LabVIEW Based Ventilator Algorithm Design

Chunyan Meng

Center of Computer, Taiyuan Normal University
Taiyuan, China
lvshuyang@163.com

Abstract

This paper focused on the design of ventilator algorithm and Hardware-in-Loop Simulation on the LabVIEW based ventilator control system. The ventilation algorithm had a double-layer structure. The upper layer was ventilation mode algorithm, while the lower layer was PID control algorithm. They worked at different periods. The lower layer served the upper layer and to achieve the constant voltage or constant current control the upper layer required. The upper layer determined the functionality of the ventilator, while the lower layer determined the performance of the ventilator.

Keywords: *Ventilation, PID control, Ventilator, LabVIEW*

1. Introduction

Ventilator is a mechanical device that helps patients with their respiration. The ventilator for therapy is for the severely sick and asks for the ability to realize a variety of ventilations depending on the needs [1]. The algorithm of ventilator directly determines the ventilation modes and performance, so reliable ventilator algorithm design is the key technology. It is a guarantee to the performance of ventilator algorithm to configurate and simulate the software and the hardware of the ventilator flexibly when the LabVIEW based ventilator control system is at the prototype developing stage [2].

2. The LabVIEW based ventilator control system

The system is available to various ventilation modes, depending on the on-off state of the proportional valve and the motor. The system structure chart is shown in Figure 1.

The proportional valve, the motor with its driving circuit, and the sensor with its amplifying circuit in the system are practical physical devices, while the human-computer interface, the ventilation mode algorithm and the PID controller are fulfilled by running a LabVIEW program on two computers. The upper computer with the Window XP Operating System achieves the human-computer interface and the ventilation mode algorithm, and the lower Desktop PC with the NI Data Acquisition Card and the LabVIEW RT module achieves data acquisition and PID control [3]. The adoption of the system converts the prototype to the embedded product easily and seamlessly, accelerating the development speed and development efficiency.

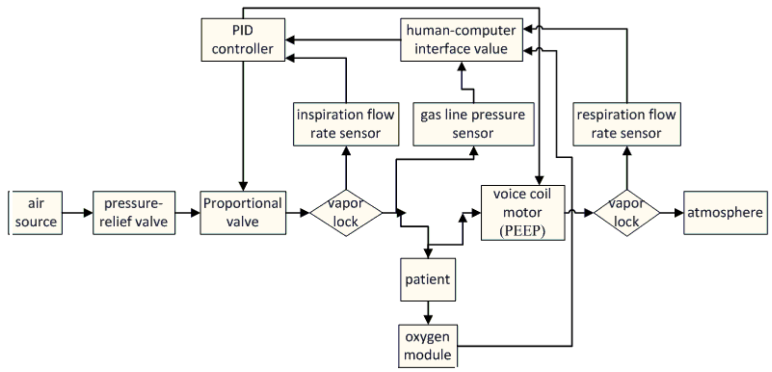

Figure 1. The System structure chart

3. Algorithm Structure

The ventilation algorithm design has a double-layer structure, as shown in Figure 2. The upper layer is the ventilation mode algorithm, while the lower layer is the PID control algorithm. The lower layer serves the upper layer and to achieve the constant voltage or current control the upper layer required. It is a Nested structure control algorithm in time sequence structure, as shown in Figure 3. The ventilation mode algorithm and the PID control algorithm work at different periods. The former has a ventilation period R, characterized by seconds. The latter has a practical flow rate and pressure control sampling period T, characterized by milliseconds. The period R is three orders larger than the period T. The upper layer determines the functionality of the ventilator, while the lower layer determines the performance of the ventilator.

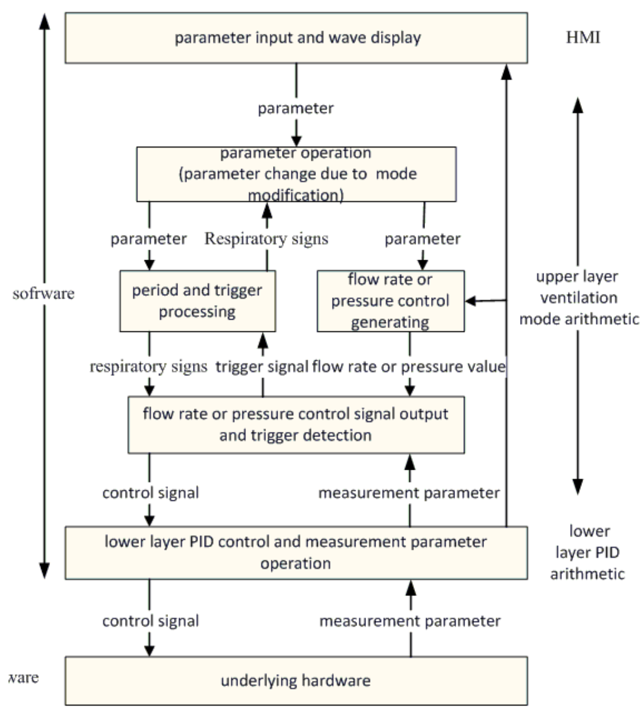

Figure 2. Algorithm structure chart

4. Lower layer PID controller design

The PID controller of the ventilator adopts the constant current PID controller or the constant voltage PID controller, depending on different ventilation modes and ventilation stages. The best proportional, integral and differential coefficient of different flow rates or pressures are determined by different constant current or constant voltage levels, i.e. the sectional PID controller, through experimental methods. The PID control uses incremental PID control algrithm, and limits the value of increment and ouput [4, 5]. The algorithm is as follows:

$\triangle u(k)=KP*\triangle e(k)+KI*e(k)+KD*[\triangle e(k)-\triangle e(k-1)]$, where $\triangle e(k)=e(k)-e(k-1)$.

If $\triangle u(k)>$ upper range value, then $\triangle u(k)=$ upper range value;

If $\triangle u(k)<$lower range value, then $\triangle u(k)=$lower range value. $u(k)=u(k-1)+\triangle u(k)$,where $u(k)$ is the Kth sampling value of the controller's output.

If $u(k)>$ upper range value, then $u(k)=$ upper range value;

If $u(k)<$lower range value, then $u(k)=$lower range value.

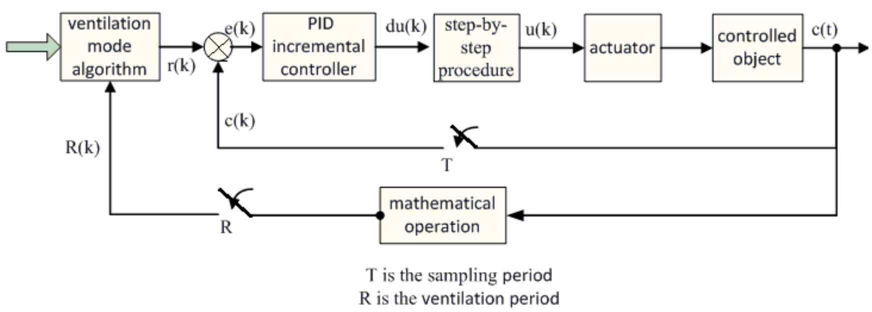

Figure 3. Algorithm structure chart

Where KP, KI, KD are proportional, integral and differential coefficient, respectively, and they are determined by the ventilation flow rate or the pressure level. e(k) is the difference between the designed value and the output value at the moment of K. The output of the PID controller controls the proportional valve and the motor directly, stabilizing the ventilation flow rate and the pressure at a given value. The algorithm flow chart is shown in Figure 4.

5. Upper layer ventilation mode design

The ventilation mode algorithm is carefully designed through building block design. There are five ventilation modes, comprising VCV, BiLevel, PRVC, SIMV and CPAP [6]. Each ventilation mode is individually designed and comprises four modules, as shown in Figure 2, which are the parameter operation module, the period and trigger processing module, the flow rate or pressure control generating module and the flow rate or pressure control signal output and trigger detection module.

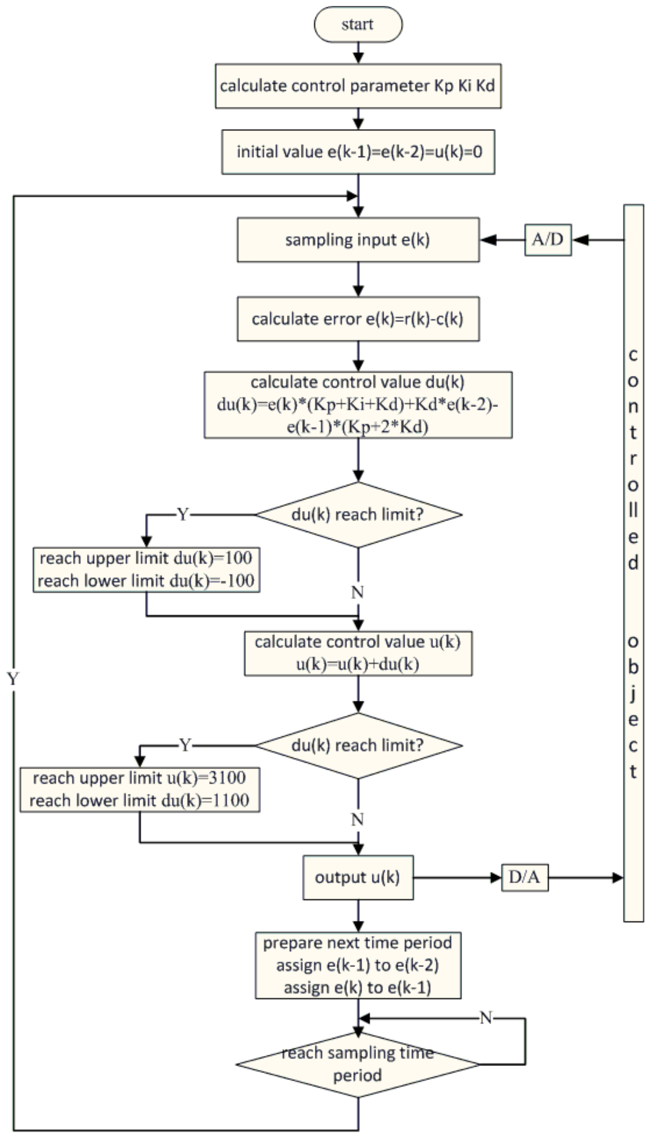

Figure 4. Incremental PID flow chart

The period and trigger processing module is the core of each mode algorithm, which generates ventilation time sequence, deals with errors such as trigger and pressure excess, and creates respiratory signs to control the rhythm of the other three modules. The period and trigger processing module flow chart of the BiLevel mode is shown in Figure 5. The design rules are:

1. Take expiration as the start of each ventilation period;
2. Take each period as the time tick, and each time tick accomplishes the tasks within it;

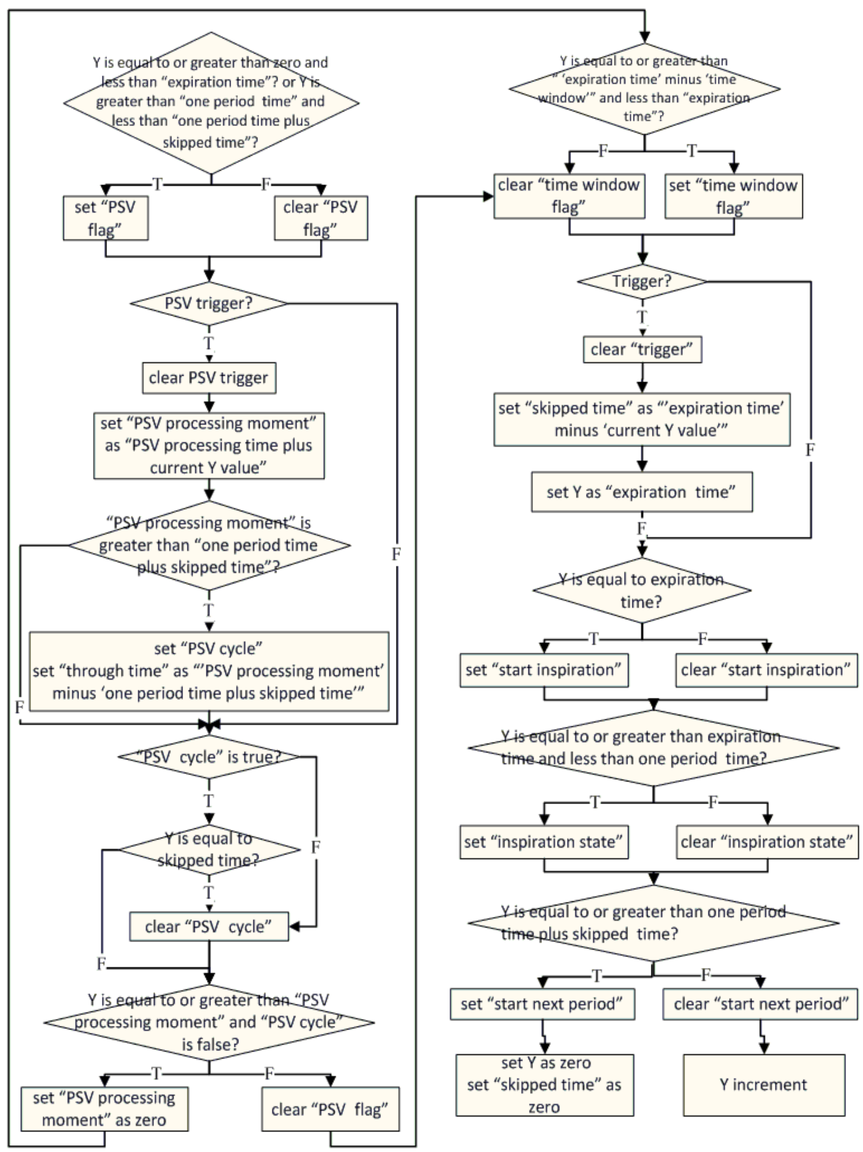

Figure 5. The period and trigger processing module flow chart of the BiLevel mode

3. Each time tick comprises of the available trigger state and the ventilation state, and the tasks are mutually exclusive and accomplished individually at each time area. In other words, only one task is dealt at one time point;

The parameter operation module controls user defined ventilation parameters at human-computer interface, such as the operation time of the ventilation mode and tidal volume, in order to avoid abrupt changes due to parameter modification when ventilation is in progress and to guarantee the accomplishment of the current tasks at the current period.

The flow rate or pressure control generating module converts user defined ventilation parameters at human-computer interface, such as ventilation frequency and tidal volume, to corresponding flow rate or pressure value. It receives the sampling value of sensors to modify flow rate or pressure control value at the same time, in order to accomplish feedback control, as shown in Figure 3. PRVC is the typical application of the feedback control.

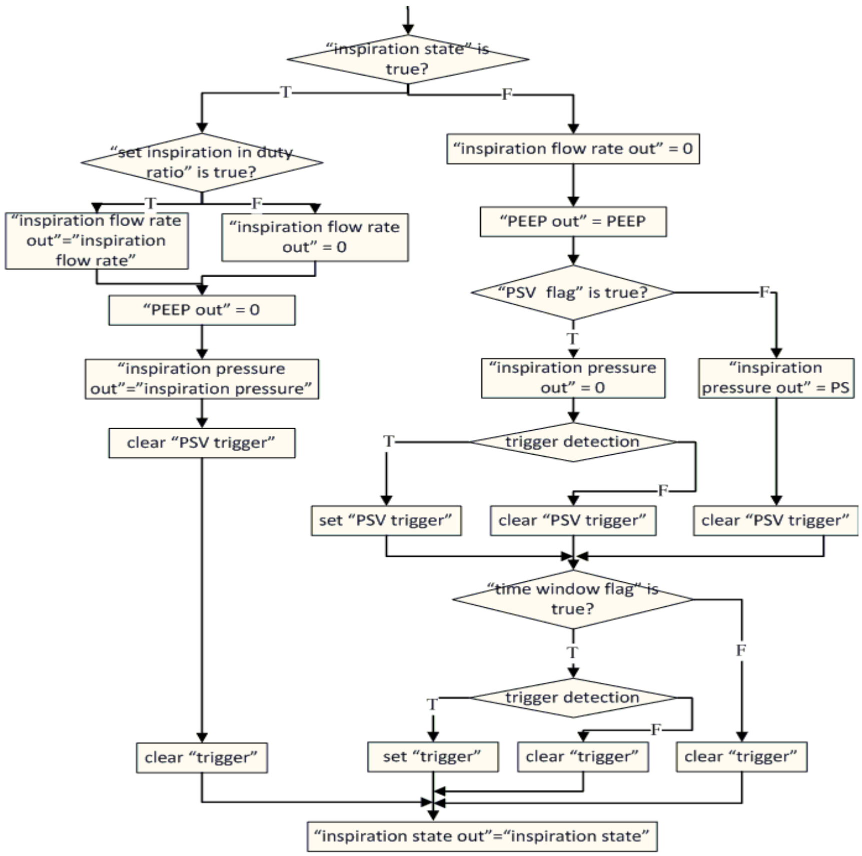

Figure 6. The flow rate or pressure control signal output and trigger detection module flow chart of the BiLevel mode

The flow rate or pressure control signal output and trigger detection module converts the flow rate or pressure value, which the flow rate or pressure control generating module generates, to the flow rate or pressure control signal according to the respiratory signs, which period and trigger processing module generates. It is applied to the lower layer PID controller and detects the trigger which the patient's active expiration generates. The flow rate or pressure control signal generates and triggers detections under close supervision of the respiratory signs, and the control signal or trigger detections are generated only if the corresponding respiratory signs is valid. The flow rate or pressure control signal output and trigger detection module flow chart of the BiLevel mode is shown in Figure 6.

The ventilation mode algorithm program is shown in Figure 7. The program could do Hardware-in-Loop Simulation when embedded in the LabVIEW based ventilator control system.

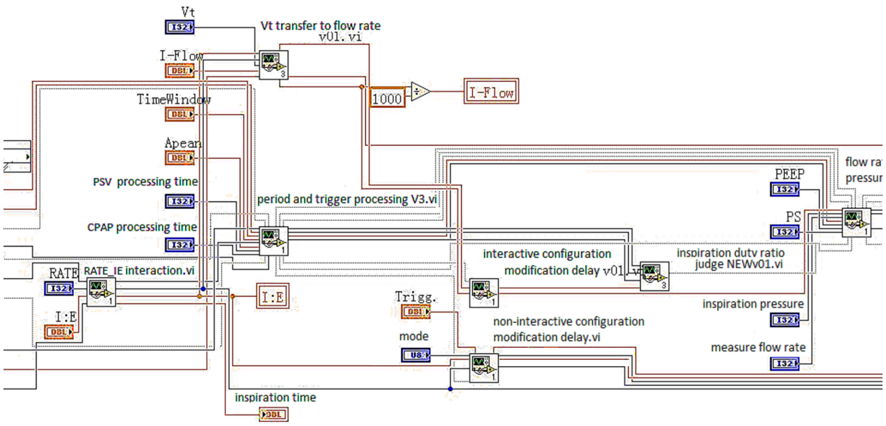

Figure 7. The ventilation mode algorithm program

6. Algorithm simulation

At human-computer interface, first configure the TCP/IP port and the ventilation parameters, and then run the program. Different ventilation waves are achieved when modifying the ventilation parameters, and one group of the VCV and BiLevel ventilation modes' ventilation wave curves are shown in Figure 8 and Figure 9, respectively. The curves are pressure wave, flow rate wave and configured flow rate/pressure wave from top to bottom. In the VCV mode, the configured tidal volume is 500ml and the practical control tidal volume is about 520ml, so it meets the precision requirements "the larger one of 10% or +/-20mL the measured value". In the BiLevel mode, the configured inspiratory pressure is 20 cmH2O and the practical control inspiratory pressure is about 20.8 cmH2O, so it meets the precision requirements "+/-(5% the measured value +0.5 cmH2O)".

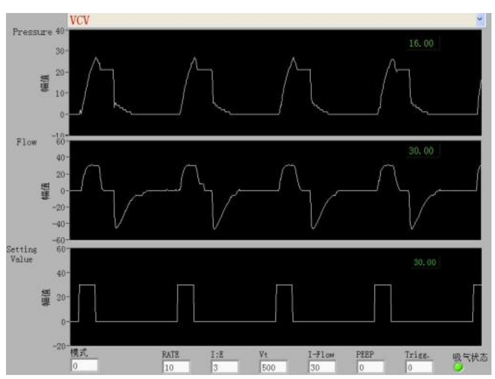

Figure 8. VCV experimental results of curve

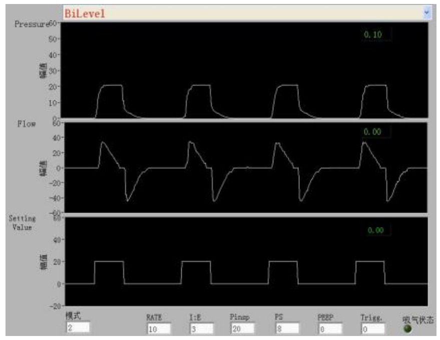

Figure 9. BiLevel experimental results of curve

7. Conclusion

This paper focused on the design of the ventilator algorithm and Hardware-in-Loop Simulation on the LabVIEW based ventilator control system. The whole design dealt with the ventilation mode algorithm and the PID control algorithm in different layers, resulting in a clear algorithm structure. The mode classification in the ventilation mode algorithm simplified the algorithm design further. The Hardware-in-Loop Simulation on the LabVIEW based ventilator control system corrected the algorithm error timely, optimized the performance of the algorithm and met the design specification perfectly.

References

[1] B. Wang, "Clinical Application of Ventilators", Beijing: PEOPLE'S MEDICAL PRESS, **(2005)**, pp. 108-130.
[2] M. Tang, "Based on the LabVIEW platform for rapid construction of medical electronic equipment prototype EDN", Design of electronic technology, vol. 7, **(2008)**, pp. 138-140.
[3] National Instruments LabVIEW v8.6, Real Time Module user Manual, **(2008)**.
[4] Y. Tao, "The new PID control and its application", Beijing: CHINA MACHINE PRESS, **(2001)**, pp. 1-27.
[5] J. Liu, "Advanced PID control MATLAB simulation (Second Edition)", Beijing: PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, **(2007)**, pp. 10-90.
[6] SIEMENS Servo-i User Manual, **(2001)**.
[7] M. Ali, S. Khan, M. Waleed and Islamuddin, "Application of an Intelligent Self-Tuning Fuzzy PID Controller on DC-DC Buck Converter", IJAST, vol. 48, **(2012)** November, pp. 139-148.
[8] R. Anbuselvi and R. S. Bhuvaneswaran, "Simulation of Path Finding Algorithm – a Bird's Eye Perspective", IJAST, vol. 9, **(2009)** August, pp. 31-38.
[9] M. Z. A. Rashid, M. S. M. Aras, M. A. Kassim, Z. Ibrahim and A. Jamali, "Dynamic Mathematical Modeling and Simulation Study of Small Scale Autonomous Hovercraft", IJAST, vol. 46, **(2012)** September, pp. 95-114.

Author

Chunyan Meng

Female, Associate Professor, was born in Taiyuan of Shanxi Province in 1971. Her research areas is computer network technology and computer application technology.