

# ProCoDA

## An Automated Method for Testing Process Parameters

Author: Monroe Weber-Shirk [mw24@cornell.edu](mailto:mw24@cornell.edu)

- [An Automated Method for Testing Process Parameters](#)
  - [Abstract](#)
  - [Introduction and Objectives](#)
  - [Procedures](#)
  - [Results and Discussion](#)
  - [Conclusions](#)
  - [Acknowledgments](#)
  - [References](#)

[word document of this file](#)

### Abstract

We describe here easily configured process control and data acquisition software and hardware that significantly enhance experimental evaluation of unit processes and that permit fully automated parametric studies. The process control software is a LabVIEW executable and incorporates sensors, states, rules, an external microprocessor for controls, the ability to link to external LabVIEW code to add new features, and data logging. The composite system is flexible and user friendly. First year undergraduates have used the software to control a bench top drinking water treatment plant and both undergraduate and graduate students have used the software to conduct parametric studies of methane generation, the activated sludge process, flocculation, and porous media filtration. The system offers the user the ability to program an entire series of experimental runs including variation of a parameter over a series of values and inclusion of replicate tests to confirm repeatability of the results. The automated parametric testing has greatly increased our ability to study the response of complex systems. Evaluation of colloidal particle flocculation and porous media filtration in bench-scale experiments are presented here as example applications of the ProCoDA software.

**Keywords:** process control, data acquisition, automation, parametric testing, flocculation, filtration

### Introduction and Objectives

Process control is commonly used in undergraduate teaching laboratories in electrical and mechanical engineering with an emphasis on the PID (proportional, derivative, integral) control algorithm (Braatz and Johnson, 1998). Process control laboratory experiences typically focus on control of a single parameter such as flow rate, temperature, liquid level, pressure (Reh et al., 1999), or motor angular velocity (Choi, 2004). Control of real processes generally requires integration of individual process control components into a larger system that includes data acquisition and data logging, as well as the concepts of process state and operating rules. Although researchers have frequently created customized process control code for specific applications we present a highly configurable system that can be used to automate the control of parametric experiments or complex processes.

Experimental work is often characterized by repetitive experiments and the need for an individual to configure each of those experiments and then measure and record the data. In the case of engineered environmental systems we are often interested in optimizing the performance of unit processes and exploring the response of processes to a range of inputs. It is often necessary to cycle a selected process through a sequence of operational stages or states, to change process related variables, and to subject the process to a range of inputs. For example, operation of porous media filtration requires implementation of backwash, optional pretreatment, and particle challenge states. Filtration process variables might include flow velocities or the duration of a state, and inputs might include variation in the type and concentration of colloidal particles or coagulant type and dose. In conventional jar tests an experimenter would evaluate flocculation efficiency based on the turbidity of settled water or measure the critical sedimentation velocity by taking a batch sample of the flocs and measuring the turbidity as the sample settles. The measurement of settled water turbidity requires first loading the sample vial with flocs, then measuring turbidity as the flocs settle, and finally purging the sample vial with clean water. Although the states to control porous media filtration and to measure the effectiveness of flocculation could be implemented by a human operator, they are well suited for automation.

We have taken the automation components that are commonly used to control industrial and municipal processes and adapted them for use in the laboratory. Our objective was to create process control software that was highly configurable and link it to sensors and outputs for control. The ProCoDA software began as data acquisition software capable of remote monitoring of experiments with easily configured sensors. The ability to control on/off devices (valves, pumps, mixers, etc) and the ability to control one continuously variable device such as a variable flow peristaltic pump was then added. The process control and data acquisition components were integrated and a user friendly programming environment that is highly configurable was created that incorporates the concepts of states, rules, constants, and variables.

Prior to creating the standalone ProCoDA application we developed a first year laboratory focused on drinking water treatment in which the students programmed the states and output settings in LabVIEW. Students spent much of their time focused on programming rather than on experimenting with the water treatment plant and the programming requirements were too advanced for first year students. From that experience it became apparent that a programming environment that was specifically designed for process control could make it easier for students to experiment with the process rather than focusing on the code.

The guiding principle in developing the ProCoDA code was to create an application that was general and easily configured to handle a wide variety of challenges. Thus the code is not designed for a particular unit process. The goal of designing code that was highly adaptable required implementing features that make it possible for users to customize the software for specific tasks. The ProCoDA software includes a built in programming environment for defining variables, constants, and sensors and for writing the logic that controls moving from one process state to the next. To extend the capabilities of the software without making the interface overly complex the ability to use external code to define new functions was added. Examples of external code that we have used to extend the capabilities of the ProCoDA software include proportional integral derivative control algorithms, acquisition of data from turbidimeters using the serial port, functions that incrementally vary a parameter over a specified range and functions that take a flow rate and tubing size and calculate the peristaltic pump speed.

## Procedures

The ProCoDA software was written using LabVIEW. The high level predefined functions, the ease of developing graphical user interfaces, and the integration with input and output devices made LabVIEW the preferred programming environment. In addition to the software we designed hardware interfaces to simplify the use of sensors and controls.

Sensor outputs and power supply inputs have been integrated in a single cable to make sensors easy to use in a student laboratory environment. The sensors use a standardized wiring scheme so that a variety of sensors can be used in a connect-calibrate-measure approach. The elimination of power supplies at each experiment location and all of the associated wiring has made sensor use easy for students. The process control software interface for sensors allows adding and configuring sensors while simultaneously collecting data.

The output control module can vary the state of on/off devices as well as control devices that require a 4-20 mA signal. The on/off outputs are 24 V with sufficient current to operate small solenoid valves or to switch relays for controlling the power to any number of devices. The 4-20 mA analog outputs are used to control peristaltic pumps. Control of digital devices has not yet been implemented; however, it is possible to add external code that would send commands to a serial or GPIB port or across the internet to control an instrument.

## Data Acquisition

The ProCoDA software is designed to receive data from two types of digital sources. Sensors or meters that have their own digital communication interface can transmit their data to the ProCoDA software through external code that is written specifically for each meter. For example this method is used to acquire data from HF Scientific turbidimeters. Sensors that can be configured to produce an analog voltage such as pressure, pH and dissolved oxygen are monitored with a National Instruments USB data acquisition board.

Sensors are wired using Category 5 or 6 Ethernet cable using our own [analog wiring standard](#). Two wires are used to carry the sensor output voltage designated as S- and S+. Five wires are used for different voltage power supplies, and one wire is used for the power supply ground. It is convenient if student laboratories are hardwired with data acquisition ports at workstations, but this is not required for use of the Process Control software. When used, each data acquisition port is wired with up to five different power supplies. Sensors are wired to connect to the power supply that they require. Thus, the sensor is powered and monitored through a single port. This scheme is used for a variety of sensors including pressure, strain gage, pH, dissolved oxygen, turbidity, and temperature. For several of the sensors the power supplies are also used to power a signal conditioning circuit to transform the sensor output into a voltage that can easily be monitored with a data acquisition system.

Wiring standard used for combining power supplies and analog data acquisition in a Category 5 Ethernet cable.

T-568B standard	T-568A standard	voltage
white/orange	white/green	S-
orange	green	S+
white/green	white/orange	ground
blue	blue	-5 V
white/blue	white/blue	+5 V
green	orange	+10V
white/brown	white/brown	-15 V
brown	brown	+15 V

## Sensors, Signal Conditioning, and Calibration

Monitoring sensors requires conversion of the measured voltage into a physically meaningful unit. The data acquisition module of the ProCoDA software uses conversion files to implement a variety of conversion algorithms including polynomials and correspondence tables as well as the calibration algorithms required for pH and dissolved oxygen. The pressure sensor conversion files make it easy to use pressure sensors to measure pressures in various physical units, to measure reactor volumes (of known cross sectional area), and flow rates (through devices with known relationships between flow and pressure drop). In addition to the application of a conversion to a physical unit it is possible to calibrate the pressure sensor output to a specific value by changing an offset.

### pH Sensors

pH sensors produce a voltage output in the range that would normally be easy to measure using standard data acquisition hardware. Unfortunately, the impedance requirement for a pH sensor is orders of magnitude higher than the inputs of standard data acquisition hardware and thus a signal conditioning circuit must be used to amplify the pH sensor output. The circuit consists of unity gain amplifiers that have less than 0.1 pA input leakage current (anonymous , 1993). A circuit diagram is available at <http://ceeserver.cce.cornell.edu/mw24/Labdocumentation/pH%20Circuit.pdf>. The ProCoDA software includes automatic pH buffer recognition and piecewise linear calibration between buffers.

### Pressure sensors



### Differential pressure sensor.

We use 1 psi (7 kPa) and 30 psi (200 kPa) pressure sensors (Omega sensor models PX26-001DV, and PX26-030DV) in our laboratory to measure water depth, reactor volumes, flow rates, and head loss. These sensors have maximum output voltages of 16.7 mV and 100 mV respectively with a power supply of 10 V. We use the differential pressure model since the sensors can be used to measure gage pressure or differential pressure. The [sensors](#) can directly measure water pressure although the electrical connections must be kept dry.

The ProCoDA software converts the voltage output from the pressure sensors into the physical units of water column height or pressure using linear conversion algorithms. The sensors can also be zeroed or set to a measured value using a one point calibration.

## Dissolved Oxygen

Dissolved oxygen probes produce a current in the pA range that is proportional to the oxygen concentration in the bulk solution. The signal conditioning circuit is designed to convert this very small current into a measurable voltage and to isolate the probe from the effects of fluctuations in the voltage level of the solution containing the probe. This isolation is critical if the solution is monitored with additional probes or if the solution is electrically connected to a building plumbing system or to any other voltage source. The circuit design is available at ([http://ceeserver.cee.cornell.edu/mw24/Labdocumentation/Dissolved\\_oxygen/circuit.pdf](http://ceeserver.cee.cornell.edu/mw24/Labdocumentation/Dissolved_oxygen/circuit.pdf))

## Temperature

We use a linear temperature sensor coupled with a simple voltage dividing circuit such as the Omega sensor model OL703. The voltage output is converted to a temperature using a linear equation.

## Controls

The ProCoDA hardware is designed and fabricated around an NI USB data acquisition board is used for on/off control of up to six devices and for variable control of up to two peristaltic pumps. The on/off devices are controlled with 24 V outputs that can be used to control solenoid valves, pinch valves, relays, or other low current devices.

## ProCoDA Software

The ProCoDA software runs five parallel processes during normal operation. The five processes are



1. Data acquisition from the ProCoDA hardware
2. Calculation of output values, processing of rules to determine state changes, sending commands to the ProCoDA hardware, and logging of state changes and the rule that caused the state to change
3. Data logging of sensor data as well as all user defined variables
4. Graphical display of all sensor data as well as all user defined variables
5. An event loop that responds to 20 user events to initiate processes such as sensor configuration, rule editing, pump calibration, and hardware selection.

The parallel processes are asynchronous and share data through the use of code objects (LabVIEW SubVIs) that contain both data and methods.

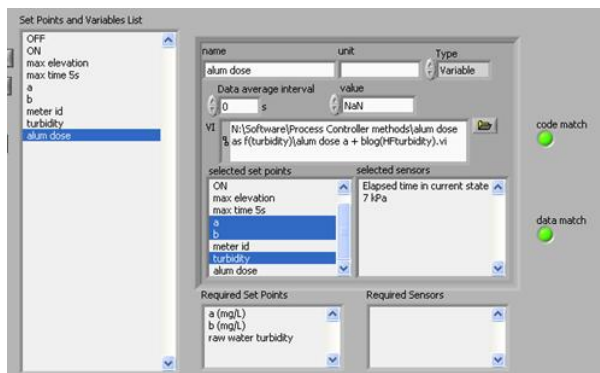
A core feature of the ProCoDA software is the user programmable feature that facilitates customization of the control logic for specific tasks. The rule editor provides a programming environment for setting up states, control logic, set points, variables defined by links to external code, and selecting which user defined parameter controls each output in each state.

## States

		output settings	output descriptions
	0	ON	V0-1
	1	OFF	V1
	2	ON	V0-2
	3	OFF	V3
	4	ON	V4
	5	OFF	Str bar
	6	OFF	
	7	OFF	
	8	OFF	
	9	OFF	
	10	OFF	
	11	OFF	
	12	OFF	
0	pump off/on	ON	
	pump cw/ccw	OFF	
	pump speed	Pretreatment water pump	clean water
1	pump off/on	ON	
	pump cw/ccw	OFF	
	pump speed	alum pump control	alum
2	pump off/on	OFF	
	pump cw/ccw	OFF	
	pump speed	acid pump control	acid
3	pump off/on	ON	
	pump cw/ccw	OFF	
	pump speed	base pump control	base
4	pump off/on	OFF	
	pump cw/ccw	OFF	
	pump speed	pump speed zero	back wash
5	pump off/on	OFF	
	pump cw/ccw	OFF	
	pump speed	OFF	claw stock

View of the controls used to set all of the Stamp® Microprocessor outputs. These controls can have different values for every state. The "output settings" in the middle column are drop down menus containing a list of all the defined constants and variables. States represent different control configurations. For example, rapid sand filters could have several operating states including normal down flow operation, backwash, and filter to waste. Each of these states has different control settings (valves are in different positions) and each of the states has different rules determining when the state ends and which state is next. The ProCoDA software has independent [control settings for each state](#). The rule editor is used to set the value of each output control. Outputs can be set either to 0  or 1  or they can be set to intermediate values. If a fractional value between 0 and 1 is assigned to a Boolean output (such as solenoid valves) the ProCoDA software will toggle between 0 and 1 to attain the target value. If a fractional value is assigned to a pump speed control the pump will operate at the fractional value of its maximum speed. The parameters that are assigned to each output can be either constants or variables.

## Rules



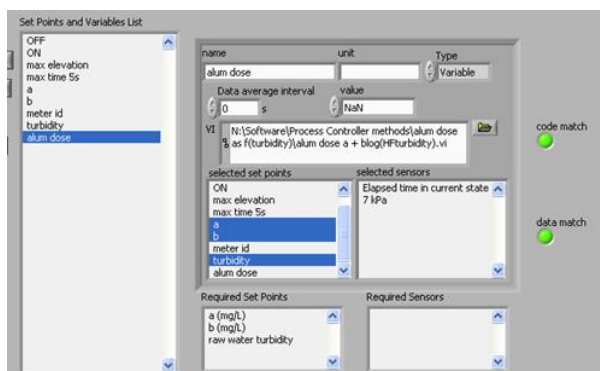
Rule that ends the state named "BW filter after challenge" when the time in that state exceeds the set point "backwash (filter) time". The rule also indicates that the next state will be "Acid wash".

Rules are used to set the exit conditions for states. A common **#exit rule** might be based on elapsed time in the current state, but it could also be based on sensor values or on a parameter that is calculated from a combination of sensors and other parameters. At a water treatment plant, backwash might be automatically initiated after 72 hours in the down-flow state, when pressure drop across the filter exceeds 1 m, or when the filtered turbidity exceeds 0.05 NTU. Each state can have multiple exit rules and the rules can have multiple conditions. The rules are executed and the output settings are updated approximately 20 times per second with the execution speed controlled by the software running on the Stamp microprocessor. Rules with multiple conditions are used when both conditions must be true simultaneously and thus create a logical "and" condition. Multiple rules are executed sequentially and the first rule that is true will cause the state to change to the state selected by that rule. Thus, multiple rules create the logical "or" condition, but with the added feature that each condition can have a different outcome. This feature is useful for handling emergency conditions that then cause the process to shutdown or for creating programs that cycle through different states depending on measured process values.

## External logic

LabVIEW block diagram showing the external code that adds two variables or set points and returns the result.

LabVIEW executables can be enabled to connect to external code. This capability makes it possible to easily extend the capabilities of the ProCoDA software. The external code must be designed to meet specific requirements for the data types of inputs and outputs. An external code interface has been created to take a variable number of numeric inputs and produce a single numeric output. The external code can be used for a wide variety of functions including simple **#math functions**, a specialized function (such as one which sets a **#coagulant dose** based on raw water turbidity proportional-integral-derivative control that can be used to force a controlled parameter to a desired set point, and data acquisition functions that acquire digital data from instruments.



Screen shot from the Process Controller showing how inputs are sent to external code. In this case the external code estimates an alum dose based on measured raw water turbidity and a simple model that relates turbidity and alum dose.

The programming environment for creating rules that determine exit conditions for states and which state to go to readily facilitates setting up the algorithms for controlling simple repetitive processes such as a sequencing batch reactors or rapid sand filters. For experimental purposes it is desirable to

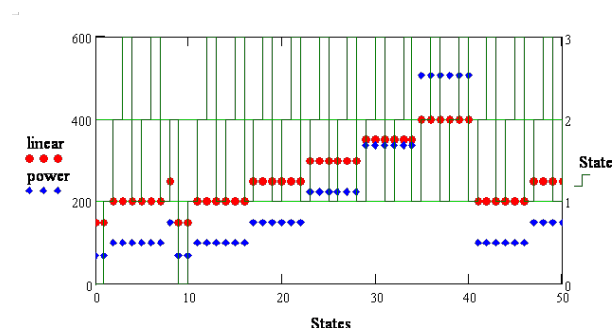
have the capability to systematically vary a parameter to test the performance of the process over a range of input values. This is accomplished via an external code that compares the number of specified replicates to a parameter that increments when the ProCoDA enters a specified state. The output parameter can be used to control pump speeds, times, or can be an input to subsequent calculations.

Incrementing functions that increment linearly and that increment following a power law relationship are as follow:

$$y_{linear} = slope \cdot x + intercept$$

$$y_{power} = coefficient \cdot base^x$$

The parameter, x, is an integer that increments from zero to a maximum value set by the user. The output parameters,  $y_{linear}$  and  $y_{power}$  could be used to vary a flow rate, a chemical dose, or any other parameter.



Increment functions showing how the parameter varies as a function of the state. In this example the state cycled between states 1, 2, and 3. The increment state was 2, the number of replicates was 2, the reset state was 0, the y intercept was 200, the slope was 50, and the maximum value of x was 4. The power law relationship used a coefficient of 100 and a base of 1.5.

The two [#increment functions](#) are dependent on the state cycles. The state was set to cycle between states 1, 2, and 3 with the exception of a manual reset to state 0 and then to state 1 to illustrate how the increment function is reset. The power law function is useful when it is desirable to explore a larger parameter space. However, care must be taken to ensure that the controlled processes have the ability to deliver the desired range of the varied parameter.

ProCoDA can be configured to stop an experiment after the reaching the maximum value of the parameter. Otherwise it will reset the parameter to its initial value and begin the increment process again. It is also possible to systematically vary more than one parameter.

## Data and process logging

ProCoDA creates two types of data files. The first column in both files is a time stamp and new files are created automatically at midnight. The datalog file contains a column for each sensor and for each variable. The datalog accumulates data at the rate set by the user. Sensor data is averaged over the data interval, but the instantaneous values are recorded for variables. The statelog file contains a row for every time the state changes. The row contains the state ID number and the name of the rule that caused the state to change. If the user caused the state to change that is indicated as well.

By using both the data and state logs it is possible to reconstruct the events that occurred during an experiment and correlate sensor and variable values with the state of ProCoDA. The data can be analyzed to troubleshoot and to determine what event caused ProCoDA to perform unexpectedly.

## Filter Test Apparatus

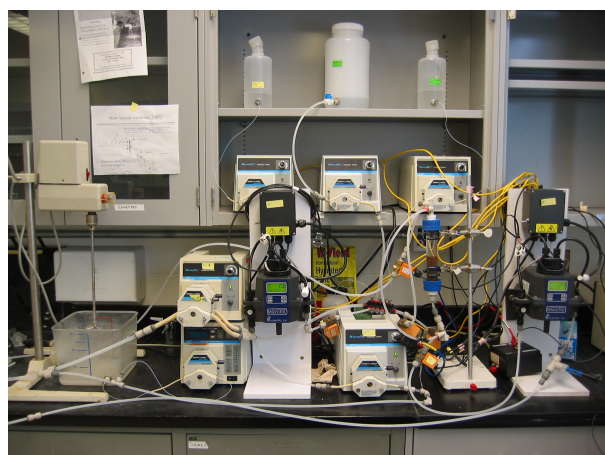


Photo of the filter test apparatus.

## Results and Discussion

Over the past 10 years ProCoDA software and hardware have been used by students to automate bench scale drinking water treatment plants, activated sludge sequencing batch reactors, temperature controllers, flow controllers, pH controllers, and a large number of other experimental configurations. These experiments were conducted as part of the Cornell University undergraduate curriculum in Environmental Engineering including the AguaClara program. In addition, the ability to automatically vary a parameter over a range of values has significantly increased our ability to characterize performance in research on unit processes for water treatment. As an example of this capability we present data from an experiment that was conducted to determine the effect of an aluminum hydroxide coating on porous media to enhance the removal of kaolin clay. The aluminum concentration in the feed during the pretreatment step was  $2.5 \times 10^{-4}$  mole/liter. The amount of aluminum applied to the filter was varied using the power law increment function to adjust the duration of the pretreatment state. Each of the pretreatment conditions was replicated and the results from the two tests were almost identical. The [data surface](#) is from the particle challenge state from the 6 treatment levels and a replicate for a total of 12 tests.

The filter performance is measured as  $pC^*$  where  $C^*$  is the turbidity of the effluent water normalized by the turbidity of the influent water and  $p$  is the  $-\log$  function. Thus a value of zero for  $pC^*$  would indicate no particle removal and a value of 1 indicates 90% particle removal. The x axis is time and data is shown between 600 and 1000 s. The data from the first residence time for the experimental apparatus are not shown since it takes several residence times to purge the filter column of the water from the previous state. The aluminum pretreatment is reported on the y axis as moles of aluminum per square meter of filter top surface area. The data clearly show that the aluminum pretreatment enhances particle capture and that at low doses the filter performance improves as the amount of aluminum applied to the filter increase. However, at the very highest dose the filter performance begins to deteriorate.

We will compare aluminum sulfate with other filter aids and discuss the significance of these experimental results in a future paper.

## Conclusions

The automation of unit processes and the integration of process control and data acquisition greatly facilitate characterization of unit process performance. The ProCoDA software makes it easy for students in laboratory courses and in research laboratories to configure and automate a wide variety of unit processes. The ability to generate reproducible data including information on the process control states greatly facilitates data analysis. The ability to collect data at higher sampling rates than possible with manual sample also increases confidence in the significance of the data. Automation of a sequence of experimental conditions and of data collection also permits researchers to conduct many more experiments than would otherwise be possible if they were performed manually.

## Acknowledgments

This work was funded by the National Science Foundation and Cornell University. The author would like to thank Leonard Lion for his thoughtful review of this paper, Po-Hsun Lin for the rapid sand filtration data, and Cameron Willkens and Paul Charles for design and fabrication of the ProCoDA hardware and the sensor signal conditioning circuits.

## References

- anonymous (1993) MAXIM 1.2 mA Max, Single/Dual/Quad, Single-Supply Op Amps. Accessed 4/16 2007. <http://www.ortodoxism.ro/datasheets/maxim/MAX406-MAX419.pdf>
- Braatz, R. D. and Johnson, M. R. (1998) Process control laboratory education using a graphical operator interface. *Computer Applications in Engineering Education* 6(3), 151.
- Choi, C. H. (2004) Undergraduate controls laboratory experience. In: *ASCE Annual Conference Proceedings*, 14425.
- Rehg, J. A., Swain, W. H., Yangula, B. P., Wheatman, S. (1999) Fieldbus in the process control laboratory - its time has come. In: *Proceedings - Frontiers in Education Conference*, 13b4-12.