Actual step function response below:

Section 1: system equation (not necessary for the simulation)

$$\frac{2\zeta}{\omega_n}\dot{z}_i(t) + z_i(t) = \frac{\varepsilon A}{2k\left(d_i - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij}V_jU\left(z_j(t) - d_j\right) + \theta_i\right)^2 - \sum_{k=1}^{3} w_{in,k}\frac{1}{\omega_n^2}\ddot{y}_k(t) \qquad (1)$$

$$\tau = \frac{2\zeta}{\omega_n} = 0.0017$$

$$\frac{\varepsilon A}{2k} = 8.1539e - 19$$

$$\frac{1}{\omega_n^2} = 6.8356e - 07$$

$$0.0017\dot{z}_i(t) = -z_i(t) + \frac{8.1539\times 10^{-19}}{\left(42\times 10^{-6} - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij}V_jU\left(z_j(t) - 42\times 10^{-6}\right) + \theta_i\right)^2 - \ddot{y}(t) - \sum_{k=1}^{3} w_{in,k}\times 6.8356\times 10^{-7}\times \ddot{y}_k(t) \qquad (2)$$

Section 2: Discretized system equation (to be used in the simulation)

$$z_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right)z_i(t) + \frac{\Delta t}{\tau}\frac{\varepsilon A}{2k\left(d_i - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij}V_jU\left(z_j(t) - d_j\right) + \theta_i\right)^2 - \frac{\Delta t}{\tau}\sum_{k=1}^{3} w_{in,k}\frac{1}{\omega_n^2}\ddot{y}_k(t) \qquad (3)$$

$$\Delta t = 0.0001$$

$$z_i(t+0.0001) = (0.9395)z_i(t) + \frac{4.9311\times 10^{-20}}{\left(42\times 10^{-6} - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij}V_jU\left(z_j(t) - 42\times 10^{-6}\right) + \theta_i\right)^2 - 4.1339\times 10^{-8}\ddot{y}(t) - \sum_{k=1}^{3} w_{in,k}\times 4.1339\times 10^{-8}\times \ddot{y}_k(t) \qquad (4)$$

In here, z_i is the state of the i^th^ "Neuron", t is the time, w_{ij} is the connection weight from the j^th^ neuron to the i^th^ neuron, V_j is the total voltage signal on the j^th^ neuron (defined below), $U(x-a)$ is a unit step function (defined below), θ_i is a biasing signal applied to the i^th^ neuron, $\ddot{y}(t)$ is an acceleration signal (viewed by the system as an input signal) and $w_{in,k}$ is an input weight.

This is the equation to be used. Note that $z_i(t) \le 42\times 10^{-6}$. If, at any step we compute $z_i(t+\Delta t) \ge 42\times 10^{-6}$, please force it to go down to 42×10^{-6}. Otherwise, you will get very high and unrealistic numbers.

Additional information:

- $U\left(z_j(t) - 42\times 10^{-6}\right)$ is a unit step function with $U\left(z_j(t) - 42\times 10^{-6}\right) = 1$ if $z_j(t) = 42\times 10^{-6}$ and zero otherwise.

- $V_i = \left(\sum_{j=1}^{N} w_{ij} V_j U\left(z_j(t) - d_j\right) + \theta_i\right)$ for every neuron and must be re-evaluated at every time step.
- $\ddot{y}_k(t)$ is the acceleration signal (in m/s^2) measured from the accelerometers in the UCI dataset in the k^{th} direction (k=1: x-direction, k=2: y-direction, k=3: z-direction). For here, we should assume that each MEMS device can have a maximum of ONE non-zero $w_{in,k}$ (when non-zero, $w_{in,k} \in \{-1,1\}$ only)
- Here, we want to train the system to optimize for the values of w_{ij} and θ_i

Approximation below:

Section 1: system equation (not necessary for the simulation)

$$\frac{2\zeta}{\omega_n}\dot{z}_i(t) + z_i(t) = \frac{\varepsilon A}{2k\left(d_i - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij} V_j \sigma_M\left(z_j(t) - d_j\right) + \theta_i\right)^2 - \sum_{k=1}^{3} w_{in,k} \frac{1}{\omega_n^2} \ddot{y}_k(t) \qquad (1)$$

$$\tau = \frac{2\zeta}{\omega_n} = 0.0017$$

$$\frac{\varepsilon A}{2k} = 8.1539e - 19$$

$$\frac{1}{\omega_n^2} = 6.8356e - 07$$

$$0.0017\dot{z}_i(t) = -z_i(t) + \frac{8.1539\times10^{-19}}{\left(42\times10^{-6} - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij} V_j \sigma_M\left(z_j(t) - 42\times10^{-6}\right) + \theta_i\right)^2 - \ddot{y}(t) - \sum_{k=1}^{3} w_{in,k} \times 6.8356\times10^{-7} \times \ddot{y}_k(t) \qquad (2)$$

Section 2: Discretized system equation (to be used in the simulation)

$$z_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) z_i(t) + \frac{\Delta t}{\tau}\frac{\varepsilon A}{2k\left(d_i - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij} V_j \sigma_M\left(z_j(t) - d_j\right) + \theta_i\right)^2 - \frac{\Delta t}{\tau}\sum_{k=1}^{3} w_{in,k} \frac{1}{\omega_n^2} \ddot{y}_k(t) \qquad (3)$$

$$\Delta t = 0.0001$$

$$z_i(t+0.0001) = (0.9395) z_i(t) + \frac{4.9311\times10^{-20}}{\left(42\times10^{-6} - z_i(t)\right)^2}\left(\sum_{j=1}^{N} w_{ij} V_j \sigma_M\left(z_j(t) - 41.8\times10^{-6}\right) + \theta_i\right)^2 - 4.1339\times10^{-8}\ddot{y}(t) - \sum_{k=1}^{3} w_{in,k} \times 4.1339\times10^{-8} \times \ddot{y}_k(t) \qquad (4)$$

In here, z_i is the state of the i^{th} "Neuron", t is the time, w_{ij} is the connection weight from the j^{th} neuron to the i^{th} neuron, V_j is the total voltage signal on the j^{th} neuron (defined below), $\sigma_M(x)$ is a logistic function that approximates the unit-step function properties of the MEMS switching (defined below), θ_i is a biasing signal applied to the i^{th} neuron, $\ddot{y}(t)$ is an acceleration signal (viewed by the system as an input signal) and $w_{in,k}$ is an input weight.

This is the equation to be used. Note that $z_i(t) \leq 42 \times 10^{-6}$. If, at any step we compute $z_i(t+\Delta t) \geq 42 \times 10^{-6}$, please force it to go down to 42×10^{-6}. Otherwise, you will get very high and unrealistic numbers.

Additional information:

- $\sigma_M(x)$ is logistic function with $\sigma_M(x) = 1/(1 + \exp[-\alpha\ (x)])$. Where $\alpha = 50 \times 10^6$ Hence, for the given function $\sigma_M\left(z_j(t) - 41.8 \times 10^{-6}\right), substitute\ 1/(1 + \exp\left[-50 \times 10^6\ \left(z_j(t) - 41.8 \times 10^{-6}\right)\right])$.
- $V_i = \left(\sum_{j=1}^{N} w_{ij} V_j \sigma_M\left(z_j(t) - d_j\right) + \theta_i\right)$ for every neuron and must be re-evaluated at every time step.
- $\ddot{y}_k(t)$ is the acceleration signal (in m/s^2) measured from the accelerometers in the UCI dataset in the k^{th} direction (k=1: x-direction, k=2: y-direction, k=3: z-direction). For here, we should assume that each MEMS device can have a maximum of ONE non-zero $w_{in,k}$ (when non-zero, $w_{in,k} \in \{-1,1\}$ only)
- Here, we want to train the system to optimize for the values of w_{ij} and θ_i

Approximation comparison with the actual response (just as a check):

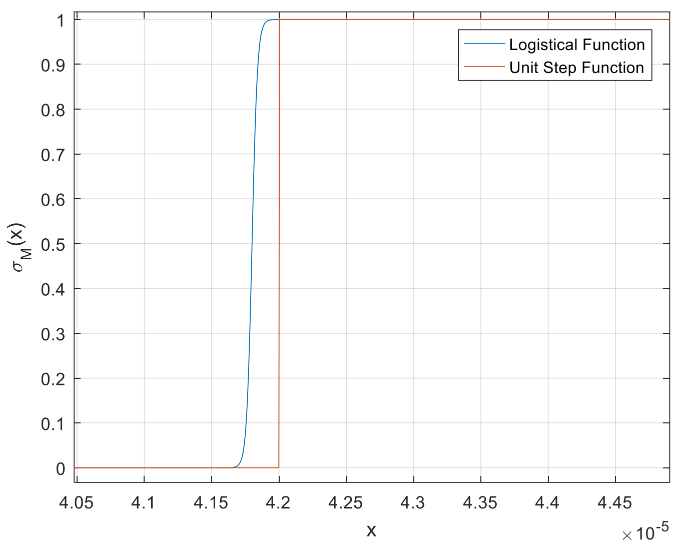
Working on pull-in coupling between MEMS devices translates to MEMS devices acting as switches. In this case, the electrical signals between MEMS devices is extremely discontinuous (like a step signal). This can cause problems in the training procedure. Instead, here, we try to model the switching action as a logistic function with a very high slope.

For example, for a MEMS device that switches at a deflection of 42×10^{-6}, the following sigmoidal function can be used:

$$\sigma_M(x) = 1/(1 + \exp[-\alpha\ (x - \beta)])$$

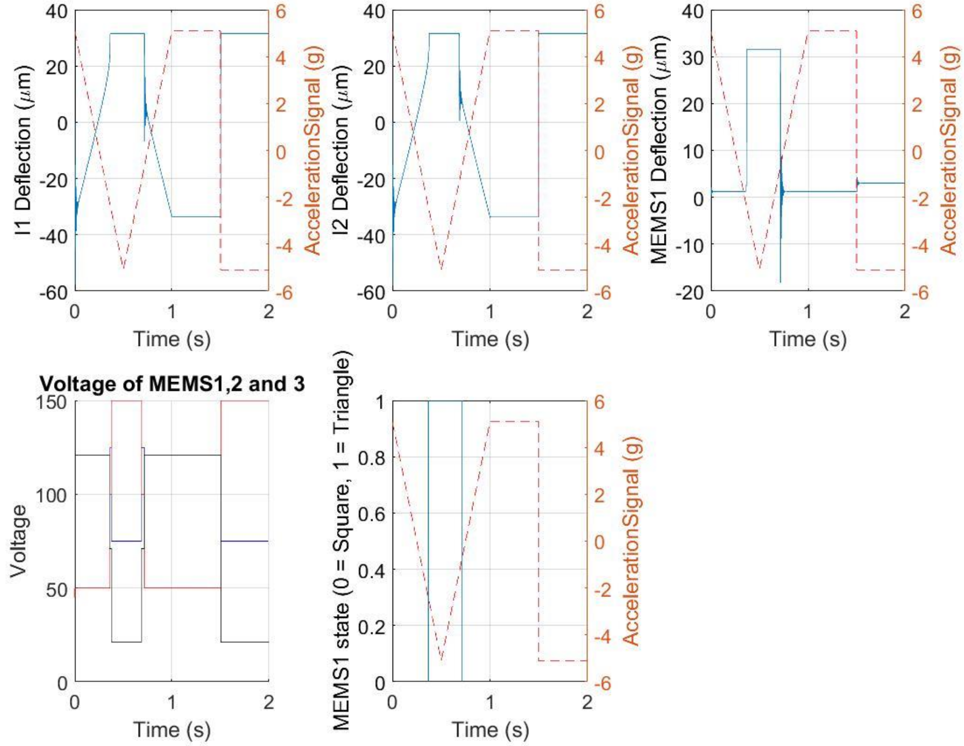
Here, $\alpha = 50 \times 10^6$ and $\beta = 41.8 \times 10^{-6}$

For comparison, the response plot is shown below:



Comparison between actual outputs and approximations:

1. Actual Outputs:



2. Approximation