# Radiant Valve and Damper Tuning: Ziegler-Nicholas Method


Author

YoungWoong Cho

*The Cooper Union*


December 2, 2020

# Contents

**1 ZN Tuning: Open Loop Tuning** **1**
1.1 System description . . . . . 1
1.2 Tuning method . . . . . 1
1.3 Open loop reaction curve . . . . . 1
1.4 Calculation of tuning parameters . . . . . 2
1.5 Determination of the gains . . . . . 4
1.6 Responses with different sets of gains . . . . . 4
1.7 Proposed gains . . . . . 5

**2 ZN Tuning: Closed Loop Tuning** **6**
2.1 System description . . . . . 6
2.2 Tuning method . . . . . 6
2.3 Closed loop reaction curve . . . . . 6
2.4 Calculation of tuning parameters . . . . . 6
2.5 Determination of the gains . . . . . 7
2.6 Responses with different sets of gains . . . . . 7
2.7 Proposed gains . . . . . 8

# 1 ZN Tuning: Open Loop Tuning

## 1.1 System description

Temperature reaction curves of room 719 and 619 were analyzed for tuning. The input of the system is HW valve, and the output of the system is the temperature in degree Fahrenheit.

## 1.2 Tuning method

Zeigler-Nicholas tuning method for an open loop response curve was used to tune the system. The proposed strategy for the calculation of PID gains are as follows:

| Control Algorithm | $k_p$ | $k_i$ | $k_d$ |
|---|---|---|---|
| P | $\frac{\Delta u}{NL}$ | - | - |
| PI | $0.9\frac{\Delta u}{NL}$ | $0.3\frac{k_p}{L}$ | - |
| PID | $1.2\frac{\Delta u}{NL}$ | $0.5\frac{k_p}{L}$ | $\frac{k_p L}{2}$ |

with

| | |
|---|---|
| $\Delta u$ : | %change in input |
| N: | $\frac{\Delta\%PV}{\Delta t}$ |
| L: | Time lag until the change in temperature is sensed |

## 1.3 Open loop reaction curve

The reaction curve of the system is plotted below. Note that the horizontal axis represents the elapsed time in total seconds.

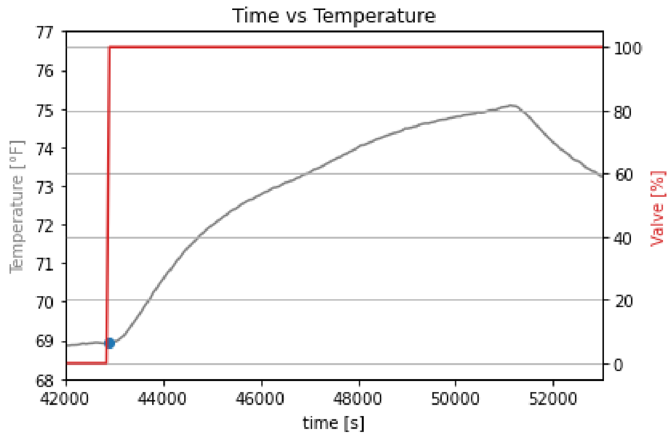


Figure 1: Open Loop reaction curve

## 1.4 Calculation of tuning parameters

In order to calculate the time lag, a line that is tangent to and passes through the inflection point had to be calculated. First, the time derivative of the reaction curve was taken twice to obtain the inflection point, i.e. time when the second derivative becomes zero. Next, the corresponding slope at that time is obtained from the first derivative plot. Finally, the tangent line is drawn, whose intersections with the initial temperature line and the final temperature line were calculated.

Figure below describes how the calculations are made. Note that the horizontal axis represents time in seconds, whose values are obtained by converting the `datetime` format into `timedelta.total_seconds()`. The plot at the top shows the first derivative of the response curve, which is filtered to reduce the sensor noise. The plot in the middle shows the second derivative of the filtered response curve, which is again filtered to reduce the noise. This plot is used to find out the inflection time, i.e. where the second derivative becomes zero. Inflection points are marked as green dots in the first and the second plots. Finally, the plot at the bottom indicates several points of interest: the inflection point(red), the intersection between the line tangent to the inflection point and the initial temperature(green), the intersection between the tangent line and the final temperature(purple), and the point where the actuator begins the input(orange), all of which are used to calculate $N$ and $L$.

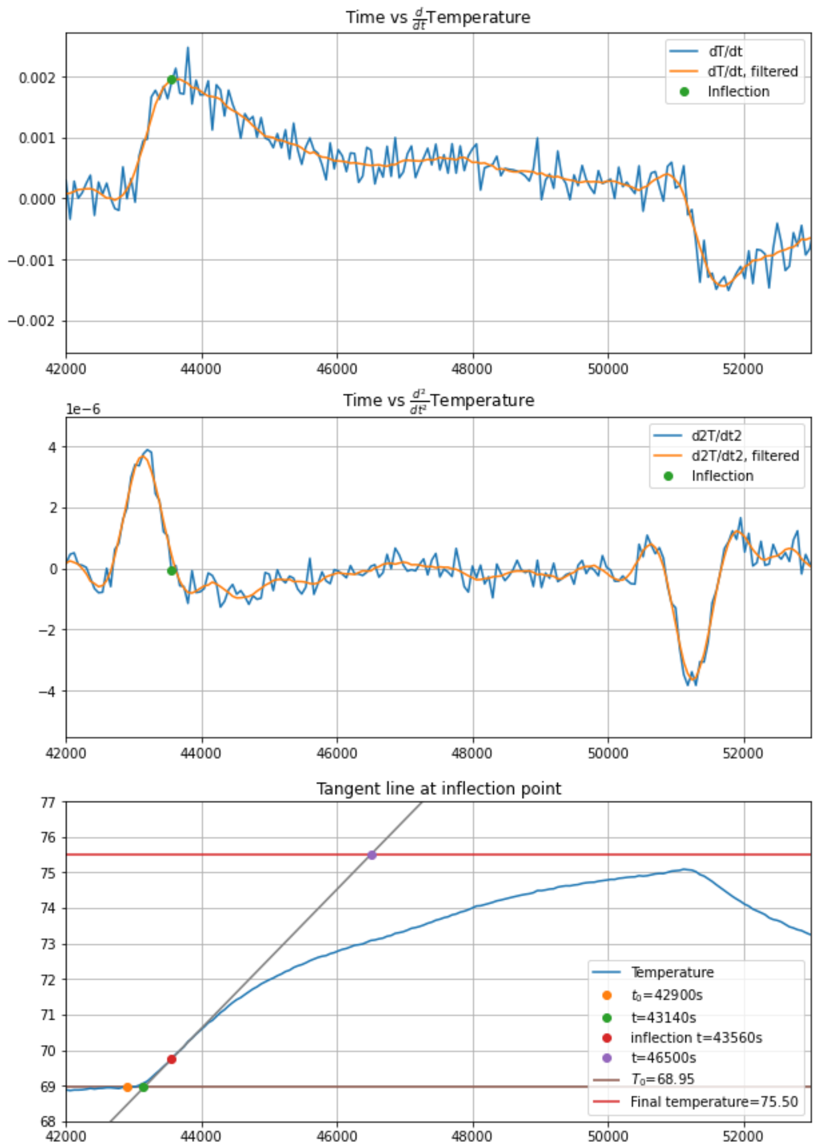


Figure 2: ZN tuning of open loop reaction curve

The noise of the derivative plots were reduced by using the Savitzky–Golay filter as follows:

```
dTdt = (T[1:]-T[:-1])/(t[1]-t[0]); dTdt_hat = scipy.signal.savgol_filter(dTdt, 51, 10)
```

Each orange curve in the first and the second plots represents the filtered curves. Also, since the system did not ever reach the final temperature due to the CHW, the final temperature was approximated to be $75.5°F$, which is plotted in red. This approximation can be a source of an error, but is still a reasonable guess to provide a good starting point for the tuning.

Since the input was increased from 0% to 100%, $\Delta u$ is set to be 100%. The time lag $L$ between the time when the input was initiated(plotted in orange) and the intersection between the $T_0$ line and the tangent line(plotted in green) is found to be 240 seconds, or 4 minutes. Lastly, the value of $N$ was found to be $0.117211\%/min$by calculating the slope of the reaction curve at the inflection point.

### 1.5 Determination of the gains

The tuned gains for PI and PID controllers are tabulated below.

| Control Algorithm | $k_p$ | $k_i$ | $k_d$ |
|---|---|---|---|
| P | 213.290 | - | - |
| PI | 191.961 | 14.397 | - |
| PID | 255.948 | 31.993 | 511.90 |

### 1.6 Responses with different sets of gains

The actual response curves of the system with different sets of gains were analyzed to validate the values of the gains calculated above. Figure 3 shows the responses of the system with various sets of gains.
The plot at the top is a response curve of the system with $k_p = 1750$, $k_i = 52.5$, and $k_d = 14000$. It can be seen from the plot that the significant overshoot is present, which is due to high $k_p$ gain and large error at the beginning(notice that the difference between the setpoint and the actual temperature of this plot is the largest among three plots). Also, high $k_d$ gain results in quick drop of HW VLV when the temperature exceeds the setpoint, which is a desirable behavior to reduce the effect of overshoot.
The plot at the middle is a response curve of the system with $k_p = 333$, $k_i = 3$, and $k_d = 0$. Although $k_p$ is smaller than the first response, the response still displays some overshoot, which implies that smaller $k_p$ is still desired. Adding some $k_d$ can help reduce the overshoot by slowing down the temperature as it approaches the setpoint quickly.
The plot at the bottom is a response curve of the system with $k_p = 2750$, $k_i = 52.5$, and $k_d = 0$. As pointed out above, $k_p$ of 2750 is too large so that it takes some time until the oscillatory behavior is diminished. This controller can potentially be improved by reducing $k_p$, and introducing some amount of $k_d$ that can reduce the overshoot.

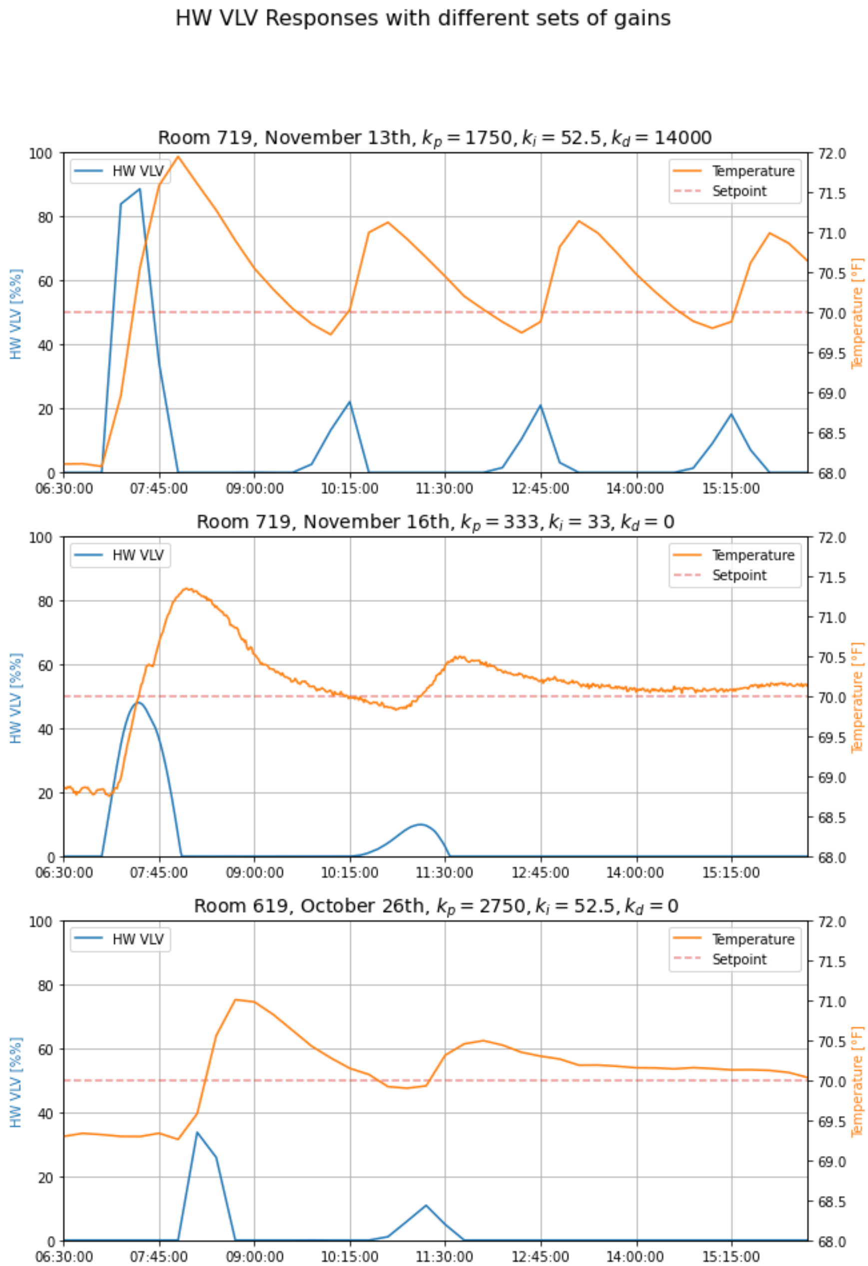


Figure 3: Responses of HW and temperature with various gains

## 1.7 Proposed gains

From the result of ZN tuning method and the actual response curves, following set of gains is expected to generate a system that is both responsive and stable:

$$k_p : 240$$

$$k_i : 30$$

$$k_d : 500$$

# 2 ZN Tuning: Closed Loop Tuning

## 2.1 System description

CFM responses from room 719 were analyzed for tuning. The input of the system is a damper, and the output of the system is the CFM.

## 2.2 Tuning method

Zeigler-Nicholas tuning method for a closed loop response curve was used to tune the system. The proposed values of PID gains are as follows:

| Control Algorithm | $k_p$ | $k_i$ | $k_d$ |
|---|---|---|---|
| P | $0.5k_{p,cr}$ | - | - |
| PI | $0.45k_{p,cr}$ | $1.2\frac{k_p}{T_{cr}}$ | - |
| PID | $0.6k_{p,cr}$ | $\frac{2k_p}{T_{cr}}$ | $\frac{k_pT_{cr}}{8}$ |

with

$k_{cr}$: $k_p$ at the neutrally stable reaction

$T_{cr}$: Period of the sinusoidal reaction curve

## 2.3 Closed loop reaction curve

The neutrally stable oscillatory behavior could be observed for $k_{p,crit} = 160$. The reaction curve of the system is plotted below. Note that the horizontal axis represents the elapsed time in total seconds, same as before.

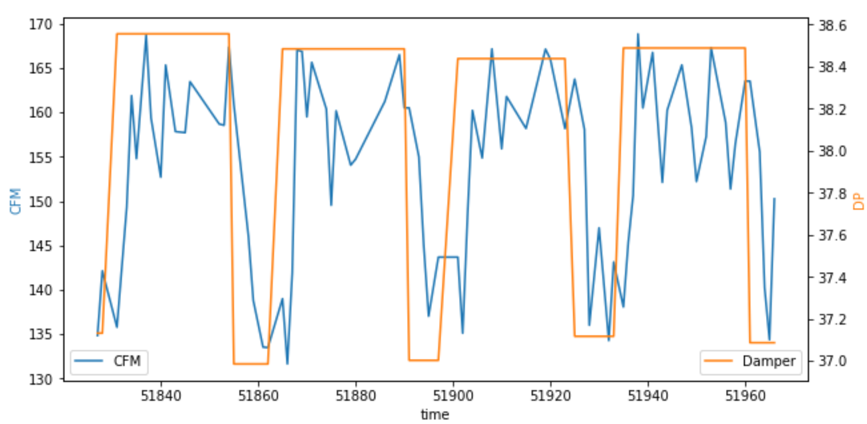


Figure 4: Closed Loop reaction curve

## 2.4 Calculation of tuning parameters

In order to calculate $T_{cr}$, a since curve was fitted to the reaction curve. This was done by utilizing the optimizing method from `scipy` package as follows.

```
# fit sine curve
def sin(t, A, w, phi, y0):
```

```
    return A*np.sin(w*(t-phi)) + y0
param, _ = curve_fit(sin, t, CFM, p0=np.array([17, 2*np.pi/40, np.pi/3, 130]))
CFM_fit = param[0]*np.sin(param[1]*(t-param[2])) + param[3]
```

A sine curve with optimal amplitude, frequency, phase shift and y-intercept was found by internal error-minimizing feature. The fitted curve is plotted below.

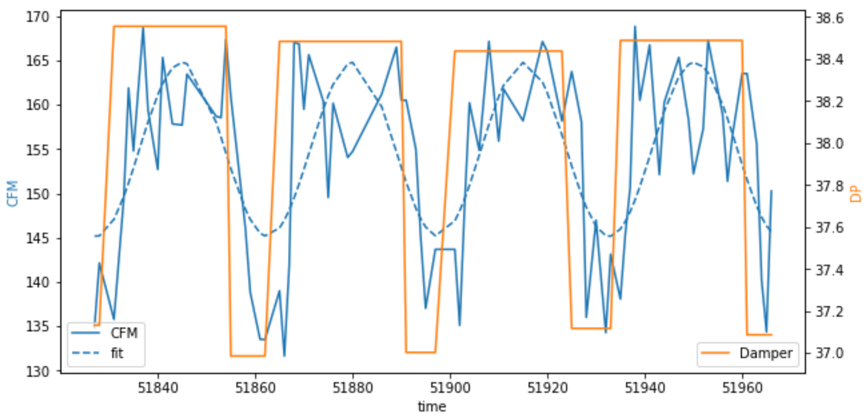


Figure 5: ZN tuning of closed loop reaction curve

Dashed line represents the fitted line. From the fitted sine curve, $T_{cr}$ is found to be 35.1s.

### 2.5 Determination of the gains

The tuned gains for PI and PID controllers are tabulated below.

| Control Algorithm | $k_p$ | $k_i$ | $k_d$ |
|---|---|---|---|
| P | 80.00 | - | - |
| PI | 72.00 | 2.463 | - |
| PID | 96.00 | 5.472 | 421.02 |

### 2.6 Responses with different sets of gains

The actual response curves of the system with different sets of gains were analyzed to validate the values of the gains calculated above. Figure 6 shows the responses of the system with various sets of gains. All of the plots show the responses of the system for the same time intervals as the setpoint changes from 40 to 120.
The plot at the top left is a response curve of the system with $k_p = 76.7$, $k_i = 7.43$, and $k_d = 29.7$. This system has a huge overshoot at the beginning, which is due to high $k_p$ gain and relatively small $k_d$ gain. It is expected that smaller $k_p$ can reduce the magnitude of the overshoot, and higher $k_d$ can increase the damping effect slightly, which will allow the system to achieve the steady state in less time.
The plot at the top right is a response curve of the system with $k_p = 50$, $k_i = 3$, and $k_d = 40$. The $k_p$ value is decreased down to 50, which results in the system that takes significantly less time to achieve the setpoint. A cotroller with slightly smaller $k_p$ and slightly larger $k_d$ is expected to further reduce the effect of overshoot. This is the best performing controller among four.
The plot at the bottom left is a response curve of the system with $k_p = 80$, $k_i = 3$, and $k_d = 0$. Large $k_p$ value resulted in significant overshoot, and the absence of $k_d$ produced unnecessary oscillation of the response.
The plot at the bottom right is a response curve of the system with $k_p = 20$, $k_i = 3$, and $k_d = 0$. This is the

system with the smallest $k_p$ among four, and as a result is the most sluggish and has the smallest overshoot in magnitude. This sluggish behavior can be ameliorated by choosing higher $k_p$ and $k_i$ which will increase the natural frequency of the system, and reduce the oscillatory behavior of the system by increasing $k_d$ gradually.

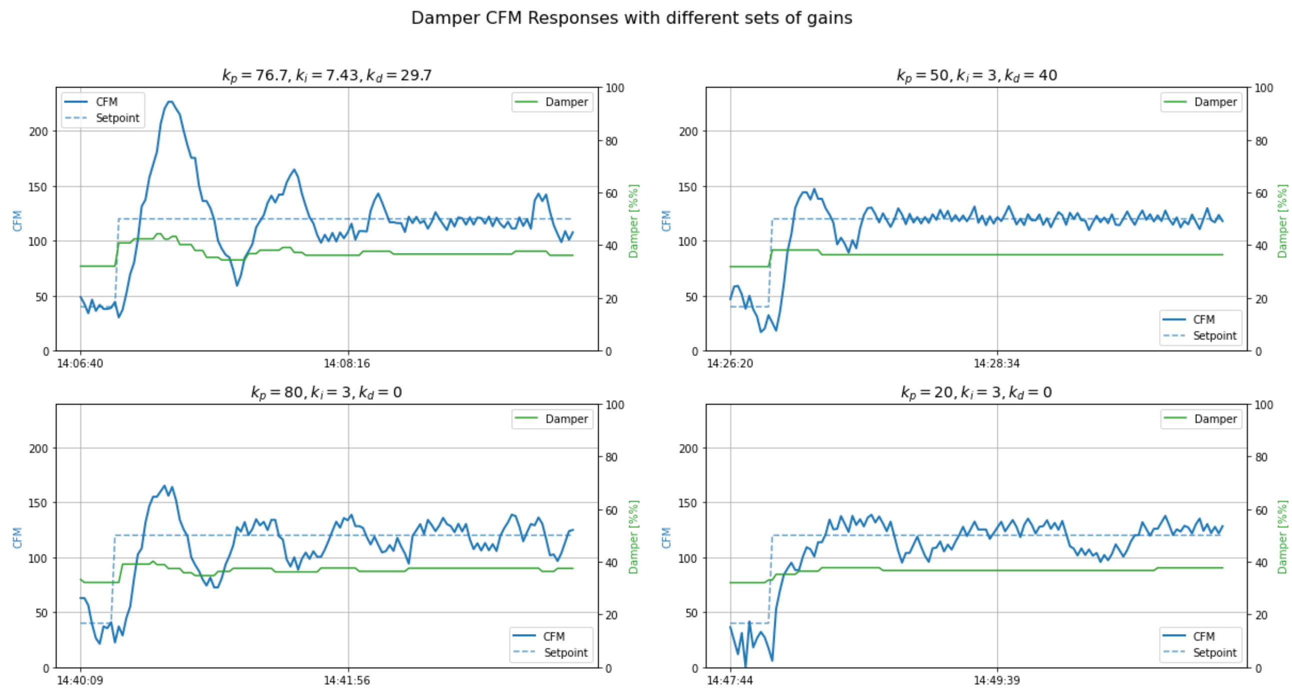


Figure 6: Responses of Damper and CFM with various gains

## 2.7 Proposed gains

From the result of ZN tuning method and the actual response curves, following set of gains is expected to generate a system that is both responsive and stable:

$$k_p : 40$$
$$k_i : 6$$
$$k_d : 100$$