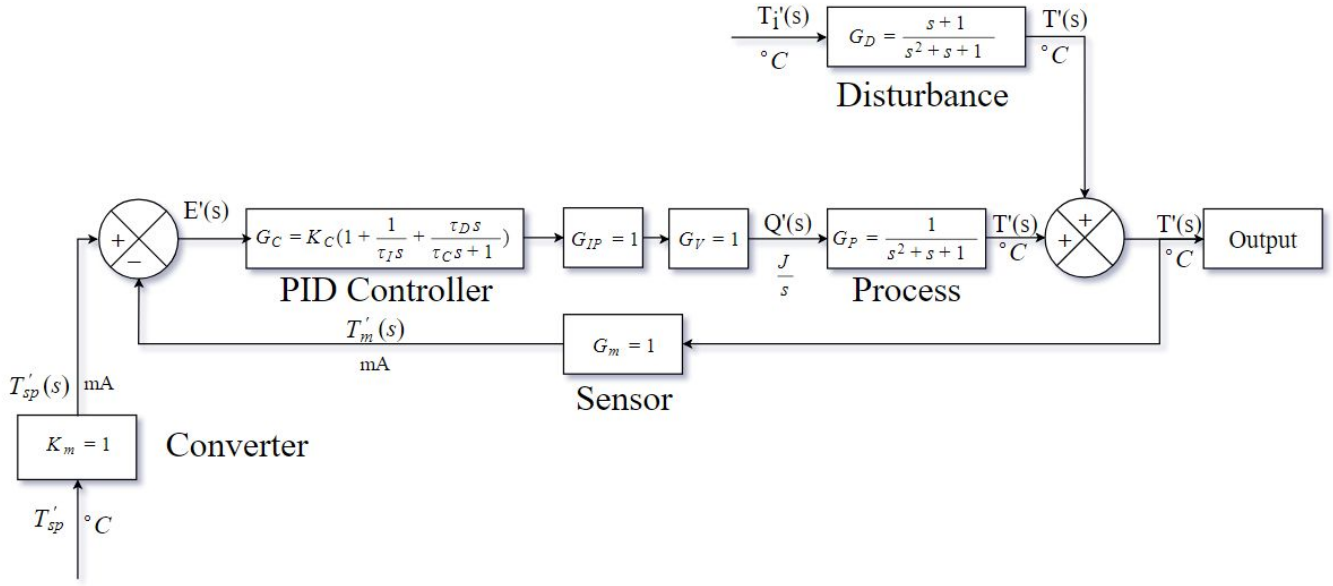
James Cheon 57184146 | Hugo Dignoes 59119164 | Timothy Hillmer 23435167 | Samantha Lee 24999161 | Ariel Zheng 57201148



If the performance metric is defined as a function of the parameters, "P(K_C, τ_I, τ_D,τ_C)," then it is not a smooth one. Any given four-coordinate point is surrounded by countless local minima, such that gradient-based minimization of P changes neither P nor the parameters significantly. The more effective approach is a random search (*random_search.ipynb*, pages A38, A44) where the parameters are varied uniformly on a logarithmic scale and used to evaluate P.

To improve the search, several univariate trials are performed (changing one variable while others are held constant). While P oscillates very rapidly when any parameter is smoothly adjusted, it shows a general positive correlation with K_C (capped at 10000) and a negative correlation with τ_I. The search is narrowed to the most successful regions with respect to each variable, keeping the results of the tests in mind: K_C is constrained to the interval [1000, 10000]; τ_I to [0.0001, 0.001], τ_D to [0.05, 1.5], and τ_C to [0.0005, 0.1]. The narrow region can be searched more thoroughly than a wide region. Lower values of τ_I cannot not be tested because they frequently flag errors in "control.forced_response" due to the controller volatility they cause.

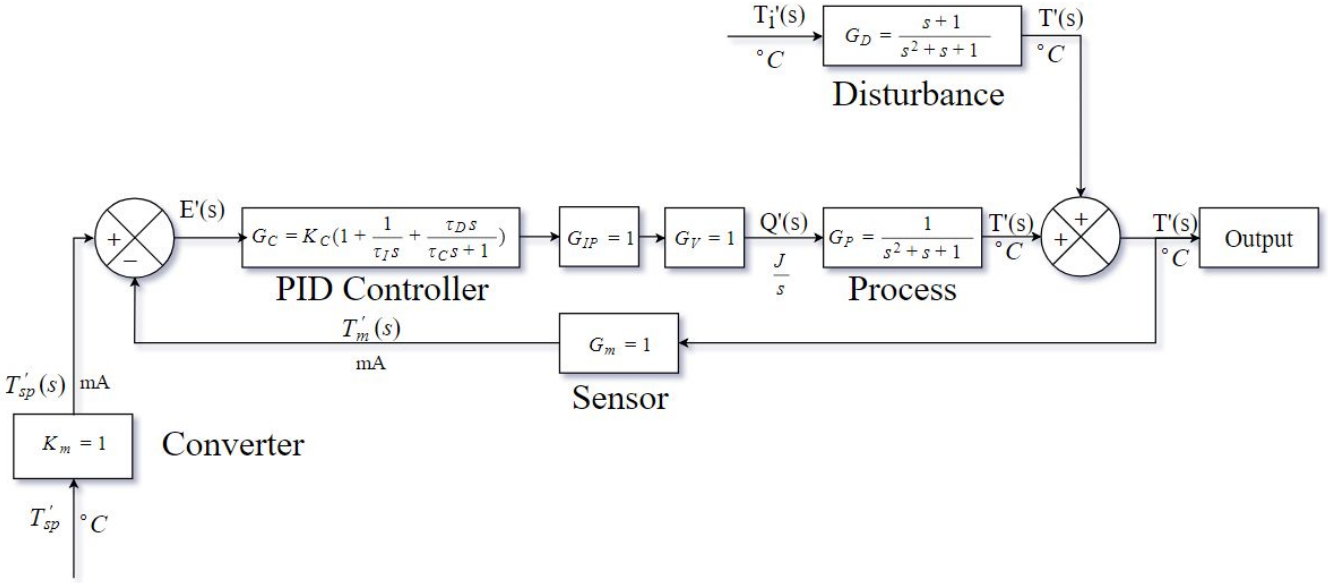
The minimum performance metric obtained is **6.603490145150944** using the parameters (K_C, τ_I, τ_D, τ_C) = (5688.814947611792, 0.0002280222206125743, 0.4941522520819933, 0.006245557968152839).

The simplest way of simulating a controller is with transfer functions: mathematical functions that describe a component within a system. They are commonly defined as the ratio between the output and input of the component in the frequency (s) domain, and are generally obtained by modelling the system, and then using the Laplace Transform to obtain the function in the s-domain. The relevant transfer functions were given in this assignment, as was the disturbance profile used. The inputs were processed into outputs using the G_{Total} function described in A40, and the difference between the setpoint and system output (i.e. T_{SP}-T) was passed through G_C (page A40) to obtain the new input. Python's Control Systems library was used throughout to define and use the transfer functions (pages A40, A41, A44).

The results are checked using the *Verify_(Yankai).ipynb* code (pages A38, A39) to evaluate the performance metric for each disturbance profile using "control.forced_response" and return the average of the one hundred performance metrics. *Skeleton_(Masterplan_for_plots).ipynb* (pages A38, A41) performs the same calculations, plotting the data as it is generated.

Appendix: Figures, Plots, and Codes

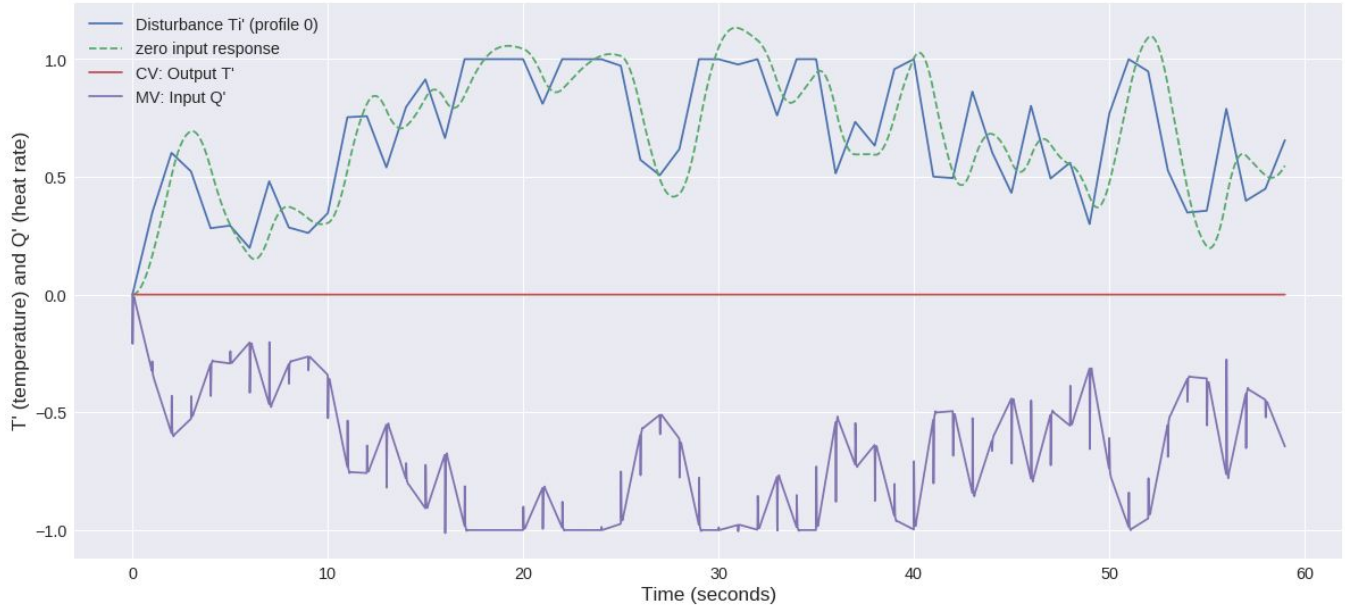
Block Diagram

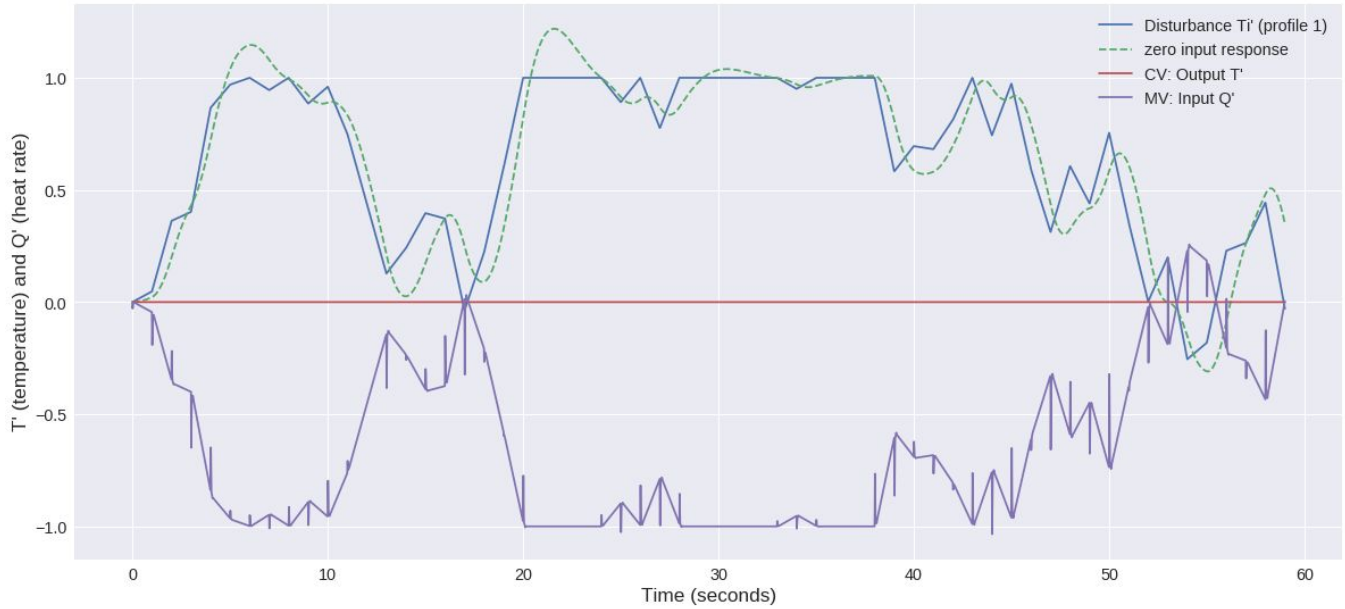


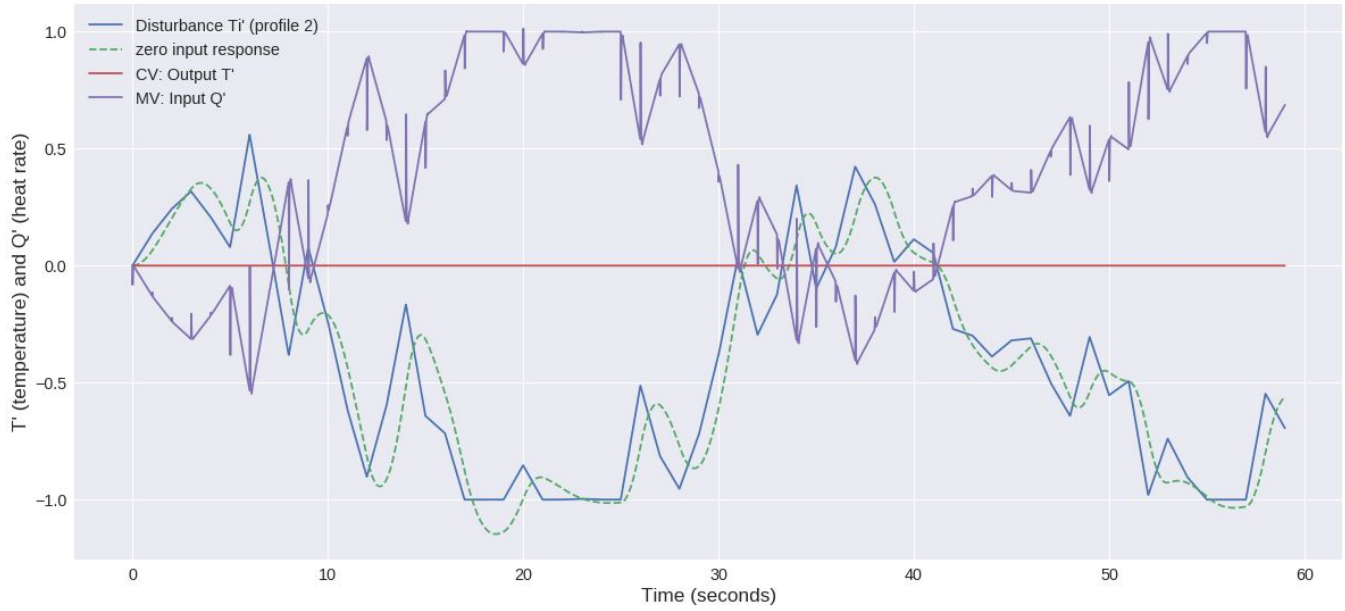
Controlled variable (red) and Manipulated variable (purple) plots:

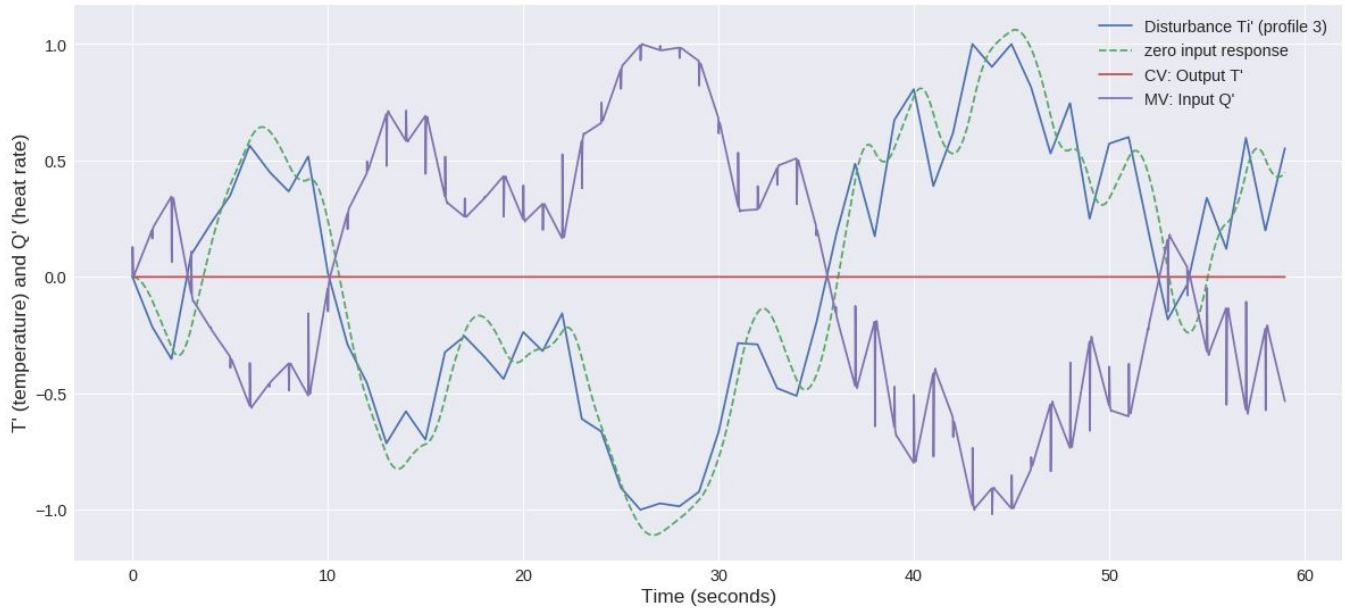
With high KC and low Tau values, the controller theoretically responds very quickly. Because of this, the input mirrors the output quite closely.

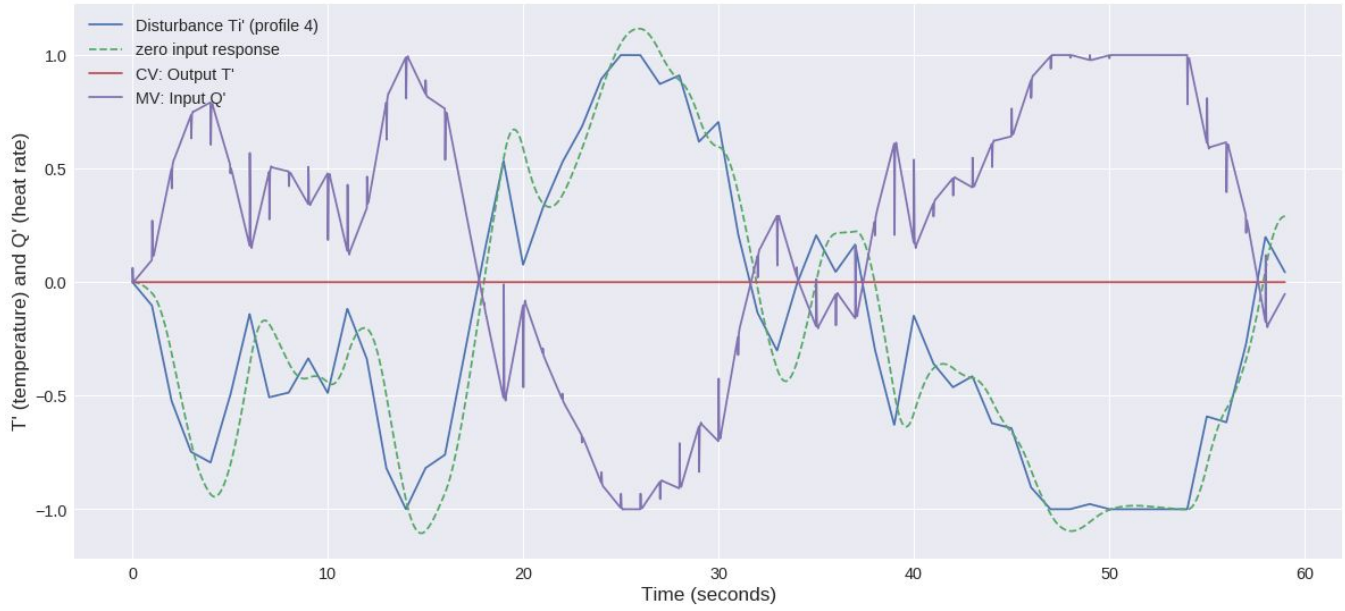
Plots are given in order, starting with Profile 0:

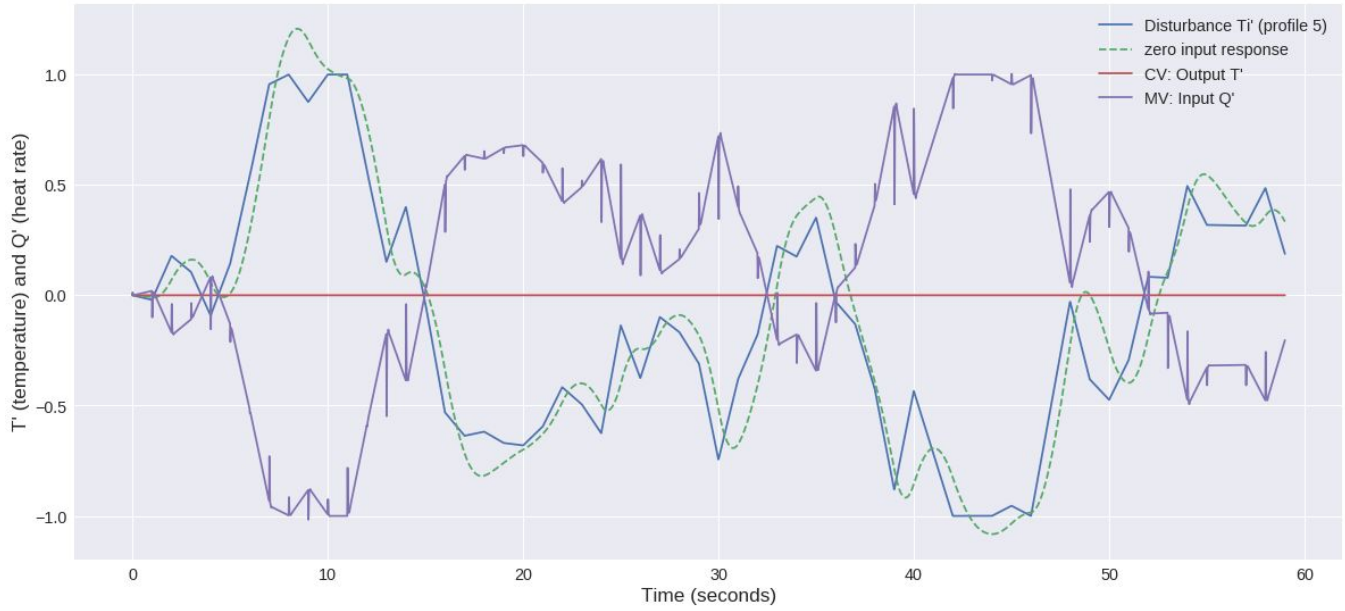


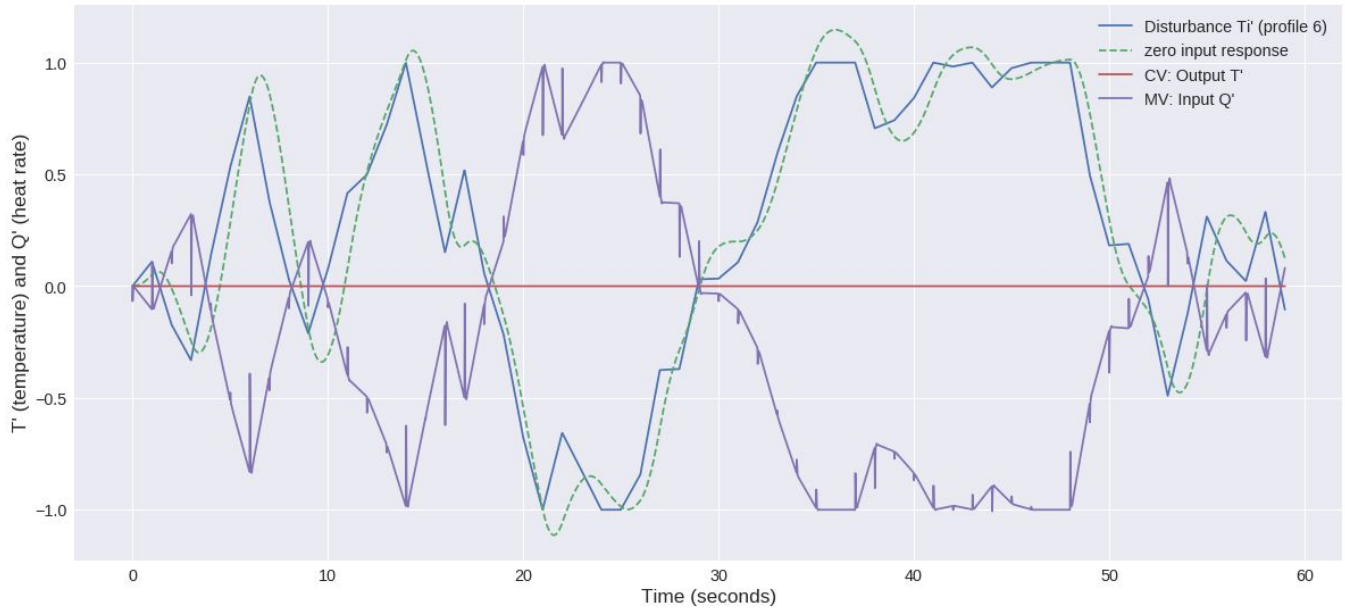


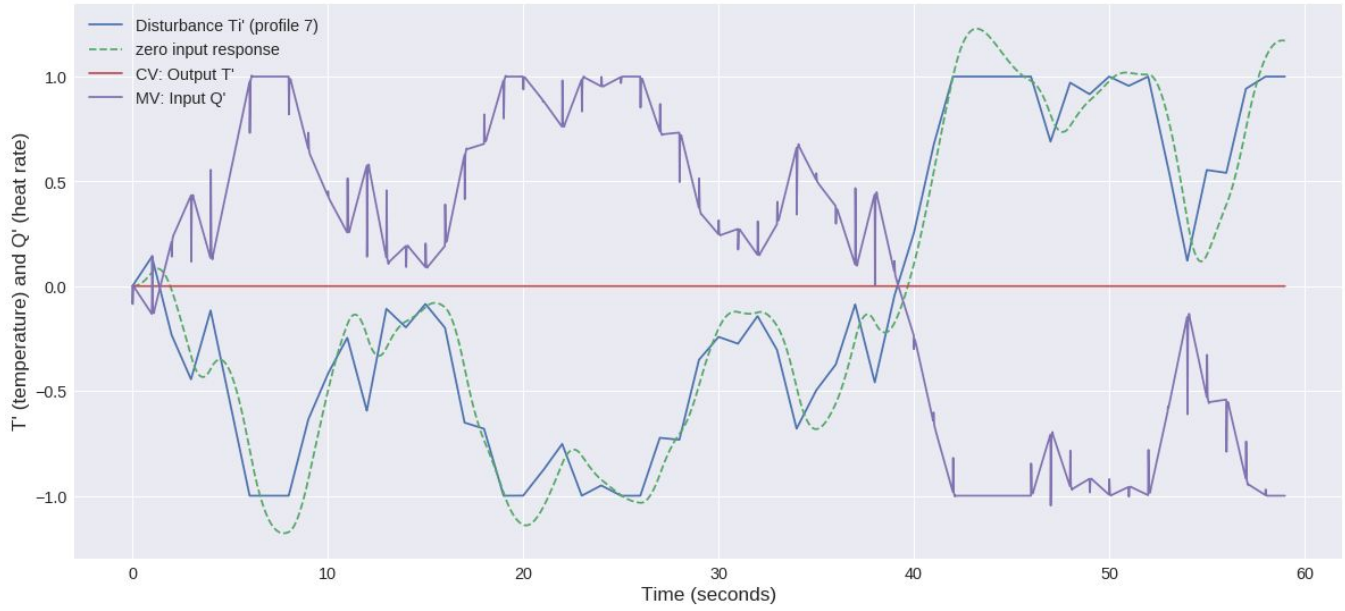

Disturbance Ti' (profile 2)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

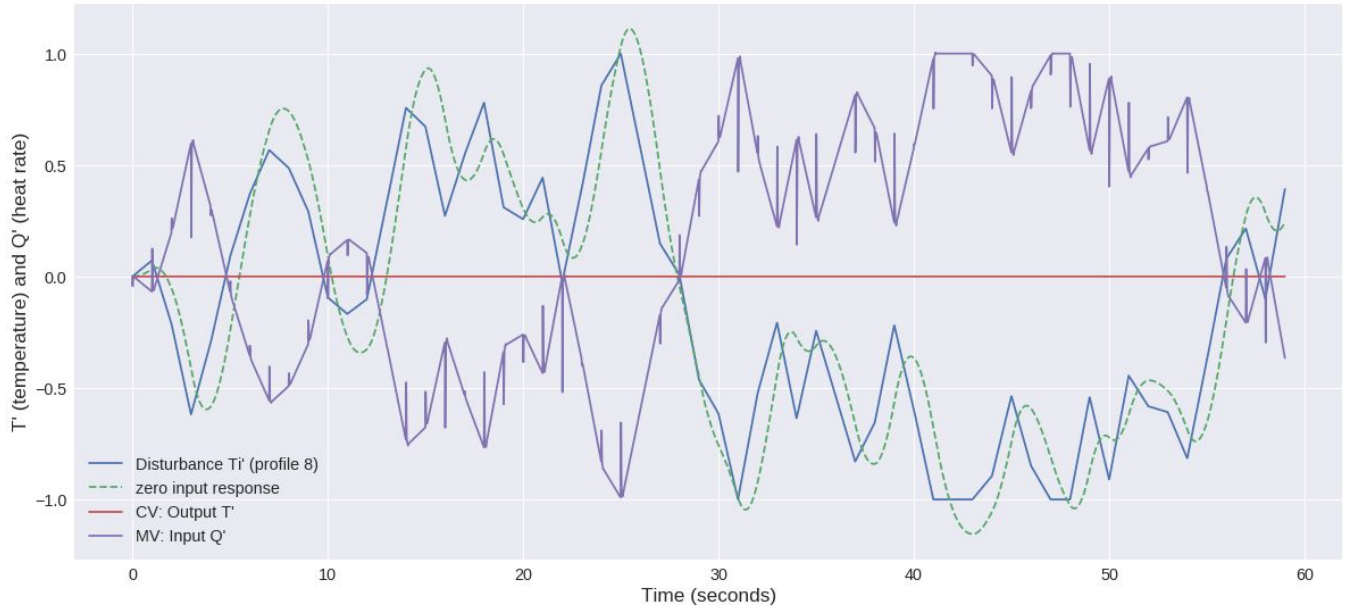

Disturbance Ti' (profile 3)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

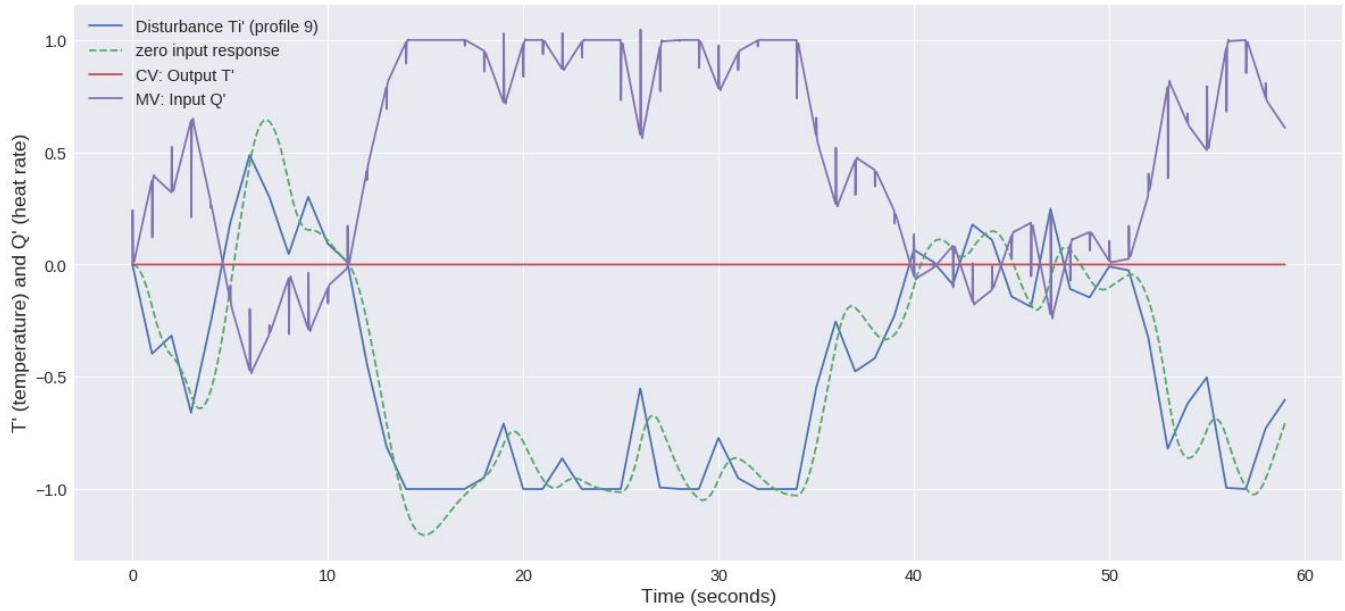

Disturbance Ti' (profile 4)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

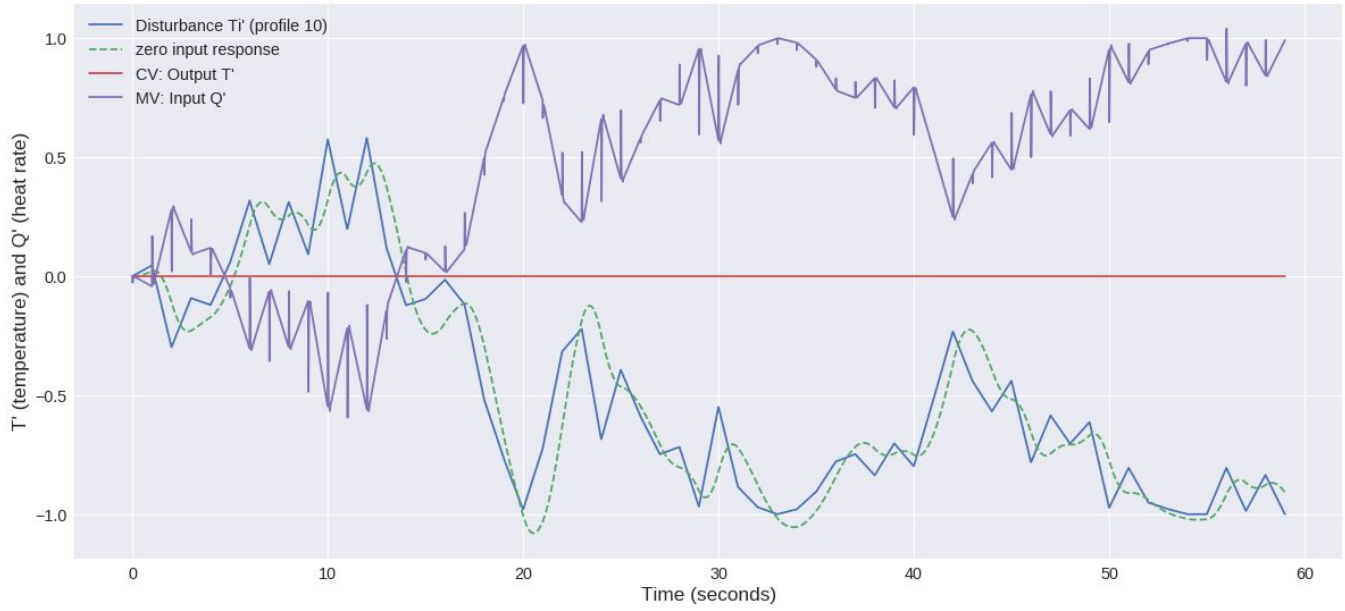

Disturbance Ti' (profile 5)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

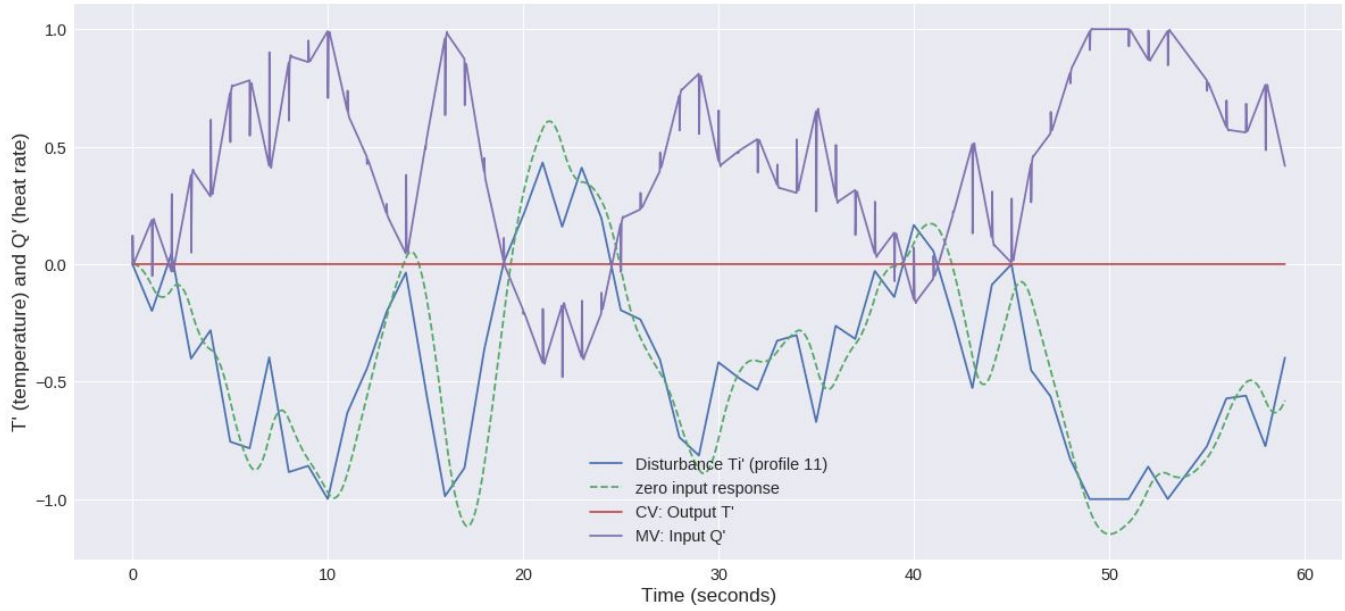

Disturbance Ti' (profile 6)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

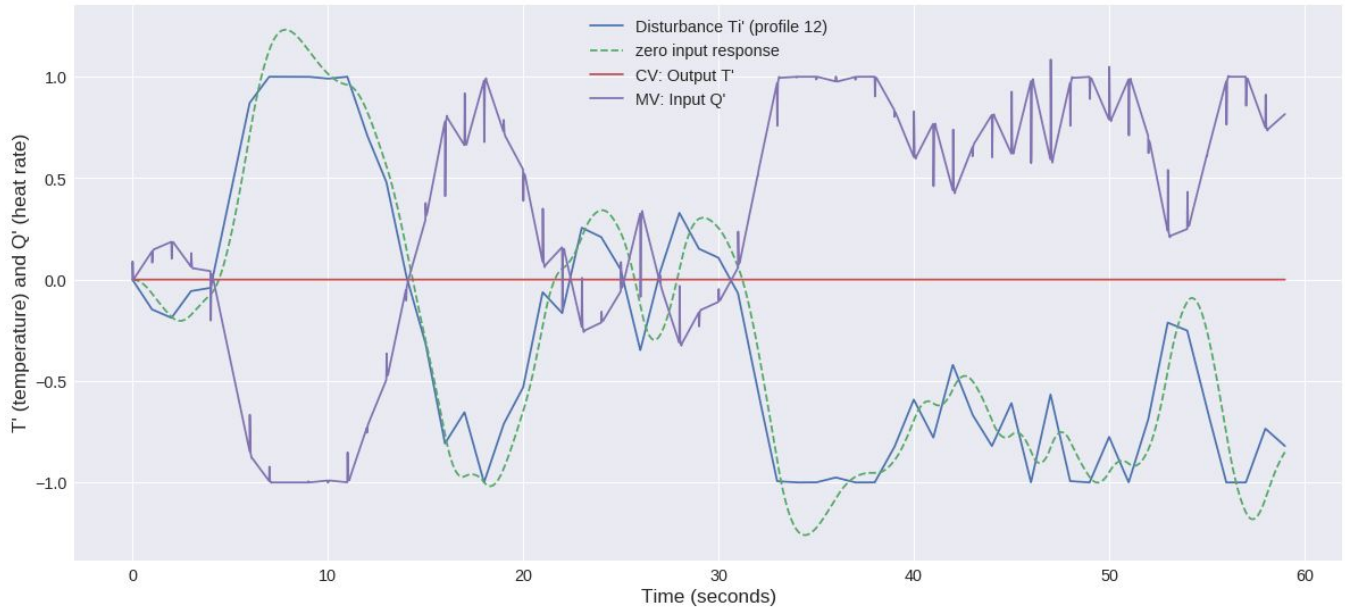

Disturbance Ti' (profile 7)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

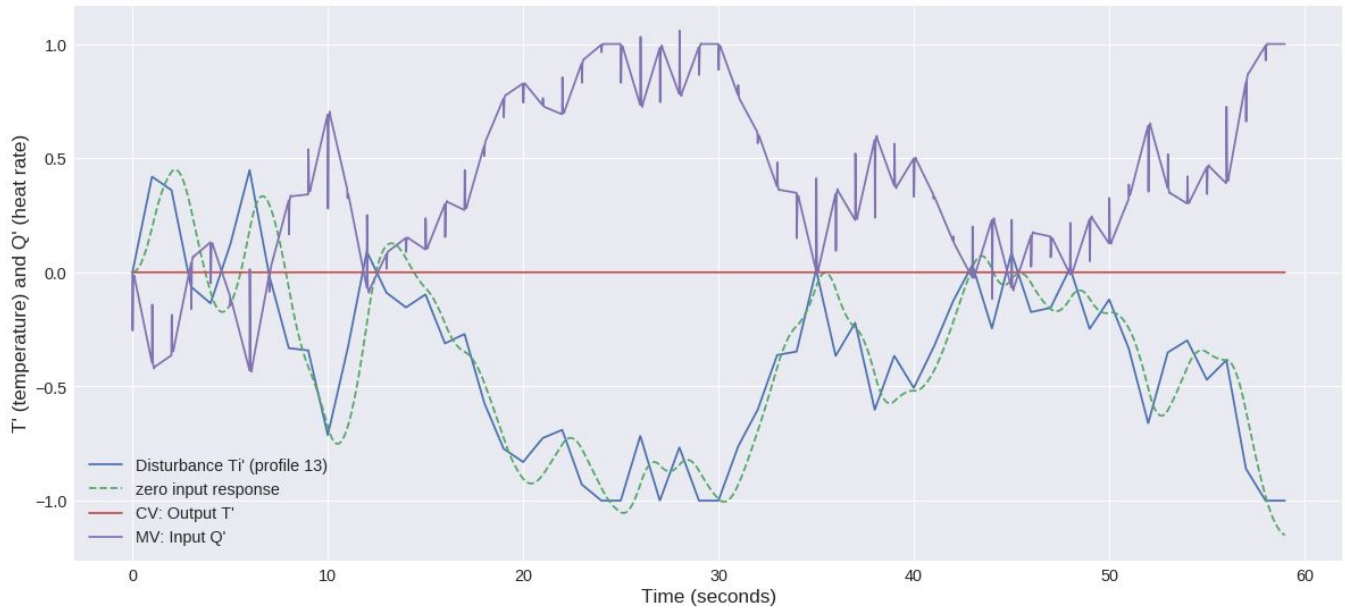

Disturbance Ti' (profile 8)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

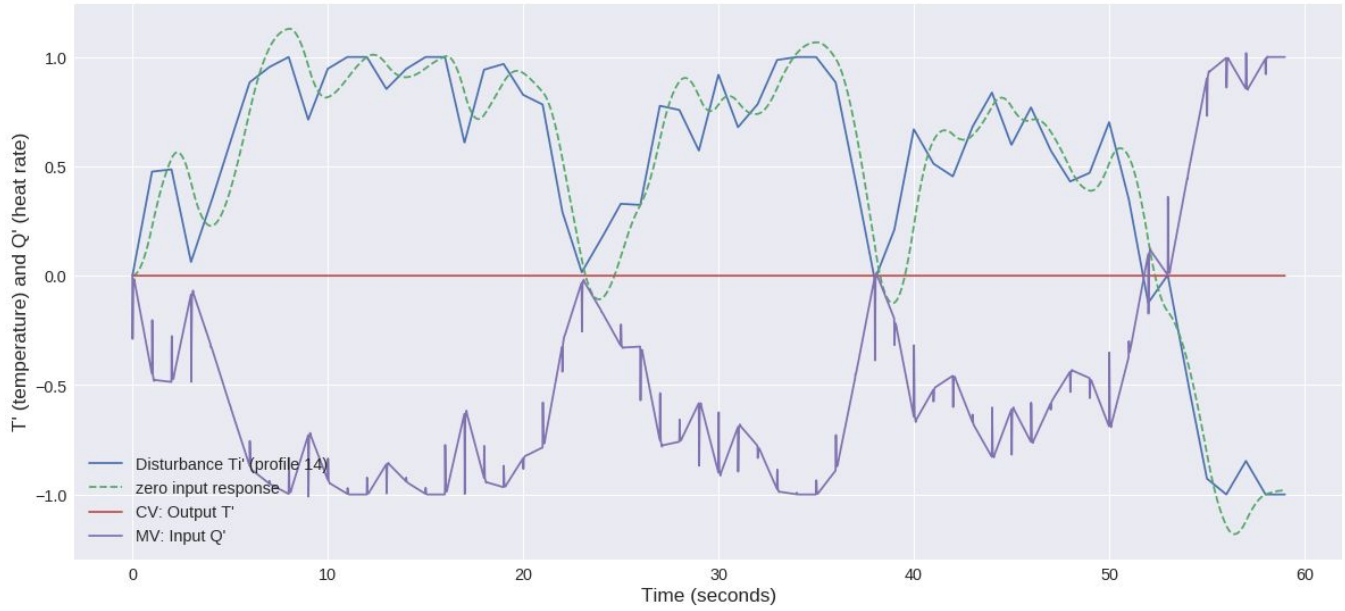

Disturbance Ti' (profile 9)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

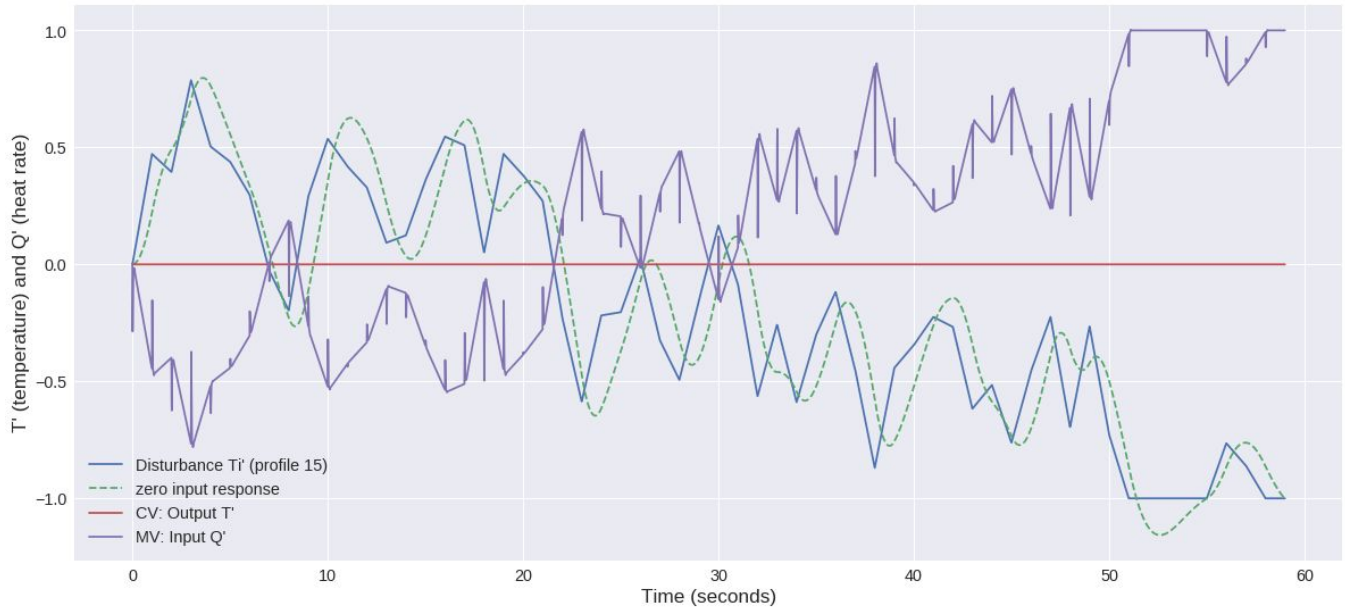

Disturbance Ti' (profile 10)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

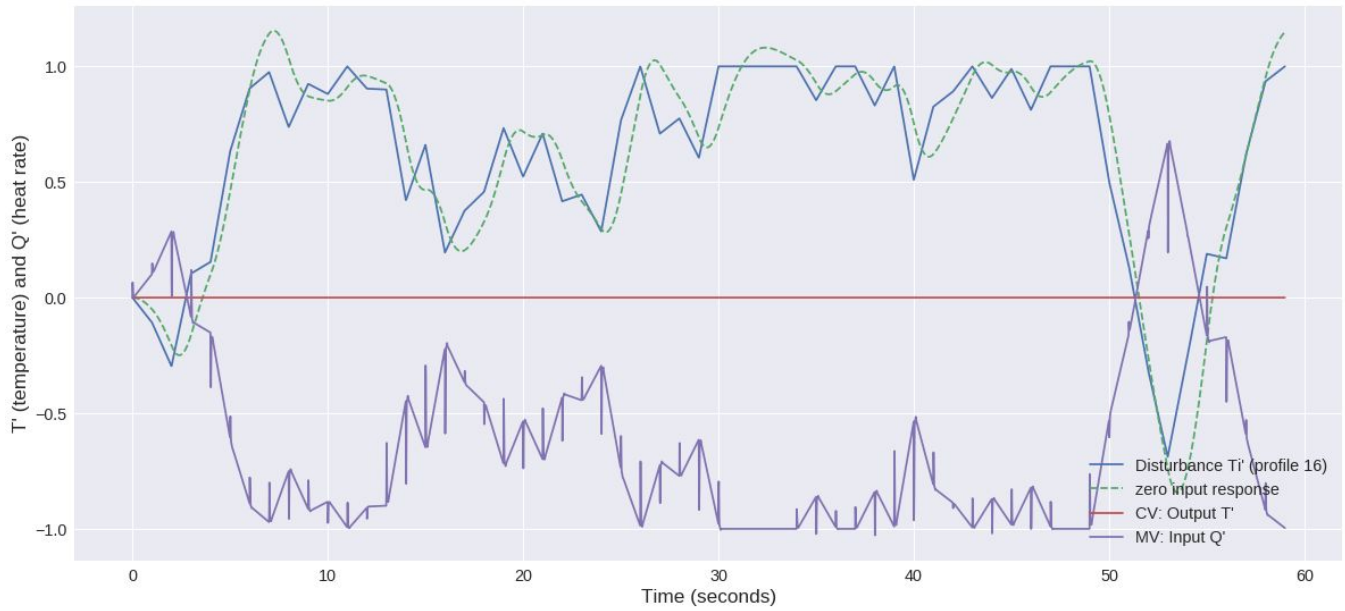

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 11)
zero input response
CV: Output T'
MV: Input Q'

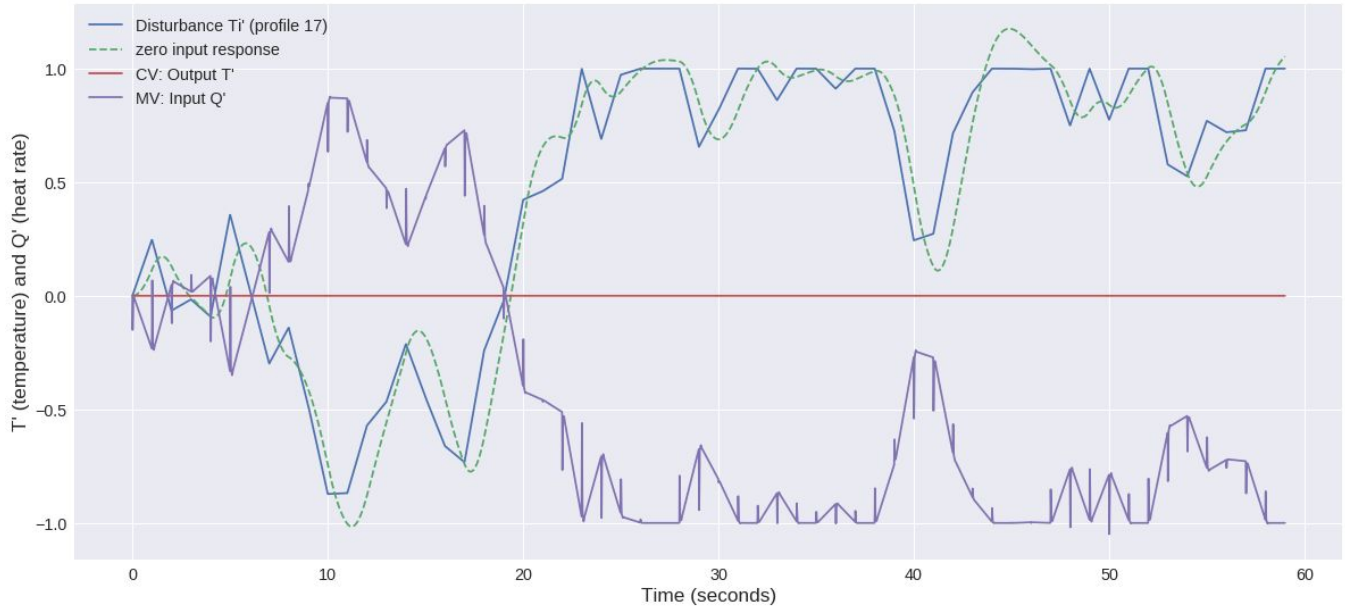

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 12)
zero input response
CV: Output T'
MV: Input Q'

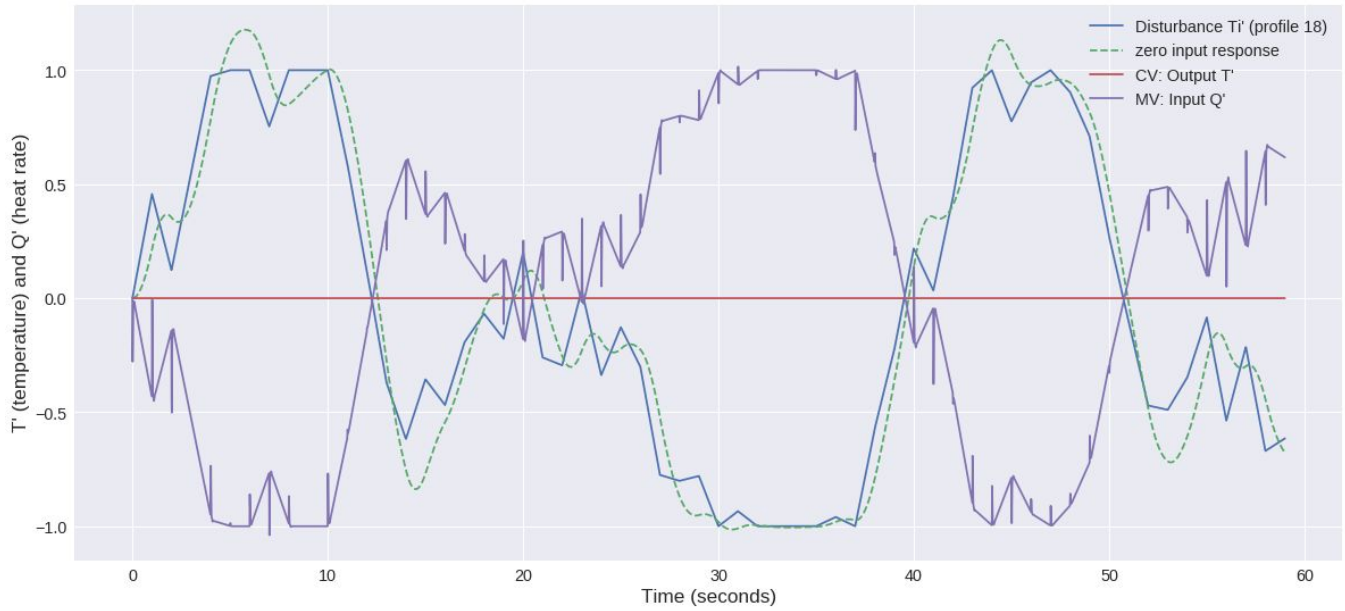

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 13)
zero input response
CV: Output T'
MV: Input Q'

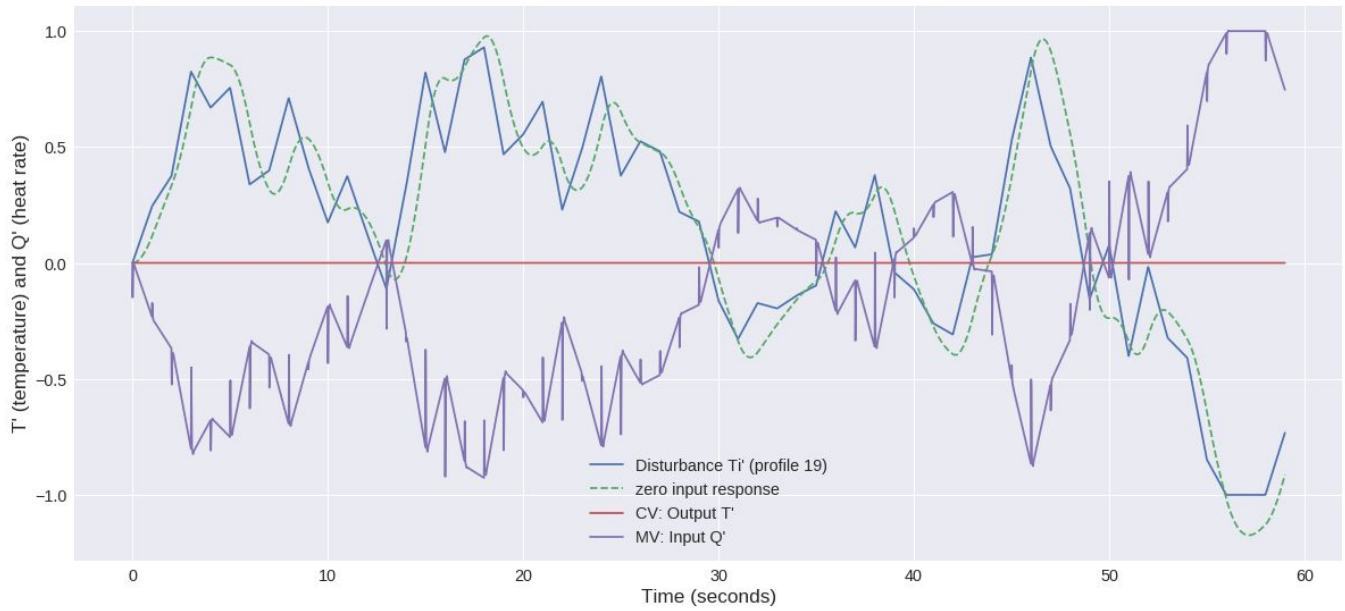

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 14)
zero input response
CV: Output T'
MV: Input Q'

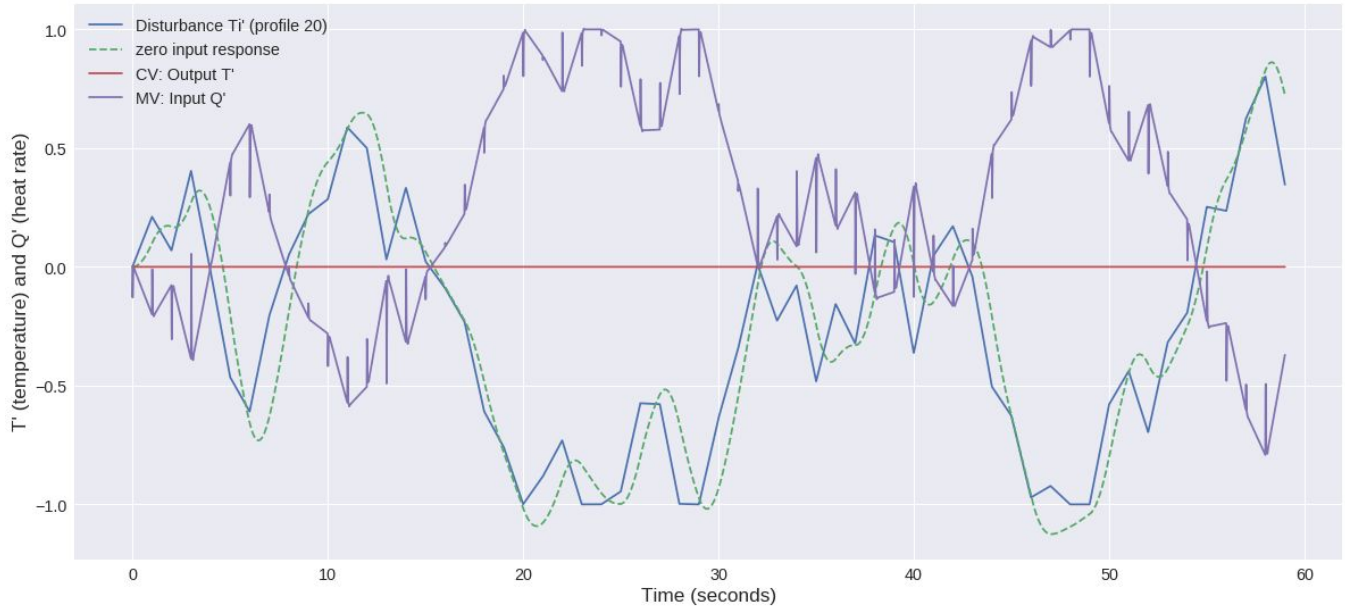

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 15)
zero input response
CV: Output T'
MV: Input Q'

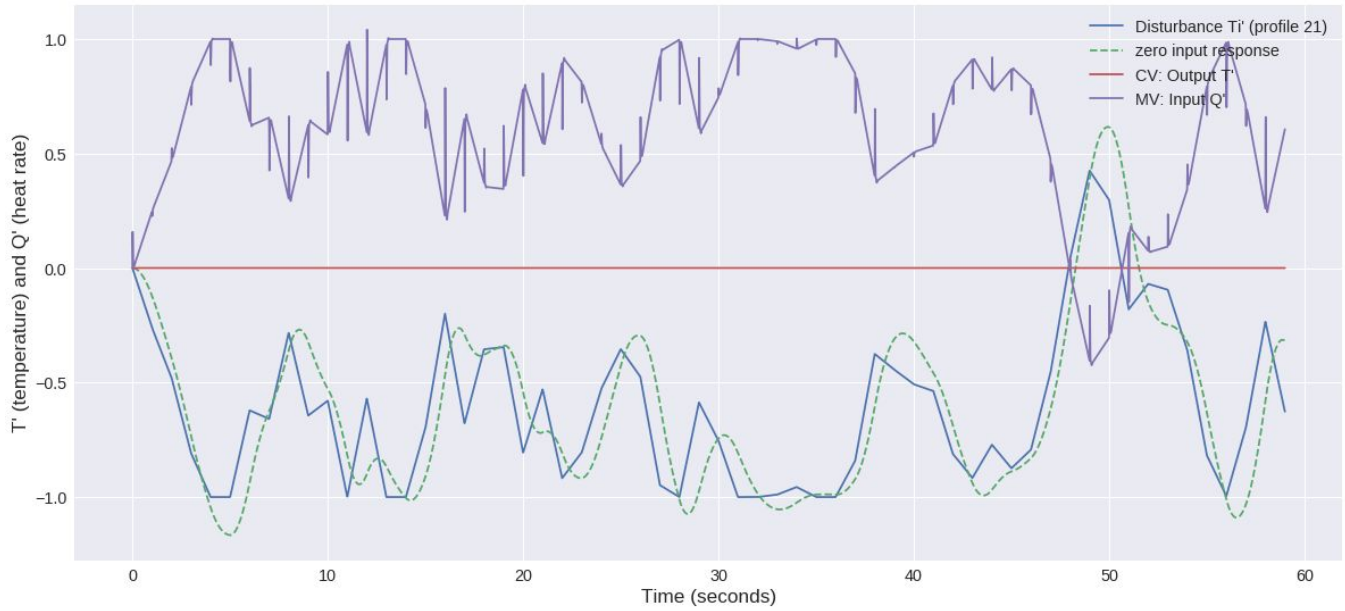

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 16)
zero input response
CV: Output T'
MV: Input Q'

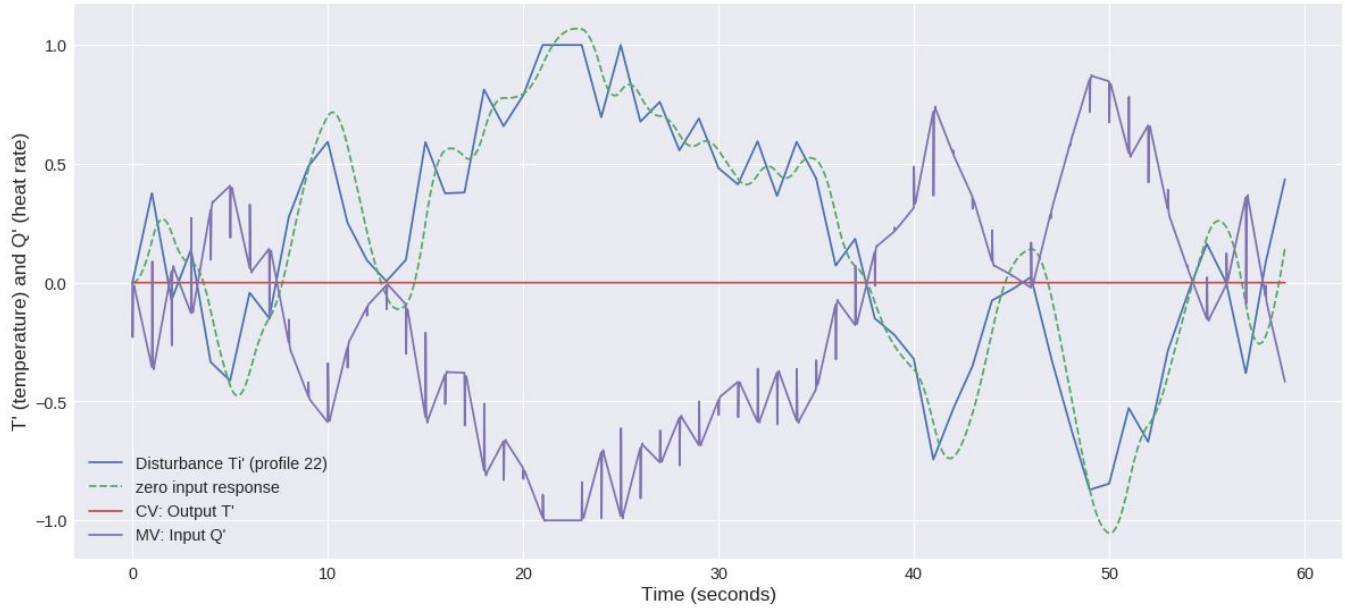

Disturbance Ti' (profile 17)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

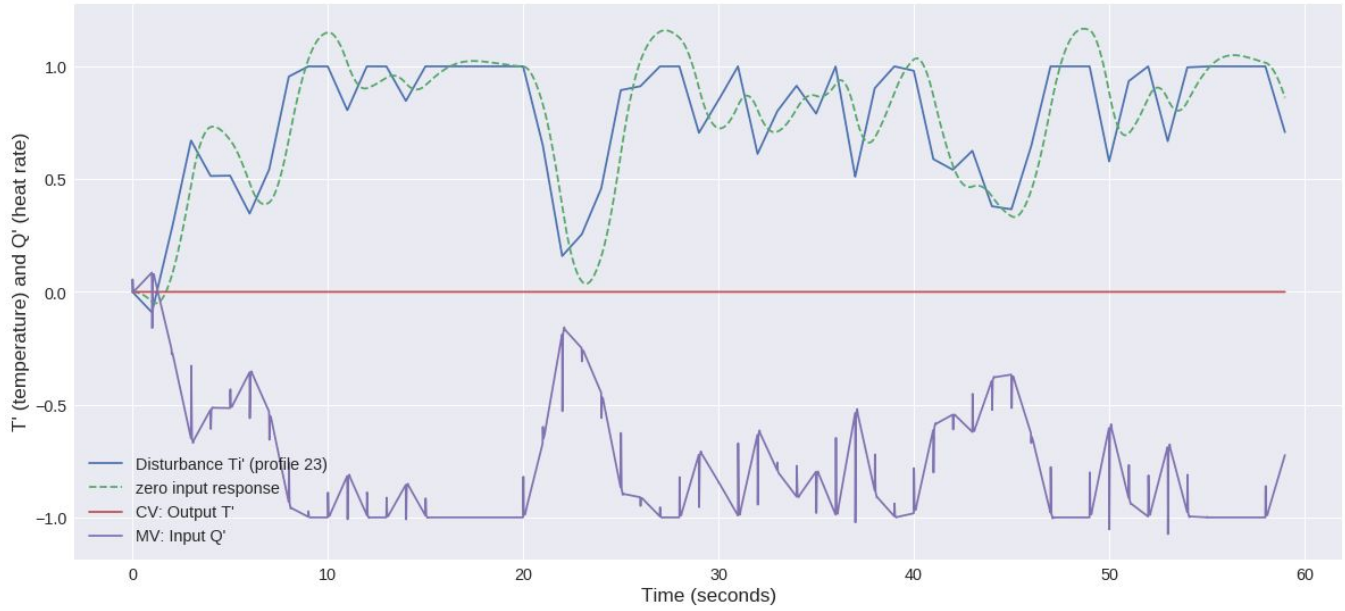

Disturbance Ti' (profile 18)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

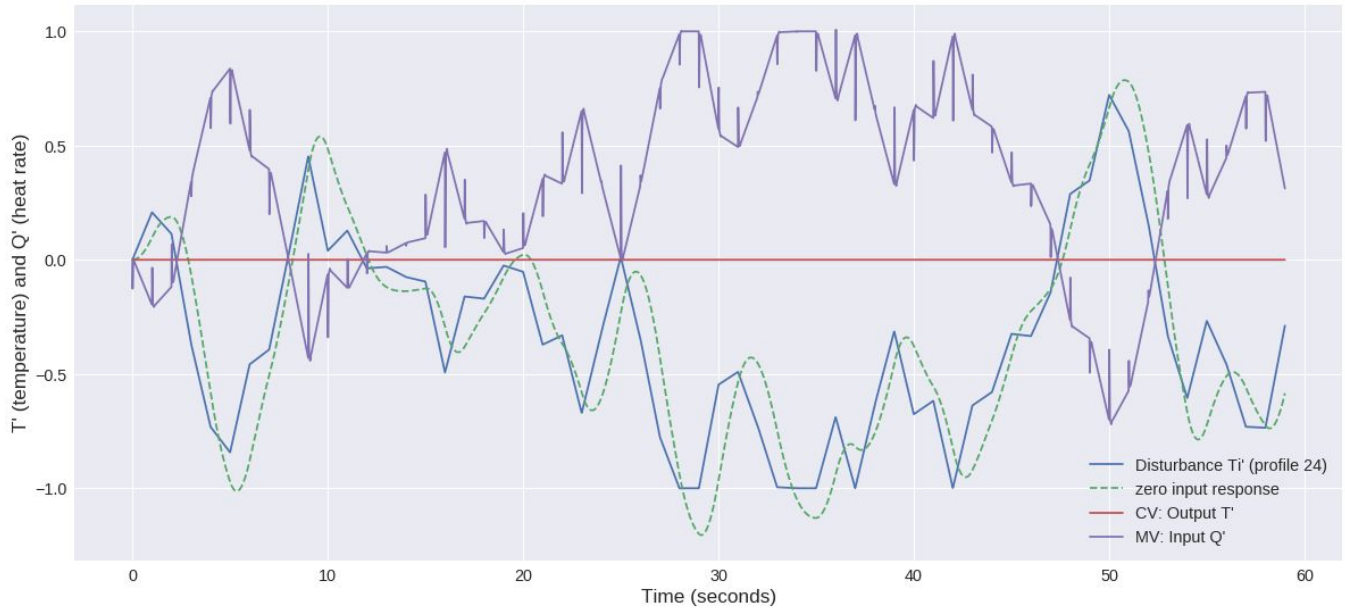

Disturbance Ti' (profile 19)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

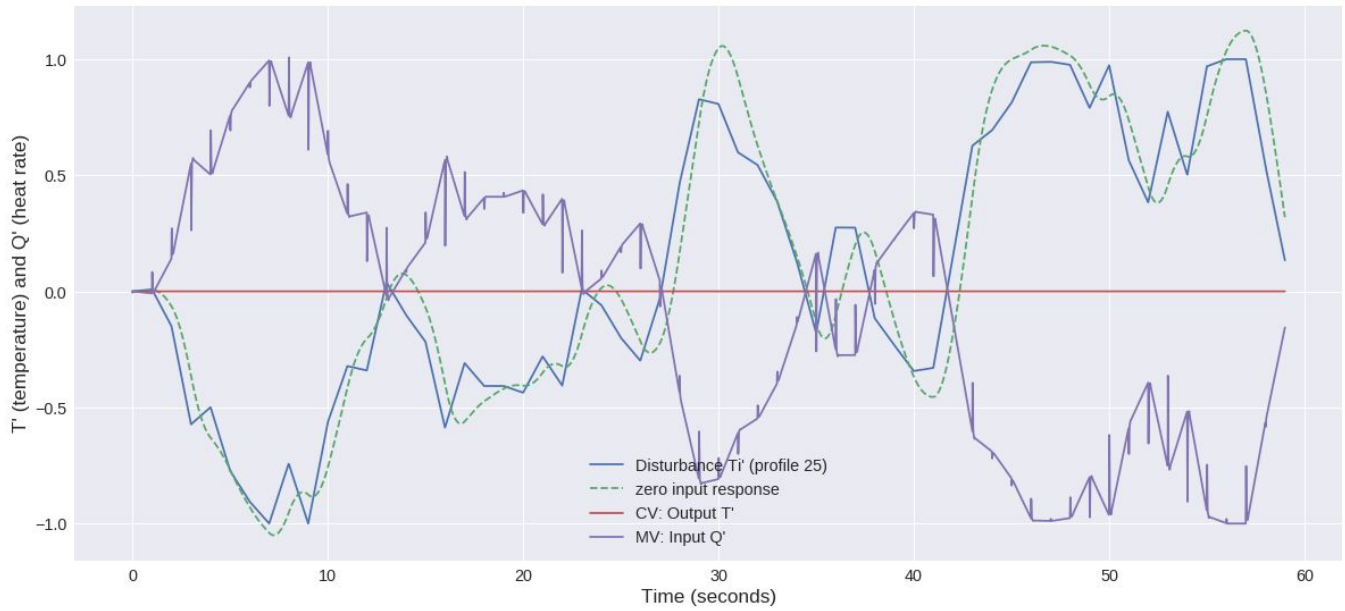

Disturbance Ti' (profile 20)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

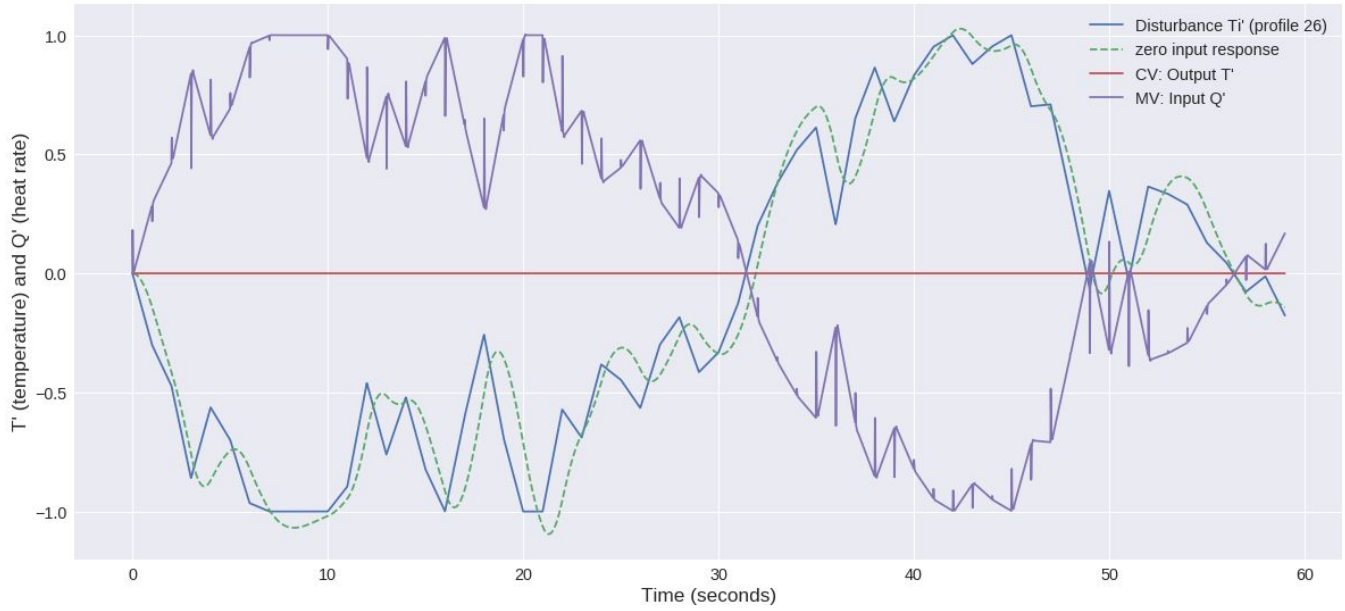

Disturbance Ti' (profile 21)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

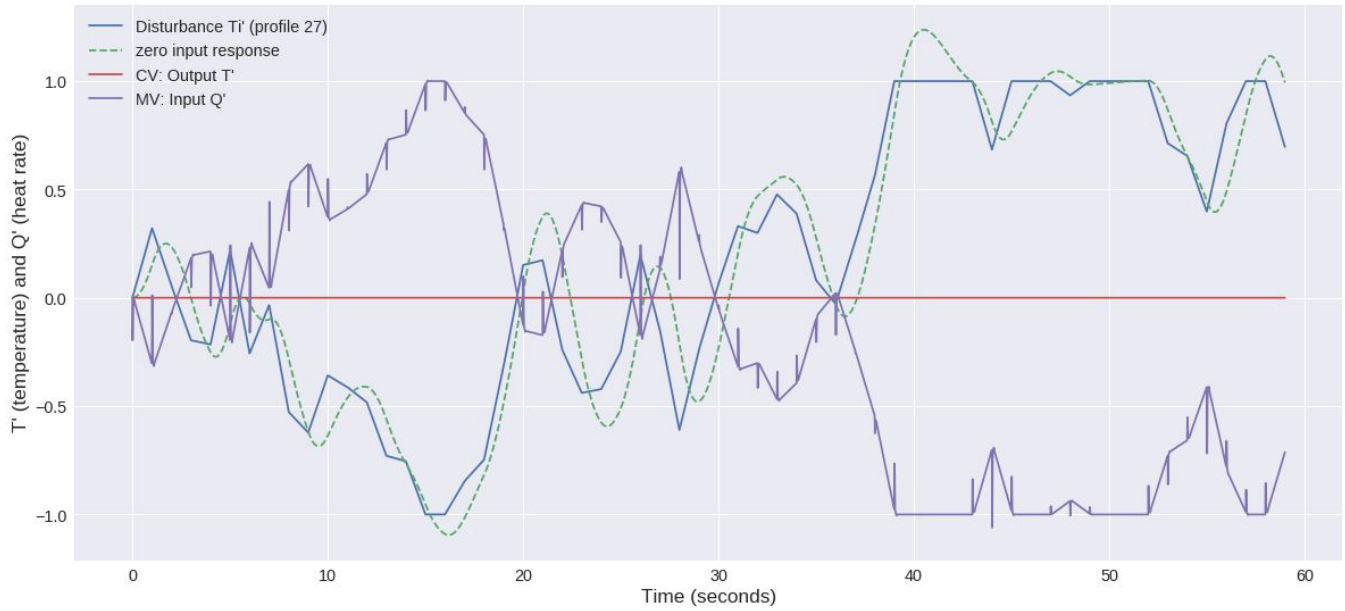

Disturbance Ti' (profile 22)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

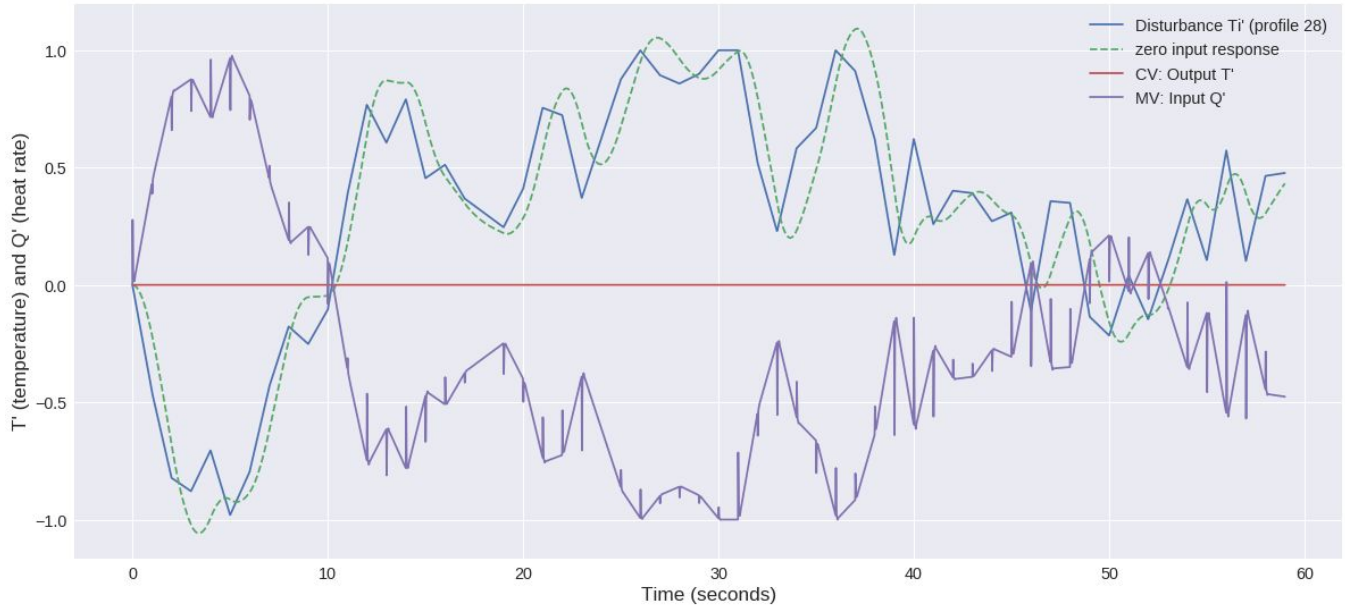

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 23)
zero input response
CV: Output T'
MV: Input Q'

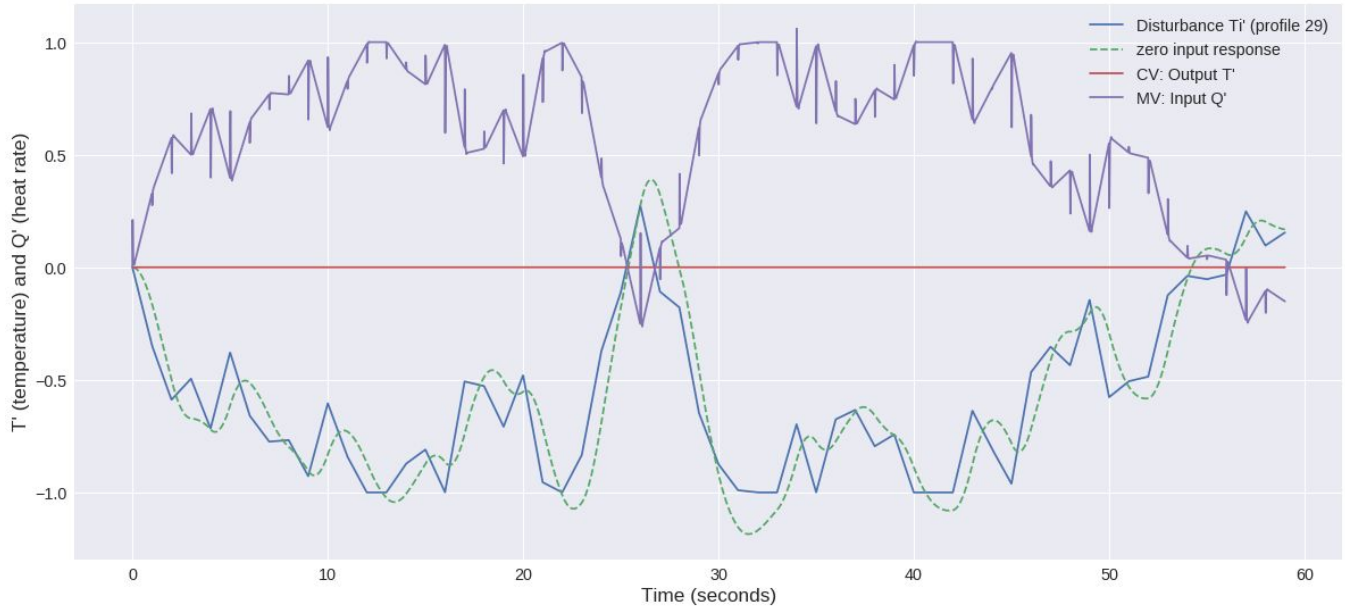

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 24)
zero input response
CV: Output T'
MV: Input Q'

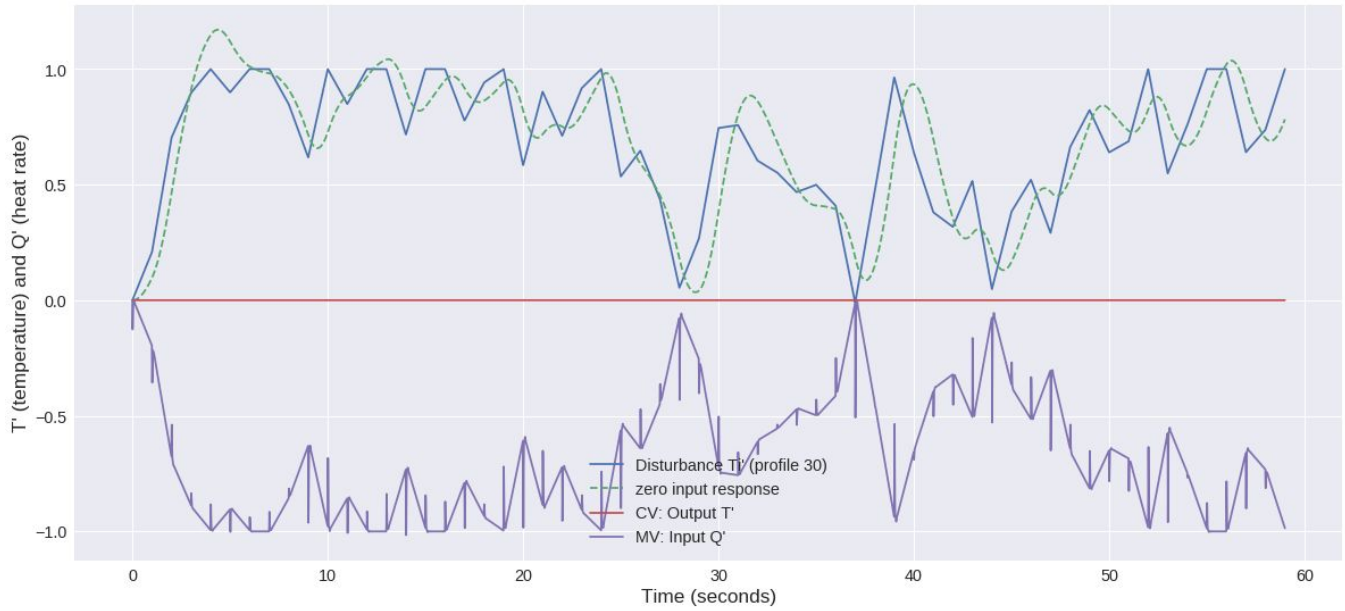

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 25)
zero input response
CV: Output T'
MV: Input Q'

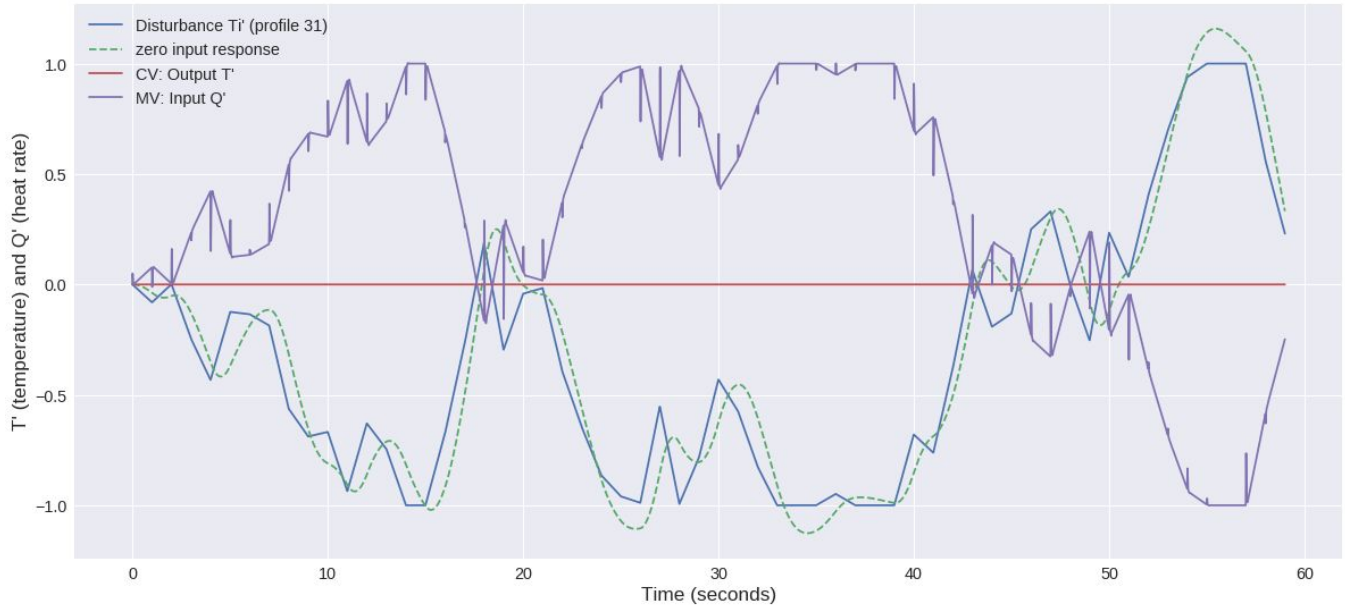

Disturbance Ti' (profile 26)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

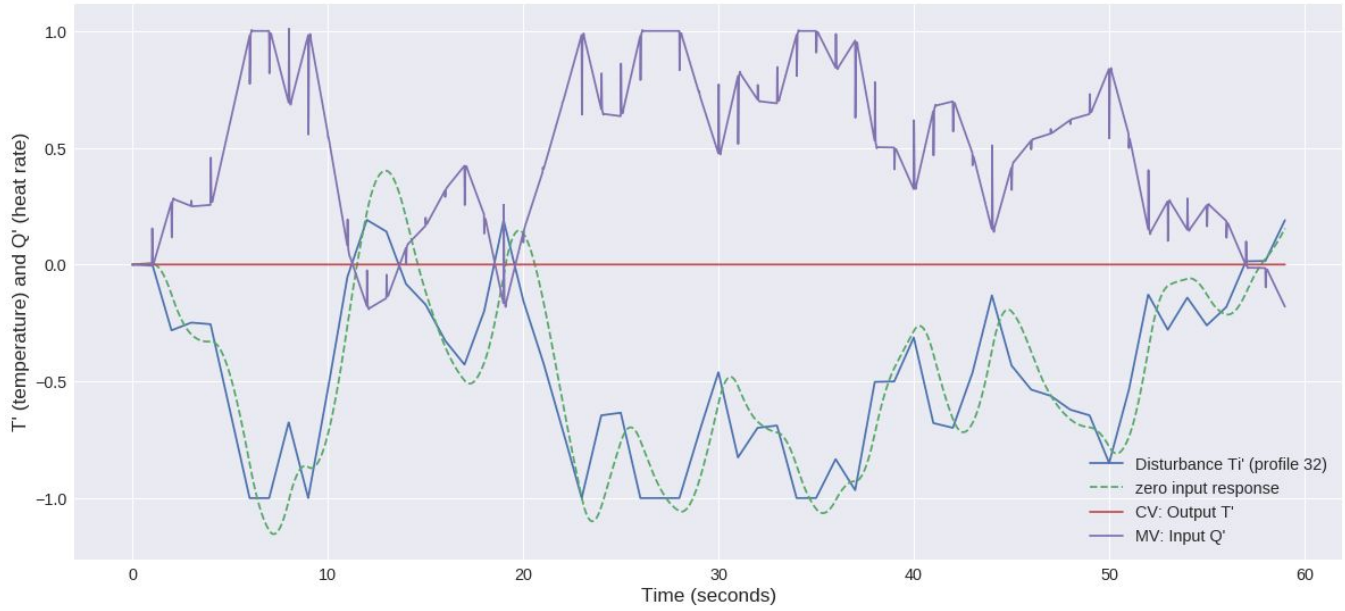

Disturbance Ti' (profile 27)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

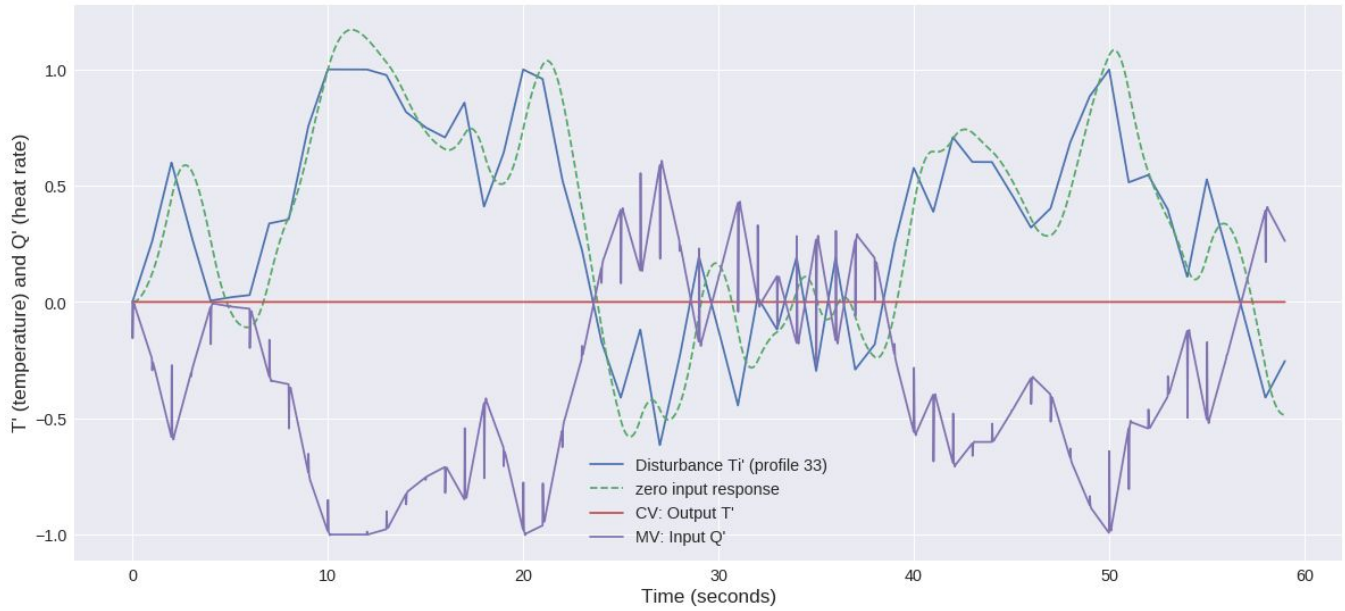

Disturbance Ti' (profile 28)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

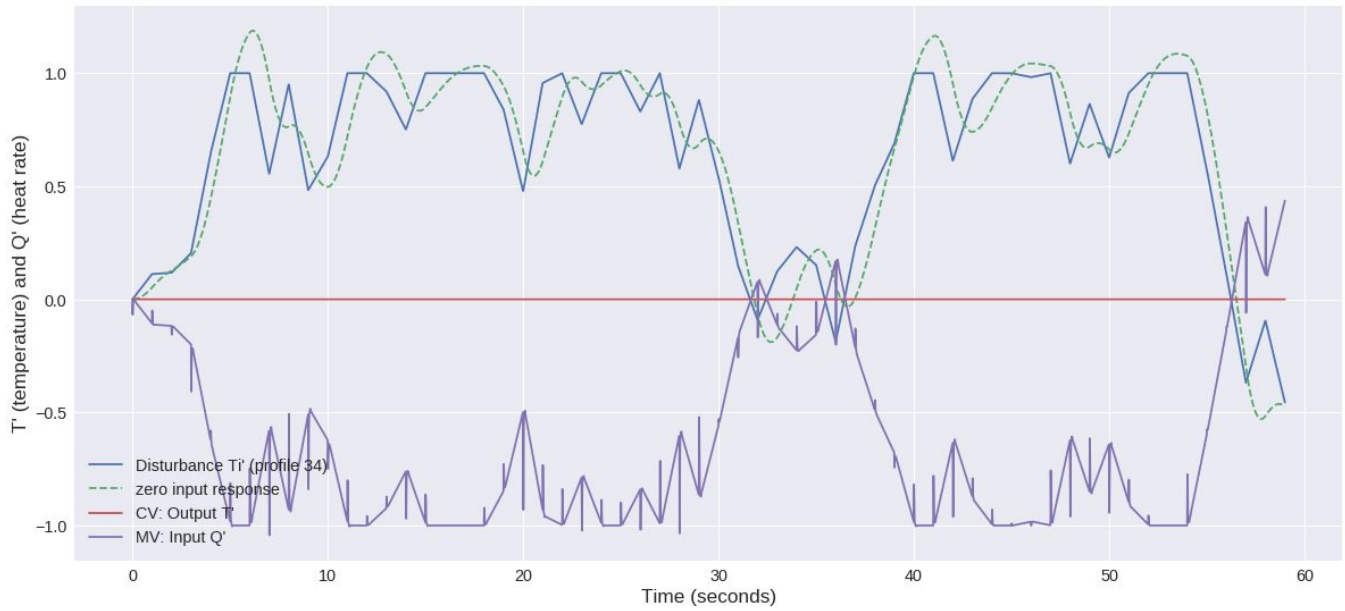

Disturbance Ti' (profile 29)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

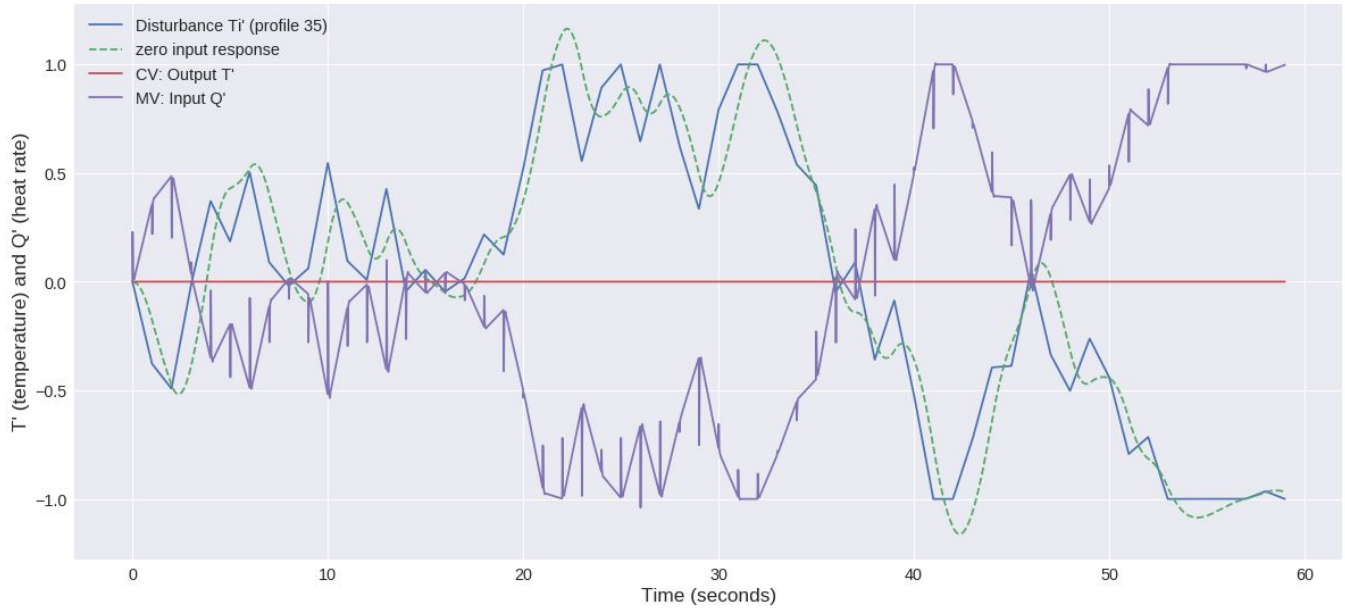

Disturbance Ti' (profile 30)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

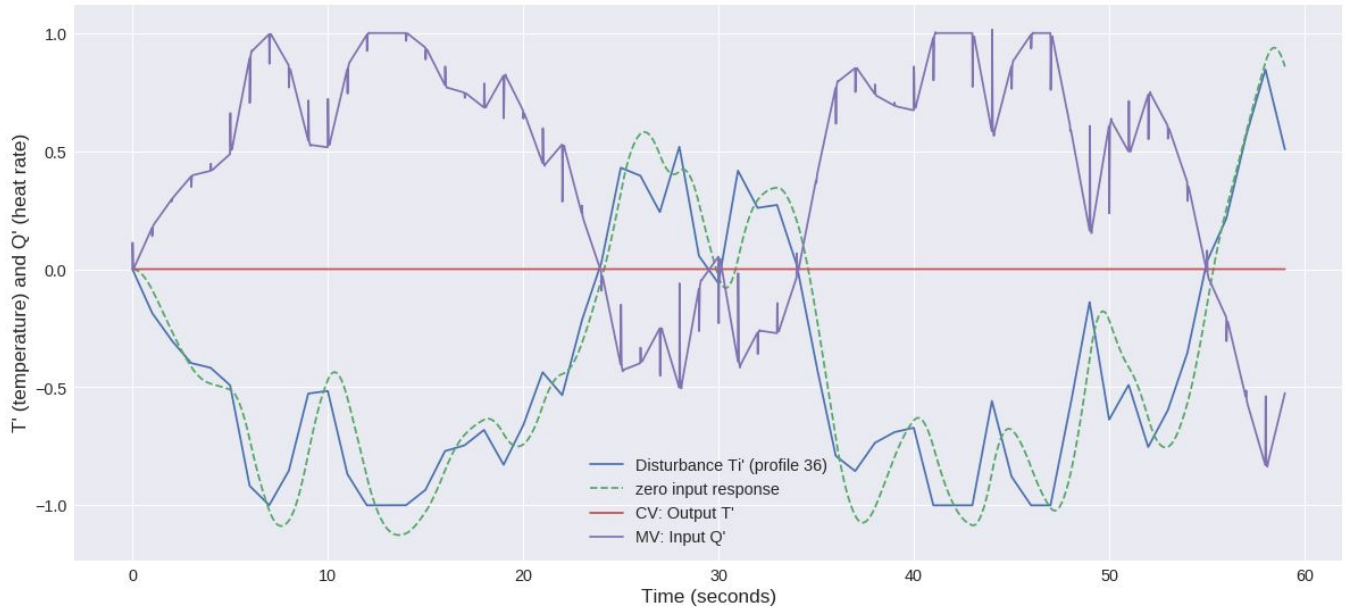

Disturbance Ti' (profile 31)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

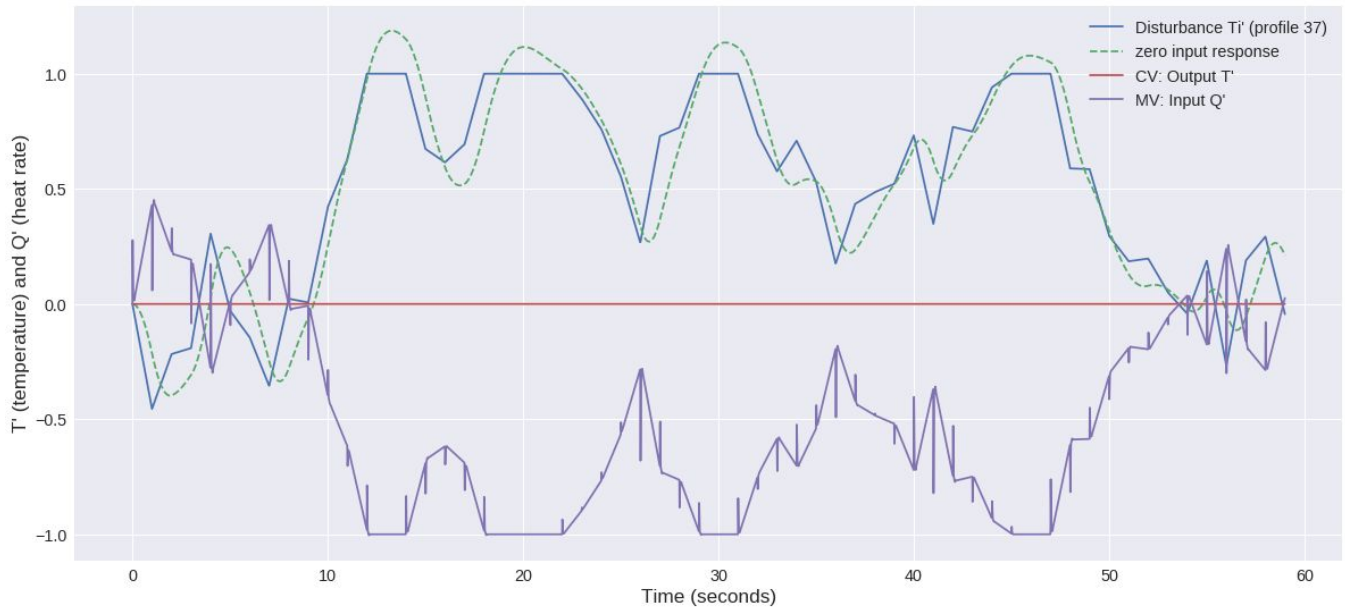

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 32)
zero input response
CV: Output T'
MV: Input Q'

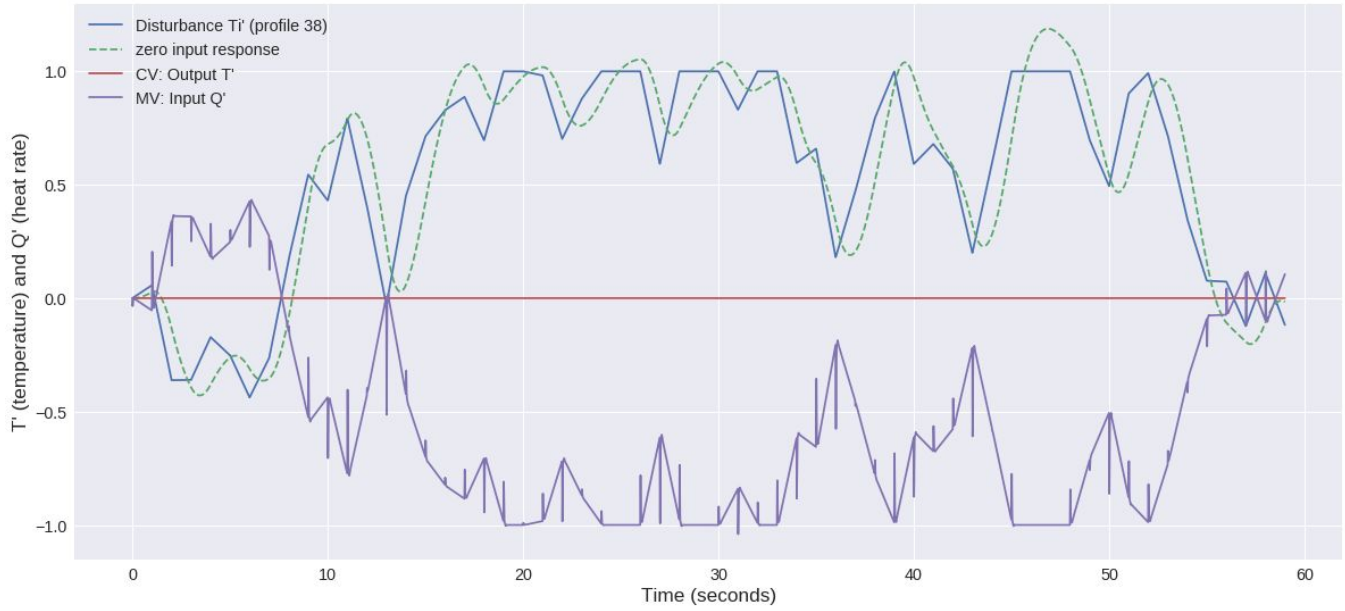

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 33)
zero input response
CV: Output T'
MV: Input Q'

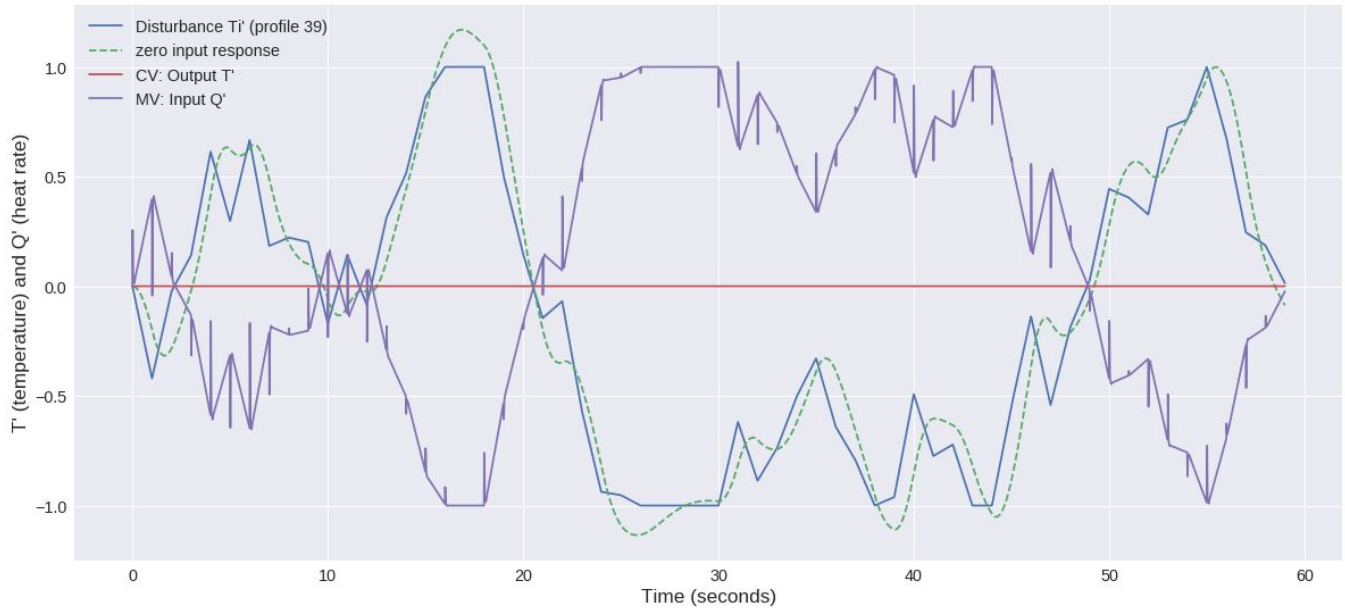

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 34)
zero input response
CV: Output T'
MV: Input Q'

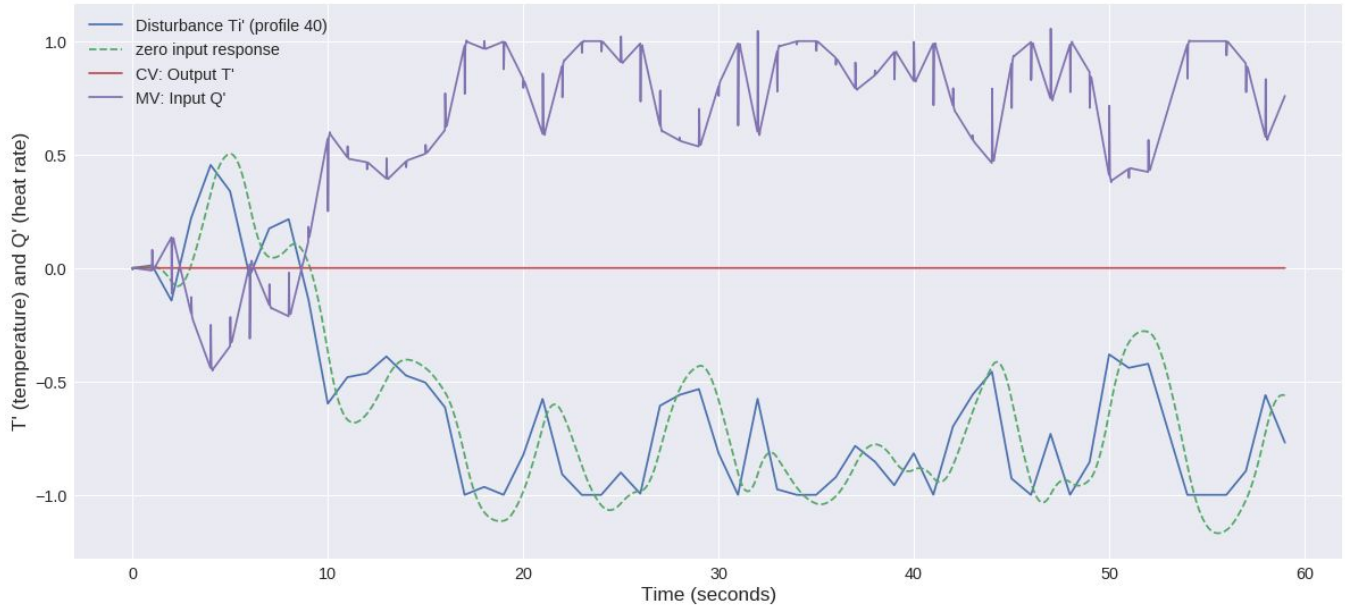

Disturbance Ti' (profile 35)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

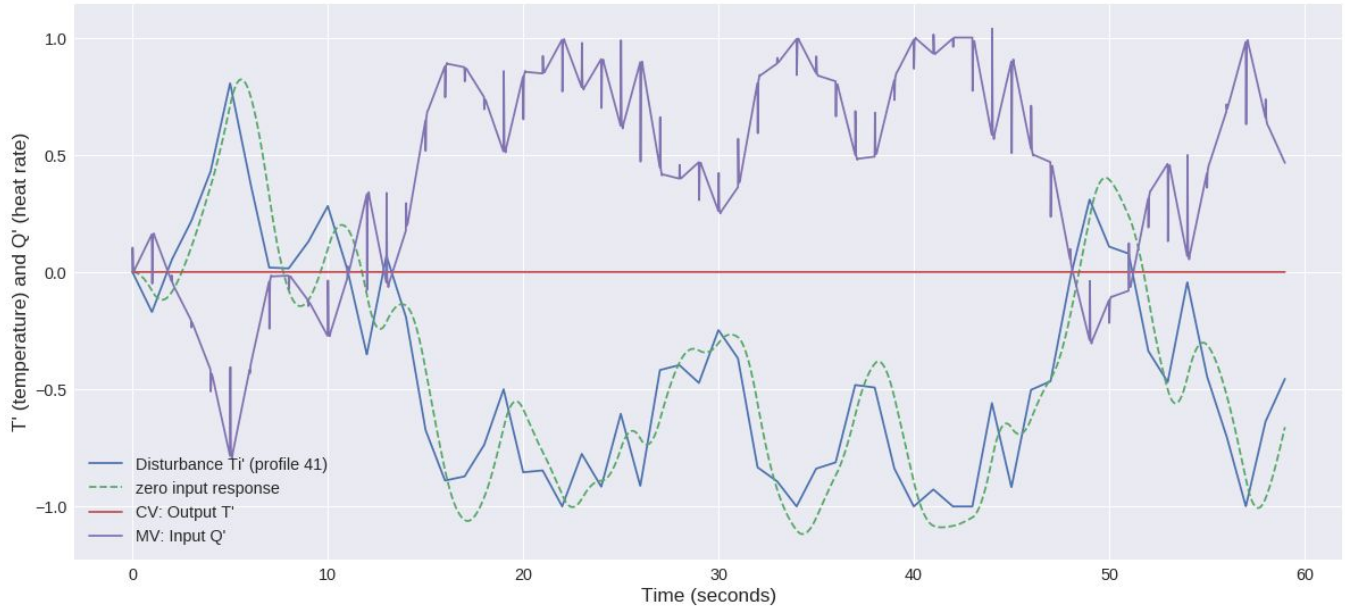

Disturbance Ti' (profile 36)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

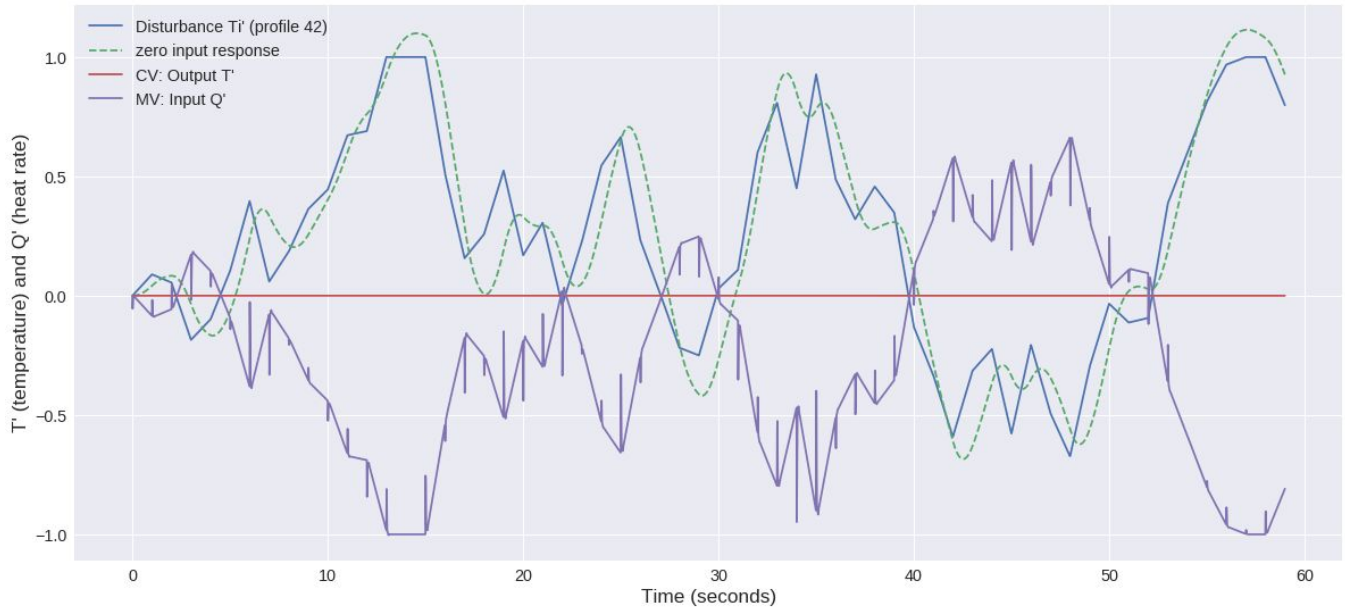

Disturbance Ti' (profile 37)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

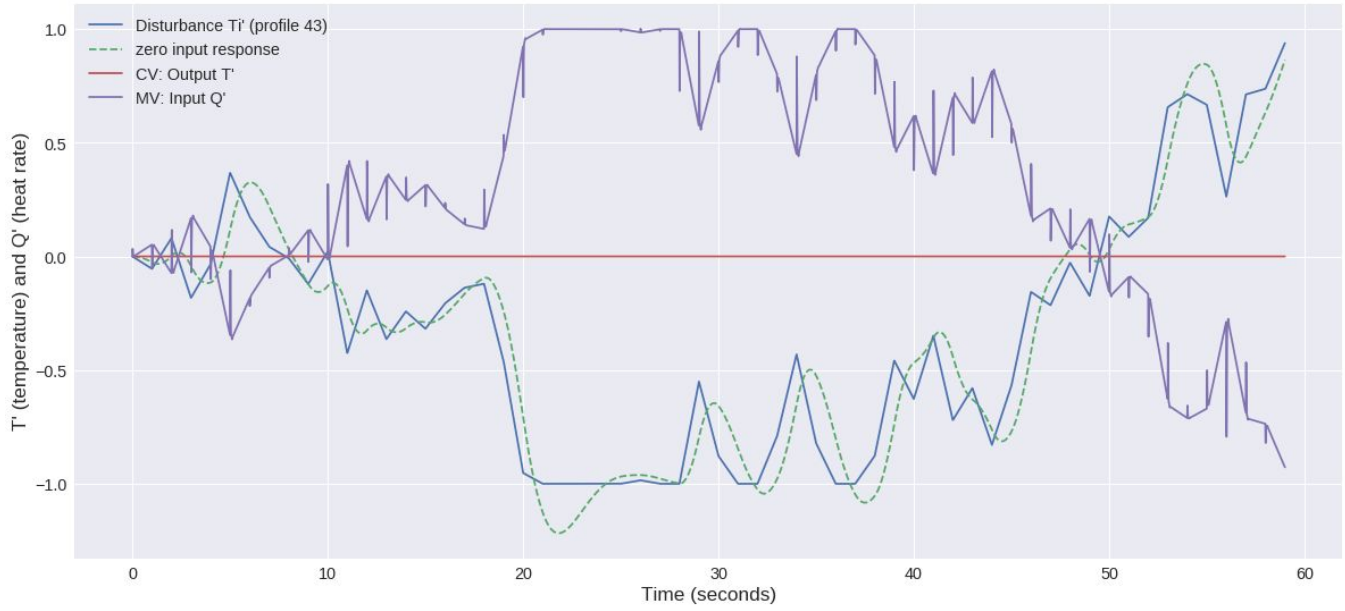

Disturbance Ti' (profile 38)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

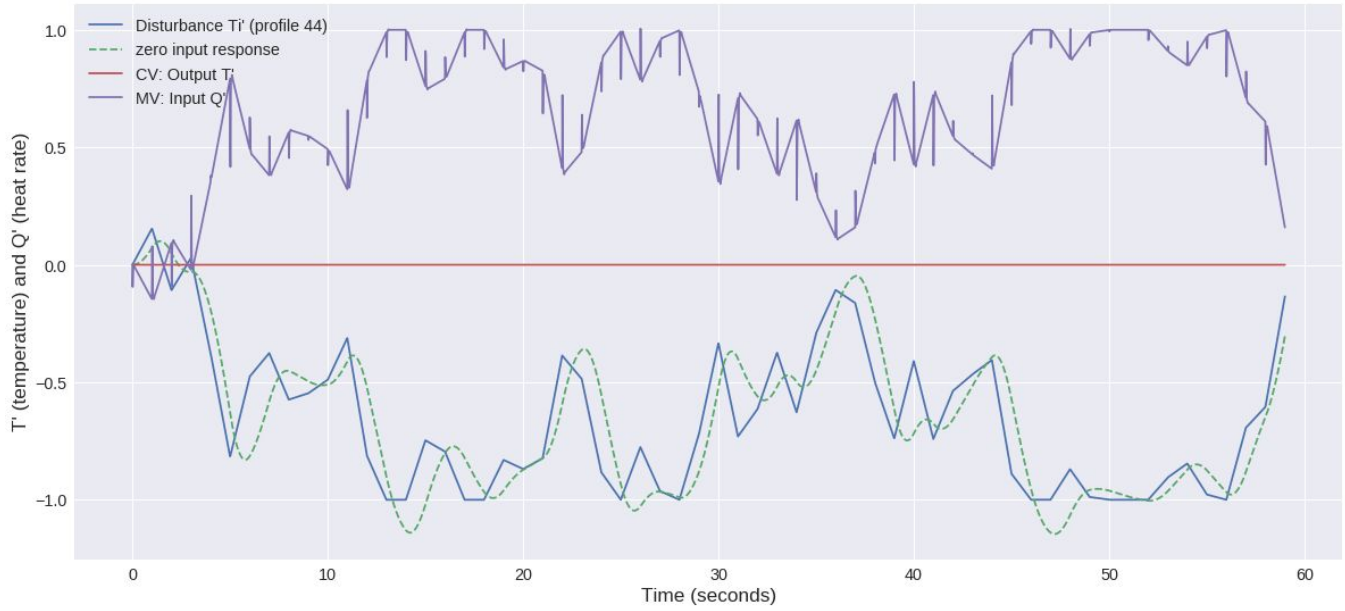

Disturbance Ti' (profile 39)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

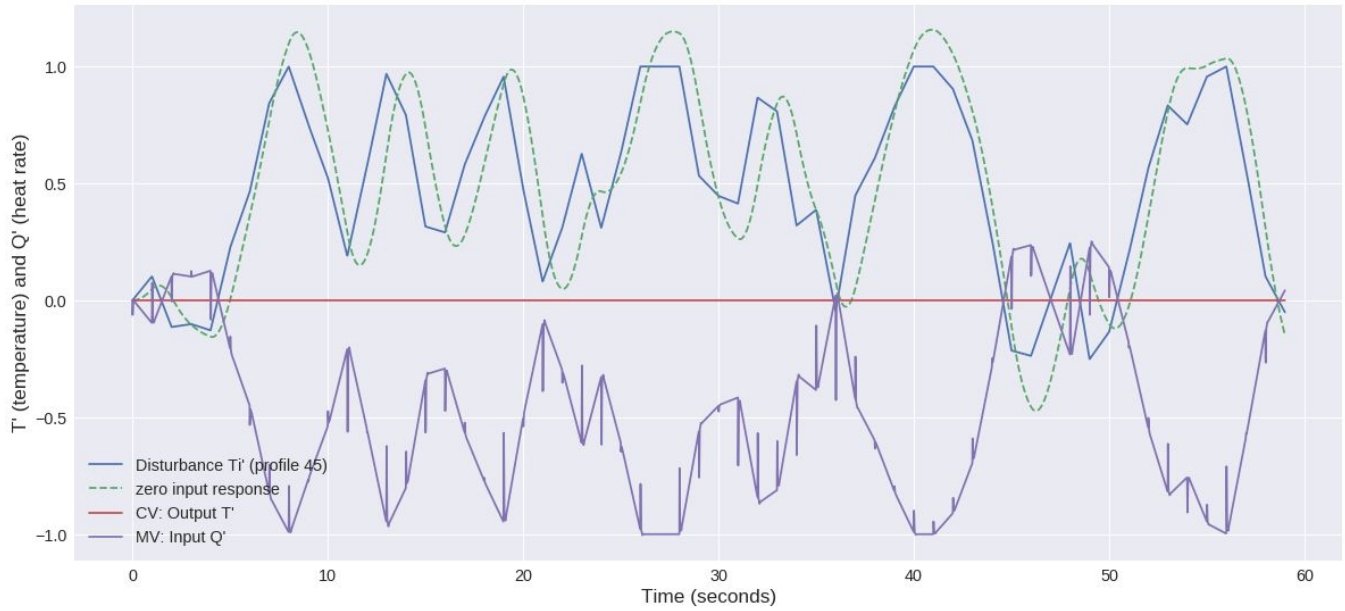

Disturbance Ti' (profile 40)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

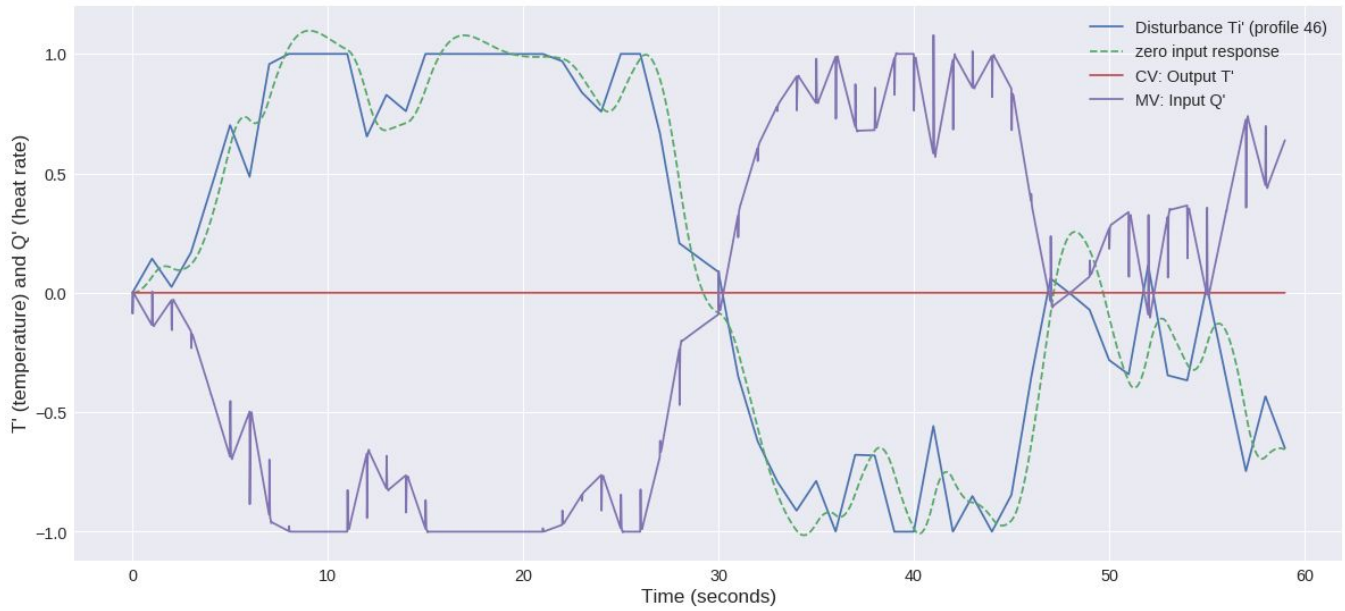

Disturbance Ti' (profile 41)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

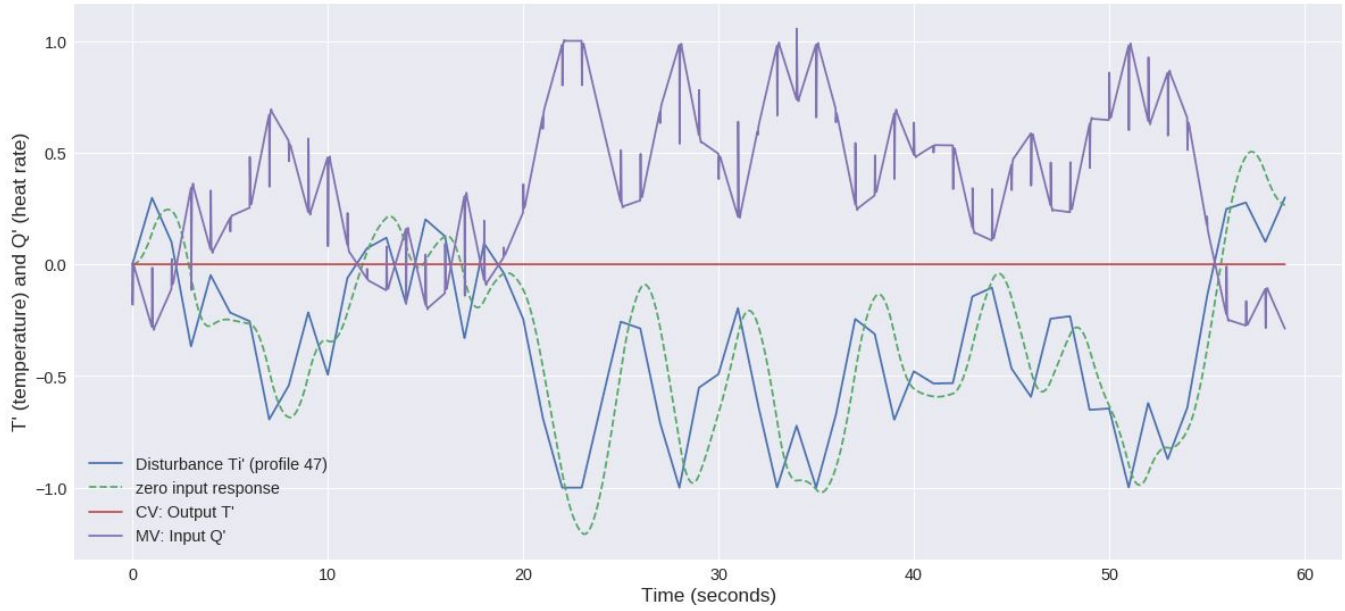

Disturbance Ti' (profile 42)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

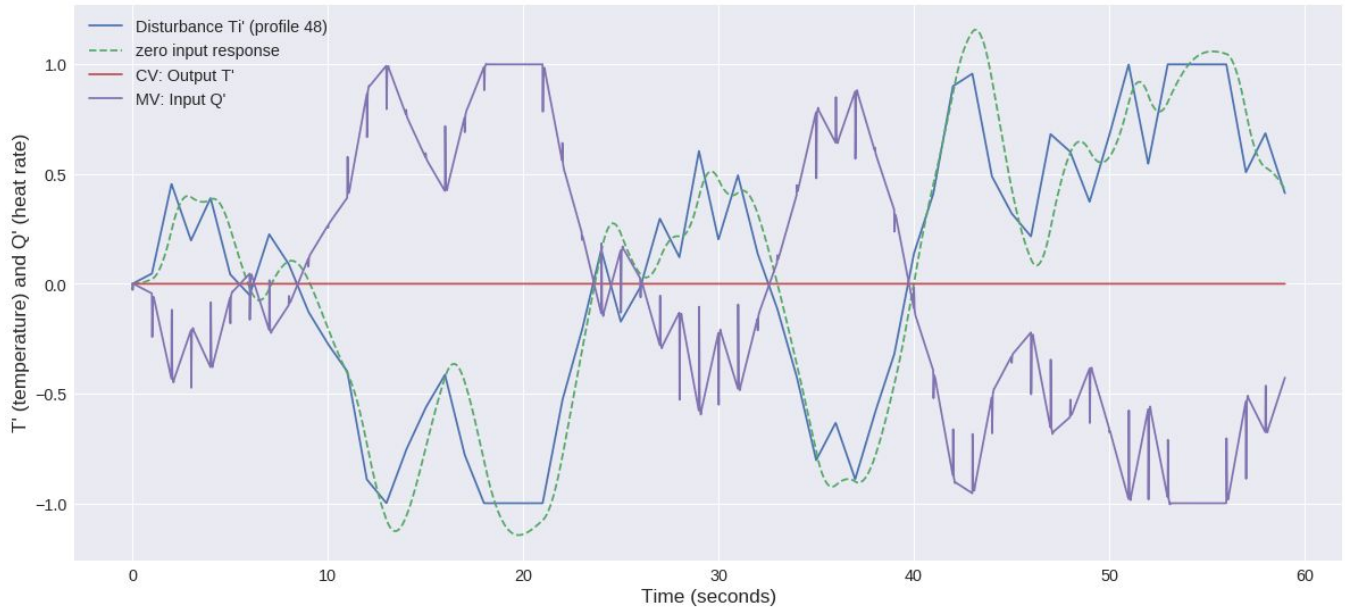

Disturbance Ti' (profile 43)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

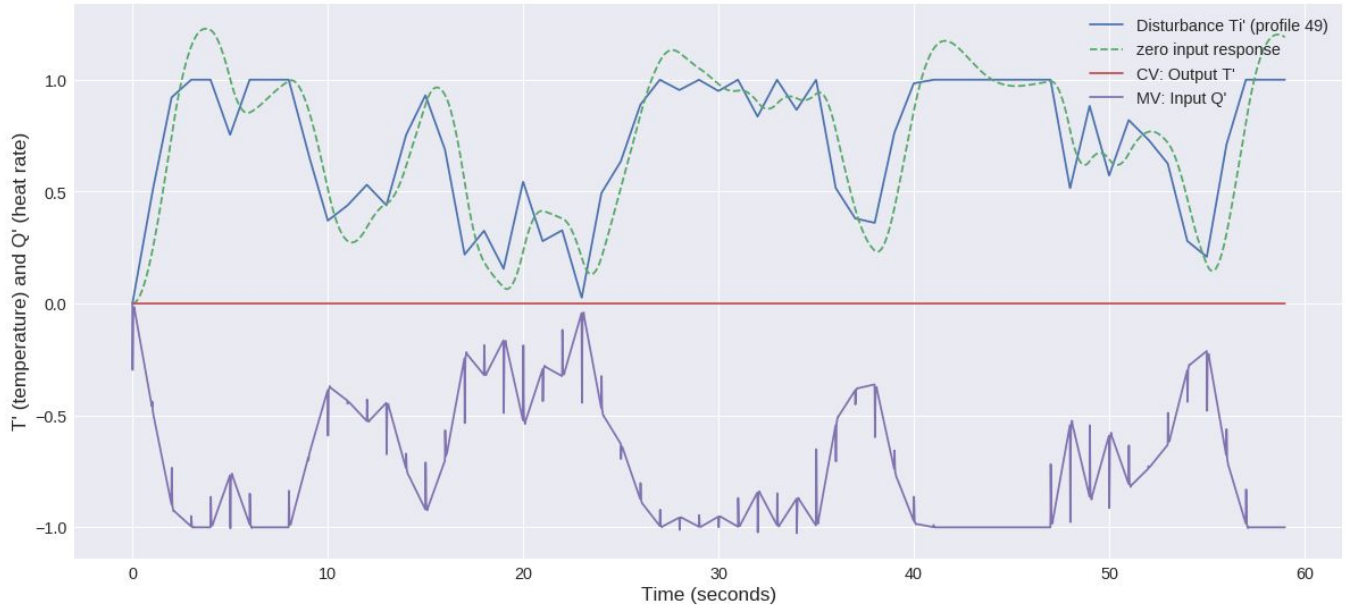

Disturbance Ti' (profile 44)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

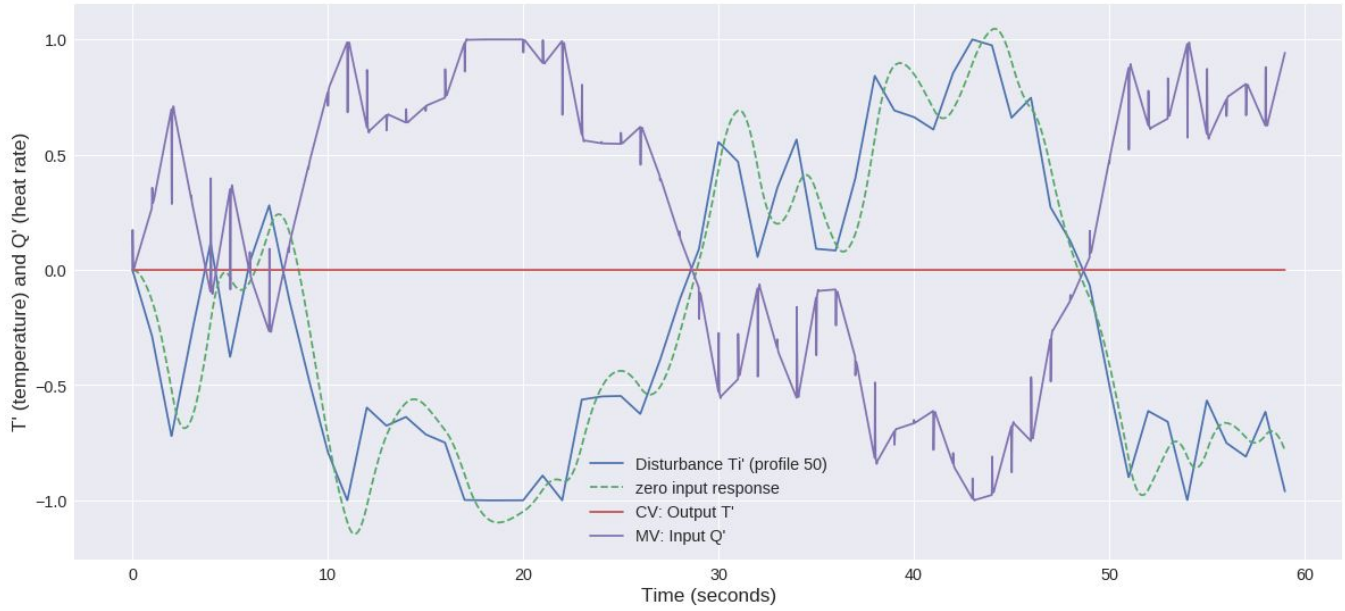

Disturbance Ti' (profile 45)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

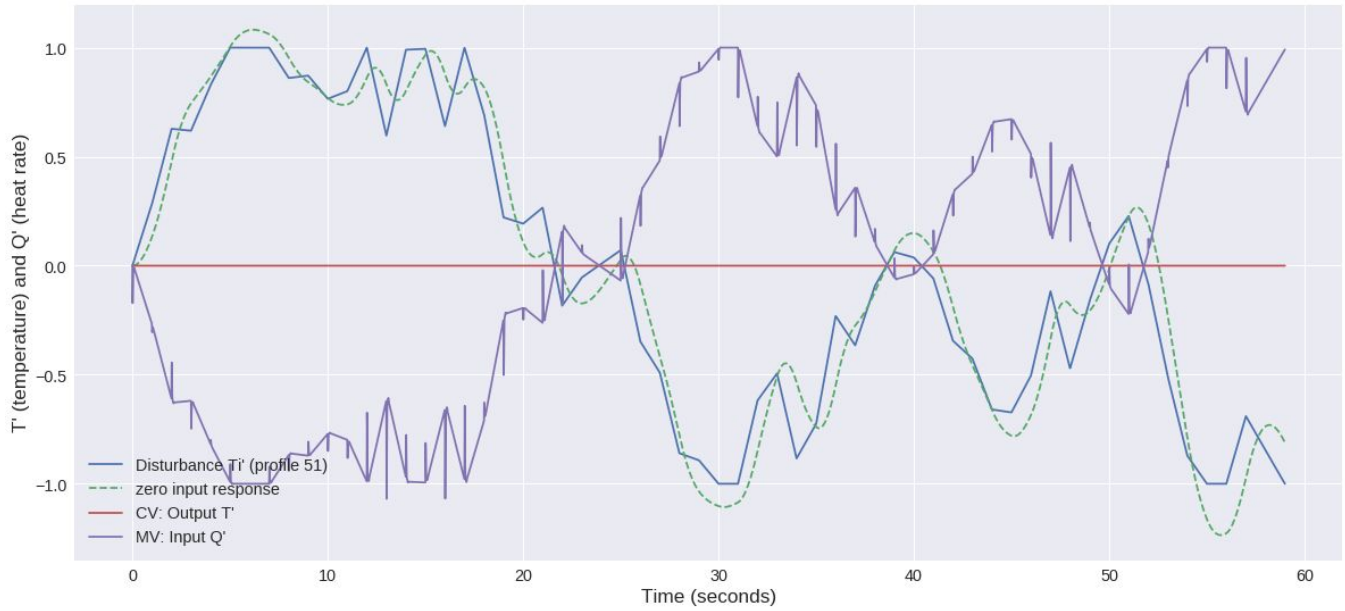

Disturbance Ti' (profile 46)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

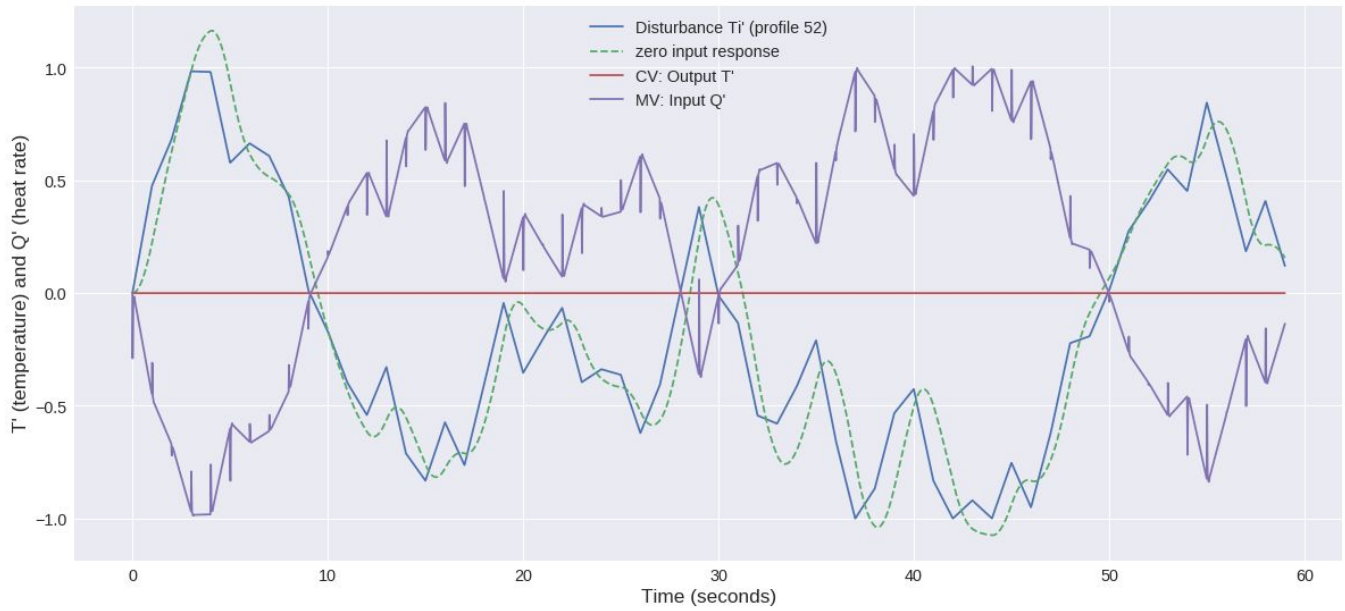

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 47)
zero input response
CV: Output T'
MV: Input Q'

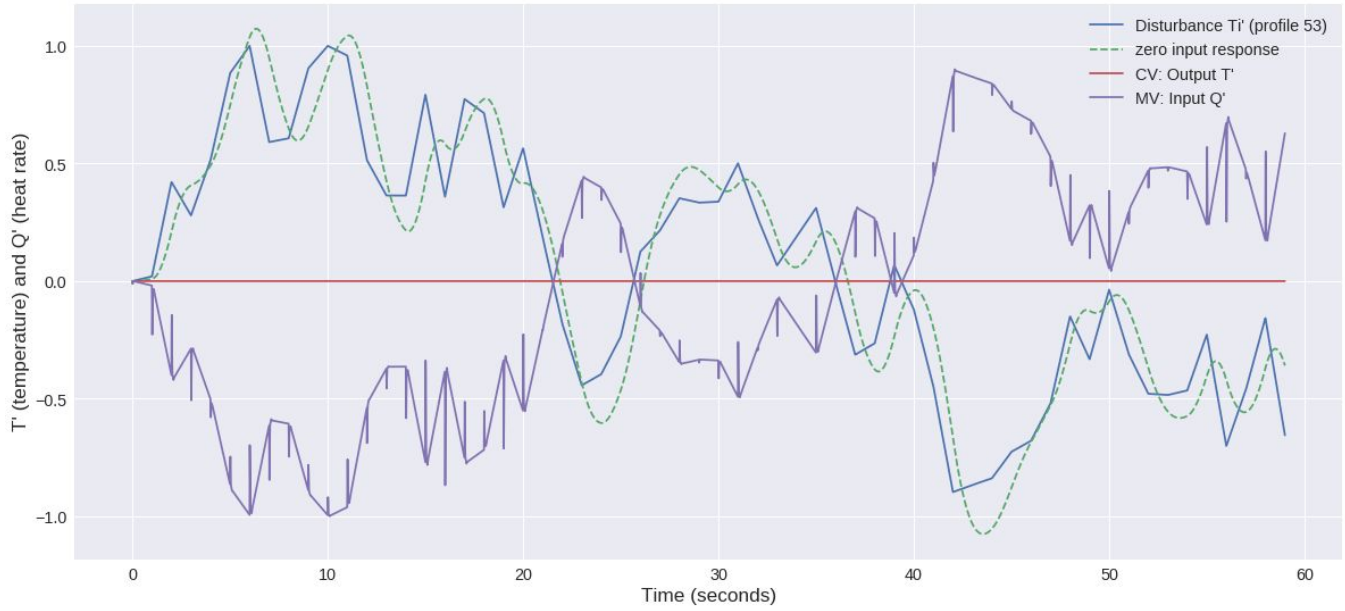

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 48)
zero input response
CV: Output T'
MV: Input Q'

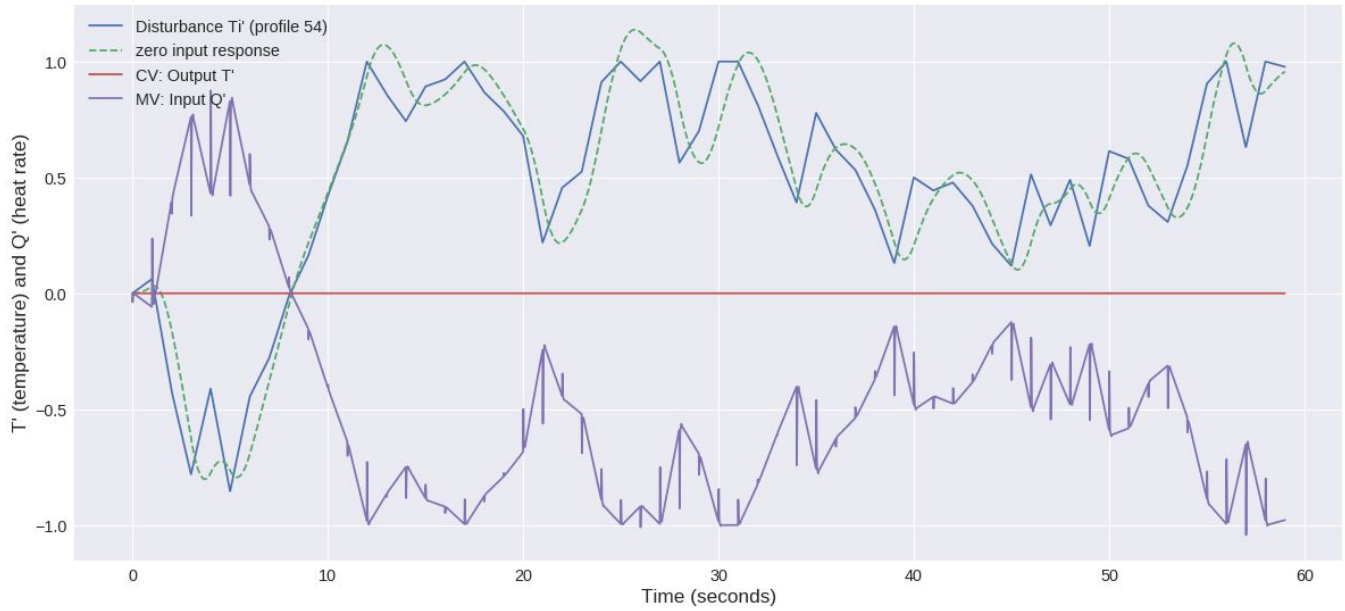

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 49)
zero input response
CV: Output T'
MV: Input Q'

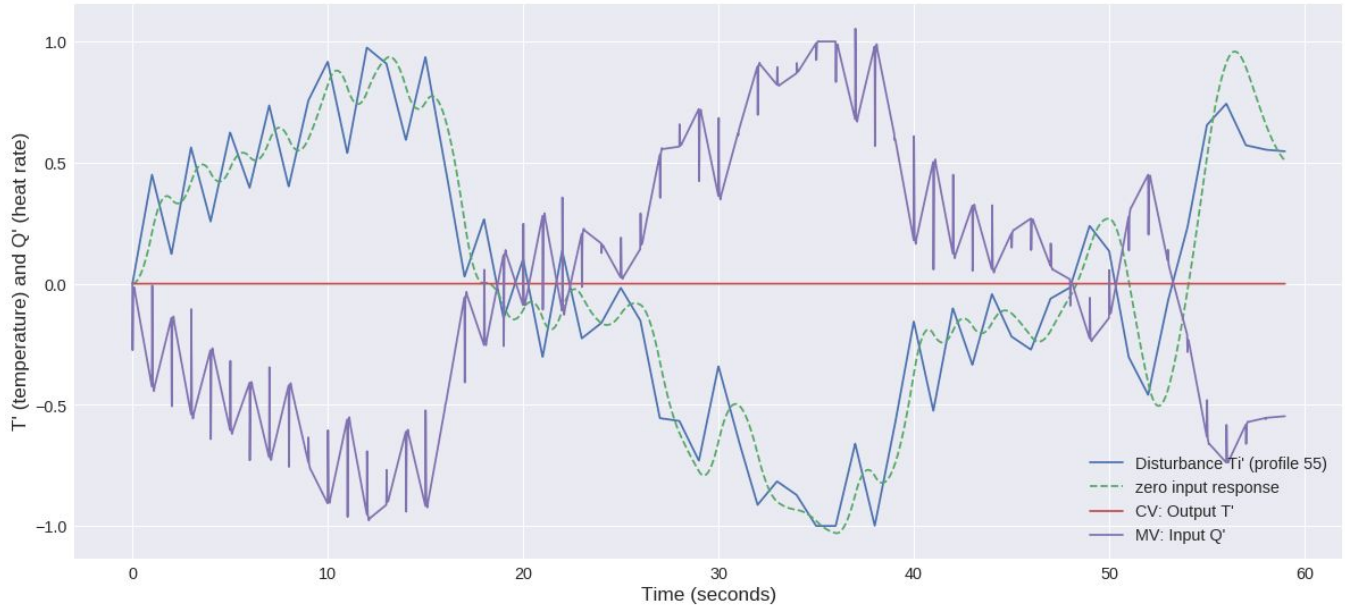

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 50)
zero input response
CV: Output T'
MV: Input Q'

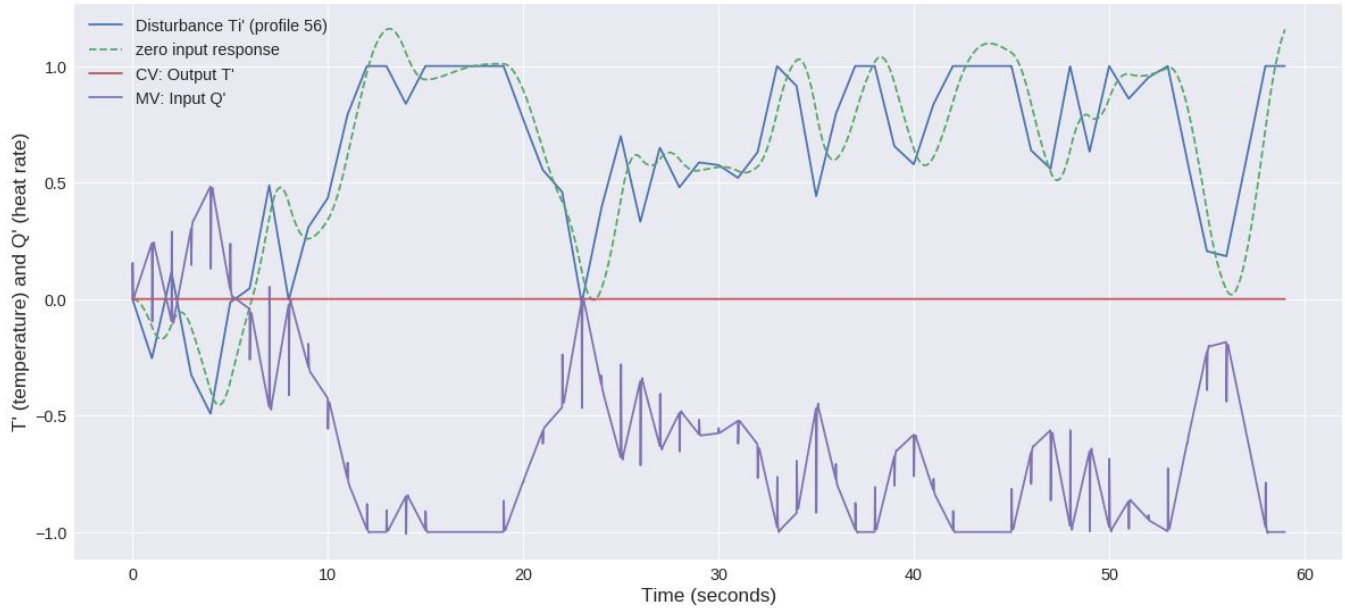

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 51)
zero input response
CV: Output T'
MV: Input Q'

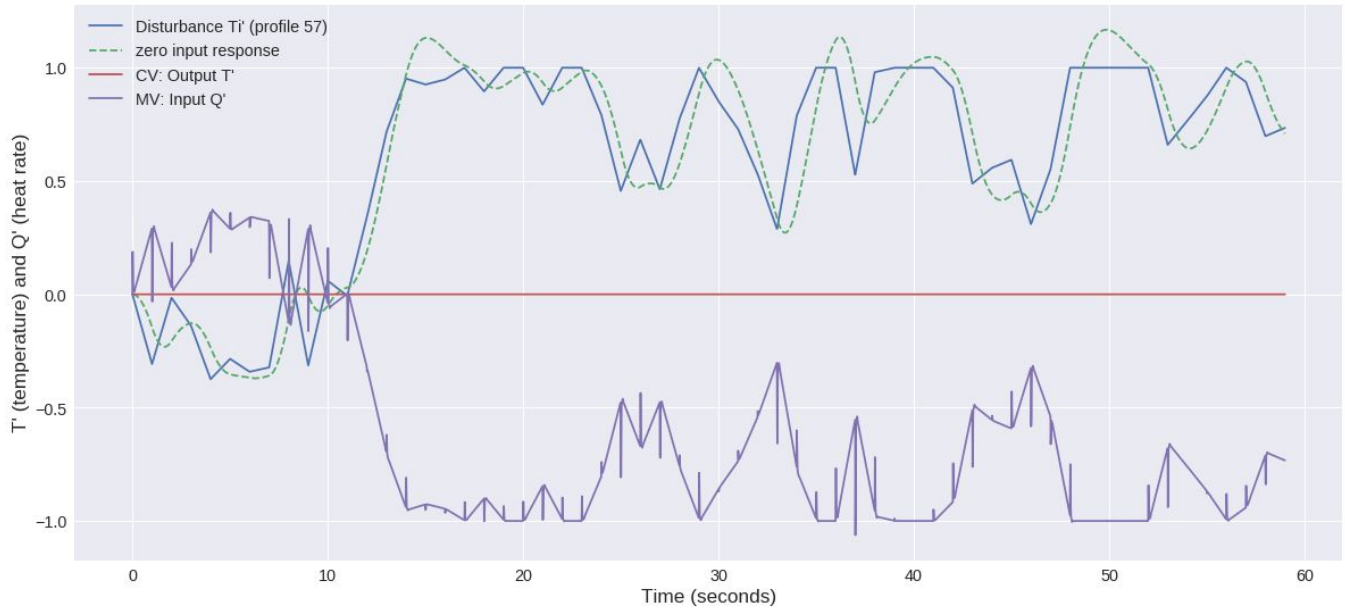

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 52)
zero input response
CV: Output T'
MV: Input Q'

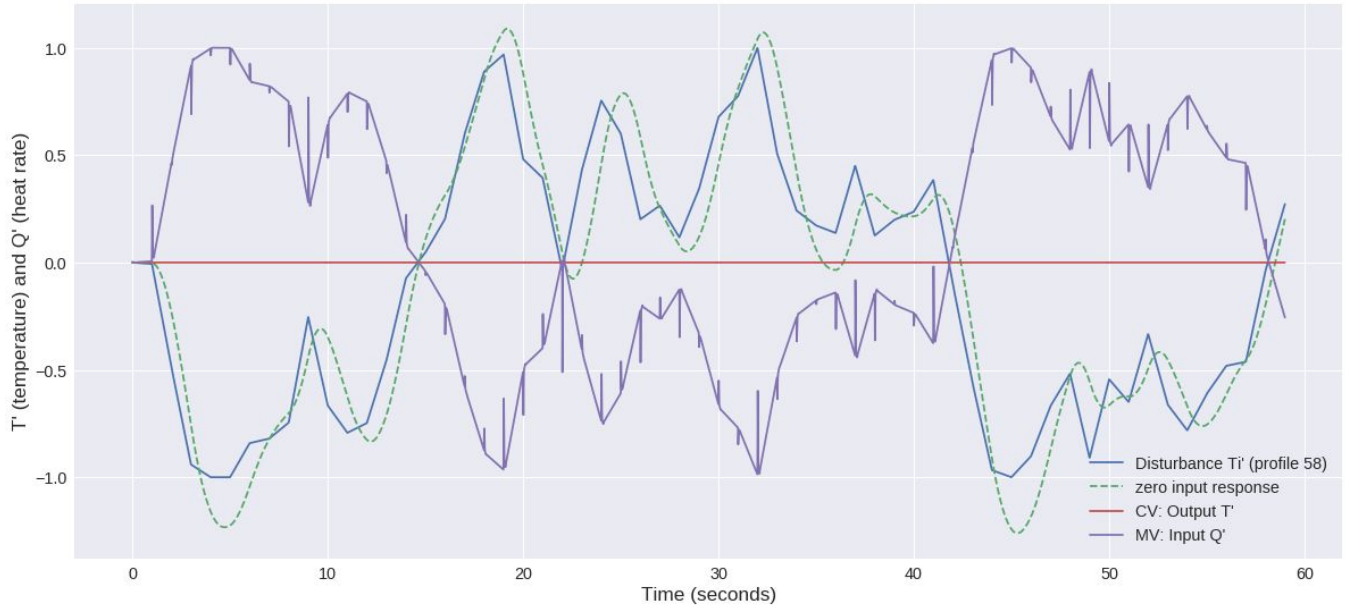

Disturbance Ti' (profile 53)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

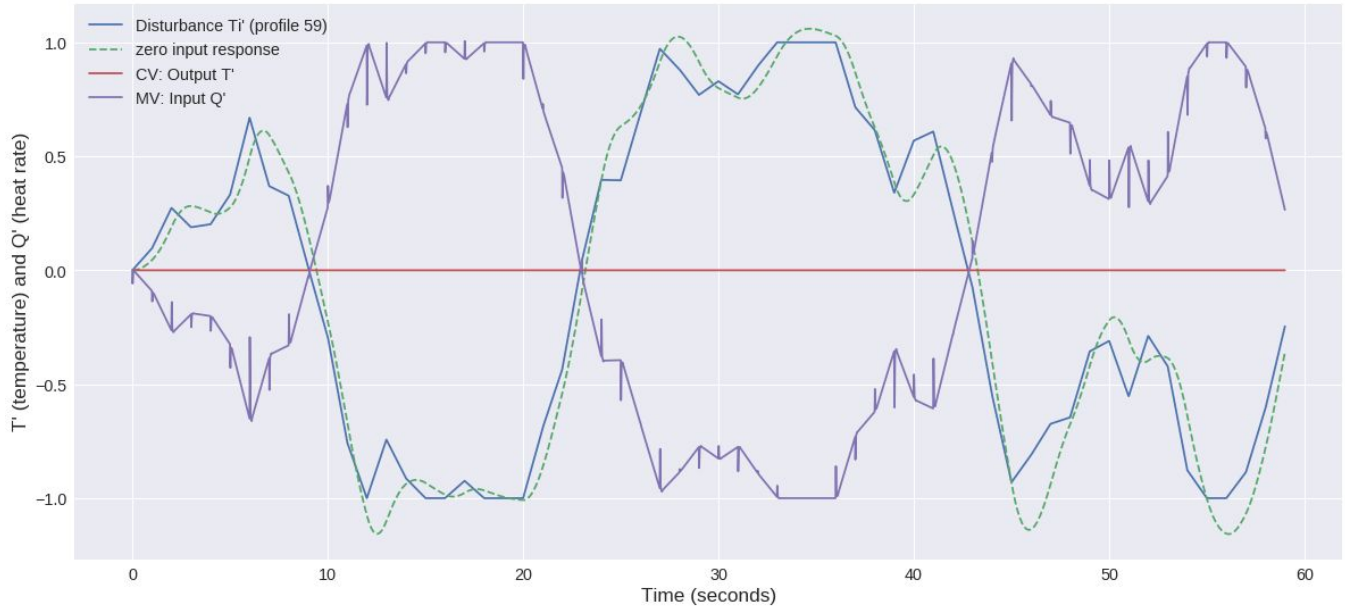

Disturbance Ti' (profile 54)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

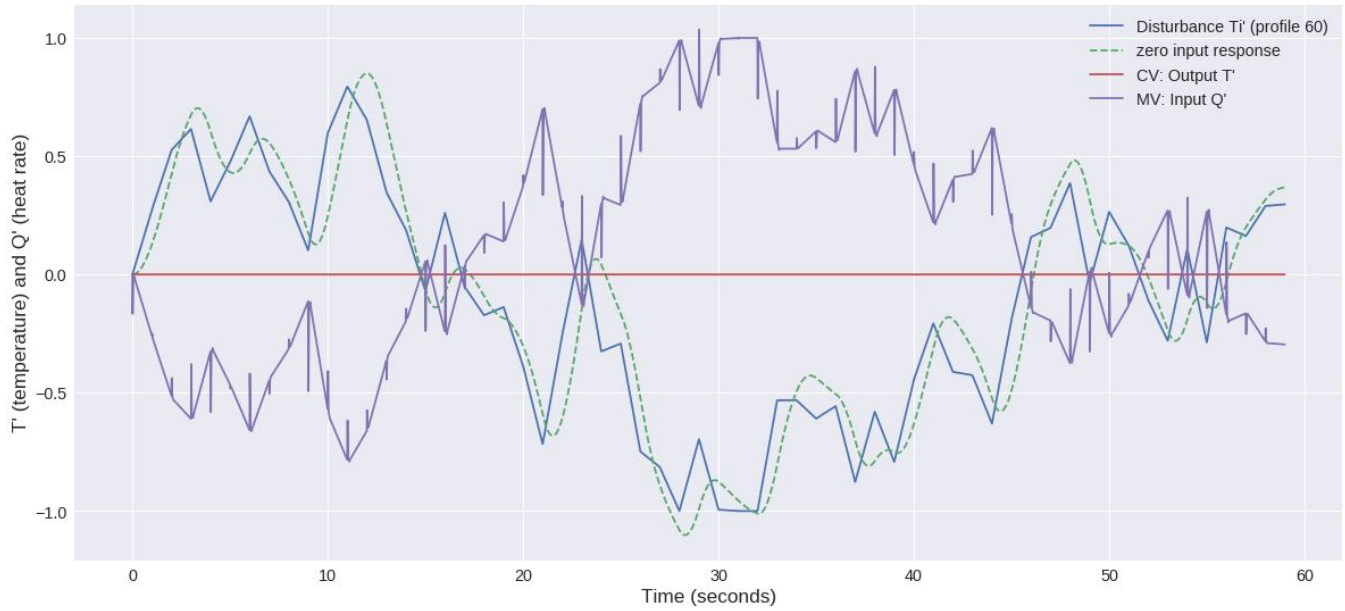

Disturbance Ti' (profile 55)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

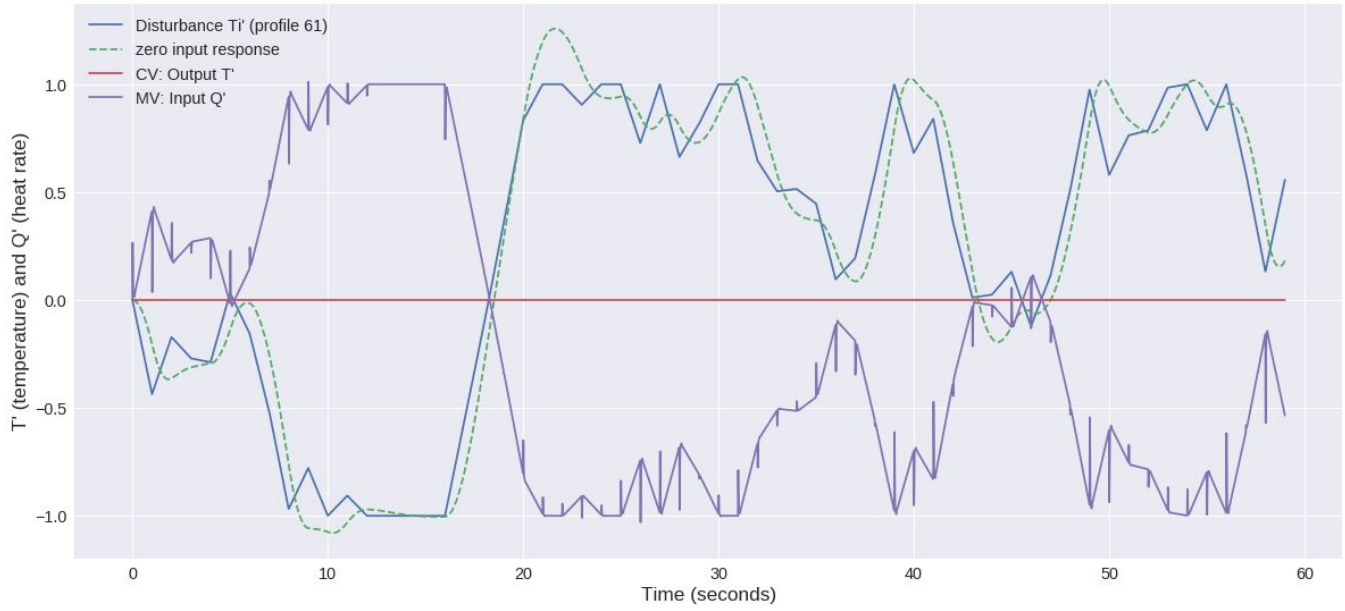

Disturbance Ti' (profile 56)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

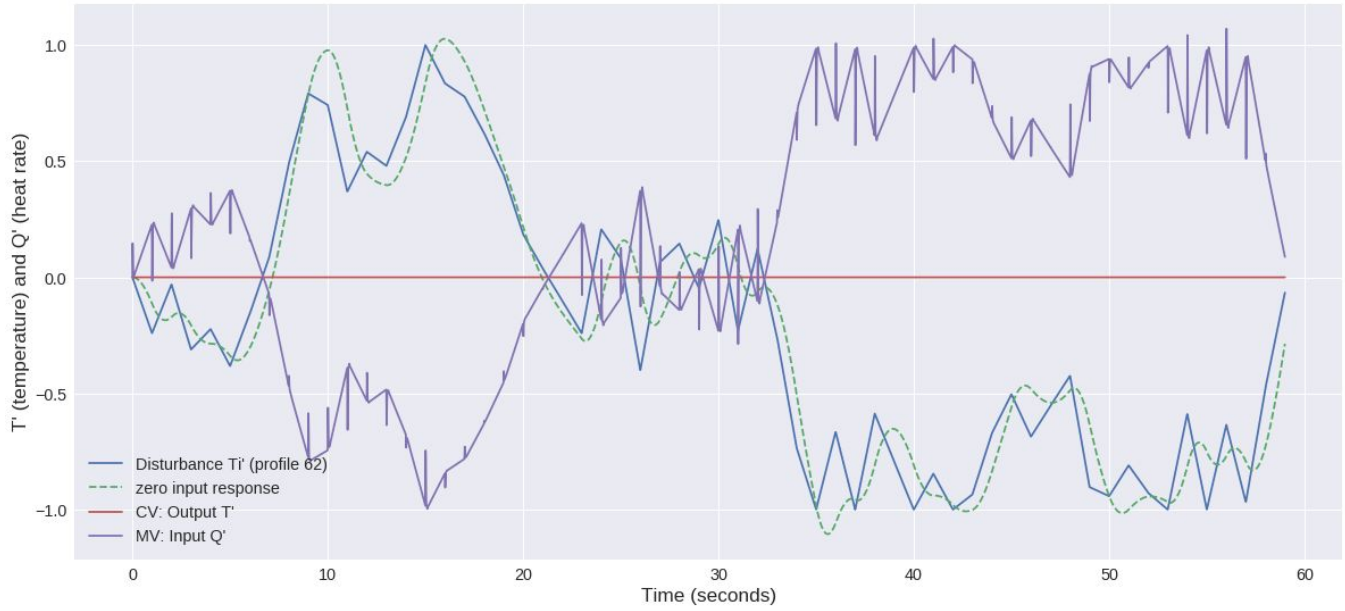

Disturbance Ti' (profile 57)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

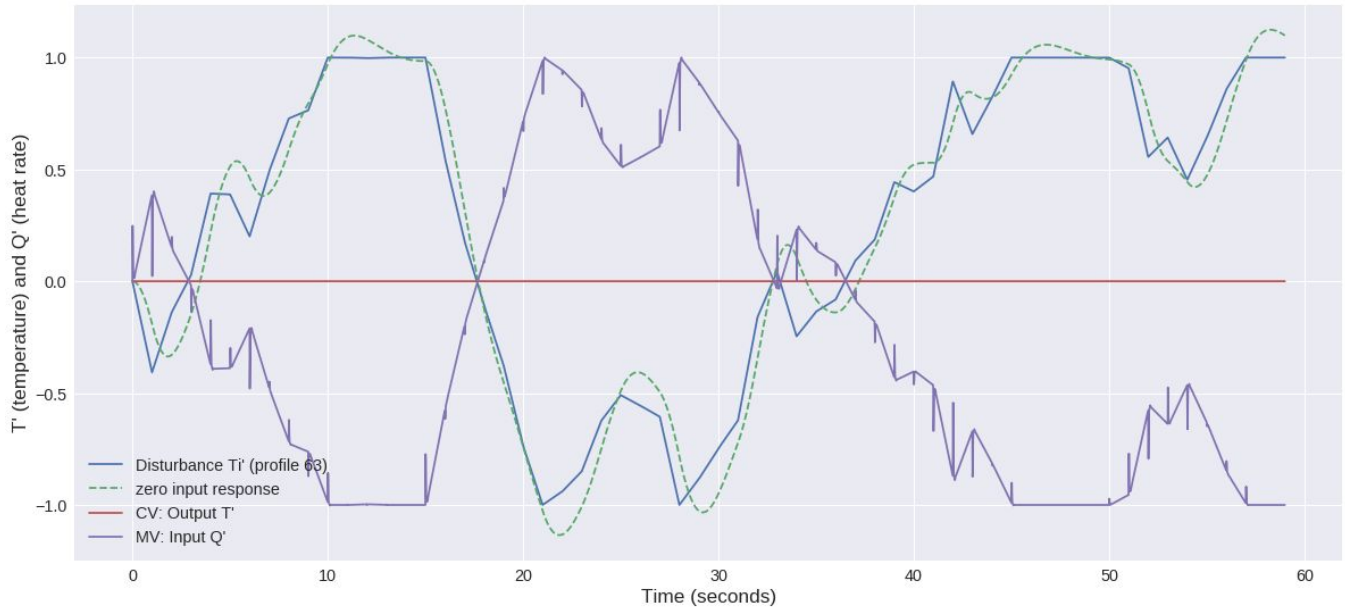

Disturbance Ti' (profile 58)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

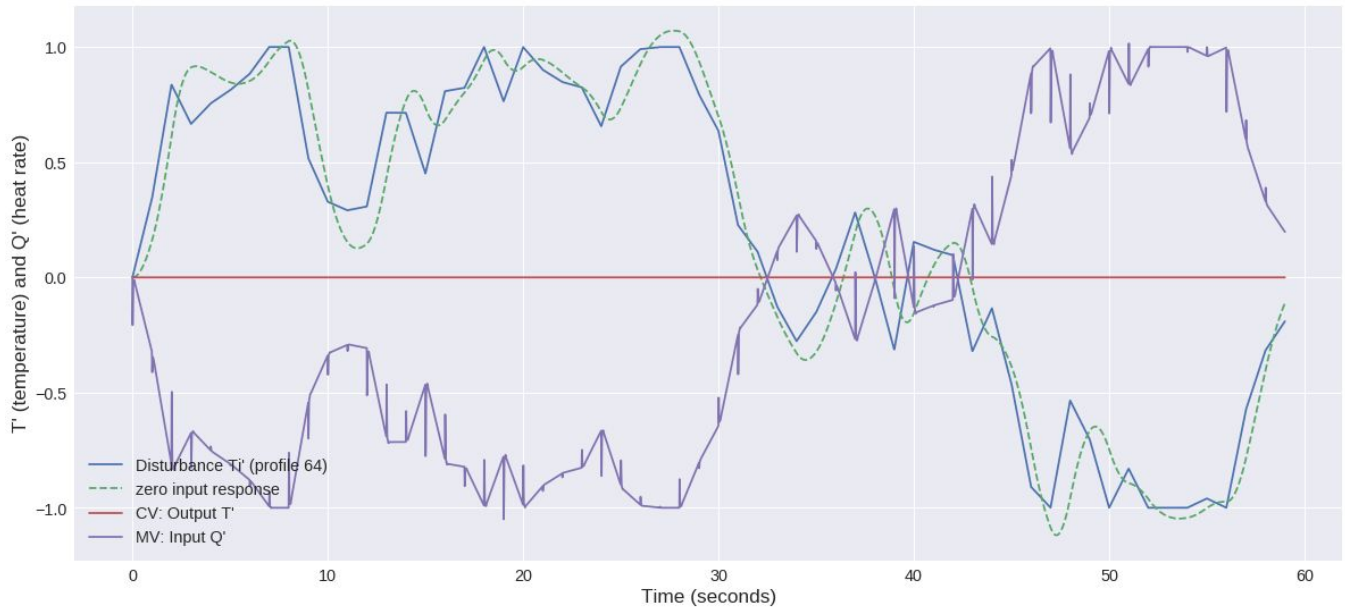

Disturbance Ti' (profile 59)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

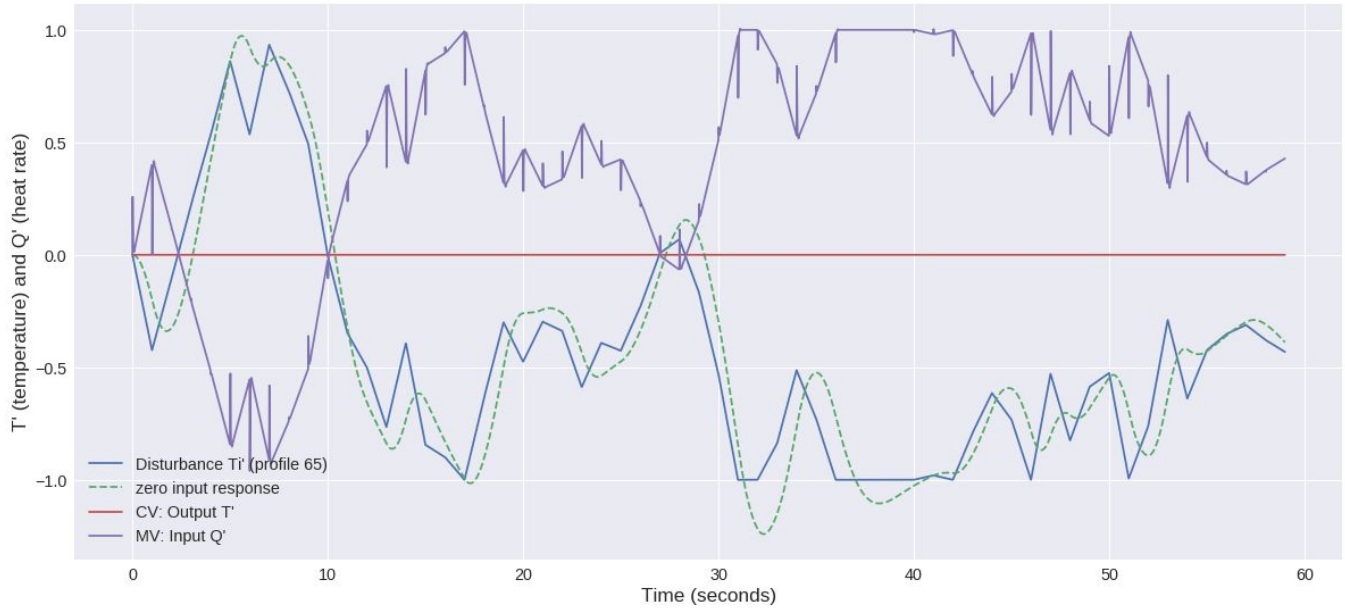

Disturbance Ti' (profile 60)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

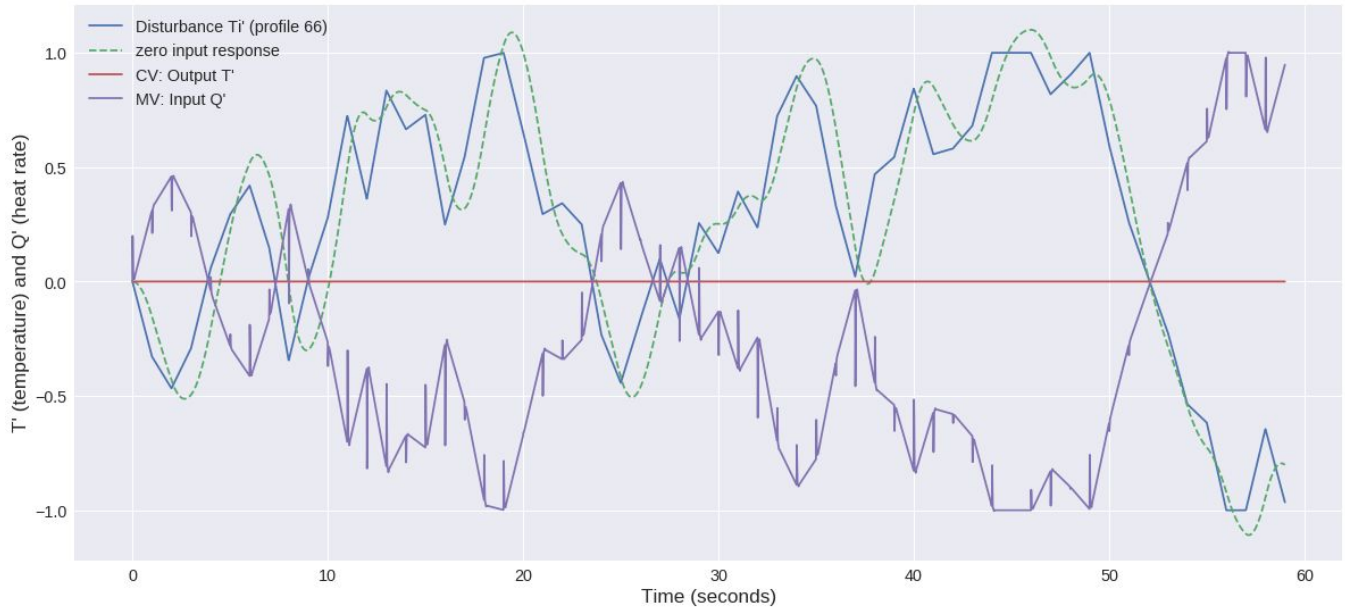

Disturbance Ti' (profile 61)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

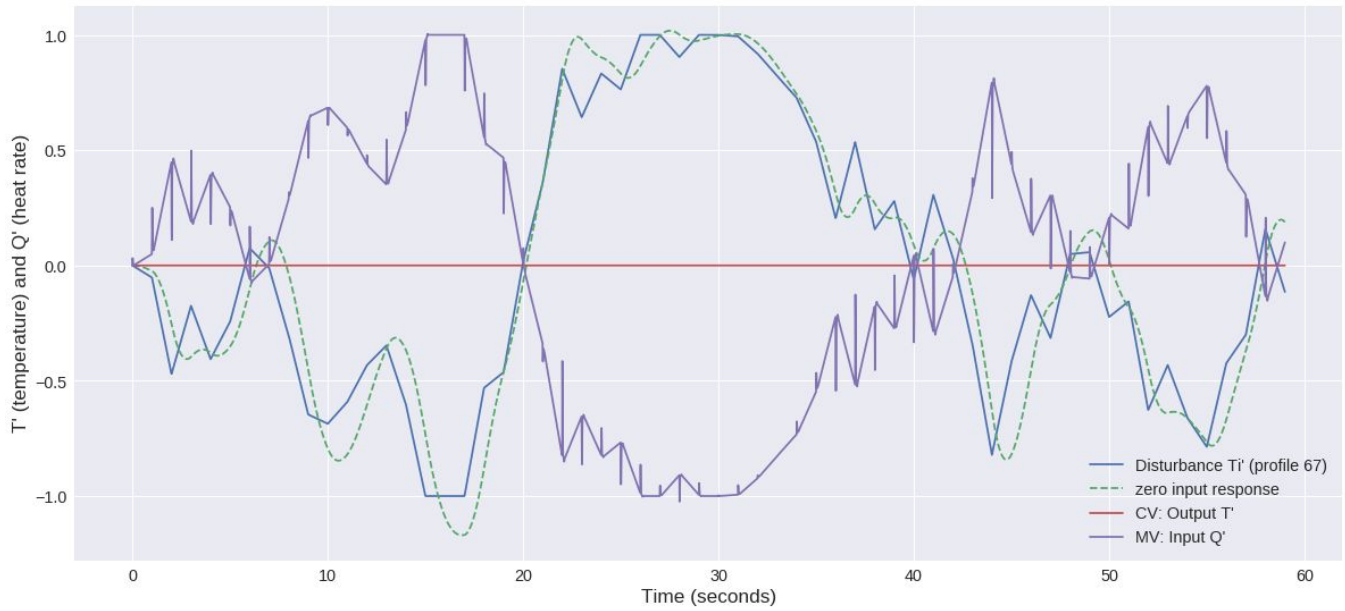

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 62)
zero input response
CV: Output T'
MV: Input Q'

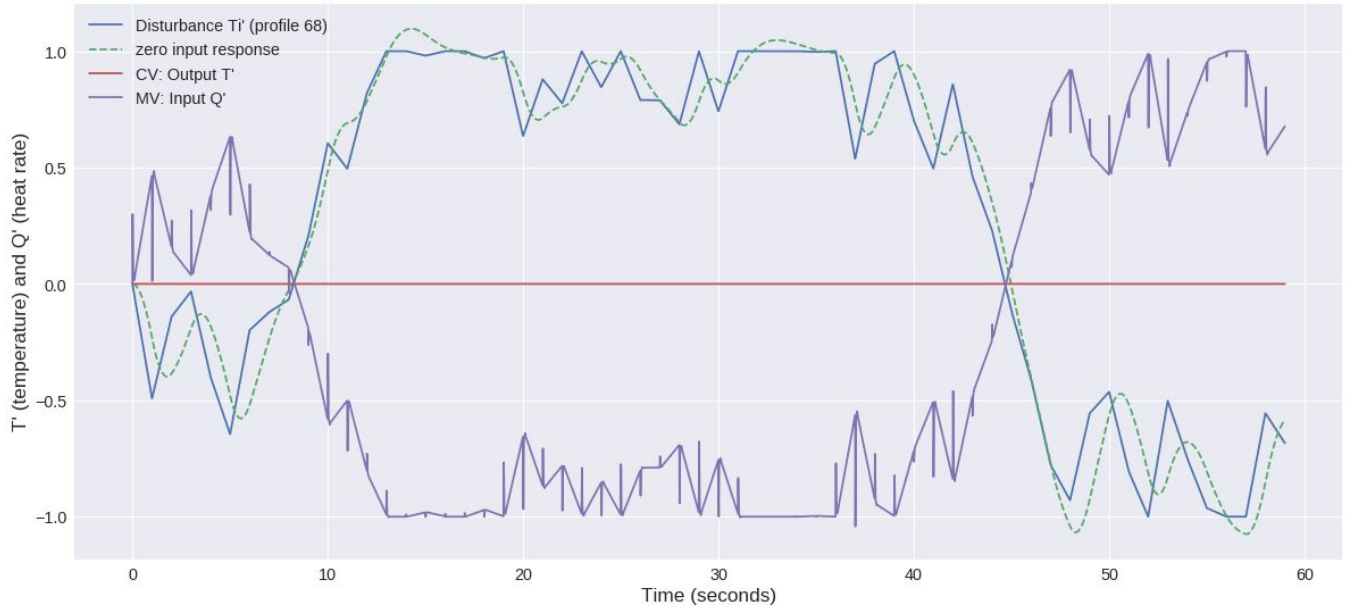

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 63)
zero input response
CV: Output T'
MV: Input Q'

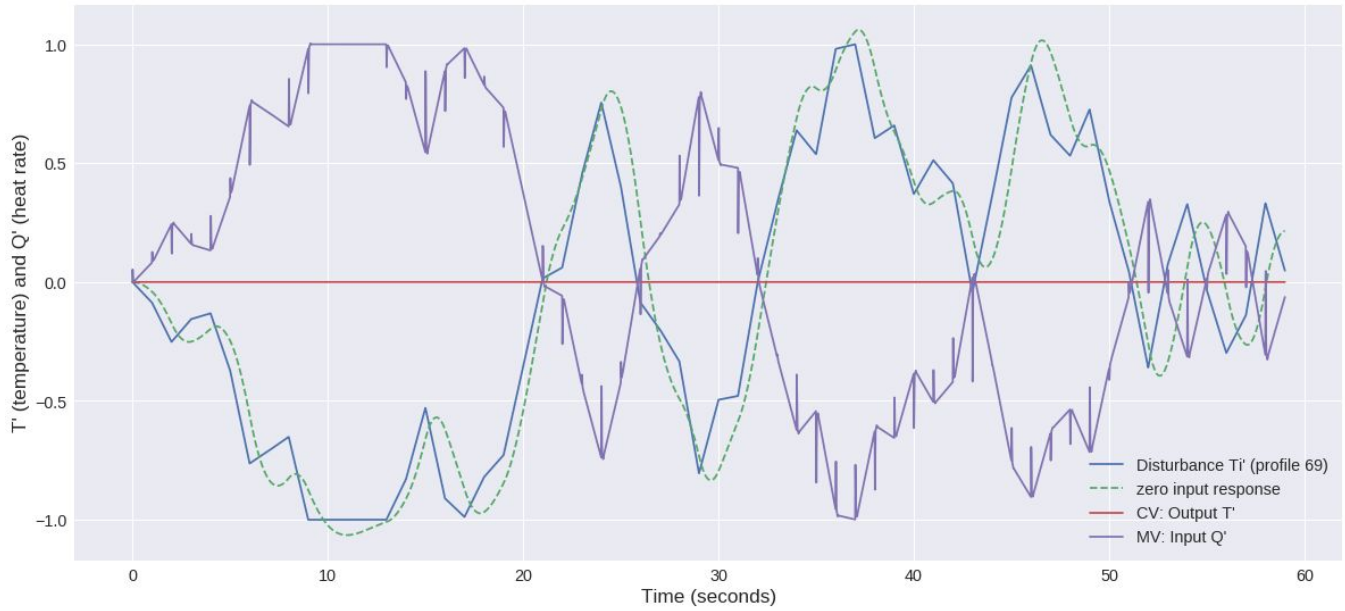

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 64)
zero input response
CV: Output T'
MV: Input Q'

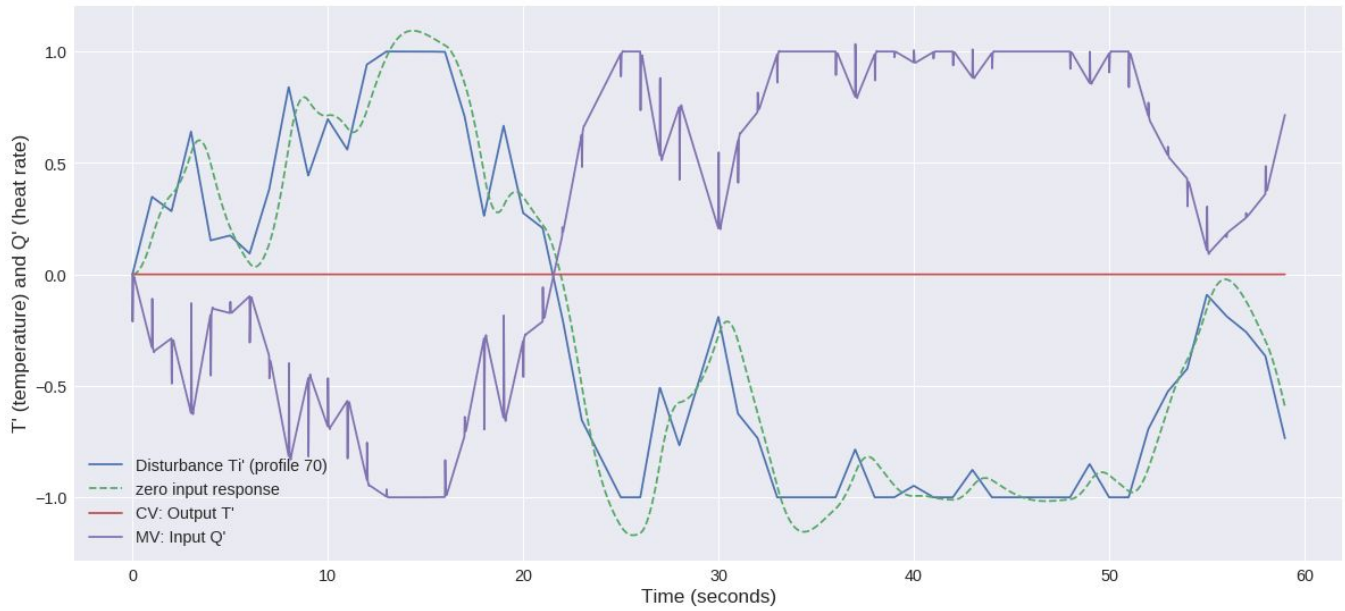

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 65)
zero input response
CV: Output T'
MV: Input Q'

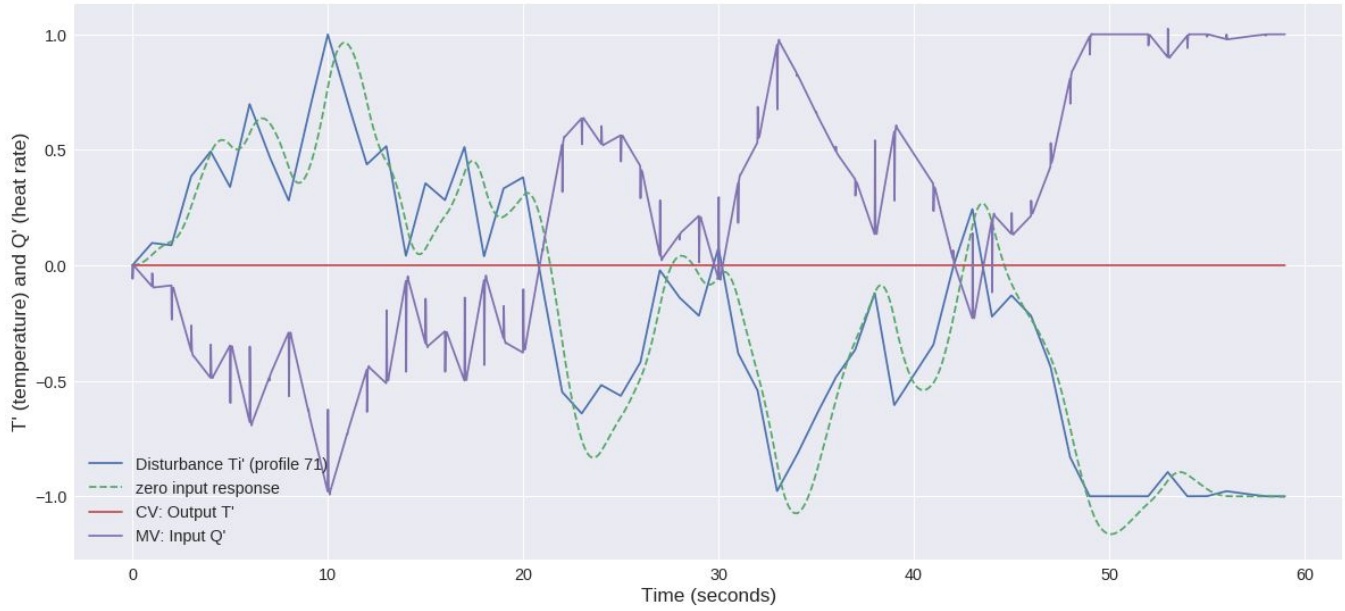

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 66)
zero input response
CV: Output T'
MV: Input Q'

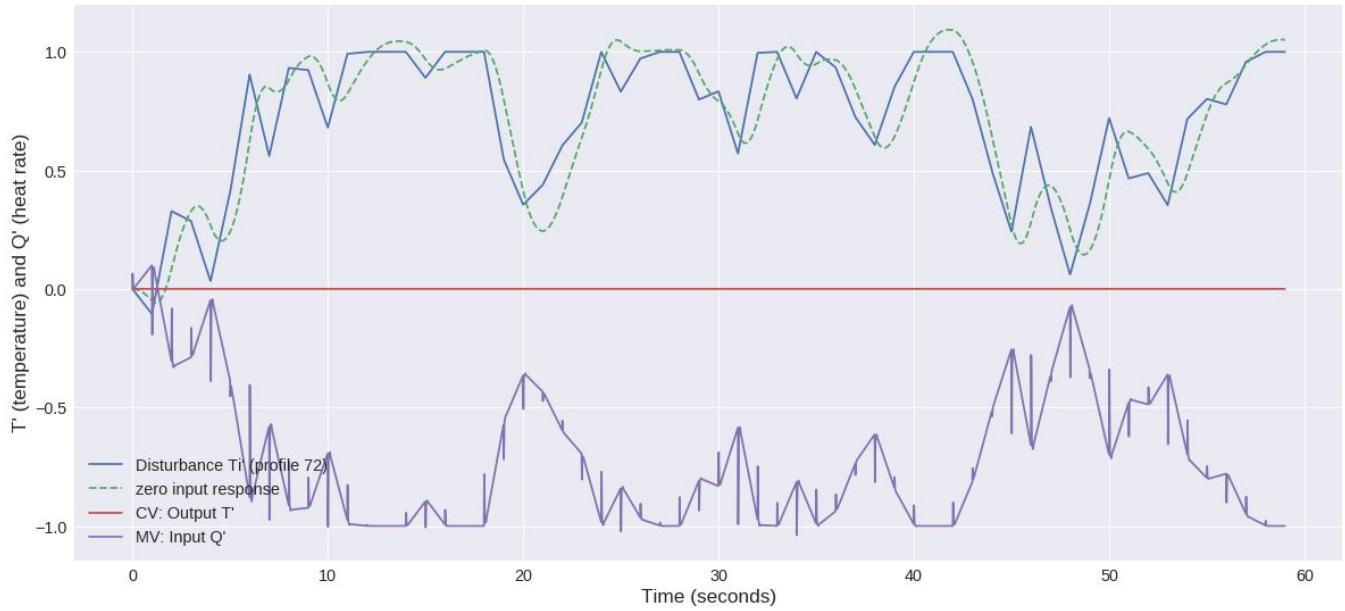

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 67)
zero input response
CV: Output T'
MV: Input Q'

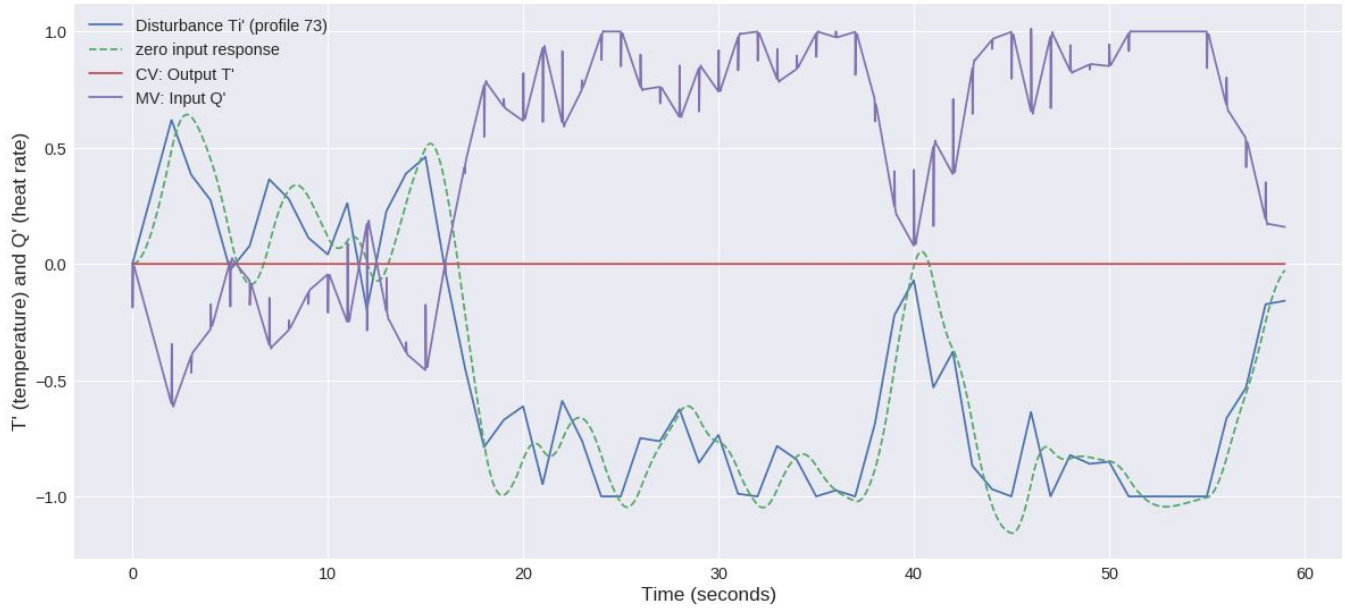

Disturbance Ti' (profile 68)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

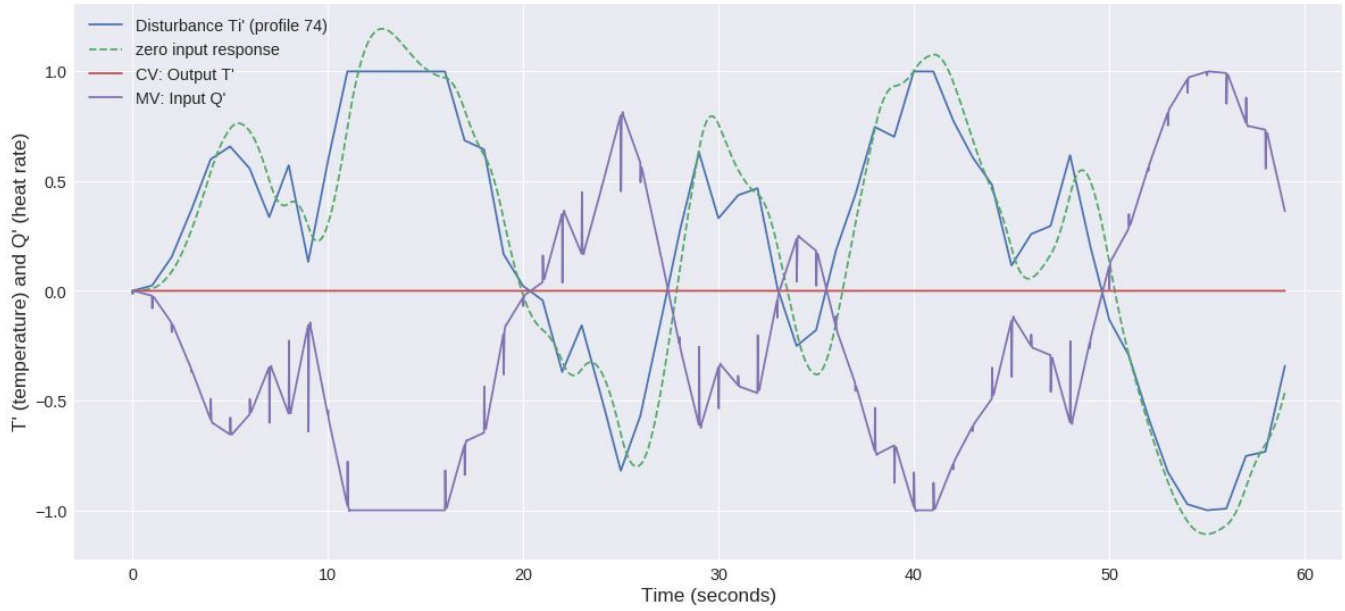

Disturbance Ti' (profile 69)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

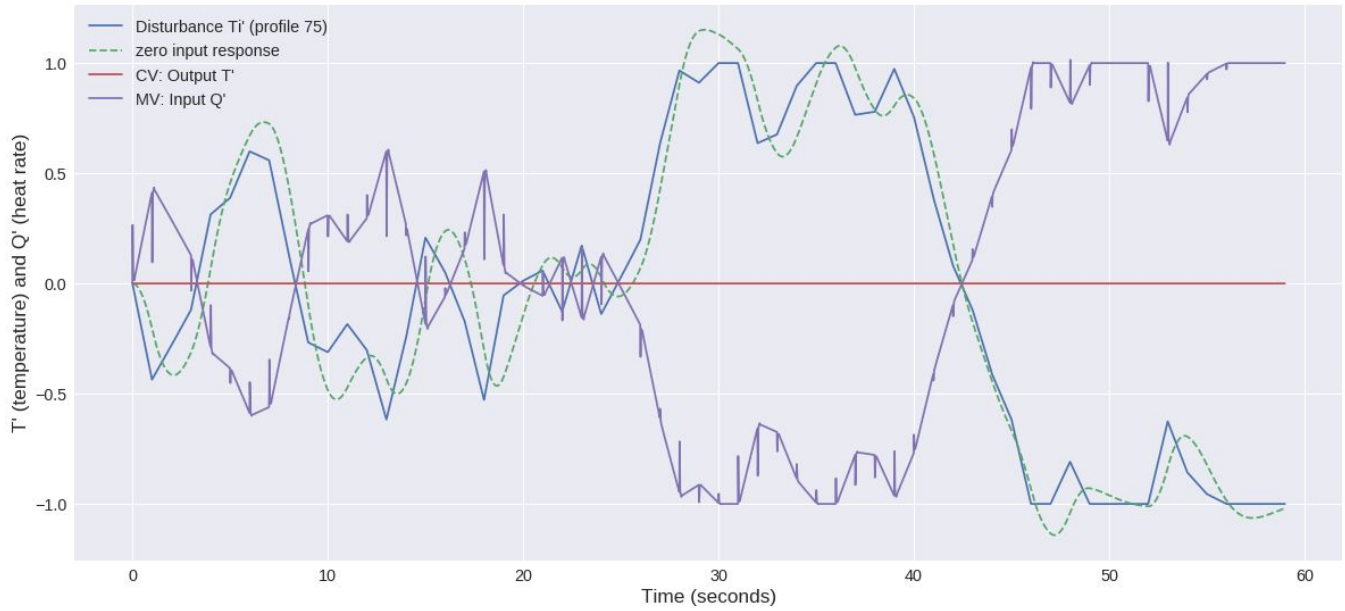

Disturbance Ti' (profile 70)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

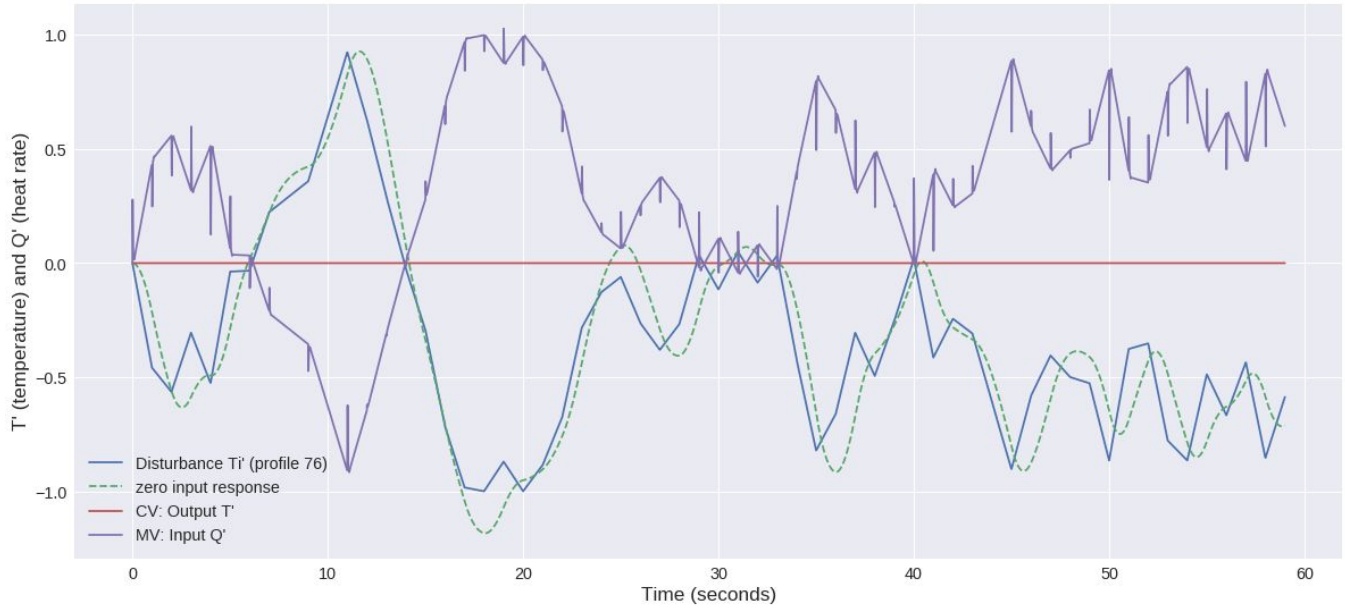

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 71)
zero input response
CV: Output T'
MV: Input Q'

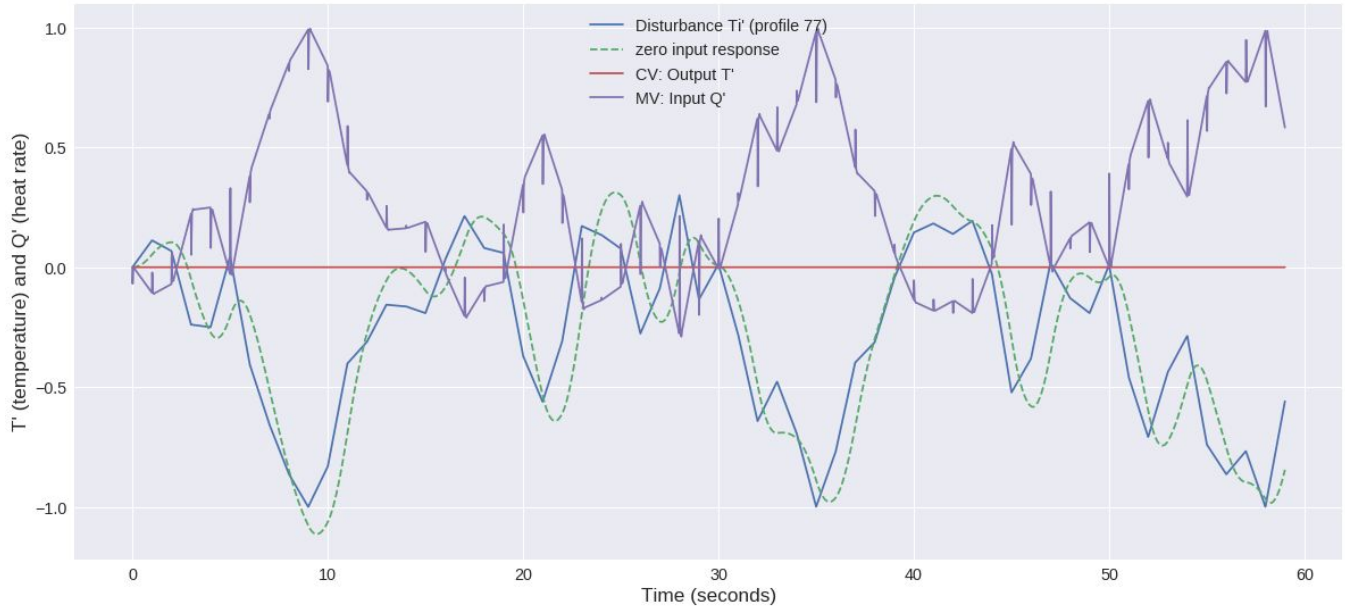

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 72)
zero input response
CV: Output T'
MV: Input Q'

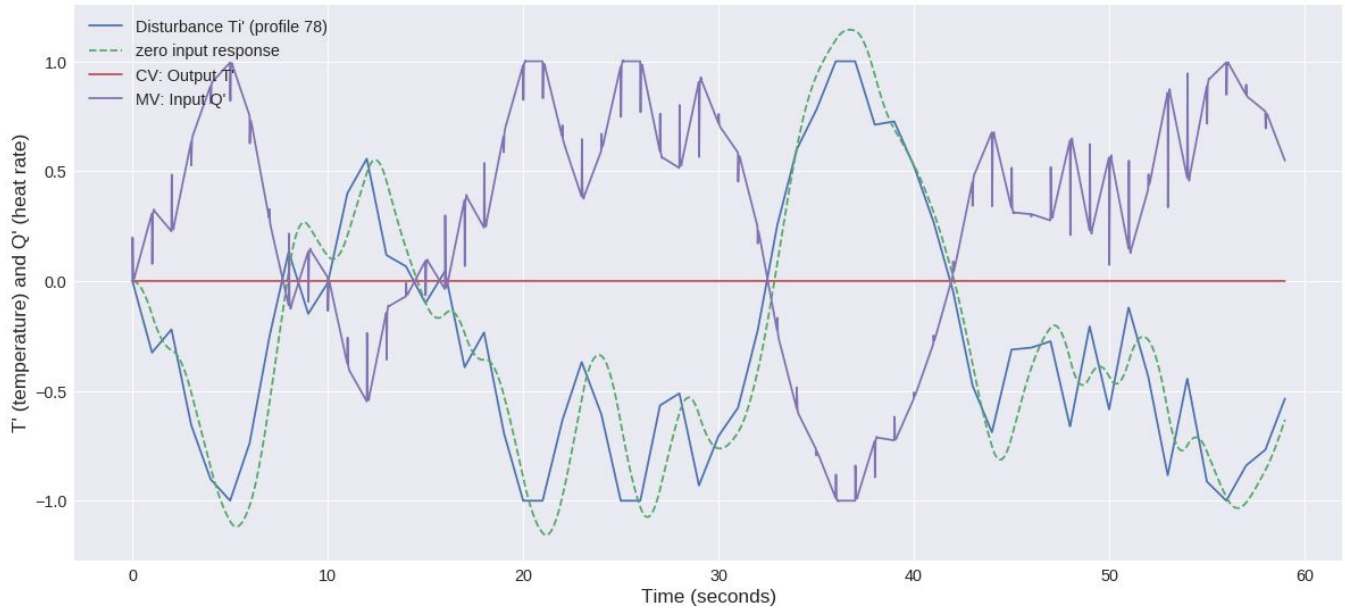

T' (temperature) and Q' (heat rate)
Time (seconds)
Disturbance Ti' (profile 73)
zero input response
CV: Output T'
MV: Input Q'

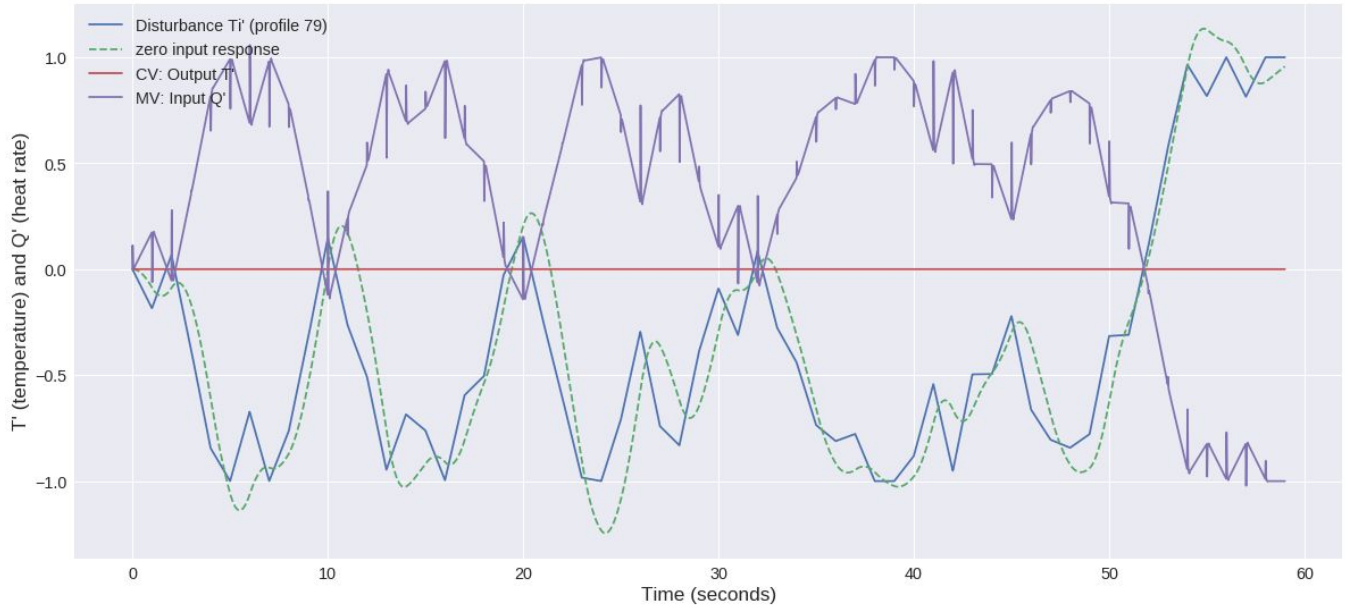

Disturbance Ti' (profile 74)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

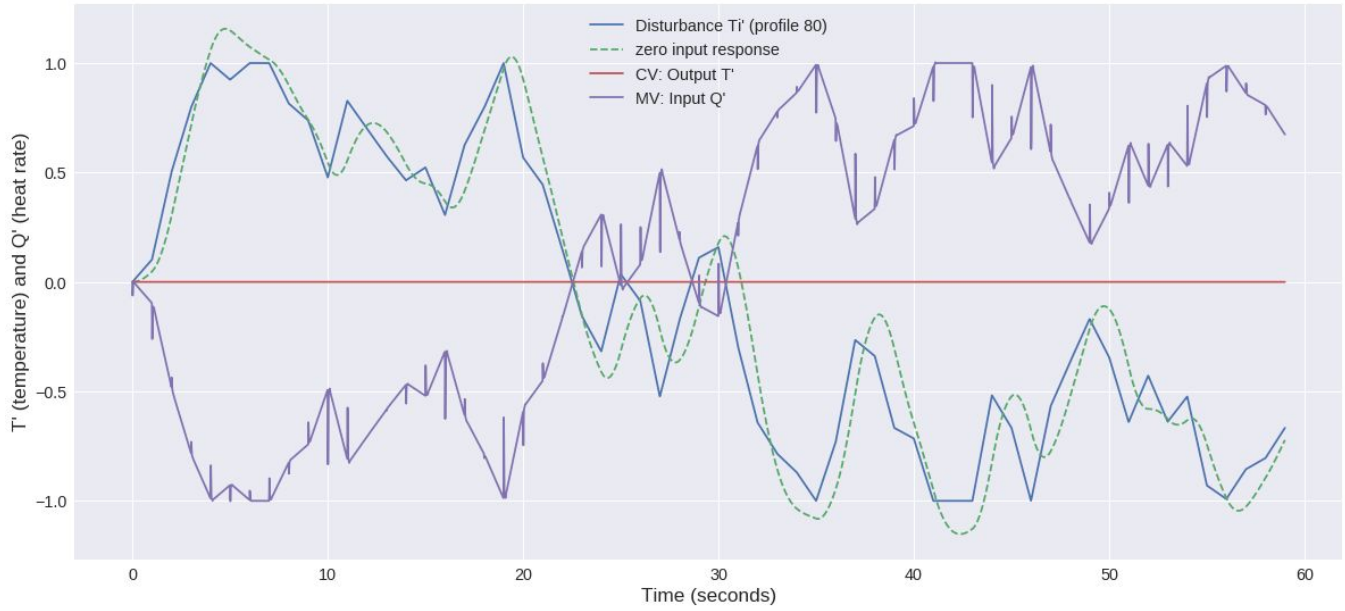

Disturbance Ti' (profile 75)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

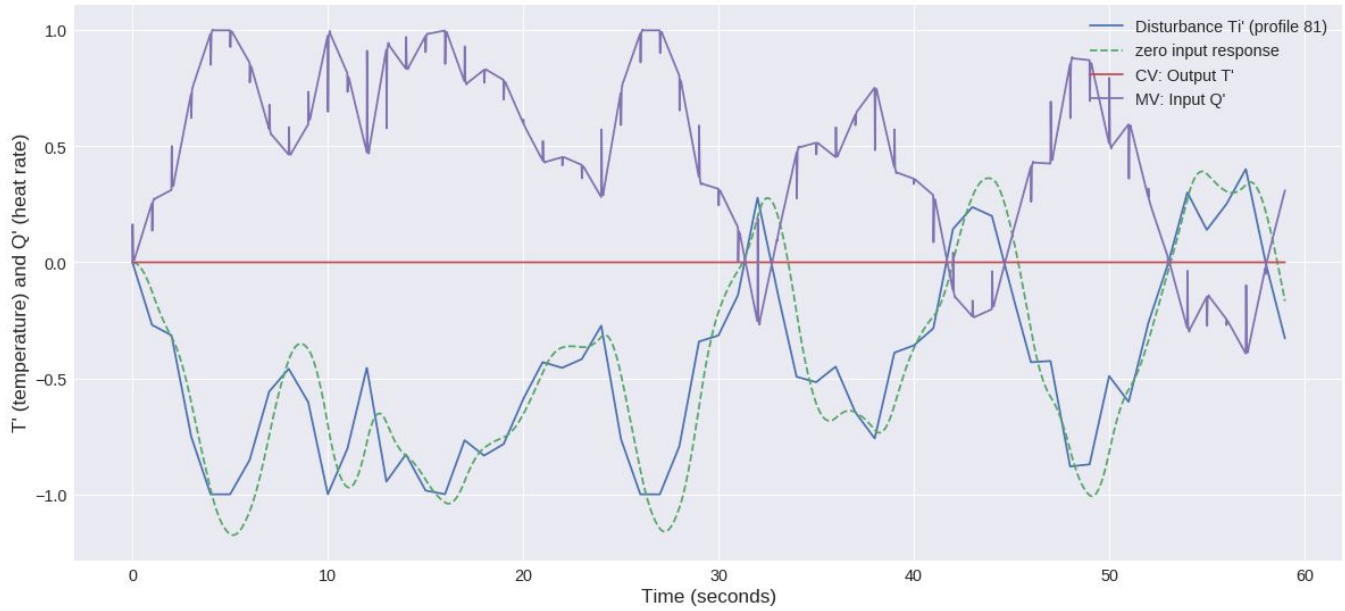

Disturbance Ti' (profile 76)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

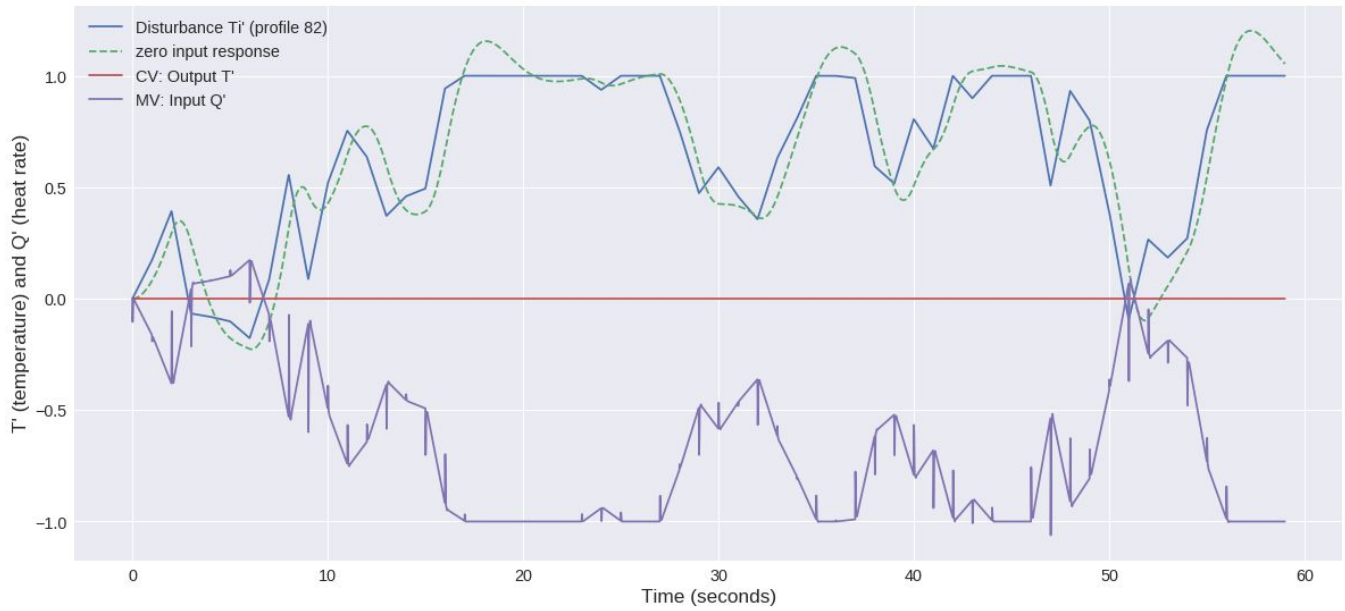

Disturbance Ti' (profile 77)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

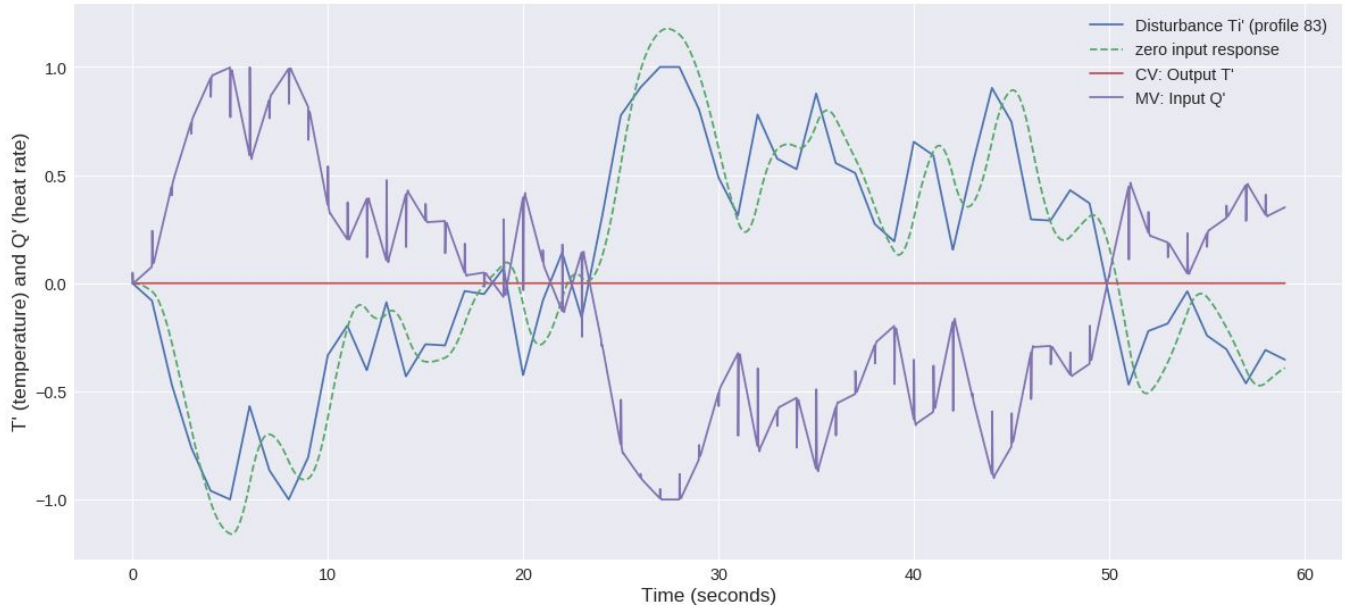

Disturbance Ti' (profile 78)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

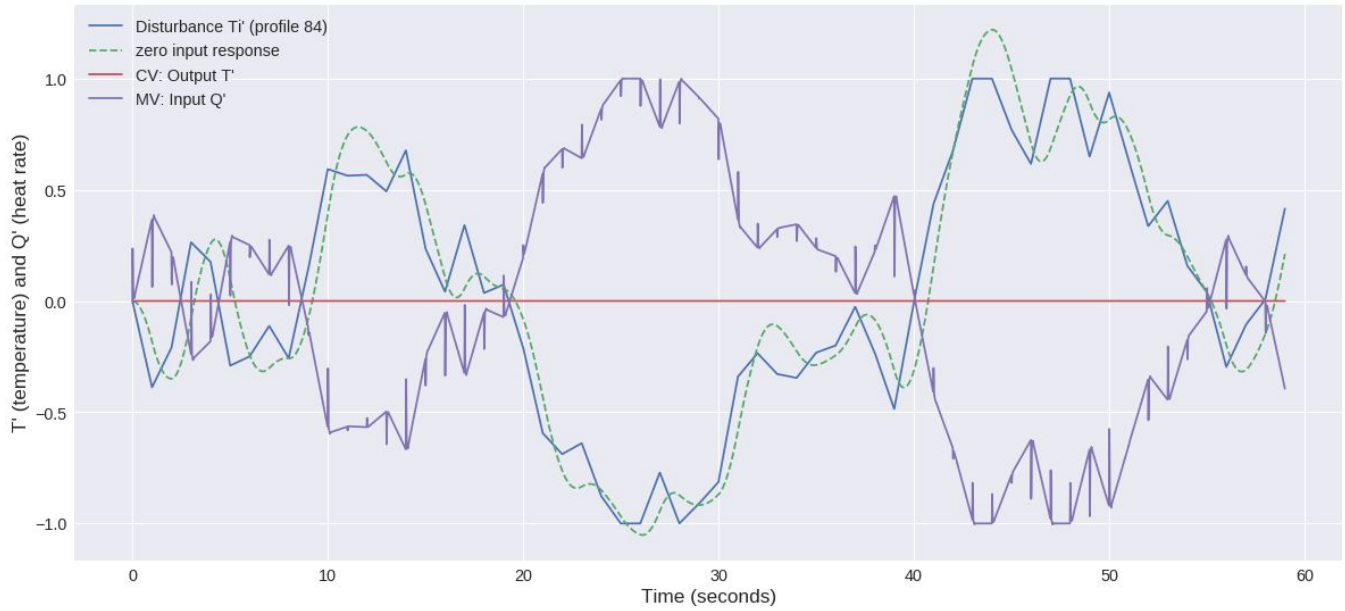

Disturbance Ti' (profile 79)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

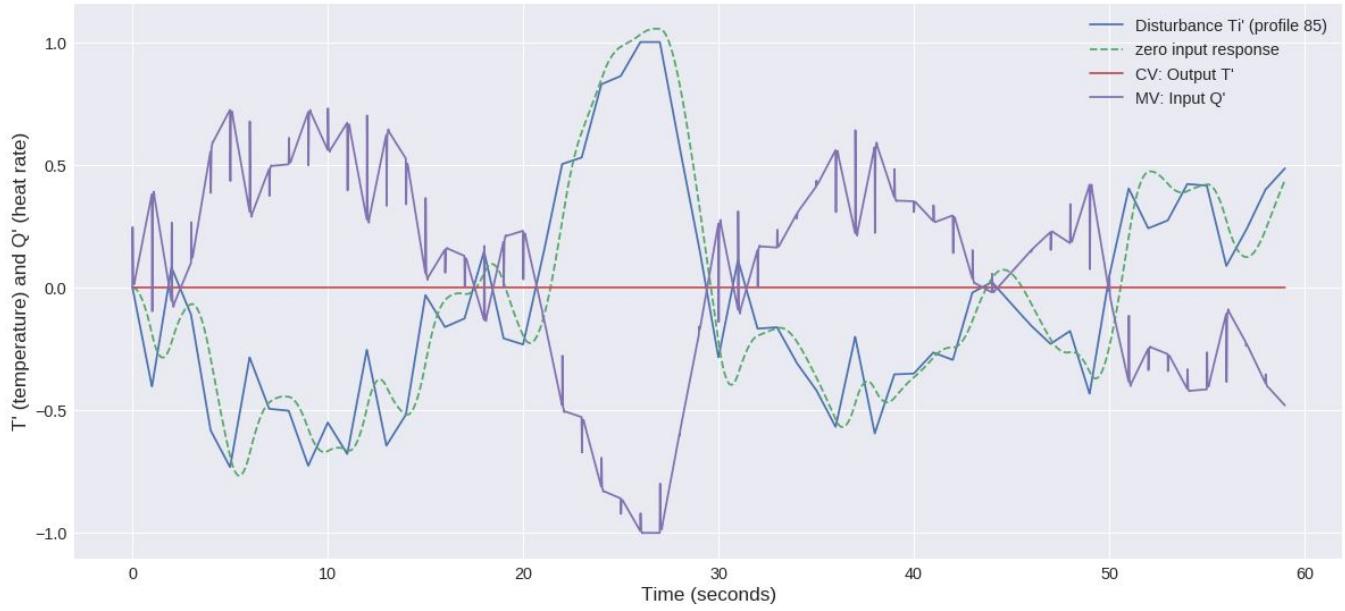

Disturbance Ti' (profile 80)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

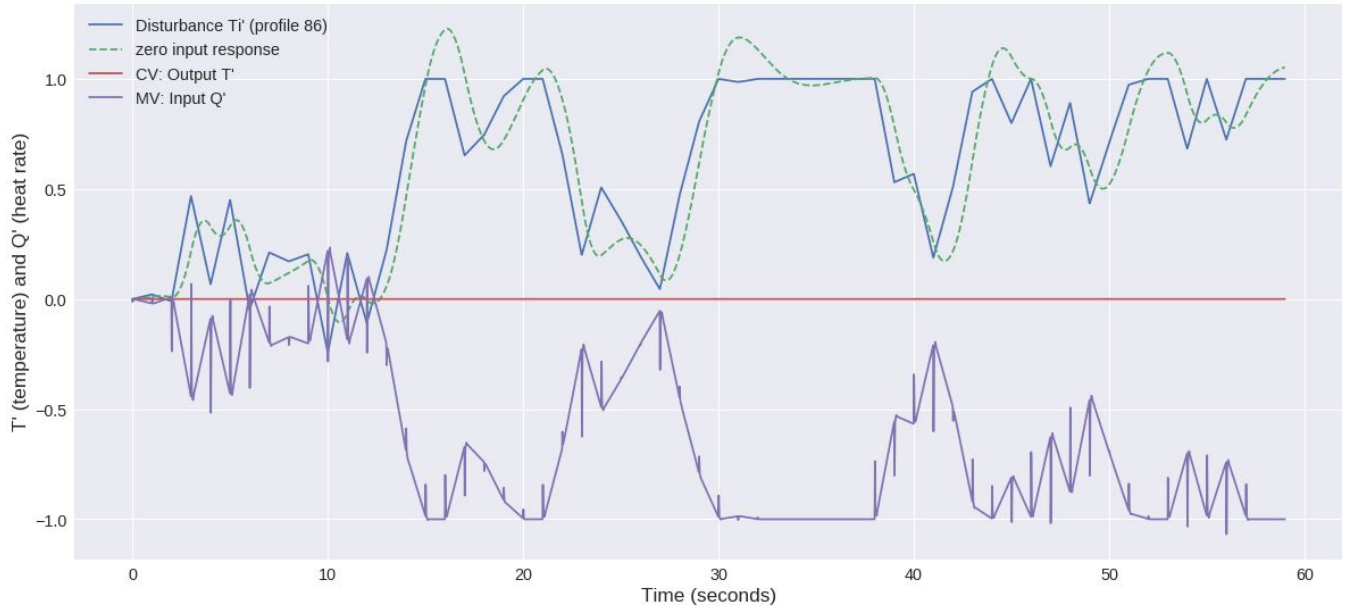

Disturbance Ti' (profile 81)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

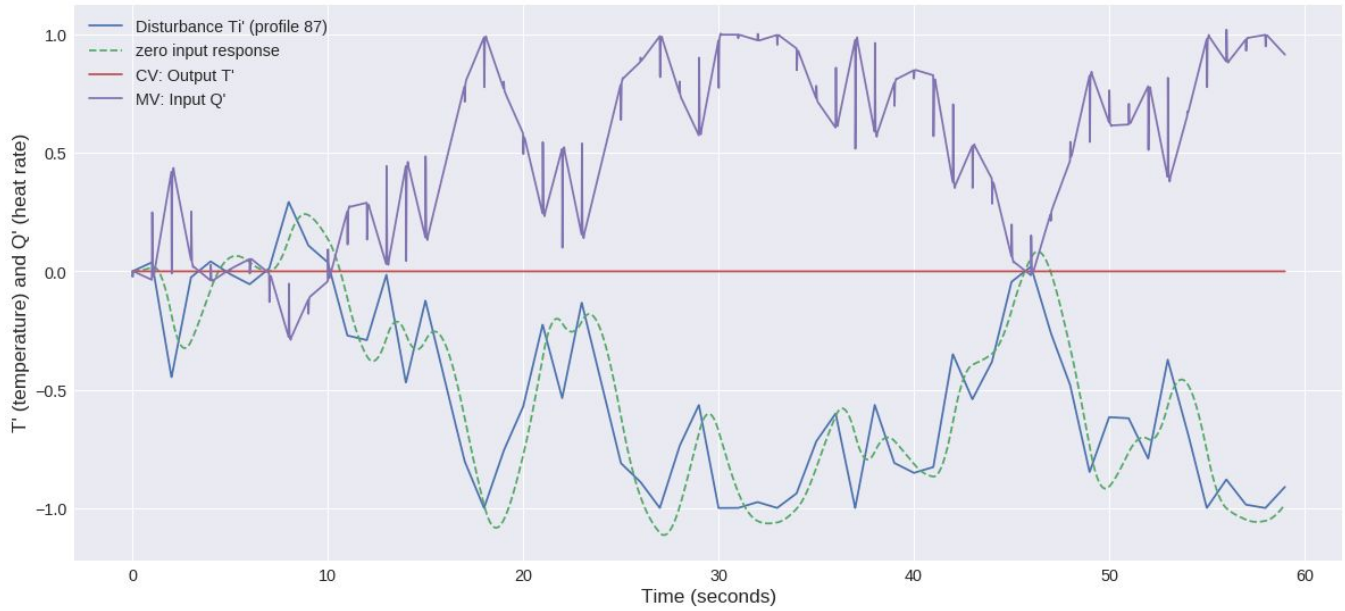

Disturbance Ti' (profile 82)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

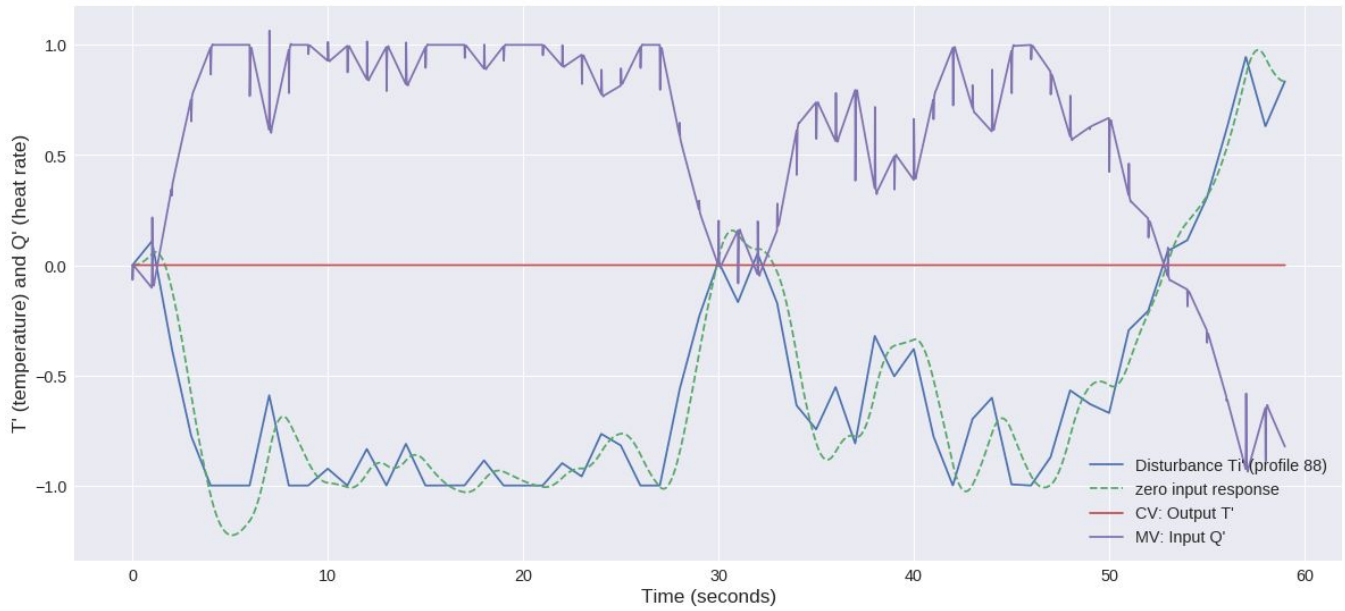

Disturbance Ti' (profile 83)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

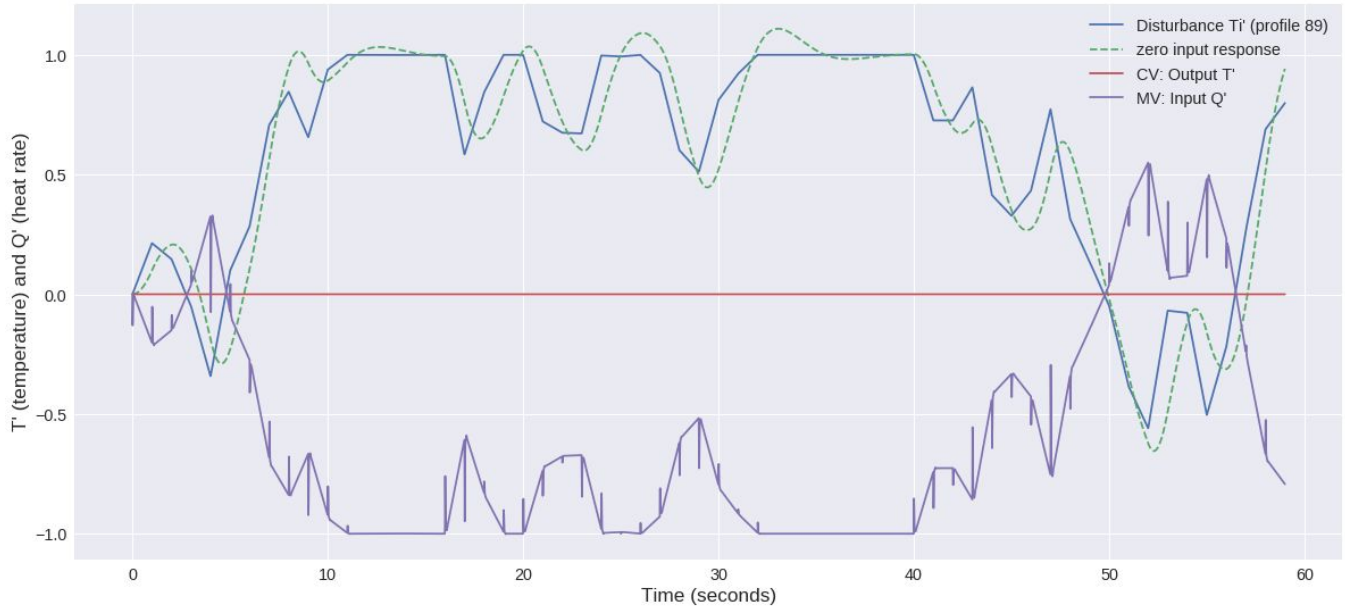

Disturbance Ti' (profile 84)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

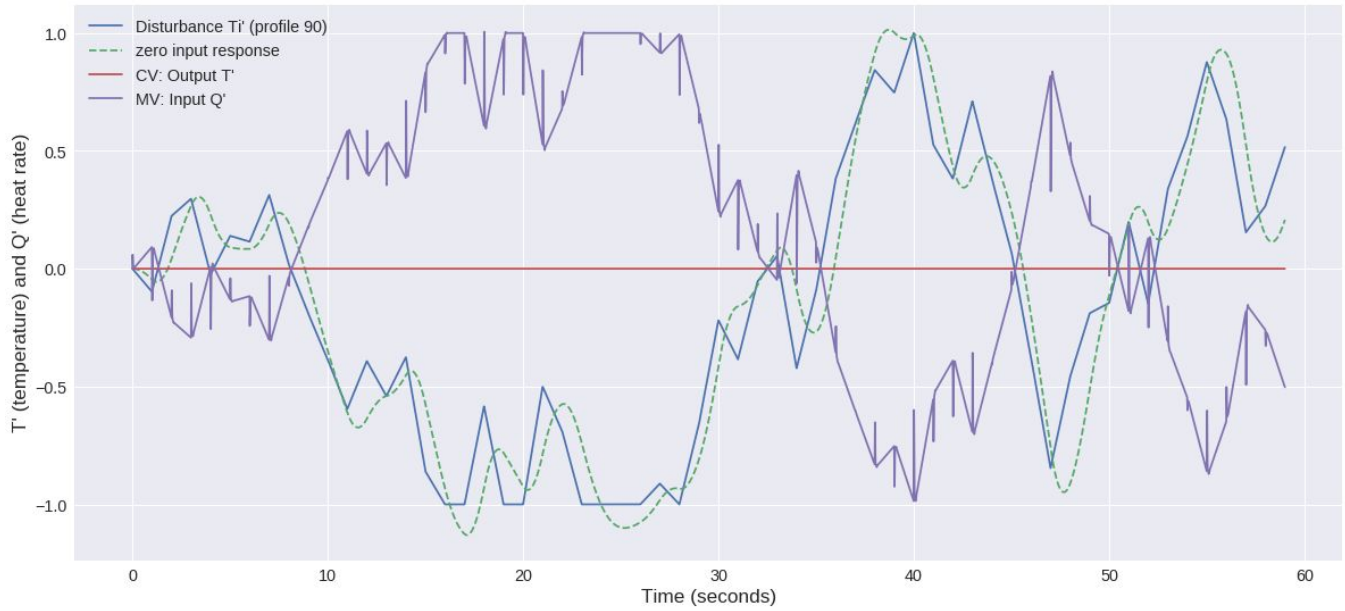

Disturbance Ti' (profile 85)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

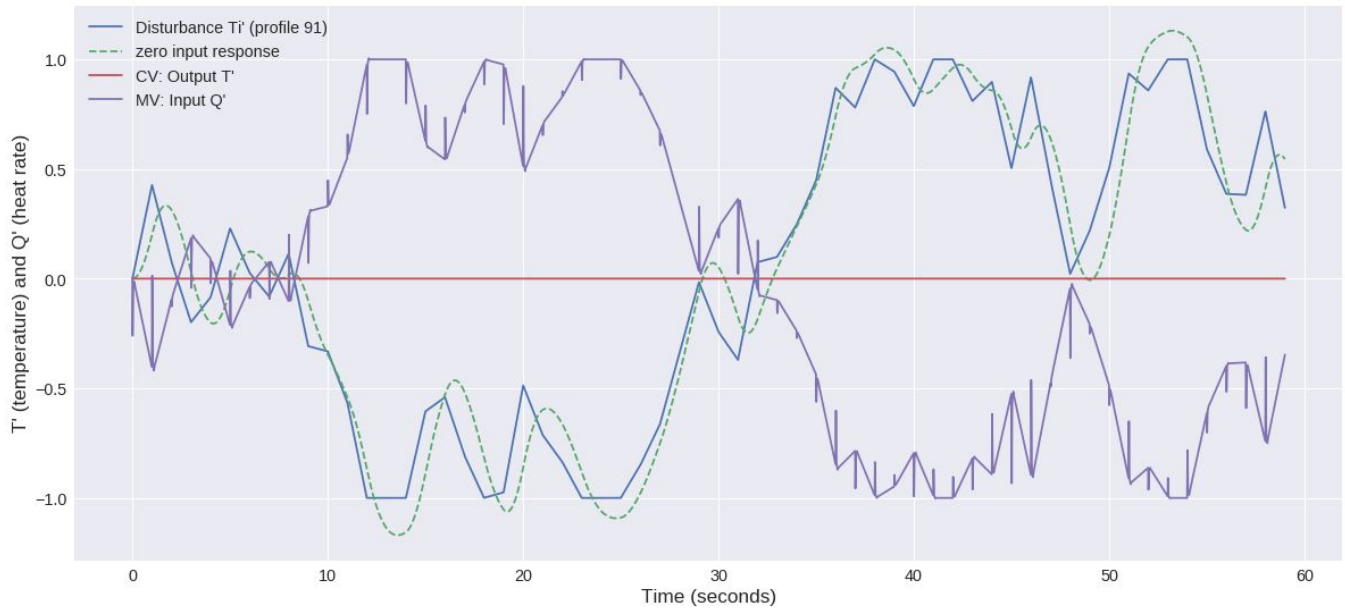

Disturbance Ti' (profile 86)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

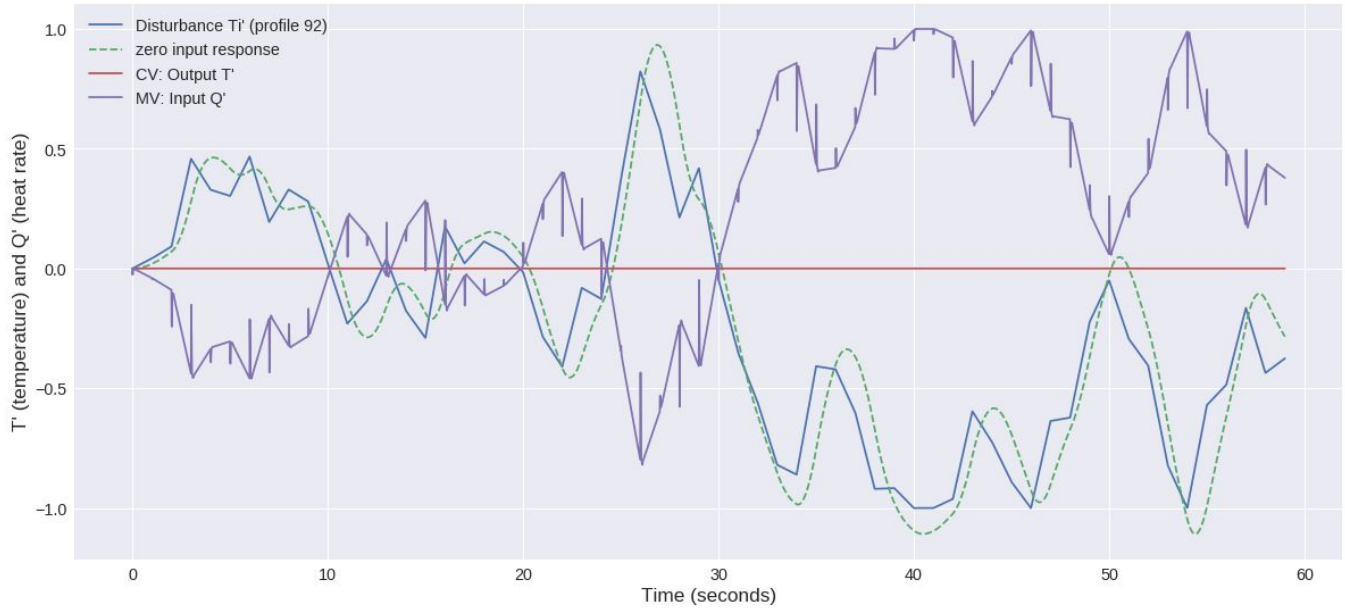

Disturbance Ti' (profile 87)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

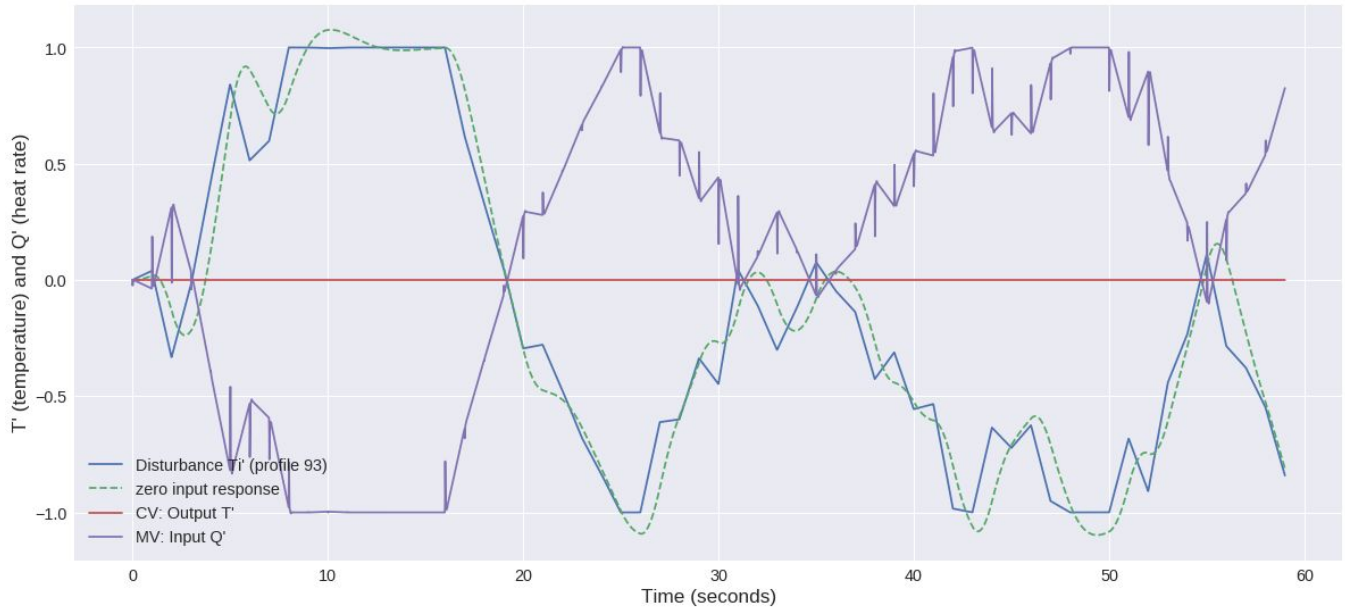

Disturbance Ti' (profile 88)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

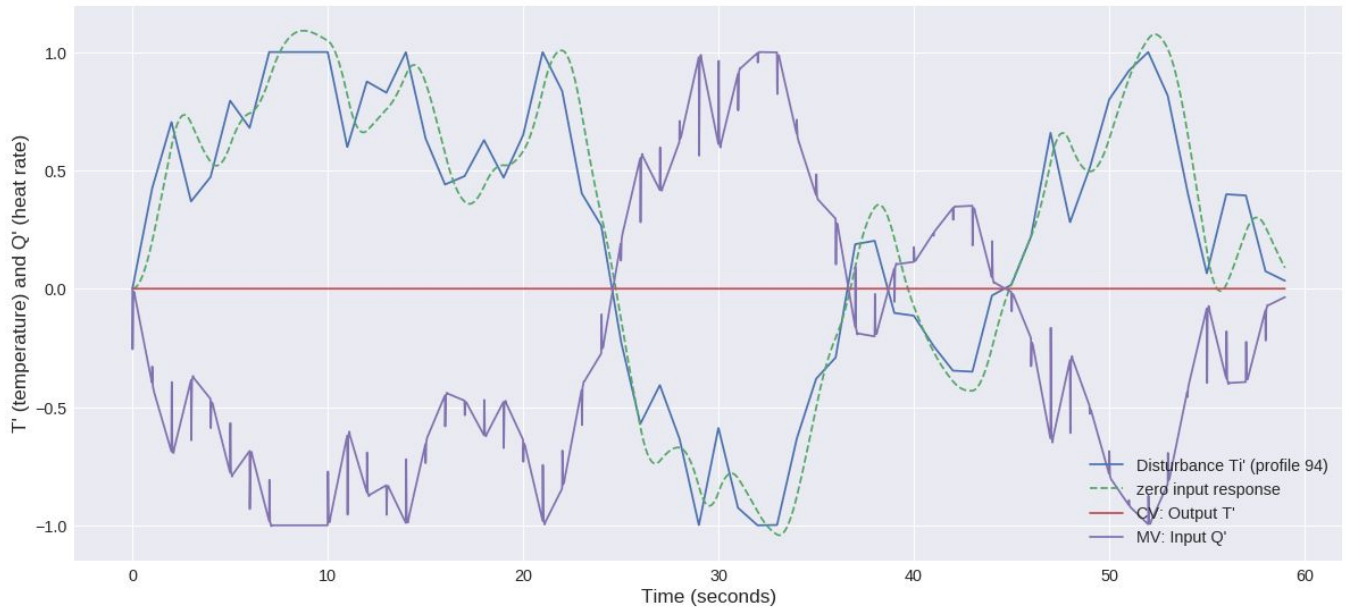

Disturbance Ti' (profile 89)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

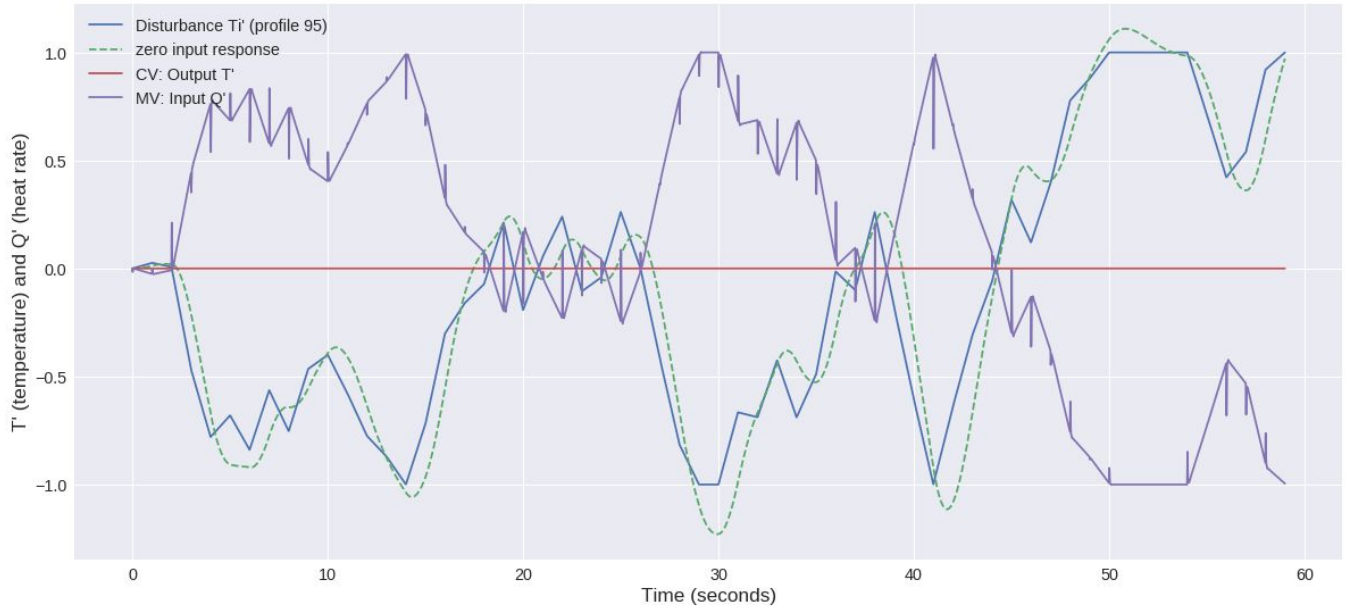

Disturbance Ti' (profile 90)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

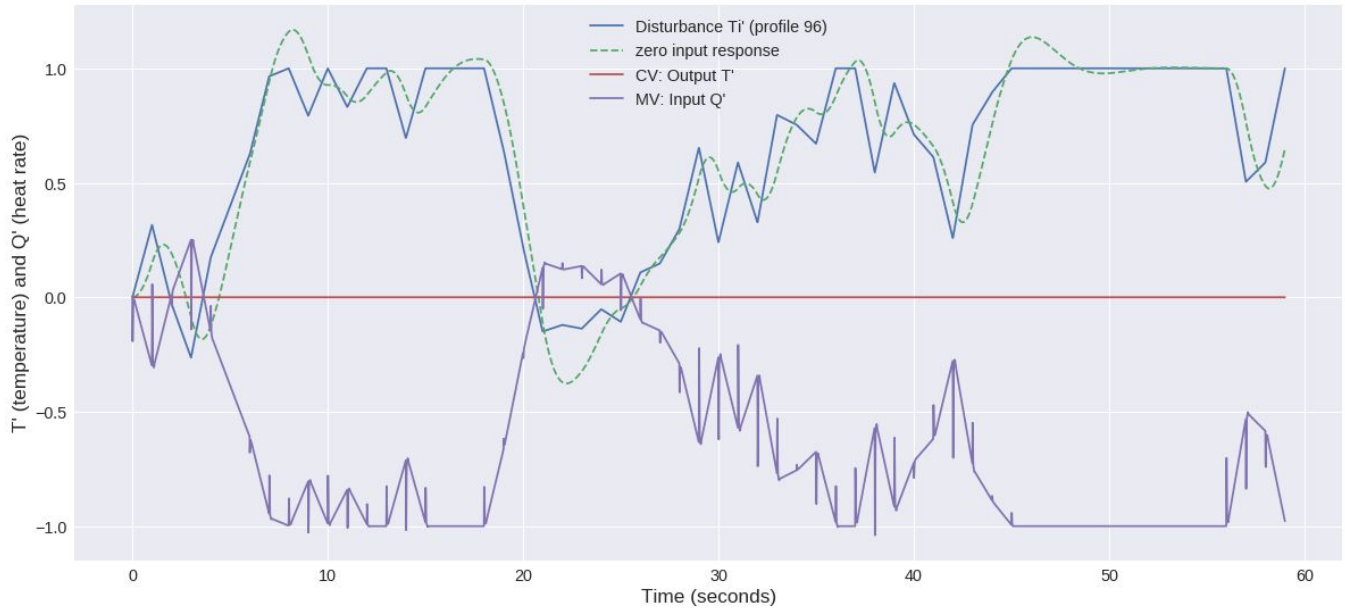

Disturbance Ti' (profile 91)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

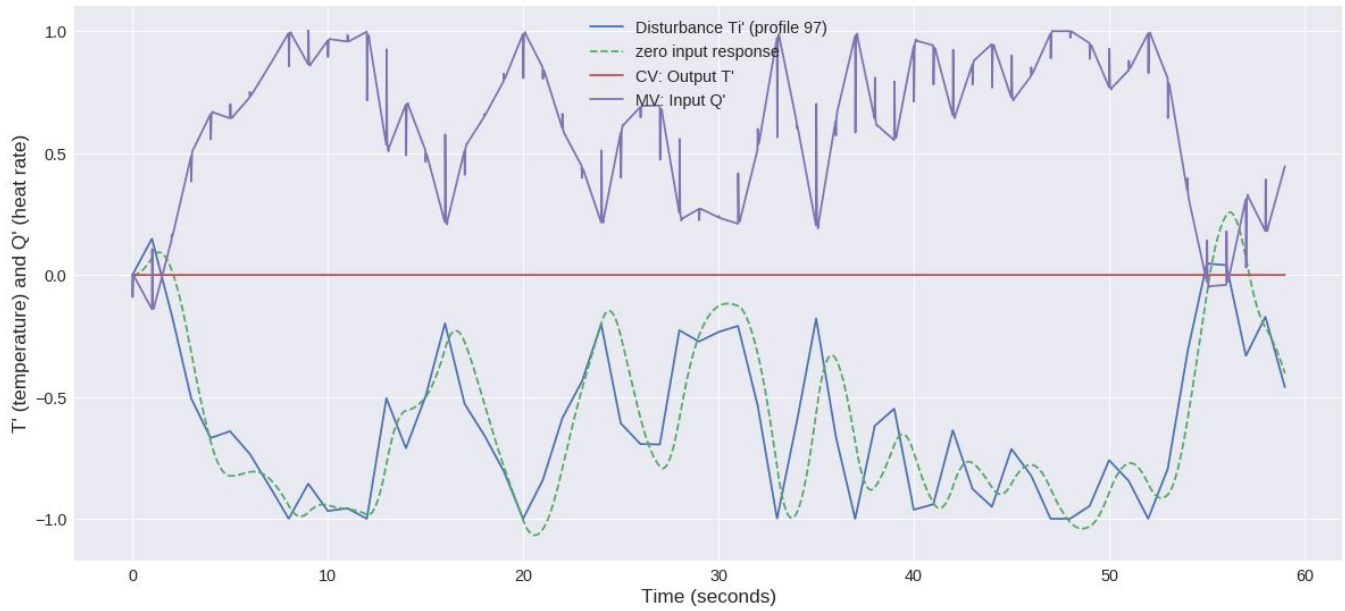

Disturbance Ti' (profile 92)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

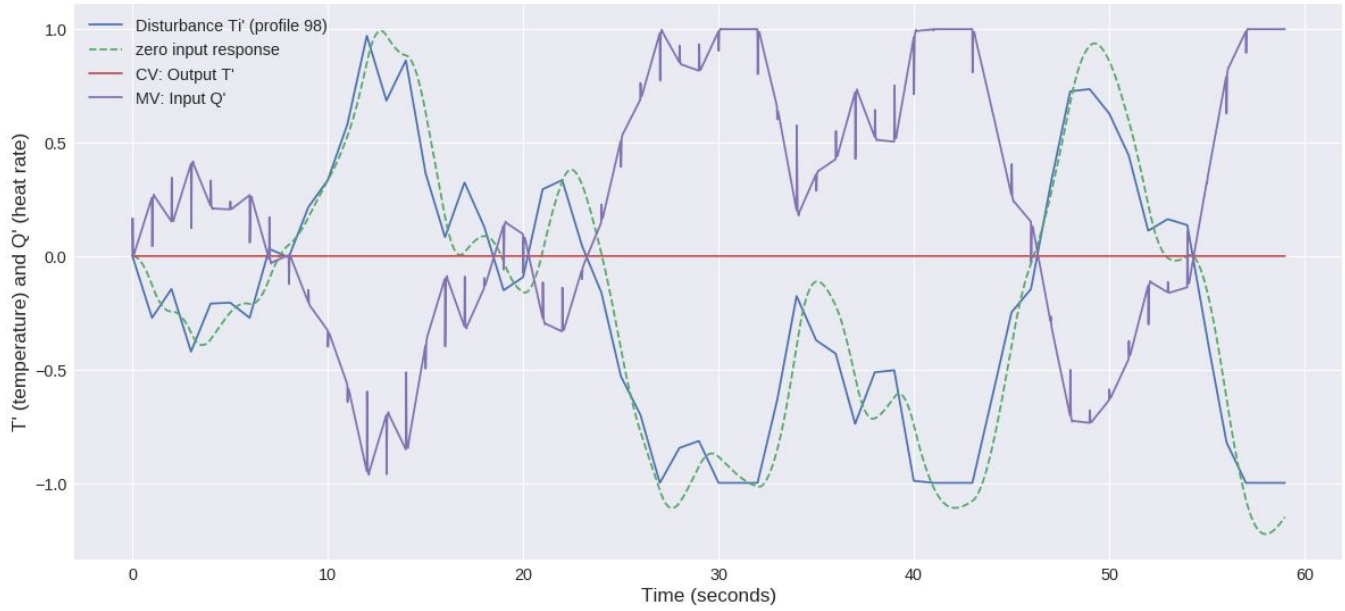

Disturbance Ti' (profile 93)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

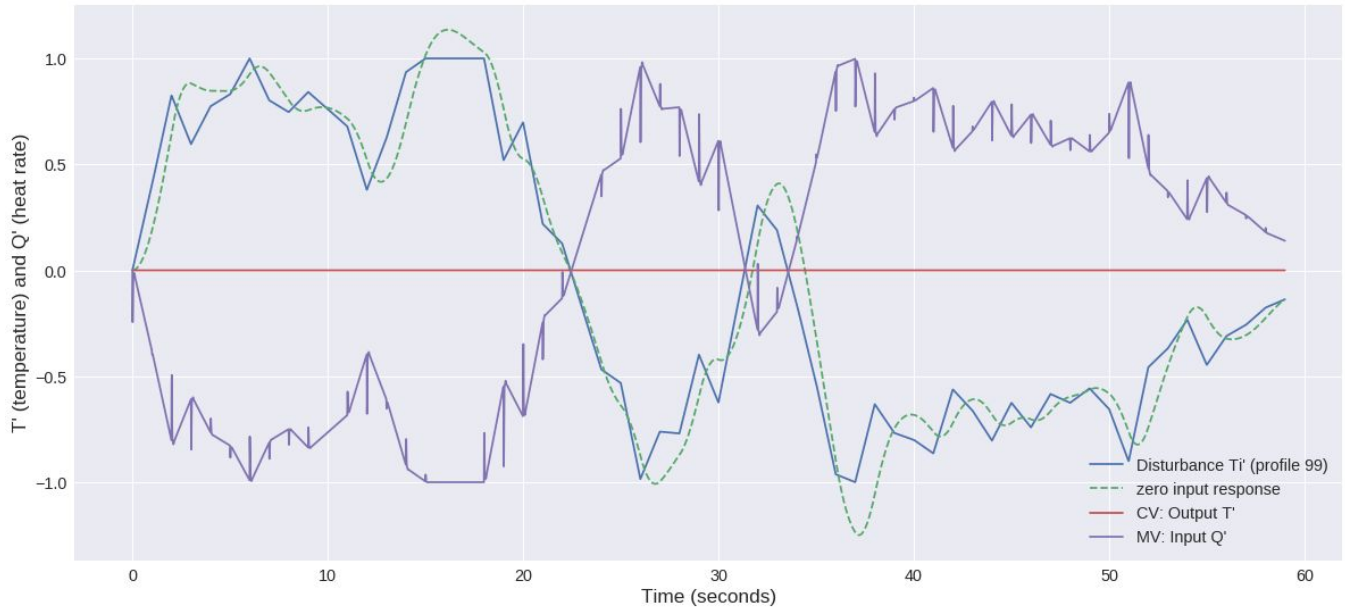

Disturbance Ti' (profile 94)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)


Disturbance Ti' (profile 95)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)


Disturbance Ti' (profile 96)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

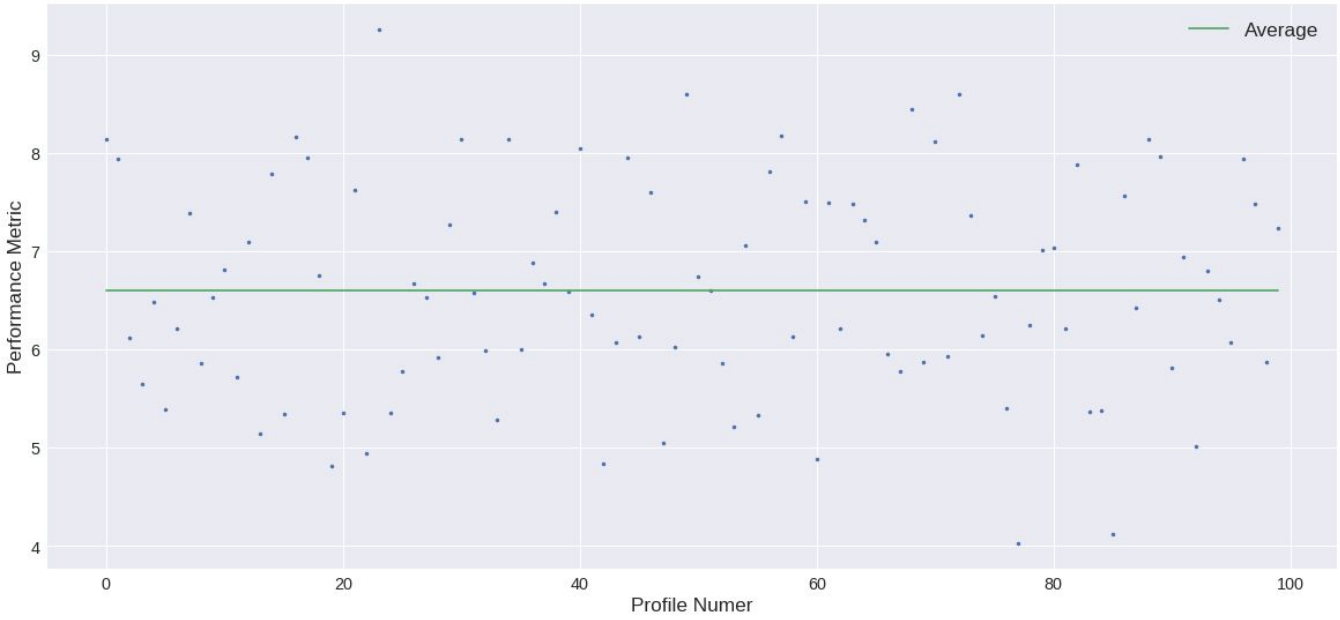

Disturbance Ti' (profile 97)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)


Disturbance Ti' (profile 98)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)


Disturbance Ti' (profile 99)
zero input response
CV: Output T'
MV: Input Q'
T' (temperature) and Q' (heat rate)
Time (seconds)

Performance Metric for each disturbance profile:

```
Minimum value of Perf =  4.033307205120107
Maximum value of Perf =  9.255942643545449
Average value of Perf =  6.603490145150944
```



Python Code: All codes can be found in the project's dedicated GitHub.

Calculate Performance Metric:
https://github.com/TimHillmer/CHBE356-data/blob/master/Results/**Verify_(Yankai)**.ipynb
This code is nearly identical to the one sent by Yankai. The file input lines have been mildly edited for group use. No numerical calculations have been changed.

Generate plots of CV, MV, DV with respect to time:
https://github.com/TimHillmer/CHBE356-data/blob/master/Results/**Skeleton_(Masterplan_for_plots)**.ipynb
All plots in this appendix came from running this code once. Calculations in this code are functionally equal to Yankai's.

Search for optimal tuning parameters:
https://github.com/TimHillmer/CHBE356-data/blob/master/Results/**random_search**.ipynb
This code randomly searches in user-defined intervals for good parameters, and downloads any that yield a performance metric below a user-defined threshold.

Raw Code Text
Three blank lines are inserted between cells in a notebook. (No cells contain three blank lines in a row.)

Verify (Yankai).ipynb

```
! pip install
git+https://github.com/python-control/python-control@601b58152080d89575cc677474ec7714e1a34ee
2
import control
import numpy as np
from scipy.interpolate import interp1d



NUM_PROFILES = 100
MAX_TIME = 60
T_sp = 0
K_c, tau_I, tau_D, tau_c = [5688.814947611792, 0.0002280222206125743, 0.4941522520819933,
0.006245557968152839]

# Disturbance profiles were uploaded to Github so markers can access them through this notebook.
Let's get them.
!rm -rf CHBE356-data
!git clone https://github.com/TimHillmer/CHBE356-data/

# obtain a numerically ordered list of the paths to the disturbance profiles
import glob
filenames = glob.glob('CHBE356-data/Disturbance_profiles/*.csv')
filenames.sort()

# obtain an array containing all disturbance profiles (data in [profile#, timevalue] format)
import pandas as pd
df = pd.concat([pd.read_csv(files).transpose() for files in filenames], ignore_index = True)
# each disturbance profile is a *row*
Ti_data = df.values



def simulate(i, K_c, tau_I, tau_D, tau_c):
    time_data = np.linspace(0, MAX_TIME-1, MAX_TIME)
    T_i_data = Ti_data[i]
    f = interp1d(time_data, T_i_data)
    points_per_seond = 100
    time = np.linspace(0, MAX_TIME-1, MAX_TIME*points_per_seond)
    T_i = f(time)
```

```
    s = control.tf([1,0],[0,1])
    G_p = 1/(s**2 + s + 1)
    G_d = (s+1)/(s**2 + s + 1)
    G_c = K_c * (1+ 1/(tau_I*s) + (tau_D*s)/(tau_c*s + 1))
    sys_D = G_d / (1 + G_p * G_c)

    Tsp = 0
    _, T, _ = control.forced_response(sys_D, time, T_i)
    _, Q, _ = control.forced_response(G_c, time, Tsp - T)
    error = (sum(abs(T)) + 0.2*sum(abs(Q)))/points_per_seond
    if sum(abs(T)>= 5) or sum(abs(Q)>= 5):
        error = 1e6
    return error



total_error = 0
for i in range(NUM_PROFILES):
    error = simulate(i, K_c, tau_I, tau_D, tau_c)
    total_error += error
print(f'Average error: {total_error / NUM_PROFILES}')
```

```
# Disturbance profiles were uploaded to Github so markers can access them through this notebook.
Let's get them.
!rm -rf CHBE356-data
!git clone https://github.com/TimHillmer/CHBE356-data/

# obtain a numerically ordered list of the paths to the disturbance profiles
import glob
filenames = glob.glob('CHBE356-data/Disturbance_profiles/*.csv')
filenames.sort()

# obtain an array containing all disturbance profiles (data in [profile#, timevalue] format)
import numpy as np
import pandas as pd
df = pd.concat([pd.read_csv(files).transpose() for files in filenames], ignore_index = True)
# each disturbance profile is a *row*

# define Yankai's 100 point interpolation
from scipy.interpolate import interp1d
Ti_data = df.values
time_data = np.arange(len(Ti_data[0]))  # assume Ti' data is given at each second
f = interp1d(time_data, Ti_data)
points_per_second = 100
time = np.linspace(time_data[0], time_data[-1], len(time_data)*points_per_second)
TiPrimes = f(time)

# Any remaining install/imports here
! pip install
git+https://github.com/python-control/python-control@601b58152080d89575cc677474ec7714e1a34ee
2
import control
import matplotlib.pyplot as plt


def masterplan(KC, TauI, TauD, TauC, plots=True, summary=True, failwarning=True):
    s = control.tf([1,0],[0,1])  # Define the s variable
    Gd = (s+1) / (s**2+s+1)  # T' / Ti' (disturbance) transfer function
    Gp = 1 / (s**2+s+1)  # T' / Q' (process) transfer function
    Gc = KC*(1 + 1/(TauI*s) + TauD*s/(TauC*s+1))  # Controller transfer function

    Gtotal = Gd / (1 + Gp*Gc)  # no term for (temperature deviation) setpoint because it is always zero

    if plots or summary:
```

```
        plt.style.use('seaborn')  # plot style for if 'plots' or 'summary' are marked True
        params = {'legend.fontsize': 'x-large',
          'figure.figsize': (20, 9),
         'axes.labelsize': 'x-large',
         'axes.titlesize':'x-large',
         'xtick.labelsize':'large',
         'ytick.labelsize':'large'}
        plt.rcParams.update(params)

    Perf = np.zeros(len(TiPrimes))  # array to store Perf for each disturbance profile
    for i in range(len(TiPrimes)):  # main loop for calculation and plotting
        # T' response including controller
        t, Tprime, x = control.forced_response(Gtotal, time, TiPrimes[i])

        # controller response to error (=input)
        t, Qprime, x = control.forced_response(Gc, time, -Tprime)

        Perf[i] = (sum(np.absolute(Tprime)) + 0.2*sum(np.absolute(Qprime)))/points_per_second
        # performance metric for each profile

        if plots == True: # plot results from the forced_repsonse calculations. Delete the dashed lines
when submitting.
            plt.plot(time, TiPrimes[i], label = f"Disturbance Ti' (profile {i})")  # disturbance signal

            # response to disturbance (as if there were no controller)
            t, zero, x = control.forced_response(Gd, time, TiPrimes[i])
            plt.plot(t, zero, '--', label = 'zero input response')

            plt.plot(t, Tprime, label = "CV: Output T'")  # output value used for performance metric
            plt.plot(t, Qprime, label = "MV: Input Q'")  # input value used for performance metric

#             # T' response to input (as if disturbance were always =0)
#             t, ideal, x = control.forced_response(Gp, time, Qprime)
#             plt.plot(t, ideal, '--', label = 'response to input')  # should be output minus disturbance response

            plt.legend(fontsize='large')
            plt.xlabel('Time (seconds)')
            plt.ylabel("T' (temperature) and Q' (heat rate)")
            plt.show()

    if summary == True:  # prints parameters, three-point summary, and a "density" distribution
        print(f'Variables:\nKC =   {KC},\nTauI = {TauI},\nTauD = {TauD},\nTauC = {TauC}\n')
        print('Minimum value of Perf = ', min(Perf))
        print('Maximum value of Perf = ', max(Perf))
        print('Average value of Perf = ', np.average(Perf))
        plt.plot(Perf, '.')
        plt.plot(np.arange(len(TiPrimes)), np.full(len(TiPrimes),np.average(Perf)), label = 'Average')
```

```
        plt.xlabel('Profile Numer')
        plt.ylabel("Performance Metric")
        plt.legend()
        plt.show()

    if failwarning and ( any(abs(Tprime)>5) or any(abs(Qprime)>5) ):  # failure condition defined by
project statement
        print(f'The case ({KC}, {TauI}, {TauD}, {TauC}) will score you an automatic ZERO')

    return np.average(Perf)



A = masterplan(5688.814947611792, 0.0002280222206125743, 0.4941522520819933,
0.006245557968152839)



print(f'Average error is {A}')
```

random search.ipynb

```
# Disturbance profiles were uploaded to Github so markers can access them through this notebook.
Let's get them.
!rm -rf CHBE356-data
!git clone https://github.com/TimHillmer/CHBE356-data/


# obtain a numerically ordered list of the paths to the disturbance profiles
import glob
filenames = glob.glob('CHBE356-data/Disturbance_profiles/*.csv')
filenames.sort()

# obtain an array containing all disturbance profiles (data in [profile#, timevalue] format)
import numpy as np
import pandas as pd
df = pd.concat([pd.read_csv(files).transpose() for files in filenames], ignore_index = True)
# each disturbance profile is a *row*

# define Yankai's 100 point interpolation
from scipy.interpolate import interp1d
Ti_data = df.values
time_data = np.arange(len(Ti_data[0]))  # assume Ti' data is given at each second
f = interp1d(time_data, Ti_data)
points_per_second = 100
times = np.linspace(time_data[0], time_data[-1], len(time_data)*points_per_second)
TiPrimes = f(times)

# Any remaining install/imports here
! pip install
git+https://github.com/python-control/python-control@601b58152080d89575cc677474ec7714e1a34ee
2
import control
import matplotlib.pyplot as plt
import random


def fasterplan(KC, TauI, TauD, TauC, s, times, Gd, Gp):  # streamlined version to save precious
milliseconds each run
    Gc = KC*(1 + 1/(TauI*s) + TauD*s/(TauC*s+1))  # Controller transfer function
    Gtotal = Gd / (1 + Gp*Gc)  # no term for (temperature deviation) setpoint because it is always zero
    Perf = np.zeros(len(TiPrimes))  # array to store Perf for each disturbance profile
    for i in range(len(TiPrimes)):  # main loop for calculation and plotting
        # T' response including controller
        t, Tprime, x = control.forced_response(Gtotal, times, TiPrimes[i])
```

```
        # controller response to error (=input)
        t, Qprime, x = control.forced_response(Gc, times, -Tprime)

        Perf[i] = (sum(np.absolute(Tprime)) + 0.2*sum(np.absolute(Qprime)))/points_per_second
        # performance metric for each profile
    return np.average(Perf)


def lograndom(lo, hi):  # samples from log(X), where X in uniformly distributed within the provided
bounds
    return(lo*10**(random.uniform(0, np.log10(hi/lo))))


# initial information for use in "fasterplan"
s = control.tf([1,0],[0,1])  # Define the s variable
Gd = (s+1) / (s**2+s+1)  # T' / Ti' (disturbance) transfer function
Gp = 1 / (s**2+s+1)  # T' / Q' (process) transfer function



Ntests = 500
Nhappy = int(Ntests/10)

counter = 0
happycounter = 0

data = np.zeros((Ntests, 5))
happydata = np.zeros((Nhappy, 5))
from google.colab import files

for i in range(Ntests):
    Kc = lograndom(1e3, 1e4)
    Ti = lograndom(1e-4, 1e-3)
    Td = lograndom(0.05,1.5)
    Tc = lograndom(5e-4, 0.1)

    Perf = fasterplan(Kc, Ti, Td, Tc, s, times, Gd, Gp)
    if Perf <= 6.95:
        happydata[happycounter] = np.array([Perf, Kc, Ti, Td, Tc])
        happycounter += 1
        print('happy', Perf, Kc, Ti, Td, Tc)

    data[counter] = np.array([Perf, Kc, Ti, Td, Tc])
    counter += 1

    if happycounter == Nhappy:
```

```
    break
```

```
from google.colab import files  # this notebook was run with Colaboratory, which downloads files using this library
np.savetxt("Data.csv", data, delimiter=",")
np.savetxt("HappyData.csv", happydata, delimiter=",")
files.download('HappyData.csv')
```

```
fasterplan(5688.814947611792, 0.0002280222206125743, 0.4941522520819933, 0.006245557968152839, s, times, Gd, Gp)
# happy 6.789263931270320 8157.95407778562 0.00014069840043239406 0.13349633544031736 0.0010902891735579206
# happy 6.769239197968654 6780.9431099381745 0.0008394599924216926 0.6232736552360005 0.01064341130569476
# happy 6.661657297636582 9535.386683958759 0.00010033885457162226 0.819218758695484 0.004582526747423241
# happy 6.604902873671916 8763.00366411034 0.00021935638604842907 0.17331925291467143 0.0034497938673252715
# happy 6.603490145150944 5688.814947611792 0.0002280222206125743 0.4941522520819933 0.006245557968152839
```