You have been sitting at rest for at least ten time-steps at position 10. Starting at time t=1, you start moving to get to position 50.

(a) Assuming your motion strategy is:
\[ \text{Motion}_t = \frac{2}{5} \times (\text{Target} - \text{Actual}_t) \]
What is your location at time t=4?

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>10</td>
<td>26</td>
<td>35.6</td>
<td>41.36</td>
</tr>
<tr>
<td>Motion</td>
<td>16</td>
<td>9.6</td>
<td>5.76</td>
<td></td>
</tr>
</tbody>
</table>

41.4

(b) Assuming your motion strategy is:
\[ \text{Motion}_t = \frac{2}{5} \times (\text{Target} - \text{Sensed}_t) \]
and your Sensed location is your actual location 2 steps in the past
\[ \text{Sensed}_t = \text{Actual}_{t-2} \]
What is your location at time t=4?

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>10</td>
<td>26</td>
<td>32</td>
<td>42</td>
</tr>
<tr>
<td>Delay</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Motion</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

48 58

(c) Given the motion strategy:
\[ \text{Motion}_t = \frac{2}{5} \times (\text{Target} - \text{Sensed}_t) \]
provide a sensory delay that will cause you never to converge at your target location. Explain your answer in 1-3 sentences.

A delay of 4 will prevent convergence. The delay needs to be large enough that you will overshoot your target by increasingly larger amounts at each bend of the wave around the target. (By time t=8, you have overshot by >45, which is farther distance than the original location’s distance from the target.)
Any delay greater than 4 will work too.
(d) Given a 1 time step sensory delay \((\text{Sense}_t = \text{Actual}_{t-1})\), and the motion strategy
\[
\text{Motion}_t = ?? \times (\text{Target} - \text{Sensed}_t)
\]
Provide a multiplication value ?? that will cause you never to converge at your target location.
Explain your answer in 1-3 sentences.

A multiplication of any value greater than 1 will prevent convergence. E.g., multiply by 1.2.
Need multiplier large enough to overshoot your target by increasingly larger amounts at each bend of the wave around the target.

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
<th>t=5</th>
<th>t=6</th>
<th>t=7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>10</td>
<td>58</td>
<td>106</td>
<td>96.4</td>
<td>29.2</td>
<td>-26.48</td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>10</td>
<td>10</td>
<td>58</td>
<td>106</td>
<td>96.4</td>
<td>29.2</td>
<td></td>
</tr>
<tr>
<td>Motion</td>
<td>48</td>
<td>48</td>
<td>-9.6</td>
<td>-67.2</td>
<td>-55.68</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. You have been sitting at rest for at least ten time-steps at position 40. Starting at time \(t=1\), you start moving to get to position 100.

Assume your motion strategy is:
\[
\text{Motion}_t = \frac{2}{3} \times (\text{Target} - (\text{Forward}_t - (\text{Sensed}_t - \text{Delay}_t)))
\]
with sensory delay of 1 time point.

(a) Provide the entries in the following table for time steps \(t=2, 3, \text{ and } 4\)

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
<td>97.8</td>
</tr>
<tr>
<td>Forward</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
<td>97.8</td>
</tr>
<tr>
<td>Delay</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
</tr>
<tr>
<td>Sens</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
</tr>
<tr>
<td>Motion</td>
<td>40</td>
<td>13.3</td>
<td>4.44</td>
<td>99.3</td>
</tr>
</tbody>
</table>

(b) While you are taking your step forward at \(t=2\) (by the distance specified in \(\text{Motion}_{t=2}\)) a strong gust of wind blows you backward by 10.

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>40</td>
<td>80</td>
<td>103.3</td>
<td>107.8</td>
</tr>
<tr>
<td>Forward</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
<td>97.8</td>
</tr>
<tr>
<td>Delay</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>103.3</td>
</tr>
<tr>
<td>Sens</td>
<td>40</td>
<td>40</td>
<td>80</td>
<td>93.3</td>
</tr>
<tr>
<td>Motion</td>
<td>40</td>
<td>13.3</td>
<td>4.44</td>
<td>99.3</td>
</tr>
</tbody>
</table>
i. At what time point do you first incorporate your error in forward prediction into your Motion command?

**Time t=4**

ii. What change in motion is caused by the first use of error correction (e.g., add extra 5 to motion or subtract extra 20 to motion at current time step).

**Without Delay-Sens term, motion would be:** \( \frac{2}{3} \times (\text{Target} - \text{Forward}) \)

**WITH Delay-Sens term, motion is:**

\[
\frac{2}{3} \times (\text{Target} - \text{Forward}) - \frac{2}{3} (83.3 - 93.3) = \\
\frac{2}{3} \times (\text{Target} - \text{Forward}) + \frac{2}{3} \times 10
\]

*Subtract 10 from motion*

*Add extra +6.67 motion forward to correct for past blow back from wind*

3. Review of functions in Matlab

(a) Let us say we define a function as follows:

```matlab
function actual = myMoves(fract, timeSteps)
    actual=zeros(1,timeSteps);
    for t=2:timeSteps,
        move=fract*(10-actual(t-1));
        actual(t)=actual(t-1)+move;
    end;
```

When we run:

```matlab
actual=myMoves(0.4, 3);
```

What are the values inside the variable actual?

\[[0, 4, 6.4]\]
(b) Let us say we define a function as follows:

```matlab
function c = testFunc(a, b)
c = 2*a + b
```

When we run:
```
c = testFunc(0.4, 3);
```
What are the values inside the variable `c`?

3.8

(c) Consider the following code to compute a model “leaky integrate and fire” neuron’s voltage over 1000 time points:

```matlab
V=zeros(1,1000);
% initialize neuron simulation parameters
V0=-55; E1=-68; tau=0.04; RI=10; step=0.01;
Vthresh=-20; Vreset=-75;
V(1)=V0;
% compute voltage at t=2
deltaV=(-(V(1)-E1)+RI)/tau;
V(2)=V(1)+deltaV*step;
if (V(2)>Vthresh)
    V(2)=Vreset;
end;
...% compute voltage at t=1000
deltaV=(-(V(999)-E1)+RI)/tau;
V(1000)=V(999)+deltaV*step;
if (V(1000)>Vthresh)
    V(1000)=Vreset;
end;
```

We now wish to define a function `neuronSimulate` that will produce a vector `V` containing the neuron’s voltage at each `T` time-points. In order to compute these values, `neuronSimulate` will use the parameters `V0`, `El`, `tau`, `RI`, `step`, `Vthresh`, and `Vreset`. Specifically:

```matlab
V=neuronSimulate(1000, -55, -68, 0.04, 10, 0.01, -20, -75)
```

will produce the vector `V` with the exact same contents as produced by the block of code in blue above.

Provide the code to define the function `neuronSimulate`.

(You will get partial credit just converting the blue code to a for loop even without defining a function)

```matlab
function V=neuronSimulate(totalT,V0,El,tau,RI,step,Vthresh,Vreset)
    V=zeros(1,totalT);
```
\( V(1) = V_0; \)
\[
\text{for } t = 2 : \text{totalT},
\]
\[
\quad \% \text{ compute voltage at } t
\]
\[
\quad \text{deltaV} = \frac{-(V(t-1) - E_l) + RI}{\tau};
\]
\[
\quad V(t) = V(t-1) + \text{deltaV} \times \text{step};
\]
\[
\quad \text{if } (V(t) > V_{\text{thresh}})
\]
\[
\quad \quad V(t) = V_{\text{reset}};
\]
\[
\quad \text{end};
\]
\[
\text{end;}
\]

4. Reflecting the binding hypothesis/binding problem, list the set of objects (e.g., “big red ball,” “pink flamingo”) present in the scene producing the following spiking patterns. Each row reflects the spiking of a neuron encoding the feature named at the beginning of the row. In this assignment, spikes are considered to be synchronous if they occur within 1 ms of one another.

(a)

Open-door blue building; old white house; open-door blue tent; white building; old blue house

(b)
Dry flower; red small grass; dangerous blue tree; red tree

(c) Let us say I redefine synchrony as spikes occurring within 10 ms of one another. Will this definition of synchrony change the interpretation of the spiking patterns for part b above?

Yes. Several objects will be perceived as one object with all the properties mixed together.

(d) Estimate the widest spike synchrony window you can define without changing the interpretation of the spiking patterns above. Provide a separate answer for parts a and b each.

a and b both around 4 ms

5. Let us consider the dynamics for the following neural network.

Feedforward weights:
\[ w_{in,A} = 0.8 \quad w_{in,B} = 0.8 \quad w_{in,C} = 0.8 \]
Lateral weights:
Outputs from neuron A: \( w_{A,B} = 1.5 \)
Outputs from neuron B: \( w_{B,A} = 0.5 \) \( w_{B,C} = 0.5 \)
Outputs from neuron C: \( w_{C,B} = 1.5 \)
Memory feedback weights:
\( w_{\text{mem},A} = 1.5 \) \( w_{\text{mem},B} = 1 \) \( w_{\text{mem},C} = 0.5 \)

The table below provides the following activity from feedforward (in) and feedback (mem).
(a) Provide the A, B, and C neuron outputs in time \( t=2 \) to \( t=6 \)

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
<th>t=5</th>
<th>t=6</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mem</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>2.3</td>
<td>3.2</td>
<td>5.05</td>
<td>6.05</td>
<td>7.325</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1.8</td>
<td>7.1</td>
<td>9.1</td>
<td>14.65</td>
<td>16.65</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1.3</td>
<td>2.2</td>
<td>4.05</td>
<td>5.05</td>
<td>7.325</td>
</tr>
</tbody>
</table>

(b) Provide changes to weights to cause growth activity (never-terminating firing). (It is possible the current weights work.)
Keeping the weights the same will work. Or you can make any or all the weights larger.

(c) Provide changes to weights to cause decay activity (failure to retain firing with feedback memory input). (It is possible the current weights work.)
An easy way to achieve decay activity is to make all the weights 0. Otherwise, you can make the weights sufficiently close to 0. Specifically, set memory weights to 0 and set lateral weights to be below 0.5

(d) We can represent \([\text{in, mem, A,B,C}]\) neuron inputs and responses at time \( t \) as vector \( V_1 \) and represent \([A,B,C]\) neuron responses at time \( (t+1) \) as vector \( V_2 \) (dropping the in and mem at time \( t+1 \)). How can we compute \( V_2 \) from \( V_1 \) using matrix multiplication? (What matrix/matrices do you need to define and multiply so \( V_2 = \ldots \) )

\[
V_2 = \begin{bmatrix}
  w_{\text{in},A} & w_{\text{mem},A} & 0 & w_{B,A} & 0 \\
  w_{\text{in},B} & w_{\text{mem},B} & w_{A,B} & 0 & w_{C,B} \\
  w_{\text{in},C} & w_{\text{mem},C} & 0 & w_{B,C} & 0
\end{bmatrix}
\begin{bmatrix}
  \text{in} \\
  \text{mem} \\
  A \\
  B \\
  C
\end{bmatrix}
\]