12-2016

Sensor Based Nitrogen Management for Corn Production in Coastal Plain Soils

Nicholas G. Rogers
Clemson University

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Recommended Citation
Rogers, Nicholas G., "Sensor Based Nitrogen Management for Corn Production in Coastal Plain Soils" (2016). All Theses. 2579.
https://tigerprints.clemson.edu/all_theses/2579

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kookefe@clemson.edu.
SENSOR BASED NITROGEN MANAGEMENT FOR CORN PRODUCTION IN COASTAL PLAIN SOILS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Plant and Environmental Science

by
Nicholas G. Rogers
December 2016

Accepted by:
Dr. Ahmad Khalilian, Co-Advisor
Dr. Young J. Han, Co-Advisor
Dr. Joe Mari Maja
ABSTRACT

The first objective of this project was to develop, refine, and employ sensor-based algorithms to determine the mid-season nitrogen requirements for the production of irrigated corn in Coastal Plain soils. The second objective of this project was to develop an “on-the-go” variable rate nitrogen application system. One (1) production field at Clemson’s Edisto Research and Education Center in Blackville, SC was used in the development and refinement of the sensor-based algorithm. This field was equipped with overhead irrigation, and was used in both the 2015 and 2016 growing seasons. The crop was not under scheduled irrigation, the overhead (lateral) irrigation system was only utilized to provide deficit irrigation for corn. The field was divided into two separated zones based on soil electrical-conductivity (EC) data. The algorithm was developed using varied prescription rate nitrogen plots. These plots received nine different rates of nitrogen fertilizer (0, 20, 40, 80, 120, 160, 200, 240, and 260 lbs. N/ac). Nitrogen treatments were replicated 5 times in plots of each zone using a Randomized Complete Block design. To apply nitrogen to the research plots during both growing seasons, a previously constructed custom built applicator utilizing a hydraulic pump in combination with an in-cab control system was used.

Optical sensor readings were collected from the test plots between the V6 to V8 growth stages to determine the corn plants Normalized Difference Vegetation Index (NDVI). The sensor readings were used to develop the algorithm to be used in the estimation of side-dress nitrogen application in corn. There was a good correlation between combined sensor readings collected during 2015 and 2016 growing seasons (V6
to V8 stage) and actual corn yields ($R^2 > 0.68$). In Season Estimated Yield (INSEY) was used along with the actual yield to produce a yield potential (YP0) for each growing season for deficit irrigated corn crop.

During the 2015 growing season the algorithm for estimating the amount of mid-season side-dress nitrogen application was developed. Data collected from the 2016 growing season was then used to further refine this algorithm. The algorithm developed during 2015 and 2016 growing seasons was used during the 2016 to estimate the amount of mid-season side-dress nitrogen required for corn in two different management zones of a 5-acre production field. The algorithm recommended reduced rates of nitrogen, 21% and 34% in zone 1 and 2, respectively, compared to the normal grower practice (200 lbs. N/acre) with no reduction in corn yields.

The accuracy of the optical sensor (GreenSeeker®) was tested to determine if the time of day, temperature, or solar radiation affect its performances. The results of this test showed that the sensor is affected by the time of day that the readings are taken, but during the time frame of three hours after sunrise and one hour before sunset the readings were not affected.

The second objective of this study was specifically to develop an “on-the-go” variable rate nitrogen application system. Current nitrogen application systems are designed to apply a relatively uniform amount of nitrogen to agricultural fields. There are several commercially available variable rate nitrogen application systems, such as John Blue pumps equipped with Rawson controller (Porter, 20010) or John Blue’s Direct...
Drive Hydraulic Piston Pump (CDS-John Blue Company, Huntsville, AL). However, with these systems only limited range of flow rates can be achieved by changing the drive shaft speed. This limited flow range is not sufficient for applying variable-rate crop inputs (such as nitrogen) in the Southeastern USA, with tremendous amount of variations in field conditions and soil types. There is a need for a controller which can adjust the pump stroke on-the-go, for real-time, variable-rate application of crop inputs. As a result of this study a control system called “The Clemson electro-mechanical controller for adjusting pump stroke on-the-go” was developed to replace the current manual stroke adjustment system on positive displacement piston pumps. This affordable system could be retrofitted on any existing piston pump (such as John Blue), which makes it possible to change the flow rate of the pump automatically from zero to a pumps full capacity. The resulting system has the ability to be controlled either manually by an electronic dial (rotary potentiometer) from tractor’s cab or by a map-based control program.
DEDICATION

I would like to dedicate this thesis to my parents, David and Renea Rogers for their continued encouragement through my college career, to my grandparents Carl and Sue Jones and Bill Rogers, for their continual support in my educational career, and to my wife Holly Rogers for her encouragement to continue with my pursuit of a Master’s Degree.

I would also like to dedicate this thesis to all of my friends who helped me stay motivated throughout my pursuit of a higher education.
ACKNOWLEDGMENTS

I would like to thank all of the people that have helped me in the course of this project and those that made it all possible. I would like to thank Dr. Ahmad Khalilian for his role as my research advisor, for all of his support and research guidance, and the financial support for this project. I would also like to thank Dr. Young J. Han for his role as my academic advisor, and his support in helping me become a successful graduate student. I would also like to thank Dr. Ali Nafchi for all of his help and expertise. I would also like to thank Richard Hallman, Paul Thomas, Philip Williams, and Bobby Fralix for all of their help with the field work and data collection that made this project possible.

I would like to acknowledge Dr. Kendall R. Kirk, for his role as a teacher and his advice. I would like to also acknowledge Hunter F. Massey for his support and friendship.

I would like to thank Christina Leard for her handling of all of my administrative needs.

I would like to acknowledge Henry Ramsey, Reid Miller, and the undergraduate capstone group for all of their help with data collection.

Lastly I would like to thank all of my friends and family for their continuing support of my education and for helping to keep me motivated and helping me to achieve my goals.
TABLE OF CONTENTS

Page

TITLE PAGE ................................................................................................................................. i
ABSTRACT ................................................................................................................................. ii
DEDICATION ............................................................................................................................... v
ACKNOWLEDGMENTS ............................................................................................................... vi
LIST OF FIGURES ..................................................................................................................... ix

CHAPTER

I. DEVELOPMENT AND REFINEMENT OF A SENSOR-BASED ALGORITHM FOR NITROGEN APPLICATION ON CORN UNDER DEFICIT IRRIGATION IN COASTAL PLAIN SOILS ............... 1

1.1 Introduction ....................................................................................................................... 1
1.2 Objectives ......................................................................................................................... 4
1.3 Review of Literature ......................................................................................................... 5
   1.3.1 Nitrogen Use by Corn ................................................................................................. 5
   1.3.2 Yield Goals ............................................................................................................... 6
   1.3.3 Remote Optical Sensors .......................................................................................... 7
   1.3.4 Optical Sensor Based Nitrogen Management ......................................................... 8
1.4 Materials and Methods ..................................................................................................... 11
   1.4.1 Equipment ............................................................................................................... 11
   1.4.2 Field Experiments .................................................................................................. 15
1.5 Results and Discussion ..................................................................................................... 22
   1.5.1 Algorithm Development and Refinement ............................................................... 22
   1.5.2 Algorithm Testing .................................................................................................... 35
1.6 Conclusions ....................................................................................................................... 37
1.7 References ......................................................................................................................... 38

II. DEVELOPMENT AND TESTING OF EQUIPMENT TO ACCURATELY APPLY SENSOR-BASED VARIABLE RATE NITROGEN FERTILIZER ON CORN IN COASTAL PLAIN SOILS .......... 40

2.1 Introduction ....................................................................................................................... 40
2.2 Objectives ......................................................................................................................... 42
2.3 Review of Literature ......................................................................................................... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1 Variable Rate Fertilizer Applicators</td>
<td>43</td>
</tr>
<tr>
<td>2.3.2 Factors Affecting the GreenSeeker Performance</td>
<td>46</td>
</tr>
<tr>
<td>2.4 Materials and Methods</td>
<td>48</td>
</tr>
<tr>
<td>2.4.1 Nitrogen Fertilizer Applicator</td>
<td>48</td>
</tr>
<tr>
<td>2.4.2 Normalized Difference Vegetation Index (NDVI)</td>
<td>55</td>
</tr>
<tr>
<td>2.5 Results and Discussion</td>
<td>59</td>
</tr>
<tr>
<td>2.5.1 Nitrogen Fertilizer Applicator</td>
<td>59</td>
</tr>
<tr>
<td>2.5.2 Factors Affecting the GreenSeeker Performance</td>
<td>60</td>
</tr>
<tr>
<td>2.6 Conclusions</td>
<td>64</td>
</tr>
<tr>
<td>2.7 References</td>
<td>66</td>
</tr>
<tr>
<td><strong>APPENDICES</strong></td>
<td>68</td>
</tr>
<tr>
<td>A: Visual Basic Code for Manual Control</td>
<td>69</td>
</tr>
<tr>
<td>B: Visual Basic Code for Map Based Control</td>
<td>75</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Geo-Referenced Veris 3100 Soil EC Meter</td>
</tr>
<tr>
<td>1.2</td>
<td>Schematic of Veris 3100 Soil EC Meter (Veris Technologies)</td>
</tr>
<tr>
<td>1.3</td>
<td>Shallow and Deep EC Data Points</td>
</tr>
<tr>
<td>1.4</td>
<td>The Clemson Ramp Applicator used to apply different N rate to test plots</td>
</tr>
<tr>
<td>1.5</td>
<td>Sprayer Mounted GreenSeeker® RT-200 System</td>
</tr>
<tr>
<td>1.6</td>
<td>The Experimental Field Used</td>
</tr>
<tr>
<td>1.7</td>
<td>Raw Soil EC Points with Plots</td>
</tr>
<tr>
<td>1.8</td>
<td>Average EC Plots</td>
</tr>
<tr>
<td>1.9</td>
<td>Typical NRCS Soil Survey Map and Soil EC Data</td>
</tr>
<tr>
<td>1.10</td>
<td>Plot Layout Plan</td>
</tr>
<tr>
<td>1.11</td>
<td>The correlation between INSEY and actual corn yields, 35 days after planting, 2015</td>
</tr>
<tr>
<td>1.12</td>
<td>The correlation between INSEY and actual corn yields, 39 days after planting, 2015</td>
</tr>
<tr>
<td>1.13</td>
<td>The correlation between INSEY and actual corn yields, 43 days after planting, 2015</td>
</tr>
<tr>
<td>1.14</td>
<td>The correlation between INSEY and actual corn yields, 46 days after planting, 2015</td>
</tr>
<tr>
<td>1.15</td>
<td>The correlation between INSEY and actual corn yields, 49 days after planting, 2015</td>
</tr>
<tr>
<td>1.16</td>
<td>The correlation between INSEY and actual corn yields, 51 days after planting, 2015</td>
</tr>
</tbody>
</table>
List of Figures (Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.17</td>
<td>Yield Prediction Equation from 2015 (All EC zones combined)</td>
</tr>
<tr>
<td>1.18</td>
<td>$R^2$ values and graph of the two management zones</td>
</tr>
<tr>
<td>1.19</td>
<td>The correlation between INSEY and actual corn yields, 34 days after planting, 2016</td>
</tr>
<tr>
<td>1.20</td>
<td>The correlation between INSEY and actual corn yields, 38 days after planting, 2016</td>
</tr>
<tr>
<td>1.21</td>
<td>The correlation between INSEY and actual corn yields, 40 days after planting, 2016</td>
</tr>
<tr>
<td>1.22</td>
<td>The correlation between INSEY and actual corn yields, 42 days after planting, 2016</td>
</tr>
<tr>
<td>1.23</td>
<td>The correlation between INSEY and actual corn yields, 45 days after planting, 2016</td>
</tr>
<tr>
<td>1.24</td>
<td>The correlation between INSEY and actual corn yields, 54 days after planting, 2016</td>
</tr>
<tr>
<td>1.25</td>
<td>The correlation between INSEY and actual corn yields, 56 days after planting, 2016</td>
</tr>
<tr>
<td>1.26</td>
<td>2016 Yield Prediction Equation</td>
</tr>
<tr>
<td>1.27</td>
<td>Combined 2015 and 2016 Yield Prediction Equation</td>
</tr>
<tr>
<td>1.28</td>
<td>Nitrogen Response Based on Soil EC Zones (2015)</td>
</tr>
<tr>
<td>1.29</td>
<td>Nitrogen Response Based on Soil EC Zones (2016)</td>
</tr>
<tr>
<td>1.30</td>
<td>Clemson vs. Farmer Practice N Usage</td>
</tr>
<tr>
<td>1.31</td>
<td>Clemson vs. Farmer Practice Yield</td>
</tr>
<tr>
<td>2.1</td>
<td>Pump Data</td>
</tr>
</tbody>
</table>
List of Figures (Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Clemson System on a John-Blue NGP 6055</td>
<td>50</td>
</tr>
<tr>
<td>2.3</td>
<td>KBH Applicator</td>
<td>50</td>
</tr>
<tr>
<td>2.4</td>
<td>Controller for Nitrogen Applicator</td>
<td>52</td>
</tr>
<tr>
<td>2.5</td>
<td>Clemson Variable Rate Applicator Control Program</td>
<td>52</td>
</tr>
<tr>
<td>2.6</td>
<td>Controller Wiring Diagram</td>
<td>52</td>
</tr>
<tr>
<td>2.7</td>
<td>Electronic Dial Control Flow Chart</td>
<td>53</td>
</tr>
<tr>
<td>2.8</td>
<td>The side-dress Nitrogen Applicator Retrofitted with Hydraulic system</td>
<td>54</td>
</tr>
<tr>
<td>2.9</td>
<td>Automatic Control Flow Chart</td>
<td>55</td>
</tr>
<tr>
<td>2.10</td>
<td>NDVI Shape File Example</td>
<td>56</td>
</tr>
<tr>
<td>2.11</td>
<td>John Deere 6700 with NTech GreenSeeker® RT200 System</td>
<td>57</td>
</tr>
<tr>
<td>2.12</td>
<td>Aerial Photo of Plots</td>
<td>58</td>
</tr>
<tr>
<td>2.13</td>
<td>GreenSeeker®</td>
<td>58</td>
</tr>
<tr>
<td>2.14</td>
<td>Trimble Nomad Handheld Computer</td>
<td>59</td>
</tr>
<tr>
<td>2.15</td>
<td>GreenSeeker® Data Collected Over Corn and Green Cloth</td>
<td>60</td>
</tr>
<tr>
<td>2.16</td>
<td>Static Test Results</td>
<td>62</td>
</tr>
<tr>
<td>2.17</td>
<td>GreenSeeker® Sensor Readings throughout the Day</td>
<td>63</td>
</tr>
<tr>
<td>2.18</td>
<td>Comparison of Corn and Green Cloth NDVI Sensor Readings</td>
<td>63</td>
</tr>
<tr>
<td>2.19</td>
<td>Temperature vs. NDVI</td>
<td>64</td>
</tr>
<tr>
<td>2.20</td>
<td>Solar Radiation and NDVI</td>
<td>65</td>
</tr>
</tbody>
</table>
CHAPTER ONE

DEVELOPMENT AND REFINEMENT OF A SENSOR-BASED ALGORITHM FOR NITROGEN APPLICATION ON CORN UNDER DEFICIT IRRIGATION IN COASTAL PLAIN SOILS

1.1 INTRODUCTION

Total corn \((Zea\ mays\ L.)\) acreage in the United States has varied over the years and currently is 96 million acres (USDA, NASS, 2013). Growers are currently spending more money on inputs on the same amount of crop grown with the production remaining the same.

The rising trend of nitrogen prices coupled with a dropping corn commodity price over the past several decades are leaving producers with a challenging decision to make. High production costs make it increasingly important for our growers to reduce crop input costs while maximizing yields to stay competitive in the global market. A method and tool must be devised to allow producers to make an educated decision on the amount of nitrogen they should apply to any given field, during any given growing season. The method will allow, based on the given conditions, a variable or reduced nitrogen rate to be applied to reduce input costs. The method can be most easily implemented by the formation of a nitrogen prediction algorithm.

Nitrogen management is an important factor in the production of corn, as both deficiencies and excess negatively affect plant growth, thus reducing yields. Insufficient nitrogen supply often reduces leaf area, leaf photosynthesis rate, and biomass production in corn resulting in lower yields. Environmental factors also have to be considered, when excess amounts of
nitrogen are applied, it becomes susceptible to leaching, which could in turn containment ground water (Inman 2005). Excessive nitrogen applications also increase production costs, and can have a negative effect on corn yields. In addition, in coastal plain soils, the yield response to nitrogen application also varies significantly among different sections of a production field, even in small fields (Khalilian et al., 2011). They reported that, in low EC areas (lighter soil texture), cotton yield increased as nitrogen rates increased. However, in medium EC areas there was no yield response above 90 lbs./acre nitrogen. In high EC areas (heavier soils), nitrogen rates above 60 lbs./acre, had no effect on cotton yield. Therefore, uniform application of N fertilizer over the entire field can be both costly and environmentally unsound.

On average, growers in the US apply about 140 lbs./acre nitrogen for corn, for a total of 6.8 million tons (USDA, NASS, 2013). High production costs make it increasingly important for our growers to reduce crop input costs while maximizing yields to stay competitive in the global market. For example, a 20% reduction in nitrogen usage could save US corn growers over $1.2 billion annually. In Coastal plain soils, considerable soil variation occurs within production fields in soil texture, soil type, water holding capacity and other major factors which affect crop production and will have a major impact on fertilizer management strategies. Mobile nutrients (N) are used, lost, and stored differently as soil texture varies.

Several researchers across the cotton and corn producing states have developed algorithms for nitrogen fertilization based on optical sensors (Earnest and Varco, 2005; Scharf et al., 2008; Arnall et al., 2008; Raun et al., 2005). However, due to higher annual precipitation, significant variation in soil type and texture, low soil organic matter content, and low nutrient holding capacity of soils in Coastal Plain regions, N-application algorithms, developed at other regions, either under- or over-estimated nitrogen rates for crop production (Khalilian et al. 2011).
This study is focused on developing nitrogen-algorithms for corn, specifically designed for Coastal Plain region, utilizing plant NDVI and soil electrical conductivity (EC) data and determining the effects of soil texture on corn yield in response to nitrogen.
1.2 OBJECTIVES

Main Objective: To develop an algorithm that utilizes an optical sensor and soil EC to predict side-dress nitrogen application for corn on irrigated fields in the Coastal Plain regions of the United States.

The specific objective was:

1. To develop/refine an algorithm for prediction of side-dress nitrogen requirements for deficit irrigated corn utilizing plant normalized difference vegetation index (NDVI) and soil electrical conductivity data.

2. To compare the Clemson algorithm to a typical grower’s practice in terms of effects on corn yields and nitrogen requirements in the.
1.3 REVIEW OF LITERATURE

The focus of this section is to review the literature related to the study objectives. It consists of four subheadings:

1. Nitrogen Use by Corn
2. Yield Goals
3. Remote Optical Sensors
4. Optical Sensor Based Nitrogen Management

1.3.1 Nitrogen Use by Corn

Plant nutrients such as N, P, and K are often applied to ensure economically viable grain yields in large-scale cropping systems (Swanson, 1982). Nitrogen is often the most limiting nutrient in agro-ecosystems and is therefore applied in the highest quantities (Havlin et al. 1999; FAO 2001). Nitrogen is one of the three macronutrients that account for the majority of fertilizer applied to meet crop demand. Although other macro- and micronutrients are needed, they are not regularly applied by producers due to the relatively small amounts needed and adequate available natural levels. The rate of increase and the amount of each nutrient (N, P, and K) accumulated during the growing season differs based on the plant component measured (Abendroth et al. 2011). Accumulated soil N is highly susceptible to leaching and can potentially threaten groundwater supplies. One way of maintaining soil N fertility levels without exceeding crop N requirements is to tailor N inputs to meet the specific crop N requirements (Inman et al. 2005). Significant nutrient accumulation, within grain does not occur prior to R2, although a rapid and near linear increase begins at R2 to approximately R5. During R5, N and P continue to be
accumulated at a similar rate as before but K accumulation is lessened (Abendroth et al. 2011). Except for grain, leaf blades have the highest fraction of N, with maximum accumulation at R2. Components decrease in total nitrogen at R1 or R2 due to remobilization of the nitrogen to the developing grain as well as senescence of the lower leaves. When mature (R6), 67% of total plant nitrogen is contained in the grain (Abendroth et al. 2011).

A three year study conducted in Colorado found that grain yield and N uptake across irrigated corn production fields exhibited significant spatial variability, and between management zones, N uptake and grain yield were statistically different (Inman et al. 2005). Grain yield response to N was also shown to be significantly different. The results suggested that spatially variable crop parameters could potentially be managed using site specific management zones (Inman et al. 2005). It has been shown that different soil types and textures require different nutrient management to ensure that plants receive proper amounts of nitrogen at the proper time.

The environment also has to be considered when making decisions determining fertilizer rates and timing of applications. Inman et al. (2005) and Jaynes et al. (2005) agreed that when high amounts of N are accumulated in the soil, it becomes highly susceptible to leaching and could contaminate the ground water. Jaynes et al. (2005) further suggests that splitting the N application between planting and early season is a sound agronomic and environmental practice for corn production.

1.3.2 Yield Goals

Corn nitrogen recommendations have previously been determined by using the yield goal approach by establishing yield goals based on field history and/or current capability. The amount
of N required by the targeted yield is then determined from crop nutrient removal tables (Agronomy, 2016). Pioneer suggests that available N can come from residual N from prior to planting, soil organic matter, and animal wastes and other organic amendments. According to Arnall (2008) the result of using yield goals is explained to be the minimum quantity of fertilizer N needed to ensure sufficient N to achieve the yield goal. Fertilization based on yield goals is a vast improvement over simply applying the same amount of N year after year, especially when credits and residual N are accounted for (Arnall 2008). This method can be limited depending of the accuracy of historical field averages, and has been shown growing conditions have an effect on the N mineralization from soil organic matter. Results from an experiment done in Nebraska showed that maximum yields were attainable with an additional in-season N applications when N stress symptoms were not severe, it was also shown that less N was applied in-season with sensor monitoring compared to N applied only at planting (Barker et al. 2012).

1.3.3 Remote Optical Sensors

Scientists with the Agricultural Research Service have provided much of the fundamental information and theory relating to spectral reflectance properties of crops to their agronomic and biophysical characteristics. This knowledge has been used for the development and use of non-destructive monitoring of plant growth and development. Coupled with new remote-sensing and position-locating technologies the spatial and temporal information on plant response to their local environment which is required for site specific management approaches is now available (Pinter et al. 2003).

Remote sensing approaches can provide growers with final yield assessments and show variations across fields. Optical sensors differ from, and have an advantage over, yield monitors in that they can be taken frequently during the season, thus providing temporal information on
growth rates and plant response to dynamic weather conditions and management practices (Pinter et al. 2003). The GreenSeeker® optical sensor is an active sensor that emits two bands of light, red and Near Infrared (NIR), and measures the amount of reflectance. The value reported from this measurement is the indices termed Normalized Difference Vegetation Index (NDVI) (Raun et al., 2005):

$$NDVI = \frac{(NIR_{reflectance} - Red_{reflectance})}{(NIR_{reflectance} + Red_{reflectance})}$$

According to Yoon and Thai (2009) live green plant canopies absorb solar radiation at the blue and red spectral regions for photosynthesis, NDVI was developed to utilize the properties of the different light energy absorption characteristics of plants at the near-infrared (NIR) and red spectral regions, and has been widely used in remote sensing as an index to estimate various vegetation properties including chlorophyll concentration in leaves, leaf area index, biomass, plant vigor, plant productivity, and stress. The sensor works because plants with more leaf area and chlorophyll absorb higher levels of red light. Therefore, healthy plants are able to reflect more NIR than less healthy plants due to turgid and healthy mesophyll cells (Arnall 2008). The ratio of the level of reflectance of red and NIR are highly useful when using NDVI as an indirect measure of plant health (Arnall 2008). Simply put, the higher the NIR reflectance combined with a low reflectance of visible red, means greener and healthier plants.

1.3.4 Optical Sensor Based Nitrogen Management

A situation in production corn fields where active sensor technologies may improve N management are those where the crop fertilization requirement is high and varies considerably or when N losses from excessive rainfall are uncertain or unavoidable (Barker et al. 2012). An
optical sensor could be used as the solution to N management in the fields with large soil variability’s in most cases. Daughtry et al. (2000) noted that leaf chlorophyll concentration is an indicator of corn N status. Changes in leaf chlorophyll concentrations produce rather broad-band differences in leaf reflectance. Based on this the authors agreed that the use of remote sensing of chlorophyll content has the potential to estimate corn N status (Daughtry et al. 2000).

Taking an indirect approach, Raun et al. (2001) reasoned that a mid-season, remote estimate of potential yield would help growers adjust top-dress N applications based on pre-plant soil N tests, within season rates of mineralization, and projected N removal (Pinter et al. 2003). In a variable N rate plot study conducted at Iowa State University by Barker and Sawyer (2010) an active canopy sensor detected corn N stress. The results showed that canopy sensors can measure N stress during mid-vegetative growth stages in corn and that canopy biomass or canopy chlorophyll can be used as an N rate algorithm for applying N fertilizer in-season. The results of another study conducted by Barker and Sawyer (2012) showed that sensor-directed N application gives corn growers options for addressing in-season N deficiency and protecting yield potential.

Research has shown that leaf chlorophyll content is correlated to NIR and red bands and that the vegetation indices containing red reflectance are significantly correlated to leaf N concentration. The characteristics of leaf chlorophyll concentration at the field level holds promise as a valuable aid for decision making in managing N application (Daughtry et al. 2000).

The relationship between NDVI sensor readings and the growing degree days (GDD), for predicting grain yield, has been found in multiple studies (Wiatrak et al., 2008; Raun et al., 2008). In a study conducted in South Carolina (Wiatrak et al., 2008) it was found that the optimum dates of sensing corn plants for calculating nitrogen requirement, falls somewhere
within the time frame of 39-67 days after emergence. They reported that, readings taken before or after this date were poorly correlated with the actual yield and therefore could not be used to predict the in-season N requirements. The optimum sensing range of these studies fall within the range of being able to apply side-dress nitrogen for corn that will not result in lost yield potential (Raun et al., 2008). The purpose of this study was to determine the optimum range of days after planting (DAP) in which sensor readings would provide an accurate prediction of side-dress nitrogen requirements in coastal plain soils.

Several researchers across the cotton and corn producing states have developed algorithms for N fertilization based on optical sensors (Earnest and Varco, 2005; Scharf et al., 2008; Arnall et al., 2008; Raun et al., 2005). However, due to higher annual precipitation, significant variation in soil type and texture, low soil organic matter content, and low nutrient holding capacity of soils in Coastal Plain regions, N-application algorithms, developed at other regions, either under- or over-estimated nitrogen rates for crop production (Khalilian et al. 2011). There is a need for sensor-based nitrogen-algorithms for corn, specifically designed for Coastal Plain region, to account for soil and climatic variables characteristic of this region.
1.4 MATERIALS AND METHODS

1.4.1 Equipment

1.4.1.1 Soil Electrical-Conductivity Meter

A commercially available Veris 3100 soil electrical conductivity meter (Veris Technologies, Salina, Kansas) was used to measure soil-texture variability of the test fields. The Veris 3100 EC meter (Lund et al., 1999) consists of six straight-blade coulter-electrodes attached to the back of a trailer-type frame (Figure 1.1). The height and depth into the soil of the six disks are controlled with a hydraulic cylinder. As the sensor is pulled through the field, one pair of the coulters emits an electric current into the soil while the other coulter pairs receive the current and measure the amount of current that was conducted through the soil in milliSems per meter (mS/m). A loss in voltage is then correlated to the soils ability to conduct electricity and thus to its properties and type (Figure 1.2).

Figure 1.1. Geo-Referenced Veris 3100 Soil EC Meter.
The Veris Soil EC meter is capable of measuring at two depths, shallow and deep, 0-12 inches and 0-36 inches respectively; the greater the distance between the pairs of coulter-electrodes, the deeper the soil EC measurement that can be collected, as seen from Figure 1.2. The Veris can be linked with a Global Positioning System (GPS) to produce a continuous georeferenced soil texture map. The Veris logs data on a 1 Hz cycle, thus creating a new data point for each second, at each sampling depth (Figure 1.3).
The collected data points can be viewed using any Geographic Information System (GIS), such as SSToolbox, Ag Leader/SMS, or FarmWorks. The GIS program can be used to average the EC data within designated plots.

1.4.1.2 Nitrogen Applicator

Conventionally, growers apply side-dress nitrogen applications twice a growing season. The typical application process is as follows: the first application occurs at planting with a rate of 20 to 30 lbs/ac, then one a side-dress application occurs at the V-8 to V10 growth stage at a rate of 200 lbs/ac for irrigated corn, or two split side-dress applications could be used. During this study, a modified multi-boom applicator was used to apply different rates of nitrogen to test plots (Figure 1.4). This system (called a RAMP applicator) is capable of applying 16 different rates of nitrogen (0 to 300 lbs. N/acre). The ramp applicator was equipped with four sets of nozzles (booms) selected to apply 1x, 2x, 4x, and 8x rates. The rate controller selects combinations of these nozzles to apply desired applications rates. For example, in this study these rates were 0, 20, 40, 80, 120, 160, 200, 240, and 260 lbs. N/acre). A ground radar was used to adjust desired length over which each application rate is applied.
1.4.1.3 *GreenSeeker® RT200 System* 

To collect NDVI data, a commercially available optical sensor, the GreenSeeker® RT-200 mapping system (NTech Industries, Inc. Ukiha, CA), was utilized during the 2015 and 2016 growing seasons. This system utilizes six separate optical sensors that were retrofitted to a John Deere 6700 Hi-Boy self-propelled sprayer (Figure 1.5). This system was designed to map the center six rows of the eight row plots and is on 38 inch centers. The readings collected from the six separate sensors are averaged into one reading, though individual sensor readings can be exported and viewed individually if necessary. Sensor data is logged on a 1 Hz cycle, and is captured and stored on a Trimble Nomad handheld onboard computer that is also linked to the Trimble Differential GPS receiver. The data is stored as a shape file and can be exported to any GIS software, such as SSToolbox or FarmWorks, to be analyzed and averaged based on plot design.
1.4.2 Field Experiments

1.4.2.1 Test Fields

The study was conducted at Clemson University’s Edisto Research and Education Center in Blackville, SC, USA, 34°17’19.2”N 79°44’37.7”W, elevation 38m. A 5-acre field located on the research station was used to conduct the various field experiments (Figure 1.6). The test field had a lateral irrigation (250 ft. long), and was only used to irrigate crop when conditions required irrigation to prevent yield losses (deficit irrigation).
Soil management zones were created in the test field based on the deep and shallow EC data. The field was divided into two EC zones (low and high) and 8-row by 60 ft. plots were established in each zone. The soil management zones are shown in Figures 1.7 and 1.8. Figure 1.7 shows the raw EC points with overlaid plots, and Figure 1.8 shows the EC data averaged by research plot.
Though there were two divisions based on soil EC, low and high, the NRCS soil survey map lined up well with the EC data (Figure 1.8) The test plots were arranged in a randomized block design, and 9 treatment rates (0, 20, 40, 80, 120, 160, 200, 240, 260 lbs./acre) of nitrogen were applied to develop the algorithm. The treatment rates were replicated 5 times and randomly applied to plots in each management zone. This layout was replicated during the 2016 growing season. Figure 1.9 shows the plot layout, with each rate indicating a nitrogen-rate in lbs./acre.

Figure 1.8. Average EC values in test plots.

Figure 1.9. Typical NRCS Soil Survey Map.
During the 2015 and 2016 growing seasons, Monsanto Dekalb GENVT2P DKC67-58 corn variety was planted. The crop was carried to yield using typically recommended practices for seeding, insect, and weed control. Irrigation was applied only during conditions where it was required to keep from incurring losses. Total rain plus irrigation was 23.85 inches and 21.33 inches during the 2015 and 2016 growing seasons, respectively. During each year of the study the corn was planted in the field during the time period between mid- to late March.

Nitrogen treatments were applied each season around mid-April. The corn was harvested at maturity using a John Deere 4 row combine retrofitted with an AgLeader yield monitor to create the yield map for all plots in the field.

1.4.2.2 Data Collection and Algorithm Development

The main objective of the 2015 growing season was to develop an algorithm for variable rate nitrogen application for corn production utilizing soil EC and NDVI under irrigated conditions. The corn was planted on March 16th, and the crop was carried to yield using typically recommended practices for seeding, insect, and pest control. Plant NDVI sensor readings were
collected during each growing season using a 6-row sprayer-mounted GreenSeeker® RT-200 mapping system. During the 2015 growing season, readings were taken from the test field 35, 39, 43, 46, 49, and 51 days after planting. For each sampling date, the sensor height from the canopy was measured to be between 30 and 36 inches from the top of the canopy.

In-season estimated yield (INSEY) was calculated by dividing NDVI measurements by the number of days from planting to sensing (Raun et al., 2005):

\[
INSEY = \frac{NDVI}{\text{# of days after planting}}
\]  

To determine the relationships between the actual corn yield and the INSEY, both linear and non-linear regression models were utilized to determine the yield potential \( (YP_0) \) and the N prediction algorithm. As the time after planting is a critical factor in determining the INSEY, the days where the Growing Degree Days (GDD) were equal to or less than zero were eliminated from this count. The GDD values were calculated as (Wiebold, 2002):

\[
GDD = \left[ \frac{T_{\text{min}} + T_{\text{max}}}{2} \right] - 50^\circ F
\]

The efficacy of the Clemson algorithm (developed during 2015 and 2016) was determined by comparing the sensor-based nitrogen application method to a typical grower’s practice (200 lbs. N/acre) under deficit irrigation corn production. The test was replicated 5 times in each zone. A “Nitrogen-Rich-Calibration-Strip (NRCS), pre-plant nitrogen rates where nitrogen will not be limiting throughout the season (260 lbs. N/acre), was established in each management zone of the test field. The Response Index (RI), the extent to which the crop will respond to additional nitrogen (Raun et al. 2005), was calculated by dividing the highest NDVI reading from the NRCS by NDVI measurements from plots of each management zones.
The predicted attainable yield \( (YP_N) \) with added nitrogen was calculated by multiplying \( YP_0 \) by RI (Raun et al. 2005).

\[
YP_N = RI \times YP_0
\]

The objective for the 2016 growing season was to further refine the algorithm for variable rate nitrogen application in corn utilizing soil EC and plant NDVI readings under deficit irrigation conditions. The corn for the 2016 growing season was planted on March 31st. During the 2016 growing season, the same 6-row sprayer-mounted GreenSeeker® RT-200 mapping system was utilized, readings were taken 34, 38, 40, 42, 45, 54, and 56 days after planting, the difference in sensing dates is due to weather and field conditions. For each sampling date the sensor height from the canopy was measured to be between 30 and 36 inches. During the 2016 growing season the plots were irrigated 5 times for a total of 1.05 inches, it also rained 7 times for a total of 6.32 inches.

Yield data was collected during the end of each growing season using an AgLeader Yield Monitor retrofitted on a John Deere 4 row combine. During the 2015 growing season the yield data was used to develop the N algorithm. The 2016 growing seasons yield data was used to further refine the developed algorithm.

The predicted attainable yield should not exceed the maximum corn yield \( (YP_{MAX}) \) for the given region and management practices. In this case the \( YP_{MAX} \) was set 11,200 lbs./ac for deficit irrigated field conditions in the Coastal Plain Region of South Carolina. The nitrogen fertilizer rate was then determined using equation 1.6:

\[
N_{Rate} = \frac{(YP_N-YP_0)+\%N}{NUE}
\]
Where YP₀ is yield prediction algorithm developed at during 2015 and 2016, %N is the percentage of nitrogen in crop seeds after harvest (1.3 % for corn), and NUE is the nitrogen use efficiency (60% for corn), Wiatrak et al., 2005.
1.5 RESULTS AND DISCUSSION

1.5.1 Algorithm Development and Refinement

Figures 1.11, 1.12, 1.13, 1.14, 1.15, 1.16 show the correlations between the in season estimated yield (INSEY) measured at 35, 39, 43, 46, 49, and 51 days after corn planting, respectively, and actual corn yields for the 2015 growing season. As days after planting increased, the correlations between the INSEY and actual corn yields became stronger (as evident by the coefficients of determinations) up to 49 days after planting. The coefficients of determinations decreased after 49 days.

![35 Days After Planting](image)

Figure 1.11. The correlation between INSEY and actual corn yields, 35 days after planting, 2015.
Figure 1.12. The correlation between INSEY and actual corn yields, 39 days after planting, 2015.

Figure 1.13. The correlation between INSEY and actual corn yields, 43 days after planting, 2015.
Figure 1.14. The correlation between INSEY and actual corn yields, 46 days after planting, 2015.

Figure 1.15. The collection between INSEY and actual corn yields, 49 days after planting, 2015.
Figure 1.16. The correlation between INSEY and actual corn yields, 51 days after planting, 2015.

NDVI values increased with days after planting for all sampling dates and all soil EC zones. Also NDVI values increased as nitrogen rates increase at the beginning of the growing season, and then leveled off for nitrogen rates higher than 140 lbs./acre. Management zone one (low soil EC values-blue line) had higher NDVI values until 43 days after planting where zone 2 (high EC values-red line) had the higher values.

Figure 1.17 shows the yield prediction equation developed from the 2015 growing season. The yield prediction algorithm for 2015 is given in equation 1.7. There was a high correlation (0.6914) between INSEY and actual corn yield.

\[
y = 3554.9e^{112.65x}
\]

Where \( x \) is INSEY and \( y \) is predicted corn yield \( (YP_0) \).
Figure 1.18 shows the effects of nitrogen rates on corn yields for two EC zones. The $R^2$ values increased significantly when the yield values were divided into the two predetermined soil EC zones. The results show that there is a potential to use mid-season site-specific side-dress nitrogen application in corn production. This also shows that soil EC (mS/m) needs to be included in the N-rate prediction equation.

$$y = 3554.9e^{112.65x}$$

$R^2 = 0.6914$
Figures 1.19, 1.20, 1.21, 1.22, 1.23, 1.24, and 1.25 show the correlations between the in-season estimated yield (INSEY) measured at 34, 38, 40, 42, 45, 54, and 56 days after corn planting, respectively, and actual corn yields for the 2016 growing season. As days after planning increased, the correlations between the INSEY and actual corn yields became stronger (as evident by the coefficients of determinations) up to 54 days after planting. The coefficients of determinations decreased after 54 days.
Figure 1.19. The correlation between INSEY and actual corn yields, 34 days after planting, 2016.

Figure 1.20. The correlation between INSEY and actual corn yields, 38 days after planting, 2016.
Figure 1.21. The correlation between INSEY and actual corn yields, 40 days after planting, 2016.

Figure 1.22. The correlation between INSEY and actual corn yields, 42 days after planting, 2016.
Figure 1.23. The correlation between INSEY and actual corn yields, 45 days after planting, 2016.

Figure 1.24. The correlation between INSEY and actual corn yields, 54 days after planting, 2016.
During the 2016 growing season, better results were attained; which would help to further refine the prediction algorithm. Figure 1.26 shows the yield prediction equation for the 2016 growing season (using combined data from 45 and 54 days after planting). This equation has a better potential to estimate mid-season nitrogen requirements, and will be utilized in further refinement of the prediction algorithm. The yield prediction algorithm for 2016 is given in equation 1.8:

$$y = 85.255e^{336.95x}$$

1.8

Where x is INSEY and y is predicted corn yield (YP₀).
Figure 1.26. 2016 yield prediction equation.

Figure 1.27 shows the algorithm developed using data from 2015 and 2016. The combined algorithm still have a good correlation ($R^2 = 0.6789$) between the INSEY values and actual corn yields, that could accurately predict the mid-season N requirements for corn production. Therefore, the final yield prediction equation (using two years data) for corn under deficit irrigation in Coastal Plain soils is:

$$y = 369.23e^{218.56x}$$

Where $x$ is INSEY and $y$ is predicted corn yield ($YP_0$).

$$y = 84.255e^{336.95x}$$

$R^2 = 0.7106$
Figures 1.28 and 1.29 confirm that soil EC data is a critical component for making accurate nitrogen recommendations for corn in the Coastal Plains. Figures 1.28 and 1.29 represent the 2015 and 2016 growing seasons respectively. During the 2015 season results showed that soil EC had a significant effect on corn yield in response to N. In management zone one, corn yield positively responded to nitrogen as rates increased. In management zone two, nitrogen rates above 160 lbs./ac, had no further effect on corn yield. Similar results were obtained in 2016.
Figure 1.28. Nitrogen response based on soil EC zones (2015).

Figure 1.29. Nitrogen response based on soil EC zones (2016).
The nitrogen prediction algorithm was developed using data from both the 2015 and 2016 growing seasons, and was developed for the accurate prediction of nitrogen application rates for corn production in the Coastal Plains Region.

1.5.2 Algorithm Testing

In the Coastal Plain Region of South Carolina, the typical farmer practice for corn under irrigation is to apply a flat rate of 200 lbs.-N/a. During 2016 growing season, the Clemson sensor-based algorithm was compared to a typical grower’s practice in terms of effects on corn yields and nitrogen requirements. The results (Figure 1.30) showed that the sensor-base nutrient management reduced nitrogen rates by 21% and 34% in zones one and two, respectively, compared to traditional farmer fixed rate practice without any reduction in corn yields (Figure 1.31). Applying side-dress nitrogen based on Clemson algorithm and the GreenSeeker optical sensor data, increased corn yields by 12% and 5% in zones one and two, respectively, compared to grower’s standard production method. However, the yield increases were not statistically significant. This technology has a potential to apply nitrogen where it is needed in a field at optimum rate, which will help our growers to reduce production costs, increase farm profits, while enhancing environmental quality. The algorithm should be further refined by testing over multiple growing seasons to increase its accuracy.
Figure 1.3. Clemson vs. Farmer Practice N Usage.

Figure 1.31. Clemson vs. Farmer Practice Yield.
1.6 CONCLUSIONS

During this study, a mid-season side-dress nitrogen prediction algorithm was developed for corn production under deficit irrigation. There were good positive correlations between the in-season estimated yield (INSEY) and the actual corn yields during both 2015 and 2016 growing seasons. Strong correlations were also found between plant nitrogen requirements and NDVI readings between the time period 45 and 56 days after planning, this is in the V-8 growth stage. The developed algorithms have the ability to accurately predict the mid-season side-dress nitrogen requirements for corn production under deficit irrigation in the Coastal Plain region of the southeastern United States.

The sensor-base nutrient management reduced nitrogen rates by 21% and 34% in zones one and two, respectively, compared to traditional farmer fixed rate practice without any reduction in corn yields. Applying side-dress nitrogen based on Clemson algorithm and the GreenSeeker optical sensor data, increased corn yields by 12% and 5% in zones one and two, respectively, compared to grower’s standard production method. However, the yield increases were not statistically significant. This technology has a potential to apply nitrogen where it is needed in a field at optimum rate, which will help our growers to reduce production costs, increase farm profits, while enhancing environmental quality. The algorithm should be further refined by testing over multiple growing seasons to increase its accuracy.
1.7 REFERENCES


CHAPTER TWO

DEVELOPMENT AND TESTING OF EQUIPMENT TO ACCURATELY APPLY SENSOR-BASED VARIABLE RATE NITROGEN LIQUID FERTILIZER IN CROP PRODUCTION

2.1 INTRODUCTION

Nutrient application systems are designed to apply a relatively uniform amount of fertilizer to agricultural fields. Currently most farmers apply uniform N rates across entire fields. Considerable variation occurs within across production fields in soil texture, soil type, and other major factors which affect crop production and will have a major impact on fertilizer management strategies. Therefore, uniform application of a fertilizer over the entire field can be both costly and environmentally unsound.

This study (Chapter one) showed that the sensor-based algorithm for mid-season side-dress nitrogen application, has the potential to reduce fertilizer rates significantly, without any negative effects on crop yield and farm income. However, currently there is no affordable variable-rate fertilizer applicator commercially available, which can be retrofitted on growers’ existing equipment. Typical nitrogen fertilizer applicators used by growers are usually a pull type system equipped with a ground driven crankshaft type piston pump. These types of pumps (such as John Blue) are widely used in agricultural settings because of their rugged design. These types of pumps have limitations to apply variable rate N due to the lack of system to be able to change the setting “on-the-go”.

40
Currently, there are over 2 million crankshaft-type positive-displacement piston pumps in the US, which are used by row-crop and hay farmers for applying crop inputs. The outlet flow of these pumps can be changed by adjusting pump stroke manually (stop and go), using specific tools provided by the company. The “on-the-go” outlet flow can only be varied by changing the drive shaft speed. However, for each manual setting, only limited range of flow rate can be achieved by changing the drive shaft speed. This limited flow range is not sufficient for applying variable-rate crop inputs in fields with tremendous amount of variations in soil types, resulting in practice that is wasteful, costly, and environmentally questionable. There is a need for a controller which can adjust the pump stroke on-the-go, for real-time, variable-rate application of crop inputs.

The developed algorithm, when coupled with optical sensors can accurately predict the mid-season nitrogen requirements for corn production. Optical sensors, such as GreenSeeker® (Trimble Navigation Limited Flows Division, Ukiah, CA), measure the plant Normalized Difference Vegetation Index (NDVI); which have been shown to be a good indicator of overall plant health (Raun et al. 2001). The GreenSeeker® readings are a good indicator of plant health because plants with more leaf area and chlorophyll absorb higher levels of red light and blue light. Therefore, healthy plants are able to reflect more NIR than less healthy plants due to turgid and healthy mesophyll cells. The ratio of the level of reflectance of red and NIR are highly useful when using NDVI as an indirect measure of plant health (Arnall 2008).
2.2 OBJECTIVES

The main objective of this study was to develop equipment for variable rate application of nitrogen fertilizer for crop production.

The specific objectives were to:

1. Develop a variable rate nitrogen application system that can be retrofitted onto growers’ existing fertilizer applicators.

2. Determine the effects of time of day on performance of the GreenSeeker® optical sensor.
2.3 REVIEW OF LITERATURE

The focus of this section is to review the literature related to the study objectives and consist of the two subheadings including:

1. Variable Rate Fertilizer Applicators
2. Factors Affecting the GreenSeeker® Performance

2.3.1 Variable Rate Fertilizer Applicators

Through traditional uniform N applications, in most cases, results in over- and under application of N in various parts of the field due to in-field spatial variability (Khosla et al. 1999). Precision agriculture techniques, such as the use of global positioning systems, and variable-rate application of pesticides and fertilizer, can improve crop yield and the economic efficiency of crop production (Nash et al., 2009). Improved efficiencies in corn production are of particular interest because corn is by most measures the most important crop in the United States (Li et al., 2001). Variable rate fertilization aims to improve fertilizer use efficiency and reduce leaching by varying fertilizer rates according to the needs of each area within a field (Yang, 2001). Presently, most growers in the USA utilize a ground driven crankshaft type piston pump (such as John Blue) for fertilizer application. These pumps are used widely in agricultural settings due to their rugged design. Development of an improved fertilizer management practice combine with affordable variable-rate equipment has the potential to increase fertilizer use efficiency and improve environmental quality (Mengel, 1990).

Many research projects have been conducted in the development of variable rate systems (Robert et al., 1991; Cahn et al., 1995; Yang, 2001, Yang et al. 2001), and several companies are currently marketing variable rate application equipment (Clark and McGuckin, 1996). While
many studies agree that optical sensors and algorithms have shown success in predicting optimal rates of nitrogen application, there are few systems that can automatically regulate application rates, and currently, there is system commercially available that can automatically change the stroke of a piston type pumps widely used in agriculture. Without this proper equipment, the development of a prescription map is useless for practical farmer applications.

The major components of a typical variable rate control system consist of an in-cab computer loaded with an application software and variable rate application maps, a global positioning system (GPS) receiver that provides vehicle position information to the computer, and a controller that controls material rates under direction of the computer (Yang, 2001). The equipment that is currently available needs to be refined and simplified so that the hardware and software is more operator-friendly and provides real-time and accurate variable rate nitrogen applications.

Several studies have been conducted to evaluate the performance of variable-rate control systems (Porter, 2010, Yang, 2001). Tests were conducted by Porter (2010) to determine the static and dynamic performance of a commercially available Rawson hydraulic control system for changing fertilizer rates. He reported that on average application errors were less than 1% for the static test, and significantly higher for the dynamic test (18%). The accurate delivery rate of the piston pump used occurs between the rotational speeds of 150 to 500 rpm for each stroke length setting. However, when the pump speed is outside of this range the application rate becomes skewed. He mentioned that the errors during the dynamic test was due to the above described pump limitations. He also reported that because of software lag time when using this controller, it lacked the accuracy for research and plot work, but it was likely acceptable for growers’ use. Tests conducted by Yang (2001) on a custom designed liquid knife nitrogen
applicator gave promising results in static testing, while dynamic test had an average rise time of 0.5 seconds. Most of these tests mentioned earlier, involved servo-valves; DC motor operated valves, centrifugal pumps, and electronically controlled flow valves, such as the Rawson Accu-Rate (Trimble Navigation Limited Flows Division, Ukiah, CA). These systems either did not provide satisfactory results or were too expensive for farm use.

Variable-rate, positive displacement piston pumps are widely used for metering chemicals with high level of repetitive accuracy and are capable of pumping a wide range of chemicals. These pumps have ability to vary capacity manually or automatically as process conditions require. Crankshaft type variable-rate piston pumps (such as John Blue) are used widely in agriculture, due to their rugged design. The “on-the-go” outlet flow of these pumps can only be varied by changing the drive shaft speed. The outlet flow also can be changed by adjusting pump stroke manually, using specific tools provided by the company. However, for each manual setting, only limited range of flow rate can be achieved by changing the drive shaft speed. This limited flow range is not sufficient for applying variable-rate crop inputs (such as nitrogen) in the Mid-South and the Southeastern USA, with tremendous amount of variations in field conditions and soil types. There is a need for a controller which can adjust the pump stroke on-the-go, for real-time, variable-rate application of crop inputs.

Ability to adjust the pump stroke on-the-go would make it possible to change the flow rate of a piston pump automatically from zero to pump’s full capacity. This will allow map-based application of variety of crop inputs in agriculture to match crop needs, or automatically controlling chemical rates in industries where variable dosing is controlled by computer, microprocessor, etc.
2.3.2 Factors Affecting the GreenSeeker® Performance

Researchers have been using spectral sensing in croplands for over two decades (Ramirez et al., 2010). The Trimble® GreenSeeker® crop sensing system is one of the most widely used active optical sensor to determine the mid-season nitrogen requirement for corn. Trimble states that the GreenSeeker® crop sensing system measures and quantifies the variability of the crop, and can add an additional $13.13 profit per acre when using to determine fertilizer prescriptions. They also state that the GreenSeeker® works in any weather condition, day or night, if operated at a sensing height of in-between 13 to 48 inches above the plant canopy. These factors should be evaluated under actual field conditions to improve the performance of the sensor (Trimble, 2016).

In a study conducted by Kim et al., (2010), the performance of an active spectral sensor was evaluated to study the effects of partial canopy coverage, target off-center, standoff distance, target surface tilting, solar bidirectional effect temperature and illumination, and diurnal radiation change. They reported that the acceptable range of leaf coverage was 30%-100%; with 30% leaf coverage the target was determined to be within 6 inches from the center; the acceptable standoff distance was determined to be 40-78 inches, which is well above the manufactures recommendations; an acceptable range of tilting angle was 0-50°; that there was no significant effect observed within 0-60° zenith angle; there was no significant effects due to changes in temperature; and that NDVI response was reduced when the solar radiation increased.

In a study conducted by Vellidis et al (2010), they reported that the optimal sensor height to be located between 30 and 48 inches above the plant canopy, which is at the upper limits of the manufactures recommendations. In tests conducted by Porter (2010), he concluded that the sensor would perform best from the height range of 30 to 36 inches above the crop canopy. With
the exception of Porter (2010), the majority of the studies were conducted under laboratory conditions and did not evaluate direction of travel, or time of day. Porter (2010) concluded that at the optimal sensor height, the sensor was not direction sensitive throughout the day between the hours of 10 a.m. and 8 p.m. (EST). The sensors returned a different number once the sun had set but the main reason for the deference is due to the physiological response of the cotton plants. It was found due to the response of the plant that it is not possible to obtain an accurate sensor reading at night.

This study was performed to evaluate the effects of time of day on the performance of the GreenSeeker® optical sensor for measuring NDVI of corn crop. There was no published data available in the literature related to GreenSeeker optical sensor’s responses due to time of the day for corn.
2.4 MATERIALS AND METHODS

2.4.1 Nitrogen Fertilizer Applicator

The crankshaft type variable-rate piston pump is widely used in agriculture; this is due to their rugged design. The outlet flows of these pumps can be changed by adjusting the pumps stroke length manually; this is done by using specific tools that are supplied by the manufacturer. There current method for providing “on-the-go” variability involves changing the drive speed of a hydraulic drive motor. The first commercially available system is the Rawson Accu-Rate control system; this system only changes the drive speed by changing the outlet flow by adjusting the speed of the pump by using an electronic flow control valve. This system can either be controlled manually by turning an electronic dial (rotary potentiometer) or map based (using GPS and a prescription map). With this method the operator still has to stop and adjust the pump setting in order to utilize the full range of the pump. The other system that is commercially available is a bolt on hydraulic motor sold by John-Blue. This motor eliminates the operator’s ability to adjust the stroke length and limits them to only the pump range of the setting that the pump is set at. When installing this motor the company sets the pump at setting 5. Figure 2.1 shows the pump curves that were collected during our testing. The limited flow range that is available at this setting is not sufficient for applying variable rate crop inputs in fields that have large amounts of spatial variation. This results in a practice that is wasteful, costly, and environmentally questionable.
The Clemson “Electro-Mechanical Controller for Adjusting Pump Stroke on-the-go” is designed to replace the current manual stroke adjustment system on a positive displacement piston pump, which makes it possible to change the flow rate of a piston pump automatically from zero to a pump’s full capacity. The Clemson system was developed on a John-Blue NGP 6055, this is a positive displacement variable stroke metering pump that was designed specifically for liquid fertilizer applications (Figure 2.2). This type of pump is widely used for liquid fertilizer application systems used in the agriculture industry because of its rugged and durable design.
Figure 2.2 Clemson System on John-Blue NGP 6055.

The platform for the Clemson system was the KBH 4 row liquid fertilizer applicator shown in Figure 2.3.

Figure 2.3 KBH Applicator.

The controller was designed to control both the hydraulic drive system and the electro-mechanical stroke adjustment system, and is shown in Figure 2.4. The controller utilized several different components, a Phidgets I/O board, and a Phidgets Dual Relay Board, also included was a micro USB power supply for our tablet that was used to run our control program; also required is a windows based laptop or other windows device with the application program downloaded on
During the testing of the applicator a hydraulic drive system was also developed. The wiring schematic of the controller is shown in Figure 2.6.

This system is manually controlled by a dial (rotary potentiometer) and only controls the electro mechanical control system for adjusting the pump stroke on-the-go. This system utilizes several sensors, control boards, and mechanical features. For the physical system a Phidgets I/O board and dual relay board are used. The I/O board reads the signals coming from a linear potentiometer and the rotary potentiometer. The signal from the linear potentiometer is used to determine the linear actuators position and therefore the pump's current setting. The signal from the rotary potentiometer is used to determine what setting the operator want the pump to be set at. If the rotary potentiometer is turned clockwise, which indicates a desired increase in pump setting, the control program (Appendix A) then tells the I/O board sets the output channel 0 to true, which is sent to the relay board applying a positive voltage to the linear actuator, thus extending the actuator until it is in the desired location in which time the I/O board returns the output channel 0 to false, which turns off the relay. When the rotary potentiometer is turned counterclockwise, indicating a desired decrease in pump setting, the program will then sets the I/O board to output channel 1 to true, which is sent to the relay board to reverse the polarity, retracting the actuator until it has reached the desired pump setting, then returns the output channel 1 to false, thus turning off the relay. A flow chart for this program can be seen in figure 2.7.
Figure 2.4 Controller for Nitrogen Applicator.

Figure 2.5. Clemson Variable Nitrogen Applicator Control Program.

Figure 2.6 Controller Wiring Diagram.
To test the new application system, we also developed a hydraulically controlled system to control the drive speed of the pump so that indoor trials could be conducted. The hydraulic system (Figure 2.8) consisted of a Brand Hydraulics Electrical Adjustable Proportional Pressure Compensated Flow Control Valve (EFC), a Brand Hydraulics Sealed Interface Control System (EC), and a Dynamic Low Speed High Torque hydraulic motor. The EC requires a 12 VDC voltage supply, and a 0-5 V input that is converted into a Pulse Width Modulated (PWM) output that is suitable for the Brand EFC valve. The combination of the EC and EFC gave the ability to control the drive speed of the pump from 0 to 500 rpm.
Figure 2.8. The side-dress Nitrogen Applicator retrofitted with hydraulic system.

To provide the completely automated option a map based system was developed. This system allows the user to download a pre-made nitrogen prescription map, and then begin the application. As the user drives through the field, the program reads what nitrogen rate needs to be applied and then adjusts the pump setting accordingly.

This system is automatically controlled, and does not currently allow for manual override (Figure 2.9). This program begins by downloading a map that is set as the background for the GPS’s current location. This map is not used for determining the current location, only to show through the user selected map background of choice where the GPS says they are. As the program initializes, the map background is loaded, and the user chooses the com port that the GPS unit is being brought in through, the user also has the ability to adjust the Baud Rate. Once the GPS is attached, the program shows the user that it is attached, what the current Latitude and Longitude coordinates that they are currently positioned at, the quality of the signal, how many satellites have been acquired, the current altitude, the degree heading, and the offset (in feet) of the signal. To begin the application process, the user will then open the shape file containing the currents fields NDVI readings (such as the one showed in Figure 2.10), and then select the proper file heading (attribute) that contains the NDVI readings. Once this is complete, and as the
user begins to drive through the field, the program will read the current NDVI reading at that location and utilize the Clemson N prediction algorithm to determine the required amount of nitrogen. The determined output rate is then shown on screen as output in lbs-N/ac. The program code can be seen in Appendix B.

Figure 2.9. Automatic Control Flow Chart.

Figure 2.10. NDVI Shape File Example.
For testing the map based program a hydraulic drive system was developed to attach to the current John Blue system that had been retrofitted with the Clemson system. The control program was modified to show pump setting and speed of the hydraulic motor as well as a function to override the GPS signal and to determine GPS coordinates based on the current cursors position. Once the NDVI shape file was loaded, the cursor was moved over an area of the field containing and NDVI value. Using our algorithm, proper nitrogen rates were calculated for that particular location in the field. The program will then adjust pump setting to deliver the desired nitrogen rate in lbs-N/ac. If the required amount of Nitrogen fell within that specific settings recommended range, then the speed was adjusted to achieve the required output. If the required rate fell outside of that range of the pump setting, then the setting was changed automatically to achieve higher or lower rates. Therefore, by changing the combination of pump setting and pump rotational speed, the system delivered from zero to pump full capacity. As it is easier to adjust the pump speed, priority was given to the pump setting.

2.4.2 Normalized Difference Vegetation Index (NDVI)

To collect NDVI data from the test field, the GreenSeeker RT-200 six-row NDVI system (Figure 2.11) was used. Figure 2.12 shows an aerial photo of the variations in the test field during the 2016 growing season. During the same time period every 3-5 days, this system was used to collect NDVI data. Figure 2.13 shows one of the GreenSeeker® sensors mounted to the boom of a John Deere 6700 Hi-Boy sprayer. Figure 2.14 shows the Trimble® Nomad handheld computer used to collect NDVI data.
Figure 2.11. John Deere 6700 with NTech GreenSeeker® RT-200 system.

Figure 2.12. Aerial photo of plots.
The second test to be conducted was the time of day test. To ensure the accuracy of this test one sensor was placed 36 inches from the top of the canopy. The height was chosen based on Porter (2010). He reported that this height would return more uniform readings from the sensors. A second sensor was placed beside the first sensor, and was placed over a piece of green cloth. Both sensors were then allowed to collect NDVI data for a 24 hour period (Figure 2.15). Both sensors collected data on a 1 Hz signal for the testing period. The data was then exported to be analyzed using Microsoft Excel. The hourly readings were then averaged and compared.
Figure 2.15. GreenSeeker® data collection over corn and green cloth.
2.5 RESULTS AND DISCUSSION

2.5.1 Nitrogen Fertilizer Applicator

Static calibration tests were conducted to determine if the linear actuator was traveling the proper distance to achieve the proper pump setting. Travel distance was measured with a linear potentiometer and calipers. The reading from the linear potentiometer was then used so that the program could determine the current pump setting and which direction it would need to travel to reach the required setting. This was set in relation to a rotary potentiometer to change the pump setting manually.

The results of the static calibration test (Figure 2.16) showed an excellent correlation (R² = 0.9986) between required travel distance of the linear actuator and actual travel distance. The small variations in the actual travel distances (< 0.05 inches) from the required distance are due to the programming. Because of the accuracy of the linear potentiometer, it is impossible to set the dials position equal to the linear potentiometers position. Thus, a travel range was set for each setting, causing the slight variations from the required distance. The static test proved to be very accurate with minimal error.
John Blue piston pump’s application rate is directly related to its rotational speed (rpm) and the piston stroke length which could be manually controlled by operator. The designed system allows for continuous “on-the-go” control of the pumps stroke length. The program was written to only change when a change in the rotary potentiometers position is changed.

### 2.5.2 Factors Affecting the GreenSeeker® Performance

The first test performed was the “Time of Day Test”. Figure 2.17 represents GreenSeeker® sensor readings taken over a 24 hour period. The sensors returned higher readings between 8 pm and 6 am, and slight spikes between the hours of 10 and 11 am, and again at 4 pm. While there were slight differences in readings during the daylight hours, this can be attributed to the vigorous photosynthesis process that takes place in corn during these hours. Overall this test concludes that the sensors are not sensitive to angle of the sun and can be accurately used between three hours after sunrise and one hour before sunset.
Figure 2.17. GreenSeeker® sensor readings throughout the day.

Figure 2.18 shows a comparison between GreenSeeker® readings over corn and over green cloth. The results show that the sensor over the green cloth was relatively flat. This shows that sensor readings are not affected by time of day. Though the more NIR light is reflected from plants, the green cloth had higher readings than the corn because of the color of the cloth.
The GreenSeeker® readings are a good indicator of plant health because plants with more leaf area and chlorophyll absorb higher levels of red light and blue light. Therefore, healthy plants are able to reflect more NIR than less healthy plants due to turgid and healthy mesophyll cells. The ratio of the level of reflectance of red and NIR are highly useful when using NDVI as an indirect measure of plant health (Arnall 2008). Simply put, the higher the NIR reflectance combined with a low visible reflectance means the greener the plant, meaning the healthier the plant; and the lower the NIR reflectance combined with a high visible reflectance the yellower the plant and therefore the less healthy the plant is.

Figure 2.19 and 2.20 show NDVI compared to temperature and solar radiation data during the testing period respectively. No correlation was found between NDVI reading and temperature or solar radiation. The sensors perform acceptably during the day, with exception to the early morning hours.
The GreenSeeker® is an accurate sensor for collecting NDVI data that can be used as an index to estimate various vegetation properties including chlorophyll concentration in leaves, leaf area index, biomass, plant vigor, plant productivity, nutrient requirements, and stress (Yoon 2009). When the sensor is used at the optimal height and sensor readings are taken between 3 hours after sunrise and one hour before sun, it can accurately measure NDVI values, which can be used to predict in-season estimated yield (INSEY) in corn, and calculate optimum nitrogen requirements.
2.6 CONCLUSIONS

A “on-the-go” variable rate nitrogen application system that can be retrofitted to a farmers existing nitrogen application system was developed throughout the course of this study. Two versions of this system was developed, the first, was a user controlled manual system utilizing an electric dial located inside a tractor cab; the second, was an automated map-based control system. The controller was connected to a windows based laptop, which was designed to operate either system. Our design utilizes a closed-loop (feedback) control system for real-time adjustment of the piston pump stroke and could be retrofitted on any existing piston pump. The system can be controlled using a pre-described position sequences or real-time sensor-based commands and can be connected to computers, micro-controllers, or PLCs (Programmable Logic Controllers). In addition, the system can adjust pump stroke manually, using a pre-calibrated dial. The controller can communicate with GPS and any GIS software such as "Farm Site Mate" (Farm Works Software LTD) for precise and map-based application of products.

The GreenSeeker® sensor was tested for effects in performance under various sun angles, temperatures, solar radiation, and time of day. Results showed that the angle of the sun, temperature, and solar radiation did not have any significant effect on the performance of the sensor. The results of the time of day test showed to have effects on the sensor readings, which can be related to the plants physiological effects during daylight hours. Based on the results from the NDVI collected during the 24 hour period, it was found that the time period between three hours after sunrise and one hour before sunset was the optimal time to collect NDVI readings. The test performed over the green cloth showed that there were no significant differences between readings collected during the daylight and night hours. The difference
between the sensor over the crop and the sensor over the green cloth is associated with the
differences in the plants physiological effects during the day and night.
2.7 REFERENCES


APPENDICES
Appendix A

Visual Basic Code for Manual Control

Public Class Form1
    Dim WithEvents phidgetIFK As Phidgets.InterfaceKit
    Dim File_Name As String = "C:\Users\Nick\Documents\Visual Studio 2010\Projects\DialBlackv2.0\DialBlackv2.0\bin\Debug\Help.txt"
    Public Sub New()
        ' This call is required by the Windows Form Designer.
        InitializeComponent()
        ' Add any initialization after the InitializeComponent() call.
    End Sub

    'initialize the device
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'To reduce code complexity we assume that there is one PhidgetInterfacekit
        'attached to the PC before the program is run.
        Try
            phidgetIFK = New Phidgets.InterfaceKit
            phidgetIFK.open()
        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try
    End Sub

    'attach event handler... here we'll display the interface kit details as well as
    'fields to display as well as determine the range of values for the output simulator
    'slider
    Private Sub phidgetIFK_Attach(ByVal sender As Object, ByVal e As Phidgets.Events.AttachEventArgs) Handles phidgetIFK.Attach
        attachedTxt.Text = phidgetIFK.Attached.ToString()
        nameTxt.Text = phidgetIFK.Name
        serialTxt.Text = sender.SerialNumber.ToString()
        versionTxt.Text = sender.Version.ToString()
        digiInNumTxt.Text = phidgetIFK.inputs.Count.ToString()
        digiOutNumTxt.Text = sender.outputs.Count.ToString()
        sensorInNumTxt.Text = sender.sensors.Count.ToString()

        Try
            Dim i As Integer
            For i = 0 To phidgetIFK.sensors.Count - 1
                phidgetIFK.sensors(i).Sensitivity = 10
            Next
        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString())
        End Try
    End Sub
'ifkit detach event handler... here we display the status, which will be false as the
device is not attached. We
'will also clear the display fields and hide the inputs and outputs.
Private Sub phidgetIFK_Detach(ByVal sender As Object, ByVal e As
Phidgets.Events.DetachEventArgs) Handles phidgetIFK.Detach
attachedTxt.Text = phidgetIFK.Attached.ToString()
nameTxt.Text = ""
serialTxt.Text = ""
versionTxt.Text = ""
digiInNumTxt.Text = ""
digiOutNumTxt.Text = ""
sensorInNumTxt.Text = ""
stopActuator() End Sub

Private Sub phidgetIFK_Error(ByVal sender As Object, ByVal e As
Phidgets.Events.ErrorEventArgs) Handles phidgetIFK.Error
MessageBox.Show(e.Description)
End Sub

Private Sub phidgetIFK_SensorChange(ByVal sender As Object, ByVal e As
Phidgets.Events.SensorChangeEventArgs) Handles phidgetIFK.SensorChange
Dim OnSensorChange As Boolean
Dim SensorChangeTrigger As Double
Dim SensorValue As Integer
Dim OnSensorChange1 As Boolean
Dim SensorChangeTrigger1 As Double
Dim SensorValue1 As Integer
Dim Dial As Decimal
Dim Position As Decimal
SensorValue = phidgetIFK.sensors(0).Value
SensorChangeTrigger = 0
OnSensorChange = (phidgetIFK.sensors(0).Value)

SensorValue1 = phidgetIFK.sensors(1).Value
SensorChangeTrigger1 = 0
OnSensorChange1 = (phidgetIFK.sensors(1).Value)

CheckDial()
CheckPosition()
stopActuator()

Dial = txtDial.Text
Position = txtPosition.Text
'Dial = phidgetIFK.sensors(0).Value
'Position = phidgetIFK.sensors(1).Value

If Dial > Position Then
extendActuator()
End If
If Dial < Position Then
retractActuator()
End If
If Dial = Position Then
    stopActuator()
End If

End Sub

'When the application is terminating, close the Phidget.
Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    RemoveHandler phidgetIFK.Attach, AddressOf phidgetIFK_Attach
    RemoveHandler phidgetIFK.Detach, AddressOf phidgetIFK_Detach
    RemoveHandler phidgetIFK.Error, AddressOf phidgetIFK_Error
    RemoveHandler phidgetIFK.SensorChange, AddressOf phidgetIFK_SensorChange
    Application.DoEvents()
End Sub

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHelp.Click
    If System.IO.File.Exists(File_Name) = True Then
        Dim objReader As New System.IO.StreamReader(File_Name)
        MessageBox.Show(objReader.ReadToEnd)
        objReader.Close()
    Else
        MessageBox.Show("No Help Exist's")
    End If
End Sub

Private Sub extendActuator()
    Dim OutputState As Boolean
    OutputState = phidgetIFK.outputs(0)
    OutputState = True
    phidgetIFK.outputs(0) = OutputState
End Sub

Private Sub retractActuator()
    Dim OutputState As Boolean
    OutputState = phidgetIFK.outputs(1)
    OutputState = True
    phidgetIFK.outputs(1) = OutputState
End Sub

Private Sub stopActuator()
    Dim OutputState As Boolean
    Dim OutputState1 As Boolean
    OutputState = phidgetIFK.outputs(0)
    OutputState = False
    phidgetIFK.outputs(0) = OutputState
    OutputState1 = phidgetIFK.outputs(1)
    OutputState1 = False
    phidgetIFK.outputs(1) = OutputState1
End Sub

Private Sub CheckDial()
Dim Dial As Decimal
Dial = phidgetIFK.sensors(0).Value

If (0 <= Dial) AndAlso (Dial <= 29) Then
    txtDial.Text = "1"
ElseIf (30 <= Dial) AndAlso (Dial <= 59) Then
    txtDial.Text = "2"
ElseIf (60 <= Dial) AndAlso (Dial <= 89) Then
    txtDial.Text = "2.25"
ElseIf (90 <= Dial) AndAlso (Dial <= 119) Then
    txtDial.Text = "2.5"
ElseIf (120 <= Dial) AndAlso (Dial <= 149) Then
    txtDial.Text = "2.75"
ElseIf (150 <= Dial) AndAlso (Dial <= 179) Then
    txtDial.Text = "3"
ElseIf (180 <= Dial) AndAlso (Dial <= 209) Then
    txtDial.Text = "3.25"
ElseIf (210 <= Dial) AndAlso (Dial <= 239) Then
    txtDial.Text = "3.5"
ElseIf (240 <= Dial) AndAlso (Dial <= 269) Then
    txtDial.Text = "3.75"
ElseIf (270 <= Dial) AndAlso (Dial <= 299) Then
    txtDial.Text = "4"
ElseIf (300 <= Dial) AndAlso (Dial <= 329) Then
    txtDial.Text = "4.25"
ElseIf (330 <= Dial) AndAlso (Dial <= 359) Then
    txtDial.Text = "4.5"
ElseIf (360 <= Dial) AndAlso (Dial <= 389) Then
    txtDial.Text = "4.75"
ElseIf (390 <= Dial) AndAlso (Dial <= 419) Then
    txtDial.Text = "5"
ElseIf (420 <= Dial) AndAlso (Dial <= 449) Then
    txtDial.Text = "5.25"
ElseIf (450 <= Dial) AndAlso (Dial <= 479) Then
    txtDial.Text = "5.5"
ElseIf (480 <= Dial) AndAlso (Dial <= 509) Then
    txtDial.Text = "5.75"
ElseIf (510 <= Dial) AndAlso (Dial <= 539) Then
    txtDial.Text = "6"
ElseIf (540 <= Dial) AndAlso (Dial <= 569) Then
    txtDial.Text = "6.25"
ElseIf (570 <= Dial) AndAlso (Dial <= 599) Then
    txtDial.Text = "6.5"
ElseIf (600 <= Dial) AndAlso (Dial <= 629) Then
    txtDial.Text = "6.75"
ElseIf (630 <= Dial) AndAlso (Dial <= 659) Then
    txtDial.Text = "7"
ElseIf (660 <= Dial) AndAlso (Dial <= 689) Then
    txtDial.Text = "7.25"
ElseIf (690 <= Dial) AndAlso (Dial <= 719) Then
    txtDial.Text = "7.5"
ElseIf (720 <= Dial) AndAlso (Dial <= 749) Then
    txtDial.Text = "7.75"
ElseIf (750 <= Dial) AndAlso (Dial <= 779) Then
    txtDial.Text = "8"
ElseIf (780 <= Dial) AndAlso (Dial <= 809) Then
    txtDial.Text = "8.25"
ElseIf (810 <= Dial) AndAlso (Dial <= 839) Then
    txtDial.Text = "8.5"
txtDial.Text = "8.5"
ElseIf (840 <= Dial) AndAlso (Dial <= 869) Then
    txtDial.Text = "8.75"
ElseIf (870 <= Dial) AndAlso (Dial <= 899) Then
    txtDial.Text = "9"
ElseIf (900 <= Dial) AndAlso (Dial <= 929) Then
    txtDial.Text = "9.25"
ElseIf (930 <= Dial) AndAlso (Dial <= 959) Then
    txtDial.Text = "9.5"
ElseIf (960 <= Dial) AndAlso (Dial <= 989) Then
    txtDial.Text = "9.75"
ElseIf (990 <= Dial) AndAlso (Dial <= 1000) Then
    txtDial.Text = "10"
Else
    MessageBox.Show("Dial Error")
End If

End Sub

Private Sub CheckPosition()
    Dim Position As Decimal
    Position = phidgetIFK.sensors(1).Value
    If (0 <= Position) AndAlso (Position <= 18) Then
        txtPosition.Text = "1"
    ElseIf (19 <= Position) AndAlso (Position <= 37) Then
        txtPosition.Text = "2"
    ElseIf (38 <= Position) AndAlso (Position <= 56) Then
        txtPosition.Text = "2.25"
    ElseIf (57 <= Position) AndAlso (Position <= 75) Then
        txtPosition.Text = "2.5"
    ElseIf (76 <= Position) AndAlso (Position <= 94) Then
        txtPosition.Text = "2.75"
    ElseIf (95 <= Position) AndAlso (Position <= 113) Then
        txtPosition.Text = "3"
    ElseIf (114 <= Position) AndAlso (Position <= 132) Then
        txtPosition.Text = "3.25"
    ElseIf (133 <= Position) AndAlso (Position <= 151) Then
        txtPosition.Text = "3.5"
    ElseIf (152 <= Position) AndAlso (Position <= 170) Then
        txtPosition.Text = "3.75"
    ElseIf (171 <= Position) AndAlso (Position <= 189) Then
        txtPosition.Text = "4"
    ElseIf (190 <= Position) AndAlso (Position <= 208) Then
        txtPosition.Text = "4.25"
    ElseIf (209 <= Position) AndAlso (Position <= 227) Then
        txtPosition.Text = "4.5"
    ElseIf (228 <= Position) AndAlso (Position <= 246) Then
        txtPosition.Text = "4.75"
    ElseIf (247 <= Position) AndAlso (Position <= 265) Then
        txtPosition.Text = "5"
    ElseIf (266 <= Position) AndAlso (Position <= 284) Then
        txtPosition.Text = "5.25"
    ElseIf (285 <= Position) AndAlso (Position <= 303) Then
        txtPosition.Text = "5.5"
    ElseIf (304 <= Position) AndAlso (Position <= 322) Then
        txtPosition.Text = "5.75"
    ElseIf (323 <= Position) AndAlso (Position <= 341) Then
        txtPosition.Text = "6"
ElseIf (342 <= Position) AndAlso (Position <= 360) Then
    txtPosition.Text = "6.25"
ElseIf (361 <= Position) AndAlso (Position <= 379) Then
    txtPosition.Text = "6.5"
ElseIf (380 <= Position) AndAlso (Position <= 398) Then
    txtPosition.Text = "6.75"
ElseIf (399 <= Position) AndAlso (Position <= 417) Then
    txtPosition.Text = "7"
ElseIf (418 <= Position) AndAlso (Position <= 436) Then
    txtPosition.Text = "7.25"
ElseIf (437 <= Position) AndAlso (Position <= 455) Then
    txtPosition.Text = "7.5"
ElseIf (456 <= Position) AndAlso (Position <= 474) Then
    txtPosition.Text = "7.75"
ElseIf (475 <= Position) AndAlso (Position <= 493) Then
    txtPosition.Text = "8"
ElseIf (494 <= Position) AndAlso (Position <= 512) Then
    txtPosition.Text = "8.25"
ElseIf (513 <= Position) AndAlso (Position <= 531) Then
    txtPosition.Text = "8.5"
ElseIf (532 <= Position) AndAlso (Position <= 550) Then
    txtPosition.Text = "8.75"
ElseIf (551 <= Position) AndAlso (Position <= 569) Then
    txtPosition.Text = "9"
ElseIf (570 <= Position) AndAlso (Position <= 588) Then
    txtPosition.Text = "9.25"
ElseIf (589 <= Position) AndAlso (Position <= 607) Then
    txtPosition.Text = "9.5"
ElseIf (608 <= Position) AndAlso (Position <= 626) Then
    txtPosition.Text = "9.75"
ElseIf (627 <= Position) AndAlso (Position <= 645) Then
    txtPosition.Text = "10"
Else
    MessageBox.Show("Position Error")
End If

End Sub

End Class
Appendix B

Visual Basic Code for Map Based Control

Imports GMap.NET.WindowsForms
Imports GMap.NET.WindowsForms.Markers
Imports GMap.NET
Imports ArcViewShapeFileDLL
Imports Phidgets
Imports System.Windows.Forms
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports
Imports System.Collections.Generic
Imports System.Data
Imports System.Drawing
Imports System.Text

Public Class Form1
    Dim WithEvents phidg
    'Dim WithEvents GPS As Phidgets.GPS
    'Need to add trimble gps that will be read by com port
    Public Overridable Property MaxLength As Integer
    Dim markersoverlay As GMapOverlay = New GMapOverlay("markers")
    Dim marker As GMarkerGoogle

    Dim ShapeFolder As String
    Dim strShapeFileName As String

    Dim buf As String = ""
    Dim LatMin As Double = Nothing
    Dim LatMax As Double = Nothing
    Dim LongMin As Double = Nothing
    Dim LongMax As Double = Nothing

    Dim ShapeIn As New ArcViewShapeFileDLL.ShapeFiles
    Dim Layer_Polygon As New GMapOverlay
    Dim polygonPoints As New List(Of PointLatLng)()
    Dim myTestPolygon As New List(Of GMapPolygon)()
    Dim TotalRecords As Integer

    Dim myPort As Array 'COM Ports detected on the system will be stored here
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent threading errors during receiving of data
    Dim GPSstring As String

    Dim strLatitude As String
    Dim strLongitude As String
    Dim lastHeading As Single

    Dim File_Name As String = "C:\Users\Nick\Documents\Visual Studio 2010\Projects\MapDial\MapDialm\Help.txt"
'Need to look for better way to have a help file with a better file path name

Public Sub New()
    ' This call is required by the Windows Form Designer.
    InitializeComponent()
    ' Add any initialization after the InitializeComponent() call.
End Sub

Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    RemoveHandler phidgetIFK.Attach, AddressOf phidgetIFK_Attach
    RemoveHandler phidgetIFK.Detach, AddressOf phidgetIFK_Detach
    RemoveHandler phidgetIFK.Error, AddressOf phidgetIFK_Error
    RemoveHandler phidgetIFK.SensorChange, AddressOf phidgetIFK_SensorChange
    Application.DoEvents()

    phidgetIFK.close()
    SerialPort1.Close()    'Close our Serial Port
    tmrReadSerial.Enabled = False
    Me.Close()
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    myMap.Position = New PointLatLng(33.364449, -81.329485)
    myMap.MapProvider = MapProviders.BingHybridMapProvider.Instance
    myMap.MinZoom = 1
    myMap.MaxZoom = 20
    myMap.Zoom = 18
    myMap.Manager.Mode = AccessMode.ServerAndCache
    trkZoom.Minimum = myMap.MinZoom
    trkZoom.Maximum = myMap.MaxZoom
    trkZoom.Value = myMap.Zoom
    cboMapProvider.Items.Add("Bing Satellite")
    cboMapProvider.Items.Add("Bing Hybrid")
    cboMapProvider.Items.Add("Mapquest Hybrid")
    cboMapProvider.Text = "Bing Hybrid"

    'Me.WindowState = FormWindowState.Maximized
    Me.WindowState = FormWindowState.Normal

    'need to rename cmbBaud boxes
    myPort = IO.Ports.SerialPort.GetPortNames() 'Get all com ports available
    cmbBaudGPS.Items.Add(4800)    'Populate the cmbBaud Combo box to common baud rates used
    cmbBaudGPS.Items.Add(9600)    'Populate the cmbBaud Combo box to common baud rates used
    cmbBaudGPS.Items.Add(19200)
    cmbBaudGPS.Items.Add(38400)
    cmbBaudGPS.Items.Add(57600)
    cmbBaudGPS.Items.Add(115200)

    For i = 0 To UBound(myPort)
        cmbPortGPS.Items.Add(myPort(i))
Next
cmbPortGPS.Text = cmbPortGPS.Items.Item(0)  'Set cmbPort text to the first COM port detected
cmbBaudGPS.Text = cmbBaudGPS.Items.Item(0)  'Set cmbBaud text to the first Baud rate on the list
btnDisconnect.Enabled = False
cboHeaders.Enabled = False
chkGPSOverride.Visible = False

Try
  phidgetIFK = New Phidgets.InterfaceKit
  phidgetIFK.open()
Catch ex As Exception
  MessageBox.Show(ex.ToString())
End Try
End Sub

Private Sub phidgetIFK_Attach(ByVal sender As Object, ByVal e As Phidgets.Events.AttachEventArgs) Handles phidgetIFK.Attach
  attachedTxt.Text = phidgetIFK.Attached.ToString()
nametxt.Text = phidgetIFK.Name
serialTxt.Text = sender.SerialNumber.ToString()
versionTxt.Text = sender.Version.ToString()
digiInNumTxt.Text = phidgetIFK.inputs.Count.ToString()
digiOutNumTxt.Text = sender.outputs.Count.ToString()
sensorInNumTxt.Text = sender.sensors.Count.ToString()

Try
  Dim i As Integer
  For i = 0 To phidgetIFK.sensors.Count - 1
    phidgetIFK.sensors(i).Sensitivity = 10
  Next
Catch ex As Exception
  MessageBox.Show(ex.Message.ToString())
End Try
End Sub

'Ifkit detach event handler... here we display the status, which will be false as the device is not attached.  We
'will also clear the display fields and hide the inputs and outputs.
Private Sub phidgetIFK_Detach(ByVal sender As Object, ByVal e As Phidgets.Events.DetachEventArgs) Handles phidgetIFK.Detach
  attachedTxt.Text = phidgetIFK.Attached.ToString()
nametxt.Text = ""
serialTxt.Text = ""
versionTxt.Text = ""
digiInNumTxt.Text = ""
digiOutNumTxt.Text = ""
sensorInNumTxt.Text = ""
stopActuator()
Private Sub phidgetIFK_Error(ByVal sender As Object, ByVal e As Phidgets.Events.ErrorEventArgs) Handles phidgetIFK.Error
    MessageBox.Show(e.Description)
End Sub

Private Sub phidgetIFK_SensorChange(ByVal sender As Object, ByVal e As Phidgets.Events.SensorChangeEventArgs) Handles phidgetIFK.SensorChange
    Dim OnSensorChange As Boolean
    Dim SensorChangeTrigger As Double
    Dim SensorValue As Integer
    Dim OnSensorChange1 As Boolean
    Dim SensorChangeTrigger1 As Double
    Dim SensorValue1 As Integer
    Dim Dial As Decimal
    Dim Position As Decimal

    SensorValue = phidgetIFK.sensors(0).Value
    SensorChangeTrigger = 0
    OnSensorChange = (phidgetIFK.sensors(0).Value)
    SensorValue1 = phidgetIFK.sensors(1).Value
    SensorChangeTrigger1 = 0
    OnSensorChange1 = (phidgetIFK.sensors(1).Value)

    CheckPosition()
    stopActuator()

    Dial = txtDial.Text
    Position = txtPosition.Text
    'Dial = phidgetIFK.sensors(0).Value
    'Position = phidgetIFK.sensors(1).Value

    If Dial > Position Then
        extendActuator()
    End If
    If Dial < Position Then
        retractActuator()
    End If
    If Dial = Position Then
        stopActuator()
    End If
End Sub

Private Sub btnZoomIn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnzoomin.Click
    Try
        myMap.Zoom += 1
    Catch ex As Exception
    End Try
End Sub

Private Sub btnZoomOut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnzoomout.Click
    Try
    End Try
End Sub
myMap.Zoom += -1
Catch ex As Exception
End Try
End Sub

Private Sub trkZoom_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles trkZoom.Scroll
    myMap.Zoom = trkZoom.Value
End Sub

Private Sub cboMapProvider_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboMapProvider.SelectedIndexChanged
    Select Case cboMapProvider.Text
        Case "Bing Satellite"
            myMap.MapProvider = MapProviders.BingSatelliteMapProvider.Instance
        Case "Bing Hybrid"
            myMap.MapProvider = MapProviders.BingHybridMapProvider.Instance
        Case "Mapquest Hybrid"
            myMap.MapProvider = MapProviders.OpenStreetMapQuestHybridProvider.Instance
    End Select
End Sub

Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnConnect.Click
    Try
        SerialPort1.PortName = cmbPortGPS.Text
        'Set SerialPort1 to the selected COM port at startup
        SerialPort1.BaudRate = cmbBaudGPS.Text
        'Set baud rate to the selected value on

        'Other Serial Port Property
        SerialPort1.Parity = IO.Ports.Parity.None
        SerialPort1.StopBits = IO.Ports.StopBits.One
        SerialPort1.DataBits = 8
        'Open our serial port
        SerialPort1.Open()

        tmrReadSerial.Enabled = True
        txtConnecting.Visible = True
        btnConnect.Enabled = False
        'Disable Connect button
        btnDisconnect.Enabled = True
        'and Enable Disconnect button
        Catch
            MessageBox.Show("No GPS Available")
        End Try
    End Sub

Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisconnect.Click
    SerialPort1.Close()
    'Close our Serial Port
    tmrReadSerial.Enabled = False
    btnConnect.Enabled = True
    btnDisconnect.Enabled = False
End Sub

Private Sub ReceivedText(ByVal [text] As String)
    'This is based off of NMEA standards
'compares the ID of the creating Thread to the ID of the calling Thread
Dim boolConnected As Boolean = False

If [text] = "" Then Exit Sub

'The following advances timer if feed is not at beginning of sentence, as
indicated by dollar sign
If [text].Substring(0, 1) <> "$" Then
tmrReadSerial.Interval = tmrReadSerial.Interval + 10
txtConnecting.Visible = True
Else 'if feed IS at beginning of sentence, then it will catch (note, this is
written for 1 Hz input rate)
tmrReadSerial.Interval = 1000
txtConnecting.Visible = False
End If

'Splits feed on $ signs
Dim lines As String() = [text].Split(New Char() {"$"c})

For Each line As String In lines 'Declares "line" as variable....we brought in 4
lines, cycle through them one by one
    If line.Contains("GPGGA") Then
        Dim parts As String() = line.Split(New Char() {","c}) 'split
        sentence on comma characters
        Dim strMinutes As String
        Dim strDegrees As String

        If parts.Count >= 10 Then 'if sentence at least contains everything
            Try
                strLatitude = parts(2) 'part 2 or word 2 (zero based) in the GGA
                strDegrees = strLatitude.Substring(0, 2) 'Set degrees as the
                first two characters (DD) of strLatitude
                strMinutes = strLatitude.Substring(2, strLatitude.Length - 2)
                'Sets minutes as the rest of the characters (MM.MMMMM...), starting at the character 2
                (zero based)
                txtLat.Text = CStr(CInt(strDegrees) + CDbl(strMinutes) / 60)
                'convert degrees string to integer, convert minutes string to double, divide minutes
double by 60 minutes per degree to get decimal degrees, add decimal degrees to integer
degrees

                If parts(3) = "S" Then txtLat.Text = CStr(CDb1(txtLat.Text) * -1)
                'If word 3 (zero based) is S, then latitude is in southern hemisphere then make ",-"
                'the rest of this is for longitude but does the same thing
                strLongitude = parts(4)
                strDegrees = strLongitude.Substring(0, 3)
                strMinutes = strLongitude.Substring(3, strLatitude.Length - 3)
                txtLong.Text = CStr(CInt(strDegrees) + CDbl(strMinutes) / 60)
                If parts(5) = "W" Then txtLong.Text = CStr(CDb1(txtLong.Text) * -1)
            CalculateGpsOffset() 'Reports gps position relative to
prescribed offset and last heading
Try

Catch ex As Exception
    txtLat.Text = "Unknown"  'This means it was not a complete sentence
    txtLong.Text = "Unknown"  'This means it was not a complete sentence
End Try

Select Case parts(6)  'look at word 6 (zero based), report the fix quality
    Case "0"
        txtFixQuality.Text = "Invalid"
    Case "1"
        txtFixQuality.Text = "GPS"
    Case "2"
        txtFixQuality.Text = "DGPS"
    Case "3"
        txtFixQuality.Text = "PPS"
    Case "4"
        txtFixQuality.Text = "RTK Fixed"
    Case "5"
        txtFixQuality.Text = "RTK Float"
    Case "6"
        txtFixQuality.Text = "Estimated"
    Case "7"
        txtFixQuality.Text = "Manual"
    Case "8"
        txtFixQuality.Text = "Simulation"
    Case Else
        txtFixQuality.Text = "Unknown"
End Select

If parts(7) = "" Then   'word 7 (zero based) is number of satellites fixed
    txtNumSatellites.Text = "0"
Else
    txtNumSatellites.Text = CStr(CInt(parts(7)))  'converts to an integer, e.g 07 goes to 7, then back to string
End If

If parts(9) = "" Then   'word 9 (zero based) is the altitude in meters
    txtAltitude.Text = "Unknown"
Else
    txtAltitude.Text = (CSng(parts(9)) * 3.281).ToString("F1")
End If

End If

If line.Contains("GPVTG") Then   'if it is VTG sentence
    Dim parts As String() = line.Split(New Char() {","c})  'split sentence into words (parts) on comma character
    If parts.Count >= 8 Then   'if sentence contains all the words I want
        then...

82
If parts(1) = "" Then 'word 1 (zero based) is heading in degrees.
North = 0. East = 90, South = 180, West = 270
	txtHeading.Text = "Unknown"
Else
	xtHeading.Text = CSng(parts(1)).ToString("F1")
End If

If parts(7) = "" Then 'word 7 (zero based) is speed in km/hr

txtMPH.Text = "Unknown"
Else
	xtMPH.Text = (CSng(parts(7)) / 1.61).ToString("F1") 'converts km/hr to mph

' The following sets "lastheading" at the current heading if the speed exceeds some set threshold, in this case 0.5 mph
If (CSng(parts(7)) / 1.61) > 0.5 Then
	lastHeading = CSng(parts(1))

txtLastHeading.Text = lastHeading.ToString
End If

End If

End If

Next

End Sub

Private Sub cmbPort_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmbPortGPS.SelectedIndexChanged
If SerialPort1.IsOpen = False Then SerialPort1.PortName = cmbPortGPS.Text 'pop a message box to user if he is changing ports
Else 'without disconnecting first.
	MsgBox("Valid only if port is Closed", vbCritical)
End If
End Sub

Private Sub cmbBaud_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmbBaudGPS.SelectedIndexChanged
If SerialPort1.IsOpen = False Then SerialPort1.BaudRate = cmbBaudGPS.Text 'pop a message box to user if he is changing baud rate
Else 'without disconnecting first.
	MsgBox("Valid only if port is Closed", vbCritical)
End If
End Sub

Private Sub tmrReadSerial_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrReadSerial.Tick
ReceivedText(SerialPort1.ReadExisting())
End Sub
Private Sub CalculateGpsOffset()
    'Offsets GPS by 20 ft, or 6 m, based on last heading reading
    Dim RevHeading As Double = lastHeading - 180
    Dim RevHeading As Double = lastHeading - 180  'calculates the opposite direction to where you were heading
    Dim QuadDeg As Double = 90 - RevHeading 'converts heading from cardinal system to radial (or ordinal system, kendall doesn't know what the hell it is) system
    Dim QuadRad As Double = QuadDeg * Math.PI / 180 'converts ordinal heading from degrees to radians
    Dim DeltaX As Double = (CDbl(txtOffset.Text) / 3.281) * Math.Cos(QuadRad)  'calculates deltaX
    Dim DeltaY As Double = (CDbl(txtOffset.Text) / 3.281) * Math.Sin(QuadRad)  'calculates deltaY
    Dim latCorr As Double
    Dim longCorr As Double
    If IsNumeric(txtLat.Text) Then
        latCorr = CDbl(txtLat.Text) + (180 / Math.PI) * (DeltaY / 6378137)  'calculates corrected lat = lat ... see spread sheet (gps offset.xls)
    Else
        latCorr = 999
    End If
    If IsNumeric(txtLat.Text) And IsNumeric(txtLong.Text) Then
    Else
        longCorr = 999
    End If
End Sub

Private Sub myMap_OnMapZoomChanged() Handles myMap.OnMapZoomChanged
Try
    trkZoom.Value = myMap.Zoom
Catch ex As Exception
End Try
End Sub

Private Sub btnOpenShape_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOpenShape.Click
With OpenFileDialog1 'Executes a series of statements making repeated reference to a single object or structure.
    .Title = "Select a Shape File to Import" 'title
    .InitialDirectory = ShapeFolder 'browse start directory
    .Filter = "Shape File (*.shp)|*.shp" 'only possible to select this extensions
    .FileName = strCurrShape
    .RestoreDirectory = True
    Dim answ = .ShowDialog
    If answ = DialogResult.OK Then 'if answer not cancel, etc..
        cboHeaders.Items.Clear()  
        'Clear previous shape data
        'ResetAll()
        'If .FilterIndex = 1 Then 'ShapeFile
        ShapeFolder = System.IO.Path.GetDirectoryName(OpenFileDialog1.FileName)
strShapeFileName = .SafeFileName

ReadShape()
cboHeaders.Enabled = True

End If
End With
End Sub

Private Sub ReadShape()

Dim ThisRecord As Long, ThisData As Long, ThisPart As Long, VertCount As Long
Layer_Polygon.Clear()

' ShapeIn is the name of the ShapeOCX instance
With ShapeIn

.ReadDataOnMove()
.OpenShape(ShapeFolder & "\" & strShapeFileName, 0)
.ReadDataOnMove = ArcViewShapeFileDLL.ShapeFiles.eReadMode.FastRead

' Read in each shape record in turn
For ThisRecord = 1 To .RecordCount
  buf &= vbCrLf & "Record = " & ThisRecord.ToString & vbCrLf

  ' Print out the name and values of the ShapeFile attribute data.
  For ThisData = 1 To .ShapeFields.Count
    Try
      If ThisRecord = 1 Then
        cboHeaders.Items.Add(.ShapeFields(ThisData).FieldName.ToString)
      End If
      buf &="Field = " & ThisData.ToString & vbCrLf
      buf &=".ShapeFields(ThisData).FieldName.ToString & " = "
      buf &=".ShapeFields(ThisData).Value.ToString
      buf &= vbCrLf
      'Debug.Print.ShapeFields(ThisData).FieldName,
      .ShapeFields(ThisData).Value
      Catch ex As Exception
      End Try
    Next ThisData

    ' Print out the coordinates that make up this shape file
    ' Because we're dealing with a polygon file, it is best to output by Parts
    ' You can also just output from 1 To .Vertice.Count

    For ThisPart = 1 To .Parts.Count

buf &= "Part = " & ThisPart.ToString & vbCrLf

For VertCount = .Parts(ThisPart).Begins To .Parts(ThisPart).Ends
    buf &= "Vertex = " & VertCount.ToString & vbCrLf
    'x coordinate is longitude
    'y coordinate is latitude

    If LongMax = Nothing Then
        LongMax = .Vertices(VertCount).X_Cord
    ElseIf .Vertices(VertCount).X_Cord > LongMax Then
        LongMax = .Vertices(VertCount).X_Cord
    End If

    If LatMax = Nothing Then
        LatMax = .Vertices(VertCount).Y_Cord
    ElseIf .Vertices(VertCount).Y_Cord > LatMax Then
        LatMax = .Vertices(VertCount).Y_Cord
    End If

    If LongMin = Nothing Then
        LongMin = .Vertices(VertCount).X_Cord
    ElseIf .Vertices(VertCount).X_Cord < LongMin Then
        LongMin = .Vertices(VertCount).X_Cord
    End If

    If LatMin = Nothing Then
        LatMin = .Vertices(VertCount).Y_Cord
    ElseIf .Vertices(VertCount).Y_Cord < LatMin Then
        LatMin = .Vertices(VertCount).Y_Cord
    End If

    buf &= .Vertices(VertCount).X_Cord & ", "
    buf &= .Vertices(VertCount).Y_Cord & ", "
    buf &= .Vertices(VertCount).PartNo & vbCrLf
Next VertCount

Try
    If polygonPoints.Count >= 3 Then
        myTestPolygon.Add(New GMapPolygon(polygonPoints, "Poly"))
    End If
Catch ex As Exception
    myTestPolygon(0) = New GMapPolygon(polygonPoints, "Poly")
End Try

polygonPoints.Clear()

Next ThisPart
.MoveNext()
TotalRecords = CInt(ThisRecord)

Next ThisRecord

End With

For Each n As GMapPolygon In myTestPolygon
    If Not n Is Nothing Then
        n.Fill = New SolidBrush(Color.FromArgb(0, Color.White))
        n.Stroke = New Pen(Color.Orange, 3)
    End If
    Next

For Each n As GMapPolygon In myTestPolygon
    Layer_Polygon.Polygons.Add(n)
    Next

myMap.Overlays.Add(Layer_Polygon)

Try
    myMap.Position = New PointLatLng((LatMin + LatMax) / 2, (LongMin + LongMax) / 2)
    myMap.SetZoomToFitRect(GMap.NET.RectLatLng.FromLTRB(LongMin, LatMax, LongMax, LatMin))
    Catch ex As Exception
End Try

End Sub

Private Sub FindShapeData()

If IsNumeric(txtLat.Text) And IsNumeric(txtLong.Text) Then
    ShapeIn.FindbyXY(CDbl(txtLong.Text), CDbl(txtLat.Text))

    If ShapeIn.NoMatch = True Then
        rtbShapeInfo.Text = "NoMatch"
    Else
        buf = "Record = " & ShapeIn.CurrentRecord.ToString & vbCrLf
    For ThisData = 1 To ShapeIn.ShapeFields.Count
        If ShapeIn.ShapeFields(ThisData).FieldName = cboHeaders.Text Then Try
            txtndvi.Text = ShapeIn.ShapeFields(ThisData).Value.ToString
            Catch ex As Exception
                txtndvi.Text = "Nothing"
            End Try
            buf &= ShapeIn.ShapeFields(ThisData).FieldName.ToString & " = "
            Try
                buf &= ShapeIn.ShapeFields(ThisData).Value.ToString & vbCrLf
                Catch ex As Exception
                    buf &= vbCrLf
                End Try
        End If
    End For

End If

buf &= rtbShapeInfo.Text & vbCrLf
End Sub
Next ThisData

rtbShapeInfo.Text = buf

End If
End If
End If
End Sub

Private Sub myMap_MouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventHandler) Handles myMap.MouseClick
  If chkGPSOverride.Checked Then
    'Feed actual GPS here instead of mouse position as needed
    txtLat.Text = CStr(myMap.FromLocalToLatLng(e.X, e.Y).Lat)
    txtLong.Text = CStr(myMap.FromLocalToLatLng(e.X, e.Y).Lng)
    FindShapeData()
  End If
End Sub

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHelp.Click
  If System.IO.File.Exists(File_Name) = True Then
    Dim objReader As New System.IO.StreamReader(File_Name)
    rtbHelp.Text = objReader.ReadToEnd
    objReader.Close()
    rtbHelp.Visible = True
    btnHelpClose.Visible = True
  Else
    MessageBox.Show("No Help Exist's")
  End If
End Sub

Private Sub btnHelpClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHelpClose.Click
  rtbHelp.Visible = False
  btnHelpClose.Visible = False
End Sub

Private Sub Output(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHelpClose.Click
  txtOutput.Text = (phidgetIFK.sensors(1).Value * 2.15)
End Sub

Private Sub extendActuator()
  Dim OutputState As Boolean

  OutputState = phidgetIFK.outputs(0)
  OutputState = True
phidgetIFK.outputs(0) = OutputState
End Sub
Private Sub retractActuator()
    Dim OutputState As Boolean
    OutputState = phidgetIFK.outputs(1)
    OutputState = True
    phidgetIFK.outputs(1) = OutputState
End Sub
Private Sub stopActuator()
    Dim OutputState As Boolean
    Dim OutputState1 As Boolean
    OutputState = phidgetIFK.outputs(0)
    OutputState = False
    phidgetIFK.outputs(0) = OutputState
    OutputState1 = phidgetIFK.outputs(1)
    OutputState1 = False
    phidgetIFK.outputs(1) = OutputState1
End Sub
Private Sub CheckPosition()
    Dim Position As Decimal
    Position = phidgetIFK.sensors(1).Value
    If (Position <= 18) Then
        txtPosition.Text = "1"
    ElseIf (Position <= 37) Then
        txtPosition.Text = "2"
    ElseIf (Position <= 56) Then
        txtPosition.Text = "2.25"
    ElseIf (Position <= 75) Then
        txtPosition.Text = "2.5"
    ElseIf (Position <= 94) Then
        txtPosition.Text = "2.75"
    ElseIf (Position <= 113) Then
        txtPosition.Text = "3"
    ElseIf (Position <= 132) Then
        txtPosition.Text = "3.25"
    ElseIf (Position <= 151) Then
        txtPosition.Text = "3.5"
    ElseIf (Position <= 170) Then
        txtPosition.Text = "3.75"
    ElseIf (Position <= 189) Then
        txtPosition.Text = "4"
    ElseIf (Position <= 208) Then
        txtPosition.Text = "4.25"
    ElseIf (Position <= 227) Then
        txtPosition.Text = "4.5"
    ElseIf (Position <= 246) Then
        txtPosition.Text = "4.75"
    ElseIf (Position <= 265) Then
        txtPosition.Text = "5"
    ElseIf (Position <= 284) Then
        txtPosition.Text = "5.25"
    ElseIf (Position <= 303) Then
        txtPosition.Text = "5.5"
    Else
        txtPosition.Text = "6"
txtPosition.Text = "5.5"
ElseIf (304 <= Position) AndAlso (Position <= 322) Then
    txtPosition.Text = "5.75"
ElseIf (323 <= Position) AndAlso (Position <= 341) Then
    txtPosition.Text = "6"
ElseIf (342 <= Position) AndAlso (Position <= 360) Then
    txtPosition.Text = "6.25"
ElseIf (361 <= Position) AndAlso (Position <= 379) Then
    txtPosition.Text = "6.5"
ElseIf (380 <= Position) AndAlso (Position <= 398) Then
    txtPosition.Text = "6.75"
ElseIf (399 <= Position) AndAlso (Position <= 417) Then
    txtPosition.Text = "7"
ElseIf (418 <= Position) AndAlso (Position <= 436) Then
    txtPosition.Text = "7.25"
ElseIf (437 <= Position) AndAlso (Position <= 455) Then
    txtPosition.Text = "7.5"
ElseIf (456 <= Position) AndAlso (Position <= 474) Then
    txtPosition.Text = "7.75"
ElseIf (475 <= Position) AndAlso (Position <= 493) Then
    txtPosition.Text = "8"
ElseIf (494 <= Position) AndAlso (Position <= 512) Then
    txtPosition.Text = "8.25"
ElseIf (513 <= Position) AndAlso (Position <= 531) Then
    txtPosition.Text = "8.5"
ElseIf (532 <= Position) AndAlso (Position <= 550) Then
    txtPosition.Text = "8.75"
ElseIf (551 <= Position) AndAlso (Position <= 569) Then
    txtPosition.Text = "9"
ElseIf (570 <= Position) AndAlso (Position <= 588) Then
    txtPosition.Text = "9.25"
ElseIf (589 <= Position) AndAlso (Position <= 607) Then
    txtPosition.Text = "9.5"
ElseIf (608 <= Position) AndAlso (Position <= 626) Then
    txtPosition.Text = "9.75"
ElseIf (627 <= Position) AndAlso (Position <= 645) Then
    txtPosition.Text = "10"
Else
    MessageBox.Show("Position Error")
End If
End Sub

End Class