AUTOMATIC GRAIN UNLOADING FOR CROP HARVEST MACHINE

by

Ziping Liu

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Mechanical Engineering

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Gregory M. Shaver, Co-chair

School of Mechanical Engineering

Dr. John T. Evans IV, Co-chair

School of Agricultural & Biological Engineering

Dr. Andrea Vacca

School of Mechanical Engineering

Dr. Daniel A. DeLaurentis

School of Aeronautics and Astronautics

Dr. John H. Lumkes

School of Agricultural & Biological Engineering

Approved by:

Dr. Nicole L. Key

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my advisors, Professor Gregory Shaver, and Professor John Evans. They provide me with valuable advice for both academics and life, continuous support and warm encouragement. I truly appreciate them allowing me to work on different roles in an interesting project. I learned so much from them which I could have never imagined before I joined the group. I am also thankful to Professor Daniel Delaurentis, Professor John Lumkes, and Professor Andrea Vacca, who serve on my dissertation committee and gave me brilliant feedback.

The work in this thesis was sponsored by John Deere. I would like to thank my contact at Deere for supporting the research, especially Corwin Puryk, Mark Sahlin, Ryan White, Mark Chaney, Brian Gilmore, Brandon MacDonald, Adam Royal, Bradley Yanke, Stephen Corban, Eric Anderson, Emily Horn, Randy Sergesketter, and Craig Amann.

I am so fortunate to work with many amazing people at the Herrick Labs, including Dr. Cody Allen, who is an extraordinary team leader guiding me and teaching me from my very first day in the group; Shveta Dhamankar, a diligent engineer, and always dependable friend; Dr. Xu Zhang, who is not only a brilliant researcher but also a great friend to enjoy the afternoon break with; Weijin Qiu, who can always share his innovative ideas and stimulate delightful conversation. Although I did not work in the test cell, but I did have great fun when chatting with the engine gurus in our group: Mrunal Joshi, Chisom Emegoakor, Shubham Agnihotri, Vrushali Deshmukh, Devarshi Patel, Adil Shaikh, and Doni Thomas. Additionally, I truly enjoyed my learning experiences with Mile Droege, Harsha Rayasam, Brady Black, Ifeoluwa Ibitayo, John Foster, Shubham Ashta, Tyler Swedes, Michael Anthony, Yunpeng Xu, Aishwarya Ponkshe, Zar Ahmad, Evan Parshall, and Raghav Kakani.

Besides my advisors and labmates, this thesis would not have been possible without the wonderful colleagues at Purdue who participate in the research project. I would like to express my sincere gratitude to Professor Tony Vyn, Rachel Stevens, Aaron Etienne, Logan Heusinger, Jim Beaty, Gautham Vinod, Varun Sudarsanan, Eric Kong, and Cesare Guariniello.

Lastly, and most importantly, I am deeply grateful to my family. My mom Kui and my dad Yueming are always backing me up, providing me with unconditional love and unparalleled support. I am forever indebted to them for giving me the experiences that had made me who I am. I am so thankful that I could meet, get to know, and marry my Wife Chufan at Purdue. Thank you for sharing my happiness and sorrow. Thank you for being my muse, my teammate, and most of all, my best friend. Thank you for always being by my side.

Thank you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AU	Automatic unloading
B2F	Back to front
CAD	Computer-aided design
CAN	Controller Area Network
CPU	Central processing unit
DEM	Discrete element method
EKF	Extended Kalman filter
EnKF	Ensemble Kalman filter
F2B	Front to back
F2B2F	Front to back to front
FOV	Field of view
ECU	Electronic control unit
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HIL	Hardware-in-the-loop
IP	Internet protocol
IPM	Image Processing Module
IVT	Infinitely variable transmission
KF	Kalman filter
LiDAR	Light detection and ranging
MAE	Mean absolute error
MS	Machine Sync
PST	Power shift transmission
RCP	Rapid controller prototyping
RGB	Red Green Blue
UDP	User Datagram Protocol
UKF	Unscented Kalman filter
UI	User interface

SIL	Software-in-the-loop
SISO	single-input-single-output
SPFH	Self propelled forage harvester

ABSTRACT

The world is facing a higher demand for food as the population is expected to grow to 9.1 billion by 2050, but the expected growth of arable land is much slower. In the meantime, the US has seen farm labor shortages for many years. These trends indicate the need for improving agricultural productivity while lowering the demand for skilled labor for farm operations. Automation of agricultural operation is one approach to achieve these goals. An automated unloading system is desirable as it can improve productivity and reduce the requirement for high-skill labor by lowering the complexity of the unloading on the go operation.

Agricultural machinery companies have developed various products to automate or assist parts of the unloading operations. Some researchers built unloading automation systems, but the limited performance, strict constraints, and the high cost curb their impact on productivity improvement or adaption for commercialization. Additionally, several companies have released product to automate the forage harvester unloading. However, no existing system can fully automate the combine harvester unloading on the go. Therefore, a system was proposed to automate combine harvester unloading on the go by automatically monitoring grain fill status, determining preferred auger location to fulfill prescribed fill strategy, and controlling the auger operation and location to achieve the desired fill.

An automatic unloading strategy for grain unloading automation was developed. The automatic unloading system is built by integrating a controller and a perception system to the combine harvester with an existing vehicle guidance technology, Machine Sync. Machine Sync is used to control the combine-tractor relative position by automatically changing the speed and moving direction of the tractor.

To develop the automatic unloading system, simulation tools were built to model the unloading on the go process and validate the model accuracy with in-field testing. The tools include:

- A grain fill model to simulate how grain pile up in a container such as grain cart or wagon given the grain unloading location and unloading rate. A grain fill model

benchmark system was built with LiDAR and validated that the grain fill model can achieve an accuracy of 0.2 m during a static grain cart unloading.

- A vehicle dynamics model to simulate the dynamics of the relative position between the tractor and the combine harvester. The relative motion between the combine and the tractor controlled by Machine Sync was treated as an aggregated system. To characterize the dynamics of the aggregated system, the instrumental variable approach was used to identify the model parameter based on black-box model simulation results. After that, a testing pipeline was developed to validate and refine the model parameters with in-field testing.

- A perception model to simulate the raw data of the perception sensors (i.e., stereo camera) during unloading with different lighting conditions, vehicle configurations, and sensor properties. To validate the perception model, stereo camera data were collected during automatic unloading in some typical conditions and compared them with the simulation results.

The simulation tools together build a virtual environment to simulate the unloading process. Based on these tools, the automatic unloading controller was developed. The controller automatically determines the desired auger location to fill the grain cart based on the current filling status and prescribed fill strategy. The controller also includes a closed-loop movement controller synthesized with H_∞ mixed sensitivity loop shaping that closes the loop around Machine Sync to enhance its tracking performance and robustness. After that, the automatic unloading system was validated in the virtual environment.

After validating the automatic unloading system in simulation, the automatic unloading system was implemented in hardware and a camera-based perception system (IPM) was integrated to monitor the unloading status. In-field testing demonstrated that the automatic offloading system can effectively automate the unloading-on-the-go of a combine harvester to fill a grain cart to the desired level under nominal harvesting conditions. The achievable fill level for a 1000-bushel grain cart without spillage ranges from -0.7 m to -0.2 m for the near-edge grain height relative to cart edge.

The in-field testing shows that the camera-based perception system (IPM), which is susceptible to environmental changes, can lose track of the grain cart and cause the automatic unloading to stop. To make the grain perception more robust to different lighting conditions, a fusion algorithm was developed by leveraging both the IPM and grain fill model. In-field testing data demonstrate that the fusion result can achieve higher accuracy, greater coverage, and better robustness than either the IPM or the grain fill model alone.

1. INTRODUCTION

The Food and Agriculture Organization (FAO) of the United Nations has reported that the world is facing a 70 % higher demand for food as the population is expected to grow to 9.1 billion by 2050 [1]. However, this staggering growth in demand is counter to the available arable land. Approximately 2-5 million hectares of global arable land is lost every year to soil erosion, and 3 million ha is lost annually as a result of severe land degradation [2]. In the US, the major crop yields are expected to decline because of increasing temperature, water availability, soil erosion, disease, and pest outbreak [3]. Moreover, the availability of skilled labor in the agriculture industry in the US has been declining for decades and the US has seen farm labor shortages for many years [4], [5].

These trends altogether reveal the importance of improving productivity while reducing the requirements for skilled labor in farming operations. Agricultural machinery feature automation is a key step to achieve these goals as proposed in "Agriculture 4.0" [6], [7].

Grain harvest involves coordination between multiple machines and is one of the most time sensitive operations. Unloading is the process of transferring grain from the on-board grain hopper of a combine harvester to a tractor-towed grain cart, ideally as the combine continues to move while harvesting grain. Grain unloading-on-the-go is a desirable operation that improves productivity, but requires skilled labor and attentive operation throughout a day of grain harvesting. During unloading-on-the-go, the combine continues to harvest while unloading grain to a grain cart moving alongside, allowing the combine to reduce unproductive time [8]. Delchev et. al. showed that on-the-go unloading can provide up to 30% time reduction per unit area in harvesting [9]. However, unloading-on-the-go also requires high-skill labor for both combine and tractor drivers. In particular, the combine operator must carry out multiple tasks simultaneously:

1. Monitor the filling status inside the grain cart
2. Determine the appropriate location for the auger to unload
3. Communicate with the tractor operator to move the auger to the desired location by changing the speed and heading of both vehicles

4. Watch for the clearance between two vehicles and react to obstacles, terrain change, and waterways in the traveling path

5. Monitor crop harvesting conditions and accordingly adjust harvester settings or vehicle movement

A system to automate the unloading process can unburden the combine operator of tasks 1-3. Considering the productivity and labor impact of the unloading process, a system to fully automate the unloading-on-the-go process can help to:

- **Improve operator experience and performance.** The system will improve user experience, and reduce driver fatigue during operation. Furthermore, [10] demonstrated that the automation on agricultural machinery can improve the driving performance.

- **Lower the demand for skilled labor.** The system can help to offload the combine operator from the first three tasks in unloading and thus significantly reduce the unloading operation complexity. In addition, the system can also help to coordinate the tractor movement with combine and thus reduce the skill requirements for the tractor driver.

- **Improve productivity.** The system could extend the running window of the harvester during the tight harvest season by making it possible to continue unloading during challenging scenarios (e.g., night time), and as a result of a reduction in operator fatigue. In the meantime, consistently monitoring the system can help to reduce the chances of spillage during unloading, one source of harvest loss.

To assist operators during unloading-on-the-go, agricultural machine manufacturers and researchers have proposed or developed various operator-assisting technologies.

Ag Leader built a CartACE system to simplify the unloading-on-the-go operation from the perspective of the grain cart operator. Leveraging wireless communication and data sharing between the combine and the tractor, this system automatically designs the tractor trajectory and controls the steering to move the tractor-towed grain cart alongside the combine harvester when the combine needs to unload. To build this system, a trajectory design

method was introduced [11] and a method to improve the smoothness of the articulated system, i.e., the tractor-grain cart pair, was demonstrated in [12]. In [13], a method was proposed to allow the grain cart driver to know in real-time how full the combine hopper is, so the tractor operator can prepare to engage in unloading operation in advance. Such a system can significantly reduce the workload and task complexity of the grain cart operation, and thus reduce the requirement for highly skill labor during unloading-on-the-go. However, in this system, the combine operator still has to monitor the filling status and communicate with the tractor driver to adjust the relative vehicle location to complete a desired fill. As a result, the CartACE system does not help address the main operational complexity that the combine driver is facing.

John Deere launched the Machine Sync system in 2011. Machine Sync is a guidance technology that enables sharing harvest and guidance information across multiple John Deere Machines. During unloading operation, the Machine Sync system allows the combine operator to command the relative position between the tractor and the combine via home point setting and adjust the relative position via nudge commands. Machine Sync system automatically controls the speed and steering of the tractor to move it to the commanded location. The method connects two agricultural machines with a leader-follower model in [14], and the system acquires and controls the relative position between two agricultural machines [15], [16]. Machine Sync has further been shown to allow simplification of the unloading process in [17]. Compared with CartACE, the Machine Sync system reduces both the workload of the tractor operator and the combine operator by reducing the need to communicate with the tractor operator, since the combine operator can adjust the position of the tractor directly. However, to achieve the desired fill, the combine operator is still required to monitor the grain filling status, decide the auger on/off, and command nudges to the tractor manually.

Case IH proposed a guidance system called V2V with a similar leader-follower concept as that of Machine Sync to allow the combine operator to control the speed and position of the tractor towing a grain cart and facilitate unloading-on-the-go. [18], [19] demonstrate the details on Case IH's ideas on coordinating multiple agricultural vehicles. However, since V2V was announced in 2011, the system has not been commercially available.

Kinze proposed a driverless tractor in its Autonomous Harvest system project [20]. SmartAg (now part of Raven Industries) announced a similar product called AutoCart. Neither system is available commercially. According to [21], the AutoCart system does not require a driver in the tractor cab. For unloading, the combine operator can call the tractor that tows a grain cart from another location in the field, and the tractor can automatically plan its path to follow the combine. When it starts following the combine, the combine operator can start unloading to the grain cart and nudge the tractor location to achieve the desired fill. However, during the unloading operation the AutoCart system still requires the combine operator to monitor the grain filling status, make decisions about the auger movement, and manually command nudges to the tractor.

Among the three combine operator tasks related to grain unloading during unloading-on-the-go, the above driver assisting technology developed by agricultural companies can only help to reduce the operator workload in task 3, but they still require the operator to execute task 1-2 manually. Consequently, the unloading-on-the-go remains a very challenging operation for the combine driver. To further reduce the unloading workload, some researchers have proposed systems to automate the other two tasks.

At Kyoto University, Kurita et al. developed a combine harvester robot to automatically identify the grain container in the field with a stereo camera [22] and position the combine auger to the center of the grain container to unload grain [23]. The stereo camera in [22], [23] requires installing a planar marker on the grain container for cart recognition and auger positioning. To further improve the versatility of the unloading system, Cho et al. updated the perception system with a laser scanner and GNSS sensors to automatically identify the grain cart and position the auger without the need for planar markers [24]. In their system configuration, the combine harvester only unloads to a small grain container (length: 1.83 m, width: 1.3 m) so it does not change auger location based on profile change during unloading. As such, the method cannot be applied to filling up grain carts or wagons with larger capacity, which are widely used in the United States. Moreover, the system does not monitor the grain container fullness, so the combine operator has to start/stop the auger manually.

Researchers at Iowa State University collaborated with John Deere and the National Robotics Engineering Center at Carnegie Melon University to develop a SmartUnload system. The SmartUnload system used a perception system based on two stereo cameras to monitor the status of unloading. One stereo camera was on the combine body to locate the grain cart, and another on the combine auger to monitor filling status [25]. With the raw data from the perception system on filling status, Jennett developed a computation framework to augment the perception system data and determine the desired auger location based on certain fill strategies [26]. Finally, the desired auger location was achieved by controlling the combine auger angle with an on-off or proportional controller [27]. The SmartUnload system built a complete framework for automating the unloading operation and developed a system prototype. However, according to the assessment results in [27], the performance of this system for controlling fill level in a grain cart was limited because the perception system was inconsistent under different lighting conditions. In addition, the perception system required mounting fiducial markers on the grain cart to function properly, and used two stereo cameras. Furthermore, auger position control through auger rotation reduced the fillable area inside a cart to prevent spillage, which generally leads to a more conservative fill and reduced productivity in grain transfer logistics. Moreover, for a given relative location between the tractor and the combine, controlling the auger location by only swinging the auger limits unloading range inside the cart. Thus, while using a larger grain cart, the operator may still have to control the relative location between the two vehicles manually.

In addition to the aforementioned efforts to automate combine harvester unloading, automatic offloading has also been pursued for forage harvesters. Forage harvesters chop plants and unload the forage simultaneously to a trailer truck driving alongside. In a manual setup, the harvester operator adjusts the angle of the unloading spout with a hydro-handle to aim the spout at a proper location inside the forage container. John Deere developed an automatic offloading system called Active Fill Control with Carnegie Melon University. The Active fill Control system uses a stereo camera to monitor the filling status of the forage container [28]. Based on the fill status image, the Active Fill Control system automatically controls the angle of the unloading spout to execute the fill strategy specified by the opera-

tor [29]. The system displays the fill level indicator of the forage container overlayed with a camera image in real-time [30]. Claas developed a similar system called AutoFill to assist the forage harvester unloading by automatically aiming the auger spout to an empty portion of trailer [31]. Instead of using a stereo camera, the AutoFill system uses a triple-lens camera (a stereo camera plus a color camera) to measure the fill status and auger angle [32]. New Holland released IntelliFill system to automate the auger spout control for forage harvester unloading [33], [34] based on the fill status and the truck edges as detected by a combination of Time-of-flight camera and a stereo camera [35]. In these systems, because the range of spout angle adjustment is usually smaller than the trailer truck length, the forage harvester operator still needs to manually adjust the relative position between two vehicles to fill the whole truck.

Forage unloading and combine unloading have some significant differences that affect the potential of, and viable approaches to, automating the unload-on-the-go process:

1. The material harvested by a combine harvester usually has a higher value than forage, making spillage much more undesirable

2. The height difference between combine spout and grain cart edge is smaller, in comparison to forage harvesting, making it more challenging to find a good placement for the perception system sensor(s) to see the entire grain cart.

3. In practice, the auger position control in forage harvesters is usually achieved by adjusting both the auger spout angle and the relative position of the tractor. However, the auger position control in combine harvesters is mostly achieved by changing the relative position of the tractor.

In summary, an automated unloading-on-the-go system for combine harvesting will improve productivity and reduce the requirement for high-skilled labor by lowering the complexity of the unloading-on-the-go operation. Agricultural machinery companies have developed various products to automate or assist parts of the unloading operations. Some researchers have built combine harvester unloading-on-the-go automation systems, but their limited performance (e.g., tracking performance, fillable region in cart), strict constraints (e.g., grain

cart size, use of fiducial patterns), high cost, and the remaining burden on the combine operator curb their impact on productivity improvement or adaption for commercialization. Several companies have released products to automate the forage harvester unloading, a similar application to combine unloading. However, the difference between forage harvester and combine harvester unloading system design is still too large to allow direct the replication of forage harvester technology. To the best of our knowledge, no existing system can fully automate the unloading-on-the-go operation of a combine harvester. A system to accomplish this is outlined and demonstrated in this paper. This automatic offloading system automatically monitors grain fill status, determines preferred auger location to achieve prescribed fill strategy, and controls the auger status and location to achieve the desired fill with intervention, as required, from either the combine or the tractor operator.

In Ch. 2 introduces the strategy to automate grain unloading and how the automatic unloading system was built. Chapter 3 details the simulation tools to enable designing and validating the automatic unloading system in a virtual environment. Based on the tools, Ch. 4 introduces how the automatic unloading controller was built and validated in the virtual environment. After validating the automatic unloading virtually, Ch. 5 shows the integration of the automatic unloading system in the machinery and in-field validation of the automatic unloading system. In Ch. 6, a data fusion strategy is developed to improve the robustness and accuracy of the grain profile perception.

2. GRAIN UNLOADING AUTOMATION STRATEGY

2.1 System architecture

The main objective of the automatic unloading project is to develop a system to automate grain unloading from a combine harvester to a grain cart towed by a tractor while harvesting. The automatic unloading (AU) system needs to automatically fill up the cart to a desired fill level following the fill strategy specified by the operator.

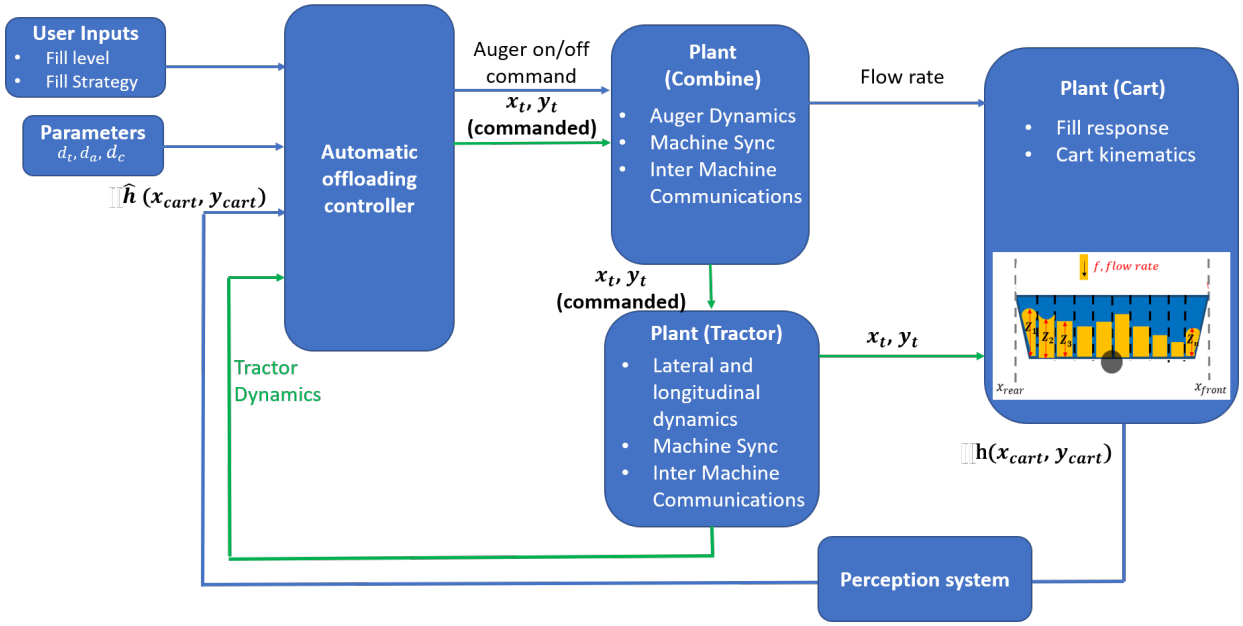


Figure 2.1. High-level automatic unloading system architecture

The high-level architecture is shown in Fig. 2.1 for the automatic unloading system. The automatic unloading system is built by integrating an automatic unloading controller and a perception system to the vehicles in unloading operation. The automatic unloading system takes advantages of Machine Sync, a machine-to-machine communication technology, running on current John Deere vehicles. One function of Machine Sync is to control the combine-tractor relative position by automatically changing the speed and moving direction of the tractor.

The automatic unloading controller runs on the combine harvester. It automatically calculates the desired vehicle locations and auger on/off status to achieve the target fill

specified by the operator based on the current grain fill profile inside the grain cart. After that, it sends the desired location and auger on/off command to the combine harvester. The auger on/off command controls the flow rate of the combine auger. In the meantime, the Machine Sync system will pass the desired location to tractor and controls the speed and steering of tractor to move it to the desired location. Ultimately, the tractor location determines the combine auger location relative to the grain cart, and together with the auger flow rate, affects how grain pile in the grain cart. Finally, the perception system measures the grain profile inside the grain cart and provides the fill level feedback to the AU controller, closing the automatic unloading control loop.

To develop the automatic unloading system presented in Fig. 2.1, four tasks were accomplished in this research:

1. Developed simulation tools to model the unloading on the go process and validated the model with in-field testing

2. Designed an automatic unloading controller and validate the controller with the simulation tools

3. Implemented and validated the automatic unloading system in hardware

4. Developed a fusion algorithm for camera-based perception and grain fill model to improve the quality and robustness of grain profile measurement

Chapter 3 shows the first stage of the automatic unloading system development, building simulation tools for automatic unloading and validating the tools with in-field testing. Tools were developed to simulate the major components in the automatic unloading system: the grain fill model, the vehicle dynamics model, and the perception model.

The grain fill model simulates how grain piles up in a container such as grain cart or wagon with given grain unloading location and unloading rate. The grain fill model can not only be used for modeling the unloading operation in automatic unloading simulation but also serve as a real-time grain profile estimator in the automatic unloading controller. Traditionally, the simulation of granular material such as grain or sands can be simulated with discrete element method (DEM) to capture the details of the grain flowing behavior of

individual particles. However, DEM method takes a long time to simulate especially in large scale, so it is not suitable to apply in the AU controller. Besides, the level of details in DEM may not be necessary for unloading application. Therefore, this paper proposed to simulate the piling dynamics based on the geometric constraint of granular material to achieve fast computation. The proposed method can simulate the grain fill process in a grain cart in a significantly shorter time than the physical process. Moreover, an experimental benchmark system was developed with LiDAR to validate the accuracy of the grain fill model. According to the experimental validation, the grain fill model achieves ± 0.2 m accuracy in most of the grids inside the container.

Vehicle dynamics model simulates the dynamics of the relative position between the tractor and the combine harvester, which ultimately leads to the change of auger location relative to the grain cart. In this vehicle system, Machine Sync is used as the low-level controller to control the the speed and moving direction of the tractor to achieve the desired relative location. Therefore, instead of building separate models to describe the detailed dynamics of the tractor plant and combine plant, the relative motion between the combine and the tractor controlled by Machine Sync was treated as an aggregated system. As a result, the aggregated Machine Sync - tractor - combine plant takes the input as commanded location while outputs the actual relative position. To characterize the dynamics of the MS-tractor-combine plant, a data-driven system ID method was used to identify the model parameter via instrumental variable approach. In simulation, the plant parameters from the output of the black-box Machine Sync model were identified and the first pass of vehicle dynamics model was built. After that, an experimental system was developed to collect vehicle dynamics data from vehicles running Machine Sync system and built a data processing pipeline to further refine the parameters in vehicles dynamics model based on experimental data.

Perception model simulates the raw data of the perception sensors (i.e., stereo camera) during unloading operation. Modern perception systems usually use Machine learning algorithms to process the raw data from perception sensors. However, Machine learning algorithms require enormous amount of labeled data to train and validate the algorithms. Traditionally, the data are collected from in-field testing, which is costly and slow. More-

over, to ensure the robustness of the algorithms, the data needs to cover various edge cases, including different vehicle configurations and operating conditions, making the data collection from in-field testing more challenging. To generate large amount of sensor raw data for algorithm development and validation, a perception model was developed to simulate the sensor response during unloading on the go. The perception model can simulate both a camera sensor or a LiDAR sensor. For camera sensor, it can simulate different lighting conditions, vehicle configurations and sensor properties. Besides generating raw data, the perception model was also used to recommend stereo camera placement and proposed an algorithm validation pipeline with Software-in-the-loop (SIL) perception model simulation. Finally, the unloading-on-the-go testing was conducted with stereo camera collecting data in several typical conditions to validate the perception model simulation results.

The simulation tools in Ch. 3 together build a virtual environment to simulate the unloading process. Based on these tools, an automatic unloading controller was developed and its performance was validated in this virtual environment in Ch. 4.

The automatic unloading controller works in two stages:

1. **Control strategy**: In the first stage, The controller divides the grain cart into several sections and determines the fullness in each section based on the grain fill profile feedback and the fill target specified by the operator. After that the Control strategy automatically determines the desired section that the auger should move to according to the user preference. It can fill the cart from front to back (F2B) or from back to front (B2F) or a combination of F2B and B2F.

2. **Movement controller**: In the next stage, the movement controller translates the desired location into specific commands to the vehicles. two movement control frameworks were developed: an open-loop framework that solely relies on Machine Sync system to move auger to the desired location and a closed-loop framework that close the loop around Machine Sync to further improve the tracking performance and the robustness of the automatic unloading system. The Machine Sync system control was linearized to a classic single-input-single-output (SISO) linear system control problem

and applied H_∞ mixed sensitivity loop shaping to synthesize the closed-loop controller. Then, both controllers were evaluated in sub-system testing.

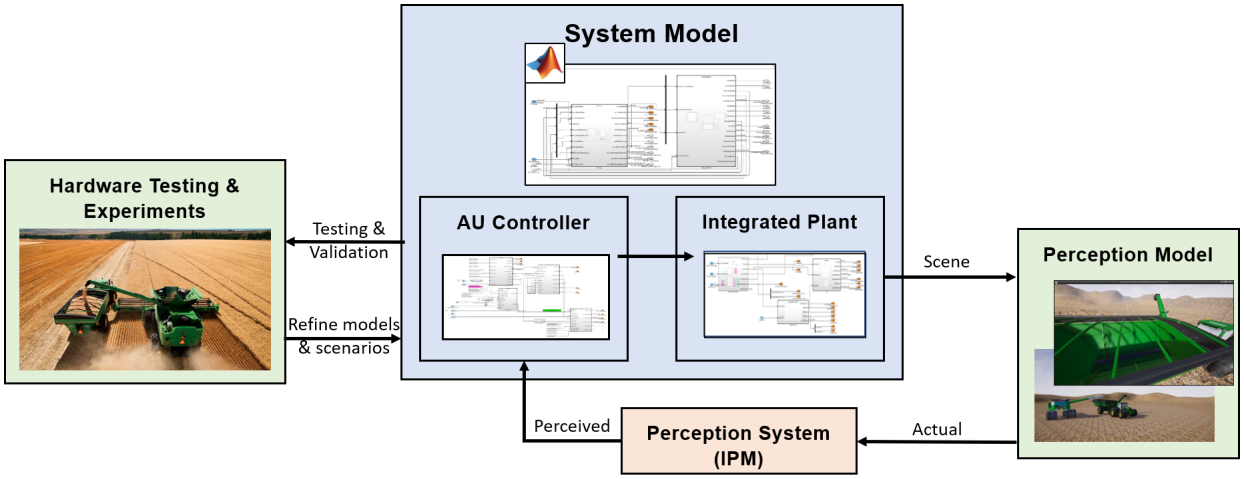


Figure 2.2. Grain unloading automation modeling strategy

After designing the controller and building all the simulation tools, the tools and the controller were integrated to run the full automatic unloading simulation to validate the system performance. A co-simulation framework was proposed as shown in Fig. 2.2 that consists of three major components:

- A system model in MATLAB Simulink that simulates the dynamics of the automatic unloading (AU) controller as well as the response of plants involved in the unloading processing including a combine harvester, a tractor, and a grain cart towed by the tractor.

- A perception model in Unreal engine to visualize the unloading scene in the virtual 3D world so that it can generate raw data for vision sensors.

- A perception system to measure the unloading scenes with vision sensors and analyze the sensor data to provide unloading status messages to the AU controller. In this application, a stereo camera-based Image Processing Module (IPM) developed by John Deere and National Robotics Engineering Center was used as the perception system.

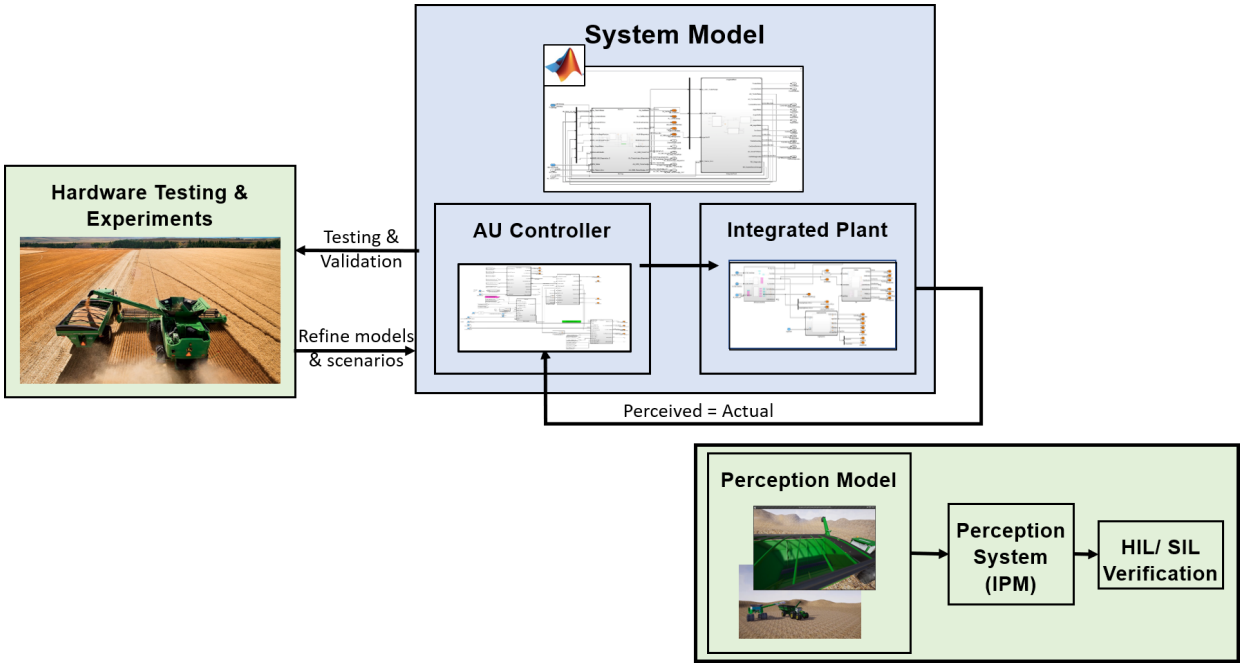


Figure 2.3. Alternative grain unloading automation modelling strategy

The co-simulation framework is advantageous when demonstrating the overall system performance. However, simulation and validation of each individual components in the system are still required to iterate the simulation tools. To enable simulating unloading process for each component while validating the effectiveness of the controller, the modeling framework from Fig. 2.2 was adapted to Fig. 2.3. In the system model, the perceived status by the AU controller will be the ground-truth simulated by the plant models. Meanwhile, the perception model simulates pre-defined scenarios and outputs the sensor raw data to perception system. Since the scenario is pre-defined, one can compare the output from perception system and the scenario definition for hardware-in-the-loop (HIL) or software-in-the-loop (SIL) algorithm verification.

The results from system model simulation shown in sec. 4.2 demonstrates that with a good fill profile feedback, the AU controller can automatically unload the grain from a combine harvester to a grain cart to a desired fill level without manual intervention. Additionally, the simulation shows that the controller works with different fill strategies and initial grain profiles in the cart.

After validating the automatic unloading system in simulation, Ch. 5 shows the process to implement and test the automatic unloading system in hardware. The camera-based perception system (IPM) was integrated to the combine harvester to monitor the grain fill status. Based on the characteristics of the grain profile feedback from the IPM, algorithms were developed to augment and utilize its data. Finally, the in-field testing demonstrates the effectiveness of the automatic unloading on controlling the vehicle location and auger on/off to achieve a desirable fill with different fill strategies in a nominal environment.

However, poor lighting conditions or algorithm error could cause the camera-based perception system (IPM) to lose track of the grain cart intermittently. Sometimes the automatic unloading system has to halt operation as a result. To make the perception more robust and improve the perception accuracy, a data fusion algorithm for the IPM data and the grain fill model was developed and validated in Ch. 6. Data from the in-field testing demonstrate that the perception results from the fusion algorithm have better accuracy, greater coverage, and are more robust to environmental changes compared with either the IPM or the grain fill model alone.

2.2 Assumptions and uncertainties

Several major assumptions were made when designing the entire automatic unloading system:

- The unloading is performed when vehicles are moving in a straight line on a flat terrain
- The combine harvester cruises at a constant speed during harvesting

When modeling the unloading system, the following assumptions were made:

- Grain fill model
  - The angle of repose is constant during unloading because grain properties do not change in such a short period of time
  - The vehicles move on a flat terrain
  - The grain cart's jostling magnitude does not change during unloading

- Vehicle dynamics model
  - Combine cruises at constant speed during unloading
  - The latency of wireless transmission for Machine Sync is negligible
- Perception model
  - Because the lens distortion is rectified before entering the image processing pipeline, the camera model only simulates the rectified image without lens distortion
  - The grain bed inside the cart can be modeled as a continuous surface
  - The distance error of a LiDAR sensor is normally distributed

To augment the perception results for grain profile, an algorithm was proposed to fuse the data from the camera-based perception system (IPM) and the grain fill model. The algorithm works with the following assumptions that are based on the observed characteristics of the IPM:

- IPM provides accurate grain height profile measurement when the auger is off
- The overall height of the measured profile (average of available grids) is reliable even when the auger is on

The uncertainties of the system are important for controller design. When designing the controller, the following system uncertainties were considered:

- The grain profile measurement from the stereo camera-based perception system. The height measurement from the perception system is provided in the form of a 2D heightmap. Because of the error from the camera sensor and depth estimation, the height measurement can have an error of more than 15 cm. To alleviate the measurement uncertainty, the cart fullness of each row was estimated by the average grain height both spatially (with neighboring grids) and temporally (with prior measurement).

- The tracking performance of Machine Sync. In-field testing has demonstrated that the Machine Sync has nearly no steady-state error in nominal conditions. However, the smallest nudge of Machine Sync is 6 inches, so the tracking has a 6-in discretization error.

- Machine Sync dynamics uncertainty. In-field testing shows that Machine Sync dynamics are significantly different based on the nudge size. To deal with this uncertainty, in controller design, the stability and closed-loop system performance were evaluated for both small-nudge dynamics and large-nudge dynamics.

- The relative location measurement between combine and harvester. This measurement comes from the GNSS locations of both machines. Because both machines use StarFire RTK GPS for localization with a quarter-inch accuracy. Its error is much smaller than the scale of unloading, so the localization error was not modeled in the virtual tools.

3. AUTOMATIC UNLOADING SIMULATION TOOLS DEVELOPMENT

3.1 Grain fill model

3.1.1 Introduction

The piling of granular materials has attracted much interest because of its unique properties [36], [37] and wide range of engineering applications including pharmaceutical (e.g., capsule, tablets solid processing), agriculture (e.g. seed processing, in-silo grain storage), construction (e.g., construction material storage), etc. Among the wide range of applications, some require real-time simulation of the granular material piling process on devices with low computing power. One particular application of interest is the automatic grain unloading-on-the-go from an agricultural combine harvester to a tractor-driven grain cart [38]. For this application, a model to simulate the grain piling process inside the grain cart is not only a crucial component for system-level simulation and validation, but also a potential means to provide real-time fill status estimation for the unloading controller

Many models have been developed to simulate the dynamics of granular flow. One example is the Discrete element method (DEM), which was first proposed by Cundall & Strack [39]. DEM-based methods simulate each particle in the granular material as an individual object. [40] proposed a computational framework to use DEM to simulate the piling of granular material. By simulating the dynamics of each individual particle in the granular material, the DEM-based method can reflect the details of the piling dynamics and achieve the highest level of accuracy. However, simulating each particle separately also results in a very high requirement for computation. The increasing number of particles further increases the computation time for DEM. Consequently, DEM is usually used for small-scale simulation [41]. Recent modeling technique advancement and the introduction of parallel programming have dramatically improved the computing speed of DEM [42] and have made it feasible to simulate larger-scale applications [43]. However, considering the number of grain particles in a typical grain cart, the DEM method is still too computational-

demanding for real-time piling computation for an application like combine harvester grain transport.

Another category for granular flow simulation is the continuum method. Continuum methods are usually categorized to 3 regimes: quasi-static [44], inertial [45], [46], and intermediate [47], [48]. The continuum method simulates the granular materials as continuous media and the time evolution of the material is governed by the mass and momentum conservation equations. As a result, continuum methods do not reflect the detailed motion of individual particles in the granular flow, but do usually have a faster computation compared to DEM. Some researchers have developed methods to simulate the grain piling process with continuum-based methods for computer graphics application that can simulate one frame in the scales of seconds depending on the size of simulation [49], [50]. However, these methods still require a high computing power machine and the simulation speed is not likely fast enough to be used as real-time feedback for controllers.

In summary, traditional granular flow models can generate accurate simulation results for the grain piling process, but they lack the real-time capacity, especially when running on a low computing power device like an embedded ECU. Moreover, for some applications, the details of how small particles interact and the detailed shape of the final piling profile may not be necessary – with example including estimation of the in-cart grain height profile during a combine harvester-to-grain-cart unloading process, prediction of a sand pile at a construction site, or the grain accumulation in a grain silo.

As such applications that require real-time computation without fine details of the piling dynamics and final shape, a computational-efficient geometric method could be preferred. For granular material, angle of repose is a widely-used geometric characteristic that can be measured [51], [52] and serve as a general principle to describe how grain piles [53].

Previous research efforts have experimentally identified the angle of repose of silage, and then simulated the silage pile in the trailer unloaded by a forage harvester by constraining the new pile shape with the identified angle of repose [54]. Experimental fill profile was measured from a series of 2D LASER scans and then compared with the simulated profile to validate the model accuracy. Unfortunately, the experimental results indicated that when the maximum height of the silage profile is about 2.5m, the maximum height error could be

as large as 1.13m. Such high maximum error are impractical to be used as a robust grain profile feedback for a real-time controller.

This paper will outline a novel piling model to predict the profile change and spillage during the unloading process. The model was developed and validated with corn piling inside a grain cart, but the same principle could be applied to similar granular material piling process (e.g., sands, soybeans) if computational efficiency is required. The approach is demonstrated through the simulation of the grain fill process from a combine harvester to a grain cart. On one hand, the model is based on the geometric characteristics of granular material rather than granular particle dynamics to dramatically reduce the computation complexity. On the other hand, the model also takes in to account more physical constraints on grain flow and cart jostling besides the angle of repose to enhance its accuracy. As shown in the following, the model can run in real-time while achieving 0.2-m estimation accuracy, in comparison to high fidelity LiDAR experimental data.

3.1.2 Principle

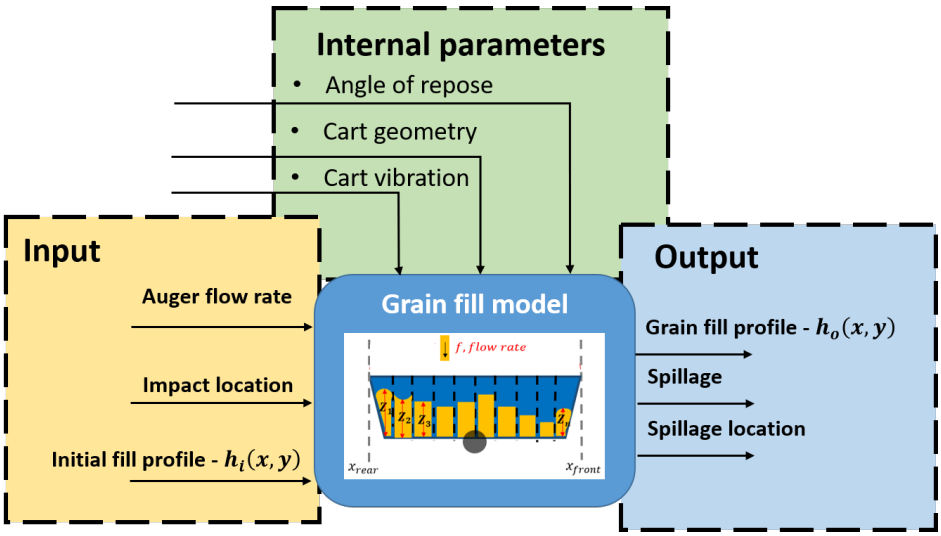


Figure 3.1. Grain fill model diagram

The granular material piling model is developed to simulate the grain fill process inside a grain cart during combine harvester grain unloading as shown in Fig. 2.1. During the

unloading, the combine harvester unloads grain to a tractor-driven grain car via an auger boot. Figure 3.1 summarizes the input, output, and internal parameters of the piling model. In the grain fill model, the grain profile is discretized as a 2D height matrix $h(x, y)$. At each time step, the grain fill model updates the grain profile $h(x, y)$ based on the flow rate of auger V and the impact location of grain $(x_{\mathrm{i}}, y_{\mathrm{i}})$ at the current time step. The grain fill models require the angle of repose α, cart geometry $h_{cart}(x, y)$ and the magnitude of cart vibration for the calculation.

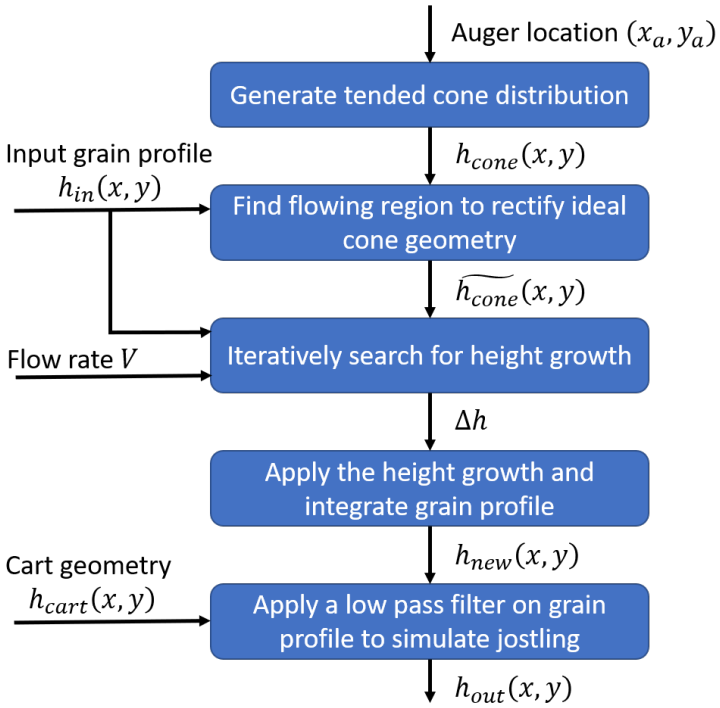


Figure 3.2. Grain fill model pipeline

Figure 3.2 illustrates the simulation pipeline of the grain fill model and Fig. 3.3 shows a 2D schematic for each step in the pipeline. The key grain characteristics used in this grain fill model is the angle of repose of the grain. The angle of repose α is the steepest angle relative to the horizontal plane that granular material such as grain can pile. The angle of repose of grain is different for different grain types and grain moisture. For instance, [55] lists some empirical angles of repose for common grain types in the United States.

During an unloading process, when bulk granular material are poured into a horizontal plane, the pile will have a conical shape and the angle relative to the horizontal plane of this

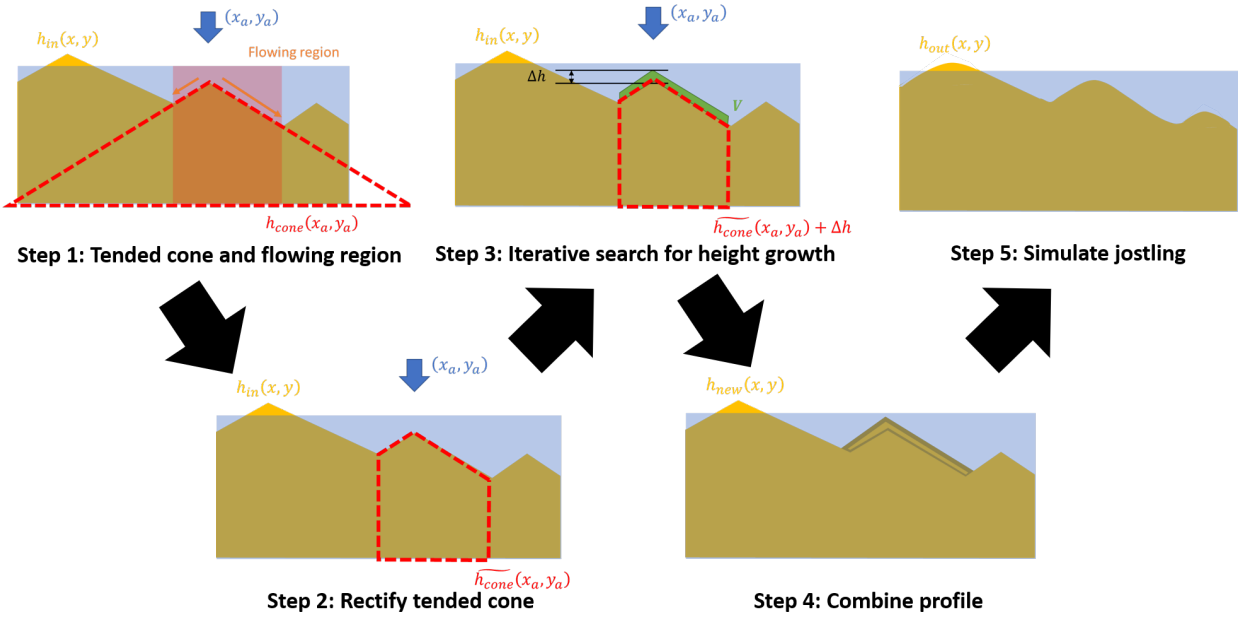


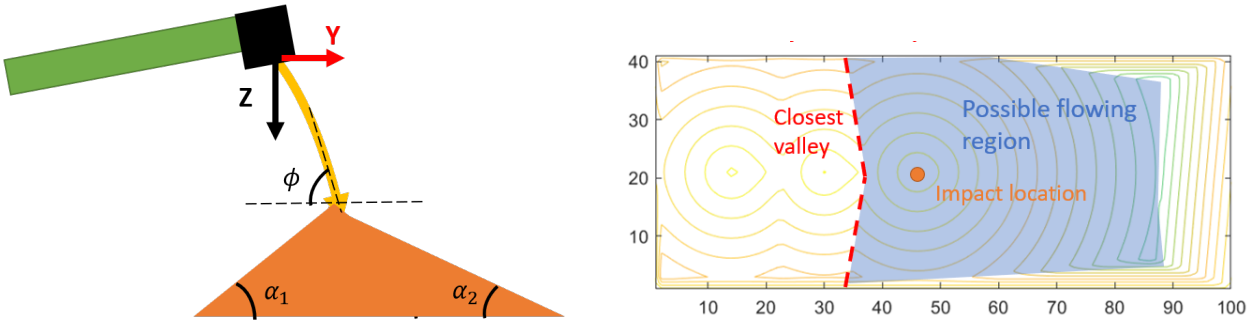
Figure 3.3. Grain fill model simulation 2D schematic

cone is also the angle of repose. With a given impact location, at the current time step, the grain pile that the incoming grain tends to form can be mathematically described as

$$h_{cone}\,(x, y) = h_{\text{in}}(x_{impact}, y_{impact}) - \frac{\sqrt{(x - x_{impact})^2 + (y - y_{impact})^2}}{\tan\alpha} \tag{3.1}$$

where h_{in} is the initial profile, (x_{impact}, y_{impact}) is the impact location, and α is the angle of repose.

In Eq. 3.1, the required input to generate the expected cone distribution is the impact location, not the combine harvester auger location. The impact location is not generally the same as the auger location because of the auger exit grain velocity of in horizontal direction and wind. For a combine harvester, the initial velocity of the grain flow in the lateral direction as depicted in Fig. 3.4a plays a significant role on the difference between the impact location and auger location. The trajectory of the grain flow is approximated



(a) (b)

Figure 3.4. Step 1 in grain fill model: generate the tended distribution and find the possible flowing region. (a) Impact of initial velocity; (b) Possible flowing region of the incoming grain labelled in a contour plot.

as a projectile motion in y-z plane. Therefore, the relation between the auger location and impact location can be described as

$$\begin{aligned} x_{impact} &= x_{auger} \\ y_{impact} &= y_{auger} + v_y t \\ h\left(x_{impact}, y_{imapct}\right) &= z_{auger} + v_z t + \tfrac{1}{2} g t^2 \end{aligned} \tag{3.2}$$

where (x_{impact}, y_{impact}) is the impact location, $(x_{auger}, y_{auger}, z_{auger})$ is the auger location in 3D space, (v_y, v_z) is the initial velocity of grain in the y-z plane, and g is the gravitational constant, and t is the time the grain take to travel from the auger boot to the grain pile.

Besides the impact location, the initial velocity of the grain flow will also change the angle in the expected cone distribution. For example, [54] reports that when the impact angle ϕ in Fig. 3.4a decreases, the angle of the cone on the back of the pile α_2 also decreases. By factoring initial velocity into the expected conical shape calculation, Eq. 3.1 is rewritten as

$$h_{cone}\left(x, y\right) = h_0 - \frac{\sqrt{\left(x - x_{\mathrm{i}}\right)^2 + \left(y - y_{\mathrm{i}}\right)^2}}{\tan\left[\lambda\left(x, y, \phi\right)\alpha\right]} \tag{3.3}$$

where the angle of cone, instead of being a constant, will be reduced by a factor of λ, which depends on the grain property and impact angle ϕ.

However, even though the grain tends to distribute as a conical shape like h_{cone} if the grain is poured on a horizontal plane, the actual distribution $\widetilde{h_{cone}}$ will be influenced by the initial grain profile h_{in}. One major influence from the initial profile to grain distribution is the region that the grain can possibly flow to. Generally speaking, the granular material flows from the higher region to the lower. As a result, there is a possible flowing region for the incoming grain bounded by the closest valley to the impact location in the initial profile, as shown in the example in Fig. 3.4b.

To find the possible flowing region algorithmically, the possible flowing region is solved from the seed point (impact location) with the downward flowing criteria, i.e., the grain will only flow to lower region. The first step is to discretize the grain profile as 2D grid $h_{\mathrm{in}}\left(x_{\mathrm{i}}, y_{\mathrm{j}}\right)$ bounded by the cart edge:

$$h_{\mathrm{in}}\left(x_{\mathrm{i}}, y_{\mathrm{j}}\right), \text{where} 1 \leq \mathrm{i} \leq \frac{L_{cart}}{d}; 1 \leq \mathrm{j} \leq \frac{W_{cart}}{d} \tag{3.4}$$

where i is the index of grid in the longitudinal direction of the cart, j is the index of grid in the lateral direction of the cart, L_{cart} is the length of cart, W_{cart} is the width of the cart, and d is the grid size.

The possible flowing region evolves inductively. Denote P_{i} as the possible flowing region set at step i. At each iteration, the algorithm looks at the boundary grids in the current region, denoted as B_{i}. For each grid $b_{\mathrm{i,j}}$ that belongs to the boundary B_{i}, it will look for its candidate neighboring grid set $C_{\mathrm{i,j}}$ that are not in P_{i} and has a lower height than the boundary grid height $h(b_{\mathrm{i,j}})$.

where $C_{\mathrm{i,j}}$ is the candidate set for possible flowing region derived from boundary point $b_{\mathrm{i,j}}$, $N\left(b_{\mathrm{i,j}}\right)$ is the neighboring grid to $b_{\mathrm{i,j}}$ and $h\left(b_{\mathrm{i,j}}\right)$ is the height of $b_{\mathrm{i,j}}$.

After getting the candidate sets from all the boundary points, the flowing region will grow by adding the new candidate sets.

$$P_{\mathrm{i+1}} = P_{\mathrm{i}} \cup \bigcup_{\mathrm{j}=1}^{N_B} C_{\mathrm{i,j}} \tag{3.5}$$

where N_B is the number of boundary points in P_{i}.

The iteration continues until there is no more candidate points $\bigcup_{\mathrm{j}=1}^{N_B} C_{\mathrm{i,j}} = \emptyset$

After determining the possible flowing region, as shown in Fig. 3.3, the tended cone distribution will be rectified as

$$\widetilde{h_{cone}}(x,y) = \begin{cases} h_{cone}(x,y), & if \quad (x,y) \in P_N \\ NaN, & Otherwise \end{cases} \tag{3.6}$$

where P_N is the possible flowing region, NaN denotes that a grid in which grain will not be added at this time step.

With the rectified tended grain distribution $\widetilde{h_{cone}}(x,y)$, the new grain profile with the additional grain from the current time step will be a combination of the initial profile and the tended distribution

$$h_{new}(x,y) = \begin{cases} \widetilde{h_{cone}}(x,y) + \Delta h, & if \quad \widetilde{h_{cone}}(x,y) + \Delta h > h_{in}(x,y) \\ h_{in}(x,y), & if \quad \widetilde{h_{cone}}(x,y) + \Delta h \leq h_{in}(x,y) \\ h_{in}(x,y), & if \quad \widetilde{h_{cone}}(x,y) = NaN \end{cases} \tag{3.7}$$

here Δh is the height growth results from the incoming grain at current time step. Δh can be solved iteratively by

$$\sum_{\mathrm{i}=1}^{L_{cart}/d} \sum_{\mathrm{j}=1}^{W_{cart}/d} h_{new}(x,y) - h_{in}(x,y) = \Delta V$$

where ΔV is the amount of grain unloaded into the container at a given time step.

Finally, if the cart is moving, cart jostling will shake down the grain and reduce the angle in the conical distribution. To simulate the jostling effect, for the grain portion in h_{new}, a low-pass Gaussian filter was applied to the grain profile

$$h_{out}(x,y) = h_{new}(x,y) \otimes G(x,y) \tag{3.8}$$

where $G(x, y)$ is a 2D Gaussian kernel that can be described as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.9}$$

here the standard deviation σ depends on the magnitude of the jostling.

The grain fill model can also be used to estimate spillage. To enable spillage calculation, the same framework is applied but the range of the grain profile definition is expanded as shown in Eq. 3.4. Instead of bounding by the edge of the grain container, the range of the profile grids will be extended with extra auxiliary grids, which represent the ground. Consequently, the spillage is the grain amount added to the auxiliary grids, and the container grids adjacent to these auxiliary grids are the spillage location.

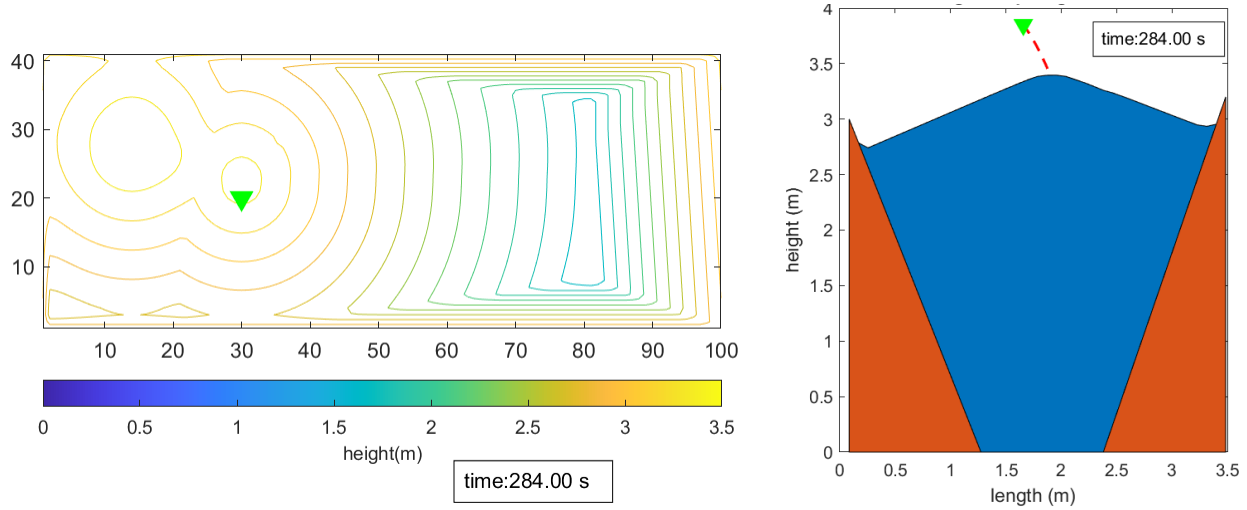
3.1.3 Simulation results

The granular material piling model was implemented in MATLAB r2019a on a Windows PC (CPU: Intel Core i7-8700, RAM: 16GB). A continuous unloading process from a combine harvester (model: John Deere S660) to a grain cart (model: Parker 1348) was simulated. The cart length is 8.5 m and the width is 3.5 m. The maximum auger flow rate is 3.8 bu/s. The auger boot is 3.85 m high relative to the bottom of the grain cart. The grain flow from the auger boot has an initial velocity of 1.5 m/s in the horizontal direction and 2 m/s downward in the vertical direction. The angle of repose of the grain is 23 degrees. The grid size of the grain profile grid is 8.5 cm. In the unloading process, the auger unloads from the front of the grain cart to the back. The auger stays at the front for an extended period of time to create a spillage scenario.

The simulated unloading process takes 384 seconds in the physical world, and the simulation time in the computer is 89 seconds with a simulation step size of 0.2 seconds. Because the simulation time is about 77% shorter than the time it takes in the physical world, the grain fill model can be used as a real-time estimator for an automated system from a computational efficiency perspective.

Visualization 1 and Visualization 2 show the simulation results from this unloading process. Figure 3.5 shows one representative frame from the simulation results. Figure. 3.5a is

the contour plot visualization, in which the green triangle marks the auger location at the current time step. The cone distribution of the grain is following the angle of repose from the impact location. Figure. 3.5b is the lateral cross-section at the auger location, in which the orange block labels the cart geometry, the blue block labels the grain, and the red dash line shows the simulated grain flow trajectory. The influence of the initial velocity of the grain flow can be examined clearly from the lateral cross-section: the impact location is farther away from the auger location, and the angle of the pile at the back is smaller than the one at the front.



(a) (b)

Figure 3.5. Grain fill model simulation results. (a) Contour plot of the grain profile (associated with Visualization 1); (b) Lateral cross-section of the grain profile at the auger location (associated with Visualization 2).

Figure 3.6 shows the percentage of spillage during this unloading process, demonstrating the spillage simulation capacity of the grain fill model. At around $t = 230s$, because the auger continues unloading at the front of the cart for an extended period of time, the spillage at the front of the cart edge increases the grain spillage percentage. After around $t = 250s$, the spillage percentage decreases because the auger moves to the back of the cart, and the following grain are unloaded into the grain cart.

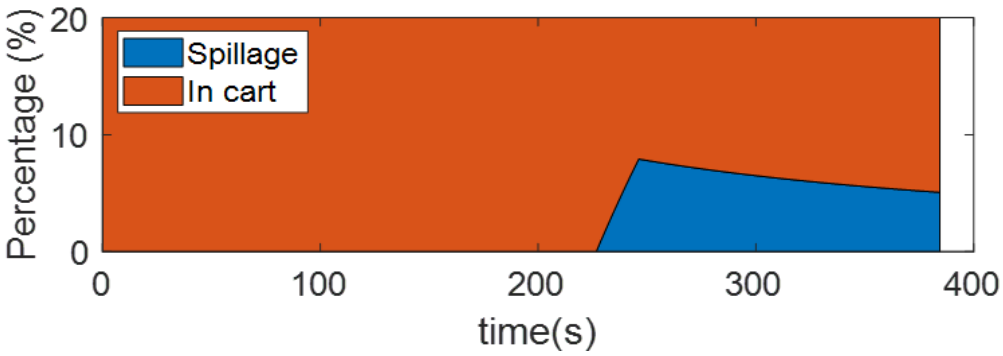


Figure 3.6. Spillage percentage of grain in grain fill model simulation.

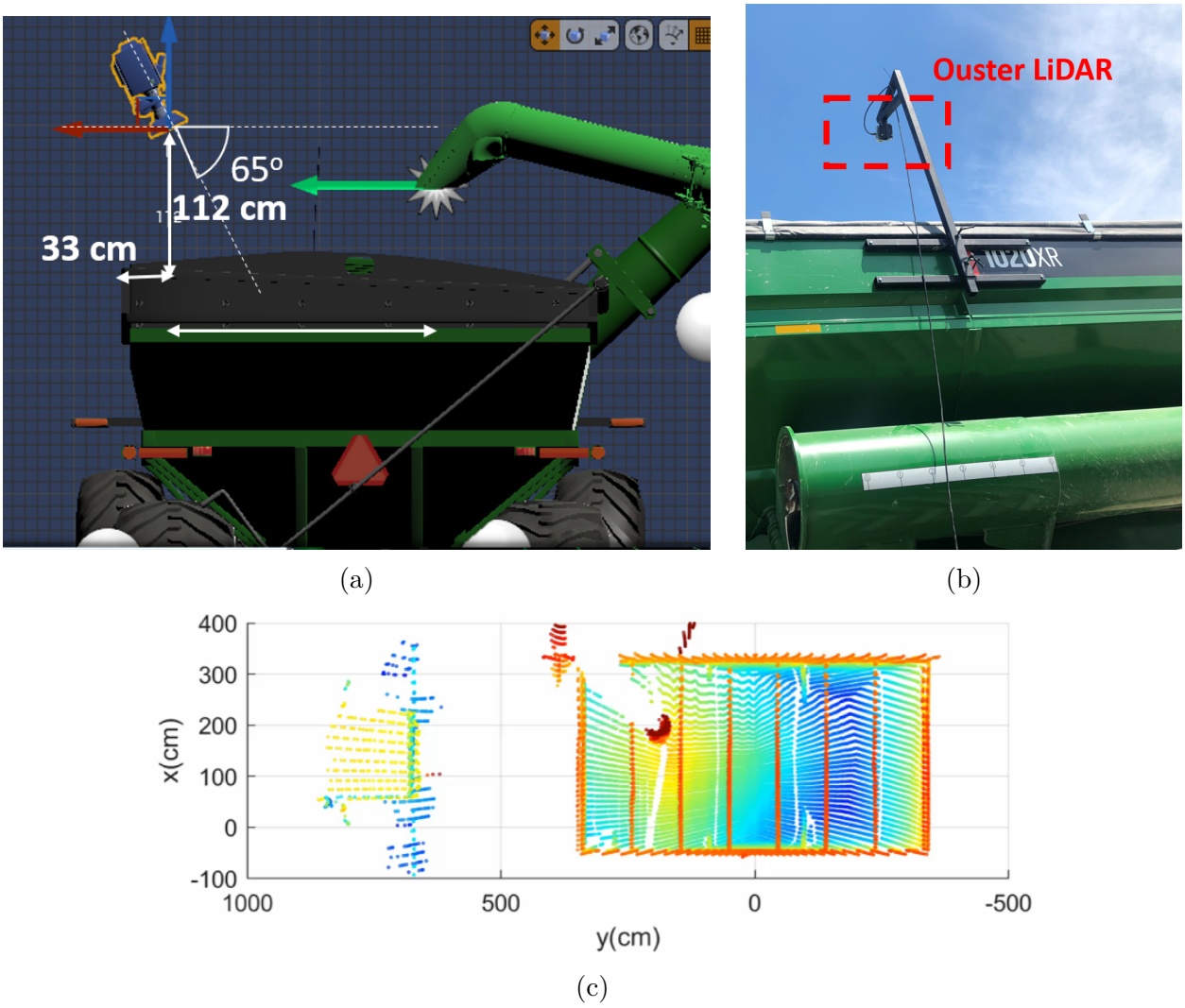
3.1.4 Benchmark system development

Grain fill model benchmark system

To validate the accuracy of the grain fill model, a benchmark system was built to measure the grain profile experimentally during unloading and the experimental results were compared with the grain piling model simulation results. The benchmark system was built around an Ouster OS0-64 LiDAR with 90^o vertical field of view, 64-channel vertical resolution, 360^o horizontal field of view, and $\pm 1.5 - 5$ cm precision. The large field of view of the LiDAR allows the system to measure the complete profile during the entire unloading process.

The LiDAR sensor was mounted on the unloading side of the grain cart so it would not interfere with the combine unloading. An unloading scenario was developed in the Unreal engine to validate the LiDAR placement as shown in Fig. 3.7a. Figure 3.7c shows the simulated LiDAR data from Unreal. By placing the LiDAR at 112 cm above the grain cart with a 65^o viewing angle, the LiDAR field of view can cover the entire grain cart. Meanwhile, most of the LiDAR measurements come from the region of interest, the area inside the grain cart, so the LiDAR point cloud density for the grain profile can be maximized.

After determining the LiDAR placement on the grain cart, an L-shape bracket was used to mount the LiDAR on the grain cart at the designed location as shown in Fig. 3.7b. In experiments, the LiDAR sensor was connected to a Linux PC via an Ethernet cable for sensor configuration and data logging.



(a) (b)

(c)

Figure 3.7. Grain fill model benchmark system. (a) placement of grain fill model benchmark system; (b) picture of grain fill model benchmark system mounted on the grain cart; (c) simulated LiDAR coverage from Unreal engine.

Benchmark experiment design and implementation

In the grain fill model benchmark experiment, grain was unloaded from a combine harvester (model: John Deere S790) to a static grain cart (model: Brendt 1020 XR). The harvester unloaded the grain from the back of the cart to the front in multiple small batches. Each batch of unloading takes 10-15 seconds. After unloading a batch of grain, the auger was stopped to allow the dust to settle down before LiDAR measurement to minimize dust

impact on LiDAR data quality. After the dust settled, the experimental data required by the grain fill model to simulate the same scenario were collected, including:

- LiDAR scans from the benchmark system to obtain benchmark grain fill profile and experimental angle of repose
- CAN data from combine and tractor to extract auger location from vehicles' GPS location
- Grain cart weight reading for auger flow rate estimation
- Weight of spilled grain if applicable

After unloading tests, some constant properties for the grain fill model were measured including:

- LiDAR scan of an empty grain cart for grain cart geometry
- LiDAR scan on flat ground to calibrate the LiDAR placement
- Density measurement of the grain to convert weight measurements to volume measurements

Benchmark data processing

To validate the accuracy of the grain fill model, the experimental results ought to have the same format as the grain fill model output, which is a 2D height map. However, the raw data from a LiDAR sensor is a point cloud containing the 3D coordinate of each point relative to the LiDAR sensor. Moreover, the experimental LiDAR point cloud includes unrelated data (e.g., ground, cart tarps) that could impact the ground-truth profile extraction. Therefore, a data processing pipeline was developed to convert the point cloud data to a ground-truth 2D height map $h_{exp}(x.y)$.

Figure 3.8 shows an example of point cloud data processing to obtain the experimental height map for benchmarking. As shown in Fig. 3.8a, the raw LiDAR point cloud is in the LiDAR coordinate and contains data outside the region of interest, the grain cart, so the

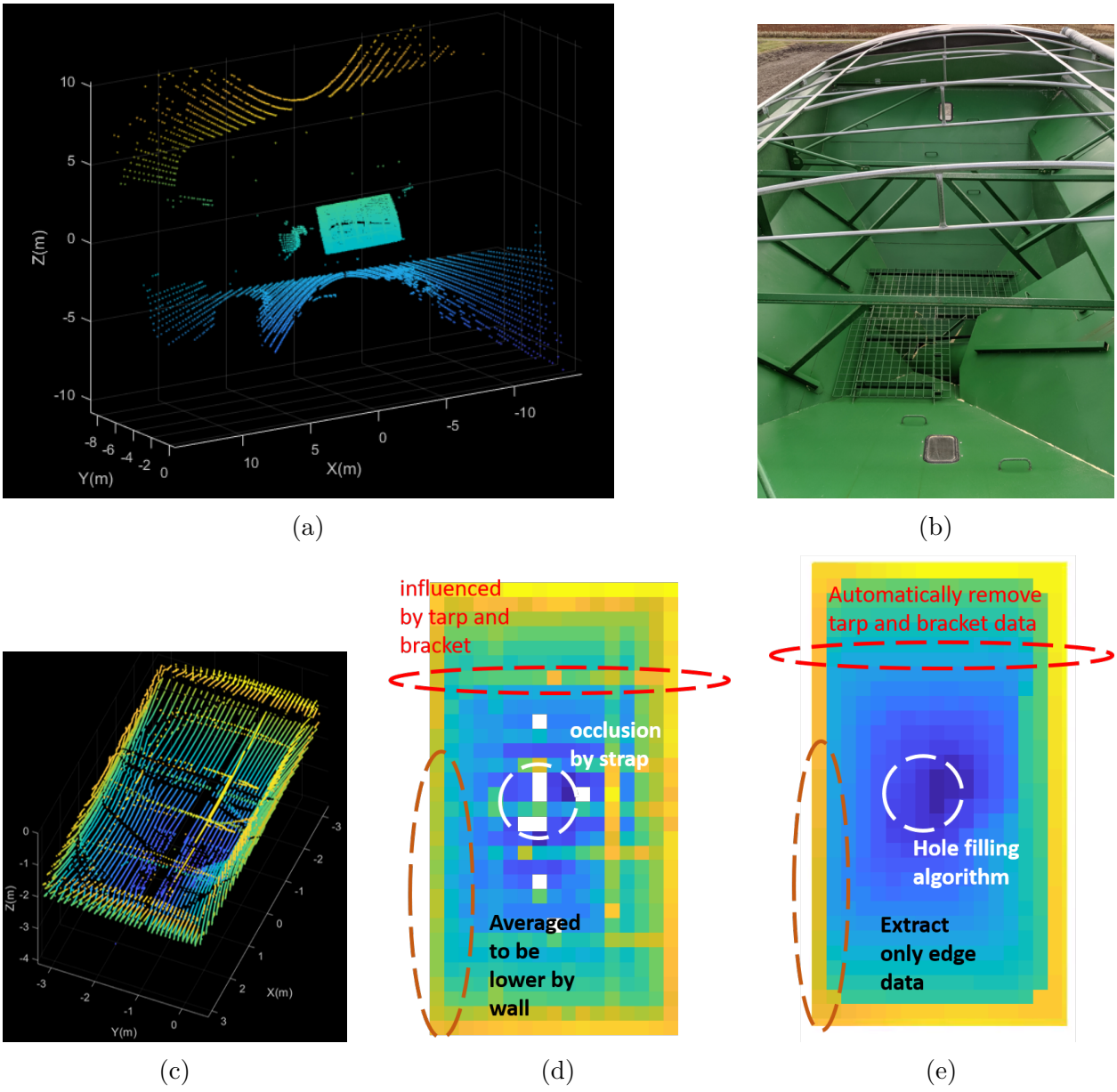


Figure 3.8. Benchmark data processing for point cloud data from empty cart (a) raw LiDAR point cloud; (b) a picture of the grain cart; (c) grain cart LiDAR point cloud; (d) unprocessed height map; (e) processed height map for grain fill model benchmark.

point cloud data outside the grain car were removed first and the points in the region of interest are then converted to world coordinate, which is used in the grain fill model.

To determine the transformation from LiDAR coordinate to world coordinate, the LiDAR pose was calibrated by a flat ground LiDAR scan with the same placement. After that, a flat plane was fitted from the point cloud data of the ground. The normal vector of the fitted plane is denoted as $\overrightarrow{n_p} = [x_p, y_p, z_p]^T$. Because the normal vector of flat ground on world coordinate is parallel to $[0, 0, 1]^T$, the transformation from LiDAR coordinate to world coordinate can be calculated by solving for a 3×3 matrix R_{LW} such that

$$R_{LW}\overrightarrow{n_p} = [0, 0, 1]^T \tag{3.10}$$

Via the transformation R_{LW} obtained from calibration, the raw LiDAR point cloud is rotated to the world coordinate. After that, based on the LiDAR placement and the grain cart geometry, the point cloud was cropped to only retain the portion of point cloud from inside the grain cart as shown in Fig. 3.8c. One can discretize a 3D point cloud to a 2D height map by taking the average of all the point inside a 2D grid:

$$h_{exp}(x, y) = \sum_{\mathrm{i}=1}^{N} z_{\mathrm{i}} \tag{3.11}$$

where N is the number of points inside the grid $x - \frac{d}{2} \leq x_{\mathrm{i}} < x + \frac{d}{2}, y - \frac{d}{2} \leq y_{\mathrm{i}} < y + \frac{d}{2}$

However, if the preprocessed point cloud in Fig. 3.8c by Eq. 3.11 was directly discretized by Eq. 3.11, the resultant height map did not represent the grain cart geometry. As shown in Fig. 3.8d, there are 3 major issues preventing this simple discretization from getting the desired geometry:

1. For boundary grids (as shown in the brown circle in Fig. 3.8d), the desired ground-truth geometry should be the height of the edge, but the simple average takes into account the data points from the wall as well, leading to a lower height value.

2. For grids that contain tarp or bracket data points (as shown in Fig. 3.8b and red circle in Fig. 3.8d), the height map should reflect only the bottom of the grain cart or grain height, and not be corrupted by the tarp or bracket.

3. Because of the occlusion from the tarps and brackets, some areas at the bottom of the grain cart (circled in white in Fig. 3.8d) do not have valid data, leaving holes in the height map.

To solve these issues, an enhanced height map extraction pipeline was proposed:

1. For boundary grids, the pipeline calculates the grid height by averaging the highest 5% points inside this grid to reduce the impact from wall data points.

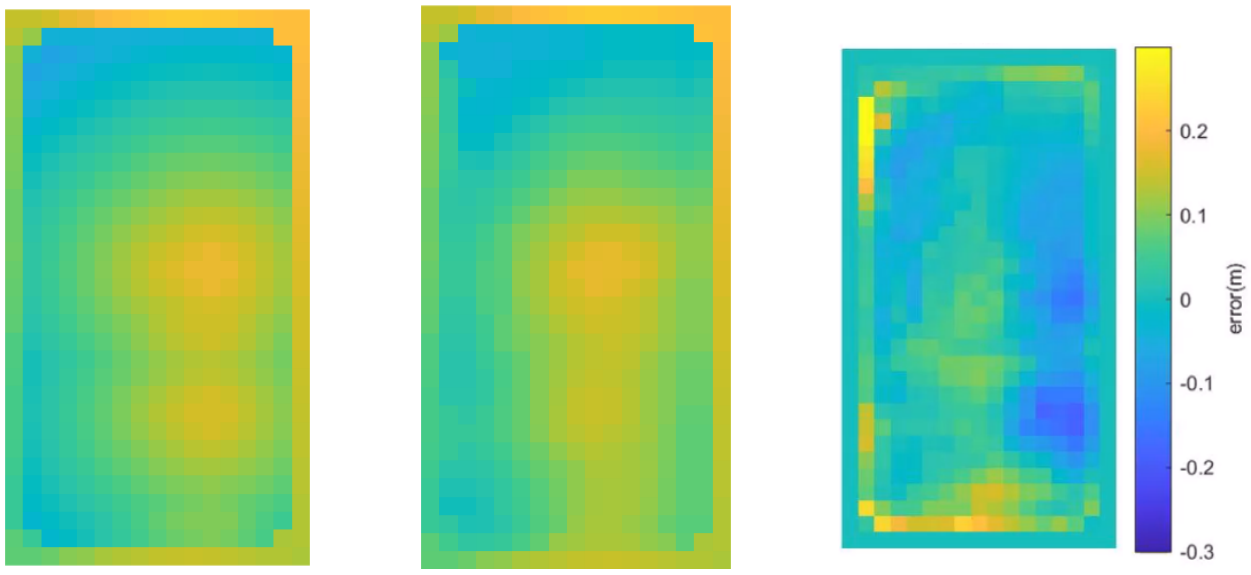
2. For non-boundary grids, the algorithm looks for the cluster of points with the lowest z value in the z distribution and calculate the average height in this cluster. In this way, the tarp or bracket data with higher z can be excluded.

3. After calculating all the grids with available data, the height value of the remaining empty grids is calculated by a bi-linear interpolation.

By applying the pipeline, the resultant height map of an empty grain cart is shown in Fig. 3.8e.

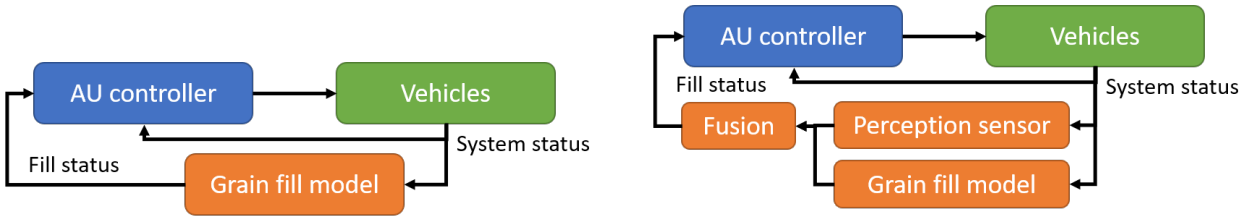
3.1.5 Experimental validation

The same benchmark data processing pipeline was applied for the entire unloading process and obtained the experimental ground-truth height map. With the same configurations as the experiment, the same process was simulated by the grain fill model. The experimental and simulation results are shown in Visualization 3. Visualization 4 compares the difference between the experimental and simulated profile. Figure 3.9 shows a typical step in this unloading process. The simulated profile has a similar shape to the experimental profile. The simulated profile error of most grids are within ± 0.3 m. The error near the edge is higher because the replicate padding for the low-pass filter to simulate the jostling effect introduces



(a) (b) (c)

Figure 3.9. Grain fill model benchmark result (associated with Visualization 3 and Visualization 4): (a) simulated grain fill profile; (b) experimental grain profile; (c) difference between simulated and experimental profile.



(a) (b)

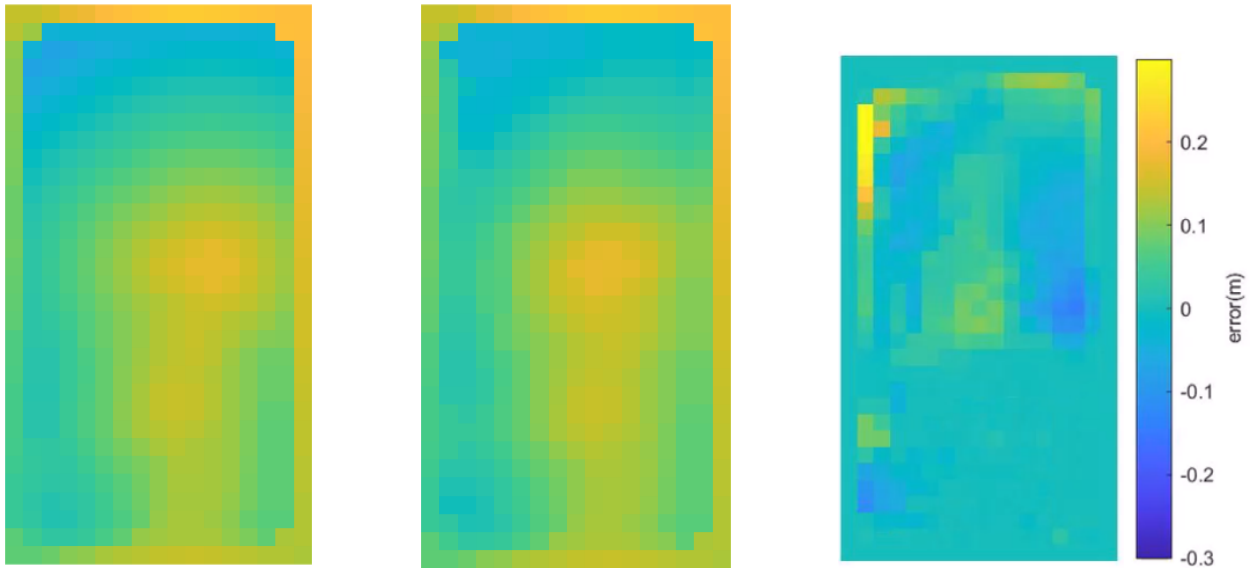
Figure 3.10. Grain fill model benchmark approaches: (a) approach 1: grain fill model is the only fill status feedback; (b) approach 2: grain fill model provides supplementary feedback for perception sensor (e.g., IPM, LiDAR).

error for the boundary data. For the center part of the profile, most error are within ±0.2 m.

Given the computational efficiency of the model, apart from being used in the simulation, the model can also be used as a real-time grain fill profile estimator in a controller to automate the granular material piling process. For example, the model can be used as a grain fill model to provide feedback for grain unloading automation. There are two different approaches to apply the model as a grain fill profile estimator as shown in Fig. 3.10. Approach 1 (Fig. 3.10a)

uses the grain fill model as the only means for grain fill profile feedback. In this case, the grain fill model calculates the profile solely based on the vehicle status, including vehicle GPS and auger status. As a result, the difference between the simulated feedback and the actual profile will be similar to the results in Fig. 3.9, in which the grain fill model runs the whole unloading process from an empty cart without any calibration. Because of the iterative nature of the grain fill model, the error in the simulated profile accumulates.

Approach 2 in Fig. 3.10b shows another possible application where the grain fill model works together with a perception system for grain fill profile feedback. Due to various lighting conditions and heavy dust during unloading operation, a perception system based on LiDAR or camera could lose track of the profile or be occluded by dust clouds. On one hand, to complement the perception system, the grain fill model can be used to estimate the grain height at the occluded area or provide unstopped feedback when the perception system loses track. On the other hand, the grain profile from the perception system can be used for calibrating the grain fill model by reinitializing the input profile to the model.



(a) (b) (c)

Figure 3.11. Grain fill model benchmark result with reinitialization at the beginning of each grain batch (associated with Visualization 5 and Visualization 6): (a) simulated grain fill profile; (b) experimental grain profile; (c) difference between simulated and experimental profile.

Table 3.1. Grain spillage validation

	Spillage amount (lbs)	**Spillage volume (bu)**
Ground truth measurement	90.23	1.61
Approach 1	251.02	4.48
Approach 2	155.54	2.78

To evaluate the accuracy of the grain fill model in approach 2, the grain fill model simulated the same process while reinitializing the grain profile at the beginning of each grain batch. After that, the simulation error was calculated by comparing the difference after unloading a batch of grain. Each batch has 40-80 bushels of grain and takes about 10-20 seconds to unload. Visualization 5 and Visualization 6 show the results of approach 2. Figure 3.11 shows a representative frame of the unloading process. Compared with approach 1 in Fig. 3.9, with an accurate grain profile from a perception system to calibrate the grain fill model, the models show a better accuracy with error generally within ± 0.15m.

In addition to grain height validation, the spillage estimation of grain fill model is also validated by intentionally overfill the cart. In the same testing, the final batch of grain was unloaded to the same location and causes 90.23 lbs of spillage according to the weight measurement of the spilled corns. Table 3.1 summarizes the spillage estimation. Both benchmark approaches of the grain fill model detects the spillage successfully. In comparison, approach 2, which updates the initial profile with ground truth data at the beginning of every batch, demonstrates a better accuracy.

The overestimate of the spillage amount in both cases indicates a more conservative estimation on spillage for the grain fill model, and could result in a more conservative unloading target. However, the magnitude of the overestimation is only 2.87 bu for grain fill model only and 1.17 bu for grain fill model with perception feedback correction before the unloading of each batch. Considering that the auger flow rate is around 3.5 bu/s, if applied in an actual unloading scenario, the estimation only advances the auger off by no more than 1 second. Moreover, for an 1000-bu grain cart, the impact on the carrying capcity insdie the cart is no more than 0.3% for both cases, which is negligible.

(a) (b)

Figure 3.12. Spillage location validation for the grain fill model. (a) benchmark system measurement after spillage; (b) spillage location predicted by the grain fill model shown in red.

The spillage location is also assessed qualitatively. The spillage happened when unloading the last batch of grain. Figure 3.12b shows the estimated spillage location (highlighted in red) by the grain fill model. Both approach 1 and approach 2 of the grain fill model give the same estimation of the spillage location, which is at the back of the grain cart. It is consistent with the observation in the field. Additionally, Figure 3.12a shows the grain profile after spillage. Consistent with the grain fill model prediction, the grain profile inside the cart has the same height as the cart edge at the longitudinal back of the cart, indicating the same spillage location.

3.1.6 Conclusions

In this paper, a computationally efficient model was proposed to simulate the piling process of granular material inside a container. Based on the geometric characteristics of

granular material with additional refinements on flow dynamics and jostling effect, the model can be run in real-time while maintain sufficient accuracy for the general profile estimation. The model is applied to simulate the grain piling process during agricultural unloading from a combine harvester to a tractor-driven grain cart. By comparing with the measurement results from a high-accuracy LiDAR, the in-field testing validates the accuracy of the piling model during a static grain unloading to be $\pm 0.2m$. Given the real-time capacity, the paper presents two possible approaches for the model to be used as a real-time estimator for the grain unloading automation applications.

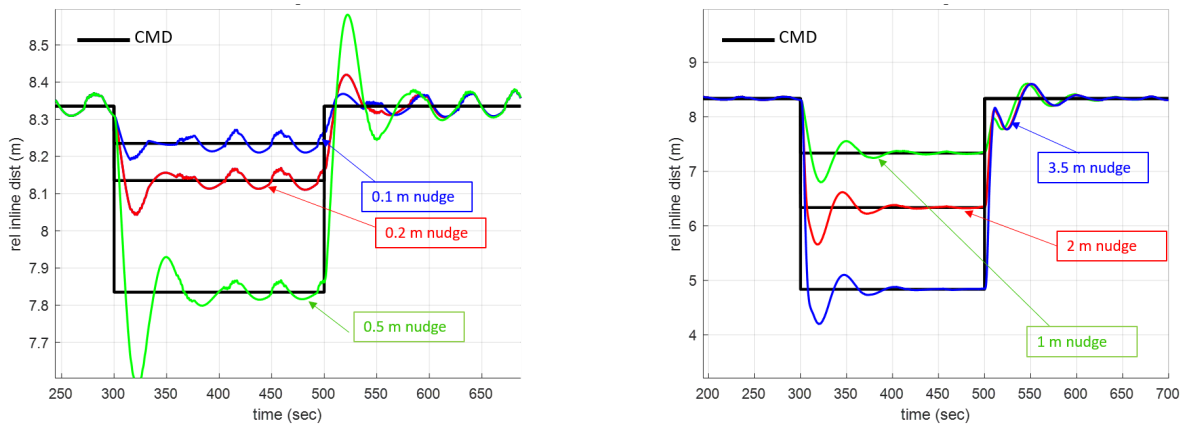
3.2 Vehicle dynamics model

The vehicle dynamics model simulates the dynamics for the relative position between the tractor and the combine harvester, which ultimately leads to the change of auger location. In this vehicle system, Machine Sync is used as the low-level controller to control the speed and the moving direction of the tractor to achieve the desired relative location. Therefore, instead of modeling the detailed dynamics for the tractor plant and the combine plant separately, the vehicle dynamics model simulates the relative motion between the combine and the tractor controlled by Machine Sync as an aggregated system. As a result, the aggregated Machine Sync - tractor - combine plant takes the input as commanded relative location and outputs the actual relative position. As a way to identify the system dynamics, a data-driven system identification method was used to estimate the system model.

3.2.1 System ID in simulation

In practice, the Machine Sync system takes nudge commands from the operator and updates its target relative position between the two vehicles by a fixed distance, so nudges of different sizes were commanded to the Machine Sync model to observe the vehicle dynamics response.

Figure 3.13 shows some representative responses from the Machine Sync system with nudge commands in the longitudinal direction. Machine Sync system response can be categorized into two plants based on nudge sizes. Fig. 3.13a shows the small nudge plant response



(a) (b)

Figure 3.13. Vehicle response to Machine Sync nudge in longitudinal direction. (a) small nudge, nudge size smaller than 1m; (b) larger nudge, nudge greater than 1m.

when the nudge size is smaller than 1 m. In this case, for both nudge up (i.e., moving tractor forward) and nudge down (i.e., moving tractor backward), the actual location overshoots by about 50% and then converges to the commanded location in more than about 100 seconds. However, for nudge size greater than 1 m shown in Fig. 3.13b the responses for nudge up and nudge down are different. The nudge down response is similar to small nudge plant in which the overshoot is high and the response is sluggish. In comparison, the nudge up response starts with an undershoot of about 90% and then converges to the commanded location at a faster rate of about 80 seconds.

After generalizing the Machine Sync dynamics into two distinct plants depending on the nudge size and direction, the large nudge up plant was modeled as a third-order system and the small nudge plant was modeled as a second-order system. After that, the response simulated from the same plant was normalized. Instrumental variable estimation method [56] was used to identify the parameters in the transfer function. Figure 3.14 and 3.15 show the system ID results from small nudge plant and large nudge plant respectively. Both identified plants show a similar response to the normalized raw simulation data, especially for the oscillating frequency and overshoot percentage, demonstrating that the identified plants can represent the Machine Sync - combine - tractor system dynamics in simulation.

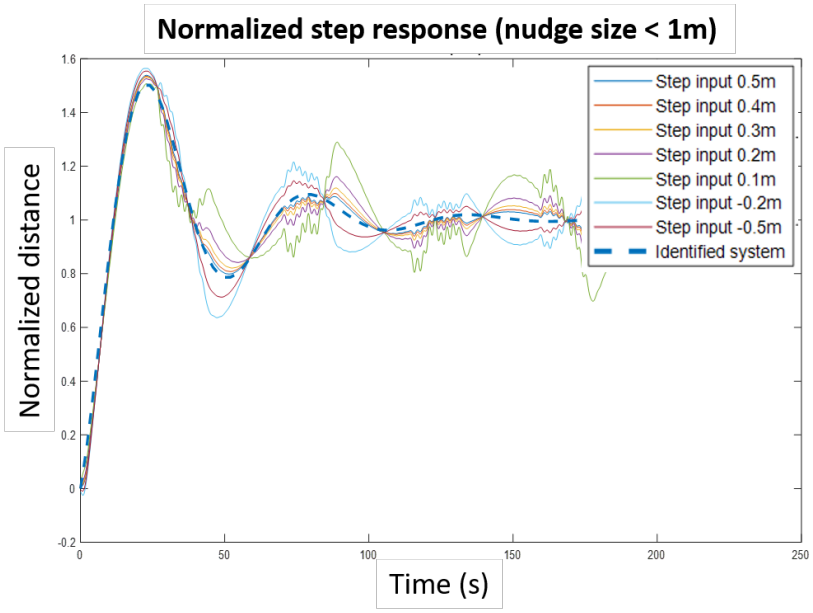


Figure 3.14. System ID from Machine Sync simulation response for small nudge plant

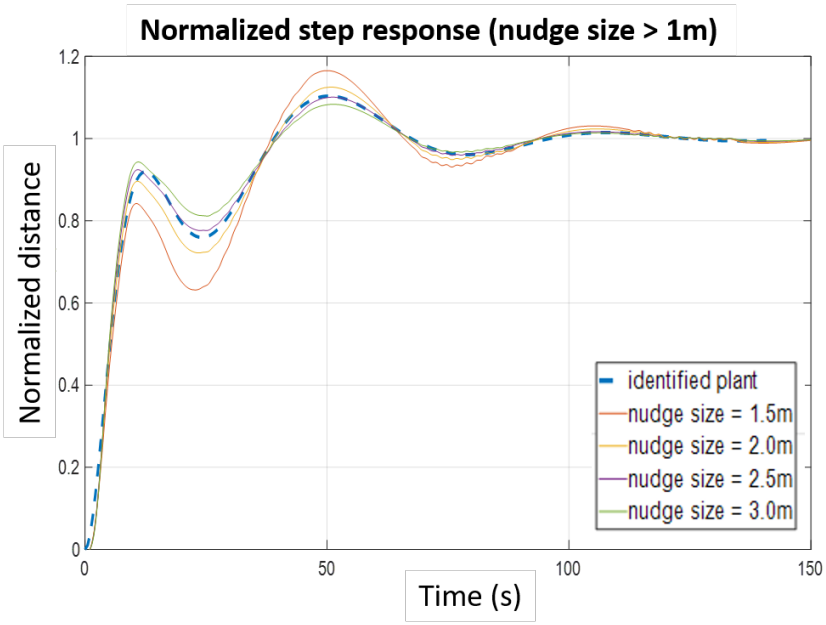


Figure 3.15. System ID from Machine Sync simulation response for large nudge up plant

3.2.2 System ID from in-field testing

An experimental platform was developed to identify the characteristics of the Machine Sync controlled combine/tractor relative position response. The vehicles used in experiments

included a combine harvester (model: John Deere S660), a tractor (model: John Deere 8345R with Infinitely Variable Transmission), and a grain cart (model: Brandt 1020XR). In the combine and the tractor, two CAN loggers (model: Kvaser Memorator Pro 2xHS v2) were used to log the CAN data from the combine CAN and the tractor CAN simultaneously.

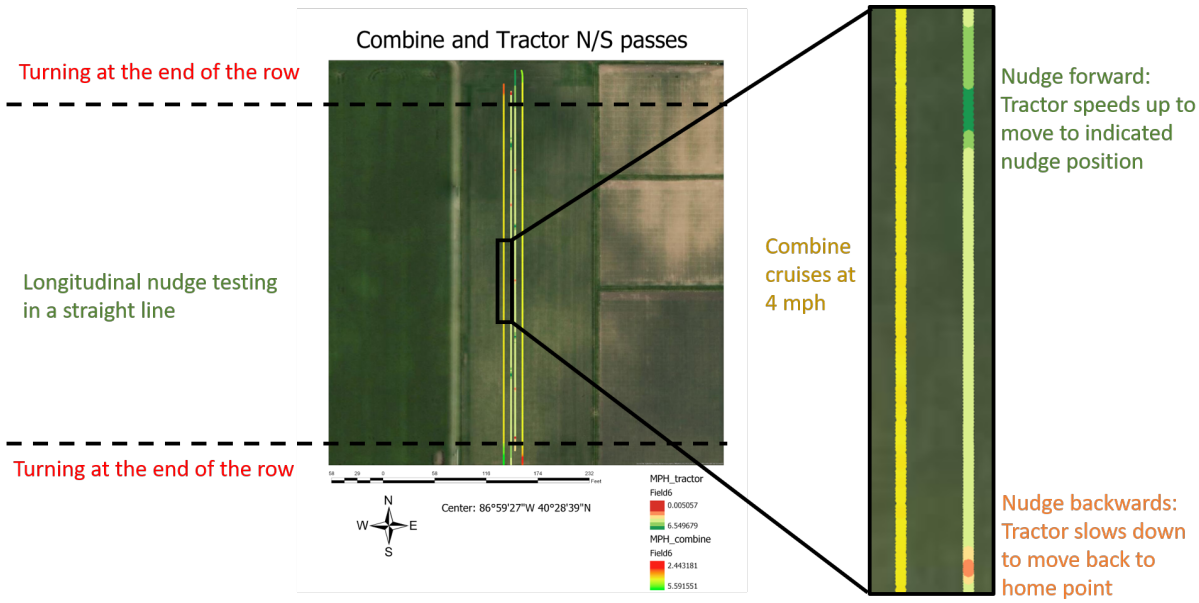


Figure 3.16. GIS visualization of vehicle trajectory during Machine Sync system ID.

The system ID experiments were carried out on a flat terrain. During testing, AutoTrac system was engaged to keep the harvester moving in a straight line and the combine throttle remained unchanged to maintain a stable speed. After the tractor moved into the Machine Sync operation zone, both the tractor and the combine engaged Machine Sync to take control of the tractor speed and heading direction. After the tractor stabilized at the first desired location, a nudge command was sent to the Machine Sync system by the operator from the combine display. After receiving the nudge signal, Machine Sync controlled the tractor speed to move it to the new desired location. Figure 3.16 shows a typical example of the vehicle trajectories during the system ID test. The tractor sped up when the operator nudged forward and slowed down when the operator nudged backwards.

The nudge tests for Machine Sync system ID were conducted with various combine speeds, nudge sizes, nudges directions, the grain cart loads. Figure 3.17 summarizes all 280 nudge

Grain cart loading	Speed (mph) \ Nudge size (ft)	Small nudge (plant 1)						Large nudge (plant 2)							
		0.5	1.0	1.5	2.0	2.5	3	3.5	4	5	6	7	8	9	10
Full	2	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	3	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	4	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	5	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	6	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
Empty	2	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	3	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	4	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	5	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-
	6	+	+	+	+	+	+	+	+	+	+	+	+	+	+
		-	-	-	-	-	-	-	-	-	-	-	-	-	-

Legend +: nudge up -: nudge down

Figure 3.17. Machine Sync system ID configurations.

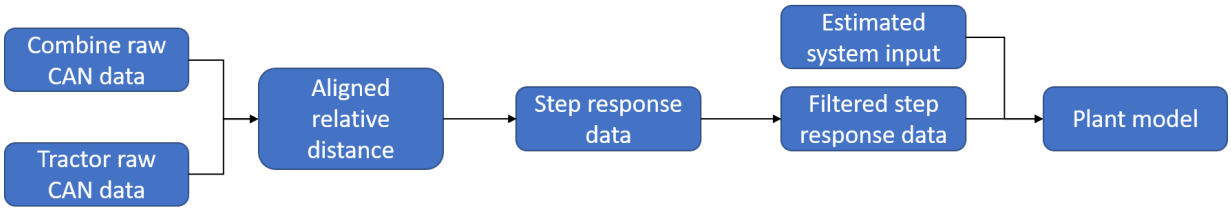


Figure 3.18. Data processing pipeline for experimental Machine Sync system ID

tests. A data processing pipeline was developed to extract step responses from the GPS information in the CAN logs as shown in Fig. 3.18.

First, because the combine CAN data and the tractor CAN data were collected separately, the CAN bus data from the combine CAN and the tractor CAN were aligned based on the GPS time in the logs. After data alignment, the relative distance in the nudging direction was computed from the GPS location. In these experiments, because the vehicles are moving in north/south line, the relative location in nudge direction d_{T-C} can be calculated from the latitude by

$$d_{T-C} = R_{earth} \left(\lambda_T - \lambda_C \right) \tag{3.12}$$

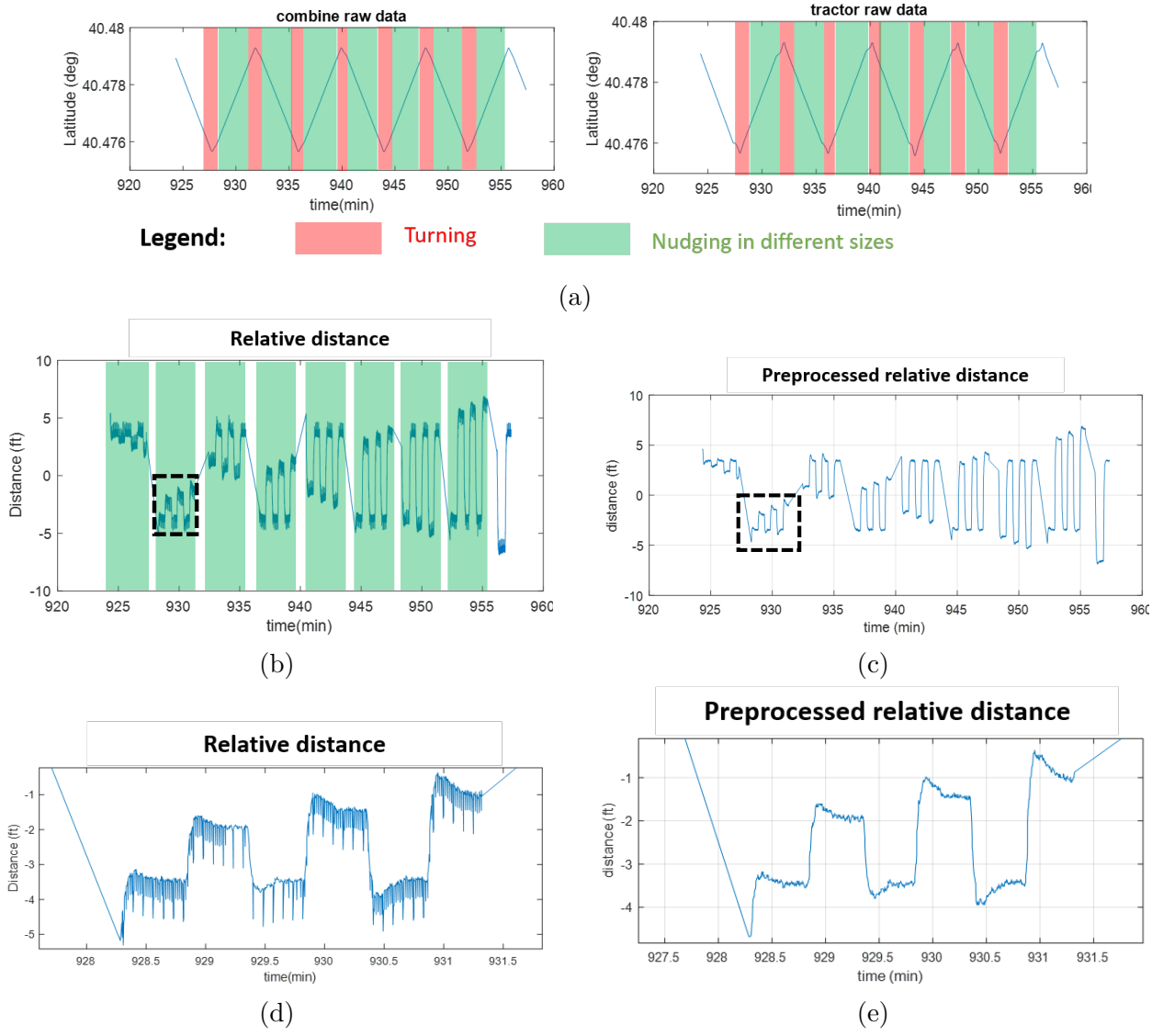


Figure 3.19. Experimental system ID data prepossessing. (a) raw vehicle GPS; (b) relative distance between tractor and combine; (c) filtered relative distance between tractor and combine; (d) zoomed-in view of (b) at the boxed region; (e) zoomed-in view of (c) at boxed region.

where R_{earth} is the radius of the earth, λ_T is the latitude of the tractor and λ_C is the latitude of the combine.

As highlighted in red in Fig. 3.19a, the vehicles were not controlled by Machine Sync when they were making turns at the end of the rows, so the combine compass bearing was used to remove the data from turning points. The resultant relative distance during the effective periods of system ID tests is shown in Fig. 3.19b. Figure 3.19d shows a zoomed-in view of Fig. 3.19b. The relative distance directly calculated from the GPS has noise, because tractor GPS and combine GPS data were not acquired at the same time. To address this issue, a median filter was applied to remove the noise, as shown in Fig. 3.19c and Fig. 3.19e.

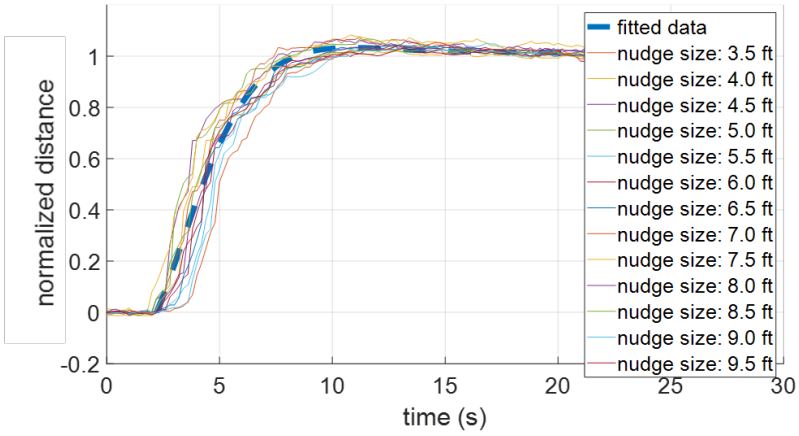


Figure 3.20. System ID from step response data for large nudge (greater than 1m), 4-mph combine, and empty grain cart: normalized step response data with identified plant

The step response data were then extracted from the filtered distance. By normalizing the step response and applying the instrumental variable method, the coefficients of the transfer functions were determined. A similar system ID process was applied for all testing conditions listed in Fig. 3.17. Figure 3.20 shows the system identified from step response data for large nudges when the combine was moving at 4 mph and the tractor was towing an empty grain cart.

Figure 3.21 demonstrates the Machine Sync dynamics obtained from experimental system ID. In summary, the dynamics of Machine Sync only depend on the nudge size but do not seem to change with nudge direction, combine speed, or grain cart loading. When the nudge

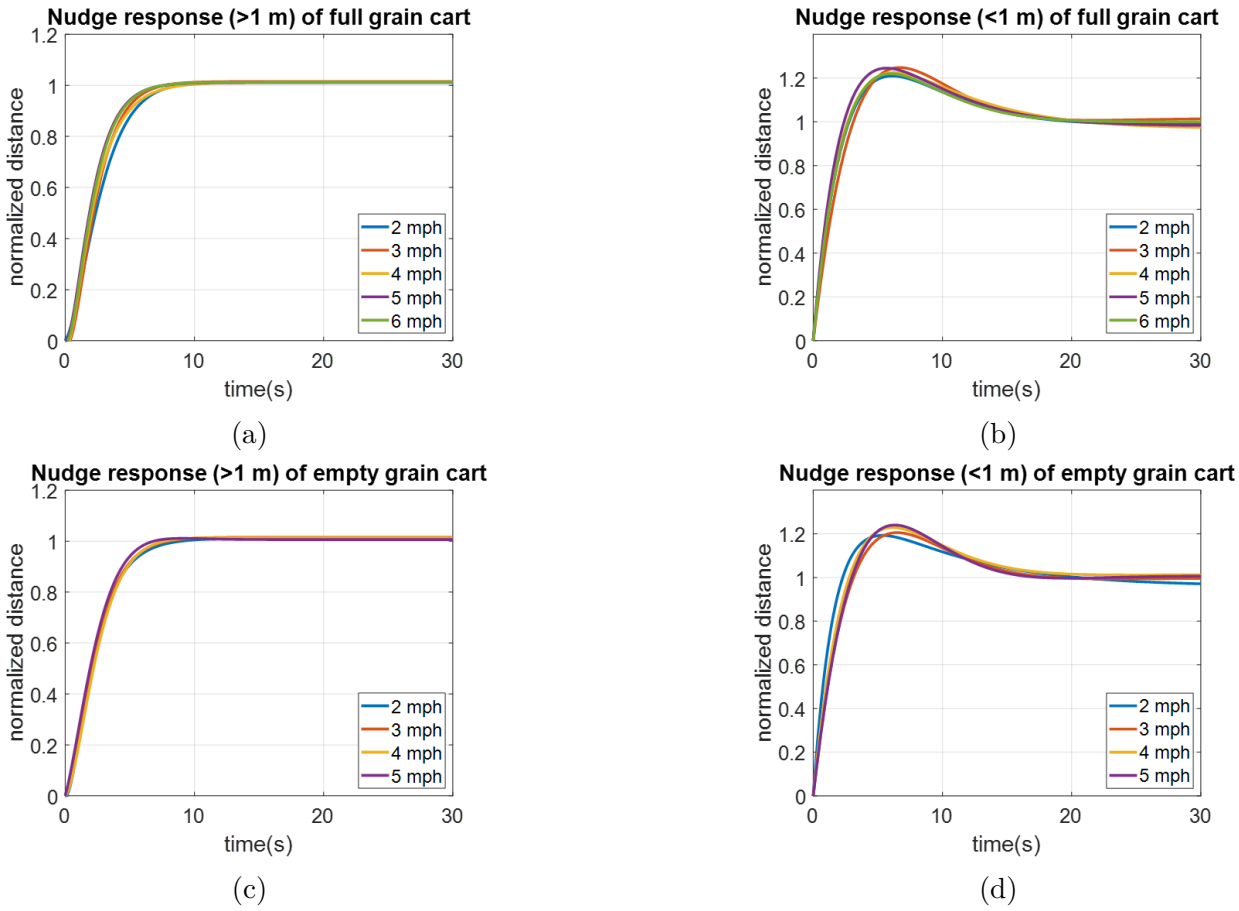


Figure 3.21. Experimental system ID results: (a) large nudge with full grain cart; (b) small nudge with full grain cart; (c) large nudge with empty grain cart; (d) small nudge with empty grain cart.

size is greater than one meter, Machine Sync can be modeled as a overdamped second-order system with a six-second settling time.

$$G_L = \frac{0.0453s + 0.4942}{s^2 + 1.315s + 0.4942} \tag{3.13}$$

When nudge size is smaller than one meter, Machine Sync can be modeled as a second-order system with a settling time of 18 seconds and 20% overshoot.

$$G_S = \frac{0.6049s + 0.1021}{s^2 + 0.5458s + 0.1021} \tag{3.14}$$

3.2.3 System characteristics with different transmissions

The development of the automatic unloading system is majorly based on the system setup with an IVT (Infinitely Variable Transmission) tractor. However, a number of PST (Power Shift Transmission) tractors are being used in the US farms. Therefore, the system ID in-field testing was also conducted for the Machine Sync system with a PST tractor. The experimental system ID for the PST tractor uses the same methodology as the IVT tractor. The PST tractor model used in the in-field testing is John Deere 8R340.

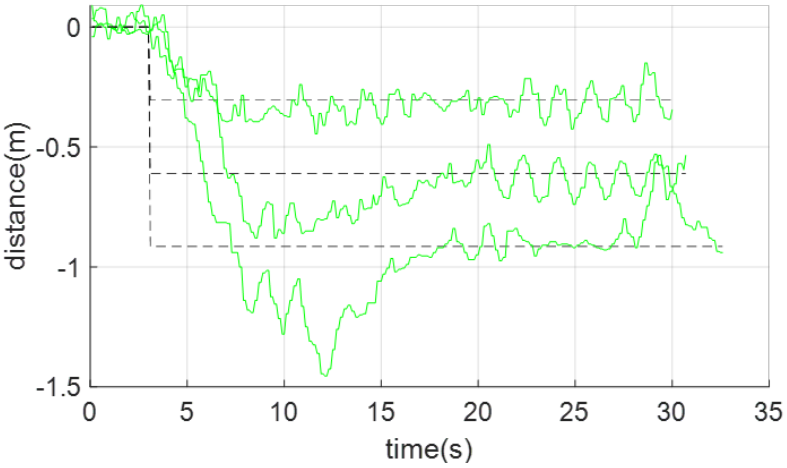


Figure 3.22. Relative location of PST tractor to the combine harvester controlled by Machine Sync system with sensitivity 2

When using the Machine Sync system, there are five sensitivity levels that can be selected to adjust how fast the tractor goes to the desired location. The higher the sensitivity level, the faster the tractor goes to the desired location. The default sensitivity level is three, so the system ID results from the IVT tractor shown in Fig. 3.21 is based on sensitivity 3. However, when using the PST tractor, the tractor is going to have significant oscillation when the sensitivity level is equal or greater to 2. Figure 3.22 shows an example of the relative location between two vehicles when the sensitivity is 2. The oscillation remains significant even after the tractor reaches the desired location, making it uncomfortable for an operator to stay in the cabin for a long time, and thus not practical for the use of automatic unloading.

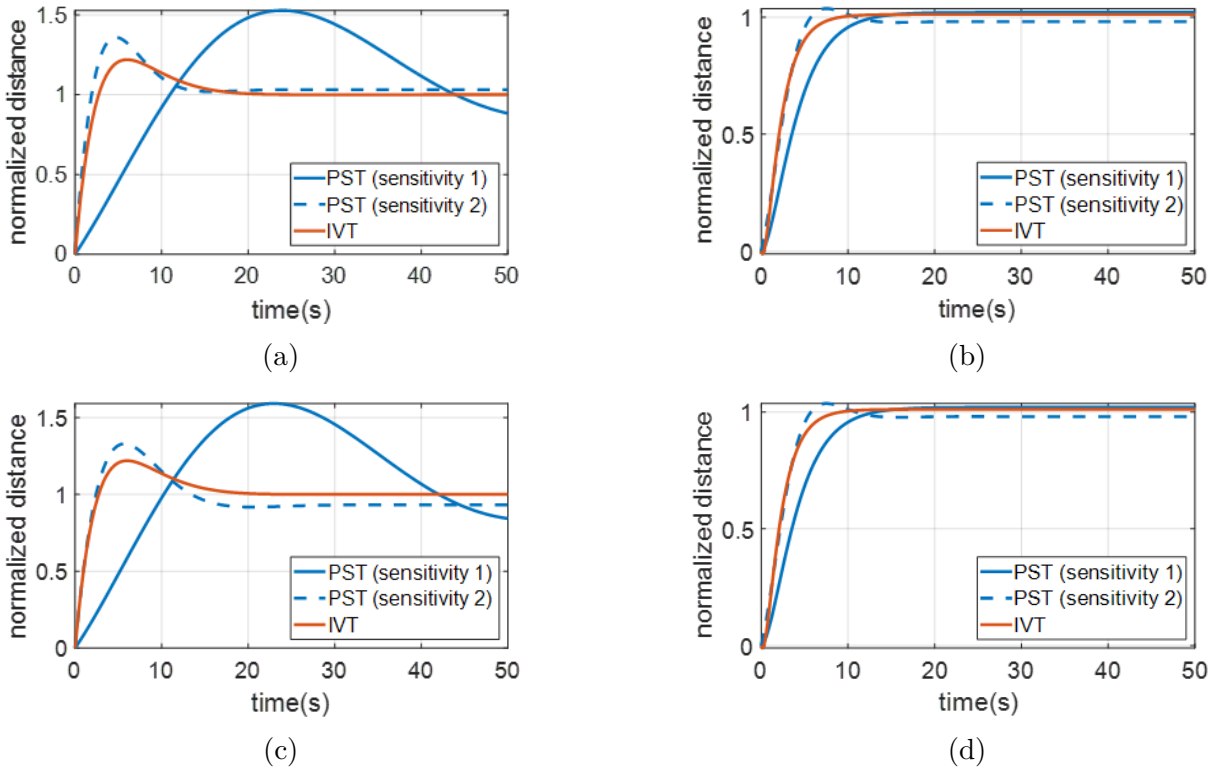


Figure 3.23. Machine Sync experimental system ID results comparison between PST tractor and IVT tractor: (a) small nudge forward; (b) large nudge forward; (c) small nudge backward; (d) large nudge backward.

Therefore, the system ID experiments were only conducted on the PST tractor with low Machine Sync sensitivities. Figure 3.23 compares the Machine Sync dynamics difference between the PST tractor with two different MS sensitivities and the IVT tractor.

With a PST tractor, the Machine Sync dynamics does not change with regard to nudge direction, grain cart fullness, or vehicle speed either. Similar to IVT tractor, the Machine Sync system has two different dynamics based on nudge size with a threshold of 1 meter.

However, the Machine Sync dynamics with PST tractor and IVT tractor are drastically different from each other. For sensitivity 1 response, Machine Sync with PST has a much longer settling time than IVT. For small nudge (less than 1 meter) response, the PST system has a 50% overshoot while the IVT system only has about 15% overshoot. For large nudges (greater than 1 meter), the 5 % settling time is about 11 seconds, longer than 5-second settling time for an IVT tractor.

The sensitivity 2 response of a PST tractor shows a similar settling time to the IVT system. However, it shows greater overshoot than the IVT system for both the small nudge response and the larger nudge response.

The difference between the system dynamics with different tractors can impact the design of the automatic unloading controller design. For example, the closed-loop movement controller to improve system dynamics may need to be synthesized differently for the automatic unloading application on machines with a different transmission.

3.3 Perception model

3.3.1 Introduction

The automation in agricultural machinery provides an immense opportunity while also poses various challenges to the development of the perception system used in the automated machinery. Among these challenges, one of the most critical issues is the safety and reliability of the perception system. To build trust and expand the acceptance of automated machinery for the operator, the perception system not only needs to work properly in nominal scenarios, but also has the capacity to deal with various less frequent scenarios. This applies to both the development and the validation stage of the perception system.

Most recent paradigms for developing a perception system have benefited enormously from Machine learning [57]. The Machine learning methods that are widely used in automated machinery, such as reinforced learning, deep learning, require a huge amount of data for training and testing [58], [59]. Traditionally, the data are collected from in-field testings. However, in-field testings are labor-intensive and costly. To build a reliable algorithm that works for different vehicle configuration and operating conditions, the data set used for training and testing the algorithm also needs to encompass these various conditions.

To address the limitation of obtaining training and testing data from in-field testing, generating data from a virtual environment for training and testing gains significant attention recently [60], [61]. Compared with in-field testing, data set generation from virtual environments are more versatile and cost-effective. Therefore, simulation data have been widely used for the development of autonomous vehicles including the perception system [62].

Furthermore, recent development in game engines provides a solid foundation to generate photo-realistic images using computer graphics technology. Researchers develop perception systems that are trained from these types of virtual data [63]. Some researchers report that perception systems trained with virtual data can be comparable with the ones trained with real-world data [64].

The extensive use of virtually generated data in perception system development in on-road vehicles reveals a great opportunity to apply the same techniques for agricultural machinery automation. Moreover, in agricultural applications, generating these data sets from in-field testing could sometimes be challenging. For instance, to collect data for actual machinery's operations, the tests can only be conducted in specific windows within a year due to the nature of the plant growing cycle. Additionally, within these already limited time windows, data collection for edge cases such as various weather conditions will further shrink the viable time.

Therefore, a perception model in Unreal engine [65] was developed to simulate the response of both stereo-camera based perception systems and LiDAR-based perception systems during unloading on the go. Additionally, the model can simulate the unloading operations in different edge cases and different machine configurations to provide a broad range of data. Besides modeling the unloading process, the perception model was also used to recommend system configurations and propose a validation pipeline to evaluate the perception system performance.

3.3.2 Camera data simulation

Figure 3.24 shows the architecture of the perception model of an unloading process for a stereo camera-based perception system. The key function of a perception model is to simulate the raw camera images for the stereo camera under different scenario definitions including vehicle models, lighting conditions, camera parameters. After that, the simulated stereo camera image can be used to train or validate the perception system algorithm.

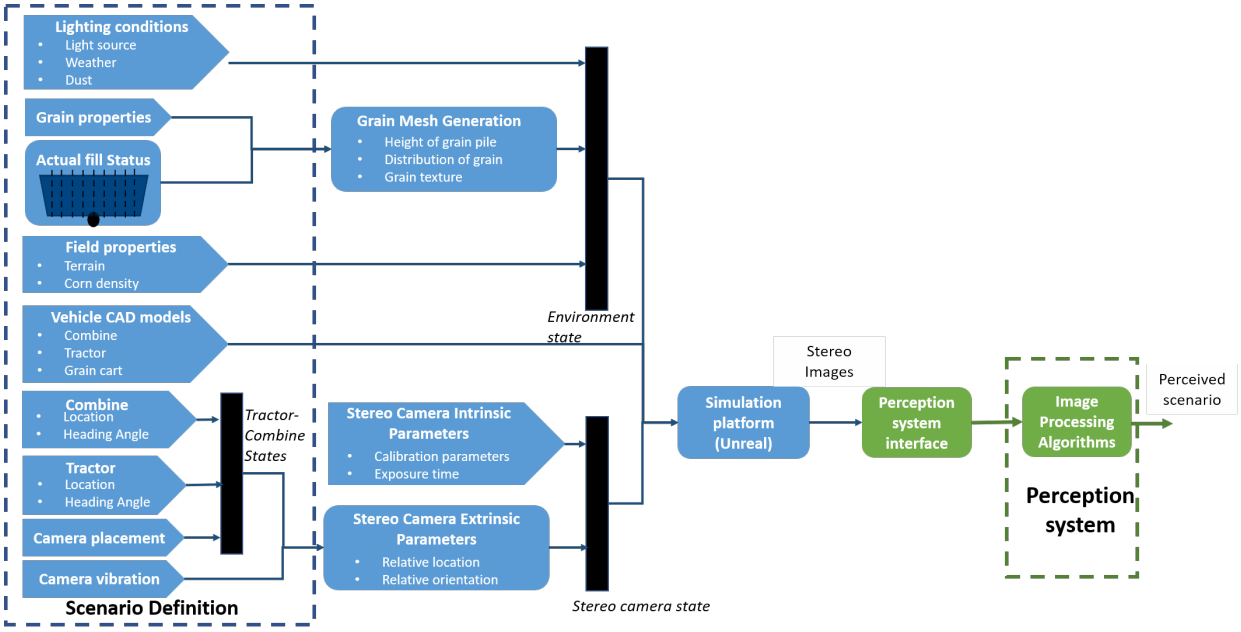


Figure 3.24. Perception model simulation architecture

Camera model

The pin-hole camera model was adopted to simulate the left and right lens of the stereo camera simultaneously. The projection from the 3D world coordinate (x^w, y^w, z^w) in Unreal to the 2D camera coordinate in each camera image (u^c, v^c) can be described as

$$s^c \begin{Bmatrix} u^c \\ v^c \\ 1 \end{Bmatrix} = \mathbf{A^c}[\mathbf{R^c}, \mathbf{t^c}] \begin{Bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{Bmatrix} = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R^c}_{3\times3} & \mathbf{t^c}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{Bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{Bmatrix} \tag{3.15}$$

where s^c is the scaling factor, $\mathbf{R^c}$ and $\mathbf{t^c}$ are the rotation matrix and the translation vector from the world coordinate to camera lens coordinate, $\mathbf{A^c}$ is the intrinsic parameter of the camera showing the transformation from camera lens coordinate to the imaging plane.

In Eq. 3.15, the rotation matrix and the translation vector are the extrinsic parameters of the camera, and the change of camera placement or vehicle location will alter these parameters. The intrinsic parameters, on the other hands, includes the properties of the

camera: α_x and α_y are the focal lens of the camera in horizontal and vertical direction; γ is the skew factor, and (u_0, v_0) is the principle point in the image plane.

Because Unreal uses an ideal pin-hole camera. The focal length will be the same for each direction of the camera $\alpha_x = \alpha_y = f$; There is no skew effect $\gamma = 0$; and principle point will be the center of the image. As a result, the intrinsic parameter can be simplified to

$$\mathbf{A^c} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.16}$$

By alternating the intrinsic parameter of the camera, the impact of stereo camera parameter was simulated. For example, the field of view of an ideal camera, which shows how wide the camera can see, can be calculated from the intrinsic matrix in Eq. 3.16 by

$$fov_y = 2\arctan\frac{u_0}{f_y}, fov_x = 2\arctan\frac{v_0}{f_x} \tag{3.17}$$

where fov_y is the vertical field of view, fov_x is the horizontal field of view.

Figure 3.25 shows the impact from different camera horizontal FoVs. With a greater field of view, the camera can have better coverage of the grain cart and thus obtain more complete information about the unloading process. Nonetheless, a larger field of view also results in larger distortion, which could impact the final accuracy of the perception results.

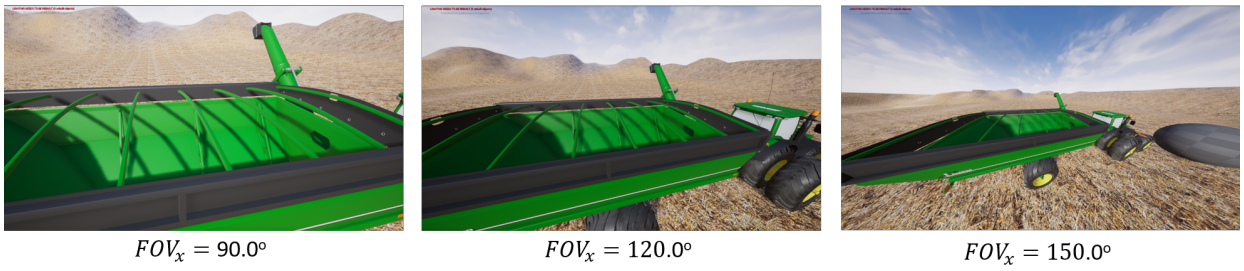
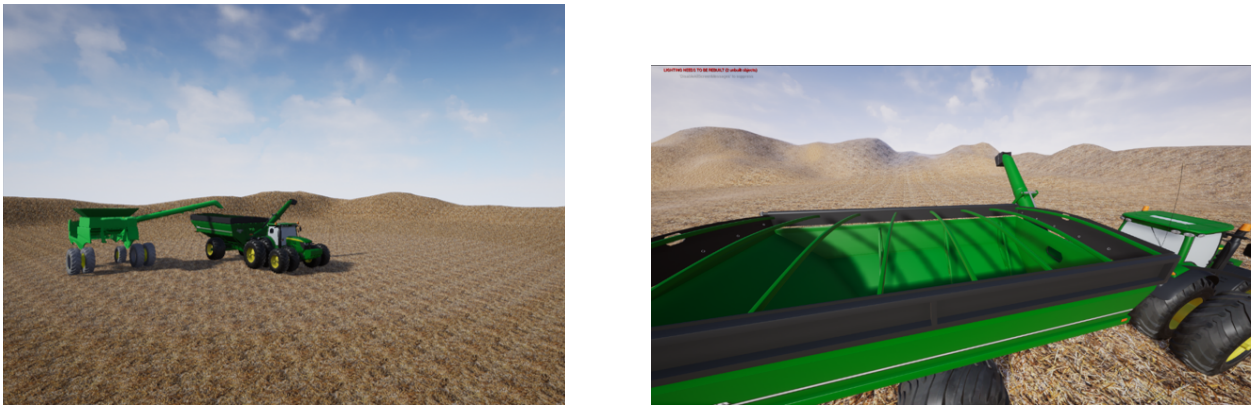


Figure 3.25. Simulated camera images with different field of views

Scenario setup

After defining how the camera captures images in the Unreal environment to generate raw data for the perception system, the next step is to build the virtual environment for unloading on the go operation. As shown in Fig. 3.24, the scene is constructed with a combine harvester, a tractor towing a grain cart, and cornfields.



(a) (b)

Figure 3.26. Perception model components (a) a perspective view of perception model components in Unreal; (b) simulated camera view from auger in perception model.

The vehicle CAD models are imported to the Unreal environment with the Datasmith feature. Figure 3.26a shows a perspective view of the perception system with the imported vehicles and terrain and texture designed in Unreal. After generating the virtual world, a stereo camera was placed on the combine auger to simulate the response from the perception sensors. Fig. 3.26b shows the simulated camera image from one of the stereo camera pair.

Besides the objects with fixed geometry such as vehicles and terrain, the unloading process simulation also involves dynamic objects, such as the grain pile inside the cart and the grain flow from the auger. The visualization techniques used for static meshes are no longer applicable to these dynamic objects.

To enable the grain fill profile visualization, a pipeline was created to convert the grain fill profile (2D array) simulated from the grain fill model in MATLAB Simulink to a 3D procedural mesh in the Unreal engine. On the Simulink side, the process to simulate the

2D grain fill profile array has been introduced in sec. 3.1. After generating the 2D grain fill map, the height value was parsed to a floating-point number array and then pass the array to Unreal with a shared memory module developed by Mathworks.

On Unreal side, a procedural mesh was initialized with same vertices number as the gird number in the grain fill profile. The vertices are defined in the grain cart coordinate with location as

$$\begin{bmatrix} x_{\mathrm{i,j}} & y_{\mathrm{i,j}} & z_{\mathrm{i,j}} \end{bmatrix}^T = \begin{bmatrix} d\mathrm{i} & d\mathrm{j} & h_{\mathrm{i,j}} \end{bmatrix}^T \tag{3.18}$$

where $[x_{\mathrm{i,j}}, y_{\mathrm{i,j}}, z_{\mathrm{i,j}}]$ is the position of the vertice i in cart longitudinal direction, j in cart lateral direction; d is the grid size, and $h_{\mathrm{i,j}}$ is the height value generated by grain fill model in the corresponding grid.

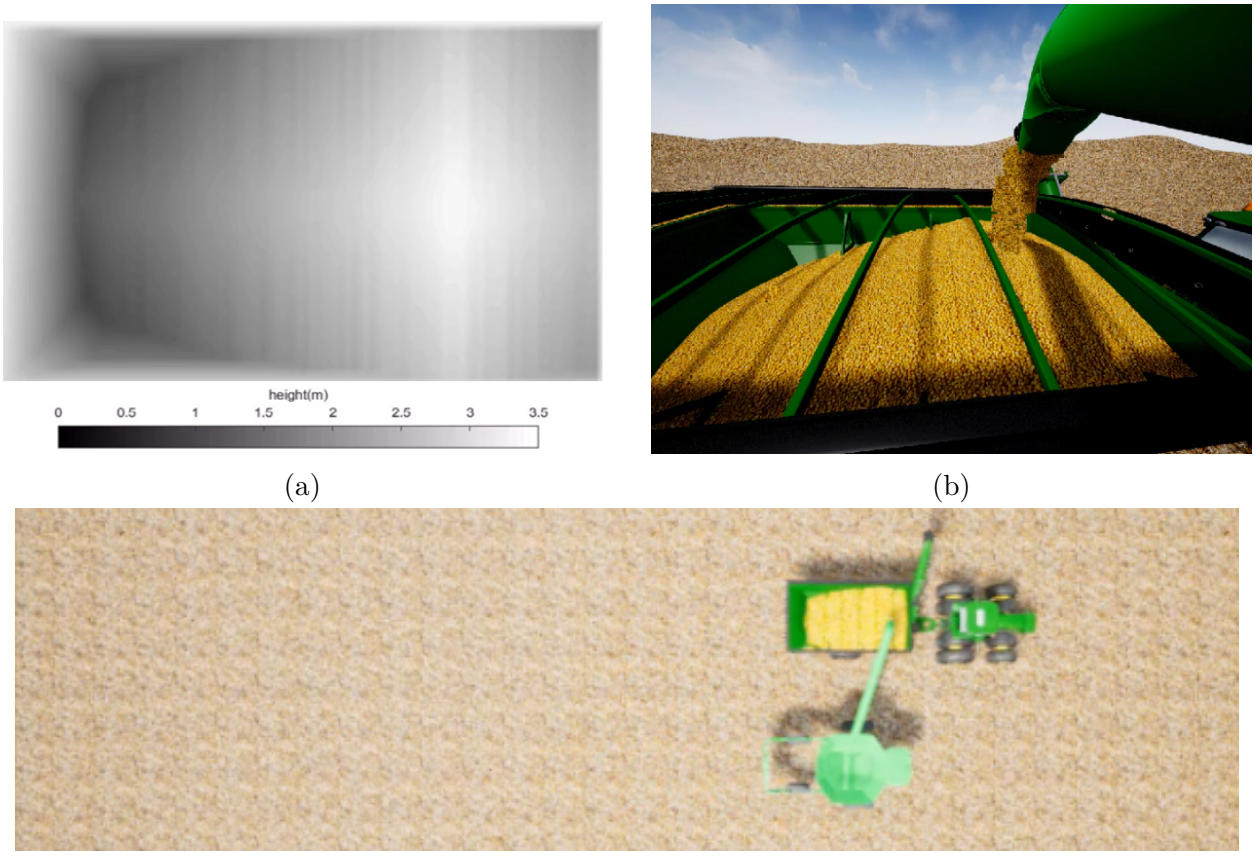
With vertices in the mesh defined, the edge are defined by connecting the vertice [i,j] with neighboring vertices [i+1, j], [i, j+1], and [i-1, j-1] if they exist. Finally, the triangles in the mesh are defined by the closed neighboring edges.

Another dynamic object is the grain flow. To simulate the grain flow effect, the particle system in Unreal was used by creating a particle source at the auger boot location. After that, a number of particles with grain texture will be originated from the auger spout location with an initial velocity. The particles are subject to a constant downward acceleration to simulate the gravitation effect. Finally, the particles will disappear after they hit the grain bed mimicking the actual scenario in the field.

Figure 3.27 is one representative frame in Visualization 7 showing the simulation with the dynamic grain profile and grain flow incorporated. Figure 3.27a is the grain profile generated in the grain fill model in Simulink. After passing the profile into Unreal engine, Fig. 3.26b and 3.27c show the auger camera view and top-down view of this unloading processing respectively.

Lighting condition

Lighting conditions are important for stereo camera-based perception system as they could significantly impact the imaging quality or the overall look of the image, and thus potentially altering the perception results. Therefore, it is crucial to train and test the



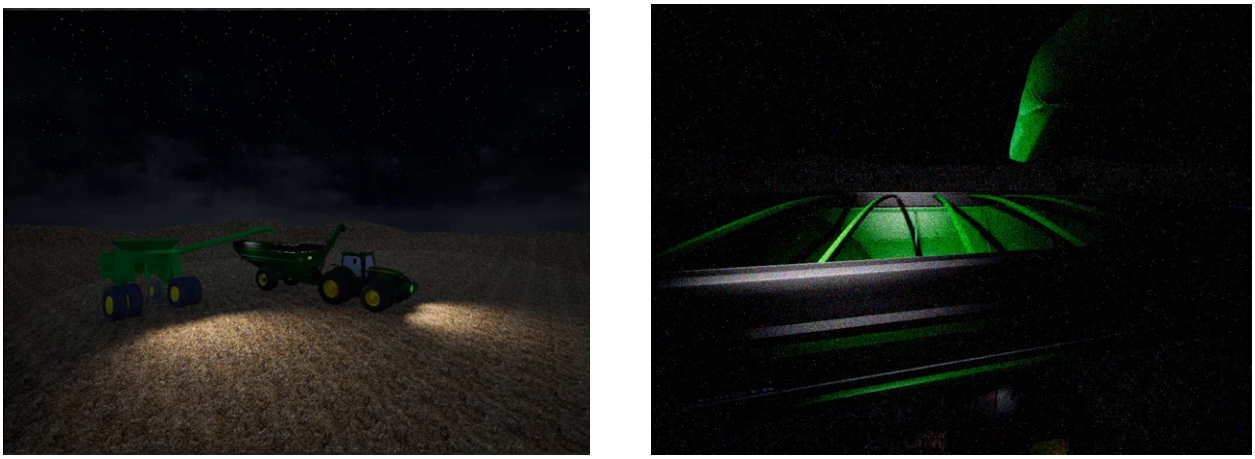
(a) (b)

(c)

Figure 3.27. Grain profile and grain flow visualization in perception model (associated with Visualization 7) (a) The grain profile simulated from grain fill model; (b) auger camera view in perception model; (c) top-down view of the unloading process.

perception system algorithms with different lighting conditions to enhance the system's robustness against various lighting. In this perception model, challenging lighting conditions are simulated include nighttime, rainy day, and dusty environment with direct sunlight.

Figure 3.28 shows the perception model simulation results for a night time unloading operation. As shown in Fig. 3.28a, the major light sources during night time become the artificial lights including the front lights on the combine and the tractor illuminating the field and the combine auger light illuminating the field. Moreover, Fig. 3.28b demonstrates the characteristics of camera images during night time. Compared with the simulation result for a day-time operation in Fig. 3.26b, the portion of the grain cart that is not illuminated



(a) (b)

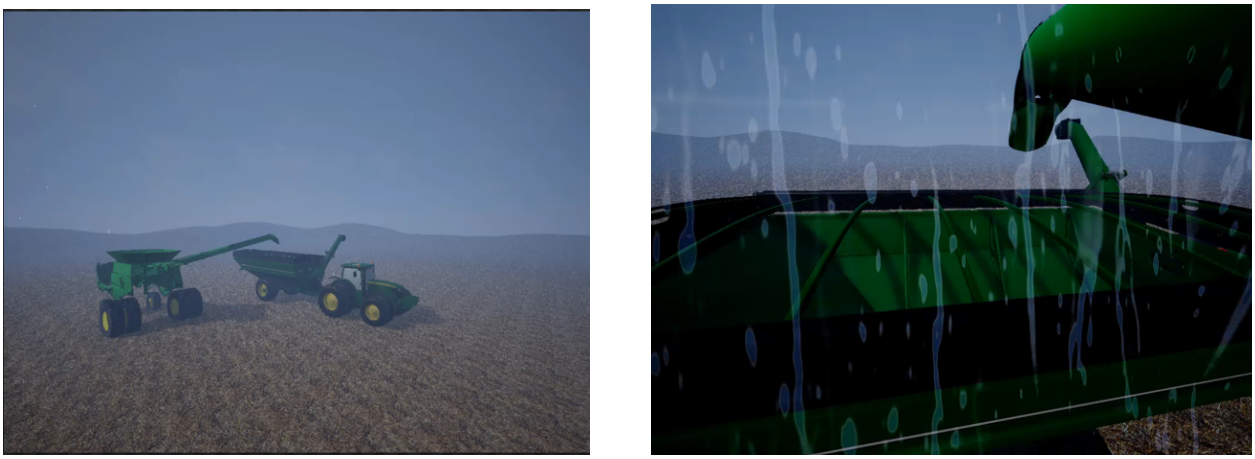
Figure 3.28. Perception model simulation for night time. (a) perspective view; (b) auger camera view.

by artificial light source becomes too dark to observe the features. Apart from the look of the image, the low lighting can also introduce additional sensor noise, degrading the image quality [66]. In the perception model, Gaussian noise is added to the image to simulate the imaging noise at low light condition, and the distribution of the Gaussian noise can be described as

$$p\left(z\right) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-(z-\overline{z})^2/2\sigma^2} \tag{3.19}$$

where z represents the intensity of an image, $p(z)$ is the probability density function of z, $\overline{z}$ is the average value of z, depending on the light intensity, and σ is the standard deviation of the intensity distribution, which is larger when the environment light becomes weaker.

Rain day is another challenging environment that could have a great impact on perception system visibility. In Fig. 3.29, the unloading operation in rain was simulated. Because of the overcast on a rainy day the sunlight is more diffusive. Besides, the contrast between illuminated area and shadow is less obvious compared with the nominal condition in Fig. 3.26a. Additionally, the foggy environment during the rainy day could reduce the transmissivity of light, and thus makes the scene at a farther distance blurry. On the simulated camera image in Fig. 3.29b, a dynamic raindrop effect is also added on the camera lens, because



(a) (b)

Figure 3.29. Perception model simulation for a rainy day. (a) perspective view; (b) auger camera view.

the raindrops that stay on the camera lens could cause significant occlusion, and the moving raindrop may confuse the algorithm.

A dusty environment with direct sunlight is a challenging unloading environment even for a human operator, so a perception system that could work on such a scenario will be more valuable. In Fig. 3.30, such an unloading environment is simulated in Unreal. Figure 3.30a shows a perspective view of the unloading process, where sunset creates a large shadow on the scene and the whole image has a warm tone. The impact of direct sunlight can be observed more clearly in the simulated camera image in Fig. 3.30b. Because of the high brightness of the sun in the background, the region of interest (i.e., the grain cart) becomes very dark. The large dynamic range of the image could pose a unique challenge for the perception system.

Another challenge in this scenario comes from the dust. The impacts from dust come from two sources, floating dust particles in the air and dust particles stuck to the camera lens.

Some moving dust particles were simulated with the particle system in Unreal with a translucent texture to mimic the dust generated from the impact of the grain flow. Although dust is translucent, for areas where dust particles are very dense, the dust could occlude the camera view to the grain profile behind. Besides, because dust particles often have

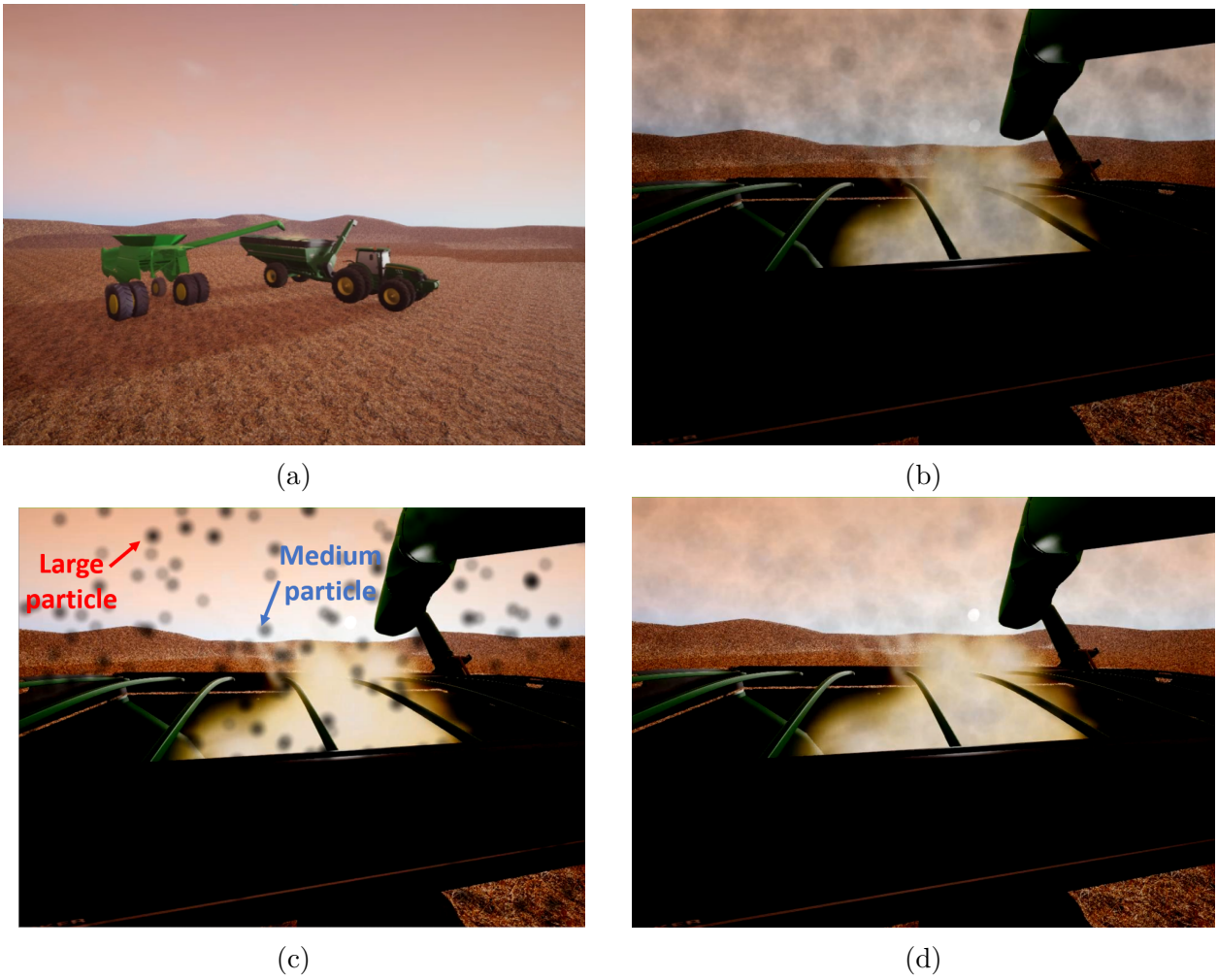


Figure 3.30. Perception model simulation for a dusty environment with direct sunlight. (a) perspective view; (b) auger camera view with 40,000 small dust particles on lens; (c) auger camera view with 300 large or medium dust particle on lens; (d) auger camera view with 10,000 small particles on lens.

a lighter color than the grain or cart, when the scene is illuminated by direct sunlight as Fig. 3.30b- 3.30d, the bright color could further exacerbate the impact of large dynamic range in the scene.

The impact of dust on the lens is simulated in MATLAB with the method introduced in [67]. This method calculates the dust impact by calculating the light rays transmitting from physical objects to the camera imaging sensor, and the amount of light occluded by the particle on the lens. By assuming that the aperture in the camera is circular, in geometric optics, its projection on the lens plan is also circular. Therefore, the occlusion effect calcu-

lation can be approximated by the convolution between the projected aperture and the dust particle. In summary, there are 3 different cases for dust occlusion depending on the dust particle size relative to the camera aperture:

1. Large dust particles will create a total occlusion area at the particle's center and the occlusion effect fades out at its edge. Fig. 3.30c shows some examples of large dust particles (one marked with a red arrow).

2. Medium-sized particle does not create a total occlusion. Its occlusion effect also peaks at the center and fade through the edge. Fig. 3.30c shows some examples of medium dust particles (one marked with a blue arrow).

3. Small particles does not create a total occlusion, and its occlusion effect plateau at the center of the dust and decreases at the edge. Fig. 3.30b and 3.30d show example image with small particles on the lens only. This is the most common case for a camera attached to a harvester as the dust particles in the field usually have a small size. In comparison, Fig. 3.30b simulates a lens with more dust particles, and thus the whole image is darker than Fig. 3.30d.

3.3.3 LiDAR data simulation

Another commonly used perception sensor for automation and robotics is the LiDAR (Light detection and ranging) sensor. However, unlike camera, Unreal does not have native support for the LiDAR data generation. Therefore, a LiDAR model was developed in the perception model based on the camera model discussed in Sec. 3.3.2 used in the Unreal engine.

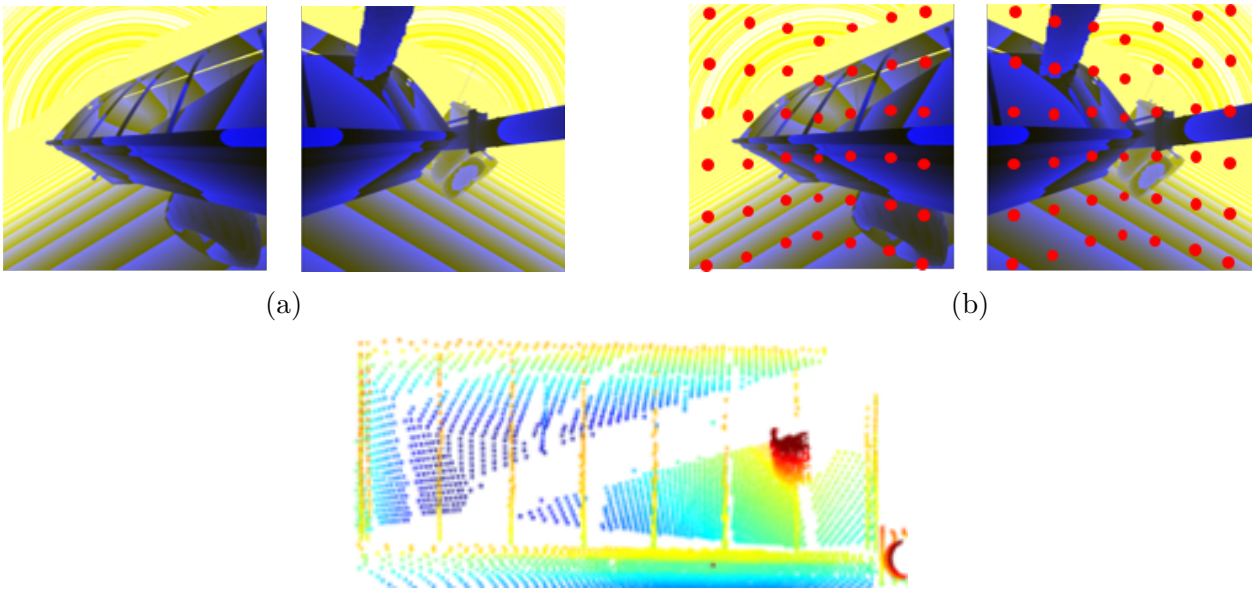
The LiDAR sensor was modeled in four steps:

1. Because the data is transmitted in the form of image between the Unreal engine and Simulink. The depth information of the scene is first encoded into depth images.

2. Since LiDAR is a 3D sensor, the 3D coordinate for each camera pixel was then subsequently retrieved from the depth image.

3. Because the LiDAR sensor and camera sensor sample the world in different patterns, a data resampling was required to generate point cloud data following the LiDAR pattern.

4. To emulate the sensor error in an actual LiDAR sensor, sensor noise was added to mimic LiDAR sensor error.



(a) (b)

(c)

Figure 3.31. LiDAR data simulation. (a) depth encoded images; (b) LiDAR data resampling; (c) simulated LiDAR point cloud.

Step 1: Render camera image with depth information (in Unreal)

A LiDAR sensor provides a 3D measurement of the environment geometry. However, the virtual camera in Unreal only provides a 2D RGB image, and does not directly provide depth information for the camera sensor. To simulate a 3D LiDAR sensor, 3D information from the sensor point of view is required, and therefore must be calculated. Although Unreal does not directly provide a 3D sensor model, it allows users to retrieve the depth information for each pixel in the rendered image. The authors developed a pixel shader to encode the depth information of each pixel into color values (as shown in Fig. 3.31a).

In Unreal the depth information is a float number and the color image is stored by an 8-bit integer for each R, G, B channel. To encode the depth value into color image with a broad range and high precision, the following equations are used for depth encoding:

$$R(u,v) = G(u,v) = round(\min((\frac{d(u,v)}{\alpha})^{\gamma}, 1) \times 255) \tag{3.20}$$

$$B(u,v) = round((\frac{d(u,v) \mod \beta}{\beta})^{\gamma} \times 255) \tag{3.21}$$

where $R(u,v)$, $G(u,v)$, $B(u,v)$ are red, green and blue channel of pixel (u,v) on the depth encoded color image. Each channel is stored by an 8-bit integer so the value is within $[0, 255]$. $d(u,v)$ is the depth value to be encoded with the unit of cm, α and β are constant scaling factors with $\alpha > \beta$. α should be larger than the maximum value of d to avoid data truncation. The *mod* is the modulo function and *round* is the rounding function. The gamma correction constant $\gamma = 2.2$. Figure 3.32 is an example of the depth encoding implementation in the Unreal visual scripting editor with $\alpha = 1000$ and $\beta = 100$.

One may notice that only use two color channels were used for depth encoding, specifically, the red and green channel stores digits of d with greater place value to obtain a large number range, and B channel stores digits of d with smaller place value to achieve high precision. Using two channels is adequate to encode the depth value required by our applications, and the same encoding method can easily be extended to three channels if a larger range or higher precision is required.

The screen manufacturers usually apply a power-law function on image intensity to make the monitor color more vivid to human eyes [68], [69]. This is called gamma correction and it is not desirable for 3D rendering. To counteract the effect of the monitor's gamma correction, Unreal internally applies an inverse gamma correction on texture color. This will cause inconsistency between the depth encoded image rendered by Unreal and the depth value to be embedded. Therefore, a gamma function was added in the depth encoding before Unreal renders the texture with inverse gamma correction.

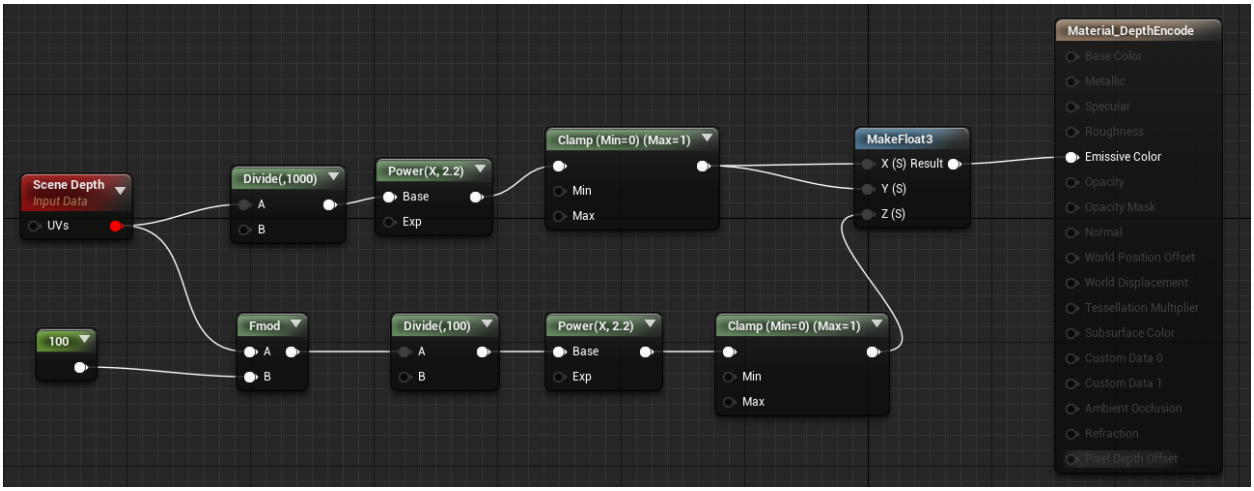


Figure 3.32. An example of the depth encoding implementation.

Step 2: Retrieve 3D coordinate from depth image (in Simulink)

After generating the depth encoded image, the 3D coordinate for each pixel (u, v) on the image can be retrieved. Since the depth value is encoded into R, G, B channels of a color image, the decoding can be achieved in an inverse manner using Eq. 3.22

$$d'(u,v) = \left[round(\frac{R(u,v)}{256} \times \frac{\alpha}{\beta} - \frac{B(u,v)}{256}) + \frac{B(u,v)}{256} \right] \times \beta, \tag{3.22}$$

where $d'(u, v)$ is the decoded depth value at pixel (u, v). The depth $d'(u, v)$ is the Z coordinate on camera sensor coordinate. To retrieve 3D information from a 2D depth image, the camera model used in Unreal needs to be investigated.

Unreal uses a basic pinhole model [70] as shown in Fig. 3.33 to simulate how the camera sensor captures an image of the 3D world. O^c is the camera center and $X^cY^cZ^c$ is the camera 3D coordinate. The image plane is perpendicular to the Z^c axis with a constant distance f called focal length and Z^c passes through the center of the image plane. A 3D point $P(X, Y, Z)$ is mapped to a 2D point $P_c(u, v)$ on the camera image where $P_c(u, v)$ is the intersection between the line O^cP and the image plane. Equation 3.23 describes the mapping relationship from a 3D point to a 2D image pixel

$$\overrightarrow{CP} = (\bar{a}u + \bar{b}v + \bar{c})W \tag{3.23}$$

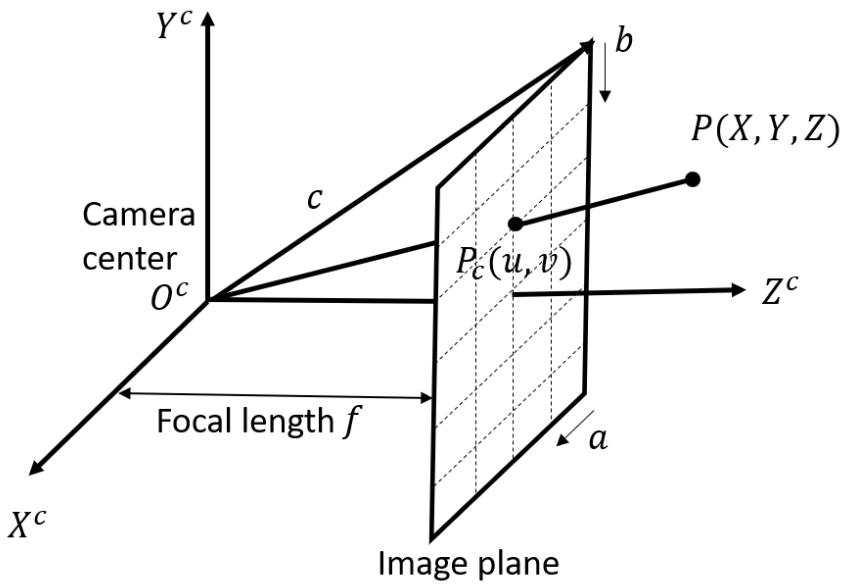


Figure 3.33. Illustration of the pinhole camera model

Since $P = [X, Y, Z]^T$ and $C = [0, 0, 0]^T$, Eq. 3.23 can also be written as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \bar{a} & \bar{b} & \bar{c} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} W. \tag{3.24}$$

Here $\bar{a}$ and $\bar{b}$ are unit vectors of two axes of the image plane with $\bar{a} = [1, 0, 0]^T$ and $\bar{b} = [0, -1, 0]^T$. $\bar{c}$ is the vector from camera center to the origin of the image plane. For a camera sensor with $w \times h$ resolution and $hfov$ degrees horizontal field of view, $\bar{c} = \left[-\frac{w}{2}, \frac{h}{2}, \frac{w}{2\tan(hfov/2)}\right]^T$. W is the augmented dimension in homogeneous coordinate that is used for simplifying the calculation. In Eq. 3.24, the coordinate of each camera pixel (u, v) is known, and the decoded depth $d'(u, v)$ from Eq. 3.22 is the Z coordinate of P. Therefore, there are three unknowns X, Y, W and three equations in Eq. 3.24, the 3D coordinate $P(X, Y, Z)$ can be solved for each pixel $P_c(u, v)$ of the depth image.

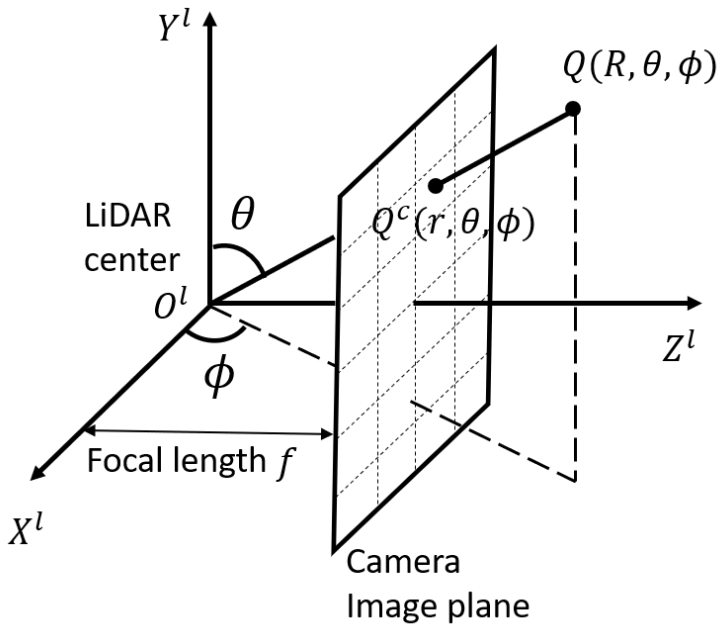


Figure 3.34. Illustration of the LiDAR model built upon the camera model

Step 3: Resample camera image based on LiDAR patterns (in Simulink)

Camera and the LiDAR sensors sample the world in different ways. As shown in Fig. 3.33, the camera pixels on the image plane distribute evenly in horizontal and vertical directions. On the other hand, the LiDAR sensor samples the world evenly in angle instead of in distance. After getting the depth encoded image from Step 1, the image is resampled based on LiDAR's scanning patterns. As illustrated by Fig. 3.34, the LiDAR sensor scans the world with evenly spaced angles and thus its point cloud data is typically represented by spherical coordinates with a distance R, an azimuth angle θ, and a polar angle ϕ. The goal is to derive the LiDAR model from the camera model and thus LiDAR coordinate $X^lY^lZ^l$ is aligned with the camera coordinate. Then the image point $Q^c(r,\theta,\phi)$ corresponding to Q can be determined on the image plane. Since Q^c and Q lie on the same LiDAR ray, they share the same angles θ and ϕ. However, Q^c has to be on the image plane and it has a different distance from Q. Consider the fact that the camera image plane is perpendicular

to Z-axis with a constant distance of focal length, the distance r can be computed based on the focal length and two angles:

$$r = f/\sin(\theta)/\sin(\phi). \tag{3.25}$$

Here $f = \frac{w}{2\tan(hfov/2)}$, θ is integer times of LiDAR vertical resolution and ϕ is integer times of horizontal resolution. Now the spherical coordinate of Q^c can be converted into the camera image pixel $\left(u^l, v^l\right)$ by Eq. 3.26.

$$\begin{aligned} u^l &= round(r\sin(\theta)\sin(\phi) + w/2) \\ v^l &= round(r\cos(\theta) + h/2) \end{aligned} \tag{3.26}$$

Through Eqs. 3.25 and 3.26, each LiDAR sample point is mapped onto one camera pixel $\left(u^l, v^l\right)$ and then extract the 3D coordinate of this pixel from depth image using Eq. 3.24 in Step 2. Typically the LiDAR sensor has a larger horizontal field of view than what a camera sensor has. Therefore, two camera models with the same location but different viewing angles are stitched together to simulate one LiDAR sensor with an ultra-wide field of view.

Step 4: Add sensor noise (in Simulink)

The LiDAR data generated in Step3 is an ideal sensor data without any noise. The developed LiDAR model also provides an option to add sensor noise and users can select either use ideal clean sensor data or use noisy sensor data. This option is helpful for data processing algorithm development. Algorithm development usually start with clean data to build the basic algorithm architecture, and is then tested with different levels of noisy data and iterate algorithm design to achieve a robustness performance in noisy conditions. Assume that the sensor noise happens on distance measurement with a Gaussian distribution. For a clean LiDAR data point $Q(X, Y, Z)$, the sensor noise $n \sim \mathcal{N}(\mu, \sigma^2)$ is added along direction $\mathbf{r} = (X, Y, Z)$.

Figure 3.31c shows a simulated LiDAR sensor data using the developed LiDAR sensor model. The LiDAR model allows simulation of different LiDAR sensor specifications including field of view, angular resolution, minimum and maximum measurement range.

3.3.4 Perception model application

Camera placement

The placement of a stereo camera for a camera-based perception system can impact the perception quality significantly as different camera locations result in different perspectives and different coverage on the objects of interest. Optimizing the camera placement is an iterative process between design and testing. However, designing and testing with in-field testings are time-consuming and costly, especially for multiple machine configurations. Therefore, the perception model can be used to simulate the impact of camera placement before testing in the field.

(a) (b)

Figure 3.35. Camera image simulation results with different camera placement. (a) combine auger; (b) combine body.

The most important function of the perception in the automatic unloading system is to provide fill status feedback for the controller, so the camera needs to be looking at the grain profile. This leaves two options for the camera placement, combine body or the combine auger. Figure 3.35 compares the imaging results between placing the camera on the auger or placing the camera on the combine body. Comparing with the placement on combine body, by placing the camera on the combine auger, the camera can get a better perspective on the grain cart and thus a better coverage of the profile in lateral location. Additionally, the higher location also allows the camera to start seeing the grain profile before it becomes

higher than the grain cart edge, which provides a longer reaction time for the controller. On the flip side, placement on the auger has a smaller coverage of the grain profile in the longitudinal direction, but it can still cover most portion of the cart. Overall, the advantages of on-auger placement outweigh its disadvantages, so placement on auger is more preferable.

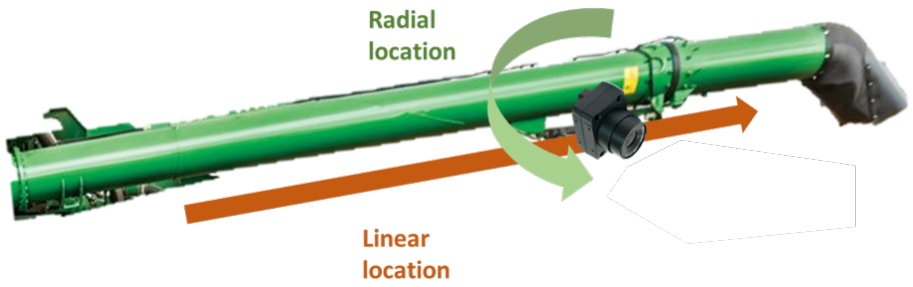
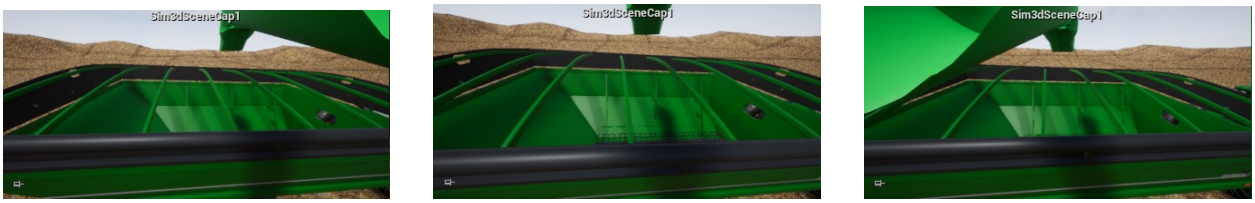


Figure 3.36. Camera placement options on combine auger.

(a) (b) (c)

Figure 3.37. Camera image simulation results with on-auger camera placement in different linear location. (a) closer to auger boot; (b) middle of the auger tube; (c) far from the auger boot.

For on-auger placement, the placement can be further optimized for different radial locations and linear locations as shown in Fig. 3.36. Therefore, different linear locations were compared in Fig. 3.37: with a closer camera location to the auger boot, the camera can have a better perspective to the grain profile (more top-down view). However, on the other hand, the closer placement to the auger boot also limits the camera coverage. An extreme case will be similar to Fig. 3.37a, in which the camera can only see the farther edge of the grain cart, making it challenging for cart edge detection. Therefore, an optimized linear location will be a location in the middle such that the camera can have a good perspective for great profile view while still be able to both edges of the grain cart for more robust cart



(a) (b) (c)

Figure 3.38. Camera image simulation results with on-auger camera placement in different radial location. (a) lower left of the auger tube (b) right down below the auger tube; (c) lower right of the auger tube.

tracking. The influence of radial location as shown in Fig. 3.38 is most from the occlusion of the auger. Although placing the camera right down below the auger will minimize the occlusion as alternative placement at the lower left (Fig. 3.38a) and lower right (Fig. 3.38c) of the auger could improve the perspective without occluding the grain cart.

The camera placement practice above is based on qualitative assessment. A more optimized placement requires quantitative assessment, which can also be achieved with the perception model simulation with the evaluation framework discussed in Fig. 3.39.

Perception system evaluation

An evaluation pipeline was proposed to validate the perception system performance with the perception model in Fig. 3.39. The perception system, instead of taking the image from a physical stereo camera, uses the virtual raw image simulated by the virtual stereo camera in the perception model. In the meantime, the output from the perception system, the perceived scenario will be compared with the ground truth scenario provided by the perception model. The ground truth scenario in the perception model does not need to be manually labeled like real-world data, because the virtual scenario is pragmatically defined. Finally, the difference between the perceived scenario and the ground-truth scenario can be used to quantify the performance of a perception system. With quantified analysis results for the perception system performance, one can iterate on the perception system algorithm or design.

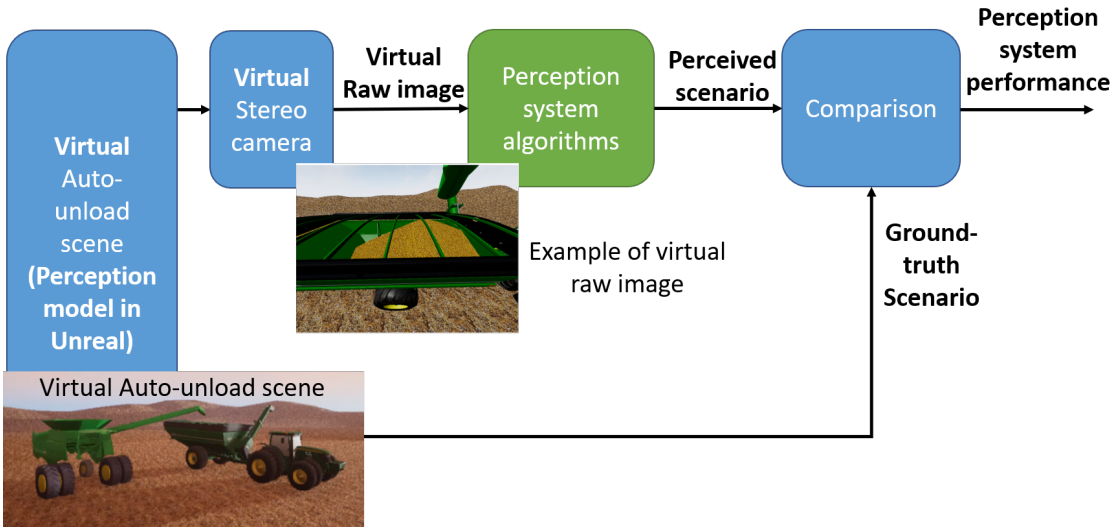
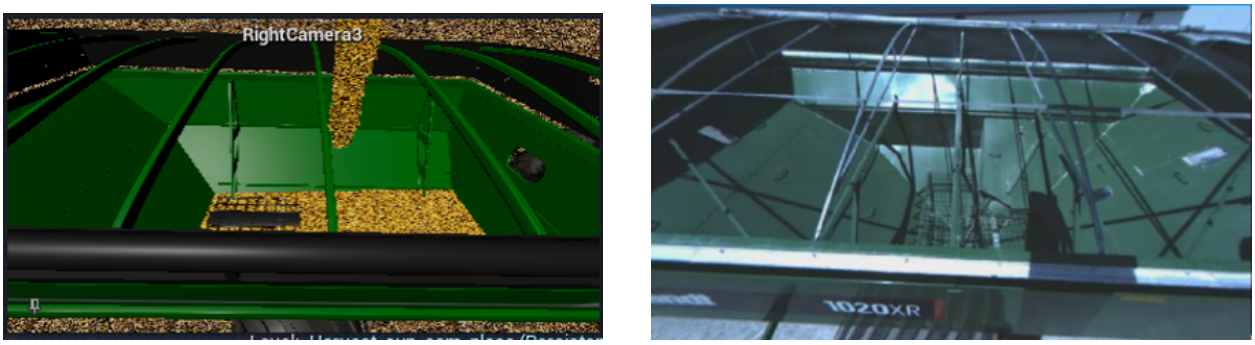


Figure 3.39. Perception system evaluation pipeline with perception model in Unreal.

3.3.5 Experimental validation

To validate the accuracy of the perception model and further improve the perception model configurations, in-field testing was conducted by unloading on the go in different conditions with a stereo camera on the auger to collect image data.

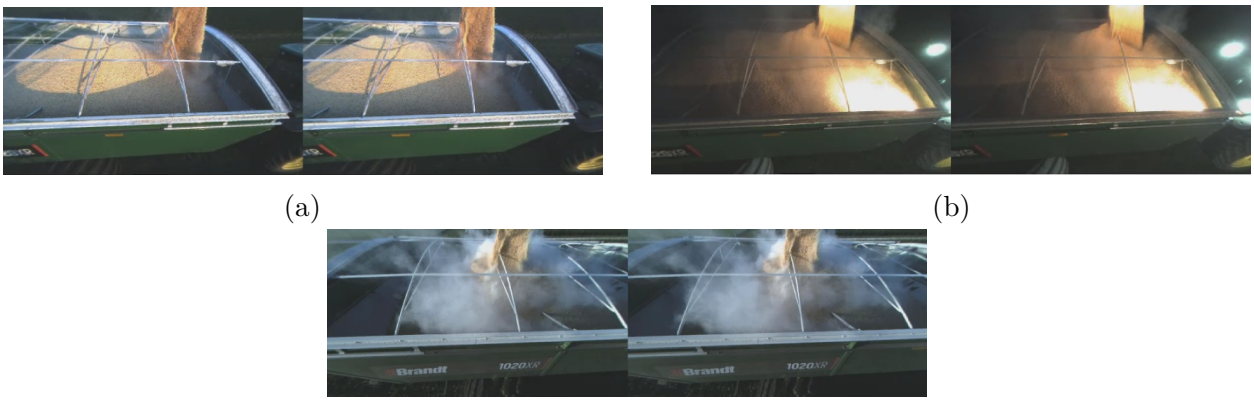


(a) (b)

Figure 3.40. Images from camera on the designed location. (a) simulated image; (b) experimental image.

The placement of the camera is designed and simulated in the perception model. Figure 3.40 shows a comparison between the simulated image and the experimental image when the camera is placed at the designed location. Both images show a similar perspective and

coverage of the grain cart, demonstrating that the perception model can be a useful tool to simulate the impact of camera placement.



(a) (b)

(c)

Figure 3.41. Experimental stereo images from stereo camera on auger in different test conditions. (a) normal sunlight; (b) night time; (c) heavy dust.

After placing the camera on the auger, stereo camera images in various conditions were collected as shown in Fig. 3.41. The image on the left is from the left lens in the stereo camera and the image on the left is from the right lens. Images during unloading were collected with nominal sunlight as shown in Fig. 3.41a, which shows a similar shadow effect from the sun as the simulated image in Fig. 3.27b. Figure 3.41b shows the stereo camera at night time. Compared with Fig. 3.28b, the experimental images for night time has similar noise effect but its different lighting configuration causes illumination on different part of the image. However, the information revealed from both simulated and experimental data are similar, the most critical issues for night time operation are non-uniform illumination and image noises. Figure 3.41c shows the experimental data from a dusty unloading. Similar to the simulated image in Fig. 3.30d, the experimental image demonstrates a similar translucent occlusion effect on the camera view from dust. However, because the lens used in the in-field testing has not been exposed to dust for an extended period of time, the dust-on-lens effect is not as strong as the simulated image.

4. AUTOMATIC UNLOADING CONTROLLER DESIGN AND SIMULATION

4.1 Controller design

A movement controller was designed to close the automatic offloading control loop shown in Fig. 4.9 and simulate the automatic offloading process in a virtual environment.

4.1.1 Fill strategy

The automatic offloading controller calculates the desired location and auger on/off based on the current fill status, current impact location, and user-desired behavior. Because a grain cart is usually longer than it is wide, the relative vehicle movement in the longitudinal direction has a more significant impact on the fill quality. Therefore, to reduce the system complexity, the automatic offloading controller keeps the auger at the center in the lateral location and only controls the auger movement in the longitudinal direction.

To automatically determine the desired auger location, the automatic offloading controller first partitions the grain profile, which is a 2D array $h(x, y)$, in the longitudinal direction into R rows. Within each row, the fullness of the cart can be calculated based on the average height of all the grids in this row (other fullness methods can also/instead be used). The algorithm compares the average fill level in each row with the desired fill level specified by the operator to binarize the fullness within each cart into "full" or "not full". The first and last row are always be marked as "full" to prevent spillage.

$$F_{\mathrm{i}} = \begin{cases} 1, & \overline{h_{\mathrm{i}}(x,y)} > h_{des} \quad or \quad \mathrm{i} = 1, R \\ 0, & Otherwise \end{cases} \tag{4.1}$$

where F_{i} is the binary fullness of the i-th row, $\overline{h_{\mathrm{i}}(x,y)}$ is the average grain height in the i-th row, and h_{des} is the desired grain height based on the target fill level specified by the operator.

The desired location from the automatic offloading controller is determined by the binary row fullness and the fill strategy specified by the operator. Three commonly-used strategies were implemented in the automatic offloading controller:

- Front to back fill strategy (F2B): Filling the grain cart from front to back. The automatic offloading controller looks for the first "not full" row from the front of the cart and sets it as the desired location.
- Back to front fill strategy (B2F): Filling the grain cart from back to front. The automatic offloading controller looks for the first "not full" row from the back of the cart and sets it as the desired location.
- Front to back to front fill strategy (F2B2F): Filling the grain cart from front to back and then top off the cart to the desired fill level from back to front. The algorithm runs the F2B strategy with a lower fill target and then run the B2F strategy to top off the cart to the desired level.

In addition to turning the auger off according to the Fill strategy, the automatic offloading controller turns it off, if for any reason, the auger moves too close to any cart edge to prevent spillage.

4.1.2 Movement controls

Two different options were considered for generating the position command for the Machine Sync system: "open-loop" movement controls and "closed-loop" movement controls. Figure 4.1 shows the block diagram of open-loop control. Figure 4.2 shows the block diagram of the closed-loop movement controller.

The open-loop control approach sends nudges to the Machine Sync system solely based on the desired location from the automatic offloading fill strategy $x_{desired}$. Because Machine Sync location is defined between the tractor and the combine, the automatic offloading controller first converts the desired location from the auger-grain cart coordinate system to the tractor-combine coordinate system based on the vehicle geometry. After that, since Machine Sync takes nudge commands to update its target location by a fixed distance, the

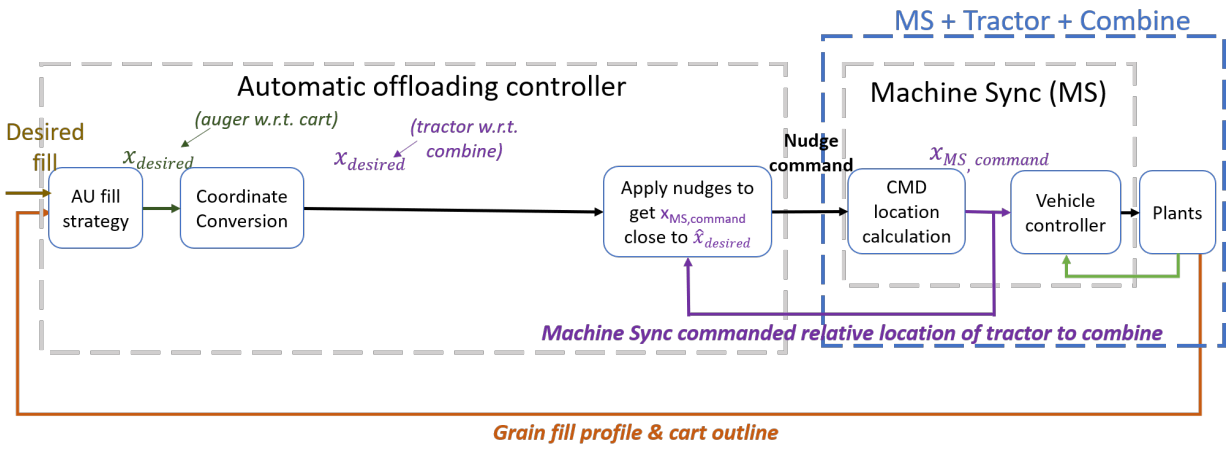


Figure 4.1. automatic offloading controller block diagram with open-loop movement controls. Note: the automatic offloading system also includes the perception system, but it is not explicitly included in this figure.

automatic offloading controller translates the desired location to nudge commands with a nudge handler. The nudge handler commands nudges to get the target location in Machine Sync $x_{MS,commanded}$ as close to the desired location $x_{desired}$ as possible.

One underlying assumption of the open-loop controller is that the Machine Sync system will successfully control the relative location between the two vehicles. However, in practice, external disturbance or plant uncertainty could result in a tracking error of the Machine Sync system and the assumption would no longer hold true. Consequently, open-loop controls is not expected to be as robust to disturbance or plant uncertainty.

As a result, another movement control strategy was proposed and developed: the closed-loop movement control strategy shown in Fig. 4.2. The closed-loop controller does not directly command nudges based on desired location from automatic offloading fill strategy $x_{desired}$. Instead, it calculates an adjusted desired location $\hat{x}_{desired}$ based on the tracking error, which is the difference between the desired location $x_{desired}$ and the actual location x_{actual}. By tracking the error, the closed-loop controller is expected to be more robust than the open-loop controller.

The closed-loop movement controller was synthesized based on the Machine Sync dynamic model identified from in-field testing as described above in sec. **??**. Given that during an

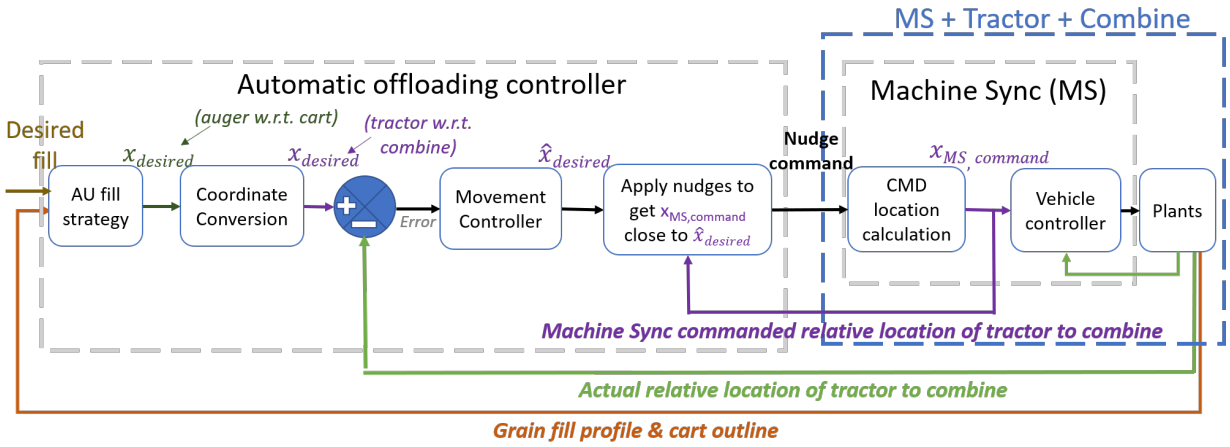


Figure 4.2. automatic offloading controller block diagram with closed-loop movement controls.

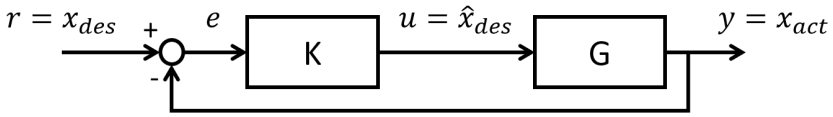


Figure 4.3. SISO controller design for small nudge plant.

unloading process most auger movements are shorter than one meter, the small nudge plant of Machine Sync was treated as the nominal plant to design the controller. Additionally, because the smallest nudge size of the Machine Sync is six inches, the discretization error from the nudge handler is negligible. Consequently, the movement controller design for the Machine Sync system can be described as a classic single-input-single-output (SISO) system as shown in Fig. 4.3. In Fig. 4.3, K is the controller to be synthesized, G is the Machine Sync small nudge plant, the reference input r is the desired location x_{des}, the control effort u is the adjusted desired location $\hat{x}_{desired}$, and the system output y is the actual relative location x_{act}. After defining the variables, the following transfer functions related to the system can be defined as

$$\begin{aligned} S &= (1 + KG)^{-1} = \frac{E}{R} \\ T &= 1 - S = \frac{Y}{R} \\ KS &= \frac{U}{R} \end{aligned} \tag{4.2}$$

where S is the sensitivity function and the transfer function from the reference input r to tracking error e, T is the complementary sensitivity function and the transfer function from reference input r to system output y, and KS is the transfer function from reference input r to control effort u.

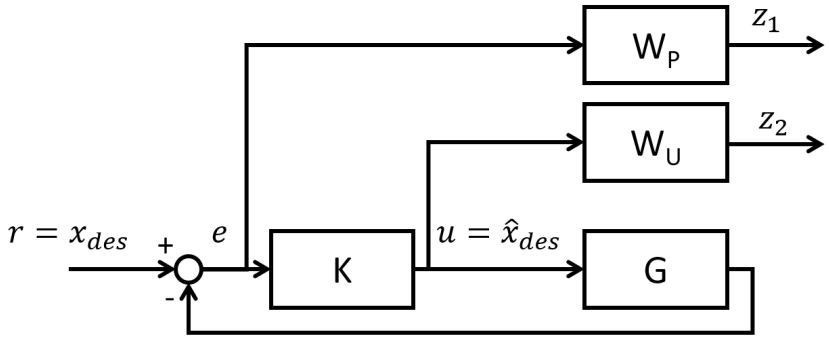


Figure 4.4. Augmented block diagram for H_∞ mixed-sensitivity loop shaping.

The controller was synthesized with H_∞ mixed-sensitivity loop shaping technique [71]. First, two weighting functions W_P and W_U were formulated to penalize the tracking error and control effort respectively as shown in Fig. 4.4. As a result, the transfer function M for the augmented plant from r to z can be described as

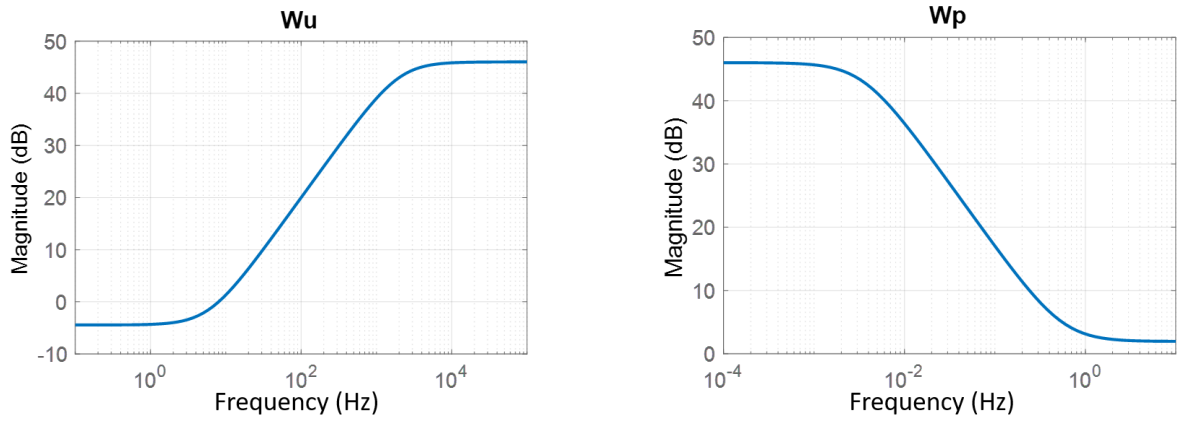
$$M = \begin{bmatrix} W_P S \\ W_U K S \end{bmatrix} = \frac{Z}{R} \tag{4.3}$$

and the overall system requirement for controller synthesis is

$$\|M\|_\infty = \max_\omega \sqrt{|W_P S|^2 + |W_U K S|^2} < 1 \tag{4.4}$$

Figure 4.5a shows the specification to limit the controller control effort magnitude. In this way, Machine Sync is less likely to switch from small nudge plant to large nudge plant, and the high-frequency cut-off on W_U is designed to prevent the abrupt change of the Machine Sync command to avoid large vehicle acceleration or jerk.

Figure 4.5b shows the controller specification on tracking error. The large penalty on low frequency reduces the steady state error, and the cut-off frequency of 0.7 Hz maintains the responsiveness of the controller.



(a) (b)

Figure 4.5. System specification for controller synthesis. (a) weighting function for control effort; (b) weighting function for tracking error.

With this specification of W_P and W_U, the H_∞ optimal controller is obtained by solving the following optimization with methods proposed in [72], [73]

$$\min_K \|M(K)\|_\infty \tag{4.5}$$

Figure 4.6 shows the Bode plot of the loop transfer function $L = KG$, which reflects the frequency response of the closed-loop system. The closed-loop system has a high phase margin of 85^o while maintaining a grain-crossover frequency of 0.49 rad/s.

In the time domain, the performance of the closed-loop controller can be evaluated by two criteria: reference tracking and disturbance rejection. Reference tracking evaluates how fast and accurate the output of the system tracks the reference input. Figure 4.7a shows a simulation comparison between the closed-loop system and the open-loop system when there is a 0.8-m step input. With the addition of a controller, the closed-loop system settles in 7 seconds, while the open-loop system takes 18 seconds to settle (evaluated by 5% settling time). In addition, the closed-loop controller eliminates the overshoot in step response, reducing the oscillation of the system. Figure 4.7b compares the disturbance rejection performance of the system when there are a 0.2-m input disturbance and no change to the reference input. The open-loop system, because it does not account for the tracking

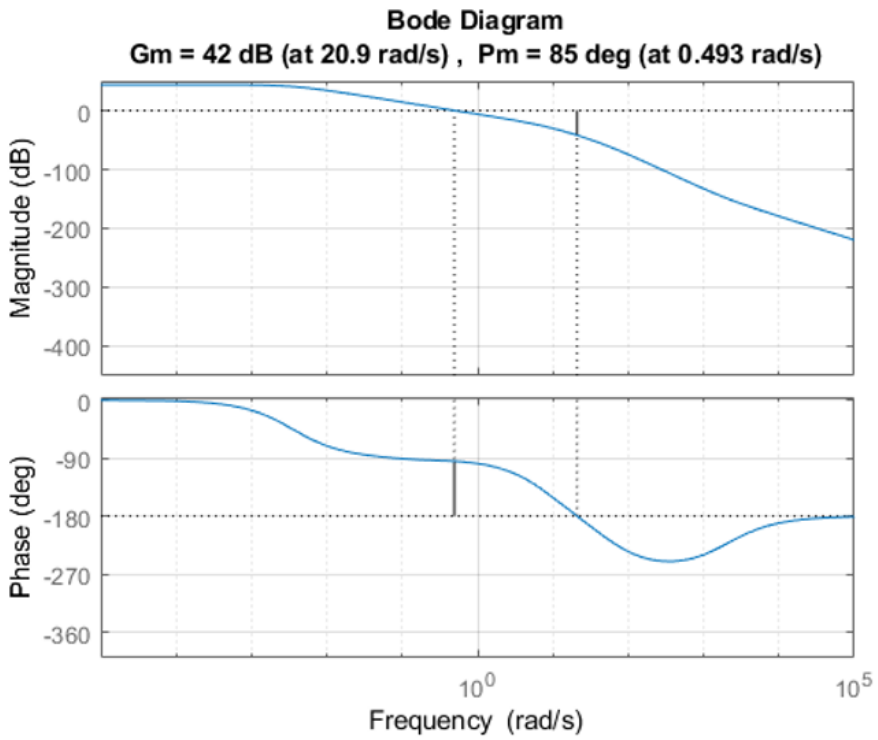


Figure 4.6. Bode plot of system loop transfer function with H_∞ controller.

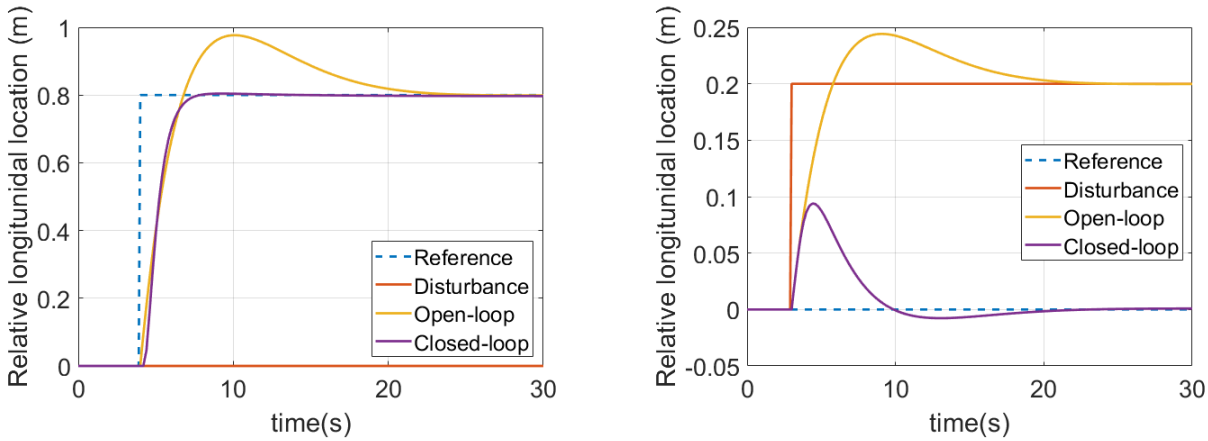


Figure 4.7. Comparison between closed-loop movement controls and open-loop movement controls for small nudge plant (nominal plant): (a) reference tracking; (b) disturbance rejection.

error, fails to reject the disturbance and results in a tracking error of 0.2 m. On the contrary,

the closed-loop system rejects the impact of disturbance and settles to the reference input in 13 seconds.

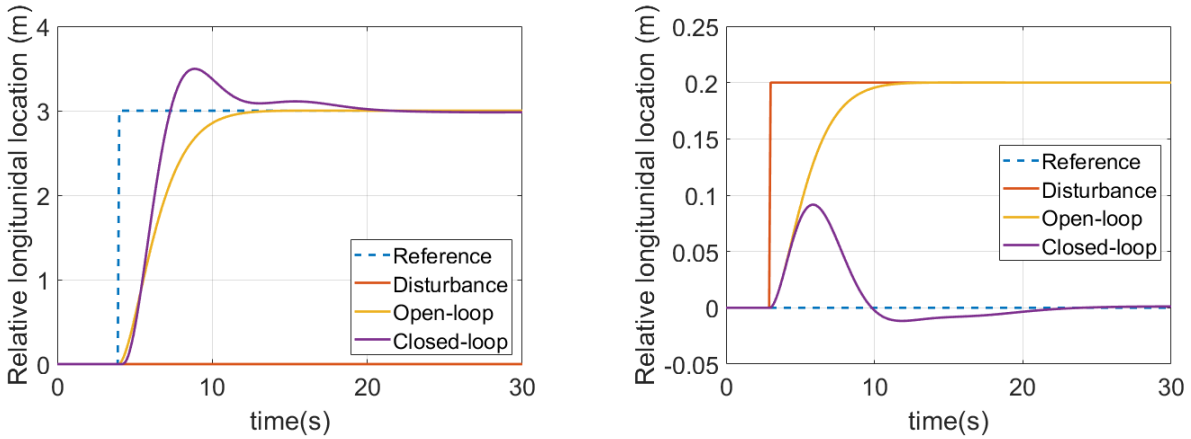


Figure 4.8. Comparison between closed-loop movement controls and open-loop movement controls for large nudge plant: (a) reference tracking; (b) disturbance rejection.

During unloading operation, even though most desired location changes are smaller than 1m, depending on the grain profile or disturbance to the system, the control strategy can request an auger position change greater than 1 m. Therefore, the closed-loop controller also needs to be stable when the Machine Sync system is the large nudge plant. To evaluate the closed-loop system performance with a large nudge plant, the system response with both the reference tracking and disturbance rejection test were simulated as well.

Figure 4.8a shows the reference tracking comparison between the closed-loop and open-loop system with Machine Sync large nudge plant when there is a 3-meter step input. The closed-loop controller still has a stable response when controlling the large nudge plant and has a similar 5% settling time. However, compared with the open-loop result, the closed-loop controller introduces a slight overshoot to the system for about 4%. For the disturbance rejection test shown in Fig. 4.8b with a 0.2-meter disturbance, the closed-loop controller continues to show better performance as the closed-loop system can reject the input disturbance in about 25 seconds but the open-loop system ends up with a 0.2-meter tracking error.

4.2 Automatic offloading system model simulation

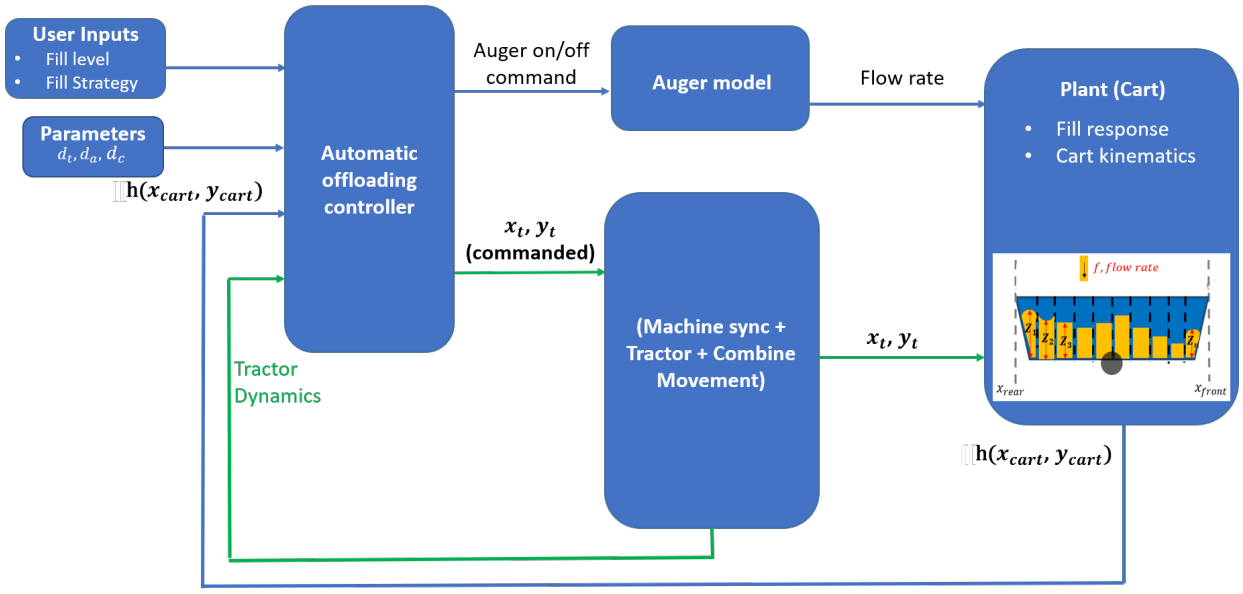


Figure 4.9. Simplified high-level system model architecture

The vehicle models, the grain fill model and the automatic offloading controller were implemented in Simulink to simulate a complete unloading-on-the-go procedure. Figure 4.9 shows the high-level architecture of the simulation. As discussed in sec. 4.1.1, automatic offloading is achieved by controlling the auger movement in the longitudinal direction. In the lateral direction, the auger location was kept constant, with tracking control done via Machine Sync. Nonetheless, the approach taken for longitudinal control is expected to work in the lateral direction as well, if lateral controls are deemed beneficial.

The unloading-on-the-go process from a combine harvester (model: John Deere S660) to a tractor (model: John Deere 8345R IVT) towing a grain cart (model: Parker 1348) was simulated. The unloading is on a flat terrain and the vehicles are moving in straight lines. In the lateral direction, the two vehicles maintain a fixed distance to keep the auger in the middle of the grain cart. In the longitudinal direction, the automatic offloading controller sends command messages to Machine Sync to control the relative position of two vehicles. The simulation results in this section do not take into account the impact of combine hopper capacity. Consequently, the unloading finished in one continuous event.

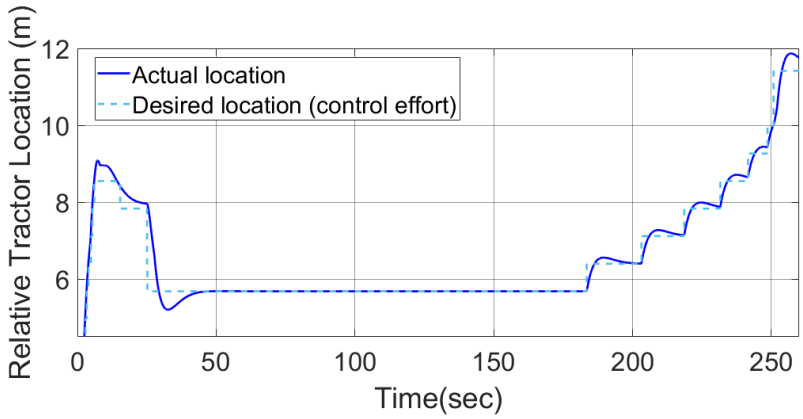
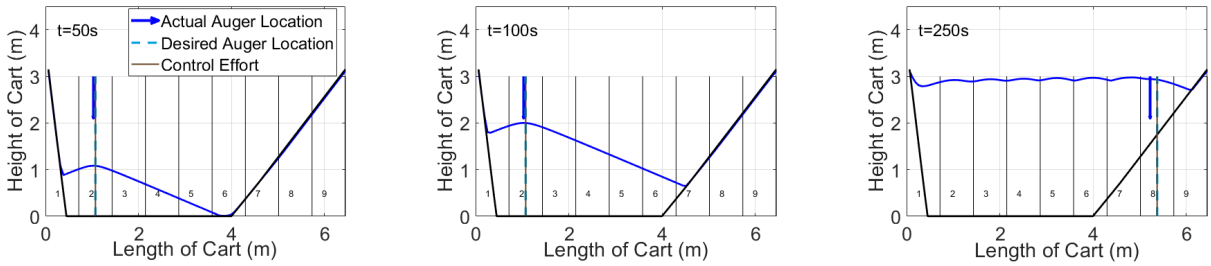


Figure 4.10. Relative grain cart position from automatic offloading simulation with open-loop controller.



(a) (b) (c)

Figure 4.11. Grain profile from automatic offloading simulation with open-loop controller (associated with Visualization 8). (a) t = 50 s; (b) t = 100 s; (c) t = 250 s.

The first automatic offloading simulation runs with a Front to Back strategy and an open-loop movement controller (per Fig. 4.1). The controller divides the grain cart into 9 rows in the longitudinal direction and controls the auger to fill up the grain cart from empty to full. Visualization 8 shows the longitudinal cross-section of the grain cart filling status during the unloading. Figure. 4.11 shows three representative frames during the unloading process in Visualization 8. Because the controller fills up the grain cart from front to back, the grain cart has full rows at the front and is being filled at the back. Figure. 4.10 shows the auger location relative to grain cart. As the desired location moves from front to back, the commanded location to Machine Sync stays the same as the desired location due to the open-loop movement control strategy.

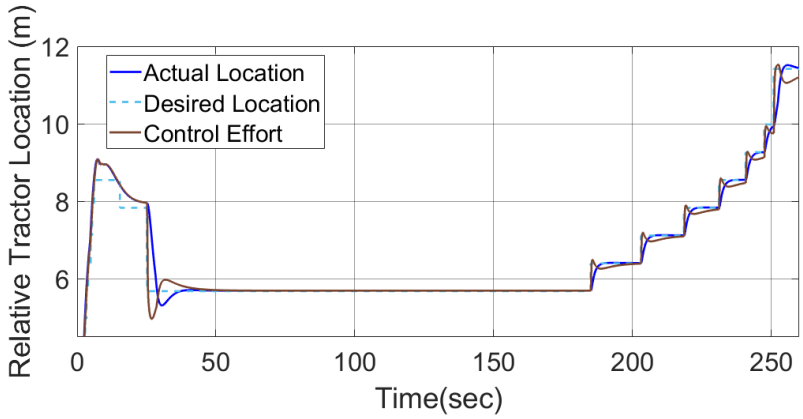
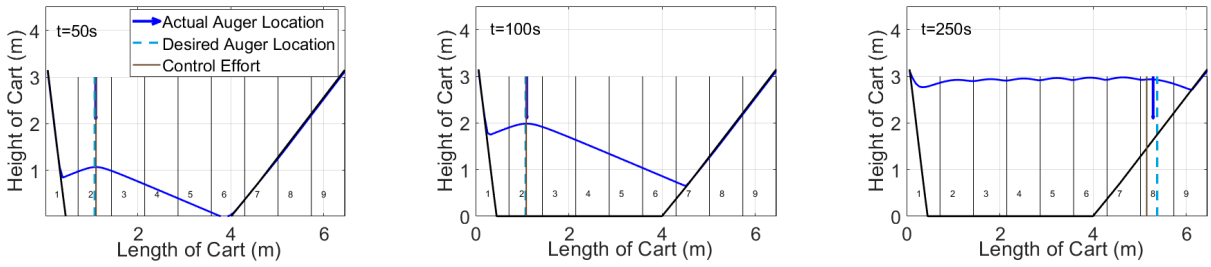


Figure 4.12. Relative grain cart position from automatic offloading simulation with closed-loop controller.



(a) (b) (c)

Figure 4.13. Grain profile from automatic offloading simulation with closed-loop controller (associated with Visualization 9). (a) t = 50 s; (b) t = 100 s; (c) t = 250 s.

Visualization 9 shows the automatic offloading simulation with closed-loop movement controls (per Fig. 4.2). Compared with open-loop control, as shown in Fig. 4.13, the commanded location (control effort) was not the same as the desired location because the movement controller automatically computes the command to Machine Sync based on the tracking error. Because of the closed-loop controller, the actual location of the auger better tracks the desired location, as shown in Fig. 4.12. At the final stage of unloading, when the desired location changes rapidly because of the high grain profile. The closed-loop system was still able to track the desired location but the open-loop system sometimes did not settle down before the next desired location update.

In summary, the automatic offloading simulation validates the hypothesis that the automatic offloading system can be used to automate unloading-on-the-go operation with a nominal fill target and reliable fill level response. Both open-loop and closed-loop strategies can be used to achieve the fill target by controlling the Machine Sync system, and the closed-loop movement controller can further improve the tracking performance.

5. HARDWARE INTEGRATION AND IMPLEMENTATION

After being designed and validated in the virtual environment, the automatic unloading system was implemented in a rapid prototyping system and validated with in-field testings. The implementation of the automatic unloading was done in two stages. Automatic unloading with human in the loop to validate system performance with an ideal perception system (i.e., human operator), and automatic unloading with a camera-based perception system in the loop (minimal operator feedback).

5.1 Automatic unloading with human in the loop

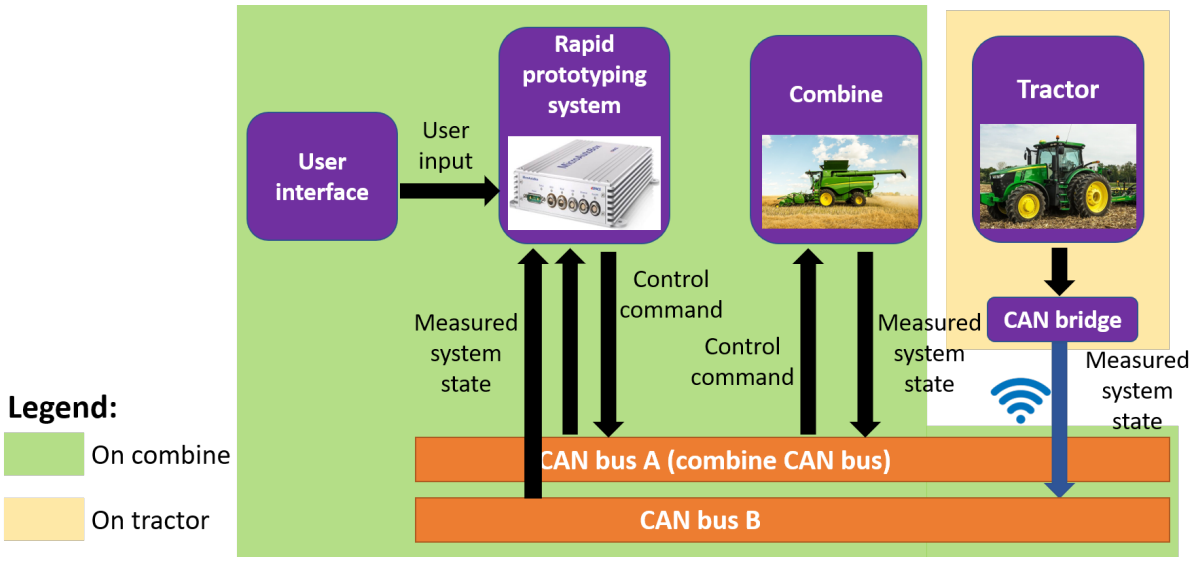


Figure 5.1. Hardware diagram for automatic unloading implementation with human in the loop

Figure 5.1 shows the hardware diagram for automatic unloading in-field implementation with human in the loop. The vehicles in the automatic unloading system consists of a combine harvester (model: John Deere S660) and a tractor (model: John Deere 8345R IVT) towing a grain cart (Brandt 1020 XR).

The automatic unloading controller is implemented in a rapid prototyping system (model: dSPACE MicroAutoBox II). The rapid prototyping system is located in the combine harvester. There are two CAN buses to transmit data among the system components. The

first CAN bus, the CAN bus A is the combine bus, connects the AU controller with the combine harvester. The controller sends control commands to the combine harvester including nudges to the Machine Sync system and auger on/off messages. In the meantime, the controller receives the combine running status from the combine CAN including the GPS location and auger status. CAN bus B is built between the rapid prototyping system and a wireless CAN bridge (model: Kvaser Air Bridge Light HS) that connects to the tractor CAN. The AU controller receives the tractor states including its GPS locations from this bus.

To allow the user to interact with the automatic unloading system, a switch box as the user interface was built for the user to start/stop the AU controller, command emergency stop, calibrate the system and indicate system status to the AU controller.

5.1.1 Controller adaption for hardware implementation

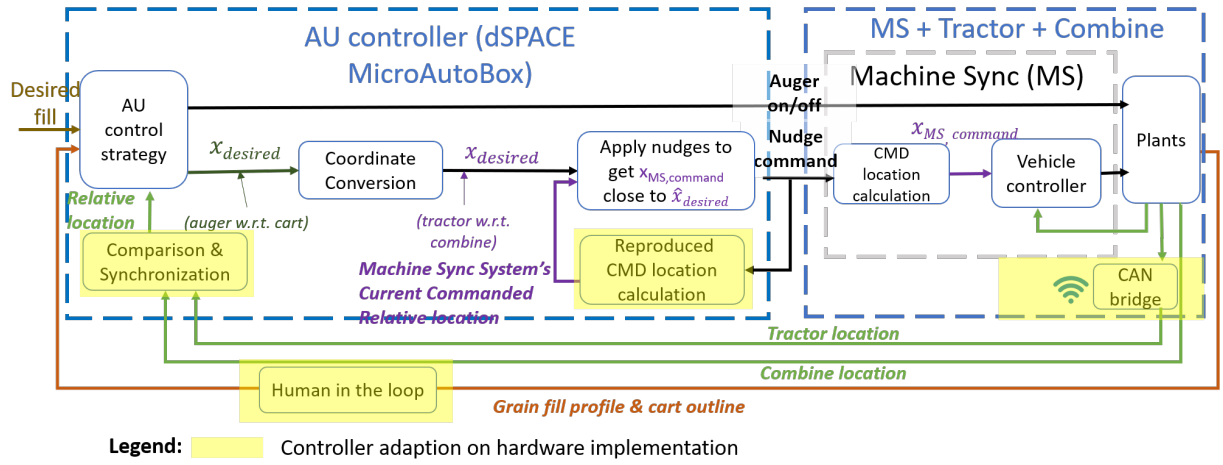


Figure 5.2. Controller block diagram for automatic unloading implementation.

The automatic unloading controller was implemented with open-loop movement controls and made adaption to the automatic unloading controller in simulated virtual environment in Fig. 4.1 due to the hardware constraints. Figure 5.2 shows the updated block diagram with adaptions for hardware implementation highlighted in yellow. In summary, three key updates were made from the system in simulation.

1. The fill level feedback to the controller is achieved by a human in the loop instead of a perception system in the loop.

2. The relative location between two vehicles cannot be directly accessed from the Machine Sync system with regular CAN communication, so the relative location was calculated from vehicles' GPS location in the AU controller and built wireless CAN bridge to allow the controller to access the CAN messages from both vehicles.

3. Because the commanded location in Machine Sync is not accessible in hardware directly, for the nudge handler to work properly, a commanded location estimator was built in the AU controller.

For update 1, to allow the operator to inform the controller of the cart filling status, a button was added to the user interface for the user to indicate the current section of the cart is full. After receiving the full message, the automatic unloading control updates the desired location and auger on/off status based on the fill strategy specified by the user, same as the AU controller validated in the virtual model.

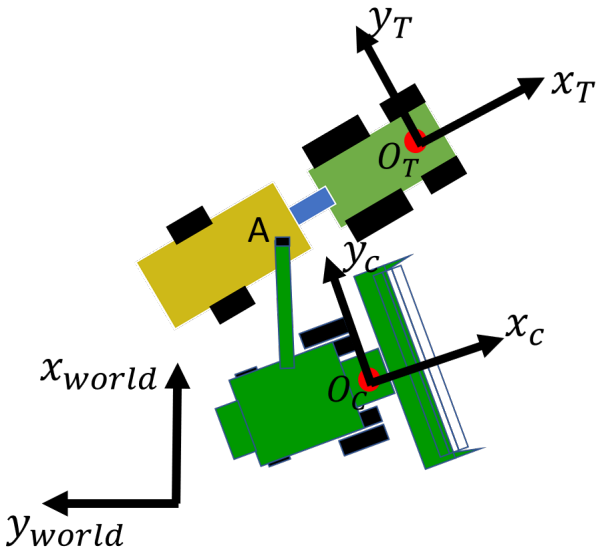


Figure 5.3. Schematic for auger location calculation from GPS information.

For update 2, an algorithm was developed to calculate the auger boot location relative to the grain cart from vehicles' GPS information provided in CAN bus. Figure 5.3 shows a

vehicle schematic with the 3 coordinates involved in auger location calculation. (x_{world}, y_{world}) is the world coordinate with x_{world} pointing north and y_{world} pointing west. (x_C, y_C) is the combine coordinate built on combine GPS globe O_C with x_C pointing to the combine moving direction. (x_T, y_T) is tractor coordinate built on tractor GPS globe O_T with x_C pointing to the tractor moving direction. A is the auger boot location.

Because the tractor GPS location and combine GPS location are obtained asynchronously, the AU controller needs to estimate their current GPS location based on their speed. A first-order estimation is implemented for the GPS coordinate for both vehicles. For example, the calculation for tractor latitude is

$$\widehat{\lambda_T} = \lambda_T + \frac{d\lambda_T}{dt}(t - t_T) \tag{5.1}$$

where $\widehat{\lambda_T}$ is the tractor latitude estimate at current time, λ_T is the tractor latitude from tractor CAN, t is the current time, and t_T is the time for last tractor coordinate update.

After that the relative distance between tractor and combine globe in world coordinate can be calculated by

$$\left(\overrightarrow{O_C O_T}\right)_{world} = \begin{bmatrix} \left(\widehat{\phi_T} - \widehat{\phi_C}\right)\cos\lambda_T R_{earth} \\ \left(\widehat{\lambda_T} - \widehat{\lambda_C}\right) R_{earth} \end{bmatrix} \tag{5.2}$$

where $(\widehat{\lambda_T}, \widehat{\phi_T})$ is the estimated tractor coordinate, $(\widehat{\lambda_C}, \widehat{\phi_C})$ is the estimated combine coordinate, and R_{earth} is the earth radius.

If the impact from grain cart articulation is negligible, the auger location in grain cart can be represented by the auger location in tractor coordinate $\left(\overrightarrow{O_T A}\right)_T$ and it can be computed by

$$\begin{aligned} \left(\overrightarrow{O_T A}\right)_T &= \left(\overrightarrow{O_T O_C}\right)_T + \left(\overrightarrow{O_C A}\right)_T \\ &= R_{world,T}\left(\overrightarrow{O_T O_C}\right)_{world} + R_{C,T}\left(\overrightarrow{O_C A}\right)_C \end{aligned} \tag{5.3}$$

where $\left(\overrightarrow{O_C A}\right)_C$ is the auger boot position relative to combine globe, a known geometry, $R_{world,T}$ is the transformation from world coordinate to tractor coordinate, and $R_{C,T}$ is the transformation from combine coordinate to tractor coordinate.

The transformation between coordinates in Eq. 5.3 can be calculated from the compass bearing of vehicles by

$$
\begin{aligned}
R_{world,T} &= \begin{bmatrix} \cos\theta_T & -\sin\theta_T \\ \sin\theta_T & \cos\theta_T \end{bmatrix} \\
R_{C,T} &= \begin{bmatrix} \cos\left(\theta_T - \theta_C\right) & -\sin\left(\theta_T - \theta_C\right) \\ \sin\left(\theta_T - \theta_C\right) & \cos\left(\theta_T - \theta_C\right) \end{bmatrix}
\end{aligned}
\tag{5.4}
$$

where θ_T is the compass bearing of tractor and θ_C is the compass bearing of combine.

For update 3, a counter in the AU controller was built to mimic the behavior of the commanded location calculation inside Machine Sync. The counter counts the number of nudge up and nudge down and estimate the change of commanded location from the initial commanded location (home point)

$$
\widehat{x_{CMD}} = x_{HP} + d_{nudge}\left(N_+ - N_-\right) \tag{5.5}
$$

where $\widehat{x_{CMD}}$ is the estimated commanded location, x_{HP} is the initial commanded location, d_{nudge} is the nudge size, and N_+, N_- are the nudge up and nudge down commands the controller sends to Machine Sync.

As shown in Eq. 5.5, the commanded location estimation requires the initial commanded location (home point) to start the counting, which requires a calibration. To calibrate the home point, before the first use of the automatic unloading system, the operator needs to engage Machine Sync and click a button on the user interface when the tractor reaches the home point. After that, the controller will record the current relative location as the initial commanded location.

5.1.2 In-field testing results

The automatic unloading system was tested by unloading from a combine harvester with a full hopper of corns to a half-full grain cart towed by a tractor to emulate the typical

Figure 5.4. Automatic unloading testing condition

unloading on the go process. The vehicles are running on flat terrain. Figure 5.4 shows a perspective view of the testing configurations.

At the beginning of the unloading, the combine operator kept a constant throttle to have the combine harvester cruise at a constant speed and turned on the John Deere AutoTrac control the steering to keep the vehicle in a straight line. After Auto Tractor engaged, both the combine harvester and the tractor started the Machine Sync function to control the relative position of the vehicles. After that, the combine operator will turn on automatic unloading to automatically controls the vehicle movement and the auger on/off.

In the first unloading test, the AU controller used the Front to Back to Front fill strategy Visualization 10 shows a video record of this automatic unloading test and Fig. 5.5 is a representative frame in this video. The left of the video shows a camera view from the combine cabin: the operator on the buddy seat clicks the "section full" button on the interface box to indicate the current unloading section is full. Based on the user input, the automatic unloading controller moves the auger to fill up the cart in two runs. In the first run, the auger fills up the cart from front to back. In the second run, the auger top off the cart from back to the front and stops after the entire cart is full. On the right side of the view is the view of the unloading process from a drone camera.

Figure 5.5. A representative frame in automatic unloading operation with Front to Back to Front fill strategy (associated with Visualization 10)

In the second unloading test, the AU controller used the Front to Back fill strategy. Visualization 11 shows a video record of this automatic unloading test and Fig. 5.6 is a representative frame in this video. In this test, the controller fills up the cart in one pass from front to back. The top left of the video shows a grain cart view from the combine, the bottom left of the video shows the grain profile visualization from a camera inside the grain cart, and the video on the right is the combine cabin view showing how the operator interacts with the automatic unloading system.

The implementation results demonstrated that the automatic unloading system help operator to reduce workload by automatically deciding the target auger location and auger status based on user-prescribed filling strategy. After that, to offload the operator from two-way communication for relative location control, the AU system can automatically coordinate the vehicle motion to achieve the desired fill profile.

Figure 5.6. A representative frame in automatic unloading operation with Front to Back to Front fill strategy (associated with Visualization 11)

5.2 Automatic unloading with perception system in the loop

Figure 5.7 shows the hardware configuration of the automatic offloading system with perception system in the loop. The automatic offloading controller is implemented in a Rapid Control Prototyping (RCP) system (model: dSPACE MicroAutoBox II). The controller is placed on the combine harvester. The automatic offloading controller has 3 interfaces running simultaneously:

1. Controller interfaces with the combine harvester via on-combine CAN bus. The CAN communication between the controller and the combine harvester not only provides the vehicle status to the controller for feedback control, but also allows the controller to automatically send positional commands to Machine Sync and auger on/off commands to the auger controller on the combine.

2. Controller communicates with the perception system (IPM2) via an Ethernet cable. The perception system receives the vehicle status from the Ethernet for its detection

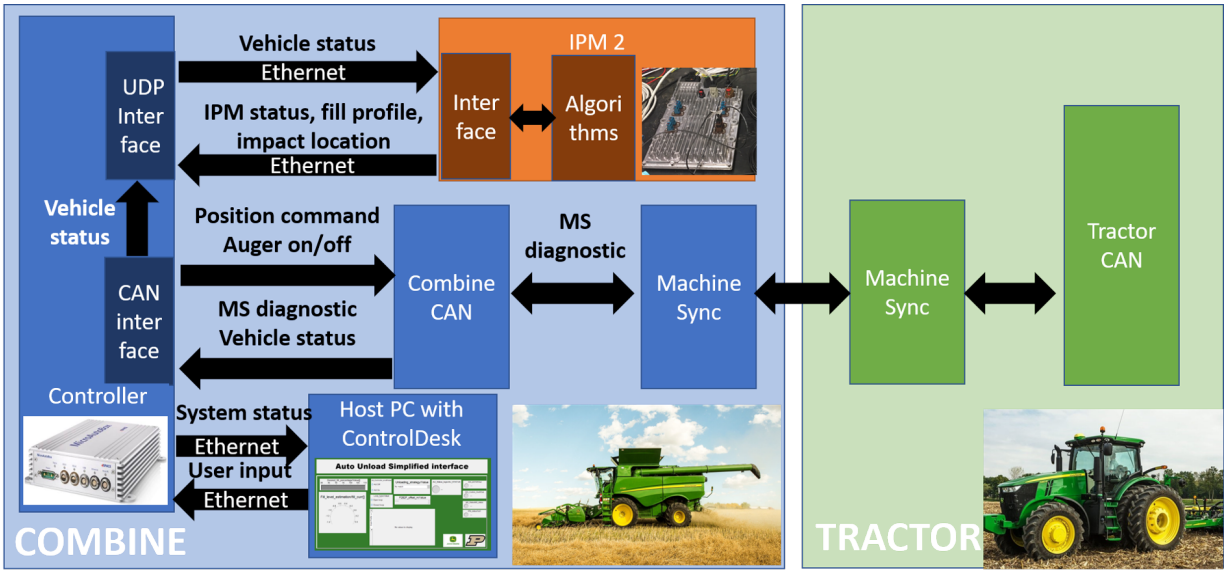


Figure 5.7. automatic offloading hardware diagram

algorithms. While simultaneously, the perception system transmits the cart filling status and perception algorithm status to the controller.

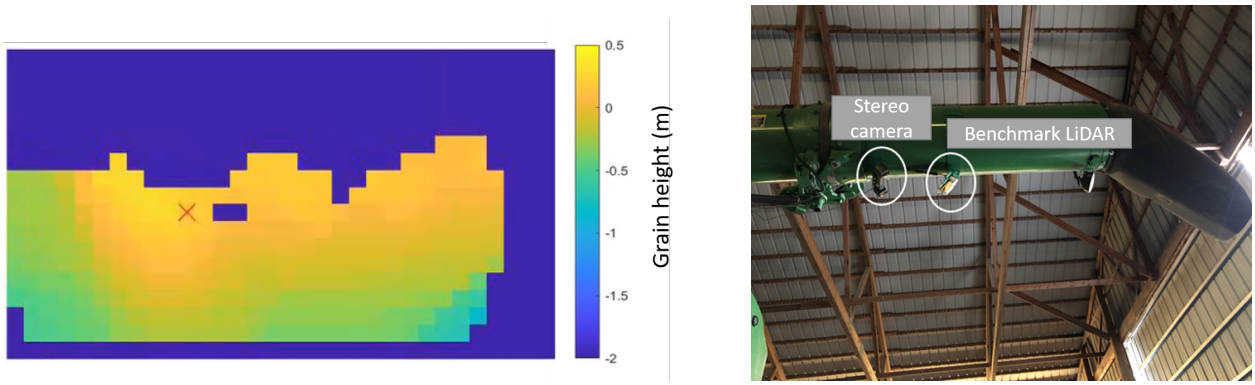
3. Controller also communicates with a user interface implemented on the host PC of the dSPACE MicroAutoBox via an Ethernet cable. The controller sends the current system status to the UI for visualization and receives the user-specified automatic offloading configurations from the UI.

5.2.1 Perception system integration

The automatic offloading system uses a stereo-camera-based perception system IPM (Image Processing Module) to monitor the fill status of the grain cart. A similar stereo-based system was used in the automatic unloading system for the Self Propelled Forage Harvesters (SPFH) [28], [74].

The IPM includes a stereo camera to capture images of the scene, and an industrial PC to process camera data and interface with the controller. As shown in Fig. 5.8b, the stereo camera is placed on the combine auger. The stereo camera has a field of view of 130 degree

with a baseline distance of 20 cm. The industrial PC is placed in the combine cabin. The perception data are transmitted via UDP/IP protocol to the controller at 5 Hz.



(a) (b)

Figure 5.8. Perception system (IPM) (a) An example data returned from perception system. The heat map shows the grain height relative to the edge, and the red cross shows the estimated grain impact location; (b) Placement of perception system and benchmark perception system.

Together with the stereo images captured by IPM, the IPM provides perception feedback for the automatic offloading system. Figure 5.8a shows an example data from the IPM, discretizing the grain cart via a 32×18 matrix. Each grid in the matrix represents the measured the grain height relative to the grain cart edge nearest the combine. The height value was provided at 5 cm increments. Additionally, the perception system also estimated the impact location of the grain. The grain impact location data was provided as a coordinate in the matrix.

The IPM results were quantitatively evaluated with a LiDAR-based benchmark perception system. Figure 5.8b shows the placement of the LiDAR relative to the stereo camera. The benchmark ran simultaneously with the IPM and generatds data with the same format as the IPM. [75] outlines the details of the benchmark system development and benchmark implementation.

According to the benchmark results, there are two major challenges for using the IPM as a perception system

1. Incomplete profile: Because the stereo camera is placed on the combine auger, its view can be occluded by the grain cart wall or grain flow. As a result, the grain profile is not complete due to occlusion. Figure. 5.8a shows an example of an incomplete profile.

2. The imaging quality of the stereo camera depends on the lighting conditions. Under certain circumstances (e.g., low-light, high dust), the IPM may loose track of the grain cart and stop updating the grain profile.

5.2.2 Perception system data augmentation

Perception data augmentation algorithms were developed, per below, to alleviate the above noted impacts.

Fill metrics adaptation

Since the complete profile is not available, using a simple average to estimate the fullness in each row, as shown in Eq. 4.1, may not provide consistent results and the results could be biased towards the available grids. Moreover, underestimation of fill level can lead to spillage.

Therefore, to prevent spillage and take advantage of the partial fill profile, an adapted fill metrics was proposed, specifically, an estimated grain height just adjacent to the grain cart edge. Ideally, the edge grain height directly comes from the perception system measurement. However, the data near the edge is not always available as such an area is often occluded by the grain cart wall. Furthermore, even if the data is available, using data from a single measurement (one grid in a matrix) is more susceptible to error. Therefore, the adapted fullness metric uses all of the grids in the same row for edge grain height calculation:

$$h_{edge,\mathrm{i}} = \frac{\sum_{\mathrm{j}=1}^{m} h_{\mathrm{i,j}} - \widehat{h}_{\mathrm{i,j},shape}}{m} \tag{5.6}$$

where $h_{edge,\mathrm{i}}$ is the estimated grain height near the edge at row i, $h_{\mathrm{i,j}}$ is the grain height at the j-th column of the i-th row, and $\widehat{h}_{\mathrm{i,j},shape}$ is the theoretical relative height at the j-th column.

Since the piling of granular material such as corn or beans follows the angle of repose [53], the theoretical height at the j-th column relative to cart edge can be defined as

$$\hat{h}_{\mathrm{i,j},shape} = \mathrm{j}_{impact} d \tan\theta - |\mathrm{j} - \mathrm{j}_{impact}| d \tan\theta \tag{5.7}$$

where j_{impact} is the impact location of the grain flow, and d is the grid size in matrix.

With the estimated grain height near the edge, a fullness indicator, similar to the one illustrated in Eq. 4.1 can be applied to determine the fullness from the edge height with corresponding desired height value.

Sensor down time fill level estimation

Loss of perception data is another common challenge due to lighting conditions or dust in agricultural application. To handle this circumstance, an message transmission of perception system status was established between the IPM and the automatic offloading controller. Once the status message shows the perception system loses signal, the automatic offloading controller will estimates the grain height change:

$$h_{edge,\mathrm{i}}(t) = h_{edge,\mathrm{i}}(t_0) + \hat{\dot{h}}_{edge,\mathrm{i}}(t - t_0) \tag{5.8}$$

Here the t_0 is the last time step when the perception system is working, $\hat{\dot{h}}_{edge,\mathrm{i}}$ is the estimated rate of grain height change. The rate is estimated by a time average of the height change rate at the current section when the perception system response is valid.

5.2.3 In-field testing results

The adapted automatic offloading system was tested in the field under nominal harvesting conditions and various unloading configurations. Figure 5.9 shows the cabin setup during unloading testing captured by a 360 camera. The automatic offloading system was validated with a John Deere S790 combine, and a John Deere 8R340 tractor towing a Brandt 1020 XR grain cart.

Figure 5.9. Hardware configuration during automatic offloading

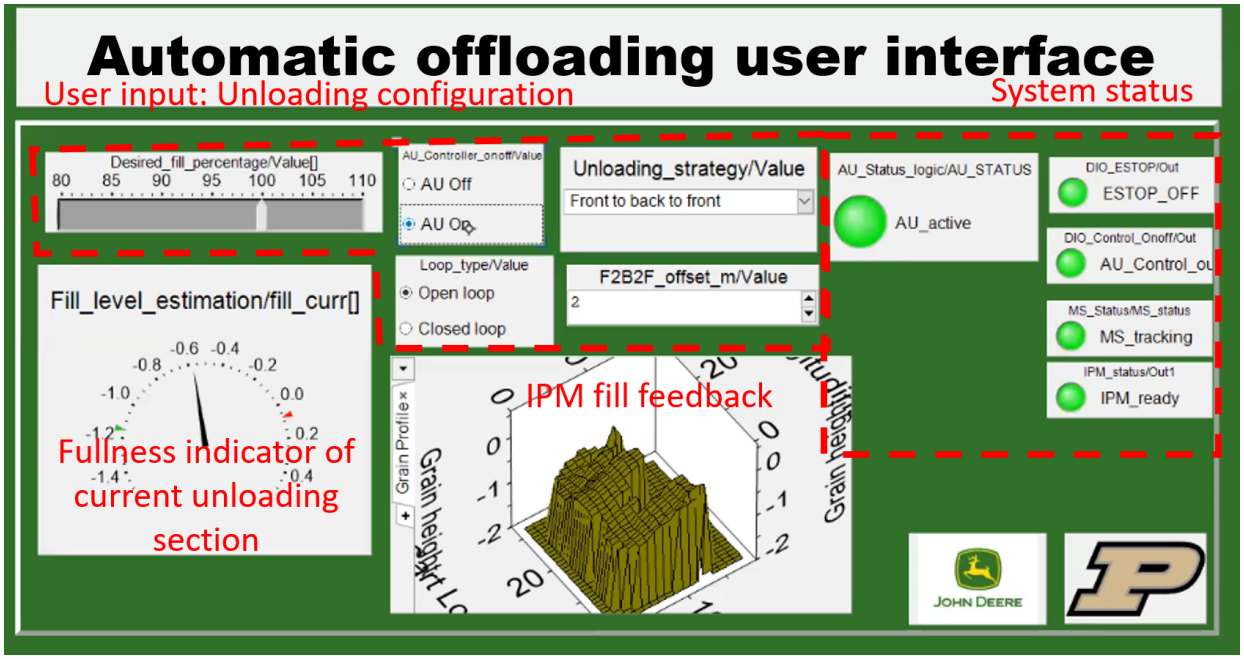


Figure 5.10. User interface for automatic offloading

Figure. 5.10 shows the user interface (UI) of the automatic offloading system for the operator to interact with. The user interface was used solely for testing, and is not the

intended approach for any commercial application. It allowed the operator to monitor the measured grain profile, the fill level in the current row, as well as the system status of different automatic offloading components. The operator was able to turn on/off the automatic offloading system, and configure the unloading parameters (e.g., unloading strategy, desired fill level) from the interface.

During the automatic offloading testing, the combine operator began harvesting and adjusted the hydro-handle to set the combine harvester at a constant cruising speed. AutoTrac system [76] was used to keep the combine harvester on a straight line. When the combine hopper was full and ready to unload, the tractor operator drove the tractor near the combine.

After Machine Sync was engaged, the combine operator started the automatic offloading system from the user interface. The automatic offloading system then automatically moved the grain cart to the first desired location based on the specified fill strategy. When the grain cart arrived at the desired location, the automatic offloading system indicated on the user interface that the automatic offloading system was ready. The combine operator used the combine handle to manually enable the auger. The automatic offloading system automatically filled the grain cart by controlling the relative position of the vehicles and stopped the auger when the desired fill level was achieved.

The automatic offloading system was tested at a nominal combine harvester speed of four mph, executed on a flat terrain with no end-of-row turning. The testing covered different fill strategies, initial fill profiles, and desired fill level to validate the performance of the system under different conditions.

Figures 5.11-5.13 show automatic offloading testing scenario A in which the unloading started from a half-full grain cart, front to back strategy, and an open-loop movement controller. The desired fill level was $h_{edge,\mathrm{i}} = -0.3m$ as based on the metrics defined in Eq. 5.6. The grain cart was divided into 0.6-meter rows.

The left axis in Fig. 5.11 shows the relative movement in the longitudinal direction (vehicle moving direction) between the two vehicles represented by the relative distance between the combine GPS globe and the tractor GPS globe. The right axis in Fig. 5.11 shows the fill metrics indicator at the current row that the auger was operating in. Visualization 13 shows the unloading process recorded by a camera in the combine cabin.

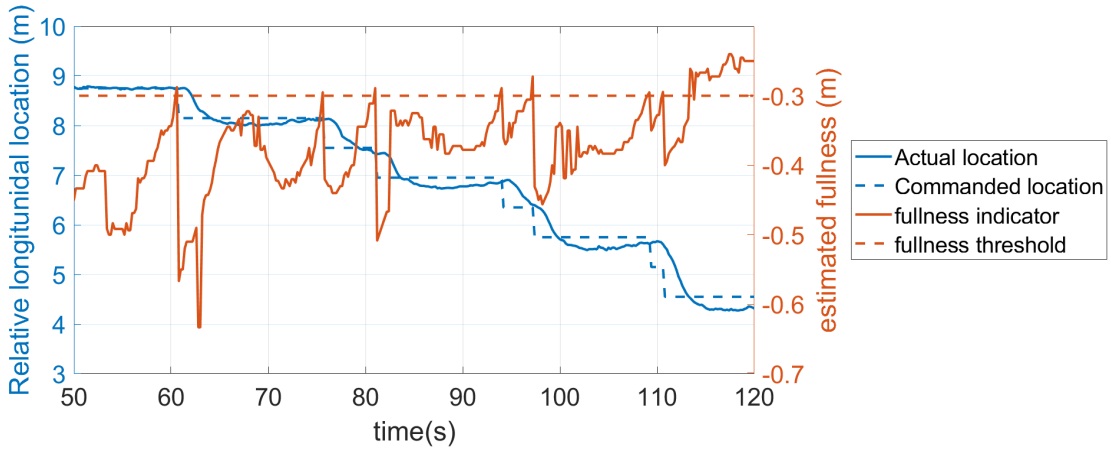
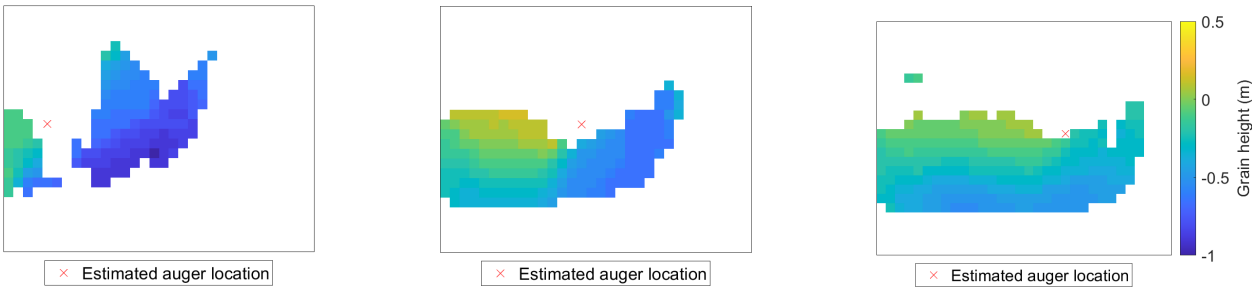


Figure 5.11. Relative location between vehicles and fill level metrics during automatic offloading testing scenario A. Fill strategy: Back to Front. Desired fill level: $h_{edge,\mathrm{i}} = -0.3m$. Initial profile: half-full. Combine speed: four mph. Open-loop movement control.

Figure 5.12. Unload scenario of automatic offloading testing scenario A. Associated with in-cabin video recording in Visualization 12

At the start of the test in Visualization 13, the auger boot is at the back of the grain cart. After it stabilized at the first unloading row, the automatic offloading controller informed the operator to turn on the auger via the user interface. At around 55s of the experimental log, the grain started to pile up on the grain cart, so the fullness metrics started to increase. Once the fullness at the current section had reached the desired fill level set by the operator (at around 60 s), the automatic offloading controller commanded a nudge and moved the auger



(a) (b) (c)

Figure 5.13. Grain profile change during automatic offloading testing scenario A, associated with Visualization 13 (a) t = 55.8s; (b) t = 84.0s; (c) t = 105.2s.

to the next section. The controller followed a similar pattern to fill up the grain cart from the back to the front. At around 114 s, the last section was filled so the automatic offloading controller did not continue to nudge the tractor, but turned off the auger automatically instead. The final fill level in the last section (at t=115s in Fig. 5.11) was slightly higher than the specified threshold because of the delay in turning on the auger.

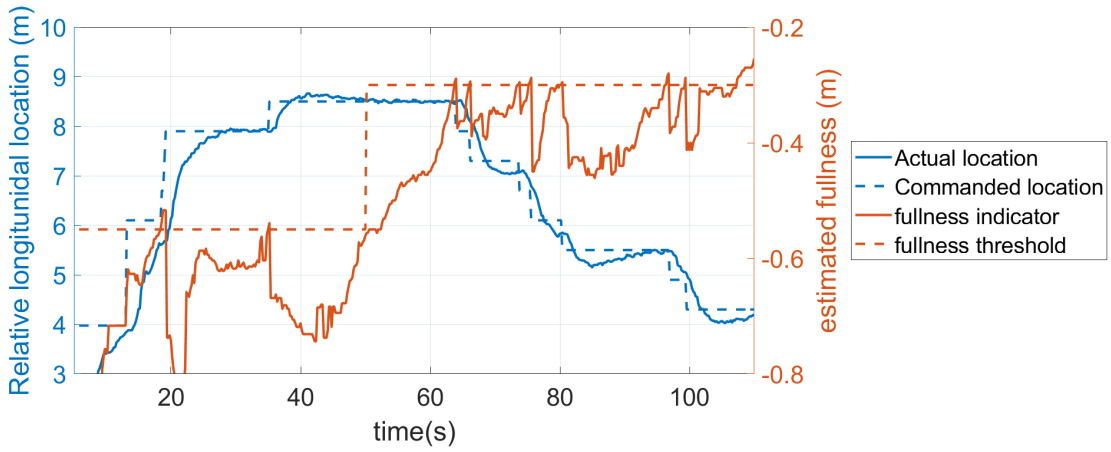
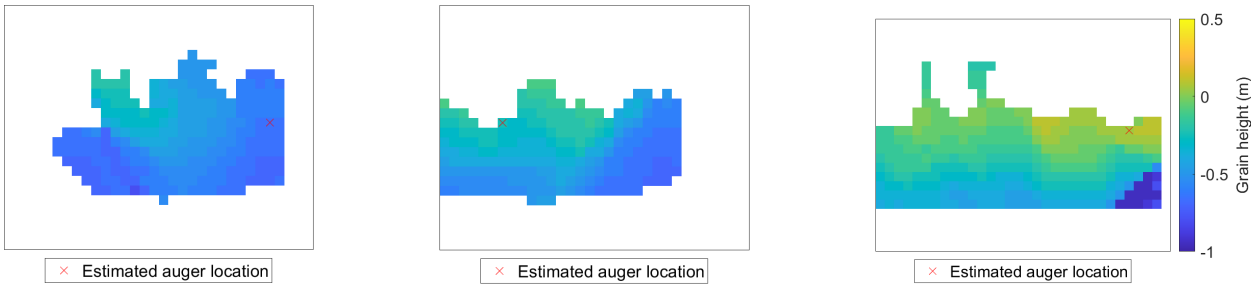


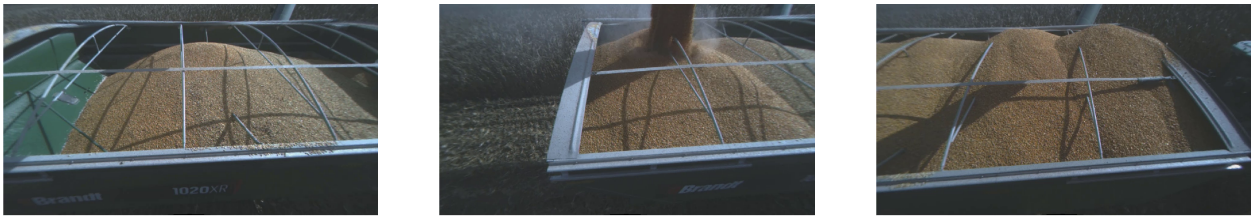
Figure 5.14. Relative location between vehicles and fill level metrics during automatic offloading testing scenario B. Fill strategy: Middle to Back to Front. Desired fill level: $h_{edge,\mathrm{i}} = -0.3m$. Initial profile: half-full. Combine speed: 4 mph. Open-loop movement control.

Figures 5.14-5.15 illustrates automatic offloading test scenario B, which has the same configuration as test scenario A except for the fill strategy. In test scenario B, Middle to



(a) (b) (c)

Figure 5.15. Grain profile change during automatic offloading testing scenario B, associated with Visualization 14 (a) t = 15.2s; (b) t = 51.6s; (c) t = 104.8s.



(a) (b) (c)

Figure 5.16. Rectified camera image from the left lens of stereo camera during test scenario B, associated with Visualization 15 (a) t = 15s; (b) t = 50s; (c) t = 105s.

Back To Front strategy was employed. As shown in Fig. 5.14, the unloading was finished in two subsequent stages. In the first stage (middle to back), the fullness threshold was at a lower value ($h_{edge,\mathrm{i}} = -0.55m$) than the desired fill level, so the system could create an even profile that was close to full. In the second stage, the fullness threshold is based on the value set by the user, so the automatic offloading system could top off the cart to the desired fill level. Figure 5.15 (associated with Visualization 14) shows the profile change during the test scenario B. In particular, Fig. 5.15b shows the profile when the first stage was finished. After that, the automatic offloading controller automatically raised the fullness threshold to the user-defined value ($h_{edge,\mathrm{i}} = -0.3m$) and started topping off the grain cart from back to front. Figure 5.15c shows the profile when the unloading is finished, which is a overall higher profile than the one shown in Fig. 5.15b. To visualize the same process, Visualization

15 shows the rectified image from the left lens of the stereo camera, and Fig. 5.16 shows the three typical frames from the footage.

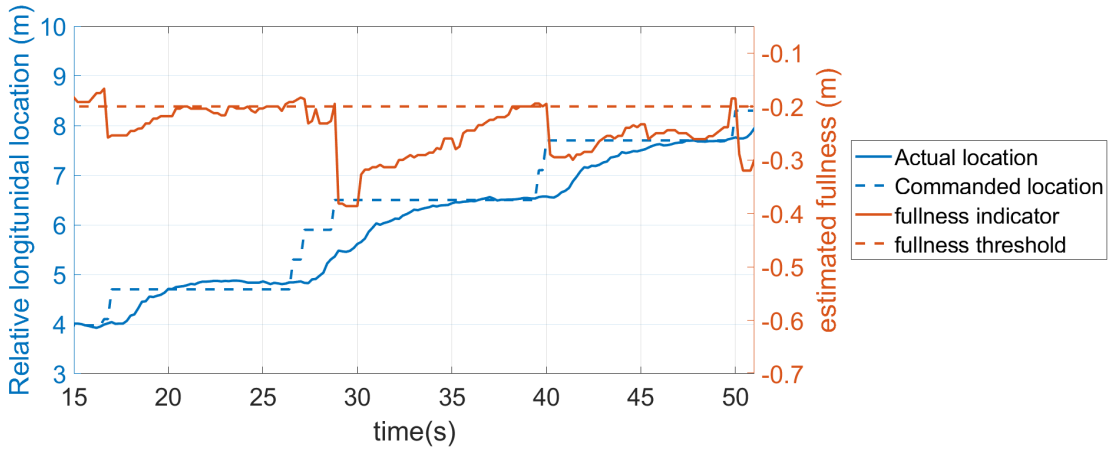
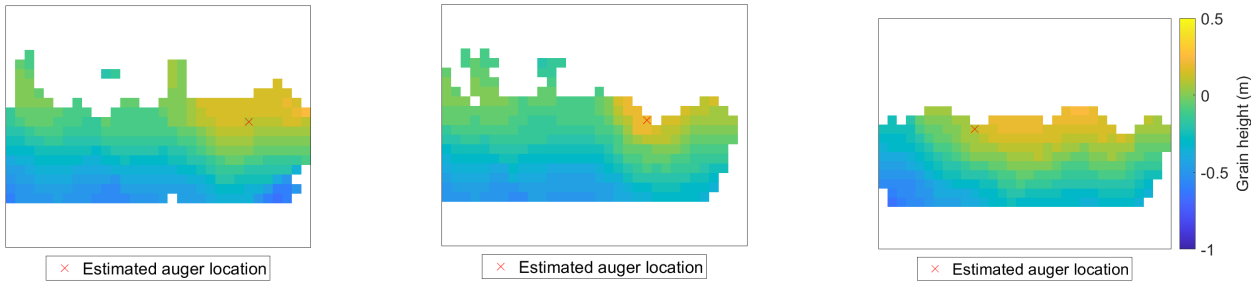


Figure 5.17. Relative location between vehicles and fill level metrics during automatic offloading testing scenario C. Fill strategy: Front to Back. Desired fill level: $h_{edge,\mathrm{i}} = -0.2m$. Initial profile: almost full. Combine speed: 4 mph. Open-loop movement control.



(a) (b) (c)

Figure 5.18. Grain profile change during automatic offloading testing scenario C, associated with Visualization 16 (a) t = 20.0s; (b) t = 30.0s; (c) t = 49.0s.

Figures 5.17-5.18 demonstrates test scenario C with Front to Back fill strategy. Compared with test A and test B, the desired fill level is higher ($h_{edge,\mathrm{i}} = -0.2m$), and the initial profile, as shown in Fig. 5.18a is also higher. Consequently, at the start of the test, the auger did not turn on when it was at the front part of the cart. Automatic offloading system automatically moved the auger to the middle of the cart before indicating on the UI to turn on the auger.

At t = 30s, when the commanded location was at the middle of the cart, the auger was turned on by the operator. After about 15 seconds, automatic offloading system finished filling up the middle rows of the cart, and the unloading stopped because the combine hopper became empty. Visualization 16 shows the profile change in test scenario C.

The in-field testing demonstrates that the automatic offloading system works under nominal harvesting condition to automatically top off a half-full grain cart.

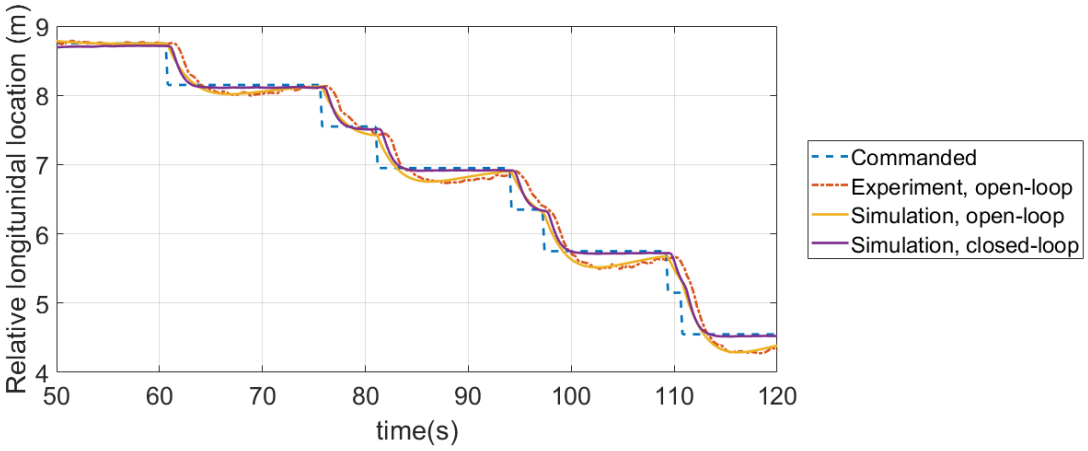


Figure 5.19. Comparison between simulated vehicle location and experimental vehicle location in automatic offloading test A.

The experimental results were also used to validate the fidelity of the simulation model used to design the system. Figure 5.19 shows a comparison between the simulated vehicle location and experimental vehicle location. The experiment was conducted with the open-loop movement control method as shown in Fig. 4.1. Its location response was consistent with the simulated open-loop system with the same commanded location, as shown in Fig. 5.19.

It can also be observed from both the simulation and the experiment that the open-loop system may not be able to track the reference as well when the commanded location changes frequently (e.g., t=95s in test A). In comparison, the closed-loop position controller, per the simulation results in Fig. 5.19, demonstrates a better tracking performance thanks to its shorter settling time and smaller overshoot. Therefore, the closed-loop movement control can instead be used when warranted.

5.3 Conclusion

The design and validation of an automatic grain unloading system for combine harvester unloading-on-the-go was described in this paper. To simulate the automatic offloading process, system models for grain fill dynamics and vehicle dynamics were developed and verified with in-field testing. The automatic offloading controller provided three different fill strategies and two movement control options, both of which were simulated to validate the unloading performance. A stereo-camera-based perception system was integrated in the automatic offloading system to provide feedback to the controller. Perception data augmentation algorithms were proposed to accommodate the characteristics of the perception system. The automatic offloading system was implemented on a combine harvester and a tractor driven grain cart. The system was tested with different scenarios to demonstrate successful automatic offloading under nominal harvesting conditions with different configurations.

6. GRAIN PROFILE DATA FUSION OF CAMERA-BASED PERCEPTION SYSTEM AND GRAIN FILL MODEL

6.1 Introduction

The in-field testing demonstrated that, given a properly working perception system, the automatic unloading system can automate the unloading on the go process and offload the combine operator from grain cart monitoring, auger location control, and vehicle coordination.

However, a camera-based perception system can frequently lose track of the grain cart because of poor lighting conditions, different viewing perspectives, or dust occlusions. Moreover, even when perception algorithms are working properly, depending on stereo camera placement and field of view, certain parts of the grain profile cannot be seen. The tracking lost and incomplete fill profile can degrade automatic unloading system performance, or even worse, force the automatic unloading to stop.

Figure 6.1. An example scenario with high dust impacting the performance of perception system. Associated with Visualization 17

As shown in Fig. 6.1, Visualization 17 demonstrates an example scenario where poor lighting condition and heavy dust result in intermittent perception system feedback. This testing was conducted at dusk, so the environmental lighting was relatively dark, resulting

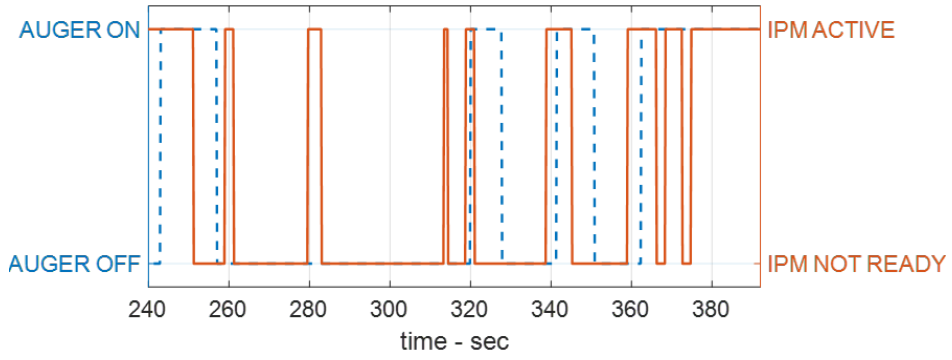


Figure 6.2. Perception system and auger status in poor lighting and heavy dust

in degraded imaging quality. Additionally, the dry corns also cause higher dust level from the grain flow.

Figure 6.2 shows the system status in this process due to the intermittent perception system signal. For example, at $t = 243s$, $t = 319s$, and $t = 340s$, after auger turns on, the excessive dust causes the perception system to lose track of the grain cart, and the perception system state switches from "active" state to "not ready" state. As introduced in Sec. 5.2.2, the auger automatically turned off after the perception system (IPM) loses track for 5 seconds. Therefore, the automatic unloading system turned off the auger multiple times during this unloading process, making the system unusable in such a challenging scenario.

On the other hand, Sec. 3.1 shows a physical model to estimate the grain profile change during unloading operation, which is also the most critical feedback from the perception system. Experimental validations demonstrate that the grain fill model can achieve 0.2-m accuracy given reliable model inputs. Unlike the perception system, the calculation of the grain fill model is not impacted by environmental conditions and can always return a complete grain profile. However, the grain fill model requires an initial profile inside the grain cart to start the simulation and due to the time-evolving characteristics of the grain fill model, the modeling error will accumulate if the model input has bias. Consequently, the feedback from a perception system is a great calibration means for the grain fill model to get the initial fill profile and reduce accumulation error.

Various algorithms have been developed to fuse the data from sensors and models. The Kalman filter (KF) [77], [78] is one of the most popular data fusion algorithms in the field of data processing. It has been proven that the KF is the optimal estimator for a one-dimensional linear system with Gaussian error [79]. However, the Kalman filter is not designed for the nonlinear system, such as the grain pile process in the unloading application. To extend its capacity on nonlinear systems, its variants, extended Kalman filter (EKF) [80] and unscented Kalman filter (UKF) [81] were introduced and applied in different applications.

Nonetheless, the computational cost is very high for Kalman filter approaches because the state vector in grain profile estimation has at least hundreds of states. For example, the grain profile returned by the perception system has a dimension of 32×18 (i.e., 576 states for grain height). In the Kalman filter approach, an error covariance matrix for the model state needs to be stored and propagated in time. With such a large number of states, it is computationally infeasible to perform the computation in real time [82]. For large state vectors, ensemble Kalman filters (EnKF) were proposed to approach the state estimation method as a sequential Monte Carlo simulation to handle large state spaces and nonlinear error evolution [83]. [84] shows a list of ensemble Kalman filter applications ranging from highly dynamical models to high-dimensional systems with $\mathcal{O}(10^6)$ number of states. However, EnKF still requires the state error to be Gaussian distributed, making it not suitable for the grain profile estimation, whose error distribution is not Gaussian especially around the boundary of the flowing region.

To achieve data fusion with a relatively low computational cost, a novel fusion algorithm was developed. Instead of computing the error covariance for the entire state space (i.e., the height of each grid), which is computationally expensive, the fusion algorithm only uses the perception system measurement to provide initial state estimation and correct for a limited number of critical parameters during unloading.

By applying the fusion algorithm on experimental data, it is demonstrated that the fusion algorithm can make the fill level feedback to the automatic unloading controller more robust to environmental change, less susceptible to accumulation error, and provide more complete coverage. It has also been shown that the fusion algorithm can run in real-time.

6.2 Principles

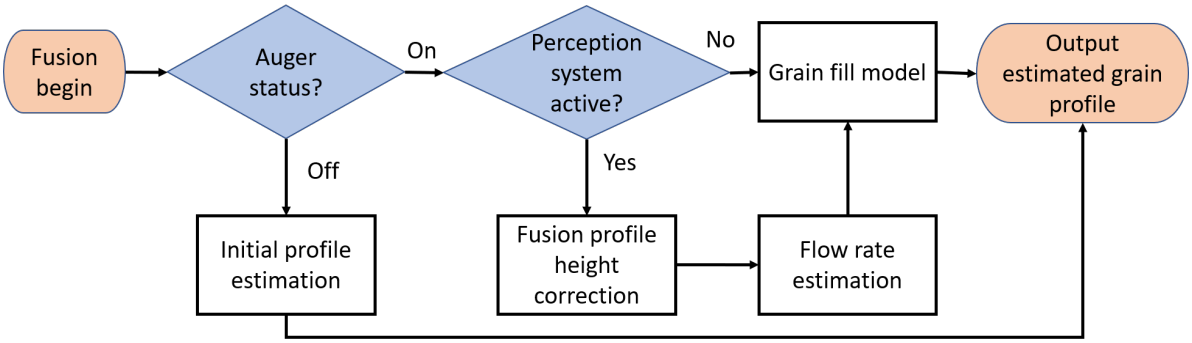
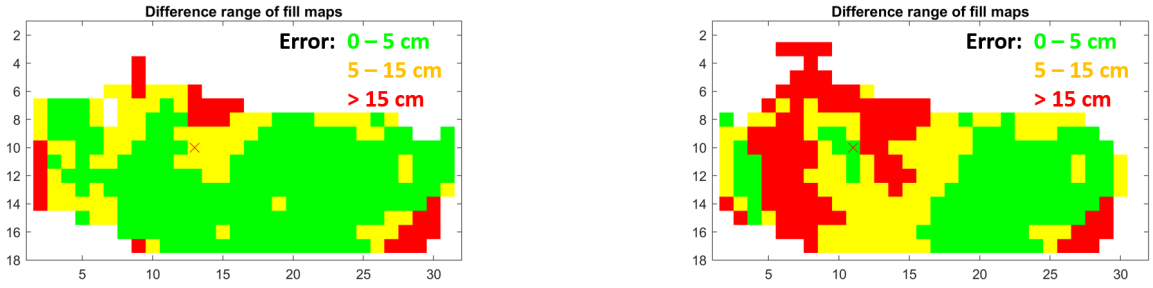


Figure 6.3. The computational pipeline for grain fill profile estimation by fusing the perception system and grain fill model

Figure 6.3 shows the computational pipeline for the fusion algorithm. The estimation pipeline is based upon the grain fill model, and the fusion process consists of three major components: initial profile estimation, fusion profile height correction, and flow rate estimation.



(a) (b)

Figure 6.4. Perception system (IPM) error distribution with similar height. (a) auger off; (b) auger on.

At the first stage, the algorithm will select an estimation strategy based on the auger status. By comparing with a LiDAR-based benchmark perception system, it has been observed from the in-field testing that the error of grain profile measurement surges dramatically after

auger turns on [75]. Figure 6.4 shows an example of the perception system error distribution at a similar grain height before and after the auger turns on. On one hand, after the auger turns on, because of the increased level of dust and occlusion due to the grain flow, with a similar grain profile, the percentage of the grid with height error greater than 15 cm increases dramatically, making it unreliable to estimate the grain profile from only the perception system measurement. On the other hand, before the auger starts, the majority of the visible grids have a measurement error of less than 5 cm, making it a reliable source for grain profile estimation

Therefore, when the auger is off, the estimation uses the perception system feedback directly for the available grids because of the high data reliability. In the meantime, to obtain the grain profile for the unavailable grids, Sec. 6.2.1 introduces the algorithm to estimate the grain height of the entire grain cart.

After the auger turns on, the grain fill model will be used as the major means for grain profile estimation. As shown in Fig. 6.3, when the perception system is not active, grain profile estimation can only be obtained from the grain fill model, so the estimation will directly come from the grain fill model estimation with initial conditions and parameters estimated when the perception system is active.

On the contrary, when the perception system is active, the measurement data is used to improve the grain fill model estimation accuracy in two approaches. Section 6.2.2 outlines the algorithm to use grain profile measurement from the perception system to adjust the overall height of the estimated grain profile and thus reduce the accumulated error of the grain fill model. Section 6.2.3 introduces the methods to estimate the actual flow rate during the unloading.

Grain pile characteristics related to grain volumes such as the overall height and the flow rate were extracted from perception system measurement. That is because the perception system can maintain satisfactory accuracy on the overall grain height (which shows the grain volume) even when the auger is on. Figure 6.5 shows a typical example of the grain height error bias during an unloading operation, in which the grain height error bias is less than 0.07 m even when the auger is on. The frequency of the grain height error bias based on multiple

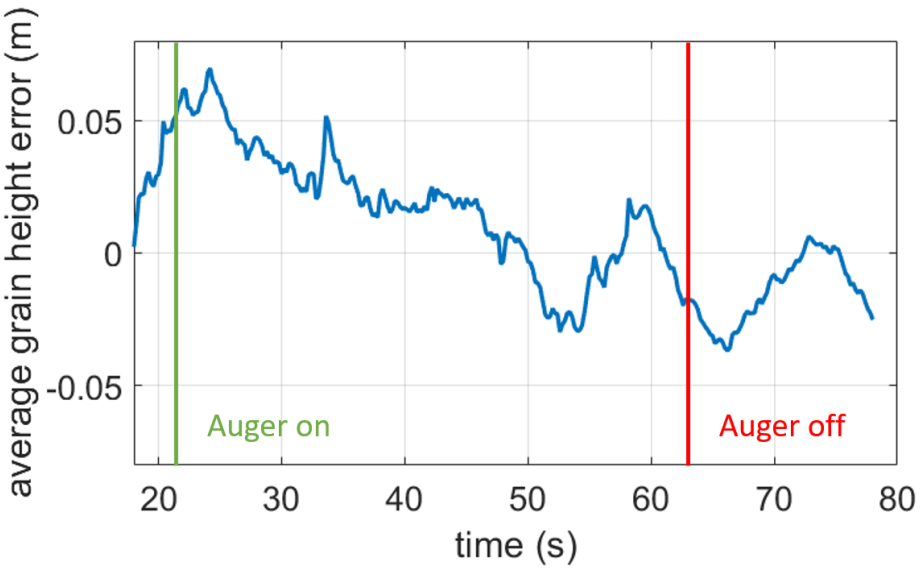


Figure 6.5. Grain height error bias of the perception system in an unloading operation

unloading operations as shown in Fig. 6.6 also confirms this characteristics observation. In more than 97% of unloading time, the bias of grain height error is within ±0.7 meters.

Notice here the bias of grain height error is much smaller than the error of most individual grids shown in Fig. 6.4b. This is because the bias of the overall grain height is calculated from the average of multiple grids. According to the law of large numbers, the average result from more trails is going to be closer to the expected value, and has a lower variance. Therefore, the variance from the average of multiple grids is lower than the variance of individual grids, making the average grain height reliable for volume-related estimation.

Table 6.1. The estimation source of the parameters in the grain fill model

Model parameters	**Estimation source**
Auger flow rate	Auger mode; perception system feedback
Angle of repose	Prior knowledge about the crop (can also be estimated from grain moisture and grain type)
Impact location	Inferred from real-time GNSS vehicle location and prior knowledge on grain flow velocity
Initial profile	Geometry: from perception system measurement before auger turns on Height: real-time perception system measurement

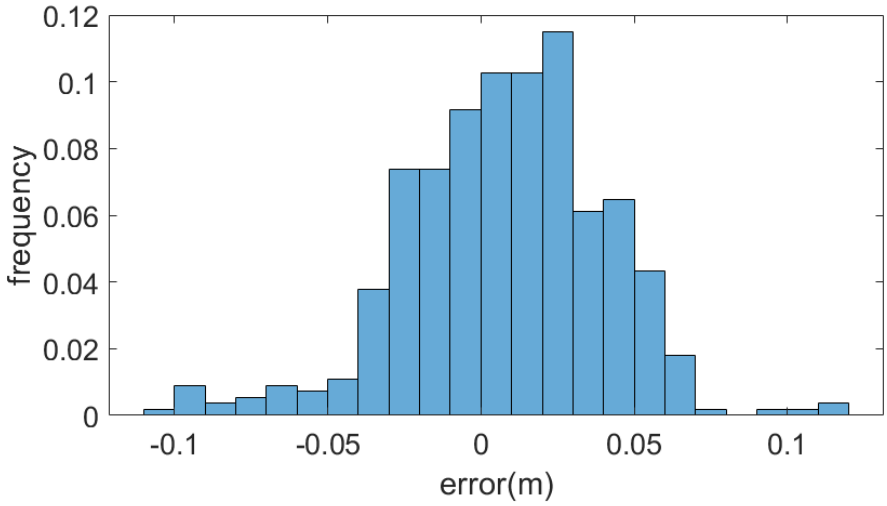


Figure 6.6. The frequency of the grain height error bias of the perception system in multiple unloading testing

However, the fusion pipeline does not estimate other geometry-related parameters from the measured grain profile such as the angle of repose and the grain impact location. That is because the grain profile measured from the perception system after the auger turns on does not have an accurate geometry representation of the actual profile. Figure 6.7 shows an example of the difference between the perception system feedback and the benchmark data. By comparing the height at several cross-sections in Fig. 6.7c, it can be observed that the perception system data not only has height deviation from the benchmark, it also reflects wrong geometric characteristics of the piles (e.g., angle of repose, peak locations). Because these geometric characteristics can be more accurately extracted from vehicle location from GNSS and grain properties [85], the algorithm does not extract geometric parameters from perception system feedback but uses prior knowledge and other sensors for calculation. Table 6.1 summarizes the estimation source for the parameters required by the grain fill model.

6.2.1 Initial profile estimation

Because of the occlusion and field of view limitation of any vision-based sensor, even when the auger is off, the measured grain profile from the perception system is going to be incomplete. However, the grain fill model requires a complete grain profile to estimate the

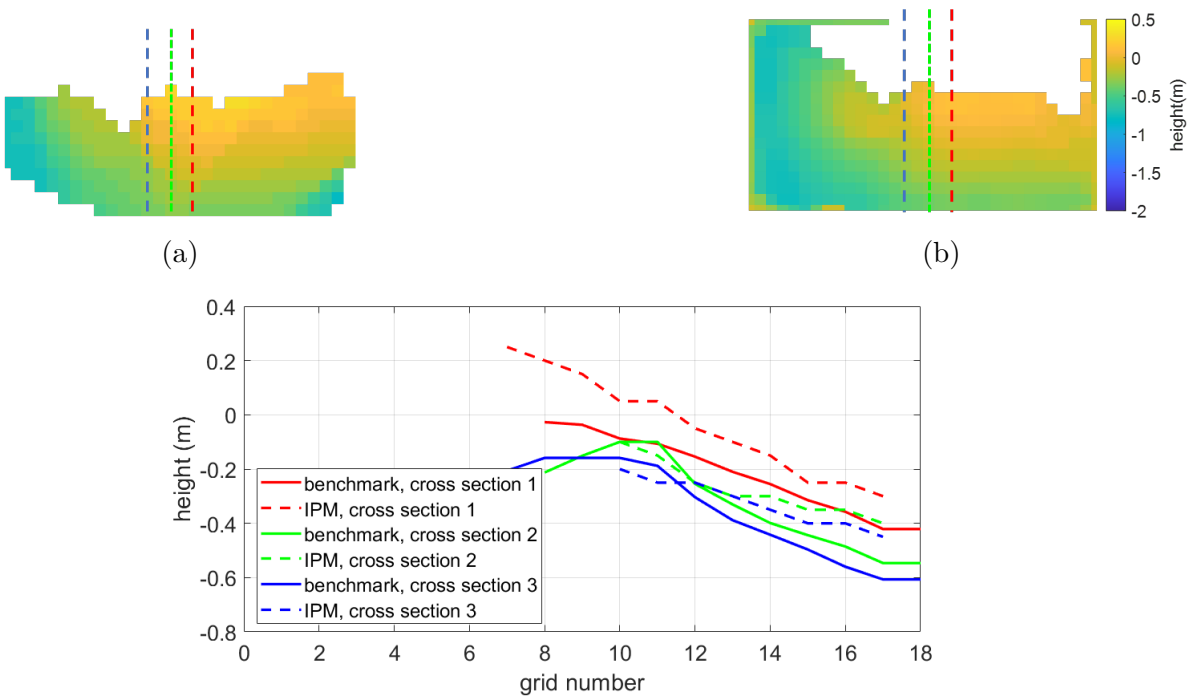


Figure 6.7. Difference between the perception system feedback and benchmark profile when auger is on. (a) perception system feedback; (b) benchmark profile; (c) three cross section views highlighted in color.

profile change. Therefore, an estimation algorithm is developed to estimate the grids that are not visible to the perception system.

Figure 6.8 shows the algorithm for the initial profile estimation. The estimation is based on the grain profile measured by the perception system when the auger is off because of the higher measurement reliability. For the first step, the algorithm looks for the peaks of the profile. Peaks are defined as the grids that satisfy the following conditions:

- The neighborhood grids are not higher than such grid
- Not a grid at the edge of the profile
- The majority of its 8 neighboring grids are visible

Besides identifying the peaks, the algorithm also estimates the piling angle of the current profile. The piling angle is estimated by doing a linear fitting for the lateral cross-sections

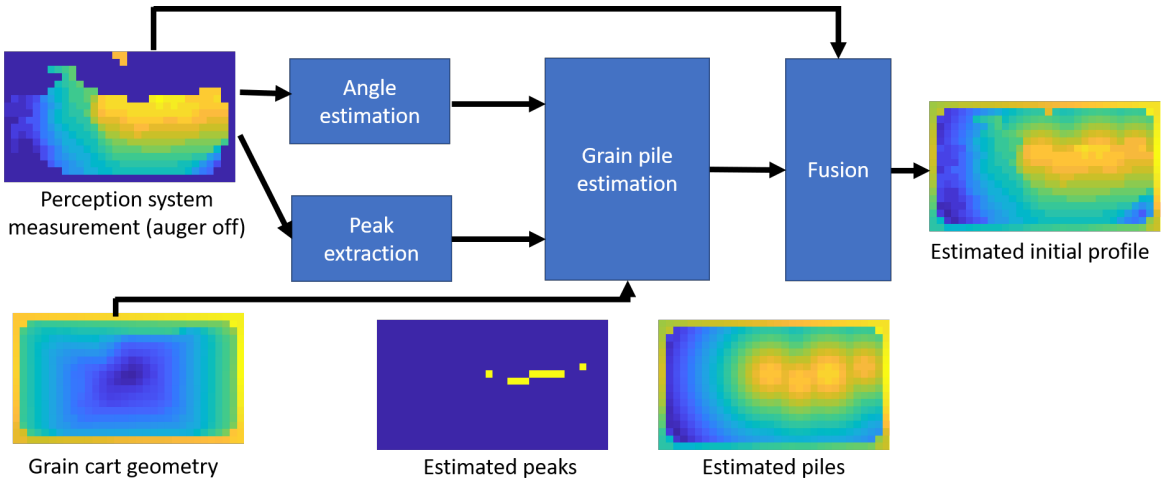


Figure 6.8. Computational pipeline for initial profile estimation.

of the profile near the peaks. The linear fit only includes the data between the highest and lowest point for each cross-section to exclude the height data from the grain cart or from the further half of the grain profile. After obtaining the slopes k_{i} for all the cross-sections, the piling angle can be calculated from the maximum angle

$$\alpha_p = \arctan\left(\max k_{\mathrm{i}}\right) \tag{6.1}$$

Notice the piling angle α_p here is not necessarily the same as the angle of repose α in Eq. 3.1 because the jostling of cart between unloading can cause the piling angle to lower.

After obtaining the piling angle $alpha_p$ and the peaks location $\{x_{p,\mathrm{i}}, y_{p,\mathrm{i}}\}$, the estimated combination of all the piles can be expressed as

$$h_p\left(x, y\right) = \max\left(h_{\mathrm{i}}\left(x, y\right), h_{cart}\left(x, y\right)\right) \tag{6.2}$$

where h_{cart} is the grain cart geometry to ensure the profile is higher than the cart, and h_{i} is the grain pile correspond to the i-th peak

$$h_{\mathrm{i}}\left(x, y\right) = h_{perception}(x_{\mathrm{i}}, y_{\mathrm{i}}) - \frac{\sqrt{\left(x - x_{\mathrm{i}}\right)^2 + \left(y - y_{\mathrm{i}}\right)^2}}{\tan \alpha_p} \tag{6.3}$$

here $h_{perception}$ is the measured grain profile.

Finally, because the measured grain profile when auger off is reliable, the initial profile will only uses the estimated pile profile h_p when the measured data is not available.

$$h_{\text{ini}}(x,y) = \begin{cases} h_{perception}(x,y), \text{grid} & \text{available} \\ h_p(x,y), \text{otherwise} \end{cases} \tag{6.4}$$

With the complete initial profile, now the grain fill model can estimate the grain profile change over time.

6.2.2 Fusion profile height correction

One of the major disadvantages of using the grain fill model alone to predict the grain profile change is the accumulation error. Because the grain fill model is an time-evolving model, a small bias in the model parameter can be accumulated over time and results in a large estimation error. Meanwhile, the perception system measurement can maintain satisfactory accuracy during unloading.

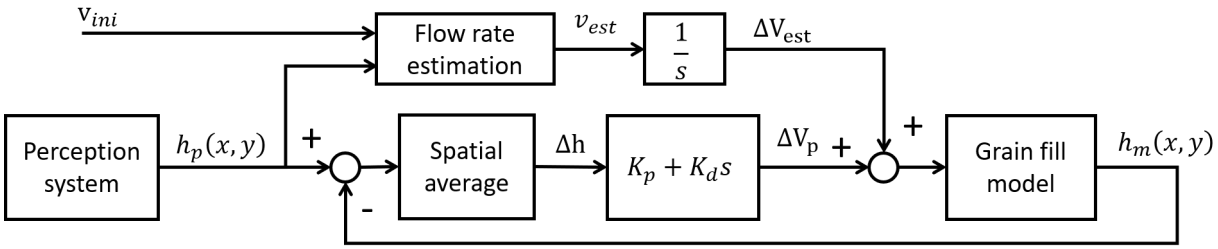


Figure 6.9. Closed-loop diagram for correcting the overall height of estimated profile

Therefore, to alleviate the accumulation error, the overall height of the estimation results was adjusted based on the perception system measurement. Figure 6.9 shows the closed-loop diagram on adjusting the overall height of the estimated grain profile. The profile height correction is done by adding/subtracting grain from the grain fill model. Besides the feed-forward flow rate estimation v_{est} from Sec. 6.2.3 and its corresponding volume change ΔV_{est}, the additional volume ΔV_p was also fed to the grain fill model to adjust the overall height.

The compensation flow v_p is calculated from the average height difference between grain fill model estimation and perception system measurement Δh.

$$\Delta V_p = K_P \Delta h + K_d \frac{d\Delta h}{dt} \tag{6.5}$$

where K_P is the proportional gain, K_d is the derivative gain, and Δh is the average height difference. Δh is calculated from the visible grids from the perception system measurement.

$$\Delta h = \frac{\sum_{\mathrm{i}=1}^{N_{visible}} h_p(x_\mathrm{i}, y_\mathrm{i}) - h_m(x_\mathrm{i}, y_\mathrm{i})}{N_{visible}} \tag{6.6}$$

where $h_p(x, y)$ is the perception system measured grain profile, $h_m(x, y)$ is the grain fill model estimated profile, and $N_{visible}$ is the number of visible grid inside the cart.

With the addition of the compensation flow, the average height of $h_m(x, y)$ is going to track the average height of $h_p(x, y)$, making the overall height consistent with the measured overall height.

6.2.3 Flow rate estimation

The fusion profile height correction is Sec. 6.2.2 can be used to adjust the overall height when the measurement from the perception system is active. However, when the perception system became inactive because of challenging environment or failed algorithms, the compensation flow cannot be calculated, and the estimated profile will be subject to the accumulation error from the bias in the initial flow rate estimate v_{ini}

To minimize the accumulation error even when the perception system is inactive, an algorithm is developed to give a more accurate estimation of the actual flow rate v_{est} than the initial estimate v_{ini} when the perception system is active. After that, when the perception system turns off, the grain fill model can use the updated v_{est} to continue the simulation.

The flow rate is estimated from the grain height change around the impact location. When the relative position between the two vehicle is stabilized around a certain location,

the height growth around the impact location neighborhood can be approximated as a linear growth.

$$h_{impact}\left(t\right) = k_h t - h_{impact,ini} \tag{6.7}$$

where h_{impact} is the average grain height around the impact location, $h_{impact,ini}$ is the initial grain height, and k_h is the growth rate of the grain height.

By doing a first-order fit on h_{impact}, the height growth rate can be estimated as

$$\widehat{k_h} = \frac{\sum\left(t_{\mathrm{i}} - \bar{t}\right)\left(h_{impact}\left(t_{\mathrm{i}}\right) - \overline{h_{impact}}\right)}{\sum\left(t_{\mathrm{i}} - \bar{t}\right)^2} \tag{6.8}$$

With the growth rate of the average height around the impact location, the flow rate can then be calculated as

$$v_{est} = r\frac{\widehat{k_h}}{\left(dH/dV\right)} + \left(1 - r\right)v_{\mathrm{i}ni} \tag{6.9}$$

where dH/dV denotes the unit height growth per unit grain, and r is the confidence on the estimated flow rate, which depends on the number of data points and the correlation coefficient in the linear regression.

Notice that dH/dV is not a constant, but a variable depending on the impact location and the initial profile. This value can be approximated as a constant in a short period with a stabilized impact location. Therefore, in Eq. 6.9, this value is computationally estimated from the grain fill model by virtually unloading a small amount of grain ΔV and observing the grain height change around the impact location.

$$\left(dH/dV\right)_{h(x,y)=h_m(x,y),x_{impact},y_{impact}} = \frac{h_{impact}(V + \Delta V) - h_{impact}(V)}{\Delta V} \tag{6.10}$$

Both Eqs. 6.7 and 6.10 are based on the assumption that the auger location is relative constant (i.e., the auger unloads to one location of the grain cart). This scenario is fairly common for all the control strategies developed in the automatic unloading system introduced in Ch. 4. Moreover, because the flow rate is relatively stable during unloading operation, the estimated flow rate can be reused for the harvesting of the same machine and same field.

Therefore, to ensure reliable estimation, the flow rate estimation algorithm is only engaged if the auger unloads to one location for some time (in this application, 10 seconds).

6.3 Results

The fusion algorithm was applied to in-field testing data in both manual unloading and automatically unload to validate the performance of the fusion algorithm.

6.3.1 Manual unloading with continuously working camera perception

In this test, the data were collected during manual unloading. In this test, the grain cart (model: Brandt 1020XR) was topped off by a human operator with a front-to-back strategy. Both the tractor (model: John Deere 8R340) and the combine (model: John Deere S790) were manually controlled by operators. During the test, both the stereo camera-based perception system (IPM) and the LiDAR-based benchmark perception system [75] are measuring the grain profile.

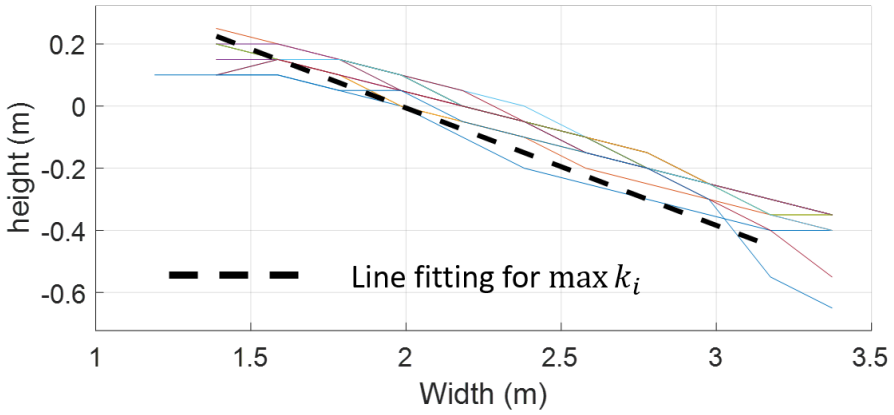
(a) (b)

Figure 6.10. Initial profile estimation for manual unloading test: (a) camera-based perception system feedback; (b) initial fill profile estimation.

As the first stage of the algorithm, the initial profile estimation algorithm estimates the complete profile from the perception system feedback. Figure 6.10 shows the raw grain profile measured by the perception system and the estimated initial profile.

The initial profile estimation algorithm first extracts the peaks from the IPM profile as shown in Fig. 6.11a. Besides, the piling angle is also estimated from the IPM profile by finding the maximum angle from the lateral cross-sections from the IPM profile as shown

(a) (b)



(c)

Figure 6.11. Initial profile estimation process for manual unloading test: (a) peaks extracted from the IPM feedback; (b) estimated piles with cart geometry; (c) piling angle estimation.

in Fig. 6.11c. With the estimated peak locations and the current piling angle of the grain inside the cart, the combined piles from the peaks and the cart geoemtry can be calculated from Eq. 6.3 as shown in Fig. 6.11b. Finally, the algorithm fuses the estimated piles with the IPM profile to a final complete map as shown in Fig. 6.10b.

At the beginning of the unloading process at $t = 22s$, the relative position of the auger stabilizes and the flow rate estimation algorithms starts. The estimation result as shown in Fig. 6.12 is calculated from the local height growth of the profile and its growth rate estimation as shown in Fig. 6.13. The estimation results do not go into the grain fill model

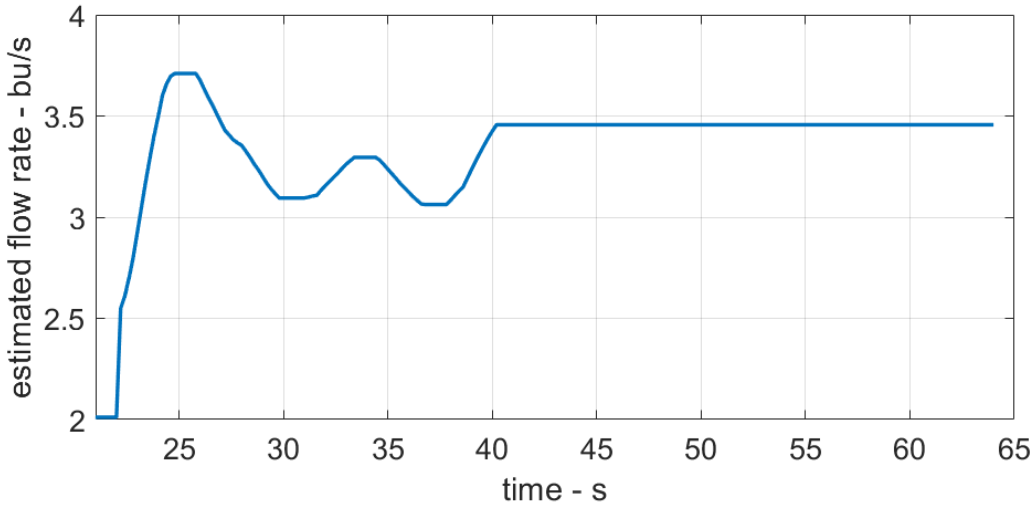


Figure 6.12. Flow rate estimation in manual unloading test

until $t = 31s$ when the auger position has stayed at a fixed location for more than 10 seconds. After $t = 40.2s$, the flow rate estimation algorithm stops updating the flow rate value because the relative location between two vehicles begins to move. However, to alleviate the accumulation error and track the overall height measured by the camera-based system, the overall height adjustment algorithm continues calculating the grain volume adjustment to the grain fill model input. Figure 6.14 shows the grain volume adjustment in this process. The value keeps updating after the auger moves.

To illustrate the grain profile change during the unload, Fig. 6.15 (associated with Visualization 18) shows the comparison among the grain profile resulting from three different methods: grain fill model only, camera perception only, and fusion from these two approaches. Figure 6.16 shows the corresponding error compared with the benchmark data.

To quantify the error in the grain profile estimation method, the mean absolute error of the profile was calculated as

$$MAE\left(\mathrm{e}\left(x,y\right)\right) = \frac{\sum_{\mathrm{i}=-p}^{p}\sum_{\mathrm{j}=-q}^{q}\left|\mathrm{e}\left(x_{auger}+w\mathrm{i},y_{auger}+w\mathrm{j}\right)\right|}{N_{available}} \tag{6.11}$$

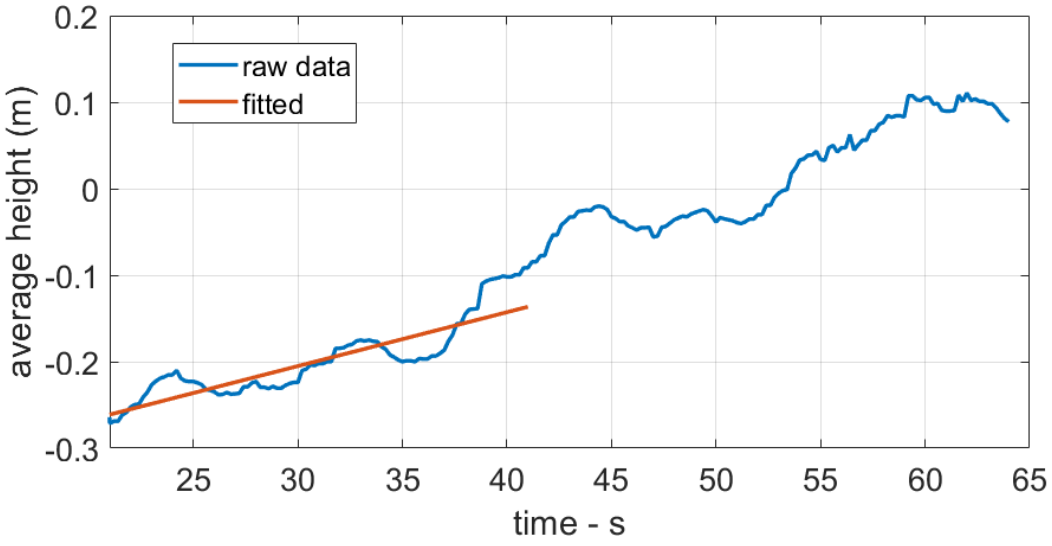


Figure 6.13. The local height growth and its growth rate estimation in manual unloading test

where w is the width of a grid, $N_{available}$ is the number of visible grids from the perception system feedback, p, q indicates the size of neighborhood in longitudinal and lateral direction separately, (x_{auger}, y_{auger}) is the auger location, and $\mathrm{e}(x, y)$ is the height error map defined as

$$\mathrm{e}(x, y) = h(x, y) - h_{benchmark}(x, y) \tag{6.12}$$

here $h(x, y)$ is the grain profile to be evaluated, $h_{benchmark}(x, y)$ is the benchmark profile obtained from the LiDAR-based benchmark system.

Figure 6.17 shows the mean absolute error comparison among the perception system measurement, grain fill model only, and fusion results. The estimation results after fusion are better than both the grain fill model and the camera-based perception system alone. As shown in the yellow line in Fig. 6.17, the estimation with the grain fill model only suffers from the accumulation error due to a bias in the flow rate input. Meanwhile, the error of the fusion estimation does not vary too much over time, showing the effectiveness of flow rate estimation and grain height compensation. When compared with the perception system only estimation, the fusion results not only have a more complete coverage than the perception

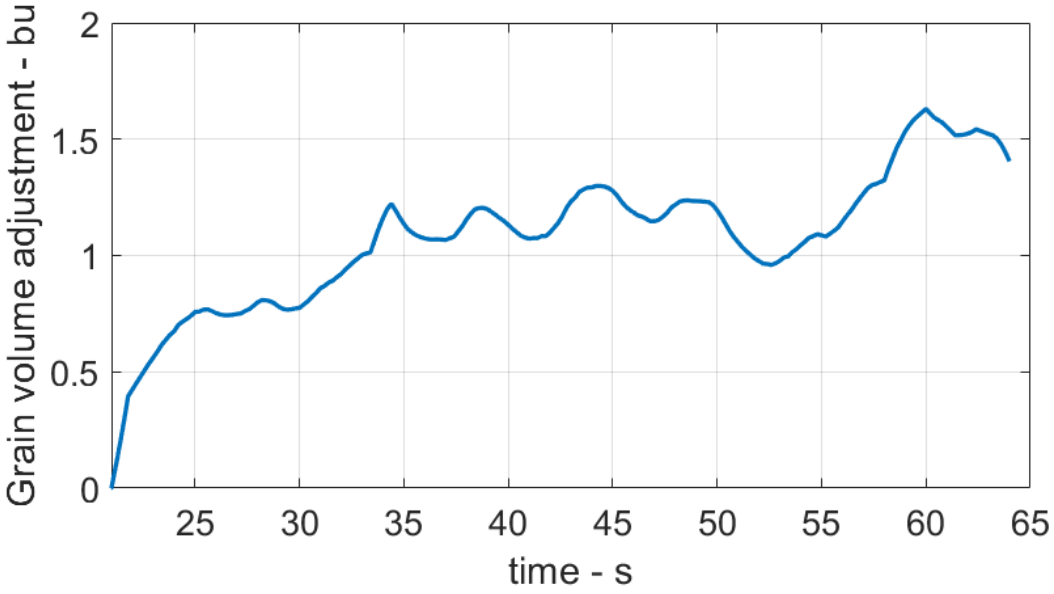


Figure 6.14. Grain volume adjustment in manual unloading test

system measurement as shown in Fig. 6.15, but also has a lower error thanks to the more accurate geometry from the grain fill model.

6.3.2 Automatic unloading with intermittently working camera perception

The same algorithm was applied to the data collected from an automatic unloading in-field testing. In this test, the relative position of the tractor and the combine harvester is controlled by an automatic unloading controller with a front-to-back-to-front unloading strategy. In this unloading operation, the camera-based perception system lost track of the grain cart intermittently. Within the 32-second unloading time, the camera-based perception did not provide fill profile feedback for more than 15 seconds, causing the system to halt automatic offloading.

Before the auger turns on, IPM has stable feedback. For the first step, as shown in Fig. 6.18, the initial profile estimation algorithm estimates the entire grain profile based on the IPM feedback.

After the auger turns on, the flow rate estimation algorithm begins to estimate the flow rate. Figure 6.19 shows the flow rate estimation result during unloading. Ten seconds after

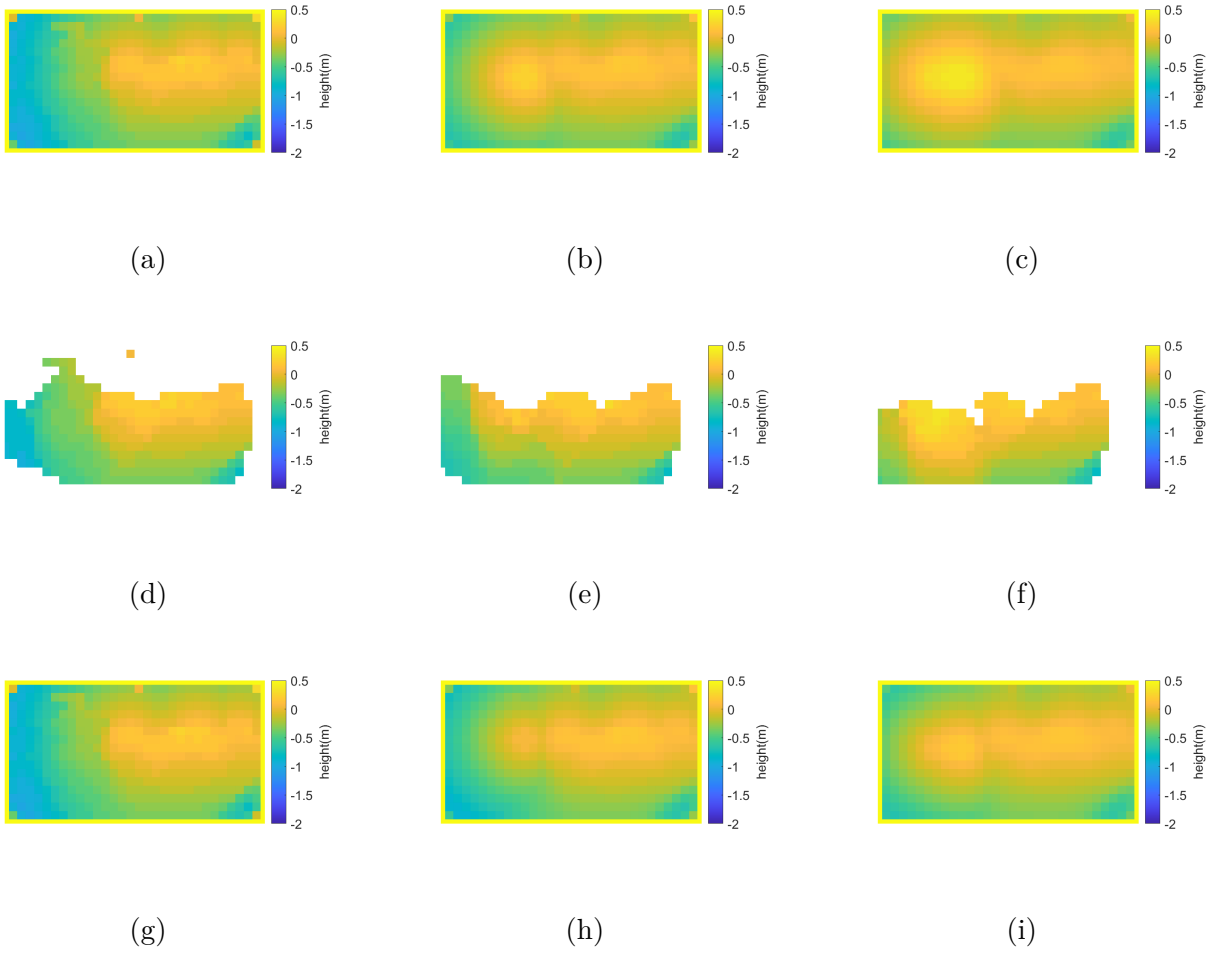


Figure 6.15. Grain profile in manual unloading test (associated with Visualization 18): (a) fusion profile at t = 22 s; (b) fusion profile at t = 40 s; (c) fusion profile at t = 58 s; (d) perception system measurement at t = 22 s; (e) perception system measurement at t = 40 s; (f) perception system measurement at t = 58 s; (g) grain fill model alone at t = 22 s; (h) grain fill model alone at t = 40 s; (i) grain fill model alone at t = 58 s.

the auger starts at $t = 46.2s$, the estimated flow rate replaces the original initial flow rate in the grain fill model simulation. After that, at $t = 49s$, IPM loses track of the grain cart, and the active period of the IPM did not exceed ten seconds, the flow rate was not updated after $t = 49s$ in this unloading operation.

Figures 6.20-6.21 (associated with Visualization 19) show the estimated/measured change of grain profile with three different approaches and their corresponding error. Because the

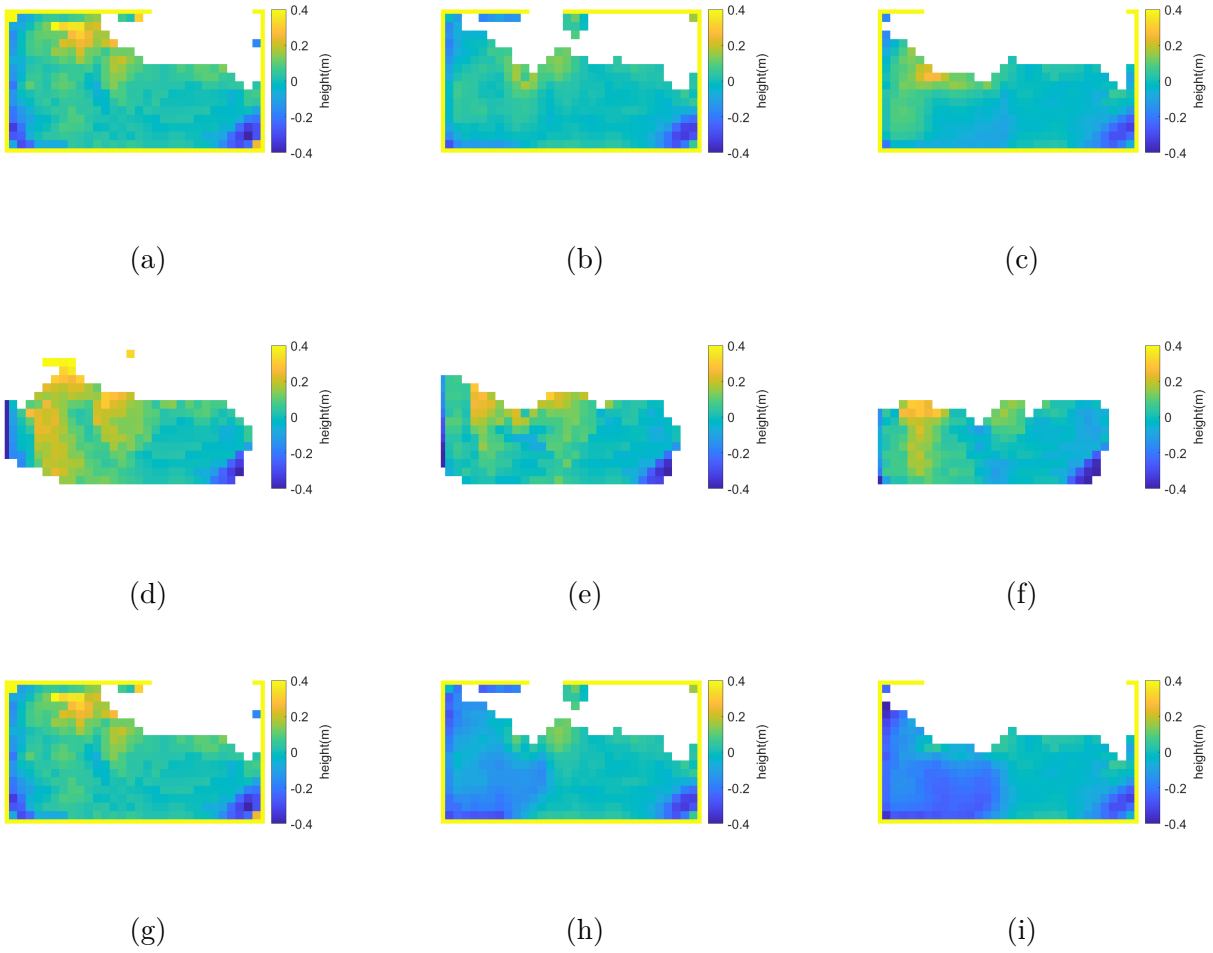


Figure 6.16. Grain profile error in manual unloading test (associated with Visualization 18): (a) fusion profile at t = 22 s; (b) fusion profile at t = 40 s; (c) fusion profile at t = 58 s; (d) perception system measurement at t = 22 s; (e) perception system measurement at t = 40 s; (f) perception system measurement at t = 58 s; (g) grain fill model alone at t = 22 s; (h) grain fill model alone at t = 40 s; (i) grain fill model alone at t = 58 s.

camera-based perception system lost track during unloading, its data are not available in certain periods. Additionally, after IPM regained the tracking of the grain cart, as shown in Fig. 6.20f, the number of visible grids significantly reduces, making it challenging to monitor the grain profile change across different parts of the grain cart. In comparison, both the model-only method and the fusion method maintain full coverage throughout the entire unloading operation.

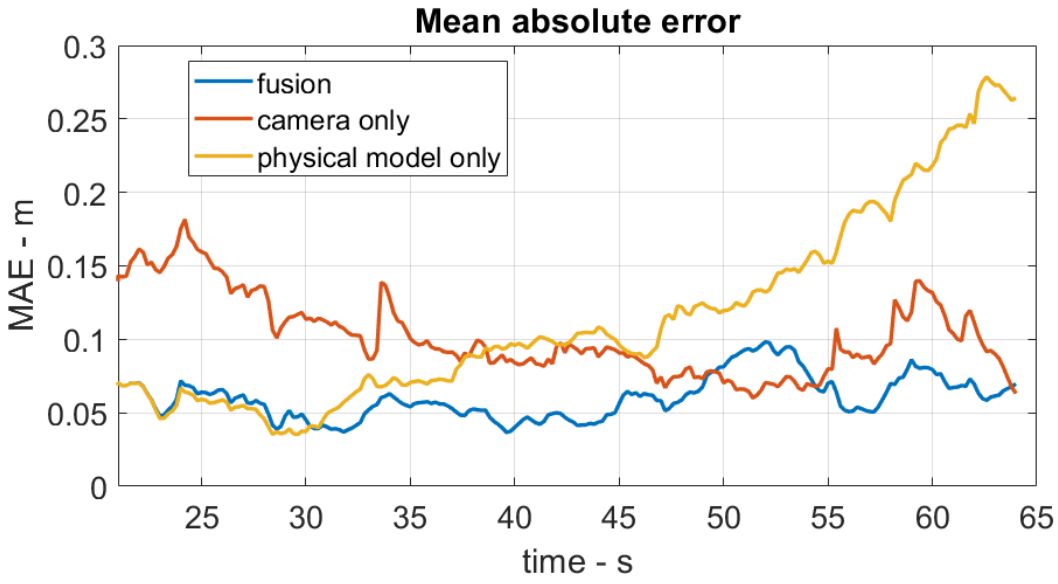


Figure 6.17. Mean absolute error analysis for the neighboring grids around the auger location in manual unloading test

(a) (b)

Figure 6.18. Initial profile estimation in the automatic unloading test (a) camera-based perception system feedback; (b) initial fill profile estimation.

Figure 6.22 summarizes the mean absolute error change in the operation. The fusion method demonstrates a higher accuracy and consistency for the data from automatic unloading in-field testing as well. The fusion method not only significantly reduces the accumulation error compared with the model-only method, but also shows better accuracy, larger coverage, and higher robustness to dust compared with the camera-only approach.

In summary, the in-field testing data for both manual unloading and automatic unloading demonstrate that the fusion result between camera measurement and grain fill model

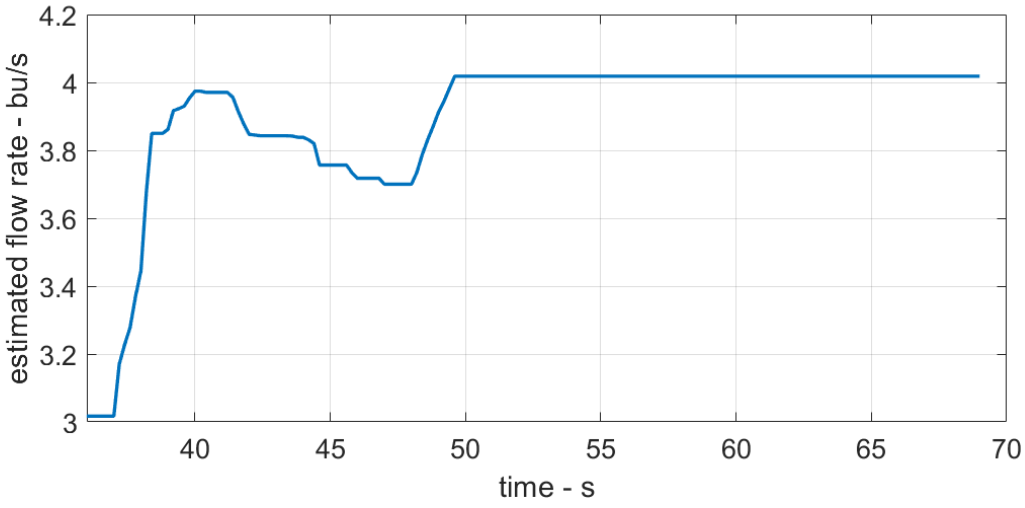


Figure 6.19. Flow rate estimation in automatic unloading test

has higher accuracy, greater coverage, and more robustness to the environment than either camera measurement or grain fill model alone.

More specifically, the accumulated MAE for the grain fill model alone can double in only 20 seconds, making the algorithm not feasible for a typical unloading operation, which can last more than 40 seconds.

On the other hand, the maximum mean absolute error of the IPM data can be about 0.1 meters higher than the fusion results. In other words, to prevent spillage, the target fullness for using a camera-based perception system only needs to be lower than the perception based on fusion results. For the grain cart model used in the testing, it would result in about 60 bushels (i.e., 6% of grain cart capacity) less grain for each cart-load.

6.3.3 Manual unloading with continuously loss camera perception

It has been demonstrated that the fusion algorithm can improve perception quality in both manual unloading and automatic unload. However, in these two cases, the perception system is either always available (manual unloading) or has inactive periods of less than 10 seconds (automatic unload) because a longer inactive period will stop the unloading. Besides these two cases, another case of interest is an unloading session with inactive camera

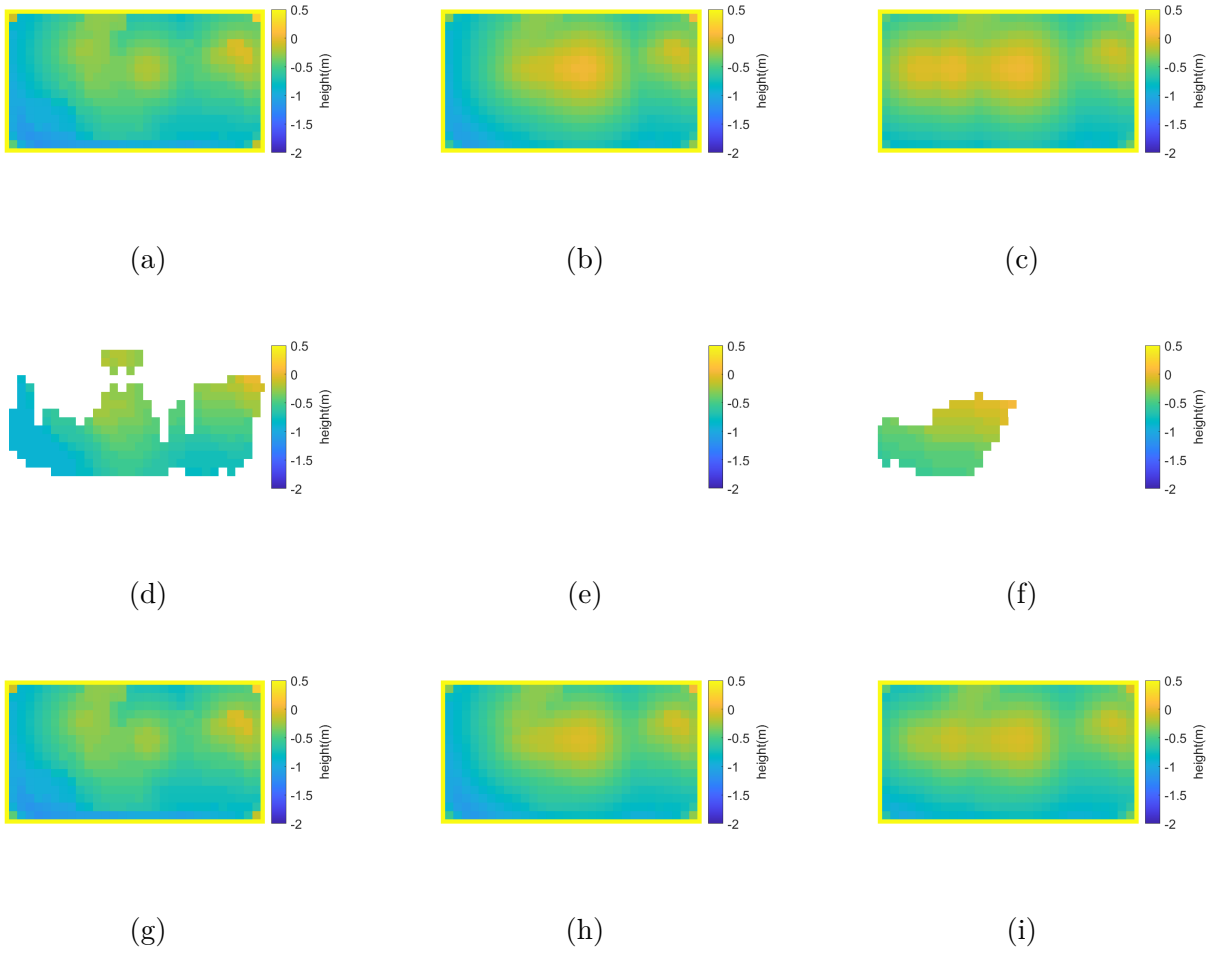


Figure 6.20. Grain profile in automatic unloading test (associated with Visualization 19): (a) fusion profile at t = 40 s; (b) fusion profile at t = 53 s; (c) fusion profile at t = 66 s; (d) perception system measurement at t = 40 s; (e) perception system measurement at t = 53 s; (f) perception system measurement at t = 66 s; (g) grain fill model alone at t = 40 s; (h) grain fill model alone at t = 53 s; (i) grain fill model alone at t = 66 s.

perception for longer. Data with a longer inactive period can validate fusion algorithm performance in more challenging conditions.

Because the length and frequency of the inactive period of IPM are fairly random, it is challenging to directly obtain desirable data from in-field testing. Therefore, to obtain equivalent data, the manual unloading data in Sec. 6.3.1 was adapted by removing the camera

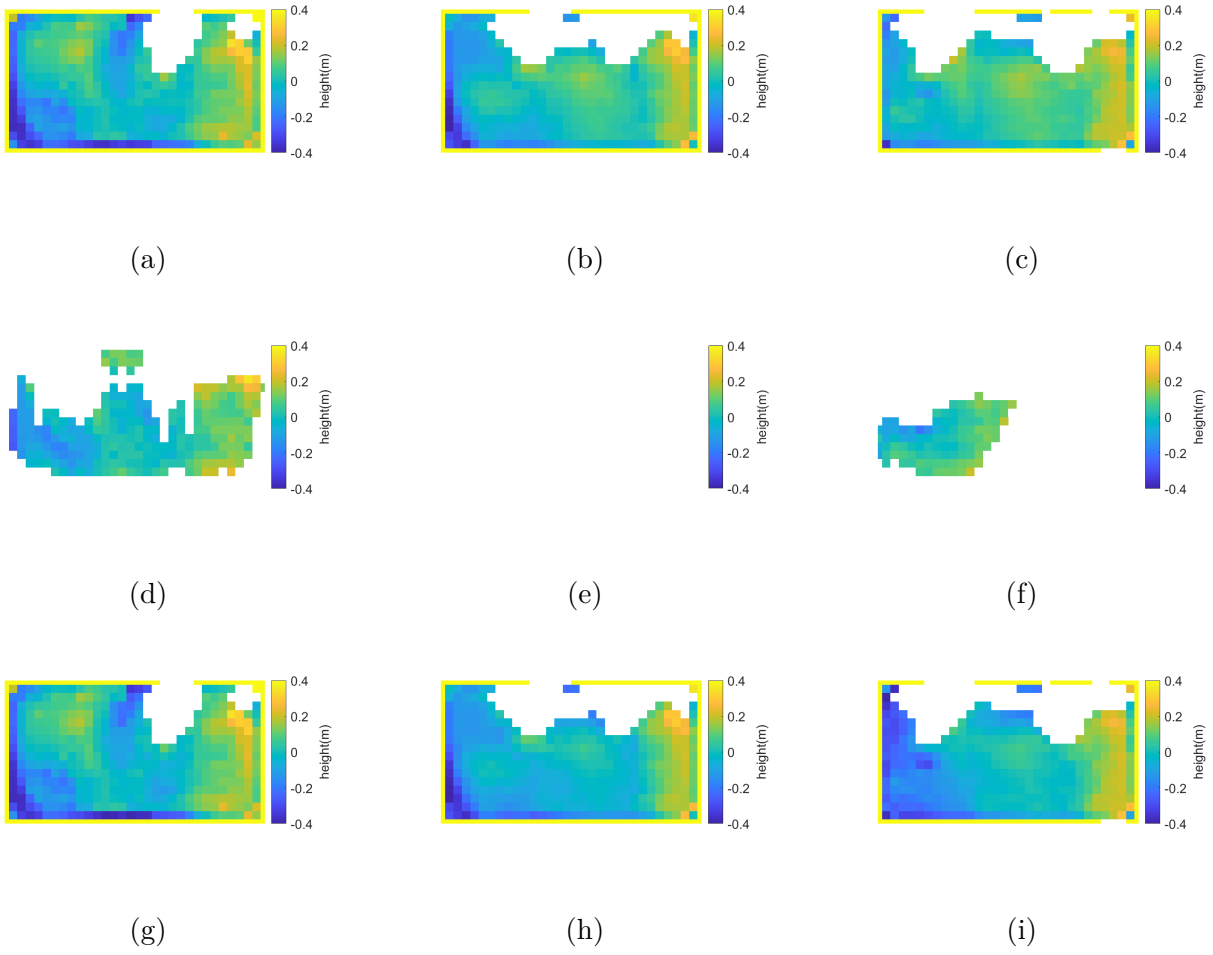


Figure 6.21. Grain profile error in automatic unloading test (associated with Visualization 19): (a) fusion profile at t = 40 s; (b) fusion profile at t = 53 s; (c) fusion profile at t = 66 s; (d) perception system measurement at t = 40 s; (e) perception system measurement at t = 53 s; (f) perception system measurement at t = 66 s; (g) grain fill model alone at t = 40 s; (h) grain fill model alone at t = 53 s; (i) grain fill model alone at t = 66 s.

feedback for certain period. A 20-second inactive period from $t = 34s$ to $t = 54s$ was added to the manual unloading data to validate the algorithm.

Figure 6.23 shows the flow rate estimation. Because the raw data is the same as Sec. 6.3.1, the estimated flow rate is the same as the estimation in Fig. 6.12 from $t = 21s$ to $t - 34s$. However, at $t = 34s$, no camera feedback goes into the fusion algorithm, the flow rate estimation does not update anymore. Additional, the estimation of the flow rate stays

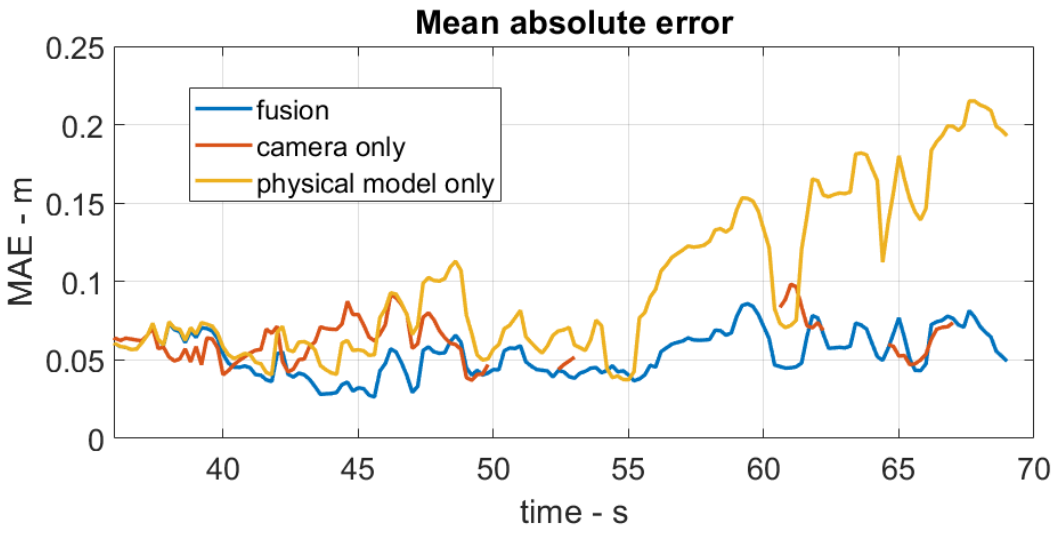


Figure 6.22. Mean absolute error analysis for the neighboring grids around the auger location in automatic unloading test

constant after $t = 54s$ when the IPM becomes active again because the auger does not stay in one grain cart location for more than 10 seconds.

Even though the flow rate does not update after $t = 54s$ because of the changing relative position, the fusion algorithm still uses the camera perception feedback to adjust the overall height of the estimated profile by adding or subtracting grain volume from the estimation. Figure 6.24 shows the grain volume adjustment amount based on the difference between the estimated grain height and the IPM measured grain height. Unlike the flow rate estimation that requires the relative location to be static, the adjustment keeps working whenever the IPM is active.

Figures 6.25 - 6.26 show the grain profile change and its corresponding error with three different approaches. Visualization 20 shows the change in the entire process. It can be seen that the both the fusion approach and the model-only approach provide complete coverage the entire time, while as shown in Figs. 6.26g - 6.26i, the bias accumulate in model-only approach over time.

Figure 6.27 shows the mean absolute error for the manual unloading data with extended period of inactive camera perception. Even with an inactive period of the camera perception for as long as 20 seconds, the result from the fusion algorithm still demonstrates better

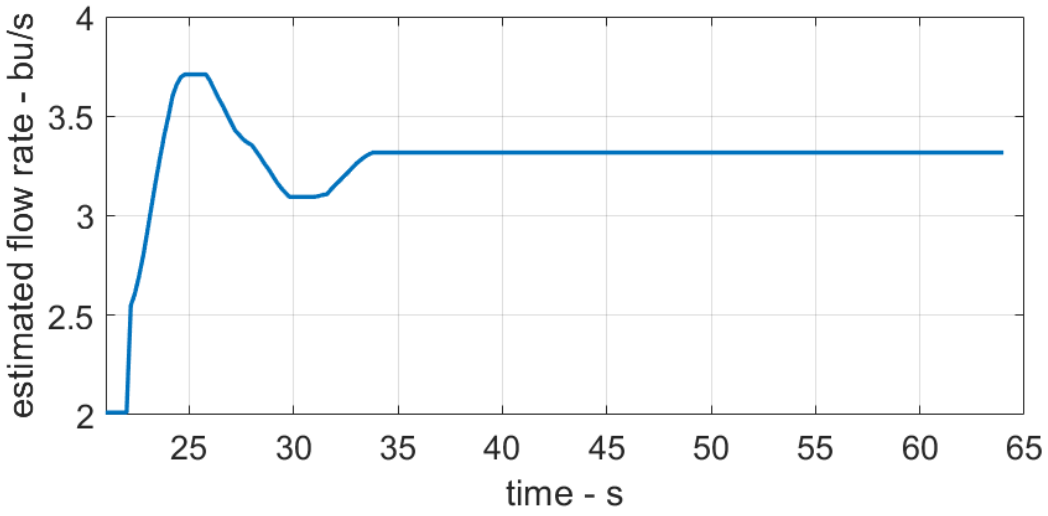


Figure 6.23. Flow rate estimation in manual unloading test with long inactive period

accuracy than the camera-only approach and the model-only approach. Because of the flow rate estimation before the inactive period, the fusion approach has much slower accumulation error growth during the inactive time compared with the model-only approach. Additionally, after IPM becomes active again at $t = 54s$, the additional volume calculated from the height correction as shown in Fig. 6.24 further brings down the error.

The estimation accuracy shown in this test shows that the fusion model can be an effective way to provide grain fill profile feedback even if the camera perception is down for a long period of time as long as there has already been a good estimate on the grain flow rate. The flow rate estimation can either comes from the current load if the relative location of the two vehicles stays relatively static for more than 10 seconds, or from a previous estimation with similar crop unloading at the same day.

6.3.4 Computation efficiency

To demonstrate the computation efficiency of the fusion algorithm, the computation time was counted for the data processing. The algorithm was implemented in MATLAB 2019b and run on a Windows laptop (CPU: Intel Core i7-9750H). Table 6.2 shows the run time of each algorithm component of the fusion algorithm.

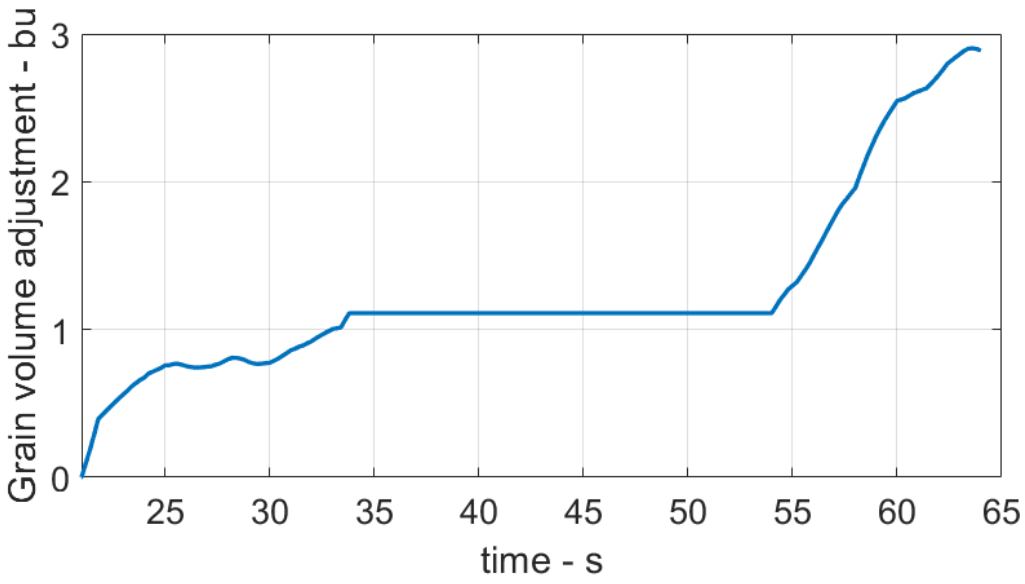


Figure 6.24. Grain volume adjustment in manual unloading test with long inactive period

Table 6.2. Algorithm run time for the fusion algorithm with 5Hz updating rate

Algorithm	Computation time - s	In-field testing time - s
Initial profile estimation	0.35	N/A
Grain profile fusion - manual unload test	2.74	43
Grain profile fusion - automatic unloading test	3.54	33

The computation time was evaluated in two different parts, the initial profile estimation and the fusion algorithm (including fusion profile height correction, flow rate estimation, and grain fill model). The initial profile only needs to run once for each unloading, so its computation time was not compared with the actual time. The fusion part, however, needs to be computed contiguously during the unloading operation. Therefore, Tab.6.2 compares it with the actual testing time. Running at 5Hz, the fusion algorithm runs much faster than the actual unloading operation for both the manual unloading test and the automatic unloading test (with intermittent camera feedback). The computation time is one magnitude smaller than the actual time in the in-field testing, showing that the fusion algorithm can run in

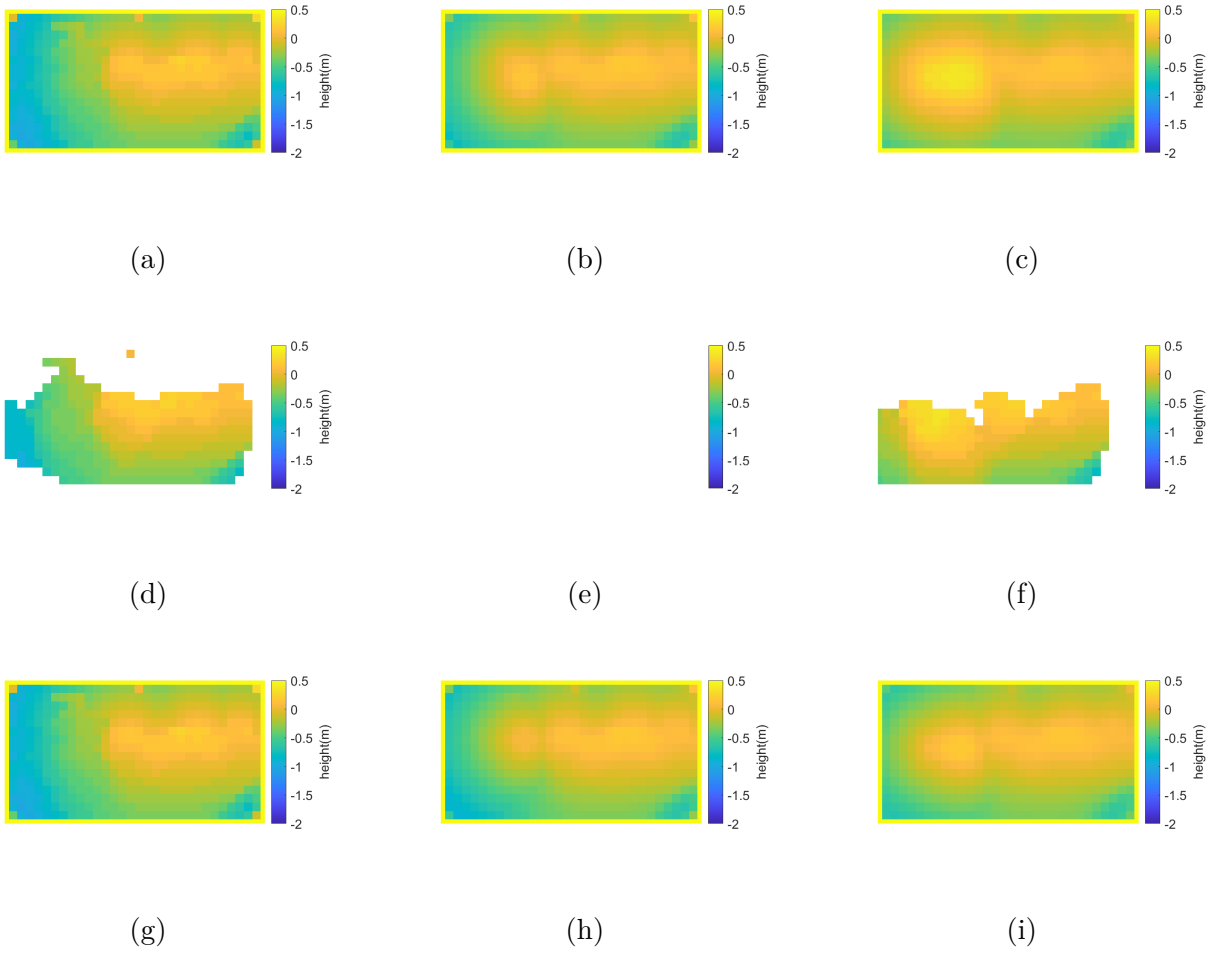


Figure 6.25. Grain profile in manual unloading test with extended period of inactive camera perception (associated with Visualization 20): (a) fusion profile at t = 22 s; (b) fusion profile at t = 40 s; (c) fusion profile at t = 58 s; (d) perception system measurement at t = 22 s; (e) perception system measurement at t = 40 s; (f) perception system measurement at t = 58 s; (g) grain fill model alone at t = 22 s; (h) grain fill model alone at t = 40 s; (i) grain fill model alone at t = 58 s.

real-time for the unloading operation and potentially be added to the automatic unloading system without too much computational cost.

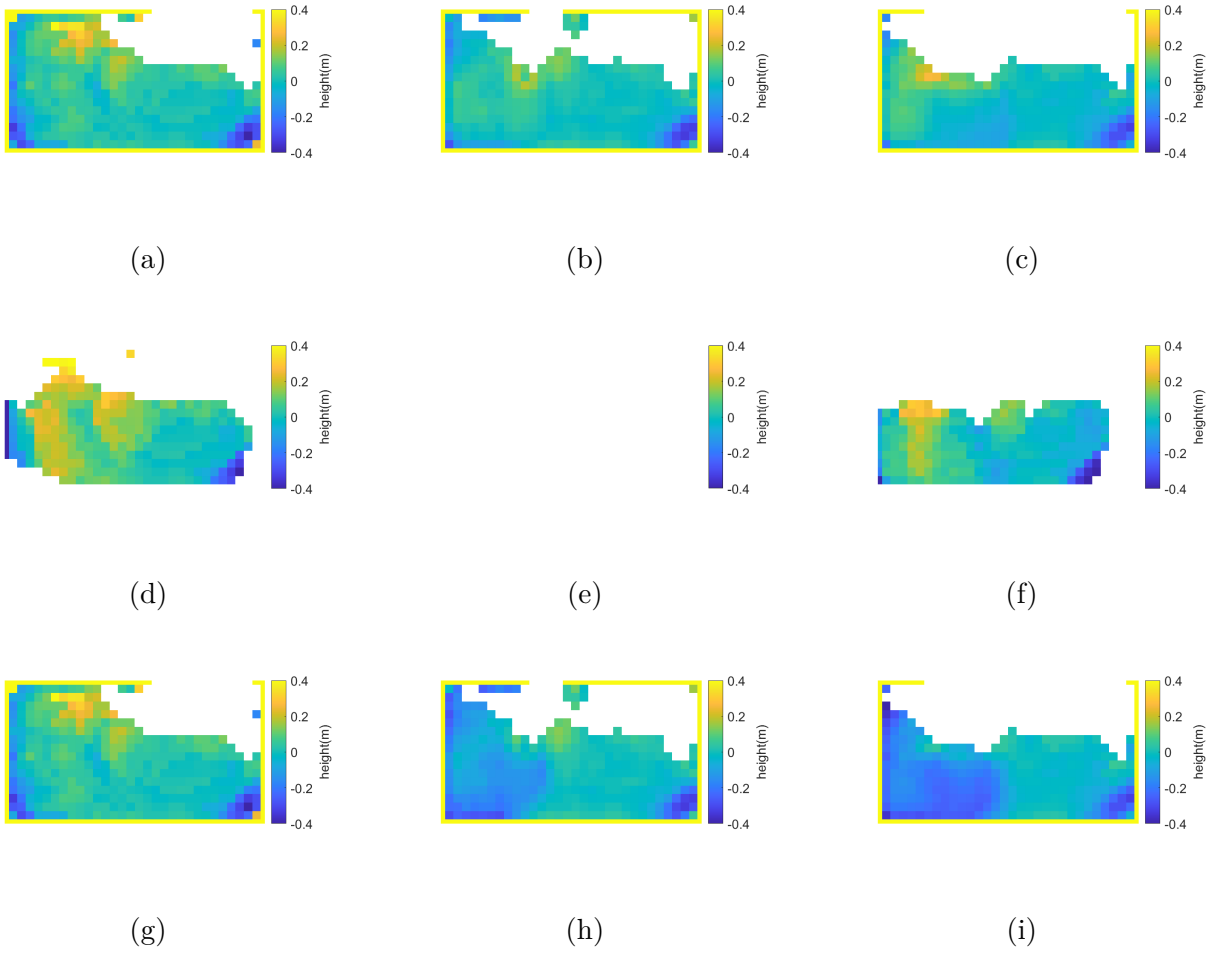


Figure 6.26. Grain profile error in manual unloading test with extended period of inactive camera perception (associated with Visualization 20): (a) fusion profile at t = 22 s; (b) fusion profile at t = 40 s; (c) fusion profile at t = 58 s; (d) perception system measurement at t = 22 s; (e) perception system measurement at t = 40 s; (f) perception system measurement at t = 58 s; (g) grain fill model alone at t = 22 s; (h) grain fill model alone at t = 40 s; (i) grain fill model alone at t = 58 s.

6.4 Discussion

The results from the in-field testing data for both manual unloading and automatic unloading demonstrate the effectiveness and computational efficiency of the fusion algorithm. For the fusion algorithm to work properly, there are several assumptions for the unloading operation:

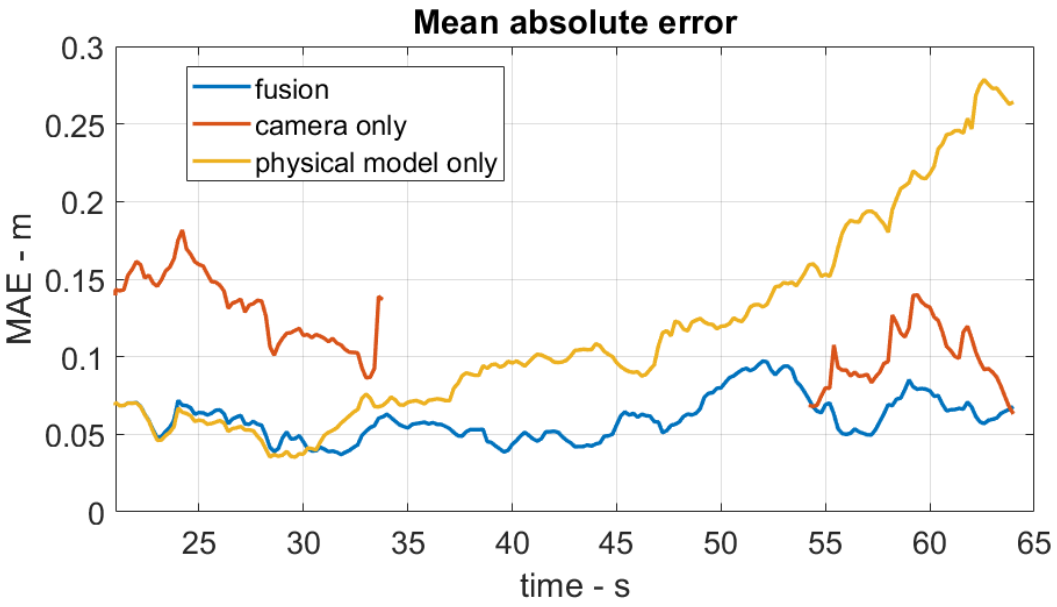


Figure 6.27. Mean absolute error analysis for the neighboring grids around the auger location in manual unloading test with extended period of inactive camera perception

- A good flow rate estimation from either prior estimation with similar operating conditions or from the current unloading with the auger stabilizes at one location for more than 10 seconds. For the strategy adopted in the automatic unloading system developed in this thesis, one row of the grain cart needs to be filled up before the auger moves to the next row, As a result, it happens frequently that the auger stays at one location for a longer time.

- Prior knowledge of the vehicle geometry for the auger location estimation and initial profile estimation. Such information can usually be extracted from the specifications of the vehicles.

- The camera-based perception system has an accurate measurement when the auger is off so the initial profile can estimated more accurately.

- The overall height of the camera-based perception system is still reliable after auger turns on, so the volume of the profile can be corrected.

In summary, by applying the fusion algorithm to the automatic unloading system when the assumptions hold true, it can

- Increase grain transfer efficiency potentially. During unloading, the fusion algorithm results can have an mean absolute error (MAE) 0.1 m lower than the camera perception alone. To prevent spillage, unloading with camera perception alone may require a more conservative fill target. For the grain cart used in this project, 0.1-meter height difference results in 60 bushel (6%) capacity difference.

- Make the automatic unloading system more robust to different environment. The fusion result can still provide reliable grain profile feedback with negligible accumulation error when the perception system is inactive for as long as 20 seconds. This allows the automatic unloading system to continue working for longer after camera perception is down.

- Prevent spillage. The fusion algorithm estimates the grain profile of the entire grain cart, and thus enables the spillage detection at the farther edge of the grain cart, which is not visible to the camera-based perception system due to the occlusion of the grain piles.

6.5 Conclusion

A fusion algorithm for the grain pile perception was developed to fuse the data from the camera-based perception system and a grain fill model based on the characteristics of the system. The fusion algorithm consists of three components: initial profile estimation, overall height correction, and flow rate estimation. Data from in-field testing of combine harvester unloading-on-the-go demonstrates the effectiveness and computation efficiency of the algorithm. For both the manual unloading and automatic unloading, the results from the fusion algorithm has higher accuracy, more complete coverage of the grain cart, and more robust perception than either the camera-based system or the grain fill model alone. If applied in an automatic unloading system, such advantageous perception results from

fusion could increase grain transfer efficiency, make the automated system more robust to environmental changes, and prevent spillage.

REFERENCES

[1] N. Alexandratos and J. Bruinsma, "World agriculture towards 2030/2050: The 2012 revision," 2012.

[2] S. Bringezu, M. O'Brien, W. Pengue, M. Swilling, and L. Kauppi, "Assessing global land use and soil management for sustainable resource policies," *Scoping Paper. International Panel for Sustainable Resource Management. UNEP*, 2010.

[3] D. R. Reidmiller, C. W. Avery, D. R. Easterling, *et al.*, "Impacts, risks, and adaptation in the united states: Fourth national climate assessment, volume ii," 2017.

[4] T. Hertz and S. Zahniser, "Is there a farm labor shortage?" *American Journal of Agricultural Economics*, vol. 95, no. 2, pp. 476–481, 2013.

[5] J. E. Taylor, D. Charlton, and A. Yúnez-Naude, "The end of farm labor abundance," *Applied Economic Perspectives and Policy*, vol. 34, no. 4, pp. 587–598, 2012.

[6] M. De Clercq, A. Vats, and A. Biel, "Agriculture 4.0: The future of farming technology," *Proceedings of the World Government Summit, Dubai, UAE*, pp. 11–13, 2018.

[7] I. Charania and X. Li, "Smart farming: Agriculture's shift from a labor intensive to technology native industry," *Internet of Things*, vol. 9, p. 100 142, 2020.

[8] M. Reinecke, H.-P. Grothaus, G. Hembach, S. Scheuren, and R. Hartanto, "Dynamic and distributed infield-planning system for harvesting," in *2013 Kansas City, Missouri, July 21-July 24, 2013*, American Society of Agricultural and Biological Engineers, 2013, p. 1.

[9] N. Delchev, K. Trendafilov, G. Tihanov, Y. Stoyanov, *et al.*, "Grain combines productivity according to various unloading methods-in the field and at the edge of the field.," *Agricultural Science and Technology*, vol. 8, no. 3, pp. 221–226, 2016.

[10] B. Bashiri and D. D. Mann, "Impact of automation on drivers' performance in agricultural semi-autonomous vehicles," *Journal of agricultural safety and health*, vol. 21, no. 2, pp. 129–139, 2015.

[11] J. D. Yegerlehner, *Curved path approximation in vehicle guidance systems and methods*, US Patent 9,804,603, Oct. 2017.

[12] Y. Gunbatar, *Transformer (modifier) design for controlling articulated vehicles smoothly*, US Patent 10,599,151, Mar. 2020.

[13] R. R. Zielke and J. F. Howard, *Combine bin level monitoring system*, US Patent 9,043,096, May 2015.

[14] M. A. Schmidt, K. E. Hunt, D. R. Holm, and S. A. Stephens, *Method and system for following a lead vehicle*, US Patent 7,593,811, Sep. 2009.

[15] P. Metzler, W. Flohr, and M. Höh, *System for determining the relative position of a second farm vehicle in relation to a first farm vehicle*, US Patent 7,480,564, Jan. 2009.

[16] O. Peters, N. Tarasinski, K. Hahn, and P. Pickel, *Method and device for steering a second agricultural machine, which can be steered to drive over a field parallel to a first agricultural machine*, US Patent 8,755,976, Jun. 2014.

[17] C. R. Mott and A. K. W. Rekow, *Method and system for controlling the loading of a container associated with a vehicle*, US Patent 8,060,283, Nov. 2011.

[18] R. Morselli and J. Posselius, *Method and system for coordinated vehicle control with wireless communication*, US Patent 8,160,765, Apr. 2012.

[19] T. J. Dollinger, B. R. Ray, A. R. Rusciolelli, *et al.*, *System and method for coordinated control of agricultural vehicles*, US Patent 9,915,952, Mar. 2018.

[20] K. Manufacturing, *Kinze adds new features to its autonomous harvest system*, Dec. 2014. [Online]. Available: https://www.kinze.com/kinze-adds-new-features-to-its-autonomous-harvest-system/, (accessed: 11.28.2020).

[21] N. T, *Autonomous tractors are in fields today!* Jul. 2019.

[22] H. Kurita, M. Iida, M. Suguri, R. Masuda, and W. Cho, "Efficient searching for grain storage container by combine robot," *Engineering in agriculture, environment and food*, vol. 7, no. 3, pp. 109–114, 2014.

[23] H. Kurita, M. Iida, M. Suguri, and R. Masuda, "Application of image processing technology for unloading automation of robotic head-feeding combine harvester," *Engineering in Agriculture, Environment and Food*, vol. 5, no. 4, pp. 146–151, 2012.

[24] W. Cho, H. Kurita, M. Iida, M. Suguri, and R. Masuda, "Autonomous positioning of the unloading auger of a combine harvester by a laser sensor and gnss," *Engineering in agriculture, environment and food*, vol. 8, no. 3, pp. 178–186, 2015.

[25] Z. T. Bonefas, *Method and stereo vision system for facilitating the unloading of agricultural material from a vehicle*, US Patent 8,868,304, Oct. 2014.

[26] A. T. Jennett, "Decision support system for sensor-based autonomous filling of grain containers," 2012.

[27] B. C. Potter, "Design and assessment of an automated grain auger position control system," 2012.

[28] H. Herman, Z. T. Bonefas, C. Vallespi-Gonzalez, and J. J. Zametzer, *Optical image capture for controlling a position of a harvester transfer device*, US Patent 9,313,951, Apr. 2016.

[29] Z. T. Bonefas, *Method and stereo vision system for managing the unloading of an agricultural material from a vehicle*, US Patent 8,649,940, Feb. 2014.

[30] Z. T. Bonefas, H. Herman, and J. Campoy, *Fill level indicator for an automated unloading system*, US Patent 10,019,790, Jul. 2018.

[31] J. Huster, R. Hartmann, and N. Diekhans, *Transfer-assistance system*, US Patent 7,537,519, May 2009.

[32] J. Möller, "Computer vision–a versatile technology in automation of agricultural machinery," *Journal of Agricultural Engineering*, vol. 47, no. 4, pp. 28–36, 2010.

[33] B. M. Missotten, D. O. Verhaeghe, W. M. Byttebier, and K. M. Viaene, *Method for directing an unloading apparatus of a harvesting machine to a container*, US Patent 8,682,540, Mar. 2014.

[34] B. M. Missotten, J. Boydens, B. Depestel, P. Vanysacker, and K. Viaene, *Unloading apparatus controller for agricultural harvesting machines*, US Patent 9,655,301, May 2017.

[35] J. Boydens, B. M. Missotten, P. Vanysacker, and K. M. Viaene, *Sensor arrangement for monitoring an unloading process of an agricultural harvester*, US Patent 9,973,710, May 2018.

[36] D. Fenistein and M. van Hecke, "Wide shear zones in granular bulk flow," *Nature*, vol. 425, no. 6955, pp. 256–256, 2003.

[37] S. Sundaresan, "Some outstanding questions in handling of cohesionless particles," *Powder Technology*, vol. 115, no. 1, pp. 2–7, 2001.

[38] Z. Liu, S. Dhamankar, J. T. Evans, *et al.*, "Development and experimental validation of a system for agricultural grain unloading-on-the-go," *TBD*, vol. TBD, TBD–TBD, 2021.

[39] P. A. Cundall and O. D. Strack, "A discrete numerical model for granular assemblies," *geotechnique*, vol. 29, no. 1, pp. 47–65, 1979.

[40] A. P. Grima and P. W. Wypych, "Discrete element simulations of granular pile formation," *Engineering Computations*, 2011.

[41] T. Tian, J. Su, J. Zhan, S. Geng, G. Xu, and X. Liu, "Discrete and continuum modeling of granular flow in silo discharge," *Particuology*, vol. 36, pp. 127–138, 2018.

[42] H. Zhu, Z. Zhou, R. Yang, and A. Yu, "Discrete particle simulation of particulate systems: Theoretical developments," *Chemical Engineering Science*, vol. 62, no. 13, pp. 3378–3396, 2007.

[43] P. W. Cleary, "Large scale industrial dem modelling," *Engineering Computations*, 2004.

[44] D. G. Schaeffer, "Instability in the evolution equations describing incompressible granular flow," *Journal of differential equations*, vol. 66, no. 1, pp. 19–50, 1987.

[45] C. Lun, S. B. Savage, D. Jeffrey, and N. Chepurniy, "Kinetic theories for granular flow: Inelastic particles in couette flow and slightly inelastic particles in a general flowfield," *Journal of fluid mechanics*, vol. 140, pp. 223–256, 1984.

[46] J. T. Jenkins, S. B. Savage, *et al.*, "Theory for the rapid flow of identical, smooth, nearly elastic, spherical particles," *Journal of fluid mechanics*, vol. 130, no. 1, pp. 187–202, 1983.

[47] G. I. Tardos, S. McNamara, and I. Talu, "Slow and intermediate flow of a frictional bulk powder in the couette geometry," *Powder Technology*, vol. 131, no. 1, pp. 23–39, 2003.

[48] G. MiDi, "On dense granular flows," *The European Physical Journal E*, vol. 14, no. 4, pp. 341–365, 2004.

[49] G. Daviet and F. Bertails-Descoubes, "A semi-implicit material point method for the continuum simulation of granular materials," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–13, 2016.

[50] B. Zhu and X. Yang, "Animating sand as a surface flow.," in *Eurographics (short papers)*, 2010, pp. 9–12.

[51] P. Li, M. Ucgul, S.-H. Lee, and C. Saunders, "A new approach for the automatic measurement of the angle of repose of granular materials with maximal least square using digital image processing," *Computers and Electronics in Agriculture*, vol. 172, p. 105 356, 2020.

[52] M. Ucgul, C. Saunders, P. Li, S.-H. Lee, and J. M. Desbiolles, "Analyzing the mixing performance of a rotary spader using digital image processing and discrete element modelling (dem)," *Computers and Electronics in Agriculture*, vol. 151, pp. 1–10, 2018.

[53] H. M. B. Al-Hashemi and O. S. B. Al-Amoudi, "A review on the angle of repose of granular materials," *Powder Technology*, vol. 330, pp. 397–417, 2018.

[54] G. Happich, H.-H. Harms, and T. Lang, "Loading of agricultural trailers using a model-based method," *Agricultural Engineering International: CIGR Journal*, 2009.

[55] R. Bhadra, M. E. Casada, S. A. Thompson, *et al.*, "Field-observed angles of repose for stored grain in the united states," *Applied engineering in agriculture*, vol. 33, no. 1, pp. 131–137, 2017.

[56] L. Lennart, "System identification: Theory for the user," *PTR Prentice Hall, Upper Saddle River, NJ*, pp. 1–14, 1999.

[57] A. Giusti, J. Guzzi, D. C. Cireşan, *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[58] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[59] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[60] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1278–1286.

[61] H. Hattori, V. Naresh Boddeti, K. M. Kitani, and T. Kanade, "Learning scene-specific pedestrian detectors without real data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3819–3827.

[62] S. D. Pendleton, H. Andersen, X. Du, *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[63] W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," in *European Conference on Computer Vision*, Springer, 2016, pp. 909–916.

[64] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, Springer, 2016, pp. 102–118.

[65] *Unreal Engine official website*, https://www.unrealengine.com/en-US/, Accessed: 2021-03-27.

[66] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Pearson Education India, 2004.

[67] R. G. Willson, M. W. Maimone, A. E. Johnson, and L. M. Scherr, "An optical model for image artifacts produced by dust particles on lenses," 2005.

[68] C. J. Bartleson and E. J. Breneman, "Brightness perception in complex fields," *Josa*, vol. 57, no. 7, pp. 953–957, 1967.

[69] C. Bartleson, "Optimum image tone reproduction," *Journal of the SMPTE*, vol. 84, no. 8, pp. 613–618, 1975.

[70] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[71] H. Kwakernaak, "Robust control and h infinity optimization tutorial paper," *automatica*, vol. 29, no. 2, pp. 255–273, 1993.

[72] K. Glover and J. C. Doyle, "State-space formulae for all stabilizing controllers that satisfy an h infinity norm bound and relations to relations to risk sensitivity," *Systems & control letters*, vol. 11, no. 3, pp. 167–172, 1988.

[73] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard h 2 and h infinity control problems," in *1988 American Control Conference*, IEEE, 1988, pp. 1691–1696.

[74] *Active Fill Control*, https://www.nrec.ri.cmu.edu/solutions/agriculture/other-agriculture-projects/active-fill-control.html, Accessed: 2021-03-11.

[75] C. Jiang, Z. Liu, L. J. Heusinger, G. M. Shaver, J. T. Evans, and C. M. Puryk, "Lidar-based benchmark approach development and validation for unloading-on-the-go systems incorporating stereo camera-based perception," *TBD*, vol. TBD, TBD–TBD, 2021.

[76] *John Deere AutoTrac*, https://www.deere.com/en/technology-products/precision-ag-technology/guidance/auto-trac/, Accessed: 2021-03-18.

[77] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[78] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," 1961.

[79] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[80] A. Gelb, *Applied optimal estimation*. MIT press, 1974.

[81] E. A. Wan, R. Van Der Merwe, and S. Haykin, "The unscented kalman filter," *Kalman filtering and neural networks*, vol. 5, no. 2007, pp. 221–280, 2001.

[82] G. Evensen, "The ensemble kalman filter for combined state and parameter estimation," *IEEE Control Systems Magazine*, vol. 29, no. 3, pp. 83–104, 2009.

[83] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994.

[84] G. Evensen, *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[85] K. Ileleji and B. Zhou, "The angle of repose of bulk corn stover particles," *Powder Technology*, vol. 187, no. 2, pp. 110–118, 2008.