

Development of Test-Control System of the Hydraulic Experiment

Peichuan Gao
Nanjing Institute of Technology
Nanjing ,China

Abstract—This article is about the principles of hydraulic experiments measurement and control simulations under Windows interface and the ways to realize them. Using C# and PLC , you can monitor, measure, output data and eventually finish your experiments. All your experiments are flexible and respond quickly and accurately.

Keywords-hydraulic experiments; measurement and control simulations

I. INTRODUCTION

Hydraulic drive is an important way to realize automation in modern machinery and equipment. It is a basic curriculum for applied technology. It requires not only listening in class, but also doing experiments after class. These can exercise students' skill of operating hydraulic equipments.

Many traditional hydraulic experiment tables use relay control. In these experiments, Students not only need to press the button for reverse-directional electromagnetic valves and adjust hydraulic valve, at the same time , they should record pressure and flow rate. The experimental data is manually processed, then we plot curves, and conclude the experimental results, the experiment data accuracy is not high, the speed is slow. It is low efficiency, every time you spend too much time finishing an experiment.

The experiments introduced in this article, however, are much easier and have high accuracy. Students need to program the PLC and fix the hydraulic flow circuit. When all preparation work is done, it is a computer's job to record, process, storage and finish the curve plotting.

This experiment system, uses the sensor to acquire pressure, flow rate and stroke, then we store these data in the PLC storage unit. When we need these data, The PC can communicate with PLC through serial port when we need these data. The PC reads the data from the PLC unit and shows it in software interface. At the same time, it can deal with the data, have the curve plotted and original data saved and eventually generate the experiment report and print preview. It can fundamentally enhance experiment table function, improve the real-time performance of the data acquisition and data processing accuracy, and improve flexibility of the system, make the operation easier.

II. HYDRAULIC EXPERIMENTS MEASUREMENT AND CONTROL SIMULATIONS

Hydraulic measurement and control simulation experiment design can be divided into two parts, one is to test and control, the other part being simulation playing. These can manifest the superiority of computer aided teaching, make experiments more accurate, fast and vivid.

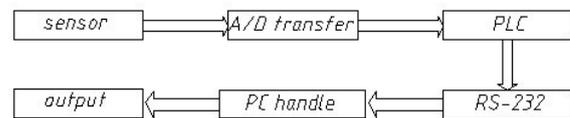


Figure 1 data flow diagram

1)The test part is to use sensors (pressure sensor, displacement sensor, flow sensor) to provide analog current and voltage parameters. We should transfer analog signal into digital signal. The signal can be passed to the computer via RS-232 communication interface, the C# MsComm control in the computer can call the signal, and then change it into figure, animation or number and curve shown in the screen. At the same time we can also plot figure of the theoretical simulation so we can compare and find the differences between them.

2)First, open the software and you can see the main form.



Figure 2 System main form diagram

Second, you can find the basic information about the experiment. Such as the principle diagram, the experiment content and steps.

This hydraulic experiment measurement and control experiment system using PC and PLC control, combined with data acquisition, serial communication technology, to realize the automation of hydraulic experimental control, data acquisition, data processing, real-time display. It is a very practical experimental measurement and control system.

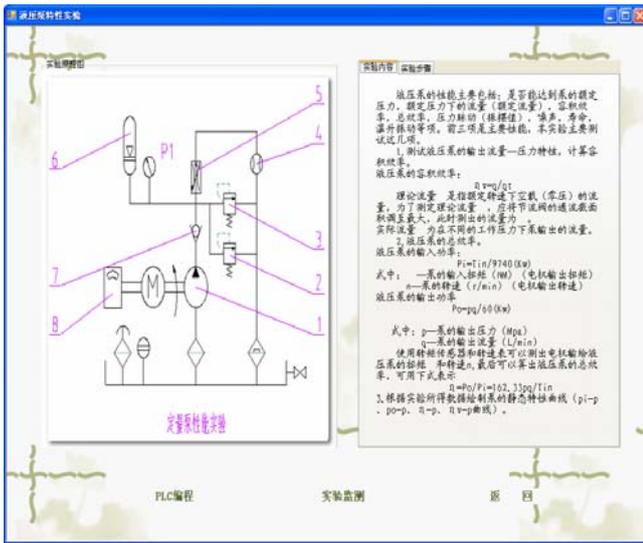


Figure 3 basic information preview diagram

The third is the insert of software. In the hydraulic experiment we need to use PLC to realize the motion control of electromagnetic valve, so we just have the PLC program software direct embedded in this system software, students in the course can experiment directly through the software to open S7-200 programming software operation, and don't have to look for its installation position to open it. This makes the experiment process more convenient and orderly.

Each experiment needs to use PLC programming control, so we use a function to call it. Here is the program:

public static void PLC()//function:call plc program software.It is wrote in the main form,so other forms can call it.

```

{ System.Diagnostics.Process p = new
  System.Diagnostics.Process();

  try
  { p = System.Diagnostics.Process.Start("microwin.exe");
    p.WaitForExit();
    if (p.ExitCode == 0)
    if (p.ExitCode == -1)
      System.Console.WriteLine("operation over");
  }
  catch (Exception ex)
  {
    MessageBox.Show("operation failed" + "failure
    reasons" + ex.ToString());
  }
  finally
  {
    p.Close();
  }
}

```

}
 }
 Click "PLC program" button in basic information diagram. In mouseclick event type code "MainForm.PLC ();"to call the PLC program software. According to the analysis of the I/O points, we choose S7-200 series PLC, the interface is as followed:

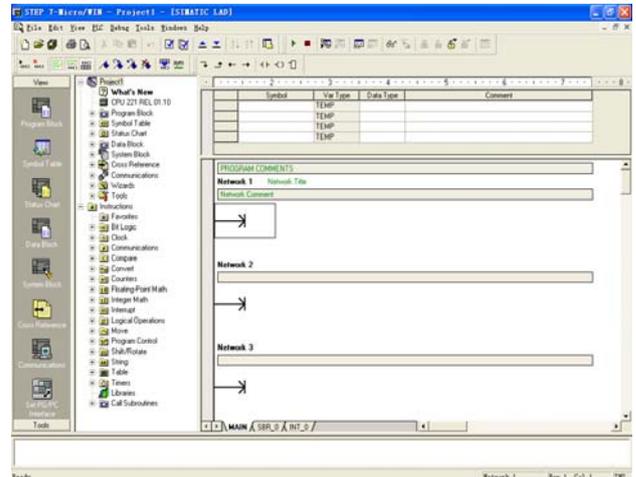


Figure 4 PLC program interface

The fourth is to plot curve. To automatically generate graph from the attained data, we need to connect with database, then fill the data into database and bound Chart control. So, when there is data coming, the Chart control will display a point and join them into a curve. We can realize real-time curve plotting. We can set the specific style of the Chart control curve in the RDLC.

(1) Static curve plotting

Static curve plotting, namely read a set of data, draw a point and together with the previous point to make up of a curve. In hydraulic fixed displacement pump characteristic experiment, the code to draw a curve is as followed:

```

string conn = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=db1.mdb";

OleDbConnection OleDbConn1 = new
  OleDbConnection(conn);

OleDbConn1.Open();

string str = "INSERT INTO
sy2_yyb(ID,p,q,Po,Pi,Eff,Effv)VALUES('"+ Idx + "','"+
P[Idx].ToString("#0.00") + "','"+ Q[Idx].ToString("#0.00")
+ "','"+ pOUT[Idx].ToString("#0.00") + "','"+
pIN[Idx].ToString("#0.00") + "','"+ Eff[Idx].ToString("#0.00")
+ "','"+ Effv[Idx].ToString("#0.00") + "')";

OleDbCommand cmd = new OleDbCommand(str,
  OleDbConn1);

cmd.ExecuteNonQuery();

```

```

OleDbDataAdapter mydataadapter = new
    OleDbDataAdapter();
DataSet mydataset = new DataSet();
string com = "SELECT*FROM sy2_yyb Order By p
ASC";

mydataadapter.SelectCommand = new
OleDbCommand(com, OleDbConn1);

OleDbCommandBuilder mycb = new
OleDbCommandBuilder(mydataadapter);
mydataadapter.Fill(mydataset, "sy2_yyb");
mydataadapter.Update(mydataset, "sy2_yyb");
reportViewer1.LocalReport.DataSources.Clear();
reportViewer2.LocalReport.DataSources.Clear();

reportViewer1.LocalReport.DataSources.Add(new
Microsoft.Reporting.WinForms.ReportDataSource("DataSet1
_sy2_yyb", mydataset.Tables["sy2_yyb"]));

reportViewer2.LocalReport.DataSources.Add(new
Microsoft.Reporting.WinForms.ReportDataSource("DataSet1
_sy2_yyb", mydataset.Tables["sy2_yyb"]));

this.reportViewer1.RefreshReport();
this.reportViewer2.RefreshReport();
OleDbConn1.Close();

```



Figure 5 hydraulic fixed displacement pump characteristic experiment static curve

Next is to dynamic draw. click the “Dynamic Data Acquisition” button to read a set of data and generate a curve. Dynamically reading need to change the timer frequency ,the chosen timer frequency can change the way of the reading. To read according to the particular timer frequency ,we need to input the code in Click event:

```

private void dcj_Click(object sender, EventArgs e)
{
timer1.Enabled = true;
switch (listBox1.SelectedIndex)

```

```

{
case 0: timer1.Interval = 15; break;
case 1: timer1.Interval = 20; break;
case 2: timer1.Interval = 25; break;
case 3: timer1.Interval = 30; break;
case 4: timer1.Interval = 40; break;
case 5: timer1.Interval = 50; break;
case 6: timer1.Interval = 100; break;
default: timer1.Interval = 20; break;
}
}

```

In Tick event under timer ,write the code to show the curve:

```

private void timer1_Tick(object sender, EventArgs e)
{
if (Math.Abs(yali[num] - ptemp) > 0.01)
{
ptemp = yali[num];
string conn = @"Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=db1.mdb";

```

```

OleDbConnection OleDbConn1 = new
OleDbConnection(conn);

```

```

OleDbConn1.Open();

```

```

string str = "INSERT INTO sy3_dt(ID,p,t)VALUES(" +
num
+ "," + yali[num] + "," + num + ")";

```

```

OleDbCommand cmd = new OleDbCommand(str,
OleDbConn1);

```

```

cmd.ExecuteNonQuery();

```

```

OleDbDataAdapter mydataadapter = new
OleDbDataAdapter();

```

```

DataSet mydataset = new DataSet();

```

```

string com = "SELECT*FROM sy3_dt Order By t ASC";

```

```

mydataadapter.SelectCommand = new
OleDbCommand(com, OleDbConn1);

```

```

mydataadapter.Fill(mydataset, "sy3_dt");

```

```

mydataadapter.Update(mydataset, "sy3_dt");

```

```

reportViewer2.LocalReport.DataSources.Clear();
reportViewer2.LocalReport.DataSources.Add(new

```

```

Microsoft.Reporting.WinForms.ReportDataSource("DataS
et1_sy3_dt",mydataset.Tables["sy3_dt"]));

```

```

this.reportViewer2.RefreshReport();

```

```

OleDbConn1.Close();

```

```

num = num + 1;
}
else
{
timer1.Enabled = false;
}
}。

```

This system has a friendly monitoring interface, can accurately accept the data (pressure, flow rate, power) from machine , and display in the monitoring interface, at the same time dynamically drawing up the curve. This system is all round, convenient to operation and test, adapted to the general universities in hydraulic transmission experiment requirements.

REFERENCES

- [1] Bangjie Lin. c# Program design. Beijing: China Railway Press, 2005
- [2] Yao Tang . c# Program design teaching. Beijing: China Waterpower Press, 2005
- [3] Pingge Zhang. Hydraulic drive and control. Beijing: Metallurgical Industry Press, 2004
- [4] Hejlsberg Anders . The C# Programming Language(Second Edition). Addison Wesley, 2006
- [5] PeterXumark.Beginning C# From Concept To Object



Figure 6 relief valve dynamic property curve diagram