

INTERPRETE DE CAD Y SU APLICACIÓN PARA OPTIMIZACIÓN DE PARÁMETROS DE CONTROL DE UN ROBOT INDUSTRIAL

**Ing. Gustavo González, Ing. Guillermo Gutiérrez,
Ing. Carlos Cagliolo, Sr. Darío Chans**

CUDAR, UTN-FRC, Córdoba, Argentina, ggonzalez@scdt.frc.utn.edu.ar, fgutierrez@scdt.frc.utn.edu.ar, ccagliolo@iua.edu.ar, dchans@scdt.frc.utn.edu.ar

Abstract: este trabajo presenta la descripción del desarrollo de un software experimental intérprete de CAD que además ayuda a la puesta a punto del sistema de control de un robot industrial.

This paper presents the description of a CAD interpreter experimental software development and also helps to set up an industrial robot control system.

Palabras Claves: Autocad®, Trayectoria, Gráfico, Puntos, Comunicación, Control, Interpolación.

1. INTRODUCCIÓN

El proyecto “Desarrollo de un robot con arquitectura de control abierta”, tiene como objetivo fundamental poner operativo un robot industrial de seis grados de libertad, la finalidad de mismo es proporcionar una herramienta destinada a la profundización de prácticas académicas.

Este robot marca ACMA Robotique fue donado por la empresa Renault Argentina sin el sistema de control, por lo que una de las tareas del proyecto fue la implementación de un sistema de control electrónico de los actuadores de la estructura electromecánica (que sí estaba operativa), con el reacondicionamiento de algunas partes menores.

Con la finalidad de usar el robot con una estructura abierta y teniendo en cuenta que se debía poner a punto el sistema de control, surgió la necesidad de realizar la comunicación desde un medio con el sistema, siendo una de las maneras más directas y rápidas la que se implementa a través de un ordenador.

En base a esto es que se decidió emitir órdenes desde una PC, pero éstas debían tener algo más que un par ordenado que sugiriese un punto u otro: debía tener coherencia en una cierta secuencia. Sin dudas que para lograr esto es conveniente realizar previamente un gráfico que emule la trayectoria deseada y luego obtener el conjunto de puntos que la conforma. Una de las maneras más adecuadas y eficaz de lograrlo con una computadora fue mediante el uso de un CAD (Computer Aided Design) y de un software traductor o intérprete el cual fue desarrollado en Visual Basic® bajo entorno Windows® y si bien no está finalizado en todas sus facetas ha servido a modo de primera prueba de comunicación con el exterior del Robot ante la solicitud de algún recorrido.

2. DESARROLLO

Una vez definida como primera herramienta al AUTOCAD®, surge la necesidad de desarrollar un Software Intérprete entre los archivos de salida del Autocad y el software de control del Robot.

El programa fue propuesto como herramienta de laboratorio, con el objeto de probar y poner a punto los algoritmos de control y los sistemas de actuación del Robot. Evaluando así su capacidad para seguir trayectorias programadas.

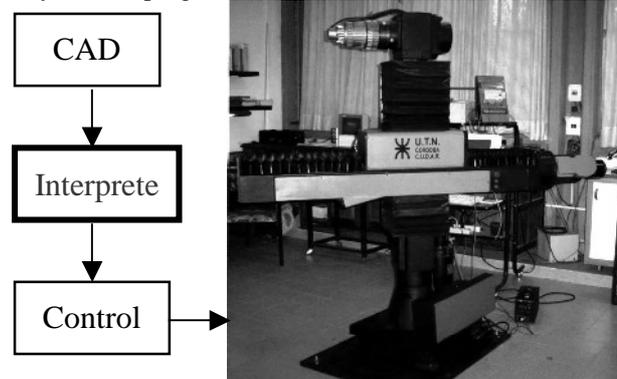


Figura 1. Secuencia de procesos.

Existen ciertas aplicaciones para manejar objetos de AutoCAD de una manera distinta de lo que lo hace éste, ya que realiza una vectorización en un lenguaje de máquina muy complicada de interpretar para quien no es especialista en ese tipo de programación. Una de estas aplicaciones se denomina Auto LISP®, la cual permite manejar la información en forma distinta y programar en un lenguaje accesible. La intención del trabajo no fue aprender este lenguaje de programación, sino aplicar uno difundido, como por ejemplo Microsoft Visual Basic, que es ampliamente conocido y utilizado.

El hecho de utilizar un lenguaje visual permite realizar un cambios en forma dinámica, inclusive por

personas que no posean conocimientos de programación o computación teniéndolo en cuenta para posibles aplicaciones en el futuro.

En primera instancia lo realizado en AutoCAD debe ser almacenado en un archivo de tipo DXF (Drawing Exchange Format) que será la forma de exportar la trayectoria requerida en el plano. Este formato contempla todos los objetos realizados por AutoCAD y que pueden ser utilizados por la aplicación AutoLISP para el manejo de elementos de dibujo y diseño.

Para dejar documentado el comportamiento del robot se adicionó en el extremo del brazo un electroimán con un dispositivo mecánico que soportaba un lápiz o fibrón (figura 2) y de esta forma se utiliza similarmente a un plotter o mesa XY sobre una superficie plana (un papel) para observar y documentar así la trayectoria o camino desarrollado.

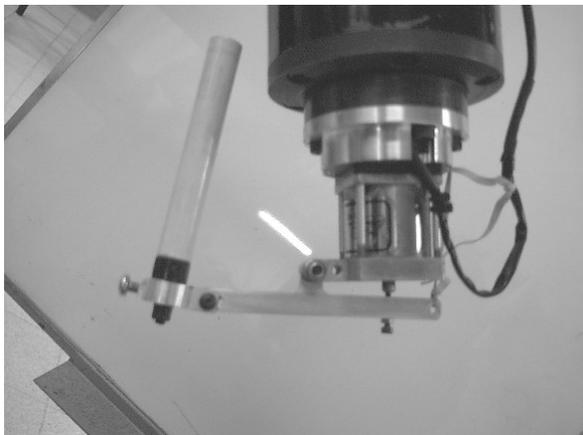


Figura 2. Electroimán se soporte para lápiz.

En base a las disponibilidades del sistema electrónico y electromecánico es que se requirió del software traductor la identificación de:

- La trayectoria a partir de lo almacenado por AutoCAD en un archivo DXF.
- Determinación en la secuencia de puntos.
- Los instantes de operación que debía tener el lápiz del robot.

De acuerdo a esto es que se resolvió comenzar con la identificación de dos tipos de trayectorias: recta y circular. Para la primera, se dibujó en un plano previamente definido y limitado, una recta con inicio y final bien determinados. En este caso particular, las coordenadas son (110;70) y (210;80).

Los archivos del AutoCAD fueron generados con la extensión DXF y posteriormente fueron abiertos como archivos de texto. Así se identificó la recta dibujada de la siguiente manera:

```
1 LINE
2 5
```

```
3 1D9
4 330
5 1F
6 100
7 AcDbEntity
8 8
9 0
10 100
11 AcDbLine
12 10
13 110.0 Coord x de inicio
14 20
15 70.0 Coord y de inicio
16 30
17 0.0 Coord z de inicio
18 11
19 210.0 Coord x final
20 21
21 80.0 Coord y final
22 31
23 0.0 Coord z final
24 0
```

De esto se observa: en la línea 1, el tipo de elemento dibujado; la línea 2 a 10 son elementos propios para el dibujo en Autocad; en la 11 se da nuevamente el tipo de elemento parcial y luego en las líneas 12, 14 y 16 se identifican con 10, 20 y 30 las coordenadas x-y-z de inicio que son escritas inmediatamente después de cada una; desde las líneas 18 a 24 ocurre lo mismo para las coordenadas finales.

Si bien este formato brinda la posibilidad de trabajar en el espacio (3D), esta aplicación se realizó sólo en el plano (2D) debido a la restricción física del sistema electromecánico, donde estaban habilitados solamente dos ejes del robot. De todas maneras, se puede hacer extensivo a 3D y cuando esté habilitado el tercer eje del robot, este software intérprete se podrá utilizar para una aplicación espacial.

El segundo elemento con el que se decidió trabajar es el arco de círculo para lo cual se dibujó un arco de círculo con centro en (120,120) y radio 25, determinándose la escritura del formato DXF como:

```
1 ARC
2 5
3 1DA
4 330
5 1F
6 100
7 AcDbEntity
8 8
9 0
10 100
11 AcDbCircle
12 10
13 120.0 Coord x del centro
14 20
15 120.0 Coord y del centro
16 30
```

17	0.0	Coord z del centro
18	40	
19	25.0	Radio
20	100	
21	AcDbArc	
22	50	
23	180.0	Angulo de Inicio
24	51	
25	0.0	Fin de ángulo
26	0	

En la línea 1 se identifica el elemento como un arco de círculo, desde la línea 2 a la 10 ocurre lo mismo que para el caso de la línea recta, en la 11 el tipo de elemento que la compone; 12, 14 y 16 son los indicadores de las coordenadas x, y, z del centro del arco mediante los números 10, 20 y 30; en la 18 el número 40 indica que el valor siguiente será el radio (25); las líneas 20 y 21 son tomadas como la 2 a la 10; finalmente la 50 y la 51 son indicadores del ángulo de inicio y final respectivamente a partir de los cuales se dibujará el arco.

En este contexto es lo mismo hablar de arcos o círculos ya que los segundos los podemos asumir como arcos cerrados que tienen un ángulo de inicio de 0° y final de 360°.

Una vez identificado el formato de los elementos descriptos la principal tarea fue reconocerlos dentro de un archivo (DXF) que contuviera varios elementos distintos, que se consiguió a partir de los elementos de búsqueda del software desarrollado.

El software realizado en Visual Basic presenta una pantalla como se observa en la figura 3. Hay 3 zonas bien demarcadas: el panel de graficación, la zona de Proceso de Datos y Graficación y finalmente la zona de Transmisión de Datos:

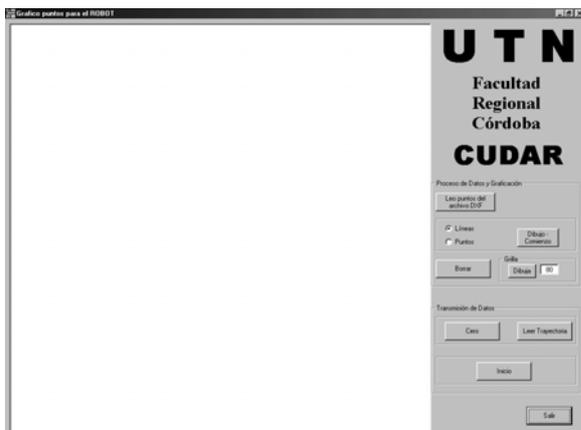
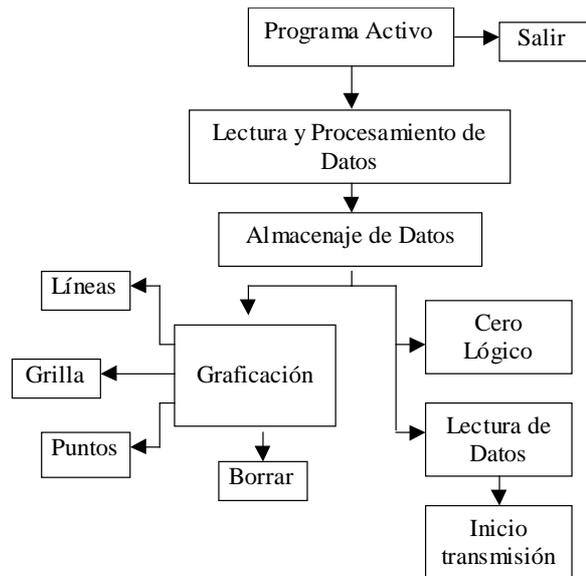


Figura 3. Presentación de la pantalla.

El resumen en bloques del programa es el siguiente:



Primero se opera en la zona de Proceso de Datos y Graficación con el comando “leer puntos del archivo DXF” el cual en su rutina tiene como principal tarea identificar las palabras "AcDbLine" y "AcDbCircle" ya que como se vio antes, a continuación de estas se encuentran las coordenadas y datos necesarios para reproducir ya sea en el panel (cuadro blanco en la izquierda de la figura 1) o en el robot lo realizado en AutoCAD.

El bloque del software que realiza esto presenta un lazo (Do - Loop) en el cual se lee como archivo de datos a modo de texto línea por línea del archivo hasta que se encuentran las palabras que corresponden a líneas o círculos. Cuando ocurre esto se procede a leer y a almacenar en otro archivo de texto (.txt) en forma secuencial y encolumnadas las coordenadas o datos correspondientes. Estos son previamente multiplicados por un factor de escala que en nuestro caso es 80 debido a los pulsos del sensor que posee el robot y luego son redondeados a fin de trabajar con números enteros.

En el caso especial del arco de círculo se debe generar su trazo con pequeñas líneas rectas que en nuestro caso se han hecho con el criterio de dividir el ángulo contenido en 50 partes iguales. Sin dudas que mientras más divisiones posea este ángulo mejor será el trazado del arco, aunque no se debe permitir un exceso indiscriminado de puntos por las consecuencias que puede acarrear un exceso en la memoria del control.

Una vez que se finaliza con la lectura de datos una etiqueta anuncia esto y se pasa a un filtrado automático donde:

- 1) Se añaden varios ceros de coordenadas para asegurar que el robot inicie y finalice su movimiento en este punto y salvando así la posibilidad de que se pierda alguno de estos en su transmisión.

- 2) Los datos obtenidos sirven para 2 motores y se desea dejar el control previsto para los 6 motores así que se incluyen 4 columnas de ceros más para estos.
- 3) Se agrega una columna más, indicadora para el estado del lápiz del robot (1 ó 0) identificando para cada punto si es inicio o final del total de la línea lo que se realiza en el software comparando de a dos puntos, ya que en el programa utilizado, una línea es indicada siempre por un punto de inicio y otro final; si el final de una línea coincide con el inicio de la siguiente significará que el lápiz debe permanecer asentado.

Una vez generado el archivo completo se puede dibujar en el panel correspondiente que tiene dimensiones limitadas en (0,0) – (400,400) [medidas en milímetros] simulando el panel real de dibujo sobre el cual trabajará la mano del robot. Ordenamos el inicio del dibujo mediante el comando **Dibujar – Comienzo** el que a su vez admite dos posibilidades de ilustración, una de línea y otra de puntos. Esto significa que si hacemos uso de la opción línea se representará en el panel el dibujo tal cual fue realizado en AutoCAD y que posteriormente deberá hacer el robot, con lo que podemos además comprobar si las operaciones de lectura del archivo DXF y el filtrado fueron realizadas correctamente. Con la opción puntos se podrá visualizar cada uno de los puntos intervinientes en el gráfico y de este modo detectar posibles desvíos de la mano del robot por insuficiencias del control. Las figuras 4 y 5 muestran respectivamente los gráficos de líneas y puntos que se utilizaron para poner a punto el sistema de control. Se eligieron 3 figuras básicas: un círculo para estudiar las distintas segmentaciones (división del ángulo de contención del arco), un triángulo para estudio de interpolación lineal de puntos y un rectángulo con vértices redondeados para comprobar el control de cada motor independiente y la posterior actuación de ambos.

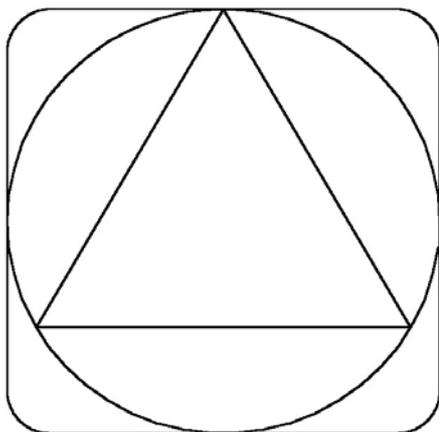


Figura 4. Gráfico de calibración.

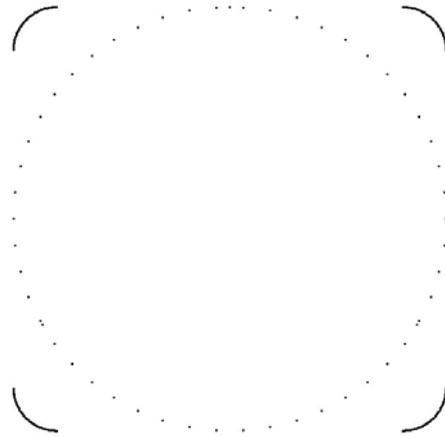


Figura 5. Gráfico de calibración.

Una vez realizado el gráfico en el panel se puede realizar un zoom o ampliación de ventana en zonas que se deseen observar detalles con mayor nitidez para contrastar lo realizado por el robot en posibles zonas de fallo. Esto es mostrado en las figuras 6 y 7. Otro elemento importante que se encuentra disponible en esta zona es un comando para la creación de una grilla establecido previamente su valor el cual por defecto es 80 y está relacionado justamente con los puntos del sensor del robot. Con esta facilidad se pueden verificar puntos muy próximos o trazos de recta de un punto a otro en donde por alguna deficiencia en el control del motor correspondiente se desvíe levemente.

Finalmente con la tecla BORRAR se puede limpiar el panel gráfico para generar nuevas trayectorias a probar o estudiar.

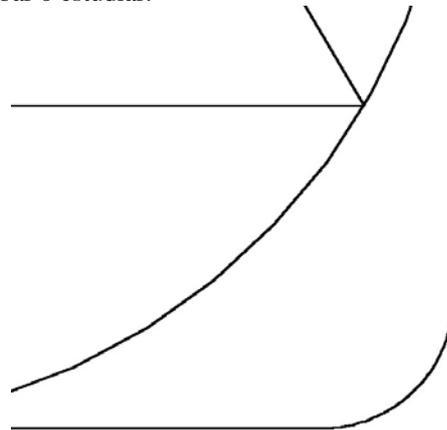


Figura 6.



Figura 7.

Una vez que se tienen procesados y bien definidos los puntos de la trayectoria se puede proseguir con la transmisión de los datos en forma serial RS485 al control trabajando en la zona de pantalla correspondiente.

Primeramente se realiza la búsqueda del cero de la zona de trabajo del robot para coordinar este y el cero de la zona dibujo o movimiento. En el panel hay etiquetas de aviso de las operaciones que se le ordena al control. Luego mediante la tecla “Leer trayectoria” se envían los datos que posteriormente serán ejecutados a través del comando “Inicio”.

Dado que en esta etapa del desarrollo el sistema de control no se encontraba preparado para tener en cuenta cambios de los parámetros físicos con la posición de la articulación, es decir, las variaciones de velocidades o aceleraciones debido a la inercia, o efectos de acoplamiento entre articulaciones, es que se procede a hacer una interpolación lineal, mediante software, con pares de puntos que se encuentran tan distantes subdividiendo la distancia en partes más pequeñas ya que sino los dos ejes no podrán alcanzar los puntos requeridos al mismo tiempo¹. En un futuro este software no se incluirá ya que el objetivo final en el tipo de control es que sea monoarticular y de esta manera adaptar las velocidades de cada uno de los motores de los ejes para que lleguen al punto ordenado de manera simultánea.

Una vez realizada la interpolación se procedió en una primera instancia a la puesta a punto del control de los motores obteniéndose resultados que se pueden apreciar en las figuras 8 (a, b, c y d). En este documento sólo se han incluido algunos pasos donde se puede apreciar como influyó la puesta a punto que se fue realizando al control hasta conseguir el resultado de la figura 8d.

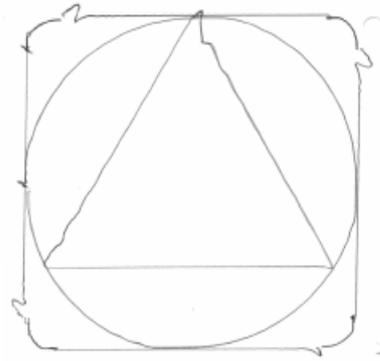


Figura 8a.

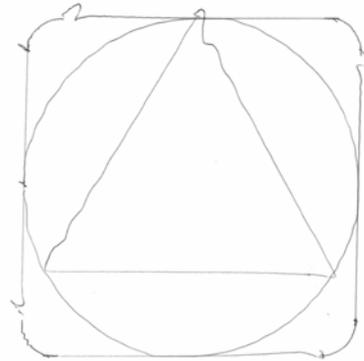


Figura 8b.

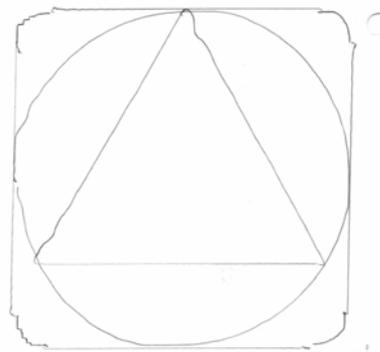


Figura 8c.

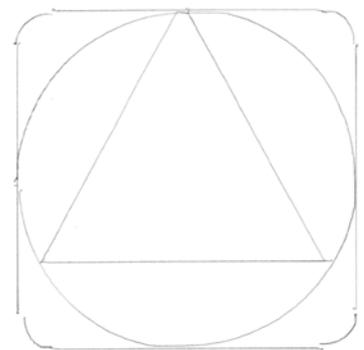


Figura 8d.

¹ Software de control para Robot Industrial de seis grados de libertad

3. CONCLUSIONES

Los resultados obtenidos hasta el momento han sido altamente satisfactorios ya que se consiguieron los objetivos planteados, es decir: se pudo realizar una interpretación de algunos de los elementos utilizados por AutoCAD, se logró transmitir los datos en un formato previamente acordado y se optimizaron los parámetros de control del robot que hasta ese momento estaban disponibles. Dadas las posibilidades del trabajo desarrollado es que se finalizó realizando un gráfico como lo muestran las figuras 9 (en líneas) y 10 (en puntos) obteniéndose como resultado una gráfica realizada por el lápiz adosado a la mano del robot como se ve en la figura 11.

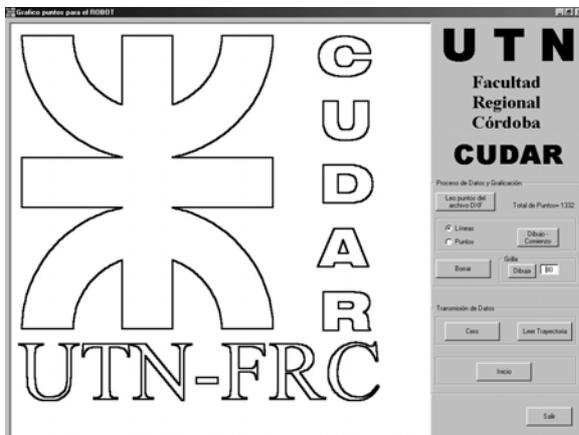


Figura 9. Gráfico en líneas.

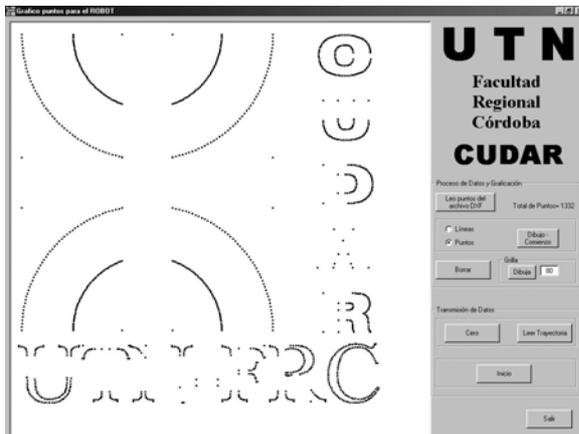


Figura 10. Gráfico en puntos.

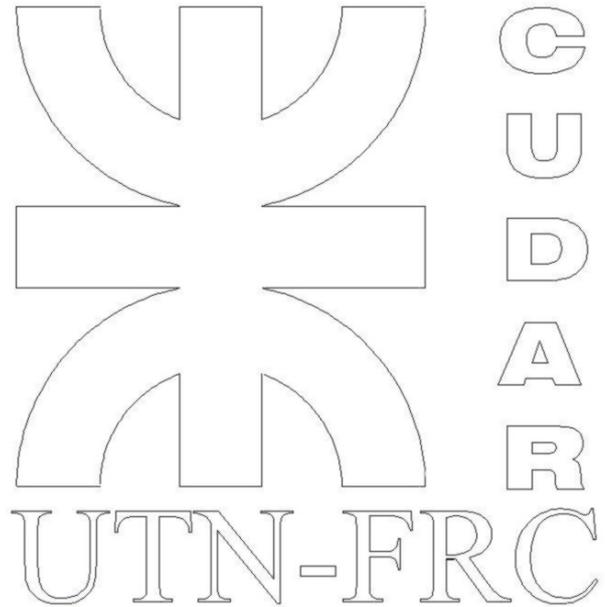


Figura 11. Resultado obtenido sobre un papel en el panel de dibujo.

Queda planteado para el futuro hacer una extensión de las sentencias detectadas en AutoCAD para incluir otros elementos posibles como polilíneas y curvas además de añadir la tercera dimensión.

BIBLIOGRAFÍA

- Balfagón, Albert y Pere Ripio (1995) *Visual Basic – Programación avanzada para Windows*.
- Autocad 2000 User guide – Autodesk