Natural Gas Prediction Using The Group Method of Data Handling

James C. Howland III
Computer Information Systems
College of Technical Sciences
Montana State University – NORTHERN
Havre, Montana, USA
howland@msun.edu

Mark S. Voss
Civil Engineering Technology
College of Technical Sciences
Montana State University – NORTHERN
Havre, Montana, USA
MarkVoss@EvolutionaryStructures.com

ABSTRACT
The flow of natural gas from a system of wells is a highly nonlinear process. In this paper we are taking a time series approach to the prediction of natural gas. In particular, the Group Method of Data Handling (GMDH) is employed to develop a nonlinear polynomial time series model for a small natural gas collection system.

KEY WORDS
Natural Gas Prediction, GMDH, Inductive Learning

1 Introduction

The prediction of natural gas flow into a small collection system was investigated using the Group Method of Data Handling. The source data was input into a GMDH application written by the author in Java. The GMDH application generated a model for the prediction of gas production based on the time series readings from the collection system. The application was executed using two inputs and six inputs with the results being presented in this paper. Previous research [1] [2] [3] demonstrated the application of the GMDH methodology with respect to the prediction of natural gas flow. This work represents a more detailed study into predicting natural gas production using the GMDH methodology. In particular, more focus was placed on the development of training, testing, and validation data.

2 Group Method of Data Handling

The Group Method of Data Handling is a combinatorial multi-layer algorithm in which a network of layers and nodes is generated using a number of inputs from the data stream being evaluated. The Group Method of Data Handling (GMDH) was first proposed by Alexy G. Ivakhnenko [4]. The GMDH network topology has been traditionally determined using a layer by layer pruning process based on a pre-selected criterion of what constitutes the best nodes at each level. The traditional GMDH method [5] [6] is based on an underlying assumption that the data can be modeled by using an approximation of the Volterra Series or Kolmorgorov-Gabor polynomial as shown in equation(1).

$$\begin{aligned} y = a_0 &+ \sum_{i=1}^{m} a_i x_i \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} x_i x_j \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_{ijk} x_i x_j x_k ... \end{aligned} \tag{1}$$

2.1 GMDH Layers

When constructing a GMDH network, all combinations of the inputs are generated and sent into the first layer of the network. The outputs from this layer are then classified and selected for input into the next layer with all combinations of the selected outputs being sent into layer 2. This process is continued as long as each subsequent layer$_{(n+1)}$ produces a better result than layer$_{(n)}$. When layer$_{(n+1)}$ is found to not be as good as layer$_{(n)}$, the process is halted.

2.2 GMDH Nodes

Each layer consists of nodes generated to take a specific pair of the combination of inputs as its source. Each node produces a set of coefficients a_i where $i \in \{0, 1, 2, \ldots, 5\}$ such that equation (2) is estimated using the set of *training* data. This equation is tested for fit by determining the mean square error of the predicted $\hat{y}$ and actual y values as shown in equation (3) using the set of *testing* data.

$$\hat{y}_n = a_0 + a_1 x_{i_n} + a_2 x_{j_n} + a_3 x_{i_n} x_{j_n} + a_4 x_{i_n}^2 + a_5 x_{j_n}^2 \tag{2}$$

$$\mathbf{e} = \sum_{n=1}^{N} (\hat{y}_n - y_n)^2 \tag{3}$$

In determining the values of **a** that would produce the "best fit", the partial derivatives of equation (3) are taken with respect to each constant value a_i and set equal to zero.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{a_i}} = 0 \tag{4}$$

Expanding equation (4) results in the following system of equations that are solved using the *training* data set.

$$\sum_{n=1}^{N} y = \sum_{n=1}^{N} a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \tag{5}$$

$$\sum_{n=1}^{N} y x_i = \sum_{n=1}^{N} a_0 x_i + a_1 x_i^2 + a_2 x_i x_j + a_3 x_i^2 x_j + a_4 x_i^3 + a_5 x_i x_j^2 \tag{6}$$

$$\sum_{n=1}^{N} y x_j = \sum_{n=1}^{N} a_0 x_i + a_1 x_i x_j + a_2 x_j^2 + a_3 x_i^2 x_j + a_4 x_i^2 + a_5 x_i x_j^3 \tag{7}$$

$$\sum_{n=1}^{N} y x_i x_j = \sum_{n=1}^{N} a_0 x_i x_j + a_1 x_i^2 + a_2 x_i x_j^2 + a_3 x_i^2 x_j^2 + a_4 x_i^3 x_j + a_5 x_i x_j^3 \tag{8}$$

$$\sum_{n=1}^{N} y x_i^2 = \sum_{n=1}^{N} a_0 x_i^2 + a_1 x_i^3 + a_2 x_i^2 x_j + a_3 x_i^3 x_j + a_4 x_i^4 + a_5 x_i^2 x_j^2 \tag{9}$$

$$\sum_{n=1}^{N} y x_j^2 = \sum_{n=1}^{N} a_0 x_j^2 + a_1 x_i x_j^3 + a_2 x_i^3 + a_3 x_i x_j^3 + a_4 x_i^2 x_j^2 + a_5 x_j^4 \tag{10}$$

The equations can be simplified using matrix mathematics as follows.

$$\mathbf{Y} = \begin{pmatrix} 1 & x_i & x_j & x_i x_j & x_i^2 & x_j^2 \end{pmatrix} \tag{11}$$

$$\mathbf{A} = \mathbf{Y}^\top \mathbf{Y} \tag{12}$$

$$\mathbf{A} = \begin{pmatrix} 1 & x_i & x_j & x_i x_j & x_i^2 & x_j^2 \\ x_i & x_i^2 & x_i x_j & x_i^2 x_j & x_i^3 & x_i x_j^2 \\ x_j & x_i x_j & x_j^2 & x_i x_j^2 & x_i^2 x_j & x_j^3 \\ x_i x_j & x_i^2 x_j & x_i x_j^2 & x_i^2 x_j^2 & x_i^3 x_j & x_i x_j^3 \\ x_i^2 & x_i^3 & x_i^2 x_j & x_i^3 x_j & x_i^4 & x_i^2 x_j^2 \\ x_j^2 & x_i x_j^2 & x_j^3 & x_i x_j^3 & x_i^2 x_j^2 & x_j^4 \end{pmatrix} \tag{13}$$

$$\mathbf{x} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \end{pmatrix} \tag{14}$$

$$\mathbf{b} = \begin{pmatrix} y\mathbf{Y} \end{pmatrix}^\top \tag{15}$$

This system of equations then can be written as:

$$\sum_{n=1}^{N} \mathbf{A}\mathbf{x} = \sum_{n=1}^{N} \mathbf{b} \tag{16}$$

The node is now responsible for evaluating all inputs of x_{i_n}, x_{j_n}, y_n data values in **A** and **b** for the *training* set of data. Solving the system results in **x** being the node's computed set of coefficients. Using these coefficients in equation (2), the node then computes its error by processing the set of *testing* data in equations (2) and (3). The error is the measure of fit that this node achieved.

2.2.1 GMDH Nodal $3D$ Phase-Space

Each node in the network can be understood as a 3 dimensional phase. To help understanding and visualization of the node's task, *trial data* was generated using equation (17). The *trial data* is a general sin wave with decay.

$$y = \sin(x) e^{\frac{-x}{10}} \tag{17}$$

Figure (1) shows the graph of this data.

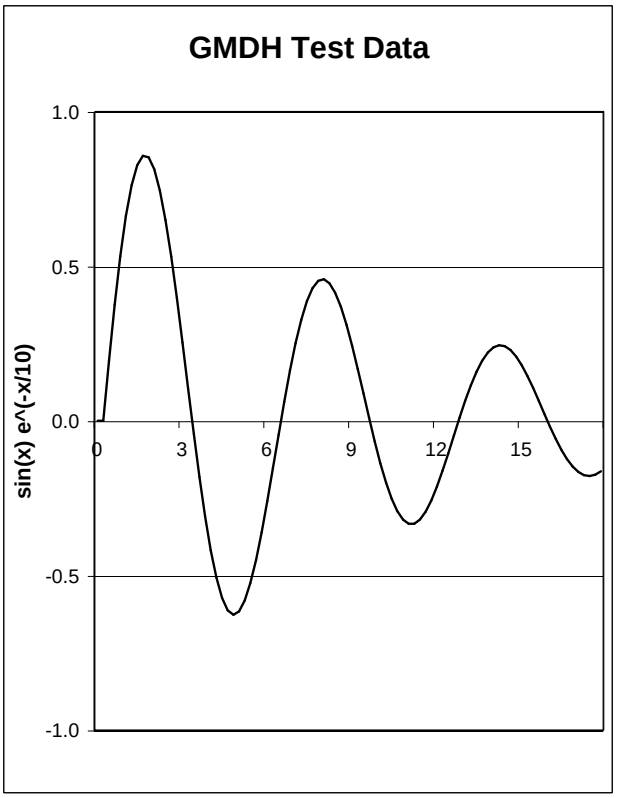

Figure 1. Trial Data

In [1], a surface representation of a nodal $3D$ phase-space was presented. In figure (2), the actual data points for the first node (y, y_{t-1}, y_{t-2}) in the GMDH network generated to learn the *trial data* were plotted. This clearly visualizes the problem to be solved by the network. The phase-space plot is coplanar so is shown with a slight $3D$ rotation.

As expected, the GMDH network produced a solution that was 100% accurate using a single layer and single node.

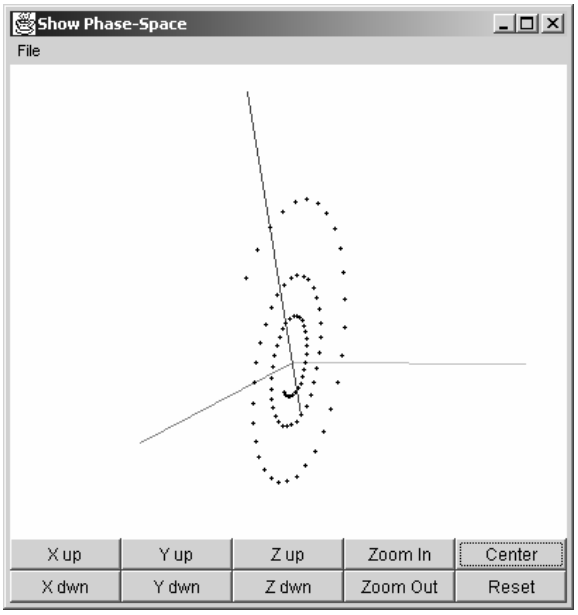

Figure 2. Trial Data Phase-Space Plot

2.3 GMDH Connections

A GMDH layer sorts its nodes based on the error produced, saving the best **N** nodes. Each nodes' generated y_n values become one set of inputs to be used by the next layer when it combines all outputs from the previous layer's nodes assigning them to the new layer's nodes. (See figure 3.) The layer must remember the nodes that were saved so that other data submitted to the network will follow the same generated path to the output.

2.4 GMDH Network

When the GMDH network is completed, there is a set of original inputs that filtered through the layers to the optimal output node. This is the computational network that is to be used in computing predictions. For example, if a time-series input stream was used and it was determined from the network that y_{t-1}, y_{t-2}, and y_{t-5} produced the best output after three layers, a prediction computation ($\hat{p}$) would need to have p_{t-1}, p_{t-2}, and p_{t-5} as inputs into the network, cascading through the layers and appropriate nodes, until the output is obtained from the last layer.

3 North Bear Paw Gas Analysis

The data that was analyzed using the GMDH method was a time-series of monthly volumes produced within a small natural gas gathering system in the Bear Paw mountains in

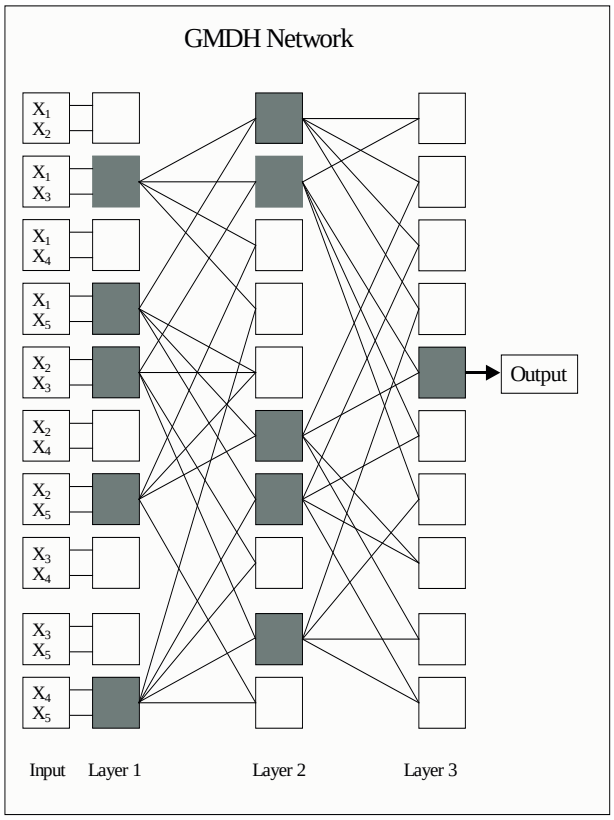

Figure 3. GMDH – Feed-Forward Network

Montana. This gas volume information is the sum of all wells producing into the gathering system for the month. The data was separated into three groups: *training*, *testing*, and *attempts*. The data separation for this study was done using the pattern: *train*, *train*, *train*, *attempt*, *test*, *test*, *test* resulting in 21 training points, 18 testing points, and 7 attempt points. The GMDH network was operated two times, first by specifying two inputs, then by specifying six inputs.

3.1 Two Input GMDH

A two input GMDH network is limited to a single layer and a single node as the two inputs are combined by the node producing a single output. Once the network was trained using the set of *training* data with two inputs (y, y_{t-1}, y_{t-2}), the set of *attempt* data was input into the network to see how well it computed. Figure (4) shows the original time-series along with the plotted predicted volume points.

3.2 Six Input GMDH

The GMDH program was executed specifying six inputs $(y_{t-1}, y_{t-2}, \cdots, y_{t-6})$ which generated a three layer network. The attempt data was input into the resulting network with the graph showing the original data and the plot-

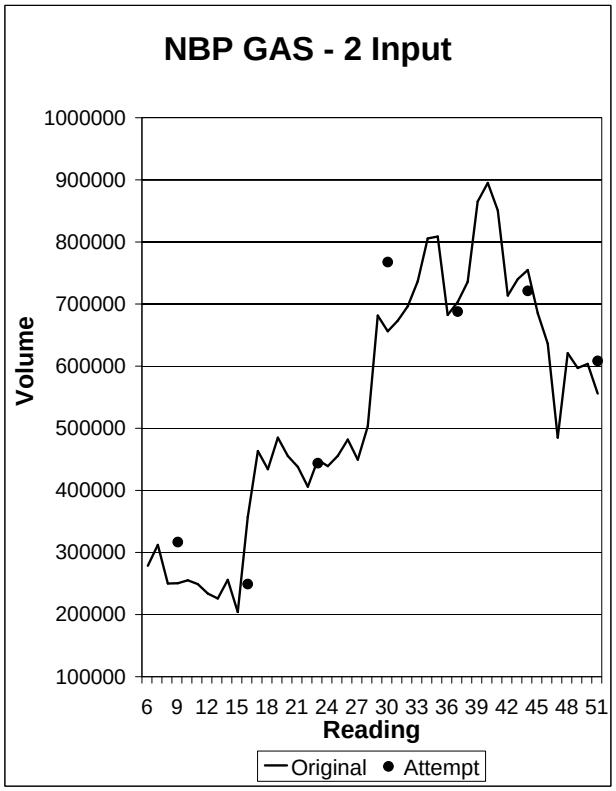

Figure 4. Two Input GMDH Prediction Results

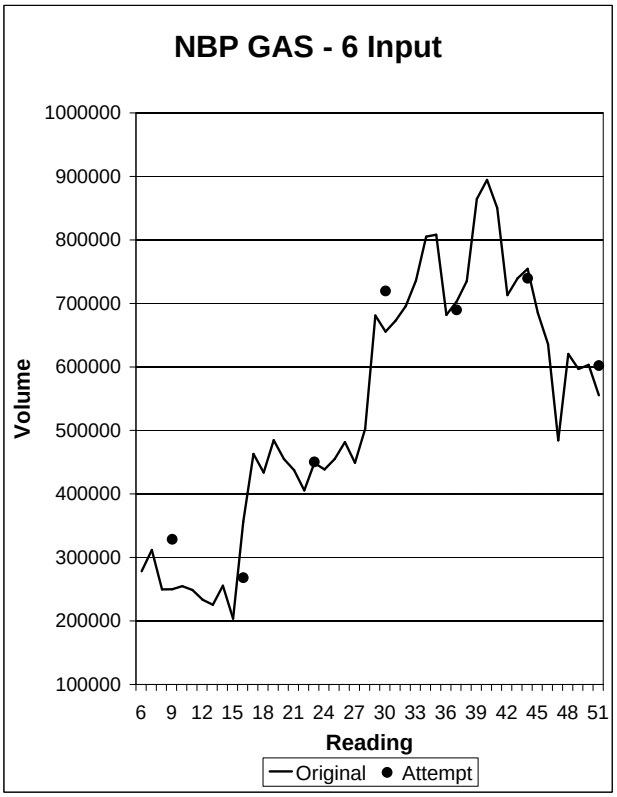

Figure 5. Six Input GMDH Prediction Results

ted predicted volumes in figure (5).

3.3 Two Input Nodal $3D$ Phase-Space

To help visualize the problem being attempted by the GMDH network, the input data stream was plotted as a $3D$ phase-space as shown in figure (6). The generated equation (2), when plotted, should result in a surface that approximates the phase-space plot that the node was attempting to match. Three views of the generated phase-space surface with the data stream points plotted are included. These figures (7, 8, 9) show how the GMDH attempted to produce a surface to map to the input stream. This surface was produced using only the (y_n, y_{t-1}, y_{t-2}) node.

4 Conclusion

The prediction of flow from a natural gas collection system is significant for both the pipeline operators and the producers. When contracts for deliveries are typically made a month or more in advance, the over production/under production imbalance can become significant. This study was an initial use of the Group Method of Data Handling to study the natural gas production in the Bear Paw field. The GMDH network with two inputs produced acceptable results. The six input network learned the data a little better with the resulting predictions being somewhat better. Both two and six input GMDH networks predicted better than using a simple moving average.

Future studies can be done to improve the error and outputs from this set of data.

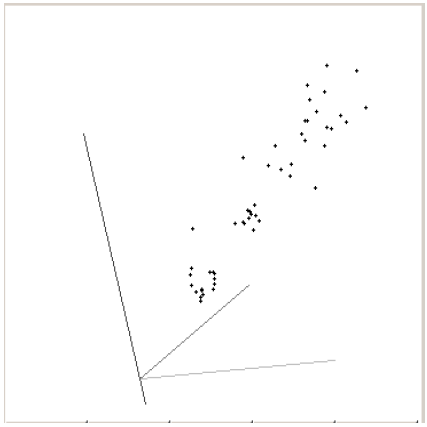

Figure 6. Two Input $3D$ Phase-Space

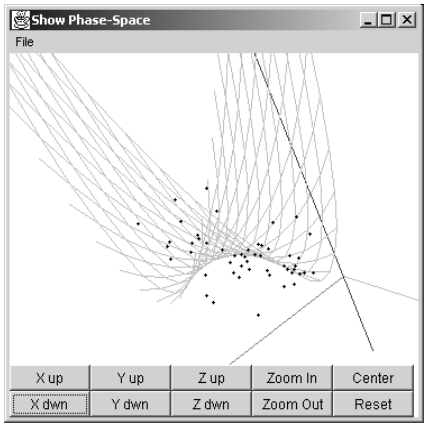

Figure 7. Two Input $3D$ Phase-Space with plot (1)

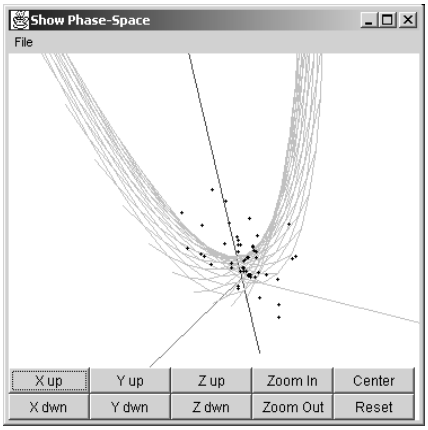

Figure 8. Two Input $3D$ Phase-Space with plot (2)

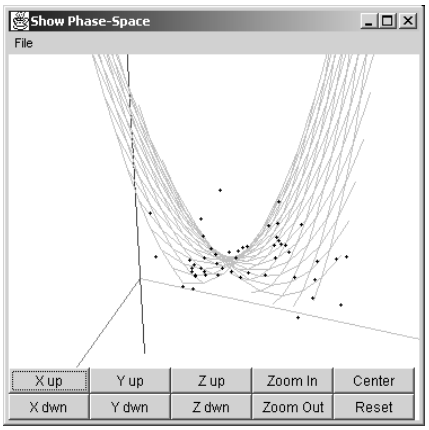

Figure 9. Two Input $3D$ Phase-Space with plot (3)

References

[1] M. S. Voss and X. Feng, "Emergent system identification using particle swarm optimization," in *Complex Adaptive Structures*, (Hutchinson Island, Florida).

[2] M. S. Voss and X. Feng, "A new methodology for emergent system identification using particle swarm optimization (PSO) and the group method of data handling (GMDH)," in *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference* (W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, eds.), (New York), pp. 1227–1232, Morgan Kaufmann Publishers, 9-13 July 2002.

[3] M. S. Voss, *The Group Method of Cartesian Programming: A New Methodology for Complex Adaptive Functional Networks*. PhD thesis, Marquette University, Milwaukee, Wisconsin.

[4] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 364–378, 1971.

[5] S. J. Farlow, *Self-Organizing Methods in Modeling: GMDH Type Algorithms*. New York: Marcel Dekker, 1984.

[6] H. R. Madala and A. G. Ivakhnenko, *Inductive Learning Algorithms for Complex Systems Modeling*. CRC Press, 1994.