Example Calculation

Client: INTERNAL

Project: Compare uCHP with Air Source Heat Pump in Calgary

Calc: CALC-003

Authentication

> Stamp, Permit

Revision History

Revision	Date	Description	By	Reviewer
1	2021/10/31	DRAFT	KCD	

Objective

One of my previous articles looked at the effectiveness of a Cold Climate Air Source Heat Pump for heating a home in Calgary. Since Alberta derives most of it's electricity from gas turbine power plants, it was assumed that the additional electricity would be produced by a CCGT. Both the amount of CO2 emitted by the additional load on the CCGT and the annual cost was compared to the traditional way of heating a home: a high efficiency natural gas furnace.

The conclusion from the analysis was that the electrical expense for the CCASHP would be higher than the cost of natural gas in a furnace. This was not surprising. However, the net CO2 emissions caused by the additional load on the CCGT power supply would be lower than the emissions from the natural gas furnace. This was surprising.

In this article, we go in a different direction. Examine the effectiveness of a micro Combined Heat and Power facility to heat ones home, and compare with the cost and net CO2 emissions from the natural gas furnace and CCASHP that consumes power generated by a CCGT.

The same data sources are used:

- the utility bills for my home are correlated with monthly ambient temperature to determine a simple model for heat demand and electrical demand as a function of ambient temperature.
- 12 years of hourly temperature data are used to provide a decent statistical view of the typical ambient temperature in Calgary.
- the performance data for a research-based CCASHP is used to determine the COP as a function of temperature.

A CHP system is operated with the primary objective of recovering waste heat to keep your house warm. Electricity is produced as a valuable byproduct. This allows a CHP unit to extract more value from natural gas, compared to a gas furnace. This is the exact oppositie of a gas turbine power plant, where the objective is to produce power, and none of the waste heat is utilized.

For the CHP system, performance data for the 6 kW XRGI-6 (EC Power, from Denmark) [1] is used.

- the electrical efficiency is 30.1% (LHV basis)
- waste heat recovery efficiency is 72.3% (LHV basis).

This is a fairly large unit, about the same size as stove or washing machine and quite heavy at 440 kg. This would be located outside ones home.

For net metering in Alberta, we will compare the amount of electricity produced by the CHP device to the amount of electricity that my home consumes. For electricity that backs out my demand from the grid, we will apply the net electrical cost of $\$0.14$ per kW.hr. For surplus electricity for the month, it is assumed that the electricity is pushed to the grid and we get nothing in return.

In [1]:
```
import numpy as np
import pandas as pd
import csv

import math as ma
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model

import scipy as sp
import copy
from scipy import interpolate



from sklearn.linear_model import LinearRegression

regressor = LinearRegression()
```

Get COP data for the heat pump

In [2]:
```
COP = pd.read_csv('newCCASHP_COP.csv');
COP['t_C']  = (COP['t_F'] - 32.0)* 5.0/9.0;
```

COP for uCHP

In [3]:
```
# fuel consumption for Kubota DF-752 on LPG
# This data is not used
lpgFuelCons = 225.0 # g of C3H8 per kW.hr
hhvC3H8 = 50158.0 # kJ/kg
hhvCH4 =  55384.0
lhvCH4 =  50000.0

# CH4 + 1.O2 = CO2 + 2.H2O, 44 per 16
# C3H8 + 5.O2 = 3.CO2 + 4.H2O, 3*44 per (12*3+8)
specCO2C3H8 = 44.0/16.0 #wt per wt
specCO2CH4 = 3*44.0/(12.0*3 + 8.0)

engineCOP = 1/(lpgFuelCons*(1/1000)*hhvC3H8/3600) # shaft wrok per fue
l energy
overallCOP = 0.85 # recovery 85% of total energy
heatCOP = overallCOP - engineCOP
generatorCOP = 0.85
```

In [4]:
```
# use COP values from EC 6kW device
# this uses LHV, not HHV
chpElecCOP = 0.301
chpHeatCOP = 0.723
```

Heat required for house in winter months

In [5]:
```
# my fuel consumption data
fuelConsumptionTable = {'Month': ['September','October','November','De
cember', 'January','February'],
                        'GJ': [3,13, 16, 19, 21, 27],
                        'days':     [30., 31., 30., 31., 31., 28.],
                        'ambientT': [13.2, 3.1, -0.2, -2.4, -4.2, -11.2
]}
# and I need the kW of electrical consumption as well
# there is a base load of 2 GJ per month that is not related to temper
ature.
fuelCons = pd.DataFrame(fuelConsumptionTable, columns = ['Month', 'GJ'
, 'days', 'ambientT'])
fuelCons['kJhr'] = fuelCons['GJ']*1000000/(fuelCons['days']*24)
```

In [6]:
```
# note that our electrical consumption is equal to
# kW = 1.5938 - 0.022198*T_c
```

In [7]:
```
# Create linear regression object
regr = linear_model.LinearRegression()
X = fuelCons['ambientT'].values.reshape(-1,1)

# The data is in GJ per month and we want kJ per hour
# account for the 2 GJ/month offset in the summer
# and use 75% as fuel efficiency of my furnace
Y = 0.75*(fuelCons['kJhr'].values - 2.0)

# Train the model using the training sets
regr.fit(X, Y)
```

Out[7]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, norma
lize=False)
```

In [8]:
```
model = LinearRegression().fit(X, Y)
(model.intercept_, model.coef_) # this is in kJ/hr
```

Out[8]:
```
(16913.119598794845, array([-1097.23352951]))
```

Hourly data for ambient temperature

I compiled 12 years of hourly data for Calgary International Airport (Station ID 3031094). then divide the data into individual months.

In [9]:
```
# lets get the ambient temperature data
weatherData = pd.read_csv('calgaryWeather4.csv');
```

```
/Applications/anaconda3/lib/python3.7/site-packages/IPython/core/int
eractiveshell.py:3058: DtypeWarning: Columns (13,15,17,25) have mixe
d types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [10]:
```
weatherData['realTempC'] = weatherData['tempCfilled'].astype(float)
weatherData['intMonth'] = weatherData['Month'].astype(int)
```

In [11]:
```
# calculate a bunch of stuff on the hourly data
# I want the hourly heat rate
designX = np.array(list(weatherData['realTempC']))
designHeat = model.predict(designX.reshape(-1,1))

weatherData['heatRateRaw'] = designHeat # in kJ/hr
# impose a minimum value of zero heatRate
weatherData['allZero'] = 0.0 # fill with zeros
weatherData['heatRate'] = weatherData[['heatRateRaw','allZero']].max(a
xis=1)

# note that our electrical consumption is equal to
# kW = 1.5938 - 0.022198*T_c
weatherData['elecDemandkW'] = 1.5938 - 0.022198*weatherData['realTempC
'] # elec demand in kW, and this is for one hour
```

In [12]:
```
# and I also want to calculate the COP for each point in time
interpType = 'linear'
weatherData['heatpumpCOP'] = interpolate.interp1d(COP['t_C'], COP['COP
'],  kind=interpType, bounds_error = False, fill_value="extrapolate",
)(designX)
weatherData['heatpumpElec'] = weatherData['heatRate'] / weatherData['h
eatpumpCOP']
weatherData['heatpumpKWelec'] = weatherData['heatpumpElec']/3600

# sumHeatRate['chpElecValued'] = sumHeatRate[['elecDemandkW','chpElecM
adekW']].min(axis=1)/12
```

In [13]:
```
# get the CO2 intensity for burning methane
fuelHHV = 55385.0 # kJ/kg, HHV
fuelLHV = 50000.0
ch4IntensityCO2 = 44.0/16.0
fuelCO2IntensityHHV = ch4IntensityCO2 / fuelHHV # kg CO2 per kJ HHV or
LHV
fuelCO2IntensityLHV = ch4IntensityCO2 / fuelLHV

# we need the fuel needed to produce 1 kJ of electricity from a CCGT.
# Assume this is what provides our electricity
CCGTHeatRate = 7649.0 # BTU per kW.hr from NatGas, HHV
# this COP is about 2.2 kJ fuel per kJ electricity
copCCGT = CCGTHeatRate * 1.0550559 / 3600.0   # BTU / (kJ/s . hr) * 1.
055e3 kJ/MMBTU * 1 hr / 3600 sec

copFurnace = 0.9 # 90% efficient furnace

fuelPrice = ((193.65-8.52)/(11.86+9.21))  # $ per GJ, and this is equi
valent to about 0.03 $ per kW.hr
fuelPriceLHV = fuelPrice*fuelHHV/fuelLHV
# wholesale gas price is about 3.5$ per GJ for this point in time
# with 90% conversion of LHV to electricity, this is about 3.5 c/kW.hr

# and we will have electricity price in $ per GJ
# $ per (kJ/s . hr) * 1 hr / 3600 sec * 1e6 kJ/1 GJ
elecPricekwhr = (217.46 - 20.97 - 7.48 - 6.92)/(941.7+346.3) # total u
tility bill less local access fee, admin charge rate riders
elecPrice = (1e6/3600.0) * elecPricekwhr # this is in $/GJ
```

In [14]:
```
elecPricekwhr,fuelPrice*3600/0.9/1e6
```

Out[14]: (0.1413742236024845, 0.03514570479354532)

In [15]:
```
# provide the CHP COP values and such
weatherData['chpFuelLHV'] = weatherData['heatRate'] / chpHeatCOP #kj/h
r
weatherData['chpElecMade'] = weatherData['chpFuelLHV'] * chpElecCOP
weatherData['chpElecMadekW'] = weatherData['chpElecMade']/3600 # this
is in kW
weatherData['furnaceFuel'] = weatherData['heatRate'] / copFurnace

weatherData['chpElecValued'] = weatherData[['elecDemandkW','chpElecMad
ekW']].min(axis=1)
```

In [16]:
```
weatherData['furnaceCO2'] = weatherData['furnaceFuel']*fuelCO2IntensityHHV
weatherData['heatPumpCO2'] = (weatherData['heatpumpElec']*copCCGT)*fuelCO2IntensityHHV
weatherData['chpCO2direct'] = weatherData['chpFuelLHV']*fuelCO2IntensityLHV
weatherData['chpCO2displace'] = (weatherData['chpElecMade']*copCCGT)*fuelCO2IntensityHHV
weatherData['chpCO2'] = weatherData['chpCO2direct'] - weatherData['chpCO2displace']
```

In [17]:
```
# lets get the cost values for furnace, heat pump and chp
# furnace fuel is in kJ per hour. Fuel price is in $/GJ HHV
weatherData['furnaceFuelCost'] = weatherData['furnaceFuel']*fuelPrice/1.e6
weatherData['heatPumpElecCost'] = weatherData['chpElecMadekW']*elecPricekwhr
weatherData['chpFuelCost'] = weatherData['chpFuelLHV']*fuelPriceLHV/1.e6
weatherData['chpElecMade'] = weatherData['chpElecMadekW']*elecPricekwhr # each data point is an hour, thus kW becomes kW.hr
weatherData['chpNetCost'] = weatherData['chpFuelCost'] - weatherData['chpElecMade']
```

In [18]:
```
list(weatherData)
```

Out[18]:
```
['Longitude',
 'Latitude',
 'StationName',
 'ClimateID',
 'DateTime',
 'Year',
 'Month',
 'Day',
 'Time',
 'TempC',
 'TempFlag',
 'tempCfilled',
 'DewPointC',
 'DewPointTempFlag',
 'RelHum',
 'RelHumFlag',
 'PrecipMM',
 'PrecipAmountFlag',
 'WindDir',
 'WindDirFlag',
 'WindSpdkmh',
 'WindSpdFlag',
```

```
               'Visibilitykm',
               'VisibilityFlag',
               'StnPresskpa',
               'StnPressFlag',
               'Hmdx',
               'HmdxFlag',
               'WindChill',
               'Wind ChillFlag',
               'Weather',
               'realTempC',
               'intMonth',
               'heatRateRaw',
               'allZero',
               'heatRate',
               'elecDemandkW',
               'heatpumpCOP',
               'heatpumpElec',
               'heatpumpKWelec',
               'chpFuelLHV',
               'chpElecMade',
               'chpElecMadekW',
               'furnaceFuel',
               'chpElecValued',
               'furnaceCO2',
               'heatPumpCO2',
               'chpCO2direct',
               'chpCO2displace',
               'chpCO2',
               'furnaceFuelCost',
               'heatPumpElecCost',
               'chpFuelCost',
               'chpNetCost']
```

In [19]:
```
shortList = copy.copy(weatherData[[
    'DateTime','Year','Month','intMonth','Day','Time',
    'realTempC','heatRate','elecDemandkW',
    'heatpumpElec',
 'furnaceCO2',
 'heatPumpCO2',
 'chpCO2direct',
 'chpCO2displace',
 'chpCO2',
 'furnaceFuelCost',
 'chpElecMadekW',
 'heatPumpElecCost',
 'chpFuelCost',
 'chpNetCost',
 'heatpumpElec',
 'chpFuelLHV',
 'furnaceFuel',
]])
# drop the rows where we do not need heat
#shortList.drop(shortList[shortList.heatRate < 0.0].index, inplace=Tru
e)
shortList.head()
```

Out[19]:

	DateTime	Year	Month	intMonth	Day	Time	realTempC	heatRate	elecDemandkW	h
0	2010-01-01 0:00	2010	1	1	1	0:00	-21.6	40613.363836	2.073277	
1	2010-01-01 1:00	2010	1	1	1	1:00	-21.2	40174.470424	2.064398	
2	2010-01-01 2:00	2010	1	1	1	2:00	-20.8	39735.577013	2.055518	
3	2010-01-01 3:00	2010	1	1	1	3:00	-20.4	39296.683601	2.046639	
4	2010-01-01 4:00	2010	1	1	1	4:00	-20.4	39296.683601	2.046639	

5 rows × 23 columns

In [20]:
```
sumHeatRate = shortList.groupby(['intMonth']).sum()['heatRate'].reset_
index()
meanTemp = shortList.groupby(['intMonth']).mean()['realTempC'].reset_i
ndex()
#numHours = shortList.groupby(['intMonth'])['Day'].count().reset_index
()
furnaceCO2 = shortList.groupby(['intMonth']).sum()['furnaceCO2'].reset
_index()
heatPumpCO2 = shortList.groupby(['intMonth']).sum()['heatPumpCO2'].res
et_index()
chpCO2 = shortList.groupby(['intMonth']).sum()['chpCO2'].reset_index()
numHours = shortList.groupby(['intMonth']).count()['realTempC'].reset_
index()

# and the cost values
furnaceFuelCost = shortList.groupby(['intMonth']).sum()['furnaceFuelCo
st'].reset_index()
heatPumpElecCost = shortList.groupby(['intMonth']).sum()['heatPumpElec
Cost'].reset_index()
chpNetCost = shortList.groupby(['intMonth']).sum()['chpNetCost'].reset
_index()

#elecDemandkW
elecDemandkW = shortList.groupby(['intMonth']).sum()['elecDemandkW'].r
eset_index()
chpElecMadekW = shortList.groupby(['intMonth']).sum()['chpElecMadekW']
.reset_index()
```

In [21]:

```
# dataframe is initialized with the sum of the heatRate in kJ/hr, which is the kJ of heat
sumHeatRate['sumHeatRate'] = sumHeatRate['heatRate']*(1.0/(12.0*1e6))
sumHeatRate['numHours'] = numHours['realTempC']/12
sumHeatRate['meanTemp'] = meanTemp['realTempC']
sumHeatRate['furnaceCO2'] = furnaceCO2['furnaceCO2']/12
sumHeatRate['heatPumpCO2'] = heatPumpCO2['heatPumpCO2']/12
sumHeatRate['chpCO2'] = chpCO2['chpCO2']/12

sumHeatRate['furnaceFuelCost'] = furnaceFuelCost['furnaceFuelCost']/12
sumHeatRate['heatPumpElecCost'] = heatPumpElecCost['heatPumpElecCost']/12

sumHeatRate['elecDemandkW'] = sumHeatRate['numHours']*(1.5938 - 0.0222*sumHeatRate['meanTemp'])
sumHeatRate['chpElecMadekW'] = chpElecMadekW['chpElecMadekW']/12
sumHeatRate['chpElecValued'] = sumHeatRate[['elecDemandkW','chpElecMadekW']].min(axis=1)/12

sumHeatRate['chpNetCost'] = chpNetCost['chpNetCost']/12

sumHeatRate['monthText'] = ['Jan','Feb','Mar','Apr','May','June','July','Aug','Sept','Oct','Nov','Dec']
#sumHeatRate.set_index('monthText')
sumHeatRate.set_index('intMonth')
```

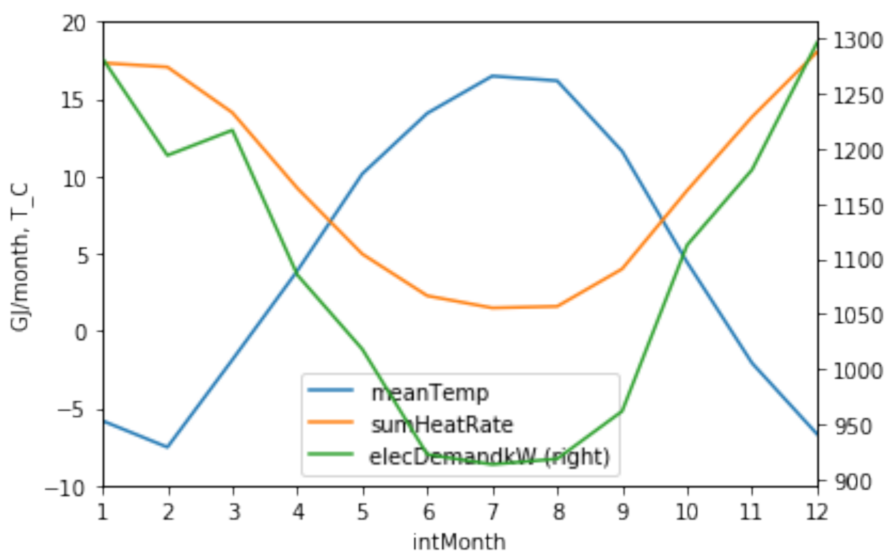
Out[21]:

	heatRate	sumHeatRate	numHours	meanTemp	furnaceCO2	heatPumpCO2	
intMonth							
1	2.079575e+08	17.329791	744.0	-5.814135	956.073638	735.855478	51
2	2.047717e+08	17.064308	678.0	-7.523697	941.427120	754.197004	50
3	1.693783e+08	14.114857	744.0	-1.869220	778.707776	541.774169	41
4	1.107370e+08	9.228082	720.0	3.885174	509.107467	295.146932	27
5	5.966585e+07	4.972154	744.0	10.155197	274.310601	139.991050	14
6	2.722132e+07	2.268444	720.0	14.073866	125.148600	58.408649	6
7	1.782941e+07	1.485784	744.0	16.480634	81.969756	39.291394	4
8	1.898229e+07	1.581858	744.0	16.174978	87.270088	40.309842	4
9	4.809326e+07	4.007772	720.0	11.617373	221.106235	110.896045	11
10	1.089741e+08	9.081179	744.0	4.438262	501.002916	288.298997	27
11	1.660789e+08	13.839907	720.0	-2.084965	763.538976	525.487904	41
12	2.163715e+08	18.030961	744.0	-6.673185	994.756751	768.779478	53

Analysis

First we will look at the amount of energy needed to keep our house warm, and which temperature consumes the majority of the heat.

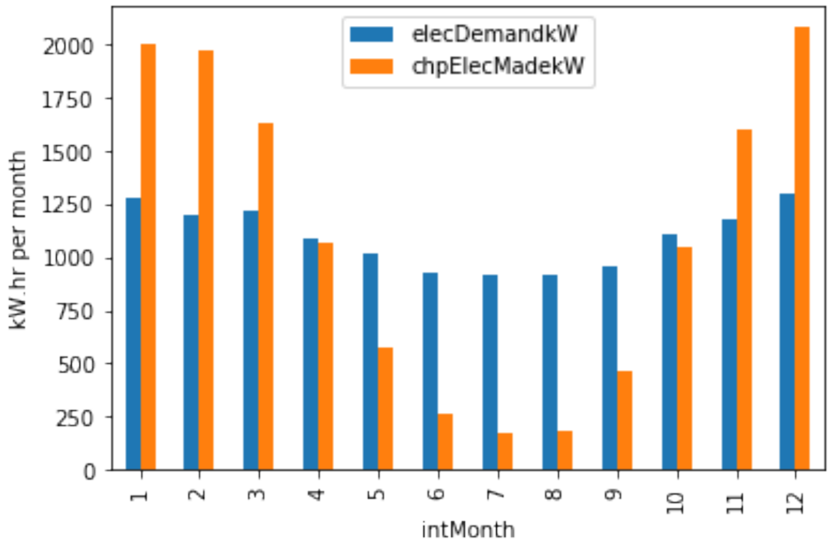
In [22]:
```
compareMonths = sumHeatRate[["intMonth","meanTemp","sumHeatRate"]]
compareMonths.set_index('intMonth',inplace = True);

compareMonths2 = sumHeatRate[["intMonth","elecDemandkW"]]
compareMonths2.set_index('intMonth',inplace = True);



ax = compareMonths.plot(kind='line')
ax.set_ylabel("GJ/month, T_C")
plt.ylim([-10,20])
plt.xticks(np.arange(1, 13, 1))
compareMonths2.plot(kind='line',secondary_y=True, ax=ax)
plt.savefig('comparemonths.png')
```



The coldest five winter months are November through March. Both heat demand and electrical demand are highest for these months. February has a curious dip in electrical demand: this is because February has only 28 days where the other winter months have 30 or 31.

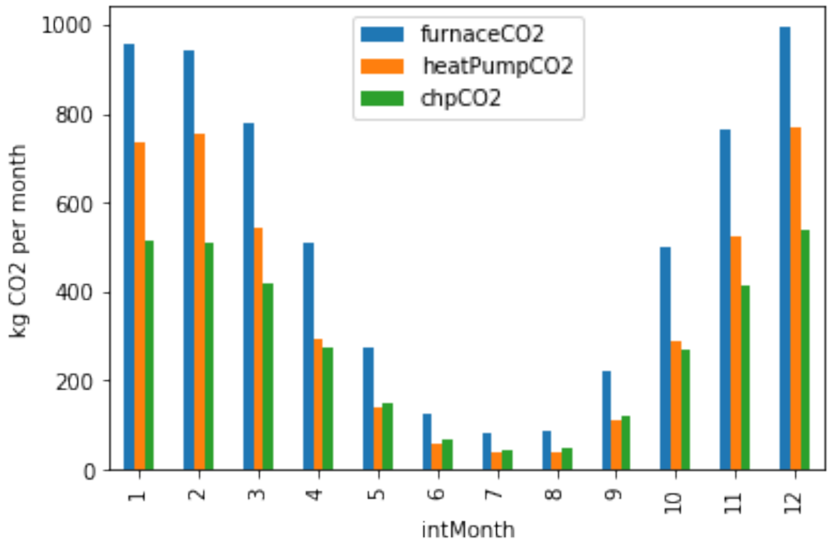
Our CHP unit will make the most power in the coldest months.

In [23]:
```
compareElec = sumHeatRate[["intMonth","elecDemandkW","chpElecMadekW"]]
compareElec.set_index('intMonth',inplace = True);

ax = compareElec.plot(kind='bar')
ax.set_ylabel("kW.hr per month")

plt.savefig('compareElectrical.png')
```



During the coldest months of the winter (November - March) the CHP unit produces about 50% more power than is consumed. Displacing our electric bill is worth about 0.14$ per kW.hr till we back out all of the consumption. I am assuming that we get zero value from the Utility company for surplus electricity that we produce, and is consumed by our neighbour.

In [24]:
```
compareCO2 = sumHeatRate[["intMonth","furnaceCO2","heatPumpCO2","chpCO2"]]
compareCO2.set_index('intMonth',inplace = True);
ax = compareCO2.plot(kind='bar')
ax.set_ylabel("kg CO2 per month")

plt.savefig('compareCO2.png')
```



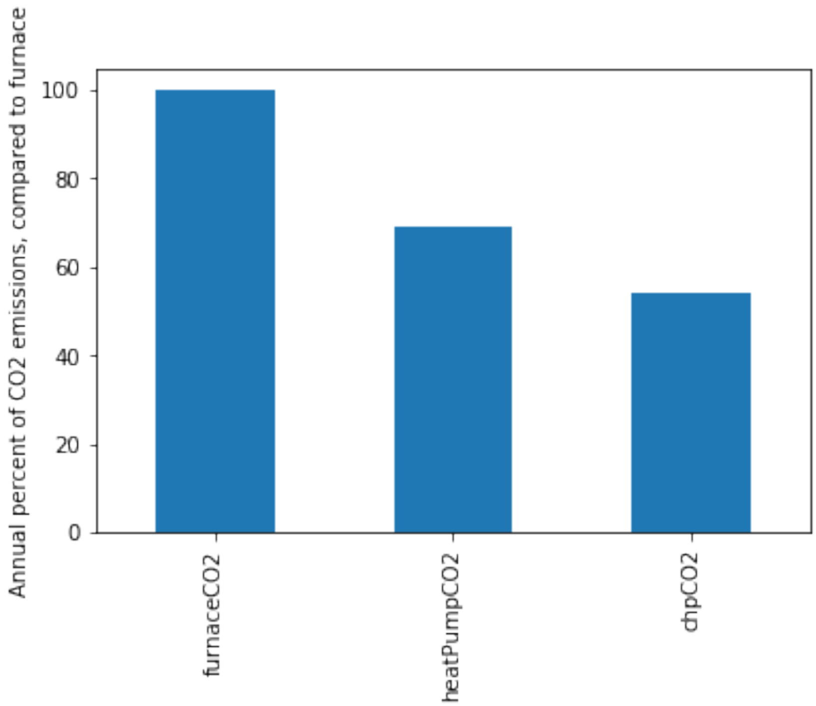
In the previous post we saw the surprising result with the net CO2 emissions from a CCASHPT in Calgary. Even though Alberta produces electricity by burning natural gas (with a CCGT), the heat pump has about 30% lower emissions than the natural gas furnace.

But the micro CHP unit goes a large step further. Because the CHP unit produces power and backs out the CCGT, the CHP unit results in about 40% lower CO2 emissions than the gas furnace. This is considerably better than the air source heat pump.Our low ambient temperature in the winter gives the edge to the CHP unit.

In [25]:
```
(compareCO2.sum()/compareCO2.sum().max()*100)
```

Out[25]:
```
furnaceCO2     100.000000
heatPumpCO2     68.946863
chpCO2          53.893952
dtype: float64
```

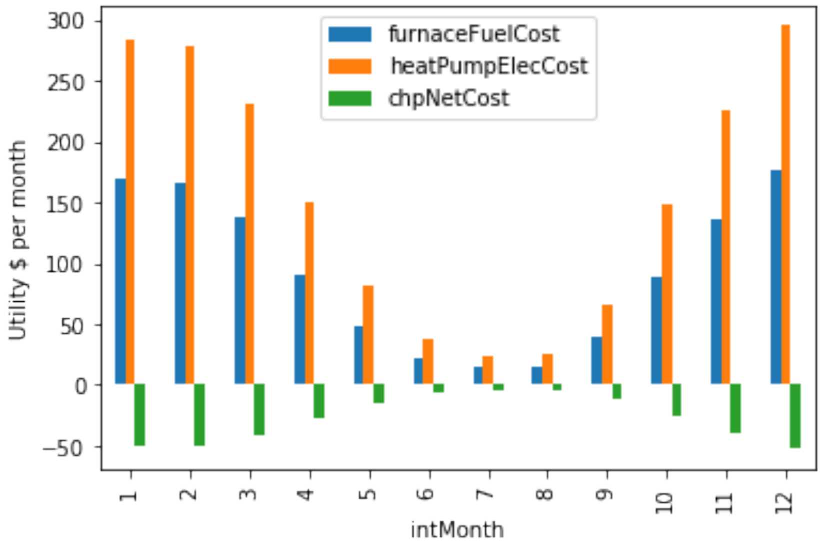
In [26]:
```
ax = (compareCO2.sum()/compareCO2.sum().max()*100).plot(kind='bar')
ax.set_ylabel("Annual percent of CO2 emissions, compared to furnace")


plt.savefig('annualCO2.png')
```



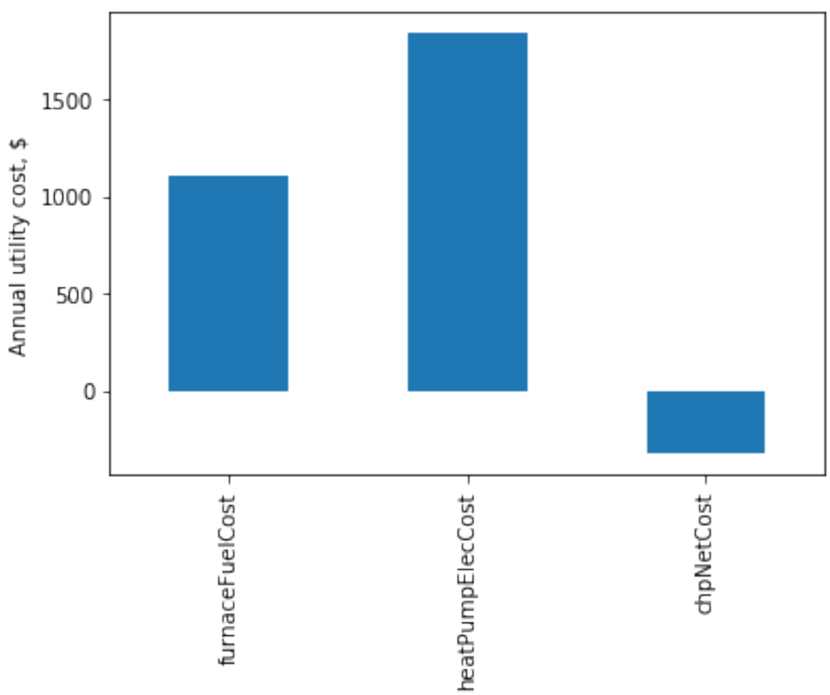
Over a year, the CHP unit has about 45% lower CO2 emissions compared with a furnace, and about 15% lower CO2 emissions than a heat pump.

Now, lets compare utility costs.

In [27]:
```
compareCost = sumHeatRate[["intMonth","furnaceFuelCost","heatPumpElecC
ost","chpNetCost"]]
compareCost.set_index('intMonth',inplace = True);
ax = compareCost.plot(kind='bar')
ax.set_ylabel("Utility $ per month")

plt.savefig('compareCost.png')
```



In [28]:
```
ax = compareCost.sum().plot(kind='bar')
ax.set_ylabel("Annual utility cost, $")

plt.savefig('annualCost.png')
```



Hang on. Why is the utility cost *negative* for the micro CHP unit?

From the last post, we saw that a heat pump costs about 40% more to operate than a gas furnace. We could argue that is the cost that an environmentally conscious person needs to bear. A CHP unit, on the other hand, produces electricity. This produced electricity is valued at only $\$0.035$ per kW.hr. And we backed out electricity valued at $\$0.14$ per kW.hr. Our little CHP unit has a significant impact on the utility expense for the home.

Summary

We can be environmentally conscious and save money. All it takes is an investment into a CHP unit to heat your home instead of the traditional gas furnace.

Keep in mind that this analysis assumes the worst case: surplus electricity is pushed to the grid and we get no compensation for it.

Suppose we had a system for purchasing surplus renewable power from micro producers. Then people with CHP units could recover more value from their surplus renewable power, and others could benefit from purchasing this energy. This would provide even greater incentive for conserving our resources.

What could you do with a winter supply of low cost, renewable electrical power? Operate grow lights for a small greenhouse or a cannabis operation? Mine Bitcoin? This excess supply is only available for 4 - 5 months per year, and depends on whether we have a brutal cold snap. But if your demand can take advantage of an opportunistic supply, or you have the greatest demand in the winter, then this could help your business.

All we need is for people in Alberta to change our ideas about how we heat our homes.

References

1. https://www.ecpower.eu/en/xrgi-systems.html (https://www.ecpower.eu/en/xrgi-systems.html)
2. Tom Marsik and others, Air Source Heat Pumps in Cold Climates, NREL, CCHRC (slide 9) (2018)