自動控制II-HW4

102303022 機械4A 賴言厚

**使用Matlab實現Z-transform系統響應**

**一、說明**

|  |
| --- |
| 1. **使用已知的txt內容,反推出系統的轉移函數** 2. **求出的轉移函數為Gp,在前面再放一Gs,使系統穩定** 3. **求出Gs的方法分為root locus 與 pole assignment** |

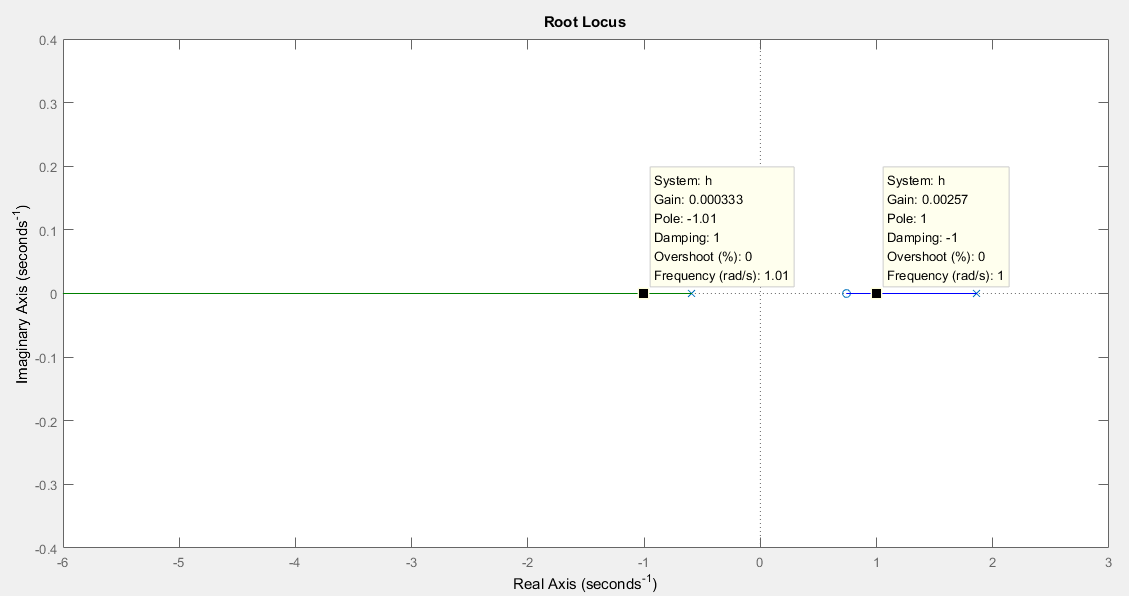
**二、程式碼(以下會逐段解釋):**

|  |  |
| --- | --- |
| clc;  clear;  **%% 設定參數**  totalstep = 98;  totaltheta = [0;0;0;0];  Lefttotal = [0;0;0;0];  Righttotal = [0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0];  theta = [0;0;0;0];  phi = [0;0;0;0];  out = load('xx.txt'); **%匯入矩陣u,y**  u = out(:,1);  y = out(:,2);  u1 = 0; u0 = 0; y1 = 0; y0 = 0; | 使用上課所提到的誤差法求解,先將係數都設好,並將初值設為0。  載下LMS上的TXT檔匯入矩陣,左行為u 右行為y。 |
| **%% 匯入y u的結果 算出轉移函數**  for i=1:totalstep  phi = [y(i+1);y(i);u(i+1);u(i)];  phiT = phi';  phiM = phi \* phiT;  Lefttotal = Lefttotal + y(i+2) .\* phi;  Righttotal = Righttotal + phiM;  end | 帶入最後誤差法推導出的公式,算出各矩陣關係。 |
| **%% 找root locus 觀察穩定狀況的gain值**  theta = double(inv(sym(Righttotal))) \* Lefttotal  b1 = theta(3); b0 = theta(4); a1 = -theta(1); a0 = -theta(2);  h = tf([b1 b0],[1 a1 a0]);  figure(1); rlocus(h); | 求出theta也就是轉移函數的係數a1 a0 b1 b0。  並畫出root locus(大圖在後幾頁),發現gain要小於0.000333且大於0.00257時,才會在Z-domain單位圓內,因此系統光靠Gain值無法使系統穩定。 |
| **%% pole assignment法**  **%設pole在 +0.5j -0.5j 0.4 => 係數=[1 -0.4 0.25 -0.1]**  A = inv([1 b1 0;a1 b0 b1;a0 0 b0]) \* [-0.4-a1;0.25-a0;-0.1];  x0 = A(1);  x1 = A(2);  x2 = A(3);  r1 = 0; r0 = 1; u1 = 0; u0 = 0; y2 = 0; y1 = 0; y0 = 0;  for i=1:100  %r = sin(i\*pi/20);  u0 = u1;  u1 = -x0\*u0 + x1\*(r1-y1) + x2\*(r0 - y0);  y2 = -a1\*y1 - a0\*y0 + b1\*u1 + b0\*u0;  y0 = y1;  y1 = y2;  r0 = r1;  yarr(i) = y2;  end  figure(2); plot(yarr); | 因此我們使用pole assignment求出Gs值。  把pole設在單位圓內,希望系統穩定。  這裡的程式碼r(k),也就是input為impulse。 |
| **P.S. 此程式碼最終跑出結果僅為 輸入=impulse時,經由pole assigment法後的系統響應圖。** | |

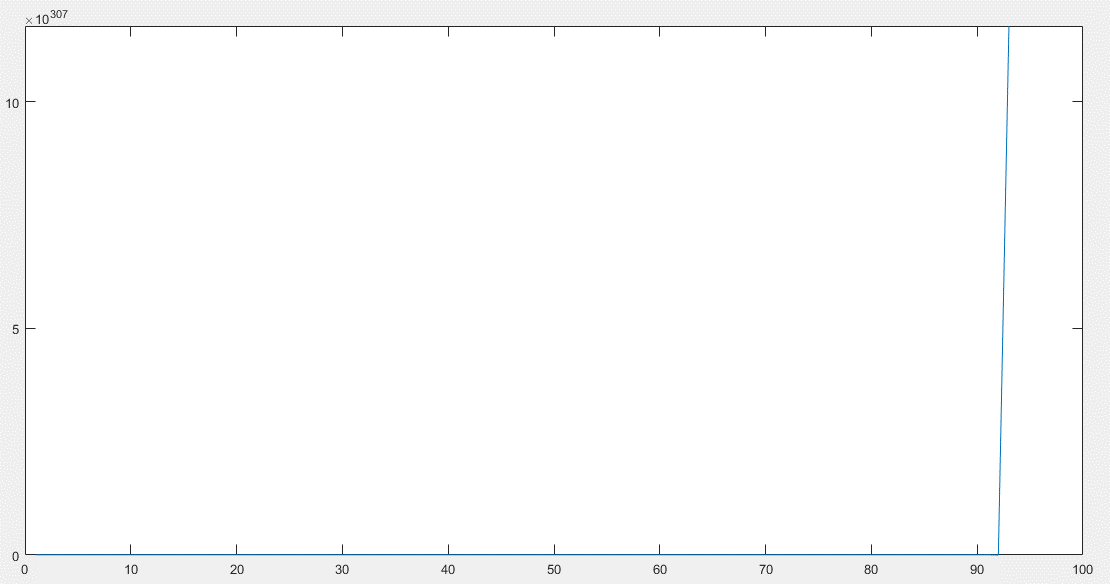
**三、圖:**

**原轉移函數:**

原轉移函數的root locus:

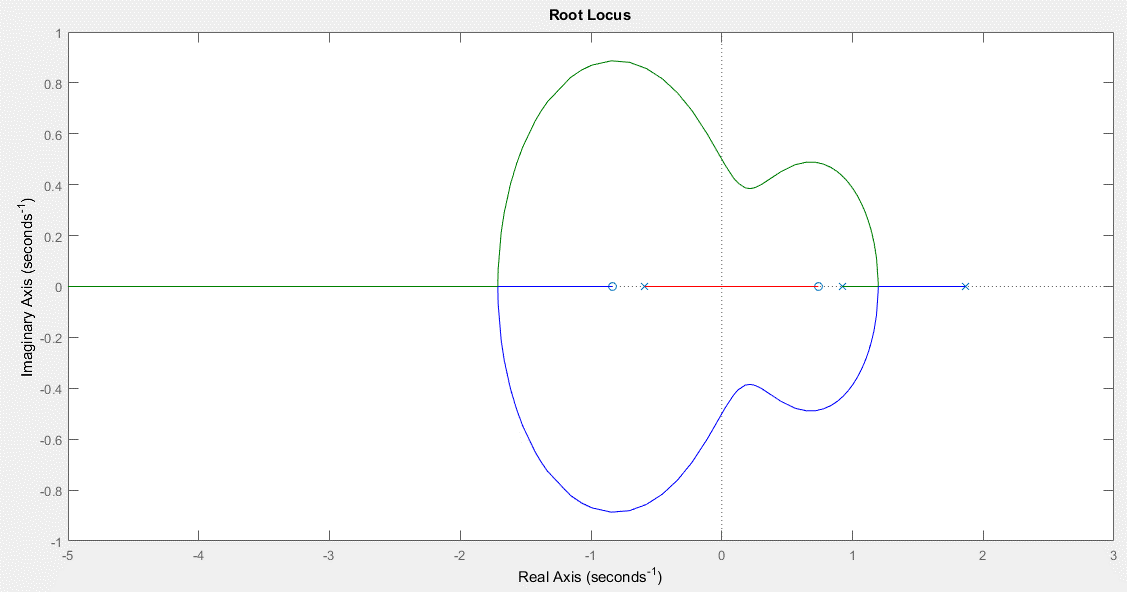


原轉移函數的響應圖 (當r(k)為impulse):

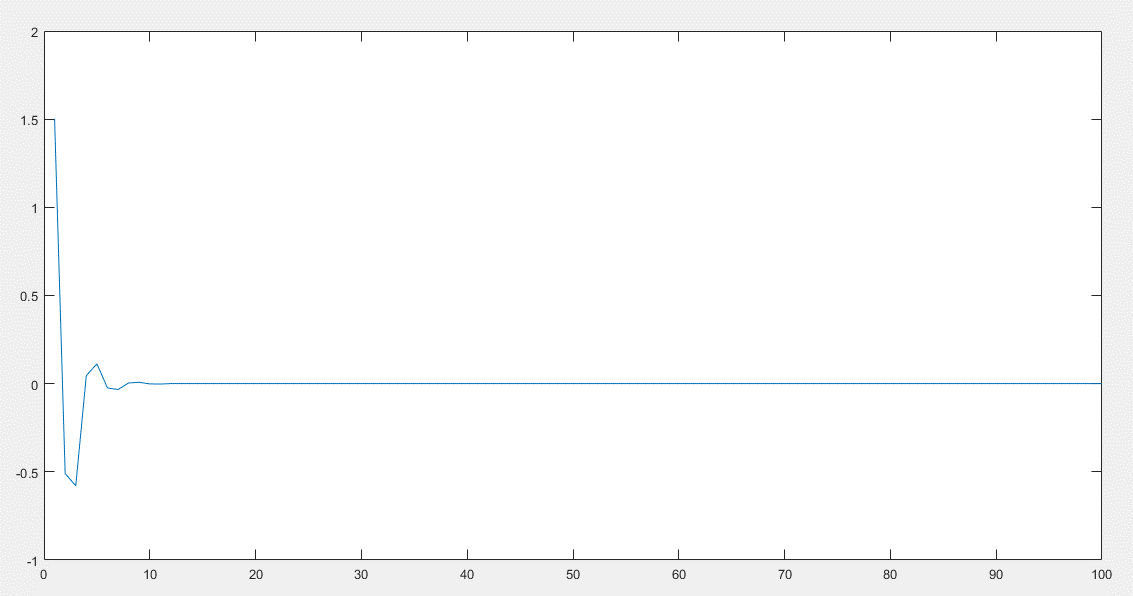


**使用Pole assignment法求出Gs後:**

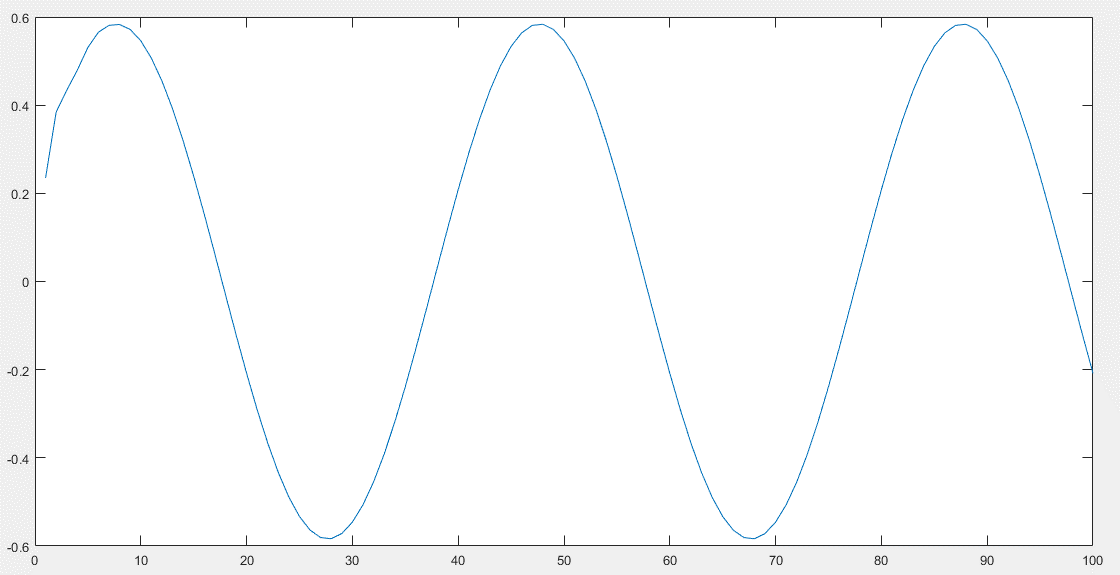
Gs\*Gp的Root locus



當r(k)為impulse時,系統響應圖



當r(k)為sin(i\*pi/20)時,系統響應圖



結果 : 系統穩定,無發散情形出現