**Examen Parcial Fundamentos de Programación**

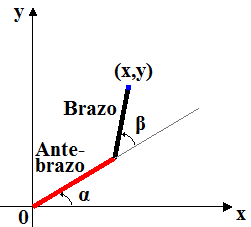
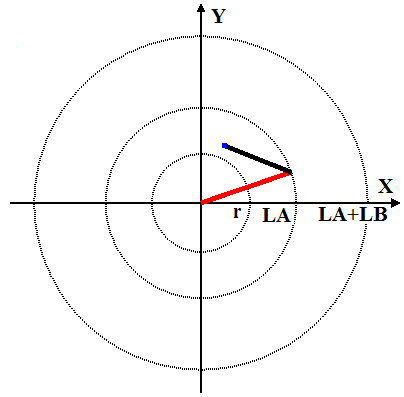
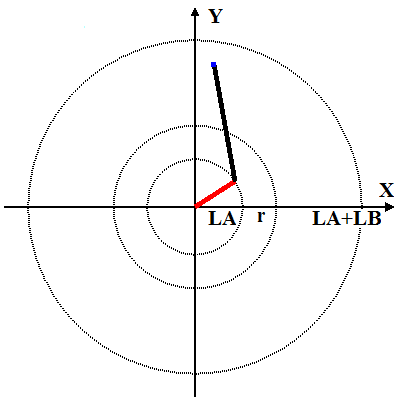
**(Grados en Ingeniería Mecánica, Electrónica Industrial y Química Industrial)**

**15 de abril de 2011**

***Ejercicio 1 (4.0 p):*** construir un programa en C lo más modular posible (atendiendo a los criterios de modularidad). Documentar el diseño preliminar con el diagrama de módulos (estructura del programa) y los prototipos de las funciones en C, y el diseño detallado con las definiciones de las respectivas funciones en C.

***Ejercicios 2 y 3 (3.0 p/ejercicio):*** implementar en C solamente la función indicada (no hay que construir el programa completo).

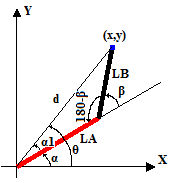
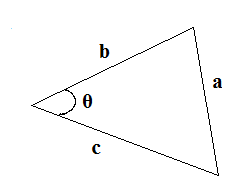
**Ejercicio 1:** Considerar un robot articulado bidimensional que consta de dos elementos rectangulares, un antebrazo de longitud **LA** con un extremo fijo anclado al origen de coordenadas, que puede girar un ángulo **α** cualquiera respecto del eje **X+**, y un brazo de longitud **LB** unido al otro extremo del antebrazo, que puede girar un ángulo **β** (comprendido entre 0º y 150º) en sentido anti-horario con respecto al eje del antebrazo. El movimiento del sistema robotizado es controlado mediante los dos ángulos **α** y **β**, pudiendo acceder el extremo libre del brazo solamente a las posiciones cartesianas comprendidas en la corona circular de radios **r** y **LA+LB**, donde:

Construir un programa en C para controlar el movimiento del robot a posiciones específicas del plano; el programa leerá por teclado una a una las posiciones finales **(x,y)** de posicionamiento del robot; cada punto introducido será validado para comprobar si es accesible, desplazándose a continuación el robot a la posición indicada (escribir en pantalla los desplazamientos angulares en grados correspondientes de los dos elementos del robot y la posición angular final). En caso de que dicha posición sea inalcanzable, se presentará un mensaje volviéndose a leer un nuevo punto. El programa finalizará cuando se introduzca como posición final el origen del sistema de coordenadas **(0,0)**, en cuyo caso los dos elementos del robot se mueven a la posición angular **(α,β)=(0,0)**. Ejemplo: LA=100.0 LB=75.0 (α0,β0)=(0,0)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Posición final** | | **Desplazamiento angular** | | **Posición angular final**  Nota:  **αf:** [0,360] grados  **∆α:** [-180,180] grados  (menor desplazamiento angular del antebrazo posible) | |
| **x** | **y** | **∆α (°)** | **∆β (°)** | **αf (°)** | **βf (°)** |
| 100 | 25 | -29.28 | 109.47 | 330.72 | 109.47 |
| 200 | 200 | Posición inalcanzable | | 330.72 | 109.47 |
| 25 | 75 | 53.06 | 19.21 | 23.79 | 128.68 |
| 125 | -25 | -71.10 | -41.07 | 312.69 | 87.61 |
| 0 | 0 | 47.31 | -87.61 | 0.00 | 0.00 |

Nota: en las siguientes figuras se ilustran respectivamente, la relación entre las coordenadas del punto final y el posicionamiento angular del robot, así como las relaciones dentro de un triángulo entre los lados y un ángulo del mismo.

**(x,y)**: coordenadas cartesianas

**(d,θ)**: coordenadas polares

**α=θ-α1**

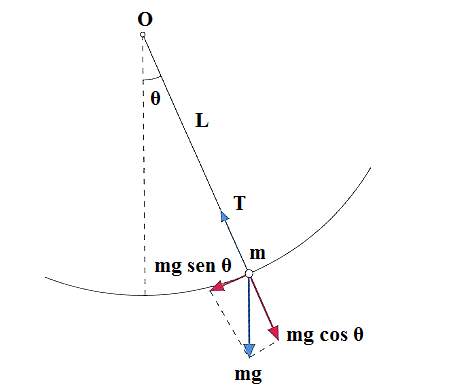
Nota: la función **acos** devuelve un valor comprendido en **[0,PI]** radianes

**Ejercicio 2:** Un péndulo simple consta de una pequeña partícula material de masa **m** suspendida de un hilo inextensible de masa despreciable y longitud **L**, que se separa un pequeño ángulo **θ** respecto de su posición vertical de reposo, empezando a oscilar la partícula con un movimiento armónico simple. La ecuación general del periodo de oscilación **T** viene dada por la serie:

El periodo de oscilación se puede determinar con suficiente precisión si se considera el suficiente número de términos de la serie anterior.

Para oscilaciones de pequeña amplitud **θ**, el periodo se puede calcular mediante la siguiente expresión:

El error relativo cometido con esta aproximación viene dado por la expresión:



Construir una función en C, lo más eficiente posible, que devuelva el valor en grados de la máxima elongación o amplitud posible **θ** con una precisión dada, de manera que el error relativo cometido en dicha aproximación del periodo de oscilación (calculada con 100 términos de la serie) sea inferior a un valor dado como argumento. Construir una segunda función que presente en pantalla una tabla con los valores de los errores relativos cometidos en las elongaciones que cumplan dicha condición calculados con diferente número de términos (1, 10, 100 y 1000) y con el formato indicado en el ejemplo. Ejemplo:

Error relativo máximo: 0.001 (0.1%)

Precisión: 1.0 (grados)

Elongación máxima: 7.000 grados

Tabla de errores relativos

Numero de términos de la serie

Elongacion(Grados) 1 10 100 1000

0.000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000

1.000 0.000019038105451 0.000019038921006 0.000019038921006 0.000019038921006

2.000 0.000076146622613 0.000076159671566 0.000076159671566 0.000076159671566

3.000 0.000171308155678 0.000171374216705 0.000171374216705 0.000171374216705

4.000 0.000304493717522 0.000304702506088 0.000304702506088 0.000304702506088

5.000 0.000475662738532 0.000476172485987 0.000476172485987 0.000476172485987

6.000 0.000684763078966 0.000685820116037 0.000685820116037 0.000685820116037

7.000 0.000931731044835 0.000933689390824 0.000933689390824 0.000933689390824

**Ejercicio 3:** Construir una función en C que escriba en pantalla un patrón de rayas inclinadas (135º) formado por asteriscos (\*), dados como argumentos el número de filas, el nº de columnas y la distancia de separación (en líneas/caracteres) entre dos líneas del patrón (medida vertical/horizontalmente); si esta última distancia fuera negativa, no se dibujaría nada. El patrón comienza en la primera línea de la primera columna. Ejemplos:

1

2

3

4

5

6

7

8

9

0

1

2

NF=12 NC=30 SEP=5

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \***

NF=12 NC=30 SEP=3

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \***

NF=12 NC=30 SEP=1

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \* \* \* \* \* \* \***

123456789012345678901234567890

123456789012345678901234567890