Open Source Dataset Generator for Data Analytics

Patrick J Dixon, PE, PMP: Vice President of Automation, Pulmac Systems
Alia Rezvi: Student/Software Developer, City University of London

Abstract

Data analytics requires a dataset to provide the data that will be analyzed. The most common source of a dataset is from a real time connection to data historian or an exported file from a historian. When data analytics is used for training, testing, or demonstration purposes, the following challenges have to be overcome:

- Requiring an industrial process and a control system with a historian is prohibitive for schools, vendors, and other such users
- Obtaining a dataset from an industrial firm is difficult due to proprietary intellectual property
- Data from an industrial process may lack sufficient excitation in variables to perform accurate analysis

The authors have developed a general-purpose industry wide dataset generator tool to generate a dataset. The specific application shown in this paper is for modeling paper final product quality (principally strength properties such as tensile, tear, burst, etc) with process data, quality control system (QCS) data, and pulp quality data as inputs and lab samples as the modeled properties. However, this tool can be customized and configured to generate simulated process data for any industrial process. The benefit of this tool includes:

- Reduce the need to obtain proprietary data from an industrial process
- Facilitate training and education of students in data analytics
- Testing data analytics techniques and model prediction
- Demonstration of data analytics and automation solutions
- Reduce the time to generate a dataset from months to hours

This dataset generator tool is an open-source tool for anyone to use free of charge.

Introduction

Imagine you are one of the following:

- A student or young engineer that wants to learn data analytics
- An instructor that wants to teach students and young engineers data analytics
- A vendor that wants to develop a solution using data analytics, test the solution, and demonstrate it
- An engineer for a manufacturer that wants to develop a methodology for performing data analytics using various techniques

In all of these cases, there is a fundamental requirement: process data. However, there are some challenges to obtaining process data:

- Unless you work for a manufacturer, you don't have an industrial process. Therefore, getting process data means asking a manufacturer to give you theirs. Manufacturers regard their data as intellectual property and are not looking to hand it out to others.
- Even if you have process data, it may not have sufficient excitation of the process to yield useful prediction models or analysis. Some industrial processes are single setpoint dominant; they run the same way all the time. In processes with grade changes, often the changes are occurring simultaneously and therefore are not decoupled. Coupled changes make it impossible to determine the independent magnitude and nature of these changes. To yield results that are useful, there must be sufficient independent movement of the variables of interest that significantly exceed the level of process noise.
- Typical datasets from an industrial process contain several months of data. If step tests are performed to provide decoupled responses, the time to conduct step tests for each variable of interest and wait for those responses to settle to steady state can represent an enormous amount of time. Obtaining a dataset from an industrial process can be a very significant investment of time.

If the use case is to perform data analytics or predictive modeling for an industrial process, of course the actual data needs to be used. The authors do not suggest actual process data can be replaced with artificially generated data. However, in the use cases considered in this paper, actual process data is not required.

The goals of this project were as followed:

- Provide a general-purpose tool that can be customized and configured to produce a dataset representative of an actual industrial process
- Reduce the time to generate that dataset from the months required for actual data to hours for simulated data
- Provide the tool as open source, which can be obtained, customized, and improved at no cost.

Material and Methods

Below is background and details of the development of this dataset generator tool.

Prior Work

While it is unknown if any such dataset generator tool has been previously developed, there is prior work that is helpful in the development of this tool.

The dataset generator relies upon a general understanding of first principles for modeling the outputs. In some cases, there are well established first principle models that can relate process inputs to outputs. In other cases, there are no explicit first principle models that can predict the process parameters, yet there are rules of thumb or a general understanding of which direction a process parameter will move and the shape of that move.

If we use the example of a paper machine, Smook (1) presents curves showing the relationship of several strength properties to refining. These curves and others are helpful in understanding the nature of curves that the dataset generator needs to accommodate.

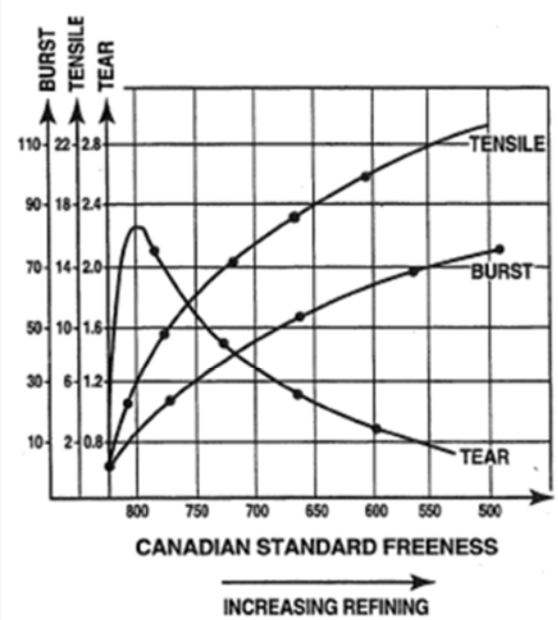


Fig. 1.16. Influence of refining on paper properties (from Handbook for Pulp and Paper Technologists by G. A. Smook)

From the author's experience as a paper science and engineering student at Miami University, the course materials for "Introduction to Paper Properties: PPS 102 Reading Materials" (2) contain similar curves and tables that help establish relationships that can be used when configuring the dataset generator for a specific application. The following figures and tables are further examples of this class of helpful general relationships.

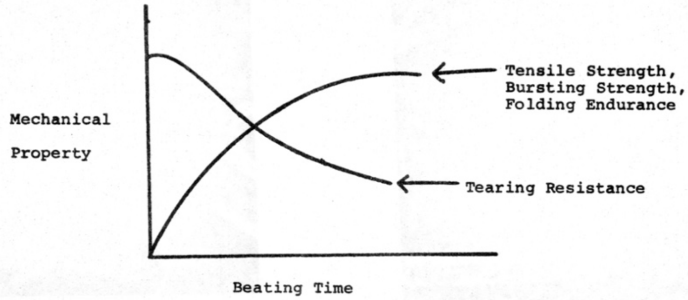


Figure 7.12 Effect of beating on the mechanical properties of paper. The plot illustrates the point that tearing strength is decreased by beating while tensile strength, bursting strength and folding endurance all increase up to a point. Consequently, the papermaker must strike a compromise between the levels of these mechanical properties that he achieves in his product.

1. Fiber shortening.

2. Removal of layers of fibrils from the outer part of the fiber wall with the formation of fines.

3. Roughening and loosening of the fiber surface -- referred to as fibrillation.

4. Breaking of intra-fiber bonds between various fiber wall layers with their replacement by water molecules within the fiber wall. This leads to a swelling and plasticizing of the fiber.

Figure 7.13 Major effects of beating and refining on fiber morphology.

Independent Variables

	Increased Moisture Content	Increased Refining	Increased Pressing	Increased Surface Size	Increased Calendering	Increased Long Fiber-to-Short Fiber Ratio	Increased Basis Weight	Increased Titanium Dioxide
Tensile Strength	-	+	+	+	0	+	+	-
Tearing Resistance	+	-	-	-	0	+	+	-
Bursting Strength	-	+	+	+	0	+	+	-
Folding Endurance	+	+	+	+	0	+	+	-
Stiffness	-	∓[a]	∓[a]	+	-	+	+	-
Opacity	0	-	-	-	0	-	+	+
Brightness	0	0	0	0	0	0	0	+
Thickness	+	-	-	0	-	+	+	0
Water Permeability	0	0	-	+	0	0	-	-
Air Permeability	-	-	-	-	-	+	-	-
Oil Permeability	-	-	-	-	-	+	-	-
Smoothness	-	+	+	+	+	-	0	-

[a]The affects on stiffness of pressing and refining cannot be predicted. Both lead to increased interfiber bonding and decreased thickness, which have opposing influences on stiffness.

Figure 7.18. Qualitative interrelationships between paper properties. The plusses, minuses and zeros indicate increases, decreases and no change, respectively, in the dependent paper properties as a result of a change in the papermaking process or paper components. The table refers to a single grade of paper. Only small changes in the independent variables are intended. The qualitative relationships that exist between common paper properties can be observed by studying the changes indicated within a given column.

Explicit models of tensile strength of paper were developed by Page (3). Such a model does not use inputs that are measurable with online instrumentation, and therefore would not appear in a historical dataset. Relative bonded area (RBA) is an elusive property that would not be in a dataset. Below are examples of such relationships.

If we assume that all fiber-to-fiber bonds act cooperatively along the length of a fiber, the bond strength β is given by,

$$\beta = bP\frac{L}{4}(\text{R.B.A.}) \qquad (9)$$

where

b = shear bond strength per unit bonded area
P = perimeter of the fiber cross section
L = fiber length (and hence $L/4$ is the mean pulled length)
R.B.A. = relative bonded area of the sheet

Combining Eqs. 7, 8, and 9, we arrive at final equations for the tensile strength of paper.

$$\frac{1}{T} = \frac{9}{8Z} + \frac{12A\rho g}{bPL(\text{R.B.A.})} \qquad (10)$$

or

$$T = \frac{8ZbPL(\text{R.B.A.})}{9bPL(\text{R.B.A.}) + 96A\rho gZ} \qquad (11)$$

This may be expressed further in a form that will not be used in this paper but may be of value elsewhere.

$$\frac{1}{T} = \frac{9}{8Z} + \frac{12g}{bL\alpha} \qquad (12)$$

where

α = the bonded area per gram of fibrous material

There are several other references to paper strength models using hardwood and softwood (4,5) and eucalyptus kraft pulp tensile (6) that are helpful in understanding the first principle nature of models to include in a dataset generator.

These are among the resources that helped formulate the design for the dataset generator. As explained previously, the dataset generator is not specific to a paper machine example. It can be customized and configured as general purpose for any industrial process. The prior work combined with industry experience enabled a design for the dataset generator that would be applicable for general purpose.

Steady State Models

A fundamental design component is the means of applying steady state process gain. Steady state gains are not time dependent; they show the magnitude of change that will occur when a model input yields its final resting place. The work cited above helped formulate the categories of gain expressions/curves to include in the dataset generator.

It is important to note that these relationships are not assumed to be linear. Doing so would greatly simplify the scope of work, yet it would not yield the desired accuracy and conform to established first principles.

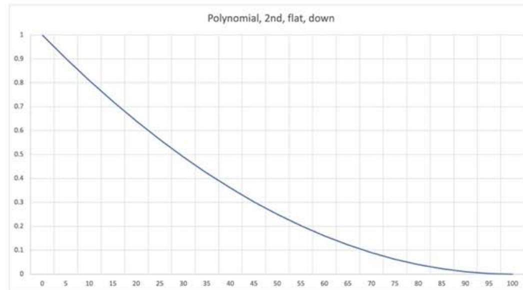
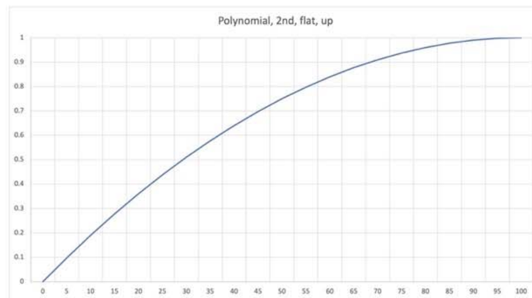
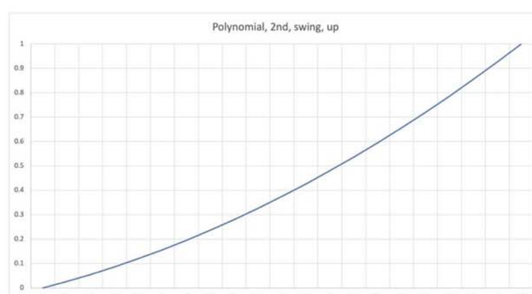
The plot of gains as a function of the input range are referred to as sensitivity plots. These sensitivity plots used a scaled input range of 0% to 100% and produce a gain ranging 0 to 1 of the output range. In other words, a gain of zero would produce the lowest possible range of an output and a gain of one would produce the highest possible range of the output. In this way, all inputs and outputs are normalized, and the sensitivity plots can be compared.

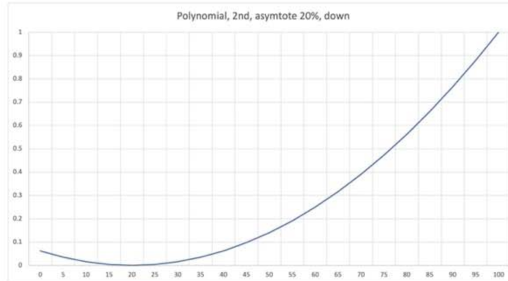
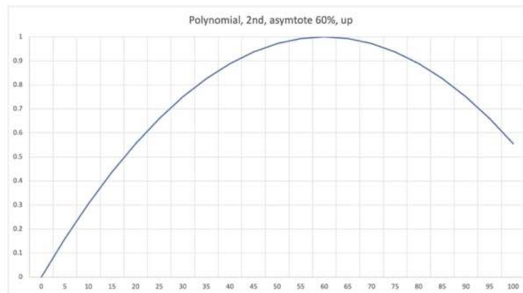
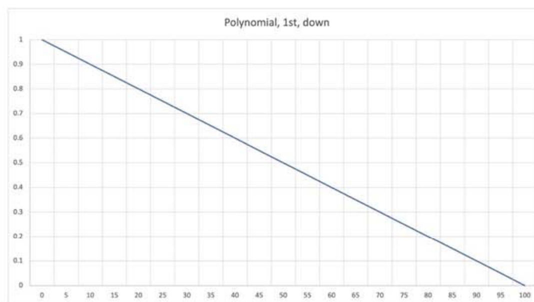
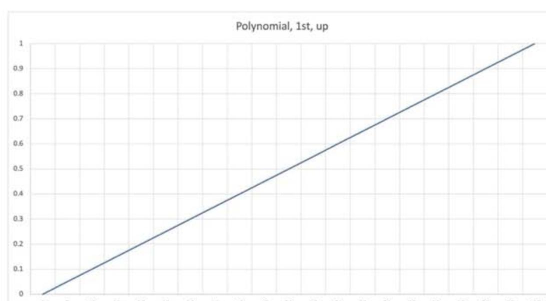
Gain class: Polynomial

The simplest class of gains is a second order polynomial. Rarely would a steady state gain in an industrial process require a polynomial form higher than second order. A first order (linear) gain is easily obtained by setting the second order term G2 to 0. This form can yield asymptotic as well as inverse response gains using appropriate settings of the terms G2, G1, and G0. For inverse gains, an asymptote can be specified to locate the inflection point in % of input range. A shape designated as flat will flatten out as the input approaches 100% of range. A swing shape will increase slope as the input approaches 100% of range. An asymptote shape will have an inflection point specified by the asymptote value. Below are tables showing configuration terms and the resulting curves in the sensitivity plots.

POLYNOMIAL

$$OUTPUT = G2 * GainInput^2 + G1 * GainInput + G0$$

GainModel	Order	GainShape	Direction	Asymtote	Slope	OUTPUT
0 Polynomial	2	0 Flat	0 Down			
0 Polynomial	2	0 Flat	1 Up			
0 Polynomial	2	1 Swing	0 Down			
0 Polynomial	2	1 Swing	1 Up			

GainModel	Order	GainShape	Direction	Asymtote	Slope	OUTPUT
0 Polynomial	2	2 Asymtote	0 Down	20%		
0 Polynomial	2	2 Asymtote	1 Up	60%		
0 Polynomial	1		0 Down			
0 Polynomial	1		1 Up			

Gain class: Exponential

Exponential relationships that use Euler's constant "e" (or exp() in an Excel function) are very common in first principle models. A benefit of this form is that it is easy to shape the curve with a slope parameter, which makes it more general purpose than a polynomial form. This can be seen in the chart below comparing the shape of 1st order up swing curves with slopes of 1 and 10. Setting the order to 2 allows a Gaussian shaped response with an asymptote specified in % of input range.

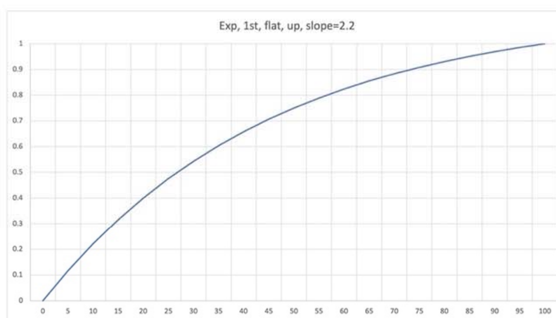
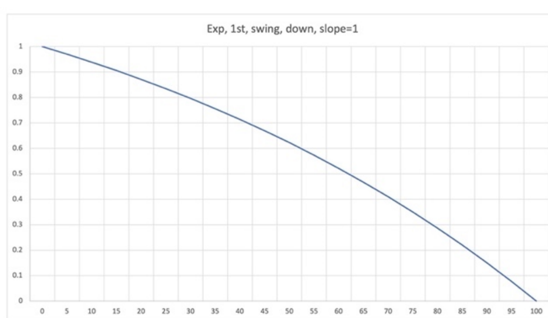
EXPONENTIAL

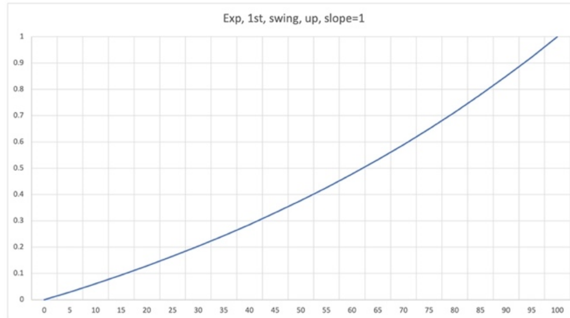
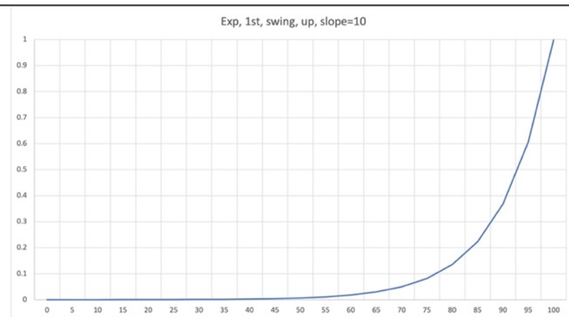
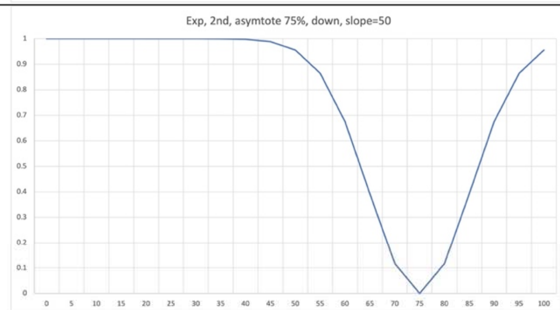
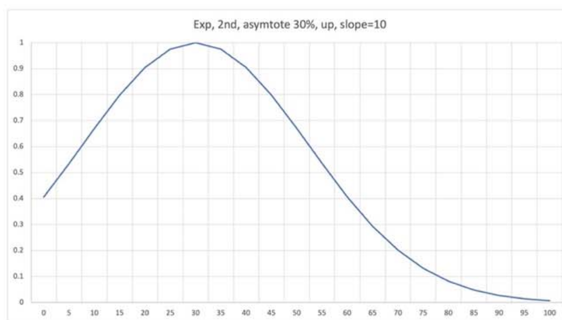
$$ExpNumerator = = exp(Slope * SlopeSign * (GainInput - GainAsymtote)^{Order}) - 1$$

$$ExpDenominator = \exp(Slope * SlopeSign) - 1$$

$$ExpFraction = \frac{ExpNumerator}{ExpDenominator}$$

$$OUTPUT = GainDirection - (2 * GainDirection - 1) * ExpFraction$$

GainModel	Order	GainShape	Direction	Asymtote	Slope	OUTPUT
1 Exponential	1	0 Flat	0 Down		2.2	
1 Exponential	1	0 Flat	1 Up		2.2	
1 Exponential	1	1 Swing	0 Down		1	

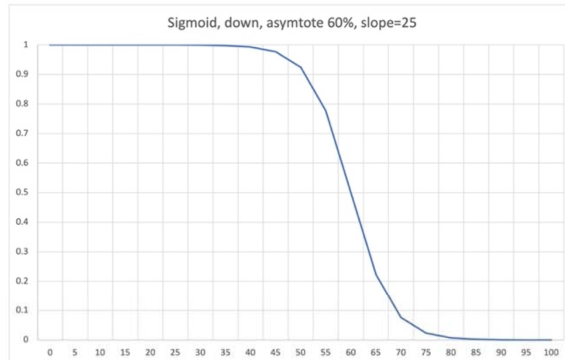
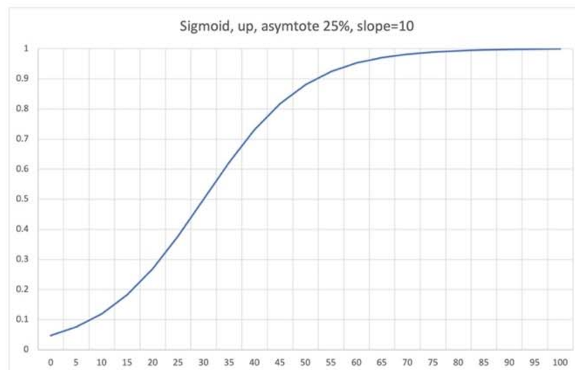
GainModel	Order	GainShape	Direction	Asymtote	Slope	OUTPUT
1 Exponential	1	1 Swing	1 Up		1	
1 Exponential	1	1 Swing	1 Up		10	
1 Exponential	2		0 Down	75%	50	
1 Exponential	2		1 Up	25%	10	

Gain class: Sigmoid

The sigmoid shape is pertinent to pH. As in prior gain classes, the asymptote specified in % of input range can locate the inflection point of the curve. The slope parameter determines whether the curve has a sharp or gentle (near linear) curve.

SIGMOID

$$OUTPUT = 1 - \left[Direction - \frac{2 * Direction - 1}{1 + e^{-1*Slope*(GainInput - GainAsymtote)}} \right]$$

GainModel	Order	GainShape	Direction	Asymtote	Slope	OUTPUT
2 Sigmoid			0 Down	60%	25	
2 Sigmoid			1 Up	25%	10	

Dynamic Model

The dataset has dynamic responses to realistically yield the process deadtime and lags (time constants) of an industrial process. These dynamics are specified for each input. It is assumed all outputs in the dataset are at the same endpoint of the process, so the input dynamics are specified by their dynamic response to the process endpoint. The example of a paper machine, the endpoint would be the reel.

The steady state models listed above yield a gain that is multiplied by a dynamic model with deadtime and second order lags, as shown below:

$$Y = Gain(x) \frac{e^{-sd}}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

$$s = LaPlace\ operator, dynamic\ input$$
$$d = Deadtime$$
$$\tau_1 = First\ order\ lag, first\ time\ constan$$
$$\tau_2 = First\ order\ lag, second\ time\ constant$$
$$x = Input$$
$$Y = Output$$

Configuration

Configuration of the dataset generator consists of the following parts:

Inputs

A list of process inputs is specified along with their ranges and the process area that identifies the dynamic model (deadtime and lags) associated with the response of the output variables to a change in the input. This list of inputs can exceed the number of inputs required to model the outputs. In data analytics, an important part of the process is to narrow down the number of input variables required. This design allows for created superfluous inputs that the data analyst would need to identify and remove from models.

Inputs also specify an order in which they are moved. The dataset generator allows configuration of stair-step independent (de-coupled) movement of inputs. This is to ensure sufficient excitation of the process for each input.

Coupled movements can also be configured for combinations of inputs. This is especially useful to generate a validation dataset for verification of the model derive from training data, which would be the de-coupled data generated above.

Noise ranges can be configured for inputs to give realistic looking process data.

State variables

There are often state variables in data that can be measured in the process but are the result of input settings. An example is the QCS value for basis weight on a paper machine. Process inputs such as stock flow, consistency, machine speed, and moisture can be used to calculate basis weight. The dataset generator allows custom coding for the mass balance calculations to yield basis weight. Likewise, the QCS moisture is a state variable resulting from the same process inputs with the addition of dryer steam pressure and assumptions about drainage on the wire and the impact of press load, which is another input. These state variables will obviously be coupled to the inputs used to calculate them, but this is a realistic challenge to using actual process data in data analytics.

In a similar way to the inputs, the state variables are listed, and ranges are specified with configurable noise ranges. There is no order required to step these as inputs since they are the result of calculations in which inputs are already moved.

Outputs

Outputs typically represent lab samples, which are the result of process inputs and state variables. It is assumed that these values are time stamped at the endpoint of the process at the time the samples were collected. Ranges and noise settings can be configured for each output.

Sample times

Not every item in a dataset has the same sampling frequency. Process data from a programmable logic controller (PLC) or distributed control system (DCS) controller/field device can have sampling frequencies of 1 second to 5 seconds. QCS scan averages typically have sample frequencies of 1 minute. Online sampling devices can have sample times that vary, typically from 5 minutes to 15 minutes. Lab samples can be infrequently collected, but 20 minutes to 30 minutes may be typical for a paper machine.

The dataset generator allows specifying different sample periods for different classes of variables.

Gain class

The curves shown previously for the polynomial, exponential, and sigmoid gains can be configured for each output and input. Selection of the gain class and the parameters that shape the curve can be specified. The user can determine which of the configured outputs apply to each output and can independently configure the gain relationships. The result is a full multiple input multiple output (MIMO) model, with the dynamics already specified for each input.

Initial Development

After initial development of the tool in Excel using Visual Basic (VBA), the 547 lines of VBA code took nearly 19 hours to complete the production of a dataset for 27 inputs, 15 state variables, and 8 outputs with 10 uncoupled moves. The image below shows that the DynamicInputs routine took over 3 hours and the CalcLab routine took over 9 hours to complete. The configuration of the tool consisted of data entry into sheets and cells in Excel, which can be an error prone user interface. Additional outside resources were sought to help improve the efficiency and user interface for this tool.

VALIDATION		AllModules				
START_DATE	1/1/21					
SPARSITY	75					
PROCESS_PERIOD (sec)	5	CreateSheets	START_CreateSheets	2/1/23 18:57:42	DURATION	
QCS_PERIOD (sec)	60		END_CreateSheets	2/1/23 18:57:43		0:00:01
LAB_PERIOD (sec)	1200					
PULPEYE_PERIOD (sec)	300	CreateInputs	START_CreateInputs	2/4/23 7:09:51	DURATION	
NUM_INPUTS	27		END_CreateInputs	2/4/23 8:42:27		1:32:36
MAX_DEADTIME (sec)	1800					
MAX_LAG1 (sec)	900	CreateValidation	START_CreateValidatior	2/1/23 21:25:46	DURATION	
MAX_LAG2 (sec)	300		END_CreateValidation	2/1/23 21:39:17		0:13:31
MAX_LEAD (sec)	0					
Added settle time (sec)	10000					
MAX_SETTLE (sec)	13000	CreatePulpEye				
INPUT_SETTLE (sec)	13000					
COUPLED_MOVES	0	CalcStateVariables_In	START_CalcStateVariabl	2/4/23 8:42:27	DURATION	
UNCOUPLED_MOVES	10		END_CalcStateVariable_	2/4/23 9:32:32		0:50:05
NUM_STATE	15					
NUM_OUTPUTS	8	DynamicInputs	START_DynamicInputs	2/4/23 9:32:32	DURATION	
TRIM (ft)	20		END_DynamicInputs	2/4/23 12:48:01		3:15:29
DRAW	1.1					
LASTUNCOUPLEDROW	774802	CalcStateVariables_Dyn	START_CalcStateVariabl	2/4/23 12:48:15	DURATION	
			END_CalcStateVariable_	2/4/23 13:37:41		0:49:26
		CalcQCS	START_CalcQCS	2/4/23 13:37:41	DURATION	
			END_CalcQCS	2/4/23 13:56:46		0:19:05
		CalcLab	START_CalcLab	2/4/23 13:56:46	DURATION	
			END_CalcLab	2/4/23 23:19:36		9:22:50
		CopyToVOA	START_CopyToVOA	2/7/23 15:16:10	DURATION	
			END_CopyToVOA	2/7/23 16:10:29		0:54:19
		CreateDataset	START_CreateDataset	2/7/23 16:55:01	DURATION	
			END_CreateDataset	2/7/23 18:32:11		1:37:10
						18:54:32

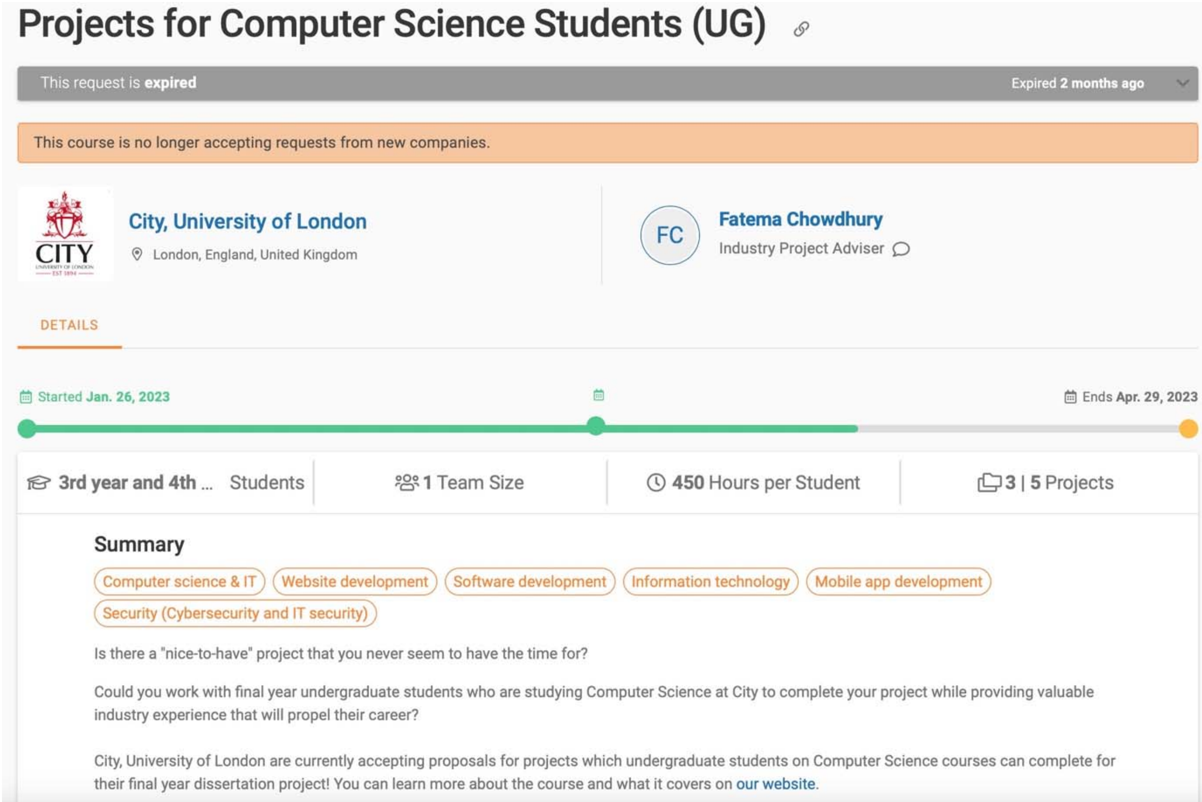
Collaboration with Universities

It was desirable to collaborate with universities in the development of this tool. Engineering students often seek senior design projects or equivalent experience as part of their study. Several universities were solicited in which there was a direct relationship, but no programs were found that expressed interest or commitment to the project.

A service called Riipen (https://www.riipen.com) was discovered which lists projects sought by universities in which an industrial partner can provide collaboration and guidance. They describe their service as:

> "Immersing students in industry projects equips them with work-ready skills. Riipen brings industry and academia together, with real company projects. Projects are embedded directly into curriculum or completed as remote internships."

The project opportunity was posted and submitted to 11 programs that were pertinent to software development. Through this process, a student named Alia Rezvi from City University of London was selected to work on the project.



Project opportunity in Riipen.com from City, University of London

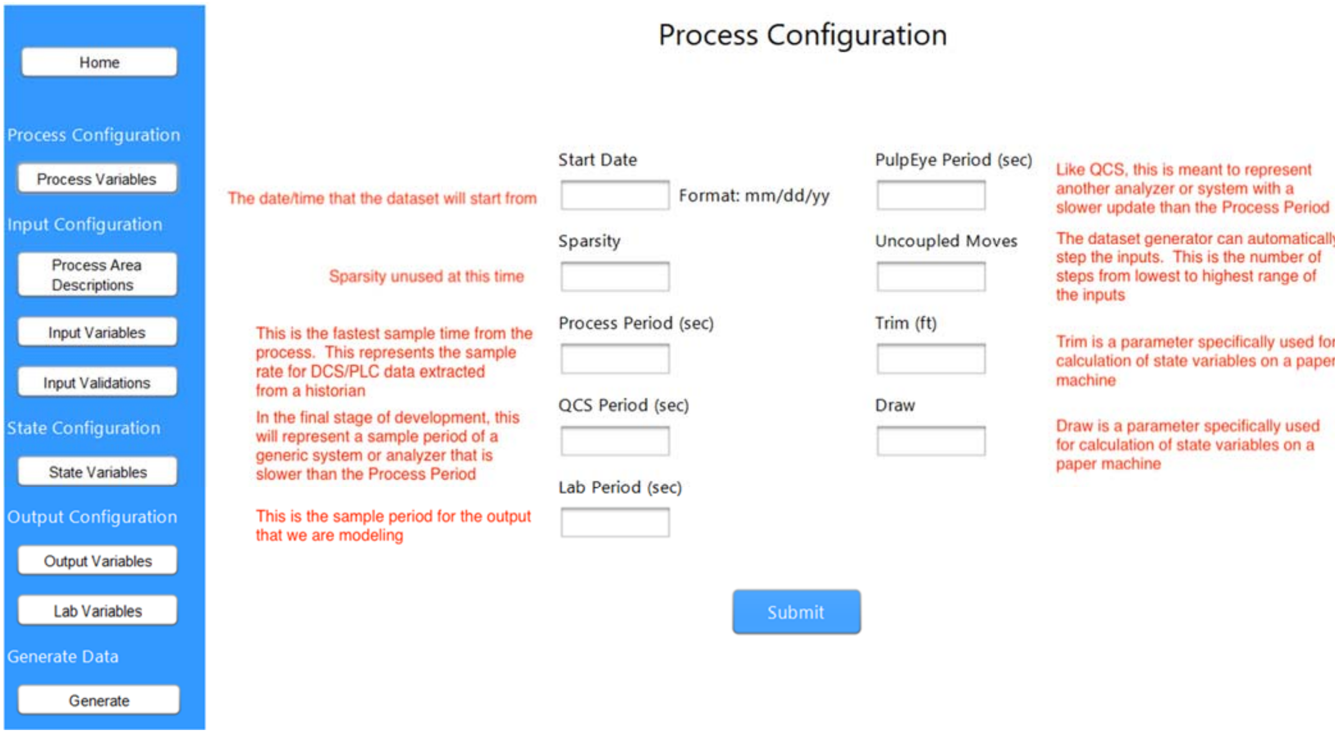
Alia Revzi, student, City University of London

Results and Discussion

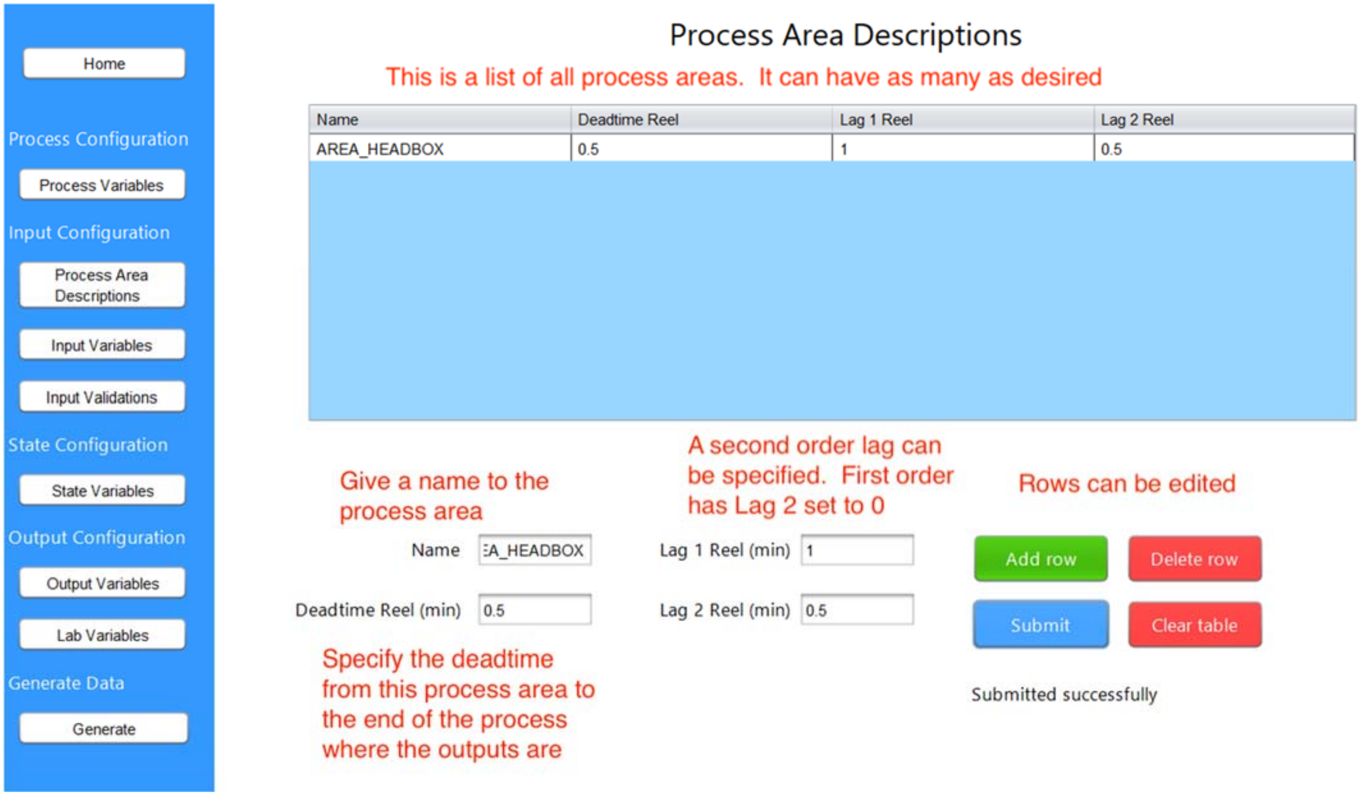
Development

Alia Rezvi began work Feb 9, 2023 to develop the dataset generator. Java was chosen as the programming language due to its ability to run in multiple platforms and its familiarity and popularity in software development. Agile project management techniques were used to create specifications and track progress.

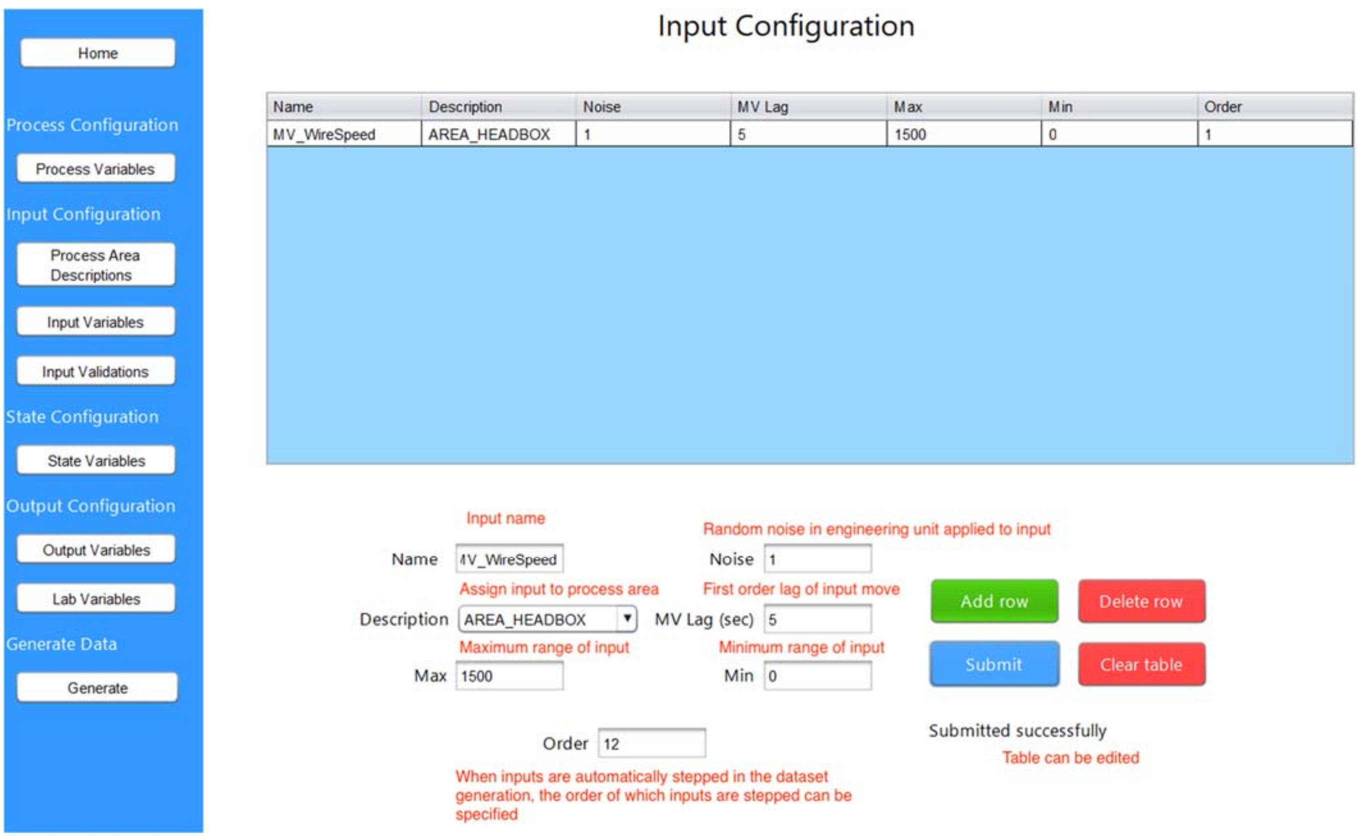
At the time of this writing, the user interface has been partially developed. The overall process configuration specifies parameters that will be used for all inputs/state variables/outputs.



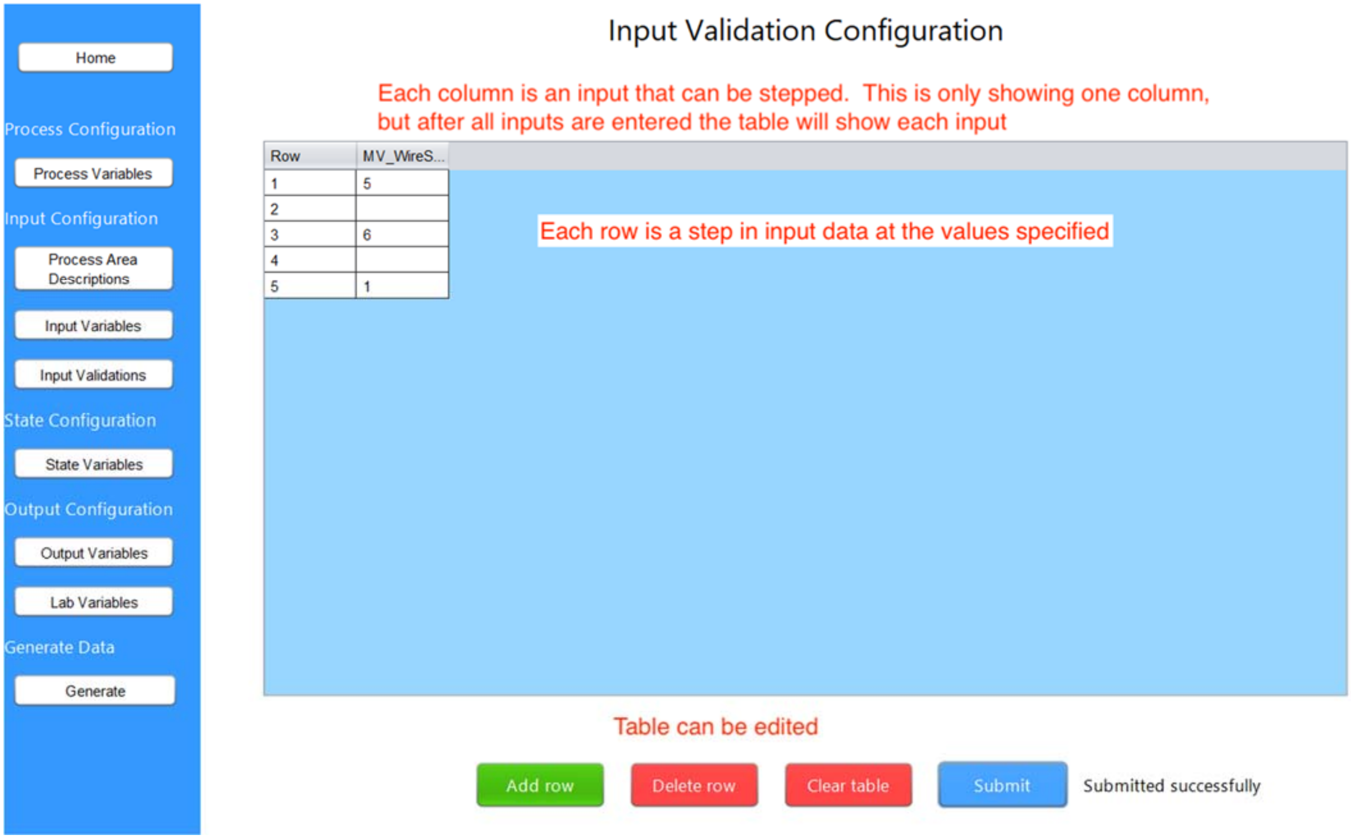
Process areas can be specified so that the process dynamics can be applied. These process areas will be applied to inputs and state variables when they are configured so that the dynamic second order plus deadtime relationship to the outputs can be specified.



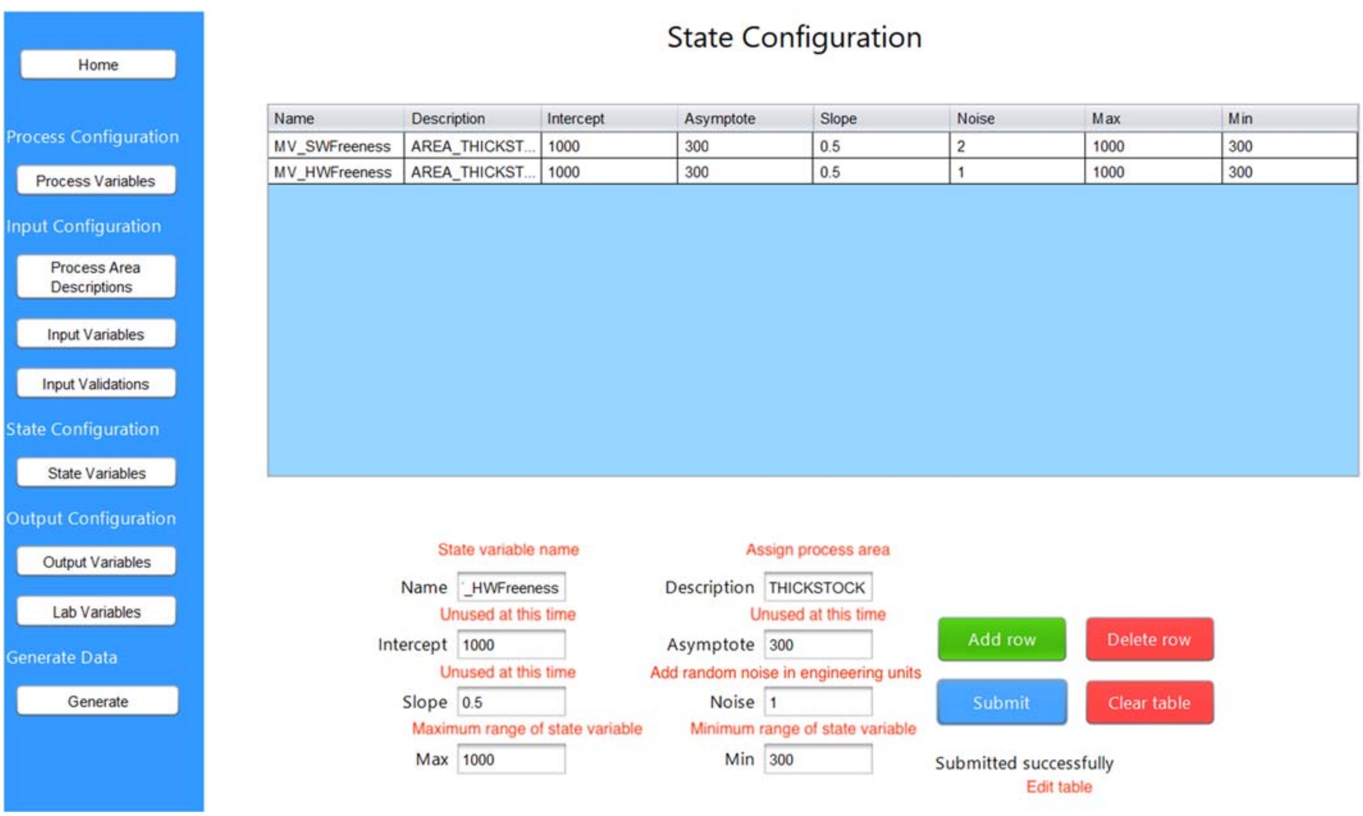
A table of inputs is configured as shown:



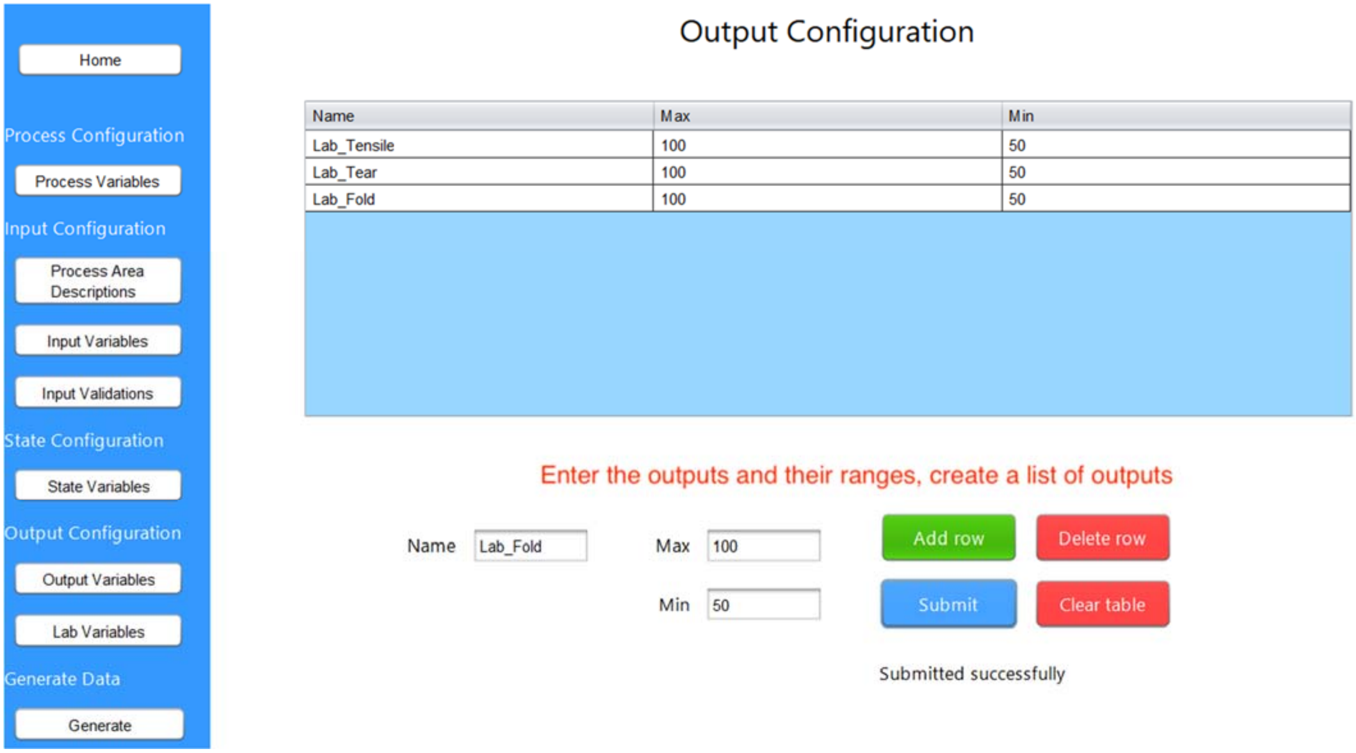
After inputs are stepped with uncoupled moves, these configured custom moves will be generated. This allows the user to create coupled moves or validation data for prediction models.



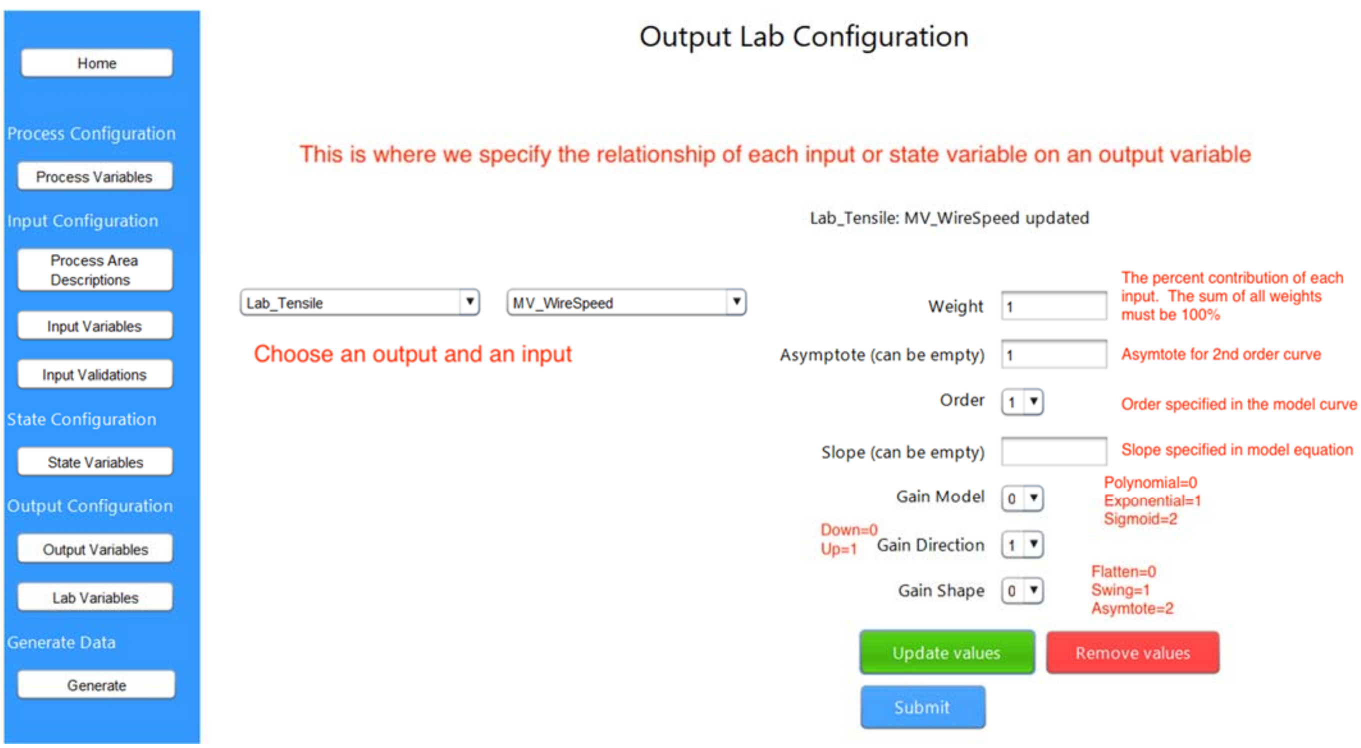
A table of state variable is configured as shown. State variables use custom code, so the configuration could be through a user interface or hard coded.



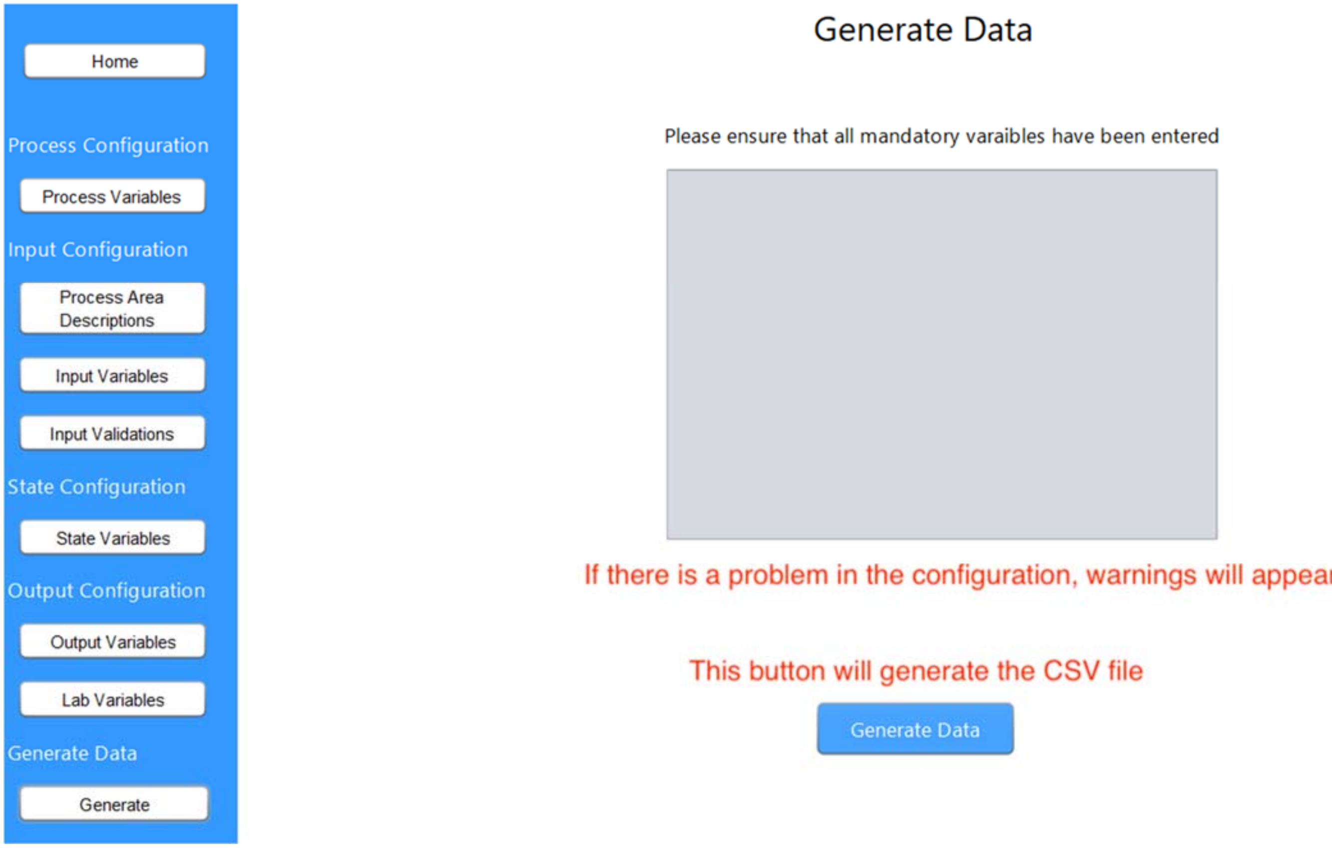
The list of outputs and their ranges are then specified.



The model configuration for the outputs uses the Polynomial, Exponential, or Sigmoid functions to specify the steady state relationships.



After all configuration is complete,



After completion of the data generation in Java code, the application ran dramatically faster. Below are performance results on 2 different platforms for identical data configurations:

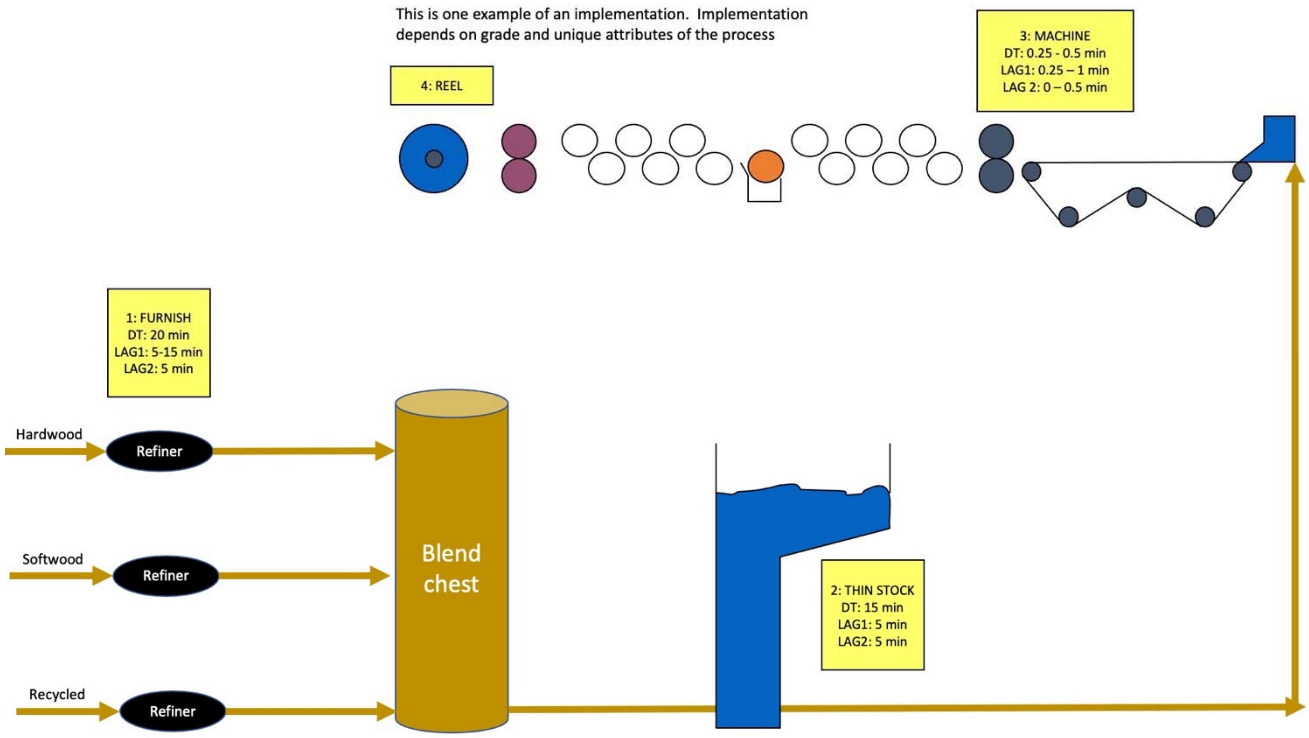
SYSTEM	MEMORY	PROCESSOR	TIME: Excel/VBA	TIME: JAVA
MacBook Pro (16 inch, 2019)	16 GB (2667 MHz DDR4)	2.6 GHz 6-Core Intel Core i7	Almost 19 hours	Not yet tested
Amd Am4 Gen3 (custom built)	16 GB	AMD Ryzen 5 5600X 6-Core Processor 3.70 GHz	About 3 hours	About 9 minutes

The application is available on GitHub. This allows anyone to download it, run it, modify the source code, and maintain revision control. The location for the application is:

https://github.com/Paramount10/dataset-generator.

Example Process: Paper Machine

The initial process of interest was to generate a dataset for a paper machine. The machine would have lab samples for optical and strength properties, and the dataset would be used for modeling these relationships. The process was configured with 4 process areas, depicted below with their associated process dynamics from their point in the process to the process endpoint, which in this case is the reel.

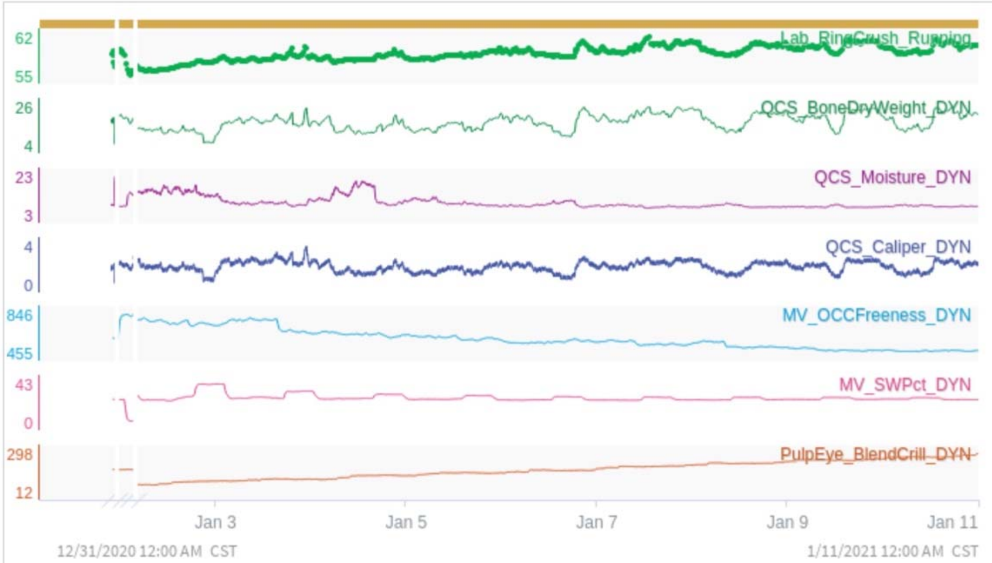


As mentioned previously, this dataset consisted of 27 inputs, 15 state variables, and 8 outputs with 10 uncoupled moves. In addition, there were 40 coupled moves of inputs to generate a validation dataset. The outputs consisted of the following lab samples:

- Opacity
- Brightness
- Tensile
- Burst
- Tear
- RingCrush
- Stiffness
- Fold

Data Analytics

The data analytics solution Seeq was used to import the dataset for analytics work. The objective was to see if the dataset was a realistic representation of process data to demonstrate the development of prediction models for the lab properties. Dynamic adjustment of data was configured to align data in time for model identification. Since the dataset included data that had outliers, downtime, and intentionally erroneous calculations, conditions were configured to identify these time periods and remove data that would not be useful for modeling. An example of a portion of the data used for a ring crush model is shown:



As a result of pre-processing data and configuration of prediction models, a set of inputs was derived for each output. Below is an example of the inputs associated with the ring crush model.

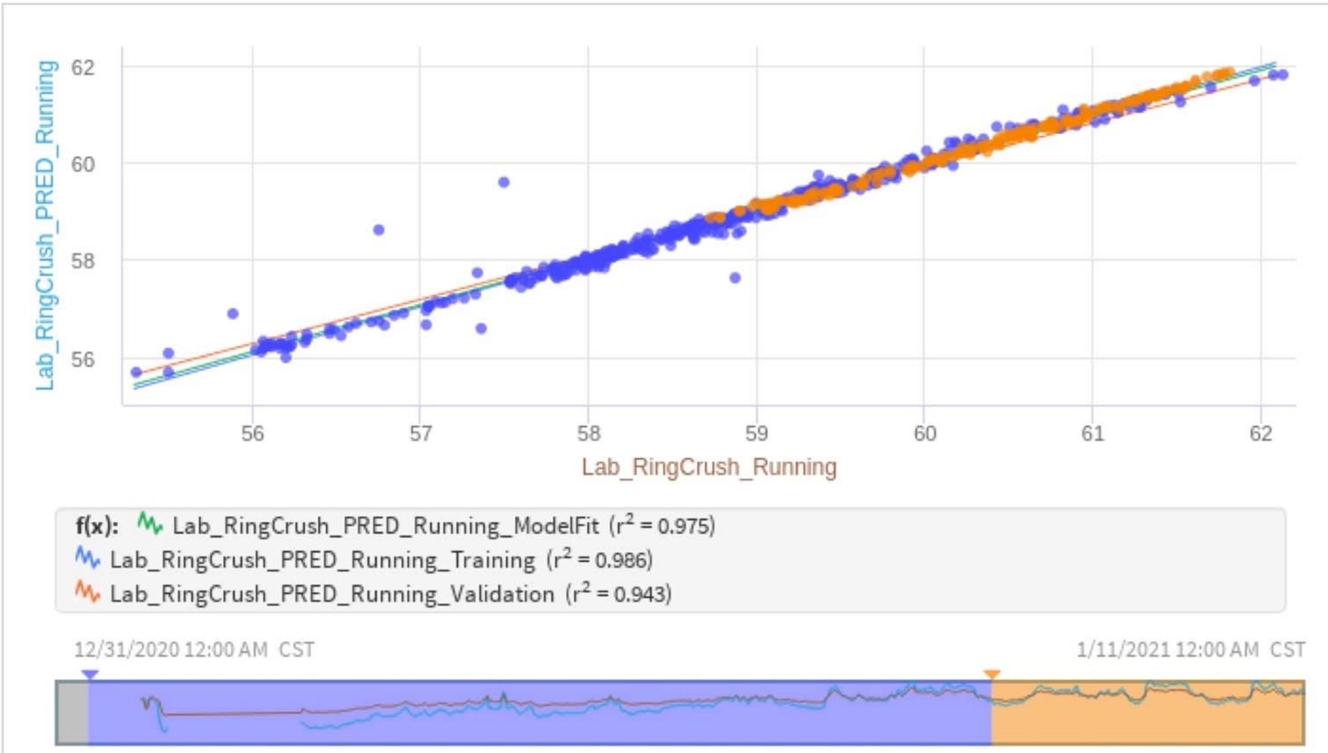
Name	Avg	Max	Min	Range	S.D.	Count	Description
Lab_RingCrush_Running	58.846	62.139	55.307	6.8318	1.6926	653	AREA_LAB Lab tag for running process
Lab_RingCrush_PRED_Running	59.249	61.854	55.683	6.171	1.3462	653	AREA_LAB Lab tag for running prediction
QCS_BoneDryWeight_DYN	17.504	26.497	4.136	22.361	4.7544	13,098	REEL Dynamic Adjusted tag
QCS_Moisture_DYN	8.8533	22.777	3.0409	19.736	3.0358	13,098	REEL Dynamic Adjusted tag
QCS_Caliper_DYN	1.9538	3.8532	0.4026	3.4506	0.488	13,098	REEL Dynamic Adjusted tag
MV_OCCFreeness_DYN	590.02	846.18	454.63	391.56	106.71	156,830	THICKSTOCK Dynamic Adjusted tag
MV_SWPct_DYN	26.648	42.879	-0.021	42.9	4.4962	156,830	THICKSTOCK Dynamic Adjusted tag
PulpEye_BlendCrill_DYN	172.78	298.43	11.966	286.46	64.646	156,842	THINSTOCK Dynamic Adjusted tag

Predictive Models

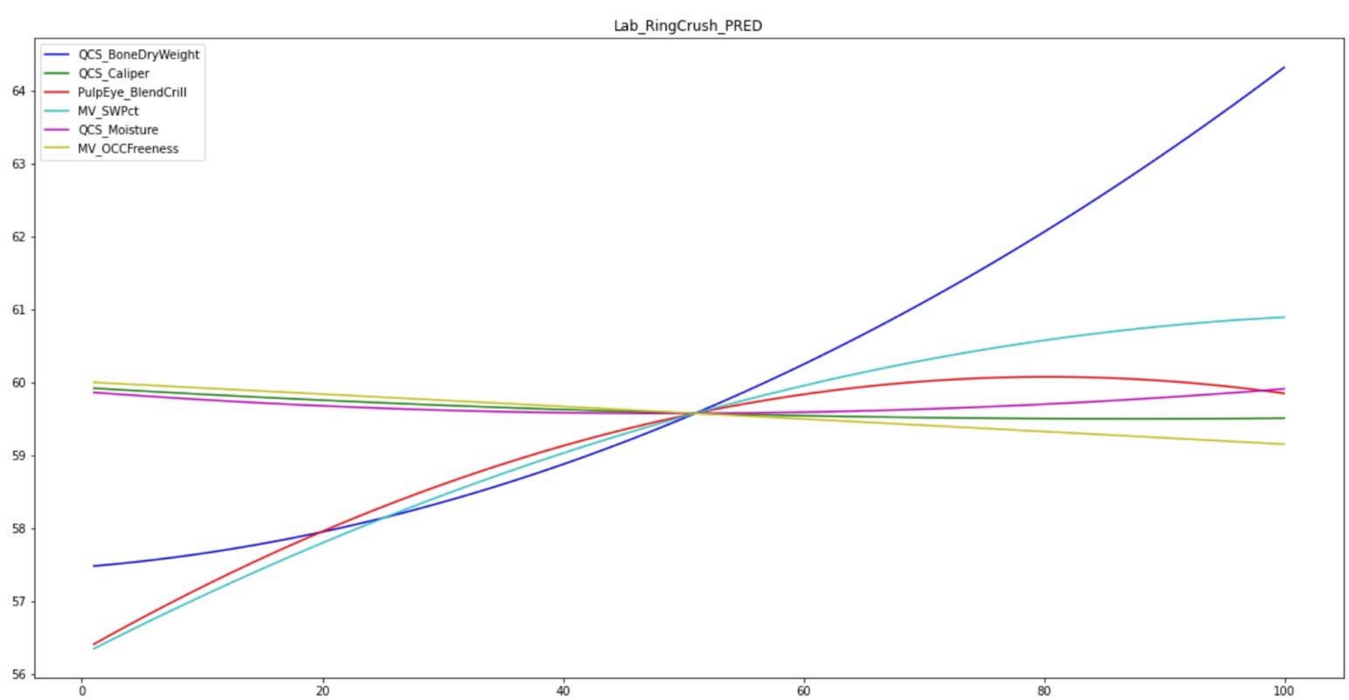
Prediction models were built to show the degree of fit. Since the relationships were specified in the dataset generator, the expected results were known. It was expected that the models would show a good fit, but that the process noise and realism of the dataset generator would still show some outliers or points that did not exactly fit the model. As shown below, the resulting model for ring crush showed excellent fitness to both training and validation data while still showing realistic deviations from noise inherent in any industrial process.

Training data is blue

Validation data is orange



Of particular interest was the sensitivity plots. It was expected that these would match the configured gain classes and configuration parameters in the dataset generator. The result shown below shows very good matchup with the expected results. Notice that some relationships appear to be linear, and others have significant non-linear curvature.



Conclusion

The result of this work is a dataset generator that can help the student, young engineer, instructor, vendor, or manufacturer obtain a realistic dataset for training, development, and demonstration purpose in a matter of hours instead of months. This work is not considered complete. Improvements in performance and the user interface can make it much more user friendly. The ongoing work with Alia Rezvi from City University of London should get closer to those goals by the end of her term in May. Following this, the code will be available for download and revision to continue its improvement and extensibility.

References

1. "Handbook for Pulp & Paper Technologists", Gary Smook, TAPPI Press, 4th edition
2. "Introduction to Paper Properties: PPS 102 Reading Materials", The Paper Science and Engineering Department, Miami University, Oxford OH, 1982
3. "A Theory for the Tensile Strength of Paper", D.H. Page, presented at the 54th annual meeting of TAPPI held in New York, NY Feb 17-20, 1969
4. "Morphology of Pulp Fiber from Softwoods and Influence on Paper Strength", Richard A Horn, Forest Products Laboratory, US Department of Agriculture, 1974
5. "Morphology of Pulp Fiber from Hardwoods and Influence on Paper Strength", Richard A Horn, Forest Products Laboratory, US Department of Agriculture, 1978
6. "On the usage of online fiber measurements for predicting bleached eucalyptus kraft pulp tensile index — an industrial case", Demuner, R.B., Faria, C.A., de Almeida, A.A.R., et al., *TAPPI J.* 21(7): 365(2022).