

Cálculo da taxa de corrosão em estruturas metálicas.

Clara Ferreira e João Vinícius Duarte, Prof Drº Robson
Marinho da Silva
Salvador, BA, junho de 2024

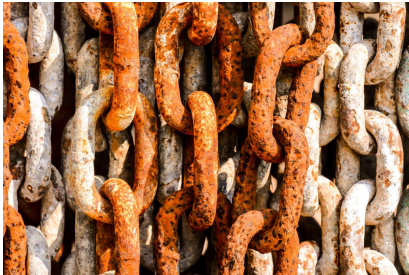
Universidade do Estado da Bahia (UNEB)



Sumário

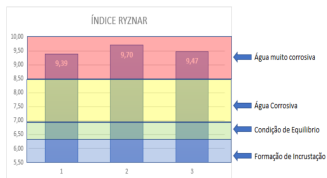
- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

Introdução:



A corrosão é um processo natural de deterioração dos metais causado por reações químicas ou eletroquímicas entre o metal e o meio corrosivo. No Brasil, as construções com estruturas metálicas enfrentam desafios devido às variações climáticas, salinidade e à poluição, o que pode causar anomalias nas estruturas, comprometendo sua integridade.

Introdução:



Para avaliar a água em termos de agressividade e incrustação são utilizados métodos indiretos de avaliação como o índice de Saturação e o índice de estabilidade de Ryznar (IER). Os valores obtidos para a água analisada sob o parâmetro proposto por Ryznar poderá ser analisado através da relação entre o valor obtido para o IER e o estágio da corrosão. A severidade da corrosão, ou seja, a relação do estado de saturação do carbonato de cálcio e a camada formada podem ser classificadas pelo (IER).

Motivações e justificativas

- O desenvolvimento de um programa em Python para calcular taxas de corrosão e avaliar a qualidade da água é motivado pela necessidade de ferramentas precisas, eficientes e acessíveis na engenharia civil. Este programa aborda questões relacionadas à integridade estrutural, a identificação precoce de altas taxas de corrosão e água agressiva, permitindo a tomada de medidas preventivas e contribuindo, dessa forma, para uma infraestrutura mais segura e durável.

- Sendo assim, a criação de um programa em linguagem Python para o cálculo da taxa de corrosão em estruturas metálicas é crucial para a avaliação correta do risco de degradação, implementação de estratégias de manutenção, a segurança operacional, planejamento de manutenções preventivas, evitando reparos emergenciais caros e a sustentabilidade de infraestruturas metálicas em diversas áreas da engenharia.

Sumário

- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

OBJETIVO: A proposta do programa, nessa fase intermediária, consiste no cálculo da taxa de penetração da corrosão (TPC) ou a perda da espessura do material de estruturas metálicas, possibilitando a categorização da taxa de corrosão em função de sua intensidade. Além disso, o programa faz a análise e classificação da qualidade da água, calculando o pH de saturação e o índice de Estabilidade de Ryznar (IER), os valores obtidos para a água investigada sob o parâmetro proposto por Ryznar poderá ser analisado através da relação entre o valor obtido para o IER e o estágio da corrosão que são indicadores importantes para informar sobre a propensão da água a causar incrustações ou ser agressiva.

Sumário

- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

Importação de módulos

- O programa começa importando o módulo “metal potenciais”, que contém as funções para obter propriedades de metais.
- Importa o módulo “math” para realizar cálculos matemáticos, como “log10”, necessários para cálculos químicos.

Definição da classe “Metal”

- A classe “Metal” é definida para representar as propriedades e cálculos relacionados à corrosão de metais.
- O método “__init__” inicializa os atributos da classe “metal” (tipo de metal), “corrente” (corrente de corrosão em amperes), “area” (área do eletrodo em cm^2) e “aquoso” (indicador se o metal está em meio aquoso).
- Define a constante de Faraday “F” (96485 C/mol), usada nos cálculos de corrosão.

Método para calcular a taxa de corrosão

- O método “calcular taxa corrosao” obtém as propriedades do metal (“E padrao”, “massa molar”, “n”) usando a função “obter propriedades metal” do módulo “metal potenciais”.
- Verifica se o metal foi encontrado na tabela de potenciais padrão. Se não, imprime uma mensagem de erro e retorna “None”.
- Calcula a taxa de oxidação em g/s por unidade de área e retorna esse valor.

Método para classificar a corrosão

- O método “classificar corrosao” classifica a taxa de corrosão em “Alto”, “Moderado” ou “Baixo” com base no valor da taxa fornecida.

Definição da classe “Água”

- A classe “Água” é definida para realizar cálculos relacionados à qualidade da água.
- O método “calcular pHS” calcula o pH de saturação (pHS) usando temperatura, STD (Sólidos Totais Dissolvidos), concentração de Ca^{+2} de CaCO_3 , e alcalinidade de CaCO_3 .
- Utiliza funções matemáticas como “math.log10” para calcular os termos.

Método para calcular o Índice de Estabilidade de Ryznar (IER)

- O método “calcular IER” calcula o Índice de Estabilidade de Ryznar (IER) com base no pH de saturação (pHS) e o pH real da água.

Método para classificar a água

- O método “classificar agua” classifica a água em quatro categorias com base no valor do IER. Por exemplo:
- Com base no Índice de Estabilidade de Ryznar (IER), o programa determina o nível de agressividade da água.
- Para valores de IER inferiores a 6,5, a água é classificada como “não incrustante e agressiva”.
- Para valores de IER entre 6,5 e 7, a água é considerada “não incrustante e ligeiramente agressiva”.
- Quando o IER está entre 7 e 8, a água é identificada como “incrustante e ligeiramente agressiva”.
- Caso contrário, o programa imprime “água é incrustante”.

Solicitando entradas do usuário

- O programa solicita entradas do usuário para o metal e suas propriedades, bem como para a análise da água.

Instanciando a classe “Metal” e calculando a taxa de corrosão

- Cria uma instância da classe “Metal” com os dados fornecidos pelo usuário.
- Chama o método “calcular taxa corrosao” para calcular a taxa de corrosão.
- Se a taxa de corrosão for calculada com sucesso, chama o método “classificar corrosao” para determinar o nível de corrosão e imprime os resultados.

Calculando pHS e IER

- Usa os métodos estáticos da classe “Água” para calcular o pH de saturação (pHS), o Índice de Estabilidade de Ryznar (IER) e a classificação da água.
- Imprime os resultados para o usuário.

Sumário

- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
biblioteca
biblioteca
Programa Principal

- 5 Rede petri

O programa: biblioteca

```
# metal_potenciais.py

# Definindo os dados de potenciais padrão e outras propriedades
v dados_potenciais = {
    'Li': {'E° (V)': 3.04, 'Massa Molar (g/mol)': 6.94, 'n': 1},
    'K': {'E° (V)': 2.92, 'Massa Molar (g/mol)': 39.10, 'n': 1},
    'Ca': {'E° (V)': 2.87, 'Massa Molar (g/mol)': 40.08, 'n': 2},
    'Na': {'E° (V)': 2.71, 'Massa Molar (g/mol)': 22.99, 'n': 1},
    'Mg': {'E° (V)': 2.37, 'Massa Molar (g/mol)': 24.31, 'n': 2},
    'Al': {'E° (V)': 1.66, 'Massa Molar (g/mol)': 26.98, 'n': 3},
    'Zn': {'E° (V)': 0.76, 'Massa Molar (g/mol)': 65.38, 'n': 2},
    'Cr': {'E° (V)': 0.74, 'Massa Molar (g/mol)': 52.00, 'n': 3},
    'Fe': {'E° (V)': 0.44, 'Massa Molar (g/mol)': 55.85, 'n': 2},
    'Ni': {'E° (V)': 0.25, 'Massa Molar (g/mol)': 58.69, 'n': 2},
    'Pb': {'E° (V)': 0.13, 'Massa Molar (g/mol)': 207.2, 'n': 2},
    'H2': {'E° (V)': 0.00, 'Massa Molar (g/mol)': 2.02, 'n': 2},
    'Cu': {'E° (V)': -0.34, 'Massa Molar (g/mol)': 63.55, 'n': 2},
    'Ag': {'E° (V)': -0.80, 'Massa Molar (g/mol)': 107.87, 'n': 1},
    'Au': {'E° (V)': -1.50, 'Massa Molar (g/mol)': 196.97, 'n': 1}
}
```

O programa: biblioteca

```
def obter_propriedades_metal(metal):  
    """  
    Função para obter o potencial padrão de oxidação, massa molar e n de um metal.  
  
    Args:  
    metal (str): Símbolo do metal (ex: Zn, Cu).  
  
    Returns:  
    tuple: (E° (V), Massa Molar (g/mol), n) se o metal for encontrado, caso contrário (None, None, None).  
    """  
    propriedades = dados_potenciais.get(metal)  
    if propriedades:  
        return propriedades['E° (V)'], propriedades['Massa Molar (g/mol)'], propriedades['n']  
    else:  
        return None, None, None
```

O programa: Programa Principal

```

v import metal_potenciais
import math

v class Metal:
v     def __init__(self, metal, corrente, area, aquoso):
        self.metal = metal
        self.corrente = corrente
        self.area = area
        self.aquoso = aquoso
        self.F = 96485 # Constante de Faraday em C/mol

v     def calcular_taxa_corrosao(self):
        E_padrao, massa_molar, n = metal_potenciais.obter_propriedades_metal(self.metal)

v         if E_padrao is None:
            print("Metal não encontrado na tabela de potenciais padrão.")
            return None

        # Taxa de oxidação em g/s por unidade de área
        taxa_ox = (self.corrente * massa_molar) / (n * self.F * self.area)

        return taxa_ox
```

O programa: Programa Principal

```
def classificar_corrosao(self, taxa):  
    if taxa > 1e-4:  
        return "Alto"  
    elif taxa > 1e-6:  
        return "Moderado"  
    else:  
        return "Baixo"
```

```
class Agua:
```

```
    @staticmethod  
    def calcular_pHS(temperatura, STD, Ca_CaCO3, alcalinidade_CaCO3):  
        A = (math.log10(STD) - 1) / 10  
        B = -13.12 * math.log10(temperatura + 273) + 34.55  
        C = math.log10(Ca_CaCO3) - 0.4  
        D = math.log10(alcalinidade_CaCO3)  
        pHS = (9.3 + A + B) - (C + D)  
        return pHS
```


O programa: Programa Principal

```
@staticmethod
def calcular_IER(pHS, pH_real):
    return 2 * pHS - pH_real

@staticmethod
def classificar_agua(IER):
    if IER < 6.5:
        return "Água não incrustante, água agressiva"
    elif 6.5 <= IER <= 7:
        return "Água não incrustante, ligeiramente agressiva"
    elif 7 < IER <= 8:
        return "Água incrustante, ligeiramente agressiva"
    else:
        return "Água incrustante"

# Solicitando entrada do usuário
metal_input = input("Digite o metal (ex: Zn, Cu): ")
corrente_input = float(input("Digite a corrente de corrosão observada (em amperes): "))
area_input = float(input("Digite a área do eletrodo (em cm^2): "))
aquoso_input = input("O metal está em meio aquoso? (sim/não): ").strip().lower() == 'sim'
```

O programa: Programa Principal

```
# Entradas para a análise da água
temperatura_input = float(input("Digite a temperatura da água (em °C): "))
STD_input = float(input("Digite o STD da água: "))
Ca_CaCO3_input = float(input("Digite a concentração de Ca2+ de CaCO3 (em mg/L): "))
alcalinidade_CaCO3_input = float(input("Digite a alcalinidade do CaCO3 (em mg/L): "))
pH_real_input = float(input("Digite o pH real da água: "))

# Instanciando a classe Metal
metal_obj = Metal(metal_input, corrente_input, area_input, aquoso_input)

# Calculando a taxa de corrosão
taxa_de_corrosao = metal_obj.calcular_taxa_corrosao()

if taxa_de_corrosao is not None:
    nivel_corrosao = metal_obj.classificar_corrosao(taxa_de_corrosao)
    meio = 'meio aquoso' if aquoso_input else 'meio não aquoso'
    print(f"A taxa de corrosão para {metal_input} em {meio} é {taxa_de_corrosao:.6e} g/s por cm^2.")
    print(f"O nível de corrosão é {nivel_corrosao}.")
```

O programa: Programa Principal

```
# Calculando pHS e IER
pHS = Agua.calcular_pHS(temperatura_input, STD_input, Ca_CaCO3_input, alcalinidade_CaCO3_input)
IER = Agua.calcular_IER(pHS, pH_real_input)
classificacao_agua = Agua.classificar_agua(IER)

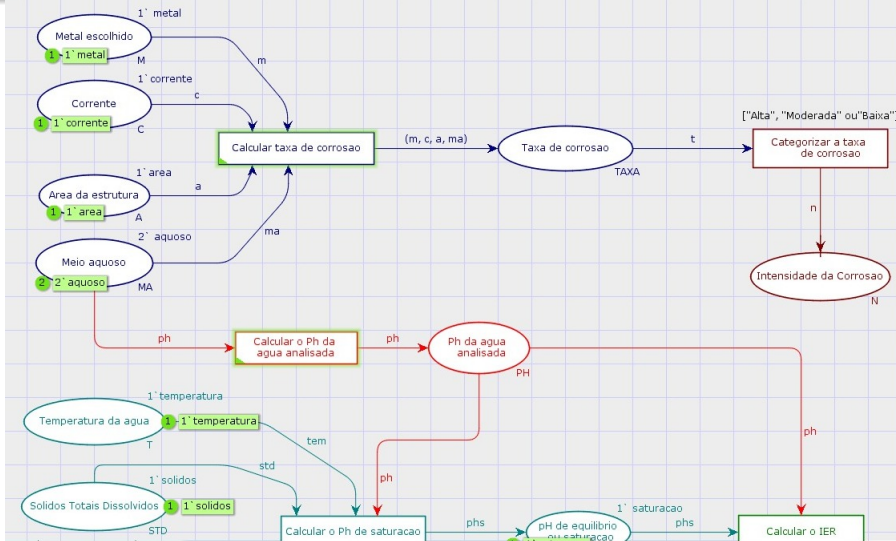
print(f"O pH de saturação (pHS) é: {pHS:.2f}")
print(f"O Índice de Estabilidade de Ryznar (IER) é: {IER:.2f}")
print(f"A classificação da água é: {classificacao_agua}")
```

Sumário

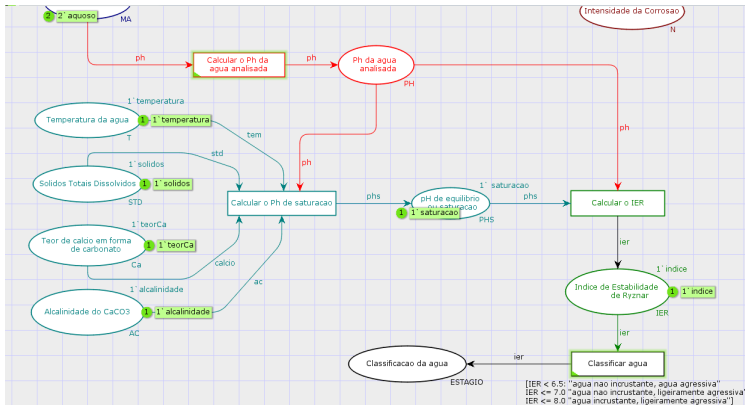
- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

Introdução
Proposta do programa
Descrição narrativa
O programa:
Rede petri
Rede petri
considerações finais
Referências

Introdução
Proposta do programa
Descrição narrativa
O programa:
Rede petri
Rede petri
considerações finais
Referências



Rede petri



Rede de Petri

- **Calcular Taxa de Corrosão:** Requer os valores de M, C, A e MA.
Saída: Taxa de corrosão (TAXA).
- **Categorizar a Taxa de Corrosão:** Entrada: Taxa de corrosão (t).
Saída: Intensidade da Corrosão (N).
- **Calcular o pH da Água Analisada:** Entrada: Meio aquoso (MA).
Saída: pH da água analisada (PH).
- **Calcular o pH de Saturação:** Requer os valores de T, STD, Ca e AC e do pH da água analisada. Saída: pH de equilíbrio ou saturação (PHS).

Rede de Petri

- **Calcular o Índice de Estabilidade de Ryznar (IER):** Requer pH da água analisada (PH) e pH de saturação (PHS). Saída: Índice de Estabilidade de Ryznar (IER).
- **Classificar Água:** Entrada: Índice de Estabilidade de Ryznar (IER). Saída: Classificação da água (ESTÁGIO).

Como a Rede de Petri contemplou o projeto?

- A CPN ajudou a estruturar o programa de uma maneira clara e visual. Cada lugar e transição na rede representa um estado ou evento significativo do programa. Isso facilita a visualização do fluxo de dados e operações, tornando o sistema mais intuitivo e compreensível.
- Além disso, foi possível realizar análises formais e simulações para verificar e validar o comportamento do sistema. Através da simulação, podemos observar como os tokens se movem pela rede, verificando se os cálculos e classificações ocorrem conforme o esperado. Isso é útil para detectar e corrigir erros lógicos no programa, garantindo que todas as condições e transições sejam corretamente implementadas.

Sumário

- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

Considerações finais

- A partir da pesquisa realizada, é notória a importância de calcular as taxas de corrosão. Visto que, ao calcular essa taxa para diferentes metais em diferentes ambientes, ele permite uma melhor seleção de materiais, planejamento de manutenção, proteção ambiental e, acima de tudo, a preservação da integridade estrutural, essencial para a segurança pública. Além disso, a análise do potencial da água para causar corrosão ou incrustação, se mostra útil, por exemplo, para o gerenciamento de sistemas de abastecimento de água e esgoto. Logo, engenheiros civis podem usar esses dados para escolher metais que ofereçam maior resistência à corrosão. Isso ajuda a otimizar os recursos disponíveis e a reduzir custos associados à manutenção corretiva.

Sumário

- 1 Introdução
- 2 Proposta do programa
- 3 Descrição narrativa
- 4 O programa:
- 5 Rede petri
- 6 Rede petri

Referências

- LTC Gentil, V. editora, 5ª edição. Rio de Janeiro, 2007.
- William Callister. Ciência E Engenharia de Materiais: Uma Introdução. Grupo Gen-LTC, 2000.
- Stephan Wolyneec. Técnicas eletroquímicas em corrosão Vol. 49. Edusp, 2003.
- Nilo Ney Coutinho Menezes. Introdução à programação com Python. São Paulo: Novatec, página 34, 2010.
- Felipe Cruz. Python: Escreva seus primeiros programas. Editora Casa do Código, 2015.