

Gestão do controle com requisitos temporais de um tanque para fluidos

Daniel do Carmo Figueiredo - Matrícula: 160057922

Victor Hugo Marques Vieira - Matrícula: 150047649

Abstract

The importance of time requirements for the control of fluid storage tanks is discussed. Processing tasks and time requirements are established for the control management of a vertical cylindrical tank, in relation to the physical equipment present in the tank. The management of these tasks is implemented through a loop with interruptions. The implemented system presented answers compatible with the established time requirements.

1. Introdução

Diversos tipos de controladores são utilizados em ambientes industriais, com a intenção de controlar parâmetros de um processo. Estes parâmetros podem ser os mais diversos possíveis, a depender do tipo de processo abordado.

Para estes diversos tipos de desenvolvimentos há também diferentes tipos requisitos aos quais devem ser planejados antecipadamente, para que a função de controle seja aplicada com sucesso. Estes cenários devem tanto ser abordados individualmente, como sistematicamente, para garantir que o sistema funcione como devido para suas finalidades.

Neste trabalho é abordado a estrutura de um software responsável pela administração dos atributos, necessários para o controle da vazão de um tanque de armazenamento de líquidos, utilizados para processos industriais.

Entre todos os requisitos essenciais para este modelo de processo, serão discutidos neste trabalho os requisitos temporais. Estes requisitos são de extrema importância para a administração do controle de processos industriais por diversos motivos, entre eles:

- **Controlador:** os controladores modernos atualmente utilizados na indústria, via de regra, são digitais. Desta forma para o bom desempenho destes controladores é de extrema importância que os requisitos temporais sejam satisfeitos completamente.
- **Segurança:** neste tipo de processo o monitoramento dos parâmetros de segurança devem ser realizados com a frequência correta, ou seja, devem ser realizados

em períodos predeterminados conforme a necessidade do projeto, visando garantir os padrões de segurança necessários.

1.1. Tanques de armazenamento em industria

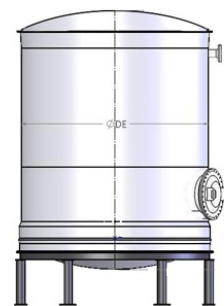
Tanques são equipamentos estáticos utilizados para o armazenamento de fluidos. Estes fluidos podem dos mais diferentes tipos, tal como combustíveis, gases, insumos para processos químicos, subprodutos de processos, resíduos etc.

O formato dos tanques de armazenamento variam conforme as necessidades do processo que será realizado e das características do produto armazenado. Entre seus formatos mais comuns estão cilindros verticais e horizontais, abertos ou fechados, com fundo plano, côncavo ou convexo, teto fixo, flutuante ou móvel, entre outros [4].

Neste trabalho será utilizado como modelo um tanque cilíndrico vertical, utilizado para a vazão de insumo para processo químico de mistura. Optou-se por abordar este cenário, pois é um padrão corriqueiro de projetos para processos industriais, assim como um modelo de controle aplicável em diversos setores.

A imagem 1 apresenta um exemplo para este modelo de tanque:

Figure 1. Exemplo de tanque cilíndrico vertical



No cenário que iremos abordar este tanque será responsável por armazenar o insumo utilizado em um processo químico de mistura. O insumo presente neste tanque deverá ser liberado de forma controlada, para que o processo de mistura a ser realizada em outro tanque de maior

capacidade seja realizada com a adição dos demais insumos necessários para que a reação química ocorra como esperado.

Para que este cenário seja possível o tanque possui alguns dispositivos de controle instalados, para o sensoramento do processo e controle da vazão para a conexão do tanque com o processo. Entre estes dispositivos:

- Controlador de vazão: controla a vazão de saída do fluido. Usualmente utiliza-se um controlador PI para tanto.
- Sensor Temperatura.
- Sensor Altura do fluido.
- Sensor Pressão.
- Sensor vazão.

O controlador de vazão possui integrado em si um sensor de vazão, porém estamos instalando um segundo sensor independente para o acompanhamento externo desta grandeza e um melhor monitoramento.

2. Modelo desenvolvido

Precisamos definir quais os requisitos temporais presentes neste sistema, para que a estrutura do software escolhido atenda da melhor forma possível tais requisitos temporais.

Devemos definir as threads que devem ser processadas neste programa, e suas propriedades temporais, para sabermos a respeito da melhor solução possível.

As threads de processamento de controle deste tanque estão associadas aos seus dispositivos presentes, de modo que podemos definir:

- Thread do Controlador vazão.
- Thread do Sensor Temperatura.
- Thread do Sensor Altura do fluido.
- Thread sensor Pressão.
- Thread Sensor vazão.

Duas outras situações carecem de definição. Esta gestão do controle carece ainda de tratativas relacionadas a interrupção do processo. Dois tipos de interrupção são factíveis:

- Interrupção ocasionada pela parada normal do processo devido a tempo ou fim do processo controlado.
- Interrupção ocasionada pelo acionamento de um botão de emergência, ou devido a leitura de parâmetros perigosos dos sensores do processo.

2.1. Estrutura do programa

As threads definidas nesta aplicação podem ser classificadas em dois tipos:

- Threads periódicas que demandam um tempo limite de tempo de processamento que variam de um valor mínimo até um valor máximo.
- Threads de interrupção que devem ter baixo tempo de resposta, porém são de acionamento único durante a execução do programa.

Estes dois tipos de processos são um indicativo de que uma boa solução para o software proposto pode ser alcançada utilizando uma implementação no formato de Laço Principal com interrupções. Nesta implementação simples de tarefas periódicas, constrói-se um ciclo que atenda aos requisitos temporais do sistema, levando em consideração os possíveis cenários de interrupção [3].

Foi definido para essas threads o seu tempo de processamento no pior caso (WCET), e sua periodicidade. Estes valores foram definidos arbitrariamente, porém levando em consideração padrões comuns de projetos similares desenvolvidos no passo. A tabela 1 apresenta os valores definidos em microssegundo:

Table 1. Parâmetros das threads processadas

Thread	Duração pior caso	Período
Sensor Altura	100 ms	1000 ms
Sensor Pressão	100 ms	1000 ms
Sensor Vazão	100 ms	1000 ms
Sensor de Temperatura	100 ms	1000 ms
Controlador de Vazão	300 ms	650 ms
Atualização Interrupções	20 ms	não periódica

Escolheu-se o menor período entre as thread para ser o período do Laço principal da estrutura do programa. Desta forma garantimos que esta será sempre atendido, enquanto encaixamos os demais nos tempos remanescentes. Com esta iniciativa montou-se o pseudocódigo apresentado na figura 2:

Desta forma garantimos que as interrupções sempre serão tratadas juntamente com as demais threads, em qualquer situação as interrupções serão tratadas em pelo menos 300 ms, tempo de pior caso a thread referente ao controle da vazão.

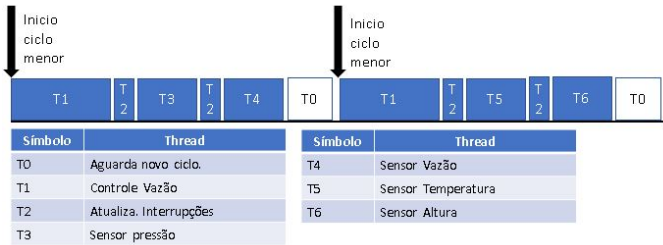
Uma linha do tempo com as threads analisadas é apresentada na figura 3. Observa-se que a soma dos tempos no pior caso dessas threads resulta em um tempo de processamento por período de 560 ms, logo o programa ficará ocioso em pelo menos 90ms por ciclo. Como os testes realizados não são em ambiente dedicado, pode haver pequenas alterações.

Figure 2. Pseudocódigo implementado.

```
Ciclo_menor = 650;
while(true){
    espera_proximo_ciclo_menor();
    atualiza_interrupcoes();
    atualiza_controle_vazao();
    atualiza_interrupcoes();
    atualiza_sensor_pressao();
    atualiza_interrupcoes();
    atualiza_sensor_vazao();
    atualiza_interrupcoes();

    espera_proximo_ciclo_menor();
    atualiza_interrupcoes();
    atualiza_controle_vazao();
    atualiza_interrupcoes();
    atualiza_sensor_temperatura();
    atualiza_interrupcoes();
    atualiza_sensor_altura();
    atualiza_interrupcoes();
}
```

Figure 3. Linha do tempo do pseudocódigo



2.2. Tecnologia empregada

Diversas tecnologias podem ser utilizadas para o desenvolvimento do algoritmo apresentado na figura 2. Atribui-se a sistemas de tempo real, tecnologias voltadas para um fator de performance. Tecnologias geralmente utilizadas são Assembly, C, C++, Ada e Java [1].

Neste trabalho, optou-se por desenvolver este algoritmo utilizando a linguagem C/C++. Pois é uma tecnologia que apresenta bons desempenhos para os processos deste segmento, além de otimizar o ambiente de desenvolvimento nas ferramentas já dispostas.

Foi utilizado a IDE Code Blocks versão 17.12, com as suas bibliotecas nativas. Código compilado com o GNU GCC compiler, com padrão c++11 ISO [2].

2.3. Detalhes de implementação

A estrutura desta gestão de controle foi implementada e colocada em funcionamento, referenciando as threads como funções externas que possam ser chamadas em um momento de real funcionamento do sistema.

Para sua simulação em tempo computacional, simulou-se um tempo variável para cada thread, com uma variação de até 25% do seu tempo de pior caso, simulando o funcionamento real deste software. Geram-se valores aleatórios para cada chamada das threads apresentadas na

tabela 1, e pausa-se o processamento do software neste tempo que seria atribuído ao processamento da thread. A tabela 2 apresenta o range de tempo atribuído a cada thread.

Table 2. Range do tempo de processamento simulado

Thread	Tempo mínimo	Tempo máximo
Sensor Altura	75	100
Sensor Pressão	75	100
Sensor Vazão	75	100
Sensor de Temperatura	75	100
Controlador de Vazão	225	300
Atualização Interrupções	15	20

Para testar a efetividade das interrupções neste software, é recebido em sua chamada dois parâmetros:

- Caractere que simboliza o tipo de teste a ser realizado, sendo 'T' para tempo final de processamento, e 'P' para simular o acionamento do botão de emergência.
- Número com decimal que associa o tempo de acionamento da interrupção.

3. Resultados encontrados

Para acompanhar os resultados encontrados durante o processamento do software, imprimiu-se as threads e o tempo aleatório de seus processamentos, de acordo com a tabela 2.

Observando ainda o tempo de pausa para o início do próximo ciclo podemos ver se a periodicidade das threads esta sendo respeitada. Um tempo de pausa maior que o esperado significa que alguma thread consumiu mais tempo que o esperado.

As saídas encontradas respeitam os limites de tempo de-limitados na tabela 1. Desta forma podemos afirmar que a estrutura desenvolvida respeita os requisitos temporais discutidos e exigidos para esse problema. A figura 4 apresenta o modelo de saída utilizado para um teste de tempo limite ('T') em apenas um segundo de funcionamento do software.

Apresentando o tempo para o processamento das threads sendo dividido de forma correta, foram testados diferentes tempos para a interrupção do processo. Estes valores são apresentados na figura 3:

Table 3. Tempo para resposta da interrupção

Tempo interrupção (seg)	Tempo resposta (seg)
1.0	1.076
5.3	5.569
10	10.198
50	50.221
100.5	101.764

Todos os testes foram realizados respeitando o limite para a resposta de uma interrupção de 300ms. Devemos salientar ainda que este tempo esta associado a uma

Figure 4. Teste de tempo para as threads

```
.\\modeloControlador.exe T 1.0  
TEMPO PASSADO: 0.010
```

```
Controle Vazao Tempo: 261  
Interrupcao tempo: 17  
Sensor Pressao Tempo: 90  
Interrupcao tempo: 17  
Sensor Vazao Tempo: 93  
Interrupcao tempo: 15  
Fim de ciclo: 102
```

```
TEMPO PASSADO: 0.669
```

```
Controle Vazao Tempo: 235  
Interrupcao tempo: 18  
Sensor Temperatura tempo: 94
```

```
TEMPO PASSADO: 1.060
```

```
Interrupcao devido a botao tempo.  
Chamada da funcao que realiza tratativa de finalizacao de processo.
```

chamada de interrupção que ocorre no início da execução da thread de Controle de Vazão. Para os demais casos com uma interrupção em 1 segundos, o tempo de resposta está associado ao tempo de processamento de um sensor.

Para os testes realizados foi indiferente o tipo de chamada, dado que as funções de interrupção não estão sendo implementadas em si, mas sendo analisado a disponibilização de tempo para as mesmas. De modo análogo para as funções de acesso aos sensores ou controlador dos parâmetros físicos do tanque.

4. Conclusão

Utilizando conhecimentos básicos do processo de controle e gestão de processo em um tanque de armazenamento de fluidos para processos industriais, estabeleceu-se um modelo de software para gestão de parâmetros. Levando em consideração aspectos de programação deste software e seus requisitos temporais.

Definiu-se baseado em conhecimentos preliminares deste modelo de processo os tipos de threads a serem processadas, em relação aos equipamentos de controle e sensoriamento do tanque. Estes equipamentos possuem threads associadas que processam sua atualização.

A gestão destas tarefas é realizada por meio de um algoritmo de um laço principal com tratadores de interrupção. Desenvolveu-se este algoritmo para os parâmetros temporais estabelecidos, assim como para as interrupções do processo como um todo. Os testes realizados mostram que o algoritmo implementado realiza estas tarefas respeitando o tempo dedicado para cada uma, assim como o tempo limite de resposta para as interrupções devidas.

References

- [1] D. M. Alberto. Linguagens de programação para sistemas de tempo real. *Universidade Federal De Ouro Preto*, 2017.
- [2] Code::Blocks. The free c/c++ and fortran ide., 2020.
- [3] R. S. de Oliveira. Fundamentos dos sistemas de tempo real, 2020. Implementação de tarefas em sistemas pequenos.
- [4] Fimaco. Tanques de armazenamento, 2018. Conheça as principais aplicações dos tanques industriais.