

# ControlPowerFlow.jl

## Análise de Fluxo de Potência com Ações de Controle

**Baseado no relatório de:**

Iago Sichinel Chávarry

# 1. Motivação e Contexto

- **Cenário:** O Sistema Interligado Nacional (SIN) possui dimensão continental e desafios de estabilidade de tensão.
- **O Problema:** Softwares Open-Source tradicionais resolvem o fluxo de potência "rígido" ( $f(x) = 0$ ). Se o ponto de operação for difícil, eles frequentemente não convergem.
- **A Solução Comercial (ANAREDE):** Utiliza controles (FACTS, LTCs, Shunts) para ajustar o sistema e garantir convergência.
- **O Objetivo da Biblioteca:** Trazer essa capacidade de **ações de controle** para o ambiente Julia/Open-Source.

## 2. O Que é o ControlPowerFlow.jl?

É um pacote em Julia desenvolvido para realizar análises de fluxo de carga considerando ajustes automáticos da rede.

- **Fundação:** Construído sobre o **PowerModels.jl** e **JuMP**.
- **Abordagem Matemática:** Transforma o problema de equações algébricas em um problema de **otimização**.
  - Ao invés de falhar quando um limite é atingido, o solver penaliza desvios na função objetivo.
- **Integração:** Funciona em conjunto com o **PWF.jl** (leitor de arquivos ANAREDE).

### 3. Como a Biblioteca Implementa os Controles?

A biblioteca modifica a formulação padrão do PowerModels de três formas principais:

1. **Novas Variáveis de Decisão:** Transforma parâmetros fixos (ex: susceptância shunt) em variáveis.
2. **Novas Restrições:** Adiciona limites físicos (ex: limite de reativo).
3. **Variáveis de Folga (Slack):** Insere variáveis de folga em restrições rígidas e adiciona uma penalidade quadrática na função objetivo para mantê-las próximas ao valor nominal.

## 4. Funcionalidades Implementadas (O "Menu")

Inspirado nas opções do ANAREDE (campo `DOPC`), o pacote implementa:

- **QLIM:** Limites de Geração de Potência Reativa.
- **VLIM:** Limites de Magnitude de Tensão (barras PQ).
- **CSCA:** Controle Automático de Reator Shunt (Contínuo e Discreto).

*(Outros citados na estrutura: CTAP, CTAF, CPHS)*

## 5. Detalhe: QLIM (Reactive Generation Limits)

O que acontece quando uma barra PV não consegue manter a tensão devido ao limite de geração de reativos ( $Q_{lim}$ )?

- **Implementação:**
  - A tensão especificada ( $v_{spec}$ ) deixa de ser fixa.
  - Insere-se uma variável de folga  $sl$  na equação:  $v = v_{spec} + sl$ .
  - Minimiza-se  $sl^2$  na função objetivo.
- **Resultado:** O modelo mantém a tensão fixa, a menos que viole os limites de  $Q$ , momento em que ele ajusta a tensão o mínimo necessário.

## 6. Detalhe: CSCA (Shunt Reactor Control)

Controla bancos de capacitores/reatores para manter a tensão.

- **Modos de Operação:**
  1. **Contínuo:** A susceptância ( $b^{sh}$ ) varia livremente dentro dos limites para fixar a tensão.
  2. **Discreto:** O modelo chaveia bancos para manter a tensão dentro de uma faixa ( $V_{min} \leq V \leq V_{max}$ ).
- **Aproximação:** Nesta versão, o modo discreto é aproximado como variáveis contínuas relaxadas para facilitar a convergência do solver.

## 7. Exemplo de Uso (Código)

O fluxo de trabalho é simples e integrado ao ecossistema Julia:

```
using PWF, ControlPowerFlow, Ipopt

# 1. Definir caminho do arquivo ANAREDE
file = "Example.PWF"

# 2. Ler arquivo e ativar leitura de dados de controle
# (Lê seções como DOPC e limites de shunts)
pwf_data = PWF.parse_file(file; add_control_data = true)

# 3. Executar o fluxo de potência com controle
# O solver Ipopt lida com a otimização não-linear
results = run_control_pf(pwf_data, optimizer = Ipopt.Optimizer)
```