**VI**

**1. 수소 생산 공정 운전 최적화**

1. **공정 설명**

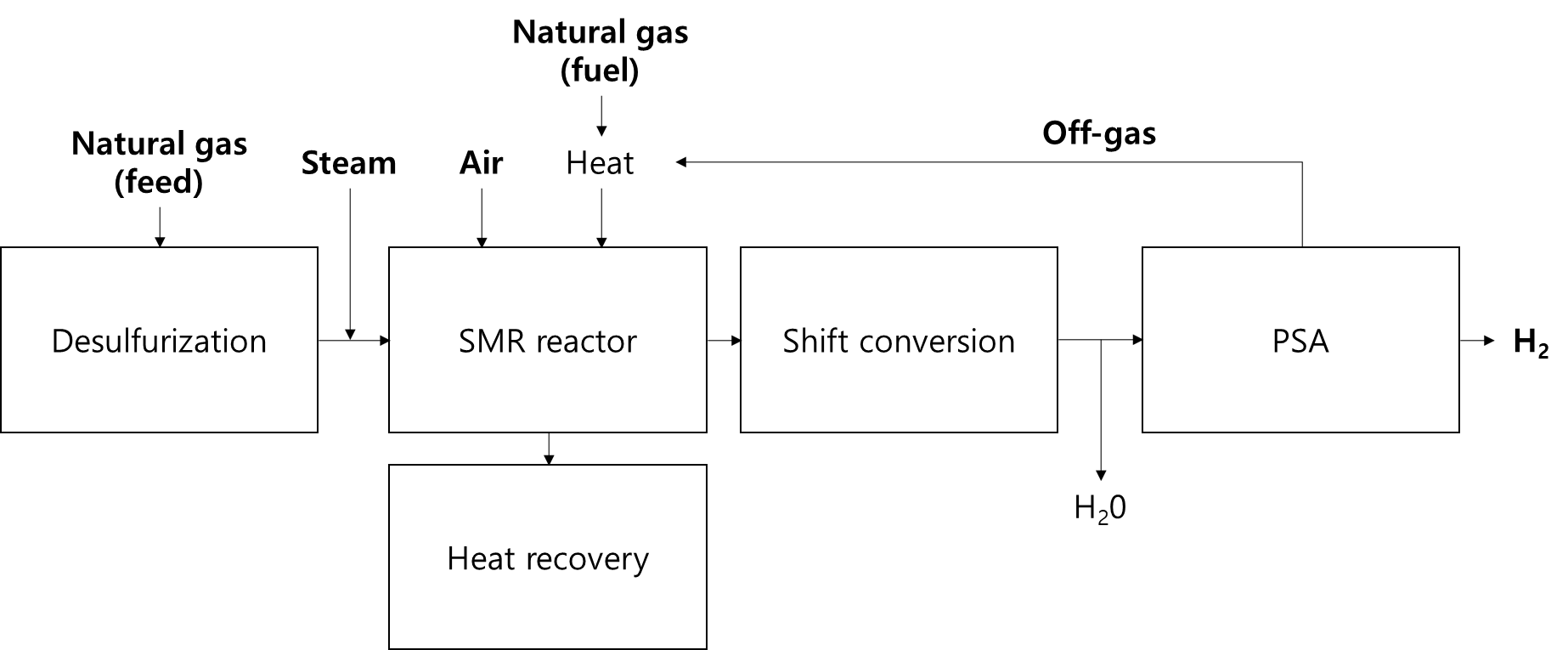


그림 1. 수증기 개질 공정의 흐름도

수증기 개질(steam methane reforming, SMR) 공정은 크게 주요 반응이 이루어지는 SMR 반응기, 황성분을 제거하는 탈황장치(hydrodesulfurization, HDS), 잔존 CO를 H2로 전환하는 전환반응기(shift reactor), 생성된 합성가스로부터 고순도의 수소를 분리하는 압력 스윙 흡착(pressure swing adsorption, PSA) 장치, 열회수를 위한 열교환기(heat exchanger)로 이루어져 있다 (그림 1).

원료인 천연가스는 한국 전역에 설치되어 있는 천연가스배관 인프라를 통하여 공급된다. 천연가스는 C1~C5의 혼합물로 구성되어 있으며 주성분은 CH4로 약 93.1% 차지한다. 천연가스는 일부 황을 포함하고 있는데, 황 성분은 촉매독 역할을 하여 촉매에 큰 악영향을 가하므로 원료 투입 전에 제거 되어야한다. 원료인 천연가스는 약 300℃로 예열되어 CoMo와 ZnO 촉매로 충전된 탈황장치로 투입되고 황 성분이 제거된다. 또한, 각종 불순물이 제거된 물은 특수 제작된 열교환기를 통하여 180℃ 이상으로 승온 되어 과열 증기상태로 공급된다. 결과적으로 황 성분이 제거된 천연가스와 과열 스팀이 섞인 후 약 580℃ 조건으로 승온되어 SMR 반응기로 투입된다. SMR 반응기에서 아래의 세가지 주요 반응이 진행되며 수소가 생성된다. 전체적인 SMR 반응은 강한 흡열반응이며 약 700~800℃, 그리고 8barg 조건에서 반응이 일어난다.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |

반응 종료된 합성가스는 CO, CO2, CH4, H2, H2O로 이루어져 있으며 약 750℃ 조건으로 SMR 반응기를 빠져나와 후단 공정에 열을 전달한다. 이 합성가스는 180–250℃ 조건으로 저온수성가스화(low temperature shift, LTS) 반응기로 투입된다. LTS 반응기에서는 합성가스에 잔존하고 있는 CO를 제거하는 반응이 일어난다. CO는 수소 연료전지에 악영향을 미치기 때문에 최대한 제거하는 것이 좋다. 또한 LTS 반응기에서 CO가 제거되며 추가적인 H2를 생성하는 장점이 있다. LTS 반응을 거치고 나온 흐름은 40℃까지 온도가 낮춰지고, 물이 제거된 합성가스를 순도 99.999%, CO농도 0.2ppm 이하의 스펙을 갖는 수소로 분리하기 위해서 PSA 장치에 투입된다.

**(문제)**

**투입되는 천연가스, 스팀, 공기의 유량과 PSA 장치의 회수율에 따라서 최종적으로 생성되는 수소의 유량을 계산하는 인공신경망 모델을 구축하고, 구축된 모델을 활용하여 SMR 공정의 운전조건을 최적화하라.**

**Q1. 데이터를 작업 환경으로 불러오자. 데이터의 불러오기와 저장하기 등 데이터 작업에 유용한 파이썬 라이브러리는 무엇인가?**

A1. 일반적으로 데이터는 Excel과 같은 관계형 데이터베이스에 많이 들어있다. 데이터는 pandas 라이브러리를 이용하여 다음 명령어를 통해 읽어 들인다.

|  |
| --- |
| import pandas as pd  data\_smr = pd.read\_csv(“./smr\_process.csv”, header=0) |

pd.read\_csv 함수는 해당경로의 데이터를 불러와 데이터프레임 형태로 저장한다. 여기서 “header=0”은 첫 행을 열 이름으로 지정한다는 의미이다.

**Q2. 데이터의 기본 정보를 파악하여라. 데이터 샘플 개수, 특성 개수, 각 특성의 타입, 최솟값, 최댓값, 평균값은 어떻게 되는가?**

A2. head() 사용하여 불러온 데이터의 정보를 확인해보자.

|  |
| --- |
| data\_smr.head() |

테이블이(가) 표시된 사진

자동 생성된 설명

각 행은 하나의 데이터 샘플을 나타낸다. head() 괄호 안에 숫자를 입력하면 처음부터 그 숫자만큼의 행을 보여주고, 기본 값은 5개로 설정 되어있다.

info()는 데이터에 대한 간단한 설명을 나타낸다. 데이터 샘플의 수, 각 특성의 데이터 타입, 결측치 개수 등의 정보를 파악할 수 있다.

|  |
| --- |
| data\_smr.info() |

텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명

describe()는 숫자형 데이터의 통계 정보를 보여준다.

|  |
| --- |
| data\_smr.describe() |

테이블이(가) 표시된 사진

자동 생성된 설명

count는 결측치를 제외한 데이터 샘플의 개수, mean, min, max는 각 데이터 특성의 평균값, 최솟값, 최댓값을 나타낸다. std 행은 값이 퍼져 있는 정도를 측정하는 표준편차이다. 25%, 50%, 75% 행은 백분위수를 나타낸다.

**Q3. 신경망 모델의 학습과 평가를 위해 데이터를 분리하여라. 학습 데이터와 평가 데이터의 차이점은 무엇인가?**

A3. 모델 훈련과 평가를 진행하기 위해서 사이킷런의 train\_test\_split 함수를 이용하여 훈련 데이터와 평가 데이터를 분리한다. 학습 데이터는 모델 학습을 할 때 사용하는 데이터이며, 평가 데이터는 학습된 모델의 성능을 평가하는 데이터이다. 평가 데이터는 모델의 일반화 오류 정도를 평가하기 위해 마지막에 한번만 사용되며, 이를 통해 모델의 과적합을 방지할 수 있다.

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  test\_ratio = 0.3  random\_state = 42  data\_train, data\_test = train\_test\_split(data\_modeling, test\_size = test\_ratio,  shuffle=True, random\_state=random\_state) |

test\_ratio 값은 데이터 샘플들로부터 테스트 데이터셋을 얼마만큼 분리할 지를 결정한다. 여기서는 전체의 30% 만큼 테스트 데이터셋으로 분리하였다. random\_state는 랜덤으로 분리되는 데이터를 다음 시행에서도 동일하게 분리하기 위해(동일한 실험 결과를 얻기 위해서) 고정하는 값으로, 42로 설정되었다.

**Q4. 모델의 입력변수와 출력변수를 정의하고, 스케일러를 이용하여 데이터의 값을 조정하여라. 훈련데이터와 평가데이터의 스케일링 시 주의사항은 무엇인가?**

A4. 모델이 데이터의 정보를 효율적으로 파악하고 학습할 수 있도록 모든 공정 변수의 값을 표준화하는 데이터 스케일링을 진행한다. 데이터 스케일링을 진행하기에 앞서, 스케일러 용이성을 위하여 모델의 입력변수와 출력변수를 구분하여 스케일링을 진행한다.

|  |
| --- |
| from sklearn.preprocessing import StandardScaler  var\_x = ["NG feed”, "NG fuel”, "Water”, "Air”, "PSA recovery”]  var\_y = ["H2”]  scaler\_x = StandardScaler()  scaler\_y = StandardScaler()  train\_x = scaler\_x.fit\_transform(data\_train[var\_x])  train\_y = scaler\_y.fit\_transform(data\_train[var\_y])  test\_x = scaler\_x.transform(data\_test[var\_x])  test\_y = scaler\_y.transform(data\_test[var\_y]) |

모델의 입력변수는 시스템에 주입되는 천연가스의 원료와 연료, 공기와 물의 유량, PSA 장치에서의 회수율로 설정되었고, 출력변수는 시스템에서 빠져나가는 합성가스의 유량과 그 조성, 그리고 각 장치의 온도로 설정되었다. 스케일러는 표준화를 위한 StandardScaler가 사용되었으며, 모델 훈련 시 테스트 데이터셋의 정보가 반영되지 않도록 훈련 데이터셋에 맞추어 스케일링을 진행한다.

**Q5. 공정 모델링을 위한 인공신경망을 구축하여라. 인공신경망 구성 시 필요한 요소들은 무엇인가?**

A5. 정제된 데이터를 이용하여 SMR 공정을 모델링하기 위해서는 데이터 학습을 위한 인공신경망 구축이 필요하다. 인공신경망 함수는 다음과 같이 정의된다.

|  |
| --- |
| from tensorflow import keras  from tensorflow.keras.layers import \*  def NeuralNet(num\_x, num\_y, num\_layers, num\_neurons, learning\_rate):  model = keras.Sequential()  model.add(Dense(num\_neurons,  activation="relu",  input\_shape=[num\_x]))  for n in range(num\_layers-1):  model.add(Dense(num\_neurons,  activation="relu"))  model.add(Dense(num\_y))  optimizer = keras.optimizers.Adam(learning\_rate = learning\_rate,  beta\_1=0.9, beta\_2=0.999)  model.compile(loss='mse', optimizer = optimizer)  return model |

인공신경망 함수는 입력변수 개수, 출력변수 개수, 은닉층 개수, 은닉뉴런 개수, 학습률을 인자로 갖는다. 은닉층과 은닉뉴런의 개수는 증가할수록 모델 파라미터의 수가 증가하여 더 섬세한 모델링이 가능하지만 오버피팅(overfitting) 문제가 발생할 수 있다. 따라서 문제에 따라 적절한 개수로 설정해주어야 하며, 일반적으로 하이퍼파라미터 최적화 과정을 통해 결정된다.

다음으로 인공신경망 함수를 이용하여 모델을 만든다.

|  |
| --- |
| num\_x = len(var\_x)  num\_y = len(var\_y)  num\_layers = 3  num\_neurons = 20  learning\_rate = 0.001  nn\_model = NeuralNet(num\_x, num\_y, num\_layers, num\_neurons, learning\_rate) |

신경망의 구조는 3개의 은닉층, 각 층마다 20개의 뉴런, 0.001의 학습률로 설정하였다. 그리고 학습 종료조건으로 조기종료(early stopping) 기법을 사용하였다. 이제 모델을 훈련해보자. 모델 훈련 시에는 다양한 훈련조건이 설정된다.

|  |
| --- |
| training\_epoch = 10000  patience = 30  early\_stopping\_cb = keras.callbacks.EarlyStopping(patience=30,  restore\_best\_weights= True,  monitor='val\_loss')  # 모델 저장 경로  import os  def CreateFolder(directory):  try:  if not os.path.exists(directory):  os.makedirs(directory)  except OSError:  print ('Error: Creating directory. ' + directory)  path\_save = f'{Path\_project}/Model'  model\_name = 'nn\_model'  CreateFolder(path\_save)  # 모델 학습  if not os.path.exists(f"{path\_save}/{model\_name}"):  history = nn\_model.fit(train\_x, train\_y,  epochs = training\_epoch, callbacks=[early\_stopping\_cb],  verbose = 2,  validation\_data=(test\_x, test\_y))  # 모델 저장  nn\_model.save(f'{path\_save}/{model\_name}')  # 모델 불러오기  else:  nn\_model = keras.models.load\_model(f"{path\_save}/{model\_name}") |

텍스트이(가) 표시된 사진

자동 생성된 설명

조기종료는 정해진 횟수(tolerance)동안 검증 데이터에 대한 손실 함수값이 더 이상 줄어들지 않으면 학습을 종료한다. 따라서 정해진 훈련 에포크(training epoch)만큼의 반복이 끝나기 전에 훈련이 조기종료될 수 있으며, 이 조건은 모델이 훈련데이터에 과적합(overfitting)되는 문제를 방지할 수 있다.

**Q6. 학습된 모델의 정확도를 평가하여라. 평가지표 R2와 RMSE의 차이점에 대해서 설명하여라.**

A6. 훈련이 끝난 모델은 R2와 RMSE 등의 평가지표를 통해 성능을 평가한다. R2과 RMSE 함수는 다음과 같이 설정된다.

|  |
| --- |
| import numpy as np  def R\_Squared(prediction, actual\_value):  actual\_mean = np.mean(actual\_value, axis=0)  SSR = np.sum( (prediction - actual\_value)\*\*2 , axis=0)  RSS = np.sum( (prediction - actual\_mean )\*\*2, axis=0 )  TSS = np.sum( (actual\_value - actual\_mean )\*\*2, axis=0 )  r2 = 1 - SSR/TSS  return r2  def RMSE(prediction, actual\_value):  rmse = np.sqrt(np.mean((prediction - actual\_value)\*\*2, axis=0))  return rmse |

예측(prediction)은 훈련된 모델에 입력 변수를 입력하였을 때 출력되는 예측값을 나타내고, 실제값(actual value)은 입력 변수 값에 해당하는 실제 출력 데이터값을 나타낸다. R2 지표는 0과 1사이의 값을 가지며, 1에 가까울수록 예측과 실제가 유사하다는 것을 의미한다. 반면에 RMSE 지표는 예측과 실제의 오차를 나타내는 값으로, 0에 가까울수록 두 값이 일치한다는 것을 의미한다.

|  |
| --- |
| prediction = pd.DataFrame(scaler\_y.inverse\_transform(nn\_model.predict(test\_x)),  index=data\_test.index,  columns=var\_y)  actual\_value = data\_test[var\_y]  r2 = R\_Squared(prediction, actual\_value)  rmse = RMSE(prediction, actual\_value)  display(pd.DataFrame([r2, rmse], index=["R2", "RMSE"])) |

텍스트이(가) 표시된 사진

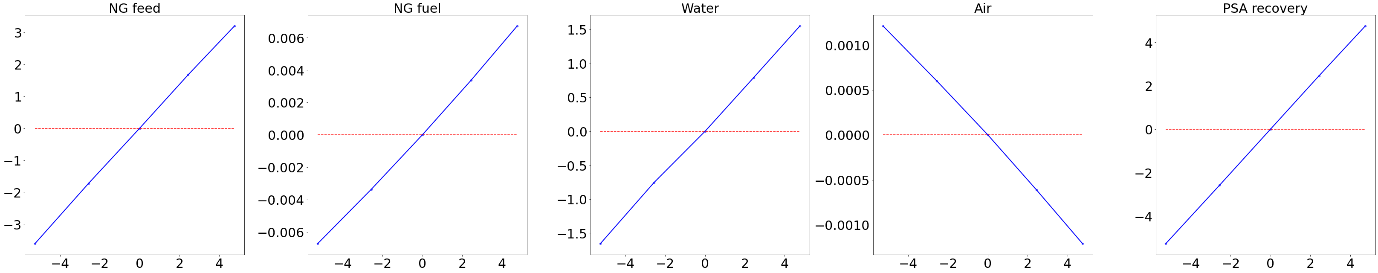
자동 생성된 설명

이 결과는 SMR 공정의 수소생산량에 대한 모델의 예측 성능을 나타낸다.

**Q7. 민감도 분석을 이용하여 공정 모델에 대한 각 입력변수의 영향도를 분석하여라.**

A7. 민감도 분석은 각 입력변수를 평균값의 -5%~+5% 범위에서 변화시키면서 공정의 반응을 관찰한다. 이 분석을 통해 각 변수들의 영향도를 알 수 있다.

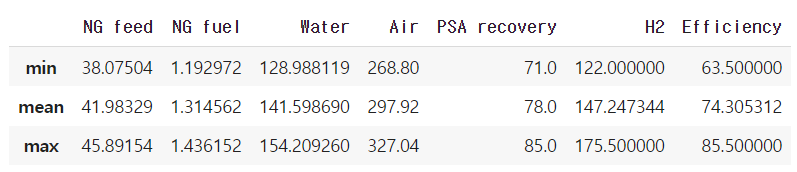
|  |
| --- |
| x\_base = data\_smr[var\_x].mean()  sensitivity\_points = 5  sensitivity\_result = SensitivityAnalysis(x\_base, sensitivity\_points,  nn\_model, var\_x, var\_y, scaler\_x, scaler\_y) |

수소생산량에 대해 3 개의 변수(천연가스 원료, 물, PSA 장치의 회수율)가 유의미한 영향도를 나타내고, 2 개의 변수(천연가스 연료, 공기)가 미미한 영향도를 나타냈다. 또한, PSA 장치의 회수율, 천연가스 원료, 물, 천연가스 연료, 공기의 순서대로 수소 생산량에 큰 영향력을 갖는다.

**Q8. 학습된 모델을 기반으로 격자탐색법을 이용하여 SMR 공정의 운전 조건을 최적화하여라. 최적화하고자 하는 목적함수에 따라서 최적 운전 조건이 어떻게 달라지는가?**

A8.앞에서 구축된 인공신경망 모델의 빠른 계산속도를 활용하여 최적의 운전조건을 탐색한다. 운전조건은 수집된 데이터에서 각 변수의 최솟값과 최댓값 사이에서 탐색하며, 이 범위를 탐색공간이라고 정의한다.

|  |
| --- |
| data = data\_smr  data.describe().loc[['min', 'mean', 'max']] |



최적 조건을 찾기 위한 알고리즘으로 격자탐색법(grid search)을 사용하였다. 격자탐색법은 탐색공간 내에서 격자점을 생성하여 모든 격자점에서의 목적함수 값을 확인하며 최적 해를 탐색하는 방법으로, 간단하고 빠르지만 탐색공간이 커지거나 결정 변수의 개수가 많아질수록 계산비용이 크게 증가하는 특징이 있다.

|  |
| --- |
| bins = 11  gridsearch\_result = GridSearch(data, bins, nn\_model, var\_x, var\_y, scaler\_x, scaler\_y) |



5개 입력변수마다 11개의 격자점을 조합하여 총 161,051개의 운전조건을 생성하여 최적의 조건을 탐색하였다. 생성된 운전조건들에 대해서 수소 생산량을 계산하고, 내림차순으로 정렬함으로써 수소 생산량이 최대가 되는 운전조건을 찾을 수 있다.

|  |
| --- |
| objective\_y = "H2"  Optimization(gridsearch\_result, var\_x, objective\_y) |

테이블이(가) 표시된 사진

자동 생성된 설명

수소 생산량이 최대가 되는 최적 운전조건을 탐색한 결과, 모든 입력변수의 값이 최대일 때 수소가 가장 많이 생산되었다. 한편, 수소 생산량을 바탕으로 SMR 공정의 열효율을 계산할 수 있다. 공정의 열효율은 다음과 같이 정의된다.

|  |  |
| --- | --- |
|  | (4) |

투입되는 원료 및 연료의 양을 고려했을 때, 수소를 효율적으로 생산하기 위해서는 공정 열효율을 높이는 것이 중요하다.

|  |
| --- |
| objective\_y = "Efficiency"  Optimization(gridsearch\_result, var\_x, objective\_y) |

테이블이(가) 표시된 사진

자동 생성된 설명

공정 열효율이 최대가 되는 운전조건을 탐색한 결과, 천연가스의 원료 및 연료, 공기의 유량은 줄고 물과 PSA 장치의 회수율은 증가하였다. 이와 같이 격자탐색법을 이용하여 격자점을 한번 생성하고 나면, 원하는 목적함수에 맞는 최적의 운전조건을 빠르게 탐색할 수 있다.

**결론**

본 장에서는 수소 생산을 위한 수증기 개질 공정을 인공신경망 기법을 이용하여 모델링하고, 최적의 운전조건을 탐색하는 최적화 문제를 해결하였다. 개발된 모델의 신뢰성 및 타당성을 검증하기 위하여 검증 데이터에 대한 모델의 R2 및 RMSE 값을 계산하였고, 민감도 분석을 통해 예측 변수에 가장 영향력 있는 입력 변수를 분석하였다. 민감도 분석 결과에서는 영향력의 크기가 PSA 장치의 회수율, 천연가스 원료 유량, 증기 유량, 공기 유량, 천연가스 연료 유량 순서대로 나타났다. 이를 통해 공정 운전 시 중심적으로 모니터링하고 제어할 변수를 쉽게 결정할 수 있다. 마지막으로, 격자탐색법을 기반으로 탐색공간 내에서 SMR 공정의 운전조건을 최적화하였다. 수소 생산량을 최대화하는 경우, 천연가스 원료와 연료의 유량을 늘려 최대한 많은 개질 반응이 일어나도록 해야 한다. 반면에 공정 열효율을 최대화하는 경우, 천연가스 원료와 연료의 유량을 줄임으로써 개질 반응의 전환율이 높아지기 때문에 투입되는 천연가스 대비 더 많은 수소를 생산할 수 있다. 이와 같이, 인공신경망 모델과 격자탐색법을 공정 최적화에 적용하면 다양한 목적함수에 대해 신속하고 정확하게 최적의 솔루션을 얻을 수 있다.