CHIYODA CORPORATION

CPW CHIYODA & PUBLIC WORKS Co., Ltd.

Specification of
LNG Plant Optimum Parameter Search

1. Overview

1.1 Background and Objective

- In 2019 Chiyoda succeeded in developing the "LNG Plant AI Optimizer™".
  For the PT Donggi-Senoro LNG plant in Indonesia Chiyoda confirmed increased LNG production through improved efficiency using the "LNG Plant AI Optimizer™".
- However, the optimal condition search process takes time, and it is necessary to prepare an environment for calculation for that process.
- In order to solve this problem, the purpose of this study is to create the regression model which calculate the optimum operating parameters directly.
  In order to create the regression model, comprehensive data set is created including various external conditions and the optimum operating parameters.

1.2 AI development policy

- The purpose of this study is to create the regression model which calculate the optimum operating parameters directly.
- The scope is Natural Gas Liquefaction and Refrigeration System of LNG Plant.
- The original dataset is from traced simulation.

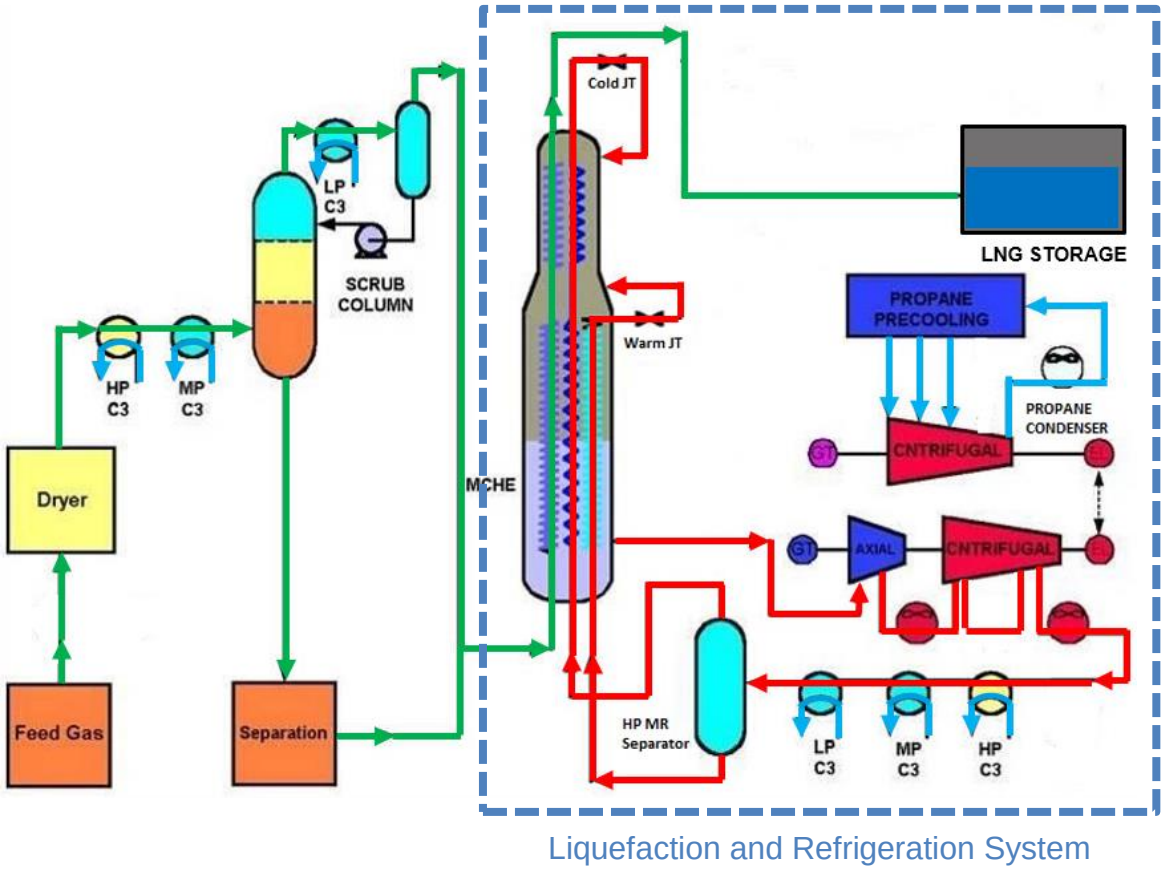
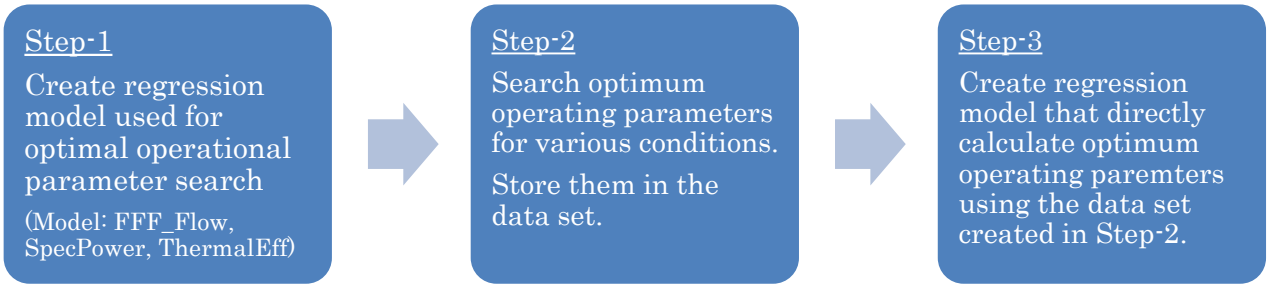


Fig. 1-1: Natural Gas Liquefaction Process

1.3 Overall Development Flow

- Create the final regression model according to the following flow.
- As Step-0, CGH provide the simulation result dataset used for Step-1.



2. Details of Developments

2.1 Step-1 Create Regression Model for Optimum parameter Search

2.1.1 Overall Description

- Using the dataset provided by CGH (conditions.csv), create a regression models required for the step-2.
  - Data Set Tags:
    FEED_N2, FEED_C1, FEED_C2, FEED_C3, FEED_iC4, FEED_nC4, FEED_C5+, AmbTemp, C3_CON_OUT, P_HPMR, P_LPMR, MCHE_BTM_DT, WB_UA, CB_UA, MR_N2, MR_C1, MR_C2, MR_C3, LMR_GASFLOW, FFF_Flow, SpecPower, ThermalEff, MCHE_OUT_T, MCHE_OUT_C_T, MCHE_MID_T, MCHE_MID_C_T, MCHE_IN_T, MCHE_IN_C_T, MR_GASFLOW, HMR_VFLOW, C3_GT_Power, LPMR_GT_Power, HPMR_GT_Power, LNG_Dens, Feasible, Note
  - Regression Models (Target Value): FFF_Flow, SpecPower, ThermalEff
  - Data Set File Location:
    https://github.com/efexis-com/regression_dslng/blob/master/40_Optimum_Parameter_Search/Dataset/conditions.csv

2.1.2 Coding

- Data Visualization and Preprocessing
  - Exclude the data which has 0 in “Feasible” Tag.
  - Visualize original dataset tags by histogram plot. (Fig. 2-1)
  - If the accuracy of the regression model is not good, consider excluding outliers based on the visualization results.

```
num_row = math.ceil(len(df.columns) / 3)

plt.figure(figsize=(15, 5 * num_row))
i = 0

for tag in df.columns:
    i += 1

    plt.subplot(num_row, 3, i)
    plt.hist(x=df[tag], bins=200)
    plt.grid()
    plt.title(label=tag)

plt.show()
```

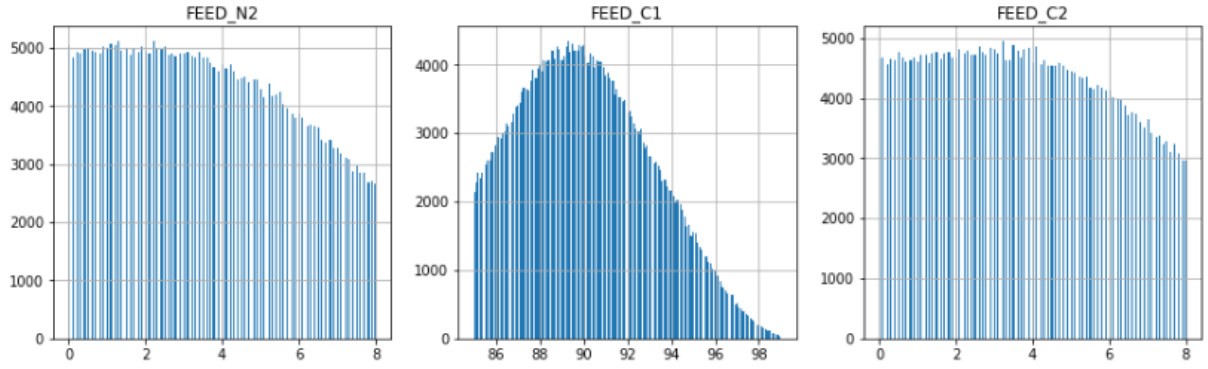


Fig. 2-1

◆ Model Input and Output

➢ Input values are list_X (list_u + list_v) as described below.
list_u is an external condition and list_v is an operable parameter.
(list_v values will be searched target in step-2.)

```
list_u = [
        'FEED_N2', 'FEED_C1', 'FEED_C2', 'FEED_C3',
        # 'FEED_iC4', # 'FEED_nC4', # 'FEED_C5+',
        'AmbTemp', 'C3_CON_OUT', # 'MCHE_IN_T',
        'WB_UA', 'CB_UA',
    ]
    # explanatory variable - search
list_v = [
        'P_HPMR', 'P_LPMR', 'MCHE_BTM_DT', # 'MR_N2',
        'MR_C1', 'MR_C2', 'MR_C3',
    ]
    list_X = list_u + list_v
```

- Output values
  FFF_Flow, SpecPower, ThermalEff, MCHE_OUT_T
  (Create each dedicated model. 4 models will be created.)

◆ Hyper Parameter Tuning

- Consider whether you can reduce the parameters while maintaining accuracy.
- Parameters:
  Number of intermediate layers: (1), 2, 3, 4
  Activation functions: ReLU, Tanh
  Learning rate: 0.001
  Batch size: 128
  Numbers of units:
  1st layer: 5, 10, 25 / 2nd layer: 0, 5, 10 / 3rd layer: 0, 5, 10 / 4th layer: 0, 5, 10

2.1.3 Output

◆ Regression Models (with optimized hyper paramters) and Scaler Files:
FFF_Flow, SpecPower, ThermalEff, MCHE_OUT_T

◆ Folder:

https://github.com/efexis-com/regression_dslng/tree/master/40_Optimum_Parameter_Search/Step1

2.2 Step-2 Search optimum operating parameters for various conditions

2.2.1 Overall Description

- Search optimum operating parameters for various conditions and Store them in the data set.
- Optimum operable parameters (list_v) are searched based on the external condition parameters (list_u) in the dataset (conditions.csv). The thermal efficiency (ThermalEff) and fuel from feed flow (FFF_Flow) of each combination of various operable parameters is calculated via regression models, and select the best combination of operable parameters based on the regression results (Fig. 2-2). The criteria for the selection is described in section 2.2.2.

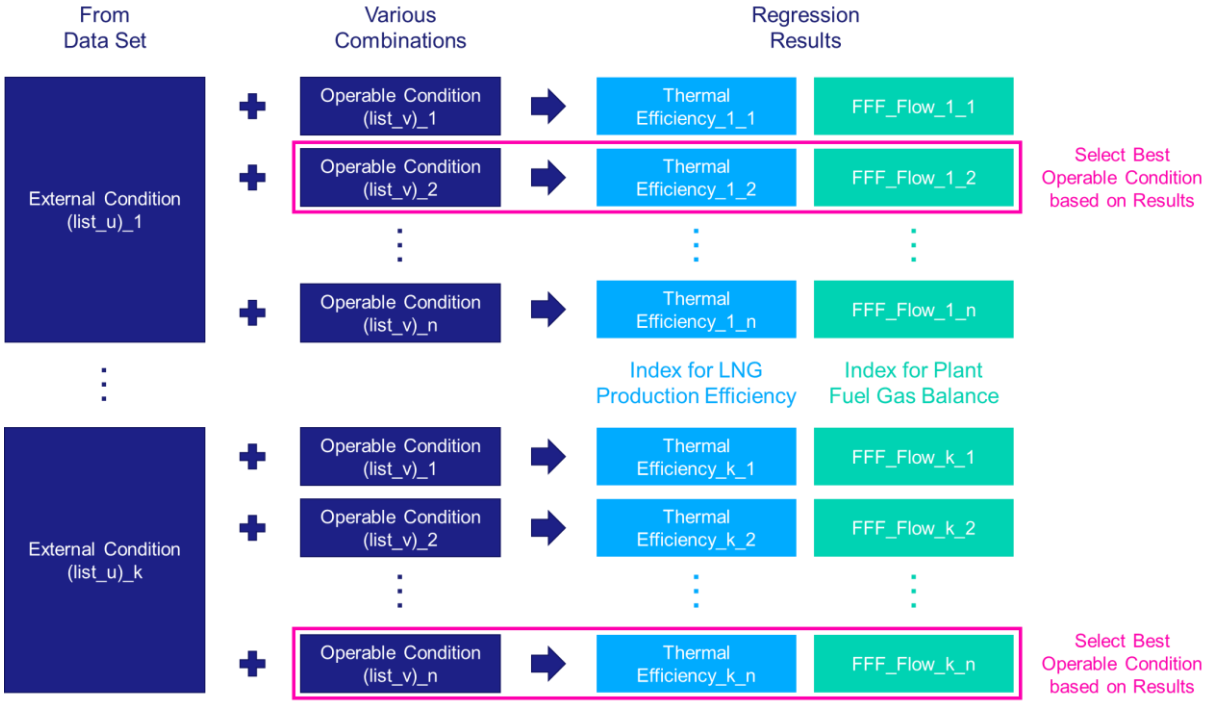


Fig. 2-2: Image of Optimum (Best) Parameter Search

2.2.2 Coding

- General:
  - Create dedicated program file (source code) for Step2.
  - Save the file in Step2 Folder.
  - Reference Code:

    https://github.com/efexis-com/regression_dslng/blob/master/40_Optimum_Parameter_Search/Step2/Reference/mrsearch.py

- Model and Scaler:
  - Read Model and Scaler Files saved in Step1 folder.
- Regression Model Input Values:
  - External Conditions (list_u):
    - ✓ Combinations are based on Data Set (conditions.csv) with avoiding duplication. (e.g. df = df.drop_duplicates())
    - ✓ For 'WB_UA' and 'CB_UA' (Main Cryogenic Heat Exchanger UA), the values should be fixed to their mean value.
  - Operable Conditions (list_v): Refer Table 2-1 for the combinations.
    (For reference, please show (print()) the number of the combinations in the code.)

Table 2-1: Combination of Operable Conditions (list_v)

Input Tags	Min	Max	Step (Note 1)	Note
[P_HPMR, P_LPMR]	[52.4, 1.81]	[59.4, 2.18]	Note 2	Note 2
MCHE_BTM_DT	1.0	7.5	0.5	
MR_N2	5	7	0.1	Note 3
MR_C1	38	43	0.1	Note 3
MR_C2	38	44	0.1	Note 3
MR_C3	8	14	0.1	Note 3

Note:

1) Step can be reduced considering the calculation time.
2) Combination of [P_HPMR, LPMR]:
   [5.24e+01, 1.81e+00], [5.54e+01, 1.97e+00], [5.74e+01, 2.08e+00], [5.94e+01, 2.18e+00]
3) Consider that MR composition should totally become 100%.
   Fig. 2-3 is an image of coding considering this basis. Since it was used for other purposes, it cannot be used as it is. (e.g. N2 is not required as input value for this study.)

```
composition = []
n2_range = range(int(2.0*100), int(20.0*100), int(0.05*100))
c1_range = range(int(25.0*100), int(50.0*100), int(0.1*100))
c2_range = range(int(25.0*100), int(50.0*100), int(0.1*100))
c3_range = range(int(5.0*100), int(30.0*100), int(0.1*100))
for n2 in n2_range:
    for c1 in c1_range:
        for c2 in c2_range:
            c3 = 100 * 100 - (n2 + c1 + c2)
            if c3 in c3_range:
                composition.append([n2/100, c1/100, c2/100, c3/100])
```

Fig. 2-3: Image of MR Combination Creation

- Regression Model Output Values and Optimum Value Selection Criteria:
  - Fuel from FEED Flow (FFF_Flow): Should be higher than 0.

```
rs = rs[rs['FFF_Flow_1'] > 0]
```

  - Thermal Efficiency (ThermalEff): Higher is better. Please sort the condition by Thermal Efficiency and extract the top 0.01% (as best1) and 0.02 (as best2) data.

```
max_thermal = rs['ThermalEff_1'].max()
rs_best1 = rs[rs['ThermalEff_1'] > max_thermal * 0.9999]
rs_best2 = rs[rs['ThermalEff_1'] > max_thermal * 0.9998]
```

  - Specific Power (SpecPower): Not used as a criteria, but should be calculated for validation.
  - MCHE_OUT_T (Outlet Temp): Not used as a criteria, but should be calculated for validation.

- Optimum Operating Condition:
  - Calculate the minimum, maximum, mean, and standard deviation (std) value of the top 0.01% (best1) and 0.02%(best2) of Thermal Efficiency, where FFF_Flow is higher than 0.
  - Add optimum values (list_v and regressed values (e.g. “TheralEff”, “SpecPower”)) as “Best” value (e.g. Best_MR_N2) into the Data Set (e.g. df_training in Fig.2-4).

```
        # write result
        for col in list_result:
            df_training.loc[row, 'Best1_' + col + '_MIN'] = rs_best1[col].min()
            df_training.loc[row, 'Best1_' + col + '_MAX'] = rs_best1[col].max()
            df_training.loc[row, 'Best1_' + col + '_MEAN'] = rs_best1[col].mean()
            df_training.loc[row, 'Best2_' + col + '_MIN'] = rs_best2[col].min()
            df_training.loc[row, 'Best2_' + col + '_MAX'] = rs_best2[col].max()
            df_training.loc[row, 'Best2_' + col + '_MEAN'] = rs_best2[col].mean()
```

Fig. 2-4: Image of writing the result into original df (df_training).

2.2.3 Output

- Data set including optimum (best) operable values.

- Folder:

  https://github.com/efexis-com/regression_dslng/tree/master/40_Optimum_Parameter_Search/Step2

2.3 Step-3 Create regression model that calculate optimum parameters.

2.3.1 Overall Description

- Using the dataset created in Step-2, create a regression models of best parameters (mean values).

  - Data Set Tags:
    Original Datasets (FEED_N2, FEED_C1, FEED_C2, FEED_C3, FEED_iC4, FEED_nC4, FEED_C5+, AmbTemp, C3_CON_OUT, P_HPMR, P_LPMR, MCHE_BTM_DT, WB_UA, CB_UA, MR_N2, MR_C1, MR_C2, MR_C3, LMR_GASFLOW, FFF_Flow, SpecPower, ThermalEff, MCHE_OUT_T, MCHE_OUT_C_T, MCHE_MID_T, MCHE_MID_C_T, MCHE_IN_T, MCHE_IN_C_T, MR_GASFLOW, HMR_VFLOW, C3_GT_Power, LPMR_GT_Power, HPMR_GT_Power, LNG_Dens, Feasible, Note)
    + Best Parameters
  - Regression Models (Target Value):
    Best Parameters (e.g. Best_MR_N2, Best_ThermalEff)

2.3.2 Coding

- Same process as 2.1.2.
  Target Values are only changed.

3. Output

3.1 Deliverables

- Program (or Source Code)
  - Python (3.8), Tensor Flow (2.3) with Keras
- Description of “how to use the program, how to install it, the libraries you use, etc. (You may include comments in the source code.)”
  ※ Not Required in case of using the typical libraries in the program.
- Report (Includes empirical results, a brief description of the algorithm used, and problems)

3.2 Scheduled Date for Deliverables

- Service Agreement Schedule (10th of March, 2021)