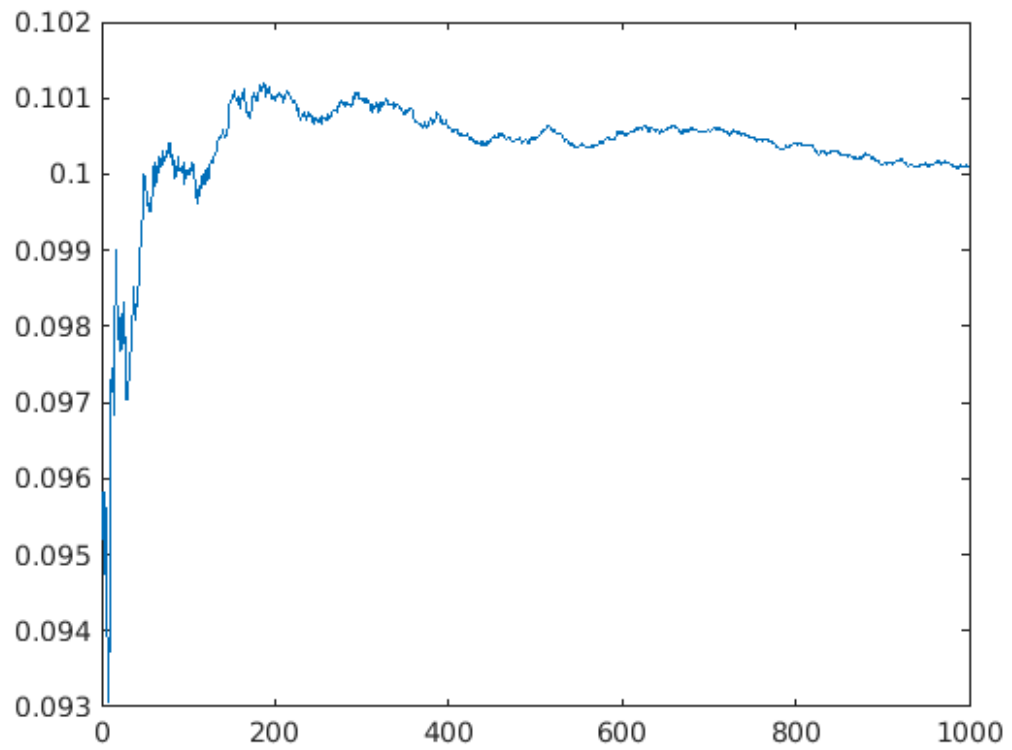

```
Q= zeros(1,10); % change 10 to n if there are n actions (eg. in case
    of binary bandit n=2)
N= zeros(1,10); % change 10 to n if there are n actions (eg. in case
    of binary bandit n=2)
Rn= zeros(1,1000);
itr =zeros(1,1000);
epsilon=0.1;
sum=0;
alpha=0.2;

for iter = 1:1000
    itr(iter)=iter;
    if rand > epsilon
        [m,id]= max(Q);
        A= id;
    else
        temp= randperm(10); % change 10 to n if there are n actions
        (eg. in case of binary bandit n=2)
        A= temp(1);
    end
    R= bandit_nonstat(A);
    N(A)= N(A)+1;
    Q(A)= Q(A)+ alpha*(R-Q(A));
    sum = sum + R;
    Rn(iter)= sum/iter;
end

plot(itr,Rn);
```

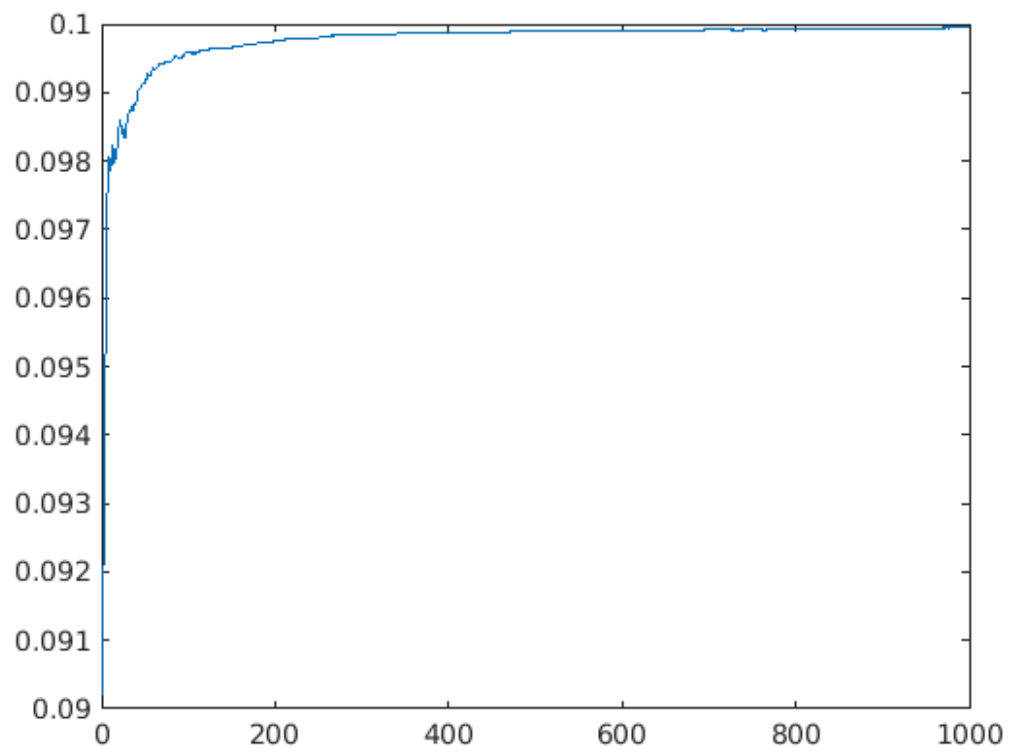


Published with MATLAB® R2021a

```
Q= zeros(1,10);
N= zeros(1,10);
Rn= zeros(1,1000);
Rn1= zeros(1,1000);
itr =zeros(1,1000);
epsilon=0.1;
sum=0;
alpha=0.2;
sum1=0; % used for taken average for those 1000 iterations.

% reward for each action is calculated for 1000 times
% then after computing reward again it is computed for 1000 to and
  taken average
% Therefore, total no. of iterations = 1000*1000= 1000000

for itrt= 1:1000
    Rn = zeros(1,1000);
    sum =0;
    for iter= 1:1000
        itr(iter)=iter;
        if rand > epsilon
            [m,id]= max(Q);
            A= id;
        else
            temp= randperm(10);
            A= temp(1);
        end
        R= bandit_nonstat(A);
        N(A)= N(A)+1;
        Q(A)= Q(A)+ (R-Q(A))*alpha;
        sum = sum + R;
        Rn(iter)= sum/iter;
    end
    sum1 = sum1 + Rn(itrt);
    Rn1(itrt)=sum1/itrt;
end
plot(itr,Rn1);
```



Published with MATLAB® R2021a