**Fundamentos de Programación (grupos A y C)**

**(Grados en Ingeniería Mecánica, Electrónica Industrial y Química Industrial)**

**13 de junio de 2014**

***Ejercicios 1 y 2* (5*.0 p/ejercicio*)*:*** construir un programa en ***C*** ó ***Java*** lo más modular posible (atendiendo a los criterios de modularidad) y documentar el diseño preliminar con la definición de nuevas tipologías de datos, el diagrama de módulos (estructura del programa) y las interfaces de los módulos, y el diseño detallado con las definiciones de los respectivos sub-programas.

**Ejercicio 1:** Construir un programa para analizar la estabilidad de un sistema dinámico mediante el teorema de Routh-Hurwitz. Un sistema dinámico viene descrito por su ecuación característica G(s) (polinomio de grado n):

El criterio de estabilidad de Routh permite determinar en qué semiplano (izquierdo o derecho) del plano complejo están localizadas las raíces de una ecuación polinómica sin resolverla, y en consecuencia conocer si dicho sistema es estable (todos los polos están en el semiplano izquierdo) o inestable (hay al menos un polo en el semiplano derecho). El procedimiento consiste en lo siguiente:

1. Si el polinomio tiene al menos un coeficiente nulo o bien algún coeficiente negativo en presencia de al menos un coeficiente positivo, entonces existe al menos una raíz (o raíz imaginaria) con parte real positiva, y el sistema es inestable. Si todos los coeficientes son negativos, se cambian de signo.
2. Si todos los coeficientes son positivos y no nulos se procede a construir la siguiente tabla: en las dos primeras filas se alternan los coeficientes del polinomio, y las siguientes filas se calculan según las fórmulas indicadas: (nótese que el grado del polinomio va disminuyendo una unidad al pasar a la siguiente fila, y que cada fila solo contiene bien términos pares o bien términos impares)

…

…

El criterio de Routh establece que la cantidad de raíces positivas (o con parte real positiva) es igual a la cantidad de cambios de signo en los coeficientes de la primera columna de la tabla anterior. ***Nota***: si durante este proceso se obtiene una fila cuyo primer elemento es ***0*** pero algún otro elemento de la misma es no nulo, se produce una degeneración, deteniéndose el proceso. Si toda la fila es cero, se sustituye dicha fila por los coeficientes de la derivada de la fila inmediatamente anterior y se continúa el proceso, generándose un sistema inestable si hay raíces positivas y un sistema críticamente estable si no las hay.

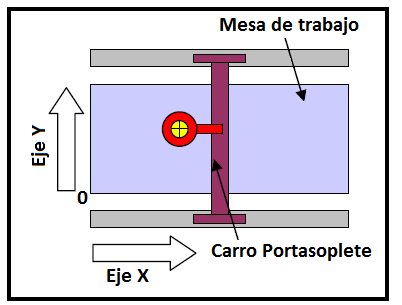
El programa leerá inicialmente por teclado el grado del polinomio (máximo ***100***) y los coeficientes de cada término del mismo. A continuación, comprobará los casos triviales de inestabilidad, y si no hay coeficientes nulos y todos tienen el mismo signo, se calculará e imprimirá en pantalla la matriz de Routh y se procederá a determinar si el sistema dinámico asociado es estable, inestable (al menos una raíz tiene parte real positiva), críticamente estable ó se ha producido una degeneración en el cálculo, presentando el correspondiente mensaje en la pantalla. Ejemplos:

|  |  |  |  |
| --- | --- | --- | --- |
| **G(s)= s4+10\*s3+35\*s2+50\*s+124** | **G(s)= s5+3\*s4+2\*s3+6\*s2+3\*s+3** | **G(s)= s5+4\*s4+2\*s3+5\*s2+3\*s+6** | **G(s)= s4+2\*s3+11\*s2+18\*s+18** |
| s4🡪 1.00 35.0 124.0  s3🡪 10.00 50.00  s2🡪 30.00 124.00  s1🡪 8.67  s0🡪 124.00 | s4🡪 1.00 2.00 3.00  s3🡪 3.00 6.00 3.00  s2🡪 0.00 2.00 | s5🡪 1.00 2.00 3.00  s4🡪 4.00 5.00 6.00  s3🡪 0.75 1.50  s2🡪 -3.00 6.00  s1🡪 3.00  s0🡪 6.00 | s4🡪 1.00 11.00 18.00  s3🡪 2.00 18.00  s2🡪 2.00 18.00  s1🡪 0.00  s1🡪 4.00 **🡨d(2\*s2+18)/ds**  s0🡪 18.00 |
| Sistema estable 🡪 no hay raíces en semiplano derecho | Degeneración en el cálculo 🡪 primer elemento de fila es cero | Sistema inestable 🡪 dos raíces positivas (hay dos cambios de signo en primera columna) | Sistema críticamente estable 🡪  Fila cero  No hay raíces positivas |

**Ejercicio 2:** considerar una máquina de corte por láser de chapas finas de metal como la mostrada en la figura. Las dimensiones de la mesa son 4000 mm x 2000 mm. Construir un programa interactivo que permita generar un archivo de texto con los códigos de control numérico para los diferentes cortes a efectuar en una lámina metálica dada. Considerar que solo se pueden efectuar cortes rectangulares y circulares:

|  |  |
| --- | --- |
| **Código CN** | **Significado** |
| **1 x,y** | Mover soplete a posición en coordenadas absolutas |
| **2** | Inicio de ciclo de corte (activa láser) |
| **3** | Fin de ciclo de corte (desactiva láser) |
| **4 ±x,±y** | Corte lineal hasta la posición final en coord. relativas |
| **5 ±xc,±yc,±** | Corte circular. (xc,yc): centro circunferencia en coord. relativas, **±:** sentido corte (+:anti-horario, -: horario) |

Nota: las coord. relativas son con respecto a la posición actual del soplete. Todas las coordenadas están expresadas en milímetros



El corte por láser produce una ranura de 1 mm aproximadamente, la cual es compensada automáticamente por la máquina, generándose ésta siempre a la derecha del movimiento. Nótese que si el corte es para producir una pieza, el movimiento del soplete deberá ser anti-horario, y si es para un agujero (desperdicio) el sentido deberá ser horario, con el fin de obtener las piezas/agujeros con las medidas exactas.

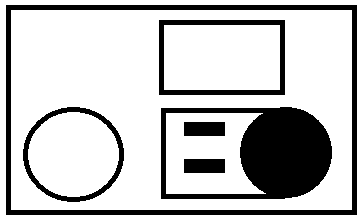
El programa pedirá inicialmente por teclado las dimensiones de la chapa metálica (en mm), las cuales deberán ser inferiores a las de la mesa de trabajo, y presentará a continuación un menú en pantalla para gestionar la lista de cortes a efectuar (hasta un máximo de 100), con las siguientes opciones, seleccionándose una de ellas por teclado, tras cuya ejecución se volverá a presentar nuevamente el menú de opciones, terminándose la ejecución del programa cuando se seleccione explícitamente la opción de finalización:

1. Añadir un nuevo corte. El programa leerá por teclado y registrará en la lista (si ello es posible) la siguiente información adecuadamente validada:
   * Tipo de corte: pieza o desperdicio.
   * Forma del corte: rectangular (en posición estándar) o circular.
   * Dimensiones del corte:
     + Corte rectangular:
       - Abscisa y ordenada de la esquina inferior izquierda.
       - Anchura del corte.
       - Altura del corte.
     + Corte circular:
       - Abscisa y ordenada del centro de la circunferencia.
       - Radio de la circunferencia.

Al leer la información del corte, el programa deberá ir comprobando que el corte está dentro de los límites de la chapa.

1. Listar todos los cortes registrados, de uno en uno, presentando en pantalla su información asociada.
2. Eliminar un corte. Se solicitará por teclado las coordenadas de un punto del plano, y se presentarán en pantalla todos los cortes que incluyan dicho punto, de uno en uno, preguntándole al usuario que confirme el borrado del mismo y realizando la acción pertinente.
3. Clasificar los cortes según las coordenadas de su punto central. Los cortes se ordenarán de izquierda a derecha (clave primaria) y de abajo a arriba (clave secundaria).
4. Generar un archivo de texto con el código de control numérico de los cortes a efectuar, ejecutándose en primer lugar los cortes de los desperdicios y a continuación los de las piezas. A fin de optimizar los movimientos del carro, el orden de los desperdicios será el descrito en el apartado 4, y el de las piezas el orden inverso.
5. Finalizar la ejecución del programa.

|  |
| --- |
| **Cortes clasificados** |
| Pieza-Circular-(500,500),400  Desperdicio-Rectangular-(1300,300),300,100  Desperdicio-Rectangular-(1300,700),300,100  Pieza-Rectangular-(1100,100),1100,800  Pieza-Rectangular-(1100,1100),1100,800  Desperdicio-Circular-(2200,500),400 |



Ejemplo: