Date:2021-11-13

Finish implementation of the first two algorithms mentioned in the reference paper, and transform some other equations into some function in Matlab, all of the functions and script have been put into one live script in Matlab for convenience.

Received Power PR

```
function P=PR(lambda,PD,h_k,hD,PC)
syms N0
P=lambda*(PD*hD+PC*h_k+N0);
end
```

Throughput equation for CUE links

```
function T=Throughput_C(PD,h_iB,h_kc,P_kc)
syms N0 N1
T=log2(1+(P_kc*h_kc)/(PD*h_iB+N0+N1));
end
```

Throughput equation for D2D links.

For this function, we take the power splitting ratio lambda, transmission power fo D2D links PD, transmission of CUE, interference hki, a

```
function T=Throughput_D(lambda,PD,P_kc,h_ki,hD)
syms N0 N1
T=log2(1+((1-lambda)*(PD*hD))/((1-lambda)*(P_kc*h_ki+N0)+N1));
end
```

$$T_i^D = \log_2\left(1 + \frac{(1-\lambda_i^e)P_i^D h_i^D}{(1-\lambda_i^e)\left(P_k^C h_{k,i} + N_0\right) + N_1}\right) \qquad (5)$$

Piecewise Energy Harvesting model

we take a received power as an input argument for the Energt Harvesting model where the received power is from different segemnts,

then we compare each segment's received power and compare it to the threshold power Pth, then we can get the ouput Harvested power **E.**

```
function E=Eenergy_harvesting(PR)
syms k b Pmax Pth L
 if PR>Pth(1) &&  PR<Pth(2)
     E=0;
 % Traverse the Pth array then compare each segment's power
 elseif PR>Pth(2) && PR<Pth(L+1)
     for j=2:L
         if PR>Pth(j) && PR<Pth(j+1)
             E=k(j)*PR+b(j);
             break;
         end
     end
 else
     E=Pmax;
 end

end
```

$$\mathrm{EH}_i^D = \begin{cases} 0, & P_i^R \in \left[P_{th}^0, P_{th}^1\right] \\ k_j P_i^R + b_j, & P_i^R \in \left[P_{th}^j, \ P_{th}^{j+1}\right], \quad j \in 1, \ldots, L-1 \\ P_{\max}^{\mathrm{EH}}, & P_i^R \in \left[P_{th}^L, P_{th}^{L+1}\right] \end{cases} \qquad (7)$$

Total Energy Consumption EC_iD

For this function we want to find the total energt consumption EC, so the basic idea will be taking the transmission power PD , circuit power consumption P_iR and the harvested power EH, and the equatio in the picture below will be used to caculate the energy consumption, the EH can be

obtained from the function **EH**

$$\mathrm{EC}_i^D = P_i^D + 2P_{\mathrm{cir}} - \mathrm{EH}_i^D \qquad (8)$$

Prematching Algorithm

```
function [SiD,InfD,EhaD] = Prematch(D,C,PC,Pth1,Pmax,TminD)
% set the channel responses of D2D link and channel responses from D2D link
% and CUE link
syms hD h_interference_D2D_CUE
SiD=C;
%Non-EH D2D links
InfD=[];
%SWIPT-Eanbled communication system
EhaD=[];
for i=1:size(D,2)
    for k=1:size(C,2)
        %\lamda ie
        lambda_min=Pth1/(Pmax*hD(i)+PC(k)* h_interference_D2D_CUE(k,i)+N0);
        Tid_max=log2(1+(Pmax*hD(i))/(PC(k)*h_interference_D2D_CUE(k,i)+N0+(N1)/1-lambda_min));
        %compare the minimum power splitting ratio and maximum
        %Throughput,if both of the requiremnts can not be met, then the k
        %will be removed from the selection set.
        if lambda_min>1 || Tid_max<TminD
            SiD(SiD==k)=[];
        end
    end
    %After the CUE set was scanned, then check if the selection set is
    %empty
    if isempty(SiD)==1
        InfD(end+1)=i;
    end

    if isempty(SiD)~=1
        EhaD(end+1)=i;
    end
end


end
```

Outer loop Algorithm

```
function [Pid_optimal,lambda_optimal,EE_optimal]=Outer(EhaD,SiD,lambda,P_ijD,EE_ijD)
%first of all, we need to search the whole SWIPT-Enabled D2D communication
%system, then search for the partner selection set obtained by the
%pre-matching algorithm. Then we set Nmax segments first then search them,
%for each segment, we need to find the maxmimum Energy Efficiency of each
%segment and the j value at which Energy efficiecny reaches its maximum
%value, then update the optimal value of J, transmission power PiD_optimal,
%power plitting ratio lambda_optimal, and ofcourse the energy efficiency EE_optimal
for i =1:size(Ehad,2)
    for k=1:size(SiD,2)
        for j=1:Nmax
            [argvalue,argmax]=max(EE_ijD(i:j));
            j_optimal=argmax;
            Pid_optimal=P_ijD(i,j_optimal);
            lambda_optimal=lambda(i,j_optimal);
            EE_optimal=EE_ijD(i,j_optimal);
        end
    end
end
end
```