THE DEVELOPMENT OF A VIRTUAL OPERATOR MODEL TO ENABLE CLOSED-LOOP, COMPUTER-BASED DESIGN

by

Yu Du

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Dr. Michael Dorneich, Major Professor
Dr. Brian Steward
Dr. Stephen Gilbert
Dr. Charles Jahren
Dr. Richard Stone

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation.

First, I would like to thank my advisor, Dr. Michael Dorneich for his incredible guidance, patience, and support throughout the course of this research. His insights and words of encouragement have always inspired me and renewed my hopes for completing my graduate education. I also would like to thank Dr. Brian Steward for his patience and incredible guidance throughout my research project. I would also like to thank my committee members, Dr. Stephen Gilbert, Dr. Charles Jahren, and Dr. Richard Stone, for their efforts and contributions to this study.

I would also like to thank my mother Lvyi Ma, and my father Hengyun Du, whose advice, attitudes, and actions have been such a great positive influence on my life and my work.

I would additionally like to thank my wife Huan Zhang and my sons Lucas Du and Alexander Du who accompanied me along this journey.

Finally, I would like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience. I want to also offer my appreciation to those who were willing to participate in my experiments and surveys, without whom, this dissertation would not have been possible.

ABSTRACT

This research investigates how machine operator expertise, strategies, and decision-making can be integrated into operator models that simulate authentic human behavior in construction machine operations. Physical prototype tests of construction machines require significant time and cost. However, computer-based simulation is often limited by the fidelity in which human operators are modeled. A greater understanding of how highly skilled operators obtain high machine performance and productivity can inform machine development and advance construction automation technology. The initial effort of this work was to develop a virtual operator model (VOM) through a combination of human factors and dynamic modeling techniques. Operator interviews were conducted to build a framework of tasks, strategies, and cues commonly used while controlling an excavator through repeating work cycles. A closed loop simulation demonstrated that an operator model could simulate the trenching work cycle. Once a VOM has been developed that is capable of closing the loop to simulate equipment operation, machine assessment can be performed earlier in the development process without physical prototyping, which reduces cost and development cycles. Advancing the state of the art in operator modeling requires models that can adapt and learn. This work investigated approaches to enable a generic virtual operator model to adapt to machines with different dimensions and capabilities without need to tune the model, adapt to changes in the environment based on the operator's actions, and adapt to differences in operator skill levels. Finally, learning capabilities and strategy models are going to be developed for the VOM, which will enable virtual operator model to understand machine models, learn during operation and choose appropriate strategies for operation.

CHAPTER I: INTRODUCTION

This overall objective of this research was to investigate how machine operator expertise, strategies, decision-making, and learning can be integrated into virtual operator models (VOMs) that simulate authentic human behavior in construction machine operations. A VOM can be used in closed loop simulation with a vehicle simulation early in the design phase of new vehicle development. This research was divided into three phases. The objective of the first phase was to capture the behavior and performance of a human operator and represent the operator in a VOM that simulates authentic human behavior in a well-defined construction machine operation. The objective of the second phase was to advance VOM adaptability to changes in environment, adaptability to changes in the dimensions of vehicle models, and better represent expertise. The objective of the third phase was to represent the process by which a VOM can learn the optimal control inputs for operation of a virtual excavator.

Introducing new product features can affect machine performance goals such as higher productivity or fuel economy. Traditional validation and assessment methods include physical machine prototype, human operator, and working tasks, which is a typical operator in the loop system (Filla, Ericsson, & Palmberg, 2005). Virtual design, the process by which new features are modeled and tested in a simulation environment, is applied intensively in the modern product design. Model-based or virtual design provides a means for achieving machine design improvements with reduced time and costs (Eppinger, Whitney, Smith, & Gebala,, 1994). In the product development process, virtual design is often used for feature or system validation (Tseng, 1998). Virtual design is typically conducted early in the design process when it is less expensive to make changes.

However, virtual design of off-highway machines with operators in the loop has often been limited by the fidelity of the model of human operators. This limitation is particularly an issue when with virtual design is used for validation and assessment. Traditional validation and assessment methods, by way of comparison, utilize physical machine prototypes, human operators, and real-world testing in a controlled environment (Filla, Ericsson, & Palmberg, 2005). While machines have been modeled with a fidelity that enables robust testing, current operator models struggle to capture operator expertise and require time-intensive tuning to each new machine design. These limitations hamper engineers from making solid comparisons in the virtual prototyping stage between different design alternatives, and limits their ability to do virtual design. Given the tightly coupled, non-linear nature of off-road vehicle dynamics, combined with human-in-the-loop control, dynamic simulation of the complete vehicle system must include the operator, environment, and tasks. To advance machine testing, a virtual operator model (VOM) needs to be developed to represent how human operators operate machines. The fidelity of VOMs needs to be increased by using a more human-centered basis for virtual operator modeling, and increasing the fidelity of operations modeling.

A VOM aims to simulate the human operator's perception, decision-making, and actions leading to control inputs for the trenching operation. The VOM is designed to represent the control behaviors of human operators, which is expected to simulate the machine model similar as a human operator does, and enables human-in-the-loop dynamic evaluation in the virtual design stage.

Current VOM efforts have largely been restricted to developing models that mimic known trajectories, usually recorded from actual vehicle operations (Filla, 2005; Elezaby, 2011). This implies that any change to the vehicle design would require a time-intensive process of “re-tuning” the VOM to mimic new vehicle trajectories. This limits their utility in fast-paced iteration in model-based design cycles. Furthermore, the work cycle has been modeled as discrete, sequential series of tasks, as the operator completes one task before moving to the next (Elezaby, 2011).

Human operators are all unique, they have different skill levels, background, and cognitive processes. For instance, novice and expert operators operate the same machine for the same task differently. The expert operators may have different strategies for different situations. Depending on the environments, operators can adjust their control inputs to adapt to different locations, different worksite conditions, and different machine capabilities. They can also learn, what the optimal method is to operate the machine to complete the tasks over time. The behaviors for adaptation and learning can differ for different operators. It is quite a challenge to capture human operators’ behavior.

Human operator decision-making and behaviors are varied and complex. Because of this complexity, it is difficult to develop and validate human operator models. Currently, only a few studies (Filla, 2005; Elezaby, 2011) have documented virtual operator model development and validation. To advance machine testing, the fidelity of VOMs needs to be increased along multiple dimensions, developing a more human-centered basis for virtual operator modeling, representing human operator expertise, representing the ability to adapt to changes in the work site environment and different machines, and developing the ability to learn and develop expertise.

Most approaches are focused on replicating the control trajectory of the vehicle, rather than the operator behavior that generates the control inputs. Human operators generate control inputs based on their perception of the environment and their decision-making processes.

Virtual operator modeling can enable human-in-the-loop dynamic evaluation in the virtual design stage, which results in cost and time reductions compared to the traditional product development (Becker, Salvatore, & Zirpol, 2005). This capability will enable simulation of model-based machine prototypes for performance analysis including fuel efficiency, productivity, and component loading. Virtual operator models enable closed-loop, whole system evaluations of the capability of new design features early in the design process. By enhancing the VOM along the lines of representing human operator expertise and representing the ability to adapt to changes in the work site environment and different machines, the closed loop simulation of VOM-Vehicle systems is advanced with higher fidelity. With representation of human operators' adaptation abilities, it is more confident that VOM drive the machine the way an expert operator would in a productivity test with a real prototype. To test a new vehicle, the best strategy is always unknown, which can be uncovered by human operators using learning capabilities. Similarly for new vehicle design test in simulation, the VOM with a learning capability would uncover the best strategy and may lead to more realistic simulation of human experts.

The excavator trenching operation was selected as the modeling target. Trenching using an excavator is a common operation in the construction environment, which requires multiple tasks within the work cycle. The operator needs to finish a trench with predefined dimensions, location and orientation within a certain time period and then must then deposit the material in a defined area or container. It also requires the operator adapt to environment changes from cycle to cycle.

An overview of the research effort is presented in Figure 1.

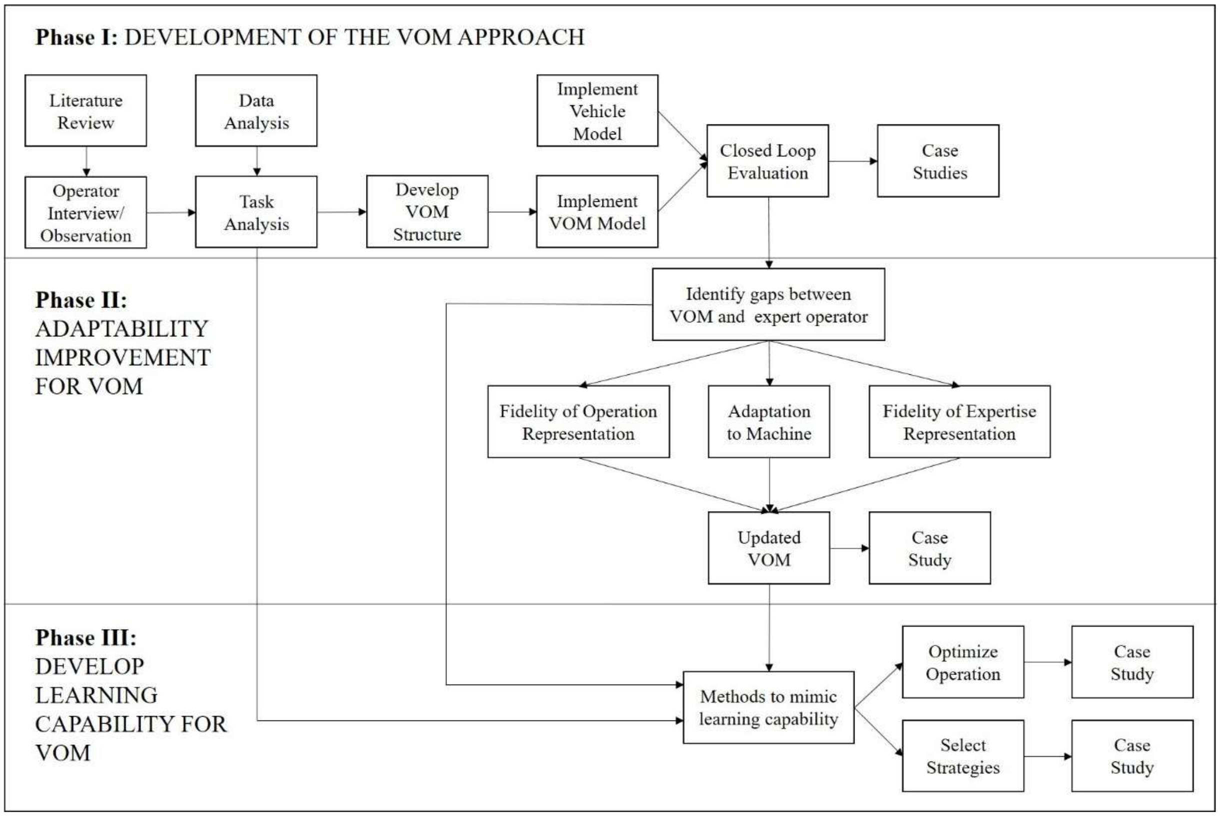


Figure 1. Overall vision of development of the VOM

In Phase I, a human-centered systems process was developed to capture and represent operators' tasks, strategies, cues, and constraints. The process included interviews and observation, and the analysis of machine data acquired from an excavator performing a trenching operation. A virtual operator model architecture was developed and implemented using various techniques to capture the fluid nature of tasks within an operation. The virtual operator model was tested by integrating it into a closed loop simulation with a vehicle model. The model was exercised by conducting tests using different digging strategies, varying vehicle hydraulic pump speeds, different pile locations, and different trench depths.

Phase II advanced the VOM to enable it to adapt to the dynamical changes in the environment, adapt to changes in the geometry of different vehicle models, and better represented expert behavior. As human operators complete a work cycle, they affect changes to the work site environment. Simply, the trench becomes deeper and the pile grows higher with each work cycle until the operation ends with the desired trench depth. To adapt to the changes in the work site after each work cycle, it requires that the VOM adjust control inputs after each work cycle. An environment model was developed to describe the current work site environment conditions at any point during the operation, much like the mental model of a human operator is continuously updated. To adapt to different vehicle models, reference commands needed to be determined by considering the dimensions of the vehicle components. Initialization module was developed and variables were used throughout each modules of the VOM to calculate the reference commands using initialized dimensions of the vehicle components.

Phase III of this work focused on developing the learning capability for a task within the trenching work cycle in order for the VOM to learn the best strategy to complete the task most efficiently. The learning process was developed as an optimization problem by using genetic algorithm to find the best combinations of different control methods.

Virtual operator modeling is advanced through this work by basing the model structure on the human information processing model. This allows a structure on which to improve the fidelity of virtual operator modeling, including: representing full operation more realistically, modelling different strategies of human operators, and modelling the adaptation ability and learning capability of human operators. The fidelity of the simulation of human-machine system is improved, which enables the assessment of machine designs with simulated operation behaviors similar to human operators in the virtual environment. More realistic models enables computer based simulation that can realize a low cost and efficient machine assessment method. With high fidelity of VOM-Vehicle simulation system, model-based design can be applied for more aspects of design process, which can reduce the time for the process of product development and save expenses on physical prototype testing with human operators.

This dissertation is formatted as three journal papers. Chapter II describes the development of the initial VOM architecture based on a human-centered process to understand operators, define domain concepts, and developing a basic VOM based on a human information processing model (Du, Dorneich & Steward, 2016). Chapter III describes the methods developed to increase the adaptability of the VOM to the environment, adaptability to different machines, and representing operators (Du, Dorneich, & Steward, 2018). Chapter IV introduces methods used to realize the learning capabilities in the VOM (Du, Dorneich & Steward, accepted). Chapter V summarizes the overall work, discusses the contribution, and suggests future work.

CHAPTER II: DEVELOPMENT OF THE VOM APPROACH

Material in this chapter appeared as a journal paper:

Du, Y., Dorneich, M.C., & Steward, B.L. (2016). "Virtual Operator Modeling Method for Excavator Trenching," Automation in Construction. Vol. 70, No. , pp. 14-25.

DOI:10.1016/j.autcon.2016.06.013

Introduction

The human operator of off-road vehicles is an integral part of the human-machine system performance. High fidelity machine models are used in simulation to test new vehicle designs. However, the fidelity of human operator models is often a limiting factor in the overall ability to conduct closed-loop simulation testing. This research investigated how machine operator expertise, practices, and decision making can be integrated into an operator model for virtual simulation of closed-loop construction vehicle operation. The goal of the research was to capture the behavior and performance of a human operator and represent the operator in a virtual operator model that simulates authentic human behavior in a well-defined construction machine operation.

Considering the complexity and non-linear nature of off-road vehicle dynamics, and the fact that the operator is intimately enmeshed in the closed-loop control system of the vehicle operation, field testing with human operators is the most common method used to test designs with physical prototypes and human operators in real working environment (Filla, Ericsson, & Palmberg, 2005). Vehicle field testing requires significant cost and time compared to computer-based simulation. Virtual design or model-based design, the process by which new features are modeled and tested in a simulation environment, is typically conducted early in the design process where it is less expensive to make changes. While machines have been modeled with a fidelity that enables robust testing, operator models are still in early stages of development. Methods for closing the simulation loop around operator, vehicle, and environment models need to be investigated.

Human operator decision-making and behaviors are varied and complex. Because of this complexity, it is difficult to develop and validate human operator models. Currently, only a few studies (Filla, 2005; Elezaby, 2011) have documented virtual operator model development and validation. These limitations on virtual operator technology limit design engineers' ability to make reliable comparisons in the virtual prototyping stage between different design alternatives.

Additional challenges exist in the development of virtual operator models. Operator models are typically created by tuning control models to mimic trajectories. Often they are tuned to be specific to a particular vehicle operating under specific conditions. If the vehicle design is changed, or the operating conditions are varied, the model often has to be re-tuned to match the new operating profile. These models focus on trajectories, not on operator perception and decision making processes. Human operators, in contrast, can adapt to changes in the machine or changes in the environment. Standard methods to model operator behavior and ability to adapt have not been established in this domain. Most approaches are focused on the control of the vehicle, rather than the operator behavior that generates the control inputs. Cognitive modeling has been developed as computational representations of internal cognitive processes; however, they are designed to be task-independent (Byrne & Kirlik, 2005), and focus on modeling constructs such as working memory (Baddeley, 1998). These computational cognitive models focus on how human operators interact with the environment and make decisions, but are not designed to produce the control inputs of a human in vehicle operation. In the domain of off-road vehicle operations, the challenge is to summarize complicated cognitive processes in a model that is dynamical in nature, with the goal of creating an input/output model that faithfully represent operator expertise, sophistication, and adaptability.

An automated system can significantly improve consistency of repeated tasks in a stable, controlled environment, which does not have much variation (Bradley, 1998; Wu, 2003). However, when the operating environment or conditions within which an automated system operates changes, higher-level machine intelligence technologies (beyond closed-loop control) must be in place for the autonomous system to adapt to these changes. Developing these types of behavioral responses for autonomous systems is challenging. A robust automation system with perception of external cues and use of internal goals may be able to exhibit adaptive behavior. For this behavior, expert human operator behavior and decision making processes may have great utility. A virtual operator model aims to capture key behaviors of human operators, enabling autonomous system to adapt to external environment changes.

Virtual operator modeling can enable human-in-the-loop dynamic evaluation in the virtual design stage, which results in cost and time reductions compared to the traditional product development (Becker, Salvatore, & Zirpol, 2005). This capability will enable simulation of model-based machine prototypes for performance analysis including fuel efficiency, productivity, and component loading. Virtual operator models enable closed-loop, whole system evaluations of the capability of new design features early in the design process.

The excavator trenching operation was selected as the modeling target. A virtual operator model was developed to simulate the human operator's perception, decision-making, and actions leading to control inputs for the trenching operation. Trenching using an excavator is a common operation in the construction environment, which requires multiple tasks within the work cycle. The operator needs to finish a trench with predefined dimensions, location and orientation within a certain time period and then must then deposit the material in a defined area or container. Operators judge their performance by time and quality of the trench, which means operators seek to finish the trench with maximum efficiency. A human-centered systems process was developed to capture and represent operators' tasks, strategies, cues, and constraints. The process included interviews and observation, and the analysis of machine data acquired from an excavator performing a trenching operation. A virtual operator model architecture was developed and implemented using various techniques to capture the fluid nature of tasks within an operation. The virtual operator model was tested by integrating it into a closed loop simulation with a vehicle model. The model was exercised by conducting tests using different digging strategies, varying vehicle hydraulic pump speeds, different pile locations, and different trench depths.

Related Work

Given the tightly coupled, non-linear nature of the sub-system dynamics in off-road vehicles, combined with a strong human-in-the-loop involvement of operators, dynamic simulation of the complete vehicle system must include the operator, environment, and working tasks (Filla, Ericsson, & Palmberg, 2005). Human factors methods can provide deeper insights into the behavior of human operators, including decision-making, cues that trigger actions, and strategies that help adapt to changing conditions. This information could be incorporated into a virtual operator model. Existing operator modeling approaches for off-highway vehicles fit into two categories: 1) task-oriented operations in which the operator controls the machine through a repeated sequence of tasks to accomplish high-level goals (*e.g.*, Filla, 2005), and 2) reference-oriented operations in which the operator is guiding the machinery along a particular path to accomplish some types of operation (*e.g.,* Zhang et al., 2003). Beyond the virtual operator literature, other relevant research exists in the area of mobile equipment automation, where a typical approach was to model operator behavior for a particular operation as the strategy for automating that operation (Bradley & Seward, 1998; Wu, 2003; Enes, 2010). A virtual operator approach could potentially be applied as the control logic for adaptive systems, where the automation has the authority and ability to change its mode of operation to best support joint human-automation performance (Feigh, Dorneich, & Hayes, 2012).

Capturing and Modeling Human Expertise

Expert human operators exhibit several characteristics: humans can adapt quickly to context using prior experience and training; humans have the ability to integrate contextual cues and strategies; and expert operators can often outperform automated functions. As human operators gain experience, their operations progress from a primarily knowledge-based behavior, to rule-based behavior, and finally to skill-based behavior (Rasmussen, 1983). Knowledge-based behavior depends on explicitly formulated goals and plans. With more practice, operators become rule-based, where sequences of action become rules to follow. Eventually, the expert exhibits skill-based behavior, where much of the action takes place without conscious control (Rasmussen, 1983). These human characteristics are quite different from those of automated machine systems.

Human factors methods can be used to gather, organize, and represent information on how expert humans perform operations. The goal of the process is to understand as much as possible about users, their task, and their context in order to produce a stable set of requirements to guide design. The requirements arise from understanding users' needs and should be justified and related to data collected from and about users. Contextual inquiry (Holtzblatt, 2003) and task analysis (Stanton & Walker, 2005) methods, including interviews, questionnaires, observation, and the study of artifacts inform the process. Task and user analysis can be used to develop a set of representative tasks that cover the functionality, manual and mental workload, durations, complexity, equipment and environmental requirements of the system (Kirwan & Ainsworth, 1992).

Operator Modeling Approaches

A task-oriented operation consists of a sequence of tasks, which are repeated and simulated to achieve the operational goals. Operator models developed for task-oriented operations, specifically wheel loader loading cycles, have employed finite state machines to represent the work cycle structure as a series of tasks using finite states (Filla, 2005; Elezaby, 2011). The operator models generated appropriate control inputs for machine models. Validation was limited to the comparison of simulated paths with experimental paths for different vehicle components (Filla, 2005).

A reference-oriented operation is one in which an operator guides a machine along a predefined reference path to achieve operational goals. In the context of wheel loader steering control, Norris (2001) developed a design framework for modeling human behavior, with the human considered to be an element in a control system. An algorithm was developed which enabled control system adaptation to human operator steering control behaviors through the use of a valve modulation curve representing human decision making (Norris and Zhang, 2003). Fuzzy controllers generated machine control inputs. Validation was based on comparison of the simulated vehicle trajectories to reference paths.

Autonomous Control

The design of an autonomous vehicle control systems requires the development of a controller, which can be thought of as a type of virtual operator model. Control modules were developed based on operation strategies and the behaviors of human operators, which were able to choose an appropriate control strategy in response to obstacles such as rocks. For example, different strategies were determined for the excavator trenching operation: the bucket was forced into the soil and drug across the surface for dense soil, while the bucket was inserted into the material and rotated for the loose soil (Bradley, 1998). In another study, a control module was developed using a combination of neural networks and fuzzy logic to adapt to different materials for a wheel loader loading operation (Wu, 2003).

An autonomous system is one with the ability to perceive information or cues from the environment and machine and generate the appropriate control inputs to adapt to the environment with varying conditions. To achieve autonomous or robotic operations of off-highway machine systems, researchers have recognized that technology beyond closed-loop control is required. In fact, a structure for defining behaviors is required to carry-out field operations in the context of situational uncertainty. Fountas et al. (2007) promoted a structure defining human-like behaviors required for agricultural field robotic applications. These behaviors can be broadly classified into planning and supervision. Planning includes determining the best course of action to achieve a particular operational goal. Supervision involves monitoring the machine and work environment so that planned actions are modified as needed based on new information. This behavioral approach can be extended from agriculture to construction applications and be embedded in a multi-layered design framework to plan an autonomous system (Han et al., 2015). Bradley et al. (1998) developed an autonomous robotic excavator to realize high quality autonomous, rectangular trenching. The control system was designed to imitate the actions and strategies of a human operators working with obstacles.

Adjustable Human-Centered Autonomous Technology

A well-developed virtual operator model can be used to drive automation that can adapt to different situations. Adjustable automation can allow the human to initiate the level or function of automation to ensure that the system is behaving appropriately given the current situation (Dorais, Bonasso, Kortenkamp, Pell, & Schreckenghost, 1999). Adaptive automation is similar, in which the automation can change its own behavior, based on its understanding of the situation (Feigh, Dorneich, & Hayes, 2012). A robust virtual operator model can update strategies and change the behavior or automation. Types of adaptation include dynamic function allocation for the sharing and trading of functions between the automation and the human operator to increase efficiency. Adaptive automation has different levels or automation, and dynamically adjusts the authority between human and the control system (Inagaki, 2003). Issues in adaptive systems include a loss of situation awareness, automation visibility, authority and responsibility, trust, coordination demands, and workload (Goodrich, Olsen, Crandall, & Palmer, 2001; Inagaki, 2003; Mathan, Dorneich, & Whitlow, 2005; Feigh et al., 2012). If the virtual operator model becomes part of the automation decision logic, it has the advantage of behaving much like a human operator would (utilizing the same cues and strategies), increasing the understandability of the automation logic (automation visibility), and perhaps increasing the ease of coordination between the automation and the human.

Materials and Methods

Excavator trenching was selected as the target operation to be modeled, and a virtual operator model was developed to represent excavator operators' decision making processes and behaviors. Operator interviews and task analysis were conducted to learn the behavior and decision-making processes of operators and derive operator model requirements. The virtual operator model was formulated to include perception, decision making, and action modules to produce the control inputs for a vehicle simulation.

Operator Interviews and Data Collection

An interview protocol was designed to acquire information about operators' operating experience, behavior, strategies, and possible problems during operation (Du, Dorneich, & Steward, 2014). The interview was structured as a set list of questions that first queried operators about their background (experience, types of operations, equipment) and then asked detailed questions about what they do before, during, and after operations. All the questions were treated as open-ended questions in the interview; participants were encouraged to expand their answers and knowledge freely. Example questions include “What kind of information do you want to know before an operation?”, “Can you describe the tasks/steps in the operation, in terms of procedures, tasks, and goals?”, and “How do you know when you are performing well?” The interviews were documented with audio recording and written notes. Three participants with different backgrounds and skill levels participated in the interviews. Participants had experience with wide range of different machines. Interview questions for the trenching operation were not specific to a particular machine type. Videos, which were recorded while the participant operated the machine the participant, were reviewed with the participant using a think-aloud technique (Lewis, 1982; Ericsson & Simon, 1993) to provide verbal identification of tasks, needs, goals, strategies, and behavior. Both descriptive data and quantitative data were collected. A combination of knowledge-based and entity relationship-based analysis was conducted for accurate task analysis (Dix, Finlay, Abowd, & Beale, 2004).

Machine data were collected during an excavator trenching operation, which were used to analyze the operator’s behavior and relate it to vehicle operation. To acquire machine operation data, the excavator was equipped with video cameras inside the cab and outside the cab, which captured both video and audio records of the operations. Sensors mounted on the machine were used to acquire operator inputs at joysticks for commanding, boom, arm, bucket and swing motion, as well as boom, arm, and bucket cylinder extension lengths and relative speed and direction of excavator swing motion. The data collected from operator interview and machine operation were used to understand the operators' operation behavior and strategies. These behavior and strategies were used by the virtual operator model to drive vehicle machine.

Virtual Operator Model Architecture

A closed loop operator-vehicle simulation model was developed consisting of dynamic operator and vehicle models in the Simulink platform (ver. 2015a, The Mathworks, Natick, MA). Both models were developed as sub-system modules with a well-defined interface facilitating interchange of vehicle models, so that different combinations of operator and vehicle models could be easily exercised. The operator model has four elements: a vehicle kinematic model, a human perception model, a human decision-making model, and a human action model (see Figure 2). The virtual operator model generates the control inputs that a human operator would provide to control a physical machine. The inputs to the human perception model of the virtual operator model are the environmental conditions and the human-observable states of the machine from the kinematic model. In the development stage of a virtual operator model, the kinematics model can be bypassed, and all observable and non-observable vehicle states can be passed to the perception model. However, the use of the kinematics model enables the perception model to operate only on human observable states, which allows the model to depend only on the cues that a human would use to control a vehicle. The human decision-making model was developed through operator interviews conducted to understand the operation tasks, cue, strategies, and behaviors of skilled operators (Du, Dorneich, & Steward, 2014).

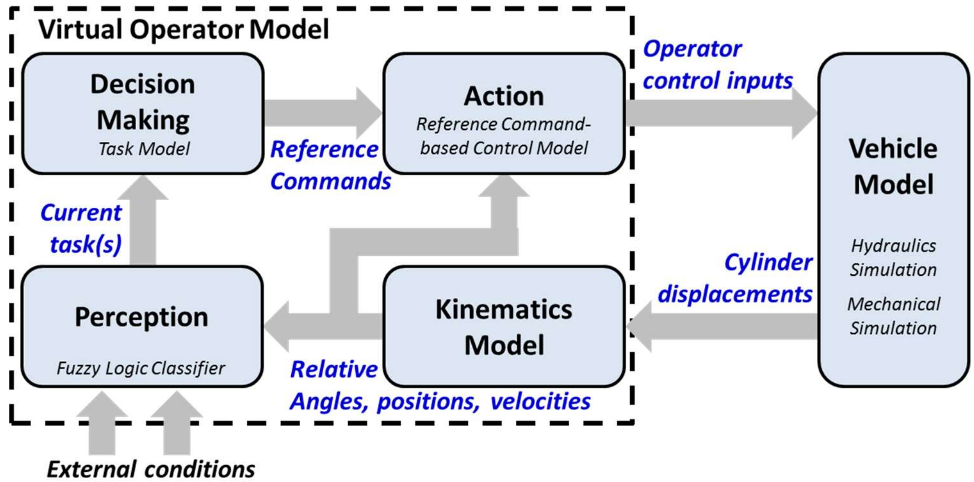


Figure 2. Closed loop simulation of a virtual operator and a vehicle.

Vehicle Model

The vehicle model represented the dynamic characteristics of a representative excavator and included a dynamic model of the hydraulic system and a multi-body dynamic model of the bucket-arm-boom mechanism along with the swing degree of freedom. The vehicle model accepts as control inputs the actuation signals to the hydraulic valves from the virtual operator model, and the simulation of the hydraulic and mechanical systems resulting in cylinder displacements and swing angle as outputs. The vehicle model was purposely developed to be modular and independent of the virtual operator model. As such, it can be replaced with higher fidelity vehicle models. The hydraulic system, modeled in SimHydraulics (ver. 2015a, Mathworks, Natick, MA), was a closed center system with a pressure-compensated pump, pressure-compensated proportional directional control valves for the work function circuits controlling the boom, arm, and bucket cylinders and the hydraulic motor controlling the swing motion.

The hydraulic system model was not intended to model any particular system, but to provide a reasonable response of a hydraulic system on a typical excavator. SimMechanics was used to model the multi-body dynamics of the excavator's boom/arm/bucket mechanism along with the swing degree of freedom. Geometry was based on machine dimensions of a test machine, and mechanism component masses and moments of inertia were estimated using machine component geometry.

Kinematic Model

A human operator does not observe the hydraulic cylinder displacements for cues during operations; rather he or she observes machine dynamic variables such as the relative height of the bucket off the ground or the swing angle of the boom. The kinematic model was intended to map vehicle state information that is commonly measured with sensors into signals that human operators could perceive. Specifically, for this case, the kinematic model related cylinder extension lengths to the location and orientation of machine elements, which were relative to the trench location and were used as operator cues. For example, cues such as bucket height and swing angle were used for decision-making during operation. From a simulation perspective, all of these vehicle states should be available from the vehicle model. However, the kinematic model can simplify the vehicle/operator model interface by reducing the number of signals in the interface. This approach has the advantage of encapsulating the operator model and the vehicle model as well-defined software components. It also enabled the operator model to be driven with experimental data from vehicle tests for model troubleshooting and validation. A simplified model with joints and rigid bodies was used to represent the boom, arm and bucket movement (Figure 3). The kinematic model was derived mathematically using kinematic equations and was coded in MATLAB script. This model described the position of each critical point identified in Figure 3, relative to the coordinate system with origin O.

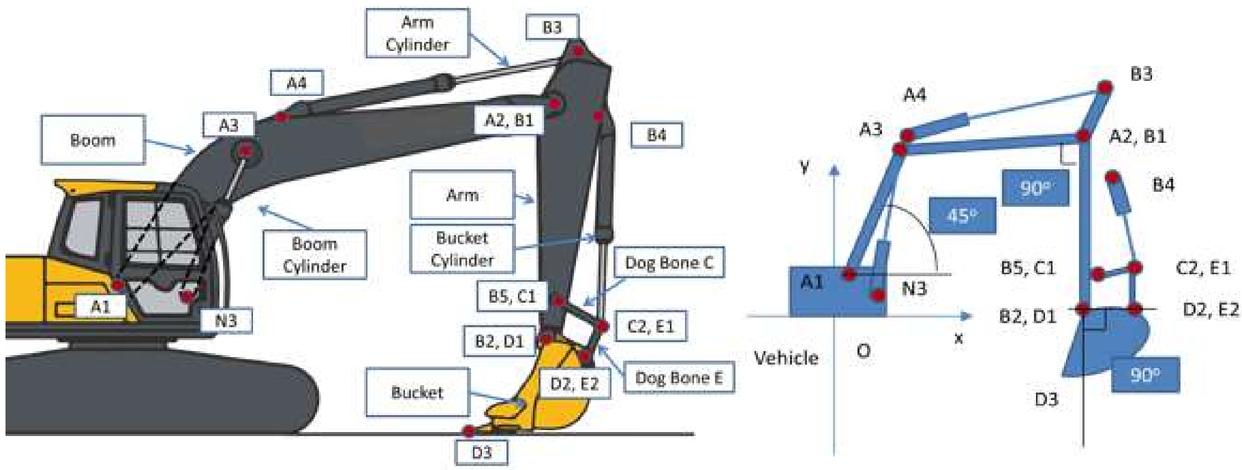


Figure 3. Typical excavator mechanism with labeled rigid bodies and joint nomenclature (left) and simplified representation of the excavator mechanism with vehicle coordinate system defined and joint locations labeled (right), all used in the development of the kinematic model of the Boom, Arm, and Bucket.

Human Perception Model

Based on operator interviews, operator behavior was summarized in terms of what information were perceived, how the information was used for operating the machine, and what control inputs were applied. The excavator operation can be broken down into a series of tasks. Human operators usually perceive visual cues or information about the physical position of machine components and use these perceived cues to make decisions. For excavator operators, information like bucket height, swing rotation angle, and bucket extended length can be directly perceived, and were used to help the operator to determine the current task. For example, a human operator knows that he or she can start to swing the bucket towards the trench only when the bucket is filled, lifted out of the trench, and above the ground. The human perception model uses the kinematic information from the vehicle model, and a predefined fuzzy variable membership function to determine the bucket position, which can provide information similar to that which human operators can perceive. In this way, the human perception model simulated the human operator perception process in determining the current task in the operation work cycle.

To model human-like decision-making process, numerical signals from the kinematic model were fuzzified into fuzzy classes representing linguistic statements about the relative location and orientation of the bucket at a human perception level. The structure and design of the fuzzy classes were derived from the operator's mental model of tasks and cues. Through operator interviews, five signals were identified as human perceivable cues used to control the machine: bucket height, swing angle, bucket extension distance (between the bucket and operator cab), bucket rotation, and bucket vertical velocity. The excavator bucket height relative to the ground was mapped to fuzzy membership value in three fuzzy classes: above soil, near surface, or below surface. Based on these fuzzy classes, a fuzzy classification system was developed based on operator interviews and task/data analysis. It was determined that the expert operators are able to overlap the beginning and ends of tasks. Fuzzy logic allows multiple states to be active simultaneously, and thus can be used to represent operations that include task overlaps. The current version of the model implements five finite-states of the trenching operation without overlaps. The next phase will focus on developing classifiers to detect the start and end of each task, which can then be integrated to determine the overlaps between tasks. For example, if both the end of the swing to the dump pile task and the start of the dumping task were detected, then the overlap between these two tasks can be determined. Five continuous variables were used to represent the current operator perceivable machine state, and were fuzzified into a degree of membership in the classes associated with those variables (Table 1).

Table 1. Conditions that represent states of the machine

Continuous Variable	Fuzzy Classes
BucketHeight	AboveSurface, NearSurface, BelowSurface
SwingAngle	SwingLeft, NearTrench, SwingRight
ExtensionDistance	Retracted, MidRange, Extended
BucketRotation	Uncurled, Curled
BucketVerticalVelocity	UpFast, UpSlow, DownFast, DownSlow
BucketFillTransition	BucketFill, Swing2Dig

Fuzzy rules were derived from operator interviews and data analysis, which uses similar information (and a similar vocabulary) that human operator uses to determine their actions. For example, a human operator uses bucket height, swing angle and bucket rotation to determine when and where to dump material from the bucket. In fuzzy classifiers, similar information was used to mimic the human operator's decision making. Fuzzy classifiers used these rules to identify the transition between the five tasks of the work cycle (Bucket Filling, Bucket Lifting, Swing to Dump, Dumping, and Swing to Trench - see next section) based on common cues and triggers that operators used. Five individual classifiers were developed, one for each transition. The outputs from the fuzzy classifiers represented the degree of membership that the current machine state is associated with the five tasks. By successfully identifying the transitions between tasks, the correct prediction of next task onset can be made, which can lead to appropriate reference commands being generated. For example, one fuzzy classifier has a set of rules for the transition between the Swing to Trench task and the Bucket Filling task (Table 2).

Table 2. Rules in the Fuzzy Classifier to detect the transition between Swing to Trench and Bucket Filling.

1. If (BucketHeight is BelowSurface) and (SwingAngle is NearTrench) and (ExtensionDistance is Extended) then (BucketFillTransition is BucketFill)	3. If (BucketHeight is BelowSurface) and (SwingAngle is NearTrench) and (ExtensionDistance is Retracted) then (BucketFillTransition is BucketFill)
2. If (BucketHeight is BelowSurface) and (SwingAngle is NearTrench) and (ExtensionDistance is MidRange) then (BucketFillTransition is BucketFill)	4. If (BucketHeight is NearSurface) and (SwingAngle is NearTrench) and (ExtensionDistance is Extended) then (BucketFillTransition is Swing2Dig)
	5. If (BucketHeight is AboveSurface) then (BucketFillTransition is Swing2Dig)

The fuzzy classifier was tested in two ways. First, vehicle data from the machine test data set was used as input to the classifiers and the task transitions were compared with transitions that were manually determined based on the observation. To assess the ability of the classifiers to detect task transitions, the transitions were classified and counted into the number of transitions detected (both prior to and after actual transition) and transitions not detected.

While the fuzzy classifier was built to detect the transition between tasks, the membership rules can also be used to detect the current task. Thus, the second method of testing the classifier was to determine how well, on a moment to moment basis, it detected the correct task given the machine data (Ground Truth). The results were represented in a confusion matrix to show the accuracy of the detection results by counting the number of hits, correct rejections, misses, and false positives for all five tasks. The overall accuracy was calculated by the number of hits and correct rejections over the total number of points.

Human Decision-Making Model

The human decision making model consisted of a finite state machine modeling tasks as states and included rules for task transitions. Based on the current task, the reference commands for the actuators are provided to the human action model.

Task analysis identified five tasks: Bucket Filling, Bucket Lifting, Swing to Dump, Dumping, and Swing to Trench, that make up the trenching operation work cycle. A state machine was developed to model this sequence of tasks (Figure 4). The state machine was coded in MATLAB script to provide the correct sequence and status of each task based on the task transition detected from the fuzzy classifiers. By combining of all identified transitions within the trenching operation, the sequence of tasks and current state of the operation can be represented. When a transition between tasks was detected, the model generated reference commands for the human action model.

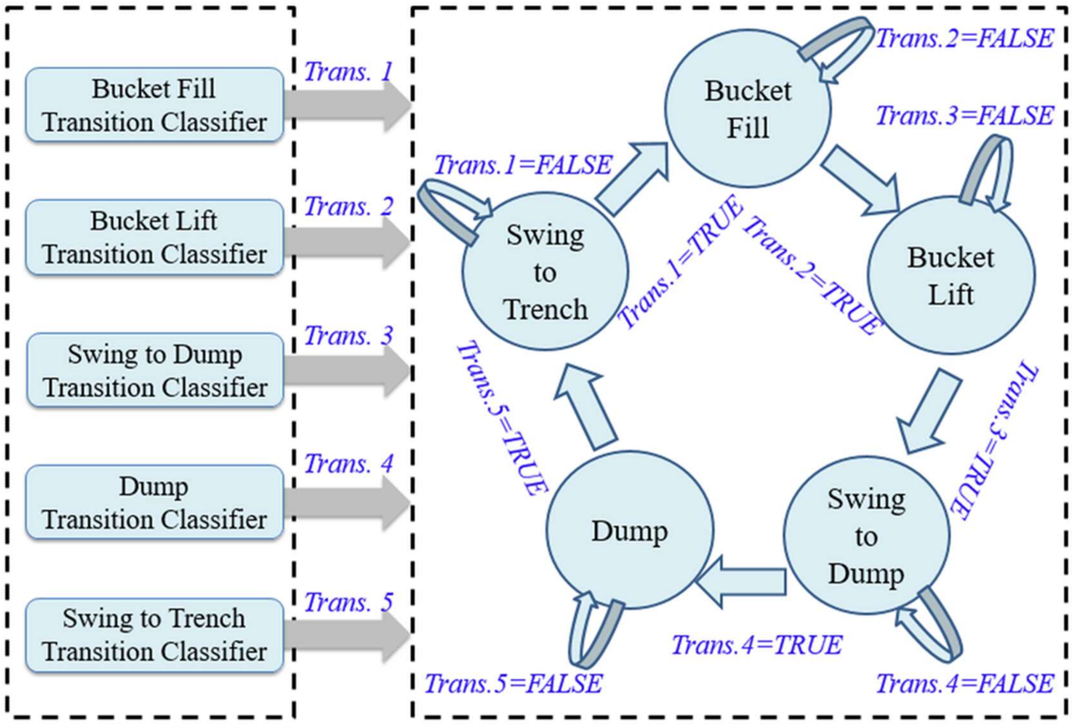


Figure 4. Task Model with Transition Conditions.

To test the task model in isolation, the machine data were provided to the fuzzy classifiers, which provided the transition detection results for the task model. The output of the classifiers was compared to the manually identified task start and end times of the machine data.

Based on operator interviews, different strategies are employed for certain situations. To test the virtual operator model's ability to implement different strategies, two strategies for the Bucket Fill task were implemented and compared. The first strategy was "rotate and fill," commonly used when trenching softer materials like dirt and loose gravel. In this strategy, the operator slowly curls the bucket while simultaneously moving the bucket from the end of the trench towards the cab. A second strategy was "scrape and scoop," which is used to fill the bucket with hard materials such as rocks. In this strategy, operators keep the bucket at a constant angle relative to the ground as they scrape the surface of the trench to push material into the bucket, and then rotate the bucket at the end of the task to scoop the material firmly into the bucket.

Closed loop simulations were conducted to produce trenching operation work cycle trajectories. The trench was modeled at a zero-degree swing angle relative to the axis extending from the front of the vehicle operator cab. The pile was modeled as being at a 29 degrees clockwise swing angle from the trench looking down on the excavator. The digging surface was located approximately two meters below the ground surface, and the trench was six meters long.

As a test of the virtual operator model, these two strategies were implemented with specific reference commands for each strategy (see Table 3). Bucket teeth trajectory, machine responses, and state sequences were used to compare the resulting trajectory of the strategies. The reference commands are expressed as angles for the boom, arm, bucket, and swing (see Figure 5).

Table 3. Angle reference commands in degrees for two different Bucket Filling strategies.

	Rotate and Fill Strategy				**Scrape and Scoop Strategy**			
Tasks	**Boom Angle**	**Arm Angle**	**Bucket Angle**	**Swing Angle**	**Boom Angle**	**Arm Angle**	**Bucket Angle**	**Swing Angle**
Bucket Fill	-40^{o}	-23^{o}	52^{o}	0^{o}	-40^{o}	-23^{o}	0^{o} /52^{o}	0^{o}
Bucket Lift	-77^{o}	17^{o}	69^{o}	0^{o}	-77^{o}	17^{o}	69^{o}	0^{o}
Swing to Dump	-77^{o}	26^{o}	57^{o}	29^{o}	-77^{o}	26^{o}	57^{o}	29^{o}
Dumping	-77^{o}	17^{o}	-52^{o}	29^{o}	-77^{o}	17^{o}	-52^{o}	29^{o}
Swing to Trench	-34^{o}	29^{o}	-52^{o}	0^{o}	-34^{o}	29^{o}	-52^{o}	0^{o}

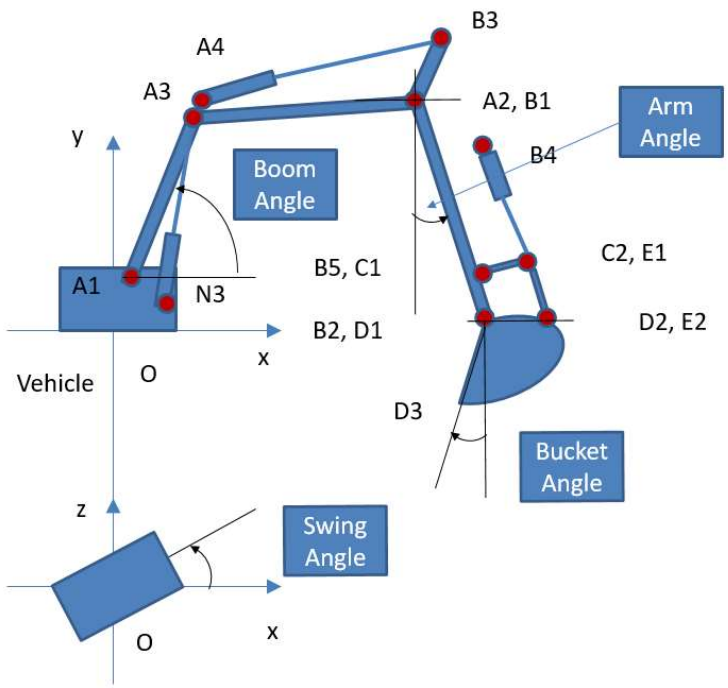


Figure 5. Reference Angle Representation. The Boom Angle is in negative direction, Arm Angle is in positive direction, Bucket Angle is in positive direction, and Swing Angle is in positive direction.

Human Action Model

The human action model was developed to generate appropriate control inputs similar to those a human operator would provide to the vehicle controls. The inputs were control signals to the proportional valves associated with the four actuators. Reference commands from the human decision model were provided to this model along with the feedback signals from the vehicle model. The error signals were input to PID controllers, one for each actuator.

Currently the reference commands are constant values (Table 4), but they can be made more sophisticated, such as commanded trajectory or changes within a task as functions of time or other machine states. By triggering task transitions, appropriate reference commands of the next task will be selected.

Table 4. Control Reference Commands for each of the five tasks in the work cycle.

Tasks	Reference Commands
Bucket Filling	Lower bucket to certain position (Boom Angle = -40°) Pull bucket along trench with 45° (Arm Angle vs. Boom Angle) Curl bucket (Bucket Angle= 52°)
Bucket Lifting	Lift bucket (Boom Angle = -77°) Maintain bucket curl angle (Bucket Angle = 69°)
Swing to Dump	Swing Angle = 29° Trajectory to Pile (Swing angle VS. Boom Angle)
Dumping	Bucket Angle = -52°
Swing to Trench	Swing Angle = 0 Trajectory to Pile (Swing angle VS. Boom Angle)
Swing to Trench	Swing Angle = 0 Trajectory to Pile (Swing angle VS. Boom Angle)

Test Cases of the Closed-Loop Simulation System

Validation of the closed loop combination of the virtual operator model and the vehicle model involved testing whether the virtual operator model behaves as a human operator would under different conditions. Four test cases were developed and are intended to show that the virtual model operator produces appropriate behavior under changing machine and work site conditions. The first test case utilized different digging strategies: “rotate and fill" and “scrape and scoop.” The second test case used different pump speeds; the rotational speed of the hydraulic pump was varied between 2,771-3,917 revolutions per minute to demonstrate the effect of additional hydraulic flow on the work cycle time and the virtual operator model. The third test case used different pile locations, resulting in different swing angles of 28.6°, 57.3°, and 85.9°. The fourth test case tested different trench depths of 1.6 m, 2.2 m, and 2.9 m.

Results

Operator Interviews and Data Collection

Three operators participated in the interviews. All the participants were male, and averaged 14 years of experience (range: 8 to 20). They all had experience with a wide range of different equipment (*e.g.,* excavators, skid-steer loaders, backhoes, scrapers, tractors, wheel loaders, dozers, roller compactors, and pavers) and brands (*e.g.,* John Deere, Caterpillar, CASE, Bobcat, Kobelco, Doosan, Volvo, Hyundai, JCV, Hitachi). The participants had differing formal training, from formal operator school to on-the-job training. Their work experience ranged from small-to-medium sized jobs in construction to experience operating agricultural equipment. One participant had been an owner-operator for four years; however, all worked as an operator in a firm. The time spent in a vehicle for one stretch during operations varied from five minutes to 16 hours, with a typical duration of two to three hours.

Task analysis based on the interviews and observations led to the definition of a task model (Figure 6) consisting of the sequence and timing of the tasks and sub-tasks in the trenching operation work cycle. The timing of the start and end of each task was estimated through review and analysis of trenching operation video acquired with one of the participants operating the excavator. The timing data was not used in the model; rather it provided a qualitative benchmark upon which to judge the work cycle timing of the virtual operator model outputs. An important observation from both interviews and video analysis was that of task overlap. Task overlap was a consistent theme among all participants – one participant said that the more expert the operator, the more he or she can overlap tasks to increase efficiency and reduce cycle time. While the video analysis of timing is a qualitative estimation of the overlap of tasks, vehicle data analysis was used to get more precise estimates of task timing, which represent the average cycle time for each of the task based on the video analysis. Ten work cycles of the test data were analyzed. The average work cycle time was 17.7 seconds with a standard deviation of 2.9 seconds. The standard deviation is large because the work cycle time is changing as the trench becomes deeper and the pile becomes larger. The interviews, observations, and analysis of the test data were all done to characterize the work cycle, tasks, strategies, and cues. The results from the operator interview provide knowledge about how humans operate machine and their strategies. This understanding was used to develop the operator model. Here we are trying to realize human operators' behavior to drive the vehicle model instead of parameterizing the mean value of the work cycle's time length into the model. Thus, the work cycle time on the figure was not used in the model. Later the simulation results can be compared to this mean value to see, if the work cycle time lies in a reasonable range.

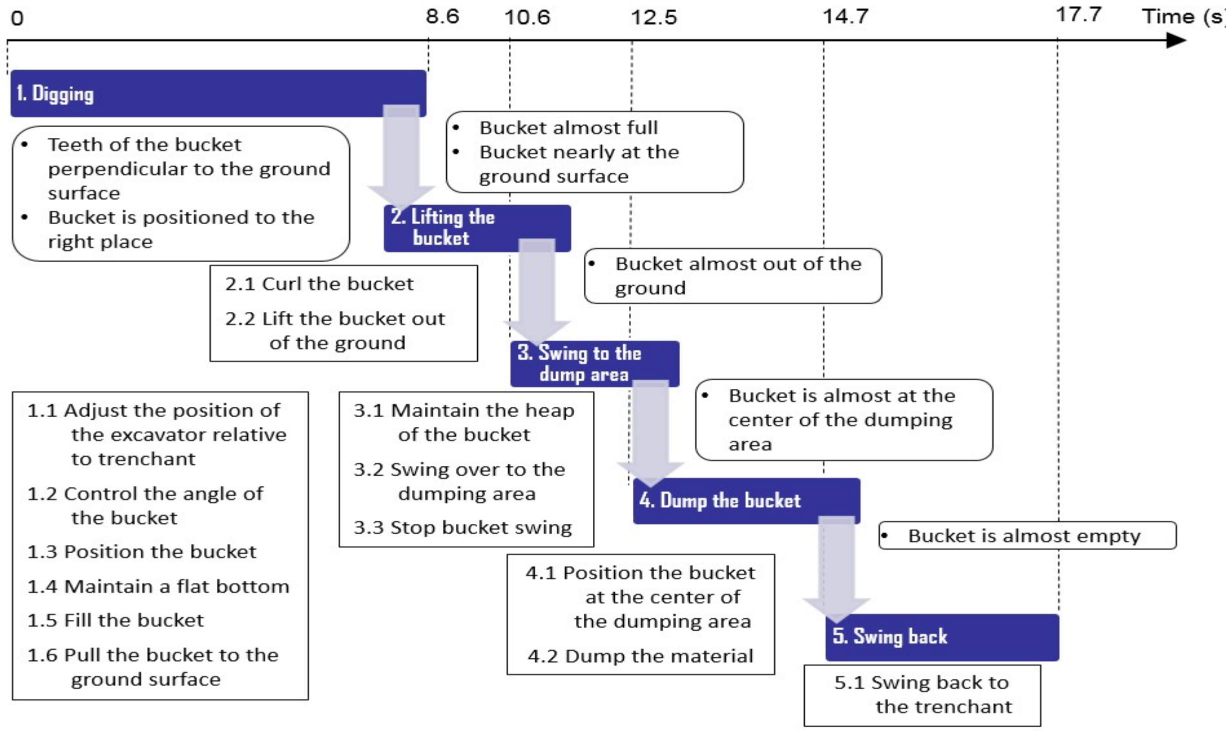


Figure 6. The task model for the excavator trenching operation consists of five tasks and associated sub-tasks associate with the work cycle along with timing and task overlap.

In addition to the task model, several observations resulted from the operator interviews and analysis of the machine data for the excavator trenching operation. Firstly, given the repeating work cycle made up of sequential tasks, a task-oriented modeling approach was chosen as the basis of the virtual operator model for an excavator performing a trenching operation, as compared with reference-oriented operations, which can also occur in construction operations, but are more typical in agricultural operations. Secondly, human operators will not necessarily observe the same physical phenomena or dynamic variables that are typically measured on machines or available from simulation. Human operators cue off relative locations of the bucket, for instance, rather than cylinder displacements. In addition, human operators are cueing off of multiple phenomena such as the position of the bucket relative to the trench sidewalls and bottom or height of the receiver. Also, when removing material, they are observing the velocity of the bucket and the perceived force that is required to remove material. For example, during operator interviews, one participant indicated that they used visual cues during dumping to detect the relative cohesion of the material. These cues were used to choose a proper bucket filling strategy. When the bucket is under the vehicle, the operator cannot see the bucket and uses the arm speed to judge the progress of the bucket filling task. While many of the cues are related to the vehicle, environmental cues are important as well, such as the soil type, working conditions, and locations of the trench and pile on the worksite. The implication is that the reference commands for driving operator commands should be derived from these multiple cues and not just a trajectory to be tracked, as is done currently in some state-of-the-art operator models. To check normality of data, Shapiro-Wilk test was employed. Bartlett's test was used to test the homogeneity of variance. Measured data including ratings of emotional states and usability, TLX survey, target identification correct score, and reaction time were analyzed with ANOVA tests. The results are reported as highly significant for a significance level alpha $<.001$, significant for alpha $<.05$, and marginally significant for alpha $<.10$. Additionally, the EDA signal data was normalized and averaged to create profiles of signals in different conditions.

Task Difficulty Human Perception Model

Transition Classifier

Figure 7 presents a comparison of the outputs of the classifier to the machine data, for the transition between the Swing to Trench task and the Bucket Filling task. The classification results were represented with membership degree from fuzzy classifiers.

The Bucket Filling task was detected for 6 work cycles within 120 seconds. Durations of the task were varied for different work cycles. By comparing the traces, correct detection is illustrated when the green line starts to rise slightly ahead of blue line, since the goal of the classifier is to predict a transition between tasks. If the green line rises later than the blue line, the transition is detected late.

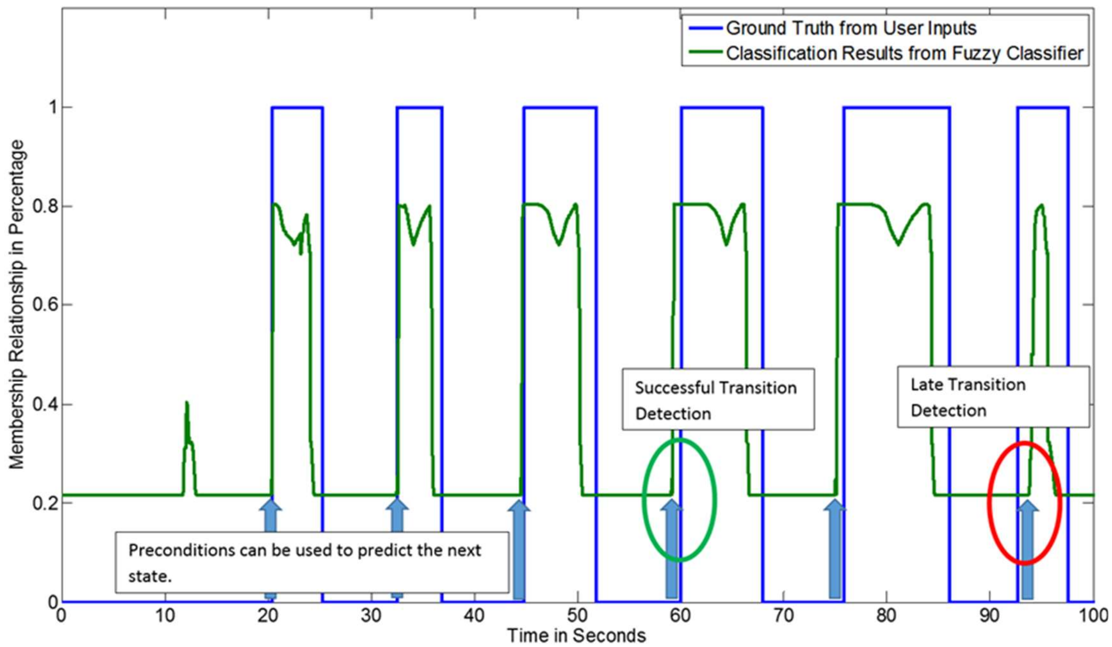


Figure 7. Transition Detection Results between Swing back to Trench and Bucket Filling.

To assess the ability of the classifiers to detect task transitions, the transitions were classified and counted into the number of transitions detected (both prior to and after actual transition) and transitions not detected (Table 5). The classifiers, on average, were able to detect 99% of the transitions. Additionally, the classifiers were able to correctly predict 75% of the transitions before they occurred, with the remaining 24% of detections being detected after they occurred in the test data.

Table 5. Transition Detection Results

Task	Transition Detection Prior	Transition Detection After	Transitions Not Detected	% Transitions detected
Bucket Fill	33	7	0	100%
Bucket Lift	29	11	0	100%
Swing To Dump	22	16	2	95%
Dumping	34	6	0	100%
Swing To Trench	32	8	0	100%

State Classifier

The classifiers were tested by comparing their output (when the recorded machine data was input) to the manually determined correct classification (ground truth). The overall accuracy of the state classifiers to correctly classify each task on a moment-to-moment basis was 90.9% (Table 6). The results were represented in a confusion matrix to show the accuracy of the detection results by counting the number of hits, correct rejections, misses, and false positives for all five tasks.

Table 6. Confusion matrix for each State Classifier

State Classifier	Correct Classification		Incorrect Classification		Accuracy (%)
	Hit (%)	Correct Reject (%)	Miss (%)	False Positive (%)	
Bucket Fill	40.1	50.7	6.1	3.1	90.8
Bucket Lift	17.9	72.7	1.4	8.0	90.6
Swing To Dump	11.2	79.1	3.0	5.7	90.3
Dumping	10.3	84.1	3.5	2.1	94.4
Swing To Trench	11.8	72.3	11.7	4.2	84.1

States Decision Making Model

The decision making model determined the correct tasks, and the transitions between tasks. Figure 8 visualizes the sequence of the tasks with information about when each task starts, which can be considered as a state sequence model. An accurate task sequence is important for the timing of the control signal generation. Machine data were classified using fuzzy classifiers to provide transition detection results. On the x-axis, the transition start time can be read. The colored lines on Figure 8 were the task sequence of the experimental data with its timing information, which identified the start time as well as the end time for each of the task. By combining the actual task sequence with the transition detection results, the comparison of the start times for the detected results and actual sequence could be illustrated. Successful transition detection happened, when black arrow started before the start of the next task. The task model focused on generating the state sequence based on the transition detection results, which may indicate late transition. For the decision making model of virtual operator model, a state sequence is based on correct transition detection.

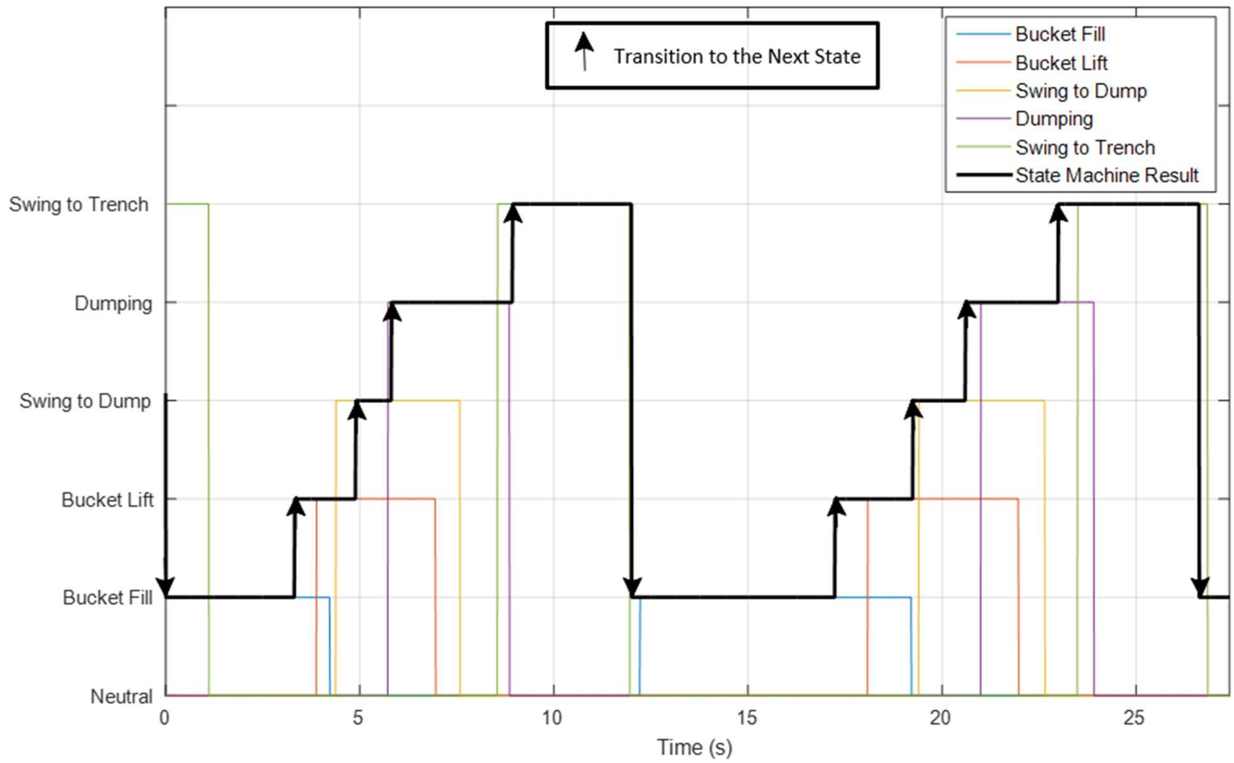


Figure 8. State sequence derived from fuzzy transition classifiers represents transition time to each task.

Closed Loop Simulation Results under Different Conditions

When the virtual operator model was placed in a closed-loop simulation, providing inputs to the vehicle model, it produced operator behavior that was consistent with human operator behavior over the four different test cases. Each test case represented different operator strategies, machine parameters, or work site conditions.

Simulated Operator Digging Strategies

The closed loop simulation was able to successfully simulate two different digging strategies (see Figure 9). The bucket teeth trajectory in three dimensions is shown for one complete work cycle, and in which the five tasks are labeled. The vectors on the graphic represent the orientation of the bucket teeth at certain positions. The rectangular dotted-line box represents the trench.

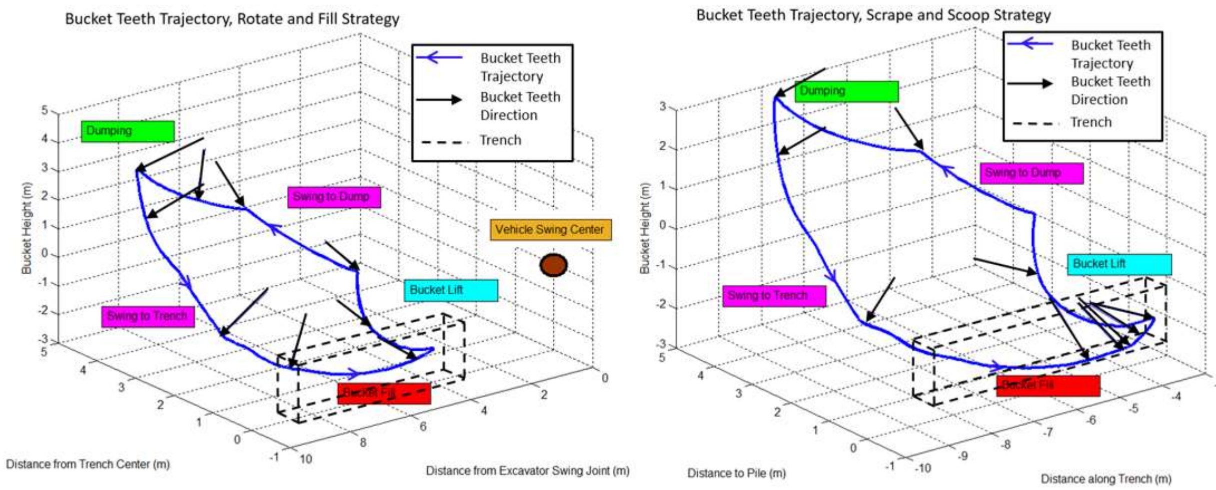


Figure 9. Bucket Teeth Trajectory (blue line) comparison for the rotate and fill and scrape and scoop strategy. The arrows represent Bucket Teeth Orientation. The dotted line rectangular box represents the trench

The vehicle model responses to the inputs of virtual operator model are represented by Bucket Height, Bucket Angle, Swing Angle, and Extension Distance (distance between bucket teeth and cab) (Figure 10). The colored bar on top of the chart represents the task sequence of the operation. The Bucket Fill (BF) task started when bucket height was at the bottom of the trench, bucket angle was at its minimum, swing angle was zero, and extension distance was at maximum. As the bucket was filled and moved closer to the vehicle, the start of the Bucket Lift (BL) task approached. At the transition, the bucket height was under the ground surface, bucket angle was curled around at maximum, swing angle was zero, and extension distance was at its minimum. The Swing to Dump (SD) task started when the bucket height was above the ground, the bucket was curled horizontally, the swing angle was zero, and the extension distance increased to approximately 6 m. The Dumping (D) task start when the swing angle reached the pile location and the extension distance started to increase rapidly. The Swing to Trench (ST) task began when bucket height was at its maximum, bucket angle was at its minimum, swing angle was at its maximum, and the extension distance was around 8 m.

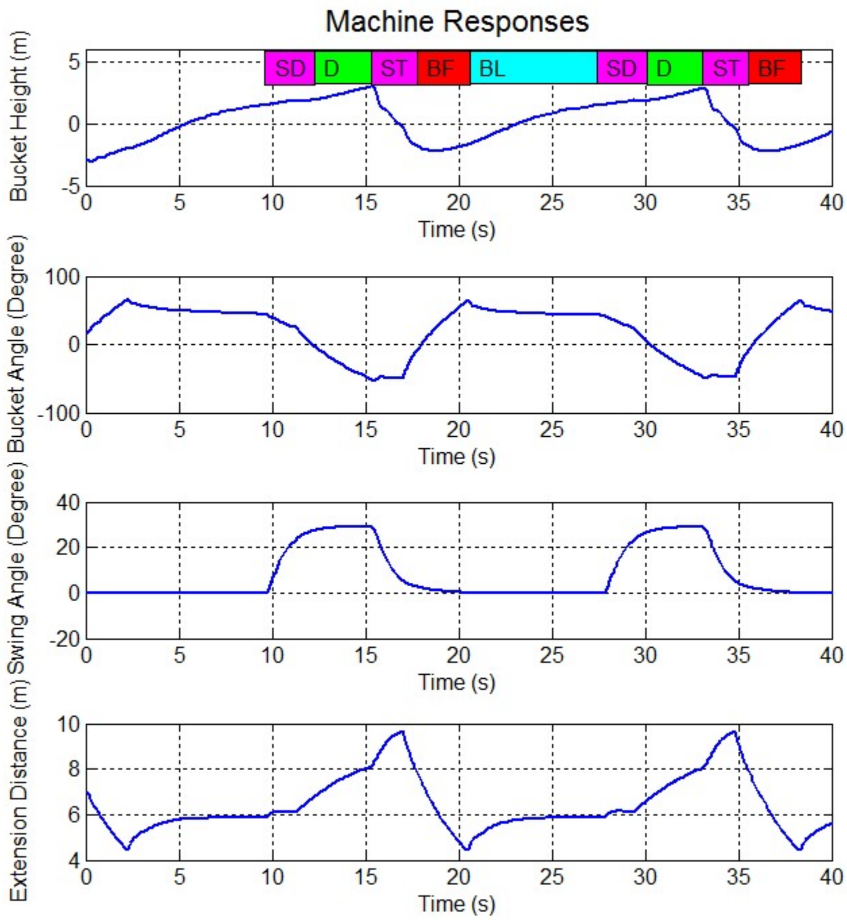


Figure 10. Machine Responses for rotate and fill strategy. The colored tabs represent the five tasks of the trenching operation.

The simulated task sequence and transitions from one task to another was similar that observed in the machine data recorded during the observed excavator operation (Figure 11). However, the simulated work cycle was longer than the machine data work cycle by 35%. At the task level, the simulated bucket fill task result was about twice as long as the average observed in the machine operation. Overall, this result was expected because the simulated task model did not include task overlap referenced by expert human operators, which would result in more efficient (i.e. shorter) work cycles.

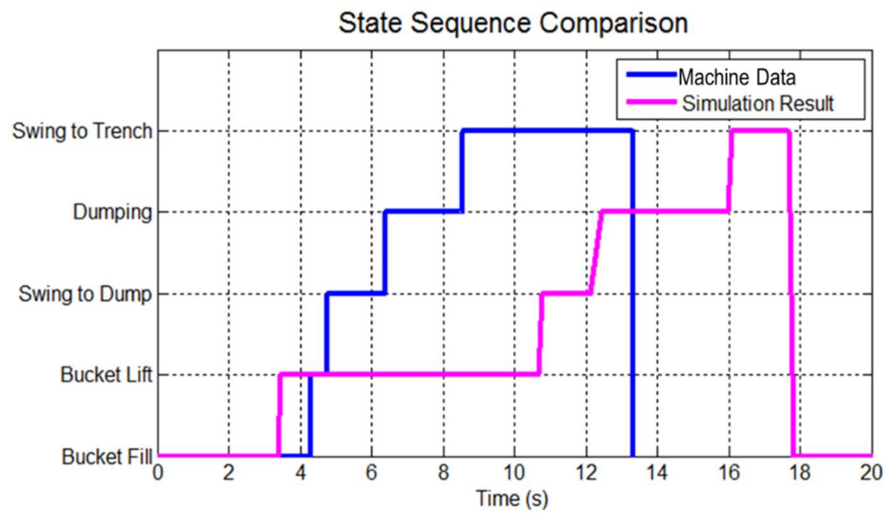


Figure 11. State Sequences of Observation Result vs. Simulation Result

When the scrape and scoop strategy was simulated, the bucket was rotated mainly near the end of the Bucket Fill task (Figure 12, see red circle). When compared with the work cycle of the rotate and fill strategy (depicted in blue in Figure 13), the work cycle of the state sequence of the scrape and scoop strategy (depicted in purple in Figure 13) is longer because of the separation of bucket movement and bucket rotation within the Bucket Fill task.

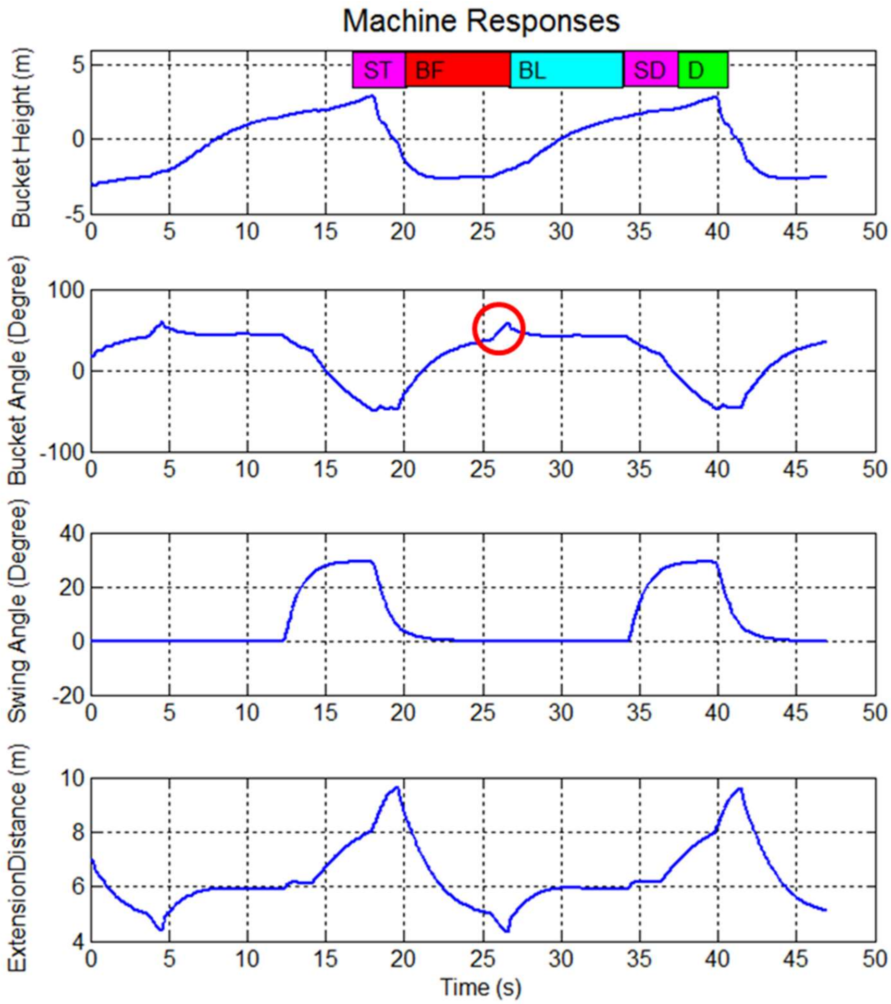


Figure 12. Machine Responses for scrape and scoop strategy. The colored tabs represent the five tasks of the trenching operation.

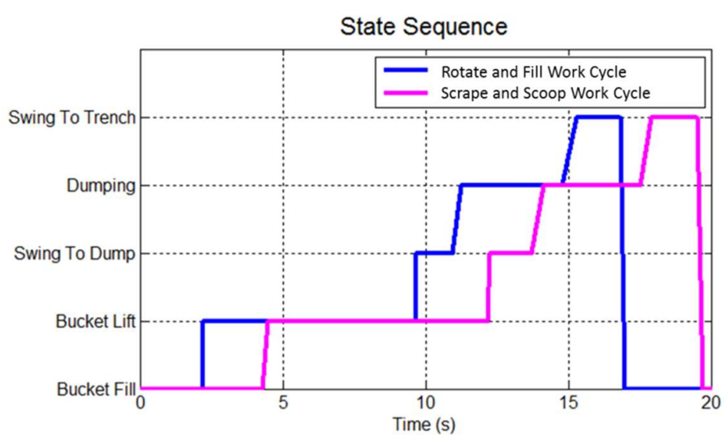


Figure 13. State Sequences comparison between rotate and fill, and scrape and scoop strategy.

Different Hydraulic Pump Speeds

Different hydraulic pump speeds will result in differing maximum pump flow capabilities. Flow will be a constraint on actuator speed when multiple actuators are demanding more flow than the pump can produce. Thus as pump speed is increased, reduction in work cycle time were expected and were exactly what was observed in simulations. The work cycle time was influenced by different hydraulic pump speeds (Figure 14). The total work cycle time decreased about 25% while increasing the pump speed by around 1000 rev/min. Most of the decrease in cycle time occurred during the Swing to Dump and Swing to Trench tasks, and the Dump task to a lesser degree. The flow demand during these tasks would be highest to simultaneously power the swing motor as well as the boom and arm cylinder. Thus any additional flow available through increased pump speed has a maximum impact during these tasks. These observations illustrate the robustness of this operator model to machine design variations and also demonstrate how the impact of machine design changes on machine performance can be assessed through a closed-loop simulation of the coupled operator and machine models.

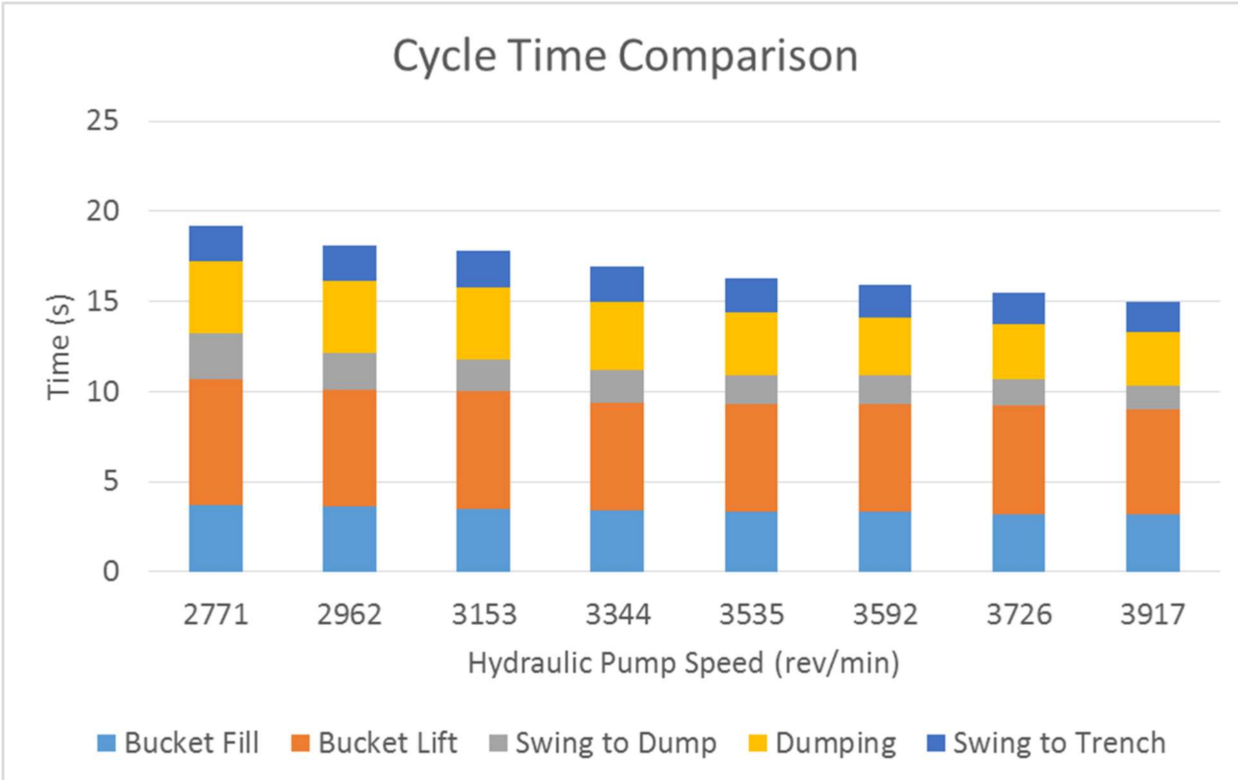


Figure 14. Cycle Time Comparison for Different Hydraulic Pump Speed.

<u>Different Pile Locations</u>

When the environment model was varied to have the machine dump at three pile locations (defined by 0.5, 1.0, and 1.5 radian swing angles between pile location and trench), the closed-loop simulation of the operator and vehicle models provided results that represented reasonable changes in the operator model behavior. The resulting swing angles increased to the representative pile locations (Figure 15) resulting in different lengths of time in the two swing tasks. The cycle time ranged from about 20 seconds for the pile at 0.5 radians from the trench to about 25 seconds for the pile at 1.5 radians (Figure 16). The time associated with the other tasks was relatively unchanged.

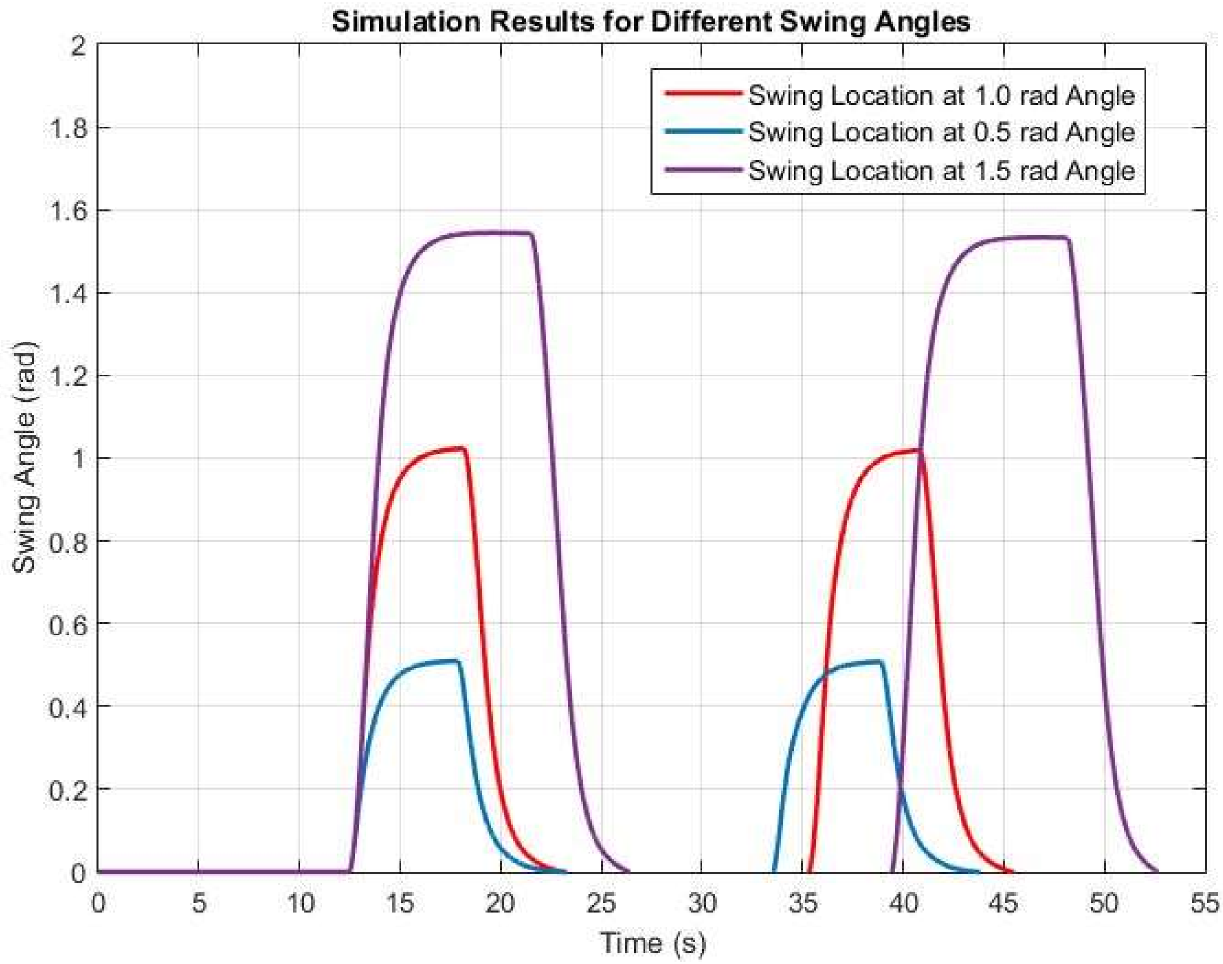


Figure 15. Simulation Results for Different Pile Locations.

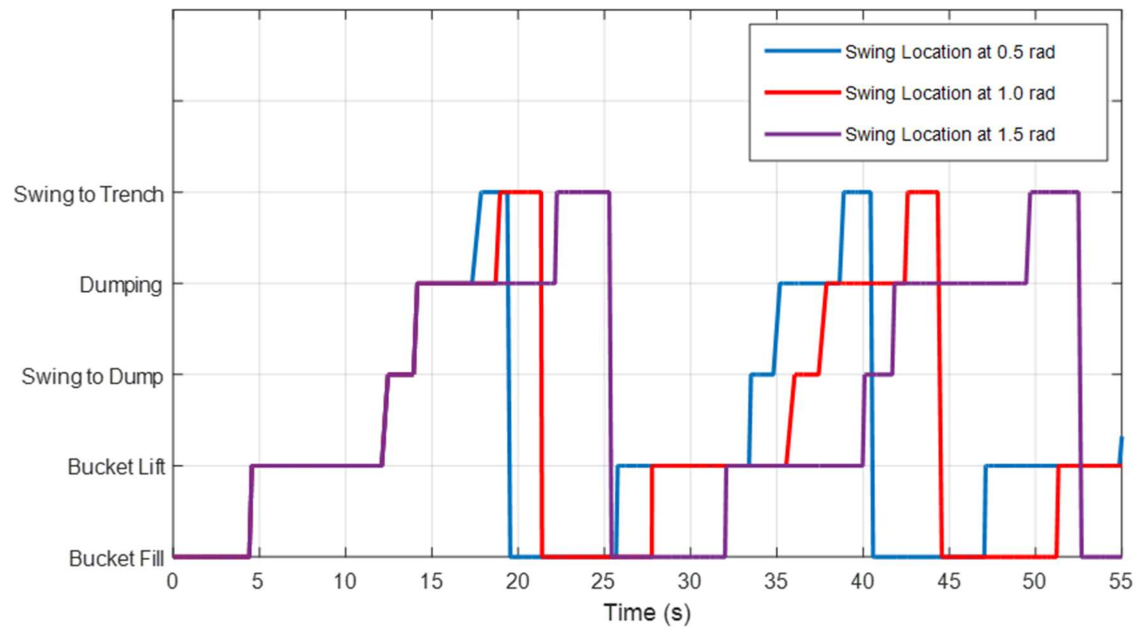


Figure 16. Task Sequence with Different Pile Locations.

Different Trench Depths

Similar to the pile location experiments, the closed-loop simulation of the operator and vehicle models also provided results representing reasonable changes in operator model behavior to three different trench depths (1.6 m, 2.2 m and 2.9 m). During the bucket filling cycle, the operator model commanded the bucket to move down to depths that were near the commanded depth with the additional time required to move the mechanism through this greater distance (see Figure 17). These results, along with those associated with the pile locations, illustrate the virtual operator model's capability to adapt to varying work cycle goals by varying operator behavior.

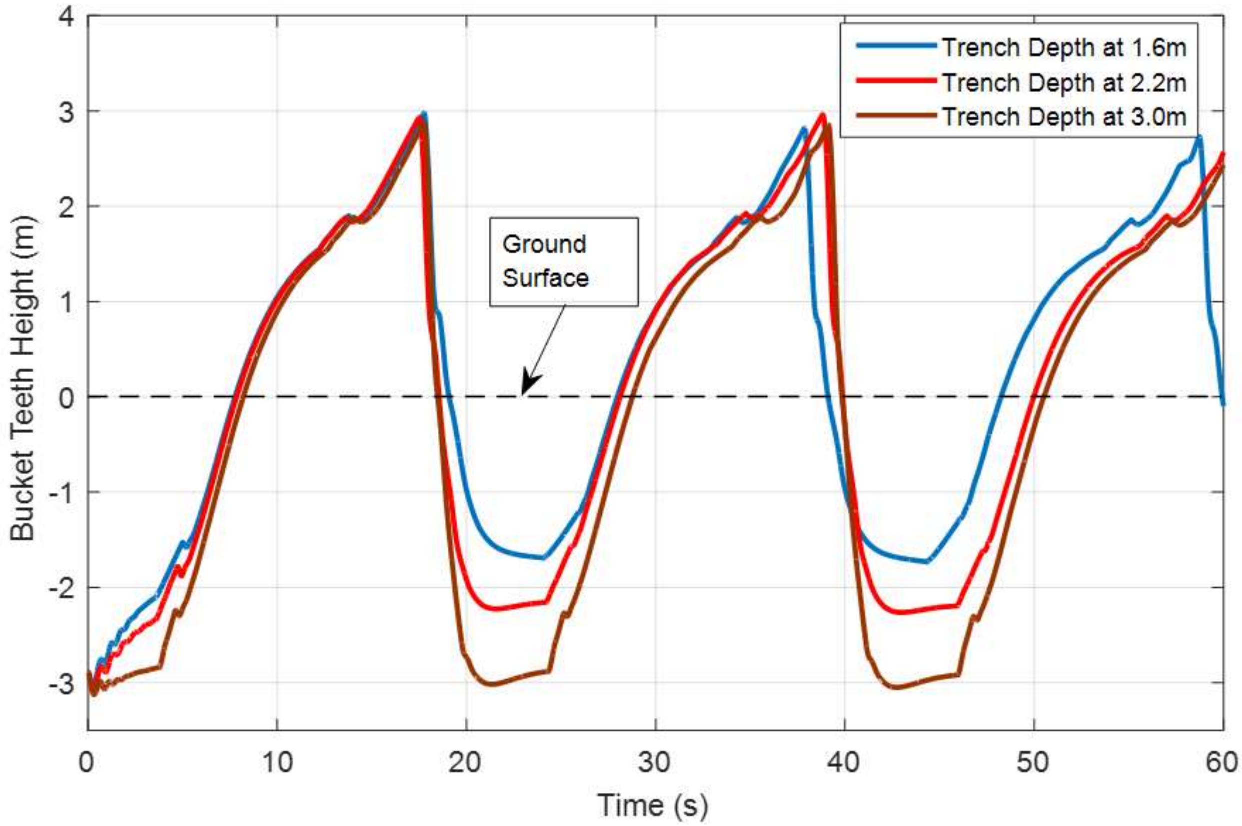


Figure 17. Simulation Results for Different Trench Depths

Conclusion

An approach or methodology for virtual operator model development was developed, resulting in the capability to simulate the function, response, and characteristics of operator behavior to simulate vehicle control inputs for an excavator trenching operation. This capability will enable simulation of virtual machine prototypes for performance analysis including fuel efficiency, productivity, and component loading. Virtual operator models enable closed-loop, whole system evaluations of new design feature early in the design process.

The approach developed in this paper combined human factors methods with dynamical system modeling techniques to capture and model operator expertise in a virtual operator model that can be used in closed loop vehicle simulation. The model is designed to capture the behavior and performance of a human operator and represent the operator in a virtual operator model that simulates authentic human behavior for a well-defined construction machine operation. The approach can be generalized to off-road vehicle simulation, and the virtual operator modeling approach can inform the machine automation design.

Through interviews and machine data analysis, it became clear that a hallmark of expert operators is the ability to overlap tasks in trenching operations, which is expected to be the case for other construction operations as well. However, virtual operator models to date have assumed discrete states for tasks. Developing a modeling approach to enable task overlap is an important direction for this work. The use of fuzzy logic allows multiple states to be active simultaneously, and thus it can be used to represent operations that include task overlap. Fuzzy logic also uses human-like reasoning rules to perceive information, and mimics the perception process of a human operator.

This work was different from the prior work in three ways. First, an explicit human factors approach was used involving human operator interviews, machine data and video analysis. Some prior work has indicated that operator models were developed with some operator considerations (Filla et al., 2005), but an explicit approach to incorporating observed human operator behavior into operator model was not found prior to our efforts. Second, prior operator model structures were not designed with operator cognitive processes in view (i.e. perception, decision-making, action), but were simply a finite state machine (Filla et al., 2005) or a combination of a finite state machine and a control module (Elezaby, 2011). Third, the other operator models generally do trajectory tracking and are based on deterministically defined processes. Our work represents an early operator-centric effort to model human decision-making and generate of behaviors based on operator goals, control strategies, and human perceivable cues.

The current state of the model generates the human operator control inputs to execute a work cycle of an excavator trenching operation. The simulation results in a work cycle that is generated by executing a series of tasks in the way a human operator would – perceiving the state of the machine, deciding when to transition from one task to the next, and controlling the machine to move the bucket through the tasks. The virtual operator model appropriately adapted to different operator control strategies, machine parameters changes (i.e. pump speed) and a change in work site goals (trench depth, pile location). The model generated outputs based on human-like perception, decision-making, and action selection.

Future work will focus on modeling the adaptability that characterizes expert human operators. The operator model should adapt to environmental conditions, such as soil properties, and operator effects. An optimal operator model should have the ability to adapt to variations in the environment by adjust operator strategies and results control inputs to the machine. Next steps include the development of an environment model, development of a task overlap paradigm to capture different operator skill levels, and development of a strategy model to enable adaptation to changing conditions. Longer term future work will investigate the utility of this virtual operator approach to the design of adaptive systems, where the automation has the authority and ability to change its mode of operation to best support joint human-automation performance. Designed with a human-information processing inspired architecture, the virtual operator model approach holds promise to develop a control logic that will be understandable to human operators, and behave in ways consistent with human operation.

CHAPTER III: MODELING ADAPTABILITY IN VIRTAL OPERATOR MODELS

Material in this chapter appeared as a journal paper:

Du, Y., Dorneich, M. C., & Steward, B. (2018). Modeling expertise and adaptability in virtual operator models. Automation in Construction, 90, 223-234. DOI: 10.1016/j.autcon.2018.02.030

Introduction

Improved machine designs are needed to meet the increasing demands on construction machines for greater functionality, productivity, and efficiency. Yet in human-machine systems, human operators play a significant role and affect system performance. Typical product design processes measure performance of a new construction machine design using expert human operators driving physical prototypes in defined test environments (Filla, Ericsson, & Palmberg, 2005). While this method produces high fidelity data, it is time consuming, resource intensive, and necessarily requires that the physical prototype of the machine be built. To advance machine design and testing, model-based design and virtual operator models can be used to explore machine designs virtually. Increasing efforts in model-based design in industry has yielded high fidelity models to test new machine designs and new features. Fidelity in this context describes the degree to which a simulation reproduces accurate and reliable behaviors of real-world phenomenon (Gross, 1999).

Virtual design, the process by which new features are modeled and tested in a simulation environment, is applied iteratively in the modern product design. Model-based or virtual design provides a means for achieving machine design improvements with reduced time and costs (Eppinger, Whitney, Smith, & Gebala, 1994). In the product development process, virtual design is often used for feature or system validation (Tseng, 1998). Virtual design is typically conducted early in the design process when it is less expensive to make changes.

Closed-loop simulation-based virtual design uses simulations that include a representation of the machine and the operator, which has feedback loops or paths between its output and its input. However, virtual design of construction machines with operators-in-the-loop has often been limited by the fidelity of the model of human operators. This limitation is particularly an issue when virtual design is used for validation and assessment. Traditional validation and assessment methods, by way of comparison, utilize physical machine prototypes, human operators, and real-world testing in a controlled environment (Filla, Ericsson, & Palmberg, 2005).

While machines have been modeled with a fidelity that enables robust testing, current operator models struggle to capture operator expertise and require time-intensive tuning to each new machine design. These limitations hamper engineers from making solid comparisons in the virtual prototyping stage between different design alternatives, and limits their ability to do virtual design. Given the tightly coupled, non-linear nature of construction machine dynamics, combined with human-in-the-loop control, dynamic simulation of the complete system must include the operator, environment, and tasks. To advance machine testing, a virtual operator model (VOM) needs to be developed to represent how human operators operate machines. The fidelity of VOMs needs to be increased by using a more human-centered basis for virtual operator modeling, and increasing the fidelity of operations modeling.

Current VOM efforts in construction have largely been restricted to developing models that mimic known trajectories, usually recorded from actual machine operations

(Filla, 2005; Elezaby, 2011). This implies that any change to the machine design would require a time-intensive process of “re-tuning” the VOM to mimic new machine trajectories. This limits their utility in fast-paced iteration in model-based design cycles. Furthermore, the work cycle of an operation has been modeled as discrete, sequential series of tasks, as the operator completes one task before moving to the next (Elezaby, 2011). However, operating the machine in such a discrete manner is typical behavior of novice users (Yu, Dorneich, & Steward, 2016). Experts can overlap tasks, beginning a new task of the work cycle while still completing the previous task. This enables the operator to “push” the machine to increase efficiency and performance. The current state of the art VOMs (Filla, 2005; Elezaby, 2011; Du et al., 2016) were developed under fixed environment conditions for particular machine models, and use finite state machine to model each of the tasks discretely in the operation.

In recent work, the authors developed a VOM based on the human information processing model to generate operator inputs based on an understanding of how humans process cues from the environment to make decisions on how to control the machine (Du et al., 2016). To inform the design of a VOM, human factors methods were used to study the behavior of human operators, including decision making, perception, and control strategies. The VOM represented the human operator decision-making process and aims to replicate how human operators operate machines. That effort simulated one work cycle, for one machine type, and assumed that each task in the work cycle was discrete. However, a robust virtual closed-loop, simulation-based design capability requires the interaction of high-fidelity models of the machine, the operator, and the environment.

To advance the utility of model-based machine testing in virtual environment, the fidelity of VOMs needs to be enhanced along the lines of representing human operator expertise in multiple ways: representation of expert human work cycle operation, and an expert's ability to adapt to changes in the work site environment and different machines. These three dimensions center on the theme of expertise and adaptability, and are the subject of this paper. There are many ways that that expertise is manifested in construction machine operators, but the three focused on in this paper are the ones that emerged from our interaction with operators and engineers in industry (Yu, Dorneich, & Steward, 2016). Increasing the fidelity of the VOM will result in a more realistic simulation of operations. Enhanced closed-loop, computer-based simulation capabilities will affect the development process through better efficiency, lower cost, and more flexibility compared to traditional machine testing in the early design process.

For this project, the excavator trenching operation was selected as the target construction machine operation for virtual operator development. Excavator trenching is a common construction operation, which contains multiple tasks that are applied and adapted to multiple situations and conditions. Based on interviews and observation, the five *tasks* making up a complete trenching *work cycle* were identified: Bucket Fill, Bucket Lift, Swing-to-Dump, Dumping, and Swing-to-Trench (Du, Dorneich, & Steward, 2016). An *operation* consists of multiple work cycles to dig a trench of a pre-specified depth. During the operation, an operator needs to dig a trench at a predetermined location and orientation with defined dimensions, and dump the material either in a defined area or into a truck.

This work was motivated by trying to model the expert behavior found in the kinds of productivity tests done in industry during the design process, where expert test operators run pre-defined operations to evaluate the machine. These productivity tests are explicitly designed to push the operator and machine to maximum effort to understand the limits of the machine (e.g. Link-Belt, 2009). Thus, industry test operators tend to work at their maximum ability to finish trenching as soon as possible. To enable closed-loop simulation of a trenching operation, the VOM must generate human operator behavior based on cues that are perceivable to the operator, account for changes in the environment affecting the operation as it progresses, and adapt to different situations or disturbances during the operation.

The VOM should simulate expert control of the machine. It takes an expert "pushing" the machine to its limits to test the capability of a new design to increase productivity. In interviews with construction machine operators (Yu, Dorneich, and Steward, 2016), the concept of overlapping tasks emerged very quickly in those discussions as a key way that expertise is manifested in a repetitive task-based work cycle in construction machines like the case of the excavator being used to dig trenches. However, current VOMs are developed without consideration of how expertise is manifested by real operators. Expert human operators can start attending to the next task while the current task is nearing completion. A VOM that models overlaps in operator attention to multiple tasks is needed to generate more realistic control inputs.

Experts are able to adjust the machine operation based on changes in the work site. Simulation using current VOMs can only simulate and repeat a work cycle without adaptation to the changes in the environment. However, for the trenching operation, the dimensions of the trench and material dump pile change after each work cycle. Human operators adapt to the changes in the environment and adjust their control of the machine. It typically takes multiple work cycles to complete the operation. A model that tracks changes to the environment is needed, where the VOM adjusts operator control inputs as the work site environment is changed by the machine operations.

Finally, another aspect of expertise is an operator's ability to start using different machine makes and models and very quickly operate them at a high level of productivity. Different excavators can be used in the same construction site depending on the capability required. Different excavators share general control features, and so expert operators can apply their general knowledge of excavator operation when switching between different excavator models and capabilities. Human operators use their generalized knowledge of machine control to understand the differences between excavators and adjust their control behavior to operate different equipment without much effort. However, current VOMs based on trajectory mimicking are unable to adjust to changes in machine dimensions, power, and capabilities. Based on discussions with industry experts, significant effort is required to tune the current VOMs to simulate a different machine models. Current VOMs cannot adapt to differences in machine models (Yu et al., 2016). However, the VOM architecture approach described in this paper generates control input by simulating operator processing of information (cues from the machine and the environment) to generate control inputs based on operational goals, not on a pre-defined, pre-learned trajectory. This method provides the possibility of the model automatically adapting to different machine models since the VOM reasoning is based on operator perceptible cues, and not machine geometric dimensions. The VOM must be generalized such that new machine characteristics are accounted for as the operation is simulated.

In this work, a fixed VOM (Du et al., 2016) was extended to simulate expert behavior by enabling tasks to overlap in the work cycle. The VOM was also extended to simulate a complete trenching operation where the operator model adapted to changes in the work site environment. Finally, the VOM was generalized to be independent of the machine model, and generates the machine model control inputs based on a model of human decision making rather than tracing pre-defined trajectories.

The following section reviews the previous work related to operator modeling. The VOM approach is presented, and the methods to address the three areas limiting the fidelity of current VOMs. Four case studies are represented, with results demonstrating the approach. Finally, current and future work is discussed.

Related Work

The current state of the science in virtual operator modeling for off-road machinery is comprised primarily of three other examples: Two VOMs for wheel loaders demonstrating a task-oriented modeling approach (Filla, 2005 and Elezaby, 2011), and a VOM for a steering controller demonstrating a reference-oriented approach (Norris, 2001). Task-oriented operations are those in which the operator controls the machine through a repeated sequence of tasks to accomplish high-level goals (Alami, Chatila, Fleury, Ghallab, & Ingrand, 1998). In reference-oriented operations, the operator is guiding the machinery along a particular path to explicitly follow speed or position references (Zhang, Alleyne, & Carter, 2003). Both task-oriented operator models treat the operator as a finite state machine without a clear structure to capture human operator perception, decision, and action processes. Fuzzy logic was used for the reference-oriented approach to model the perception process of the human operator. These operator models did not adapt and modelled only a single work cycle for a specific machine. Some off-road vehicle automation research also has relevancy (Bradley & Seward, 1998; Wu, 2003; Enes, 2010) in which operator behavior and strategy were modeled to automate certain tasks. The current literature does not address representation of expertise nor adaption to environment changes and different machines.

Representation of Expert Operation

Elezaby (2011) used feedback from the loader machine model to transition the finite state machine from one state (or task) to another. Control reference inputs defined by initialization of task description and performance requirements associated with each sub-task were sent to controllers. Five evaluation (Elezaby & Cetinkunt, 2011) tests were designed by exercising the machine model to move to defined positions. Similarly, Bradley et al. (1998) modeled trenching as a series of discrete tasks in a task-oriented approach similar to those of Filla et al (2005) and Elezaby (2011). In these examples, tasks were modeled in finite sequence, behavior that is typically associated with novice operators (Du et al., 2016).

Filla et al., (2005) took a generic task analysis approach based on operator interviews to derive different tasks for the wheel loader loading cycle, but specific operator behaviors with expertise representation were not considered in this approach. They developed machine harmony diagrams of bucket height and machine location, based on recorded data of actual machine operation. A machine harmony diagram is used to represent the relationship between two different motions during the machine operation. A machine harmony diagram was developed to characterize operator behavior by representing the bucket height at different locations during operation, based on recorded data of actual machine operation (Filla, 2005). Different harmony diagrams could potentially represent different human operator behavior (and expertise levels) in terms of the travel distance of the wheel loader and the lifting height of the bucket. However, this was not implemented in Filla's work.

Norris et al. (2003) provided a reference-oriented operator modeling approach where the operator model follows a defined trajectory to steer machines. Fuzzy logic was used to develop a human operator performance model, which was used as an expert reference model. Fuzzy rule membership functions were defined by mapping empirical expert knowledge and data from experiments, corresponding to different command levels. The commands were transferred to different levels of control inputs by relationships determined by training the operator model with experimental data. Operator behavior could be replicated, but no attempt was made to differentiate operator behaviors.

Wu et al. (2003) developed an automatic digging controller for a wheel loader operating on rock or soil. The operational techniques and strategies derived from the operator interviews and data analysis were replicated by using fuzzy neural networks in the digging controller. This could be a way to replicate certain level of expertise, but is limited the data set available.

Representation of Operator Adaptability

Human operators adapt to work site changes dynamically by adjusting control inputs. Both Filla (2005) and Elezaby (2011) modeled the short loading operation with defined tasks, identified through task and data analysis. Elezaby created a strategy model, which contains a set of rules and a finite state machine, and chooses a specific task with appropriate reference commands according to the information received from initialization of task description, loader conditions, static site conditions, and feedback from the machine model. The site conditions were defined with help of GPS signals and were given at the beginning of the simulation. The site conditions were used only as the initialization information, and cannot be updated dynamically from work cycle to work cycle. Therefore, this operator model did not adapt to changes in the work environment, which limits their simulation to a single work cycle, but not an entire operation that requires multiple work cycles.

Filla et al., (2005) developed a VOM for wheel loader operator, and divided the loading operation into finite tasks with a defined sequence. The VOM could be initialized with different site layouts. However, once initialized, the environmental conditions did not change dynamically during the simulation of the operation. This limited the simulation to one work cycle. Filla suggested that the adaptability of the models to certain changes in the environment could be an area for future improvement.

Bradley et al. (1998) developed the LUCIE autonomous robotic excavator based on human operator strategies for trenching. This research stated that imitating the operator’s behavior could be effective basis for automation system design (Bradley, 1998). They developed multiple strategies for digging a trench, and a strategy to dig around an obstacle. Based on the properties of the soil, the system chose a predefined strategy. When encountering an obstacle of known size and location, the system employed a specialized strategy to dig out the obstacle. The control inputs were determined by the strategy chosen. In this way, the system adapted to the environmental condition of soil type and obstacles, but not changes in the trench or pile.

In summary, the literature directly related to VOMs provide limited guidance on how to represent human operator adaptability and expertise. This work aims to take the next step in the representation of expertise and adaptation.

Approach

The VOM previously developed by the authors could simulate one excavator model completing one work cycle. The work documented in this paper extends this approach by developing methods to represent expert operator behavior, and to represent an operator’s ability adapt to a changing work site environment and different machine models. The following subsections introduce the VOM architecture and methods to represent expertise and adaptability.

Virtual Operator Model (VOM) Architecture

Previous work established the VOM architecture (Du, et al., 2016). The VOM was developed to simulate human operator control inputs to an excavator model trenching operations. The VOM was explicitly designed to be independent of the excavator model (Du, Dorneich, & Steward, 2015). Without this independence, operator models that are highly tuned for particular machine models must be retuned when machine designs are changed. To avoid the cumbersome nature of this tight dependency, an operator model should adapt to changes in machine capabilities such as available power or mechanical linkage constraints.

The architecture of the VOM (Figure 18) is based on the Human Information Processing model (Newell & Simon, 1972) to represent the internal decision-making processes of a human operator. Humans detect signals from the environment, analyze the information based on their knowledge and skills to make decisions, and then take actions to execute their decisions. This information processing sequence is mirrored in the structure of the VOM via a series of four modules. The VOM was implemented in Matlab (Ver. 2015a, Mathworks, Natick, MA) and was interfaced with the machine model. The Machine Model was developed using SimMechanics, and SimHydraulics, which are also part of Matlab.

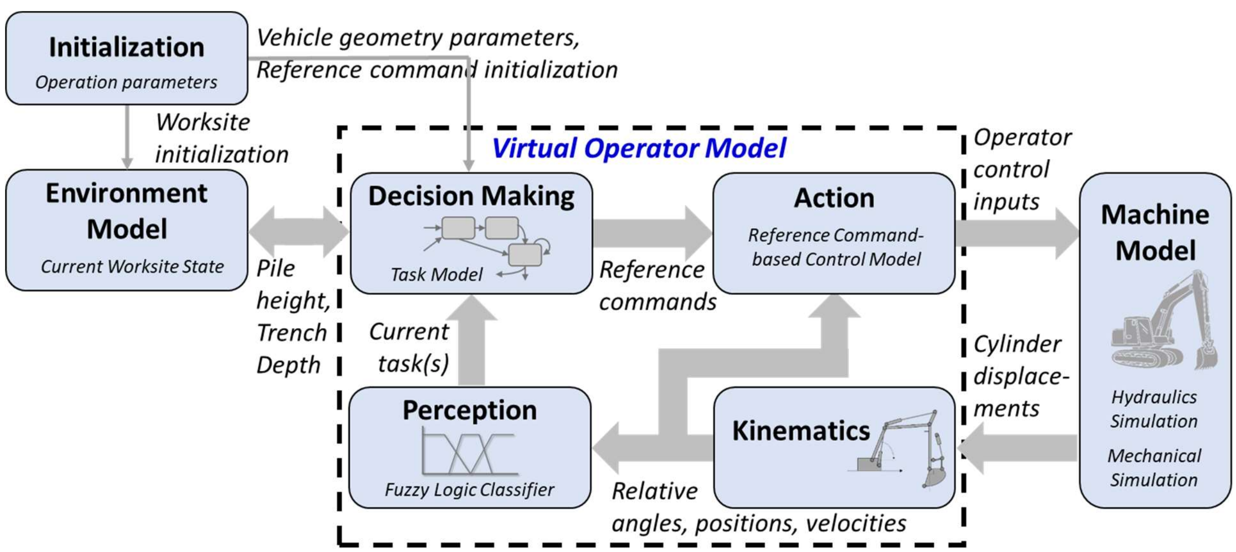


Figure 18. Virtual operator model structure consists of several models representing human information processing.

The modules are summarized here; for a detailed description, see Yu et al. (2006). Before an operator can respond to environmental cues, machine information must be converted to human-perceivable cues. For example, human operators can perceive the height of the bucket relative to the ground, but not the pressure in a cylinder. The machine model provides machine signals like cylinder extension length and velocity. The kinematics module is responsible for acquiring the machine model signals and transforming them into information at the human perception level that human operators use for decision making. Machine data can be translated into the absolute positions and orientations of machine elements (such as buckets and booms) through the kinematic module of the machine. These dynamic variables are closely related to the visual cues that the human operator uses for decision making during the work cycle.

The human perception module receives information about environment conditions and machine components, which are observable by human operators. Fuzzy classifiers then classify the human-perceived information to trigger task transitions between different tasks of the trenching work cycle. The outputs of the classifiers are the inputs to the state machine in the human perception module, which determines the current task state. The human decision module determines the reference commands by considering environment information, machine geometry and the current state. The reference commands are target positions of machine components for the current task, which are sent to the human action module. In the human action module, reference commands are compared to the current position of machine components. The differences, or errors, are used to generate control inputs for machine model through PID controllers. A PID controller stands for a proportional-integral-derivative controller, which is a standard closed loop feedback controller.

Representation of Expert Operation

Expert human operation can be represented in different ways, such as advanced strategies and how they coordinate work cycle tasks. This paper focuses on representing the expert human operation through the operator's ability to overlap tasks. To enable task to overlap in the work cycle, the way tasks are classified in the human perception model must be modified and the task model must be updated to include overlap states.

Task Classification

The human perception model was developed to simulate how human operators perceive information to classify tasks. Previous work defined tasks as discrete, and so classification of tasks needed only to detect the transition from one task to the next (Du et al., 2016). However, expert human operators start to shift their attention from their current task to the next before the current task is complete. This shifting of attention that occurs simultaneous to the control requires the modeling of overlapped tasks (Salvucci, 2009). The possible overlaps happen between two consecutive tasks in the work cycle. It is also possible to overlap the three tasks of Bucket Lift, Swing-to-Dump, and Dump. The complete task model contained 11 states, five major tasks and six overlap tasks (Figure 19). In the perception module, ten classifiers were developed to determine the start and end of each of the five tasks, based on the human perceptible cues from the kinematic module. The classifiers were designed and implemented with the Matlab Fuzzy Logic Toolbox. With the start and end of each task, time spans for each task were determined. Overlaps are determined by comparing the timing of each of the tasks. A state machine was programmed in the Simulink using scripts to represent the task model and determine the current state using the results from classifiers. Table 7 describes the tasks corresponding to the state numbers.

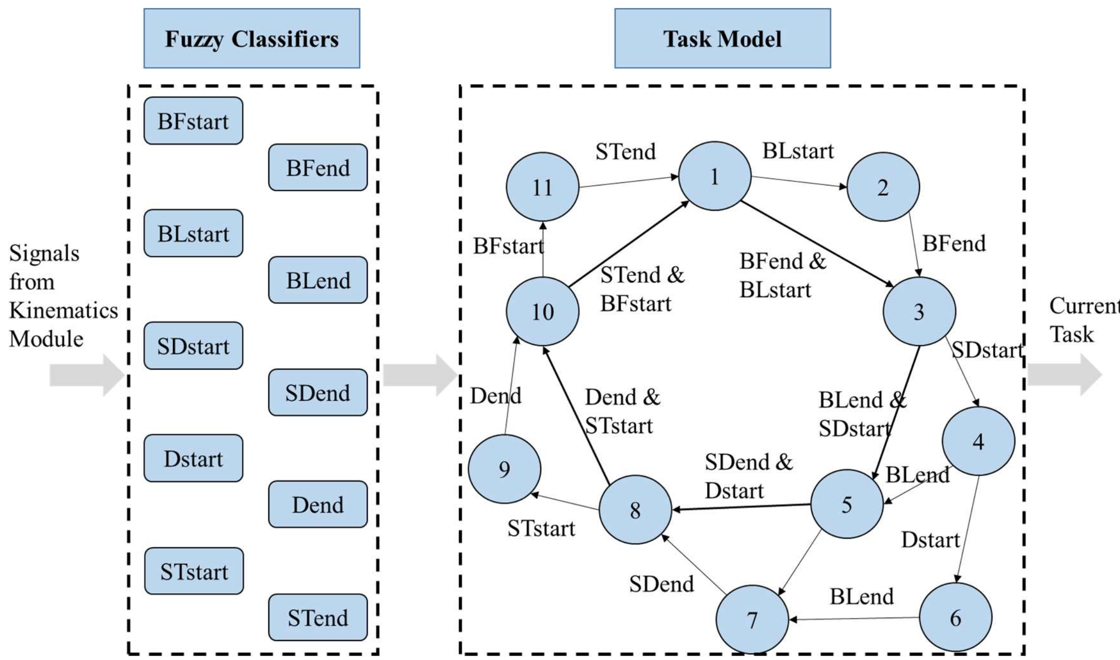


Figure 19. The fuzzy classifiers received signals from the kinematics module and determined the start and end transitions for the tasks, which were used by the task model which was a finite state machine containing five task states and six overlap states.

Table 7. The trenching operation was modeled with using 11 states which represented five tasks and the six possible task overlap conditions.

State	Tasks
1	Bucket Fill
2	Overlap of Bucket Fill and Bucket Lift
3	Bucket Lift
4	Overlap of Bucket Lift and Swing-to-Dump
5	Swing-to-Dump
6	Overlap of Bucket Lift, Swing-to-Dump, and Dump
7	Overlap of Swing-to-Dump and Dump

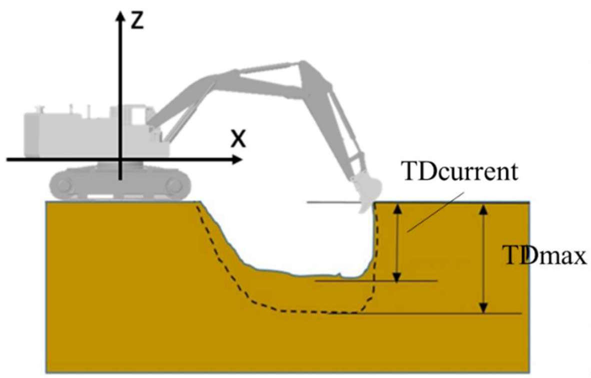
8	Dump
9	Overlap of Dump and Swing-to-Trench
10	Swing-to-Trench
11	Overlap of Swing-to-Trench and Bucket Fill

Modeling Human Adaptability to Changes in the Work Site Environment

As human operators complete a work cycle, they affect changes to the work site environment. Simply, the trench becomes deeper and the pile grows higher with each work cycle until the operation ends with the desired trench depth. To increase the fidelity of how the trenching operation is represented, changes in the work site after each work cycle need to be modeled, which requires that the VOM adjust control inputs after each work cycle to adapt to changes in the work site environment. An environment model was developed to describe the current work site environment conditions at any point during the operation, much like the mental model of a human operator is continuously updated. The model maintained the current trench depth and pile height, and the changes from the previous work cycle to current work cycle. The model represents the operator's simple internal representation about the observable changes in the worksite environment. The information from an environment model was used to determine appropriate reference commands for tasks.

Trench Model

The trench model describes the dimensions, location, and current depth of the trench. It was assumed that the trench was located in front of the excavator. The model used five parameters to describe the trench (Figure 20). The excavator was assumed to be positioned with the trench directly in front of the cab at a zero swing angle to the excavator. The parameters of the operation were defined at initialization, including the trench depth start (*TDstart*), depth increase per cut (*ΔTD*), and maximum trench depth (*TDmax*).



Environment Parameters	
Parameter	**Description**
TDmax	Max. Trench Depth intended
TDcurrent	Current Trench Depth
ΔTD	Trench Depth Increase per Cut
TDstart	Trench Depth start
n	Number of Cut

Figure 20. The trench model specifies the location of the trench and updates the trench depth after every work cycle.

The trench depth is updated every work cycle using the relationship,

$$TDcurrent = TDstart + n \times \Delta TD$$

(1)

where the current pile height, *TDcurrent*, was the sum of initial trench depth, *TDstart* and trench depth increment, *ΔTD*, multiplied by the number work cycles, *n*, completed.

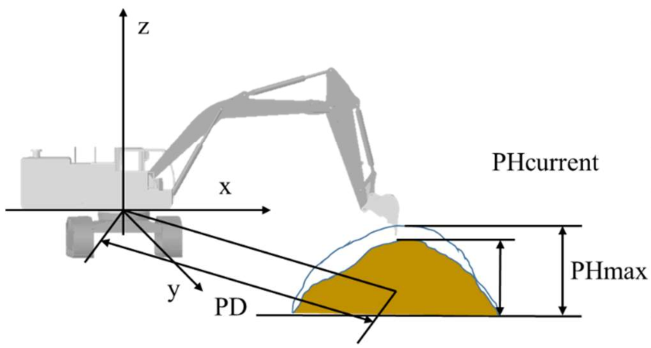
The operation continues until the current pile height reaches the target pile height, which is stated mathematically with the condition,

$$TDcurrent \leq TDmax \tag{2}$$

where *TDmax* is the maximum trench depth. The trench depth is rest to zero once the maximum is reached, much as if the machine moved to a new position and then continue to lengthen the trench. Based on these changing relationships throughout the operation, the reference commands are updated and provided to the action module to command the machine model.

Pile Model

A pile model describes the pile dimensions and location. The height of the pile is updated after each work cycle. The pile model is parameterized by seven parameters (Figure 21).



Environment Parameters	
Parameter	**Description**
PHmax	Max. Pile Height intended
PHcurrent	Current Pile Height
ΔPH	Pile Height Increase per Dump
PHstart	Pile Height start
PLA	Pile Location Angle
PD	Pile Distance
n	Number of Dump

Figure 21. The pile model specifies the location of the pile and updates **the height of the pile after every work cycle. Seven parameters described the pile using this model.**

The angle that the excavator needs to swing through from trench to pile (*PLA*) and the extension distance of the bucket between the cab and the pile (*PD*) are initialized at the beginning of simulation to describe the worksite conditions. After initialization, these variables are available for the VOM, which calculates reference commands for each simulation step. The pile height is updated every work cycle using the relationship,

$$PHcurrent = PHstart + n \times \Delta PH \tag{3}$$

where the current pile height, *PHcurrent*, was the sum of initial pile height, *PHstart*, and pile height increment, ΔPH, multiplied by the number work cycles, *n*, completed.

The operation continues until the current pile height reach the target pile height, which is stated mathematically with the condition,

$$PHcurrent \leq PHmax \tag{4}$$

where $PHmax$ is the maximum pile height. The pile is reset to zero once the maximum pile height is reached, much as a pile would be cleared or trucked away once it had reached a certain size. Based on these changing relationships throughout the operation, the reference commands are updated and provided to the action module to command the machine model.

Adaptation to Different Machines

Differences exists between the type and models of excavators; human operators can operate different excavators without much effort to adapt to these differences. Previous work in VOMs (Filla, 2005; Elezaby, 2011; Du, 2015) focused on one particular machine model. Considerable efforts are typically required to modify the VOM to modify it to control a different machine model. Initialization and parametrization methods were developed to create a generalized VOM, which can adapt to different machine models and generate control inputs accordingly. When different machine models are simulated, the VOM only needs to be initialized with geometric parameters such as length of the boom, arm, and bucket. To simulate different machine models, the VOM adapts general knowledge of different machine model geometries to enable simulation of different machine models without modification of the VOM. The VOM updated the fuzzy classifiers to use the relative rather than absolute signals to detect tasks within the work cycle. The reference commands are updated automatically for use by the action module to correctly control each new machine,

VOM Initialization and Parameterization

At the beginning of each simulation, all the variables were initialized by reading values from an external file. Information about machine geometry, environment, and strategies was included in this file. Machine geometry information was used to describe the kinematics in the initialization process and was used to calculate reference commands. Operator strategies were represented in strategy variables, examples of which include the bucket height during the Swing-to-Trench task. The bucket height during swing depended on human operator perception of the environment, how high above the ground the human operators would feel is safe in order to not hit obstacles or avoid adjusting the bucket height. For example, reference commands for arm angles can be defined in the initialization file to ensure certain positions of the arm during digging. All the information can be modified and defined outside of the VOM, which means only the initialization files need to be modified to simulate different machine models, in different environment settings, or with different operation strategies.

Reference Commands Calculation

Reference commands were used to set the targets locations of the machine components for each task in a work cycle, and the machine was guided to reach the targets through the action module. In real operations, human operators adjust their targets based on the environment, machine, and strategy, and adjust their control inputs to adapt to these changes. The initialization file sets the strategy and specifies the machine dimensions. The environmental model updates the pile height and trench dimensions each work cycle. The VOM uses all this information to update the reference commands for each work cycle. The updated reference commands adjust the control inputs in the action module.

Generalize Fuzzy Classifiers

To enable robust simulation with different machine models, fuzzy classifiers need to classify current state correctly independent of machine dimensions. Signals used in fuzzy classifiers should be human perceivable and general for different machines (see Table 8). Membership functions within each classifier were defined with relative positions and distances, rather than pre-defined numerical thresholds. Signals for physical positions are constructed with general meanings: for example, Extension Distance relative to target location was classified to values of large, small, and reached (target). Similarly, Bucket Height is classified to values relative to the ground of *AboveSurface* (above ground surface), *NearSurface* (near ground surface), and *BelowSurface* (underground), independent of machine dimensions. Signals for relative positions are used in the fuzzy classifiers to determine whether the machine reaches the target position commanded by reference commands. Relative Bucket Height, Relative Bucket Rotation, and Relative Swing Angle are the difference between the current values and the target values. Relative Extension Distance is the ratio between the current value and the target value.

Table 8. Eight signals were used for by the fuzzy classifiers to estimate the start and end transitions of the five tasks.

Signal Name	**Unit**	**Description**	**Levels**
Bucket Height	m	height between bucket teeth and ground surface	*BelowSurface*, *NearSurface*, and *AboveSurface*
Bucket Rotation	rad	angle between the line of bucket teeth and arm bucket joint, and vertical direction	*Uncurled, CurledMiddle*, and *CurledHigh*
Swing Angle	rad	angle of rotation of cab	*AtTrench*, *InBetween*, and *DumpArea*
Distance	m	relative value of comparison of the distance between bucket and cab and the distance between joint boom arm and cab	*Retracted, Midrange*, and *Extended*
Relative Bucket Height	m	Relative value of comparison of current bucket height to target bucket height determined by reference commands.	*Small, Reached, Large*
Relative Extension Distance	(unit-less)	Relative value of comparison of current extension distance to target extension distance determined by reference commands.	*Small, Reached, Large*
Relative Bucket Rotation	rad	Relative value of comparison of current bucket rotation angle to target bucket rotation angle determined by reference commands.	*Small, Reached, Large*
Relative Swing Angle	rad	Relative value of comparison of current swing angle to target swing angle determined by reference commands.	*Small, Reached, Large*

Material and Methods

Three case studies were developed to demonstrate the ability of the VOM to represent expertise, and to adapt to changes in the environment and different machines. A fourth case study compared VOM and human-generated data. Several different environments models and different machine models were developed for use in the case studies.

Work Site Environment Model

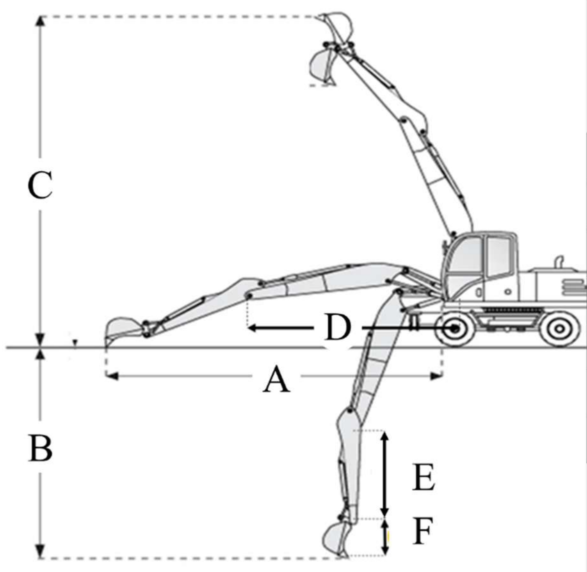
Three different work site environment scenarios were defined by changing the location of the pile, the maximum height of the pile, and the maximum depth of the trench (Table 9). The pile was assumed to be removed when it reached the maximum pile height. For instance, a truck, once filled, will be replaced by an empty truck between work cycles.

Table 9. Three environment scenarios were defined by different Pile and Trench Parameters

Environment Scenario	Pile Location				Trench Dimension	
	Angle between Trench and Pile	Distance between Pile and Cab	Desired Pile Height	Pile Height Increment per Dump	Desired Trench Depth	Trench Depth Increment per Cut
	PLA **(rad)**	***PD*** **(m)**	***PHmax*** **(m)**	***ΔPH*** **(m)**	***TDmax*** **(m)**	***ΔTD*** **(m)**
Env1	0.8	6.5	1.5	0.3	1.5	0.3
Env2	1.0	7.0	2	0.3	2	0.3
Env3	1.3	7.5	2.5	0.3	2.5	0.3

Machine Models

Three machine models were used in the simulations to represent the geometry of three excavators. Figure 22 describes the dimension for three different excavator models, where A is the furthest reach, B is the deepest reach, C is the highest reach, D is the length of Boom, E is the length of Arm, and F is the length between bucket teeth and joint between Bucket and Arm. The three excavators were labeled MM1, MM2, and MM3. MM1 is smaller than MM2 and MM3 in terms of geometry. MM2 and MM3 have small differences in dimensions. MM1 was chosen to compare large differences in dimensions. MM2 and MM3 were chosen to see the effect of small differences in the machine geometries.



Index	Description	MM1	MM2	MM3
A	Max. Extension (m)	9.95	11.12	10.76
B	Max. Depth (m)	6.42	7.22	6.68
C	Max. Height (m)	9.85	10.74	9.93
D	Boom Length (m)	5.68	6.52	6.45
E	Arm Length (m)	2.91	3.92	3.2
F	Bucket Length (m)	1.47	1.74	1.61

Figure 22. Three machine models with different dimensions, which define the maximum reaches of the excavator.

Case studies

Several case study were developed to demonstrate the adaptability of the VOM. In case study 1, the close-loop simulation was run to demonstrate overlap between tasks. The percentage of time that two or more tasks overlapped during a trenching operation was calculated for two excavator models: MM1 and MM3 under the Env2 condition. The third excavator model, MM2 was simulated under all three different environment conditions. The fourth case study compared VOM and human-generated data for the bucket height for one trenching operation.

Case Study 2 tested the VOM’s adaptability to different environment settings and dynamic changes in the environment. This case study assumed expert behavior with task overlap. The VOM and the same excavator model (MM2) were simulated for all three environment models. The environment settings were initialized at the beginning of simulations. For each environment model, total operation time and the total number of work cycles to dig trenches of different maximum depths, different pile locations, and different maximum pile heights were recorded.

Case Study 3 demonstrated the VOM's ability to simulate different excavator models by adapting to the differences in machine dimensions and adjust control inputs accordingly. The environment scenario used in this case study was Env2. Differences between machines using the VOM were demonstrated by comparing the combinations of the boom, arm, and bucket at different stages of the work cycle. In addition, the trajectories through the work cycles of the operation were compared across the three excavator models.

Case Study 4 compared the VOM results to an actual human-operated trenching operation for which machine data was recorded. The simulation results were compared to the actual trenching operation under the similar environment conditions, and same dimensions used in vehicle model. The bucket height over the pile was 2.7m, trench depth is 4.9m, and the swing angle was 45 degrees. The dimensions related information used to model the vehicle model include boom geometry, arm geometry, bucket geometry, and cylinder geometries for boom, arm, and bucket. These parameters were matched accordingly in the initialization file for the simulation, which was initialized the environment information at the beginning of the simulation.

Results

Case Study 1: Expert Operation

The proportion of overlap time was consistent over different combinations of excavator models and environment models (Table 4). Under the same environment situation, Env2, the overlap rate for MM3 was the largest at 31.7%, while the overlap proportion of the overall work cycle for MM1 was the smallest at 26.8%. By using the same excavator model MM2, the overlap rate under Env1 was the largest at 31.1%, while the overlap rate under Env3 was the smallest at 27.5%.

Table 10. Overlap Rate for different excavator models under different environment situations

	Overlap Percentage		
	MM1	**MM2**	**MM3**
Env1	--	27.1%	--
Env2	26.8%	29.3%	31.7%
Env3	--	26.6%	--

The overlap tasks varied between 26.6% to 31.7% of work cycle time, which increases as trench depth increases and pile height increase. The different pile locations affect the total work cycle time, mainly through the extension of the bucket to the pile area, since different distances between pile and cab were defined. Env1 has the shortest time for the single task swing due to shorter distance between pile and cab, which resulted a larger proportion for the overlap states for MM2. Env3 has the longest time for single task swing due to longer distance between pile and trench, which resulted a smaller proportion for the overlap states for MM3.

Case Study 2: Simulation Results with Different Environment Parameters

As the maximum trench depth increases, more work cycles were required to dig to the required depth (Figure 23). The increase in the number of work cycles increased in a nonlinear fashion with depth. For the case of a 1.5-meter maximum trench depth, five work cycles were required to reach the desired depth. Seven work cycles were required to reach the 2-meter depth. Eight work cycles were required to reach the 2.5-meter depth.

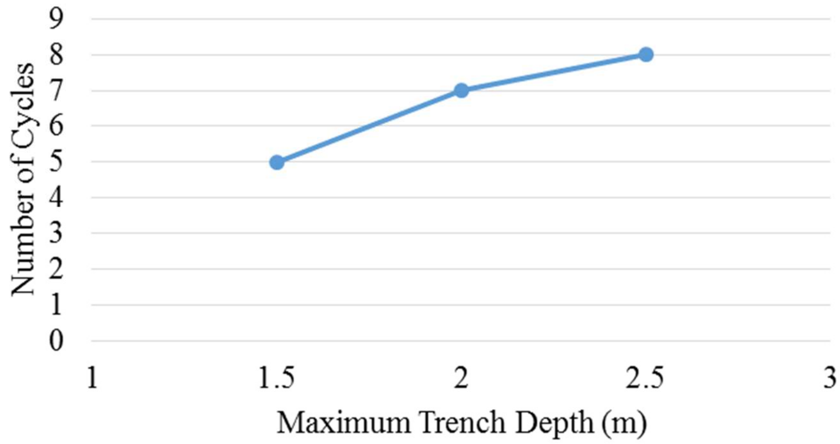


Figure 23. Number of Cycles to reach different maximum trench depths

Likewise, the time to reach the maximum trench depth increased with deeper trenches, as illustrated in Figure 24. For the case of a 1.5-meter maximum trench depth, 118 seconds were required to reach the desired depth. A time of 166 seconds was required to reach 2-meter depth. A time of 188 seconds was required to reach the 2.5-meter depth.

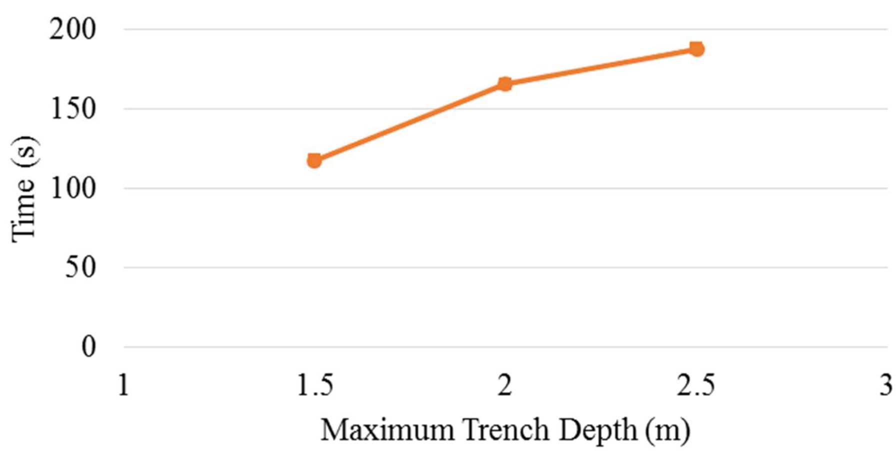


Figure 24. Time needed to reach different maximum trench depths

As the maximum pile height increases, more work cycles were required to achieve the required height, as illustrated in (Figure 25). The increase in the number of work cycles increased in a linear fashion with depth. For the case of a 1.5-meter maximum pile height, two work cycles were required to reach the desired height. Three work cycles were required to reach the 2-meter height. Four work cycles were required to reach the 2.5-meter height.

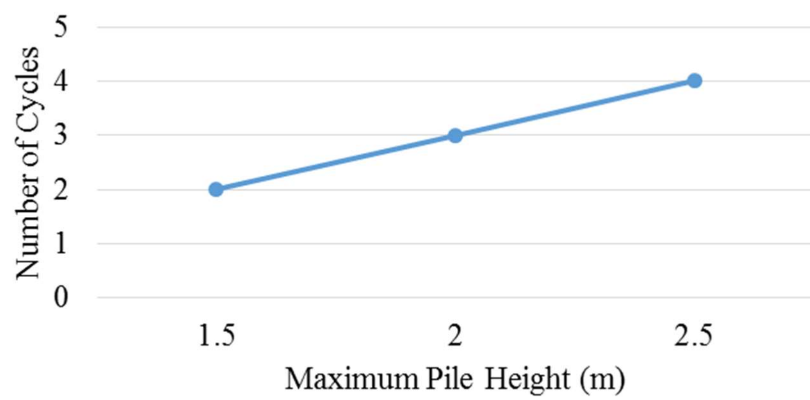


Figure 25. The number of cycles to reach different maximum pile heights.

Likewise, the time to reach the maximum pile height will increase with higher pile height, as illustrated in Figure 26. For the case of a 1.5-meter maximum pile height, 35 seconds were required to reach the desired height. A time of 56 seconds were required to reach the 2-meter height. A time of 78 seconds were required to reach the 2.5-meter height.

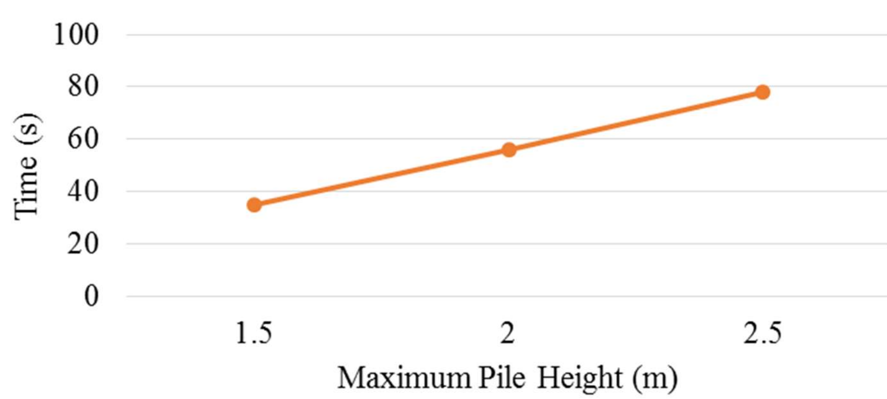


Figure 26. Time needed to reach different maximum pile heights

Figure 27 represents the bucket height during an operation with different environment settings. The operation took more cycles for the bucket to reach deeper trench depths. The dashed black line arrow indicates the trench depth is getting lower during the simulation. The solid black line arrow indicated the pile height was getting higher during the simulation. The first five work cycles followed the same trace; however, the plots began to diverge when the shallower trenches were completed and the simulation reset to the initial trench depth. Env1 took four work cycles, Env2 took five work cycles, and Env3 took seven work cycles.

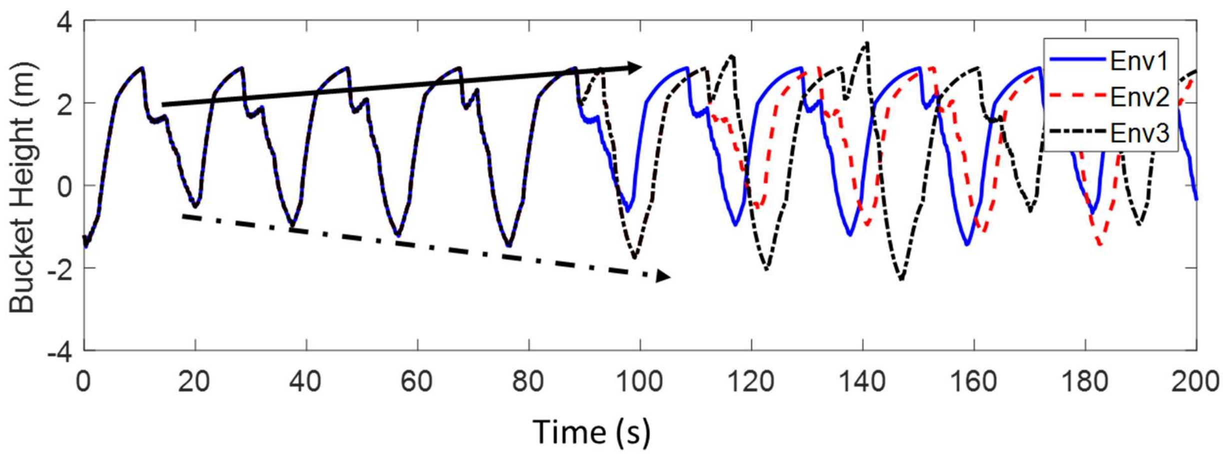


Figure 27. Machine response to different environment information with indicating of the changing height of the pile and depth of the trench.

Figure 28 represents the swing angle of the arm-boom-bucket assembly between the trench and the pile during an operation with different environment settings. It took a longer time to reach the larger swing angle. The larger swing angle introduced a larger error signal in the controller resulted in larger acceleration in the swing speed. The arrows in Figure 28 indicate the different swing times needed to reach the pile, given the different pile locations.

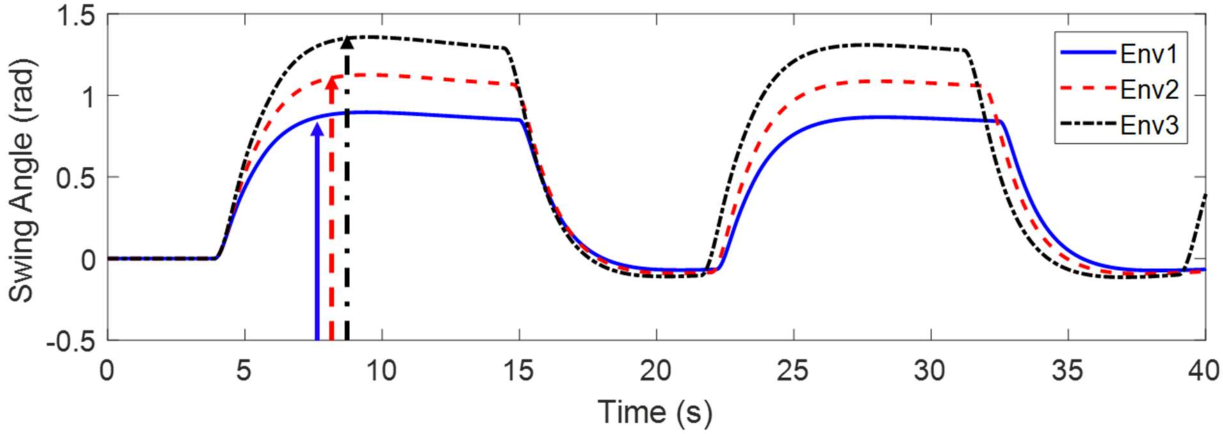


Figure 28. Machine response to different environment information with indicating of Different Swing Angles

Case Study3: Adaptation to Different Excavator Models

Three excavator models with different dimensions were simulated using the same VOM. Figure 29 demonstrates how the orientations of the boom, arm, and bucket compare between three different excavator models compare at different stages of the first work cycle.

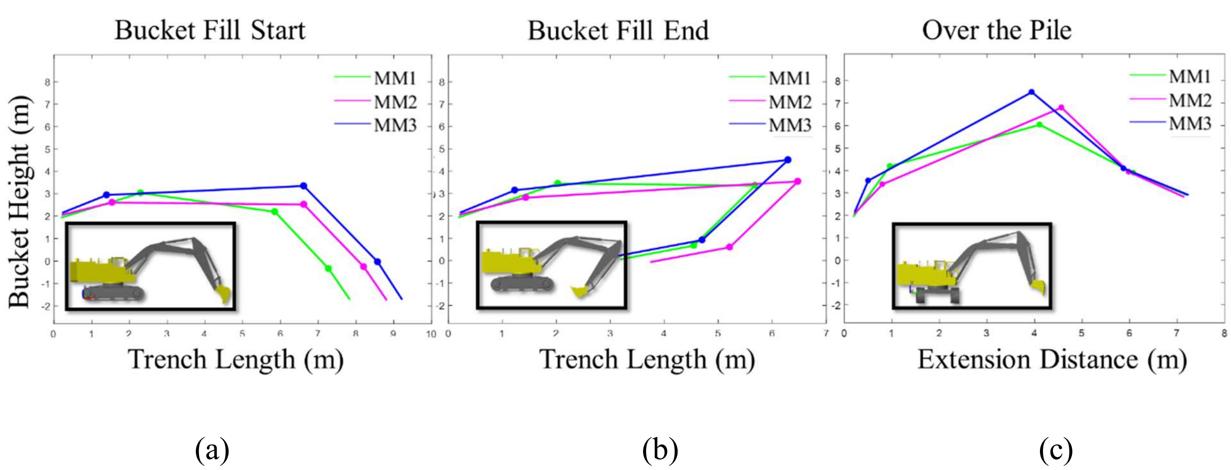


Figure 29. The boom, arm, and bucket positions for different excavator models at (a) start of Bucket Fill, (b) end of Bucket Fill, and (c) over the pile.

Figure 29a represents the different combinations of boom, arm, and bucket for the three excavator models at the start of the Bucket Fill task. The initial arm angle, bucket angle, and the bucket height were identical for three different excavator models at the start of the simulation. Since the excavators are of different sizes, the length of the trench they each dig is different. Figure 29b shows the different combinations of Boom, Arm, and Bucket for the three excavator models at the end of the Bucket Fill task. The three excavator models ended the bucket fill task with similar arm angle, bucket angle, and bucket height. This was expected as the location of this part of the work cycle is not dependent on the target trench depth or pile height. Finally, Figure 29c showed how the boom, arm, and bucket of the three excavator models moved to different positions to reach the 2-meter height of the pile. The three excavator models reached the same location with different configurations of the boom, arm, and bucket since the three excavator geometries (e.g. arm length) were different.

The total time required for the VOM operating different excavator models to reach the same trench depth (2.0 m) varied from machine to machine from 136 to 190 seconds (Figure 30). This results demonstrated that even though the machines were different in terms of their geometry, inertial properties and trench length digging capabilities (resulting in different cycle times), the VOM was able to adapt to these differences and operate the machines through multiple varying work cycles.

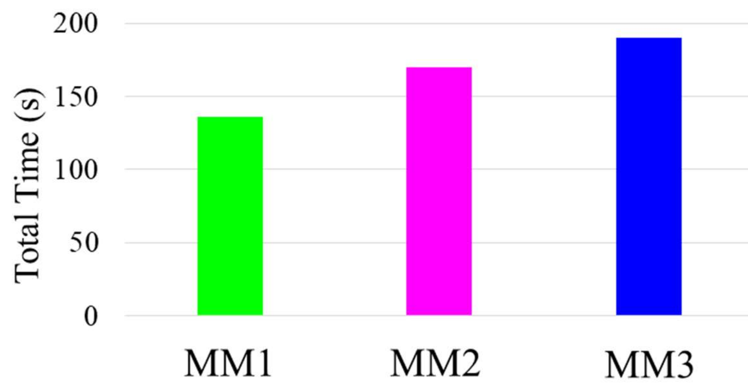


Figure 30. Total operation time to dig a trench to the depth of 2.5 meters for three excavator models.

The individual bucket teeth trajectories of the three excavator models during all the work cycles in a complete trenching operation were similar (Figure 31). The increasing trench depths and pile heights during the operation demonstrates the ability of the VOM to adapt to different excavator models under dynamic environment changes during the simulation.

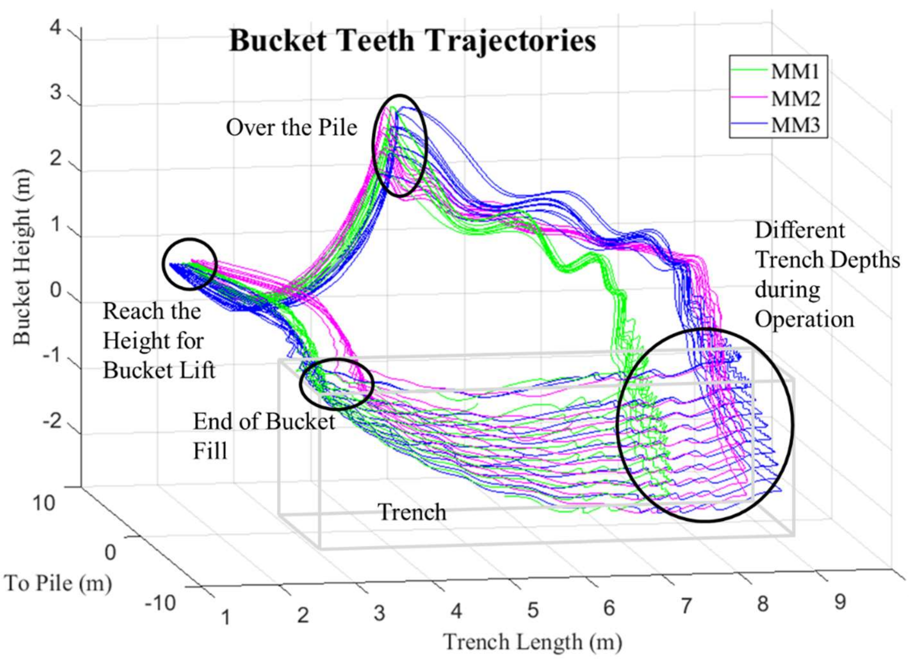


Figure 31. Bucket trajectories of three excavator models with different positions to start Bucket Fill, Bucket Lift, and Dump, and to end Bucket Lift

Case Study 4: Comparison with Human Operator Data

The VOM operated an excavator model for which we had measurements of cylinder lengths while being operated by a human operator. These measurements enabled a comparison of the trenching operation by the VOM and the human operator. Observing the traces of bucket height as a function of time, similar patterns were observed; however, the cycle periods for the human operator were shorter than those produced with the VOM (Figure 32a), most likely due to the fidelity of the machine model (e.g. hydraulics model). When the time axis was scaled by a factor of about three, then the bucket height trajectories followed the same general shapes (Figure 32b).

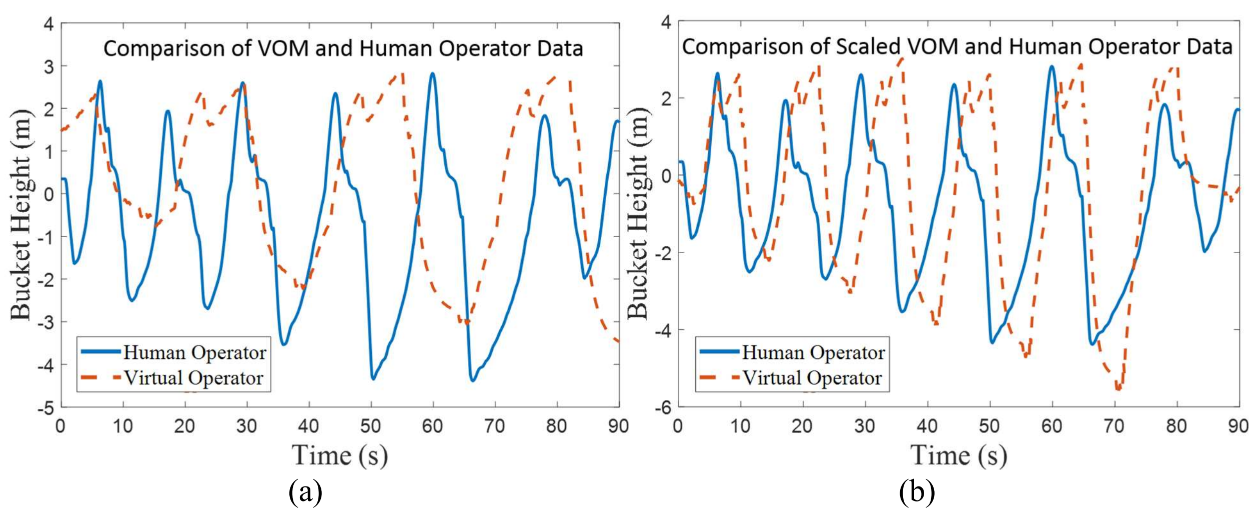


Figure 32. Comparison between VOM-generated and human-generated data, where (a) is the original traces, where differences in work cycle time is likely due to the fidelity of the machine model, and (b) where the work cycle time of the VOM data is compressed to show that the operation traces have the same shape.

Discussion

This work represented expert operation by enabling tasks in the work cycle to overlap. Different human operators can perceive and act upon cues from the machine and environment differently, depending on their level of expertise. Expert operators can find efficiencies by starting and ending tasks differently than novice operators, resulting in overlaps between tasks. Human operators tend to perceive the environment in relative terms (e.g. the bucket is near the end of the trench), rather than in absolute numbers (e.g. the bucket is extended 2.2 meters from the cab). The fuzzy classifiers therefore model human perception more closely when membership functions are based on relative signals rather than absolute ones. This work used the relative positions to construct the classifiers to enable the perception of both the start and end of tasks. This enabled tasks to overlap, which matched what was observed from the recorded data from real operations and operator interviews (Du et al., 2015). The degree of the overlap between tasks can be used to differentiate a human operator's expertise. In this work, we simulated a consistent level of an expert operator, although in future work this could be varied. It may be possible to manipulate the level of expertise in demonstrated by the VOM. A beginning machine operator, someone with little expertise will tend to operate a machine through the tasks of a work cycle in a way that the tasks are completed in serial, with no overlaps of sequential tasks (or 0%) in the work cycle (Yu, Dorneich, & Steward, 2016). However, as operators improve their skill or expertise in how the machine operates (they get a "feel" for the machine meaning they come to understand how the machine responds to their input because of the machine dynamics and kinematics), they will start to increase the overlap between tasks, which we found in operator interviews. Overlap implies that the operator is still attending to the end of the current task, they are also starting to attend to the next task and thus while continuing to give commands to finish the current task, they are also starting to give the commands for the next task as well. So increasing % overlap (= time that tasks are overlapped with in a work cycle/work cycle time * 100%) is a measure of increasing levels of a expertise that an operator possesses. The tasks in Case Study 1 overlapped on average 30% of total cycle time, which implies an approximately 23% productivity increase compared to the operation without overlap tasks. Simulation of task overlap enables the VOM to more accurately reflect real operator performance.

Note that there will be an upper bound on % overlap, because there are parts of the work cycle that must be non-overlapping. For example, when the excavator is swinging from the trench to the pile and then when swinging back from the pile to the trench, there is no possibility of overlap and the time in that part of the work cycle will be determined by the angle through which the machine swings and the average velocity with which it swings. Based on our analysis of two expert excavator operations, we found that overlap ranged from 20% - 60%, depending on the machine, the work site configuration, and which work cycle in the operation. Our estimates of overlap will typically be in this range, depending on the combination of machine and work site configuration.

Adaptability was demonstrated with the results of Case Study 2 and 3. With environment model, the VOM adapted to changes in environments and adjusted control inputs accordingly, which lead to the simulation of work cycles that change during a trenching operation. Human operators interact with the environment and machines during the operation all the time. They use a mental model to accommodate the information from the environment and machines, which are constantly updated as they work through the work cycles. An environment model was developed as an operator's continuously updated mental model of the changes in the environment, enabling the VOM to adapt the target reference commands as the work site environment changed between work cycles. The results reveal that the number of cycles and the amount of time needed to reach different target trench depths and pile heights changed, as would be expected if the VOM adapted the changing work site environment conditions.

The initialization file contained the worksite specifications (pile and trench locations, maximum pile height and maximum trench depth, initial strategy, and excavator geometries). This file represents the initial conditions and initial plan for the operation, much like a human starts an operation with initial knowledge and plan. The initialization file along with the environment model act as a mental model to create a general understanding of the work site, the plan, and the machine.

A human would also have a general knowledge of how to operate a machine. When a human operator then switches between one machine to the next, he or she applies that general knowledge and then translates it to specific updates on the control inputs. A level of generic knowledge of how to operate a machine is important to enable the adaptation to different machines. The human operator applies the generic knowledge to a specific machine, when they move from one machine to another. By using the generic knowledge and general understanding about the environment and machines, the reference commands were adjusted by the VOM to accomplish the operation with the new machine. To simulate different machine models only the initialization file was updated with machine model's geometric information of each component, and no modification was needed for the VOM. Different machine models were simulated by the VOM without tuning need to tune the VOM. The power source for different machine models were currently set to the same value, and so the larger machine responded more slowly. The larger machine is expected to dig more slowly, dig more dirt, and dig a longer trench, which can be seen in the results. Although the VOM was developed based on the human perception and decision-making process, it also generated similar trajectories of operation as the trajectory following modeling method.

Differences between the VOM-generated and human-generated operations were due to when the bucket was rotated, as the VOM tended to rotate bucket near the peak of the bucket height curve where the human operator rotated the bucket when the bucket was being lowered after the peak height. The bucket height was defined to be the height at the bucket teeth, so the "plateaus" in the curve were due to bucket rotation relative to the entire bucket mechanism being lowered. The other difference in the cycle space mentioned in the results, in the speed of cycle, was most likely due to the limited fidelity in the machine model, particularly in the hydraulics models.

More research is needed on multiple aspects of the work described here. To address the expertise representation several aspects can be investigated. It is important to know how different skill levels impact the proportion of or decision to overlap tasks. A more nuanced understanding of how expertise is realized will enable VOMs to simulate different levels of skill. In the current work, we assumed that the virtual operator attended to all the cues provided by the human perception module. But in real operations, there is a level of uncertainty in the perception of all available information. For instance, as humans become fatigued, they start to miss information or their attention becomes increasingly narrowly focused. Future work would model some level of information perception uncertainty, perhaps depending on a model of attention that could be affected by operator fatigue, environmental noise, or distraction.

The environment for the construction site can be described in many ways. For excavation, the machine interacts with soil, which can have impact for operators' strategies and machine performance. A soil model can enable testing of a excavator model while digging different materials. Weather conditions are another important environment factor, which can impact operations.

Human operators learn how to operate their machines over time, building up not only expertise but strategies and specific decision points of how to operator a particular machine model for maximum productivity. Future work could explore how to develop a VOM that can iteratively simulate this learning process over time to arrive at an optimal control strategy for a given machine model.

Conclusions

This work focused on improving VOM fidelity by representing human expertise and human adaptability to different worksite environment and machines. The VOM was based on the way human operators operate machines based on how they perceive signals, how they understand the environment and machines, and how they adjust their controls for adaptations.

From this work, we can conclude that the representation of human operator expertise and adaptability has several requirements. Modeling expertise requires modeling operator shifts in attention from tasks that are nearly completed to those that are next in the cycle sequence, Human operator tendencies to perceive machine and work site cues is processed in fuzzy and relative abstractions. Finally, operators use mental models of the current work site state indicating the degree of progress made in completing the operation.

In addition, once a VOM incorporates these aspects of human expertise, investigations into the behavior of the closed, human-in-the-loop system can be initiated resulting in useful observations into a dynamic full operation with different machines. The capabilities of the VOM developed in this work are essential to advance VOM model fidelity to the point where designers can rapidly test design iterations virtually. By enabling the VOM to represent expert behavior, the simulation can push the machine model to its limits. Currently test operators can push machines during productivity tests by exploiting all the capabilities of the machine. By more accurately representing human expertise in a VOM, design engineers can be more confident that model-based simulations more accurately reflect what human operators can achieve with the machine. Furthermore, by building a VOM that can adapt to changes in the environment, complete operations can be simulated, further enhancing the utility of model-based testing. Finally, the ability of the VOM to adapt to different machines without time-consuming re-tuning is essential to enabling the rapid design iterations. The design engineer's time will be spent on iterating the machine design, rather than tuning the VOM to test a particular machine design. This work is a step towards the vision of developing VOMS with a fidelity that matches the current fidelity of machine models.

The work here has several limitations. This paper presents a proposed model of a virtual operator that needs to be compared to human performance, and is an area of future work. The comparison shown for case study 4 demonstrated that the bucket height trajectory traces were of the same shape between the VOM and human-generated data.

However, validation of the model will require a high-fidelity machine model, and likely improvements to other areas of fidelity. This work has thus far focused on three aspects of expertise that emerged from our interactions with operators and engineers in industry, but there are many ways that that expertise is manifested in construction machine operators. On-going work, building upon this work, seeks to further model other aspects of expertise, such the ability to learn. In addition, expertise would also include being able to make adjustments to exception cases, like running into a boulder while digging a trench.   This type of expertise would require higher level decision making processes that were beyond the scope of the paper. The fidelity of the model can also be improved by accounting for human performance moderators such as attention limitations and fatigue.

Acknowledgements

The authors would like to give appreciation to the following people for their contributions to this work: Eric R. Anderson, Lawrence F. Kane, Brian J. Gilmore, and Tristan Griffith. This work was funded by Deere & Company. The opinions expressed herein are those of the authors and do not necessarily reflect the views of Deere & Company.

CHAPTER IV: MODELING LEARNING IN VIRTAL OPERATOR MODELS

Material in this chapter accepted as a journal paper:

Du, Y., Dorneich, M. C., & Steward, B. (accepted) Development of A Learning Capability In Virtual Operator Models. International Journal of Commercial Vehicles.

Introduction

Model based design improves the product development process with reduced time and costs. The combination of virtual operator model (VOM) and machine model simulation introduces a more rapid and lower-cost strategy for testing and screening possible machine design alternatives. This model-based design approach can reduce the product development cycle and cost, enabling a more intensive exploration of the design space. Traditionally, machine testing and validation employs human operators operating physical prototypes. *Closed-loop simulation-based design capability* is the capability of design engineers to simulate models of the entire construction machine system which includes the machine and the operator. A VOM is designed to represent the control behaviors of human operators, resulting in machine model simulations which are more similar to physical machine operation with a human operator.

Human operators use different strategies depending on the machines, environment, and skill levels. In this paper, we define a *strategy* as the combination of control methods and the timing of the transitions between control methods to accomplish a task. *Control methods* are defined by what feedback signals the operator is using when adjusting control inputs (i.e. speed, position, etc.). *Control inputs* are the moment-to-moment inputs from the operator to the vehicle controls to move the machine elements.

Examples of operator strategies can be found in the excavator trenching operation. Trenching under softer soil conditions requires different strategies during the bucket fill task than under conditions with larger aggregates such as gravel or crushed rock. Under this latter condition operators will typically scrape along the bottom of the trench with the bucket uncurled and then curl at the end of the scraping process. Under the former condition, operators tend to remove material with a scooping action in which the bucket is curled throughout the digging task.

Another strategy, speed control, aims to control the vehicle motion through controlling the speed of controllable components of the vehicle. For example, in the case of an excavator, operators can control the swing speed of arm, boom, and bucket assembly. The speed control strategy can be used to swing the assembly to the pile, or can be used when digging in soil, since soil has high viscosity and the soil is not likely to fall from the bucket. When digging in rocks, however, both force and speed need to be monitored and controlled, since rocks are heavy and it is easier for them to fall out of the bucket due to larger inertia.

The current state of art in virtual operator modeling reveals gaps between human operators and the virtual operator models. Human operators differ from one another having various background experiences and skill levels. Human operators learn through repetition over time to determine which strategies work well and make adjustments within those strategies to achieve their goals. It can be time-consuming and quite difficult to learn what strategies human operator use for machine operations.

Interviews with operators can reveal strategies, although expert behavior is often represented abstractly and performed so automatically that the experts themselves have a hard time articulating exactly what they do (Ackerman et. al., 2003). In previous work, extensive interviews, observations, and data analysis were used to learn the strategies expert operators used to conduct a trenching operation (Du, Dorneich, & Steward, 2016).

Existing operator modeling efforts approximated human operators control behavior using trajectory mimicking to approximate human operator behavior, which resulted in a trajectory similar to the recorded data from a human operator. In previous work, we developed a VOM based on the human information processing model (Newell & Simon, 1972), but it did not differentiate different skill levels, and used a fixed strategy for the same operation (Du, Dorneich, & Steward, 2016; Du, Dorneich, & Steward, 2018). Human operators are unique due to their learning capability, which can impact their operation strategy during operations. Current VOMs use a fixed strategy, and do not demonstrate learning capability as human operators, which limit their ability to represent expert operator behavior when used with an unfamiliar vehicle model in a close-loop simulation. For each new machine model, the VOM (much like a human operator) should learn over time the best control strategies given the machines capabilities and limitations, work site, and the goals of operations.

The current VOM can represent certain a generic level of human expertise, and the human ability to adapt to different worksite environments and machines (Du, Dorneich, & Steward, 2018). With this capability, a VOM can only operate the machine model with pre-defined control methods, which may not be the most efficient or safest way to perform the tasks in a work cycle. On the other hand, human operators learn from repeated operation of a machine the best combination of control strategies and parameters. For example, a power-limited machine may require more active control to guide the vehicle to complete the tasks than a highly-powered machine where the operator can utilize large accelerations and the inertia of the machine to optimize task time. The challenge is to replicate the learning process of a human operator to enable VOMs to learn the best control strategy for each new machine model.

The best strategy to operate a new machine is not known *a priori*. Human operators use experience and trial and error to learn the best strategy over time. The best strategy is identified through a learning stage where operators experiment repeatedly with the same task. Similarly, by integrating the learning capability, a VOM is able to learn the characteristics of a new vehicles and choose the best strategy to most efficiently accomplish the tasks in a closed loop simulation of the operation. This process can increase the confidence that a VOM drives the machine the way an expert operator would in a productivity test with a real prototype. The fidelity of closed loop simulations of VOM-Vehicle systems can thus be improved. *Fidelity* is the faithfulness with which a model represents the behavior-of-interest of a physical system. With high fidelity VOM-Vehicle simulations, model-based design can be applied to more aspects of design processes, which can reduce product development time and cost. Determining the best strategy for an operation is a complex process, which depends on the machines, environment, and human operators' skill levels. Therefore, it is necessary to learn the best strategy rather than pre-defining it. Identifying the best strategy through a learning process and then using this strategy in closed loop simulation will lead to more realistic simulation of human experts.

The excavator trenching operation was selected for operator modeling, as it is one of the most common construction activities in the construction site. An *operation* is defined as the job that needs to be accomplished, such as digging a trench. An operation is comprised of multiple *work cycles*, which are repeated *tasks* in sequence. Tasks are the specific activities needed to be performed during each work cycle. A trenching work cycle has five tasks in sequence: bucket fill, bucket lift, swing-to-pile, dump, and swing to trench. A hallmark of expert operator behavior is to overlap the end of one task with the start of the next task (Du et al., 2016). Human operators learn the capabilities of machines during operation and practice. Examples of the types of things operators learn when transitioning to a new machine are the maximum amount of material that can fill the bucket without material spilling out, or how to provide different control inputs to swing the bucket to dump area efficiently and stop the bucket in the dump area.

A skill is learned over time, and humans acquire knowledge or skills through experience, study, or training (Adams, 1987; Pear, 1927). For excavator trenching, learning means specifically the process of acquiring knowledge or skills through experience. According to the operator interviews (Du, Dorneich, & Steward, 2016), in real construction site operations, human operators often learn to maximize the speed of the excavator tasks to maximize productivity of the operation. Productivity is often the most important factor when judging the performance of human operators (Du et al, 2015). Learning is a unique capability of human operators, enabling human operators to become experts over time. Operator skill levels are advanced continuously through learning.

Operators can improve efficiency by changing control strategies during tasks. Operators may determine different strategies for the same operations in different environments and with different machines. For instance, the operator may accelerate the bucket quickly when swinging it to the pile, but ease back on the controls and let the inertia finish the swing to stop the bucket over the pile. For different machines, the operator may actively decelerate the bucket near the pile, if the swing speed is too fast, then the bucket will overshoot the pile, and the operator will lose time to bring the bucket back over the pile to dump the material. Through multiple operations, human operators learn machine limits and the best way to use machine capabilities to minimize task time.

This research developed methods to enable a virtual operator model (VOM) to learn the optimal control inputs for operation of a virtual excavator. The following section reviews related work in learning methods as applied in modeling. The approach section reviews the VOM modeling architecture, and introduces more detailed information about the way learning was investigated in the VOM. The methods section describes two design iterations of the same vehicle that were developed for use in two test cases to exercise the learning capability. The chapter concludes with the results, discussion and conclusions.

Related Work

To evaluate human-machine system performance, human learning capability is one of the characteristics of human operators may affect the overall performance. The product development process can benefit from the learning capability of a VOM. To investigate realization of learning capability, methods were found in the autonomy area, which were developed to enable the learning in the environment.

Model-based design in product development

As more advanced and complex systems are integrated into vehicle designs, the traditional methods of product testing became time and cost consuming. Therefore model-based design has become increasingly important in the automobile and off-road machinery industry to enable a time and cost-efficient product development process (Filla, 2003; Zorriassatine, 2003). Model–based design is using computer technology to develop models and simulations enabling virtual testing and validating of product designs (Zorriassatine, 2003). Sub-systems as well as complete vehicles are being simulated to mostly evaluate many aspects such as durability and fatigue analysis, dynamic analysis, safety analysis, noise, vibration and harshness (Zorriassatine, 2003). Originally, vehicle performance analysis required physical prototypes with human operators. With the development of simulation capabilities, physical prototypes were replaced by virtual porotypes, but testing still required human operators. Our work aims to develop virtual operator models to replace the human operator closed-loop simulation. This will enable developers to analytze efficiency and performance earlier in the design process, and reduce product development cost. Filla introduced a revised design process by implementing the simulation in the product development process. The modifications were required when the design targets were not satisfied in the simulation. Aoyama and Kimishima (2006) proposed a method to evaluate the designability and operability of the product, which used mixed reality using a physical control system to control a virtual prototype.

Trajectory optimization and obstacle detection

The classification of machine tasks is very important for modeling of machine operations and provides detailed information of machine activities during an operation. Modeling the learning process of human operators can utilize classification information of machine states as inputs. Akhavian & Behzadan (2015) presented a classification method for machine actions by collecting data through sensors such as global positioning system (GPS), accelerometer, and gyroscope with different algorithms. With these data, they which aimed to accurately classify the machine actions into tasks to improve reliable operational decisions (Akhavian, & Behzadan, 2015).

Another method used widely for classification is conventional neural networks. This method has been used to recognize the environment and detect obstacles, which is needed to adjust the trajectory during trajectory following. Convolutional neural networks (CNNs) have been applied to detect and classify the presence of obstacles (LeCun, 2006; Hadsell et al., 2009) and the location of the ground (Bojarsk et al., 2016) through data collected via image processing. These efforts utilized images recorded from the environment with labels to train the CNN, which were the used to classify new data. The authors describe the use of CNN as part of the learning process to recognize elements of the environment. Hadsell et al. (2009) focused on the long-range vision system, to identify obstacles from long range. Bojarski et al. (2016) focused on directly sending CNN classification results for use in generating steering for the self-driving car.

Hamner, Singh, and Scherer (2006) investigated methods of avoiding obstacles in autonomous mobile robots based on the observation of human operators' driving behavior. A trajectory planning method was developed to determine the trajectory of the vehicle by detecting the obstacles. Hamner at al. used a genetic algorithm to determine the parameters for the trajectory planning, which were realized in trajectories that successfully avoided the obstacle.

In simulation, optimization of operations has typically been done via mathematical optimization of trajectory for environments. Given a starting and ending point, optimization will determine the optimal path, given any obstacles detected in the environment. However, expert operators often outperform automated trajectory-based control systems, due to the application of strategies based on experience, and recognition of the environment (Du, Dorneich, & Steward, 2016). In our approach, we developed the VOM to model the human-level perception and decision process of human operators to mimic human operator behavior. So, the VOM reasons at the level of optimizing between strategies rather than a pure trajectory optimization.

Learning based on operator behavior

Albus et al. (2007) integrated learning of the terrain in the environment using image processing and the generated behavior of the vehicle control system. The system included learning in three ways: learning from example, learning from experience, and learning to optimize trajectory. The system determined operator behaviors, based on actual evaluations by human. The human classification results were used as ground truth to train the classifier. Along with the model of the world and goals of the task, behaviors of how to drive over terrain were generated, which guided the vehicle to follow certain series of waypoints. Autonomous operation then relies on signal inputs from sensory processing, which was considered a perception process, and the selection of behaviors from the vehicle control system. The learning in this paper mainly focused on the recognition of the work site environment. The updated work site environment information was used to determine the appropriate behavior to control the vehicle. Similar to our approach, they model included human to develop an classification process human agrees. However, they do not integrate the actual operational human behaviors to control the field robotic.

Approach

Architecture

A VOM was developed which integrated of human expertise and adaptability to worksite environment and machines (Du et al. 2018). The VOM structure consisted of four modules, which were the kinematics, perception, decision making, and action modules (Figure 33). The kinematics module translated the signals from the vehicle model into human perceivable information for decision making. The signals from the vehicle model were cylinder extension length and velocity, which were translated into the absolute position and orientation to describe the exact positions for the elements of the vehicle like bucket and its rotations. These translated signals relate to the visual cues for the human operator's decision-making process during operation.

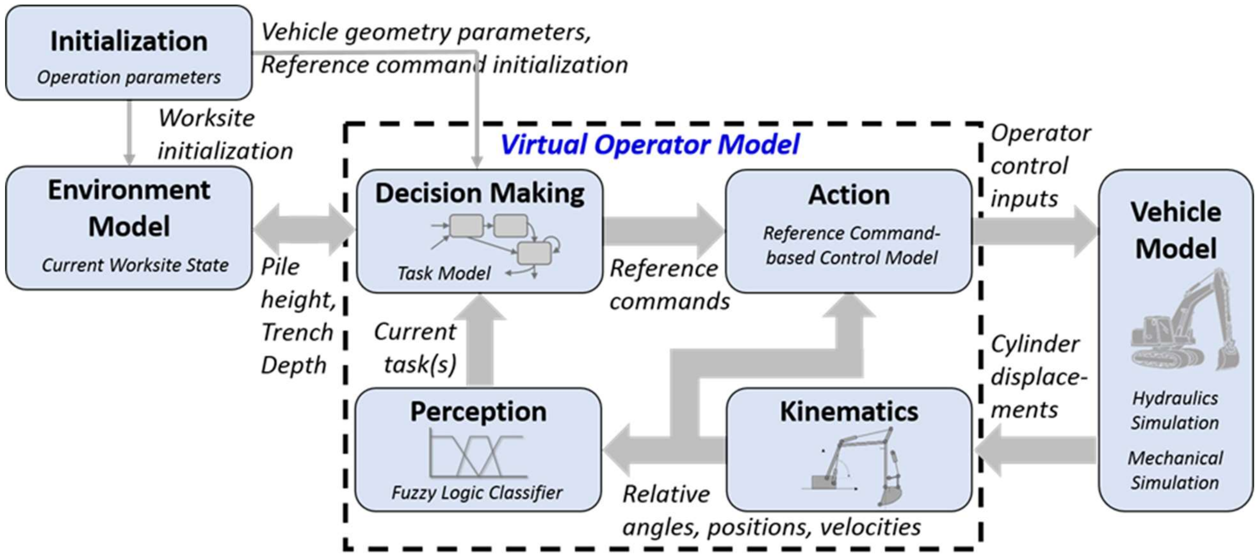


Figure 33. The Virtual Operator Model Structure

The human perception module received information from the kinematics module, and detected task transitions between different tasks using fuzzy classifiers. The state machine used the outputs of the classifiers to determine the current state. The reference commands were determined in the human decision module by considering environment information, machine geometry and the current state. The human action module used the reference commands as target positions of vehicle components, which were compared to the current positions of the vehicle components to provide errors for feedback controllers to generate control inputs for vehicle model.

Integration of a Learning Capability into Virtual-Based Design

The learning capability can be integrated into the model-based design process to test the productivity of a new vehicle model (Figure 34). For a new vehicle model, the best strategy is not known for the operation. The best strategy needs to be determined prior to run a productivity test with the closed loop simulation. This work happens in the Learning VOM step, which is performed by using genetic algorithm to learn the best strategy for specific tasks in the operation. The best strategy is parameterized in the initialization file for the VOM, which is used by the VOM at the beginning of the complete operation simulation to set the strategy for the closed loop simulation. Finally, the traditional closed loop simulation of the VOM-VM combination will use the best strategies in the course of simulating the work cycles of the operation. The results from the simulation can then be analyzed to inform the next iteration of the VM design, and the entire process can be repeated.

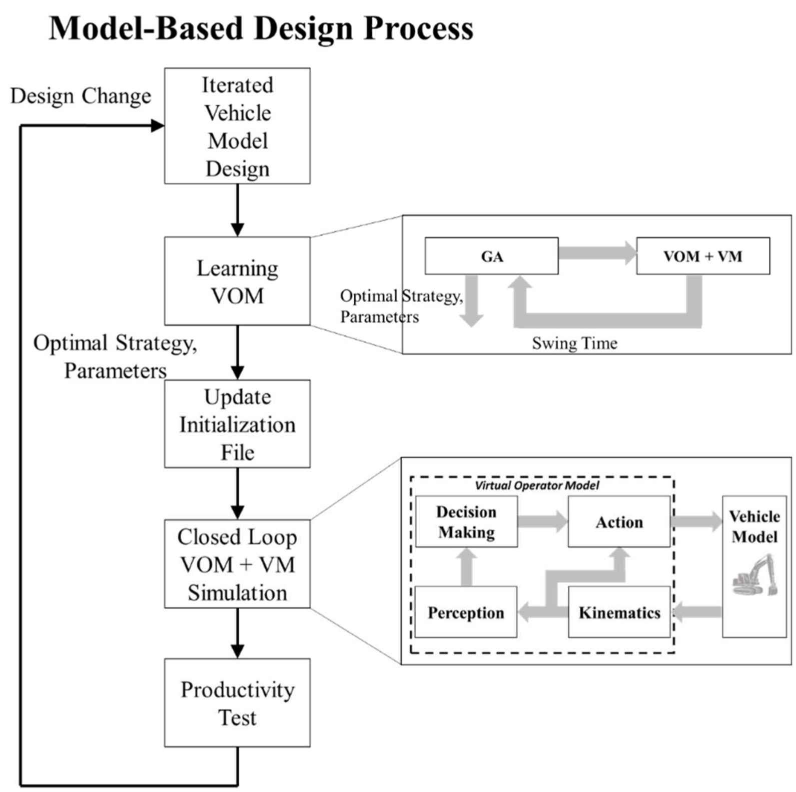


Figure 34. Procedure to Use Learning Capability for Productivity Test for the New Vehicle Model

Learning approach

The swing-to-pile task is one of the tasks in the trenching operation where different strategies of applying control methods can greatly affect efficiency. Therefore, the swing-to-pile task was chosen as the task to develop the learning capability into the current VOM. Parameters refer to different swing angles, which trigger different control methods. During the swing-to-pile task, the operator rotates the cab to swing the bucket from the trench to the pile, and simultaneously extends the bucket to reach the pile, and raises the bucket over the pile. Human operators aim to minimize the time for this task to be efficient, since it can be the major factor to rate the overall work cycle performance, according to the operator interviews (Du et al, 2015). Human operators often start by giving the largest control input to maximize the swing speed until the vehicle rotates certain angle of the total target rotation, then human operators zero the control input for swing and let it coast via inertia, and then make final adjustments to reach the full target swing angle by providing control input to position the bucket over the pile. If they reach the pile with too much velocity, they overshoot the pile, the operator must reverse the control inputs to guide the vehicle back to the pile location.

Based on this generic strategy, three phases of control methods were identified for human operators: speed control, coast, and position control. *Speed control* is the control phase that the Action Module provides control inputs and guided the vehicle model to reach the target speed. *Coast* is defined as the control phase that no control input is provided to the vehicle model, and the vehicle model swings about a horizontal axis with inertia. *Position control* is defined as the control phase that the Action Module provides control inputs and guides the vehicle model to reach the target swing angle. Other control methods are possible, but for the purposes of this work, these three control methods were modeled. The control strategy variables are the rotation angles of the swing where the transitions between different control methods occurs.

Human operators learn over time, through repeated experience with a particular machine and interaction with the work site, the most efficient combinations of these control methods to conduct the swing-to-pile task. To replicate human operators' the learning process for swing-to-pile task, the combination of control methods, and the parameters of when to make transition are identified to result in minimum swing time. The search for the best combination of control methods and timing was then formulated as an optimization process. This work investigated the way to conduct the optimization to replicate the outcomes of the learning purpose.

Method to enable learning scenarios

To enable learning using the VOM, several enhancements were needed in the VOM architecture. New control methods were integrated, since the current VOM only used position signals for PID controller to control the vehicle model to reach the target swing angle. The speed control and coast control methods were added to the action module. In the decision-making module, logic to transition between control methods were added.

The VOM was integrated with an optimization platform to enable rapid simulation of the learning process that for humans happens over many years. This optimization platform simulated the predefined tasks repeatedly until it learned the optimum parameters and strategies, using a combination of Matlab script and Simulink models. The learned parameters and strategies were then used for the full operation simulation.

Materials and Methods

The learning scenario was defined for the excavator trenching operation. Two iterations of a vehicle design were developed to compare the results from the optimization method for the learning scenario.

Work Cycle

Based on the operator interviews and operation observations, multiple work cycles are needed to accomplish the trenching operation. In real operations, expert operators are able to overlap two or three tasks during the work cycle (Figure 35). For example, the human operator may overlap bucket lift, swing-to-pile, and dump tasks.

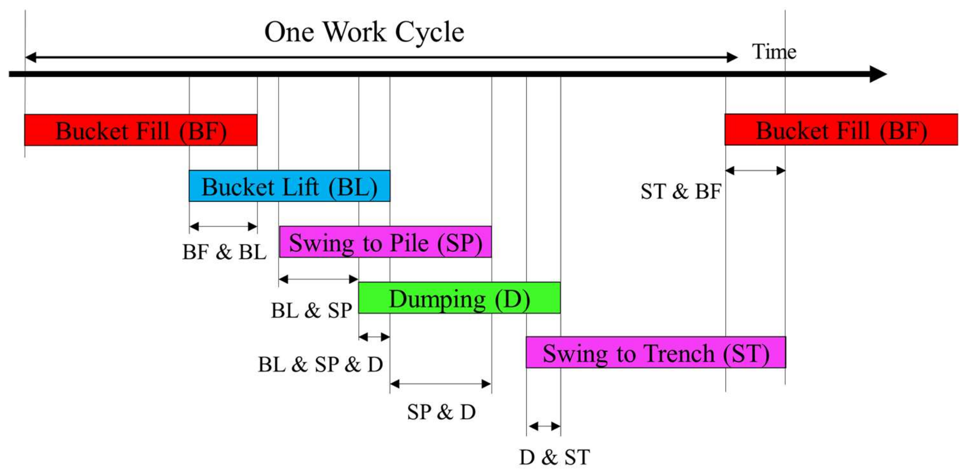


Figure 35. One work cycle of excavator trenching with task overlap.

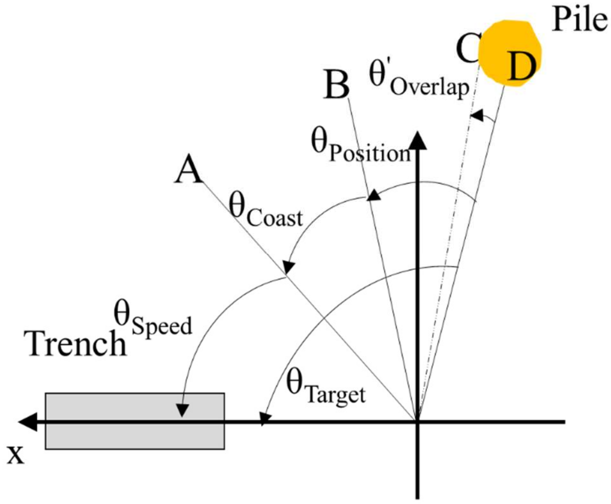
Control Input Strategies and Parameters

The learning scenario realized in this work is aims to find the best combination of control methods for the swing-to-pile task. Three different control methods were defined for the learning scenario: 1) Speed control, 2) coast, and 3) position control.

The learning scenario was defined based on four swing orientations (Figure 36). The swing orientation labeled “A” indicates where the coast method is triggered, and

θ_{Coast} is the range of swing angles where no control inputs are sent to vehicle model. The B orientation indicates where the position control was triggered, and $\theta_{Position}$ is the angle range were position control inputs are generated. The orientation C indicates where the dumping started, which was also the start of the overlap state, and $\theta_{Position}$ was used to represent the angle for the overlap state until the bucket reaches the pile. The D orientation is where the bucket is over the pile location. θ_{Speed} is the angle for the speed control. θ_{Target} is the angle defined by the pile location and trench, which represented the target angle through which the vehicle needed to swing.

The operator starts to command swing-to-pile when the bucket is level with the ground surface. When the bucket reaches the boundary where the pile is defined, the operator commands dumping of the material. The operator simultaneously dumps material while completing the swing, so the tasks overlap. Due to the nature of the fuzzy classifiers, the overlap of task swing-to-pile and dumping was triggered when the bucket reached the leading boundary of the pile. The operator dumps the dirt in the pile area, therefore the dumping task contained the commands to guide the vehicle to be at the pile. To represent this in control, dumping task enforced position control in the overlap state to enable to dump the dirt in the correct position. Once the overlap task started, the control input strategy transitioned to position control, since dumping task guided the bucket to be in the area where the pile was.



Variable	Description
A	Start of Coast
B	Start of Position Control
C	Start of Swing + Dumping
D	Pile Location
θ_{Coast}	Angle for Coast
$\theta_{Position}$	Angle for Position Control
$\theta'_{Overlap}$	Angle for Swing + Dump
θ_{Speed}	Angle for Speed Control
θ_{Target}	Target Swing Angle

Figure 36. Positions and angles of the swing-to-pile task.

In the learning mode, the Learning VOM only needed to simulate the swing-to-pile task, since this was the task under investigation. The simulation of the swing-to-pile task was stopped when the dumping task was initiated. If the bucket never reached the pile because the velocity reached zero too soon, the simulation stopped after a time threshold was reached. If the swing completed, but the velocity was not minimal over the pile, the operator would normally overshoot the pile and have to navigate back to the target swing angle. Thus, the complete swing task should include the swing time for overshoot and time to swing back to the pile. Since the simulation was stopped upon the when the bucket first reached the pile, a penalty was used to determine the complete time for the swing task that included the time to swing back in the case of an overshoot. The penalty was calculated based on the swing characteristics of the vehicle models.

The learning mode was designed to find the best combination of the coast angle (θ_{Coast}) and position angle ($\theta_{Position}$) to represent the best transition locations between different control methods in order to complete the task with shortest time. After θ_{Swing} was defined, θ_{Speed} can be calculated by the relationship,

$$\theta_{Speed} = \theta_{Swing} - \theta_{Coast} - \theta_{Position} \tag{5}$$

A genetic algorithm (GA) was employed to optimize over θ_{Coast} and $\theta_{Position}$ to determine the combination that would accomplish swing-to-pile task in the shortest time. The GA manipulated the different combinations of $\theta_{Position}$ and θ_{Coast} to determine the best combination with the least swing time.

Genetic Algorithm Components and Operators

Genetic algorithms (GA) search for the best parameters in a simulation to generate optimal results (Carson & Maria, 1997). A GA was chosen to optimize the timing of the transition between operator strategies during the swing-to-pile for since the GA can search the space thoroughly without a priori knowledge of the shape of the curve, which is needed for numerical approaches to avoid optimizing local minima. A set of solutions is generated from different combinations of parameters based on a fitness function. The next generation is then created from a combination of the solutions from the first generation. This continues until the best parameters are found to satisfy a fitness function. There are several genetic algorithm parameters that must be set up before optimization can proceed. The parameters are described below (Davis, 1991):

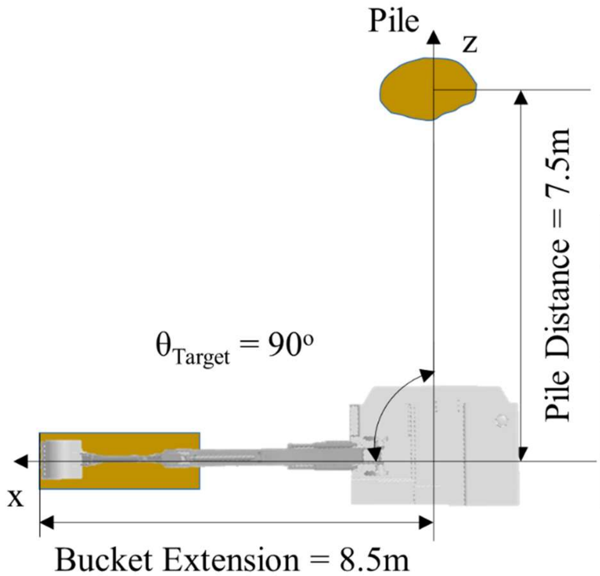
- Chromosome: The chromosome was used to represent the different combinations of the $\theta_{Position}$ and θ_{Coast}, and so was defined as [$\theta_{Position}$, θ_{Coast}]. Based on the problem set up for the learning scenario, the lower bound was [0, 0] and the upper bound was [1.6, 1.6] for the chromosomes.
- Population Size: specifies how many individual solutions are in one generation. With larger population size, the GA gives more possible solutions at the beginning of the

search, which is likely to avoid local minimas, but it takes much longer time for GA to find the optimal result. According to the rule of thumb utilized by Storn (1996), a population size of 10 was used

- Selection and Crossover: Since no previous knowledge was defined, the Stochastic uniform was used for selection. The crossover fraction was 0.8 to specify the next generation produced by crossover.
- Mutation: The mutation used Gaussian function to use random numbers from Gaussian distribution to create next generation.
- Fitness: The fitness for individual generation was used for optimization. The GA aims to find the minimal fitness value of different combinations of $\theta_{Position}$ and θ_{Coast}.

Case Study Definition

The case study describes two iterations of a vehicle design in a model-based design process (Figure 34). The work site geometry used to test the vehicle design iterations was defined by three parameters: pile distance, distance between pile and cab; bucket extension, distance between bucket and cab; and target angle, swing angle from trench to pile (Figure 37). For all productivity tests of the vehicle design iterations, the target swing angle was 90º, Different vehicle design iterations may respond to control inputs differently, and so human operators may apply different methods to achieve the most efficient swing.



Variable	Value	Description
Pile Distance	7.5 m	Distance between pile and cab
Bucket Extension	8.5 m	Distance between bucket and cab
θ_{Target}	90°	Target swing angle

Figure 37. Worksite Dimension for the Learning Scenario

<u>Performance Metrics</u>

The performance of the task was evaluated by the required to swing and be positioned over the pile so that material could be dumped on to the pile. Some time is required to swing from the trench to the pile, called swing time. If there is a non-zero swing velocity when the bucket reaches the pile, it will overshoot the pile and some time is required to return to the pile. This time was called the penalty time (Table 11).

Table 11. Task metrics.

Metric	Description
Swing time	Time to reach the pile
Penalty time	Amount of time to return to pile if overshoot
Total Swing Time	Swing time + penalty time

Multiple simulations were conducted to span 289 different combinations of θ_{Coast} and $\theta_{Position}$, where both angles ranged from 0° to 91.7° with step size 5.7 ° and from 0° to 45.6 ° with step size 2.7 °. Simulation results recorded from the swing-to-pile task simulation included the time when the simulation terminated, the penalty time for the overshoot, and the total swing time considering the penalty for the different combinations of θ_{Coast} and $\theta_{Position}$. Results were used to investigate the relationship of swing speed at the pile vs. swing angle, swing speed at the pile vs. time, and the distribution of the total swing time.

The penalty time was the product of the swing velocity when the bucket reached the pile and a factor used capture swing acceleration effects or:

$$T_{Penalty} = P \times \omega_{end} \tag{6}$$

where

$T_{Penalty}$ is the penalty time associated with the non-zero speed,

P is the acceleration effect factor, and

ω_{end} is the translational swing velocity when the bucket first reaches the pile.

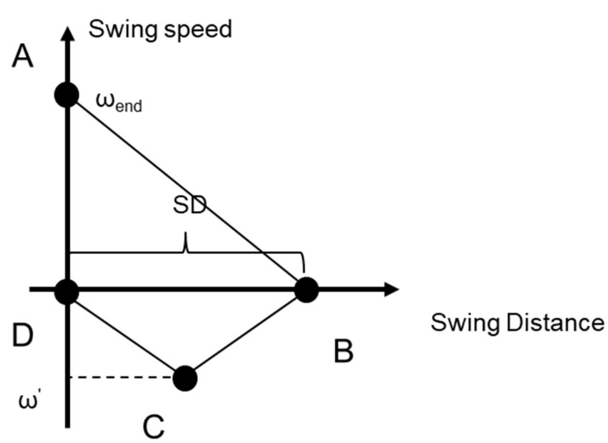


Figure 38. Physical relationship between swing speed and distance during pile overshoot used to estimate the penalty time acceleration effects factor.

The acceleration effects factor was calculated based on the swing overshoot physics (Figure 38). Suppose the bucket reached the pile at A with swing speed ω_{end}, and then started to decelerate at α. The swing continued beyond the pile to a swing distance of SD where the speed decreased to 0 at point B. The bucket then swung back to the pile and finally stopped at the pile at point D. The swing distances of AB and BD are the same and the deceleration and acceleration were assumed to be the same magnitude. Based on these assumptions, the following equations were written based on the swing physics:

$$\mathbf{SD} = \boldsymbol{\omega_{end}} \mathbf{t_1} - \frac{1}{2} \boldsymbol{\alpha t_1^2} \tag{7}$$

$$t_1 = \frac{\omega_{end}}{\alpha} \tag{8}$$

where t_1 is the time required to swing from A to B. Then the time required to swing back to the pile was estimated using similar mathematical relationships. First, since the bucket needs to swing the same distance to return back to the pile, that distance must be covered by first accelerating from point B to C. Then at C, deceleration takes place to so that when the bucket returns back to pile, it will stop. Based on this return trajectory with constant acceleration α, the distance will be covered is:

$$\mathrm{SD} = \frac{1}{2}\alpha t^2 + \omega' t' - \frac{1}{2}\alpha t'^2 \tag{9}$$

where t is the time for swing from B to C, and

t' is the time for swing from C to D.

Now if the deceleration and acceleration times in the return trip are set equal, then the velocity at C, ω', will be:

$$\omega' = \alpha t = \alpha t' \tag{10}$$

Then defining the penalty time as time to make the round, it is expressed mathematically as:

$$T_{Penalty} = t_1 + t + t' \tag{11}$$

By solving the equations (3 to 7), the penalty time is calculated as:

$$T_{Penalty} = \frac{(1+\sqrt{2})\omega_{end}}{\alpha} \tag{12}$$

Then the acceleration effect factor is

$$\mathrm{P} = \frac{(1+\sqrt{2})}{\alpha} \tag{13}.$$

Vehicle Model Design Iteration 1

Iteration 1 of the vehicle model design (VM1) had a larger displacement (0.1 m^3/rev) swing motor which required a large flow of hydraulic fluid, and thus responded to reference commands more slowly. Based on the ideal relationship, the load torque, T_{load}, produced by the motor is the product of the differential pressure across the motor ports, P_{Motor}, and the swing motor displacement, D_{Motor}. A swing motor with a larger displacement can provide a larger load torque, when the maximum pressure is held by a pressure relief valve during coast. On the mechanical side, the load torque T_{load} is dominated by inertial effects (vehicle swing inertia $I_{Vehicle}$, and the shaft angular acceleration α_{Shaft}, so with a larger motor, swing motion can be accelerated more quickly. The ideal flow rate $F_{Flowrate}$ consumed by the motor is the product of the shaft speed ω_{Shaft} of the swing motor and the motor displacement D_{Motor}, which reveals that a larger flow rate is required to reach a certain shaft speed with larger motor displacement.

Vehicle Model Design Iteration 2

For the second vehicle model design (VM2), the swing motor's displacement was reduced, resulting in a faster response. Compared to VM1, the VM2 had a smaller swing motor, therefore the swing motion experienced higher acceleration and deceleration with VM2. Nevertheless, VM2 resulted in a higher speed than VM1.

Results

Vehicle Model Design Iteration 1

Four combinations of $\theta_{Position}$ and θ_{Coast} were compared to demonstrate the dynamics of different combinations of control methods for vehicle 1. The swing speed as a function of time is illustrated in Figure 39. The results using four different combinations of $\theta_{Position}$ and θ_{Coast} are displayed. The labels denote potential shifts in control method: A indicates the start of coast, B indicates the start of position control, C indicates the start of overlap state with dumping task, and D indicates the target swing angle. The Swing speed as afunction of swing angle is illustrated in Figure **40**.

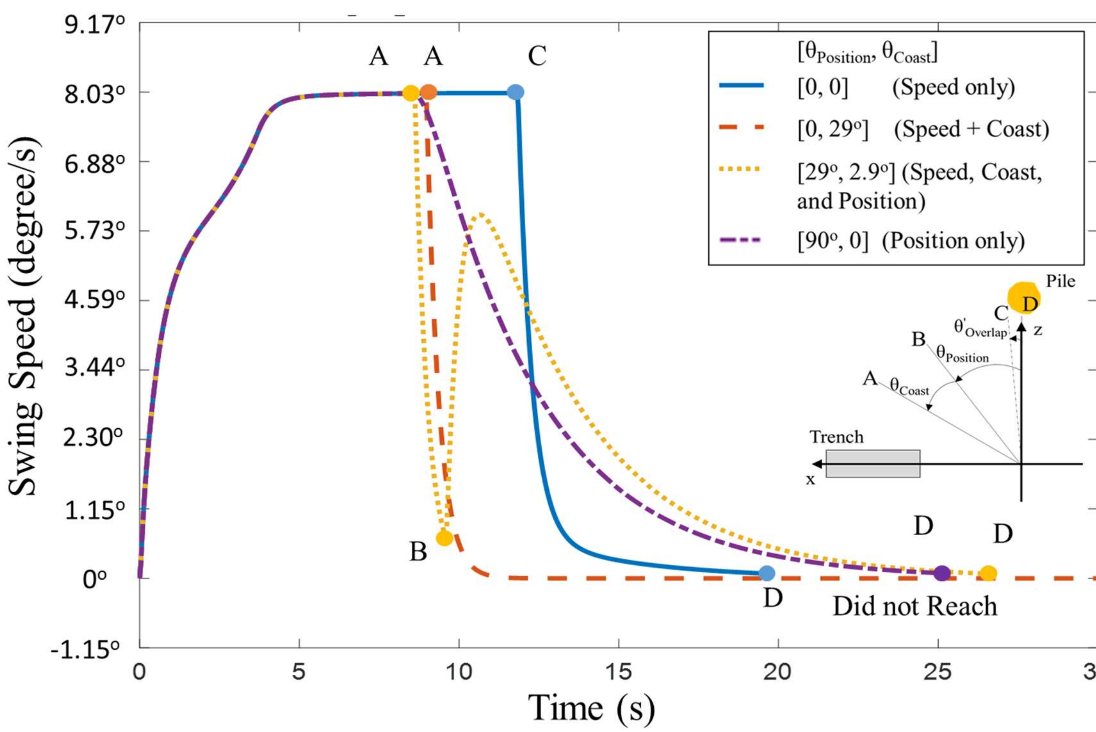


Figure 39**. Swing Speed vs. Swing Time for VM1**

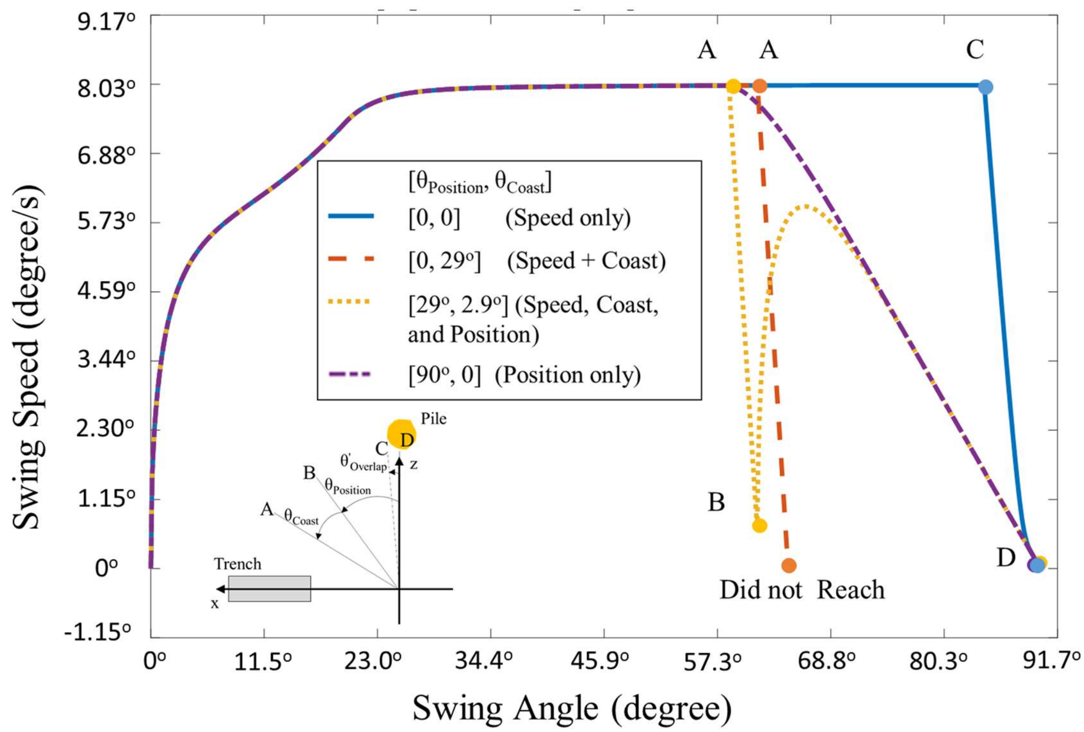


Figure 40**. Swing Speed VS. Swing Angle for VM1**

The bucket reached the target swing angle (D) under the three different combinations shown in

Figure **40**, but with different total swing time and with different swing speeds when the bucket reached the pile at the target angle shown in Figure **39**. The first combination ($\theta_{Position} = 0$, $\theta_{Coast} = 0$) used only speed control, and reached the target swing angle with the shortest time and the small swing speed. The second combination ($\theta_{Position} = 0^o$, $\theta_{Coast} = 0.28.66^o$) used speed control and coast, which never reached the target swing angle. The third combination ($\theta_{Position} = 28.66^o$, $\theta_{Coast} = 2.87^o$) used all three control methods (speed control, coast and position control), reached the pile with the longest time and small swing speed. The fourth combination ($\theta_{Position} = 90^o$, $\theta_{Coast} = 0^o$), which used only the position control, reached the target swing angle with the second shortest time and small swing speed. Three cases reached the target swing angle with small swing speed. For the combination ($\theta_{Position} = 0$, $\theta_{Coast} = 0$), the swing directional control valve closed when the swing reached pile and dumping task started, which resulted in the quick deceleration in swing-to-pile task shown in Figure 39 and indicated with C.

Swing Time for all combinations of control methods

Table 12 records the time to reach the pile at the specified target swing angle for a combination of ($\theta_{Position}$, θ_{Coast}). The cells with "∞" indicate that the combination of parameters resulted in a swing that never reached the pile. This was because the swing stopped (speed=0) before reaching the target swing angle. The greyed cells indicate the combinations that not feasible, where $\theta_{Coast} + \theta_{Position} > \theta_{Target}$.

Table 12. Swing Time to initially reach the Pile for VM1

		θ_{Coast} (degree)																
		0.00	**2.87**	**5.73**	**8.60**	**11.46**	**14.33**	**17.20**	**20.06**	**22.93**	**25.80**	**28.66**	**31.53**	**34.39**	**37.26**	**40.13**	**42.99**	**45.86**
$\theta_{Position}$ (degree)	**0.00**	19.54	19.54	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	5.73	19.54	23.36	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	
	11.46	23.33	25.17	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞		
	17.20	24.54	25.91	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞			
	22.93	25.06	26.21	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞				
	28.66	25.26	26.34	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞					
	34.39	25.28	26.28	∞	∞	∞	∞	∞	∞	∞	∞	∞						
	40.13	25.28	26.25	∞	∞	∞	∞	∞	∞	∞	∞							
	45.86	25.28	26.28	∞	∞	∞	∞	∞	∞	∞								
	51.59	25.28	26.24	∞	∞	∞	∞	∞	∞									
	57.32	25.28	26.26	∞	∞	∞	∞	∞										
	63.06	25.28	26.39	∞	∞	∞	∞											
	68.79	25.28	∞	∞	∞	∞												
	74.52	25.28	∞	∞	∞													
	80.25	25.28	∞	∞														
	85.99	25.28	∞															
	91.72	∞																

A penalty was imposed for combinations of $\theta_{Position}$ and θ_{Coast} resulting in a no-zero speed when the bucket reaches the pile. The penalty represents the time it would take the operator to overshoot the pile, reverse the direction of the bucket, and bring the bucket to a stop over the pile (Table 13).

Table 13. Time Penalty for VM1

$\theta_{Position}$ (degree) \ θ_{Coast} (degree)	0.00	2.87	5.73	8.60	11.46	14.33	17.20	20.06	22.93	25.80	28.66	31.53	34.39	37.26	40.13	42.99	45.86
0	0.03	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.73	0.03	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11.46	0.03	0.03	0	0	0	0	0	0	0	0	0	0	0	0	0		
17.20	0.03	0.03	0	0	0	0	0	0	0	0	0	0	0	0			
22.93	0.03	0.03	0	0	0	0	0	0	0	0	0	0	0				
28.66	0.03	0.03	0	0	0	0	0	0	0	0	0	0					
34.39	0.03	0.03	0	0	0	0	0	0	0	0	0						
40.13	0.03	0.03	0	0	0	0	0	0	0	0							
45.86	0.03	0.03	0	0	0	0	0	0	0								
51.59	0.03	0.03	0	0	0	0	0	0									
57.32	0.03	0.03	0	0	0	0	0										
63.06	0.03	0.03	0	0	0	0											
68.79	0.03	0	0	0	0												
74.52	0.03	0	0	0													
80.25	0.03	0	0														
85.99	0.03	0															
91.72	0																

Table 14 represents the total time of the swing–to-pile task, representing the time to reach pile (Table 12) plus the penalty (Table 13).

Table 14. Total Swing Time (Swing time plus Penalty) for VM1

		θ_{Coast} (degree)																
		0	**2.87**	**5.73**	**8.60**	**11.46**	**14.33**	**17.20**	**20.06**	**22.93**	**25.80**	**28.66**	**31.53**	**34.39**	**37.26**	**40.13**	**42.99**	**45.86**
$\theta_{Position}$ (degree)	**0**	19.57	19.57	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	5.73	19.57	23.38	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	
	11.46	23.35	25.20	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞		
	17.20	24.56	25.93	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞			
	22.93	25.08	26.24	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞				
	28.66	25.28	26.36	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞					
	34.39	25.31	26.30	∞	∞	∞	∞	∞	∞	∞	∞	∞						
	40.13	25.31	26.28	∞	∞	∞	∞	∞	∞	∞	∞							
	45.86	25.31	26.30	∞	∞	∞	∞	∞	∞	∞								
	51.59	25.31	26.27	∞	∞	∞	∞	∞	∞									
	57.32	25.31	26.28	∞	∞	∞	∞	∞										
	63.06	25.31	26.41	∞	∞	∞	∞											
	68.79	25.31	∞	∞	∞	∞												
	74.52	25.31	∞	∞	∞													
	80.25	25.31	∞	∞														
	85.99	25.31	∞															
	91.72	∞																

The total swing time is illustrated in Figure 42 as both a 3D surface plot (left) and a contour plot (right). The shortest time for swing-to-pile task was with the combination of $\theta_{Position} = 0^o$ and $\theta_{Coast} = 0^o$. A strategy of only speed control resulted in the most efficient swing-to-pile.

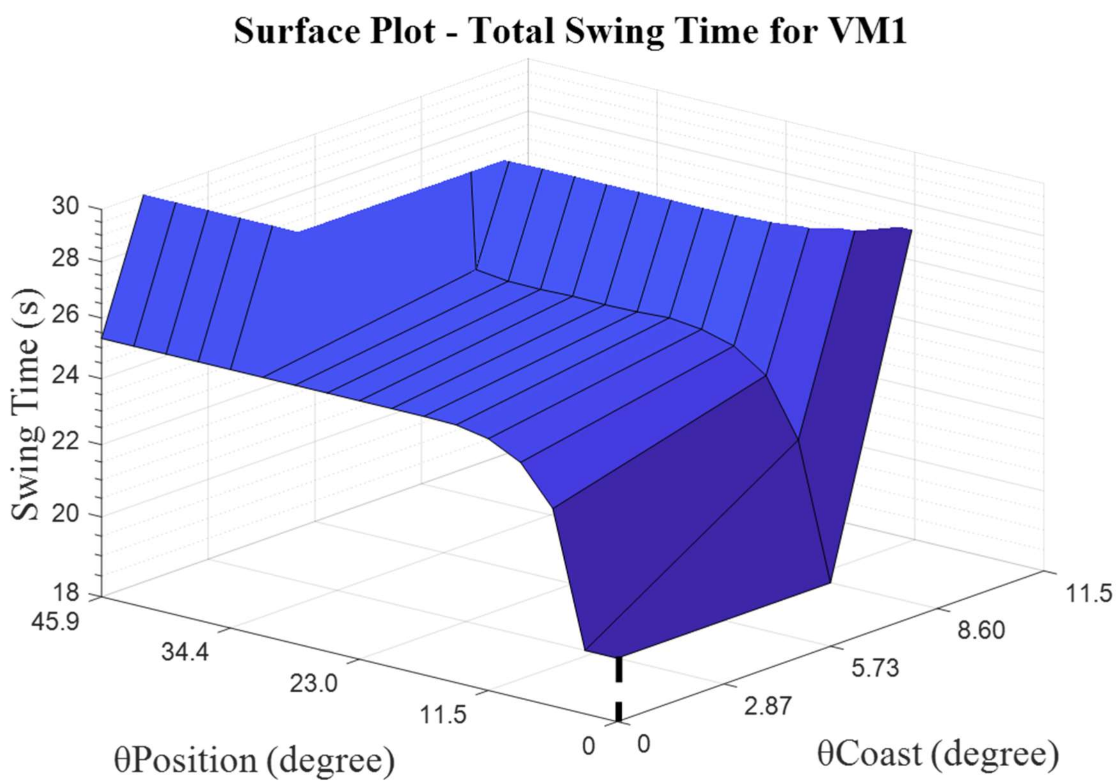


Figure 41. Surface Plot of Total Swing Time (Swing Time + Penalty) for VM1

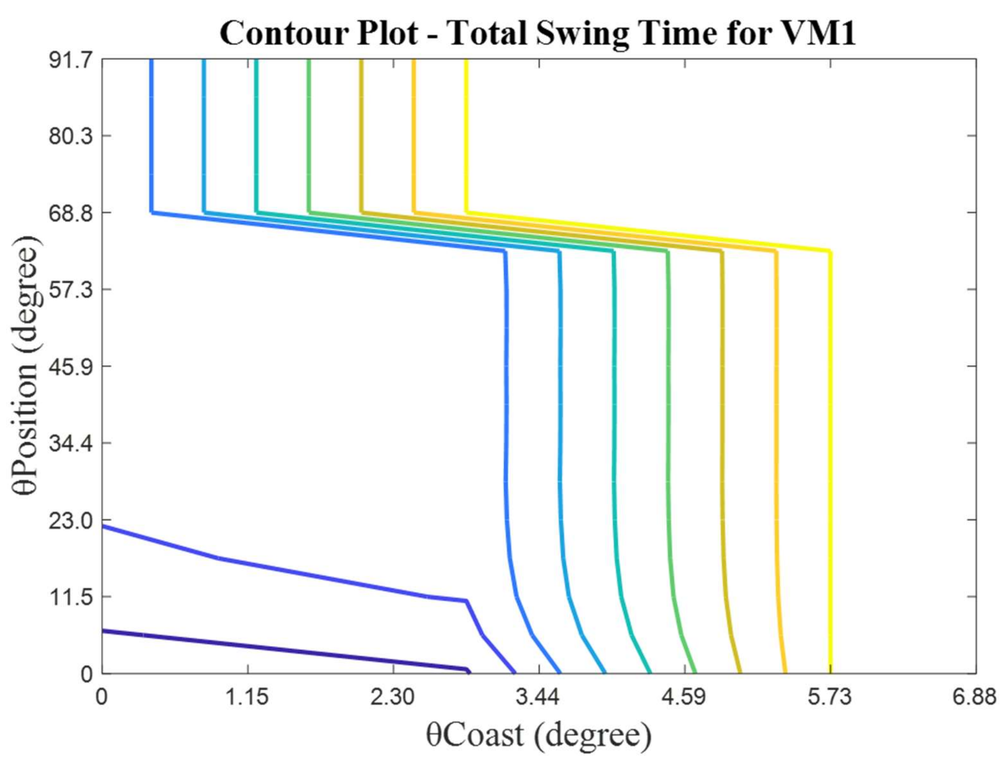


Figure 42. Contour plot of Total Swing Time (Swing Time + Penalty) for VM1.

. The GA progressed for 16 generations until the stopping condition was satisfied, resulting in a fitness value of 19.57 seconds. The GA found the best combination of ($\theta_{Position}$ = 0°, θ_{Coast} = 0.99°) illustrated in Figure 43, which matched with the swing time distribution illustrated both in the table (Table 14) and surface plot (Figure 42).

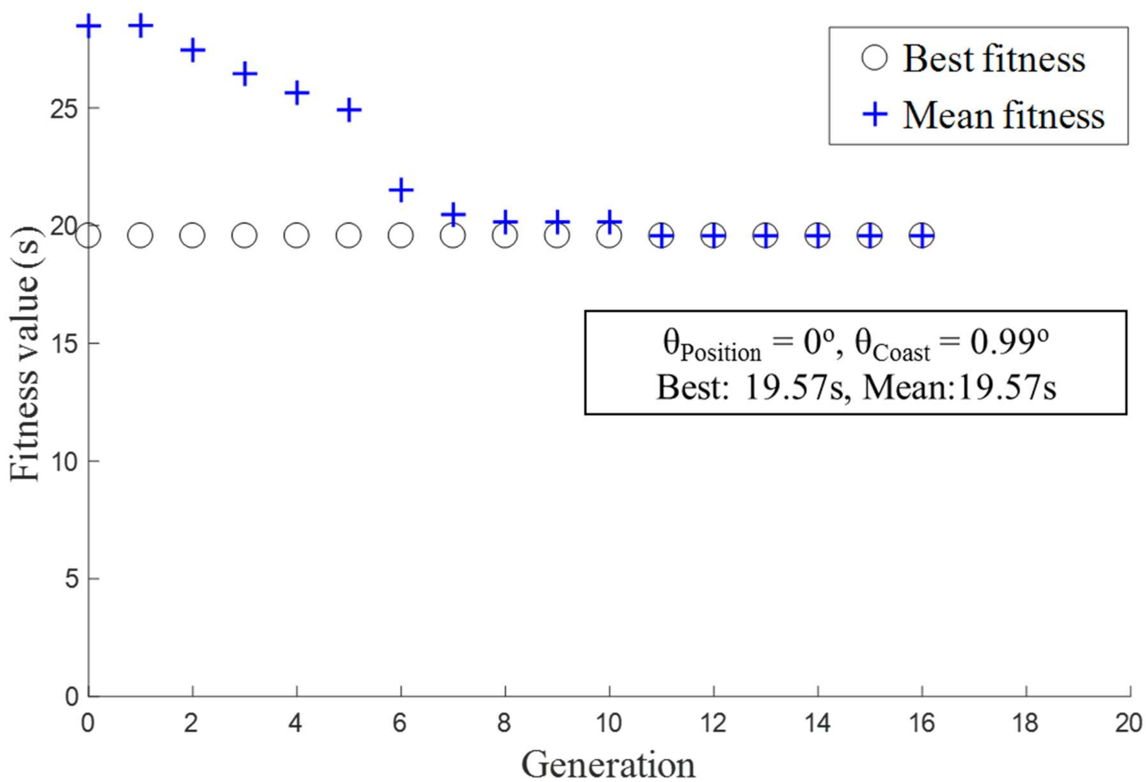


Figure 43. GA Optimization Result for VM1.

Vehicle Model Design Iteration 2

Swing speed vs Swing Angle for a Few Illustrative Combinations

Four combinations of $\theta_{Position}$ and θ_{Coast} were compared to understand the dynamics of different combinations of control methods for VM2. The Swing speed as a function of time is illustrated in Figure 44. The results using four different combinations of $\theta_{Position}$ and θ_{Coast} are displayed. The labels denote potential shifts in control method: A indicates the start of coast, B indicates the start of position control, C indicates the start of overlap state with dumping task, and D indicates the target swing angle. The Swing speed as a function of swing angle is illustrated in Figure 45.

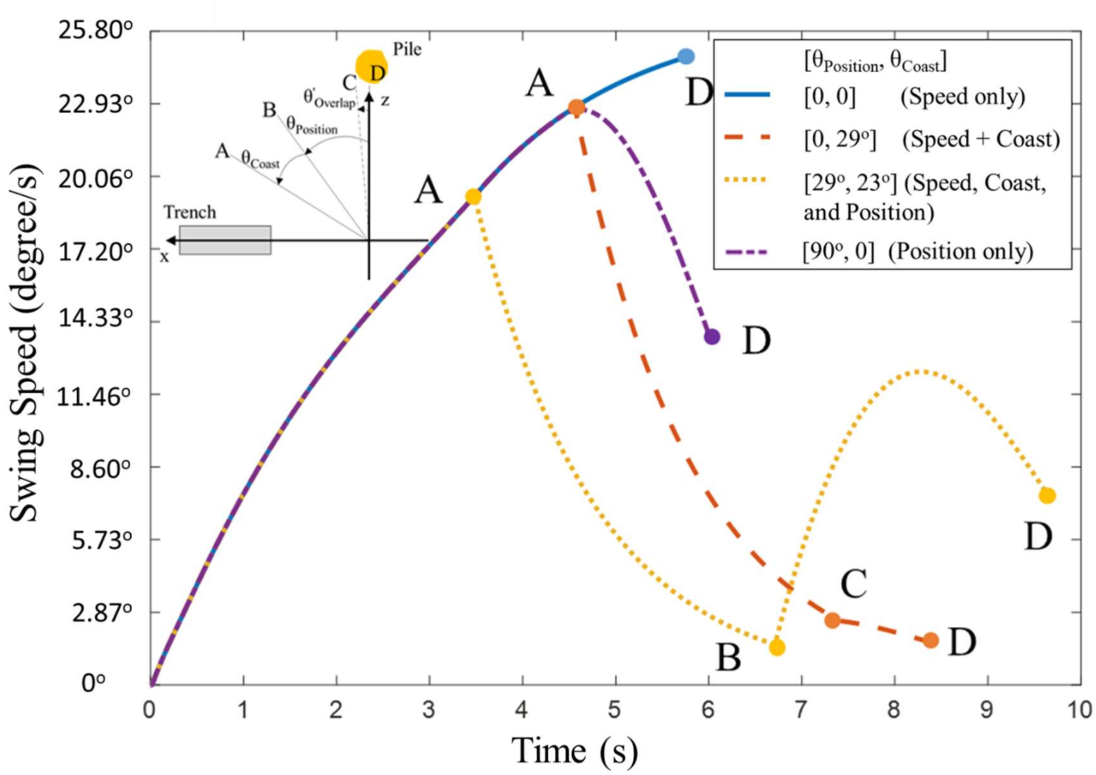


Figure 44. Swing Speed VS. Swing Time for VM2

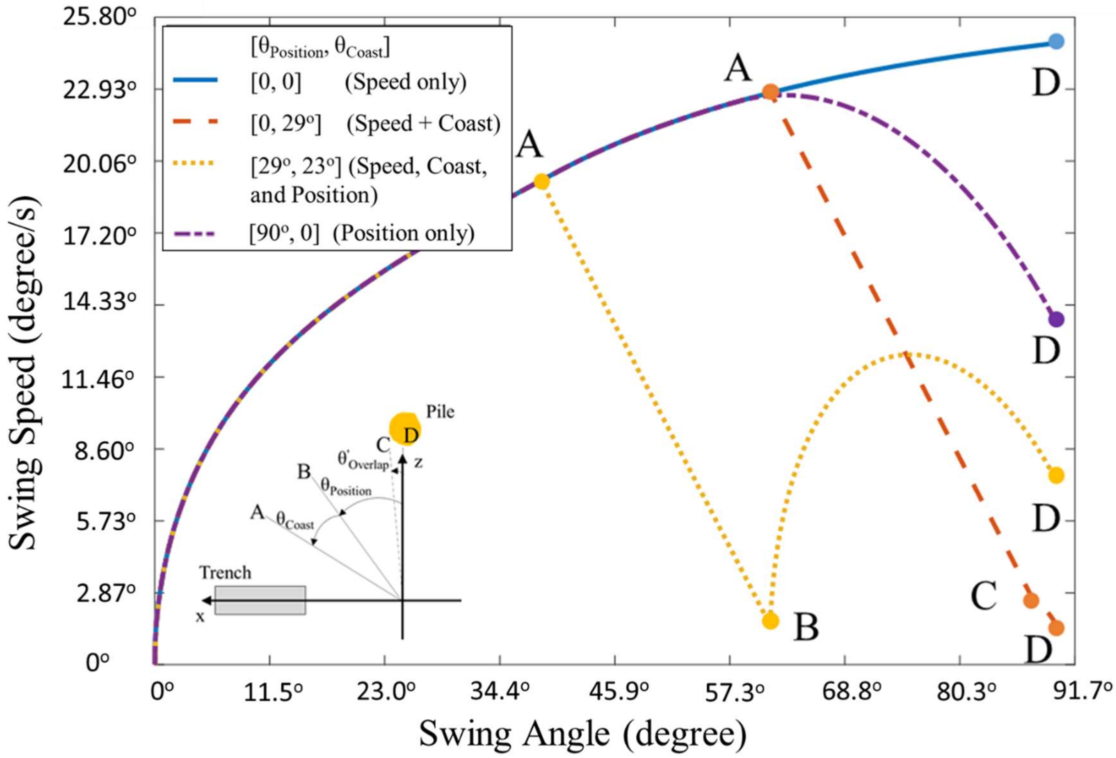


Figure 45. Swing Speed VS. Swing Angle for VM2

The bucket reached the target swing angle (D) under the four different combinations shown in Figure 44 and Figure 45, but with different total swing time and with different swing speeds when the bucket reached the pile at the same target angle. The first combination ($\theta_{Position} = 0^o$, $\theta_{Coast} = 0^o$) used only speed control, and reached the target swing angle with the shortest time but the largest swing speed. The second combination ($\theta_{Position} = 0^o$, $\theta_{Coast} = 28.66^o$) used speed control and coast, which reached the target swing angle with the second longest time and smallest swing speed. The third combination ($\theta_{Position} = 28.66^o$, $\theta_{Coast} = 22.93^o$) used all three control methods (speed control, coast and position control), reached the pile with the longest time and second smallest swing speed. The fourth combination ($\theta_{Position} = 90^o$, $\theta_{Coast} = 0^o$), which used only the position control, reached the target swing angle with the second shortest time but the second the largest swing speed.

Swing Time for all combinations of control methods

Table 15 records the swing time to reach the pile at the specified target swing angle 90°. The cells with infinity (∞) indicate that the combination of parameters resulted in a swing that never reached the pile. This was because the swing stopped (speed=0) before reaching the target swing angle. The greyed cells indicate the combinations where $\theta_{Coast} + \theta_{Position} > \theta_{Target}$ were infeasible

Table 15. Swing time to initially reach the pile for VM2.

	θ_{Coast} (degree)																
$\theta_{Position}$ (degree)	**0**	**5.73**	**11.46**	**17.20**	**22.93**	**28.66**	**34.39**	**40.13**	**45.86**	**51.59**	**57.32**	**63.06**	**68.79**	**74.52**	**80.25**	**85.99**	**91.72**
0	5.75	5.77	5.87	6.08	6.59	8.37	14.73	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
5.73	5.77	5.86	6.06	6.52	7.74	14.73	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	
11.46	5.82	5.99	6.34	7.06	8.53	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞		
17.20	5.90	6.15	6.61	7.41	8.89	∞	∞	∞	∞	∞	∞	∞	∞	∞			
22.93	5.97	6.27	6.78	7.59	9.13	∞	∞	∞	∞	∞	∞	∞	∞				
28.66	6.01	6.34	6.86	7.65	9.65	∞	∞	∞	∞	∞	∞	∞					
34.39	6.01	6.36	6.87	7.73	11.69	∞	∞	∞	∞	∞	∞						
40.13	6.01	6.37	6.91	7.90	∞	∞	∞	∞	∞	∞							
45.86	6.01	6.39	6.97	8.17	∞	∞	∞	∞	∞								
51.59	6.01	6.40	7.03	8.70	∞	∞	∞	∞									
57.32	6.01	6.40	7.12	11.91	∞	∞	∞										
63.06	6.01	6.40	7.31	∞	∞	∞											
68.79	6.01	6.42	7.88	∞	∞												
74.52	6.01	6.48	∞	∞													
80.25	6.01	6.77	∞														
85.99	6.01	∞															
91.72	∞																

Table 16 calculates the penalty imposed for combination θ_{Coast} and $\theta_{Position}$ where the speed when the bucket reaches the pile is greater than zero. The penalty represents the time it would take the operator to overshoot the pile, reverse the direction of the bucket, and bring the bucket to a stop over the pile.

Table 16. Penalty time for VM2.

		θ_{Coast} (degree)																
		0	**5.73**	**11.46**	**17.20**	**22.93**	**28.66**	**34.39**	**40.13**	**45.86**	**51.59**	**57.32**	**63.06**	**68.79**	**74.52**	**80.25**	**85.99**	**91.72**
$\theta_{Position}$ (degree)	**0**	8.64	7.11	5.41	3.78	2.06	0.59	0.58	0	0	0	0	0	0	0	0	0	0
	5.73	7.29	5.64	4.06	2.48	1.11	0.58	0	0	0	0	0	0	0	0	0	0	
	11.46	6.18	4.68	3.25	2.02	1.26	0	0	0	0	0	0	0	0	0	0		
	17.20	5.43	4.11	2.95	2.11	1.66	0	0	0	0	0	0	0	0	0			
	22.93	4.99	3.88	2.98	2.39	2.12	0	0	0	0	0	0	0	0				
	28.66	4.81	3.88	3.19	2.77	2.60	0	0	0	0	0	0	0					
	34.39	4.79	4.02	3.48	3.17	3.07	0	0	0	0	0	0						
	40.13	4.79	4.16	3.74	3.53	0	0	0	0	0	0							
	45.86	4.79	4.28	3.97	3.83	0	0	0	0	0								
	51.59	4.79	4.39	4.18	4.09	0	0	0	0									
	57.32	4.79	4.52	4.35	4.30	0	0	0										
	63.06	4.79	4.62	4.53	0	0	0											
	68.79	4.79	4.72	4.65	0	0												
	74.52	4.79	4.80	0	0													
	80.25	4.79	4.89	0														
	85.99	4.79	0															
	91.72	0																

Table 17 represents the total time of the Swing-to-pile task, representing the time to reach pile (Table 15) plus the penalty (Table 16).

Table 17. Total Swing time (Swing time plus penalty) for VM2

		θ_{Coast} (degree)																
		0	**5.73**	**11.46**	**17.20**	**22.93**	**28.66**	**34.39**	**40.13**	**45.86**	**51.59**	**57.32**	**63.06**	**68.79**	**74.52**	**80.25**	**85.99**	**91.72**
$\theta_{Position}$ (degree)	**0**	14.39	12.88	11.28	9.86	8.65	8.96	15.31	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	5.73	13.06	11.50	10.12	9.00	8.85	15.31	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	
	11.46	12.00	10.67	9.59	9.08	9.79	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞		
	17.20	11.33	10.25	9.56	9.51	10.55	∞	∞	∞	∞	∞	∞	∞	∞	∞			
	22.93	10.96	10.15	9.76	9.98	11.25	∞	∞	∞	∞	∞	∞	∞	∞				
	28.66	10.81	10.22	10.04	10.42	12.25	∞	∞	∞	∞	∞	∞	∞					
	34.39	10.80	10.38	10.34	10.90	14.75	∞	∞	∞	∞	∞	∞						
	40.13	10.80	10.53	10.65	11.42	∞	∞	∞	∞	∞	∞							
	45.86	10.80	10.66	10.94	12.00	∞	∞	∞	∞	∞								
	51.59	10.80	10.79	11.21	12.79	∞	∞	∞	∞									
	57.32	10.80	10.92	11.47	16.21	∞	∞	∞										
	63.06	10.80	11.01	11.84	∞	∞	∞											
	68.79	10.80	11.13	12.53	∞	∞												
	74.52	10.80	11.28	∞	∞													
	80.25	10.80	11.66	∞														
	85.99	10.80	∞															
	91.72	∞																

By considering the penalty, the total swing time with penalty was calculated, and is demonstrated in Figure 47 illustrated the total Swing-to-pile time and is the sum of the Swing Time and penalty for VM2. The shortest time for swing-to-pile task was with the combination of $\theta_{Position}$ = 0° and θ_{Coast} = 22.93°. The combination of speed control and coast resulted in the most efficient swing-to-pile.

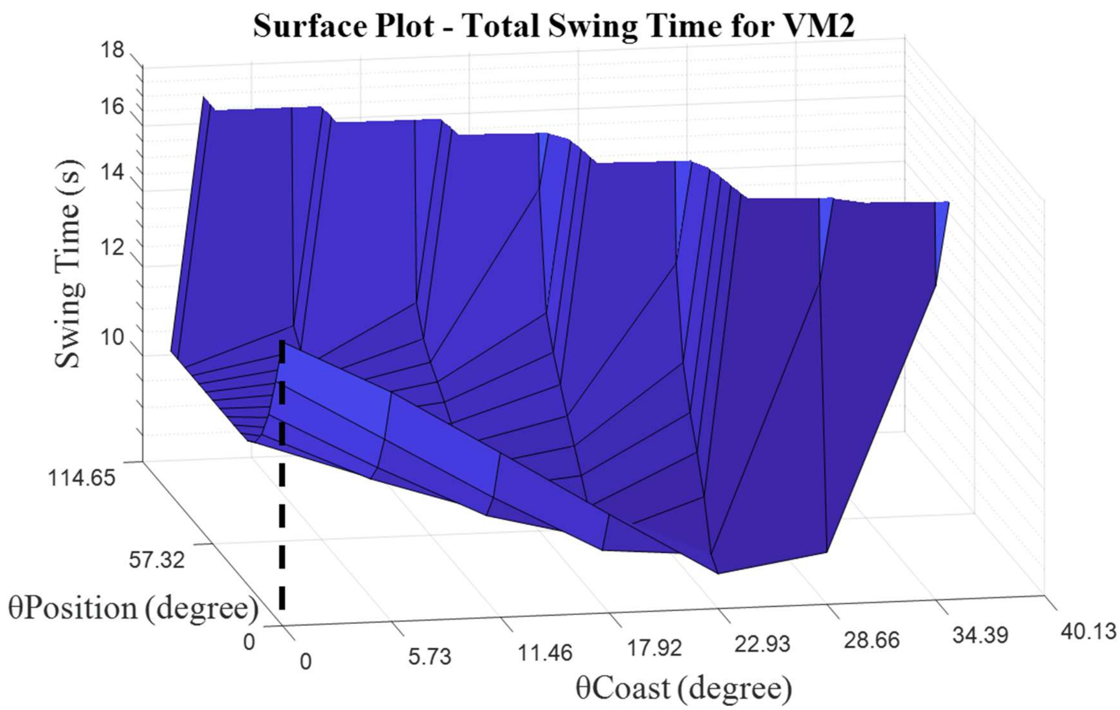


Figure 46. Surface Plot of Total Swing Time (Swing Time + Penalty) for VM2

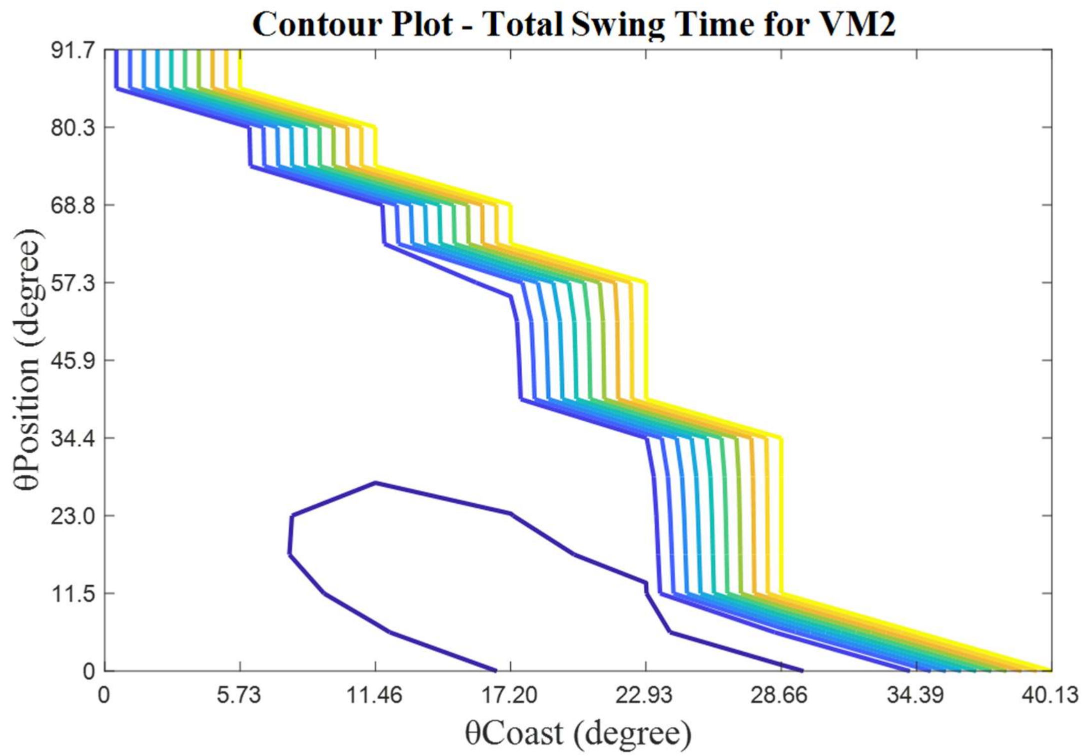


Figure 47. Contour Plot of Total Swing Time (Swing Time + Penalty) for VM2

Optimization

The GA population size of 10 used for the optimization. Twenty-five generations were conducted to achieve the best fitness value of 8.36. The optimization using genetic algorithm found the optimal combination (θ1 = 0.68º, θ2 = 25.2º), which was demonstrated in Figure 48. The combination of the parameters matched with the surface plot and contour plot (Figure 47).

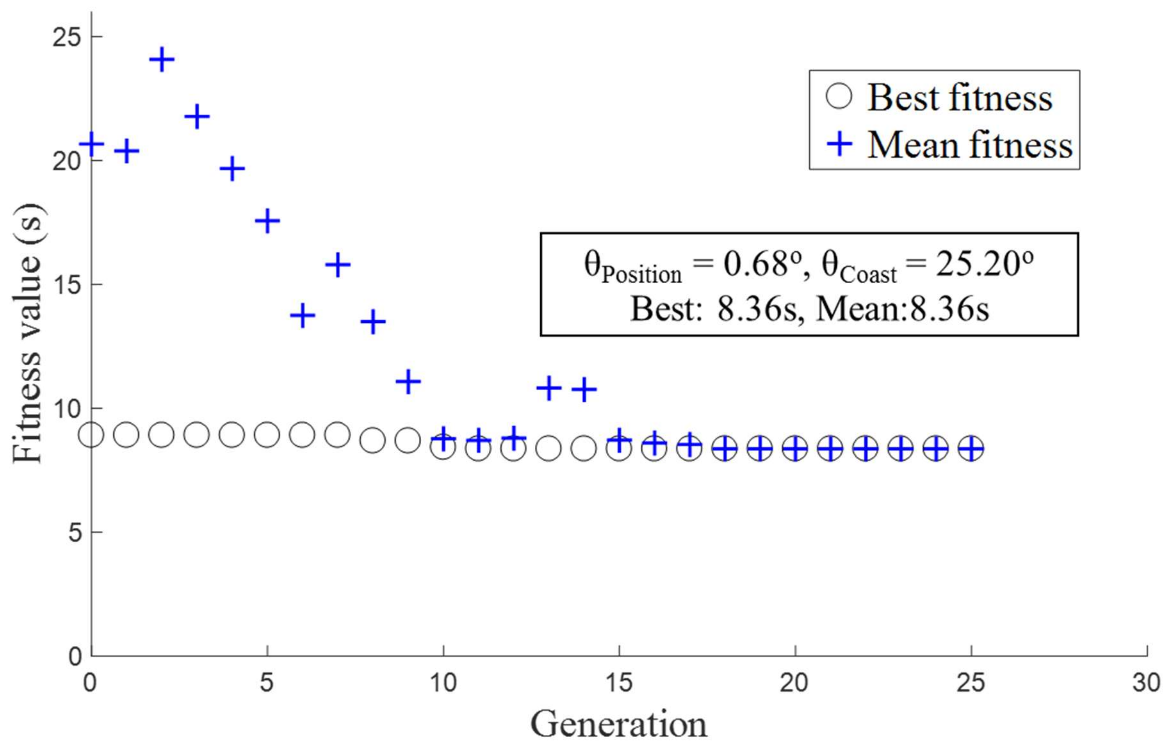


Figure 48. GA Optimization Result for VM2.

Discussion/Conclusion

Learning results of different iterations comparison

In real operations, human operator accelerates the machine to a certain speed, and when a particular swing angle is achieved, the human operator stops providing control inputs. Then the vehicle is swung by inertia. In the best case, the excavator stops swinging or swings with a very low speed when the bucket reaches the pile, and overshoot of the pile can be reduced.

The two learning cases resulted in different combinations of control methods to complete the swing-to-pile task with least time Learning case 1 identified the best combination of the control methods used only the speed control. The vehicle model used in learning case 1 required large flow rate, and the vehicle had a maximum 2.87° rad coast ability. The simulation results revealed that the vehicle model quickly reached the flow rate limit, which limited the swing speed. VM1 took a longer time to reach the pile. The deceleration of the VM1 was very fast due to large load torque, which essentially prevented coasting. Based on the observations above, the VM1 did not provide capabilities for the operator to apply different control strategies, and it quickly decelerated when coast initiated. To swing with maximum swing speed was the most efficient way to complete the swing-to-pile task.

By reviewing the swing time with penalty results in Table 14, the shortest time is 74.1% of the longest time, which means that choosing the best transitions between does make a difference, and can decrease the swing time substantially. This phenomenon also illustrates the importance of utilizing a learning capability in a VOM to derive the best expert strategy to be used in model-based design.

The second iteration of the vehicle model (VM2) used the swing motor with a smaller displacement. The results for learning case 2 identified the best combination of the control methods to be the combination of speed control and coast. Based on the simulation results, the swing motor with a smaller displacement reached a higher swing speed, therefore VM2 took a shorter time to reach the pile. VM2 decelerated slower due to smaller load torque, which resulted in 34.39° of coast. In Figure 44 and Figure 45, the simulation results represent the different speed at the pile for different combinations of control methods. The larger speed resulted in longer time penalty, during which the vehicle would overshoot, stop and swing back to pile. Based on the results, different combinations of control methods contributed differently to the swing-to-pile task, which allow the operator to apply different control methods to complete the task. Compared to VM1, triggering of the overlap task swing and dump does not have much impact on the task swing-to-pile since it does not require much fluid flow for the swing motor, which means the tasks can be commanded at the same time without slowing down the swing-to-pile and dump.

By comparing the shortest time to the longest time in Table 17, the shortest time is 53.36% of the longest time. It implies the more significant impact of applying the best strategy in simulation.

The learning capability can optimize the product design process to modify design under correct guidance between different design iterations. It is important to derive the best strategy via a combination of control methods to control the vehicle during a task. If a VOM used the same strategy no matter what the vehicle model characteristics, it cannot be guaranteed that it is using the best strategy. For instance, the best strategy for VM1 was solely speed control. If that strategy had been applied to VM2 (a design iteration aimed at improving the performance), there would be a 26.5% efficiency increase from iteration VM1. But by learning the best strategy to operate VM2, the result was a 55.8% efficiency increase between design iterations (VM1 to VM2). The difference is a factor of two. Different vehicle models require different strategies to reach the most efficient operation performance. Human operators learn the strategies to operate different machines efficiently with practice over a period of time. The learning capability of the VOM replicates this learning process. If the same strategy for all tests of vehicle model iterations. The VOM may not be operating the machine at the most efficient manner to “push” the machine to it limits. Human operators do this all the time, tailoring their strategies to get the most out of the machine. The learning capability of the VOM allows it to do the same.

The learning cases demonstrated the model-based design process (Figure 34) and how important the learning capability can impact on this process. Learning case 1 revealed a suboptimal design of the vehicle. Learning case 2 learned the best strategy using the modified vehicle model VM2 and showed that the VM2 could be used in a way that was more efficient that VM1. The different strategies can be identified for the different vehicle models. Much like a human, over repeated use of a machine, will eventually learn the best way to control it, the VOM learning module calculates the best combination of control strategies and parameters. Human operators learn in the process of adapting to different machines. Since there was nearly no coast capability for the VM1, and the effect of the speed drop during the overlap state, the learned strategy was speed control for swing-to-pile task. The operator utilized the coast capability of the VM2, which the VOM learned the operation strategy was the combination of speed control and coast. This work focused on developing the learning scenario to model the learning process of real operators based on the operator interview. The learning scenario demonstrated the meaningful learning outcomes to represent how real operators develop the strategy and expertise for swing-to-pile task. The iterations also demonstrated the possible uses in the model-based design process.

Future work

For the complete trenching work cycle, it is necessary to investigate the learning methods for other tasks, such as bucket fill, bucket lift, swing to trench, and dump. The current VOM was only designed with three possible control strategies (speed control, coast, and position control). Additional control methods could be developed for specific learning scenario. For instance, developing learning methods for bucket fill, a soil model needs to be incorporated, which could describe the interactions between the bucket and soil. Using the feedback from the soil model, such as reaction force, and soil type, can be used to develop control methods for bucket fill by adapting to different soil types.

CHAPTER V: CONCLUSION

Summary

This work aimed to develop a VOM to provide control inputs to a vehicle mode in the same way an expert human operator would. A high fidelity VOM paired with high fidelity vehicle models would increase the utility of model-based design process, providing reliable simulation results for machine design assessment in the virtual environment. Both operator interviews/observations and experimental operation data were used to derive a task analysis of the excavator trenching operation and provided information about human operators' behavior. Based on the task analysis, the structure of the VOM was developed to mirror the human information processing model: perception, decision-making, and action execution.

Phase I of the work developed a VOM with repeating one work cycle with finite tasks. Phase II advanced the VOM with more realistic representations of dynamic work cycles adapting to environment changes, operator adaptation to different vehicle models, and modeling the human expert operator ability to overlap tasks.. Phase III implemented a VOM learning capability to learn the optimal parameters for a task, much in the way a human does over time with a new machine, so that the VOM can find optimal strategies with different combinations of control methods to operate the machine in the most efficient manner.

Virtual design of off-highway machines with operators in the loop has often been limited by the fidelity of the model of human operators. Compared to the current state of the art, our approach modeled the virtual operator in a way that is similar to the internal process as human operator. This structure provided an extensible foundation to continually improve the fidelity of the model by adding aspects of adaptability and learnability. The work of this project has provided key improvements to the fidelity and utility of VOMs. The VOM represented human expertise by modeling how experts improve the productivity of an operation by overlapping that tasks of the operation to improve efficiency. Traditional validation and assessment methods, by way of comparison, utilize physical machine prototypes, human operators, and real-world testing in a controlled environment (Filla, Ericsson, & Palmberg, 2005). Model-based design has been limited by the need to painstakingly "re-tune" trajectory-based VOMs each time the vehicle model design ins changed. Our VOM, on the other hand, is a generalizable model that relies on human-level perception of the machine operating characteristics, machine dimensions, and the environment. Thus, it can automatically adjust to new vehicle models without re-tuning, potentially greatly decreasing the effort needed to test new design iteratively. Finally, out work demonstrated an approach that represents the ability of a VOM to "learn" how to optimize the control parameters of a task.

Future Work

More research is needed on multiple aspects of the work described here. To address the expertise representation several aspects can be investigated. It is important to know how different skill levels impact the proportion of or decision to overlap tasks. A more nuanced understanding of how expertise is realized will enable VOMs to simulate different levels of skill.

Future work needs to be done to increase the fidelity of the environment representation, and how that interacts with operations. Adaptation to environment changes was limited to changing depths of trench and heights of pile during the operation. The conditions of the worksite can be considered for the future work, such as soil type, and obstacles. The adaptation to different machines focused purely on dimensions of the vehicle components. More vehicle differences can be adapted in the future work, such as the different power levels of different vehicles.

In the current work, we assumed that the virtual operator attended to all the cues provided by the human perception module. But in real operations, there is a level of uncertainty in the perception of all available information. For instance, as humans become fatigued, they start to miss information, or their attention becomes increasingly narrowly focused. Future work would model some level of information perception uncertainty, perhaps depended on a model of attention that could be included by operator fatigue, environmental noise, or distraction. Additional control methods could be developed for specific learning case. The results of the learning cases can be used as the optimal methods to conduct certain tasks, which should be applied in the VOM to simulate the whole operation. By completion these improvements, the VOM can be used to test machine model under certain conditions and provide more realistic results.

Contribution

The contributions of this work are focused on the following areas: 1) an VOM structure was developed based on human perception and decision-making system; 2) closed loop simulation was enabled by connecting VOM and Vehicle Model; 3) the method was developed to represent the expertise; 4) the methods were developed to enable the VOM to adapt automatically to changes in the environment, and to different machines; 5) a learning method was developed to enable the VOM to optimize the control parameters within a task.

The capabilities of the VOM developed in this work are essential to advance VOM model fidelity to the point where designers can rapidly test design iterations virtually. By enabling the VOM to represent expert behavior, the simulation can push the machine model to its limits. Currently test operators can push machines during productivity tests by exploiting all the capabilities of the machine. By more accurately representing human expertise in a VOM, design engineers can be more confident that model-based simulations more accurately reflect what human operators can achieve with the machine. Furthermore, by building a VOM that can adapt to changes in the environment, complete operations can be simulated, further enhancing the utility of model-based testing. Additionally, the ability of the VOM to adapt to different machines without time-consuming re-tuning is essential to enabling the rapid design iterations. The design engineers' time will be spent on iterating the machine design, rather than tuning the VOM to test a particular machine design. Finally, the learning capability can result in the meaningful learning outcomes to represent how real operators develop the strategy and expertise for the task, which can be used to determine strategies for the operations in simulation. This would avoid using the predefined strategies for simulations. This work is a step towards the vision of developing VOMS with a fidelity that matches the current fidelity of machine models.

REFERENCES

Ackerman, M. S., Pipek, V., & Wulf, V. (Eds.). (2003). Sharing expertise: Beyond knowledge management. MIT press.

Adams, J. A. (1987). Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. Psychological bulletin, 101(1), 41.

Akhavian, R., & Behzadan, A. H. (2015). Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. Advanced Engineering Informatics, 29(4), 867-877.

Alami, R., Chatila, R., Fleury, S., Ghallab, M., & Ingrand, F. (1998). An architecture for autonomy. *The International Journal of Robotics Research*, *17*(4), 315-337.

Aoyama, H., & Kimishima, Y. (2006). Development of system using mixed reality technology for evaluating designability and operability of product. Research in Interactive Design, 2.

Althoefer, K., Tan, C. P., Zweiri, Y. H., & Seneviratne, L. D. (2009). Hybrid soil parameter measurement and estimation scheme for excavation automation. IEEE Transactions on Instrumentation and Measurement, 58(10), 3633-3641.

Albus, J., Bostelman, R., Hong, T., Chang, T., Shackleford, W., & Shneier, M. (2007). Integrating learning into a hierarchical vehicle control system. *Integrated Computer-Aided Engineering*, *14*(2), 121-139.

Azadivar, F., & Tompkins, G. (1999). Simulation optimization with qualitative variables and structural model changes: A genetic algorithm approach. European Journal of Operational Research, 113(1), 169-182.

Becker, M. C., Salvatore, P., & Zirpoli, F. (2005). The impact of virtual simulation tools on problem-solving and new product development organization. Research Policy, 34(9), 1305-1321.

Baddeley, A. (1998). *Human Memory: Theory and Practice*. London: Taylor & Francis.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J. & Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

Bradley, D., & Seward, D., (1998). The Development, Control and Operation of an Autonomous Robotic Excavator. *Journal of Intelligent and Robotic Systems* 21: 73-97, 1998.

Byrne, M., & Kirlik, A. (2005). Using computational cognitive modeling to diagnose possible sources of aviation error. *International Journal of Aviation Psychology*, 15 (2): 135-55.

Carson, Y., & Maria, A. (1997, December). Simulation optimization: methods and applications. In *Proceedings of the 29th conference on Winter simulation* (pp. 118-126). IEEE Computer Society.

Cun, Y. L., Muller, U., Ben, J., Cosatto, E., & Flepp, B. (2006). Off-road obstacle avoidance through end-to-end learning. In Advances in neural information processing systems (pp. 739-746).

Davis, L. (1991). Handbook of Genetic Algorithms Van Nostrand Reinhold New York.

Dix, A., Finley, J., Abowd, G. & Beale, G. (2004). Human-computer interaction. England: Pearson Education Limited.

Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B., & Schreckenghost, D. (1999, August). Adjustable autonomy for human-centered autonomous systems. *In Working notes of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems* (pp. 16-35).

Du, Y., Dorneich, M.C., Steward, B.L., Anderson, E.R., Kane, L.F., & Gilmore, B. (2015). Virtual Operator Modeling Approach for Construction Machinery. *2015 Conference on Autonomous and Robotic Construction of Infrastructure*. Ames, IA, June 2-3.

Du, Y., Dorneich, M. C., & Steward, B. (2016). Virtual operator modeling method for excavator trenching. *Automation in construction*, *70*, 14-25.

Du, Y., Dorneich, M. C., & Steward, B. (2018). Modeling expertise and adaptability in virtual operator models. *Automation in Construction*, 90, 223-234.

Du, Y., Dorneich, M. C., & Steward, B. (2019accpted) Development of A Learning Capability In Virtual Operator Models. *International Journal of Commercial Vehicles*.

Elezaby, A. A. (2011). Virtual Autonomous Operator Model for Construction Equipment Applications. Dissertation, University of Illinois – Chicago, Chicago, Illinois.

Enes, A. R. (2010). Shared Control of Hydraulic Manipulators to Decrease Cycle Time. *Ph.D. thesis*. Georgia Tech.

Ericsson, K., & Simon, H. (1993). Protocol Analysis: Verbal Reports as Data (2nd Ed.). Boston: MIT Press. ISBN 0-262-05029-3

Filla. R. (2005). Operator and Machine Models for Dynamic Simulation of Construction Machinery. Thesis No. 1189. Linkoping University, Linkoping, Sweden.

Filla, R., Ericsson, A., & Palmberg, J.-O. (2005). Dynamic Simulation of Construction Machinery: Towards an Operator Model. *International Fluid Power Exhibition 2005 Technical Conference*, Las Vegas, (NV), USA, pp. 429-438.

Filla, R., & Palmberg, J. O. (2003). Using dynamic simulation in the development of construction machinery. arXiv preprint cs/0305036.

Fountas, S., Blackmore, B. S., Vougioukas, S., Tang, L., Sorensen, C. G., & Jorgensen, R. (2007). Decomposition of agricultural tasks into robotic behaviors.

Goodrich, M. A., Olsen, D. R., Crandall, J. W., & Palmer, T. J. (2001, August). Experiments in adjustable autonomy. *In Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents* (pp. 1624-1629).

Gross, D. C. (1999, March). Report from the fidelity implementation study group. In *Fall Simulation Interoperability Workshop Papers*.

Guo, P., Wang, X., & Han, Y. (2010, October). The enhanced genetic algorithms for the optimization design. In Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on (Vol. 7, pp. 2990-2994). IEEE.

Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Scoffier, M., Kavukcuoglu, K., Muller, U.,& LeCun, Y. (2009). Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, *26*(2), 120-144.

Hamner, B., Singh, S., & Scherer, S. (2006). Learning obstacle avoidance parameters from operator behavior. *Journal of Field Robotics*, *23*(11-12), 1037-1058.

Han, S., Steward, B. L., and L. Tang. (2015). Intelligent Agricultural Machinery and Field Robots. In Precision Agriculture Technology for Crop Farming, ed. Zhang, Q. CRC Press.

Holtzblatt, K. (2003). Contextual Design. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications* (Jacko, J.A., Ed.). CRC press. Pp. 983-1002.

Hughes, K., & Jiang, X. (2010). Using discrete event simulation to model excavator operator performance. Human Factors and Ergonomics in Manufacturing & Service Industries, 20(5), 408-423.

Inagaki, T. (2006). Design of human–machine interactions in light of domain-dependence of human-centered automation. *Cognition, Technology & Work*,8(3), 161-167.

Inagaki, T. (2003). Adaptive automation: Sharing and trading of control.Handbook of cognitive task design, 8, 147-169.

Karkee, M., B. L. Steward, A. G. Kelkar, and Z. T. Kemp II. 2011. Modeling and Real-time Simulation Architectures for Virtual Prototyping of Off-Road Vehicles. Virtual Reality 15(1):83-96. DOI: 10.1007/s10055-009-0150-1.

Kirwan, B. and Ainsworth, L. (Eds.) (1992). *A guide to task analysis*. Taylor and Francis

Lewis, C. H. (1982). *Using the "Thinking Aloud" Method in Cognitive Interface Design* (Technical report). IBM. RC-9265.

Link-Belt (2009). “24 Ton Class Hydraulic Excavator Productivity and Fuel Consumption Comparison “, White paper. Retrieved November 9, 2017, from Link-Belt: https://www.lbxco.com/White-papers/240X2-Deere.pdf

Mathan, S., Dorneich, M., & Whitlow, S. (2005). “Automation Etiquette in the Augmented Cognition Context”, *Foundations of Augmented Cognition* (D.D. Schmorrow, Ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

Monga, M, M. Karkee, S. Sun, L. K. Tondehal, B. L. Steward, J. Zambreno. 2012. Real-time Simulation of Dynamic Vehicle Models using a High-performance Reconfigurable Platform. Procedia Computer Science 9:338-347.

Mensing, F., Bideaux, E., Trigui, R., & Tattegrain, H. (2013). Trajectory optimization for eco-driving taking into account traffic constraints. *Transportation Research Part D: Transport and Environment*, *18*, 55-61.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Newell, A., & Simon, H. A. (1972). Human problem solving (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Norris, W. R. (2001). A Design Framework for Qualitative Human-in-the-Loop System Development. Dissertation, University of Illinois at Urbana Champaign, Champaign, Illinois.

Norris, W. R., Zhang, Q., Sreenivas, R., & Lopez-Dominguez, J. C. (2003). A design tool for operator-adaptive steering controllers. *Transactions-American Society of Agricultural Engineers*, 46(3), 883-892.

Pear, T. H. (1927). Skill. Journal of Personnel Research, 5, 478-489.

Reyneri, L. M. 2005. An Introduction to Fuzzy State Automata, Biological and Artificial Computation: From Neuroscience to Technology, Lecture Notes in Computer Science Volume 1240(1997): 273-283.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *Systems, Man and Cybernetics, IEEE Transactions*, (3), 257-266.

Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009, April). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1819-1828). ACM.

Stanton, N. A., & Walker, G. H. (2013). Human factors methods: a practical guide for engineering and design. Ashgate Publishing, Ltd.

Storn, R. (1996, June). On the usage of differential evolution for function optimization. In Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American (pp. 519-523). IEEE.

Tang, L., Tian, L., & Steward, B. L. (2000). Color image segmentation with genetic algorithm for in-field weed sensing. Transactions of the ASAE, 43(4), 1019.

Tseng, M. M., Jiao, J., & Su, C. J. (1998). Virtual prototyping for customized product development. *Integrated Manufacturing Systems*, *9*(6), 334-343.

Wu, L., (2003). A Study on Automatic Control of Wheel Loaders in Rock/Soil Loading. Ph.D. thesis, University of Arizona.

Zhang, R., Carter, D. E., & Alleyne, A. G. (2003, November). Multivariable control of an earthmoving vehicle powertrain experimentally validated in an emulated working cycle. In *Conference paper, ASME 2003 International Mechanical Engineering Congress and Exposition. http://citeseerx. ist. psu. edu/viewdoc/summary*.

Zieba, S., Polet, P., & Vanderhaegen, F. (2011). Using adjustable autonomy and human–machine cooperation to make a human–machine system resilient–Application to a ground robotic system. *Information Sciences*, 181(3), 379-397.

Zorriassatine, F., Wykes, C., Parkin, R., & Gindy, N. (2003). A survey of virtual prototyping techniques for mechanical product development. Proceedings of the institution of mechanical engineers, Part B: Journal of engineering manufacture, 217(4), 513-530.

APPENDIX

Operator Field Observation –Interview Protocol (IRB 14-203)

The following is a list of possible interview questions. Participants will not see this list.

Operator Information

1. How many years of experience do you have as a vehicle operator?
2. What kind of training did you receive for operating vehicles?
3. How often do you receive training to update your skills?

Operations Information

4. In what capacity do you operate vehicles (e.g. single owner-operator, employed at a large company, etc.)?
5. What is the typical size of operations in which you work?
6. Do you typically work alone or as part of a team?

Equipment Information

7. What types of equipment / brands do you drive?
8. What types of equipment / brands are your favorite, if you have a favorite one? Why?
9. Do you own your equipment? If so, for how long?
10. What kinds of features are important to you as a vehicle operator?
11. What kinds of features bother you when you are operating?
12. What is it like transition from one machine to another machine?
13. Do you need to do something different for different machines? How do you adjust behaviors to fit the machine? (May be task or operation specific)

Before operation

14. What do you do to prepare for an operation? What is most important? What is most difficult?
15. What kind of information do you want to know before an operation?
16. Do you inspect the vehicle before operation? What are you looking for?

During Operation

(since we don't know what tasks they will be doing, we will adapt these general questions to be task specific as possible)

17. Can you describe the tasks / steps in the operation, in terms of procedures, subtasks, and goals?
18. What cues / feedback / triggers do you use during each to accomplish goal?
19. How long does it take for each task?
20. What control input do you use for each task?
21. How do you know when you are performing well?
22. What errors / failure / difficulties can occur?
23. Are there things that you would like to sense or control that you cannot now?
24. Do different materials affect your operation? If so, how?
25. Do environmental factors affect your operation? If so, how?
26. How long do you drive a vehicle at a time?
27. What kinds of factors can because you fatigue?
28. Does vibration of the vehicle affect your operation? In which ways?

After Operation

29. After completion the task, what do you need to do?

Task Analysis Form

Task No.	Tasks description	Time	Sub-tasks	Strategies	Cues, feedback, or triggers to accomplish the goal of each task	Operator Control Inputs	How do you know when you are performing well?

Operator Field Observation –Focused Interview Protocol

In order to understand operation details following questions were prepared for each stage of the tasks. The questions are related to cues, strategies, control inputs, failure modes, dangers, and change of the environment that can be used by operator or happen during operation. Before the interview the terminologies will be discussed with the operator in order to ensure a consistent communication. A table is created to help participants provide detail information about their operation.

Generic Question:

1. What cues do you monitor before start of [Task]?
2. What cues do you use to determine the start of [Task]?
3. What control inputs do you use when start of [Task]?
4. During [Task] what do you monitor?
5. What control inputs do you use during [Task]?
6. What strategies do you use for [Task]?
7. What cues do you use to determine the strategy?
8. What cues do you use to decide the end of [Task]?
9. What control input do you use to stop of [Task]?
10. What kind of environmental changes can affect [Task]?
11. What are the strategies to adapt these environmental changes?
12. If there is overlapping between tasks, how do you coordinate the control inputs?

Follow up Questions:

Based on participants answer to the generic question, the interview may continue with more targeted, specific questions. For instance, if the participants said in question 2 that the use the cue of hitting the bottom of the trench, to know to start bucket filling, we may ask a follow up question such as, "How do you know the Bucket is at the bottom of the trench during Bucket Filling?". These follow up question will be generated dynamically through the interview process as a direct results of their answers to the generic questions.

Please specify information about cues, strategies, dangers, failure modes, and change in environment during operation in the following table. The blank columns can be used to fill in additional information related to the operation.

Tasks	Before the Task (cues, strategies, control inputs, dangers, failure modes, and change in environment)	Start of the Task (cues, strategies, control inputs, dangers, failure modes, and change in environment)	During the Task (cues, strategies, control inputs, dangers, failure modes, and change in environment)	End of the Task (cues, strategies, control inputs, dangers, failure modes, and change in environment)	What cues do you use to determine good performance	notes
Task1:						
Task2:						
Task3:						
Task4:						
Task5:						

ISU IRB # 1	14-203
Approved Date:	16 June 2014
Expiration Date:	6 May 2016

ATTACHMENT A

CONSENT FORM FOR: OPERATOR FIELD OBSERVATION

This form describes a research project. It has information to help you decide whether or not you wish to participate. Research studies include only people who choose to take part—your participation is completely voluntary. Please discuss any questions you have about the study or about this form with the project staff before deciding to participate.

Who is conducting this study?

This study is being conducted by Iowa State University researchers, Dr. Michael Dorneich, Dr. Brian Steward and graduate student Yu Du. This project is funded by Deere &Company.

Why am I invited to participate in this study?

You are being asked to take part in this study because you are a farm equipment or construction machinery operator.

What is the purpose of this study?

This study is intended to observe operator interaction with off-road technology. This study will provide a deeper understanding of the operator's needs and behavior during operating the equipment for enabling virtual machine prototype simulation early in product design process.

What will I be asked to do?

If you agree to participate, there are several potential aspects to your participation in this study: 1) operation of machinery, 2) shadowing of operations, and 3) interviews. You may be asked to participate in one or more of these aspects.

If you agree to participate, you may be asked to operate farm equipment or construction machinery to complete a working task from your normal working practice. If you are a Deere employee who has operated machinery as part of your work duties in the past, Deere may share video and vehicle data from those tests with the research team.

ATTACHMENT A

You may be shadowed by a research team member as you complete routine activities for your work task throughout the day. As you complete individual tasks in your day, the research team member will be taking notes and asking questions.

Finally, the research team may interview you to better help us understand your process. The interview may include written or verbal responses.

As much of the session as possible (surveys and activities) will be recorded via video and audio. Asking questions during your individual tasks and the interview will not exceed 2 hours in addition to your regular work task. If you would like to stop at any point during the day, you can decide to take a break or conclude the session.

What are the possible risks and benefits of my participation?

Risks — No additional risk other than what you would normally encounter during a normal work day. If at any point you would like a break or do not wish to converse about a specific activity, the research team member will disengage.

Benefits— No direct benefit to you. However, we hope that this study will provide a deeper understanding of how professional operators prepare and use farm or construction equipment within their operation. The benefit to society is more efficient and productive agricultural equipment and construction machinery performance.

How will the information I provide be used?

The information you provide will be used for the following purposes:

Your responses will be recorded and analyzed using various programming and statistical techniques to identify tasks, needs, goals, strategies, and behavior within your operation. This information will be used to create a virtual simulation model of operators for use with vehicle simulations during the design and test phase of product development and research.

The findings will be made available to the scientific community. The Deere Company will receive a report of the results reported in summary form. If a video or audio is used in presentations or public reports, the faces will be blurred and the voice will be changed. No information will be made available that would identify you or your individual responses.

ATTACHMENT A

What measures will be taken to ensure the confidentiality of the data or to protect my privacy?

Records identifying you will be kept confidential to the extent allowed by applicable laws and regulations. Records will not be made publicly available. However, federal government regulatory agencies, auditing departments of Iowa State University, and the ISU Institutional Review Board (a committee that reviews and approves research studies with human subjects) may inspect and/or copy your records for quality assurance and analysis. These records may contain private information.

To ensure confidentiality to the extent allowed by law, the following measures will be taken: All data obtained will be kept within a controlled access lab or within the locked office of the primary investigator. All electronic data will be stored on a password protected computer. Your identity will not be associated with any data collected.

If the results are published, your identity will remain confidential.

The only exception to the above is that the research sponsor may view brief representative sample excerpts of participants' audio and video recordings. Your identity will not be shared or associated with these recordings, but you may be recognizable if someone at the research sponsor organization knows you. These excerpts may or may not be destroyed after the study.

Will I incur any costs from participating or will I be compensated?

You will not be charged for participating in this study.

What are my rights as a human research participant?

Participating in this study is completely voluntary. You may choose not to take part in the study or to stop participating at any time, for any reason, without penalty or negative consequences. Discontinuing the study will not impact the your relationship with ISU or the sponsor. The interview may include written or verbal responses. You may skip any questions that you do not wish to answer.

Whom can I call if I have questions or problems?

You are encouraged to ask questions at any time during this study.

ISU IRB # 1 14-203
Approved Date: 16 June 2014
Expiration Date: 6 May 2016

ATTACHMENT A

- For further information about the study contact Yu Du (315) 244-3452, Dr. Michael Dorneich at (515) 294-8081, or Dr. Brian Steward at (515) 294-1452.
- If you have any questions about the rights of research subjects or research-related injury, please contact the IRB Administrator, (515) 294-4566, IRB@iastate.edu, or Director, (515) 294-3115, Office for Responsible Research, 1138 Pearson Hall, Iowa State University, Ames, Iowa 50011.

Consent and Authorization Provisions

Your signature indicates that you voluntarily agree to participate in this study, that the study has been explained to you, that you have been given the time to read the document and that your questions have been satisfactorily answered. You will receive a copy of the written informed consent prior to your participation in the study.

☐ I am aware that the recordings and results of this study will be used as data available to the scientific community and the sponsor.

Participant's Name (printed)

____

(Participant's Signature) (Date)

____