

Esta función recibe como entrada 2 matrices G y H de un sistema de control y devuelve la solución para cada uno de los 7 pasos para calcular el LDR a mano

```
function trazarLDR(G, H)
    % Paso 1: Identificar la ubicación de polos y ceros en el lazo abierto
    GLazoAbierto=G*H;
    polosLazoAbierto=pole(GLazoAbierto)
    cerosLazoAbierto=zero(GLazoAbierto)

    % Paso 2: Número de ramas (orden del sistema en lazo cerrado)
    numeroRamas=max(size(polosLazoAbierto),size(cerosLazoAbierto))

    % Paso 3: Identificación de segmentos sobre el eje real
    combinedNumbers=[polosLazoAbierto; cerosLazoAbierto];
    combinedRealNumbers=[];

    for i = 1:length(combinedNumbers)
        if isreal(combinedNumbers(i))
            combinedRealNumbers=[combinedRealNumbers,combinedNumbers(i)];
        end
    end

    orderedReal=sort(combinedRealNumbers);

    for i = 1:2:length(orderedReal)
        segmentoDelEjeReal = [orderedReal(i), orderedReal(i + 1)]
    end

    % Paso 4: Cálculo de asíntotas (LGR para  $K \rightarrow \infty$ )
    asintotas=length(polosLazoAbierto) - length(cerosLazoAbierto)

    for i=0:asintotas-1
        anguloAsintotas(i+1)=(2*i+1)*180/asintotas;
    end
    anguloAsintotas

    cenroideAsintotas=(sum(polosLazoAbierto)-sum(cerosLazoAbierto))/asintotas

    %Para los siguientes pasos necesitamos los coeficientes para
    %poder aplicar Routh-Hourwitz
    [numG, denG]=tfdata(G);
    [numH, denH]=tfdata(H);

    syms s K
    GSimbolica=poly2sym(cell2mat(numG), s)/poly2sym(cell2mat(denG), s);
    HSimbolica=poly2sym(cell2mat(numH), s)/poly2sym(cell2mat(denH), s);

    lazoAbiertoSimbolico=collect(K*GSimbolica/(1+K*GSimbolica*HSimbolica));

    [numLC, denLC]=numden(lazoAbiertoSimbolico);
```

```

coeficientesLazoAbierto=coeffs(denLC, s, 'all');

% Paso 5: Puntos de corte con el eje imaginario
a1=coeficientesLazoAbierto(1);
a2=coeficientesLazoAbierto(3);
a3=coeficientesLazoAbierto(5);
b1=coeficientesLazoAbierto(2);
b2=coeficientesLazoAbierto(4);

c1=((b1*a2)-(b2*a1))/b1;
c2=((b1*a3)-(0*a1))/b1;

d1=((c1*b2)-(c2*b1))/c1;

K_crit=double(solve(d1==0, K));
j=0;
for i=1:length(K_crit)
    if K_crit(i)>0
        j=j+1;
        aux(j)=K_crit(i);
    end
end
K_crit=aux;
poly_Aux=subs((c1*s^2+c2), K, K_crit);
ptos_Corte=double(solve(poly_Aux==0, s))

% Paso 6: Puntos de ruptura o salida del eje real
simbolicaDerivada=collect(diff(GSimbolica*HSimbolica,s));

puntosRuptura=double(subs(solve(simbolicaDerivada==0, s)))

% Paso 7: Ángulo de salida (o llegada) de las raíces
poloComplejo=polosLazoAbierto(3);
polosLazoAbiertoAngulo=polosLazoAbierto([1:2,4:end]);

angulosPolos=0;
for i=1:length(polosLazoAbiertoAngulo)
    aux=atan((imag(poloComplejo)-imag(polosLazoAbiertoAngulo(i)))/
(real(poloComplejo)-real(polosLazoAbiertoAngulo(i))));
    angulosPolos=angulosPolos+aux;
end
angulosZeros=0;
for i=1:length(cerosLazoAbierto)
    aux=atan((imag(poloComplejo)-imag(cerosLazoAbierto(i)))/
(real(poloComplejo)-real(cerosLazoAbierto(i))));
    angulosZeros=angulosZeros+aux;
end
angulosSalidaPolosComplejos=(pi-angulosPolos+angulosZeros)*180/pi

end

```