

杭の施工誤差のパターン検討

本プログラムの説明

本プログラムは、強化学習手法を用いて、杭の施工誤差について最も危険側となるパターンを求めるツールです。従来の施工誤差の検討では、寄り方向のパターン1つで解析・検定を行っていましたが、本プログラムでは、基礎ごとに8方向9パターンの組み合わせを試行回数分計算するなかで、検定比が最大となるパターンを学習しながら効率よく選定することを試みたものです。

実際に基礎の寄りを与えて解析するため、同時に生じえないパターンでは検討しません。

本プログラムの実行方法

1. データ準備

KUI_sekogosa.py をエディタで開き、12行目のパスをSS7物件データのパス名に設定します。

```
# 対象物件のpath指定(必須)
str_path = r"D:\UsrData\Ss7Data\AI_Kui\kensyou.ikn"
```

14行目で施工誤差 (mm) を指定します。

```
# 施工誤差 任意の誤差を指定(適宜)
er = 100
```

70～78行のハイパーパラメータなどを適宜調整します。

```
# ハイパーパラメータ
learning_rate = 0.05 # 学習率。Q値を更新する際の重み付けに使用されます。
discount_factor = 0.9 # 割引率。将来の報酬をどれだけ重視するかを調整します。
exploration_prob = 0.2 # ランダムな行動を取る確率を決定します。
num_episodes = 500 # 試行回数
target_reward = 10000 # 終了条件となる報酬の値
target_reward_Max = 2 # 終了条件を満たした回数
target_reward_count = 0 # 終了条件を満たした報酬のカウンター
action_at_target_reward = None # 報酬が指定値となったときの行動
```

サブプログラムの設定

ss7_calc.py をエディタで開き、15～17行目で考慮する符号を定義します。

```
# 検定比を考慮する基礎梁符号
beam_codes = ["FG1", "FG2", "FG3", "FG11", "FG12", "FG13"]
found_codes = ["F1", "F2", "F3"]
picket_codes = ["P1", "P2"]
```

19行目で重み付けを指定します。

```
# 検定比の重み付け
weight_beam=1.0 # 基礎梁
weight_found=1.0 # 基礎
weight_picket=1.0 # 杭
```

25～27行目で検定比の項目数を定義し、67行目、127行目以降で結果CSVファイルの名称と項目数を合わせます（基礎梁ハンチの有無や杭種、計算条件等により項目数が変わります）。

```
# 検定比を保持するリスト
prev_beam_ratio = [None] * (len(beam_codes) * 14) # 基礎梁の検定比14項目
prev_found_ratio = [None] * (len(found_codes) * 5) # 基礎の検定比5項目
prev_picket_ratio = [None] * (len(picket_codes) * 4) # 杭の検定比4項目
```

```
67行目
# 出力項目名
output_name = "7.1.1.RC梁;8.9.5.検定比一覧（杭基礎）;8.10.2.検定比一覧（既製杭）"
～省略～
127行目
table = csv.search_table_case("RC梁検定比一覧", "", "")
if table:
    col = table.search_col3("M", "(上端/下端)", "左端")
    col = int(col)
    for beam_code in beam_codes:
        row = int(table.get_key_row2(bottom, beam_code, col))
        current_ratio.extend(float(table.get(row, col + i)) if table.get(row, col + i) not
            in (None, "") else 0 for i in range(9))
        row += 1
        current_ratio2.extend(float(table.get(row, col + i)) if table.get(row, col + i) not
            in (None, "") else 0 for i in range(5))

table = csv.search_table_case("杭基礎検定比一覧", "", "")

if table:
    col = table.search_col2("曲げ", "x")
    col = int(col)
    for found_code in found_codes:
        row = int(table.get_key_row1(found_code, col))
        current_ratio2.extend(float(table.get(row, col + i)) if table.get(row, col + i) not
            in (None, "") else 0 for i in range(5))

table = csv.search_table_case("既製杭検定比一覧", "", "")
if table:
    col = table.search_col2("曲げ", "上杭")
    col = int(col)
    for picket_code in picket_codes:
        row = int(table.get_key_row1(picket_code, col))
        current_ratio3.extend(float(table.get(row, col + i)) if table.get(row, col + i) not
            in (None, "") else 0 for i in range(4))
```

2. プログラムの実行

コマンドラインからプログラムを実行します。

```
例) C:\example\srcにソースコードがある場合
cd C:\example\src
python KUI_sekogosa.py
```

3.プログラム終了

試行回数または終了条件に達すると、検定比が最大となった施工誤差パターンの物件データを生成して断面算定まで計算し、報酬の図をプロットして、プログラムが終了します。

注意事項

試行回数によっては、数時間から十数時間を要する場合があります。

[13.5.基礎の寄り] の入力データを書き換えて検討するため、データが無い場合はエラーとなり実行できません。すべての基礎あるいは施工誤差が考えられる基礎について [13.5.基礎の寄り] のデータ（例えば、押さえ位置55、X,Yともに0のデータ）をあらかじめ入力してください。

本来は、検討事項ごとに各部材のそれぞれについての危険なパターンが、複数存在すると思われます。このツールを用いることで、検定比がより大きくなるパターンが選定されますが、危険側となる可能性のひとつとしてご理解くださいますようお願いいたします。

『Op.Python実行』の設定手順

Ss7Pythonライブラリを使用するための設定手順です。

1. 『SS7』を起動し、[ツール - 環境設定 - Op.Python実行]画面を表示します。
2. “利用可能なPython言語のバージョン”を選択し、[デスクトップへコピー]ボタンをクリックします。
3. デスクトップにある「Python」フォルダごと、「src」フォルダにコピーします。

必要な外部ライブラリ

本プログラムは以下のPythonライブラリを使用します。

```
pip install numpy
pip install matplotlib
```

ss7_csv_input.py,ss7_csv_result.pyは、KUI_sekogosa.pyと同じフォルダに置いてください。
外部ライブラリのライセンスは「LICENSES/ライセンスについて.txt」を確認してください。

著作者

Copyright (C) 2025 UNION SYSTEM Inc.

ライセンス

本プログラムは MIT License に基づいています。「LICENSE」を確認してください。