Research on Control Strategy of Hydraulic Control Unit Test Device Control System

Feng Qian*, Yongyi He* and Pengfei Cheng
School of Mechatronic Engineer and Automation, Shanghai University, Shanghai, China

*Corresponding author e-mail: qianfeng11@shu.edu.cn, *yongyi.he@shkelai.com

Abstract. Hydraulic control unit test (HCU) is a complicated process with nonlinear, hysteresis and large parameter change. This paper focuses on the research of PID control strategy which comes from the BP neural network and HCU test strategy, and combines Matlab/Simulink, AMESim and other software for simulation and verification, in order to design a reliable and high-quality control scheme.

1. Introduction
The main functions of a hydraulic control unit on the vehicle are to drive wet dual clutch combination and separation, control seven forward and one backward gears, and regulate the lubrication and cooling flow of gear shaft and clutch.

The schematic diagram of HCU's internal hydraulic pressure is shown in figure 1. below.

![Figure 1. HCU hydraulic schematic diagram](image-url)
It shows that the whole HCU 10 solenoid valves, including one main oil line pressure control valve, two clutch pressure control valves, four gear shift flow control valves, two gear shift pressure control valves and one lubrication flow control valve. The operating characteristics of the ten solenoid valves play a decisive role in determining whether the HCU can operate reliably and stably in the expected mileage life.

According to the requirements of testing, the design hydraulic schematic diagram is shown in FIG. 2. The hydraulic system clock contains a temperature control unit that controls the temperature of hydraulic oil in the pipeline at 40℃. At the bottom is the main oil tank, whose main function is to store hydraulic oil. Because some test items need to soak HCU in hydraulic oil, another test tank is set in the oil path. During the test, HCU is installed in the interior, fixed and sealed on one side wall.

2. Conventional PID control

PID control undergoes a great long history and is one of the earliest control strategies developed [1] cars only. It has become the most popular control method in agricultural production and industrial production because of its good control effect, high reliability, good robustness, no need to establish mathematical model, convenient adjustment and other advantages.

2.1. Introduction to conventional PID control

PID controller is a linear tool. The whole process is that the system value r (t) and the actual output value y (t) input from control deviation e (t), go through the deviation ratio of unit (P), integral unit (D) (I) and differential unit, by linear combination going into the controller output u (t), and finally to control the to-be-controlled object, so PID is short for proportion, integral and differential parts, as shown in Figure 3.
Figure 3. PID controller schematic diagram

Its general control law is:

\[ u(t) = k_p \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \tag{1} \]

In the above equation, \( k_p \) -- proportionality coefficient, \( T_i \) -- integral time constant, \( T_d \) -- differential time constant.

Generally, the effect analysis of each calibration link in the PID controller is shown below:

1. Proportional regulator
   The differential equation of the proportional regulator is:
   \[ u(t) = k_p e(t) \tag{2} \]

   From equations (2), it can be seen that the output and input deviation of the proportional regulator is proportional, and if the proportional coefficient \( k_p \) becomes larger, the response speed of the system will be faster. In case one deviation appears, the tool can timely generate the adjustment function in proportion to it, so as to reduce the deviation and have the characteristics of timely adjustment. But the large \( k_p \) scale coefficient will cause the control system to oscillate and even make the control system end up unstable.

2. Integral regulator
   Integral regulator focus on decreasing and eliminating static fault and increase the preciseness of control system. The integral formula is:
   \[ y(t) = \frac{1}{T_i} \int e(t) dt \tag{3} \]

   \( T_i \) is the integral time constant, which decides to what extent the integral adjustment shall be applied on the system. When the integral \( T_i \) becomes smaller, the integral function is more enhanced, and when the integral function \( T_i \) becomes larger, the integral function becomes weaker. As can be seen from equation (3), if the deviation exists even it is quite small, it would accumulate gradually and generate corresponding control quantity until there no deviation occurs to bother, with the system achieving a perfect stable state. Integral adjustment is usually not applied independently, but combined with proportional adjustment.

3. Differential regulator
   Can reflect the change rate of deviation signal. When it is enough big, an influential correction signal will go into the whole system to make the deviation disappear in the bud. The differential formula is:
From equations (4), it can be seen that the differential component will have a control effect on any variation of the deviation, so as to adjust the final value of the control system and prevent the deviation from changing. If the deviation varies faster, the output of the differential control and the blocking effect become greater. Increasing the differential time constant can reduce the operation time of the whole system and reduce the overshoot, but it would also be more sensitive to disturbance and have weaker anti-interference ability. In the control and adjustment process in the real life, the selection of constant should change with the time constant of the object in control, so as to achieve better control effect.

\[ u(t) = T_d \frac{de(t)}{dt} \]  

(4)

2.2. Digital PID Control

Digital PID control has started being widely applied to the continuous system control [2] since it was found. Digital PID control provides better performance than continuous PID control since the flexibility of the computer program, it is not hard to overcome the problems in the continuous PID control system, through algorithm correction and get a more perfect PID control. Digital PID control, in a way, is able to be categorized into two parts: position PID control and incremental PID control. To discretize the differential and integral terms in equation (1-1), the sampling period Ts must be very short. The sampling time kTs represents time t. For the convenience of writing, kTs is simplified and expressed as k to gain the position PID control algorithm:

\[
u(k) = k_p e(k) + \sum_{j=0}^{k} e(j) + k_i e(k) - e(k-1)] + U_0
\]  

(6)

Type in the u0-- the base value of the control quantity, i.e., the control when k=0; U (k) -- computer output value at the KTH sampling moment; E (k) -- the input deviation value at the k sampling moment; E (k-1) -- the input deviation value at the k-1 sampling moment; 

\[ k_p \] -- proportional amplification factor;

\[ T_i \] -- integral amplification factor;

\[ T_d \] -- differential amplification factor.

In the position-type PID control, the controller outputs \( u(k) \) to directly control the actuator, and \( u(k) \) corresponds to the actuator's position one by one. Incremental PID control means what the controller outputs is the newly added part of the control quantity. As the actuator (for example, stepper motor) needs the increment of the control quantity, incremental PID control shall be utilized. According to the recursive principle:

\[
u(k-1) = k_p e(k-1) + k_i \sum_{j=0}^{k-1} e(j) + k_d [e(k-1) - e(k-2)] + U_0
\]  

(7)

By subtracting equation (7) from equation (6), the algorithm of incremental PID control is able to be obtained as the following equation:
\begin{equation}
  u(k) = u(k-1) + k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2))
\end{equation}

Compared with position PID control algorithm, the algorithm of incremental PID control does not need the accumulation of the deviation, and the determination of the increment of the control quantity just has relationship with the current deviation value and the deviation value of the first two times.

2.3. PID controller parameter setting

In PID control system, what vitally influences the overall system are three critical factors, namely integral time constant, proportional coefficient and differential time constant, none of which can be dispensed with. There are many literatures about PID parameter tuning. [3-6].

Since Ziegler and Nichols proposed the empirical formula method for PID controller parameter setting, many methods have been used for manual and automatic PID parameter setting. These parameter setting methods are categorized into two sections: the one of intelligent PID controller and that for conventional PID controller, as per the development stage. In accordance with the number of controlled objects, the parameter setting methods are able to categorized into the one of single-variable PID controller and that for multi-variable PID. Considering the combination of control variables, parameter setting methods are categorized into one of linear PID controller and one of nonlinear PID controller. Here is a brief introduction of Ziegler-Nichols parameter setting.

The method named Ziegler-Nichols is a PID parameter adjustment way on the basis of system stability analysis. Although the setting method is not complicated, the effect is quite good. The basic way to set parameters is to first set the integral \( k_i \) and differential coefficients \( k_d \) to 0, and the proportional coefficient \( k_p \) should be increased till the system starts to destabilize. Then multiply the proportionality coefficient \( k_m \) by 0.6, which is the proportionality coefficient \( \omega_m \) after setting. Other parameters can be calculated according to the following setting formula:

\begin{equation}
  k_p = 0.6 k_m
\end{equation}

\begin{equation}
  k_d = \frac{k_p \omega}{4 \omega_m}
\end{equation}

\begin{equation}
  k_i = \frac{k_p \omega_m}{\omega}
\end{equation}

2.4. Limitations of conventional PID controllers

To sum up, in the regard of principle, PID controller is not complex, easy to utilize, robust and can play its full role in industrial production sites with harsh scenario or situation. An integrate and comprehension design method and parameter setting plans of PID control algorithm has been formed and it is not hard for engineers or technical staff to absorb and apply. However, the conventional PID controller has no adaptive control ability, and the control effect is poor for objects with highly nonlinear, fast time-varying uncertainty, lag and other characteristics. In the process of conventional PID control, even if a set of parameters have good control effect in a small control range, we can say it is hard to ensure the stability and control quality of the controlled object when its characteristics change, so the control effect of this kind of complex object using conventional PID control is not very ideal.

In this paper, HCU test system is a complicated process with nonlinear, hysteresis and large parameter changes. In this paper, there are many kinds of pressure regulators, and the structure and characteristics of different kinds of pressure regulators are different, so to set the control parameters is
not easy. Therefore, it is required to use the corresponding intelligent algorithm together to address the problem of setting parameters in the process of pressure setting control.

3. Design of PID Controller on the Basis of BP Neural Network

Artificial neural network (Ann) includes great many neurons (processing units) which are widely connected with each other [7]. Through using neural network method with conventional PID control together, the designed control system will have stronger learning ability, adaptability and better robustness.

3.1. Neuron model

References are cited in the text just by square brackets [1]. Two or more references at a time may be put in one set of brackets [3, 4]. The references are to be numbered in the order in which they are cited in the text and are to be listed at the end of the contribution under heading references, see our example below.

McCulloCh and PittS firstly invented the MP model that is one of neural network model proposed in 1943 [8]. Figure 4 Shows that, it is a multi-input, multi-output nonlinear information processing unit.

*Figure 4. MP neuron model structure*

\[ y_i = f(\sum_{j=1}^{n} w_{ij} x_j - \theta_j) \quad (i \neq j) \]  

Set

\[ u_i = \sum_{j=1}^{n} w_{ij} x_j - \theta \]  

Then

\[ y_i = f(u_i) \]  

f(x) is an action function, also called an excitation function. MP neuron model is the basis of artificial neuron model and neural network theory.
3.2. **PID controller on the basis of BP neural network**

BP neural Network, now acknowledged as Back Propagation Network, is utilized across all fields and has quite good reputation. With a special feature of one-way propagation, it is defined as a multi-layer forward network. In BP network, there are two kinds of signals. One is a function signal, which is an input signal. It comes from the endpoint of the input layer of the network and propagates forward into the neurons of every level of the network. The other is the error signal, which is produced from the network’s output. Then, according to the way the error signal goes down, connection weights of each layer are modified through every output level and the concealed level, and then ultimately returning the original input layer. [9,10].

1. **BP neural network PID controller structure**

![Figure 5. BP neural network PID control system structure](image)

**Figure 5.** Shows that, the structure of PID control system on the basis of BP neural network is made up of BP neural network and traditional PID controller. The traditional PID controller directly controls the controlled object in a enclosed cycle, while the neural network adjusts the three parameters of the PID controller \( k_p, k_i, \) and \( k_d \). BP neural network finds the PID controller parameters under some desirable control law through continuous learning and adjustment of weighting coefficient.

2. **PID control algorithm on the basis of BP neural network** [11-14].

The classical incremental digital PID control equation could be lighten up as follows:

\[
\begin{align*}
    u(k) &= u(k-1) + k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2)) \\
    &= f(u(k-1), k_p, k_i, k_d, e(k), e(k-1), e(k-2)) \quad (15)
\end{align*}
\]

If \( k_p, k_i \) and \( k_d \) are considered as regulable coefficients dependent on the system operating status, equation (2-4) is able to be expressed like below:

\[
    u(k) = f(u(k-1), k_p, k_i, k_d, e(k), e(k-1), e(k-2)) \quad (16)
\]

In the above equation (16), \( f(.) \) is a nonlinear function related to..., u(k-1), e(k), etc. \( k_p, k_i \), and \( k_d \). BP neural network can be used to obtain a desirable control law through continuous learning.

The BP neural network designed in this academic essay adopts a BP neural network in three layer. **Figure 6.** shows its structure. It has M input layer nodes, Q hidden layer nodes and 3 output layer nodes. The complex of the system determines the input variables’ number. The three nodes of the output layer are \( k_p, k_i \), and \( k_d \), which are the PID controller’s three vital parameters.
Figure 6. BP neural network structure diagram

The input and output of the input layer node of BP neural network are:

\[
I_i^{(0)}(n) = x_i(n) \quad I = 1, 2, ..., M
\]

\[
O_i^{(0)}(n) = I_i^{(0)}(n)
\]

Where, \( O_i^{(0)}(n) \) is the output of the \( i \)th node on the input layer, and \( M \) means the number of input nodes.

The complexity of the controlled system determines the value of \( M \). \( O_i^{(0)}(n) \) In this paper, the upper corners of variables \((0), (1)\) and \((2)\) are used to symbol, separately, the input layer, hidden layer and output layer.

The input and output of hidden layer of neural network are:

\[
V_j^{(1)}(n) = \sum_{i=0}^{M} w_{ji}^{(1)}(n)O_i^{(0)}(n)
\]

\[
O_j^{(1)}(n) = \varphi(v_j^{(1)}(n)) \quad J = 1, 2, ..., Q
\]

Where there is the weight coefficient of concealed layer neurons; \( w_{ji}^{(1)} \), \( Q \) is the quantity of neurons in the concealed layer. \( \varphi(\cdot) \) is the activation function of the implicit neuron, select the hyperbolic tangent function (Sigmoid function):

\[
\varphi(x) = \tanh x = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}
\]

The following are the input and output of the network output level:

\[
V_k^{(2)}(n) = \sum_{j=0}^{Q} w_{kj}^{(2)}(n)O_j^{(1)}(n)
\]

\[
O_k^{(2)}(n) = f(v_k^{(2)}(n)) \quad K \text{ is equal to } 1, 2, 3
\]
And, \( k_r = O_i^{(1)}(n) \), \( k_i = O_i^{(2)}(n) \), \( k_d = O_i^{(3)}(n) \), that is, they separately represent the regulatable parameters of the PID controller. Since they are always positive, the activation function of the output layer selects the non-negative sigmoid function:

\[
f(x) = \frac{1}{2} \left( 1 + \tanh x \right) = \frac{\exp(x)}{\exp(x) + \exp(-x)}
\]

The expression of the performance index function of the system is as follows:

\[
\varepsilon(n) = \frac{1}{2} \left( r_{in} - y_{out} \right)^2 = \frac{1}{2} e^2(n)
\]

BP neural network back-propagation algorithm adjusts the weighting coefficient with the learning rule of error correction: that is, it modifies along the negative gradient direction of the weight coefficient, and adds an inertial term that makes the search converge rapidly and globally minimal.

\[
\Delta w_{ij}^{(2)}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ij}^{(2)}(n)} + \alpha \Delta w_{ij}^{(2)}(n-1) = \delta_k^{(2)}(n)O_i^{(1)}(n) + \alpha \Delta w_{ij}^{(2)}(n-1)
\]

(2-12), \( \eta \) for vector; Alpha is the inertia factor.

\[
\frac{\partial \varepsilon(n)}{\partial w_{ij}^{(2)}(n)} = -\frac{\partial \varepsilon(n)}{\partial y_{out}^{(n)}} \frac{\partial y_{out}^{(n)}}{\partial \Delta u(n)} \frac{\partial \Delta u(n)}{\partial O_i^{(2)}(n)} \frac{\partial O_i^{(2)}(n)}{\partial \delta_k^{(2)}(n)} \frac{\partial \delta_k^{(2)}(n)}{\partial w_{ij}^{(2)}(n)}
\]

(24)

\[
\frac{\partial \delta_k^{(2)}(n)}{\partial w_{ij}^{(2)}(n)} = O_i^{(1)}(n)
\]

(25)

Because of the \( \frac{\partial y_{out}^{(n)}}{\partial \Delta u(n)} \) unknown, the approximate symbol function \( \text{sgn} \left( \frac{\partial y_{out}^{(n)}}{\partial \Delta u(n)} \right) \) is used here instead, and the effect of improper result can be compensated by regulating the learning rate.

From equation (2-4), the following can be obtained:

\[
\begin{align*}
\frac{\partial \Delta u(n)}{\partial O_i^{(1)}(n)} &= e(n) - e(n-1) \\
\frac{\partial \Delta u(n)}{\partial O_i^{(2)}(n)} &= e(n) \\
\frac{\partial \Delta u(n)}{\partial O_i^{(3)}(n)} &= e(n) - 2e(n-1) + e(n-2)
\end{align*}
\]

(26)

Thus, the modified formula of weight adjustment of network output layer is:
\[
\begin{align*}
\Delta w_{ij}^{(2)}(n) &= \eta \delta_{i}^{(2)}(n)O_{j}^{(1)}(n) + \alpha \Delta w_{ij}^{(2)}(n-1) \\
\delta_{i}^{(2)}(n) &= e(n) \text{sgn} \left( \frac{\partial Y_{\text{out}}(n)}{\partial \Delta u(n)} \right) \frac{\partial \Delta u(n)}{\partial O_{j}^{(2)}(n)} f'(v_{i}^{(2)}(n))
\end{align*}
\]  

Likely, the weight learning algorithm of the concealed level:

\[
\begin{align*}
\Delta w_{ji}^{(1)}(n) &= \eta \delta_{j}^{(1)}(n)O_{i}^{(0)}(n) + \alpha \Delta w_{ji}^{(1)}(n-1) \\
\delta_{j}^{(1)}(n) &= \varphi'(v_{j}^{(1)}(n)) \sum_{k=1}^{3} \delta_{k}^{(2)}(n)w_{jk}^{(2)}(n)
\end{align*}
\]

The PID controller control algorithm on the basis of BP network is summarized as the steps below:

1. Initialization: select the BP neural network architecture, that is, decides on the node quantity in each level, give the initial value sum of the weight coefficient of each layer \( w_{ij}^{(2)}(0) \) and \( w_{ji}^{(1)}(0) \), and select the appropriate learning rate \( \eta \) and inertia coefficient \( \alpha \), at this time \( n=1 \);
2. \( \text{rin}(n) \) and \( \text{yout}(n) \) are gained by random, and the error \( e(n) \) at this time is computed as \( \text{rin}(n) - \text{yout}(n) \).
3. The input and output of neurons at each level of the network are calculated. The output of the output layer is the three adjustable parameters of PID controller, and \( k_p, k_i, k_d \).
4. Figure out the output value \( u(n) \) of PID controller;
5. Neural network learning, online regulation of weight coefficient \( w_{ij}^{(2)}(k) \) \( w_{ji}^{(1)}(k) \); Achieve the adaptive adjustment of PID control parameters.
6. Set \( n=n+1 \) and go back to step 2.

4. Simulation Research of BP Neural Network PID Control Algorithm

4.1. Introduction to AMESim and Simulink/Matlab

4.1.1. Introduction to AMESim and Simulink/Matlab. AMESim is system engineering Advanced Modeling and simulation platform (Advanced Modeling Environment for Simulations of engineering systems). It is a modeling platform based on the intuitive graphical interface. During the whole simulation process, the simulation system is displayed intuitively through the graphical interface. Customers could construct complicated system models on this single platform, and conduct simulation calculation and in-depth analysis on this basis, or study the dynamic and static performance of components or systems on this platform.[15].

AMESim simulation software adopts the graphical modeling method based on physical model, providing users with a large number of component application library, freeing users from complex mathematical modeling, so as to focus on the simulation design of actual physical system. As a standard system simulation platform, AMESim has the following characteristics [16]:

1. System engineering modeling and simulation in interdisciplinary field is realized on its own single platform.
2. The intelligent solver can automatically select the optimal algorithm according to the characteristics of the model, and can dynamically switch integral algorithm and adjust step size according to the characteristics of the system at different moments in the simulation process, so as to solve the problem of simulation discontinuity.
3. Extract the smallest elements that constitute the engineering system, so that as few elements as possible to build as detailed as possible to reflect the complex model of system and parts function.

4. The graphical modeling method based on physical model makes it more convenient for engineers and technicians to use.

5. It provides a graphical, standardized and standardized secondary development platform that can integrate C or FORTRAN code models into AMESim software package.

6. Provide a variety of simulation operation modes, such as: dynamic simulation mode, steady-state simulation mode, batch simulation mode, etc.) and complete analysis tools (linearization analysis tools: eigenvalue solution, Nichols diagram, modal analysis tools, Bode diagram, root trajectory analysis, etc.); Spectrum analysis tools: spectrum graph, fast Fourier transform, order analysis, etc.

7. It provides rich interfaces with other software (MatrixX, MATLAB, Adams, xPC, iSIGHT, Excel, etc.).

Simulink is an environment developed by MathWorks software for modeling and simulating control systems under MATLAB software. The so-called model block diagram is that some basic system modules are provided by Simulink, these modules are categorized by function, it is necessary for users to get to know the function of these modules, and do not have to study how the module internal implementation, so that users can focus more on the design of the system itself.

Simulink is a powerful simulation tool that allows users to simulate the running process of a real dynamic system with minimal cost in the model block diagram mode. Simulink has the following features [17]:

1. Can establish dynamic system model and carry on simulation, can analyze all kinds of real dynamic system almost.

2. Using the visual modeling method, the block diagram model can be quickly established.

3. Allows users to add custom module components and their own written code (such as C, FORTRAN, etc.).

4. SIMULINK solvers ensure fast and accurate simulation of continuous or discrete systems.

5. The complex model system can be expressed by sub-model hierarchy.

6. The calculated results can be saved to the MATLAB workspace, and the data can be operated by using the analysis and visualization tools in MATLAB.

4.1.2. Significance of AMESim and Simulink/Matlab co-simulation. The reason why this article adopts the two joint simulation software AMESim and Simulink, quill because HCU test system is relatively complex, which includes both the hydraulic part, control part, if only using Simulink software simulation, will is complicated, and using AMESim simulation of hydraulic part, keep the control section in the Simulink simulation, using the AMESim interface technology of Simulink, the two simulation software, at the same time can also use the Matlab powerful numerical analysis ability, take full advantage of it, More perfect simulation effect is achieved [18].

4.2. BP neural network PID controller design on the basis of S function

S-function here is referred to System Function. it is a kind of computer language used to describe the dynamic system.

Although Simulink simulation function is powerful, but for some complex control laws, such as neural network control, fuzzy control and other intelligent algorithms, Simulink has no ready-made modules to build. At this time we need to introduce Simulink new extension function - S function.

The basic format of the S function is: [19]

Function [sys, x0, STR, ts] = Function name (t, x, u, flag).

Where: X is the state variable of the system; t is the current simulation time; U is the input vector; Flag is the return variable flag used to distinguish calls to different control functions. The following table 1 Shows the corresponding calling functions and function descriptions of flag flags.
Table 1. Corresponding call function and function description of flag flags

| Flag Symbol | Call a Function                  | Functional Specifications                                      |
|-------------|----------------------------------|----------------------------------------------------------------|
| 0           | mdlInitilizeSizes                 | System model initialization function                           |
| 1           | mdlDerivatives                    | Calculate the continuous state variable differential equation  |
| 2           | mdlUpdate                        | Update discrete state variables                                |
| 3           | mdlOutputs                        | Calculate the s-function model output                          |
| 4           | mdlGetTimeOfNextVarHit           | Calculate the time at the next sampling point                 |
| 9           | mdlTerminate                     | End simulation task                                            |

In this paper, the core S function of BP neural network PID controller [20,21] is shown in figure 7. (a) and figure 7. (b):

![function](image)

Figure 7. (a). S function design of PID controller based on BP neural network.
The controller in this paper utilizes the three-layer (4-5-3) BP neural network structure, and the input of the four nodes of the input layer (IN) are: system set value ($r_{in}(k)$), actual value ($y_{out}(k)$), error($k$) and 1. The number of hidden layer (H) nodes is 5, and the number of output layer (OUT) nodes is 3, corresponding to the three PID controller adjustment parameters $K_p$, $K_i$ and $K_d$. $W_i$ and $w_0$ are the implicit layer weighting coefficient and the output layer weighting coefficient respectively, and their initial values are generated by the rand function. $X_i$ and $\alpha$ are learning rate and inertia factor separately. The initial setting of the established $S$ function system model is 5 inputs and 4 outputs.

4.3. Establishment of AMESim simulation model of HCU control system

There are many basic component libraries in AMESim, such as: signal control library, mechanical library, hydraulic library, pneumatic library and so on, as shown in figure 8 (a).

The simulation model of pressure control system built by AMESim is mainly divided into the following four steps: [22].
To ensure better verify the influence of neural network PID algorithm on pressure regulator pressure control, this paper will simplify the actual hydraulic system structure appropriately, remove some elements that have little impact on the pressure control effect. The simplified AMESim model [23-25] established is shown in figure 9.

Figure 9. AMEsim model of hydraulic control unit test device control system
Main parameters are shown as follows:
1. Viscosity of medium: 0.0007 Pa*s, Medium density: 720g/L
2. Hydraulic gear pump displacement q= 1.76cc /rec, the highest speed n=3000r/min
3. The detecting range of pressure sensor is p=0~65bar
4. The detecting range of the flow sensor is q=0.01~60L/min
5. Temperature: 40℃

In the AMESim software, click model-> interface module -> to create the interface icon, and the dialog box shown in figure 3.8 will appear. AMESim can conduct co-simulation with a variety of simulation software, including Simulink, Adam, etc., this paper chooses Simulink here. There are two inlet interfaces of the designed "bp-pid. In Simulink” module. The "Pressure" port returns the Pressure signal to the controller and obtains it via the pressure sensor. The “Flow” port returns the Flow signal to the controller and obtains it via the flow sensor. The counterpart output ports are "Servo4Q" and "Servo4P" respectively, ”Servo4Q” means the gear pump servo motor control signal, ”Servo4P” means the press servo motor control signal.

Figure 10. Simulink interface creation

When the model sketches are completed and the parameters of each component submodel are configured, the system can be compiled. As shown in figure 11, Simulink standard S function was successfully created.

Figure 11. AMESim system compilation

4.4. AMESim and Simulink/Matlab joint simulation platform implementation

4.4.1. Simulink imports S function. In this paper, AMESim and Simulink/Matlab joint simulation platform, Simulink model needs to import two S functions, one is Matlab language prepared by BP neural network PID algorithm S function; The other is the S function created by AMESim software, Click the AMESim tools menu ->Simulink, you can open Simulink software, if the interface Settings of the two software are correct, Simulink path will automatically set to the current AMESim model path. The establishment of S function in Simulink, the name needs to be set to add "_" form on the name of AMESim model, in order to achieve the combination of AMESim model and S function. In parameter setting, the first parameter "1" represents the generation of simulation result file, and the second parameter is used for sampling time interval, which is set to 0.01 seconds in this paper, as shown in figure 12.
4.4.2. Simulink modeling and simulation. Combined with the BP neural network PID controller S function established above and the system model S function established by AMESim software, the Simulink control model as shown in figure 13 is established.

In FIG. 13, Signal value is input to the system through Signal Builder module. When the hydraulic system enters the stable status and after it takes about two seconds, 4bar step Signal is given. Figure 14 and figure 15 respectively show the pressure and flow response curves under the 4bar step signal. Look at the simulation results, we can find that the neural network PID control algorithm designed in this paper can greatly decrease the overshooting of pressure, the pressure control is steady and the flow fluctuation is small, which can satisfy the desirable pressure setting request of the pressure regulator for HCU testing hydraulic pipeline of the pressure setting system.

Figure 12. communication parameter setting operation

Figure 13. Simulink control model
Figure 14. Pressure response curve of 4bar step signal input

Figure 15. Flow response curve under step signal input of 4bar

Figure 16. Parameter tuning curve of PID controller based on BP neural network

Figure 16 shows the self-tuning curve of KP, Ki and Kd of the PID controller on the basis of BP neural network in the simulation process. We can find that by learning BP neural network, KP, Ki and
kd of the PID controller can be modified online, so that the parameters can be quickly stabilized at the optimal value, which is incomparable to the traditional PID.

5. Conclusion
In this paper, in accordance with the control needs of HCU test system and taking the characteristics of hydraulic circuit into consideration, neural network and traditional PID control are combined and selected as a control mode. Firstly, the PID controller of BP neural network based on S function was designed by Matlab language. Then, combining the powerful hydraulic system modeling and simulation capabilities of AMESim and Simulink software, the simulation of the designed BP neural network PID controller was conducted to examine the effectiveness of the control algorithm, which was fundamental for the design of subsequent software control algorithm.

References
[1] Tao yonghua. New PID control and its application -- first lecture on PID control principle and self-tuning strategy [J]. Industrial instrument and automation equipment, 1997, 04: 60-64+46.
[2] NI's official website, http://sine.ni.com/nips/cds/view/p/lang/zhs/nid/210669.
[3] Optimum Settings for automatic controllers [J]. Trans. ASME, 1942, 64 (11).
[4] Astrom K J, Hang C C, Persson P, et al. Towards intelligent PID control[J]. Automatica, 1992, 28 (1): 1-9.
[5] Section mechanics. Classification and overview of PID parameter setting method [J]. Modern computer (professional edition), 2012, 07:23-26.
[6] Sun hang, han hongxia, cao lihua, geng aihui. Fuzzy self-tuning of PID parameters in velocity loop of large photoelectric theodolite [J]. Chinese journal of instrumentation, 2013, 10: 2388-2394.
[7] Artificial Intelligence, 2009. JCAI'09. International Joint Conference on. IEEE, 2009: 148-150.
[8] Xi P, Song y. Application Research on BP Neural Network PID Control of the Belt reference [J]. Journal of Digital Information Management, 2011, 9(6): 267.
[9] Xu S, Huang Y, Qu L, et al. FPGA Realization of PID Controller Based on BP Neural Network [J]. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013, 11 (10): 6042-6050.
[10] Huo X, Hu J, Li z. BP neural network based PID control for ship steering [C]. Information and Communication Technologies (WICT), 2012 World Congress on. IEEE, 2012: 1042-1046.
[11] F.J.P ineda. Recurrent Backpropagation and Dynamical Approach to Adaptive Neural Computation [J]. Journal of Neural Computation, 1989, 1:161-172.
[12] F.J.P ineda. The Generalization of the Back - Propagation to Recurrent Neural Networks [J]. Phys. Rev. let, 1985, 59 (19) : 2229-2232.
[13] Tan yonghong. Adaptive control based on BP neural network [J]. Control theory and application, 1994, 01: 84-88.
[14] Zhai F F, Ma S L, Liu w. A Study on PID Control and Simulation Based on BP Neural Network[C]. Advanced Materials Research. Trans Tech Publications, 2012, 468:742-745.
[15] Li guang-rui, jiao nong, hong shu-hua. Research on co-simulation and comparison of the servo system of the range adjustment propeller based on AMESim/Simulink [J]. Marine engineering, 2014, 01: 49-51+103.
Based on ADAMS/Simulink/AMESim [J]. Journal of agricultural machinery, 2010, 10:11-17.

[20] Cheng weiguo. Application guide of MATLAB5.3 [M]. Beijing: People's posts and telecommunications press, 2000.

[21] Yang yi, tiger en. BP neural network PID controller based on S function and Simulink simulation [J]. Electronic design engineering, 2014, 04: 29-31+35.

[22] Li shao-ming, zhao wei. BP neural network PID controller Simulink simulation based on S function [J]. Programmable controller and factory automation, 2008, 03: 95-96+89.

[23] Dahai Z, Wei L, Yonggang L, et al. Modeling and Simulation of Marine Current Turbine with Hydraulic Transmission System Based on AMESim [J]. Acta Energiae Solaris Sinica, 2010, 2: 017.

[24] Hao M, Jiang w. AMEsim -based simulation on hydraulic experiment rig for the assembly of stator components [J]. Journal of Shanghai Jiaotong University (Science), 2013, 18:570-576.

[25] Ma changlin, huang xianxiang, hao Lin. Simulation and optimization of electro-hydraulic servo system based on AMESim [J]. Chinese hydraulics pneumatics & seals, 2006, 01:32-34.