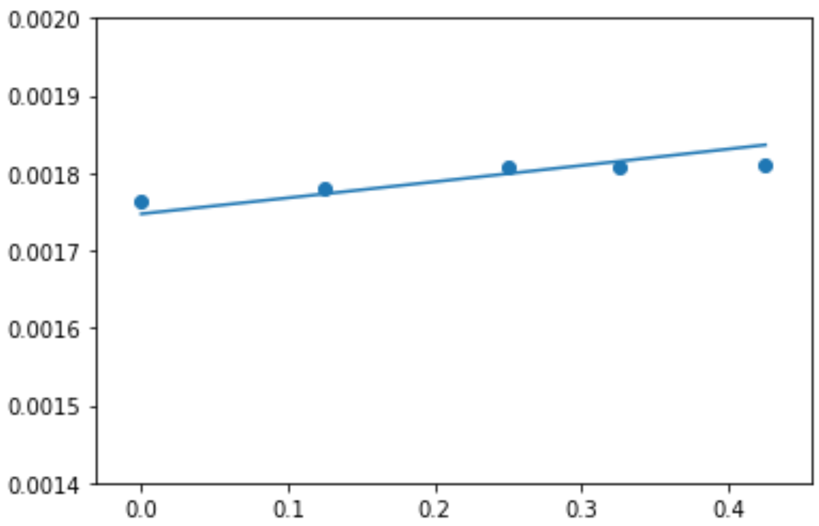
In [11]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from IPython.display import Image
from IPython.core.display import HTML
#%matplotlib
```

Fit of infrared beam going to filter cavity

1. oringial data and roughly manual fit

In [3]:
```
wW = np.array([3623, 3615, 3618, 3562, 3527], dtype = np.float)*10*
*-6/2
wV = np.array([3763, 3691, 3614, 3576, 3441], dtype = np.float)*10*
*-6/2
wW1 = np.flipud(wW)
wV1 = np.flipud(wV)
x = np.array([0, 5, 10, 13, 17])*2.5*1e-2
xp = np.linspace(0, 5, 200)
l = 1064e-9
w1 = 1200e-6
z1 = -4.5
y = w1*np.sqrt(1+((x-z1)/(np.pi*w1**2/l))**2)
plt.scatter(x, wW1)
plt.plot(x, y)
plt.ylim(0.0014, 0.002)
#print(np.shape(wW))
```

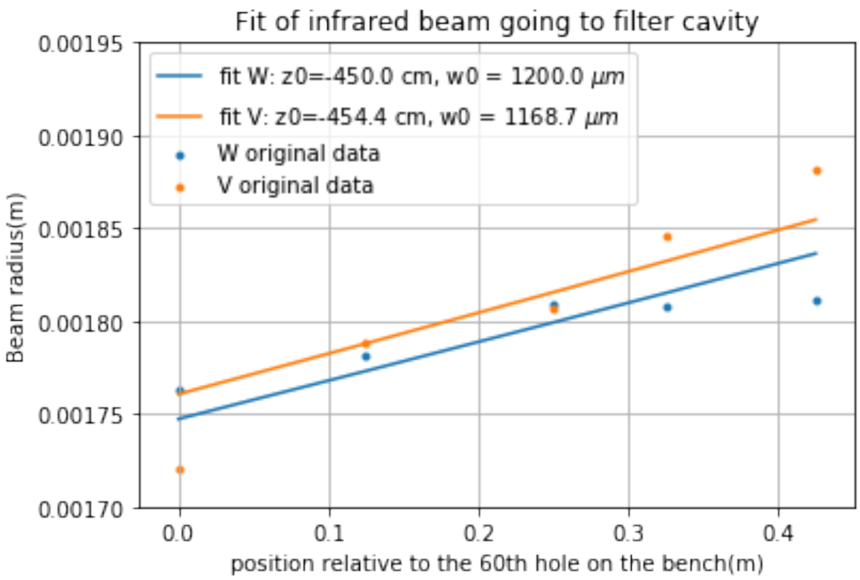
Out[3]:
```
(0.0014, 0.002)
```



2. fit by program

In [4]:
```
def gaussian(x, z0, w0):
    return w0*np.sqrt(1+((x-z0)/(np.pi*w0**2/l))**2)
```

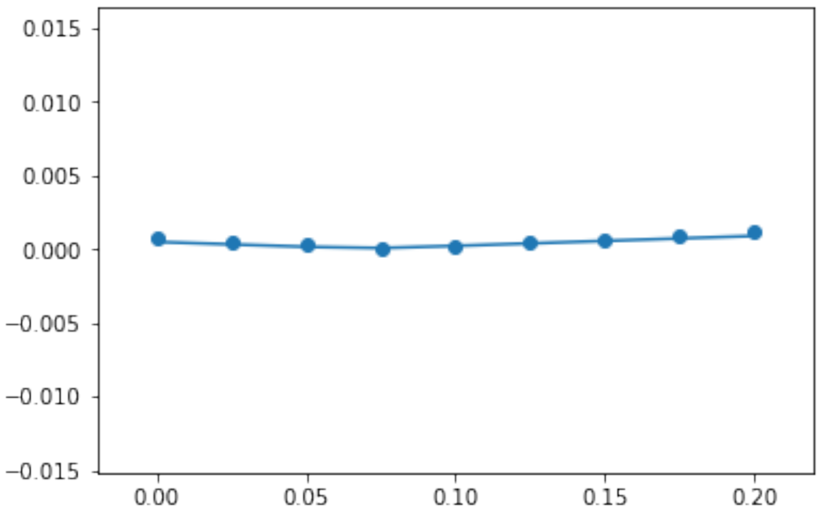
In [5]:
```
popt, pcov = curve_fit(gaussian, x, wW1, bounds=([-6, 1100e-6], [-3
, 1300e-6]))
popt2, pcov2 = curve_fit(gaussian, x, wV1, bounds=([-6, 1100e-6], [
-3, 1300e-6]))
plt.scatter(x, wW1, marker=".", label='W original data')
plt.scatter(x, wV1, marker=".", label='V original data')
plt.plot(x, gaussian(x, *popt),   label='fit W: z0=%4.1f cm, w0 = %
4.1f $\mu m$' % tuple(popt*[100, 1000000]))
plt.plot(x, gaussian(x, *popt2),   label='fit V: z0=%4.1f cm, w0 =
%4.1f $\mu m$' % tuple(popt2*[100, 1000000]))
plt.xlabel('position relative to the 60th hole on the bench(m)')
plt.ylabel('Beam radius(m)')
plt.title('Fit of infrared beam going to filter cavity')
#plt.xlim([0.184,0.1901])
plt.ylim([0.0017, 0.00195])
plt.legend()
plt.grid()
plt.show()

# residual sum of squares
ss_res = np.sum((x - gaussian(x, *popt)) ** 2)

# total sum of squares
ss_tot = np.sum((x - np.mean(gaussian(x, *popt))) ** 2)

# r-squared
r2 = 1 - (ss_res / ss_tot)

#print(r2)
```



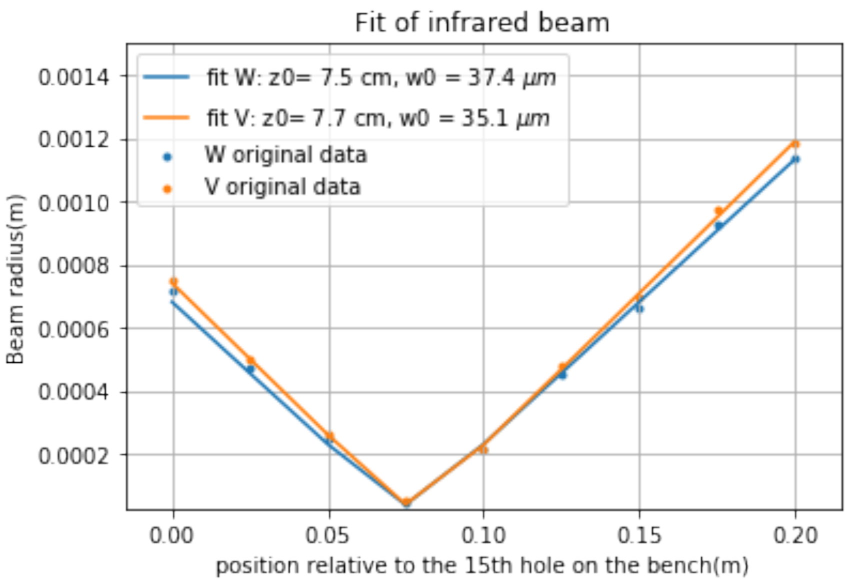
Fit of infrared beam after the first lens on the rail

In [6]:
```
wW2 = np.array([2275, 1845, 1327, 897.9, 433.1, 93.4, 489.7, 936.6,
1427], dtype = np.float)*10**-6/2
wV2 = np.array([2370, 1944, 1397, 950.4, 420.6, 105.2, 521.0, 1003,
1494], dtype = np.float)*10**-6/2
wW21 = np.flipud(wW2)
wV21 = np.flipud(wV2)
x2 = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])*2.5*1e-2
l = 1064e-9
w2 = 50e-6
z2 = 0.07
y2 = w2*np.sqrt(1+((x2-z2)/(np.pi*w2**2/l))**2)
plt.scatter(x2, wW21)
plt.plot(x2, y2)
#plt.ylim(0.0014, 0.002)
#print(np.shape(wW))
```

Out[6]: [<matplotlib.lines.Line2D at 0x1020a88f60>]



In [7]:
```
popt22, pcov22 = curve_fit(gaussian, x2, wW21, bounds=([0.05, 30e-6
], [0.09, 50e-6]))
popt21, pcov21 = curve_fit(gaussian, x2, wV21, bounds=([0.05, 30e-6
], [0.09, 50e-6]))
plt.scatter(x2, wW21, marker=".", label='W original data')
plt.scatter(x2, wV21, marker=".", label='V original data')
plt.plot(x2, gaussian(x2, *popt22),   label='fit W: z0=%4.1f cm, w0
= %4.1f $\mu m$' % tuple(popt22*[100, 1000000]))
plt.plot(x2, gaussian(x2, *popt21),   label='fit V: z0=%4.1f cm, w0
= %4.1f $\mu m$' % tuple(popt21*[100, 1000000]))
plt.xlabel('position relative to the 15th hole on the bench(m)')
plt.ylabel('Beam radius(m)')
plt.title('Fit of infrared beam')
#plt.xlim([0.184,0.1901])
plt.ylim([0.000025, 0.0015])
plt.legend()
plt.grid()
plt.show()

# residual sum of squares
ss_res = np.sum((x - gaussian(x, *popt)) ** 2)

# total sum of squares
ss_tot = np.sum((x - np.mean(gaussian(x, *popt))) ** 2)

# r-squared
r2 = 1 - (ss_res / ss_tot)

#print(r2)
```



Design of telescope going to IR mode-cleaner

1. The calculation of beam waist in IR mode-cleaner

The RoC of gaussian beam is

$$R = z[1 + (\frac{z_R}{z})^2]$$

And rayleigh range is

$$z_R = \frac{\pi \omega_0^2}{\lambda}$$

Then we can get

$$\omega_0 = \sqrt{\frac{\lambda \sqrt{Rz - z^2}}{\pi}}$$

We know from the design of mode-cleaner that RoC of MC's end mirror is 1m, the distance from end mirror to beam waist is $\sqrt{281.22^2 + 11^2} = 281.435mm$. Then we take in all these values, we can get

$$\omega_0 = np.sqrt(1064e - 9 * np.sqrt(0.281435 - 0.281435 * *2)/np.pi) = 390\mu m$$

I counted holes on the lastest optical layout, the distance from the first lens to MC's waist is 20.5 holes. It means distance is 20.5 * 2.5 = 0.5125m.

In [8]:
```
np.sqrt(1064e-9*np.sqrt(0.281435-0.281435**2)/np.pi)
```

Out[8]:
```
0.00039026258705858187
```

2. The simulation of JamMt

Since we know the initial beam is $\omega_0 = 36\mu m, z_0 = 0.076m$ and target beam is $\omega_0 = 390\mu m, z_0 = 0.5125m$.

So we can use JamMt to find good combanition of lenses. According to the lenses we have, I found quite a lot of combinations. Among them I choose one, which is not overlaping with any mirrors or existing lens.

In [15]:
```
PATH = "/Users/ihong/Desktop/work201809/20180905_infrared_character
ization_go_to_mode_cleaner/1.png"
Image(filename = PATH, width=1000, height=1000)
```

Out[15]:
```
Nr. 10 : f=75 mm @ z=0.178, f=-150 mm @ z=0.324,  (v=99.166 %)
Nr. 11 : f=75 mm @ z=0.177, f=-150 mm @ z=0.336,  (v=99.879 %)
Nr. 12 : f=75 mm @ z=0.176, f=-150 mm @ z=0.35,  (v=99.915 %)
Nr. 13 : f=75 mm @ z=0.175, f=-150 mm @ z=0.361,  (v=99.347 %)
Nr. 14 : f=50.2 mm @ z=0.142, f=-200 mm @ z=0.193,  (v=99.669 %)
Nr. 15 : f=62.9 mm @ z=0.157, f=-200 mm @ z=0.262,  (v=99.326 %)
Nr. 16 : f=62.9 mm @ z=0.157, f=-200 mm @ z=0.272,  (v=99.903 %)
Nr. 17 : f=75 mm @ z=0.172, f=-200 mm @ z=0.342,  (v=99.163 %)
```