, apresentados no capítulo 2, são atendidos pelo código elaborado. Proponha soluções para as eventuais não conformidades;
- Conclusões sobre o experimento;
- Resultados da atividade extra-classe descrita a seguir.

6. Atividade extra-classe para entrega posterior

Os alunos devem desenvolver um código em FBD para um controlador do tipo histerese, citado no item 3.6, para ser utilizado no lugar do controlador PID e codificador PWM desenvolvidos durante a experiência de laboratório. **As telas do código em FBD para o controlador por histerese devem ser capturadas e entregues junto do relatório da experiência.**

Para facilitar esse desenvolvimento, e até permitir testes do algoritmo de controle, no *Moodle* da disciplina serão disponibilizados alguns arquivos de apoio, incluindo um projeto completo no RSLogix 5000, contendo os códigos básicos do CP da experiência 3, e arquivos com a IHM PanelView utilizada. Além desses é fornecido um documento denominado “Guia para uso do RSLogix Emulate 5000”.

No arquivo do projeto do RSLogix 5000 fornecido é feita uma abordagem muito comum em projetos de engenharia que consiste em utilizar o próprio CP para simular o comportamento da planta, enquanto são desenvolvidos os algoritmos para seu controle. Para isso foi criada uma tarefa periódica especial, contendo um modelo matemático para o reator químico. Essa tarefa é agendada para execução junto das demais tarefas do controlador e, dessa forma, sem nenhuma Giga adicional ou entradas e saídas físicas, o usuário engenheiro pode desenvolver o software do CP, com resultados simulados e em tempo real.

O controlador, por sua vez, pode ter seu comportamento emulado em um computador pessoal comum com o uso do software *RSLogix Emulate 5000*, instalado nos computadores da Sala Energia. Sua utilização é explicada brevemente no “Guia para uso do RSLogix Emulate 5000”.

As telas fornecidas para o sistema supervisão também podem ser usadas na Sala Energia através do recurso “*Test Application*” encontrado na barra de tarefas do aplicativo *FactoryTalk View*.

7. Bibliografia

Automation, R. (Novembro de 2012). *Manuais de usuário e de referência para programação no RSLogix 5000*. Acesso em Novembro de 2012, disponível em RSLogix Software:

<http://www.rockwellautomation.com/rockwellsoftware/design/rslogix5000/>

Bradley, A. (Novembro de 2012). *Manuais de usuário e de referência para os controladores CompactLogix série 1769*. Acesso em Novembro de 2012, disponível em Allen Bradley Homepage:

<http://ab.rockwellautomation.com/Programmable-Controllers/CompactLogix>

Ogata, K. (2011). *Engenharia de Controle Moderno*. São Paulo: Pearson Educations.