rTPC_protoTPC_flowrate_v1

November 2, 2020

[1]:
```
import numpy as np
from scipy import interpolate
```

[2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams["figure.figsize"] = (15,8)
x=np.linspace(5.0,150.0,300)
```

[3]:
```
#from confidence_ellipse import confidence_ellipse
from quadrature_error import quadrature_error_relative
```

1 Executive Summary

This document shows a successful application of the rotameter calibration curves posted on the manufacturer's website. However, the use of more refined instruments to determine the flow of a gas mixture is necessary given the discrepancy between the measurement made from the individual components of the mixture and the calculations made on the basis of the rotometer in the gas mixture supply line as shown in the Section **??**.

2 Calibration

2.1 Argon

Table at atmospheric pressure with stainless steel floater: $\phi_{\mathrm{Ar}}(z)$

[4]:
```
ro,cal=np.loadtxt('ARGON_602(E300)_SS_0_PSIG.dat',unpack=True) # sccm
arflowSS=interpolate.interp1d(ro, cal)
```

2.2 Carbon Dioxide

Table at atmospheric pressure with glass floater: $\phi_{\mathrm{CO_2}}(z)$

[5]:
```
ro,cal=np.loadtxt('CARBON_DIOXIDE_602(E300)_GLASS_0_PSIG.dat',unpack=True) #␣
↪sccm
co2flowPY=interpolate.interp1d(ro, cal)
```

Table at atmospheric pressure with stainless steel floater (used later).

[6]:
```
ro,cal=np.loadtxt('CARBON_DIOXIDE_602(E300)_SS_0_PSIG.dat',unpack=True) # sccm
co2flowSS=interpolate.interp1d(ro, cal)
```

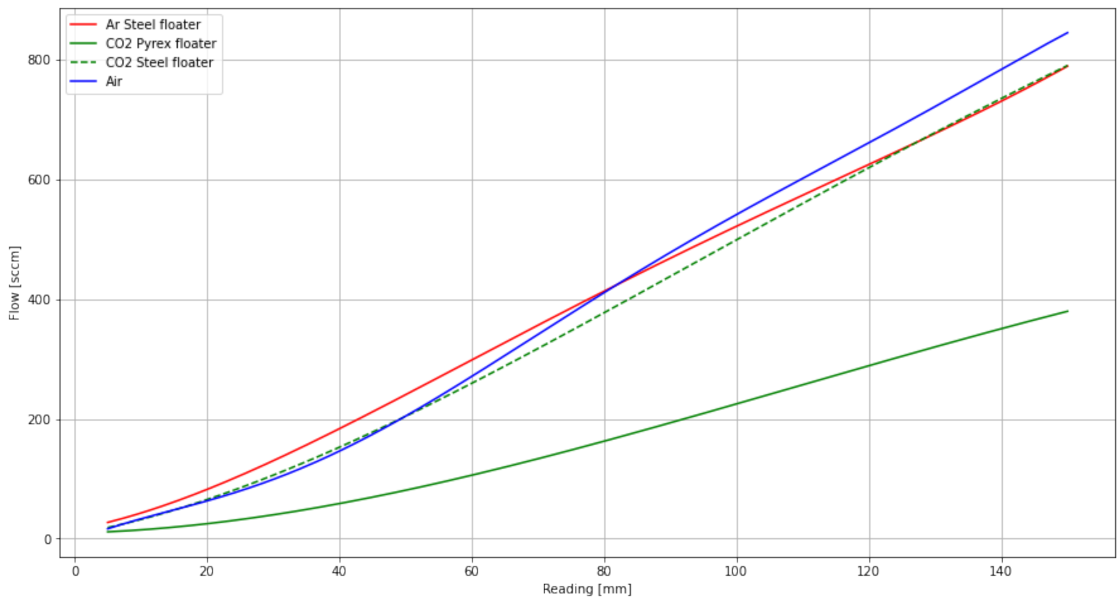
2.3 Air

Table at atmospheric pressure with stainless steel floater.

[7]:
```
ro,cal=np.loadtxt('AIR_602(E300)_SS_0_PSIG.dat',unpack=True) # sccm
airflow=interpolate.interp1d(ro, cal)
```

2.4 Plots

[8]:
```
plt.plot(x, arflowSS(x), 'r', label='Ar Steel floater')
plt.plot(x, co2flowPY(x),'g', label='CO2 Pyrex floater')
plt.plot(x, co2flowSS(x),'g--', label='CO2 Steel floater')
plt.plot(x, airflow(x), 'b', label='Air')
plt.xlabel('Reading [mm]')
plt.ylabel('Flow [sccm]')
plt.grid()
plt.legend(loc='upper left')
plt.show()
```



3 Determination of the flow prior to mixing

Carbon Dioxide flow $\phi_{CO_2}(z_1)$ for reading z_1

pic goes here

[9]:
```
z1=85 # reading of glass floater
inCO2Flow=co2flowPY(z1)
print(f'CO2 inflow {inCO2Flow:.0f} sccm')
```

```
CO2 inflow 178 sccm
```

Argon flow $\phi_{\mathrm{Ar}}(z_2)$ for reading z_2

pic goes here

[10]:
```
z2=90 # reading of stainless stell floater
inArFlow=arflowSS(z2)
print(f'Ar inflow {inArFlow:.0f} sccm')
```

```
Ar inflow 469 sccm
```

Total flow prior to mixing $\phi_{\mathrm{sum}} = \phi_{\mathrm{CO_2}}(z_1) + \phi_{\mathrm{Ar}}(z_2)$

[11]:
```
totFlow=inCO2Flow+inArFlow
print(f'Total input Flow {totFlow:.0f} sccm')
```

```
Total input Flow 647 sccm
```

Calculate CO_2 fraction $\eta_{\mathrm{CO_2}} = \frac{\phi_{\mathrm{CO_2}}(z_1)}{\phi_{\mathrm{sum}}}$

[12]:
```
co2frac=inCO2Flow/totFlow
print(f'CO2 fraction {co2frac*100:.0f}%')
```

```
CO2 fraction 28%
```

3.1 Error Estimation

This document reports that the accuracy of the tube is ±5% of full scale flow rate.

The full scale rate for Argon using the calibration above is 790 sccm

[13]:
```
Arerr=790.*0.05
print(f'Argon flow rate measurement accuracy: {Arerr:.1f}sccm')
```

```
Argon flow rate measurement accuracy: 39.5sccm
```

The full scale rate for Argon using the calibration above is 380 sccm

[14]:
```
CO2err=380.*0.05
print(f'Carbon Dioxide flow rate measurement accuracy: {CO2err:.1f}sccm')
```

```
Carbon Dioxide flow rate measurement accuracy: 19.0sccm
```

Therefore the total flow rate is

[15]:
```
totErr=Arerr+CO2err
print(f'({totFlow:.0f} +/- {totErr:.0f}) sccm')
```

```
(647 +/- 58) sccm
```

The error for the Carbon Dioxide fraction is calcualted in quadrature

[16]:
```
co2fracErr=co2frac*quadrature_error_relative(np.array([inCO2Flow,totFlow]),np.
 ↪array([CO2err,totErr]))
print(f'CO2 fraction ({co2frac*100:.0f} +/- {co2fracErr*100:.0f})%')
```

```
CO2 fraction (28 +/- 4)%
```

4 Determination of the flow of a gas mixture

4.1 Calculate correction for different gas mixture - Version 1

There are few reasons to believe that this approach works correctly given this and this. However, it's the method recommended by the manufacturer.

According to this document, the recipe to determine the flow rate of a gas mixture ϕ_{mix} is to calculate the weighted average of the gas factors f_i

$$F = \sum_i f_i \frac{v_i}{V_{\text{tot}}}$$

and to multiply the calibration for air $\phi_{\text{air}}(z)$ by it

$$\phi_{\text{mix}}(z) = F\phi_{\text{air}}(z)$$

.

[17]:
```
Ar_factor=0.851
CO2_factor=0.808
```

The weights are given by the fractional volume of the mixture $\eta = \frac{v_i}{V_{\text{tot}}}$.

By indicating the mixture fraction by $\eta_{\text{Ar}} = 1 - \eta_{CO_2}$, F is given by

[18]:
```
mix_factor=((1.0-co2frac)*Ar_factor)+(co2frac*CO2_factor)
print(f'The mix factor is {mix_factor:.5f}')
```

```
The mix factor is 0.83916
```

4.2 Calculate correction for different gas mixture - Version 2

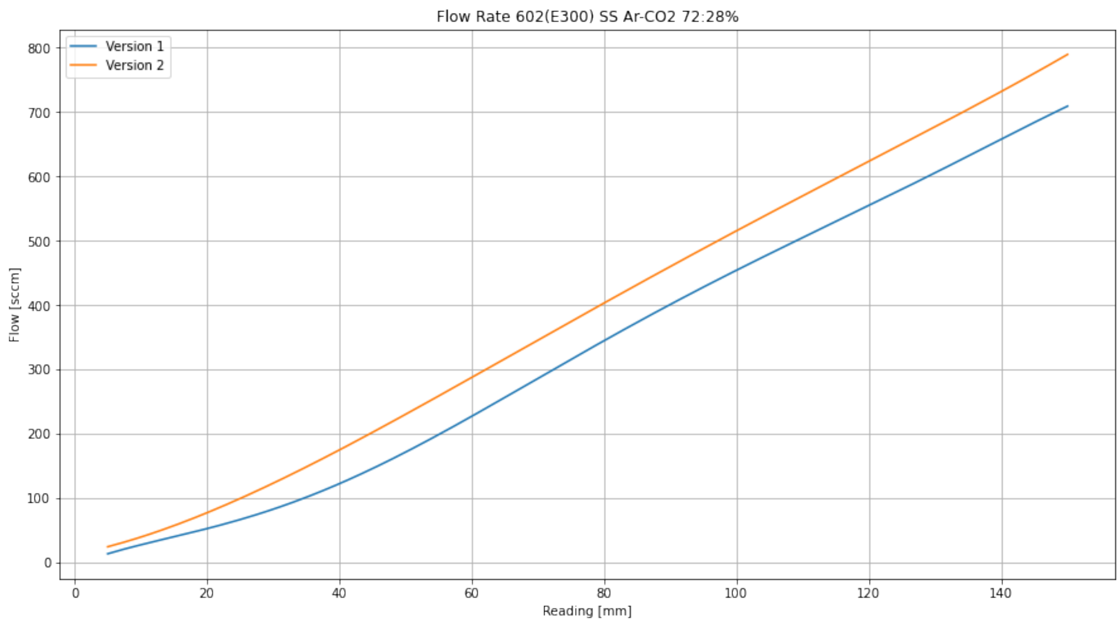
There is another way to calcualte the flow rate of a gas mixture and is as follows

$$\phi_{\text{mix}}(z) = \eta_{\text{Ar}}\phi_{\text{Ar}}(z) + \eta_{CO_2}\phi_{CO_2}(z)$$

[19]:
```
mixflow=interpolate.interp1d(x,(1.0-co2frac)*arflowSS(x)+co2frac*co2flowSS(x))
```

4.3 Plots

[20]:
```
plt.plot(x, airflow(x)*mix_factor, label='Version 1')
plt.plot(x, mixflow(x), label='Version 2')
plt.xlabel('Reading [mm]')
plt.ylabel('Flow [sccm]')
plt.grid()
plt.title(f'Flow Rate 602(E300) SS Ar-CO2 {(1.-co2frac)*100:1.0f}:{co2frac*100:
↪1.0f}%')
plt.legend(loc='upper left')
plt.show()
```



4.4 Readout of the SS floater

pic goes here

[21]:
```
inSS=105
outSS=55
```

Determine input flow post mixing with calculation "version 1"

[22]:
```
inFlowV1=airflow(inSS)*mix_factor
print(f'Gas Mixture inflow {inFlowV1:.0f} sccm')
```

```
Gas Mixture inflow 480 sccm
```

Determine input flow post mixing with calculation "version 2"

[23]:
```
inFlowV2=mixflow(inSS)
print(f'Gas Mixture inflow {inFlowV2:.0f} sccm')
```

```
Gas Mixture inflow 543 sccm
```

Determine output flow

[24]:
```
outFlow=airflow(outSS)*mix_factor
print(f'Gas Mixture outflow {outFlow:.0f} sccm')
```

```
Gas Mixture outflow 199 sccm
```

Determine return fraction

[25]:
```
print(f'Return fraction {outFlow/inFlowV1*100.0:.0f}%')
```

```
Return fraction 41%
```

4.5 Error Estimation

The error on the "version 1" calcuation is obtained from the full scale rate for the gas mixture using the calibration above and is as follows:

[26]:
```
mix_maxrate=airflow(150.)*mix_factor
print(f'Max flow rate for mixture {mix_maxrate:.0f} sccm')
mixerrV1=mix_maxrate*0.05
```

```
Max flow rate for mixture 709 sccm
```

The error on the "version 2" is determined in quadrature

[27]:
```
CO2err=790.*0.05
temp1=(1.-co2frac)*arflowSS(inSS)*quadrature_error_relative(np.array([(1.
 ↪-co2frac),arflowSS(inSS)]),
                                                                    np.
 ↪array([co2fracErr,Arerr]))
temp2=co2frac*co2flowSS(inSS)*quadrature_error_relative(np.
 ↪array([co2frac,co2flowSS(inSS)]),
                                                                np.
 ↪array([co2fracErr,CO2err]))
mixerrV2=temp1+temp2
```

Therefore the input and output flow rate of the mixture are

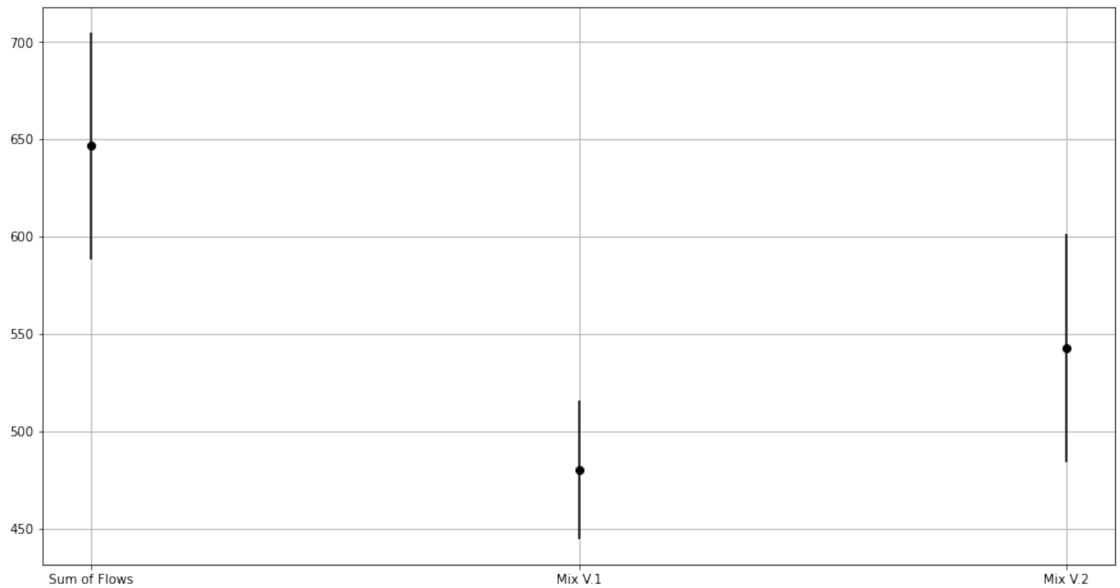
[28]:
```
print(f'Input ver.1: ({inFlowV1:.0f} +/- {mixerrV1:.0f}) sccm')
print(f'Input ver.2: ({inFlowV2:.0f} +/- {mixerrV2:.0f}) sccm')
print(f'Output: ({outFlow:.0f} +/- {mixerrV1:.0f}) sccm')
```

```
Input ver.1: (480 +/- 35) sccm
Input ver.2: (543 +/- 59) sccm
Output: (199 +/- 35) sccm
```

4.6 Plot

Put everything together

[29]:
```
n=range(3)
plt.errorbar(n, [totFlow,inFlowV1,inFlowV2], yerr=[totErr,mixerrV1,mixerrV2],␣
↪label='', fmt='ok')
plt.xticks(n, ('Sum of Flows','Mix V.1','Mix V.2'))
_,ymax=plt.ylim()
#plt.ylim(0.,ymax)
plt.grid()
plt.show()
```



Section ??

[ ]:
```

```