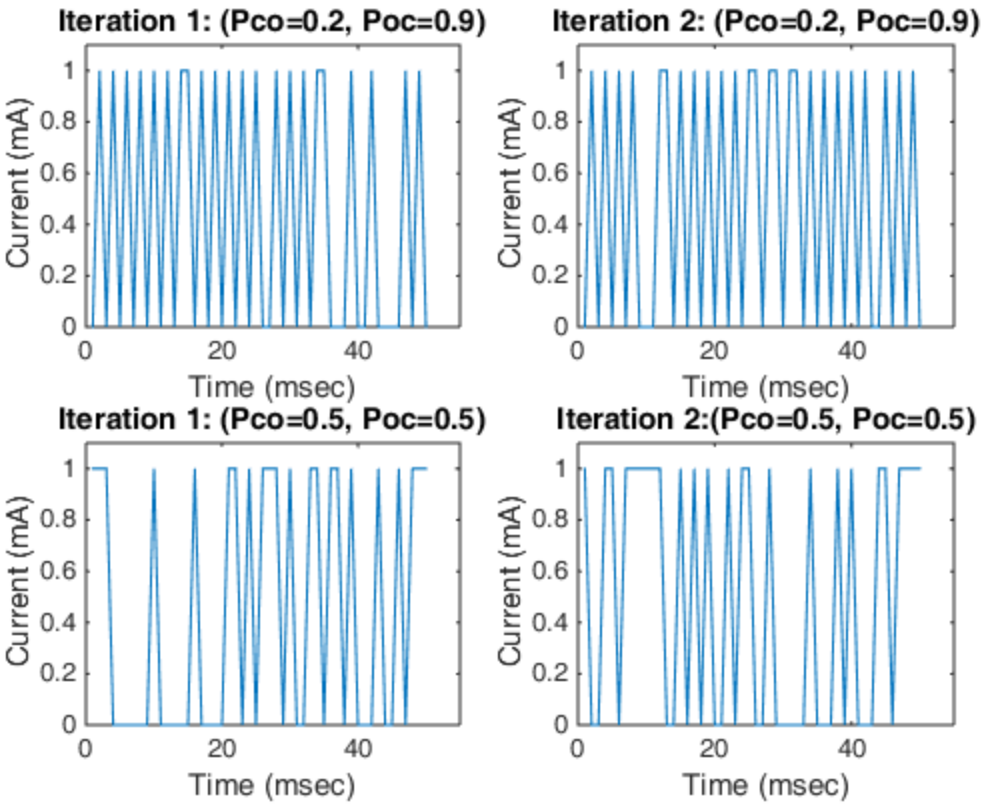
```
%MONTE CARLO SIMULATION

time_arr = zeros(50);
for j=1:50
    time_arr(j)= j;
end
%%section 1

A = zeros(50);
Poc = 0.9;
Pco = 0.2;
Poo = 0.1;
Pcc = 0.8;

for i=1:50
    no=rand(1);
    if(i==1)
        if(no<Pco)
            A(i)=0;
        else
            A(i)=1;
        end
    else
        if(A(i-1)==0)
            if(no<Pco)
                A(i)=0;
            else
                A(i)=1;
            end
        else
            if(no<Poc)
                A(i)=0;
            else
                A(i)=1;
            end
        end
    end
end


B = zeros(50);
for i=1:50
    no=rand(1);
    if(i==1)
        if(no<Pco)
            B(i)=0;
        else
            B(i)=1;
        end
    else
        if(B(i-1)==0)
            if(no<Pco)
```

```
                B(i)=0;
            else
                B(i)=1;
            end
        else
            if(no<Poc)
                B(i)=0;
            else
                B(i)=1;
            end
        end
    end
end

%%section 2

C = zeros(50);
Poc = 0.5;
Pco = 0.5;
Poo = 0.5;
Pcc = 0.5;

r=0.5;
for i=1:50
    no = rand(1);
    if(no<r)
        C(i)=0;
    else
        C(i)=1;
    end
end



D = zeros(50);

r=0.5;
for i=1:50
    no1 = rand(1);
    if(no1<r)
        D(i)=0;
    else
        D(i)=1;
    end
end

figure
subplot(2,2,1)
plot(time_arr,A), axis([0 55 0 1.1])
title('Iteration 1: (Pco=0.2, Poc=0.9)')
xlabel('Time (msec)')
ylabel('Current (mA)')

subplot(2,2,2)
```

```
plot(time_arr,B), axis([0 55 0 1.1])
title('Iteration 2: (Pco=0.2, Poc=0.9)')
xlabel('Time (msec)')
ylabel('Current (mA)')

subplot(2,2,3)
plot(time_arr,C), axis([0 55 0 1.1])
title('Iteration 1: (Pco=0.5, Poc=0.5)')
xlabel('Time (msec)')
ylabel('Current (mA)')

subplot(2,2,4)
plot(time_arr,D), axis([0 55 0 1.1])
title('Iteration 2:(Pco=0.5, Poc=0.5)')
xlabel('Time (msec)')
ylabel('Current (mA)')


%The Monte Carlo Simulation is very close to the theoretical model as given in the
%Challenge problem: For different random number arrays, we get different
%simulations.
%The plotting justifies the transitional probabilities chosen in the start.
%Depending on the Pon->off and Poff->on probablities, we can find the
%threshold. If the values of the transitional probabilities changes, the probabili
```



Published with MATLAB® R2014b