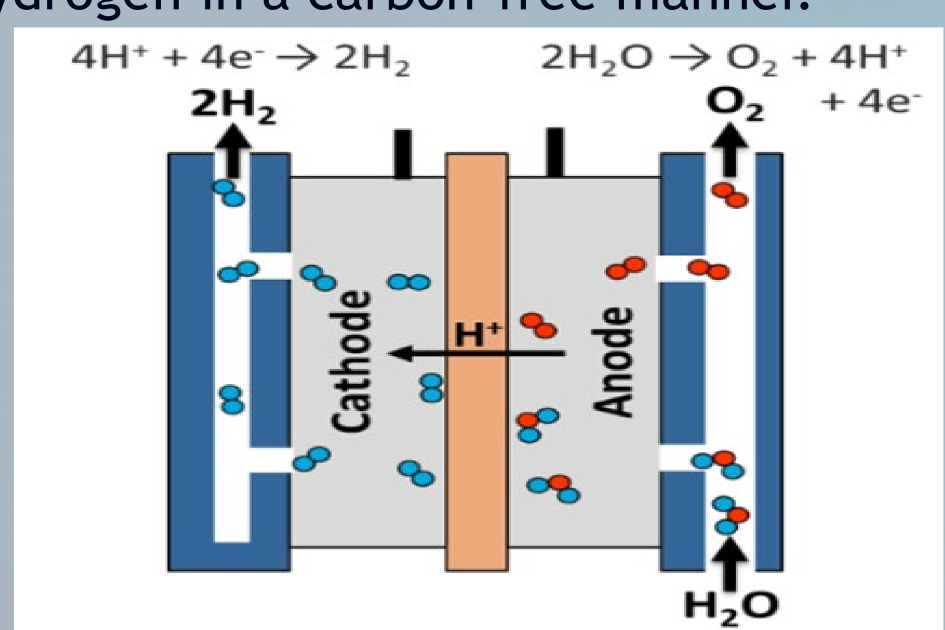
Hydrogen Optimization Model

Zhihan Wang

EME Summer Research Internship Program 2022

Introduction

Nowadays, the development of technologies and the influence of global warming have begun to receive attention, and our consumption on fossil energy start becoming an issue. With the high carbon emission and the non-renewability of fossil energy, the concept of energy transition is introduced to our society. Energy transition is a process of switching energy from fossil fuels to other low-carbon energy sources. More generally, the energy transition is a structural change in the supply and consumption of the existing energy system. A major step of the energy transition is to reduce or eliminate the usage of fossil fuels for automobiles. Hydrogen is considered as an alternative source for automobiles. Hydrogen can be generated from renewable energy sources, such as wind, and solar. Then, using electrolysis to split water into oxygen and hydrogen in a carbon-free manner.



Objective

- To build an economic optimization model for hydrogen facility using python.
- Define the total output and profit of the hydrogen facility.
- Use SciPy.optimize from python to construct an optimization function.
- To optimize the size of pipeline capacity to maximize facility's profit.
- To optimize the size of storage capacity to maximize facility's profit.
- To optimize the size of injection capacity to maximize facility's profit.

Method

Assume a joint generator/storage/pipeline facility. The generator has attached to it a pipeline and a storage facility with certain amount of capacity and injection ability. Each time period, hydrogen will be generated from the generator. Here are some steps:

1. Send hydrogen to the market through pipeline.
2. Inject remaining hydrogen into the storage.
3. Release Hydrogen into the atmosphere if hydrogen generated beyond facility capacity.
4. Withdraw hydrogen from storage if pipeline capacity not filled.

concept	ariable nam	Amount
Nameplate capacity	MW	105
Initial storage	STOR0	14
Storage capacity	CAPRES	20
Mean of distribution		0.28
Energy Efficiency	EE	0.75
MWH to hydrogen constantEE	$K_{MW\text{-}H}$	1
Fraction of hours in time intervals	FR	1
Pipeline capacity	CAPPIPE	5
efficiency sending directly to pipeline	EP	0.95
Injection capacity of reservoir	INJECT	9
efficiency of injecting into reservoir	ERES	0.92
efficiency of injecting from reservoir into pipeline	ERESPIPE	0.96

Formulas	Description
STORBt	Amount of H stored in the reservoir at the beginning of the period t.
CAPAVAILt = CAPRES-STORBt	Capacity available in the reservoir in period t
It	The amount of solar or wind energy the generator receives at period t.
$H_t = K_{MW\text{-}H\text{-}}$*EE*FR*MW* I_t	total amount H generated in period t
HIPt = EP*Ht	amount H available to delivery to pipeline from the generator
$HIPD_t$=Min (HIPt, CAPPIPE)	amount H delivered to the pipeline from generator
HIPGt = 1/EP *HIPDt or HIPGt = HIPDt/EP	total amount of H spent to inject into the pipeline
HAVAILRESt = Ht - HIPGt	amount of H send to the reservoir
RESAVAILt= min(INJECT, CAPAVAILt)	amount of H that is available to inject into the reservoir
HIRt= min(RESAVAILt, HAVAILRES*ERES)	amount of H available to be injected into the reservoir
HWAS= HAVAILRESt - HIRt/ERES	amount of H releases into the atmosphere
PIPECAPAVAILRESt=CAPPIPE-HIPDt	remaining space in the pipeline
=Min(INJECT , STORBt*ERESPIPE, PIPECAPAVA	amount of H available to deliver from resvoir to pipeline
HRESPIPEDPt= HRESt/ERESPIPE	amount of H delivered to the pipeline from reservoir
NETRESt=HIRt - HRESt	net H injection into reservoir
STOREt=STORBt+ HIRt - HRESPIPEDPt	the amount of hydrogen stored in the reservoir at the end of period t
HRESIJABt=CAPRES - STOREt	remaining storage space in the reservoir
HPIPEt=HRESPIPEDPt + HIPDt	amount H delivered to pipeline

Result

Python algorithms are created using the defined variables and formulas.

```
import numpy as np
from scipy.optimize import minimize
from scipy import optimize
import csv
```

```
MW = float(105) #Nameplate capacity, float creats numbers in decimal points, 105 is the initial value
STOR = [14] #list of storage values
EE = float(0.75) #energy efficiency
K_mwh = float(1) #MWH to hydrogen constant EE
FR = float(1)#Fraction of hours in time intervals
EP = float(0.95)#efficiency sending directly to pipeline
ERES = float(0.92)#efficiency of injecting into reservoir
ERESPIPE = float(0.96)#efficiency of injecting from reservoir into pipeline
I = []
```

```
file = open('solar.working.2021.csv', 'r')
csv_reader = csv.reader(file, delimiter=',')
for row in csv_reader:
    I.append(row[5])
```

```
def tot_inj(x):
    HIP = []
    HIR = []
    HRES = []
    CAPPIPE = x[0]
    CAPRES = x[1]
    INJECT = x[2]
    TF = len(I)#Input the number of time periods
    HPIPE = [0]#Stores the amount of H2 eventually delivered to the pipeline in each period
    HWAS = []
    H = []
    for t in range(1, TF):#Calculation program going through every period
        CAPAVAIL_t = CAPRES - STOR[t-1]
```

```
        HPIPE_t = HRES_t + HIPD_t
        #Total amount of H2 delivered to pipeline = amount of H2 delivered from reservoir + amount of H2 delivered from generator
        HPIPE.append(HPIPE_t)
        #Stores amount of H2 sent to pipeline in each period
        HPIPE_tf = sum(HPIPE)
        HWAS_tf = sum(HWAS)
    HIP_tf = sum(HIP)
    HIR_tf = sum(HIR)
    HRES_tf = sum(HRES)
    H_tf = sum(H)
    #print('The total amount of Hydrogen injected into pipeline at the end of', TF-1, 'periods are:', '%.2f' % HPIPE_tf, 'The total
    return (-1) * (30*HPIPE_tf - 26000*CAPPIPE - 62000*(CAPRES**0.5) - 1000*(INJECT**1.5))
```

```
def constraint1(x):
    return x[0] - 2*x[1] - 2*x[2]
def constraint2(x):
    return x[0]+x[1]+x[2]-50
x0 = [60,15,15]
print(tot_inj(x0))
b = (60, 80)
b2 = (10, 30)
b3 = (10, 20)
bnds = (b, b2, b3)
cons1 = {'type': 'eq', 'fun': constraint1}
cons2 = {'type': 'ineq', 'fun': constraint2}
cons = [cons1, cons2]
sol = optimize.minimize (tot_inj, x0, bounds = bnds, constraints = cons)
print(sol)
```

Conclusion

The algorithm have defined the optimal scenarios for both solar plant and wind plant from PJM.

Solar:

Optimal pipeline capacity: 56.65

Optimal reservoir capacity: 10

Optimal injection capacity : 18.32

Maximum profit: $2,627,905.69

```
     fun: -2627905.6887573665
     jac: array([-3250.     ,  9802.625  ,  6420.65625])
 message: 'Optimization terminated successfully'
    nfev: 35
     nit: 6
    njev: 6
  status: 0
 success: True
       x: array([56.6458194 , 10.00093683, 18.32197288])
According to the calculation, the firm could make $ 2627905.69
 with the optimal capacities [56.6458194  10.00093683 18.32197288]
```

Wind:

Optimal pipeline capacity: 60.11

Optimal reservoir capacity: 10.05

Optimal injection capacity : 20

Maximum profit: $5,168,260.05

```
     fun: -5168260.053854374
     jac: array([-4870.     ,  9777.     ,  6708.1875])
 message: 'Optimization terminated successfully'
    nfev: 103
     nit: 15
    njev: 15
  status: 0
 success: True
       x: array([60.10688086, 10.05344043, 20.        ])
According to the calculation, the firm could make $ 5168260.05
 with the optimal capacities [60.10688086 10.05344043 20.        ]
```

Acknowledgement

I would like to thank the Penn State EME Department for providing me with this excellent opportunity to gain research skills and experience.

I would also like to thank Dr. Kleit and Dr. Dahi for their guidance and support throughout the research process.

Contact

Zhihan Wang – EME Summer Research Internship

Email: zpw5162@psu.edu

PennState College of Earth and Mineral Sciences

John and Willie Leone Family Department of Energy and Mineral Engineering