



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
CENTRO DE TECNOLOGIA  
ESCOLA DE QUÍMICA**

Controle e Instrumentação de Processos - EQE487

**Trabalho 2**

**Grupo 5**

**Alunos:** Bruno Videira Pinho e Luisa de Hollanda Passos

**Prof.:** Bruno Capron

Rio De Janeiro, 2025

Os modelos xcos/zcos e os scripts utilizados para a resolução de problemas se encontram no seguinte repositório web dentro: [github link](#) (<https://github.com/Facul-101/controle-2025/tree/main/T2>).

Toda resolução do segundo trabalho está dentro da pasta “T2”, tanto os códigos quanto as simulações. Dentro de tal pasta existe uma pasta para cada questão (4 pastas no total). Todas as questões utilizam um script de utilidades, funções utilizadas em diversos scripts. Para que o script funcione, é necessário abrir a pasta “controle-2025” como o diretório atual do scilab. Caso contrário os scripts não conseguirão importar as funções implementadas no arquivo de utilidades (“utils.sci”).

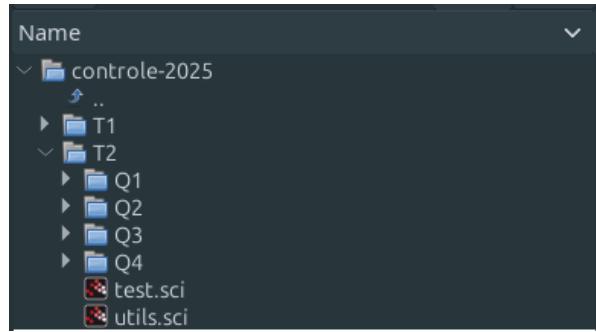


Imagen 1: Diretório que deve ser aberto no acilab

Como o repositório foi estruturado de acordo com as questões, a lógica por trás da resolução de cada questão será abordada durante a discussão dos resultados de cada questão. Além de arquivos “.sci” (script) e “.zcos” (simulação), pode-se encontrar “.scg” e “.png” que foram gerados a partir dos scripts eles carregam as informações gráficas geradas pelos scripts, o “.scg” seria um arquivo específico do scilab usado para guardar resultados gráficos.

OBS: como referência toda vez que “função utilitária” ou “função utilidade” for mencionada isso significa que a função em questão se encontra no arquivo de utilidade “T2/utils.sci”.

## 1) Diagrama de bode de um sistema MA com coeficientes e funções pré determinadas

O script “T2/Q1/Q1.sci” é capaz de:

- Calcular a equação de  $G_{ol}$ , com os parâmetros determinados na questão e as equações determinadas no trabalho, pela função utilitária “get\_gol”.
- Gerar o diagrama de bode do sistema
- Calcular as margens de ganho e de fase pela função “get\_margins” que na verdade apenas junta as função padrão do scilab “g\_margin” e “p\_margin” numa só função.
  - Como a “g\_margin” retorna um ganho em decibel o valor de retorno na mesma foi para o ganho de margem foi ajustado pela seguinte equação:

$$GM = 10^{\frac{GM_{dB}}{20}}$$

- Note que essa conversão não seria necessário pois para determinar se um sistema é estável ou não com GM em decibel basta validar se  $GM_{dB} > 0$ , se tal afirmação for verdade o sistema é estável. Porém, como tal equação foi

- explicitamente mencionada no vídeo de apoio a transformação foi implementada.
- Determinar se o sistema é estável ou instável

Resultados obtidos:

Este foi o diagrama gerado pelo código:

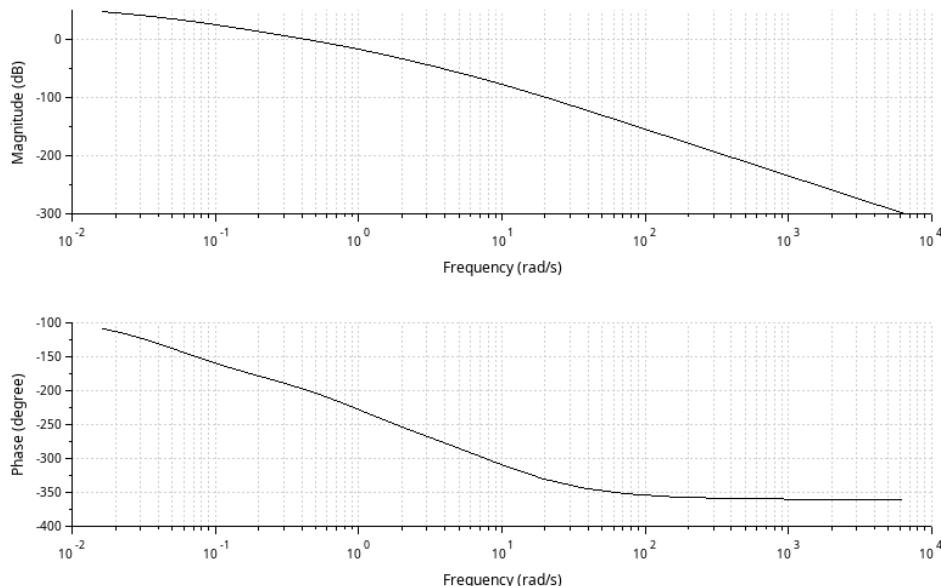


Gráfico 1: Diagrama de bode questão 1 ( $K_c = 70$ ,  $\tau_I = 5$ )

Este foi o resultado final obtido pelo código:

```
This system is unstable
- GM: 0.242514
- PM: -18.622993 rads/s

0.196
-----
0.05s +1.055s^2 +1.105s^3 +0.1s^4
```

Note que com  $GM < 1$ , o sistema é instável. A equação que aparece na resposta é justamente o  $G_{ol}$ .

## 2) Diagrama de bode de um sistema MA com coeficientes (diferentes do anterior) e funções pré determinadas

Pela natureza do script desenvolvido, a resolução desta questão é muito similar à anterior, mudando apenas o coeficiente de  $K_c$  saiu de 70 para 7. Por esse motivo, o código implementado para esta questão tem as mesmas capacidades da questão anterior.

## Resultados obtidos:

Este foi o diagrama gerado pelo código:

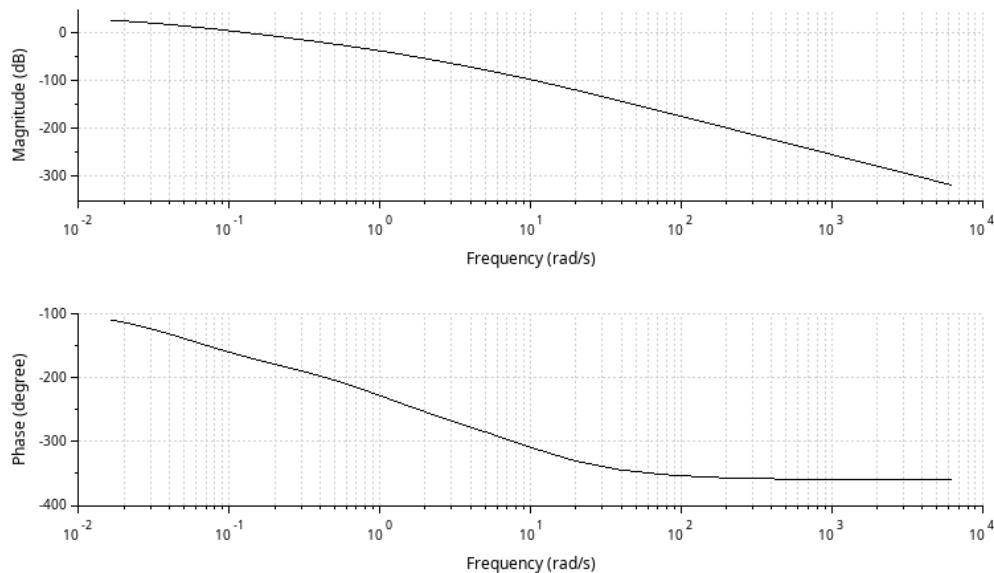


Gráfico 2: Diagrama de bode questão 1 ( $K_c = 7$ ,  $\tau_I = 5$ )

Este foi o resultado final obtido pelo código:

```
This system is stable
- GM: 2.425143
- PM: 11.870639 rads/s

0.0196
-----
0.05s +1.055s^2 +1.105s^3 +0.1s^4
```

Note que dessa vez  $GM > 1$  então o sistema é estável. E a única mudança de entre a equação de  $G_{ol}$  obtida nesta questão e na anterior é o seu valor de  $K$  diminuiu em 10 vezes, a mesma relação de diminuição entre os valores de  $K_c$ , como esperado.

### 3) Parametrização do controlador

Nessa questão uma série de restrições são passadas pela questão para que o “melhor” controlador possível seja encontrado. A resolução do problema está de acordo com todas as restrições impostas. Porém, uma breve modificação foi implementada em uma delas, na função objetivo. Sua implementação sugerida fazia com que os valores de GM e PM fossem para um dos limites dos intervalos (GM e PM sempre iam para o menor valor possível). Com a função modificada GM tende a 4 e PM tende a 45:

$$F_o(GM, PM) = \frac{|4 - GM|}{4} + \frac{|45 - PM|}{45}$$

Para a determinação do valor ótimo, foi implementado um algoritmo PSO (particle swarm optimization). Sua implementação se encontra no arquivo de funções de utilidade e para a resolução do problema foram utilizadas 50 partículas e um total de 125 iterações. O número

de partículas e iterações foi fixado em valores bem altos (para o sistema) de maneira premeditada, para inferir que um bom valor ótimo será encontrado na primeira vez que o sistema for executado. Note que o método implementado tem característica estatística e nem sempre vai resultar no mesmo valor.

## Resultados obtidos:

```
PSO minimization result
Kc: 24.270962
Tau: 150.000000
Ki: 0.161806
GM: 4.000000
PM: 40.027302 rads/s
Fo: 0.110504
```

Note que o valor de tau\_I acabou ficando preso no limite superior. Tal comportamento é comum nesse tipo de algoritmos dependendo da superfície do sistema. O algoritmo foi executado mais 2 vezes e o mesmo resultado foi obtido, isso significa que muito provavelmente o “melhor” foi encontrado. Para garantir tal afirmação o espaço do sistema deveria ser conhecido, para tal uma função utilitária foi implementada “plot\_grath”. No script da questão atual ela foi comentada pois demora bastante para finalizar. Esse foi o resultado obtido:

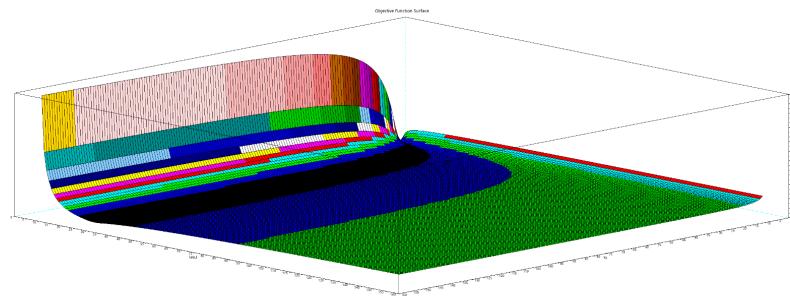


Gráfico 3: Superfície Kc vs Tau\_I vs Fo

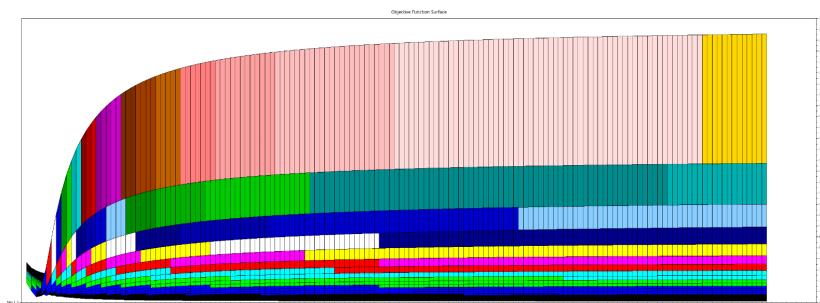


Gráfico 4: Superfície Kc vs Tau\_I

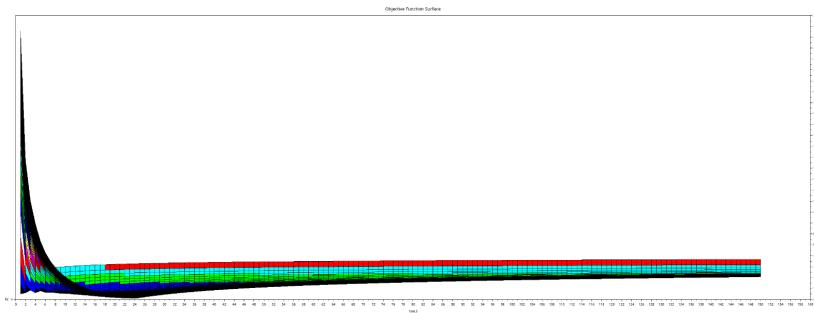


Gráfico 6: Superfície Tau\_I vs Fo

Note que o gráfico 6 é bem claro, realmente o resultado obtido pode ser considerado como um bom valor “ótimo” para o sistema. Por consequência, os valores de margem encontrados pelo script são válidos como resposta para a pergunta.

OBS: um gráfico de bode também pode ser gerado pelo script (basta descomentar sua linha)

#### 4) Resposta do sistema em malha fechada

Para a resolução do problema duas simulações foram implementados, uma para mudança no setpoint e a outra no distúrbio:

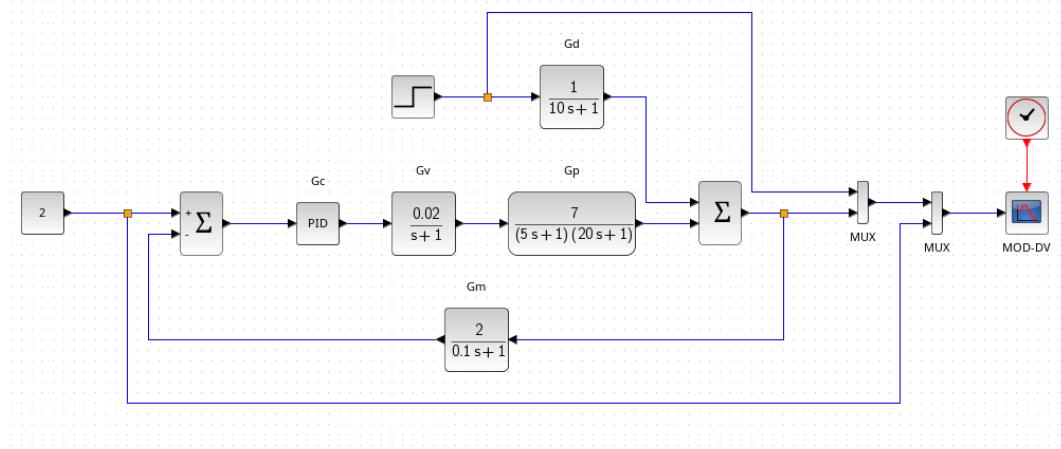


Imagen 2: Exemplo de uma das simulações implementadas

#### Resultados obtidos:

Para mudança no setpoint o gráfico gerado se encontra a seguir. Seguindo a mesma lógica de cores que o trabalho anterior **Vermelho** referência o SP(s), **verde** o CV(s) e preto o DV(s).

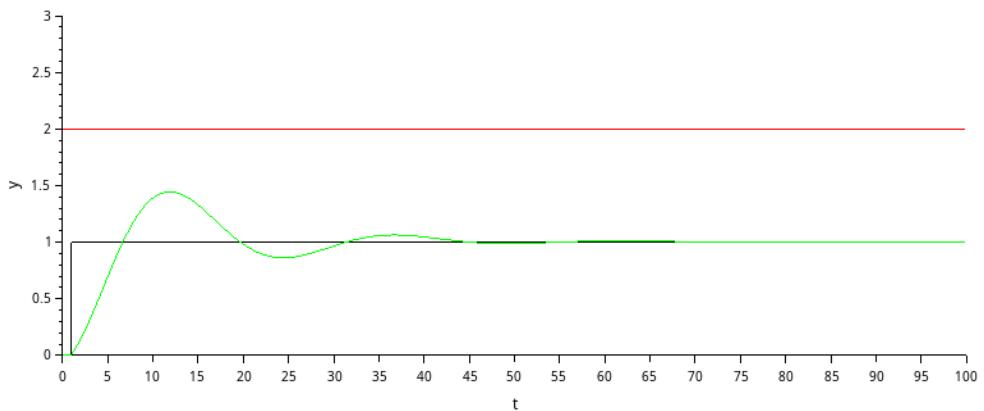


Gráfico 7: Degrau unitário no distúrbio

Note que como  $K_m1$  é igual a 2 o sinal de entrada para  $SP(s)$  se inicia em 2. Para esse sistema o controlador se comporta bem levando o sinal ao valor desejado (1) em menos de 100 segundos.

Uma pequena observação, também foi gerado um sinal sem degrau, não está presente neste trabalho, e a curva de  $CV(s)$  não mudou muito (não conseguir identificar mudanças visualmente) o que leva a acreditar que o degrau no distúrbio interfere pouco se ele acontecer nos momentos iniciais.

Para tal, um novo gráfico foi gerado com a mesma simulação anterior. Mudando apenas o momento do distúrbio e a duração da simulação.

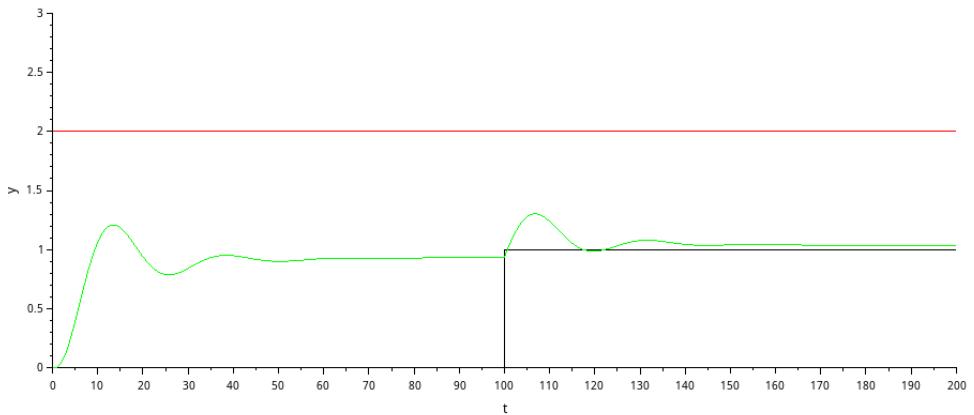


Gráfico 9: Degrau unitário no distúrbio depois do sistema estabilizar

Note que antes do degrau o sinal entra em estado estacionário, e depois do degrau o sistema entra em estado de equilíbrio relativamente rápido (seguindo uma margem de erro aceitável  $1.05 \approx 1$ ) e com uma amplitude inicial resultante do degrau bem baixa.

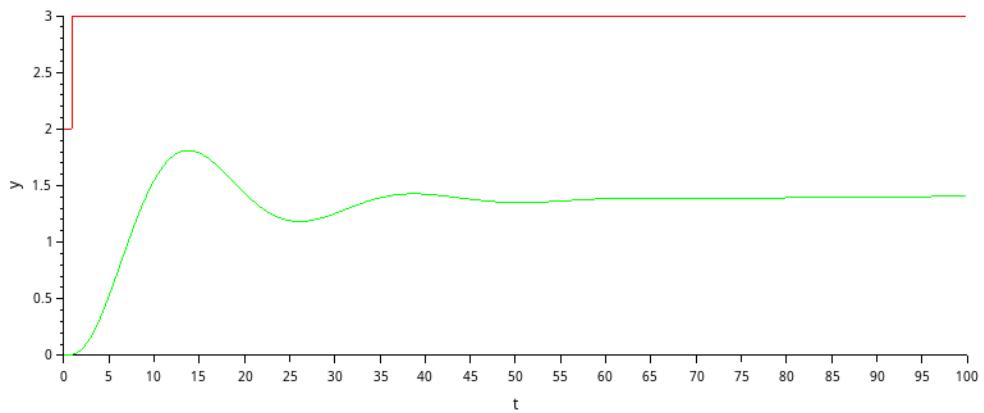


Gráfico 9: Degrau unitário no setpoint

Assim como no sistema anterior,  $SP(s)$  inicia em 2 e então aumenta para 3. Nesse sistema o controlador também se comporta bem, levando o sinal para o valor desejado (1,5) em menos de 100 segundos.

Como resultado final é importante afirmar que todos os sistemas testados são estável, como esperado.