Data Modelling & Analysis in Real-time Process Control

S. SriDevi MCA, M.Phil, (Ph.D.) and Dr. P. Sakthivel, B.E, M.E., Ph.D.

1PhD Scholor, Dept. of Computer Science, Bharathiyar University, Coimbatore, India; 2Associate Professor, Department of Electronics and communication, Anna University, Chennai, India

ABSTRACT

Rapid advancements in technology today exert major influences on the course of simulation in many areas like interactive simulation & prediction models. Research in continuous event simulation modeling techniques is an evolving area with number of useful applications being developed. In this paper a comprehensive integrated simulation model development is proposed which uses various related modelling techniques to solve real time process objective. Details of each of the modelling techniques, their training and predicting parameters, capabilities, synergized control mechanism useful for applications in process industries are discussed.

INTRODUCTION

This paper aims to elaborate a phased approach, for design & construction of below generic data modelling and analysis system to address specific industrial process pain points.

The aim is to build Supervisory software program with integrated data modeling/tuning tools like Neuro Fuzzy, MPC, PLS, PCA bundled together by a rule based optimization System.

This solution named as data modeling as in stricter sense, the models or relationship between entities (“Parameters”) are derived from data rather than from experience (“heuristics”) or by virtue of mathematical equations. The system maintains the properties of individual artificial neural networks (ANN)/Model Predictive Control (MPC)/ Partial least square (PLS) models and manages valid online data collection from the plant for training/modeling. Other features include deciding the time to go for results of a particular model and trigger online/offline training of individual networks.

Related Work:

In recent years, an artificial neural network-based model predictive control has shown good performance with in the model/plant uncertainties / non-linearity boundary. Neural networks & MPC are frequently used in many control systems and can be applied in various chemical processes, since it offers a promising strategy for process modeling & control of complex nonlinear systems. For eg., (Kittisupakorn, 2010) proposed the NNMPC for controlling a steel picking process; ANN based modelling for compressive strength of concrete was proposed by (Vinay Chandwani, 2014) and (Alireza Najigivi, 2013) proposed ANN to study permeability properties of cement etc.

Therefore, the objective of this work is to design a comprehensive control system based on data modelling and predicting algorithms which could adapt to process dynamics and solve the non-liner multivariate optimization process objective effectively. The paper is divided into two major sections; the first section explains the proposed model, components, tuning parameters and their usage, while the second section elaborates the real-time application of the system identification, prediction and controlling of a continuous process problem.

Data / process modeling & prediction:

The proposed solution has the below components:
(a) Data modeling component:
To model the process behavior the past history data (I/O) is collected, validated and used to train ANN/MPC offline. Once tuned with various parameters explained below, we leverage best of Neural/MPC/PLS predictions...
of output variable with proper data validation and suitable training/prediction algorithms. These algorithms are parameterized to customize for tuning the modelling and control aspects of the model to suit specific needs.

(b) Supervisory control system program:
To maintain desired set point (SP) of the controlled variable (CV), the error (SP-predicted) and the change of error direction are fed to a typical fuzzy membership functions to determine the desired Manipulated variable (MV) changes. A Control strategy component which integrate the calling of these model blocks with If-then-else statements, built over process variables and constraints. The communication interface fetches the process values to these strategies in real-time and the Rules engine execute these strategies every second, to integrate the system input/outputs and transmit the suggested control actions by the model to plant automation systems.

Below diagram summarizes the above proposed integrated model:

Fig. 1: Real-time process control with modelling techniques.

Modelling/Prediction Tools:
This section elaborates the important concepts of prediction techniques which are used in the Data modeler solution.

(i) Artificial Neural Network (ANN):
ANN has wide range of applications in process industries. This is composed of a number of highly interconnected processing nodes (neurons) working in collaboratively to address specific process problems. An ANN is tuned to learn/model the process which could be applied to meet objectives, such as prediction, classification and pattern recognition etc.

Some of the advantages of neural network are
1. There is no clearly defined mathematical model available which considers all the inputs and constraints of the system.
2. Neural Nets (ANNS) are good at learning multi-input relationships
3. ANNs perform well with noisy data.

Fig. 2: ANN Structure to map input/output association

The application of neural networks to Control Systems works by obtaining existing plant data for the desired parameters and train through a standard Back-Propagation ANN structure to learn the relationship between the input parameters and the outputs. The number of hidden layers and the weights on each of these layers determine the accuracy of output from the network.
The weights on these hidden layers are free to get updated in each iteration, till the prediction output is within the acceptable prediction range. This is a kind of creating associative mapping of input set to a specific pattern.

Once the relationship is learnt this knowledge can be used for predicting the output with the available inputs.

(ii) Model Predictive Control (MPC):

MPC derives a model of the process, and uses it to predict the future evolution of process to control the target variance. Model predictive control refers to set of algorithms that compute a sequence of manipulated variable (MV) adjustments in order to optimize the future behavior of plant. This simplifies & eliminates the need for complex controller design.

MPC has applications in multivariate control systems (K. S. Holkar, 2010) which use the model dynamics of the process; history of previous control actions & an optimization cost function on the receding prediction horizon, to calculate the required control steps.

At time a given time ‘t’ the current plant state is sampled and a cost minimizing control strategy is for a relatively short time horizon in the future: [t, t+T].

In this algorithm explicit dynamic model of the plant is used to predict the effect of future actions of the manipulated variable on the output. The future moves of the manipulated variables are determined by optimization with the objective of minimizing the predicted error subject to operating constraints. The optimization is repeated at each sampling time based on updated information (Measurement) from the plant.

The below schematic diagram proposes a system identification/ continuous learning phase where the model is tuned to process dynamics. The tuned model when introduced as a predictor controller, responds to the real time data from the plant and the output is fed to the optimizer module where the required change are decided and transmitted back to process as a closed loop controller.

MPC prediction example for new set of input data:

For predicting output value for new set of input data, the coefficient calculated from known value of input and output data set is used.

For single input single output system with dependency size 3 Coefficient values are 8.333334 3.333330 -1.666667

For predicting value for input 11, previous value are taken as last set of data in the training set. Predicted value for 12 is 8.333334 * 11 + 2.965367*10 + (-1.666667) * 9 = 118.699968

For 2 input 1 output system with dependency size 3, coefficient values are 2.705628 2.965367 3.225109 0.108226 0.367964 0.627705
The predicted value for input 11 and 21 is 2.705628*11 + 2.965367*10 + 3.225109*9 +
0.108226 * 21 + 0.367964*20 + 0.627705 * 19 = 109.999977
For input 12 and 22 the predicted value is 2.705628*12 + 2.965367*10 + 3.225109*9 +
0.108226*22 + 0.367964*20 + 0.627705 * 19 = 112.6459979

(iii) **Partial Least Square analysis (PLS):**

Partial least squares regression is an extension of the multiple linear regression models. Observations(X) denoted by Y dependent variables are stored in an X × Y matrix represented by Z. Predictors(U) on these X observations are collected in the X × U matrix A.

The objective of PLS is to predict Z from A and to describe their common format. We have used SIMPLS (Statistically Inspired Modification of Partial Least Square) algorithm (de Jong, 1993) for computing PLS factors.

(iv) **Principal Component Analysis (PCA):**

In neural networks applications, typically the number of the input vectors is high, but the attributes of the vectors are highly redundant or correlated. PCA helps us to minimize the input dimensions and identify influential vectors.

PCA orthogonalizes the components of the input vectors such that they are uncorrelated with one another. The resultant orthogonal components (PC) are ordered, from larger to smaller variations & the items contributing minimal variation in the data set are removed.

In Neural network offline training module, the data is pre-processed by to PCA, before the training happens. This eliminates those principal components, which contribute less than fractional variance to the total variation in the data set. Fractional variance used: in the range of 0.01 to 0.1.

The below 3 methods of PCA data reduction are used and compared
a) Correlation method: based on correlation matrix of cross products input data,
b) Variance method: This algorithm is based on variation matrix of cross products input data,
c) Sum of squares method: based on sums-of-squares-cross-products of input data

All these method generate reduced data size matrix and Transformation matrix. During predicting process this input data is multiplied with transformation matrix then the result is used for prediction.

(v) **Sensitivity Analysis:**

The system automatically simulates one of the critical input variables, between the specified ranges, keeping all other inputs as constant. The average sensitivity of each input over the output is stored. The validation module continuously observes this output, with the predictive output of Neural/MPC/PLS to validate major deviations in the results and alerts for retraining as needed.

**Features of Proposed Data / Process Modeling System:**

1. Single user interface for configuring & processing all the Neural, MPC, PLS and Interpolation Techniques.
2. Single Data Entry point, for all the systems where the input/output validations are done.
3. Offline/online Neural/MPC Training & Prediction.
4. Data collection: Can be Manual entry or through online.
5. Online data collection can be enabled the system connected to a plat automation system [PLS/DCS] and getting continuous feed of input/output parameters through online data collection interface drivers like DDE (Dynamic Data Exchange/OPC client (OLE for process control)
6. Parameterized network attributes and properties for tuning.
7. Automatic handling of multiple training, weights files.
8. Continuous Online Network Learning process to have frequent updates on system dynamics
9. Maintenance of Training data size and training frequency.

We propose to build an interface wizard to configure ANN/MPC/PLS networks as per below simpler prototype built using C++/flat files.

In the following section, we shall discuss in detail about the parameters used to fine tune the model, in the above prototype:

(a) **Neural Control Parameters:**

*Fixing a Network:* The weight matrix W of a neural network represents the information learnt. With this, these parameters help to adjust the learning pattern and accuracy of output:

*Fixed networks* are those, where we cannot change the weights, i.e. dW/dt=0. In such networks, the weights are fixed by default according to the problem to solve. *Adaptive networks* which are able to change their weights, i.e. dW/dt not= 0.
Self-fixing:
The proposed model has an option to Self-Fixing ANN, so that network architecture is fixed automatically.

Manual fixing:
The user has been given option to select the required number of layers, nodes per hidden layer and number of epochs as applicable to the current scenario

**Fig. 5:** Data modelling tool with configurable parameters.

**Fig. 6:** Neural Parameters.

**Activation Function selection:**
Relevant activation functions like sigmoidal can be chosen / parameterized.

**Fig. 7:** Activation functions selection.

**Thresholds:**
This output threshold validation is done for all the output variables, at the time of online data entry. The predicted value is to be between the two thresholds, for the pattern to be inserted in the training file. Out of threshold might be due to some of the input parameters have deviated much leading to abrupt prediction and need not go in to the training file.

**Online training:**
Online option enables the particular network to go for Automatic Online training in a particular frequency to keep the system updated on recent plant dynamics. This applies for the system connected to a plat automation system and getting continuous feed of input/output parameters through online data collection.
Error tolerance: This factor controls the allowable difference between the available and predicted outputs.

Momentum:
This is the amount of influence of previous pass’s weights change to be taken for current cycle’s weights change. Using this term the network is less likely to get in a local minimum early on in training. The momentum will essentially "push" the changes over local increases in the error function.

Learning Rate:
Determines the amount of weight change to be effected on each pass through the training data set. Speed of learning is governed by the learning rate. If this is increased too much, learning becomes unstable; the net oscillates back and forth across the error minimum.

Noise Factor:
This factor specifies the amount of input noise. This also helps the network from memorizing the input patterns.

Training Frequency:
Based on this online training frequency, the time to go for online training is decided. If the number of records entered online, since last online training, crosses this value, the network is trained online (with the saved weights).

Bias Addition:
Bias input to the neural network; Allowable values: +1, 0 & -1

Data Set Size:
The number of records to be maintained in the training set.

(b) MPC Attributes:

Fig. 8: Proposed MPC Parameters.

Dependency Size:
denotes current pattern output dependency on how many previous patterns inputs. If this value is 3; current pattern output depends on current input pattern & previous 2 input patterns.

Weight on Input:
Specifies the weight age given to the input in MPC Cost function.

Control Horizon:
This value specifies the number of future manipulated variable moves to be predicted.

(c) PLS Attributes:
NC: This attribute denotes the Number of Components of PLS Model.

Real-Time Application of Data Modelling:
In this section a practical application of the data modeling, prediction & control techniques discussed above, are detailed out for solving Cement grinding mill control objective in a modern Cement plant.
**Process Problem: Blaine Prediction:**

As Cement is a reactive powder having PCD (particle size distribution) as main characteristic, determining its total surface area & potential strength. Since long time, cement fineness was measured by sieving the cement. The amount of mass passing the sieve (typically 95% in modern cement) is related to the overall strength of the cement, because the smaller particles are essentially reactive.

The total surface area in square meters of all the cement particles in one Kg of cement is called the specific surface or **blaine** [unit: cm²/g]. This measured by Principle of Air permeability method. The time taken for a fixed quantity of air to flow through compacted cement bed of specified dimension (Pallets) and porosity are measured,(E. T. Harrigan, 2013). The cement particle size distribution determines the size & number of individual pores and in turn impacts the time for the specified air flow. Average Blaine fineness of modern cement ranges from 3,000 to 5,000 cm²/g (300 to 500 m²/kg).

A more comprehensive measure of fineness (in sub-micrometer levels) can be achieved by laser particle size analysis, but it is yet to get widespread adoption & used mainly as a research tool. With lack of continuous Cement fineness analyzer, our cost effective predictive model fills the gap in suggesting control actions which are transmitted to plant control system to achieve the target fineness.

Typically as the Cement fineness is measured in one hour frequency, the action taken in adjusting input parameters (Manipulated variables), are usually reactive and the production happened for the elapsed hour, cannot be corrected. So the need is continuous proactive control.

**PCA Analysis for input data reduction:**

Typically Cement fineness varies linearly with the mill power consumption as shown in figure-9 below. But to make the model more precisely resonate to real-time dynamics of the mill which are controlled by various parameters which all might impact mill power consumption / fineness. Field study conducted to collect 10 such input parameters of the grinding unit and PCA correlation analysis done to find out most critical parameters influencing cement blain output.

**Fig. 9:** Input/output relations.

**Data Analysis and Model building:**

The below figure-10 depicts, how the multiple parameters impacts the Cement fineness. We study & derive the input parameters which are most influential in deriving the output / controlled variable with PCA and order them.

**Fig. 10:** Predicting Cement fineness – input parameters.
These parameters impacting the output fineness (blaine) could be Input feed weight flowing to the mill, Grinding/Hydraulic pressure (HP) over the grinding table, Airflow across the mill and the temperature/moisture of the material being ground. We take these 4 important parameters as input variables impacting the output. The input/output forms the training set to neural network.

Typically the blaine is be measured each hour and it would take 15 minutes for a sample to be taken, undergo compression to a pallet and results are analyzed.

**Online data collection for predictive model:**

The sample taken event is transferred to the PLC and the mill conditions (4 input variables) at the time of sample taken are persisted and transferred to our model.

The actual output blaine result from analyzer is transferred to PLC/modeler. The set of input and output combinations are recorded in a training file. Such hourly results are collected for a period of 1-2 weeks to get sufficient representative sample data.

The same training file can also be created by manually recording all these parameters each hour.

1. Each row of training data should contain the number of inputs followed by the output for that pattern.
2. Determine which input parameters affect the value of the output.
3. Collect sufficient input and corresponding output data to give us as widely representative a sample of data as possible. There is no limit to the number of inputs or outputs and are usually delimited between the values as in the following format and save the file. Below is the sample data collected from cement plant:

**Training file format:**

Input1 Input2 …. Input_n     Output1 Output2….  Output_n

Example:

Feed         HP       Airflow    Inlet-
154.52    306.83     95.92         59.69            3219  
155.72    304.79   102.44         60.58            3190  
155.97    296.48    100.67        60.58            3321  


In the training file, historical data of 4 input parameters and the blain output corresponding to them are captured and stored in the format shown above and which is read and processed by the training module.

**Training the network:**

Soft computing methods such as Neural/MPC enriches non-liner statistical, probabilistic data and act as adaptive optimization tools for learning, predicting and classifying new patterns based on historical data (Vinay Chandwani, 2014). The input and output set is fed to the neural network to enable supervised learning in back propagation algorithm. Once the Network is trained the final weights are saved in weights file. Similarly the during MPC training the resultant coefficients are stored in related files.

**Model in Action & Result Analysis:**

The parameters discussed above momentum, dependency, activation function, thresholds etc., are adjusted to arrive at optimally trained network. After the network is trained, weights/coefficients are optimized for near zero errors and ready for prediction. To make the control more proactive and predict in real-time, we read the 4 parameters of the mill every second through plant communication drivers and feed to the model. It was observed that for Neural, we get optimal results for the parameters such as 5 hidden layers with 6 nodes each and momentum of 0.2 in a 1000 epoch cycles with a sigmoidal activation function resulted in optimal outputs for the problem chosen.

The trained model (optimized weights & Coefficients) could predict the output (blaine) with these online inputs and without the guiding outputs. By linking the model to a control engine program, prediction process can be triggered for each second. The output from Neural/MPC predictions are compared and averaged. The output is fed to Sensitivity analysis with major contributing input variables to compare whether the predicted output has any major variance. The threshold validation ensures the results are within the prediction horizon. Like if the accepted range of blaine is between 3200 to 3500, then the predicted output most suiting all the constraints is chosen as next probable value.

The validated output then flows through the optimizer which is a control strategy / fuzzy block. The difference between the predicted output and the blaine target (Set point) are derived (error). To minimize the error (objective function), rule strategies / fuzzy blocks are configured to suggest input parameter changes (MV) & transmitted to field. Based on this the inputs like mill feed-rates, grinding pressure, airflow & separator settings are adjusted to maintain constant specific target/set point (CV). As a closed loop control system, the output error (set point-predicted) drive the input changes direction and magnitude for reaching process stability.
If the prediction range is continuously moving away from control objectives, offline training of Neural/MPC networks with more recent data set collected from plant to reflect process dynamics.

**Conclusion:**

An integrated Data modelling & automation system is envisaged. Neural/MPC results are cross validated, resultant error is fuzzified & the Rule engines communicate the decision to plan automation for controlling the desired output parameter.

The solution uses comprehensive closed loop feedback system integrated with plant communication drivers. This enables getting auto data feed for Model tuning, prediction and optimal control of the input variables to achieve the desired set points based on error reduction strategies.

The model aims to be simpler to be built with standard algorithms & libraries in C++ modules/sequence, enabling users with tuning parameters and validation options. Various customizable parameters introduced in constituent algorithms for data validation and model tuning, which helps the user to replicate the process dynamics to closest possible levels of accuracy.

As Cement grinding is the major consumer of electric power in a cement plant, any potential savings in power sufficient to produce required quality (fineness) of cement could yield substantial savings. Since we get the output prediction more frequent as opposed to typical Blaine sampling/measurement cycle of 1 hour from lab we could impose more proactive control on manipulated variables.

As the model is simple, generic & Scalable we could add number of networks in each of the techniques, say for example, different models for each type of cement produced as input scenarios are different.

**Scope for future research:**

As the current prototype concentrates more on the data modelling aspects, all the peripherals such as linking with robust Fuzzy / rule engines / plant communication systems could be detailed to make it as an end to end solution. The proposed model to be made more generic and scalable to fit to solve any process control objective

**REFERENCES**


