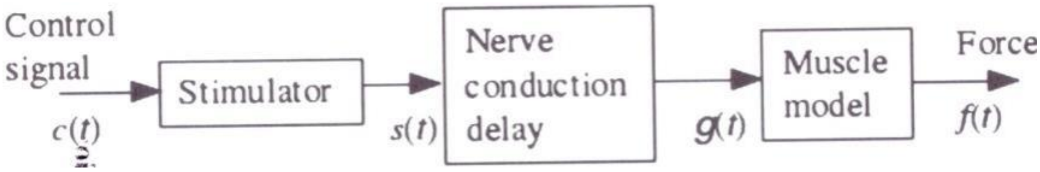
AM5510 BIOMEDICAL SIGNALS & SYSTEMS
Programming Assignment #3: Simulating Skeletal Muscle Model

DINESH KUMAR M
AM23M022

STIMULATION OF SKELETAL MUSCLE MODEL

A muscle can be activated by artificially stimulating the nerve supplying the muscle. A block diagram representation of the process to activate skeletal muscle by artificially stimulating the nerve is given below.



The control signal is a continuous signal that controls the frequency of the stimulator output. The stimulator outputs a train of pulses with frequency proportional to the voltage input to it. But the minimum and maximum frequency of the stimulator are limited.
The minimum frequency of $s(t)$ is 5 Hz and the maximum frequency is 50 Hz. Between 5 and 50 Hz the frequency of $s(t)$ is related to $c(t)$ by a constant K_3. The interpulse interval for the p th stimulus
is
$\tau_p - \tau_{p-1} = K_3/c(t)$
where $K_3 = 0.3$ second-volt.
$s(t) = \sum_p \delta(t - \tau_p)$
The nerve conduction delay introduces a constant delay of θ second:
$g(t) = s(t - \theta)$
where $\theta = 0.005$ s.

The muscle model is a model of the muscle twitch, with impulses as the input to the muscle.
The output force for a single action potential is
$f_h(t) = G(e^{-at} - e^{-bt})u(t)$
The unit of force is Newton. $a = 5/\text{s}$, $b = 20/\text{s}$, and $G = 30$ N/Hz. The cumulative force is
$f(t) = f_h(t) * g(t)$

CREATING TWO FUNCTIONS

1) $s(t) = \sum_{p} \delta(t - \tau_p)$

CODE

```
function delta_signal = discrete_delta(n, n0)
%n is the input time and n0 is the values of np
  delta_signal = zeros(size(n));
  for i = 1:length(n)
   for j= 1:length(n0)
    if n(i)==n0(j)
     delta_signal(i) = 1;
   end
  end
 end
end
```

2) $f(t) = f_h(t) * g(t)$ in dicrete time

$f[n] = \sum_{k=0}^{n} f_h[k] \cdot g[n-k]$.

CODE

```
function results = discreteconvolution(n_values, f_h, g)
   num_n_values = length(n_values);
   results = zeros(1, num_n_values);

   for i = 1:num_n_values
    n = n_values(i);
    result = 0;
    for k = 0:n
      result = result + f_h(k+1) * g(n-k+1);
    end
    results(i) = result;
  end
end
```

Simulating the output of the muscle block to a single impulse (twitch)
Code:

```
clear all;
close all;
clc;


% Define parameters
theta = 0.005; % Nerve conduction delay in seconds
G = 30; % N/Hz
a = 5; % 1/s
b = 20; % 1/s
T = 0.0001; % Sampling time in seconds% Define time vector and sampling points
t = 0:T:1; % Time vector for 0.1 seconds
n=0:1:1/T;


% Initialize arrays to store the output force and impulse response
force_single = zeros(size(t));
impulse_response = zeros(size(n));
np = [0/0.0001];


% Simulate the system for two impulses 200 ms apart

% Calculate impulse response s[n]
impulse_response = discrete_delta(n,np);


% Calculate nerve conduction delay g[n]
m=theta/T;
g=zeros(size(n));
g(m+1:end) = impulse_response(1:end-m);
fh=zeros(size(n));
for i = 1:length(t)
    fh(i) = G * (exp(-a*t(i)) - exp(-b*t(i))) * (t(i) >= 0);
end


% Calculate cumulative force f[n]
force_single = discreteconvolution(n,fh,g);


%  Plot the results
figure();
subplot(2,1,1);
plot(t, impulse_response)
title('Impulse response s[n]');
```

```
xlabel('Time (s)');
ylabel('Amplitude');
subplot(2,1,2);
plot(t, g)
title('Delayed Impulse response g[n]');
xlabel('Time (s)');
ylabel('Amplitude');
figure();
plot(t, force_single);
title('Output Force for single twitch');
xlabel('Time (s)');
ylabel('Force (N)');
```

SIMULATION OF THE OUTPUT OF THE MUSCLE BLOCK (TWO IMPULSES 200ms APART)

```
clear all;
close all;
clc;


% Defining parameters
theta = 0.005; % Nerve conduction delay in seconds
G = 30; % N/Hz
a = 5; % 1/s
b = 20; % 1/s
T = 0.0001; % Sampling time in seconds


% Define time vector and sampling points
t = 0:T:1; % Time vector for 0.1 seconds
n=0:1:1/T;


% Initialize arrays to store the output force and impulse response
force_double = zeros(size(t));
impulse_response = zeros(size(n));
np = [0/0.0001 0.200/0.0001]; % Two impulses at 200 ms apart




% Simulate the system for two impulses 200 ms apart
% Calculate impulse response s[n]
impulse_response = discrete_delta(n,np);
```

```
% Calculate nerve conduction delay g[n]
m=theta/T;
g=zeros(size(n));
g(m+1:end) = impulse_response(1:end-m);
fh=zeros(size(t));
for i = 1:length(t)

   % Calculate muscle model fh[n]
   fh(i) = G * (exp(-a*t(i)) - exp(-b*t(i))) * (t(i) >= 0);
end

%  Calculate cumulative force f[n]
force_double = discreteconvolution(n,fh,g);


 % Plot the results
figure();
subplot(2,1,1);
plot(t, impulse_response)
title('Impulse response s[n]');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(2,1,2);
plot(t, g)
title('Delayed Impulse response g[n]');
xlabel('Time (s)');
ylabel('Amplitude');
figure();
plot(t, force_double);
title('Output Force for Two Impulses 200 ms Apart');
xlabel('Time (s)');
ylabel('Force (N)');
```

Simulating the output of the muscle block for c(t)

For the rectangular pulse

```
function y = rect_pulse(t,a,b)
duration = a; % Duration in seconds
amplitude = b;
% Define the rectangular pulse
y = amplitude * (abs(t) <= duration/2);
end
```

For calculating the force

```
clear all;
close all;
clc;
% Define parameters
```

```
theta = 0.005; % Nerve conduction delay in seconds
G = 30; % N/Hz
a = 5; % 1/s
b = 20; % 1/s
T = 0.0001; % Sampling time in seconds
K3=0.3;

% Define time vector and sampling points
t = 0:T:10; % Time vector for 0.1 seconds
n=0:1:10/T;

% Condition 1: c(t) is a rect function of amplitude 4 and duration 10 secs
c1 = rect_pulse(t,20,4);
interpulse_intervals=zeros(size(t));
tau=zeros(size(t));

% Calculate pulse timings ?p and interpulse intervals
for i = 1:length(t)
if i==1
interpulse_intervals(i) = 0;
else
interpulse_intervals(i) = K3 / c1(i);
end
end
for i=1:length(interpulse_intervals)
tau(i)=sum((interpulse_intervals(1:i)));
end
np=round(tau/T);

% Initialize arrays to store the output force and impulse response
force = zeros(size(t));
impulse_response = zeros(size(n));

% Simulate the system for two impulses 200 ms apartimpulse_response =
discrete_delta(n,np);


% Calculate impulse response s[n]
% Calculate nerve conduction delay g[n]
m=theta/T;
g=zeros(size(n));
g(m+1:end) = impulse_response(1:end-m);
fh=zeros(size(n));
for i = 1:length(t)

  % % Calculate muscle model fh[n]
  fh(i) = G * (exp(-a*t(i)) - exp(-b*t(i))) * (t(i) >= 0);
end

%  Calculate cumulative force f[n]
```

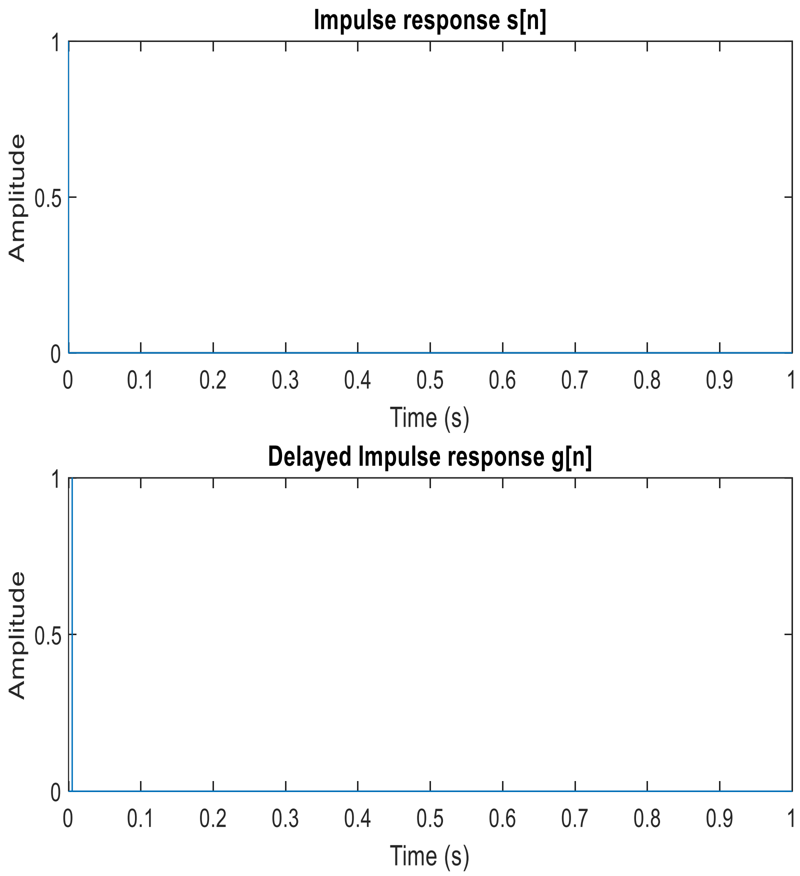
```
force = discreteconvolution(n,fh,g); % Plot the results
figure();
subplot(3,1,1);
plot(t, c1)
title('Control Function c(t)');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(3,1,2);
plot(t, impulse_response)
title('Impulse response s[n]');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(3,1,3);
plot(t, g)
title('Delayed Impulse response g[n]');
xlabel('Time (s)');
ylabel('Amplitude');
figure();
plot(t, force);
title('Output Force');
xlabel('Time (s)');
ylabel('Force (N)');
```

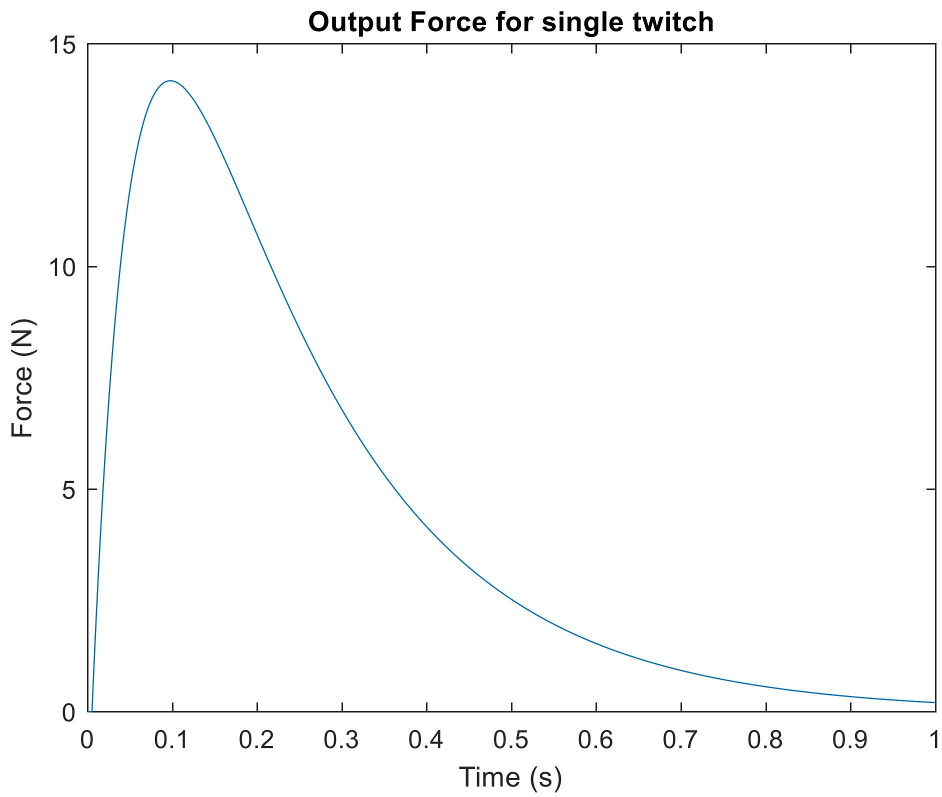
Simulating the output of the muscle block for c(t) is a rect function of amplitude 10 and duration 5 secs

```
clear all;
close all;
clc;
% Define parameters
theta = 0.005; % Nerve conduction delay in seconds
G = 30; % N/Hz
a = 5; % 1/s
b = 20; % 1/s
T = 0.0001; % Sampling time in seconds
K3=0.3;
% Define time vector and sampling points
t = 0:T:10; % Time vector for 0.1 seconds
n=0:1:10/T;
%
% Condition 1: c(t) is a rect function of amplitude 4 and duration 10 secs
c1 = rect_pulse(t,10,10); % Rectangular pulse centered at 5 seconds with amplitude 4
interpulse_intervals=zeros(size(t));
% Calculate pulse timings ?p and interpulse intervals
for i = 1:length(t)
if i==1
interpulse_intervals(i) = 0;
else
interpulse_intervals(i) = K3 / c1(i);
end
end
```

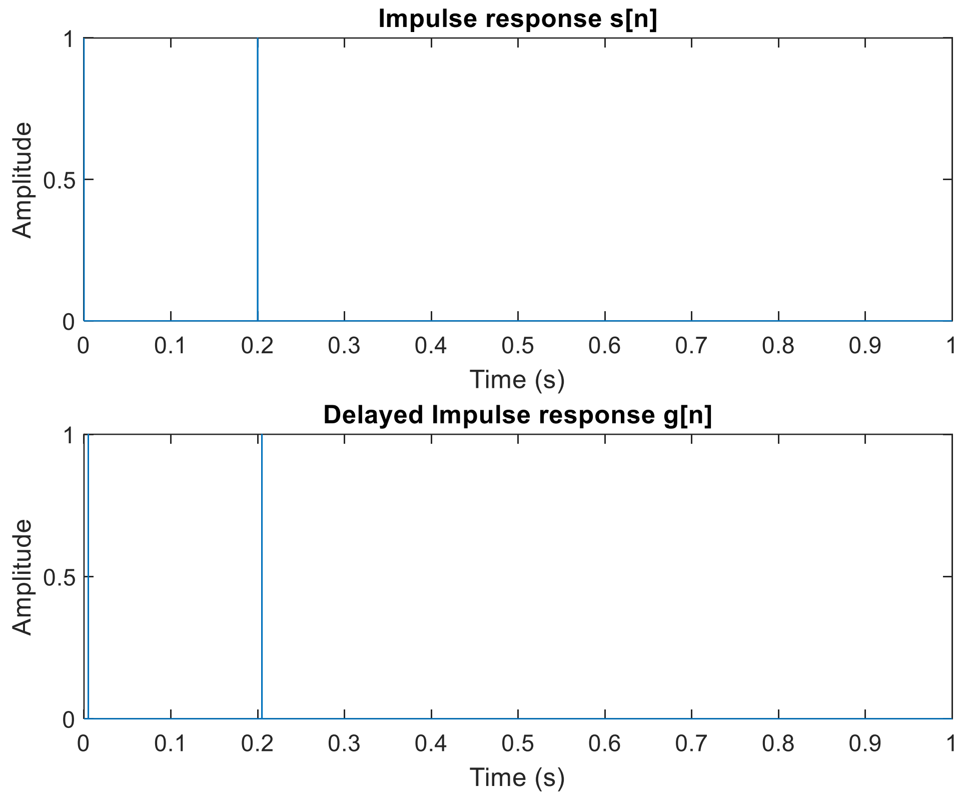
```
%remove ininity values
interpulse_intervals_new=interpulse_intervals(interpulse_intervals ~= inf);
tau=zeros(size(interpulse_intervals_new));
for i=1:length(interpulse_intervals_new)
tau(i)=sum(interpulse_intervals_new(1:i));
end
tau=tau(tau <= 5);
np=round(tau/T);
% Initialize arrays to store the output force and impulse response
force = zeros(size(t));
impulse_response = zeros(size(n));
% Simulate the system for two impulses 200 ms apart
impulse_response = discrete_delta(n,np);
% Calculate impulse response s[n]
% Calculate nerve conduction delay g[n]
m=theta/T;
g=zeros(size(n));g(m+1:end) = impulse_response(1:end-m);
fh=zeros(size(n));
for i = 1:length(t)
% % % Calculate muscle model fh[n]
fh(i)= G * (exp(-a*t(i)) - exp(-b*t(i))) * (t(i) >= 0);
% % % % Calculate cumulative force f[n]
% force_single(i) = sum(fh.* g(1:length(g)));
end
% % Calculate cumulative force f[n]
force= discreteconvolution(n,fh,g);
% % % Plot the results
figure();
subplot(3,1,1);
plot(t, c1)
title('Control Function c(t)');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(3,1,2);
plot(t, impulse_response)
title('Impulse response s[n]');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(3,1,3);
plot(t, g)
title('Delayed Impulse response g[n]');
xlabel('Time (s)');
ylabel('Amplitude');
figure();
plot(t, force);
title('Output Force');
xlabel('Time (s)');
ylabel('Force (N)');
```

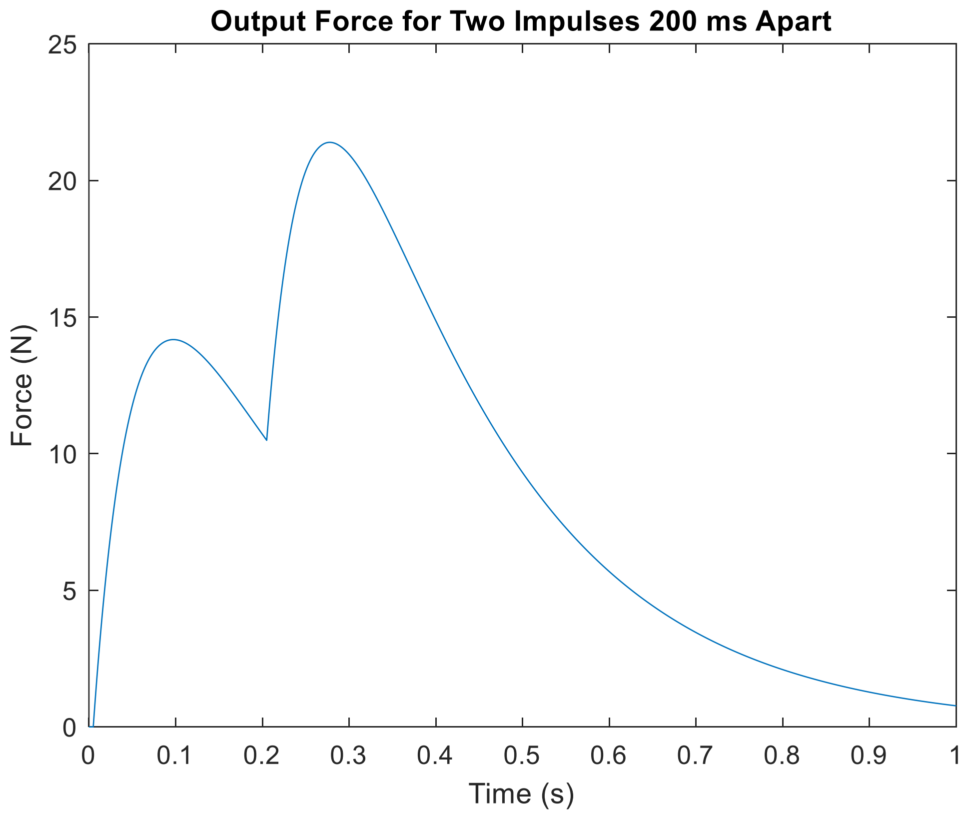
Output for single twitch:



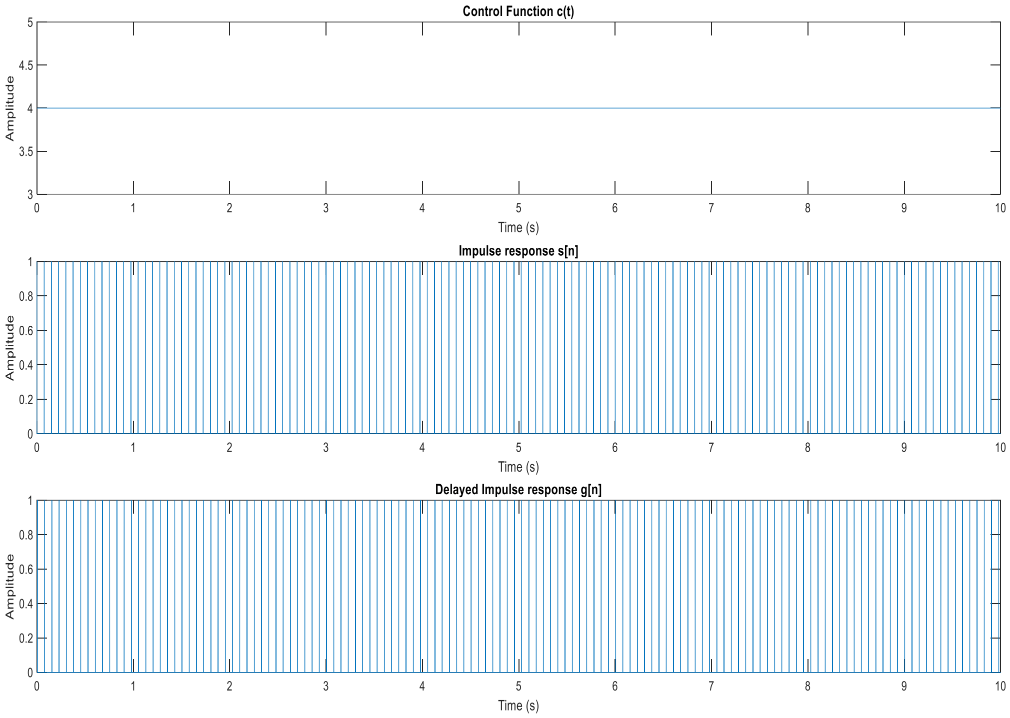


Output for two impulses at 200 ms apart:

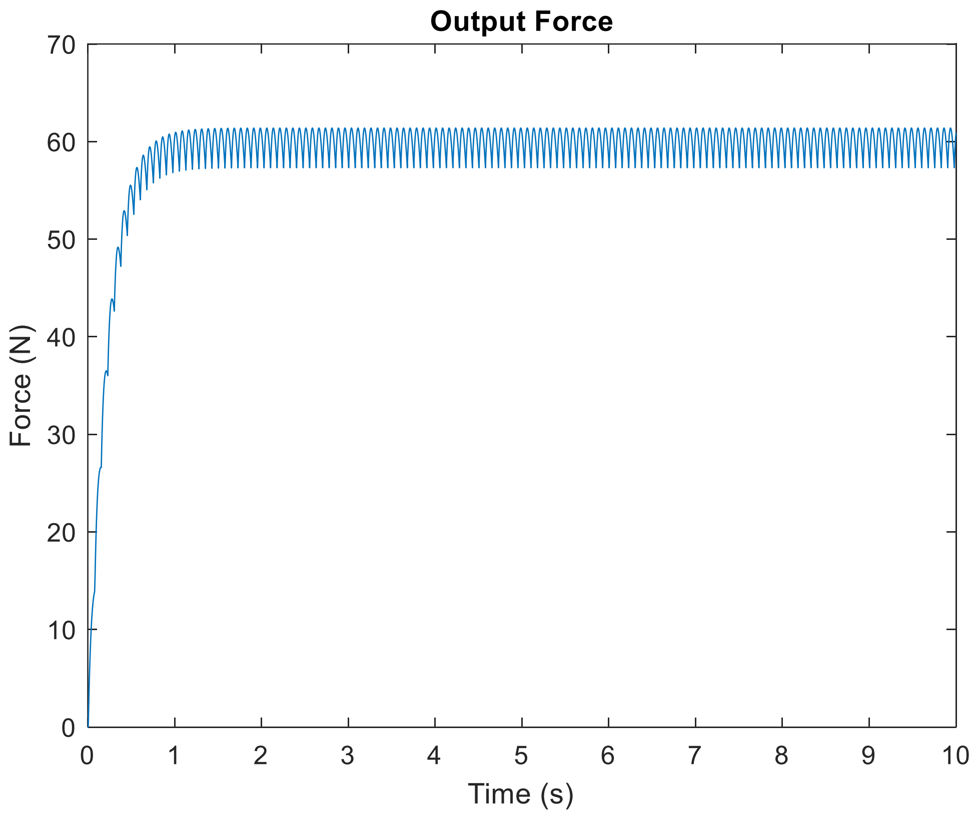




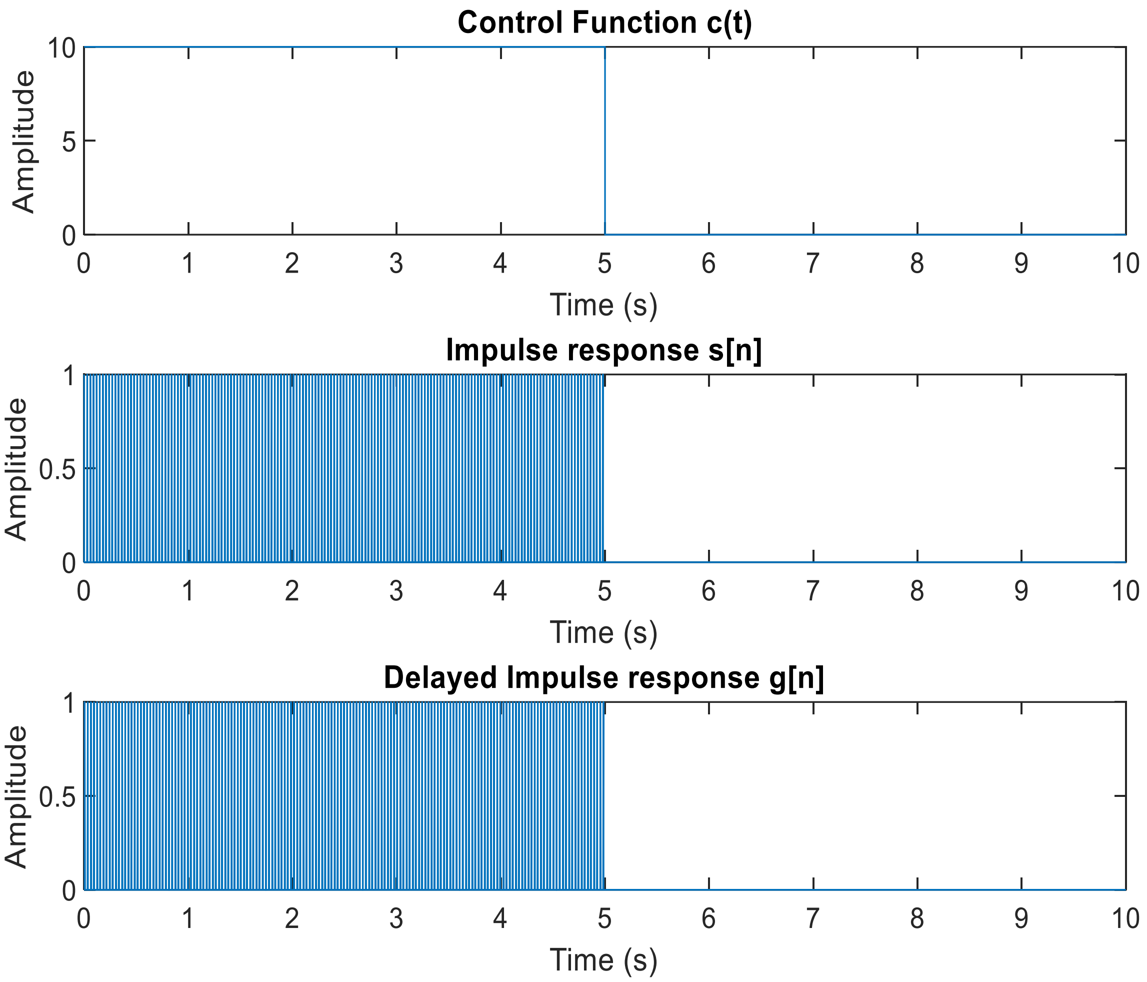
Output:

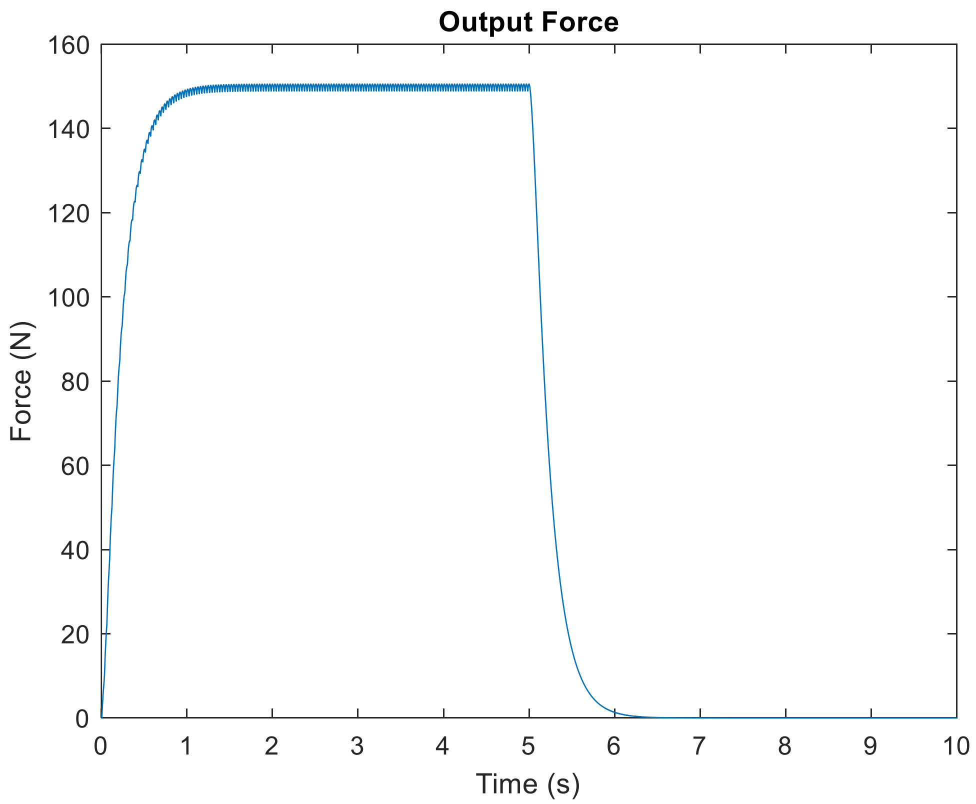


Control Function c(t)
Amplitude
Time (s)
Impulse response s[n]
Amplitude
Time (s)
Delayed Impulse response g[n]
Amplitude
Time (s)



Output Force
Force (N)
Time (s)


Control Function c(t)
Amplitude
Time (s)
Impulse response s[n]
Amplitude
Time (s)
Delayed Impulse response g[n]
Amplitude
Time (s)


Output Force
Force (N)
Time (s)

Observations

- A single stimulus caused a burst of force with a maximum at the time of stimulation
- Large stimulus interpulse intervals cause the force to fluctuate, rising and declining in
response to different stimuli.
- A muscle's ability to produce force rises as action potentials or stimuli occur more frequently. The reciprocal of the stimulation interpulse interval is the stimulus frequency. As the frequency increases the force value increases.

Write the discrete recursive equations for each block in the system so that the values at each point in time may be calculated. Use BL transform to obtain the Z-transform and then write the DT recursive equations.

AM23M022

Assignment

Given:

A block diagram representing the process of activating skeletal muscle by artifically stimulating the nerve.

Control signal $c(t)$ $\longrightarrow$ Stimulator $\xrightarrow{s(t)}$ Nerve Conduction Delay $\xrightarrow{g(t)}$ Muscle force model $\xrightarrow{f(t)}$ Force

Discrete recursive function / equation for every block in the system:-

i) Stimulator.

$$s(t) = \sum \delta(t - t_p)$$

where t_p represents the pulse timing

As the delta function is discrete, there is no point of performing bilinear transformation

$$\therefore \quad s(n) = \sum_{p} \delta(n - t_p).$$

2) Nerve conduction Delay $g(t)$

$$g(t) = s(t-\theta)$$

Given

$\theta = 0.005s$.

The discrete time function for nerve conduction delay,

$$g(n) = s[n-m]$$

where $m = \theta/T$ etc.;

T = sampling time

3) Muscle model:

$$f_h(t) = G(e^{-at} - e^{-bt})\,u(t).$$

$$f_h(t) \xrightarrow[\text{Transform}]{\text{Laplace}} f_h(s)$$

$$f_h(s) = LT\{G(e^{-at} - e^{-bt}).\,u(t)\}.$$

$$\boxed{f_h(s) = G\left[\frac{1}{s+a} - \frac{1}{s+b}\right]}$$

Performing bilinear transformation,

$$f_h(s) \xrightarrow{BLT} f_h(z)$$

$$f_n(z) = f_n(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

$$= G\left[\frac{1}{\frac{2}{T}\left(\frac{z-1}{z+1}\right)+a} - \frac{1}{\frac{2}{T}\left(\frac{z-1}{z+1}\right)+b}\right]$$

$$= G\left[\frac{T(z+1)}{2(z-1)+a(z+1)\cdot T} - \frac{T(z+1)}{2(z-1)+b(z+1)\cdot T}\right]$$

$$f_n(z) = G\cdot T(z+1)\cdot\left[\frac{1}{2(z-1)+a(z+1)\cdot T} - \frac{1}{2(z-1)+b(z+1)\cdot T}\right]$$