Colégio Técnico da UFMG

GABARITO DA PROVA ESCRITA

Concurso público de provas e títulos para provimento de Professor Efetivo da Carreira de Magistério do Ensino Básico, Técnico e Tecnológico, junto ao Núcleo de Tecnologia da Informação - Área de conhecimento: **Automação e Eletrônica**, do Colégio Técnico da Unidade Especial de Educação Básica e Profissional da Universidade Federal de Minas Gerais, conforme Edital nº 1221, de 25 de maio de 2026, Seção 3, páginas 56 a 62 e retificada pelo Edital nº 1.293, de 2 de junho de 2026, publicado no Diário Oficial da União (DOU) em 08 de junho de 2026, Seção 3, página 71.

Data: 19/10/2026

Horário: 14 horas

Local: Sala 201 do Colégio Técnico da UFMG

Orientações para a prova escrita

- Quando da realização das provas do concurso, o candidato deverá, obrigatoriamente, apresentar documento oficial e original de identidade com fotografia e assinatura. Serão aceitos documentos digitais com foto e assinatura (CNH digital e RG digital ou qualquer outro documento digital, com foto e assinatura, válido como documento de identificação, nos termos da legislação vigente), desde que apresentados nos respectivos aplicativos oficiais.
- Será automaticamente eliminado o candidato que não apresentar documento oficial e original de identidade com fotografia e assinatura quando da apuração da presença dos candidatos, em cada etapa ou prova.
- A prova escrita terá duração total de 5 (cinco) horas, sendo permitida a consulta a materiais impressos ou manuscritos durante a primeira hora. Nesse período, o candidato poderá fazer anotações nas folhas de rascunho. Encerrada a primeira hora, todos os materiais de consulta deverão ser guardados, sendo permitido ao candidato prosseguir com a prova pelas 4 (quatro) horas restantes utilizando apenas o rascunho previamente anotado.
- Não será permitida a consulta a arquivos em formato eletrônico, de qualquer natureza. Da mesma forma, celulares, relógios e quaisquer outros dispositivos eletrônicos não poderão ser utilizados durante a prova, devendo permanecer desligados e guardados.
- Para a realização da prova escrita, o candidato receberá um Caderno de Respostas no formato livreto, folhas para rascunho em papel almaço, carimbadas e rubricadas, e as questões de prova.

- O Caderno de Respostas deverá ser preenchido com caneta esferográfica. Os rascunhos poderão ser feitos a lápis ou a caneta, a critério do candidato.
- A Prova Escrita deve ser anônima, identificada por código aleatório, fornecido pelo departamento/estrutura equivalente, para efeito de avaliação da Comissão Examinadora. Os códigos serão distribuídos aleatoriamente pelo departamento/estrutura equivalente aos candidatos antes do início da Prova Escrita. Os códigos aleatórios deverão ser lacrados em envelope, que somente será aberto pela Comissão Examinadora após os trabalhos de correção.
- Conforme o item 8.12 do edital, "será automaticamente eliminado do Concurso o candidato que durante a realização das provas for flagrado em tentativa de fraude ou em desrespeito aos membros da Comissão Examinadora, bem como à Comunidade Universitária".

QUESTÃO 01

Um laboratório precisa monitorar a temperatura de um ambiente utilizando um computador. Para isso, será empregado um sensor analógico conectado à entrada de um conversor analógico-digital (A/D) integrado a um microcontrolador. O microcontrolador troca mensagens com o computador por uma interface UART e uma ponte USB–UART, que aparece no sistema operacional como uma porta serial.

O programa de aquisição deve ser desenvolvido em linguagem `C`, para execução no computador, e deve solicitar uma medida de temperatura a cada 30 segundos. Considere o computador executando Linux e o dispositivo acessível pela porta `/dev/ttyUSB0`.

O sistema possui as seguintes características:

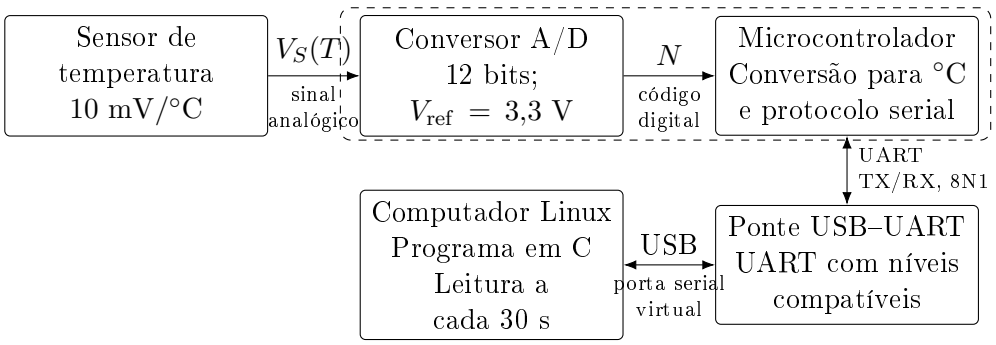
1. Sensor ideal, faixa de 0 a 100 °C e sensibilidade de 10 mV/°C, sem deslocamento de tensão;
2. Conversor A/D de 12 bits, entrada unipolar de 0 a 3,3 V e referência $V_{\text{ref}} = 3{,}3$ V.
3. Comunicação serial: 9600 bit/s, 8 bits de dados, sem paridade e 1 bit de parada (8N1), sem controle de fluxo.
4. Sensor, conversor e UART utilizam níveis elétricos compatíveis; considere conexão ideal, sem necessidade de amplificação, filtragem ou isolamento nesta questão.

Considerando esse sistema, pede-se:

a) **Desenhe o diagrama de blocos, indicando claramente o tipo de sinal em cada estágio (analógico/digital), os níveis/interfaces de comunicação física utilizadas entre os blocos e o identificador do dispositivo no sistema operacional.**

GABARITO: O diagrama deve conter, pelo menos, 4 blocos principais em sequência funcional. São obrigatórios os seguintes dados:

- Identificação do sinal do sensor como analógico $V_S(T)$;
- Agrupamento (ou indicação) do A/D como periférico interno do microcontrolador;
- Identificação da comunicação bidirecional UART (TX/RX) entre o microcontrolador e a ponte;
- Indicação da interface USB/porta serial virtual conectada ao computador Linux.



b) **Caracterize a conversão A/D, a interface UART e a ponte USB–UART considerando o tipo de grandeza física envolvida, o tipo de dado trafegado e a disponibilidade no contexto do sistema.**

GABARITO:

- **Conversor A/D:** Opera com tensão elétrica analógica contínua do sensor (0 a 1,0 V para a faixa de 0 a 100 °C), gerando como dado um código digital discreto de 12 bits (0 a 4095) proporcional à temperatura. Sua disponibilidade é sob demanda, operando pontualmente apenas quando acionado pelo microcontrolador.
- **Interface UART:** Opera com níveis lógicos discretos padrão TTL/CMOS (0 V e 3,3 V) em linhas dedicadas (TX/RX), transmitindo dados na forma de caracteres ASCII serializados em quadros assíncronos 8N1. Sua disponibilidade é assíncrona e eventual, ativando-se apenas durante o envio de comandos e respostas.
- **Ponte USB–UART:** Opera com sinais de tensão diferencial de alta velocidade no barramento USB, trafegando pacotes de dados do protocolo USB que encapsulam a comunicação serial. Sua disponibilidade é contínua sob o sistema operacional, mantendo o arquivo de dispositivo (`/dev/ttyUSB0`) permanentemente acessível para o programa em `C`.

---

c) **Determine a expressão que relaciona o código N à temperatura T, utilizando a convenção fornecida. Calcule a variação de temperatura correspondente a uma unidade de código, a temperatura estimada para $N = 1536$ e avalie o resultado.**

GABARITO: Pela relação do sensor (10 mV/°C), a tensão de saída é dada por:

$$V_S = 0{,}010 \cdot T \implies T = \frac{V_S}{0{,}010}.$$

Considerando a quantização do conversor A/D de 12 bits ($2^{12} = 4096$ níveis) com $V_{\text{ref}} = 3{,}3$ V:

$$V_S = N \cdot \frac{3{,}3}{4096} \text{ V}$$

Substituindo V_S na equação da temperatura:

$$T(N) = \frac{N \cdot 3{,}3}{4096 \times 0{,}010} \qquad T(N) \approx 0{,}081\, N \quad (^\circ\text{C})$$

Para $N = 1536$:

$$T = 0{,}081 \times 1536 \approx 123{,}75\,^\circ\text{C}$$

O valor calculado ($T \approx 123{,}75\,^\circ$C) é fisicamente incompatível com o sistema. O sensor opera na faixa de 0 a 100 °C, gerando uma tensão V_S de 0 a 1,0 V, o que corresponde a um código máximo de $N \approx 1241$.

A leitura de $N = 1536$ ($V_S = 1{,}2375$ V) excede o limite físico do sensor e indica uma condição de falha no sistema, como sobreaquecimento crítico do ambiente, curto-circuito da linha de sinal com uma tensão superior, ou falha no próprio sensor/conversor A/D.

---

d) **Devido a uma atualização no sistema, o *firmware* do microcontrolador passou a responder às solicitações enviando as medições em um *frame* fixo de 8 bytes:**

`[STX] [END] [T1] [T2] [T3] [T4] [CHK] [LF]`

- `STX` (`0x02`): Delimitador de início;
- `END` (`0x01`): Endereço do dispositivo;
- `T1..T4`: 4 caracteres ASCII com a temperatura em décimos de °C;
- `CHK`: Byte de verificação (*Checksum*);
- `LF` (`0x0A`): Delimitador de fim.

Considere que a comunicação serial e o tempo limite (*timeout*) já são gerenciados pela função auxiliar `int leitura_serial(unsigned char *buffer);` que armazena os bytes recebidos da porta `/dev/ttyUSB0` no vetor `buffer` e retorna a quantidade total de bytes lidos (ou `0` em caso de *timeout*).

Escreva um programa em `C` que:

- **Implemente a função**

  `int processa_frame(const unsigned char *buffer, float *temp_out)`

  que valida o *frame* proposto, sendo `0` o *frame* inválido e `1`, *frame* válido;
- **Implemente o laço principal (`main`) que solicite a leitura, chame a função `leitura_serial()`, valide o *buffer* resultante e exiba o valor da temperatura em tela caso o pacote seja válido, ou uma mensagem de erro em caso de falha.**

GABARITO: São considerados ps principais pontos no código:

- Validação dos delimitadores de início, endereço e fim nos índices corretos do *buffer*;
- Cálculo e validação do *checksum* comparado ao byte recebido;
- Conversão da carga útil em ASCII para valor numérico (`float`) em °C a partir dos décimos de grau;
- Retorno adequado de indicador de falha para quadros corrompidos ou inválidos;
- Chamada correta da função de leitura serial, tratamento do tempo limite (*timeout*) e temporização do laço principal.

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

// a) Funcao de validacao e conversao do frame de 8 bytes
int processa_frame(const unsigned char *buffer, float *temp_out) {
  // 1. Validacao dos delimitadores de enquadramento (STX, END, LF)
  if (buffer[0] != 0x02 || buffer[1] != 0x01 || buffer[7] != 0x0A) {
    return 0;
  }

  // 2. Verificacao do Checksum: CHK = (END + T1 + T2 + T3 + T4) % 256
  unsigned char chk_calculado = (buffer[1] + buffer[2] + buffer[3] + buffer[4] + buffer[5]) % 256;
```

```
14        if (chk_calculado != buffer[6]) {
15          return 0;
16        }
17
18        // 3. Validacao dos caracteres ASCII numericos ('0' a '9')
              nos dados T1..T4
19        for (int i = 2; i <= 5; i++) {
20          if (buffer[i] < '0' || buffer[i] > '9') {
21            return 0;
22          }
23        }
24
25        // 4. Conversao do payload de decimos de C para float
26        int decimos = (buffer[2] - '0') * 1000 +
27        (buffer[3] - '0') * 100 +
28        (buffer[4] - '0') * 10 +
29        (buffer[5] - '0');
30
31        *temp_out = decimos / 10.0f;
32        return 1;
33      }
34
35      // b) Laco principal de aquisicao
36      int main(void) {
37        unsigned char rx_buffer[8];
38        float temperatura;
39
40        while (1) {
41          solicita_medida();
42
43          int n_bytes = leitura_serial(rx_buffer);
44
45          if (n_bytes == 8) {
46            if (processa_frame(rx_buffer, &temperatura)) {
47              printf("Temperatura medida: %.1f C\n", temperatura);
48            } else {
49              printf("[ERRO] Frame invalido ou falha de Checksum.\n
                    ");
50            }
51          } else if (n_bytes == 0) {
52            printf("[ERRO] Tempo limite esgotado (Timeout).\n");
53          } else {
54            printf("[ERRO] Pacote incompleto (%d/8 bytes recebidos)
                  .\n", n_bytes);
55          }
56
57          sleep(30);
58        }
59
60        return 0;
61      }
```

e) **Explique por que executar uma espera fixada de 30 segundos (como `sleep(30)`) logo após receber a resposta pode produzir um intervalo entre solicitações maior que 30 segundos. Explique uma estratégia lógica para programar os instantes de aquisição de forma periódica.**

GABARITO: Se o programa aguardar a resposta e, posteriormente, executar uma espera fixada de 30 segundos, o intervalo efetivo entre solicitações consecutivas será aproximadamente:

$$\text{intervalo} = 30\ \text{s} + \text{tempo da transação e do processamento.}$$

Por exemplo, uma transação serial com tempo de resposta e processamento de 0,2 s resultará em solicitações espaçadas por cerca de 30,2 s. Esse atraso se acumula a cada ciclo, gerando um desvio temporal acumulado (*drift*) ao longo das aquisições.

A solução consiste em programar instantes absolutos baseados em um relógio monotônico (`CLOCK_MONOTONIC`):

$$t_k = t_0 + 30k, \qquad k = 0, 1, 2, \ldots$$

O programa aguarda cada instante t_k utilizando a função `clock_nanosleep()` configurada com a flag `TIMER_ABSTIME`. Ao finalizar uma leitura, o próximo instante alvo é calculado somando-se 30 segundos ao instante anteriormente previsto ($t_{k+1} = t_k + 30$ s), e não ao horário em que o processamento da leitura terminou.

Caso o sistema fique suspenso ou ocorra um atraso excepcional que exceda a janela da amostragem, os instantes já vencidos devem ser saltados para evitar uma sequência contínua e indesejada de solicitações em rajada (*burst*) na tentativa de compensar medições passadas.

Como o Linux convencional não é um sistema operacional de tempo real rígido (*hard real-time*), a execução do processo pode sofrer pequenos atrasos decorrentes do escalonamento do SO. A estratégia de tempo absoluto elimina o acúmulo da latência de leitura, porém não garante determinismo temporal estrito.

PONTOS CHAVE:

- Mapear as etapas de conversão e transmissão do sinal, desde a transdução da grandeza física até a disponibilização do arquivo de dispositivo no sistema operacional.
- Identificar a natureza física do sinal, o formato do dado trafegado e a disponibilidade de operação do A/D, da interface UART e da ponte USB–UART.
- Obter a expressão de conversão entre o código digital e a temperatura, determinar a resolução por LSB e avaliar o significado físico de leituras fora da faixa nominal de operação do sensor.
- Validar os delimitadores de enquadramento, verificar o *Checksum*, converter a carga útil em ASCII para valor numérico (`float`) e tratar erros de resposta e *timeout*.
- Explicar o acúmulo de atraso (*drift*) decorrente de esperas relativas e propor a programação por instantes absolutos no relógio monotônico para manter a periodicidade.

QUESTÃO 02

Uma indústria deseja automatizar um processo de preparação de um líquido em um tanque. O sistema deve realizar automaticamente as etapas de enchimento, aquecimento com mistura e esvaziamento, utilizando um Controlador Lógico Programável (CLP).

O tanque possui os seguintes dispositivos:

- **Chave de nível mínimo:** Indica quando o líquido atinge ou ultrapassa o nível mínimo. Seu sinal lógico é 1 quando o líquido está no nível mínimo ou acima dele, e 0 quando está abaixo desse nível.
- **Chave de nível máximo:** Indica quando o líquido atinge ou ultrapassa o nível máximo. Seu sinal lógico é 1 quando o nível máximo é atingido, e 0 quando não é atingido.
- **Transmissor de temperatura:** Fornece um sinal de 4–20 mA correspondente, linearmente, à faixa de $-100°C$ a $200°C$.
- **Bomba elétrica de enchimento:** Responsável por abastecer o tanque, acionada por contator.
- **Válvula solenoide de esvaziamento:** Responsável por liberar o líquido do tanque quando energizada.
- **Misturador elétrico:** Responsável por homogeneizar o líquido durante o aquecimento, acionado por contator.
- **Resistência elétrica:** Responsável por aquecer o líquido, acionada por contator.
- **Botão de partida:** Inicia um ciclo de operação. Seu sinal lógico é 1 quando pressionado, e 0 quando não pressionado.
- **Botão de emergência:** Solicita a interrupção imediata do processo. Seu sinal lógico é 0 quando pressionado, e 1 quando não pressionado.

O funcionamento desejado é o seguinte:

- O processo permanece em repouso até que o operador acione o botão de partida. Um novo ciclo somente poderá começar se não houver emergência acionada e se o tanque não estiver no nível máximo.
- Durante o enchimento, a bomba permanece ligada até que o sinal de nível máximo seja ativado. Nesse momento, a bomba é desligada e tem início a etapa de aquecimento com mistura.
- Durante o aquecimento, a resistência e o misturador permanecem ligados. Quando a temperatura medida atingir ou ultrapassar $80°C$, ambos serão desligados e terá início o esvaziamento.
- Durante o esvaziamento, a válvula solenoide permanece acionada até que o sinal da chave de nível mínimo passe para 0, indicando que o líquido está abaixo desse nível. Em seguida, a válvula é desligada e o ciclo é considerado concluído.
- Após a conclusão de um ciclo, o sistema retorna ao estado de repouso e aguarda um novo acionamento do botão de partida.

- Se a emergência for acionada em qualquer etapa, todos os comandos de acionamento da bomba, da válvula, da resistência e do misturador deverão ser desativados. Após a liberação da emergência, o processo deverá permanecer em repouso, exigindo um novo acionamento do botão de partida para retomar o ciclo na etapa na qual foi interrompido (não será permitida a retomada automática).

- Considere que o CLP possui entradas e saídas digitais e uma entrada analógica de 4–20 mA com conversor A/D de 10 bits. Para a conversão analógica, considere um conversor ideal com 1024 códigos digitais e adote o tamanho nominal do passo de quantização como a faixa de entrada dividida por 1024.

- O sistema deverá contabilizar a quantidade de ciclos concluídos, registrar o tempo de enchimento do ciclo atual e disponibilizar a temperatura, o nível do tanque e o estado dos equipamentos para monitoramento no sistema de supervisão. Não é necessário implementar diagnóstico de falhas dos sensores ou atuadores.

Com base nesse cenário, responda aos itens a seguir.

a) **Determine o span do transmissor de temperatura e a resolução nominal da medição de temperatura associada ao conversor A/D de 10 bits. Apresente os cálculos e expresse os resultados nas unidades adequadas.**

GABARITO: O *Span* corresponde à amplitude total da faixa de medição (diferença entre o valor máximo T_{max} e o valor mínimo T_{min}):

$$\text{Span} = T_{max} - T_{min} = 200°\text{C} - (-100°\text{C}) = 300°\text{C}$$

A resolução nominal é dada pela variação de temperatura correspondente a 1 LSB do conversor A/D de 10 bits ($2^{10} = 1024$ códigos/passos):

$$\Delta T = \frac{\text{Span}}{2^N} = \frac{300°\text{C}}{1024} \approx 0{,}293°\text{C}$$

---

b) **A edição clássica da norma IEC 61131-3 estabelece cinco linguagens de programação utilizadas no desenvolvimento de programas para Controladores Lógicos Programáveis (CLPs). Considerando o sistema automatizado de enchimento, aquecimento, mistura e esvaziamento de um tanque descrito anteriormente, cite essas cinco linguagens, identificando quais são gráficas e quais são textuais.**

GABARITO: Conforme a norma IEC 61131-3, as cinco linguagens de programação padronizadas dividem-se em duas categorias:

- **Linguagens Gráficas:**
  - **Diagrama Ladder (LD - *Ladder Diagram*):** Linguagem baseada em diagramas de contatos elétricos/relés.
  - **Diagrama de Blocos Funcionais (FBD - *Function Block Diagram*):** Linguagem baseada na conexão de blocos lógicos funcionais.
  - **Diagrama de Funções Seqüenciais (SFC - *Sequential Function Chart*):** Linguagem estruturada em etapas, transições e ações, ideal para processos seqüenciais como o controle do tanque.

- **Linguagens Textuais:**
  - **Lista de Instruções (IL - *Instruction List*):** Linguagem de baixo nível baseada em código de montagem (*assembly*).
  - **Texto Estruturado (ST - *Structured Text*):** Linguagem de alto nível com sintaxe semelhante ao Pascal e C, apropriada para cálculos matemáticos complexos.

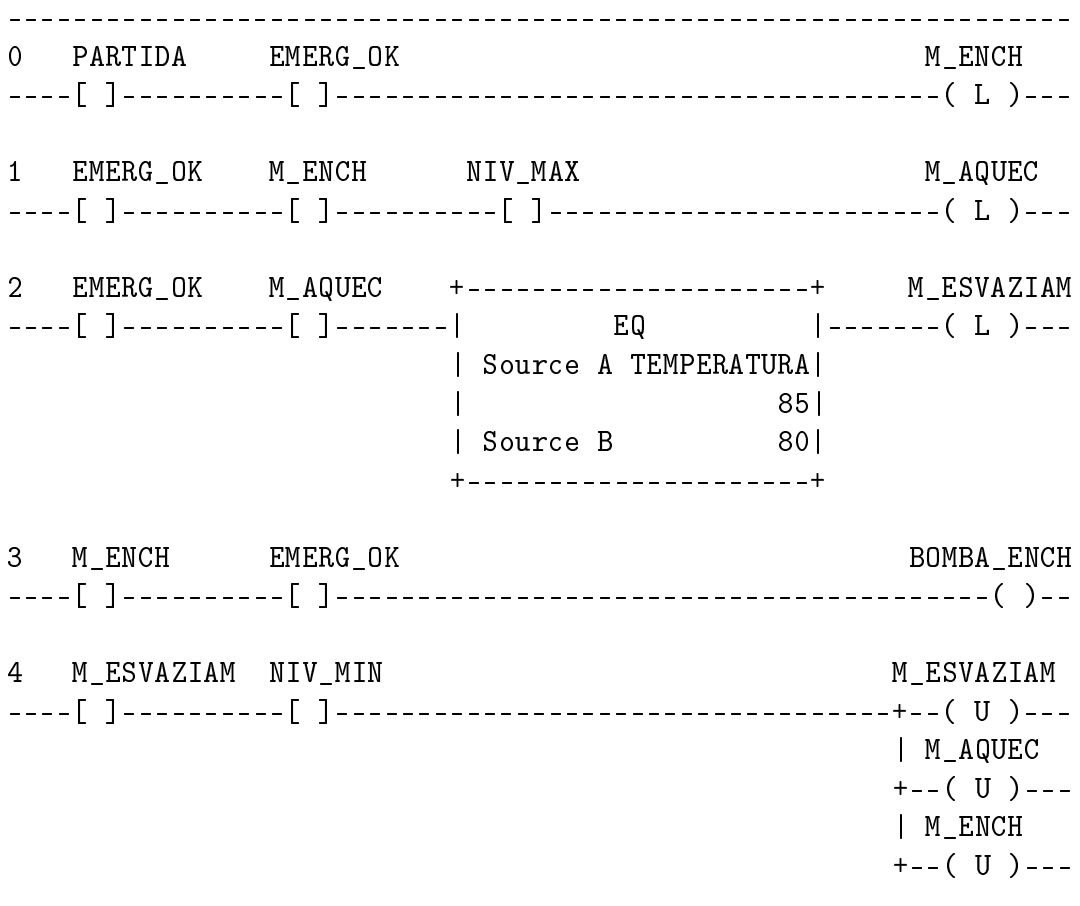
---

c) **Considerando o modelo hierárquico tradicional da automação industrial (Pirâmide da Automação), identifique os níveis em que normalmente se encontram os sensores e atuadores, o Controlador Lógico Programável (CLP) e o Sistema SCADA (*Supervisory Control and Data Acquisition*)**

GABARITO: Conforme o modelo hierárquico tradicional da automação industrial (Pirâmide da Automação):

- **Nível de Campo (Processo):** Sensores de nível, transmissor de temperatura e atuadores. Responsáveis pela aquisição de sinais e atuação direta no processo.
- **Nível de Controle:** CLP. Responsável pelo processamento da lógica sequencial e pelos comandos locais em tempo real.
- **Nível de Supervisão:** Sistema SCADA. Responsável pela visualização do processo, gestão de alarmes, registros de tendências e interface/interação com o operador.

Durante a análise de um programa Ladder desenvolvido para o controle do tanque descrito no enunciado, um técnico identificou os trechos de lógica apresentados a seguir.



Considere que os demais trechos do programa, incluindo os comandos de saída e o tratamento da emergência, estão implementados corretamente. A descrição das memórias é dada a seguir:

- `M_ENCH`: indica que a etapa de enchimento está ativa;
- `M_AQUEC`: indica que a etapa de aquecimento com mistura está ativa;
- `M_ESVAZIAM`: indica que a etapa de esvaziamento está ativa;
- `NIV_MAX`: assume valor lógico 1 quando o nível máximo é atingido;
- `NIV_MIN`: assume valor lógico 1 quando o nível mínimo é atingido;
- `EMERG_OK`: assume valor lógico 1 quando a emergência não está acionada;
- `PARTIDA`: assume valor lógico 1 quando o botão de partida é pressionado;
- `TEMPERATURA`: é uma variável inteira que armazena o valor da temperatura na faixa de -100°C a 200°C.

Ao executar o programa, verificou-se que:

- Após o tanque atingir o nível máximo, a bomba continua acionada durante a etapa de aquecimento.
- A etapa de esvaziamento nem sempre é ativada. Registrou-se isso, por exemplo, para a sequência de leituras do sensor durante o aquecimento dada por (em graus Celsius): $\{25, 26, 27, 29, 33, 38, 45, 55, 67, 81, 98\}$.
- O ciclo é encerrado antes da etapa de esvaziamento ser concluída.

Com base nessas informações complementares:

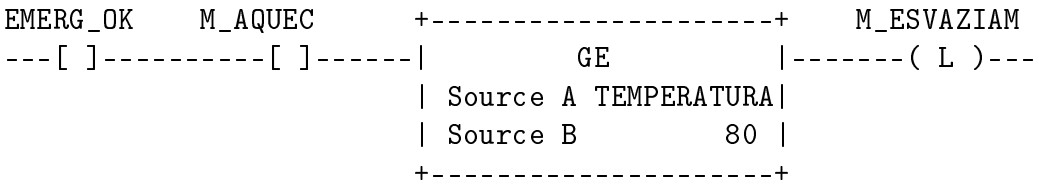
d) **Identifique o erro na lógica de sequenciamento da bomba que a mantém acionada durante a etapa de aquecimento. Apresente a correção necessária na Rede 3, sem alterar nenhuma outra rede.**

GABARITO: Falta desativar, por meio de RESET, a memória `M_ENCH` quando ocorre a transição do enchimento para o aquecimento na Rede 1. Como a instrução dada exige alterar exclusivamente a Rede 3, a correção consiste em acrescentar um contato normalmente fechado (NF) com a memória `M_AQUEC` em série na linha de comando da bomba (Rede 3).

---

e) **Identifique o erro na lógica de esvaziamento, que fez com que essa etapa nem sempre seja ativada. Apresente a correção alterando apenas uma das redes do código.**

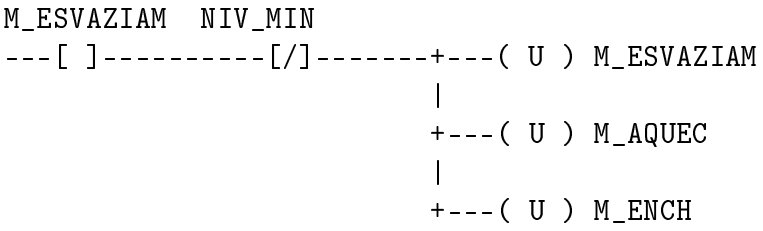
GABARITO: O bloco utilizado para comparação da temperatura com o valor desejado de 80°C é um bloco de igualdade (`EQ`), o que faz com que essa rede seja ativada apenas se a temperatura for exatamente igual a esse valor, e não maior. Na sequência mostrada, o valor 80 não aparece, e por isso a etapa de esvaziamento não é ativada. A ação necessária na Rede 2 é alterar o bloco `EQ` (igual a) para `GE` (*Greater Than or Equal* / maior ou igual).



---

f) **Identifique o erro na lógica de encerramento de ciclo, que encerra o ciclo antes do esvaziamento estar concluído. Apresente a correção alterando apenas uma das redes do código.**

GABARITO: A condição utilizada para finalizar o esvaziamento está invertida, pois quando o nível estiver abaixo do sensor de nível mínimo, esse sensor estará enviando o nível lógico 0. Assim, a ação necessária na Rede 4 para correção é alterar o tipo do contato da variável `NIV_MIN` de contato aberto (NA) para contato fechado (NF).



PONTOS CHAVE:

- Conhecimento conceitual e cálculo de parâmetros de desempenho do sistema de medição, tais como a determinação do *span* do sensor e a resolução nominal da conversão A/D;
- Domínio da norma IEC 61131-3, compreendendo a classificação e a distinção entre as linguagens de programação gráficas e textuais utilizadas em CLPs;
- Identificação dos níveis da Pirâmide da Automação (modelo ISO 22400 / ISA-95) e o correto mapeamento funcional de dispositivos de campo, controle e supervisão;
- Análise crítica de diagramas Ladder para identificação de erros de lógica sequencial, problemas de sincronismo por amostragem discreta (*scan*), sensoriamento com lógica invertida e proposição de correções restritas por intertravamento.

QUESTÃO 03

Considerando o diagrama de blocos ilustrado na Fig. 1, em que o sinal é dado por:

$$x(t) = \frac{220}{\sqrt{3}} + 180\sin(\omega t), \quad \omega = 377 \text{ rad/s}$$

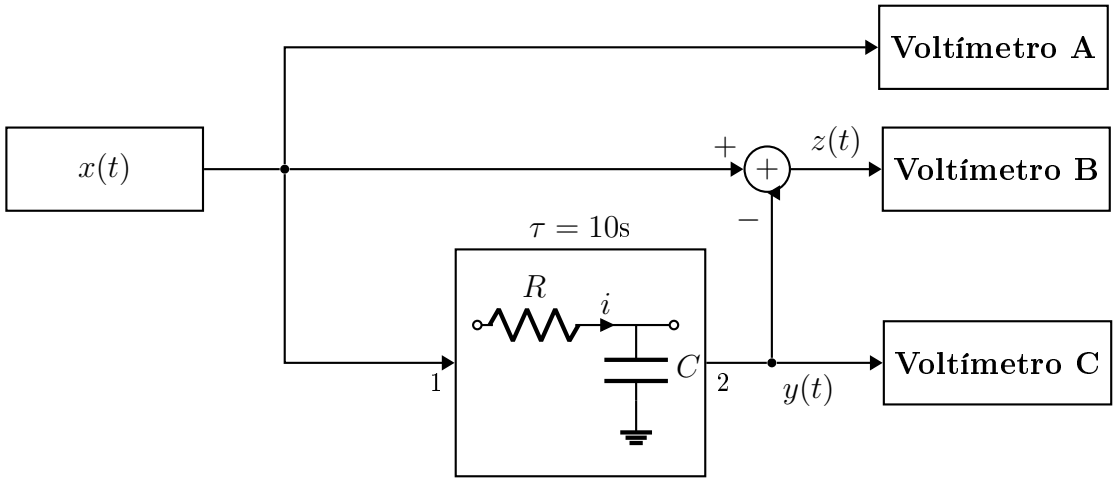


Figure 1: Diagrama de blocos com circuito de medição RC e voltímetros alinhados.

pede-se:

a) **Demonstre a função de transferência $\frac{Z(s)}{X(s)}$ (Domínio de Laplace) usando o conceito de impedância, escrevendo equações com a notação de nós como ilustrada.**

GABARITO: Representando os componentes do filtro RC no domínio de Laplace por meio de suas impedâncias operacionais:

$$Z_R(s) = R \quad \text{e} \quad Z_C(s) = \frac{1}{sC}$$

Aplicando a Lei das Correntes de Kirchhoff (LCK) no nó de saída **2**, cuja tensão operacional é $Y(s)$, considerando a tensão de entrada no nó **1** igual a $X(s)$:

$$\frac{X(s) - Y(s)}{Z_R(s)} = \frac{Y(s)}{Z_C(s)} \implies \frac{X(s) - Y(s)}{R} = sCY(s)$$

Isolando a relação $Y(s)/X(s)$ com a constante de tempo $\tau = RC = 10$ s:

$$X(s) = Y(s)(1 + sRC) \implies Y(s) = \frac{X(s)}{1 + \tau s} = \frac{X(s)}{10s + 1}$$

A partir do nó somador, a saída $Z(s)$ é definida pela diferença entre o sinal de entrada $X(s)$ e a saída do filtro $Y(s)$:

$$Z(s) = X(s) - Y(s)$$

Substituindo a expressão de $Y(s)$ em função de $X(s)$:

$$Z(s) = X(s) - \frac{X(s)}{10s+1} = X(s)\left(1 - \frac{1}{10s+1}\right) = X(s)\left(\frac{(10s+1)-1}{10s+1}\right)$$

Portanto, a função de transferência $H_{ZX}(s) = \dfrac{Z(s)}{X(s)}$ do filtro passa-baixas equivalente resulta em:

$$\frac{Z(s)}{X(s)} = \frac{10s}{10s+1}$$

---

b) **Calcule e justifique o valor indicado pelo Voltímetro C TRUE RMS.**

GABARITO: O Voltímetro C mede o sinal $y(t)$, que é a saída do filtro passa-baixas RC com constante de tempo $\tau = 10$ s, o que corresponde a uma frequência de corte $\omega_c = \frac{1}{\tau} = 0{,}1$ rad/s ($f_c \approx 15{,}9$ mHz).

Como a frequência angular da componente alternada do sinal de entrada é $\omega = 377$ rad/s ($f = \frac{377}{2\pi} = 60$ Hz), a frequência de operação está localizada mais de 3,5 décadas acima da frequência de corte do filtro ($\omega \gg \omega_c$).

Nessa região, o filtro RC de primeira ordem atua com um decaimento de -20 dB/decada, proporcionando uma atenuação.

Essa atenuação superior a 3700 vezes reduz a amplitude da componente senoidal de 180 V para cerca de 0,048 V na saída ($V_{y,\mathrm{AC,RMS}} \approx 0{,}034$ V).

Com a componente alternada praticamente eliminada, a contribuição de sua energia na medição True RMS torna-se desprezível. Portanto, o multímetro C indicará apenas o valor médio, ou seja, a componente contínua (DC) do sinal:

$$V_{C(\mathrm{RMS})} \approx V_{\mathrm{DC}} = \frac{220}{\sqrt{3}} \approx 127 \text{ V}$$

---

c) **Calcule e justifique o valor indicado pelo Voltímetro B TRUE RMS.**

GABARITO: O Voltímetro B mede o sinal $z(t) = x(t) - y(t)$, o que equivale a um filtro Passa-Altas com constante de tempo $\tau = 10$ s e frequência de corte $\omega_c = \frac{1}{\tau} = 0{,}1$ rad/s ($f_c \approx 15{,}9$ mHz).

Como o filtro passa-baixas apresenta ganho unitário em DC ($y_{\mathrm{DC}} = x_{\mathrm{DC}}$), a componente contínua é totalmente cancelada na subtração do somador:

$$V_{z,\mathrm{DC}} = x_{\mathrm{DC}} - y_{\mathrm{DC}} = \frac{220}{\sqrt{3}} - \frac{220}{\sqrt{3}} = 0 \text{ V}$$

Para a frequência $\omega = 377$ rad/s ($f = \frac{377}{2\pi} = 60$ Hz), que está mais de 3,5 décadas acima da frequência de corte ($\omega \gg \omega_c$), a saída do filtro passa-baixas é atenuada em mais de 70 dB (> 3700 vezes), resultando em $y_{\mathrm{AC}}(t) \approx 0$ V.

Dessa forma, a componente alternada passa pelo somador sem sofrer atenuação:

$$z(t) \approx x_{\mathrm{AC}}(t) = 180\sin(377t) \text{ V}$$

O valor eficaz indicado pelo instrumento True RMS para a senóide pura de amplitude $V_p = 180$ V é:

$$V_{B(\mathrm{RMS})} = \frac{V_p}{\sqrt{2}} = \frac{180}{\sqrt{2}} \approx 127 \text{ V}$$

---

d) **Calcule e justifique o valor indicado pelo Voltímetro A TRUE RMS.**

GABARITO: O Voltímetro A mede o sinal de entrada $x(t) = \frac{220}{\sqrt{3}} + 180\sin(377t)$ V, que é um sinal composto por uma componente contínua (DC) e uma componente alternada (AC) senoidal.

Por definição, o valor eficaz (RMS) de um sinal composto por componentes ortogonais em frequência é a raiz quadrada da soma do quadrado da parcela contínua com o quadrado do valor eficaz da parcela alternada:

$$V_{A(\text{RMS})} = \sqrt{V_{x,\text{DC}}^2 + V_{x,\text{AC,RMS}}^2}$$

Substituindo $V_{x,\text{DC}} = \frac{220}{\sqrt{3}}$ V e $V_{x,\text{AC,RMS}} = \frac{180}{\sqrt{2}}$ V:

$$V_{A(\text{RMS})} = \sqrt{\left(\frac{220}{\sqrt{3}}\right)^2 + \left(\frac{180}{\sqrt{2}}\right)^2} = \sqrt{\frac{48400}{3} + \frac{32400}{2}} = \sqrt{16133{,}33 + 16200} \approx 179{,}81 \text{ V}$$

Como demonstrado nos itens anteriores, $V_{C(\text{RMS})} \approx 127$ V e $V_{B(\text{RMS})} \approx 127$ V:

$$V_{A(\text{RMS})} = \sqrt{V_{C(\text{RMS})}^2 + V_{B(\text{RMS})}^2} = \sqrt{127^2 + 127^2} = 127\sqrt{2} \approx 179{,}6 \text{ V}$$

PONTOS CHAVE:

- O circuito de medição atua separando o sinal composto $x(t)$ em suas componentes fundamentais. O sinal de entrada possui 2 componentes distintas sendo uma de conrrente contínua e outra de corrente alternada.
- O Voltímetro A não possui isolamento, considerando todo o sinal original $x(t)$.
- O Voltímetro B isola a componente alternada ($V_{\text{AC,RMS}}$) via bloco passa-altas complementar.
- O Voltímetro C isola a componente contínua (V_{DC}) via filtro passa-baixas;
- Voltímetros True RMS medem o valor eficaz total do sinal processando a energia conjunta das parcelas ortogonais no domínio da frequência, o que obedece à relação quadrática:

$$V_{\text{Total, RMS}} = \sqrt{V_{\text{DC}}^2 + V_{\text{AC,RMS}}^2}$$

- A escolha das amplitudes do sinal de entrada resulta em parcelas com valores eficazes idênticos e, principalmente, baseado na filtragem dos sinais.

QUESTÃO 04

Um sistema de monitoramento industrial registra eventos produzidos por equipamentos. Cada evento contém um identificador inteiro de equipamento, um instante de ocorrência em milissegundos (inteiro sem sinal) e um código: 0 (informação), 1 (advertência) ou 2 (falha). Os eventos ficam armazenados em um vetor, em ordem de chegada. Eventos com código 2 devem ser contabilizados por equipamento.

Considere a definição, com n elementos válidos e $0 \leq n \leq$ capacidade:

```
typedef struct {
  int equipamento;
  unsigned int instante_ms;
  int codigo;
} Evento;
```

a) **Defina uma função que verifica se um registro é válido, considerando identificador de equipamento maior que zero e código entre 0 e 2. Explique o valor retornado.**

GABARITO: Um evento é válido se `equipamento > 0` e `codigo >= 0 && codigo <= 2`. A função pode retornar 1 para válido e 0 para inválido.

---

b) **Implemente, em C, `int contar_falhas(const Evento v[], int n, int id)`, que retorna a quantidade de eventos válidos de falha do equipamento informado. Considere $0 \leq n$ e vetor válido.**

GABARITO: Uma solução:

```
int valido(Evento e) {
  return e.equipamento > 0 &&
  e.codigo >= 0 && e.codigo <= 2;
}
int contar_falhas(const Evento v[], int n, int id) {
  int total = 0;
  for (int i = 0; i < n; i++)
  if (valido(v[i]) && v[i].equipamento == id &&
  v[i].codigo == 2)
  total++;
  return total;
}
```

---

c) **Implemente `int primeiro_instante_falha(const Evento v[], int n, int id, unsigned int *t)`. Retorne 1 e armazene em `*t` o menor instante de falha do equipamento, ou retorne 0 se não houver falha. Considere que `t` é não nulo.**

GABARITO: Percorrer todo o vetor, pois a ordem de chegada não implica ordem crescente do instante. Uma solução:

```
int primeiro_instante_falha(const Evento v[], int n, int id,
unsigned int *t) {
  int achou = 0;
  unsigned int menor = 0;
  for (int i = 0; i < n; i++) {
    if (valido(v[i]) && v[i].equipamento == id &&
```

```
        v[i].codigo == 2) {
          if (!achou || v[i].instante_ms < menor) {
            menor = v[i].instante_ms;
            achou = 1;
          }
        }
      }
      if (achou) *t = menor;
      return achou;
    }
```

---

d) **Para os registros (equipamento, instante_ms, código): (3,100,1), (2,130,2), (2,90,2), (2,190,0), (3,210,2), (2,140,3), determine o resultado das duas funções para `id=2` e justifique.**

GABARITO: Para `id=2`, há duas falhas válidas, nos instantes 130 e 90 ms. `contar_falhas` retorna 2; `primeiro_instante_falha` retorna 1 e define `*t=90`. O registro (2,140,3) é inválido. O registro com código 0 não é falha.

PONTOS CHAVE:

- Cada elemento do vetor é um registro composto por três campos essenciais: i) identificador do equipamento (`int`), ii)tempo de ocorrência em milissegundos (`unsigned int`) e o nível de severidade (`int` com valores restritos a 0, 1 ou 2).
- Os eventos estão armazenados de forma sequencial (ordem cronológica de chegada) em um vetor contíguo de tamanho n, em que $0 \leq n \leq$ capacidade.
- O objetivo principal do algoritmo é identificar e filtrar apenas os eventos de severidade crítica/falha, ou seja, aqueles em que `codigo` $==$ 2.
- É necessário realizar uma contagem ou agregação da quantidade de falhas (`codigo` $==$ 2) associadas individualmente a cada identificador de equipamento.

QUESTÃO 05

O circuito da Figura 2 é um filtro Sallen–Key com resistores e capacitores iguais na rede de filtragem. Considere um amplificador operacional ideal, operando na região linear:

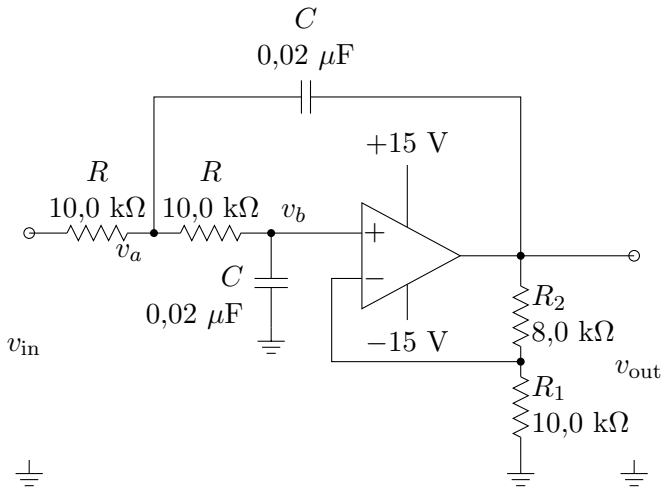


Figure 2: Filtro Sallen–Key

a) **Determine a função de transferência do circuito, $H(s) = \frac{V_{\text{out}}(s)}{V_{\text{in}}(s)}$, considerando condições iniciais nulas. Apresente a expressão algébrica em função dos componentes do circuito (R, C, R_1, R_2).**

GABARITO: As entradas do amplificador não absorvem corrente, portanto, no nó V_b, a lei dos nós no domínio de Laplace fornece:

$$\frac{V_b - V_a}{R} + sCV_b = 0 \implies V_a = (1 + sRC)V_b$$

O ganho de tensão (K) do amplificador não inversor é dado por:

$$V_{\text{out}} = \left(1 + \frac{R_2}{R_1}\right) V_b = K \cdot V_b \implies V_b = \frac{V_{\text{out}}}{K}$$

No nó V_a, o nó intermediário da rede RC, tem-se:

$$\frac{V_a - V_{\text{in}}}{R} + \frac{V_a - V_b}{R} + sC(V_a - V_{\text{out}}) = 0$$

$$V_a(2 + sRC) - V_b - sRCV_{\text{out}} = V_{\text{in}}$$

Substituindo $V_a = \frac{V_{\text{out}}}{K}(1 + sRC)$ e $V_b = \frac{V_{\text{out}}}{K}$, tem-se:

$$\frac{V_{\text{out}}}{K}\left[(1 + sRC)(2 + sRC) - 1 - KsRC\right] = V_{\text{in}}$$

Consequentemente:

$$\boxed{H(s) = \frac{K}{R^2C^2s^2 + (3 - K)RCs + 1}, \qquad K = \left(1 + \frac{R_2}{R_1}\right),}$$

ou,

$$\boxed{H(s) = \frac{\frac{K}{R^2C^2}}{s^2 + \left(\frac{3-K}{RC}\right)s + \frac{1}{R^2C^2}}, \qquad K = \left(1 + \frac{R_2}{R_1}\right).}$$

---

b) **Determine os valores numéricos do ganho em malha fechada (K), da frequência angular natural (ω_0), da frequência de corte (f_0) e do fator de amortecimento (ζ) do circuito apresentado.**

GABARITO: Considerando os componentes da Figura 2 ($R = 8{,}2$ kΩ, $C = 0{,}033$ μF, $R_1 = 10$ kΩ e $R_2 = 8{,}2$ kΩ) e comparando a função de transferência obtida no item anterior com a forma padrão de segunda ordem:

$$H(s) = \frac{\frac{K}{R^2C^2}}{s^2 + \left(\frac{3-K}{RC}\right)s + \frac{1}{R^2C^2}} \qquad \Longleftrightarrow \qquad H(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

Pela característica não inversora do circuito e pela análise comparativa dos termos, obtêm-se as relações literais:

$$K = 1 + \frac{R_2}{R_1}, \qquad \omega_0 = \frac{1}{RC}, \qquad \zeta = \frac{3-K}{2}$$

(a) **Ganho em malha fechada (K):**

$$K = 1 + \frac{R_2}{R_1} = 1 + \frac{8\ \text{k}\Omega}{10\ \text{k}\Omega} \implies \boxed{K = 1{,}8}$$

(b) **Frequência angular natural (ω_0):**

$$RC = (10{,}0 \times 10^3\ \Omega) \cdot (0{,}02 \times 10^{-6}\ \text{F}) = 2{,}0 \times 10^{-4}\ \text{s}$$

$$\omega_0 = \frac{1}{RC} = \frac{1}{2{,}0 \times 10^{-4}\ \text{s}} \implies \boxed{\omega_0 = 5000\ \text{rad/s}}$$

(c) **Frequência de corte (f_0):**

$$f_0 = \frac{\omega_0}{2\pi} = \frac{5000}{2\pi} \implies \boxed{f_0 \approx 795{,}77\ \text{Hz}}$$

(d) **Fator de amortecimento (ζ):** Igualando os termos em s: $2\zeta\omega_0 = \dfrac{3-K}{RC} = (3-K)\omega_0 \implies 2\zeta = 3 - K$

$$\zeta = \frac{3 - 1{,}8}{2} = \frac{1{,}2}{2} \implies \boxed{\zeta = 0{,}6}$$

---

c) **Com base nos parâmetros obtidos no item b, analise o comportamento dinâmico do sistema. Avalie a estabilidade do filtro e a resposta transitória ao degrau. Caso estável, indique a condição técnica para que o circuito se torne instável.**

GABARITO: A partir do polinômio de 2ª ordem e dos parâmetros calculados, pode-se explicitar as seguintes características do sistema:

i) **Estabilidade Assintótica:** Como o fator de amortecimento é estritamente positivo ($\zeta = 0{,}6 > 0$), os coeficientes do polinômio característico são todos positivos ($2\zeta\omega_0 > 0$ e $\omega_0^2 > 0$). Pelo critério de Routh–Hurwitz, os polos do sistema possuem parte real estritamente negativa, garantindo que o sistema é assintoticamente estável.

ii) **Condição de Instabilidade:** No filtro Sallen–Key de componentes iguais, o termo de amortecimento é dado por $\zeta = \frac{3-K}{2}$. O circuito só se tornará instável ($\zeta \leq 0$) se o ganho em malha fechada K for igual ou superior a 3,0 ($K \geq 3$), o que faria os polos se deslocarem para o eixo imaginário ou para o semiplano direito. Como $K = 1{,}8 < 3$, a estabilidade está assegurada.

iii) **Comportamento Transitório (Oscilações):** Como o fator de amortecimento é menor que a unidade ($0 < \zeta < 1$), o sistema opera no regime subamortecido. À aplicação de um degrau de tensão na entrada, a resposta no tempo apresentará um comportamento oscilatório amortecido antes de atingir o valor final em regime permanente, apresentando um ganho de pico (sobresinal/overshoot) no domínio do tempo.

---

d) **Considerando a resposta a um degrau unitário na entrada, determine a primeira tensão máxima e a primeira tensão mínima atingidas pela saída durante o período transitório. (*Considere* $e^{-0{,}75\pi} \approx 0{,}10$).**

GABARITO: Para uma entrada em degrau unitário ($V_{\text{in}} = 1$ V), o valor da tensão em regime permanente ($t \to \infty$) é dado pelo ganho em malha fechada do circuito:

$$V_{\text{final}} = K \cdot 1 \text{ V} = 1{,}8 \text{ V}$$

Como o circuito opera no regime subamortecido ($0 < \zeta \approx 0{,}6 < 1$), a resposta no tempo apresenta comportamento oscilatório em torno do valor de regime permanente.

i) **Primeira Tensão Máxima ($V_{\max}$ – 1º Pico / Sobresinal):** Ocorre no tempo de pico $t_p = \frac{\pi}{\omega_d}$. O sobresinal relativo M_p é calculado por:

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

Substituindo os parâmetros obtidos ($\zeta = 0{,}6$ e $\sqrt{1-\zeta^2} = 0{,}8$):

$$M_p = e^{-\frac{0{,}6\pi}{0{,}8}} = e^{-0{,}75\pi} \approx 0{,}10 \quad (10\%)$$

$$V_{\max} = V_{\text{final}} \cdot (1 + M_p) = 1{,}8 \cdot (1 + 0{,}10) = 1{,}8 \cdot 1{,}10 \implies \boxed{V_{\max} \approx 1{,}98 \text{ V}}$$

ii) **Primeira Tensão Mínima ($V_{\min}$ – 1º Vale / Undershoot do Transitório):** Ocorre no instante do primeiro vale após ultrapassar o valor final ($t = 2t_p = \frac{2\pi}{\omega_d}$):

$$V_{\min} = V_{\text{final}} \cdot \left(1 - e^{-\frac{2\zeta\pi}{\sqrt{1-\zeta^2}}}\right) = V_{\text{final}} \cdot (1 - M_p^2)$$

Como $M_p^2 \approx (0{,}10)^2 = 0{,}01$:

$$V_{\min} = 1{,}8 \cdot (1 - 0{,}01) = 1{,}8 \cdot 0{,}99 \implies \boxed{V_{\min} \approx 1{,}782 \text{ V}}$$

---

e) **Deseja-se limitar o sobresinal (M_p) da resposta ao degrau a no máximo 5% sem alterar as frequências características do filtro. Assuma como aproximação de projeto a condição $\zeta = \frac{\sqrt{2}}{2} \approx 0{,}707$, na qual $M_p \approx 4{,}3\% \leq 5\%$. Indique a modificação mais simples a ser realizada no circuito e determine a nova relação entre os componentes afetados.**

GABARITO: Como a frequência angular natural ($\omega_0 = \frac{1}{RC}$) depende exclusivamente dos capacitores e resistores da malha do filtro, os componentes R e C devem permanecer inalterados para preservar a resposta em frequência.

A alteração mais simples consiste em ajustar o ganho em malha fechada K através da razão entre os resistores de ganho (R_1 e R_2).

Foi-se calculado que, no filtro Sallen–Key:

$$\zeta = \frac{3 - K}{2}$$

Para garantir a condição de Butterworth ($\zeta = \frac{\sqrt{2}}{2} \approx 0{,}707$):

$$\frac{\sqrt{2}}{2} = \frac{3 - K}{2} \implies K = 3 - \sqrt{2} \approx 1{,}59$$

Como $K = 1 + \frac{R_2}{R_1}$, a nova relação entre os resistores de ganho deve ser:

$$1 + \frac{R_2}{R_1} \leq 1{,}59 \implies \boxed{\frac{R_2}{R_1} \leq 0{,}59}$$

Tomam-se como possíveis soluções para a limitação do sobresinal a menos que 5%:

(a) Reduzir a razão $\frac{R_2}{R_1}$ de 0,82 para no máximo 0,59.

(b) Reduzir R_2 para $R_2 \leq 5{,}9$ kΩ (mantendo $R_1 = 10$ kΩ).

(c) Aumentar R_1 para $R_1 \geq 13{,}9$ kΩ (mantendo $R_2 = 8{,}2$ kΩ).

PONTOS CHAVE:

- Modelagem do sistema alcançando a função de transferência de 2º ordem em função dos componentes e do ganho K;
- Determinação numérica correta dos parâmetros do circuito (K, ω_0, f_0, ζ e f_d) acompanhados de suas respectivas unidades;
- Indicação de estabilidade assintótica via $\zeta > 0$ e do limite técnico de instabilidade ($K \geq 3{,}0$);
- Identificação do regime subamortecido ($0 < \zeta < 1$) com resposta oscilatória e sobresinal;
- Identificação que o sobresinal pode ser facilmente ajustado pelo ajuste de R_1 e R_2.

Banca Examinadora

Documento assinado digitalmente
gov.br
MARCIO FANTINI MIRANDA
Data: 20/10/2026 14:47:53-0300
Verifique em https://validar.iti.gov.br

Presidente: Prof. _____

Documento assinado digitalmente
gov.br
NATHAN AUGUSTO ZACARIAS XAVIER
Data: 20/10/2026 11:31:53-0300
Verifique em https://validar.iti.gov.br

Secretário(a): Prof. _____ _____

Documento assinado digitalmente
gov.br
WILLIAM CAIRES SILVA AMORIM
Data: 20/10/2026 11:36:00-0300
Verifique em https://validar.iti.gov.br

Membros: Prof. _____ _____

Documento assinado digitalmente
gov.br
MARCIO JUNIOR NUNES
Data: 20/10/2026 11:49:50-0300
Verifique em https://validar.iti.gov.br

Prof. _____ _____

Documento assinado digitalmente
gov.br
TIAGO ALVES DE OLIVEIRA
Data: 20/10/2026 11:39:57-0300
Verifique em https://validar.iti.gov.br

Prof. _____ _____