

análisis

September 15, 2015

1 Análisis de la importancia del tamaño de la Tolva para la alimentación de la extruora

Al intentar conseguir un modelo matemático de la extrusora para intentar cerrar el lazo mediante un PID, se ha comprobado que el sistema en lazo abierto no es lo suficientemente estable y el throughput de la extrusora no es constante. Una hipótesis que se tiene es que la alimentación de la granza en el extrusor no es del todo constante. Por ello una solución que se propone es la de intentar que la alimentación sea constante.

Se diseña una tolva que permita meter más granza para comprobar si el peso que ejerce la propia granza en la entrada del alimentador puede ser una posible solución.

La tolva diseñada permite introducir hasta un máximo de 150gr frente a los 42gr de la tolva de serie. Se va a hacer una producción en la que se va a muestrear los datos obtenidos para su posterior análisis.

```
In [2]: import numpy as np
import pandas as pd
import seaborn as sns
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
In [3]: print ("Numpy v{}".format(np.__version__))
print ("Pandas v{}".format(pd.__version__))
print ("Seaborn v{}".format(sns.__version__))
```

Numpy v1.9.2
Pandas v0.16.2
Seaborn v0.6.0

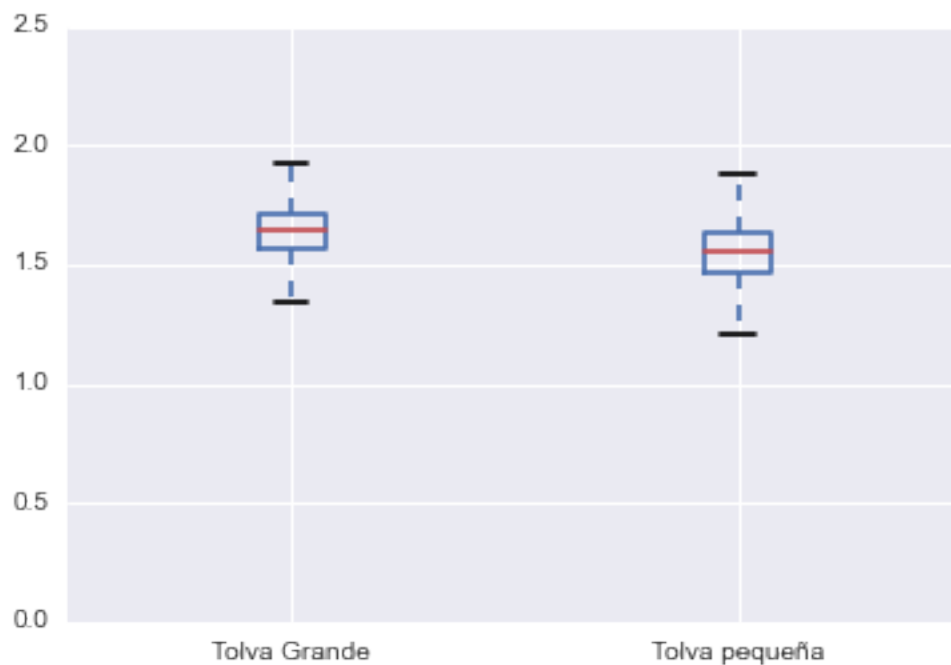
```
In [4]: #Abrimos los ficheros con los datos de las pruebas
#Tolva pequeña
data_p1 = pd.read_csv('prueba1.CSV')
#Tolva Granded
data_p2 = pd.read_csv('prueba2.CSV')
```

```
In [5]: #El ensayo se realizo a distintas velocidades, se filtran los datos por
#velocidad rápida= v1 = 4.42mm/s
#velocidad lenta= v2 = 3.816mm/s
#Filtramos los datos con velocidad de tracción rápida
data_p1_v1 = data_p1[(data_p1['RPM TRAC'] >= 4) & (data_p1['RPM TRAC'] <= 4.42)]
data_p2_v1 = data_p2[(data_p2['RPM TRAC'] >= 4) & (data_p2['RPM TRAC'] <= 4.42)]
#Filtramos los datos con velocidad de tracción lenta
data_p1_v2 = data_p1[(data_p1['RPM TRAC'] > 0) & (data_p1['RPM TRAC'] <= 3.816)]
data_p2_v2 = data_p2[(data_p2['RPM TRAC'] > 0) & (data_p2['RPM TRAC'] <= 3.816)]
```

```
In [6]: #d ={'Tolva pequeña V1':data_p1_v1.ix[:,"Diametro X"],'Tolva Grande V1':data_p2_v1.ix[:,"Diametro X"],
d ={'Tolva pequeña':data_p1.ix[:,"Diametro X"],'Tolva Grande':data_p2.ix[:,"Diametro X"]}
da = pd.DataFrame(data=d)
da.boxplot(return_type='axes',figsize=(0,0))
da.describe()
```

```
Out[6]:
```

	Tolva Grande	Tolva pequeña
count	2000.000000	2000.000000
mean	1.630594	1.558390
std	0.140171	0.153613
min	1.023350	0.010000
25%	1.562434	1.460000
50%	1.642723	1.550000
75%	1.711542	1.630000
max	2.204747	2.400000



```
In [7]: graf=da.plot(figsize=(20,10))
graf.axhspan(1.65,1.85, alpha=0.2)
```

```
Out[7]: <matplotlib.patches.Polygon at 0x8a87810>
```



Histogram of the ratio

```
In [8]: da.hist(bins=50)
```

```
Out[8]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x08ABEF30>,
               <matplotlib.axes._subplots.AxesSubplot object at 0x08D8F3B0>]], dtype=object)
```

