```
function rect_sim(R_load)
% given
L_ac = 1e-3;
L_dc = 5e-3;
w_ac = 100;
T_ac = (2*pi)/w_ac;
esp = 0.01;
tau = 1e-5;
%tau = 1e-2;

R_load_12 = 124.0490/433.0127; % borderline mode 1&2
R_load_23 = 71.6197/750; % borderline mode 2&3

if R_load > R_load_12
    mode = 1;
elseif (R_load < R_load_12) & (R_load > R_load_23)
    mode = 2;
elseif R_load == R_load_12
    mode = 12;
elseif R_load == R_load_23
    mode = 23;
end

% init
i_d1(1) = 0;
i_d3(1) = 0;
i_d5(1) = 0;
i_a(1) = 0;
i_b(1) = 0;
i_c(1) = 0;
i_dc(1) = 0;
v_rect(1) = 0;
v_dc(1) = 0;

% time
del_t = tau/100;
t_end = 5*T_ac;
t(1) = 0;
k = 1;

while t(k) < t_end

    e_a(k) = 100*cos(100*t(k));
    e_b(k) = 100*cos(100*t(k) - 2*pi/3);
    e_c(k) = 100*cos(100*t(k) + 2*pi/3);

    if i_a(k) > esp
        v_ag(k) = v_rect(k);
    elseif i_a(k) < -esp
        v_ag(k) = 0;
    else
        v_ag(k) = (v_rect(k)/(2*esp))*i_a(k) + (0.5*v_rect(k));
```

```
    end

    if i_b(k) > esp
        v_bg(k) = v_rect(k);
    elseif i_b(k) < -esp
        v_bg(k) = 0;
    else
        v_bg(k) = (v_rect(k)/(2*esp))*i_b(k) + (0.5*v_rect(k));
    end

    if i_c(k) > esp
        v_cg(k) = v_rect(k);
    elseif i_c(k) < -esp
        v_cg(k) = 0;
    else
        v_cg(k) = (v_rect(k)/(2*esp))*i_c(k) + (0.5*v_rect(k));
    end

    v_a(k) = (2/3)*v_ag(k) - (1/3)*v_bg(k) - (1/3)*v_cg(k);
    v_b(k) = (2/3)*v_bg(k) - (1/3)*v_ag(k) - (1/3)*v_cg(k);
    v_c(k) = (2/3)*v_cg(k) - (1/3)*v_ag(k) - (1/3)*v_bg(k);

    i_a(k+1) = i_a(k) + (del_t/L_ac)*(e_a(k) - v_a(k)); % need init
    i_b(k+1) = i_b(k) + (del_t/L_ac)*(e_b(k) - v_b(k)); % need init
    i_c(k+1) = i_c(k) + (del_t/L_ac)*(e_c(k) - v_c(k)); % need init

    if i_a(k+1) > 0
        i_d1(k+1) = i_a(k+1); % need int
    else
        i_d1(k+1) = 0;
    end

    if i_b(k+1) > 0
        i_d3(k+1) = i_b(k+1); % need init
    else
        i_d3(k+1) = 0;
    end

    if i_c(k+1) > 0
        i_d5(k+1) = i_c(k+1); % need init
    else
        i_d5(k+1) = 0;
    end

    i_dc(k+1) = i_d1(k+1) + i_d3(k+1) + i_d5(k+1); % need init

    v_rect(k+1) = (1/(1+(del_t/tau))) * (v_rect(k) + ((L_dc*(i_dc(k+1)
 - i_dc(k)))/tau) + ((del_t*i_dc(k+1)*R_load)/tau)); % need init
    v_dc(k+1) = R_load*i_dc(k+1); % need init

    t(k+1) = t(k) + del_t;
    k = k+1;
end
```

```
theta_ac = w_ac.*t;

n = find(t >= 3*T_ac & t < 4*T_ac);
m = find(t >= 2*T_ac & t < 4*T_ac);

v_dc_avg = average(v_dc(n),T_ac,del_t)
i_dc_avg = average(i_dc(n),T_ac,del_t);

p_out_avg = v_dc_avg*i_dc_avg

p_in = v_a(n).*i_a(n) + v_b(n).*i_b(n) + v_c(n).*i_c(n);
p_in_avg = average(p_in,T_ac,del_t)

efficiency = (p_out_avg/p_in_avg)*100

subplot(2,2,1)
plot(theta_ac(m),i_a(m))
ylim([-750 750])
xlabel('\theta_a_c')
ylabel('i_a')
title('Phase a cuurent')

subplot(2,2,2)
plot(theta_ac(m),v_rect(m))
ylim([0 200])
xlabel('\theta_a_c')
ylabel('V_d_c'''')
title('V_d_c'''')

subplot(2,2,3)
plot(theta_ac(m),v_dc(m))
ylim([0 200])
xlabel('\theta_a_c')
ylabel('V_d_c')
title('V_d_c')

subplot(2,2,4)
plot(theta_ac(m),i_dc(m))
ylim([-800 800])
xlabel('\theta_a_c')
ylabel('i_d_c')
title('Output DC current')

if mode == 12
    sgtitle(['Plots for R_l_o_a_d = ', num2str(R_load_12), '\Omega
 (Borderline Mode 1 & 2)'])
elseif mode == 23
    sgtitle(['Plots for R_l_o_a_d = ', num2str(R_load_23), '\Omega
 (Borderline Mode 2 & 3)'])
else
    sgtitle(['Plots for R_l_o_a_d = ', num2str(R_load), '\Omega (Mode
 ', num2str(mode),')'])
end
```

Published with MATLAB® R2019a