

Sesión 3 de RobotSTUDIO

Realizada por Jesús Carrascosa Carro

##Actividad realizada en clase

Durante la clase, he desarrollado la versión del programa propuesta, cuyo objetivo es trazar o dibujar una onda sinusoidal a partir de una amplitud dada en los ejes X e Y, y con una resolución determinada por un número n de puntos ($nPuntos$).

Esta onda sinusoidal se encuentra contenida dentro de un rectángulo imaginario de dimensiones $2 * amplitudY$ por $2 * amplitudX$.

El desplazamiento con respecto al punto inicial ($pini$) se ha calculado utilizando reglas de tres, relacionando el índice i . del punto actual con su posición dentro del total de $nPuntos$, de la siguiente manera:

- **Desplazamiento en X:** Se aplica una regla de tres sencilla. Se divide el doble de la amplitud en X entre el número total de puntos y se multiplica por el índice i . Esto proporciona el desplazamiento horizontal correspondiente a ese punto.
- **Desplazamiento en Y:** Se reparten los 360 grados (una vuelta completa) entre $nPuntos$, y se multiplica por el índice i para obtener el ángulo correspondiente. A este ángulo se le aplica la función seno, y el resultado se multiplica por la amplitud en Y para obtener el desplazamiento vertical.

A cada uno de estos desplazamientos se le suman las coordenadas iniciales en X e Y del punto inicial ($pini$). Este proceso se realiza de forma iterativa para cada uno de los puntos que componen el vector $pVector$, el cual almacena la trayectoria completa de la onda. Posteriormente, esta trayectoria se recorre en un bucle por separado para su ejecución.

Avanzado

Se solicita que, adicionalmente, la herramienta esté siempre **orientada según la tangente** de la trayectoria en el plano X-Y.

Para resolverlo, he utilizado la función $ATan2$, que permite calcular el **ángulo de la tangente** a partir de las coordenadas X e Y del punto actual. Esta función devuelve el ángulo de orientación correspondiente a la dirección del vector tangente en ese punto.

Con el ángulo obtenido, he sustituido el valor del **ángulo Z** en la orientación del nuevo punto, **manteniendo los valores originales de los ángulos X e Y** del punto `pini` . La diferencia entre las versiones proporcionadas radica en el cálculo de la `Atan2` :

```
xDesp:= (i/nPuntos) * 2*amplx;  
yDesp:= Sin ((i/nPuntos)*360)* amplY;  
angleAux := ATan2(xDesp +pini.trans.x,yDesp+pini.trans.y); V2  
angleAuxX:= EulerZYX(\X,pini.rot);  
angleAuxY:= EulerZYX(\y,pini.rot);  
oriAux:= OrientZYX (angleAux,angleAuxY,angleAuxX) ;  
pVector{i}:= pini;  
pVector{i}.rot:= oriAux;  
pVector{i}.trans.x:= pVector{i}.trans.x + xDesp;  
pVector{i}.trans.y:= pVector{i}.trans.y + yDesp;
```

- V1: Se calcula utilizando sólo los desplazamientos.
- V2: Se calcula utilizando los desplazamientos más las coordenadas x e y del punto inicial.

No he observado ninguna diferencia perceptible entre ellas.

Anotaciones varias

- Los videos se encuentran en sus carpetas correspondientes.
- Tecnicamente la resolución de la onda se podría considerar como `nPuntos+1`(el inicial) ya que el bucle empieza en 1 , se puede solucionar de 2 formas:
 - i. Codificando que el primer indice del vector corresponda al punto inicial.
 - ii. Haciendo la asignación con `nPuntos-1` y luego incorporar a la posición inicial el punto inicial desplazando en una posición el resto de puntos(requiere al menos 2 vectores uno de `nPuntos-1` que guarda la asignación inicial sin el punto inicial y otro de `nPuntos` con la asignación final, más un bucle for adicional).