**UMSA-FI**

**NOMBRE: moisés martín CONDORI YUJRA**

**DOCENTE: Ing. Javier SANABRIA GARCÍA**

**MATERIA: ANÁLISIS DE SISTEMAS LINEALES**

**SIGLA: ETN-506**

**CI:9103365 L.P.**

**1.- Realizar un programa en lenguaje C/C++, para el criterio de Routh, para los analizar los polos y ceros, si la FT. Esta en estabilidad.**

|  |
| --- |
| //UMSA-INGENIERIA ELECTRONICA-ETN506  //SISTEMAS LINEALES  //UNIV. moises martin CONDORI YUJRA  //CI 9103365LP  //SISTEMAS LINEALES ETN-506  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  int main(){  //declarando variables  int n,w,z,i,j,p,q,a,x,k,b,e,c,contador,l;  float t;  //definiendo los tamaños de las matrices de cada variable.  int A[50][50];  float P[50][50];  float R[50][50];  float P1[50][50];  float P2[50][50];  //datos personales  printf("ESTUDIANTE : moises martin CONDORI YUJRA \n");  printf("\n\n\t criterio de routh ");  //grado del polinomio  printf("\n\n digite el grado del polinomio: ");  scanf("%d",&z);  n=z+1;  t=0.000001;  //Introducir variables del polinomio por teclado  for(i=1;i<=1;i++)  {  for(j=1;j<=n;j++)  {  printf("coeficiente del termino de grado S[%d]: ",n-j);  scanf("%d",&A[i][j]);  }  }  printf("\n los coeficientes del poinomio : \n\n");  for(i=1;i<=1;i++)  {  for(j=1;j<=n;j++)  {  printf("\t[%2d]",A[i][j]);  }  }  //desarrollo del criterio de routh  printf("\n\ncriterio de routh: \n");  //separando pares e impares  if(z%2==0)  {  q=z/2;  w=q+1;  q=w;  a=w;  b=q;  p=1;  for(i=1;i<=1;i++)  {  for(j=n;j>=1;j--)  {  p=p+1;  if (p%2==0)  {  P[i][w]=A[i][j];  w=w-1;  }  else  {  P1[i][q-1]=A[i][j];  q=q-1;  }  }  }  }  else  {  q=n/2;  w=q;  a=w;  b=q;  p=1;  for(i=1;i<=1;i++)  {  for(j=n;j>=1;j--)  {  p=p+1;  if (p%2==0)  {  P1[i][w]=A[i][j];  w=w-1;  }  else  {  P[i][q]=A[i][j];  q=q-1;  }  }  }  }  printf("\n");  printf("\n");  //primer polinomio  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  printf("\t[%2f]",P[i][j]);  }  }  printf("\n");  for(i=1;i<=1;i++)  {  for(j=1;j<=b;j++)  {  printf("\t[%2f]",P1[i][j]);  }  }  //guardando el primer polinomio  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  R[i][j]=P1[i][j];  }  }  //operando los polinomios obtenidos  contador=0;  l=0;  for(k=0;k<=z-2;k++)  {  for(x=1;x<=a;x++)  {  P2[1][x]=(P1[1][1]\*P[1][x+1]-P[1][1]\*P1[1][x+1])/P1[1][1];  //contador de signo cambiante  if(P2[1][1]<0)  {  l=l+1;  }  }  //analizando el primer caso especial del criterio  if(P2[1][1]==0)  {  P2[1][1]=t;  //analizando el segundo caso especial del criterio  if(P2[1][2]==0)  {  if(P2[1][1]-t==0)  {  for(e=1;e<=a;e++)  {  c=n-contador-2\*e;  if(c>0)  {  P2[1][e]=c\*R[1][e];  }  else  {  P2[1][e]=0;  }  }  printf("\n");  printf("\n\t......caso especial....(caso 2) derivando.. \n");  }  }  else  {  printf("\n");  printf("\n\t.caso especial....(caso 1) reemplazando (e=0.000001).. \n");  }  }  //Mostrando el polinomio del residuo  printf("\n");  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  printf("\t[%2f]",P2[i][j]);  }  }  //Guardando el polinomio del residuo  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  R[i][j]=P2[i][j];  }  }  //obteniendo nuevos polinomios para ejecutarlos como bucles  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  P[i][j]=P1[i][j];  }  }  for(i=1;i<=1;i++)  {  for(j=1;j<=a;j++)  {  P1[i][j]=P2[i][j];  }  }  contador=contador+1;  }  printf("\n");  printf("\n");  //mostrar si el sistema es estable ó inestable  if(l>=1)  {  printf("\n\t...........el sistema es inestable.............\n ");  printf("\n");  printf("\n");  }  else  {  printf("\n\t...........el sistema es estable...........\n");  printf("\n");  printf("\n");  }  } |