

Memoria P2

Aplicación de adhesivo con perfil senoidal configurable

Francisco José Caballero del Campo

Contenido

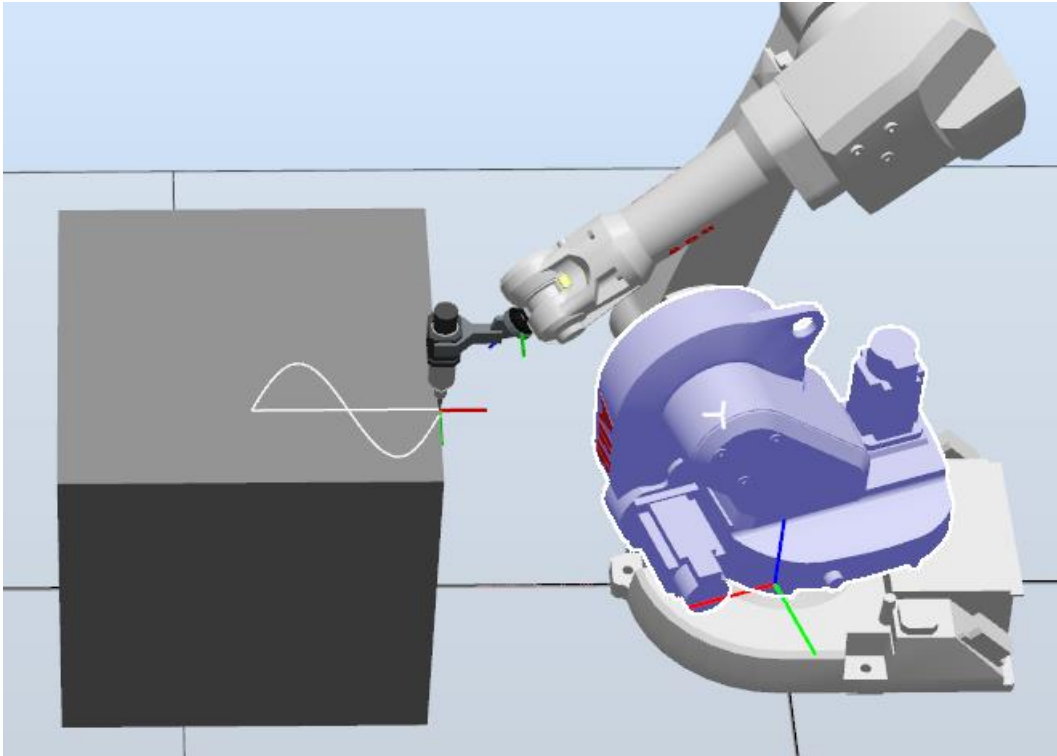
Ejercicio 12

Ejercicio 23

Ejercicio 2 extendido4

Ejercicio 1

Estación de partida.



Montaje del entorno de trabajo según lo descrito en clase por el profesor.

Ejercicio 2

Tarea a realizar.

En este ejercicio, tenemos que dibujar la onda del seno, definiendo para ello una serie de puntos (definidos en una constante) y además, a la función senoidal, le pasaremos como parámetros, un punto de partida y la amplitud y la longitud de la onda.

Lo he resuelto según las explicaciones dadas en clase. He hecho uso de los incrementos, calculando su valor según el numero de puntos que queramos.

Dentro del bucle “FOR”, calculamos las posiciones de los puntos tanto en el eje X como en el Y, siendo el valor en el eje X el ángulo que se le proporciona a “sin(x)” y el valor del eje Y el valor devuelto tras calcular el seno multiplicado por la amplitud para ajustarlo a la amplitud requerida (la función seno solo da valores entre 1 y -1).

Código utilizado:

```
PROC main()
    !Añada aquí su código
    MoveL pInicio,v100,z0,MyTool\WObj:=wobj0;
    Senoide pInicio, 300, 50;
    MoveL pInicio,v100,z0,MyTool\WObj:=wobj0;
ENDPROC

PROC Senoide(robtarg pIni, num amplX, num amplY)

    CONST num nPuntos := 100+1;
    VAR robtarget tabPos{nPuntos}; !Array de puntos
    VAR num posX;
    VAR num posY;
    VAR num incr;
    VAR robtarget pApoyo;

    !Añadimos el punto inicial
    tabPos{1} := pIni; ! Punto inicial añadido a la tabla de puntos

    !Calculamos el incremento
    incr := (amplX*2) / nPuntos;

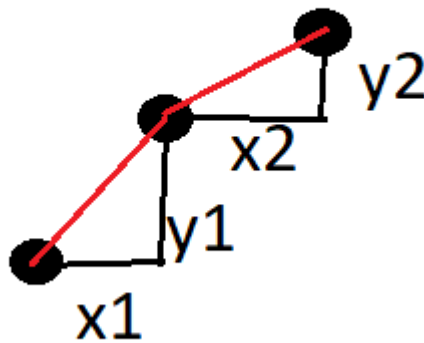
    !Creamos la tabla de puntos
    FOR i FROM 2 TO nPuntos DO ! Empieza en 0 o 1 ( empezamos en 1 porque el 0 será la posición inicial)
        !funcion seno
        posX := ((i*incr) * 360) / (amplX * 2); !Angulo que le pasamos al seno
        posY := amplY * Sin(posX); !Calculamos la altura de Y según su posición en el eje X (calculado el angulo previamente posX)

        pApoyo := Offs (pIni, incr * (i - 2), posY, 0); !Hacemos el punto
        tabPos{i} := pApoyo; !Guadamos el punto en la tabla
    ENDFOR

    !Recorremos la tabla de puntos
    FOR i FROM 1 TO nPuntos DO
        MoveL tabPos{i},v100,z0,MyTool\WObj:=wobj0;
    ENDFOR
ENDPROC
```

Ejercicio 2 extendido

Para este ejercicio, usaremos el mismo código que en el apartado anterior, pero usando el cuaternio de Euler para conocer la rotación inicial de los puntos, para, tras calcular la rotación requerida, utilizar la función de Euler “orient”.



Sabemos que la función senoidal del apartado anterior, esta formada por una sucesión de puntos, de forma que si usamos una buena cantidad de puntos, las líneas rectas que unen cada punto serán imperceptibles a simple vista.

Si observamos el dibujo arriba mostrado, podemos observar que existen 2 variables (X e Y), siendo ambas variables conocidas.

- La variable X es igual al incremento, es decir, cada punto esta separado entre si por la misma distancia en el eje X.
- La variable Y es diferente para cada par de puntos, pero conociendo el punto de partida, podemos ir restando, de manera recursiva, el valor de la variable Y anterior.

De esta forma, y sabiendo 2 de sus lados, podemos calcular la distancia entre 2 puntos consecutivos aplicando el algoritmo de Pitágoras.

Haciendo uso de la trigonometría y sabiendo el valor de todos los catetos de los triángulos que forman dos puntos consecutivos, usando la fórmula conocida de $\sin(x) = \text{Cateto opuesto} / \text{hipotenusa}$, podremos calcular el valor del seno que forma el cateto X con la hipotenusa, tras realizar esta operación, despejamos el valor del ángulo, usando el arcoseno, y ya tendríamos resuelto el problema.

Usando el cuaternio de EulerZYX() para conocer la rotación inicial de los puntos y sumándoles el valor del ángulo calculado al eje Z en la función OrientZYX(), el aplicador de silicona iría de manera tangente a la función senoidal.

Código realizado:

```
PROC main()
!Añada aquí su código
MoveL pInicio,v100,z0,MyTool\WObj:=wobj0;
Senoide pInicio, 150, 100;
MoveL pInicio,v100,z0,MyTool\WObj:=wobj0;

ENDPROC

PROC Senoide(robtarget pIni, num amplX, num amplY)

CONST num nPuntos := 100+1;
VAR robtarget tabPos(nPuntos); !Array de puntos
VAR num posX;
VAR num posY;
VAR num incr;
VAR robtarget pApoyo;
!Variables para la rotacion de la pieza
VAR num angX;
VAR num angY;
VAR num angZ;
VAR robtarget pAux;
VAR num hipotenusa;
VAR num CatOp;
VAR num CatCo;
VAR num posY2;
VAR num seno;
VAR num angulo;

!Añadimos el punto inicial
tabPos(1) := pIni; ! Punto inicial añadido a la tabla de puntos

!Calculamos el incremento
incr := (amplX*2) / nPuntos;

!Inicializamos
posY2 := 0;

!Creamos la tabla de puntos
FOR i FROM 2 TO nPuntos DO ! Empieza en 0 o 1 ( empezamos en 1 porque el 0 será la posición inicial)
!funcion seno
posX := ((i*incr) * 360) / (amplX * 2); !Angulo que le pasamos al seno
posY := amplY * Sin(posX); !Calculamos la altura de Y según su posicion en el eje X (calculado el angulo previamente posX)

pApoyo := Offs (pIni, incr * (i - 2), posY, 0); !Hacemos el punto

!Calculamos la hipotenusa, el cateto contiguo y el cateto opuesto
CatCo := incr;
CatOp := posY - posY2; !Y actual - anterior
posY2 := posY; !Actualizamos el valor anterior al actual
hipotenusa := Sqrt((CatCo * CatCo) + (CatOp * CatOp));

!Sacamos el seno para sacarle el angulo haciendo arcoseno
seno := CatOp / hipotenusa;
angulo := ASin(seno);

!Sacamos el cuaternio de EulerZXY()
angZ := EulerZYX(\Z, pApoyo.rot);
angY := EulerZYX(\Y, pApoyo.rot);
angX := EulerZYX(\X, pApoyo.rot);

!Guardamos la rotación de los puntos
pAux := pApoyo;
pAux.rot := OrientZYX(angZ + angulo,angY,angX);

tabPos(i) := pAux; !Guadamos el punto en la tabla

ENDFOR

!Recorremos la tabla de puntos
FOR i FROM 1 TO nPuntos DO

!Movemos el brazo
MoveL tabPos(i),v100,z0,MyTool\WObj:=wobj0;

ENDFOR

ENDPROC
```