

SHP_projeto_portas

January 27, 2025

```
[1]: import numpy as np
import os
import matplotlib.pyplot as plt
import pandas as pd

PATH = os.getcwd()
PATH
```

```
[1]: '/home/andre/Documents/Git/SHP_MEC1515'
```

1 Funções

1.1 Compressor

- Q_c : somatório da vazão demandada por cada elemento
- K : Fator que varia entre 1,2 e 1,5 (leva em consideração os vazamentos ou eventuais ampliações na planta pneumática)
- I : Coeficiente de inserção
 1. T_t : Tempo de funcionamento do compressor
 2. T_s : Tempo de parada do compressor

```
[2]: class Reservatorio:
    def __init__(self, Tt, Ts, Qc):
        self.Tt = Tt
        self.Ts = Ts
        self.Qc = Qc
        self.I = 0
        self.K = 0

    # Consideracao dos vazemaetos (entre 1.2 e 1.5)
    def set_K(self, K):
        self.K = K

    # I
    def coeficiente_de_insercao(self):
        self.I = (self.Tt / (self.Tt + self.Ts))
        return self.I
```

```
def vazao_compressor(self):
    return (self.Qc * self.K / self.I)
```

```
[3]: def coeficiente_de_insercao(Tt, Ts):
    return (Tt / (Tt + Ts))

def vazao_compressor(Qc, K, I):
    return (Qc * K / I)
```

1.2 Atuadores (Cilindros)

- D : diâmetro do pistão (mm)
- d : diâmetro da haste (mm)
- c : curso do pistão (mm)
- n : número de ciclos por minuto (geralmente por ciclo entende-se uma fase de avanço e recuo)
- pr : pressão relativa de regime (MPa - 0,1 MPa = 1 bar)
- pa : pressão absoluta de regime (MPa - $pa = pr + 0,1$)
- pan : pressão absoluta de regime da câmara negativa (lado haste)
- Q : consumo do ar (litros / min)

```
[4]: class Atuador:
    def __init__(self, D, d, c, pa, n, pressao_igual):
        self.D = D
        self.d = d
        self.c = c
        self.pa = pa
        self.n = n
        self.pressao_igual = pressao_igual

        self.consumo = 0

    # Consumo de ar de cilindros d0e simples efeito
    def consumo_cilindros_simples(self):
        self.consumo = (self.D**2 * self.c * self.pa * self.n / 127000)
        return self.consumo

    # Consumo de ar de cilindros de duplo efeito
    def consumo_cilindros_duplo_efeito(self):
        if self.pressao_igual == False:
            self.consumo = (self.c * self.n * (self.D**2 * self.pa + (self.
↪D**2 - self.d**2) \
                * self.pa * self.n / 127000))
            return self.consumo

        self.consumo = (self.D**2 * self.c * self.pa * self.n / 63500)

        return self.consumo
```

```
[5]: # Consumo de ar de cilindros d0e simples efeito
def consumo_cilindros_simples(D, c, pa, n):
    return (D**2 * c * pa * n / 127000)

# Consumo de ar de cilindros de duplo efeito
def consumo_cilindros_duplo_efeito(D, d, c, n, pa, pressao_igual):
    if pressao_igual == False:
        return (c * n * (D**2 * pa + (D**2 - d**2) * pa * n / 127000))

    return (D**2 * c * pa * n / 63500)
```

1.3 Reservatório

- pr : pressão relativa (MPa)
- pa_0 : pressão absoluta do ar no estado livre (MPa)
- V_0 : Volume de ar no estado livre (m^3)
- pa_1 : Pressão absoluta do ar armazenado (MPa)
- V_1 : Capacidade Total (Vazão total) em m^3

```
[6]: def capacidade_reservatorio(pr, V0, pa0=0.1):
    pa1 = pr + pa0
    return (V0 * pa0 / pa1)
```

2 Dados do projeto

```
[7]: # Perimetro total
perimetro_tubulacao = 2*18 + 7*13 + 2.0

# Quantidade extensoes que irao para
# os atuadores das portas
extensoes = 20

# Comprimento equivalente dos pontos
# de estrangulamentos
Ts = 11 * 1.7
joelho_90 = 20 * 0.7 + 4 * 0.7
joelho_45 = 20 * 0.4
valvulas_de_pressao = 2.4

Comprimento_total_da_tubulacao = perimetro_tubulacao + Ts + \
    joelho_45 + joelho_90 + valvulas_de_pressao

print(f'Perimetro total (m): {perimetro_tubulacao}')
print(f'extensoes: {extensoes}')
print(f'Comprimento equivalente dos Ts: {Ts}')
print(f'Comprimento equivalente dos joelhos de 90 graus: {joelho_90}')
print(f'Comprimento equivalente dos joelhos de 45 graus: {joelho_45}')
```

```
print(f'Comprimento total: {Comprimento_total_da_tubulacao} m')
```

Perimetro total (m): 129.0

extensoes: 20

Comprimento equivalente dos Ts: 18.7

Comprimento equivalente dos joelhos de 90 graus: 16.8

Comprimento equivalente dos joelhos de 45 graus: 8.0

Comprimento total: 174.9 m

2.1 Preço Componentes

2.1.1 Links

1. Pneu_3WaysValve:

- [Pneu_3WaysValve](#) (operation: 0 - 8 bar)
- [Pneu_3WaysValve](#) (operation: 0 - 8 bar)
- [Pneu_3WaysValve](#) (operation: 0 - 8 bar)

2. Pneu_RegisterValve:

- [Pneu_RegisterValve](#) (operation: 0 - 16 bar)
- [Pneu_RegisterValve](#) (operation: 0 - 6 bar)

3. Pneu_DrainValve:

- [Pneu_DrainValve](#) (operation: 0 - 16 bar)
- [Pneu_DrainValve](#) (operation: 0 - 8 bar)
- [Pneu_DrainValve](#) (operation: 0 - 8 bar)

4. Pneu_Actuator:

- [Pneu_Actuator](#) (operation: 0 - 8 bar)
- [Pneu_Actuator](#) (pages 205 and 251)
- [Pneu_Actuator](#)

5. Lubrifil

- [Lubrifil](#)
- [Lubrifil](#)
- [Lubrifil](#)
- [Lubrifil](#)

6. Compressor

- [Compressor](#) (PCM: 20)
- [Compressor](#) (PCM: 20)
- [Compressor](#) (PCM: 20)

```
[8]: # 20 salas com uma valvula para o acionamento da porta
Pneu_3WaysValve = np.mean([646.95, 710.97, 789.99]) * 22

# 20 salas com 1 registro em cada
Pneu_RegisterValve = np.mean([21.75, 26.26]) * 23

# 2 valvulas de dreno para o sistema, uma qm cada extremidade do predio
Pneu_DrainValve = np.mean([155.97, 209.90, 190.97]) * 2

# 15 janelas e 2 atuadores em cada
Pneu_Actuator = np.mean([119.93, 340.55, 413.97]) * 15 * 2

Lubrifil = np.mean([48.99, 66.54, 90.0, 54.9])

Compressor = np.mean([6199 + 381.23, 6799.90 + 305.96, 4299 + 319.05])

total = Pneu_3WaysValve + Pneu_Actuator + Pneu_DrainValve + \
        Pneu_RegisterValve + Lubrifil + Compressor

print(f'Valvulas direcionais: R$ {Pneu_3WaysValve}')
print(f'Valvulas de Registro: R$ {Pneu_RegisterValve}')
print(f'Valvulas de dreno: R$ {Pneu_DrainValve}')
print(f'Janelas: R$ {Pneu_Actuator}')
print(f'Lubrifil: R$ {Lubrifil}')
print(f'Compressor: R$ {Compressor}')
print(f'Total: R$ {total}')
```

```
Valvulas direcionais: R$ 15751.339999999998
Valvulas de Registro: R$ 552.115
Valvulas de dreno: R$ 371.22666666666667
Janelas: R$ 8744.5
Lubrifil: R$ 65.1075
Compressor: R$ 6101.38
Total: R$ 31585.669166666663
```

```
[12]: atuadores_janelas = Atuador(40, 40, 250, 1, 4, False)
vazao_janelas = atuadores_janelas.consumo_cilindros_simples() * 15 * 2

print(f'Vazao Janelas: {vazao_janelas} L/min')
print(f'Vazao Janelas: {vazao_janelas / 1e3} m^3/min')
print(f'Vazao Janelas: {vazao_janelas / 1e3 * 35.314} PCM')
```

```
Vazao Janelas: 377.9527559055118 L/min
Vazao Janelas: 0.3779527559055118 m^3/min
Vazao Janelas: 13.347023622047244 PCM
```