University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Biological Systems Engineering--Dissertations, Theses, and Student Research | Biological Systems Engineering

Spring 2018

DEVELOPMENT OF A SPRAYER PERFORMANCE DIAGNOSTIC TOOL USING IMPROVED MAPPING AND ERROR QUANTIFICATION PRACTICES

Aaron Shearer
University of Nebraska - Lincoln, aaron.shearer.93@gmail.com

Follow this and additional works at: https://digitalcommons.unl.edu/biosysengdiss

Part of the Bioresource and Agricultural Engineering Commons

Shearer, Aaron, "DEVELOPMENT OF A SPRAYER PERFORMANCE DIAGNOSTIC TOOL USING IMPROVED MAPPING AND ERROR QUANTIFICATION PRACTICES" (2018). *Biological Systems Engineering--Dissertations, Theses, and Student Research*. 80.
https://digitalcommons.unl.edu/biosysengdiss/80

This Article is brought to you for free and open access by the Biological Systems Engineering at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Biological Systems Engineering--Dissertations, Theses, and Student Research by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

DEVELOPMENT OF A SPRAYER PERFORMANCE DIAGNOSTIC TOOL USING IMPROVED MAPPING AND ERROR QUANTIFICATION PRACTICES

by

C. Aaron Shearer

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Agricultural and Biological Systems Engineering

Under the Supervision of Professor Joe D. Luck

Lincoln, Nebraska

February, 2018

DEVELOPMENT OF A SPRAYER PERFORMANCE DIAGNOSTIC TOOL USING IMPROVED MAPPING AND ERROR QUANTIFICATION PRACTICES

C. Aaron Shearer, M.S.

University of Nebraska, 2018

Advisor: Joe D. Luck

While sprayer technologies have advanced greatly over the past decade and a half, chemical application errors are still prominent in many in-field operations. Over-application of pesticides can cause harm to the crop, reducing yield, and result in added pollution to the environment. Under-application of pesticides fails to control pests within the field, again lowering crop yields, and causing profit loss for the producer. Current operator feedback from in-field pesticide application operations conveys limited information and often times does not allow the operator to visualize a true representation of their performance. Farm Management Information Systems (FMIS) typically do not account for overlap, varying application rates across the width of the spray boom during turns, or off-rate errors due to controller response. Improved mapping systems and product distribution summaries would allow operators to make better-informed decisions leading to improved management practices during spraying operations.

The Pesticide Application Coverage Training (PACT) tool was developed to deploy data analytics methodologies to sprayer operations data collected during field applications. The goal was to provide improved operator feedback allowing for better management practices by providing enhanced feedback to operators over the course of two years. Data were collected for multiple Nebraska fields and processed by the PACT program which generated high-resolution as-applied maps and quantified error reports. PACT program output metrics were compared with currently available sprayer feedback software and previous studies related to high-resolution as-applied maps. Field-average metrics were not found to be significantly different when comparing the PACT program with these systems. However, when examining how in-field errors were distributed amongst various application rate ranges, significant differences were found. Differences in errors broken down by application rate ranges implied successful inclusion of previously unaccounted for error types by the PACT program. In addition, the program showed potential for technology adoption decision support. The PACT program successfully improved upon current sprayer operator feedback systems which will offer a platform for supporting better management practices in the future.

Dedication

I dedicate this thesis to my family, close friends, and Braxton. Without your continued support I would not have been able to get to where I am today.

Acknowledgments

I would like to first thank my advisor, Dr. Joe Luck, for the guidance and support throughout my Master's program. I would also like to acknowledge Randy Nuss, as industry partner he was integral to the success of the study. As well, I must thank the AMS lab group as a whole, and specifically John Evans for assisting me with my research efforts.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background

Pesticide application has been common practice on farms for many years. Row crop operations employ the use of pesticides to reduce crop damage and increase yield by limiting fungi, insects, and weeds. While benefits of pesticide applications are undisputed, improper use of these chemicals can lead to a variety of short and long-term concerns. Over-application of pesticides can cause harm to the crop, reducing yield, and result in added pollution to the environment. Under-application of pesticides fails to effectively control pests within the field, again lowering crop yields and causing economic loss for the producer, as well as giving rise to herbicide resistant weeds. The impact of improper pesticide application spans beyond that of just the farming operation. Various environmental and health impacts have been linked to pesticide use.

While efforts continue on the research front for improved precision application technologies, another, more immediate, focus to improve operator knowledge and feedback is necessary. Even with new technologies, optimal application will depend on efficient operator performance. One would be misinformed to assume the knowledge level of each individual involved is at a sufficient level for efficient chemical application. Current operator feedback from in-field pesticide application operations conveys limited information and often times does not allow the operator to visualize a true representation of their performance. Current Farm Management Information Systems (FMIS) typically do not account for overlap or varying application rates across the width of the spray boom during turns, and seldom effectively convey off-rate errors due to controller response. It must become important to highlight these errors to operators and use it as a platform for continued education and performance reflection.

The focus of the research efforts were development of improved mapping systems and product distribution summaries that would allow operators to make better-informed decisions leading to improved management practices during spraying operations. Development of improved mapping systems and product distribution summaries would provide more quality feedback to operators and make them aware of how and where their chemicals are truly being applied. In addition, such a system would aid operators in their decision making regarding whether advanced sprayer technology adaptations are justifiable for their fields.

1.2 Organization of Thesis

An overview of pesticide application and the respective level of operator feedback is described in Chapter 1. Chapter 2 details the overall research goal and consequent objectives. Chapter 3 details development of the improved pesticide application reporting program, including data collection and program outputs. Chapter 4 examines validation and verification of the developed program's accuracy. Chapter 5 details final conclusions and explores potential future work. Chapters 3 and 4 were written in paper format with the intent to submit each as individual journal articles.

Chapter 2. Project Goals

The primary goal of this research was to improve pesticide application uniformity through the development of a data analytics tool to increase post-application operator feedback. The specific objectives were to 1) utilize Controller Area Network (CAN) bus data to create high resolution as-applied maps such that operators could observe locations where off-rate and off-target errors may have occurred, 2) generate post-application reports to quantify the impact of operator-based decisions and control system response on observed errors, and 3) compare the errors estimated from the developed program to current information provided to operators via Farm Management Information Systems (FMIS).

Chapter 3. Pesticide Application Coverage Tool (PACT) Development

3.1 Literature review

Extensive herbicide, fungicide, and insecticide use to control pests has become commonplace on farms across the U.S. According to the 2012 U.S. census of agriculture, 285 M acres of cropland were treated with herbicides while an additional 100 M acres were treated with insecticides in 2012 (USDA, 2012). In 2016, U.S. crop producers spent $15.2 B on these products to control pests in their fields (USDA, 2017). At a global level, 6 billion pounds of pesticide were used annually in 2011 and 2012, with nearly 900 million pounds of that being used by United States' producers alone annually (U.S. EPA 2017). Considering the magnitude of pesticides (i.e., herbicides, fungicides, and insecticides) applied, and the amount of time, effort, and money invested in their application, it should be critically important to recognize at what level of efficiency at which they are applied. A Grisso et al. (1988) study found only one in three applicators applying within ±5% of their intended application rate. More recently, a 2010 central Kentucky case study showed three fields, all of which showed only 25% to 36% of the field areas receiving within ±10% of the target application rate. An application problem still remains. On a field-by-field basis these errors may vary, but, in aggregate, considering the wide spread use of pesticides, these errors are worth noting.

Even as precision spraying technologies continue to develop, there will always be some level of error present during field applications. These errors can be broken down into four main categories: off-rate, off-target, controller response, and spray drift. Off-rate errors may result from sprayer turning movements as the effective velocity across the boom varies while a uniform flowrate is applied from the nozzles. Off-target errors may occur when active boom sections overlap onto previously sprayed areas of the field. Controller response error creates over- or under-application within field regions as the spray rate control system works to maintain the desired target rate as the sprayer experiences acceleration and deceleration or boom section actuation. Spray drift is another error commonly present (which for the purposes of these research efforts will not be considered) during field applications, which is defined by American Society of Biological and Agricultural Engineers standards (ASABE, 2016) as, "the movement of liquid or particulate material outside the intended target area by air mass transport or diffusion." In general, as more technological innovations become available in the agricultural sector, improvements to sprayer control systems have been made to mitigate application errors.

One major innovation in spray application technology was the inclusion of automatic boom section control on self-propelled sprayers. These systems have significantly reduced multiple-application and unintentional-application outside of field boundaries. Intuitively, there will always be some level of off-target application errors with sprayers. Numerous studies have examined and confirmed the ability to reduce off-target applications and increase savings with the addition of automatic boom section control. Potential reduction in pesticide application was noted up to 30% of the field area using a high-resolution control system where groups of three nozzles or less were automatically actuated (Luck et al., 2010a). In a follow-up study, Luck et al. (2010b) showed a significant reduction in over-application of chemicals when an automatic boom section control was introduced to a sprayer. Results from the case study showed a significant reduction of overlap from an initial season using manual section control

(12.4% over-application) to the following season with automatic boom section control (6.2% over-application). Field shape and size can play an important role in the extent of off-target errors within the field in addition to the number of control sections. A study by Zandonadi et al. (2011) suggested that a relationship between field shape descriptors (e.g., perimeter-to-area ratio, convex hull, longest pass, square-perimeter index, etc.) and the extent of off-target errors existed among fields. The study also concluded that off-target errors were more prevalent as field shapes increased in complexity as well as for increasing implement widths (or sub-section widths). Another study conducted by Luck et al. (2011a) suggested a strong direct relationship between the field shape factor perimeter-to-area ratio (*P/A*) and overlap error; *P/A* increased as more inclusions and concavities were contained within a field. Erickson and Widmar (2015) reported automatic section control adoption for sprayers at around 33% in 2015 with forecasted adoption to reach nearly 50% by 2018. While this technology is proven to reduce over-application of pesticides, overlap errors will still be present during field applications. However, with better post-application data reporting, there is hope that additional industry adoption would occur to further reduce those errors in the agricultural sector.

Another potential error results in over- and under-application of chemicals during turning movements as a uniform flowrate is applied across the width of the spray boom. This problem is exacerbated as spray booms increase in length on modern, self-propelled agricultural sprayers. A recent study examined three fields where only 25% to 36% of each field received application rates within ±10% of the desired target rate (Luck, et al., 2011b). Results suggested that turning movements contributed to a substantial portion of the misapplication documented. Further research estimated off rate errors across several fields of varying shape and size which indicated high potential for off-rate errors due to paths chosen by the operator (Luck et al., 2011c). Several studies have examined system control structures for addressing non-uniform application rates across the width of the boom during turning movements. One study successfully designed a scalable control architecture using data received from the sprayer CAN bus to actuate varying flow outputs across the width of an implement, applicable to sprayers (Sama et al., 2015). Previous work by Giles and Camino (1990) has led to the successful commercialization of a spray control system designed to provide turn compensation on sprayers using pulse width modulation (PWM) controlled solenoid valves. This strategy introduced a solenoid valve at each nozzle to control poppet position for varying spray droplet size and fluid flow output. The commercially available variable rate spray application technology using PWM control was tested by Porter et al. (2013) which suggested the product worked with in a tolerable error range (target rate ±10%). Variable rate spraying technologies do exist, and are continually being researched for improvements, but there must be a way for producers to identify when these technologies are appropriate for them and sprayer performance feedback will play a key role in that.

For improvements in spray application error feedback to be widely adopted among operators, economic feasibility must be considered. The introduction of costly instrumentation onto an existing sprayer would likely discourage use by most operators. One possible alternative to explore is the use of existing sprayer operational information published across the sprayer Controller Area Network (CAN) bus. CAN bus use in the agriculture industry began in the 1980s with the introduction of microcontrollers for tractor and implement control. Standardization of CAN communication protocols has allowed these integrations to continue to advance more successfully. Amongst the agricultural machinery community, the International Standards Organization (ISO) 11783 document (based off of the SAE J1939 standard) originated in 1989 and continues to be revised and developed with growing technology today (Stone et al., 2008)). The ISO 11783 standard has allowed for inter-connectivity among machinery and implement manufacturers across the agricultural industry today. Through the development of standardized CAN bus protocols, CAN has become an efficient platform for vehicle diagnostics, machinery and implement control, and overall improved management (Darr, 2012). The deployable nature and automation of CAN data collection devices has offered vast benefits among agricultural machinery research (Darr, 2012). The ability to collect a wide array of vehicle metrics has allowed researchers to examine machinery performance metrics and associated relationships with in-field tasks. Studies conducted by Marx et al. (2015) suggested a sufficient level of accuracy originating from CAN bus data which could simplify machinery data acquisition systems. Marx et al. (2016) demonstrated the use of various data acquisition and processing tools to simplify use of agricultural machinery CAN buses as an efficient means for data collection. Various field studies have been enhanced through successful use of the CAN bus as a data collection platform. Pitla et al. (2015) successfully developed a CAN bus-based method of determining tractor operating efficiencies during field operations. A subsequent study (Pitla et al., 2016) was able to estimate tractor operational load states based (e.g., idle, turning, implement engaged) on specific CAN bus messages collected during in-field operations. The study was able to successfully decode the necessary CAN messages through use of the J1939 standard. Kortenbruck et al. (2017) demonstrated the ability to use a data logger to collect and pair a variety of machinery metrics recorded over CAN bus with corresponding Global Navigation Satellite System (GNSS) data. In both cases, visual representations of the field events were generated through the pairing of GNSS data and machine operating parameters along with subsequent mapping in a Geographic Information System (GIS).

Though precision spraying technology development continues to advance, it is unlikely that solutions will be perfect (i.e., completely eliminate application errors). It will remain important for operators to be cognizant of their true performance in terms of application uniformity. Current operator feedback does offer a substantial amount of post-application information when data are extracted from FMIS programs. However, limitations or barriers that could easily be remedied to improve operator feedback are possible with revised mapping and analysis practices. Instrumentation necessary to collect data for such analyses is commonplace on most modern self-propelled sprayers via the CAN bus. In most cases, data missing from the CAN network could easily be supplemented with additional sensor systems. By accessing these data, the development of an improved coverage mapping and error analysis software program could be simplified. Such software would provide improved feedback to enhance operator knowledge as well as offer support for producers in regards to decision making for technology adoption.

3.2 Objectives

The overarching goal of this research was to develop a user-friendly tool to aid in development of improved management practices of sprayer operators. The necessary objectives for reaching this goal were to 1) develop an automated program capable of utilizing data analytics tools to post-process sprayer operational CAN bus data, 2) utilize field-collected CAN bus data to create high resolution as-applied maps such that operators could observe locations where off-rate and off-target errors may have occurred, and 3) generate post-application reports to quantify the impact of operator-based decisions and control system response on observed errors.

3.3 Materials and Methods

3.3.1 Data Collection

Data necessary for the development of the PACT program and accompanying analyses came from University of Nebraska-Lincoln research fields located in the eastern portion of the state. Fields providing data for the study covered a range of shapes and sizes. Chemical spray application procedures were performed as normally conducted by on-farm research staff, with minimal interference during data collection. The sprayer from which data were recorded, was operated by a staff field technician with 5 years of spraying experience. Precision spraying technologies utilized during field operations included: automatic boom section control based off previously defined field boundary maps, and auto-guidance technologies, both of which utilized a local Real Time Kinematic (RTK) Global Positioning System (GPS) correction network. The target application rate for the fields used in this study was 140 L ha^{-1}.

A single self-propelled sprayer (Model 4830, Deere & Company, Moline, IL) with a wet boom was used for all spraying operations. The sprayer boom consisted of 60 equally spaced nozzles, set at 50 cm spacing, spanning a total width of 30.5 m. The sprayer automatic section control system divided the nozzles into seven boom sub-sections. A uniform set of nozzles (XR11006VS, TeeJet, Wheaton, IL) were used across the width of the boom.

The spray rate controller regulated the actuation of seven solenoid valves located at the manifold block, upstream of each respective boom section. A map of the sprayer boom control sections and associated nozzles per boom section is shown in Figure 3.1. Integrated pressure and flow rate sensor data (used for system functionality) were published to the sprayer implement CAN bus, and were recorded for use in this project. If pressure or flow meter readings were not available, a system was developed to incorporate these data for simultaneous logging with any available CAN bus data. A detailed procedure for adding such a system for data collection can be found in Appendix A.

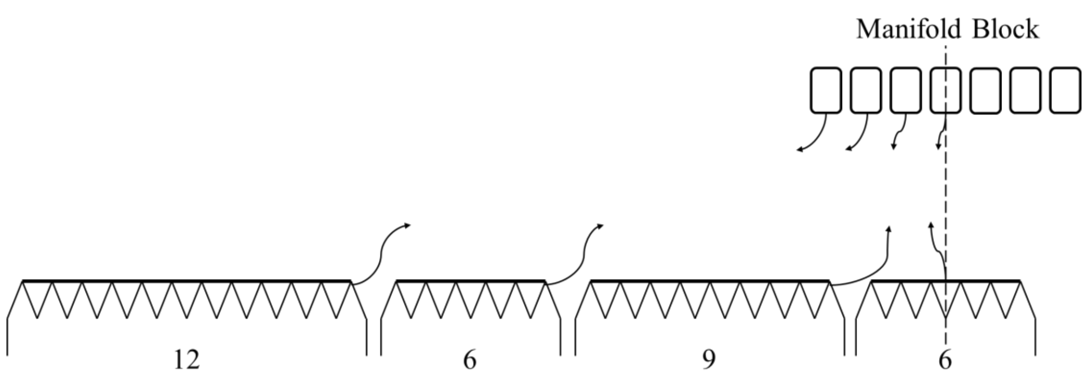


Figure 3.1: Symmetrical half boom control section and nozzle configuration diagram of John Deere 4830 sprayer used during data collection.

To simplify program development, input datasets for the PACT program were restricted to two primary formats. These two data formats were both primarily CAN bus-based and originated from two separate data collection sources: a Farmobile passive uplink connection (PUC) (PUC Generation 2, Farmobile, Overton, KS) and Kvaser Memorator (Memorator) (USBcan Pro 2xHS v2, Kvaser, Mission Viejo, CA).

3.3.1.1 Farmobile Data Logger

The Farmobile data collection system was a telematics solution comprised three main components: the PUC which was connected to the in-cab CAN bus diagnostic port, an accompanying antenna used for cellular and GPS connectivity, and a cloud-based storage platform accessible via internet. The PUC provided two channel monitoring capabilities which enabled CAN message recording simultaneously from the tractor (SAE J1939-based) and implement (ISO 11783-based) buses. CAN bus messages were collected by the PUC and transmitted via a 3G cellular network to the cloud-based storage system. A 16 GB built-in storage device on the PUC was also available to temporarily store data in the event of lost cellular connectivity, stored data were transferred as soon as connectivity had been restored. The PUC contained a GPS receiver to offer sub-meter horizontal location accuracy using Wide Area Augmentation System (WAAS) correction while the system was logging data (**Figure** 3.2).

Figure 3.2: Farmobile PUC unit (right) with associated CAN bus diagnostic port connection and cellular/GPS antennae (left).

Data were collected on a daily (24 hr) basis (whenever the sprayer operated) and processed into engineering units (based on proprietary Farmobile filters) at a frequency of

1 Hz prior to storage in the cloud-based system. Once accessed, 24 hr data files (in .shp format) were downloaded from the Farmobile server. Sprayer performance data of interest collected by the PUC for input into the PACT program included: GPS location with a paired timestamp, sprayer operating system pressure, and boom section status (i.e., on or off). The GPS location data (generated by the internal PUC WAAS receiver) were provided in latitude/longitude (decimal degrees), system pressure values (kPa) with a resolution of 1 kPa, and boom section status recoded as on (100) or off (0).

3.3.1.2 Kvaser Memorator Data Logger

The Memorator was the secondary in-cab data logger used for recording the sprayer operational status via CAN bus. The Memorator functioned as a two CAN channel data logger, allowing it to record messages from the J1939-based tractor and ISO 11783-based implement buses spanning a potential range of baud rates from 50 kbits s^{-1} to 1 M bits s^{-1}. The Memorator was capable of sending and receiving up to 20,000 msgs s^{-1} per channel, which allowed it to record even the highest frequency messages across both sprayer buses as noted in the Memorator factory data sheet (Appendix B). External SD cards (exceeding 64 GB in memory) were placed in the Memorator to store all the messages received. The Memorator (Figure 3.3) was connected to the in-cab diagnostic port and powered from the sprayer 12 V DC battery switch power. As such, anytime the sprayer key switch was on, messages were being recorded and logged from the CAN bus. Raw CAN messages are initially saved to the SD card as a Kwalitan memo file (.kmf) type, extracted to a comma separated value (.csv) or text (.txt) file and post-processed using a case specific developed MATLAB script to filter unwanted messages and convert necessary CAN messages to engineering units.

Figure 3.3: Kvaser Memorator CAN bus data logging device.

While the necessary data for the PACT program was essentially identical to that provided by the PUC (i.e., GPS location with paired timestamp, sprayer boom operating pressure, and boom section on/off status), all messages were recorded as raw CAN message values processed from signals such as seen in Table 3.1. The Memorator logged location data from the sprayer GPS receiver via messages transmitted on the CAN bus, differing from the PUC's accompanying antenna. The sprayer GPS system received RTK correction signals from a local dealership network which provided centimeter-level accuracy. The data were recorded as latitude and longitude at a resolution of 10^{-7} decimal degrees, or roughly 1 cm in the WGS 1984 geographic coordinate system. The Memorator clock, necessary for the timestamp, is set pre-installation, using the accompanying configuration software. Sprayer boom operating pressure was recorded at a resolution of one tenth of a psi. Boom section status was observed as bits stored as either 0 (off) or 1 (on) for corresponding sprayer boom sections. Information necessary for parsing of the required CAN messages can be found in Table 3.2. The GPS, sprayer boom pressure, and section status messages are recorded at rates of 5, 10, and 10 Hz respectively.

Table 3.1: Representation of CAN bus GPS location message as exported by Kvaser showing message identifier (Msg ID) and corresponding message Latitude data (Byte1 – Byte4) and Longitude data (Byte5 – Byte8).

CAN Chan.	Msg ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Time Stamp
1	0x0CFEF31C	8	E3	9F	B6	95	7D	51	AB	43	1187.331

Table 3.2: Values necessary for decoding PACT required sprayer CAN messages based on the John Deere 4830 sprayer used for data collection.

Message	Message ID	Value	Start Bit	Bit Length	Factor	Offset	Endianness	Unit
GPS Location	0x0CFEF31C	Latitude	0	32	0.0000001	-210	Little	Degrees
		Longitude	32					
Section Status	0x10EFD2E1	Section L3	40	1	N/A	N/A	N/A	On (1)
		Section L2	38	1				Off (0)
		Section L1	36	1				
		Section C	34	1				
		Section R1	48	1				
		Section R2	46	1				
		Section R3	44	1				
Spray Pressure	0x18FFFFE1	Pressure	40	16	0.0291666	0	Little	Lbs s^{-1}

3.3.2 PACT Development and Functionality

The PACT program was developed in the MATLAB (MATLAB R2017a, MathWorks, Natick, MA) programming environment to combine the sprayer performance dataset with user defined sprayer set-up parameters to generate as-applied maps and informational reports containing quantitative post-application error information. The final spray application data were compiled in the forms of an as-applied map and an easy to read report format with quantitative values for the various errors. A graphical user interface (GUI) was developed in the MATLAB environment to improve the ease of data entry for potential users. The GUI (Figure 3.4) was divided into three primary sections for data entry: basic field application information, sprayer set-up parameters, and sprayer performance dataset type.

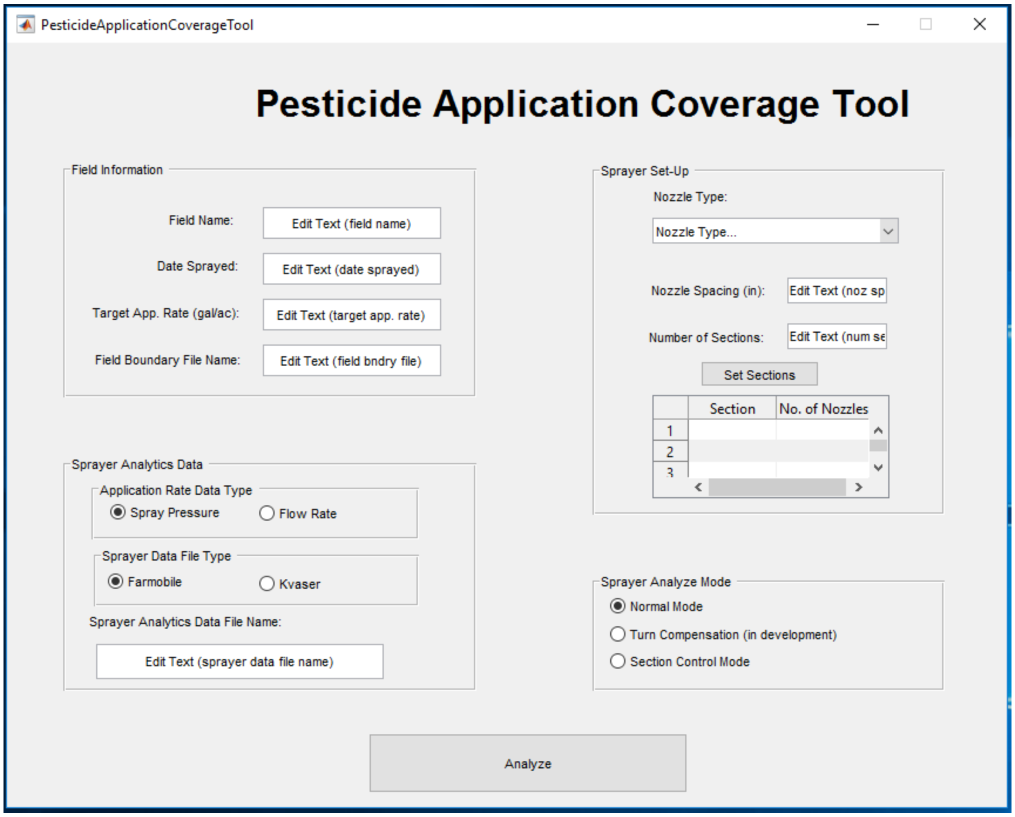


Figure 3.4: GUI created for user interface with the PACT Program.

Following data and sprayer set-up parameters entry a general algorithm is followed (Figure 3.5). Initial data extraction and pre-processing are executed. Following, section locations are identified throughout the field for each GIS data point. From the section locations coverage polygons are projected and application rate, off-target error, off-rate error, out-of-boundary error, and skipped region error calculations are evaluated. A final error processing and report generation concludes the process.

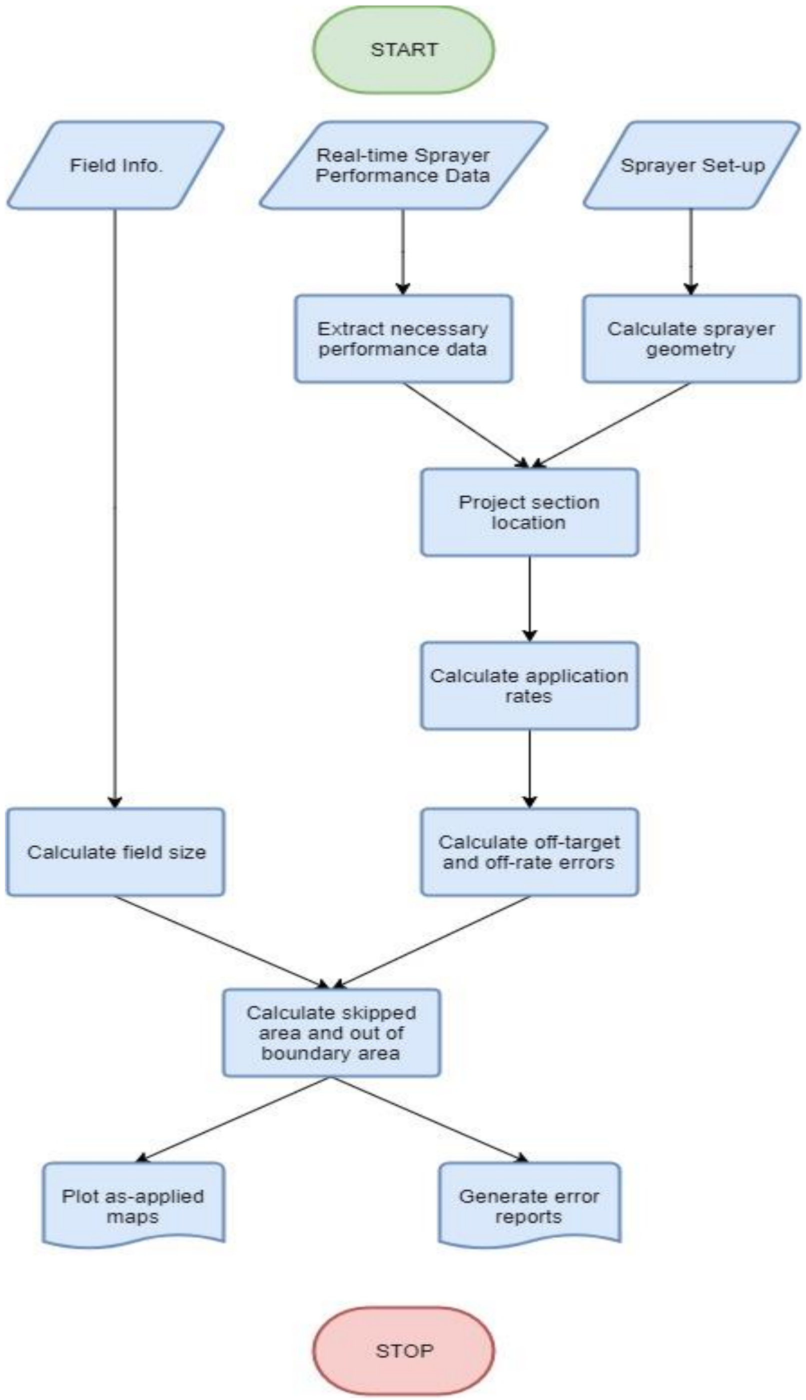


Figure 3.5: Data flow for the PACT program.

3.3.2.1 PACT Program Data Inputs

Basic field application information was entered by the user into the first section of the program GUI which included the field name, application date, and target application rate. The field name entered by the user was utilized within the program to automatically populate the as-applied maps and error reports generated by the program. The target application rate entered into this section was used to compare program-generated application rates to estimate errors throughout the field and identifying their underlying causes. A point shapefile containing a previously defined field boundary is also imported; the boundary was used in error calculations and comparisons throughout the PACT program.

Sprayer set-up and configuration information were entered by the user in the second section of the GUI. Required information for the sprayer set-up section were: the nozzle type and spacing, the number of boom control sections, and corresponding numbers of nozzles per boom control section. The program utilized the number of boom control sections, nozzles per section, and nozzle spacing to format the sprayer geometry used for mapping and creating coverage polygons. The nozzle type was used in the event that boom pressure readings were imported. Flow versus pressure equations were entered into the program for a set of nozzles based on datasheets available from the manufacturer (Teejet, 2017) that were available from the GUI dropdown list. Thus, calculated flow rate values based on pressure readings and nozzle type or system recorded flow rate (total flow was divided by the number of nozzles and then assigned to corresponding boom control sections) were used to apply an associated application rate value for each plotted polygon. Based on the sprayer set-up and GPS data from the performance input file, projected locations were calculated for each individual boom section at each sprayer GPS location using basic geometric relationships as reported by Luck et al. (2010).

The sprayer performance dataset represented the last final required section for user inputs. One of two file types (PUC or Memorator) was selected by the user and the corresponding sprayer metrics filename was typed into the GUI. The data file was placed in the appropriate file folder directory for access by the program such that the data could be imported and prepared in the proper format for program processing. Further data pre-processing was dependent on which of the two selected file types were imported. The PUC (.shp) files represented sprayer application performance data already processed into useable engineering units. The Memorator (.txt) files required additional post-processing once imported as they were still in raw hexadecimal units. In either case, the sprayer performance data including GPS locations, boom section status, and boom flowrate or pressure values were represented in each file.

3.3.2.2 Imported Data Pre-Processing

Data pre-processing were performed using separate steps depending on the sprayer performance file type designated to organize the data into a format such that PACT program analyses could be completed. PUC (.shp) files employed the use of the *shaperead* MATLAB function to define and extract appropriate attribute columns containing the data of interest. The *shaperead* function imports shapefiles and processes the data into a MATLAB table with geographic locations and associated attribute values into individual columns. GIS data imported from the PUC were ran through MATLAB's *smooth* filter, which employed a moving average with a window size of five values to reduce any noise in the WAAS corrected GPS data. The default window size of five was chosen as to not shorten turning movements by including excessive values beyond the turning movement.

Memorator (.csv) files contained substantially more data and required CAN message conversion (to engineering units) prior to sprayer coverage and error analyses. Once the user-specified sprayer performance file was located in the directory, the *datastore* and *tall* MATLAB functions created a table (outside of the MATLAB environment) from which data were queried without the need to download any unnecessary CAN messages (which saved significant processing time). The creation of a *tall* table allowed the use of various MATLAB functions to process the data on a remote disk or file location before importing any data into the MATLAB environment. Prior attempts found multiple millions of unnecessary CAN bus messages being imported and effectively shutting down the program. The CAN message identifiers of required sprayer performance data (e.g., GPS, boom section status, etc.) for further analyses were identified from this table and imported into PACT program for processing. Due to the inability for multiple messages to be sent simultaneously resampling of the necessary CAN messages was required to synchronize timestamps. The timestamp of the lowest frequency message, GPS location, was chosen as the designated timestamp for resampling. A loop in the program synchronized other required messages to the GPS timestamp message frequency by selecting the nearest timestamp corresponding to the desired message.

Once the data from either import type was finalized, the performance variables were split up and placed into separate arrays for further analyses in the PACT program. GPS data imported (from either PUC or Memorator datasets) represented latitude and longitude (units of decimal degrees) in the World Geodetic System (WGS) 1984 geographic coordinate system. To take advantage of metric units, each GPS dataset was converted into the Universal Transverse Mercator (UTM) North American Datum of 1983 (NAD83) projected coordinate system. The appropriate UTM zone was determined using an average of the GPS coordinates for each field dataset imported, and all points were converted to a northing and easting coordinate (units of meters).

3.3.2.3 Boom Section Coverage Polygon Generation

Once sprayer performance datasets were pre-processed, the PACT program proceeded with the creation of coverage polygons for each boom section using the GPS coordinates and user-defined sprayer set-up information. At each sprayer GPS coordinate recorded, an angle was found between the line created from the sprayer path points and a 0° line (east-west orientation). From the angle calculated at each respective sprayer coordinate, boom section midpoints were plotted along an orthogonal line to the sprayer path. Spacing necessary for plotting the section midpoints comes from the user-input information regarding sprayer setup. Along the same line orthogonal to the sprayer path, points were plotted on either side of the section midpoints representing the ends of each section. Through a loop, these section endpoints were projected for the current sprayer location and the next, yielding four convex polygon forming points. The resulting trapezoidal polygons represented the field area covered by each sprayer boom section in that specific time interval (Figure 3.6). Nozzle flowrates were calculated at each sprayer location using a calibration curve modeled from manufacturer data (Figure 3.7). Calibration curves were created for a variety of nozzles available in the GUI Nozzle Type drop down tab. The estimated flowrate (L s^{-1}) from each individual boom section was divided by the corresponding coverage polygon area (m^2 s^{-1}) to estimate an application rate (converted to L/ha) for each coverage polygon. Each coverage polygon was then plotted to a map using the *patch* MATLAB function and the display color symbology for each polygon face was plotted corresponding to an application rate scale set to percentage intervals of; 0% to 50%, 50% to 90%, 90% to 110%, 110% to 150%, and greater than 150% with a cap set at 200% (compared to the user entered target rate).

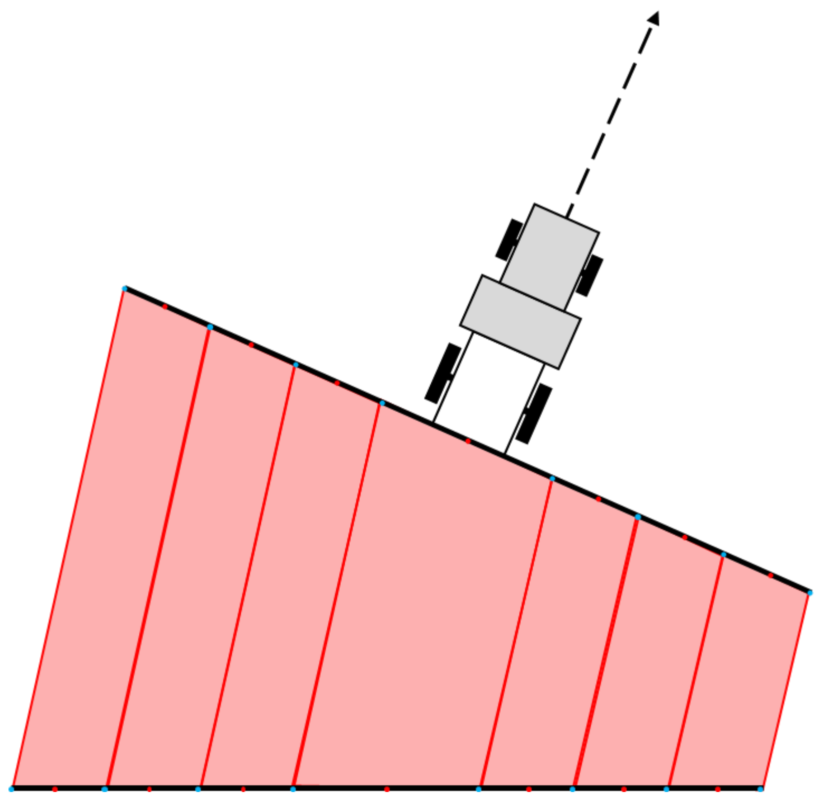


Figure 3.6: Projected coverage polygons spanning between the sprayers current point and previous point.

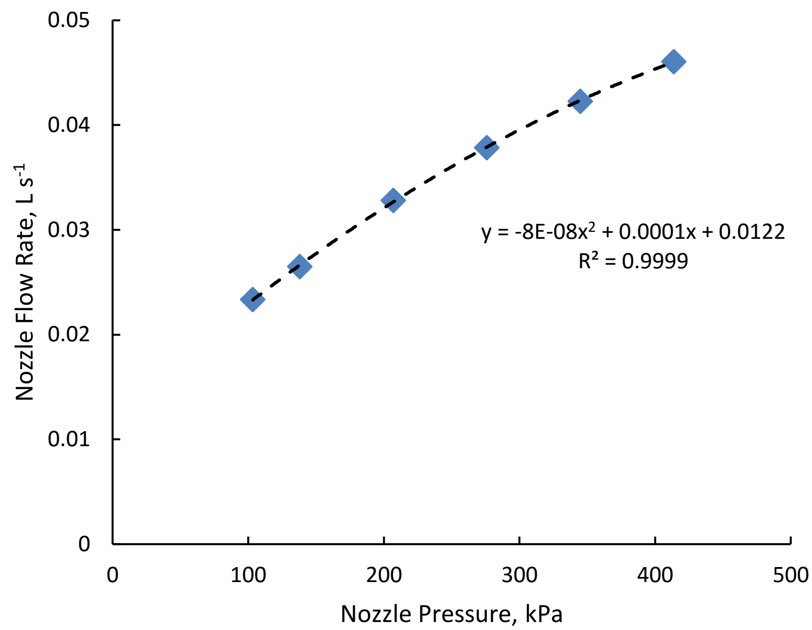


Figure 3.7: Calibration curve for a TeeJet XR11006 nozzle tip modeling nozzle flowrate versus nozzle pressure based on manufacturer data.

3.3.2.4 Boom Overlap (Off-Target Error) Processing

Off-target errors due to overlapping boom section coverage areas were of particular interest for PACT program quantification which represent an over-application error not normally accounted for in current FMIS systems. These errors can often be observed as the sprayer passes in and out of headlands (i.e., point rows) as the automatic section control system must shut-off a portion of the boom over previously sprayed areas.

An *overlap-check* function (refer to Appendix D) was created within the PACT program to inspect each coverage area polygon (and its corresponding application rate) against the other polygons contained within the field application file. The polygons were stored as new variables within the *overlap-check* function which was composed of a loop. The first polygon from the application file was designated as a reference polygon, at which point it was removed from the array of coverage polygons. Every polygon remaining in the array of coverage polygons was inspected for possible intersection with the reference polygon. Any coverage polygons that intersected with the reference polygon were indexed, and the *polybool* MATLAB function (operation setting of *intersect*) was used to identify overlap section points. The identified overlap section points were entered into MATLAB's *polyarea* function to calculate overlap section area. Once the overlap area between the reference polygon and the remaining coverage polygons was defined (i.e., as a new, 'overlap polygon'), the application rates were summed for this overlapping region. Then, the area of the overlap polygon was subtracted from the reference polygon in the original variable array containing all coverage polygons. This process was repeated until all polygons had been used as the reference polygon. Following the completion of the loop identifying any overlapping polygons, all overlap polygons with an area less than 0.001 m^2 were removed from the array. Finally, all of the overlap polygons were stored as a new layer in the program which contained corresponding application rates and coverage areas that had been sprayed multiple times.

3.3.2.5 Boom Off-Rate Error Processing

Off-rate errors resulting from constant spray output across the width of the boom during turning movements are not accounted for with current FMIS programs. The result is outer boom sections (i.e., on the outside of the turn) covering substantially more field area than inner boom sections with a corresponding reduction in application rates (and vice-versa for inner sections). Thus, the PACT program was designed to incorporate these errors into both as-applied coverage maps and error reports.

Off-rate errors identified through the PACT tool analysis were based off of coverage polygon areas generated within the program. For each sprayer coordinate, the areas covered by each boom section were essentially compared with the area covered by the central-most boom section(s). The area covered by each boom section was divided by its respective number of nozzles to determine an average area covered per nozzle. Any section consisting of an area-per-nozzle exceeding ±10% of the central section(s) was indexed and its corresponding areas stored in a separate array for off-rate section errors. As the PACT program generated error reports, these areas were summed and displayed as a total off-rate area for the field analyzed.

3.3.2.6 Skips and Out-of-Boundary Error Processing

Areas sprayed outside of the intended field boundary or areas left unsprayed within the field represent inefficient spray application errors that were documented by the PACT program as well.

Skipped areas were calculated by building a polygon representation of the sprayer path and subtracting that polygon from the field boundary polygon. A loop was created using the *polybool* MATLAB function (with the *union* property enabled) to combine the coverage polygons from each active boom section to create a final coverage file representing the areas sprayed within the field. Once the total sprayer path was summed into one or more large polygons, they were subtracted from the overall field boundary polygon, using the *polybool* MATLAB function (with the *subtraction* property enabled). Resulting areas not covered by the sprayer path were stored as individual polygons (to be used later in error reports) and were plotted in green on corresponding as-applied maps.

Regions receiving spray application outside of the field were determined by identifying any coverage polygons that existed outside of the uploaded field boundary map. The *inpolygon* MATLAB function was used to specifically identify any coverage polygon points outside of the field boundary. Polygons with a vertex outside of the field boundary were indexed and separated out for further comparison. Intersection points between the field boundary and indexed coverage polygons were identified using the *polyxpoly* MATLAB function. The points of intersection between the field boundary and the coverage polygons as well as the coverage polygons' vertices existing outside the field boundary were grouped into matrices on a per polygon basis and entered into a cell array. From those points, the *convhull* MATLAB function was used to rearrange the points in counterclockwise order to form a polygon. These newly formed polygons represented the regions outside of the field boundary receiving spray application. Areas associated with each external field coverage polygon was stored (for later error reporting) and plotted in blue on corresponding as-applied maps.

3.3.2.7 Spray Application Error Calculations

The main goal of the PACT program was to provide useable feedback to sprayer operators. To do so, the errors calculated must be translated into more interpretable and informative units. Thus far, various field application errors have been assessed and quantitative values have been calculated, but meaningful units have yet to be provided. Each error type has associated arrays containing: 1) areas of coverage polygons representing one error type (i.e., off-target or off-rate errors), and 2) associated application rates for each of those areas. Further analysis steps describe how reports were generated from those values in understandable metrics.

The arrays containing areas for each error type were summed and divided by overall field size (reported by the field boundary) which yielded the amount (expressed as percent of field area) within the field in which each error occurred. Using data contained in the application rate arrays, the amount of area receiving different rates was subdivided into five different percentage ranges exceeding the target application rate. The resulting data for each error type were plotted as a histogram identifying the percent field area covered within an acceptable range (i.e., ±10% of the target application rate) as well as the percent field areas covered above and below that range. Five application rate ranges selected for error breakdown were set up as: 0% to 60%, 60% to 90%, 90% to 110%, 110% to 140%, and greater than 140% (compared to the user defined target application rate).

3.3.2.8 Error Report Export

Following the conversion of the application metrics into useable units, all error values were summarized, compiled and exported into an easy-to-read report. A preset error report excel sheet template was developed for the PACT program directory. After the program processed a field dataset, the error report template was copied from the PACT directory and renamed based on the user-defined field name and date of field application. Designated cells within the template were used to import the spray application error values previously calculated by the program. Pre-defined cells were linked to graph templates and used imported data to present error feedback to the operator (Appendix D).

3.4 Results and Discussion

The PACT tool proved to be efficient in the handling of sprayer performance data for report generation. Maps were created for each of the eight fields in this study with corresponding quantified error reports using both PUC and Memorator data input sources.

3.4.1 Maps

While as-applied maps do not to tell the whole story regarding field applications, visual performance representations offer value in locating where in-field errors may be occurring. Improved resolution maps were generated using both Kvaser and Farmobile datasets (Figure 3.8 & Figure 3.9).

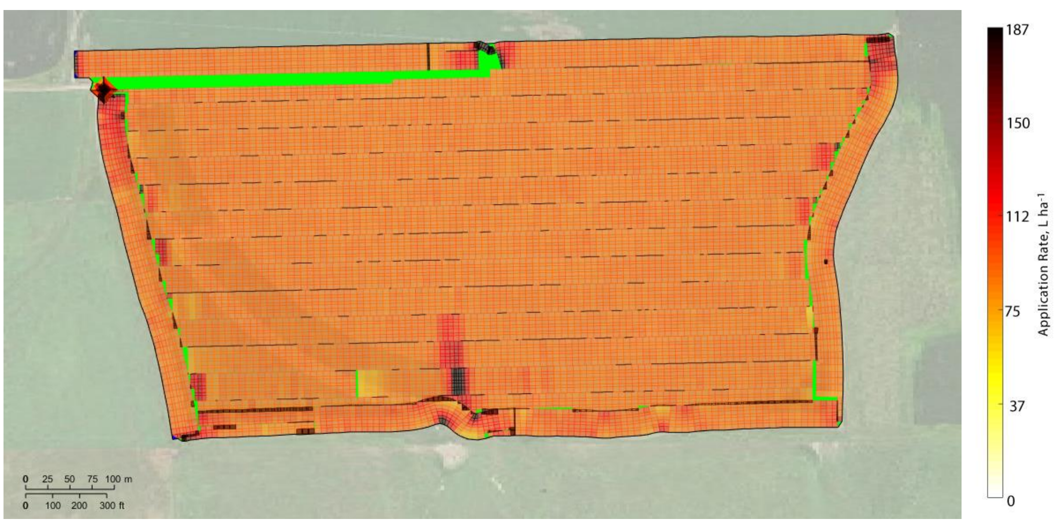


Figure 3.8: High resolution as-applied map generated by the PACT program utilizing a Kvaser based dataset.

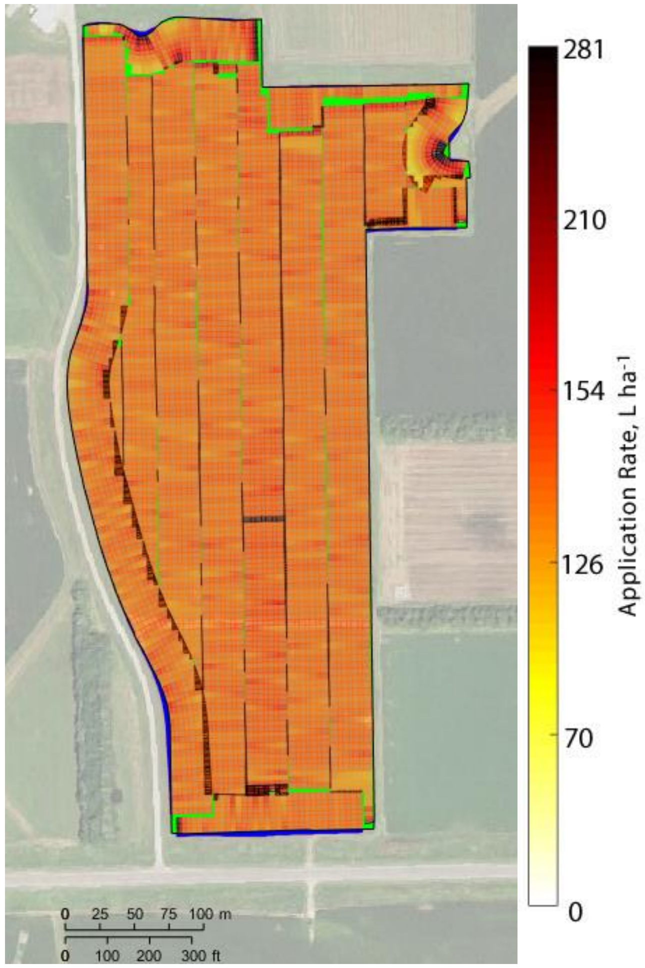


Figure 3.9: High resolution as-applied map generated by the PACT program utilizing a Farmobile based dataset.

Subdividing the boom and mapping based on these control sub-sections improved resolution, and allowed for more descriptive feedback. Off-rate errors were easily distinguishable in the resulting as-applied maps by observing the gradient color scale corresponding to estimated application rates (Figure 3.10). Off-rate errors are important to identify and eliminate because both over- and under-application of pesticides are represented. Under-application often fails to efficiently reduce weeds or other pests competing for water and nutrients or otherwise harming the crop. Under-application to target weeds may also increase the likelihood of herbicide weed resistance due to improper dosing of the active ingredients. Over-application directly wastes money for the producer through excess chemicals used, as well as potentially indirectly through damaged crop. Off-target errors were accounted for and display symbology corresponding to the summed application rates were visible in affected areas. Resulting off-target locations where application rates were doubled (or more) were easily identifiable due to their significantly darker color. Controller errors also were detectable, when present, due to the use of the gradient color scale in the high resolution maps. These errors were present mostly where the sprayer accelerated or decelerated entering headland areas as the control system attempted to adjust the boom pressure to compensate for speed changes.

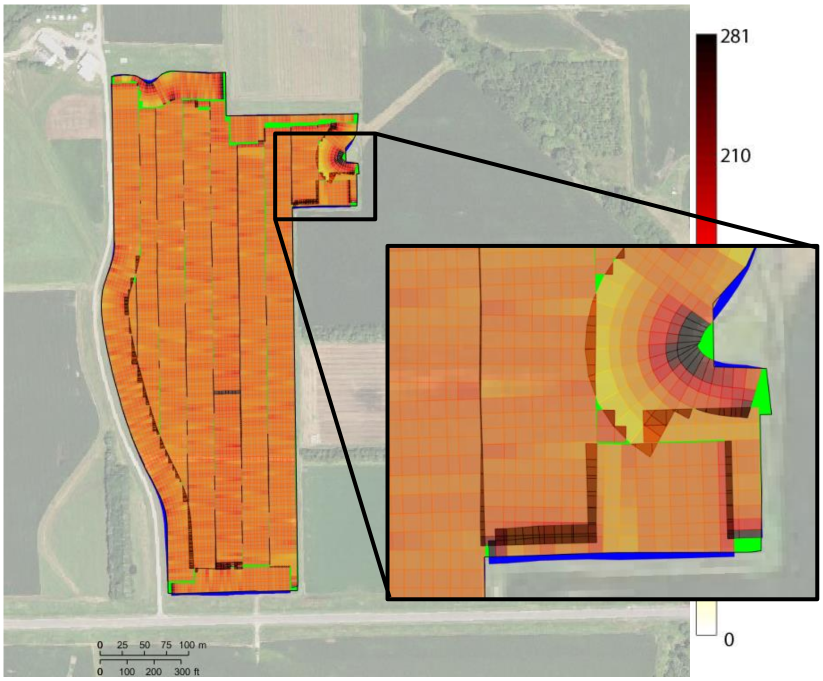
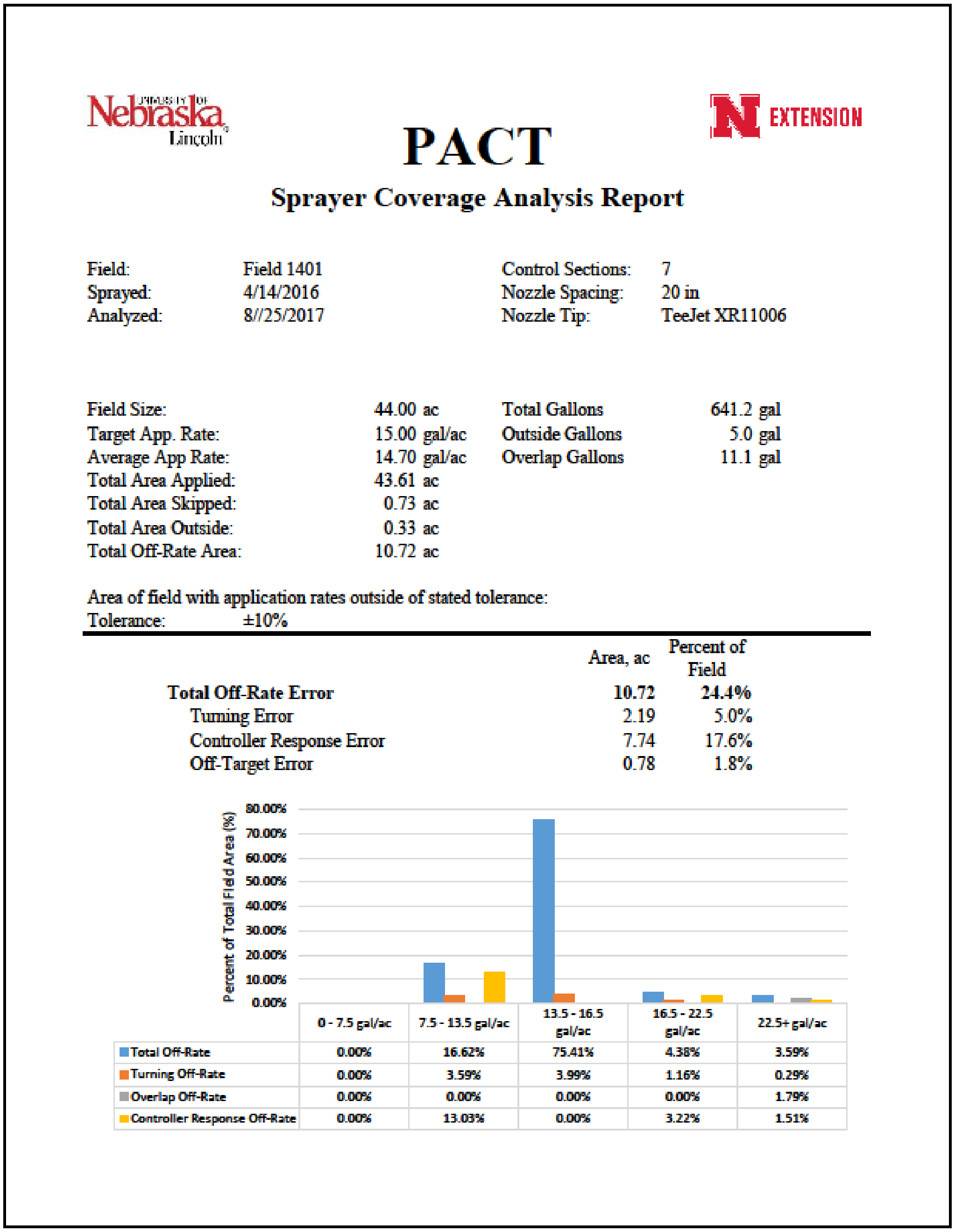


Figure 3.10: High resolution as-applied map generated by the PACT program with areas identifying off-rate areas due to turning movements.

Contrasting blue and green colors were used to successfully highlight out-of-boundary applications and skipped application areas, respectively. These errors are of importance due to both weed resistance concerns and economic losses from inefficient chemical use. Weed pressure in fields negatively impacts crop yields as they compete with the crop for nutrients and water. Spraying outside of the field boundaries leads to direct loss of money in misapplied chemical, with a negative environmental impact.

3.4.2 Error Reports

PACT generated error reports accompanied the as-applied maps for each field, a sample report from one of the studied fields can be seen in Figure 3.11**.** The reports come in an easy to understand single page format, detailing quantified sprayer errors estimated for that field. Record keeping notes can be found at the top of each report (e.g., field name, date, target application rate, etc.) which were previously entered by the user in PACT GUI. Overall field metrics including field size and average application rate are reported in the subsequent report section. The remainder of the report was dedicated to estimating the different application error types. A total amount of field receiving application rates outside ±10% of the desired application rate was reported along with the estimated field area receiving off-target and off-rate errors. For each error listed, a numeric acreage value was listed along with the corresponding percentage of total field area. Skipped areas and out-of-boundary application area totals were also included in the report. An estimate of excess gallons sprayed was calculated for the overlap errors and the out-of-boundary application and reported. A histogram representing the estimated errors for the previously specified error ranges (Figure 3.11). Each application rate range consisted of a percent of field area as well as the contribution of off-target and off-rate errors to respective percent of field area. High resolution as-applied map and corresponding error report for each of the eight Farmobile data set based fields used in this study, and an additional Kvaser data set based field can be found in Appendix C.



Nebraska Lincoln | N EXTENSION

PACT

Sprayer Coverage Analysis Report

Field:	Field 1401	Control Sections:	7
Sprayed:	4/14/2016	Nozzle Spacing:	20 in
Analyzed:	8//25/2017	Nozzle Tip:	TeeJet XR11006

Field Size:	44.00 ac	Total Gallons	641.2 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	5.0 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	11.1 gal
Total Area Applied:	43.61 ac		
Total Area Skipped:	0.73 ac		
Total Area Outside:	0.33 ac		
Total Off-Rate Area:	10.72 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**10.72**	**24.4%**
Turning Error	2.19	5.0%
Controller Response Error	7.74	17.6%
Off-Target Error	0.78	1.8%

	0 - 7.5 gal/ac	7.5 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 22.5 gal/ac	22.5+ gal/ac
Total Off-Rate	0.00%	16.62%	75.41%	4.38%	3.59%
Turning Off-Rate	0.00%	3.59%	3.99%	1.16%	0.29%
Overlap Off-Rate	0.00%	0.00%	0.00%	0.00%	1.79%
Controller Response Off-Rate	0.00%	13.03%	0.00%	3.22%	1.51%

Figure 3.11: Final sprayer coverage report generated for one field using the PACT program.

3.5 Conclusions

A program was successfully developed to aid in the planning of improved management practices for sprayer operators and report potential errors during post-application analyses. An automated algorithm was used to accept user inputted sprayer information and CAN bus-based performance metrics data set to produce as-applied maps and quantified error reports. High resolution as-applied maps, based on the CAN bus-based data sets, were successfully generated to illustrate locations where errors were expected to occur and help quantify off-target and off-rate errors. Visual observation of the as-applied maps revealed locations where turning off-rate, off-target, and controller response errors occurred. Post-application reports were generated with quantifiable measurements of the various errors present. It is important to note that the developed program currently only examines an ideal performance of the system, once the fluid has left the nozzles the droplets are treated as if they were to fall ideally. Environmental factors (i.e. wind, topography, weather, etc.) that may affect the spray application are not taken into account. This combination of higher resolution coverage maps and calculated areas where off-target and off-rate errors occur should allow operators to make more informed decisions during spraying operations and while considering technology adoption.

Chapter 4. PACT Program Verification and Validation

4.1 Literature review

In spraying operations, as with any agricultural operation, efficiency is important. A timely yet accurate chemical application is critically important during any spraying operation to maximize efficiency while minimizing waste and potential harms to the environment. Continuous improvement in any operation relies on quality performance feedback to help drive any decision making towards accomplishing goals or objectives.

One of the earliest studies aimed at improving application performance through quality control verification mapping was conducted by Giles and Downey (2003). The primary effort was directed at locating areas where pressure and wind speed affected spray droplet size (i.e., spray drift) and combined sprayer performance data and GIS analysis techniques. More recent studies have been directed at improving the resolution (and thus reporting capability) of as-applied mapping. Luck et al. (2010a) demonstrated that analytical tools could be deployed to better estimate field coverage areas using sprayer performance data, however accompanying maps were unable to be generated. In a follow up effort, Luck et al. (2011c) developed field coverage maps that identified areas where off-rate errors (caused by turning movements) were likely to occur. Further efforts by Sharda et al. (2013) utilized GIS analysis to identify locations where spray rate controller response impacted boom distribution uniformity for three study fields. Using those same fields, Luck et al. (2011b) successfully merged sprayer trajectory with sprayer boom pressure and status to generate high resolution maps depicting estimated application rates affected by sprayer turning, boom section actuation, and controller response. One limitation of this work was that full coverage (i.e., polygon-based) as-applied maps were unable to be generated by the analysis tools used. Thus, off-target errors (from boom overlap) were unaccounted for in final application rate summaries. A significant finding from the Luck et al. (2011b) study was that up to 35% of the three fields sprayed received rates of the target rate ±10% once sprayer trajectory was incorporated into as-applied maps.

Current FMIS programs typically generate as-applied maps based off of estimates of boom flow (from system flow meter) and area covered (from active boom sections and travel distance) over a certain time period (1 second). However, there is no standard for such sprayer performance feedback systems at this time.

The current trend of data-driven decisions in the agricultural sector has progressed with regards to the availability of digital visual representations and reports. However, currently available sprayer performance report systems generated by FMIS programs still lack the level of detail required for true application assessments. A recently developed software platform, the Pesticide Application Coverage Tool (PACT) was created with the idea of providing high resolution application maps coupled with error reports which quantify off-target and off-rate errors. While the PACT program incorporated analytical techniques from previous studies (Luck et al., 2011b), several improvements were added to automate data processing and add functionality compared to previous efforts. Ultimately, the need to test the PACT program output compared to existing commercially available tools or research efforts was of primary interest.

4.2 Objectives

The overall goal of this study was to support improved management practices and technology adoption by making sprayer operators cognizant of field application errors through the implementation of high resolution as-applied maps and quantitative error reports. Specific objectives for reaching the goal were to: 1) compare and assess the PACT program to current application reporting made possible by commercially available FMIS software, 2) validate PACT program output based on previously published research regarding spray application errors, and 3) evaluate the potential for program use as means of providing technology adoption and financial decision making support.

4.3 Methods

4.3.1 Comparisons with other FMIS Software and Previous Research

In an effort to show validity amongst the PACT program some existing models/systems were used for verification and comparison. Two accepted systems have been chosen to test the validity of the PACT program as well as demonstrate shortcomings currently found in sprayer mapping software. Visual comparisons and quantitative reports were used to gauge similarities, while exemplifying the PACT program’s introduced error types. In general, total sprayed field area numbers were expected to remain similar, along with observations of increased errors and shifting of application rates.

4.3.1.1 Comparison with SMS Software

SMS software (version 17.2, Ag Leader, Ames, IA) was used to compare the PACT program output with that of a commercially available FMIS software package. This software platform was chosen for two reasons. First, the Nebraska-Lincoln research farm currently utilizes this software for field data collection which includes spraying operations. Secondly, SMS software usage is wide-spread among agricultural professionals for creating as-applied maps from a variety of field equipment including sprayers. The comparison with SMS software provided the ability to gauge differences between the PACT program output and a common FMIS software. The SMS system is built off similar principles of plotting and assessing a high density of polygons representing areas covered by the sprayer with estimated application rates. Thus, the SMS software ultimately employed similar inputs (e.g., GPS, boom section status, and flow rate measurements) and outputs (e.g., as-applied maps and a histogram of as-applied rates) to the PACT tool. Both programs are automated systems designed to take sprayer performance metric files and generate output as-applied maps capable of quantifying field areas for multiple application rate ranges.

Data were collected from eight fields at the University of Nebraska research farm using a Farmobile PUC as outlined in Chapter 3. Data collected with the PUC were imported into the PACT program and corresponding output maps and error reports were generated. The self-propelled sprayer (Model 4730, Deere & Company, Moline, IL) as-applied data were downloaded from the in-cab display and imported into the SMS software. The legends used for as-applied map generation were set to be identical between the PACT and SMS programs. As such, visual comparison between maps created by both programs were conducted to identify areas where the high resolution capacity of the PACT program could offer a better representation of application rates. The SMS program was capable of estimating the amount of each field receiving different application rates based on adjustment of as-applied histogram rate ranges. Thus, the histogram summary of rate ranges from the PACT program were used to set these values in the SMS software. By doing so, it was then possible to compare how much field area each software estimated as receiving various rates of spray application. For instance, the target rate ±10% represented one band or range, which the percent of field area receiving rates below 90% or below 50% of the target rate could be quickly compared. As witnessed in as-applied maps generated by the SMS software, off-target and off-rate errors (from turning movements) were not present. This comparison was conducted to prove the hypothesis that the PACT program could successfully quantify these errors (both in map and report form) which was not done in the SMS program output. To test for significant differences in the output generated by the two programs, statistical analysis was performed using a two way ANOVA test followed by a Tukey-Kramer HSD comparison test at an alpha value of 0.05. The percent field area within a respective error band or range was used to test for significant differences among the two treatments (i.e., PACT and SMS programs). Comparisons were made between the two treatments as a whole across all included application rate ranges. As well, comparisons on a more detailed level of each application rate range independently were also assessed to verify improvement of error identification by the PACT program. Replicates for each measured value were represented by the error values from each of the eight fields.

4.3.1.2 Comparison with Luck et al. (2011b) Study

A previously published study regarding high resolution as-applied maps and error reporting by Luck et al. (2011b) was used as a secondary comparison for the PACT program. The Luck et al. (2011b) study represented seminal work accounting for off-rate errors not typically represented by FMIS systems. Limitations of this previous research included the inability to account for off-target (i.e., overlap) errors as well as generating full coverage as-applied maps. However, access to the Luck et al. (2011b) results provided a way to gauge success of the PACT program in the determining of off-rate errors. Data from the Luck et al. (2011b) study were made available and used as inputs into the PACT program to allow for a direct comparison. Like the PACT program, Luck et al. (2011b) used similar input values (e.g., sprayer GPS coordinates, boom section status, and pressure-based nozzle flow rate rates), though data were processed through differing algorithms and different outputs were ultimately provided in the application reports. While the Luck et al. (2011b) study represented initial efforts to generate as-applied maps which incorporated boom sub-section coverage areas (accounting for off-rate errors), much of the work MS Excel and required substantial manual editing. The system provided a map displaying color scaled points displaying application rates as opposed to full coverage polygon. However, an important similarity was that both systems generated quantified error output values estimated from in-field applications and separated those errors into multiple application rate ranges for reporting purposes.

Data from the study were made available in a Microsoft Excel (.xls) file which were pre-processed prior to import into The PACT program. Data were imported into MATLAB then placed into a structure using field names (e.g, ‘SprayPress’, ‘TS’, ‘SectionR3’, etc.) to match those found in PUC .shp file attributes. The structure was then exported using MATLAB’s *shapewrite* function, and could then be successfully used for the PACT program. Necessary data required from the Luck et al. (2011b) study included northing and easting UTM points with associated boom pressure and section control values. The Luck et al. study used pressure recordings from fourteen transducers distributed within the 30 boom control sections. During the data pre-processing active section pressure values were averaged to a single value per GIS data point. A recent study by Forney et al. (2017) found little to no pressure variation when comparing sprayer system pressure with boom sub-section pressure which supported the averaging of these values. Slight modifications were made to the PACT program to output as-applied maps to create a matching color scale as those from the Luck et al. (2011b) study. Visual comparisons were again conducted between as-applied maps. Histograms containing percent of field area affected by overall error values and off-rate and controller response error values were provided from Luck (2012). The percent of field area affected by these errors were distributed across five application ranges of: 0% to 40%, 40% to 90%, 90% to 110%, 110% to 160%, and greater than 160% (compared to the target rate of 93.5 L ha^{-1}). PACT output error reports were set to the same application rate ranges to allow for statistical comparison between output data from the two systems. Comparisons were conducted to confirm the hypothesis that the PACT program could quantify the same overall errors, as well as off-rate errors, for the application rate ranges and calculate additional errors not provided by the Luck et al. (2011b) study (e.g., off-target errors). Significant differences in the output generated by the two programs were tested through a statistical analysis performed using a two way ANOVA test followed by a Tukey-Kramer HSD comparison test at an alpha value of 0.05. The percent field area within respective overall application rate ranges and off-rate error application rate ranges were used to test for significant differences among the two treatments (i.e., PACT and Luck et al. systems). Comparisons between the two treatments as a whole across all included application rate ranges were assessed. Comparisons on a more detailed level of each application rate range independently were also assessed to verify improvement of error identification by the PACT program. Replicates for each measured value were represented by the error values from each of the three fields included in the Luck at al. (2011b) study.

4.3.2 Information Technology Adoptions

The ability of the PACT program to run various scenarios allowed it to be a starting place for the assessment of sprayer technology adoption. With the PACT program, the user can choose to take entered sprayer data and run the analysis mimicking a single control section for the entirety of the boom. Sprayer data was entered as usual, when the 'Section Control' (SC) mode button was selected on the GUI, all sprayer sections were set as active when any one single section indicated activity. An operator's highest priority when operating with single section control be that of covering the entirety of the field as opposed to optimizing coverage and minimizing over-application. Regular PACT outputs were created for the SC analysis mode. Maps were used for visual comparison of the fields using normal functionality and under the SC analysis mode conditions. The quantitative reports were used for numerical comparison and statistical analysis of the two models, and subsequently used to assess chemical application costs. Estimated prices and recommended application rates were compiled from the UNL Crop Watch (2018) budget report and Nebraska Extension 2014 Guide to Weed Management (Nebraska Extension, 2014).

To demonstrate its capabilities, five fields were chosen and used to assess potential savings from manual full boom control and multiple section automatic control. As-applied maps from the PACT program were used for visual comparison to assess the expected increase in off-target errors from the single full boom control mode. Field-wide average application rates were compared for both cases in each of the five fields. An

ANOVA test was conducted to test the hypothesis that reduced section control would result in higher application rates. Estimated pesticide costs were paired with off-target errors to assess potential savings from increased boom section control on a per acre level. A sample herbicide mix applied to multiple fields included in the study was used for economic analysis of automatic section control. A sample break-even analysis was conducted for an average Nebraska farm of 364 ha (USDA 2017). The sample chemical mix consisted of the following herbicides, adjuvants, and water; methylated seed oils (MSO), Corvus, Buccaneer, 2,4D-LV6, Atrazine 4L, ammonium sulfate (AMS), and water (Table 4.1).

Table 4.1: Per acre pesticide mix costs based off chemical purchase prices and recommended application rates.

Pesticide Mix	Purchase Price	Purchase Unit	Recommended Application Rate (per acre)	Applied Unit	Applied Cost (per acre)
MSO	$ 20.00	gal	1.5	pint	$ 3.75
Corvus	$ 915.00	gal	4.5	oz	$ 32.17
Buccaneer	$ 15.00	gal	30	oz	$ 3.52
2,4D-LV6	$ 29.00	gal	12	oz	$ 2.72
Atrazine 4L	$ 3.25	lb	1.8	lb	$ 5.85
Total					**$ 48.00**

4.4 Results and Discussion

4.4.1 Overall PACT Findings

Field analyses showed substantial field area percentages for each of the eight fields analyzed in this study receiving application rates outside of the desired application rate range (±10% compared to the target rate of 140 L ha^{-1}). Field area percentages receiving undesirable application rates ranged from 16.7% to 37.5% of total sprayed area throughout the eight fields. Both off-target and off-rate errors were present in each field, averaging 3.2% and 11.0% of field areas respectively. The remainder of error observed in field was attributed to controller response error. It is important to note that off-rate and off-target errors are not mutually exclusive and at times appeared to contribute to compounding errors. Of the fields examined few demonstrated high levels of field complexity. A full error breakdown per field can be seen below in Table 4.2.

Table 4.2: Field area percentages for application errors residing outside ±10% of the target application rate.

Field	1401	1404	2306	2311	2635	2017	2018	2020	AVG
All Error	24.6%	23.2%	31.3%	27.2%	15.9%	26.2%	22.6%	37.7%	26.1%
Off-Rate Error	5.0%	9.8%	14.5%	18.9%	7.4%	7.7%	8.5%	16.0%	11.0%
Off-Target Error	1.8%	2.9%	3.8%	2.6%	1.9%	4.5%	4.9%	3.6%	3.2%

PACT reports broke errors down into one of five application rate ranges of 0% to 50%, 50% to 90%, 90% to 110%, 110% to 150%, and greater than 150% (compared to the target rate of 140 L ha^{-1}). A few noticeable trends appeared when displayed on a per application rate range level. Intuitively, off-target errors appeared most frequently in the highest application rate range, as application rates were frequently doubled due to multiple application. Off-rate errors, present during turning movements, were observed almost exclusively in the three center application rate ranges, with the 70 to 126 L ha^{-1} application rate range receiving the highest amount. Off-rate errors occurring most frequently in the lower application rate ranges of the middle three ranges can be expected as the lower rates occur in the outer boom sections during a turn, which consequently cover larger areas. Average error rates per application rate range for the eight fields examined can be seen below (Figure 4.1).

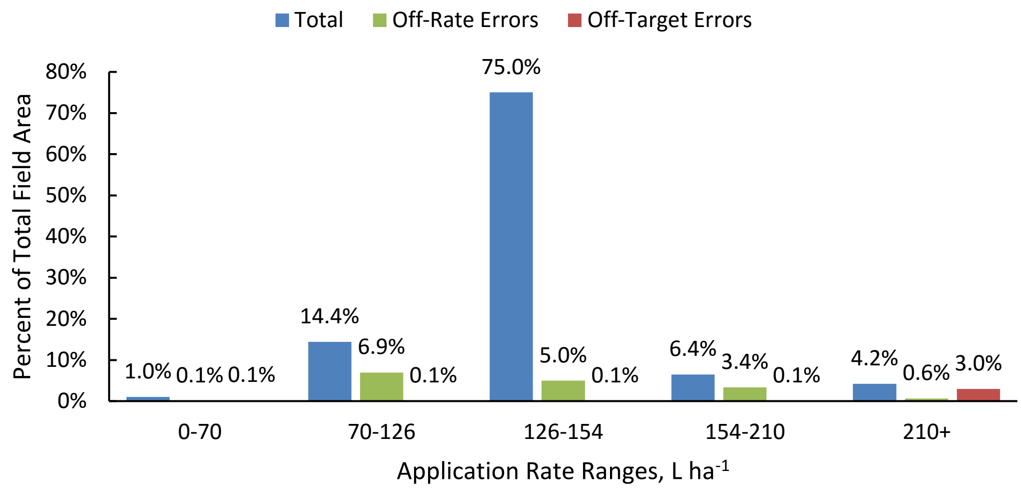


Figure 4.1: Averaged field area percentages for the various errors identified in PACT reports for the eight fields analyzed.

Error amounts varied from field to field, but substantial error amounts were consistently present. Across the nearly 73 ha examined 17 ha received undesirable application rates. If these numbers were to be extrapolated across a full farm scenario excess costs and profit losses could add up quickly.

4.4.2 Comparisons with other FMIS Software and Previous Research

Visual and numeric comparisons were made between the PACT program and two currently accepted systems. The PACT program displayed positive similarities between both systems, while showcasing its ability to include off-target errors previously unaccounted for. Statistical analysis were ran and confirmed the initial visual and numeric similarities observed between the systems.

4.4.2.1 Comparison with SMS Software

A visual comparison between as-applied maps generated for a portion of Field 2635 by both SMS and PACT (with matching SMS legend) programs is shown in Figure

4.2. Visual inspection between these two SMS and PACT as-applied maps indicated the overall application rates, specifically towards the centers and in the middle of passes, were similar. However, as the sprayer navigated through turns and entered and exited the headlands, errors became more pronounced. During turning movements, the expected variation across the width of the boom became clearly visible in the PACT as-applied map, transitioning from dark green (over-application on inner boom sections) to orange (under-application on outer boom sections). These off-rate errors could not be visually observed from the as-applied map generated by the SMS program. Off-target errors were also presented as dark green polygons, located most prominently on the interior headlands passes where boom control sections were turning off to minimize overlapping with a portion of the previously sprayed headland region. An additional as-applied PACT map (using the standard legend) was included in Figure 4.2 to illustrate how selected color scale range and resolution made errors even more visually prominent. . Off-target errors became more distinguished due to the contrasting colors chosen for this legend which improved the visual gradient across the boom during the turning movements. Lesser off-rate errors during turning movements became more visible with the improve color scale, and the PACT as-applied maps allowed for visualization of compound off-target and off-rate errors. The inclusion of locations where skipped (unsprayed) areas existed (bright green), and out-of-boundary application occurred (bright blue) also provided valuable insight into application efficiency. Overall the as-applied maps generated by the commercially available FMIS program did not provide the level of detail which truly represented field spraying operations. While the coverage polygons plotted in the SMS as-applied map may have indicated controller response issues, errors resulting from boom overlap or turning movements were not well defined. Unless the user were to input additional application ranges in the map legend, most controller-related application rate issues would likely go unnoticed. The PACT program as-applied clearly provided a better visualization of the sprayers performance and potential errors during field application.

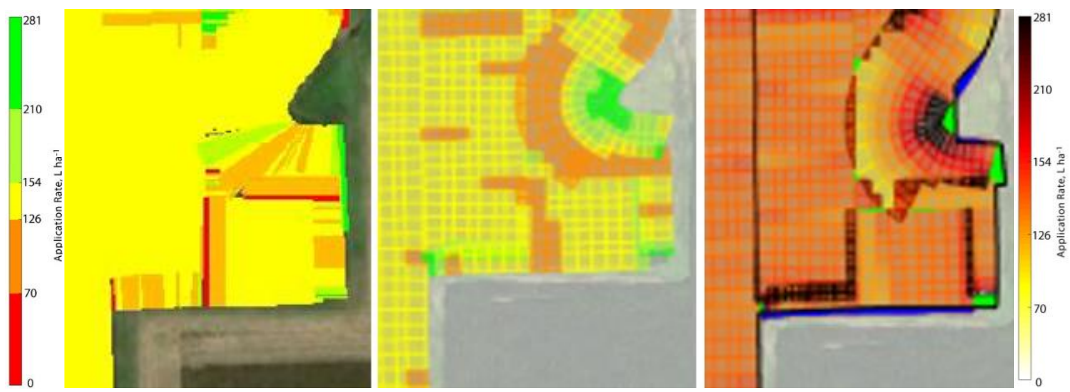


Figure 4.2: Comparison of SMS as-applied map (left), PACT as-applied map using identical SMS legend (center), and PACT as-applied map with standard PACT legend (right).

The two initial numeric comparisons made between the SMS and PACT field reports were the total sprayed area and the average application rates across each field. The percent change in area applied between the SMS and PACT reports indicated an average decrease of -6.5% in coverage areas estimated by the PACT program for the eight fields examined. The percent change in average application rate for the fields ranged from a -2.2% decrease to a 6.3% increase, averaging out to 0.2% higher rate estimates for the PACT program as compared to the SMS software. For both values separate ANOVA tests analyzed the effect of the two programs on estimated field coverage area and average application rate. In both cases no significant differences existed between the two programs. The lack of significant difference between the two programs, especially with regards to field area, provided support for the PACT program's ability to replicate a commercially available FMIS system.

The underlying hypothesis of this study, however, was that the PACT program would better quantify field areas receiving various application rates due to higher resolution maps and error quantification compared to the SMS software. Both systems reported the percent of field area receiving application rates subdivided into five ranges. The application rate ranges chosen for comparison were: 0% to 50%, 50% to 90%, 90% to 110%, 110% to 150%, and greater than 150% (compared to the target rate of 140 L ha^{-1}). Observation of percent field area for each application rate range showed consistent results throughout all eight fields for both SMS and PACT programs (i.e., a substantial portion of the field was covered within 90% to 110% of the target rate with lesser portions covered beyond this range), as was expected. However, the 50% to 90% of target rate range, 90% to 110% of target rate range, and greater than 150% of target rate range were visibly different when comparing the reports from the SMS and PACT programs. The decrease (14.9% of field areas) in middle application rate range was expected as the PACT program accounted for both off-target and off-rate errors not previously accounted for by the SMS system (Figure 4.3). The introduction of these two particular errors in the analysis, as expected, estimated that field areas likely received application rates the target rate ±10%. Since the SMS program lacked the ability to account for off-target errors, it was expected that the PACT program would demonstrate increases in double (or triple) applications as off-target errors were accounted for. Thus, there was a marked increase in the portions of each field receiving application rates exceeding 150% of the target rate as shown in Figure 4.3. As expected, PACT program estimates of field areas covered within the 50% to 90% of target rate range also displayed a consistent change as well, increasing on average by 10.1%. The observed increase here was likely accounted for by off-rate errors present in turning movements. As the sprayer navigated a turn, the outer boom sections effectively applied a lower application rate, and in doing so, covered more field area which led to larger portions of field areas affected by this application error. This finding supports the fact that turning movements during field application are more likely to results in lower application rates across the field. The primary concern being inefficient control of pests and the potential for buildup of weed resistance or further crop loss due to completion.

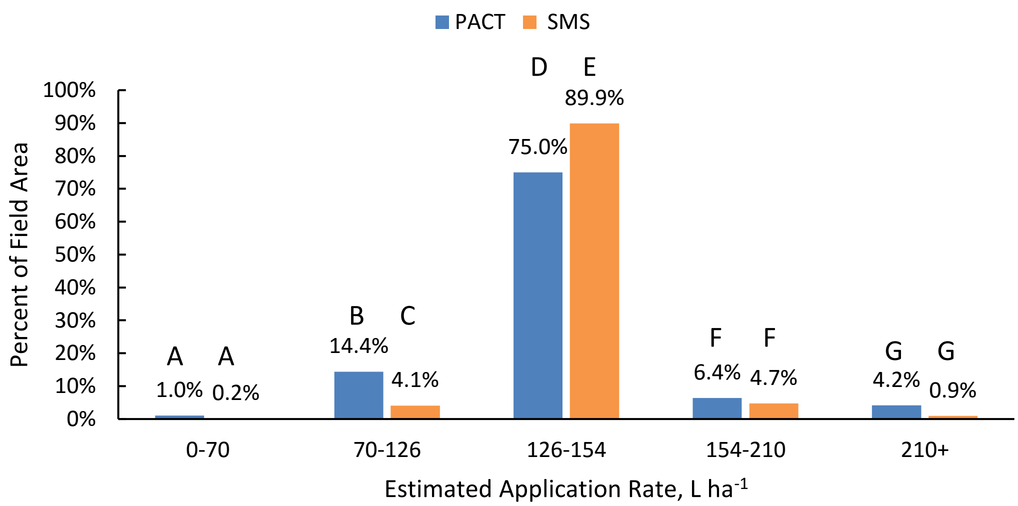


Figure 4.3: Field area percentage averages from the eight fields analyzed by SMS and PACT sprayer performance feedback systems (different letters within each application rate range indicate significant differences between PACT and SMS software estimates).

Statistical analysis results yielded in a p-value of 0.61 when comparing the total field coverage areas and average application rates generated by both programs, which indicated no significant difference between these values. However, the results of the statistical analysis comparing percent field area covered within the separate application rate ranges yielded some significant differences. Field area estimates for the 50% to 90%, and 90% to 110% of target rate ranges were significantly different between the SMS and PACT program results. These indications of significant differences suggest the ability of the PACT program to account for errors not found in the SMS report as the field area percentages began to diverge from the target rate ±10% (i.e., middle application rate range). A summary of the statistical analysis (ANOVA) test results can be seen in Table 4.3.

Table 4.3: ANOVA results for SMS and PACT comparison by band.

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Program	1.2	1	1.16	0.26	0.614
Application Rate Range	78471.9	4	19617.98	4399.79	0.000
Field	1.1	7	0.16	0.04	1.000
Program* Application Rate Range	1372.5	4	343.12	76.95	0.000
Program*Field	1.5	7	0.22	0.05	1.000
Application Rate Range * Field	554.1	28	19.79	4.44	0.000
Error	124.8	28	4.46		
Total	**80527.2**	**79**			

4.4.2.2 Comparison with Luck et al. (2011b) Study

A visual comparison between ‘Field 1’ from the Luck et al. (2011b) study and that of the PACT program demonstrated overall similarities between the two methods for generating as-applied maps (Figure 4.4). However, in this case (as opposed to comparison with the commercially available FMIS) both systems were able to clearly indicate regions where off-rate errors were expected from sprayer turning movements. The most distinct location for noticing this issue was along the north end of the field as the sprayer turned to navigate curved field boundary. Controller response errors were visible in both systems, as seen in Figure 4.4 in the northern portion of the most western headland pass, and becomes distinctly more visible with the PACT legend. The application rate showed a gradient transition from yellow (under-application) to an orange-red (near target application). The main differentiation between the PACT program and Luck et al. (2011b) methods for generating as-applied maps was the ability of the PACT program to include of off-target errors. Field 1 off-target errors were observed as the sprayer entered and exited the headland passes, as well as slight overlaps between the primary field passes. The majority of off-target errors stand out more as a contrasting red color (using the standard PACT program legend) in Figure 4.4. An interesting observation existed in Figure 4.4 on the PACT as-applied map (using the Luck et al. 2011b legend) near the bottom right field edge. Here, two blue coverage areas (under-application) overlapped onto each other forming an off-target error. However, in this case, the overlap error of two low application rate regions which fell within the desired application rate range ±10%. As noticed with the FMIS program comparison, the PACT as-applied map with standard legend allowed for a more detailed representation of application rates due to the darker contrasting colors where application rates were doubled.

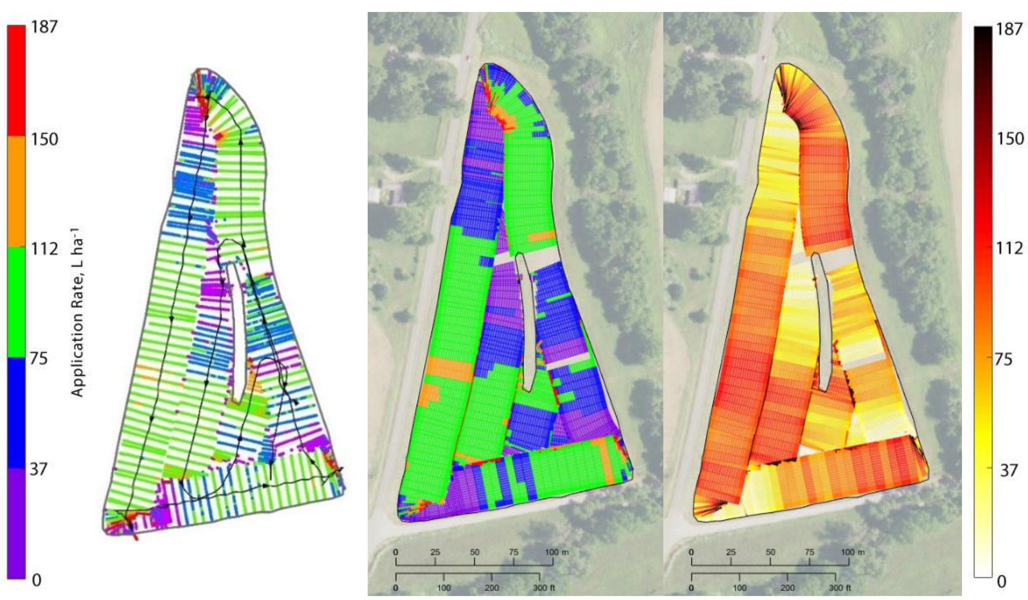


Figure 4.4: Comparison of Luck et al. (2011b) as-applied map (left), PACT as-applied map (with Luck et al. 2011b legend) (center), and PACT as-applied map (with standard PACT legend).

The validation through comparison with the Luck et al. (2011b) study required statistical comparison. Comparisons between total field sprayed area and average application rates from both programs served as an initial assessment. Due to the inclusion of off-rate errors in the Luck et al. study both metrics were expected to remain similar in the PACT reports; however, off-target errors accounted for in the PACT program could result in some differences. The percent change from the Luck et al. 2011b study to the PACT program for total sprayed area for fields 1, 2, and 4 came in at -5.6%, -6.0% and -4.4%, respectively. When a statistical multiple comparison test was conducted for total sprayed area, no significant difference was reported between the two systems. Average application rates between the two systems showed little variation, percent changes were -20.7%, -22.0%, and -24.6% for fields 1, 2, and 4, respectively. When assessed through statistical analysis a significant difference was found between the two systems. Decrease in average application rate stemmed mostly from the reduction of pressure data used. The PACT program was designed to handle a single pressure input. The Luck et al. (2011b) study recorded readings from fourteen pressure transducers. An average of the active boom section transducers resulted in a substantially lower pressure reading being applied across all active sections in the PACT program (Figure 4.5). The lower pressure value carried through the PACT algorithm resulted in lower application rates. In addition, it is important to note that in both cases the statistical power was low due to the lack of replicates (only three fields) and may not offer strong support for or against similarities.

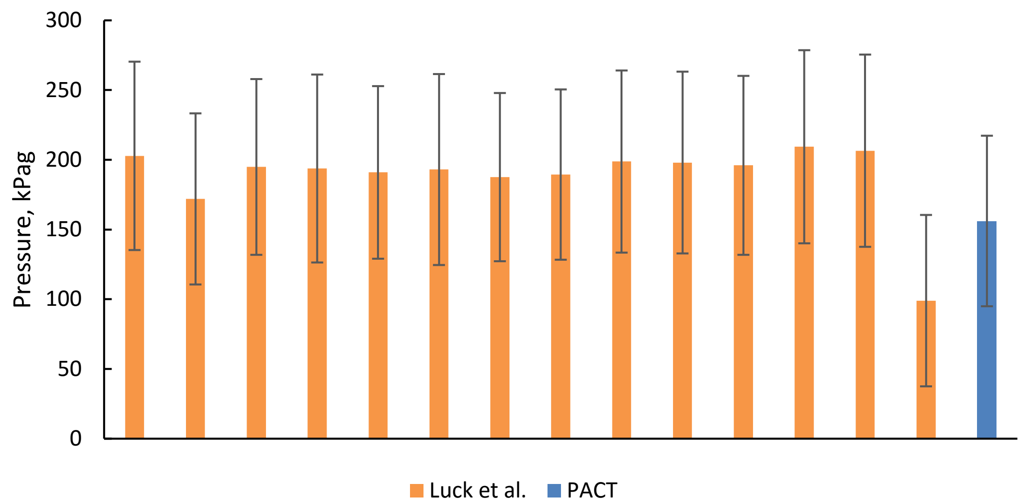


Figure 4.5: Average pressure reading with standard deviation error bars for each Luck et al. (2011b) recorded pressure transducer compared with single used pressure value used in the PACT program.

Both systems reported the percent of field area receiving application rates subdivided into five ranges. In this instance, application rate ranges chosen for comparison were set to match that of the Luck et al. 2011b study at: 0% to 40%, 40% to 90%, 90% to 110%, 110% to 160%, and greater than 160% (compared to the target rate of 93.5 L ha^{-1}). Averages of percent field area covered from the three fields at each application rate range for the two systems can be seen in Figure 4.6. The two lowest application rate ranges saw slight increase in field area percentage, however larger deviations were noted in the highest two application rate ranges in Figure 4.6. The shifting of field area to lower application rate ranges can again be attributed here to the reduced pressure values. Overall, each application rate range remained similar to the Luck et al. (2011b) study. When the systems were subjected to an ANOVA and multiple comparison analysis, the two systems exhibited no significant difference. A more detailed analysis revealed that no application rate ranges were significantly different between the two systems.

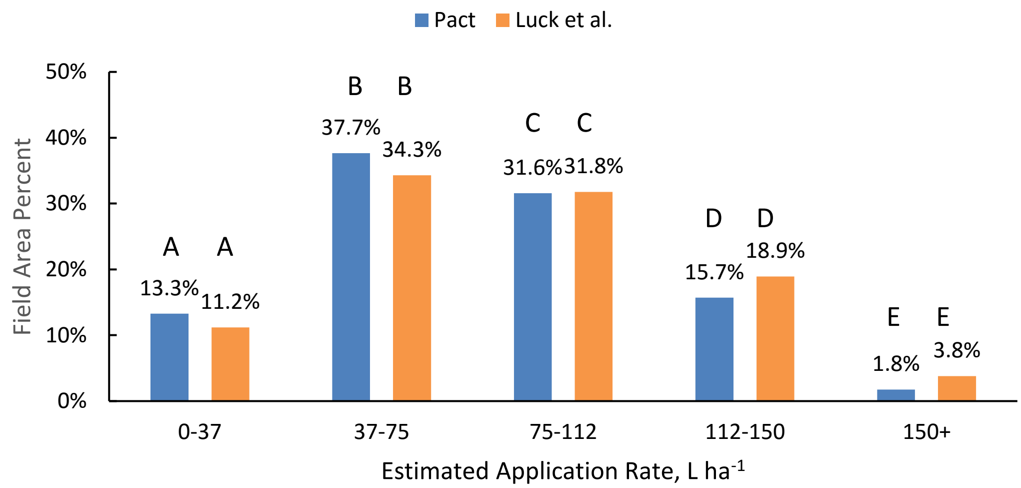


Figure 4.6: Field area percentage averages from the 3 fields analyzed in the Luck et al. (2011b) study and as analyzed by the PACT sprayer performance feedback system (different letters within each application rate range indicate significant differences between PACT and Luck et al. (2011b) system estimates).

Table 4.4: ANOVA results for Luck et al. (2011b) and PACT comparison by band.

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Program	0.000	1	0.00	0.00	0.999
Application Rate Range	4357.460	4	1089.36	57.36	0.000
Field	0.000	2	0.00	0.00	1.000
Program*Application Rate Range	28.420	4	7.11	0.37	0.821
Program*Field	0.000	2	0.00	0.00	1.000
Application Rate Range*Field	205.930	8	25.74	1.36	0.339
Error	151.930	8	18.99		
Total	**4743.730**	**29**			

Because off-rate errors were summarized for the three study fields by Luck (2011), further analysis was conducted to compare off-rate errors estimated from the Luck et al. (2011b) data and PACT program. Comparison application rate ranges remained on either side of target rate ±10%: 0% to 40%, 40% to 90%, 110% to 160%, and greater than 160% (compared to the target rate of 93.5 L ha^{-1}). A comparison of the average field areas affected by off-rate errors within the five application rate ranges is shown in Figure 4.7. The average change in field area percentage for the four application rate ranges spanned from -2.9% to 5.2%. The shift of field area to lower application rate ranges under the PACT program remained consistent with the earlier findings due to the reduced pressure values used. Statistical analysis revealed a significant difference between the two systems in the 0 to 37 L ha^{-1} and 103 to 150 L ha^{-1} rate ranges. However, when the two systems were analyzed across the four application rate ranges as a whole there was no significant difference indicated between the systems. The overall lack of significant difference between the two systems was a positive step in confirming the PACT program could successfully account for off-rate application errors.

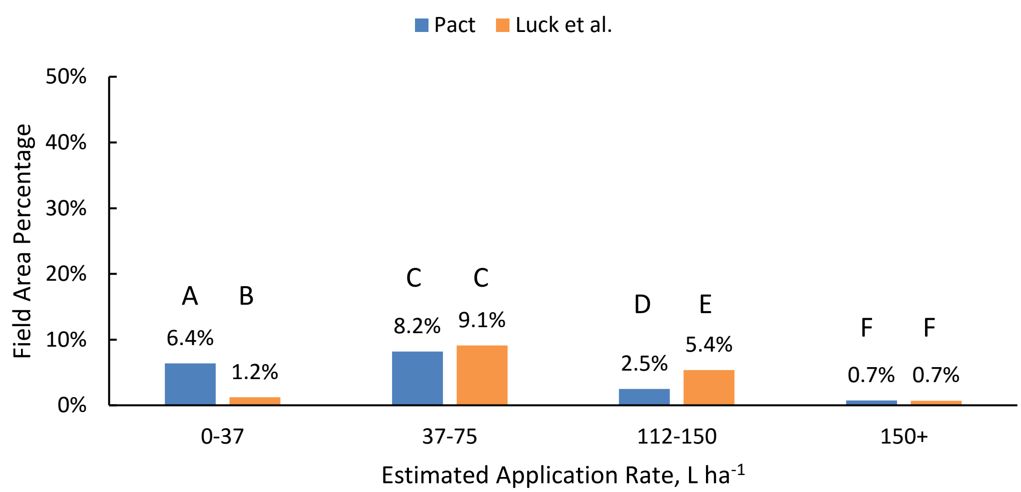


Figure 4.7: Field area percentage averages of off-rate turning errors from the 3 fields analyzed in the Luck et al. (2011b) study and as analyzed by the PACT sprayer performance feedback system (different letters within each application rate range indicate significant differences between PACT and Luck et al. system estimates).

Table 4.5: ANOVA results for Luck et al. (2011b) and PACT comparison by band for off-rate turning errors.

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Program	0.058	1	0.06	0.10	0.758
Application Rate Range	197.968	3	65.99	118.38	0.000
Field	19.382	2	9.69	17.38	0.003
Program*Application Rate Range	39.341	3	13.11	23.52	0.001
Program*Field	1.089	2	0.54	0.98	0.429
Application Rate Range*Field	41.617	6	6.94	12.44	0.004
Error	3.345	6	0.56		
Total	**302.800**	**23**			

4.4.3 Informing Technology Adoption

The section control (manual versus automatic) comparison analysis most affected off-target sprayer performance errors. Off-target field area percentages saw an average increase of 12.4% across the five fields analyzed when modeling full boom manual control as opposed to the seven section automatic boom control present on the sprayer. An ANOVA test ran for off-target field area percentages was ran between the two boom section control types, and suggested a significant difference between the seven section automatic boom control and manual full boom control types. In response to the increased off-target errors, fields examined saw an average 11.3% percent change in average field wide application rate as it rose from 141.2 L ha^{-1} to 157.1 L ha^{-1}. Off-target error increase remained visibly evident across each of the five fields. Figure 4.8 below shows as-applied map comparisons for the seven section control data compared to the modeled full boom manual control.

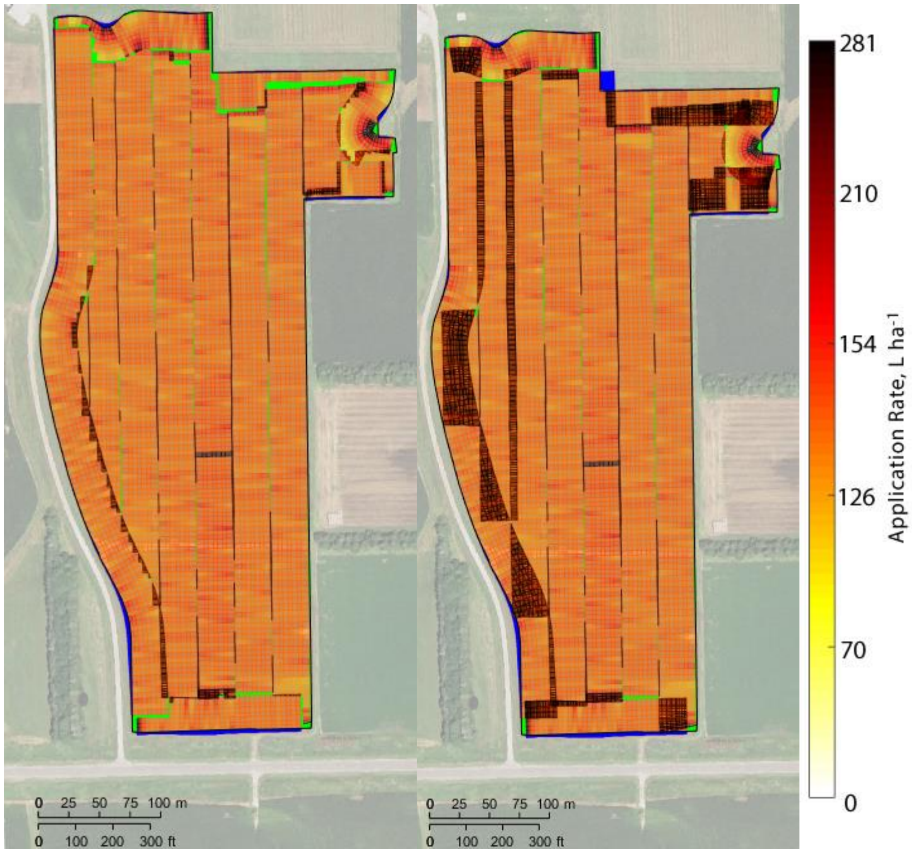


Figure 4.8: Comparison between as-applied map with automatic section control (left) versus manual control (any section 'on' resulted in all sections simulated as 'on') for one field (right).

An assumed pesticide mix of MSO, Corvus, Buccaneer, 2,4D-LV6, and Atrazine 4L was used for an economic analysis of automatic section control adoption. Per unit purchase prices and recommended application rates were used to calculate an estimated value of $48.00 per acre of application for the pesticide mix (Table 4.1). PACT program reports from the five fields yielded an average 3.4% of field area receiving off-target application when seven section automatic boom control was used. Assuming no other errors present during those off-target errors an average wasted chemical cost of $1.65 per acre was assumed during application. When the same fields were modeled assuming manual full boom control the average percent of field area receiving off-target errors rose to 15.8%, raising the wasted chemical costs to $7.60 per acre. Applying those numbers to an average Nebraska farm of 900 acres (USDA 2017) would add up quickly; a per acre savings of $5.95 from seven section automatic boom control across such a farm resulted in an estimated yearly savings of $5,355. The estimated savings were based off a single post-emergence application per growing season. At an assumed cost of around $5,000, an added automatic section control unit would reach a break-even cost in one growing season.

Table 4.6: Estimated yearly savings resultant from added seven section automatic boom control.

	Manual Full Boom Control		Seven Section Automatic Boom Control	
Farm Size		900		900
Per Acre Pest. Cost	$	48.00	$	48.00
Estimated Off-target Field Area		15.8%		3.4%
Excess Per Acre Pesticide Cost	$	7.60	$	1.65
Total Excess Pesticide Cost	$	6,840.35	$	1,481.22
Savings			**$**	**5,359.12**

4.5 Conclusions

A continued need for an improved sprayer feedback system was furthered through reported PACT findings. Substantial field area percentages were found to remain outside of desired application rate ranges. PACT visual and quantified error outputs were successfully compared against one widely used commercial FMIS and one previously published research study regarding high resolution as-applied mapping. Visual comparisons yielded key similarities, while demonstrating new capabilities previously not included in these systems. Statistical analyses on quantified error reports found few implied significant differences for field average metrics, which supported the PACT program ability to successfully identify in-field application rates. When comparisons of total error were made on a per rate range basis significant differences were implied for two rate ranges between the PACT program and SMS system. These differences showcased the lack of detail present among many current FMIS systems. The PACT program offered increased detail regarding application rate errors exceeding ±10% of the target rate along with locations of errors, while the SMS system primarily grouped application rates within ±10% of the target rate range. The PACT program, as with the Luck et al. 2011b study began to account for all of the various errors present in a spraying operation. When observing only errors defined as off-rate errors the PACT program and Luck et al. (2011b) systems displayed no significant difference between the systems, confirming successful identification of off-rate errors by the PACT program. Expected trends due to newly included error types in the PACT program were observed or accounted for during comparison of error totals by application rate range. Quantified error reports generated using the PACT program's regular and "section control" analysis modes proved capable assets for determining potential savings of automatic boom section control. A break even analysis was successfully performed for the scenario of the seven section automatic boom control used in the fields in this study. The ability to model the difference in manual full boom control and automatic section control set up is a significant step in providing producers with decision making support.

Chapter 5. Summary and Future Work

Efforts were successful in development of a data analytics tool for the use of improved pesticide application operator feedback. An automated system was designed to handle CAN bus-based data from previously existing sprayer instrumentation to generate high resolution as-applied maps and quantified error reports. The improved high resolution as-applied maps distinguished off-target and off-rate error locations throughout the field and showed potential for aiding in better management practice planning for reduced error. Post-application error reports were generated on a per field basis reporting error totals in units of area and field area percentage. Errors were broken down by application rate ranges to further aid in determining the origin or the error. PACT program report findings were compared against current FMIS reports and showed similarities in field-average metrics, verifying the validity of the PACT program. As expected, several differences were noted between the PACE program and current FMIS reports on a more detailed level as previously omitted errors were accounted for. Overall, the system provided a platform to continue moving forward toward a mo

re true reporting of sprayer performance feedback and aiding in developing better management practices for chemical application operations.

Development of more analysis modes for the PACT program would serve as a great next step. Inclusion of a modeling system for analyzing an increased number of boom sections would serve as a great decision making tool for producers. Necessary for an improved section control analysis mode would be an algorithm for determining assumed on/off states for “new” theoretical sections being added, similar to that modeled in the Zandonadi et al. (2013) study. Error reduction totals would serve as a way for producers to begin comparing breakeven analyses. A second analysis mode modeling variable rate sprayer technologies would again serve well for technology adoption decision making. Key developments in modeling section based pressure or flow rate values through turning movements would be necessary for the inclusion of such analysis mode.

On a larger scale the PACT program can serve as a starting point for development of improved spraying practices as well as a teaching tool for operators. Future programs based upon PACT program styled reports could be developed for improved management practices such as path planning. By weighing the consequences of various in field movements (e.g., sharp turns, rapid acceleration or deceleration, etc.) path planning algorithms could be modified to reduce application errors. As agriculture machinery continues towards more automated systems performance feedback will become key in developing necessary mathematical models for the required control systems. More immediately, results from various scenarios could be compiled and serve as case studies to educate operators on various driving choices and the resulting consequences. The main goal remains the reduction of misapplied chemicals and the potentially harmful impacts they can have.

References

ASABE Standards. (2016). S327.4: Terminology and Definitions for Applications of Crop or Forestry Production and Protective Agents. St. Joseph, MI: ASABE.

Darr, M.J. (2012). CAN bus technology enables advanced machinery management. *Resource: Engineering & Technology for a Sustainable World, 19.5*, 10-11.

Erickson, B. and D.A. Widmar. (2015). 2015 Precision Agricultural Services Dealership Survey Results. Dept. of Agricultural Economics/Dept. of Agronomy, Purdue University. Available at: http://agribusiness.purdue.edu/files/resources/2015-crop-life-purdue-precision-dealer-survey.pdf. Accessed 01/30/2018.

Giles, D.K., and J.A. Comino. (1990). Droplet size and spray pattern characteristics of an electronic flow controller for spray nozzles. *Agric. Eng. Res., 47*(4), 249-267.

Giles, D. K., Downey, D. (2003). Quality Control Verification and Mapping for Chemical Application. *Precision Agriculture, 4*, 103-124. https://doi.org/10.1023/A:1021871207195

Grisso, R. D., Hewett, E. J. , Dickey, E. C., Schnieder, R. D., Nelson, E. W. (1988). Calibration Accuracy of Pesticide Application Equipment. *Applied Engineering in Agriculture, 4*(4), 310–315. https://doi.org/10.13031/2013.26624

Kortenbruck, D., Griepentrog, H. W., Paraforos, D. S. (2017). Machine operation profiles generated from ISO 11783 communication data. *Computers and Electronics in Agriculture*, *140*, 227–236. https://doi.org/10.1016/j.compag.2017.05.039

Luck, J.D., S.K. Pitla, S.A. Shearer, T.G. Mueller, C.R. Dillon, J.P. Fulton, and S.F. Higgins. (2010a). Potential for pesticide and nutrient savings via map-based automatic boom section control of spray nozzles. *Computers Electronics Agric., 70*(1), 19-26.

Luck, J.D., R.S. Zandonadi, B.D. Luck, and S.A. Shearer. (2010b). Reducing pesticide over-application with map-based automatic boom section control on agricultural sprayers. *Trans. ASABE, 53*(3), 685-690.

Luck, J.D., R.S. Zandonadi, and S.A. Shearer. (2011a). A case study to evaluate field shape factors for estimating overlap errors with manual and automatic section control. *Trans. ASABE, 54*(4), 1237-1243.

Luck, J.D., A. Sharda, S.K. Pitla, J.P. Fulton, and S.A. Shearer. (2011b). A case study concerning the effects of controller response and turning movements on application rate uniformity with a self-propelled sprayer. *Trans. ASABE, 54*(2), 423-431.

Luck, J.D., S.K. Pitla, R.S. Zandonadi, M.P. Sama, and S.A. Shearer. (2011c). Estimating off-rate pesticide application errors resulting from agricultural sprayer turning movements. *Precision Agric.*, *12*(4), 534-545.

Marx, S.E., J.D. Luck, R.M. Hoy, S.K. Pitla, M.J. Darr, and E. Blankenship. (2015). Validation of machine CAN Bus J1939 fuel rate accuracy using Nebraska Tractor Test Laboratory fuel rate data. *Computers Electronics Agric.*, *118,* 179-185.

Marx, S.E., J.D. Luck, R.M. Hoy, and S.K. Pitla. (2016). Comparing various hardware/software solutions and conversion methods for Controller Area Network (CAN) bus data collection. *Computers Electronics Agric.*, *128*, 141-148.

Nebraska Extension. (2014). 2014 Guide for Weed, Disease, and Insect Management in Nebraska. Lincoln, NE: Nebraska Extension.

Pitla, S.K., L. Nin, J.D. Luck, and S.A. Shearer. (2015). Use of controller area network (CAN) data to determine field efficiencies of agricultural machinery. *App. Eng. Agric.*, *30*(6), 829-839.

Pitla, S.K., J.D. Luck, N. Lin, J.P. Werner, and S.A. Shearer. (2016). In-field fuel use and load states of agricultural machinery. *Computers Electronics Agric.*, *121*, 290-300.

Porter, W.M., J.A. Rascon, Y. Shi, R.K. Taylor, and P.A. Weckler. (2013). Laboratory evaluation of a turn compensation control system for a ground sprayer. *Applied Eng. Agric.*, *29*(5), 655-662.

Sama, M.P., J.D. Luck, and T.S. Stombaugh. (2015). Scalable control architecture for variable-rate turn compensation. *App. Eng. Agric.*, *31*(3), 425-435.

Sharda, A., J.D. Luck, J.P. Fulton, T.P. McDonald, and S.A. Shearer. (2013). Field application uniformity and accuracy of two rate control systems with automatic section capabilities on agricultural sprayers. *Precision Agric.*, *14*(3), 307-322.

Stone, M.L., R.K. Benneweis, and J.V. Bergeijk. (2008). Evolution of Electronics for Mobile Agricultural Equipment. *Trans. ASABE.*, *51*(2), 385-390.

UNL Crop Watch. (2018). 2018 Nebraska Crop Budgets: Corn Dryland. Available at: https://cropwatch.unl.edu/budgets. Accessed 02/05/2018

U.S. EPA. (2017). Pesticides Industry Sales and Useage: 2008-2012 Market Estimates. Availabe at: https://www.epa.gov/sites/production/files/2017-01/documents/pesticides-industry-sales-usage-2016_0.pdf. Accessed 01/30/2018.

USDA. (2012). Census of Agriculture, United States, Summary and State Data. Available at:

https://www.agcensus.usda.gov/Publications/2012/Full_Report/Volume_1,_Chapter_1_US/. Accessed 01/30/2018.

USDA. (2017). Farm Income and Wealth Statistics. Available at: https://data.ers.usda.gov/reports.aspx?ID=17834. Accessed 01/30/2018.

Zandonadi, R.S, J.D. Luck, T.S. Stombaugh, M.P. Sama, and S.A. Shearer. (2011). A computational tool for estimating off-target application areas in agricultural fields. *Trans. ASABE*, *54*(1), 41-49.

Zandonadi, R.S., J.D., Luck, T.S. Stombaugh, and S.A. Shearer. (2013). Evaluating field shape descriptors for estimating off-target application area in agricultural fields. *Computers Electronics Agric.*, *96*, 217-226.

Appendices

Appendix A: Retrofitting Sprayer Systems for Necessary Data Collection

In the case a sprayer is not equipped with the necessary sensors, or questions regarding accuracy arise, machinery can easily be retrofitted with the necessary components. Flowmeter and/or pressure transducers can be introduced into the boom system and, through use of a Danfoss microcontroller (MC024-120, Danfoss Power Solutions Company, Ames, IA), sensor readings can be published across the sprayer or implement BUS. Resulting CAN messages can be published through previously described data collection methods.

Step 1: Data sheets for the Raven flowmeter (RFM100) and Danfoss microcontroller (MC024) controller suggested compatibility. Power requirements for the MC024 were satisfied by standard power supply present at the sprayer's in-cab diagnostic port. The MC024 possessed sufficient power output and ground pins (i.e., C1-P8 and C1-P9, respectively) for powering the RFM100. The MC024's ability to read sensor frequency output signals (i.e., C2-P1 and C2-P2) pins made the microcontroller ideal for flowmeter data collection. In the case of the inclusion of a pressure transducer using an analog output multiple MC024 (i.e., C2-P1 and C2-P2) pins are capable of recording the transducer signals. MC024 microcontrollers support j1939 CAN protocols, and have available CAN hi and CAN low pins for communication (i.e., C1-P3 and C1-P4, respectively).

Step 2: After confirming a compatible microcontroller and sensors had been selected, a wiring diagram was developed for sensor, microcontroller and sprayer connectivity. The sprayer's in-cab diagnostic was selected for the interfacing connection. A sufficient wiring diagram for sensor setup can be seen in FIG.

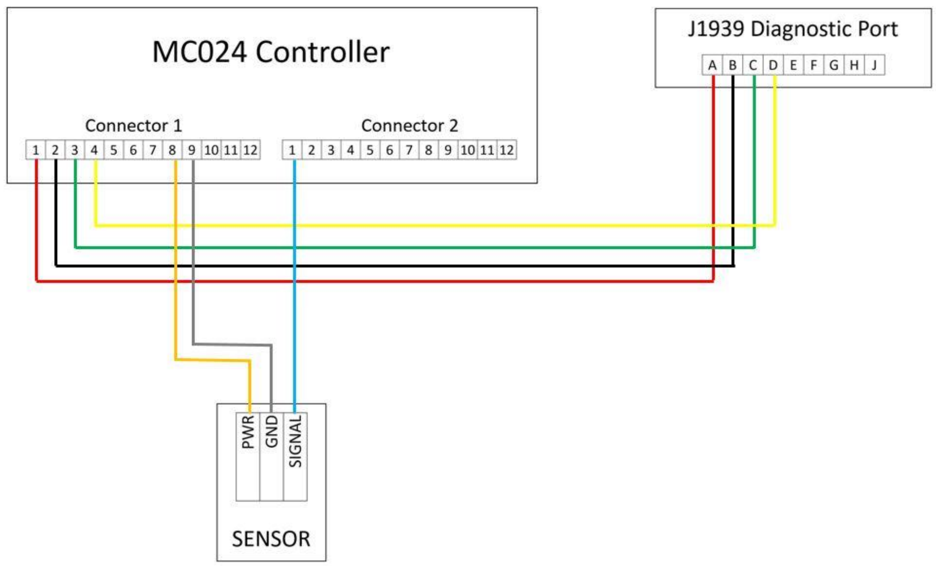


Figure A.0.1: Wiring diagram for publishing RFM100 signals to the sprayer's tractor CAN bus.

Step 3: Using Danfoss Plus+1 Guide a program to read sensor values and publish messages containing the reading in the data bytes to the system's CAN bus was developed.

a. The appropriate hardware template was selected and dragged into the main work area, in this case the MC024-110/112 template.

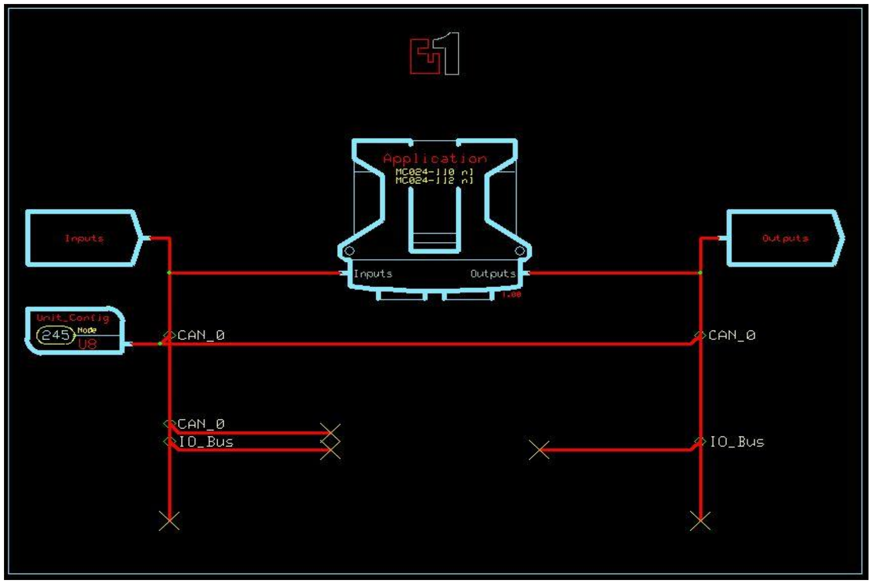


Figure A.0.2: Top level screen for Danfoss Plus+1 sprayer instrumentation code.

b. Entering into the MC024-110/112 page, on the left hand side of the page the R hot-key was used to start a wire segment, using the K hot-key switch the wire type until the bus wire type (i.e., bold red wire) was selected. The bus wire was connected to the Inputs bus. Once connected use the C2p03 pin was selected for reading the RFM100 frequency signal in this case. The process was repeated on the right side on the right hand side of the work area, this time connecting CAN_0 to the Outputs bus.

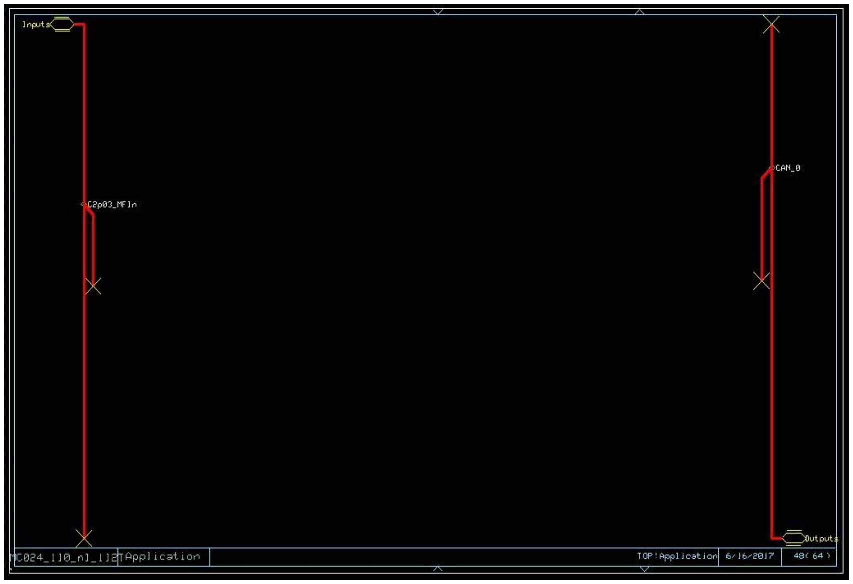


Figure A.0.3: Addition of bus wire segments to instrumentation code.

c. In the SD Basic FB Library the CAN_Tx function block was selected and placed onto the work screen. Using the R hot-key, the Port output on the CAN_Tx block was connected to the CAN_0 bus segment using a data wire type (i.e., green wire). Port was selected as the connection type. A True component block was placed onto the work screen and connected to the Extended output on the CAN_Tx function block using a data wire. A Multi-character component block was placed onto the work screen. Using the Q hot-key to change the constant type to U32, the extended CAN message ID was entered. Recommended CAN extended IDs are: 0x18FF0080, 0x18FF0081, 0x18FF0082. A data wire was used to connect the multi-character component block to the ID output of the CAN_Tx function block. From the Data output of the CAN_Tx function block bus wire segment was branched off using the bus wire type. Six separate 3 Character component blocks were placed onto the work screen. For each 3 Character component block the Q hot-key was used to set the type to U8. Place holder values were entered for unused bytes in the CAN message. Each 3 Character component block was wired using the data wire to the bus wire segment branched from the Data. Once connected "CAN_Data_0" and so on, were selected for the value names. Two empty spots were left unused for sensor data to be stored in.

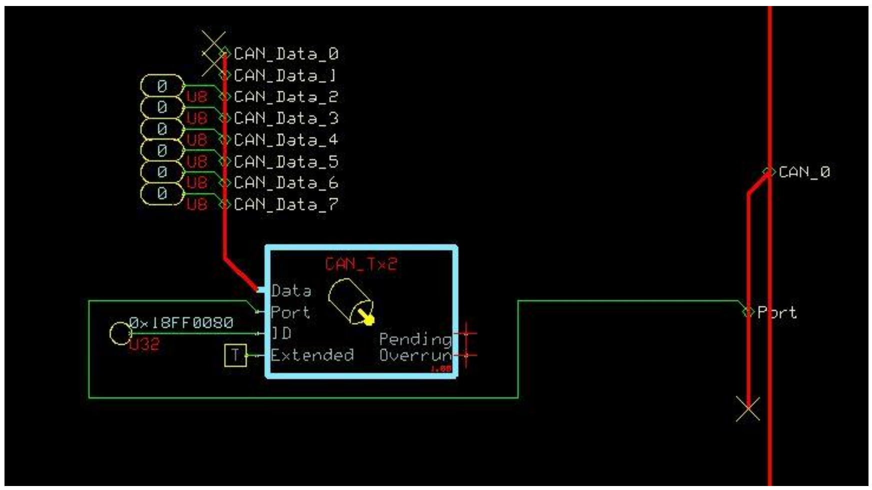


Figure A.0.4: CAN_Tx block component used for publishing recorded sensor values across sprayer implement bus.

d. Incoming frequency pulses per second were converted to useable units of gallons per minute. A Multiply component block and a Divide component block were placed onto the work screen, as well as two 3 Digit Autotype component blocks. In the first 3 Digit Autotype component block a value of 60 was entered to convert seconds to minutes, and was connected to the Multiply component block. A manufacturer provided calibration for pulses per gallon was used for the second 3 Digit Autotype component block connected to the Divide component block. The Multiply input was connected to the previously created bus wire on the left hand side, with Freq selected as the input type. The Multiply output was connected to the Divide block input. Along the wire segment from the input to the Multiply component block provided an effective place to place a checkpoint to be used to verify the input value being read.

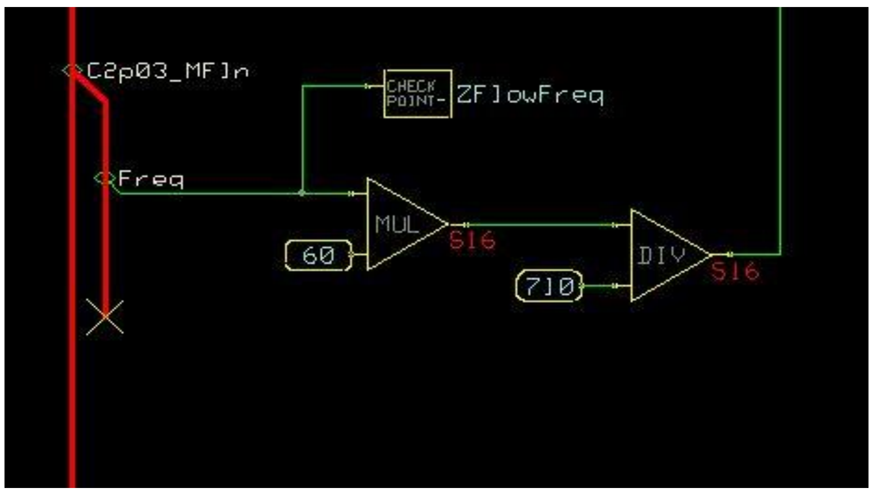


Figure A.0.5: Conversion of pulses per second flow meter sensor readings to gallons per minutes units.

e. The now adjusted input value needed to be split from a 16 bit integer into two smaller 8 bit integers. The 8 bit integers were stored into two data bytes on the CAN message. A Split component block was selected to separate the 16 bit integer. The gallon per minute value from the Divide component block was connected to the front end of the Split component block, and subsequently connected to the two empty CAN data bytes created in Step 3c.

Figure A.0.6: Splitting 16-bit sensor value to two 8-bit values to be stored in CAN message.

f. Back at the Top Level page the Unit_Config page was used to check the MC024's baudrate setting. The Baudrate input was set to 500 kbd to match that of the sprayer's tractor bus.

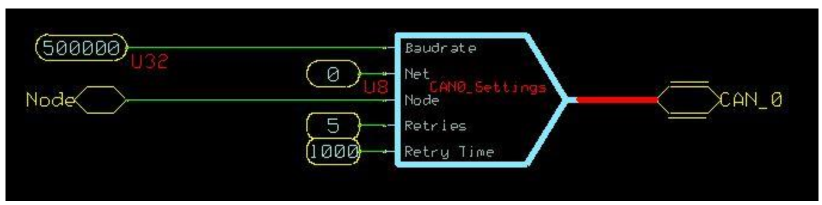


Figure A.0.7: Setting of CAN channel baud rate to match John Deere 4830 sprayer's implement bus baud rate.

Step 5: Following setup of the MC024 program any required sensors would need to be installed into the sprayer's boom system. Recommended positioning for either flow meter or pressure transducer sensors are up system of the sprayer's manifold block. The flow meter is recommended to be placed in a horizontal position to reduce any chances of cavitation and produce the most accurate readings. Care should be taken no to reduce the system's hoses inside diameter.

Step 6: After installation published CAN messages by the MC024 can be recorded in conjunction with pre-existing sprayer CAN messages using a Kvaser Memorator.

Appendix B: Kvaser Memorator 2xHS V2 Data Sheet

KVASER

KVASER MEMORATOR PRO 2XHS V2

EAN 73-30130-00819-9

Major Features

- Monitor two CAN channels simultaneously using just one device.
- Log data to an expandable SD card slot.
- Supports CAN FD up to 8 Mbit/s (with proper physical layer implementation).
- Script functionality allows users to develop customised t-script applications written in the Kvaser t programming language.
- Power derived from the USB connection, CAN or an in-built power supply.
- Supports silent mode for analysis tools – listens to the bus without interfering.
- Detection and generation of error frames and remote frames.
- LED lights alert user to device status, including signaling a full SD card or card error.
- Galvanically isolated CAN bus drivers.
- Automatically time-synchronises the data transmitted and received across both buses.
- Built-in Kvaser MagiSync™ technology time-synchronises with other Kvaser interfaces connected to the same PC, resulting in simpler and more accurate multichannel data capture.
- Extended operating temperature range from -40 to 85 °C.
- Compatible with J1939, CANopen, NMEA 2000® and DeviceNet.
- Plug-and-play installation, and a comprehensive user guide to help make t-script development quick and easy.

Technical Data

CAN Bit Rate	50 kbit/s to 1 Mbit/s
CAN FD Bit Rate	Up to 8 Mbit/s (with proper physical layer)
Temp Range	-40° C to +85° C
Power Consumption	Up to 3W
Height	23 mm
Length	150 mm
Width	55 mm
Weight	150 g
Messages Per Second Sending	20000 msg/s per channel
Messages Per Second Receive	20000 msg/s per channel
Channels	2
Certificates	CE, RoHS
PC Interface	USB 2.0
OS	Linux, Win Vista, Win 10, Win 8, Win 7, Win XP
Galvanic Isolation	Yes
Error Frame Generation	Yes
Error Frame Detection	Yes
Silent Mode	Yes
Time Stamp Resolution	1 ms
MagiSync	Yes

WARRANTY
2-Year Warranty. See our General Conditions and Policies for details.

SUPPORT
Free Technical Support on all products available by contacting support@kvaser.com

SOFTWARE
Documentation, software and drivers can be downloaded for free at: www.kvaser.com/downloads

Kvaser CANlib SDK is a free resource that includes everything you need to develop software for the Kvaser CAN interfaces. Includes full documentation and many program samples, written in C, C++, C#, Delphi, Java, Python, and Visual Basic.

All Kvaser CAN interface boards share a common software API. Programs written for one interface type will run without modifications on the other interface types!

J2534 Application Programming Interface available.

RP1210A Application Programming Interface available.

Online documentation in Windows HTML-Help and Adobe Acrobat format.

Appendix C: PACT Generated Field Maps and Reports

This appendix contains PACT generated as-applied maps and quantified error reports for eight Nebraska fields used in this study. Fields from the Luck et al. (2011b) study have been omitted as they did not include all data necessary for generating full as-applied maps and quantified error reports.

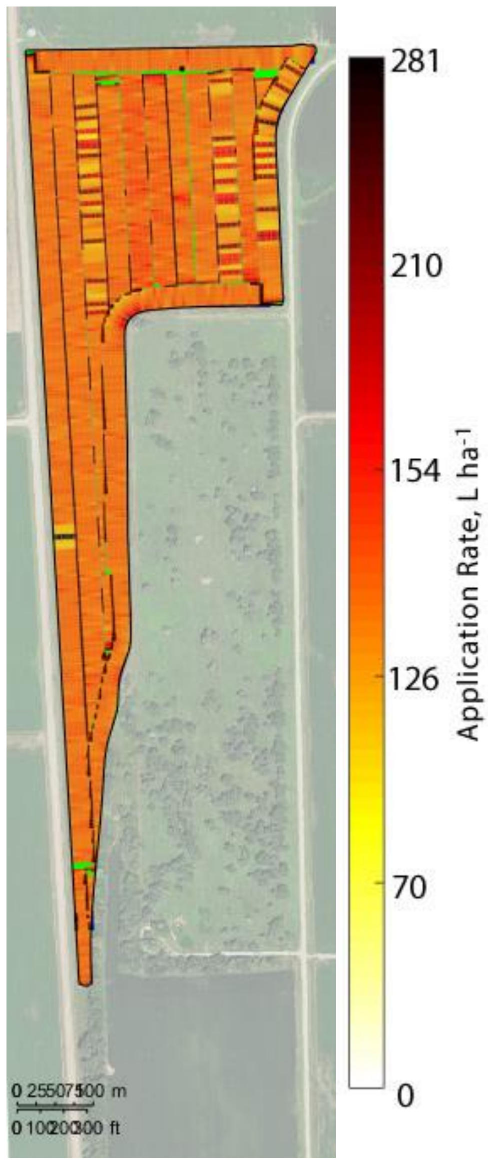


Figure C.1: PACT generated improved high resolution as-applied map for Nebraska Field 1401 based off Farmobile data set.

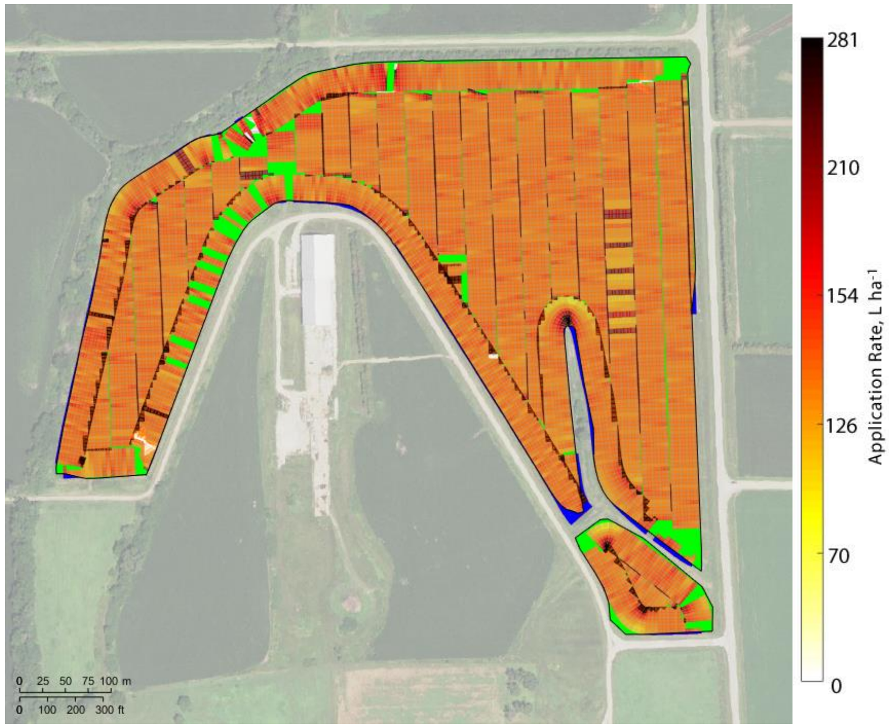


Figure C.2: PACT generated improved high resolution as-applied map for Nebraska Field 1404 based off Farmobile data set.

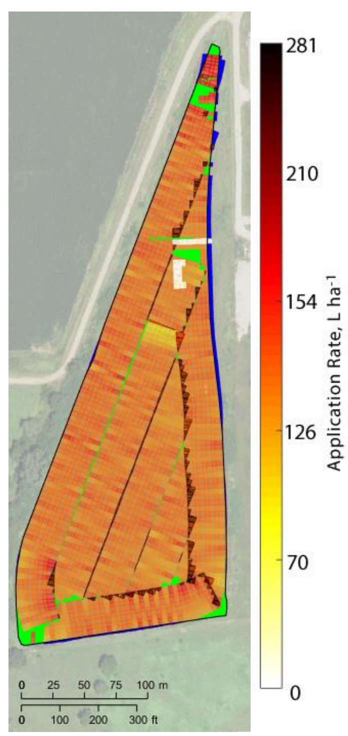


Figure C.3: PACT generated improved high resolution as-applied map for Nebraska Field 2306 based off Farmobile data set.

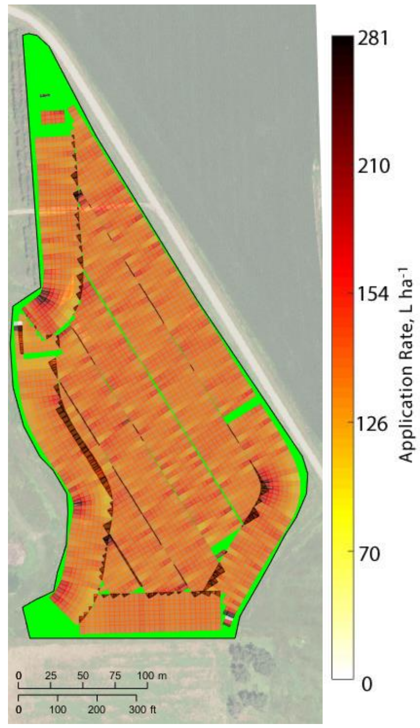


Figure C.4: PACT generated improved high resolution as-applied map for Nebraska Field 2311 based off Farmobile data set.

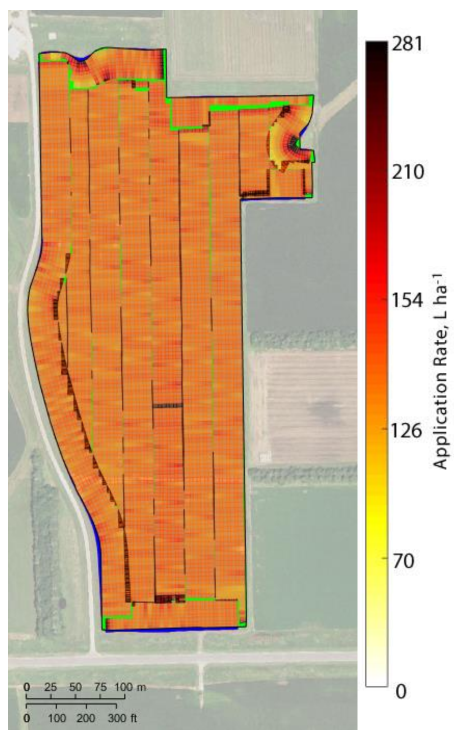


Figure C.5: PACT generated improved high resolution as-applied map for Nebraska Field 2635 based off Farmobile data set.

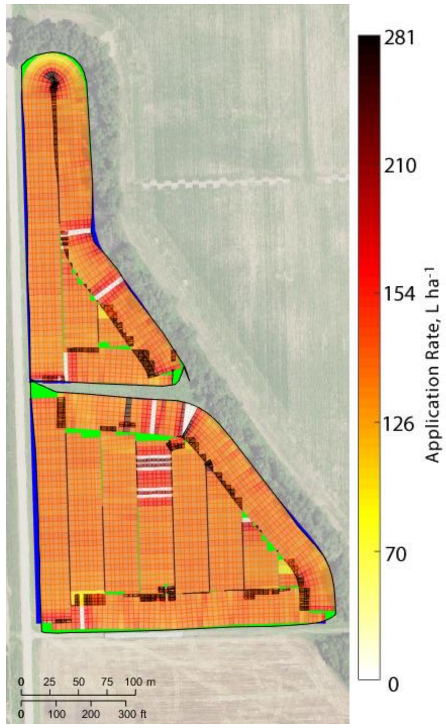


Figure C.6: PACT generated improved high resolution as-applied map for Nebraska Field 2017 based off Farmobile data set.

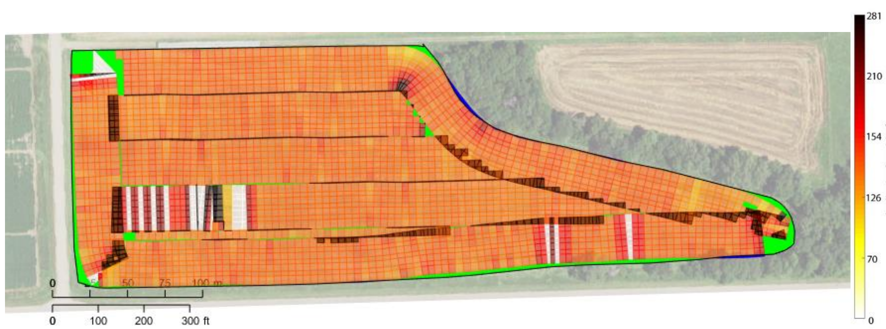


Figure C.7: PACT generated improved high resolution as-applied map for Nebraska Field 2018 based off Farmobile data set.

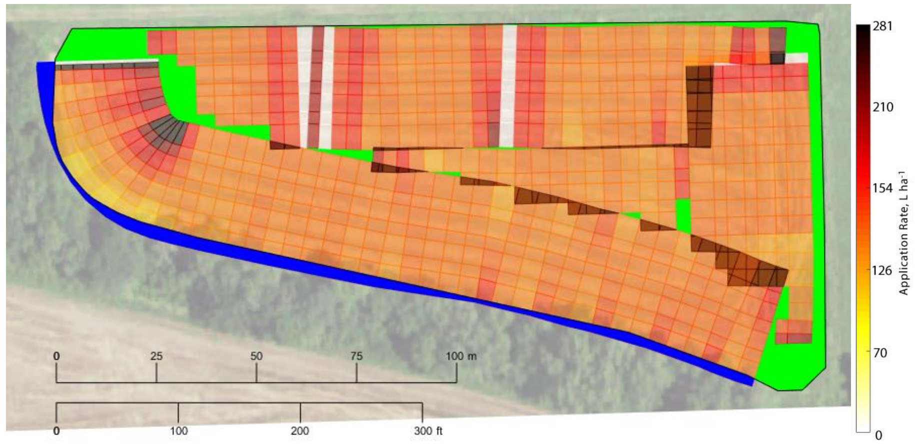


Figure C.8: PACT generated improved high resolution as-applied map for Nebraska Field 2020 based off Farmobile data set.

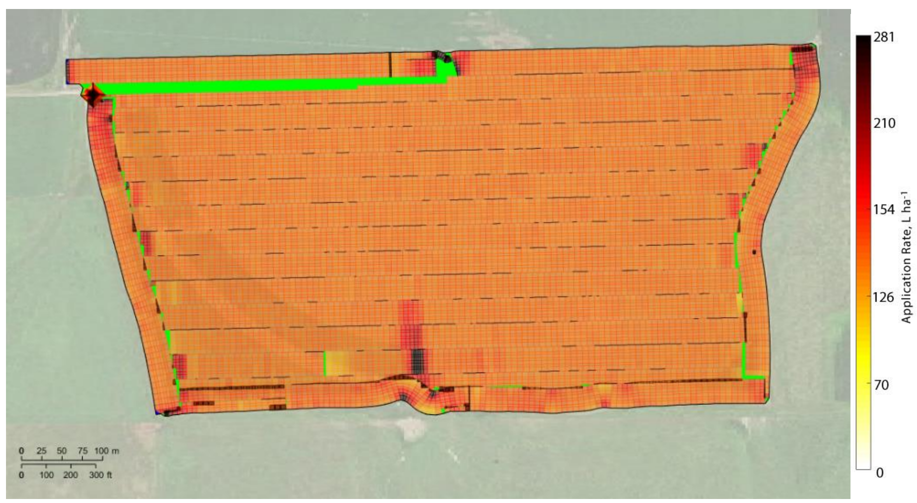


Figure C.9: PACT generated improved high resolution as-applied map for Nebraska Field 1864 based off Kvaser data set.

Nebraska Lincoln | N EXTENSION

PACT

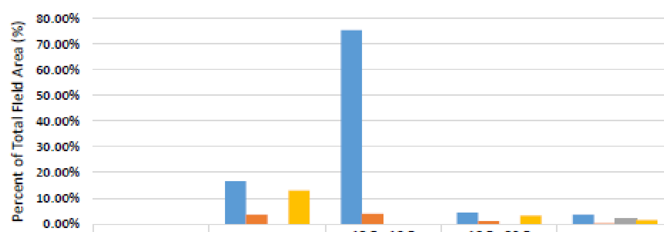
Sprayer Coverage Analysis Report

Field:	Field 1401	Control Sections:	7
Sprayed:	4/14/2016	Nozzle Spacing:	20 in
Analyzed:	2/18/2018	Nozzle Tip:	TeeJet XR11006

Field Size:	44.00 ac	Total Gallons	641.2 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	5.0 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	11.1 gal
Total Area Applied:	43.61 ac		
Total Area Skipped:	0.73 ac		
Total Area Outside:	0.33 ac		
Total Off-Rate Area:	10.72 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**10.72**	**24.4%**
Turning Error	2.19	5.0%
Controller Response Error	7.74	17.6%
Off-Target Error	0.78	1.8%



	0 - 7.5 gal/ac	7.5 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 22.5 gal/ac	22.5+ gal/ac
Total Off-Rate	0.00%	16.62%	75.41%	4.38%	3.59%
Turning Off-Rate	0.00%	3.59%	3.99%	1.16%	0.29%
Overlap Off-Rate	0.00%	0.00%	0.00%	0.00%	1.79%
Controller Response Off-Rate	0.00%	13.03%	0.00%	3.22%	1.51%

Figure C.10: PACT generated quantified error report for Nebraska Nebraska Field 1401.

Nebraska Lincoln — EXTENSION

PACT

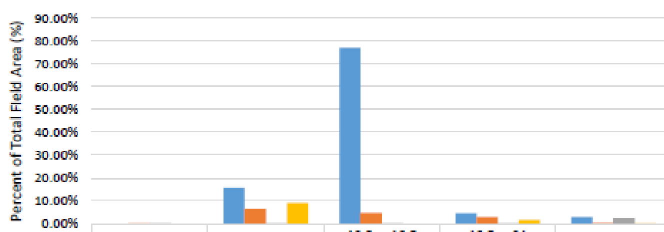
Sprayer Coverage Analysis Report

Field:	Field 1404	Control Sections:	7
Sprayed:	4/4/2016	Nozzle Spacing:	20 in.
Analyzed:	3/13/2018	Nozzle Tip:	XR11006

Field Size:	52.87 ac	Total Gallons	786.6 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	10.3 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	633.0 gal
Total Area Applied:	53.49 ac		
Total Area Skipped:	3.02 ac		
Total Area Outside:	0.70 ac		
Total Off-Rate Area:	11.67 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**11.67**	**22.1%**
Turning Error	4.96	9.4%
Controller Response Error	5.27	10.0%
Off-Target Error	1.44	2.7%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	-0.01%	15.74%	76.88%	4.44%	2.98%
Turning Off-Rate	0.04%	6.46%	4.66%	2.95%	0.38%
Overlap Off-Rate	0.16%	0.19%	0.16%	0.05%	2.46%
Controller Response Off-Rate	-0.21%	9.08%	0.00%	1.43%	0.15%

Figure C.11: PACT generated quantified error report for Nebraska Nebraska Field 1404.

Nebraska Lincoln | N EXTENSION

PACT

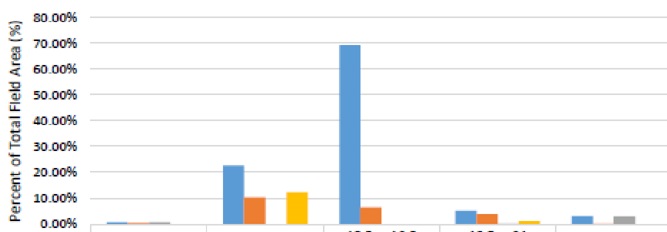
Sprayer Coverage Analysis Report

Field:	Field 2306	Control Sections:	7
Sprayed:	4/4/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	10.43 ac	Total Gallons	163.9 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	4.2 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	3.8 gal
Total Area Applied:	11.14 ac		
Total Area Skipped:	0.34 ac		
Total Area Outside:	0.26 ac		
Total Off-Rate Area:	3.23 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**3.23**	**31.0%**
Turning Error	1.50	14.3%
Controller Response Error	1.34	12.9%
Off-Target Error	0.39	3.7%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	0.66%	22.58%	69.18%	5.09%	2.96%
Turning Off-Rate	0.39%	10.21%	6.31%	3.85%	0.05%
Overlap Off-Rate	0.70%	0.00%	0.00%	0.15%	2.92%
Controller Response Off-Rate	-0.42%	12.38%	0.00%	1.09%	-0.01%

Figure C.12: PACT generated quantified error report for Nebraska Nebraska Field 2306.

PACT

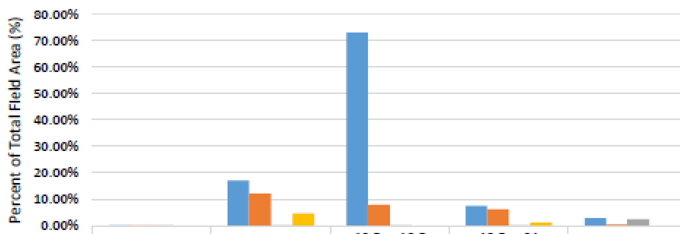
Sprayer Coverage Analysis Report

Field:	Field 2311	Control Sections:	7
Sprayed:	4/4/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	14.78 ac	Total Gallons	200.4 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	0.0 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	4.7 gal
Total Area Applied:	13.63 ac		
Total Area Skipped:	28.13 ac		
Total Area Outside:	0.00 ac		
Total Off-Rate Area:	3.51 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**3.51**	**23.8%**
Turning Error	2.44	16.5%
Controller Response Error	0.75	5.1%
Off-Target Error	0.33	2.2%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	0.07%	17.01%	72.99%	7.38%	2.72%
Turning Off-Rate	0.07%	12.20%	7.85%	6.15%	0.44%
Overlap Off-Rate	0.02%	0.11%	0.07%	0.04%	2.39%
Controller Response Off-Rate	-0.02%	4.71%	0.00%	1.19%	-0.11%

Figure C.13: PACT generated quantified error report for Nebraska Nebraska Field 2311.

PACT

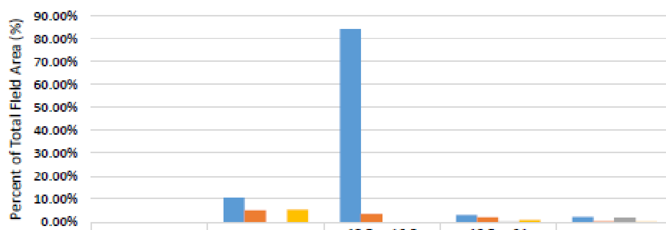
Sprayer Coverage Analysis Report

Field:	Field 2635	Control Sections:	7
Sprayed:	4/4/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	28.32 ac	Total Gallons	427.8 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	4.2 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	8.0 gal
Total Area Applied:	29.09 ac		
Total Area Skipped:	0.56 ac		
Total Area Outside:	0.28 ac		
Total Off-Rate Area:	4.46 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**4.46**	**15.7%**
Turning Error	2.09	7.4%
Controller Response Error	1.85	6.5%
Off-Target Error	0.53	1.9%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	0.00%	10.64%	84.10%	2.98%	2.28%
Turning Off-Rate	0.00%	5.12%	3.62%	2.01%	0.31%
Overlap Off-Rate	0.00%	0.00%	0.00%	0.07%	1.81%
Controller Response Off-Rate	0.00%	5.52%	0.00%	0.91%	0.16%

Figure C.14: PACT generated quantified error report for Nebraska Nebraska Field 2635.

PACT

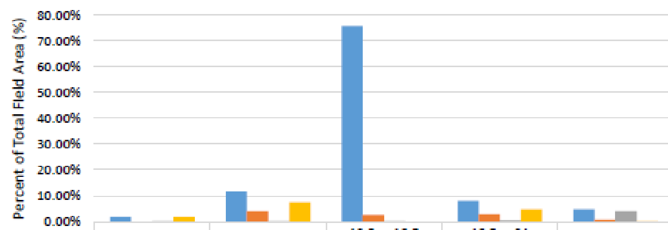
Sprayer Coverage Analysis Report

Field:	Field 2017	Control Sections:	7
Sprayed:	9/27/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	16.15 ac	Total Gallons	257.0 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	6.0 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	11.1 gal
Total Area Applied:	17.48 ac		
Total Area Skipped:	0.53 ac		
Total Area Outside:	0.40 ac		
Total Off-Rate Area:	4.12 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**4.12**	**25.5%**
Turning Error	1.21	7.5%
Controller Response Error	2.19	13.6%
Off-Target Error	0.71	4.4%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	1.82%	11.58%	75.68%	8.04%	4.73%
Turning Off-Rate	0.00%	4.06%	2.70%	2.94%	0.72%
Overlap Off-Rate	0.03%	0.06%	0.09%	0.42%	4.00%
Controller Response Off-Rate	1.79%	7.46%	0.00%	4.67%	0.01%

Figure C.15: PACT generated quantified error report for Nebraska Nebraska Field 2017.

Nebraska Lincoln | N EXTENSION

PACT

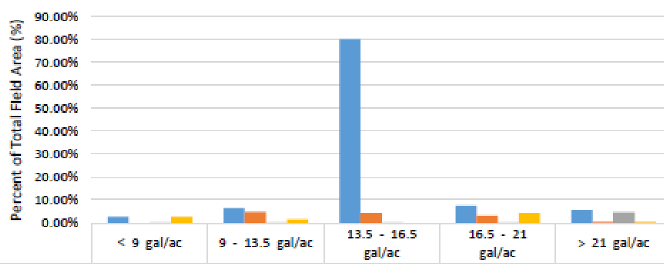
Sprayer Coverage Analysis Report

Field:	Field 2018	Control Sections:	7
Sprayed:	9/27/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	13.63 ac	Total Gallons	213.4 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	0.7 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	9.5 gal
Total Area Applied:	14.51 ac		
Total Area Skipped:	0.47 ac		
Total Area Outside:	0.04 ac		
Total Off-Rate Area:	2.91 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**2.91**	**21.3%**
Turning Error	1.09	8.0%
Controller Response Error	1.18	8.7%
Off-Target Error	0.63	4.7%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	2.86%	6.46%	80.22%	7.63%	5.70%
Turning Off-Rate	0.00%	4.85%	4.37%	3.15%	0.51%
Overlap Off-Rate	0.05%	0.04%	0.15%	0.16%	4.69%
Controller Response Off-Rate	2.81%	1.57%	0.00%	4.32%	0.49%

Figure C.16: PACT generated quantified error report for Nebraska Nebraska Field 2018.

University of Nebraska Lincoln — N EXTENSION

PACT

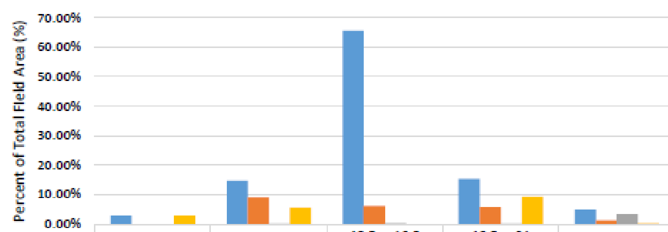
Sprayer Coverage Analysis Report

Field:	Field 2020	Control Sections:	7
Sprayed:	9/27/2016	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	3.15 ac	Total Gallons	47.9 gal
Target App. Rate:	15.00 gal/ac	Outside Gallons	1.9 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	1.4 gal
Total Area Applied:	3.25 ac		
Total Area Skipped:	0.24 ac		
Total Area Outside:	0.12 ac		
Total Off-Rate Area:	1.11 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**1.11**	**35.1%**
Turning Error	0.47	15.0%
Controller Response Error	0.53	16.8%
Off-Target Error	0.11	3.4%



	< 9 gal/ac	9 - 13.5 gal/ac	13.5 - 16.5 gal/ac	16.5 - 21 gal/ac	> 21 gal/ac
Total Off-Rate	2.88%	14.70%	65.55%	15.22%	4.88%
Turning Off-Rate	0.00%	9.06%	6.24%	5.80%	1.18%
Overlap Off-Rate	0.00%	0.07%	0.21%	0.12%	3.44%
Controller Response Off-Rate	2.88%	5.57%	0.00%	9.30%	0.26%

Figure C.17: PACT generated quantified error report for Nebraska Nebraska Field 2020.

Nebraska Lincoln — UNIVERSITY OF

N EXTENSION

PACT

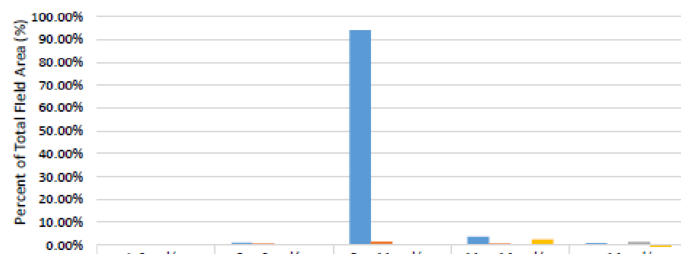
Sprayer Coverage Analysis Report

Field:	Field 1864	Control Sections:	7
Sprayed:	Summer 2017	Nozzle Spacing:	20 in.
Analyzed:	2/18/2018	Nozzle Tip:	XR11006

Field Size:	89.56 ac	Total Gallons	1325.1 gal
Target App. Rate:	10.00 gal/ac	Outside Gallons	0.2 gal
Average App Rate:	14.70 gal/ac	Overlap Gallons	185.4 gal
Total Area Applied:	90.12 ac		
Total Area Skipped:	1.89 ac		
Total Area Outside:	0.01 ac		
Total Off-Rate Area:	5.11 ac		

Area of field with application rates outside of stated tolerance:
Tolerance: ±10%

	Area, ac	Percent of Field
Total Off-Rate Error	**5.11**	**5.7%**
Turning Error	1.73	1.9%
Controller Response Error	1.92	2.1%
Off-Target Error	1.46	1.6%



	< 6 gal/ac	6 - 9 gal/ac	9 - 11 gal/ac	11 - 14 gal/ac	> 14 gal/ac
Total Off-Rate	0.00%	1.22%	94.01%	3.62%	1.01%
Turning Off-Rate	0.00%	0.83%	1.68%	0.81%	0.35%
Overlap Off-Rate	0.00%	0.00%	0.04%	0.21%	1.46%
Controller Response Off-Rate	0.00%	0.39%	0.00%	2.61%	-0.80%

Figure C.18: PACT generated quantified error report for Nebraska Nebraska Field 1864.

Appendix D: PACT Program Code

PACT GUI

```
function varargout = PesticideApplicationCoverageTool(varargin)
% PESTICIDEAPPLICATIONCOVERAGETOOL MATLAB code for
PesticideApplicationCoverageTool.fig
%      PESTICIDEAPPLICATIONCOVERAGETOOL, by itself, creates a new
PESTICIDEAPPLICATIONCOVERAGETOOL or raises the existing
%      singleton*.
%
%      H = PESTICIDEAPPLICATIONCOVERAGETOOL returns the handle to a new
PESTICIDEAPPLICATIONCOVERAGETOOL or the handle to
%      the existing singleton*.
%
%
PESTICIDEAPPLICATIONCOVERAGETOOL('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in PESTICIDEAPPLICATIONCOVERAGETOOL.M with the
given input arguments.
%
%      PESTICIDEAPPLICATIONCOVERAGETOOL('Property','Value',...) creates a new
PESTICIDEAPPLICATIONCOVERAGETOOL or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before PesticideApplicationCoverageTool_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to
PesticideApplicationCoverageTool_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
PesticideApplicationCoverageTool

% Last Modified by GUIDE v2.5 11-Sep-2017 14:00:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@PesticideApplicationCoverageTool_OpeningFcn, ...
                   'gui_OutputFcn',
@PesticideApplicationCoverageTool_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before PesticideApplicationCoverageTool is made visible.
function PesticideApplicationCoverageTool_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PesticideApplicationCoverageTool (see
VARARGIN)

% Choose default command line output for PesticideApplicationCoverageTool
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PesticideApplicationCoverageTool wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = PesticideApplicationCoverageTool_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes during object creation, after setting all properties.
function unlLogoAxes_CreateFcn(hObject, eventdata, handles)
% hObject    handle to unlLogoAxes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate unlLogoAxes

axes(hObject);

imshow('Nh_EXTENSION__4c.jpg')



% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%                      Field Information Section
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



function fieldNameEdit_Callback(hObject, eventdata, handles)
% hObject    handle to fieldNameEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fieldNameEdit as text
%        str2double(get(hObject,'String')) returns contents of fieldNameEdit as
a double
```

```
% --- Executes during object creation, after setting all properties.
function fieldNameEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fieldNameEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function dateSprayedEdit_Callback(hObject, eventdata, handles)
% hObject    handle to dateSprayedEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dateSprayedEdit as text
%        str2double(get(hObject,'String')) returns contents of dateSprayedEdit
as a double


% --- Executes during object creation, after setting all properties.
function dateSprayedEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dateSprayedEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function targetAppRateEdit_Callback(hObject, eventdata, handles)
% hObject    handle to targetAppRateEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of targetAppRateEdit as text
%        str2double(get(hObject,'String')) returns contents of
targetAppRateEdit as a double


% --- Executes during object creation, after setting all properties.
function targetAppRateEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to targetAppRateEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end


function fieldBndryFileEdit_Callback(hObject, eventdata, handles)
% hObject    handle to fieldBndryFileEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fieldBndryFileEdit as text
%        str2double(get(hObject,'String')) returns contents of
fieldBndryFileEdit as a double


% --- Executes during object creation, after setting all properties.
function fieldBndryFileEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fieldBndryFileEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%                    Sprayer Analytics Data Section
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



% --- Executes during object creation, after setting all properties.
function appRateDataTypeButtonGroup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to appRateDataTypeButtonGroup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



% --- Executes during object creation, after setting all properties.
function fileTypeButtonGroup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fileTypeButtonGroup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



function sprayerDataFileNameEdit_Callback(hObject, eventdata, handles)
% hObject    handle to sprayerDataFileNameEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sprayerDataFileNameEdit as
text
%        str2double(get(hObject,'String')) returns contents of
sprayerDataFileNameEdit as a double


% --- Executes during object creation, after setting all properties.
```

```
function sprayerDataFileNameEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sprayerDataFileNameEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%                      Sprayer Set-Up Section
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



% --- Executes on selection change in nozTypePopUpMenu.
function nozTypePopUpMenu_Callback(hObject, eventdata, handles)
% hObject    handle to nozTypePopUpMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns nozTypePopUpMenu
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
nozTypePopUpMenu


% --- Executes during object creation, after setting all properties.
function nozTypePopUpMenu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nozTypePopUpMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function nozSpacingEdit_Callback(hObject, eventdata, handles)
% hObject    handle to nozSpacingEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of nozSpacingEdit as text
%        str2double(get(hObject,'String')) returns contents of nozSpacingEdit
as a double


% --- Executes during object creation, after setting all properties.
function nozSpacingEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nozSpacingEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function numSectionsEdit_Callback(hObject, eventdata, handles)
% hObject    handle to numSectionsEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numSectionsEdit as text
%        str2double(get(hObject,'String')) returns contents of numSectionsEdit
as a double


% --- Executes during object creation, after setting all properties.
function numSectionsEdit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numSectionsEdit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in setSectionsPushButton.
function setSectionsPushButton_Callback(hObject, eventdata, handles)
% hObject    handle to setSectionsPushButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

numSec = str2double(get(handles.numSectionsEdit, 'String'));

tableSectionsAndNozzles = cell(numSec,2);
evenOdd = mod(numSec,2);

switch evenOdd
    case 0
        for i = 1:floor(numSec/2)
            tableSectionsAndNozzles{i,1} =
['SectionL',num2str(floor(numSec/2)+1-i)];
            tableSectionsAndNozzles{numSec+1-i,1} = ['SectionR',...
                num2str(floor(numSec/2)+1-i)];
        end
    case 1
        for i = 1:floor(numSec/2)
            tableSectionsAndNozzles{i,1} =
['SectionL',num2str(floor(numSec/2)+1-i)];
            tableSectionsAndNozzles{numSec+1-i,1} = ['SectionR',...
                num2str(floor(numSec/2)+1-i)];
        end
        tableSectionsAndNozzles{ceil(numSec/2),1} = ['SectionC'];
end

set(handles.secSetUpTable, 'data', tableSectionsAndNozzles);
```

```
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%                      PACT Mode Section
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


% --- Executes during object creation, after setting all properties.
function analyzeModeButtonGroup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to analyzeModeButtonGroup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%                      PACT Analyze Button
% XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



% --- Executes on button press in analyzePushButton.
function analyzePushButton_Callback(hObject, eventdata, handles)
% hObject    handle to analyzePushButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% collect all user input variables
FieldName = get(handles.fieldNameEdit, 'String');
DateSprayed = get(handles.dateSprayedEdit, 'String');
TargetApp = str2double(get(handles.targetAppRateEdit, 'String'));
FieldBoundaryFile = get(handles.fieldBndryFileEdit, 'String');
SprayerDataFile = get(handles.sprayerDataFileNameEdit, 'String');
NozzleSpacing = str2double(get(handles.nozSpacingEdit, 'String'));
NumberSections = str2double(get(handles.numSectionsEdit, 'String'));
SprayerSectionsSetUp = get(handles.secSetUpTable, 'data');
NozzleTypesMatrix = get(handles.nozTypePopUpMenu, 'string');
NozzleTypesValue = get(handles.nozTypePopUpMenu, 'value');

% create sprayer and field data structures for ease of passing inputs along
FieldData = struct;
SprayerData = struct;

FieldData.fieldName = FieldName;
FieldData.dateSprayed = DateSprayed;
FieldData.targetApp = TargetApp;
FieldData.fieldBndry = FieldBoundaryFile;

SprayerData.nozSpacing = NozzleSpacing;
SprayerData.numSec = NumberSections;
SprayerData.nozTip = NozzleTypesMatrix{NozzleTypesValue};



switch get(get(handles.appRateDataTypeButtonGroup,'SelectedObject'),'Tag')
    case 'sprayPresButton'
        ApplicationRateDataType = 'Pressure';
    case 'flowRateButton'
        ApplicationRateDataType = 'Flow Rate';
end

switch get(get(handles.fileTypeButtonGroup,'SelectedObject'),'Tag')
    case 'farmobileButton'
```

```
        SprayerDataFileType = 'Farmobile';
    case 'kvaserButton'
        SprayerDataFileType = 'Kvaser';
end

switch get(get(handles.analyzeModeButtonGroup,'SelectedObject'),'Tag')
    case 'normalRadioButton'
        AnalysisMode = 'Normal';
    case 'sectionControlRadioButton'
        AnalysisMode = 'Section Control';
    case 'turnCompensationRadioButton'
        AnalysisMode = 'Turn Comp';
end


[NozzlesPerSection,SectionWidths,SectionSpacing] =
sprayerSetUp(NumberSections,...
    SprayerSectionsSetUp,NozzleSpacing);

[Lat,Lon,X,Y,SectionStatus,SprayPressure, Time] =
extractSprayerData(SprayerDataFile,...
    SprayerDataFileType,NumberSections);

[SectionStatus,SprayPressure] = setAnalysisMode(AnalysisMode,...
    SectionStatus,SprayPressure);

orthoImage(Lat,Lon,X,Y,SectionStatus);

[Theta] = pathAngle(X,Y);

[SectionLocX,SectionLocY] = sectionLocation(X,Y,Theta,SectionSpacing,...
    SectionStatus);

[SectionPolygons,ApplicationRate,AreaSprayed] = sectionPolygon(SectionLocX,...
    SectionLocY,SectionWidths,Theta,SectionStatus,SprayPressure,...
    NozzlesPerSection,Time,NozzleTypesMatrix,NozzleTypesValue,...
    ApplicationRateDataType,TargetApp);

[OverlapArea,OverlapApplicationRate,OverlapGallons,PostOverlapArea,PostOverlapA
pp] = sectionIntersect(SectionPolygons,...
    ApplicationRate,AreaSprayed,TargetApp);

[TurningOffrateArea,TurningOffrateApplicationRate] =
turningOffrate(ApplicationRate,...
    AreaSprayed,NumberSections,NozzlesPerSection);

[TotalFieldArea,FieldBoundaryX,FieldBoundaryY] =
fieldBoundaryCalc(FieldBoundaryFile);

[SkippedSectionArea] = skipSectionPolygons(FieldBoundaryX,FieldBoundaryY,...
    SectionPolygons);

[OutsideSectionPolygonsArea,OutsideGallons] =
outsideSectionPolygons(FieldBoundaryX,...
    FieldBoundaryY,SectionPolygons,ApplicationRate);

%[q] = sprayerPath(Point,Theta);
%[GroundSpeed] = sprayerSpeed(X,Y,Time);


FieldMetrics = struct;
FieldMetrics.app = ApplicationRate;
FieldMetrics.areaField = TotalFieldArea;
```

```
FieldMetrics.areaSprayed = AreaSprayed;
FieldMetrics.olApp = OverlapApplicationRate;
FieldMetrics.olArea = OverlapArea;
FieldMetrics.orApp = TurningOffrateApplicationRate;
FieldMetrics.orArea = TurningOffrateArea;
FieldMetrics.skippedArea = SkippedSectionArea;
FieldMetrics.outsideArea = OutsideSectionPolygonsArea;
FieldMetrics.outGal = OutsideGallons;
FieldMetrics.olGal = OverlapGallons;

excelExport(FieldData,SprayerData,FieldMetrics)
```

Sprayer Set-up

```
function [nozPerSec,secWidths,secSpac] = sprayerSetUp(numSec,secSetup,nozSpac)
% This function takes user provided user inputs for their sprayer to come
% up with the necessary sprayer setup variables


% extract number of nozzles per section into separate matrix
nozPerSec = cellfun(@str2num,secSetup(:,2))';

% calculate width of each boom section (in m)
% NozPerSec X NozSpacing X 2.54cmPerInch / 100cmPerMeter
secWidths = nozPerSec.*(nozSpac).*(2.54)./(100);

% preallocate section spacing matrix and deteremine if number sections is odd
% or even
secSpac = zeros(1,numSec);
oddOReven = mod(numSec,2);

% based on number of sections calculate distance from center of boom to
% center of boom section
switch oddOReven
    case 0 % even
        for i = 1:numSec/2
            secSpac(0+i) = 0;
            secSpac(numSec+1-i) = 0;

            for j = 1:numSec/2+1-i
                if j == numSec/2+1-i
                    secSpac(0+i) = -secWidths(numSec/2+1-j)/2 + secSpac(0+i);
                    secSpac(numSec+1-i) = secWidths(numSec/2+j)/2 +
secSpac(numSec+1-i);
                else
                    secSpac(0+i) = -secWidths(numSec/2+1-j) + secSpac(0+i);
                    secSpac(numSec+1-i) = secWidths(numSec/2+j)
+secSpac(numSec+1-i);
                end
            end

        end
    case 1 % odd
        for i = 1:floor(numSec/2)
            secSpac(0+i) = -secWidths(ceil(numSec/2))/2;
            secSpac(numSec+1-i) = secWidths(ceil(numSec/2))/2;

            for j = 1:ceil(numSec/2)-i
                if j == ceil(numSec/2)-i
                    secSpac(0+i) = -secWidths(ceil(numSec/2)-j)/2 +
secSpac(0+i);
                    secSpac(numSec+1-i) = secWidths(ceil(numSec/2)+j)/2 +
secSpac(numSec+1-i);
                else
                    secSpac(0+i) = -secWidths(ceil(numSec/2)-j) + secSpac(0+i);
                    secSpac(numSec+1-i) = secWidths(ceil(numSec/2)+j) +
secSpac(numSec+1-i);
                end
            end

        end
end
```

Extract Data

```
function [Lat,Lon,X,Y,SectionStatus,SprayPres,...
    Time] = extractSprayerData(sprayerDataFile,sprayerDataFileType,numSec)
% this function takes the sprayer data file type and the sprayer data file
% and extracts the necessary sprayer analytics data to create high
% resolution as-applied maps and error reports


switch sprayerDataFileType
    case 'Farmobile'
        [Lat,Lon,X,Y,SectionStatus,SprayPres,Time] =
extractFarmobile(sprayerDataFile,numSec);
    case 'Kvaser'
        [Lat,Lon,X,Y,SectionStatus,SprayPres,Time] =
extractKvaser(sprayerDataFile);
end




function [Lat,Lon,X,Y,SectionStatus,SprayPres,Time] =
extractFarmobile(fileName,numSec)
% extracts attributes from Farm Mobile sprayer shapefiles
% ***needs .shp and .dbf file included in folder***


warning('off');

S = shaperead(fileName);

% extract Lat and Lon
Lon = extractfield(S,'X');
Lat = extractfield(S,'Y');
Lon = Lon';
Lat = Lat';

% extract timestamp
TS = extractfield(S,'TS');
TS = TS';
Time = timeAdjust(TS);
% Time = TS; % UK

% convert lat and lon to UTM x and y values
fieldzone = utmzone(mean(Lat,'omitnan'),mean(Lon,'omitnan'));
utmstruct = defaultm('utm');
utmstruct.zone = fieldzone;
utmstruct.geoid = wgs84Ellipsoid;
utmstruct = defaultm(utmstruct);
[X,Y] = mfwdtran(utmstruct,Lat,Lon);
X = smooth(X);
Y = smooth(Y);

%  X = extractfield(S,'utmX'); % UK
%  X = X';
%  Y = extractfield(S,'utmY');
%  Y = Y';

% extract spray pressure
SprayPres = extractfield(S,'SprayPress');
SprayPres = SprayPres';
```

```
% extract section status
SectionStatus = zeros(length(X),numSec);

if mod(numSec,2) == 1

    SectionStatus(:,ceil(numSec/2)) = extractfield(S,'SectionC');
    for i = 1:floor(numSec/2)
        SectionStatus(:,i) = extractfield(S,['SectionL',num2str(ceil(numSec/2)-
i)]);
        SectionStatus(:,numSec+1-i) =
extractfield(S,['SectionR',num2str(ceil(numSec/2)-i)]);
    end

else

    for i = 1:floor(numSec/2)
        SectionStatus(:,i) = extractfield(S,['SectionL',num2str((numSec/2)+1-
i)]);
        SectionStatus(:,numSec+1-i) =
extractfield(S,['SectionR',num2str((numSec/2)+1-i)]);
    end

end

idx = SectionStatus > 0;
SectionStatus(idx) = 1;
```

```
function [Lat,Lon,X,Y,SectionStatus,SprayPres,Time] = extractKvaser(filename)
% extracts attributes from Kvaser can data
% ***needs .csv file included in folder***

ds = tabularTextDatastore(filename);

% ds.SelectedVariableNames = {'Time','id','Data0','Data1','Data2','Data3',...
%     'Data4','Data5','Data6','Data7'};   % for hex data
ds.SelectedVariableNames = {'Var1','Var3','Var6','Var7','Var8','Var9',...
    'Var10','Var11','Var12','Var13'};
% ds.SelectedFormats = {'%f','%q','%q','%q','%q','%q','%q','%q','%q','%q'};
tt = tall(ds);

% extract message IDs for John Deere sprayers
idx = tt.Var3 == 218034972;
gpsMsgs = tt(idx,:);
idx = tt.Var3 == 284163845;
secMsgs = tt(idx,:);
idx = tt.Var3 == 419430369;
presMsgs = tt(idx,:);

[gpsData,secData,presData] = gather(gpsMsgs,secMsgs,presMsgs);


% parse gps data
gpsData = table2cell(gpsData);
gpsTime = cell2mat(gpsData(:,1));

latRaw = gpsData(:,3:6);
```

```
lonRaw = gpsData(:,7:10);

latRaw = fliplr(latRaw);
lonRaw = fliplr(lonRaw);

latBi = cellfun(@(x) de2bi(x,8,'left-msb'),latRaw,'UniformOutput',false);
lonBi = cellfun(@(x) de2bi(x,8,'left-msb'),lonRaw,'UniformOutput',false);

latBi = cell2mat(latBi);
lonBi = cell2mat(lonBi);

lat = bi2de(latBi,'left-msb');
lon = bi2de(lonBi,'left-msb');

lat = lat.*(10^-7);
lon = lon.*(10^-7);

lat = lat-210;
lon = lon-210;

idx = abs(lat) > 200 | abs(lon) > 200;
lat(idx,:) = [];
lon(idx,:) = [];
gpsTime(idx,:) = [];

zone = utmzone(mean(lat,'omitnan'),mean(lon,'omitnan'));
utmstruct = defaultm('utm');
utmstruct.zone = zone;
utmstruct.geoid = wgs84Ellipsoid;
utmstruct = defaultm(utmstruct);
[gpsX,gpsY] = mfwdtran(utmstruct,lat,lon);

gpsUTM = [gpsTime gpsX gpsY];
gpsLatLon = [gpsTime lat lon];


% parse section status data
secData = table2cell(secData);

secData(:,7:10) = [];
secData(:,3:4) = [];

secOnOff = secData(:,3:4);

secOnOffBi = cellfun(@(x) de2bi(x,8,'left-
msb'),secOnOff,'UniformOutput',false);

secOnOffBi = cell2mat(secOnOffBi);

secStat = [cell2mat(secData(:,1)) secOnOffBi(:,2) secOnOffBi(:,4)
secOnOffBi(:,6)...
    secOnOffBi(:,8) secOnOffBi(:,12) secOnOffBi(:,14) secOnOffBi(:,16)];

% idx = secStat > 0;
% secStat(idx) = 1;


% parse pressure data
presData = table2cell(presData);

presRaw = [presData(:,8) presData(:,9)];

presRaw = fliplr(presRaw);
```

```
presDataBi = cellfun(@(x) de2bi(x,8,'left-msb'),presRaw,'UniformOutput',false);

presDataBi = cell2mat(presDataBi);

presVal = bi2de(presDataBi,'left-msb');

presVal = presVal.*(0.0291666);

pres = [cell2mat(presData(:,1)) presVal(:,1)];


% resample data to similar timestamp
gpsStart = gpsTime(1,1);
gpsEnd = gpsTime(end,1);

idx = secStat(:,1) < gpsStart;
secStat(idx,:) = [];
idx = pres(:,1) < gpsStart;
pres(idx,:) = [];

idx = secStat(:,1) > gpsEnd;
secStat(idx,:) = [];
idx = pres(:,1) > gpsEnd;
pres(idx,:) = [];

presDif = diff(pres);
presDif = [0 0; presDif];
idx = presDif(:,1) > 0.05;
pres(idx,:) = [];

adjSecStat = zeros(1,8);
adjPres = zeros(1,2);

gpsTime = downsample(gpsTime,3);
gpsLatLon = downsample(gpsLatLon,3);
gpsUTM = downsample(gpsUTM,3);

for i = 1:length(gpsTime)

    t = gpsTime(i,1);

    [~,idxSec] = min(abs(secStat(:,1)-t));
    [~,idxPres] = min(abs(pres(:,1)-t));

    adjSecStat(i,:) = secStat(idxSec,:);
    adjPres(i,:) = pres(idxPres,:);

end

Lat = gpsLatLon(:,2);
Lon = gpsLatLon(:,3);
X = gpsUTM(:,2);
Y = gpsUTM(:,3);
SectionStatus = adjSecStat(:,2:8);
SprayPres = adjPres(:,2);
Time = gpsTime;
```

Analysis Mode

```
function [secStatus,sprayPres] =
setAnalysisMode(AnalysisMode,secStatus,sprayPres)
% Function is used to adjust variables to represent sprayer coverage
% analysis modes to represent various adjustments available to sprayers


switch AnalysisMode
    case 'Normal'
        % currently change nothing

    case 'Section Control'

        % find all rows which contain at least one section on
        onOff = any(secStatus,2);
        % turn all sections on
        secStatus(onOff,:) = 1;


    case 'Turn Comp'
        % currently place holder for turn compensation analysis

end
```

Ortho Imagery

```
function [] = orthoImage(lat,lon,x,y,secStatus)
% function is used to bring in orthoimagery for the field being analyzed
% original from john evans
% edited by aaron shearer

% reduce lat & lon to only area where spraying is happening
onOff = max(secStatus,[],2);
del = find(onOff == 0);
lat(del) = [];
lon(del) = [];
x(del) = [];
y(del) = [];


% create boundaries for ortho image selection
latMin = min(lat);
latMax = max(lat);
lonMin = min(lon);
lonMax = max(lon);

latDif = latMax-latMin;
lonDif = lonMax-lonMin;

latLim = [latMin-0.2*latDif, latMax+0.2*latDif];
lonLim = [lonMin-0.2*lonDif, lonMax+0.2*lonDif];

xMin = min(x);
xMax = max(x);
yMin = min(y);
yMax = max(y);

xDif = xMax-xMin;
yDif = yMax-yMin;

xLim = [xMin-0.2*xDif, xMax+0.2*xDif];
yLim = [yMin-0.2*yDif, yMax+0.2*yDif];


% attempt to reach web server to retrieve imagery
numberOfAttempts = 5;
attempt = 0;
info = [];
serverURL =
'https://services.nationalmap.gov/arcgis/services/USGSNAIPImagery/ImageServer/W
MSServer?';

while(isempty(info))
    try
        info = wmsinfo(serverURL);
        orthoLayer = info.Layer(1);
    catch e
        attempt = attempt + 1;
        if attempt > numberOfAttempts
            throw(e);
        else
            fprintf('Attempting to connect to server:\n"%s"\n', serverURL)
        end
    end
end
imageLength = 1024;
```

```
[A, R] = wmsread(orthoLayer, 'Latlim', latLim, 'Lonlim', lonLim, ...
    'ImageHeight', imageLength, 'ImageWidth', imageLength);


% Figure display setup (axes, scale ruler, titles, etc.)

xloc = min(xLim)+0.1*diff(xLim);
yloc = min(yLim)+0.05*diff(yLim);
m2ft = unitsratio('ft','m');

% defining image appeance for multiple images to be used throughout script
figure(1)
axesm('MapProjection','utm', 'Zone', utmzone(latLim, lonLim), ...
    'MapLatlimit', latLim, 'MapLonlimit', lonLim, ...
    'Geoid', wgs84Ellipsoid,'Frame','off','AngleUnits','degrees',...
    'parallellabel','on')
geoshow(A,R,'FaceAlpha',0.5)
axis off
scaleruler on
setm(handlem('scaleruler1'),...
    'units','m',...
    'XLoc',xloc,...
    'YLoc',yloc,...
    'MajorTick',0:25:100,...
    'MajorTickLength',5,...
    'MinorTick',0,...
    'TickDir','up',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
scaleruler('units','ft')
setm(handlem('scaleruler2'),...
    'units','ft',...
    'XLoc',xloc,...
    'YLoc',yloc-5,...
    'MajorTick',0:100:300,...
    'MajorTickLength',5*m2ft,...
    'MinorTick',0,...
    'TickDir','down',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
title(['\fontsize{22}\bf{Sprayer Coverage Map}' char(10) '\fontsize{18}\rm{Off-
Rate and Off-Target Application}'])
hold on

figure(2)
axesm('MapProjection','utm', 'Zone', utmzone(latLim, lonLim), ...
    'MapLatlimit', latLim, 'MapLonlimit', lonLim, ...
    'Geoid', wgs84Ellipsoid,'Frame','off','AngleUnits','degrees',...
    'parallellabel','on')
geoshow(A,R,'FaceAlpha',0.5)
axis off
scaleruler on
setm(handlem('scaleruler1'),...
    'units','m',...
    'XLoc',xloc,...
    'YLoc',yloc,...
    'MajorTick',0:25:100,...
```

```
    'MajorTickLength',5,...
    'MinorTick',0,...
    'TickDir','up',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
scaleruler('units','ft')
setm(handlem('scaleruler2'),...
    'units','ft',...
    'XLoc',xloc,...
    'YLoc',yloc-5,...
    'MajorTick',0:100:300,...
    'MajorTickLength',5*m2ft,...
    'MinorTick',0,...
    'TickDir','down',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
title(['\fontsize{22}\bf{Sprayer Coverage Map}' char(10) '\fontsize{18}\rm{Off-
Rate Application}'])
hold on

figure(3)
axesm('MapProjection','utm', 'Zone', utmzone(latLim, lonLim), ...
    'MapLatlimit', latLim, 'MapLonlimit', lonLim, ...
    'Geoid', wgs84Ellipsoid,'Frame','off','AngleUnits','degrees',...
    'parallellabel','on')
geoshow(A,R,'FaceAlpha',0.5)
axis off
scaleruler on
setm(handlem('scaleruler1'),...
    'units','m',...
    'XLoc',xloc,...
    'YLoc',yloc,...
    'MajorTick',0:25:100,...
    'MajorTickLength',5,...
    'MinorTick',0,...
    'TickDir','up',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
scaleruler('units','ft')
setm(handlem('scaleruler2'),...
    'units','ft',...
    'XLoc',xloc,...
    'YLoc',yloc-5,...
    'MajorTick',0:100:300,...
    'MajorTickLength',5*m2ft,...
    'MinorTick',0,...
    'TickDir','down',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
title(['\fontsize{22}\bf{Sprayer Coverage Map}' char(10) '\fontsize{18}\rm{Off-
Target Application}'])
```

```
hold on

figure(4)
axesm('MapProjection','utm', 'Zone', utmzone(latLim, lonLim), ...
    'MapLatlimit', latLim, 'MapLonlimit', lonLim, ...
    'Geoid', wgs84Ellipsoid,'Frame','off','AngleUnits','degrees',...
    'parallellabel','on')
geoshow(A,R,'FaceAlpha',0.5)
axis off
scaleruler on
setm(handlem('scaleruler1'),...
    'units','m',...
    'XLoc',xloc,...
    'YLoc',yloc,...
    'MajorTick',0:25:100,...
    'MajorTickLength',5,...
    'MinorTick',0,...
    'TickDir','up',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
scaleruler('units','ft')
setm(handlem('scaleruler2'),...
    'units','ft',...
    'XLoc',xloc,...
    'YLoc',yloc-5,...
    'MajorTick',0:100:300,...
    'MajorTickLength',5*m2ft,...
    'MinorTick',0,...
    'TickDir','down',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
title(['\fontsize{22}\bf{Sprayer Coverage Map}' char(10)
'\fontsize{18}\rm{Skipped Application}'])
hold on

figure(5)
axesm('MapProjection','utm', 'Zone', utmzone(latLim, lonLim), ...
    'MapLatlimit', latLim, 'MapLonlimit', lonLim, ...
    'Geoid', wgs84Ellipsoid,'Frame','off','AngleUnits','degrees',...
    'parallellabel','on')
geoshow(A,R,'FaceAlpha',0.5)
axis off
scaleruler on
setm(handlem('scaleruler1'),...
    'units','m',...
    'XLoc',xloc,...
    'YLoc',yloc,...
    'MajorTick',0:25:100,...
    'MajorTickLength',5,...
    'MinorTick',0,...
    'TickDir','up',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
scaleruler('units','ft')
```

```
setm(handlem('scaleruler2'),...
    'units','ft',...
    'XLoc',xloc,...
    'YLoc',yloc-5,...
    'MajorTick',0:100:300,...
    'MajorTickLength',5*m2ft,...
    'MinorTick',0,...
    'TickDir','down',...
    'RulerStyle','ruler',...
    'FontSize',8,...
    'Color','k',...
    'LineWidth',0.5,...
    'FontWeight','normal')
title(['\fontsize{22}\bf{Sprayer Coverage Map}' char(10) '\fontsize{18}\rm{Out
of Boundary Application}'])
hold on
```

Sprayer Angle

```
function [theta] = pathAngle(X,Y)
% Function is used to determine the angle from horizontal of the line
% from previous point to the current point


%initialize quadrant matrix
theta = zeros(length(X),1);

for i = 2:length(X)
    % calculate angle using four quadrant inverse tangent function
    theta(i) = atan2(Y(i)-Y(i-1),X(i)-X(i-1));
end
```

Section Location

```
function [secX,secY] = sectionLocation(x,y,theta,...
    secSpac,secStatus)
% Function determines location of each boom section midpoint throughout
% the field and applies on/off section status in a manner useful for
% plotting spray coverage.


% initialize boom section locations
secX = zeros(length(x),length(secSpac));
secY = zeros(length(x),length(secSpac));

for i = 1:length(secSpac)

    % calculate location of individual boom sections at each point through
    % field
    secX(:,i) = x+secSpac(i).*cos(pi/2-theta);
    secY(:,i) = y-secSpac(i).*sin(pi/2-theta);

end

% % all sections locations on or off
% allSecX = secX;
% allSecY = secY;

% set section locations to NaN when section status is off
secX(secStatus<1) = NaN;
secY(secStatus<1) = NaN;
```

Coverage Polygon Projections

```
function [polygon,app,area] = sectionPolygon(secX,secY,...
    secWidths,angle,secStatus,secPres,nozPerSec,time,...
    nozzleTypesMatrix,nozzleTypesValue,appRateDataType,TargetApp)
% Function is used to create and plot polygons along the path of each
% boom section to represent the area sprayed by the sprayer. Polygons
% strecth from boom sections previous location to the current.


% rename input variables
width = secWidths; theta = angle; t = time;
str = nozzleTypesMatrix; val = nozzleTypesValue;

% preallocate space for polygon memory
polygon = cell(size(secX));
coverage = cell(size(secX));
app = zeros(size(secX));
area = zeros(size(secX));


for i = 1:length(secX)-1
    for j = 1:length(width)

        x = zeros(4,1); y = zeros(4,1);

        if secStatus(i,j) > 0

            % project the x cordinates
            x(1) = secX(i,j)+width(j)./2.*cos(theta(i)-pi/2);
            x(2) = secX(i,j)-width(j)./2.*cos(theta(i)-pi/2);
            x(3) = secX(i+1,j)+width(j)./2.*cos(theta(i+1)-pi/2);
            x(4) = secX(i+1,j)-width(j)./2.*cos(theta(i+1)-pi/2);

            % project the y cordinates
            y(1) = secY(i,j)+width(j)./2.*sin(theta(i)-pi/2);
            y(2) = secY(i,j)-width(j)./2.*sin(theta(i)-pi/2);
            y(3) = secY(i+1,j)+width(j)./2.*sin(theta(i+1)-pi/2);
            y(4) = secY(i+1,j)-width(j)./2.*sin(theta(i+1)-pi/2);

            % temporarily store polygon points
            points = [x,y];

            % identify number of sides to polygon
            polyUnique = unique(points,'rows');
            [rp,~] = size(polyUnique);

            TF = isnan(points);
            checkNaN = sum(sum(TF,1),2);

            if rp > 3 && checkNaN == 0 %?!?!?!

                % order points to form polygon
                k = convhull(x,y);

                % separate out x & y coordinates
                x = x(k);
                y = y(k);

                % order points clockwise
                [x,y] = poly2cw(x,y);
```

```
                % store polygon
                polygon{i,j}(:,1) = x;
                polygon{i,j}(:,2) = y;

            else

                % if polygon is not 3 sided or more set to NaN
                x = NaN;
                y = NaN;
                polygon{i,j}(:,1) = x;
                polygon{i,j}(:,2) = y;

            end

        else

            % if section status is off set to NaN to avoid mapping
            x = NaN;
            y = NaN;
            polygon{i,j}(:,1) = x;
            polygon{i,j}(:,2) = y;

        end

        % calculate area of each polygon
        area(i,j) = polyarea(polygon{i,j}(:,1),polygon{i,j}(:,2));

        % calculate flowrate
        switch appRateDataType
            case 'Pressure'
                % using gui nozzle type input value get matching empirical
                % equation
                [flwrtEqn] = flowrateEquation(str,val);
                fr = flwrtEqn(secPres(i));
            case 'Flow Rate'
                % NEED TO ADD FLOWRATE VARIABLE TO DATA EXTRACTION
                % FUNCTIONS
                fr = FlowRate(i);
        end

        % calculate application rate
        dt = t(i+1)-t(i);
        app(i,j) = fr*dt*(1/area(i,j))*(4046.86/60)*nozPerSec(j);

        % create coverage used to color polygons
        coverage{i,j} = ones(size(polygon{i,j}(:,1)))*app(i,j);

    end
end

% remove last polygon thats unused
polygon(end,:) = [];
area(end,:) = [];
app(end,:) = [];

% identify size of polygon for for loops
[m,n] = size(polygon);

% % alternative color map scheme
% mapLuck = [0.5 0 0.9; 0 0 1; 0 1 0; 1 0.6 0; 1 0 0];
% mapSMS = [1 0 0; 1 0.5490 0; 1 1 0; 0.6784 1 0.1843; 0 1 0];

% plot polygons using patch, color based on coverage variable
```

```
figure(1) % off rate & off target
for i = 1:m
    for j = 1:n

        p1 = patch(polygon{i,j}(:,1),polygon{i,j}(:,2),coverage{i,j}(:,1),...
            'FaceColor','interp','EdgeColor','interp');
        p1.CData;
        p1.CDataMapping = 'scaled';
        caxis([0 2*TargetApp]);
        colormap(flipud(hot));
        %colormap(mapLuck);
        p1.FaceAlpha = 0.5;
        p1.EdgeAlpha = 0.5;
        hold on

    end
end

c = colorbar('Ticks',[0 .4*TargetApp 0.8*TargetApp 1.2*TargetApp...
    1.6*TargetApp 2*TargetApp]);
c.Label.String = 'Application Rate, gal/ac';
c.FontSize = 18;
axis equal

% plot polygons using patch, color based on coverage variable
figure(2) % off rate
for i = 1:m
    for j = 1:n

        p2 = patch(polygon{i,j}(:,1),polygon{i,j}(:,2),coverage{i,j}(:,1),...
            'FaceColor','interp','EdgeColor','interp');
        p2.CData;
        p2.CDataMapping = 'scaled';
        caxis([0 2*TargetApp]);
        colormap(flipud(hot));
        %colormap(mapLuck);
        p2.FaceAlpha = 0.5;
        p2.EdgeAlpha = 0.5;
        hold on

    end
end

c = colorbar('Ticks',[0 .4*TargetApp 0.8*TargetApp 1.2*TargetApp...
    1.6*TargetApp 2*TargetApp]);
% c = colorbar('Ticks',[0 0.5*TargetApp 0.9*TargetApp 1.1*TargetApp...
%     1.5*TargetApp 2*TargetApp]);
c.Label.String = 'Application Rate, gal/ac';
c.FontSize = 18;
axis equal
```

Coverage Polygon Intersection

```
function [olArea,olApp,olGal,newArea,newApp] =
sectionIntersect(polyOrg,appOrg,areaOrg,targetApp)
% Function is used to find intersecting areas of section polygons in the
% field and sum the app rates to account for the overlap of the sprayer

% rename input variables
poly = polyOrg(:); app = appOrg(:); area = areaOrg(:);

% create duplicates of original variables
initPoly = poly; initApp = app; initArea = area;


% delete last polygon, process uses previous loacation to current
[r1,~] = find(isnan(area));
poly(r1) = [];
app(r1) = [];
%area(r1) = [];
initPoly(r1) = [];
initApp(r1) = [];
initArea(r1) = [];

% identify and remove any non four sided polygons (should be little to none)
s = zeros(length(poly),1);
for i = 1:length(poly)
    [r2,~] = size(poly{i});
    s(i) = r2;
end

s = find(s ~= 5);
poly(s) = [];
app(s) = [];
%area(s) = [];
initPoly(s) = [];
initApp(s) = [];
initArea(s) = [];

% separate out x- and y-components from each polygon into individual cells
x = cell(length(poly),1);
y = cell(length(poly),1);
for i = 1:length(poly)
    x{i} = poly{i}(:,1);
    y{i} = poly{i}(:,2);
end

% place all polygons into one vector, an x-component vector and a
% y-component vector, separating polygons by NaN
polyX = ones(1);
polyY = ones(1);
for i = 1:length(poly)
    polyX(end+1:end+6,1) = [poly{i}(:,1);NaN];
    polyY(end+1:end+6,1) = [poly{i}(:,2);NaN];
end
polyX(1) = [];
polyY(1) = [];


% preallocate space to store metrics for any section polygons displaing an
% intersection
[r2,~] = size(initPoly);
olX = cell(r2,1);
```

```
olY = cell(r2,1);
olIndex = cell(r2,1);
olApp = zeros(r2,1);
olArea = zeros(r2,1);
olCover = cell(r2,1);

% overlap checking process
q = 1;

for i = 1:r2

    % section polygon being examined
    refX = initPoly{i}(:,1);
    refY = initPoly{i}(:,2);
    refApp = initApp(i);
    refArea = initArea(i);

    % remove polygon being referenced so it is not counted twice or checked
    % against itself
    poly(1) = [];
    x(1) = [];
    y(1) = [];
    app(1) = [];
    %area(1) = [];
    polyX(1:6) = [];
    polyY(1:6) = [];

    % find all polygons with at least one intersection
    [xi,~,ii] = polyxpoly(polyX,polyY,refX,refY);

    ii = ceil(ii./6);
    ii = unique(ii);

    if ~isempty(xi)

        for j = 1:numel(ii)

            % find intersection of the referenced polygon and create
            % polygon of the overlapping areas
            [olx,oly] = polybool('intersection',refX,refY,...
                x{ii(j)},y{ii(j)});

            if ~isempty(olx)
                % store data for new overlapping polygon
                olX{q,1} = olx;
                olY{q,1} = oly;
                olIndex{q,1} = [i,i+ii(j)];
                olApp(q,1) = refApp+app(ii(j));
                olArea(q,1) = polyarea(olx,oly);
                initArea(i) = initArea(i)-olArea(q,1);
                initArea(i+ii(j)) = initArea(i+ii(j))-olArea(q,1);

                olCover{q,1} = ones(size(olx))*olApp(q,1);

                q = q+1;
            end

        end

    end

end
```

```
% find and delete any empty cells in overlap matrix
empt = find(cellfun(@isempty,olX));

olX(empt) = [];
olY(empt) = [];
olIndex(empt) = [];
olApp(empt) = [];
olArea(empt) = [];
olCover(empt) = [];

% find and delete any excessively small area cells in overlap matrix
small = find(olArea < 0.001);

olX(small) = [];
olY(small) = [];
olIndex(small) = []; %#ok<NASGU>
olApp(small) = [];
olArea(small) = [];
olCover(small) = [];

% overlap removed area
newArea = [initArea;olArea];
newApp = [initApp;olApp];

% calculate excess gallons of chemical mix used due to overlap
difApp = olApp-targetApp;
olGal = difApp.*(olArea./4046.86);
olGal = sum(olGal);

% % alternative color map scheme
% mapLuck = [0.5 0 0.9; 0 0 1; 0 1 0; 1 0.6 0; 1 0 0];
% mapSMS = [1 0 0; 1 0.5490 0; 1 1 0; 0.6784 1 0.1843; 0 1 0];

figure(1) % off rate & off target
hold on
for k = 1:numel(olX)
    p1 =
patch(olX{k},olY{k},olCover{k},'FaceColor','interp','EdgeColor','interp'); %
section polygons
    p1.CData;
    p1.CDataMapping = 'scaled';
    caxis([0 2*targetApp]);
    colormap(flipud(hot));
    %colormap(mapLuck);
    p1.FaceAlpha = 0.5;
    p1.EdgeAlpha = 0.5;
    hold on
end
axis equal
hold on


figure(3) % off target only
hold on
for k = 1:numel(olX)
    p2 =
patch(olX{k},olY{k},olCover{k},'FaceColor','interp','EdgeColor','interp'); %
section polygons
    p2.CData;
    p2.CDataMapping = 'scaled';
    caxis([0 2*targetApp]);
    colormap(flipud(hot));
```

```
    %colormap(mapLuck);
    p2.FaceAlpha = 0.5;
    p2.EdgeAlpha = 0.5;
    hold on
end
c = colorbar('Ticks',[0 .4*targetApp 0.8*targetApp 1.2*targetApp...
    1.6*targetApp 2*targetApp]);
c.Label.String = 'Application Rate, gal/ac';
c.FontSize = 18;
axis equal
```

Turning Off-Rate

```
function [orArea,orApp] = turningOffrate(appRate,secArea,numSec,nozPerSec)
% Function is used to calculate offrate applications due to turning

app = appRate; area = secArea; noz = nozPerSec;

orArea = zeros(size(area));
orApp = zeros(size(app));

[m,~] = size(app);

% even number of sections
if mod(numSec,2) == 0
    center = [floor(numSec/2) ceil(numSec/2)];

    for i = 1:m
        numNoz = noz;

        rowArea = area(i,:)./numNoz;

        centerArea = (area(i,center(1))./numNoz(center(1))+...
            area(i,center(2))./numNoz(center(1)))/2;

        rowArea(rowArea>0.9*centerArea & rowArea<1.1*centerArea) = NaN;

        outside = find(~isnan(rowArea));

        orArea(i,outside) = area(i,outside);
        orApp(i,outside) = app(i,outside);
    end

% odd number of sections
elseif mod(numSec,2) == 1
    center = ceil(numSec/2);

    for i = 1:m
        numNoz = noz;

        rowArea = area(i,:)./numNoz;

        centerArea = area(i,center)./numNoz(center);

        rowArea(rowArea>0.9*centerArea & rowArea<1.1*centerArea) = NaN;

        outside = find(~isnan(rowArea));

        orArea(i,outside) = area(i,outside);
        orApp(i,outside) = app(i,outside);
    end

end

del = isnan(orArea);
orArea(del) = 0;

blank = orApp == 0;
orApp(blank) = NaN;
```

Field Boundary Assessment

```
function [area,x,y] = fieldBoundaryCalc(fldBndryFile)
% Function is used to calculate the total area of the field and plot the
% boundary


% label filename for field bndry shapefile
filename = fldBndryFile;

FB = shaperead(filename);

[r,~] = size(FB);

% if r == 1
%     lat = FB.Y';
%     lon = FB.X';
% else
%     for i = 1:r
%         lat{i,1} = FB(i).Y';
%         lon{i,1} = FB(i).X';
%     end
%     lat = cell2mat(lat);
%     lon = cell2mat(lon);
% end

% fieldzone = utmzone(mean(lat,'omitnan'),mean(lon,'omitnan'));
% utmstruct = defaultm('utm');
% utmstruct.zone = fieldzone;
% utmstruct.geoid = wgs84Ellipsoid;
% utmstruct = defaultm(utmstruct);
% [x,y] = mfwdtran(utmstruct,lat,lon);

 x = FB.X'; % <--- UK
 y = FB.Y';

[Ax,Ay] = polysplit(x,y);

for i =1:length(Ax)
    area(i) = polyarea(Ax{i},Ay{i});
end

area = sum(area);

figure(1)
hold on
p1 = plot(x,y,'k');
%p1.LineWidth = 2;
hold on

figure(2)
hold on
p1 = plot(x,y,'k');
%p1.LineWidth = 2;
hold on

figure(3)
hold on
p1 = plot(x,y,'k');
%p1.LineWidth = 2;
hold on
```

Skipped Sections

```
function [area] = skipSectionPolygons(fldBndryX,fldBndryY,SecPoly)
% Function is used to find any area skipped by sprayer inside of the field
% boundary


% rename input variables
fbX = fldBndryX; fbY = fldBndryY; secPoly = SecPoly;

% elminate NaNs
idx = cellfun(@isnan,secPoly,'UniformOutput',false);
idx = cellfun(@sum,idx,'UniformOutput',false);
idx = cellfun(@sum,idx,'UniformOutput',false);
idx = cell2mat(idx);
idx = idx > 0;
secPoly(idx) = [];

% reorder cells to form one matrix with NaN to separate polys
secPoly = cellfun(@(A) [A;nan(1,2)],secPoly,'UniformOutput',false);
secPoly = reshape(secPoly,[length(secPoly),1]);

% separate secPoly into its x- and y-components
x = secPoly{1}(:,1);
y = secPoly{1}(:,2);

for i = 1:length(secPoly)-1

    [x,y] = poly2cw(x,y);
    [xi,yi] = poly2cw(secPoly{i}(:,1),secPoly{i}(:,2));

    [xu,yu] = polybool('union',x,y,xi,yi);

    x = xu;
    y = yu;

end

% get polygons for skipped regions and calculate area
[skX,skY] = polybool('subtraction',fbX,fbY,x,y);
[adjX,adjY] = polysplit(skX,skY);
A = cellfun(@(x,y) polyarea(x,y),adjX,adjY,'UniformOutput',false);
A = cell2mat(A);
area = sum(A);
area = area/4046.86; % area in acres

% convert to faces and vertices for patch
[f,v] = poly2fv(skX,skY);

% plot skipped regions
figure(1)
hold on
patch('Faces',f,'Vertices',v,'FaceColor','g','EdgeColor','none')
hold on
plot(fbX,fbY,'k-')
hold on
axis equal

figure(4)
patch('Faces',f,'Vertices',v,'FaceColor','g','EdgeColor','none')
hold on
plot(fbX,fbY,'k-')
```

```
hold on
axis equal
```

Out of Boundary Application

```
function [secPolyOutArea,galOut] =
outsideSectionPolygons(fldBndryX,fldBndryY,SecPoly,SecApp)
% Function is used to find any area covered by sprayer outside of the field
% boundary

fbX = fldBndryX; fbY = fldBndryY; secPoly = SecPoly; secApp = SecApp;

% elminate NaNs
idx = cellfun(@isnan,secPoly,'UniformOutput',false);
idx = cellfun(@sum,idx,'UniformOutput',false);
idx = cellfun(@sum,idx,'UniformOutput',false);
idx = cell2mat(idx);
idx = idx > 0;
secPoly(idx) = [];
secApp(idx) = [];

% find section points inside field bndry
ptsIn = cellfun(@(poly) inpolygon(poly(:,1),poly(:,2),fbX,fbY),... % add in
ptsOn??
    secPoly,'UniformOutput',false);
ptsOut = cellfun(@(poly,idx) poly(~idx,:),secPoly,ptsIn,...
    'UniformOutput',false);

% find intersection points of polygon and field bndry
[ptsIntrscX,ptsIntrscY] = cellfun(@(poly) polyxpoly(poly(:,1),poly(:,2),...
    fbX,fbY),secPoly,'UniformOutput',false);
ptsIntrsc = cellfun(@(pts1,pts2) horzcat(pts1,pts2),ptsIntrscX,ptsIntrscY,...
    'UniformOutput',false);

% vertically concatenate ptsOut and ptsIntrsc
secPolyOut = cellfun(@(pts1,pts2) vertcat(pts1,pts2),ptsOut,ptsIntrsc,...
    'UniformOutput',false);

% delete empty cells
idx = cellfun(@isempty,secPolyOut);
secPolyOut(idx) = [];
secApp(idx) = [];

% start to reorder polygons and eliminate duplicate pts
idx = cellfun(@(poly) unique(poly,'rows'),secPolyOut,'UniformOutput',false);
idx = cellfun(@length,idx,'UniformOutput',false);
idx = cell2mat(idx);
idx = idx < 3;
secPolyOut(idx) = [];
secApp(idx) = [];

[K,secPolyOutArea] = cellfun(@(poly) convhull(poly(:,1),poly(:,2)),...
    secPolyOut,'UniformOutput',false);
secPolyOut = cellfun(@(poly,k) poly(k,:),secPolyOut,K,...
    'UniformOutput',false);

% calculate gallons sprayed outside fb
secPolyOutArea = cell2mat(secPolyOutArea);
galOut = secApp.*(secPolyOutArea/4046.86);
secPolyOutArea = sum(secPolyOutArea);
galOut = sum(galOut);

% rearrange secPoly for ploting
secPolyOut = reshape(secPolyOut,[length(secPolyOut) 1]);
```

```
%secPolyOut = cellfun(@(x) [x;x(1,1) x(1,2)],secPolyOut,'UniformOutput',false);
% <-- UK fields
secPolyOut = cellfun(@(x) [x;nan(1,2)],secPolyOut,'UniformOutput',false);
secPolyOut = cell2mat(secPolyOut);


if ~isempty(secPolyOut)

    [f,v] = poly2fv(secPolyOut(:,1),secPolyOut(:,2));
    % [x,y] = polysplit(secPolyOut(:,1),secPolyOut(:,2)); % <-- UK

    % plot regions sprayed outside of fld bndry
    figure(1)
    hold on
    patch('Faces',f,'Vertices',v,'FaceColor','b','EdgeColor', 'none')
    %cellfun(@(x,y)
patch(x,y,'blue','EdgeColor','none'),x,y,'UniformOutput',false); % <-- UK
    hold on
    plot(fbX,fbY,'k-')
    hold on
    axis equal

    figure(5)
    patch('Faces',f,'Vertices',v,'FaceColor','b','EdgeColor', 'none')
    %cellfun(@(x,y)
patch(x,y,'blue','EdgeColor','none'),x,y,'UniformOutput',false); % <-- UK
    hold on
    plot(fbX,fbY,'k-')
    hold on
    axis equal

end
```

Excel Export

```
function [] = excelExport(FieldData,SprayerData,FieldMetrics)
% Function is used to generate the desired report output values and move
% write them to a pre-made excel template

% create filename for new file
fn = {'PACT Report ',FieldData.fieldName,FieldData.dateSprayed,'.xlsx'};
fn = strjoin(fn);

% copy PACT report template to new file for field
copyfile('PACT Report Template.xlsx', fn);

% write field info to excel sheet
xlswrite(fn,cellstr(FieldData.fieldName),1,'C6');
xlswrite(fn,cellstr(FieldData.dateSprayed),1,'C7');
xlswrite(fn,cellstr(date),1,'C8');
FieldMetrics.areaFieldac = FieldMetrics.areaField/4046.86;
xlswrite(fn,FieldMetrics.areaFieldac,1,'D13');
xlswrite(fn,FieldData.targetApp,1,'D14');
FieldMetrics.totalSprayedArea = (nansum(nansum(FieldMetrics.areaSprayed),2)+...
    nansum(FieldMetrics.olArea))/4046.86;
xlswrite(fn,FieldMetrics.totalSprayedArea,1,'D15');
xlswrite(fn,FieldMetrics.skippedArea,1,'D16');
FieldMetrics.outsideArea = FieldMetrics.outsideArea/4046.86;
xlswrite(fn,FieldMetrics.outsideArea,1,'D17');


% write sprayer set-up info to excel sheet
xlswrite(fn,SprayerData.numSec,1,'H9');
SprayerData.nozSpacing = num2str(SprayerData.nozSpacing);
SprayerData.nozSpacing = strjoin({SprayerData.nozSpacing,' in.'});
xlswrite(fn,cellstr(SprayerData.nozSpacing),1,'H10');
xlswrite(fn,cellstr(SprayerData.nozTip),1,'H11');



% calculate necessary field metrics
allOffrate = FieldMetrics.app < 0.9*FieldData.targetApp |...
    FieldMetrics.app > 1.1*FieldData.targetApp;
olOffrate = FieldMetrics.olApp < 0.9*FieldData.targetApp |...
    FieldMetrics.olApp > 1.1*FieldData.targetApp;
turnOffrate = FieldMetrics.orApp < 0.9*FieldData.targetApp |...
    FieldMetrics.orApp > 1.1*FieldData.targetApp;

allOffrateArea = FieldMetrics.areaSprayed(allOffrate);
olOffrateArea = FieldMetrics.olArea(olOffrate);
turnOffrateArea = FieldMetrics.orArea(turnOffrate);

allOffrateArea = sum(allOffrateArea)/4046.86;
olOffrateArea = sum(olOffrateArea)/4046.86;
turnOffrateArea = sum(turnOffrateArea)/4046.86;

totalOffrateArea = allOffrateArea+olOffrateArea;


% write error totals to excel sheet
xlswrite(fn,totalOffrateArea,1,'G23');
xlswrite(fn,turnOffrateArea,1,'G24');
xlswrite(fn,olOffrateArea,1,'G26');

xlswrite(fn,totalOffrateArea,1,'D18');
```

```
% calculate percent of field area for different offrate bands
% overlap
olOffrate1 = FieldMetrics.olApp < 0.6*FieldData.targetApp;
olOffrate2 = FieldMetrics.olApp >= 0.6*FieldData.targetApp &...
    FieldMetrics.olApp < 0.9*FieldData.targetApp;
olOffrate3 = FieldMetrics.olApp >= 0.9*FieldData.targetApp &...
    FieldMetrics.olApp <= 1.1*FieldData.targetApp;
olOffrate4 = FieldMetrics.olApp > 1.1*FieldData.targetApp &...
    FieldMetrics.olApp <= 1.4*FieldData.targetApp;
olOffrate5 = FieldMetrics.olApp > 1.4*FieldData.targetApp;

olOffrateAreaBands = [sum(FieldMetrics.olArea(olOffrate1))...
    sum(FieldMetrics.olArea(olOffrate2))
sum(FieldMetrics.olArea(olOffrate3))...
    sum(FieldMetrics.olArea(olOffrate4)) sum(FieldMetrics.olArea(olOffrate5))];
olOffratePercentBands = olOffrateAreaBands./FieldMetrics.areaField;


% all
allOffrate1 = FieldMetrics.app < 0.6*FieldData.targetApp;
allOffrate2 = FieldMetrics.app >= 0.6*FieldData.targetApp &...
    FieldMetrics.app < 0.9*FieldData.targetApp;
allOffrate3 = FieldMetrics.app >= 0.9*FieldData.targetApp &...
    FieldMetrics.app <= 1.1*FieldData.targetApp;
allOffrate4 = FieldMetrics.app > 1.1*FieldData.targetApp &...
    FieldMetrics.app <= 1.4*FieldData.targetApp;
allOffrate5 = FieldMetrics.app > 1.4*FieldData.targetApp;

allOffrateAreaBands = [sum(FieldMetrics.areaSprayed(allOffrate1))...
    sum(FieldMetrics.areaSprayed(allOffrate2))
sum(FieldMetrics.areaSprayed(allOffrate3))...
    sum(FieldMetrics.areaSprayed(allOffrate4))
sum(FieldMetrics.areaSprayed(allOffrate5))];
allOffrateAreaBands = allOffrateAreaBands+olOffrateAreaBands;
allOffratePercentBands = allOffrateAreaBands./FieldMetrics.areaField;


% turning offrate
orOffrate1 = FieldMetrics.orApp < 0.6*FieldData.targetApp;
orOffrate2 = FieldMetrics.orApp >= 0.6*FieldData.targetApp &...
    FieldMetrics.orApp < 0.9*FieldData.targetApp;
orOffrate3 = FieldMetrics.orApp >= 0.9*FieldData.targetApp &...
    FieldMetrics.orApp <= 1.1*FieldData.targetApp;
orOffrate4 = FieldMetrics.orApp > 1.1*FieldData.targetApp &...
    FieldMetrics.orApp <= 1.4*FieldData.targetApp;
orOffrate5 = FieldMetrics.orApp > 1.4*FieldData.targetApp;

orOffrateAreaBands = [sum(FieldMetrics.orArea(orOffrate1))...
    sum(FieldMetrics.orArea(orOffrate2))
sum(FieldMetrics.orArea(orOffrate3))...
    sum(FieldMetrics.orArea(orOffrate4)) sum(FieldMetrics.orArea(orOffrate5))];
orOffratePercentBands = orOffrateAreaBands./FieldMetrics.areaField;


% write the percent bands to the excel sheet
xlswrite(fn,allOffratePercentBands',1,'M9:M13');
xlswrite(fn,orOffratePercentBands',1,'N9:N13');
xlswrite(fn,olOffratePercentBands',1,'O9:O13');

% create band labels for report graph
band60 = 0.6*FieldData.targetApp;
```

```
band90 = 0.9*FieldData.targetApp;
band110 = 1.1*FieldData.targetApp;
band140 = 1.4*FieldData.targetApp;

band60 = num2str(band60);
band90 = num2str(band90);
band110 = num2str(band110);
band140 = num2str(band140);

band1 = {'< ',band60,' gal/ac'};
band2 = {band60,' - ',band90,' gal/ac'};
band3 = {band90,' - ',band110,' gal/ac'};
band4 = {band110,' - ',band140,' gal/ac'};
band5 = {'> ',band140,' gal/ac'};

band1 = strjoin(band1);
band2 = strjoin(band2);
band3 = strjoin(band3);
band4 = strjoin(band4);
band5 = strjoin(band5);

xlswrite(fn,cellstr(band1),1,'L9');
xlswrite(fn,cellstr(band2),1,'L10');
xlswrite(fn,cellstr(band3),1,'L11');
xlswrite(fn,cellstr(band4),1,'L12');
xlswrite(fn,cellstr(band5),1,'L13');
```

Appendix E: MATLAB Statistics Code

PACT vs. SMS

```
% Thesis Statistics
% Aaron Shearer
% 01/29/2018

clear; clc;

% Comparison of error bands from PACT and SMS programs for Fields 1-8

fa = [0 16.62 75.41 4.38 3.59...
0.02 15.71 76.88 3.94 3.47...
0.66 22.58 69.18 4.95 3.10...
0.07 17.01 72.99 7.14 2.96...
0 10.64 84.10 2.88 2.38...
1.82 11.58 75.68 7.37 5.40...
2.86 6.46 80.22 6.66 6.66...
2.88 14.70 65.55 14.17 5.93...
...
0.15 2.35 95.16 2.15 0.22...
0.15 5.12 89.51 4.38 0.76...
0.45 7.51 85.88 5.07 0.81...
0.53 4.52 86.51 6.99 1.48...
0.34 3.69 93.92 1.60 0.32...
0 3.08 90.76 4.98 1.30...
0 3.02 92.16 3.81 0.86...
0 3.42 85.46 9.01 1.55];

prgm = {'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS';...
'SMS';'SMS';'SMS';'SMS';'SMS'};

bnd = [1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
```

```
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5];

fld = [1 1 1 1 1 ...
2 2 2 2 2 ...
3 3 3 3 3 ...
4 4 4 4 4 ...
5 5 5 5 5 ...
6 6 6 6 6 ...
7 7 7 7 7 ...
8 8 8 8 8 ...
1 1 1 1 1 ...
2 2 2 2 2 ...
3 3 3 3 3 ...
4 4 4 4 4 ...
5 5 5 5 5 ...
6 6 6 6 6 ...
7 7 7 7 7 ...
8 8 8 8 8];

[p1,tbl1,stats1,terms1] = anovan(fa,{prgm bnd fld},'model','interaction',...
'varnames',{'prgm','bnd','fld'});
figure()
results1 = multcompare(stats1,'Dimension',[1 2]);


% comparing calculated field sizes from SMS and PACT

fs = [43.6005 44.87...
50.4323 54.4...
10.3173 11.05...
12.9314 14.16...
28.0429 29.02...
15.734 16.88...
12.8366 13.91...
2.9343 3.218];
prgm = {'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS'};
fld = [1 1 ...
2 2 ...
3 3 ...
4 4 ...
5 5 ...
6 6 ...
7 7 ...
8 8];
[p2,tbl2,stats2,terms2] = anovan(fs,{prgm fld},'varnames',{'prgm','fld'});
figure()
results2 = multcompare(stats2,'Dimension',[1 2]);


% comparing calculated average application rates from SMS and PACT

ar = [14.7047 14.87...
14.7702 14.98...
14.7114 14.91...
```

```
14.9715 15.31...
14.7765 14.86...
15.2943 15.10...
15.8915 14.95...
15.447 15.37];
prgm = {'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS', ...
'PACT','SMS'};
fld = [1 1 ...
2 2 ...
3 3 ...
4 4 ...
5 5 ...
6 6 ...
7 7 ...
8 8];
[p3,tbl3,stats3,terms3] = anovan(ar,{prgm fld},'varnames',{'prgm','fld'});
figure()
results3 = multcompare(stats3,'Dimension',[1 2]);
```

PACT vs. Luck et al.

```
% Thesis Statistics
% Aaron Shearer
% 01/29/2018

clear; clc;

% Comparison of error bands from PACT and Luck et al. systems for
% UK Fields 1,2,&4

fa = [14.41 36.07 29.69 18.78 1.06 ...
8.87 38.95 37.22 12.81 2.16 ...
15.67 36.42 25.13 18.85 3.94 ...
10.88 36.88 35.75 12.40 4.09 ...
7.02 29.62 34.45 25.51 3.40 ...
15.67 36.42 25.13 18.85 3.94];

prgm = {'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';'PACT';...

'UK';'UK';'UK';'UK';'UK';...
'UK';'UK';'UK';'UK';'UK';...
'UK';'UK';'UK';'UK';'UK'};

bnd = [1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5 ...
1 2 3 4 5];

fld = [1 1 1 1 1 ...
2 2 2 2 2 ...
4 4 4 4 4 ...
1 1 1 1 1 ...
2 2 2 2 2 ...
4 4 4 4 4];

[p1,tbl1,stats1,terms1] = anovan(fa,{prgm bnd fld},'model','interaction',...
'varnames',{'prgm','bnd','fld'});
figure()
results1 = multcompare(stats1,'Dimension',[1 2]);


% Comparison of off-rate error bands from PACT and Luck et al. systems for
% UK Fields 1,2,&4

fa = [5.32 5.58 1.72 0.52 ...
3.66 12.51 3.42 0.88 ...
7.11 6.53 2.38 0.76 ...
1.45 6.15 4.29 0.47 ...
1.05 12.89 7.32 1.07 ...
1.12 8.34 4.47 0.59];

prgm = {'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';...
'PACT';'PACT';'PACT';'PACT';...

'UK';'UK';'UK';'UK';...
'UK';'UK';'UK';'UK';...
```

```
'UK';'UK';'UK';'UK'};

bnd = [1 2 4 5 ...
1 2 4 5 ...
1 2 4 5 ...
1 2 4 5 ...
1 2 4 5 ...
1 2 4 5];

fld = [1 1 1 1 ...
2 2 2 2 ...
4 4 4 4 ...
1 1 1 1 ...
2 2 2 2 ...
4 4 4 4];

[p2,tbl2,stats2,terms2] = anovan(fa,{prgm bnd fld},'model','interaction',...
'varnames',{'prgm','bnd','fld'});
figure()
results2 = multcompare(stats2,'Dimension',[1 2]);


% comparing calculated field sizes from PACT and Luck et al.

fs = [5.5712 5.904 ...
46.3118 49.2872 ...
15.8318 16.5526];
prgm = {'PACT','UK', ...
'PACT','UK', ...
'PACT','UK'};
fld = [1 1 ...
2 2 ...
4 4];
[p3,tbl3,stats3,terms3] = anovan(fs,{prgm fld},'varnames',{'prgm','fld'});
figure()
results3 = multcompare(stats3,'Dimension',[1 2]);


% comparing calculated average application rates from PACT and Luck et al.

ar = [8.2249 10.3763 ...
8.6859 11.1332 ...
8.1451 10.7951];
prgm = {'PACT','UK', ...
'PACT','UK', ...
'PACT','UK'};
fld = [1 1 ...
2 2 ...
4 4];
[p4,tbl4,stats4,terms4] = anovan(ar,{prgm fld},'varnames',{'prgm','fld'});
figure()
results4 = multcompare(stats4,'Dimension',[1 2]);
```

Tech. Adoption

```
% SecControl Analysis


% COmparison of overlap area between 7 section automatic boom control and
% single section manual control

% percent field area displaying overlap
ol = [1.79 3.77 1.87 4.62 5.09 ...
13.02 17.91 13.89 12.94 21.41];

% control type
cont = {'ASC';'ASC';'ASC';'ASC';'ASC'; ...
    'MAN';'MAN';'MAN';'MAN';'MAN';};

[p,tbl,stats] = anova1(ol,cont);
figure()
results = multcompare(stats,'Dimension',[1 2]);
```