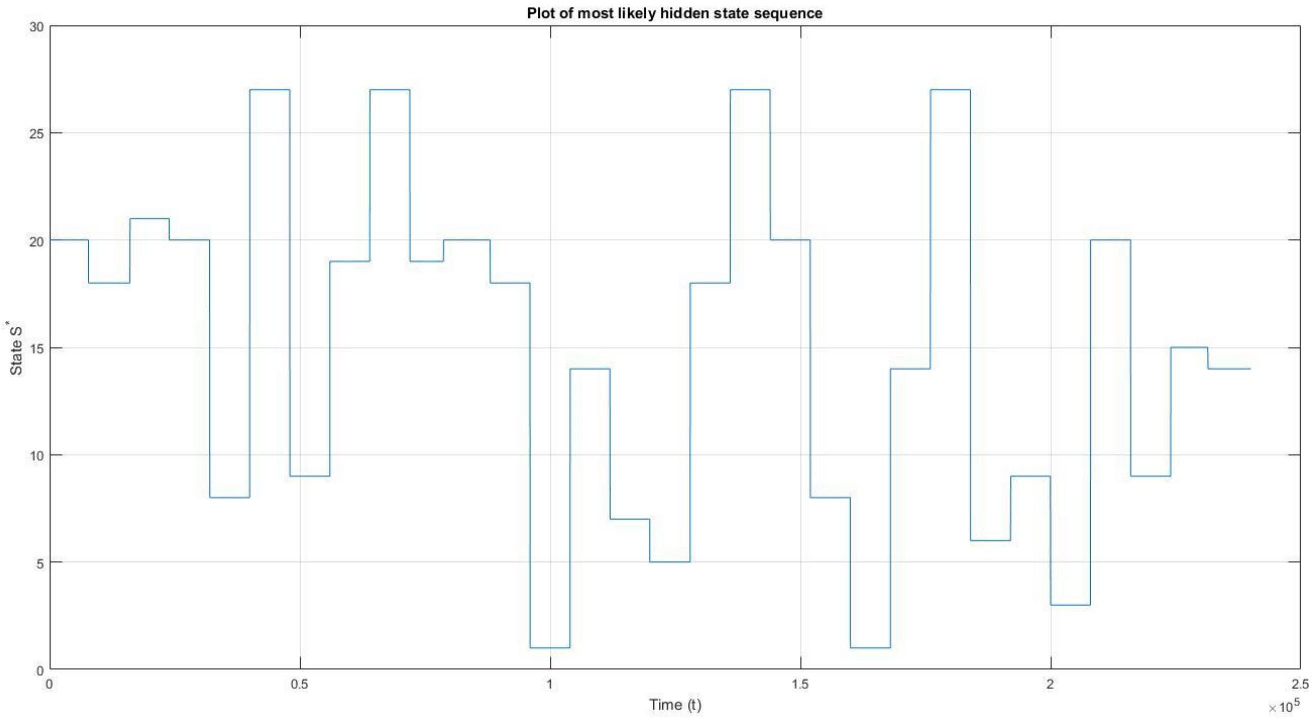
7.1) Viterbi algorithm

(a) Source code –

```
clc;
clear all;
close all;

%% Read in datasets

% Initial state probabilities
p=textread('initialStateDistribution.t
xt','%f');
nS = length(p);

% Transition matrix
tempA=textread('transitionMatrix.txt',
'%f');
for t=1:nS
    a(t,:)=tempA(nS*(t-1)+1:nS*t);
end

% Emission matrix
tempB =
textread('emissionMatrix.txt','%f');
n0 = length(tempB)/nS;
for t=1:nS
    b(t,:)=tempB(n0*(t-1)+1:n0*t);
end

% Observations
O = textread('observations.txt','%d');
T = length(O);

clear t tempA tempB n0

%% Viterbi algorithm

% Log-prob of most likely state paths
l = zeros(nS,T);

% Most-likely state transitions
phi = ones(nS,T);

% Base case
l(:,1)=log(p.*b(:,O(1)+1));

% Recursion
for t=1:T-1
    for j=1:nS

[term1,phi(j,t+1)]=max(l(:,t)+log(a(:,
j)));
        l(j,t+1)=term1 +
log(b(j,O(t+1)+1));
    end
end

clear j t term1 nS

% Back tracking
S = ones(1,T);

[~,S(T)]=max(l(:,T));
for t=T-1:-1:1
    S(t) = phi(S(t+1),t+1);
end

% Plot of most likely sequence of
hidden states
figure;
set(gcf,'color','w');
plot(1:T,S);
grid on;
xlabel('Time (t)');
ylabel('State S^{*}');
title('Plot of most likely hidden
state sequence');

%% Decode hidden message

% find the unique sequence of states
k = 2;
uniqueStates(1)=S(1);
for t=2:T
    if(S(t)~=S(t-1))
        uniqueStates(k)=S(t);
        if(S(t)==27)
            uniqueStates(k)=32-64;
        end
        k=k+1;
    end
end
clear k t

% Get letters corresponding to
alphabets
message = char(uniqueStates+64);
```

(b) Plot of most likely sequence of hidden states versus time



The decoded message is :

TRUTH IS STRANGER THAN FICTION