

**2003 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM**

**JOHN F. KENNEDY SPACE CENTER  
UNIVERSITY OF CENTRAL FLORIDA**

**Development of Algorithms for Control of Humidity  
in Plant Growth Chambers**

Thomas A. Costello, P.E.  
Associate Professor  
Biological and Agricultural Engineering Department  
University of Arkansas

KSC Colleagues: John C. Sager and Raymond M. Wheeler  
Biological Sciences Office  
Spaceport Engineering and Technology Directorate

**ABSTRACT**

Algorithms were developed to control humidity in plant growth chambers used for research on bioregenerative life support at Kennedy Space Center. The algorithms used the computed water vapor pressure (based on measured air temperature and relative humidity) as the process variable, with time-proportioned outputs to operate the humidifier and de-humidifier. Algorithms were based upon proportional-integral-differential (PID) and Fuzzy Logic schemes and were implemented using I/O Control software (OPTO-22) to define and download the control logic to an autonomous programmable logic controller (PLC, 'Ultimate' ethernet brain and assorted input-output modules, OPTO-22), which performed the monitoring and control logic processing, as well the physical control of the devices that effected the targeted environment in the chamber. During limited testing, the PLC's successfully implemented the intended control schemes and attained a control resolution for humidity of less than 1%. The algorithms have potential to be used not only with autonomous PLC's but could also be implemented within network-based supervisory control programs. This report documents unique control features that were implemented within the OPTO-22 framework and makes recommendations regarding future uses of the hardware and software for biological research by NASA.

# Development of Algorithms for Control of Humidity in Plant Growth Chambers

Thomas A. Costello

## 1. INTRODUCTION

Considerable plant growth chamber infrastructure has been maintained by NASA within the Biological Sciences Office at Kennedy Space Center (KSC). The systems were used to study response of crop plants to changes in artificial light regimes, controlled gaseous composition, temperature, atmospheric pressure, and nutrient/water delivery. The intent of the research was to optimize systems to grow food, regenerate oxygen, remove carbon dioxide, recycle nutrients, stabilize wastes and purify water for long-term space missions. At present, this infrastructure for advanced life support research, including 6 plant growth chambers, is being moved from Hangar L (and Little L) at the Cape Canaveral Air Force Station to a new NASA/KSC facility owned by the State of Florida (known as 'SERPL'). With the move, there will be installation of 9 additional plant growth chambers, along with the associated environmental systems, controls and software.

Although commercial environmental controllers were typically included with each chamber, researchers have preferred to use these only as a backup, and have relied principally upon custom data acquisition and control hardware/software. In the past, custom NASA-developed software (i.e., UNDAACE, Universal Networked Data Acquisition and Control Engine) operated on a Unix workstation, and was interfaced via serial communications to input/output modules that provided connection to sensors and actuators. The UNDAACE scheme was successful not only in providing real-time control and automatic data acquisition but it also provided a visual interface for researchers to easily review growth chamber conditions during an on-going experiment. Despite its past functionality, UNDAACE will not be implemented in SERPL because of the inability to transfer and maintain the code to a newer web-based platform.

In the new facilities, the function of UNDAACE will be performed either by autonomous control algorithms residing in local programmable logic controllers (PLC's, i.e., 'Ultimate' ethernet brain with associated input/output modules, OPTO-22, Temecula, CA) or by supervisory control programs residing on a network server. In either case, researchers should have the ability to select/design specialized control algorithms to optimize the control of lights, carbon dioxide, heating-cooling equipment, humidifying-dehumidifying equipment and other devices which impact the life support experiments.

In growth chambers, humidity is a relatively difficult parameter to control because of its interactions with the heating-cooling equipment. For example, the process of cooling air often removes water via condensation at the cooling coil. This impact depends upon the dynamic energy balance of the chamber, which can be affected not only by the selected inside temperature setpoint and the ambient temperature external to the chamber, but also by changes in heat introduced by artificial light fixtures/lamps. Moreover, variations in chamber temperature (T) necessarily impose concurrent fluctuations in relative humidity (RH) because RH depends upon saturation vapor pressure, which is a direct function of T. Control is further complicated by the use of two devices, a humidifier (HUM) and de-humidifier (DEH), to independently effect increases and decreases in humidity in an attempt to achieve the targeted humidity setpoint.

A simple approach to humidity control has been previously tested using autonomous PLC's at KSC. The algorithm used an RH setpoint (with specified deadband) as thresholds to activate timed on/off control of the HUM and DEH. Whenever HUM or DEH was triggered, the specified on-time was followed by a

required, fixed wait (off) time. The disadvantage of this type control was that the fixed on-times and off-times do not allow the controller response to be sensitive to the magnitude of the deviation between the current RH and the setpoint; thus, the devices may respond too slowly or they may overshoot. Overshoot causes the two devices to operate alternately in sequence with one device counteracting the overshoot of the other. Although this control scheme provided approximate control resolution of 5% of the RH setpoint, excessive HUM and DEH activation was inherent.

Because of the interaction between T (and temperature control) and RH, it was hypothesized that a more stable control might be obtained if *absolute* humidity (e.g., vapor pressure, VP) were used as the process variable. To accomplish this, the VP would need to be computed in real time based on measured values of T and RH. This would necessitate the use of a controller that is able to perform real-time digital computations. Given that such a controller is used, then the flexibility would also exist to implement a variety of control algorithms, including those using PID (proportional-integral-differential) or Fuzzy Logic.

### Objective

The objective of this project was to develop and test humidity control algorithms that operate with the following desired characteristics: a) deviations from a VP setpoint will be minimized to achieve the desired humidity conditions, b) activation times of the HUM and DEH will be minimized to extend the life of the devices, and c) frequency of cycling of the HUM and DEH will be minimized to extend the life of control relays, valves, and motors.

## 2. DESCRIPTION OF THE ALGORITHMS

Control algorithms were designed to use the computed water vapor pressure (VP) as the process variable, with time-proportion outputs (TPO) to operate the HUM and DEH. Multiple algorithms were developed, including those based upon a proportional-integral-differential (PID) scheme and others using a Fuzzy Logic scheme. The algorithms were implemented using I/O Control software (OPTO-22) to generate and download the algorithms to the autonomous Ultimate PLC, which performed the monitoring and control logic processing, as well the physical control of the devices, which effected the targeted environment in the chamber.

With the I/O Control package, logic could be defined using a limited set of instructions represented in flow-chart form, or the control engineer could define a control program using a script language similar to Visual BASIC or C, with real-time access to sensor readings and the ability to manipulate digital and analog outputs. Use of the script language was preferred for this project due to the prevalence of detailed conditional logic that is tedious to define using the flow-charting facility. The actual script text for the algorithms can be obtained by contacting the senior author. A summary of algorithm features is presented here.

### PID Algorithm

Control of a process using a PID algorithm is a form of feedback control in which a device/actuator is fed a control output that is intended to effect changes in the selected process variable toward some targeted setpoint value. The control output is computed as a function of the dynamic trend in the process variable, given by the classic PID equation (e.g., Harrison and Bollinger [1]):

$$O = P * E + I \int E dt - D \frac{dE}{dt} \quad \text{Eq. [1]}$$

where, O = the control output,  
P = proportional gain parameter,  
E = error = (set point – process variable),  
I = integral gain parameter,  
t = time, and  
D = derivative gain parameter.

Historically, the PID control was used with analog controllers. The algorithm can operate on a digital controller by computing the output at some arbitrary time interval called the scan time ( $t_{scan}$ ). During each scan, the error, the accumulated error (a numeric integration) and the derivative of the error (the numeric approximation) are updated and then used to compute the contributions to the output based on P, I, and D, respectively. The P-gain is the primary control component that provides increased output as the process variable deviates from the setpoint (note that it can be positive or negative depending upon the sign of the error). The I-gain provides an adjustment to the output to avoid fixed offsets (persistent error) that are often associated with proportional controllers. The D-gain provides an adjustment that tends to decrease the rate of approach toward the setpoint or decrease the rate of divergence away from the setpoint, in an effort to attain control at the setpoint in minimal time with minimal overshoot.

Modern implementation of a PID in a digital controller provides the opportunity to modify the classic PID output using conditional logic to optimize system response. A number of modifications to the classic PID were implemented using the script language in I/O Control to operate the HUM and DEH, as described below. The computed VP was the process variable, computed using the method of ASAE [2], based on measurements of T and RH.

The PID algorithm output was defined in the range from –10 to 10, with negative values (-10 to 0) referring to DEH duty cycle from 100% (steady ON) to 0% (steady OFF), and positive numbers (0 to +10) referring to HUM duty cycle from 0% (steady OFF) to 100% (steady ON), respectively. This scheme did not allow both devices to be ON at the same time and included a center position (0) with both devices OFF. At regular time intervals, the output from the PID algorithm was interpreted as the duty cycle to control the ON/OFF sequence of either the HUM or DEH. This regular updating of the duty cycle of a discrete device (that has only two states, ON and OFF) is known as ‘time proportioned output’ (TPO).

A TPO period parameter,  $t_{TPO}$ , was specified to control the rate at which the PID output was sampled and interpreted. The ON-time for each cycle was computed as (PID\_Output/10 multiplied by  $t_{TPO}$ ), with the OFF-time computed as ( $t_{TPO}$  minus the ON-time). The I/O Control script code was written to execute the TPO-sequence (turn ON device, delay, turn OFF device, delay) for the needed device depending upon current readings of the VP relative to the VP setpoint, as processed by the PID algorithm. The overall scheme is shown in Figure 1. The PID calculations repeated every  $t_{scan}$  and called the appropriate TPO sequence every  $t_{TPO}$ .

The P-gain component of the PID algorithm was adjusted by inclusion of a multiplicative factor that was proportional to E, scaled to vary linearly between 1 and 5, with scaling based on a specified tuning parameter. This factor produced a non-linear response that was intended to offset observed behavior in which VP seemed to be especially sensitive to changes in HUM and DEH duty cycle near the setpoint.

Other non-linear functions could also be investigated to provide an alternative to standard proportional output that might better match system dynamics. Separate parameters for proportional gain were defined for the HUM and DEH.

The I-gain component was ignored when the VP process variable was considered to be 'out of control'. This method avoided the build-up of accumulated error (that would later need to be offset by forcing errors of opposite sign) during transient periods (following setpoint change or large disturbances such as opening the chamber door, turning off the lights, etc.). The 'out of control' condition was detected when VP was outside an arbitrary zone near the setpoint, defined by a tuning parameter,  $VP_{ControlZone}$  (a specified fraction of the VP set point). The process was also considered 'out of control' when the derivative of the VP exceeded some arbitrary limit defined by the tuning parameter,  $VP_{SmoothZone}$  (a specified fraction of the VP setpoint, per unit time). Thus, the I-gain only operated when the error was small and the VP was fairly steady.

Early versions of the PID algorithm for VP computed the D-gain component by estimating the derivative using a simple finite-difference approximation, namely  $(VP_{now} - VP_{previous})/(t_{scan})$ . Based on observations in real time, the resulting derivative did not appear to be a smooth function due to noise (apparent random time variations) in the VP input readings. The noise created D-gain adjustments that did not correspond to the intended effect, which was to decrease the control output when the VP was converging toward the setpoint (to avoid overshoot) or to boost the control when VP was drifting away from the setpoint (to increase responsivity). Hence, alternative methods of derivative calculation were tried, including simply using finite difference over multiples of  $t_{scan}$  with corresponding VP values taken from a stack from previous scans. Linear regression was also used to estimate the slope of the VP trend over the last few scans (using up to 16 previous values).

Based upon trial and error experimentation, the D-gain component was eventually modified using the concept of 'anticipated error' in which the VP trend was extrapolated to the next ( $t = t_{TPO}$ ) to provide a quantitative measure of where the VP was headed. The predicted deviation (rather than the derivative itself) was multiplied by an anticipated error gain parameter (different for HUM and DEH) to get D-gain. The anticipated error was extrapolated by fitting a second-order polynomial to the VP stack using multiple linear regression techniques. The script language did not support matrix algebra; hence, the regression parameters were estimated based upon the solution of a system of 3 simultaneous equations (the so-called 'Normal' equations, Draper and Smith [3]) using Cramer's Rule (i.e., finding ratios of 3X3 determinants algebraically). This scheme allowed the D-gain to detect and respond to curved trends in VP.

Observations of VP trends in real time indicated that the 'braking' and 'boosting' instances of derivative control needed to be detected and scaled according to the magnitude of the error. The braking control was augmented as VP approached the setpoint, and the boost control was augmented as VP diverged from the setpoint. This adjustment was accomplished using a multiplicative factor that was computed as a linear function of error and a specified tuning parameter.

Nominal PID output (sum of P, I and D components) was conditioned according to an additional set of tuning parameters. For HUM and DEH, the maximum and minimum values of the duty cycle were specified as well as the maximum output change per scan time. After conditioning the nominal output, the final output value was used to specify the duty cycle for either the HUM or DEH for the next TPO cycle.

### Fuzzy Logic Algorithm

Fuzzy Logic provides a framework to compute a control output for many complex processes for which human operators seem to develop an intuitive feel for successful manual control that cannot be duplicated by traditional automatic feedback loops (such as PID). The basis for a Fuzzy controller is a set of rules developed in consultation with experienced operators. The rules represent the knowledge of the operator and capture the manual control response to various input scenarios, converting that knowledge to code that can run automatically.

To implement a Fuzzy controller for VP, a Fuzzy algorithm was defined that provided a real time estimate of the VP-output. This code was substituted for the PID code within the OPTO-script program. The Fuzzy controller used the same code for actuating the HUM and DEH on a TPO basis. The steps in obtaining the Fuzzy output included: 1) classifying the inputs (VP and its derivative, VP') as: ('very dry', 'dry', 'VP-OK', 'humid', 'very humid') and ('rapid drying', 'slow drying', 'steady VP', 'slow wetting', 'rapid wetting'), 2) processing the rules that defined HUM/DEH-output for given combinations of the classified VP and VP', and 3) combining results of the rules to obtain a quantitative output.

The classification of the inputs, the so-called fuzzification (e.g., Paraskevopoulos [4]) was implemented using linear interpolation for each input class (or membership function) based on pre-defined function shapes and breakpoints. Breakpoints were coded as multiples of specified error or derivative values (with tuning parameters  $VP_{ControlZone}$  and  $VP_{SteadyZone}$ ), respectively. Tuning was performed by changing the zone parameter to simultaneously narrow or widen the class definitions. The membership function shapes were defined as overlapping 'Z', 'pi' and 'S' patterns, with function values at the breakpoints specified as 0 or 1 (see [4]).

Various sets of rules were tried, including the use of additional input variables (such as T, T' and the temperature control output,  $O_T$ , and  $O_T'$ ,  $O_T''$ ) in an attempt to anticipate and adjust for characteristic VP trends and their interaction with T. Many rules represented action analogous to the P and D of PID control, that is, outputs were increased for large error, and boosting and braking functions were coded into the rules based on VP'. Eventually, a simple set of 25 rules was defined, consisting of one rule for each combination of the 5 classes for VP and VP', see Table 1.

Rules were sorted to collect all rules that applied to each classification of the VP control output ('high dehumidify', 'medium dehumidify', 'low dehumidify', 'HUM/DEH off', 'low humidify', 'medium humidify', 'high humidify'), respectively. Rules within each group were tested individually, with the result of each rule being the minimum membership value among the antecedents (in the 'if' part of the rule). The output classification was then assigned the maximum value among the results for all rules in that group. The script language had MAX and MIN functions that were used in this process. Since the script language did not allow for multiple-dimension subscripted variables (only single dimension arrays), the rule processing could not be coded within a loop but was repeated explicitly for each group of rules. Single dimension arrays were used to represent membership function values for each input (VP and VP').

The so-called 'crisp' output from the Fuzzy algorithm was computed using the Center of Area method [4] in which the values for each output classification (from rule processing) were used to compute a weighted average for the output. This was done using a set of HUM and DEH duty cycles that were specified to correspond to the 7 classes of the output. These values were specified as parameters that could be altered in the tuning process.

### 3. PRELIMINARY TESTING

Testing of the control algorithms was performed in CEC-4 in Hangar L at KSC during July 2003. The chamber was a reach-in chamber (1 m x 1 m x 1 m) equipped with a commercial controller (model TC2, Environmental Growth Chambers, Chagrin Falls, OH) that actuated a 'Barber-Coleman' temperature control valve. No plants were grown in the chamber during the tests but water was added to the hydroponic trays (providing a free water surface for evaporation over approximately 75% of the bottom surface area). Two, 400-W high-pressure sodium lights were energized continuously during the tests.

Before testing to compare algorithm performance, each algorithm was tuned by a trial and error process in which dynamic trends in VP were monitored in real time and then adjustments were made to tuning parameters to improve the control response. Some delay in finding an acceptable set of tuned parameters for both the PID and Fuzzy controls was due to an interaction between the tested VP controls and the PID algorithm defined within I/O Control to control the air temperature in the chamber. This problem was overcome by re-wiring the control relays to allow the TC2 to control the Barber-Coleman valve while the PLC controlled the HUM and DEH.

After tuning was completed, the performance of both the PID and Fuzzy algorithms was tested. For each test, the chamber was initially controlled using the TC2 controller and was then sequenced to the selected I/O Control algorithm to control the HUM and DEH. Device activation states, as well as sensor (T, RH and computed VP) readings, were recorded every 1 s during each 30-min. test period.

### 4. RESULTS AND DISCUSSION

During limited testing, the PLC's successfully implemented the intended control schemes with acceptable resolution of humidity control after limited tuning. Compared to the commercial controller, the PID for temperature control implemented with I/O Control (not described here) introduced periods of accelerated drying and wetting that the VP controls could not overcome (see Figure 2). Eventually, the VP controllers were tested with the temperature controlled by the commercial controller. Modifications to the PID algorithm for temperature control need to be developed to avoid this behavior. It is possible that this anomaly might not be exhibited for larger growth chambers or for chambers with different temperature control valves. Testing is needed to verify this. The commercial controller output for the temperature control valve exhibited a technique called 'dithering' (visible in Figure 2) in which an arbitrary boost voltage is superimposed upon the PID output at some regular interval. This may have prevented the excessive condensation/re-evaporation that apparently occurred with the PLC control.

Examples of steady control of the VP using the PLC algorithms--PID (with a 5-s TPO period), PID (with an 8-s TPO period) and Fuzzy--along with the commercial control algorithm, are shown in Figures 3-6. The PID control (5-s TPO) provided the least VP variations around the setpoint. Additional verification of the control stability under other combinations of temperature and humidity is still needed. Formal testing of each algorithm's response to disturbances such as setpoint changes, door opening, lights on/off, etc. has not been done although abbreviated testing has indicated that the algorithms seem to be able to respond to disturbances adequately (within 1 min.).

Based upon this limited testing, the different algorithms seem to vary in their ability to maintain the targeted VP setpoint and their impact upon the HUM and DEH device utilizations, see Table 2. The mean VP measured over the 30-min. test with PID (5-s TPO) was nearly identical to the VP setpoint

(deviation was much less than 1%), with the other controllers achieving accuracy of 1% or less. The PID (5-s TPO) also performed best in terms of maintaining a steady VP with standard deviation of less than 1% of the mean. The commercial control operated the humidifier and dehumidifier 36% and 19% of the time, respectively, while the PLC-based controls operated them less than 10% and 4%, respectively. The cycle times for the commercial controller were longer than the other controls, which may affect the life of relays and actuating valves/motors.

## 5. CONCLUSIONS

The humidity inside plant growth chambers can be controlled using a PLC-based controller (“Ultimate” ethernet brain, OPTO-22) with computed vapor pressure as the process variable. The PLC script language provided adequate memory and process speed to implement PID and Fuzzy Logic controllers. The programming package (‘I/O Control’, OPTO-22) had a few operational quirks (mostly unexpected, intermittent results when real-time processing may have exceeded the CPU capacity) but these were overcome by streamlining the code. Otherwise, the package was easy to learn and easy to use. The combination of the Ultimate brain and I/O Control makes an efficient tool with adequate flexibility and power to perform virtually all anticipated control functions for growth chambers, bioreactors and apparatus for other life support experiments.

The PID and Fuzzy controllers reduced the required activation time for the humidifier and dehumidifier to 10% to 30% of that required by the commercial controller, due to the derivative control components and the TPO activation scheme. The PID algorithm exhibited the best performance in these tests; however, more testing is needed to confirm that the control is stable under many other test conditions. The Fuzzy control had the advantage of being intuitively simple and should prove to be very robust across many control scenarios. Since I/O Control does not support a Fuzzy tool set, tuning the Fuzzy algorithm is quite tedious. Tuning the PID algorithm could be facilitated with a simple operator interface that would allow on-line adjustments to setpoints and parameters. Once tested and tuned, either the PID or Fuzzy algorithms should provide acceptable humidity control, whether implemented autonomously on the PLC brain or operated by a supervisory control program on a web-server.

Further improvements to the performance of algorithms presented here might be accomplished by implementing a relay ‘anti-clicking’ routine (that was coded but not tested) to skip relay activation or deactivation when the duty cycle is nearly 100% or 0%, respectively. Tuning processes should include exploration of the impact of TPO period on VP control. The concept of ‘dithering’ also needs further exploration as a method to improve system response time.

## REFERENCES

- [1] Harrison, H. L., and Bollinger, J. G., *Intro. to Automatic Controls*, International Textbooks (1970).
- [2] American Society of Agricultural Engineers (ASAE), *Psychrometric Data: D271.2* (2000).
- [3] Draper, N. R., and H. Smith. *Applied Regression Analysis*, Wiley (1981).
- [4] Paraskevopoulos, P. N., *Modern Control Engineering*, Marcel-Dekker (2002).

## Acknowledgements

The author would like to thank Ms. April Lovelady for her contributions to the design of the Fuzzy Logic algorithm. The author would also like to thank Mr. Paul Burke for supplying the real time interface to the OPTO hardware, and also for suggestions and feedback on the design and tuning of the algorithms. Appreciation is also extended to Dr. Vadim Rygalov and Mr. Larry Koss for supporting the process of sensor calibration.

Table 1. Visualization of Fuzzy Rule Space.

	rapid drying	slow drying	steady VP	slow wetting	rapid wetting
very humid	off	low deh	med deh	high deh	high deh
humid	low hum	off	low deh	med deh	med deh
VP OK	med hum	low hum	off	low deh	med deh
dry	high hum	med hum	low hum	off	low deh
very dry	high hum	high hum	med hum	low hum	off

Table 2. Summary of VP Control Algorithm Testing.

Test Result	Control Algorithm			
	Commercial Control	PID (5-s TPO)	PID (8-s TPO)	Fuzzy
Ave. Dev. from VP Setpoint (Pa)	-	-0.8	-7.5	-15.4
% of Mean VP	-	0.05%	0.49%	0.96%
Standard Deviation of VP (Pa)	21.9	9.46	11.1	16.9
% of Mean VP	1.44%	0.61%	0.72%	1.05%
Ave. HUM Duty Cycle	35.9%	4.1%	8.0%	9.7%
Ave. HUM Cycle Time, s	78.3	38.3	30	32.7
Ave. DEH Duty Cycle	19.5%	2.8%	3.9%	2.3%
Ave. DEH Cycle Time, s	85.7	47.4	40.9	58.1

Note: VP setpoint deviation could not be determined for the commercial controller because the control was based on output from a different RH sensor that may have significant offset compared to the sensor used to record VP.

Figure 1. Schematic of the PID control scheme implemented using I/O Control software to control vapor pressure.

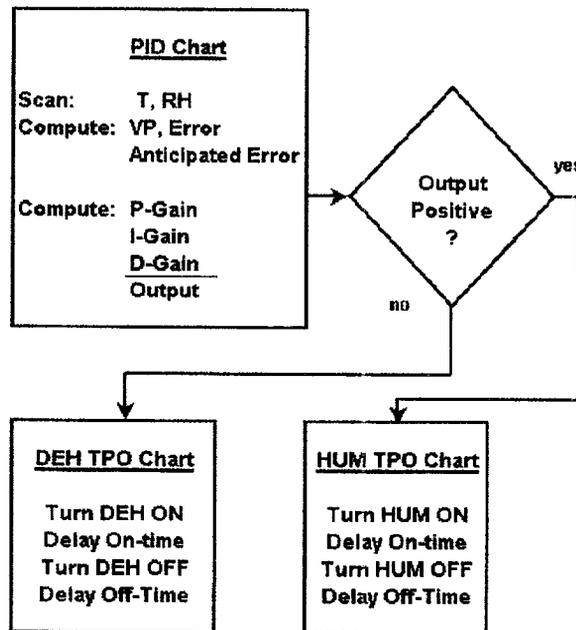


Figure 2. Center section shows humidity control using Fuzzy algorithm and temperature control using PID algorithm. Control prior to and after the center section--bounded by the dotted lines--was obtained using the commercial controller. The associated output to the temperature control valve is shown with scale 9 V = 900 Pa, read from the right hand axis.

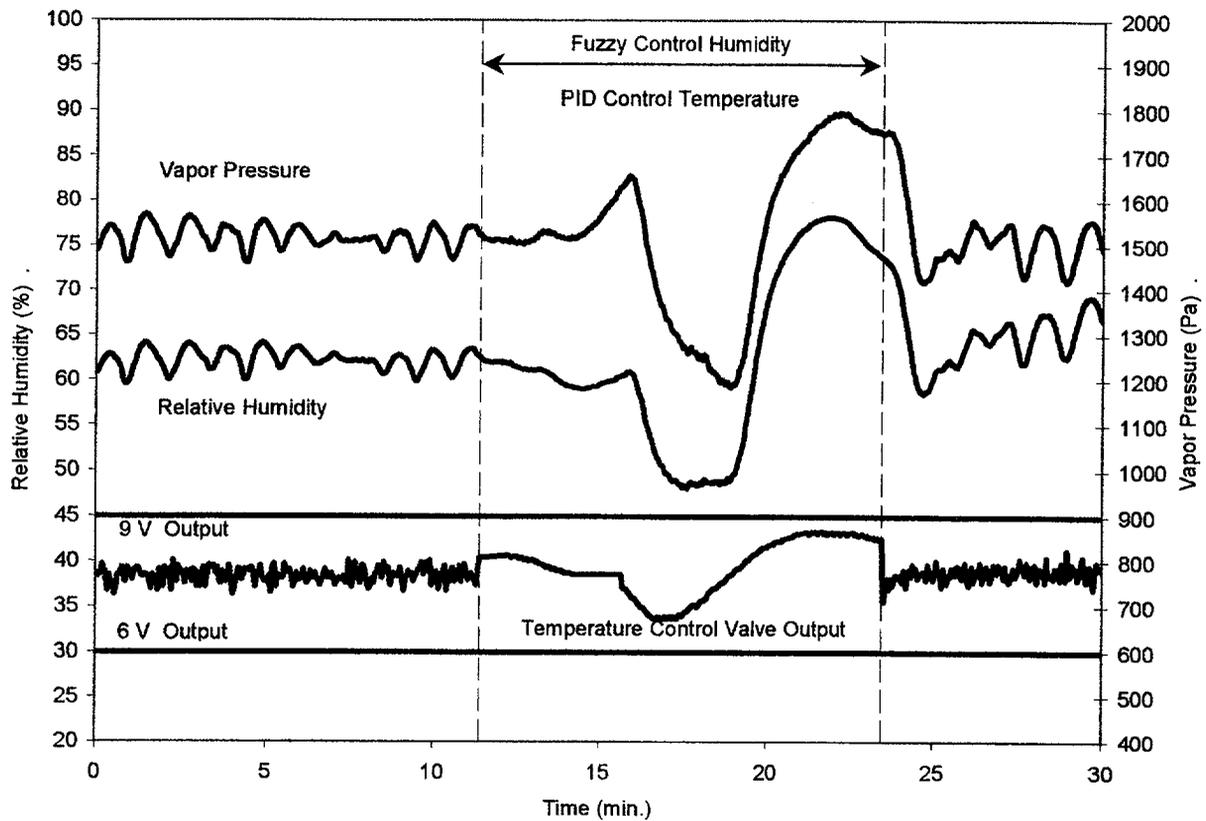


Figure 3. Humidity control with commercial controller. The pulse train for the humidifier and de-humidifier are also shown.

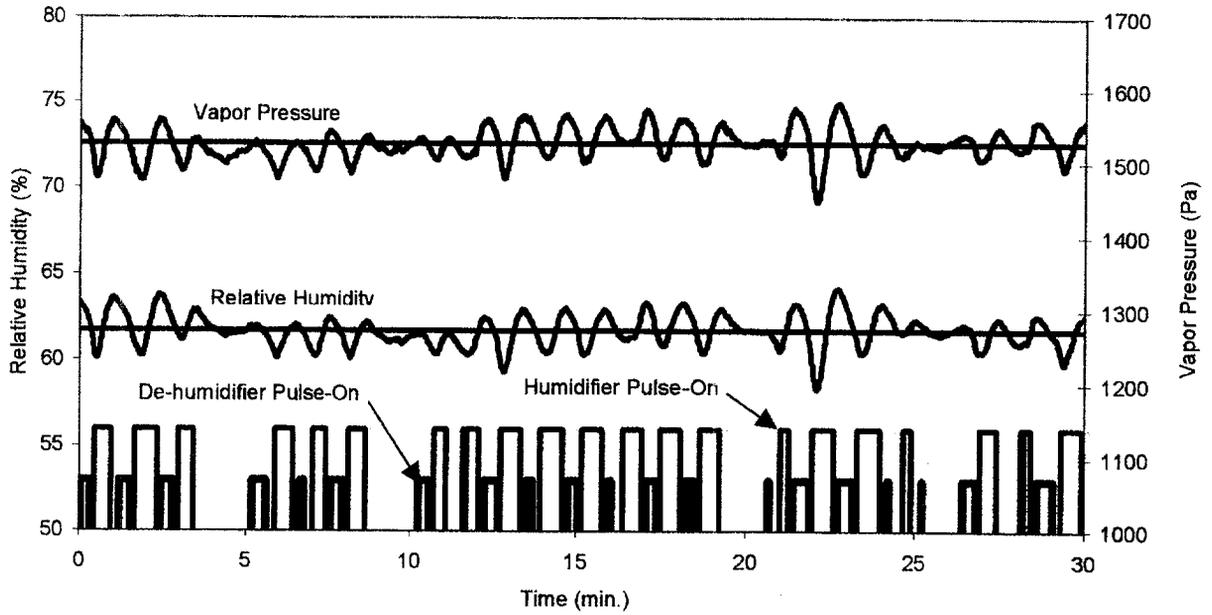


Figure 4. Humidity control with the PID algorithm (with 5-s TPO period). Temperature was controlled using the commercial controller.

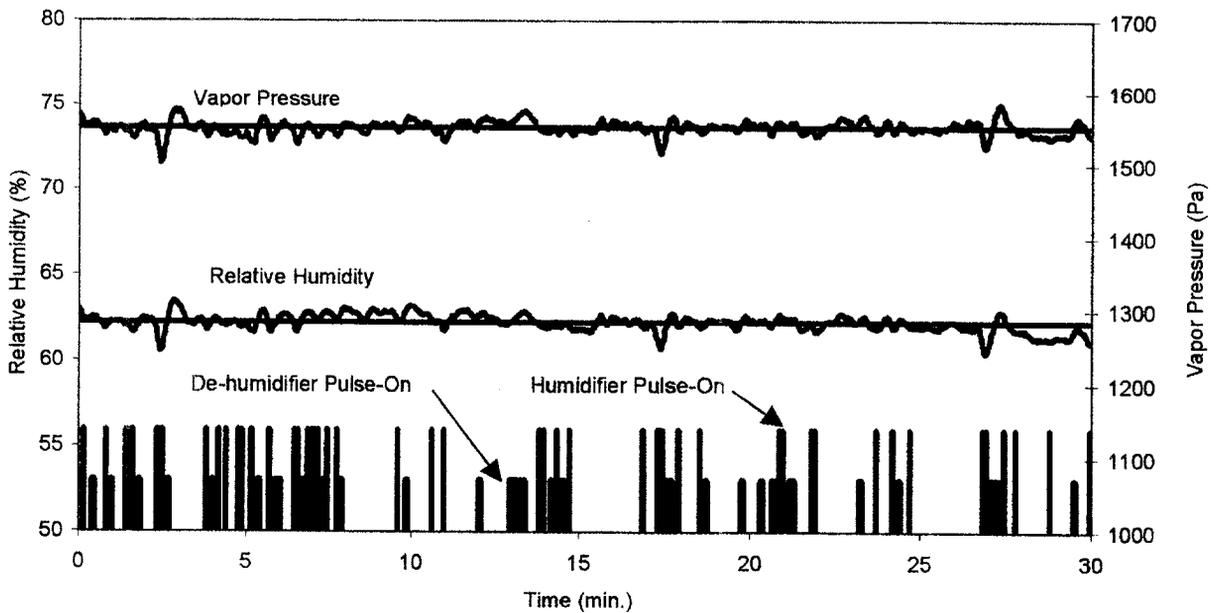


Figure 5. Humidity control using PID algorithm (with 8-s TPO period). Temperature was controlled using the commercial controller.

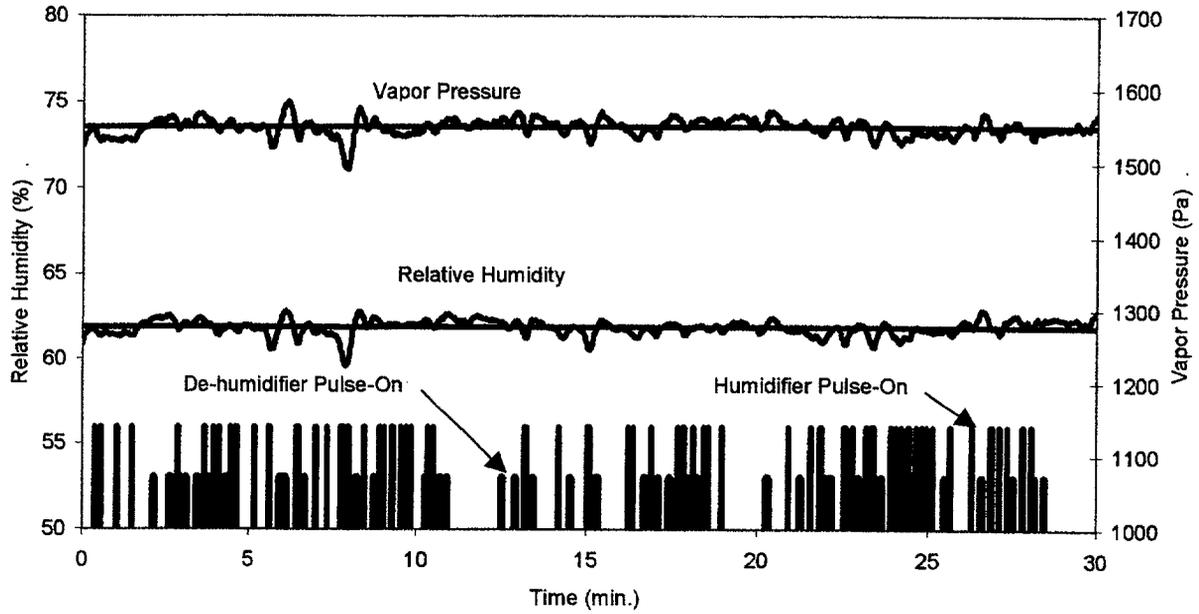


Figure 6. Humidity control using Fuzzy Logic. Temperature was controlled using the commercial controller.

