

Based on AutoCAD software and using AutoLisp programming language to develop a customized pipeline drawing system

Hui-Chin Chang

HungKuo Delin University of Technology, Department of Creative Product Design, Taipei, Taiwan
chang.hcjang@gmail.com

Abstract—A pipeline diagram is an engineering drawing used to show and describe the layout and connection relationships of pipes, pipelines, valves, instruments and other components in a system. It not only provides detailed layout and connection information of the piping system, but also serves as the basis for optimal design analysis of the entire piping system, so it plays an important role in various engineering fields. AutoLisp is a programming language developed specifically for AutoCAD software. It allows users to extend and automate AutoCAD functions by writing custom programs. Therefore, this article is based on AutoCAD software and uses AutoLisp programming language to develop a customized pipeline diagram drawing system to improve the efficiency of pipeline diagram drawing.

Keywords—*Pipelining diagram, AutoLisp*

I. INTRODUCTION

Pipeline transport of fluids is the most common and convenient method. Common parts of the piping system include pipes, pipe joints, pipe valves, pipe brackets, etc. The diagram used to depict the piping device is called a piping diagram. It is an important tool in system design, helping engineers plan and design piping systems to ensure that all components are connected correctly to meet the system's functional and performance requirements. Detailed layout and connection information of the piping system is also provided to help construction personnel correctly install and connect pipes, valves, instruments and other components to ensure the normal operation of the system. While the system is running, the piping diagram can be used to locate problems and perform maintenance. When a system failure occurs, engineers can refer to the piping diagram to find the fault location and repair it faster. It also helps to identify and manage dangerous points in the pipeline system, such as high-pressure areas, dangerous goods transportation pipelines, etc. This is important for developing safe operating procedures and emergency plans. It can also be used to train new employees to let them understand the structure and operating procedures of the pipeline system. In addition, it is also part of the system documentation, providing a traceable record of the system's history and configuration. By analyzing the pipeline diagram,

engineers can identify bottlenecks and improvement points in the system, thereby optimizing the design and improving system efficiency and performance. Generally speaking, piping diagrams play a key role in different stages of engineering projects. From design and construction to operation and maintenance, piping diagrams are required to ensure the safe, reliable and efficient operation of the system.

Although there are many pipeline drawing software on the market, they all have various shortcomings. For example, AutoCAD Plant 3D is a powerful three-dimensional drawing software specially made for factory design and pipeline design. It provides rich features, including automatic report generation, model collision detection, etc. However, the learning curve is steep and may require more training time than other software. In addition, the higher price of software is also one of its disadvantages [1]. AVEVA PDMS is a professional 3D factory design software. It has powerful functions and supports a variety of design needs, so it is widely used in large-scale engineering projects. However, its interface may not be friendly enough for beginners and requires considerable learning costs. Compared with other lighter-weight software, PDMS may seem too complex for small projects [2]. SmartPlant 3D is a three-dimensional model-based engineering design software with powerful collaborative design and information management functions. However, its disadvantages are that it is more expensive, requires higher hardware, and requires considerable training and resource investment [3]. SolidWorks Piping is an add-on module for SolidWorks that can be integrated with SolidWorks 3D modeling software. It provides an intuitive user interface and advanced 3D modeling capabilities. However, its performance for large-scale engineering projects may not be as good as some professional factory design software, and it also requires SolidWorks 3D modeling capabilities to get started. Therefore, the initial investment cost and the training cost for novices are factors that users must consider [4]. AutoCAD P&ID is professional software for drawing piping and instrumentation diagrams that is easy to use and compatible with other AutoCAD products. However, its functions are relatively simple and may not be enough to meet the needs of large-scale and customized projects [5].

AutoLISP (AutoCAD Lisp) is a programming language developed specifically for AutoCAD (Auto

Computer-Aided Design) software. It allows users to expand and automate AutoCAD functions by writing custom programs. It has the following functions; AutoLISP allows users to create custom commands to simplify and speed up common tasks in AutoCAD. By writing LISP programs, users can customize project- or industry-specific commands to better fit their workflow. It can also be used to automate a series of complex tasks, such as drawing, modifying entities, creating layers, etc. This helps increase productivity and reduces manual time and errors. And parametric design can be realized. By defining parameters in the program, the design project has adjustable characteristics. Such a design approach improves design flexibility and maintainability. AutoLISP can also be used to process and modify graphic elements in AutoCAD, including extracting information from drawings, modifying the properties of entities, creating or deleting objects, etc. It also allows users to write programs to process and convert graphics data to interact with other software or formats. This is useful for data import and export, format conversion and integration with external systems. You can also create customized user interface elements, such as dialog boxes, toolbar buttons, etc., to simplify user operations [6]. Therefore, this article is based on AutoCAD software and uses AutoLisp programming language to develop a customized pipeline diagram drawing system to improve the efficiency of pipeline diagram drawing.

II. INTRODUCTION TO AUTOLISP FUNCTIONS

As mentioned earlier, AutoLISP is a programming language developed specifically for AutoCAD software. It allows users to extend and automate the functions of AutoCAD by writing custom programs. Classified according to its usage function, its functions can be roughly divided into; mathematical operations, logical operations, conversion operations, serial processing, string and character processing, waiting for input, geometric operations, object processing, selection set and symbol table processing, query and control, judgments and loops, function processing and errors processing, display and printing control, element and expression processing...etc. The following is a brief explanation of several more commonly used functions [7].

- (1) Mathematical operation function: mainly used to handle simple mathematical operations. For example;
 - ◆ (+ <value> <value>); Returns the accumulated value
 - (+ 4 5) ; Return 9
 - ◆ (abs <value>); Returns the absolute value of the value
 - (abs -100) ; Return 100
- (2) Logical operation function: mainly used to process the results of logical operations. For example;
 - ◆ (/= <expression> < expression > ...); Return "T" or "nil"

- (/= 10 20); Return "T"
 - (/= "you" "you"); Return "nil"
- (3) Conversion operation function: mainly used to convert numerical values or strings into another data type. For example;
 - ◆ (atof <string>); Return "real number"
 - (atof "34.8") ; Return 34.8
 - ◆ (itoa <integer>); Return "string"
 - (atof 34.8) ; Return "34.8"
 - (4) Serial operation function: mainly used to process elements in the series. For example;
 - ◆ (car <list>); Returns the first element in the list
 - (car '(a b c)) ; Return A
 - ◆ (cadr <list>); Returns the second element in the list
 - (car '(1 2 3)) ; Return 2
 - (5) String and character processing functions: mainly used to process string or file elements. For example;
 - ◆ (strlen <string>); Returns the number of characters in the string
 - (strlen "abcde") ; Return 5
 - ◆ (strcat <string> <string>); Combine strings into one string
 - (strcat "a" "bout") ; Return "about"
 - (6) Waiting for input function: mainly used to capture the value input by the user. For example;
 - ◆ (getpoint <point> <prompt>); Return the data of this point
 - (setq p1 (getpoint "\n Please input first point : ")); Return p1 data
 - ◆ (getreal <prompt>); Returns the value of the real number
 - (setq ww (getreal "\n Please input length :")); Return ww value
 - (7) Geometric operation function: mainly used to process geometric relationships between graphics. For example;
 - ◆ (angle <point1> <point2>); Returns the angle value of 2 points
 - (angle '(2.0 1.0) '(4.5 10)); Return 1.29985
 - ◆ (distance <point1> <point2>); Returns the distance value of 2 points
 - (distance '(10 20) '(20 40)); Return 22.3607

III. SIMPLE DIAGRAM OF PIPELINES AND RELATED COMPONENTS

According to the National Standard of the Republic of China (CNS3), the commonly used pipeline drawing representation symbols are as follows;

(1) General pipeline representation

Items	Line diagram
Main process line	
Broken pipe	
Pipe end view	
Pipe section	
Pipeline flow direction	

Figure 1 General pipeline representation

(2) Pipeline jointed representation

Items	Line diagram
Threaded joint	
Welded joint	
flange joint	
plug-and-socket joint	

Figure 2 Pipeline jointed representation

(3) Commonly used welded pipe accessories

90° elbow joint	
tee joint	
Concentric reducer	
pipe cap	

Figure 3 Commonly used welded pipe accessories

(4) Commonly used flange pipe accessories

Items	Line diagram
90° elbow joint	
flange	
Slide into weld flange	

Weld neck flange	
blind flange	
Gate valve	
Check valve	

Figure 4 Commonly used flange pipe accessories

IV. PRINCIPLE OF SYSTEM ESTABLISHMENT

This article is based on AutoCAD software and uses AutoLisp programming language to develop a customized pipeline diagram drawing system to improve the efficiency of pipeline diagram drawing. Below are several examples to illustrate the principles of their establishment.

(1) Welded pipe accessories -90° elbow joint

- Step 1: input the pipe diameter.
- Step 2: input the "joint corner" (p1), "connector direction" (p2, p3), "main process line" (s1, s2).
- Step 3: calculating the welded points (s3, s4).
- Step 4: use the "fillet" command, to create the arc of main process line.
- Step 5: use the "donut" command, to create the welded points.

The execution results are shown in Figure 5.

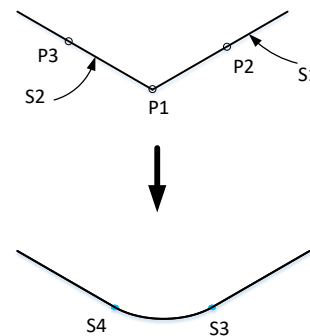


Figure 5 Welded pipe accessories -90° elbow joint

The AutoLisp program of welded pipe accessories - 90° elbow joint are shown in Figure6;

```
(defun C:ELB1 ()
  (COMMAND "-LAYER" "S" "PIP5" "")
  (setvar "osmode" 4279)
  (setq slayer (getvar "clayer"))
  (setq osm (getvar "osmode"))

  (progn
    (setq d1 (getdist "\nPipe diameter:"))
    (setq rad (* d1 ed))
    (setq p1 (getpoint "\nJoint corner: "))
    (setq p2 (getpoint p1 "\nConnector direction 1: "))
    (setq p3 (getpoint p1 "\nConnector direction 2: "))
    (setq s1 (entsel "\nSelect pipeline 1: "))
    (setq s2 (entsel "\nSelect pipeline 2: "))
    (setq ang (angle p1 p2))
    (setq ang1 (angle p1 p3))
    (setq S3 (polar p1 ang rad))
    (setq S4 (polar p1 ang1 rad))

    (setvar "osmode" 20663)
    (command "Fillet" "R" rad)
    (command "Fillet" s1 s2)
    (COMMAND "-LAYER" "S" "PIP140" "")
    (command "donut" 0 gap s3 s4)

    (command "color" "blylayer")
    (command "linetype" "s" "blylayer" "")
    (SETVAR "clayer" SLAYER)
    (setvar "osmode" osm)
  )
)
```

Figure 6 the AutoLisp program of -90° elbow joint

(2) Flange pipe accessories -90° elbow joint

- Step 1: input the pipe diameter.
- Step 2: input the “joint corner” (p1), “connector direction” (p2, p3), “main process line” (s1, s2).
- Step 3: calculating the welded points (s3, s4).
- Step 4: calculating the welded lines (L1,L2).
- Step 5: calculating the flange lines (L11,L22).
- Step 6: use the “fillet” command, to create the arc of main process line.
- Step 7: use the “break” command, to break main process line.
- Step 8: use the “line” command, to create welded and flange lines.
- Step 9: use the “donut” command, to create the welded points.

The execution results are shown in Figure 7.

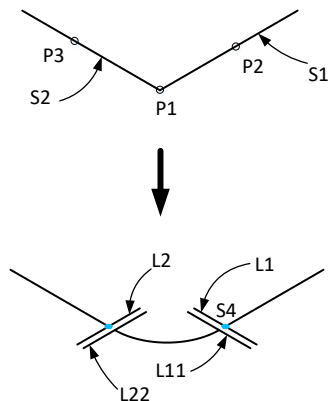


Figure 7 Flange pipe accessories -90° elbow joint

The AutoLisp program of flange pipe accessories -90° elbow joint are shown in Figure8;

```
(defun C:ELB1A ()
  (COMMAND "-LAYER" "S" "PIP5" "")
  (setvar "osmode" 4279)
  (setq slayer (getvar "clayer"))
  (setq osm (getvar "osmode"))

  (progn
    (setq d1 (getdist "\n Pipe diameter:"))
    (setq rad (* d1 ed))
    (setq p1 (getpoint "\n90 Joint corner: "))
    (setq p2 (getpoint p1 "\n Connector direction 1: "))
    (setq p3 (getpoint p1 "\n Connector direction 2: "))
    (setq s1 (entsel "\n Select pipeline 1: "))
    (setq s2 (entsel "\n Select pipeline 2: "))
    (setq ang (angle p1 p2))
    (setq ang1 (angle p1 p3))
    (setq S3 (polar p1 ang rad))
    (setq s3a (polar s3 (+ ang (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq s3b (polar s3 (- ang (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq S4 (polar p1 ang1 rad))
    (setq s4a (polar s4 (+ ang1 (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq s4b (polar s4 (- ang1 (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq S33 (polar s3 ang fgap))
    (setq s33a (polar s33 (+ ang (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq s33b (polar s33 (- ang (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq S44 (polar s4 ang1 fgap))
    (setq s44a (polar s44 (+ ang1 (/ pi 2.)) (/ (* d1 fd)2.)))
    (setq s44b (polar s44 (- ang1 (/ pi 2.)) (/ (* d1 fd)2.)))
    (setvar "osmode" 20663)
    (command "Fillet" "R" rad)
    (command "Fillet" s1 s2)
    (command "break" s33 "F" s33 s3)
    (command "break" s44 "F" s44 s4)
    (command "line" s3a s3b "")
    (command "line" s4a s4b "")
    (command "line" s33a s33b "")
    (command "line" s44a s44b "")
    (COMMAND "-LAYER" "S" "PIP140" "")
    (command "donut" 0 gap s33 s44)

    (command "color" "blylayer")
    (command "linetype" "s" "blylayer" "")
    (SETVAR "clayer" SLAYER)
    (setvar "osmode" osm)
  )
)
```

Figure 8 the AutoLisp program of -90° elbow joint

(3) Flange pipe accessories - Weld neck flange

- Step 1: set same diameter or reducing diameter.
- Step 2: input the pipe diameter.
- Step 3: input the “Flange center position” (C).
- Step 4: input the “neck direction” (D).
- Step 5: calculating the welded points (w1, w2).
- Step 6: calculating the flange lines (L1, L2).
- Step 7: calculating the neck lines (L11, L12, L21, L22).
- Step 8: use the “break” command, to break main process line.
- Step 9: use the “line” command, to create flange and neck lines.
- Step 10: use the “donut” command, to create the welded points.

The execution results are shown in Figure 9.

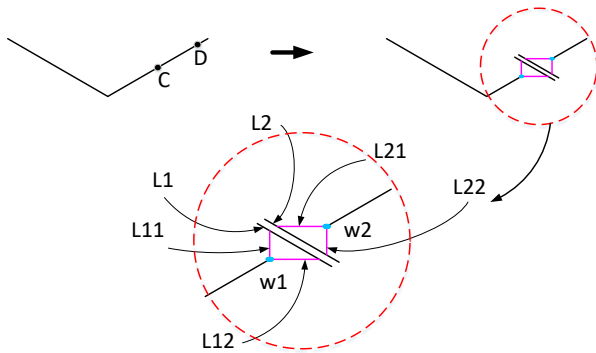


Figure 9 Flange pipe accessories - weld neck flange

The part of AutoLisp program of flange pipe accessories –weld neck flange are shown in Figure10;

```
(defun C:FLG2 ()
  (COMMAND "-LAYER" "S" "PIP5" "")
  (setvar "osmode" 4279)
  (setq slayer (getvar "clayer"))
  (setq osm (getvar "osmode"))

  (setq N (getstring "\n(1)Same diameter/(2)Reducing diameter:"))

  (cond ((= N "1")
    (setq d1 (getdist "\nPipe diameter:"))
    (setq p1 (getpoint "\nFlange center position:"))
    (setq p2 (getpoint p1 "\neck direction:"))

    (setq ww1 (/ d1 2.))
    (setq ww2 (+ (/ d1 2.) (/ d1 4.)))
    (setq ang (angle p1 p2))
    (setq ptm (polar p1 ang (+ ww1 (/ fgap 2.))))
    (setq ptm1 (polar p1 (+ ang pi) (+ ww1 (/ fgap 2.))))
    (setq p3 (polar p1 ang (/ fgap 2.)))
    (setq p4 (polar p1 (+ ang pi) (/ fgap 2.)))
    (setq ptm2 (polar p3 (+ ang (/ pi 2.)) ww1))
    (setq ptm3 (polar p3 (- ang (/ pi 2.)) ww1))
    (setq ptm22 (polar p4 (+ ang (/ pi 2.)) ww1))
    (setq ptm33 (polar p4 (- ang (/ pi 2.)) ww1))
    (setq ptm4 (polar p3 (+ ang (/ pi 2.)) ww2))
    (setq ptm5 (polar p3 (- ang (/ pi 2.)) ww2))
    (setq ptm44 (polar p4 (+ ang (/ pi 2.)) ww2))
    (setq ptm55 (polar p4 (- ang (/ pi 2.)) ww2))

    (progn
      (setvar "osmode" 20663)
      (COMMAND "-LAYER" "S" "PIP5" "")
      (COMMAND "break" P1 "F" ptm ptm1)
      (command "LINE" ptm4 ptm5 "")
      (command "LINE" ptm44 ptm55 "")
      (COMMAND "-LAYER" "S" "PIP3" "")
      (command "LINE" ptm ptm2 "")
      (command "LINE" ptm ptm3 "")
      (command "LINE" ptm1 ptm22 "")
      (command "LINE" ptm1 ptm33 "")
      (COMMAND "-LAYER" "S" "PIP140" "")
      (command "donut" 0 gap ptm ptm1 "")
    )
  )
)
```

Get the needed data

Calculating the needed value

Break the main process lines

create the flange and neck lines

create the welded points

Figure 10 part of the AutoLisp program of flange pipe accessories

(4) Flange pipe accessories – Check valve

- Step 1: input the pipe diameter.
- Step 2: input the “check valve position” (C).
- Step 3: input the “Pipeline flow direction” (D).
- Step 5: calculating the welded points (w1, w2).
- Step 6: calculating the flange lines (L1, L2).

- Step 7: calculating the neck lines (L11, L12, L21, L22).
- Step 8: calculating the check lines (L3, L4, L5).
- Step 9: calculating the flow direction lines (L6).
- Step 8: use the “break” command, to break main process line.
- Step 8: use the “line” command, to create flange and neck lines.
- Step 9: use the “donut” command, to create the welded points.

The execution results are shown in Figure 11.

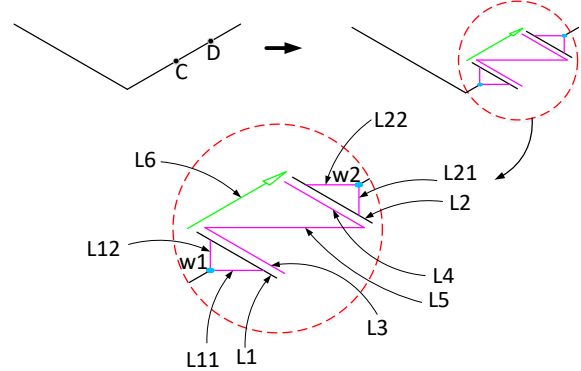


Figure 11 Flange pipe accessories – check valve

The part of AutoLisp program of flange pipe accessories –check valve are shown in Figure12;

```
(defun C:VALSA ()
  (COMMAND "-LAYER" "S" "PIP5" "")
  (setvar "osmode" 4279)
  (setq slayer (getvar "clayer"))
  (setq osm (getvar "osmode"))
  (setq d1 (getdist "\nPipe diameter:"))
  (setq flg (+ d1 (/ d1 2.)))
  (setq p1 (getpoint "\nvalve position:"))
  (setq p2 (getpoint p1 "\nPipeline flow direction:"))
  (setq tt (/ d1 4.))
  (setq ww1 (/ d1 2.))
  (setq ww2 (+ (/ d1 2.) tt))
  (setq ang (angle p1 p2))
  (setq ang1 (- ang (/ pi 2.)))
  (setq ang2 (+ ang (/ pi 2.)))
  (setq ang3 (+ ang pi))
  (setq ptr (polar p1 ang (/ flg 2.)))
  (setq ptr1 (polar ptr ang fgap))

  .
  .
  .

  (setq op (polar p1 ang2 (+ (/ flg 2.) fgap)))
  (setq op1 (polar op ang (+ (/ flg 2.) (* fgap 1.2))))
  (setq op2 (polar op ang3 (+ (/ flg 2.) (* fgap 1.2))))

  (progn
    (setvar "osmode" 20663)
    (COMMAND "-LAYER" "S" "PIP5" "")
    (COMMAND "break" p1 "F" ptr1 ptl11)
    (command "LINE" ptrf1 ptrf2 "")
    (command "LINE" ptlf1 ptlf2 "")
    (COMMAND "-LAYER" "S" "PIP3" "")
    (command "LINE" ptrv1 ptrv2 ptlv1 ptlv2 "")
    (command "line" ptr11 ptm2 "")
    (command "line" ptr11 ptm3 "")
    (command "line" ptl11 ptm22 "")
    (command "line" ptl11 ptm33 "")
    (COMMAND "-LAYER" "S" "PIP1" "")
    (command "QLEADER" op1 op2 \\\\)
    (COMMAND "-LAYER" "S" "PIP140" "")
    (command "donut" 0 gap ptr11 ptl11 "")
    (command "point" ptr11 "point" ptl11)
  )

  (command "color" "blyayer")
  (command "linetype" "s" "blyayer" "")
  (SETVAR "clayer" SLAYER)
  (setvar "osmode" osm)
)
```

Get the needed data

Calculating the needed value

create the flange, check and flow direction lines

Figure 12 part of the AutoLisp program of check valve accessories

(5) Flange pipe accessories – Gate valve

- Step 1: input the pipe diameter.
- Step 2: input the “gate valve position” (C).
- Step 3: input the “switch direction” (D).
- Step 5: calculating the welded points (w1, w2).
- Step 6: calculating the flange lines (L1, L2).
- Step 7: calculating the gate lines (G1, G2, G3, G4).
- Step 8: calculating the switch lines (S1, S2, S3).
- Step 9: calculating the handle lines (H1, H2).
- Step 8: use the “break” command, to break main process line.
- Step 8: use the “line” command, to create flange, gate, switch and handle lines.
- Step 9: use the “donut” command, to create the welded points.

The execution results are shown in Figure 13.

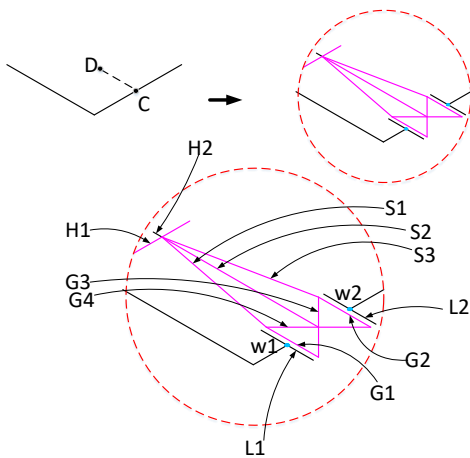


Figure 13 Flange pipe accessories – gate valve

The part of AutoLisp program of flange pipe accessories –gate valve are shown in Figure14;

```
(defun C:VAL2 ())
```

```
(COMMAND "-LAYER" "S" "PIP5" "")
(setvar "osmode" 4279)
(setq slayer (getvar "clayer"))
(setq osm (getvar "osmode"))
```

```
(setq d1 (getdist "\nPipe diameter:"))
(setq flg (+ d1 (/ d1 2.)))
(setq tt (/ d1 4.))
(setq p1 (getpoint "\nvalve position:"))
(setq p2 (getpoint p1 "\nswitch direction:"))
(setq ww1 (/ d1 2.))
(setq ang (angle p1 p2))
(setq ang1 (- ang (/ pi 2.)))
(setq ang2 (+ ang (/ pi 2.)))
(setq ang3 (+ ang pi))
(setq ptr (polar p1 ang1 (/ flg 2.)))
(setq ptr1 (polar ptr ang1 fgap))
(setq ptrv1 (polar ptr ang (/ flg 2.)))
(setq ptrv2 (polar ptr ang3 (/ flg 2.)))
```

Get the needed data

Calculating the needed value

```
(progn
  (setvar "osmode" 20663)
  (COMMAND "-LAYER" "S" "PIP5" "")
  (COMMAND "break" p1 "F" ptr1 ptl1)
  (command "LINE" ptrf1 ptrf2 "")
  (command "LINE" ptlf1 ptlf2 "")
  (command "line" OP opT "")
  (COMMAND "-LAYER" "S" "PIP3" "")
  (command "LINE" ptrv1 ptrv2 ptlv1 ptlv2 ptrv1 "")
  (command "line" p1 op "")
  (command "line" ptrv1 op ptlv1 "")
  (command "line" OP1 op2 "")
  (COMMAND "-LAYER" "S" "PIP140" "")
  (command "donut" 0 gap ptr1 ptl1 "")
  (command "point" ptr1 "point" ptl1)
)
(command "color" "bylayer")
(command "linetype" "s" "bylayer" "")
(SETVAR "clayer" SLAYER)
(setvar "osmode" osm)
)
```

create the flange, gate, switch and handle lines

Figure 14 part of the AutoLisp program of gate valve accessories

V. SYSTEM OPERATION STEPS

In this article is based on AutoCAD software and uses AutoLisp programming language to develop a customized pipeline diagram drawing system to improve the efficiency of pipeline diagram drawing. The following uses Figure 15 as an example to illustrate the execution steps of this system.

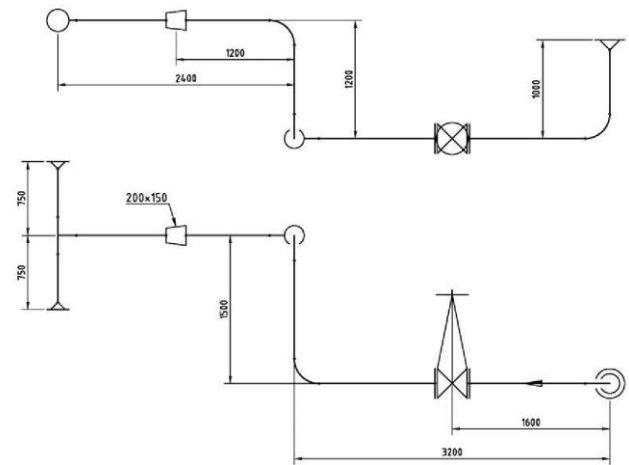


Figure 15 The example of piping diagram

Step 1: Draw main process lines, as shown in Figure 16.

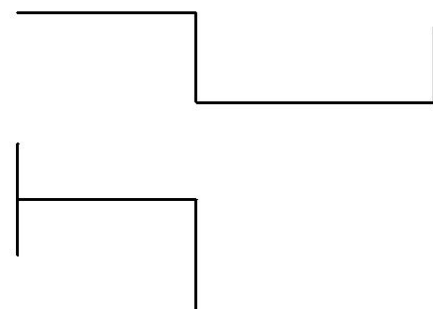


Figure 16 main process lines

Step 2: Set cutting plane, as shown in Figure 17.

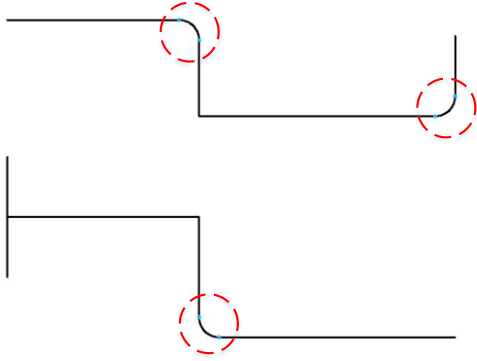


Figure 17 90° elbow joint

Step 3: Setting tee joint, as shown in Figure 18.

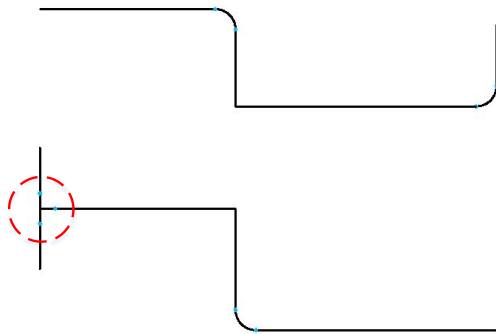


Figure 18 tee joint

Step 4: Setting weld neck flange. The results are shown in Figure 19.

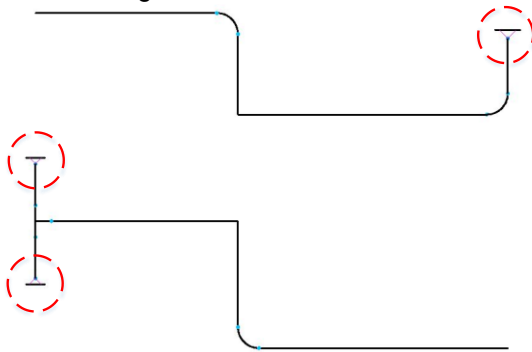


Figure 19 weld neck flange

Step 5: Setting concentric reducer. The results are shown in Figure 20.

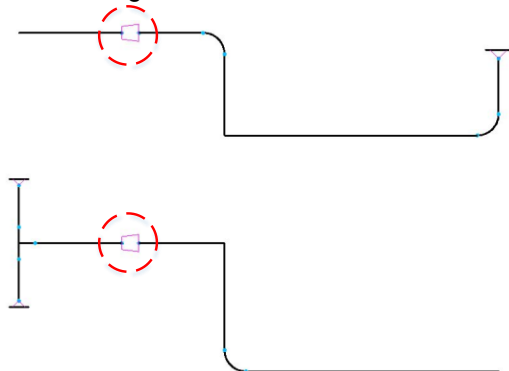


Figure 20 concentric reducer

Step 6: Setting the gate valve, as shown in Figure 21.

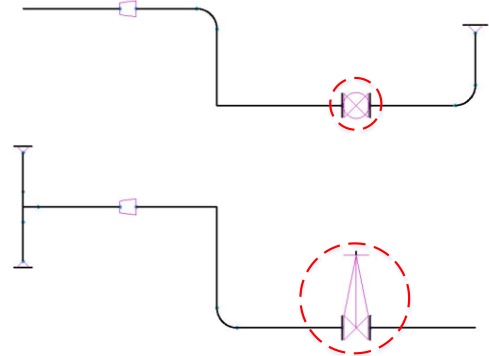


Figure 21 gate valve

Step 7: Setting Elbow end view and flange end view, as shown in Figure 22.

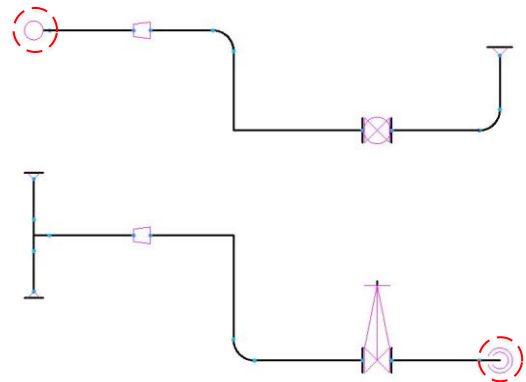


Figure 22 The Elbow end view

Step 8: Setting Elbow turning symbol, as shown in Figure 23.

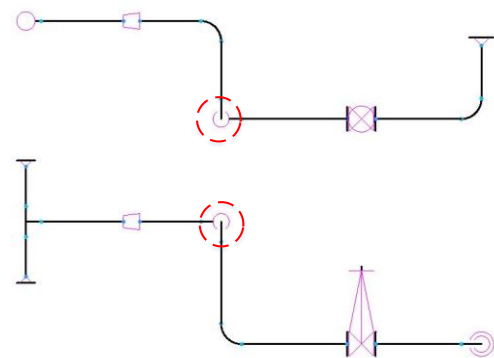


Figure 23 Elbow turning symbol

Step 9: Setting Pipeline flow direction symbol, as shown in Figure 24.

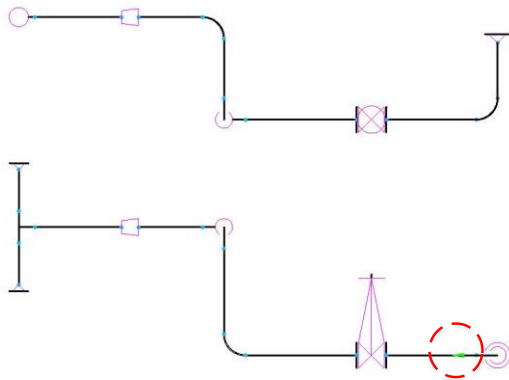


Figure 24 Pipeline flow direction symbol

It can be seen from the above execution steps that with the assistance of this system, during the pipeline drawing process, only simple settings are needed to completely draw the pipeline diagram we need. It can indeed improve the efficiency of pipeline drawing.

VI. CONCLUSION

A piping diagram is an engineering drawing used to show and describe the layout and connection relationships of pipes, pipelines, valves, instruments and other components in a system. It not only provides detailed layout and connection information of the piping system, but also serves as the basis for optimal design analysis of the entire piping system. Therefore, this article is based on AutoCAD software and uses the AutoLisp programming language to develop a customized pipeline diagram drawing system. Through actual execution, it can indeed improve the efficiency of pipeline diagram drawing.

REFERENCES

1. <https://www.autodesk.com.tw/>
2. https://www.utitech.com.tw/Product_AVEVA.html
3. <https://baike.baidu.hk/item/smartplant%20d/517891>
4. <https://www.solidworks.com/support/training/solidworks-routing-piping-and-tubing>
5. <https://www.autodesk.com/solutions/pid-software>
6. <https://zh.wikipedia.org/zh-tw/AutoLISP>
7. Wu Yon gjin and Lin Mei ying, AutoCAD Programming Magic Secrets, GOTOP INFORMATION INC., 2008.