```
% ECE414 - Take home exam
% Name: Mohammed H. Al-Sayegh
% pidtune Contorller Tabale

function specs_table(G)
C_p = pidtune(G, 'P');
C_pd = pidtune(G, 'PD');
C_pi = pidtune(G, 'PI');
C_pid = pidtune(G, 'PID');
C_pidf = pidtune(G, 'PIDF');
C_pdf = pidtune(G, 'PDF');

for(conrtoller_num = 0:9)

    switch(conrtoller_num)
    case 0
        %%
        h = zeros(19,7);
        Info_type =  ["Info type"; "RiseTime"; "SettlingTime";
        "SettlingMin"; "SettlingMax"; "Overshoot"; "Undershoot"; "Peak";
        "PeakTime"; "Umax"; "EssStep"; "EssRamp"; "Gm";
        "Pm"; "Wcg"; "Wcp"; "Vm"; "Wvm"; "Smax";];
        h = [Info_type];
        type = "Info type";

    case 1
        %%
        c = getallspecs(G,C_p);
        type = "P Controller";

    case 2
        %%
        c = getallspecs(G,C_pd);
        type = "PD Controller";

    case 3
        %%
        c = getallspecs(G,C_pi);
        type = "PI Controller";
        % Using pidsearch to generate a calibrate version of PI controller
        C_pi = pidsearch(G,C_pi,'OS');

    case 4
        %%
        c = getallspecs(G,C_pid);
        type = "PID Controller";

    case 5
        %%
        c = getallspecs(G,C_pidf);
        type = "PIDF Controller";
```

```
        % Using pidsearch to generate a calibrate version of PI controller
        C_pidf = pidsearch(G,C_pidf,'OS');

    case 6
        %%
        c = getallspecs(G,C_pdf);
        type = "PDF Controller";
        % Using pidsearch to generate a calibrate version of PI controller
        C_pdf = pidsearch(G,C_pdf,'OS');

    case 7
        %%
        type = "A unity feedback lineara lgebraic Controller w/ stepitae";
        [N,D] = stepitae(6,5,10,'classic');

        % D is the controller in tf or zpk form.
        % T is the closed loop transfer function in zpk form.
        % Tu is the control effort transfer function in zpk form.
        % Td is the disturbance transfer fuction in zpk form.
        % L is the loop transfer function D(s)*G(s) in zpk form.

        [D,T,Tu,Td,L]= lamdesign(G,D);
        c = getallspecs(G,D);
        nyquist(L);

    case 8
        %%
        type = "A unity feedback lineara lgebraic Controller w/ stepshape";
        [N,D] = stepshape(6,5,10);

        % D is the controller in tf or zpk form.
        % T is the closed loop transfer function in zpk form.
        % Tu is the control effort transfer function in zpk form.
        % Td is the disturbance transfer fuction in zpk form.
        % L is the loop transfer function D(s)*G(s) in zpk form.

        [D,T,Tu,Td,L]= lamdesign(G,D);
        c = getallspecs(G,D);

    case 9
        %%
        type = "2-parameters linear-alg controller";
        T = steplqr(G,0.2698);

        % F is the feedforward controller in zpk form.
        % H is the feedback controller in zpk form.
        % Tu is the control effort transfer function zpk form.
        % Td is the disturbance transfer fuction in zpk form.
        % L is the loop transfer function G(s)*H(s) in zpk form.

        T = tf(T);
```

```
        s = tf('s');
        s.Numerator{1,1} = T.Denominator{1,1};

        R = roots(s.Numerator{1,1});
        R = real(R);
        R = R';

        [F,H,Tu,Td,L]= lamdesign(G,T,R);
        c = getallspecs(G,H);
    end

    if(conrtoller_num > 0)
        specs = [type; c.RiseTime; c.SettlingTime; c.SettlingMin;
                 c.SettlingMax; c.Overshoot; c.Undershoot; c.Peak;
                 c.PeakTime; c.Umax; c.EssStep;c.EssRamp; c.Gm; c.Pm;
                 c.Wcg; c.Wcp; c.Vm; c.Wvm; c.Smax;];

        h = [h specs];

        % save a controllers matrix specs into an excell file
        xlswrite('specs.xls', h);
    end
end


% Infinite peak control effort (the size of the signal at the plant input)
consider_umax = zeros(1,6);
array_inf = h(10,:) ~= "Inf";

% Steady state error not equal to zero for a step input
consider_ess = zeros(1,6);
array_umax = h(11,:) == "0";

disp("The consider contorller are :")

for i = 1:10
    if (array_inf(i) && array_umax(i))
        switch(i)
            case 2
                disp("P Controller");
                type = 'P';
                pidsearch_calibration(G,type);
            case 3
                disp("PD Controller");
                type = 'PD';
                pidsearch_calibration(G,type);
            case 4
                disp("PI Controller");
                type = 'PI';
                pidsearch_calibration(G,type);
            case 5
```

```
                disp("PID Controller");
                type = 'PID';
                pidsearch_calibration(G,type);
            case 6
                disp("PIDF Controller");
                type = 'PIDF';
                pidsearch_calibration(G,type);
            case 7
                disp("PDF Controller");
                type = 'PDF';
                pidsearch_calibration(G,type);
            case 8
                disp("A unity feedback lineara lgebraic Controller w/ stepitae");
                [N,D] = stepitae(6,5,10,'classic');
                [D,T,Tu,Td,L]= lamdesign(G,D);
                figure(8);
                nyquist(L);
            case 9
                disp("A unity feedback lineara lgebraic Controller w/ stepshape");
                [N,D] = stepshape(6,5,10);
                [D,T,Tu,Td,L]= lamdesign(G,D);
            case 10
                disp("two parameter linear algebraic design controller");
        end
    end
end
```