COORDINATED RATE CONTROL USER INTERFACE AND TASK IDENTIFICATION OF AN EXCAVATOR

A Thesis
Presented to
The Academic Faculty

by

Beau B. Domingue

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2016

COORDINATED RATE CONTROL USER INTERFACE AND TASK IDENTIFICATION OF AN EXCAVATOR

Approved by:

Professor Wayne J. Book, Advisor
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Associate Professor Nader Sadegh
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Assistant Professor Aaron Young
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: November 29, 2016

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Excavators are ubiquitous machines used in construction and agriculture globally. Many fluid power machines, including excavators, are operated directly by humans. For this reason, the efficacy of the communication channels between the human and machine have a high impact on system performance. While current excavator controls have been mastered by experts in the field, novel methods of control can improve operator performance and accelerate the learning process for novice operators.

The current state of the art in excavator control uses 2 dual degree of freedom joysticks to control the flow to each of the excavator joints. There is a steep learning curve associated with this interface due to the large cognitive load it places on the operator. A coordinated rate control scheme was developed to alleviate the need for the operator to mentally perform the inverse kinematics of the linkage, and implemented using the standard joysticks to ensure compatibility with current state of the art hardware. In a 20 person experiment, subjects using coordinated rate control consistently removed more soil/time and soil/fuel than subjects using the conventional control.

A novel method of task identification was developed to determine which phase of a trenching cycle the excavator is in at each time step. A supervised Artificial Neural Network with eight inputs, the four joystick velocity inputs and four joint positions, is used to classify the data into one of the three phases of a trenching cycle: dig, unload, and return. The ability to segment data enabled further analysis of the controllers within each phase, and can potentially be used to change the hydraulic priorities real time or to augment the operator input to achieve an optimal command.

CHAPTER I

INTRODUCTION

Excavators are ubiquitous machines used in various industries globally. Many fluid powered systems and machines, including excavators, are operated directly by a human. Consequently, the optimality of the communication channels between the human and machine is critical for desirable system performance. While current excavator controls have been mastered by experts in the field, novel methods of control can improve operator performance and accelerate the learning process for novice operators. This research focuses on optimizing the communication channels, specifically the user interface, between the human operator and a hydraulic excavator.

The current state of the art in excavator control uses joysticks to control the flow to each of the excavator joints. There is a steep learning curve associated with this interface due to the large cognitive load it places on the operator. A better user interface, hereafter UI, would reduce the cognitive load on the operator, yielding a more intuitive control method. Intuitive controls yield safer excavator operation during the learning process due to a reduced number of erroneous commands, and more efficient and effective performance that reduces system costs.

An intuitive control method would allow the operator to use high level commands such as forward, back, up and down, while the necessary inverse kinematics calculations to achieve the desired motion are performed on a computer, rather than mentally by the operator. This method of control is known as coordinated control, and has been demonstrated to be a more intuitive and effective UI in previous studies [5, 12, 19, 20].

This thesis presents a method of optimizing the communication channels between an excavator and human operator that requires minimal modifications to the current UI hardware. In Chapter II, the conventional UI and coordinated control UI developed in prior research will be presented. Chapter III will describe the developed UI, and the necessary computation associated with the control scheme. Chapter IV discusses practical considerations for implementing coordinated control on a live excavator. Chapter V presents the results of a 20 person human subject experiment comparing conventional control and coordinated rate control. The development and performance of a task identification algorithm for an excavator will be presented in Chaper VI. The task identification algorithm is used in Chapter VII to segment the data, and analyze the performance of operators per trenching cycle. Chapter VIII introduces some other potential applications of the task identification algorithm, as well as the necessary modifications for real time implementation. Chapter IX concludes the thesis by discussing the contributions of this work, as well as further extensions of this work for future research.

CHAPTER II

BACKGROUND

2.1 Conventional Control

An excavator arm is composed of four links: the cab or swing joint, boom, stick, and bucket, shown in Figure 1. Cab rotation is hereafter referred to as swing. The stick is sometimes called the arm, but in this work the arm refers to the complete excavator arm linkage. The conventional excavator UI uses two joysticks, each with two degrees of freedom, to control the flow to each of the excavator joint pistons as seen in Figure 1. Alternatively, ISO controls typically used on backhoes have the boom and stick control switched. Most modern excavators and backhoes can switch between the two control configurations with the flip of a switch. For each, flow of hydraulic fluid yields pressure that produces torque on the joint, which induces motion. This implementation is desirable due to the simplicity of the hardware. However, there is a steep learning curve for this UI. Human operators naturally decompose tasks into cartesian space, but the transformation from motion in cartesian space to joint commands is nontrivial.

The complexity of the commands in joint space is amplified by the fact that the control vectors are not constant in reference to the operator, or even to one another. Thus, giving the same input in different arm configurations yields different end effector motion. As the arm moves through the workspace, the necessary inputs to achieve identical motion are different, as illustrated in Figure 2. The radical change in control vectors between Figures 2(a) and 2(b) shows that vectors do not remain constant in relation to each other or the operator.

To achieve a desired motion, the operator must mentally perform the inverse

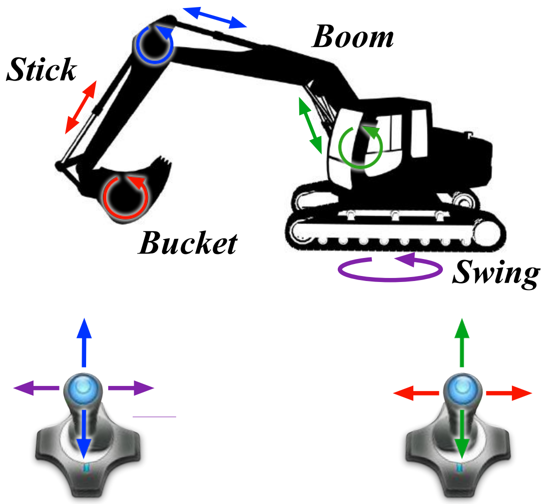
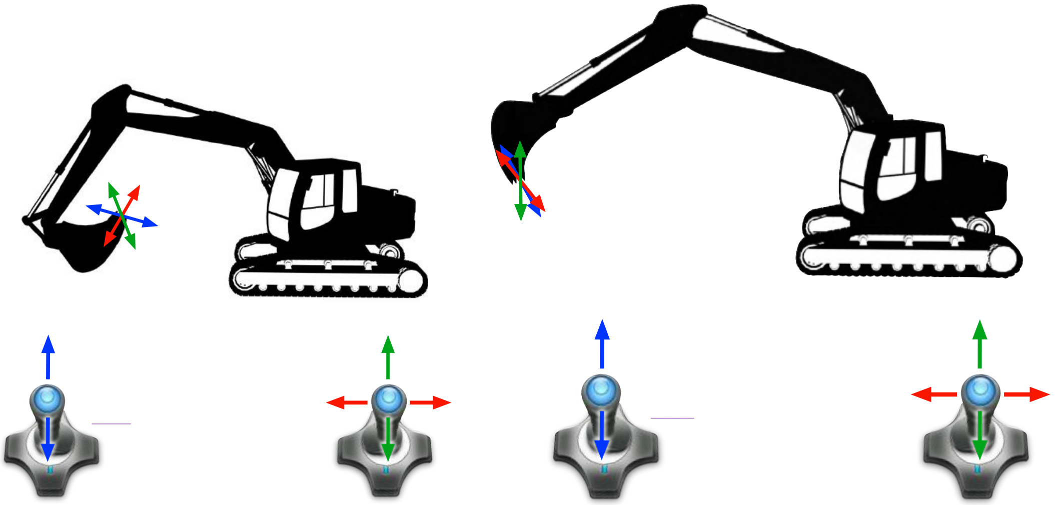


Figure 1: Mapping of joystick degrees of freedom to joint motion for the SAE conventional control scheme used on hydraulic excavators.

kinematics of the linkage. This is a process of converting desired Cartesian motion into joint angles and angular velocities, and thus joystick positions, to achieve that motion. Expert operators can perform these mental calculations near real time, but they log many hours before attaining the capability. A novice operator's confusion can be furthered by the fact that the kinematics of the linkage are dependent on the lengths of the boom, stick, and bucket. Consequently, operators must adapt to the kinematics of each new excavator they use.

2.2 Coordinated Control

In remote manipulator technology, coordinated control is when the operator controls a multi degree-of-freedom, hereafter DOF, system on a coordinate system natural to the operator. Coordinated control is imperative for manipulators with many joints, similar to the 6 degree of freedom manipulator seen in Figure 3, where joint control would be cumbersome. The point of control for a 6 DOF serial manipulator is typically the wrist. This is the joint that is connected to the end effector, which are often modular tools. Controlling the wrist in Cartesian space enables the user to position the end effector to complete the desired task with the end effector.



(a) Arm configuration with control vectors.

(b) A second arm configuration with different control vectors.

Figure 2: The control vectors change as the operator moves the arm around the workspace.

Coordinated control of an excavator end effector requires the operator to control four degrees-of-freedom: three for the end effector position in space and one for bucket rotation. Similar to the 6 DOF manipulator, The resulting UI allows operators to give high level commands in the task space of the end effector: forward, back, up, and down, alleviating the need to mentally perform the inverse kinematics.

2.3 Prior Work

2.3.1 Specialized Joysticks

This research has explored several non-traditional user interfaces that alleviate the need for the operator to mentally perform the inverse kinematics. One implementation used a Phantom Omni 6 degree-of-freedom controller, shown in Figure 4(a), to implement coordinated control where commands were given as both positions and velocities [5, 13].

Coordinated position control was implemented by mapping all of the possible end effector locations, shown in Figure 4(b), to the joystick space. So, when the operator

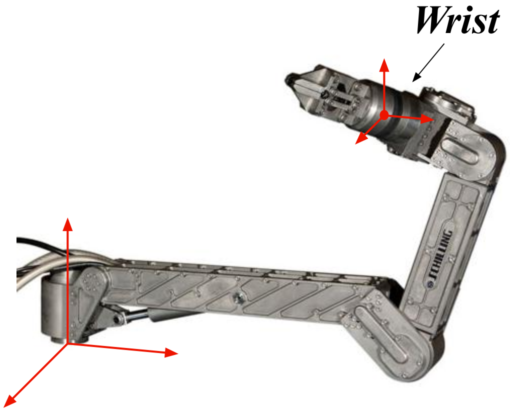
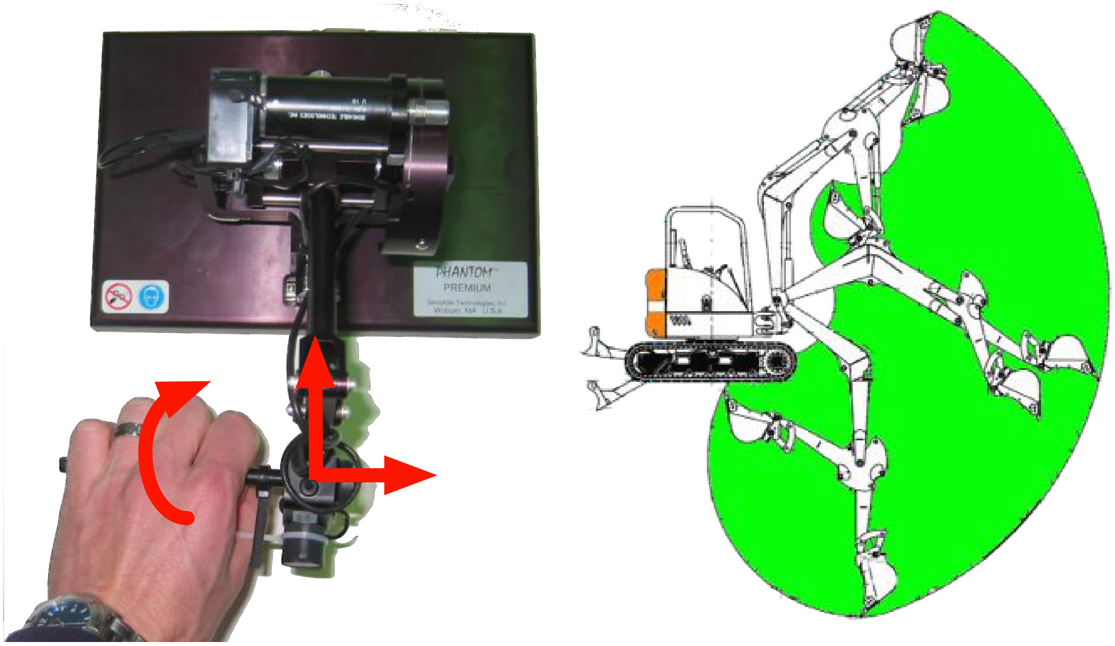


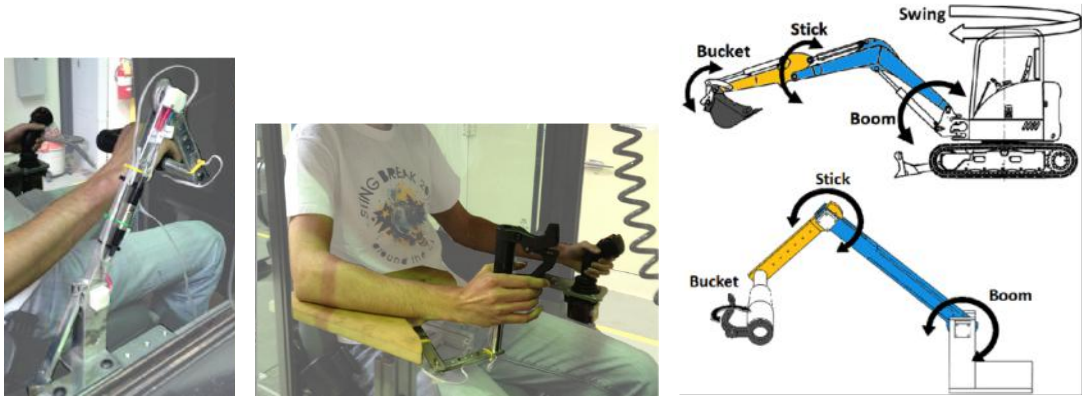
Figure 3: A 6 degree of freedom serial manipulator with a clamping end effector.



(a) Omni Phantom 6 DOF joystick.

(b) Range of motion of the bobcat excavator arm.

Figure 4: Phantom Joystick



(a) Kinematically similar joystick in upright position.

(b) Rotated kinematically similar joystick.

(c) Joystick and excavator arm in master/slave configuration.

Figure 5: Kinematically Similiar Joystick

moved the joystick to a new location, the excavator end effector would move to the corresponding location in it's workspace. Coordinated rate control was implemented by initializing a zero point for the phantom. Moving the joystick from the zero point would cause the excavator end effector to move in the same direction with a speed proportional to the distance the joystick was displaced from the zero point. Haptic feedback was used to aid the operator in returning to the zero point when desired [5].

In a human subject experiment, coordinated rate control was shown to be more effective than coordinated position control. In further studies, Elton showed that this was because the joystick dynamics are much faster than the excavator dynamics. Thus, the operator did not know what they were commanding until the arm stopped. Using coordinated rate control, operators performed 85% faster and 18% more efficiently than operators using conventional control [5, 6].

Another joystick implemented a kinematically similar position controller that performed the inverse kinematics mechanically and functioned in a master/slave configuration [20], shown in Figures 5(a) and 5(b). The joystick was designed so that the ratio between it's two links was equivalent to the ratio of the lengths of the excavator boom and stick, as seen in Figure 5(c). During longer sessions, operators suffered from fatigue issues because the interface required the user to hold their hand out in space [20]. For this reason, the joystick was rotated, as shown in Figure 5(b). In this configuration, the operator's arm was supported, and the control plane is rotated 90 degrees so that left and right of the joystick correspond to up and down of the excavator end effector. Through a human subject experiment, operators using the kinematically similar joystick removed 90% more soil/time and 44% more soil/fuel than the operators using conventional control [20].

Through human subject experiments, the two advanced user interfaces have been shown to perform better than the conventional UI, increasing operator time efficiency by up to 90% and fuel efficiency by 44% [6, 20]. Thus, these studies support the hypothesis that operators would perform more efficiently and effectively using an intuitive user interface.

While these performance increases are desirable, the specialized hardware used to obtain them impede system adoption by industry. Excavator manufacturers fear that such a change in hardware would alienate current consumers. The hardware was also fragile and expensive, and added a layer of complexity not present in current excavators. Operators also suffered from fatigue and biodynamic feedthrough because the user interfaces required the user to hold their hand out in space. Operator fatigue was especially prevalent during longer experiments [5, 20]. All of these roadblocks with the specialized hardware suggest that a simpler approach to the user interface development should be taken.

2.3.2 Joysticks

A previous study implemented coordinated control on the traditional joysticks. The mapping from joystick motion to end effector motion was identical to the rotated plane mapping from the kinematically similar joystick [17]. However, operators using coordinated rate control did not out perform operators using conventional control. It is hypothesized that the mapping used was sub-optimal, and that an optimal mapping will yield the same performance benefits demonstrated on the specialized hardware. The developed mapping will be presented in Chapter III.

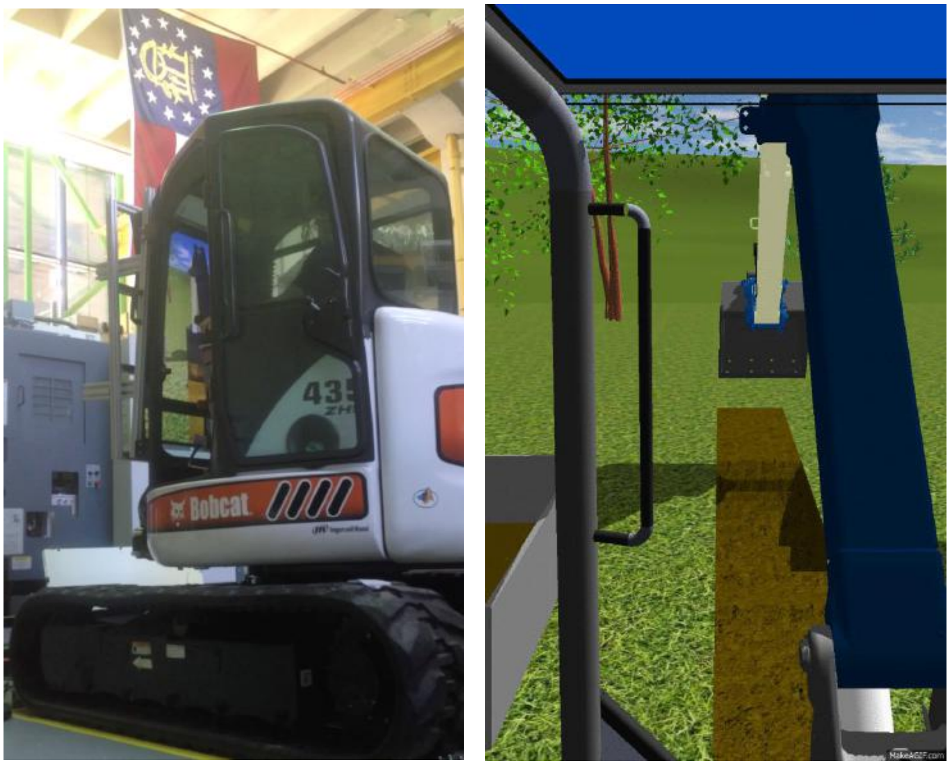
2.4 Excavator Simulator

All of the data for this work has come from an excavator simulator of a variable displacement pump controlled Bobcat 435 compact excavator. While testing should eventually be completed on live equipment, the simulator provides a cost efficient controlled environment that guarantees the system parameters are the same for each test. The simulator was designed by Elton, enhanced by Seifert, then modified for this research [7, 17].

The simulator's dynamic model of an excavator was validated against the pump controlled excavator in Zimmerman's work [21, 7]. The soil model is based on DiMaio's work in [4]. Researches at Purdue used data from a Bobcat 435 excavator to develop an empirical fuel map that was a mapping from energy needed to fuel consumed [21]. The fuel model for the simulator uses the fuel map in [21] and interpolates between points.

The simulator is a virtual environment inside of a rotating Bobcat 435 compact excavator, shown in Figure 6(a). The excavator arm has been removed, and a 3D television screen has been placed in the front window. The screen displays a 3D environment which contains the excavator arm, a trench with soil, and two bins to place the soil, shown in Figures 6(b) and 7.

Cab rotation is included to provide a more immersive experience. Excavator noise is played through a pair of headphones during operation. The volume of the noise is proportional to the amount of energy being consumed. This valuable form of feedback provides the operators with a notion of when the excavator arm is pressed against a hard stop, e.g. the ground or a bin. The simulator provides a realistic setting for



(a) Rotating Bobcat 435 compact excavator.

(b) First person perspective of the excavator virtual environment.

Figure 6: Excavator Simulator

users where repeatable tests are possible. In experiments, all subjects interact with an identical environment yielding performance metrics that are controller dependent.

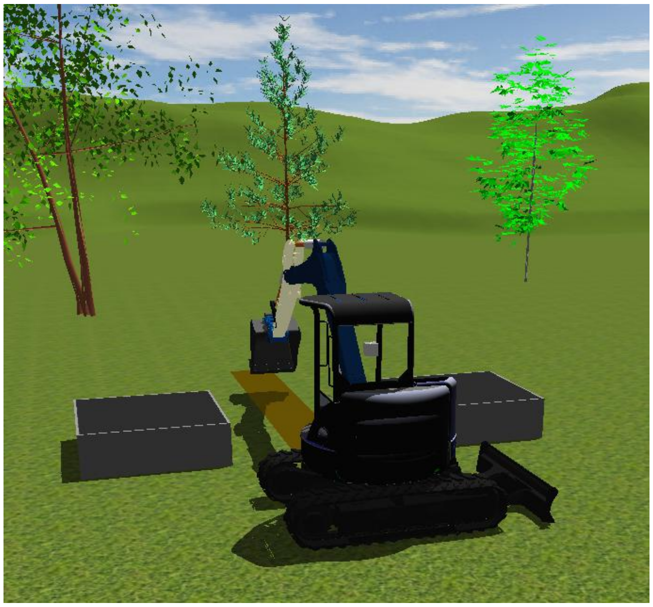


Figure 7: Virtual environment of the excavator simulator.

CHAPTER III

CONTROLLER DESIGN

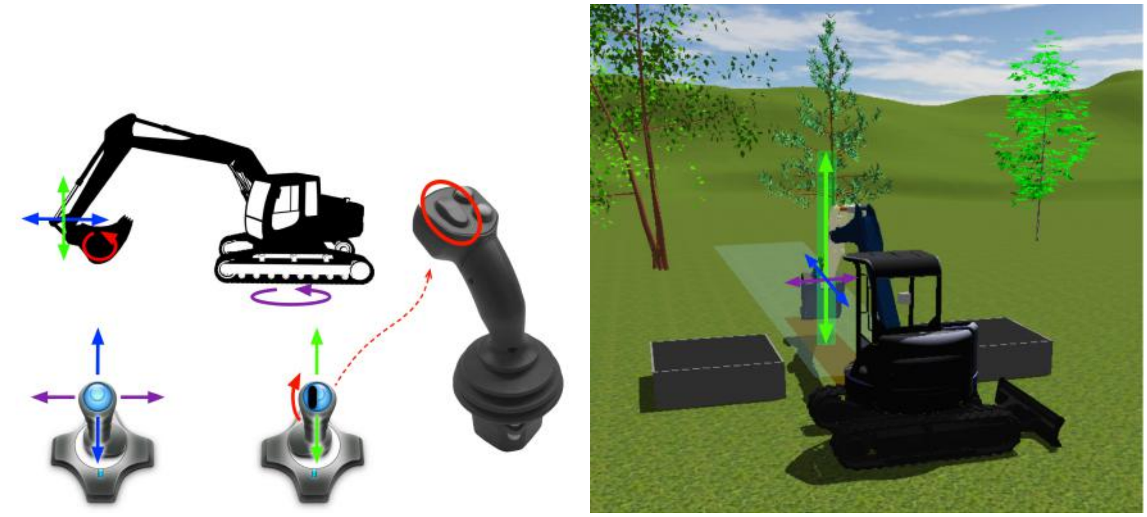
The prior work has demonstrated that coordinated control is a more intuitive interface than conventional excavator control with operators removing 90% more soil/time and 44% more soil/fuel. However, as aforementioned, the specialized hardware impedes system adoption for multiple reasons: cost, operator fatigue, complexity, and biodynamic feedthrough issues.

The ideal user interface would reap the benefits of coordinated control, and be compatible with current state of the art hardware. Thus, coordinated rate control, hereafter CRC, was implemented on the conventional dual DOF joysticks. One major advantage of implementing coordinated control on the conventional hardware is that operators would be capable of toggling between the conventional UI and CRC UI based on personal preference.

3.1 Joystick Mapping

Clearly, there are a limited number of combinations of joystick DOFs to end effector motion. The objective is to maximize operator performance through intuitiveness. This is critical because the joystick motions do not give the operator a constant reference frame like the more advanced hand controllers. A five-person pilot study was completed to evaluate the effectiveness of multiple iterations of mappings. The mapping selected for further testing yielded the most effective operator performance and was deemed most intuitive by all participants of the pilot study.

The mapping selected, shown in Figure 8(a), uses the right joystick to control the wrist's vertical position, and a continuous rocker to control the curl of the bucket. The left joystick controls the end effector in a horizontal plane, as shown in Figure 8(b).



(a) Coordinated rate control mapping with Danfoss joystick rocker.

(b) Planar mapping description

Figure 8: Coordinated rated control mapping illustration

The pilot study participants noted several attributes they found particularly intuitive. The group consensus was that using the rocker to control the bucket was a large improvement over a left/right DOF of a joystick. Participants felt they made fewer incorrect commands using the rocker to control the bucket. This UI also smooths the transitions from the conventional UI to the CRC UI because the two control mappings are similar. In most arm configurations, moving the boom and stick using the conventional UI correspond to up/down and forward/back, respectively. This is exactly what CRC does for the same joystick DOFs. The distinct advantage of CRC is when the operator is performing a grading task, hoisting task, or any other task when straight line motion is a mandate. It is possible that expert operators would switch to coordinated control intermittently, depending on the task at hand.

3.2 Coordinated Control Derivation

3.2.1 Excavator Kinematics

The kinematics of the excavator can be broken up into two parts. Using the nomenclature of robotic manipulators, the states of the excavator can be defined in both task space and joint space, as well as cylindrical space which is not derived in this work. In task space, the states of the excavator are defined in a Cartesian frame. In joint space, the states are defined by joint positions, angular velocities, and angular accelerations.

Using the Denavit-Hartenburg parameters of the excavator manipulator, the governing kinematic equations of the excavator can be derived. First, some nomenclature will be introduced.

Denavit-Hartenburg parameters:

a_i: The length of link i.

d_i: The joint offset between links $i-1$ and i. (These do not appear in derivation, but are included for completeness)

α_i: The constant twist angle between links i and $i+1$. (These do not appear in derivation, but are included for completeness)

θ_i: The variable angle between links i and $i+1$.

P_{ij}: The position vector between origins O_i and O_j.

c_i: $cos(\theta_i)$

c_{ij}: $cos(\theta_i + \theta_j)$

c_{ijk}: $cos(\theta_i + \theta_j + \theta_k)$

s_i: $sin(\theta_i)$

s_{ij}: $sin(\theta_i + \theta_j)$

s_{ijk}: $sin(\theta_i + \theta_j + \theta_k)$

The excavator linkage consists of the cab and the arm. However, the intention of deriving the kinematic equations is to enable task space commands for the manipulator. The swing joint is used to position the arm so that it is in the desired plane. Thus, it is decoupled from the arm and controlled independently. This simplifies the linkage to a 3 link planar manipulator with revolute joints. The Jacobian of the manipulator is dependent on the Denavit-Hartenburg parameters of the excavator arm. The position vectors of the excavator arm are shown in Equations 1 - 4.

$$P_{00} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

$$P_{01} = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \tag{2}$$

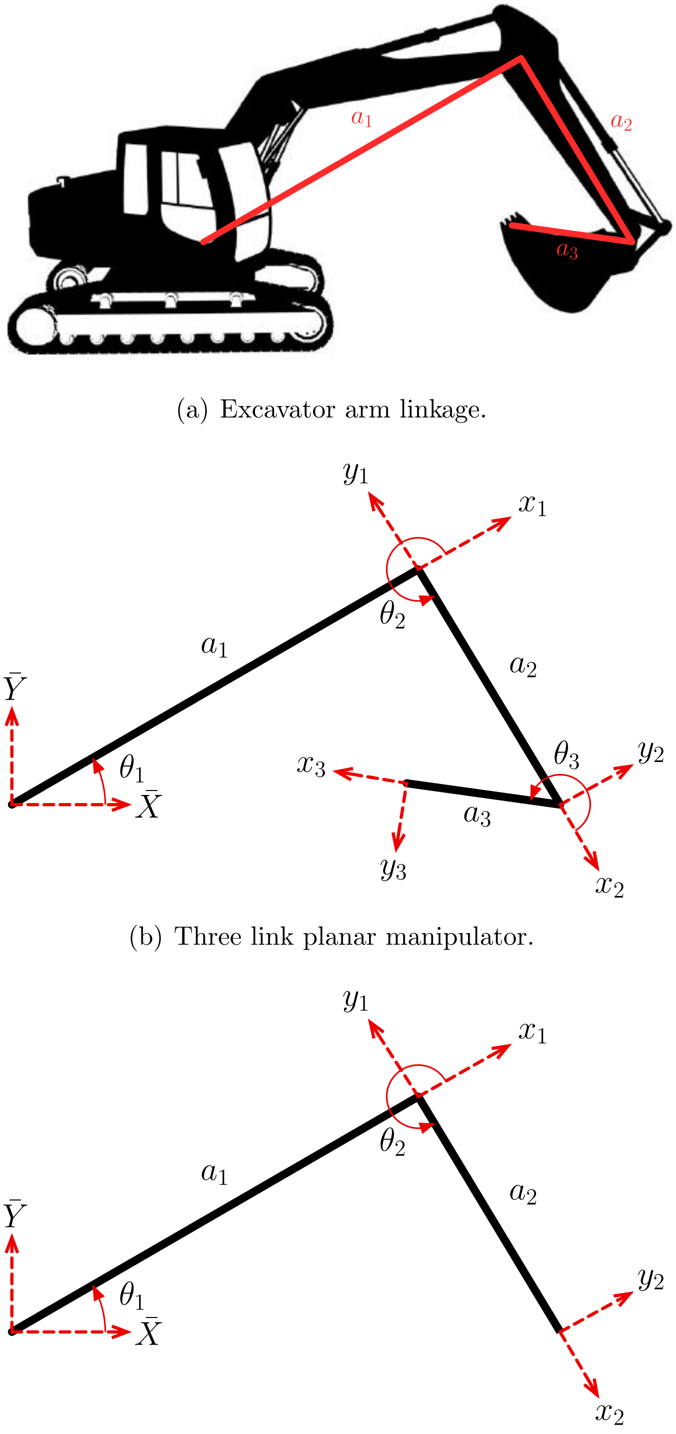
$$P_{02} = P_{01} + \begin{bmatrix} a_2 c_{12} \\ a_2 s_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix} \tag{3}$$

$$P_{03} = P_{02} + \begin{bmatrix} a_3 c_{123} \\ a_3 s_{123} \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix} \tag{4}$$

The standard robotics Jacobian of the 3 link excavator arm is defined in Equation 5.

$$J = \begin{bmatrix} z_0 \times (P_{03} - P_{00}) & z_1 \times (P_{03} - P_{01}) & z_2 \times (P_{03} - P_{02}) \\ z_0 & z_1 & z_2 \end{bmatrix} \tag{5}$$

The z axes of each joint are equivalent because the three revolute joints are coplanar.



(a) Excavator arm linkage.

(b) Three link planar manipulator.

(c) Two link planar manipulator.

Figure 9: Excavator arm linkage with Denivit-Hartenburg parameters labeled.

$$z_0 = z_1 = z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

However, it is undesirable to make the tip of the bucket the point of control. Control using the tip of the bucket would be extremely unintuitive, and would make operators less effective. As discussed in Chapter II, it is often more intuitive to control the wrist of the manipulator. Thus, the wrist of the excavator arm, or O_2 in Figures 9(b) and 9(c), is the point to be controlled. This is a more intuitive implementation. Similar to the human arm, the wrist can be manipulated to the desired position, and the end effector (the hand or excavator bucket) is used to complete a task. Thus, both the swing and the bucket are decoupled from control of the excavator arm. The linkage is simplified to 2 links further simplifying the kinematic equations. The Jacobian of the 2 link planar excavator arm is defined in Equation 6

$$J = \begin{bmatrix} z_0 \times (P_{02} - P_{00}) & z_1 \times (P_{02} - P_{01}) \\ z_0 & z_1 \end{bmatrix} \tag{6}$$

This Jacobian is a 6 × 2 matrix. Rows 4 through 6 correspond to the angular velocities of the links. Thus, these parameters have no significance because commands are only being issued as Cartesian velocity commands, not angular velocity. The resulting Jacobian is shown in Equation 7.

$$J = \begin{bmatrix} z_0 \times (P_{02} - P_{00}) & z_1 \times (P_{02} - P_{01}) \end{bmatrix} \tag{7}$$

Computing the Jacobian in Equation 7 and dropping the third row, which corresponds to velocity in the z direction that is to be controlled by the swing, yields a 2 × 2 nonsingular rank 2 Jacobian.

$$J = \begin{bmatrix} -a_1s_1 - a_2s_{12} & -a_2s_{12} \\ a_1c_1 + a_2c_{12} & a_2c_{12} \end{bmatrix} \tag{8}$$

The Jacobian is used to map between the task space and the joint space of the excavator arm. This can be seen in Equation 9 where $\dot{\theta}$ are the joint velocities and u_v are the Cartesian velocity commands.

$$\dot{\theta} = \mathbf{J}^{-1} \cdot \mathbf{u_v} \tag{9}$$

3.2.2 Coordinated Rate Control

The inverse kinematics of the 2 link excavator arm can be computed using Equation 9. Thus, the computer would use Equation 10 to compute the necessary joint rates subject to the operator's commands.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -a_1s_1 - a_2s_{12} & -a_2s_{12} \\ a_1c_1 + a_2c_{12} & a_2c_{12} \end{bmatrix}^{-1} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{10}$$

From the operator's perspective, the joystick commands, $u_{+x}, u_{-x}, u_{+y}, u_{-y}$ correspond to forward, back, up, and down respectively. These commands are normalized between -1 and 1, and multiplied by a max linear speed in meters/sec. Commanding maximum speed in the x direction corresponds to $u_x = 1$; maximum speed in the $-x$ direction is $u_x = -1$, as illustrated in Figure 10. The computer on the excavator would use the measured boom and stick angles and the operator joystick inputs to compute the necessary boom and stick angular velocities. Then, mapping the joint rates from joint space to cylinder space enables computation of the necessary flow to achieve the desired angular velocities. More information on the mapping from joint space to cylinder space can be found in [8].

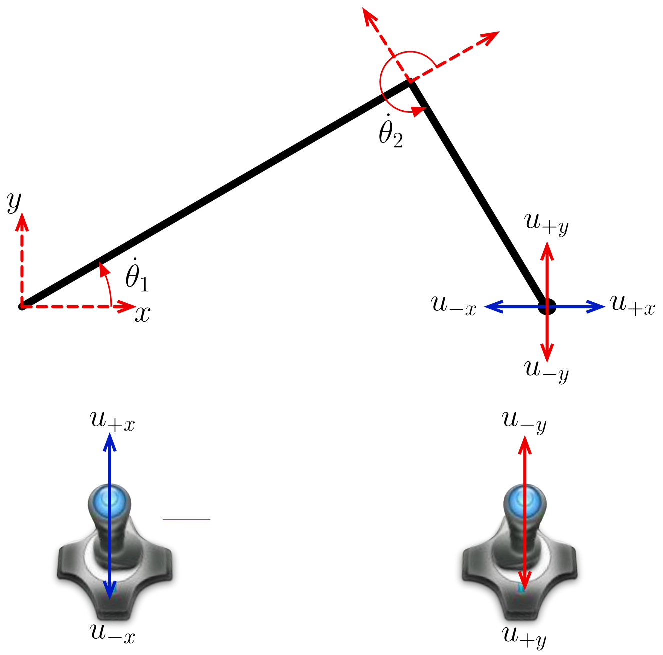


Figure 10: Joystick commands where $u_{+x}, u_{-x}, u_{+y}, u_{-y}$ correspond to forward, back, up, and down respectively.

CHAPTER IV

PRACTICAL CONSIDERATIONS

To implement coordinated rate control on an excavator, multiple issues would need to be addressed. This chapter discusses the specific hardware and software components necessary to implement an effective and safe user interface.

4.1 Computer and Sensor

Excavators use various methods for internal microprocessor communication. To implement coordinated control on a live machine, a computer would need to be placed between the joysticks and the rest of the network. While operating in conventional control mode, the computer would simply receive the joystick signals and broadcast the unconditioned signal as joint commands. In coordinated rate control mode, the computer would use the joystick signals to compute the inverse kinematics of the excavator linkage, shown in Equation 10 in the previous chapter, and broadcast the resulting flow commands. In this implementation, the operator could select the control mode using a switch, as shown in Figure 11. To compute the inverse kinematics, the current joint angles must be measured. The joint angles could be acquired by measuring the joint angle directly, or calculated from the piston position. Alternatively, a camera could be mounted to the excavator, and the joint angles could be realized via computer vision [16].

4.2 Kinematic Constraint

During the pilot study discussed in Chapter 3, participants noted an unexpected behavior of the end effector when using CRC. The user would input a forward command, and the end effector would move forward. However, the end effector would eventually

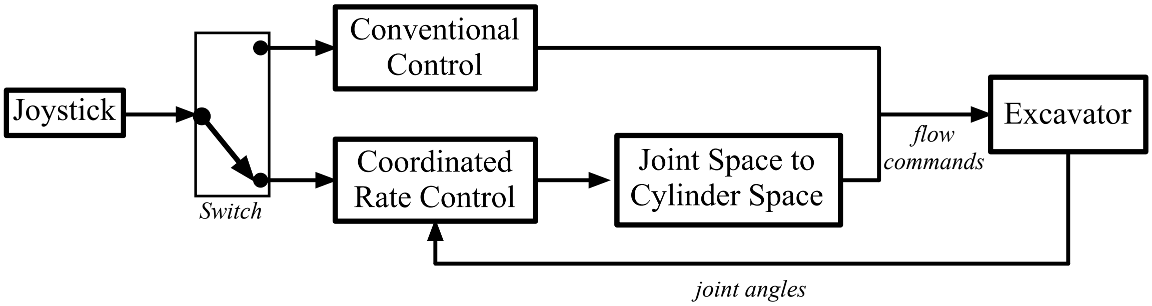


Figure 11: A simplified block diagram that illustrates how an operator could switch between control schemes using a switch.

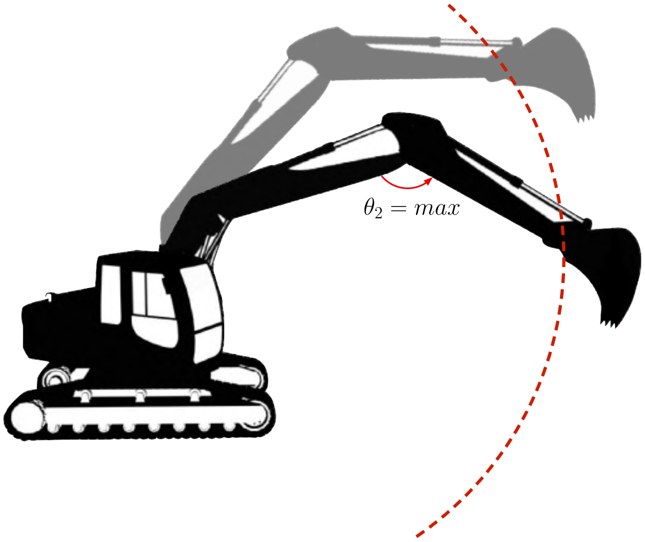
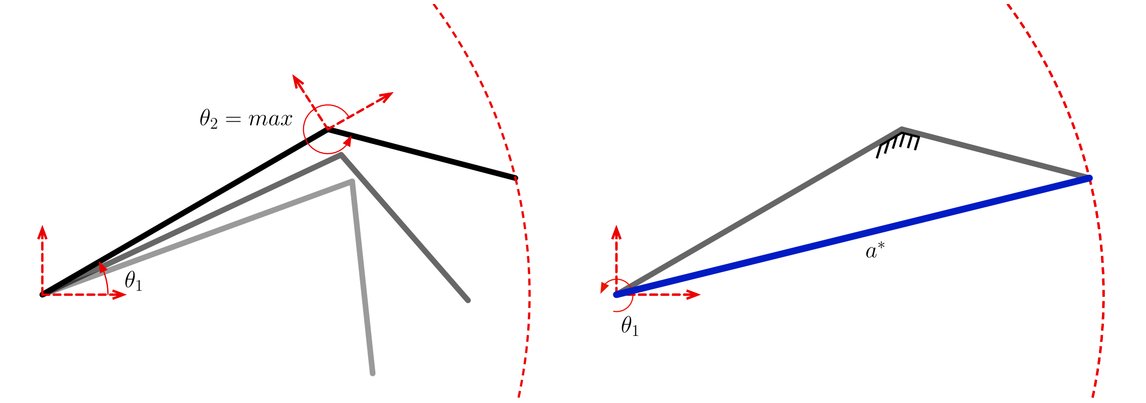


Figure 12: Kinematic constraint of the excavator arm when stick is fully extended.

encounter a kinematic constraint, shown in Figure 12, and begin to descend. The dotted radius in Figure 12 shows the end effector's range of motion when the stick is at it's maximum angle. In the configuration depicted by the middle dark link, a forward command would force the end effector to follow this radius until it reached a critical point in the center, and could no longer move strictly forward.

The excavator arm encounters a kinematic constraint when the stick rotates to it's maximum attainable angle, shown in Figure 13(a). In this situation, the arm is constrained, and the excavator's degrees-of-freedom is reduced from 4 to 3. Thus, in the kinematic constraint, the swing, boom, and bucket joints are still free to move.



(a) Kinematic constraint of the excavator arm when stick attains maximum angle

(b) Constrained linkage with effective link length a^*.

Figure 13: Diagram of the kinematic constraint

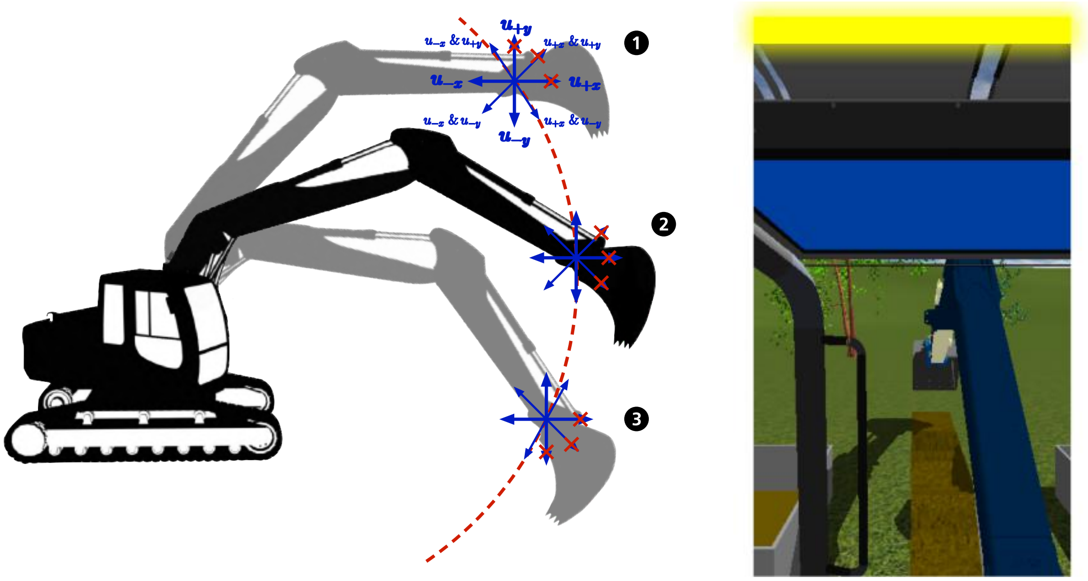
In this configuration the boom and stick links behave like a single rigid body with length a^* in Equation 11, illustrated in Figure 13(b).

$$a^* = \|P_{02}\| = \left\| \begin{bmatrix} a_1 c_1 + a_2 c_{12_{max}} \\ a_1 s_1 + a_2 s_{12_{max}} \end{bmatrix} \right\| \tag{11}$$

In practice, unexpected motions of the end effector would be dangerous. It is critical to ensure that the end effector's motion matches the operator's intended motion. If the operator is issuing a command, the command will only induce motion if the motion is possible given the current arm position in the workspace. In other words, a forward command will induce forward motion only if the arm is capable of moving strictly forward. Otherwise, identical commands would yield different motions depending on the arm position in the workspace.

The limitation in acceptable operator commands when subject to the kinematic constraint is illustrated in Figure 14(a). The command $u_{+/-axis}$ corresponds to a velocity command in the positive or negative direction, while u_x & u_y is when both commands are being issued. In position 1 in Figure 14(a), the boom angle is large enough so that the end effector is above the kinematic constraint's critical point. In this arm position, the forward command and up command are not sensible because the motion cannot be completed by the arm. Thus, before computing the joint rates using Equation 10, the software checks if the arm is in the constrained position. If the arm is constrained, as it is in the figure, then $\dot{\theta}_2 = 0$ because it cannot extend any further, and $\dot{\theta}_1$ is calculated only if the motion is possible, i.e. the commands without red X's in Figure 14(a). In position 1, the commands forward, up, and a combination of forward & up would yield $\dot{\theta}_1 = \dot{\theta}_2 = 0$. Down, back, and other permutations of the commands are still acceptable because they do not conflict with the end effector's range of motion in this position. Other contradicting combination commands, such as up & down, would yield zero motion. The same rationale can be applied to positions 2 and 3.

Due to the change in the system's capabilities, a light is used to alert the operator of the kinematic constraint. When the stick reaches the maximum attainable angle, a light appears at the top of the screen, as shown in Figure 14(b). In a live excavator, this notification could be replicated using an LED in the operators line of sight inside the cab, on the boom itself, or another type of feedback such as joystick vibration. While simple, the feedback to the operator is necessary to maximize intuitiveness of the user interface. For a trenching task, the kinematic constraint does not have a large effect on the operator's use of the machine. For other excavator tasks such as craning, it is imperative that the arm move as the operator intends.



(a) The valid and invalid commands when subject to kinematic constraint.

(b) Yellow Light Notification

Figure 14: The feasible commands when at a mechanical hard-stop, and the yellow light used to notify the operator of the constraint.

CHAPTER V

HUMAN SUBJECT EXPERIMENT

The proposed CRC UI has been evaluated through a 20 person human subject experiment on the Georgia Tech Excavator Simulator. Experiment participants used the conventional control UI or CRC UI to complete a trenching task. The data collected during the experiment is used to evaluate the operators' performance. This chapter will compare how operators performed with each of the controllers using high level performance metrics. The high level metrics evaluate performance by analyzing the gross trial quantities of interest: total soil removed and total fuel consumed.

5.1 Experiment Structure

The experiment structure is shown in Table 1. The subjects are separated into two groups of 10. All subjects in both groups completed three sessions. Each session consists of a 5 minute warm up and three 5 minute trials. Group A uses the conventional control mode for the first two sessions, then switches to CRC for the third and final session. Subjects in Group B use CRC for all three sessions. This structure was selected because each of the prior experiments testing different UIs have evaluated the performance of novice operators using conventional control. This experiment is

Table 1: Experiment structure

<table>
<tr><th></th><th>Session 1</th><th>Session 2</th><th>Session 3</th></tr>
<tr><th></th><th>warm-up | trial1 | trial2 | trial3</th><th>warm-up | trial1 | trial2 | trial3</th><th>warm-up | trial1 | trial2 | trial3</th></tr>
<tr><td><b>Group A</b></td><td>Conventional</td><td>Conventional</td><td><i>CRC</i></td></tr>
<tr><td><b>Group B</b></td><td><i>CRC</i></td><td><i>CRC</i></td><td><i>CRC</i></td></tr>
</table>

structured to compare the two control schemes through the first 6 trials, and to evaluate the ability of novice operators to transition from conventional control to the CRC UI. Group B uses CRC for all 9 trials to develop an understanding of the beginning of the learning curve associated with the CRC UI.

At the beginning of the first session, subjects read and signed a consent form that details the experiment procedure and process. Then, the subjects filled out a survey, shown in Appendix A, describing their previous experience with excavators, backhoes, and the excavator simulator. Before each session, subjects were instructed to remove soil from the trench and dump the soil into a bin on the left side of the trench. Excavated soil that was mistakenly dumped outside the bin is not counted. Subjects were told that the objective is to remove as much soil as possible throughout the duration of each trial while maintaining control of the excavator.

The subjects then enter the excavator cab, and the control scheme they were using was described using printed versions of Figures 1 and 8(a). Then, the image was placed in view inside the excavator. It is typical for excavators to have the joystick mapping to end effector motion marked near the arm rest. Next, with the simulator running, each joystick was moved independently to demonstrate how the simulator and joystick mapping works. Before beginning the trial, the operators are shown the switch that can be used to turn off the powered cab rotation if necessary. Finally, the subjects were reminded of each trial's duration, the experiment objective, and told to begin their warm up.

After the subject completed the 5 minute warm up, the software would reset the simulator. The subjects were given 1 minute to rest before the start of the first trial. This process repeated through the third trial, completing the session. Following each session, the subjects filled out a survey about how they perceive their own capabilities using the UI, shown in Appendix A.

5.2 Subject Information

Subject participants were recruited using flyers and online flyer postings. As subjects arrived for the experiment, they were placed into the opposite group of the previous subject. There were a total of 17 males and 3 females who participated, with at least 1 female in each group. The average age of all subjects who participated is 24. In the initial pre-session survey shown in Appendix A, subjects were asked the following:

- Are you left or right handed?
- How many hours have you operated an excavator or backhoe?
- How many hours have you operated *this* simulation before today?
- How often do you use a joystick (playing video games, other simulators)?
- How would you rate your level of experience in playing with or operating joysticks?

The results of these questions are tabulated below. Table 2 shows that all left handed participants were in Group A, which was not intentional. Handedness should not have any significant affect on operator performance; both hands must be used independently. Table 3 shows the number of hours that subjects have previously operated a live excavator. There are 5 subjects from Group A in the experiment with minimal experience and 4 with zero experience. The subject in Group A who indicated greater than 10 hours of experience operating an excavator is an expert operator who currently works full time operating a backhoe. Only one of the subjects in Group B had any experience with an excavator. Only 1 subject, who is in Group A, had any previous experience with the simulator, as indicated by Table 4. Thus, the participating subjects, except the expert, had little to no consequential experience and are novice operators with zero bias. While the developed control scheme should be tested amongst a group of experts, it is beneficial to explore the effectiveness of coordinated rate control using a sample of individuals with no prior disposition towards either of the control schemes.

The results of this experiment illustrate the intuitiveness of each controller because the novice operators will have a fresh perspective on each control scheme. The results also reveal the rate of learning with each UI, and show how operators perform during the learning process. An experiment using expert operators would allow for more direct claims about the immediate performance benefits of the CRC UI in an industrial setting. Tables 5 and 6 show there is a similar distribution of joystick users in each group.

Table 2: The number of subjects with left and right dominant hands in each group

	Left Handed	*Right Handed*
Group A	3	7
Group B	0	10

Table 3: Number of hours subjects from each group have operated a live excavator.

	$0\ hrs$	$0-2\ hrs$	$2-5\ hrs$	$5-10\ hrs$	$>10\ hrs$
Group A	4	4	1	0	1
Group B	9	0	1	0	0

Table 4: Number of hours subjects from each group have operated the Georgia Tech excavator simulator

	$0\ hrs$	$0-2\ hrs$	$2-5\ hrs$	$5-10\ hrs$	$>10\ hrs$
Group A	9	0	1	0	0
Group B	10	0	0	0	0

Table 5: The frequency that subjects indicated they use joysticks while playing video games, other simulators, etc.

	Never	*<Monthly*	*Monthly*	*Weekly*	$> \frac{Once}{Week}$	*Daily*
Group A	3	4	1	0	0	2
Group B	3	2	2	2	0	1

Table 6: The level of each subject's perceived abilities using a joystick from novice (1) to expert (5).

	Novice				*Expert*
	1	2	3	4	5
Group A	1	3	4	1	1
Group B	2	1	3	3	1

5.3 Results

During the experiment, the amount of soil excavated and dumped into the bin, the fuel consumed, the joystick commands, and the excavator joint positions were recorded. Multiple metrics are used to measure operator performance: time efficiency, fuel efficiency, and idle time.

5.3.1 Time Efficiency

Time efficiency corresponds to the main objective identified to the subjects before each session: the total amount of soil removed from the trench and placed into the bin each 5 minute trial. Figure 15 shows the total mean time efficiency of each controller on the left of the dashed line, and each trial's mean time efficiency for each controller on the right. The error bars represent the standard deviation of each trial's data set. This figure format will be used multiple times. Trials 1-3 and 4-6 correspond to Sessions 1 and 2. It is clear by observation that subjects using the CRC UI outperform the subjects using the conventional UI. On average, the subjects perform 13.6% more effectively using CRC than subjects using the conventional control. Superior

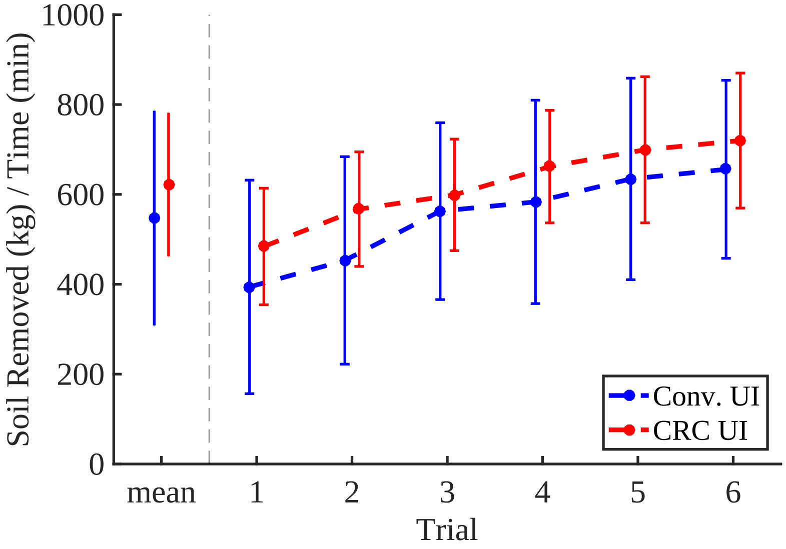
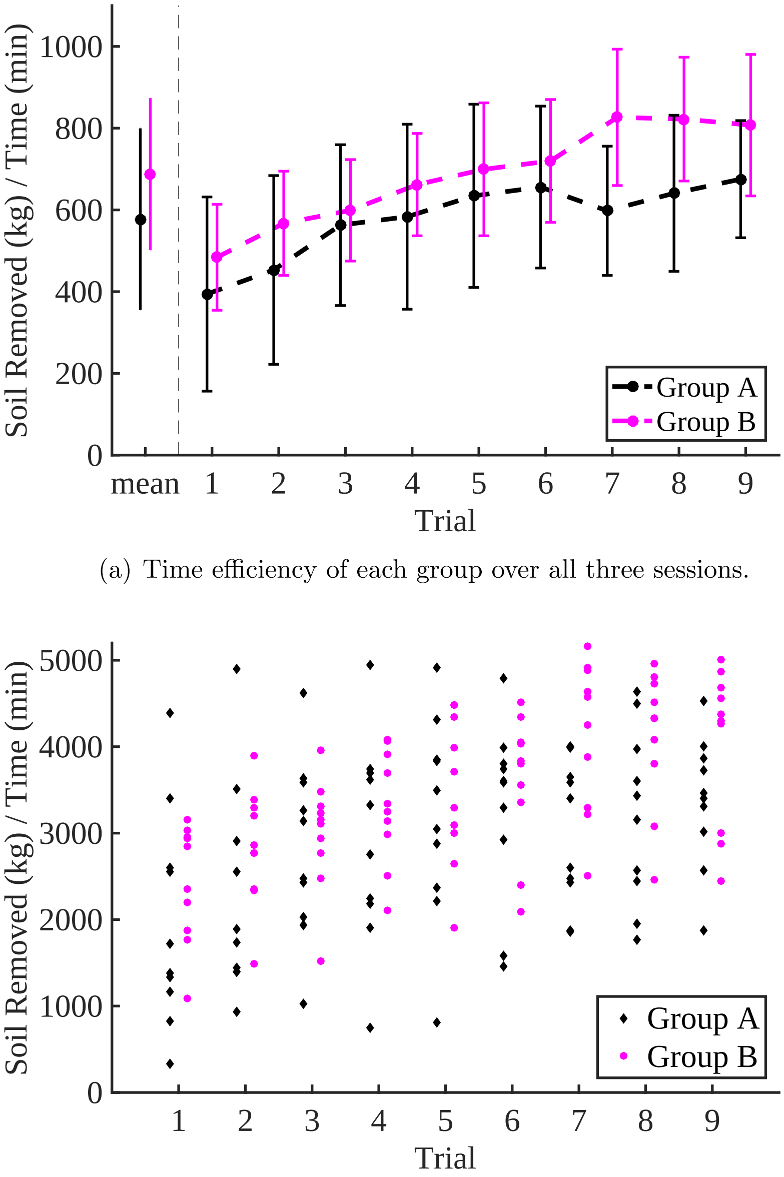


Figure 15: Time efficiency total mean on left of dashed line and individual trial means on right. Error bars represent the standard deviation of each trial's distribution.

performance of CRC is consistent with the prior studies using specialized hardware, but previously subjects showed an 84% and 90% increase in soil removed [5, 6, 20]. It is theorized that the specialized joysticks attained much higher performance benefits due to the highly intuitive physical hardware. The continuous reference frame the multi-degree of freedom joystick provides the operator is significantly more intuitive than the joysticks.

Figure 16(a) shows the performance of groups A and B across all 9 trials. Recall that group A uses conventional control for the first 6 trials, then switches to CRC on trial 7. It is interesting to note that the subjects' performance did not drop off much after the transition. With no previous exposure to the UI, the subjects were able to quickly improve so that they performed very similarly to their second session with conventional control. This indicates that the transition from conventional control to CRC is relatively straightforward. Group B uses the CRC UI for all 9 trials. The curve for group B in Figure 16(a) is the beginning of the learning curve associated with the CRC UI. Further tests would be necessary to analyze the full duration of the learning period.



(a) Time efficiency of each group over all three sessions.

(b) Time efficiency of each individual subject for each of the nine trials.

Figure 16: Time efficiency across all three sessions of the experiment. Group A switched from conventional control to CRC for the last three trials. Group B used CRC only.

Another interesting result is the large difference between the two groups' standard deviations; group A's standard deviation is much larger than that of group B. Figure 16(b) shows the performance of each subject in groups A and B across all 9 trials. It is visibly apparent that the subjects in group B are more tightly clustered. For trials 1 to 6, group A's average standard deviation is 46% greater than that of group B. Initially, this appears to suggest that the two groups are sampled from different populations. However, the mean trial standard deviation of Group A decreases by 25% after switching to CRC in trial 7, which resembles the trial distributions of group B. This supports that the two groups are from the same population, and that the CRC UI is the cause of the smaller standard deviation. The concentrated nature of the distribution suggests that there is a naturally intuitive element of the CRC UI that all of the subjects are able to grasp. The smaller standard deviation is also beneficial since the UI yielded more effective performance.

5.3.2 Fuel Efficiency

The amount of fuel a subject consumes during operation is also a critical measure of performance. This metric in particular has a high impact on the commercial viability of a UI. The fuel efficiency of each controller is shown in Figure 17. Users were able to consistently remove more soil per unit of fuel using the CRC UI. On average, users removed 10% more soil per unit fuel with CRC. Thus, the increase in the soil removed using CRC is not wholly due to higher fuel consumption, but is an artifact of using the excavator more effectively. This means that subjects used slightly more fuel using CRC each trial, but removed an even greater amount of soil yielding a higher fuel efficieny. The fuel efficiency metric is highly correlated to the time efficiency metric, as can be seen by the similarity of the curves in Figures 15 and 17. The increase in fuel efficieny is also consistent with previous experiments which showed 18% and 44% improvements in fuel efficiency using coordinated control on specialized hardware

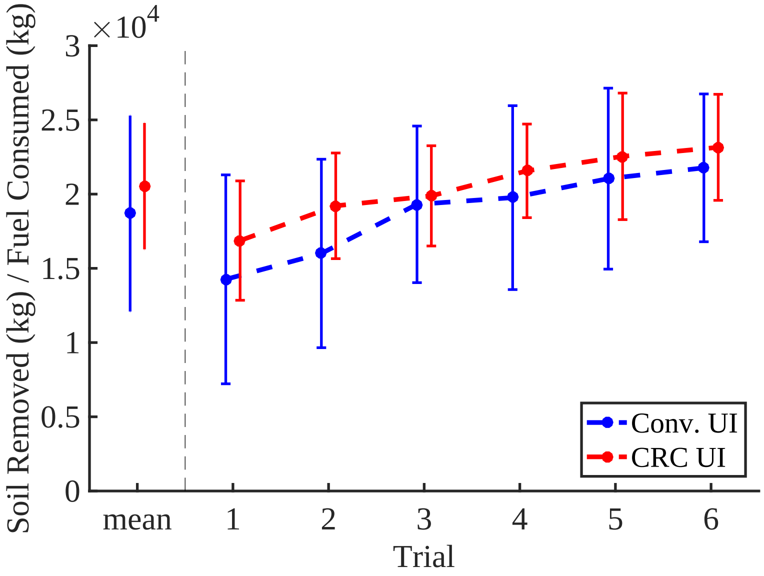
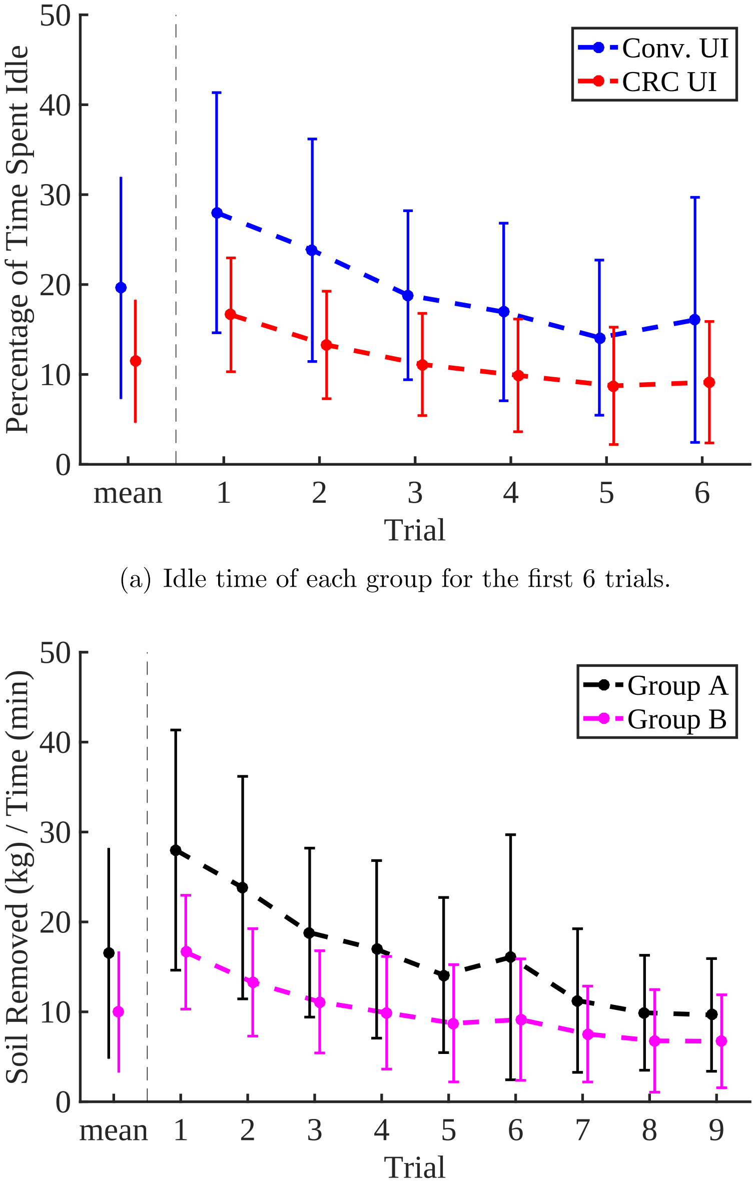


Figure 17: Fuel efficiency total mean of each controller on left of dashed line and the individual trial means on the right. The error bars represent the standard deviation of each trial's distribution.

[6, 5, 20]. Similar to before, it is theorized that the gap is due to the lack of the continuous reference frame the specialized hardware provides.

5.3.3 Idle Time

Figure 18(a) shows the amount of time users were idle during the experiment. Idle time is defined as the amount of time that the system is stationary and the user is not issuing any commands. In other words, the time users spend planning the next command sequence to achieve the desired motion. Using CRC, the users spend 41% less time idle on average. This suggests that the CRC UI is a more intuitive interface. This result is consistent with the previous experiment that implemented CRC on the joysticks [17]. Additionally, Group A's mean idle time decreased by 34.5% between sessions 2 and 3 after switching to CRC, as shown in Figure 18(b). The operators' natural tendency to decompose motion to the cartesian coordinate system yields shorter null periods for planning when using CRC. This drop in idle time further emphasizes the intuitiveness of the coordinated control method.



(a) Idle time of each group for the first 6 trials.

(b) Idle time of each group across the experiment. Group A idle time decreases 34.5% after switch to CRC.

Figure 18: Idle time total mean on the left of dashed line and the individual trial means on the right. The error bars represent the standard deviation of each trial's distribution.

5.3.4 Statistical Analysis

Analysis of Variance (ANOVA) was used to compare performance metrics for each controller. The analysis found that the 13.6% difference in time efficiency defined by the 90 individual trials of each group was significant with a p value of .04. The 10% difference in fuel efficiency was not statistically significant with a p value of .073. The 41% difference in idle time was found to be significant with a p value << .001.

However, because the subjects repeat the trials, they are not strictly independent data points. To reduce the variability between data points, ANOVA was then run on the distribution defined by the mean of each subject for each respective metric. In this case, the only metric found to be statistically significant is the difference in idle time between the two controllers, with a p value of 0.045. The time efficiency difference yielded a p value of 0.36, and fuel efficiency a p value of 0.40.

The observed Cohen's d effect size for the time efficiency metric is 0.38, which is considered small according to standard convention. The observed effect size can be used to compute the number of subjects an experiment would need to use to ensure sufficient statistical power. For the observed effect size, a sample size of 220 subjects would be necessary to achieve 80% power for a significance criterion of 0.05.

5.4 Survey Results

Following each session, the subjects filled out a post-session survey about the control scheme, shown in Appendix A. The final post-session survey asks the following questions to both groups and includes an additional field for group A subjects to respond to questions relative to each controller:

- 1 - How difficult were the controls to learn? *(rated on scale from 1 to 5)*
- 2 - How accurately were you able to control the position of the bucket? *(rated on scale from 1 to 5)*

- 3 - How confident are you of your ability to operate this excavator simulator? *(rated on scale from 1 to 5)*
- 4 - Additional Comments?

The results of these surveys are shown in Figures 19 and 20. Figure 19(a) shows group A's survey responses to question 1. The graph indicates that 8 of the 10 subjects found that the coordinated rate control scheme was easier to learn. This response is consistent with the other metrics previously discussed about Group A's switch to CRC; namely the improved performance, rapid learning, and decrease in idle time. Additionally, the expert operator found that the coordinated rate control scheme was easier to learn than the conventional control. Group A's preference of learning CRC furthers the claim that CRC is the more intuitive of the two control schemes.

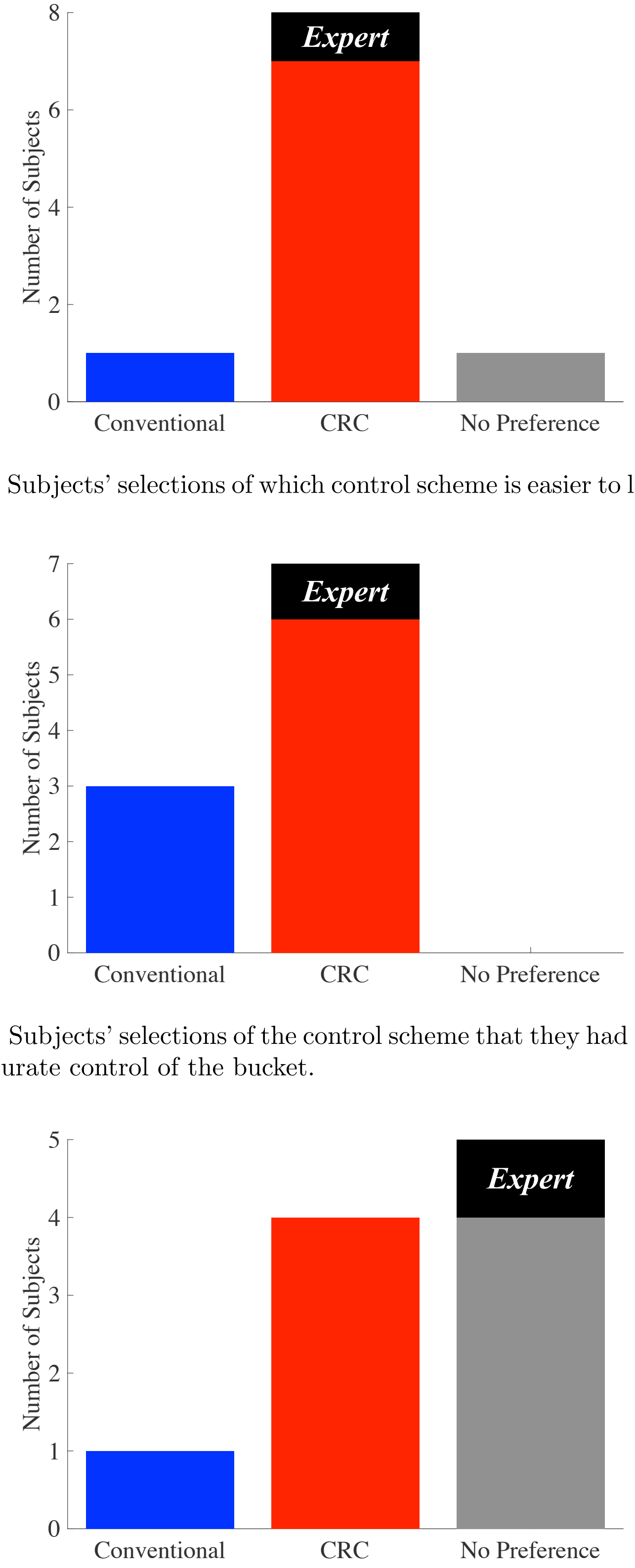
The expert operator noted on the survey that he operated a backhoe full time, but used the ISO controls. The conventional controls implemented for this work are the SAE controls. The two controls are similar, but have the boom and stick motions on the opposite joysticks. Typically, both control schemes are available on excavators and backhoes and can be selected using a switch. The expert still out performed the majority of the subjects when using conventional control. The unintended result of having the expert learn a new form of the conventional control and compare it to the CRC is potentially more interesting than the originally intended comparison. The expert specifically commented that CRC simplified the digging motion by coordinating the boom and arm joints to get linear motion.

In response to the second question, seven of the ten subjects, including the expert, felt they had more accurate control of the bucket with the CRC UI's rocker opposed to the joystick of the conventional UI, as shown in Figure 19(b). In the third question, eight subjects marked that they were at least somewhat confident (a marking of 3 or greater in the survey) with conventional control, while two subjects indicated they were not confident (a marking of 2 or less in the survey) using conventional control. All 10 subjects marked that they were at least somewhat confident using CRC. In total, five of the subjects, including the expert, marked that they were equally confident with both controls, while four subjects felt more confident using CRC, as shown in Figure 19(c).

Subjects in group B answered the same questions about CRC only. Thus, in the absence of a direct comparison to conventional control, the important metric is how each subject perceived their ability to learn the controls. Eight of the ten subjects felt that it was an easy control system to learn as shown in Figure 20. All 10 subjects indicated that they felt they had accurate control of the bucket and were confident using the controls.

5.5 Commercialization Benefits

This technology could potentially affect two markets: the contracting industry and the rental market. At a basic level, a contracting company may be hired to perform a job with an excavator. The job could be a series of trenching tasks to lay pipe. Assuming the contractor owns the equipment, the operating costs are the excavator fuel and operator salary. The price of the job is a reflection of the estimated time the task will take. In this case, the effect of the technology is dependent on the operator's skill level. If the operator is an expert, the data from this experiment is not appropriate for quantifying the cost difference from this technology. Importantly, the operator would not be restricted from using the conventional UI if that was their preference. If the operator is still learning, the results from this experiment are applicable. The combination of the increased task time and fuel efficiency would reduce the estimated job completion time as well as operating costs. The specific reduction in cost is dependent on the required task, but the estimated task time would be reduced by 13.6%, and fuel costs lowered by 10%.



(a) Subjects' selections of which control scheme is easier to learn.

(b) Subjects' selections of the control scheme that they had more accurate control of the bucket.

(c) Subjects' selections of the control scheme they felt more confident using.

Figure 19: The survey responses from subjects in group A.

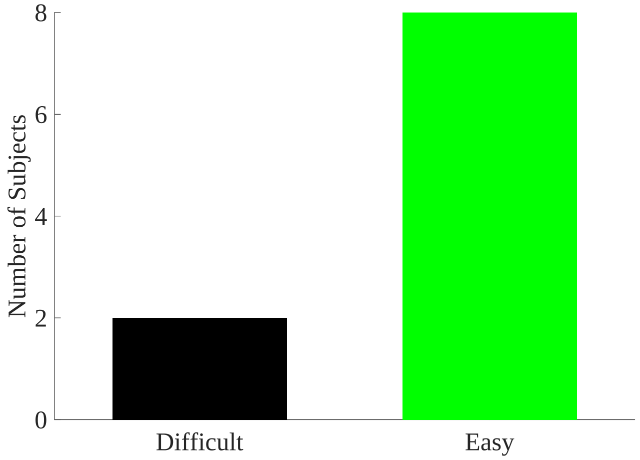


Figure 20: Group B subjects' selections of which control scheme is easier to learn.

The rental market would directly benefit from this technology. When renting an excavator, the consumer often has a single job that they need to complete. Ideally, the controls would be as intuitive and easy to use as possible. In this sense, the CRC UI would make an appropriate UI in rental excavators. Typically, a consumer can rent an excavator for various durations. Assuming the consumer is a novice operator, it would take the operator 13.6% less time to complete the task while using 10% less fuel with the CRC UI. These potential monetary savings for the consumer are not trivial, particularly for larger projects. The reduced task time would result in shorter rental durations and lower fuel consumption. In addition to the shorter time to completion, the lower variability of operators using CRC suggests that operators are in better control of the machine and accidents should be reduced.

The critical point is that with minimal changes to current exavator hardware, the overall operating costs are significantly reduced. The specific reduction in cost is dependent on the job at hand, but using the data from this experiment, the job can be completed 13.6% faster with a 10% reduction in fuel costs using the CRC UI.

5.6 Summary

The results from the experiment indicate that the coordinated rate control user interface retained the performance benefits of the specialized hardware used in [6, 5, 20], but not to the same degree. The previous studies compared the results of operators using conventional control and coordinated control with a specialized joystick. The specialized joysticks are highly intuitive, as the motion of the excavator end effector mimics the motion of the operator's hand. This provides the user with a continuous reference frame, and mitigates the amount of time that an operator needs to plan an action. However, due to the price, complexity, non-ergonomic design, and fatigue issues, the specialized hardware is not practical for the industrial environment. Implementing coordinated control on the standard joysticks allows manufacturers to continue to take advantage of the low cost and ergonomic features of the joysticks, and allow contractors and consumers to reap the performance and cost benefits associated with coordinated control.

Neither the conventional UI or CRC UI limit end effector motion in the workspace. Specifically, the feasible manipulator workspace is identical for each controller. With the same attainable workspace, the theoretical optimal performance is the same for each UI. As a user approaches the optimal performance limit, the space for increased performance is reduced. However, not every operator reaches an optimal level of performance after significant training. Direct claims about how each controller affects the operator's maximum attainable performance would need to be evaluated though longer tests.

The goal of a novel intuitive UI is to both increase the max attainable performance by the human operator, and to accelerate the learning process to reduce the training necessary to approach this performance limit. Over the first 6 trials, a total of 40 minutes of operation including warm ups, operators using CRC removed 13.6% more soil, were 10% more fuel efficient, and spent 41% less time idle. Additionally, in the post-session survey, all 10 subjects from group A marked that they were confident using both controllers. However, the majority of group A, including the expert, found CRC easier to learn, and the subjects were able to attain the same performance level with CRC that was attained through 6 trials with the conventional UI. The superior performance of group B, the rapid learning rate of group A after the switch, and survey responses of group A suggest that operators using CRC would require significantly less time and training to approach the theoretical performance limit when compared to the conventional UI.

CHAPTER VI

TASK IDENTIFICATION OF AN EXCAVATOR

The high level performance metrics presented in Chapter V illustrate some of the general characteristics of each controller. However, the measures do not show when each controller is advantageous with any specificity. Comparing operator performance during subtasks of the trenching cycle would indicate when each controller is beneficial to performance. Specifically, this low level analysis would show when each controller enabled more effective and efficient operator performance. In this context, the term low level analysis refers to the evaluation and comparison of performance per trenching cycle, rather than the comparison of quantities that encapsulate all operation throughout a trial. For this reason, the segmentation of the trenching data into different subtasks is key to the evaluation of these controllers. Data segmentation enables enables analysis of low level metrics that lead to a deeper understanding of the advantages of each control scheme. Chapter VIII will present other applications of the task identification algorithm.

6.1 Excavator Trenching Cycle

Excavators are common machines in various industries, and are used for a variety of operations and processes. Some of the processes include craning or hoisting, hammering, grading, and various operations using different end effector tools. Most often, excavators are used for trenching and bulk excavation. The trenching operation is a cyclic process that is composed of three main subtasks: digging, unloading, and returning. Each of these subtasks can be broken down further into sequences of motion primitives as Enes did in [9], but this is not necessary in this context. The trenching cycle will be the excavator task of study, as not all tasks are suitable for task

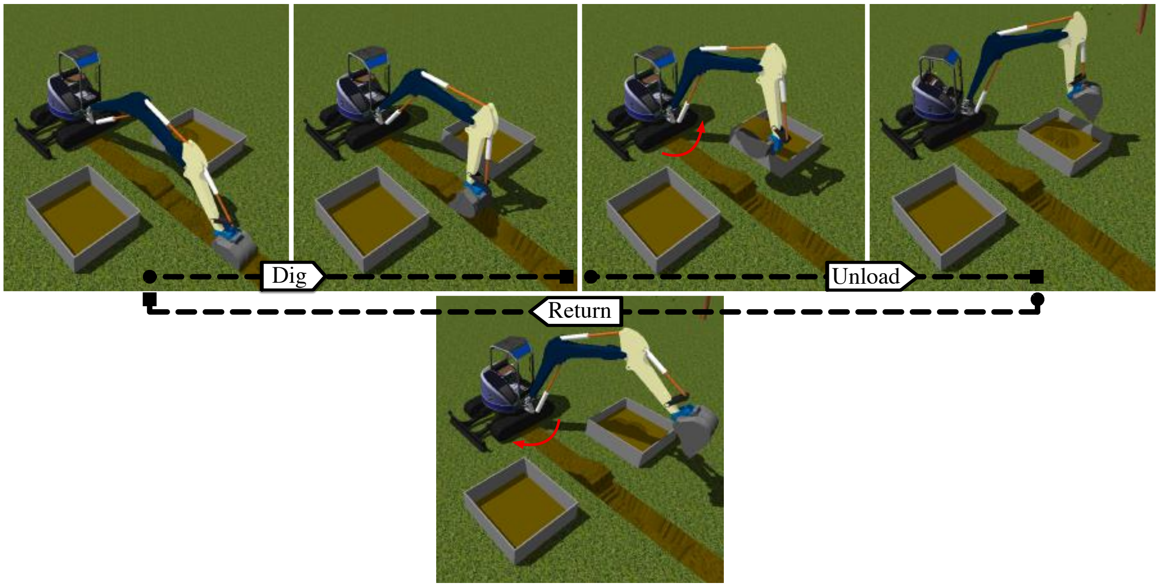


Figure 21: A standard trenching cycle proceeding through the Dig, Unload, and Return phases.

identification nor relevant to the data collected during the experiment.

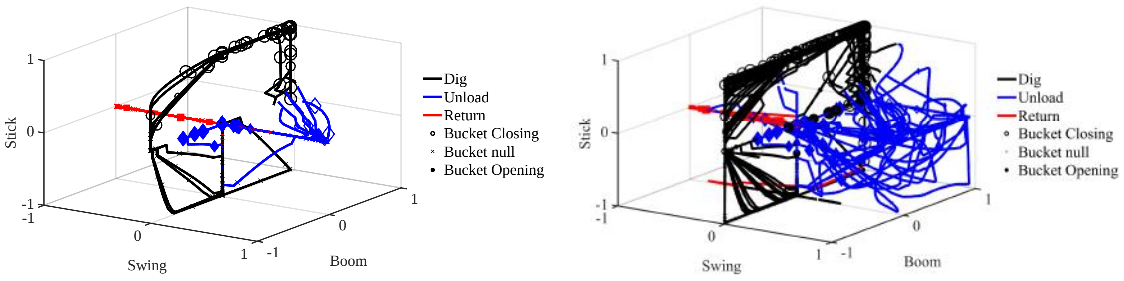
A typical trenching cycle is shown in Figure 21. The cycle can be decomposed into three subtasks: digging, unloading, and returning. If desired, each subtask could be further delineated into other subtasks. Another feasible sequence of subtasks is digging, swinging out, soil dumping, and swinging back; however, it is desirable that the subtasks encapsulate enough operation such that comparing performance during subtasks is meaningful. In the example sequence, there would be relatively little operation during the swinging out, soil dumping, and swinging back subtasks and would be ineffective for performance comparisons. For this reason, the subtasks in Figure 21 were chosen, but the work in this chapter could be applied to any set of desired subtasks.

The dig phase encompasses all motion necessary to remove soil from the trench. The unload phase begins when the operator turns toward the soil pile to dump the soil, and concludes when the operator turns back toward the trench. Often, the operator begins positioning the end effector while turning toward the trench, and simply opens the bucket once at the soil bin. The final and shortest phase, the return phase, is complete when the operator is centered over the trench. Each operator's preferred series of commands is referred to as their digging style, and can cause slight variations in the cycles between operators. The process dynamics also influence the cycle during operation. During the process, the depth of the trench increases and the height of the soil pile increases. These two variations affect the duration of each subtask, as well as the commands used to complete each subtask towards the end of a trenching process. The variations are small enough that they generally do not interrupt the repetitive nature of the process.

6.2 Task Identification Problem Formulation

Defining the trenching process as a composition of subtasks lends itself to the formulation of a task identification problem. In this case, a task identification algorithm has been developed to identify the current subtask being completed using available system information consisting of operator velocity commands and system states. Theoretically, standard rule based programming could be used to develop a suitable mapping from system information to the current phase; however, rule based programming is not practical because the system phase varies in a complex pattern subject to different operator styles and process variations.

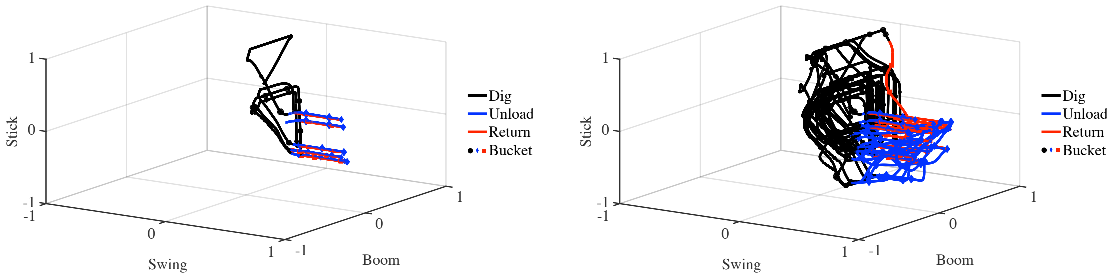
A four dimensional plot showing the phase of motion as a function of the normalized operator commands is shown in Figure 22(a). The value of the bucket command is represented by the scale and fill of the shapes on each line, with the different shapes belonging to the different phases. Null inputs are represented by small a small 'X'. As the process progresses, the command pattern is affected by the dynamic nature of the process; namely the increasing trench depth, and other encountered variations like missing the trench or bumping into the bin. The relationship between the operator commands and phase of motion increases in complexity as the process continues, as



(a) Four trenching cycles.

(b) Eighteen trenching cycles

Figure 22: Motion classifications shown as a function of joystick inputs at the beginning and end of a trenching cycle.



(a) Four trenching cycles.

(b) Eighteen trenching cycles.

Figure 23: Motion classification as a function of the excavator joint positions at the beginning and end of a trenching cycle.

shown by the increased input space used after 18 cycles in Figure 22(b). Operators' command patterns, or digging styles, can vary appreciably; yet, there is a limit to the differences in command patterns. While the specific sequence and magnitudes of the commands can vary, the same excavator functions must be used by all operators to complete the three subtasks. While this is a seemingly trivial statement, it is important to note that there is an underlying consistency between operator digging styles. This underlying consistency is made visible by analyzing the excavator joint positions during operation, where there is less variation in joint position paths among subjects. The joint positions that result from the commands in Figure 22 are shown on a normalized scale in Figure 23, where the range of motion between -1 to 1 is realized by considering the midpoint of the range 0.

The solution to the task identificaiton problem is to view this problem as a pattern recognition or machine learning problem. The visibly separable phases in Figures 22 and 23 suggest the feasibility of finding an underlying pattern. The mapping from system information to the phase of motion can be learned directly from the data using various machine learning strategies. Artificial Neural Network (ANN) classifiers have been used for pattern recognition in various domains [1, 2]; a relevant implementation used an ANN to classify human action from sensed joint angles [11]. An ANN classifier was developed to use the operator inputs and excavator joint positions for subtask identification for this research.

6.3 Artificial Neural Network

An ANN was used to learn the mapping between the system information and the current subtask being completed, hereafter referred to as the phase. Due to the well defined cyclic nature of the trenching process, a supervised classification approach can be used to classify each data point into one of the 3 discrete phases: the dig, unload, or return phase. Unsupervised learning can recognize complex patterns in data, but does not conform to a desired preset structure. To implement the supervised neural network, the following tasks must be accomplished: pre-processing of the data, ANN input selection, ANN hyper parameter selection, training data generation via manual classification, ANN training and testing, and classification post-processing.

6.3.1 Signal Conditioning

To smooth the data, two standard filtering methods were used. As mentioned in Chapter V, 19 of the subjects that participated in the experiment are novice operators with no consequential experience. During the learning process, it is common for operators to issue incorrect commands. An incorrect command, hereafter referred to as a glitch, is defined as an operator input that does not match the intended motion of the operator. Glitches are fundamentally destructive to task estimation because they are not representative of the user's true intent. Typically, the duration of the glitch is very short, as the operator quickly realizes the desired function is not moving and reverts to a null input. Operators are often somewhat hesitant during learning; the hesitancy manifests in glitches of small magnitude on a normalized scale. Thus, it was determined that using a threshold to determine whether to accept or ignore the issued command was an effective and simple solution. The thresholding filter is shown in equation 12, where u_i is one of the four inputs, and α is a tunable parameter that sets the threshold.

$$u_i = \begin{cases} 0 & \text{if } \; u \leq \alpha \\ \\ u & \text{otherwise} \end{cases} \qquad \text{for } i = 1 \text{ to } 4 \tag{12}$$

A moving average finite impulse response filter (FIR) was used to filter both the velocity inputs and position data. The filter mitigated the affects of glitches via smoothing. For completeness, the fifth order moving average used is shown in equation 13 where x is the raw data at discrete step n, and y is the filtered data. Figure 24 shows the result of applying the two filters to a swing command issued during a subject's trial.

$$y(k) = \sum_{n=0}^{5} \frac{1}{5} x(k-n) \tag{13}$$

6.3.2 Manual Data Classification

As previously stated, the trenching process is composed of the dig, unload, and return phases, as shown in Figure 21. To manually classify the data, the transition conditions must be identified. The dig phase is composed of all operation completed to fill the bucket with soil from the trench. With the soil in the bucket, the transition to the unload phase occurs once the operator begins rotating the cab toward the soil pile. More specifically, the transition occurs when the operator uses a positive swing command (turn left). The unload phase includes all operation through the emptying of the bucket, and concludes when the operator begins rotating back towards the

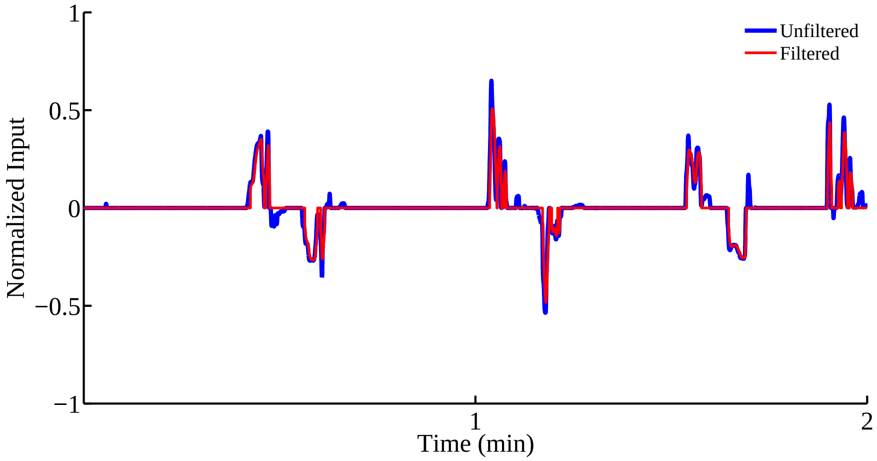


Figure 24: Unfiltered and filtered swing input.

trench, or when a negative swing command (turn right) is issued. The return phase includes all motion while the operator is rotating back toward the trench, and completes when the operator stops the cab rotation and is approximately centered over the trench. The cycle restarts when the operator begins to dig. This transition can be ambiguous, but is critical for acceptable ANN classification. Often, the operators would overshoot the trench while returning and then oscillate around the trench. For return phases that included an oscillatory component, the transition back to the dig phase was selected as the end of oscillation because the operator cannot begin to dig until centered over the trench.

Using the defined cycle phases and transition events, a data set from a 5 minute trial was selected for each controller and manually classified. The data was sampled at 100 Hz during the simulation; therefore, a 5 minute trial consists of 30,000 data points.

6.4 ANN Initialization

To implement the supervised neural network, the network's input, hidden, and output layer sizes must be selected. These hyperparameters define the ANN's static structure and dictate the behavior of the network. The number of inputs is a design choice.

The number of neurons in the hidden layer is slightly more arbitrary. Increasing the size of the hidden layer can increase the complexity of the decision boundary, but also increases training time. The number of output neurons for an ANN classifier is determined by the number of classes, in this case the 3 trenching phases. The network's activation function must also be selected. Each neuron output is the net of the inputs evaluated using the activation function. The hyperbolic tangent sigmoid function was used for this network. The ANN was trained using a standard scaled conjugate backpropogation routine.

6.4.1 Two Input Artificial Neural Network

Initially, the four velocity commands and four joint positions were used as inputs to the neural network, and 50 neurons were used in the hidden layer. The trained neural network had greater than 95% classification accuracy on the training data. Typically, ANN's perform well for the training data. A better measure of the ANN's performance is to evaluate the classification of data not seen during training. It is standard practice to leave a small percentage, in this case 15%, of the training data to be used for testing. In other words, the test set is not used during training, but is used post-training to determine the networks generalization. For this network, the unseen testing data was classified with 95% accuracy as well; however, for this application a better test of the generalization of the network is to classify trenching data from a different subject to test the network's robustness to digging style. The ANN yielded unacceptable results for data sets from other subjects. In other words, the ANN was not robust to operator style. It is a common problem for the network to be overfit to a certain template, in this case a certain subject's digging style. The quantity and quality of the training data dictate the ANN's performance. More training data will yield a more robust network with a more accurate representation of the mapping between the input space to output space.

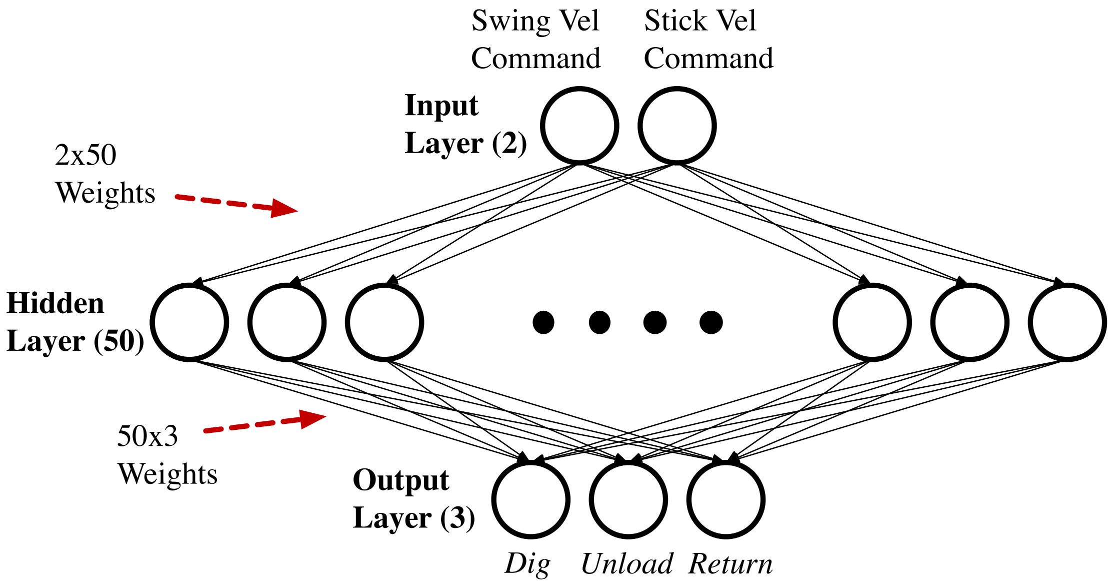
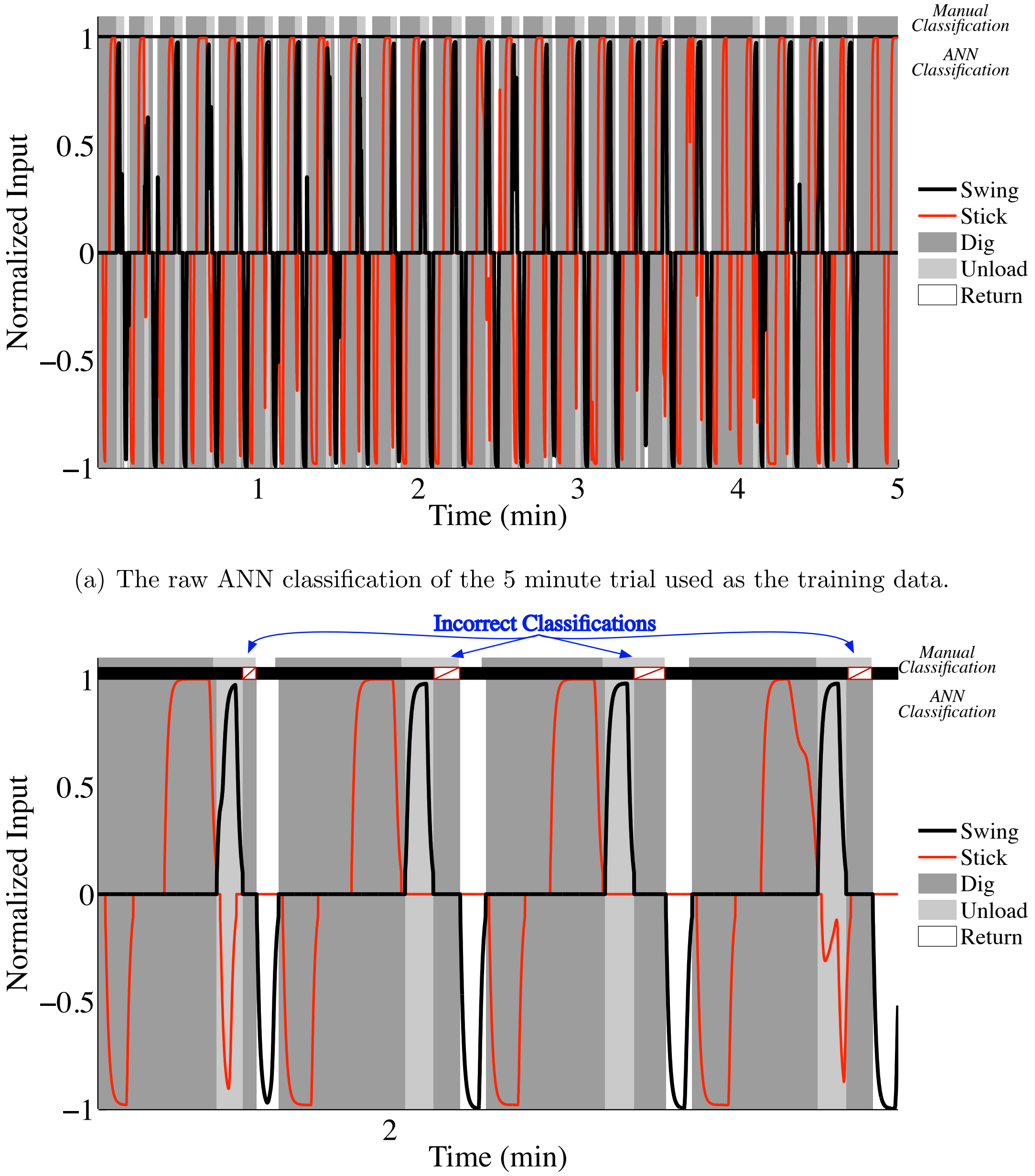


Figure 25: The feedforward 2 input Artificial Neural Network (2ANN).

To circumvent this issue, a simple ANN using only using the swing velocity and stick velocity commands as inputs was developed to generate additional training data. The feedforward 2 input ANN is shown in Figure 25. The simple network is not capable of robust classification because of the information deficient input space, but for clean data sets, the network produces decent classifications due to the consistent use of the swing and stick commands regardless of subject style. Clean data sets are defined to contain few glitches and have generally consistent command patterns. The network was trained using a manually classified trial from a single subject. The trained network classified the training data with a sub par accuracy of 85%, shown in Figure 26. The network described here is used to classify phase of motion for the conventional control scheme. Another ANN was developed to classify the phase of motion for the coordinated rate control scheme using the swing and forward/back velocity commands. The results are near identical, so only the ANN for the conventional UI is discussed in this section.

The network's decision boundary in Figure 27 explains the network's inability to accurately classify the data. Figure 27 illustrates how the trained ANN classifies



(a) The raw ANN classification of the 5 minute trial used as the training data.

(b) Section of the ANN classification.

Figure 26: The raw ANN classification of the data set used for training.

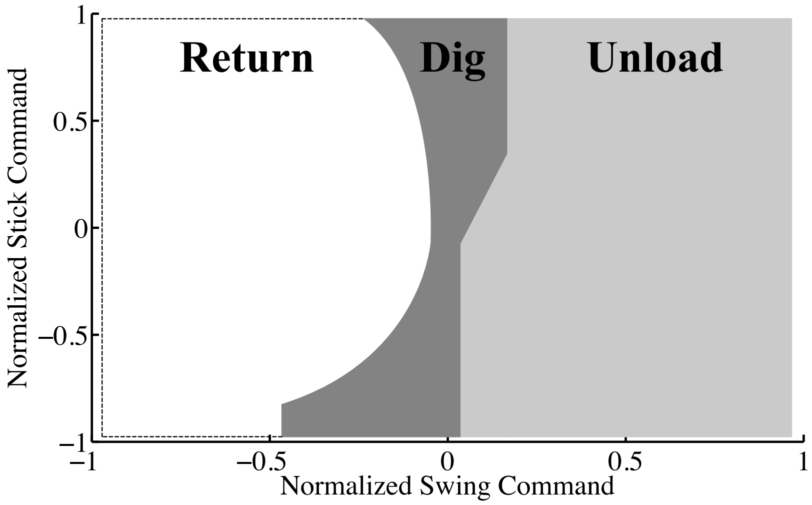


Figure 27: The decision boundary for the 2 input ANN.

the data into the three classes based on the 2D input, where each discrete point corresponds to a feasible operator input. The 2D input space limits the ANN to simple decisions. In this case, the classification is approximately a function of the sign and magnitude of the swing command. For this reason, there is a consistent incorrect classification following each unload phase when the swing command goes to 0 as the subject dumps the soil before the return phase, as pointed out in Figure 26(b). By enforcing ordered transitions of the phases, dig–unload–return, the classification is improved to 97% accurate for the training data. More importantly, the ANN yielded classifications of approximately >90% accuracy for all subjects, and higher for data sets with minimal glitches. An example of an accurately classified unseen data set using the correction is shown in Figure 28.

The two input ANN, hereafter 2ANN, has significant limitations, and could not be considered a robust task identification method. However, it is simply needed for training data generation. The limitations of the network and the enforced correction are exposed in Figure 29. Figure 29(a) shows the poor quality classification, while Figure 29(b) shows that reducing noise with the correction comes at the cost of nearly missing entire phases.

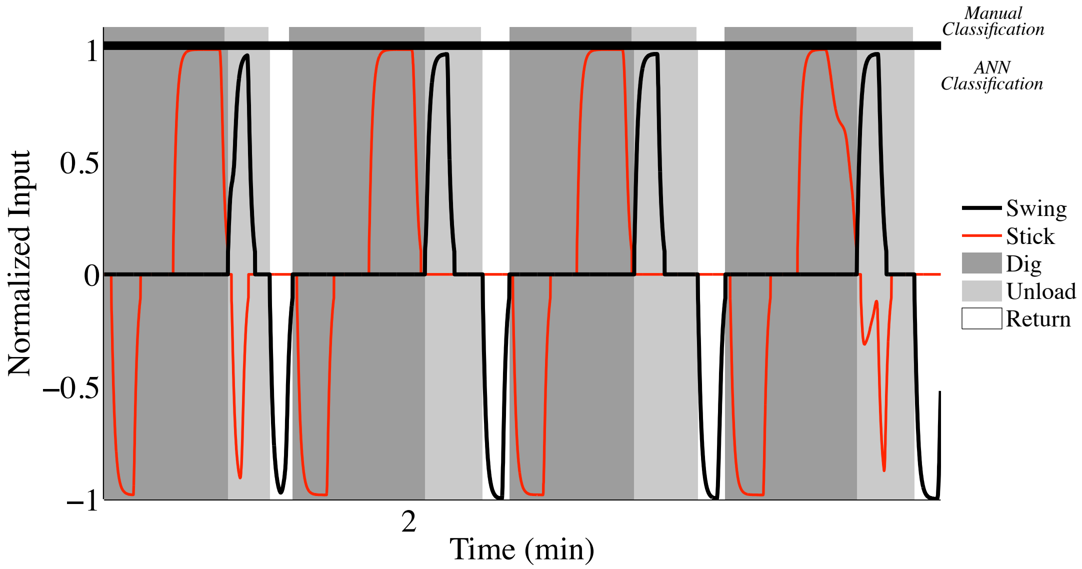
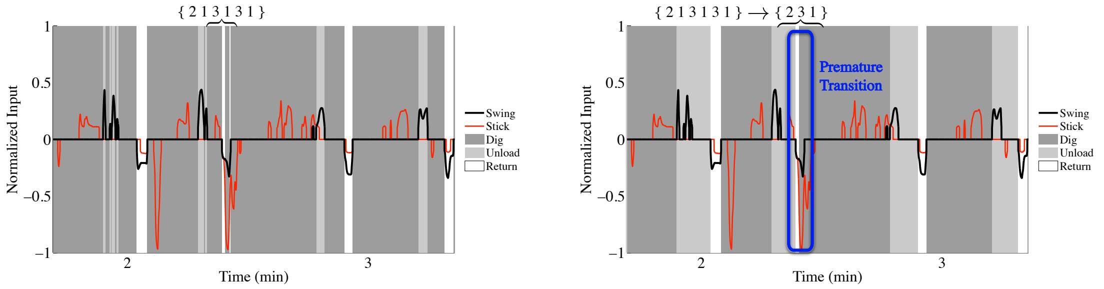


Figure 28: The 2ANN classification of the training data with the correction enforcing ordered transitions.



(a) The raw ANN classification of unseen data.

(b) The ANN classification of unseen data with the correction enforcing ordered transitions.

Figure 29: The ANN classification of a section of unseen data.

The 2ANN can classify arbitrary data sets from other subjects with decent accuracy using an order enforcing correction. Recall that in the experiment, each of the subjects were instructed to dump the soil into the left bin for consistency. It would be impossible for the 2ANN to accurately classify the data into the unload and return phases if the operators could unload in both directions with the 2ANN's learned decision boundary in Figure 27. Additionally, the 2ANN is not robust to oscillation at the end of the return phase; each time the swing velocity command changes sign, the classification changes.

The 2ANN does not produce robust classifications, but does provide clear insight into some of the issues that need to be addressed to develop a robust task identification algorithm for a trenching cycle. The 2ANNs for the conventioanl UI and CRC UI were used to classify all 90 trials from Group A and B. Five classified trials from different subjects in each group were selected to be used as training data for another ANN. Classified data sets from multiple subjects must be used as training data to yield an ANN that is robust to style. Classified data from 5 subjects was used, opposed to all 10, to test the accuracy of the ANN against the 5 unseen subject styles for each controller. In choosing the trials to be used for training data, it was important that data sets with correctly classified glitches be selected to train the ANN to be robust to glitches.

The five trials correspond to 150,000 data points of generated training data consisting of the system information at each time step and the target class. To ensure that the new ANN is robust to unload direction, the swing commands and swing positions were negated for each trial while keeping the same classification to be used as an additional 150,000 data points. The result is 300,000 sets of inputs with their corresponding target outputs in the training data.

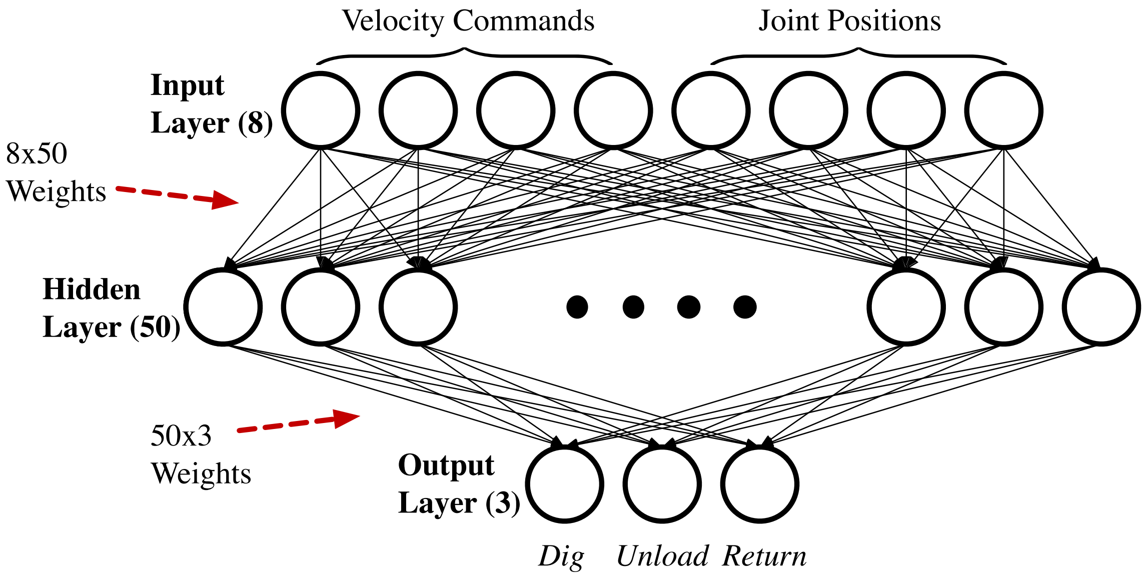
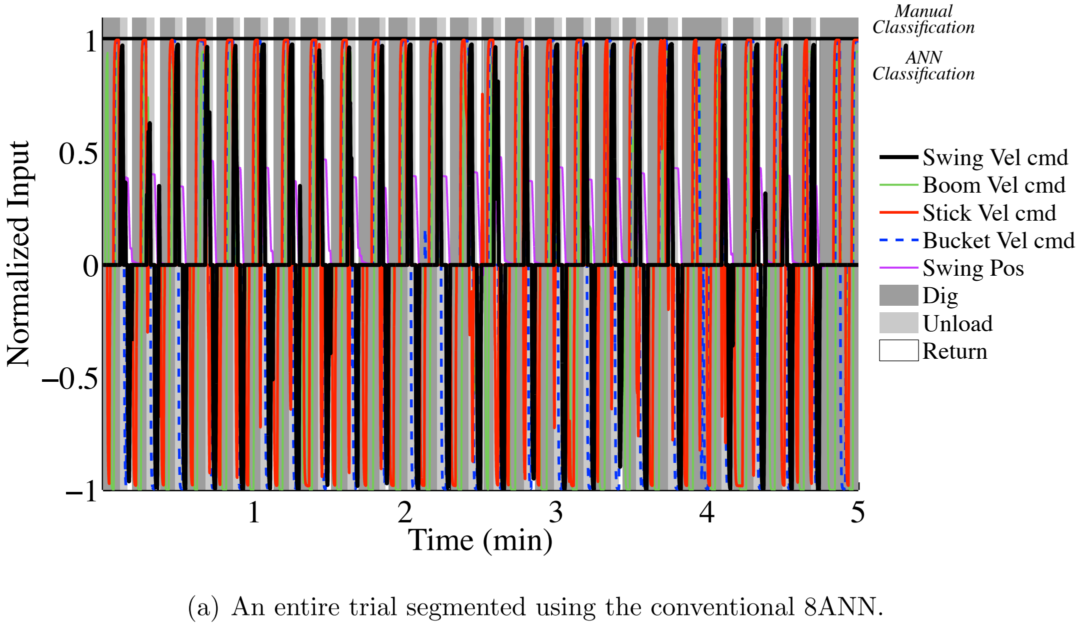


Figure 30: The feedforward 8 input Artificial Neural Network (8ANN).

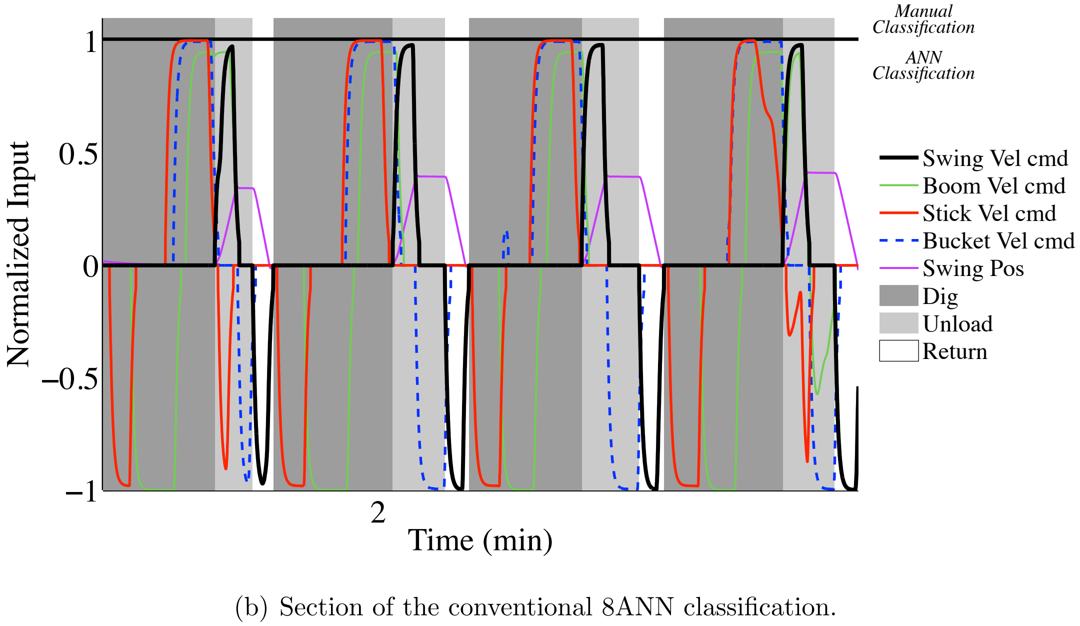
6.4.2 Eight Input Artificial Neural Network

Using the generated training data, two eight input artificial neural networks, hereafter 8ANN, were trained to classify data for the conventional UI and CRC UI. In this context the conventional 8ANN and CRC 8ANN correspond to the neural networks for each control scheme. If referred to as the 8ANN, the information is general and applies to both classifiers. To segment the data for further analysis, it is necessary to have a classifier that has the highest possible accuracy for each data point. For this reason, the four velocity commands and four joint positions at each time step were used directly as the inputs to the 8ANN classifier. Feature extraction is not imperative because each of the signals has a high signal to noise ratio; thus, each data point provides accurate state information. Chapter VIII will present a method for performing real time classification using feature extraction.

Similar to the 2ANN, there are 50 neurons in the hidden layer, and 3 neurons in the output layer for the 3 phases. The static feedforward 8ANN can be seen in Figure 30. The trained conventional 8ANN classified the training data with 99.9% accuracy, and the CRC 8ANN was 99.6% accurate. This extremely high classification accuracy



(a) An entire trial segmented using the conventional 8ANN.



(b) Section of the conventional 8ANN classification.

Figure 31: The raw conventional 8ANN classification of the one of the trials used for training.

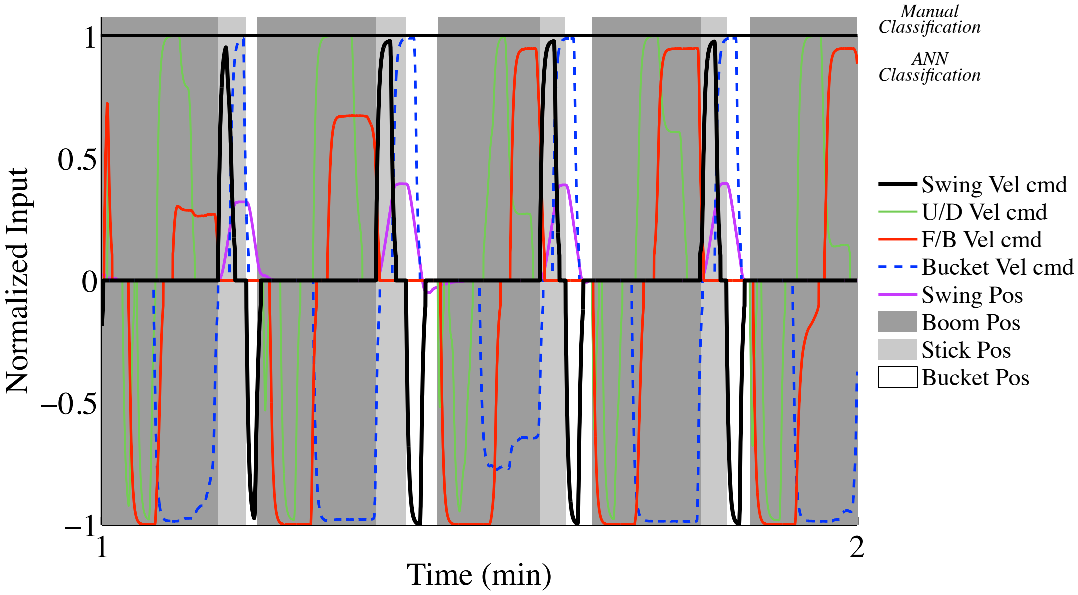


Figure 32: The raw CRC 8ANN classification of one of the trials used for training.

reveals that the there are not large overlapping subspaces within the 8-Dimensional input space that correspond to different classifications. In other words, the different operator digging styles do not yield mutually exclusive classifications.

To test the performance on unseen data, 15% of the training data is used as testing data. The conventional 8ANN classified the unseen testing data with 99.9% accuracy as well. The classification of one of the five trials used in the training data is shown in Figure 31. Figure 31(a) is a qualitative illustration of the primary goal: identify each data point as one of the three phases of motion. Figure 31(b) is a blown up portion of the full trial around the 2 minute mark. Similarly, the CRC UI classified testing data with 99.6% accuracy, and the classification of data used for training the CRC 8ANN is shown in Figure 32.

6.5 Backwards Merge Algorithm

The 8ANN is accurate for training data, and is robust to the glitches and oscillations it was trained to correct. The 8ANN yielded similar results for unseen data, but classifications could be improved by post-processing the 8ANN classifications. Instead of forcing the classes to follow an order like the correction used for the 2ANN, a better method is to use the Backwards Merge Algorithm, similar to what Enes used in his work to filter motion primitives [9].

The BMA is used to correct erroneous classifications that result from glitches or unintended motion. During a trenching cycle, the 8ANN classifications yield a series of identical classes whose aggregate compose a phase. In the formulation presented, the BMA filters each individual classification as well as entire phases. Some notation will first be introduced.

BMA parameters:

Y_k: The k^{th} phase. The previous phase is Y_{k-1}. The phase can only be dig, unload, or return.

$\mathbf{q}(t)$: A 4x1 vector of the absolute positions of each joint.

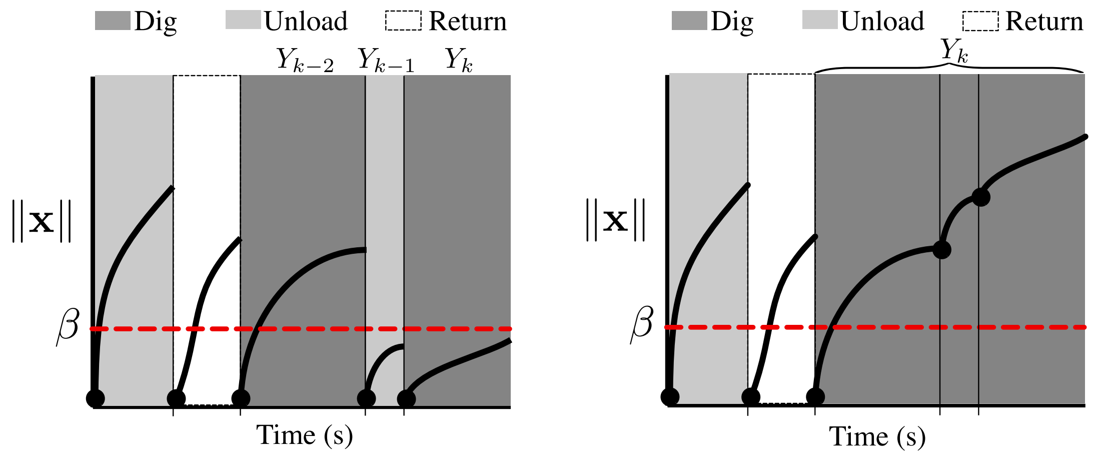
$\mathbf{x}_{k_i}$: A 4x1 vector of the distances traveled by each joint at the beginning of phase k.

$\mathbf{x}_{k_f}$: A 4x1 vector of the distances traveled by each joint at the end of phase k.

β: A tunable parameter to set a threshold.

γ: A tunable parameter to set a threshold.

The BMA merges two sequential phases if the distance traveled by all joints through the duration of the second phase is negligibly small. A phase that only contains small motions is presumed to be erroneous and the result of glitches. For instance, assume that the 8ANN classification has changed from Y_{k-1} to Y_k, as shown in Figure 33(a). The 2-norm of the distance traveled by each joint through the duration of phase Y_{k-1} is computed and is compared to a threshold to determine if it is a valid phase. If the norm is less than the threshold, $\left\|\mathbf{x}_{k-1_f}\right\| < \beta$, the phase is not valid. If the phase was valid, $\mathbf{x}_{k_i}$ is set to $\mathbf{0}$. Otherwise, the previous phase, Y_{k-1}, is merged backwards and re-coded to equal the it's preceding phase, $Y_{k-1} := Y_{k-2}$,



(a) The raw ANN classification of unseen data. The current phase is Y_k. Y_{k-1} is deemed an incorrect classification due to small motion.

(b) The filtered classification using BMA. Phase Y_{k-1} is merged with Y_{k-2}, and because $Y_{k-1} = Y_k$, merged with it as well. $\mathbf{x}_{k_i}$ is shifted to equal $\mathbf{x}_{k-1_f} + \mathbf{x}_{k-2_f}$

Figure 33: The filtered classification using the backwards merge algorithm

shown in Figure 33(b). If the current phase is the same as the corrected previous phase, $Y_{k-1} = Y_k$, then the origin of the current phase is shifted, $\mathbf{x}_{k_i} = \mathbf{x}_{k-1_f} + \mathbf{x}_{k-2_f}$, to ensure that the full motion of the corrected phase is accounted for in future iterations of the algorithm. This logic is shown in Algorithm 1. In practice, it is useful to have one tunable parameter for each class because the amount of motion required to complete each subtask varies. Specifically, the dig phase requires much more gross motion than the return phase. Consequently, more accurate classifications can be acquired by using a different tuned parameter dependent on the value of Y_{k-1}.

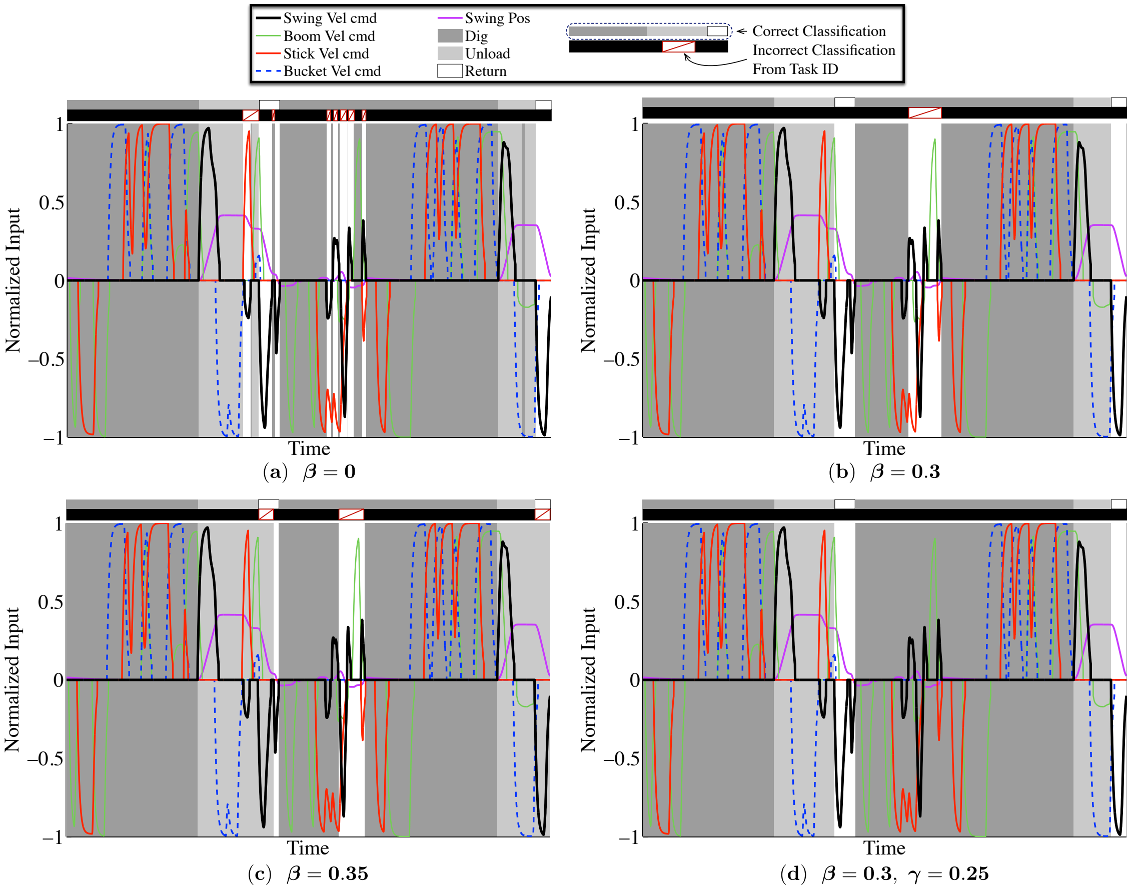
In this general form, the BMA improves the classification. Figure 34 shows two cycles of a 5 minute trial for different values of the threshold, β, in the BMA. Figure 34 has the correct manual classification above the plots, with red rectangles indicating the 8ANN and BMA misclassifications. When returning to the trench, the operator issues discontinuous swing command pulses. At the beginning of the second dig phase, the operator begins digging, indicated by the zeroed swing position and the nonzero boom and stick commands. Then, the user issues a glitch, accidentally using the joystick for swing. The operator is then forced to correct this mistake, and oscillates

```
if *the phase has changed* then
    if $\|\mathbf{x}_{k-1}\| < \beta$ then
        $Y_{k-1} := Y_{k-2}$ ;
        if $Y_{k-1} = Y_k$ then
            $Y_k := Y_{k-1}$ ;
            $\mathbf{x}_{k_i} = \mathbf{x}_{k-1_f} + \mathbf{x}_{k-2_f}$ ;
        else
            $\mathbf{x}_{k_i} = \mathbf{0}$ ;
        end
    else
        $\mathbf{x}_{k_i} = \mathbf{0}$ ;
    end
else
    *the phase has not changed - do nothing*
end
Result: Filtered Classification, $Y_{k-1}$
```

Algorithm 1: Backwards Merge Algorithm

about the center of the trench.

Figure 34(a) shows the raw output of the 8ANN without the BMA. The discontinuous swing commands cause the 8ANN to incorrectly switch between phases, as indicated by the red rectangles. The BMA with a low value for β improves this classification in Figure 34(b). The BMA successfully filters the many switches and correctly classifies the first return phase, but the low value of β causes the glitch in the second dig phase to be incorrectly classified. In this case, $\|\mathbf{x}_{k-1}\| > \beta$, and thus the glitch is considered valid. To eliminate this issue, β is increased, shown in Figure 34(c). However, simply increasing β is not an appropriate solution because the joints travel further during the oscillation than the required motion of the return phase; this results in missing the return phase, and still incorrectly classifying the glitch. β could be increased further to correct the glitch, but this would be detrimental to the task estimation because it causes multiple return phases to be ignored. A solution is to modify the BMA when merging the return phase. The modification changes the definition of a valid return phase. A return phase is deemed valid if



(a) $\beta = 0$ (b) $\beta = 0.3$ (c) $\beta = 0.35$ (d) $\beta = 0.3,\ \gamma = 0.25$

Figure 34: The classification of the task identification algorithm (the 8ANN and BMA) for different values of β. This extreme example shows it is necessary to add a second tuned parameter, γ, when merging the return phase to ensure correct classification.

$\|\mathbf{x}_{k-1}\| < \beta$ **OR** $(\mathbf{x}_{k-1})_{swing} < \gamma$, where $(\mathbf{x}_{k-1})_{swing}$ is the element of the vector corresponding to the swing joint and γ is another tuned threshold. This implementation is shown in Figure 34(d), and is successful in filtering out the discontinuous return commands as well as the oscillating glitch in the dig phase.

6.6 Task Identification Algorithm

The task identification algorithm is the combination of the 8ANN and and the BMA. The 8ANN alone classified the training data with very high accuracy. As previously mentioned, a better measure of the network's generalization, and the task identification algorithm as a whole, is to test the classification accuracy for unseen digging

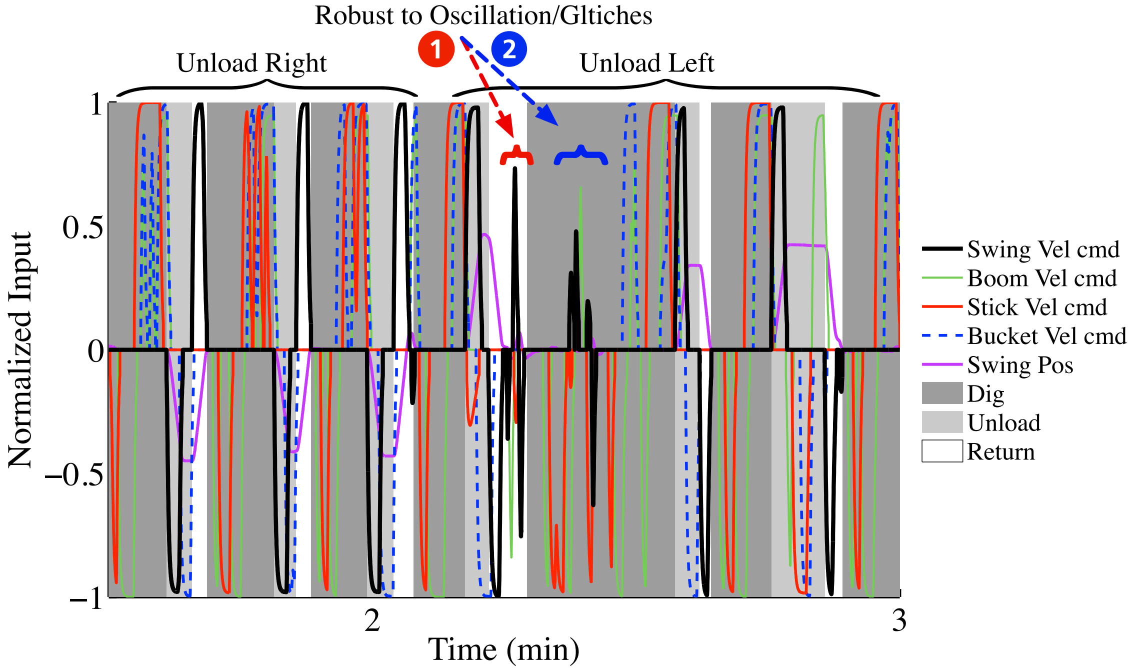


Figure 35: The classification of a portion of an unseen 5 minute conventional UI trial. The correct classification indicates that the task identification algorithm is robust to style, and the labels show the ANN is robust to unload direction and glitches.

styles. Figure 35 shows the classification of a section of a conventional trial that came from a subject who did not contribute to the training data. In other words, the 8ANN had never been exposed to this specific user's digging style. In this trial, the operator is initially unloading soil to the right, then switches to the left. These motions are correctly classified. In Figure 35, the first arrow shows that the oscillations at the end of the return phase are correctly classified. The second arrow shows that the operator mistakenly used the swing command after already starting to dig. As pointed out in the previous section, the major contribution of the BMA is to filter oscillations and other glitches. This motion is correctly identified as still being in the dig phase. Similarly, Figure 36 shows the correct classification of an unseen CRC trial with glitches at the end of two return phases.

Ensuring that the task identification algorithm was robust to operator digging

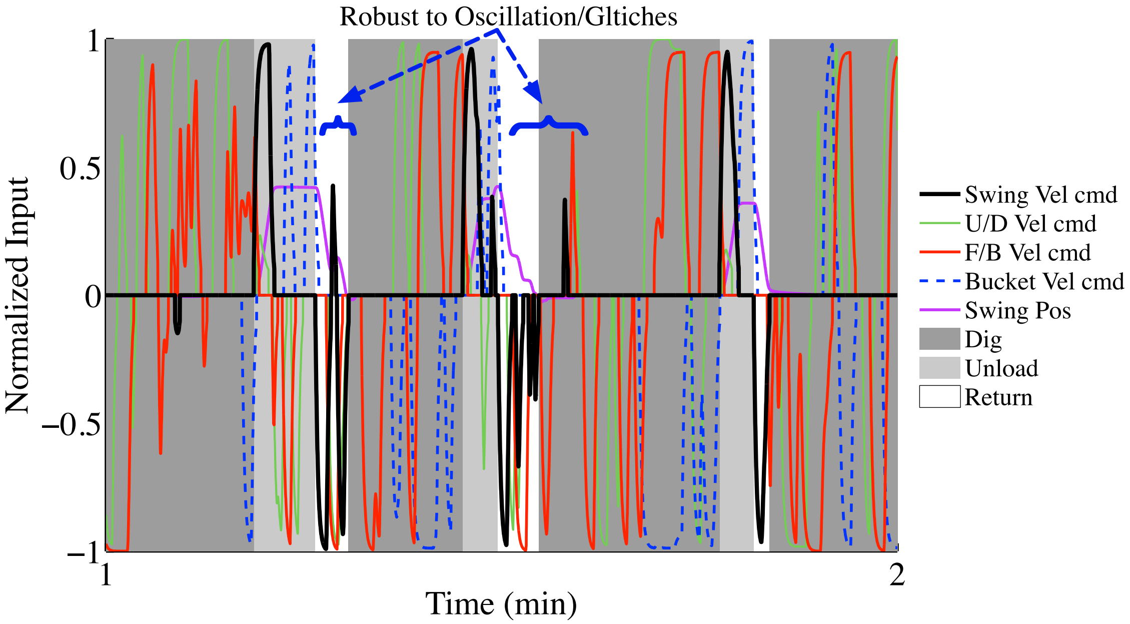
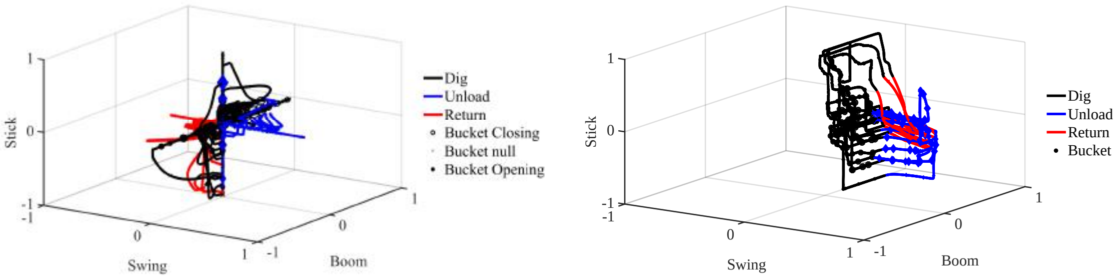


Figure 36: The classification of a portion of an unseen 5 minute CRC UI trial. The correct classification indicates that the task identification algorithm is robust to style, and the labels show the ANN is robust to glitches.

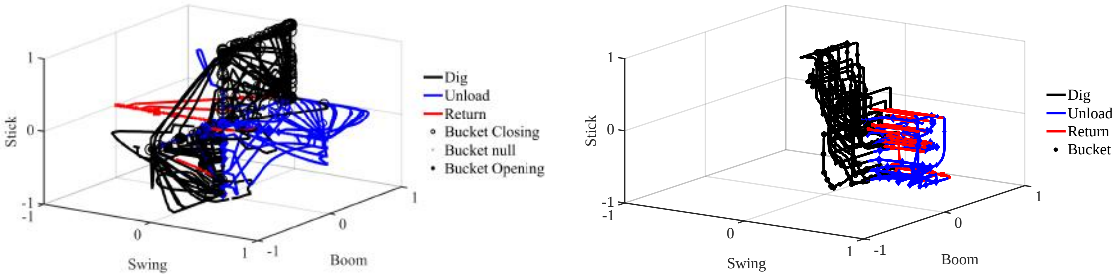
style was of critical importance. Figure 37 shows the joystick commands and excavator joint position paths during a trial in a 4 dimensional plot. The classifications are indicated by the different colored lines, and the value of the bucket command is represented by the scale of the shapes on each line. Figure 37(a) gives a qualitative representation of the operator's digging style. This particular operator uses lower magnitude velocity commands, resulting in slower motion. For comparison, the commands and joint positions from a different subject are shown in Figure 38. The difference in the command paths between Figures 37(a) and 38(a) illustrate the difference in digging style. On the other hand, Figures 37(b) and 38(b) show that the resulting joint angles are similar to each other, and are nearly independent of the different digging styles. While the 8 dimensional decision boundary is not easily visualized, the similarity in the joint angles between two operators with different styles demonstrates how a decision boundary was learned from the 8 inputs.



(a) Joystick commands.

(b) Excavator joint positions.

Figure 37: Motion classification as a function of joystick commands and excavator joint positions. The shape of the joystick commands illustrate a digging style.



(a) Joystick commands.

(b) Excavator joint positions.

Figure 38: Motion classification as a function of joystick commands and excavator joint positions for a specific subject. The joystick commands illustrate a different digging style from previous figure.

6.7 Summary

While the accuracy of task identification is readily apparent for the manually classified training data (Conventional-99.9% and CRC-99.6%), it is more difficult to develop a quantitative accuracy for data from unseen styles. The 90 trials from each group were classified, and the results were inspected. The vast majority of the classifications resembled the accuracy of Figure 35. The few data sets that did not have near perfect classifications were from noisy first trials. These trials typically had an abundance of glitches. In these cases, the resulting classifications looked similar to Figure 34(b), where glitches were misclassified due to large motion by all of the joints. The accuracy of task identification increases as the operator's commands become more aligned with their true intention. Further work on the development of a robust task identification algorithm using a supervised classifier would benefit from exploring the types of glitches to include in the training data.

The developed task identification algorithm classifies the gross motion of the excavator into three subtasks of a trenching cycle using system information at each time step. The initial classification is performed using a trained ANN, and the BMA is used to correct previous misclassifications due to operator glitches. The BMA becomes less necessary as the operator's skill level increases. This task identification algorithm can be used to segment acquired trenching data into the three phases. To implement this algorithm real time, the intuition developed by the BMA can be used to identify a classification as valid after the norm of the joints has surpassed a threshold value. Real time implementation will be discussed in Chapter VIII.

CHAPTER VII

LOW LEVEL ANALYSIS

Chapter V discussed how operators performed with each of the controllers in relation to three high level performance metrics: time efficiency, fuel efficiency, and idle time. Specifically, it showed that operators removed 13% more soil/time, 10% more soil/fuel, and were idle for 41% less time using CRC. The goal of this analysis is to show *how* operators performed more effectively and efficiently; this could be a result of operating at a much faster rate, or using the machine more effectively. Using the task identification algorithm developed in Chapter VI, all of the trenching data from the experiment can be segmented into the three different phases for low level analysis. As defined in Chapter VI, the term low level analysis refers to evaluation and comparison of performance per trenching cycle, opposed to high level analysis that compares performance per trial. Evaluating the operators' performance during each subtask will enable inter-subtask comparisons that yield quantitative advantages of each UI.

7.1 Data Segmentation

The task identification algorithm was used to segment the 90 trials from each group. Only the data that contributed to complete cycles was considered for further analysis. In other words, if the operator completed an unload phase and was returning to the trench when the 5 minutes expired, the data from the incomplete cycle was not considered. The metrics of interest are compared as the per cycle means, thus the total quantity of a measurement for a trial is not pertinent and does not have an effect on the analysis.

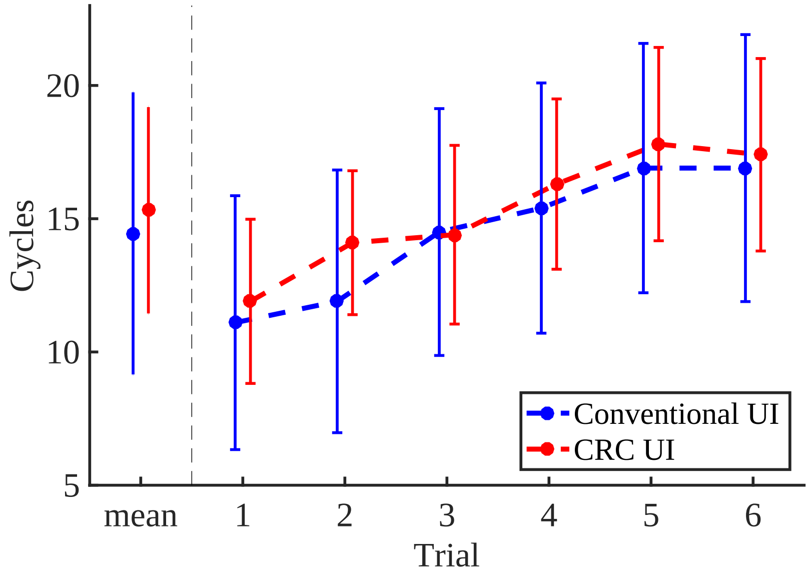


Figure 39: The mean number of cycles completed each trial using each controller.

7.2 Results

7.2.1 Time Efficiency

To determine the specific cause of the 13% greater time efficiency displayed by CRC users, both the speed and the effectiveness of the operators can be quantified. To evaluate the operator's speed, the average number of cycles completed using each controller is compared, shown in Figure 39. Operators using CRC complete approximately 0.8 more cycles per trial on average than the conventional control users. Subjects using CRC operate more quickly, but the additional 0.8 cycles completed each trial does not account for the 13% more soil/min.

The amount of soil that operators remove each cycle is a measure of how effective they are at digging. Figure 40 shows the amount of soil removed per cycle for the first six trials of the experiment. On average, CRC users remove 10% more soil each time they dig than the conventional control users. Similar to the high level time efficiency in Chapter V, there is much less variability among the CRC users; the average standard deviation of the CRC users is 56% less than the conventional

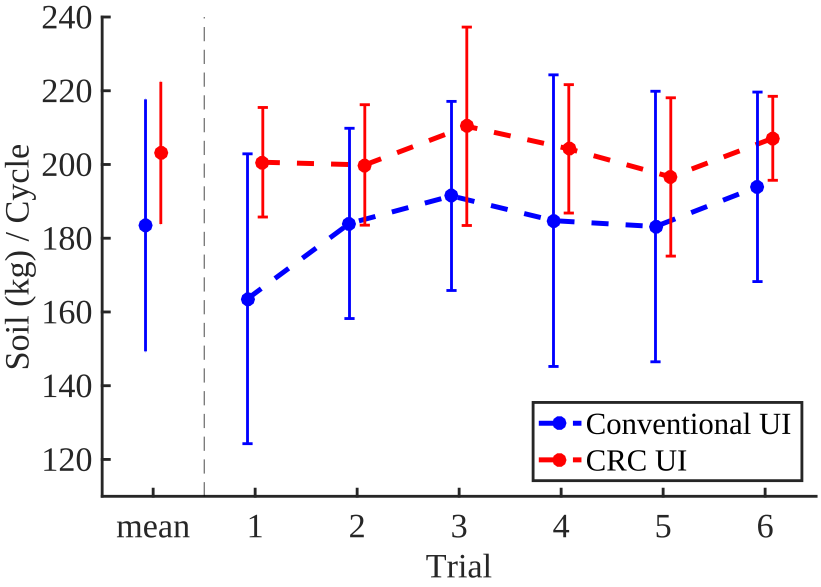
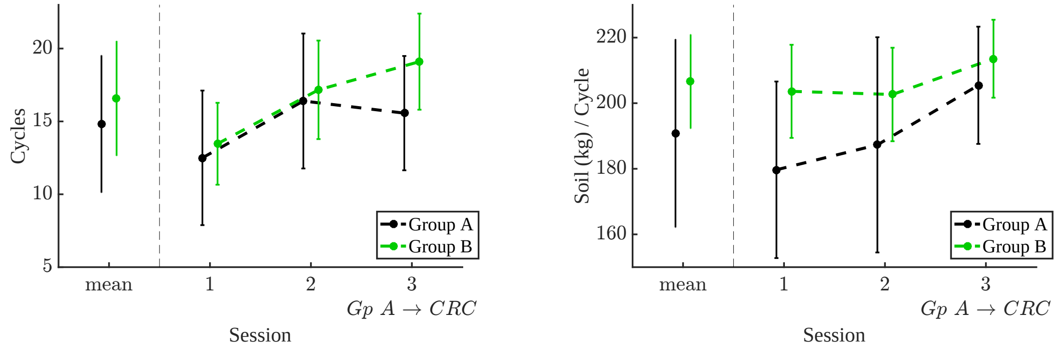


Figure 40: The mean amount of soil removed per cycle each trial.

users. Running an ANOVA on these data sets yields a p value of 0.069, thus the null hypothesis cannot be rejected for a rejection level of 0.05. Operators using CRC operate slightly more quickly and are much more effective at removing soil from the trench.

This can be further evaluated by using the same metrics to analyze the performance of group A after they switch from conventional control to CRC. The mean number of cycles per session is shown in Figure 41(a). The mean amount of soil removed per cycle for each session is shown in Figure 41(b). Each session is composed of 3 trials. Group A switches from the conventional UI to the CRC UI for session 3. After switching to CRC, the average number of cycles completed is reduced by 5%, shown in Figure 41(a). The slight decrease in speed is expected when switching UIs. The decrease in speed is not huge considering that the subjects complete 7.7% more cycles than the average number of cycles between sessions 1 and 2. After the switch, group A subjects remove 10% more soil each cycle when compared to their second session with conventional control. Additionally, the mean standard deviation is reduced by 46% from session 2 to session 3. Thus, after switching to CRC, the



(a) The mean number of cycles completed each session.

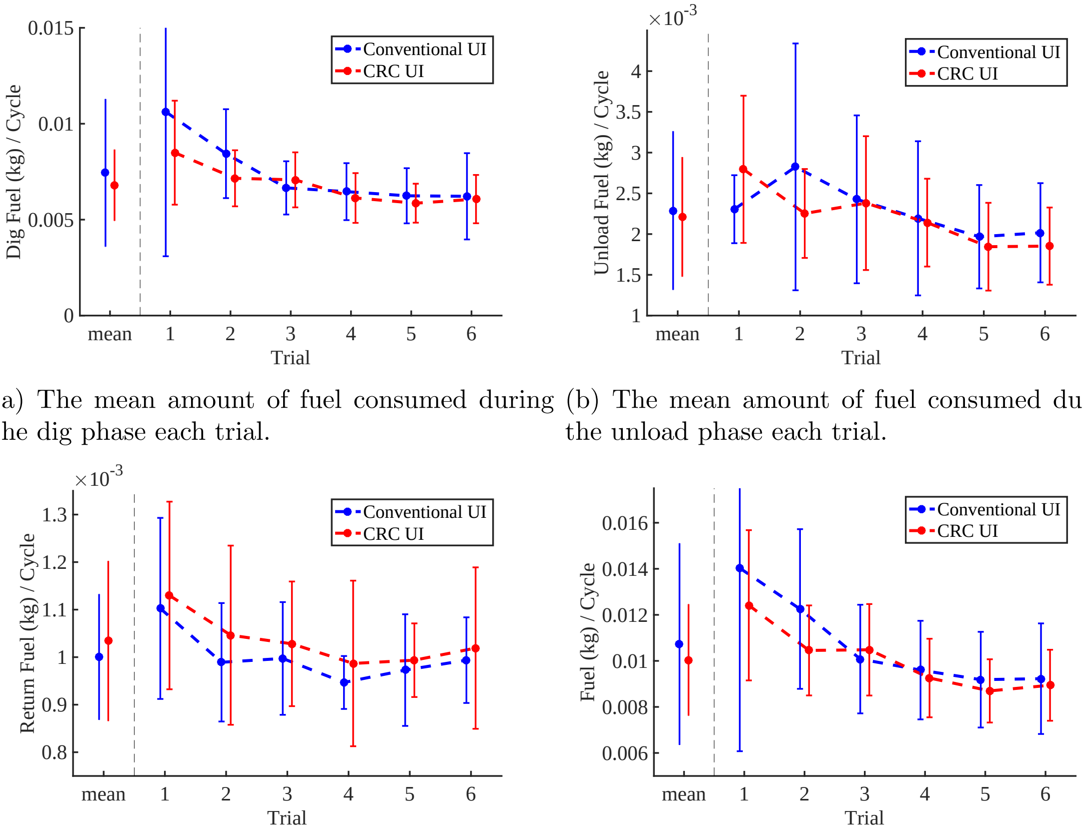
(b) The mean amount of soil removed per cycle each session.

Figure 41: Low level performance metrics for each group by session. Group A switches from conventional control to CRC for session 3.

operators completed a fewer number of cycles, but remove more soil each cycle with less variation among the group. This suggests the intuitiveness of the CRC UI, in particular the use of the rocker to control the bucket. These results also coincide with the results from the post session survey, shown previously in Chapter V in Figure 19(b). In the survey, 7 of the 10 subjects in group A felt they had better control of the bucket using CRC. The speed at which the operators attain the increased performance is further reinforced by the the survey, where 8 of the 10 subjects marked that they found CRC easier to learn, shown in Figure 19(a).

7.2.2 Fuel Efficiency

To gain a better understanding of the operators fuel consumption with each controller, fuel efficiency can be measured for each phase per cycle. The operators' fuel consumption during the three phases is shown in Figure 42. On average, operators using CRC consume 9% less fuel during the dig phase, shown in Figure 42(a). Initially, the CRC users are using much less fuel, but as the operators gain experience the means of each controller converge to be approximately the same. Operators using CRC also consume less fuel during the unload phase by a small margin of 3%, shown in Figure 42(b). During the return phase, CRC users consume 3% more fuel. In total,



(a) The mean amount of fuel consumed during the dig phase each trial.

(b) The mean amount of fuel consumed during the unload phase each trial.

(c) The mean amount of fuel consumed during the return phase each trial.

(d) The mean fuel consumption during each of the three phases for the first 6 trials.

Figure 42: The mean fuel consumption each cycle for the first 6 trials.

operators use 6.6% less fuel each cycle using CRC, as shown in Figure 42(d).

After switching from conventional control to CRC, subjects in group A consume 7% more fuel in session 2, or a 6% reduction compared to the average consumption between sessions 1 and 2, as shown in Figure 43. Group B users continue to improve their fuel efficiency through the third session.

7.3 Summary

The goal of this analysis was to reveal the specific advantages of each controller. After segmenting the data, the quantities of interest were evaluated to yield per cycle performance metrics. As stated in Chapter V, the CRC UI was developed to maximize operator performance through an intuitive UI. Over the first 6 trials,

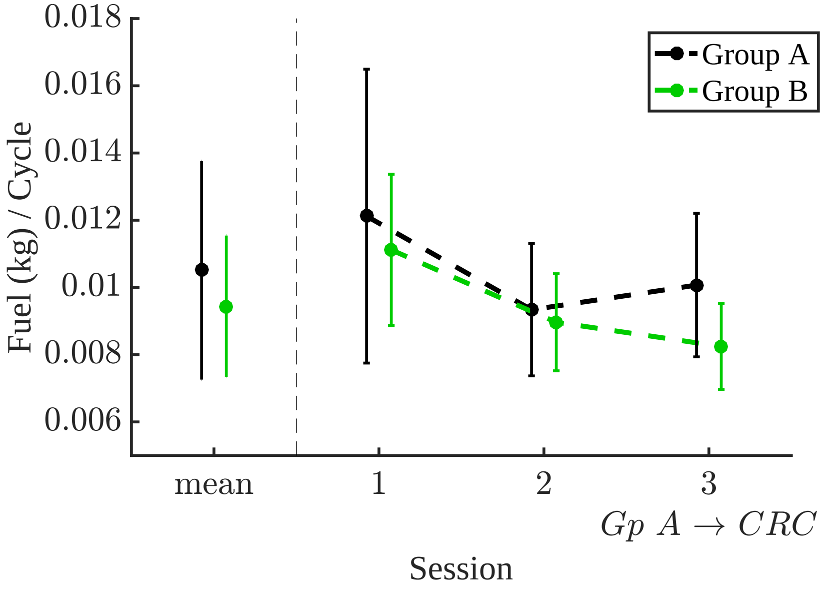


Figure 43: The mean amount of fuel consumed per cycle each session.

operators using CRC removed 10% more soil each cycle, and consumed 9% less fuel on average while digging. Notably, the operators using CRC displayed much less variability compared to the conventional control users with a 56% smaller standard deviation. When subjects in group A switched to CRC they removed 10% more soil than their second session with conventional control, and the standard deviation among the group decreased by 46%.

These low level performance metrics align with the previously discussed high level metrics and survey results, and indicate that CRC is a more intuitive interface than the conventional control method. The increased operator capabilities and lower variability suggests that the operator is fundamentally in better control of the machine when using CRC.

CHAPTER VIII

REAL TIME TASK IDENTIFICATION

Chapter VI provided a thorough discussion of the development of an offline task identification algorithm. The algorithm was then used to segment all of the trenching data from the experiment to further analyze the performance of the operators with each control scheme in Chapter VII. Another application of a task identification algorithm is for real time sub task identification. In practice, the system's capabilities could be optimized for the current subtask being completed. In this sense, real time task identification could be used to create an adaptive system.

This chapter will first present a method for performing real time classification. Some advantages and disadvantages of the method will be discussed, and results using the real time classficiation method will be presented. Then, the task identification algorithm in Chapter VI is offered as a way of performing online learning. Finally, the applications of a real time task identification system will be discussed.

8.1 Sliding Window Method

The task identification algorithm in Chapter VI used an 8 input Artificial Neural Network classifier and the Backwards Merge Algorithm for offline task identification. The 8ANN yielded accurate classifications, and the BMA was used to correct previous misclassifications. The combination of the 8ANN and BMA were used to achieve highly accurate classifications for each data point.

In this formulation, only the 8ANN could be used for real time classification. Correcting previous transitions using the BMA is not applicable real time; however, the intuition from the BMA could be used to make decisions about weather a phase transition is valid. The phase transition would only be deemed valid if the norm of the distance traveled by the joints since the start of the phase had passed a threshold value. This implementation would give decent results, but relying solely on the 8 inputs at each time step is potentially problematic due to the uncertainty in the types of glitches a novice operator would command. A specific example of the type of glitch that the 8ANN misclassified is illustrated by Figure 34(a) in Chapter VI. Thus, it is desirable to use a classification method that accounts for a short history of the commands and positions rather than data from a single time step to produce accurate classifications without a filter for real time task identification.

A better method for real time classification that accounts for more than one data point is the sliding window approach. The sliding window method has been used to solve various machine learning problems such as human activity recognition using acceleromoter data [3, 14, 18]. In this implementation, the raw data is evaluated as a series of discrete windows. Each window's width is a time duration, τ. The window step size is equal to the width of the window. The raw data is then divided into τ-second windows, and features are extracted from each window. Several trade-offs must be considered for selecting the window width. Smaller values of τ enable accurate transition identification. As $\tau \rightarrow t_{step}$, where t_{step} is the rate at which data is sampled, the sliding window degenerates to using each individual data point like the 8ANN. Similar to the 8ANN, short windows yield classifications that are sensitive to glitches. Larger values of τ tend to yield more robust classifications. However, as τ becomes greater, the precision of the transition identifications decreases. While large delays caused by large values of τ are detrimental to the accuracy of the classification, there are moderate values of τ that yield accurate classifications with acceptable delays in transition identification. The small delay is the cost of validating the phase transition.

8.1.1 Feature Extraction

Using the sliding window enables application of various machine learning techniques. From each window, features can be extracted for each signal and used as new inputs to train a supervised classifier. It was experimentally determined that a window size of $\tau = 0.5$ seconds, or 50 time steps, yielded the most robust classifications. A window width of 0.5 seconds provides sufficient time to capture data around short glitches while still maintaining accurate transition recognition.

Segmenting the training data, composed of 300,000 data points at 0.01 second intervals, into 0.5 second windows results in 6000 discrete windows. Each window contains 50 data points for the eight signals, the four velocity commands and four joint positions. Five features were used for each of the eight signals, amounting to a total of 40 features. The features used for each signal within the segmented windows are the mean, standard deviation, first principal component, the minimum value, and the maximum value.

8.1.2 Sliding Window Classification Results

The 40 features were used to train a supervised classifier. The first classifier trained was an ANN. The training process is identical to Chapter VI, except that there are 40 inputs for each 0.5 second window. The trained feature based conventional UI ANN classified the training data with 99.2% accuracy, as shown in the confusion matrix in Table 7. The values on the diagonal of Table 7 refer to the number of windows that were correctly classified for each class, and the off diagonal values indicate the number of misclassifications for each class. The trained feature based CRC UI ANN classified the training data with 99.3% accuracy, as shown in Table 8

Figure 44 shows the performance of the trained ANN for unseen data with glitches for both the conventional UI and CRC UI. Figure 44(a) shows robustness to an oscillation at about 4.5 minutes for the conventional UI. The network shows robustness to several glitches in Figure 44(b) for the CRC UI. Thus, robust classifications of unseen data has been obtained without the BMA. This network would be practical for real time task identification.

For comparison, a support vector machine (SVM), another type of classifier, was trained so results could be compared. A trained cubic SVM classified the training data with 99.5% accuracy for the conventional UI, as shown in Table 9. The trained SVM for the CRC UI classified the training data with 98.5% accuracy, as shown in Table 10. The accuracy of the SVMs show that there are certainly other supervised classifiers that should be tested for real time implementation. In addition, the window width, τ, and the features used as inputs to the classifier could be optimized.

8.2 Online Learning

The classifiers presented in the previous section are capable of robust real time task identification. However, due to the unpredictability of novice operators, the task identification process could be further improved if the classifier could learn each new operator's digging style. A supervised classifier could be implemented as a trained network, but updated for each new operator. To accomplish this, the task identification algorithm in Chapter VI (8ANN and BMA) could be used to classify the data while the operator begins the trenching task. As discussed at the beginning of this chapter, the task identification algorithm in Chapter VI provides highly accurate classifications for each data point by filtering misclassifications with the BMA. The

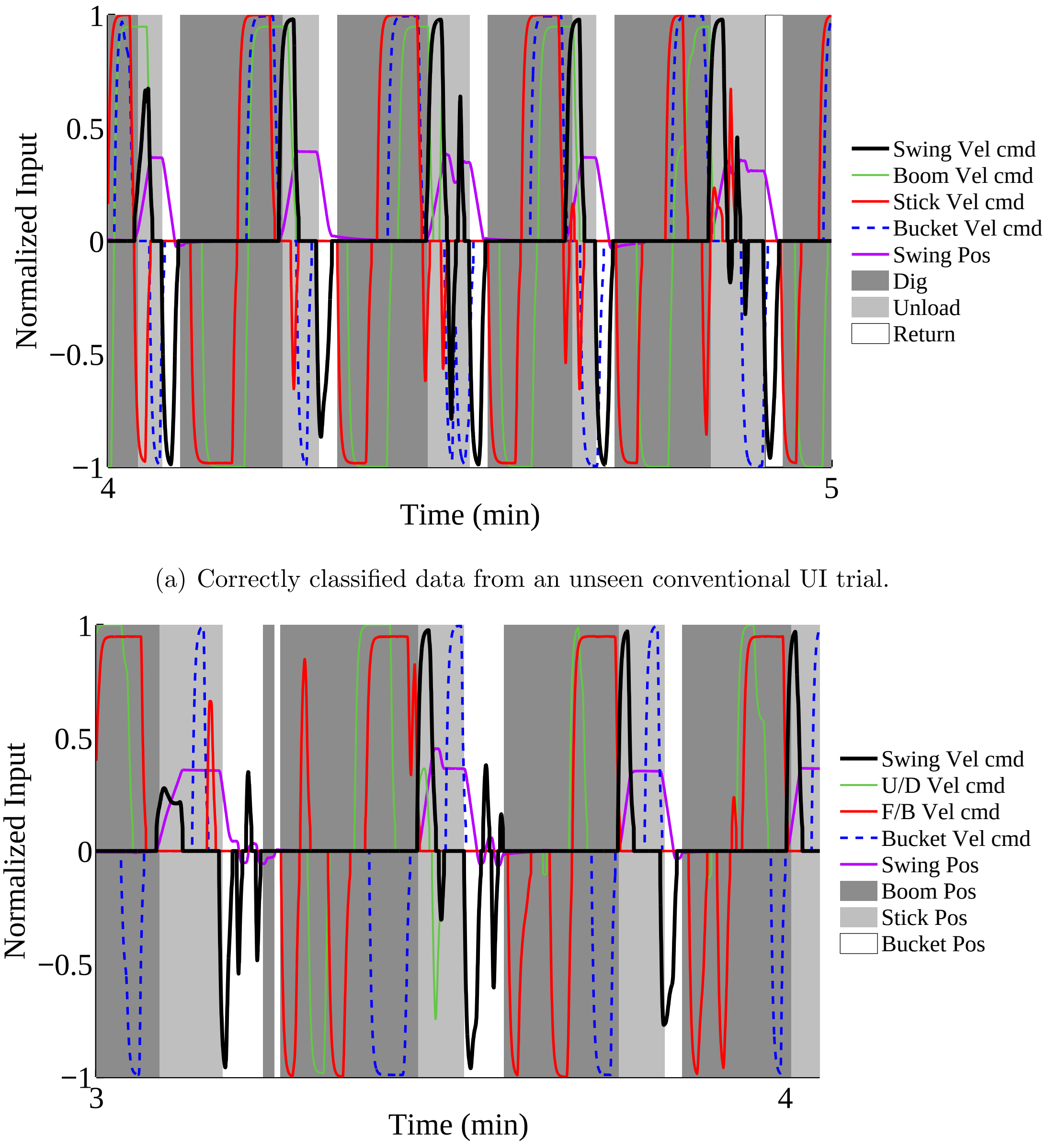
Table 7: Confusion matrix for the feature based conventional ANN. The network was 99.2% accurate for the training data.

		Actual		
		Dig	*Unload*	*Return*
Predicted	*Dig*	4191	3	15
	Unload	10	1183	9
	Return	9	2	578

99.2%

Table 8: Confusion matrix for the feature based CRC ANN. The network was 99.3% accurate for the training data.

		Actual		
		Dig	*Unload*	*Return*
Predicted	*Dig*	4307	20	5
	Unload	4	1095	2
	Return	3	7	557
				99.3%



t

(a) Correctly classified data from an unseen conventional UI trial.

(b) Correctly classified data from an unseen CRC UI trial.

Figure 44: Classifications of unseen data for both controllers.

Table 9: Confusion matrix for the feature based cubic support vector machine for the conventional UI. The network was 99.5% accurate for the training data.

		Actual		
		Dig	*Unload*	*Return*
Predicted	*Dig*	4199	9	2
	Unload	4	1181	3
	Return	7	5	590

99.5%

Table 10: Confusion matrix for the feature based cubic support vector machine for the CRC UI. The network was 98.5% accurate for the training data.

		Actual		
		Dig	*Unload*	*Return*
Predicted	*Dig*	4304	7	3
	Unload	44	1065	13
	Return	18	8	538

98.5%

resulting system information and classifications could then be added to the training data, and used to re-train the network. The result is an updated network, with all of the previous training data as well as data from the current operator. The updated classifier could then begin real time task identification.

8.3 Applications

8.3.1 Adaptive Energy Distribution

In practice, knowing the current subtask being completed using system information could be used to improve operation. Mobile hydraulic machines have various ways of distributing energy during operation. It is relatively common for the operator's commands to require more flow than is available in the system. In this situation, flow sharing valves proportionally lower the flow, and thus the velocity, to each of the excavator functions such that the output function velocities maintain the ratio commanded by the operator [15][10].

One implementation of this technology is to use the real time task identification scheme to perform adaptive energy distribution. This could be accomplished by prioritizing different actuators during different portions of the cycle. For example, when an operator enters the unload phase and begins swinging towards the soil pile, energy could be diverted from a low priority function, the bucket, to a high priority function, the swing. This would enable faster swing than would otherwise be possible.

Further research would be necessary to optimize the different priorities for each subtask. Additionally, the hydraulic circuitry would need to be designed to make this physically possible on a live excavator.

8.3.2 Shared Control

Another potential application of real time task identification is for performing shared control. Shared control consists of a human agent and computer agent; the goal is for the computer agent to drive the human's input commands towards the optimal command [9]. To perform this type of control, it is necessary to estimate the current task in order to develop a notion of what the optimal command for that task is. The problem would have to be slightly reformulated for this application. For this use, the subtasks of the trenching cycle would need to be delineated into motion primitives. Motion primitives are defined by the displacements and velocities of each of the joints. A sequence of motion primitives defines a single trajectory through the workspace. Therefore, the supervised ANN would use the current system information to predict the displacement of each of the actuators through the duration of a primitive. Then, an optimal command could be derived to achieve the estimated joint displacements.

CHAPTER IX

CONCLUSIONS

This thesis' first contribution is the development of a novel method of controlling an excavator that increases operator performance with minimal changes to the current hardware. A human subject experiment demonstrated that the coordinated rate control user interface enabled more effective and efficient operator performance. Further analysis illustrated that the distinct advantage of the CRC UI is that it facilitates more effective digging than the conventional UI.

The second contribution of this thesis is the development of a method to perform real time task identification of an excavator using a supervised artificial neural network. The supervised classifier uses the operator commands and the excavator joint positions to identify which of the three phases of the trenching cycle is being performed. A method for online learning is also presented using a classifier and an algorithm for correcting previous misclassifications.

Future extensions of this research should explore both contributions. First, the CRC UI should be tested among a group of both novice and expert excavator operators on a live excavator. The results of the novice operators should be compared to the results presented in this thesis to check for consistency. The trials should be of longer durations as a way of studying the affects of the CRC UI after learning. It would also be helpful to analyze the ability of expert operators to use CRC due to their strong bias for the conventional UI.

To perform real time task identification on a live excavator, further research would likely be necessary to optimize the features used for the supervised classifier. The accurate real time task identification scheme could then be used to enhance operator performance. To perform adaptive energy distribution, the functions that share energy during the three subtasks would need to be selected and implemented through a hydraulic circuit on a live machine. Ultimately, the task identification algorithm could be implemented real time for adaptive energy distribution, and the resulting operator performance should be compared to standard conventional control.

APPENDIX A

EXPERIMENT SURVEYS

PRE-TEST QUESTIONNAIRE SUBJECT # ___

Your Age: ______

Gender (circle one): Male Female

Handedness (circle one): Left Right

How many hours have you operated an excavator or backhoe?

1) 0 hrs
2) 0-2 hrs
3) 2-5 hrs
4) 5-10 hrs
5) >10 hrs

How many hours have you operated **THIS** simulation before today?

1) 0 hrs
2) 0-1 hrs
3) 1-2 hrs
4) 2-5 hrs
5) >5 hrs

How often do you use a joystick (playing video games, other simulators)?

1) Daily
2) Multiple times a week
3) Weekly
4) Monthly
5) Less than monthly
6) Never

How would you rate your level of experience in playing with or operating joysticks?

1 2 3 4 5

Novice *Expert*

POST-SESSION QUESTIONAIRE SUBJECT # ___

1. How difficult were the controls to learn?

Mode 1:	1	2	3	4	5
Mode 2:	1	2	3	4	5
	Not Difficult		*Somewhat Difficult*		*Very Difficult*

2. How accurately were you able to control the position of the bucket?

Mode 1:	1	2	3	4	5
Mode 2:	1	2	3	4	5
	Not Accurately		*Somewhat Accurately*		*Very Accurately*

3. How confident are you of your ability to operate this excavator simulator?

Mode 1:	1	2	3	4	5
Mode 2:	1	2	3	4	5
	Not Confident		*Somewhat Confident*		*Very Confident*

4. Please provide any additional comments about your experience with the hand controllers.

POST-SESSION QUESTIONAIRE SUBJECT # ___

1. How difficult were the controls to learn?

1 2 3 4 5

Not Difficult *Somewhat Difficult* *Very Difficult*

2. How accurately were you able to control the position of the bucket?

1 2 3 4 5

Not Accurately *Somewhat Accurately* *Very Accurately*

3. How confident are you of your ability to operate this excavator simulator?

1 2 3 4 5

Not Confident *Somewhat Confident* *Very Confident*

4. Please provide any additional comments about your experience with the hand controllers.

REFERENCES

[1] Bishop, C. M., *Neural networks for pattern recognition.* Oxford university press, 1995.

[2] Bishop, C. M., *Neural networks for pattern recognition.* Oxford university press, 1995.

[3] Dietterich, T. G., "Machine learning for sequential data: A review," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 15–30, Springer, 2002.

[4] DiMaio, S. P., Salcudean, S. E., and Reboulet, C., "A virtual environment for the simulation and programming of excavation trajectories," *Presence: Teleoperators and Virtual Environments*, vol. 10, no. 5, pp. 465–476, 2001.

[5] Elton, M. D. and Book, W. J., "Comparison of human-machine interfaces designed for novices teleoperating multi-dof hydraulic manipulators," in *RO-MAN, IEEE*, pp. 395–400, 2011.

[6] Elton, M. D. and Book, W. J., "Operator efficiency improvements from novel human-machine interfaces," in *6th FPNI – PhD Symposium, June 15-19 2010, West Lafayette, Indiana.*, Georgia Institute of Technology, 2010.

[7] Elton, M. D., Enes, A. R., and Book, W. J., "A virtual reality operator interface station with hydraulic hardware-in-the-loop simulation for prototyping excavator control systems," in *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pp. 250–255, IEEE, 2009.

[8] Elton, M. D., "An efficient haptic interface for a variable displacement pump controlled excavator," *M.S. Thesis, Georgia Institute of Technology Woodruff School of Mechanical Engineering*, 2009.

[9] Enes, A. R., "Shared control of hydraulic manipulators to decrease cycle time," *PhD Thesis, Georgia Institute of Technology Woodruff School of Mechanical Engineering*, 2010.

[10] Eriksson, B. and Palmberg, J.-O., "How to handle auxiliary functions in energy efficient, single pump, flow sharing mobile systems," in *The 7th International Fluid Power Conference, 22th-24th March, Aachen, Germany*, vol. 1, pp. 65–78, Apprimus Wissenschafts Verlag, 2010.

[11] Jenkins, O. C. and Mataric, M. J., "Deriving action and behavior primitives from human motion data," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2551–2556, IEEE, 2002.

[12] Johnson, D. W., Lovell, G., and Murray, J. J., "Development of a coordinated motion controller for a front shovel excavator," in *ANS 7th Topical Meeting on Robotics and Remote Systems*, vol. 1, pp. 239–246, 1997.

[13] Kim, W. S., Tendick, F., Ellis, S. R., and Stark, L. W., "A comparison of position and rate control for telemanipulations with consideration of manipulator system dynamics," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 5, pp. 426–436, 1987.

[14] Kwapisz, J. R., Weiss, G. M., and Moore, S. A., "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[15] Lunzman, S. V., "Hydraulic flow priority system," Feb. 13 1996. US Patent 5,490,384.

[16] Mulligan, I. J., Mackworth, A. K., and Lawrence, P. D., "A model-based vision system for manipulator position sensing," in *Interpretation of 3D Scenes, 1989. Proceedings., Workshop on*, pp. 186–193, IEEE, 1989.

[17] Seifert, S. F. and Book, W. J., "Coordinated rate control of a hydraulic excavator using traditional joystick hardware," in *ASME 2015 Dynamic Systems and Control Conference*, pp. V003T45A004–V003T45A004, American Society of Mechanical Engineers, 2015.

[18] Weiss, G. M. and Hirsh, H., "Learning to predict rare events in event sequences.," in *KDD*, pp. 359–363, 1998.

[19] Whitney, D. E., "The mathematics of coordinated control of prosthetic arms and manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 94, no. 4, pp. 303–309, 1972.

[20] Winck, R. C., Elton, M., and Book, W. J., "A practical interface for coordinated position control of an excavator arm," *Automation in Construction*, vol. 51, pp. 46–58, 2015.

[21] Zimmerman, J. D., Pelosi, M., Williamson, C. A., and Ivantysynova, M., "Energy consumption of an ls excavator hydraulic system," in *ASME 2007 International Mechanical Engineering Congress and Exposition*, pp. 117–126, American Society of Mechanical Engineers, 2007.