2012

Design and assessment of an automated grain auger position control system

Benjamin Charles Potter
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd

Part of the Agriculture Commons, and the Bioresource and Agricultural Engineering Commons

Design and assessment of an automated grain auger position control system

by

Benjamin Charles Potter

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Agricultural Engineering (Advanced Machinery Engineering)

Program of Study Committee:
Matthew J. Darr, Major Professor
Steven J. Hoff
Brian L. Steward

Iowa State University

Ames, Iowa

2012

Table of Contents

v

List of Figures

ix

List of Tables

Acknowledgements

Abstract

Automation of new applications has become a major focus of the agricultural machinery industry in recent years. Automation technologies are able to improve productivity, efficiency, and the operator experience. Original equipment manufacturers use automation to market new equipment and offset the added cost of Tier IV engines incurred by the customer. To advance automation, new applications are being explored.

During harvest, combines must frequently be unloaded to make room for more crop intake. Large farming operations will normally perform the unloading process on-the-go, meaning that crop harvesting will continue while grain is unloaded into a grain cart towed by a tractor in parallel with the combine. Manual unloading on-the-go is stressful, because the combine operator must divide his/her attention between unloading and the normal harvesting tasks (steering, speed control, crop intake, and monitoring machine performance).

Automating the unloading process has the potential to reduce operator stress and improve in-field productivity. John Deere has recognized this potential and partnered with Iowa State University to develop the SmartUnload system. One requirement of an automated unloading system is a means of actuating to control the location of the auger boot within the grain cart.

The primary goal of this work was to develop an auger-swing control system as the primary means of actuation. The response characteristics of the original auger system as well as a proportional solution were investigated and the corresponding in-field performance evaluated. Results indicate that on-off and proportionally controlled auger swings are effective means of actuation for the automated unload system. Proportional control significantly improves control precision, allowing for smoother fill profiles and better grain cart utilization.

Future work should be completed to integrate production-intent proportional hardware. Implementing a more advanced control, such as gain-scheduled proportional control, with a more discrete fill strategy would further improve auger-swing fill performance. For the auger system studied, swinging out had a slightly different response than swinging in; more consistent performance could be possible by implementing a separate control function for each swing direction.

Chapter 1 Introduction

1.1 SmartUnload Project Background

The SmartUnload project began in the spring of 2010 with the goal of automating the combine unloading process by 2014. The catalysts for this project were the automation of synonymous systems in biomass and forage trailer-filling applications and John Deere's drive to differentiate its machinery through automation.

Iowa State University had already worked with John Deere on SmartSpout, a machine-vision based solution for automatically actuating a spout (mounted to the rear of a combine) to fill trailers and wagons with cellulosic biomass; this made a partnership for this new application a logical choice. The National Robotics Engineering Center (NREC) at Carnegie Melon University has also joined the project and is responsible for developing the machine-vision portion of the system. NREC has a long history of success in machine-vision applications and brings much expertise to the project. Multiple business units within John Deere have contributed to the project. The project is funded by Harvester Works and Harvester Works contributes machine expertise, project guidance & oversight, and testing support to the development process. Intelligent Solutions Group (ISG), the precision agriculture division within John Deere, leads project management and hardware integration activities. Embedded Systems Shared Services (E3S), the lead embedded systems group within John Deere, is responsible for taking the C-code generated at Iowa State University and creating a build compatible with the production ECU. Together, the team includes about 18 managers and developers; several other secondary contributors are not included in that number.

1.2 Machinery

John Deere has provided great resources in the form of funding and machinery for this project to add to Iowa State University's already substantial facilities, faculty, and capabilities.

Three combines have been used at Iowa State University for various purposes during the SmartUnload project. Test machine 109 will be the primary combine mentioned within

this document, because it includes all of the hydraulics and instrumentation required for proportional control and controls testing. Machine #109 is a John Deere 9870 STS with a high-capacity unloading auger (over 3 bu/s); the combine is labeled as a 9860 STS and has been a prototype machine for several years (Figure 1). A biomass spout was mounted at the rear of the machine as a result of a previous project, and the chopper and other components still remain.

Figure 1: Combine 109 was used for the testing in this thesis.

Throughout the project multiple tractors were used for pulling grain carts, wagons, and an auger. The three most used tractors were a John Deere 8R, Case IH 8930 Magnum, and Mahindra utility tractor. The 8R and 8930 were used to pull the grain carts and wagons. The Mahindra utility tractor was used to drive a small auger. The auger was used to refill a combine for unload testing during the non-harvest seasons (Figure 3).

Figure 2: Unloading on-the-go at the test track

Figure 3: Refilling the combine at the test track

Several fields were utilized by the SmartUnload project during harvest testing. Nearly 100 acres of soybeans and 353 acres of corn were utilized for project and thesis-related testing. The field used for the dynamic testing discussed in this document was South Woodruff, a 56 acre corn field located only a couple miles southwest of Ames, Iowa.

1.3 Testing

System testing has been a highly-iterative process during SmartUnload development. Testing usually starts with software simulations which involve feeding the model a set of inputs and observing that the outputs are appropriate. The next step in testing, sometimes skipped over, is static testing; during static testing, the system is repetitively tested while the combine is not moving. The benefit of static testing is that it requires less area, results in faster repetition, and provides better control of the test procedure. The last step in system testing is always dynamic testing; dynamic testing is done with the combine and tractor at ground speeds intended to be representative of typical harvest field speeds. By testing at field speeds, the results more closely indicate the performance that will be observed by the project sponsor or farmers.

Chapter 2 Literature Review

2.1 Automation in Agriculture

Agricultural processes that were previously manual are being automated. The success of automated systems such as GPS-guided automated steering and variable rate application has motivated original equipment manufacturers (OEMs) to find and develop more automated systems. Automated systems can be used as product differentiators. The new Tier IV engine implementations have increased equipment costs without providing value to the farmers; OEMs want to market the added costs under newly automated systems and encourage new purchases. Whether the automated systems increase profit or operator comfort or both, many farmers are adopting these technologies. From 1999 to 2010, adoption of precision agriculture technologies increased from 23.6% of farmers surveyed to 38.7% of farmers surveyed; the adoption rate is even higher for large farms, which makeup the largest portion of the new-equipment market (Batte, Forster, Surjandari, Hudson, & Rodriguez-Solis, 1999) (Diekmann & Batte, 2010). Equipment manufacturers can capitalize on the precision agriculture adoption trend by including desirable, automated systems in new product releases.

Another driver behind automation in agriculture is the potential for increased productivity. The amount of arable land available has been holding steady while the world's population and demand for grain is increasing. According to the World Bank data presented by Google, the world population has been increasing almost linearly since 1960 at a rate of 76.8 million people per year (from about 3 billion to 6.84 billion); and over about the same period of time, the quantity of land available for farming has remained about the same (Google, 2011). Oklahoma State presented data from the 2010 USDA NASS Prospective Plantings Report that showed very little change in acres for major crops over the past ten years [Figure 4] (Oklahoma State University, 2011). To meet the increases in demand with the same amount of land, productivity must also increase.
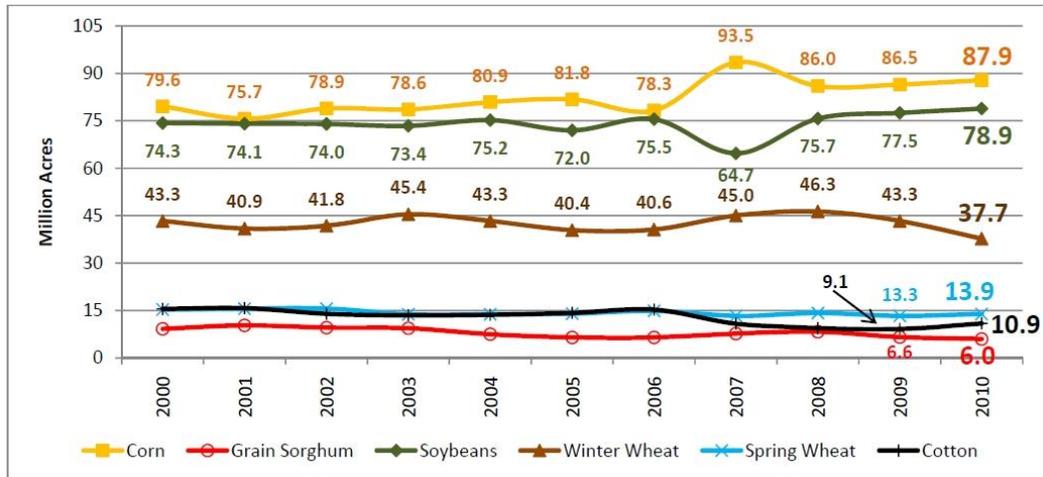
Figure 4: Prospective Plantings History

In the agricultural machinery industry, many engineers (and marketing teams) envision complete autonomy of in-field operations. Automation of the unloading application brings the industry one step closer to realizing this goal. The Demeter project demonstrated the capability of complete autonomy; in 1997, a New Holland windrower (Demeter) was able to continuously harvest 40ha without an operator (Pype & Posselius, 2011). Since this project, marketing teams everywhere have been predicting fully autonomous field operations (Deere & Company, 2011).

Figure 5: John Deere prototype futuristic tractor

2.2 The Combine Unloading Application

One application with potential for improvement through automation is the combine unloading process. A combine travels through the field harvesting crop. The grain is stored in a hopper on the combine. Periodically the combine hopper must be unloaded. Typically for large operations, the unload operation is performed on-the-go into a specialized wagon called a grain cart. The unloading operation requires a high level of attention and interaction

from the combine operator which can cause reduced productivity and increased operator stress/fatigue.

2.3 The Trailer-filling Synonym

A synonym to automating combine unloading is the automation of forage harvester spout actuation to fill a trailer. Claas, New Holland, and John Deere are all working on different approaches. Much of the resulting knowledge can be useful in the combine unloading application.

2.3.1 Claas AutoFill

The Claas trailer-filling system is called AutoFill (Claas, 2011). AutoFill utilizes a three-sensor camera (two gray-scale lenses and one color lens) mounted on the lower side of the forage harvester spout to detect and track a wagon or trailer as well as map the material level within the container (Moller, 2010). The application projects lines over the edges of the trailer opening to indicate its location. Many details pertaining to the control functions are not available publically.
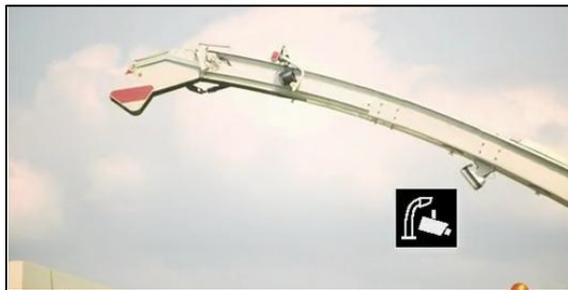
Figure 6: Claas AutoFill (Claas, 2011)

Claas partnered with a German university (Technical University of Braunschweig [Brunswick]) during the development, and some of the fill research by Happich et al. and Weltzien et al. was published. The primary goal of that research was to model fill profiles (resultant of forage delivery into a trailer or opening over time) in order to develop throughput and machine-vision based control functions.

Weltzien et al. tested two strategies for a throughput-based trailer loading assist system. The first strategy was to utilize a light bar mounted on the forage harvester to direct the tractor operator to the optimum relative position while the spout was kept at a fixed orientation. The problem with only moving the tractor was that experienced operators would often ignore the light bar. The second strategy was to keep the tractor at a constant location

relative to the harvester and swing the spout to fill the trailer. This strategy performed better but did not feel natural to the operators; also, moving the spout exposed the system to greater wind disturbance. The accuracy achieved was 0.5m. Control dead-bands were required due to the hysteresis in the sensor linkages and mechanical components. Fill completion was based on throughput and elapsed time, though operators could push a button to terminate the system when a full trailer was observed. For this system, the spout flap was responsible for lateral adjustments to material placement, and a triangular or sinusoidal pattern was used to evenly fill the trailer during the entire process (Weltzien, Graefe, Bonig, & (Claas) Diekhans, 2004).

The work completed by Weltzien et al. opened up the opportunity for fill profile modeling by Happich et al. Fill modeling was investigated as a function of several factors: crop type, the location of adjacent bulk heaps and trailer edges, the material impact angle, and bulk heap peak ("apex") shifting (Happich & Lang, 2009). The performance of these models was not included in the publications.

Additional work is being completed by the Institute of Control Engineering at the Technical University of Brunswick on the machine-vision based fill control. Material has not been published related to that research, probably due to the close relationship to the proprietary Claas system.

2.3.2 New Holland IntelliFill

The New Holland trailer-filling system is called IntelliFill. This system also uses a sensor or sensors mounted to the forage harvester spout, but even less has been published about the details of its operation. Marketing has published videos to advertise the systems capabilities without revealing technical specifications (New Holland, 2011).

Figure 7: New Holland IntelliFill in action

Figure 8: New Holland IntelliFill Time-of-Flight Camera

2.3.3 John Deere Single-Pass Forage Harvester (SPFH)

John Deere has also begun developing its own system of automated trailer filling. The project is closely tied to the combine SmartUnload project. Iowa State University is working with NREC and John Deere Zweibrucken to get the system ready for production in 2015. Currently, it is expected that a stereo camera will be attached to the spout, allowing for closed-loop control on material placement and fill level in the trailer.

2.4 Competitive Combine Systems

Two other systems have been advertised as having automated combine unloading assist capabilities: the Case-New Holland (CNH) Vehicle-to-Vehicle (V2V) and the Kinze Autonomous Grain Cart. These two systems use GPS information and in-field wireless communication to control the grain cart's location relative to the combine. However, these two systems have high liability due to their reliance on uninterrupted RTK-grade GPS data and wireless communication. Neither system has sensors allowing for closed-loop control on the location of material placement or fill level in the cart (Fiat Industrial, 2011) (Kinze Manufacturing, 2011) (Jaybridge Robotics, 2011).

Figure 9: CNH V2V

Figure 10: Kinze Autonomous Grain Cart

2.5 The Unmet Need

SmartUnload addresses the gap in functionality between the forage harvester and combine applications currently being developed. SmartUnload takes advantage of the strategies employed in the forage application to close the loop on material placement and fill level while avoiding the need for wireless communication and GPS data. This makes SmartUnload the first truly automated unloading system for combines; Table 1 shows the comparison.

Table 1: Combine Unloading and Trailer Filling System Comparison

Company	Origin	Platform	Closed-Loop on Fill Level	GPS and Wireless Communication Required
Claas	Germany	Forage harvester	Yes	No
New Holland	Belgium	Forage harvester	Yes	No
Case-New Holland	Nebraska	Combine	No	Yes
Kinze	Iowa	Combine	No	Yes
John Deere	Illinois	Combine	Yes	Optional

2.6 Subject Matter Leveraged

A wide range of subject matter was leveraged for the development of the Control Implementation subsystem and, specifically, proportional swing control of the auger. The combination of disciplines was unique. The courses at Iowa State University provided necessary learning and resources.

SmartUnload utilized several feedback (closed-loop) control functions in making decisions and executing actuations. Feedback control is defined as control that operates on the difference (and/or integral of the difference, and/or derivative of the difference) of the desired state and the current state of a system. Specifically, on-off and proportional control types only operate on the difference between the desired and currents states without integrals

or derivatives (Ogata, 2004). A diagram (see Figure 11) and associated function (Equation 1) represent the relationship.
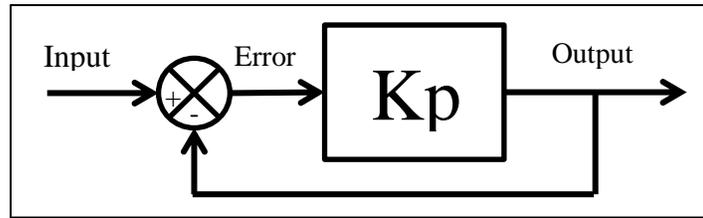
Figure 11: Closed-loop Proportional Control

$$Output = K_p \times (Input - Output_{-1})$$ <div style="text-align:right">**Equation 1**</div>

Hydraulic circuits and fittings is a broad subject area. For proportional control, a circuit design, valve selection, and fittings selection was required. O-ring Face Seal, O-ring Boss, and 37-degree Flare fittings are all defined in SAE standards J514 and J5143; pipe thread fittings were also used (Valley Hydraulic Service, Inc., 2011). Basic circuit pressure and flow analyses were completed based on two laws: the sum of flows at a node being equal to zero, and the sum of pressures through a closed loop is equal to zero (Steward, 2011).

Controller Area Networks (CAN Bus) has become the most common method used on agricultural equipment for multiple ECUs to communicate. CAN is unique in that it allows for multiple ECUs to communicate over a single network and has the high level of reliability required in automotive and agricultural applications. Standards have been created for communication protocol (CAN 2.0B), implementation protocol (ISO 11783), and message protocol (SAE J1939). A message contains two main portions or frames: the ID, and the Data. The ID will specify the type of data contained in the message as well as the source and destination addresses of the appropriate ECUs. A typical message will contain 64 bits of data, usually broken into eight 8-bit bytes. ISO 11783 defines the wiring protocol, including the maximum length of the CANbus, type of wiring required, acceptable interactions between multiple buses, baud rate (data rate), and network termination hardware. SAE J1939 provides a standard for common messages while leaving room for proprietary message creation. The common messages are important to allow for implements and tractors of different brands to function together. However, many companies want to hide the internal functionality of some systems by using special message definitions. (Darr, AE 410X/510X Electronic Systems Integration for Agricultural Machinery Production Systems, 2011)

State charts (aka state machines, state diagrams, logic charts) are commonly used to make control decisions. The Stateflow library within Simulink (graphical programming language created by MathWorks) was used to represent the state logic with software (MathWorks, 2011). In general, state charts can represent any type of exclusive or parallel logic sequence. A state chart is fed with a series of inputs which, depending on the current state, will directly correspond to a specific set of outputs (Darr, AE 410X/510X Electronic Systems Integration for Agricultural Machinery Production Systems, 2011). A detailed description of Stateflow is contained within the appendix.

Signal noise is a result of signal interferences and includes false frequency and magnitude components. Methods for reducing or eliminating noise include the use of twisted pair or shielded wire, software filtering, and hardware filtering. Low-pass, band-pass, and high-pass filters can be implemented using software or hardware. Hardware filters remove unwanted frequencies in the electrical signal before conversion to digital values and storage of data; in general, hardware filters will induce far less lag in a signal than a software filter. Software filters require a digital version of a transfer function. For many challenging applications of software filtering a Discrete Fourier Transform and Inverse Fourier Transform analysis is required (Hoff, 2011). In a simple application, a single-pole software filter can be manually tuned to restrict most high-frequency noise (Smith III, 1985). Two major constraints of a software filter are the rate at which the filter will sample and the acceptable amount of signal lag. If the filter rate changes, the filter coefficients will also need to change to maintain consistent performance. As filter aggressiveness increases, filter lag will also increase. Equation 2 is a single-pole filter, with X representing the sensor input, α representing the smoothing factor, and Y representing the filter output.

$$Y = \alpha \times X + (1 - \alpha) \times Y_{-1} \hspace{4cm} \textbf{Equation 2}$$

Chapter 3 Objectives

The use of automated machinery in agriculture has gained momentum in recent years. In many cases where automation has been used, there has been a significant increase in productivity and/or operator comfort (ex: GPS auto steering, planter seed-rate controllers). To achieve even greater benefit from automation, additional applications must be developed. One such possibility is automated combine unloading.

Currently the location of grain delivery is controlled by the combine operator. The combine operator uses in-cab controls to change the speed of the combine to place the grain at the desired location within the grain cart. This method reduces the throughput capacity of the combine during unloading; in addition, the percentage of time spent unloading has been increasing due to larger grain heads, higher horsepower combines, and higher yields (without proportional increases in unloading rate). An enhancement to this manual grain delivery system could include a control system which would place grain accurately within the cart, avoid unnecessary grain loss, and not reduce the throughput capacity of the combine.

The *long-term goal* of our research team is to enhance the overall efficiency of agricultural machine systems. Our *objective in this project* is to better understand and improve the grain unloading process through system analysis, automated control, and performance testing of the selected control method. Our *central hypothesis* is that a semi-automated or fully-automated approach to grain delivery to a cart from a harvester, incorporating auger-swing actuation, will maximize the performance potential. *Key deliverables* will include a system analysis, control design & implementation, and performance testing & analysis. Testing will require extensive planning in order for performance to be evaluated throughout the possible range of use cases. Our *rationale*, based on in-field experience with traditional unloading systems, is that if the combine operator can focus solely on field conditions and crop intake (not on the unloading processes), then improvements in field efficiency and throughput may be realized.

Specific Goal 1: Design a control system for automated auger actuation. Functionality must include both intelligent unloading and spill prevention practices; the primary challenge will be determining the best logic for each feature. Desired unloading

location will be dynamic based on a separate fill strategy. Performance will be based on the ability to accurately and efficiently place grain at the desired location and prevent spillage. System analysis and characterization will be completed to improve performance and understand system limitations.

Specific Goal 2: Design and analyze a proportional auger swing control system. By understanding the unloading system, enhanced control methods can be developed and optimal performance conditions can be evaluated for the existing system. By replacing/adding some physical hardware (additional valves and instrumentation), it will be possible to implement and test more sophisticated control methods.

Specific Goal 3: Quantify performance through field evaluation of automated combine unloading control methods. Data will be collected in order to evaluate several performance criteria. The overall performance of auger-swing control will determine the success of the initial system and the areas for improvement in future work.

This project is innovative in that, to date, all grains placement is done by manually changing the relative velocity of the combine. By incorporating automatic control of the auger-swing to change the relative boot location within the cart, the potential increases in combine productivity can be fully evaluated. This is of significant value to agricultural machinery suppliers and contemporary farming operations.

Chapter 4 Terms

Combine: in the context of this project, a large machine used to harvest fields of grain such as corn, soybeans, wheat, milo, and rice

Figure 12: Combine Component Definitions

SmartUnload (SU): the title selected for the John Deere automated combine unloading system

Smart Unloading Controller (SUC): the name of the controller containing the SmartUnload decision logic and output commands

Fill Strategy (FS): the subsystem within the SUC responsible for processing fill data and determining the best location within the cart for unloading

Control Implementation (CI): the subsystem within the SUC responsible for actuating the auger boot within the cart to the location desired by FS, starting and stopping grain flow, and preventing grain spillage

Stereo Cameras: are cameras that each contain two lenses, allowing for calculation of distances from the image data

MicroAutoBox (MAB): high-end rapid-prototyping ECU, manufactured and sold by dSpace, used for iterative software validation activities and data collection

Stereo Data: data produced from the camera images, including fill and tracking values used by the SmartUnload control functions to make command decisions

Targets (fiducials): two-dimensional geometric patterns fastened to the side of a grain cart and used by the image processing software to identify the grain cart location and size

Grain cart: special grain wagon used to transport grain through the field. These carts have scales and unloading augers to facilitate measurement of grain harvested and fill of transport wagons and trailers.

Figure 13: Cart-Edge Reference Definition

109: the number corresponding to a specific test combine provided to Iowa State University by John Deere for the SmartUnload project; the proportional control hardware was installed solely on this machine

8R: new model front-wheel-assist John Deere tractor provided to Iowa State University by John Deere for the SmartUnload project

Dynamic testing: testing completed while the combine's average velocity was greater than 2mph with the intention of simulating typical harvest unloading-on-the-go

Static testing: testing not done on-the-go (dynamically). Testing completed while the combine's velocity is within -1mph to 1mph, most commonly 0mph.

Chapter 5 System Definition

5.1 The SmartUnload System Architecture

The SmartUnload system architecture is relatively large and will be discussed in terms of three sub-architectures: hardware, control model, and CAN. The main hardware configuration utilized an MAB as the primary control ECU. The CAN message architecture consists of several messages and their definitions correspond to the inputs and outputs of the control model (see the Appendix for the message definitions).

5.1.1 Hardware Configuration

The hardware configuration is based on the combination of Smart Unloading Controller (SUC) ECU and combine electrical architecture. The electrical architecture describes the number and function of the ECUs on the combine. The combine had a Lynx electrical architecture.

Test combine number 109 was used for all of the auger-controllability studies, proportional testing, and field testing for Control Implementation. The 109 remained a Lynx-only machine during development and continued to use the developmental stereo software and hardware. The MAB was always used as the SUC and a Panasonic ToughBook transmitted the stereo data over serial to the MAB. The 109 swing cylinder was instrumented for pressure, the auger boot instrumented for grain flow, and the proportional control system installed (see Figure 14).
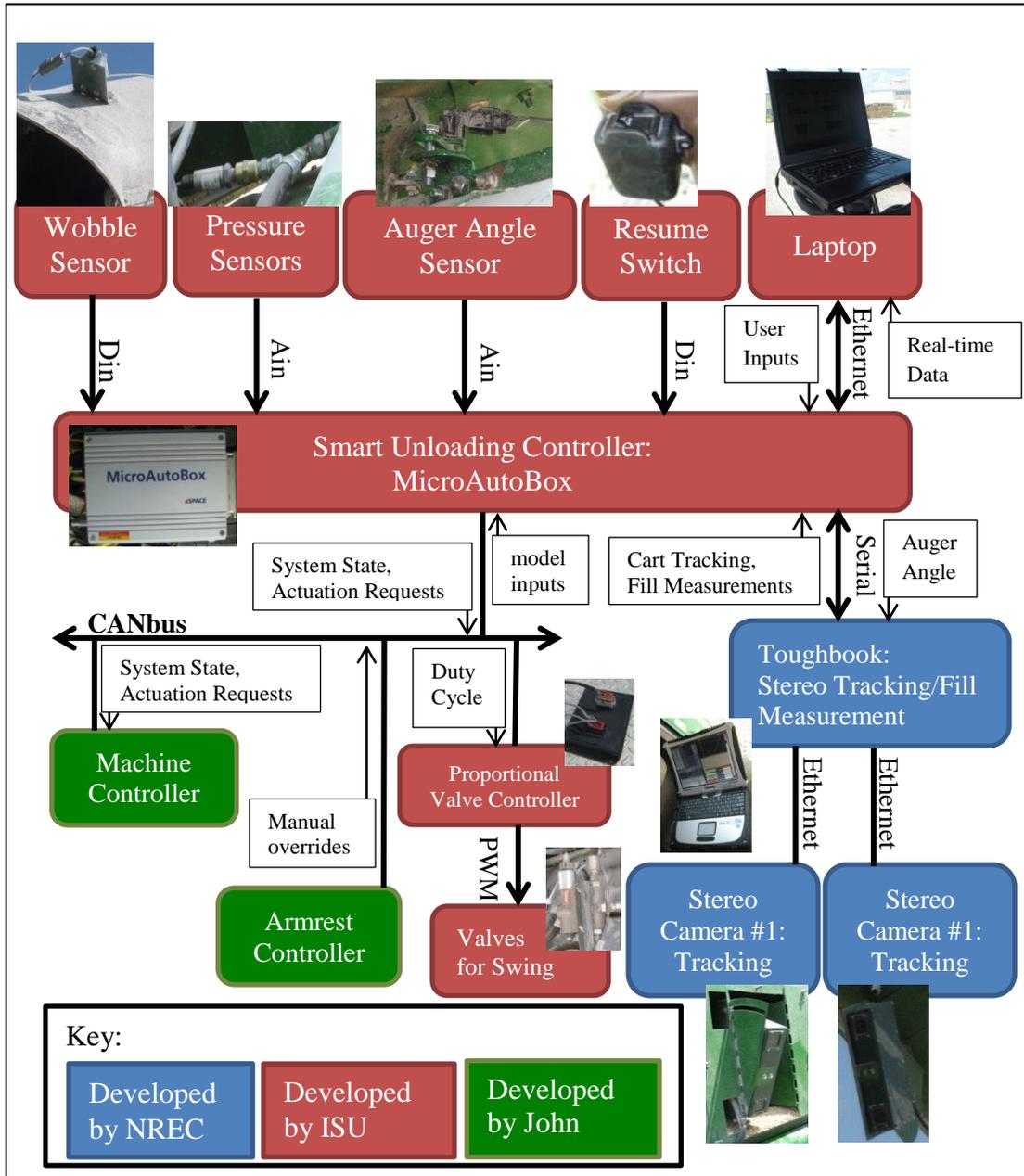
Figure 14: SmartUnload System Diagram for the 109 machine

5.2 Auger-Swing Hydraulics & Control

The auger-swing is actuated by a hydraulic cylinder. For implementation of proportional control it was necessary to study and understand the existing hydraulic system and then design and install a modified circuit.

5.2.1 Factory Hydraulics

A single manifold on the circuit supplied hydraulic power to multiple work functions on the combine, including the auger swing cylinder. A generalization of the auger-swing portion of the original hydraulic circuit is provided in Figure 15.
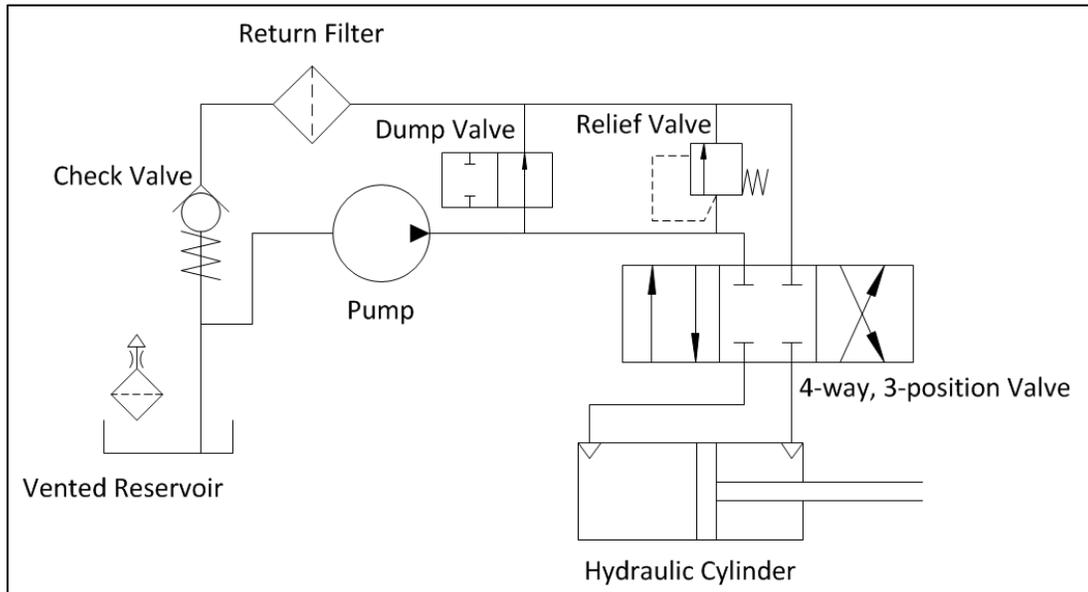
Figure 15: From-Factory Auger-swing Circuit

Engineers from John Deere Harvester Works were consulted on the details of the circuit. A relief valve set at 3100psi will allow flow to go directly to the return side; this is the primary circuit protection. A 1.181mm diameter orifice is threaded into each port of the cylinder to limit flow to about 1.7gpm (Figure 16). The cylinder is also designed to have cushions at full extend and full retract (though further testing indicates this is minimal on 109).

Figure 16: Orifice in the Cylinder Port

Some dimensions of the cylinder were measured. The ID of the cylinder body is 2in and the OD of the cylinder rod is 1.25in. From these dimensions, the effective area of the

cap-end and rod-end were calculated. Also, given stroke length, the change in volume on either side of the cylinder is available.

5.2.2 Proportional Hydraulics

An intermediate hydraulic circuit was designed for proportional control. The ability to physically switch back to on-off control quickly was a requirement of this system. Proportional cartridge valves with the correct specifications were not available to replace the on-off valves already in the manifold so two free-reverse-flow, flow control valves from Brand Hydraulics were used between the manifold and the hydraulic cylinder. The free-reverse-flow option was necessary to allow flow on the low-pressure side of the cylinder to return to the reservoir. Ball valves were used to bypass the flow control valves during on-off operation and to close off the bypass to ensure all the flow was limited during proportional control operation. The modified circuit is shown in Figure 17, and Figure 18 contains a picture of the final installation.
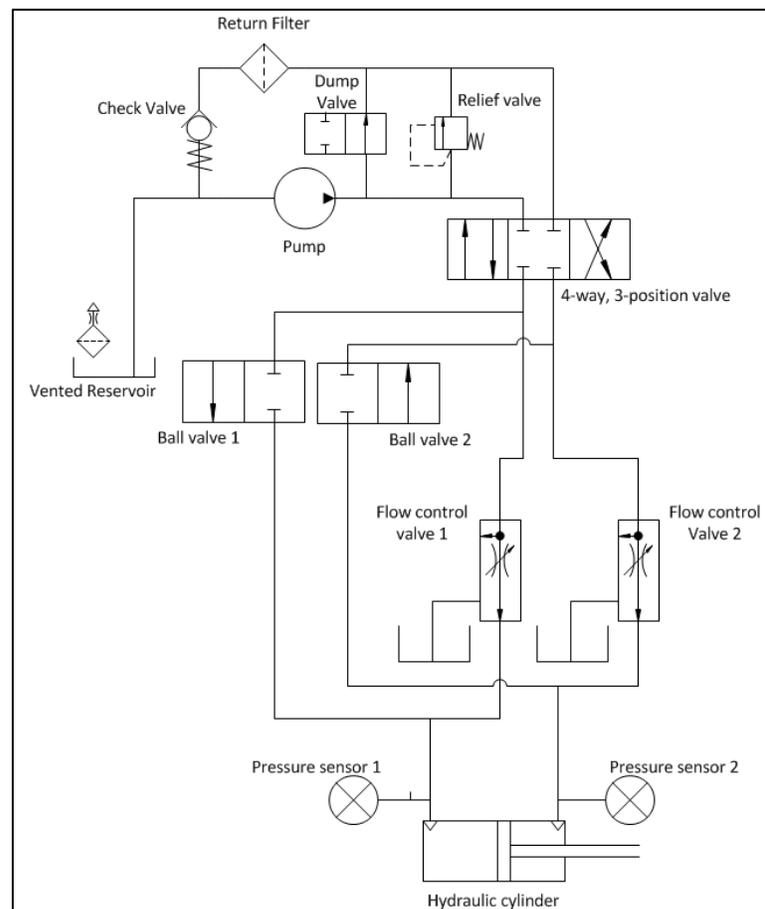
Figure 17: Hydraulic Circuit with Proportional Control
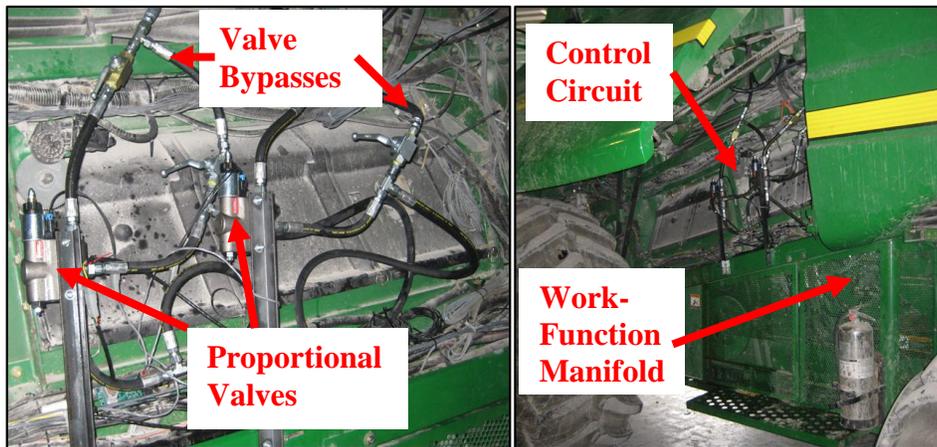
Figure 18: Final Proportional Control Circuit Installation

The Brand valves (Figure 19) are model number EFCC12-05-12 ('EFC' defines electronic flow control, 'C' indicates the free-reverse-flow option, '12' is the SAE port size, '05' is the maximum desired flow in gallons per minute, '12' is the power supply required in volts). The specified PWM frequency range is 90-115hz and the maximum rated pressure is 3000psi. The engineers at Brand stated that the safety factor on maximum pressure was large and that a circuit relief setting of about 3100psi would not be a problem. (Brand Hydraulics, 2011)
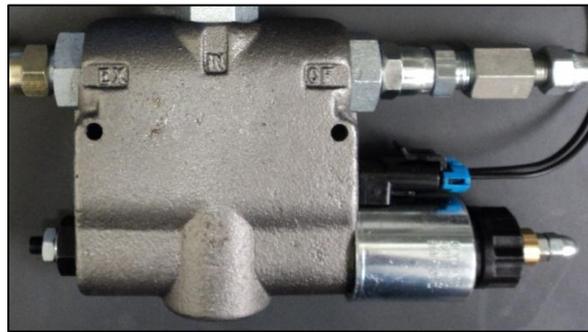
Figure 19: Brand Hydraulics Flow Control Valve

5.3 Flow Control ECU

A separate ECU was used to control the proportional flow valves. The microprocessor on the ECU is a MicroChip dsPIC30F5013. The ECU interfaces with CAN and has the high-current PWM outputs needed to drive the valves. A CAN message that designates the PWM duty cycle for each valve is sent from the MAB and received by the ECU. C-code was written to perform the required logic and flashed to the board.

The c-code for the main function is provided in the Appendix. First the CAN-configuration header file is called. Then variables and registers are initialized. A CAN filter is configured to ensure only the message corresponding to the valve duty-cycle command is used. On a time-based 100hz interrupt, the PWM is output. An interrupt will also trip when a message is received that meets the filter requirements; the bytes in the CAN message are then converted to the correct duty-cycle values for both chip output compares (PWM). Finally, a diagnostic CAN message is sent containing the same data bytes received from the command message; this is useful to verify that the command message is being received and unpacked correctly.

5.4 Acquiring Tracking & Fill Data

Two stereo cameras were used to acquire the needed tracking and fill data. Cart tracking data is needed in order to calculate the location of grain placement relative to the edges of the cart. Fill data is primarily used by Fill Strategy to determine how full the cart is and where more grain should be unloaded.

The tracking camera is mounted on the side of the combine, in a cutout from the grain tank. The fill camera is mounted to the auger to have a better field of view over the grain in the cart. Stereo cameras use two lenses at a fixed distance from each other to perceive depth and take measurements within the shared field of view. The tracking software looks for fiducials (targets) mounted on the side of the cart and provides XYZ coordinates of the front and rear inside corners of the grain cart.

The cameras and processing software on 109 were supplied by NREC. The cameras are circled in Figures 20 & 21. The targets consist of known geometric shapes (Figures 22 & 23) which enables the processing software to more easily find and track their locations.

Figure 20: 109 Tracking (grain tank) Camera

Figure 21: 109 Fill (auger) Camera

Figure 22: Kinze Grain Cart (840bu) with Fiducials

Figure 23: Brent Grain Cart (880bu) with Fiducials

The origin of the camera coordinate system is at the center of the tracking camera (Figure 24). The x-axis is parallel to the ground plain and the direction of travel. The y-axis is parallel to the ground plane and perpendicular to the direction of travel (positive in the direction of the cart). The z-axis is perpendicular to the ground and parallel to the side of the combine or cart (positive towards the sky).
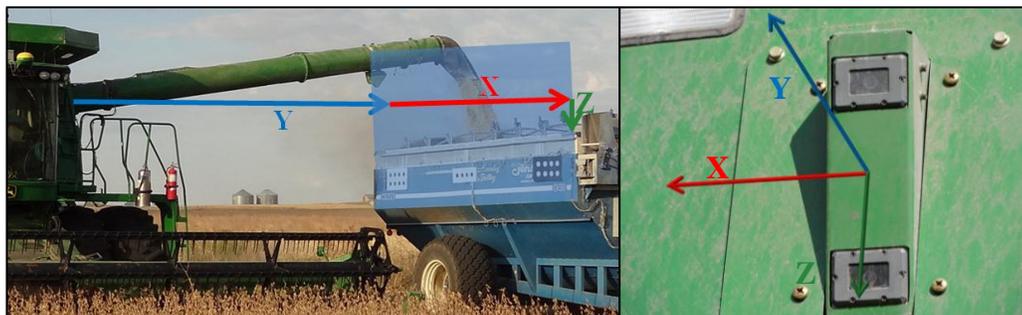
Figure 24: Camera Coordinate System

5.5 MicroAutoBox

The MAB is a high-end, rapid prototyping ECU. It has very high processing capacity and can perform fixed-point or floating-point math. The MAB can interface with a wide range of inputs and outputs including analog, digital, serial, and CAN.

The MAB has been used to test new software versions. Because it is relatively easy to reprogram the MAB through a laptop, multiple software versions can be tested daily and model changes can be made from the combine cab.

DSpace also has a software application called ControlDesk which can be used to observe and change software variables on the MAB in real-time. This has been heavily utilized for software debugging and controls tuning.

Data acquisition through the MAB is also user-friendly. The sampling rate can be manually specified (1-1000hz) to correspond to how the data will be used. Individual variables are selected for logging; file size is reduced because only the desired variables are saved. The acquired data logs have been used for quantifying system performance.

5.6 Auger Angle Calibration

Auger angle is defined as the angle in the plane of motion between the current auger position and the transport auger position. Auger angle must be known in order to calculate the location of the auger boot within the cart. The output voltage from the analog sensor is not linearly related to the auger angle; therefore, a calibration method is required to ensure that auger angle is correctly calculated from the voltage input.

5.6.1 Auger Angle Sensor

A John Deere production sensor (and corresponding mounting kit) was used to sense auger rotation (Figure 25). The sensor is a rotary potentiometer. The output voltage is related linearly to the rotation of the sensor shaft; however, the slotted linkage connecting the sensor to the auger is not linear.

Figure 25: Production Auger-Angle Sensor

5.6.2 Scaling the Sensor Voltage

First the voltage is linearly scaled to a 0-250 value. This allows for the value to have a good resolution within the unsigned-integer 8-bit data type (uint8). The 0-250 is fit between the minimum and maximum voltages. The minimum voltage is recorded when the auger is swung all the way in (transport position), and the maximum voltage is recorded with the auger swung all the way out. The uint8 value is packed into a structure and read into the control model.

5.6.3 Primary Calibration Procedure

Next, the 0-250 value is related to auger angle by completing the primary calibration procedure. The calibration is based on the law of cosines (Equation 3). Figure 26 is a depiction of the triangle of concern.

$$C^2 = A^2 + B^2 - 2 \times A \times B \cos \gamma \qquad \text{Equation 3}$$

*where 'γ' is the interior angle of the triangle opposite side 'C'

Figure 26: Auger Angle Calibration Triangle of Concern

A plumb-bob, normal tape measure, and a long tape measure are the needed equipment to perform a calibration. The combine is parked on flat ground with room to swing the auger. The plumb-bob is hung from the end of the auger where the black boot overlaps the green-painted steel; when the auger is all the way in, the tip of the plumb-bob just touches the ground. Directly below the plumb-bob, the loose end of the long tape is fixed to the ground (a stake or duct tape have each been used successfully). At fourteen auger angles (evenly spaced throughout the range of motion and including the minimum and maximum angles) the height of the plumb-bob off the ground, the distance from the fixed end of the long tape measure to the point on the ground directly below the plumb-bob, and the 0-250 auger position value are recorded (Figure 27).

Figure 27: Ground Measurement in Progress for Auger Calibration

A single measurement of the effective auger length (aka radius of calibration) is needed for calculating auger angle. The effective auger length is the distance from the center of the auger-pivot axis along the top of the auger to where the plumb-bob is attached (Figure 28); the measurement is equal to the lengths of side A and side B of the triangle. The calibration is highly sensitive to this measurement; therefore, taking a careful measurement with accuracy of $\pm$1in is important and normally requires two people.
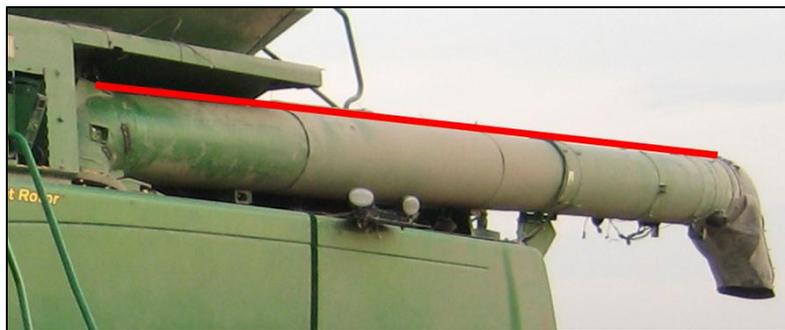
Figure 28: Effective Auger Length for Auger Calibration

The ground tape measurements are used to calculate the auger angle at each location (Eqs 4 & 5). The plumb-bob height and the distance from the plumb-bob to the stake point are first used to calculate the length of side C. Then the law-of-cosines is solved for the angle; in this application, the length of side A is equal to the length of side B.

$$C = \sqrt{plumbbob\_height^2 + dist\_to\_stakepoint^2}$$ **Equation 4**
     *Pythagorean Theorem

$$\gamma = \cos^{-1}\left(\frac{C^2 - A^2 - B^2}{-2 \times A \times B}\right)$$

Equation 5

Figure 29: Alternate Auger Angle Depiction

Auger angle is plotted against the 0-250 value and a cubic function is fit to the data. Numerous calibrations have shown that a cubic relationship provides a strong/reliable fit (Figure 30). The coefficients of the function are scaled up to fit the int32 data type in order to maintain high resolution within the model calculations.
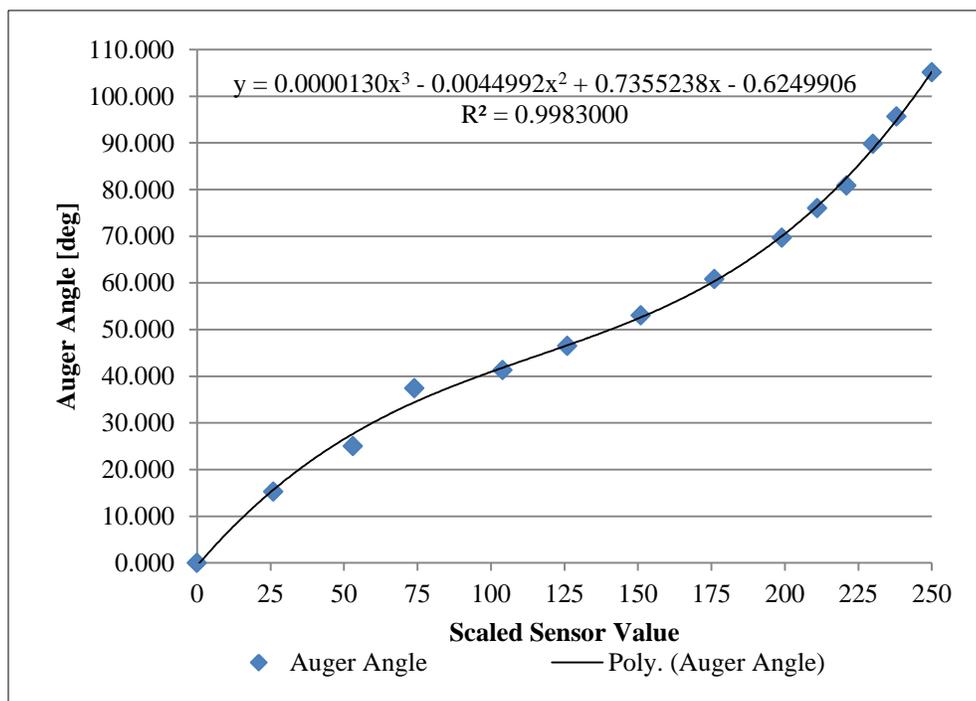
Figure 30: Auger Angle Calibration Fit Curve and Equation

5.7 Additional Sensors

Multiple auxiliary sensors were used throughout the project. A limit switch (wobble sensor) was used to detect grain flow through the auger boot, pressure sensors were used to measure the pressure at each port of the auger-swing hydraulic cylinder, and a rotary potentiometer was used to sense auger angle.

5.7.1 Wobble Sensor

A wobble sensor is a specific type of limit switch. A spring-loaded stick projects out from the sensor body. When the stick is pushed, the switch is depressed. A Honeywell wobble sensor (model number SZL-WL-K-N) was used to sense grain flow (see Figure 31). This switch could be wired for the normally-open or normally-closed configuration. The normally-open configuration was selected. Because the MAB is configured for pull-up digital inputs, when the stick is pressed and the switch closes the digital input value will go to '0' from '1' because it is connected to GND.

Figure 31: Honeywell SZL-WL-K-N

To get the stick in the boot grain flow, a hole was cut in the end of the boot and the sensor body was mounted to the boot using a custom bracket (see Figure 32). Two-conductor wire was routed along the auger and into the cab, where it could be connected to the MAB. Initial testing proved that the grain flow in the auger boot was significant enough to depress the switch.

Figure 32: Wobble Sensor Installation

5.7.2 Pressure Sensors

The pressure sensors were threaded into a tee-fitting at each port of the auger-swing cylinder (see Figure 33). The purpose of instrumenting cylinder pressure was to allow for the pressure profiles during a swing to be quantified.

Figure 33: Pressure Sensor Installation

The sensor selected was the Omega PX309-5KG5V (see the Appendix for detailed information). The '5KG' stands for 0-5000psi range and the '5V' means that the output is scaled 0-5Vdc. The sensor is linear and pre-calibrated. The 5Vdc voltage output was selected to maximize the resolution of the MAB A/D channel. The maximum expected pressure was 3000psi but the 5000psi range provided an appropriate safety factor.

5.8 Summary

The system layout for SmartUnload is large and the interactions between system components occur through different modes. In order to conduct proportional control testing, the hydraulic circuit was modified to include a proportional flow-control valve. The flow-control valves are driven by a separate ECU. The stereo system provides cart fill data and cart tracking data; with the stereo data, the SmartUnload system can close the loop on fill level and locate the cart relative to the combine (which is necessary for accurate control of grain placement). The MAB was used as the primary control ECU which allowed for rapid, iterative testing of new software. Table 2 contains a summary of the instrumented sensors.

Table 2: Sensor Summary

Sensor	Supplier	Model Number	Purpose
Auger Angle	John Deere	AXE14246 A	Sense the auger angle
Wobble	Honeywell	SZL-WL-K-N	Detect grain flow through the auger boot
Pressure	Omega	PX309-5KG5V	Sense swing cylinder port pressure

Chapter 6 Auger System Characterization

6.1 Objectives

The overall objective of the work described in this chapter was to quantify the control characteristics of the combine auger in order to understand the performance potential of auger swing and auger engagement. The response of the auger played an important role in the development of combine auger-boot position control. The first set of tests was conducted to quantify the basic auger swing displacement profile in terms of steady-state angular velocity and overshoot; quantification of the general response profile supported the first phase of design. The relationship between orifice size and auger swing response was determined from the second set of tests; initial on-off control testing demonstrated that auger-swing control was a challenge with the original orifices, and an understanding of the potential improvements through slowing down the auger was desired. The third test set quantified the machine latencies associated with auger swing and auger engagement. The purpose of identifying these individual sources of latency was to justify the current level of control performance and to quantify the potential for improvement in the areas with the largest latencies.

6.2 Materials & Methods

6.2.1 Angular Displacement Profiling

Initial angular displacement profiles were collected over various ranges of motion in the swing-in and swing-out directions. Four tests were completed for two throttle settings (low and high): a full swing-in, a full swing-out, short actuations out at approximately perpendicular, and short actuations in at approximately perpendicular. The purpose of the full swings was to determine the differences in angular velocity throughout the range of motion and between swing directions. The purpose of the impulse (or short step) response tests was to determine the amount of angular overshoot on a given actuation and to look at the difference in angular velocity for a short motion as compared to a full swing. The goal was to develop a general understanding of auger swing dynamics to support initial control design.

Table 3: Angular Displacement Profiling Treatments (1 rep)

Treatment	Factors		
	Swing Range	Swing Direction	Throttle Setting
1	Full	In	High
2	Full	Out	High
3	Short	In	High
4	Short	Out	High
5	Full	In	Low
6	Full	Out	Low
7	Short	In	Low
8	Short	Out	Low

6.2.2 Data Acquisition

LabView and a Measurement Computing USB-1408FS personal measuring device (PMD) were used for data collection; the analog voltage from each channel of the auger angle sensor was collected at 64hz. The PMD analog inputs were configured for differential-ended measurement, and the sensor was powered and grounded at the PMD. LabView recorded a time stamp with each measurement and saved the data in a file format that could be opened in Excel (see Figure 34).
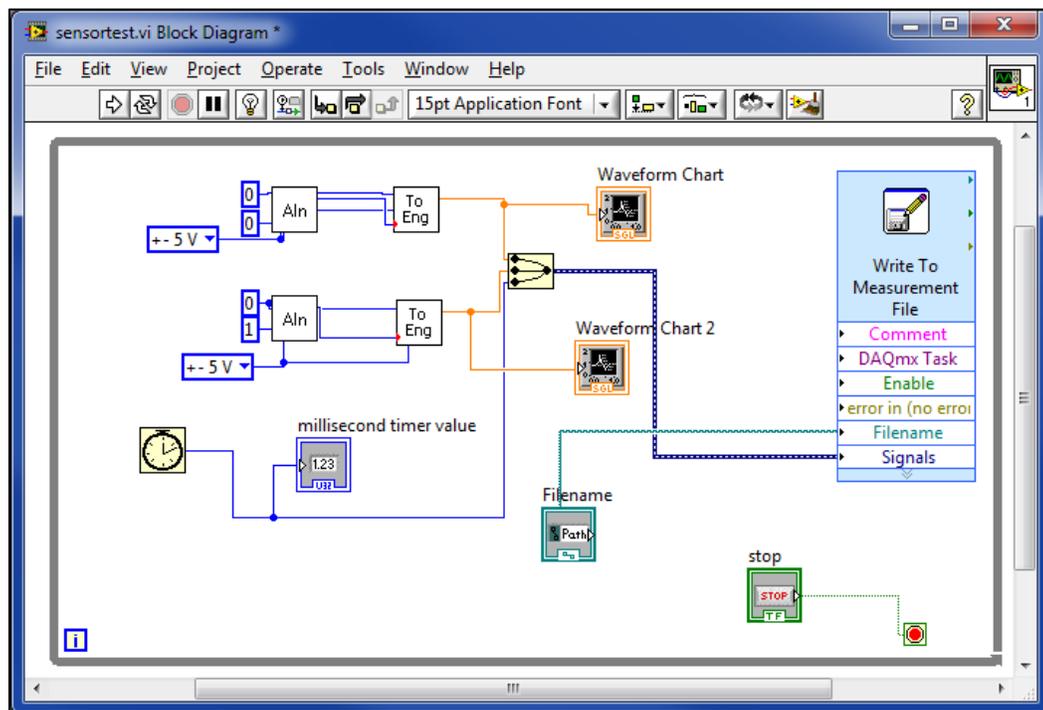
Figure 34: LabView Block Diagram

6.2.3 Orifice Testing

Tests were completed to determine the effect of orifice diameter on swing dynamics. Several orifices are available from John Deere. The factory orifice, part number H118400, is 1.181mm in diameter. Two additional orifices were ordered: N155290 (0.711mm) and N310435 (0.94mm). As a ratio of cross-sectional area compared to the H118400 in the order of smallest to largest, these are 36.2%, 63.4%, and 100%.

To change orifices, the cylinder was removed from the combine; the hydraulic hoses were disconnected from the cylinder ports and the pins were removed from each end before it could be lowered to the ground. After threading in the new orifices, the cylinder could be reinstalled for testing. For each set of orifices, multiple data sets were collected (Table 4). Three factors were varied during the testing: orifice size, swing direction, and swing duration. There were three different levels of orifice size and two different levels of swing direction and swing duration. The nudges and short swings were primarily used to determine the lag times between a commanded actuation and motion as well as auger angle overshoot. The full swings were used to quantify the steady-state angular velocities associated with each orifice (see Figure 35 for a depiction of the response characteristics).
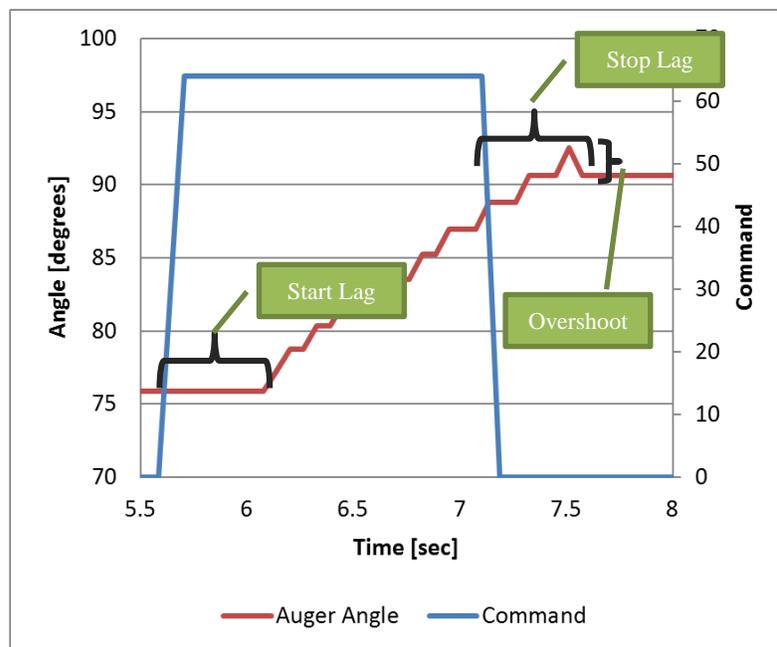
Figure 35: Defining Dynamic Swing Response Properties

Table 4: Orifice Test Treatments (3 reps)

	Factors		
Treatment	Orifice Size	Swing Direction	Swing Duration
1	Small	In	Short
2	Small	Out	Short
3	Small	In	Full
4	Small	Out	Full
5	Medium	In	Short
6	Medium	Out	Short
7	Medium	In	Full
8	Medium	Out	Full
9	Large	In	Short
10	Large	Out	Short
11	Large	In	Full
12	Large	Out	Full

During the collection of this data, the analog voltage from the auger angle sensor was scaled by an intermediate ECU before being sent out on the CANbus. All CAN data was collected using a Vector CANcaseXL board and CANalyzer software. The angle message was available (and collected) at a frequency of 16hz. The first data sets were collected with a scaled-value resolution of 1% and the final data sets were collected with a scaled-value resolution of 0.5%. The resolution was changed throughout the development process to improve controllability, the resolution of the angle used by the stereo software, and to conform to John Deere signal standards; the most recent resolution is 0.4%. The scaled values were converted to an auger angle during data processing (post data collection) by using the auger calibration equation. Commands to actuate were input using the combine hydro-handle buttons; the CAN message containing the hydro-handle signals was available and collected at 5hz. The CAN logs were first sorted by message and plotted against each other in time. Lag times and overshoot values were manually pulled from the data sets and tabulated. Average steady-state angular velocities were obtained by calculating slope between two points in the dataset for several repetitions and tabulated. During all testing, the high throttle setting (2300-2400rpm) was used; this was to ensure consistent pressure was available on the auger-swing portion of the work-function circuit.

6.2.4 System Latencies

The last tests quantified the system latencies associated with auger swing and auger engagement/disengagement. The auger swing latency data was pulled from the factory orifice size data.

Auger engagement latency includes the time for the machine to respond to the command CAN message, the time required for the auger to start rotating, and the time required for the auger to begin discharging grain. Auger disengagement latency includes the time required for the machine to respond to the command CAN message, the time required for the auger to stop rotating, and the time required for the grain flow to fall below a significant level. Auger engagement and disengagement data was collected using the MAB and ControlDesk. Four signals were recorded at 100hz (though the CAN signals updated at slower rates): the digital input for the wobble sensor, the CAN signal for engaging the auger from the hydro-handle, the CAN signal for the unloading auger drive status, and the auger angle. The wobble sensor was connected to the MAB through a digital input and the representative value changes from a '0' to a '1' when grain is flowing through the boot. The hydro-handle button for engaging or disengaging the auger has an associated high-value when pressed. The unloading auger drive signal has a high value when the auger is on. The engagement latency is the time between when the hydro-handle command value goes high and when the wobble sensor value goes high. The disengagement latency is the time between when the hydro-handle command value goes high and when the wobble sensor value goes low. Three separate data logs were collected corresponding to a different treatment combination of auger angle and engine speed. The two engine speed levels were high-throttle and medium-throttle. The two auger angle levels were 85 degrees and 105 degrees. For each treatment, six auger engagement and disengagement cycles were completed. By including these treatment combinations and repetitions, the interaction between engagement and disengagement latencies with auger angle and engine speed could be determined with statistical confidence.

Table 5: Auger Engagement & Disengagement Test Treatments (6 reps)

	Factors		
Treatment	**Throttle Setting**	**Auger Angle [deg]**	**Engagement Transition**
1	Mid	106	Engage
2	Mid	106	Disengage
3	High	106	Engage
4	High	106	Disengage
5	High	85	Engage
6	High	85	Disengage

6.3 Results

6.3.1 Swing profiles

The initial angular displacement profile testing yielded some interesting results. The angular displacement profile is nearly linear and similarly sloped for swing-in and swing-out. A linear swing profile was not expected due to the kinematic relationship between the swing cylinder and the rotation of the auger; also, the fluid and inertial dynamic relationships were expected to cause non-linearity. Collecting and analyzing machine data prior to controls development can reduce development time and improve performance. For high and low throttle settings respectively, the average angular velocities over a full swing-in were 9.7 deg/s and 9.0 deg/s, and the average angular velocities over a full swing-out were about 9.7 deg/s and 10.1 deg/s (Figure 36). It is beneficial to have a consistent angular velocity over the entire range in both directions, because it allows for the control function to remain the same without sacrificing much performance.
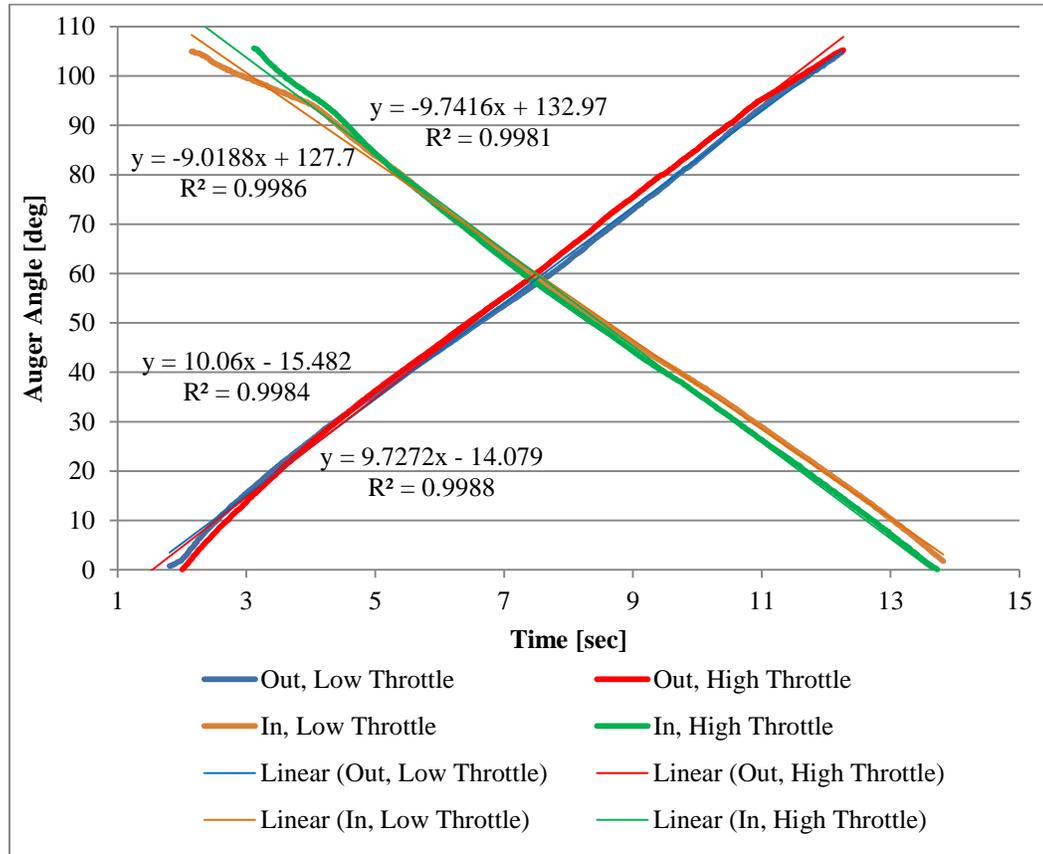
Figure 36: Auger Swing-in and Swing-out Profiles

Figure 37 contains two short actuations in the swing-out direction just past the point when the auger is perpendicular to the side of the combine. The two nudges each have 0.2 degrees of overshoot. The nudge-in profiles were noticeably different than the nudge-out profiles. The nudges in had no overshoot. Overshoot is caused by pressure equalization in the swing cylinder after the valves close; there is no slop in the pinned ends of the cylinder or in the auger elbow (aka turret). The average angular velocity over a short swing is similar to the full swings, so swing distance does not have a large impact on swing speed.
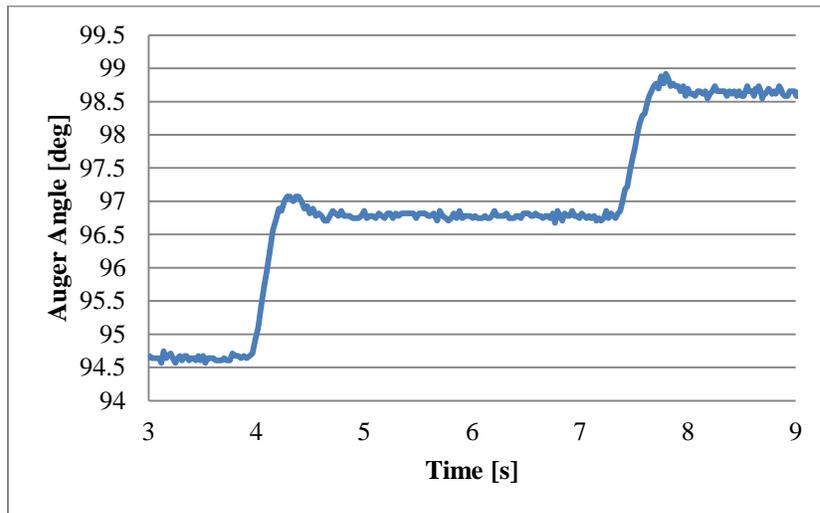
Figure 37: Short Swings Out

Table 6 contains the results for each test. There was no overshoot in the swing-in direction but there was overshoot in the swing out direction; even in the swing-out direction overshoot was relatively small and, therefore, did not affect the control strategy that was implemented for on-off valve control. The range of angular velocities was 8-12 deg/s. To be conservative with control, the dead-bands were tuned for the 12deg/s rate of swing. With larger dead-bands, the control strategy could work for all combinations of swing range and direction but precision of boot placement is limited.

Table 6: Auger Swing Profiling Results

Treatment	Average Angular Velocity [deg/s]	Overshoot [deg]
1	10.1	NA
2	9.7	NA
3	9.38	0.2
4	9.75	0.0
5	9.02	NA
6	9.73	NA
7	7.99	0.0
8	11.92	0.5

6.3.2 Orifice analysis

As expected, the effect of reducing orifice size was a slower auger swing. Figures 38 and 40 contain some representative response profiles and associated commands for each

orifice in the swing-out and swing-in directions respectively. Figures 39 and 41 have applied linear fits in the portion of the curves corresponding to auger motion. The slopes of these curves do not directly match the values in Table 6, because they represent averages over an entire motion (not short swing durations in the 70-105 degree range); however, the slopes do indicate that the auger will complete a full swing-out slightly faster than a full swing-in.
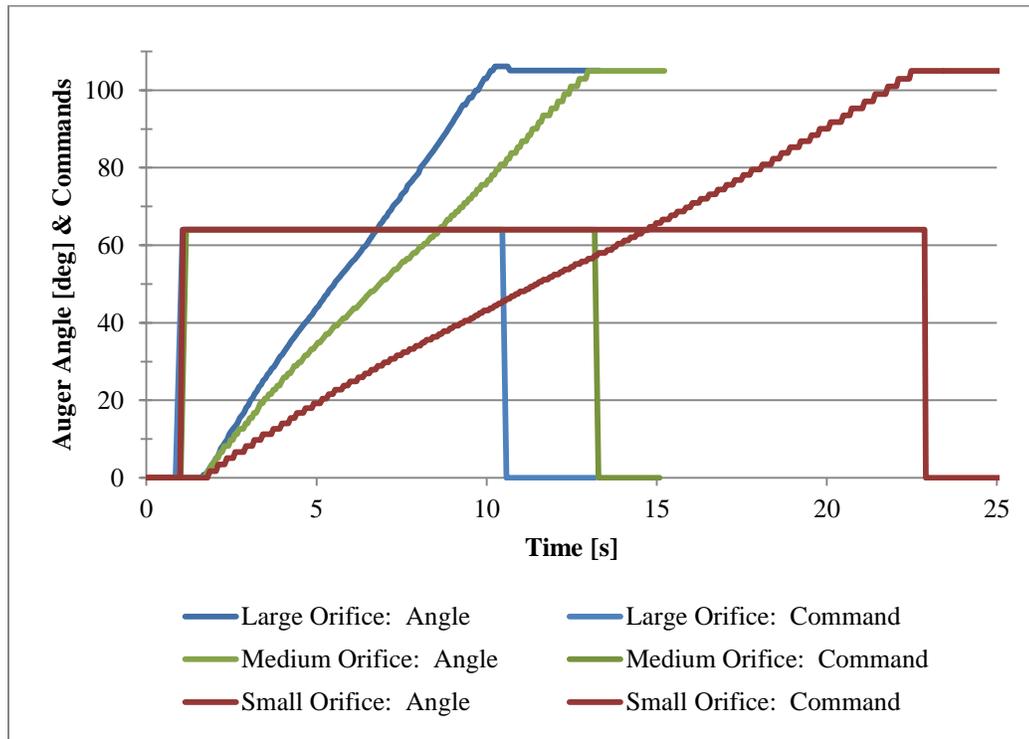
Figure 38: Swing-out Profiles for Orifices
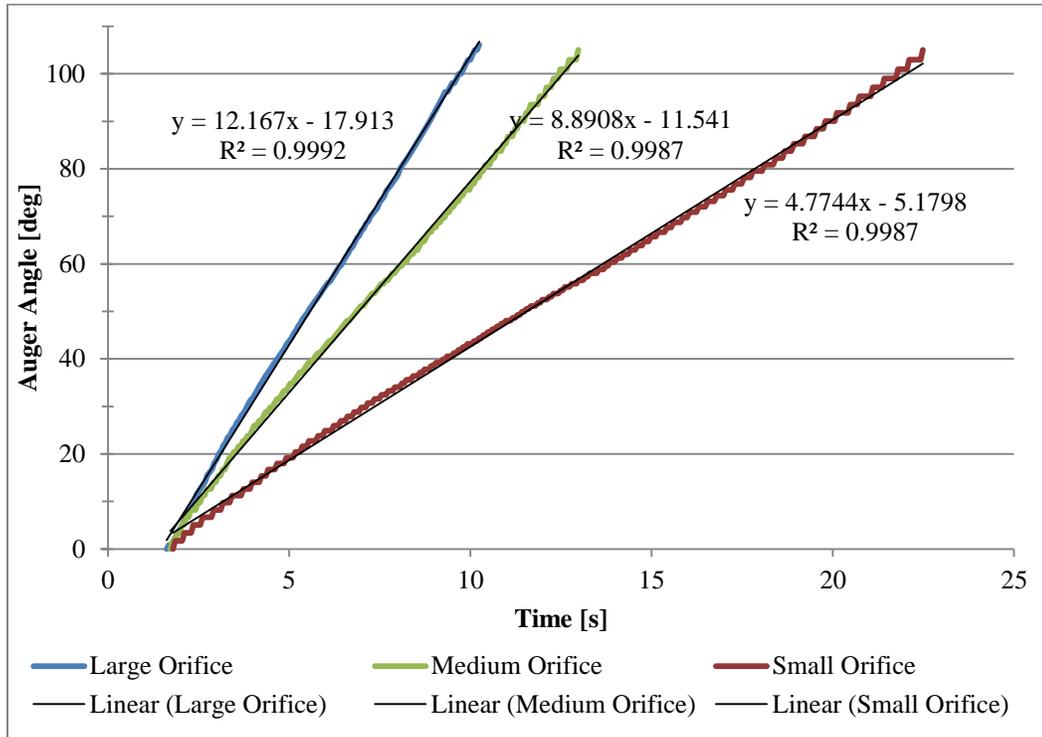
Figure 39: Linear Fit for Swing-out
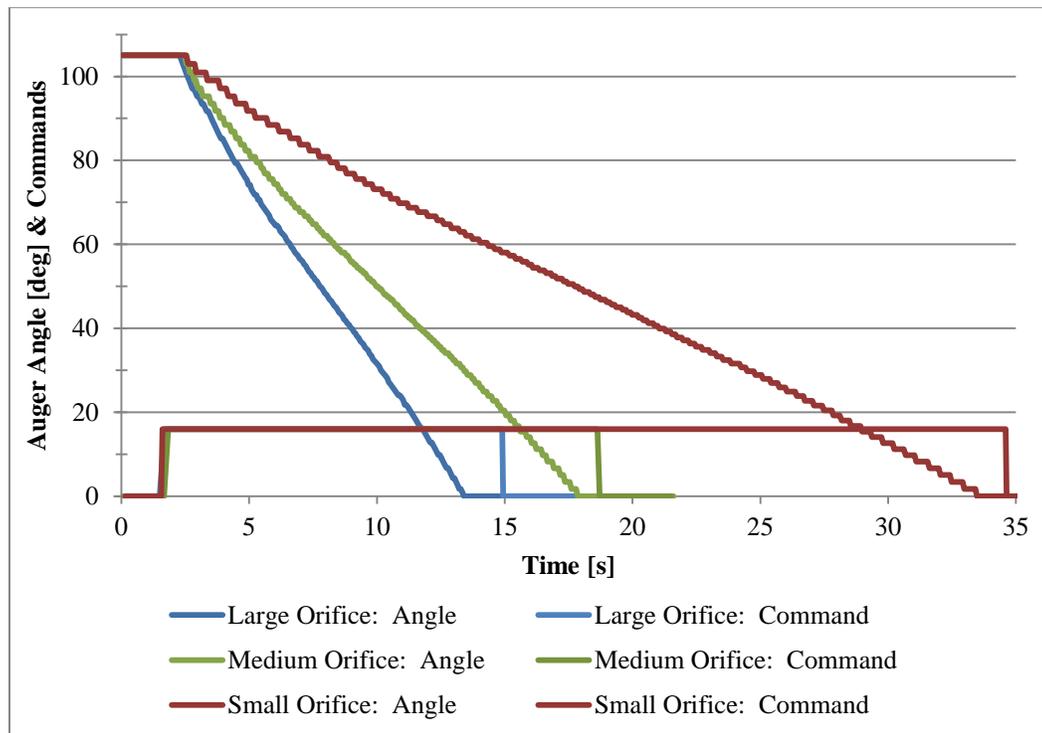
Figure 40: Swing-in Profile for Orifices
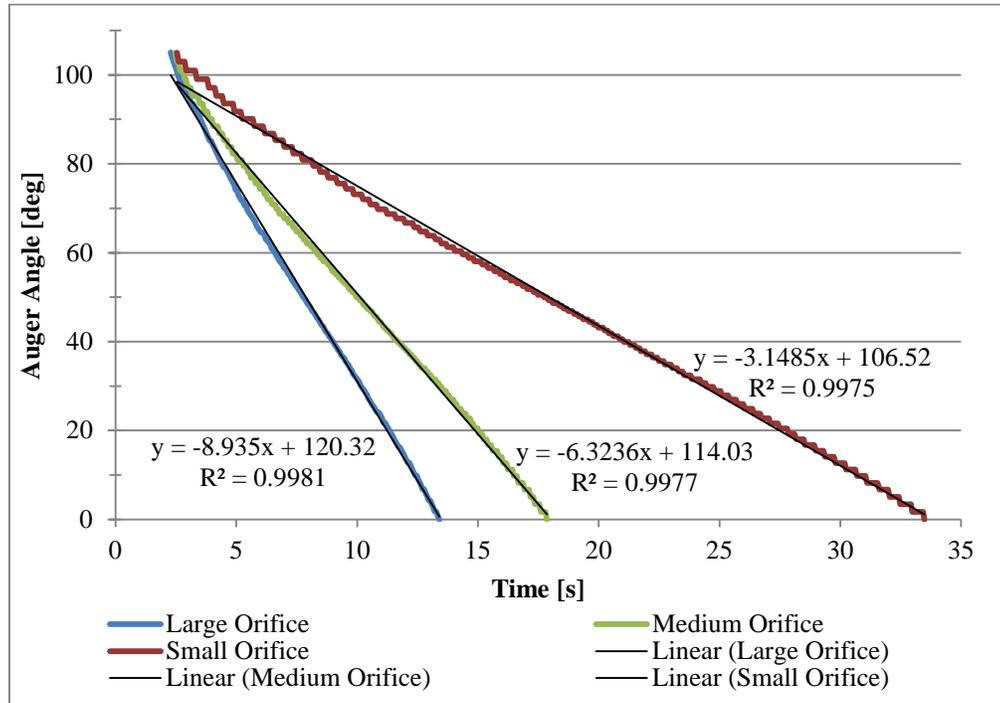
Figure 41: Linear Fit for Swing-in

For swing in, the average angular velocities (over short swing durations) for the small, medium, and large (factory) orifices were 4.9 deg/s, 9.0 deg/s, and 10.9 deg/s respectively. For swing out, the average angular velocities for small, medium, and large orifices were 5.7 deg/s, 11.4 deg/s, and 12.0 deg/s (Table 7 & Table 8). It was surprising that the lag times for starting and stopping the swing did not show any trend based on orifice size; instead, all lag averages were 0.5±0.2 seconds. A clear trend was also not shown for overshoot as a function of orifice size. The latest data collected had twice the angular resolution as the earlier data which resulted in smoother plots and more accurate overshoot values; though the old, lower-resolution data is still useful, analyses moving forward will benefit from the higher data quality. The results indicate that the only benefit to Control Implementation created by reducing orifice size is caused by slower swing speeds; lower angular velocities and consistent lags allow for reduced control dead-bands around a set-point. With reduced control dead-bands, better fill profiles could be possible. Implementing proportional valves or variable orifices to reduce angular velocity around a set point would improve precision; in addition, the higher swing rates needed during spill prevention would still be possible.

Table 7: Swing Characteristics Summary (with Average and Standard Deviation)

Orifice Size	Angular Velocity [deg/s]				Response Stop Lag [s]			
	Swing-In		Swing-Out		Swing-In		Swing-Out	
	$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s
Small	4.9[a]	0.9	5.7[a]	1.5	0.35[a]	0.03	0.50[a]	0.33
Medium	9.0[b]	2.1	11.4[b]	0.8	0.52[a]	0.08	0.41[a]	0.13
Large	10.9[c]	0.4	12.0[b]	0.6	0.38[a]	0.06	0.54[a]	0.17

Table 8: Cont. Swing Characteristics Summary

Orifice Size	Response Start Lag [s]				Overshoot [deg]			
	Swing-In		Swing-Out		Swing-In		Swing-Out	
	$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s
Small	0.45[a]	0.19	0.13[a]	0.02	0.0[a]	0.0	1.3[a]	1.1
Medium	0.44[a]	0.14	0.31[b]	0.05	0.0[a]	0.0	1.3[a]	1.1
Large	0.18[a]	0.17	0.23[a,b]	0.08	0.7[b]	0.2	1.2[a]	0.5

6.3.3 Latency analysis

The latencies for auger-swing start and stop were taken directly from the values in Tables 7&8 for the large orifice. The average time required for the auger to respond to a command to start swinging in is 0.18 seconds. The average time required for the auger to start swinging out is 0.23 seconds. The average time for the auger to stop swinging in is 0.38 seconds, and the average time for the auger to stop swinging out is 0.54 seconds.

Auger disengagement and engagement lags were also determined. Table 9 contains the average and standard deviation for each test. Treatment 1 was conducted at high throttle (typical of unloading on-the-go during harvest) and low-angle (85 degrees). Treatment 2 was at medium throttle and maximum auger angle. Treatment 3 was at high throttle and maximum auger angle. All three treatments have means within 14% of each other for both, disengagement and engagement. The most similar results were between treatments 1 & 3, which also correspond to the most likely operating conditions for SmartUnload during harvest. The disengagement lag is important; the time required to stop grain flow directly relates to the performance potential of spill prevention in the SmartUnload application. The engagement lag is also important to understand, because operators may incorrectly associate

the lag with the SmartUnload software instead of the mechanics of the combine's auger engagement system.

Table 9: Auger Disengagement & Engagement Lag Statistics

		Disengagement Lag [s]	Engagement Lag [s]
Treatment 1	$\bar{x}$	0.95	3.01
	s	0.07	0.14
Treatment 2	$\bar{x}$	0.86	3.22
	s	0.08	0.22
Treatment 3	$\bar{x}$	0.98	2.07
	s	0.08	0.06

The average overall actuation latency for each type of actuation is given in Figure 42; there is room for significant reduction (improvement) in these response times. The most important reduction in latency could be accomplished by speeding up the auger disengagement mechanism; doing so would greatly improve spill prevention capabilities and increase the amount of the cart that could actively be filled without increased spill risk. Decreasing auger engagement time would improve the operator's perception of the system as it starts up; the existing delay is very noticeable and could impact product acceptance in the market. The swing latencies are small compared to the auger engagement latencies. The time required to stop the swing relates directly to the control dead-bands required around the desired location; the latency could be reduced by using a faster controller and valve. The time required to start swinging is small and does not have a large impact on system performance or operator perception. The portion of each lag associated with stereo data and control latencies is only 0.7 seconds on average and no improvements are possible at this time.
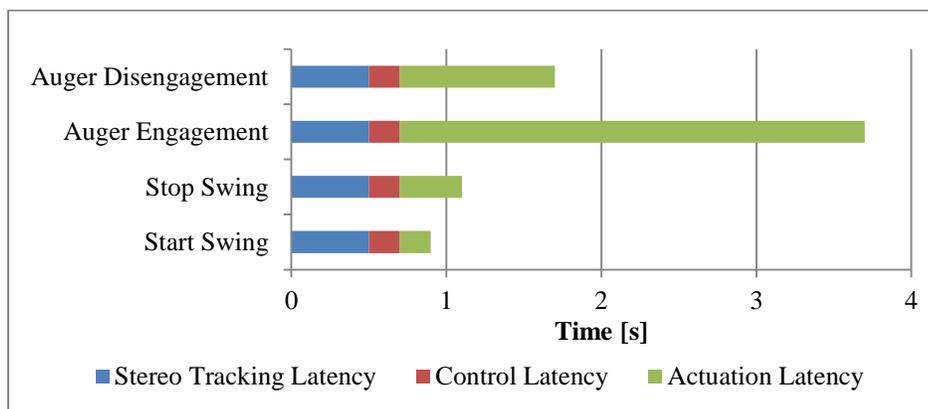
Figure 42: Time-series Latency Plot

44

6.4 Conclusions

Based on the swing profile testing, the angular velocity throughout the full range of motion is close to 10deg/s in each direction and independent of the duration of motion. Also, overshoot values were relatively small. Because of these results, the same on-off control strategy was used for all initial SmartUnload operating conditions.

Orifice-testing results fit the expected trend; as orifice size decreases, the angular velocity of the auger decreases. Though start lags, stop lags, and overshoot were not impacted, the slower auger velocities do improve controllability and allow for finer position control. During a given stop-lag period, the auger will not swing as far and thus the dead-band around the desired location can be smaller without causing oscillations. It is not desirable to make significant sacrifices in the maximum swing speed; proportional control was pursued to allow for better controllability near the desired location and faster swing speeds for larger displacements or spill prevention.

A latency analysis was completed to understand the overall system latency and the portion of that associated with each step in the control process. The swing latencies were pulled from the orifice data taken with the factory-size orifice. Separate testing was completed for auger engagement/disengagement. The total lag associated with SmartUnload -commanded auger disengagement is about 1.5seconds. Auger engagement lag is about 3.5 seconds. Swing-stop lag is about 0.9 seconds and swing-start lag is about 0.7 seconds. Figure 43 is a flowchart of lag accumulation for SmartUnload, using the average for each sub-component rounded to the nearest tenth of a second. Up to 0.2 seconds was included for the execution time of the control function because it is operating at 5hz. Moving forward, work can be done to improve the quality of the tracking data; given a relative velocity, it may be possible to calculate a correction factor for tracking. A significant improvement in spill prevention capability can be made by increasing the actuator speed for auger engagement/disengagement; less grain will be spilt if the auger can be turned off more quickly. By increasing the actuator speed on auger engagement/disengagement, SmartUnload efficiency and operator perception will be improved, because the auger will spend less time in the disengaged state during an unloading event.
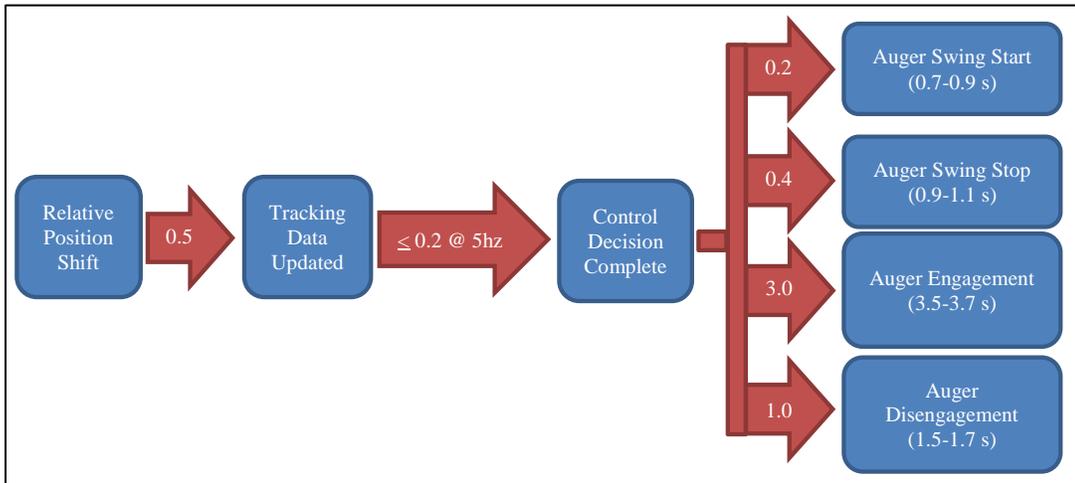
Figure 43: SmartUnload Latency Depiction

If this testing were to be completed a second time, hydraulic pressure data should be collected for both swing-cylinder ports. Collecting pressure data would allow for a comparison of swing forces as a function of orifice size and swing speed. Also, using the MAB to collect the data at 100hz with the most recent auger position resolution of 0.4% would improve accuracy and quality.

Chapter 7 Static Proportional Control Tests

7.1 Objectives

The primary goal of completing static proportional control testing was to quantify the difference in performance between proportional control and on-off control. Several factors besides control-type were included to determine their respective impact on the performance comparison. The two main metrics used to compare performance were maximum port pressure magnitudes and average angular velocity. A difference in maximum port pressures indicates a difference in component stress and wear. Average angular velocities were compared to determine the change in total time required to make a swing.

Due to the large number of factors considered, several secondary objectives were achieved. The magnitude of pressure at each port of the swing cylinder was quantified for proportional and on-off control. The performance of proportional control at two different orifice sizes, 1.181mm (factory) and 2.33mm (drilled out), was evaluated. The effect of auger fullness on swing response and hydraulic pressure was also quantified. A single comparison of proportional gains was made. Lastly, the effect of swing direction and swing range was considered.

Steady-state error was analyzed, and the results did not show a difference between control types; the dead-bands were tuned to minimize steady-state error for each control type. The difference in precision capability is instead indicated by the control dead-bands needed to prevent instability.

7.2 Methods & Materials

7.2.1 Control Methods

The first step in preparing for formal proportional control tests was to understand the functional range of the PWM valve-driver signal and to program the valve-driver board to have the maximum control resolution within that range. The PWM output value is represented by an unsigned, 16-bit integer (0-65535 corresponds to 0-100% duty-cycle). To determine the functional PWM range, the valve-driver was programmed to accept a PWM value directly. The PWM range was varied until the maximum angular velocity for the

factory orifice was determined. Then the PWM value was reduced to the minimum duty-cycle required to just barely swing the auger. The effective maximum and minimum PWM values for swing-in are 28250 (43.1% duty-cycle) and 11500 (17.5% duty-cycle). The effective maximum and minimum values for swing-out are 30750 (46.9% duty-cycle) and 11500 (17.5% duty-cycle). Linearizing over an input of 0-250 (unsigned, 8-bit integer) gives Equations 6 & 7 for swing-in and swing-out. These equations are used to control the duty-cycle of the 100hz PWM signal.

$$PWM = 67 \times Input + 11500$$
<div align="right">**Equation 6**</div>

$$PWM = 77 \times Input + 11500$$
<div align="right">**Equation 7**</div>

The proportional-control implementation within the model used for fall testing will be described in greater detail in the Appendix and was not used for this control testing. Instead, a separate model was developed to close the control loop on auger angle instead of boot location (Equation 8). A dead-band of ± 1 degree was used to prevent valve cycling caused by noise in the auger angle.

$$Output = P \times (DesiredAngle - CurrentAngle)$$
<div align="right">**Equation 8**</div>

This reduced the resolution of control, because auger angle in 0.5 degree increments is only a value between 0-214 whereas boot location is a value measured in centimeters and is a value between 0-600 for a typical cart size. However, the benefit of controlling on auger angle is to remove as many uncontrolled factors from the experiment as possible (such as stereo system inaccuracies). The proportional output was scaled between 0-250, because the unsigned, 8-bit integer data type was required for this CAN signal.

The on-off control was based solely on a ± 4.5 degree dead-band (Equation 9).

$$\begin{cases} SwingIn\ IF\ (DesiredAngle - CurrentAngle < -4.5) \\ SwingOut\ IF\ (DesiredAngle - CurrentAngle > 4.5) \\ ELSE\ NoSwing \end{cases}$$
<div align="right">**Equation 9**</div>

7.2.2 Test Design

Several factors were included in the test design: control type, orifice size, auger fullness, p-gain, swing range, and swing direction. Control type levels included proportional and on-off. Orifice sizes were 1.181mm (factory) and 2.33mm (drilled out). Auger fullness levels were empty, full, and unloading. A high and low p-gain was selected for swing-in and swing-out; swing-in used 80 and 50, and swing-out used 80 and 40. The 85-100 degree

swing range was compared to the 90-97degree swing range (15 degree and 7 degree step inputs). The swing-out direction was compared to the swing-in direction.

In the end, a test matrix including twenty-two treatments, with three repetitions of each, was used to guide the testing (Table 10). Treatments one through twelve were used to compare proportional control to on-off control. A full factorial with auger-fullness and swing direction was completed. Orifice size remained fixed because using on-off control with a much larger orifice could have damaged the machine.

Treatments thirteen to twenty-two utilized the larger orifice. P-gain, auger fullness, and swing direction were the additional factors varied.

Table 10: Static-Control Test Matrix (3 reps)

Treatment	Start Angle [deg]	Stop Angle [deg]	Angle Dead-band [deg]	Control Type	Orifice Size [mm]	Auger Fullness	P-gain
1	85	100	4.5	On-off	1.181	Empty	0
2	100	85	4.5	On-off	1.181	Empty	0
3	85	100	4.5	On-off	1.181	Full	0
4	100	85	4.5	On-off	1.181	Full	0
5	85	100	4.5	On-off	1.181	Unloading	0
6	100	85	4.5	On-off	1.181	Unloading	0
7	85	100	1.0	Proportional	1.181	Empty	80
8	100	85	1.0	Proportional	1.181	Empty	80
9	85	100	1.5	Proportional	1.181	Full	80
10	100	85	1.0	Proportional	1.181	Full	80
11	85	100	1.0	Proportional	1.181	Unloading	80
12	100	85	1.0	Proportional	1.181	Unloading	80
13	100	85	1.5	Proportional	2.33	Empty	50
14	85	100	1.0	Proportional	2.33	Empty	80
15	85	100	1.0	Proportional	2.33	Full	40
16	100	85	1.0	Proportional	2.33	Full	40
17	100	85	1.5	Proportional	2.33	Full	50
18	85	100	1.0	Proportional	2.33	Full	80
19	90	97	1.0	Proportional	2.33	Full	80
20	97	90	1.0	Proportional	2.33	Full	80
21	100	85	1.5	Proportional	2.33	Unloading	50
22	85	100	1.0	Proportional	2.33	Unloading	80

The control dead-bands were selected based on control type. For on-off control, the dead-band had to be at least 3 degrees to prevent oscillation and instability during the tests; a dead-band of 4.5 degrees was selected to reduce the steady-state error. The p-gain of 80 for treatments 7-12 was selected to provide a compromise between fast response and high precision. For the proportional treatments and the original orifice, the dead-band could be 1.0 degree which highlights the improved precision. For the 2.33mm orifice tests, multiple p-gain levels were tested and the dead-band had to be increased to 1.5 degrees to prevent oscillations; a p-gain of 80 also caused small oscillations, so 40 and 50 were tested.

Orifices threaded to fit the cylinder ports were not available in a larger size, so extras were purchased and drilled out. The purpose of testing the larger orifices was to determine if proportional control would still function well at higher maximum angular velocities. Increasing the maximum angular velocity would negate the loss in average angular velocity caused by proportional control when close to the desired angle. Maintaining or increasing the average angular velocity of the auger would allow for faster swing performance. Relative to the factory size, the modified orifice has a 389% cross-sectional area.

The auger fullness levels were labeled empty, full, and unloading. An empty auger contains the minimum amount of grain and is achieved by unloading the auger until no more grain is coming out (the grain tank is emptied). A full auger contains the most grain possible without continuously unloading; the auger is filled by unloading at full throttle until full grain flow is achieved and then turning it off. An unloading auger is defined by auger engagement with the combine at full throttle, and with a non-empty grain tank, to achieve maximum grain flow.

7.2.3 Test Setup

These tests were completed in the gravel yard at BCRF. Figure 44 shows the setup. A small utility tractor hooked to an auger was used to run grain from the wagon into the combine for the full-auger and unloading-auger tests. For the unloading tests, the combine needed to be refilled several times. When the cart was full, it was dumped into the wagon. The combine remained stationary throughout the testing.

Figure 44: BCRF Yard Testing

7.2.4 Data Analysis Methods

Data processing and analysis included calculations and plots at the repetition level and at the treatment level. Signals (cylinder port pressures, auger angle, and PWM duty-cycle) were plotted over time to determine the effective start and stop times for each repetition and to make sure all of the data was present and valid. The effective cylinder cross-sectional area was calculated for the rod-end and cap-end; the value corresponding to the swing direction was selected for each repetition. Average angular velocity and the steady-state error were also calculated for each repetition.

The average angular velocity calculation is given in Equation 10. Steady-state error is the difference between the true final swing angle and the target final swing angle.

$$\dot{\theta}_{avg} = \frac{\theta_f - \theta_0}{t_f - t_0}$$ **Equation 10**

The average and standard deviations of the three repetitions for each treatment were calculated for comparison. The data was also imported into MiniTab to utilize the advanced statistical plotting functions. Additional statistics were calculated pertaining to the differences between proportional control treatments and on-off control treatments; these values quantitatively describe the trends seen in the individual value plots.

7.3 Results

Figure 45 contains the pressure and angular profiles for a proportionally-controlled swing-out from eighty-five to one hundred degrees. The proportional-control angular profile has a decreasing slope as the desired angle is approached, as expected. As the angle approaches the desired value, the valve duty-cycle is reduced and the inlet pressure (cap-end) decreases. At the desired angle, the inlet pressure becomes constant and the outlet pressure remains low.
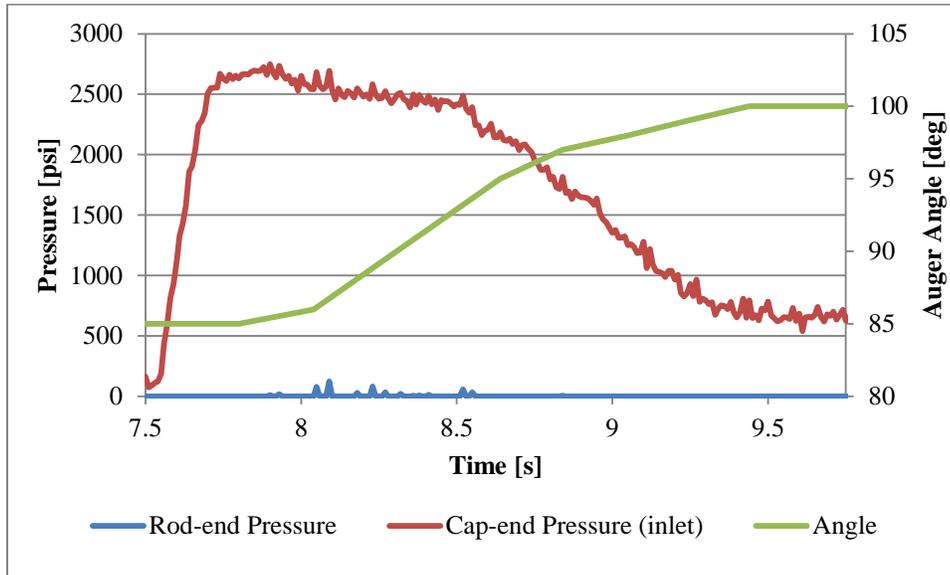
Figure 45: Proportional Control Pressure Response Sample

Figure 46 contains the on-off control profiles. The slope of the angular displacement plot remains consistent throughout the swing. The inlet pressure stays nearly constant at the maximum value during the swing. The auger quickly stops at the desired angle and the outlet pressure spikes; the inlet and outlet pressures are nearly equal at steady-state.
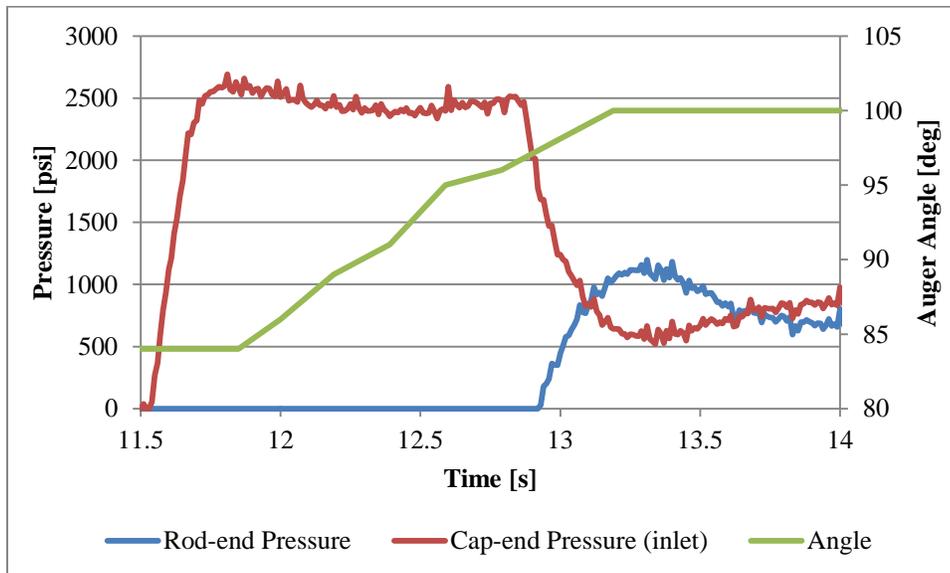
Figure 46: On-off Pressure Response Sample

Figures 45 and 46 are representative of the angular and pressure profile characteristics for the respective control types. Comparing the two control types, it is evident that proportional control has a reduced average pressure at the inlet. Proportional control also prevents the spike in outlet pressure as the swing stops. Maximum outlet pressure is reduced

by 1384psi on average with a 95%-CI of 274psi. This is significant because it may mean that parts will wear less and last longer without failure.

The average angular velocity is also reduced when using proportional control (Figure 47). Quantitatively, the angular velocity reduction is 17% on average with a 95%-CI of about 6%. This means that a controlled actuation of equal displacement will take 17% longer on average. 17% greater swing time is not detrimental to performance and the gain in precision outweighs the loss in swing speed.
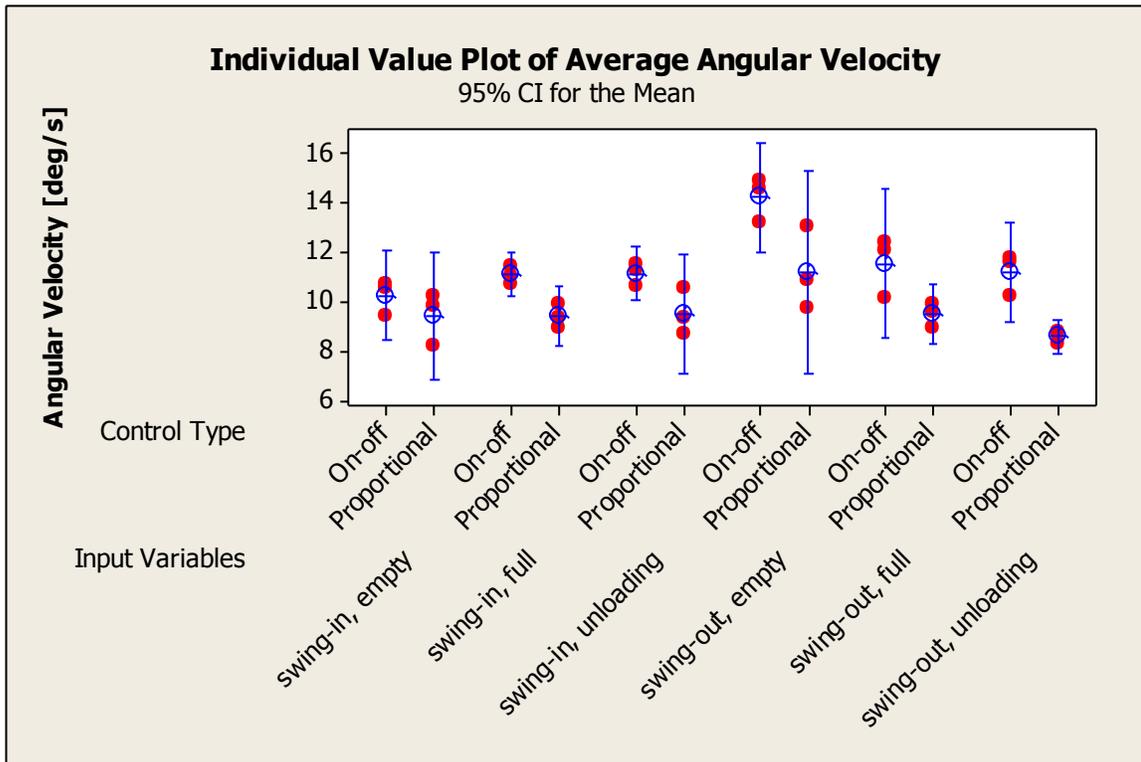
Figure 47: Impact of Control Type on Average Angular Velocity

Control type did not impact steady-state error consistently across the different treatments (see Figure 48). The dead-bands were tuned to minimize steady-state error and prevent instability, so the output data does not show the improved control precision possible with proportional control. However, the specified dead-band does show the improvement; for proportional testing, the dead-band was less than half the magnitude required for on-off control stability.
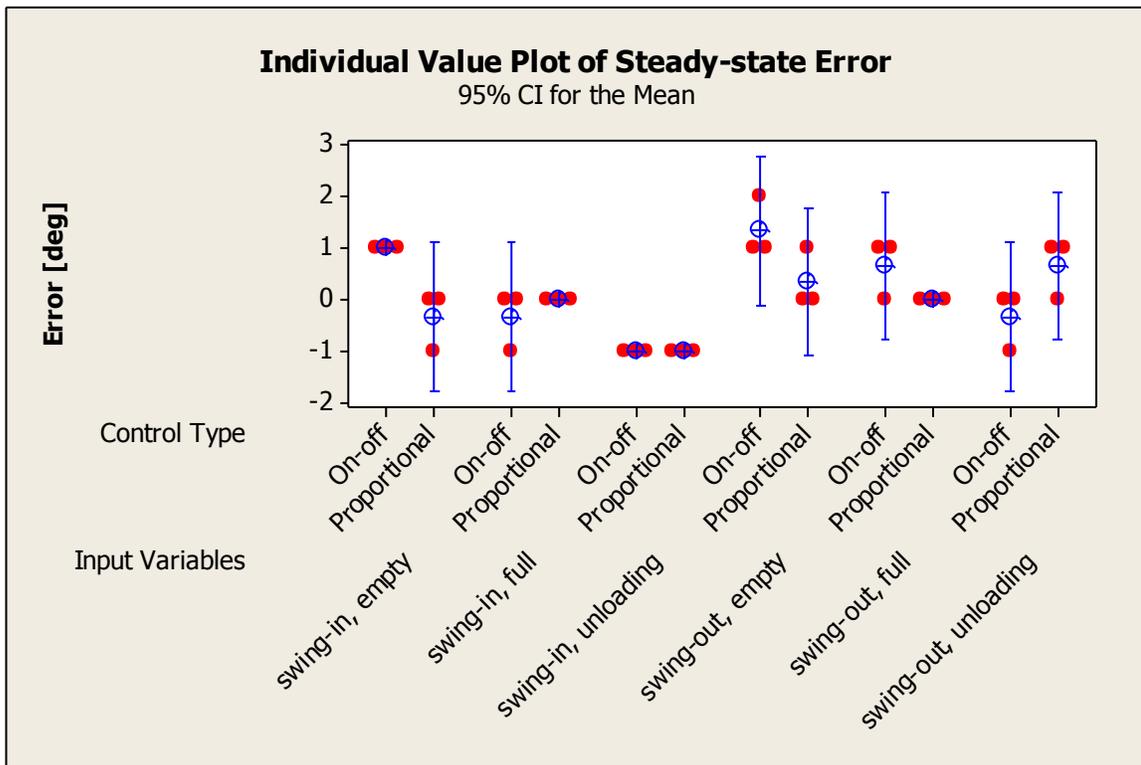
Figure 48: Impact of Control Type on Steady-state Error

Orifice size has a large impact on the expected maximum outlet pressure when using proportional control (Figure 49); however, proportional control with the large orifice still has a lower maximum outlet pressure than using the smaller orifice with on-off control. Using proportional control will reduce the pressure rise at the cylinder outlet port during swing stops.
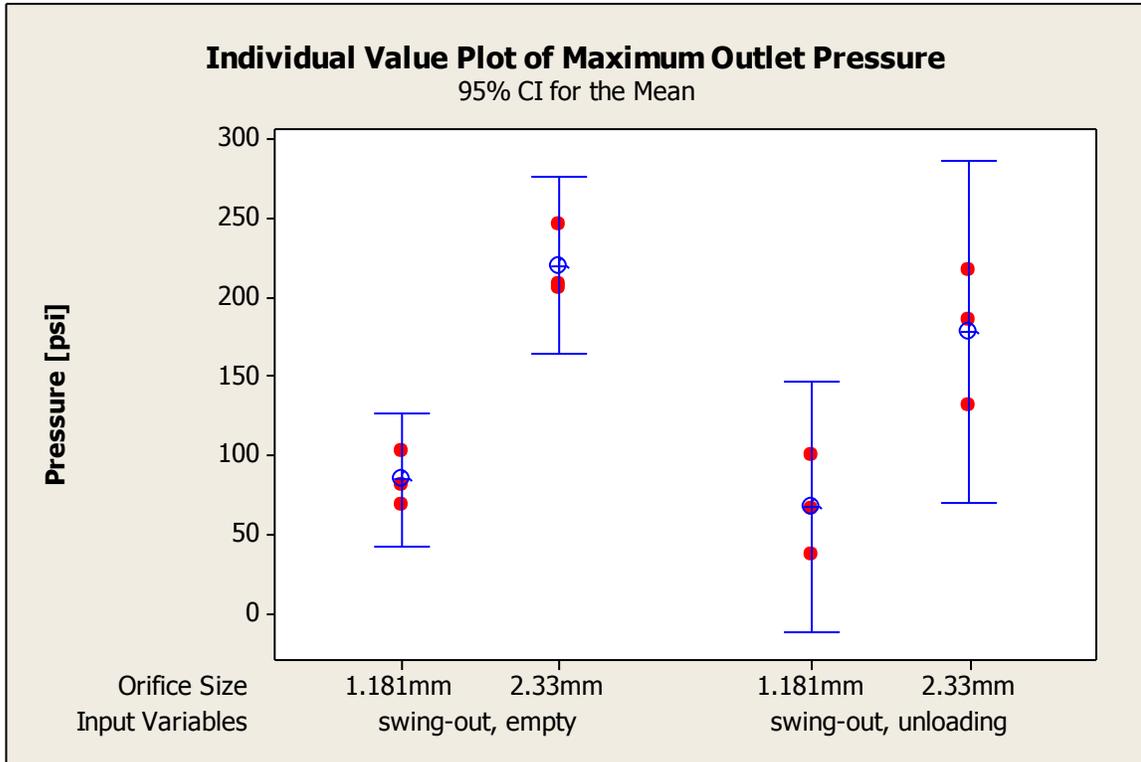
Figure 49: Impact of Orifice Size on Outlet Pressure

A larger diameter results in a smaller pressure drop across the orifice. With a smaller pressure drop across the orifice, less pressure is developed on the inlet side when swinging the auger. A larger orifice also allows more flow to the cylinder and the average angular velocity increases (Figure 50). This was the main reason for testing a larger orifice size. The intention was to make up for the slower swing rates required for higher precision near the desired location by allowing for higher maximum swing rates. Using only a single proportional gain, the system was not able to perform well with large orifices. To maintain stable operation, a small gain was needed but then no increase in maximum swing rate was possible. Gain scheduling may be a solution to this problem. Larger proportional gains can be used when the error is large but should be decreased as error decreases to ensure the swing can be stopped within the desired range.
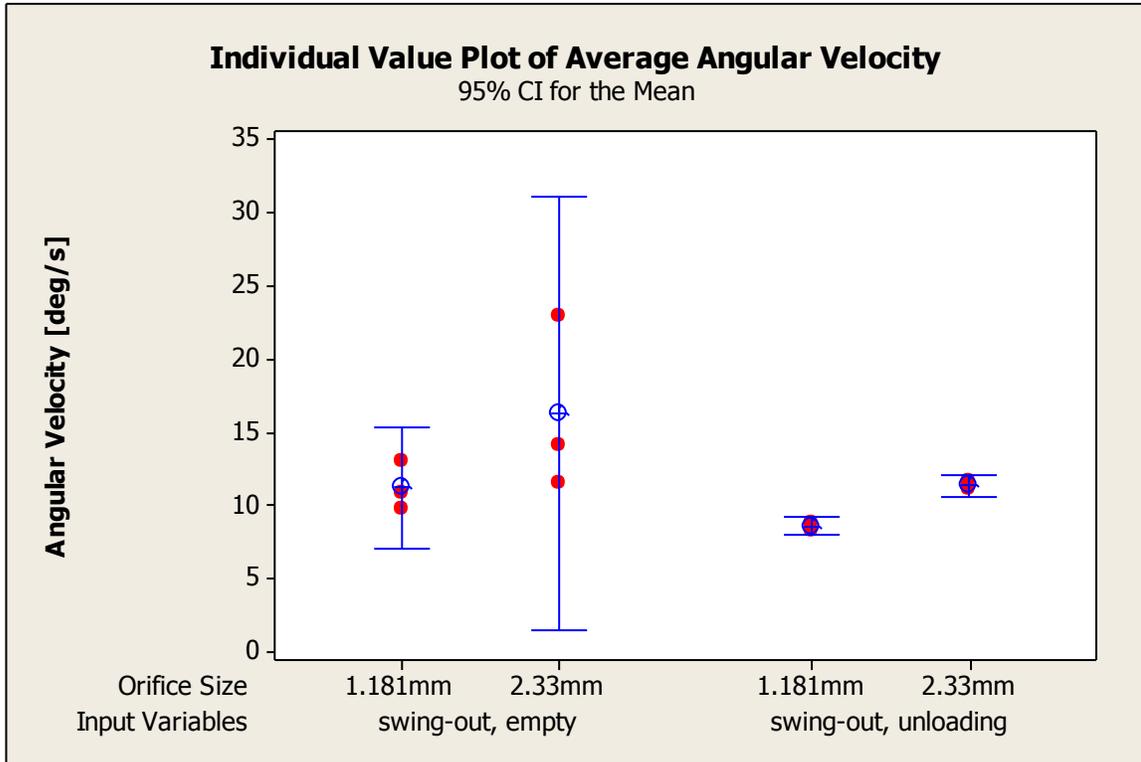
Figure 50: Impact of Orifice Size on Average Angular Velocity

Figure 51 clearly indicates that the required inlet pressure to swing the auger increases as auger fullness increases, especially between the empty and full levels. The majority of the SmartUnload duty-cycle will be with an unloading auger, but the auger will also spend a portion of time in the other two categories. This supports the need to conduct test treatments using all three levels of auger fullness. The effect of auger fullness on angular velocity was inconsistent across different treatments, so no control changes can be recommended based on auger fullness.
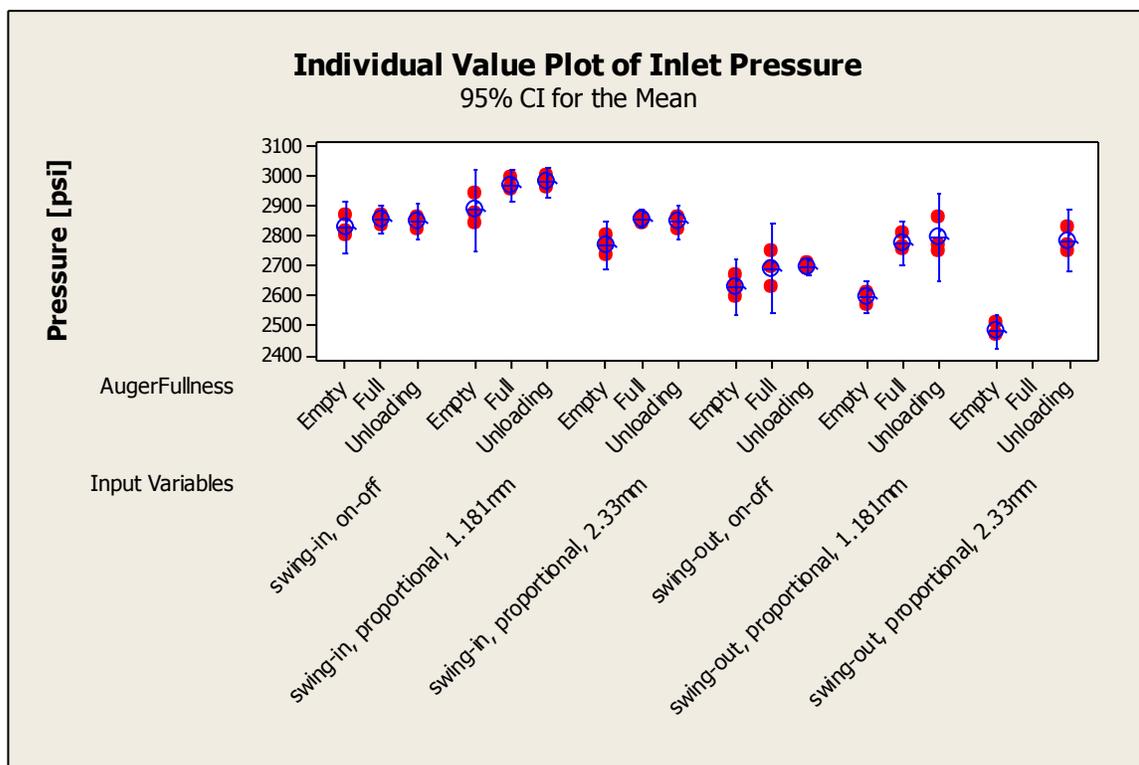
Figure 51: Impact of Auger Fullness on Inlet Pressure

7.4 Conclusion

The results of the static proportional control testing were very promising. Proportional control results in lower average inlet pressures and higher control precision through the use of smaller dead-bands. Standard John Deere auger lengths are 6.9m and 7.9m; the combine used for this testing had the 6.9m auger. With control dead-bands of 1deg and 4.5deg for proportional and on-off control respectively, the corresponding boot placement precisions are $\pm$12cm and $\pm$54cm; Figure 52 further emphasizes this performance advantage. Additionally, the maximum outlet pressure is reduced for controlled swings, because the inertia of the auger is decelerated more slowly; this could result in longer component life.

Figure 52: Boot Precision Depiction (red zone with proportional, blue zone with on-off)

Using proportional control does require a longer time-period to swing over the same displacement when using the same orifice size. However, the increased time requirement is not detrimental to fill quality (Chapter 8). By using proportional control, the dead-band around the desired location can be much smaller allowing for higher fill-location resolution. With some modifications to fill strategy, there will be potential for smoother fill profiles and higher maximum cart weights. The higher control resolution is not obvious by looking at only the steady-state error for a set of swing profiles, because on-off control dead-bands have been tuned to eliminate oscillations and minimize steady-state error. The goal was to compare the best possible performance of each control type; this will allow for a more useful cost-benefit analysis of proportional control.

Increasing orifice diameter has some benefits including reduced inlet pressure and higher potential swing velocities. Because of the higher maximum swing velocity, the p-gain must be reduced in order to decelerate the auger in time to prevent overshoot of the desired location. P (proportional) control with a larger orifice has not been shown to be beneficial due to the reduced controllability or the reduced p-gains. Proportional control with gain scheduling should be investigated as a possible solution to effectively utilize larger orifices.

Auger fullness does impact swing dynamics in a meaningful way and each level should be included in any future testing. As auger fullness increases, the maximum inlet

pressure increases. Other secondary test factors were p-gain, swing range, and swing direction. The results associated with those factors were not utilized during the initial controls development, but bar charts indicated some interesting trends. When p-gain increased, the maximum inlet pressure increased and the average angular velocity increased; allowing more flow through the valve results in higher line pressures and faster actuations. Swinging farther gives the system more time at the maximum angular velocity (resulting in higher average angular velocities). Swing direction was varied for each test set so there were some strong trends discovered. Maximum inlet and outlet pressures are higher when swinging in. Each swing direction was impacted differently by changes in p-gain and implementing separate control functions is likely to improve performance.

Chapter 8 Field Evaluation of Swing Control

8.1 Objectives

The primary objective of the field testing was to quantify the benefit (additional weight in a grain cart) that results from using smaller, more discrete, actuations when unloading. The smaller swing actuations are only possible when using proportional control.

For the SmartUnload project, it was also desirable to quantify the typical actuation effort and efficiency of an automated unload sequence. Actuation effort is defined by the number of swings per unload, the number of auger engagements per unload, the total angular displacement, the average swing displacement, and the average duration of a swing (measured in seconds). Efficiency is defined as the percentage of time the auger is on during full system engagement (while tracking a non-full cart).

8.2 Methods & Materials

Significant planning went into fall testing. The test matrix was designed to allow for three replications of each treatment. Resource allocation in the form of acres, equipment, and people was part of the planning. Efficient test execution and data collection was a high priority, because other requirements within the SmartUnload project had demand for equipment and the harvest window was relatively narrow. Each data file was checked during harvest to ensure that all of the needed signals were recorded for the duration of each test. The data was processed using a MatLab script and the calculated metrics were written to Excel for further analysis.

8.2.1 Test Matrix Definition

The factors considered during harvest testing were control type and the number of fill zones used (the Appendix contains detailed descriptions of these factors). The levels of control type were proportional and on-off. The levels of fill-zones were 2, 3, and high. A high number of fill zones is considered anything greater-than-or-equal-to 4. A full factorial was executed for these two factors; however, no tests were conducted using on-off control with a high number of fill zones, because the system is not capable of functioning under those constraints without significant oscillation. The target number of repetitions for each

treatment was selected prior to harvest, but additional repetitions were completed opportunistically during harvest. The numbers of repetitions planned and executed are provided in Table 11.

Table 11: Swing-Control Harvest Test Matrix

Treatment	Control Type	Number of Fill Zones	Repetitions Planned	Repetitions Executed
1	Proportional	2	5	7
2	Proportional	High	5	6
3	On-off	2	5	5
4	On-off	3	3	5
5	Proportional	3	3	3

8.2.2 Resource Allocation

The equipment required and used to complete the control testing was a combine, grain cart, grain cart tractor, wagons, and wagon tractors (Figure 53). The number of people required for testing was four. Three equipment operators ran the combine, grain-cart tractor, and wagon tractors respectively. The last person was responsible for data collection.

The number of acres reserved for the control testing was a function of the number of tests in the test matrix, the predicted number of acres required per test, and a factor of safety based on the previous year's SmartUnload testing. The number of test repetitions planned was 21. The predicted number of acres required per test (for corn) was about 1.5, based on 165 bu/ac yield and 250 bu/test. The factor of safety based on testing experience during harvest 2010 was 2; only half of the attempted test runs were successful in 2010. Multiplying the number of tests by the acres required per test and the factor of safety resulted in a predicted land requirement of 63 acres. This coincides very closely with the amount of land utilized during harvest. The South Woodruff field, at the ISU Research Farm, was completely dedicated to swing-control testing and a few acres of the Been field were utilized to finish the remaining repetitions.

Figure 53: South Woodruff Staging Area

8.2.3 Test Execution and Data Collection

The treatments and repetitions were intermixed though not formally randomized. Due to breakdowns and harvest conditions, the testing had to be somewhat opportunistic. To maintain higher harvest rates, the proportional tests and on-off tests were grouped together; this prevented having to frequently, manually, close and open bypass valves in the hydraulic circuit. The testing was completed on October 28th, 29th, and 31st, 2011.

Each data signal was collected at 100hz using the MAB and a Capture block in ControlDesk. The control model still only ran on a 5hz interrupt so any signals from within the model still show a step-wise response. A digital picture of the final cart profile was also taken; it was predicted that using a higher number of fill zones, only possible with proportional control, would result in smoother profiles.

The time of day and lighting conditions were not controlled, because testing was completed as soon as all equipment was functional. Testing practices were only modified if the current lighting conditions prevented standard SmartUnload functionality. Lighting and problems with serial connectivity between the ToughBook and MAB intermittently caused the system to disengage before the current repetition was complete; in those cases the repetition would be finished and two data files and video logs were recorded.

The biggest hindrance to completing the test repetitions as desired was a changing cart length (in units of columns in the fill grid) provided by the stereo software. The current implementation of Fill Strategy (Jennett, 2012) would take the number of columns in the fillable area of the cart and divide by the desired number of fill zones; any remainder would be added to the rear of the cart as unfillable area. To fill the entire cart, it is desirable to have this remainder be zero; however, it was not possible to select the corresponding desired number of fill zones, because the cart length provided by the stereo system changed dynamically during unloading events. As the desired number of fill zones is increased, the quantity and frequency of remainder columns also increased. Reduced fillable area negated any increased fill caused by more discrete actuations. The changing number of columns in the cart restricted the maximum desired number of fill zones to six; as the cart length changed between repetitions, the number of zones selected for the 'high' treatments was altered to minimize the amount of remainder columns.

62

As many parameters as possible were held constant during testing. Throughout testing the edge dead-band was 99cm, the desired fill level was 95%, the FS-threshold-offset was 100%, and the Kinze cart was used. During all of the proportional control tests the p-gain was 13 and the swing dead-band was 12cm. During all of the on-off control tests the swing dead-band was 50cm. (the Appendix contains detailed descriptions of these parameters)

Several metrics were calculated from the data and used to compare test treatments and quantify actuation effort. Plots were also generated for each repetition to observe disturbances and validate data. Statistics and associated plots were generated on a treatment or factor-level basis. The results from those analyses are provided below.

8.3 Results

A strong trend was not present in fill weight as a function of either control type or the number of fill zones alone. Table 12 and Figure 54 represent this relationship. The reason neither of the main test factors had an impact was primarily the existence of remainder columns and inconsistent stereo data.

Table 12: Control Type and Fill Zone Effect on Fill Weight

Control Type	Number of Zones	Fill Average [lbs]	Fill Standard Deviation [lbs]
Proportional	2	50652	1298
Proportional	3	50070	911
Proportional	high	51358	982
On-off	2	52474	2108
On-off	3	49220	368
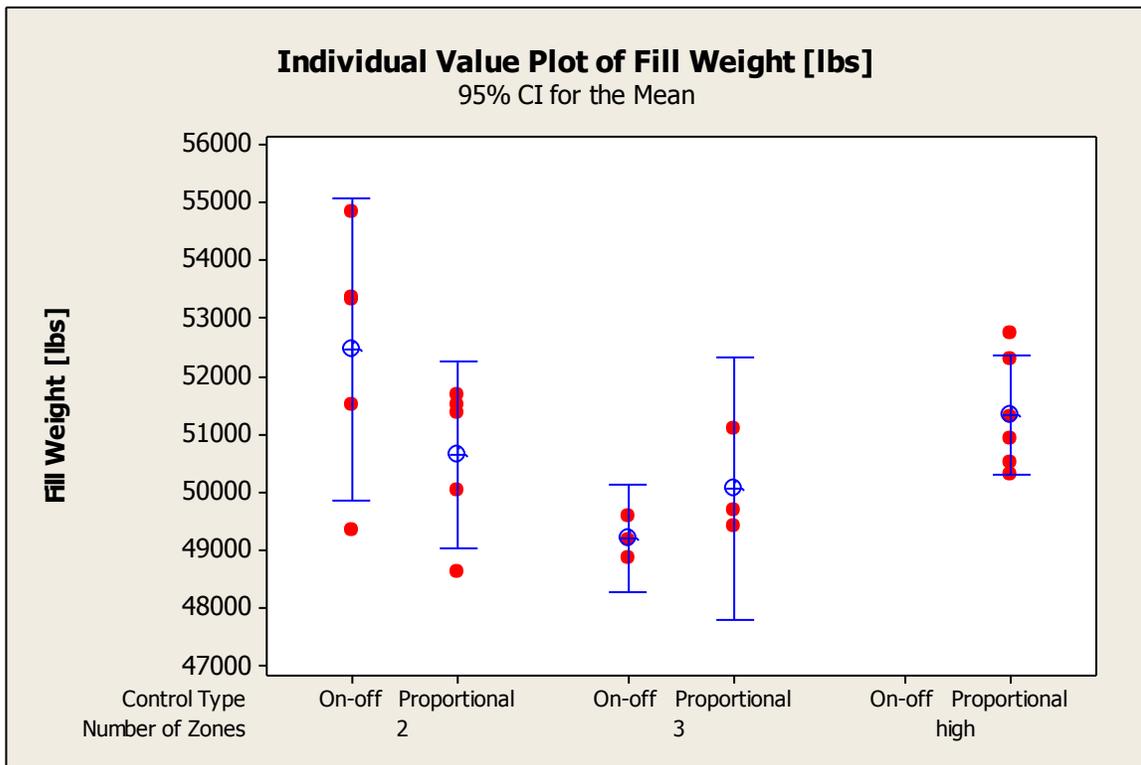
Figure 54: Control Type and Fill Zone Effect on Fill Weight

To reinforce the remainder column and stereo data problems, the number of columns and remainder columns were plotted for two different test repetitions. The first repetition (Figure 55) only required two fill zones so the highest remainder possible was one column; in addition, the number of columns given in the stereo data was relatively consistent. The second repetition (Figure 56) required five fill zones and the stereo data was less consistent; during the run, the number of remainder columns fluctuated from 0 to 3 and the number of columns started at 24 and was as low as 0. For any given test, the maximum number of remainder columns is one less than the number of fill zones; therefore, the risk of a large unfillable area increases as the desired number of fill zones increases.
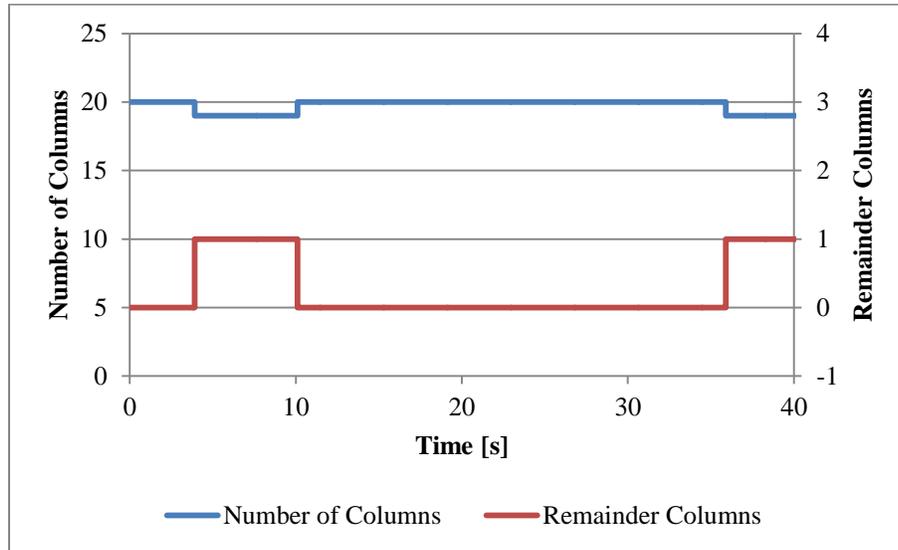
Figure 55: Remainder columns resulting from two fill zones and consistent stereo data.
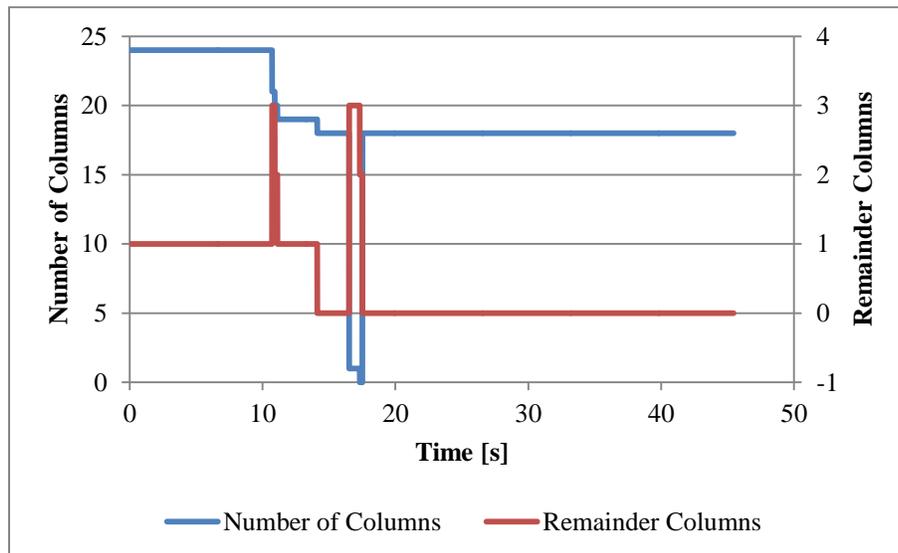
Figure 56: Remainder columns resulting from five fill zones and inconsistent stereo data.

Due to the challenges represented by Figures 55 and 56, it was necessary to look at the digital pictures taken after successful proportional or on-off tests to determine the impact of control type and number of fill zones on fill profile. Figures 57 and 58 show the fill profile contrast between two fill zones and five fill zones. The repetition with two fill zones shows two discrete piles of grain and a similar fill at the rear of the cart as at the front; there was only one remainder column, which caused the slight offset of the rear peak. The repetition requiring five fill zones is smoother and fuller throughout most of the cart; however, the higher number of fill zones resulted in a greater number of remainder columns

direction of light entering the cameras and the amount of cloud cover also play a role. The impact of lighting on the harvest data could not be quantified using the collected data. In the future, a direct measurement of light intensity near the camera lenses should be taken to better monitor this relationship throughout a harvest day.
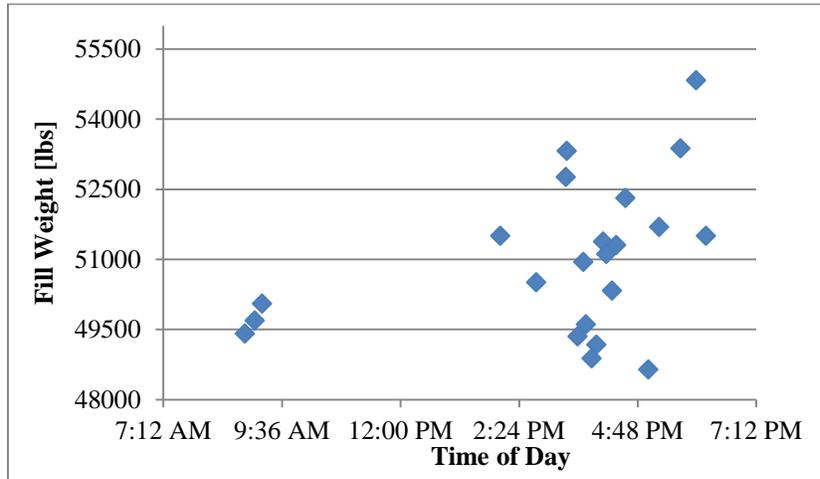
Figure 59: Fill Weight by Time-of-Day

The data was collected over three days. To rule out possible shift in system performance over time as a factor in these results, the average cart weights were tabulated by day (Table 13) and the individual cart weights were plotted (Figure 60). The averages did vary despite the target fill level being the same, but the confidence intervals greatly overlap; there was no statistical difference.

Table 13: Fill Weight by Date

Date	Fill Average [lbs]	Fill Standard Deviation [lbs]
Oct 28	51728	1531
Oct 29	50253	1551
Oct 31	49717	321
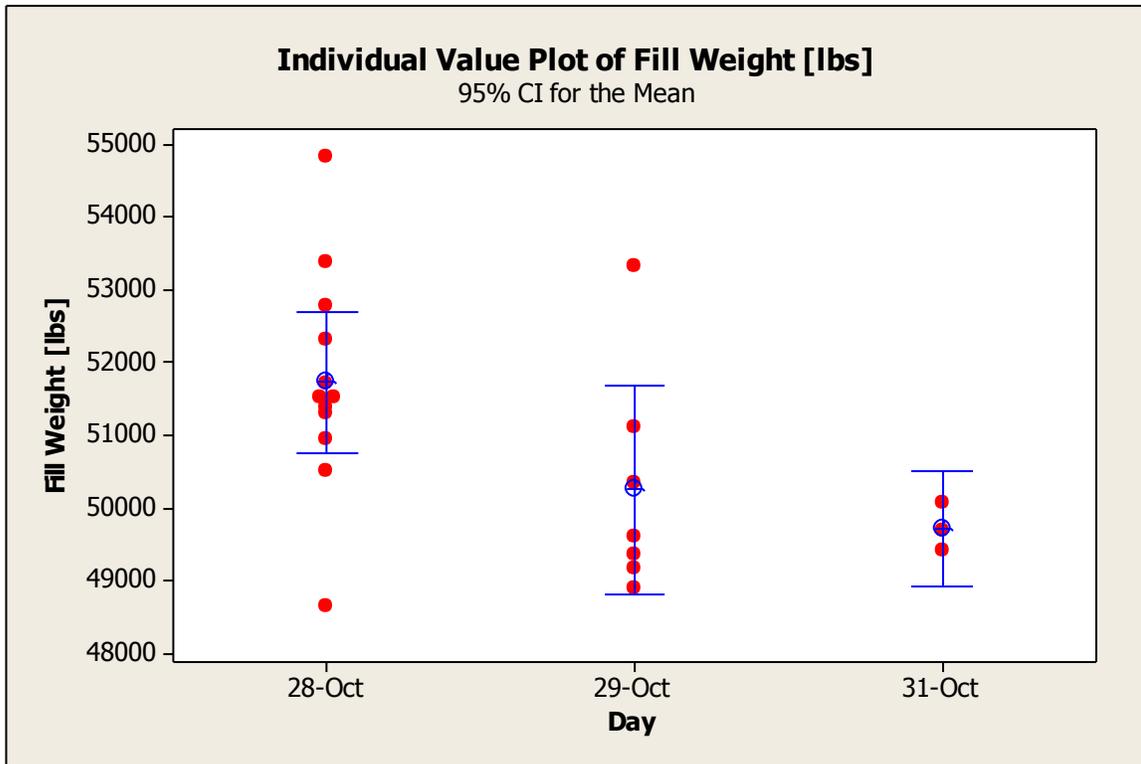
Figure 60: Fill Weight by Date

None of the factors tested seemed to have any considerable impact on cart weight; no obvious trends can be seen in the plots. The variation over all of the repetitions was also calculated. The 95% CI of the mean was only $\pm$14bu (Table 14). This general conclusion coincides with the fill modeling research completed here at Iowa State University (Jennett, 2012); this work showed that only the desired fill level and the offset of the fill zones towards the front and rear cart edges had a significant impact on final cart weight. Those variables were not factors for the swing control testing, so it is not surprising that little variation is seen in the data.

Table 14: Overall Fill Weight Variation

Metric	[lbs]	[bu] (corn)
Average	50870	908
Standard Deviation	1660	30
95% Confidence Interval	800	14
Maximum Weight- Minimum Weight	6190	111

The actuation effort results were much more conclusive (Tables 15 and 16). In Table 14, 'Delta Count' represents the number of times the auger swung more than 0.5 degrees; alternatively, 'Swing Count' is the number of times that the command signal changed. 'Swing Count' is greater than 'Delta Count' because noise in the auger angle or in the tracking data causes the swing command to change for single model calls but no motion occurs. On average the auger only swung about seven times and engaged twice. The efficiency of control is considered the amount of time spent engaged while tracking the cart and was 91% on average. The average total angular displacement for an unload was 51.5 degrees and the average displacement per swing was 8.2 degrees. The time elapsed between auger engagement commands was 34 seconds and the time between auger swing commands was 9 seconds. These results are very useful for the project for comparing the actuator duty-cycles during SmartUnload operation to those during manual operation.

Table 15: Actuation Effort Results

Metric	Track Time [sec]	Engagement Time [sec]	Swing Count	Delta Count	Engagement Count
Average	54.73	49.48	11.6	6.7	2.1
Standard Deviation	16.43	14.08	4.1	2.6	1.6
95% Confidence Interval	7.9	6.8	2.0	1.3	0.8

Table 16: Actuation Effort Results Cont.

Metric	Efficiency [%]	Total Angular Displacement [deg]	Average Swing Displacement [deg]	Time Elapsed per Engagement [sec]	Time Elapsed per Swing [sec]
Average	91.1	51.5	8.2	34.47	8.94
Standard Deviation	8.7	21.1	3.3	22.41	2.96
95% Confidence Interval	4.2	10.2	1.6	10.8	1.4

8.4 Conclusion

Due to performance inconsistency of the stereo system and the Fill Strategy used during the control testing, no final quantitative conclusion can be made about the possible

benefit of filling the cart in more discrete zones by using proportional swing control. Future testing should be done with a more robust version of the stereo system and lighting should be closely monitored during the tests. To fully realize the potential of proportional control, a Fill Strategy should be developed that does not put remainder columns in the rear unfillable area. The uncontrolled factors did not seem to have a major impact but the combination of disturbances makes trend recognition very difficult. With the high number of disturbances and possible interdependencies, the use of dynamic testing to compare performance based on small differences in controlled experimental factors is not recommended.

Dynamic testing is very useful for quantifying overall system performance. In this case, useful data was collected and analyzed pertaining to actuation effort during SmartUnload with swing-control. The current model implementation is over 90% efficient and results in an engagement cycle every 34 seconds and a swing actuation every 9 seconds. These metrics can be used by the project sponsor, John Deere, to compare SmartUnload performance to manual unloading performance. Additional metrics should be included going forward that allow for the quantification of operator stress and distraction when using SmartUnload compared to manual unloading. Also, these tests should be done at a high number of repetitions with uncontrolled starting cart weights to get an understanding of how often top-off unloads really occur and the impact of empty carts on actuation effort.

The comments of expert operators during operation of the system are invaluable and resulted in some of the most critical improvements to operation. One key feature for proportional swing control moving forward will be the shift to maximum valve duty-cycle when swinging to prevent a spill or when the boot leaves the fillable area of the cart.

Chapter 9 Conclusions

There is a drive in recent years to advance automation in agriculture. The conventional combine unloading process is challenging. Automating the combine unloading process has the potential to increase productivity and reduce operator stress. To automate the unloading process, an actuation system must be developed to effectively place grain at the correct location in a grain cart; an automated auger swing system has been successfully used to meet this need. A hydraulic circuit was designed to selectively include proportional valves; the circuit was successfully used to apply proportional control to auger swing. The main objectives of this work were to design, test, and evaluate on-off and proportional swing control.

To support the initial controls development, the response of the initial swing system was investigated. It was found that the angular velocity of the auger ranges from 8-12 deg/s, depending on duration and direction of swing and other (uncontrolled) variables. Over a full swing-in or swing-out the angular velocity is nearly constant and within the 9.0-10.1 deg/s range; in other words, the angular displacement profile is linear. There is only 0.2 degrees of overshoot in the swing-out direction and no overshoot in the swing-in direction; the overshoot was not large enough to impact the control strategy. For a typical auger length, 0.2 degrees of overshoot is only 1 inch of movement at the auger boot. It was not expected that angular velocity would be so consistent throughout the range of motion due to the kinematics and dynamics of the system. The response consistency is very beneficial for controls design. The same control function (for proportional or on-off respectively) was able to be used for all swing directions and durations. The control dead-bands were selected to prevent instability (oscillations around the set point).

An orifice is threaded into each port of the hydraulic cylinder to limit the swing rate. Due to initial control challenges with on-off control and the standard orifice, testing with smaller orifices was completed. It was expected that using smaller orifices would allow for tighter control dead-bands and make precise boot placement more feasible. Reducing orifice size did reduce the angular velocity of the auger as expected and did not impact overshoot or the start-swing and stop-swing lag times. With reduced angular velocity and consistent lag

times, control dead-bands can be reduced to accomplish precise boot placement. Reducing the angular velocity of the auger limits the system's ability to prevent grain spillage which is a problem with using smaller orifices. For that reason, proportional control was pursued; with proportional control the auger can be swung fast during spill prevention or swung slowly for accurate grain placement.

Swing and engagement latencies were quantified to determine their impact on control. On average, it takes 0.18 seconds to start a swing-in and 0.23 seconds to start a swing-out. It takes 0.38 seconds to stop a swing-in and 0.54 seconds to stop a swing-out. Auger engagement requires 2 seconds and disengagement requires 1 second on average. The lags associated with starting a swing are not detrimental to the unload system, because the amount of grain that would fall in that time period is only approximately 1 bu; that amount of grain would not increase the risk of grain spillage. Control dead-bands (precision) are a function of angular velocity and the swing stop lags; for a given angular velocity, dead-band size is proportional to the stop lag. The engagement lag is noticeable as an operator; operators may incorrectly associate the engagement latency with the SmartUnload system. The disengagement lag is very detrimental to the system; to prevent grain spillage, large areas at the front and rear of the grain cart must be designated as unfillable so that the auger will disengage before it has left the cart (for reasonable relative velocities [~2mph]).

Static proportional control testing was done to determine the impact of several factors on proportional swing response and to compare it to on-off control. When using proportional control, the dead-band around the desired location was less than 25% of that used during on-off control (for the same orifice size); with proportional control, the dead-band was not necessary for stability (the low valve duty-cycle prevented any noticeable swing) but prevented frequent valve cycling (due to noise in the auger angle sensor signal). The reduced control dead-bands are synonymous with improved grain placement precision and the ability to make smaller actuations without oscillation. The motivation for increased precision is primarily the potential for smoother fill profiles. Table 17 contains a final precision comparison between on-off control (for the three orifices tested) and proportional control (only the factory-sized orifice); the angular dead-bands should apply to most combines, and, assuming a 7.9m auger length, the associated linear values were calculated.

Table 17: Boot Placement Precision Comparison

Control Type	Orifice Size [mm]	Angular Precision [deg]	Linear Precision [cm]
On-Off	0.711	±1.7	±23
On-Off	0.94	±2.8	±41
On-Off	1.181	±4.3	±59
Proportional	1.181	±1.0	±14

During static testing, swing cylinder port pressure data was also collected and a comparison made between control types. At the inlet port, the pressure decreases as the desired angle is approached when using proportional control but stays constant during on-off control. At the desired angle, the outlet pressure stays nearly 0psi with proportional control but steps to 1100-2200psi during on-off control. The less rapid pressure changes seen with proportional control could prevent component wear and premature failure; the increased duty-cycle of the swing components did cause part failure during the fall, and steps should be taken to prevent future issues.

Harvest field testing results were used to evaluate the benefit of higher precision grain placement (only possible with proportional swing control) on fill weight and profile. In addition, SmartUnload system efficiency and actuation effort were quantified. The fill weight data did not show any response to precision grain placement. The combination of noise in the stereo data and a fill strategy tuned for larger movements prevented the full benefit of proportional control from being captured. Pictures were taken of each cart profile and it was observed that proportional control resulted in smoother, more even, fills while on-off control resulted in two or three discrete piles of grain. The SmartUnload system was 91% efficient during field testing; efficiency was defined as the percentage of time the auger was engaged during the unloading event. The total average angular displacement for an unload was 51.5 degrees and the average displacement per swing was 8.2 degrees. The average time between commands was 9 seconds for swing and 34 seconds for engagement. These numbers do apply specifically to this particular combine when filling the cart from 80-95%; however, the mechanics of the unloading system have not seen major design changes in recent years, so the general conclusions should apply to newer machines.

Several recommendations for future work can be made based on the results provided in this thesis. Currently the precision of auger angle is 0.43 degrees per bit (on average over

the range of motion); at the tip of the boot, this corresponds to more than 2 inches per bit. A displacement of 2 inches does not seem like a lot but is possible with proportional control; to reduce dead-bands and still prevent valve cycling, the resolution of auger angle should be increased. To reduce the unfillable area required as a buffer for spill prevention, the speed of the auger engagement/disengagement actuator should be increased by 50-75%; doing so will result in better cart utilization and customer acceptance. To maximize the potential of proportional control, gain scheduling should be implemented with production intent hardware and larger orifices to allow for even higher maximum velocities during spill prevention or large shifts in desired fill location without loss of stability or precision. While using proportional control, the maximum valve duty-cycle should be used for spill prevention. The two swing directions have slightly different responses during proportional control, and a separate control function for each would result in the most consistent performance. To see the benefit of proportional control on fill, a fill strategy should be developed to utilize smaller actuations and evenly spread the number of fill zones across the entire cart. Lighting had a noticeable impact on stereo data this harvest, and further testing should be completed while recording data from ambient-light sensors to quantify that relationship. There is a lack of performance data for manual unloads (completed by expert operators) to compare with the SmartUnload system. Logistics data should also be collected to determine the frequency of top-off unloading events for a range of operations to determine the market for an automated unloading system.

Bibliography

ADACONN INSERTA. (2002). *SAE Straight Thread Port Dimensions*. Retrieved April 1, 2011, from inserta.com: http://www.inserta.com/PDF/AI-STR-THD-PORT-DIM.pdf

Batte, M. T., Forster, D. L., Surjandari, I., Hudson, W. E., & Rodriguez-Solis, J. (1999). *1999 Ohio Precision Agriculture Survey.* The Ohio State University. AEDE.

Batte, M. T., Pohlman, C., Forster, D. L., & Sohngen, B. (2003). *Adoption and Use of Precision Farming Technologies: Results of a 2003 Survey of Ohio Farmers.* The Ohio State University. AEDE.

Brand Hydraulics. (2011). *Brand Hydraulics Product Catalogue*. Retrieved from brand-hyd.com: http://www.brand-hyd.com/efc/efc-info.htm

Claas. (2011). *Forage Harvesters*. Retrieved November 21, 2011, from Claas: http://www.claas.com/cl-pw/en/products/forage_harvester/jaguar_980-930/video_animation/autofill_trailer/start,lang=en_EU.html

Claas. (2011). *Press releases*. Retrieved November 21, 2011, from claas.com: http://claas.com/cl-gr/en/press/mitteilungen/start,cid=565120,lang=en_UK.html

Cundiff, J. S. (2002). *Fluid Power Circuits and Controls: Fundamentals and Applications.* Boca Raton, Florida, United States: CRC Press LLC.

Darr, M. (2011, August). AE 410X/510X Electronic Systems Integration for Agricultural Machinery Production Systems. Ames, Iowa, United States: Iowa State University.

Darr, M. (2011, November 5). TSM 333: Automation Economics. Ames, IA, United States: Iowa State University.

Deere & Company. (2011). *Technology In Our Products*. Retrieved November 21, 2011, from deere.com: http://www.deere.com/en_IN/home_page/ag_home/products/technology/technology.html

Diekmann, F., & Batte, M. T. (2010). *2010 Ohio Farming Practices Survey: Adoption and Use of Precision Farming Technology in Ohio.* The Ohio State University. AEDE.

Fiat Industrial. (2011). *Combine with V2V*. Retrieved November 21, 2011, from
   fiatindustrial.com: http://www.fiatindustrial.com/en-
   US/innovation/cnh/Pages/video_case_ih.aspx

FittingsandAdapters.com. (2011). *SAE J1453 O-Ring Face Seal*. Retrieved April 1, 2011,
   from FittingsandAdapters.com: http://fittingsandadapters.com/saej1orfacse.html

Foster, C. A., Morselli, R., Vanhercke, O., Wang, G., Missotten, B., Paquet, B. J., et al.
   (2010, March 24). *Patent No. EP2165586.* Italy.

Google. (2011, November 1). *public data explorer beta*. Retrieved November 21, 2011, from
   Google:
   http://www.google.com/publicdata/explore?ds=d5bncppjof8f9_&met_y=sp_pop_totl
   &tdim=true&dl=en&hl=en&q=trends+in+world+population

Happich, G., & Lang, T. (2009). *Model-based loading of agricultural trailers.*
   Braunschweig, Germany: Technical University of Braunschweig, Institute of
   Agricultural Machinery and Fluid Power.

Hoff, S. J. (2011, August). Transducer Based Instrumentation. Ames, Iowa, United States:
   Agricultural and Biosystems Engineering Department, Iowa State University.

Huster, J., & Diekhans, N. (2007). An Assistance System for the Spout Control on Forage
   Harvesters. *VDI Berichte*, 225-230.

International Organization for Standardization. (2007, June 15). ISO 11783. *First*.
   Switzerland: ISO.

Jaybridge Robotics. (2011, August 2). *News & Events*. Retrieved November 21, 2011, from
   Jaybridge Robotics: http://www.jaybridge.com/news/story/?p=116

Jennett, A. T. (2012). *Decision Support System for Sensor-based Autonomous Filling of
   Grain Containers.* Ames: Iowa State University.

Kinze Manufacturing. (2011, August 4). *News Archive*. Retrieved November 21, 2011, from
   kinze.com: http://www.kinze.com/news/viewNewsArticle.html?id=36

Kurt Hydraulics. (2011). *Pipe Threads*. Retrieved March 30, 2011, from kurthydraulics.com:
   http://www.kurthydraulics.com/threads.php

Madsen, T. E., Kirk, K., & Blas, M. R. (2009). 3D Camera for Forager Automation. *VDI
   Berichte*, 147-152.

MathWorks. (2011). *Product Documentation*. Retrieved May 15, 2010, from
    mathworks.com: http://www.mathworks.com/help/?s_cid=global_nav

MathWorks. (2011). *Stateflow-Documentation*. Retrieved May 15, 2010, from
    mathworks.com: http://www.mathworks.com/help/toolbox/stateflow/

McMaster-CARR. (2011). *Quick-Disconnect Hose Couplings*. Retrieved March 30, 2011,
    from mcmaster.com: http://www.mcmaster.com/#quick-disconnect-hose-
    couplings/=f1np31

Microchip. (2009). *dsPIC30F5013*. Retrieved 2010, from microchip.com:
    http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010347

molex. (2011). *Molex Connectors*. Retrieved from molex.com:
    http://www.molex.com/molex/index.jsp

Moller, J. (2010). *Computer vision - A versatile technology in automation of agriculture
    machinery.* Bologna: EIMA International.

MoTec Mobile Camera Solutions. (2011). *MC3000A*. Retrieved November 26, 2011, from
    motecgmbh.de: http://www.motecgmbh.de/index.php?id=mc3000a&L=1

New Holland. (2011). *Forage New Holland FR9000*. Retrieved November 21, 2011, from
    agriculture.newholland.com:
    http://agriculture.newholland.com/uk/en/Products/Forage/FR9000/Pages/Innovationti
    onsAwards_details.aspx#prod_details

Ogata, K. (2004). *System Dynamics* (Fourth ed.). Upper Saddle River, New Jersey, United
    States: Pearson Prentice Hall.

Oklahoma State University. (2011). *World Wheat, Corn & Rice*. Retrieved October 25, 2011,
    from NUEweb: http://nue.okstate.edu/crop_information/world_wheat_production.htm

Palm, W. J. (2009). *System Dynmaics* (2nd ed.). McGraw Hill.

Pype, W., & Posselius, J. (2011). Autonomous Harvesters: The Demeter Story. *Resource*, 24.

Smith III, J. O. (1985). *Introduction to Digital Filter Theory.* Stanford, Department of Music.
    Center for Computer Research in Music and Acoustics.

Society of Automotive Engineers. (1998, October). J1939. Warrendale, Pennsylvania, United
    States: SAE.

Steward, B. (2011, August). AE 403/503 Modeling and Controls for Agricultural Systems.
    Ames, Iowa, United States: Iowa State University.

Valley Hydraulic Service, Inc. (2011). *Port and Fitting Size Chart*. Retrieved November 22, 2011, from valleyhydraulic.com: http://www.valleyhydraulic.com/Fluid_Ports.html

Valley Hydraulics Service, Inc. (2011). *37degree Hydraulic Fittings (JIC)*. Retrieved April 1, 2011, from valleyhydraulic.com: http://www.valleyhydraulic.com/JIC_Fittings.html

Weltzien, C., Graefe, F., Bonig, I., & (Claas) Diekhans, N. (2004). Automated Overloading with Selfpropelled Forage Harvesters. *VDI-Berichte*(1855), 123-130.

Appendix

Auger Angle Noise Analysis

The auger angle signal was observed to be very noisy. Upon analysis, the noise was determined to be about 40 Hz frequency and ±0.015 Vdc. To rule out vibration as a possible cause, the signal was captured at four different engine speeds; little or no change in noise frequency or magnitude occurred between engine speed treatments (compare the following four figures) [although aliasing is possible if the true signal frequency was greater than 50 Hz].
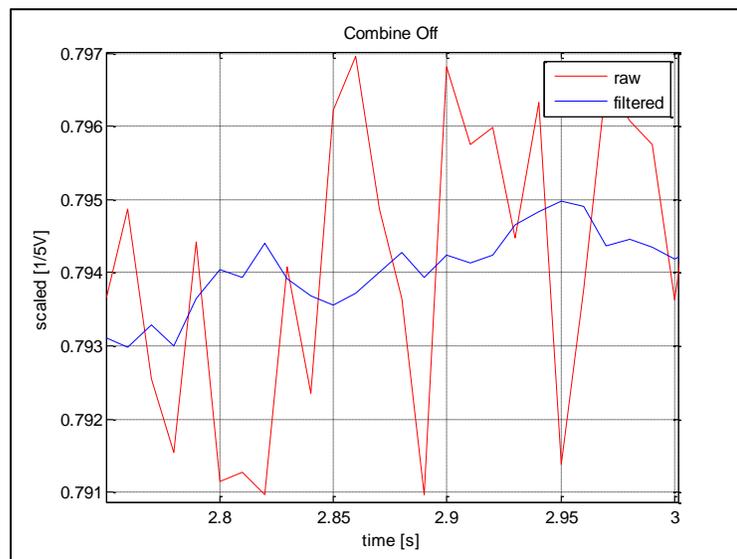
0.25s sample with the engine speed at 0 rpm
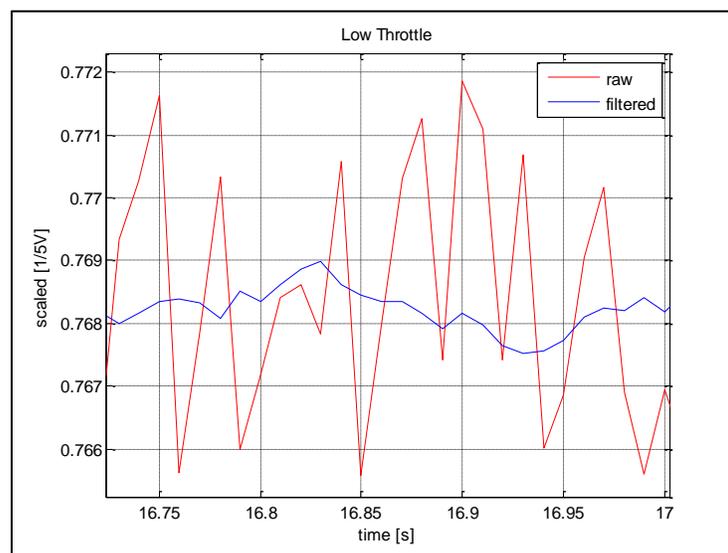
0.25s sample with the engine at low throttle
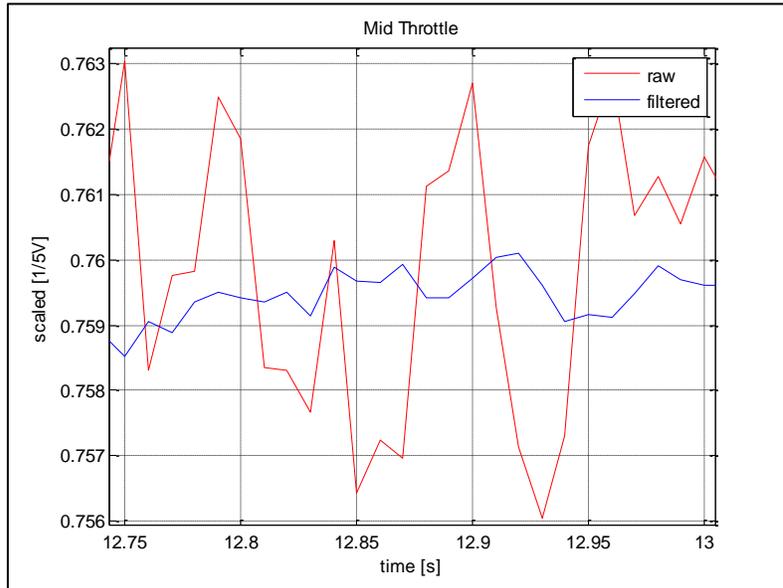
0.25s sample with engine at mid throttle
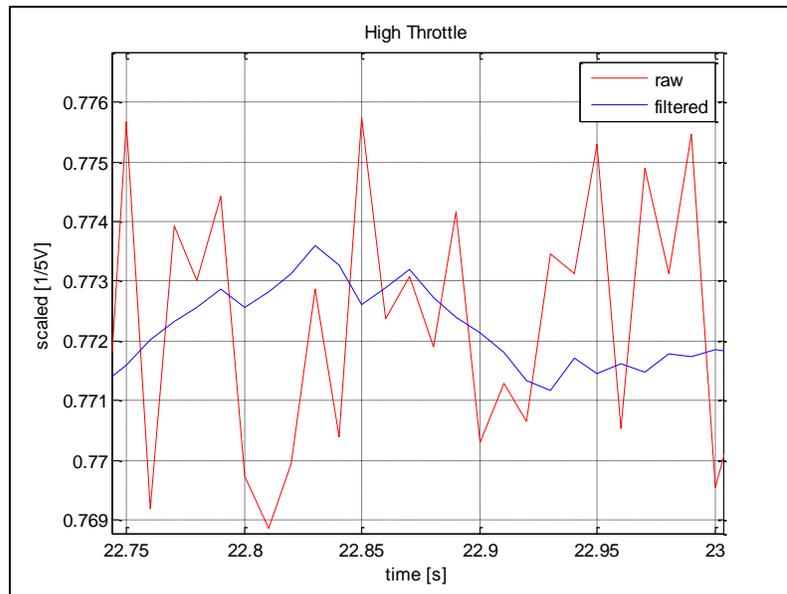
0.25s sample with the engine at high throttle

The auger angle signal was also captured through unshielded and shielded wire simultaneously. No difference in noise frequency or magnitude resulted from using shielded wire (shown in the next figure), suggesting external EM is not the cause of the noise.
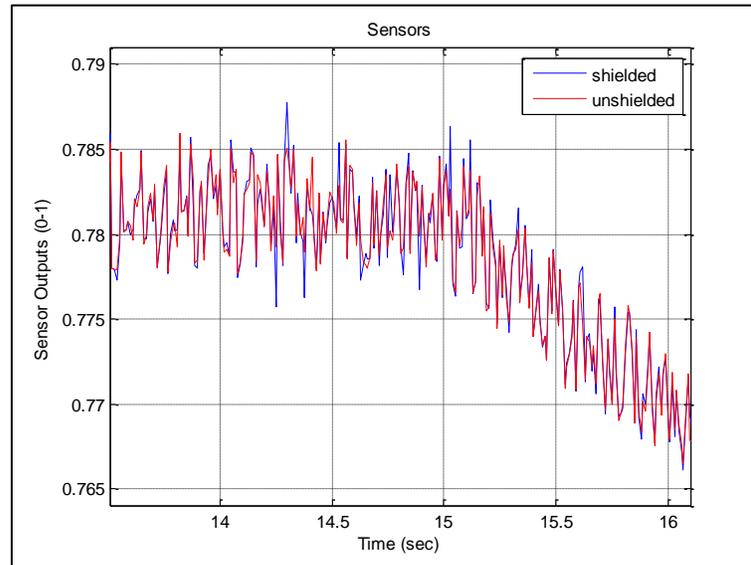
With vibration and external EM eliminated as possible causes, the only remaining possibility being investigated is sensor-induced noise (though this seems to be the least likely).

A "single-pole" filter was implemented in software. The filter runs at 100hz. The filter output was much cleaner than the input. The filtered signal is shown in the four engine-speed plots. Further testing was completed to determine the effect of "smoothing factor" on the quality of the output signal; four treatments of smoothing factor were tested (0.1, 0.15, 0.2, and 0.5) (shown in the following four figures). As smoothing factor increases, the signal-noise reduction decreases. There is a known trade-off between noise reduction and induced lag associated with selecting a smoothing factor; this is also important to consider when selecting a smoothing factor.

$$Y = smoothing_{factor} \times X + (1 - smoothing_{factor}) \times Y_{-1}$$
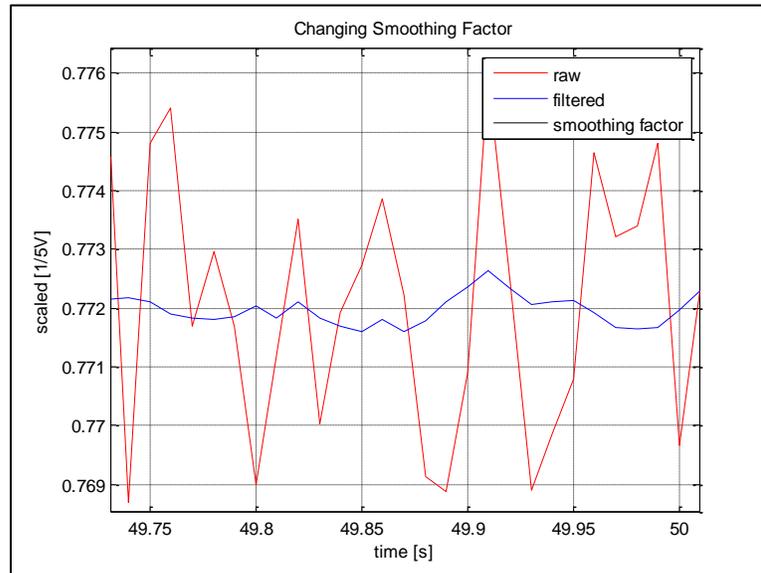
Smoothing factor at 0.1
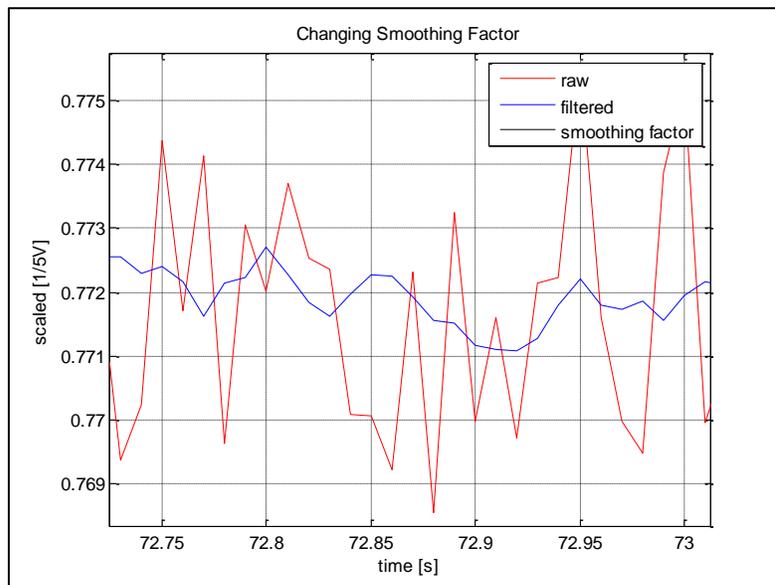
Smoothing factor at 0.15
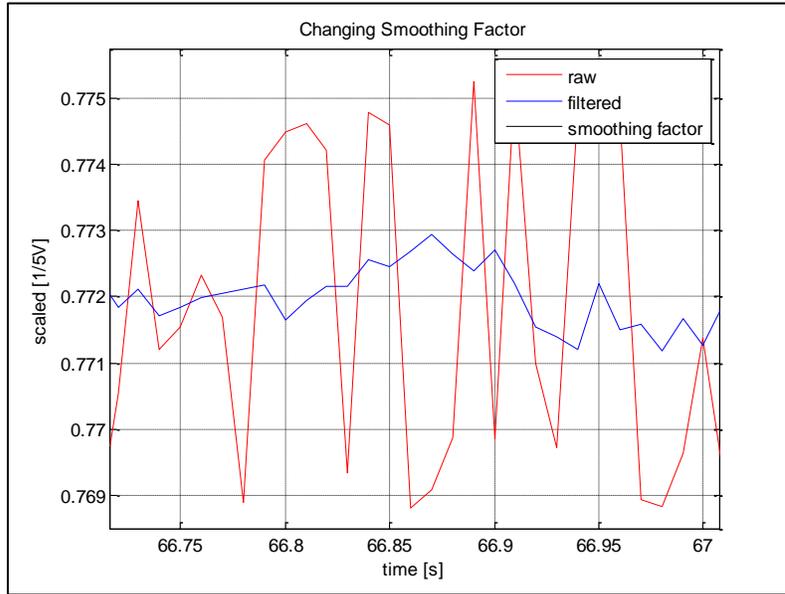
Smoothing factor at 0.2
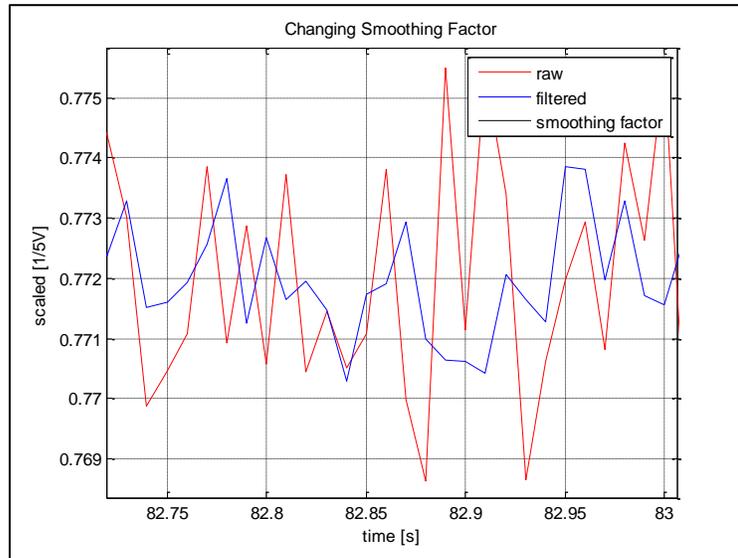
Smoothing factor at 0.5

CAN Messages

In general, a CAN message consists of an identifier (ID) and eight, eight-bit data bytes. The CAN standard uses an extended (29-bit) identifier. An identifier includes a priority, parameter group number (PGN), source address (SA), and destination address (DA). The data bytes contain any signals the message is packed with. Signals can be any length; a single data byte may contain multiple signals or one signal may require multiple data bytes. The first one or two data bytes may be used to further define the message beyond the information contained in the ID; in that case, those data bytes are called command bytes. For any yes/no (on/off, engaged/disengaged, etc) signals, John Deere uses two bits. A value of '0' means 'off', a value of '1' means 'on', a value of '2' means 'unknown', and a value of '3' is 'reserved'. Messages can be proprietary or standardized depending on their purpose and the manufacturer's desire to keep system functionality secret. In SAE J1939, a range of PGNs are reserved just for proprietary messages.

Several CAN messages are used in the SmartUnload system: Smart Unloading Combine Command, Smart Unloading Status, SUC Address Claim, Cab Controller, Combine VI, Grain Tank Unloading System, Hydro-Handle Status, Unloading Auger Drive, Diagnostics Matrix, and the Diagnostic Addresses.
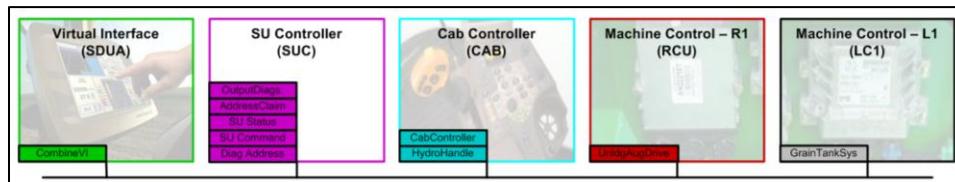
The Smart Unloading Combine Command message contains ten signals and its main purpose is to request actuations from the machine controller. The first signal is the command byte. The command byte is sixteen bits long and further defines the content of the message (past the message ID). The other signals in this message are all two bits long except for the last signal which is eight bits long. The second signal denotes if the unloading lights are being requested to be on or off. The third signal is the on/off command for the row-finder light (light mounted into the side of the grain tank directly above and slightly forward from where the tracking camera is mounted). The fourth signal tells the machine controller whether or not it should be responding to the message. The fifth signal is the request to

engage or disengage the auger. The sixth and seventh signals are the auto swing-out and swing-in commands, which are placeholders for future functionality (these signals are not currently used). Signals eight and nine are the auger fold and unfold requests; these are also placeholders and would only be used on machines that have folding augers. Signal ten indicates the direction that the auger should swing (251=swing in, 252=swing out, and 253=don't swing).

```
SA: SUC

Byte 1-2:  CMDbytes
Byte 3:      Bit 1-2:  CAN_UnldgLightReq
             Bit 3-4:  CAN_RowFindersLeftReq
             Bit 5-6:  CAN_MasterUnldngSysCtlrState
             Bit 7-8:  CAN_AugersEngageMode
Byte 4:      Bit 1-2:  CAN_UnloadingAugerSwingOutAuto
             Bit 3-4:  CAN_UnloadingAugerSwingInAuto
             Bit 5-6:  CAN_UnloadingAugerFoldSmUnl
             Bit 7-8:  CAN_UnloadingAugerUnfoldSmUnl
Byte 5:  CAN_AugRotnCmdPcnt
```

The Smart Unloading Status message contains six signals and the information in the message is used to populate the VI objects with useful information. The first signal is the command byte and is eight bits long. The second signal indicates the engagement state of SmartUnload and is four bits long. The third signal indicates whether or not a cart is detected and is also four bits long. Signals four, five, and six are the grain cart fill level, combine relative position request, and the disengagement alarm (all eight bits long). The grain cart fill level is calculated by fill strategy and is a value between 0 and 110 (with 100 indicating the grain level is flush with the top of the cart). The combine relative position request signal is currently not in use but can be used to populate a VI indicator telling the combine operator to speed up or slow down. The disengagement alarm signal contains the most recent reason for the system's disengagement and is displayed on the diagnostics page of the VI.

```
SA: SUC

Byte 1:  CMDbyte
Byte 3:      Bit 1-4:  CAN_SmrtUnldSysState
             Bit 5-8:  CAN_SmrtUnldgCartDetnState
Byte 4:  CAN_GrainCartFillLvl
Byte 5:  CAN_CombRelPstnReq
Byte 6:  CAN_SmrtUnldgDisAlrm
```

The SUC Address Claim is sent when the vehicle is started or upon request from other controllers and establishes the SUC as a valid/approved controller on the CANbus. Address claims include signals that identify the manufacturers codes, the ECU instance, the function instance, the function name, the device class (ex: Harvest Equipment=7), the device class instance, the industry group, and a self-configurable name. More information on address claims can be found in the CAN standards ISO-11783 and J1939. ECUs on the John Deere CANbus will also send a TLA (three-letter-acronym) claim; this TLA is used to identify the ECU in the VI diagnostics tool.

```
SA:  SUC

Bit 1-21:  Name_IdentityNumber (0 or 1 likely)
Bit 22-32:  Name_MfgrCode (JD = 0x21 = 33)
Bit 33-35:  Name_ECUInstance (0)
Bit 36-40:  Name_FunctionInstance (0)
Bit 41-49:  Name_Function
                (Harvester Machine Control = 132)
Bit 50-51:  Name_DeviceClass
                (Harvest Equipment = 7)
Bit 57-60:  Name_DeviceClassInstance (0)
Bit 61-63:  Name_IndustryGroup
                (Ag & Forestry = 2)
Bit 64:  Name_SelfConfigurable (no = 0)
```

The Cab Controller message contains a command byte, the auger length [decimeters], and the SmartUnload engage command. The SmartUnload engage command comes from the operator when the resume switch is pressed.

```
SA:  CAB

Byte 1:  CMDbyte
Byte 3:  CAN_UnldgAugLength
Byte 6:      Bit 5-6:  CAN_SmrtUndgEngCmd
```

The Combine VI message comes from the VI (a John Deere GS2, GS3, or Command Center) and includes a command byte, the control mode, the desired cart fill level, the fill mode, and the system enable state. The control mode indicates which actuation method(s) should be used. The desired cart fill level is used within Fill Strategy to determine when the cart is full. The fill mode indicates how Fill Strategy should execute. The system enable state determines if the system can advance from state 2 to state 3 and is controlled by the operator through the VI.

```
SA: SDUA

Byte 1:  CMDbyte
Byte 4:  CAN_SmrtUnldgCtrlModeCmd
Byte 5:  CAN_GrainCartFillLvlSetpntMax
Byte 6:      Bit 1-4: CAN_SmrtUnldgFillModeCmd
             Bit 7-8: CAN_SmrtUnldgEnblState
```

The Grain Tank Unloading System message contains a command byte and the auger position signal. The auger position signal is a 0-250 value linearly scaled to the auger-angle sensor output voltage. Some machines that have a direct analog voltage coming into the IPM or MAB do not use this message; only machines that have an intermediate ECU to put the position value out on CAN need it.

```
SA: LC1

Byte 1:  CMDbyte
Byte 2:  CAN_UnloadingAugerPstn
```

The Hydro-Handle Status message contains the operator override signals which are populated when the operator presses the buttons on the hydro-handle. The operator can override the SmartUnload system (take the system out of state 4 to state 3) by pressing the buttons for auto-swing-out, auto-swing-in, swing out, swing in, auger engagement/disengagement, or auger fold/unfold.

```
SA: CAB

Byte 1: CMDbyte
Byte 5:      Bit 1-2: CAN_UnloadingAugerSwingInAuto
             Bit 3-4: CAN_UnloadingAugerSwingOutAuto
             Bit 5-6: CAN_UnloadingAugerSwingIn
             Bit 7-8: CAN_UnloadingAugerSwingOutButton
Byte 8:      Bit 1-2: CAN_UnloadingAugerUnfold
             Bit 3-4: CAN_UnloadingAugerFold
             Bit 5-6: CAN_AugersEngageMode
```

The Unloading Auger Drive message contains a command byte and a status signal for the auger rotation. Currently, this message is not used in the model though the signals are populated within the input structure.

```
SA: RCU

Byte 1:  CMDbyte
Byte 2:     Bit 1-2:  CAN_UnloadingAugerDrive
```

The Diagnostic Matrix is an extended-protocol message containing the signals for many internal model variables useful for debugging work. This message is sent selectively by changing the value of a diagnostic address; for normal operation, it is desirable not to output this message, because the combine CANbus is already operating above the recommended data-load.

```
ISUDiagnosticsMatrix

SA:  SUC

Byte 1-2:  CMDbytes
Byte 3-200: Data
```

The Diagnostic Address message(s) includes custom inputs that are mostly used for developmental, tune-able parameters. The messages are transmitted form the VI to the SUC.

Model Hierarchy

The model, developed in Simulink, contains all Smart Unloading logic and control functions. The model inputs are all packaged within the input structure, also called the input bus (not to be confused with the CANbus); similarly, outputs are also packaged within a structure. There are fifty-five input signals and eighteen output signals. Combining the signals into a structure greatly improves model readability.
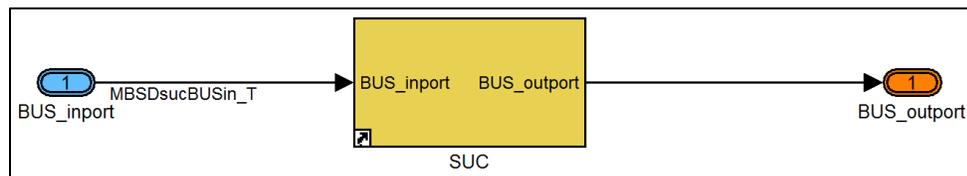
The model was compartmentalized into four main pieces: the State Machine, Fill Strategy, Control Implementation, and Speed Control. Other smaller subsystems include the Diagnostics and Error-Code subsystems. Andy Jennett was the lead developer of Fill Strategy and Alex Nykamp was the lead developer of Speed Control. The overall model integration and diagnostics were a joint effort. The purpose of the State Machine is to control the system engagement level in a safe and robust way. Fill Strategy processes the fill data and determines the desired fill location within the cart. Control Implementation actuates the auger swing and auger engagement to try and reach the desired location (from Fill Strategy) while preventing spill. Control Implementation also provides key information to the stereo system, Fill Strategy, and Speed Control: auger angle, current location within the cart, desired shift magnitude in centimeters, and other parameters. Speed Control utilizes velocity changes of the combine (Active Combine Speed Control) or cart tractor (Machine Sync) to control the position of the boot within the cart. The 'ErrorCodes' subsystem generates a single value that defines the reason for the most recent system disengagement and displays that on the virtual interface (VI). The Diagnostics subsystem outputs a matrix containing useful internal model variables from each main subsystem; that matrix can then be output on the CANbus, enabling efficient model debugging by the system developers.
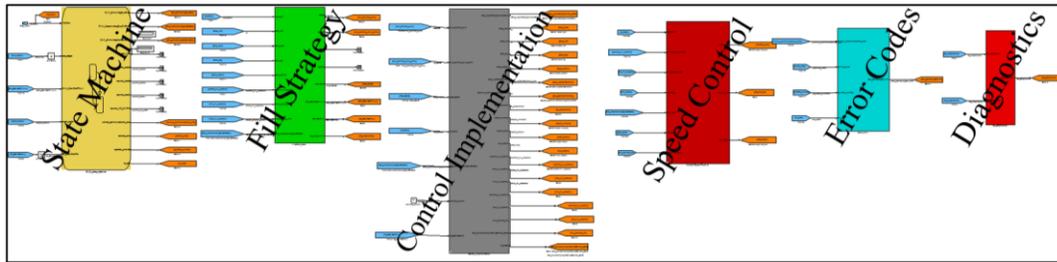
Simulink models can be saved as library blocks. Library blocks can be called by other models through a library link. Library links appear the same as the master but have a small white square with a black arrow in the lower-left corner; an example is the SUC. The benefit of configuring the main subsystems as library blocks is that then they can be developed on by multiple developers concurrently; for example, Andy Jennett could develop Fill Strategy while Alex Nykamp tested Speed Control in the SUC using the most recent Fill Strategy release. Also, library blocks can be called by multiple higher-level models; the SUC was called by simulation models, builds for the MAB, and builds for the IPM. John Deere calls this development strategy "hierarchical modeling."

Fill Strategy and the Fill Grid

Multiple references were made to FS (Jennett, 2012) and the stereo fill grid in the main thesis. This section is intended to provide a short description of each to facilitate understanding of their impact on CI and field testing.

The primary purpose of FS is to decide where grain should be placed within the cart. This decision is based on the range of locations possible given the current mode of actuation and the fill grid.

The fill grid is a two-dimensional matrix (16x64). Each element of the matrix contains a fill height relative to the top plane of the grain cart opening representative of a 25cm x 25cm area. Only part of the fill grid will contain valid values and corresponds to the cross-sectional area of the grain cart. The fill grid is generated by the stereo system. The next figure is a screenshot of the stereo system interface; the fill grid is represented by the grey-scale grid at the right-center (whiter is higher). (National Robotics Engineering Center, 2012)
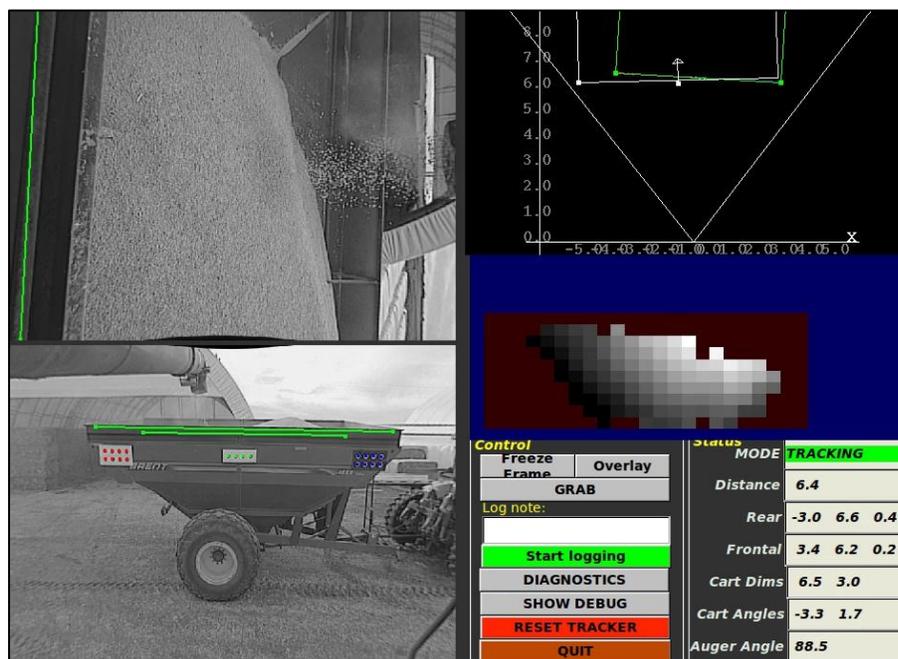
FS averages the fill level for each column of valid data and uses that to determine where in the cart is still empty. Several additional inputs will determine the operational procedure of FS: number of zones, edge dead-band, strategy, target fill level, and threshold

fill level. The number of fill zones designates how many discrete fill locations will be used within the fillable area of the cart. The edge dead-band determines the minimum amount of unfillable area required at the front and rear edges of the grain cart for spill prevention. The strategy determines the order in which the fill zones will be filled and includes front-to-back, back-to-front, or front-to-back-to-front. The target fill level will determine how full each zone needs to be before FS will request actuation to the next zone. The threshold fill level determines how full the cart needs to be before FS will actively request a specific fill zone location.

For harvest CI testing and validation, the only parameter that was varied was the desired number of fill zones; this minimized data skew resulting from FS execution. The target fill level was 95% and the edge dead-band was 4 columns (round up 99cm). The strategy was front-to-back, meaning that the front of the grain cart would be filled first. The threshold fill level was set at 0% (offset 100%), so that FS would actively request a specific fill zone at the start of SmartUnload execution; doing so improved fill consistency.

Modes of Actuation

Through Control Implementation and Speed Control, three main modes of control are available: auger swing, Active Combine Speed Control, and Machine Sync. There are tradeoffs in performance, cost, and reliability with each mode of control. Auger-swing control is the least expensive, has the fastest response, and can function without GPS availability or wireless communication; however, swinging the auger does increase wear on those components and the range of motion is limited. Active combine speed control provides a wide range of motion and also does not require GPS or wireless communication; however, changing combine speed while harvesting can be throughput-limited (the combine may not have power available to speed up or speeding up may cause the combine to plug). Machine Sync avoids concern of overloading the combine but requires RTK GPS, wireless communication, and a Machine-Sync-ready John Deer cart tractor; the reliability of Machine Sync is limited to the reliability of the RTK GPS signal and wireless communication, and Machine-Sync-ready tractors are currently power-limited (not well suited for carts over 1000bu).

Mode of Actuation	Combine Speed Change Required	GPS required	Wireless Communication Required	John Deere cart tractor with IVT
Swing	No	No	No	No
Active Combine Speed Control	Yes	No	No	No
Machine Sync	No	Yes	Yes	Yes

Smart Unloading Controller

The Smart Unloading Controller (SUC) is the ECU on which the SmartUnload control function runs. During the SmartUnload project, multiple ECUs have been used for this purpose including the MAB and IPM.

The entire control model was constructed in Simulink. The Real-Time Workshop tool within Simulink allows for C-code to be generated from the models. A specific hardware target can also be selected if known. For the IPM, the micro-processor executing the SmartUnload function is an Infineon XC2267 (16-bit); conveniently, the XC16x is selectable as a target in Simulink and used when generating code for the IPM. When executing SmartUnload on the MAB, the generic target is used but a specific target file is selected so that the build will be automatically configured appropriately.

To run on the IPM, a member of John Deere Embedded Systems Shared Services (E3S) will take the C-code generated out of Simulink and create a build within the John Deere Operating System (JDOS). The E3S engineer sets up all of the CAN messaging on the input and output side and also interfaces with the other signal types (camera LVDS, analog, etc.). Executing on the MAB is similar to executing on the IPM, but the message handling is done within a Simulink model using the appropriate MAB driver blocks (and assistance from E3S is not required).

When the SUC is connected to CAN, messages are received intermittently and the associated data is queued in the input structure. The stereo fill and tracking data will also get packed into the input structure. The model runs at a constant 5hz and will read the current values within the input structure each time. The model will pack the output structure. The signals from the output structure will be packed into the appropriate CAN messages or transmitted internally to the processor executing the image processing software.

Stateflow

Stateflow is a graphical tool for developing state charts within the Simulink programming environment. Once Simulink is open, a user can add a Stateflow Chart from the Stateflow menu in the Library Browser. Just click and drag the Chart symbol into the model.

The first step in designing a state chart is to determine the required inputs and outputs. These inputs and outputs will be represented by variables within Stateflow. Variables can be added two ways: through the menus in the Stateflow window or through the Model Explorer. The Stateflow window opens when the Chart symbol is double-clicked. To add a variable just click 'Add' on the menu bar and select 'Data' and 'Input from Simulink' or 'Output to Simulink'. The Model Explorer can be accessed from the model window menu bar or by clicking ctlr+h. From the Model Explorer select the Chart from the model tree and click the 'Add Data' symbol on the toolbar. This will open a menu on the right from which the new variable can be configured. Important configuration settings include name, scope, size, and data type. For beginners, adding variables through the Stateflow window can be faster and more intuitive; however, the Model Explorer is a powerful tool for organizing data flow throughout the entire model.
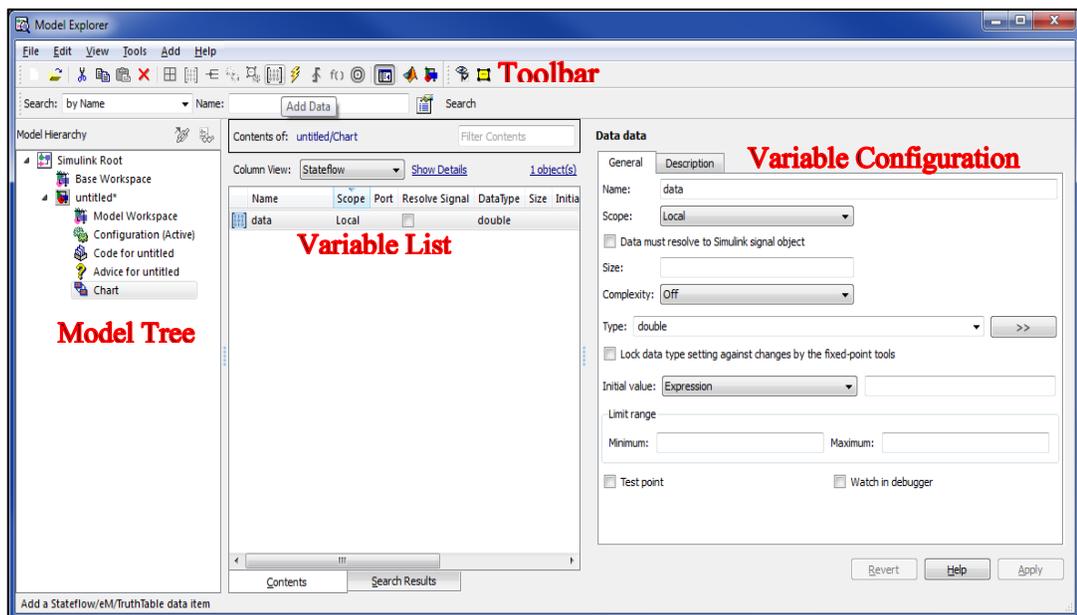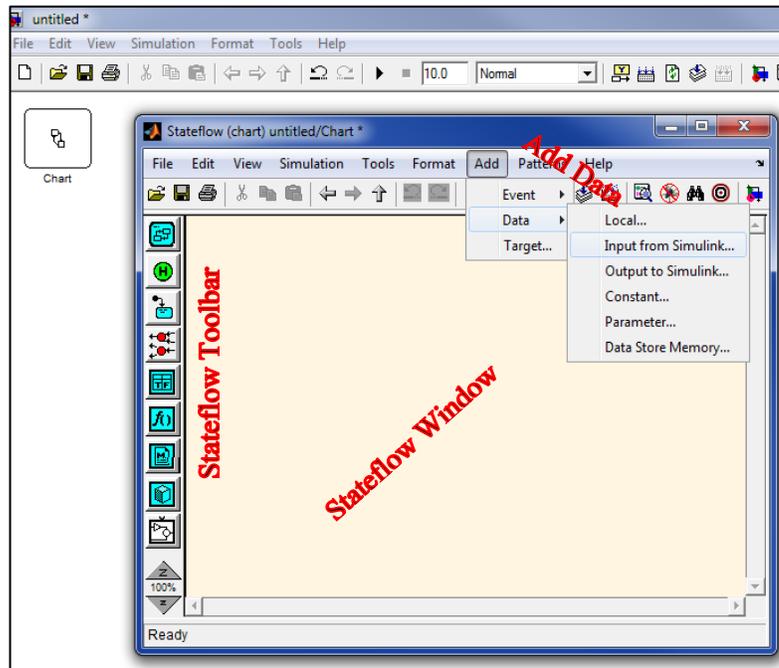
There are three main building blocks in Stateflow: states, nodes, and transitions. States and nodes can each be added by clicking the associated symbol on the left side of the Stateflow window and then clicking the location where the object should be placed.

States appear as boxes with rounded corners. States at each level of the state machine have a selectable decomposition. Decomposition is either exclusive (OR) or parallel (AND). Exclusive means that only one state within that level can be selected at a time. Parallel

means that each state on that level is active simultaneously. Exclusive states are identified by solid borders; alternatively, parallel states have dotted borders. Each exclusive level of the state machine must have a default state selected. A default state is selected by connecting a Default Transition, which is a transition connected only to that state; the loose end is bulleted. An action assigns a value to an output variable. Actions can occur within each state upon entry (en:), exit (ex:), or during (du:) residence.

Nodes, or Connective Junctions, represent a decision point and are used to connect multiple Transitions. Nodes are represented by small circles in the chart. The tool bar symbol for a node is the stacked red dots with arrows in and out of each.

Transitions (events) define the conditions required to pass between different states (on Exclusive levels). Transitions are represented by arrows connecting states and/or nodes. To create a Transition, click the edge of the departure node or state and drag to the destination node or state. The arrow head is connected to the destination that results from the required condition being met. Conditions are accessed by clicking the Transition and then clicking the text (a '?' for empty transitions). A condition contains variables and logical operators and is defined between square brackets. Variables can also be assigned values along a Transition; this happens inside of curly brackets.

At the top level, the SmartUnload model interfaces through structure variables. A structure can contain multiple signals, each of a different data type. By using the structure, the number of inputs and outputs at the high level is significantly reduced. In Simulink, structures are defined as Bus Objects from the Bus Editor tool. Instead of pulling out the variables from the input structure (input bus) needed for a state chart, it can be easier to input the entire structure. To access a structure variable within Stateflow, the dot notation is used. For example, the variable 'one' from structure 'b' is accessed on the transition from the 'Exclussive1' state to the connective junction by the notation 'b.one'.

Simulating the state chart within Stateflow is very useful when trying to validate functionality. As the model runs, the current location within the state chart will be highlighted blue. Input values can be changed from the model window and the transitions and change in output values can be observed. This method has been used many times throughout the project and has reduced development time.
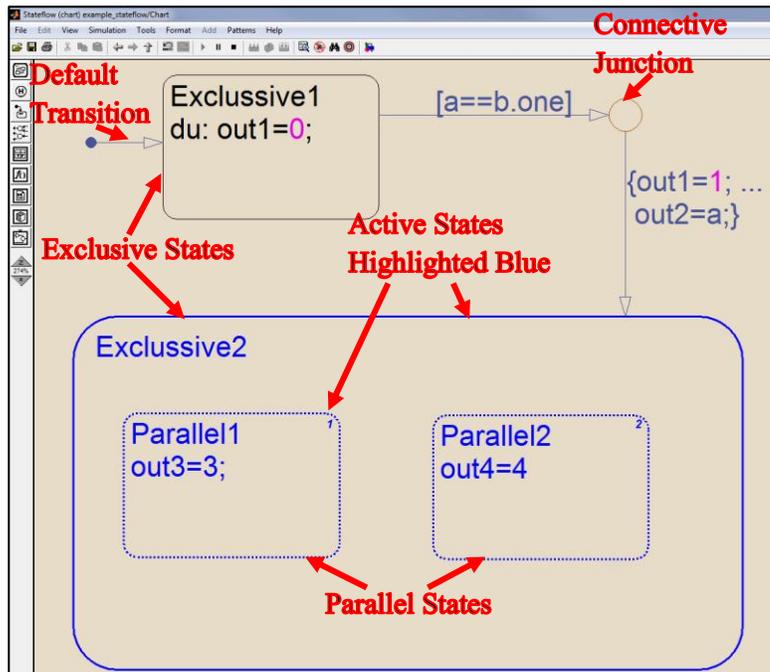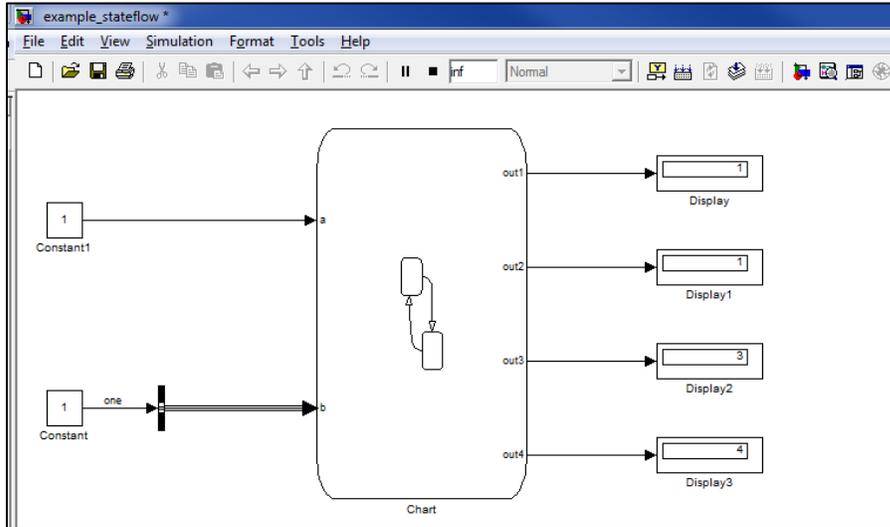
State Chart Development

Objectives

The goal of developing a state machine is to execute system-level logic and control the engagement level of SmartUnload. The first objective is to develop logic in a way that ensures operator safety. Second, the chart needs to conform to John Deere standard practices and fundamentals, so that farmers will recognize the flow when the system runs on their new machines; standardizing the format will also make the future transition of design responsibility from Iowa State University to John Deere smoother. Third, the state chart should be easily modified and adapted to include new features. New features that have been added to the state chart during the project include a transition from state 4 to state 3 based on a time-out event, an error code output, system engagement latching, and a transition from state 4 to state 3 that includes a full swing-out event. A high volume of testing has been completed while running the state machine and its operation has been very reliable.
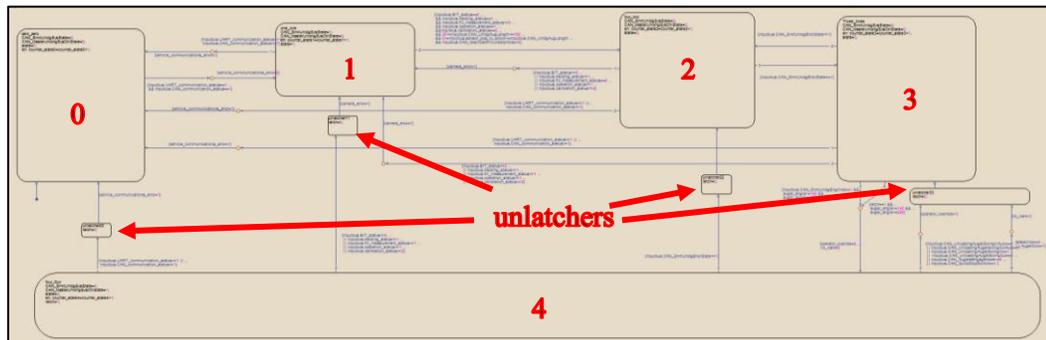
Methods & Materials

Considerations related to safety and conforming to sponsor standards as well as architecture to support required features and functions were important drivers for the design process. The design was validated using computer-based testing and simulation, test-track testing, and field testing. Stateflow, a toolset within Simulink, was used to construct the state chart and build it into the overall model.

Considerations

The primary purpose of the state chart is to protect the operator. The system has the potential to cause damage and harm people if it is fully engaged at incorrect times. Several state machine inputs are used to verify that the system is fully functional and ready to run. In addition, the operator is asked to confirm that he/she intends to operate with SmartUnload and must press a switch to fully engage the system to begin each unloading sequence.

John Deere has several conventions related to state charts. These conventions have already been applied in other applications. One precision agriculture application is AutoSteer; the AutoSteer state chart was studied and the state progression was emulated for SmartUnload. In general, a state chart should include states zero through four, and each lower state can be entered from any upper state according to defined events. States are considered to be analogous to pie pieces. A VI will show a circle with four equal-sized wedges; as state increases, the number of pieces highlighted in the circle (aka pie) also increases.

For the state chart to progress in a logical way, each consecutive state was designed to be at a noticeably higher level of system readiness/engagement. Defining what each state meant made the selection of conditions checked at each transition/event more straightforward.

The state chart was developed to protect the machinery too. The state chart checks that each piece of equipment is in a safe operating condition before allowing full engagement. This is important because customers/farmers will not accept the system if it causes equipment breakdowns; also, equipment breakdowns resulting from SmartUnload would increase warranty costs for John Deere.

Required Features & Functions

The state chart design also needed to support several additional features besides control of the system engagement level. The state chart performs a time-out routine based on the absence of a cart. The resume switch value is also latched when pressed and then unlatched when the system goes from state 4 to state 3. An error code is output that indicates the most recent reason for leaving state 4; this error code determines what is displayed to the operator on the diagnostics page of the VI. The most recent feature supported in the state chart is that the transition from state 4 to state 3 at the completion of a fill should happen

only after the auger is automatically swung all the way out; a Boolean variable that indicates when the auger is all the way out triggers the system disengagement. The design of the state chart supports the addition of new features, which is sure to continue as SmartUnload matures.

Design

Engaging the System

State '0' is the inactive state; the system is in state '0' when communications are not functioning correctly. The most basic requirement of the system is that CAN and serial communications must be operational. The condition to get from '0' to '1' is that the variables 'CAN_communication_status' and 'UART_communication_status' are equal to one, meaning both CAN and serial communications are functional.

Before transitioning from '1' to '2', the stereo system must be fully functional and the user-input parameters must be assigned reasonable values. For the stereo system to be fully functional, it must be calibrated, able to track, and able to measure fill. In other words, the variables 'tracking_status', 'fill_measurement_status', and 'operation_status' must equal one. The 'BIT_status' must equal zero and the 'calibration_status' must equal two. The operator must select reasonable values for auger length, fill-camera mounting location, and desired fill level. The 'CAN_UnldgAugLength' must be a value between thirty and one hundred decimeters. The 'sensor_pos_to_elbow' must be between zero and the 'CAN_UnldgAugLength'. The 'CAN_GrainCartFillLvlSetpntMax' must be greater than zero. In general, this state transition confirms that the system is ready for operation ("ready to be enabled").

State '3' is the enabled state. To reach state '3', the operator must press a button on the VI which indicates intention to use SmartUnload in the field; this is represented by 'CAN_SmrtUnldgEnblState' being equal to one.

State '4' is the active or engaged state. The system can go to '4' if the auger angle is within the safe range for operation and the operator presses the engagement switch. The 'auger_angle' must be greater than one-hundred and forty and less than two-hundred and twenty; the units of 'auger_angle' are half-degrees and that is the expected possible range of operation. 'CAN_SmrtUnldgEngCmd' has to equal one.

To summarize the system engagement process:

- Status of necessary modes of communication are checked

- System hardware and configuration is confirmed to be operable

- The user enables the system by pressing a button to confirm the intended use of SmartUnload

- The auger angle is confirmed to be within operational limits and the operator presses the engagement switch to allow the system to begin the SmartUnload process

Disengaging the System

The conditions for reducing the engagement level of the system are checked between the destination state and each higher state; this is important because those conditions can occur at any time (within any state) and seriously reduce safety and performance if the system were to remain engaged. If CAN or serial communication is lost, the system will enter '0' from any higher state. If the stereo system has a problem with calibration, tracking, or fill measurement, the state will be reduced to '1' from any higher state. If the operator disables the system through the VI, the system will return to '2' from either higher state.

The system will drop from '4' to '3' if the operator manually overrides an actuation, the emergency stop button is pressed, a cart is not detected for a specified period of time (cart-timeout), or the automatic swing-out is completed after the cart is full. Manual overrides are represented by five variables: 'CAN_UnloadingAugerSwingInAuto', 'CAN_UnloadingAugerSwingOutAuto', 'CAN_UnloadingAugerSwingIn', 'CAN_UnloadingAugerSwingOut', and 'CAN_AugersEngageMode'. The emergency stop variable is 'CAN_QuickStopSwitch'. If a manual override or emergency stop variable is equal to one, the transition occurs. The cart-timeout occurs when 'statecmd' is equal to three. 'AugerOut' will equal one when the automatic swing-out has been completed.

Supporting Additional Functionality

The state machine must also interact with features outside of controlling the system engagement level. To ensure that the system does not prematurely disengage, the operator engagement switch (momentary) value must be latched. This ensures that if the function is called again from state '0' the system will correctly be able to reach state '4'. To accomplish

the latch, a variable internal to the state machine called 'latch' was defined. A parallel event from state '3' to state '4' checks the condition of 'latch' instead of 'CAN_SmrtUnldgEngCmd'. If the system leaves state '4' for any reason, 'latch' is set equal to zero within the 'unlatcher##' states.

The system should timeout if a cart is not detected for a user-selected period of time. One good way to create a timer when the function runs at a fixed rate is to iteratively sum the task period (time in between function calls) and compare that sum to the timeout value. When the sum reaches the timeout period, the variable 'statecmd' is set equal to three.

A relatively new feature was added to SmartUnload to automatically swing the auger out after a cart is full and then drop to state '3'. Swinging the auger out and disengaging informs the operators of the combine and cart tractor that the fill is complete and puts the auger in the safest place for the cart to pull away from the combine. A variable 'AugerOut' indicates when the automatic swing-out is finished; if 'AugerOut' is equal to one the state will fall from '4' to '3'.

The state machine also outputs several diagnostic variables. A counter variable was created for each state as a developmental diagnostic to keep track of how many times each state is entered; this allows for the model design team to narrow down the cause of disengagement. There are also error codes output from the state machine. "Error code" is the term used by John Deere for diagnostic variables that indicate the reason behind not being in a more advanced state. State machine error codes are active when equal to one and include 'operator_override', 'no_cart', 'camera_error', and 'vehicle_communications_error'; respectively, these represent an operator override, a cart timeout, a camera malfunction or poor calibration, and the loss of a needed mode of communication. The various state machine error codes are combined to create the model output 'CAN_SmrtUnldgDisAlrm'. 'CAN_SmrtUnldgDisAlrm' is interpreted by the VI and displayed to the combine operator on the SmartUnload diagnostics page.

Validation

The robustness of the state machine is very critical to the safety and performance of SmartUnload. To validate the state machine, simulations were completed on the computer and testing was completed on the test track and in the field.

When using Simulink, it is possible to set up an automated test to check for the correct functionality of a model. A matrix of input conditions was designed to correspond to a specific matrix of outputs. By sending each set of inputs through the model, saving the model outputs, and then comparing the true model outputs to the expected outputs many initial errors were discovered and eliminated. Since the initial testing phase, most software simulations are done manually with specific checks in mind. A simulation model was created for manual testing with static inputs and a display for each output. While the simulation is running, the inputs can be changed and the corresponding output change can be observed. Each new feature is tested in software before it is built to the MAB and tested on a machine.

Early in development, some testing was centered on validating state machine functionality. As the system grew and changes to the state machine became smaller and less frequent, state machine testing on the combine has primarily been done concurrent to testing other SmartUnload features. The state machine runs every time the model is called; therefore, errors are very quickly discovered related to its functionality. As with the rest of the model, the state machine is tested on the MAB here at Iowa State to verify correct performance before the model is built and sent to John Deere for additional testing; it is important to ensure efficient transfer of new models and to reduce the amount of resources John Deere must dedicate to the process. John Deere will send a build back that can be put on the production-intent ECU; the final testing is done on that ECU to confirm that performance is no different than when executing from the MAB.

Testing on a combine was done by the team members at Iowa State as well as employees of John Deere. The Iowa State team was able to test year round. Multiple fields were harvested within a few miles of BCRF to validate SmartUnload performance in real crop conditions. During the rest of the year, testing was completed in the gravel yard at BCRF and the test track at the AEA farm. John Deere employees tested with the production-intent ECU in multiple crops throughout the Midwest and the South.

Results & Conclusions

A state machine has been designed to ensure the safe, reliable, and functional performance of SmartUnload. Discrete levels of engagement are identified by basic

communication requirements, system configuration, operator intent, and final engagement of the SmartUnload process. John Deere conventions were used to create a four pie-piece (five-state) state machine; this allows for faster customer recognition and understanding.

Stateflow was used to construct the state chart and tie it into the rest of the SmartUnload model. Testing was completed using computer simulations and a large quantity of machine tests. The functionality of the state chart has been demonstrated to work well and be highly adaptable to the addition of new features. This work provides a strong foundation in system-level logic execution as the SmartUnload project continues into the future.

Control Implementation Model

Objectives

Control Implementation (CI) has three main purposes: calculate & provide key parameters, actuate to meet the requests of Fill Strategy, and prevent spillage.

Key parameters are calculated using the highest resolution possible within the required fixed-point data type. Some parameters are used within Control Implementation only and others are used by other subsystems or the stereo software. The calculated parameters include auger angle, the location of the boot relative to the grain cart, the potential column location range, the desired fill location and shift, and the mode-of-control.

Fill Strategy sends an actuation request to Control Implementation for swing and auger engagement based on the current fill state within the cart. Control Implementation executes actuations based on the requests and the system status.

Spill prevention is a key SmartUnload feature. Spilling less grain than a typical operator is an absolute requirement. There are many cases for potential spillage. One example is when the combine operator needs to quickly stop the machine; the relative velocity of the auger boot with respect to the rear of the cart will be very high, and the auger must shut off as quickly as possible to prevent grain loss.

The Control Implementation system has been implemented using a modular model structure for readability, adaptability, and scalability. Each sub-system has a specific function and the flow of data is organized left-to-right. Variables and operations are named to match their purpose. The overall SmartUnload model was developed using a hierarchical approach to support concurrent development of the major subsystems (Fill Strategy, Control Implementation, and Speed Control).

Inputs

Mode of Control & Auger Angle

Cart Edge Locations

Boot Locations

Smart Edge Dead-band & Current Fill Location

Desired Shift & Dead-band Output

Decision Logic & Proportional Control

Conflict Recognition & Signal Packing

Conflict Resolution

Diagnostics
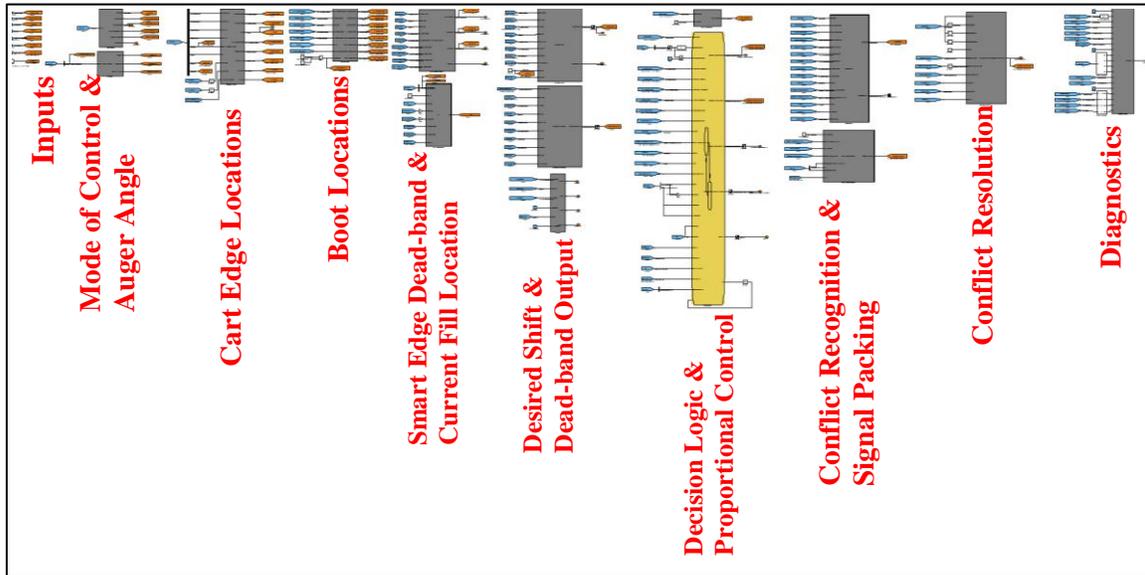
Methods & Materials

Simulink and its libraries were used to create the model. The most recent CI model is always saved as a library file in the correct model directory, so that it can be referenced by the higher-level SmartUnload model. The most recent CI has eleven main features/functions (comprised of fifteen total subsystems), which are each defined in this section.

Auger Angle Calculation

The purpose of the 'AugerAngleSub' is to produce all values related to auger angle and auger geometry, so that other subsystems can perform related downstream calculations. These values include 1-degree resolution auger angle, the maximum auger angle, the 0.5-degree resolution auger angle, the offset distance between the tracking (grain-tank) camera and the elbow, the machine type, and auger velocity.

Auger angle is calculated using the cubic calibration equation. The coefficients of the equation are either selected based on machine-type and auger length or manually input through the user interface. Machine type and auger length are both available in the model input structure. The calibration equation uses 32-bit integer math. To fully utilize the range of that data-type the equation is scaled up; to get the final value for angle, a division by 500000 is required. The 0.5-degree resolution value, used by the stereo software and most model operations, is calculated by dividing by 250000 instead.

The maximum auger angle is considered to be one degree less than the output of the calibration equation for an input of 250. The maximum auger angle is used as a limit for outputting a swing-out command to prevent actuation against the mechanical stops.

The offset distance between the tracking camera and the auger elbow is known for each specific machine type.

Auger velocity is the three-period average of the point-to-point slope calculation using auger angle and task period (time between function calls). To date, this signal has not been used. Auger velocity will not be useful until noise is reduced, probably through more advanced filtering.
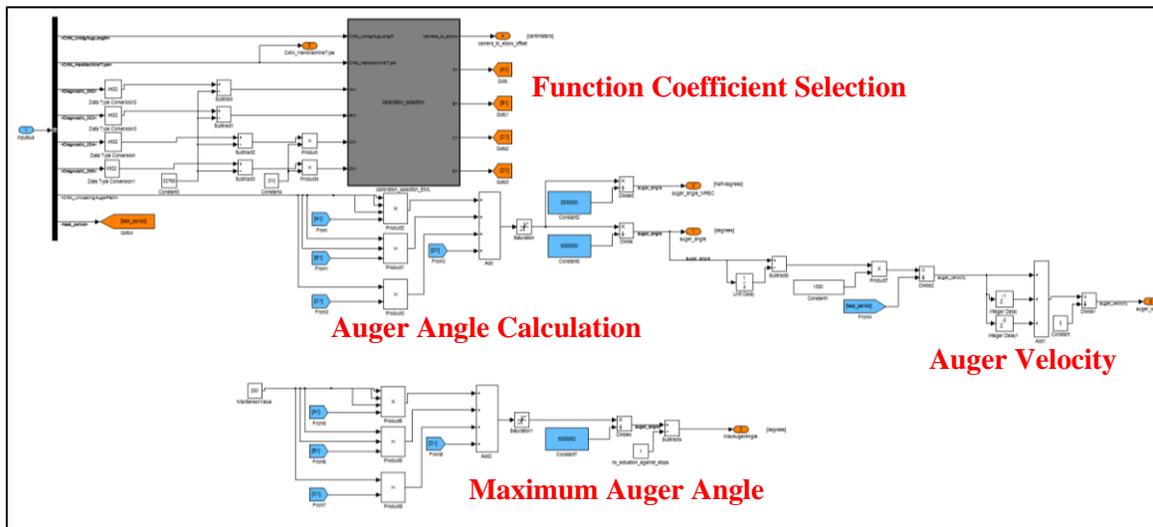
Mode-of-Control Determination

The mode-of-control (MOC) input to the model specifies which type of actuation will be used. Swing actuation was the focus of this thesis but several other modes can be selected: passive combine speed control, active combine speed control, light-bar, and the various combinations. The MOC variable is an unsigned, 8-bit integer and the modes are represented bit-wise. The first bit is high if the default mode, auger swing and engagement, is selected. Bits two and three correspond to light-bar and passive combine speed control respectively; light-bar is no longer a viable option so if either are selected, passive combine speed control will execute. Bit six is high if only auger engagement is controllable by SmartUnload. The other modes and associated bits are not relevant here. The MOC input is fed through a 'Bitwise AND' within the model for each of the eight bits and their output is

used to specify a '0' or '1' value for the corresponding mode. If a mode has a value of '1', it can be used later as a command output.
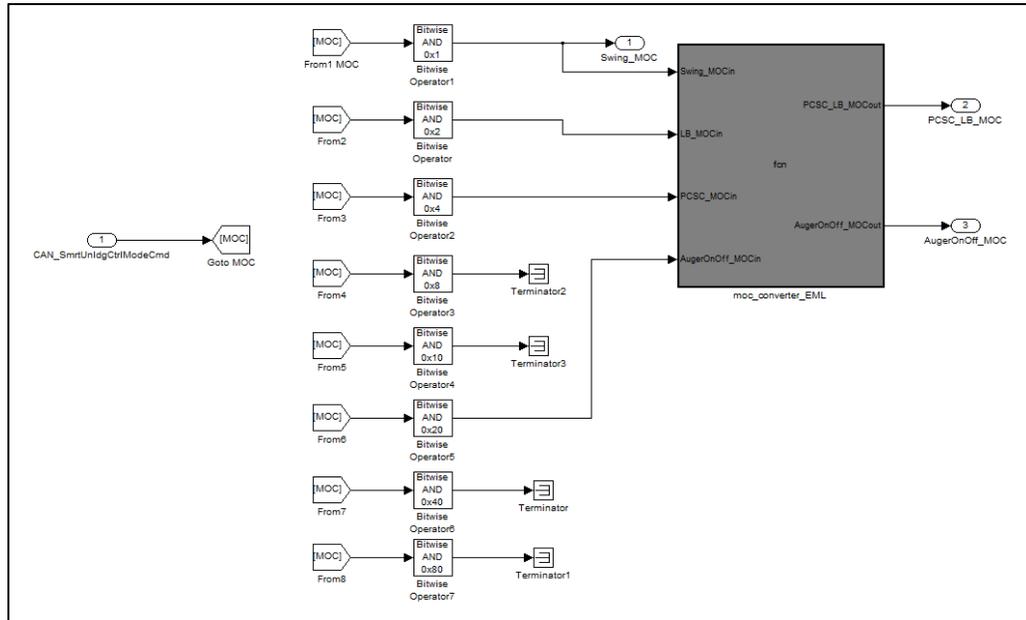
Stereo Tracking Data Processing

Stereo data is processed in the embedded MatLab function (EML) named 'Edge_Locator_EML'. The stereo tracking inputs to this EML include the xyz-coordinate and tracking status of the front-inside and rear-inside corners of the grain cart, the number of columns and rows in the cart, and the cell dimensions of the fill grid. The offset from the tracking camera to elbow is also an input. The primary outputs are the distance from the combine elbow to each edge of the grain cart, an overall tracking status, and a value indicating which edge of the cart is being referenced.

Cart width is calculated first. If the number of rows in the cart is greater than five, indicating that a valid cart is present, then the cart width is calculated as the number of rows multiplied by the cell height. If that is not true and the cart width calculated during the previous model call is greater than 125cm, then the new cart width is equal to the previous cart width. If neither of those is met, then the cart width is assumed to be 300cm. Cart width is used downstream to determine the distance from the boot to the outside cart edge. The operation was designed to be conservative while not preventing system operation if the value for number of rows is invalid.

Cart length is determined next. If both the front edge and rear edge are actively being tracked by the stereo system, the cart length is equal to the front x-coordinate minus the rear x-coordinate. If neither, the front or back, is actively tracked then the cart length defaults to 600cm. If neither of those conditions is met then the cart length is set equal to the cart length from the previous model iteration.

Following the cart width and cart length calculations, the distance from the elbow to the front and rear edges is determined. Within this portion of code, the variables indicating the reference edge and overall tracking status are also assigned values. If the front and rear edges are both actively tracked, the front edge is selected as the default reference edge; the distance from the rear edge to the elbow is equal to the rear x-coordinate minus the offset between the tracking camera and elbow, and the distance from the front edge to the elbow is equal to the front x-coordinate plus the offset between the tracking camera and elbow. The various combinations of front and rear tracking status will determine which edge is used for a reference during calculations. Next the distance from the combine to the inside and outside edges of the cart is found; the distance to the inside edge is the larger value of the front and rear y-coordinates, and the distance to the outside edge is the distance to the inside edge plus the cart width (code provided here).

```matlab
function [right_edge_to_elbow, left_edge_to_elbow, tracking,
bottom_edge_to_elbow, ...
    top_edge_to_elbow, track_off_front, ...
    cart_length, cart_width] = Edge_Locator_EML(front_coordinate, ...
    rear_coordinate, front_status, rear_status, cell_height, nrows,
cart_width_old, cart_length_old, offset_camera_to_elbow)
%#eml

%-----Parameters
tracking=uint8(1);

if nrows>5
   cart_width=int16(int16(nrows)*int16(cell_height));
elseif cart_width_old>125
   cart_width=int16(cart_width_old);
else
   cart_width=int16(300);
end

%------Cart Length Calc

if front_status==1 && rear_status==1 %too restricting? 1&&0 || 0&&1
allowed?
```

```matlab
    cart_length=int16(int16(front_coordinate(1))-
int16(rear_coordinate(1)));
elseif front_status==0 && rear_status==0
    cart_length=int16(600); %[cm] default length
else
    cart_length=int16(cart_length_old); %[cm] default length
end

%-----Left and Right Edge Locations and Tracking
if front_status==1 && rear_status==1
    track_off_front=uint8(1);
    left_edge_to_elbow1= int16(-int16(rear_coordinate(1))-
int16(offset_camera_to_elbow));

right_edge_to_elbow1=int16(int16(front_coordinate(1))+int16(offset_camera_
to_elbow));
elseif front_status==1 && rear_status==2
    track_off_front=uint8(1);

right_edge_to_elbow1=int16(int16(front_coordinate(1))+int16(offset_camera_
to_elbow));
    if cart_length~=600
        left_edge_to_elbow1=int16(int16(cart_length)-
int16(right_edge_to_elbow1));
    else
        left_edge_to_elbow1=int16(-int16(rear_coordinate(1))-
int16(offset_camera_to_elbow));
    end
elseif front_status==2 && rear_status==1
    track_off_front=uint8(0);
    left_edge_to_elbow1=int16(-int16(rear_coordinate(1))-
int16(offset_camera_to_elbow));
    if cart_length~=600
        right_edge_to_elbow1=int16(int16(cart_length)-
int16(left_edge_to_elbow1));
    else

right_edge_to_elbow1=int16(int16(front_coordinate(1))+int16(offset_camera_
to_elbow));
    end
elseif front_status==2 && rear_status==2
    track_off_front=uint8(1);
    left_edge_to_elbow1=int16(-int16(rear_coordinate(1))-
int16(offset_camera_to_elbow));

right_edge_to_elbow1=int16(int16(front_coordinate(1))+int16(offset_camera_
to_elbow));
else
    track_off_front=uint8(1);
    tracking=uint8(0);
    left_edge_to_elbow1=int16(100);
    right_edge_to_elbow1=int16(100);
end

left_edge_to_elbow=int32(left_edge_to_elbow1);
```

```
right_edge_to_elbow=int32(right_edge_to_elbow1);

%-----bottom and top edge locations
if front_coordinate(2)>rear_coordinate(2) %select the most conservative y-
distance from the combine to the cart
    bottom_edge_to_elbow1=int16(front_coordinate(2));
else
    bottom_edge_to_elbow1=int16(rear_coordinate(2));
end
top_edge_to_elbow1=int16(int16(bottom_edge_to_elbow1)+int16(cart_width));
bottom_edge_to_elbow=int32(bottom_edge_to_elbow1);
top_edge_to_elbow=int32(top_edge_to_elbow1);
```

Calculating Boot Location

The boot location is calculated for control and also as an input for determining the current fill location. The distance from the boot to each edge is a function of auger angle and requires the use of sine and cosine functions. The inputs required to calculate a boot location are angle, auger length, and cart position (relative to the combine-auger elbow). The following four functions represent the relationships used.

$$BootToOutsideEdge = OutsideEdgeToElbow - AugerLenght \times \sin(\theta)$$

$$BootToInsideEdge = AugerLength \times \sin(\theta) - InsideEdgeToElbow$$

$$BootToRearEdge = RearEdgeToElbow - AugerLength \times \cos(\theta)$$

$$BootToFrontEdge = AugerLength \times \cos(\theta) + FrontEdgeToElbow$$

Trigonometric functions are not supported on the fixed-point processor, so approximations were developed for sine and cosine. The intention of the approximation is to accurately represent the functions in the operational range of SmartUnload in order to calculate an accurate position during engagement. The outputs from the sine and cosine functions were plotted over the 60-115 degree range with a point every five degrees. Cosine is linear in that range and sine is quadratic. Trend-lines were fit to the points and the coefficients were scaled for fixed-point math. Additional scaling was required to use the approximated functions with 0.5degree resolution. The fit curves are in the figure below.
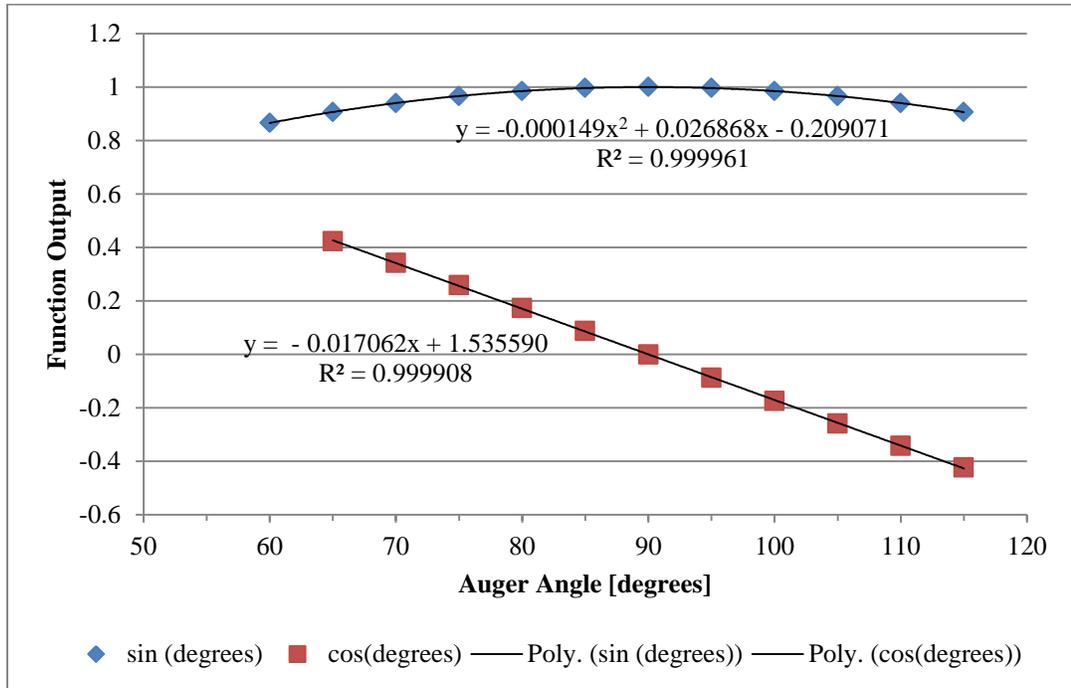
The calculations were repeated three times. The first set was used to determine the current boot location based on auger angle. The second and third calculations were used to get the boot locations possible for the minimum and maximum auger angles (see below).
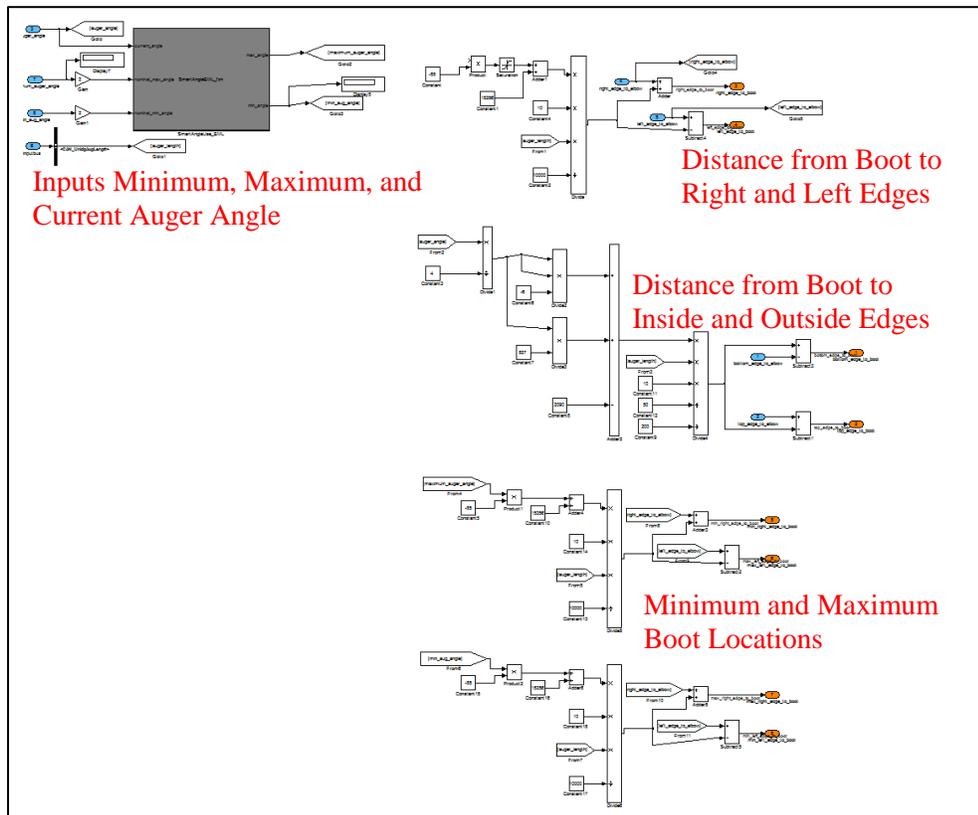
Current, Minimum, and Maximum Fill Location

The EML called 'Current_Fill_Location_EML' outputs the current, minimum, maximum, and real fill locations. FS uses the current location to determine if the grain flow is in the desired location. The minimum and maximum fill locations are used by FS to determine the range of possible desired locations. The real fill location is the absolute column number used by the Speed Control (SC) subsystem. All locations are given as a column number corresponding to the fill grid. The locations used by FS are saturated between zero and the number of columns in the cart. The real fill location given to SC is not saturated and can be plus or minus one-hundred and twenty-five columns (with the first column just inside the rear edge of the cart). The equations below are generalized boot location (column) calculations corresponding to using the rear or front edge as a reference. The code is given below the equations.

$$FillLocation = RearEdgeToBoot/ColumnWidth$$

$$FillLocation = NumberOfColumns - \left(\frac{FrontEdgeToBoot}{ColumnWidth}\right)$$

```matlab
function [current_fill_location, max_fill_location, min_fill_location,
real_fill_location] = Current_Fill_Location_EML(track_off_front,
ncolumns_FS, cell_width_FS, left_edge_to_boot, right_edge_to_boot, ...
    min_right_edge_to_boot, max_left_edge_to_boot, max_right_edge_to_boot,
min_left_edge_to_boot)
%#eml

new_offset=int32(int32(cell_width_FS)/int32(2)); %offset to make current
fill location more accurate

%if we track off the front, then the current fill location should be based
%on the front edge location....etc
if track_off_front~=1
current_fill_location1=int32((int32(left_edge_to_boot)+int32(new_offset))/
int32(cell_width_FS));
    if current_fill_location1>ncolumns_FS
        current_fill_location=uint8(ncolumns_FS);
    elseif current_fill_location1<1
        current_fill_location=uint8(1);
    else
        current_fill_location=uint8(current_fill_location1);
    end
max_fill_location1=int32((int32(max_left_edge_to_boot)+int32(new_offset))/
int32(cell_width_FS));
    if max_fill_location1>ncolumns_FS
        max_fill_location=uint8(ncolumns_FS);
    elseif max_fill_location1<1
        max_fill_location=uint8(1);
    else
```

```
        max_fill_location=uint8(max_fill_location1);
    end
min_fill_location1=int32((int32(min_left_edge_to_boot)+int32(new_offset))/
int32(cell_width_FS));
    if min_fill_location1>ncolumns_FS
        min_fill_location=uint8(ncolumns_FS);
    elseif min_fill_location1<1
        min_fill_location=uint8(1);
    else
        min_fill_location=uint8(min_fill_location1);
    end
else
    current_fill_location1=int32(int32(ncolumns_FS)-
((int32(right_edge_to_boot)-int32(new_offset))/int32(cell_width_FS)));
    if current_fill_location1>ncolumns_FS
        current_fill_location=uint8(ncolumns_FS);
    elseif current_fill_location1<1
        current_fill_location=uint8(1);
    else
        current_fill_location=uint8(current_fill_location1);
    end
    max_fill_location1=int32(int32(ncolumns_FS)-
((int32(min_right_edge_to_boot)-int32(new_offset))/int32(cell_width_FS)));
    if max_fill_location1>ncolumns_FS
        max_fill_location=uint8(ncolumns_FS);
    elseif max_fill_location1<1
        max_fill_location=uint8(1);
    else
        max_fill_location=uint8(max_fill_location1);
    end
    min_fill_location1=int32(int32(ncolumns_FS)-
((int32(max_right_edge_to_boot)-int32(new_offset))/int32(cell_width_FS)));
    if min_fill_location1>ncolumns_FS
        min_fill_location=uint8(ncolumns_FS);
    elseif min_fill_location1<1
        min_fill_location=uint8(1);
    else
        min_fill_location=uint8(min_fill_location1);
    end
end

if current_fill_location1>120
    real_fill_location=int8(120);
elseif current_fill_location1<-120
    real_fill_location=int8(-120);
else
    real_fill_location=int8(current_fill_location1);
end
```

Calculate the Smart Edge Dead-band for Spill Prevention

Several functions are required for robust spill prevention. The first iteration of the spill-prevention feature applied a single dead-band to the front and rear cart edges in which FS was not allowed to fill. If the boot entered the dead-band, the auger would swing back

towards the center of the cart and disengage. Expert operators indicated that this was not desirable performance; when possible, the auger should swing to stay in the cart and then disengage if swing is not capable of preventing grain spillage. For this to work CI needs to provide FS with the number of columns that will be considered un-fillable; the number of columns is the nearest integer greater than or equal to the edge dead-band divided by the column width. When the boot enters the un-fillable regions, a swing is initiated. If the boot enters too far into the un-fillable region then the auger will disengage; this is accomplished by applying an offset value [centimeters] to the edge dead-band [also in centimeters]. When the boot is closer to the front than the back and the auger angle is less than or equal to the minimum value, the offset is not applied; the auger will disengage immediately upon entering the front un-fillable area, because it is not capable of swinging farther into the cart. Similarly when the boot is closer to the back than the front and the auger angle is greater than or equal to the maximum value, the offset is not applied; the auger will disengage immediately upon entering the rear un-fillable area, because it is not capable of swinging farther into the cart. The code is provided here.

```
function edb = fcn(frontback_edb, db_offset, auger_angle, min_augangle,
max_augangle, right_edge_to_boot, left_edge_to_boot, swing_MOC)
%#eml

%don't have an offset on edge deadband for auger disengagement when
%swinging away from an edge is not possible
if right_edge_to_boot<left_edge_to_boot && auger_angle<=min_augangle
    edb=int16(frontback_edb);
elseif left_edge_to_boot<=right_edge_to_boot && auger_angle>=max_augangle
    edb=int16(frontback_edb);
else
    edb=int16(int16(frontback_edb)-int16(db_offset));
end

if swing_MOC==0
    edb=int16(frontback_edb);
end
```

Desired Fill Location and Shift

Two EML's are responsible for calculating the desired shift in centimeters based on two inputs, each given as a number of columns, from FS. The first input from FS represents the column shift required to reach the most optimum fill location. The second input from FS represents the column shift required to reach the best fill location within the swing range of the auger.

The other inputs to the EML's are the same. Each EML takes the column shift input and determines what the desired column location is; the desired column location is equal to the current fill location plus the desired shift value in columns (from FS). The distance from the center of each column to the reference edge and the distance from the boot to the reference edge are both in units of centimeters. From those two known values, the required shift magnitude in centimeters is calculated and used as the error input to the control function. The desired column location and centimeter shift are both outputs from CI that are used by SC. The code given below is for one of the EML's; the other EML is very similar.

```
function [relx_desired_cm, desired_fill_location] =
Desired_Shift_EML(relx_desired_cells, ...
    Diagnostic_005, ...
    dist_to_back, ...
    cell_width_FS, ...
    track_off_front, ...
    dist_to_front, ...
    desired_column, ...
    current_fill_location, ...
    CI_damping, ...
    ncolumns_FS)
%#eml


%----calculate relx_desired_cm
%if tracking off of the front, use terms relative to the front edge
location
%if the desired shift from FS changes, recalculate the desired distance to
each edge
%if the desired shift from FS doesn't change, then use the old desired
distance to each edge
if Diagnostic_005==2  % '2' is for CI only on MAB and assumes a constant
'9' column 67" cell width cart (see edge location EML also)
    if track_off_front~=1
        desired_dist_to_back=int32(desired_column)*int32(cell_width_FS)-
(int32(cell_width_FS)/int32(2));
        relx_desired_cm=int32(desired_dist_to_back)-int32(dist_to_back);
    else
        desired_dist_to_front=int32(int32(ncolumns_FS)-
int32(desired_column))*int32(cell_width_FS)+(int32(cell_width_FS)/int32(2)
);
        relx_desired_cm=int32(dist_to_front)-int32(desired_dist_to_front);
    end
    desired_fill_location=int8(desired_column);
else %normal operation with ultrasonics or stereo based fill strategy
    if track_off_front~=1
        if relx_desired_cells==0
            relx_desired_cm=int32(0);
            desired_fill_location=int8(current_fill_location);
        else
```

```
desired_dist_to_back=(int32(current_fill_location)+int32(relx_desired_cell
s))*int32(cell_width_FS)-(int32(cell_width_FS)/int32(2));
            relx_desired_cm=int32(desired_dist_to_back)-
int32(dist_to_back);
            if relx_desired_cm<0
                relx_desired_cm=int32(relx_desired_cm)+int32(CI_damping);
            elseif relx_desired_cm>0
                relx_desired_cm=int32(relx_desired_cm)-int32(CI_damping);
            end
desired_fill_location=int8(int8(current_fill_location)+int8(relx_desired_c
ells));
        end
    else
        if relx_desired_cells==0
            relx_desired_cm=int32(0);
            desired_fill_location=int8(current_fill_location);
        else
            desired_dist_to_front=(int32(ncolumns_FS)-
(int32(current_fill_location)+int32(relx_desired_cells)))*int32(cell_width
_FS)+(int32(cell_width_FS)/int32(2));
            relx_desired_cm=int32(dist_to_front)-
int32(desired_dist_to_front);
            if relx_desired_cm<0
                relx_desired_cm=int32(relx_desired_cm)+int32(CI_damping);
            elseif relx_desired_cm>0
                relx_desired_cm=int32(relx_desired_cm)-int32(CI_damping);
            end
desired_fill_location=int8(int8(current_fill_location)+int8(relx_desired_c
ells));
        end
    end
end

if desired_fill_location<1
    desired_fill_location=int8(1);
elseif desired_fill_location>ncolumns_FS
    desired_fill_location=int8(ncolumns_FS);
end
```

Proportional Control

The proportional control implementation was necessarily complicated within CI. The proportional command had to be packed within the same unsigned 8-bit integer signal used for the on-off swing commands, so that the output structure from the SUC model would be the same when operating with or without proportional control. The 0-125 range was used for the swing-in proportional value and the 126-250 range was used for the swing-out proportional value. This method resulted in a relatively low utilization of valve duty-cycle resolution compared to the PWM capability of the valve-driver board; however, the resolution was high enough to allow for smooth control.

Accomplishing this was a two-step process. First, a 0-250 proportional value was calculated in the 'Proportional_Value_Sub' subsystem. The function is given in the next figure. An 'aggressiveness' of 13 worked very well during harvest.
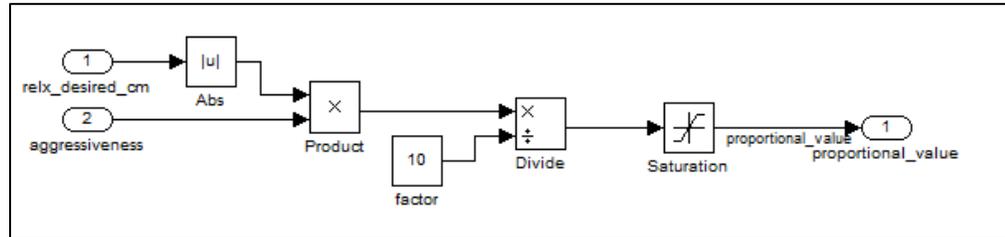
Second, within the 'Proportional_BackEnd_EML' EML, the output command was packed as a function of proportional value, swing direction, and the activation of proportional control. The code is provided here.

```
function CAN_AugRotnCmdPcnt = Proportional_BackEnd_EML(proportional_mode,
CAN_MasterUnldngSysCtrlState, proportional_value, CAN_AugRotnCmdPcnt_norm,
CAN_UnloadingAugerSwingOutButtonButton, CAN_UnloadingAugerSwingIn,
CartFull)
%#eml

if CAN_UnloadingAugerSwingOutButtonButton==1 && proportional_mode==1
    CAN_AugRotnCmdPcnt=uint8(249);
elseif CAN_UnloadingAugerSwingIn==1 && proportional_mode==1
    CAN_AugRotnCmdPcnt=uint8(124);
else
    if proportional_mode==1 && CAN_MasterUnldngSysCtrlState==1
        if CAN_AugRotnCmdPcnt_norm==251 %swing out

CAN_AugRotnCmdPcnt=uint8(uint8(proportional_value)/uint8(2)+uint8(125));
            if CAN_AugRotnCmdPcnt>250
                CAN_AugRotnCmdPcnt=uint8(250);
            end
            if CartFull==1
                CAN_AugRotnCmdPcnt=uint8(249);
            end
        elseif CAN_AugRotnCmdPcnt_norm==252 %swing in
            CAN_AugRotnCmdPcnt=uint8(uint8(proportional_value)/uint8(2));
            if CAN_AugRotnCmdPcnt>125
                CAN_AugRotnCmdPcnt=uint8(125);
            end
        else
            CAN_AugRotnCmdPcnt=uint8(253);
        end
    else
        CAN_AugRotnCmdPcnt=CAN_AugRotnCmdPcnt_norm;
    end
end
```

Proportional control only ran on combine 109 with the MAB. There was no way to turn on proportional control from the IPM, and combine 109 was the only machine with proportional valves. To switch from bang-bang to proportional control required changing one input in ControlDesk from a '0' to a '1' and turning two ball-valves. The methods used here allowed for the same CI to be used for MAB and IPM builds; that is a critical feature for efficient development and version control. A CAN message containing the proportional command signals is packed and transmitted from the MAB and received by the valve-driver ECU.

Decision Logic

Central to CI functionality is the 'CI_Decision_Chart', a Stateflow chart containing the logic responsible for determining the SmartUnload output commands. The chart is four levels deep. On the top level are four exclusive states: an inactive state, active state, and two secondary states which make up the auto-swing-out feature. The inactive state is entered by default. To enter the active state, SmartUnload must be engaged and the system has to be tracking. When SmartUnload is disengaged or tracking is lost, CI returns directly to the inactive state from the active state. FS notifies CI when the cart is full; when the cart is full, CI enters the first auto-swing-out state in which the swing-out command is given. Once the auger angle sensor value is greater than or equal to 248 out of 250, the second auto-swing-out state is entered. The purpose of the second auto-swing-out state is to delay entering the inactive state by one model call; this gives the SUC state machine time to enter state '3' and disengage SmartUnload.

The next level has two parallel states representing the auger swing and auger engagement modes of control. Within the auger swing state are three exclusive states: 'swing in', 'don't swing', and 'swing out'. The default state is 'don't swing'. If the desired shift is great in magnitude than the swing dead-band and a negative number, the auger angle is greater than the minimum, and the swing MOC is selected then the auger will swing in; once any of those conditions are no longer true, CI will return to the 'don't swing' state. The 'swing-out' state is entered if the desired shift is greater than the swing dead-band, the auger angle is less than the maximum, and the swing MOC is selected; once any of those conditions are no longer true, CI will return to the 'don't swing' state.

The auger engagement state contains two exclusive states, 'disengaged' and 'engaged'. The default state is 'disengaged'. The 'engaged' state is entered if the distance to each edge is greater than the respective edge dead-bands, the auger on-off MOC is selected, and FS is requesting that the auger is turned on. This functionality was most recently implemented for the inside and outside edges; it is important that the auger disengage if the cart gets too far away from the combine. A cart may not be traveling parallel to the combine as it approaches or leaves; to conservatively account for this, the largest y-coordinate is used for determining the distance from the boot to the inside cart edge. Testing has shown that this works robustly despite relatively inaccurate y-coordinate values from the stereo system.

System-level Conflict Recognition and Resolution

Two EMLs ('Red_Flags_EML' and 'Lockup_Breaker_EML') were developed to address a problem corner-case discovered during early phases of SmartUnload testing. Periodically, FS would desire a shift but CI would not actuate because the shift magnitude was less than the swing dead-band. The swing dead-band is the minimum distance of actuation that would not result in oscillation around the desired location.

The purpose of the 'Red_Flags_EML' is to identify when either the FS and CI swing commands or auger engagement commands do not agree. The two outputs are a red flag variable for swing and a red flag variable for auger engagement; the red flag variables indicate when FS and CI are conflicting. The auger engagement red flag is used for diagnostic purposes only. The swing red flag is an input to the 'Lockup_Breaker_EML'. If the swing red flag indicates a conflict for a set number of model iterations, then the command output for swing will be in the direction of the FS request for a single model call; this operation effectively nudges the auger until FS is satisfied with the boot location. Other parameter values that are evaluated before allowing a nudge are auger angle and the swing MOC.

Leading up to and throughout fall testing, there did not seem to be many conflicts between FS and CI; this improvement can be credited to changes in FS logic (Jennett, 2012). The two EMLs remain a good fail safe against future potential lockup conditions (the code for each is below).

```
function [RedFlag_Swing, RedFlag_AugOnOff] = Red_Flags_EML(auger_angle,
minimum_auger_angle, maximum_auger_angle, ...
    relx_desired_cells_prime, CAN_AugRotnCmdPcnt, right_edge_deadband,
left_edge_deadband, right_edge_to_boot, left_edge_to_boot, ...
    FSaugCMD, CAN_AugersEngageModeCmd, CAN_MasterUnldngSysCtrlState,
tracking, stereo_v_ultra, relx_desired_cm)
%#eml

if stereo_v_ultra~=2
    if auger_angle<maximum_auger_angle && relx_desired_cells_prime>0 &&
CAN_AugRotnCmdPcnt==253 && CAN_MasterUnldngSysCtrlState==1 && tracking==1
        RedFlag_Swing=uint8(1);
    elseif auger_angle>minimum_auger_angle && relx_desired_cells_prime<0
&& CAN_AugRotnCmdPcnt==253 && CAN_MasterUnldngSysCtrlState==1 &&
tracking==1
        RedFlag_Swing=uint8(1);
    else
        RedFlag_Swing=uint8(0);
    end
else
    if auger_angle<maximum_auger_angle && relx_desired_cm>25 &&
CAN_AugRotnCmdPcnt==253 && CAN_MasterUnldngSysCtrlState==1 && tracking==1
        RedFlag_Swing=uint8(1);
    elseif auger_angle>minimum_auger_angle && relx_desired_cm<-25 &&
CAN_AugRotnCmdPcnt==253 && CAN_MasterUnldngSysCtrlState==1 && tracking==1
        RedFlag_Swing=uint8(1);
    else
        RedFlag_Swing=uint8(0);
    end
end
```

```matlab
if FSaugCMD==1 && CAN_AugersEngageModeCmd~=1 &&
right_edge_to_boot>right_edge_deadband &&
left_edge_to_boot>left_edge_deadband && CAN_MasterUnldngSysCtrlState==1 &&
tracking==1
    RedFlag_AugOnOff=uint8(1);
else
    RedFlag_AugOnOff=uint8(0);
end


function CAN_AugRotnCmdPcnt_out  = Lockup_Breaker_EML(RedFlag_Swing,
RedFlag_Swing_old, RedFlag_Swing_oldold, RedFlag_Swing_oldoldold,
NumDelays, CAN_AugRotnCmdPcnt, CAN_AugRotnCmdPcnt_out_old,...
    relx_desired_cells_prime, stereo_v_ultra, relx_desired_cm, Swing_MOC)
%#eml

%lockup breaker always uses a 50% valve dutycycle but the initial delay
%before starting can be changed; this allows for the auger to finish the
%overshoot and reach a steady-state location before enabling the lockup
%breaker functionality

if NumDelays>3
    NumDelays=uint8(3);
else
    NumDelays=uint8(NumDelays);
end

if NumDelays==0

    if RedFlag_Swing==1 && CAN_AugRotnCmdPcnt_out_old==253 &&
stereo_v_ultra~=2 && Swing_MOC==1
        if relx_desired_cells_prime>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    elseif RedFlag_Swing==1 && CAN_AugRotnCmdPcnt_out_old==253 &&
stereo_v_ultra==2 && Swing_MOC==1
        %relx_desired_cm for ci-only
        if relx_desired_cm>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    else
        CAN_AugRotnCmdPcnt_out=uint8(CAN_AugRotnCmdPcnt);
    end

elseif NumDelays==1

    if RedFlag_Swing==1 && RedFlag_Swing_old==1 &&
CAN_AugRotnCmdPcnt_out_old==253 && stereo_v_ultra~=2 && Swing_MOC==1
        if relx_desired_cells_prime>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
```

```matlab
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    elseif RedFlag_Swing==1 && RedFlag_Swing_old==1 &&
CAN_AugRotnCmdPcnt_out_old==253 && stereo_v_ultra==2 && Swing_MOC==1
        %relx_desired_cm for ci-only
        if relx_desired_cm>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    else
        CAN_AugRotnCmdPcnt_out=uint8(CAN_AugRotnCmdPcnt);
    end

elseif NumDelays==2

    if RedFlag_Swing==1 && RedFlag_Swing_old==1 && RedFlag_Swing_oldold==1
&& CAN_AugRotnCmdPcnt_out_old==253 && stereo_v_ultra~=2 && Swing_MOC==1
        if relx_desired_cells_prime>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    elseif RedFlag_Swing==1 && RedFlag_Swing_old==1 &&
RedFlag_Swing_oldold==1 && CAN_AugRotnCmdPcnt_out_old==253 &&
stereo_v_ultra==2 && Swing_MOC==1
        %relx_desired_cm for ci-only
        if relx_desired_cm>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    else
        CAN_AugRotnCmdPcnt_out=uint8(CAN_AugRotnCmdPcnt);
    end

else

    if RedFlag_Swing==1 && RedFlag_Swing_old==1 && RedFlag_Swing_oldold==1
&& RedFlag_Swing_oldoldold==1 && CAN_AugRotnCmdPcnt_out_old==253 &&
stereo_v_ultra~=2 && Swing_MOC==1
        if relx_desired_cells_prime>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
        end
    elseif RedFlag_Swing==1 && RedFlag_Swing_old==1 &&
RedFlag_Swing_oldold==1 && RedFlag_Swing_oldoldold==1 &&
CAN_AugRotnCmdPcnt_out_old==253 && stereo_v_ultra==2 && Swing_MOC==1
        %relx_desired_cm for ci-only
        if relx_desired_cm>0
            CAN_AugRotnCmdPcnt_out=uint8(251); %swing out
        else
            CAN_AugRotnCmdPcnt_out=uint8(252); %swing in
```

```
        end
    else
        CAN_AugRotnCmdPcnt_out=uint8(CAN_AugRotnCmdPcnt);
    end
end
```

Diagnostics

       Diagnostics are an important tool for model development. Without diagnostics, debugging is a guess-and-check process and very ineffective. The SUC model outputs a 1x70 diagnostic array. All of the array elements must be in the unsigned 8-bit data-type. CI had 19 elements in the array for internal diagnostics. The variables were selected to make troubleshooting straightforward; the procedure used made narrowing down the source of the problem very efficient.

CI-only Mode

       A feature was developed to allow CI to run on its own within the SUC model. This allowed for rapid iterative development and testing without inputs from FS or running grain (especially for tuning control functions). The feature is enabled with a parameter switch (enabled=1, disabled=0). When enabled, the FS requests are ignored and a manual input for 'desired column' is referenced when determining the required shift magnitude and direction.

Results & Conclusions

       CI was field tested and proven to effectively control auger swing and engagement to meet the requests of FS and prevent spillage during an automated unloading event. The dynamic plot below shows swing response to a desired shift. At the end of the plot, the auger angle increases to the maximum; that is the auto-swing-out feature suggested by expert operators.
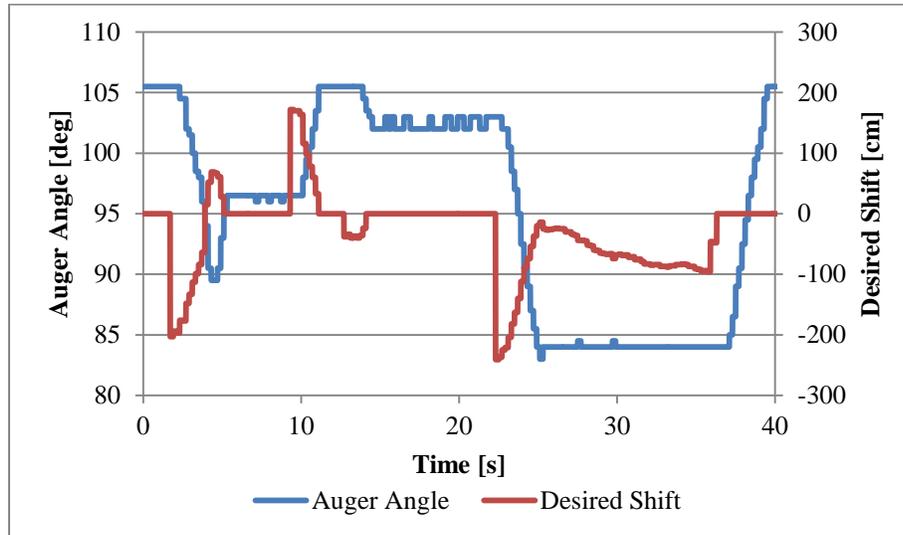
A sensitivity analysis was completed to quantify the effect of several input parameters on boot placement accuracy. The sensitivity analysis was a multi-step process: identify sources of error, vary the inputs and record model response, and quantify the effect of input error on output calculations. The sources of error identified were auger-angle resolution error, auger-angle sensor hysteresis error, out-of-calibration error, calibration-fit error, linear cosine approximation error, and auger length error. The depiction below shows the primary operations and paths of error propagation.
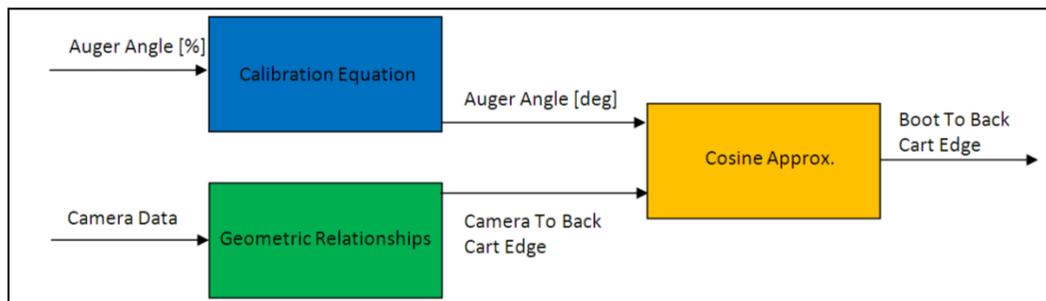
Resolution error was ±0.4 degrees, which corresponds to an average of ±5.45cm at the boot. No hysteresis error was apparent when plotting multiple swing-in and swing-out profiles on top of each other. Multiple times throughout this project, out-of-calibration error has been observed and can have a big impact on the auger angle calculation and boot placement accuracy (shown in the plot below, assuming a 6.9m auger length); auger angle sensor modifications were typically the cause of these errors and should require a recalibration in the future.

Boot Location Error
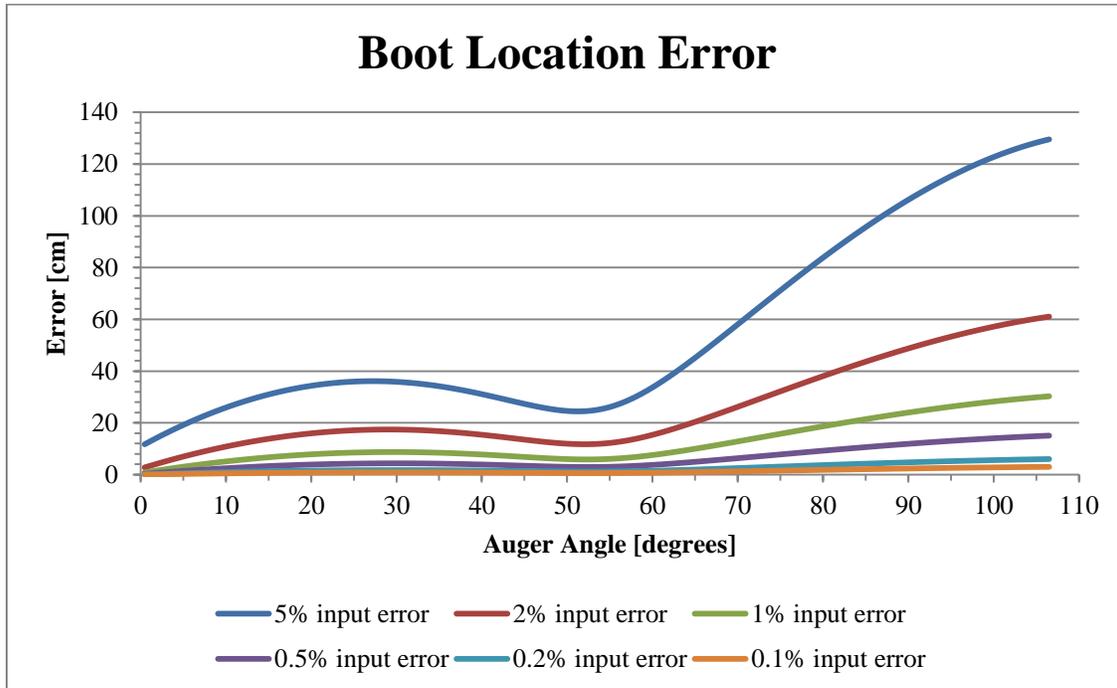
The calculation of auger angle from the 0-250 input has quantization error due to the integer math and fit limitations; however, the error for this system is small (less than 0.2 degrees). The error induced by the cosine approximation was less than 3cm over the operating range of SmartUnload. The boot location error in the x-direction resulting from auger length error is dependent on auger angle; at an auger angle of 105 degrees, an auger length error of 30cm will cause a boot location error of 8cm. The tornado plot below summarizes the sensitivity analysis using median values for each error type. Out-of-calibration error was found to have the greatest risk; other sources of error are small relative to the application and often partially cancel each other out.