PSS Process Controller and Data Analysis

Process Controller Description

The spring 2009 plate settler spacing experiments were run using the process controller software developed by Dr. Weber-Shirk. Our experimental apparatus consisted of four peristaltic pumps, two electric stirrers, one stir-plate, two turbidimeters, one pressure sensor, one solenoid valve, and one pinch valve: all of which were directly controlled by process controller.

*The process controller method for the PSS flow rate experiment can be found here

The following is an overview of how we utilized this software for our apparatus.

States, Set points, and Rules

The process controller software runs based on states, set points, and rules that are inputted by the user: *states* define the general activity of the system at any given moment; *set points* are parameters that define the *state* of the system in greater detail; and *rules* define the circumstances under which a *state* occurs.

For example, our process controller method defined two states: "Mix" and "Floc Blanket Formation". The "Floc Blanket Formation" state was characterized by certain flow rates for the influent, alum, and water through the upflow clarifier. These flow rates were the set points. An example of a rule would be:

If the time spent in "Mix" is greater than "x", go to "Floc Blanket Formation".

This rule facilitates a switch from the "Mix" state to the "Floc Blanket Formation" state based on the time set point "x".

Varying Flow Rates and Floc Blanket Height

In order to run the system continuously while varying flow rates and floc blanket height, we organized our process controller method into two loops: the first loop to increment flow rates at the low floc blanket height, and the second loop to increment flow rates at the high floc blanket height.

To do this, we used increment and counting functions. The increment function toggled between consecutive flow rates and a reset state; while the counting function regulated the number of flow rates incremented and directed the path of the system appropriately. For example, once the counting function identified that the reset state had been employed four times, the system was sent to a new state after the fifth flow rate.

Data Analysis Method

The data for this experiment was analyzed using a series of Matcad data processing code written and developed by Ian Tse and Dr. Monroe Weber-Shirk. Three mathcad files were used along with a excel spreadsheet called Meta File. The Mathcad code has the capability to consolidate and filter data collected in process controller over a period of several days. The Meta file contains all of vital information that must be inputted into the mathcad file by the user, including the dates over which the experiment occurred, the particular process controller state that indicates the beginning of an experiment, the location of the files, and an ID tag which allows the user to easily access processed data. First enter the ID tag, date, and experiment start state into to the Meta file and Data Preprocessor file.This indicates the location of the files, and by running the preprocessor the user can extract the statelog start row and number of runs from the Data Preprocessor file and input the values into the Meta file. After the Meta file is complete the Meta Data Processor can be run by entering the metaID and the statID.
The output is a graph of the effluent turbidity vs. time and a graph of average effluent turbidity vs. critical velocity as seen in the results section.

The Data Processor files can be found below
Meta File
Meta Data Processor Mathcad File
Data Preprocessor Mathcad File
Data Processor Functions Mathcad File