IDENTIFICATION OF NONLINEAR CONTROL MODELS

USING VOLterra-Laguerre SERIES

by

Dale A. Smith

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Chemical Engineering

The University of Utah

August 2010
Copyright © Dale A. Smith 2010

All Rights Reserved
The dissertation of Dale A. Smith has been approved by the following supervisory committee members:

Philip J. Smith, Chair 17 May 2010

J.D. Seader, Member 17 May 2010

Terry Ring, Member 17 May 2010

Geoffrey D. Silcox, Member 17 May 2010

Mark Minor, Member 17 May 2010

and by JoAnn Lighty, Chair of the Department of Chemical Engineering

and by Charles A. Wight, Dean of The Graduate School.
ABSTRACT

Linear model predictive control has been widely accepted in industry as an important tool for the operation of difficult interacting processes. Linear identification and control techniques are well developed and well understood. In the industry, it is rare to find a system that is truly linear. While for many systems linear modeling and control can approximate their performance in certain regions, there exist some systems whose nonlinearity is great enough that an approximate linear model and control scheme cannot yield the desired accuracy. In order to control these more complex nonlinear systems, significant research has been dedicated to extending model predictive control to nonlinear systems.

The problem of implementing nonlinear model predictive control can be split into two main tasks: making the nonlinear model and calculating control inputs. The significant contributions of this dissertation are in the area of identification of nonlinear Volterra models from input-output data. Historically, the identification of Volterra models has been limited to lower order models because of the large amount of model parameters that need to be identified. By using the Laguerre polynomials, the number of model parameters can be greatly reduced, which limits the required input-output tests. The goal of this dissertation is to move nonlinear multivariable control closer to industrial application by addressing practical model identification questions. Results from three test cases are presented and discussed. The results have shown a decrease in parameters of as much as 99% without a significant loss in model fidelity.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>ix</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>xi</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Chapters</td>
<td></td>
</tr>
<tr>
<td>I  BACKGROUND</td>
<td>5</td>
</tr>
<tr>
<td>Linear Multivariable Model Predictive Control</td>
<td>7</td>
</tr>
<tr>
<td>Volterra Series</td>
<td>14</td>
</tr>
<tr>
<td>Volterra Nonlinear Model Predictive Control</td>
<td>16</td>
</tr>
<tr>
<td>Laguerre Polynomials</td>
<td>18</td>
</tr>
<tr>
<td>II LINEAR IDENTIFICATION USING VOLterra-LAGUERRE REDUCTIONS</td>
<td>21</td>
</tr>
<tr>
<td>III NONLINEAR IDENTIFICATION USING VOLterra-LAGUERRE REDUCTIONS</td>
<td>32</td>
</tr>
<tr>
<td>Testing Procedure Issues</td>
<td>37</td>
</tr>
<tr>
<td>Input Sequence Form</td>
<td>38</td>
</tr>
<tr>
<td>Identification of SISO Second Order System</td>
<td>40</td>
</tr>
<tr>
<td>IV CASE STUDY I: ISOTHERMAL FREE RADICAL POLYMERIZATION CSTR</td>
<td>46</td>
</tr>
<tr>
<td>Linear Volterra-Laguerre</td>
<td>47</td>
</tr>
<tr>
<td>Second Order Volterra-Laguerre</td>
<td>56</td>
</tr>
<tr>
<td>V  CASE STUDY II: VAN DER VUSSE REACTION</td>
<td>64</td>
</tr>
<tr>
<td>Linear Volterra-Laguerre</td>
<td>68</td>
</tr>
<tr>
<td>Second Order Volterra-Laguerre</td>
<td>75</td>
</tr>
<tr>
<td>VI CASE STUDY III: IDENTIFICATION OF EXPERIMENTAL PH DATA</td>
<td>83</td>
</tr>
<tr>
<td>Procedure</td>
<td>85</td>
</tr>
<tr>
<td>Determination of Process Test Conditions</td>
<td>87</td>
</tr>
<tr>
<td>Process Step Testing</td>
<td>88</td>
</tr>
<tr>
<td>Data Conditioning</td>
<td>90</td>
</tr>
<tr>
<td>Mean Center and Scale the Input and Output Data</td>
<td>92</td>
</tr>
<tr>
<td>Model Identification</td>
<td>93</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Block diagram of multivariable IMC</td>
<td>6</td>
</tr>
<tr>
<td>2.</td>
<td>Volterra-Laguerre fits for identifications based on 1&lt;sup&gt;st&lt;/sup&gt; order Volterra kernel</td>
<td>28</td>
</tr>
<tr>
<td>3.</td>
<td>Laguerre approximation of Volterra system</td>
<td>30</td>
</tr>
<tr>
<td>4.</td>
<td>Linear Volterra-Laguerre approximation of process with 3 of 15 intervals deadtime</td>
<td>30</td>
</tr>
<tr>
<td>5.</td>
<td>Typical input step sequence – 100 Steps</td>
<td>39</td>
</tr>
<tr>
<td>6.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process - R = 2 (2 linear coefficients and 3 2&lt;sup&gt;nd&lt;/sup&gt; order coefficients.)</td>
<td>42</td>
</tr>
<tr>
<td>7.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process - R=3 case (3 linear coefficients 6 2&lt;sup&gt;nd&lt;/sup&gt; order coefficients)</td>
<td>42</td>
</tr>
<tr>
<td>8.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process - R=4 case (4 linear coefficients 10 2&lt;sup&gt;nd&lt;/sup&gt; order coefficients)</td>
<td>43</td>
</tr>
<tr>
<td>9.</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; order Volterra process step test data with noise</td>
<td>43</td>
</tr>
<tr>
<td>10.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process ID with noise - R=2 case</td>
<td>44</td>
</tr>
<tr>
<td>11.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process ID with noise - R=3 case</td>
<td>45</td>
</tr>
<tr>
<td>12.</td>
<td>Volterra 2&lt;sup&gt;nd&lt;/sup&gt; order process ID with noise - R=4 case</td>
<td>45</td>
</tr>
<tr>
<td>13.</td>
<td>Hardware configuration for polymerization CSTR</td>
<td>49</td>
</tr>
<tr>
<td>14.</td>
<td>Step response curves CSTR</td>
<td>49</td>
</tr>
<tr>
<td>15.</td>
<td>Improvement of linear Volterra-Laguerre</td>
<td>51</td>
</tr>
<tr>
<td>16.</td>
<td>Standard deviation based on # of steps</td>
<td>52</td>
</tr>
<tr>
<td>17.</td>
<td>Process data for 20 step identification data set with noise</td>
<td>53</td>
</tr>
<tr>
<td>18.</td>
<td>Volterra-Laguerre linear improvement for process with noise</td>
<td>53</td>
</tr>
<tr>
<td>19.</td>
<td>Volterra and Volterra-Laguerre linear predictions on process with noise</td>
<td>54</td>
</tr>
<tr>
<td>20.</td>
<td>Linear Volterra-Laguerre dependence on time scale factor</td>
<td>55</td>
</tr>
<tr>
<td>21.</td>
<td>Volterra-Laguerre 2&lt;sup&gt;nd&lt;/sup&gt; order error reduction</td>
<td>57</td>
</tr>
<tr>
<td>22.</td>
<td>Volterra-Laguerre 2&lt;sup&gt;nd&lt;/sup&gt; order error reduction</td>
<td>58</td>
</tr>
</tbody>
</table>
50. PH process ID Volterra-Laguerre improvement ................................................................. 96
51. PH identification fit with Laguerre order of four ............................................................. 96
NOMENCLATURE

a  Time scale factor for Laguerre expansion

D  Disturbance variable value

e  Difference between desired trajectory and actual trajectory

gi  i'th order impulse coefficients

m  Time intervals to horizon of fading system

p  Time horizon of fading system

u  input (independent) variable value

hi  i'th order Volterra kernel

N  Order of the Volterra series

r  Order of the Laguerre approximation

li  Continuous Laguerre operators

Li  Discrete Laguerre operators

t  Time

T  Time step size

y  Control variable value

y  Desired control variable value or control trajectory

y  Estimated control variable value

G  Impulse coefficient matrix

H  Impulse coefficient matrix for prediction of past effects

P  General nonlinear operator

d  Unmeasured disturbance vector
\( u \) Vector of future manipulated variable moves

\( \beta \) MV move penalty matrix

\( \sigma \) Delta time

\( \theta \) Real Laguerre coefficients, Combined impulse coefficient matrix

\( \Phi \) Combined impulse coefficient matrix for past moves, Laguerre model matrix

\( \tau_{\text{ref}} \) Reference trajectory

\( \Gamma \) CV weighting matrix
ACKNOWLEDGEMENTS

I would like to acknowledge and give thanks to Dr. Philip Smith for his never-ending support and mentoring in school and in life. I would also like to acknowledge the support of my long-suffering wife, Catherine.
INTRODUCTION

The petroleum and petrochemical industries have long recognized the value of good process control in solving many of the problems facing the daily operation of process plants. Processes are complex and interactive with many manipulated variables and competing control objectives. Advanced computer control is used to stabilize these processes' operation, thereby increasing efficiency while minimizing pollution and maximizing profit. Current industrial state of the art techniques utilize linear multivariable process control (MPC) in conjunction with unit optimizers to achieve these goals. Unfortunately, many of the most difficult processes are not linear. As a result, the attempt is made to apply the control over a very narrow operating region where the process can be treated as linear. Unit optimizers often move the process out of the region where the models were originally identified. These problems have spurred the need for improved control techniques that can be easily implemented in industry for complex nonlinear processes.

Linear multivariable control techniques have been widely accepted in industry as a tool to control important constrained, interactive systems. The appeal of these control techniques throughout industry has stemmed from their ease of use and ability to handle interactive constrained systems. Linear models are easily identified with process step tests and linear multivariable control problems are easily solved using linear matrix math. There is no requirement to have even a simple fundamental model of the process. Thus, linear MPC has become a standard general purpose tool used throughout industry.

There are several commercial multivariable controllers available (DMCplus - Aspentech, Profit Plus : RMPCT - Honeywell, and others). These controllers are internal model-based and
provide significant benefit over traditional single input single output (SISO) techniques. The tools for implementing these controllers are well developed and make implementation straightforward. These controllers all use linear models to represent the process being controlled. The linear models for control are obtained from step tests and input/output (I/O) data from the process. These linear controllers work well for many processes over a limited range. Their best performance is around the point of process testing and identification. As the nonlinearity of the process increases, the range of acceptable control using these techniques decreases. In addition, due to process constraints, the final operating point of these controls is often at a different point than where testing was initially done. The result is significant mismatch between the model and the process at the true operating point of the controller. Model and process mismatch can also be caused by the use of on-line plant optimizers. Optimizers (depending on plant economics) often drive individual controllers to operate in regions different than the point of original process testing. Since linear models are identified at one point, this also may cause significant mismatch between the model and the process, ultimately contributing to poor control.

Nonlinear controllers can fit nonlinear processes over a much broader range, thus solving many of the control problems associated with linear controllers. However, nonlinear controllers have traditionally had problems of their own. The mathematics involved in moving from linear to nonlinear models is significant. Unlike linear techniques, nonlinear process models are difficult to invert (an important step in implementation of control) and invertability is not usually guaranteed. Identification of nonlinear models from plant I/O data has also proven to be very difficult. Some success with nonlinear control has been obtained where first principle models (nonlinear differential equations) are available, but these cases are specific to individual pieces of equipment and thus are not attractive as a general tool (Gerstle et al., 1991, Zafiriou et al., 1991). What is needed in industry is a general form nonlinear controller that could extend the range of control of nonlinear processes significantly over that of linear MPC algorithms. In order to be
practical and general, the identification of the nonlinear model parameters would need to be obtained from process experimental testing and I/O data over a suitable operating range.

This dissertation presents an extension of work done in the area of nonlinear control using Volterra models (Al-Baiyat et al., 1986, Doyle et al., 1995, Maner et al., 1994, and Pearson et al., 1992). The Volterra approach has the advantage of using a linear formulation with iterative corrections for higher order nonlinearities. It allows for the use of the linear matrix inversion techniques that are well developed and understood while accounting for process nonlinearities. A key step in the implementation of any model-based control system is process identification. As a general tool used in industry, process parameters for these models must be obtained from plant I/O data. In the past, the number of parameters necessary to describe nonlinear Volterra systems has been prohibitive. This dissertation also builds upon the work of Zheng and Zafirou (1995) where orthogonal Laguerre functions are used to reduce the number of coefficients required for nonlinear models to a feasible number.

The significant contributions of the work in this dissertation are in the area of identification of industrial nonlinear Volterra models from input-output data. Until now, the application of nonlinear multivariable control using higher order Volterra models has been limited to cases where the Volterra model could be identified from fundamental physics-based models or on systems that could be tested ad infinitum to determine higher order Volterra coefficients. Using the approach described in this dissertation, nonlinear models end up in the form of second order Volterra models that can be used in conjunction with work done by researchers such as Doyle et al. (1995), Maner et al. (1994), and Pearson (1993). The result of combining the work done in this dissertation and the work done by the researchers above is a generic nonlinear multivariable controller identified from process step testing and input-output data. This project provides a methodology for the practical implementation of second order nonlinear multivariable control using the existing framework used by industry in linear multivariable control.
A significant benefit demonstrated in this work is the ability to identify models using I/O data for a broad range of nonlinear processes. Difficult process characteristics such as deadtime, inverse response, nonlinear dynamics, and changing gains including gain sign changes have been demonstrated.

One goal of this dissertation is to move this nonlinear multivariable control technology toward industrial application by addressing practical model identification questions. Such questions include identification of second order models from I/O data, experimental test requirements for model identification, and nonlinear parameter estimation. Since most processes in industry are nonlinear to some degree, it is expected that this nonlinear controller will significantly improve control in many industrial applications. Model improvements eliminating up to 90% of the error from linear models have been demonstrated for systems tested in this work. In addition, models for processes with gain sign changes have been demonstrated.
CHAPTER I

BACKGROUND

Model predictive control (MPC) is essentially defined as any control algorithm that uses models to predict the future behavior of a process and generates control moves based on that prediction to control the future behavior of the process. Linear model predictive control has been widely used in industry for some time. The appeal of these control techniques on industrial processes has stemmed from their ease of use and ability to handle interactive constrained systems. Linear models are simple to identify and linear multivariable control problems are easily solved using linear matrix math. The first industrial applications incorporated two separate but similar formulations. Dynamic matrix control (DMC) was presented by Cutler and Ramaker (1980) and Prett and Gillette (1980). Model algorithmic control (MAC) was first presented by Richalet et al. (1978). Garcia and Morari (1982) first pointed out the similarities of DMC and MAC approaches. Garcia and Morari also grouped these methods into a formulation they called internal model control (IMC). A block diagram of the IMC formulation is shown in Figure 1.

These approaches have found widespread industrial acceptance (Cutler and Yocum, 1991; Gerstle and Hokanson, 1991; Grosdidier et al., 1992; Richalet et al., 1978) and are the basis for industrial multivariable control packages available today (DMCplus - Aspentech, Profit Plus : RMPCT - Honeywell, and others).

It can be seen from the block diagram in Figure 1 that with model predictive control, the model is in parallel with the process. The controller is essentially a feedforward controller, with only the mismatch between the model and the process in the feedback loop. In addition,
convolution models are used and the result is a controller with several appealing characteristics, including the following:

- No parametric form of the process model is assumed. Therefore, little prior process knowledge is needed before implementing control. In addition, control algorithm application is independent of the process order. Thus, high order and complex processes where little is known about the physical laws governing the system can be modeled and controlled using the same procedures as those used for low order processes.

- Linear process models for these controllers can be obtained from time series analysis of I/O process data that do not require a significant fundamental modeling.

- The control approach handles nonminimum phase process behaviors such as dead time and inverse responses transparently.

- The control takes care of the interactive nature of multiple input multiple output (MIMO) systems optimally.

- Constraints on independent and dependent variables can be included in the control objectives.

- The control performance is often very robust. It has been shown that model gain variations of greater than four can be tolerated with only small changes in performance.
(Richalet et al., 1978). Other researchers have discussed robustness on a more general basis (Morari and Zafiriou, 1989; Muske and Rawlings, 1993).

Because of the many advantages of this control approach, research has been done on extending this formulation to the nonlinear realm. One promising approach for this extension of the IMC formulation uses Volterra series. Volterra series are power series first studied by the mathematician Vito Volterra in the 1880s (Schetzen, 1980). The Volterra series approach essentially extends the convolution integral approach currently used in many industrial linear MPC packages to include higher order models. In recent years, there has been substantial research on the use of these higher order Volterra series in control (Doyle et al., 1995; Maner, 1993; Maner et al., 1994; Pearson, 1992)

**Linear Multivariable Model Predictive Control**

Linear MPC will be reviewed here, as the proposed nonlinear MPC is an extension of and has its basis in work done previously in linear MPC. There are several good review articles on linear multivariable control (Garcia et al., 1989; Marchetti et al., 1983; Ricker, 1985; and Rouhani and Mehra, 1982). Below is a brief review of the MAC formulation theory. Linear continuous processes have been modeled using equations of the following form:

\[ y_1(t) = \int_0^\infty g_1(\sigma)u(t - \sigma)\,d\sigma \]  

(1.1)

This equation is the linear convolution integral and also represents the continuous first order Volterra kernel (Schetzen, 1980). If the system is causal, time invariant, and has fading memory, the equation can be approximated in the discrete time domain by the finite sum:

\[ y(t) = \sum_{i=1}^{m} g_i u(t - i) \]  

(1.2)
The $g_i$'s represent the impulse response coefficients of the process. $y(t)$ is the predicted output at time $t$ and $u(t - i)$ is the input to the plant at time $(t - i)$. The series has been truncated at $m$ terms because the system has fading memory. This means that effects caused by changes in independent variables at some point in time fade to approximately zero and thus, the process can be approximated with a finite series. The time horizon can be defined mathematically as $mT = p$ where $p$ is the time horizon of the process. This represents the time for input effects to fade to approximately zero. For systems that do not fade, such as integrating processes, this truncated formulation will not work. Equation (1.2) is the fundamental prediction equation for the MAC control technique discussed earlier, and also represents the first order kernel of the Volterra series. In matrix form, equation (1.2) can be rewritten as:

$$y_m = \tilde{G}u$$

(1.3)

where, $\tilde{G}$ is the first order impulse coefficient matrix model approximating the real process $G$. $y_m$ is the model predicted output over one time horizon, and $u$ is the vector of inputs over one time horizon. For the SISO case, the overall prediction equation can be written as:

$$y_m = \tilde{G}u + \tilde{H}u_{past} + d$$

(1.4)

which states that the predicted future process output over one time horizon $y_m$ is calculated based on the effect of future moves, $\tilde{G}u$ term, plus the effect of past moves of both manipulated and measured disturbance variables, $\tilde{H}u_{past}$ term, plus an unmeasured disturbance vector $d$.

The $\tilde{G}$ and $\tilde{H}$ are dynamic impulse matrices built from impulse coefficients as:
In the SISO case, \( u \) and \( u_{\text{past}} \) are column vectors containing one time horizon \( p \) of future and past independent variable positions, respectively. The vector \( d \) is a column vector of expected future disturbances. In most circumstances, \( d \) is a constant vector containing the most recent mismatch between the model and actual plant output. This assumes that the unmeasured disturbance will stay constant and the best estimate of the disturbance is the current value. It is through the unmeasured disturbance vector that the feedback action of the controller is implemented. In some cases, a filter on this feedback term called a “bias filter” is added to reduce noise and help stabilize the controller.

In order to achieve stable control, it is sometimes desirable to have the process follow some trajectory \( y^* \) which is usually defined as the filtered setpoint or deviation that would return to the setpoint \( y_{sp} \). If you assume a first order time constant for the closed loop reference trajectory, \( \tau_{\text{ref}} \), the setpoint trajectory can be calculated as:

\[
\tilde{G} = \begin{bmatrix}
g_1 & 0 & 0 & \cdots & 0 \\
g_2 & g_1 & 0 & \cdots & 0 \\
g_3 & g_2 & g_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
g_m & g_{m-1} & g_{m-2} & \cdots & g_1
\end{bmatrix}
\]

\[
\tilde{H} = \begin{bmatrix}
g_2 & g_3 & g_4 & \cdots & g_m & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
g_{m-2} & g_{m-1} & g_m & \ddots & 0 & 0 \\
g_{m-1} & g_m & 0 & \cdots & 0 & 0 \\
g_m & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\]
\[ y^*(k+1) = \left(1 - \frac{1}{1 - e^{-\tau_{ref}}} \right)y^*(k) + \left(\frac{1}{1 - e^{-\tau_{ref}}} \right)y_{sp} \]  

(1.7)

where \( y_{sp} \) is the desired setpoint value and \( T \) is the sample time. In this equation, \( \tau_{ref} \) becomes one of the primary tuning parameters of the controller. Without \( \tau_{ref} \), or some other way to dampen the control action, the controller would be too aggressive and very sensitive to model mismatch. Other methods for dampening the control action are shown by Ricker (1985). One of these methods places a penalty on manipulated variable movement. This technique will be discussed later. With \( y^* \) defined, an error vector can be represented as:

\[ \text{err} \equiv y^* - \tilde{H}u_{past} - d \]  

(1.8)

Here, \( \text{err} \) represents the error between the desired trajectory and the actual trajectory if no further control moves are implemented.

In unconstrained control, the final step is to calculate the optimal future control moves \( u_{opt} \) that will minimize the error. The least squares solution to this problem is:

\[ u_{opt} = (\tilde{G}^r \tilde{G})^{-1} \tilde{G}^r \text{err} \]  

(1.9)

For the multivariable case, the equations are expanded as follows:

\[ \theta \equiv \begin{bmatrix} \tilde{G}_{11} & \cdots & \tilde{G}_{1m_j} \\ \vdots & \ddots & \vdots \\ \tilde{G}_{cl} & \cdots & \tilde{G}_{cm_j} \end{bmatrix} \]  

(1.10)
\[
\Phi \equiv \begin{bmatrix}
\tilde{H}_{11} & \cdots & \tilde{H}_{1f} \\
\vdots & \ddots & \vdots \\
\tilde{H}_{c1} & \cdots & \tilde{H}_{cf}
\end{bmatrix}
\] (1.11)

where \( m_f \) is the number of manipulated variables, \( c \) is the number of control variables, and \( I \) is the number of independent variables (manipulated plus measured disturbance). The dynamic impulse coefficient matrix, \( \Theta \), is used for future move calculations and \( \Phi \) is the dynamic matrix used to calculate control variable trajectories based on past independent variable moves. The MIMO control move equation then becomes:

\[
\text{err} = y^* - \Phi u_{\text{past}} - d
\] (1.12)

\[
u_{\text{opt}} = (\Theta^T \Gamma \Theta)^{-1} \Theta^T \Gamma \text{err}
\] (1.13)

where \( \Gamma \) is a diagonal weighting matrix. \( \Gamma \) then becomes another tuning parameter that is used to scale control variables based on units and relative importance of error. This is of particular value in the case of an over-constrained system. In that case, it may not be possible to obtain the desired set points of all control variables (CVs). The relative magnitude of CV's error will be set by the weighting factor. In the under-constrained case, equation (1.13) has an infinite number of solutions. In practice, another term is often included which will drive the manipulated variables to ideal resting values or minimize their movement. In this case, the problem to be solved is:

\[
\max_{u_{\text{opt}}} - \frac{1}{2} \left[ \text{err}^T \Gamma \text{err} + u^T \beta u \right]
\] (1.14)
where $\beta$ is a diagonal matrix containing penalties for movement of the manipulated variables from their nominal values. The least squares solution of this problem is:

$$u_{opt} = (\theta^T \Gamma \theta + \beta I)^{-1} \theta^T \Gamma \text{err}$$  \hspace{1cm} (1.15)

Techniques for adding constraints to MIMO systems have been described by Hanson (1986) and Ricker (1985). If at each sampling time we solve equation (1.13) subject to the constraints:

$$u_L \leq u \leq u_U$$  \hspace{1cm} (1.16)

$$y_L \leq y \leq y_U$$  \hspace{1cm} (1.17)

where $u_L$ and $u_U$ are the manipulated variable upper and lower constraints, respectively. If we define a new vector:

$$u^* = u - u_L$$  \hspace{1cm} (1.18)

we can rewrite the independent variable constraints as:

$$u^* \geq 0$$  \hspace{1cm} (1.19)

$$u^* \leq u_U - u_L$$  \hspace{1cm} (1.20)

In the same fashion, we can write the control variable constraints as:
\[-\theta \mathbf{u}^* \leq -\mathbf{y}_L + \theta \mathbf{u}_L + \phi \mathbf{u}_{\text{past}} + \mathbf{d}\]  \hspace{1cm} (1.21)

\[\theta \mathbf{u}^* \leq \mathbf{y}_U - \theta \mathbf{u}_L - \phi \mathbf{u}_{\text{past}} - \mathbf{d}\]  \hspace{1cm} (1.22)

These constraints are in the form:

\[
\mathbf{C} \mathbf{u}^* \leq \mathbf{r}
\]  \hspace{1cm} (1.23)

where

\[
\mathbf{C} = \begin{pmatrix}
\mathbf{I} \\
-\theta \\
\theta
\end{pmatrix}
\]  \hspace{1cm} (1.24)

and

\[
\mathbf{r} = \begin{pmatrix}
\mathbf{u}_U - \mathbf{u}_L \\
-\mathbf{y}_L + \theta \mathbf{u}_L + \phi \mathbf{u}_{\text{past}} + \mathbf{d} \\
\mathbf{y}_U - \theta \mathbf{u}_L - \phi \mathbf{u}_{\text{past}} - \mathbf{d}
\end{pmatrix}
\]  \hspace{1cm} (1.25)

Using these transformations, Maner (1993) has shown the new optimization equation as:

\[
\max_{\mathbf{u}} \left( \mathbf{a}^T \mathbf{u}^* - \frac{1}{2} \mathbf{u}^{*T} \mathbf{B} \mathbf{u}^* \right)
\]  \hspace{1cm} (1.26)

where

\[
\mathbf{a}^T = (\mathbf{y}^* - \phi \mathbf{u}_{\text{past}} - \mathbf{d})^T \Gamma \theta - \mathbf{u}_L^T \mathbf{B}
\]  \hspace{1cm} (1.27)

\[
\mathbf{B} = \theta^T \Gamma \theta + \beta
\]  \hspace{1cm} (1.28)
which has the unconstrained least squares solution:

\[ u^* = B^{-1}a \]  

(1.29)

Other simplifying steps are also used to simplify the calculation and increase the controllers robustness. Ricker (1985) describes a technique called blocking, which reduces the number of control moves calculated while also reducing the size of the matrices in the calculations.

In this discussion, the basic formulation for the constrained MAC controller has been developed. There are some simple but important differences in the MAC controller and the DMC control strategies. First, in industrial applications of the DMC controller, the desired trajectory of the controller, \( y^* \), is always the current setpoint of the controller. There is no first order filter applied to the setpoint. The controller is suppressed and stabilized with a diagonal MV weighting factor matrix using a move suppression factor \( \beta^2 \). Second, the DMC controller uses step coefficients rather than impulse coefficients in the dynamic matrix model of the plant. As a consequence, the control moves are calculated as change from the current position rather than absolute position.

**Volterra Series**

Volterra series are power series first studied by the mathematician Vito Volterra (Schetzen, 1980). Volterra series include time invariant series that take the form:
\[
y(t) = \int\int_{-\infty}^{\infty} g_1(\sigma)u(t-\sigma_1)d\sigma_1 + \int\int_{-\infty}^{\infty} g_2(\sigma_1, \sigma_2)u(t-\sigma_2)d\sigma_1 d\sigma_2 \\
+ \int\int\int_{-\infty}^{\infty} g_3(\sigma_1, \sigma_2, \sigma_3)u(t-\sigma_1)u(t-\sigma_2)u(t-\sigma_3)d\sigma_1 d\sigma_2 d\sigma_3 + \\
\cdots + \int\int\int\int_{-\infty}^{\infty} g_n(\sigma_1, \cdots, \sigma_n)u(t-\sigma_1)\cdots u(t-\sigma_n)d\sigma_1 \cdots d\sigma_n
\]  
(1.30)

If the system is causal with fading memory, the equation (1.30) can be discretized and shown as:

\[
y(t) = y_1(t) + y_2(t) + y_3(t) + \cdots
\]  
(1.31)

where

\[
y_1(t) = \sum_{i=1}^{m} g_1(i)u(t-i)
\]  
(1.32)

\[
y_2(t) = \sum_{i=1}^{m} \sum_{j=1}^{m} g_2(i, j)u(t-i)u(t-j)
\]  
(1.33)

\[
y_3(t) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} g_3(i, j, k)u(t-i)u(t-j)u(t-k)
\]  
(1.34)

\[\vdots\]

\[
y_n(t) = \sum_{j_1=1}^{m} \sum_{j_2=1}^{m} \cdots \sum_{j_n=1}^{m} g_n(j_1, \cdots, j_n)\prod_{j=1}^{n} u(t-j)
\]  
(1.35)

In this form, \( y_i(t) \) is the \( i \)th order Volterra kernel. Thus, \( y_1(t) \) is the first order Volterra kernel and is equivalent to the linear impulse response. \( g_i \) is the \( i \)th order operator which contains \( m^i \) coefficients. Therefore, if there is a one-by-one process model with a time horizon of 60 coefficients and a second order Volterra model is used, there are 1860 impulse coefficients required. For a third order Volterra model, 217,860 coefficients would be required. The main deterrent to using Volterra models for modeling nonlinear processes in industrial applications is
the large number of coefficients required. Nonetheless, there has been significant work done in
developing a nonlinear control technique using these models.

**Volterra Nonlinear Model Predictive Control**

There has been work done by Doyle et al. (1994) and Maner et al. (1996), using Volterra
operators as a means of separating a nonlinear control problem into a linear control calculation
and an auxiliary loop, which augments a nonlinear correction term. The major advantage to this
approach is that the need for a nonlinear inverse is eliminated. This removes a major stumbling
block in nonlinear control because inversion of the nonlinear model is very difficult and even
impossible in many cases. In addition, the nonlinear correction is a straightforward extension of
the linear model predictive control currently being used in industry. The controller is constructed
based on the linear inverse with nonlinear corrections that do not require inversion of the
nonlinear portion of the model. The approach is outlined below. In a process that can be
represented by a nonlinear I/O correlation:

\[ \tilde{y} = P[u] \tag{1.36} \]

where \( P \) is a nonlinear operator that operates on the plant inputs \( u \). If the operator can be broken
into a linear and nonlinear portion:

\[ P = L + N \tag{1.37} \]

then

\[ P = L(I + L^{-1}N) \tag{1.38} \]

and
\[ P^{-1} = (I + L^{-1}N)^{-1} L^{-1} \]  

(1.39)

This separation assumes that the linear inverse \( L^{-1} \) and the inverse of \( (I + L^{-1}N) \) exist.

Both the Volterra and Volterra-Laguerre models discussed earlier fit into the category of the overall nonlinear model being represented by a linear plus nonlinear operator. As stated earlier, the linear operator of the Volterra model is the same form used in some of the commercial linear controllers. If we substitute equation (1.36) into (1.31) we get:

\[ P = P_1 + P_2 + P_3 + \cdots + P_n \]  

(1.40)

where the \( n \)'th order Volterra or Volterra-Laguerre expansion is represented as \( P_n \). This now fits the form of equation (1.37) with \( P_1 = L \) and \( N \) represents the sum of all nonlinear operators. If \( y \) represents the actual output and \( y^* \) is the desired trajectory, the error can be defined as:

\[ e = y^* - y \]  

(1.41)

In this case, the error \( e \) contains the effects caused by the combination of model mismatch and unmeasured disturbances. Combining equation (1.41) and equation (1.36) gives:

\[ y = P[u] + e \]  

(1.42)

The desired control action can be determined by the solution of:

\[ \min_u \varphi = \| y^* - y \| \]  

(1.43)
which can be shown as:

$$u_{opt} = P^{-1}[y^* - e]$$  \tag{1.44}$$

if

$$\xi = y^* - e$$  \tag{1.45}$$

Substituting equations (1.44) and (1.45) into equation (1.38) the overall control equation for the nonlinear case becomes:

$$u_{opt} = \left( I + P_1^{-1}P_2 + P_1^{-1}P_3 + \cdots + P_1^{-1}P_n \right)^{-1} P_1^{-1} [\xi]$$  \tag{1.46}$$

Once again, this solution assumes that the linear inverse and the inverse of the composite term can be obtained. Thus far, it has been assumed that the matrices $P$, $L$, and $N$ are square. However, in a typical multivariable problem, the matrices are rarely square. Doyle et al. (1994) showed that generalized inverses can be used instead of the actual inverse, yielding the same result. Nonsquare systems are therefore not a problem. Equation (1.46) represents the optimum control for a square system with no manipulated variable or control variable constraints. In the nonsquare system, all inverses are replaced with pseudo inverses and the result is the least-squares optimum of the control problem.

**Laguerre Polynomials**

There has been work done with Laguerre functions (Dumont and Fu, 1993; and Zheng and Zafiriou, 1995) in an effort to reduce the number of coefficients required to obtain a Volterra representation. Laguerre functions are orthonormal functions that can be used to approximate
Volterra series. The advantage of the Laguerre functions is that they are orthogonal and all coefficients of the discrete series are independent of all other coefficients. As a result, a selection process can be used to eliminate coefficients with small contributions to the model without the need to recalculate the remaining coefficients. Thus, the number of coefficients required to approximate a nonlinear Volterra system can be greatly reduced.

Laguerre series are obtained by orthonormalizing the power series (Lee, 1960):

\[(at)^n e^{-at} \text{ for } n = 0, 1, 2 \cdots\]

The resulting form is:

\[l_0(t) = \sqrt{2a} e^{-at}\] \hspace{1cm} (1.47)

\[l_1(t) = \sqrt{2a} (2at - 1) e^{-at}\] \hspace{1cm} (1.48)

\[l_2(t) = \sqrt{2a} (2a^2t^2 - 4at + 1) e^{-at}\] \hspace{1cm} (1.49)

\[\vdots\]

\[l_n(t) = \sqrt{2a} \left[ \frac{(2a)^n}{n!} t^n - \frac{n(2a)^{n-1}}{(n-1)!} t^{n-1} + \frac{n(n-1)(2a)^{n-2}}{2!(n-2)!} t^{n-2} \right]\] \hspace{1cm} (1.50)

This form is orthonormal over the domain \([0, \infty)\). The time scale factor "a" is a tunable parameter that should be set near the dominate root of the system (Wahlberg, 1991; Zheng and Zafiriou, 1995). The time scale factor will be discussed in more detail later. A sampled data system can be expressed as:
\[ f(k) = \sum_{i=1}^{\infty} \theta_i \cdot l_i(k) \]  

(1.51)

where \( \theta_i \) are the real Laguerre coefficients calculated as:

\[ \theta_i = \sum_{k=0}^{\infty} f(k) \cdot l_i(k) \]  

(1.52)

In practice, \( f(k) \) can be approximated using a finite series of Laguerre functions. In the case of the Volterra-Laguerre approximation, the Laguerre equations are used to approximate the Volterra kernel. The problem in identifying these are that the Volterra kernel is not known and identifying the Volterra kernel for higher order systems is not practical because there are too many coefficients to be identified. If it were practical to identify the Volterra kernel directly, Laguerre reductions would not be necessary. Thus, the approach used here is to use Laguerre functions and input-output data to approximate the Volterra kernel for the system.
CHAPTER II

LINEAR IDENTIFICATION USING VOLterra-LAGUERRE REDUCTIONS

The Identification step is usually the most difficult step in implementing linear MPC. Identification is even more difficult in nonlinear processes. As stated previously, one of the main advantages of using the Volterra-Laguerre model is that the number of coefficients required to describe the system is greatly reduced from the Volterra model alone. Work has been done on the identification problem for the Volterra model by Nowak et al. (1994), Pearson et al. (1992 and 1993), and Zheng et al. (1994). Because of the number of coefficients required, none of the papers suggest that it is practical to attempt identifying high order Volterra models from I/O data and all of the papers reviewed limit their approach to second order Volterra models. For Volterra-Laguerre models, there has been work done by Dumont et al. (1991 and 1993), Zervos et al. (1988), and Zheng et al. (1995). Zheng demonstrated the advantage of reduction of coefficients in identification and discussed control relevant identification techniques. The advantage of using Laguerre series comes from the reduction of parameters necessary to represent the system. An additional advantage is the theoretical ease with which the orthogonal coefficients can be determined. This discussion will be limited to Laguerre series which are complete in $L_2[0,\infty)$ space. It is also assumed that the process being modeled, the impulse coefficient model $\vec{g}$, is $L_2$ stable, i.e.:
This is the case for fading memory, self regulating systems. Based on this, the Riesz-Fisher theorem (Dumont et al., 1991) states that the process (in this case linear Volterra model) can be approximated as:

\[ \hat{g} = \sum_{i=1}^{\infty} \theta_i \cdot l_i(k) \]  

(2.2)

and for any positive, \( \varepsilon \) there is an \( n \) such that:

\[ \int_{0}^{\infty} \left[ g(t) - \sum_{i=1}^{n} \theta_i \cdot l_i(k) \right]^2 dt < \varepsilon \]  

(2.3)

where \( \theta \)'s are the Laguerre coefficients, \( \hat{g} \) is the approximated Volterra model, and \( \varepsilon \) represents the acceptable integrated error.

Dumont et al. (1991) and Zheng et al. (1995) proposed two ways to determine the coefficients \( \theta \). The first is called the correlation method and is a natural consequence of orthogonality. In this case, the \( i \)'th Laguerre coefficient can be calculated as:

\[ \hat{\theta}_i = \frac{1}{n} \sum_{k=1}^{n} g(kT)l_i(kT) \]  

(2.4)

This equation assumes that the process input is Gaussian white noise. In the technique being proposed, the process being represented is the first order Volterra model so an input of Gaussian noise has little meaning. Also assumed in this technique is that there are no
unmeasured disturbances or any other correlated or uncorrelated noise, and that the time scale factor in the Laguerre equations is already known. The Laguerre time scale factor will be discussed later. This approach makes little sense for approximation of the Volterra kernel.

The second approach presented by Dumont (1991) is the least-squares approach. In this approach, the Volterra kernel is represented as:

\[
\hat{g} = \Phi^T \theta + \Phi_u^T \theta_u + W
\]  

(2.5)

where the first term on the right hand side represents the model, the second term represents a similar Laguerre expansion for the unmodeled dynamics, and the third represents the effect of random noise on the process. Since there is no structure known about the unmodeled dynamics or the random noise, the best that can be done is assume that these will be zero in the mean. Because of this, Dumont goes on to determine that the expectation of the second and third terms of equation (2.5) go to zero. In the above equation, the parameters are defined as:

\[
\phi^T = [l_1(kt), \ldots, l_n(kT)]
\]  

(2.6)

\[
\phi_u^T = [l_1(kT), \ldots, l_n(kT)]
\]  

(2.7)

\[
\Phi^T = [\phi(T), \ldots, \phi(nT)]
\]  

(2.8)

\[
\Phi_u^T = [\phi_u(T), \ldots, \phi_u(nT)]
\]  

(2.9)

\[
\hat{g}^T = [\hat{g}(T), \ldots, \hat{g}(nT)]
\]  

(2.10)

The least squares estimate of \(\theta\) becomes:

\[
\hat{\theta} = [\Phi \Phi^T]^{-1} \Phi \hat{g}
\]  

(2.11)
In the previous section, it was assumed that the Volterra kernel is already known. This is not the case in industrial processes. In the Volterra-Laguerre form, the above technique is used to approximate the Volterra-Laguerre kernel from I/O data. Zheng et al. (1995) presented the correlation method approach to estimate the Volterra-Laguerre kernel. Zheng did not discuss the requirements of the identification based on I/O data in his paper. In addition, Zheng assumed the time scale factor in the Laguerre equations had already been determined, making the problem linear in parameter identification. The input excitation signal to the process has been assumed to be Gaussian white noise to this point. However, within industrial processing plants, normally Gaussian random noise inputs are not practical and not allowed. The time scale factor and the input step requirements will be discussed in more detail later.

The first step in the evaluation of Laguerre reductions is to test Laguerre functional approximations for Volterra kernels of linear systems.

\[
\underline{g} = \Phi^T \Theta
\]  

(2.12)

\(\underline{g}\) is a vector with the Volterra first order model for one time horizon into the future. Equation (2.12) shows the approximated Volterra kernel as opposed to the actual Volterra kernel. In the industrial process identification problem, the Volterra kernel is not known, and identifying it directly is impractical. Much of the derivation in the previous section assumed the true Volterra kernel was available. From this point on, \(\underline{g}\) will represent the Volterra kernel identified from process I/O data based on the Volterra-Laguerre form.

\[
\phi = [l_0(kt), \ldots, l_r(kT)]
\]  

(2.13)

\[
\Phi^T = [\phi(T), \ldots, \phi(pT)]
\]  

(2.14)
In this equation, \( p \) is the number of coefficients in one time horizon as defined by the Volterra model and \( r \) is the number of Laguerre terms used in the approximation. In expanded form:

\[
\Phi = \begin{bmatrix}
l_0(0) & l_1(0) & l_2(0) & \cdots & l_r(0) \\
l_0(T) & l_1(T) & l_2(T) & \cdots & l_r(T) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
l_0(pT) & l_1(pT) & l_2(pT) & \cdots & l_r(pT)
\end{bmatrix}
\] (2.15)

In Equation (2.15), \( \Phi \) is a transformation matrix that maps the Laguerre coefficients to Volterra coefficients. The result is a reduction in the number of coefficients required to represent the system from \( p \) to \( r \) where \( p \) is the number of coefficients in the time horizon of the Volterra model and \( r \) is the order of the Laguerre approximation. Recognizing the first order Volterra model as:

\[
y(t) = \sum_{i=1}^{p} g_i u(t - i)
\] (2.16)

The prediction equation for one point becomes:

\[
y(k) = \bar{u}_k \cdot \Phi \cdot \theta
\] (2.17)

where

\[
\bar{u}_k = [u(k), u(k-1), \ldots, u(k-p)]
\]

Equation (2.17) gives the prediction at the \( k \)'th interval. What is needed for optimal control is the control move calculation for one full time horizon into the future. In order to keep
the Volterra-Laguerre form, there must be a matrix of moves $U$. In addition, the future prediction of the process value must be broken into two predictions. One prediction is based on past moves and the other prediction is based on future moves as done in the linear MPC approach discussed in Chapter 1. The total prediction for one time horizon in the future becomes the sum of the predictions based on past independent variable moves and moves the for one time horizon in the immediate future. The independent variable move matrices and process predictions take the form:

$$
U_{\text{past}} = \begin{bmatrix}
0 & u_0 & \cdots & u_{p-2} & u_{p-1} \\
0 & 0 & \cdots & u_{p-2} & u_{p-1} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
\vdots & & \ddots & u_0 & \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
$$

(2.18)

$$
Y_{\text{past}} = U_{\text{past}} \cdot \Phi \cdot \theta
$$

(2.19)

$$
U_{\text{fut}} = \begin{bmatrix}
u_1 & 0 & \cdots & 0 & 0 \\
u_2 & u_1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
\vdots & & \ddots & u_1 & 0 \\
u_p & u_{p-1} & \cdots & u_2 & u_1
\end{bmatrix}
$$

(2.20)

$$
Y_{\text{fut}} = U_{\text{fut}} \cdot \Phi \cdot \theta
$$

(2.21)

To complete the identification of the Laguerre coefficients directly from I/O data it is necessary to group terms:

$$
Y_{\text{past}} = U_{\text{past}} \cdot \Phi \cdot \theta
$$

(2.22)
let

\[ A = U_{past} \cdot \Phi \]  \hspace{1cm} (2.23)

which leads to:

\[ Y_{past} = A \cdot \theta \]  \hspace{1cm} (2.24)

Given an input-output data set \((u, y)\), the Laguerre coefficients \(\theta\) can be approximated by the least squares fit. If \(U\) is sufficient for the identification step, this \(\theta\) should provide reasonably accurate predictions for any input \(U\) in the prediction equations. At this point, it is assumed that the time scale factor for the Laguerre equations is known.

Figure 2 shows example plots of linear predictions based on a first order plus deadtime process model. The comparison is between an impulse coefficient model, Volterra-Laguerre where the Laguerre coefficients were approximated from the Volterra kernel, and direct calculation of the Laguerre coefficients from I/O data. The input for the plots has a step up at interval two and a step down at interval seven. The final plot has random input. As can be seen, a relatively few number of Laguerre terms produce reasonably good predictions. The exponential time scale factor used in the Laguerre equations was determined from a nonlinear least squares optimization. When the time scale factor is not optimal, the Laguerre model still converges to the actual, but more Laguerre terms are required to obtain an equivalent fit.

The next step in this identification process is to use equation (2.24) with large amounts of I/O data to generate an accurate set of Laguerre coefficients. To accomplish the identification, equation (2.24) is used with the entire I/O data set. \(Y_{past}\) is the vector containing all of the output data. \(A\) is an \(nxr\) matrix, where \(n\) is the number of points in the data set and \(r\) is the order of the
Figure 2 Volterra-Laguerre fits for identifications based on 1st order Volterra kernel
Laguerre approximation. Note that $A$ is dependent on $U_{past}$. It is reasonable to expect that the quality of the least squares fit is also dependent on the form of $U_{past}$. For this work, the input signal will be pseudo-random binary sequences (PRBS), based on a paper by Pearson et al. (1993). To illustrate this identification, normally distributed random noise was imposed on a 15 coefficient first order Volterra model to simulate the process. The process had no deadtime. A new random set of input data was generated and the identified Volterra-Laguerre model was compared to the simulated process. In Figure 3, the solid line represents the process, and the “+” marks illustrate the Volterra-Laguerre approximation. As can be seen, if the process has no dead time and first order dynamics, it does not take many Laguerre coefficients to represent the system.

In Figure 3, the simulated process was changed to have 3 deadtime intervals out of 15. Again, the I/O data used for the identification had normally distributed random noise imposed upon it. As a result, more Laguerre terms are required to adequately approximate the process. The number of coefficients required to achieve reasonable results is still significantly less than that of the Volterra model. This illustrates the advantage and practicality of using Laguerre equations. The ability to approximate deadtime is based on the Laguerre equations similarity to Padé approximations for deadtime. Figure 3 and Figure 4 illustrate that the Laguerre approximation provides good results for linear systems. The ability to model deadtime is one of the most important requirements for models used in MPC applications in chemical industries.

Processes without deadtime can often be controlled with simple feedforward schemes such as lead-lag and interactive moves. In addition, deadtime is a major contributor in the failure of simple PID feedback control. The fundamentals of data regression dictate that if fewer coefficients are used in the model, then something is sacrificed in the regression model results. With the Volterra model, there is no assumption about the dynamics or smoothness of the process. With the Laguerre approximation, there is some smoothing effect. This smoothing
Figure 3  Laguerre approximation of Volterra system

Figure 4  Linear Volterra-Laguerre approximation of process with 3 of 15 intervals deadtime
works well if the process is relatively smooth or the randomness is caused by noise in the system but, if the process is repeatable and consistently erratic, Volterra-Laguerre equations may not be the optimal approach.
CHAPTER III

NONLINEAR IDENTIFICATION USING VOLTERRA-LAGUERRE REDUCTIONS

The Identification step in implementing linear MPC is typically the most difficult step. Identification of nonlinear processes is even more difficult. This difficult identification is the defining factor as to why nonlinear MPC has not been widely implemented throughout industry. The number of step tests required for regression-based nonlinear models has been prohibitive. As stated earlier, one of the main advantages of using the Volterra-Laguerre model is that the number of coefficients required to describe the system is greatly reduced from the Volterra model alone. Although the results will not be discussed here, there has been work done on the identification problem for the Volterra model by Pearson et al. (1992 and 1993), Nowak et al. (1993 and 1994), and Zheng et al. (1994). None of these papers suggest that it is practical to attempt identifying high order Volterra models from I/O data because of the number of coefficients required. The work done in these papers has been limited to obtaining Volterra coefficients from first principle models for individual systems. There has been work done by Dumont et al. (1991 and 1993), Zervos et al. (1988), and Zheng et al. (1995) for Volterra-Laguerre models. Once again, these works use first principle models to obtain Volterra-Laguerre coefficients.

The advantage of using the orthonormal Laguerre series comes from the reduction of parameters necessary to represent the system. An additional advantage is the theoretical ease with which the orthogonal coefficients can be determined. This work is limited to Laguerre series which are complete in \( L_2 [0,\infty) \) space. Laguerre series are used here because of their similarity to
Padé approximations of deadtime. This characteristic is important since deadtime is usually present and significant in chemical processes. It is also assumed that the process being modeled is L$_2$ stable, i.e.:

$$\int_0^\infty y^2(t)dt < \infty$$  \hfill (3.1)$$

This is the case for fading memory, self regulating systems. If this is the case, the Riesz-Fisher theorem (Dumont et al., 1991) states that the process can be modeled as:

$$y = \sum_{i=1}^{\infty} \theta_i \cdot l_i(k)$$ \hfill (3.2)$$

and for any positive $\varepsilon$ there is an $n$ such that:

$$\int_0^\infty \left[ y(t) - \sum_{i=1}^{n} \theta_i \cdot l_i(k) \right]^2 dt < \varepsilon$$ \hfill (3.3)$$

where, $\theta$ are the Laguerre coefficients and $\varepsilon$ represents the acceptable integrated error. Dumont et al. (1991) and Zheng et al. (1995) have proposed two ways to determine the coefficients $\theta$. The first is called the correlation method and is a natural consequence of orthogonality. In this case, the $i$'th Laguerre gain can be calculated as:

$$\hat{\theta}_i = \frac{1}{n} \sum_{k=1}^{n} y(kT)l_i(kT)$$ \hfill (3.4)$$

This equation assumes that the input is Gaussian random noise. Also assumed is that all disturbances are measured disturbances and there is no other correlated or uncorrelated noise.
The second approach presented by Dumont (1991) is the least-squares approach. In this approach, $y$ is represented as:

$$
Y = \Phi^T \theta + \Phi_u^T \theta_u + W
$$

(3.5)

where the first term on the right-hand side represents the model, the second term represents a similar Laguerre expansion for the unmodeled dynamics, and the third represents the effect of random noise on the process. In the above equation, the parameters are defined as:

$$
\phi^T = [l_1(kt), \ldots, l_n(kT)]
$$

(3.6)

$$
\phi_u^T = [l_1(kT), \ldots, l_n(kT)]
$$

(3.7)

$$
\Phi^T = [\phi(T), \ldots, \phi(nT)]
$$

(3.8)

$$
\Phi_u^T = [\phi_u(T), \ldots, \phi_u(nT)]
$$

(3.9)

$$
Y^T = [y(T), \ldots, y(nT)]
$$

(3.10)

The least squares estimate of $\theta$ becomes:

$$
\hat{\theta} = \left[\Phi \Phi^T\right]^{-1} \Phi Y
$$

(3.11)

Dumont continues on to demonstrate that if the expectation is taken, the second and third terms of equation (3.5) go to zero if the input and disturbances have zero mean. In the previous sections, the process $Y$ was approximated using Laguerre polynomials.

In fading memory systems, the finite Laguerre series can be used to approximate the Volterra kernels $g_i$ as shown in equations (1.31) through (1.34), the advantage again being that
the \( i \)th order Volterra kernel \( g_i \) has on the order of \( m^i \) coefficients where \( m \) is the time horizon and \( i \) is the order of the Volterra kernel. The Laguerre approximation has approximately \( r^i \) coefficients where \( r \) is the order of the Laguerre approximation. In practical cases, \( r \) will be much smaller than \( m \). Zheng (1995) reported a reduction from 23,750 coefficients using the Volterra model to 20 coefficients using the Volterra-Laguerre model in one test problem.

Utilizing this substitution, the Volterra-Laguerre expansion becomes:

\[
\tilde{y}(k) = \sum_{n=1}^{N} \sum_{j_1=1}^{r} \cdots \sum_{j_n=1}^{r} \theta_n(j_1, \cdots, j_n) \prod_{l=0}^{m} l_{i_l}(i_l) u(k-i_l)
\]

(3.12)

where \( n \) is the order of the Volterra series and \( r \) and \( m \) are as previously defined.

The same technique could be used to approximate the Volterra kernels. Zheng et al. (1995) presents the correlation method approach to estimate the Volterra kernel. It has been assumed that the input signal for this derivation is Gaussian random noise. Other authors, such as Pearson et al. (1993) suggest that a pure white noise input signal may not be required.

If the system is causal with fading memory, the first two terms in Volterra approximation can be represented as:

\[
y(t) = y_1(t) + y_2(t)
\]

(3.13)

where

\[
y_1(t) = \sum_{i=0}^{p} g_1(i)u(t-i)
\]

(3.14)

\[
y_2(t) = \sum_{i=0}^{p} \sum_{j=0}^{p} g_2(i, j)u(t-i)u(t-j)
\]

(3.15)
In this form, \( y_i(t) \) is the \( i \)th order Volterra kernel. Thus, \( y_1(t) \) is the first order Volterra kernel and is equivalent to the impulse response form of the linear multivariable controller. \( g_i \) is the \( i \)th order operator which contains \( p \) coefficients. The Laguerre approximation for the second order Volterra kernel is:

\[
g_2(\tau_1, \tau_2) = \sum_{j_1=0}^{r} \sum_{j_2=0}^{r} \theta_{j_1,j_2} l_{j_1}(\tau_1) l_{j_2}(\tau_2)
\]  

(3.16)
defining

\[
D = \begin{bmatrix}
\theta_{11} & \theta_{12} & \cdots & \theta_{1r} \\
\theta_{21} & \theta_{22} & \cdots & \theta_{2r} \\
\vdots & \vdots & \ddots & \vdots \\
\theta_{r1} & \theta_{r2} & \cdots & \theta_{rr}
\end{bmatrix}
\]  

(3.17)

\[
L^T(\tau) = (l_0(\tau) \quad l_1(\tau) \quad \cdots \quad l_r(\tau))
\]  

(3.18)

then

\[
g_2(\tau_1, \tau_2) = L^T(\tau_1)DL(\tau_2)
\]  

(3.19)

Equation (3.19) can be substituted into equation (3.15) to produce:

\[
y_2(t) = \sum_{i_1=0}^{p} \sum_{i_2=0}^{p} u(t-i_1) u(t-i_2) L^T(t-i_1) DL(t-i_2)
\]  

(3.20)
\[ U^T = (u(t) \ u(t-T) \ \cdots \ u(t-pT)) \]  

(3.21)

and recall that

\[
\Phi = \begin{bmatrix}
l_0(0) & l_1(0) & l_2(0) & \cdots & l_r(0) \\
l_0(T) & l_1(T) & l_2(T) & \cdots & l_r(T) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
l_0(pT) & l_1(pT) & l_2(pT) & \cdots & l_r(pT)
\end{bmatrix}
\]

(3.22)

then

\[ y_2(t) = U^T \Phi D \Phi^T U \]  

(3.23)

Equation (3.23) is the second order prediction for one point in time. The overall prediction at this time period is the sum of the linear and second order contributions as shown in equation (3.13). In this context, equation (3.23) gives the nonlinear correction for the linear model.

**Testing Procedure Issues**

In doing identification testing to obtain I/O data for nonlinear systems, there are several issues to be considered. The first concern is the input excitation required to adequately generate control variable response over the expected operating conditions and ranges. Input step size, step frequency, number of steps, duration, and technique all need to be considered. The number of steps required will be discussed later in this dissertation.
Input Sequence Form

In theory, the ideal step sequence has been shown to be completely random both in frequency and magnitude (Zheng et al., 1995). Zheng showed that if the input signal is Gaussian random noise, for a long enough time, it is expected that all of the relevant frequencies of the process are excited and the regression models are optimized. Unfortunately, this is not practical in industrial applications. Plant personnel and management will not allow for random inputs, neither in size or duration, on live process variables. It could produce dangerous conditions, equipment damage, off-spec products, or destabilize whole process units. Other authors, such as Pearson et al. (1993) have suggested that a pure white noise input signal may not be required. Pearson suggests that elliptically distributed inputs may also give good results based on autocorrelation matrix analysis. These results give credence to existing step test techniques used in industry. The most commonly used input technique in industry for linear systems identification is pseudo-random binary sequences (PRBS). PRBS does a good job of exciting the process for linear systems but does not adequately excite higher order Volterra terms (Nowak et al., 1994).

In linear identification, the step size is constant and the magnitude is chosen based on the minimum step size needed to excite the control variables in the process. This technique identifies only a linear process model around the testing point. It is assumed that the linear model represents the behavior of the process over the whole expected operating range of control.

Nowak recommends, based on analysis of the correlation matrix, that for nonlinear models, pseudo-random multilevel sequences (PRMS) be used. For the work in this dissertation, an extension of the PRMS approach is used. Instead of using a fixed number of levels for input amplitude, input amplitude is varied in a pseudo-random fashion as well. A typical input sequence is shown in Figure 5. The amplitude of the steps are restricted between a minimum step size which is required to see movement in the control variables and a maximum step size that is considered safe. The steps are randomly varied around an average step size.
follow a uniform distribution. The frequency is also randomly varied around an average step
time. The average step time is set based on the dominant time constant of the process response
and also has uniform distribution. This is necessary to insure adequate data in both the low and
high frequency ranges. In industry, it is commonly considered unacceptable to randomly move
process variables. This approach is more appealing to industry because the random nature of the
moves is limited based on prior knowledge of the process. This also has the advantage of a
pseudo-random amplitude component which is important for the nonlinear Volterra terms. This
gives a uniform distribution of amplitudes in the testing range. This is new and an extension of
the traditional PRMS which uses a fixed number of amplitudes in the step testing.
Identification of SISO Second Order System

In the truncated Volterra approximation shown in equation (1.33), the identification of the second order Volterra-Laguerre approximation from I/O data consists of two steps. The first step requires performing a linear identification to get \( y_1(t) \). The residual of \( y(t) - y_1(t) \) can then be used as the process data in the second order identification. Given equation (3.23), a solution must be found for the coefficient matrix \( D \). Note that \( D \) has dimensions \( r \times r \) and is symmetric; therefore, there are \( \frac{1}{2}(r^2 + r) \) independent coefficients in the system. The identification problem can be posed as:

\[
\min \left\| \sum_{i=1}^{NS} y_2(t) - (\Phi^T U)^T D (\Phi^T U) \right\|_2
\]

(3.24)

where \( NS \) is the number of I/O data points. Assuming there is an exact solution for \( y_2(t) \), the equation becomes:

\[
y_2(t) = (\Phi^T U)^T D (\Phi^T U)
\]

(3.25)

Multiplying both sides by \((\Phi^T U)\) and solving for \( D \) yields:

\[
D = \left[ (\Phi^T U)(\Phi^T U)^T \right]^{-1} (\Phi^T U) y_2(t)(\Phi^T U)^T \left[ (\Phi^T U)(\Phi^T U)^T \right]^{-1}
\]

(3.26)

where the term \( \left[ (\Phi^T U)(\Phi^T U)^T \right]^{-1} \) is the inverse. This is the solution for only one data point. For the complete time horizon of the second order system, it becomes necessary to perform a nonlinear optimization. In doing so, advantage can be taken of \( D \) being symmetric. The total
number of independent coefficients become $\frac{1}{2}(r^2 + r)$. Thus, equation (3.26) is used for the optimization with the objective function being equation (3.24).

The first test of these equations is to identify simulated I/O data generated from a simulation using a second order Volterra model. The Volterra process was created using 15 linear coefficients and 120 second order Volterra coefficients. The process was essentially a first order plus deadtime linear response with second order dynamics added. The results of the identification are shown in Figure 6 thru Figure 8. The optimization was calculated using the nonlinear least squares algorithm in Matlab’s (Mathworks, 1999) optimization toolbox. The errors are the least squares error between the process simulation and the approximation from the linear and second order Volterra-Laguerre models. The process was identified several times with increasing Laguerre orders until improvement diminished. In this case, a Laguerre order of 3 was completely adequate for modeling the process. This provided a reduction of coefficients from 135 Volterra coefficients to 9 Laguerre coefficients in the Volterra-Laguerre model. This illustrates the effectiveness of the Volterra-Laguerre model in representing a well-behaved second order Volterra system. The implications of this are that the work done using second order Volterra models to represent industrial processes can be used to set expectations of the Volterra-Laguerre models. With that said, this is the most optimistic case for the Volterra-Laguerre modeling. The data used for identification had no noise and the system was a well-behaved and well-defined Volterra model.

To evaluate the robustness of the identification with noise present, noise was added to the second order Volterra process step data and the process model identification was performed again. The same process data as the previous test was used but normalized random noise, with a standard deviation equal to the average process (CV) movement, was added to the simulated I/O data before the identification. Figure 9 shows the process data with and without the noise.
Figure 6  Volterra 2\textsuperscript{nd} order process - $R = 2$ (2 linear coefficients and 3 2\textsuperscript{nd} order coefficients.)

Figure 7  Volterra 2\textsuperscript{nd} order process - $R = 3$ case (3 linear coefficients 6 2\textsuperscript{nd} order coefficients)
Figure 8 Volterra 2\textsuperscript{nd} order process - R=4 case (4 linear coefficients 10 2\textsuperscript{nd} order coefficients)

Figure 9 2\textsuperscript{nd} order Volterra process step test data with noise
In the evaluation of model performance, the squared error between the Identified Volterra-Laguerre model and the process data without noise was used. The purpose of this is to illustrate the stability and noise rejection characteristics of the Volterra-Laguerre models. For this test, 75 steps were used in the process step test. The effect of the number of steps in the test will be discussed later. Figure 10 through Figure 12 show the predictions from the identified Volterra-Laguerre models and the process data without noise. As with the identification without noise, a Laguerre order of 3 produces a good fit with an error reduction over the linear model error of 94%.

![Figure 10 Volterra 2nd order process ID with noise - R=2 case](image-url)
Figure 11 Volterra 2nd order process ID with noise - R=3 case

Figure 12 Volterra 2nd order process ID with noise - R=4 case
CHAPTER IV

CASE STUDY I: ISOTHERMAL FREE RADICAL POLYMERIZATION CSTR

The following example consists of an isothermal free radical polymerization of methyl methacrylate using azo-bis-isobutyronitrile as initiator and toluene as solvent (Congalidis et al., 1989; Doyle III et al., 1994; Maner et al., 1996). The manipulated variable is initiator flow rate and the control variable is number average molecular weight (NAMW). The first principles differential equations for the process model (Daoutidis et al., 1990; Maner et al., 1996) were obtained based on the assumptions of:

- Perfect mixing
- Constant heat capacity
- Quasi steady state
- No gel effect
- Monomer inlet flow is constant

\[
\frac{dC_m}{dt} = -(k_p + k_f)C_mC_m + \frac{F(C_m^o - C_m)}{V} \quad (4.1)
\]

\[
\frac{dC_I}{dt} = -k_fC_I + \frac{F(C_t^o - FC_I)}{V} \quad (4.2)
\]

\[
\frac{dD_o}{dt} = \left(0.5k_r + k_f\right)P_o^2 + k_fC_mC_P - \frac{FD_o}{V} \quad (4.3)
\]
\[
\frac{dD_1}{dt} = M_m (k_p + k_f) C_p m P_o - \frac{FD_1}{V}
\]  

(4.4)

where

\[
P_o = \left[ \frac{2 f^* k_i C_i}{r_d + k_r} \right]^{0.5}
\]  

(4.5)

The process model parameters and steady-state operating conditions are given in Table 1 and Table 2 as given in Maner et al. (1995). In Maner’s work, Volterra models were obtained by performing Carleman linearization (Rough, 1981) of the first principle models. For this work, the same first principle models are used to generate I/O data that will then be used for identification. Figure 13 shows the hardware configuration of the CSTR reactor. Open loop step response curves are shown in Figure 14 for a .5 normalized step up and a step down of equal magnitude. As can be seen, the process is not symmetrical. This is a common nonlinearity observed in industry. This form of nonlinearity is seen in high purity distillation, combustion, process heaters, chemical reactors (reformers, polymers, alkylation, dewaxing, etc.), to name a few.

**Linear Volterra-Laguerre**

One of the results of this research is the insight gained into the value of using Laguerre reductions for identification of linear models. The most difficult step during the implementation of linear MPC on industrial processes is obtaining good models. In processes with many independent variables, the number of step tests required to obtain first order Volterra models can be significant. Thus, step testing often takes long periods of time, during which there are measurable disturbances to plant operations. If the process is noisy, more time and more steps become necessary. As the noise increases, the number of process step tests must also increase.
Table 1 Isothermal free radical polymerization parameters

<table>
<thead>
<tr>
<th>Parameters For Isothermal Polymerization CSTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{T_e}$ Equilibrium Constant $103281 \times 10^{10}$ m$^3$/kmol h</td>
</tr>
<tr>
<td>$k_{T_s}$ Equilibrium Constant $10.930 \times 10^{11}$ m$^3$/kmol h</td>
</tr>
<tr>
<td>$k_i$ Equilibrium Constant $1.0225 \times 10^{-1}$ m$^3$/kmol h</td>
</tr>
<tr>
<td>$k_p$ Equilibrium Constant $2.4952 \times 10^{6}$ m$^3$/kmol h</td>
</tr>
<tr>
<td>$k_{f_w}$ Equilibrium Constant $2.4522 \times 10^{3}$ m$^3$/kmol h</td>
</tr>
<tr>
<td>$f^*$ Constant 0.58</td>
</tr>
<tr>
<td>$F$ Monomer Inlet Flow 1.00 m$^3$/h</td>
</tr>
<tr>
<td>$V$ Volume of Tank 0.1 m$^3$</td>
</tr>
<tr>
<td>$C_{I_n}$ Initiator Inlet Concentration 8.0 kmol/m$^3$</td>
</tr>
<tr>
<td>$M_m$ Monomer MW 100.12 kg/kmol</td>
</tr>
<tr>
<td>$C_{m_n}$ Monomer Inlet Concentration 6.0 kmol/m$^3$</td>
</tr>
</tbody>
</table>

Table 2 Polymerization CSTR steady state operating conditions

<table>
<thead>
<tr>
<th>Steady-State Operating Conditions For Isothermal Polymerization CSTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = C_m$ = 5.506774 kmol/m$^3$</td>
</tr>
<tr>
<td>$x_2 = C_i$ = 0.132906 kmol/m$^3$</td>
</tr>
<tr>
<td>$x_3 = D_n$ = 0.0019752 kmol/m$^3$</td>
</tr>
<tr>
<td>$x_4 = D_i$ = 49.38182 kg/m$^3$</td>
</tr>
<tr>
<td>$u = F_l$ = 0.016783 m$^3$/h</td>
</tr>
<tr>
<td>$y$ = 25,000.5 kg/kmol</td>
</tr>
</tbody>
</table>
Figure 13 Hardware configuration for polymerization CSTR

Figure 14 Step response curves CSTR
One of the advantages of using Volterra-Laguerre reductions is the reduced number of step tests required for identifying linear models. Since Laguerre equations are used to approximate the Volterra kernel in the Volterra-Laguerre model, it would be expected that Volterra models would always produce better results than the Volterra-Laguerre models in the identification. Later in this section, this is shown to be true for pure Volterra systems without noise. Figure 15 shows a comparison of the linear Volterra and the linear Volterra-Laguerre identification improvements for the CSTR process as a function of the number of steps in the tests and the Laguerre order. Figure 15 is based on the average of several step tests for both the Volterra and Volterra-Laguerre equations. The frame of reference for these plots is the linear Volterra model. The first order Volterra identification is the current standard used throughout industry. A value of 0% improvement means that the linear Volterra-Laguerre performed as well as the first order Volterra model for the system. A value of 100% indicates that the model matched the process data perfectly and 100% of the modeling error was eliminated. Thus, the
Figure 15 Improvement of linear Volterra-Laguerre units of identification performance are indicated as percent error reduction between the linear Volterra model and the process data. Figure 16 illustrates that as the number of steps in the step test increases, the advantage of using linear Volterra-Laguerre identification over the standard first order Volterra identification diminishes. Figure 16 also provides an indication of the number of steps required for the linear Volterra identification. For this case, it appears that 30 to 40 steps are required. This would be more steps than are usually performed on industrial applications for linear control models. The characteristics of the process itself are one of the factors that affect the number of steps required in the process identification test. The fact that this case represents a nonlinear process may increase the number of steps required. Nevertheless, with less than 40 steps and a Volterra-Laguerre order of 4 or higher, the Volterra-Laguerre still performs better than the straight Volterra identification.

The second comparison for linear system identification concerns what happens when the process has noise. To illustrate this, normally distributed random noise was imposed on the
process data and the system was identified again using the noisy signal. Figure 17 illustrates a typical 20 step response with the actual process signal and the same process signal with added noise. To test this system, the standard deviation of the noise was set as 1 times the standard deviation of the process movement for the data set. Because the real process was known, both the Volterra and Volterra-Laguerre identified models were compared to the actual process without noise. Figure 18 illustrates the relative improvement of the Volterra-Laguerre prediction as a function of Laguerre order and the number of steps used in the identification. Figure 19 illustrates predictions for a typical 20 step system identified with noise. It is clear from the results shown in Figure 18 that the Volterra-Laguerre model produces an improved fit over the Volterra model. The use of the Laguerre equations to predict the Volterra kernel acts as a process filter.
Figure 17 Process data for 20 step identification data set with noise

Figure 18 Volterra-Laguerre linear improvement for process with noise
Figure 19 Volterra and Volterra-Laguerre linear predictions on process with noise

As expected, as the number of steps increases, the Volterra-Laguerre model approaches the performance of Volterra. As stated previously, if the process (without noise) was not “smooth,” the Volterra-Laguerre approximation would not work as well. It is expected that the same results could be obtained from a straight Volterra identification if a filter were used on the data before identification. The two significant advantages illustrated for the Volterra-Laguerre identification approach are the following:

- The number of coefficients required to identify the system using Laguerre reductions is always fewer than what is required for the straight Volterra form.
- The Volterra-Laguerre equations naturally smooth the data.

Therefore, model quality is improved while the number of step tests required is reduced. As a result, for processes with long times to steady state, or many independent variables, the Volterra-Laguerre identification would be superior in industrial application.
As discussed earlier, the Laguerre time scale factor should also be included in the parameter identification. The exponential time scale factor makes the identification well-behaved and nonlinear. Figure 20 illustrates the effect of the Laguerre time scale factor on the linear Volterra-Laguerre identification. This plot shows the importance of the time scale factor in the Identification. The Volterra-Laguerre identification can take either less or more CPU time than the Volterra alone because of the nonlinearity. The fact that there are fewer parameters being identified reduces the CPU requirements but the nonlinear time factor increases identification time. In this CSTR process identification, the CPU time requirements ranged from half the CPU time to 50% more CPU time for the linear Volterra-Laguerre identification than the straight first order Volterra identification. Because the development was done in Matlab (Mathworks Inc.), this may not be representative of compiled code. In any case, the identification is a one-time process and the CPU time would not prove to be a problem for industrial applications.

![Linear Volterra-Laguerre dependence on time scale factor](image)

Figure 20  Linear Volterra-Laguerre dependence on time scale factor
Second Order Volterra-Laguerre

The main thrust of this research is to identify second order Volterra-Laguerre coefficients from I/O data and quantify the potential improvement of these models over the linear models being used in industry today. A comparison of Volterra-Laguerre dynamic models obtained from I/O data to the first principle models to illustrate overall quality has also been performed. Both linear and nonlinear Volterra-Laguerre identifications are compared. Identifications were completed for several (10-15) step data sets to evaluate robustness and sensitivity of the identification procedure. Some of the topics to be discussed include the following:

- the number of steps required for the linear and nonlinear identification.
- the minimum order of the Laguerre approximation to obtain reasonable results.
- the effect and selection of the Laguerre Exponential time scale factor.
- the effect of process noise on control models.

A key indication of the practicality of the identification procedure is the minimum number of independent variable steps required to obtain acceptable nonlinear models. The relationship between the number of steps and relative error reduction for this system is shown in Figure 21 and Figure 22. These figures show the average error reduction over linear Volterra models based on the identification of ten different input-output data sets for each corresponding number of steps. Once again, a value of 0% represents no improvement over linear Volterra models, a value of 100% represents a second order Volterra-Laguerre model that matches the process perfectly, and there is no error. There are six lines plotted corresponding to the Laguerre order used in the approximation. As can be seen in these figures, the improvement over linear models ranges from 65% to about 75% at 30 steps. Figure 23 illustrates that the standard deviation of the error reduction varies with the number of steps. There is a breaking point at 30-40 steps. This is approximately the same number of steps required to obtain the Volterra linear models as discussed in the previous section. Thus, if a single set of step tests with 40 steps were
performed, error reductions of $75\% \pm 6\%$ could be expected. This is significant in that 30-40 steps would be an acceptable number of steps for nonlinear models on industrial processes that provide noteworthy benefit over linear models. The Volterra-Laguerre identification also requires a nonlinear optimization.

The question of convergability is present in this approach as well. Though there may be a solution, the problem has multiple local optima and may also be stiff or otherwise ill conditioned. There is no guarantee of convergence or repeated convergence to the same solution in the process of identification. Figure 24 and Figure 25 illustrate that even with these problems, significant improvement is repeatable and can be consistently obtained. All of the identification cases completed for this project demonstrated identification convergability. Thus, the conclusion drawn is that the global optima for the identification may or may not be the one achieved, but the improvement for whatever optima is obtained is significant over linear models.

![Figure 21 Volterra-Laguerre 2\textsuperscript{nd} order error reduction](image-url)
Figure 22 Volterra-Laguerre 2\textsuperscript{nd} order error reduction
Figure 23  Standard deviations of 2\textsuperscript{nd} order identification

Figure 24  2\textsuperscript{nd} order improvement for process with noise
In the work done for this dissertation, the mean time between steps used was one half time to steady state (TSS). Thus, the total testing time based on this manipulated variable and this control variable and a 30 step test would be 15*TSS. Industry currently uses 10 to 20 steps to identify linear models depending on the process, process noise, linearity, and other factors. There is always a tradeoff between the number of steps used in the identification and the quality of the models. Figure 26 illustrates model predictions for a typical 20 step prediction set where the input-output data has not been used in the identification. This prediction is based on models identified from a different 20 step test. In this example, the first order Volterra-Laguerre model and impulse coefficient model (first order Volterra) perform comparably while the second order Volterra-Laguerre model does a much better job of fitting the actual process data.

The number of steps required in the model is always an issue. The order of the desired model also affects the number of steps required. In step tests with fewer steps a low order
Figure 26 2\textsuperscript{nd} order predictions on 20 step data

Volterra-Laguerre model performs as well as a higher order model. This is because with the limited data set, only identification of a smaller coefficient set is supported. In addition, as the number of Laguerre coefficients in the model increase, the model begins capturing noise effects. The extreme of this effect is the straight Volterra model. When there are many steps in the step test and as the Volterra-Laguerre order increases, the quality of the model improves. Figure 27 illustrates the objective function for a second order identification as a function of the Laguerre time scale factor and Laguerre order. Figure 27 also indicates that as the Laguerre order is increased, the relative magnitude of the improvement decreases. In this case, a Laguerre approximation with 4 coefficients would appear to be adequate. These results support an approach of identifying input-output data with continually increasing orders until the improvement caused by increasing the order becomes lower than some threshold.
As in the linear case, the selection of the Laguerre time scale factor is important. A plot depicting the effect of the time scale factor on the model error is shown in Figure 27. Since the identification requires a nonlinear optimization, the question of convergability is present here as well. Even though there may be a solution, the problem has multiple local optima and may again be stiff or otherwise ill conditioned. There is no guarantee of convergence or repeated convergence to the same solution in the process of identification. The work done in this dissertation is not meant to prove unconditional convergence criteria, but to illustrate practical convergability, convergence rates, and repeatability. Statistical information on the identification convergability is given in Figure 25. Figure 27 illustrates the shape of the objective function for the nonlinear fit of the time scale factor. A gradient-based optimization (nonlinear least squares) was performed to determine parameters for the model. It is apparent, by looking at the shape of
the error function, that the solution of the optimization may not be the global optimum for each
Laguerre order. However, the differences between the local optimum and the global optimum are
not significant.
CHAPTER V

CASE STUDY II: VAN DER VUSSE REACTION

The second case is the Van Der Vusse reaction:

\[ A \xrightarrow{k_1} B \xrightarrow{k_2} C \]

\[ 2A \xrightarrow{k_3} D \]

The reaction is carried out in an isothermal CSTR as shown in Figure 28. This system has been studied by several researchers including Doyle III (1995), Maner et al. (1996), and Van Der Vusse (1964) and is considered a benchmark problem for nonlinear control. The system exhibits both nonminimum phase behavior, and nonlinear gain and dynamic effects. The problem addressed is to control the concentration of B by manipulating the inlet flow rate. The mass balance for the reacting components is given by:

\[
\frac{dC_A}{dt} = -k_1 C_A - k_3 C_A^2 + \frac{F(C_{Af} - C_A)}{V} \tag{5.1}
\]

\[
\frac{dC_B}{dt} = k_1 C_A - k_2 C_B + \frac{F}{V} C_B \tag{5.2}
\]

\[ y = C_B \tag{5.3} \]

The process model parameters and steady-state operating conditions are given in Table 3 and Table 4 as given in Doyle et al. (1995). In Maner’s work, Volterra models were obtained by
Figure 28 Hardware configuration of Van Der Vusse isothermal CSTR reactor

Table 3 Van Der Vusse reaction parameters

<table>
<thead>
<tr>
<th>Parameters For Isothermal Van Der Vusse Reaction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_1 )</td>
<td>50 hr(^{-1})</td>
</tr>
<tr>
<td>( k_2 )</td>
<td>100 hr(^{-1})</td>
</tr>
<tr>
<td>( k_3 )</td>
<td>10 L/(mol hr)</td>
</tr>
<tr>
<td>( F )</td>
<td>1.12 L/hr</td>
</tr>
<tr>
<td>( V )</td>
<td>1 L</td>
</tr>
</tbody>
</table>

Table 4 Van Der Vusse CSTR steady state operating conditions

<table>
<thead>
<tr>
<th>Steady-State Operating Conditions Van Der Vusse Reaction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{Ao} )</td>
<td>3.0149 mol/L</td>
</tr>
<tr>
<td>( C_{Bo} )</td>
<td>1.12 mol/L</td>
</tr>
<tr>
<td>( F/V )</td>
<td>34.5939 L/hr</td>
</tr>
<tr>
<td>( y )</td>
<td>1.12 mol/L</td>
</tr>
</tbody>
</table>
using Carleman linearizations (Rough, 1981) of the first principle models. For this work, the same first principle models are used to generate I/O data that will then be used for identification. Open loop step response curves are shown in Figure 29 for a .5 normalized step up and step down of equal magnitude. As can be seen, the process is not symmetrical.

The process has both changing gains and dynamic characteristics. Thus, the dominant root for the step up and the step down appear to be different. This is a common nonlinearity found in industry. In addition, this process exhibits nonminimum phase behavior with the inverse response. Inverse response is usually difficult to model and controller performance is affected dramatically if inverse response is not modeled properly.

One of the advantages of using Volterra form nonlinear models is that changing gain, inverse response, and dynamic nonlinearities can be captured. The steady state gain loci for the Van Der Vusse reaction are shown in Figure 30. One of the characteristics that makes this system interesting is that the process has a maximum value with obtainable lower values on each side. The process gain is the tangent of this curve at any given operating point and thus changes sign. The implications of the gain sign change are that if the process is operating near the maximum, a linear controller will show a zero gain between the feed flow rate and the concentration of B. If the process is identified on the left side of the maxima a positive gain will be identified. Later, if the process is pushed to operate on the other side, the process gain will change signs and the controls will become unstable. This result is extremely important in industry. This behavior is common in many reacting systems, including combustion systems. In many of these systems, controlling close to the zero gain point is the optimum process operating point. It represents optimal stoichiometry for reactants in a reacting system. The ability to capture gains that not only change values but also change signs provides the ability to control many systems over a range that has proven to be uncontrollable with standard linear model predictive control. This result helps fill a hole in existing control technology for reacting systems. Currently, operators are
Figure 29  Step response curves for .5 normalized feed steps
Figure 30  Process loci for Van Der Vusse reactor

forced to operate significantly away from the stoichiometric optimum to maintain stable control. The danger is that if they exceed the stoichiometric ratio with current controls, the process goes immediately unstable. This is a strong justification for using the higher order Volterra-Laguerre approach for reacting systems.

**Linear Volterra-Laguerre**

As in the previous case, the Volterra-Laguerre provided similar advantages for linear identification of the Van Der Vusse system. The main advantages were found in the reduced number of steps required to identify the linear models and the filtering characteristics of the Laguerre reductions. The Impulse and the Volterra- Laguerre systems were both able to capture the inverse response behavior of the system.

Figure 31 illustrates a comparison of a typical step response for the process linear impulse coefficient model and Volterra-Laguerre model. As can be seen, neither of the linear models captures the nonlinear characteristics of the process. In Figure 31, the process was
Figure 31 Model comparison for steps on Van Der Vusse system
modeled with fifty coefficients in the impulse model and five coefficients in the Volterra-
Laguerre model. Notice that the steady state gain of the second order Volterra-Laguerre model
overshoots the process data in both directions. This is a result of the multiple step sizes used in
the identification. The identification does the best fit possible for an array of step sizes. Thus,
unlike the linear binary stepped identification, there is no guarantee that for a given step size there
will be symmetrical errors around the step. Figure 32 illustrates the relative improvement for a
noise free identification. As expected, the first order Volterra-Laguerre approaches zero
improvement over the impulse coefficient identification as the number of steps increases. Figure
33 depicts the standard deviation of the linear identification based on the number of steps used in
the step test. Figure 32 and Figure 33 show it is clear that two Laguerre coefficients are not
adequate to describe the system. Even without process noise, it appears that between 10 and 20
steps are required to obtain an adequate linear model. When noise is added to the identification
data, the situation changes. Figure 34 illustrates a typical 20 step identification data set with
noise. Again, the noise is normally distributed random noise with a standard deviation equal to
the average movement of the system without noise. Figure 35 illustrates the improvement of the
Volterra-Laguerre over the impulse coefficient model. Because of the noisy signal, the Volterra-
Laguerre identification demonstrates significant improvement at fewer steps and approaches that
of the impulse model as the number of steps in the identification increases. Once again, this
demonstrates the advantage of using the Volterra-Laguerre form even in linear identification.
Figure 36 depicts the variation of identifications based on the number of steps in the step test. In
this case, a 30 step test appears to be adequate. Again, it is important to note that there is a
relationship between the number of required steps and the magnitude of the noise. For the Van
Der Vusse process with no noise, 10 to 20 steps seemed to be adequate. In this case, three
Laguerre coefficients appear to be adequate. The proposed identification approach would require
successive identifications increasing the number of Laguerre coefficients until the relative
Figure 32 Improvement of linear Volterra-Laguerre

Figure 33 Standard deviation of linear Volterra-Laguerre IDs
Figure 34 Typical 20 step Van Der Vusse ID set with noise

Figure 35 Improvement of linear Volterra-Laguerre on process with noise
Figure 36 Standard deviation of linear Volterra-Laguerre IDs with noise improvement diminished. In this process, there is significant improvement increasing from two to three Laguerre coefficients, but almost none in moving from three to four Laguerre coefficients. Using the prescribed approach, the identification would stop at four Laguerre coefficients. Figure 37 illustrates a typical plot of a linear prediction based on models obtained from a 30 step test using four Laguerre coefficients and fifty impulse coefficients. The models are compared to the process data without noise. Again, the impulse coefficient model clearly degenerates to a greater extent by the process noise than the Volterra-Laguerre model. In this case, no filtering was performed on the data. It is expected that the linear impulse model could be improved significantly by filtering the data. This step is eliminated in the Volterra-Laguerre identification.

As in the previous case, the overall optimization of parameters is highly dependent on the Laguerre time scale factor. Figure 38 illustrates the effect of time scale selection on the objective function. The time scale factor was again included as one of the parameters in the identification.
Figure 37 Linear identification comparison on Van Der Vusse data with noise

Figure 38 Time scale factor "a" effect on linear optimization
The time scale factor is very important if you want to use a low order Laguerre approximation for your model.

**Second Order Volterra-Laguerre**

Once again, the main thrust of this research is to demonstrate the feasibility and practicality of dynamic nonlinear identification based on process step testing and I/O data sets. A significant area of contribution to the science is in dealing with the practical limitations and requirements of this identification. The Van Der Vusse system has been used extensively to illustrate nonlinear process characteristics. The plot in Figure 30 illustrated the process gain loci for the Van Der Vusse system. One interesting and important aspect of this system is that the process gain actually changes sign. At a reactant flow rate of 60, the process gain is approximately zero. This characteristic would make control of this process with a standard linear controller unstable across the range where the gain changes sign regardless of how mildly the controller was tuned. In many processes, the optimal place to operate the process is at this maximum value. This makes control locally uncontrollable. This process characteristic is common in many industrial applications where reactions are occurring, the most prevalent being combustion systems. If the manipulated variable were fuel rate and the control variable were process temperature, the plot of increasing fuel would show the same temperature characteristics. Many polymeric reactions also exhibit the same behavior.

A second form of nonlinearity also exists in many processes. Not only does the gain change, but the dynamic response changes as well. The Volterra-Laguerre model does a nonlinear mapping of all of the dynamic coefficients and thus captures nonlinearities in the dynamics of the process. This nonlinear dynamic mapping provides advantage over a simpler gain scheduling model predictive control where model gains are changed but no dynamic changes are included. Figure 31 illustrates both the dynamic and the gain nonlinear effects for a simple step up and step down of the process and the models.
This process was identified using the same nonlinear identification technique as in the previous case. The approach uses the linear identification from the previous section and a second order correction matrix to yield second order nonlinearities. Figure 39 depicts the average improvement for ten identifications at each number of steps for the Van Der Vusse system without process noise. In this case, 20 to 30 steps appear to be adequate. Depending on the number of steps, a Laguerre order of three or four appears to be sufficient. These results demonstrate that 80% to 90% of the linear model error can be eliminated using the Volterra-Laguerre second order model.

An indication of the variability of the identification is illustrated in Figure 40. Thirty to forty steps produces a fairly stable response. This is more steps than required in the linear identification but still reasonable. Even when fewer steps are done in the testing, the magnitude of the improvement outweighs the testing standard deviation in the data sets. It appears that regardless of the number of steps performed, the second order Volterra-Laguerre predicts the process significantly better than a linear model.

Figure 41 illustrates a typical prediction plot where twenty feed rate steps are used in the identification. Figure 42 shows the steady state gain loci for the process, linear model, and second order Volterra-Laguerre. The feed flow rate and the concentration of B are scaled and normalized around the initial starting point for the step test. For this plot of gain loci, the mean scaled step size was 0.5. This plot demonstrates that the Volterra-Laguerre model performs well over the tested range and that the change in sign of the gain is captured. The Volterra-Laguerre model breaks down when moving outside the tested range. This verifies the well-known fact that extrapolation outside the testing range can be dangerous. Therefore, when an increased range of operation is expected, the process should be tested over the entire range.

This differs from linear identification. In linear identification, when the process is nonlinear, it is necessary to test at the expected operating point. For processes with significant nonlinearity, process gain is very dependent on the operating point. In addition, the size of the
Figure 39 Volterra-Laguerre 2nd order error reduction

Figure 40 Volterra-Laguerre standard deviation of 2nd order ID
Figure 41 Model prediction of noise free Van Der Vusse system

Figure 42 Comparison of steady state gain loci for process and discrete models
steps in the step tests on these processes can significantly affect the gain of the model. One advantage of the second order Volterra-Laguerre model is an increased range of acceptable predictions over the linear model. A second advantage is the simplicity of stepping over the expected operating range of the process. The size of the steps in the step test should be limited by the constraints of process operation, not based on fitting a linear model to a nonlinear process. This actually makes defining the step tests much easier in the sense that the process nonlinearity does not require minimizing the step size around the current operating point. Making larger steps helps with both the linear and is required for the nonlinear portions of the model. Any errors in the linear model caused by the larger step sizes are picked up in the second order portion of the model.

Figure 43 illustrates the results of the second order Volterra-Laguerre identification for the Van Der Vusse process with noise added. The process data used are the same as the noisy data used in the linear identification. Several aspects of these results are interesting. First, as with the previous cases, there is significant improvement in the second order Volterra-Laguerre model's ability to predict process behavior over the impulse model. Second, as before, a Laguerre order of two is inadequate but still better than the linear model. Third, a continual increase of Laguerre order does not necessarily improve model quality. A Laguerre order of three does better than an order of six or seven. This is true because as the number of coefficients in the model increases, the identification is able to capture unwanted noise effects. Therefore in noisy systems, increasing Laguerre order R can actually produce worse models than lower Laguerre orders.

Again, the proposed procedure will alleviate this problem. In this case, the Laguerre order in the identification would be increased until no significant improvement was shown in the model, thus resulting in a Laguerre order of four. The standard deviation of the identification based on the number of steps in the step test for the noisy data is plotted in Figure 44. The standard deviation suggests that 40 steps would be optimal but, even fewer steps would provide significant improvement. A typical prediction plot of the noisy system is shown in Figure 45.
Figure 43 Volterra-Laguerre 2\textsuperscript{nd} order error reduction - process with noise

Figure 44 Volterra-Laguerre standard deviation of 2\textsuperscript{nd} order ID - process with noise
Figure 45 Typical 30 step model prediction of Van Der Vusse system with noise

The plot displays the linear impulse and Volterra-Laguerre models and nonlinear Volterra-Laguerre prediction, as well as the actual process data. In this plot, the process data are not the same data used in the identification and thus demonstrates the ability of the models to predict data sets not used in the identification. Finally, Figure 46 illustrates the effect of the time scale factor on the identification objective function. Once again, it is clear that the time scale factor is a critical parameter in the identification process. If lower Laguerre orders are expected to be used, using the optimal time scale factor becomes critical.
Figure 46 Effect of time scale factor "a" on 2nd order optimization
CHAPTER VI

CASE STUDY III: IDENTIFICATION OF EXPERIMENTAL PH DATA

The final identification was performed on experimentally obtained PH data from a PH control experimental apparatus. This chapter will illustrate the procedure for identifying the nonlinear Volterra-Laguerre models from input-output data. Thus, the goal of this chapter is to show both the methodology of the identification and the results of an experimentally tested system.

The process consists of one feed tank containing a base solution, dissolved concentrated sodium hydroxide in water, and another feed tank with acidic solution containing hydrochloric acid in water. A hand valve is used to set the acid flow to a general value. Base flow is adjusted as the manipulated variable of the control system. A schematic of the experimental apparatus is shown in Figure 47.

Figure 47 PH control experimental apparatus
The PH control problem exists in a broad range of industries. It has proven to be a nagging problem for control engineers. This system has several characteristics that make it interesting. First, PH control is typically considered off-limits for linear controls because of the extreme process nonlinearity. Second, the logarithmic system, process drift, and other properties make PH control notoriously difficult. Unlike the previous two cases in this work, this data was obtained from the process as input-output data through a distributed control system (DCS). No attempt to do fundamental physics-based modeling was done. Because a fundamental model does not exist, this experimental system illustrates typical problems encountered during industrial process operation. These problems include the following:

The noise in the system is not random noise and is not well defined. There is drift in the base feed solution caused by adsorption of CO₂ from the surrounding air. In addition, flow rates of the base and acid solutions vary with the level in the tanks. Thus, the system is slightly time variant. There is drift in the PH sensor (a combination of three sensors) also causing variation in time. The valve used as the manipulated variable has nonlinear characteristics and exhibits a small amount of hysteresis “stickiness,” depending on step direction and magnitude.

The feed solutions are prepared off-line in batches and have slightly different characteristics. The feed system consists of two tanks and centrifugal pumps, with a control valve connecting the two, and a hand control valve on each line. There was strong coupling between the acid and base flows caused by line pressure drop effects causing additional nonlinearities in the system.

These problems are typical of what would be found in an industrial setting on an active production process. In this case, there was no fundamental dynamic model. The process consisted of multiple pieces of equipment, each with their own set of dynamics, nonlinearities, and peculiarities. The total process was a combination of mechanical, chemical, environmental, and electronic effects.
The objective of this case is not to show that the Volterra-Laguerre model is “ideal” for PH systems but to demonstrate that in systems that are not well defined, the Volterra-Laguerre nonlinear identification can provide significant improvement over linear systems. Second, the Volterra-Laguerre system can be identified from easily obtainable input-output data sets in these systems. In documenting this case, the “identification algorithm” will also be defined and illustrated. The identification algorithm represents a method of how to perform a nonlinear identification using Volterra-Laguerre systems. It includes all of the steps required from step testing to final model validation.

The Volterra-Laguerre characteristics have been illustrated extensively in the previous sections of this dissertation. This chapter will present the procedure and results of what would be a typical industrial identification.

**Procedure**

The first component in any process control identification and modeling project should be to study the process and its characteristics. Some key elements include the following:

- Identify as many peculiarities in the process as possible that may affect the process response.
- Identify any known process disturbances, measured or unmeasured.
- Check for instrumentation problems and characteristics (any problems in instrumentation will result in a degradation of the final models).
- Determine the expected response based on the physics of the process.
- Learn the objectives of the process operation as well as process constraints.
- Determine with the help of the process engineers and operators the goals of the control system.
In the PH control experiment, the process analysis revealed several characteristics that affect process dynamics, including the following:

- The PH control is inherently nonlinear because PH is a logarithmic scale. Thus, linear incremental changes in base flow rate will produce logarithmic, not linear, responses in the PH.
- The process noise in the system is not pure white noise and is not well defined.
- There is drift in the base feed solution caused by adsorption of CO₂ from the surrounding air.
- The feed solutions are prepared off-line in batches and have slightly different characteristics.
- Flow rates of the base and acid solutions vary with the level in the tanks. Thus, the system is slightly time variant.
- There is drift in the PH sensor (a combination of three sensors) also causing variation in time. The valve used as the manipulated variable has nonlinear characteristics and exhibits a small amount of hysteresis “stickiness,” depending on step direction and magnitude.
- The feed system uses two tanks and centrifugal pumps with a control valve connecting the two, and a hand control valve on each line. There is strong coupling between the acid and base flows caused by line pressure drop effects that cause additional nonlinearities in the system.

The second component in the identification process is to evaluate lower level control loops in the process. Aspects of this include the following:

- Ensure control loops are configured properly and are consistent with the overall objectives of the control strategy. Every control loop should have a function. The two primary functions for regulatory controls are to Linearize a portion of the process and to
reject process disturbances that show up in measured variables quickly. An example of this is steam flow control. The flow controller will remove any nonlinearity due to valve trim type and reject high frequency disturbances caused by fluctuating steam pressure.

- Regulatory control loops should be tuned so the process is as responsive as possible without being cyclic or approaching instability. Regulatory loop tuning will have an effect on the dynamics in the resulting models. Tuning should be done before identification of the process.

It is important to recognize that any changes in process equipment, regulatory control loop configuration or tuning, or basic operating parameters (feedstock, catalyst, etc.) can significantly change the process dynamics and process gains. Thus, any changes should be made before step testing of the process. Any changes made after the identification will show up as model mismatch between the identified process model and the actual process response.

In this PH control case, there were no regulatory controls. The dynamics of this process are relatively fast (~30 seconds) so there were no high frequency disturbances to be rejected. Thus, in this case, regulatory configuration and tuning were not an issue.

**Determination of Process Test Conditions**

The third step in the identification process is to define the operating conditions where testing is to be done and to establish the range of step sizes. Since the Volterra-Laguerre model is a regression-based model, it is necessary to define the mean operating point for the test as well. Ideally, this point will be close to what will be the desired process setpoint. When the process is operating at exactly the mean testing point, the linear portion of the Volterra-Laguerre model represents the process without the need for nonlinear corrections. In reality, the actual location of operation after the controls are implemented may change. Improved control usually makes it possible to operate the process near process constraints. In most processes, the optimal operating point is at an operational constraint. This operational shift can cause significant model error in
the linear portion of the model if the process is significantly nonlinear. It is through the nonlinear correction that the process model accounts for nonlinearity. As discussed in the previous chapter, it is important to include as much of the expected operational area as possible in the nonlinear identification test. As a result, the step size magnitude should be as broad as possible. The mean step size and the maximum and minimum limits on the manipulated variable should be selected carefully. Extrapolation of the model predictions outside the tested area often lead to significant model mismatch and thus poor process control.

The PH system testing range was limited by the PH sensor accuracy. The sensor was inaccurate at PH values above 10 and below 4. The mean valve position of 11% was decided upon after preliminary testing. The mean step size of 1.5% was selected because it resulted in a PH data variation of about 5 PH units or 5 orders of magnitude in concentration. This range is broad enough to illustrate significant nonlinearities. The mean time between steps was set at 30 seconds based on the dominant time constant in the system response. This was determined via preliminary step tests as well.

**Process Step Testing**

The next component in the identification process is the actual step testing and data collection. As discussed in Chapter 3, the step frequency and magnitude is based on pseudo-random multilevel sequences (PRMS). There are several items that should be checked before the step testing begins, including the following:

- The process should be at the predetermined mean operating point for the test.
- The regulatory controls should be in the configuration and state they will be in when the controls will be turned on. For example, if a regulatory loop will be in manual mode when the model predictive control is implemented, it should be in manual mode during the step test. There are many cases where the fallback position of the control system
(when the MPC is off) will have additional regulatory control loops activated to help maintain control of the process. These control loops should be off during step testing.

- Unmeasured disturbances should be minimized.

- Data should be collected for all variables that might become a manipulated, disturbance, or control variable in the model. In many cases, there are several variables (or combination of variables) that could become important for the controls after the step testing. Often, the plant operators have never moved the process enough to obtain dynamic cause and effect data. Additional control points for the process often become apparent during the test or during the ensuing analysis.

- The data collection frequency must be high enough to capture the highest frequency dynamics of the process response to be modeled.

- The distributed control system must have a PRMS test sequence testing program implemented that allows for the automatic execution of the manipulated variable moves. This is critical. Manual stepping of the process does not usually work well. It is difficult for even the best operator to implement random steps. The operator has been trained to make multiple moves on a process to deal with process coupling. Dealing with coupling between process variables is one of the main features of the MPC. The MPC must have models that represent interactions between all independent variables and dependent variables in order to address coupling in the process. When steps are correlated based on process response, it becomes impossible to identify the interactions and the resulting models can be seriously compromised.

Once the above items have been completed, the step test can begin. During the step test, the process should be continuously monitored with special attention given to events that will compromise the data being collected. These events include the following:
Manipulated and control variable saturation which cause errors in both process gain and process dynamics of the identified model.

Any unmeasured disturbances that might skew the data.

Any operator intervention on unmonitored or hand controlled variables. Examples of this include hand valves, pump bypass valves, pump motors, etc.

In short, any event that could affect the process data that is not recorded in the data set itself should be recorded in a testing log so that problem sections of the resulting data set can be removed before identification.

The minimum step test duration is based on the number of steps required to obtain an acceptable model, the mean frequency of the steps and the number of independent variables being tested. An optimal test would include enough moves so that data not used in the identification could be used to evaluate the identified models prediction capabilities. In practice, it is often necessary to do preliminary identification of the process during the step testing to determine if enough data have been collected. Guidelines for the number of steps required for the nonlinear identification was discussed in previous chapters.

In the PH case, data were collected at a sample rate of once per second. The dominant time constant for the process appeared to be approximately 30 seconds. The one-second sample rate allowed enough resolution for higher frequency disturbances to be observed. A graph showing raw data from a step test of this system is shown in Figure 48.

**Data Conditioning**

Once the process step testing is complete and the data is collected, the data needs to be preconditioned before the identification is performed. The elements of preconditioning include the following:
Figure 48  PH step test raw data

- Removing data slices for periods where unmeasured disturbances or other process events affecting data quality occurred and detrending the data. These removed periods should be recorded in the testing log as described in the previous section and as was done in the pH case.

- One of the characteristics of MPC is that it assumes the process being controlled is time invariant. Thus, if the manipulated variable is set at the same position at two different times, the control variable will return to the same value. In practice, this is often not exactly true. An example of this time variance can be seen by examining the raw process data in Figure 48.

It is clear that there are sections of the process data where the PH is drifting significantly while the valve position is being varied around the mean testing point. The drift can be caused by any one of many unmeasured effects. Examples of these effects include instrumentation drift and environmental changes. One of the identified causes, in the PH case, is CO₂ adsorption from the air into the base solution. This causes changes in the feed base solution PH, resulting in product
PH drift. Another source of drift noted in the PH process was due to changing levels in the feed tanks. Centrifugal pumps are used to force feed solutions through the system. The feed system is essentially a batch process where feed solutions are prepared and loaded in batches. The levels of the feed tanks and the feed rates of the acid and base drift with time. In a SISO system, drift is usually easy to identify and thus easily removed. As the number of independent variables being stepped increases, it becomes more difficult to identify the drift. One approach used to solve this problem stems from the separation of the dynamics. Drift is usually a lower frequency process than the process dynamics of the system to be modeled. One effective technique used to remove this low frequency drift is to process the data through a high pass filter. As the frequency of the dynamics of the drift approach the frequency of dynamics of the process, this technique fails. In some situations, drift is addressed by increasing the number of steps in the step test and performing the tests on each independent variable intermittently over a longer period of time. This approach treats the drift as noise or as an unmeasured disturbance.

**Mean Center and Scale the Input and Output Data**

The Volterra-Laguerre model equations described in the previous chapter of this dissertation are based on deviation variables. Current industrial linear identification algorithms are based on deviation from the mean operating point during the test. This is true for the Volterra-Laguerre nonlinear identification technique used in this work as well. The linear portion of the identification algorithm used here makes the same assumptions about superposition as the industrially available packages. The nonlinear portion of the identification calculates a second order correction matrix for the linear model and is also dependent on the mean operating point during the step testing. In linear systems, this assumption is reasonable because of the principle of superposition. In linear systems, it is assumed that the process, as described with deviation variables, will behave exactly the same over the entire operating range of the equipment. The nonlinear portion of the controller recognizes that the process is nonlinear and therefore it cannot
be assumed that the principle superposition holds. For the nonlinear model, it is necessary to keep track of the identified operating conditions where the linearization was performed. The nonlinear corrections are placed in the matrix and applied to the prediction based on the deviation of the current operating point from the identified operating point.

Scaling is necessary in this identification for exactly the same reason it is important in the current industrially available identification packages. While performing the matrix inversion for the linear identification, the matrices can become ill-conditioned if scaling is not done. The mean centered and detrended data for the PH system are shown in Figure 49.

**Model Identification**

After the data have been conditioned, the Volterra-Laguerre model can be identified. Because modeled data that have not been used in the identification is the best indication of model performance, it is best not to use all of the data in the identification. A portion of the data should be reserved for examining model predictions on data not used in the identification. The fundamental identification algorithm consists of the following steps.

1. Define the time horizon of the process and the number of Volterra coefficients desired in the linear model. The time horizon is the longest time required for the full impact of an independent variable move to be seen in control variable. The number of Volterra coefficients should be selected based on the sample rate required to model the fastest dynamics important in the control of the process. The relationship between time horizon, execution frequency, and number of coefficients was discussed in Chapter 2 of this dissertation.

2. Identify the linear Laguerre equation coefficients based on the current Laguerre order. The residual squared error, the Laguerre coefficients, and the time scale factor returned from the identification should be stored for later use.
3. Identify the nonlinear Laguerre coefficients based on the current Laguerre order. The residual squared error, the Laguerre coefficients, and the time scale factor returned from the nonlinear identification should also be stored for later use. The reduction of the squared error from the linear identification to the nonlinear identification is an indication of the extent of nonlinearity in the system over the range excited by the step testing.

4. After at least two Volterra-Laguerre models have been identified, evaluate the improvement in the model performance gained by increasing the Laguerre order. Repeat steps 3 through 5 until the improvement from increasing the Laguerre order diminishes.

5. Use the final Laguerre coefficients to calculate the linear and second order Volterra coefficients for the Volterra model.
6. Evaluate the models ability to predict process responses by performing predictions on the data set not used in the identification.

This identification procedure was performed on the data collected for the PH system. The error reduction for the linear and nonlinear Volterra-Laguerre models on the data sets not used in the identification are shown in Figure 50 as a function of Laguerre order. The process was also identified using standard linear Volterra model techniques using 30 Volterra coefficients for illustration purposes only. The error reduction reported represents the fraction of the total error eliminated over the standard linear Volterra model. In this case, a Laguerre order of 4 was the point at which the improvement stabilized. Higher Laguerre order results are shown for informational purposes. Figure 51 illustrates the predictions on a data set not used in the identification.

In this experimental PH case, the nonlinear model eliminated over half of the residual error over linear control. The range of PH values being predicted in this example represented five orders of magnitude in concentration. Given this range and the extent of nonlinearity in the process, the results are very promising.
Figure 50 PH process ID Volterra-Laguerre improvement

Figure 51 PH identification fit with Laguerre order of four
CONCLUSIONS AND RECOMMENDATIONS

A general-purpose nonlinear identification technique has been developed for identifying linear and second order Volterra models for use in industrial model predictive control. The technique uses process step testing and input-output data that are consistent with currently used industrial techniques to obtain a data set used in the identification. This technique allows those in industry to extend the range of applicability of the multivariable model predictive control to a much broader class of processes that contain significant nonlinearities. The nonlinear identification takes advantage of Laguerre equations to reduce the coefficient requirements and thus the step testing requirements of identifying these models. This work reduces the number of step tests required for linear and nonlinear identification. Linear Volterra models typically require a range of 30 – 120 coefficients. The number of coefficients required was reduced to a range of 3 - 6 coefficients using the Volterra-Laguerre models. This significantly reduces the step test requirements for linear models. Nonlinear (second order) coefficient requirements were reduced from hundreds or thousands of Volterra coefficients to less than 50 Laguerre coefficients using this technique. The step test requirements for the nonlinear Volterra-Laguerre models were shown to be on par with current testing requirements for linear multivariable controllers applied throughout industry. The application of the nonlinear Volterra models to industrial processes has been limited to processes where fundamental models were available, because of the number of coefficients required to identify the models. This work eliminates that roadblock and makes practical the application on nonlinear multivariable control in industry for a broad range of nonlinear process where limited step testing can be performed. This work also shows that process where the gain sign changes can be modeled. This is particularly important in industrial
process where stoichiometry of reacting process often cause a process gain sign change in a controlled variable. This includes processes such as combustion, polymers, and other important reacting systems. Additionally, the extent of nonlinearity in the process can now be defined based on the linear and nonlinear model differences. This will provide invaluable insight into estimating the benefits of implementing the nonlinear portion of the controller.

The test cases in this dissertation demonstrated both the applicability and the benefits of extending the models used in model predictive control from linear models to nonlinear models. Three cases were studied and a range of 50%-90% of the error in linear models was eliminated by using the nonlinear Volterra-Laguerre models. The cases studied were selected based on their process characteristics that are usually difficult to model. The results showed that the Volterra-Laguerre model did very well at modeling difficult dynamics such as deadtime and inverse response. The Volterra-Laguerre models captured nonlinearities in both process gain and dynamics. The resulting nonlinear Volterra-Laguerre models also have all of the favorable characteristics of the Volterra models studied in other works. One major advantage of the Volterra-Laguerre model is that it allows for an easy and natural extension of the current multivariable control packages being applied in industry. The linear portion of the controller is identical to the currently applied technology. The nonlinear portion control is implemented as an auxiliary loop that corrects for process nonlinearity. The nonlinear correction loop can be selectively turned on or off depending on the control requirements of the system. In addition, the supporting infrastructure built around the current linear multivariable controller remains applicable. This means the migration requirements to incorporate nonlinear control in current industrial linear multivariable control packages is minimal.

The contribution of this work is meant to be in overcoming the hurdles in the application of nonlinear dynamic model identification of industrial processes using input/output data sets. To this end, a practically attractive choice of using Volterra-Laguerre models was selected. One of the major hurdles of the identification on practical systems is the input sequence used in the
identification process. This dissertation provided an industrially attractive approach that extends the pseudo-random multilevel input sequence. The approach pseudo-randomly varies amplitude and frequency and duration of steps made on the process. The range, amplitude, and frequency of the steps is limited based on prior knowledge of process behavior. Additional contribution was made in quantifying testing requirements to get acceptable nonlinear models. One significant hurdle in practice is dealing with the nonlinear time scale factor used in the Laguerre approximations. Previous works make suggestions based on linear dominant time constants of the process. In the second order Volterra-Laguerre approach, the time scale factor is not necessarily tied to the dominant time constant. This work takes a practical approach in dealing with this time factor. Finally, the technique was used on a real process. This added to the knowledge by practical aspects of data conditioning required to obtain good models. A general testing procedure for identifying models was created.

The main work for this dissertation was completed in 2000. Since then, there have been others who have done work that is related to the work in this dissertation. Doyle, Ogunnaike, and Pearson (2001) contains a thorough overview of how Volterra models can be used for identification; several practical and theoretical topics are discussed. Zheng and Zafirou (2004) contains a formulation for a generalized Volterra-Laguerre model which can be used for obtaining a model of order greater than 2. Israelsen (2010) contains a formulation for a MIMO model of generalized Volterra order and generalized Laguerre order to accommodate different order models for different manipulated variables and different Volterra terms. The appendix contains a brief overview of the extension of the equations to multi-input single output equations.

**Recommendations for Further Work**

This technology needs to be applied in industry to illustrate the ease of implementation and benefits associated with the nonlinear multivariable model predictive control. Currently,
work is being performed in conjunction with Flying J Inc. in their North Salt Lake refinery to implement this technology on an industrial process.

Another area where further development would be beneficial is cross terms associated with the fundamental Volterra model. The fundamental Volterra models assume that there is true independence between manipulated variables. In industry, there are many cases where manipulating one manipulated variable affects not only control variables but other manipulated variables. The fundamental Volterra model needs to be expanded to include cross terms between manipulated variables in order to account for these interactions. Unfortunately, this is a formidable task.

Though the Volterra second order models significantly extend the range of acceptable control for most nonlinear processes, it would be useful to have higher order modeling and identification capabilities. For some classes of nonlinearities, higher order Volterra models would continue to provide significant benefits. Work needs to be performed to allow for identification of higher order Volterra kernels on industrial processes.

Finally, this identification technology needs to be packaged with a user friendly GUI and commercialized to facilitate the technology transfer to industry.
APPENDIX

EXTENSION OF EQUATIONS TO MULTI-INPUT SINGLE OUTPUT SYSTEMS

Given the work done in the previous chapters, the final step is to extend the identification to a multi-input single output (MISO) identification. The multi-input multi-output (MIMO) identification is a recursive reapplication of the MISO identification for each of the dependent control variables in a control problem.

Recall from equation (3.13) that the output for a second order SISO system is:

\[ y(t) = y_1(t) + y_2(t) \]

This algorithm can easily be adapted to fit a MISO system as follows:

\[ y = y_1^1 + y_1^2 + y_2^1 + y_2^2 + \cdots + y_1^l + y_2^l \]  \hspace{1cm} (A.1)

where \( l \) is the total number of independent variables in the system. Recall that \( y_1^l \) is defined as follows from equation (2.22) (some liberties have been taken with notation to allow agreement with equation (A.1)):

\[ y_1^l = U_l \cdot \Phi \cdot \theta_i \]
$U_i$, $\Phi$, and $\theta$ are defined in Chapter 2. It follows that $\theta$ can be identified by performing the following calculation:

$$\theta = (U_i \cdot \Phi)^{-1} y_i$$  \hspace{1cm} (A.2)$$

Next, $y_2^i$ is defined in equation (3.23) (shown below for reference, again with some changes in notation).

$$y_2^i(t) = U_i^T \Phi D_i \Phi^T U_i$$

$\Phi$, $U_i$, and $D_i$ are defined in Chapter 3, they are shown below for reference:

$$
\Phi_i = \begin{bmatrix}
    l_0(0) & l_1(0) & l_2(0) & \cdots & l_r(0) \\
    l_0(T) & l_1(T) & l_2(T) & \cdots & l_r(T) \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    l_0(pT) & l_1(pT) & l_2(pT) & \cdots & l_r(pT)
\end{bmatrix}
$$

$$
U_i = \begin{bmatrix}
    0 & u_0^i & \cdots & u_{p-2}^i & u_{p-1}^i \\
    0 & 0 & u_0^i & \cdots & u_{p-2}^i \\
    \vdots & \ddots & \vdots & \vdots & \vdots \\
    \vdots & \ddots & u_0^i & \vdots & \vdots \\
    0 & 0 & \cdots & 0 & 0
\end{bmatrix}
$$

$$
D_i = \begin{bmatrix}
    \theta_{11}^i & \theta_{12}^i & \cdots & \theta_{1r}^i \\
    \theta_{21}^i & \theta_{22}^i & \cdots & \theta_{2r}^i \\
    \vdots & \ddots & \ddots & \vdots \\
    \theta_{r1}^i & \cdots & \theta_{rr}^i
\end{bmatrix}
$$
\( \Phi_i \) is the Laguerre model matrix for the \( i \)th independent variable. If the time scale factor “\( a \)” for all independent variables is the same, \( \Phi_i \) can be changed to \( \Phi \) which simplifies the calculations. 

\( D_i \) is the coefficient matrix for the \( i \)th independent variable and the control variable, and can be solved for by using the following equation from equation (3.26) (shown below for reference, with changes to match the notation in (A.1)):

\[
D_i = \left[ (\Phi^T U_i)(\Phi^T U_i)^T \right]^{-1} (\Phi^T U_i)y_i^\prime(t)(\Phi^T U_i)^T \left[ (\Phi^T U_i)(\Phi^T U_i)^T \right]^{-1}
\]

In the multivariable case, the length of the independent variable move matrices are all the same dimensions, i.e., the dimension of \( U_i = \) Dimension of \( U_{i+1} \) etc. The solution of \( D_1, D_2, \ldots, D_l \) then becomes a minimization problem to reduce the sum of the squared error between the residual error between \( y \) and \( y_1^i \) predicted from the first order Volterra term. This solution can be made easier by making \( U_i \) and \( \Phi \) into block diagonal matrices and taking the inverse or pseudo-inverse (if the matrix is not square) to obtain the least squares solution (\( y_1^i \) also needs to be stacked on itself to make dimensionality work). Just as with the SISO case, if the Laguerre time scale factor “\( a \)” is optimized, the problem becomes a nonlinear optimization.

\[
\min_{D_1, D_2, \ldots, D_l} S = \sum_{i=1}^{l} (y^i - y_1^i)^2 - (y_2^i)^2
\]
REFERENCES

Al-Baiyat, S., Sain, M., 1986. Control design with transfer functions associated to higher order Volterra kernels. 25th CDC, IEEE, 1306-1311


Israelsen, B.W., 2009. Documentation of the APCO, Inc. Nonlinear System Identification (NLI) Software, Not Yet Published, APCO Inc. Salt Lake City, UT.


Matlab, A Product of The Mathworks Inc, 3 Apple Hill Drive Natick, MA.


