DYNAMIC PAYLOAD ESTIMATION IN FOUR WHEEL DRIVE LOADERS

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Mechanical Engineering
University of Saskatchewan
Saskatoon

By
Jahmy J. Hindman

PERMISSION TO USE

In presenting this thesis/dissertation in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis/dissertation work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

Reference in this thesis/dissertation to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favouring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering
University of Saskatchewan
57 Campus Drive
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

Knowledge of the mass of the manipulated load (i.e. payload) in off-highway machines, particularly Four-Wheel-Drive Loaders is useful information for a variety of reasons ranging from knowledge of machine stability to ensuring compliance with transportation regulations. This knowledge is difficult to ascertain however. This dissertation concerns itself with delineating the motivations for, and difficulties in development of a dynamic payload weighing algorithm. The dissertation will describe how the new type of dynamic payload weighing algorithm was developed and progressively overcame some of these difficulties.

The payload mass estimate is dependent upon many different variables within the off-highway vehicle. These variables include static variability such as machining tolerances of the revolute joints in the linkage, mass of the linkage members, etc as well as dynamic variability such as whole-machine accelerations, hydraulic cylinder friction, pin joint friction, etc. Some initial effort was undertaken to understand the static variables in this problem first by studying the effects of machining tolerances on the working linkage kinematics in a four-wheel-drive loader. This effort showed that if the linkage members were machined within the tolerances prescribed by the design of the linkage components, the tolerance stack-up of the machining variability had very little impact on overall linkage kinematics.

Once some of the static dependent variables were understood in greater detail significant effort was undertaken to understand and compensate for the dynamic dependent variables of the estimation problem. The first algorithm took a simple approach of using the kinematic linkage model coupled with hydraulic cylinder pressure information to calculate a payload estimate directly. This algorithm did not account for many of the aforementioned dynamic variables (joint friction, machine acceleration, etc) but was computationally expedient. This work however produced payload estimates with error far greater than the 1% full scale value being targeted. Since this initial simplistic effort met with failure, a second algorithm was needed. The second algorithm was developed upon the information known about the limitations of the first algorithm. A suitable method of compensating for the non-linear dependent dynamic variables was needed. To address this dilemma, an artificial neural network approach was taken for the second algorithm.

The second algorithm's construction was to utilise an artificial neural network to capture the kinematic linkage characteristics and all other dynamic dependent variable behaviour and estimate the payload information based upon the linkage position and hydraulic cylinder pressures. This algorithm was trained using empirically collected data and then subjected to actual use in the field. This experiment showed that the dynamic complexity of the estimation problem was too large for a small (and computationally feasible) artificial neural network to characterize such that the error estimate was less than the 1% full scale requirement.

A third algorithm was required due to the failures of the first two. The third algorithm was constructed to take advantage of the kinematic model developed and utilise the artificial neural network's ability to perform

nonlinear mapping. As such, the third algorithm developed uses the kinematic model output as an input to the artificial neural network. This change from the second algorithm keeps the network from having to characterize the linkage kinematics and only forces the network to compensate for the dependent dynamic variables excluded by the kinematic linkage model. This algorithm showed significant improvement over the previous two but still did not meet the required 1% full scale requirement. The promise shown by this algorithm however was convincing enough that further effort was spent in trying to refine it to improve the accuracy.

The fourth algorithm developed proceeded with improving the third algorithm. This was accomplished by adding additional inputs to the artificial neural network that allowed the network to better compensate for the variables present in the problem. This effort produced an algorithm that, when subjected to actual field use, produced results very near the 1% full scale accuracy requirement. This algorithm could be improved upon slightly with better input data filtering and possibly adding additional network inputs.

The final algorithm produced results very near the desired accuracy. This algorithm was also novel in that for this estimation, the artificial neural network was not used solely as the means to characterize the problem for estimation purposes. Instead, much of the responsibility for the mathematical characterization of the problem was placed upon a kinematic linkage model that then fed it's own payload estimate into the neural network where the estimate was further refined during network training with calibration data and additional inputs. This method of nonlinear state estimation (i.e. utilising a neural network to compensate for nonlinear effects in conjunction with a first principles model) has not been seen previously in the literature. It should be mentioned that this is an applied study performed on one machine type (4WD loader) and investigates the use of one particular technology applied to this machine form.

ACKNOWLEDGEMENTS

At some point in my academic life I considered this dissertation to be the culmination of my education. I no longer believe that to be the case. I have come to the realization that education, whether formal or informal starts at the beginning of life and, if life is lived well, does not culminate. I believe this realization is of more significant importance than any of the research contained herein.

There are a multitude of people that have been directly and indirectly responsible for this work. Acknowledgement should be made to John Deere, specifically Lee Tucker, Mac Klingler and Brian Rauch who really started me down this path in life with their understanding that technical skills are a valuable commodity and are worth investing in. I am also indebted to a special group of Deere researchers and engineers on two continents that have impacted this work in some way with their critiques, suggestions or data gathering. That group in no particular order consists of Aki Putkonen, Marko Paakkunainen, Timo Kappi, Jeff Dobchuk, Eric Anderson, Cory Brant and Kevin Campbell. I am grateful for your assistance. I am also indebted to my faculty advisors, Dr. Richard Burton and Dr. Greg Schoenau who made my off-campus student tenure manageable and kept me accountable for progressing through the years. I would also me remiss if I did not acknowledge Doug Meyer and Kevin Funke who supported my education commitment with allowing schedule flexibility in attending technical conferences and time to write this dissertation. Finally, I am most indebted to my wife, Carrie and my children, Ryley, Tobias, and Ainsley for their forbearance as I have committed time to this effort.

This thesis is dedicated to my father, who demystified the higher education process for me at a young age and gave me a healthy understanding of the importance of practicality in research.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

4WD	Four Wheel Drive
ANN	Artificial Neural Network
MLP	Multi Layer Perceptron
LMS	Least Mean Squares
EKF	Extended Kalman Filter
CAN	Controller Area Network
FPU	Floating Point Unit

CHAPTER 1

INTRODUCTION AND OBJECTIVES

1.1 Introduction

The topic of dynamic parameter estimation is quite broad and has received much attention for control and condition monitoring purposes over the last forty years. This dissertation approaches the topic of parameter estimation from a practical perspective by trying to solve a problem that has been an issue in the off-highway industry for decades. As the off-highway industry has grown, the industry has become more regulated [1],[2],[3],[4],[5]. One of the regulations with which this dissertation is concerned is the load limit of trucks running on public highways. When these on-highway trucks are carrying aggregate loaded by an off-highway machine, the amount of material the off-highway machine has loaded into the truck is critical to ensure compliance to the appropriate transportation regulation [6],[7]. Traditionally, this has been done by having the operator of the off-highway machine estimate the truck load through experience and visual cues or through the use of an on-board measurement scale in the off-highway machine. The first approach of allowing the operator to estimate based upon past experience and visual cues has proven to be inaccurate. The latter method requires the off-highway machine to be pseudo-static using current technology. This in turn requires the off-highway machine operator to stop doing useful work in order to measure the load of the material. This is an inefficient operation in terms of productivity and fuel consumed.

This dissertation attempts to develop an algorithm capable of estimating the payload in an off-highway machine (Four Wheel Drive (4WD) loader) to a high degree of accuracy while allowing the machine to continue it's normal operation (i.e. remain dynamic). A successful implementation of this algorithm will provide a work cycle that is more fuel efficient than the alternatives while ensuring the on-highway trucks that are loaded with the 4WD loader are compliant with the weight restrictions of the pertinent transportation regulatory agency. It should be mentioned that this is an applied study performed on one machine type (4WD loader) and investigates the use of one particular technology applied to this machine form.

1.2 Objectives

The objective of this research work is to develop an algorithm that is capable of dynamically estimating, with 1.0% full scale accuracy, the amount of material contained in the bucket of a Four-Wheel-Drive (4WD)

loader. To further constrain the algorithm, it was required that it use sensors that are publicly available (i.e. "off-the shelf") and that the number of sensors used would be minimized while still maintaining the 1% accuracy requirement.

Additionally, the algorithm was required to run on a typical John Deere microcontroller which bounds the algorithm further by requiring a fixed point implementation that could run pseudo-real time (i.e. <150 milliseconds) on a 64MHz ST10 microprocessor. The research element of this work lies in solving the technical problem bounded by the mentioned constraints by utilising a novel approach to the estimation algorithm.

1.3 Layout of Dissertation

The bulk of this dissertation is comprised of four publications written and published in the course of this research work. That said, this dissertation will begin by providing some basic knowledge of the niche of dynamic payload estimation and off-highway vehicles (particularly 4WD loaders). This will allow the reader to delve into the more complex technical papers with enough familiarity to understand them and rationalize for themselves the progression of this work.

The order of the technical papers contained within this dissertation is chronological. This order is the order of the thought process used to understand the factors that influence the dynamic payload estimation problem and then determine a solution and refine it. The first paper describes the impact manufacturing variances have on the kinematic linkage model used in this research. This is accomplished through applying the manufacturing tolerances to a stochastic linkage model developed using a Monte Carlo simulation. The second paper concerns itself with development of the first three dynamic payload weighing algorithms and details the results. The natural progression from one algorithm to the next is shown in this paper describing the process used to refine the algorithm. The third paper describes the fourth algorithm developed for the estimation purpose. The fourth algorithm is defined and the results are shown in the this paper. The fourth paper describes a typical problem associated with utilising an ANN trained with experimental data. A solution for this problem is described for input data sets of n dimensions.

There are several unique terminology items that should be identified prior to delving into the subject matter. The first of these is the terminology *payload*. This term will refer to the mass of material being manipulated by the working tool of the vehicle. The second term to define is *loader*. For the purposes of this discussion, this term is used to reference a four-wheel-drive loader utilised in the construction and aggregate industries.

1.4 Contributions of the Primary Investigator

All papers are co-authored; however, it is the mutual understanding of the authors that Jahmy Hindman, as the first author, is the primary investigator of this research. The contribution of the other co-authors has been

limited to an advisory and editorial capacity. Some of the experimental work and data collection has been performed by staff of Deere and Co. and this is acknowledged.

CHAPTER 2

BACKGROUND

Prior to reviewing the technical papers included in this dissertation, it is important to introduce and discuss the two basic tenets contained in this dissertation. Towards that purpose, a section on off-highway vehicles with particular focus on the test article, a 4WD loader are discussed. The problem of payload estimation as applied to the test article are also presented. Finally, dynamic parameter estimation will be discussed with particular attention devoted to the development and use of artificial neural networks.

2.1 Four Wheel Drive Loaders

The Four Wheel Drive (4WD) loader moniker is generally understood to describe a machine that is propelled via rubber tires and is steered via an articulation joint connecting the front and rear frames. A working tool is typically connected to the front frame of the vehicle by means of some type of hydraulically actuated kinematic linkage. It is typical for the 4WD loader to be powered with an internal combustion engine (usually running the diesel cycle) and use a mechanical drivetrain consisting of a torque converter power-shift transmission connected by drivelines to the front and rear axles. 4WD loaders range in size from 700 kg to 263,000 kg operating mass. The first wheel loader was produced by Hough (International Harvester) in 1947 as the Hough HF one cubic yard loader [8]. An example of a typical 4WD loader can be seen in Figure 2.1

4WD loaders can be found in a variety of applications. These applications include excavation and earth-moving, moving revuse at landfills and transfer stations, loading and unloading scrap steel at recycling plants, unloading and moving processed trees for pulp and paper mills, municipal snow removal and moving and loading aggregate in stone quarries. It is this last application, moving and loading aggregate, to which this research is primarily directed.

2.1.1 Quarry Application

It is of use to describe the typical operation of a stone quarry in order to best understand how the 4WD is utilised and why payload estimation is important. The product at a typical quarry is various sizes of that quarry's aggregate (i.e. limestone, granite, shale, etc). The quarry process starts by excavating a "face" of a deposit of the desired aggregate. This typically takes the form of drilling into the deposit and then inserting

Figure 2.1: Deere 744K 4WD Loader

some sort of explosive in the drill cavities and igniting the explosive. The explosion is set up to be uni-directional (i.e. the force of the blast is concentrated on a 180 degree arc facing towards the excavation site). This blast typically reduces the aggregate face to pieces of a manageable size (under 1 meter in size). This material is then "face loaded" through use of a large 4WD loader into the hopper of the rock crusher. These operations can be seen in Figure 2.2 and Figure 2.3. The rock crusher crushes the aggregate to different sizes and each size falls through a sizing screen onto conveyors that move the material into stock piles. A second 4WD loader (typically smaller than the face machine) then loads aggregate from these stock piles into on-highway trucks for delivery to an end customer as seen in Figure 2.4

Figure 2.2: Deere 744K Face Excavating

Figure 2.3: Deere 744K Loading A Crusher

Figure 2.4: Deere 824K Loading A Truck

It is worthwhile to describe the truck loading process in more detail as that is when payload weighing becomes critical. An on-road truck typically will require being fully loaded with some size and type of aggregate. The full load rating of the truck is dictated by the state or province of operation and the number of axles under the truck. This load limit is known by the truck driver and is typically stencilled on the side of the truck for the loader operator to see. The loader operator then loads the truck with the desired aggregate trying to load the truck to exactly the maximum load limit and no more. This typically requires between two and four buckets of material depending upon the size of 4WD loader and size of the truck. The truck then exits the quarry by driving across a measuring scale at the quarry exit that has been certified by the state

or province's Department of Weights and Measures. The truck stops on this scale long enough to achieve an accurate reading. Typically if the reading is within a few percent *under* the maximum truck load, the truck leaves for it's destination. If the reading is higher than the maximum load, the truck must return to the stockpile and dump material out and re-weigh. If the reading is lower than a few percent under the weight limit, the truck returns to the stockpile to have additional material loaded. This trial-and-error process is tedious and costly in terms of time and fuel utilised. A 4WD loader with an accurate on-board scale helps to mitigate this problem by loading the truck to the correct weight the first time.

2.1.2 Machine Design

In order to understand the difficulties associated with payload estimation in 4WD loaders, it is also useful to have a basic understanding of the 4WD loader machine design particularly in the areas of the hydraulic system, machine dynamics and linkage kinematics. The most common linkage used in 4WD loaders is referred to as the "Z-bar" linkage and can be seen in Figure 2.5

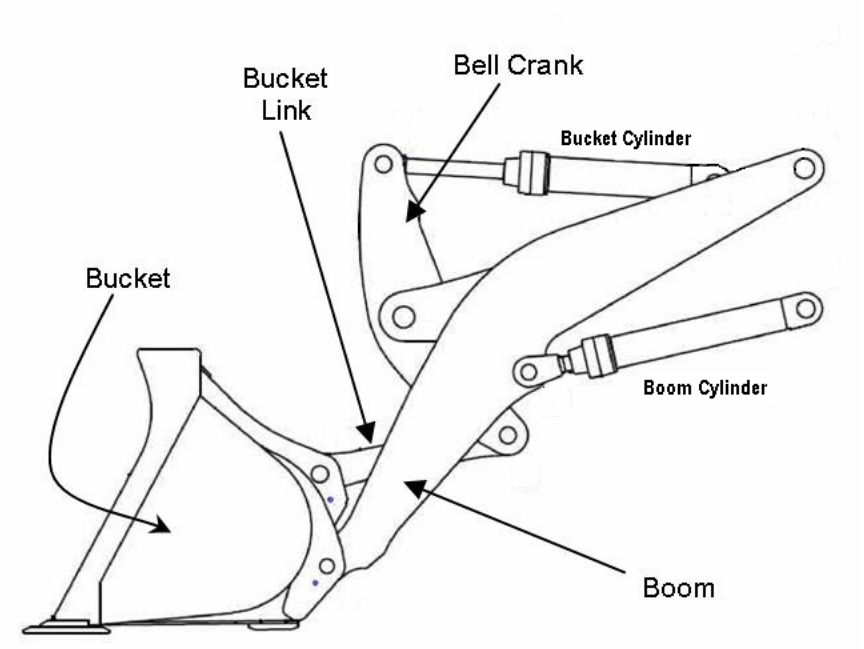


Figure 2.5: Z-Bar Loader Linkage

This linkage takes the name from the "Z" shape formed by the bucket link, bell crank and bucket cylinder geometry. The benefits of this style of linkage for a 4WD loader are two fold. First, the linkage produces significant mechanical advantage when trying to curl the bucket. This means that a higher curl force is produced than the amount of force the bucket cylinder imparts to the linkage. This characteristic provides what is termed in the industry as excellent "bucket breakout". This essentially means that the bucket function exhibits high enough forces that it can break through very dense or tough material. The second benefit of this linkage is that is relatively low weight for the forces it is capable of generating. This is useful in a wheel

loader application because of machine balance. Counterweight is added to the rear of a 4WD loader to offset the payload mass when the bucket is full. The heavier the linkage is on the front of the machine, the more counterweight needs to be added for fore-aft stability and the more the machine costs to both purchase and operate. The boom seen in the Z-bar linkage is actuated through use of two hydraulic "boom" cylinders. These cylinders provide the force necessary to lift the payload to the desired height prior to dumping.

The hydraulic systems in wheel loaders vary among manufacturers and age of the equipment. In general however, most modern wheel loaders rely on a closed center hydraulic system to perform the hydraulic related tasks required of the machine. A typical 4WD loader machine schematic may be found in Appendix A. In general terms, the hydraulic system typically consists of a single pump dedicated to providing oil flow to satisfy the loader boom, loader bucket, steering, and pilot flow. The pump is typically connected with hydraulic hoses to the main loader control valve that controls the boom and bucket functions. The loader control valve in turn is connected through a combination of hoses and tubes to the boom and bucket cylinders.

The machine dynamics of the pneumatic-tired 4WD loader are governed as usual by the mass properties of the machine and the coefficient of restitution acting between tires and the running surface. The dynamics of the operating machine create a forcing function that impacts the hydraulic system as well. This is primarily due to the whole-machine acceleration being rigidly coupled through the Z-bar linkage to the payload mass. Since there is a structural path connecting the frame to the mass, the hydraulic cylinder members of the linkage are forced to keep the linkage in equilibrium. This means that the pressures contained within these cylinders change in order to keep the linkage forces balanced in the *linkage* static state of the machine. A distinction is made here that the linkage may be static with respect to the machine while the machine is moving in rigid body motion.

2.1.3 Dynamic Payload Estimation Algorithm: Implementation Considerations

Implementation of a dynamic payload estimation algorithm on a 4WD loader is constrained by several factors including computational speed, accuracy and memory as well as system cost, sensor packaging and placement and hardware durability. Though the system cost, sensor packaging and durability are requirements that feed into the actual system design, the algorithm development needs to be concerned with computational efficiency for two primary reasons.

The first reason the estimation algorithm requires high computational efficiency is that this algorithm will be running on a microprocessor that is tasked with a multitude of other machine control functions. Many of these functions are critical to machine performance such as governing engine speed and transmission gear selection with respect to the operator inputs. Other critical machine control functions such as calculating and commanding the desired cooling fan speed and engaging traction control devices also utilize this microprocessor. If the payload weighing algorithm requires an exorbitant amount of time to perform an estimation, the microprocessor has less time to perform the machine critical control functions and the

overall machine performance suffers (i.e. engine speed lag, harsh gear shifting, etc). It is paramount then, given the finite computational resources on-board the machine, that the estimation algorithm be as efficient as possible.

The second reason the estimation algorithm requires computational efficiency is due to the nature of the way the 4WD loader machine is utilised in a truck loading application. In this application, the machine loads the bucket, backs away from the pile and propels forward to the truck to dump the material. The time from when the machine backs away from the pile (i.e. finished loading and the weight is determinate) and when the material is dumped into a truck can be as little as 10 seconds depending upon truck placement. Obviously, any algorithm that slows this process down is negatively affecting productivity of the machine. In addition, if an operator is required to wait for a load calculation when the machine could be doing productive work, the operator's desire to use the on-board weighing algorithm is diminished.

2.2 Dynamic Parameter Estimation

The heart of this study is the "estimation" of the payload under dynamic operating conditions. However,the topic of parameter estimation is quite broad. In order to make the topic more manageable to discuss, it will be broken down into some of the various methods employed to solve the estimation problem. Kalman filtering and extended Kalman filtering will be discussed briefly to expose the reader to these topics as viable parameter estimation methods as they compose a large part of the traditional work found in this area. Artificial Neural Networks (ANN) will be discussed in more depth since it is used in much detail in this research and other relevant research work.

2.2.1 Kalman Filter

Rudolph E. Kalman, Professor Emeritus in the Department of Mathematics and Electrical Engineering at the University of Florida, published his well-known paper describing an optimal recursive data processing algorithm, the Kalman filter in 1960 [9]. Due to modern advances in digital computing, the discrete Kalman filter has become the subject of a remarkable amount of research and has found application in a wide variety of fields including parameter estimation.

The discrete Kalman filter is essentially a very efficient computational method for minimizing the mean squared error. The strength of the Kalman filter is its ability to estimate not only past and present states and parameters of a system, but also future states and parameters from *apriori* system information. Additionally, the filter functions well even when there is uncertainty (noise) associated with the modeled system [10]. It is important to note here that the Kalman filter can be used as an estimation tool for both parameter and state estimation techniques. A brief discussion of the basics of the discrete Kalman filter will be provided.

In general, the discrete Kalman filter applies to the dilemma of estimating an unknown state or parameter of any discrete-time stochastic system. It is beneficial to understand the definition of "stochastic" before

proceeding. The term "stochastic" when used in the context of dynamic systems simply means that system or process involves some random variable (usually noise). In addition, the Kalman filter only applies to linear systems, meaning a linear change in the input to the system results in a linear change of the output. There has been significant success in applying a form of the Kalman filter to non-linear systems. This application is referred to as the Extended Kalman Filter (EKF) and will be discussed later. It follows then that the Kalman filter is applicable to all systems governed by a forward difference equation of the form:

$$x_{k+1} = A_k x_k + B u_k + w_k. \tag{2.1}$$

where a measurement of a state or parameter can be obtained that is represented by:

$$z_k = H_k x_k + v_k. \tag{2.2}$$

For a complete understanding of equations (2.1) and (2.2) it is beneficial to define the associated variables and assumptions. The matrix A_k relates the state or parameter at time step k to the state or parameter at time step $k+1$. The matrix B_k relates the input of the system,u, to the state or parameter x. The matrix H_k in equation (2.2) relates the state or parameter x to the measurement z_k. The variables w_k and v_k are stochastic variables and represent the system and measurement noise with the assumption that the noise be white and have a normal probability distribution. This has been termed a "predictor-corrector" algorithm by Welch and Bishop [10]. Additionally, [10] provides a good illustration of the steps by which the filter operates as seen in Figure 2.6.

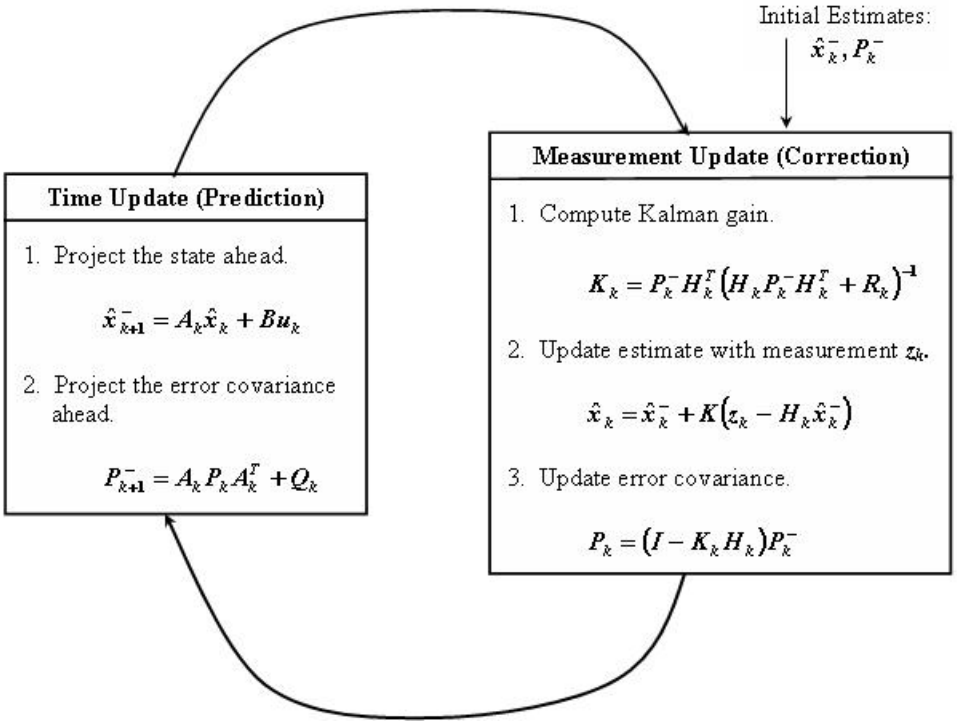


Figure 2.6: Kalman Filter Cycle and Equations

The Kalman filter is a useful algorithm, but is applicable for linear systems only. However, although it does not apply to non-linear systems, it's application to the highly non-linear problem of dynamic payload estimation is still relevant. A variant of the Kalman filter, called the Extended Kalman Filter (EKF) is used for non-linear systems. This variant is derived from linearizing the Kalman filter equations about some discrete operating point and then proceeding through the same process as the traditional Kalman filter. The Extended Kalman Filter was not chosen for this research work primarily due to it's inherent memory-intensive calculations.

2.2.2 Artificial Neural Networks

Due to the fact that Artificial Neural Networks is the estimation technique used in this research, it's background and operation will be discussed in greater detail. Artificial Neural Networks (ANN) have been a field of significant research in a wide variety of applications since the early 1960's [11]. These applications have included speech and pattern recognition [12], adaptive filtering and signal processing [13], weather forecasting [14], adaptive control [15], and adaptive noise canceling [16]. This list is not exhaustive and, due to the extraordinarily wide range of applications, a complete and comprehensive compilation may well prove infeasible. A cursory examination of the theoretical aspects of ANN technology will be provided in the following section.

2.2.2.1 Artificial Neural Network Theory

ANN's can be divided into two distinct categories: dynamic and static networks. Dynamic networks contain an effective memory, that is, their output is a function of all previous inputs and outputs. Static networks have no memory, forcing their output to be a function of the current input only. The static neural network was conceived initially and will be used for discussion purposes as it is used in this research.

There are many different forms of static neural networks. These include the Multi-Layer Perceptron (MLP), the Radial Basis Function (RBF), ADALINE and MADALINE among others. Since the MLP is perhaps the most widely used form of the static network, it will serve as the example used in the theoretical discussion.

The MLP is descended from the single perceptron devised by Rosenblatt in 1958 [17] . In order to understand the workings of the perceptron, it is beneficial to visualize the process. To this end, note the four-input perceptron in Figure 2.7. The input to the perceptron shown in Figure 2.7 is the four dimensional input vector given by:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \tag{2.3}$$

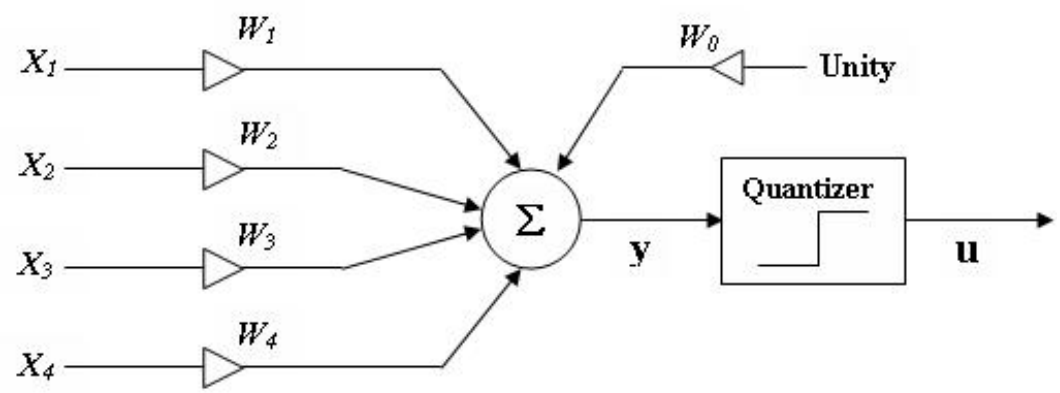


Figure 2.7: Four Input Perceptron

Likewise, the weights are given by the following four dimensional weight vector:

$$W = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} \tag{2.4}$$

The weight vector given in (2.4) is determined by training the network. This training is accomplished by either a *supervised* learning algorithm in which the weights are modified per a given set of input/output pairs, or an *unsupervised* learning algorithm in which the network is subjected to known inputs and the network groups these inputs into similar classes. The additional weight (W_0) shown in the figure is an external bias. The sum of the input vector multiplied by the respective weights is then passed through a nonlinear function (termed activation or squashing function). The original activation function given by Rosenblatt in [17] is a hard-limiting function. An example of this function can be seen in Figure 2.8. A superficial examination

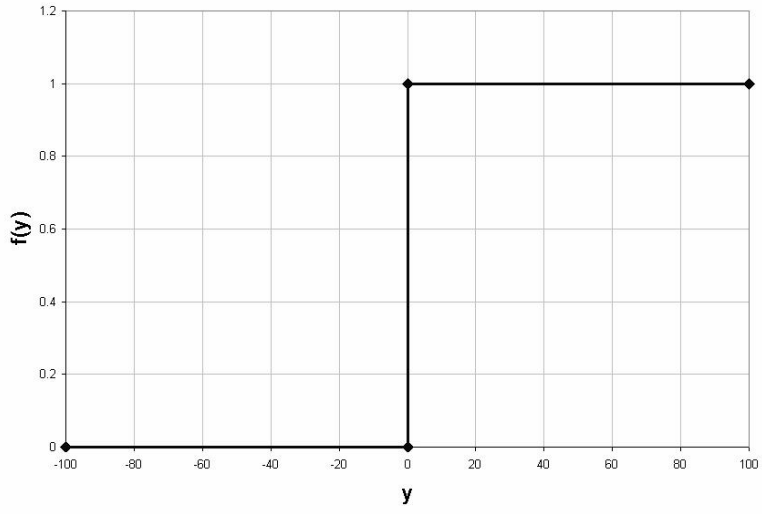


Figure 2.8: Hard Limit Function

of this function reveals that it is not differentiable due to the discontinuity at $y = 0$. This discontinuity is a disadvantage of this particular function because many of the popular training algorithms rely on a gradient search method applied to the activation function. If the activation function is not differentiable, the gradient methods used in some training algorithms are not applicable. This dilemma was solved by introducing a continuous nonlinear activation function in place of the hard limiting function. The sigmoid function fits this criteria and is perhaps the most well known activation function. It provides a continuous function and varies monotonically from 0 to 1 as the input varies from +∞ to -∞. The sigmoid activation function can be characterized by:

$$f_s(y) = \frac{1}{\left(1 + e^{-\zeta y}\right)} \tag{2.5}$$

The gain of this function, ζ, determines the gradient of the transition region within the function. This behavior can be seen in Figure 2.9. It is of interest to note that the sigmoid function devolves to the hard

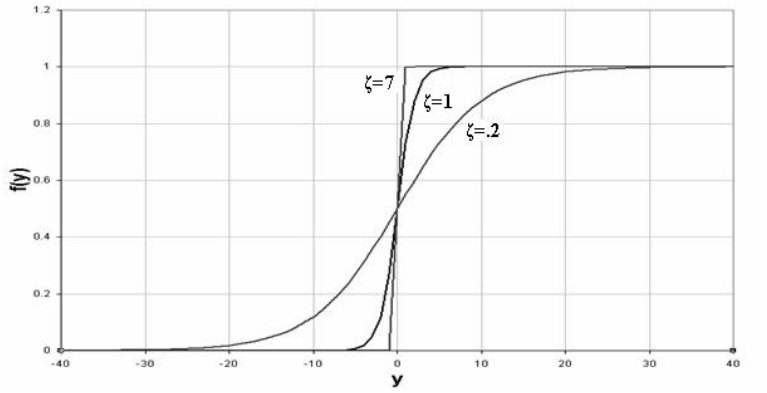


Figure 2.9: Logarithmic Sigmoid Activation Function

limiting function as ζ approaches infinity. In addition to the benefit of differentiability, the sigmoid function is often useful in applications requiring a continuous output rather than the binary output of the hard limiting function. Another common activation function is the hyperbolic tangent function. This function is a form of the sigmoid function and is given as:

$$f_s(y) = \frac{\left(1 + e^{-\zeta y}\right)}{\left(1 + e^{-\zeta y}\right)} \tag{2.6}$$

As with the logarithmic sigmoid function, the hyperbolic tangent function behaviour is also determined by the gain ζ as can be seen in Figure 2.10 The similarities between the hyperbolic tangent function seen in Figure 2.10 and the sigmoid function seen in Figure 2.9 can be ascertained from investigating the two functions. The fundamental difference is that the logarithmic sigmoid function is centered at $y = .5$ and the hyperbolic tangent function is centered at $y = 0$. The final activation function to note is the linear activation function. This function can be seen in Figure 2.11 This function is given by:

$$f(y) = \kappa y \tag{2.7}$$

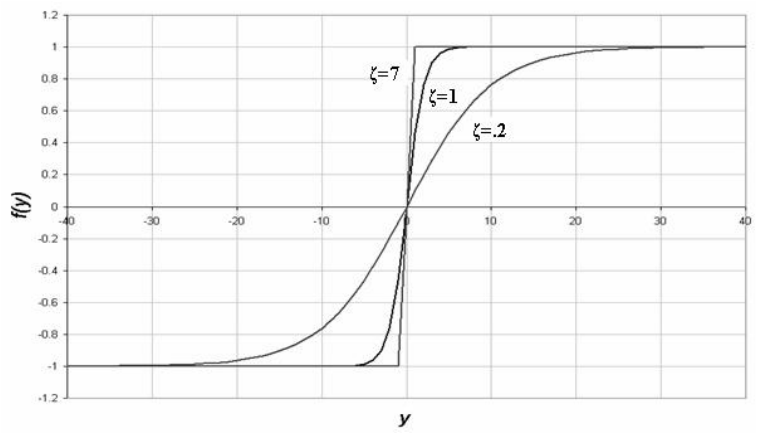


Figure 2.10: Hyperbolic Tan Sigmoid Activation Function

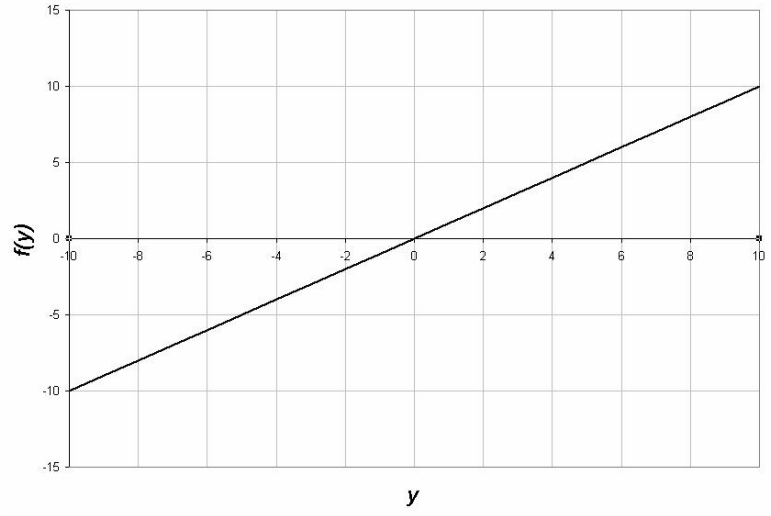


Figure 2.11: Linear Activation Function

The constant κ in (2.4) denotes the slope of the linear function. In Figure 2.11, the slope is set to unity. Upon inspection of all of these activation functions, the question arises of when each function is applicable in the construction of an ANN. This determination is largely problem specific and even then may need to be approached on a trial and error basis although it is rational if the desired outputs of a trained ANN are never negative to use logarithmic sigmoid functions as this would preclude negative values in the output. It should also be mentioned that the functions noted above represent a small sample of activation functions used in ANN technology.

Now that the basics of the perceptron have been described, the Multi-Layer Perceptron (MLP) will be investigated. The MLP did not find significant application until 1986 when a suitable backpropagation learning algorithm was introduced by Rumelhart et al. [18], [19]. There are an infinite number of configurations of MLP networks, but they all follow common rules. MLP networks are all composed of simple perceptrons, similar to the structure given in Figure 2.7, organized in a hierarchical structure giving a feed-forward

network. Some general MLP network classifications are given in Figure 2.12 and Figure 2.13. Note that

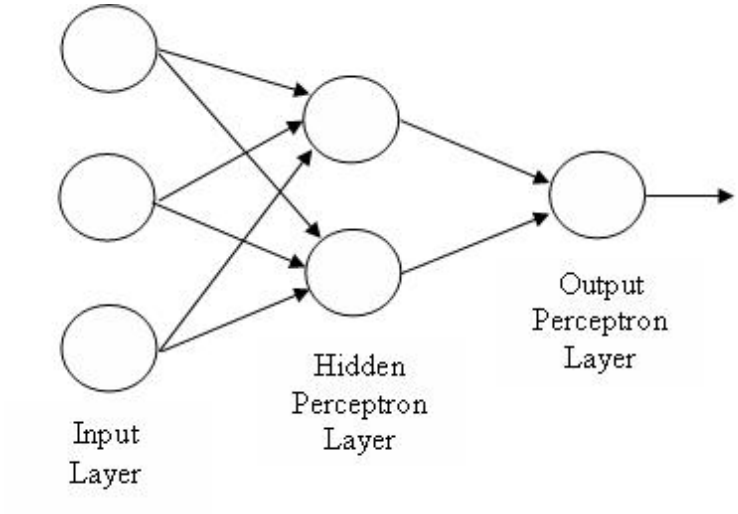


Figure 2.12: Two Layer MLP

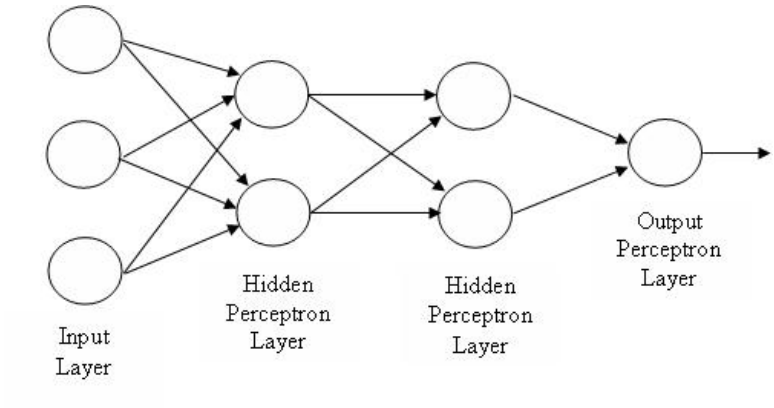


Figure 2.13: Three Layer MLP

these networks can be characterized as feed-forward (i.e. all the arrows point in the same direction). In addition, it is common to include the input layer when determining the number of layers in the network. This is not always the case however as there is no clear academic or industrial standard. It is obvious that the two networks are different in that they contain a different number of layers and perceptrons. This manifests itself in the ability (or limitations) of each network. In general, the two-layer network shown in Figure 2.12 is capable of regionally separating complex nonlinear problems.

It should be mentioned that the single layer network (a single perceptron) is only capable of solving linearly separable problems. If the problem is nonlinear in nature, multiple layers must be used to determine a solution. For example, consider a network tasked with separating the regions where $X \geq Y$ on the interval of $-5 < X < 5$ and $-5 < Y < 5$ as shown in the top surface of Figure 2.14. An example of a linearly unseparable problem can be seen in the bottom surface of Figure 2.14 where $X^2 \geq 1 - Y^2$ on the same

intervals. In the linearly separable problem, a straight line can be drawn separating the two regions. In the linearly unseparable problem, this is not the case. It is worthwhile to note that the solution provided from a MLP network to a nonlinear problem will be piecewise linear if the hard limiting activation function is used. If the activation function is modified to a sigmoid function or some other continuous function, the network output becomes smoothed and continuous.

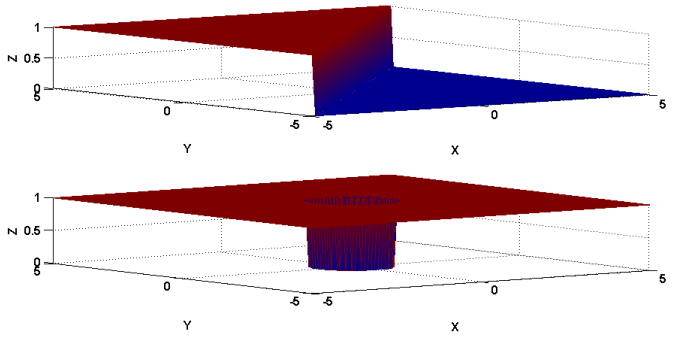


Figure 2.14: Linearly Separable (top) and Unseparable (bottom) Problems

A general investigation of the backpropagation learning method is now considered. As mentioned previously, the backpropagation learning method was popularized by Rumelhart et al in [18] and [19]. Currently it is the most widely used learning method in use in the field of ANN technology [20]. It should be noted that backpropagation as a learning method falls into the category of *supervised* learning. In order to understand the backpropogation learning method, it is beneficial to start with a simplified approach. To that end, consider a single layer network consisting of an input layer and an output layer. In addition, further simplify this by eliminating the activation function. This network is illustrated in Figure 2.15. In general, the input vector is given by:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_p \end{bmatrix} \tag{2.8}$$

In general, the output vector is defined as:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \tag{2.9}$$

The backpropogation method is contingent upon knowledge of some desired input/output data set. With this

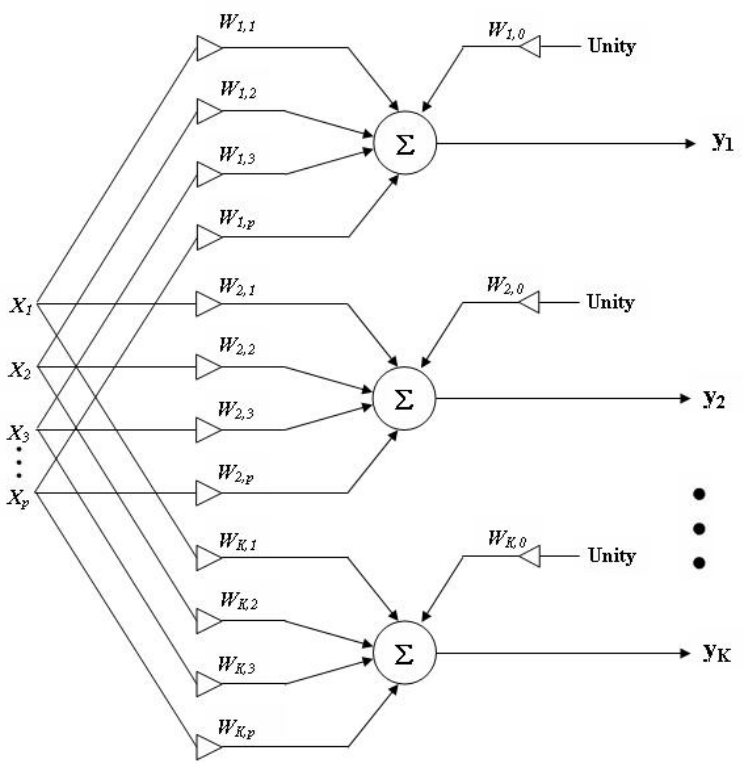


Figure 2.15: Single Layer Network

knowledge, the desired output vector is defined as:

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} \tag{2.10}$$

Equations (2.9) and (2.10) can then be combined to provide the error between the desired output and the actual output. This is given by:

$$e = (d - y) \tag{2.11}$$

Once an expression for the error is known, a gradient descent approach coupled with a Least Mean Square (LMS) error term provides a weight learning law in discrete form:

$$W(k+1) = W(k) + \mu \frac{\partial e_X}{\partial W} \tag{2.12}$$

μ in (2.12)is a scalar gain, also termed a learning rate factor, that is greater than zero. When applying this learning algorithm on a network with K output nodes, the LMS error can be determined by using (2.11) and becomes:

$$e_X = \frac{1}{2} \sum_{k=1}^{K} (d_k - y_k)^2 \tag{2.13}$$

There are several interesting details to note about the algorithm given in (2.12). The first detail of interest is that the value of μ is critical. If this value is set too high, the learning algorithm may not find the minimum

error due to the large step sizes μ would impose on the weight change. Additionally, a small value of μ provides very slow convergence. Ideally, the value of μ would be large if the error gradient was large. Conversely, as the error gradient became smaller μ would decrease. This behavior would provide optimal convergence in a minimum time. This is not trivial to implement however and, as such, care must be taken in choosing a constant for this value. The other detail to make note of is that the weight change given by the last term in equation (2.12) is driven to zero as the error gradient goes to zero. Depending upon the level of accuracy required by the network and the amount of time available to train the network, it may be infeasible to force the error gradient to identically zero. If this is the case, a limit can be set on the weight change term that provides adequate accuracy without forcing the error to identically zero.

This algorithm operates by starting with an initial set of weights in the network. The known input is then applied to the network and the network output is used in conjunction with the desired output to determine the error. The weights are then updated according to equation (2.12). With these new weights, the input at the next time step is again applied and the output of the network compared to the desired output. If the error calculated at this step is larger than at the previous step, the algorithm is progressing up the error gradient and needs to be reversed. This can be accomplished by subtracting the weight change in equation (2.12) instead of adding it to the weight of the previous step. This will effectively change the direction of the learning algorithm. This is necessary in order that the algorithm does not produce a solution that is a global or local maximum of the error function. If this is the case, the learning algorithm finds the worst set of weights that characterize the input/output relationship.

It is also important to understand that this algorithm does not guarantee finding the *global* minimum error solution. If the error function is highly nonlinear, the algorithm can easily produce a solution given by a *local* minima. This behavior can be visualized from inspecting Figure 2.16. The problem of finding a

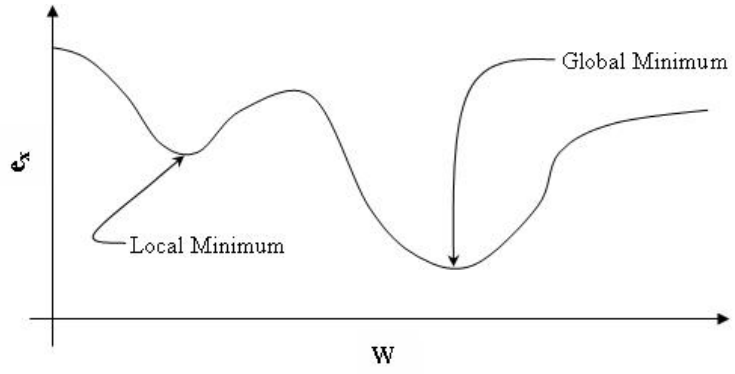


Figure 2.16: Local/Global Error Minimum

solution at a local minima can be alleviated by the addition of a "momentum" term to the learning algorithm. The momentum term concept will be introduced anon.

The learning algorithm for the single layer network with no activation function has been detailed above. The question remains of instituting a learning algorithm for a network composed of more than a single

layer with activation functions. This is not a trivial problem as the amount of error that each hidden node contributes must be calculated. As was mentioned previously, this was not possible with the use of a hard-limiting activation function given in Figure 2.8. It was not until the continuous activation functions were investigated that backpropagation became a feasible learning method. The continuous activation functions allow for the chain rule of partial derivatives to be used in calculating the weight changes for any weight in the network. Consider the two layer network with sigmoid activation functions shown in Figure 2.17. In this

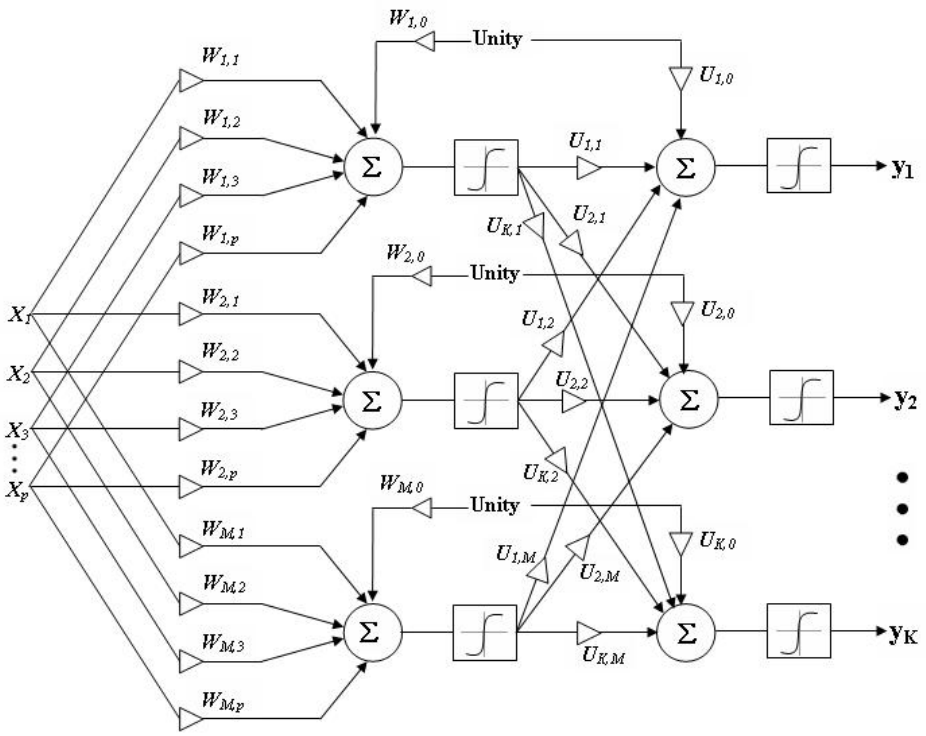


Figure 2.17: Two Layer Network

network, there are p inputs and K outputs. In addition, there are M nodes in the hidden layer and K nodes in the output layer. To clarify, the dimension of the input vector is $p \times 1$. The dimension of the output vector is $K \times 1$. The weight matrix W for the hidden layer is $((M+1) \times p)$ in dimension (including the external bias weights). The weight matrix for the output layer U is $(K \times (M+1))$ in dimension. In general, the gradient learning laws for both sets of weights can be written as:

$$W(k+1) = W(k) + \mu \frac{\partial e_X}{\partial W} \tag{2.14}$$

$$U(k+1) = U(k) + \eta \frac{\partial e_X}{\partial U} \tag{2.15}$$

μ and η are positive scalar gains determining the learning rate. The output for this network is some nonlinear function:

$$y = f(X, W, U) \tag{2.16}$$

The problem of adjusting the weights per the learning laws of (2.14) and (2.15) now becomes an issue since

the contribution of each weight to the error of the network is unknown. This is where the chain rule is utilized. This issue will be examined in some detail. The output of the network can be expressed in a more exhaustive fashion than given in equation (2.16) by including the known activation functions. This gives:

$$y = f_2(Uf_1(W,X)) \tag{2.17}$$

f_2 represents the output layer activation function and f_1 represents the hidden layer activation function. For the purposes of this discussion, f2 and f1 will be considered identical log sigmoid functions. Equation (2.17) can be expressed in more detail by including the summing junctions to give:

$$h_j = f_1\left(\sum_{i=0}^{p} W_{j,i}X_i\right) j = 1,2,\ldots M \tag{2.18}$$

where h_j is the output of the hidden layer.

$$y_k = f_2\left(\sum_{j=0}^{M} U_{k,j}h_j\right) k = 1,2,\ldots K \tag{2.19}$$

where y_k is the output of the output layer. The log sigmoid activation function is expressed as

$$f(z) = f_1(z) = f_2(z) = \frac{1}{1+e^{-z}} \tag{2.20}$$

The scaling parameter ζ that was given for the log sigmoid function in equation (2.5) has been set to unity to aid in simplifying the mathematical manipulation without sacrificing an accurate explanation of the fundamentals of the algorithm.

The change in error due to the output layer weights U will be investigated first. The partial derivative with respect to the error can be defined using the chain rule as:

$$\frac{\partial e_X}{\partial U_k, j} = \frac{\partial e_X}{\partial y_k}\frac{\partial y_k}{\partial U_k, j} \tag{2.21}$$

As can be seen, the partial derivative of the error function with respect to the output layer weights is composed of the partial derivative of the error function with respect to network output multiplied by the partial derivative of the network output with respect to the weights. Substituting the error function given in equation (2.12) and the expression for the network output given in (2.19) into (2.21) gives:

$$\frac{\partial e_X}{\partial U_k, j} = \frac{\partial\left[\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\right]}{\partial y_k} \times \frac{\partial\left[f_2\left(\sum_{j=0}^{M} U_{k,j}h_j\right)\right]}{\partial U_k, j} \tag{2.22}$$

Performing the derivative operation gives:

$$\frac{\partial e_X}{\partial U_k, j} = (y_k - d_k) \times f_2'\left(\sum_{j=0}^{M} U_{k,j}h_j\right) h_j \tag{2.23}$$

where f_2' is the derivative of (2.20) giving:

$$f_2' = \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right) \tag{2.24}$$

Grouping terms in (2.23) provides the following expression:

$$\frac{\partial e_X}{\partial U_k, j} = \delta y_k h_j \tag{2.25}$$

where δy_k represents the backpropagating error of changes in the output weights related to the hidden layer and is give by:

$$\delta y_k = (y_k - d_k) \times f_2' \left(\sum_{j=0}^{M} U_{k,j} h_j \right) \tag{2.26}$$

The change in error due to the hidden layer weights is not as straight forward since there is no feasible method of obtaining the desired outputs of the hidden layer for comparison to the actual output. This forces the partial derivative to relate the actual output of this system giving:

$$\frac{\partial e_X}{\partial W_j, i} = \frac{\partial e_X}{\partial y_k} \frac{\partial y_k}{\partial h_j} \frac{\partial h_j}{\partial W_j, i} \tag{2.27}$$

Performing the appropriate partial derivatives of (2.12),(2.18),and (2.19) gives:

$$\frac{\partial e_X}{\partial W_j, i} = \left[\sum_{k=1}^{K} (y_k - d_k) f_2' \left(\sum_{j=0}^{M} U_{k,j} h_j \right) U_{k,j} \right] f_1' \left(\sum_{i=0}^{p} W_{j,i} X_i \right) X_i \tag{2.28}$$

Alternately:

$$\frac{\partial e_X}{\partial W_j, i} = \delta h_j X_i \tag{2.29}$$

where:

$$\delta h_j = \left[\sum_{k=1}^{K} (y_k - d_k) f_2' \left(\sum_{j=0}^{M} U_{k,j} h_j \right) U_{k,j} \right] f_1' \left(\sum_{i=0}^{p} W_{j,i} X_i \right) \tag{2.30}$$

δh_j represents the backpropagation of the error from the output of the hidden layer. Now that equations (2.25) and (2.29) are developed, the weight adjustments can be made in order to minimize the error. In order to accomplish this, the weights should be adjusted in the opposite direction of the error gradient at each presentation of an input/output pair. Substituting (2.25) and (2.29) back into the learning laws given in (2.14) and (2.15) provides:

$$W(k+1) = W(k) + \mu \delta h_j X_i \tag{2.31}$$

$$U(k+1) = U(k) + \eta \delta y_k h_j \tag{2.32}$$

It is worthwhile to note again that the sign of the second term may change depending upon a positive or negative slope of the error gradient. It is also possible to express these learning laws in terms of the actual input, output and desired values by differentiating the activation function(s) and simplifying. This is particularly simple when using the sigmoid activation function since its derivative is simply a function of itself. In particular:

$$f' = f(1 - f) \tag{2.33}$$

The exercise of simplifying (2.31) and (2.32) into expressions of only input, output and desired values is given in Appendix B.

It was mentioned previously that a "momentum" term could be added to the learning equations that would increase convergence and possibly assist in escaping local minimum solutions. This momentum term simply provides a method of updating the weights by incorporating some information obtained about past updates. This manifests itself in equation form by adding a term to the learning equations as follows:

$$W(k+1) = W(k) + \mu \frac{\partial e_X}{\partial W} + \alpha (W(k) - W(k-1)) \tag{2.34}$$

$$U(k+1) = U(k) + \eta \frac{\partial e_X}{\partial U} + \beta (U(k) - U(k-1)) \tag{2.35}$$

From inspection of (2.34) and (2.35), when the gradient has the same algebraic sign on consecutive iterations, the weight change grows in magnitude. This can be thought of as the momentum accelerating on consecutive positive or negative gradients. When the sign of the gradient alternates between iterations, the weight change becomes smaller. This helps to "dampen" the oscillatory behavior of the training algorithm and facilitates convergence.

2.2.2.2 Artificial Neural Network Application

Now that a basic understanding of ANN theory has been provided, an example of this methodology to parameter estimation will be undertaken. The mass-spring-damper system of Figure 2.18 will be used. It is

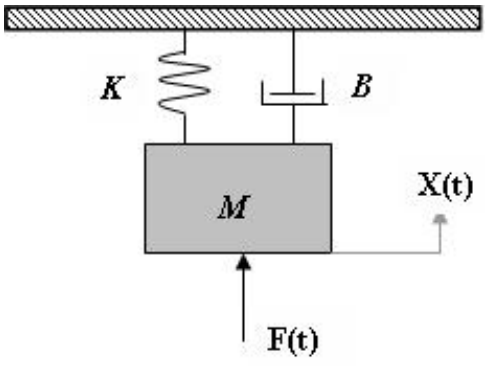


Figure 2.18: Mass-Spring-Damper System

necessary to set up a network capable of estimating the mass, spring constant, and damping parameters. The network shown in Figure 2.19 will be used for this purpose. This network was trained using backpropagation

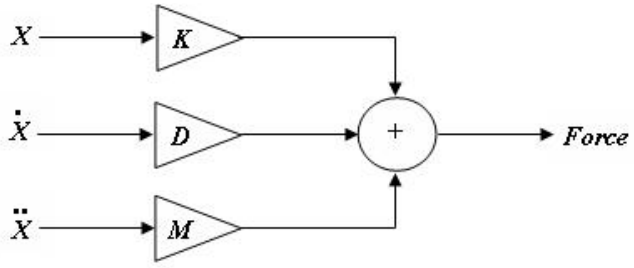


Figure 2.19: Parameter Estimation Network

over five epochs of input and output data taken from a Simulink ®model of the mass-spring-damper system. This model can be seen in Appendix C. Upon completion of the training algorithm, the weights of the network representing the mass, spring constant, and damping of the system converge to their final values. These values represent the estimation of the parameters given by the network. The estimation of the mass can be seen in Figure 2.20.

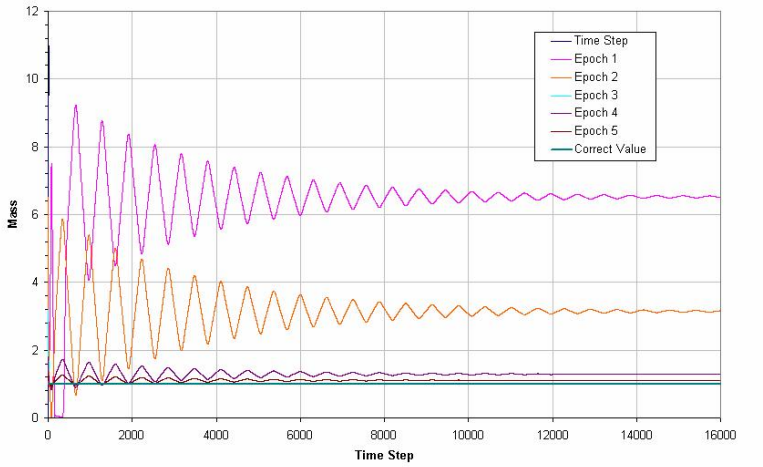


Figure 2.20: Mass Estimation

It can be seen from inspection of Figure 2.20 that the estimation of the mass has almost completely converged by the fifth epoch. The estimated value of the mass at the end of the fifth epoch is 1.106 compared to the correct value of 1.0. This estimate becomes more accurate when the network is subjected to more training epochs. The damping value is estimated as well and is shown in Figure 2.21. The estimate of the

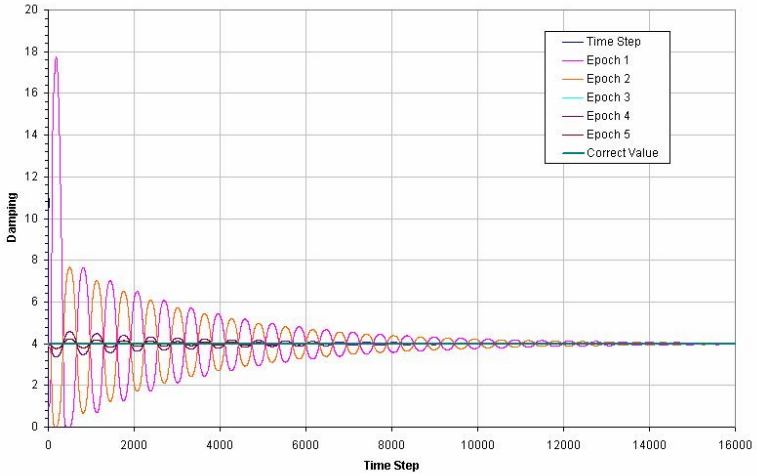


Figure 2.21: Damping Estimation

damping value is determined at the end of the fifth epoch to be 3.996 as compared to the actual value of 4.0. While this estimate is quite accurate, it can still be improved by further training. The estimate of the

spring constant is given in Figure 2.22. The final estimate for the spring constant after the fifth epoch is

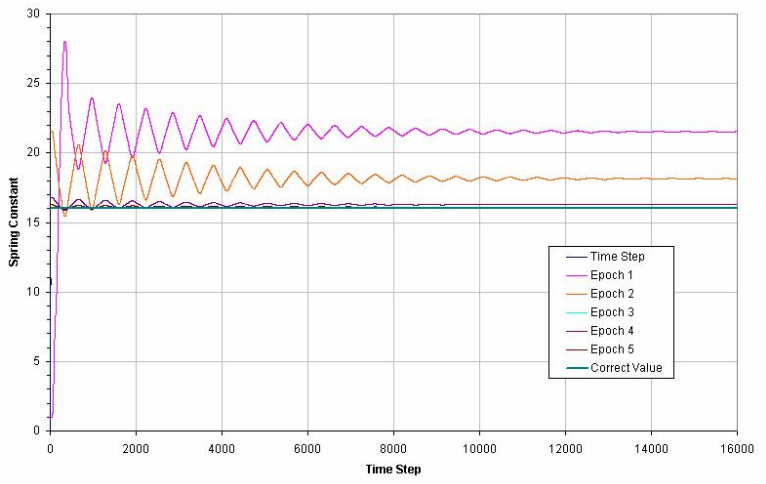


Figure 2.22: Spring Estimation

16.01 compared to the actual value of 16.0.

In addition to estimating the parameters, it is also of interest to track the mean squared error during the training epochs. This information is captured in Figure 2.23. The mean squared error shows a definite

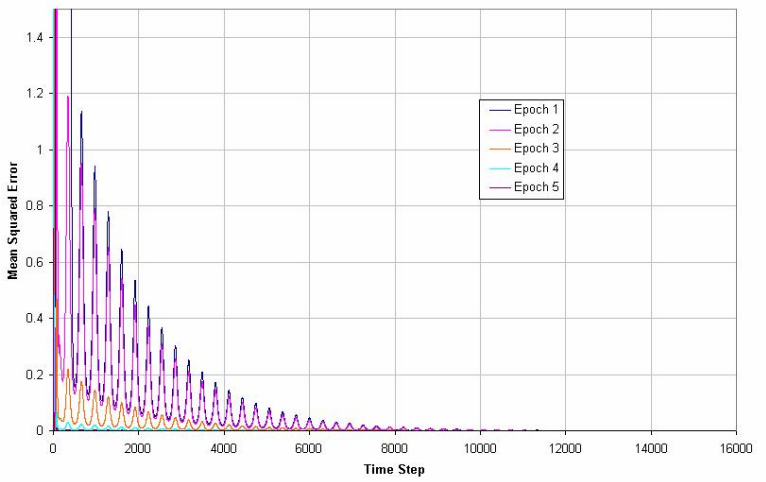


Figure 2.23: Mean Squared Error

decrease throughout the epochs with the minimum error occurring at the end of the fifth epoch. This behavior would continue if additional epochs were presented to the network. The MATLAB f̂extregistered code used to train the network is given in Appendix D.

CHAPTER 3

STOCHASTIC MODELLING OF A FOUR WHEEL DRIVE LOADER LINKAGE

Published as:

- Jahmy Hindman, Rich Burton, Greg Schoenau, "Stochastic Modelling of a Four Wheel Drive Loader Linkage". of SAE Commercial Vehicle Conference. 2006.

3.1 Objectives

This paper presents a stochastic 4WD loader linkage model. This model was used to determine what impact manufacturing variability might have on the overall linkage kinematics of the Z-bar linkage. This is an important step in understanding error that might be introduced by manufacturing variation to any estimation algorithm that is dependent upon a kinematic model of the linkage.

3.2 Methods

A kinematic model of the Z-bar loader linkage for a 644J Deere loader was developed. A Monte Carlo approach was then utilized to randomly distribute 10,000 points constrained by the positional tolerance specified by the linkage structural member design. Each set of these 10,000 points were then swept through discrete boom and bucket cylinder lengths to determine the impact of this tolerance on the position of the bucket (and by proxy, the load) center of gravity.

3.3 Results

The results of this analysis showed that, if the manufacturing process is under control and the design intent of the linkage is being met, the difference between an assumed dimensionally nominal kinematic model and the actual machine linkage was quite small (+/-.2%). This amount of error if not calibrated out, impacts the goal of +/-1% full scale accuracy by using 20% of the allowable error in kinematic dimensional error alone.

3.4 Contributions

The major contribution of the paper is the development of a stochastic method for analysing linkage manufacturing variations. Prior to this publication, the author knows of no other publication that provides a method to examine the affect of positional variation in kinematic linkages on the overall linkage performance.

Stochastic Modeling of a Four Wheel Drive Loader Linkage

Jahmy Hindman, Richard Burton, Greg Schoenau
University of Saskatchewan

ABSTRACT

The manufacturing tolerances of off-highway machine linkages have an impact on linkage position during machine use. A study was undertaken to determine the extent of these tolerances on linkage position for standard machining tolerances of each linkage pin joint for a four wheel drive loader linkage. Linkage positions and distributions for each pin joint are shown in order to determine positional accuracy of the working tool connected to the linkage and impact to the machine loads. It is determined that the affect of the machining tolerances on this linkage have a very minor impact on the linkage position when the linkage is in new condition (before use) and the maximum variation in linkage actuation loads is less than 1% of nominal load.

INTRODUCTION

Stochastic modeling of physical systems has been the subject of significant study in attempts to describe highly nonlinear systems. Stochastic modeling involves developing a mathematical model taking into account some element of randomness. This type of model is different than the more typical deterministic model because the output is a range of solutions characterized by a probability distribution instead of a single valued solution. Pitsch et al [1] describe the use of stochastic modeling in understanding the affect of scalar dissipation rate in non-premixed turbulent combustion. This study demonstrates the value in approaching highly nonlinear physical behavior from a stochastic approach. Wang et al [2] utilize a stochastic model (Kalman Filter) to minimize position errors in a real-time global positioning system with good results. Christensen [3] offers a stochastic model to estimate the distribution of crack initiation time in concrete reinforced structures. This model assumes a Wiebull distribution and shows good correlation to both Monte Carlo simulation as well as empirical data. Levene et al [4] offer a unique approach in stochastic modeling by applying a stochastic model to the evolution of the Web. A stochastic model is offered to characterize the distribution of incoming links, outgoing links, number of web pages and number of visitors to the web site. Wittner et al [5] proposed a linearized method for direct estimation of linkage position distribution and compared results favorably with a Monte Carlo simulation for an example linkage.

For off-highway mobile equipment, the revolute joints characterized by the pin joints in the linkage are machined (bored) in the large structural members comprising the linkage. The machining operation for the pin joint bores in the structure is not exact due to the tolerance of the locating fixture and machine tool during manufacturing. This study will focus on the affects of these machining tolerances on the position of the working tool (i.e. bucket) of a four-wheel-drive loader as seen in Figure 1.

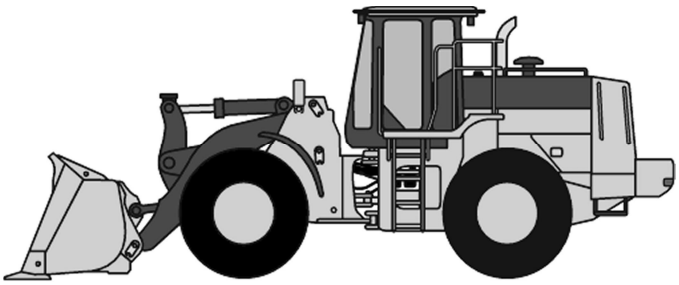


Figure 1: Four-Wheel-Drive Loader

Particular attention will be paid to the affects on position of the bucket (working tool) and the affects of the working tool position on linkage forces.

MAIN SECTION

This study involves a detailed analysis of a typical four-wheel drive loader linkage for a wheel loader in the 15 metric ton size class. The linkage to be analyzed is typically referred to as a z-bar style linkage which is a common linkage used in the wheel loader industry.

LINKAGE DESCRIPTION

The z-bar linkage consists of the boom, bell crank, and bucket as is shown in Figure 2. This linkage is actuated by three hydraulic cylinders. Two cylinders are used in parallel to actuate the boom (BD in Figure 2) and one cylinder is used to actuate the bucket (CH in Fugure 2). A two dimensional forward-kinematic model of the z-bar

linkage was derived in order to determine the linkage position for any combination of boom or bucket hydraulic cylinder lengths. Each linkage member was assumed to be a rigid structural member. Each of the revolute pin joints (A,B,C,D,E,F,G,H,I) were assumed to have a radial location tolerance of .2mm as determined from the manufacturing locating fixture and machine tool.

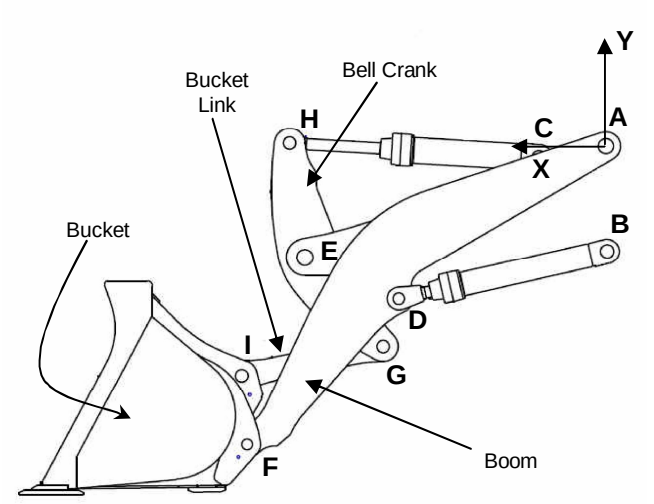


Figure 2: Z-bar Linkage

It is important to note that the tolerance defined here is not radial clearance between pin and bore, but the positional tolerance of the pin bore center. This tolerance can be visualized with Figure 3.

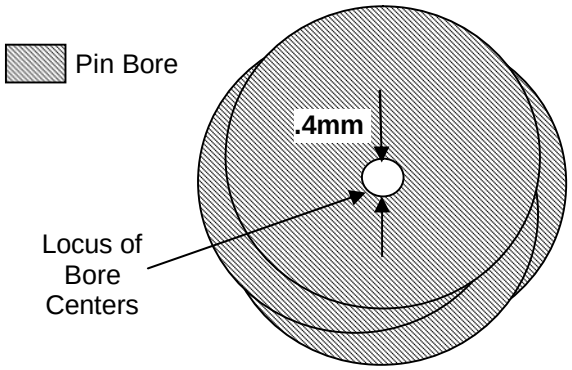


Figure 3: Pin Bore Locations

STOCHASTIC MODELING

It is useful at this point to provide a brief introduction to stochastic modeling. Stochastic modeling is useful in describing the mathematics of a random phenomenon. The definition of random here is a phenomenon that is too complex to describe exactly (i.e. does not lend itself to a deterministic solution). The outcome of the stochastic model is not a single value, but a distribution of probable outcomes. It should be stated that the stochastic model relies on the developer of the model to assume some input distribution to the model. This *a priori* information is important in terms of impact on overall model outputs and this distribution should be grounded in an understanding of the natural phenomenon surrounding the process to be modeled. One method for developing a stochastic model is to utilize a Monte Carlo algorithm The Monte Carlo method is useful in simulating the behavior of random processes. The Monte Carlo method typically require a large pseudo-random input sequence (determined by the chosen *a priori* input distribution). The output for each input point is then numerically calculated. This method is computationally intensive which limits it's usefulness in real-time applications.

LINKAGE STOCHASTIC MODEL

The stochastic model of the loader linkage was accomplished by assuming the pin bore center is normally distributed within the .2mm radial tolerance in both the X and Y direction for each revolute pin joint. A Monte Carlo simulation was employed to develop 10,000 points normally distributed in both the X and Y direction for each joint. The basic kinematic model was then adjusted for the changes in link lengths due to one set of the 10,000 points and the model was swept through discrete hydraulic cylinder lengths for that set of points. This process was repeated for all 10,000 data points. Due to the large scale required to show the complete linkage, it is not possible to visualize the distribution around each pin joint because of the relatively small change in position at each pin joint. A closer view of each pin joint in the linkage position shown in Figure 4 is given in Figures 5 through 10. These figures show the expected distribution for the base joints of the linkage (A, B, C) and increasing positional error with increasing distance in both directions from these base joints. The error is compounded in secondary members of the linkage due to the possibility of positional error between the two mating bore centers (E,G,H,I). It is possible to visualize the covariance (error) ellipse representing the boundary of constant probability. It is interesting to note the inferred shape of the covariance ellipse change significantly from the base joints to the joints near the working tool.

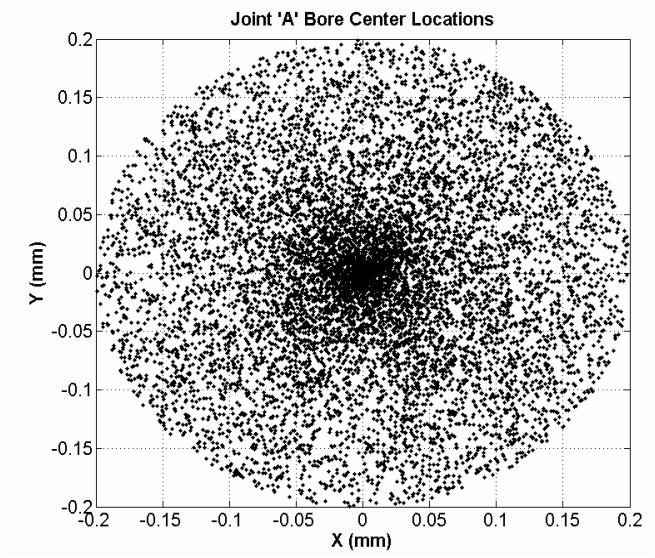


Figure 4: Joint 'A' Distribution

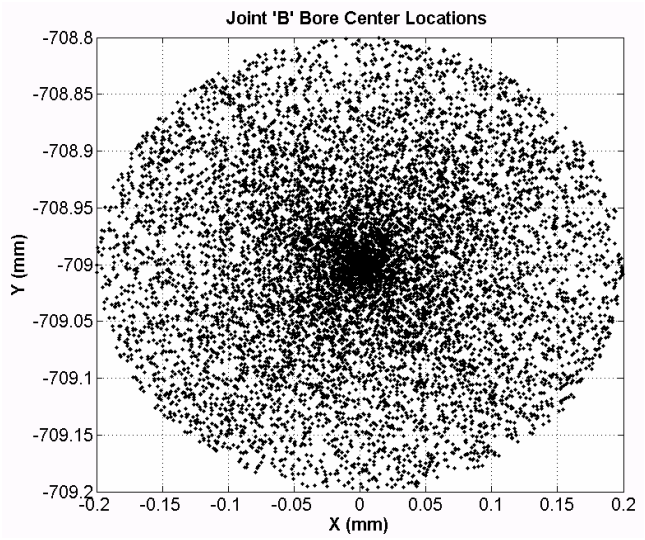


Figure 5: Joint 'B' Distribution

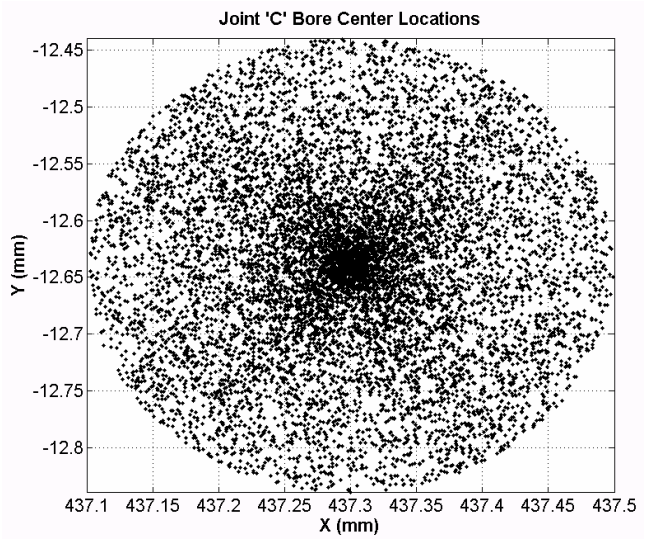


Figure 6: Joint 'C' Distribution

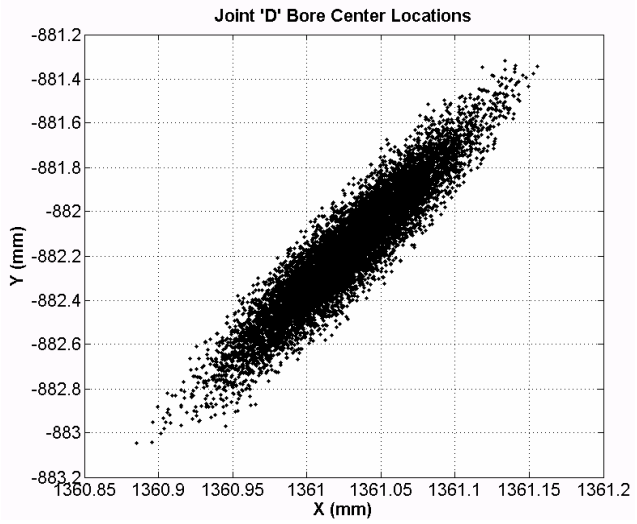


Figure 7: Joint 'D' Distribution (Derived From Kinematic Equations)

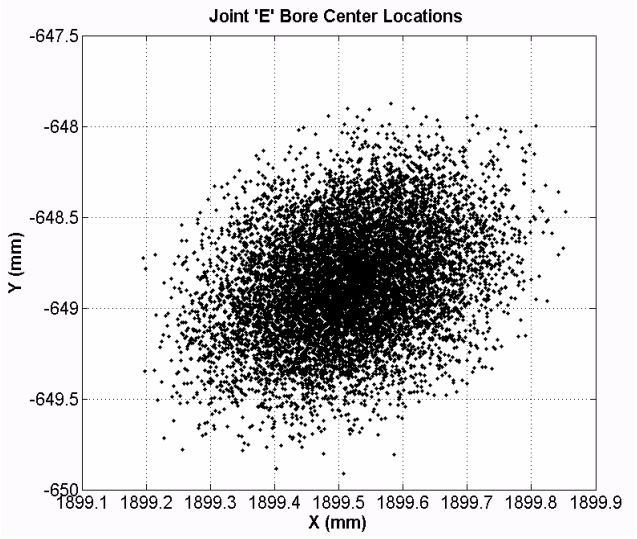


Figure 8: Joint 'E' Distribution (Derived From Kinematic Equations)

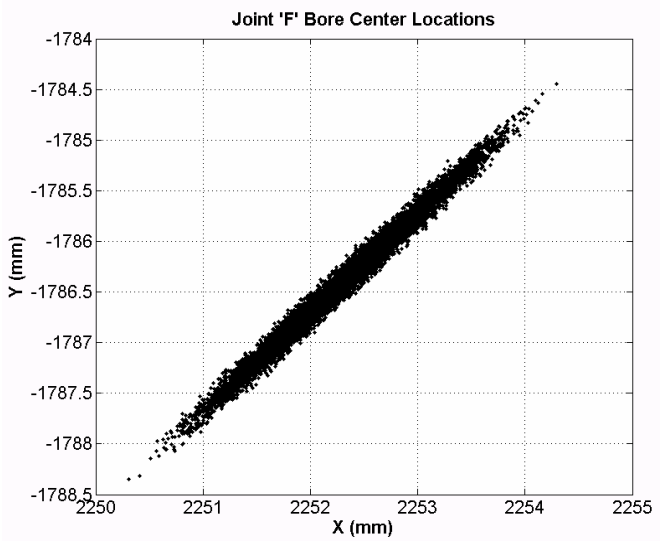


Figure 9: Joint 'F' Distribution (Derived From Kinematic Equations)

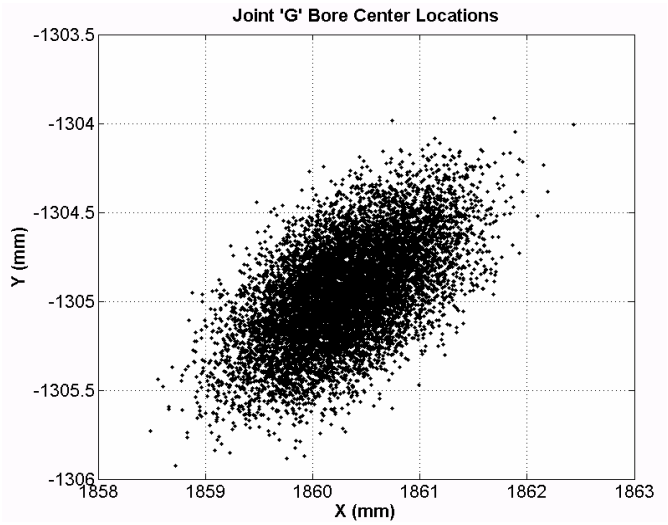


Figure 10: Joint 'G' Distribution (Derived From Kinematic Equations)

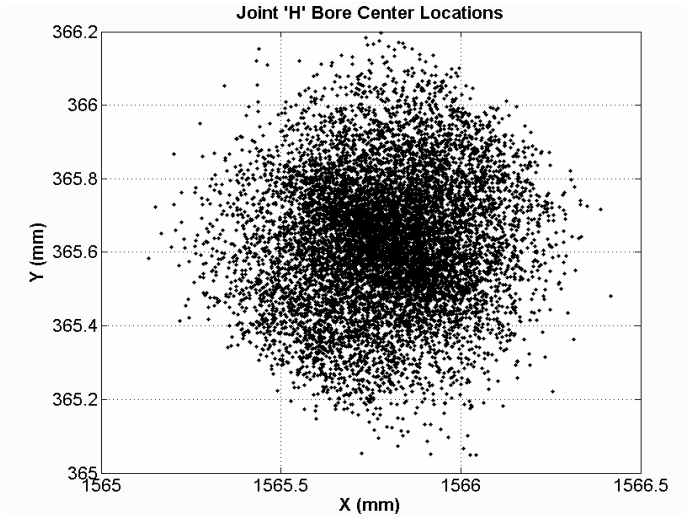


Figure 11: Joint 'H' Distribution (Derived From Kinematic Equations)

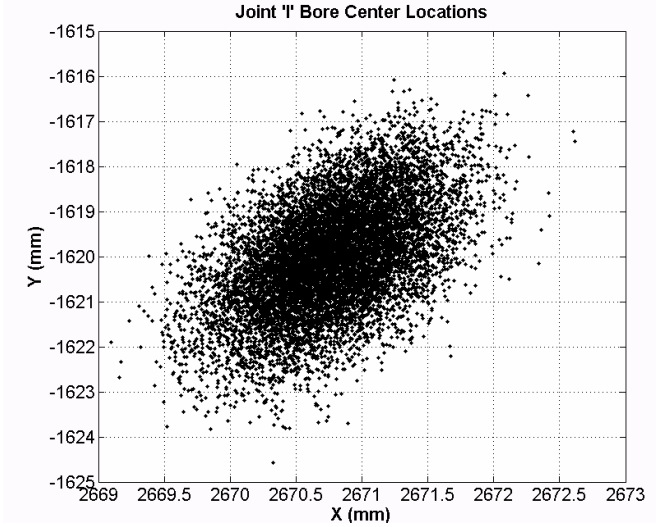


Figure 12: Joint 'I' Distribution (Derived From Kinematic Equations)

From the preceding figures, the distribution of each joint position is seen. It is worth noting that if a deterministic model were utilized, the kinematic model would produce only the single solution (mean) at the center of the error ellipse. This type of model would leave the distribution around that mean unknown and as such the extent of the error in the joint positions would not be known. The maximum variation in X and Y direction for joint D is (.2746mm, 1.7832mm). The maximum variation for joint E is (.6789mm, 1.9748mm). The maximum variation for joint F is (4.118mm, 4.038mm). The maximum variation for joint G is (3.7115mm, 1.9578mm). The variation for joint H is (1.2371mm, 1.2271mm). The variation for joint I is (3.2724mm, 7.9418mm). It is clear that the distribution range increases with distance from the base points (A,B,C). It is important to understand the probability of these distributions as well. This is shown in Figures 13 through 20.

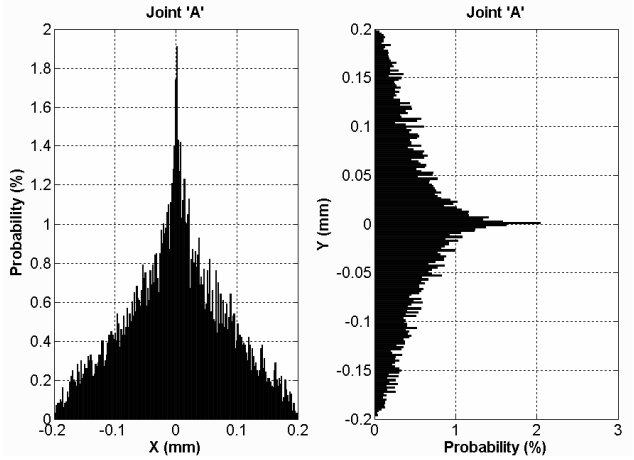


Figure 13: Joint 'A' Probability

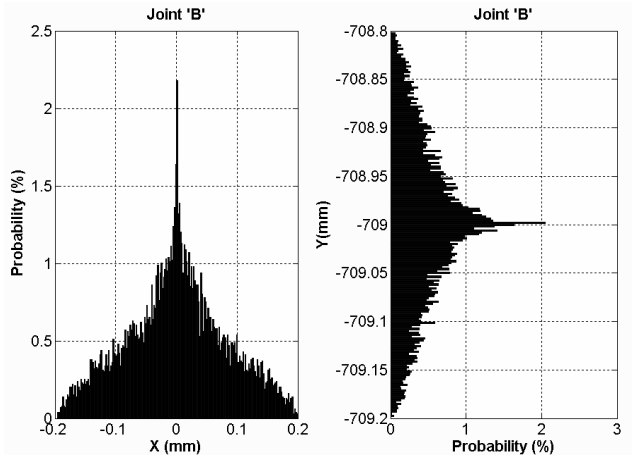


Figure 14: Joint 'B' Probability

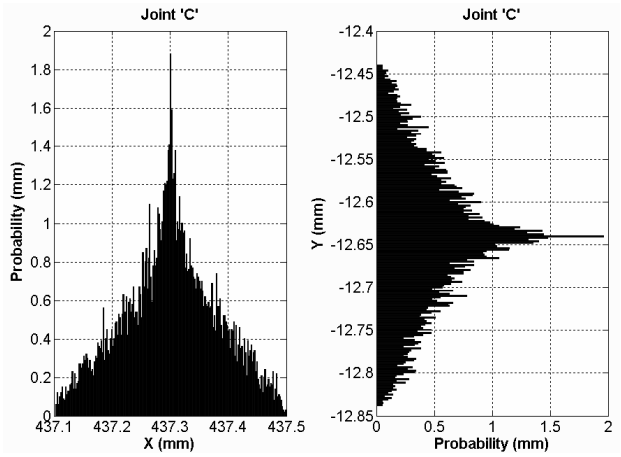


Figure 15: Joint ‘C’ Probability

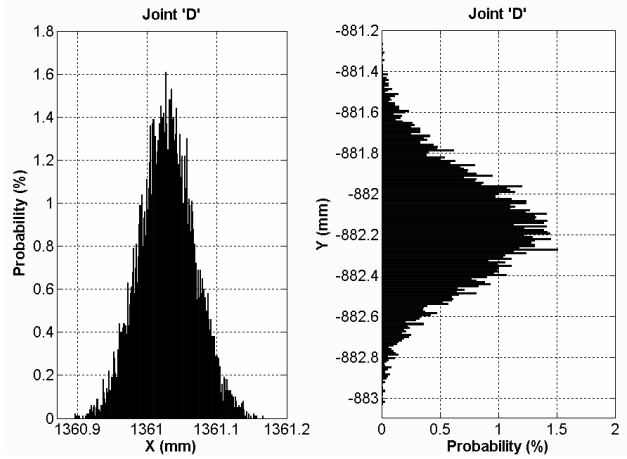


Figure 16: Joint ‘D’ Probability

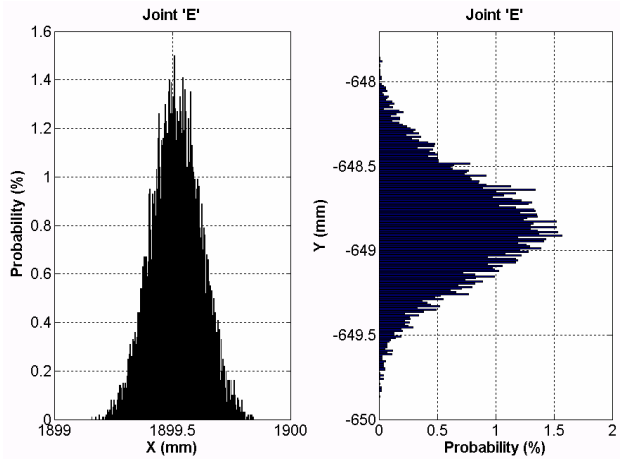


Figure 17: Joint ‘E’ Probability

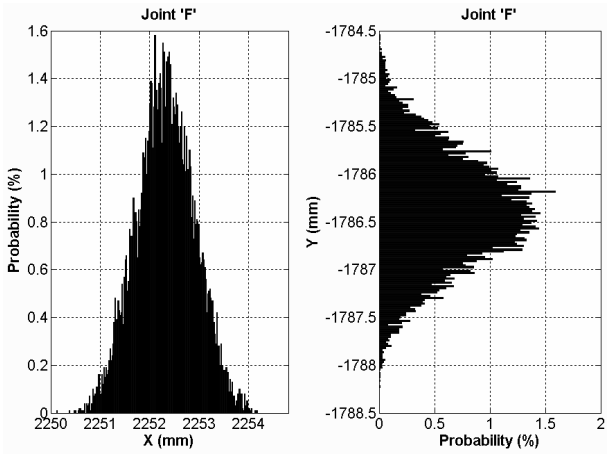


Figure 18: Joint ‘E’ Probability

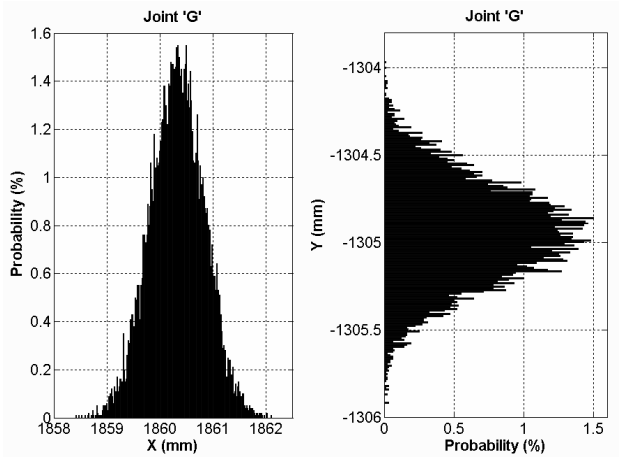


Figure 19: Joint ‘E’ Probability

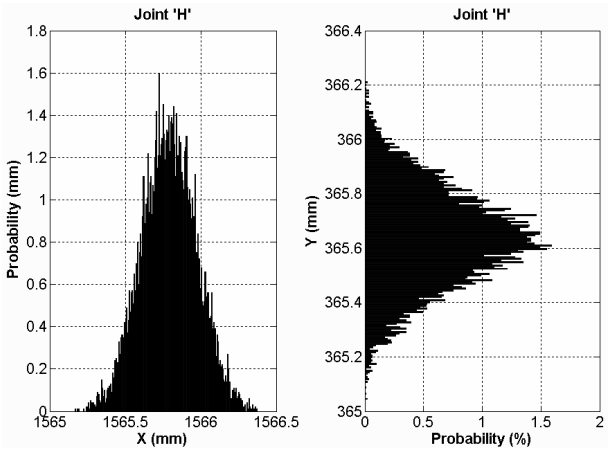


Figure 20: Joint ‘E’ Probability

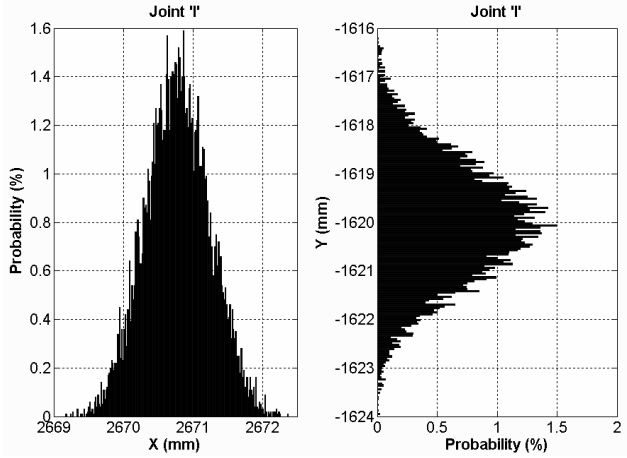


Figure 21: Joint 'E' Probability

EFFECT OF TOLERANCE ON REQUIRED LIFT FORCE

As can be seen, the variance of the joint position increases significantly with increasing distance from the base joint 'A'. For example, the standard deviation in X and Y directions for joint 'A' is (.0818, .0814). The standard deviation for joint 'F' is (.590, .578). It is important to put this positional variation in terms of the impact to linkage link lengths however to understand the full impact on machine kinematics.

The variation in link length from joint 'A' to the bucket pivot joint 'F' is 2874.7mm +/- 2.89mm. This position error amounts to .2% of the nominal link length and thus the bucket pivot joint 'F' will be located within +/-.2% of its nominal position. This shows that the machining tolerances of the linkage do not have a large affect on the positional accuracy of the linkage. It should be noted however that this same machining tolerance does impact the loads introduced to the structure by the same order of magnitude as the positional error. The maximum lifting capacity of the linkage in a maximum reach position (position F equivalent to a nominal (0,2874.7)) can be expected to vary because of the variance in geometry change as well. For instance, if the maximum lift force at the maximum reach linkage position was nominally 5000kg, the variance in this lift force would range from 4990.0 kg to 5010.0 kg. The impact of the tolerance to lifting force does not change with respect to boom position unless the positional variation changes with boom position. An investigation on that phenomenon will be the subject of a separate paper.

This impact is not significant in terms of overall machine performance, but does have an impact on issues concerning model based control of the linkage when the model is developed from nominal characteristics. In addition, estimating the mass of some amount of material in the bucket or on the working tool based on nominal kinematics would leave the error in such a calculation unknown without a stochastic model.

MODEL METHODOLOGY

The methodology used in constructing this model begins with an accurate forward kinematic model of the linkage. The model used in this example is a two dimensional (2D) model simplified from the actual three dimensional (3D) linkage. It should be noted that this simplification precludes any effects of pin joint misalignment due to inaccurate pin joint bore perpendicularity or axis alignment. These phenomena may have a significant effect on the overall linkage behavior by increasing and complicating the friction terms associated with the linkage. Whether a 2D or 3D model is developed, once this activity is complete the reference upon which the linkage error should be determined. This is typically determined from the geometrical dimensions and tolerances taken from manufacturing drawing of the linkage. The tolerances taken from this drawing represent the design intent of the linkage, not necessarily the manufacturing capability of the linkage machining. This can be an important distinction if the manufacturing process is not under process control.

Once the error reference is determined, the tolerance of the reference joint should be the first joint examined. This can be done by randomly distributing the joint tolerance among the tolerance space.

$$X_{1,n} = rand(-tol_{1,X}, +tol_{1,X}) \quad (1)$$

$$Y_{1,n} = rand(-tol_{1,Y}, +tol_{1,Y}) \quad (2)$$

Where:

n = number of data points in Monte Carlo simulation.

$tol_{1,X}$ = tolerance of the first joint position in the x-direction.

$tol_{1,Y}$ = tolerance of the first joint position in the y-direction.

After this step is complete, the distribution of the next joint connected to the link can be calculated using the distribution from the first joint.

$$X_{2,n} = rand(-tol_{2,X}, +tol_{2,X}) + \left|X_1 - X_2\right|_{nominal} \quad (3)$$

$$Y_{2,n} = rand(-tol_{2,Y}, +tol_{2,Y}) + \left|Y_1 - Y_2\right|_{nominal} \quad (4)$$

Where:

X_1 = nominal x-axis position of first joint

Y_1 = nominal y-axis position of first joint

X_2 = nominal x-axis position of second joint

Y_2 = nominal y-axis position of second joint

The process of calculating the joint position distribution for successive joints is continued on in this fashion for all joints of a rigid link in typical Monte Carlo fashion.

Once each joint distribution is calculated for each rigid link involved in the linkage, the joints connecting a base link to another link should be examined. The joint distribution of the connecting link should be considered as centered around each 'n' points in the base distribution and the connecting link distribution calculated from each 'n' points in the base distribution. In this way, the complete linkage joint distribution can be calculation.

CONCLUSION

It has been shown that the impact of a typical manufacturing tolerance to a large off-highway linkage is minute for positional accuracy of the linkage. A stochastic model of the linkage was developed and interrogated to validate this claim. While the positional variance due to the manufacturing tolerance of the linkage is small, it is only one contributor to the overall positional accuracy of the linkage.

Future work includes examining the affects of joint position changes not only from manufacturing tolerances, but wear in the pin to bore clearance as the machine is utilized.

REFERENCES

1. Heinz Pitsch and Sergei Fedotov, Stochastic modeling of scalar dissipation rate fluctuations in non-premixed turbulent combustion, Center for Turbulence Research Annual Research Briefs, pp. 91-103, 2000.
2. Jinling Wang, Hung Kyu, Lee Young, Jin Lee, Tajul Musa, Chris Rizos, Online Stochastic Modelling for Network-Based GPS Real-Time Kinematic Positioning, Journal of Global Positioning Systems Vol. 4, No. 1-2: 113-119, 2005.
3. P. Thoft-Christensen: Stochastic Modelling of the Crack Initiation Time for Reinforced Concrete Structures. ISSN 1395-7953 R0012. Structural Reliability Series Paper 191.
4. M. Levene, T. Fenner, G. Loizou, and R. Wheeldon. A stochastic model for the evolution of the web. Computer Networks, 39:277--287, 2002.
5. Jonathan W. Wittwer, Kenneth W. Chase, Larry L. Howell, The direct linearization method applied to position error in kinematic linkages, Mechanism and Machine Theory 39. 681–693, 2004.

CONTACT

Jahmy Hindman received his B.S. degree in mechanical engineering from Iowa State University in 1998. He obtained his M.Sc. degree from the University of Saskatchewan in 2002 where he is currently a PhD candidate. His interests lie in hydraulic system design and control as well as artificial intelligence. He is also the engineering supervisor of the hydraulics and electrical groups for the John Deere Four-Wheel-Drive Loader group. His email address is hindmanjahmy@johndeere.com

Richard Burton received his PhD and M.Sc. degrees from the University of Saskatchewan. He is a professor of Mechanical Engineering at the University of Saskatchewan, has professional engineering status (P.Eng) with the Association of Professional Engineers of Saskatchewan and is a Fellow of the ASME. Richard is involved with research pertaining to the control and monitoring of hydraulic systems, component design, and system analysis. His email address is richard.burton@usask.ca.

Greg Schoenau is a Professor of Mechanical Engineering at the University of Saskatchewan. He was the head of that department from 1993 to 1999 and is now the Associate Dean of Research (2006). He received his B.Sc. and M.Sc. degrees from the University of Saskatchewan in Mechanical Engineering in 1967 and 1969 respectively. In 1974 he obtained his PhD from the University of New Hampshire in fluid power control systems. He continues to be active in this area and in the thermal systems area as well. He has also held numerous positions in outside engineering and technical organizations. His email address is greg.schoenau@usask.ca.

CHAPTER 4

DYNAMIC PAYLOAD ESTIMATION IN A FOUR-WHEEL-DRIVE LOADER

Published as:

- Jahmy Hindman, Rich Burton, Greg Schoenau, "Dynamic Payload Estimation in a Four-Wheel-Drive Loader". 10th Scandinavian International Conference on Fluid Power, Tampere, Finland, May 21st 2007.

4.1 Objectives

This paper presents the first three algorithms developed to dynamically estimate the payload in a 4WD loader. This paper describes these three algorithms and their shortcomings and overall performance. The algorithms are discussed in some detail and experimental results are shown.

4.2 Methods and Results

The first method examined in the initial algorithm was to start with as simple an algorithm as possible in order to establish a baseline for both computational expediency and accuracy. The first algorithm implementation consisted of using the kinematic model developed in the previous publication and four measured parameters consisting of boom cylinder extension, bucket cylinder extension, boom cylinder head-side pressure and boom cylinder rod-side pressure. Utilising these four inputs and the kinematic model, a calculation for payload mass is possible. This effort produced an algorithm capable of 8% full scale accuracy. An algorithm with better accuracy was required.

The second method examined in algorithm development was to abandon the kinematic model and rely completely upon an ANN to characterize the linkage kinematics, machine dynamics and hydraulic system nonlinearities. This effort produced a 5x10x10x1 network trained on experimentally acquired data consisting of the same inputs used for the kinematic model and then subjected to new machine data sets. This algorithm produced significantly worse results than the direct kinematic model previously discussed.

The third method focused on trying to improve the fidelity of the ANN algorithm. Additional inputs

reducing the training set ambiguity and increasing the richness of the data set were necessary. The additional inputs used for this algorithm in addition to the four inputs previously discussed were a calculated boom cylinder velocity used to provide an input related to the viscous friction in the system as well as the kinematic model output (i.e. the kinematic model output was used as an input to the ANN). This network was trained on the same set of data used in the previous algorithm and then subjected to real-world machine use data. This algorithm succeeded in providing an algorithm capable of 3.5% full scale accuracy.

4.3 Conclusions

The results for this paper showed the progression of thought for the first three dynamic weighing algorithm approaches taken in the research work. These algorithms started with as simple an approach as possible and gradually added complexity in a successful attempt at improving the system accuracy. The final algorithm resulted in an overall accuracy of 3.5% full scale and a computational time of 126 milliseconds when run on an actual machine controller.

4.4 Contributions

There are two primary contributions in this publication. The first contribution is that this research shows for the first time an estimation algorithm that combines a first principles model (kinematic linkage model) and an ANN. This is a novel use of an ANN to capture and compensate for only those phenomenon that are not accounted for in the first principles model. This can be thought of as using the ANN as a compensating element between the complexities of the real physical system and the limited physics represented in the first principles model. The second contribution in this paper is less academic, but no less important. This contribution is a dynamic weighing algorithm that provides 3.5% full scale accuracy. Heretofore this accomplishment had not been made in either the published academic literature available to the author or in the industry.

PAYLOAD ESTIMATION IN FOUR WHEEL DRIVE LOADERS

Jahmy Hindman
Deere & Company
Construction and Forestry Division
P.O. Box 538
Dubuque, Iowa USA 52001
Phone 001 563 589 6194, Fax 001 563 589 5601
E-mail: hindmanjahmy@johndeere.com

Richard Burton & Greg Schoenau
University of Saskatchewan
Department of Mechanical Engineering
Saskatoon, SK Canada S7N 0J5
Phone 001 306 966 5373
E-mail richard.burton@usask.ca
greg.schoenau@usask.ca

ABSTRACT

Dynamic payload estimation in hydraulically actuated linkages is a diffucult task compounded by friction, compressibility, manufacturing variation, and linkage nonlinearity among other things. This problem is made even more difficult when the linkage is mobile as is often the case with off-highway equipment such as four-wheel-drive loaders, cranes, and excavators. The rigid body motion of this type of equipment affects the gravitational loads seen in the linkage and impact the payload estimate. The commercially available state-of-the-art load estimation solutions rely on the mobile machine becoming pseudo-static in order to maintain accuracy. This requirement increases the time required to move the material and decreases the productivity of the machine. A novel solution to this problem that enables the machine to remain dynamic and still accurately estimate the payload is discussed in this paper. Development and implementation on an actual four-wheel-drive loader is shown.

KEYWORDS: Dynamic, Load, Estimation, Mobile, Hydraulic, Neural Network

1. INTRODUCTION

2.1 Background

Hydraulic technology utilized in mobile hydraulic equipment such as four-wheel-drive loaders, excavators, cranes, forest machines, etc has advanced significantly in the last

twenty years. This time period has seen increased system efficiency from technology changes such as fixed versus variable displacement pumps, closed center versus open center control valves and hydrostatic transmissions among other technologies. All of these changes have been aimed at reducing the energy consumed by the equipment. The driving economic principle behind these advances stemmed from reducing energy consumed and thus reducing the amount of fuel consumed by the machine. The stark reality of the present day fossil fuel prices dictates that further technological advances be made in this area. Dynamic estimation of payload to reduce starts and stops in the work process is one such energy saving approach that is considered in this paper.

Towards that end, it is useful to briefly describe the current state of the art for mobile machine weighing systems. These systems are typically used to weigh the amount of material being moved from an excavation to a truck. The truck in turn will move the material to another location on the job-site or to another work area. It is critical to know the weight of material in the truck because of various governmental regulations in both Europe and North America that constrain the weight of hauling trucks to increase road surface durability. The state of the art for a typical four-wheel-drive loader weighing system today requires the loader to come to a pseudo-static equilibrium before an accurate measurement can be taken. This requires the machine operator to decelerate the machine and then accelerate again to reach the truck. This disruption in the work cycle uses energy that could be eliminated if an accurate dynamic measurement was possible. Additionally, if the weight of the material being carried by the mobile equipment is too much to complete a truckload under the legal limit, the operator must dispose of some material and re-weigh. The re-weigh operation, while possibly not being eliminated, should require the minimum amount of consumed energy in order to be fuel efficient. A typical truck loading cycle for a wheel loader may take 40 seconds with five seconds being used to weigh the material. If the five seconds could be eliminated, the cycle time could be improved by 12.5% and the energy consumed reduced.

The thrust of this research work was to develop a weighing system/algorithm that delivers current state-of-the-art accuracy (+/-1% full scale) while the machine is operating in its normal dynamic cycle. The economic motivation behind this work is reducing the energy consumed by the vehicle and thus reducing the input fuel costs. The machine chosen for this research work is a Deere 644J 4WD loader as shown in Figure 1.

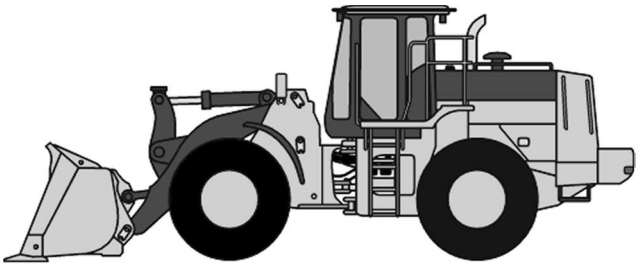


Figure 1: Deere 644J 4WD Loader

2.2 Literature Review

The topic of dynamic payload estimation and related work has created a small, but pertinent volume of research. Some of this research is involved with estimating kinematic parameters for linkages and actuators in preparation for developing a model-based payload estimation algorithm or for control purposes. Tafazoli et al [1] describe a method for estimating gravitational linkage parameters for an excavator and then use these estimated parameters in conjunction with load pins (i.e. instrumented joint pins capable of measuring joint forces) that the payload in the excavator bucket can be estimated to within 5% full scale. Similar work towards estimating gravitational and friction parameters in mobile linkages may also be found in [2][3].

Additionally, the topic of dynamic payload estimation has been researched significantly in applicable industries. Kyrtsos et al in [4] describe a method of estimating payload in a four-wheel-drive loader utilizing the lift cylinder pressures and linkage position. The method described relies on fitting a polynomial to the pressure information in order to smooth it out and provide a consistent payload estimate. This methodology is further refined in [5] and [6] with [6] describing the algorithm in entirety, presumably as implemented on wheel loaders described in the patent. The final algorithm adds correction factors for the velocity at which the lift was accomplished but is otherwise similar to the algorithm described in [4]. It should be noted that [6] claims that the algorithm can be utilized while the machine is moving, but this simply means while the lift is occurring, not rigid body motion of the entire vehicle.

2. METHODOLOGY

2.1 Description

A typical four-wheel-drive loader linkage can be seen in Figure 2. This linkage is referred to in the industry as a z-bar linkage due to the geometry of the bucket linkage (bucket cylinder, bell crank, and bucket link).

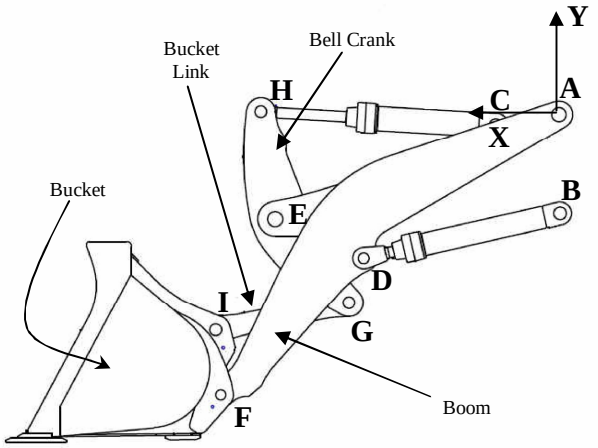


Figure 2: Four-Wheel-Drive Loader Linkage

While the research discussed here was performed specifically on the z-bar linkage, the methodology may be generalized for other hydraulically manipulated kinematic linkages.

In general, a payload weighing algorithm for a four-wheel-drive loader is focused on determining the weight of material that has been excavated and is resident in the bucket of the loader. This problem becomes difficult in practicality due to several influencing factors including volume and mass variations in material being moved, hydraulic cylinder and pin joint friction forces that vary with temperature and velocity, acceleration of the complete loader vehicle, acceleration of the loader linkage and inclination of the complete machine with respect to the gravitational force. These influencing factors make the problem a difficult one and nevertheless, the owner of a four-wheel-drive loader expects an error with the algorithm of less than +/-1% full scale. For the John Deere 644J loader shown in Figure 1, the full scale load is approximately 6,750 kg. 1% accuracy therefore means the scale needs to be within +/-67.5 kg. Additionally, the owner of a four-wheel-drive loader equipped with a scale today is required to bring the machine to a pseudo-static state during the boom lifting operation in order to achieve the accuracy described. This requirement interrupts the work cycle and introduces inefficiencies in time, fuel, and ultimately, money spent. It is this problem, estimating within 1% under dynamic conditions so as not to cause a disruption to the work cycle that this research is focused.

2.2 Algorithm Development

As a benchmark for algorithm development, a two-dimensional kinematic model of the linkage in Figure 2 was developed. This kinematic model was then utilized to calculate the linkage joint positions and linkage component (boom, bucket, bell crank, bucket link) center of gravity for any boom or bucket cylinder length. Using this information and neglecting friction terms, a simple summation of torques about joint "A" can be accomplished to solve for the remaining unknown of payload mass. This calculation assumes a known payload center of gravity which is assumed from knowledge of the bucket geometry. The resulting equation for calculating the payload weight is:

$$W_{Payload} = \frac{\left(\begin{array}{l} -M_{Boom} X_{CGBoom} - M_{BellCrank} X_{CGBellCrank} - M_{BucketLink} X_{CGBucketLink} \\ -M_{Bucket} X_{CGBucket} - \dfrac{\left(Fx_{BoomCyl} Y_D + Fy_{BoomCyl} X_D \right)}{9.8} \end{array} \right)}{X_{CGPayload}} \quad (1)$$

where:

$W_{Payload}$ = Weight of payload (kg)
M_{Boom} = Mass of boom (kg)
X_{CGBoom} = "X" coordinate of boom center of gravity (m)
$M_{BellCrank}$ = Mass of bell crank (kg)
$X_{CGBellCrank}$ = "X" coordinate of bell crank center of gravity (m)
$M_{BucketLink}$ = Mass of bucket link (kg)
$X_{CGBucketLink}$ = "X" coordinate of bucket link center of gravity (m)
M_{Bucket} = Mass of bucket (kg)
$X_{CGBucket}$ = "X" coordinate of bucket center of gravity (m)
$Fx_{BoomCyl}$ = "X" component of boom cylinder force (N)
Y_D = "Y" coordinate of joint "D" (m)
$Fy_{BoomCyl}$ = "Y" component of boom cylinder force (N)
X_D = "X" coordinate of joint "D" (m)
$X_{CGPayload}$ = "X" coordinate of payload center of gravity (m)

A set of data was collected from a John Deere 644J four-wheel-drive loader. This data included: bucket cylinder stroke, boom cylinder stroke, bucket cylinder rod-end pressure, bucket cylinder head-end pressure, boom cylinder rod-end pressure and boom cylinder head-end pressure. Data was collected on these four sensors at 40Hz sampling frequency using a 3hz low-pass filter. Utilizing equation [1] and the data set collected, the payload was estimated for a variety of machine conditions using the inputs of $Fx_{BoomCyl}$, $Y_{D,}$ $Fy_{BoomCyl}$, X_D. The mean error in the payload estimate is for these various machine conditions is seen in Figure 3.

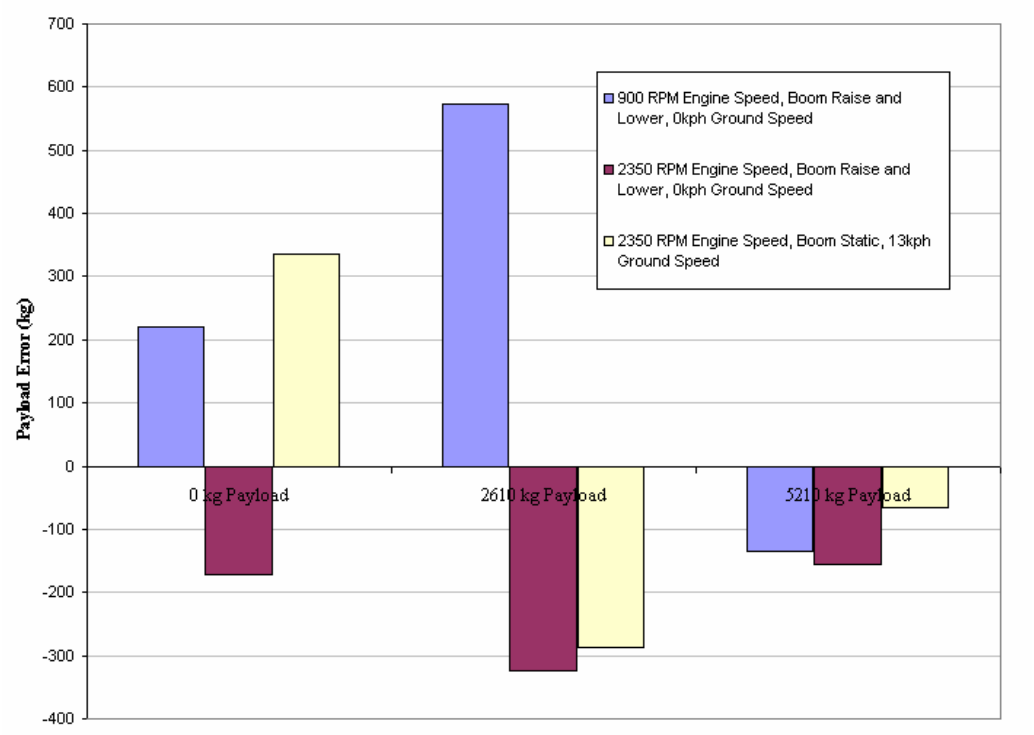


Figure 3: Payload Estimate Error from Kinematic Model

It can be seen in Figure 3 that the payload estimate is significantly higher than the desired +/-67.5 kg. Additionally, the variance in the estimate is high with a standard deviation of 306.8 kg. It is obvious from these results that this method falls short of meeting the requirements.

A second method was attempted. This method makes use of the data set collected for the kinematic model. Instead of using the boom and bucket cylinder lengths and pressures for the kinematic model, this data was utilized to train a feed-forward artificial neural network. A four-layer artificial neural network (ANN) with 5 neurons in the first layer, 10 in the second, 10 in the third and one in the output layer was trained with half of the collected data set. This training was accomplished with the Levenberg-Marquardt back-propogation algorithm [9] and implemented with Matlab®. The inputs to this network was the pressure differential across the boom cylinder (proportional to lift force) and both cylinder lengths for a total of three inputs. The network was trained for 25 epochs of the input training set until the mean squared error was less than .0005 kg. The

network was then subjected to a new set of input data obtained from the same wheel loader with 3436 kg of payload. The results of this effort are shown in Figure 4.

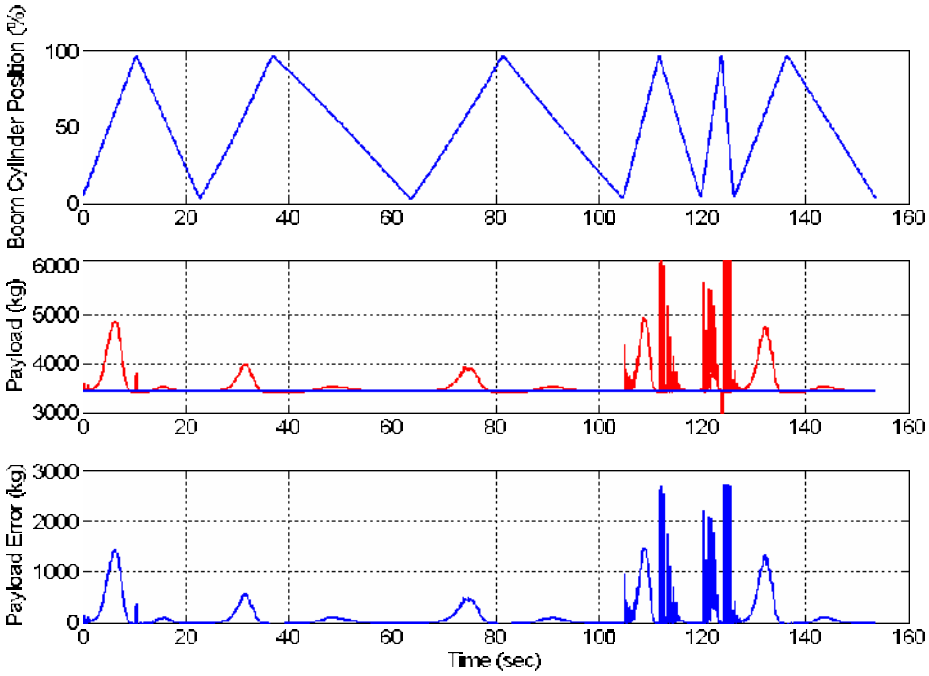


Figure 4: Neural Network Payload Estimate

In Figure 4, the data shown from time zero until 105 seconds consists of boom raise and lower events with the engine at 900 rpm and the machine in a static (not propelling) condition. From 105 seconds to 155 seconds, the machine is being driven at 15kph across rough ground while raising and lowering the boom. It can be seen that during the static boom raise and lower portion of the data, the payload estimate is well behaved with a mean error of 91.12 kg and a standard deviation of 236.3 kg. This performance is only slightly better than utilizing the kinematic model directly. In the dynamic portion of the data, the payload estimate is not well behaved with a mean error of 255.0 kg and a standard deviation of 552.7 kg. This is significantly worse performance than the kinematic model directly.

In an attempt to improve the fidelity of the payload estimate, some changes were made to the ANN. These changes consisted of adding a calculated boom velocity and the kinematic model output as inputs to the ANN. In total, the new topology consisted of five inputs: differential pressure across the boom cylinder, boom cylinder length, bucket cylinder length, boom cylinder velocity, and kinematic model payload estimate. The line of reasoning for attempting this network topology was first to improve the fidelity of the payload estimate with the additional inputs and secondly to determine if the network size could be reduced to reduce the computational burden of the algorithm. The same network utilized previously was trained with the data set containing the additional inputs of boom cylinder velocity and kinematic model payload estimate for 25 epochs of data. The results of this network when simulated using the same data obtained from the wheel loader with 3436 kg payload are shown in Figure 5.

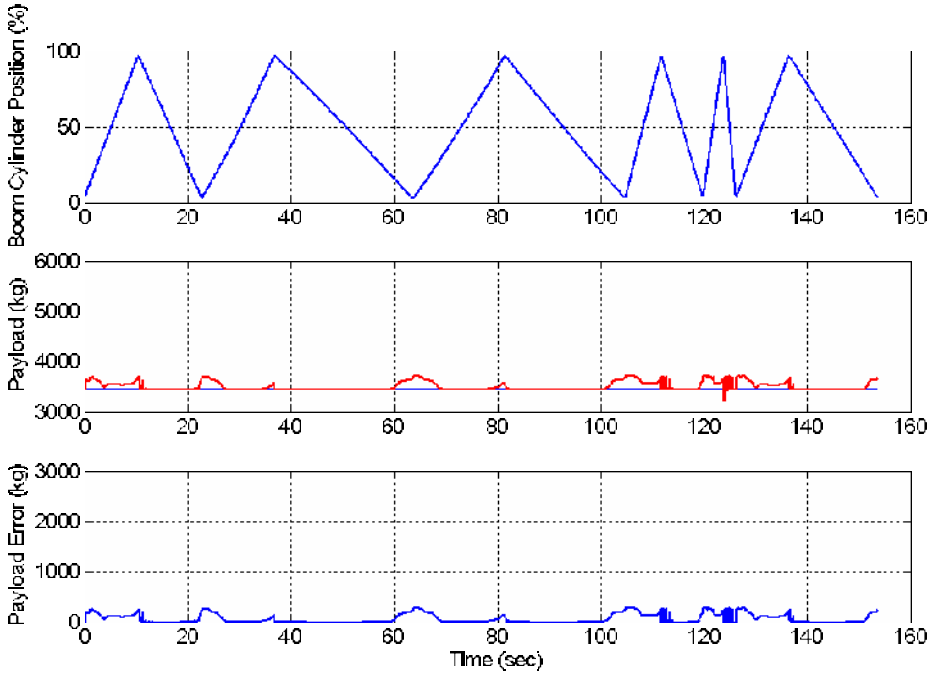


Figure 5: Neural Network Payload Estimate with Velocity and Kinematic Inputs

It can be seen visually, that the results of the ANN trained with the additional inputs is significantly better than the results in Figure 4 without the additional inputs. It can be seen that during the static boom raise and lower portion of the data, the payload estimate is well behaved with a mean error of 74.72 kg and a standard deviation of 77.4 kg. This performance is significantly better than utilizing the kinematic model directly or the ANN without the additional inputs. In the dynamic portion of the data, the payload estimate is nearly as well behaved with a mean error of 75.6 kg and a standard deviation of 80.7 kg. This is also significantly better performance than the kinematic model directly or the ANN without the additional inputs. The worst case error represented in Figure 5 for a 95% confidence interval is 75.6kg +/-158.7kg. More explicitly, the 95% confidence interval is -83kg to +234.3. This represents an accuracy of 3.47% full scale. This error is not less than the 1% full scale required by the industry and thus the algorithm as show, while presenting a significant improvement over the previous algorithms, would not be acceptable. Additional error will be seen as the application moves from being simulated in a floating point, 32-bit microprocessor to the more common fixed point, 16-bit processors of on-board mobile electronics. It is towards the topic of algorithm implementation that the next section is presented.

2.3 Algorithm Implementation

It is important to briefly discuss the computational overhead required to implement the algorithms described above. This discussion is important because any algorithm developed must not only provide 1% full scale accuracy, but also is required to run on the computational resources available on-board the mobile machine. In lieu of delving into the nuances of fixed versus floating point operations and 16 bit versus 32 bit

computations, a general metric of the time to perform the required computations when compiled from C code on a Pentium IV 2.0GHz microprocessor was used.

The kinematic model was developed through a straightforward kinematic chain analysis using a rectangular coordinate system and solving for the algebraic unknowns. This model required .002007 seconds of computation time for each sampled data point on the Pentium IV 2.0GHz processor. The 5x10x10x1 ANN with three inputs required .000006731 seconds of processor time and the 5x10x10x1 ANN with five inputs required .000008558 seconds of processor time. In the case of the ANN with five inputs however, since one of the inputs was the output of the kinematic model, the time required to run this network was the sum of both the kinematic model run time (.002007 seconds) and the five input ANN run time (.000008558 seconds) for a total of .002015558 seconds.

The C code was then compiled for the five input ANN with the kinematic model using a tasking C compiler and run on the 40MHz microprocessor utilized in the on-board microcontroller. The run time for this algorithm as implemented was .126 seconds. It should be noted that this experiment does not include any other computational overhead (i.e. other machine control) that needs to be accomplished at the same time as the calculation and thus would affect processor load. It can be said however that the minimum transport delay between measurement of required data and payload estimate output would be on the order to 126 milliseconds with the five input ANN and kinematic model algorithm. This delay is well within the more typical 500 millisecond update of critical machine performance operation (temperatures, speeds, etc) and deemed acceptable.

3. FUTURE WORK

Future work will focus on improving the payload estimate accuracy through improving the fidelity of the kinematic model with the addition of hydraulic cylinder and pin joint friction models. Additionally, improvements in filtering the pressure and position information will be pursued. Additional work may also focus in determining discrete operating points at which the operator is warned that the estimate may no longer be accurate to reduce the operational requirements of an implemented system. Additionally, the algorithm will continue to be implemented on the actual hardware utilized on the John Deere 644J four-wheel-drive loader to determine real-world accuracy and performance in a variety of conditions.

4. ACKNOWLEDGEMENTS

The authors would like to thank Deere & Company for providing the test vehicle and data with which to develop the algorithm. Specific appreciation is should be given to Cory Brant and Kevin Campbell for collecting the data used in this study.

5. REFERENCES

[1] Tafazoli S., Lawrence, P.D., Salcudean S.E., Chan D., Bachmann S., and de Silva C.W., *Parameter Estimation and Actuator Friction Analysis for a Mini Excavator*, Proceedings of IEEE International Conference on Robotics and Automation. pp. 329-334. April 1996.

[2] Zweiri Y.H., Seneviratne L.D., and Althoefer K. *Parameter Estimation for Excavator Arm Using Generalized Newton Method*, IEEE Transactions on Robotics. Vol. 20, No. 4. pp. 762-767. August 2004.

[3] Tafazoli S., Lawrence P., and Salcudean S., *Identification of Inertial and Friction Parameters for Excavator Arms*, IEEE Transactions on Robotics and Automation. Vol. 15, No. 5. pp. 966-971. October 1999.

[4] Kyrtsos C., and Worrell D., *Dynamic Payload Monitor*, United States Patent 4,919,222. April 24, 1990.

[5] Kyrtsos C., and Worrell D., *Dynamic Payload Monitor*, United States Patent 5,070,953. Dec 10, 1991.

[6] Kyrtsos C., *Dynamic Payload Monitor*, United States Patent 5,105,896. April 21, 1992.

[7] Kyrtsos C., and Gudat A., *Dynamic Payload Monitor*, United States Patent 5,182,712. January 26, 1993.

[8] Karumanchi A., *Dynamic Payload Monitor*, United States Patent 5,509,293. April 23, 1996.

[9] Hagan, M.T. & Menhaj, M. *Training Feed Forward Networks with the Marquardt Algorithm*. IEEE Transactions on Neural Networks. Vol. 5(6). pp. 989993.

CHAPTER 5

AN ARTIFICIAL NEURAL NETWORK APPROACH TO PAYLOAD ESTIMATION IN FOUR WHEEL DRIVE LOADERS

Published as:

- Jahmy Hindman, Rich Burton, Greg Schoenau, "An Artificial Neural Network Approach to Payload Estimation in Four Wheel Drive Loaders". ASME International Mechanical Engineering Congress and Exposition, Seattle, Washington, November 11th, 2007.

5.1 Objectives

The objective of this third publication was to improve the fidelity of the weighing algorithm to improve upon the 3.5% full scale accuracy provided in the algorithm developed in the previous publication. The algorithm development is shown with experimental results.

5.2 Methods

The method used in this publication was to further refine the final algorithm used in the previous publication. This was accomplished by adding a single additional input to the ANN. The new input utilised in the ANN was the 4WD loader fore-aft acceleration. This data was obtained by adding an accelerometer to the front frame of the 4WD loader and capturing this data along with the boom cylinder extension, bucket cylinder extension, boom head and rod pressures, calculated boom cylinder velocity and the kinematic model output.

5.3 Results

The algorithm developed in this publication through the use of the additional fore-aft acceleration information manages to improve the overall algorithm error to 1.96% full scale. This represents a 56% improvement over the previous algorithm. At the same time, the computational burden of the additional input on the algorithm did not appreciably increase the computational speed of the algorithm.

5.4 Contributions

The primary contribution of this publication was a dynamic weighing algorithm that pushed the envelope of accuracy previously considered state-of-the-art for this type of dynamic application. The final algorithm predicts the payload mass to within 2% accuracy with no limitations on the normal dynamics caused by machine usage. At the same time, the algorithm managed to be computationally expedient so that it might be implemented on the traditional on-board vehicle microcontrollers.

Proceedings of IMECE2007
I2007 ASME International Mechanical Engineering Congress and Exposition
November 11-15, 2007, Seattle, Washington, USA

IMECE2007-41618

AN ARTIFICIAL NEURAL NETWORK APPROACH TO PAYLOAD ESTIMATION IN FOUR WHEEL DRIVE LOADERS

Jahmy Hindman ,
Four Wheel Drive Loader Engineering, Deere & Co.

Dr. Richard Burton,
Mechanical Engineering,
University of Saskatchewan

Dr. Greg Schoenau,
Mechanical Engineering,
University of Saskatchwan

ABSTRACT

Estimation of the manipulated payload mass in off-highway machines is made non-trivial by the nonlinearities associated with the hydraulic systems used to actuate the linkage of the machine in addition to the nonlinearity of the kinematics of the linkage itself. Hydraulic cylinder friction, hydraulic conduit compressibility, linkage machining variation and linkage joint friction all make this a complex task under even ideal (machine static) conditions. This problem is made even more difficult when the linkage is mobile as is often the case with off-highway equipment such as four-wheel-drive loaders, cranes, and excavators. The rigid body motion of this type of equipment affects the gravitational loads seen in the linkage and impacts the payload estimate. The commercially available state-of-the-art load estimation solutions rely on the mobile machine becoming pseudo-static in order to maintain accuracy. This requirement increases the time required to move the material and decreases the productivity of the machine. An artificial neural network solution to this problem that enables the machine to remain dynamic and still accurately estimate the payload is discussed in this paper. Development and implementation on an actual four-wheel-drive loader is shown.

INTRODUCTION

A primary application for four-wheel-drive loaders is manipulating various forms of aggregate. This typically takes one of two task forms. The first is stockpiling where some amount of material is moved from one location in a quarry to another location. The second is truck loading where some amount of material is loaded by the four-wheel-drive loader into a truck used for on-highway transportation. In the truck loading application, the amount of material placed in the dump box of the on-highway truck is critical. The governing departments or ministries of transportation in the states or provinces where the on-highway trucks are traveling govern the load limit the trucks may carry legally without incurring financial penalty. These limits are enforced to increase longevity of the road surfaces. It is advantageous to the finances of the trucking operation to maximize the truck payload while not exceeding the weight limits for the truck. To this end, large weighing scales are utilized at the exit of the quarries to weigh all trucks leaving for on-highway use. If the trucks weigh over their maximum limit, they must return to the quarry floor where they must unload some of the aggregate before proceeding. If they unload too much weight, the loader must add aggregate back to the truck in an attempt to maximum the truck's payload. This is a tedious task that can take significant time which utilizes a significant amount of fuel which negatively affects the financial performance of the quarry. To solve this dilemma, on-board weighing scales started being developed nearly fifteen years ago. The scale mounted to the four-wheel-drive loader utilizes hydraulic pressure in the main lift cylinders to estimate the payload in the bucket. The four-wheel-drive loader scale then weighs the material in the four-wheel-drive loader bucket. When the four-wheel-drive loader dumps the bucket of material in the truck the mass of material in the truck is known. This enables the truck to weigh at the exit scale with confidence that the truck load is optimized to the legal limit.

The state of the art for four-wheel-drive loader scales is worth describing to understand why the work presented in the paper is of use. The current state of the art for a typical four-wheel drive loader scale typically requires the machine to come to some pseudo-static equilibrium before and accurate measurement can be taken. This requires the machine operator to decelerate the machine and then accelerate again to reach the truck. This disruption in the work cycle uses energy

that could be eliminated if an accurate dynamic measurement was possible. Additionally, if the weight of the material being carried by the mobile equipment is too much to complete a truckload under the legal limit, the operator must dispose of some material and re-weigh. The re-weigh operation, while possibly not being eliminated, should require the minimum amount of consumed energy in order to be fuel efficient. A typical truck loading cycle for a wheel loader may take 40 seconds with five seconds being used to weigh the material. If the five seconds could be eliminated, the cycle time could be improved by 12.5% and the energy consumed reduced.

The thrust of this research work was to develop a weighing system/algorithm that delivers current state-of-the-art accuracy (+/-1% full scale) while the machine is operating in its normal dynamic cycle. The economic motivation behind this work is reducing the energy consumed by the vehicle and thus reducing the input fuel costs. The machine chosen for this research work is a Deere 644J 4WD loader as shown in Figure 1.

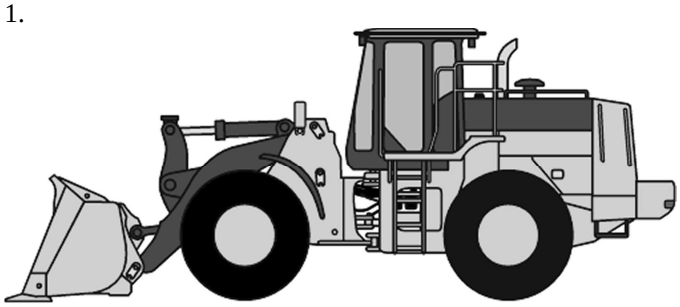


Figure 1: Deere 644J 4WD Loader

LITERATURE REVIEW

The topic of dynamic payload estimation and related work has created a small, but pertinent volume of research. Some of this research is involved with estimating kinematic parameters for linkages and actuators in preparation for developing a model-based payload estimation algorithm or for control purposes. Tafazoli et al [1] describe a method for estimating gravitational linkage parameters for an excavator and then use these estimated parameters in conjunction with load pins (i.e. instrumented joint pins capable of measuring joint forces) that the payload in the excavator bucket can be estimated to within 5% full scale. Similar work towards estimating gravitational and friction parameters in mobile linkages may also be found in [2][3].

Additionally, the topic of dynamic payload estimation has been researched significantly in applicable industries. Kyrtsos et al in [4] describe a method of estimating payload in a four-wheel-drive loader utilizing the lift cylinder pressures and linkage position. The method described relies on fitting a polynomial to the pressure information in order to smooth it out and provide a consistent payload estimate. This methodology is further refined in [5] and [6] with [6] describing the algorithm in entirety, presumably as implemented on wheel loaders described in the patent. The final algorithm adds correction factors for the velocity at which the lift was accomplished but is otherwise similar to the algorithm described in [4]. It should be noted that [6] claims that the algorithm can be utilized while the machine is moving, but this simply means while the lift is occurring, not rigid body motion of the entire vehicle.

The methodology developed in this paper is a continuation of the method developed in [7]. In [7], the methodology relied on an artificial neural network (ANN) trained using differential pressure across the boom cylinder, boom cylinder stroke, bucket cylinder stroke, boom cylinder velocity, and a kinematic model payload estimate. This algorithm was effective but produced an accuracy of 3.47% full scale which does not meet the requirement in the industry of +/- 1% full scale.

ALGORITHM

The original algorithm mentioned in [7], utilized a set of data that did not contain any information regardin ghte rigid body motions of the machine. This is a significant limitation of that methodology since the four-wheel-drive loader typically undergoes significant amounts of acceleration and deceleration in a typical work cycle as seen in Figure 2.

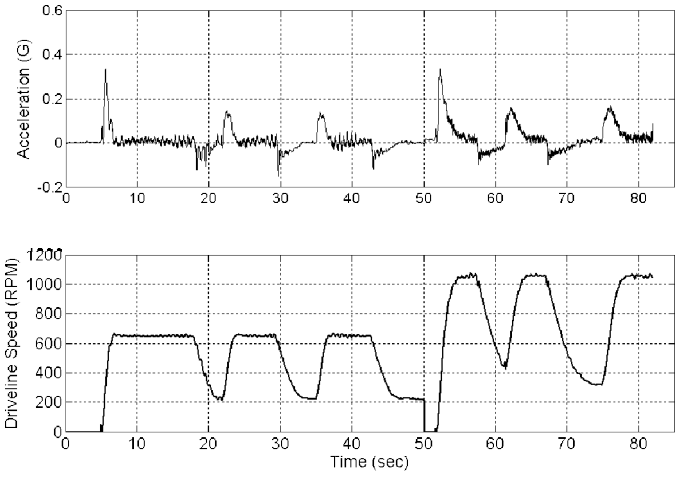


Figure 2: Acceleration During Transport

The acceleration and deceleration of the vehicle has the affect of accelerating and decelerating the payload in the four-wheel-drive loader bucket. This in turn has the affect of increasing and decreasing the boom cylinder pressure differential. This can bee seen in the boom pressure differential data shown in Figure 2 for a four-wheel-drive loader transporting across a typical quarry floor.

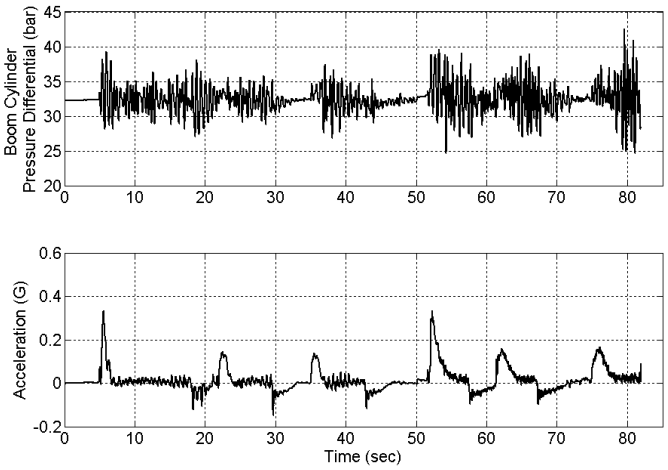


Figure 3: Boom Pressure Differential During Transport

Because of this limitation, the algorithm in [7] was modified to incorporate acceleration data. The accelerometer was mounted in the front frame that the loader linkage seen in Figure 3 attaches to. This accelerometer data was collected in both the X and Y direction as defined in Figure 4.

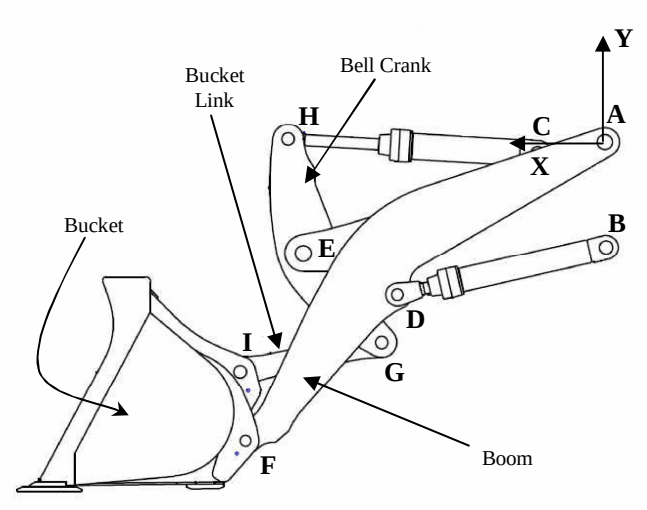


Figure 4: Four-Wheel-Drive Loader Linkage

A data set was collected from an actual John Deere 644J four-wheel-drive loader. This data set included differential pressure across the boom cylinder, boom cylinder stroke, bucket cylinder stroke, and X-direction acceleration. The data was then utilized with the kinematic model developed in [7] to calculate the payload estimate from the collected data using the kinematic model. This calculated estimate was then utilized along with the differential pressure across the boom cylinder, boom cylinder stroke, bucket cylinder stroke, and X-direction acceleration for training a 5x10x10x1 ANN. Only half of the data set was used for training while the other half was held for later testing of the accuracy of the algorithm. The ANN topology and input/output relationship is seen in Figure 4 (no network weights or biases are shown for sake of clarity.

The feed forward network in Figure 5 was trained using the Levenberg-Marquardt training algorithm. The data used for training consisted of two separate runs for three different payloads. One of the two runs was collected while driving the four-wheel-drive loader across a quarry floor while not manipulating the boom or bucket. The second run was collected while the boom was manipulated up and down. The three different payloads were 0 kg, 3945 kg, and 5563 kg.

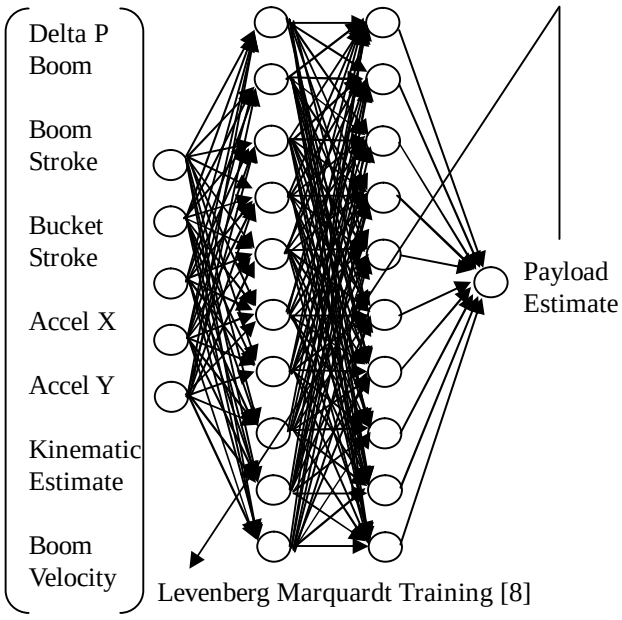


Figure 5: ANN topology

The network shown in Figure 5 was trained using the Levenberg-Marquardt algorithm on half of the data collected off the 644J four-wheel-drive loader. The results of this are shown in Figure 6.

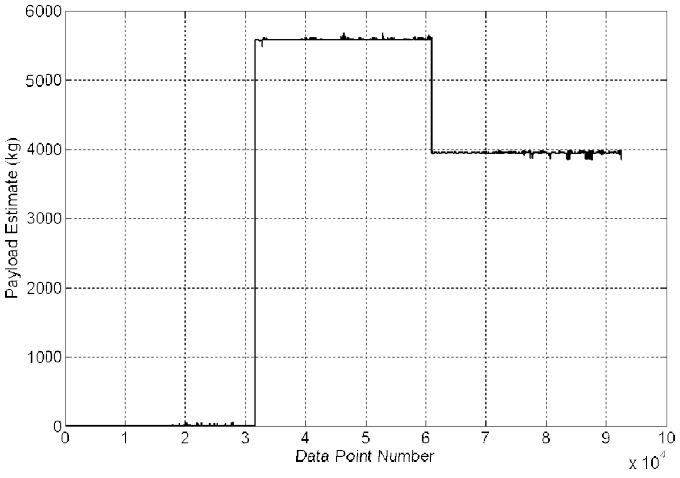


Figure 6: ANN Training Results

The numerical results of the training shown in Figure 6 give a payload estimate of .879 +/-94.92 kg for the zero load case (i.e. 0 kg payload), 5581.6 +/- 24.54 kg for the full load case (i.e. 5563 kg payload) and 3947.6 +/- 41.49 kg for the half load case (i.e. 3945 kg payload) at a 95% confidence interval. This gives a maximum error in training of 95.8 kg with a 95% confidence interval. This error should be compared to the maximum allowable of 1% full scale giving +/-55.81 kg. It can be seen that this algorithm does not meet the maximum allowable error at the zero load case, but does at the other two load cases for the training results.

The network shown in Figure 5 was then exercise by utilizing the other half of the data collected from the 644J four-wheel-drive loader that the network was not trained with. This simulation revealed that the network, when subjected to new data, performed slightly worse than on the data it had been trained with. The results from this exercise showed that the payload estimate at the zero load case was 1.3 +/- 109.3 kg, 5588.2 +/- 33.9 kg for the full load case and 3946.7 +/- 38.1 kg for the half load case. As in the training data, the worst case error occurred at the zero load condition giving an error of 110.6 kg for a 95% confidence interval. This should be compared to the maximum allowable error of 55.81 kg. The maximum error seen with this algorithm is nearly twice of the acceptable error. It is interesting to note that the deviation of the estimate decreases as the payload weight increases in general. It is believed this is due to the reduced impact of rigid body accelerations on the boom pressures relative to the impact caused by the actual payload.

It has been shown that including the acceleration information in the ANN algorithm, the worst case payload estimate is +/-1.96% full scale error. While this is not acceptable for the industry, it does represent a significant improvement over the +/-3.47% shown in [7].

FUTURE WORK

Because the ANN approach to payload estimation shown in this paper shows promise in being able to produce an accurate payload estimate under truly dynamic machine operating conditions further refinement of the algorithm will continue. Towards this end, efforts will be made to improve the algorithm by improving fidelity of the pre-filtering the training data undergoes before being used by the ANN. Additionally, size and topology of the ANN will be investigated to determine minimum size of the network required for accurate results.

The most important part of the future work in developing this algorithm will be to determine additional inputs to improve the fidelity of the results of the algorithm. Due to the computational overhead of the kinematic model calculation, effort will be given towards investigating new inputs such as driveline speed and engine speed in conjunction with the acceleration data in an attempt to remove the kinematic model data and calculation from the network inputs altogether.

NOMENCLATURE

ANN: Artificial Neural Network

ACKNOWLEDGMENTS

The authors would like to thank Deere & Co for providing the 644J four-wheel-drive loader and operating environment required for obtaining the data sets used in the paper. Additionally, the authors thank Leonard Mast for instrumenting, operating and obtaining the data.

REFERENCES

[1] Tafazoli S., Lawrence, P.D., Salcudean S.E., Chan D., Bachmann S., and de Silva C.W., *Parameter Estimation and Actuator Friction Analysis for a Mini Excavator*, Proceedings of IEEE International Conference on Robotics and Automation. pp. 329-334. April 1996.

[2] Zweiri Y.H., Seneviratne L.D., and Althoefer K. *Parameter Estimation for Excavator Arm Using Generalized Newton Method*, IEEE Transactions on Robotics. Vol. 20, No. 4. pp. 762-767. August 2004.

[3] Tafazoli S., Lawrence P., and Salcudean S., *Identification of Inertial and Friction Parameters for Excavator Arms*, IEEE Transactions on Robotics and Automation. Vol. 15, No. 5. pp. 966-971. October 1999.

[4] Kyrtsos C., and Worrell D., *Dynamic Payload Monitor*, United States Patent 4,919,222. April 24, 1990.

[5] Kyrtsos C., and Worrell D., *Dynamic Payload Monitor*, United States Patent 5,070,953. Dec 10, 1991.

[6] Kyrtsos C., *Dynamic Payload Monitor*, United States Patent 5,105,896. April 21, 1992.

[7] Hindman J., Burton R., and Schoenau G., Payload Estimation in Four Wheel Drive Loaders, Transactions of the Tenth Scandinavian International Conference on Fluid Power. May 21, 2007.

[8] Hagan, M.T. & Menhaj, M. *Training Feed Forward Networks with the Marquardt Algorithm*. IEEE Transactions on Neural Networks. Vol. 5(6). pp. 989993.

CHAPTER 6

ON AMBIGUITY IN TRAINING DATA OF ARTIFICIAL NEURAL NETWORKS

Published as:

- Jahmy Hindman, Rich Burton, Greg Schoenau, "On Ambiguity in Training Data of Artificial Neural Networks". Submitted IEEE. 2007.

6.1 Objectives

The purpose of this paper was to study the impact that ambiguous input data sets has on the training and generalization of feed-forward ANN.

6.2 Approaches

The approach taken in this paper was to define a well-known problem and train an ANN to solve that problem with a training data set that contained no ambiguous inputs. The input data set was then purposely made ambiguous and the network was retrained with this new, and ambiguous, data set. The results of both networks when both subjected to a new input data set was compared to determine the affect of the ambiguity on network training.

6.3 Results

The results of this paper show that ambiguous training data has a very negative affect on accuracy and generalization of feed-forward ANN. A methodology to examine data sets for non-unique pairings is then proposed and generalized to n dimensioned ANN.

6.4 Contributions

The contributions of this paper is simply to highlight and provide a solution for one of the common issues when using ANN with experimentally acquired data. The issues surrounding experimentally acquired data

such as noise, sensor drift, etc can easily cause ambiguous data sets to occur if care is not taken. A method is developed to examine data sets for this condition is provided.

On Ambiguity in Artificial Neural Network Training Data

Jahmy J. Hindman
John Deere Construction & Forestry Division
HindmanJahmy@JohnDeere.com

Richard Burton & Greg Schoenau
University of Saskatchewan, Mechanical Engineering Department

Abstract

Feed-forward artificial neural networks (ANN) have been used in a broad range of research and development for much of the last twenty years. Due to their unique abilities in mapping common nonlinear relationships, and their relative ease of implementation, they are sometimes applied without proper scrutiny of the input training data sets. This paper describes a simple and tractable method for examination of the training data to ensure that all input/output pairings used for training are unique.

Introduction

Non-uniqueness of the input/output data pairings used to train a feed-forward ANN can cause significant error in network training. If the existence of non-unique input/output data pairs is not known, the error in training may be incorrectly attributed to network size, network topology or training algorithm. The existence of non-unique input/output pairs may not be a highly probable issue when using well controlled simulated data sets [1], but may be evident when using empirically derived data sets containing measurement and process noise.

This issue has been investigated previously by Bullinaria [2], [3], and a method for resolving the non-unique training data was presented. Bullinaria's method examines the ANN output training error and modifies the non-unique training data by choosing a new training output that gives the lowest output error based upon prior probability that the output is the correct one. Unfortunately, this method requires that the parameters be defined to determine what constitutes non-unique data. This does not guarantee that all non-unique data is resolved. An alternate approach is given by Weingend et al. [4] that consists of simply stopping the training when the output error begins to increase. Unfortunately, this leads to networks that may not be completely trained since the output error may still be quite large at this point. Another treatment given by Trappenberg et al. [5], [6] is to develop a network performance measure by which areas of the ANN output that are significantly negatively affected by the non-unique data are simply not used. Non-uniqueness of training data is so problematic for ANNs that other methods of classification have been examined that are not as sensitive to this problem as discussed by Hashemi et al. [7] and the use of support vector machines (SVM) for classification when non-unique training data is present.

In order to illustrate the problem encountered with non-unique data pairs, consider the problem of training a feed-forward ANN to provide the product ($\boldsymbol{Y}$) of two real numbers ($\boldsymbol{X_1}$, $\boldsymbol{X_2}$) between zero and ten. The network will have two inputs that define the two numbers to be multiplied, and a single output which is the product of the two inputs. An input training vector of all possible combinations of two integers from zero to ten is created. An output vector of the corresponding product of the two integers in the input training vector is also produced. The input and output vectors contain 121 unique pairs. A feed-forward ANN with size and topology shown in Figure 1 was then initialized and trained using 200 epochs of the training data and the Levenberg-Marquardt back-propagation method [8]. The neural network was implemented using Matlab®.

The trained network was then subjected to a testing set of all combinations of input from zero to ten in .1 increments. The network output and absolute value of the error surfaces for this validation can be seen in Figure 2. These graphical results show that the network was highly capable of mapping the product of two numbers between zero and ten. Since the input training data set consisted of all combinations of integers between zero and ten and the testing data was all combinations of numbers in this same range in .1 increments this network shows the ability to generalize well on data not contained in the original training data set. The output surface also exhibits a continuous behavior conducive to good generalization. The mean error calculated from the error surface is .047.

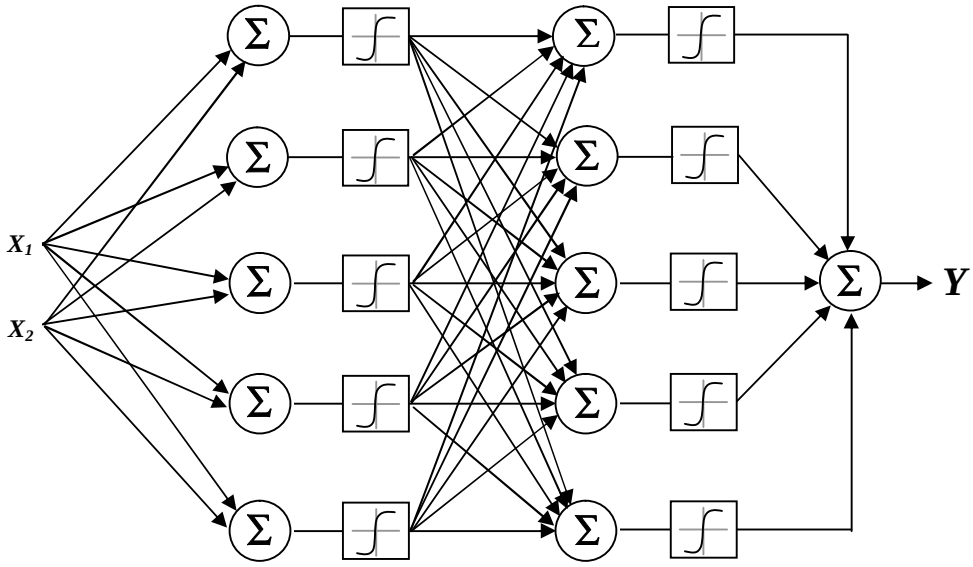


Figure 1: Network Topology

A single input/output pair was then added to the training data set. The inputs for the added pair were (5,5) and the output was wrongly prescribed a value of (50). This single added pair forces a non-unique condition in the training data for inputs of (5,5). These two inputs have two distinct and different outputs (25) and (50) in the training data. This makes the training data set consist of 120 unique input/output pairs and 2 non-unique pairs. A network of the same size and topology as that seen in Figure 1 was then initialized with the same initialization weights used when training the network

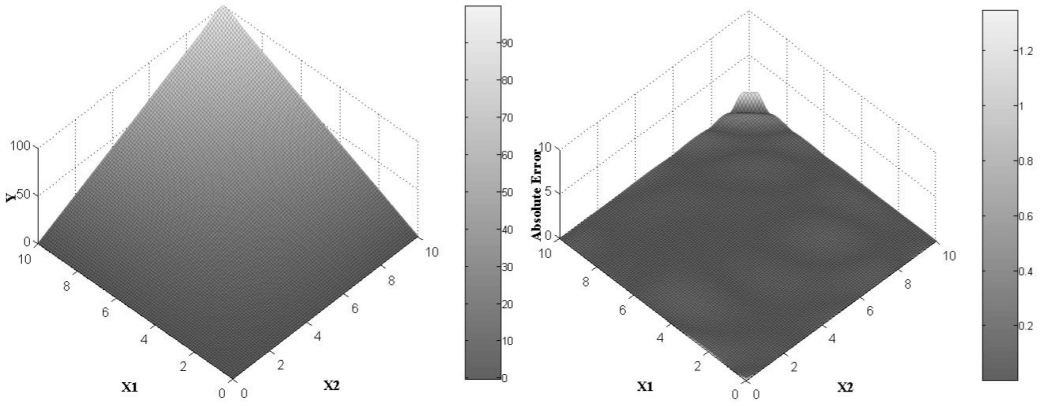


Figure 2: Unique Input Set Output and Error Surfaces

trained with the unique training data. This new network was also trained for 200 epochs of the training data, including the non-unique pairs, using the Levenberg-Marquardt back-propagation algorithm. The trained network was then subjected to a testing set of all combinations of input from zero to ten in .1 increments as in the previous example. The network output and absolute value of the error surfaces for this validation can be seen in Figure 3. It can be seen graphically that this network exhibits a "wrinkle" in the output surface passing through the non-unique point (5,5). The absolute error surface exhibits a peak centered at (5,5) as well as displaying generally increased error when compared to the error seen in Figure 2. The mean error calculated from this error surface is .7138, numerically showing the increased error in this network. The mean error associated with any product of five and of a number between zero and ten is 1.535 showing that the network error is significantly increased for inputs that

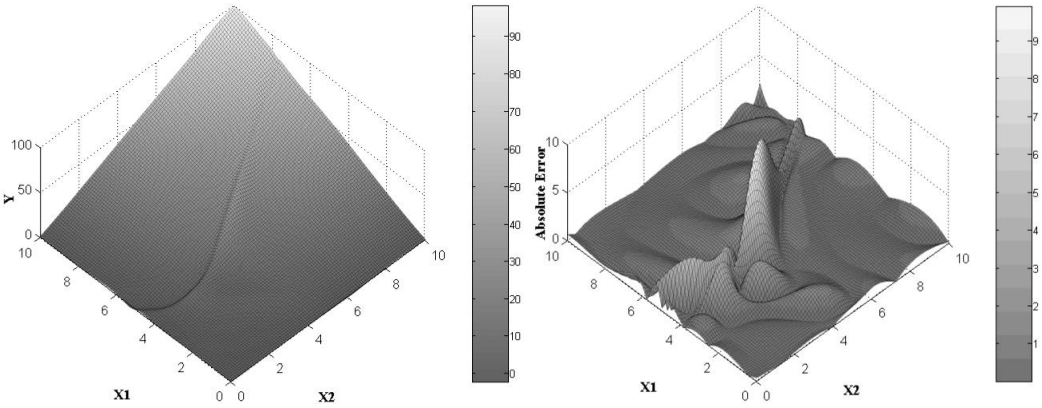


Figure 3: Non-Unique Input Set Output and Error Surfaces

include the ambiguous values. If the error for all data obtained from the product of five and any other number between zero and ten is neglected, the mean error for the remaining surface is .700 showing that the mean error for the unique data is also affected by the non-unique pair.

This example effectively shows the negative effects of training a feed-forward ANN with data that contains non-unique pairings. Intuitively, since the training data is ambiguous at a certain point, the network cannot be expected to produce the correct mapping for the ambiguous inputs. It is also intuitive that the network's capability to generalize would be compromised by this ambiguous training set. *This behavior also raises significant questions about the training of biological neural networks and the effects on human behaviors if the biological networks are subjected to non-unique training data. One possible conclusion is that ambiguous training data gleaned from sensory inputs leads to an elevated state of error and confusion in the biological neural network behavior.*

The probability of non-unique training data in a simulated research environment is limited in that the training data may be generated to intrinsically remove this possibility (i.e. the product of all integers from zero to ten may be easily verified for non-unique pairings). The problem does present itself when dealing with empirical data taken from noisy and/or faulty sensor measurements. It is also under these "real-world" conditions, that the possibility for real damage to be incurred from the output of the ANN exists (i.e. if the ANN is responsible for control of some device). It is necessary then to examine the training data and detect these ambiguous pairings and take one of two steps to mitigate their effects:

1. If the ambiguous data is warranted, additional inputs must be found to remove the ambiguity. For example, if the same inputs can indeed produce two different outputs, an additional input must be found that correlates to the change in the output to eliminate the ambiguity.

2. If the ambiguous data is not warranted, the ambiguity must be resolved by filtering out the errant data or removing all sets of ambiguous data and verifying the network generalizes well in this region of input.

The issue of un-warranted ambiguous data is relatively straightforward in its solution since the data may be exhaustively examined numerically for any ambiguous data. This may be computationally burdensome, but will result in the detection of un-warranted ambiguous data. The issue of warranted ambiguous data is different however since the data set is not incorrect, but insufficient. It is important to examine the training data set to ensure all ambiguities, whether warranted or not are resolved. The following discussion provides a method for detecting the sufficiency of the training data set.

Methodology

Consider the training data characterized by the surfaces seen in Figure 4. This data was measured from an actual engineering problem concerned with monitoring the condition of a closed center hydraulic valve and actuator [9]. In this problem, the pressure drop in both actuator chambers (rod and head) is monitored for sixty seconds and inference on the condition of the valve and actuator radial clearance is made from the amount of pressure drop measured in the actuator chambers.

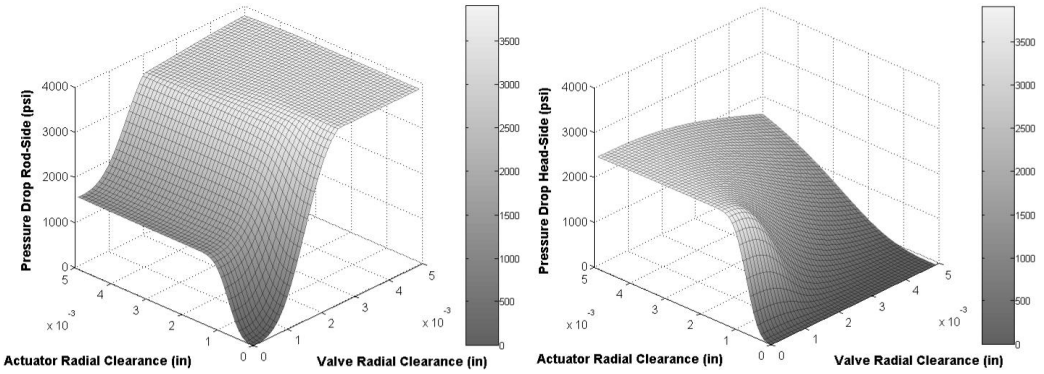


Figure 4: ANN Training Data

This data contains two regions of ambiguity where the inputs are constant while the outputs are changing. The two areas of ambiguity can be seen in Figure 5 where the two surfaces are plotted on the same axes.

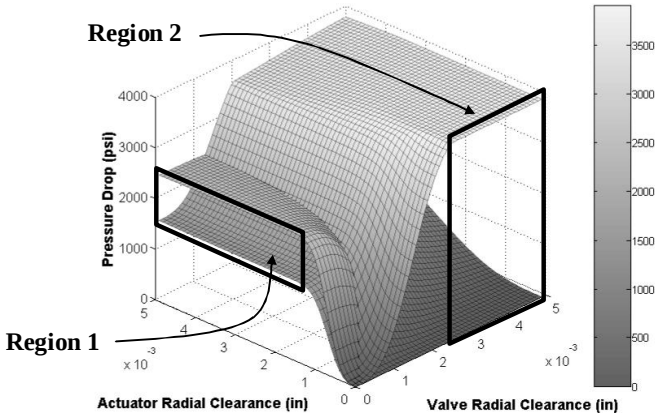


Figure 5: Ambiguous Regions

If a network were to be trained to map this input/output data, the mapping would result in significant error in these ambiguous regions and sub-optimal network performance elsewhere. It should be noted that this input/output data is not erroneous. In other words, there is nothing wrong with the input/output data other than it is not sufficient to absolve the ambiguities that arise. This is a critical problem however, and one that is often faced when investigating real-world problems where sensor limitations are enforced due to cost and availability issues.

The ambiguous regions may be examined numerically by computing the gradient of each input surface according to equation (1) and examining the difference in the resulting gradients (i.e. the gradient of the gradient).

$$\Delta_{SURF_{n,m}} = \frac{Z_{n+1} - Z_n}{X_{n+1} - X_n}\hat{i} + \frac{Z_{m+1} - Z_m}{Y_{m+1} - Y_m}\hat{j} \tag{1}$$

where: n = grid point number in x-direction
m = grid point number in y-direction
X: x-axis = Output 2 axis
Y: y-axis = Output 1 axis
Z : z-axis = Input 1 and 2 axes

The difference between the two gradients (i.e. between n and n+1, m and m+1) can be calculated by subtracting one gradient surface from the other according to equation (2).

$$\Delta\Delta_{SURF_1-SURF_2} = \Delta_{SURF_1} - \Delta_{SURF_2} \tag{2}$$

This provides the change in gradient between the two surfaces and is a set of vectors composed of $\hat{\boldsymbol{\imath}}$ and $\hat{\boldsymbol{\jmath}}$ components. There is one vector for every grid point to (n-1), (m-1) since the gradient is computed with a single forward difference method. The magnitude of these vectors was used to plot the surface seen in Figure 6.

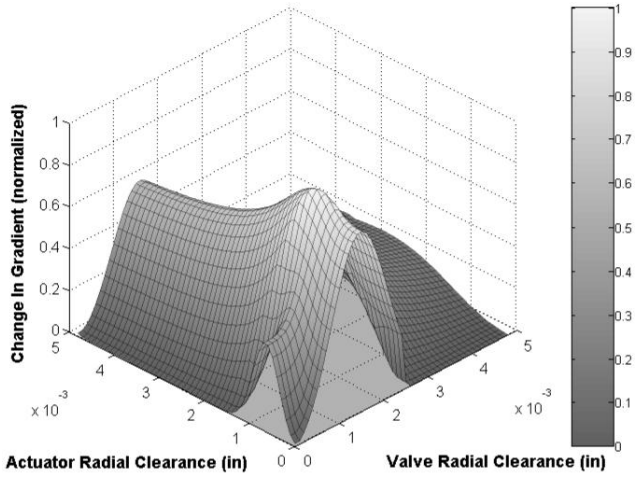


Figure 6: Change in Gradient

Any grid point that has a Z-value of zero on the surface in Figure 6 denotes a location where the input data is not sufficient to eliminate ambiguity in the training data. As can be seen from inspection, this occurs in the two regions marked in Figure 5. The surface in Figure 6 may be exhaustively investigated numerically to determine the degree to which ambiguous data may be a problem as well as the locations of this ambiguous data. Since the ambiguous training data is not incorrect, but is insufficient, additional input data is necessary to resolve the ambiguity. In the case of this actual problem, an additional data set was obtained to solve the ambiguity. The new data set is the amount of time it takes to achieve the maximum pressure drop on the rod-side of the actuator. This new data set can be seen in Figure 7.

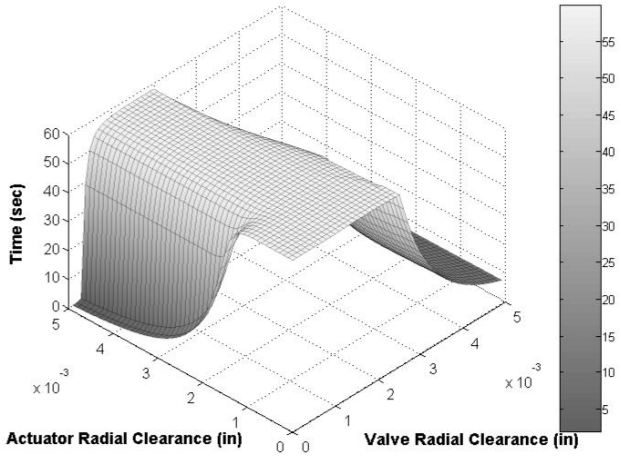


Figure 7: Additional Input

It can be verified that this surface resolves the ambiguity by calculating the gradient according to equation 1 and calculating the point-wise change in gradient among all three gradient surfaces. The

resulting surface derived from carrying out this operation is provided in Figure 8. It can be seen visually and analytically that the surface is now non-zero in all locations. A visual inspection of the new data surface in Figure 7 compared to the existing data in Figure 5 also provides some subjective evaluation of the validity of the solution.

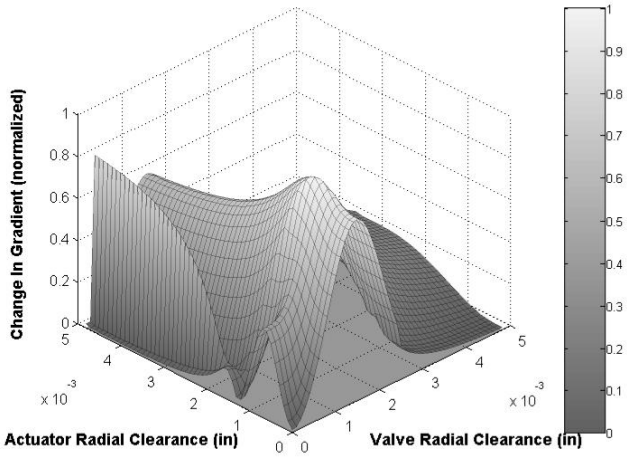


Figure 8: Change in Gradient with New Data

Conclusions

It is paramount to the success of training an ANN that the input/output training set be well conditioned without ambiguous mappings. The presence of ambiguous training data has a drastic negative consequence on the accuracy of the network. If the data set is not examined for ambiguities prior to training, and ambiguities exist, the error seen in the training may incorrectly be attributed to network topology, training limitations (i.e. local minima using gradient descent), or activation function choice. A fruitless endeavor to change the topology, training, or activation functions may ensue. It is prudent to suggest that the input/output training data be scrutinized prior to training to ensure that ambiguous pairings are addressed. Observation of the change in gradient between the network inputs and outputs as described above may be generalized for networks with more or less inputs and outputs than shown here. It has proven to be a useful algorithm to evaluate the condition of the training data prior to training.

References

[1] Ventura, D., Andersen, T., & Martinez, T., *Using Evolutionary Computation to Generate Training Set Data for Neural Networ*ks. Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms, pp. 468-71. 1995.

[2] Bullinaria, J.A., *Neural Network Learning from Ambiguous Training Data*, Connection Science, **7**, 99-122. 1995.

[3] Bullinaria, J.A., *Noise Reduction by Multi-Target Learning*, Proceedings of the European Symposium on Artificial Neural Networks, pp. 193-198. Brussels: D facto. 1994.

[4] Weingend, A.S., Huberman, B.A. & Rumelhart, D.E., *Predicting the Future: A Connectionist Approach.* International Journal of Neural Systems, Vol. 1, pp. 193. 1990.

[5] Trappenberg, T.P., Back, A.D. & Amari, S. I., *A Performance Measure for Classification with Ambiguous Data*. RIKEN Brain Science Institute Technical Report No.99-67. May 1999.

[6] Trappenberg, T.P. & Back, A.D. *A Classification Scheme for Applications with Ambiguous Data.* International Joined Conference on Neural Networks IJCNN 2000.

[7] Hashemi S. & Trappenberg T.P. *Using SVM for Classification in Datasets with Ambiguous Data.* World Multiconference on Systemics, Cybernetics and Informatics (SCI) 2002.

[8] Hagan, M.T. & Menhaj, M. *Training Feed Forward Networks with the Marquardt Algorithm.* IEEE Transactions on Neural Networks. Vol. 5(6). pp. 989-993.

[9] Hindman, J. *Condition Monitoring of Valves and Actuators in a Mobile Hydraulic System Using ANN and Expert Data*. M.Sc. Thesis, University of Saskatchewan, 2001.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

The objective of this research work as stated in Chapter 1 was to develop a payload weighing algorithm that was capable of estimating the payload in an off-highway machine to a high degree of accuracy while not restricting the motion of the machine to some pre-defined operating point. This objective was obtained over the course of this research program by incrementally improving a dynamic model of the off-highway machine. The final algorithm takes the form of a kinematic linkage model that is refined and corrected through the use of a feed-forward artificial neural network. The artificial neural network is also used as a calibration mechanism for the payload estimation. The major contributions of this research are:

- Summary of related research efforts available in academic literature, OEM publications, and granted patents.
- Development of a stochastic linkage model studying the effects of manufacturing tolerances on linkage kinematics showing minimal effects on the linkage studied.
- Development of a kinematic model for a z-bar linkage used in a 4WD loader.
- Development and training of an artificial neural network model used to characterize the kinematics of a z-bar linkage used in a 4WD loader.
- Development of a payload weighing algorithm for a 4WD loader that:
  - Utilises a minimal number of publicly available sensors.
  - Achieves a computational efficiency allowing the algorithm to run on a 40MHz 16bit microprocessor.
  - Utilises in a novel way an ANN to correct for the error in a first-principles model creating a hybrid model of the system being investigated.
  - Achieves a high degree of accuracy while allowing the machine to remain fully dynamic.
- Implementation of the developed algorithm by converting from the dynamic model to a tasking C executable installed and run on production-intent hardware.
- Verification through experimental studies of the developed payload weighing algorithm in various operating conditions.

- Investigation into the effects of ambiguous training data on ANN and development of a detection algorithm for such data.

Several comments should be made at the end of this research regarding recommendations moving forward. First, the primary take-away of this research should not be that a successful dynamic payload weighing algorithm has been developed. While that was certainly the goal established by this research at the outset, it has turned out to be only a by-product of a much larger discovery. This discovery is that, while first-principle models and ANN models each have their unique downfalls separately, when properly assembled together, they can form a hybrid model that is significantly better performing than either method on it's own. Where the first principles model typically requires extraordinary detail (and thus time) to model a complex nonlinear system well, the ANN typically requires many layers and nodes and long training times. When these two methods are coupled together however, they form a symbiotic relationship. The first principles model can be simplified to account only for part of the plant model's behaviour while the ANN can be reduced in size to only fill in the detail that the simplified first principles model is missing. This is a novel idea and one that bears further investigation. It is a recommendation then of this work that further complex nonlinear plants be modelled in this manner to determine if the results of this effort are unique to this application or if there is a broader application. A secondary recommendation is more specific to the algorithm defined in this work. This recommendation is that further inputs be determined in order to continue to improve the accuracy of the algorithm. In addition, further functionality should be added to the algorithm to detect when the system is no longer calibrated and warn the operator that a calibration is required. This would minimize the effects of weather and system wear on the output. A final recommendation is made to expand this general algorithm to other off-highway machine platforms such as excavators and articulated dump trucks.

It should be mentioned that the data collected for this study was collected in two different ways in order to validate that the data was correct. The first method utilised was to log the Controller Area Network (CAN) bus data received from the microcontroller for the sensor data required for the algorithm. The second method was to utilise a data acquisition system to log the analog sensor data directly. These two data sets were compared to each other to ensure the validity of the CAN bus data which was utilised throughout the rest of the research. An example of the data collect can be seen in Appendix E.

It is worth explicitly stating the limitations of the current algorithm here so that future work based upon this research is clear with respect to fruitful areas of investigation. The final algorithm presented in this work is limited to have a deterministic amount of material in the bucket. The bucket must also be in the completely rolled-back position and the center of gravity of the load must be similar to the geometric center of the bucket geometry. In addition, the hydraulic oil temperature must be above 45 degrees centrigrade in order to minimize the viscous friction effects of cold oil. Future work could focus on eliminating these areas of poor estimation performance by use of additional sensor information. It is also believed that the accuracy of the overall algorithm could be improved significantly by implementing the algorithm on a 32bit microprocessor with a floating point unit (FPU) that would remove the fixed point arithmetic restriction that

is placed upon the current implementation.

REFERENCES

[1] U.S. Environemental Protection Agency. Control of emissions of air pollution from nonroad diesel engines: Final Rule. *U.S. Federal Register of Rules and Regulations*, 63(205):56967–57023, October 1998.

[2] U.S. Environemental Protection Agency. Control of emissions from nonroad large spark-ignition engines, and recreational engines (marine and land-based). *U.S. Federal Register of Rules and Regulations*, 67(217):68241–68447, November 2002.

[3] U.S. Environemental Protection Agency. Control of emissions of air pollution from nonroad diesel engines and fuel: Final Rule. *U.S. Federal Register of Rules and Regulations*, 69(124):38957–39006, June 2004.

[4] The European Parliament and The Council Of The European Union. Directive 2000/14/EC of the european parliament and of the council. *Official Journal of the European Communities*, pages 1–777, May 2000.

[5] Human Resources Government of Alberta and Employment. Occupational health and safety code. *Occupational Health and Safety Act, Province of Alberta*, 2006.

[6] A. Clayton and P. Blow. U.S. department of transportation comprehensive truck size and weight study report no. 5. U.S. Department of Transportation, Federal Highway Administration, 1995.

[7] J. Poirot et al. Regulation of weights, lengths, and widths of commercial motor vehicles: TRB special report 267. Committee for the Study of the Regulation of Weights, Lengths, and Widths of Commercial Motor Vehicles, Transportation Research Board, 2002.

[8] W. Haycraft. *Yellow Steel: the story of the earthmoving equipment industry*. Uniersity of Illinois Press, 2000.

[9] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, March:35–45, 1960.

[10] G. Welch and Bishop G. An introduction to the kalman filter. Department of Computer Science, University of North Carolina at Chapel Hill. Available from http://www.cs.unc.edu/ welch/kalman/kalmanIntro.html, 2000.

[11] B. Widrow and M. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *IEEE*, 78(9):1415–1441, 1990.

[12] L. Talbert et al. A real-time adaptive speech recognition system. *Stanford University, Technical Report*, 1963.

[13] B. Widrow and S. Stearns. *Adaptive Signal Processing*. Prentice Hall, 1985.

[14] M. Hu. Application of the adaline system to weather forecasting. University of Stanfor, PhD Thesis, 1964.

[15] B. Widrow. The original adaptive neural net broom-balancer. *Proceedings of the IEEE Industrial Symposium on Circuits and Systems*, pages 351–357, 1987.

[16] B. Widrow et al. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63:1692–1716, 1975.

[17] F. Rosenblatt. The perceptron: A probabalistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[18] D. Rumelhart and J. McClelland. Parallel distributed processing, explorations in the microstructure of cognition. *Foundations, MIT Press, Cambridge, MA*, 1, 1986.

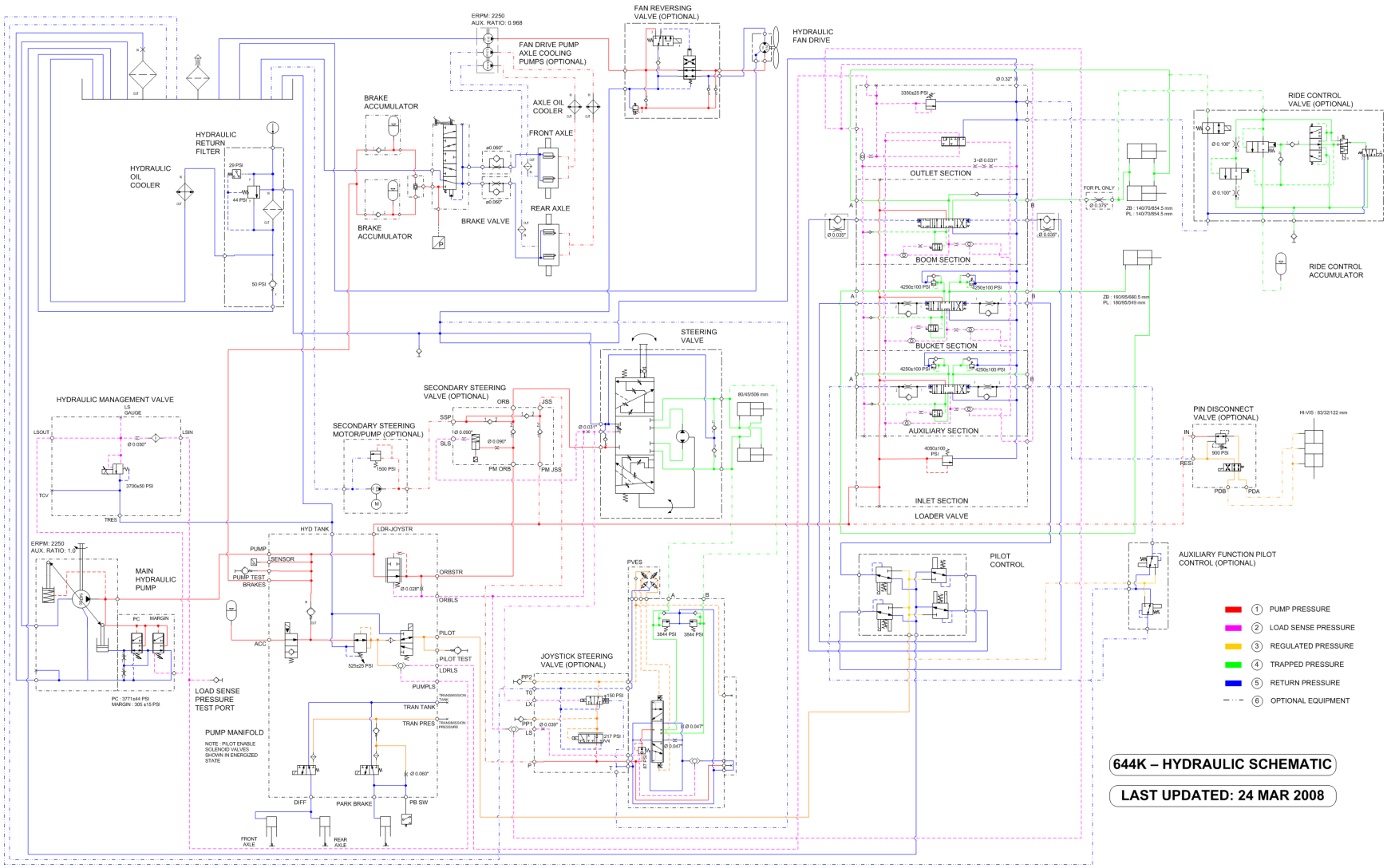
[19] D. Rumelhart, D.; Hinton and R. Williams. Learning representation by backpropagating errors. *Nature*, 323(9):533–536, 1986.

[20] P. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Note that this reference list does not include the references which appear only in the papers.

APPENDIX A

TYPICAL 4WD LOADER HYDRAULIC SCHEMATIC


ERPM: 2250
AUX. RATIO: 0.968
FAN REVERSING VALVE (OPTIONAL)
HYDRAULIC FAN DRIVE
FAN DRIVE PUMP
AXLE COOLING PUMPS (OPTIONAL)
BRAKE ACCUMULATOR
AXLE OIL COOLER
FRONT AXLE
REAR AXLE
HYDRAULIC RETURN FILTER
HYDRAULIC OIL COOLER
BRAKE VALVE
RIDE CONTROL VALVE (OPTIONAL)
RIDE CONTROL ACCUMULATOR
OUTLET SECTION
BOOM SECTION
BUCKET SECTION
AUXILIARY SECTION
INLET SECTION
LOADER VALVE
PILOT CONTROL
STEERING VALVE
HYDRAULIC MANAGEMENT VALVE
SECONDARY STEERING MOTOR/PUMP (OPTIONAL)
SECONDARY STEERING VALVE (OPTIONAL)
PIN DISCONNECT VALVE (OPTIONAL)
AUXILIARY FUNCTION PILOT CONTROL (OPTIONAL)
MAIN HYDRAULIC PUMP
LOAD SENSE PRESSURE TEST PORT
PUMP MANIFOLD
JOYSTICK STEERING VALVE (OPTIONAL)
1 PUMP PRESSURE
2 LOAD SENSE PRESSURE
3 REGULATED PRESSURE
4 TRAPPED PRESSURE
5 RETURN PRESSURE
6 OPTIONAL EQUIPMENT
644K – HYDRAULIC SCHEMATIC
LAST UPDATED: 24 MAR 2008

APPENDIX B

EXPLICIT BACKPROPOGATION LEARNING

Using the notation used in (2.31) and (2.32), the weight adjustment for the output layer weights is:

$$\Delta U_{k,j} = \eta \delta y_k h_j \tag{B.1}$$

where δy_k is defined by (2.26)
Similarly, the weight adjustment for the hidden layer weight is:

$$\Delta W_{j,i} = \mu \delta h_j X_i \tag{B.2}$$

where δh_j is defined by (2.30)

Assuming the use of the sigmoid activation function given in (2.20) the derivative is a simple function of the original equation and is given by (2.33). Using equations (B.1), (2.20), (2.33) and (2.23), the weight change can be expressed in terms of the output of the network and the desired values of the output. Performing the mathematical manipulation provides:

$$\Delta U_{k,j} = \eta \left[(y_k - d_k) f_{2k'} (1 - f_{2k'}) \right] h_j \tag{B.3}$$

$$\Delta U_{k,j} = \eta \left[(y_k - d_k) y_k (1 - y_k) \right] h_j \tag{B.4}$$

for $k = 1 \ldots K$ and $j = 1 \ldots M$ where:

$$f_{2k'} = f_{2k} \left(\sum_{j=0}^{M} U_{k,j} h_j \right) = \frac{1}{1 + e^{-\left(\Sigma_{j=0}^{M} U_{k,j} h_j \right)}} = y_k \tag{B.5}$$

It should be noted again that the validity of (B.4) is contingent upon the fact that the derivative of the sigmoid function is a function of the original sigmoid equation. A similar approach can be used to explicitly define the weight change for the hidden layer as a function of the input to the network. Using equations (B.2), (2.20), (2.33) and (2.28) the following expression can be determined:

$$\Delta W_{j,i} = \mu \left[\left(\left(\sum_{k=1}^{K} (y_k - d_k) \right) f_2' k \left(\sum_{j=0}^{M} U_{k,j} \right) U_{k,j} \right) f_{1k} (1 - f_{1k}) \right] X_i \tag{B.6}$$

$$\Delta W_{j,i} = \mu \left[\left(\sum_{k=1}^{K} \delta y_k U_{k,j} \right) f_{1k'} (1 - f_{1k'}) \right] X_i \tag{B.7}$$

$$\Delta W_{j,i} = \mu \left[\left(\sum_{k=1}^{K} \delta y_k U_{k,j} \right) h_j (1 - h_j) \right] X_i \tag{B.8}$$

for $j = 1 \ldots M$ and $k = 1 \ldots K$ where:

$$f_{1k'} = f_{1k} \left(\sum_{i=0}^{p} W_{j,i} X_i \right) = \frac{1}{1 + e^{-\left(\Sigma_{i=0}^{p} W_{j,i} X_i \right)}} = h_j \tag{B.9}$$

APPENDIX C

MASS-SPRING-DAMPER MODEL USED FOR NEURAL NETWORK TRAINING

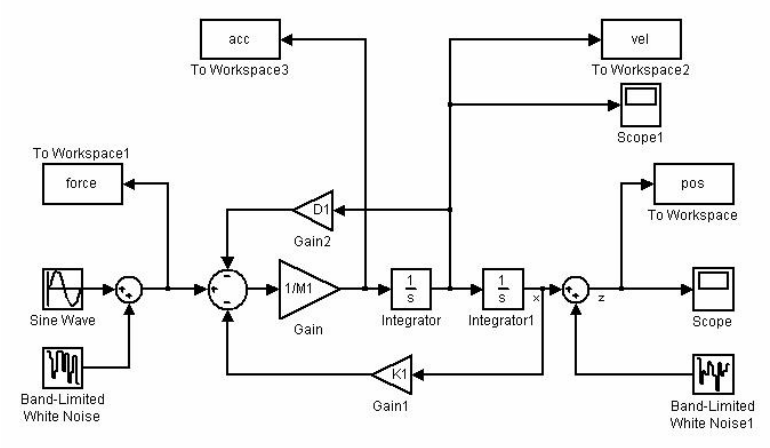


Figure C.1: Neural Network Simulation Model

APPENDIX D

NEURAL NETWORK TRAINING ALGORITHM

```
%This code employs a neural network to estimate the mass, damping and
%spring constant of a mass-spring-damper system. The inputs are:
%   Input:  pos     =  Measured Position of Mass
%           vel     =  Derivative of Position or Measured Velocity
%           acc     =  Derivative of Vel or Measured Acceleration
%           force   =  Discrete Forcing Function
%   Output: K       =  Spring Constant Estimate
%           M       =  Mass Estimate
%           D       =  Damping Estimate
%initialize variables
%THIS INCLUDES A LMS LEARNING LAW:
%Express Output as a function of input
%and provide a numerical derivative such that
%W(k+1)=W(k)+m*de/dW can be satisfied where m
%is some learning rate constant <1 and de/dW can
% be ascertained from i/o params via LMS error
%equal to 1/2sum(d-y)^2. For a single neuron, the
%output is equal to the sum of the inputs multiplied
%by the respective weights. y=K*pos+M*acc+D*vel.
%So, if e=1/2(d-K*pos+M*acc+D*vel)^2 de/dK, de/dM
%de/dD need to be calculated for minimization.
%
%Defining Initial Guesses for Parameters...If no momentum term is used in
%the training, the guesses for the second time step can be ignored.
K(1,1)=1;
K(1,2)=1;
M(1,1)=1;
M(1,2)=1;
D(1,1)=1;
D(1,2)=1;
%
%Defining the Learning Rate constants. mu is the learning rate (<1) and
%alpha is the momentum factor (<1). maxepoch is the maximum number of
%times the training data will be sent through the loop.
mu=.005;
alpha=.1;
maxepoch=5;
for q=1:maxepoch;
for n=2:length(pos);
    estforce=K(q,n)*pos(n)+M(q,n)*acc(n)+D(q,n)*vel(n);
    errornew(n)=estforce-force(n);
    lmserror(q,n)=.5*errornew(n)^2;
    K(q,n+1)=abs(K(q,n)-mu*(-force(n)*pos(n)+acc(n)*pos(n)*M(q,n)+vel(n)*pos(n)*D(q,n)+K(q,n)
            *pos(n)^2))+alpha*(K(q,n)-K(q,n-1));
    D(q,n+1)=abs(D(q,n)-mu*(-force(n)*vel(n)+acc(n)*vel(n)*M(q,n)+vel(n)*pos(n)*K(q,n)+D(q,n)
            *vel(n)^2))+alpha*(D(q,n)-D(q,n-1));
    M(q,n+1)=abs(M(q,n)-mu*(-force(n)*acc(n)+acc(n)*vel(n)*D(q,n)+acc(n)*pos(n)*K(q,n)+M(q,n)
            *acc(n)^2))+alpha*(M(q,n)-M(q,n-1));
```

```
    time(n+1)=n+1;
end
K(q+1,1)=K(q,length(pos));
K(q+1,2)=K(q,length(pos));
D(q+1,1)=D(q,length(pos));
D(q+1,2)=D(q,length(pos));
M(q+1,1)=M(q,length(pos));
M(q+1,2)=M(q,length(pos));
end
```

APPENDIX E

COLLECTED RAW DATA SAMPLE

Time (sec)	BoomRod (kpa)	BoomHead (kpa)	BoomPos (%)	BucketRod (kpa)	BucketHead (kpa)	BucketPos (%)
0	3.07E+02	1.03E+04	2.07E+00	-1.62E+03	2.40E+04	9.97E+01
0.015	3.07E+02	1.02E+04	2.21E+00	-1.63E+03	2.40E+04	9.97E+01
0.03	3.07E+02	1.01E+04	2.32E+00	-1.62E+03	2.40E+04	9.97E+01
0.045	3.10E+02	1.00E+04	2.43E+00	-1.62E+03	2.40E+04	9.97E+01
0.06	3.10E+02	9.95E+03	2.53E+00	-1.63E+03	2.40E+04	9.97E+01
0.075	3.12E+02	9.87E+03	2.64E+00	-1.63E+03	2.40E+04	9.97E+01
0.09	3.12E+02	9.79E+03	2.76E+00	-1.62E+03	2.40E+04	9.97E+01
0.105	3.17E+02	9.72E+03	2.85E+00	-1.62E+03	2.40E+04	9.97E+01
0.12	3.15E+02	9.65E+03	2.96E+00	-1.62E+03	2.40E+04	9.97E+01
0.135	3.17E+02	9.57E+03	3.05E+00	-1.62E+03	2.39E+04	9.97E+01
0.15	3.15E+02	9.53E+03	3.17E+00	-1.62E+03	2.39E+04	9.97E+01
0.165	3.07E+02	9.50E+03	3.29E+00	-1.61E+03	2.39E+04	9.97E+01
0.18	2.97E+02	9.45E+03	3.37E+00	-1.61E+03	2.39E+04	9.97E+01
0.195	2.85E+02	9.43E+03	3.46E+00	-1.60E+03	2.39E+04	9.97E+01
0.21	2.68E+02	9.43E+03	3.58E+00	-1.59E+03	2.39E+04	9.97E+01
0.225	2.52E+02	9.43E+03	3.62E+00	-1.58E+03	2.38E+04	9.97E+01
0.24	2.40E+02	9.48E+03	3.70E+00	-1.57E+03	2.38E+04	9.97E+01
0.255	2.32E+02	9.52E+03	3.81E+00	-1.55E+03	2.38E+04	9.97E+01
0.27	2.25E+02	9.53E+03	3.88E+00	-1.53E+03	2.38E+04	9.97E+01
0.285	2.20E+02	9.58E+03	3.94E+00	-1.51E+03	2.37E+04	9.97E+01
0.3	2.12E+02	9.64E+03	4.05E+00	-1.50E+03	2.37E+04	9.97E+01
0.315	2.03E+02	9.68E+03	4.15E+00	-1.48E+03	2.37E+04	9.97E+01
0.33	1.96E+02	9.76E+03	4.20E+00	-1.46E+03	2.37E+04	9.97E+01
0.345	1.86E+02	9.83E+03	4.31E+00	-1.44E+03	2.36E+04	9.97E+01
0.36	1.79E+02	9.91E+03	4.38E+00	-1.42E+03	2.36E+04	9.97E+01
0.375	1.71E+02	1.00E+04	4.45E+00	-1.40E+03	2.36E+04	9.97E+01
0.39	1.71E+02	1.01E+04	4.54E+00	-1.37E+03	2.36E+04	9.97E+01
0.405	1.74E+02	1.02E+04	4.60E+00	-1.35E+03	2.35E+04	9.97E+01
0.42	1.77E+02	1.03E+04	4.61E+00	-1.33E+03	2.35E+04	9.97E+01
0.435	1.86E+02	1.04E+04	4.68E+00	-1.31E+03	2.35E+04	9.97E+01
0.45	1.93E+02	1.04E+04	4.77E+00	-1.28E+03	2.35E+04	9.97E+01
0.465	2.03E+02	1.05E+04	4.84E+00	-1.25E+03	2.34E+04	9.97E+01
0.48	2.13E+02	1.06E+04	4.94E+00	-1.22E+03	2.34E+04	9.97E+01
0.495	2.23E+02	1.06E+04	5.01E+00	-1.19E+03	2.34E+04	9.97E+01
0.51	2.34E+02	1.07E+04	5.07E+00	-1.15E+03	2.34E+04	9.97E+01
0.525	2.46E+02	1.07E+04	5.18E+00	-1.10E+03	2.33E+04	9.97E+01
0.54	2.56E+02	1.07E+04	5.24E+00	-1.06E+03	2.33E+04	9.97E+01
0.555	2.66E+02	1.07E+04	5.35E+00	-1.00E+03	2.33E+04	9.97E+01
0.57	2.76E+02	1.06E+04	5.44E+00	-9.45E+02	2.32E+04	9.97E+01
0.585	2.81E+02	1.06E+04	5.50E+00	-8.84E+02	2.32E+04	9.97E+01
0.6	2.85E+02	1.06E+04	5.62E+00	-8.17E+02	2.32E+04	9.97E+01
0.615	2.83E+02	1.05E+04	5.70E+00	-7.45E+02	2.31E+04	9.97E+01
0.63	2.86E+02	1.05E+04	5.80E+00	-6.60E+02	2.31E+04	9.97E+01
0.645	2.90E+02	1.05E+04	5.89E+00	-5.79E+02	2.30E+04	9.97E+01
0.66	2.93E+02	1.04E+04	6.00E+00	-5.02E+02	2.30E+04	9.97E+01
0.675	2.98E+02	1.04E+04	6.08E+00	-4.19E+02	2.30E+04	9.97E+01
0.69	3.09E+02	1.03E+04	6.15E+00	-3.33E+02	2.29E+04	9.97E+01
0.705	3.12E+02	1.03E+04	6.27E+00	-2.54E+02	2.29E+04	9.97E+01